Flight Price Prediction Project Report

Submitted by

Jameel Khan

ACKNOWLEDGMENT

This includes mentioning of all the references, research papers, data sources, professionals and other resources that helped me and guided me in completion of the project.

I wish to express my sincere gratitude to Mr. Shubham Yadav, SME for providing me an opportunity to do my internship and project work in “FLIP ROBO”.

It gives me immense pleasure in presenting this project report on “Flight Price Prediction Project”. It has been my privilege to have a team of project guide who have assisted me from the commencement of this project. The success of this project is a result of sheer hard work, and determination put in by me with the help of You Tube videos, references taken from Kaggle.com, skikit-learn.org. To know more about micro finance, I read

https://www.geeksforgeeks.org/

https://github.com/

https://www.mckinsey.com/

https://www.counterpointresearch.com/

I hereby take this opportunity to add a special note of thanks for to Mr. Shubham Yadav, who undertook to act as my mentor despite his many other professional commitments. Her wisdom, knowledge and commitment to the highest standards inspired and motivated me. Without his insight, support this project wouldn’t have reached fruitfulness.

The project is dedicated to all those people of Fliprobo, Datatrained who helped me while doing this project.

Introduction:

Flight ticket prices can be something hard to guess, today we might see a price, check out the price of the same flight tomorrow, it will be a different story. We might have often heard travelers saying that flight ticket prices are so unpredictable. Huh! Here we take on the challenge! As data scientists, we are going to prove that given the right data anything can be predicted. Let us took a dataset that contains prices of flight tickets for various airlines between the months of March and June of 2019 and between various cities in India.

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, we have to work on a project where we have to collect data of flight fares with other features and work to make a model to predict fares of flights.

Conceptual Background:

From the customer point of view, determining the minimum price or the best time to buy a ticket is the key issue. The concept of “tickets bought in advance are cheaper” is no longer working. It is possible that customers who bought a ticket earlier pay more than those who bought the same ticket later. Moreover, early purchasing implies a risk of commitment to a specific schedule that may need to be changed usually for a fee. The ticket price may be affected by several factors thus may change continuously. To address this, various studies were conducted to support the customer in determining an optimal ticket purchase time and ticket price prediction.

Review of Literature:

In this report, we discuss various applications and methods which inspired us to build our project. We did a background survey regarding the basic ideas of our project and used those ideas for the collection of information like the technological stack, algorithms, and shortcomings of our project which led us to build a better project. Paytm is a web platform where buyer can buy their tickets. It is a website with a simplified user interface which asks seller parameters like source, destination, date of journey and class (economy). These allow the web model to run certain algorithms on given parameters and predict the price. It is very difficult for the customer to purchase a flight ticket at the minimum price. For this several techniques are used to obtain the day at which the price of air ticket will be minimum. Most of these techniques are using sophisticated artificial intelligence (AI) research known as Machine Learning

Motivation:

Travelling through flights has become an integral part of today's lifestyle as more and more people are opting for faster travelling options. The flight ticket prices increase or decrease every now and then depending on various factors like timing of the flights, destination, duration of flights. various occasions such as vacations or festive season. Therefore, having some basic idea of the flight fares before planning the trip will surely help many people save money and time. In the proposed system a predictive model will be created by applying machine learning algorithms to the collected historical data of flights. This system will give people the idea about the trends that prices follow and also provide a predicted price value which they can refer to before booking their flight tickets to save money. This kind of system or service can be provided to the customers by flight booking companies which will help the customers to book their tickets accordingly.

Description of dataset

- **Price**: - it is the label column.
- **Flight name**: - it is the name of airline.
- **Departure_time**: - The time at which flight leaves for the destination.
- **Arrival time**: - The time at which flight arrives at destination.
- **Source**: - It is starting point of journey.
- **Destination**: - It is the ending point of the journey.
- **Duration**: - it is the total time taken during the journey.
- **No of stop**: - it is the halt taken by flight during the journey.
- **Date**: - It is date at which journey starts.

The task which is assigned is to fetch flight price data from a website and build a required model for that, I have used Paytm website for fetching the data as its URL is easy to modify and we can fetch data easily. The dataset consists of 10238 rows and 10 attributes including target column price. The above mention features help me in building machine learning model and predicting prices.

Data pre processing

Data processing and feature engineering are crucial in machine learning to build a prediction model. Furthermore, a model cannot be made without some data processing. For instance, as shown in the experiment, the model could not be trained before handling the missing values and converting the text in the dataset into numerical values. Hence, from the experiment, we saw that pre-processing the data does improve the prediction accuracy.

