Assignment -2

Python Programming

Assignment Date	20 September 2022
Student Name	M Raagavi
Student Roll Number	820419106044
Maximum Marks	2 Marks

Questions

LOAD THE DATASET

PERFORM THE VISUALIZATIONS

- **(I) UNIVARIATE ANALYSIS**
- **(II) BI-VARIATE ANALYSIS**
- **(III) MULTI VARIATE ANALYSIS**

PERFORM DESCRIPTIVE STATISTICS ON THE DATASET

HANDLE THE MISSING VALUES

FIND THE OUTLIERS AND REPLACE THE OUTLIERS

CHECK CATEGORICAL COLUMNS AND PERFORM ENCODING

SPILT THE DATA INTO DEPENDENT AND INDEPENDENT VARIABLES

SCALE THE INDEPENDENT VARIABLES

SPLIT THE DATA INTO TRAINING AND TESTING

```
import pandas as pd
import numpy as np

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

#descriptive analysis
data.describe()
```

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

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

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

```
#median of the data
data.median()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2:
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.


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
```

```
#mode of the data
data.mode()
```

```
      RowNumber  CustomerId Surname  CreditScore Geography Gender
Age  \
0             1    15565701   Smith        850.0    France   Male
37.0
1             2    15565706     NaN          NaN       NaN    NaN
NaN
2             3    15565714     NaN          NaN       NaN    NaN
NaN
3             4    15565779     NaN          NaN       NaN    NaN
NaN
4             5    15565796     NaN          NaN       NaN    NaN
NaN
...         ...         ...     ...          ...       ...    ...   ..
.
9995       9996    15815628     NaN          NaN       NaN    NaN
NaN
9996       9997    15815645     NaN          NaN       NaN    NaN
NaN
9997       9998    15815656     NaN          NaN       NaN    NaN
NaN
9998       9999    15815660     NaN          NaN       NaN    NaN
NaN
9999      10000    15815690     NaN          NaN       NaN    NaN
NaN

      Tenure  Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0        2.0      0.0            1.0        1.0             1.0
1        NaN      NaN            NaN        NaN             NaN
```

```
2         NaN      NaN            NaN        NaN             NaN
3         NaN      NaN            NaN        NaN             NaN
4         NaN      NaN            NaN        NaN             NaN
...       ...      ...            ...        ...             ...
9995      NaN      NaN            NaN        NaN             NaN
9996      NaN      NaN            NaN        NaN             NaN
9997      NaN      NaN            NaN        NaN             NaN
9998      NaN      NaN            NaN        NaN             NaN
9999      NaN      NaN            NaN        NaN             NaN

      EstimatedSalary  Exited
0            24924.92     0.0
1                 NaN     NaN
2                 NaN     NaN
3                 NaN     NaN
4                 NaN     NaN
...               ...     ...
9995              NaN     NaN
9996              NaN     NaN
9997              NaN     NaN
9998              NaN     NaN
9999              NaN     NaN

[10000 rows x 14 columns]
```

```
#mean of the data-descriptive analysis
data.mean()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2:
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.


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
```

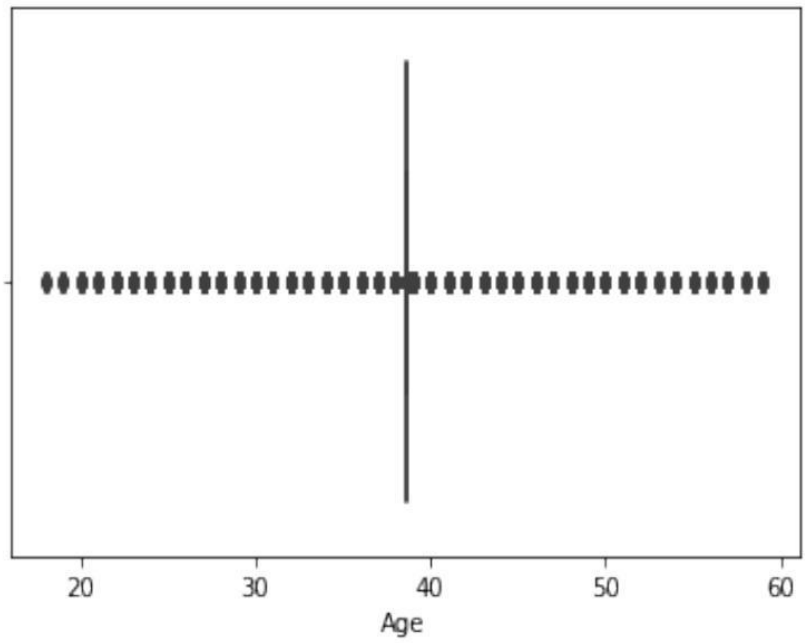
```
#missing values
data.isnull().sum()
```

