Customer Retention Analysis

Submitted by:
Assie Houphouet Olivier

ACKNOWLEDGMENT

I want to say thank you to sir Sajid who give me this project and also all YouTube content creators and Google content creators. When I started the project it was something very confusing since there more 47 attributes(columns) in the dataset. I took two days to understand the project and all those attributes. This project help me a lot to understand several time while watching YouTube videos to solve a specific problem.

The data is collected from the Indian online shoppers. Results indicate the e-retail success factors, which are very much critical for customer satisfaction.

INTRODUCTION

Problem Statement:

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.

1) Problem Definition:

The customer common in the competitive markets as the customers naturally tend to choose the firms which offers a reasonable deal and ditch the ones which doesn't. The companies do not want to lose customers as bringing the new lot would cost extra as it takes a lot for advertising. In order to keep the firm from losing money, they'll have to retain customers as much as possible. The following problem is to check the factors that make the customer to stay loyalty and we'll have to analyse the factors given to come up with a feasible solution possible to retain the customers.

2) Data Analysis:

The data is collected from the Indian online shoppers. Results indicate the e-retails success factors, which are very much critical for customer satisfaction.

There are 44 independent variable which are like factors influencing the customer intention and boost to repeat the purchase with the same company.

I first extract the encoded data from the excel sheet which I will be using for numerical analysis and loaded it on the Jupiter Netbook. The data was in excel format so I converted into csv format(comma-separated values).

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
import seaborn as sns
from sklearn.model_selection import train_test_split
warnings.filterwarnings('ignore')
```

I Will first check the Encoded that and see how it look..

Loading the dataset after extacting the first 47 attribute which i will used for my Analysis.

```
data = pd.read_csv('customer_retention_dataset1.csv')
```

```
# Let's check the shape of our dataset

data.shape
```

```
(269, 47)
```

```
data.head()
```

Check the shape of the dataset:
The shape of the dataset is something very import after loaded the dataset since it help you see the number of records and the attributes are in the dataset. Here we can see that our dataset has 269 records and 47 attributes.

Null values checking: The null values is something important check since it is since any dataset can contain Null/nan/missing values as we do in our every day life while we filling the online forms. Those field we ignore while we are fill are consider as null values or missing values. So those values can be filled by the data scientist using several techniques after they have been identified. As we can see here our dataset has no missing values.

```
[7]: data.isna().sum()
```

```
[7]: 1Gender of respondent
     0
     2 How old are you?
     0
     3 Which city do you shop online from?
     0
     4 What is the Pin Code of where you shop online from?
     0
     5 Since How Long You are Shopping Online ?
     0
     6 How many times you have made an online purchase in the past 1 year?
     0
     7 How do you access the internet while shopping on-line?
     0
     8 Which device do you use to access the online shopping?
     0
     9 What is the screen size of your mobile device?\t\t\t\t\t\t
     0
     10 What is the operating system (OS) of your device?\t\t\t\t
     0
     11 What browser do you run on your device to access the website?\t\t\t
     0
     12 Which channel did you follow to arrive at your favorite online store for the first time?
     0
     13 After first visit, how do you reach the online retail store?\t\t\t\t
     0
```

Number of classes in each columns checking:
This technique help us to see how is the data in each column and whether the column contain continuous data or categorical data, since there are different way to deal with the continuous and categorical data.

```
for i in data.columns:
    print(i, len(data[i].unique()))
```

```
1Gender of respondent 2
2 How old are you?  5
3 Which city do you shop online from? 11
4 What is the Pin Code of where you shop online from? 39
5 Since How Long You are Shopping Online ? 5
6 How many times you have made an online purchase in the past 1 year? 5
7 How do you access the internet while shopping on-line? 4
8 Which device do you use to access the online shopping? 4
9 What is the screen size of your mobile device?
10 What is the operating system (OS) of your device?
11 What browser do you run on your device to access the website?
12 Which channel did you follow to arrive at your favorite online store for the first time?
13 After first visit, how do you reach the online retail store?
14 How much time do you explore the e- retail store before making a purchase decision?
15 What is your preferred payment Option?
3
16 How 4 do you abandon (selecting an items and leaving without making payment) your shopping cart?
4
```

C) EDA:

Exploratory data analysis is technique used for analysing the dataset using statistical graphics and others visualization methods .This method is used by every data scientist in order to investigate and analyse the data through a visual format and make his own conclusions based the insight of the data.
To overcome this process several of libraries are used.
Example: matplotlib.pyplot(plt), seaborn (sns) where plt and sns are used as alias. Those libraries are helpful to visualize the data directly from the Notebook and make the conclusions.

Dropping unwanted attributes: During this process we are dropping all the columns that we are felling there not important in our analyse process.

