DATE : 13 OCTOBER 2022

TEAM ID : PNT2022TMID25540

PROJECT NAME : NATURAL DISASTER INTENSITY ANALYSIS AND CLASSIFICATION

DATA PROCESSING

1.DOWNLOAD THE DATASET

The given dataset has been downloaded successfully

2.LOAD THE DATASET

```
import pandas as pd
data=pd.read_csv("Churn_Modelling.csv")
data.head()
```

```
   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


   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

   EstimatedSalary  Exited
0        101348.88       1
1        112542.58       0
2        113931.57       1
3         93826.63       0
4         79084.10       0
```

3 A)UNI VARIATE ANALYSIS

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns


data=pd.read_csv("Churn_Modelling.csv")
```

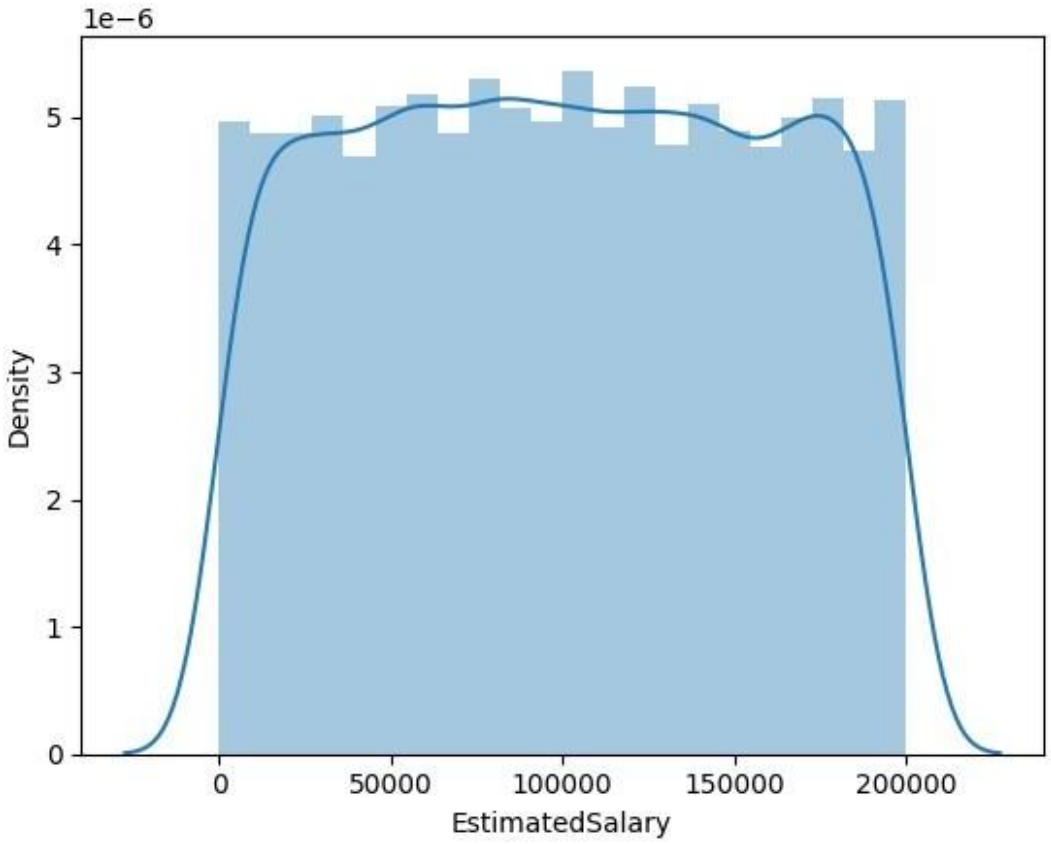
```
data.head()
```

```
   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


   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

   EstimatedSalary  Exited
0        101348.88       1
1        112542.58       0
2        113931.57       1
3         93826.63       0
4         79084.10       0
```

```
sns.distplot(data['EstimatedSalary'])
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_17588\3321298747.py:1:
UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `histplot` (an axes-level function for
histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['EstimatedSalary'])
```

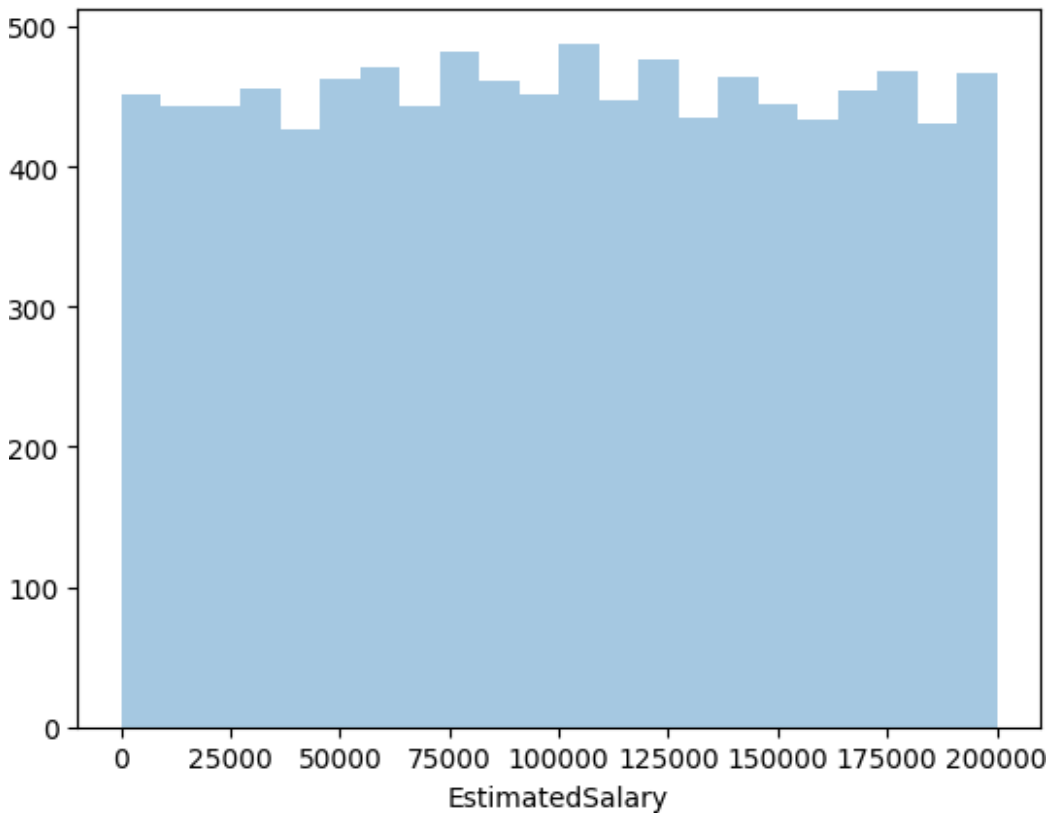
```
<AxesSubplot: xlabel='EstimatedSalary', ylabel='Density'>
```




