FLIP ROBO

Flight Price Prediction Model

Submitted by:

NIHAL SINGH

ACKNOWLEDGMENT

This Project includes the details about Flight Price Prediction for different dates. The flights data has been scrap from multiple flight booking web sites like easemytrip, yatra and other websites and the data is taken from multiple dates and multiple cities.

The flight data has been scrap using selenium web scrapping method.

The flight price prediction model is a regression model as the target column is having continuous numerical data.

INTRODUCTION

- Business Problem Framing

  The flight price prediction model is basically predict the price of flight price depend on multiple features.

- Conceptual Background of the Domain Problem

  The model is basically related to the predict the flight price between multiple cities based on their timings, dates, numbers of stops and cities.

- Review of Literature

  This model is basically build by first do the data scrapping using the automated selenium tool then after that building the model using the scrapped data.

  The Data has been fetched from the multiple flight booking web sites like yatra, MMT, easemytrip etc. The total data scraped is approx. 1.5K and using that scrapped data model has been build.

- Motivation for the Problem Undertaken

  The main challenge to build this model is scrapping the data from the multiple web sites. Fetching the 1.5 K data from the websites is basically takes lots of efforts.

  After that analysing the data and remove the unwanted data from the scrapped data and then building the model on the filtered data.

Analytical Problem Framing

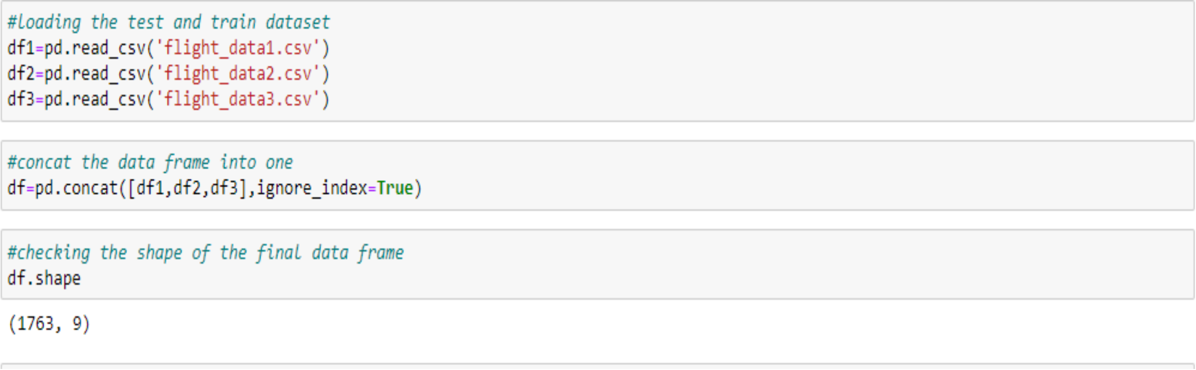
- Mathematical/ Analytical Modeling of the Problem

The Dataset is in csv format, which consist of multiple data frame form the scrapped data from web sites. Merging all the data frame into one data frame and Performing all the data analysis and EDA on the merged data frame and building the model on the merged data. Plotting the multiple plots like bar plot, dist plot, box plot, heat map, cat plot to analyse the data and which features are important and related to the target and delete the non-important features. Using z-score method to find the outliers and deleting it. Scaling the data to build model.

- Data Sources and their formats

The Dataset is in csv format which consist of multiple data frame in different csv files so merging both the file into single data frame to perform the analysis and then build model building.

