FLIP ROBO

FLIGHT PRICE PREDICTION PROJECT

Submitted by:

Amrutha Dhondale

ACKNOWLEDGMENT

I have taken efforts in this project however it would not have been completed without guidance from other people. I warmly acknowledge the invaluable supervision and an inspired guidance by our SME Ms.Swati Mahaseth, FlipRobo Technology.

I would also like to express my sincere thanks to Data trained Education and FlipRobo Technology for giving me an opportunity to work on this project. I also want to express my gratitude towards my friends and family who have patiently extended all sorts of help for accomplishing this. I am grateful to one and all who are directly or indirectly involved in successful completion of this project.

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)

The project consists of two phase:

1. Data Collecting Phase
2. Model Building Phase

In first phase we have to collect data of flights ticket from online websites. Here data is collected from “www.yatra.com” website using Selenium technique for web scraping. We have fetched data of popular flights available on website and a few flights from metropolitan cities.

Next, we have built a regression model to predict flight ticket prices. In the long term, this would allow people to better explain and review their purchase with each other in this increasingly digital world.

ANALYTICAL FRAMING

In our scrapped dataset, our target variable "Price" is a continuous variable. Therefore, we will be handling this modeling problem as classification.

This project is done in three parts:

- Data Collection phase
- Data Analysis Phase
- Model Building Phase

1. Data Collection Phase
   You have to scrape at least 1500 rows of data. You can scrape more data as well, it’s up to you, More the data better the model In this section you have to scrape the data of flights from different websites (yatra.com, skyscanner.com, official websites of airlines, etc). The number of columns for data doesn’t have limit, it’s up to you and your creativity. Generally, these columns are airline name, date of journey, source, destination, route, departure time, arrival time, duration, total stops and the target variable price. You can make changes to it, you can add or you can remove some columns, it completely depends on the website from which you are fetching the data.

2. Data Analysis Phase After cleaning the data, you have to do some analysis on the data. Do airfares change frequently? Do they move in small increments or in large jumps? Do they tend to go up or down over time? What is the best time to buy so that the consumer can save the most by taking the least risk? Does price increase as we

get near to departure date? Is Indigo cheaper than Jet Airways? Are morning flights expensive?

3. Model Building Phase After collecting the data, you need to build a machine learning model. Before model building do all data pre-processing steps. Try different models with different hyper parameters and select the best model. Follow 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

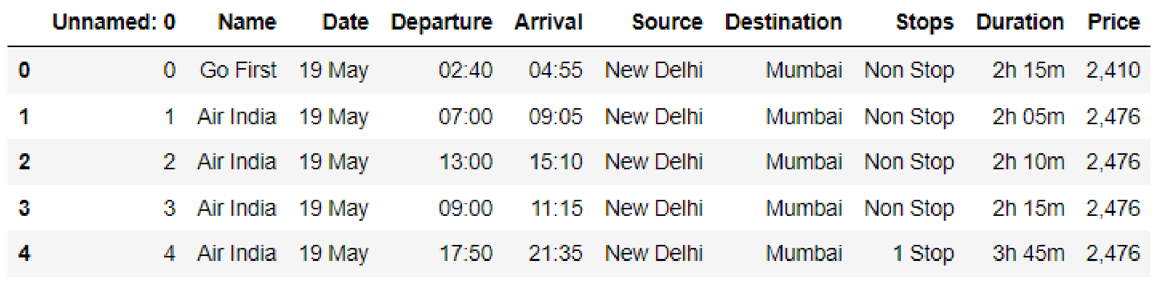
6. Selecting the best model

DATA SOURCES AND THEIR FORMATS

1. We collected the data from "yatra.com". The data is scrapped using Web scraping technique and the framework used is Selenium.
2. We scrapped 868 of the data and saved data in a data frame

3. In the end, we saved all the data in csv file as "data.csv"

4. The data fetched looks like what is shown below:



	Unnamed: 0	Name	Date	Departure	Arrival	Source	Destination	Stops	Duration	Price
0	0	Go First	19 May	02:40	04:55	New Delhi	Mumbai	Non Stop	2h 15m	2,410
1	1	Air India	19 May	07:00	09:05	New Delhi	Mumbai	Non Stop	2h 05m	2,476
2	2	Air India	19 May	13:00	15:10	New Delhi	Mumbai	Non Stop	2h 10m	2,476
3	3	Air India	19 May	09:00	11:15	New Delhi	Mumbai	Non Stop	2h 15m	2,476
4	4	Air India	19 May	17:50	21:35	New Delhi	Mumbai	1 Stop	3h 45m	2,476

Features:

Name: name of Airline

Date: date of journey

Departure: time of departure

Arrival: time of arrival

Source: the source from which service begins

Destination: the destination where service ends

Stops: total number of stops between source and destination

Duration: total duration of flight

Price: Price of flight ticket

The dataset has no null values.