- **info**: it is used to give info about not null value and datatype of features. here datatype is (object) which is been taken care and converted to float and int datatype.

```
#checking datatype of object
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9307 entries, 0 to 9306
Data columns (total 10 columns):
 #    Column           Non-Null Count  Dtype
---   ------           --------------  -----
 0    Unnamed: 0       9307 non-null   int64
 1    Flight name      9307 non-null   object
 2    Departure_time   9307 non-null   object
 3    Arrival_time     9307 non-null   object
 4    Price            9307 non-null   int64
 5    Source           9307 non-null   object
 6    Destination      9307 non-null   object
 7    Duration         9307 non-null   object
 8    No of stop       9307 non-null   object
 9    Date             9307 non-null   object
dtypes: int64(2), object(8)
memory usage: 727.2+ KB
```

	Unnamed: 0	Price
count	10238.000000	10238.000000
mean	5118.500000	8779.683044
std	2955.600362	4711.374770
min	0.000000	1604.000000
25%	2559.250000	5654.750000
50%	5118.500000	7569.000000
75%	7677.750000	10957.000000
max	10237.000000	53992.000000

- **Describe**: The describe method is used for calculating some statistical data like **percentile, mean** and **std** of the numerical valuesof the Series or Data Frame. It analyses both numeric and object series and also the Data Frame column sets of mixed data types.it also give info about distribut

```
#checking unique in dataset
df.nunique().sort_values()
```

```
Source              2
Destination         4
Flight name         6
No of stop          6
Date               10
Departure_time    178
Arrival_time      221
Duration          381
Price            2649
Unnamed: 0      10238
dtype: int64
```

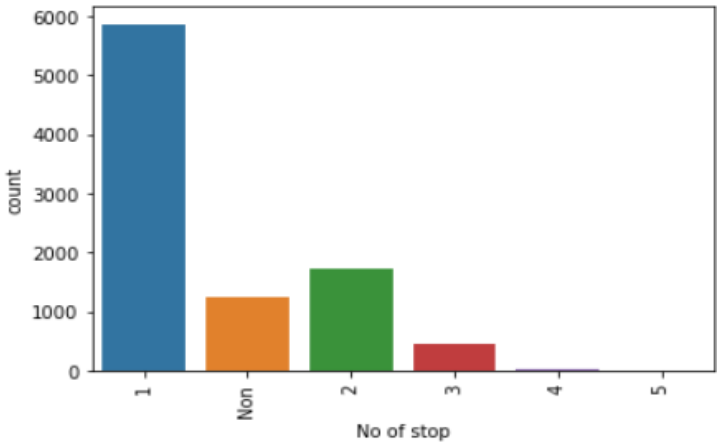
- **Unique**: This function is used to check the unique value in dataset.

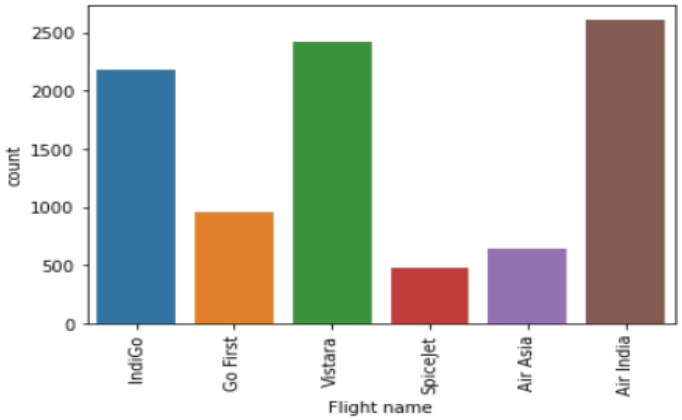
Visualization

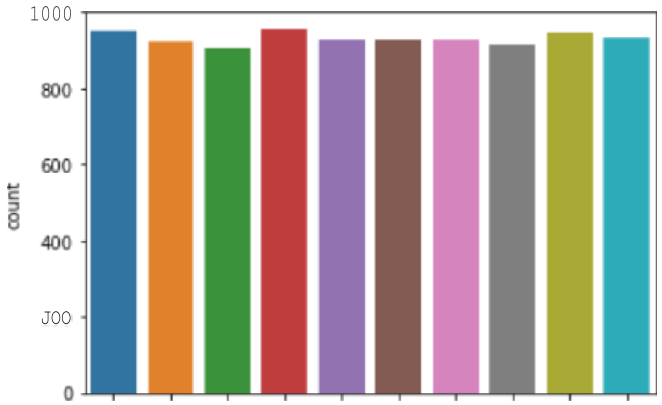
Data visualization is the discipline of trying to understand data by placing it in a visual context so that patterns, trends and correlations that might not otherwise be detected can be exposed.

