FLIP ROBO

CUSTOMER RETENTION DATASET

Submitted by:

KARISHMA YADAV.

ACKNOWLEDGMENT

I have taken efforts in this project. Customer satisfaction has emerged as one of the most important factors that guarantee the success of online store; it has been posited as a key stimulant of purchase, repurchase intentions and customer loyalty. A comprehensive review of the literature, theories and models have been carried out to propose the models for customer activation and customer retention.

INTRODUCTION

Customer satisfaction has emerged as one of the most important factors that guarantee the success of online store; it has been posited as a key stimulant of purchase, repurchase intentions and customer loyalty. A comprehensive review of the literature, theories and models have been carried out to propose the models for customer activation and customer retention. Five major factors that contributed to the success of an e-commerce store have been identified as: service quality, system quality, information quality, trust and net benefit. The research furthermore investigated the factors that influence the online customers repeat purchase intention. The combination of both utilitarian value and hedonistic values are needed to affect the repeat purchase intention (loyalty) positively. The data is collected from the Indian online shoppers. Results indicate the e-retail success factors, which are very much critical for customer satisfaction.

Customer retention includes all actions taken by organization to guarantee customer loyalty and reduce customer. Different statistical and machine-learning techniques are used to address this problem. Machine-learning techniques have been widely used for evaluating the probability of customer.

1) Regression analysis: Regression analysis techniques aim mainly to investigate and estimate the relationships among a set of features. Regression includes many models for analysing the relation between one target/response variable and a set of

independent variables. Logistic Regression (LR) is the appropriate regression analysis model to use when the dependent variable is binary. LR is a predictive analysis used to explain the relationship between a dependent binary variable and a set of independent variables. For customer churn, LR has been widely used to evaluate the churn probability as a function of a set of variables or customers' features.

2) Decision Tree: Decision Tree (DT) is a model that generates a tree-like structure that represents set of decisions. DT returns the probability scores of class membership. DT is composed of: a) internal Nodes: each node refers to a single variable/feature and represents a test point at feature level; b) branches, which represent the outcome of the test and are represented by lines that finally lead to c) leaf Nodes which represent the class labels. That is how decision rules are established and used to classify new instances. DT is a flexible model that supports both categorical and continuous data. Due to their flexibility they gained popularity and became one of the most commonly used models for prediction.

3) Support Vector Machine: Support Vector Machine (SVM) is a supervised learning technique that performs data analysis in order to identify patterns. Given a set of labelled training data, SVM represents observations as points in a high dimensional space and tries to identify the best separating hyper planes between instances of different classes. New instances are represented in the same space and are classified to a specific class based on their proximity to the separating gap. For churn

prediction, SVM techniques have been widely investigated and evaluated to be of high predictive performance.

4) Random Forest Random forests (RF) are an ensemble learning technique that can support classification and regression. It extends the basic idea of single classification tree by growing many classification trees in the training phase. To classify an instance, each tree in the forest generates its response (vote for a class), the model choses the class that has receive the most votes over all the trees in the forest. One major advantage of RF over traditional decision trees is the protection against over fitting which makes the model able to deliver a high performance.

The first step before applying the selected analytical models on the dataset, explanatory data analysis for more insights into dataset was performed. Based on the observations data was pre-processed to be more suitable for analysis.

Analytical Problem Framing

Data pre-processing:

Pre-processing includes three steps:

a) Data transformation,

b) Data cleaning and

c) Feature selection.

Data Transformation :

Two of the explanatory variables were transformed from binominal form (yes/no) into binary form (1/0) to be more suitable for the selected models.

Data cleaning :

This stage includes missing data handling/imputation: Some of the selected algorithms cannot handle missing data such as SVM. That’s why missing value can be replaced by mean, median or zero. However, missing data replacement by statistically computed value (imputation) is a better option. The used dataset included missing values in some the numerical variables. Numerical data were replaced using random forest imputation technique And binary values were imputed using the techniques.

Feature selection :

Before model training, feature selection is one of the most important factors that can affect the performance of models. In this study, the importance of the used variables was measured to identify and rank explanatory variables influence on the target/response. This allows dimensionality reduction by removing variables/predictors with low influence on the target. Random forest technique can be used for feature selection using mean decrease accuracy. Mean decrease measures the impact of each feature on model accuracy. The model permutes values of each feature and evaluates model accuracy change. Only features having higher impact on accuracy are considered important.

- Hardware and Software Requirements and Tools Used

RAM: 8GB

ROM: I3 processor

SOFTWARE: Python 3.9.6

LIBRARIES: Numpy, Pandas, Seaborn, Sklearn, Scipy .

Tools: Jupiter notebook, Matplotlib, Scikit-learn, Excel.