```
sns.distplot(data['EstimatedSalary'],kde=False)
```

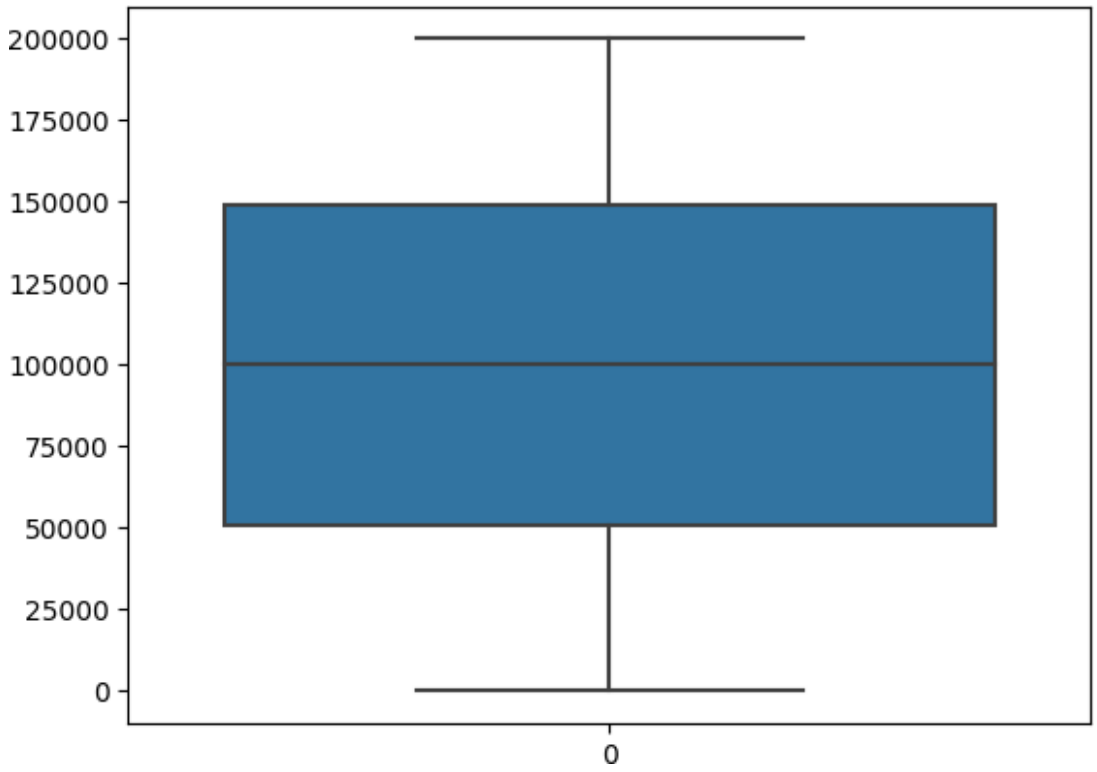
```
C:\Users\Dell\AppData\Local\Temp\ipykernel_17588\3654008557.py:1:
UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `histplot` (an axes-level function for
histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['EstimatedSalary'],kde=False)
```

```
<AxesSubplot: xlabel='EstimatedSalary'>
```



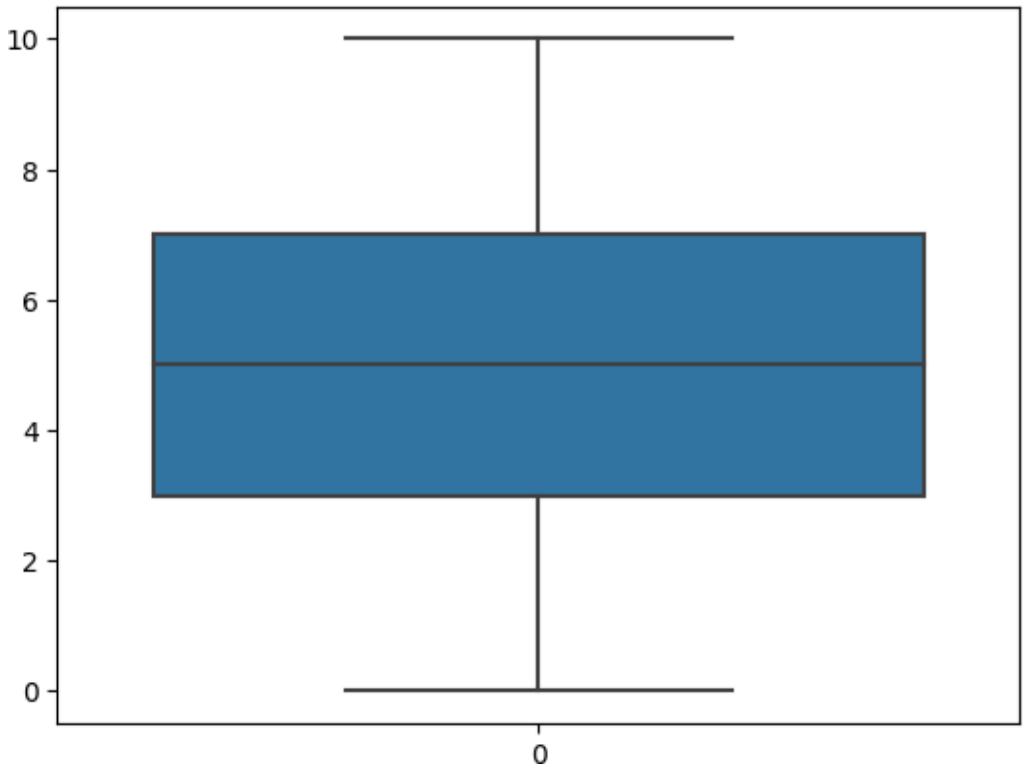

```
sns.boxplot(data['EstimatedSalary']),
```

```
(<AxesSubplot: >,)
```




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