The data having the different datatypes i.e. int and object.



```
#loading the test and train dataset
df1=pd.read_csv('flight_data1.csv')
df2=pd.read_csv('flight_data2.csv')
df3=pd.read_csv('flight_data3.csv')
```

```
#concat the data frame into one
df=pd.concat([df1,df2,df3],ignore_index=True)
```

```
#checking the shape of the final data frame
df.shape
```

```
(1763, 9)
```

- Data Pre-processing Done

The Data Pre-processing consist of below mentioned points:

1. Merge the multiple data frame from the scraped data for the Data pre-processing.
2. Checking the null values form the dataset, if null value is available, and then deleting the null values from complete data frame.
3. Deleting the data having the '_' values from the data frame.
4. Describing the data.
5. Using the encoding method to encode the categorical data into the numerical data.
6. Checking all the features and checking the impact with target if no impact is there then deleting the features.
7. Plotting the multiple plots like cat plot, dist plot, box plot to check the data distribution and outliers from the datasets.
8. Using IQR method to check the outliers and remove the outliers from the data frame.
9. Checking the skewness after removing the outliers from the dataset and skewness is present in the features so using the power transform method to remove the skewness.
10. Using the heat map and variance inflation factor to check the multicollliniarity issue between the features.
11. If issue is there then delete the feature which having such issue.
12. Scaling the data using the standard scalar method.
13. Separating the features and target to build the model.

- Hardware and Software Requirements and Tools Used

The Tool used to build the model is anaconda jupyter and selenium web driver.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

Multiple problem faced during the model building. All are listed bellows.

1. Scrapping the data from the websites for model building.
2. Checking for the unwanted columns analyse the relation with target and removal of that features.
3. Checking the null values and delete it.
4. Checking of unwanted data and delete it.
5. Managing the categorical columns using multiple methods.
6. Removal of outliers form the features using the IQR.
7. Removal of skewness using the power transform function.
8. Removal of multicolliniarity issue from the features.

- Testing of Identified Approaches (Algorithms)

This is a Regression problem as the target column has continuous data so multiple regression model has been used for the model building and basically training and testing the data on multiple models to select the best model.

The multiple models are used as given below:

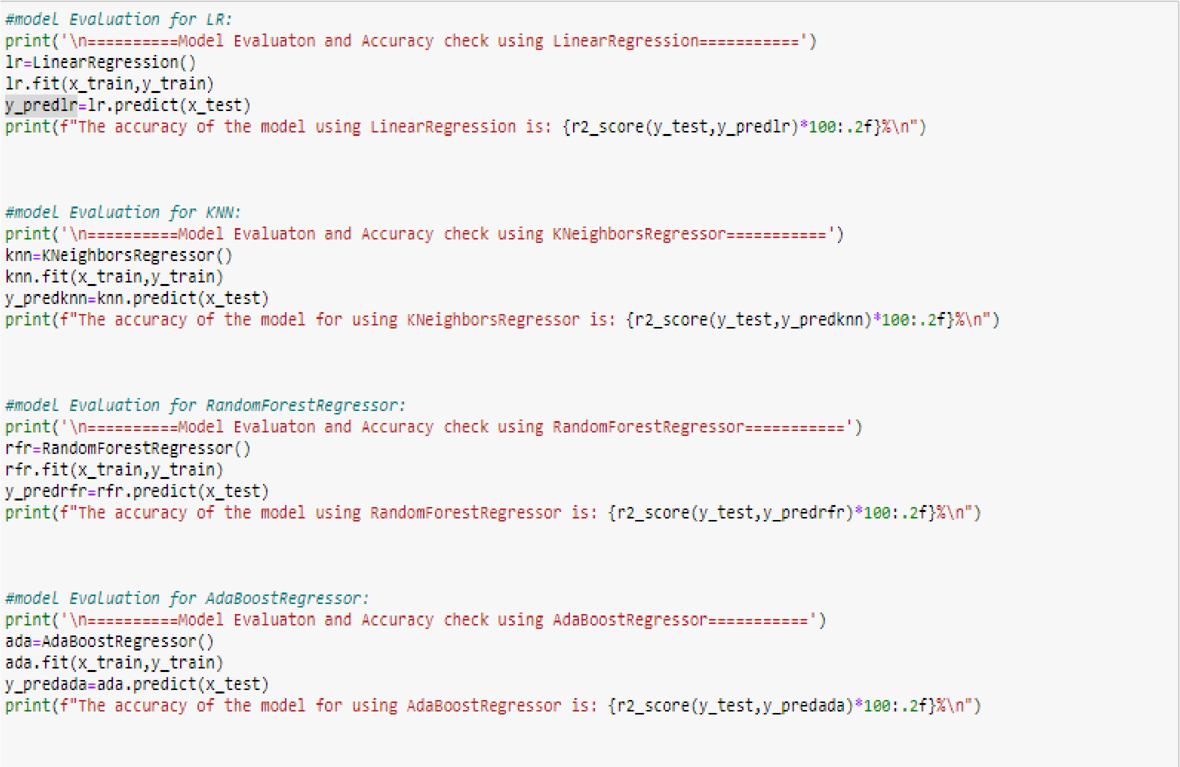
1. Linear Regression
2. KNeighbors Regressor
3. Random Forest Regressor
4. AdaBoost Regressor
5. SVR

