FLIP ROBO

FLIGHT PRICE PREDICTION

Submitted By – Akshay Shingavi

Acknowledgment

I want to extend my sincere gratitude to my SME team and the "Flip Robo Technologies" team for allowing me to work on the "Flight Price Prediction" project. I also want to extend my sincere gratitude to my academic team, "Data Trained." Their recommendations and guidance have enabled me to effectively complete this job. The research I conducted for this assignment also enabled me to learn a great deal of fresh information.

REFERENCES

1. Flip Robo Technologies provided the entire necessary dataset and information.

2. https://www.kaggle.com

INTRODUCTION

A. Business Problem Framing

To maximize income, the airline industry employs dynamic pricing tactics that are among the most complex since they are based on secret factors and proprietary algorithms. Because of this, airline firms employ sophisticated algorithms to determine the cost of aircraft tickets. The cost of the plane ticket is influenced by several different elements.

All of the criteria are known to the vendor, but purchasers may only access a small amount of information that is insufficient to forecast flight costs. The ideal time to purchase the ticket will be determined by factors like departure time, arrival time, and time of day.

B. Conceptual Background of the Domain Problem

Travel ticket costs may be difficult to predict; while we may see a price today, when we check the cost of the same flight tomorrow, it will likely be different. Perhaps you've heard passengers complain frequently about how unpredictable the cost of airline tickets is.

Anyone who has purchased a plane ticket is aware of how costs may change suddenly. The price of the least costly ticket for a specific flight decreases with time.

C. Review of Literature

A literature study examines pertinent literature to learn more about the variables that are crucial for forecasting the market price of airline tickets. In this article, we explain several applications and methodologies that motivated us to develop our supervised ML algorithms to forecast the cost of airline tickets in various locations. We conducted a thorough investigation into the fundamental concepts behind our project and used those concepts to inform the data-collecting process by doing web scraping on the www.makemytrip.com website, which serves as an online travel agency.

D. Motivation for the Problem Undertaken

The quickest mode of transportation, flying can shorten a trip's duration by hours or even days. But we are aware of how wildly the pricing might fluctuate. To assist people in finding the best tickets depending on their demands, I was interested in the listings for Flight Fares Prediction. Additionally, to gain practical experience and understand how data scientists approach and do their work in a company from beginning to end.

ANALYTICAL PROBLEM FRAMING

A. Mathematical/ Analytical Modelling of the Problem

We must create a machine learning model that forecasts the cost of airline tickets that is effective and efficient. Therefore, "Price," a continuous variable, is what we are aiming for. We must use regression methods to forecast the outcomes because it is a regression problem. Three phases make up this project:

1. I used web scraping to get flight data from the well-known website www.makemytrip.com, where I discovered more flight characteristics than on other websites and fetched information for various places. We must create the model necessary to forecast the cost of airline tickets.

2. After cleaning the data I have done some analysis on the data by using different types of visualizations.

3. After collecting the data, I built a machine-learning model. Before model building, have done all data pre-processing steps. The complete life cycle of data science that I have used in this project is as follows:

- Data Cleaning
- Exploratory Data Analysis
- Data Pre-processing
- Model Building
- Model Evaluation
- Selecting the best model

B. Data Sources and their formats

The statistics were gathered from www.MakeMyTrip.com, a company that serves as a booking/purchasing platform for airline tickets. Selenium is the framework used to scrape the data using a Web scraping approach. We extracted data from almost 2333 records, obtained records for various locations, gathered data on various flight parameters and stored the collected data in excel format. The desired variable "Price" is included in the dataset's 2333 rows and 9 columns of dimension. Both category and numerical data types are present in the specific datasets.

- Flight Name - Flight's name.
- Source: The place where the flight began.
- Destination - The location where the airplane will land.
- Duration - The amount of time it took an aircraft to go from one location to another.
- Stops - If any stops occurred in the route between the source and destination.
- Departure - The time at which the flight will take off from its origin or starting point.
- Arrival: The time at which passengers arrive at their destination.
- Price - The total cost incurred.

C. Data Pre-processing Done

Data pre-processing is the procedure of transforming unstructured data into a machine-readable format. Data pre-processing is a crucial phase in the machine learning process since the caliber of the data and the information that can be extracted from it directly influence how well our model can learn. For this reason, we must pre-process our data before introducing it to our model. The pre-processing techniques I employed were as follows:

1. Loading obtained datasets as a data frame and importing the required libraries examined certain statistical data, such as shape, the presence of unique values, info, unique (), data kinds, and value count function.

2. No missing values were discovered in the datasets after checking null values. By transforming "Departure time" and "Time of arrival" data types from object data types into Date Time data types, Timestamp variables are taken care of.

3. Some features underwent feature engineering because they had unnecessary values like "," and ":" which were changed to blank spaces.

4. Minutes and hours were represented in the Duration column's values.

5. Duration is the time it takes a plane to arrive at a location; it is the difference between the departure time and arrival time. So, using the arrival and departure time columns, I retrieved the correct duration time in terms of float data type.

6. Departure time and Time of arrival columns were extracted for the Departure Hour, Departure Menand, Arrival Hour, and Arrival Min columns, and these columns were discarded after extraction.

7. The target variable "Price" should have been continuous numeric data, but it was displayed as object data type because of certain text values, such as "," Therefore, I transformed this sign into a space and changed the data type to float.