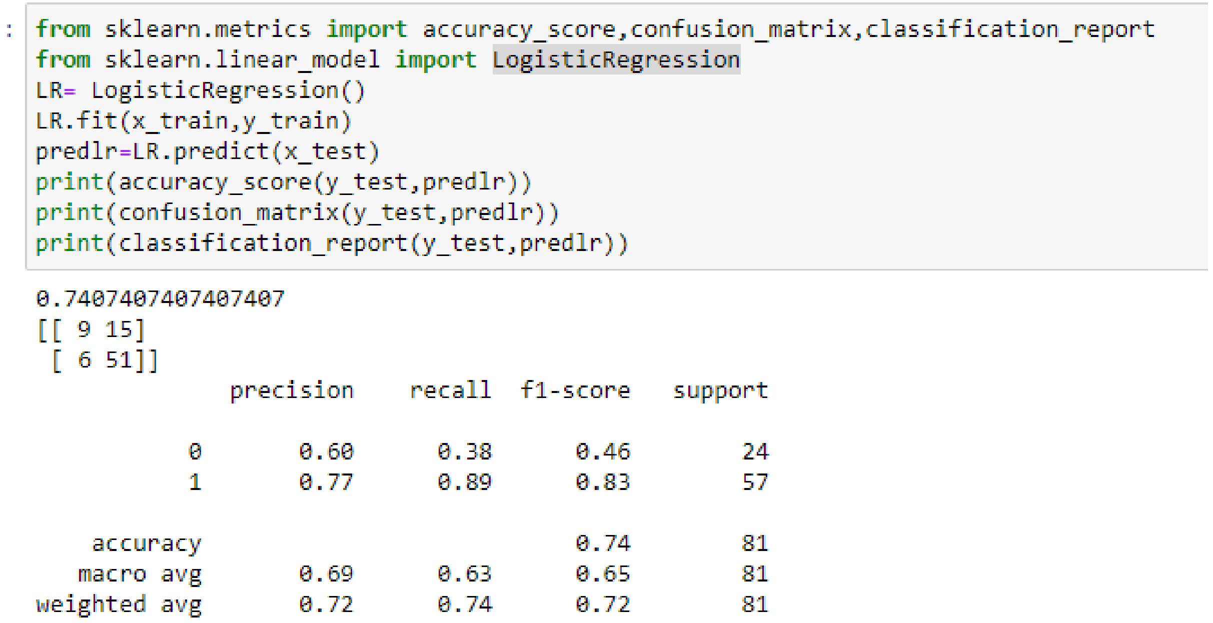
- Testing of Identified Approaches (Algorithms)

1) Decision Tree Classifier
2) Logistic Regression
3) Random Forest Classifier
4) SVC
5) KNeighborsClassifier

- Run and Evaluate selected models

- MODELS

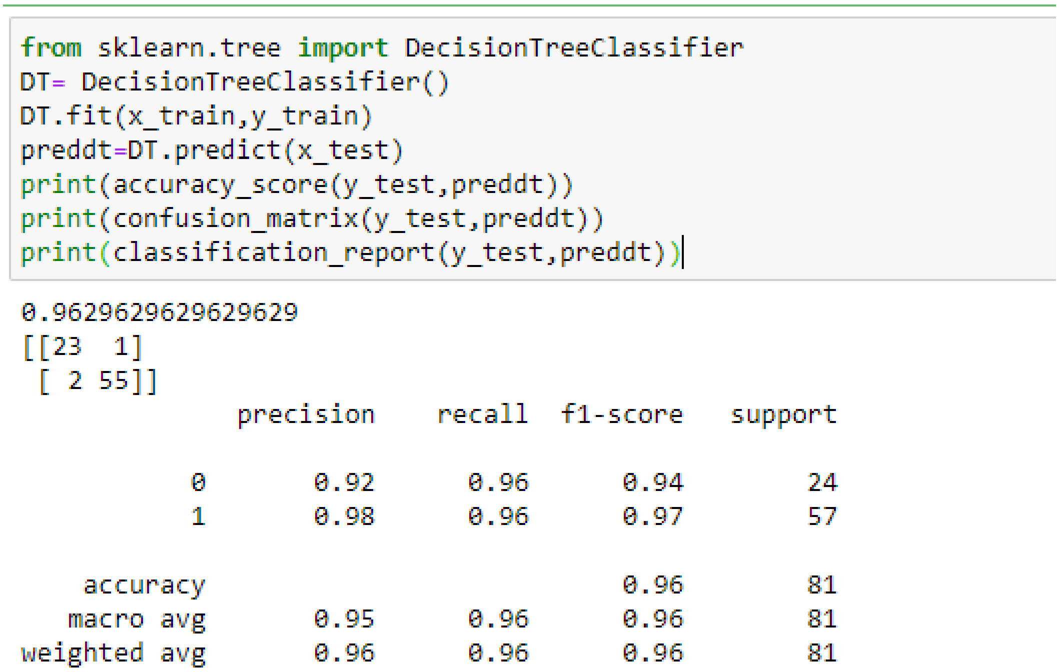
1) Logistic Regression



```
: from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
  from sklearn.linear_model import LogisticRegression
  LR= LogisticRegression()
  LR.fit(x_train,y_train)
  predlr=LR.predict(x_test)
  print(accuracy_score(y_test,predlr))
  print(confusion_matrix(y_test,predlr))
  print(classification_report(y_test,predlr))
```

```
0.7407407407407407
[[ 9 15]
 [ 6 51]]
              precision    recall  f1-score   support

           0       0.60      0.38      0.46        24
           1       0.77      0.89      0.83        57

    accuracy                           0.74        81
   macro avg       0.69      0.63      0.65        81
weighted avg       0.72      0.74      0.72        81
```

