ASSIGNMENT II

Name:Maheswari B

Reg.No:913119205023

Importing Libraries

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import scale

#Uploading CSV file
df=pd.read_csv('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
```

```
df.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

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

#Exploratory Data Analysis

df.shape #shape of the data
```

```
(10000, 14)
```

```
#Duplicate available or not
df.duplicated()
```

```
0       False
1       False
2       False
3       False
4       False
        ...
9995    False
9996    False
9997    False
9998    False
9999    False
Length: 10000, dtype: bool
```

```
df.columns
```

```
Index(['RowNumber', 'CustomerId', 'Surname', 'CreditScore',
'Geography',
       'Gender', 'Age', 'Tenure', 'Balance', 'NumOfProducts',
'HasCrCard',
       'IsActiveMember', 'EstimatedSalary', 'Exited'],
      dtype='object')
```

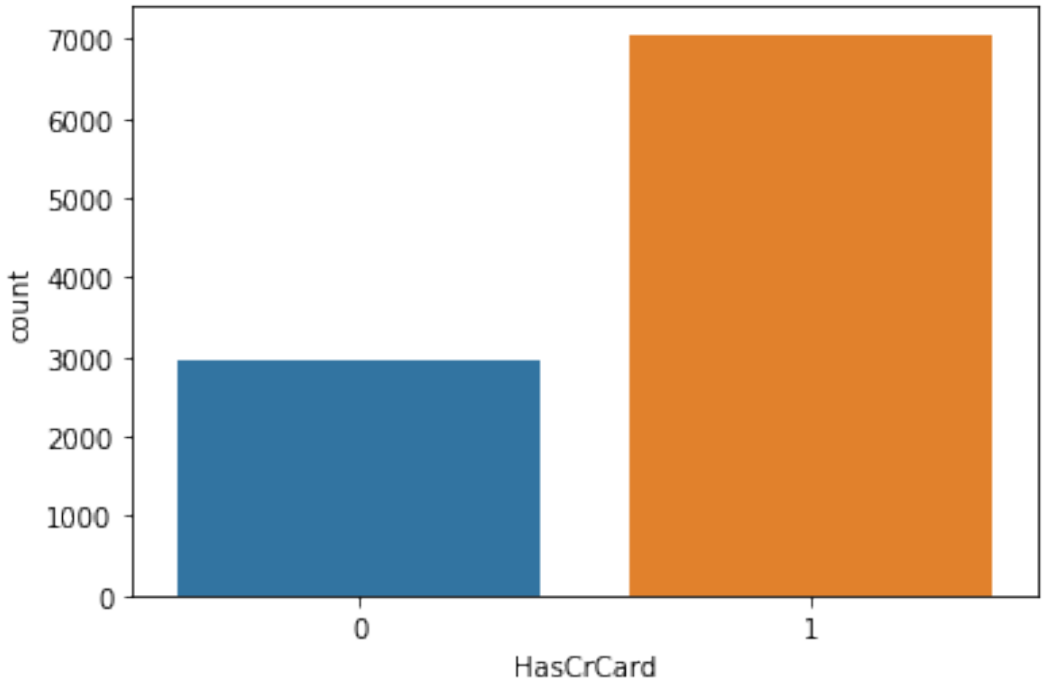
```
#Correlation
df.corr()
```

```
                 RowNumber  CustomerId  CreditScore       Age
Tenure  \
RowNumber         1.000000    0.004202     0.005840  0.000783 -
0.006495
CustomerId        0.004202    1.000000     0.005308  0.009497 -
0.014883
CreditScore       0.005840    0.005308     1.000000 -0.003965
0.000842
Age               0.000783    0.009497    -0.003965  1.000000 -
0.009997
Tenure           -0.006495   -0.014883     0.000842 -0.009997
1.000000
Balance          -0.009067   -0.012419     0.006268  0.028308 -
0.012254
NumOfProducts     0.007246    0.016972     0.012238 -0.030680
0.013444
HasCrCard         0.000599   -0.014025    -0.005458 -0.011721
0.022583
IsActiveMember    0.012044    0.001665     0.025651  0.085472 -
0.028362
EstimatedSalary  -0.005988    0.015271    -0.001384 -0.007201
```

```
0.007784
Exited           -0.016571   -0.006248    -0.027094  0.285323 -
0.014001

                  Balance  NumOfProducts  HasCrCard  IsActiveMember  \
RowNumber       -0.009067       0.007246   0.000599        0.012044
CustomerId      -0.012419       0.016972  -0.014025        0.001665
CreditScore      0.006268       0.012238  -0.005458        0.025651
Age              0.028308      -0.030680  -0.011721        0.085472
Tenure          -0.012254       0.013444   0.022583       -0.028362
Balance          1.000000      -0.304180  -0.014858       -0.010084
NumOfProducts   -0.304180       1.000000   0.003183        0.009612
HasCrCard       -0.014858       0.003183   1.000000       -0.011866
IsActiveMember  -0.010084       0.009612  -0.011866        1.000000
EstimatedSalary  0.012797       0.014204  -0.009933       -0.011421
Exited           0.118533      -0.047820  -0.007138       -0.156128

                 EstimatedSalary    Exited
RowNumber              -0.005988 -0.016571
CustomerId              0.015271 -0.006248
CreditScore            -0.001384 -0.027094
Age                    -0.007201  0.285323
Tenure                  0.007784 -0.014001
Balance                 0.012797  0.118533
NumOfProducts           0.014204 -0.047820
HasCrCard              -0.009933 -0.007138
IsActiveMember         -0.011421 -0.156128
EstimatedSalary         1.000000  0.012097
Exited                  0.012097  1.000000
```

