FLIGHT PRICE PREDICTION PROJECT

Submitted by:

NITI MARODIA

ACKNOWLEDGMENT

I want to express my great appreciation to my mentor for her valuable and constructive suggestions during the planning and development of this research work. Her willingness to give his time so generously has been very much appreciated.I would also like to thank the staff of the Data Trained for helping me with the problems I faced during the research work.Articles from the "Medium" platform werebeneficial during the whole process. It helped me clear my concepts.

INTRODUCTION

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 - 1. Time of purchase patterns (making sure last-minute purchases are expensive) 2. Keeping the flight as full as they want it (raising prices on a flight which is filling up in order to reduce sales and hold back inventory for those expensive last-minute expensive purchases) So, you have to work on a project where you collect data of flight fares with other features and work to make a model to predict fares of flights.

Objective:

This project contains two-phase-

Data Collection Phase

We scrapped more than 1500 rows of data. In this we scrapped s the data of flights from different websites (yatra.com, skyscanner.com, official websites of airlines, etc).. Generally, these columns are airline name, date of journey, source, destination, route, departure time, arrival time, duration, total stops and the target variable price.

Model Building Phase

After collecting the data, we built a machine learning model. Before model building, we did data pre-processing steps.

Followed the complete life cycle of data science. Include all the steps like.

1. Data Cleaning
2. Exploratory Data Analysis
3. Data Pre-processing
4. Model Building
5. Model Evaluation
6. Selecting the best model

Analytical Problem Framing

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 like Id. And few of them was removed due to very less correlation with dependent variable.To fix the outliers we used z score method. After this also there was a lot of skewness in dataset so power transform has been used. To check the accuracyr2 score was used also for cross validation cross_val _score is used.

DATA/ DATA PREPROCESSING:

- The dataset contains 1792 rows and 8 columns
- Fare is our dependent variable.
- We created new features from old ones.
- All columns were object data types we converted necessary ones into int and float.
- There are no null values in the dataset.
- Trimmed few columns

Hardware and Software Requirements and Tools Used

- HP 5- i5 8th generation, 8gb ram, NVidia mx130 integrated graphic,
- JuypterNotebook/Google chrome
- Libraries and packeges used:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
from sklearn.preprocessing import LabelEncoder
from sklearn.feature_selection import VarianceThreshold
from sklearn.feature_selection import mutual_info_regression
from sklearn.feature_selection import SelectPercentile
from sklearn.preprocessing import StandardScaler
from statsmodels.stats.outliers_influence import variance_inflation_factor
from sklearn.preprocessing import power_transform
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import ExtraTreesRegressor
from sklearn.svm import SVR
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import BaggingRegressor
from sklearn.ensemble import AdaBoostRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.model_selection import GridSearchCV
```

the library used here is sklearn,numpy,matplotlib,pandas and seaborn. The matpotplotlib and seaborn library has been used to make charts to visualize and understand the problem, correlation, outliers and many other things, the pandas and NumPy library is used to handle dataset and perform various tasks. The seaborn library is used for model building and cross validation of the models.

Model/s Development and Evaluation

The approach to solve this problem was to get the domain knowledge to understand the data better. Which values can be the part of the data and which is not? After exploring the data, it is found that though the data hasno missing value. It has extreme outliers and unrealistic value. We used Z-Score method to remove outliers. There was some skewness in the data, power transform method has been used so it dealt skewness. To check the accuracy, mean square error, mean absolute error, r2 score was used also for cross validation cross_val _score is used

Algoritham used for Traning and testing:

```
from sklearn.model_selection import train_test_split

from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

from sklearn.linear_model import LinearRegression

from sklearn.tree import DecisionTreeRegressor

from sklearn.neighbors import KNeighborsRegressor

from sklearn.ensemble import RandomForestRegressor

from sklearn.ensemble import ExtraTreesRegressor

from sklearn.svm import SVR

from sklearn.model_selection import cross_val_score

from sklearn.ensemble import BaggingRegressor

from sklearn.ensemble import AdaBoostRegressor

from sklearn.ensemble import GradientBoostingRegressor

from sklearn.model_selection import GridSearchCV
```

Performance of the model:

```
#spliting train test data
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.30, random_state=56)
x_train.shape,y_train.shape,x_test.shape,y_test.shape
```

```
((817, 32), (817,), (351, 32), (351,))
```

By the train test split method, 70 percent of the data has been taken for the model building while 30 percent of the data has been reserved for checking the model's performance.

```
#creating function
def model(name):
    model=name()
    model.fit(x_train,y_train)
    predict=model.predict(x_test)
    print("""mean squared error is:
    """,mean_squared_error(y_test, predict))

    print("The mean absolute error is: ", mean_absolute_error(y_test,predict))


    print("""r2 score is:

    """,r2_score(y_test,predict))

    print("cross_val_score", cross_val_score(model,x,y,cv=5).mean())
```

The code above has been used to speed up the model training and its evaluation process. Here the function name model ha been created which take the name of model as argument.