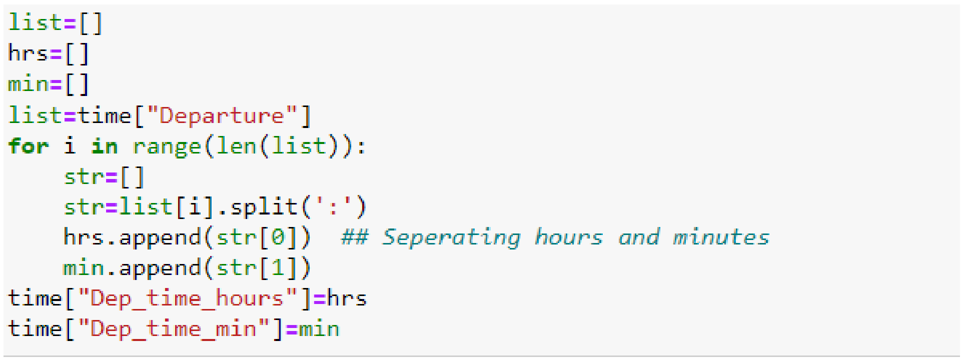
DATA PRE-PROCESSING

- First, we will clean price column by removing ',' and changing its data type to 'int

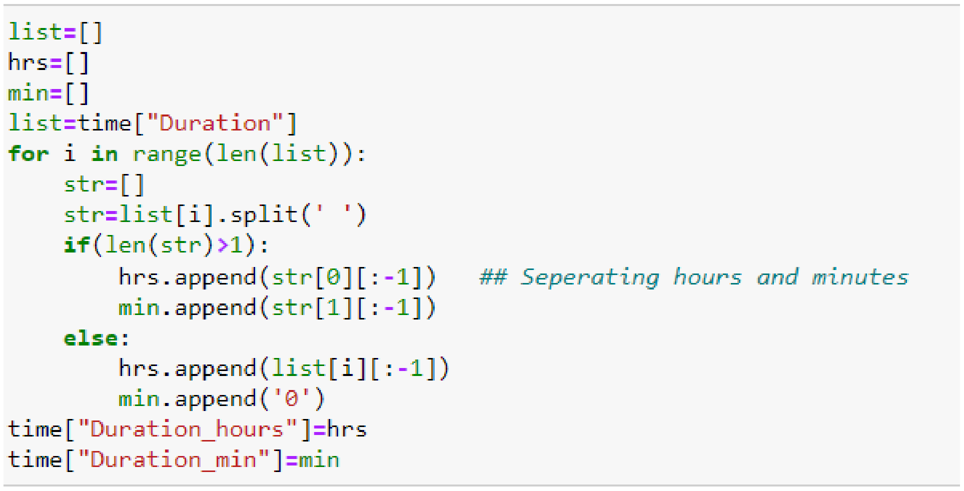
```
# Changing data type of price
p=[]
price=df["Price"]
for i in range(len(price)):
    st=price[i]
    p.append(st.replace(",",""))
df["Price"]=p

df_type_dict={'Price':int}
df=df.astype(df_type_dict)
df.dtypes
```

```
Unnamed: 0      int64
Name           object
Date           object
Departure      object
Arrival        object
Source         object
Destination    object
Stops          object
Duration       object
Price           int32
dtype: object
```

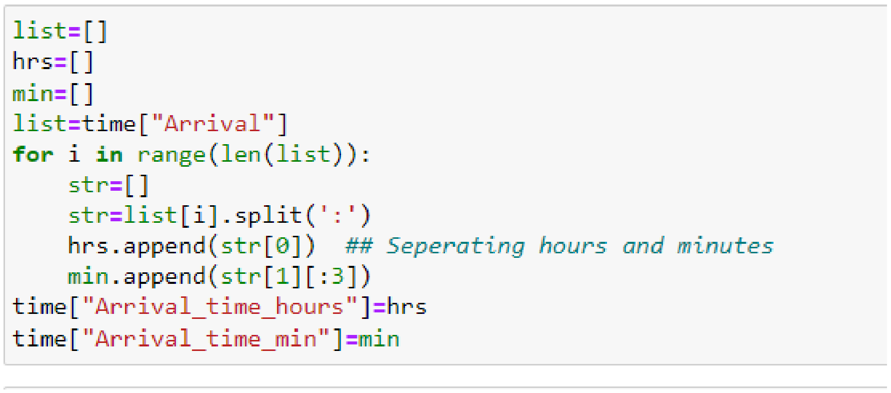
• Next, we have removed unnecessary columns and cleaned data in “Arrival”, “Departure”, “Duration” and derived new features from each given feature. I have first formed a new data frame and then done all the processing. The following code is used to clean data and deriving new features from it:



```
list=[]
hrs=[]
min=[]
list=time["Departure"]
for i in range(len(list)):
    str=[]
    str=list[i].split(':')
    hrs.append(str[0])  ## Seperating hours and minutes
    min.append(str[1])
time["Dep_time_hours"]=hrs
time["Dep_time_min"]=min
```



```
list=[]
hrs=[]
min=[]
list=time["Duration"]
for i in range(len(list)):
    str=[]
    str=list[i].split(' ')
    if(len(str)>1):
        hrs.append(str[0][:-1])   ## Seperating hours and minutes
        min.append(str[1][:-1])
    else:
        hrs.append(list[i][:-1])
        min.append('0')
time["Duration_hours"]=hrs
time["Duration_min"]=min
```



```
list=[]
hrs=[]
min=[]
list=time["Arrival"]
for i in range(len(list)):
    str=[]
    str=list[i].split(':')
    hrs.append(str[0])  ## Seperating hours and minutes
    min.append(str[1][:3])
time["Arrival_time_hours"]=hrs
time["Arrival_time_min"]=min
```

• After executing the above lines of code we will new columns Dep_time_hours, Dep_time_min, Duration_hours, Duration_min, Arrival_time_hours, Arrival_time_min. Each feature now has integer data type. Since all the usefull information is now extracted we can drop previous columns.

