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.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
			OfProducts	HasCrCard	IsActiveMe	mber \	
0	2	0.00	1	1		1	

	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.

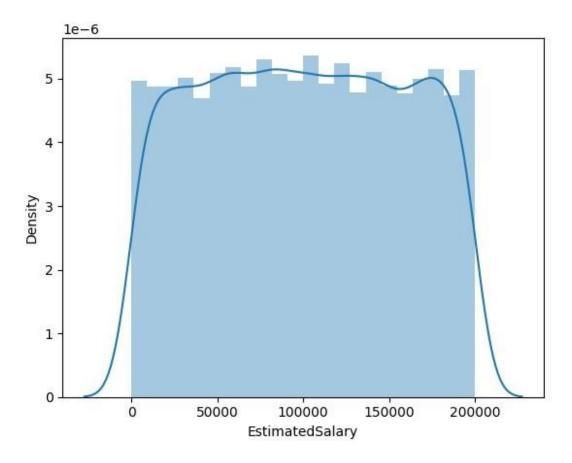
Please adapt your code to use either `displot` (a figure-level function with

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:

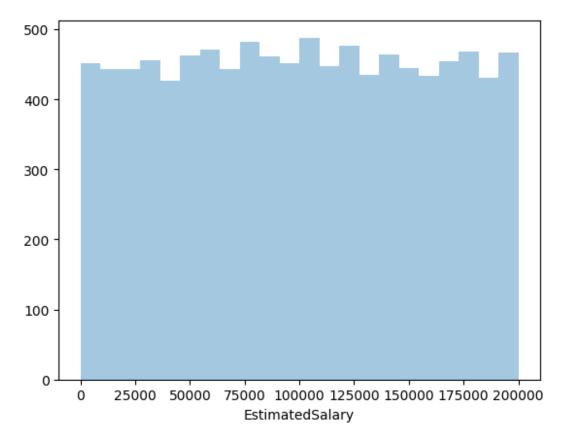
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

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

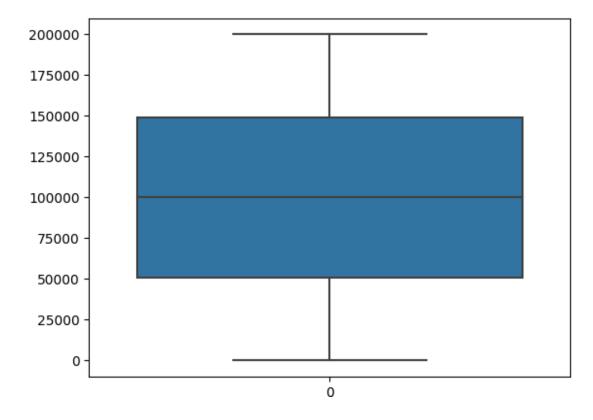
For a guide to updating your code to use the new functions, please see https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

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

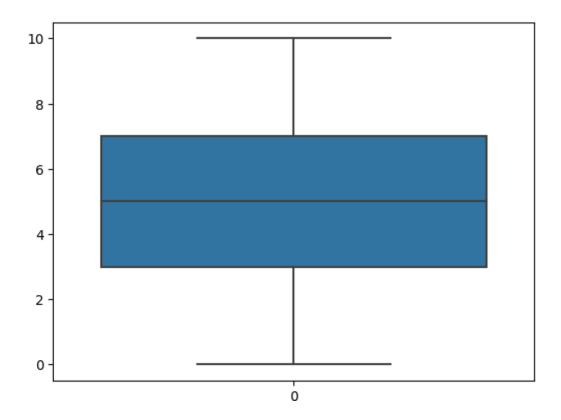
<AxesSubplot: xlabel='EstimatedSalary'>



```
sns.boxplot(data['EstimatedSalary']),
  (<AxesSubplot: >,)
```



```
sns.boxplot(data['Tenure']),
(<AxesSubplot: >,)
```



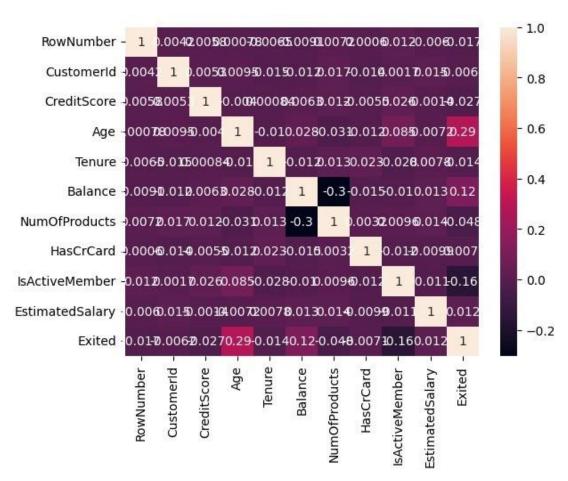
3 B)BI-VARIATE ANALYSIS

```
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()
```

