FLIGHT FARE PREDICTION

Prepared by:-
Vivek Kumar
Internship33

CONTENT

- Overview.
- Problem Statement.
- Problem Understanding.
- What is Flight Price Prediction?
- Importance of Flight price prediction.
- Exploratory data analysis.
- Visualizations.
- Analysis.
- Data cleaning steps.
- Model Building.
- Hyper Parameter Tunning.
- Saving the model and predictions from saved best model.
- Conclusion.

PROBLEM STATEMENT

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) .

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.

PROBLEM UNDERSTANDING

Flight prices are something unpredictable. It’s more than likely that we spent hours on the internet researching flight deals, trying to figure an airfare pricing system that seems completely random every day. Flight price appears to fluctuate without reason and longer flights aren’t always more expensive than shorter ones.

But now the question is how to know proper Flight price, for that I have built a Machine learning model which can predict the Flight price. Using various features like **Airline, Source, Destination, Arrival time, Departure time, Stops, Travelling date and the Price for the same travel**. So using all these previously known information and analysing the data I have achieved a good model that has **99.9% accuracy**. So let’s understand what all the steps we did to reach this good accuracy.

WHAT IS FLIGHT PREDICTION?

It is hard for the client to buy an air ticket at the most reduced cost. For this few procedures are explored to determine time and date to grab air tickets with minimum fare rate. The majority of these systems are utilizing the modern computerized system known as Machine Learning. The model guesses airfare well in advance from the known information. This framework is proposed to change various added value arrangements into included added value arrangement heading which can support to solo gathering estimation.

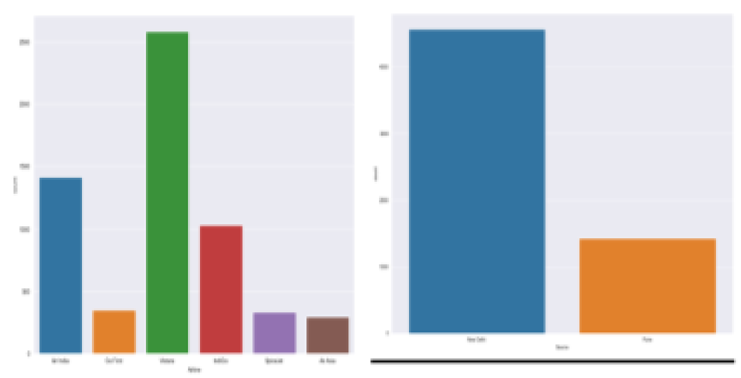
IMPORTANCE OF FLIGHT PRICE PREDICTION

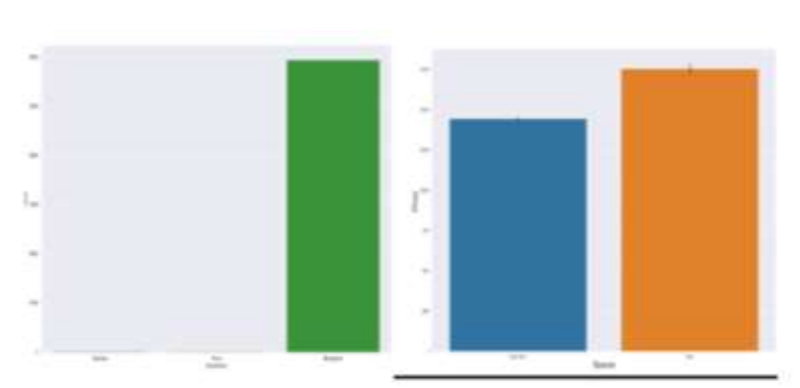
It is hard for the client to buy an air ticket at the most reduced cost. For this few procedures are explored to determine time and date to grab air tickets with minimum fare rate. The majority of these systems are utilizing the modern computerized system known as Machine Learning. The model guesses airfare well in advance from the known information. This framework is proposed to change various added value arrangements into included added value arrangement heading which can support to solo gathering estimation.

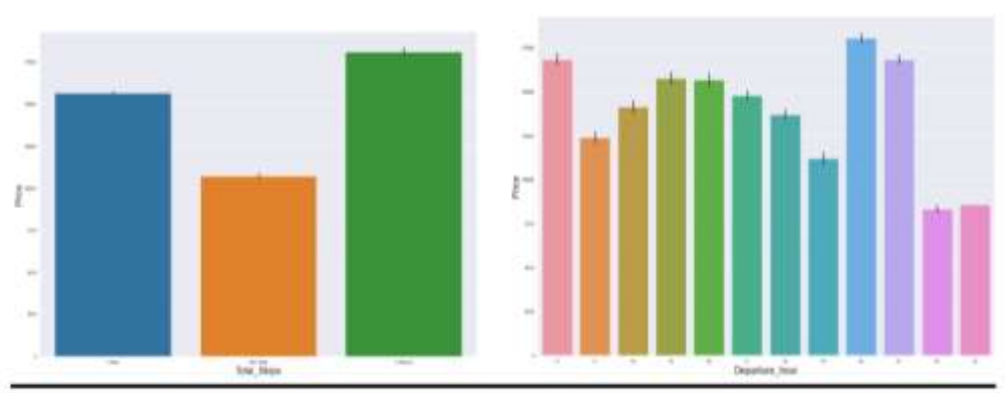
EXPLORATORY DATA ANALYSIS:

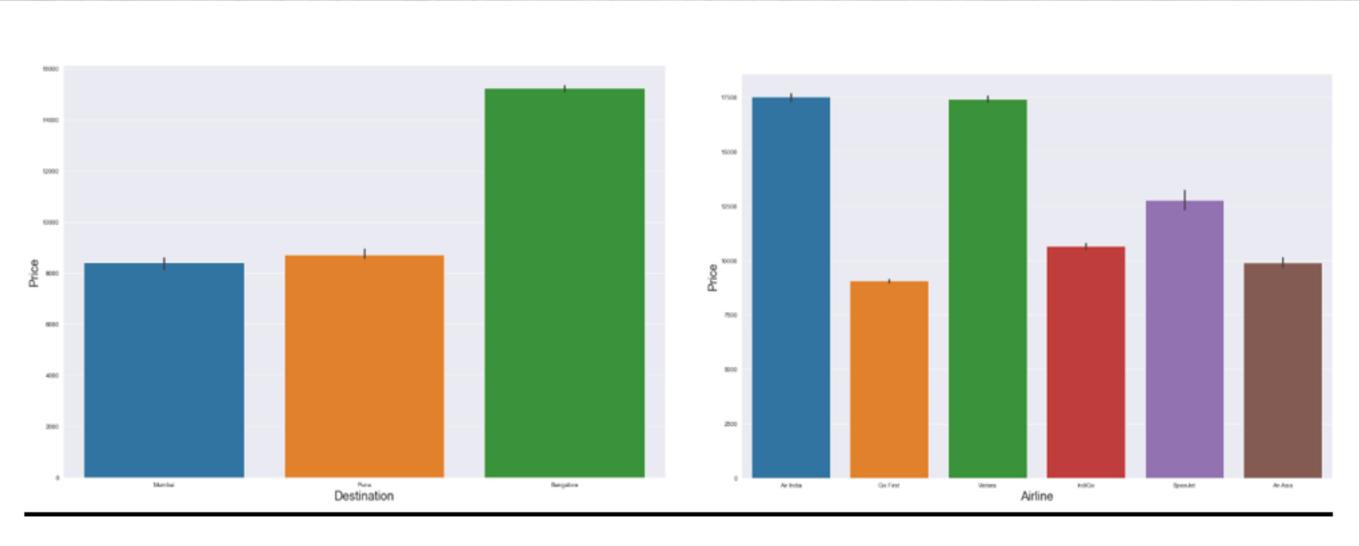
- ❑ As a first step I have scrapped the required data using selenium from yatra website.
- ❑ And I have imported required libraries and I have imported the dataset which was in csv format.
- ❑ Then I did all the statistical analysis like checking shape, nunique, value counts, info etc.
- ❑ I have also dropped Unnamed:0 column as I found it was the index column of csv file.
- ❑ Next as a part of feature extraction I converted the data types of datetime columns and I have extracted useful information from the raw dataset. Thinking that this data will help us more than raw data.