	Departure	Arrival	Duration	Date	Dep_time_hours	Dep_time_min	Duration_hours	Duration_min	Arrival_time_hours	Arrival_time_min
0	02:40	04:55	2h 15m	19 May	02	40	2	15	04	55
1	07:00	09:05	2h 05m	19 May	07	00	2	05	09	05
2	13:00	15:10	2h 10m	19 May	13	00	2	10	15	10
3	09:00	11:15	2h 15m	19 May	09	00	2	15	11	15
4	17:50	21:35	3h 45m	19 May	17	50	3	45	21	35

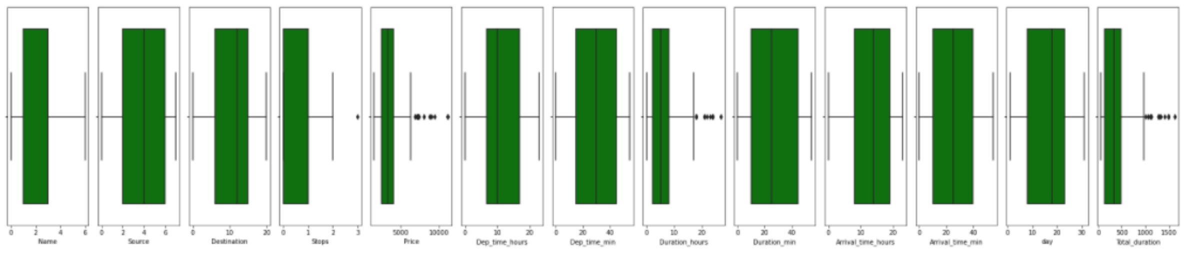
Encoding variables with object data type: We have encoded "Stops" manually and used LabelEncoder for other variables.

```
data["Stops"]=data["Stops"].replace({'Non Stop':0,'1 Stop':1,'2 Stop(s)':2,'3 Stop(s)':3,'4 Stop(s)':4})
data.head()
```

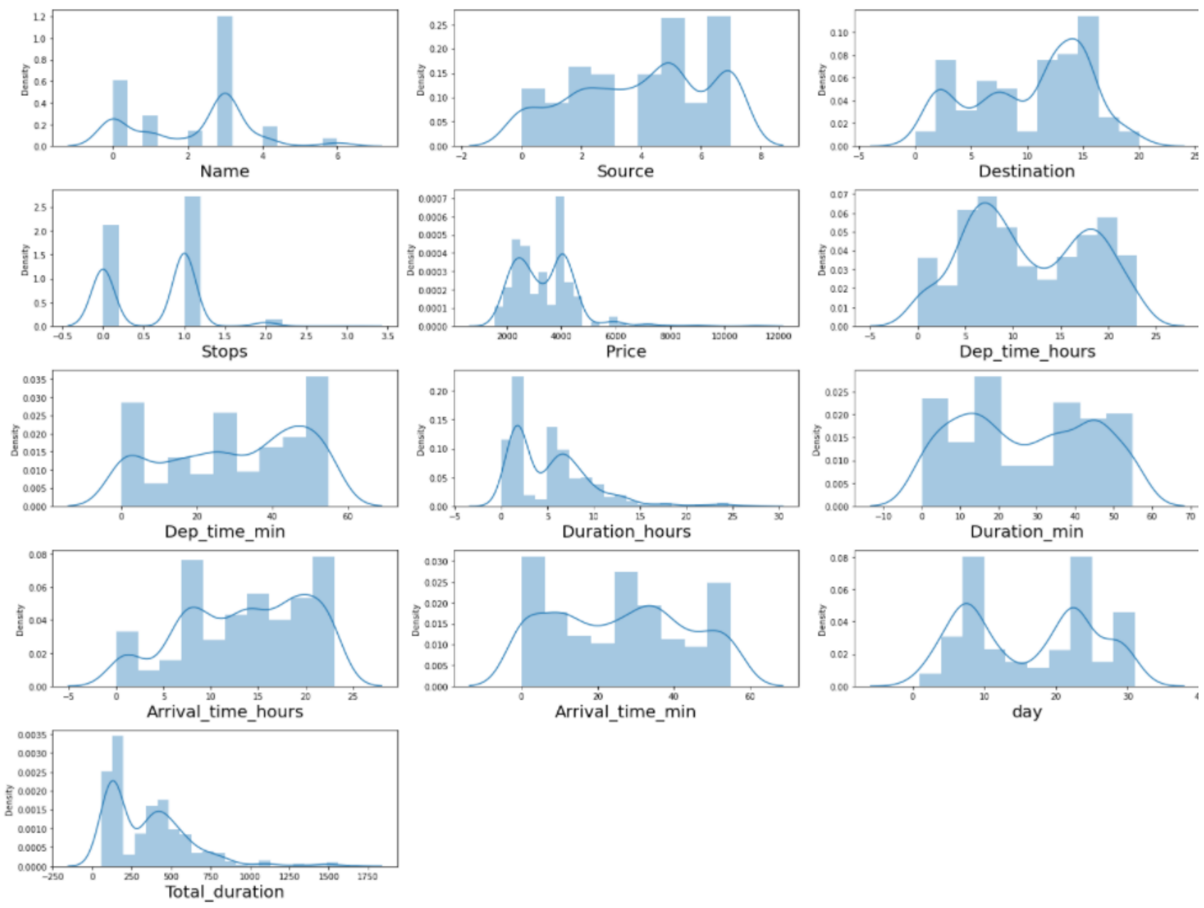
```
lab_enc=LabelEncoder()
cols=["Name", "Source", "Destination"]
for i in cols:
    df1= lab_enc.fit_transform(data[i])
    data[i]=df1
data.head()
```

