Assignment -1

Python Programming

Assignment Date	27 September 2022
Student Name	Akshaya.D
Student Roll Number	820419106006
Maximum Marks	2 Marks

Tasks:

1.Download the dataset

2.Load the data set

3.Perform below visualization

- Univariate Analysis
- Bivariate Analysis
- Multi-variate Analysis

4.Perform descriptive statistics on the dataset

5.Handle the missing values

6.Find the outliers and replace the outliers

7.Check for Categorical columns and perform encoding

8.Split the data into dependent and independent variables

9.Scale the independent variables

10.Split the data into training and testing

2.

```
import pandas as pd
import numpy as np

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

#descriptive analysis
data.describe()
```

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

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

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

```
data.mean()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.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.
  """Entry point for launching an IPython kernel.
```

```
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.median()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.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.
  """Entry point for launching an IPython kernel.
```

```
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.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]


data.skew()
```

```
RowNumber          0.000000
CustomerId         0.001149
HasCrCard         -0.901812
IsActiveMember    -0.060437
dtype: float64
```

```
data.kurt()
```

```
RowNumber         -1.200000
CustomerId        -1.196113
HasCrCard         -1.186973
IsActiveMember    -1.996747
dtype: float64
```

```
data.var()
```

```
RowNumber          8.334167e+06
CustomerId         5.174815e+09
HasCrCard          2.077905e-01
IsActiveMember     2.497970e-01
dtype: float64
```

```
data.std()
```

```
RowNumber           2886.895680
CustomerId         71936.186123
HasCrCard              0.455840
IsActiveMember         0.499797
dtype: float64
```

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

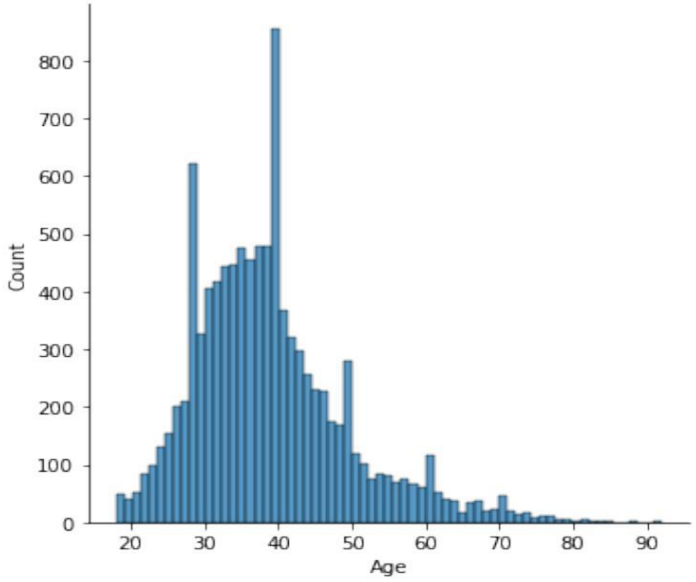
```
RowNumber          0
CustomerId         0
Surname            0
CreditScore        0
Geography          0
Gender             0
Age                0
Tenure             0
Balance            0
NumOfProducts      0
HasCrCard          0
IsActiveMember     0
EstimatedSalary    0
Exited             0
dtype: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

3.univariate analysis

```
import pandas as pd
import numpy as np
```

```
#univariate analysis
sns.displot(data,x="Age")
```

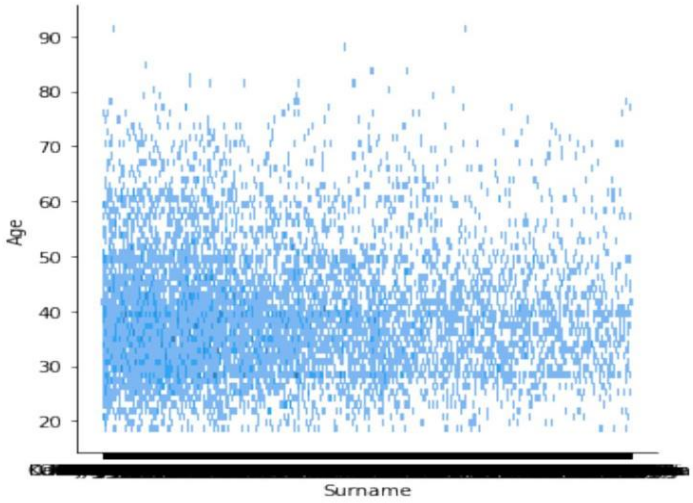
```
<seaborn.axisgrid.FacetGrid at 0x7fd7014b3910>
```