D. Hardware and Software Requirements and Tools Used

Hardware required: -

1. Processor — core i5 or i7 and above

2. RAM — 8 GB or above

3. SSD — 250GB or above Software/s required: -

1. Anaconda

2. Python

Libraries required: - To run the program and to build the model we need some basic libraries as follows

```
# Importing some liabraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns


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

Below are required for building machine learning models

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
from sklearn import metrics
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error,mean_absolute_error

```

Below models are required to build regressor models.

```
from sklearn.neighbors import KNeighborsRegressor as KNN
from sklearn.metrics import classification_report
from sklearn.model_selection import cross_val_score
from sklearn import metrics

from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, ExtraTreesRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import BaggingRegressor

from sklearn.model_selection import GridSearchCV
```

MODEL/S DEVELOPMENT AND EVALUATION

A. Identification of possible problem-solving approaches (methods)

I have utilized both statistical and analytical methods to tackle the problem, mostly by pre-processing the data. I have also used EDA methods and heat maps to examine the association between independent and dependent characteristics. utilizing the square root transformation, skewness was eliminated. used Label Encoder to encrypt the data. Additionally, I ensured that the input data was scaled and cleaned before feeding it into the machine-learning models. This was done before developing the model. Regarding the feature significance information, we looked for the ideal random state to be employed on our regression machine learning model. Multiple regression models and assessment measures were eventually established.

B. Testing of Identified Approaches (Algorithms)

I have chosen the following regression techniques since "Price" is my goal variable and it is continuous in nature, which leads me to believe that this is a regression-type problem. With 11 columns remaining after pre-processing and cleaning the data, including the target, I used these independent characteristics for model construction and prediction with the aid of the feature significance bar graph. The following are the algorithms that were utilized to train the data:

1. DecisionTreeRegressor

2. RandomForestRegressor

3. Gradient Boosting Regressor

4. Bagging Regressor

5. ExtraTreesRegressor

C. Run and evaluate selected models

We have used 5 models. Let's evaluate it below.

1. Decision Tree Regressor.

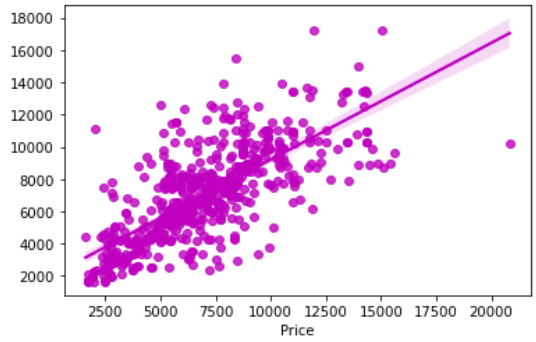
```python
# DecisionTreeRegressor
DTR = DecisionTreeRegressor()
DTR.fit(x_train,y_train)

# Checking R2 socre
predDTR = DTR.predict(x_test)
R2_score = r2_score(y_test,predDTR)*100
print('R2_Score:',R2_score)

# Evaluation Metircs
print('Mean Absolute Error:',metrics.mean_absolute_error(y_test, predDTR))
print('Mean Squared Error:',metrics.mean_squared_error(y_test, predDTR))
print("Root Mean Squared Error:",np.sqrt(metrics.mean_squared_error(y_test, predDTR)))

# plotting graph for above prediction
sns.regplot(y_test,predDTR,color="m")
plt.show()
```

```
R2_Score: 49.271028549792526
Mean Absolute Error: 1362.094410876133
Mean Squared Error: 4362012.408232628
Root Mean Squared Error: 2088.5431305655693
```



Created the Decision Tree Regression model and verified the metrics used for evaluation. The R2 score provided by the model is 49.27.

2. RandomForestRegressor

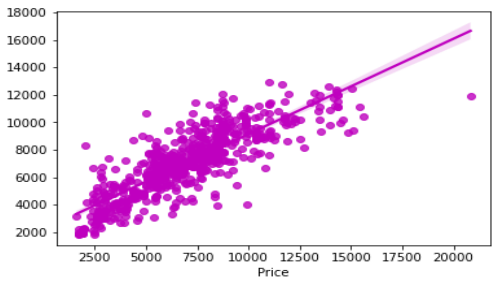
```python
# RandomForestRegressor
RFR = RandomForestRegressor()
RFR.fit(x_train,y_train)

# R2 score
predRFR = RFR.predict(x_test)
R2_score = r2_score(y_test,predRFR)*100
print('R2_Score:',R2_score)

# Evaluation Metrics
print('Mean Absolute Error:',metrics.mean_absolute_error(y_test, predRFR))
print('Mean Squared Error:',metrics.mean_squared_error(y_test, predRFR))
print("Root Mean Squared Error:",np.sqrt(metrics.mean_squared_error(y_test, predRFR)))

# Graph of prediction
sns.regplot(y_test,predRFR,color="m")
plt.show()
```

```
R2_Score: 70.3217848717258
Mean Absolute Error: 1143.3136298733993
Mean Squared Error: 2551929.1825342122
Root Mean Squared Error: 1597.475878545342
```



Created the RandomForestRegressor model and verified the metrics used for evaluation. The R2 score provided by the model is 70.32.