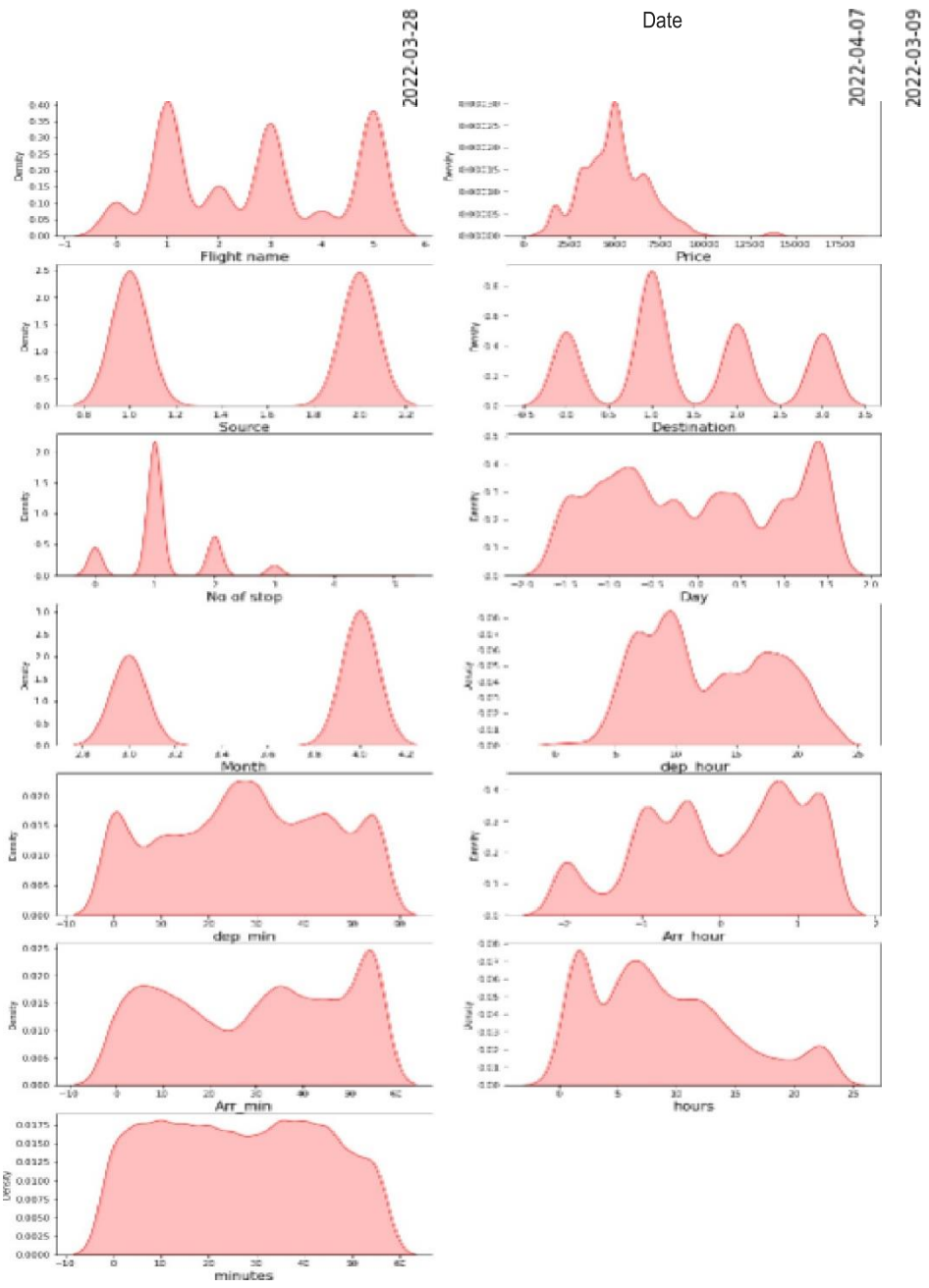
Univariate:

For univariate analysis I have used kde plot to analyse each column, it helps me in visualising skewness in the dataset as we can see price column right skewed.as we can in count plot most flight is from Vistara and mostly flight is one stop flight.