- Run and Evaluate selected models

  The model is a classification model so all the classification model building algorithm has been used to build the model and across all the model the best one has been selected as final model.

  The different models which has been used has been listed below:

1. Linear Regression : Linear Regression is a supervised machine learning algorithm where the predicted output is continuous. It's used to predict values within a continuous range.

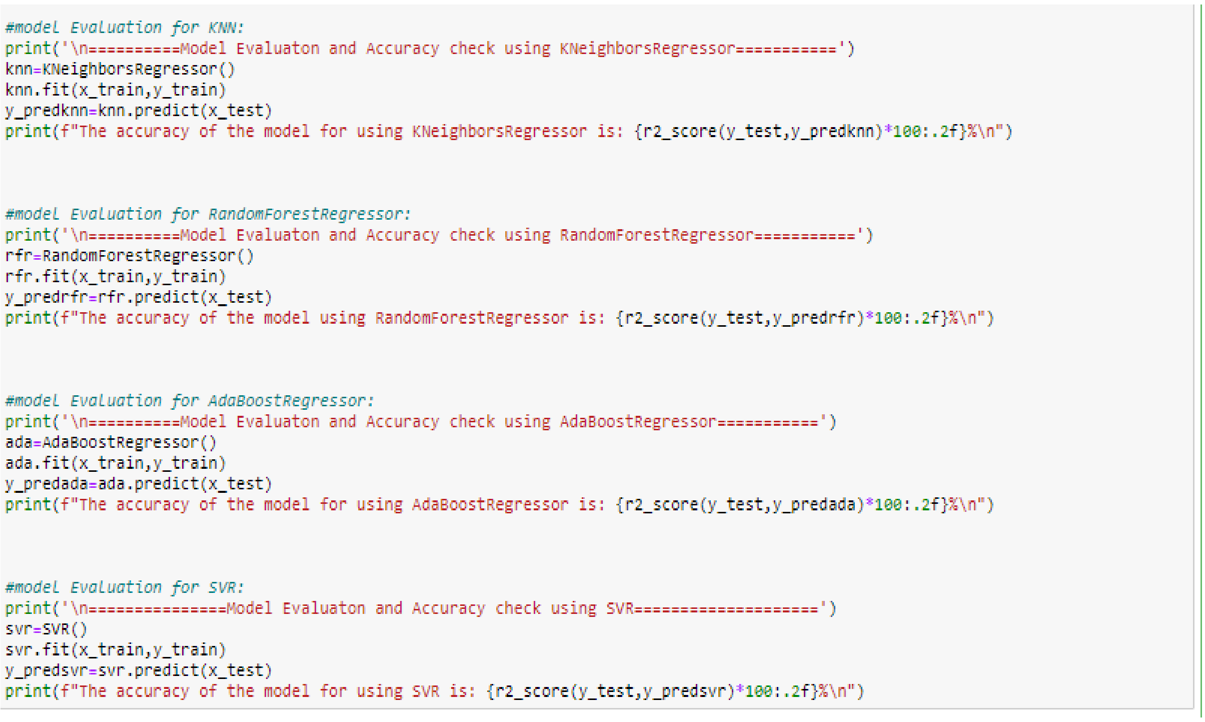


```
#model Evaluation for LR:
print('\n==========Model Evaluaton and Accuracy check using LinearRegression===========')
lr=LinearRegression()
lr.fit(x_train,y_train)
y_predlr=lr.predict(x_test)
print(f"The accuracy of the model using LinearRegression is: {r2_score(y_test,y_predlr)*100:.2f}%\n")


#model Evaluation for KNN:
print('\n==========Model Evaluaton and Accuracy check using KNeighborsRegressor===========')
knn=KNeighborsRegressor()
knn.fit(x_train,y_train)
y_predknn=knn.predict(x_test)
print(f"The accuracy of the model for using KNeighborsRegressor is: {r2_score(y_test,y_predknn)*100:.2f}%\n")


#model Evaluation for RandomForestRegressor:
print('\n==========Model Evaluaton and Accuracy check using RandomForestRegressor===========')
rfr=RandomForestRegressor()
rfr.fit(x_train,y_train)
y_predrfr=rfr.predict(x_test)
print(f"The accuracy of the model using RandomForestRegressor is: {r2_score(y_test,y_predrfr)*100:.2f}%\n")


#model Evaluation for AdaBoostRegressor:
print('\n==========Model Evaluaton and Accuracy check using AdaBoostRegressor===========')
ada=AdaBoostRegressor()
ada.fit(x_train,y_train)
y_predada=ada.predict(x_test)
print(f"The accuracy of the model for using AdaBoostRegressor is: {r2_score(y_test,y_predada)*100:.2f}%\n")
```

