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

```
df=pd.read_csv('/content/Churn_Modelling.csv')
```

```
df.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
```

#univariate analysis

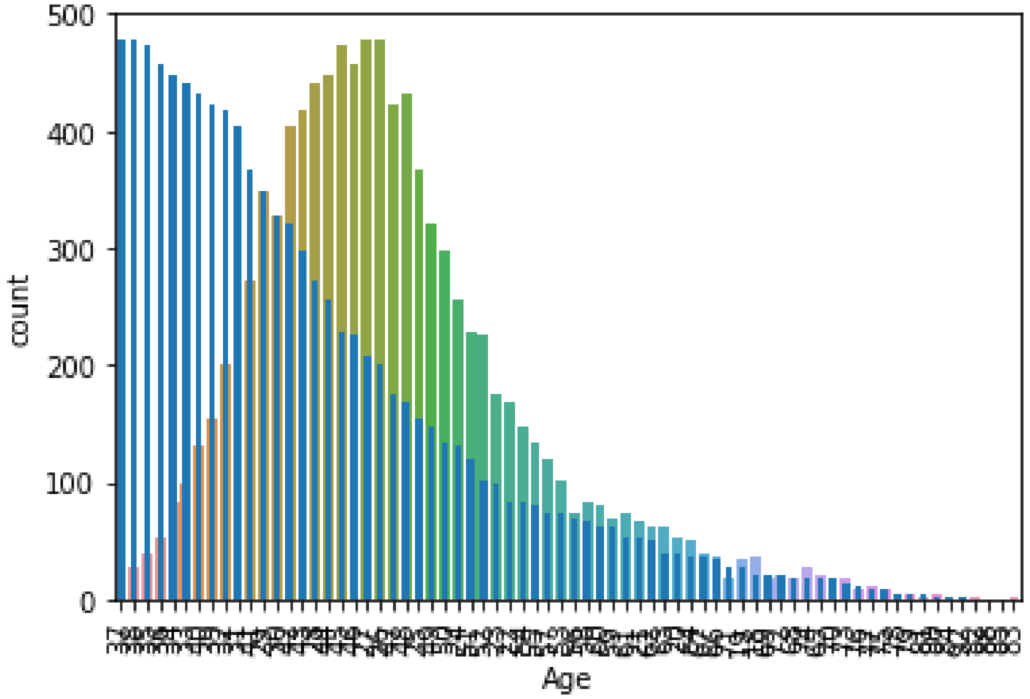
#categorical data

a.countplot

```
sns.countplot(df['Age'])
df['Age'].value_counts().plot(kind='bar')
```

```
/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
```

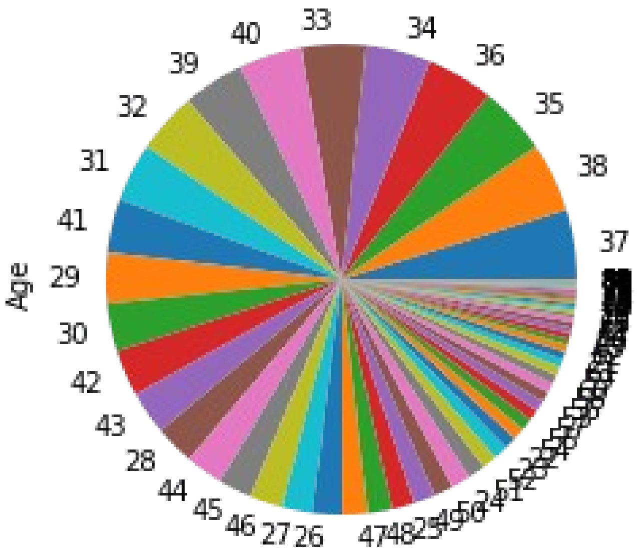
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f629de0ae10>
```



b.piechart

```
df['Age'].value_counts().plot(kind='pie')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f629db16490>
```

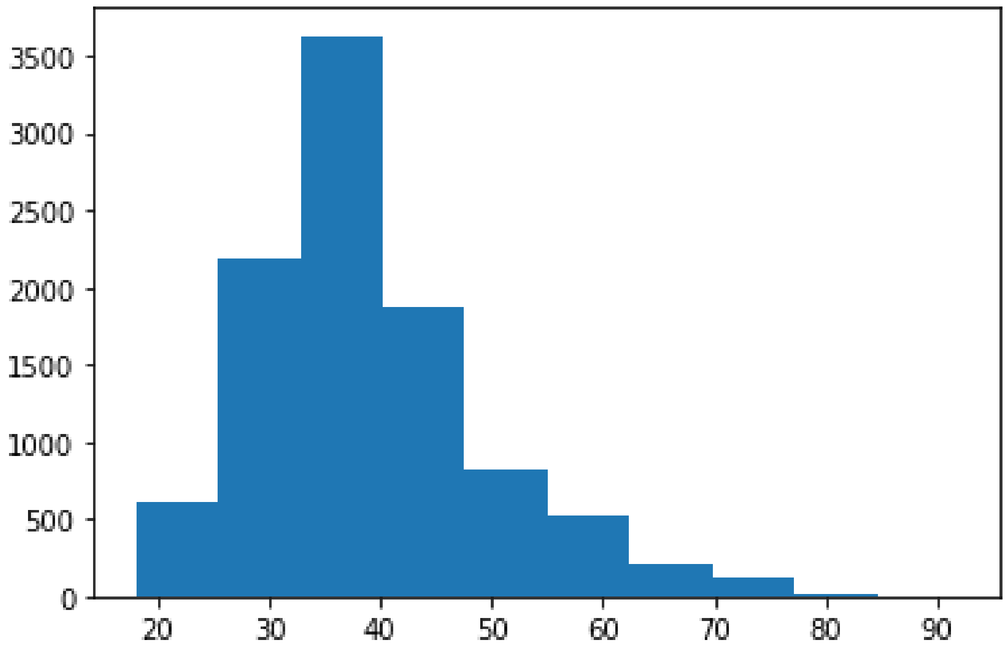


#numerical data

a. histogram