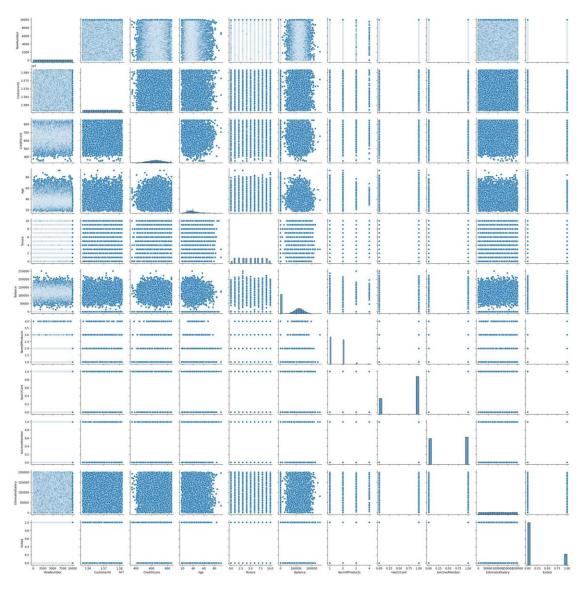
\	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age
0	1	15634602	Hargrave	619	France	Female	42
1	2	15647311	Hill	608	Spain	Female	41
2	3	15619304	Onio	502	France	Female	42
3	4	15701354	Boni	699	France	Female	39
4	5	15737888	Mitchell	850	Spain	Female	43

```
Balance
                         NumOfProducts
                                         HasCrCard
                                                      IsActiveMember
   Tenure
0
                  0.00
         2
                                                                      1
             83807.86
1
         1
                                       1
                                                    0
                                                                       1
2
            159660.80
                                       3
                                                                      0
         8
                                                    1
3
                  0.00
                                       2
                                                    0
         1
                                                                      0
4
            125510.82
                                                    1
   EstimatedSalary Exited
0
          101348.88
                             1
1
          112542.58
                             \Omega
2
          113931.57
                             1
3
           93826.63
                             0
4
           79084.10
                             \Omega
sns.heatmap(data.corr(),annot=True)
```

plt.show()



sns.pairplot(data) plt.show()



