PROJECT REPORT ON

"FLIGHT PRICE PREDICTION"

SUBMITTED BY

Ankit Dadarwala

ACKNOWLEDGMENT

I would like to express my special gratitude to "Flip Robo" team, who has given me this opportunity to deal with a beautiful dataset and it has helped me to improve my analyzation skills. And I want to express my huge gratitude to Mr. Shubham Yadav(SME Flip Robo), he is the person who has helped me in tickets to get out of all the difficulties I faced while doing the project. And also inspired me in so many aspects and also encouraged me a lot with his valuable words and with his unconditional support I have ended up with a beautiful Project.

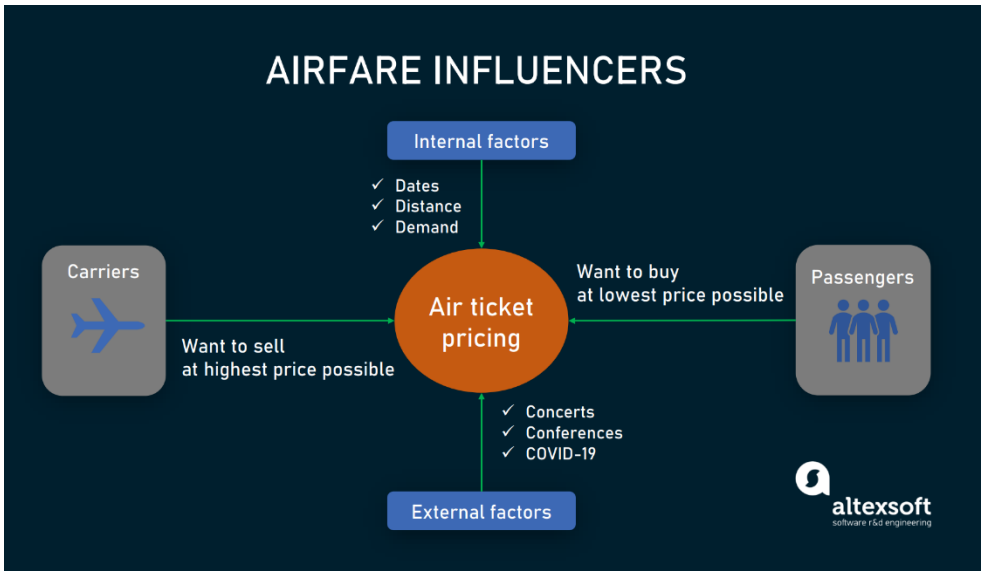
A huge thanks to my academic team "Data trained" who are the reason behind what I am today. Last but not least my parents who have been my backbone in every step of my life. And also thank you for many other persons who has helped me directly or indirectly to complete the project.

INTRODUCTION

1.1 Business Problem Framing:

The airline industry is considered as one of the most sophisticated industry in using complex pricing strategies. Nowadays, ticket prices can vary dynamically and significantly for the same flight, even for nearby seats. The ticket price of a specific flight can change up to 7 times a day. Customers are seeking to get the lowest price for their ticket, while airline companies are trying to keep their overall revenue as high as possible and maximize their profit. However, mismatches between available seats and passenger demand usually leads to either the customer paying more or the airlines company losing revenue. Airlines companies are generally equipped with advanced tools and capabilities that enable them to control the pricing process. However, customers are also becoming more strategic with the development of various online tools to compare prices across various airline companies. In addition, competition between airlines makes the task of determining optimal pricing is hard for everyone

1.2 Conceptual Background of the Domain Problem:



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)

Analytical Problem Framing

2.1 Mathematical/ Analytical modelling of the Problem

− Mathematical/Analytical modelling of the problem

For the given flight price prediction project I have scraped the flight prices along with some other features from a well known website that is ‘yatra.com’. And this data is framed into a data frame and saved into a .csv file. After that I have fetched this data set and performed some data processing and some EDA. The data set is having around 1765 rows and 9 columns.

2.2 Data Sources and their formats

```
# import dataset
df=pd.read_csv('Flightprice.csv')
df
```

	Unnamed: 0	name	deprt	arrvt	sorct	dest	stops	durat	price
0	0	Air India	21:25	09:05\n+ 1 day	Surat	Mumbai	1 Stop	11h 40m	10,674
1	1	Air India	21:25	10:10\n+ 1 day	Surat	Mumbai	1 Stop	12h 45m	10,674
2	2	Air India	21:25	11:15\n+ 1 day	Surat	Mumbai	1 Stop	13h 50m	10,674
3	3	Air India	21:25	18:00\n+ 1 day	Surat	Mumbai	2 Stop(s)	20h 35m	10,674
4	4	Air India	21:25	20:00\n+ 1 day	Surat	Mumbai	1 Stop	22h 35m	10,674
...	...	...	...	...	...	...	...	...	...
1760	1760	Air India	17:20	20:15\n+ 1 day	New Delhi	Hyderabad	1 Stop	26h 55m	11,310
1761	1761	Vistara	19:50	12:15\n+ 1 day	New Delhi	Hyderabad	1 Stop	16h 25m	11,730
1762	1762	Air India	21:15	21:35\n+ 1 day	New Delhi	Hyderabad	3 Stop(s)	24h 20m	12,071
1763	1763	Vistara	17:10	12:15\n+ 1 day	New Delhi	Hyderabad	2 Stop(s)	19h 05m	12,490
1764	1764	Vistara	19:55	12:15\n+ 1 day	New Delhi	Hyderabad	2 Stop(s)	16h 20m	15,498

1765 rows × 9 columns

Features Information:

1. name => airlines name
2. deprt => depature time
3. arrvt => arrival time
4. sorct => source city
5. dest => destination city
6. stops => No.of stops
7. duration => total duration traveling time
8. price => traveling charges of airlines

```
[855]: # dataset information
       df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1765 entries, 0 to 1764
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Unnamed: 0  1765 non-null   int64 
 1   name        1765 non-null   object
 2   deprt       1765 non-null   object
 3   arrvt       1765 non-null   object
 4   sorct       1765 non-null   object
 5   dest        1765 non-null   object
 6   stops       1765 non-null   object
 7   durat       1765 non-null   object
 8   price       1765 non-null   object
dtypes: int64(1), object(8)
memory usage: 124.2+ KB
```

2.3 Data Pre-processing Done

1. As a first step I have imported required libraries and I have imported the dataset which was scrap and save in csv format.
2. Then I did all the analysis like data types, checking shape, unique, value counts, info, duplicates value null values etc.
3. In that I found that all are objective data types so have change data types in that have some date-time format also int as well float data types.
4. While checking for null values I found no null values in the dataset.
5. I have also dropped Unnamed:0, column as I found they are useless.

```
[862]: df["arrvtime"]= new[0]
       df.drop(['arrvt'],axis=1,inplace=True)

[863]: # remove unnecessary data
       df.drop('Unnamed: 0',axis=1,inplace=True)

[864]: # change data type of arrival time into hours and min
       df["arrvtime_h"] = pd.to_datetime(df['arrvtime']).dt.hour
       df["arrvtime_m"] = pd.to_datetime(df['arrvtime']).dt.minute

[865]: # change data type of departure time into hours and min
       df['deprt_h']=pd.to_datetime(df['deprt']).dt.hour
       df['deprt_m']=pd.to_datetime(df['deprt']).dt.minute

[866]: # drop axis after conversion
       df.drop(['deprt','arrvtime'],axis=1,inplace=True)

[867]: # value count of No of stops
       df['stops'].value_counts()

t[867]: 2 Stop(s)    725
        1 Stop       690
        Non Stop     210
        3 Stop(s)    125
        4 Stop(s)     15
        Name: stops, dtype: int64

[868]: # assign the value to no of stop columns
       df['stops']=df['stops'].replace({'Non Stop':0,'1 Stop':1,'2 Stop(s)':2,
                                        '3 Stop(s)':3,'4 Stop(s)':4})
```