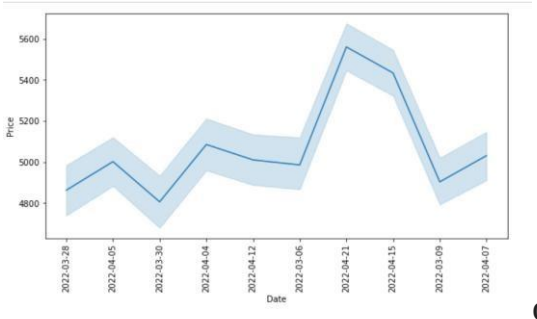




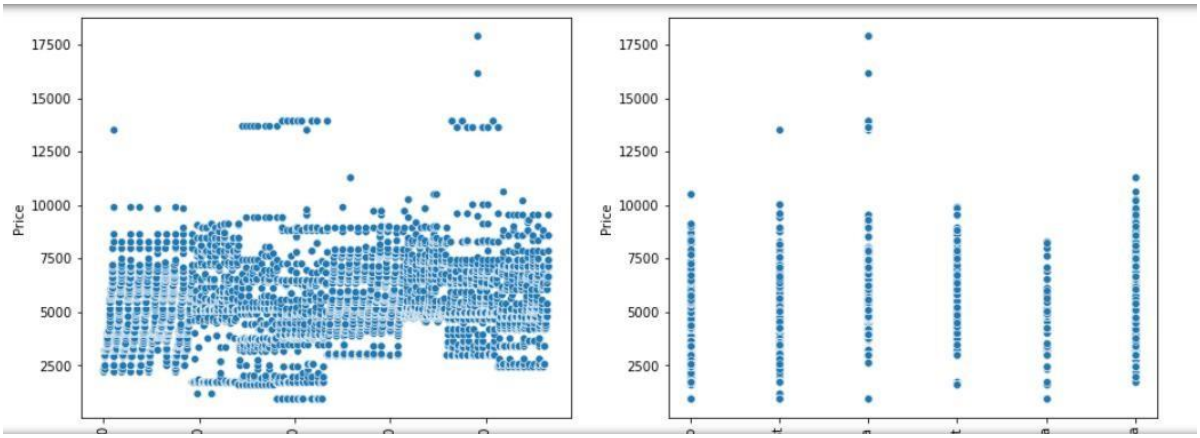

1000
800
600
400
JOO
0
count

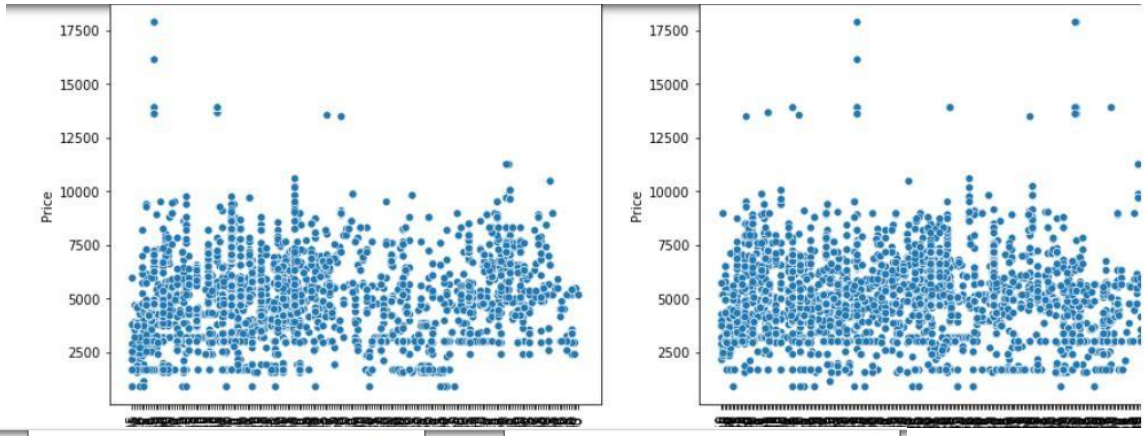

Date
2022-03-28
2022-04-07
2022-03-09
Flight name
Price
Source
Destination
No of stop
Day
Month
dep_hour
dep_min
Arr_hour
Arr_min
hours
minutes
Density

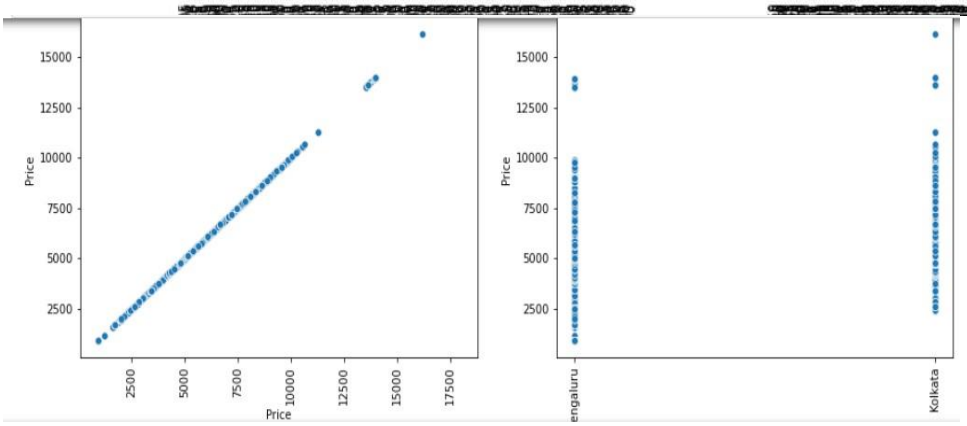
- **Univariate analysis**: Univariate analysis is the simplest form of data analysis where the data being analysed contains only one variable. Since it's a single variable it doesn't deal with causes or relationships. The main purpose of univariate analysis is to describe thedata and find patterns that exist within it. Here we can see from the graph, that price column is right skewed but it is our target column so we keep it as it is.