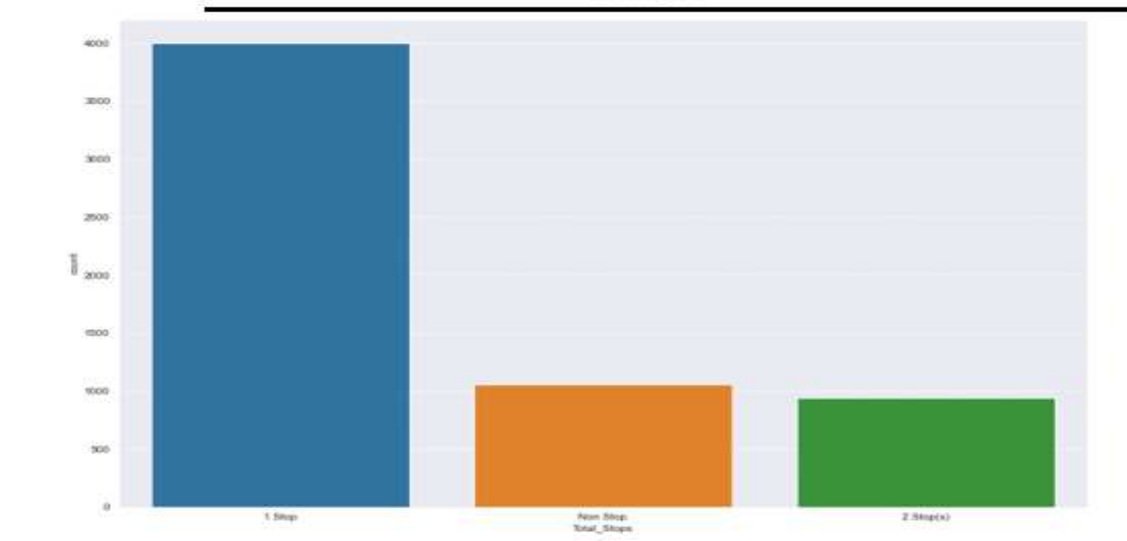
VISUALIZATION OF CATEGORICAL COLUMNS



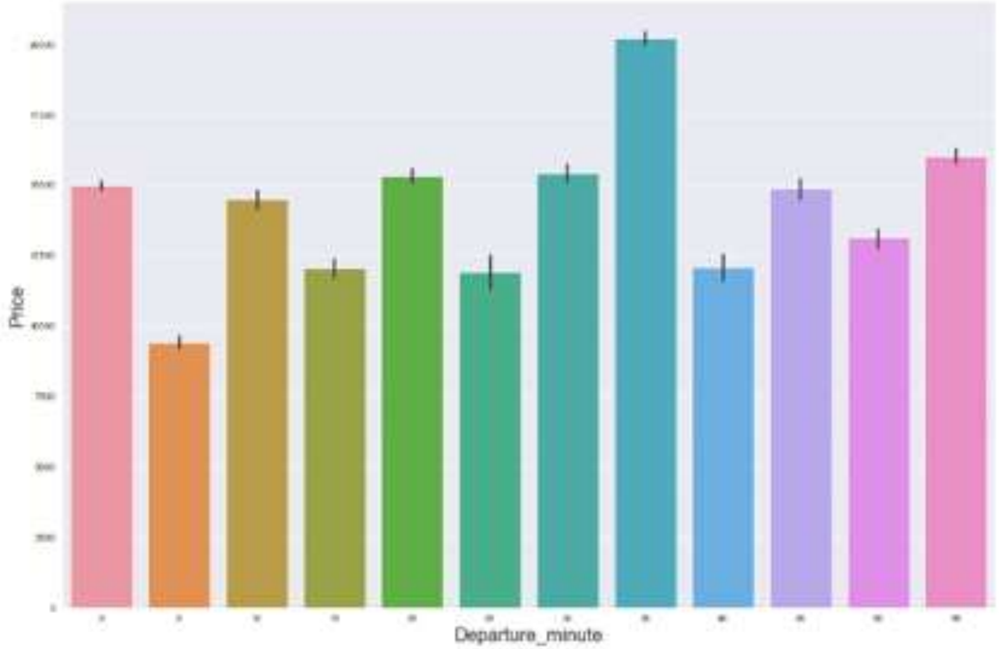


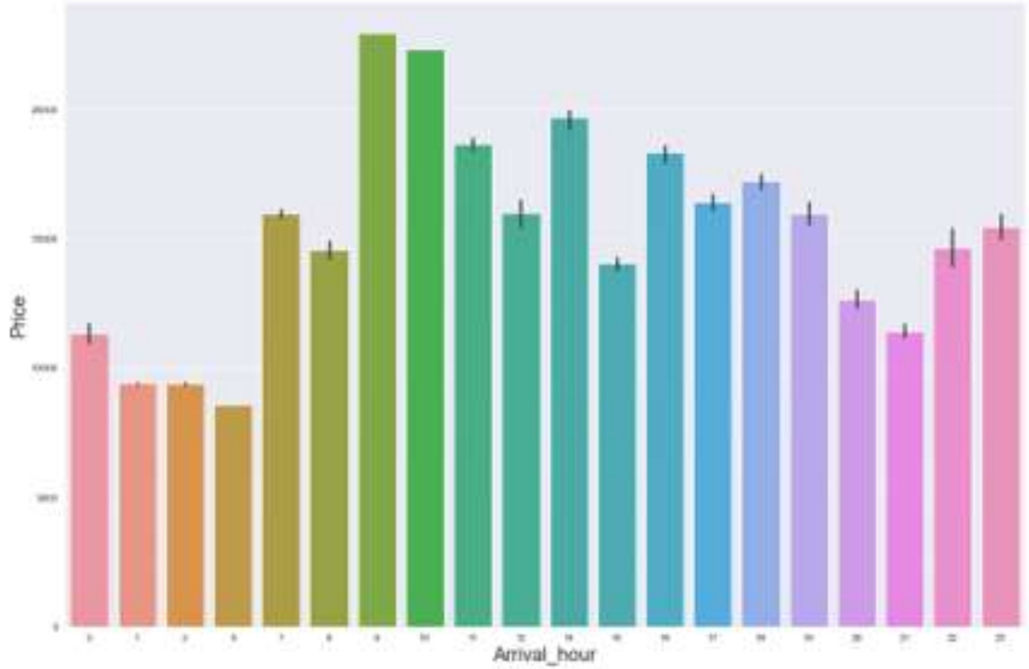






VISUALIZATION OF CATEGORICAL COLUMNS

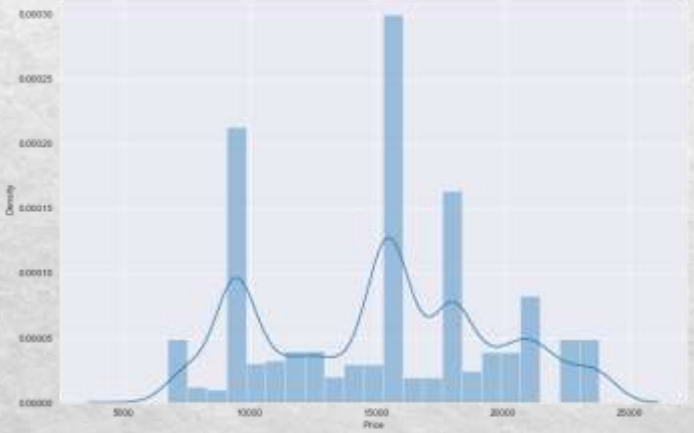


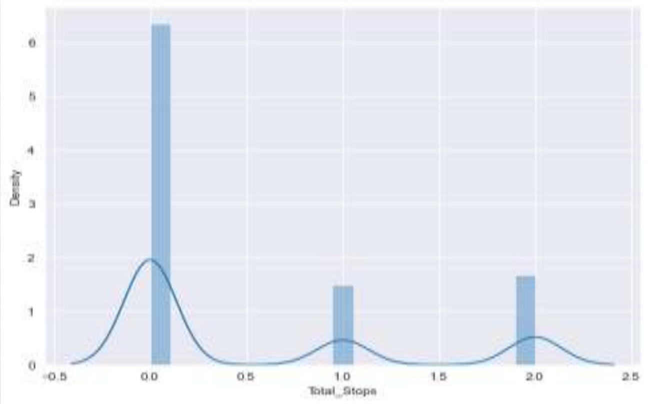


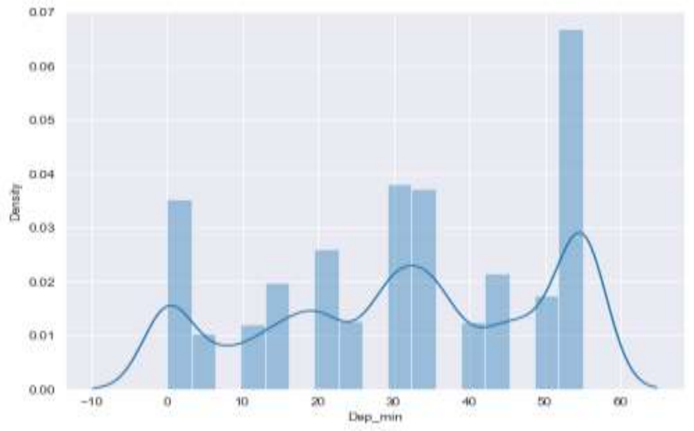
OBSERVATIONS

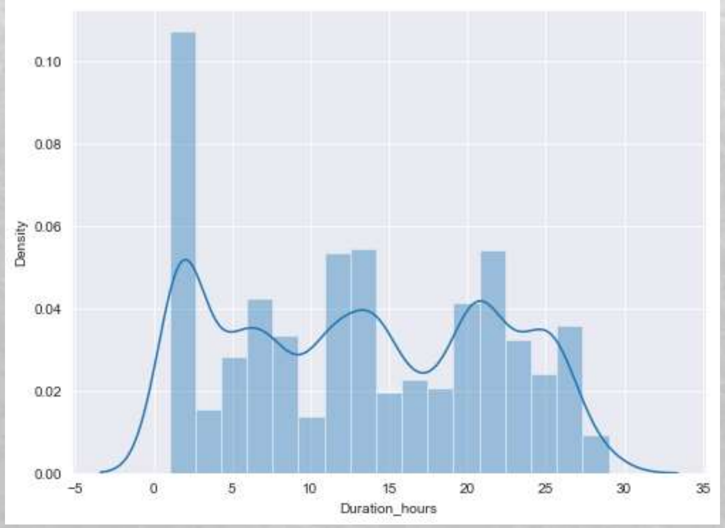
- ❑ Vistara has largest share in market followed by AirIndia and Indigo.
- ❑ In our given dataset almost 76% of fight starts from New Delhi and rest from Pune
- ❑ In our given dataset almost 99% of fights destination is to Bangalore
- ❑ Ticket price of flights starting from Pune is costlier than New Delhi
- ❑ Ticket price of flights to Bangalore is costlier than Mumbai and Pune
- ❑ Ticket prices of AirIndia and Vistara are the costliest followed by SpiceJet, IndiGo, AirAsia and GoFirst
- ❑ Ticket prices are high for fights with 2 stops. NonStop flight ticket prices are low.