```
(<AxesSubplot: >,)
```



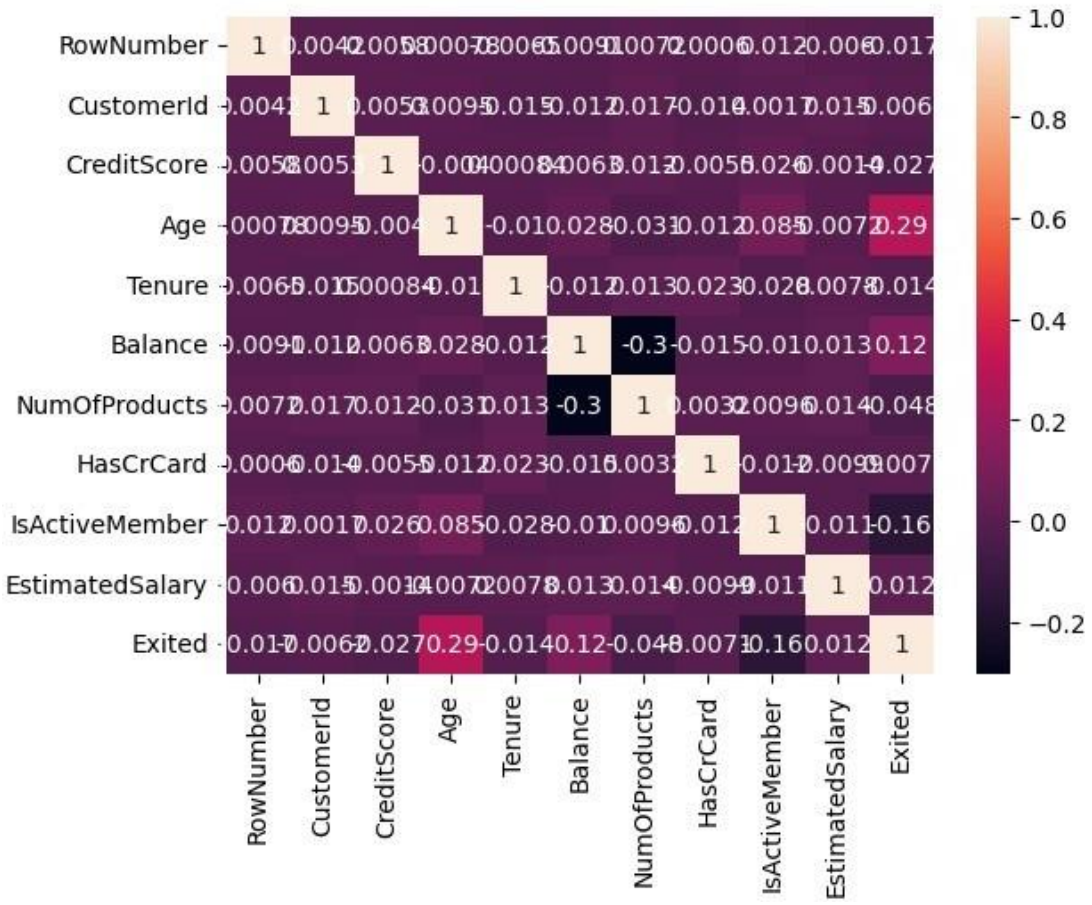
3 B)BI-VARIATE ANALYSIS

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")

data=pd.read_csv("Churn_Modelling.csv")

data.head()
```

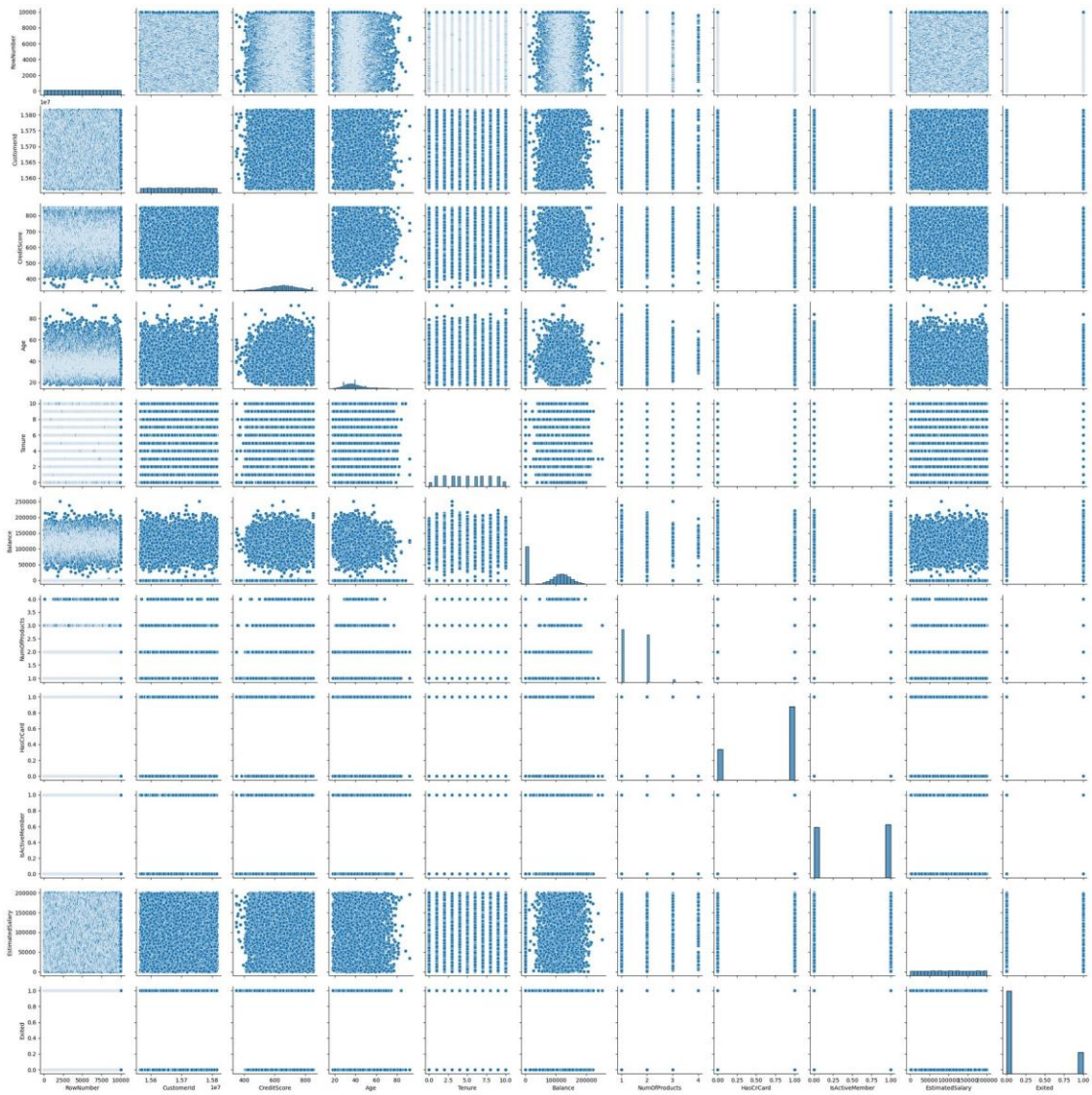
```
   RowNumber  CustomerId   Surname  CreditScore Geography  Gender  Age
\
0          1    15634602  Hargrave          619    France  Female   42

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
```

```
   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

   EstimatedSalary  Exited
0        101348.88       1
1        112542.58       0
2        113931.57       1
3         93826.63       0
4         79084.10       0
```

```
sns.heatmap(data.corr(),annot=True)
plt.show()
```




```
sns.pairplot(data)
plt.show()
```



3 C)MULTI-VARIATE ANALYSIS

```
from pydoc import help
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import scale
from sklearn.decomposition import PCA
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from scipy import stats
from IPython.display import display,HTML
%matplotlib inline
np.set_printoptions(suppress=True)
pd.set_option('display.max_rows',20)
import os
print(os.listdir("../NT project/"))
```

```
['Churn_Modelling.csv', 'datapro.ipynb', 'project.ipynb']
```

```
data=pd.read_csv("Churn_Modelling.csv")
data.head()
```

```
   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


   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

   EstimatedSalary  Exited
0        101348.88       1
1        112542.58       0
2        113931.57       1
3         93826.63       0
4         79084.10       0
```

```
data.columns
```

```
Index(['RowNumber', 'CustomerId', 'Surname', 'CreditScore',
'Geography',
       'Gender', 'Age', 'Tenure', 'Balance', 'NumOfProducts',
'HasCrCard',
       'IsActiveMember', 'EstimatedSalary', 'Exited'],
      dtype='object')
```