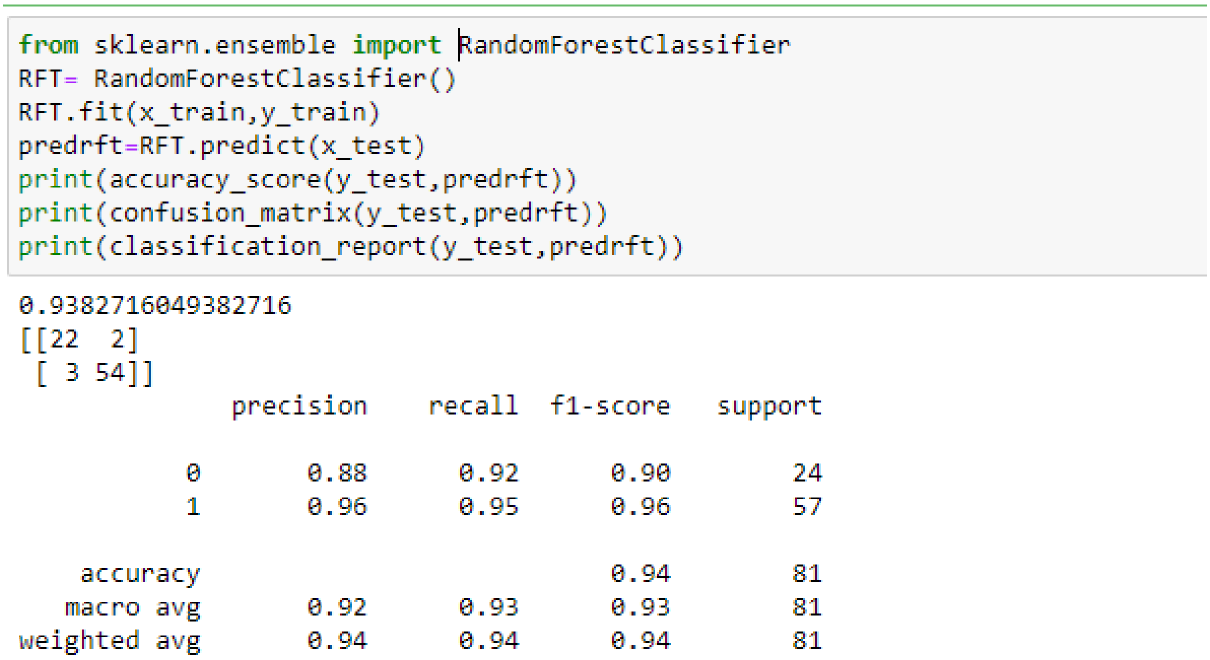
2) Decision Tree Classifier



```
from sklearn.tree import DecisionTreeClassifier
DT= DecisionTreeClassifier()
DT.fit(x_train,y_train)
preddt=DT.predict(x_test)
print(accuracy_score(y_test,preddt))
print(confusion_matrix(y_test,preddt))
print(classification_report(y_test,preddt))
```

```
0.9629629629629629
[[23  1]
 [ 2 55]]
              precision    recall  f1-score   support

           0       0.92      0.96      0.94        24
           1       0.98      0.96      0.97        57

    accuracy                           0.96        81
   macro avg       0.95      0.96      0.96        81
weighted avg       0.96      0.96      0.96        81
```