CHECKING OUTLIERS

```
collist=data.columns.values
plt.figure(figsize=(40,80))
for i in range(0,len(collist)):
    plt.subplot(15,20,i+1)
    sns.boxplot(data[collist[i]],color='green',orient='v')
    plt.tight_layout()
```



CHECKING SKEWNESS AND DATA DISTRIBUTION



To handle outliers and skewness we have used z-score method and log transformation by which we faced a data loss of 2.99%.

HARDWARE AND SOFTWARE REQUIREMENTS AND TOOLS USED

For doing this project, the hardware used is a laptop with a stable internet connection.

While coming to software part, I have used Jupyter notebook to do my python programming and analysis.

We also need Google chrome webdriver to scrap data.

Libraries Used:

- scikit-learn
- matplotlib
- pandas
- numpy
- Selenium

PREPARING DATA FOR MODEL

Making our Data ready for model Building phase we will first separate target variable from other features. Then use StandardScaler to scale data and use train_test_split to split data into train and test to make it ready for model. We found out the best random state for our linear regression model and then run each model on this random state.

MODEL BUILDING AND EVALUATION

Algorithms used are:

♣ Linear Regression

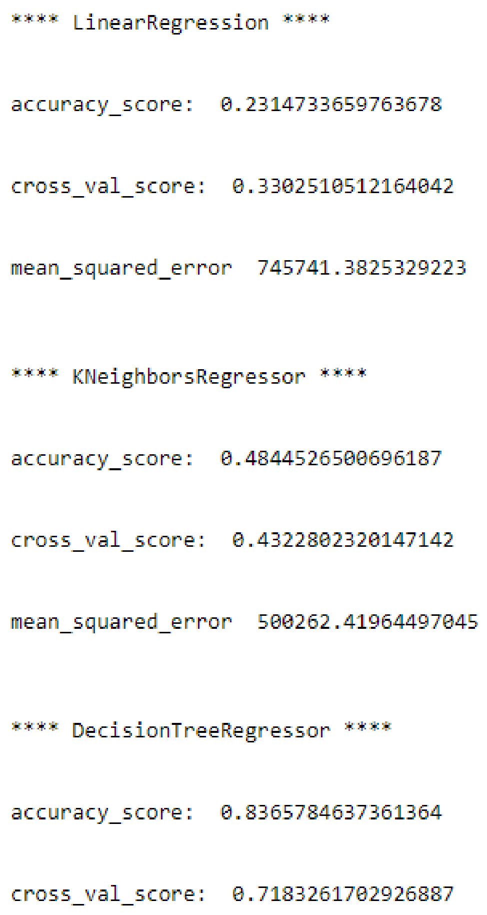
♣ Decision Tree Regressor

♣ KNN Regressor

♣ Random Forest Regressor

♣ Gradient Boosting Regressor

Since range of target variable is too high, we are getting a high value of mse therefore we will look for R2 score and CV Score to determine our best model.

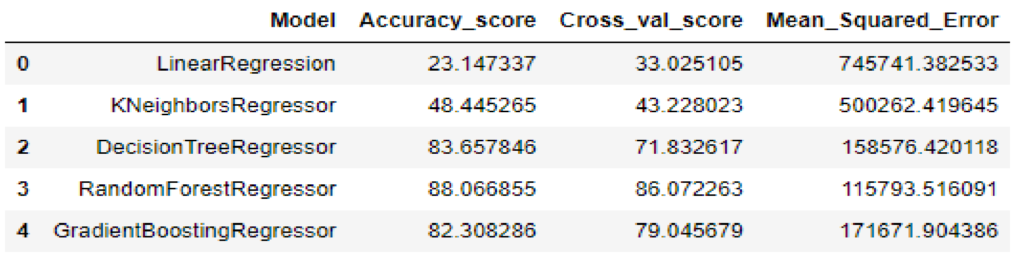


```
**** LinearRegression ****

accuracy_score:  0.2314733659763678

cross_val_score:  0.3302510512164042

mean_squared_error  745741.3825329223


**** KNeighborsRegressor ****

accuracy_score:  0.4844526500696187

cross_val_score:  0.4322802320147142

mean_squared_error  500262.41964497045


**** DecisionTreeRegressor ****

accuracy_score:  0.8365784637361364

cross_val_score:  0.7183261702926887
```

```
**** RandomForestRegressor ****

accuracy_score:  0.8806685491147601

cross_val_score:  0.8607226265370864

mean_squared_error  115793.51609053258


**** GradientBoostingRegressor ****

accuracy_score:  0.8230828623379685

cross_val_score:  0.7904567859562321

mean_squared_error  171671.90438554637
```

Choosing Best Model

After running the loop, we get a dataframe showing each model and scores obtained from it.