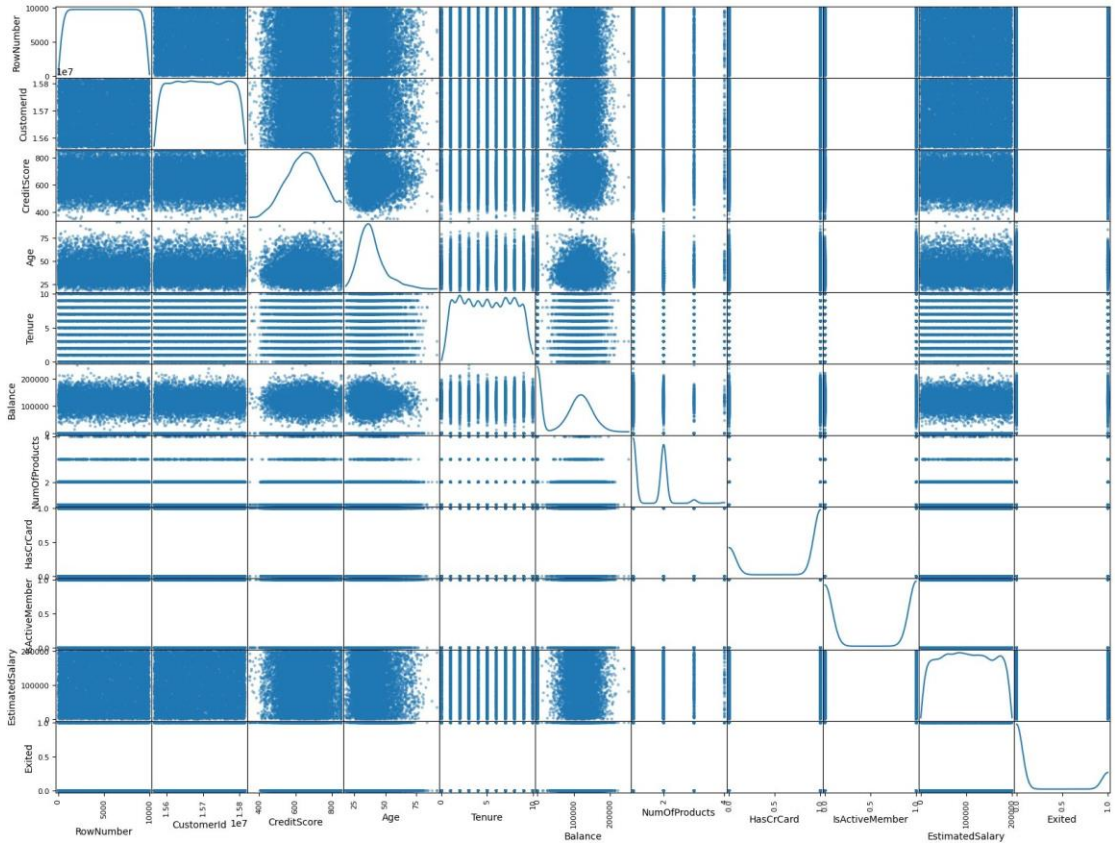
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   RowNumber        10000 non-null  int64
 1   CustomerId       10000 non-null  int64
 2   Surname          10000 non-null  object
```

```
 3   CreditScore      10000 non-null  int64
 4   Geography        10000 non-null  object
 5   Gender           10000 non-null  object
 6   Age              10000 non-null  int64
 7   Tenure           10000 non-null  int64
 8   Balance          10000 non-null  float64
 9   NumOfProducts    10000 non-null  int64
 10  HasCrCard        10000 non-null  int64
 11  IsActiveMember   10000 non-null  int64
 12  EstimatedSalary  10000 non-null  float64
 13  Exited           10000 non-null int64
dtypes: float64(2), int64(9), object(3)
memory usage: 1.1+ MB
```

MATRIX SCATTERPLOT

```
pd.plotting.scatter_matrix(data.loc[:,"RowNumber":"Exited"],diagonal="
kde",figsize=(20,15))
plt.show()
```



4.DESCRIPTIVE STATISTICS

```
import numpy as np
import pandas as pd
from pandas import Series,DataFrame
```

```
import scipy
from scipy import stats

data=pd.read_csv("Churn_Modelling.csv")
data.head()
```

```
   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


   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

   EstimatedSalary  Exited
0        101348.88       1
1        112542.58       0
2        113931.57       1
3         93826.63       0
4         79084.10       0
```

LOOKING AT SUMMARY STATISTICS THAT DESCRIBE A VARIABLE'S NUMERIC VALUES

```
data.sum()
```

```
RowNumber                                                 50005000
CustomerId                                           156909405694
Surname            HargraveHillOnioBoniMitchellChuBartlettObinnaH...
CreditScore                                                6505288
Geography          FranceSpainFranceFranceSpainSpainFranceGermany...
Gender             FemaleFemaleFemaleFemaleFemaleMaleMaleFemaleMa...
Age                                                         389218
Tenure                                                       50128
Balance                                                764858892.88
NumOfProducts                                                15302
HasCrCard                                                     7055
IsActiveMember                                                5151
EstimatedSalary                                      1000902398.81
```

```
Exited                                                          2037
dtype: object
```

```
data.sum(axis=1)
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_17588\1923841176.py:1:
FutureWarning: Dropping of nuisance columns in DataFrame reductions
(with 'numeric_only=None') is deprecated; in a future version this
will raise TypeError.  Select only valid columns before calling the
reduction.
  data.sum(axis=1)
```

```
0       15736618.88
1       15844315.44
2       15893456.37
3       15795925.63
4       15943385.92
           ...
9995    15713313.64
9996    15739522.38
9997    15637370.58
9998    15861138.83
9999    15807478.57
Length: 10000, dtype: float64
```

```
data.median()
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_17588\4184645713.py:1:
FutureWarning: The default value of numeric_only in DataFrame.median
is deprecated. In a future version, it will default to False. In
addition, specifying 'numeric_only=None' is deprecated. Select only
valid columns or specify the value of numeric_only to silence this
warning.
  data.median()
```

```
RowNumber          5.000500e+03
CustomerId         1.569074e+07
CreditScore        6.520000e+02
Age                3.700000e+01
Tenure             5.000000e+00
Balance            9.719854e+04
NumOfProducts      1.000000e+00
HasCrCard          1.000000e+00
IsActiveMember     1.000000e+00
EstimatedSalary    1.001939e+05
Exited             0.000000e+00
dtype: float64
```

```
data.mean()
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_17588\531903386.py:1:
FutureWarning: The default value of numeric_only in DataFrame.mean is
```

```
deprecated. In a future version, it will default to False. In
addition, specifying 'numeric_only=None' is deprecated. Select only
valid columns or specify the value of numeric_only to silence this
warning.
  data.mean()
```

```
RowNumber          5.000500e+03
CustomerId         1.569094e+07
CreditScore        6.505288e+02
Age                3.892180e+01
Tenure             5.012800e+00
Balance            7.648589e+04
NumOfProducts      1.530200e+00
HasCrCard          7.055000e-01
IsActiveMember     5.151000e-01
EstimatedSalary    1.000902e+05
Exited             2.037000e-01
dtype: float64
```