```
#LineraRegression
model(LinearRegression)
```

```
mean squared error is:
     3196099.2223078595
The mean absolute error is:  1433.8667941113072
r2 score is:

     0.25193402513999397
cross_val_score 0.01665777802732844
```

```
#decisiontreeregressor
model(DecisionTreeRegressor)
```

```
mean squared error is:
     2363036.6491942084
The mean absolute error is:  839.0975448536356
r2 score is:

     0.44691726018038
cross_val_score -0.10927674293391207
```

```
#randomforestregressor
model(RandomForestRegressor)
```

```
mean squared error is:
     1783233.990097756
The mean absolute error is:  826.5791080709895
r2 score is:

     0.5826235105921618
cross_val_score 0.18439391680321432
```

```
#extratreesreressor
model(ExtraTreesRegressor)
```

```
mean squared error is:
     1850607.3519434037
The mean absolute error is:  735.0998276676108
r2 score is:

     0.5668543757490115
cross_val_score 0.26765398963420817
```

```
model(KNeighborsRegressor)
```

```
mean squared error is:
     2035756.295750708
The mean absolute error is:  991.4515580736545
r2 score is:

     0.5235191675749986
cross_val_score 0.18641650875801247
```

```
from sklearn.ensemble import BaggingRegressor
from sklearn.ensemble import AdaBoostRegressor
from sklearn.ensemble import GradientBoostingRegressor
```

```
#BaggingRegressor
model(BaggingRegressor)
```

```
mean squared error is:
     1902014.1508659858
The mean absolute error is:  853.4602678954662
r2 score is:

     0.5548223096347795
cross_val_score 0.1942141606833171
```

```
#AdaBoostRegressor
model(AdaBoostRegressor)
```

```
mean squared error is:
     2740863.4334033416
The mean absolute error is:  1371.8428089522315
r2 score is:

     0.3584846609403801
cross_val_score 0.13555571274459
```

```
#GradientBoostingRegressor
model(GradientBoostingRegressor)
```

```
mean squared error is:
     1808658.1876762426
The mean absolute error is:  974.2160123596385
r2 score is:

     0.5766728263912969
cross_val_score 0.22778851239033598
```

```
#RandomForestregressor is the best model as the RMSLE is maxium
#setting parameters for hyperparameter tunning
parameter={
          "criterion":["mse","mae"],
          'max_features':['auto', 'sqrt',"log2"],
          'min_samples_split':[2, 5, 10, 15],
           'min_samples_leaf':[1, 2, 5, 10]}
```

```
#using GridSearchCV for Hyper parameter tunning
from sklearn.model_selection import GridSearchCV
gcv=GridSearchCV(RandomForestRegressor(),parameter,cv=5)
```

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

