FLIGHT TICKET PRICE PREDICTION

Submitted by:

VANISREE P G

ACKNOWLEDGMENT

First I would like to thank the Almighty for his wonderful presence with me throughout this project and helped me to make it as a successful one.

For my internship I had the pleasure of working at FILP ROBO Was a great chance for acquired knowledge, personal and Professional development.

I extend whole hearted thanks to FILP ROBO under whom I worked and learned a lot and for enlightening me with their knowledge and experience to grow with the corporate working.

This is a great pleasure to express my deep sense of gratitude and thanks to SME for his valuable ideas, instantaneous help, effective support and continued encouragement which enabled for the successful completion of the project. I also like to thank the data trained mentors and Technical team members for helping me with technical queries.

And these are the following website which I referred for the reference

1. https://www.kaggle.com/
2. https://scikit-learn.org/
3. www.stackoverflow.com
4. www.google.com
5. www.geeksforgeeks.org

INTRODUCTION

Business Problem Framing

With respect to the season flight price also differs in the market, we have seen lot of changes in the Flight ticket price. Now some flight tickets are in demand hence making them costly and some are not in demand hence cheaper. With the change in market due to covid 19 impact, our customers are facing problems with their previous flight price valuation machine learning models. So, they are looking for new machine learning models from new data. We have to make flight ticket price valuation model.

Someone who purchase flight tickets frequently would be able to predict the right time to procure a ticket to obtain the best deal. Many airlines change ticket prices for their revenue management. The airline may increase the prices when the demand is to be expected to increase the capacity. To estimate the minimum airfare, data for a specific air route has been collected including the features like departure time, arrival time and airways over a specific period. Features are extracted from the collected data to apply Machine Learning (ML) models. This paper gives the machine learning regression methods to predict the prices at the given time.

As domestic air travel in India is becoming increasingly popular with different air ticket booking channels coming online these days, passengers are trying to understand how these airline companies make decisions over time about ticket prices. Therefore, many methods are ready to provide the proper time to do so. The customer who buys an air ticket by estimating the price of the airfare is recently proposed. The majority of these strategies make use of sophisticated Computational Intelligence Prediction Models an area of science known as Machine Learning (ML). This paper

highlights the parameters and also includes the guidelines that are important for project work to be developed that is indicated above.

Airline price ticket costs modification terribly dynamically and for a similar flight day by day. It is terribly tough for a customer to buy an air ticket within the lowest value since the value changes dynamically. We addressed the matter regarding the market section level airfare ticket cost forecasting by usage of publicly obtainable datasets and completely unique machine learning model to forecast market section level price cost of airline ticket. The purpose of this study is to raise and analyze the options that influence transportation and to develop and tune models to predict the transportation well ahead.

Planes ticket prices changes as time passes, pulling out the elements which creates the difference. Reporting the correlated and models which is used to price the flight tickets. Then,using that information, building the model which helps passengers to make pull out the ticket to buy and predicting air ticket prices which progresses in the future. Duration, Arrival time, Price, Source, Destination and much more these are the attribute used for flight price prediction.

Motivation for the Problem Undertaken

To understand real world problems where Machine Learning and Data Analysis can be applied to help to predict the prices in various domains to make better decisions with the help of which they can gain profit or can be escaped from any loss which otherwise could be possible without the study of data.

The evaluating system as a result changes the charge depending on time, season, and festive days to change the header or footer on

successive pages. The ultimate aim of the airways is to earn profit whereas the customer searches for the minimum rate. Customers usually try to buy the ticket well in advance of departure date so as to avoid hike in airfare as date comes closer. But actually this is not the fact. The customer may wind up by giving more than they ought to for the same seat.

These days, domestic air travel is becoming more and more common in India. Travellers are trying to learn how these airline companies make choices over time about ticket prices with multiple air travel booking outlets coming online. For a passenger, it is a time-consuming method to search websites for deals and offers. The cost can therefore depend on various variables. This venture uses AI to show the types off light tickets after some time to estimate the costs. Both organizations have the right and the ability at any time to change their ticket prices. By reserving a ticket at the lowest cost, explorer can set aside money. People who have travelled by flight are also aware of the variations in costs. Complex revenue control policies are used by airlines for the introduction of distinctive assessment schemes.

As a result, the appraisal scheme adjusts the fee to adjust the header or footer on successive pages based on time, season, and festive days. The ultimate goal of the airways is to achieve profit ,while the customer is looking for the minimum cost. Usually consumers try to book the ticket well in advance of the departure date to prevent airfare hikes as the date gets closer. But that's not the truth, really. By giving more than they should for the same seat, the customer can finish up.

Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

In the whole research process various mathematical, statistical and analytics modelling has been done. There has been reduction of the columns because few of them was not necessary for the problem solving. And few of them was removed due to very less correlation with dependent variable. Since the dataset contains a lot of features hence feature selection has been also done.

In machine learning, several algorithms are applied to forecast the prices of flight tickets. The algorithms are:Linear regression, Decision tree, Gradient Boosting Regression, and Random Forest Algorithm. These models have been implemented using the python library Sklearn. The parameters like MAE and MSE, RMSE are considered to check the efficiency of these models

Improving the ML structure to predict the mean plane price for the business purpose.For predicting the mean plane price with modification of R squared score, feature selection techniques were proposed in our model.Comparing the production of various ML classifiers which tells the greater plane price prediction task. Facts gathered from website that sells the planes ticket through internet apps. Authors have reported that there is limited public information access which will miss the main target attribute. Final accountable prediction model is improved by two unrelated prediction models such as Random Forest and Multilayer Perceptron.Weights for drifting with R-square value and the main estimation of the metric was used.

Data Sources and their formats

The most critical aspect of this project is the accumulation of knowledge. To prepare the models, the distinct well springs of the data on various sites are used. Sites provide information on the different firms, hours, aircraft, and charges. For data scratching, various sources from API's to customer travel sites are available.

Data is Scrapped from a Yatra.com is an Indian online travel agency and travel search engine web portal for booked a flight ticket. The data is scraped from the Yatra.com website The data descriptions are as follow:- (2103, 10) rows and columns. To predict Flight prices using Regression. I will start by importing all the necessary libraries that we need for this task and import the dataset.

1) Importing libraries

2) Importing the dataset

They are totally 2103 rows and 10 columns in a csvfile. Our target is to find the insights of the data and to do thorough data analysis.