```
#Univariate Analysis

#Categorical Data

#Countplot
sns.countplot(df['HasCrCard'])
```

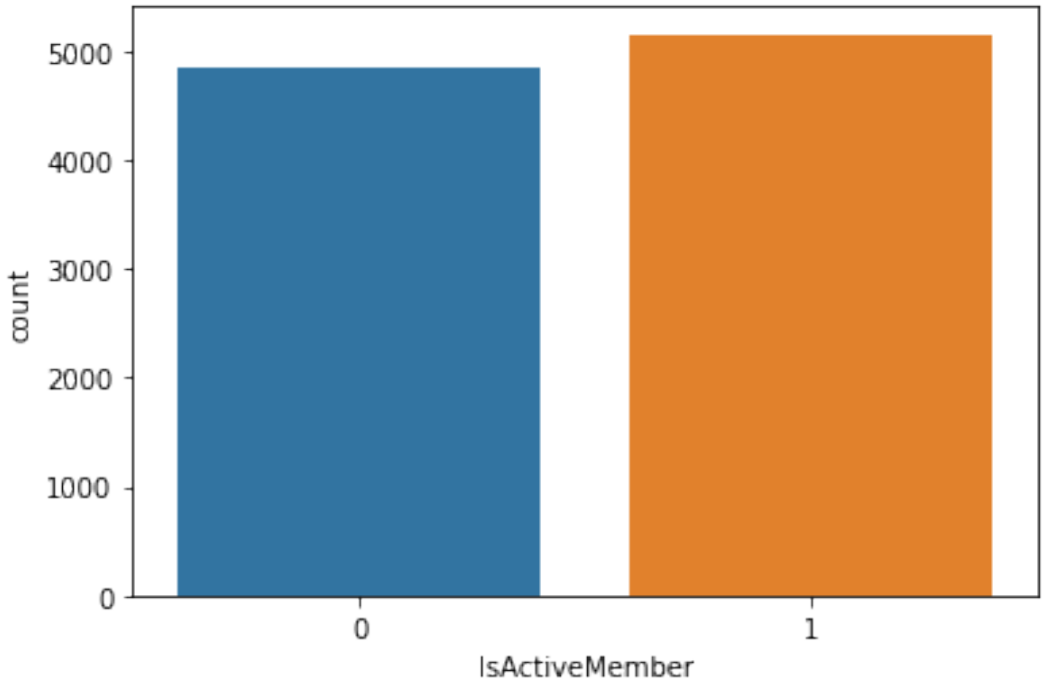
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be `data`, and
passing other arguments without an explicit keyword will result in an
error or misinterpretation.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

<matplotlib.axes._subplots.AxesSubplot at 0x7f4aeb3ba150>
```




```
sns.countplot(df['IsActiveMember'])
```

```
/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
```

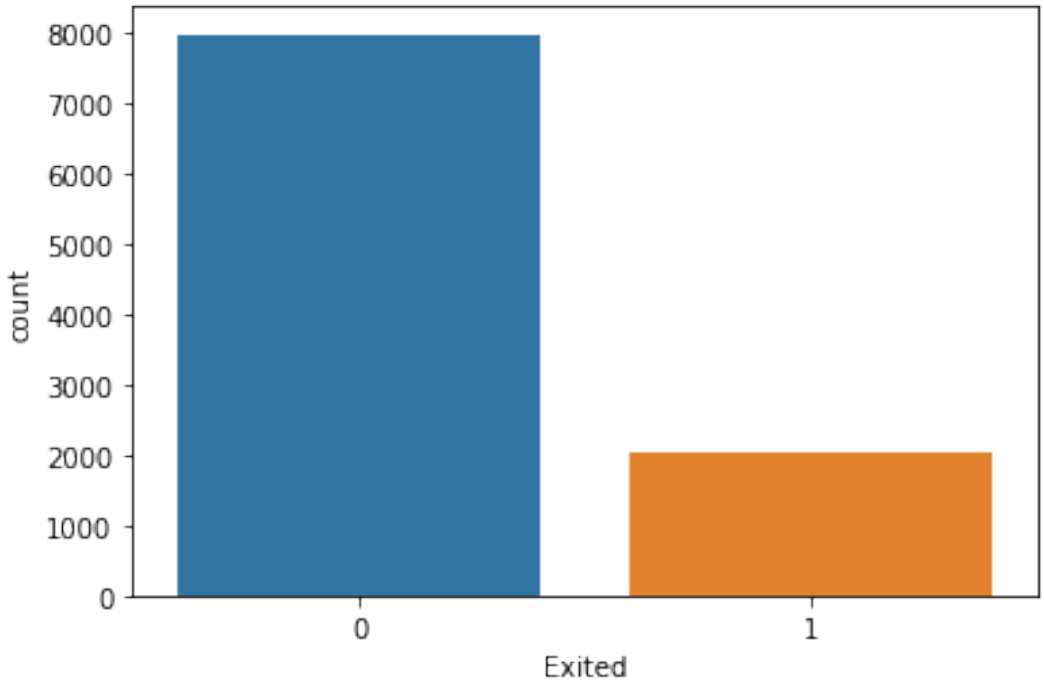
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4aeb2a5f90>
```




