Assignment -1

Python Programming

Assignment Date	19 September 2022
Student Name	Sanjana .L
Student Roll Number	820419106049
Maximum Marks	2 Marks

Question

1.Download the Dataset

2. Load the dataset

3. Perform below visualizations

- Univariate Analysis
- Bi-variate Analysis
- Multi-variate Analysis

4.perform descriptive statistics on the dataset.

5. Handle the missing values.

6. Find the outliers and replace the outliers

7. check the Categorical columns and perform encoding

8.Split the data into dependent and independent variables

9. Scale the independent variables

10. Split the data into training and testing

```
import pandas as pd
import numpy as np

data=pd.read_csv("/content/drive/MyDrive/Churn_Modelling.csv")

#descriptive analysis
data.describe()
```

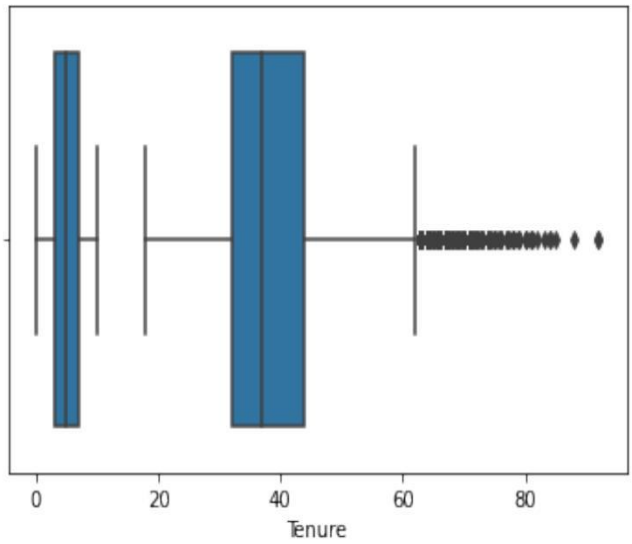
```
         RowNumber    CustomerId   CreditScore           Age
Tenure  \
count  10000.00000  1.000000e+04  10000.000000  10000.000000
10000.000000
mean    5000.50000  1.569094e+07    650.528800     38.921800
5.012800
std     2886.89568  7.193619e+04     96.653299     10.487806
2.892174
min        1.00000  1.556570e+07    350.000000     18.000000
0.000000
25%     2500.75000  1.562853e+07    584.000000     32.000000
3.000000
50%     5000.50000  1.569074e+07    652.000000     37.000000
5.000000
75%     7500.25000  1.575323e+07    718.000000     44.000000
7.000000
max    10000.00000  1.581569e+07    850.000000     92.000000
10.000000

             Balance  NumOfProducts    HasCrCard  IsActiveMember  \
count   10000.000000   10000.000000  10000.00000    10000.000000
mean    76485.889288       1.530200      0.70550        0.515100
std     62397.405202       0.581654      0.45584        0.499797
min         0.000000       1.000000      0.00000        0.000000
25%         0.000000       1.000000      0.00000        0.000000
50%     97198.540000       1.000000      1.00000        1.000000
75%    127644.240000       2.000000      1.00000        1.000000
max    250898.090000       4.000000      1.00000        1.000000

       EstimatedSalary        Exited
count     10000.000000  10000.000000
mean     100090.239881      0.203700
std       57510.492818      0.402769
min          11.580000      0.000000
25%       51002.110000      0.000000
50%      100193.915000      0.000000
75%      149388.247500      0.000000
max      199992.480000      1.000000
```

```
#dealing with missing values
data.isnull().sum()
```

```
RowNumber          0
CustomerId         0
Surname            0
CreditScore        0
Geography          0
Gender             0
Age                0
Tenure             0
Balance            0
NumOfProducts      0
HasCrCard          0
IsActiveMember     0
EstimatedSalary    0
Exited             0
dtype: int64
```

```
#dealing with outliers
import seaborn as sns

sns.boxplot(data['Age'])
sns.boxplot(data['Tenure'])
```