1.Uploading Data set

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib as mlp
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
df = pd.read_csv("Flight price data.csv")
```

```
In [2]: df = pd.read_csv("Flight price data.csv")

In [3]: df

Out[3]:
```

	Unnamed: 0	Date	Airline	Source	Destination	Dep_time	Arr_time	Duration	Total_Stops	Price
0	0	25-jul	Vistara	Chennai	Mumbai	09:45	11:40	1h 55m	Non Stop	8,666
1	1	25-jul	Vistara	Chennai	Mumbai	12:30	14:30	2h 00m	Non Stop	8,666
2	2	25-jul	Go First	Chennai	Mumbai	13:40	15:40	2h 00m	Non Stop	8,666
3	3	25-jul	Vistara	Chennai	Mumbai	20:30	22:35	2h 05m	Non Stop	8,666
4	4	25-jul	Air India	Chennai	Mumbai	15:25	17:10	1h 45m	Non Stop	8,668
...	...	...	...	...	...	...	...	...	...	...
2098	101	27-jul	Vistara	New Delhi	Goa	20:35	12:45	16h 10m	2 Stop(s)	23,171
2099	102	27-jul	Vistara	New Delhi	Goa	20:35	12:45	16h 10m	2 Stop(s)	23,171
2100	103	27-jul	SpiceJet	New Delhi	Goa	19:00	12:55	17h 55m	1 Stop	24,823
2101	104	27-jul	Vistara	New Delhi	Goa	20:40	12:45	16h 05m	2 Stop(s)	26,953
2102	105	27-jul	Vistara	New Delhi	Goa	20:40	12:45	16h 05m	2 Stop(s)	26,953

2103 rows × 10 columns

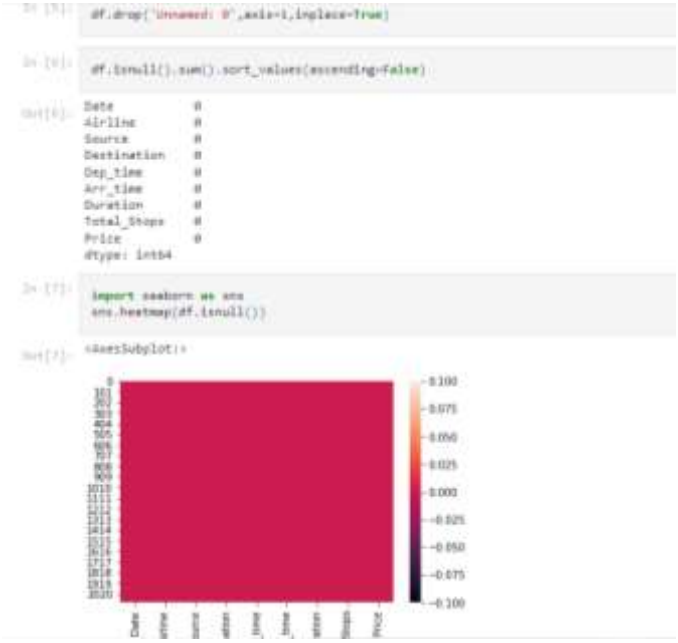
Data Pre-processing

Before building model, the data should be properly pre processed and converted to quality, clean data even the resulting machine learning model will be of great quality .The data pre-processing includes three main parts that is data integration, data cleaning, data transformation. In data integration the data collected from various sources are integrated. In data cleaning process the data containing the null values, unnecessary rows with null values are being cleared. The data transformation includes the feature scaling ,categorical data, etc to set the certain range of data.

The raw data is taken and performed various steps to reduce skewness, outlier, class imbalance and scaling. There were null value was present and removed the values from the dataset. Many outlier removal and skewness removal methods are tested and best method is chosen in order to prevent data loss.

- The dataset contains 2103 rows and 10 columns
- Price is our dependent variable.
- We created new features from old ones.
- There are no null values in the dataset.
- Removed empty cells

a) Checking missing value from the data set.



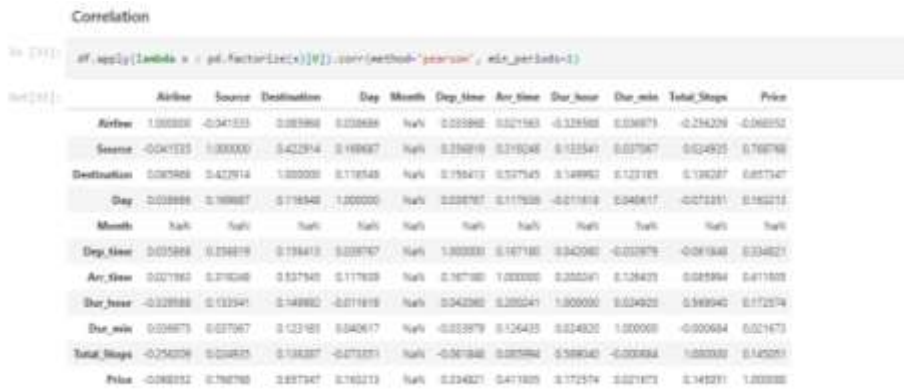
- There were no null value was present in the dataset and there is no outliers are present in the data.

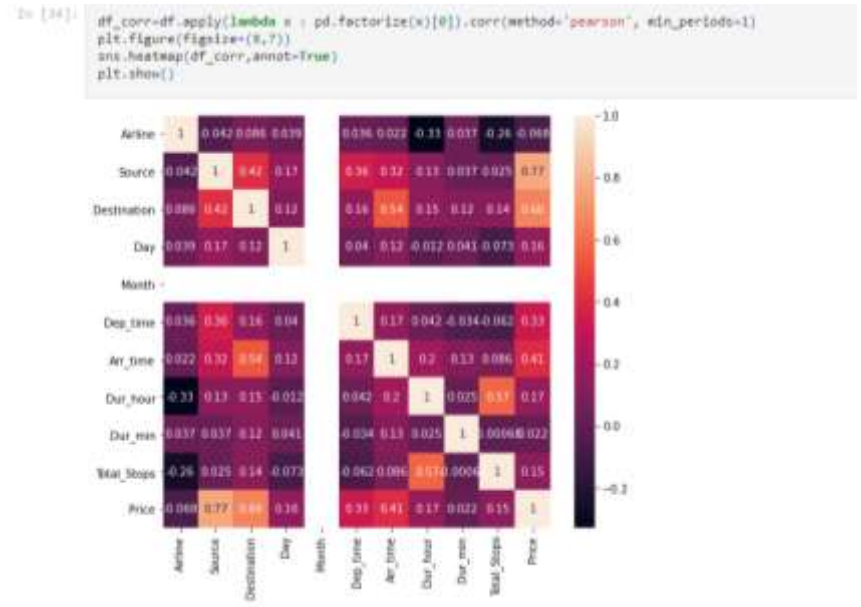
b) CORRELATION

Correlation between all the columns in the datasets. In the correlation Heat map, we have the following observations:

- Arrive time has 41 % correlation with target column which can be considered as a good bond.
- Departure time has 33 % correlation with target column which can be considered as a good bond.
- Duration has 17 % correlation with target column which can be considered as a good bond.
- Day has 16 % correlation with target column which can be considered as a good bond.

- source has 77 % correlation with target column which can be considered as a good bond.
- Airline has -0.068% correlation with target column which can be considered as a negative bond.





Data Inputs- Logic- Output Relationships

The input data contains 2103 rows and 10 columns.

Predictor variable are,

Total_stops, Durtion, Arr_time, Dep_time, Destination, Source, Airline, Date.

Target variable is Price of the flight ticket

Hardware and Software Requirements and Tools Used

Hardware used for doing the project is a ‘Laptop’ with high end specification and stable internet connection .while coming to the software part I had used ‘python jupyter notebook’ for do my python program and data analysis.

Excel file and Microsoft excel are required for the data handling. In jupyter notebook I had imported lot of python libraries are carried to this project.

1.Pandas-a library which is used to read the data ,visualisation and analysis of data.

2.Numpy-used for working with array and various mathematical operations in python.

3.Seaborn- visualization for plotting different type of plot.

4.Matplotlib- It provides an object-oriented API for embedding plots into applications .

Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods)

In machine learning, several algorithms are applied to forecast the prices of flight tickets. The algorithms are: Linear regression, Decision tree, SVR, Gradient Boosting Regression, Ridge and Random Forest Algorithm. These models have been implemented using the

python library Sklearn. The parameters like MAE and MSE, RMSE are considered to check the efficiency of these models.

Regression Model with following algorithms

- Linear Regression
- Decision Tree Regressor
- Random forest regressor
- SVR
- Gradient Boosting Regressor
- Ridge
    - Evaluation metrics

- Mean square error
- Mean absolute error
- R2 score
- Root Mean Squared Error

Testing of Identified Approaches (Algorithms)

Listing down all the algorithms used for the training and testing.

- LR=LinearRegression()
- DT=DecisionTreeRegressor()
- rf=RandomForestRegressor()
- svr=SVR()
- R=Ridge()
- GBR=GradientBoostingRegressor()

Run and Evaluate selected models

Describe all the algorithms used along with the snapshot of their code and what were the results observed over different evaluation metrics. Train-test data splits were conducted. In this situation, we split the data into training and test sets, then fit candidate models on the training set, evaluate and select them on the test set.

1. Linear Regression

i)LinearRegression