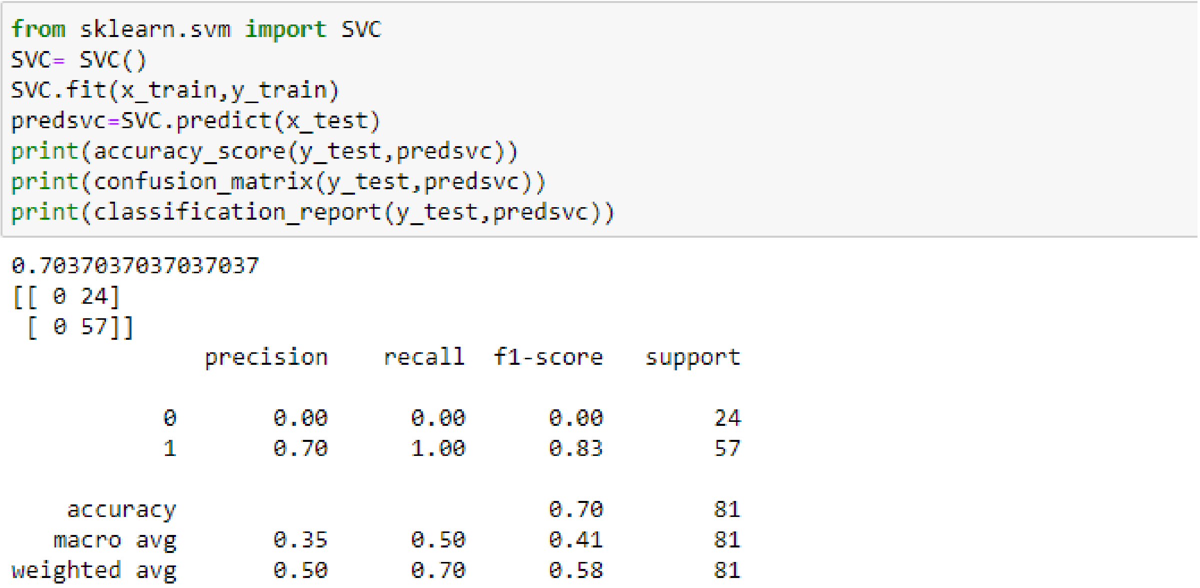
3) Random Forest Classifier



```
from sklearn.ensemble import RandomForestClassifier
RFT= RandomForestClassifier()
RFT.fit(x_train,y_train)
predrft=RFT.predict(x_test)
print(accuracy_score(y_test,predrft))
print(confusion_matrix(y_test,predrft))
print(classification_report(y_test,predrft))
```

```
0.9382716049382716
[[22  2]
 [ 3 54]]
              precision    recall  f1-score   support

           0       0.88      0.92      0.90        24
           1       0.96      0.95      0.96        57

    accuracy                           0.94        81
   macro avg       0.92      0.93      0.93        81
weighted avg       0.94      0.94      0.94        81
```

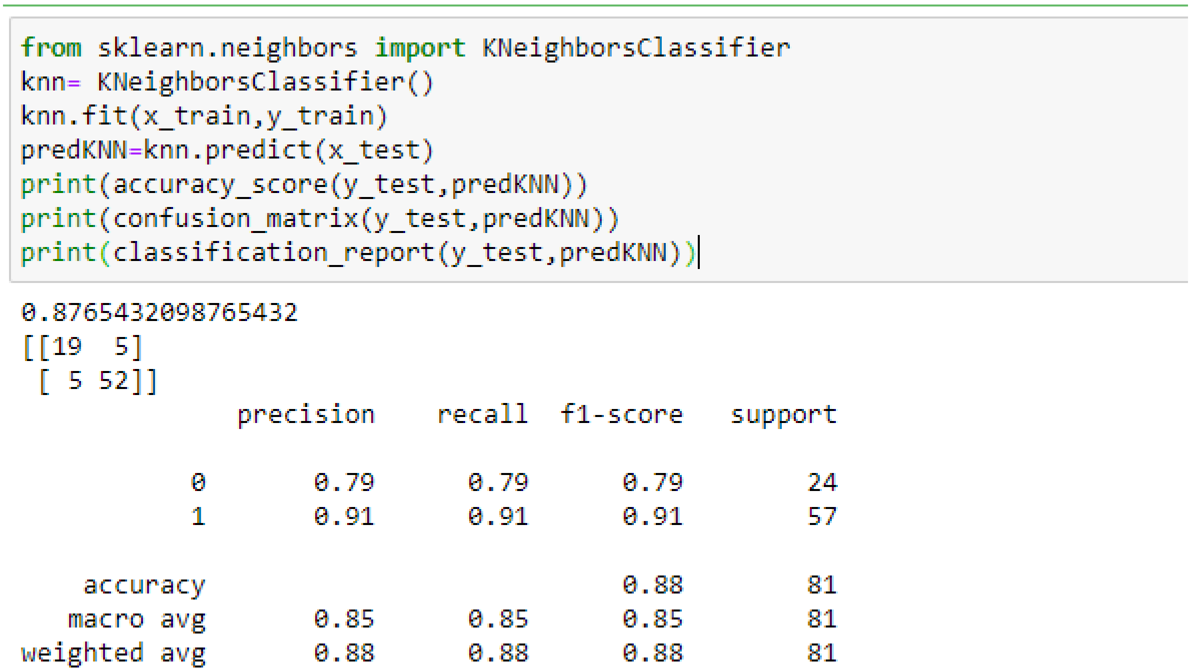
4) SVC



```
from sklearn.svm import SVC
SVC= SVC()
SVC.fit(x_train,y_train)
predsvc=SVC.predict(x_test)
print(accuracy_score(y_test,predsvc))
print(confusion_matrix(y_test,predsvc))
print(classification_report(y_test,predsvc))
```

```
0.7037037037037037
[[ 0 24]
 [ 0 57]]
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        24
           1       0.70      1.00      0.83        57

    accuracy                           0.70        81
   macro avg       0.35      0.50      0.41        81
weighted avg       0.50      0.70      0.58        81
```