```
import matplotlib.pyplot as plt
plt.hist(df['Age'])
```

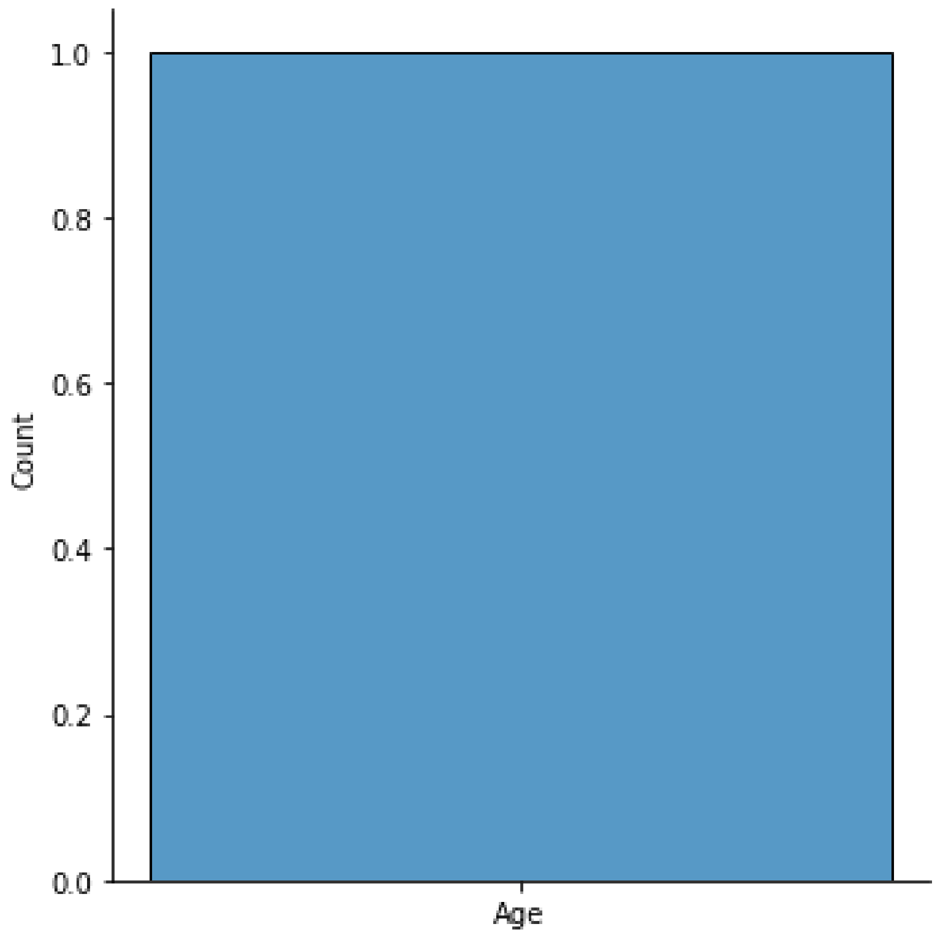
```
(array([ 611., 2179., 3629., 1871.,  828.,  523.,  208.,  127.,   20.,
           4.]),
 array([18. , 25.4, 32.8, 40.2, 47.6, 55. , 62.4, 69.8, 77.2, 84.6,
92. ]),
 <a list of 10 Patch objects>)
```



b.distplot

```
sns.displot(['Age'])
```

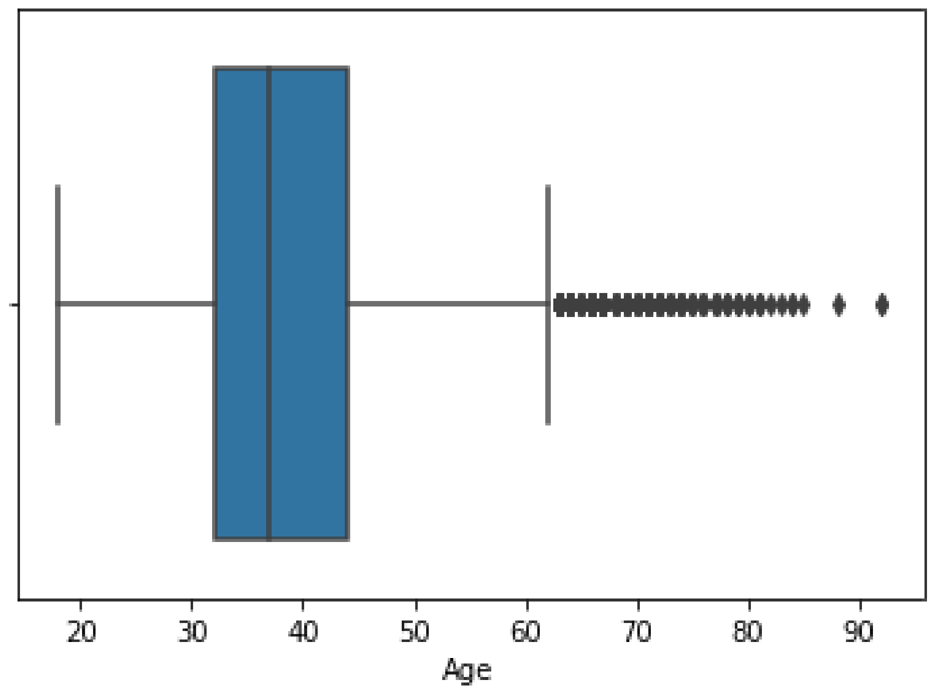
```
<seaborn.axisgrid.FacetGrid at 0x7f629d588c50>
```



c.boxplot

```
sns.boxplot(df['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 0x7f629d527a10>
```