	Model	Accuracy_score	Cross_val_score	Mean_Squared_Error
0	LinearRegression	23.147337	33.025105	745741.382533
1	KNeighborsRegressor	48.445265	43.228023	500262.419645
2	DecisionTreeRegressor	83.657846	71.832617	158576.420118
3	RandomForestRegressor	88.066855	86.072263	115793.516091
4	GradientBoostingRegressor	82.308286	79.045679	171671.904386

Looking the various metrics, we conclude “Random Forest Model” as our best model and hence we will now tune our model.

Hyper-parametric tuning

```
rmf= RandomForestRegressor()
params={'max_features':['auto','sqrt'],'n_estimators':[50,80], 'criterion':['mse','mae'],
        'max_depth':[5,10], 'min_samples_split':[4,6],
        'min_samples_leaf':[2,3]}
grd=GridSearchCV(rmf,param_grid=params)
grd.fit(x_train,y_train)
print('best params=>',grd.best_params_)
```

```
best params=> {'criterion': 'mse', 'max_depth': 10, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_e
stimators': 80}
```

```
rmf= RandomForestRegressor(criterion= 'mse', max_depth= 10, max_features= 'auto', min_samples_leaf= 2,
                           min_samples_split= 4, n_estimators= 80)
rmf.fit(x_train,y_train)
y_pred=rmf.predict(x_test)
print("Random Forest Regression: Accuracy = ",rmf.score(x_test,y_test))
print("\n Mean Squared Error= ",mean_squared_error(y_test,y_pred))
print("\n Root Mean Squared Error= ",np.sqrt(mean_squared_error(y_test,y_pred)))
print("\n Mean Absolute Error= ",mean_absolute_error(y_test,y_pred))
```

```
Random Forest Regression: Accuracy =  0.8697525647018703

 Mean Squared Error=  126385.86376904644

 Root Mean Squared Error=  355.5078955087305

 Mean Absolute Error=  214.09028697201683
```

OUR FINAL MODEL GIVES AN ACCURACY OF 86.97%

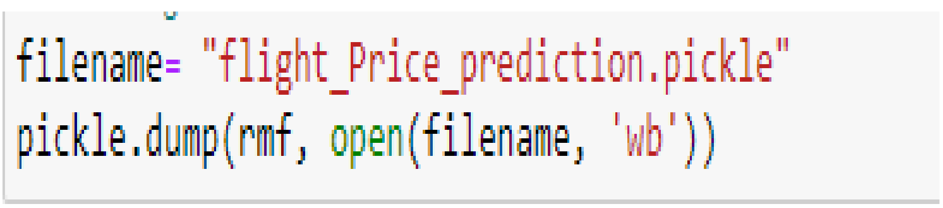
Comparing original and predicted price of the model

```
predictions=pd.DataFrame({'Original_price':y_test, 'Predicted_price':y_pred})
predictions
```

	Original_price	Predicted_price
492	3381	3425.181960
213	2000	2081.528370
755	4001	3979.120372
866	4077	4069.829291
609	4077	4076.429291
...	...	...
537	2966	3294.097594
179	3402	3251.891818
531	2966	3551.788765
573	2798	2806.364150
578	2271	2299.441426