2.4 Data Inputs- Logic- Output Relationships

- ✓ As now in dataset I have numerical and categorical values so I used Count plot, Bar plot, Box plot and Distribution plot.
- ✓ Used count plot as compare with price to show insight of the data
- ✓ Box plot to show the range of the price
- ✓ Heatmap to show the correlation with target and also inter correlation with other columns.

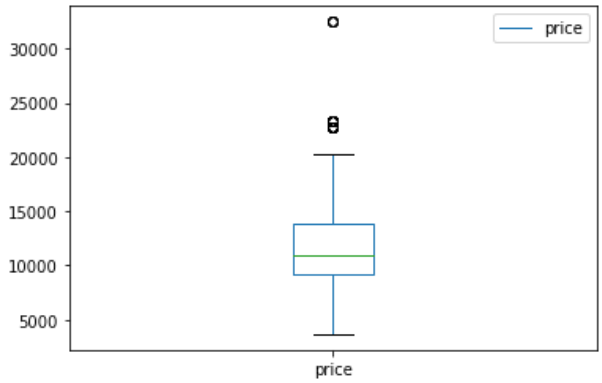
2.6 Hardware and Software Requirements and Tools Used

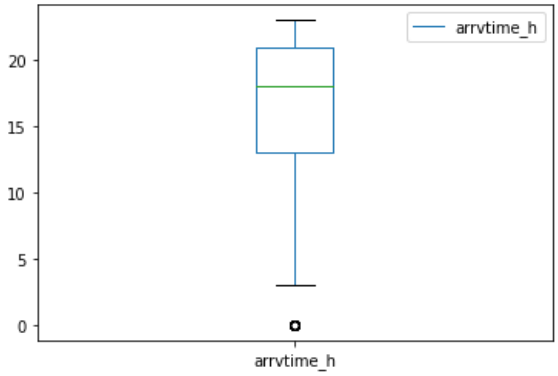
- ✓ Device name: HP Pavilion
- ✓ Processor: AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx 2.10 GHz
- ✓ RAM: 8.00 GB
- ✓ System type: 64-bit operating system, x64-based processor
- ✓ Jupyter NoteBooks Version : 6.4.3
- ✓ Python3 vesrion : 3.8.8

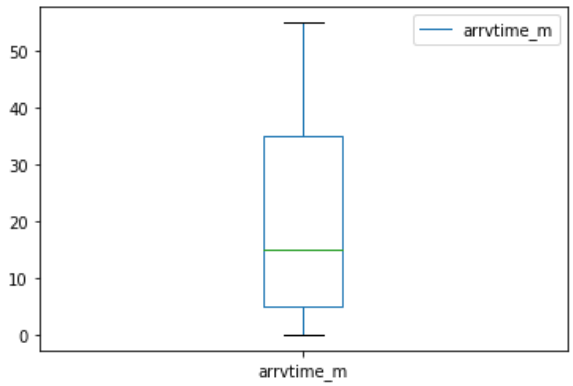
❖ **Libraries required** :-

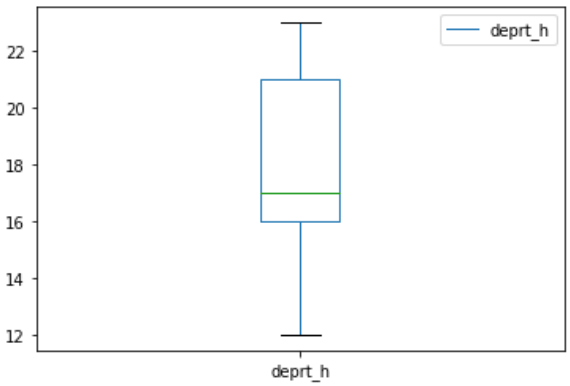
- ✓ Import pandas as pd
- ✓ Import numpy as np
- ✓ Import matplotlib.pyplot as plt
- ✓ Import seaborn as sns
- ✓ Import warings
- ✓ From sklearn.model_selection import train_test_split,GridSearchCV
- ✓ from sklearn.preprocessing import StandardScaler
- ✓ from sklearn.ensemble import RandomForestRegressor
- ✓ from sklearn.ensemble import AdaBoostRegressor
- ✓ from sklearn.ensemble import GradientBoostingRegressor
- ✓ from sklearn.ensemble import BaggingRegressor
- ✓ from sklearn.tree import DecisionTreeRegressor
- ✓ from sklearn.svm import SVR
- ✓ from sklearn.neighbors import KNeighborsRegressor
- ✓ from linear_model import LinearRegression,SGDClassifier,Lasso,Ridge
- ✓ from xgboost import XGBRegressor
- ✓ from sklearn.metrics import mean squared error
- ✓ from sklearn.metrics import mean absolute error
- ✓ from sklearn.metrics import explain variance score
- ✓ from sklearn.model_selection import cross val score