3. GradientBoostingRegressor

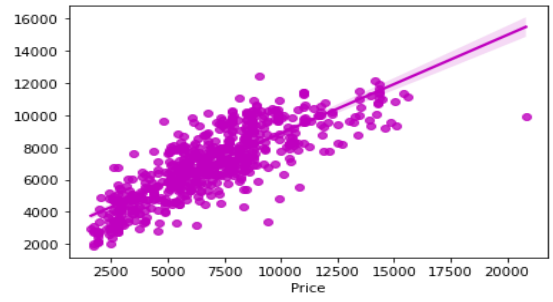
```
# GradientBoostingRegressor
GB = GradientBoostingRegressor()
GB.fit(x_train,y_train)

# R2 Score
predGB = GB.predict(x_test)
R2_score = r2_score(y_test,predGB)*100
print('R2_Score:',R2_score)

# Evaluation Metrics
print('Mean Absolute Error:',metrics.mean_absolute_error(y_test, predGB))
print('Mean Squared Error:',metrics.mean_squared_error(y_test, predGB))
print("Root Mean Squared Error:",np.sqrt(metrics.mean_squared_error(y_test, predGB)))

# Graph of prediction
sns.regplot(y_test,predGB,color="m")
plt.show()
```

```
R2_Score: 65.95529364737163
Mean Absolute Error: 1298.5605494792521
Mean Squared Error: 2927388.9712225534
Root Mean Squared Error: 1710.9614172220697
```



Created the GradientBoostingRegressor model and verified the metrics used for evaluation. The R2 score provided by the model is 65.95.

4. Bagging Regressor

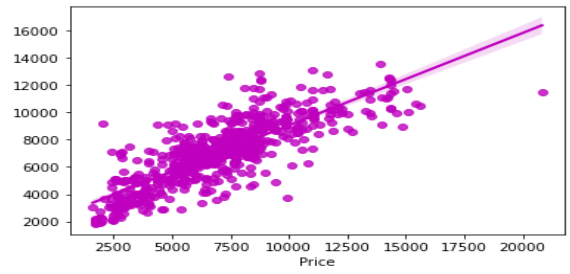
```
# BaggingRegressor
BR = BaggingRegressor()
BR.fit(x_train,y_train)

# R2score
predBR = BR.predict(x_test)
R2_score = r2_score(y_test,predBR)*100
print('R2_Score:',R2_score)

# Evalatuion Metrics
print('Mean Absolute Error:',metrics.mean_absolute_error(y_test, predBR))
print('Mean Squared Error:',metrics.mean_squared_error(y_test, predBR))
print("Root Mean Squared Error:",np.sqrt(metrics.mean_squared_error(y_test, predBR)))

# Graph of above prediction
sns.regplot(y_test,predBR,color="m")
plt.show()
```

```
R2_Score: 67.30083651159238
Mean Absolute Error: 1202.8756619551143
Mean Squared Error: 2811690.29841779
Root Mean Squared Error: 1676.8095593769108
```



5. ExtraTreesRegressor

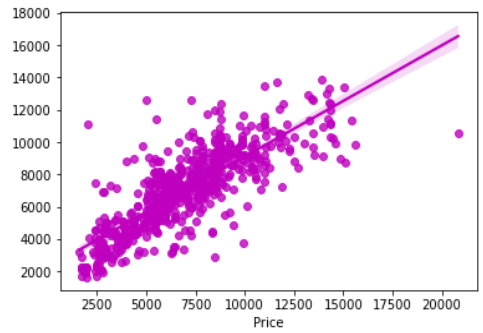
```
# ExtraTreesRegressor
XT = ExtraTreesRegressor()
XT.fit(x_train,y_train)

# R2Score
predXT = XT.predict(x_test)
R2_score = r2_score(y_test,predXT)*100
print('R2_Score:',R2_score)

# Evalation Metrics
print('Mean Absolute Error:',metrics.mean_absolute_error(y_test, predXT))
print('Mean Squared Error:',metrics.mean_squared_error(y_test, predXT))
print("Root Mean Squared Error:",np.sqrt(metrics.mean_squared_error(y_test, predXT)))

# Graph of prediction
sns.regplot(y_test,predXT,color="m")
plt.show()
```

```
R2_Score: 65.355945693953
Mean Absolute Error: 1165.5491314199396
Mean Squared Error: 2978924.871417711
Root Mean Squared Error: 1725.956219438289
```



Created the ExtraTreesRegressor model and verified the metrics used for evaluation. The R2 score provided by the model is 65.35.

Hyper Parameter Tuning

```
# Importing GridSearchCV for Hyper Parameter Tunning
from sklearn.model_selection import GridSearchCV
```

```
RF = RandomForestRegressor()

param ={'n_estimators':[100,200], 'criterion':['mse','mae'], 'min_samples_split':[2], 'min_samples_leaf':[1]}
```

```
RF_grid=GridSearchCV(RandomForestRegressor(),param,cv=4,scoring='accuracy',n_jobs=-1,verbose=2)
```

```
RF_grid.fit(x_train,y_train)
RF_grid_PRED=RF_grid.best_estimator_.predict(x_train)
```

```
Fitting 4 folds for each of 4 candidates, totalling 16 fits
```