```
## droping unwanted ciolumns

data.drop(['3 Which city do you shop online from?','4 What is the Pin Code of where you shop online from?'],axis=1,inplace=True)
```

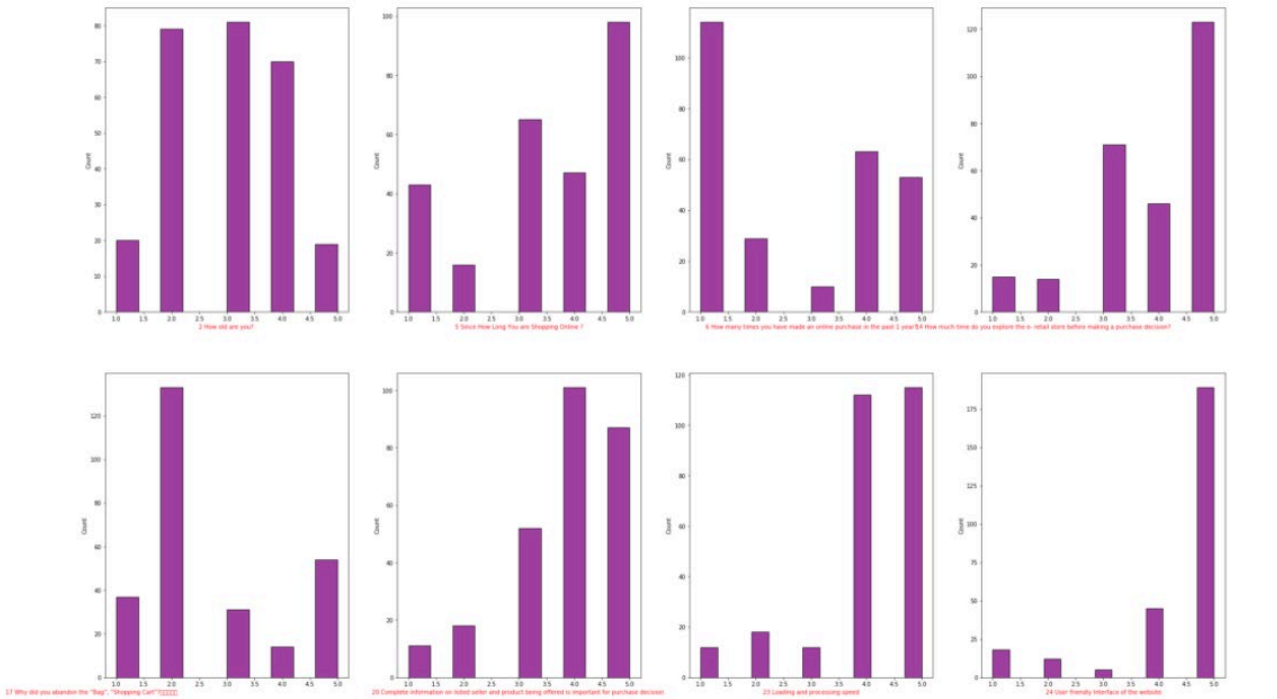
```
## let's verify it if the columns are dropped
data.shape
```

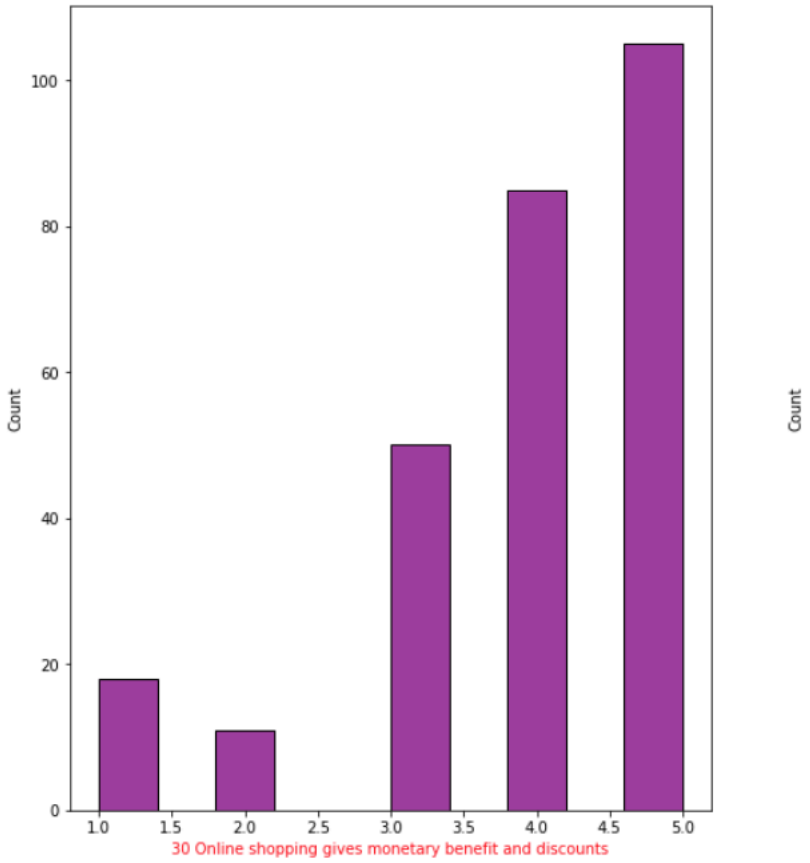
```
(269, 45)
```

Now we can see that our dataset contain 45 attributes (columns)

- **Here will check for a specific question what is the customers points of view.**