```
sns.countplot(df['Exited'])
```

```
/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
```

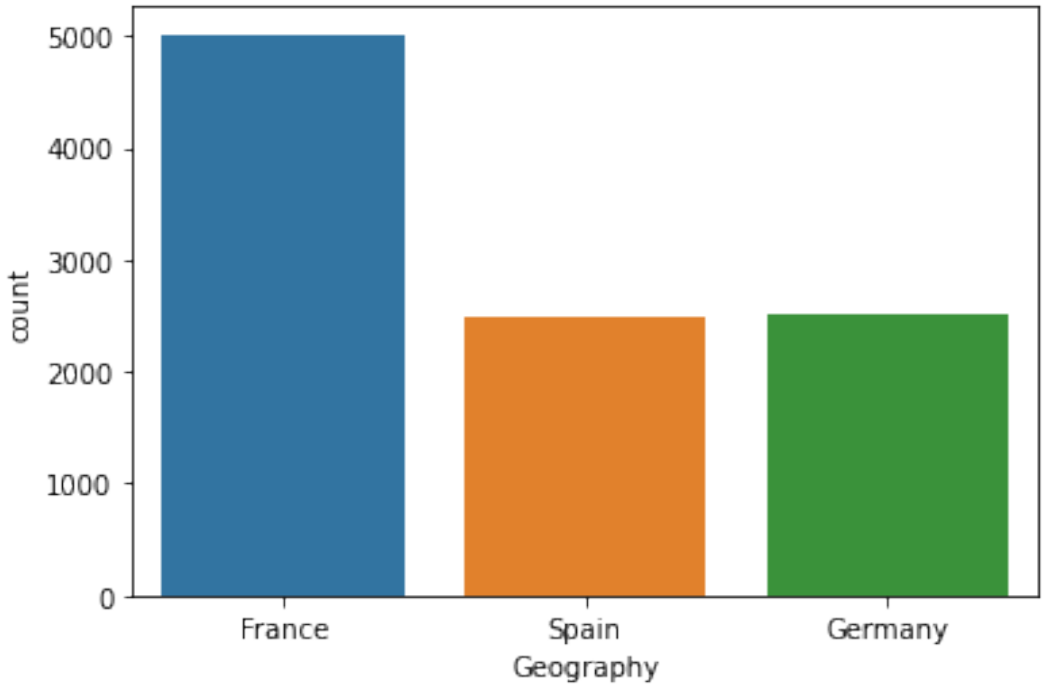
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4aeadc6e10>
```




```
sns.countplot(df['Geography'])
```

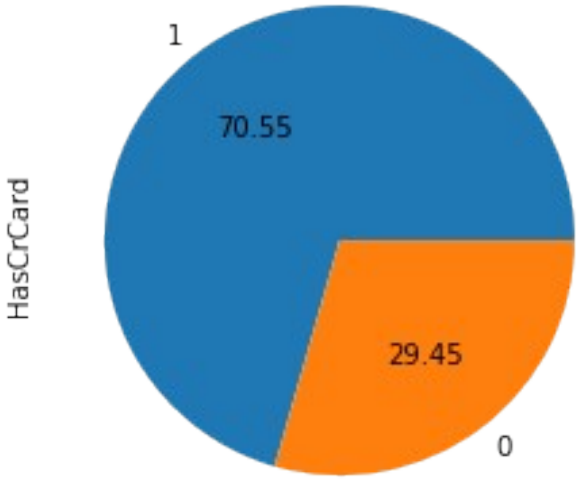
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be `data`, and
passing other arguments without an explicit keyword will result in an
error or misinterpretation.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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4aead93850>
```



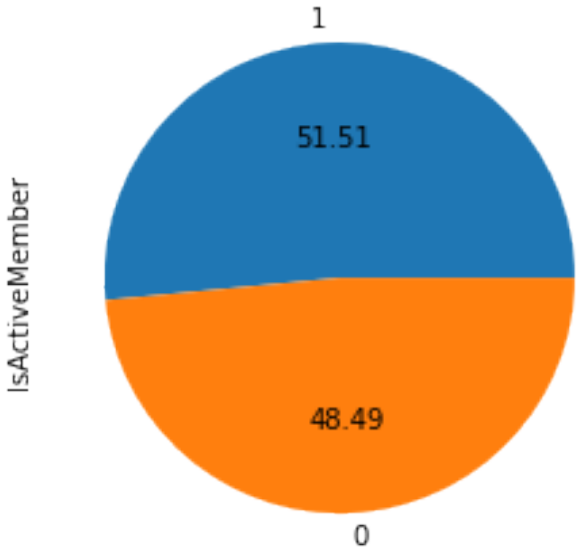

```
#Pie chart
df['HasCrCard'].value_counts().plot(kind='pie',autopct='%.2f')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4aeaceabd0>
```



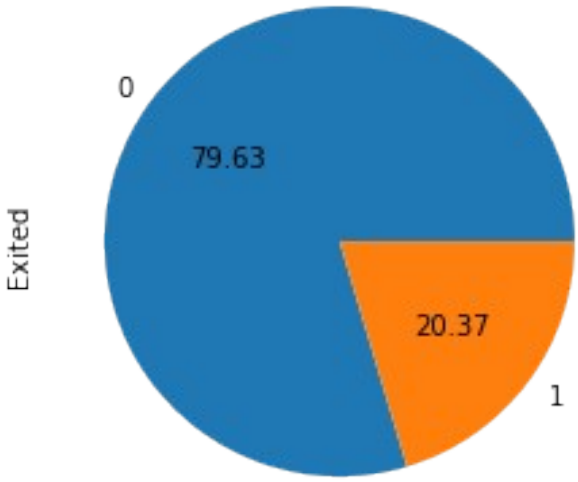