5) KNeighborsClassifier



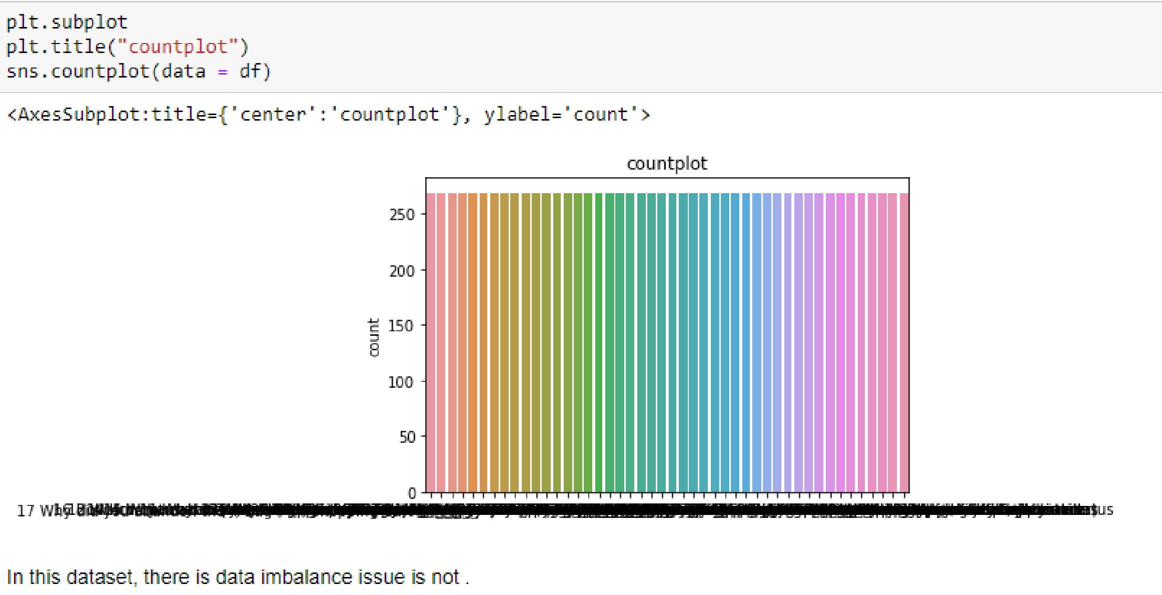
```
from sklearn.neighbors import KNeighborsClassifier
knn= KNeighborsClassifier()
knn.fit(x_train,y_train)
predKNN=knn.predict(x_test)
print(accuracy_score(y_test,predKNN))
print(confusion_matrix(y_test,predKNN))
print(classification_report(y_test,predKNN))
```

```
0.8765432098765432
[[19  5]
 [ 5 52]]
              precision    recall  f1-score   support

           0       0.79      0.79      0.79        24
           1       0.91      0.91      0.91        57

    accuracy                           0.88        81
   macro avg       0.85      0.85      0.85        81
weighted avg       0.88      0.88      0.88        81
```

- Visualizations

1) Count plot
2) Boxplot
3) Distplot
4) Scatterplot
5) Heatmap

1) Count plot

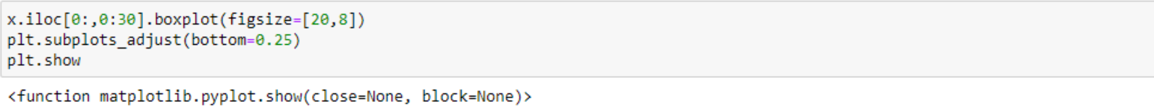


```
plt.subplot
plt.title("countplot")
sns.countplot(data = df)
```

```
<AxesSubplot:title={'center':'countplot'}, ylabel='count'>
```

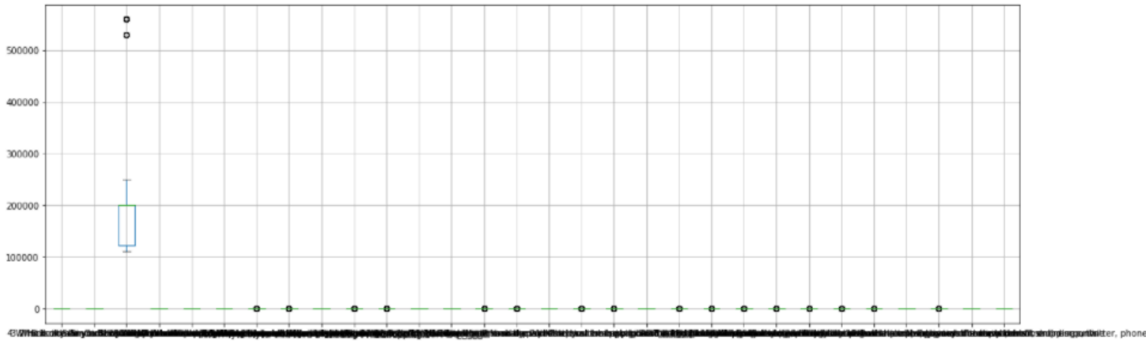
In this dataset, there is data imbalance issue is not .