```
]: plt.figure(figsize=(35, 45))

graph = 1

for column in df_1:
    if graph <= 16:
        ax = plt.subplot(4,4,graph)
        sns.histplot(df_1[column],color='purple')
        plt.xlabel(column,color ='r',fontsize = 10)
    graph+=1
plt.show()
```



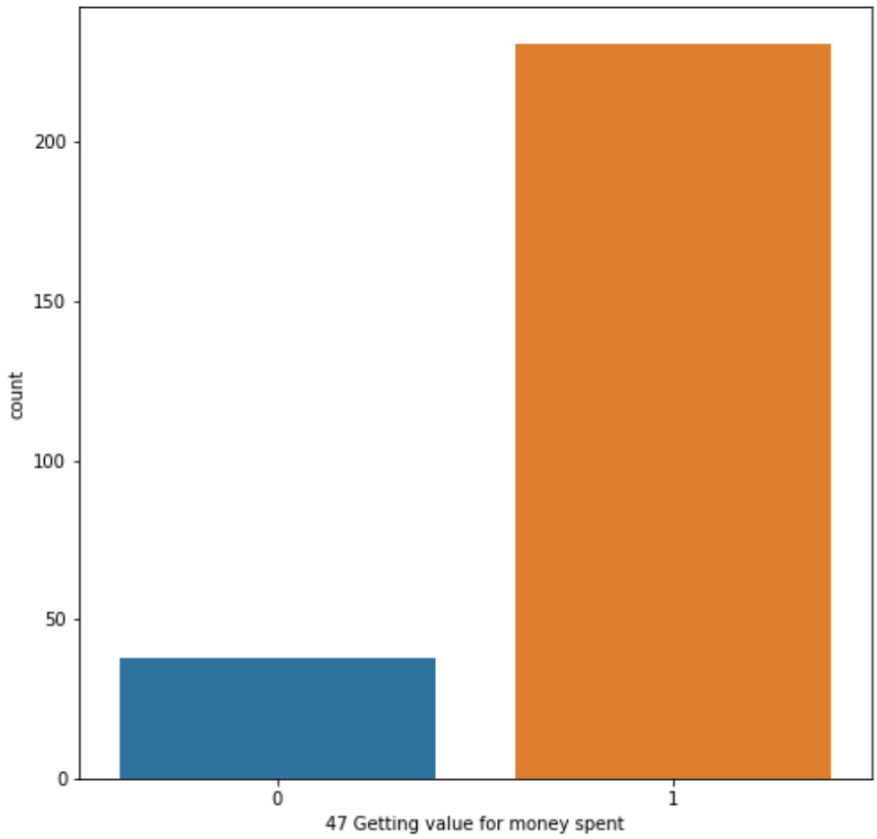


From the above histplot we can see that the majority of the customers answers is 3,4,5 since those 3 answers mean the customer is satisfied with the service provided by the company. The company can study those customers profile and see how can it compare with those customer who are not satisfied and try to overcome their satisfaction if the customer want them stay.
So using this analysis we do the same for other attributes.

- **Considering [47 Getting value for money spent] as target attribute.**

[123]:
```
## checking how many customer are agreed and how many are not using countplot.