- ❑ "Departure_Hour vs Price": From the bar plot and line plot we can see that there are some flights departing in the noon 12 AM having most expensive ticket prices compared to late evening flights. We can also observe the flight ticket prices are higher during evening (may fluctuate) and it decreases in the late night.
- ❑ Ticket prices are very high during 9-10am where as early morning flights have low prices.
- ❑ Long duration flights have higher ticket price and short duration flights have low ticket price

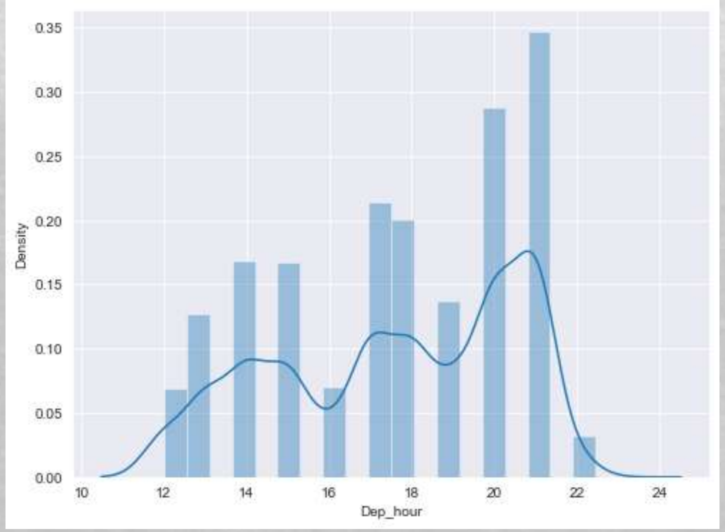
VIZUALIZATION OF NUMIRICALCAL COLUMNS

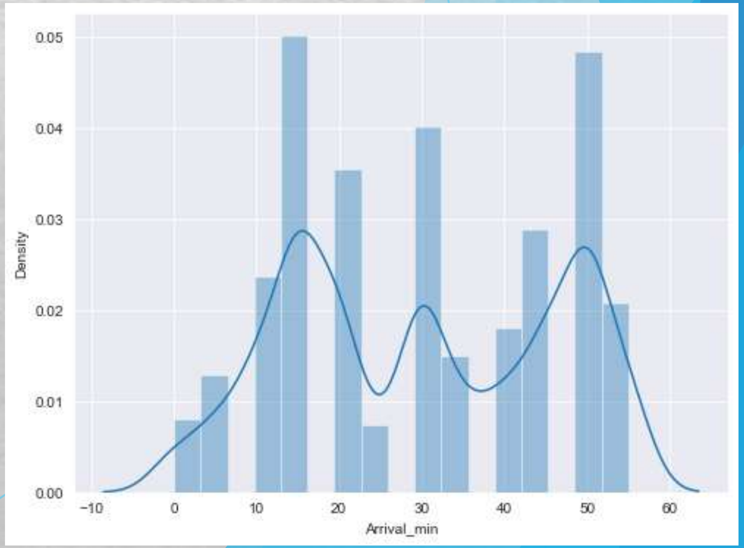












OBSERVATIONS

- From the distribution plot we can observe that most of the columns are somewhat distributed normally as they have no proper bell shape curve.
- "Price" is widely distributed between the range of 5000 - 25000.we can observe that most number of tickets are priced at 15000 and 10000.
- The data in the column Arrival Hour and Arrival_min skewed to left since the mean values is less than the median.