```
df['Age'].max()
```

```
92
```

```
df['Age'].min()
```

```
18
```

```
df['Age'].mean()
```

```
38.9218
```

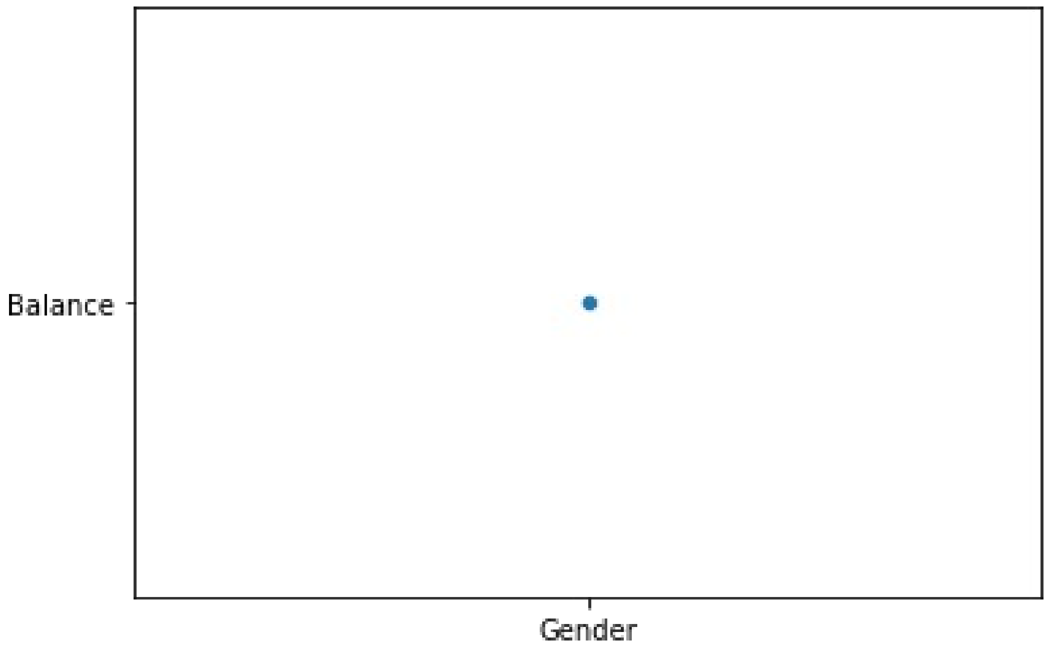
bivariate analysis

1.scatterplot(numerical-numerical

```
sns.scatterplot(['Gender'],['Balance'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: 
FutureWarning: Pass the following variables as keyword args: x, y. 
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 0x7f629b604c50>
```

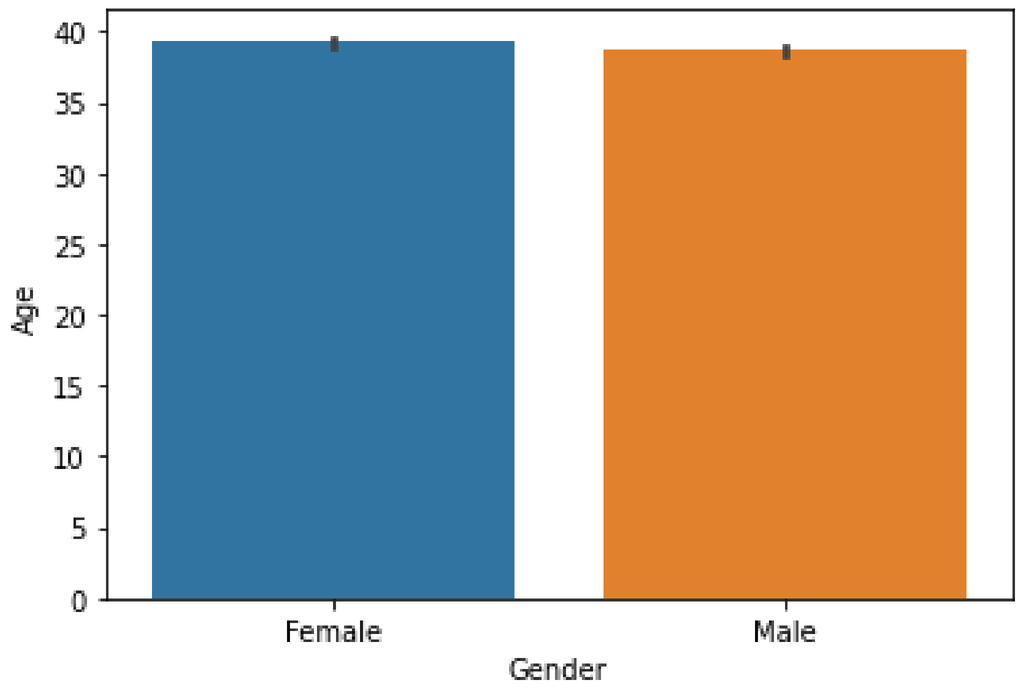


2.bar plot (numerical _categorical)