```
#bivariate analysis
sns.displot(data,x="Surname" , y="Age")
```

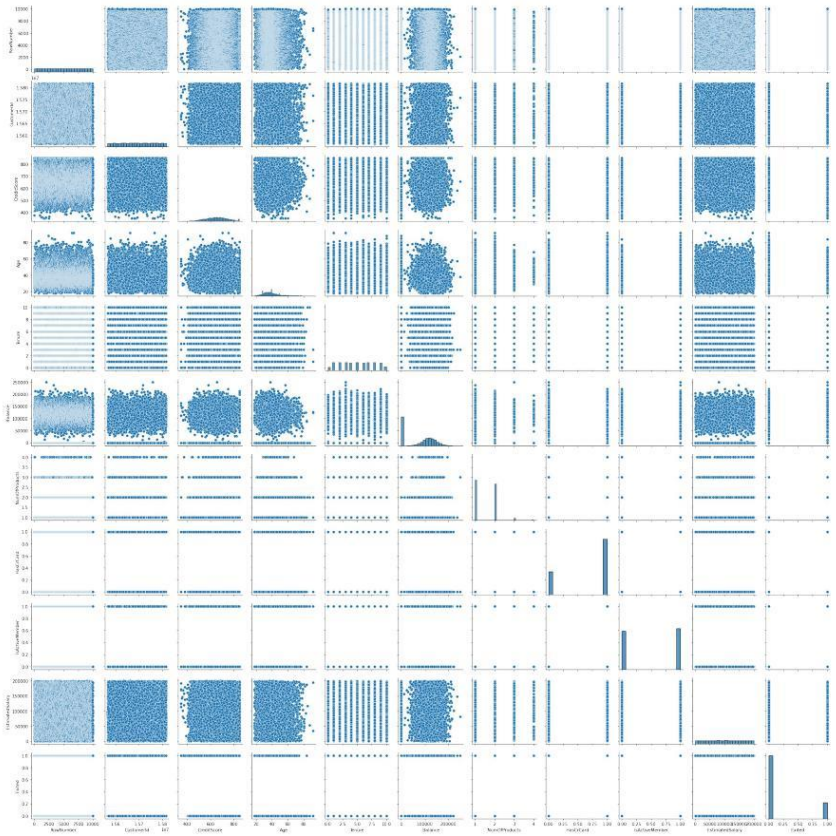
```
<seaborn.axisgrid.FacetGrid at 0x7fd70cd6fed0>
```

bivariate analysis




```
#multivariate analysis
sns.pairplot(data)
```

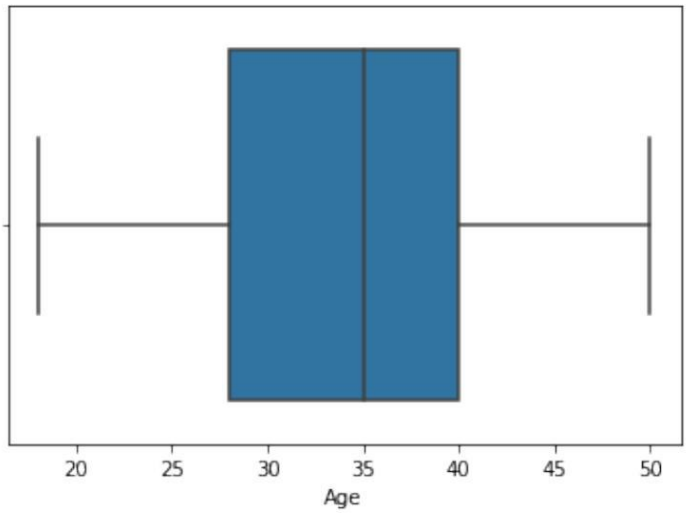
```
<seaborn.axisgrid.PairGrid at 0x7fd708557050>
```



```
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 0x7fd7012ee9d0>
```

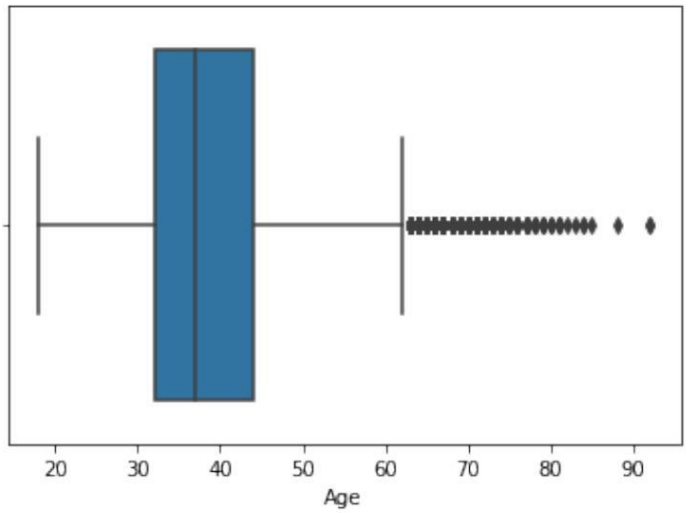