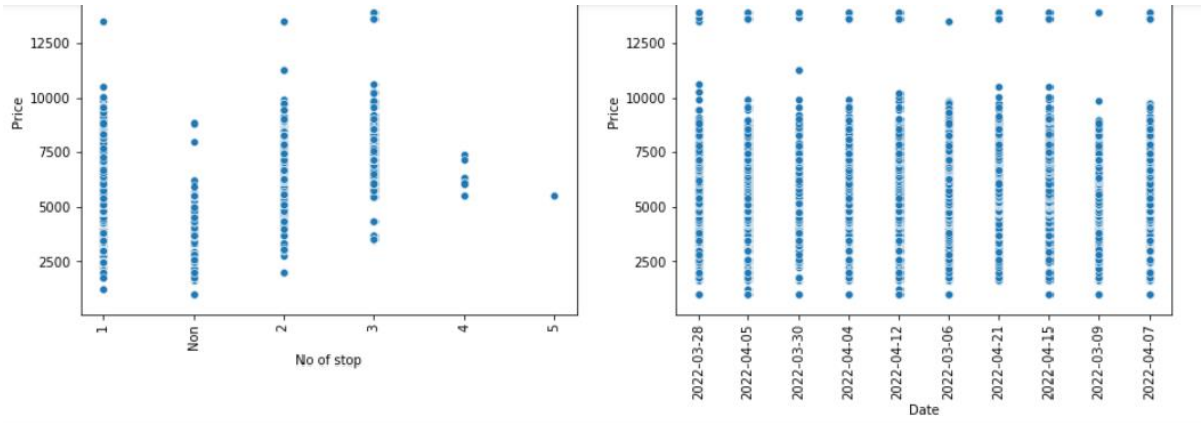
- **Bivariate analysis**: Bivariate analysis is used analyse relationship between feature and label and between two different variables. With the help 

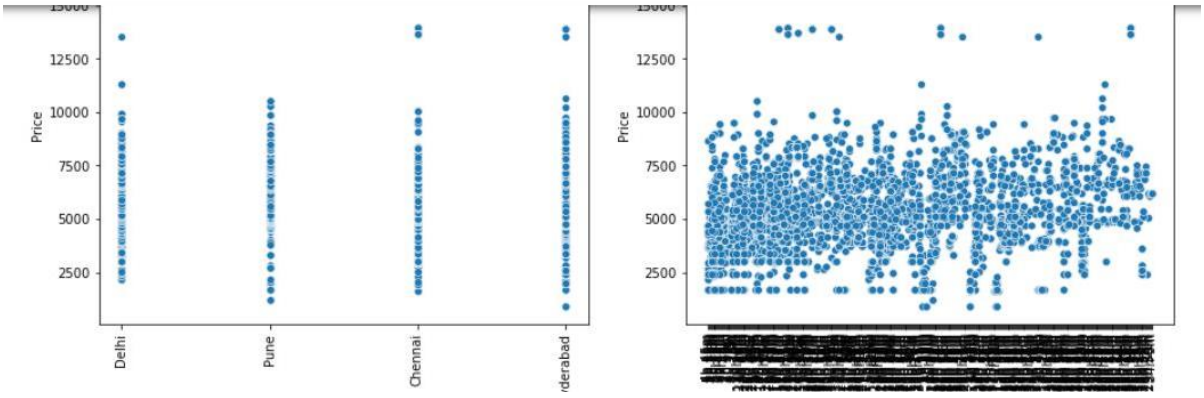
of line plot, we can see the relationship between price and how it is varying with date. Last minute flight is costly as well as flight which is near some festive occasion is costly