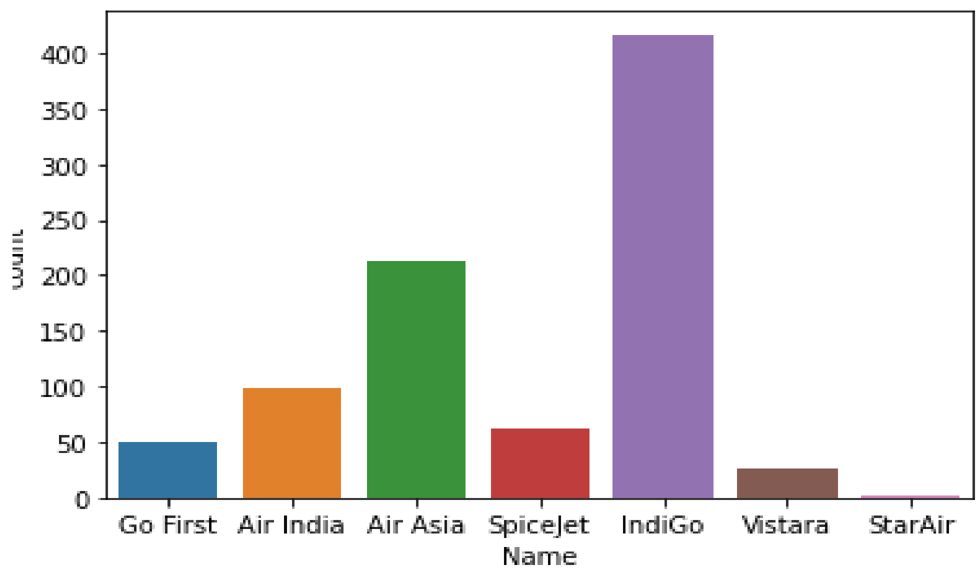
169 rows × 2 columns

Saving the Model

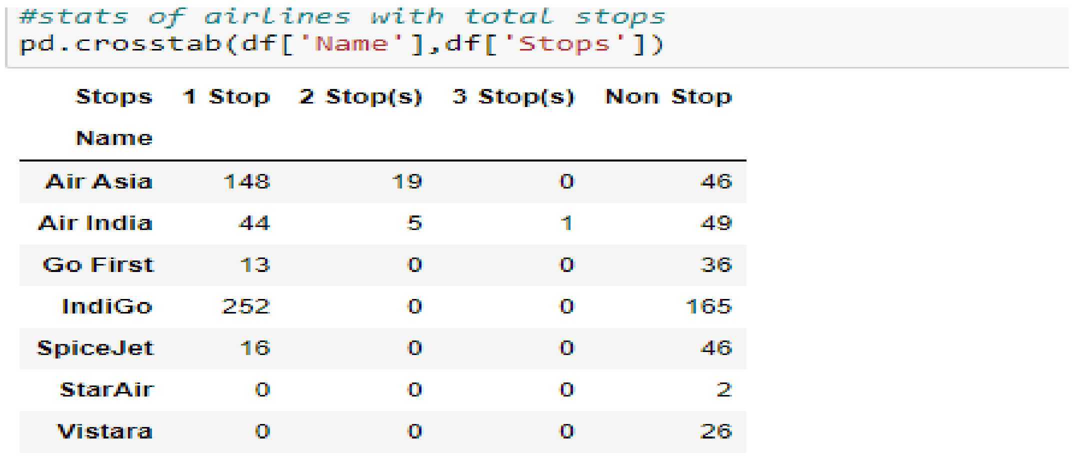


```
filename= "flight_Price_prediction.pickle"
pickle.dump(rmf, open(filename, 'wb'))
```

EXPLORATORY DATA ANALYSIS



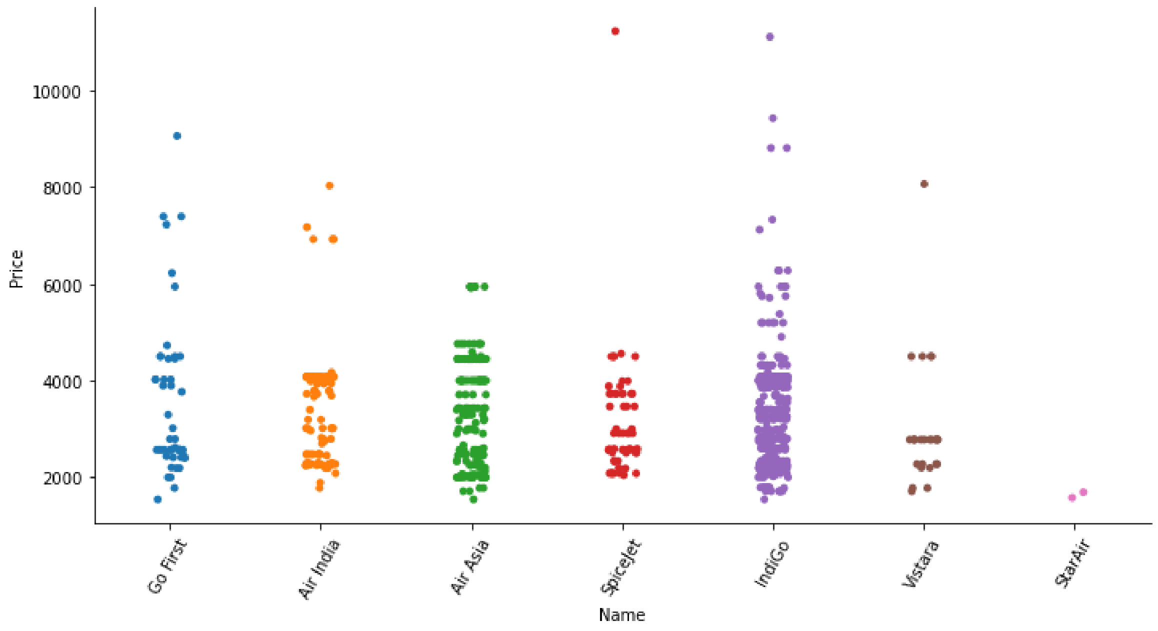
Matrix showing airlines and flights with number of stops



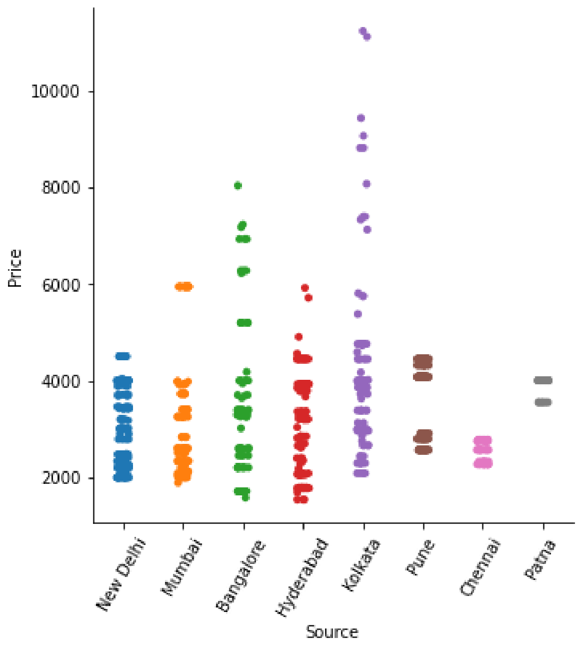
```
#stats of airlines with total stops
pd.crosstab(df['Name'],df['Stops'])
```