```
#categorical column and encoding
data.tail()
```

```
      RowNumber  CustomerId    Surname  CreditScore Geography  Gender
Age  \
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  \
9995       5       0.00              2          1               0
9996      10   57369.61              1          1               1
9997       7       0.00              1          0               1
9998       3   75075.31              2          1               0
9999       4  130142.79              1          1               0

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




```
import numpy as np
data['Age']=np.where(data['Age']>50,20,data['Age'])
```

```
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 0x7fd70127e450>
```

```
data['Gender'].replace({'Female':1,'Male':2},inplace=True)
data.tail()
```

```
      RowNumber  CustomerId    Surname  CreditScore Geography  Gender
Age  \
9995       9996    15606229   Obijiaku          771    France       2
39
9996       9997    15569892  Johnstone          516    France       2
35
9997       9998    15584532        Liu          709    France       1
36
9998       9999    15682355  Sabbatini          772   Germany       2
42
9999      10000    15628319     Walker          792    France       1
28

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

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

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

```
      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         0         0         0         1         0         0
0
9998         0         0         0         0         0         0
0
9999         1         0         0         0         0         0
0

[10000 rows x 24 columns]
```

```
#splitting of data
x=data_main['Balance']
x.head()
```

```
0         0.00
1     83807.86
2    159660.80
3         0.00
4    125510.82
Name: Balance, dtype: float64
```

```
y=data_main.drop(columns=['Balance'],axis=1)
y.head()
```

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

1          2    15647311      Hill          608     Spain       1   41

2          3    15619304      Onio          502    France       1   42

3          4    15701354      Boni          699    France       1   39

4          5    15737888  Mitchell          850     Spain       1   43


   NumOfProducts  HasCrCard  IsActiveMember  ...  Tenure_1
Tenure_2  \
0              1          1               1  ...         0         1

1              1          0               1  ...         1         0

2              3          1               0  ...         0         0

3              2          0               0  ...         1         0

4              1          1               1  ...         0         1


   Tenure_3  Tenure_4  Tenure_5  Tenure_6  Tenure_7  Tenure_8
```

```
Tenure_9  \
0         0         0         0         0         0         0
0
1         0         0         0         0         0         0
0
2         0         0         0         0         0         1
0
3         0         0         0         0         0         0
0
4         0         0         0         0         0         0
0

   Tenure_10
0          0
1          0
2          0
3          0
4          0

[5 rows x 23 columns]
```

```
#scale the independent variable
z=data_main.drop(columns=['Surname',],axis=1)
z.head
```

```
<bound method NDFrame.head of       RowNumber  CustomerId  CreditScore
Geography  Gender  Age    Balance  \
0             1    15634602          619    France  Female   42
0.00
1             2    15647311          608     Spain  Female   41
83807.86
2             3    15619304          502    France  Female   42
159660.80
3             4    15701354          699    France  Female   39
0.00
4             5    15737888          850     Spain  Female   43
125510.82
...         ...         ...          ...       ...     ...  ...
...
9995       9996    15606229          771    France    Male   39
0.00
9996       9997    15569892          516    France    Male   35
57369.61
9997       9998    15584532          709    France  Female   36
0.00
9998       9999    15682355          772   Germany    Male   42
75075.31
9999      10000    15628319          792    France  Female   28
130142.79
```

```
      NumOfProducts  HasCrCard  IsActiveMember  ...  Tenure_1
Tenure_2  \
0                 1          1               1  ...         0
1
1                 1          0               1  ...         1
0
2                 3          1               0  ...         0
0
3                 2          0               0  ...         1
0
4                 1          1               1  ...         0
1
...             ...        ...             ...  ...       ...       ..
.
9995              2          1               0  ...         0
0
9996              1          1               1  ...         0
0
9997              1          0               1  ...         0
0
9998              2          1               0  ...         0
0
9999              1          1               0  ...         0
0

      Tenure_3  Tenure_4  Tenure_5  Tenure_6  Tenure_7  Tenure_8
Tenure_9  \
0            0         0         0         0         0         0
0
1            0         0         0         0         0         0
0
2            0         0         0         0         0         1
0
3            0         0         0         0         0         0
0
4            0         0         0         0         0         0
0
...        ...       ...       ...       ...       ...       ...
...
9995         0         0         1         0         0         0
0
9996         0         0         0         0         0         0
0
9997         0         0         0         0         1         0
0
9998         1         0         0         0         0         0
0
9999         0         1         0         0         0         0
0
```

```
      Tenure_10
0             0
1             0
2             0
3             0
4             0
...         ...
9995          0
9996          1
9997          0
9998          0
9999          0

[10000 rows x 23 columns]>
```

```
#split 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,rando
m_state=0)

x_train.shape
```

```
(8000, 23)
```

```
x_test.shape
```

```
(2000, 23)
```

```
y_train.shape
```

```
(8000, 23)
```

```
y_test.shape
```

```
(2000, 23)
```