plt.figure(figsize=(8,8))
sns.countplot(x='47 Getting value for money spent',data=data)
```

[123]: <matplotlib.axes._subplots.AxesSubplot at 0x7ff290f44790>

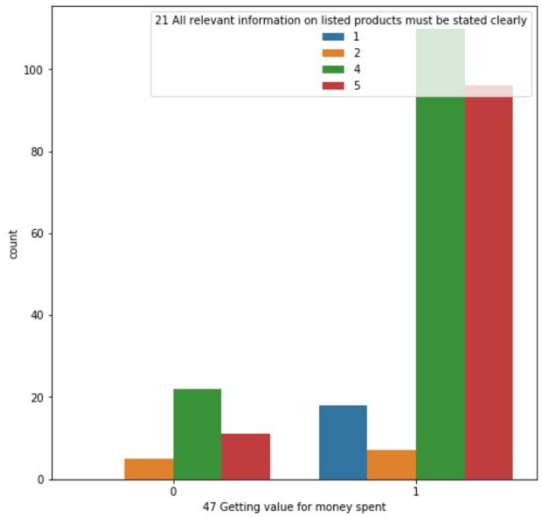


We can see by considering the column 47 as target variable we can see that most of customers who getting value for money spent (satisfied customer) is majority since the rate is above 200 and for minority is for those who are approved for the affirmation is under 50.

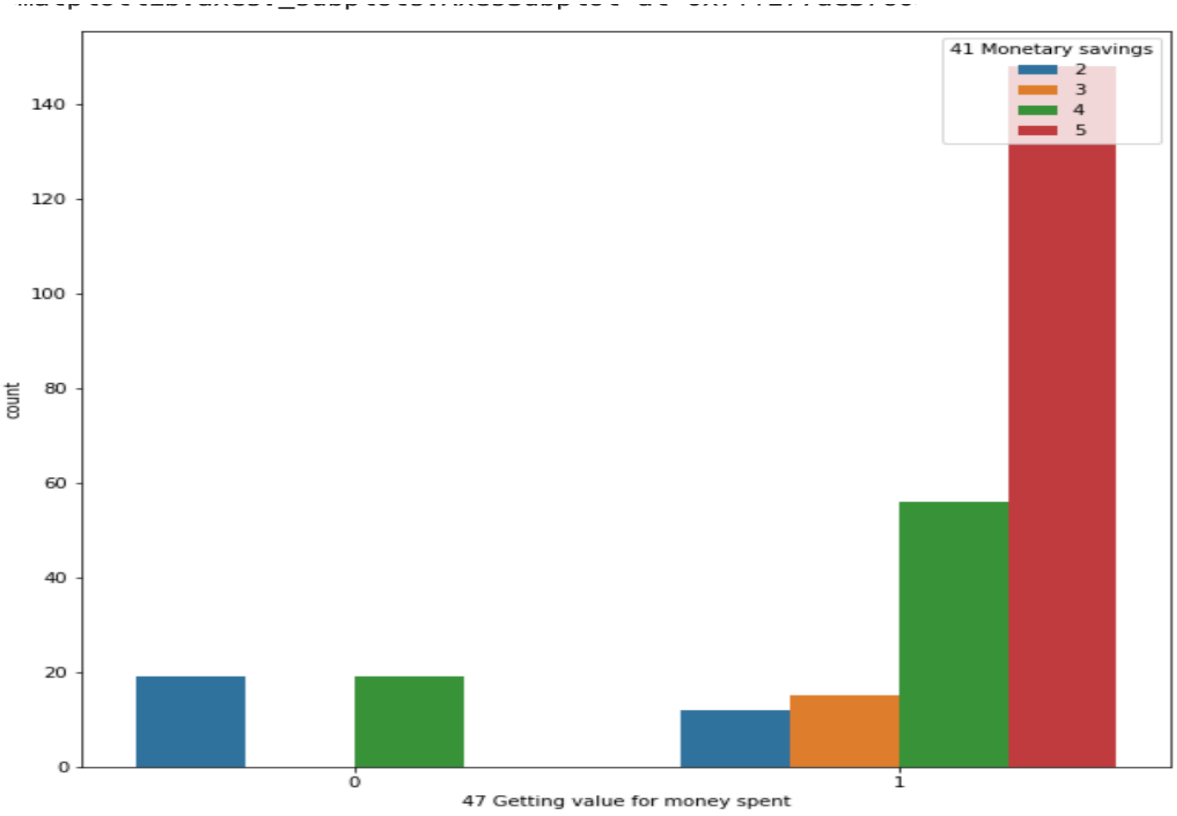
Since we are considering the attribute [47 Getting value for money spent] as dependent variable, let's see others attributes effect on it.