```
sns.barplot(df['Gender'],df['Age'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variables as keyword args: x, y.
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
```

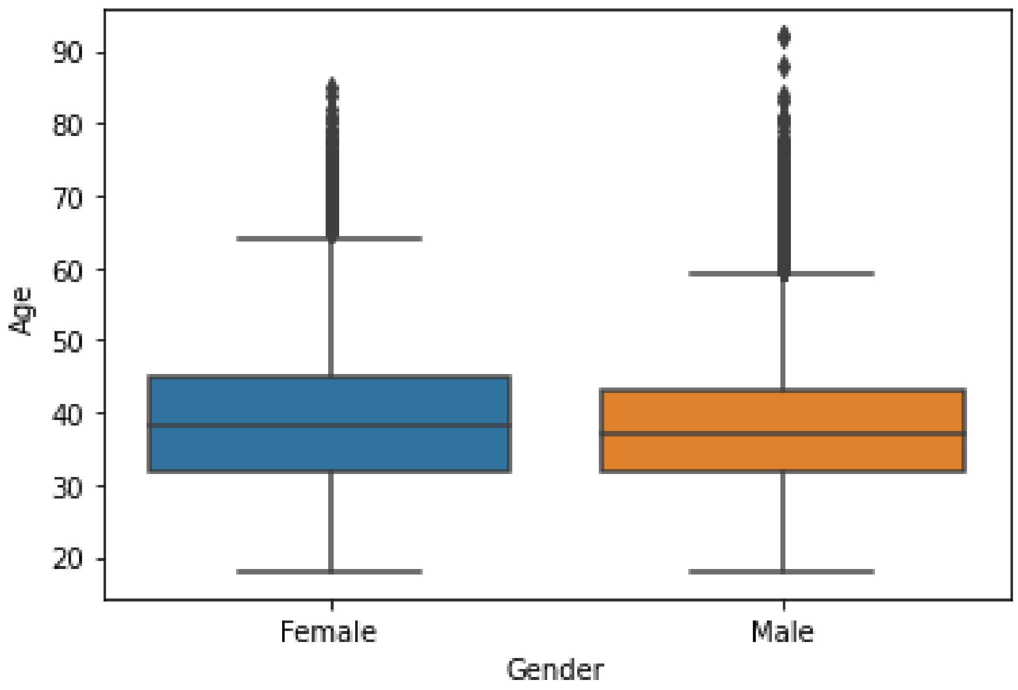
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f629b589250>
```



3.box plot(numerical_categorical)

```
sns.boxplot(df['Gender'],df['Age'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variables as keyword args: x, y.
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 0x7f629b56afd0>
```



4.heatMap(categorical-categorical)

```
df.head(3)
```

```
   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


   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

   EstimatedSalary  Exited
0        101348.88       1
1        112542.58       0
2        113931.57       1
```

```
pd.crosstab(df['Age'],df['Balance'])
```

```
Balance  0.00       3768.69    12459.19   14262.80   16893.59
23503.31   \
Age
```

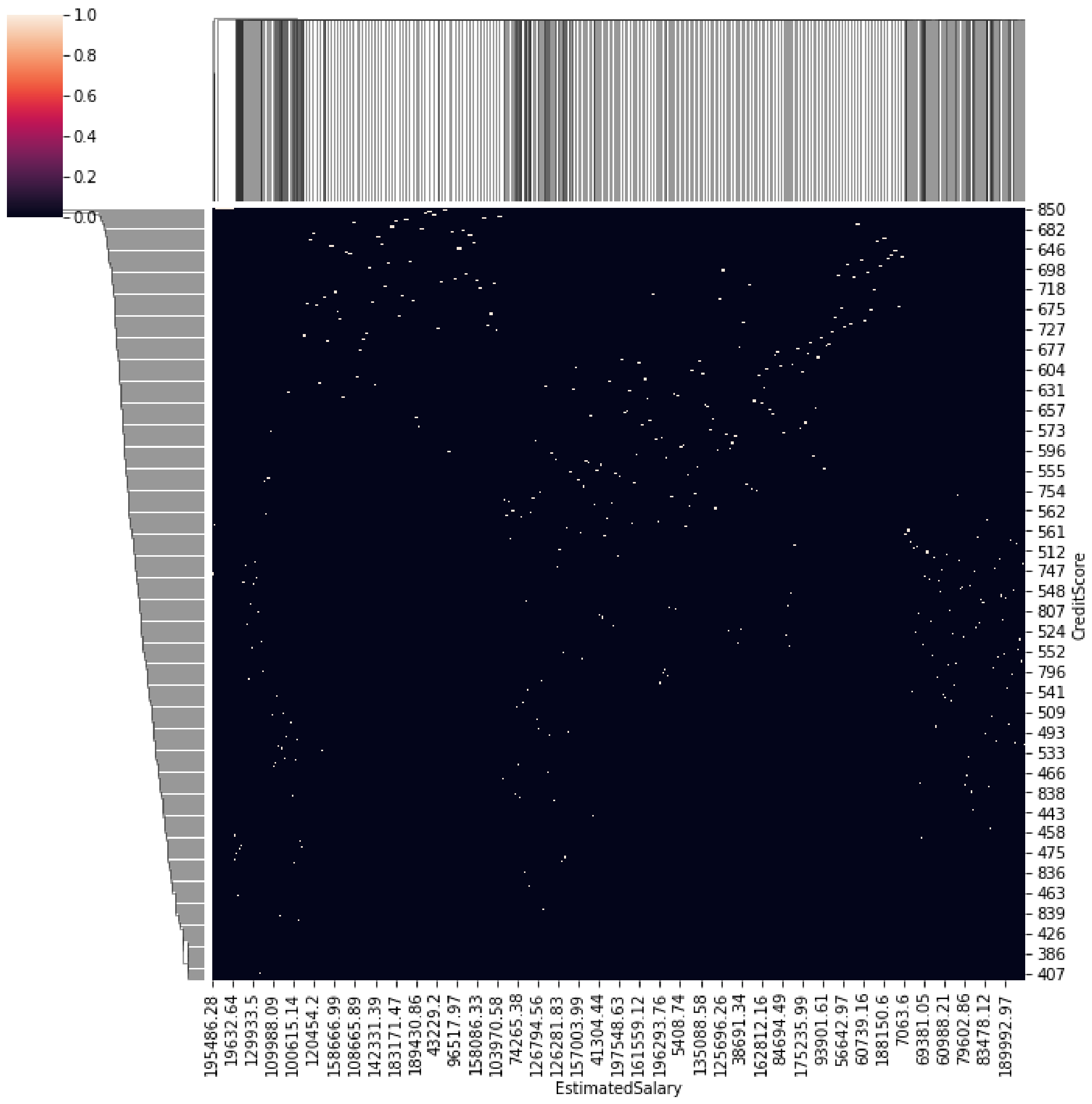
```
18              8          0          0          0          0
0
19             11          0          0          0          0
0
20             17          0          0          0          0
0
21             20          0          0          0          0
0
22             36          0          0          0          0
0
..            ...        ...        ...        ...        ...
...
83              0          0          0          0          0
0
84              0          0          0          0          0
0
85              1          0          0          0          0
0
88              1          0          0          0          0
0
92              0          0          0          0          0
0

Balance  24043.45   27288.43   27517.15   27755.97   ...  212692.97  \
Age                                                  ...
18              0          0          0          0  ...          0
19              0          0          0          0  ...          0
20              0          0          0          0  ...          0
21              0          0          0          0  ...          0
22              0          0          0          0  ...          0
..            ...        ...        ...        ...  ...        ...
83              0          0          0          0  ...          0
84              0          0          0          0  ...          0
85              0          0          0          0  ...          0
88              0          0          0          0  ...          0
92              0          0          0          0  ...          0

Balance  212696.32  212778.20  213146.20  214346.96  216109.88
221532.80  \
Age

18              0          0          0          0          0
0
19              0          0          0          0          0
0
20              0          0          0          0          0
0
21              0          0          0          0          0
0
22              0          0          0          0          0
```

```
0
..              ...        ...        ...        ...        ...
...
83                0          0          0          0          0
0
84                0          0          0          0          0
0
85                0          0          0          0          0
0
88                0          0          0          0          0
0
92                0          0          0          0          0
0

Balance  222267.63  238387.56  250898.09
Age
18                0          0          0
19                0          0          0
20                0          0          0
21                0          0          0
22                0          0          0
..              ...        ...        ...
83                0          0          0
84                0          0          0
85                0          0          0
88                0          0          0
92                0          0          0

[70 rows x 6382 columns]
```