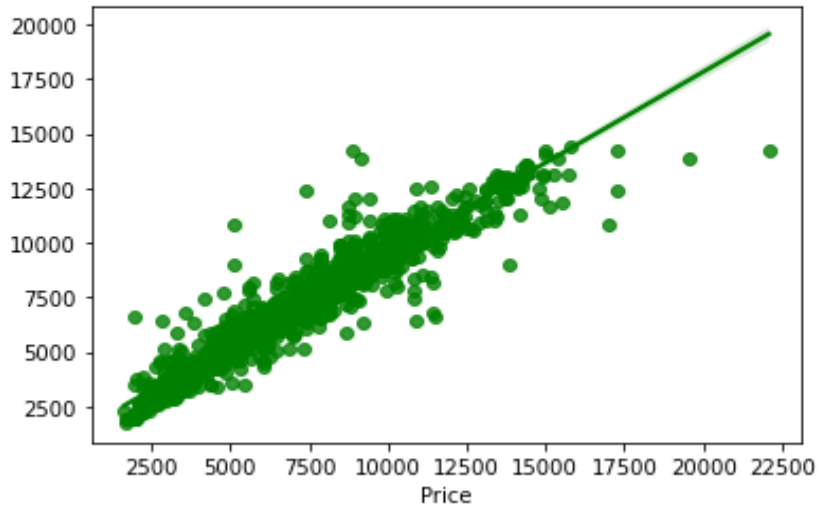
```
RF_grid.best_params_
```

```
{'criterion': 'mse',
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'n_estimators': 100}
```

```
print('MSE:',mean_squared_error(RF_grid_PRED,y_train))
print('MAE:',mean_absolute_error(RF_grid_PRED,y_train))
print ('============================================')
print('r2_score:',r2_score(RF_grid_PRED,y_train))
print ('============================================')
```

```
MSE: 813868.6847038086
MAE: 543.7438187127735
============================================
r2_score: 0.8654714368398391
============================================
```

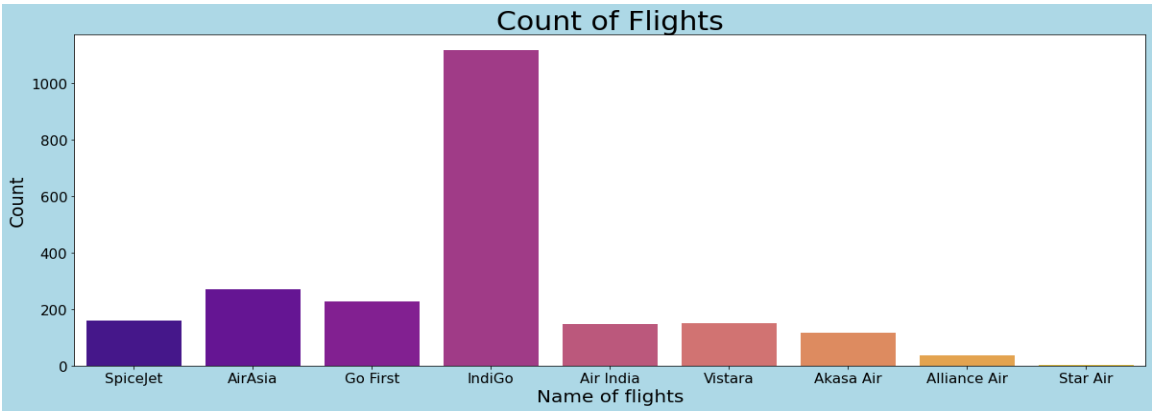
```
# visualizing the predicted values
sns.regplot(y_train,RF_grid_PRED,color="g")
plt.show()
```



After successfully incorporating hyperparameters tweaking using the optimal RandomForestRegressor values, the model's R2 score grew and now stands at 86.54% percent, which is a highly respectable result. We can see how our final model is translated from the graph. The best-fit line, which represents our real datasets, can be seen in the graph, and the dots represent the predictions that our best final model made.

D. Visualizations

a. Univariate Analysis

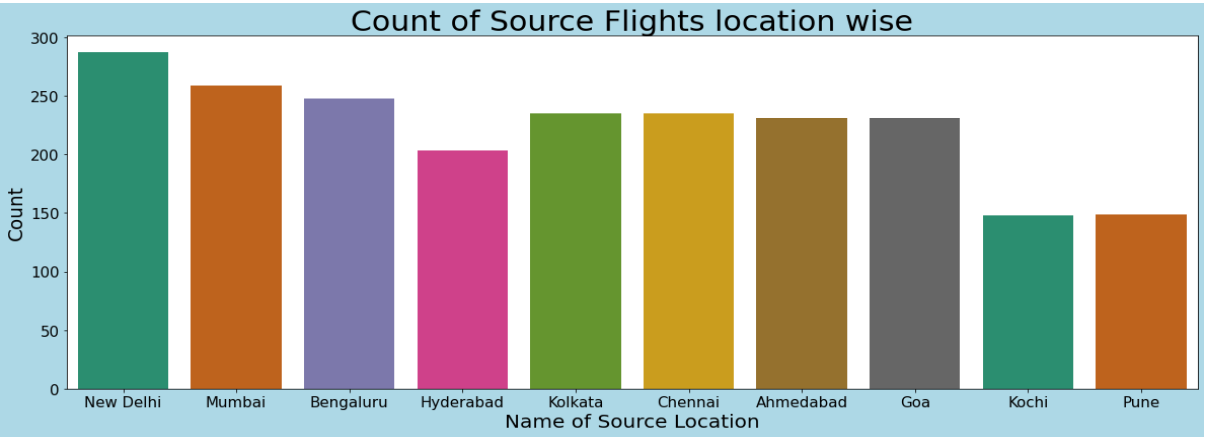


Sr No	Flight Names	Count
1	IndiGo	1117
2	AirAsia	272
3	Go First	227
4	SpiceJet	160
5	Vistara	149
6	Air India	148
7	Akasa Air	115
8	Alliance Air	37
9	Star Air	1
Grand Total		**2226**