```
df['IsActiveMember'].value_counts().plot(kind='pie',autopct='%.2f')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4aeac55390>
```




```
df['Exited'].value_counts().plot(kind='pie',autopct='%.2f')
```

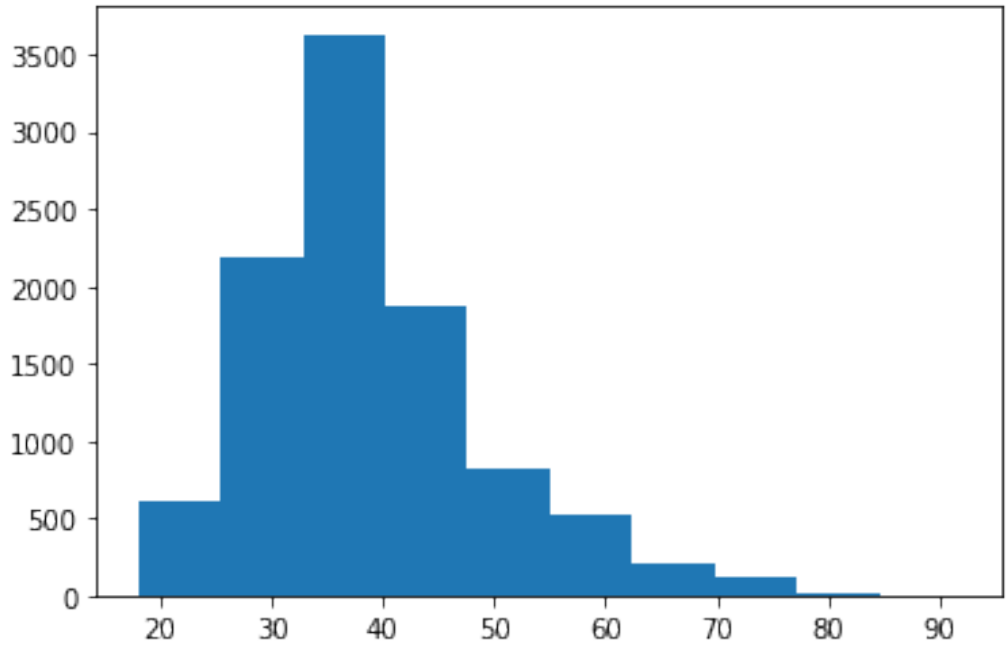
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4aeac21310>
```




```
#Numerical Data

#Histogram
plt.hist(df['Age'])
```

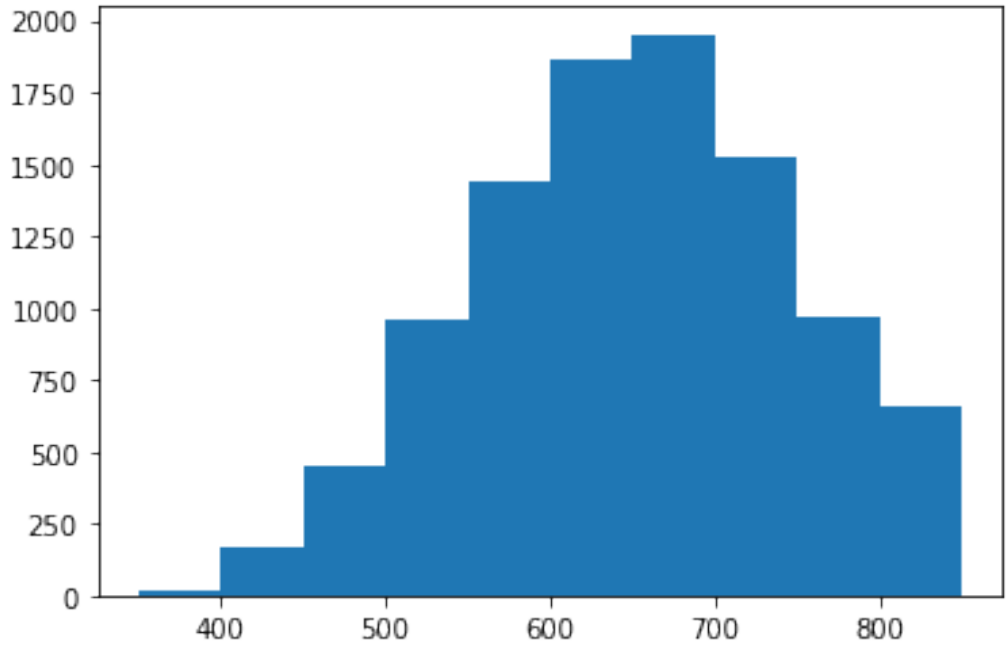
```
(array([ 611., 2179., 3629., 1871.,  828.,  523.,  208.,  127.,   20.,
           4.]),
 array([18. , 25.4, 32.8, 40.2, 47.6, 55. , 62.4, 69.8, 77.2, 84.6,
```

```
92. ]),
 <a list of 10 Patch objects>)
```




```
plt.hist(df['CreditScore'])
```