```
lR=LinearRegression()
lR.fit(x_train,y_train)
print(lR.score(x_train,y_train))
lR_predict=lR.predict(x_test)
```

```
0.3665026175034558
```

```
print('Mean Absolute Error: ',metrics.mean_absolute_error(y_test,lR_predict))
print('Mean Squared Error: ',metrics.mean_squared_error(y_test,lR_predict))
print('Root Mean Squared Error: ',np.sqrt(metrics.mean_squared_error(y_test,lR_predict)))
print('Explained Variance Score: ',metrics.explained_variance_score(y_test,lR_predict))
print('r2_score:',r2_score(y_test,lR_predict))
```

```
Mean Absolute Error:  11.89677183920881
Mean Squared Error:  245.94951129383346
Root Mean Squared Error:  15.682775175305
Explained Variance Score:  0.41795102159751174
r2_score: 0.4152533812720548
```

C-V score

```
lR_cv=cross_val_score(lR,x,y, cv = 7).mean()
lR_cv
```

```
0.21430167978793884
```

2. Random forest

ii)RandomForestRegressor

```
from sklearn.ensemble import RandomForestRegressor
rf=RandomForestRegressor()
rf.fit(x_train,y_train)
predictions1=rf.predict(x_test)
print(rf.score(x_train,y_train))
```

```
0.9667287125144835
```

```
print('Mean Absolute Error: ',metrics.mean_absolute_error(y_test,predictions1))
print('Mean Squared Error: ',metrics.mean_squared_error(y_test,predictions1))
print('Root Mean Squared Error: ',np.sqrt(metrics.mean_squared_error(y_test,predictions1)))
print('Explained Variance Score: ',metrics.explained_variance_score(y_test,predictions1))
print('r2_score:',r2_score(y_test,predictions1))
```

```
Mean Absolute Error:  5.417472150578962
Mean Squared Error:  84.98285565693553
Root Mean Squared Error:  9.218331484484201
Explained Variance Score:  0.7975725461181131
r2_score: 0.7974843612223489
```

```
rf_cv=cross_val_score(rf,x,y, cv = 4).mean()
rf_cv
```

```
0.2749264453295447
```

3. Decision Tree Regression

iii)DecisionTreeRegressor

```
from sklearn.tree import DecisionTreeRegressor

DTR=DecisionTreeRegressor()
DTR.fit(x_train,y_train)
print(DTR.score(x_train,y_train))
DTR_PRED=DTR.predict(x_test)
```

```
0.9972179639436564
```

```
print('Mean Absolute Error: ',metrics.mean_absolute_error(y_test,DTR_PRED))
print('Mean Squared Error: ',metrics.mean_squared_error(y_test,DTR_PRED))
print('Root Mean Squared Error: ',np.sqrt(metrics.mean_squared_error(y_test,DTR_PRED)))
print('Explained Variance Score: ',metrics.explained_variance_score(y_test,DTR_PRED))
print('r2_score:',r2_score(y_test,DTR_PRED))
```

```
Mean Absolute Error:  4.898514900885491
Mean Squared Error:  185.8052361782733
Root Mean Squared Error:  10.191625791318459
Explained Variance Score:  0.7512883627869802
r2_score: 0.7512895139906052
```

```
DTR_cv=cross_val_score(DTR,x,y, cv = 11).mean()
DTR_cv
```

```
0.15613228488744124
```

- **SVR**

```
from sklearn.svm import SVR

svr=SVR()
svr.fit(x_train,y_train)
print(svr.score(x_train,y_train))
svr_predict=svr.predict(x_test)
```

```
0.25254517669953438
```

```
print('Mean Absolute Error: ',metrics.mean_absolute_error(y_test,svr_predict))
print('Mean Squared Error: ',metrics.mean_squared_error(y_test,svr_predict))
print('Root Mean Squared Error: ',np.sqrt(metrics.mean_squared_error(y_test,svr_predict)))
print('Explained Variance Score: ',metrics.explained_variance_score(y_test,svr_predict))
print('r2_score:',r2_score(y_test,svr_predict))
```

```
Mean Absolute Error:  12.461182877926767
Mean Squared Error:  297.3889667973557
Root Mean Squared Error:  17.244969880418372
Explained Variance Score:  0.33525204425895516
r2_score: 0.2985372922943853
```

```
svr_cv=cross_val_score(svr,x,y, cv = 4).mean()
svr_cv
```

```
0.04301647633816455
```

4. Gradient Boosting Regression

```
from sklearn.ensemble import GradientBoostingRegressor

GBR=GradientBoostingRegressor()
GBR.fit(x_train,y_train)
print(GBR.score(x_train,y_train))
GBR_PRED=GBR.predict(x_test)
```

```
0.7684872928471248
```

```
print('Mean Absolute Error: ',metrics.mean_absolute_error(y_test,GBR_PRED))
print('Mean Squared Error: ',metrics.mean_squared_error(y_test,GBR_PRED))
print('Root Mean Squared Error: ',np.sqrt(metrics.mean_squared_error(y_test,GBR_PRED)))
print('Explained Variance Score: ',metrics.explained_variance_score(y_test,GBR_PRED))
print('r2_score:',r2_score(y_test,GBR_PRED))
```

```
Mean Absolute Error:  7.355785583316015
Mean Squared Error:  108.99052609363192
Root Mean Squared Error:  10.440235023135982
Explained Variance Score:  0.7480431041418741
r2_score: 0.7390688022551275
```

```
GBR_cv=cross_val_score(GBR,x,y, cv = 11).mean()
GBR_cv
```

```
0.5204303022003908
```

5. **Ridge regression**

```
from sklearn.linear_model import Ridge

R=Ridge()
R.fit(x_train,y_train)
print(R.score(x_train,y_train))
R_predict=R.predict(x_test)
```