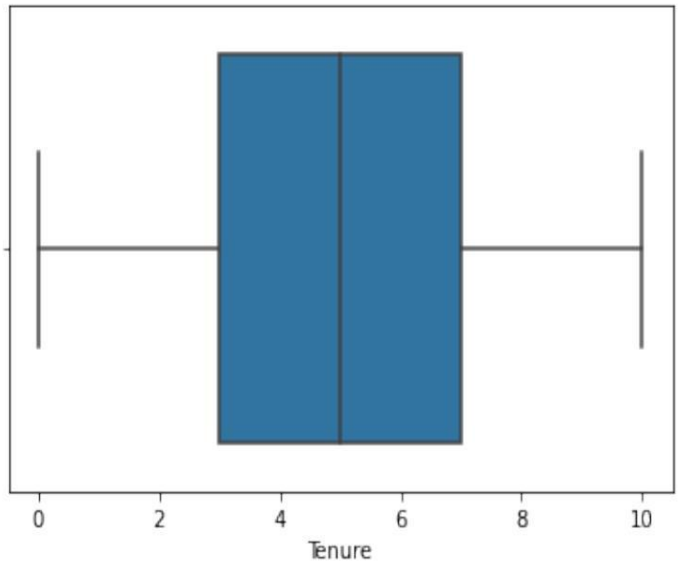
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be `data`, and
passing other arguments without an explicit keyword will result in an
error or misinterpretation.
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be `data`, and
passing other arguments without an explicit keyword will result in an
error or misinterpretation.
  FutureWarning

<matplotlib.axes._subplots.AxesSubplot at 0x7fd6259e8d10>
```




```
sns.boxplot(data['Tenure'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be `data`, and
passing other arguments without an explicit keyword will result in an
error or misinterpretation.
  FutureWarning
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd650329390>
```




```
qnt =data.quantile(q=[0.25,0.75])
```

```
qnt
```

```
      RowNumber   CustomerId  CreditScore   Age  Tenure    Balance  \
0.25    2500.75  15628528.25        584.0  32.0     3.0       0.00
0.75    7500.25  15753233.75        718.0  44.0     7.0  127644.24

      NumOfProducts  HasCrCard  IsActiveMember  EstimatedSalary
Exited
0.25            1.0        0.0             0.0       51002.1100
0.0
0.75            2.0        1.0             1.0      149388.2475
0.0
```

```
IQR =qnt.loc[0.75]-qnt.loc[0.25]
```

```
IQR
```

```
RowNumber            4999.5000
CustomerId         124705.5000
CreditScore           134.0000
Age                    12.0000
Tenure                  4.0000
Balance            127644.2400
NumOfProducts           1.0000
HasCrCard               1.0000
IsActiveMember          1.0000
EstimatedSalary     98386.1375
```

```
Exited                  0.0000
dtype: float64
```

```
upper_extreme =qnt.loc[0.75]+1.5*IQR
```

```
upper_extreme
```

```
RowNumber          1.499950e+04
CustomerId         1.594029e+07
CreditScore        9.190000e+02
Age                6.200000e+01
Tenure             1.300000e+01
Balance            3.191106e+05
NumOfProducts      3.500000e+00
HasCrCard          2.500000e+00
IsActiveMember     2.500000e+00
EstimatedSalary    2.969675e+05
Exited             0.000000e+00
dtype: float64
```

```
lower_extreme = qnt.loc[0.25]-1.5*IQR
```

```
lower_extreme
```

```
RowNumber         -4.998500e+03
CustomerId         1.544147e+07
CreditScore        3.830000e+02
Age                1.400000e+01
Tenure            -3.000000e+00
Balance           -1.914664e+05
NumOfProducts     -5.000000e-01
HasCrCard         -1.500000e+00
IsActiveMember    -1.500000e+00
EstimatedSalary   -9.657710e+04
Exited             0.000000e+00
dtype: float64
```

```
data[data['Age']>6.200000e+01]
```

```
      RowNumber  CustomerId      Surname  CreditScore Geography
Gender  Age  \
58           59    15623944        T'ien          511     Spain
Female   66
85           86    15805254      Ndukaku          652     Spain
Female   75
104         105    15804919     Dunbabin          670     Spain
Female   65
158         159    15589975      Maclean          646    France
Female   73
181         182    15789669         Hsia          510    France
Male   65
...         ...         ...          ...          ...       ...     ..
```