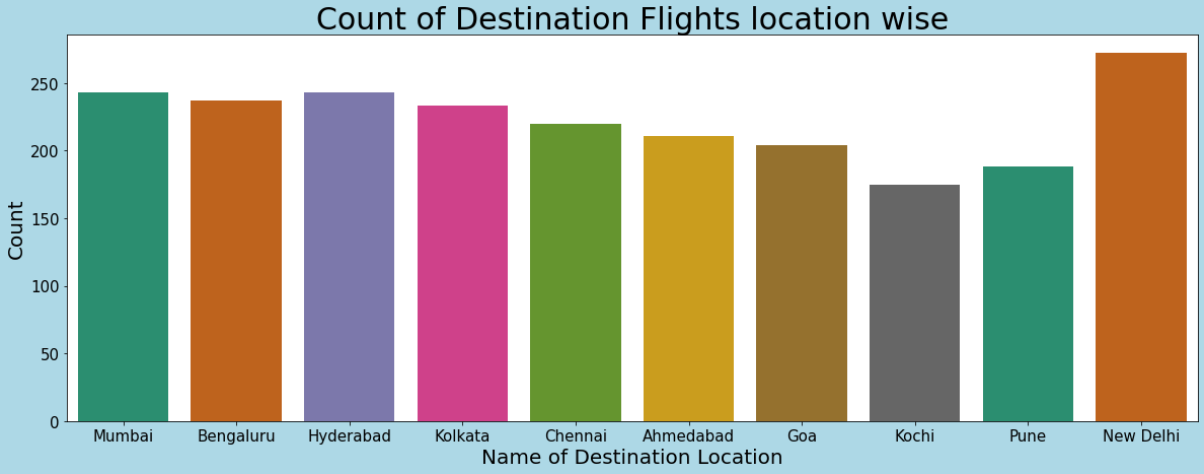
Number of Stops	Count of Stops
1 stop	882
2 stop	13
3 stop	2
Non stop	1329
Grand Total	2226

In our data, the most numbers of flights are IndiGo with 1117 flights and the least is Star Air with only one flight, AirAsia and Go First flights have a count of 272 and 227 respectively.

1329 flights are nonstop, and 882 flights are with 1 stop. Also, the count of flights with 2 or 3 stops is less, they are 13 and 2 respectively for 2 and 3 stops.



Source Location	Count of Source
New Delhi	287
Mumbai	259
Bengaluru	248
Chennai	235
Kolkata	235
Ahmedabad	231
Goa	231
Hyderabad	203
Pune	149
Kochi	148
Grand Total	2226