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

```
#dealing with outliers
import seaborn as sns
sns.boxplot(data['Age'])
```

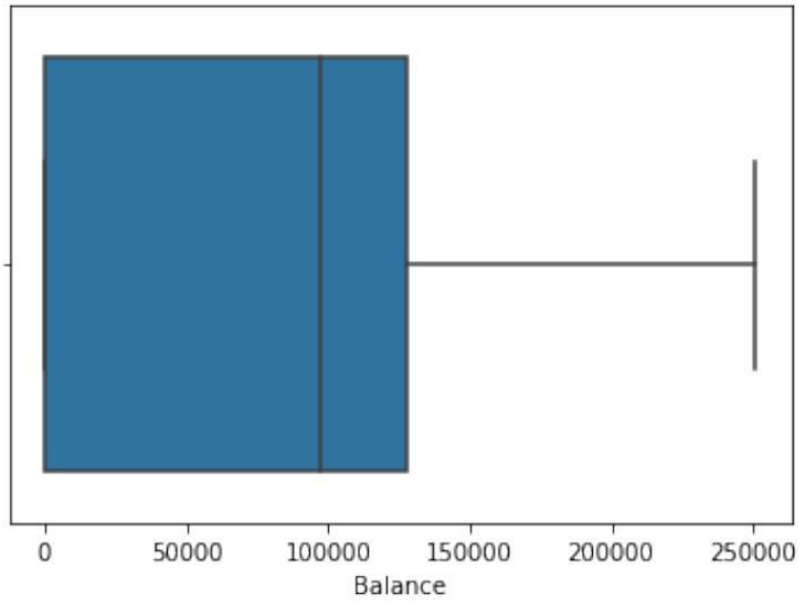
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be `data`, and
passing other arguments without an explicit keyword will result in an
error or misinterpretation.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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb6e1dc5810>
```




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

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

<matplotlib.axes._subplots.AxesSubplot at 0x7fb6e1e66690>
```




```
#finding quantile
qnt=data.quantile(q=[0.25,0,0.75])

qnt
```

```
      RowNumber   CustomerId  CreditScore   Age  Tenure    Balance  \
0.25    2500.75  15628528.25        584.0  32.0     3.0       0.00
0.00       1.00  15565701.00        350.0  18.0     0.0       0.00
0.75    7500.25  15753233.75        718.0  44.0     7.0  127644.24

      NumOfProducts  HasCrCard  IsActiveMember  EstimatedSalary
Exited
0.25            1.0        0.0             0.0       51002.1100
0.0
0.00            1.0        0.0             0.0          11.5800
0.0
0.75            2.0        1.0             1.0      149388.2475
0.0
```

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

```
IQR
```

```
RowNumber            4999.5000
CustomerId         124705.5000
CreditScore           134.0000
Age                    12.0000
Tenure                  4.0000
Balance            127644.2400
NumOfProducts           1.0000
HasCrCard               1.0000
IsActiveMember          1.0000
EstimatedSalary     98386.1375
Exited                  0.0000
dtype: float64
```

```
upper_extreme=qnt.loc[0.75]+1.25*IQR
lower_extreme=qnt.loc[0.25]-1.5*IQR
```

```
upper_extreme
```

```
RowNumber          1.374962e+04
CustomerId         1.590912e+07
CreditScore        8.855000e+02
Age                5.900000e+01
Tenure             1.200000e+01
Balance            2.871995e+05
NumOfProducts      3.250000e+00
HasCrCard          2.250000e+00
IsActiveMember     2.250000e+00
EstimatedSalary    2.723709e+05
Exited             0.000000e+00
dtype: float64
```

```
lower_extreme
```

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

```
data[data['Age']>5.900000e+01]
```

```
      RowNumber  CustomerId      Surname  CreditScore Geography
Gender  Age  \
```

```
42           43    15687946      Osborne          556    France
Female   61
44           45    15684171      Bianchi          660     Spain
Female   61
58           59    15623944        T'ien          511     Spain
Female   66
85           86    15805254      Ndukaku          652     Spain
Female   75
104         105    15804919     Dunbabin          670     Spain
Female   65
...          ...         ...          ...          ...       ...     ..
.  ...
9832        9833    15814690  Chukwujekwu          595   Germany
Female   64
9879        9880    15669414       Pisano          486   Germany
Male   62
9894        9895    15704795        Vagin          521    France
Female   77
9897        9898    15810563           Ho          678     Spain
Female   61
9936        9937    15653037        Parks          609    France
Male   77

      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
42         2  117419.35              1          1               1
44         5  155931.11              1          1               1
58         4       0.00              1          1               0
85        10       0.00              2          1               1
104        1       0.00              1          1               1
...      ...        ...            ...        ...             ...
9832       2  105736.32              1          1               1
9879       9  118356.89              2          1               0
9894       6       0.00              2          1               1
9897       8       0.00              2          1               1
9936       1       0.00              1          0               1

      EstimatedSalary  Exited
42           94153.83       0
44          158338.39       0
58            1643.11       1
85          114675.75       0
104         177655.68       1
...               ...     ...
9832         89935.73       1
9879        168034.83       1
9894         49054.10       0
9897        159938.82       0
9936         18708.76       0

[526 rows x 14 columns]
```

```
data[data['Balance']>2.871995e+05]
```