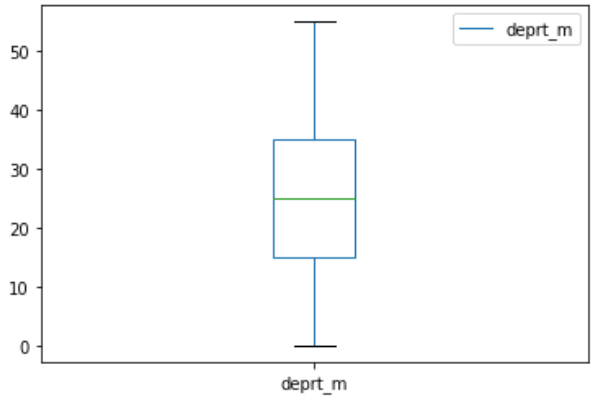
Data Analysis and Visualization

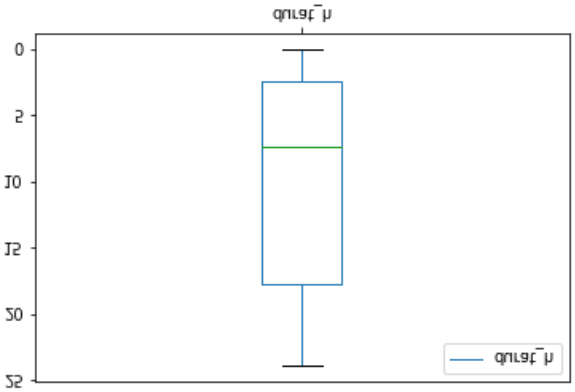


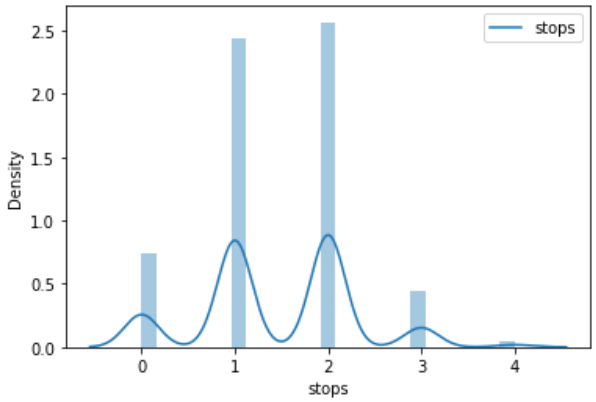


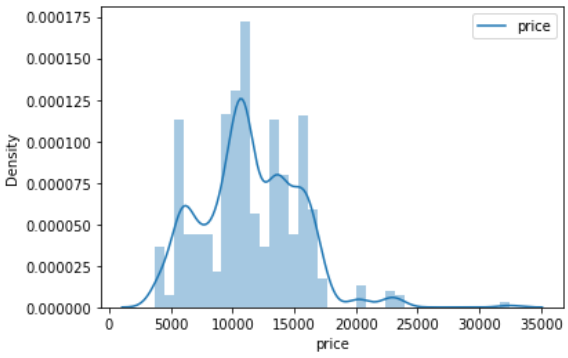


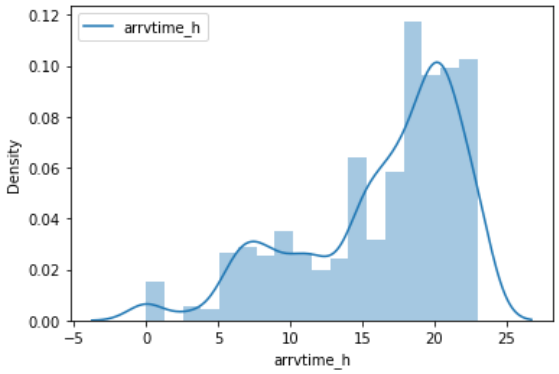


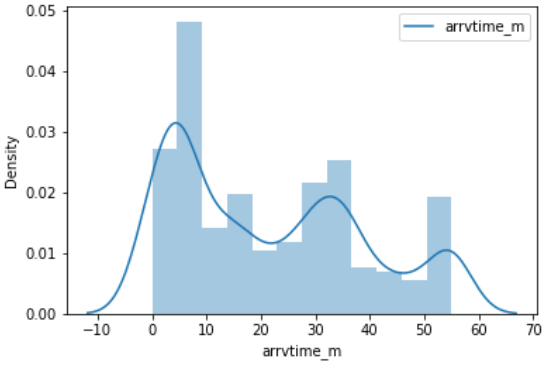


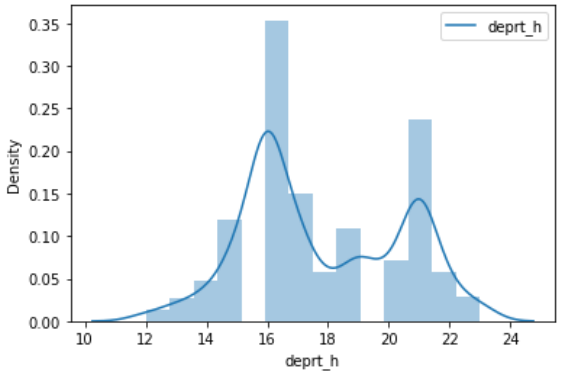


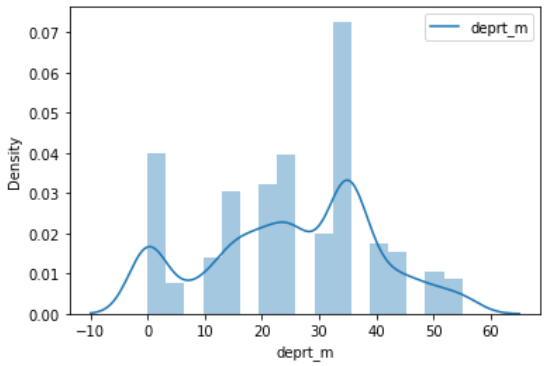


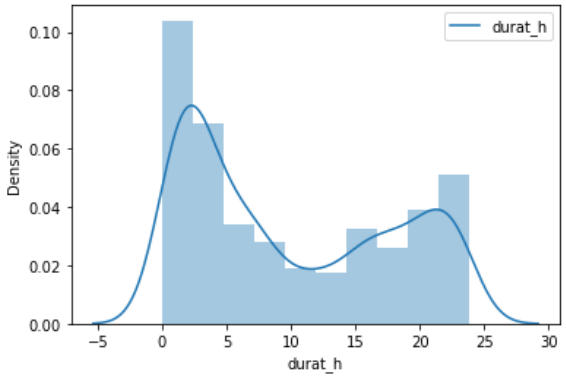


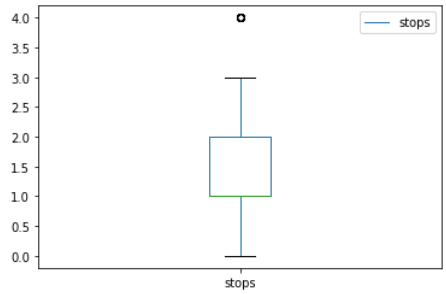