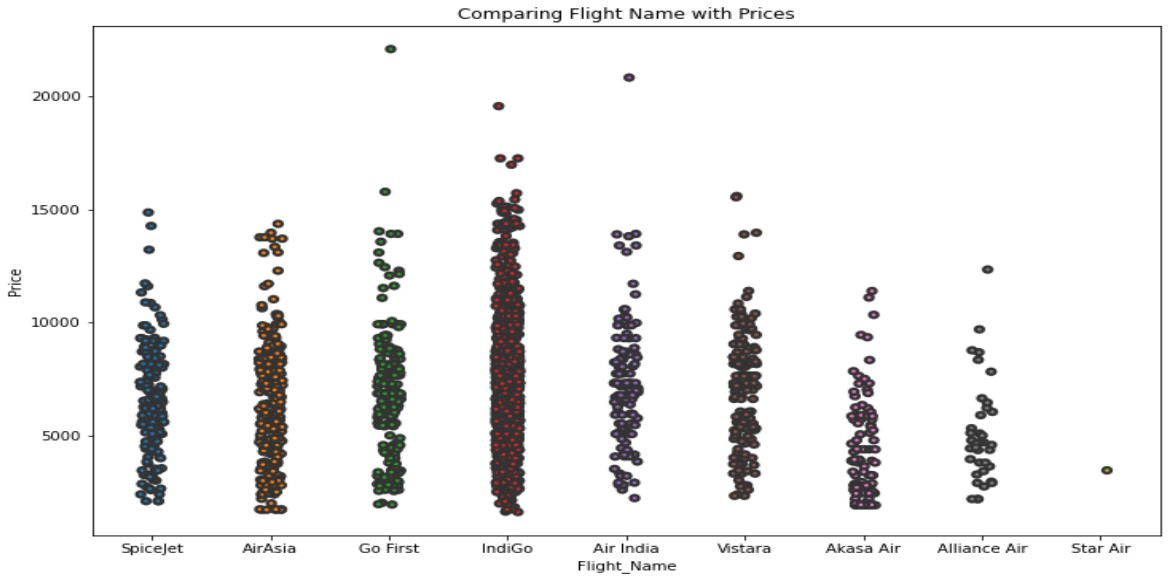


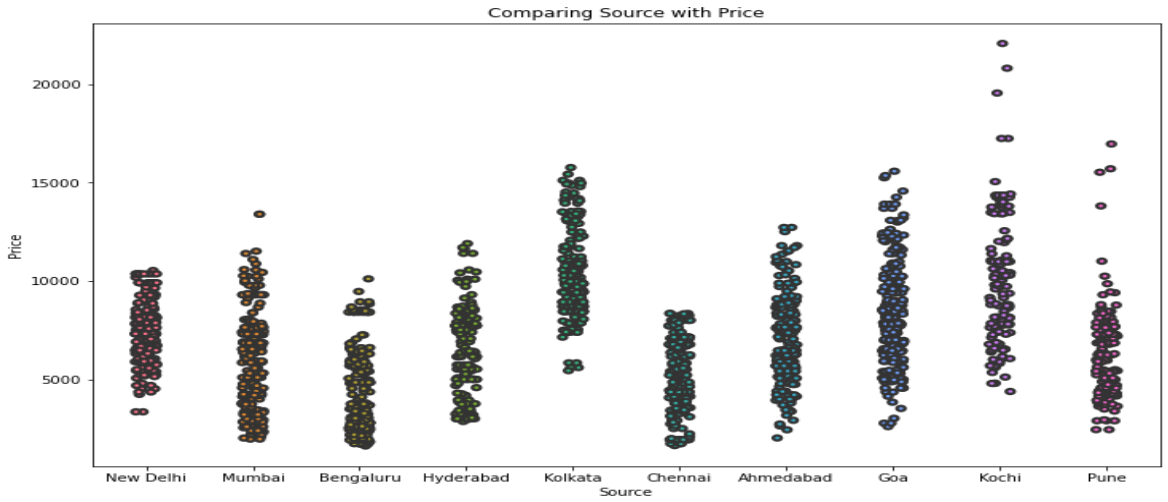
Destination Location	Count of Destination
New Delhi	272
Hyderabad	243
Mumbai	243
Bengaluru	237
Kolkata	233
Chennai	220
Ahmedabad	211
Goa	204
Pune	188
Kochi	175
Grand Total	2226

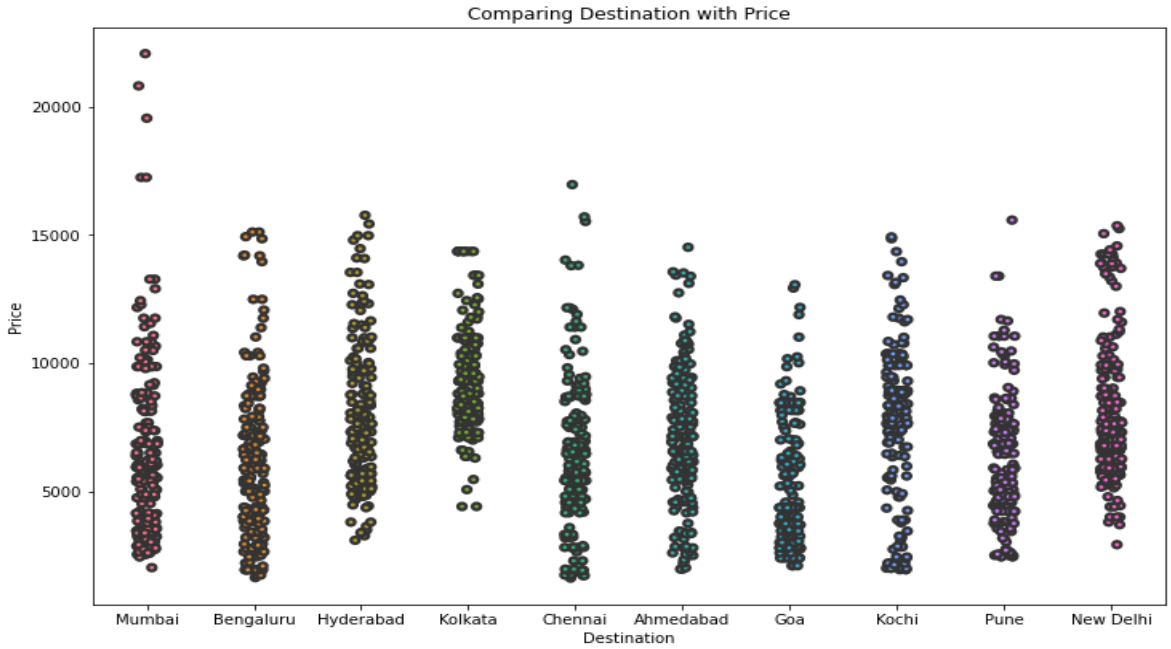
Most numbers of flights started from New Delhi in our data followed by Mumbai and Bengaluru, Pune and Kochi have the least number of flights at the source location in our dataset.