6.clusterMap(categorical_categorical)

```
sns.clustermap(pd.crosstab(df['CreditScore'],df['EstimatedSalary']))
```

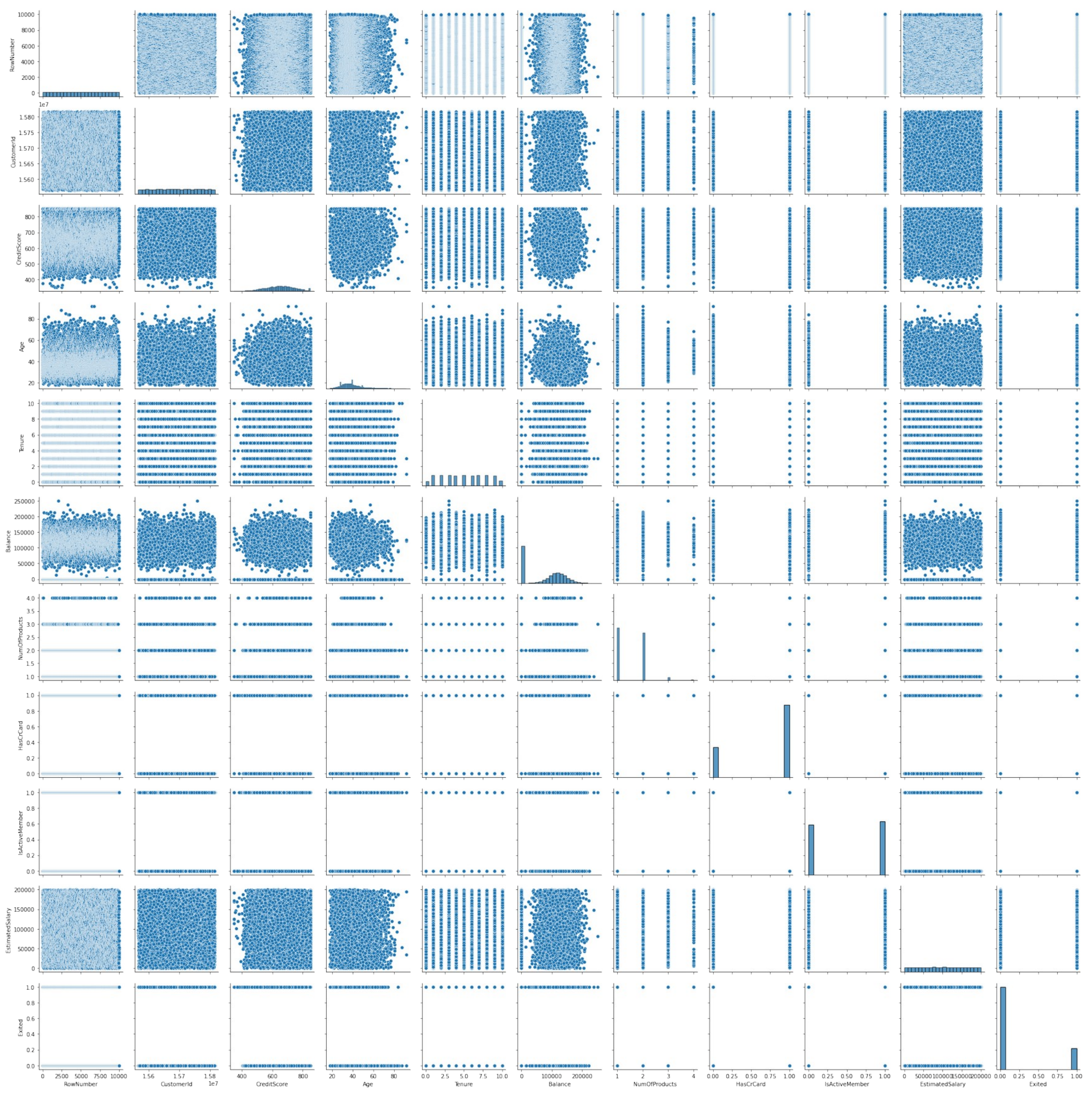
```
/usr/local/lib/python3.7/dist-packages/seaborn/matrix.py:654:
UserWarning: Clustering large matrix with scipy. Installing
`fastcluster` may give better performance.
  warnings.warn(msg)

<seaborn.matrix.ClusterGrid at 0x7f629d5e6710>
```



