Flight Price Prediction Project

BY:

SOURABH SONI

FLIP ROBO

ACKNOWLEDGMENT

I express my sincere gratitude to Flip Robo Technologies for giving me the opportunity to work on this project on Flight Price Prediction using machine learning algorithms. I acknowledge my indebtedness to the authors of the paper titled: “Airline ticket price and demand prediction: A survey” and the online article titled: “Trying to Predict Airfares When The Unpredictable Happens” for providing me with invaluable insights and knowledge of the various factors that determine the price of Flight tickets.

INTRODUCTION

Business Problem Framing

Anyone who has booked a flight ticket knows how unexpectedly the prices vary. The cheapest available ticket on a given flight gets more and less expensive over time. This usually happens as an attempt to maximize revenue based on –

- Time of purchase patterns (making sure last-minute purchases are expensive)
- 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)

Therefore, a predictive model to accurately predict Air fares is required to be made.

INTRODUCTION

Conceptual Background of the Domain Problem

Predictive modelling, Regression algorithms are some of the machine learning techniques used for predicting Flight Ticket prices. Identifying various relevant attributes like Airline Brand, flight duration, source and destination etc are crucial for working on the project as they determine the valuation of air fare.

INTRODUCTION

Review of Literature

A Research paper titled: “Airline ticket price and demand prediction: A survey” by Juhar Ahmed Abdella and online article titled: “Trying to Predict Airfares When The Unpredictable Happens” were reviewed and studied to gain insights into all the attributes that contribute to the pricing of flight tickets.

It is learnt that deterministic features like Airline Brand, flight number, departure dates, number of intermediate stops, week day of departure, number of competitors on route and aggregate features – which are based on collected historical data on minimum price, mean price, number of quotes on non-stop,1-stop and multi-stoppage flights are some the most important factors that determine the pricing of Flight Tickets.

- Airline ticket price and demand prediction: A survey - ScienceDirect
- Flight Price Predictor | American Express GBT (amexglobalbusinesstravel.com)

INTRODUCTION

Motivation for the Problem Undertaken

With airfares fluctuating frequently, knowing when to buy and when to wait for a better deal to come along is tricky. The fluctuation in prices is frequent and one has limited time to book the cheapest ticket as the prices keep varying due to constant manipulation by Airline companies. Therefore, it is necessary to work on a predictive model based on deterministic and aggregate feature data that would predict with good accuracy the most optimal Air fare for a particular destination, route and schedule.

Analytical Problem Framing

Mathematical/ Analytical Modeling of the Problem

Various Regression analysis techniques were used to build predictive models to understand the relationships that exist between Flight ticket price and Deterministic and Aggregate features of Air travel. The Regression analysis models were used to predict the Flight ticket price value for changes in Air travel deterministic and aggregate attributes. Regression modelling techniques were used in this Problem since Air Ticket Price data distribution is continuous in nature.

In order to forecast Flight Ticket price, predictive models such as ridge regression Model, Random Forest Regression model, Decision tree Regression Model, Support Vector Machine Regression model and Extreme Gradient Boost Regression model were used to describe how the values of Flight Ticket Price depended on the independent variables of various Air Fare attributes.

Analytical Problem Framing

Data Sources and their formats

The Dataset was compiled by scraping Data for various Air Fare attributes and Price from https://www.yatra.com/ and https://www.easemytrip.com/

The data was converted into a Pandas Dataframe under various Feature and Label columns and saved as a .csv file.

```
fDF.head(10)
```

	Unnamed: 0	Airline	Flight Number	Date of Departure	From	To	Duration	Total Stops	Price
0	0	Air Asia	I5-764	Tue, Feb 8	New Delhi	Mumbai	2h 10m	Non Stop	2,456
1	1	IndiGo	6E-2054	Tue, Feb 8	New Delhi	Mumbai	2h 10m	Non Stop	2,456
2	2	IndiGo	6E-5001	Tue, Feb 8	New Delhi	Mumbai	2h 10m	Non Stop	2,456
3	3	IndiGo	6E-2046	Tue, Feb 8	New Delhi	Mumbai	2h 10m	Non Stop	2,456
4	4	IndiGo	6E-5328	Tue, Feb 8	New Delhi	Mumbai	2h 15m	Non Stop	2,456
5	5	Air Asia	I5-482	Tue, Feb 8	New Delhi	Mumbai	2h 15m	Non Stop	2,456
6	6	IndiGo	6E-6278	Tue, Feb 8	New Delhi	Mumbai	2h 20m	Non Stop	2,456
7	7	IndiGo	6E-218	Tue, Feb 8	New Delhi	Mumbai	2h 20m	Non Stop	2,456
8	8	IndiGo	6E-171	Tue, Feb 8	New Delhi	Mumbai	2h 25m	Non Stop	2,456
9	9	IndiGo	6E-2081	Tue, Feb 8	New Delhi	Mumbai	2h 25m	Non Stop	2,456

Analytical Problem Framing

Dataset Description

The Independent Feature columns are:

- Airline: The name of the airline.
- Flight Number: Number of Flight
- Date of Departure: The date of the journey
- From: The source from which the service begins
- To: The destination where the service ends
- Duration: Total duration of the flight
- Total Stops: Total stops between the source and destination.

Analytical Problem Framing

Target / Label Column:

- Price: The Price of the Ticket

Analytical Problem Framing

Data Preprocessing Done

- Duplicate data elements in various columns: ‘Airline’,’From’,’To’, which had their starting letters in upper case and lower case were converted to data elements starting with uppercase letters.
- Data in column ‘Price’ was converted to int64 data type.
- Columns: Unnamed: 0(just a series of numbers) was dropped since it doesn't contribute to building a good model for predicting the target variable values.
- The Date format of certain data elements in ‘Date of Departure’ was changed to match the general Date format of majority of the data elements of the column.

Analytical Problem Framing

Feature Engineering:

- In order to better understand the relationships between Flight price and Air Fare attributes, ‘Day’,’Date’ and ‘Month’ columns were created based on data of existing column: ‘Date of Departure’.
- The values in Column: ‘Duration’ were converted from Hours-Minutes format to minute format and the data type was converted to int64.

Analytical Problem Framing

Data Inputs- Logic- Output Relationships

The Datasets consist mainly of Int and Object data type variables. The relationships between the independent variables and dependent variable were analysed.

Hardware and Software Requirements and Tools Used

Hardware Used:

- Processor: intel Core i3- 2348M
- Physical Memory: 500GB HDD
- GPU: Nvidia GeForce710M, 2GB VRAM,

Hardware and Software Requirements and Tools Used

Software Used:

- Windows 10 Operating System
- Anaconda Package and Environment Manager: Anaconda is a distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows and provides a host of tools and environment for conducting Data Analytical and Scientific works. Anaconda provides all the necessary Python packages and libraries for Machine learning projects

Hardware and Software Requirements and Tools Used

- Jupyter Notebook: The Jupyter Notebook is an open-source web application that allows data scientists to create and share documents that integrate live code, equations, computational output, visualizations, and other multimedia resources, along with explanatory text in a single document.
- Python3: It is open source, interpreted, high level language and provides great approach for object-oriented programming. It is one of the best languages used for Data Analytics And Data science projects/application. Python provides numerous libraries to deal with mathematics, statistics and scientific function.

Hardware and Software Requirements and Tools Used

- Python Libraries used:
  - Pandas: For carrying out Data Analysis, Data Manipulation, Data Cleaning etc
  - Numpy: For performing a variety of operations on the datasets.
  - matplotlib.pyplot, Seaborn: For visualizing Data and various relationships between Feature and Label Columns
  - Scipy: For performing operations on the datasets
  - Statsmodels: For performing statistical analysis
- sklearn for Modelling Machine learning algorithms, Data Encoding, Evaluation metrics, Data Transformation,Data Scaling, Component analysis,Feature selection etc.