```
GridSearchCV(cv=5, estimator=RandomForestRegressor(),
             param_grid={'criterion': ['mse', 'mae'],
                         'max_features': ['auto', 'sqrt', 'log2'],
                         'min_samples_leaf': [1, 2, 5, 10],
                         'min_samples_split': [2, 5, 10, 15]})
```

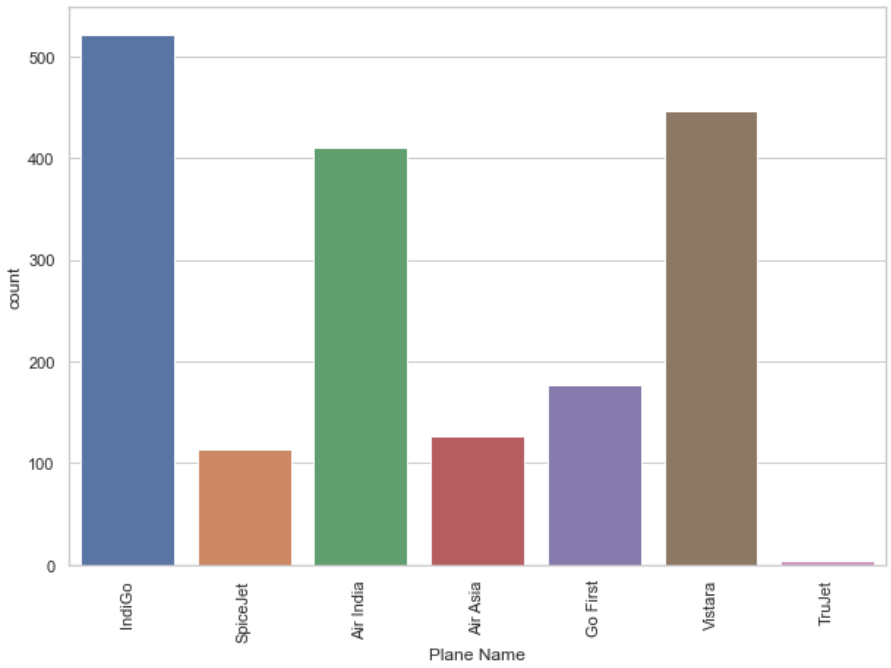
```
#checking best parameters
gcv.best_params_
```

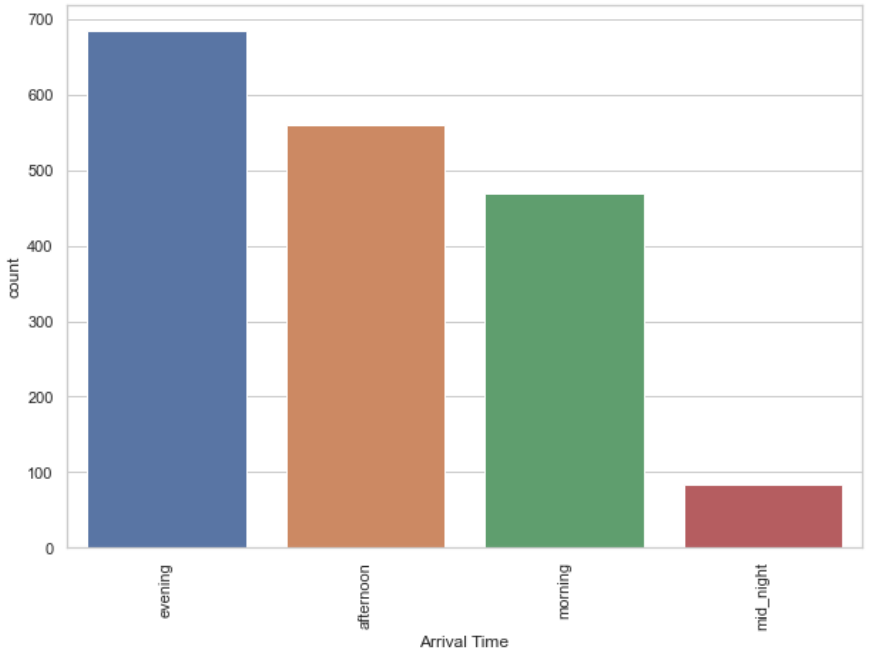
```
{'criterion': 'mae',
 'max_features': 'sqrt',
 'min_samples_leaf': 1,
 'min_samples_split': 2}
```

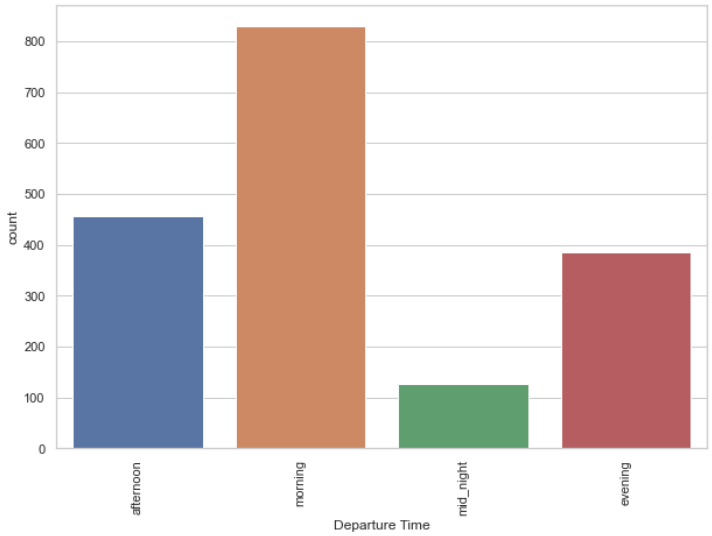
```
model=RandomForestRegressor(criterion="mse",max_features="auto",min_samples_leaf=1,min_samples_split=10)
model.fit(x_train,y_train)
pred=model.predict(x_test)

print("""mean squared error is:
    """,mean_squared_error(y_test, pred))

print("The mean absolute error is: ", mean_absolute_error(y_test,pred))