arrvtime_h
Density
arrvtime_h


arrvtime_m
Density
arrvtime_m

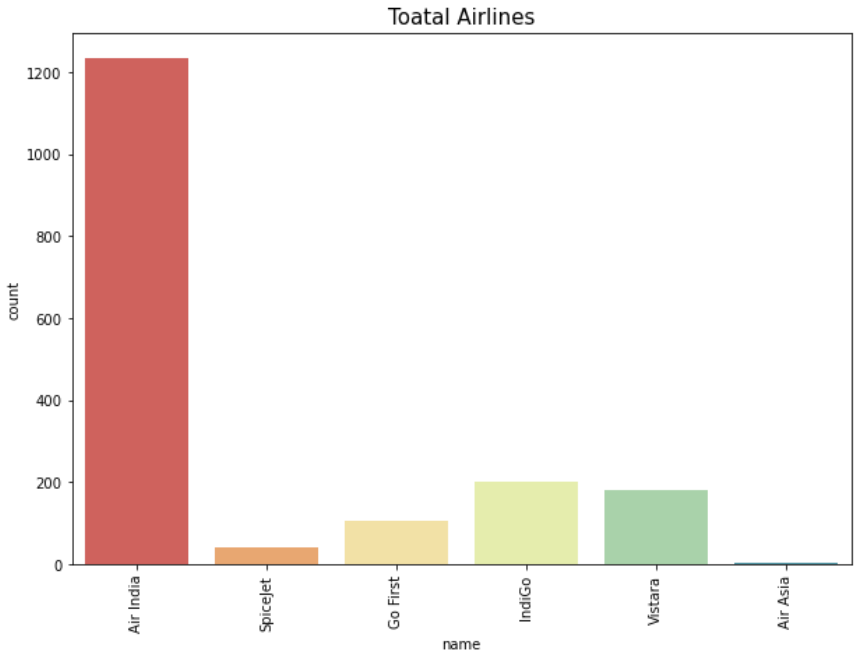

deprt_h
Density
deprt_h


deprt_m
Density
deprt_m


durat_h
Density
durat_h

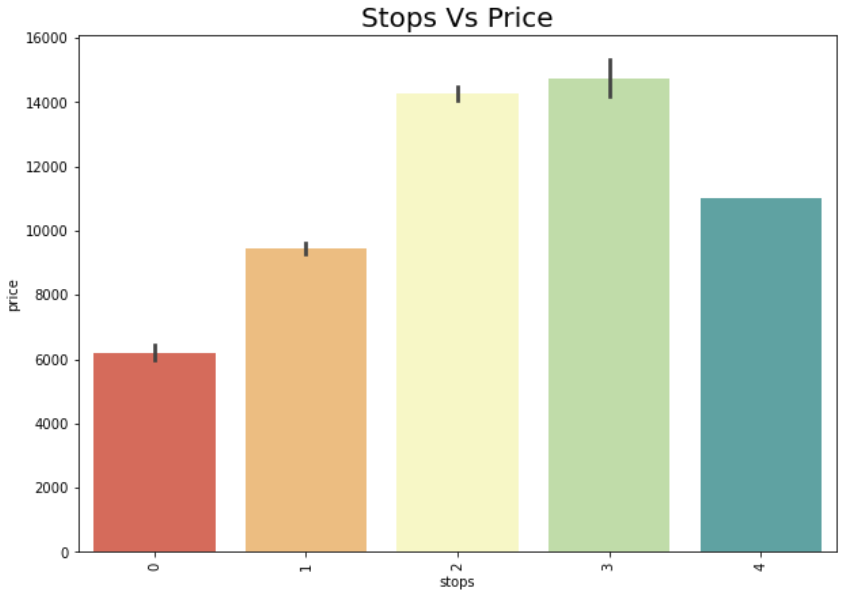

stops
stops

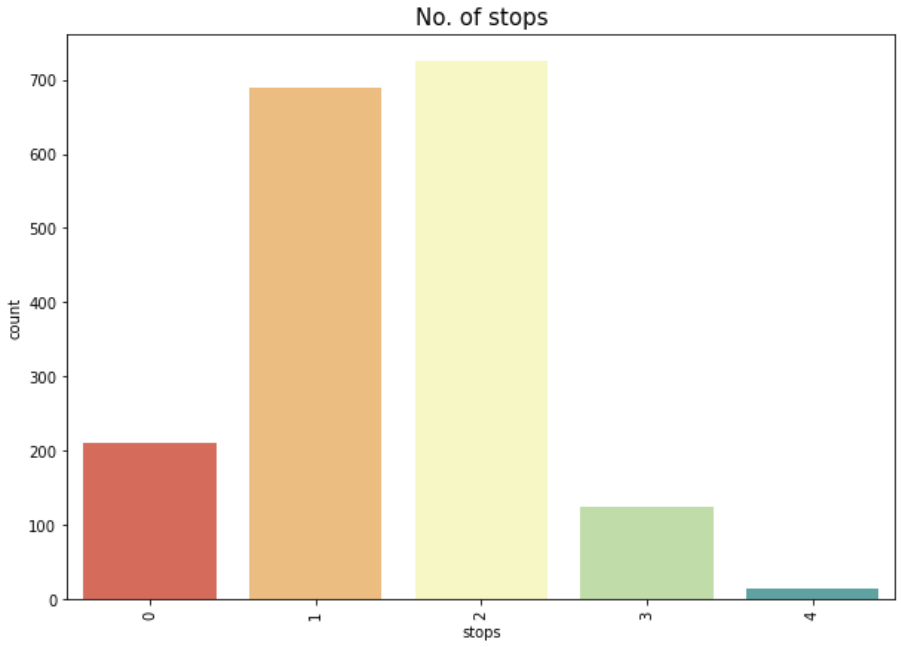
EDA

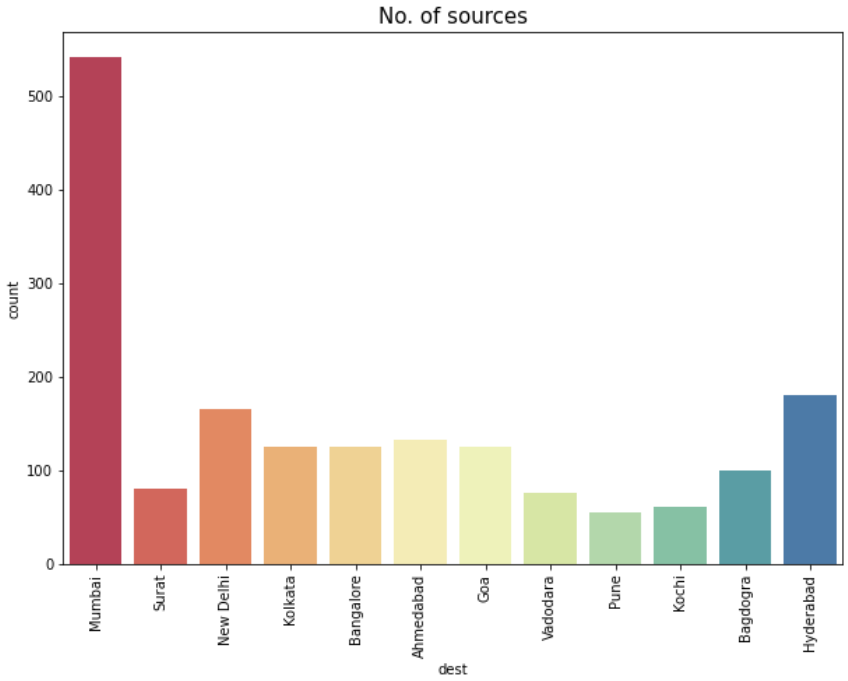


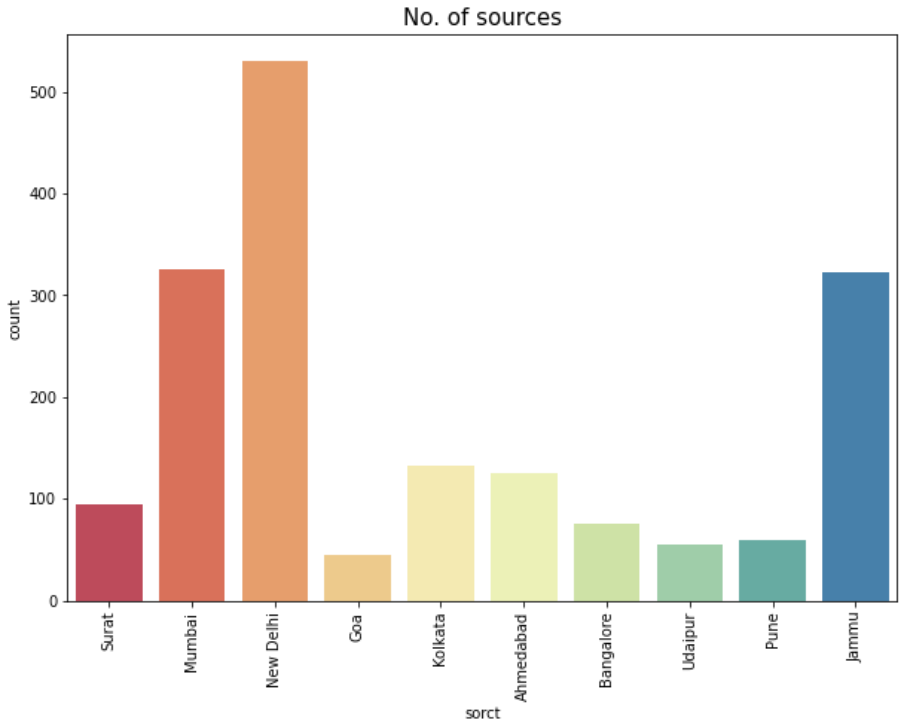
- Air India airlines are most preferable for travelling as from the graph we show.