```
0.34648191366958614
```

```
print('Mean Absolute Error: ',metrics.mean_absolute_error(y_test,R_predict))
print('Mean Squared Error: ',metrics.mean_squared_error(y_test,R_predict))
print('Root Mean Squared Error: ',np.sqrt(metrics.mean_squared_error(y_test,R_predict)))
print('Explained Variance Score: ',metrics.explained_variance_score(y_test,R_predict))
print('r2_score:',r2_score(y_test,R_predict))
```

```
Mean Absolute Error:  11.889340454009202
Mean Squared Error:  245.69948780858552
Root Mean Squared Error:  15.674804237711728
Explained Variance Score:  0.4141885569745034
r2_score: 0.4138407660855284
```

```
R_cv=cross_val_score(R,x,y, cv = 7).mean()
R_cv
```

```
0.2346779189361718
```

Evaluating the model accuracy is an essential part of the process of creating machine learning models to describe how well the model is performing in its predictions. The MSE, MAE, and RMSE metrics are mainly used to evaluate the prediction error rates and model performance in regression analysis.

- MAE (Mean absolute error) represents the difference between the original and predicted values extracted by averaged the absolute difference over the data set.
- MSE (Mean Squared Error) represents the difference between the original and predicted values extracted by squared the average difference over the data set.
- RMSE (Root Mean Squared Error) is the error rate by the square root of MSE.

After evaluating the model based on MAE, MSE, RMSE, EVS, R2 SCORE the best model choose for hyper parameter tuning are RandomForestRegressor, DecisionTreeRegressor.

A. RandomForestRegressor
B. DecisionTreeRegressor

Hyper parametertuning

A. RandomForestRegressor

```
In [68]: RF=RandomForestRegressor()
         param={
              'n_estimators':[100,200,],
             'criterion':["mse", "mae"],
             'min_samples_split':[2],
            'min_samples_leaf' : [1],
         }
```

```
In [69]: from sklearn.model_selection import GridSearchCV
         RF_grid=GridSearchCV(RandomForestRegressor(),param,cv=4)
```

```
In [70]: RF_grid.fit(x_train,y_train)
         RF_grid_PRED=RF_grid.best_estimator_.predict(x_test)
```

```
In [71]: RF_grid.best_params_
```

```
Out[71]: {'criterion': 'mse',
          'min_samples_leaf': 1,
          'min_samples_split': 2,
          'n_estimators': 200}
```

```
In [72]: rf=RandomForestRegressor(criterion='mse',min_samples_leaf=1,min_samples_split=2,n_estimators=200)
         rf.fit(x_train,y_train)
         rf_predictions=rf.predict(x_test)
```

```
In [73]: import numpy as np
         from sklearn import metrics
         print('Mean Absolute Error: ',metrics.mean_absolute_error(y_test,rf_predictions))
         print('Mean Squared Error: ',metrics.mean_squared_error(y_test,rf_predictions))
         print('Root Mean Squared Error: ',np.sqrt(metrics.mean_squared_error(y_test,rf_predictions)))
         print('Explained Variance Score: ',metrics.explained_variance_score(y_test,rf_predictions))
         print('r2_score:',r2_score(y_test,rf_predictions))
```

```
Mean Absolute Error:  5.356798154843619
Mean Squared Error:  82.15186160414999
Root Mean Squared Error:  9.063766413812196
Explained Variance Score:  0.8041543350112451
r2_score: 0.8040153308822685
```

B. Decision Tree Regression

```
In [74]: from sklearn.model_selection import GridSearchCV
         parameter = {'max_features':['auto', 'sqrt', 'log2'],
                      'criterion':['mse','friedman_mse','mae', 'poisson'],
                      'splitter':['best', 'random']}
         GCV = GridSearchCV(DecisionTreeRegressor(),parameter,cv=5)
```

```
In [75]: GCV.fit(x_train,y_train)
```

```
Out[75]: GridSearchCV(cv=5, estimator=DecisionTreeRegressor(),
                      param_grid={'criterion': ['mse', 'friedman_mse', 'mae', 'poisson'],
                                  'max_features': ['auto', 'sqrt', 'log2'],
                                  'splitter': ['best', 'random']})
```

```
In [76]: GCV.best_params_
```

```
Out[76]: {'criterion': 'friedman_mse', 'max_features': 'auto', 'splitter': 'best'}
```

```
In [77]: from sklearn.tree import DecisionTreeRegressor

         DTR=DecisionTreeRegressor(criterion='friedman_mse',splitter= 'best',max_features='auto')
         DTR.fit(x_train,y_train)
         DTR_final=DTR.predict(x_test)
```

```
print("Score of Model is",DTR.score(x_train,y_train))
print("Mean Absolute Error", mean_absolute_error(y_test,DTR_final))
print("Root Mean Squared Error", (mean_squared_error(y_test,DTR_final))**0.5 )
```

```
Score of Model is 0.9971179639636604
Mean Absolute Error 5.384139465533224
Root Mean Squared Error 10.972176243928
```

```
print('Mean Absolute Error:  ',metrics.mean_absolute_error(y_test,DTR_final))
print('Mean Squared Error:  ',metrics.mean_squared_error(y_test,DTR_final))
print('Root Mean Squared Error:  ',np.sqrt(metrics.mean_squared_error(y_test,DTR_final)))
print('Explained Variance Score:  ',metrics.explained_variance_score(y_test,DTR_final))
print('r2_score:',r2_score(y_test,DTR_final))
```

```
Mean Absolute Error:   5.384139465533224
Mean Squared Error:   120.38865148393082
Root Mean Squared Error:   10.972176243928384
Explained Variance Score:   0.7138029468332122
r2_score: 0.7127961610864142
```

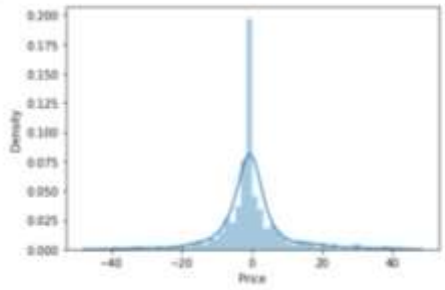
The best model after hyper parameter tuning is Random Forest Regressor

```
print("FINAL MODEL")
print("-----------------------------------------")
print('Mean Absolute Error:  ',metrics.mean_absolute_error(y_test,rf_predictions))
print('Mean Squared Error:  ',metrics.mean_squared_error(y_test,rf_predictions))
print('Root Mean Squared Error:  ',np.sqrt(metrics.mean_squared_error(y_test,rf_predictions)))
print('Explained Variance Score:  ',metrics.explained_variance_score(y_test,rf_predictions))
print('r2_score:',r2_score(y_test,rf_predictions))
```

```
FINAL MODEL
-----------------------------------------
Mean Absolute Error:  5.356798154843619
Mean Squared Error:  82.15186160414099
Root Mean Squared Error:  9.063766413812196
Explained Variance Score:  0.8041543350112451
r2_score: 0.8040153308822685
```

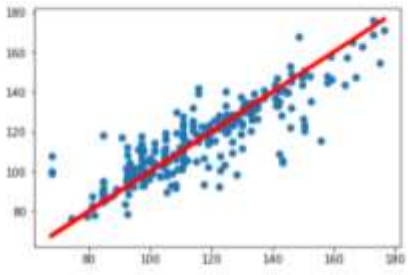
```
sns.distplot(y_test-rf_predictions)
```