```
data.max()
```

```
RowNumber              10000
CustomerId          15815690
Surname                Zuyeva
CreditScore               850
Geography               Spain
Gender                   Male
Age                        92
Tenure                     10
Balance            250898.09
NumOfProducts              4
HasCrCard                  1
IsActiveMember             1
EstimatedSalary    199992.48
Exited                     1
dtype: object
```

```
mpg=data.EstimatedSalary
mpg.idxmax()
```

```
6646
```

LOOKING AT SUMMARY STATISTICS THAT DESCRIBE VARIABLE DISTRIBUTION

```
data.std()
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_17588\2723740006.py:1:
FutureWarning: The default value of numeric_only in DataFrame.std is
deprecated. In a future version, it will default to False. In
addition, specifying 'numeric_only=None' is deprecated. Select only
valid columns or specify the value of numeric_only to silence this
```

```
warning.
  data.std()
```

```
RowNumber           2886.895680
CustomerId         71936.186123
CreditScore           96.653299
Age                   10.487806
Tenure                 2.892174
Balance            62397.405202
NumOfProducts          0.581654
HasCrCard              0.455840
IsActiveMember         0.499797
EstimatedSalary    57510.492818
Exited                 0.402769
dtype: float64
```

```
data.var()
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_17588\445316826.py:1:
FutureWarning: The default value of numeric_only in DataFrame.var is
deprecated. In a future version, it will default to False. In
addition, specifying 'numeric_only=None' is deprecated. Select only
valid columns or specify the value of numeric_only to silence this
warning.
  data.var()
```

```
RowNumber          8.334167e+06
CustomerId         5.174815e+09
CreditScore        9.341860e+03
Age                1.099941e+02
Tenure             8.364673e+00
Balance            3.893436e+09
NumOfProducts      3.383218e-01
HasCrCard          2.077905e-01
IsActiveMember     2.497970e-01
EstimatedSalary    3.307457e+09
Exited             1.622225e-01
dtype: float64
```

```
num=data.NumOfProducts
num.value_counts()
```

```
1    5084
2    4590
3     266
4      60
Name: NumOfProducts, dtype: int64
```

```
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
```

5.HANDLE MISSING VALUE

```
import pandas as pd

data=pd.read_csv("Churn_Modelling.csv")
data.head()
```

```
   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


   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

   EstimatedSalary  Exited
0        101348.88       1
1        112542.58       0
2        113931.57       1
3         93826.63       0
4         79084.10       0

data.shape

(10000, 14)

data.isnull()

      RowNumber  CustomerId  Surname  CreditScore  Geography  Gender
Age  \
0         False       False    False        False      False   False
False
1         False       False    False        False      False   False
False
2         False       False    False        False      False   False
False
3         False       False    False        False      False   False
False
4         False       False    False        False      False   False
False
...         ...         ...      ...          ...        ...     ...
...
9995      False       False    False        False      False   False
False
9996      False       False    False        False      False   False
False
9997      False       False    False        False      False   False
False
9998      False       False    False        False      False   False
False
9999      False       False    False        False      False   False
False
```

```
      Tenure  Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0      False    False          False      False           False
1      False    False          False      False           False
2      False    False          False      False           False
3      False    False          False      False           False
4      False    False          False      False           False
...      ...      ...            ...        ...             ...
9995   False    False          False      False           False
9996   False    False          False      False           False
9997   False    False          False      False           False
9998   False    False          False      False           False
9999   False    False          False      False           False

      EstimatedSalary  Exited
0               False   False
1               False   False
2               False   False
3               False   False
4               False   False
...               ...     ...
9995            False   False
9996            False   False
9997            False   False
9998            False   False
9999            False   False

[10000 rows x 14 columns]
```

```
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
```

```
data.isnull().sum().sum()
```

```
0
```

FILLING NULL VALUES

```
df=data.fillna(value=0)
df
```

```
      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]
```

```
df.isnull().sum().sum()
```

```
0
```

```
df1=data.fillna(value=5)
df1
```

```
      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]
```

FILLING NULL VALUES WITH A PREVIOUS VALUE

```
df2=data.fillna(method='pad')
df2
```

```
      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]
```

```
df2.isnull().sum().sum()
```

```
0
```

```
#filling NULL values with the next value
df3=data.fillna(method='bfill')
df3
```

```
      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]
```

DROPPING NULL VALUES

```
df4=data.dropna()
df4
```

```
      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]
```

```
df5=data.dropna(how='any')
df5
```

```
      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]
```

replace()

```
import numpy as np
df6=df.replace(to_replace=np.nan,value=8763)
df6
```

```
      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]
```

interpolate()

```
data['EstimatedSalary']=data['EstimatedSalary'].interpolate(method='li
near')
data
```

```
      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]
```

6.FIND THE OUTLIERS AND REPLACE THE OUTLIERS

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

data=pd.read_csv("Churn_Modelling.csv")
data1=data["CreditScore"]
outliers=[]
def detect_outliers(data):
    threshold=3
    mean=np.mean(data)
    std=np.std(data)
    for i in data:
        z_score=(i-mean)/std
        if np.abs(z_score)>threshold:
            outliers.append(z_score)
    return outliers

outlier_pt=detect_outliers(data1)

outlier_pt

[]
```

INTERQUANTILE RANGE

```
sorted(data1)

[350,
 350,
 350,
 350,
 350,
 351,
 358,
```

359,
363,
365,
367,
373,
376,
376,
382,
383,
386,
395,
399,
401,
404,
405,
405,
407,
408,
408,
410,
410,
410,
411,
411,
411,
411,
412,
413,
413,
413,
414,
414,
415,
415,
415,
416,
416,
416,
416,
417,
418,
418,
418,
418,
418,
418,
419,
420,
420,
420,

421,
421,
421,
422,
422,
422,
422,
423,
424,
425,
425,
425,
425,
426,
426,
427,
427,
427,
427,
427,
427,
428,
428,
428,
429,
429,
429,
429,
430,
430,
430,
430,
430,
430,
430,
430,
430,
431,
431,
431,
431,
431,
432,
432,
432,
432,
432,
433,
433,
433,

434,
434,
434,
434,
434,
434,
435,
435,
435,
435,
435,
436,
436,
437,
437,
437,
437,
437,
437,
438,
438,
438,
438,
438,
438,
439,
439,
439,
439,
439,
439,
439,
440,
441,
442,
443,
443,
443,
443,
443,
443,
443,
443,
443,
444,
444,
444,
444,
444,
444,
444,
445,

445,
445,
445,
445,
445,
445,
445,
445,
445,
445,
445,
446,
446,
446,
446,
446,
446,
447,
447,
447,
447,
448,
448,
449,
449,
449,
449,
449,
449,
449,
450,
450,
450,
450,
451,
451,
451,
451,
451,
451,
452,
452,
452,
452,
452,
452,
452,
452,
452,
452,
452,
453,
453,
453,