7.pairplot

```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7f629d4ef150>
```



#descriptive statistical

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

df=pd.read_csv('/content/Churn_Modelling.csv')
df

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

```
42
3             4    15701354       Boni          699    France  Female
39
4             5    15737888   Mitchell          850     Spain  Female
43
...         ...         ...        ...          ...       ...     ...
...
9995       9996    15606229   Obijiaku          771    France    Male
39
9996       9997    15569892  Johnstone          516    France    Male
35
9997       9998    15584532        Liu          709    France  Female
36
9998       9999    15682355  Sabbatini          772   Germany    Male
42
9999      10000    15628319     Walker          792    France  Female
28

      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0          2       0.00              1          1               1
1          1   83807.86              1          0               1
2          8  159660.80              3          1               0
3          1       0.00              2          0               0
4          2  125510.82              1          1               1
...      ...        ...            ...        ...             ...
9995       5       0.00              2          1               0
9996      10   57369.61              1          1               1
9997       7       0.00              1          0               1
9998       3   75075.31              2          1               0
9999       4  130142.79              1          1               0

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

[10000 rows x 14 columns]
```

```
df.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
```

```
df.describe(include=['object'])
```

```
       Surname Geography Gender
count    10000     10000  10000
unique    2932         3      2
top      Smith    France   Male
freq        32      5014   5457
```

```
df['Age'].value_counts()
```

```
37    478
38    477
35    474
36    456
```

```
34    447
     ...
92      2
82      1
88      1
85      1
83      1
Name: Age, Length: 70, dtype: int64
```

```
df['Age'].value_counts().to_frame()
```

```
    Age
37  478
38  477
35  474
36  456
34  447
..  ...
92    2
82    1
88    1
85    1
83    1

[70 rows x 1 columns]
```

```
model_counts.index.name='Balance'
model_counts
```

```
         Age
Balance
37       478
38       477
35       474
36       456
34       447
...      ...
92         2
82         1
88         1
85         1
83         1

[70 rows x 1 columns]
```

#handling missing values

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

```
df=pd.read_csv('/content/Churn_Modelling.csv')
df

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

      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0          2       0.00              1          1               1
1          1   83807.86              1          0               1
2          8  159660.80              3          1               0
3          1       0.00              2          0               0
4          2  125510.82              1          1               1
...      ...        ...            ...        ...             ...
9995       5       0.00              2          1               0
9996      10   57369.61              1          1               1
9997       7       0.00              1          0               1
9998       3   75075.31              2          1               0
9999       4  130142.79              1          1               0

      EstimatedSalary  Exited
0           101348.88       1
1           112542.58       0
2           113931.57       1
3            93826.63       0
4            79084.10       0
...               ...     ...
9995         96270.64       0
9996        101699.77       0
```

```
9997        42085.58       1
9998        92888.52       1
9999        38190.78       0

[10000 rows x 14 columns]
```

```
df.shape
```

```
(10000, 14)
```

```
df.isnull()
```

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

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

      EstimatedSalary  Exited
```

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

[10000 rows x 14 columns]
```

```
df.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
```

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

```
0
```

fill the null values

```
df2 = df.fillna(value=0)
df2
```

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

```
43
...          ...         ...        ...          ...       ...     ...
...
9995       9996    15606229   Obijiaku          771    France    Male
39
9996       9997    15569892  Johnstone          516    France    Male
35
9997       9998    15584532        Liu          709    France  Female
36
9998       9999    15682355  Sabbatini          772   Germany    Male
42
9999      10000    15628319     Walker          792    France  Female
28

      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0          2       0.00              1          1               1
1          1   83807.86              1          0               1
2          8  159660.80              3          1               0
3          1       0.00              2          0               0
4          2  125510.82              1          1               1
...      ...        ...            ...        ...             ...
9995       5       0.00              2          1               0
9996      10   57369.61              1          1               1
9997       7       0.00              1          0               1
9998       3   75075.31              2          1               0
9999       4  130142.79              1          1               0

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

[10000 rows x 14 columns]
```

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

```
0
```

```
df3 = df.fillna(value=5)
df3
```

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

      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0          2       0.00              1          1               1
1          1   83807.86              1          0               1
2          8  159660.80              3          1               0
3          1       0.00              2          0               0
4          2  125510.82              1          1               1
...      ...        ...            ...        ...             ...
9995       5       0.00              2          1               0
9996      10   57369.61              1          1               1
9997       7       0.00              1          0               1
9998       3   75075.31              2          1               0
9999       4  130142.79              1          1               0

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

[10000 rows x 14 columns]
```

filling null value with privious value

```
df4 = df.fillna(method='pad')
df4
```

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

      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0          2       0.00              1          1               1   
1          1   83807.86              1          0               1   
2          8  159660.80              3          1               0   
3          1       0.00              2          0               0   
4          2  125510.82              1          1               1   
...      ...        ...            ...        ...             ...   
9995       5       0.00              2          1               0   
9996      10   57369.61              1          1               1   
9997       7       0.00              1          0               1   
9998       3   75075.31              2          1               0   
9999       4  130142.79              1          1               0   

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

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

[10000 rows x 14 columns]
```

```
df4.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
```

```
#filling null values with next values
df5 = df.fillna(method='bfill')
df5
```

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

```
42
9999      10000    15628319     Walker          792    France  Female
28

      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0          2       0.00              1          1               1
1          1   83807.86              1          0               1
2          8  159660.80              3          1               0
3          1       0.00              2          0               0
4          2  125510.82              1          1               1
...      ...        ...            ...        ...             ...
9995       5       0.00              2          1               0
9996      10   57369.61              1          1               1
9997       7       0.00              1          0               1
9998       3   75075.31              2          1               0
9999       4  130142.79              1          1               0

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

[10000 rows x 14 columns]
```

```
df6 =df.fillna(method='pad',axis=1)
df6
```