   After training the data in Linear Regression the r2 score of the model is 48.78%.

   The cross validation score for the Linear Regression model is -93460588.06%

2. SVR : Support Vector Regression is a supervised learning algorithm that is used to predict discrete values. Support Vector Regression uses the same principle as the SVMs. The basic idea behind SVR is to find the best fit line. In SVR, the best fit line is the hyperplane that has the maximum number of points.

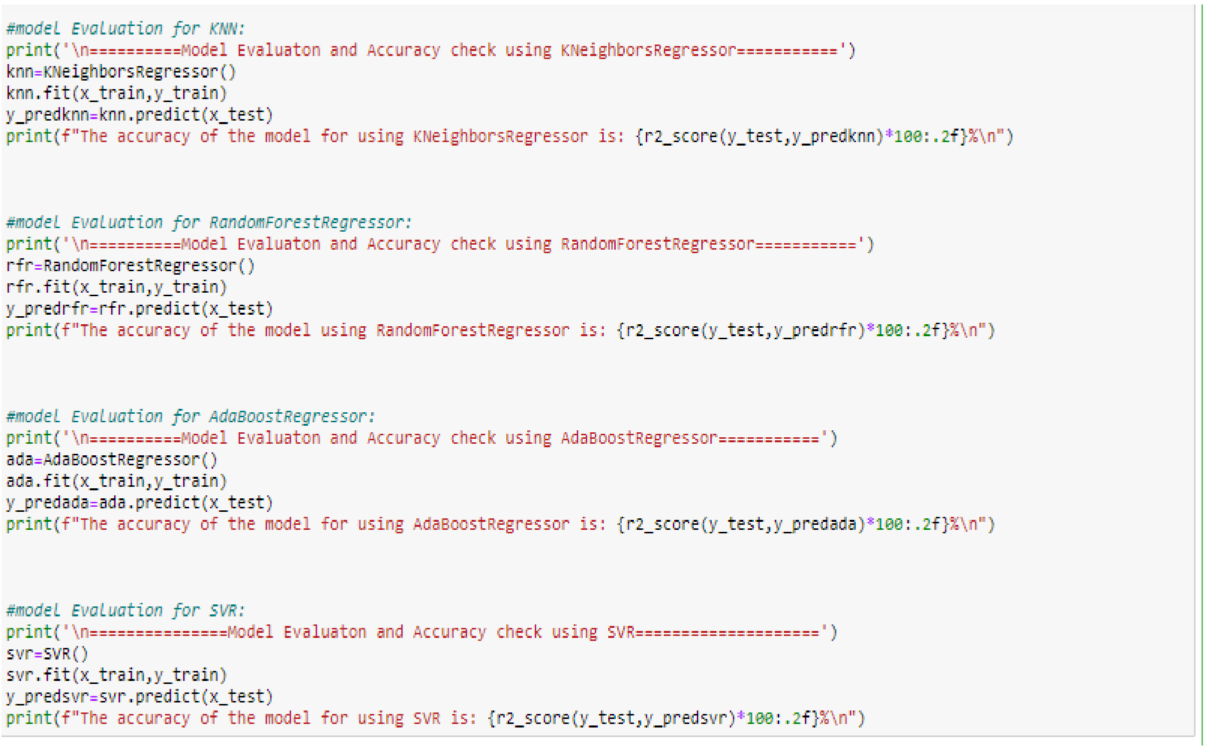


```
#model Evaluation for KNN:
print('\n==========Model Evaluaton and Accuracy check using KNeighborsRegressor===========')
knn=KNeighborsRegressor()
knn.fit(x_train,y_train)
y_predknn=knn.predict(x_test)
print(f"The accuracy of the model for using KNeighborsRegressor is: {r2_score(y_test,y_predknn)*100:.2f}%\n")


#model Evaluation for RandomForestRegressor:
print('\n==========Model Evaluaton and Accuracy check using RandomForestRegressor===========')
rfr=RandomForestRegressor()
rfr.fit(x_train,y_train)
y_predrfr=rfr.predict(x_test)
print(f"The accuracy of the model using RandomForestRegressor is: {r2_score(y_test,y_predrfr)*100:.2f}%\n")


#model Evaluation for AdaBoostRegressor:
print('\n==========Model Evaluaton and Accuracy check using AdaBoostRegressor===========')
ada=AdaBoostRegressor()
ada.fit(x_train,y_train)
y_predada=ada.predict(x_test)
print(f"The accuracy of the model for using AdaBoostRegressor is: {r2_score(y_test,y_predada)*100:.2f}%\n")


#model Evaluation for SVR:
print('\n===============Model Evaluaton and Accuracy check using SVR====================')
svr=SVR()
svr.fit(x_train,y_train)
y_predsvr=svr.predict(x_test)
print(f"The accuracy of the model for using SVR is: {r2_score(y_test,y_predsvr)*100:.2f}%\n")
```

After training the data in SVR the r2 score of the model is -5.56%.

The cross validation score for the SVR Regression model is -29.07%

3. Ada boost Regressor: AdaBoost Regressor, short for Adaptive Boosting, is a Boosting technique used as an Ensemble Method in Machine Learning. It combines multiple classifiers to increase the accuracy of classifiers. AdaBoost is an iterative ensemble method.