Stops	1 Stop	2 Stop(s)	3 Stop(s)	Non Stop
Name				
Air Asia	148	19	0	46
Air India	44	5	1	49
Go First	13	0	0	36
IndiGo	252	0	0	165
SpiceJet	16	0	0	46
StarAir	0	0	0	2
Vistara	0	0	0	26

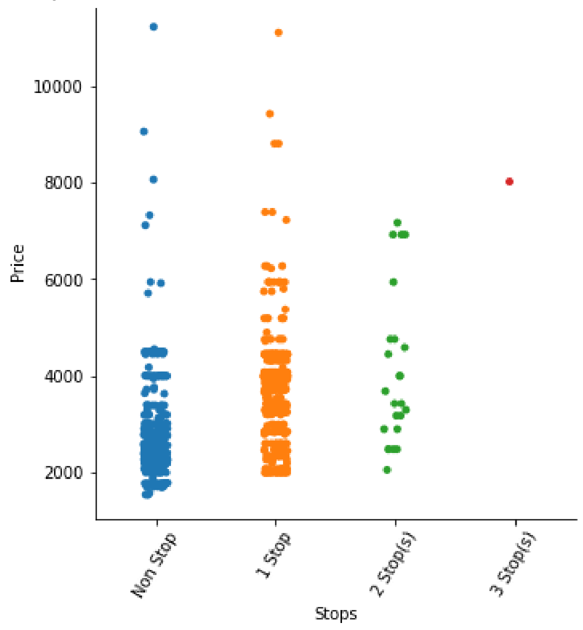
Name v/s Price



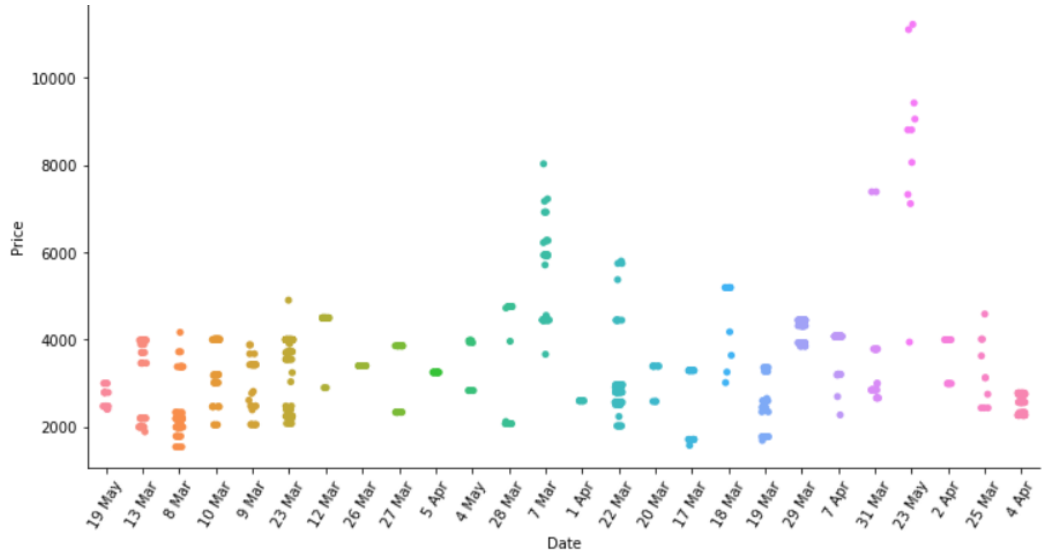
Source v/s Price



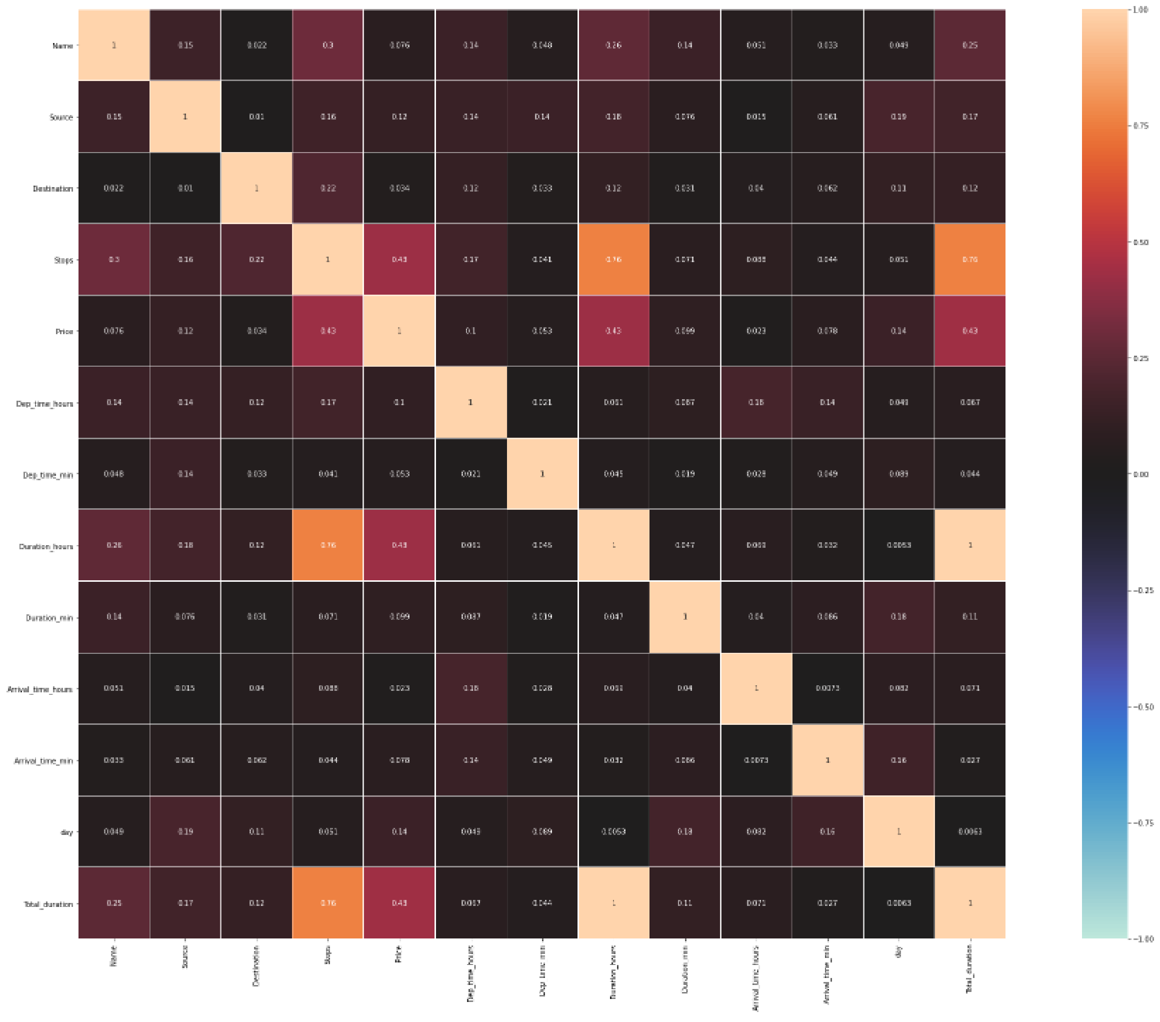
Stops v/s Price



Date v/s Price



Correlation matrix and heatmap



CONCLUSIONS

KEY FINDINGS AND CONCLUSIONS OF THE STUDY

First, we collected flight data from “yatra.com” it was done by using Web scraping. The framework used for web scraping was Selenium, which has an advantage of automating our process of collecting data. Then the scrapped data was saved in a csv file to use it for modelling purpose.

From the extensive EDA performed in this project we observe Flights from Kolkata have higher prices. Flights with longer route i.e., high number of stops have high prices. Also, prices of flight in May month are high as compared to those in coming months. From the given data we can also conclude that Go First and Air Asia flights are expensive as compared to other flights.

The model build after hyper-parametric tuning gives an accuracy for 86.97%.

LEARNING OUTCOMES OF THE STUDY IN RESPECT OF DATA SCIENCE

After the completion of this project, we got an insight of how to collect data, pre-processing the data, analysing the data and building a model. It helped me to gain conclusions from graphs. Also, it helped me in exploring multiple algorithms and metrics to get the best output.

LIMITATIONS OF THIS WORK AND SCOPE FOR FUTURE WORK

Since the data keeps changing, we cannot fully rely on this project in the distant future we need to update it with updating in data. Also, the scrapping of data took a lot of time as there was no such detail mentioned on fetching data.
Random sources and destinations are used to pick up data. This project is done with limited resources and can be made more efficient in future.

THANKYOU

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  Describe the mathematical, statistical and analytics modelling done during this project along with the proper justification.

- Data Sources and their formats

  What are the data sources, their origins, their formats and other details that you find necessary? They can be described here. Provide a proper data description. You can also add a snapshot of the data.