```
<AxesSubplot:xlabel='Price', ylabel='Density'>
```




```
plt.scatter(y_test,rf_predictions)
plt.plot(y_test,y_test,linewidth=4,color='Red')
```

```
[<matplotlib.lines.Line2D at 0x1fb3db13480>]
```



7.SAVING THE MODEL

```
import joblib
joblib.dump(rf,"final model.pkl")
```

```
['final model.pkl']
```

In our project we had implemented various Machine Learning Algorithms such as Linear Regression, Decision Tree Regression, Random Forest Regression and compared the accuracy of results based on our test data set. Based on the various accuracy levels we find that Random Forest Regression gives the highest accuracy i.e. 80%. Therefore we selected Random Forest Regression and created User Interface based on it.

Visualizations

After cleaning the data, we can visualize data and better understand the relationships between different variables. There are many more visualizations that you can do to learn more about your dataset, like scatterplots, histograms, boxplots.

- The Analysis of flight Price. The price of the ticket

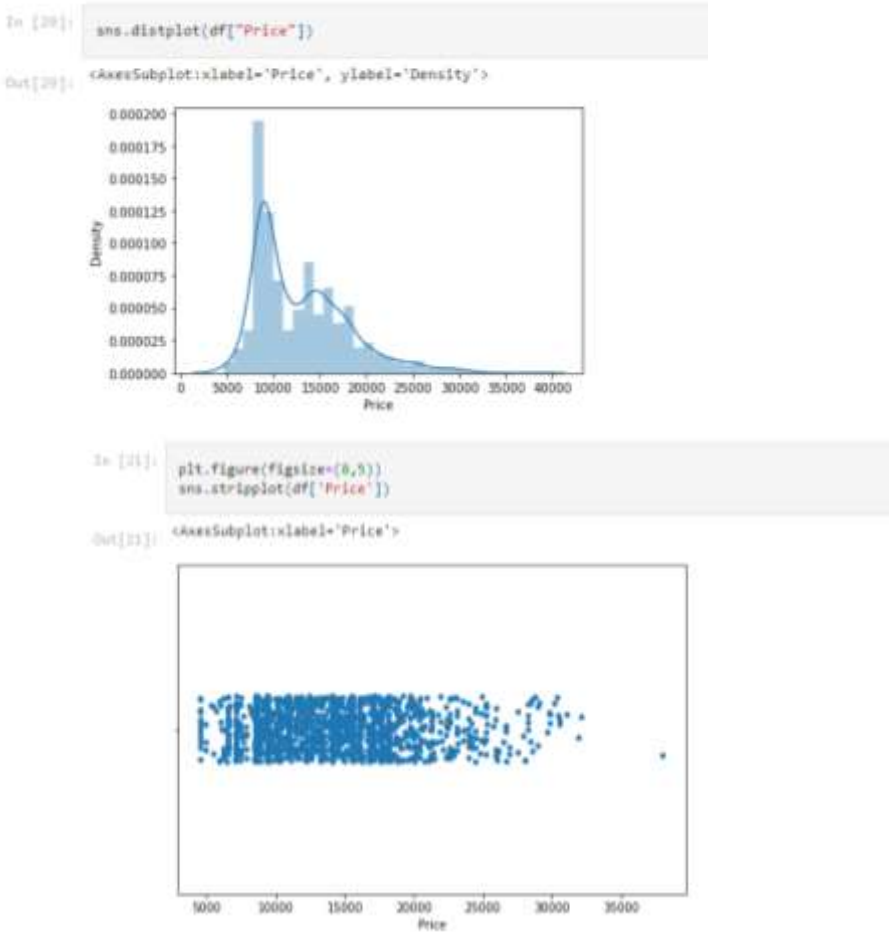


Fig. 1&2: The Analysis price of the ticket

The source from which the service begins

- In observation from New Delhi has more Flight service held compare with all other cities.
- Kochi and Pune have low flight service.

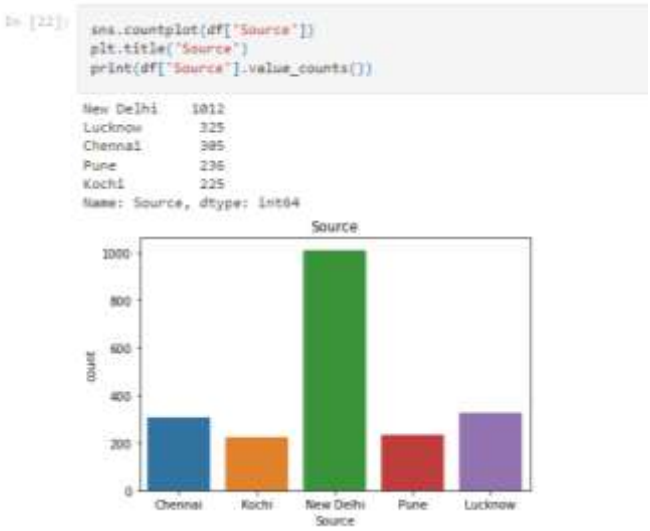


Fig. 3: The Analysis of source from which the service begins

- In observation from fig:4 Mumbai has more Flight service end point held compare with all other cities.
- Goa and Bangalore have low flight service.

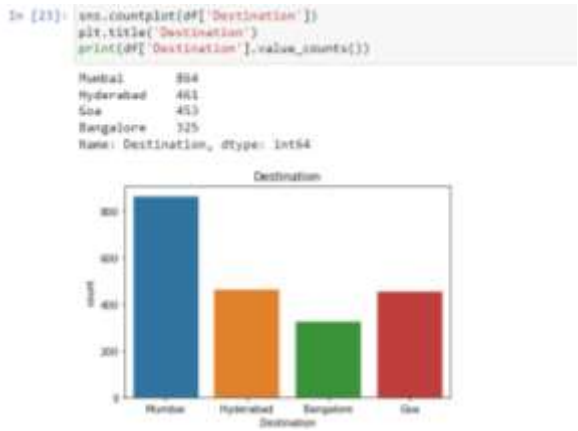


Fig. 4: The Analysis of The destination where the service ends.

- For one stop flight are higher compare with others.
- Only 20% flight are non-stop and 2 stop service

```
In [23]: fig=plt.figure(figsize=(5,3))
         sns.countplot(df['Total_Stops'])
         print(df['Total_Stops'].value_counts())
         plt.show()

