**Assignment -2**

Python Programming

| | |
|---|---|
| Assignment Date | 27 September 2022 |
| Student Name | Hemapriya R |
| Student Roll Number | 820419106019 |
| Maximum Marks | 2 Marks |

**Question**

**Perform the visualizations**

- **(i) Uni variate analysis**
- **(ii) Bi- variate analysis**
- **(iii) Multivariate analysis**

**Handle the missing values**

**Find the outliers and replace the outliers**

**Check for categorical columns and perform encoding**

**Spilt the data into dependent and independent variables**

**Scale the independent variables**

**Spilt the data into training and testing**

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import pandas as pd
import numpy as np

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

#descriptive analysis
data.describe()
```

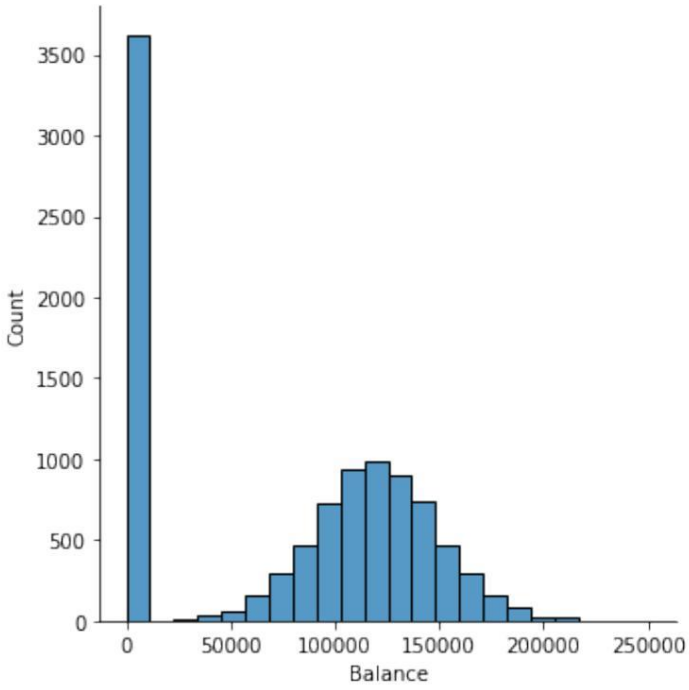
```
         RowNumber    CustomerId   CreditScore           Age
Tenure  \
count  10000.00000  1.000000e+04  10000.000000  10000.000000
10000.000000
mean    5000.50000  1.569094e+07    650.528800     38.921800
5.012800
std     2886.89568  7.193619e+04     96.653299     10.487806
2.892174
min        1.00000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

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

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

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

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

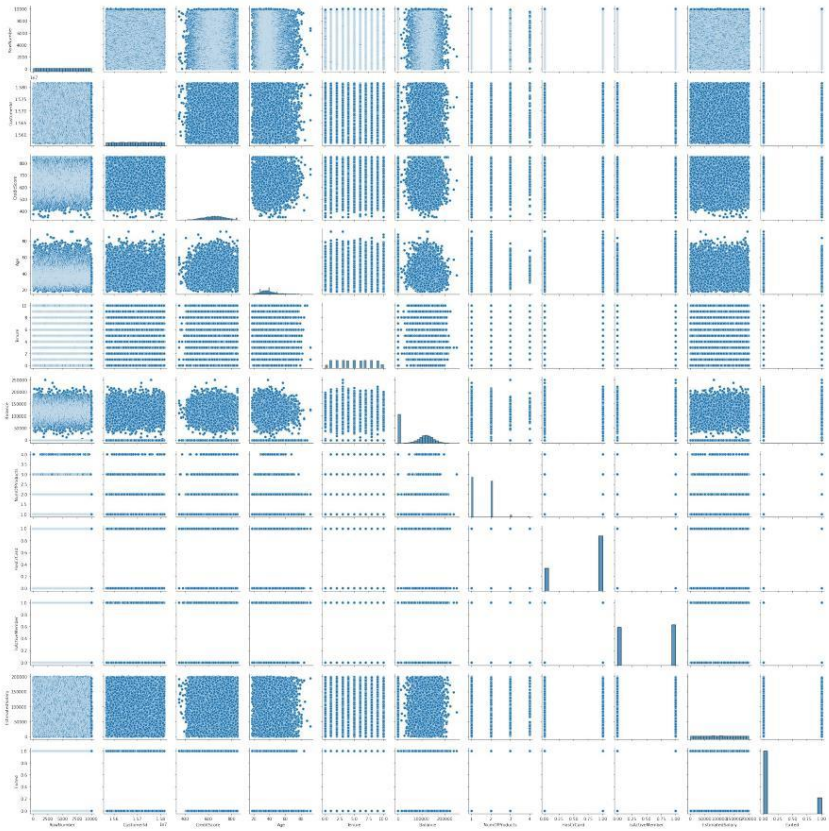
```
import seaborn as sns
sns.displot(data,x="Balance")
```

```
<seaborn.axisgrid.FacetGrid at 0x7fcd2e632850>
```