print("r2 score is:" ,r2_score(y_test,pred))
```

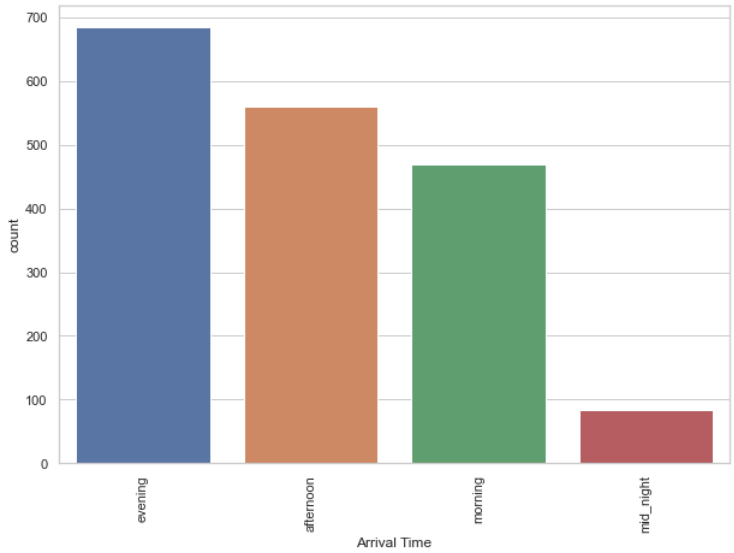
```
mean squared error is:
     1703762.375987915
The mean absolute error is:  881.7675187151985
r2 score is: 0.6012243131166367
```

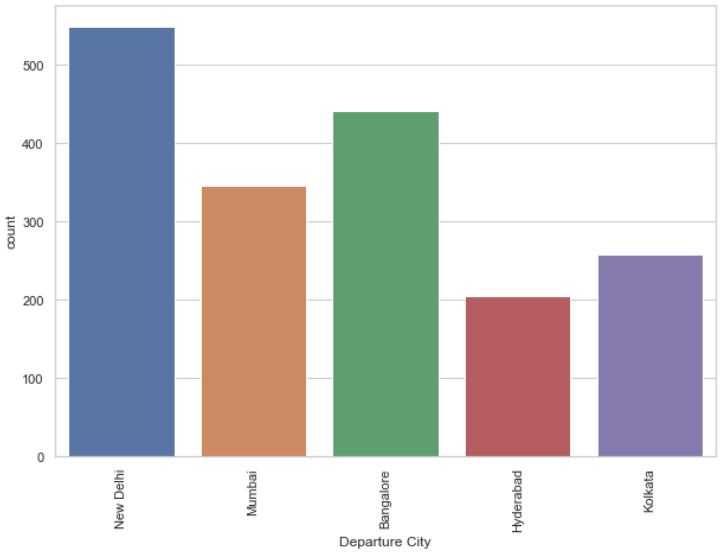
Visualization:

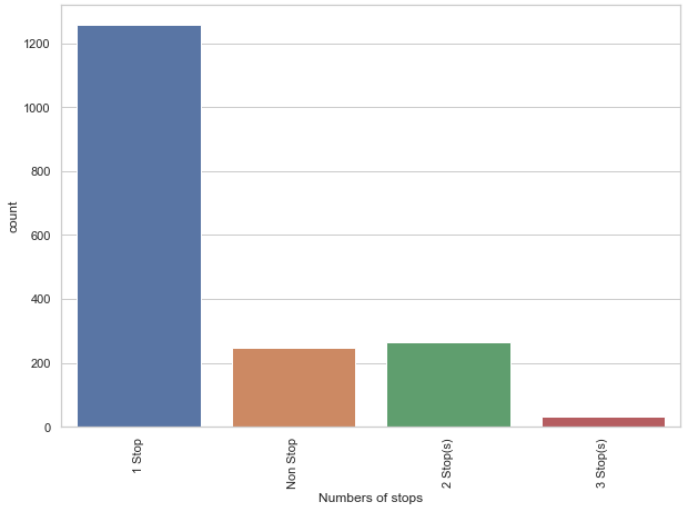


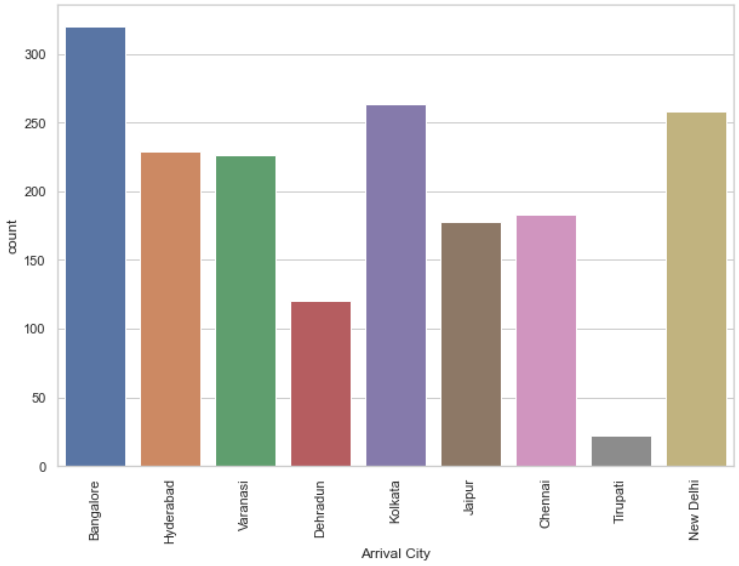


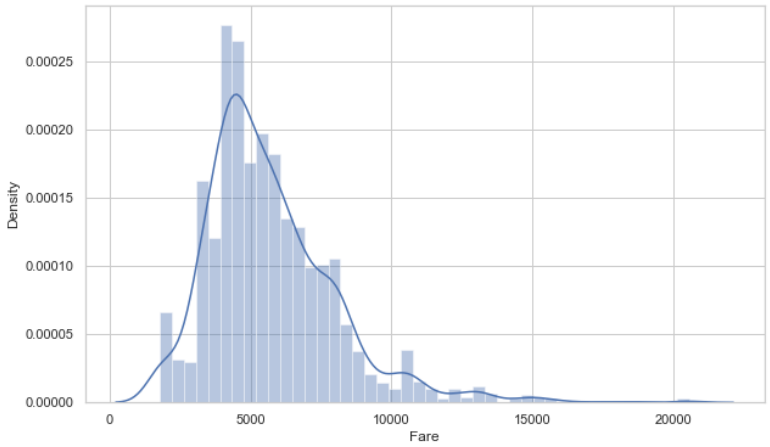

count
800
700
600
500
400
300
200
100
0
afternoon
morning
mid_night
evening
Departure Time

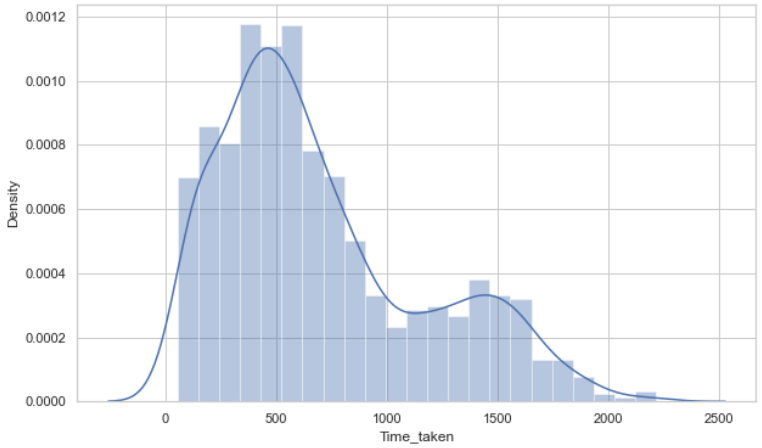

count
700
600
500
400
300
200
100
0
evening
afternoon
morning
mid_night
Arrival Time

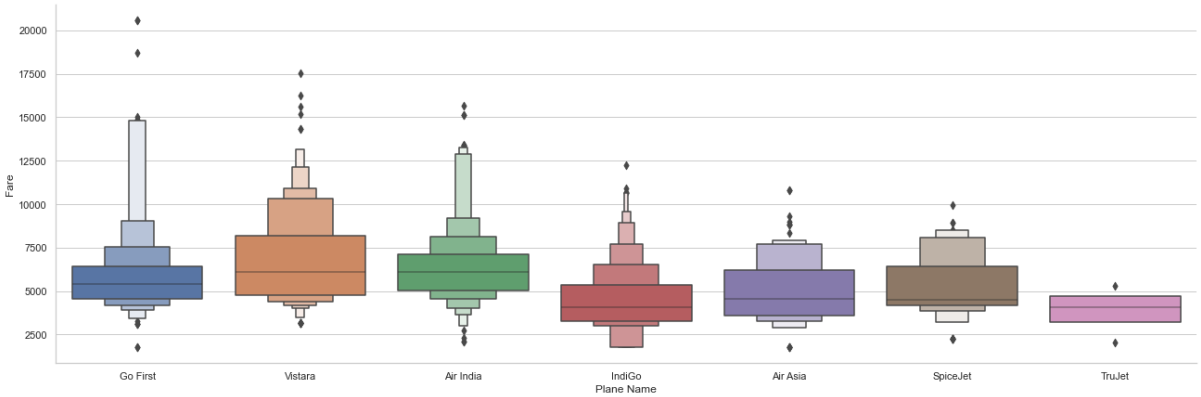

count
500
400
300
200
100
0
New Delhi
Mumbai
Bangalore
Hyderabad
Kolkata
Departure City

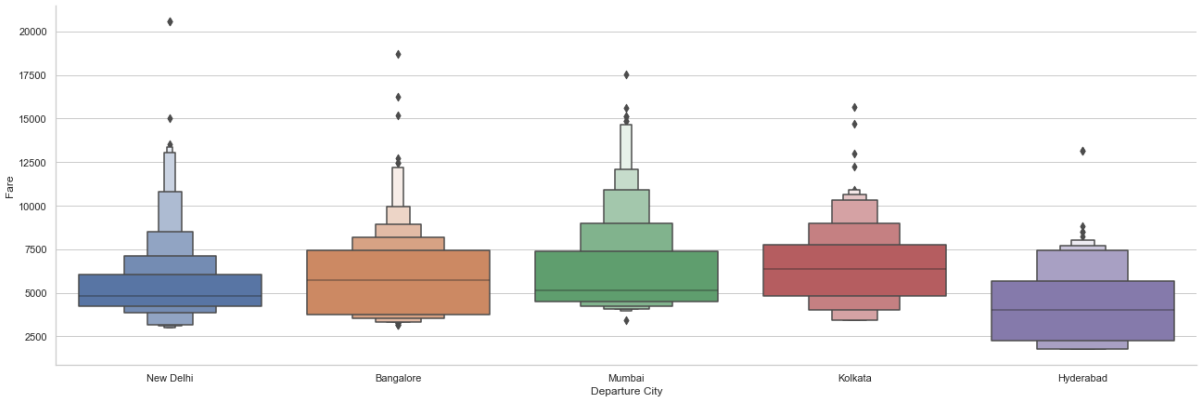

1200
1000
800
600
400
200
0
count
1 Stop
Non Stop
2 Stop(s)
3 Stop(s)
Numbers of stops

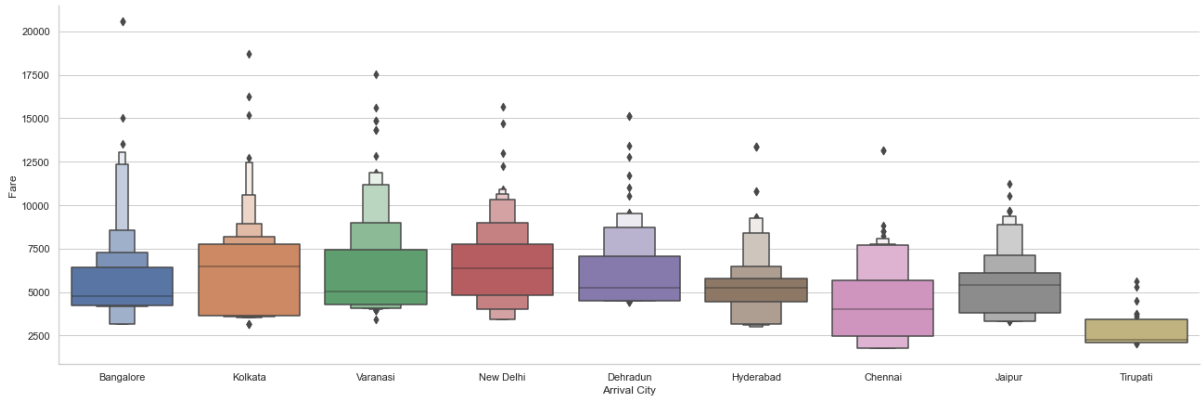

300
250
200
150
100
50
0
count
Bangalore
Hyderabad