- Data Preprocessing Done

  What were the steps followed for the cleaning of the data? What were the assumptions done and what were the next actions steps over that?

- Data Inputs- Logic- Output Relationships

  Describe the relationship behind the data input, its format, the logic in between and the output. Describe how the input affects the output.

- State the set of assumptions (if any) related to the problem under consideration

  Here, you can describe any presumptions taken by you.

- Hardware and Software Requirements and Tools Used

  Listing down the hardware and software requirements along with the tools, libraries and packages used. Describe all the software tools used along with a detailed description of tasks done with those tools.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  Describe the approaches you followed, both statistical and analytical, for solving of this problem.

- Testing of Identified Approaches (Algorithms)

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

- Run and Evaluate selected models

  Describe all the algorithms used along with the snapshot of their code and what were the results observed over different evaluation metrics.

- Key Metrics for success in solving problem under consideration

  What were the key metrics used along with justification for using it? You may also include statistical metrics used if any.

- Visualizations

  Mention all the plots made along with their pictures and what were the inferences and observations obtained from those. Describe them in detail.

  If different platforms were used, mention that as well.

- Interpretation of the Results

  Give a summary of what results were interpreted from the visualizations, preprocessing and modelling.

CONCLUSION

- Key Findings and Conclusions of the Study

  Describe the key findings, inferences, observations from the whole problem.

- Learning Outcomes of the Study in respect of Data Science

  List down your learnings obtained about the power of visualization, data cleaning and various algorithms used. You can describe which algorithm works best in which situation and what challenges you faced while working on this project and how did you overcome that.

- Limitations of this work and Scope for Future Work

  What are the limitations of this solution provided, the future scope? What all steps/techniques can be followed to further extend this study and improve the results.