Exploratory Data Analysis

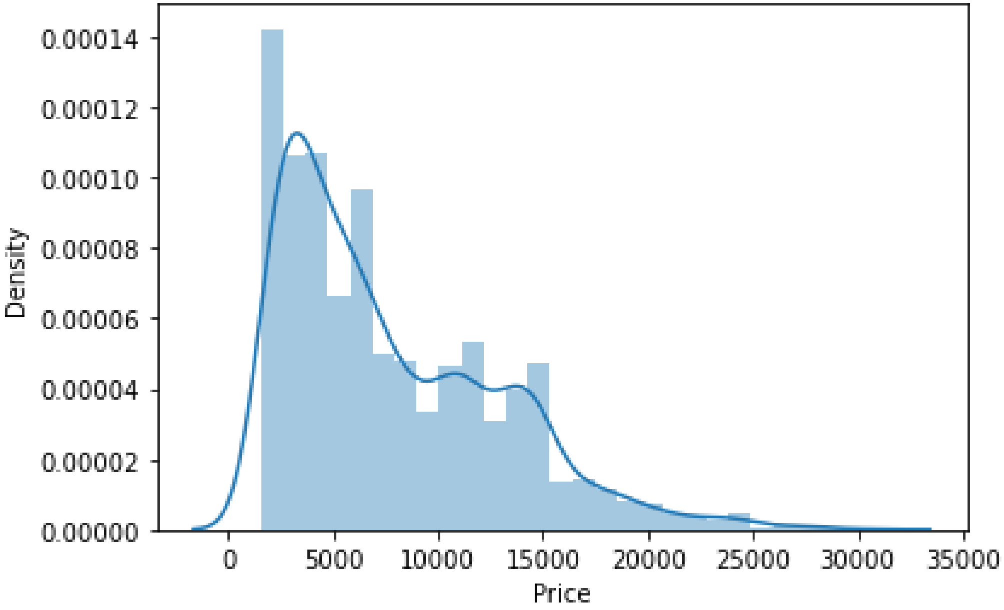
Visualizations

Barplots,Distplots,Boxplots,Countplots,lineplots were used to visualise the data of all the columns and their relationships with Target variable.

Exploratory Data Analysis

Univariate Analysis

Analyzing the Target Variable

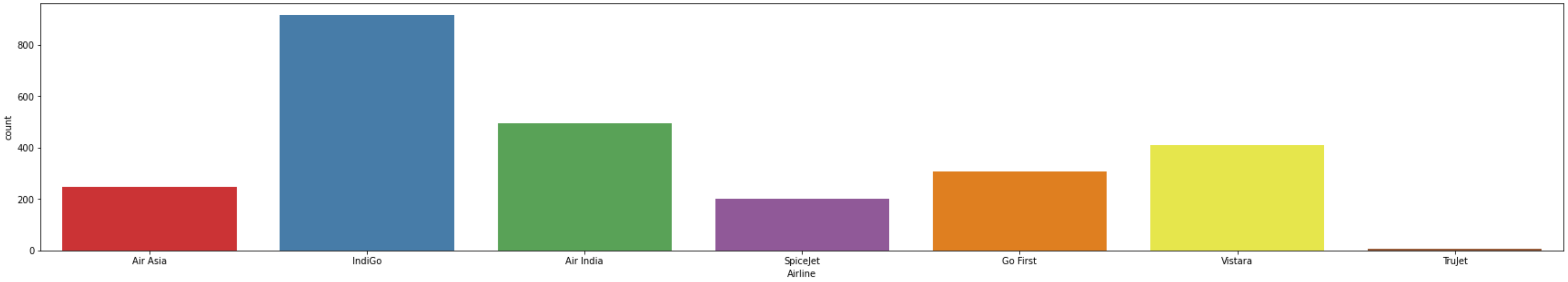


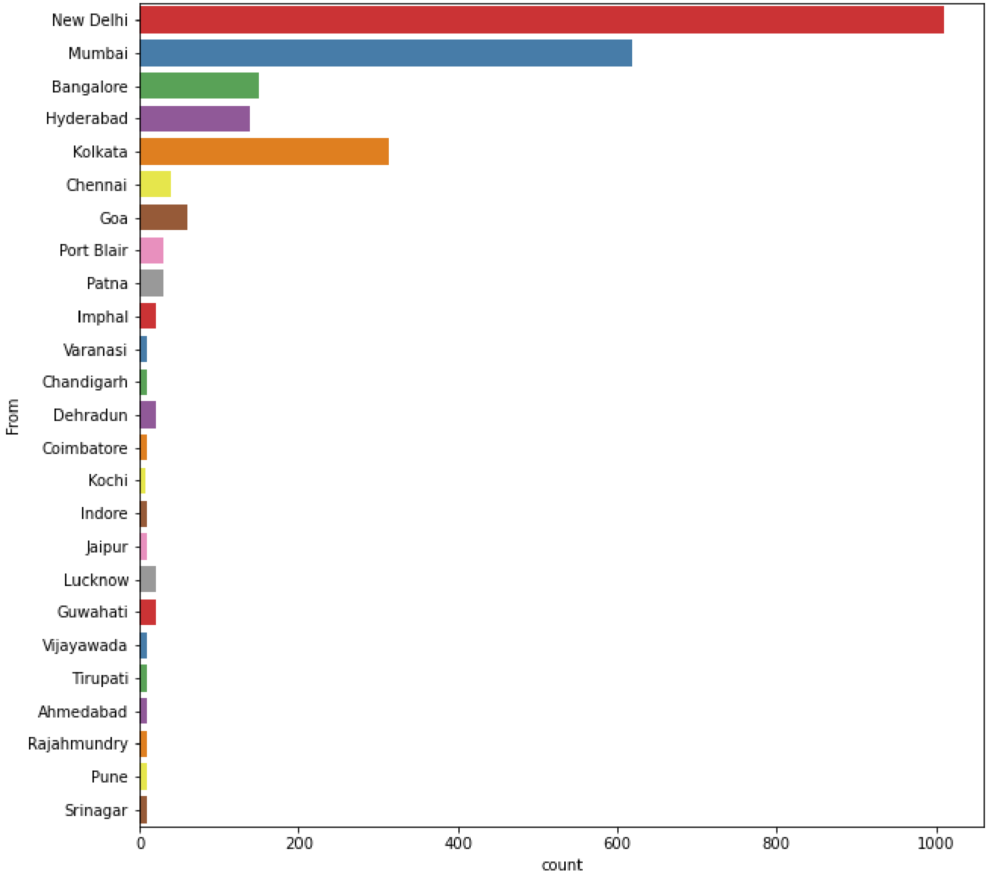
Exploratory Data Analysis

From the graph above it is observed that the Price data forms a continuous distribution with mean of 7748.33 and tails of from 15000 mark and the distribution is skewed.