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

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

        Balance NumOfProducts HasCrCard IsActiveMember EstimatedSalary
Exited
0           0.0             1         1              1       101348.88
1
1      83807.86             1         0              1       112542.58
0
2      159660.8             3         1              0       113931.57
1
3           0.0             2         0              0        93826.63
0
4     125510.82             1         1              1         79084.1
0
...         ...           ...       ...            ...             ...
...
9995        0.0             2         1              0        96270.64
0
9996   57369.61             1         1              1       101699.77
0
9997        0.0             1         0              1        42085.58
1
9998   75075.31             2         1              0        92888.52
1
9999  130142.79             1         1              0        38190.78
0

[10000 rows x 14 columns]
```

```
df7 =df.fillna(method='bfill',axis=1)
df7
```

```
     RowNumber CustomerId    Surname CreditScore Geography  Gender Age
Tenure  \
0            1   15634602   Hargrave         619    France  Female  42
2
1            2   15647311       Hill         608     Spain  Female  41
1
2            3   15619304       Onio         502    France  Female  42
8
3            4   15701354       Boni         699    France  Female  39
1
```

```
4            5   15737888   Mitchell         850     Spain  Female  43
2
...        ...        ...        ...         ...       ...     ...  ..
...
9995      9996   15606229   Obijiaku         771    France    Male  39
5
9996      9997   15569892  Johnstone         516    France    Male  35
10
9997      9998   15584532        Liu         709    France  Female  36
7
9998      9999   15682355  Sabbatini         772   Germany    Male  42
3
9999     10000   15628319     Walker         792    France  Female  28
4

        Balance NumOfProducts HasCrCard IsActiveMember EstimatedSalary
Exited
0           0.0             1         1              1       101348.88
1
1      83807.86             1         0              1       112542.58
0
2      159660.8             3         1              0       113931.57
1
3           0.0             2         0              0        93826.63
0
4     125510.82             1         1              1         79084.1
0
...         ...           ...       ...            ...             ...
...
9995        0.0             2         1              0        96270.64
0
9996   57369.61             1         1              1       101699.77
0
9997        0.0             1         0              1        42085.58
1
9998   75075.31             2         1              0        92888.52
1
9999  130142.79             1         1              0        38190.78
0

[10000 rows x 14 columns]
```

filling different values in null in different columns

```
df8=df.fillna({'HasCrCard' : 'abcd',
               'Balance':'def'})
df8
```

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

```
42
1             2    15647311       Hill         608     Spain  Female
41
2             3    15619304       Onio         502    France  Female
42
3             4    15701354       Boni         699    France  Female
39
4             5    15737888   Mitchell         850     Spain  Female
43
...         ...         ...        ...         ...       ...     ...
...
9995       9996    15606229   Obijiaku         771    France    Male
39
9996       9997    15569892  Johnstone         516    France    Male
35
9997       9998    15584532        Liu         709    France  Female
36
9998       9999    15682355  Sabbatini         772   Germany    Male
42
9999      10000    15628319     Walker         792    France  Female
28

      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0          2       0.00              1          1               1
1          1   83807.86              1          0               1
2          8  159660.80              3          1               0
3          1       0.00              2          0               0
4          2  125510.82              1          1               1
...      ...        ...            ...        ...             ...
9995       5       0.00              2          1               0
9996      10   57369.61              1          1               1
9997       7       0.00              1          0               1
9998       3   75075.31              2          1               0
9999       4  130142.79              1          1               0

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

[10000 rows x 14 columns]
```

```python
#interpolate()
df['Age']=df['Age'].interpolate(method='linear')
df
```

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

      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0          2       0.00              1          1               1
1          1   83807.86              1          0               1
2          8  159660.80              3          1               0
3          1       0.00              2          0               0
4          2  125510.82              1          1               1
...      ...        ...            ...        ...             ...
9995       5       0.00              2          1               0
9996      10   57369.61              1          1               1
9997       7       0.00              1          0               1
9998       3   75075.31              2          1               0
9999       4  130142.79              1          1               0

      EstimatedSalary  Exited
0           101348.88       1
1           112542.58       0
2           113931.57       1
3            93826.63       0
4            79084.10       0
...               ...     ...
9995         96270.64       0
```

```
9996        101699.77       0
9997         42085.58       1
9998         92888.52       1
9999         38190.78       0

[10000 rows x 14 columns]
```

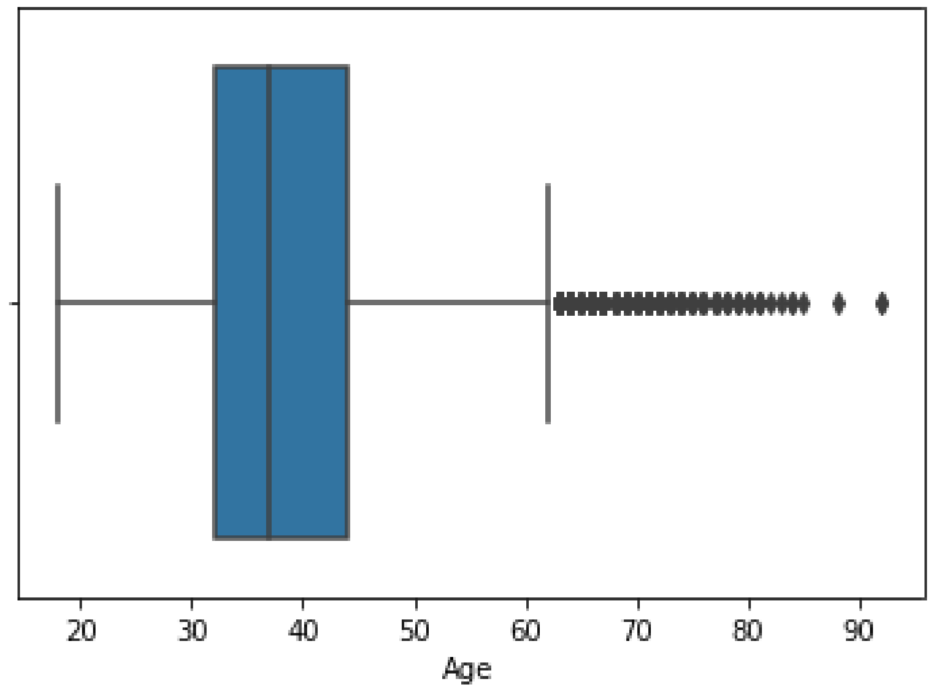
#find the outliers and replace the outliers

```
df.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
```

```
#to see outliers clearly
sns.boxplot(df['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 0x7f9eafe7a850>
```



```
#z-score method
upper_limit=df['Age'].mean()+3*df['Age'].std()
lower_limit=df['Age'].mean()-3*df['Age'].std()
print('upper limit', upper_limit)
print('lower limit', lower_limit)
```

```
upper limit 70.38521935511383
lower limit 7.458380644886169
```

```
df_new=df[(df.Age<=7.458380644886169)&(df.NumOfProducts>=4)]
df_new.shape
```