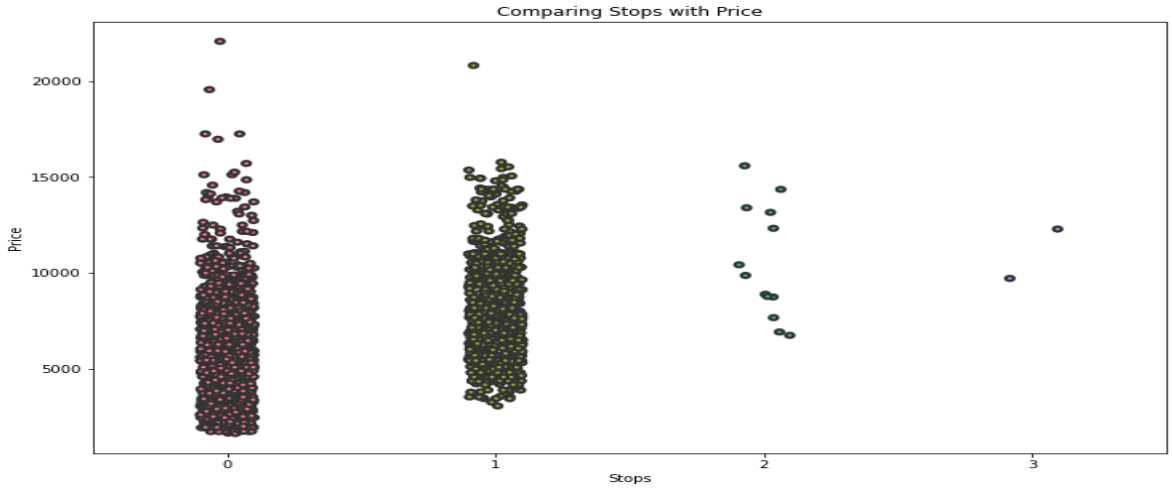
New Delhi has the highest count of Destination points followed by Mumbai and Hyderabad with the same count, and Pune and Kochi have the least count with 188 and 175 Destination counts respectively.

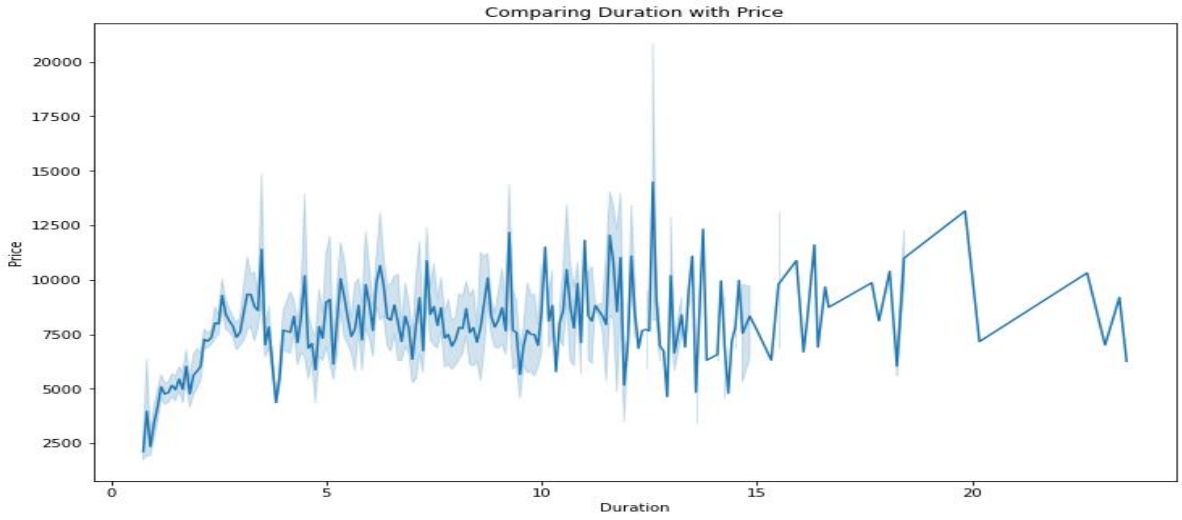
b. Bivariate Analysis


Comparing Flight Name with Prices
Price
20000
15000
10000
5000
SpiceJet
AirAsia
Go First
IndiGo
Air India
Vistara
Akasa Air
Alliance Air
Star Air
Flight_Name


Comparing Source with Price
Price
20000
15000
10000
5000
New Delhi
Mumbai
Bengaluru
Hyderabad
Kolkata
Chennai
Ahmedabad
Goa
Kochi
Pune
Source


Comparing Destination with Price
Price
20000
15000
10000
5000
Mumbai
Bengaluru
Hyderabad
Kolkata
Chennai
Ahmedabad
Goa
Kochi
Pune
New Delhi
Destination





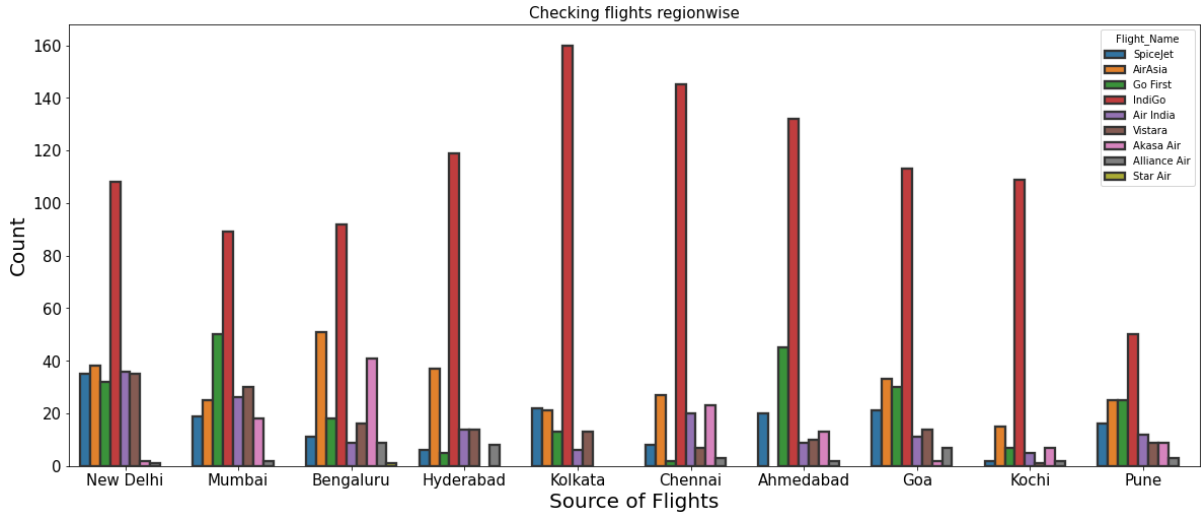
Generally, the basic price of all flights is almost the same, there might be changes according to other related features such as duration, source, duration, and several stops.