3 C)MULTI-VARIATE ANALYSIS

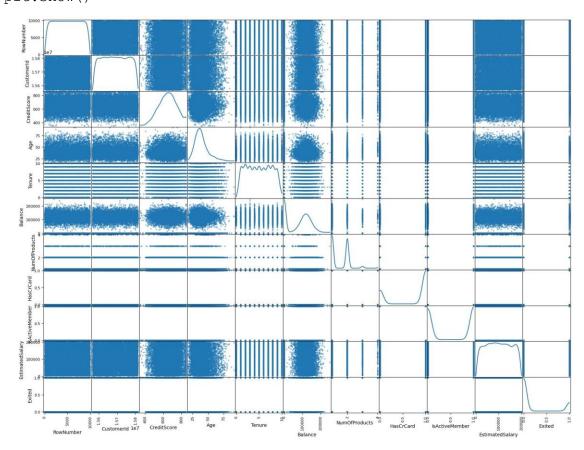
```
from pydoc import help
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import scale
from sklearn.decomposition import PCA
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from scipy import stats
from IPython.display 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
               15634602 Hargrave
0
                                                                   42
          1
                                           619
                                                  France
                                                         Female
1
               15647311
                             Hill
                                           608
                                                   Spain Female
                                                                  41
2
               15619304
                             Onio
                                           502
                                                  France Female
                                                                  42
3
          4
               15701354
                             Boni
                                           699
                                                  France Female
                                                                  39
4
               15737888 Mitchell
                                           850
                                                   Spain Female
                                                                  43
   Tenure
            Balance NumOfProducts HasCrCard IsActiveMember
0
       2
               0.00
                                1
                                            1
                                                           1
       1 83807.86
                                            0
1
                                 1
                                                           1
2
       8 159660.80
                                 3
                                            1
                                                           0
3
       1
               0.00
                                 2
                                            0
                                                           0
4
          125510.82
                                            1
                                 1
  EstimatedSalary Exited
0
        101348.88
1
        112542.58
                        0
2
        113931.57
                        1
3
         93826.63
                        0
         79084.10
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
    _____
    RowNumber
                     10000 non-null int64
 0
 1
    CustomerId
                     10000 non-null int64
    Surname
                     10000 non-null object
```

```
10000 non-null int64
 3
    CreditScore
 4
                     10000 non-null object
    Geography
 5
    Gender
                     10000 non-null object
 6
                     10000 non-null int64
    Age
 7
    Tenure
                     10000 non-null int64
8
    Balance
                     10000 non-null float64
    NumOfProducts
                    10000 non-null int64
                     10000 non-null int64
10 HasCrCard
   IsActiveMember
                     10000 non-null int64
11
12 EstimatedSalary 10000 non-null float64
                     10000 non-null int64
13 Exited
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()

\	RowNumbe	er	Custome	rId	Surname	CreditScore	Geography	Gender	Age
0		1	15634	602	Hargrave	619	France	Female	42
1		2	15647	311	Hill	608	Spain	Female	41
2		3	15619	304	Onio	502	France	Female	42
3		4	15701	354	Boni	699	France	Female	39
4		5	15737	888	Mitchell	850	Spain	Female	43
	Tenure		Balance	Num	OfProducts	HasCrCard	IsActiveMe	mber \	
0	2		0.00		1	1		1	
1	1	8	3807.86		1	0		1	
2	8	15	9660.80		3	1		0	
3	1		0.00		2	0		0	
	_				_	-		-	

1

1

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

2 125510.82

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

data.sum()

RowNumber CustomerId	50005000 156909405694
Surname CreditScore	HargraveHillOnioBoniMitchellChuBartlettObinnaH 6505288
Geography	FranceSpainFranceFranceSpainSpainFranceGermany
Gender	FemaleFemaleFemaleFemaleMaleMaleFemaleMa
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
       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 CustomerId CreditScore Age Tenure Balance NumOfProducts HasCrCard IsActiveMember EstimatedSalary	5.000500e+03 1.569074e+07 6.520000e+02 3.700000e+01 5.000000e+00 9.719854e+04 1.000000e+00 1.000000e+00 1.000000e+00
EstimatedSalary Exited	1.001939e+05 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
                  71936.186123
CustomerId
                     96.653299
CreditScore
                      10.487806
Age
                       2.892174
Tenure
Balance
                  62397.405202
NumOfProducts
                      0.581654
HasCrCard
                      0.455840
IsActiveMember
                      0.499797
                   57510.492818
EstimatedSalary
                      0.402769
Exited
dtype: float64
data.var()
C:\Users\Del1\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
Aae
                  1.099941e+02
Tenure
                  8.364673e+00
Balance
                  3.893436e+09
NumOfProducts
                3.383218e-01
                  2.077905e-01
HasCrCard
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
       60
Name: NumOfProducts, dtype: int64
data.describe()
        RowNumber CustomerId CreditScore
                                                        Age
Tenure
```

count 10000.	10000.00000	1.000000e+0	10000.000000	10000.000000	
	5000.50000	1.569094e+0	7 650.528800	38.921800	
5.0128	00				
std	2886.89568	7.193619e+0	4 96.653299	10.487806	
2.8921	74				
min		1.556570e+0	7 350.000000	18.000000	
0.0000		1 560050 .0	7	20 00000	
25% 3.0000		1.562853e+0	7 584.000000	32.000000	
50%		1.569074e+0	7 652.000000	37.000000	
5.0000		1.303071010	7 032:00000	37.000000	
75%		1.575323e+0	7 718.000000	44.000000	
7.0000	00				
max	10000.00000	1.581569e+0	7 850.000000	92.000000	
10.000	000				
		e NumOfProd			\
count		0 10000.00			
mean	76485.88928				
	62397.40520		1654 0.4558		
min	0.00000		0.000		
25%	0.00000		0.000		
	97198.54000		0000 1.0000		
75%			0000 1.0000		
max	250898.09000	0 4.00	0000 1.0000	0 1.000000	
		E			
aaiin+	EstimatedSala 10000.000	_	xited		
count	10000.000		03700		
mean std	57510.492		03760		
min	11.580		00000		
	51002.110		00000		
50%	100193.915		00000		
75%	149388.247		00000		
max	199992.480		00000		
5.HAND	LE MISSING VAL	UE			