ANALYSIS:

- I have used dist plot to check the skewness in numerical columns.
- I have used bar plot for each of categorical feature that shows the relation with the median flight price for all the sub categories in each categorical feature.
- I found that there is no much linear relationship between continuous numerical variable and Flight Price.

DATA CLEANING STEPS

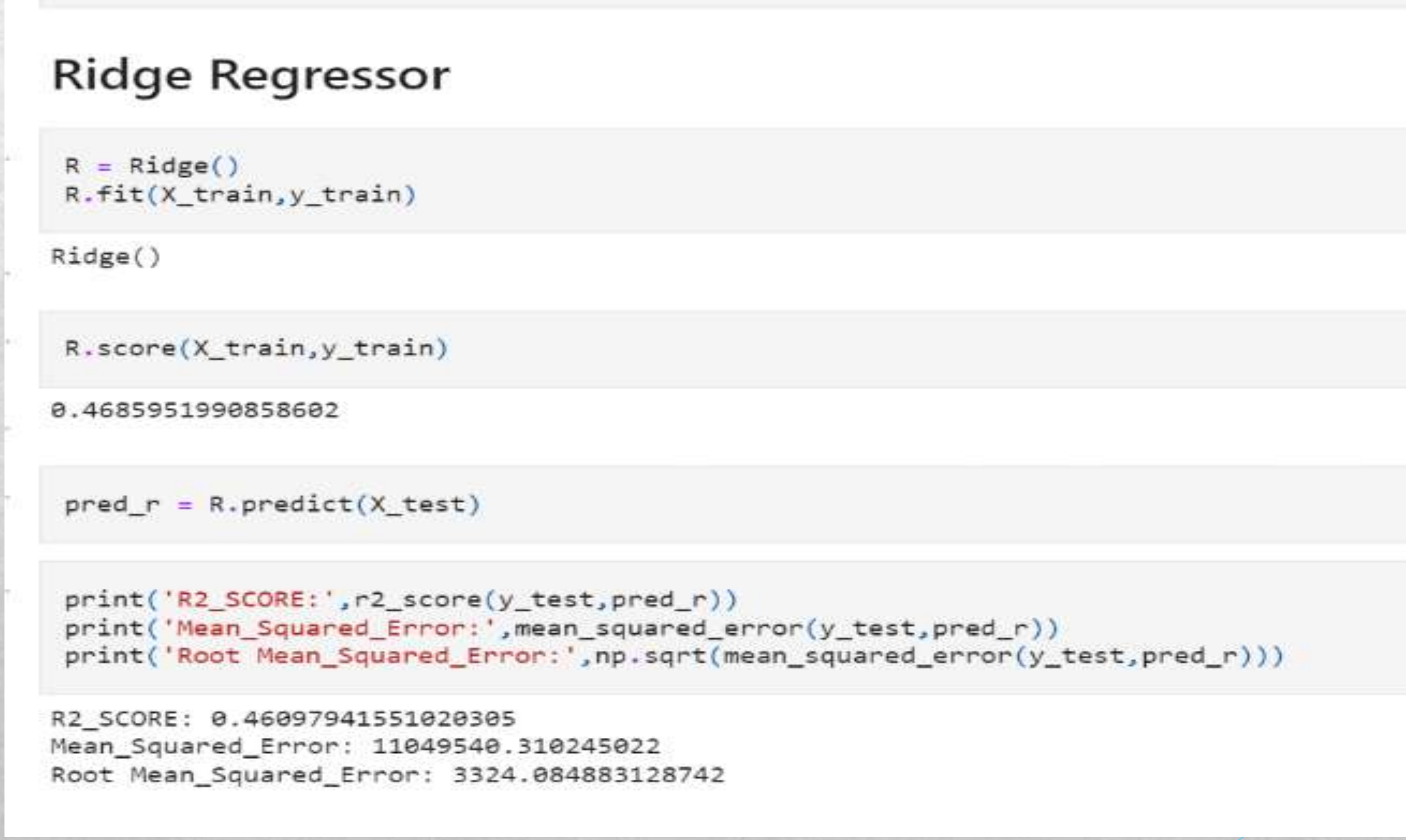
- Data has been scrapped from yatra.com website so we have to clean it for our convenience.
- In my datasets I found there is no null values, outliers and also skewness.
- To encode the categorical columns I have use Label Encoding.
- VIF method was used to check the correlation between dependent and independent features.
- Also I have used standardization. Then followed by model building with all regression algorithms.

MODEL BUILDING

Since Price was my target and it was a continuous column so this problem is considered regression problem. And I have used all regression algorithms to build my model. By looking into the difference of r2 score and cross validation score I found XGB Regressor as a best model with least difference. 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. Below are the list of regression algorithms I have used in my project.

- RidgeRegressor
- XGBRegressor
- ExtraTreesRegressor
- DecisionTreeRegressor

1.RIDGE REGRESSOR



Ridge Regressor

```
R = Ridge()
R.fit(X_train,y_train)
```

```
Ridge()
```

```
R.score(X_train,y_train)
```

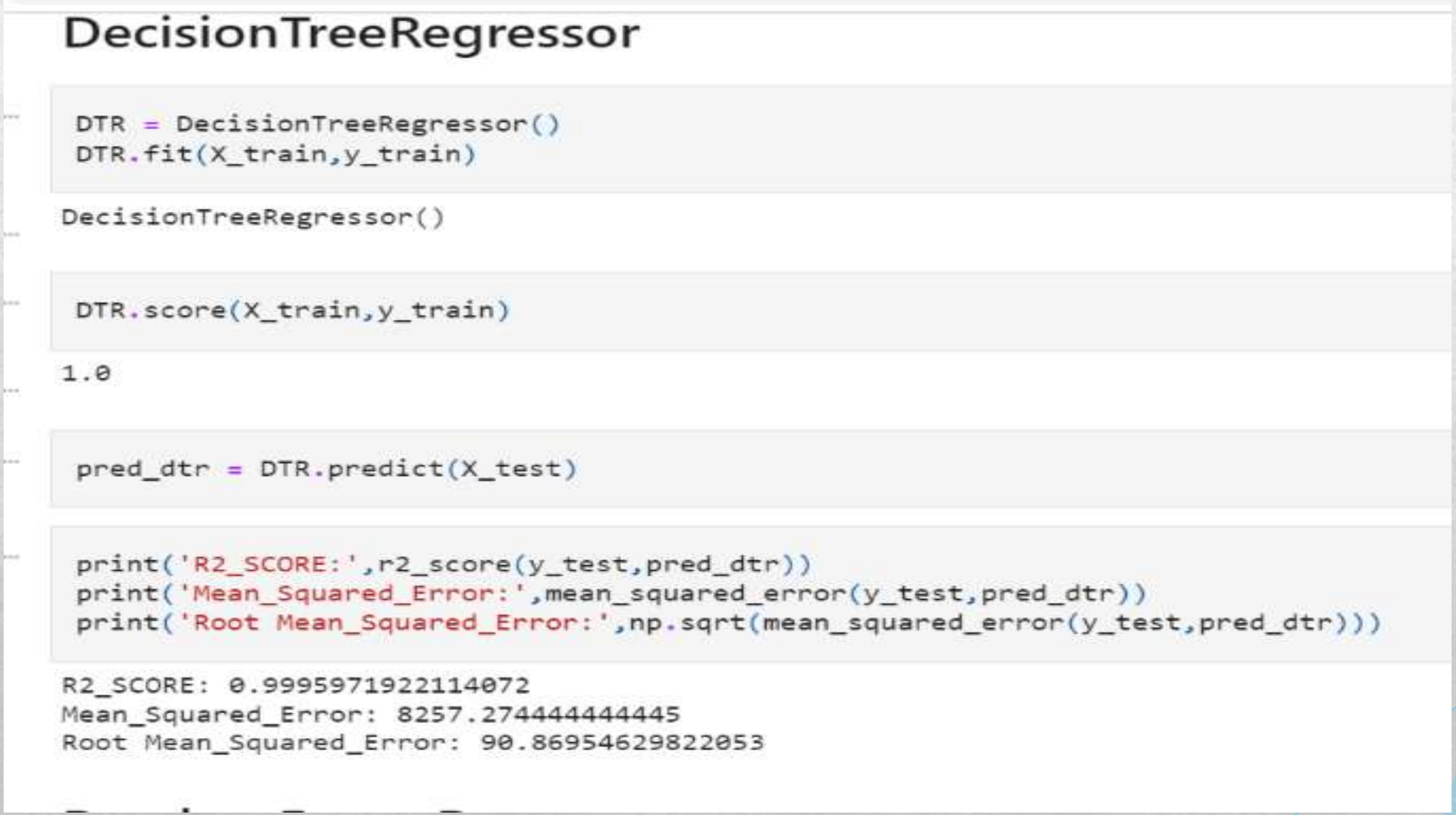
```
0.4685951990858602
```

```
pred_r = R.predict(X_test)
```

```
print('R2_SCORE:',r2_score(y_test,pred_r))
print('Mean_Squared_Error:',mean_squared_error(y_test,pred_r))
print('Root Mean_Squared_Error:',np.sqrt(mean_squared_error(y_test,pred_r)))
```

```
R2_SCORE: 0.46097941551020305
Mean_Squared_Error: 11049540.310245022
Root Mean_Squared_Error: 3324.084883128742
```