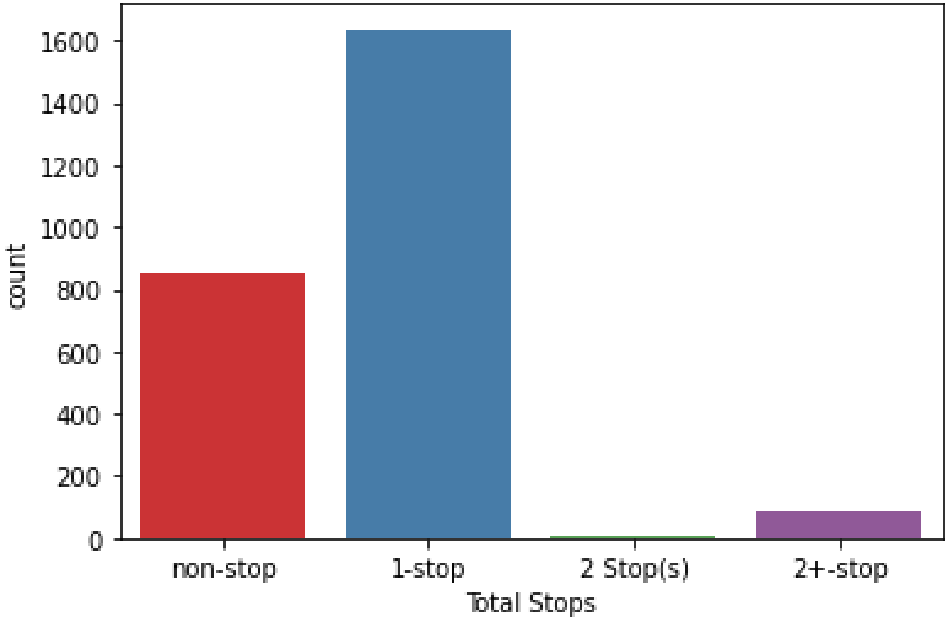
Exploratory Data Analysis

Analyzing the Feature Columns

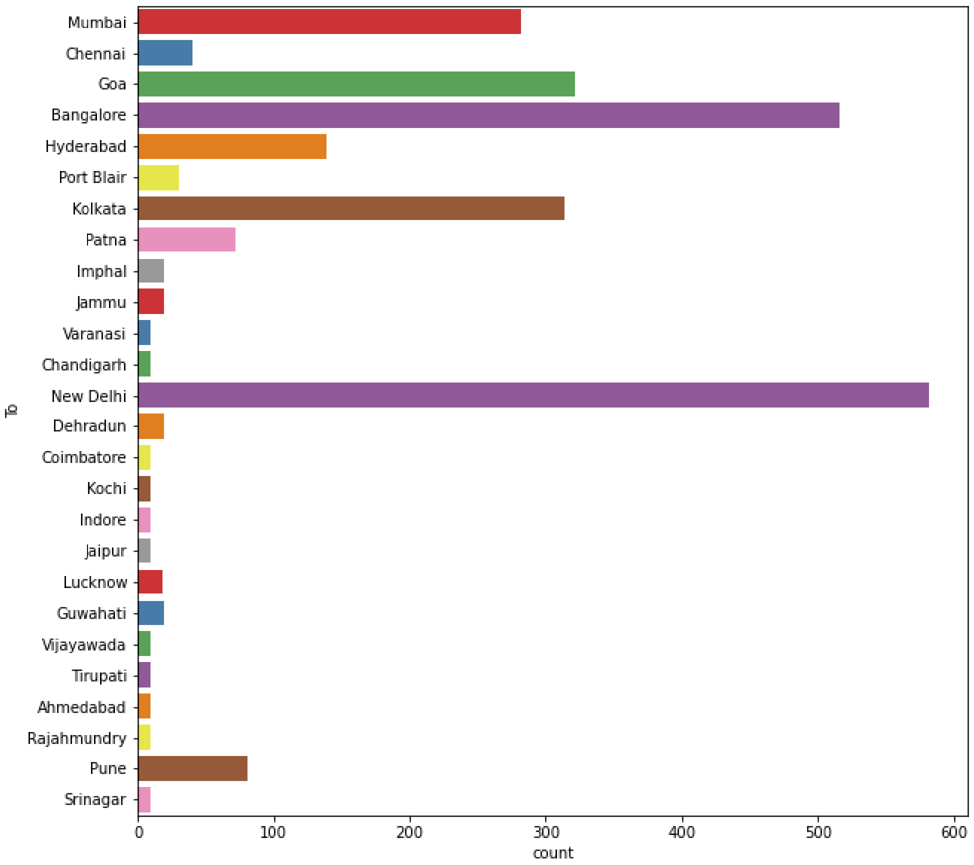




New Delhi
Mumbai
Bangalore
Hyderabad
Kolkata
Chennai
Goa
Port Blair
Patna
Imphal
Varanasi
Chandigarh
Dehradun
Coimbatore
Kochi
Indore
Jaipur
Lucknow
Guwahati
Vijayawada
Tirupati
Ahmedabad
Rajahmundry
Pune
Srinagar
From
0
200
400
600
800
1000
count



1600
1400
1200
1000
800
600
400
200
0
count
non-stop
1-stop
2 Stop(s)
2+-stop
Total Stops


Mumbai
Chennai
Goa
Bangalore
Hyderabad
Port Blair
Kolkata
Patna
Imphal
Jammu
Varanasi
Chandigarh
New Delhi
Dehradun
Coimbatore
Kochi
Indore
Jaipur
Lucknow
Guwahati
Vijayawada
Tirupati
Ahmedabad
Rajahmundry
Pune
Srinagar
To
0
100
200
300
400
500
600
count

Exploratory Data Analysis

Following observations are made from graphs above:

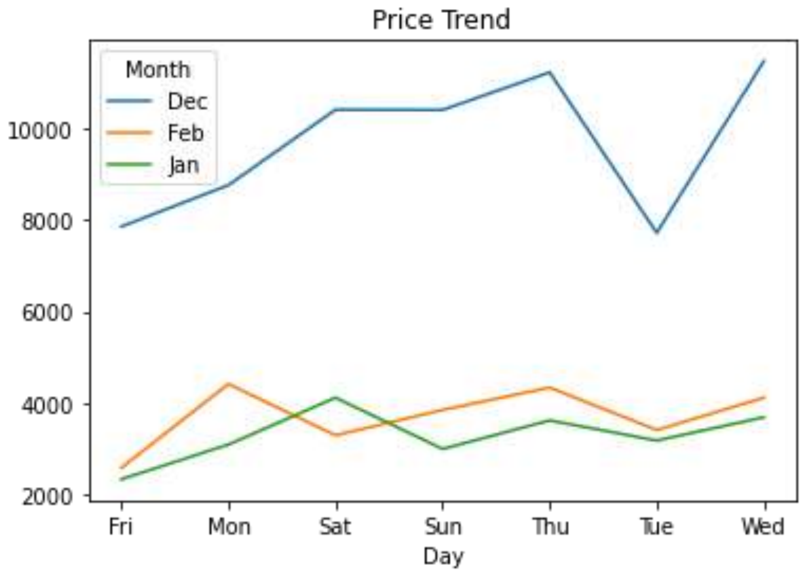
- IndiGo has the highest number of flights followed by Air India and Vistara
- Highest number of flights are from Delhi followed by Mumbai, Kolkata,Bangalore and Hyderabad
- New Delhi is the most popular destination followed by Bangalore, Goa, Kolkata and Mumbai
- Highest number of flights have only 1 stop between source and destination while 2nd highest number of flights are non stop

Exploratory Data Analysis

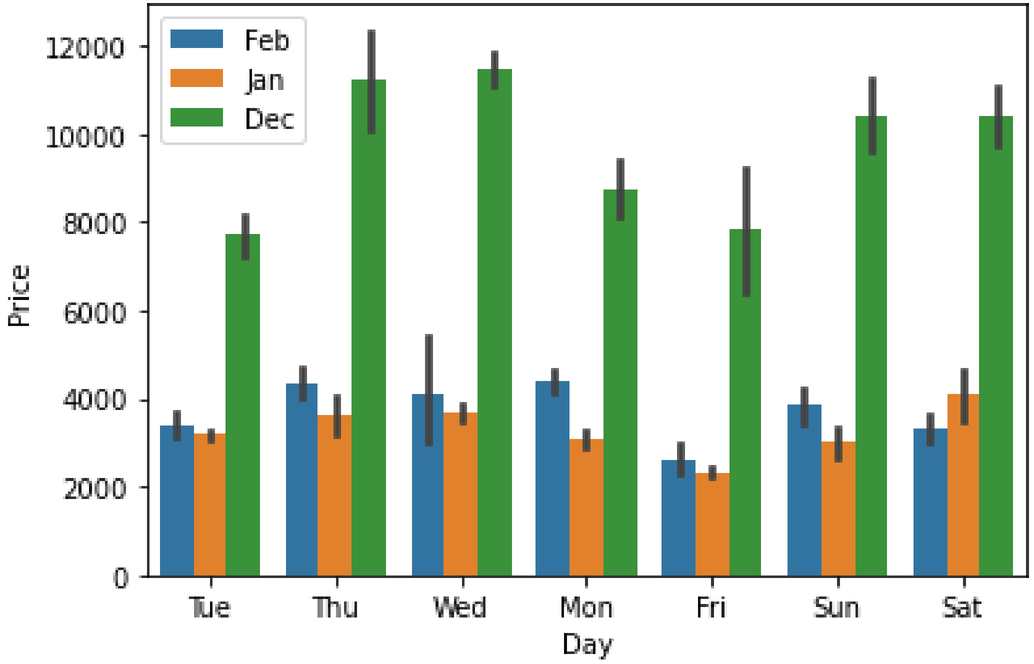
Bivariate Analysis

Interpreting Relationship between Dependent Variable and Independent Variable Columns