453,
453,
453,
454,
454,
454,
454,
455,
455,
455,
455,
456,
456,
456,
456,
456,
457,
457,
457,
457,
457,
457,
457,
458,
458,
458,
458,
458,
458,
458,
459,
459,
459,
459,
459,
460,
460,
460,
460,
461,
461,
461,
461,
461,
461,
461,
461,
462,
462,
462,

462,
463,
463,
463,
463,
464,
464,
464,
464,
464,
464,
464,
465,
465,
465,
465,
465,
465,
465,
465,
465,
466,
466,
466,
466,
466,
466,
466,
466,
466,
466,
467,
467,
467,
467,
468,
468,
468,
468,
468,
468,
468,
468,
468,
468,
468,
469,
469,
469,
469,

469,
469,
469,
470,
470,
470,
470,
470,
470,
470,
470,
470,
470,
471,
471,
471,
471,
471,
471,
471,
471,
472,
472,
472,
472,
472,
472,
472,
472,
472,
473,
473,
473,
473,
473,
473,
473,
473,
474,
474,
474,
474,
474,
474,
474,
474,
474,
474,
475,
475,
475,
475,
475,
475,
475,

476,
476,
476,
476,
476,
476,
476,
476,
476,
477,
477,
477,
477,
477,
477,
477,
477,
477,
477,
477,
477,
477,
478,
478,
478,
478,
478,
478,
478,
478,
479,
479,
479,
479,
479,
479,
479,
479,
479,
479,
479,
479,
479,
479,
479,
479,
479,
479,
479,
480,
480,
480,
480,
480,

480,
480,
480,
480,
481,
481,
481,
481,
481,
481,
481,
481,
481,
481,
481,
482,
482,
482,
482,
482,
482,
483,
483,
483,
483,
483,
483,
483,
484,
484,
484,
484,
484,
484,
484,
484,
484,
484,
484,
485,
485,
485,
485,
485,
485,
485,
485,
485,
485,
485,
485,

486,
486,
486,
486,
486,
486,
486,
486,
486,
487,
487,
487,
487,
487,
487,
487,
487,
487,
487,
487,
487,
487,
487,
488,
488,
488,
488,
488,
488,
488,
488,
488,
488,
489,
489,
489,
489,
489,
489,
489,
489,
489,
489,
489,
489,
490,
490,
490,
490,
490,

490,
490,
490,
490,
490,
490,
490,
490,
491,
491,
491,
491,
491,
491,
491,
491,
491,
491,
492,
492,
492,
492,
492,
492,
492,
492,
492,
492,
492,
492,
492,
492,
492,
493,
493,
493,
493,
493,
493,
493,
493,
493,
493,
493,
493,
493,
494,
494,
494,
494,
494,

494,
494,
494,
494,
494,
494,
494,
494,
495,
495,
495,
495,
495,
495,
495,
495,
496,
496,
496,
496,
496,
496,
496,
496,
496,
496,
496,
496,
496,
497,
497,
497,
497,
497,
497,
497,
497,
497,
497,
497,
497,
497,
497,
497,
498,
498,
498,
498,
498,
498,
498,

498,
498,
498,
498,
498,
498,
498,
498,
498,
498,
499,
499,
499,
499,
499,
499,
499,
499,
499,
499,
499,
499,
499,
499,
499,
500,
500,
500,
500,
500,
500,
500,
500,
500,
500,
500,
501,
501,
501,
501,
501,
501,
501,
501,
501,
501,
501,
501,
501,

501,
501,
501,
501,
501,
502,
502,
502,
502,
502,
502,
502,
502,
502,
502,
502,
502,
502,
502,
503,
503,
503,
503,
503,
503,
503,
503,
503,
503,
503,
503,
503,
504,
504,
504,
504,
504,
504,
504,
504,
504,
504,
504,
504,
504,
504,
505,
505,
505,
505,

505,
505,
505,
505,
505,
505,
505,
505,
505,
505,
505,
505,
506,
506,
506,
506,
506,
506,
506,
506,
506,
506,
506,
506,
506,
506,
506,
506,
506,
506,
506,
506,
507,
507,
507,
507,
507,
507,
507,
507,
507,
507,
507,
507,
507,
507,
507,
507,
507,
507,
507,
507,
507,
508,
508,
508,
508,
508,

508,
508,
508,
508,
508,
508,
508,
508,
508,
508,
508,
508,
509,
509,
509,
509,
509,
509,
509,
509,
509,
509,
509,
509,
509,
509,
509,
509,
510,
510,
510,
510,
510,
510,
510,
510,
510,
510,
510,
510,
510,
510,
510,
510,
511,
511,
511,
511,
511,
511,
511,
511,
511,
511,
511,
511,
511,

511,
511,
511,
511,
512,
512,
512,
512,
512,
512,
512,
512,
512,
512,
512,
512,
512,
512,
512,
512,
512,
512,
512,
512,
512,
512,
513,
513,
513,
513,
513,
513,
513,
513,
513,
513,
513,
513,
513,
513,
513,
513,
513,
514,
514,
514,
514,
514,

514,
514,
514,
514,
514,
514,
515,
515,
515,
515,
515,
515,
515,
515,
515,
515,
515,
515,
515,
515,
515,
515,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,
516,

516,
516,
517,
517,
517,
517,
517,
517,
517,
517,
517,
517,
517,
518,
518,
518,
518,
518,
518,
518,
518,
518,
518,
518,
518,
518,
518,
518,
518,
518,
519,
519,
519,
519,
519,
519,
519,
519,
519,
519,
519,
519,
519,
519,
519,
520,
520,
520,
520,
520,
520,
520,

```
 520,
 520,
 520,
 520,
 520,
 520,
 520,
 520,
 520,
 520,
 520,
 520,
 520,
 520,
 520,
 520,
 520,
 520,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 521,
 ...]
```

```
quantile1,quantile3=np.percentile(data1,[25,75])
```

```
print(quantile1,quantile3)
```

```
584.0 718.0
```

```
iqr_value=quantile3-quantile1
print(iqr_value)
```

```
134.0
```

```
lower_bound_val=quantile1-(1.5*iqr_value)
upper_bound_val=quantile3+(1.5*iqr_value)

print(lower_bound_val,upper_bound_val)
```

```
383.0 919.0
```

1. CHECK FOR CATEGORICAL COLUMNS AND PERFORM ENCODING

```
import pandas as pd
import numpy as np
import seaborn as sns
%matplotlib inline
```

METHOD I

```
data=pd.read_csv("Churn_Modelling.csv")
NEW_DataM1=data
data1=pd.get_dummies(NEW_DataM1["Gender"])

data1.head()
```

```
   Female  Male
0       1     0
1       1     0
2       1     0
3       1     0
4       1     0
```

```
NEW_DataM1.drop('Gender',axis='columns')
```

```
      RowNumber  CustomerId    Surname  CreditScore Geography  Age
Tenure  \
0             1    15634602   Hargrave          619    France   42
2
1             2    15647311       Hill          608     Spain   41
1
2             3    15619304       Onio          502    France   42
8
3             4    15701354       Boni          699    France   39
1
4             5    15737888   Mitchell          850     Spain   43
2
...         ...         ...        ...          ...       ...  ...
...
9995       9996    15606229   Obijiaku          771    France   39
5
9996       9997    15569892  Johnstone          516    France   35
```

```
10
9997       9998    15584532        Liu          709    France   36
7
9998       9999    15682355  Sabbatini          772   Germany   42
3
9999      10000    15628319     Walker          792    France   28
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
0          1
1          0
2          1
3          0
4          0
...      ...
9995       0
9996       0
9997       1
9998       1
9999       0

[10000 rows x 13 columns]
```

```
NEW_DataM1["Male"]=data1["Male"].to_list()
NEW_DataM1["Female"]=data1["Female"].to_list()
```