```
.  ...
9753       9754    15705174    Chiedozie          656   Germany
Male   68
9765       9766    15777067       Thomas          445    France
Male   64
9832       9833    15814690  Chukwujekwu          595   Germany
Female   64
9894       9895    15704795        Vagin          521    France
Female   77
9936       9937    15653037        Parks          609    France
Male   77

      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
58         4       0.00              1          1               0
85        10       0.00              2          1               1
104        1       0.00              1          1               1
158        6   97259.25              1          0               1
181        2       0.00              2          1               1
...      ...        ...            ...        ...             ...
9753       7  153545.11              1          1               1
9765       2  136770.67              1          0               1
9832       2  105736.32              1          1               1
9894       6       0.00              2          1               1
9936       1       0.00              1          0               1

      EstimatedSalary  Exited
58            1643.11       1
85          114675.75       0
104         177655.68       1
158         104719.66       0
181          48071.61       0
...               ...     ...
9753        186574.68       0
9765         43678.06       0
9832         89935.73       1
9894         49054.10       0
9936         18708.76       0

[359 rows x 14 columns]
```

```
data[data['Tenure']> 1.300000e+01]
```

```
Empty DataFrame
Columns: [RowNumber, CustomerId, Surname, CreditScore, Geography,
Gender, Age, Tenure, Balance, NumOfProducts, HasCrCard,
IsActiveMember, EstimatedSalary, Exited]
Index: []
```

```
data[data['Age']<1.400000e+01]
```

```
Empty DataFrame
Columns: [RowNumber, CustomerId, Surname, CreditScore, Geography,
Gender, Age, Tenure, Balance, NumOfProducts, HasCrCard,
IsActiveMember, EstimatedSalary, Exited]
Index: []
```

```
data[data['Tenure']<-3.000000e+00]
```

```
Empty DataFrame
Columns: [RowNumber, CustomerId, Surname, CreditScore, Geography,
Gender, Age, Tenure, Balance, NumOfProducts, HasCrCard,
IsActiveMember, EstimatedSalary, Exited]
Index: []
```

```
data.mean
```

```
<bound method NDFrame._add_numeric_operations.<locals>.mean of
RowNumber  CustomerId    Surname  CreditScore Geography  Gender
Age  \
0             1    15634602   Hargrave          619    France  Female
42
1             2    15647311       Hill          608     Spain  Female
41
2             3    15619304       Onio          502    France  Female
42
3             4    15701354       Boni          699    France  Female
39
4             5    15737888   Mitchell          850     Spain  Female
43
...         ...         ...        ...          ...       ...     ...
...
9995       9996    15606229   Obijiaku          771    France    Male
39
9996       9997    15569892  Johnstone          516    France    Male
35
9997       9998    15584532        Liu          709    France  Female
36
9998       9999    15682355  Sabbatini          772   Germany    Male
42
9999      10000    15628319     Walker          792    France  Female
28

      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0          2       0.00              1          1               1
1          1   83807.86              1          0               1
2          8  159660.80              3          1               0
3          1       0.00              2          0               0
4          2  125510.82              1          1               1
...      ...        ...            ...        ...             ...
9995       5       0.00              2          1               0
9996      10   57369.61              1          1               1
```

```
9997       7       0.00              1          0               1
9998       3   75075.31              2          1               0
9999       4  130142.79              1          1               0

      EstimatedSalary  Exited
0           101348.88       1
1           112542.58       0
2           113931.57       1
3            93826.63       0
4            79084.10       0
...               ...     ...
9995         96270.64       0
9996        101699.77       0
9997         42085.58       1
9998         92888.52       1
9999         38190.78       0