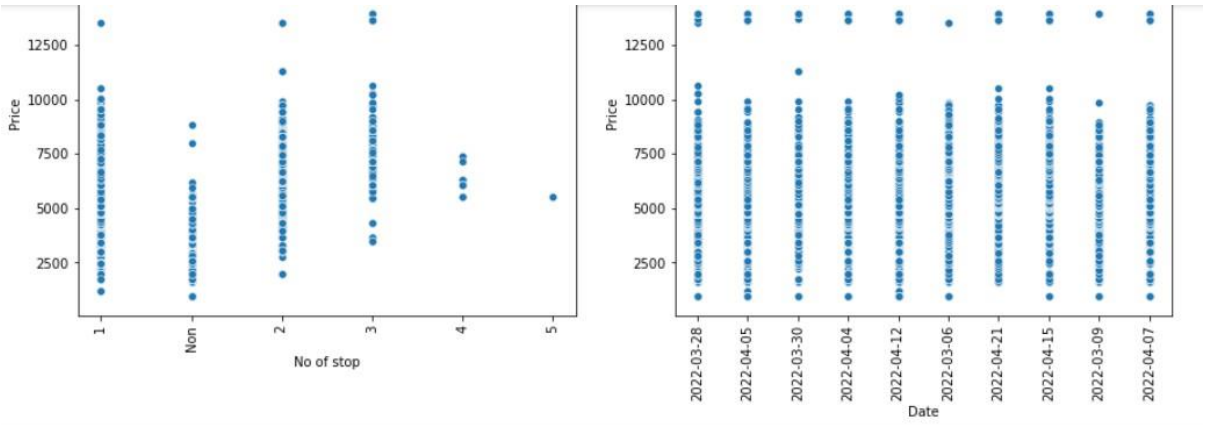


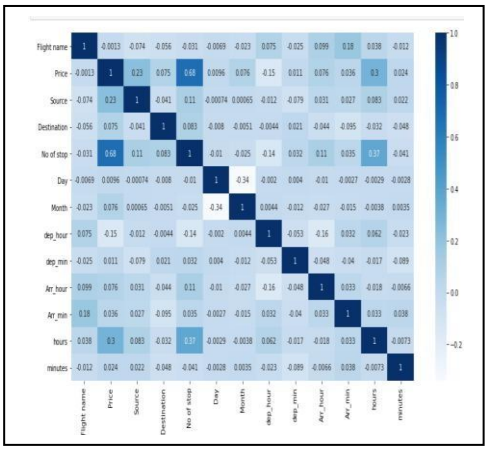

Price
17500
15000
12500
10000
7500
5000
2500


Price
15000
12500
10000
7500
5000
2500
2500
5000
7500
10000
12500
15000
17500
Kolkata


Price
12500
10000
7500
5000
2500
1
Non
2
3
4
5
No of stop
2022-03-28
2022-04-05
2022-03-30
2022-04-04
2022-04-12
2022-03-06
2022-04-21
2022-04-15
2022-03-09
2022-04-07
Date


Price
12500
10000
7500
5000
2500
Delhi
Pune
Chennai





❖ **Multivariate analysis**: multivariate analysis is done on three or more than three variables. Through multivariate analysis we try to find out correlation of different feature with target variable as well as among themselves. I have used heatmap for multivariate analysis. We don't find any multicollinearity problem, only we can see strong correlation between day and month.

Feature Engineering & assumption

```
#splitting date into date and month
df["Date"]= pd.to_datetime(df["Date"])
df["Day"]=df["Date"].dt.day
df["Month"]=df["Date"].dt.month
```

Feature engineering involves extract features from raw dataset along with the use of domain knowledge. Feature engineering is useful to improve theperformance of machine learning algorithms. Here we splitting date in day and month

```
#replacing source manually with 1 and 2
df["Source"] = df["Source"].replace("Bengaluru",1)
df["Source"] = df["Source"].replace("Kolkata",2)
```

```
# splitting duration into and hour and min in seperate column for better understanding
s=pd.to_timedelta(df['Duration'])
df['hours']=s.dt.components['hours']
df['minutes']=s.dt.components['minutes']
```

Encoding: manually encoding certain column to make it meaningful and which is helping in building a better machine learning model with good accuracy.


```
# splitting duration into and hour and min in seperate column for better understanding
s=pd.to_timedelta(df['Duration'])
df['hours']=s.dt.components['hours']
df['minutes']=s.dt.components['minutes']
```


Time Delta: this function is used to split duration into hours and minute so that it is converted into numerical column as machine learning model only understands numerical data it does not understand text.

Detection of Outliers: An outlier is an extremely high or extremely low-value value in the data set that tend to impact model performance. It is important to identify and take care of outliers to improve the model accuracy. The box plot is useful graphical display for describing the behavior of the data in the middle as well as the ends of distribution.it works on interquartile method to determine outlier.in this dataset only price column have outlier but is our target column so we don't perform any operation on target column so keeping it as it is.

Skewness: skewness is also an important factor which affects the model accuracy. In skewed data tail region act as an outlier for statistical model and may affect the model accuracy. In this dataset mostly column is categorical and skewness is only present in label column and we don't take care of skewness of label as well as categorical columns.

Model development and evaluation

Firstly, I analyse the dataset then I found out that datatype of certain columns is not correct so I corrected that and with the help of feature engineering I have created numerical column out of object column as our machine learning model

will not understand text, then for understanding I have used plots to visualise and understand data. After imputation comes data cleaning which is done by detecting outliers, checking skewness and removing unwanted columns which does not add value to model accuracy. After data cleaning splitting of data into test and train and then doing normalisation of data so that it does not get biased toward a particular feature. Finally comes model building, I have used 4 classifier algorithm () which fits best in this dataset and predict good accuracy. After finding my best model with the help of accuracy and cross validation score comes hyper parameter tuning for best fit model.

Hardware & Software Requirements& Tools Used While taking up the project

Data Science task should be done with sophisticated machine with high end machine configuration. The machine which I'm currently using is powered by intel core i5 processor with 8GB of RAM. With this above-mentioned configuration, I managed to work with the data set in Jupyter Notebook which help us to write Python codes. As I'm using low configuration machine so it took more time then usual to execute codes. The library used for the assignment are Numpy, Pandas, Matplotlib, Seaborn, Scikit learn

Librariesrequired :-

✓ To run the program and to build the model we need some basic libraries as follows

- **Pandas** for data analysis and import
- **NumPy** to perform Mathematical operation
- **Seaborn** and **matplotlib** to data visualization
- **Scikit-learn:** All the models, metrics and feature selection etc are present inside of that module. We import from this library according to our need.