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
J -	\					
0	False	False	False	False	False	False
False	False	False	False	False	False	False
False						
2	False	False	False	False	False	False
False						- 1
3 False	False	False	False	False	False	False
4	False	False	False	False	False	False
False						
				• • •	• • •	• • •
••• 9995	False	False	False	False	False	False
False		raise	raise	raise	raise	raise
9996	False	False	False	False	False	False
False						
9997	False	False	False	False	False	False
False	False	False	False	False	False	False
False		raise	raise	raise	raise	raise
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
       . . .
                              . . .
               . . .
                                         . . .
                                                        . . .
. . .
            False
9995
      False
                             False
                                        False
                                                       False
            False
9996 False
                             False
                                        False
                                                       False
                                       False
9997 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
                  0
Surname
                  0
CreditScore
Geography
                  0
Gender
                  0
                  0
Age
Tenure
                  0
Balance
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

7.00	RowNumbe	r Custome:	rId	Surname	CreditScore	Geography	Gender
Age 0		1 15634	602	Hargrave	619	France	Female
42 1		2 156473	311	Hill	608	Spain	Female
41		3 156193	304	Onio	502	France	Female
42 3 39		4 157013	354	Boni	699	France	Female
4		5 157378	888	Mitchell	850	Spain	Female
43							
9995 39	999	6 156062	229	Obijiaku	771	France	Male
9996 35	999	7 155698	892	Johnstone	516	France	Male
9997 36	999	8 15584	532	Liu	709	France	Female
9998 42	999	9 156823	355	Sabbatini	772	Germany	Male
9999	1000	0 156283	319	Walker	792	France	Female
0 1 2 3 4 9995	1	Balance 0.00 83807.86 159660.80 0.00 125510.82 0.00	NumO	fProducts 1 1 3 2 1 2	HasCrCard	IsActiveMer	mber \ 1
9996 9997	10	57369.61		1 1	1 0		1 1
9998	3	0.00 75075.31 130142.79		2	1 1		0
0 1 2 3 4	11 11 9	dSalary E: 1348.88 2542.58 3931.57 3826.63 9084.10	xited 1 0 1 0				

9995 9996 9997 9998 9999	1016 420 928 381	270.64 599.77 085.58 388.52 .90.78	0 0 1 1 0			
	00 rows x 14					
df.i	snull().sum().sum()				
0						
df1=	data.fillna(value=5)				
7	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender
Age 0 42	1	15634602	Hargrave	619	France	Female
1	2	15647311	Hill	608	Spain	Female
41 2 42	3	15619304	Onio	502	France	Female
3	4	15701354	Boni	699	France	Female
4 4 3	5	15737888	Mitchell	850	Spain	Female
• • •			• • •			
9995 39	9996	15606229	Obijiaku	771	France	Male
9996 35	9997	15569892	Johnstone	516	France	Male
9997 36	9998	15584532	Liu	709	France	Female
9998 42	9999	15682355	Sabbatini	772	Germany	Male
9999 28	10000	15628319	Walker	792	France	Female
∠ ŏ	Tenure	Balance Num	nOfProducts	HasCrCard	IsActiveMem	uber \
0	2	0.00	1	1		1
1 2		33807.86 59660.80	1 3	0 1		1 0
3	1	0.00	2	0		0
4		25510.82	1	1		1
9995		0.00	2	1		0
9996 9997		0.00	1 1	1 0		1 1
9998		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

FILLING NULL VALUES WITH A PREVIOUS VALUE

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

7 00	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender
Age 0 42	1	15634602	Hargrave	619	France	Female
1 41	2	15647311	Hill	608	Spain	Female
2 42	3	15619304	Onio	502	France	Female
3	4	15701354	Boni	699	France	Female
4 43	5	15737888	Mitchell	850	Spain	Female
4						
9995 39	9996	15606229	Obijiaku	771	France	Male
9996 35	9997	15569892	Johnstone	516	France	Male
9997 36	9998	15584532	Liu	709	France	Female
9998	9999	15682355	Sabbatini	772	Germany	Male
42 9999 28	10000	15628319	Walker	792	France	Female
0 1	2	Balance Num 0.00 3807.86	OfProducts 1 1	HasCrCard 1	IsActiveMem	ber \ 1 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

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 42	1	15634602	Hargrave	619	France	Female
1	2	15647311	Hill	608	Spain	Female
41 2 42	3	15619304	Onio	502	France	Female
3	4	15701354	Boni	699	France	Female
4 43	5	15737888	Mitchell	850	Spain	Female
		• • •	• • •	• • •		
9995 39	9996	15606229	Obijiaku	771	France	Male
9996 35	9997	15569892	Johnstone	516	France	Male