[10000 rows x 14 columns]>
```

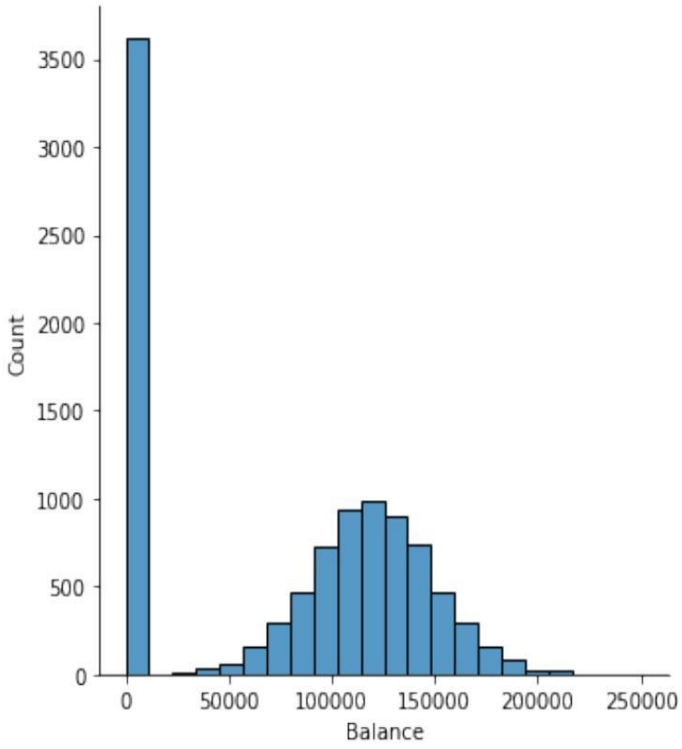
```
#Replacing outliers with mean
data['Age']=np.where(data['Age']>6.200000e+01,data['Age'].mean(),data[
'Age'])
```

```
#After replacing mean,no outliers are present for Age column
data[data['Age']>6.200000e+01]
```

```
Empty DataFrame
Columns: [RowNumber, CustomerId, Surname, CreditScore, Geography,
Gender, Age, Tenure, Balance, NumOfProducts, HasCrCard,
IsActiveMember, EstimatedSalary, Exited]
Index: []
```

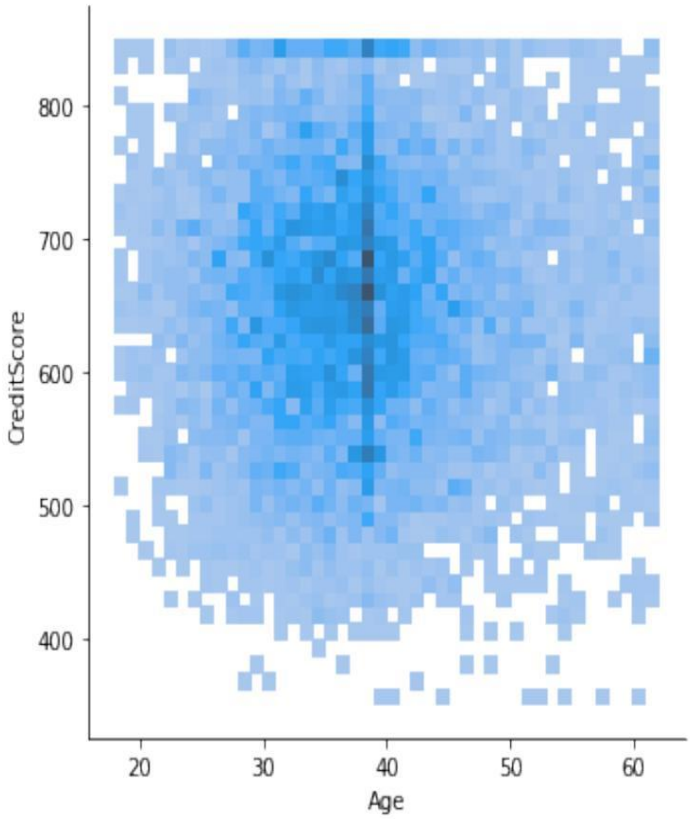
```
#univarient Analysis
sns.displot(data, x="Balance")
```