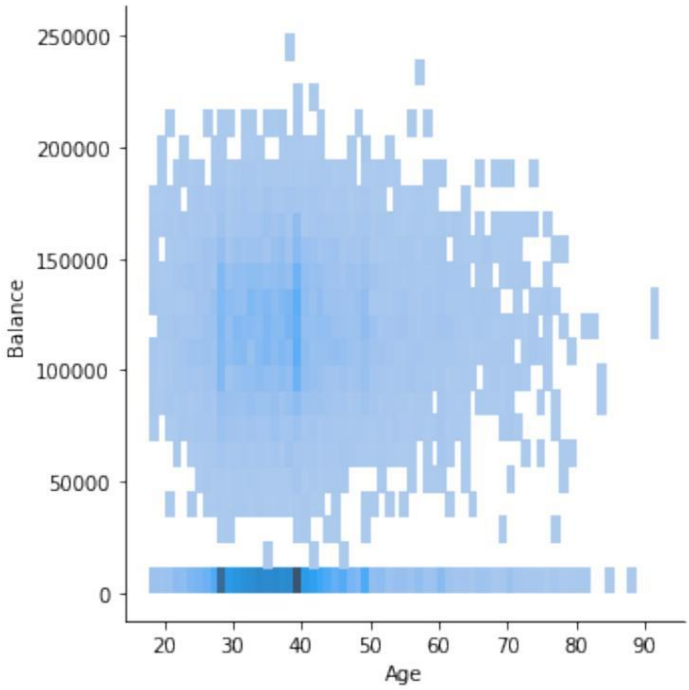

```
sns.pairplot(data)
```

```
<seaborn.axisgrid.PairGrid at 0x7fcd2a4cb290>
```



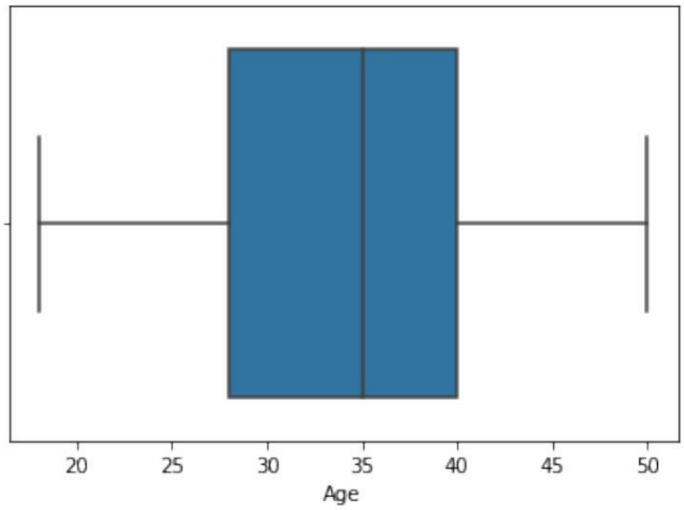
```
sns.displot(data,x="Age",y="Balance")
```

```
<seaborn.axisgrid.FacetGrid at 0x7fcd277eea50>
```




```
#outliers and replacing them
data['Age']=np.where(data['Age']>50,20,data['Age'])

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 0x7fcd25881190>
```




```
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()
```

```
/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          0.000000
CustomerId         0.001149
CreditScore       -0.071607
Age                1.011320
Tenure             0.010991
Balance           -0.141109
NumOfProducts      0.745568
HasCrCard         -0.901812
IsActiveMember    -0.060437
EstimatedSalary    0.002085
Exited             1.471611
dtype: float64
```

```
data.kurt()
```

```
/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         -1.200000
CustomerId        -1.196113
```

```
CreditScore       -0.425726
Age                1.395347
Tenure            -1.165225
Balance           -1.489412
NumOfProducts      0.582981
HasCrCard         -1.186973
IsActiveMember    -1.996747
EstimatedSalary   -1.181518
Exited             0.165671
dtype: float64
```

```
data.var()
```

```
/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          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
```

```
data.std()
```

```
/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           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
```