9997 36	99	998 1558	4532	Liu	70)9 Fr	ance	Fem	ale
9998 42	99	999 15682	2355	Sabbatini	77	'2 Ger	many	M	ale
9999 28	100	15628	8319	Walker	79	92 Fr	ance	Fem	ale
0 1 2 3 4	Tenure 2 1 8 1 2	Balance 0.00 83807.86 159660.80 0.00 125510.82	Num	OfProducts 1 1 3 2 1	HasCrCard 1 0 1 0 1	IsActi	veMem	ber 1 1 0 0	\
9995 9996 9997 9998 9999	5 10 7 3 4	0.00 57369.61 0.00 75075.31 130142.79		2 1 1 2 1	1 1 0 1			0 1 1 0 0	
0 1 2 3 4 9995 9996 9997 9998 9999	1 1 1	edSalary .01348.88 .12542.58 .13931.57 93826.63 79084.10 96270.64 .01699.77 42085.58 92888.52 38190.78		d 1 0 1 0 0 0 . 0 1 1 1 0 0 0 0 1 1 1					

DROPPING NULL VALUES

df4=data.dropna()
df4

Age	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender
0 42	1	15634602	Hargrave	619	France	Female
1 41	2	15647311	Hill	608	Spain	Female
2	3	15619304	Onio	502	France	Female
3	4	15701354	Boni	699	France	Female

4	5	15737	888	Mitchell	850	Spain	Female
43							
• • •	• • •		• • •	• • •	• • •	• • •	• • •
9995	9996	15606	229	Obijiaku	771	France	Male
39	0007	15560	0.00	- 1	F1.6	_	
9996 35	9997	15569	892	Johnstone	516	France	Male
9997	9998	15584	532	Liu	709	France	Female
36 9998	9999	15682	355	Sabbatini	772	Germany	Male
42	2222	13002	333	Sabbacini	112	ocimany	Marc
9999 28	10000	15628	319	Walker	792	France	Female
Te	nure I	Balance	NumC	fProducts	HasCrCard	IsActiveMem	nber \
0	2	0.00		1	1		1
1		3807.86		1	0		1
2 3	8 159 1	9660.80		3 2	1		0
4		5510.82		1	1		0 1
•••				• • •	• • •		
9995	5	0.00		2	1		0
9996		7369.61		1	1		1
9997	7			1	0		1
9998	3 7	5075.31		2	1		0
9999	4 130	0142.79		1	1		0
Es	timatedSa	alary E	xited	[
0		48.88	1				
1	1125	42.58	0				
2	11393	31.57	1				
3	9382	26.63	0				
4	7908	84.10	0				
9995	962	70.64					
9996		99.77	0				
9997		85.58	1				
9998		88.52	1				
9999		90.78	0				
[10000 r	ows x 14	columns]				
df5=data df5	.dropna(ł	now='any	')				
Ro	wNumber	Custome	rId	Surname	CreditScore	Geography	Gender
Age \							

1	2	15647311	Hill	608	Spain	Female
41	3	15619304	Onio	502	2 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	5 France	Male
35 9997	9998	15584532	Liu	709		Female
36 9998	9999	15682355		772		Male
42 9999	10000	15628319	Walker	792	_	
28	_,,,,					
0 1 2 3 4 9995 9996 9997 9998 9999	2 1 83 8 159 1 2 125 5 10 57 7 3 75	3alance Num 0.00 3807.86 9660.80 0.00 5510.82 0.00 7369.61 0.00 5075.31	nOfProducts 1 1 3 2 1 2 1 1 2 1	HasCrCard 1 0 1 0 1 1 0 1 1 1	IsActiveMen	nber \ 1
0 1 2 3 4 9995 9996 9997 9998 9999	7908 9627 10169 4208 9288	48.88 42.58 31.57 26.63 34.10 	ed 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0			

replace()

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

7	RowNumb	er Custome	erId	Surname	CreditScore	Geography	Gender
Age 0 42	\	1 15634	1602	Hargrave	619	France	Female
1 41		2 15647	7311	Hill	608	Spain	Female
2 42		3 15619	9304	Onio	502	France	Female
3		4 15701	L354	Boni	699	France	Female
4 4 3		5 15737	7888	Mitchell	850	Spain	Female
	•	• •					
9995 39	99	96 15606	5229	Obijiaku	771	France	Male
9996 35	99	97 15569	9892	Johnstone	516	France	Male
9997 36	99	98 15584	1532	Liu	709	France	Female
9998 42	99	99 15682	2355	Sabbatini	772	Germany	Male
9999 28	100	00 15628	3319	Walker	792	France	Female
	Tenure	Balance	Num	OfProducts	HasCrCard	IsActiveMen	mber \
0	2	0.00		1	1		1
1	1	83807.86		1	0		1
2	8 1	159660.80		3 2	1		0 0
4	2	125510.82		1	1		1
 9995	• • • 5	0.00		2	• • •		•••
9996	10	57369.61		1	1 1		0 1
9997	7	0.00		1	0		1
9998	3	75075.31		2	1		0
9999	4	130142.79		1	1		0
		_	Exited	d			
0		01348.88		1			
1 2		12542.58 13931.57) 1			
3		93826.63)			
4		79084.10)			
 9995		 96270.64		•			

```
      9996
      101699.77
      0

      9997
      42085.58
      1

      9998
      92888.52
      1

      9999
      38190.78
      0
```

interpolate()