1 Stop       1372
2 Stop(s)     396
Non Stop      308
3 Stop(s)      27
Name: Total_Stops, dtype: int64
```

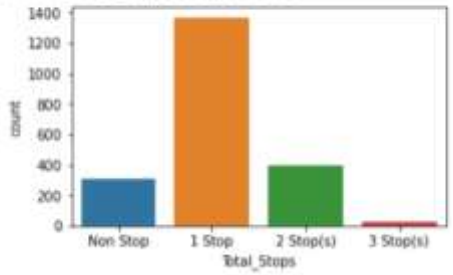


Fig. 5: The Analysis of Total stops between the source and destination.

- Vistara and IndiGo are the higher service provide.
- Air Asia has 2.3% very low service compare with other.

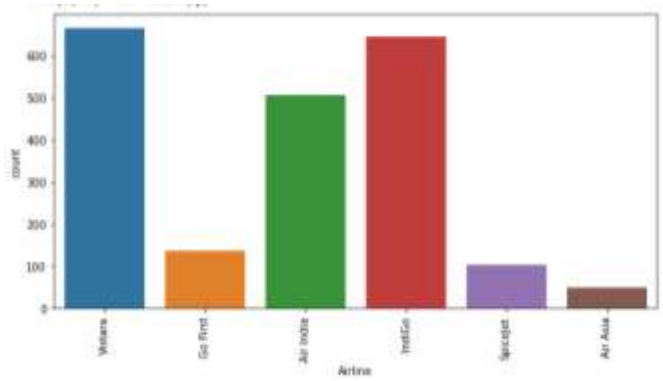


Fig. 6: The Analysis of The name of the airline.

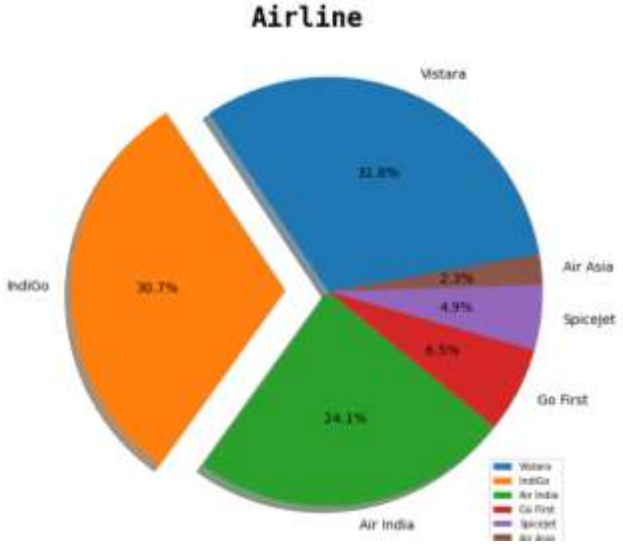


Fig.7: pie chart for The name of the airline.

- The source from which the service begin

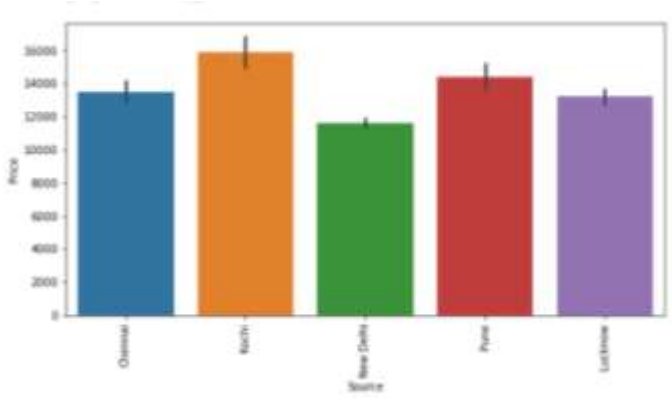


Fig. 8: The Analysis between Source and Price

- Total stops between the source and destination
- 2 stops and 3 stops have high flight price rate compare with others.
- 1 stop and Non stop have low flight price rate.

```
In [30]: plt.figure(figsize=(10,5))
         sns.barplot(df['Total_Stops'],df['Price'])