```
plt.figure(figsize=(8,8))
sns.countplot(x='47 Getting value for money spent',data=data,hue='21 All relevant information on listed products must be stated clearly')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff293b6ae20>
```



It’s clearly shown from the above count plot rate of the customers who agreed and strongly agreed (satisfied customers) is very positive highest than those who are disagreed for the affirmation(unsatisfied customers) and we say in conclusion that if the company make some effort it can satisfied all his customers.



[133]:
```
detail_data['41 Monetary savings'].value_counts()
```

[133]:
```
Strongly agree (5)    148
Agree (4)              75
Disagree (2)           31
indifferent (3)        15
Name: 41 Monetary savings, dtype: int64
```

From the above graph we can see that all the customers “strongly agree “ and “indifferent” with the affirmation 41 are all belong to customers satisfied class[1] but for those customers who “agree” and “disagree” some belong to customers satisfied [1] as well customer unsatisfied[0]. So here we can say the that the attribute [41 Monetary savings] have effect to our dependent variable but not 100% percent.

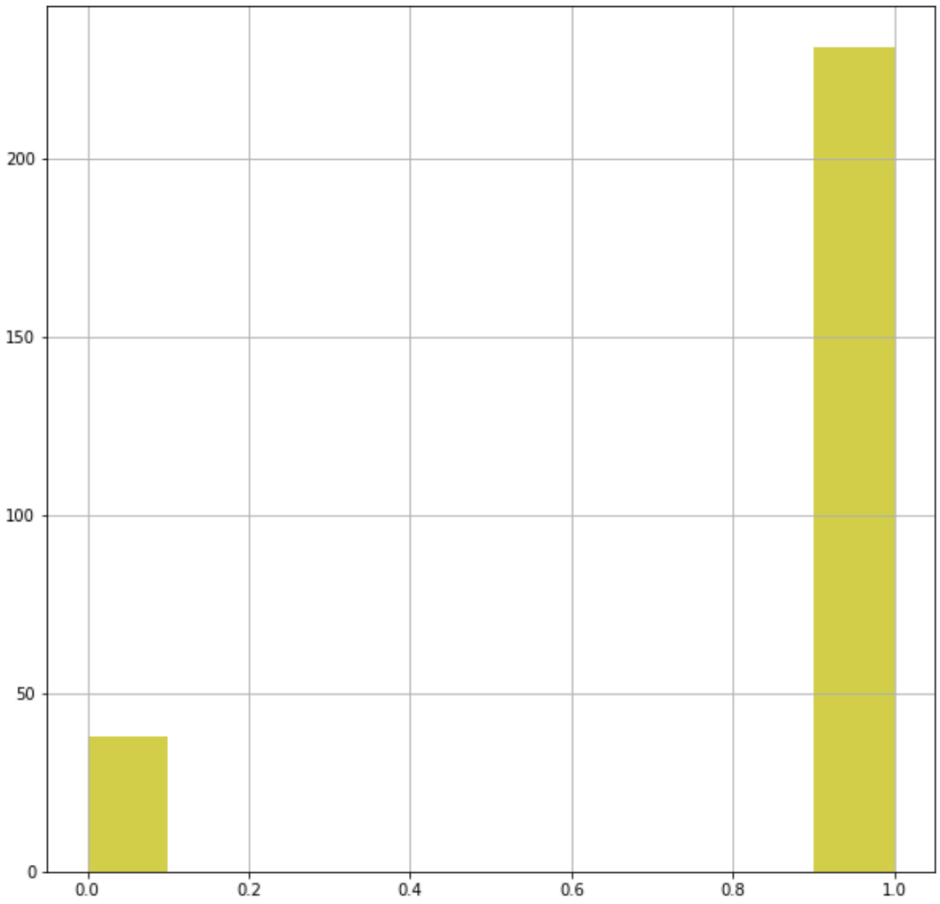
- Over-sampling

The target variable is imbalanced as we can see the distribution data is unequal.

[98]:
```
1    231
0     38
Name: 47 Getting value for money spent, dtype: int64
```

[79]:
```
### let's plot it.