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

data							
Age	RowNumb	er Custom	erId	Surname	CreditScore	Geography	Gender
0 42	,	1 1563	4602	Hargrave	619	France	Female
1 41		2 1564	7311	Hill	608	Spain	Female
2 42		3 1561	9304	Onio	502	France	Female
3		4 1570	1354	Boni	699	France	Female
39 4		5 1573	7888	Mitchell	850	Spain	Female
43							
9995	99	96 1560	6229	Obijiaku	771	France	Male
39 9996	99	97 1556	9892	Johnstone	516	France	Male
35 9997	99	98 1558	4532	Liu	709	France	Female
36 9998	99	99 1568	2355	Sabbatini	772	Germany	Male
42 9999 28	100	00 1562	8319	Walker	792	France	Female
0 1 2 3 4	Tenure 2 1 8 1 2	Balance 0.00 83807.86 159660.80 0.00 125510.82	Num	nOfProducts 1 1 3 2 1	HasCrCard 1 0 1 0 1	IsActiveMem	ber \ 1
9995 9996 9997 9998 9999	5 10 7 3 4	0.00 57369.61 0.00 75075.31 130142.79		2 1 1 2 1	1 1 0 1		0 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
. . .
                 . . .
                        . . .
           96270.64
                         0
9995
          101699.77
                          0
9996
9997
           42085.58
                          1
                         1
9998
           92888.52
9999
           38190.78
```

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,
```

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,

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,

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,

444,

444,

444,

444, 444,

444,

445,

445,

445,

445,

445,

445,

445,

445,

445,

445,

446,

446, 446,

446,

446,

447,

447,

447,

447,

448,

448,

449, 449,

449,

449,

449,

449,

450,

450,

450,

450,

451,

451,

451,

451,

451, 452,

452,

452,

452,

452,

452,

452,

452,

452,

452, 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,

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,

470,

470,

470,

470,

470,

470,

470,

470,

470,

471,

471,

471,

471,

471,

471,

471,

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,

475,

475, 475,

475,

475,

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,

479,

479,

479,

479,

479,

479,

479,

479,

479, 479,

479,

479,

479, 479,

479,

479,

479,

480,

480,

480,

480,

480,

480,

480,

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,

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,

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,

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,

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,

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,

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,

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,

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,

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,

512,

512,

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,

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,

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,

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,
 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,
 ...]
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
```

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
Tenur 0 2	e \ 1	15634602	Hargrave	619	France	42
1	2	15647311	Hill	608	Spain	41
2	3	15619304	Onio	502	France	42
8	4	15701354	Boni	699	France	39
1 4 2	5	15737888	Mitchell	850	Spain	43
				• • •		
9995 5	9996	15606229	Obijiaku	771	France	39
9996	9997	15569892	Johnstone	516	France	35