NEW_DataM1

```
      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  Male  Female
0           101348.88       1     0       1
1           112542.58       0     0       1
2           113931.57       1     0       1
3            93826.63       0     0       1
4            79084.10       0     0       1
...               ...     ...   ...     ...
9995         96270.64       0     1       0
9996        101699.77       0     1       0
9997         42085.58       1     0       1
```

```
9998          92888.52       1     1       0
9999          38190.78       0     0       1

[10000 rows x 16 columns]
```

```
NEW_DataM1.head(2)
```

```
   RowNumber  CustomerId   Surname  CreditScore Geography  Gender  Age
\
0          1    15634602  Hargrave          619    France  Female   42

1          2    15647311      Hill          608     Spain  Female   41


   Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0       2       0.00              1          1               1
1       1   83807.86              1          0               1

   EstimatedSalary  Exited  Male  Female
0        101348.88       1     0       1
1        112542.58       0     0       1
```

METHOD II

```
from sklearn.preprocessing import LabelEncoder

data=pd.read_csv("Churn_Modelling.csv")
l3=LabelEncoder()
label=l3.fit_transform(data["Gender"])

l3.classes_

array(['Female', 'Male'], dtype=object)

Data=NEW_DataM1.drop("Gender",axis='columns')
Data
```

```
      RowNumber  CustomerId    Surname  CreditScore Geography  Age
Tenure  \
0             1    15634602   Hargrave          619    France   42
2
1             2    15647311       Hill          608     Spain   41
1
2             3    15619304       Onio          502    France   42
8
3             4    15701354       Boni          699    France   39
1
4             5    15737888   Mitchell          850     Spain   43
2
...         ...         ...        ...          ...       ...  ...
...
9995       9996    15606229   Obijiaku          771    France   39
```

```
5
9996       9997    15569892  Johnstone          516    France   35
10
9997       9998    15584532        Liu          709    France   36
7
9998       9999    15682355  Sabbatini          772   Germany   42
3
9999      10000    15628319     Walker          792    France   28
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  Male  Female
0          1     0       1
1          0     0       1
2          1     0       1
3          0     0       1
4          0     0       1
...      ...   ...     ...
9995       0     1       0
9996       0     1       0
9997       1     0       1
9998       1     1       0
9999       0     0       1

[10000 rows x 15 columns]
```

```
Data["Gender"]=label
Data
```

```
      RowNumber  CustomerId    Surname  CreditScore Geography  Age
Tenure  \
0             1    15634602   Hargrave          619    France   42
2
1             2    15647311       Hill          608     Spain   41
1
2             3    15619304       Onio          502    France   42
8
3             4    15701354       Boni          699    France   39
1
4             5    15737888   Mitchell          850     Spain   43
2
...         ...         ...        ...          ...       ...  ...
...
9995       9996    15606229   Obijiaku          771    France   39
5
9996       9997    15569892  Johnstone          516    France   35
10
9997       9998    15584532        Liu          709    France   36
7
9998       9999    15682355  Sabbatini          772   Germany   42
3
9999      10000    15628319     Walker          792    France   28
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  Male  Female  Gender
0          1     0       1       0
1          0     0       1       0
2          1     0       1       0
3          0     0       1       0
4          0     0       1       0
...      ...   ...     ...     ...
9995       0     1       0       1
9996       0     1       0       1
9997       1     0       1       0
9998       1     1       0       1
9999       0     0       1       0

[10000 rows x 16 columns]
```

8.SPLIT THE DATA INTO DEPENDENT AND INDEPENDENT VARIABLES

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

data=pd.read_csv("Churn_Modelling.csv")

X=data.iloc[:,2:9]
X
```

```
        Surname  CreditScore Geography  Gender  Age  Tenure    Balance
0      Hargrave          619    France  Female   42       2       0.00
1          Hill          608     Spain  Female   41       1   83807.86
2          Onio          502    France  Female   42       8  159660.80
3          Boni          699    France  Female   39       1       0.00
4      Mitchell          850     Spain  Female   43       2  125510.82
...         ...          ...       ...     ...  ...     ...        ...
9995   Obijiaku          771    France    Male   39       5       0.00
9996  Johnstone          516    France    Male   35      10   57369.61
9997        Liu          709    France  Female   36       7       0.00
9998  Sabbatini          772   Germany    Male   42       3   75075.31
9999     Walker          792    France  Female   28       4  130142.79

[10000 rows x 7 columns]
```

```
Y=data.iloc[:,9]
Y
```

```
0       1
1       1
2       3
3       2
```

```
4       1
       ..
9995    2
9996    1
9997    1
9998    2
9999    1
Name: NumOfProducts, Length: 10000, dtype: int64
```

9.SCALE THE INDEPENDENT VARIABLES

```
import numpy as np
import pandas as pd
from pandas import Series,DataFrame
import matplotlib.pyplot as plt
from pylab import rcParams
import seaborn as sb
import scipy
import sklearn
from sklearn import preprocessing
from sklearn.preprocessing import scale

%matplotlib inline
rcParams['figure.figsize']=5,4
sb.set_style('whitegrid')
```

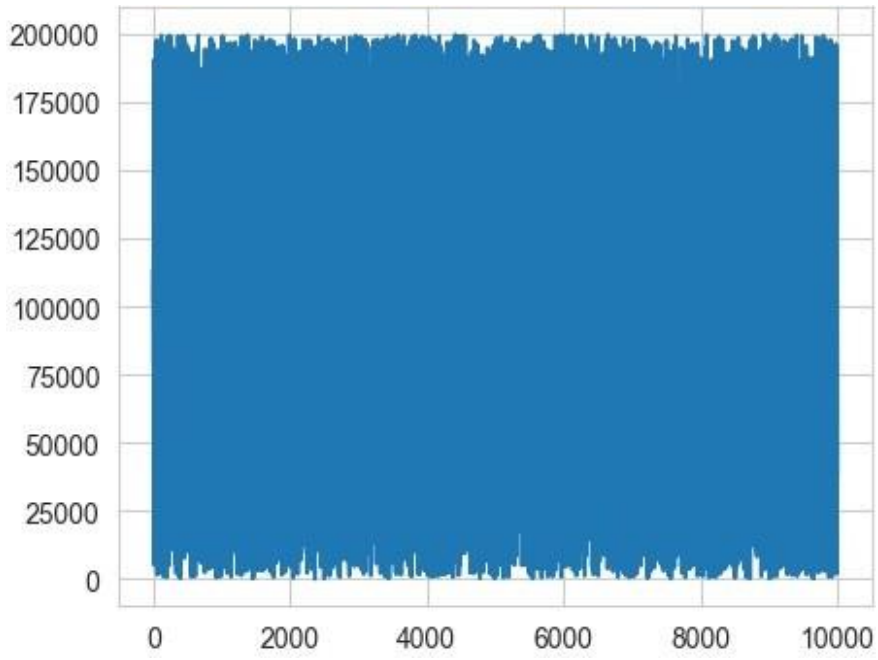
Normalizing and transfroming features with MinMaxScalar() and fit_transform()