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

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

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

```
data[data['Balance']<-1.914664e+05]
```

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

```
#Replacing outliers with mean
data['Age']=np.where(data['Age']>5.900000e+01,data['Age'].mean(),data[
'Age'])

#After replacing mean, no outliers are present for Age column
data[data['Age']>5.900000e+01]
```

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

```
#Encoding - Dummies(ONE HOT ENCODING)
pd.get_dummies(data,columns=['Geography'])
```

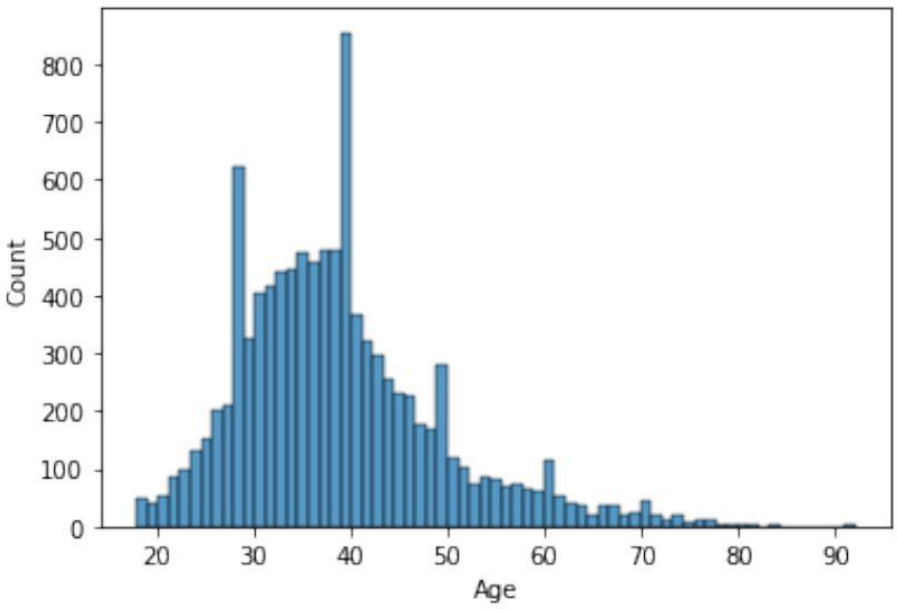
```
      RowNumber  CustomerId    Surname  CreditScore  Gender        Age
\
0             1    15634602   Hargrave          619  Female  42.000000

1             2    15647311       Hill          608  Female  38.633271

2             3    15619304       Onio          502  Female  38.633271

3             4    15701354       Boni          699  Female  39.000000

4             5    15737888   Mitchell          850  Female  38.633271
```

```
...         ...         ...        ...          ...     ...        ...

9995       9996    15606229   Obijiaku          771    Male  39.000000

9996       9997    15569892  Johnstone          516    Male  38.633271

9997       9998    15584532        Liu          709  Female  36.000000

9998       9999    15682355  Sabbatini          772    Male  38.633271

9999      10000    15628319     Walker          792  Female  38.633271


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

      Geography_Spain
0                   0
1                   1
2                   0
3                   0
4                   1
...               ...
9995                0
9996                0
9997                0
```

```
9998               0
9999               0

[10000 rows x 16 columns]
```