Analyzing Relationship between Day, Month columns and Price



Exploratory Data Analysis



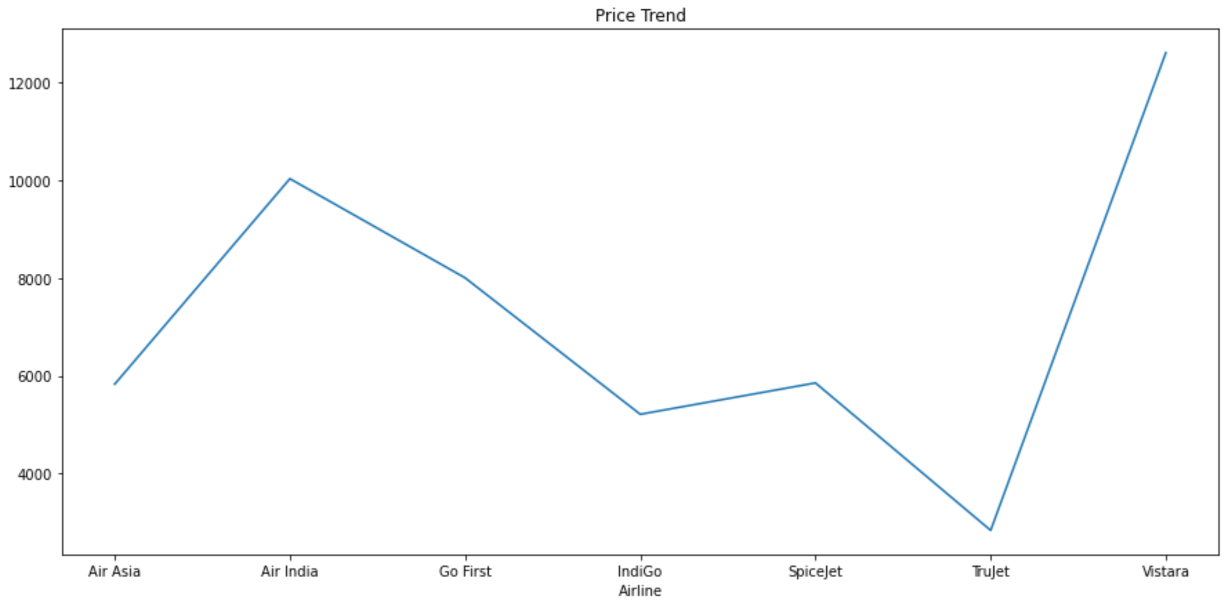
Exploratory Data Analysis

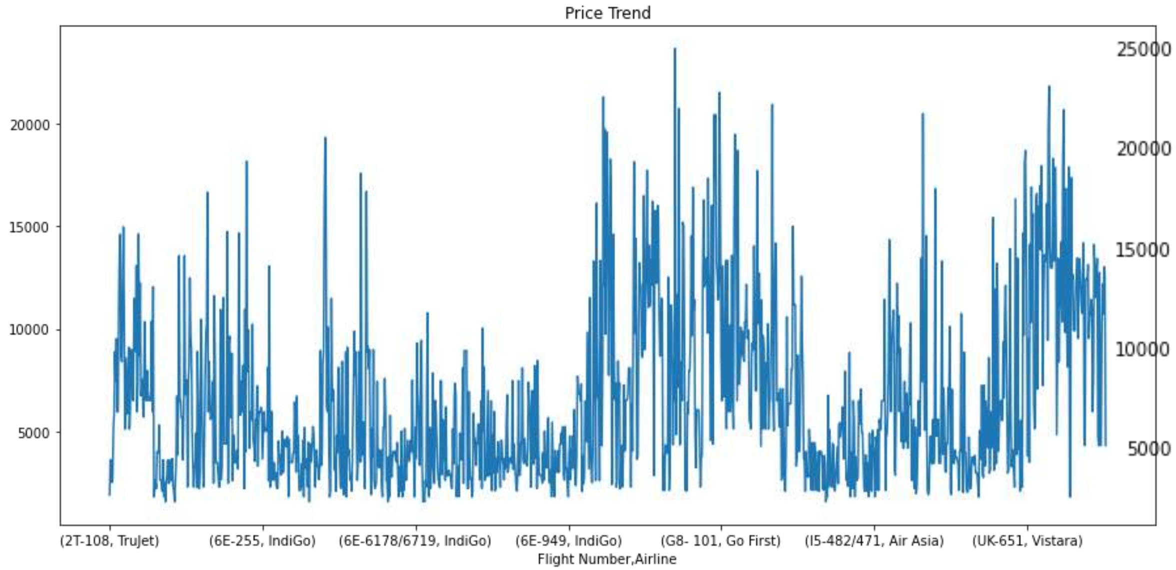
Following observations are made from graphs above:

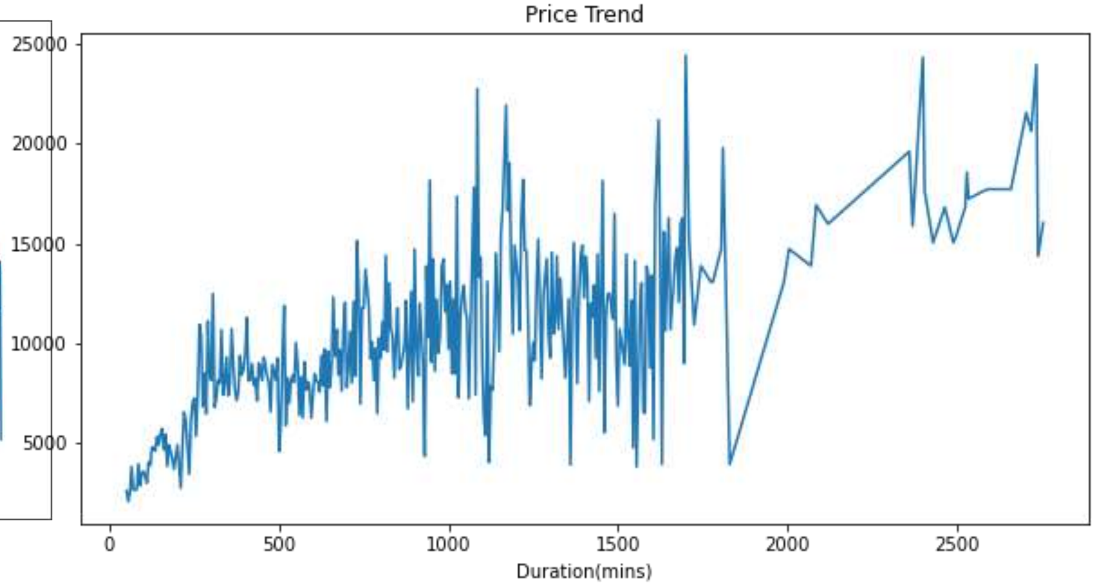
- On an average, there is a steady decline in Flight price from December to February, with the prices being lowest in January.
- Flight Prices increase on an average, as the day of departure gets nearer.
- Flight Ticket prices are the highest on Thursdays,Mondays and during the weekend on an average.