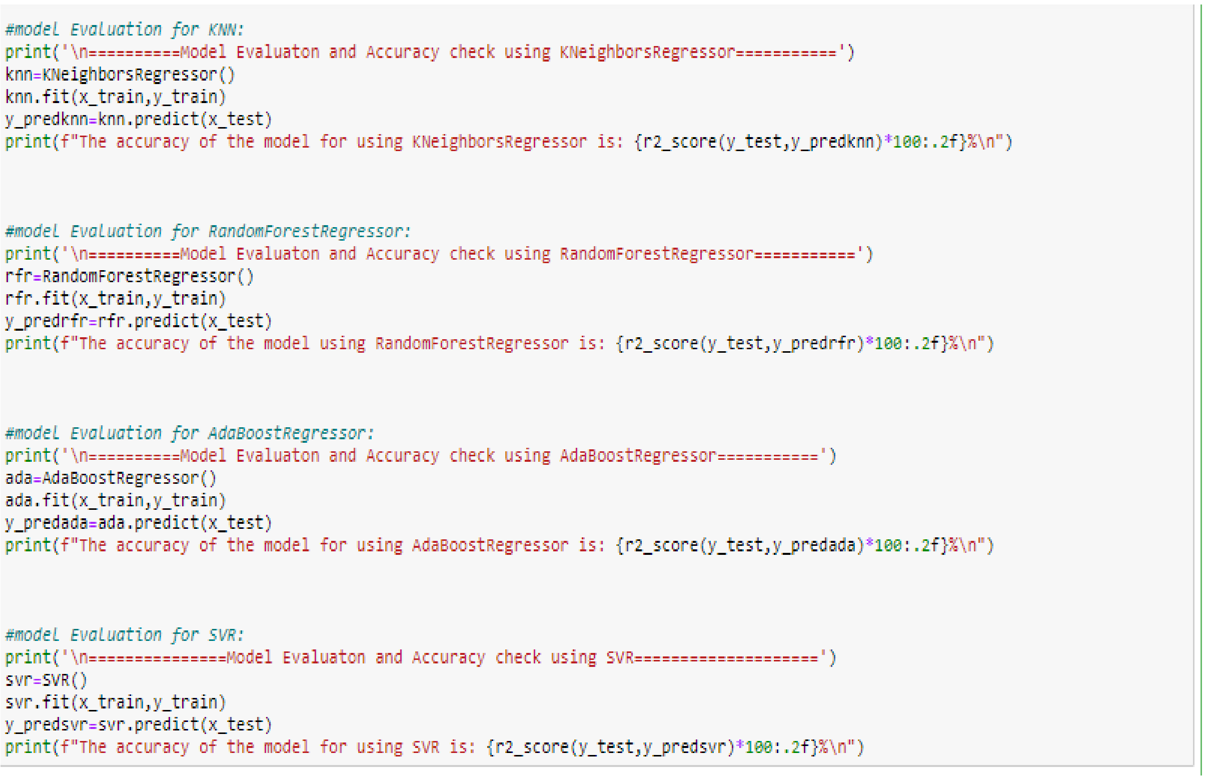


```
#model Evaluation for KNN:
print('\n==========Model Evaluaton and Accuracy check using KNeighborsRegressor===========')
knn=KNeighborsRegressor()
knn.fit(x_train,y_train)
y_predknn=knn.predict(x_test)
print(f"The accuracy of the model for using KNeighborsRegressor is: {r2_score(y_test,y_predknn)*100:.2f}%\n")


#model Evaluation for RandomForestRegressor:
print('\n==========Model Evaluaton and Accuracy check using RandomForestRegressor===========')
rfr=RandomForestRegressor()
rfr.fit(x_train,y_train)
y_predrfr=rfr.predict(x_test)
print(f"The accuracy of the model using RandomForestRegressor is: {r2_score(y_test,y_predrfr)*100:.2f}%\n")


#model Evaluation for AdaBoostRegressor:
print('\n==========Model Evaluaton and Accuracy check using AdaBoostRegressor===========')
ada=AdaBoostRegressor()
ada.fit(x_train,y_train)
y_predada=ada.predict(x_test)
print(f"The accuracy of the model for using AdaBoostRegressor is: {r2_score(y_test,y_predada)*100:.2f}%\n")


#model Evaluation for SVR:
print('\n===============Model Evaluaton and Accuracy check using SVR====================')
svr=SVR()
svr.fit(x_train,y_train)
y_predsvr=svr.predict(x_test)
print(f"The accuracy of the model for using SVR is: {r2_score(y_test,y_predsvr)*100:.2f}%\n")
```

After training the data in Ada boost Regressor the r2 score of the model is 53.41%.

The cross validation score for the Ada boost Regression model is 26.71%

4. Random forest Regressor: Random forests is a supervised learning algorithm. Random forests creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting.