Out[30]: <AxesSubplot:xlabel='Total_Stops', ylabel='Price'>
```

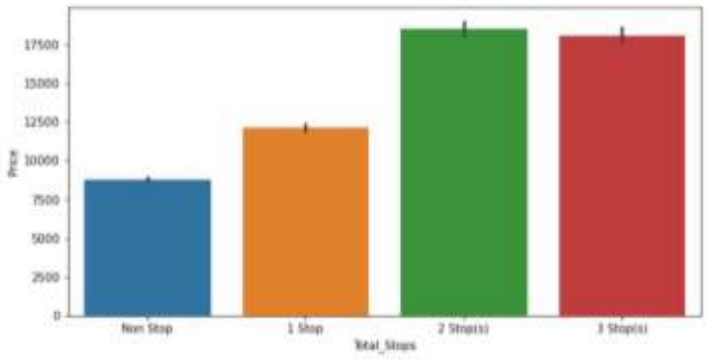


Fig. 9: The Analysis between Total stop and Price

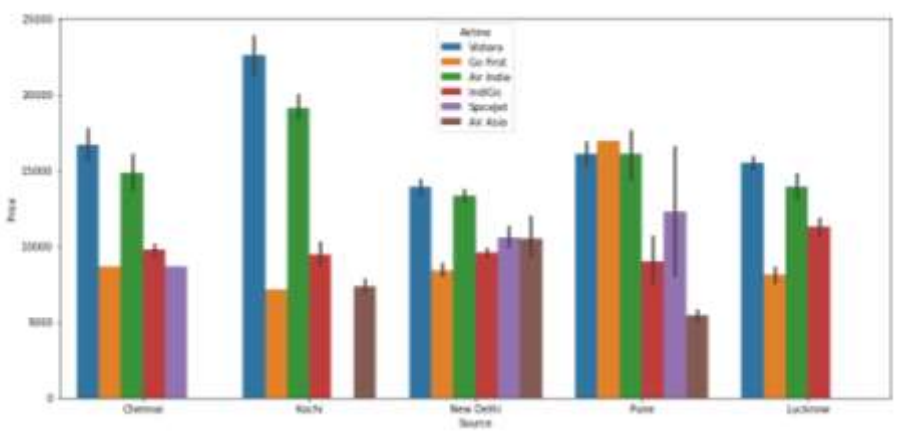


Fig. 10: The Analysis between source and Price.

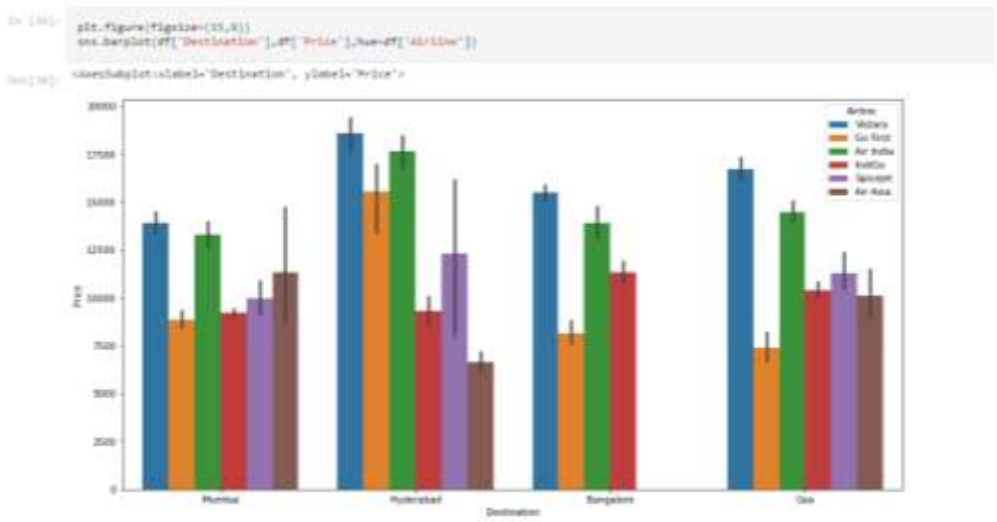


Fig. 11: The Analysis between Destination and Price.

- 24 Sunday and 25 Monday have very high price compare other day.
- Vistara and Air india are high price rated flight on eveyday.
- Indigo and Go First are low price flight .

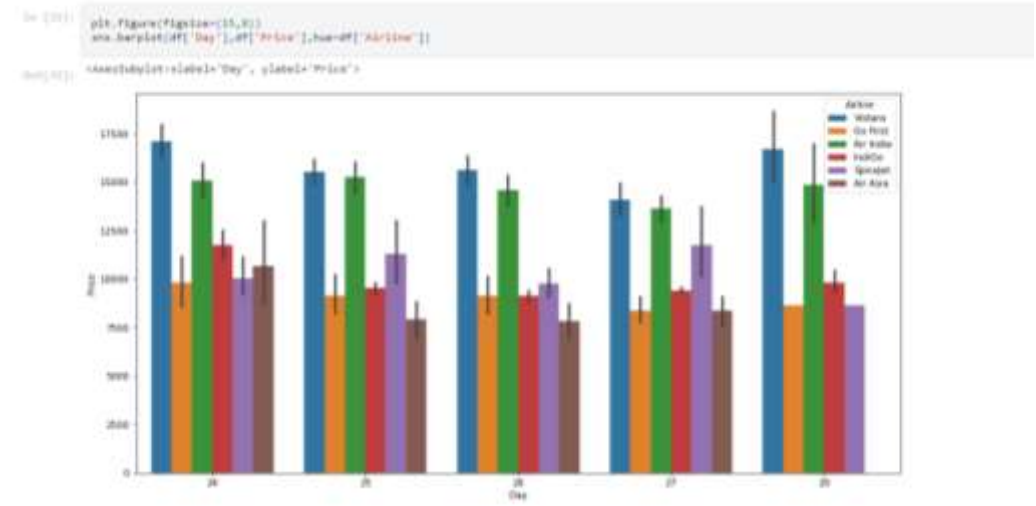


Fig. 12: The Analysis between Day and Price

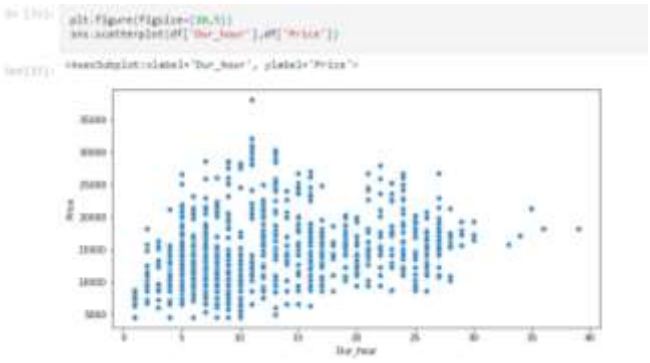
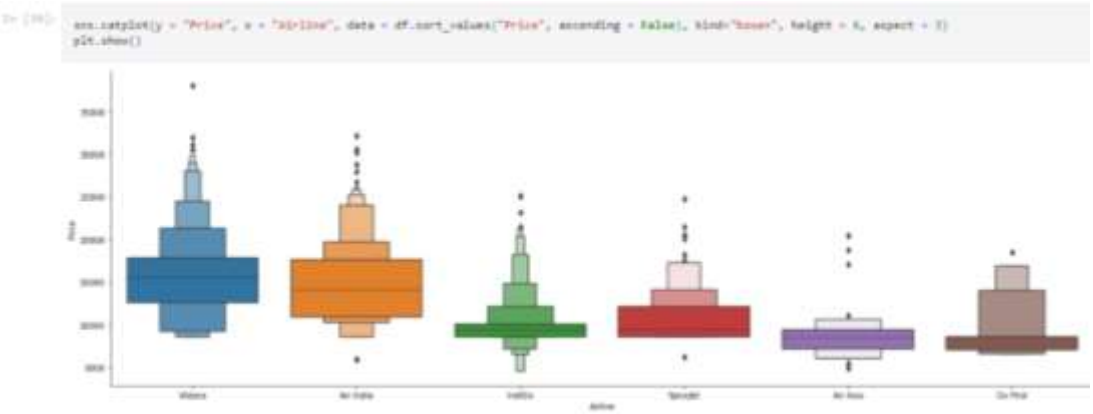


Fig. 13: The Analysis between Duration and Price.

➢ The Analysis between Airline and Price.

The observation is Vistara is having the highest price of airline compared to other airlines. The second highest is Air India airline and all other airline prices are almost similar. Through this analysis the advantage is that the customers has the option to choose the various airlines with different price and comfort with their budget to travel the new places and explore the world.



```
In [31]: sns.barplot(df['Airline'],df['Price'])
         plt.xticks(rotation=90)

Out[31]: (array([0, 1, 2, 3, 4, 5]),
          [Text(0, 0, 'Vistara'),
           Text(1, 0, 'Go First'),
           Text(2, 0, 'Air India'),
           Text(3, 0, 'IndiGo'),
           Text(4, 0, 'SpiceJet'),
           Text(5, 0, 'Air Asia')])
```

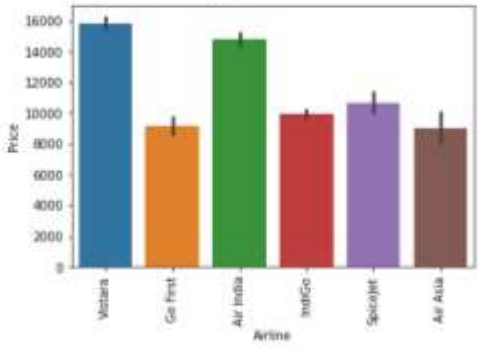
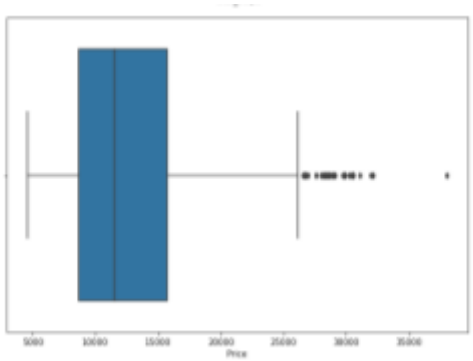


Fig.14,15&16: The Analysis between Airline and Price.

- Checking the outliers



- Checking and Removing Skewness

```
In [40]: df.skew()

Out[40]: Airline        -0.133018
         Source         -0.787096
         Destination    -0.453672
         Day             0.792582
         Month           0.000000
         Dep_time        0.038357
         Arr_time       -0.118705
         Dur_hour        0.752932
         Dur_min         0.179403
         Total_Stops     1.456016
         Price           1.874089
         dtype: float64
```

Using Square root transformation method to reduce Skewness