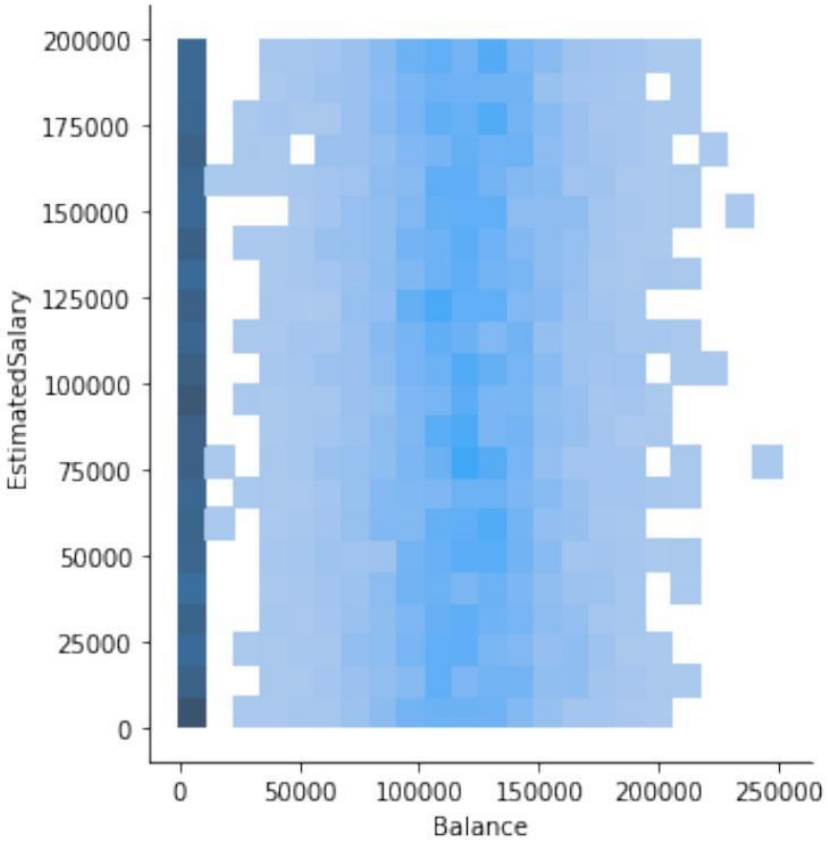
```
#Scale the independent variable
Geography = pd.get_dummies(data.Geography)
Gender = pd.get_dummies(data.Gender)

#Split the data into training and testing
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.2, random_state = 0)

import seaborn as sns

#Univariate visualization-histplot
sns.histplot(data,x='Age')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6bb107a1d0>
```



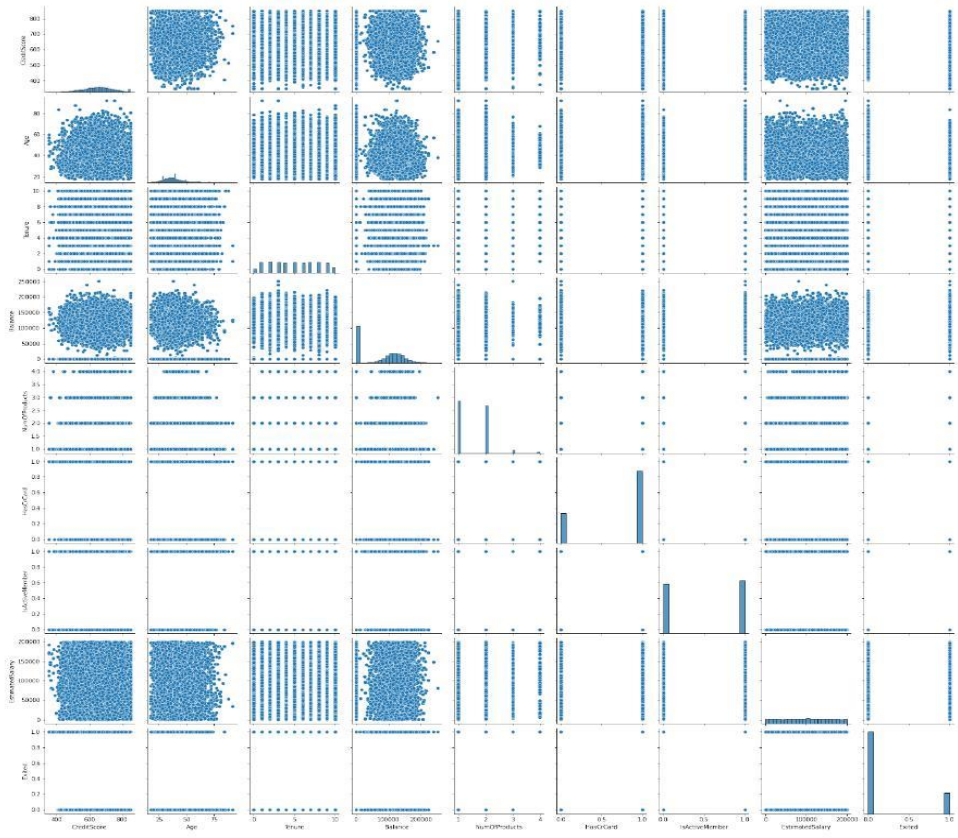

```
#Bivariate visualization-displot
sns.displot(data,x='Balance',y='EstimatedSalary')
```

```
<seaborn.axisgrid.FacetGrid at 0x7f6bb1097990>
```




```
#Multivariate visualization
sns.pairplot(data)
```

```
<seaborn.axisgrid.PairGrid at 0x7f6bb119db90>
```


CreditScore
Age
Tenure
Balance
NumOfProducts
HasCrCard
IsActiveMember
EstimatedSalary
Exited