Model Building: I use 4 different algorithms for model building

LinearRegression

```
N        sklearn.linear_model                L? n ea mR e que s s? on
```

```
LR- L1 n ea r'R eg r‘e s s1 on ( )
L R . I ¿Jz (x_6 m ain      y_Jz n a Tn )
p r'in6 ( LR. s z O r'e ( x_Jz r' a T n y Jz mai n ) )
LR_pr'e d1 ct - LR . p r‘e d1 c Jz (x_4 e st )
```

```
B . 564642427 B414G 57
```

```
N  paim   (' i'-'S E : ', me a n_ sq u a r'ed_er‘r'a r‘ ( LR_p r‘ed1 cI , y_6 e st ) )
   print('MAE:',mean_absolute_error(LR_predict,y_test))
   print('r2_score:',r2_score(LR_predict,y_test))
```

```
MSE: 1707541.4B772502G1
MAE: 937.47b003209B758
w 2 s z c› e : 8 . 848 5 8 7 8 8 5 3 224 31 G 6 5
```

RandomForestRegressor

```
▶|       sklearn.ensemble import RandomForestRegressor
```

```
R F zRa ndom For'estRegr‘e ssor'( )
RF.fit(x train,y train)
pri nI (RF . s c o r'e (x_t r'ai n, y_t   a:i n ))
R F_PR EDz R F . p r‘ed1ct:(x_t e st )
```

```
B . 9882787153444812
```

```
b| point ( ' i''S E : ' , me an_s qua r'ed er’r'ar‘ ( R F PRED , y_t: e st )
   pw1nt ( ' i''AE : ' , mean_a b sob ut e_er now ( R F_PRE D, y_t:eat: )
   p r‘i nI ( ' r° 2_ s c a r° * : ' , r‘ 2_ s c a r‘ e ( RF_PRED, y_t:esI: ) )
```

```
/'1SE : 268115 . 7861346187
/'1AE : 199. 154045 373 58116
r2_score: 0.935709498G69G991
```

DecisionTreeRegressor

```
from sklearn.tree import DecisionTreeRegressor

DTR=DecisionTreeRegressor()
DTR.fit(x_train,y_train)
print(DTR.score(x_train,y_train))
DTR_PRED=DTR.predict(x_test)
```

```
0.9952267177798488
```

```
print('MSE:',mean_squared_error(DTR_PRED,y_test))
print('MAE:',mean_absolute_error(DTR_PRED,y_test))
print('r2_score:',r2_score(DTR_PRED,y_test))
```

```
MSE: 318020.5542291197
MAE: 173.1476323119777
r2_score: 0.907002280136539
```

GradientBoostingRegressor

```
from sklearn.ensemble import GradientBoostingRegressor

GBR=GradientBoostingRegressor()
GBR.fit(x_train,y_train)
print(GBR.score(x_train,y_train))
GBR_PRED=GBR.predict(x_test)
```

```
0.7897249105040695
```

```
print('MSE:',mean_squared_error(GBR_PRED,y_test))
print('MAE:',mean_absolute_error(GBR_PRED,y_test))
print('r2_score:',r2_score(GBR_PRED,y_test))
```

```
MSE: 750318.7340662555
MAE: 625.0382581498238
r2_score: 0.6841461274485934
```

Standard Scaler:

Scaling of Features is an essential step in modelling the algorithms with the datasets. So, the data obtained contains features of various dimensions and scales altogether. Different scales of the data features affect the modelling of a dataset adversely. It leads to a biased outcome of predictions in terms of misclassification error and accuracy rates. Thus, it is necessary to Scale the data prior to modelling. StandardScaler **removes the mean and scales each feature/variable to unit variance**. This operation is performed feature-wise in an independent way.

Cross Validation Score:

Cross-validation is a statistical method used to estimate the skill of machine learning models. It is commonly used in applied machine learning to compare and select a model for a given predictive modelling problem because it is easy to understand, easy to implement, and results in skill estimates that generally have a lower bias than other methods. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation.

R-2 Score:

The most common interpretation of r-2 score is how well the regression model fits the observed data. for ex an r-2 squared of 70% revels that 70% of the data fit the regression model. Generally, a higher r-2 score indicates a better fit for the model

RESULT:

Many machine learning algorithms are used to predict. However, the prediction accuracy of these algorithms depends heavily on the given data when training the model. If the data is in bad shape, the model will be overfitted and inefficient,which means that data pre- processing is an important part of this experiment and will affect the final results. Thus, multiple combinations of pre-processing methods need to be tested before getting the data ready to be used in training. After analyzing every model XGB Regressor shows good accuracy and cv with least difference and on doing hyper parameter tuning it accuracy reaches to 89%. If we compare the flight prices of the month of January, we can see that price of the flight changes frequently near the date of November. The flight prices increase in large increments near the departure dates. But increases and decreases in very small increments afterwards. Similarly, if we compare the flight prices in the month of November and December, we can see that flight prices increases. Therefore, we concluded that the flight prices tend to decrease over time, sometimes they show sudden increases but most of the time they tend to decrease