```
In [41]: #calculating the square root for data

         df1 = np.sqrt(df)

         df1.head(10)
```

	Airline	Source	Destination	Day	Month	Dep_time	Arr_time	Dur_hour	Dur_min	Total_Stops	Price
0	1.224669	0.0	1.732051	5.0	2.645751	7.211103	7.937254	1.000000	7.416198	1.732051	55.091033
1	1.224669	0.0	1.732051	5.0	2.645751	8.888194	9.486833	1.414214	8.000000	1.732051	55.091033
2	1.414214	0.0	1.732051	5.0	2.645751	8.426081	10.049876	1.414214	0.000000	1.732051	55.091033
3	1.224669	0.0	1.732051	5.0	2.645751	12.206556	13.379087	1.414214	2.236068	1.732051	55.091033
4	1.000000	0.0	1.732051	5.0	2.645751	10.148891	10.862780	1.000000	6.708204	1.732051	55.102994
5	1.732051	0.0	1.732051	5.0	2.645751	2.000000	4.242641	1.000000	7.071068	1.732051	55.102994
6	1.732051	0.0	1.732051	5.0	2.645751	9.000000	8.831761	1.000000	7.071068	1.732051	55.102994
7	1.732051	0.0	1.732051	5.0	2.645751	8.544004	9.000000	1.000000	7.071068	1.732051	55.102994
8	1.732051	0.0	1.732051	5.0	2.645751	8.124404	8.831761	1.000000	7.071068	1.732051	55.102994
9	1.732051	0.0	1.732051	5.0	2.645751	11.357817	12.083046	1.000000	7.071068	1.732051	55.102994
10	1.732051	0.0	1.732051	5.0	2.645751	13.416075	13.638182	1.000000	7.071068	1.732051	55.102994
11	1.000000	0.0	1.732051	5.0	2.645751	4.242641	5.567764	1.000000	7.416198	1.732051	55.102994
12	1.732051	0.0	1.732051	5.0	2.645751	12.247449	13.038405	1.414214	2.236068	1.732051	55.102994
13	1.732051	0.0	1.732051	5.0	2.645751	7.000000	7.937254	1.414214	3.162278	1.732051	55.102994
14	1.732051	0.0	1.732051	5.0	2.645751	11.789826	12.206556	1.732051	7.071068	0.000000	55.102994

```
In [42]: df1.skew()

Out[42]: Airline        -0.788851
         Source         -1.386350
         Destination    -1.150431
         Day             0.726101
         Month           0.000000
         Dep_time       -0.482686
         Arr_time       -0.683590
         Dur_hour        0.191812
         Dur_min        -0.691045
         Total_Stops     0.949904
         Price           0.660609
         dtype: float64
```

Skewness is reduced

Observations

- Airfare varies according to the day of the week of travel. It is higher for weekends and Monday and slightly lower for the other days.
- The airfare varies depending on the time of departure, making timeslot used in analysis an important parameter.
- The time duration plays the important role for making the decision to board the flight with best price. With the limited amount of time the best price can be chosen by the passengers. Everybody can afford the flight ticket with best price and best offers
- That travellers can see the highest to lowest prices to know the price differences in each airline. Passengers can share the review of the flight on the each airline websites, so that the other passengers gets the benefit out of it.
- That duration (or distance) plays a major role in affecting air ticket prices, but we see no such pattern here, as there must be pattern here and other significant factors affecting air fare like type of airline, destination of flight, date of journey of flight (higher if collides with a public holiday).
- Through this analysis the advantage is that passengers might want to reach the destination sooner, so the duration plays the important role to reach sooner with good amount of price.
- Through this analysis the advantage is that the Journey Day plays the important role because the flight charges for weekdays in might be lesser and weekends the price might be higher based on the offers and airlines chosen to board the flight.

ANALYSIS ON THE DATA.

1. Do airfares change frequently?

Ans: Yes, Based on Season and Demand of the flight Tickets Price changes regularly.

2. Do they move in small increments or in large jumps?

Ans. large jumps, If you buy a ticket before few of date of journey you will get arround 44% high price.

3. Do they tend to go up or down over time?

Ans. Price of the flight goes up only down over time but its not reduces the p rice over time

4. What is the best time to buy so that the consumer can save the most by taki ng the least risk?

Ans. Consumer should buy the ticket before 30 days of date of joureney so th ey can save up to 45% to 50%

CONCLUSION

Key Findings and Conclusions of the Study

From this dataset I get to know that each feature play a very import role to understand the data. Data format plays a very important role in the visualization and Appling the models and algorithms .Importance of removing the skewness and outlier.

There are many systems which uses different machine learning algorithms such as Linear Regression (LR), Decision Tree, Support Vector Machine (SVM), Random Forest Algorithm, etc for predicting the price for flight ticket. In this ML based system, we are using Random Forest Algorithm which gives more accuracy in predicting the airfare. Considering the features such as departure time, the number of days left for departure and time of the day it will give the best time to buy the ticket .This system also helps the buyer to buy the flight ticket at lower price. It is easy to use and it gives more accuracy in prediction. It requires less time for prediction and it helps in reduction of over fitting. Travellers can save money if they choose to buy the ticket when its price is the lowest. It gives the customer the best time to buy a flight ticket for the desired destination and a period

Evaluating the algorithmic rule, a dataset is collected, pre-processed, performed data modelling and studied a value difference for the number of restricted days by the passengers for travelling .Machine Learning algorithms with square measure for forecasting the accurate fare of airlines and it gives accurate value of plane price ticket at limited and highest value. Information is collected from yatra websites that sell the flight tickets therefore restricting data which are often accessed .The results obtained by the random forest and decision tree algorithm has better accuracy, but best accuracy is predicted by random forest algorithm as shown is the above analysis. Accuracy of the model is also forecasted by the R-squared value.

Learning Outcomes of the Study in respect of Data Science

Learnt how to process the large number of data. Tried and learnt more about distribution of the data. The power of visualization is helpful for the understanding of data into the graphical representation its help me to understand that what data is trying to say, Data cleaning is one of the most important step to remove missing value or null value fill it by mean median or by mode or by 0.Setting a good parameters is more important for the model accuracy. Finding a best random state played a vital roll in finding a better model.

Flight ticket prices can be something hard to guess, today we might see a price, check out the price of the same flight tomorrow, it will be a different story. We might have often heard travellers saying that flight ticket prices are so unpredictable. As data scientists, we are going to prove that given the right data anything can be predicted. So the collected train data should be accurate if not it may result in wrong prediction. And also it is necessary to update the train data time to time for best results.

Limitations of this work and Scope for Future Work

The techniques to increase the speed of the model need to be constructed. In Upcoming days when huge amount of information is accessed as in detailed information in the dataset, the expected results in future are highly correct. For further research anyone desire to expand upon it ought to request different sources of historical data or be a lot of organized in collection knowledge manually over amount of your time to boot, a lot of different combination of plane are

going to be traversed. There is whole possibility that planes differ their execution ideas consisting characteristics of the plane. At last, it is curious to match our model accuracy with that of the business models accuracy offered nowadays.

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 .With the help of our project the travellers can find out the right time to buy their tickets at the lowest cost and also can plan accordingly. So to get result with maximum accuracy regression analysis is used. From the studies, the feature that influences the price ticket are to be considered. In future the details about number of available seats can improve the performance of the model

We might have often heard travellers saying that flight ticket prices are so unpredictable. As data scientists, we are going to prove that given the right data anything can be predicted. So the collected train data should be accurate if not it may result in wrong prediction. And also it is necessary to update the train data time to time for best results