data['47 Getting value for money spent'].hist(color = 'y',alpha = 0.7, figsize=(10,10))
```

[79]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff293a199d0>
```



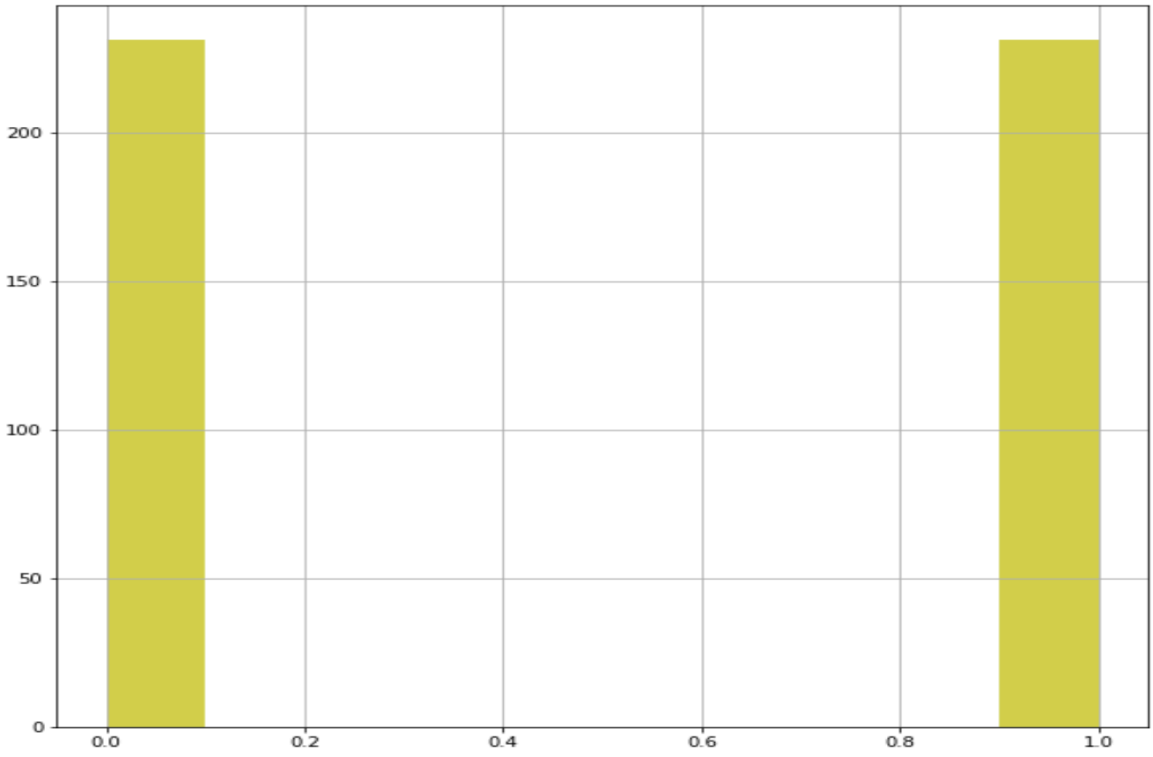
```
80]: ## let's solve the unbalance problem

     ## importang the libraries
     from imblearn.over_sampling import SMOTE

     SM = SMOTE()
```

```
88]: ## let's first split our dataset into label and features.

     y = data['47 Getting value for money spent']

     x = data.iloc[:,:44]
```

```
89]: x_over , y_over = SM.fit_resample(x, y)
```

```
91]: y_over.hist(color = 'y',alpha = 0.7, figsize=(10,10))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff275a9ee20>
```



Now we can see that our both classes of the label are balanced...

From the above histogram we can see that the imbalanced problem is solved.

E) Building Machine learning Models

The target variable is (y_over) have categorical data so we will used classification algorithms to build the model and predict it.

- Importing the libraries

```
### Importing the libraries

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

I build the first model using the Logistic Regression Algorithm.