```
10
9997 9998 15584532 Liu 709 France 36
        9999 15682355 Sabbatini
9998
                                      772 Germany 42
9999 10000 15628319 Walker
                                       792 France
                                                    28
4
     Balance NumOfProducts HasCrCard IsActiveMember
EstimatedSalary \
0.00
                      1 1
                                             1
101348.88
1 83807.86
                       1
112542.58
   159660.80
113931.57
       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
       0.00
9997
                       1
                                             1
42085.58
9998 75075.31
92888.52
9999 130142.79
                       1
                               1
                                             0
38190.78
    Exited
0
         1
1
         0
2
         1
3
         0
4
       0
. . .
       . . .
9995
       0
       0
9996
9997
         1
9998
        1
       0
9999
[10000 rows x 13 columns]
NEW DataM1["Male"] = data1["Male"].to list()
NEW DataM1["Female"] = data1["Female"].to list()
```

NEW_DataM1

7 00	RowNumb	er Custo	merId	Surna	me	CreditSco	re	Geography	Gender
Age 0 42	\	1 15634602		Hargra	ve	6	19	France	Female
1 41		2 15647311		Hill		6	08	Spain	Female
2 42		3 156	19304	Onio		5	02	France	Female
3		4 157	01354	Boni		6	99	France	Female
4 43		5 157	37888	Mitchell		8	50	Spain	Female
		• •	• • •		• •		• •	• • •	• • •
9995 39	99	96 156	06229	Obijia	ku	7	71	France	Male
9996 35	99	97 155	69892	Johnsto	ne	5	16	France	Male
9997 36	99	98 155	84532	L	iu	7	09	France	Female
9998 42	99	99 156	82355	Sabbati	ni	7	72	Germany	Male
9999 28	100	00 156	28319	Walk	er	7	92	France	Female
	Tenure	Balanc	e Num(OfProduc	ts	HasCrCard	I	IsActiveMen	nber \
0	2	0.0			1	1			1
1	1	83807.8			1	0			1
2 3	8 1	159660.8			3 2	1 0			0 0
4	2	125510.8			1	1			1
	• • •		•	•	• •	• • •			• • •
9995	5	0.0			2	1			0
9996 9997	10 7	57369.6 0.0			1 1	1 0			1 1
9998	3	75075.3			2	1			0
9999	4	130142.7			1	1			0
	Estimat	edSalary	Exite	d Male	Fer	male			
0		01348.88		1 0		1			
1	1	12542.58		0 0		1			
2	1	13931.57		1 0		1			
3		93826.63		0 0		1			
4		79084.10		0 0		1			
9995		96270.64	• •	0 1		0			
9996			'	·		O			
2220	1	01699.77		0 1		0			

```
9998
             92888.52
                             1
                                    1
9999
             38190.78
                             0
                                    \cap
                                            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
                15647311
                               Hill
                                              608
                                                       Spain
                                                              Female
                                                                        41
   Tenure
            Balance NumOfProducts HasCrCard IsActiveMember \
0
        2
               0.00
                                  1
                                              1
                                                               1
                                              0
1
        1 83807.86
                                  1
                                                               1
   EstimatedSalary Exited Male Female
0
         101348.88
                          1
                                0
1
         112542.58
                          0
                                0
METHOD II
from sklearn.preprocessing import LabelEncoder
data=pd.read csv("Churn Modelling.csv")
13=LabelEncoder()
label=13.fit transform(data["Gender"])
13.classes
array(['Female', 'Male'], dtype=object)
Data=NEW DataM1.drop("Gender", axis='columns')
Data
      RowNumber CustomerId
                                Surname CreditScore Geography
Tenure \
0
              1
                    15634602
                               Hargrave
                                                  619
                                                                   42
                                                          France
2
1
              2
                    15647311
                                   Hill
                                                  608
                                                           Spain
                                                                   41
1
2
              3
                    15619304
                                    Onio
                                                  502
                                                          France
                                                                   42
8
3
              4
                    15701354
                                                  699
                                                                   39
                                    Boni
                                                          France
1
4
              5
                    15737888
                               Mitchell
                                                  850
                                                           Spain
                                                                   43
2
                         . . .
                                     . . .
                                                  . . .
. . .
            . . .
                                                             . . .
                                                                  . . .
. . .
9995
           9996
                    15606229
                             Obijiaku
                                                  771
                                                          France
                                                                   39
```

5									
	9997	15569892	Jo	hnstone	516	France	35		
10 9997 7	9998	15584532		Liu	709 France				
9998	9999	15682355	Sa	bbatini	772	Germany	42		
9999 4	10000	15628319		Walker	792	France	28		
		NumOfProdu	cts	HasCrCard	IsActiveMe	ember			
Estimate		\	_	_		_			
0 101348.8			1	1		1			
1 8 112542.5	3807.86		1	0		1			
2 15 113931.5	9660.80		3	1		0			
3 93826.63	0.00		2	0		0			
4 12 79084.10	5510.82		1	1		1			
9995	0 00		2	1		0			
96270.64			_	Τ		U			
9996 5 101699.7			1	1		1			
9997 42085.58	0.00		1	0		1			
9998 7 92888.52	5075.31		2	1		0			
9999 13 38190.78	0142.79		1	1		0			
Ex	ited Ma	ale Female 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]

	CustomerId	Surname	CreditScore	Geography	Age
Tenure \ 0 1 2	15634602	Hargrave	619	France	42
1 2 1	15647311	Hill	608	Spain	41
2 3	15619304	Onio	502	France	42
3 4 1	15701354	Boni	699	France	39
4 5	15737888	Mitchell	850	Spain	43
9995 9996 5	15606229	Obijiaku	771	France	39
9996 9997 10	15569892	Johnstone	516	France	35
9997 9998 7	15584532	Liu	709	France	36
9998 9999 3	15682355	Sabbatini	772	Germany	42
9999 10000 4	15628319	Walker	792	France	28
	N 05D 1				
EstimatedSalary 0 0.00		ts Hascrca 1	rd IsActivel	member 1	
101348.88					
1 83807.86 112542.58		1	0	1	
2 159660.80 113931.57		3	1	0	
3 0.00 93826.63		2	0	0	
4 125510.82 79084.10		1	1	1	
				• • •	
9995 0.00 96270.64		2	1	0	
9996 57369.61 101699.77		1	1	1	
9997 0.00		1	0	1	
42085.58 9998 75075.31 92888.52		2	1	0	