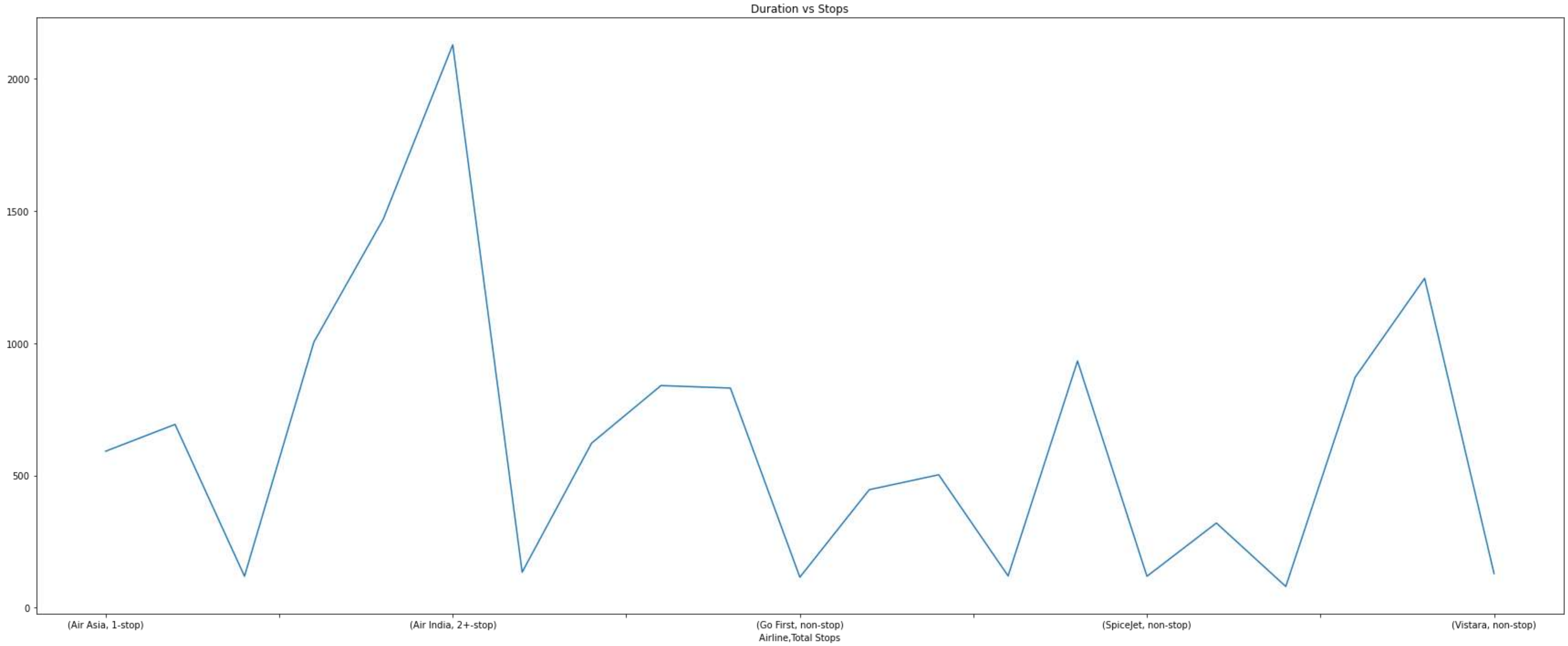
Exploratory Data Analysis

Analyzing Relationship between Airlines, Flight Duration and Price

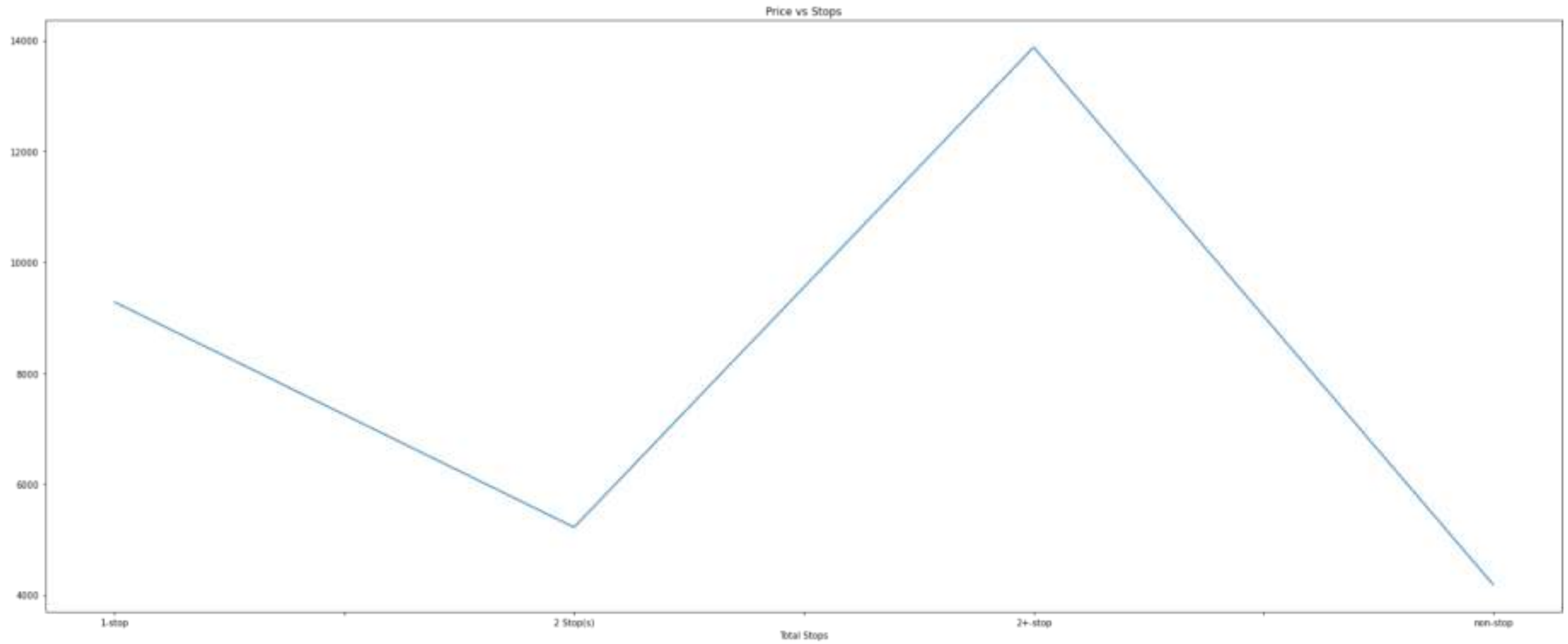



Price Trend
20000
15000
10000
5000
(2T-108, TruJet)
(6E-255, IndiGo)
(6E-6178/6719, IndiGo)
(6E-949, IndiGo)
(G8- 101, Go First)
(I5-482/471, Air Asia)
(UK-651, Vistara)
Flight Number,Airline

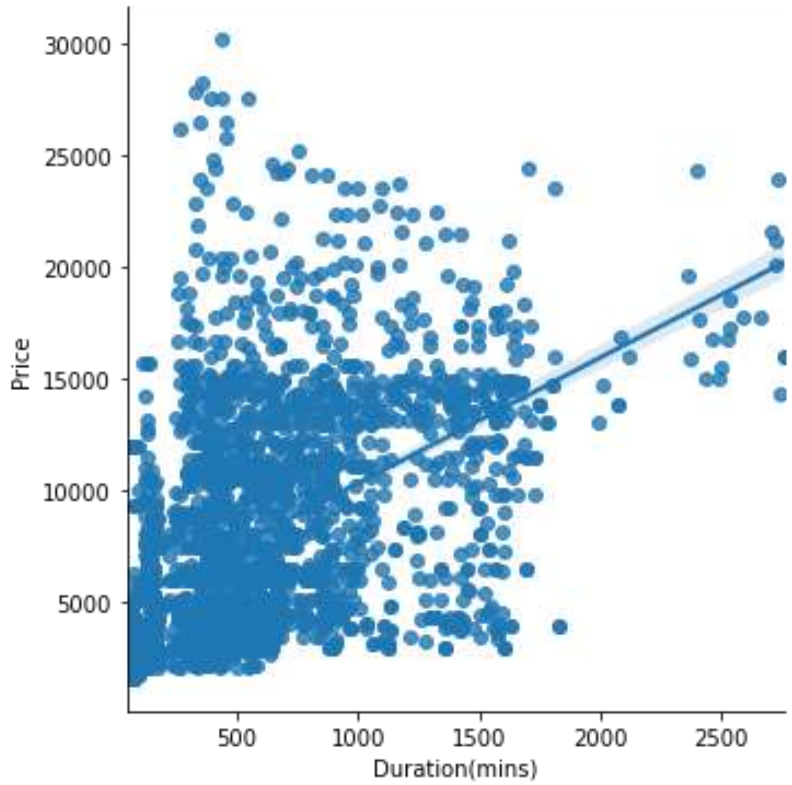

Price Trend
25000
20000
15000
10000
5000
0
500
1000
1500
2000
2500
Duration(mins)



Duration vs Stops
2000
1500
1000
500
0
(Air Asia, 1-stop)
(Air India, 2+-stop)
(Go First, non-stop)
(SpiceJet, non-stop)
(Vistara, non-stop)
Airline,Total Stops



Price vs Stops
14000
12000
10000
8000
6000
4000
1-stop
2 Stop(s)
2+-stop
non-stop
Total Stops


30000
25000
20000
15000
10000
5000
Price
500
1000
1500
2000
2500
Duration(mins)