cross validation

```
Q from sklearn.model selection inport cross val  score

   np.random.seed(lB)
   del° r'mse_cv(model   x y) :
        rmse :- (cross_val_score(model, x,y, scoring:'ne   mean_squared_error', cv:lB))
        re-tu rn ( rmse )


   models:  [LinearRegression(),
               RandomPorestRegressor(),
               DecisionTreeRegressor(),
               GradientBoostingRegressor(),]



   names:  ['LR','RF','DTR','GBR']

   for model,name in zip(models,names):
        s crime - rmse cv ( nodel , x, y)
        pr'1nt ("   }      :    :.6f)     ( :4f} " .for'mat (name, sc or'e .mean( ) s cor'e. std ()))
```

```
LR      : 1ag1e16. 236175 , 424451.832B69
RF      : 626078 . 944496, 315471.440180
DTR      : 957619 .462525, 427729 .219266
GBR      : sg6z  s. 6775B2, 367956 .852593
```

According to all metrics score selecting the RandomForestRegressor for GridSeachCV

HYPER PARAMETER TUNIHG

RandomForestRegressor

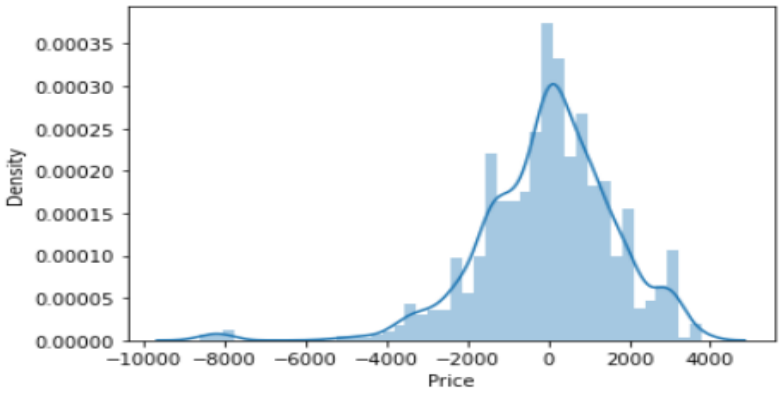
```
   from sklearn.model_selection inport GrldSearchCV

§| RF -RandomF onesI Regne ssor'( )
   param:{
        'criterion':['mse','mae'],
       'n_estimators':[100, 200],
       'max_depth':[1],
      'max_features' : [”sqrt”, ”log2”,],


   RF_grid:GridSearchCV(RandomForestRegressor(),param,cv:4,scoring:'accuracy',n_jobs:-l,verbose:2)

|8 RF_grid.fit(x_train,y_train)
   RF_grid_pred:RF rid.best_estimator_.predict(x_test)

   Fitting 4 folds for each of 8 candidates, totalling 32 fits


   RF_grid.bestgarams_

[67a: {'criterion': 'mse',
       'max_depth': 1,
       'max_features': 'sqrt',
       'n_estimators': 100}
```

```
print('MSE:',mean_squared_error(RF_PRED,y_test))
print('MAE:',mean_absolute_error(RF_PRED,y_test))
print('r2_score:',r2_score(RF_PRED,y_test))
```

```
MSE: 208115.7861340187
MAE: 199.13404537358116
r2_score: 0.9357094986696991
```

```
sn.distplot(RF_grid_pred-y_test)
```

```
<AxesSubplot:xlabel='Price', ylabel='Density'>
```



CONCLUSION:

Flight price prediction can be a challenging task due to the high number of attributes that should be considered for the accurate prediction. The major step in the prediction process is collection and preprocessing of the data which we have successfully done using different machine learning algorithms like a Decision tree, Bagging, Gradient Boosting and XGB so it's clear that XGB have more accuracy in prediction when compared to the others and also my research provides to find the attributes contribution in prediction. So, I would believe this research will be helpful for both the company and people, the future works are stated below Every system and new software technology can help in the future to predict the prices. Although, this model has achieved astonishing performance in flight price prediction problem our aim for the future research is to test this model to work successfully with various data sets.

FINDING:

The factors that have been studied in this study has a weak correlation with the label except some. Hence, by adding more features to the local dataset that affect the price prediction, such as meals, safety and many more.

LIMITATION:

As per my understanding and study of dataset I presented the best model with good accuracy, this is the best model until someone came with an extraordinary approach and technique. Moreover, this study will not cover all classification algorithms, instead, it is focused on the chosen algorithm, starting fromthe basic classification techniques to the advanced ones

FUTURE WORK:

In future this machine learning model may bind with various website which can provide real time data for price prediction. Also, we may add large historical data of flight price which can help to improve accuracy of the machine learning model. We can build an android app as user interface for interacting with user. For better performance, we plan to judiciously design deep learning network structures, use adaptive learning rates and train on clusters of data rather than the whole dataset.

Thank you

****