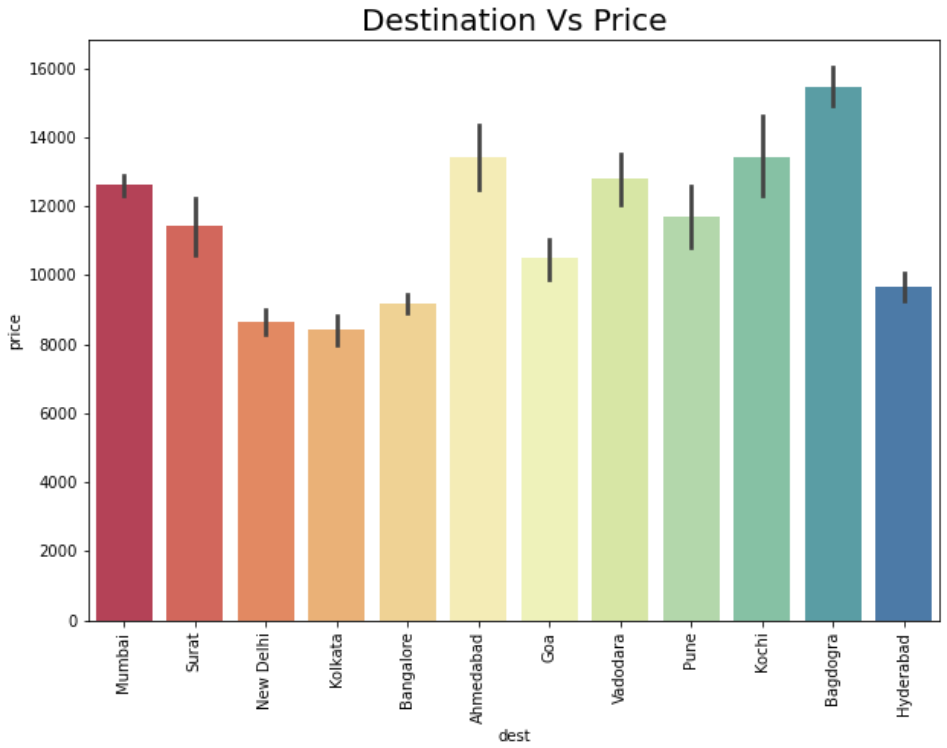
- Air India and Vistara airlines price are high as compare to others

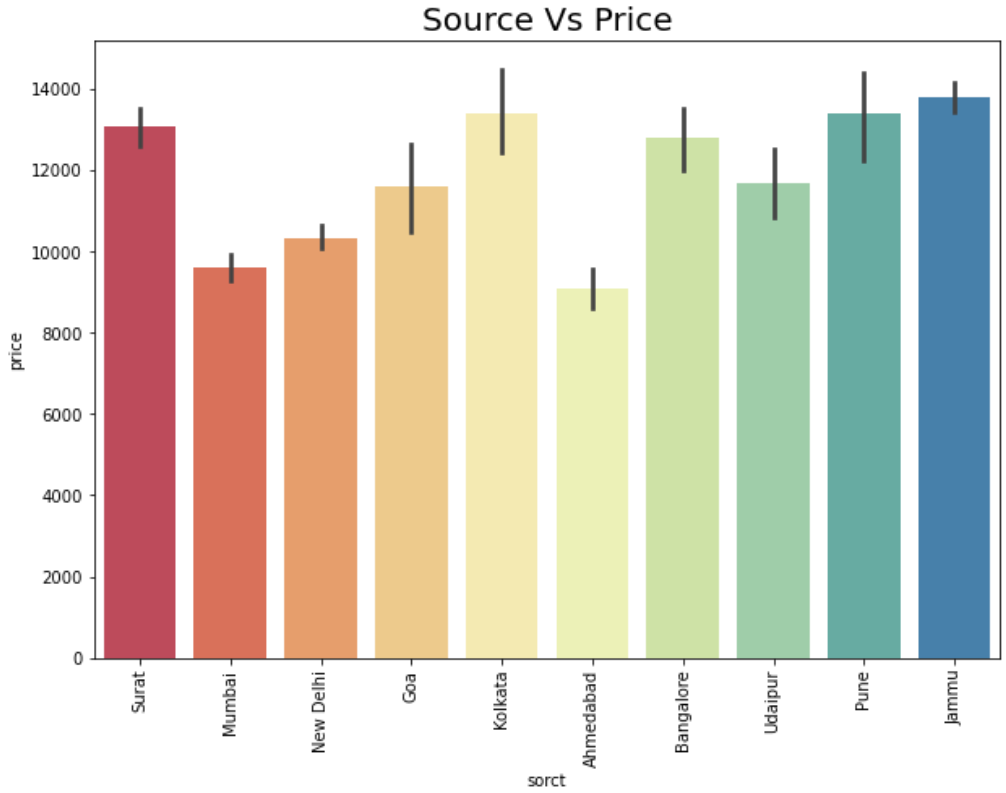

Stops Vs Price
price
stops
0
1
2
3
4
16000
14000
12000
10000
8000
6000
4000
2000
0