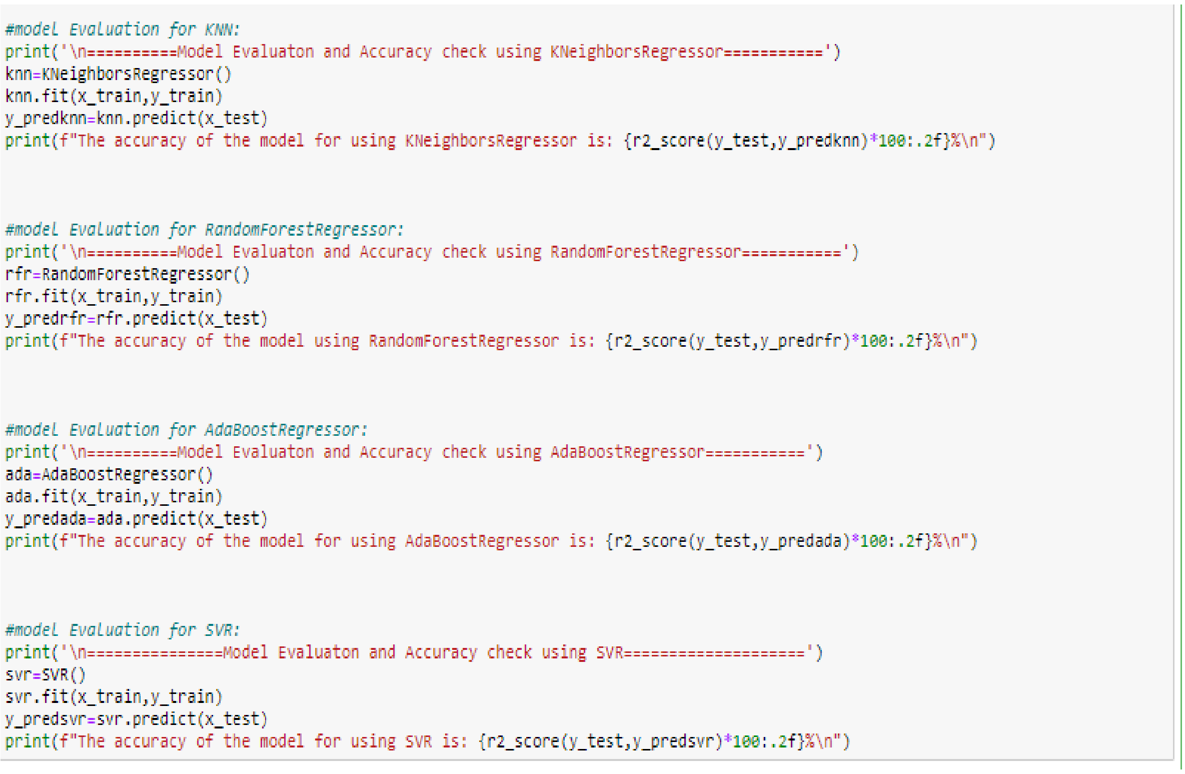


```
#model Evaluation for KNN:
print('\n==========Model Evaluaton and Accuracy check using KNeighborsRegressor===========')
knn=KNeighborsRegressor()
knn.fit(x_train,y_train)
y_predknn=knn.predict(x_test)
print(f"The accuracy of the model for using KNeighborsRegressor is: {r2_score(y_test,y_predknn)*100:.2f}%\n")


#model Evaluation for RandomForestRegressor:
print('\n==========Model Evaluaton and Accuracy check using RandomForestRegressor===========')
rfr=RandomForestRegressor()
rfr.fit(x_train,y_train)
y_predrfr=rfr.predict(x_test)
print(f"The accuracy of the model using RandomForestRegressor is: {r2_score(y_test,y_predrfr)*100:.2f}%\n")


#model Evaluation for AdaBoostRegressor:
print('\n==========Model Evaluaton and Accuracy check using AdaBoostRegressor===========')
ada=AdaBoostRegressor()
ada.fit(x_train,y_train)
y_predada=ada.predict(x_test)
print(f"The accuracy of the model for using AdaBoostRegressor is: {r2_score(y_test,y_predada)*100:.2f}%\n")


#model Evaluation for SVR:
print('\n===============Model Evaluaton and Accuracy check using SVR====================')
svr=SVR()
svr.fit(x_train,y_train)
y_predsvr=svr.predict(x_test)
print(f"The accuracy of the model for using SVR is: {r2_score(y_test,y_predsvr)*100:.2f}%\n")
```

After training the data in Random forest Regressor the r2 score of the model is 85.87%.