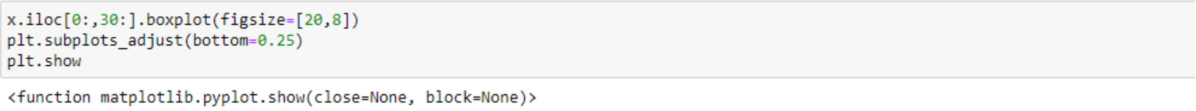
2) Boxplot



```
x.iloc[0:,0:30].boxplot(figsize=[20,8])
plt.subplots_adjust(bottom=0.25)
plt.show
```

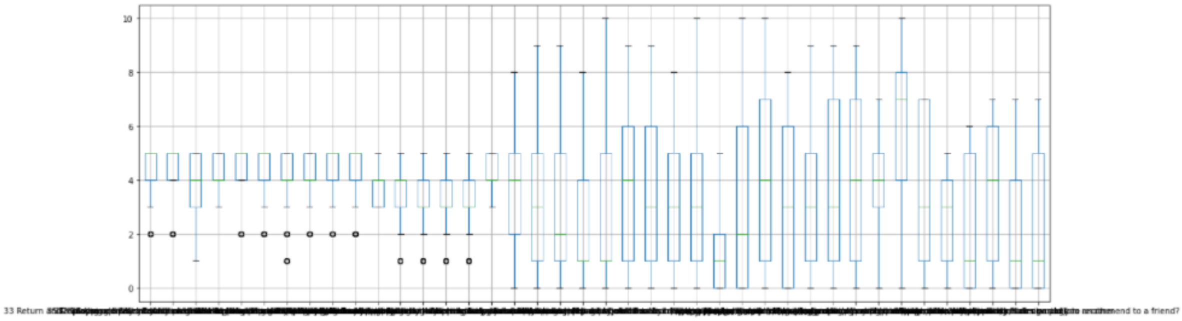
```
<function matplotlib.pyplot.show(close=None, block=None)>
```





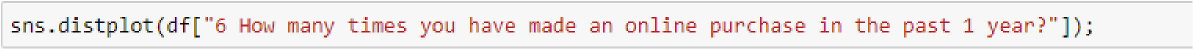
```
x.iloc[0:,30:].boxplot(figsize=[20,8])
plt.subplots_adjust(bottom=0.25)
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```

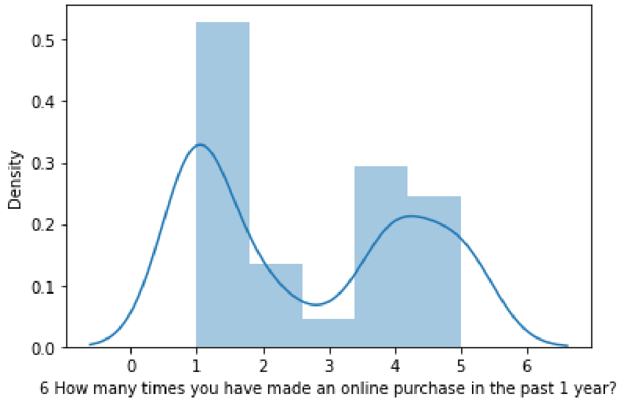


In given plots shows the outliers in this dataset.

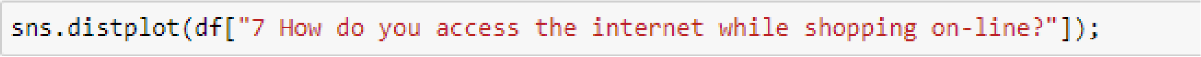
3) Distplot



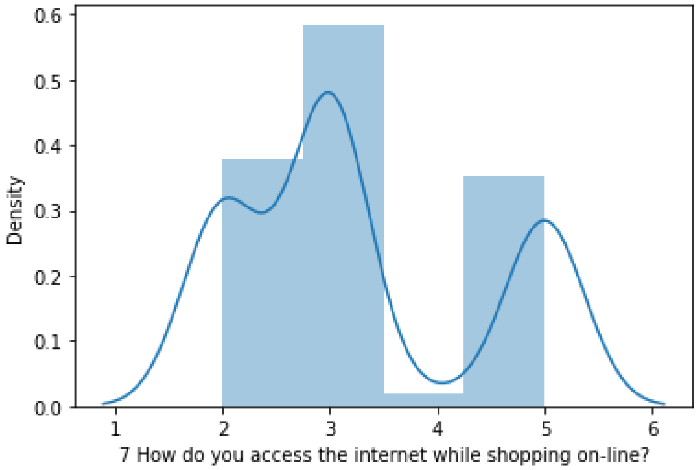
```
sns.distplot(df["6 How many times you have made an online purchase in the past 1 year?"]);
```



In this plot, data is normally distributed.

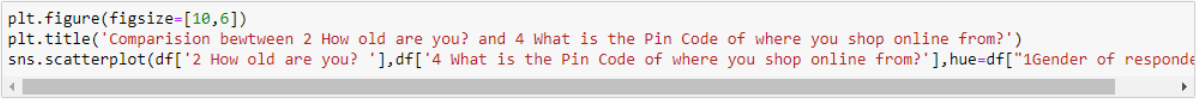