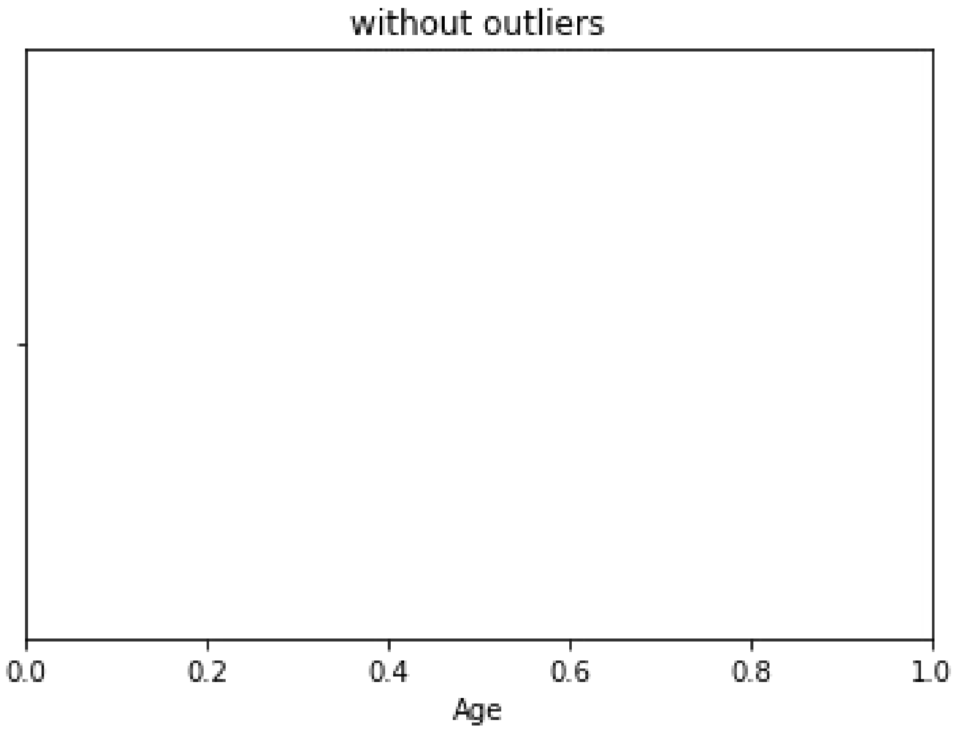
```
(0, 14)
```

```
sns.boxplot(df_new['Age']).set_title('without outliers')
```

```
/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
```

```
Text(0.5, 1.0, 'without outliers')
```



#check for categorical columns and perform encoding

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   RowNumber        10000 non-null  int64  
 1   CustomerId       10000 non-null  int64  
 2   Surname          10000 non-null  object 
 3   CreditScore      10000 non-null  int64  
 4   Geography        10000 non-null  object 
 5   Gender           10000 non-null  object 
 6   Age              10000 non-null  int64  
 7   Tenure           10000 non-null  int64  
 8   Balance          10000 non-null  float64
 9   NumOfProducts    10000 non-null  int64  
 10  HasCrCard        10000 non-null  int64  
 11  IsActiveMember   10000 non-null  int64  
 12  EstimatedSalary  10000 non-null  float64
 13  Exited           10000 non-null  int64
dtypes: float64(2), int64(9), object(3)
memory usage: 1.1+ MB
```

```
df.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
```

```
df['Geography'].value_counts()
```

```
France     5014
Germany    2509
Spain      2477
Name: Geography, dtype: int64
```

```
from sklearn.preprocessing import LabelEncoder
```

```
le = LabelEncoder()
```

```
df['Geography'] = le.fit_transform(df['Geography'])
df.head()
```

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

#split the data into dependent and independent variables

```
x = df.iloc[:,0:7].values
y = df.iloc[:,7:8].values
```

```
x
```

```
array([[1, 15634602, 'Hargrave', ..., 0, 'Female', 42],
       [2, 15647311, 'Hill', ..., 2, 'Female', 41],
       [3, 15619304, 'Onio', ..., 0, 'Female', 42],
       ...,
       [9998, 15584532, 'Liu', ..., 0, 'Female', 36],
       [9999, 15682355, 'Sabbatini', ..., 1, 'Male', 42],
       [10000, 15628319, 'Walker', ..., 0, 'Female', 28]],
dtype=object)
```

```
y
```

```
array([[2],
       [1],
       [8],
       ...,
       [7],
       [3],
       [4]])
```

```
x.shape
```

```
(10000, 7)
```

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

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

```
y.shape
```

```
(10000, 1)
```

```
print(type(y))
```

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

#split the data into testing and training

```
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,random_state=0)
```

```
x_train.shape
```

```
(7000, 7)
```

```
x_test.shape
```

```
(3000, 7)
```

```
y_train.shape
```

```
(7000, 1)
```

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

```
(7000, 1)
```

```
print(y_test.shape)
```

```
(3000, 1)
```