2. DECISION TREE REGRESSOR



DecisionTreeRegressor

```
DTR = DecisionTreeRegressor()
DTR.fit(X_train,y_train)
```

```
DecisionTreeRegressor()
```

```
DTR.score(X_train,y_train)
```

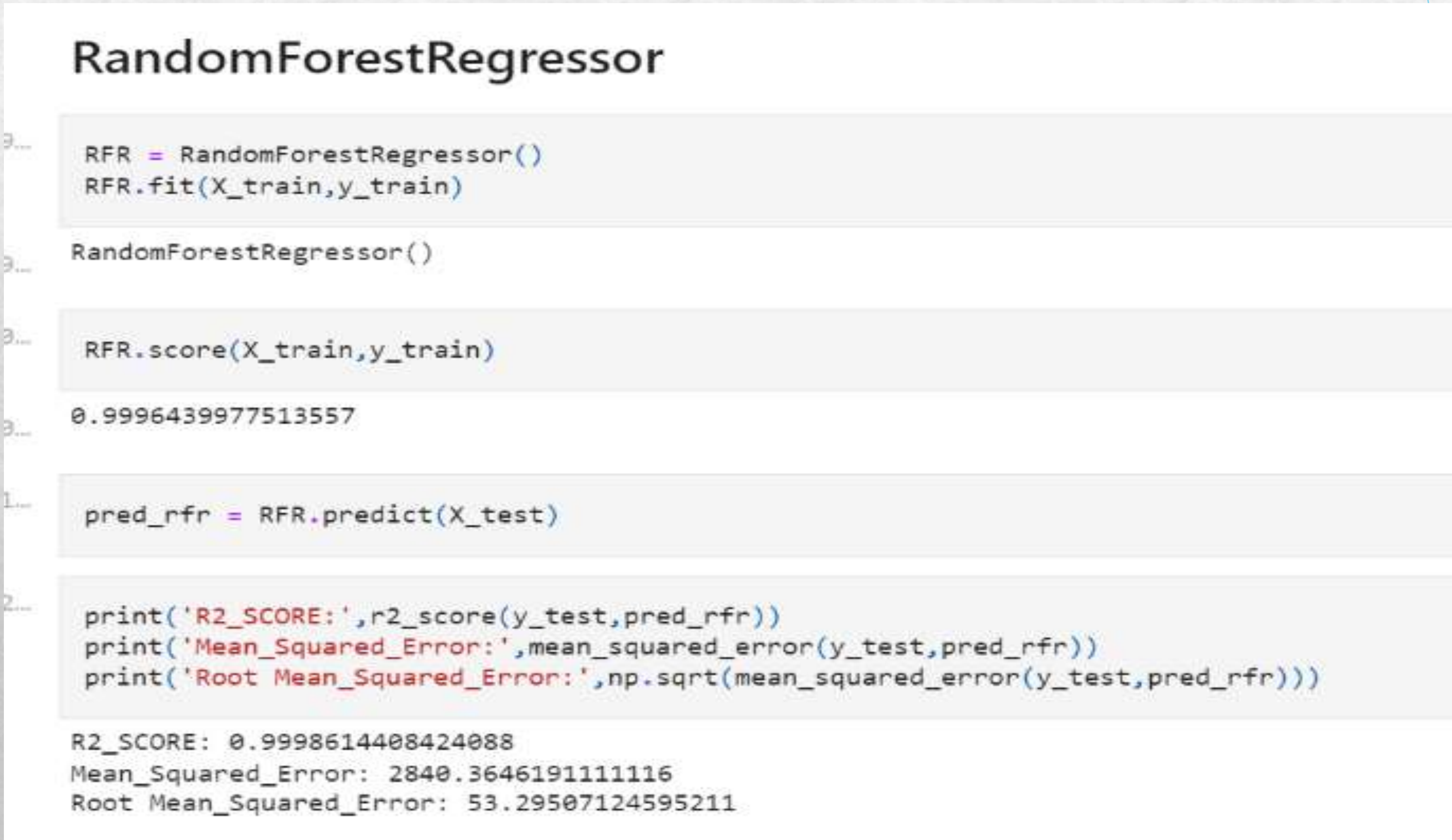
```
1.0
```

```
pred_dtr = DTR.predict(X_test)
```

```
print('R2_SCORE:',r2_score(y_test,pred_dtr))
print('Mean_Squared_Error:',mean_squared_error(y_test,pred_dtr))
print('Root Mean_Squared_Error:',np.sqrt(mean_squared_error(y_test,pred_dtr)))
```

```
R2_SCORE: 0.9995971922114072
Mean_Squared_Error: 8257.274444444445
Root Mean_Squared_Error: 90.86954629822053
```

3. RANDOM FOREST REGRESSOR



RandomForestRegressor

```
RFR = RandomForestRegressor()
RFR.fit(X_train,y_train)
```

```
RandomForestRegressor()
```

```
RFR.score(X_train,y_train)
```