```
sns.distplot(df["7 How do you access the internet while shopping on-line?"]);
```



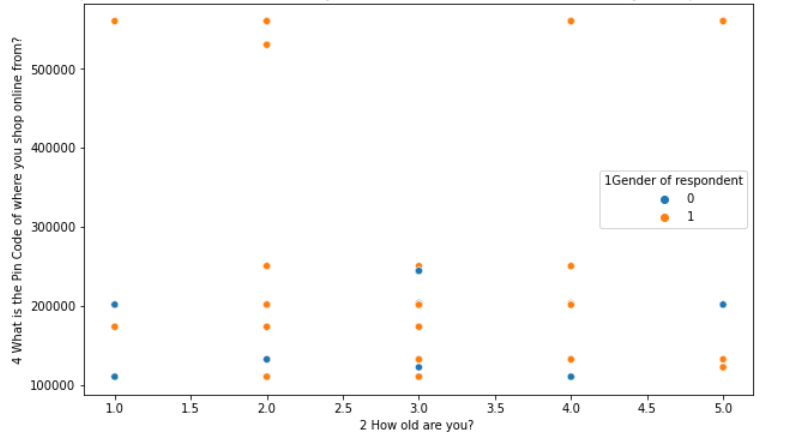
In this plot , data is not normally distributed.

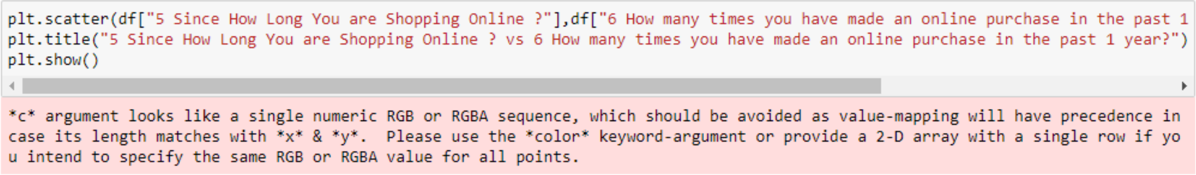
4) Scatterplot



```
plt.figure(figsize=[10,6])
plt.title('Comparision bewtween 2 How old are you? and 4 What is the Pin Code of where you shop online from?')
sns.scatterplot(df['2 How old are you? '],df['4 What is the Pin Code of where you shop online from?'],hue=df["1Gender of responde
```

Comparision bewtween 2 How old are you? and 4 What is the Pin Code of where you shop online from?

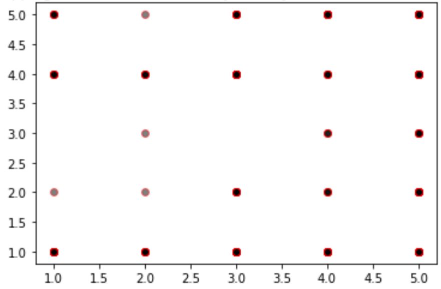