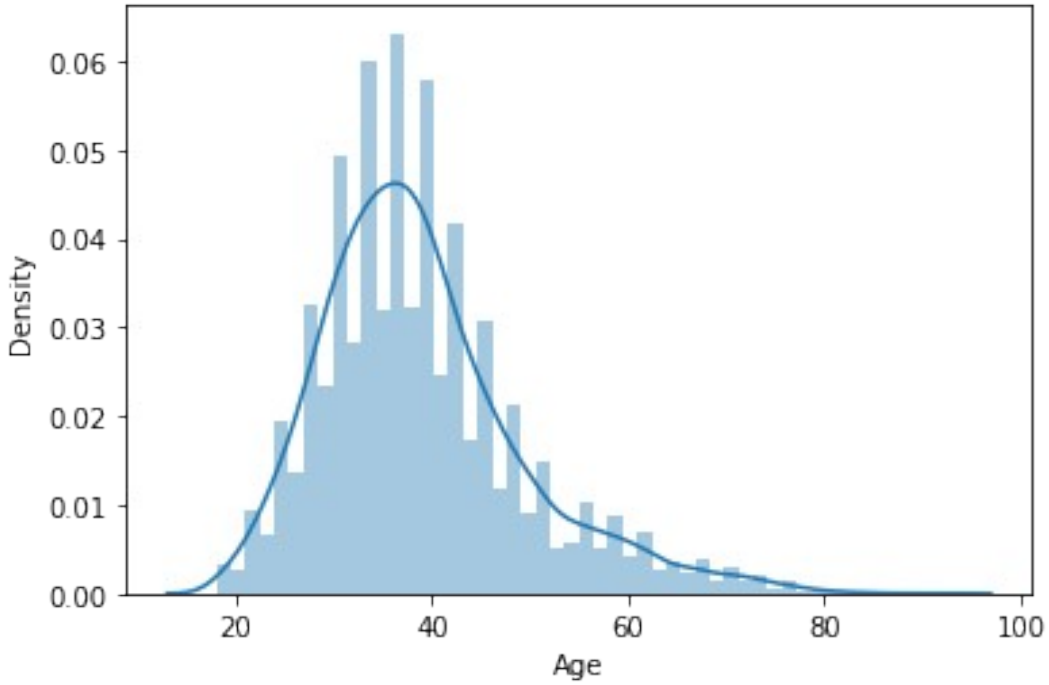
```
(array([  19.,  166.,  447.,  958., 1444., 1866., 1952., 1525.,  968.,
         655.]),
 array([350., 400., 450., 500., 550., 600., 650., 700., 750., 800.,
850.]),
 <a list of 10 Patch objects>)
```



```
#Distplot
sns.distplot(df['Age'])
```

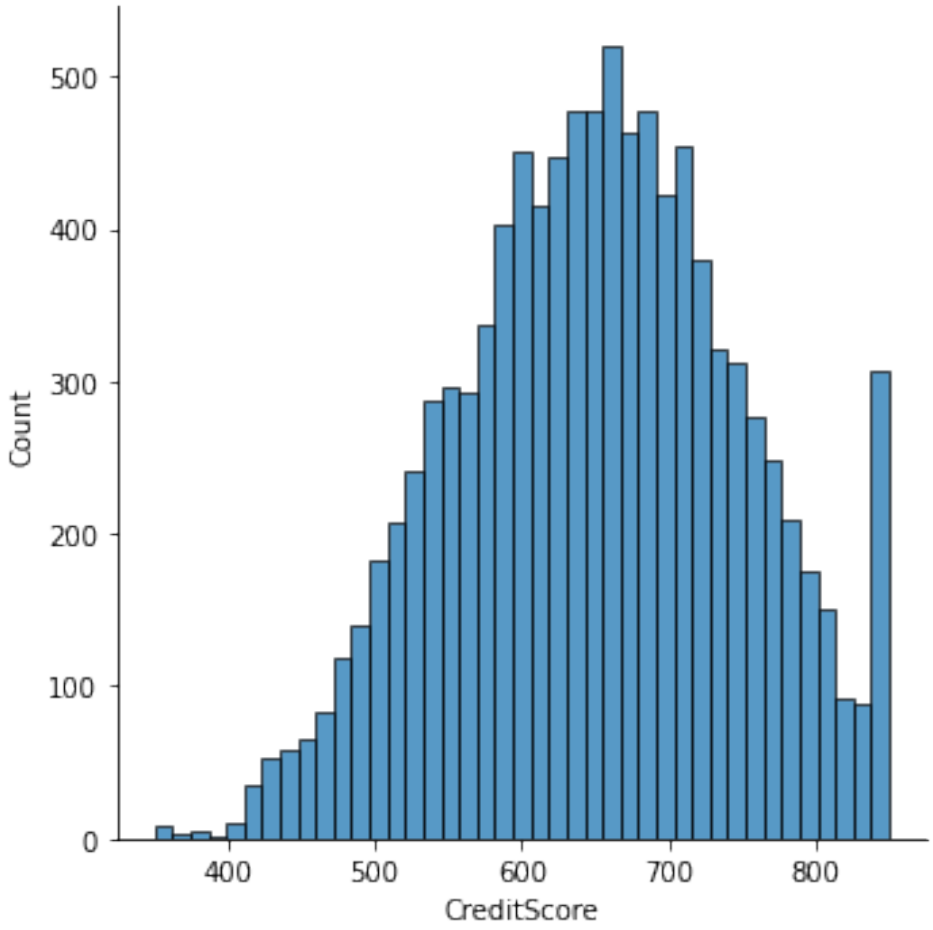
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an
axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4aeaa54ad0>
```