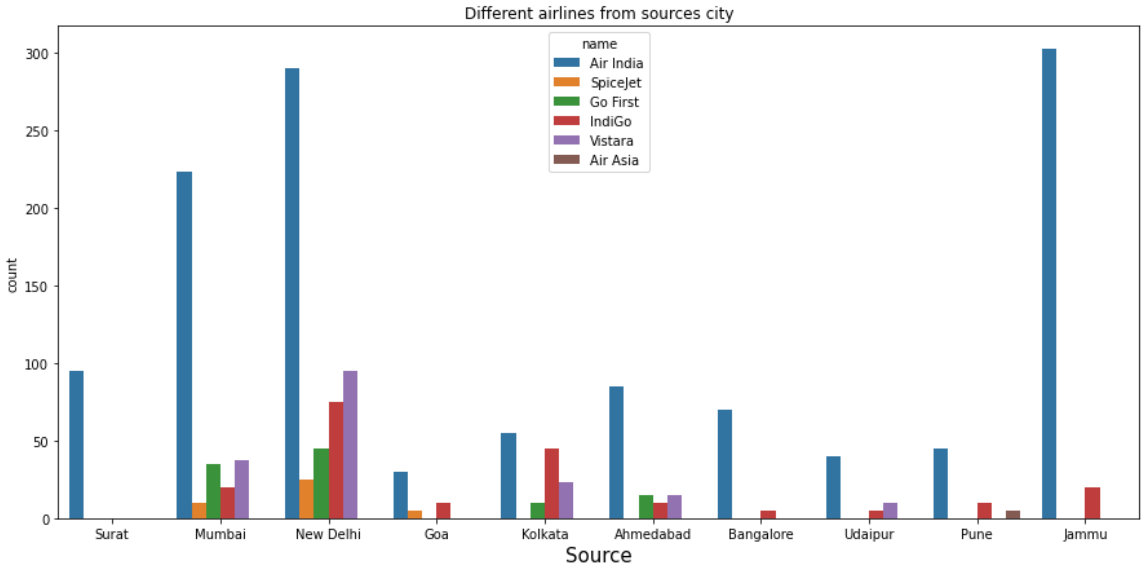

No. of stops
count
stops
0
1
2
3
4
700
600
500
400
300
200
100
0

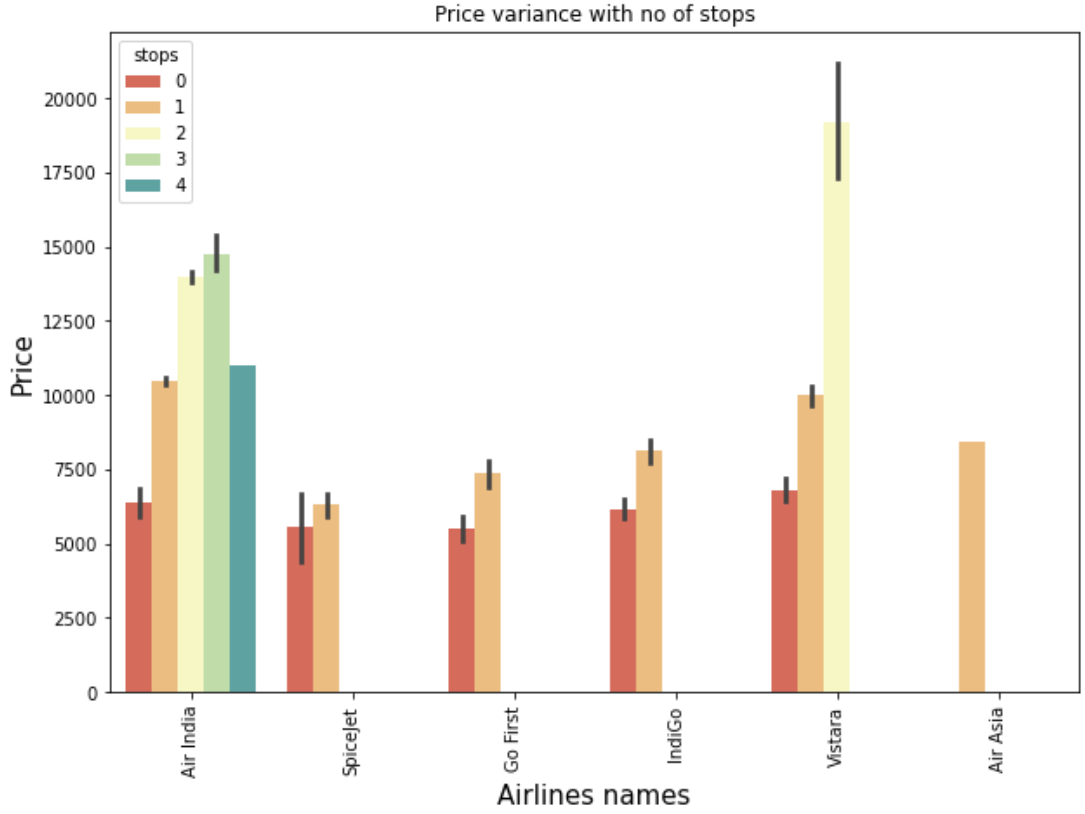

No. of sources
count
0
100
200
300
400
500
Mumbai
Surat
New Delhi
Kolkata
Bangalore
Ahmedabad
Goa
Vadodara
Pune
Kochi
Bagdogra
Hyderabad
dest

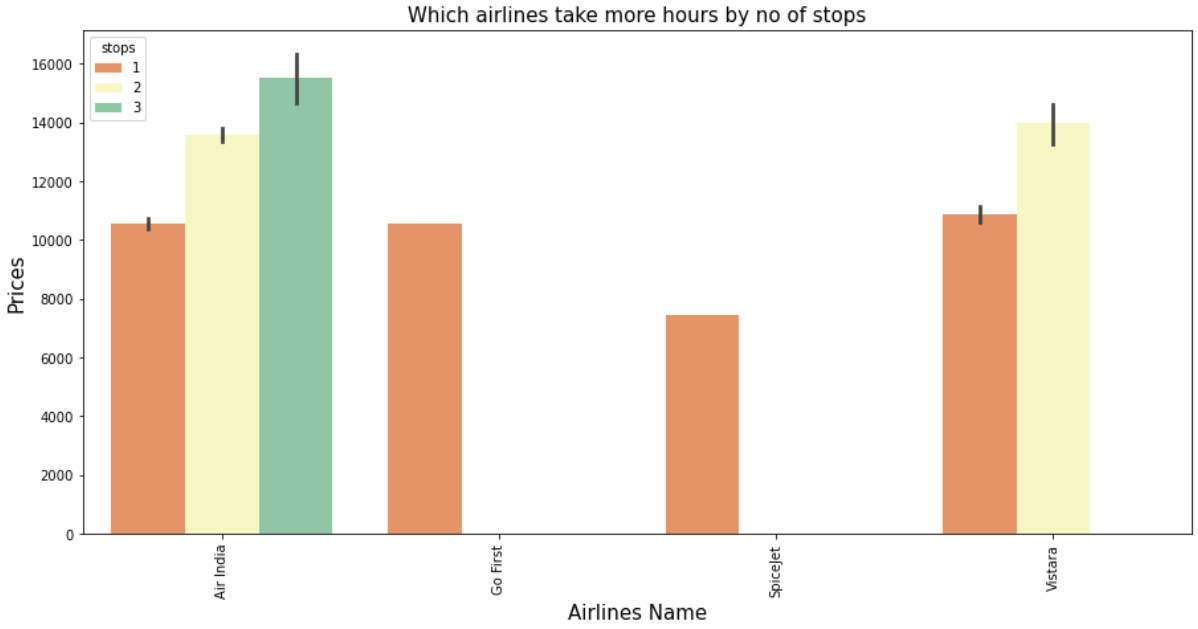

No. of sources
count
0
100
200
300
400
500
Surat
Mumbai
New Delhi
Goa
Kolkata
Ahmedabad
Bangalore
Udaipur
Pune
Jammu
sorct