Kolkata's basic price starts from above 5000 which is more than compared with other flights.

Also, where Kolkata is the destination basic price is showing above 5000. The price of Flights keeps increasing with every stop, the price with no stops has a less basic price as compared to flights with 1, 2, and 3 stops.

With the above graph, we can see that as the duration of a flight increases price range also gets increasing.

c. Multivariate Analysis



Flight Name	Source Location										Grand Total
	Ahmedabad	Bengaluru	Chennai	Goa	Hyderabad	Kochi	Kolkata	Mumbai	New Delhi	Pune	
Air India	9	9	20	11	14	5	6	26	36	12	148
AirAsia	0	51	27	33	37	15	21	25	38	25	272
Akasa Air	13	41	23	2	0	7	0	18	2	9	115
Alliance Air	2	9	3	7	8	2	0	2	1	3	37
Go First	45	18	2	30	5	7	13	50	32	25	227
IndiGo	132	92	145	113	119	109	160	89	108	50	1117
SpiceJet	20	11	8	21	6	2	22	19	35	16	160
Star Air	0	1	0	0	0	0	0	0	0	0	1
Vistara	10	16	7	14	14	1	13	30	35	9	149
Grand Total	231	248	235	231	203	148	235	259	287	149	2226

1. IndiGo flight is the most with 1117 flights in each Source Location, it has the greatest number of running flights in these locations.
2. AirAsia is the second number with 272 flights from the source location.
3. Start Air has the least number of flights in our data with only 1 flight.

E. Interpretation of the Results

I utilized distribution plots to depict the numerical variables in the variate analysis and count plots and pie plots to illustrate the counts in categorical variables. To examine the relationship between the label and the characteristics in the bi-variate analysis, I utilized bar graphs, strip plots, line plots, reg plots, box plots, and boxen plots. To examine the pair-wise relationship between the characteristics, use a pair plot.

To create the ML models and produce accurate predictions, the dataset has to be cleaned and scaled. All of the crucial characteristics are available in the dataset and prepared for model creation, therefore I've previously highlighted a few processing procedures that I've carried out.

Data cleaning and processing were followed by a train-test-split procedure to create the model. To obtain an accurate R2 score and assessment measures like MAE, MSE, and RMSE, I developed numerous regression models. Random Forest Regressor, which has a 70% R2 score, is the best model I could find. The Random Forest Regressor's R2 score climbed to 86% when the best model was tuned, and it also received favorable assessment metrics. I finally saved my final model and obtained accurate estimates for the cost of airline tickets.

CONCLUSION

A. Key Findings and Conclusions of the Study

The purpose of the case study is to provide an example of using machine learning algorithms to forecast the cost of airline tickets. when this project is finished. We gained an understanding of the data collection, pre-processing, analysis, cleaning, and model-building processes. We used web scraping to first gather the flight information from the www.makemytrip.com website. Selenium was the technology utilized for web scraping, which offers the benefit of automating our data collection process. We gathered roughly 2333 pieces of information, including the cost of airline tickets and other relevant details. The scraped data was then stored in an excel file so we could utilize it later and analyze it.

B. Learning Outcomes of the Study in respect of Data Science

I learned a lot about the characteristics of flights and the websites that sell airline tickets while working on this project, and I also developed an understanding of how machine learning models have assisted in predicting the cost of airline tickets. The project caught my attention since the dataset covers a variety of data kinds. To illustrate the relationship between the objective and the characteristics, I employed a variety of charting techniques. This graphical depiction made it easier for me to comprehend which characteristics are crucial and how they affect ticket prices. In this project, where I worked with features that included string values, feature extraction, and feature selection, data cleansing was one of the critical and vital aspects. Random Forest Regressor was finally chosen as the best model.

C. Limitations of this work and Scope for Future Work

The little number of records employed in this study is its principal drawback. Although I had taken care, several of the columns in the dataset do not have our data spread appropriately, and many of the values in the columns had string values. Our models sometimes fail to generate the appropriate patterns, which lowers the model's performance. Problems, therefore, need to be resolved.

The lack of data in this paper is its biggest flaw. Anyone who wants to build on it should look for more sources of historical data manually over time. Additionally, a wider range of flights should be investigated because it is conceivable that airlines change their price policy depending on the details of the travel (for example, fares for regional flights out of small airports may behave differently than the major, well-flown routes we considered here). Finally, it would be fascinating to compare the precision of our system to that of the current commercial systems (preferably over some time).

Thank You