```
data=pd.read_csv("Churn_Modelling.csv")

data.head()
```

```
   RowNumber  CustomerId   Surname ... IsActiveMember
EstimatedSalary Exited
0          1    15634602  Hargrave  ...               1
101348.88       1
1          2    15647311      Hill  ...               1
112542.58       0
2          3    15619304      Onio  ...               0
113931.57       1
3          4    15701354      Boni  ...               0
93826.63        0
4          5    15737888  Mitchell  ...               1
79084.10        0

[5 rows x 14 columns]
```

```
tenure=data.EstimatedSalary
plt.plot(tenure)
```

```
[<matplotlib.lines.Line2D at 0x14ec8f2b400>]
```

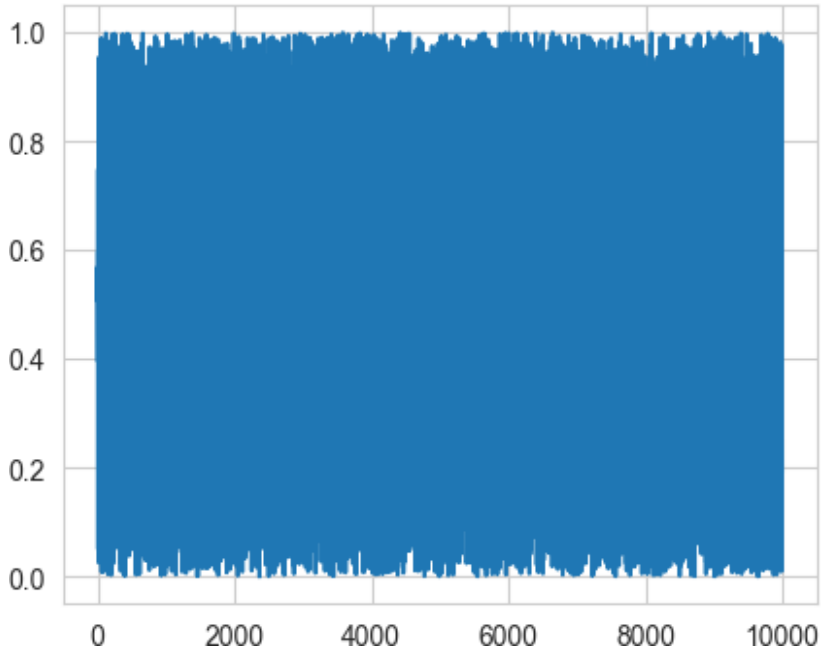



```
data[['Tenure']].describe()
```

```
             Tenure
count  10000.000000
mean       5.012800
std        2.892174
min        0.000000
25%        3.000000
50%        5.000000
75%        7.000000
max       10.000000
```

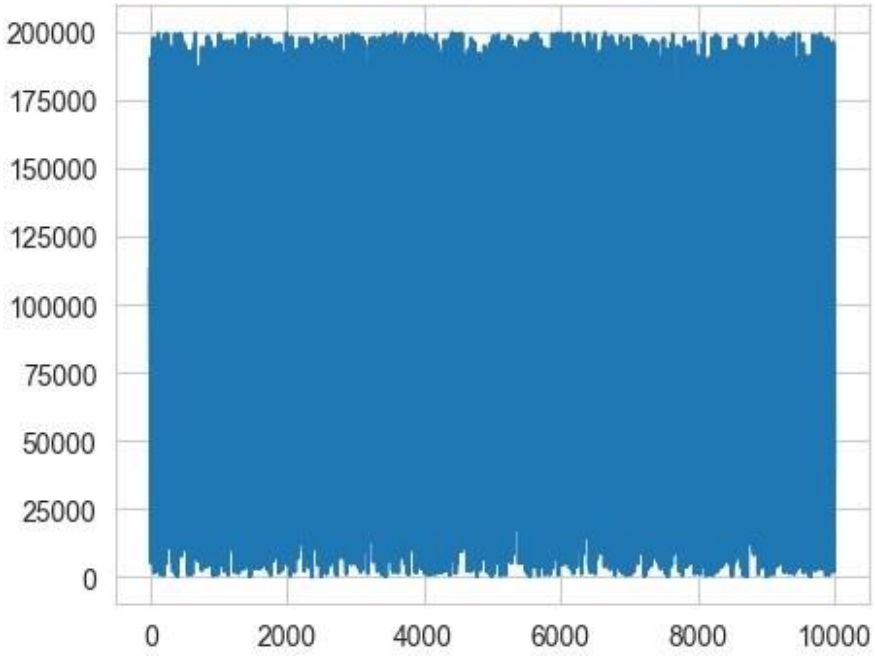
```
tenure_matrix=tenure.values.reshape(-1,1)
scaled=preprocessing.MinMaxScaler()
scaled_tenure=scaled.fit_transform(tenure_matrix)
plt.plot(scaled_tenure)
```

```
[<matplotlib.lines.Line2D at 0x14ec8dc02b0>]
```




```
std_tenure=scale(tenure,axis=0,with_mean=False,with_std=False)
plt.plot(std_tenure)
```

```
[<matplotlib.lines.Line2D at 0x14ec8ed07f0>]
```



10. SPLIT THE DATA INTO TRAINING AND TESTING

```
import pandas as pd
data=pd.read_csv("Churn_Modelling.csv")
```

```
data.describe()
```

```
          RowNumber    CustomerId  ...  EstimatedSalary        Exited
count  10000.00000  1.000000e+04  ...     10000.000000  10000.000000
mean    5000.50000  1.569094e+07  ...    100090.239881      0.203700
std     2886.89568  7.193619e+04  ...     57510.492818      0.402769
min        1.00000  1.556570e+07  ...        11.580000      0.000000
25%     2500.75000  1.562853e+07  ...     51002.110000      0.000000
50%     5000.50000  1.569074e+07  ...    100193.915000      0.000000
75%     7500.25000  1.575323e+07  ...    149388.247500      0.000000
max    10000.00000  1.581569e+07  ...    199992.480000      1.000000

[8 rows x 11 columns]
```

```
import numpy as np
```

```
x=np.array(data["CustomerId"]).reshape(-1,1)
x.shape
```

```
(10000, 1)
```

```
y=np.array(data["EstimatedSalary"])
y.shape
```

```
(10000,)
```

```
print(y)
```

```
[101348.88 112542.58 113931.57 ...  42085.58  92888.52  38190.78]
```

```
print(type(x))
```

```
<class 'numpy.ndarray'>
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30)
```

```
x_train.shape
```

```
(7000, 1)
```

```
x_test.shape
```

```
(3000, 1)
```

```
y_train.shape
```

```
(7000,)
```

```
y.shape
```

```
(10000,)
```

```
print(y_train.shape)
```

```
(7000,)
print(y_test.shape)
(3000,)
```