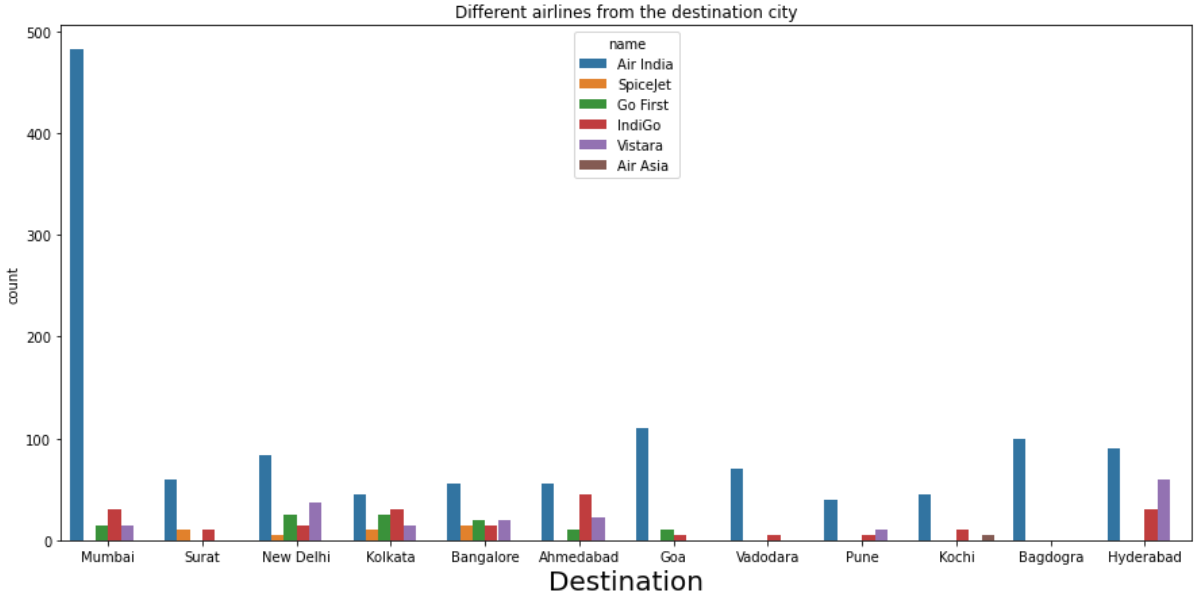

Destination Vs Price
price
dest
Mumbai
Surat
New Delhi
Kolkata
Bangalore
Ahmedabad
Goa
Vadodara
Pune
Kochi
Bagdogra
Hyderabad

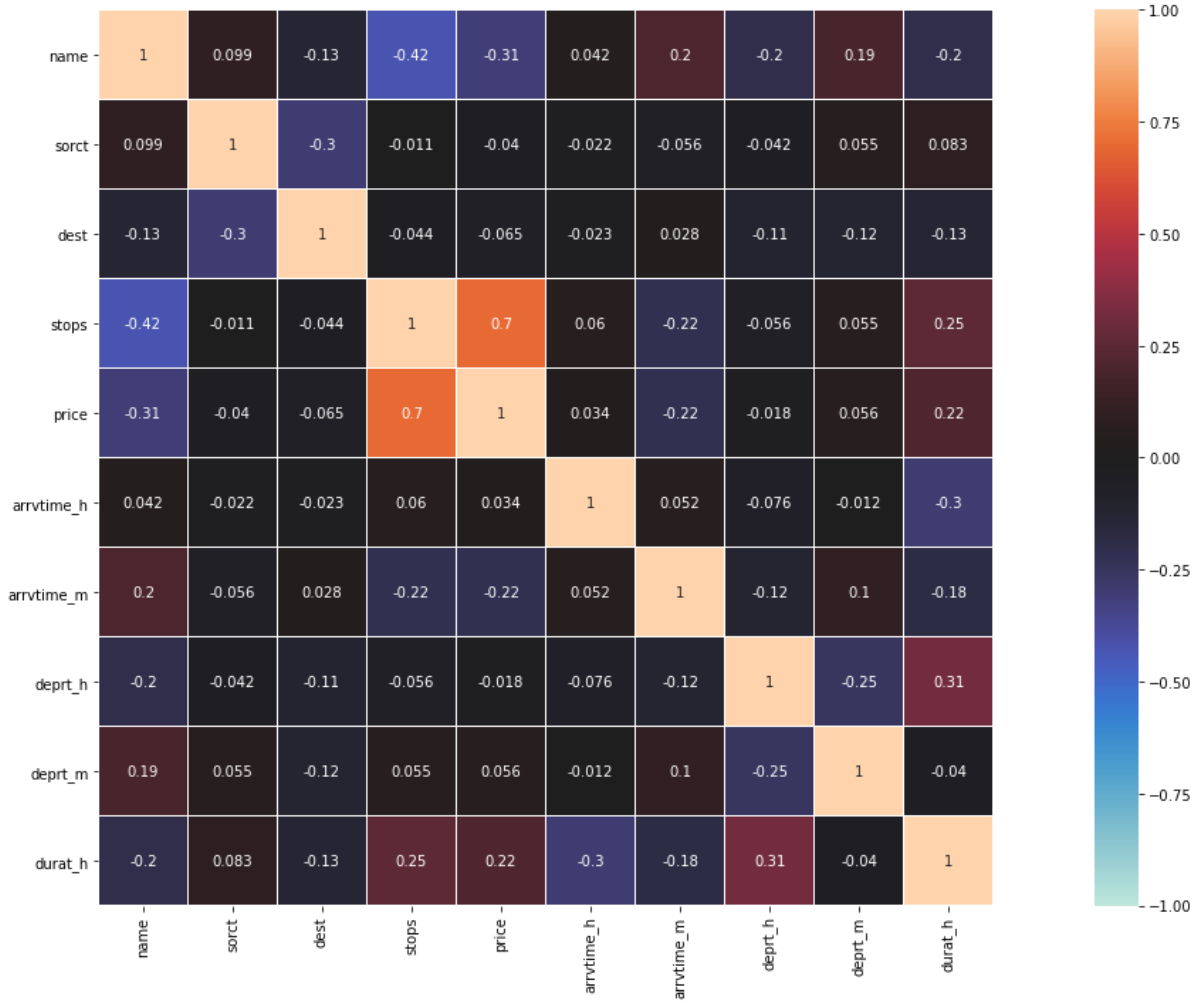

Source Vs Price
price
sorct
Surat
Mumbai
New Delhi
Goa
Kolkata
Ahmedabad
Bangalore
Udaipur
Pune
Jammu


Different airlines from sources city
name
Air India
SpiceJet
Go First
IndiGo
Vistara
Air Asia
count
0
50
100
150
200
250
300
Surat
Mumbai
New Delhi
Goa
Kolkata
Ahmedabad
Bangalore
Udaipur
Pune
Jammu
Source


Price variance with no of stops
stops
0
1
2
3
4
Price
0
2500
5000
7500
10000
12500
15000
17500
20000
Air India
SpiceJet
Go First
IndiGo
Vistara
Air Asia
Airlines names


Which airlines take more hours by no of stops
stops
1
2
3
Prices
16000
14000
12000
10000
8000
6000
4000
2000
0
Air India
Go First
SpiceJet
Vistara
Airlines Name


Different airlines from the destination city
name
Air India
SpiceJet
Go First
IndiGo
Vistara
Air Asia
count
500
400
300
200
100
0
Mumbai
Surat
New Delhi
Kolkata
Bangalore
Ahmedabad
Goa
Vadodara
Pune
Kochi
Bagdogra
Hyderabad
Destination



3.2 Testing of Identified Approaches (Algorithms)

Since continues was my target and it was a Regression column , so this particular problem was Regression problem. And I have used all Regressor algorithms to build my model. By looking into the difference of R2 score score and cross validation score I found XGBOOST Regressor as a best model. Also to get the best model we have to run through multiple models and to avoid the confusion of overfitting we have go through cross validation and select best random state. Below are the list of Regressor algorithms I have used in my project.