```
plt.scatter(df["5 Since How Long You are Shopping Online ?"],df["6 How many times you have made an online purchase in the past 1
plt.title("5 Since How Long You are Shopping Online ? vs 6 How many times you have made an online purchase in the past 1 year?")
plt.show()
```

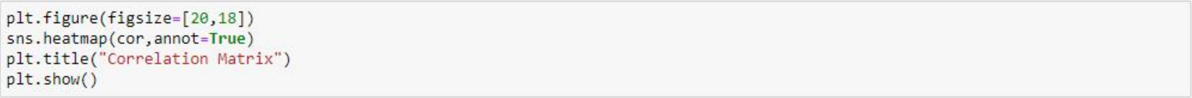
```
*c* argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in
case its length matches with *x* & *y*.  Please use the *color* keyword-argument or provide a 2-D array with a single row if yo
u intend to specify the same RGB or RGBA value for all points.
```

5 Since How Long You are Shopping Online ? vs 6 How many times you have made an online purchase in the past 1 year?

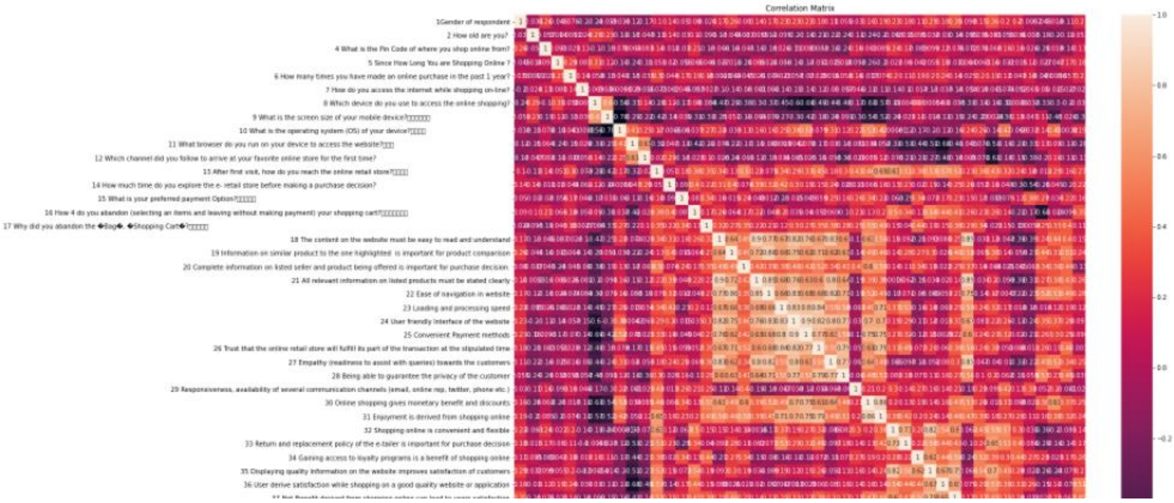


In this plot,outliers are present.

5) Heatmap



```
plt.figure(figsize=[20,18])
sns.heatmap(cor,annot=True)
plt.title("Correlation Matrix")
plt.show()
```

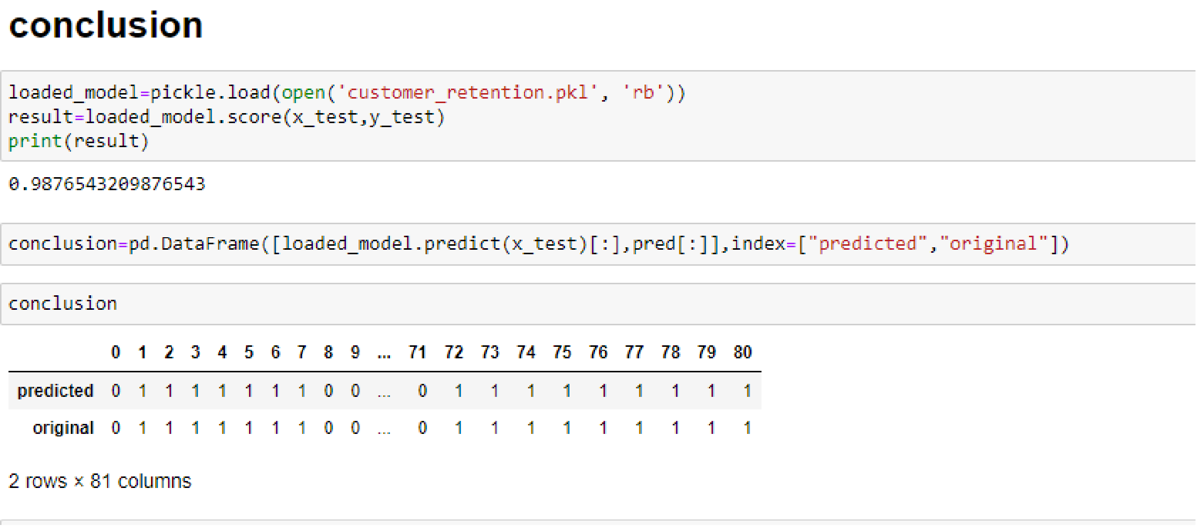


CONCLUSION

In these project, Decision Tree Classifier, Random Forest Classifier, Logistic Regression, SVC, KNeighborsClassifier these models are used.

In these project, we are getting highest accuracy with DecisionTreeClassifier (96%) and also cross validation score is best than other models.

Minimum difference in accuracy and cross validation score is for Decision Tree Classifier So this is our best model.



conclusion

```
loaded_model=pickle.load(open('customer_retention.pkl', 'rb'))
result=loaded_model.score(x_test,y_test)
print(result)
```

```
0.9876543209876543
```

```
conclusion=pd.DataFrame([loaded_model.predict(x_test)[:],pred[:]],index=["predicted","original"])
```

```
conclusion
```

	0	1	2	3	4	5	6	7	8	9	...	71	72	73	74	75	76	77	78	79	80
predicted	0	1	1	1	1	1	1	1	0	0	...	0	1	1	1	1	1	1	1	1	1
original	0	1	1	1	1	1	1	1	0	0	...	0	1	1	1	1	1	1	1	1	1

2 rows × 81 columns