The cross validation score for the Random forest Regression model is 46.05%

5. KNN Regressor: KNN also known as K-nearest neighbour is a supervised and pattern classification learning algorithm which helps us find which class the new input(test value) belongs to when k nearest neighbours are chosen and distance is calculated between them.



```
#model Evaluation for KNN:
print('\n==========Model Evaluaton and Accuracy check using KNeighborsRegressor===========')
knn=KNeighborsRegressor()
knn.fit(x_train,y_train)
y_predknn=knn.predict(x_test)
print(f"The accuracy of the model for using KNeighborsRegressor is: {r2_score(y_test,y_predknn)*100:.2f}%\n")



#model Evaluation for RandomForestRegressor:
print('\n==========Model Evaluaton and Accuracy check using RandomForestRegressor===========')
rfr=RandomForestRegressor()
rfr.fit(x_train,y_train)
y_predrfr=rfr.predict(x_test)
print(f"The accuracy of the model using RandomForestRegressor is: {r2_score(y_test,y_predrfr)*100:.2f}%\n")



#model Evaluation for AdaBoostRegressor:
print('\n==========Model Evaluaton and Accuracy check using AdaBoostRegressor===========')
ada=AdaBoostRegressor()
ada.fit(x_train,y_train)
y_predada=ada.predict(x_test)
print(f"The accuracy of the model for using AdaBoostRegressor is: {r2_score(y_test,y_predada)*100:.2f}%\n")



#model Evaluation for SVR:
print('\n==============Model Evaluaton and Accuracy check using SVR====================')
svr=SVR()
svr.fit(x_train,y_train)
y_predsvr=svr.predict(x_test)
print(f"The accuracy of the model for using SVR is: {r2_score(y_test,y_predsvr)*100:.2f}%\n")
```

After training the data in KNN Regressor the r2 score of the model is 68.72%.

The cross validation score for the KNN Regression model is 55.65%

- Key Metrics for success in solving problem under consideration

  There are multiple points which impact the final outcome for the model.

  1. Scrapping the data from web sites.
  2. Deleting the non related features from the data frame, which is not shows impact on final model.
  3. Checking the multicolliniarity issue and removing it by deleting the features from the data frame.
  4. Checking the skewness and removing the skewness using the power transform method.
  5. Performing the complete EDA and visualization on the merged data frame then perform model building on the data set .

- Visualizations

  Visualization is basically finding some outcomes after visualizing the data in form of some graph or some plots. Different visualizing methods has been use the to do the analysis on the data. In this model multiple plots has been used to do analysis on the provided data.

  1. Bar Plot – To check the relation between feature and target.
  2. Dist plot – To check the data distribution and checking the linearity of data distribution.
  3. Heat Map – To check the null count and multicolliniarity issue between the features.
  4. Cat Plot – To check the relationship of the target variable with the multiple features.
  5. Box Plot – To check the outliers in the features.

- Interpretation of the Results

  The flight Price Prediction Model is used to do the prediction on the flight price based on their multiple features available.

CONCLUSION

- Key Findings and Conclusions of the Study

  Main finding in the models are the provided features are relevant to the model building, data analysing and data filtering then converting it to binary data. The other finding is to scrap the data for the model building using selenium.

- Learning Outcomes of the Study in respect of Data Science

  There are multiple features available in the data frame but all are not useful with respect to the final outcome. Many of the unwanted features and all the features has the outliers which has been taken care of using the z method. There are multicolliniarity issue is also available in the features that is also managed by deleting the features having multicolliniarity issues. The model is built after merging the different data sets scrap from the multiple websites.