```
sns.displot(df['CreditScore'])
```

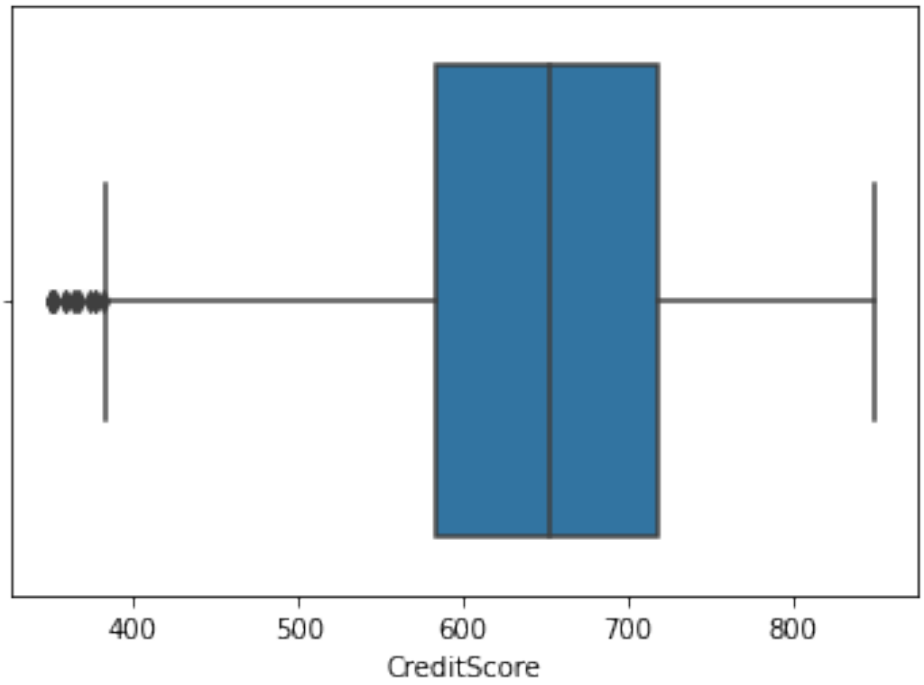
```
<seaborn.axisgrid.FacetGrid at 0x7f4aeaafde50>
```




```
#Boxplot
sns.boxplot(df['CreditScore'])
```

```
/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
```

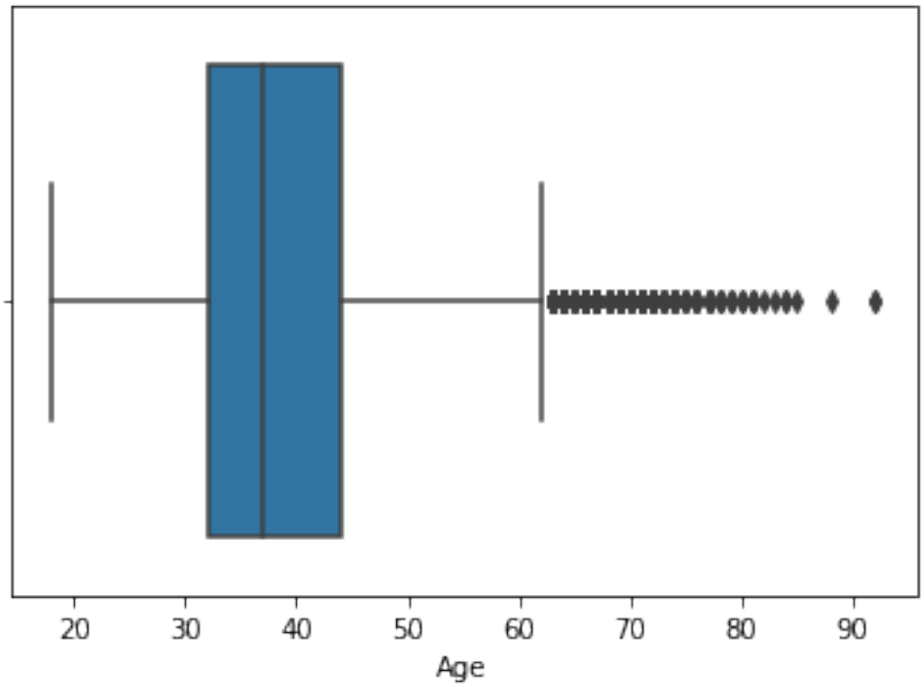
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4aea946790>
```




```
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
```

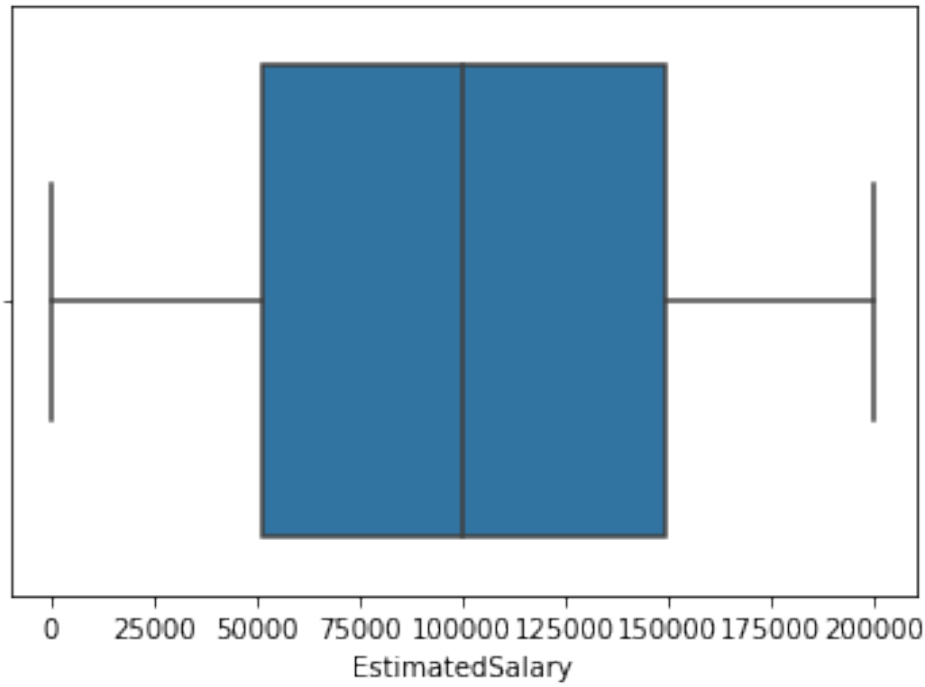
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4aeab69050>
```