```
<seaborn.axisgrid.FacetGrid at 0x7fd624255e90>
```



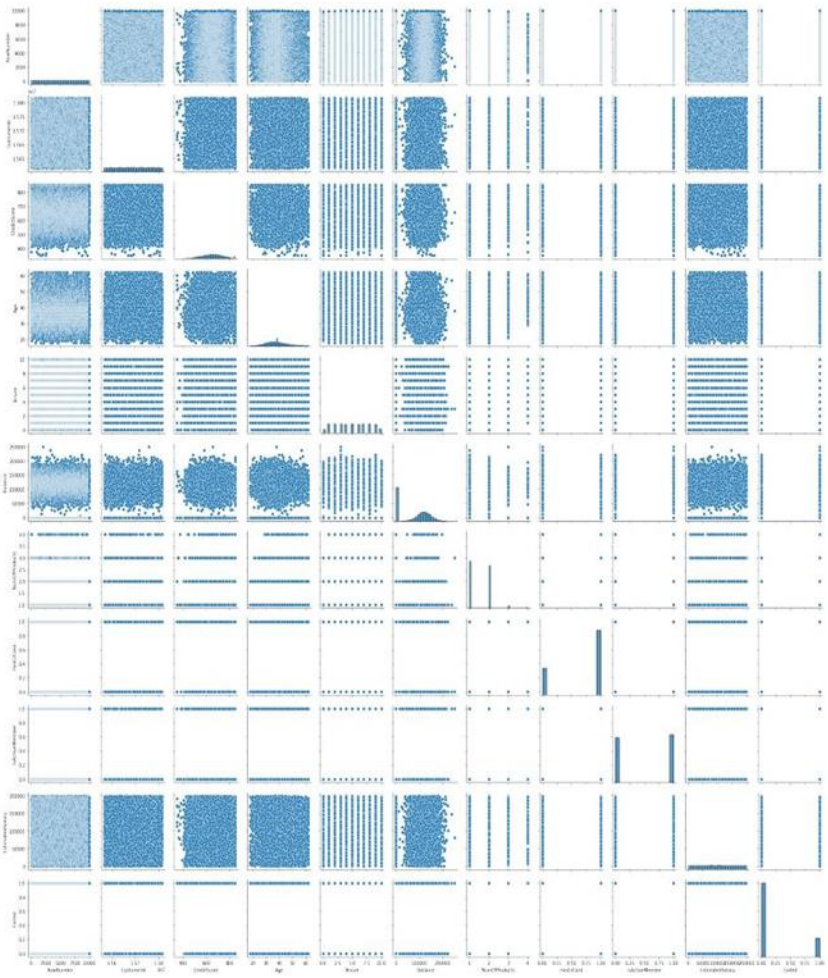

```
#Bivarient Analysis
sns.displot(data, x="Age", y="CreditScore")
```

```
<seaborn.axisgrid.FacetGrid at 0x7fd61f90c950>
```



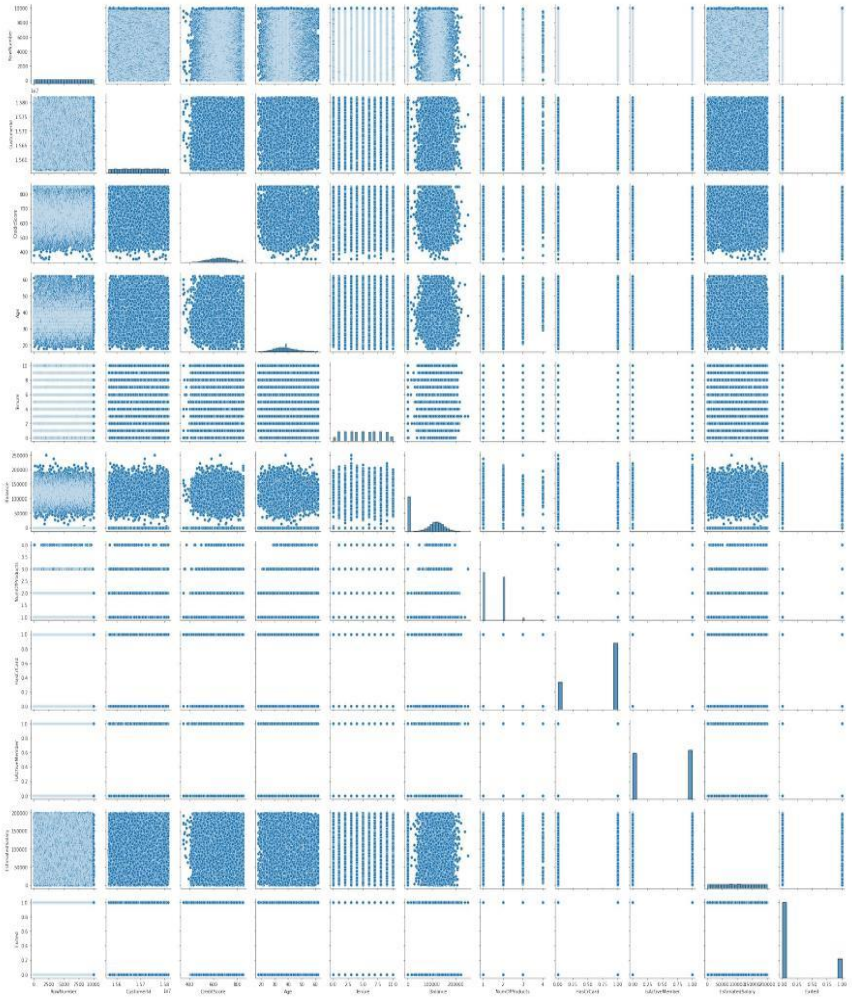