Varanasi
Dehradun
Kolkata
Jaipur
Chennai
Tirupati
New Delhi
Arrival City

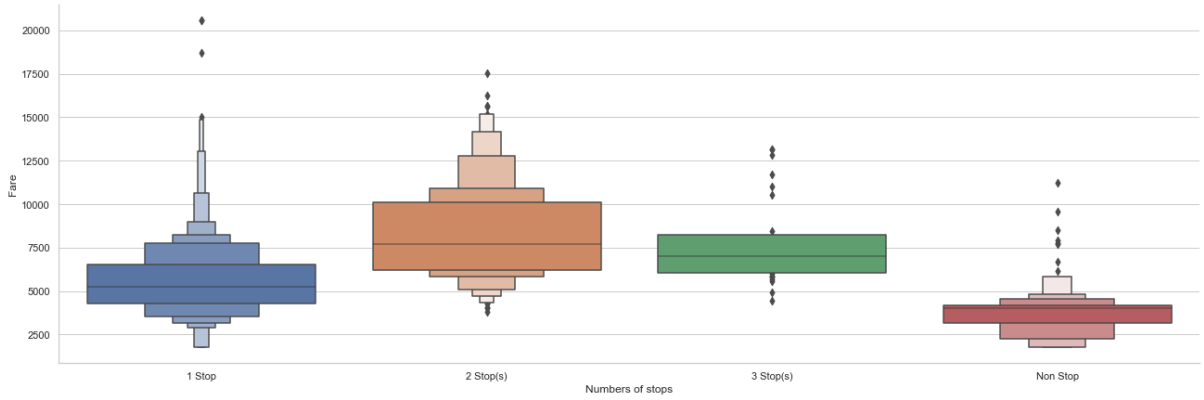

0.00025
0.00020
0.00015
0.00010
0.00005
0.00000
Density
0
5000
10000
15000
20000
Fare

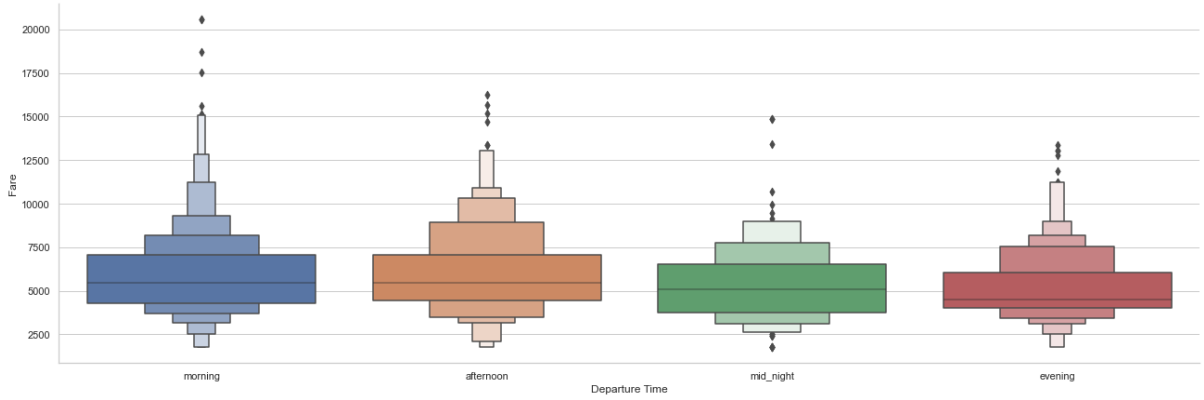

0.0012
0.0010
0.0008
0.0006
0.0004
0.0002
0.0000
Density
0
500
1000
1500
2000
2500
Time_taken

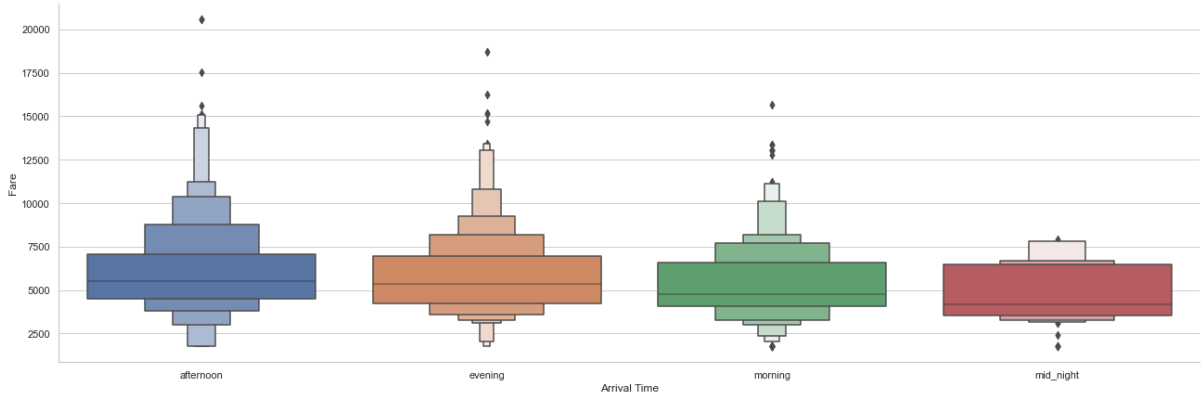

20000
17500
15000
12500
10000
7500
5000
2500
Fare
Go First
Vistara
Air India
IndiGo
Air Asia
SpiceJet
TruJet
Plane Name

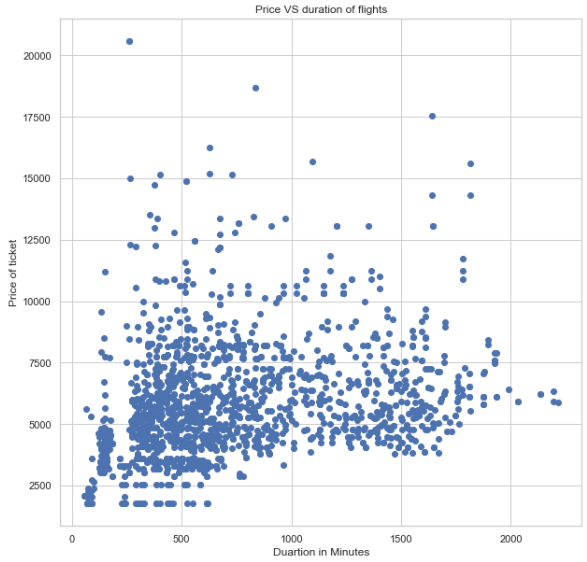

20000
17500
15000
12500
10000
7500
5000
2500
Fare
New Delhi
Bangalore
Mumbai
Kolkata
Hyderabad
Departure City

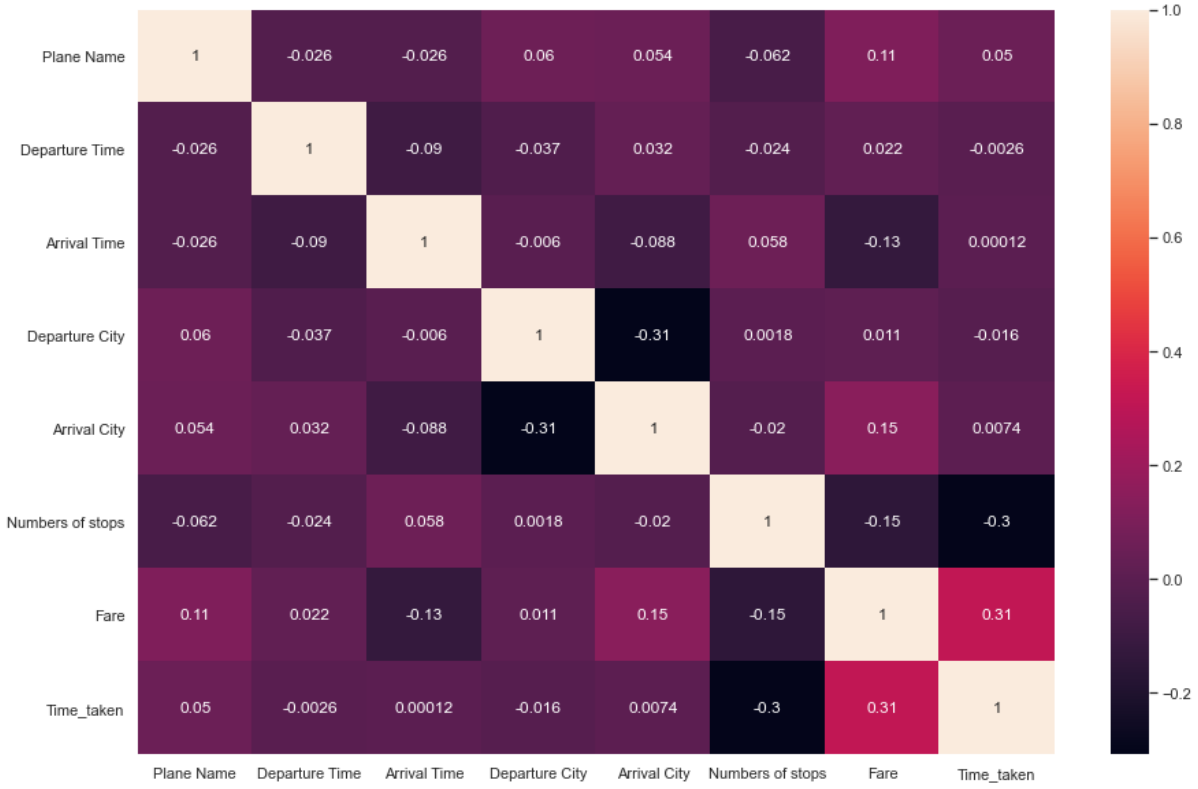

Fare
20000
17500
15000
12500
10000
7500
5000
2500
Bangalore
Kolkata
Varanasi
New Delhi
Dehradun
Hyderabad
Chennai
Jaipur
Tirupati
Arrival City


Fare
20000
17500
15000
12500
10000
7500
5000
2500
1 Stop
2 Stop(s)
3 Stop(s)
Non Stop
Numbers of stops


Fare
20000
17500
15000
12500