```
sns.boxplot(df['EstimatedSalary'])
```

```
/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
```

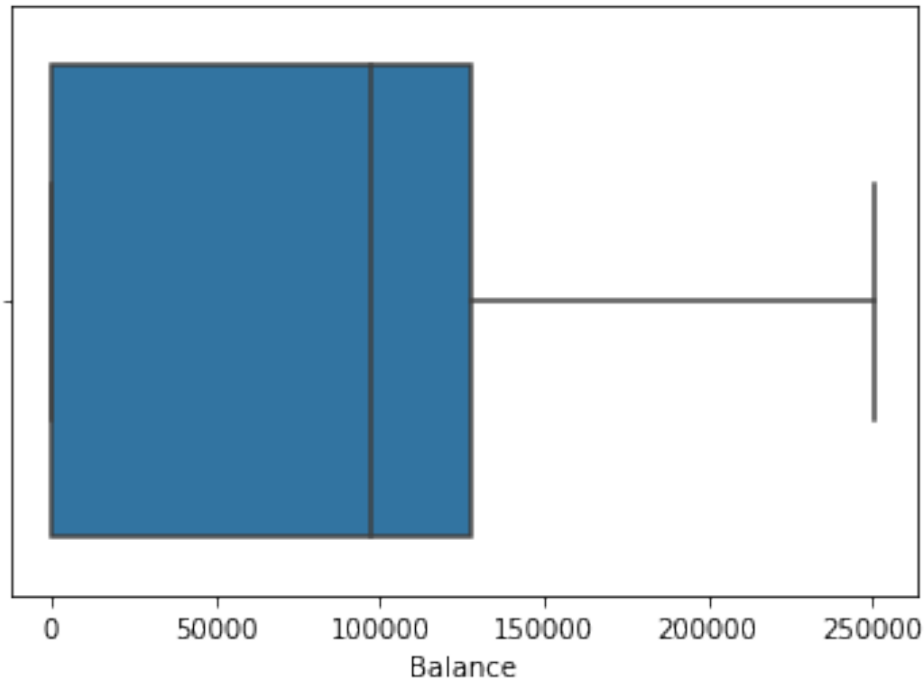
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4aeac5add0>
```




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

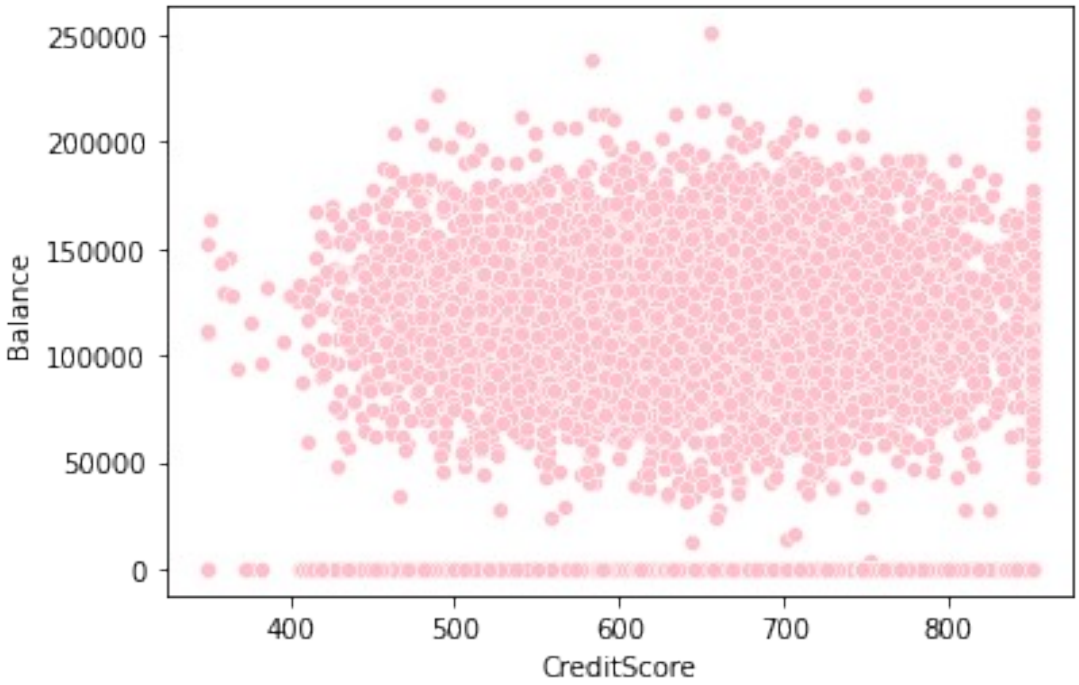
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be `data`, and
passing other arguments without an explicit keyword will result in an
error or misinterpretation.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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae7f1a090>
```