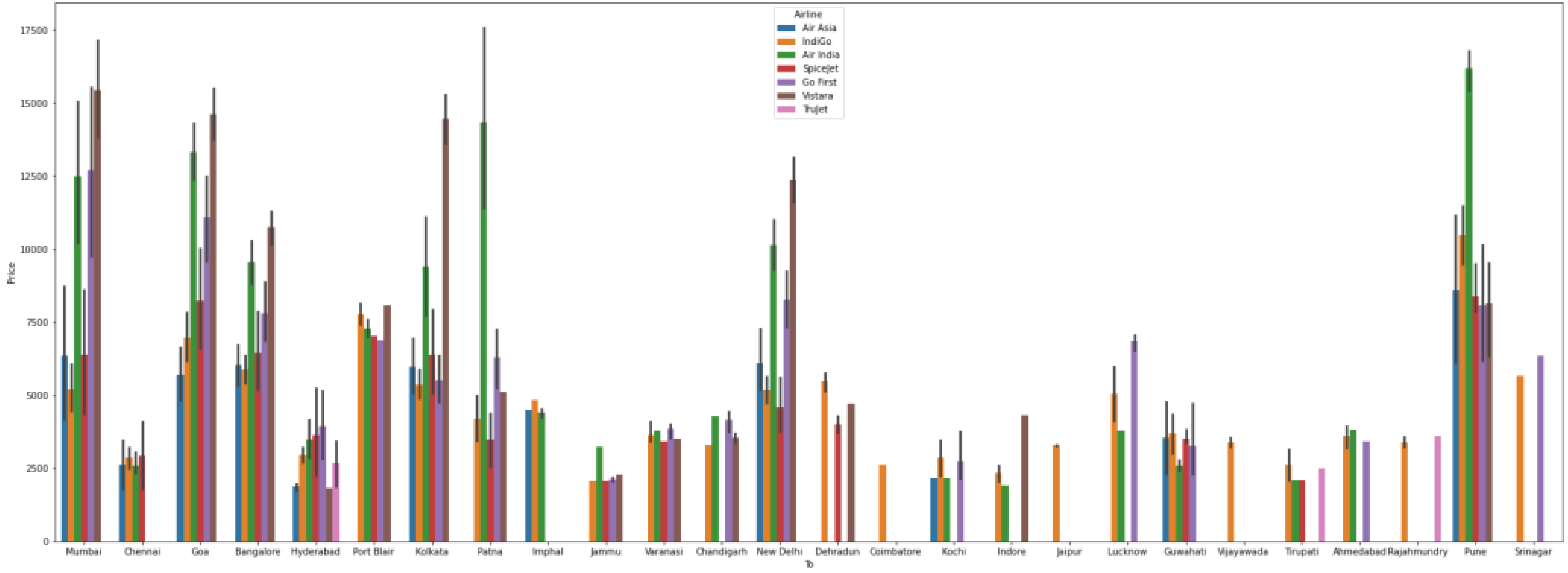
Exploratory Data Analysis

Following Observation is made from graphs above:

- Trujet, IndiGo,SpiceJet and Air Asia offer air tickets at the most affordable prices on average, whereas Vistara, Air India are the most expensive on average.
- It can be observed that Number of Stops impact the travel time of Airlines.
- It can be observed that Number of Stops impact the Air Ticket Pricing of Airlines.
- There is a linear relationship between Price and flight duration.

Exploratory Data Analysis

Multivariate Analysis



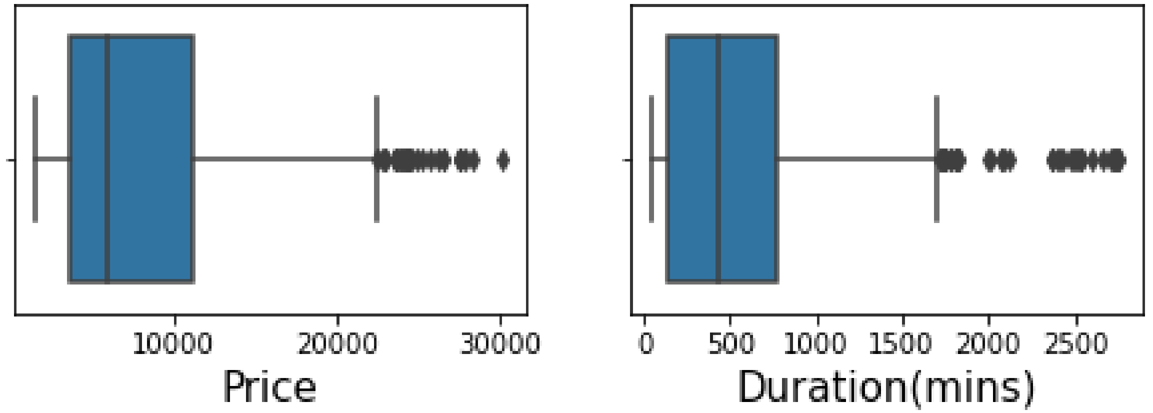
Exploratory Data Analysis

Following Observations are made from graphs above:

- There is a linear relationship between Price and flight duration.
- Indigo,Air Asia and Spicejet provide most affordable Airtickets to the destinations.

Exploratory Data Analysis

Checking for Outliers



There are considerable outliers in the columns.

Outliers were Removed using Z score method which resulted in a total data loss of 1.00%, which is within acceptable range.