```
9999 130142.79
                        1 1
                                                  0
38190.78
     Exited Male Female Gender
0
               0
                             0
         1
                      1
1
         0
               0
                      1
                             0
2
         1
               0
                      1
                             0
3
         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
```

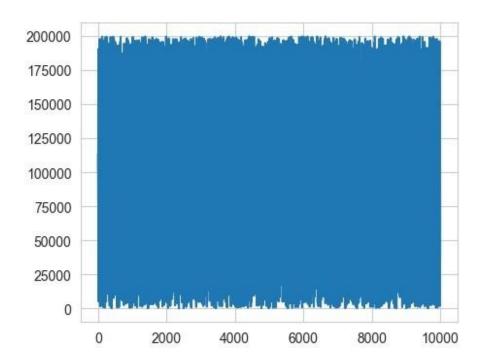
	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance
0	Hargrave	619	France	Female	42	2	0.00
1	Hill	608	Spain	Female	41	1	83807.86
2	Onio	502	France	Female	42	8	159660.80
3	Boni	699	France	Female	39	1	0.00
4	Mitchell	850	Spain	Female	43	2	125510.82
9995	Obijiaku	771	France	Male	39	5	0.00
9996	Johnstone	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
          1
             15634602 Hargrave ...
                                                     1
101348.88
             1
                         Hill ...
          2
              15647311
112542.58
              0
              15619304
          3
                            Onio ...
                                                     0
113931.57
             1
          4
              15701354 Boni ...
                                                     0
93826.63
             0
             15737888 Mitchell ...
79084.10 0
[5 rows x 14 columns]
tenure=data.EstimatedSalary
plt.plot(tenure)
[<matplotlib.lines.Line2D at 0x14ec8f2b400>]
```

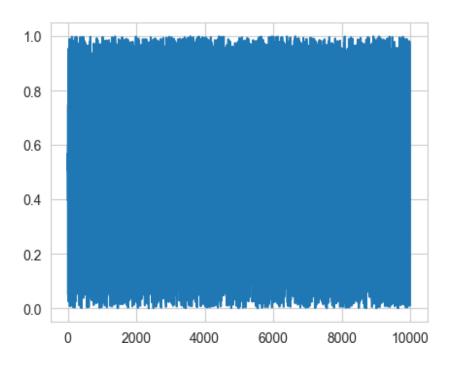


data[['Tenure']].describe()

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

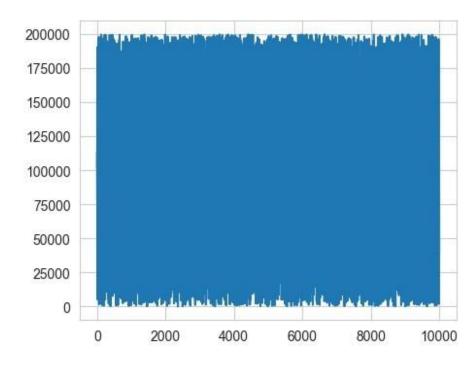
```
tenure_matrix=tenure.values.reshape(-1,1)
scaled=preprocessing.MinMaxScaler()
scaled_tenure=scaled.fit_transform(tenure_matrix)
plt.plot(scaled_tenure)
```

[<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
        5000.50000 1.569094e+07
                                       100090.239881
mean
                                 . . .
                                                           0.203700
        2886.89568 7.193619e+04
                                         57510.492818
                                                           0.402769
std
                                 . . .
           1.00000 1.556570e+07
                                             11.580000
                                                           0.000000
min
                                  . . .
                                        51002.110000
25%
        2500.75000 1.562853e+07
                                                           0.000000
                                  . . .
50%
       5000.50000 1.569074e+07
                                        100193.915000
                                                           0.000000
                                  . . .
75%
       7500.25000 1.575323e+07
                                 . . .
                                        149388.247500
                                                           0.000000
      10000.00000 1.581569e+07 ... 199992.480000
max
                                                           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,)
```