```
#Multi-varient Analysis
sns.pairplot(data)
```

```
<seaborn.axisgrid.PairGrid at 0x7fd621972dd0>
```



```
#gender Categorical column
data.tail()
```

```
      RowNumber  CustomerId    Surname  CreditScore Geography  Gender
Age  \
9995       9996    15606229   Obijiaku          771    France    Male
39.0
9996       9997    15569892  Johnstone          516    France    Male
35.0
9997       9998    15584532        Liu          709    France  Female
36.0
9998       9999    15682355  Sabbatini          772   Germany    Male
42.0
9999      10000    15628319     Walker          792    France  Female
28.0

      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
```



```
#gender Categorical column
data.tail()
```

```
      RowNumber  CustomerId    Surname  CreditScore Geography  Gender
Age  \
9995       9996    15606229   Obijiaku          771    France    Male
39.0
9996       9997    15569892  Johnstone          516    France    Male
35.0
9997       9998    15584532        Liu          709    France  Female
36.0
9998       9999    15682355  Sabbatini          772   Germany    Male
42.0
9999      10000    15628319     Walker          792    France  Female
28.0

      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
```

```
9995       5       0.00              2          1               0
9996      10   57369.61              1          1               1
9997       7       0.00              1          0               1
9998       3   75075.31              2          1               0
9999       4  130142.79              1          1               0

      EstimatedSalary  Exited
9995         96270.64       0
9996        101699.77       0
9997         42085.58       1
9998         92888.52       1
9999         38190.78       0
```

```
#Encoding
data['Gender'].replace({'Female':1,'Male':0},inplace=True)
data.tail()
```

```
      RowNumber  CustomerId    Surname  CreditScore Geography  Gender
Age  \
9995       9996    15606229   Obijiaku          771    France       0
39.0
9996       9997    15569892  Johnstone          516    France       0
35.0
9997       9998    15584532        Liu          709    France       1
36.0
9998       9999    15682355  Sabbatini          772   Germany       0
42.0
9999      10000    15628319     Walker          792    France       1
28.0

      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
9995       5       0.00              2          1               0
9996      10   57369.61              1          1               1
9997       7       0.00              1          0               1
9998       3   75075.31              2          1               0
9999       4  130142.79              1          1               0

      EstimatedSalary  Exited
9995         96270.64       0
9996        101699.77       0
9997         42085.58       1
9998         92888.52       1
9999         38190.78       0
```

```
data_main=pd.get_dummies(data,columns=['Geography'])

data_main
```

```
      RowNumber  CustomerId    Surname  CreditScore  Gender   Age
Tenure  \
0             1    15634602   Hargrave          619       1  42.0
```

```
2
1             2    15647311       Hill          608       1  41.0
1
2             3    15619304       Onio          502       1  42.0
8
3             4    15701354       Boni          699       1  39.0
1
4             5    15737888   Mitchell          850       1  43.0
2
...         ...         ...        ...          ...     ...   ...
...
9995       9996    15606229   Obijiaku          771       0  39.0
5
9996       9997    15569892  Johnstone          516       0  35.0
10
9997       9998    15584532        Liu          709       1  36.0
7
9998       9999    15682355  Sabbatini          772       0  42.0
3
9999      10000    15628319     Walker          792       1  28.0
4