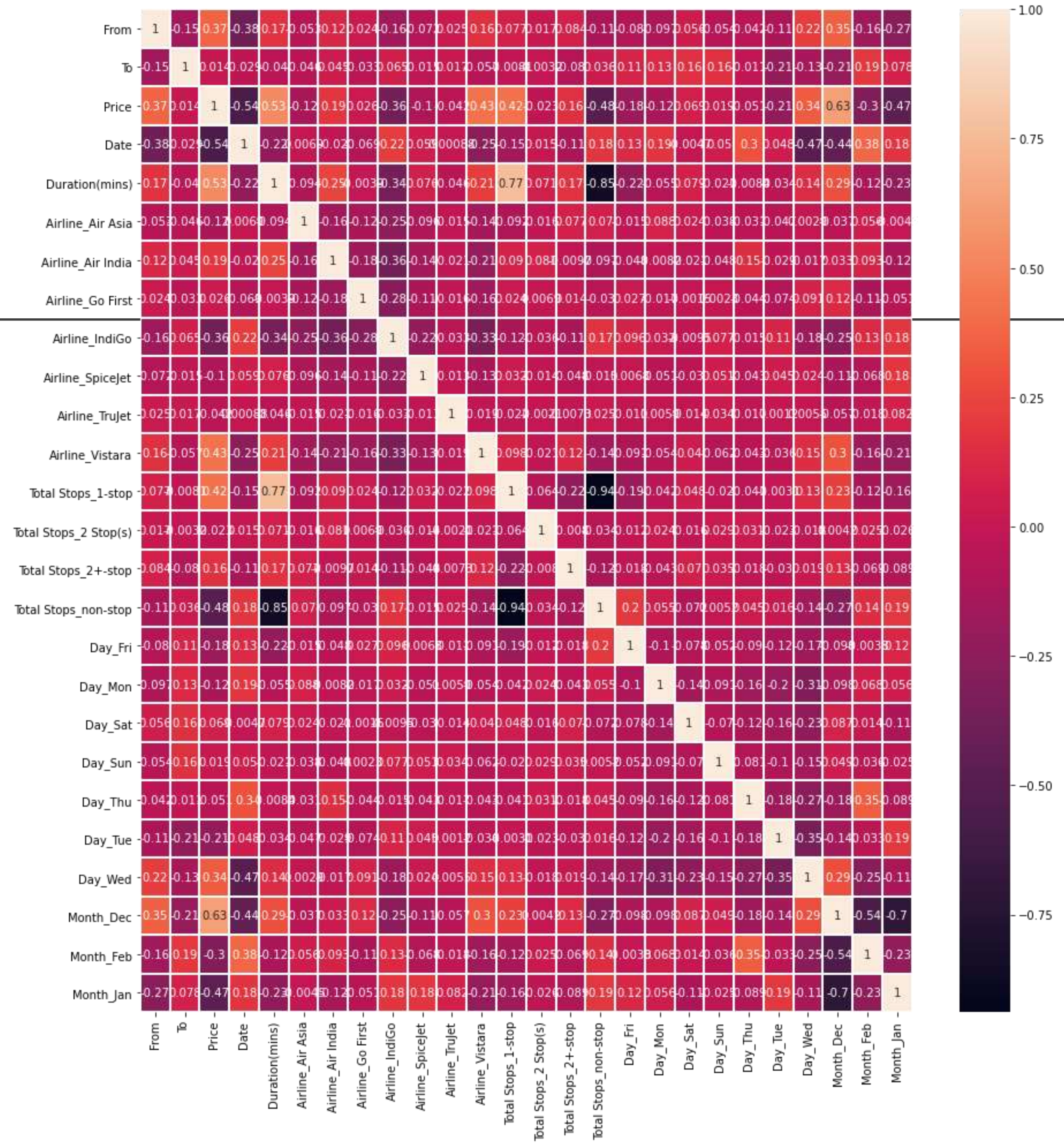
Exploratory Data Analysis

Data Normalization

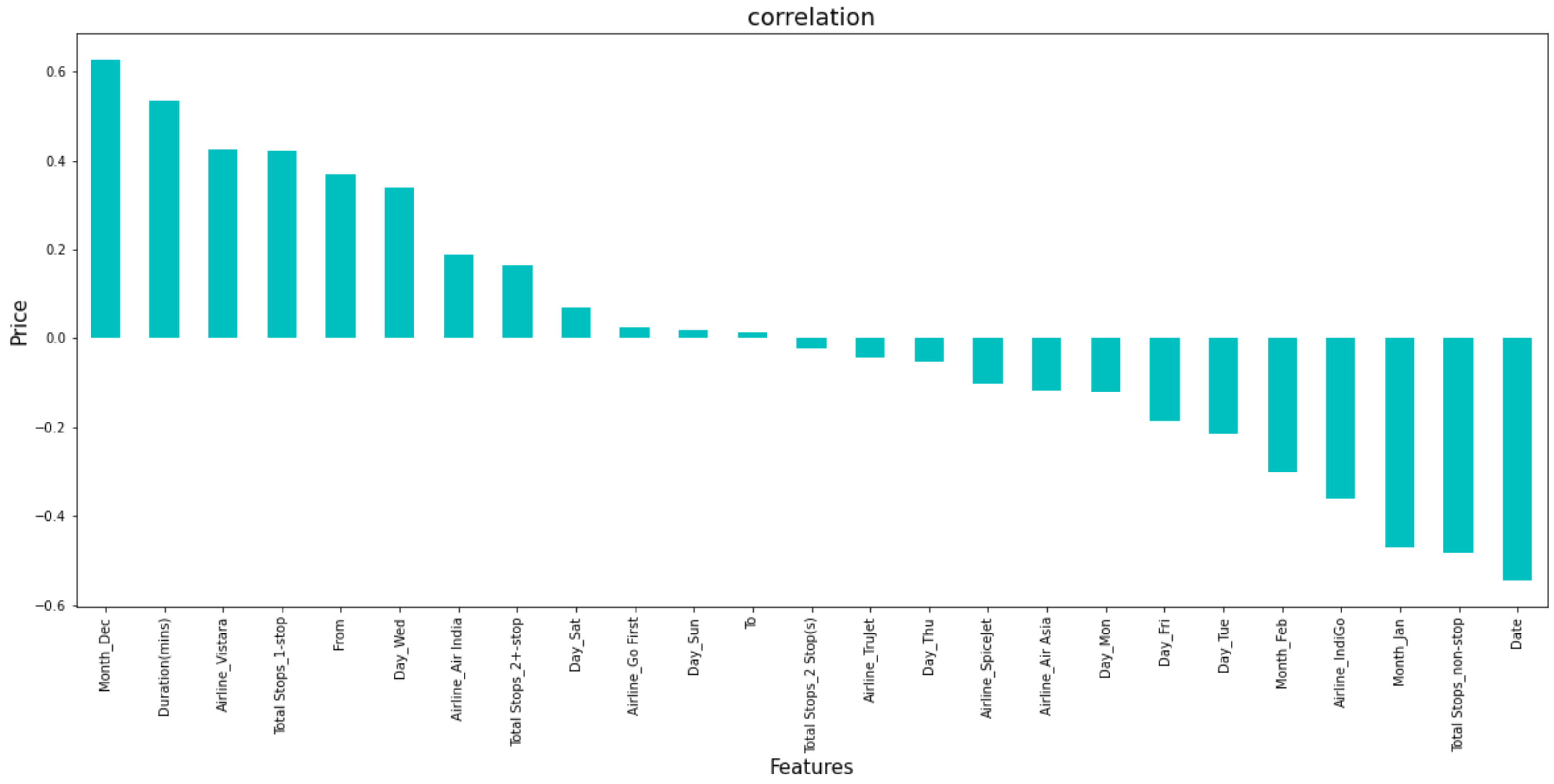
Data in Column ‘Duration(mins) was normalized using Power Transformer technique.

Exploratory Data Analysis

Finding Correlation between Feature and Target columns


From
To
Price
Date
Duration(mins)
Airline_Air Asia
Airline_Air India
Airline_Go First
Airline_IndiGo
Airline_SpiceJet
Airline_TruJet
Airline_Vistara
Total Stops_1-stop
Total Stops_2 Stop(s)
Total Stops_2+-stop
Total Stops_non-stop
Day_Fri
Day_Mon
Day_Sat
Day_Sun
Day_Thu
Day_Tue
Day_Wed
Month_Dec
Month_Feb
Month_Jan
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75

Exploratory Data Analysis



Exploratory Data Analysis

It is observed that Month_Dec, Duration(mins), Airline_Vistara,Total Stops_1-stop and From have the highest positive correlation with Price, while Date,Total Stops_non-stop,Month_Jan,Airline_IndiGo have the highest negative correlation with Price.

Model/s Development and Evaluation

Feature Selection

Features were first checked for presence of multicollinearity and then based on respective ANOVA f-score values, the feature columns were selected that would best predict the Target variable, to train and test machine learning models.

Using SelectKBest and f_classif for measuring the respective ANOVA f-score values of the columns, the best features were selected. Using StandardScaler, the features were scaled by resizing the distribution values so that mean of the observed values in each feature column is 0 and standard deviation is 1.

From sklearn.model_selection’s train_test_split, the data was divided into train and test data. Training data comprised 75% of total data where as test data comprised 25% based on the best random state that would result in best model accuracy.

Model/s Development and Evaluation

The model algorithms used were as follows:

- Ridge
- DecisionTreeRegressor
- XGBRegressor
- RandomForestRegressor:
- Support Vector Regressor:

Model/s Development and Evaluation

Regression Model Building

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.metrics import r2_score
```

Finding the Best Random State

```
from sklearn.ensemble import RandomForestRegressor
maxAcc = 0
maxRS=0
for i in range(1,100):
    x_train,x_test,y_train,y_test = train_test_split(scaled_x_best,y,test_size = .25, random_state = i)
    modRF =  RandomForestRegressor()
    modRF.fit(x_train,y_train)
    pred = modRF.predict(x_test)
    acc  = r2_score(y_test,pred)
    if acc>maxAcc:
        maxAcc=acc
        maxRS=i
print(f"Best Accuracy is: {maxAcc} on random_state: {maxRS}")
```

```
Best Accuracy is: 0.8232413568072626 on random_state: 58
```

Best random state was determined to be 58

```
x_train,x_test,y_train,y_test = train_test_split(scaled_x_best,y,test_size = .25, random_state =58)
```

```
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import Ridge
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.svm import SVR


```

```
from sklearn.metrics import r2_score,mean_squared_error
```

```
rf = RandomForestRegressor()
dt = DecisionTreeRegressor()
xg = XGBRegressor()
SV= SVR()
r=Ridge()
```

Model/s Development and Evaluation

Training The Models

```
rf.fit(x_train,y_train)
xg.fit(x_train,y_train)
SV.fit(x_train,y_train)
r.fit(x_train,y_train)
dt.fit(x_train,y_train)
```

Model/s Development and Evaluation

Analyzing Accuracy of The Models

Mean Squared Error and Root Mean Squared Error metrics were used to evaluate the Model performance. The advantage of MSE and RMSE being that it is easier to compute the gradient. As, we take square of the error, the effect of larger errors become more pronounced than smaller error, hence the model can now focus more on the larger errors.

Ridge Regression Model