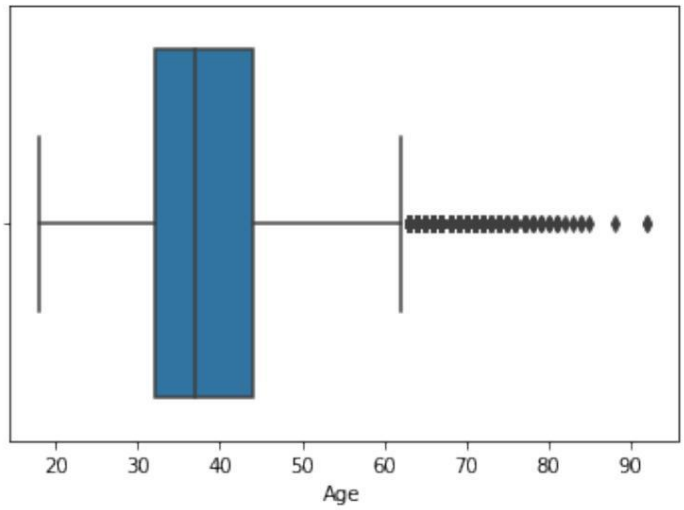
```
data.isna().sum()
```

```
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
```

```
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 0x7f7c13402550>
```




```
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
```

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

```
      RowNumber  CustomerId    Surname  CreditScore Geography  Gender
Age  \
9995       9996    15606229   Obijiaku          771    France       0
39
9996       9997    15569892  Johnstone          516    France       0
35
9997       9998    15584532        Liu          709    France       1
36
9998       9999    15682355  Sabbatini          772   Germany       0
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=['Geography'])
data_main
```

```
      RowNumber  CustomerId    Surname  CreditScore  Gender  Age
Tenure  \
0             1    15634602   Hargrave          619       1   42
2
1             2    15647311       Hill          608       1   41
1
2             3    15619304       Onio          502       1   42
8
3             4    15701354       Boni          699       1   39
1
4             5    15737888   Mitchell          850       1   43
2
...         ...         ...        ...          ...     ...  ...     .
..
9995       9996    15606229   Obijiaku          771       0   39
5
9996       9997    15569892  Johnstone          516       0   35
10
```

```
9997       9998    15584532        Liu          709       1   36
7
9998       9999    15682355  Sabbatini          772       0   42
3
9999      10000    15628319     Walker          792       1   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  Geography_France  Geography_Germany  Geography_Spain
0          1                 1                  0                0
1          0                 0                  0                1
2          1                 1                  0                0
3          0                 1                  0                0
4          0                 0                  0                1
...      ...               ...                ...              ...
9995       0                 1                  0                0
9996       0                 1                  0                0
9997       1                 1                  0                0
9998       1                 0                  1                0
9999       0                 1                  0                0

[10000 rows x 16 columns]
```

```
y=data_main['Exited']
y.head()
```

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

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

```
   RowNumber  CustomerId  CreditScore  Gender  Age  Tenure    Balance
\
0          1    15634602          619       1   42       2       0.00

1          2    15647311          608       1   41       1   83807.86

2          3    15619304          502       1   42       8  159660.80

3          4    15701354          699       1   39       1       0.00

4          5    15737888          850       1   43       2  125510.82


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

1              1          0               1        112542.58       0

2              3          1               0        113931.57       1

3              2          0               0         93826.63       0

4              1          1               1         79084.10       0


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

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

```
array([[-1.73187761, -0.78321342, -0.32622142, ...,  0.99720391,
        -0.57873591, -0.57380915],
       [-1.7315312 , -0.60653412, -0.44003595, ..., -1.00280393,
```

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

```
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, 15)
```

```
x_test.shape
```

```
(2000, 15)
```

```
y_train.shape
```

```
(8000,)
```

```
y_test.shape
```

```
(2000,)
```

If you're already familiar with Colab, check out this video to learn about interactive tables, the executed code history view, and the command palette.

Colab, or "Colaboratory", allows you to write and execute Python in your browser, with

- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch Introduction to Colab to learn more, or just get started below!

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
```