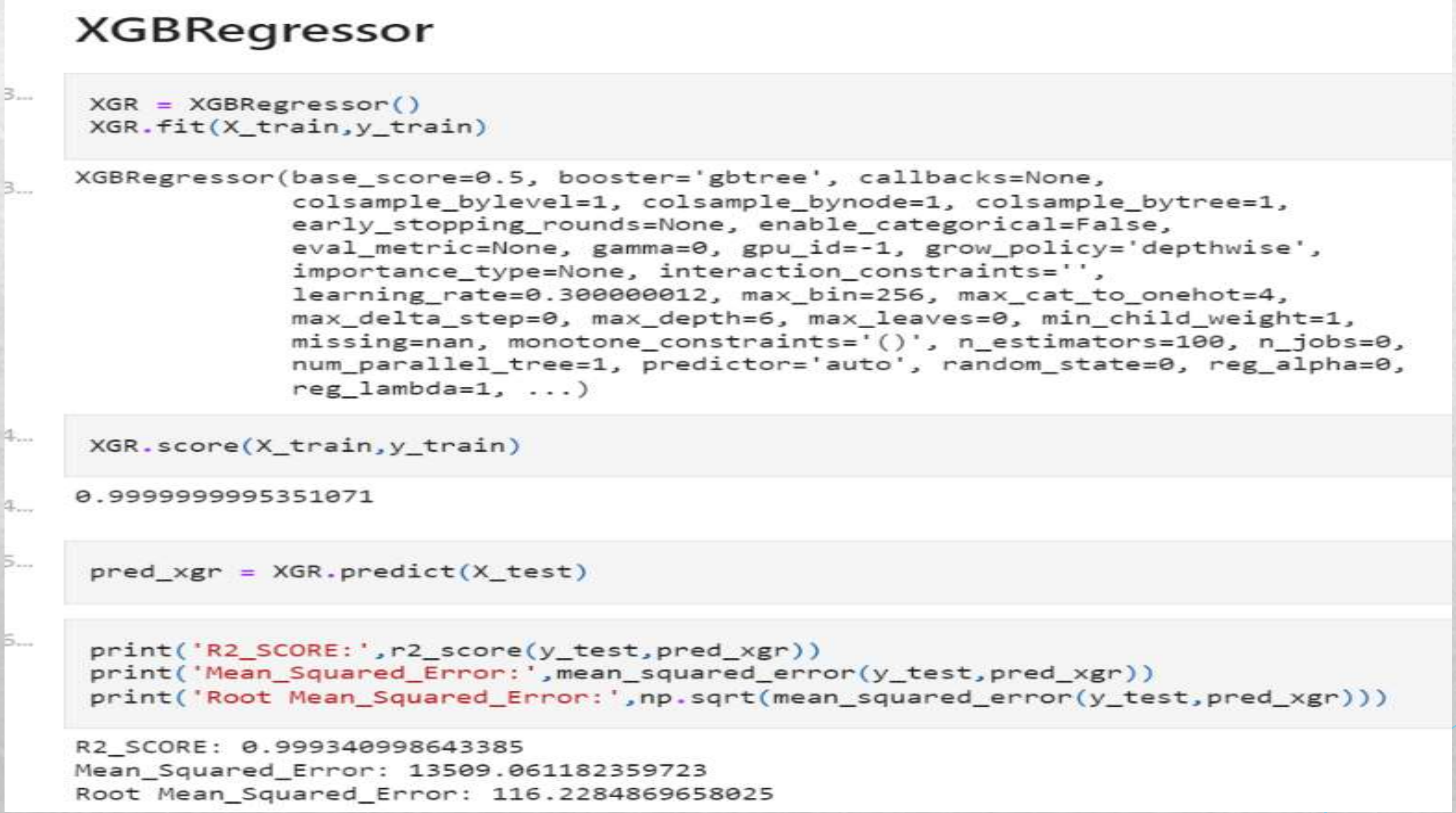
```
0.9996439977513557
```

```
pred_rfr = RFR.predict(X_test)
```

```
print('R2_SCORE:',r2_score(y_test,pred_rfr))
print('Mean_Squared_Error:',mean_squared_error(y_test,pred_rfr))
print('Root Mean_Squared_Error:',np.sqrt(mean_squared_error(y_test,pred_rfr)))
```

```
R2_SCORE: 0.9998614408424088
Mean_Squared_Error: 2840.3646191111116
Root Mean_Squared_Error: 53.29507124595211
```

4. XGB REGRESSOR



XGBRegressor

```
XGR = XGBRegressor()
XGR.fit(X_train,y_train)
```

```
XGBRegressor(base_score=0.5, booster='gbtree', callbacks=None,
             colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
             early_stopping_rounds=None, enable_categorical=False,
             eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
             importance_type=None, interaction_constraints='',
             learning_rate=0.300000012, max_bin=256, max_cat_to_onehot=4,
             max_delta_step=0, max_depth=6, max_leaves=0, min_child_weight=1,
             missing=nan, monotone_constraints='()', n_estimators=100, n_jobs=0,
             num_parallel_tree=1, predictor='auto', random_state=0, reg_alpha=0,
             reg_lambda=1, ...)
```

```
XGR.score(X_train,y_train)
```

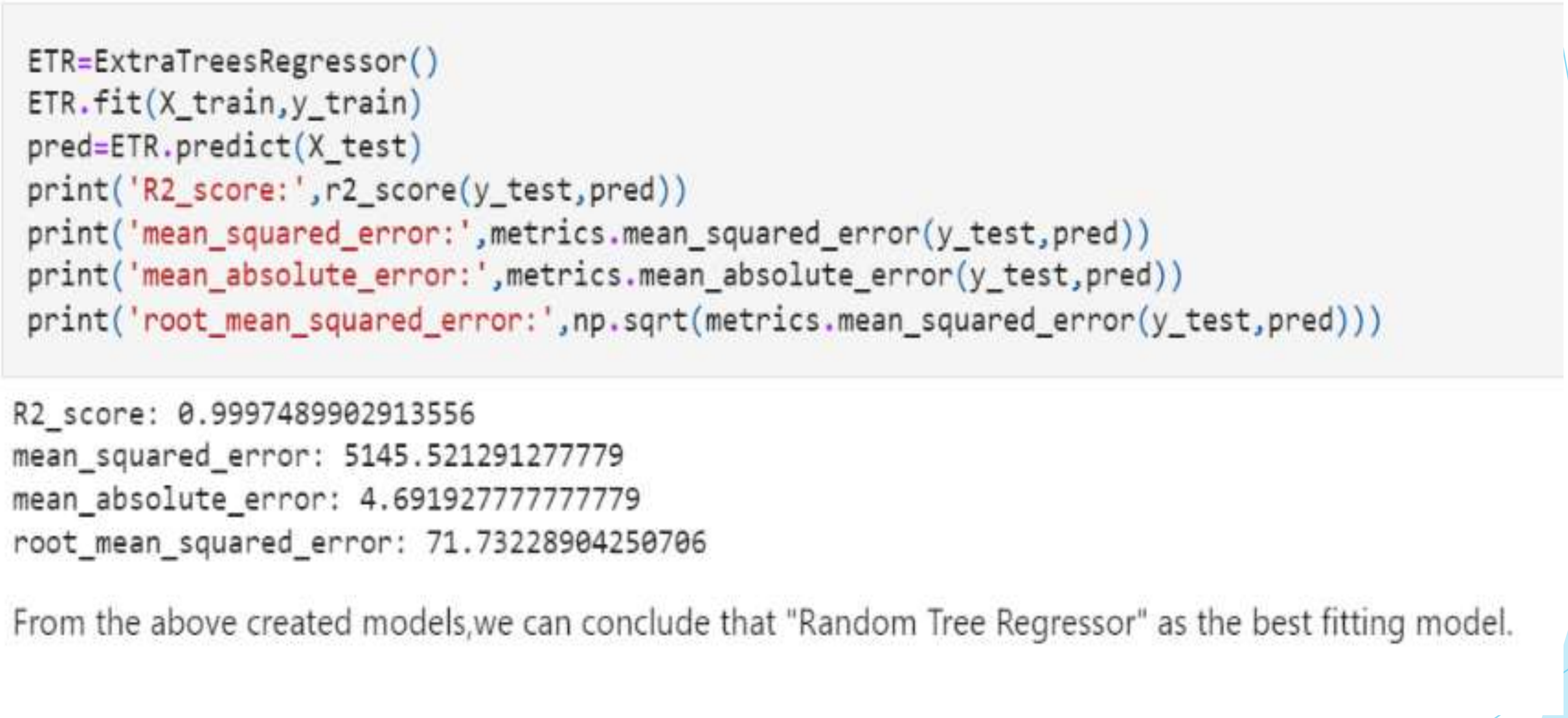
```
0.9999999995351071
```

```
pred_xgr = XGR.predict(X_test)
```

```
print('R2_SCORE:',r2_score(y_test,pred_xgr))
print('Mean_Squared_Error:',mean_squared_error(y_test,pred_xgr))
print('Root Mean_Squared_Error:',np.sqrt(mean_squared_error(y_test,pred_xgr)))
```

```
R2_SCORE: 0.999340998643385
Mean_Squared_Error: 13509.061182359723
Root Mean_Squared_Error: 116.2284869658025
```