```
1 y_r_pred = r.predict(x_test)
```

R2 Score

```
1 r2_score(y_test,y_r_pred)
```

0.7058259974328807

Mean Squared Error

```
1 mean_squared_error(y_test,y_r_pred)
```

7357205.386648208

Root Mean Squared Error

```
1 np.sqrt(mean_squared_error(y_test,y_r_pred))
```

2712.416890274835

Random Forest Regression Model

```
1 y_rf_pred = rf.predict(x_test)
```

R2 Score

```
1 r2_score(y_test,y_rf_pred)
```

0.8156395994141425

Mean Squared Error

```
1 mean_squared_error(y_test,y_rf_pred)
```

4610799.46031403

Root Mean Squared Error

```
1 np.sqrt(mean_squared_error(y_test,y_rf_pred))
```

2147.2772201823477

XGB Regression Model

```
1 y_xg_pred = xg.predict(x_test)
```

R2 Score

```
1 r2_score(y_test,y_xg_pred)
```

0.8098289634792951

Mean Squared Error

```
1 mean_squared_error(y_test,y_xg_pred)
```

4756121.758092391

Root Mean Squared Error

```
1 np.sqrt(mean_squared_error(y_test,y_xg_pred))
```

2180.853447183554

Support Vector Regression Model

```
1 y_svr_pred = SV.predict(x_test)
```

R2 Score

```
1 r2_score(y_test,y_svr_pred)
```

-0.07322019200984475

Mean Squared Error

```
1 mean_squared_error(y_test,y_svr_pred)
```

26840921.72935272

Root Mean Squared Error

```
1 np.sqrt(mean_squared_error(y_test,y_svr_pred))
```

5180.822495449223

Decision Tree Regression Model

```
1 y_dt_pred = dt.predict(x_test)
```

R2 Score

```
1 r2_score(y_test,y_dt_pred)
```

0.7254430163703851

Mean Squared Error

```
1 mean_squared_error(y_test,y_dt_pred)
```

6866589.505783421

Root Mean Squared Error

```
1 np.sqrt(mean_squared_error(y_test,y_dt_pred))
```

2620.4178113009807

Model/s Development and Evaluation

Using cross-validation, there are high chances that we can detect over-fitting with ease. Model Cross Validation scores were then obtained for assessing how the statistical analysis generalises to an independent data set. The models were evaluated by training several models on subsets of the available input data and evaluating them on the complementary subset of the data.

Model Cross Validation

```
from sklearn.model_selection import ShuffleSplit,cross_val_score
```

Ridge Regression

```
cross_val_score(r,scaled_x_best,y,cv=ShuffleSplit(5)).mean()
```

```
0.6751796574676406
```

Random Forest Regression

```
cross_val_score(rf,scaled_x_best,y,cv=ShuffleSplit(5)).mean()
```

```
0.7725019082554441
```

XGB Regression

```
cross_val_score(xg,scaled_x_best,y,cv=ShuffleSplit(5)).mean()
```

```
0.7703442726148106
```

SV Regression

```
cross_val_score(SV,scaled_x_best,y,cv=ShuffleSplit(5)).mean()
```

```
-0.06792748911 00552
```

Decision Tree Regression

```
cross_val_score(dt,scaled_x_best,y,cv=ShuffleSplit(5)).mean()
```

```
0.6749385369081847
```

Model/s Development and Evaluation

Interpretation of the Results

Based on comparing Accuracy Score results with Cross Validation results, it is determined that Random Forest Regressor is the best model. It also has the lowest Root Mean Squared Error score.

Model/s Development and Evaluation

Hyper Parameter Tuning

- GridSearchCV was used for Hyper Parameter Tuning of the Random Forest Regressor model. Based on the input parameter values and after fitting the train datasets The Random Forest Regressor model was further tuned based on the parameter values yielded from GridsearchCV.
- The Random Forest Regressor model displayed an accuracy of 83.55%. This model was then tested using a scaled Test Dataset.
- The model performed with good amount of accuracy.

```
from sklearn.model_selection import GridSearchCV
```

```
parameter = {'n_estimators':[30,60,80],'max_depth': [40,50,80],'min_samples_leaf':[5,10,20],'min_samples_split':[2,5,10],'cr
```

```
GridCV = GridSearchCV(RandomForestRegressor(),parameter,cv=ShuffleSplit(5),n_jobs = -1,verbose = 1)
```

```
GridCV.fit(x_train,y_train)
```

```
Fitting 5 folds for each of 486 candidates, totalling 2430 fits
```

```
GridSearchCV(cv=ShuffleSplit(n_splits=5, random_state=None, test_size=None, train_size=None),
             estimator=RandomForestRegressor(), n_jobs=-1,
             param_grid={'criterion': ['mse', 'mae'], 'max_depth': [40, 50, 80],
                         'max_features': ['auto', 'sqrt', 'log2'],
                         'min_samples_leaf': [5, 10, 20],
                         'min_samples_split': [2, 5, 10],
                         'n_estimators': [30, 60, 80]},
             verbose=1)
```

```
GridCV.best_params_
```

```
{'criterion': 'mse',
 'max_depth': 80,
 'max_features': 'auto',
 'min_samples_leaf': 5,
 'min_samples_split': 2,
 'n_estimators': 80}
```

```
Best_mod = RandomForestRegressor(n_estimators = 80,criterion = 'mse', max_depth= 80, max_features = 'auto',min_samples_leaf

Best_mod.fit(x_train,y_train)
```

```
RandomForestRegressor(max_depth=80, min_samples_leaf=5, n_estimators=80)
```

```
rfpred = Best_mod.predict(x_test)
acc = r2_score(y_test,rfpred)
print(acc*100)
```

```
83.55267438661498
```

```python
Prediction_accuracy = pd.DataFrame({'Predictions': mod.predict(scaled_x_best), 'Actual Values': y})
Prediction_accuracy.head(30)
```

	Predictions	Actual Values
0	2464.683598	2456
1	2462.606661	2456
2	2462.606661	2456
3	2462.606661	2456
4	2472.924672	2456
5	2470.397408	2456
6	2527.726671	2456
7	2527.726671	2456
8	2648.920489	2456
9	2648.920489	2456
10	2480.476117	2343
11	2472.233929	2343
12	2472.233929	2343
13	2498.094946	2343
14	2609.755068	2637
15	2596.585589	2637

Model/s Development and Evaluation

In summary, Based on the visualizations of the feature-column relationships, it is determined that, Features like Source,month,Duration,Total Stops,Airline,Date are some of the most important features to predict the label values. Random Forest Regressor Performed the best out of all the models that were tested. It also worked well with the outlier handling.

CONCLUSION

Key Findings and Conclusions of the Study and Learning Outcomes with respect to Data Science

Based on the in-depth analysis of the Flight Price Prediction Project, The Exploratory analysis of the datasets, and the analysis of the Outputs of the models the following observations are made:

- Air Fare attributes like Date,Month,Duration,Total Stops etc play a big role in influencing the used Flight price.
- Airline Brand also has a very important role in determining the used Flight Ticket price.

CONCLUSION

- Various plots like Barplots,Countplots and Lineplots helped in visualising the Feature-label relationships which corroborated the importance of Air Fare features and attributes for estimating Flight Ticket Prices.
- Due to the Training dataset being very small, only very small amount of the outliers was removed to ensure proper training of the models.
- Therefore, Random Forest Regressor, which uses averaging to improve the predictive accuracy and controls over-fitting. performed well despite having to work on small dataset and produced good predictions that can be understood easily.

CONCLUSION

<u>Learning Outcomes of the Study in respect of Data Science</u>

- Data cleaning was a very important step in removing plenty of anomalous data from the huge dataset that was provided.
- Visualising data helped identify outliers and the relationships between target and feature columns as well as analysing the strength of correlation that exists between them.

CONCLUSION

Limitations of this work and Scope for Future Work

- A small dataset to work with posed a challenge in building highly accurate models. This project also relied heavily on historical data and was unable to account for various other factors that influence demand and ticket pricing like pandemic status affecting demand, government regulations on air travel, shifting in routes, weather conditions, etc.

- Most airline companies also do no publicly make available their ticket pricing strategies, which makes gathering price and air fare related data sets using web scraping the only means to build a dataset for building predicting models.

- Availability of more features and a larger dataset would help build better models.