```
## Let's find the best random state.

max_accu = 0
max_RS = 0
for i in range(1,400):
    x_train, x_test, y_train,y_test = train_test_split(x_over,y_over,
                                                       test_size=30,
                                                       random_state=i)
    LR = LogisticRegression()
    LR.fit(x_train, y_train)
    predrf = LR.predict(x_test)
    acc = accuracy_score(y_test, predrf)
    if acc > max_accu:
        max_accu = acc
        max_RS = i
print('Best accuracy is :',max_accu, 'on Random_state',max_RS)
```

```
Best accuracy is : 1.0 on Random_state 1
```

From the above program we can see that we get the best random state with accuracy of 100% . So we will this random state to build other model.

```
## let's build gain the model using  best random_state = 1

x_train, x_test, y_train,y_test = train_test_split(x_over,y_over,
                                                   test_size=30,
                                                   random_state=1)
lr_model = LogisticRegression()
lr_model.fit(x_train, y_train)
y_predlr = lr_model.predict(x_test)
print("The Accuracy is :",accuracy_score(y_test, y_predlr))
print('--'*20)
print(confusion_matrix(y_test,y_predlr))
print(classification_report(y_test,y_predlr))
```

```
The Accuracy is : 1.0
----------------------------------------
[[13  0]
 [ 0 17]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        13
           1       1.00      1.00      1.00        17

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

As we can see above , we build again the same model using the best random state and we evaluate the metrics for the model performance.

- DecisionTreeClassifier

```
## import the libraries
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC # Suport Vector Machine model.
```

```
### DecisionTreeClassifier model.

dt = DecisionTreeClassifier(random_state=1)
dt.fit(x_train, y_train)
pred_dt = dt.predict(x_test)
print("The Accuracy is :",accuracy_score(y_test, pred_dt))
print('--'*20)
print(confusion_matrix(y_test, pred_dt))
print(classification_report(y_test,pred_dt))
print('**'*30)
print('**'*30)
```

```
The Accuracy is : 1.0
----------------------------------------
[[13  0]
 [ 0 17]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        13
           1       1.00      1.00      1.00        17

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

The DecisionTreeClassifier model gives 100% accuracy

- RandomForestClassifier

[46]:
```
## RandomForestClassifier

rf = RandomForestClassifier(1)
rf.fit(x_train, y_train)
pred_rf = rf.predict(x_test)

print("The Accuracy is :",accuracy_score(y_test, pred_rf))
print('--'*20)
print(confusion_matrix(y_test, pred_rf))
print(classification_report(y_test,pred_rf))
print('**'*30)
print('**'*30)
```

```
The Accuracy is : 1.0
----------------------------------------
[[13  0]
 [ 0 17]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        13
           1       1.00      1.00      1.00        17

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30

************************************************************
```

The RandomForestClassifier gives also 100% accuracy.

- Overfitting checking

[148]:
```
## importing the library
from sklearn.model_selection import cross_val_score
```

[153]:
```
## first  model lr_model

scr = cross_val_score(lr_model,x_over,y_over,cv=5)
print('Cross Validation Score of LogisticRegression model:',scr.mean())
```

```
Cross Validation Score of LogisticRegression model: 1.0
```

[150]:
```
## second  model dt

scr = cross_val_score(dt,x_over,y_over,cv=5)
print('Cross Validation Score of DecisionTreeClassifier model:',scr.mean())
```

```
Cross Validation Score of DecisionTreeClassifier model: 1.0
```

[151]:
```
## Third model rf

scr = cross_val_score(rf,x_over,y_over,cv=5)
print('Cross Validation Score of  RandomForestClassifier model:',scr.mean())
```

```
Cross Validation Score of  RandomForestClassifier model: 1.0
```

[152]:
```
## Fourth model svc

scr = cross_val_score(svc,x_over,y_over,cv=5)

print('Cross Validation Score of Suport Vector Machine model:',scr.mean())
```

```
Cross Validation Score of Suport Vector Machine model: 1.0
```

From above cross validation checking , every model gives score = 1 and this mean our models are not overfitting.

- Model Saving.

Let's save the first model .

```
4]: import joblib

5]: joblib.dump(lr_model,'Customer_Retention_Prediction_Model.pkl')

5]: ['Customer_Retention_Prediction_Model.pkl']
```

CONCLUSION

We will give this model to the company so that the company can know if a specific customer is satisfied or not and if the company knows there information about the customer while using the model to predict it and help the company to keep is customers.