1. Linear Regression
2. Lasso
3. Ridge
4. SGD Regressor
5. XGB Regressor
6. Decision Tree Regressor
7. SVM Regressor
8. KNeighbors Regressor
9. Random Forest Regressor
10. Gradient Boosting Regressor
11. Bagging Regressor

12. AdaBoost Regressor

3.3 Key Metrics for success in solving problem under consideration

- R-square is a comparison of residual sum of squares *(SSres)* with total sum of squares*(SStot)*. Total sum of squares is calculated by summation of squares of perpendicular distance between data points and the average line.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

- Where *SSres* is the residual sum of squares and *SStot* is the total sum of squares.
- Absolute mean squared error metrics, **Absolute Error is the amount of error in your measurements.** It is the difference between the measured value and "true" value.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |\boldsymbol{x}_i - \boldsymbol{x}|$$

- R-square is the main metric which I will use in this regression analysis.
- **Cross val score**: To run cross-validation on multiple metrics and also to return train scores, fit times and score times. Get predictions from each split of cross-validation for diagnostic purposes. Make a scorer from a performance metric or loss function.

3.4 Run and evaluate select models

1). Hyper Parameter Tuning for selected best model

Select best model for hypertunning

```
4]: #Lets selects different parameters for tuning
    grid_params = {
                    'n_estimators': [50,100,500,700],
                    'criterion' : ["mse", "mae"],
                    'max_features' : ["auto", "sqrt", "log2"],
                    }

5]: #train the model with given parameters using GridSearchCV
    clf=  GridSearchCV(RandomForestRegressor(), grid_params,verbose=1,refit=True,n_jobs=-1, cv = 5)
    clf.fit(x_train,y_train)

    Fitting 5 folds for each of 24 candidates, totalling 120 fits

5]: GridSearchCV(cv=5, estimator=RandomForestRegressor(), n_jobs=-1,
                 param_grid={'criterion': ['mse', 'mae'],
                             'max_features': ['auto', 'sqrt', 'log2'],
                             'n_estimators': [50, 100, 500, 700]},
                 verbose=1)

6]: clf.best_params_

6]: {'criterion': 'mse', 'max_features': 'sqrt', 'n_estimators': 700}

7]: clf.best_score_

7]: 0.9954745772496443
```

2). Result after Hyper parameter tuning

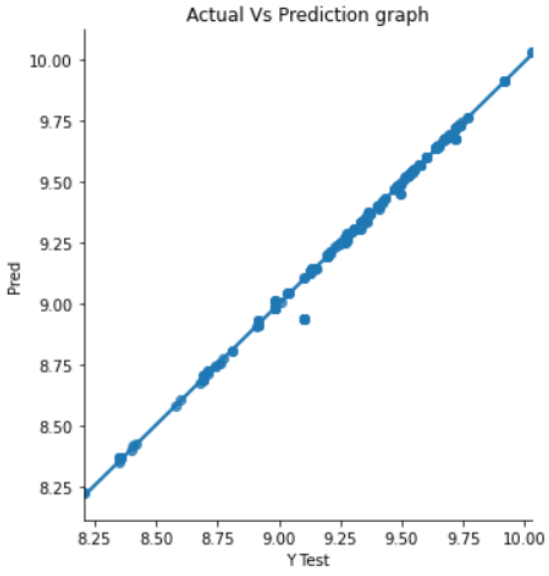
```
rf=RandomForestRegressor(criterion='mse',max_features='sqrt',n_estimators=700)
rf.fit(x_train,y_train)
print(rf,'\n',rf.score(x_train,y_train)*100)
pred_rf=rf.predict(x_test)
crs_score=cross_val_score(rf,X,np.log(y),cv=5,scoring='r2')
print('cross value score',crs_score.mean())
print('R2_score :',r2_score(y_test,pred_rf)*100)
print('error1:\n:',mean_absolute_error(y_test,pred_rf))
print('RSME:\n:',np.sqrt(mean_squared_error(y_test,pred_rf)))
```

```
RandomForestRegressor(max_features='sqrt', n_estimators=700)
 99.98262321365286
cross value score 0.9988895287448093
R2_score : 99.63170187777656
error1:
: 0.006329732224478071
RSME:
: 0.021854561073292804
```

- **I have chosen all parameters of RandomForest Regressor, after tunning the model with best parameters I have R2 score 99.63%**

3). Regression plot of Prediction

```
data = pd.DataFrame({'Y Test':y_test , 'Pred':pred_rf},columns=['Y Test','Pred'])
sns.lmplot(x='Y Test',y='Pred',data=data,palette='rainbow')
plt.title('Actual Vs Prediction graph')
plt.show()
```



3.5 Interpretation of the Results

- ✓ Scraping the flight price data from websites and from different cities are challenging.
- ✓ Firstly, the datasets were not having duplicates value
- ✓ But there was no null values in dataset.
- ✓ And proper plotting for proper type of features will help us to get better insight on the data. I found maximum categorical columns in the dataset so I have chosen bar, count, pie plot to see the relation between target and features.
- ✓ I notice a skewness in the data so we have chosen proper methods to deal with skewness. If we skewness we may end up with a bad model which has less accuracy.
- ✓ Then scaling dataset has a good impact like it will help the model not to get biased. Since we have removed outliers and skewness.
- ✓ We have to use multiple models while building model using dataset as to get the best model out of it.
- ✓ And we have to use multiple metrics like R2 score, MSE, RMSE and cross val score which will help us to decide the best model.
- ✓ I found RandomForest Regressor as the best model with 99.63.% R2 score..
- ✓ At last I have predicted Flight price using saved model. It was good!! that I was able to get the predictions near to actual values.

CONCLUSION

4.1 Key Findings and Conclusions of the Study

In this project report, we have used machine learning algorithms to predict the used car price. We have mentioned the step-by-step procedure to analyse the dataset and finding the correlation between the features. Thus we can select the features which are correlated to each other and are independent in nature. These feature set were then given as an input to four algorithms and a hyper parameter tunning was done to the best model and the accuracy has been improved. Hence we calculated the performance of each model using different performance metrics and compared them based on these metrics. Then we have also saved the best model and predicted the label. It was good the predicted and actual values were almost same.

4.2 Limitations of this work and Scope for Future Work

- ✓ First draw back is the length of the dataset it is small because of scraping data.
- ✓ Data scrap from the different websites so it can be change at any time so also we have update model with new data at time.
- ✓ Some algorithms are facing over-fitting problem which may be because of less number of features in our dataset.