        Balance  NumOfProducts  HasCrCard  IsActiveMember
EstimatedSalary  \
0          0.00              1          1               1
101348.88
1      83807.86              1          0               1
112542.58
2     159660.80              3          1               0
113931.57
3          0.00              2          0               0
93826.63
4     125510.82              1          1               1
79084.10
...         ...            ...        ...             ...
...
9995       0.00              2          1               0
96270.64
9996   57369.61              1          1               1
101699.77
9997       0.00              1          0               1
42085.58
9998   75075.31              2          1               0
92888.52
9999  130142.79              1          1               0
38190.78

      Exited  Geography_France  Geography_Germany  Geography_Spain
0          1                 1                  0                0
1          0                 0                  0                1
```

```
2          1                1                  0                0
3          0                1                  0                0
4          0                0                  0                1
...      ...              ...                ...              ...
9995       0                1                  0                0
9996       0                1                  0                0
9997       1                1                  0                0
9998       1                0                  1                0
9999       0                1                  0                0

[10000 rows x 16 columns]
```

```
#splitthe data into dependent and independent variables
y=data_main['Exited']
y.head()
```

```
0    1
1    0
2    1
3    0
4    0
Name: Exited, dtype: int64
```

```
x=data_main.drop(columns=['Exited'],axis=1)
x.head()
```

```
   RowNumber  CustomerId   Surname  CreditScore  Gender   Age  Tenure
\
0          1    15634602  Hargrave          619       1  42.0       2

1          2    15647311      Hill          608       1  41.0       1

2          3    15619304      Onio          502       1  42.0       8

3          4    15701354      Boni          699       1  39.0       1

4          5    15737888  Mitchell          850       1  43.0       2


     Balance  NumOfProducts  HasCrCard  IsActiveMember
EstimatedSalary  \
0       0.00              1          1               1
101348.88
1   83807.86              1          0               1
112542.58
2  159660.80              3          1               0
113931.57
3       0.00              2          0               0
93826.63
4  125510.82              1          1               1
```

```
79084.10
```

```
   Geography_France  Geography_Germany  Geography_Spain
0                 1                  0                0
1                 0                  0                1
2                 1                  0                0
3                 1                  0                0
4                 0                  0                1
```

```
#Scale the independent variables
x=data_main.drop(columns=['Surname',],axis=1)
x.head()
```

```
   RowNumber  CustomerId  CreditScore  Gender   Age  Tenure    Balance
\
0          1    15634602          619       1  42.0       2       0.00

1          2    15647311          608       1  41.0       1   83807.86

2          3    15619304          502       1  42.0       8  159660.80

3          4    15701354          699       1  39.0       1       0.00

4          5    15737888          850       1  43.0       2  125510.82


   NumOfProducts  HasCrCard  IsActiveMember  EstimatedSalary
Exited  \
0              1          1               1        101348.88       1

1              1          0               1        112542.58       0

2              3          1               0        113931.57       1

3              2          0               0         93826.63       0

4              1          1               1         79084.10       0


   Geography_France  Geography_Germany  Geography_Spain
0                 1                  0                0
1                 0                  0                1
2                 1                  0                0
3                 1                  0                0
4                 0                  0                1
```

```
from sklearn.preprocessing import scale
x=scale(x)
x
```

```
array([[-1.73187761, -0.78321342, -0.32622142, ...,  0.99720391,
        -0.57873591, -0.57380915],
       [-1.7315312 , -0.60653412, -0.44003595, ..., -1.00280393,
        -0.57873591,  1.74273971],
       [-1.73118479, -0.99588476, -1.53679418, ...,  0.99720391,
        -0.57873591, -0.57380915],
       ...,
       [ 1.73118479, -1.47928179,  0.60498839, ...,  0.99720391,
        -0.57873591, -0.57380915],
       [ 1.7315312 , -0.11935577,  1.25683526, ..., -1.00280393,
         1.72790383, -0.57380915],
       [ 1.73187761, -0.87055909,  1.46377078, ...,  0.99720391,
        -0.57873591, -0.57380915]])
```

```
#split the  data into training and testing
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test =
train_test_split(x,y,test_size=0.2,random_state=0)

x_train.shape
```

```
(8000, 15)
```

```
x_test.shape
```

```
(2000, 15)
```

```
y_train.shape
```

```
(8000,)
```

```
y_test.shape
```

```
(2000,)
```