10000
7500
5000
2500
morning
afternoon
mid_night
evening
Departure Time


Fare
20000
17500
15000
12500
10000
7500
5000
2500
afternoon
evening
morning
mid_night
Arrival Time


Price VS duration of flights
Price of ticket
Duartion in Minutes
20000
17500
15000
12500
10000
7500
5000
2500
0
500
1000
1500
2000


Plane Name 1 -0.026 -0.026 0.06 0.054 -0.062 0.11 0.05
Departure Time -0.026 1 -0.09 -0.037 0.032 -0.024 0.022 -0.0026
Arrival Time -0.026 -0.09 1 -0.006 -0.088 0.058 -0.13 0.00012
Departure City 0.06 -0.037 -0.006 1 -0.31 0.0018 0.011 -0.016
Arrival City 0.054 0.032 -0.088 -0.31 1 -0.02 0.15 0.0074
Numbers of stops -0.062 -0.024 0.058 0.0018 -0.02 1 -0.15 -0.3
Fare 0.11 0.022 -0.13 0.011 0.15 -0.15 1 0.31
Time_taken 0.05 -0.0026 0.00012 -0.016 0.0074 -0.3 0.31 1
Plane Name
Departure Time
Arrival Time
Departure City
Arrival City
Numbers of stops
Fare
Time_taken
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

Observations:

#indigo provides the greatest number of services while Trujet provides least number of services.

#most flights depart in the morning

#most flight arrives in the evening

#from the collected data New Delhi is the has the most departed flight

#from the collected data New Delhi is the has the most arrived flight

#majority of flights have one stops

#the prices lie b/w 5k-7k. also there are few flights with higher price

#majority of the flights takes average 500-800 minute

#go fist has the highest fare and the lowest also

the flights that departs from Delhi has more fare than others

the flights that arrive in Bangalore has more fare than others

#the flight with 1 stop is more costly

#fights that departs in the morning costs more

##the flights that lands in afternoon has more fair

#fare shows a liner relationship with time

#fare is highly correlated with time taken

#there are some outliers in Time taken

CONCLUSION

This paper showed the model training process for the prediction of the fare Price. One of the objectives of the paper was to check the important variable for the prediction of the price and how these variables describe the price. Through model training and evaluating its performance. RandomForest proved to be as best model. As the difference between the r2score and cross validation score was minimum. This project has increased my understanding of the concept. During the research I came across various challenges and while solving them I learned a lot of new things. For example. How to plot different charts. For example, I learned how to plot subplot. I learned new libraries and how to use them. I explored various methods for feature selection. Also, I came to understand how can multicollinearity can cause problem during the model training.The limitation of the solution provided is that the data carried a lot of unrealistic values. Apart from that my laptop took to much time while running certain command where I lost a lot of precious time.