5. EXTRATREES REGRESSOR

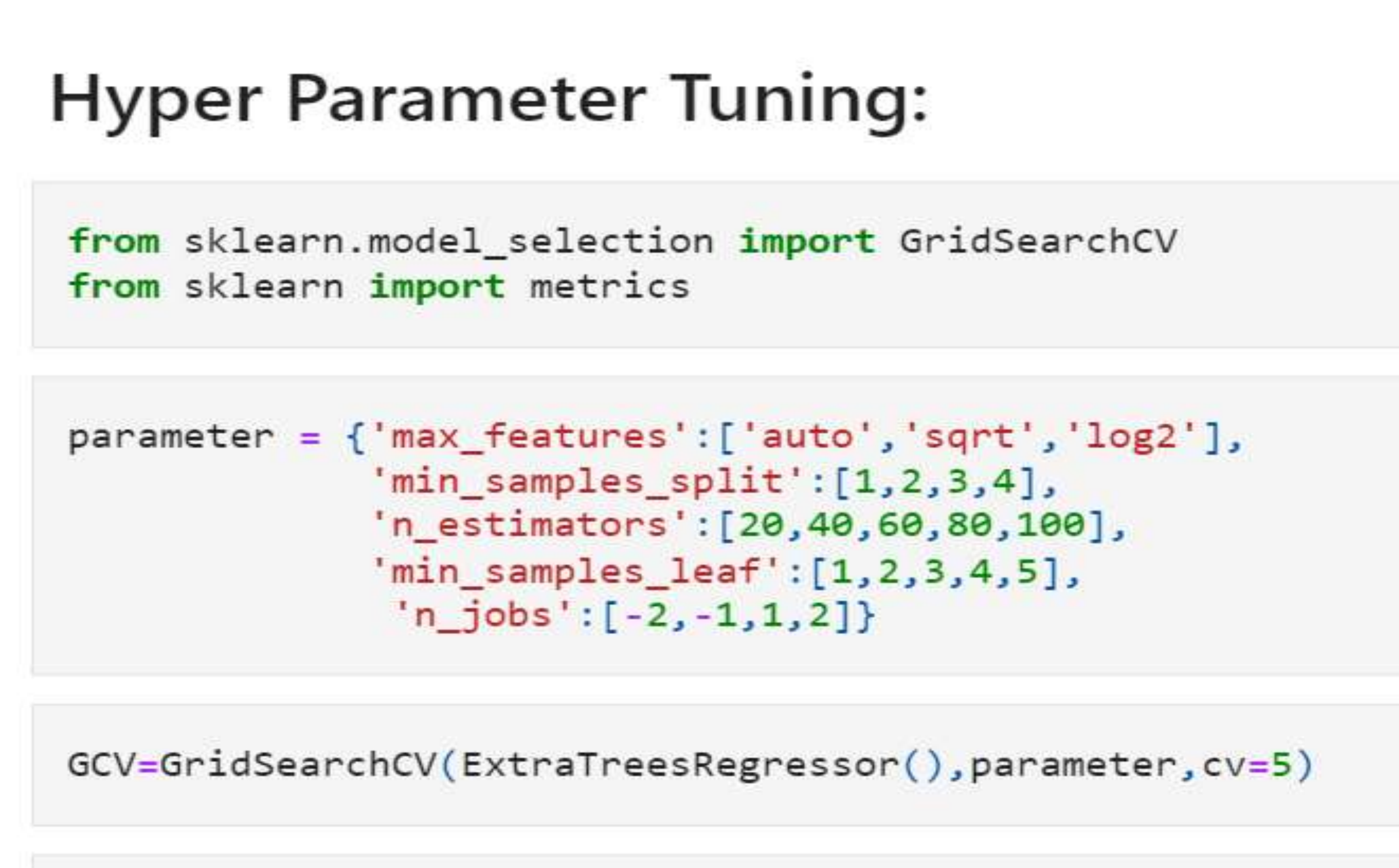


```
ETR=ExtraTreesRegressor()
ETR.fit(X_train,y_train)
pred=ETR.predict(X_test)
print('R2_score:',r2_score(y_test,pred))
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))
```

```
R2_score: 0.9997489902913556
mean_squared_error: 5145.521291277779
mean_absolute_error: 4.691927777777779
root_mean_squared_error: 71.73228904250706
```

From the above created models,we can conclude that "Random Tree Regressor" as the best fitting model.

HYPER PARAMETER TUNNING



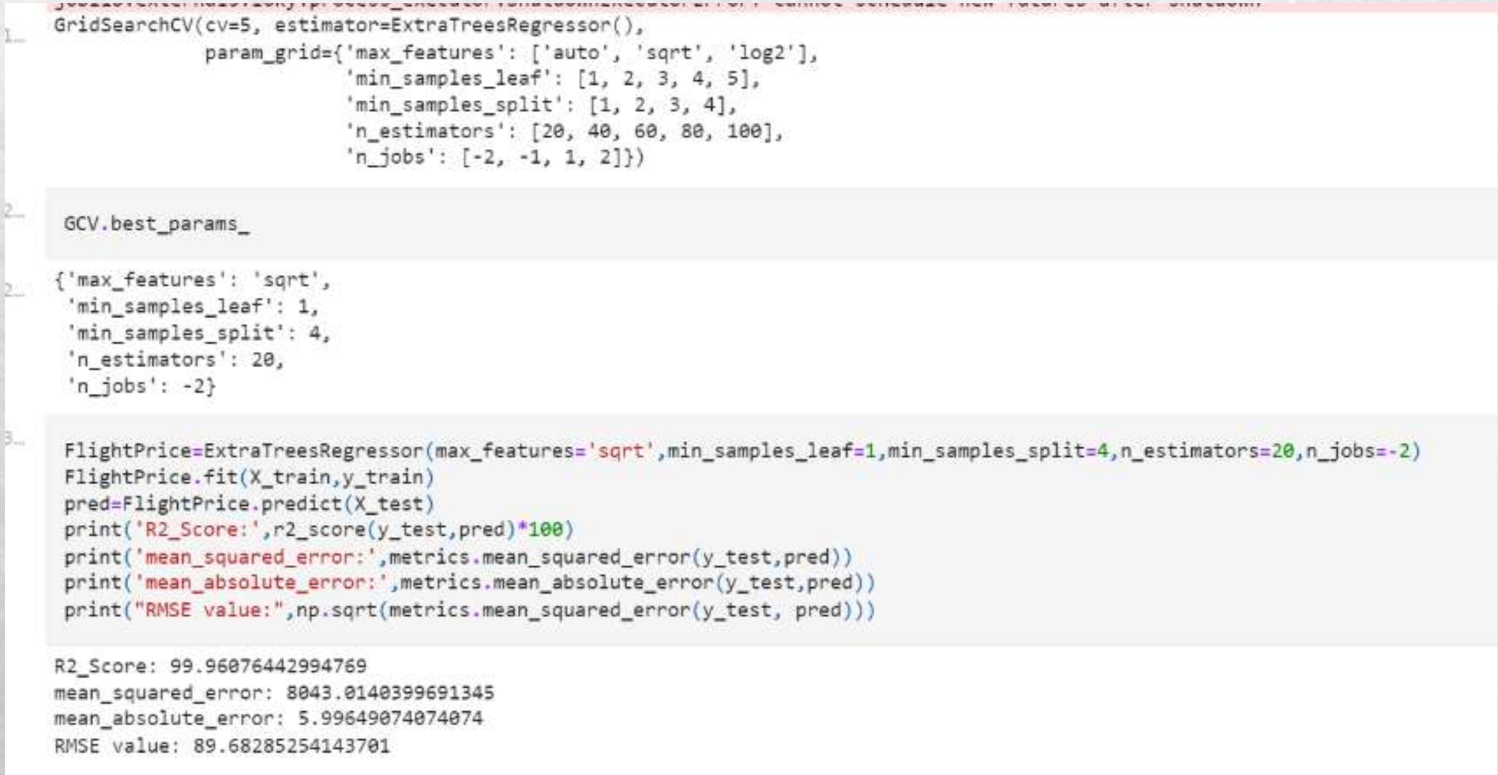
Hyper Parameter Tuning:

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

```
parameter = {'max_features':['auto','sqrt','log2'],
             'min_samples_split':[1,2,3,4],
             'n_estimators':[20,40,60,80,100],
             'min_samples_leaf':[1,2,3,4,5],
              'n_jobs':[-2,-1,1,2]}
```

```
GCV=GridSearchCV(ExtraTreesRegressor(),parameter,cv=5)
```

HYPER PARAMETER TUNNING



```
joblib.externals.loky.process_executor.ShutdownExecutorError: cannot schedule new futures after shutdown
GridSearchCV(cv=5, estimator=ExtraTreesRegressor(),
             param_grid={'max_features': ['auto', 'sqrt', 'log2'],
                         'min_samples_leaf': [1, 2, 3, 4, 5],
                         'min_samples_split': [1, 2, 3, 4],
                         'n_estimators': [20, 40, 60, 80, 100],
                         'n_jobs': [-2, -1, 1, 2]})
```

```
GCV.best_params_
```

```
{'max_features': 'sqrt',
 'min_samples_leaf': 1,
 'min_samples_split': 4,
 'n_estimators': 20,
 'n_jobs': -2}
```

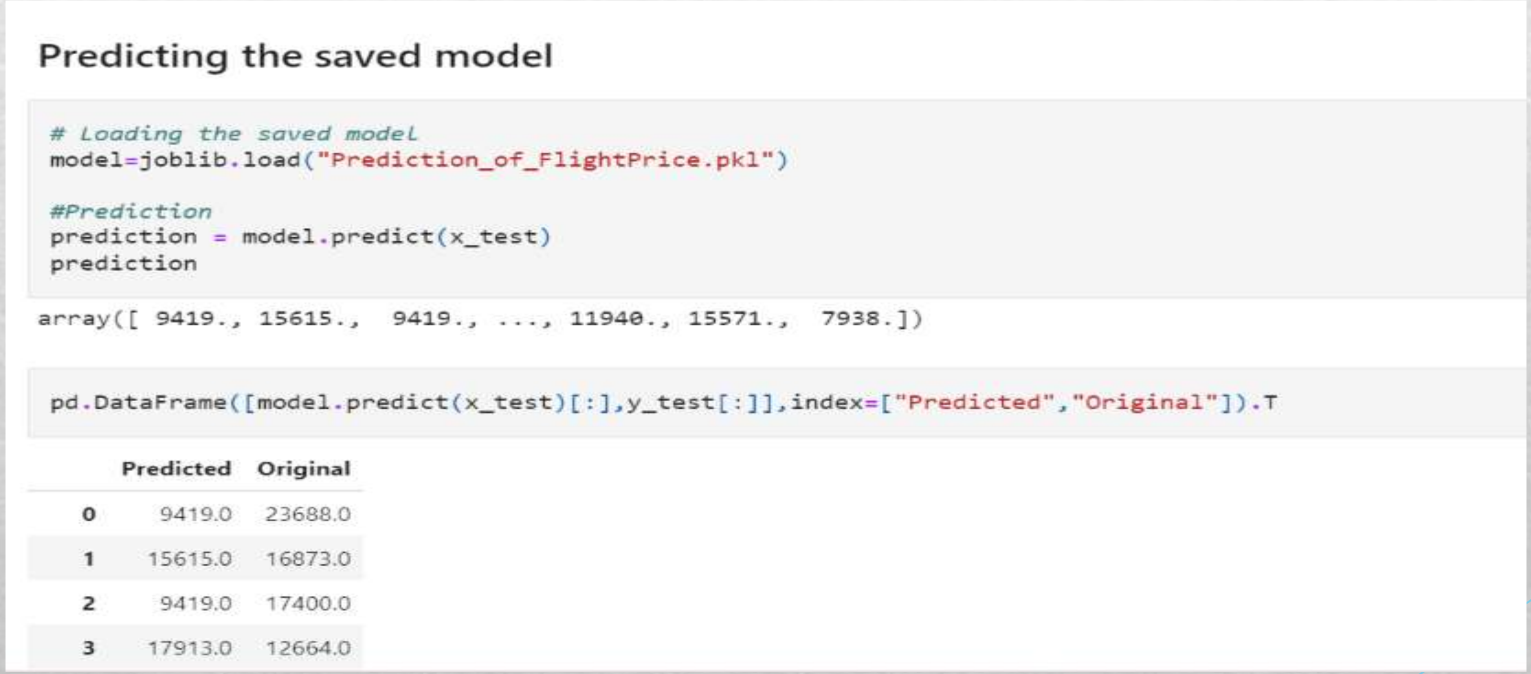
```
FlightPrice=ExtraTreesRegressor(max_features='sqrt',min_samples_leaf=1,min_samples_split=4,n_estimators=20,n_jobs=-2)
FlightPrice.fit(X_train,y_train)
pred=FlightPrice.predict(X_test)
print('R2_Score:',r2_score(y_test,pred)*100)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print("RMSE value:",np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
R2_Score: 99.96076442994769
mean_squared_error: 8043.0140399691345
mean_absolute_error: 5.99649074074074
RMSE value: 89.68285254143701
```

- **I have chosen all parameters of <u>ExtraTreesRegressor</u>, after tunning the model with best parameters model accuracy remain same as 99.96%.**

SAVING THE MODEL AND PREDICTIONS

- ❑ I have saved my best model using .pkl as follows.
- ❑ Now after saving the best model, loading my saved model and predicting the test values.
- ❑ I have predicted the Price for test dataset using saved model of train dataset, and the predictions look good. I have also saved my predictions for further analysis.



Predicting the saved model

```
# Loading the saved model
model=joblib.load("Prediction_of_FlightPrice.pkl")

#Prediction
prediction = model.predict(x_test)
prediction
```

```
array([ 9419., 15615.,  9419., ..., 11940., 15571.,  7938.])
```

```
pd.DataFrame([model.predict(x_test)[:],y_test[:]],index=["Predicted","Original"]).T
```

	Predicted	Original
0	9419.0	23688.0
1	15615.0	16873.0
2	9419.0	17400.0
3	17913.0	12664.0

CONCLUSION

- ❑ In this project report, we have used machine learning algorithms to predict the flight 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. The power of visualization has helped us in understanding the data by graphical representation it has made me to understand what data is trying to say.
- ❑ Data cleaning is one of the most important steps to remove unrealistic values and unnecessary values.
- ❑ These feature set were then given as an input to seven 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 flight price. It was good that the predicted and actual values were almost same.
- ❑ To conclude, the application of machine learning in flight price prediction is still at an early stage. We hope this study has moved a small step ahead in providing some methodological and empirical contributions to online platforms, and presenting an alternative approach to the valuation of flight price.
- ❑ Future direction of research may consider incorporating additional flight data from a larger economical background with more features.

THANK YOU!!