```
#Bivariate Analysis

#Scatterplot
sns.scatterplot(df['CreditScore'],df['Balance'],color='Pink')
```

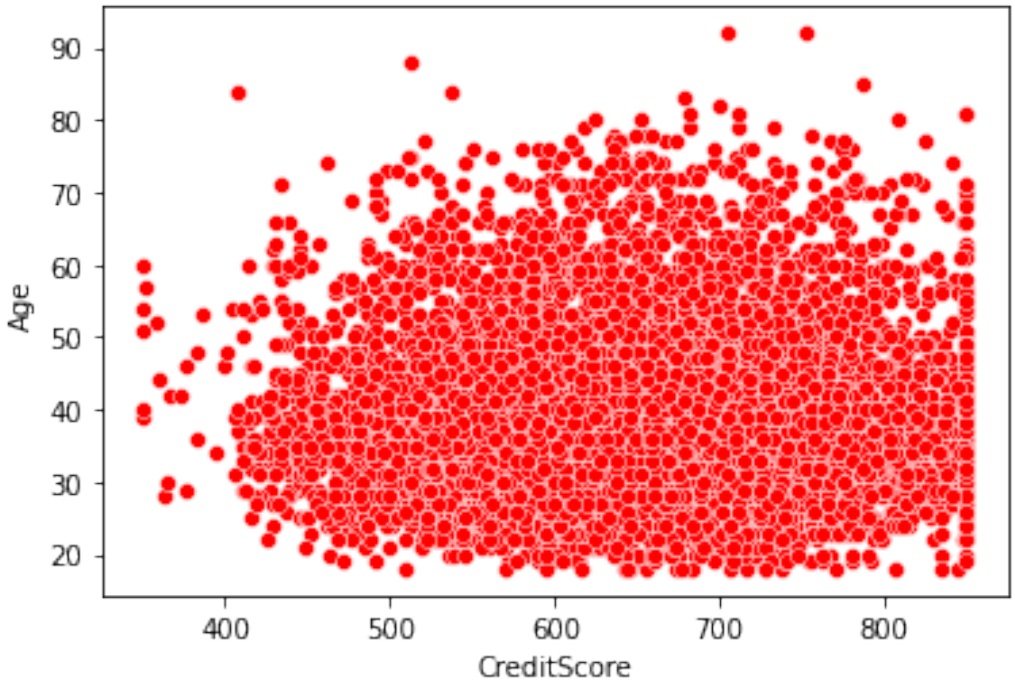
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.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

<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae7e7de90>
```




```
sns.scatterplot(df['CreditScore'],df['Age'],color='red')
```

```
/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
```

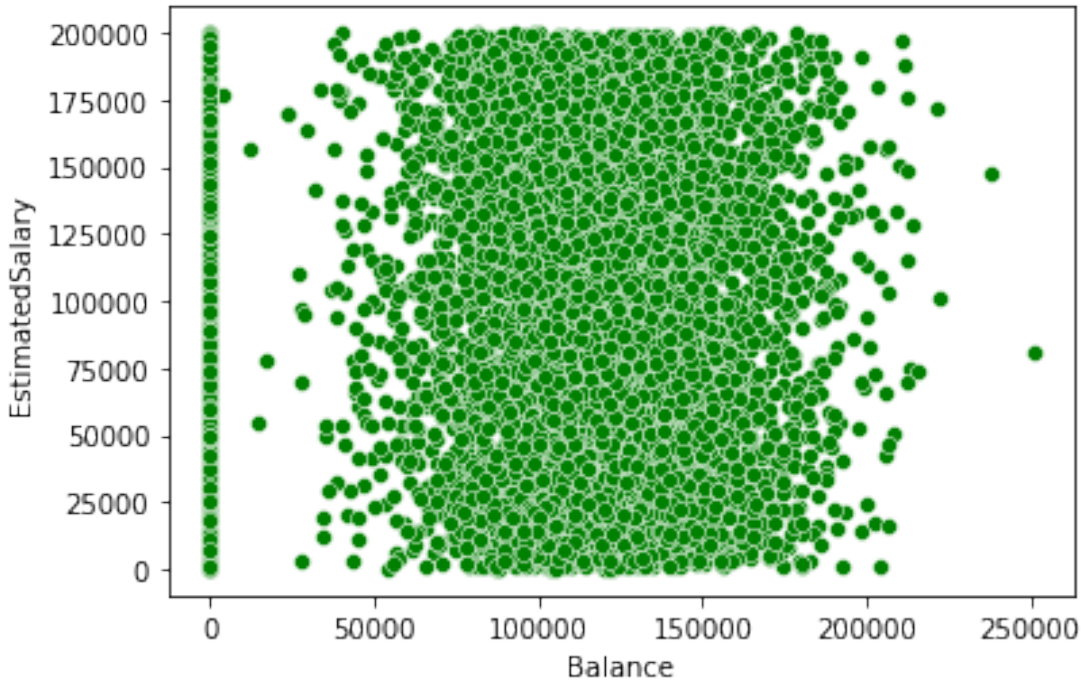
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae7dfe7d0>
```




```
sns.scatterplot(df['Balance'],df['EstimatedSalary'],color='green')
```

```
/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
```

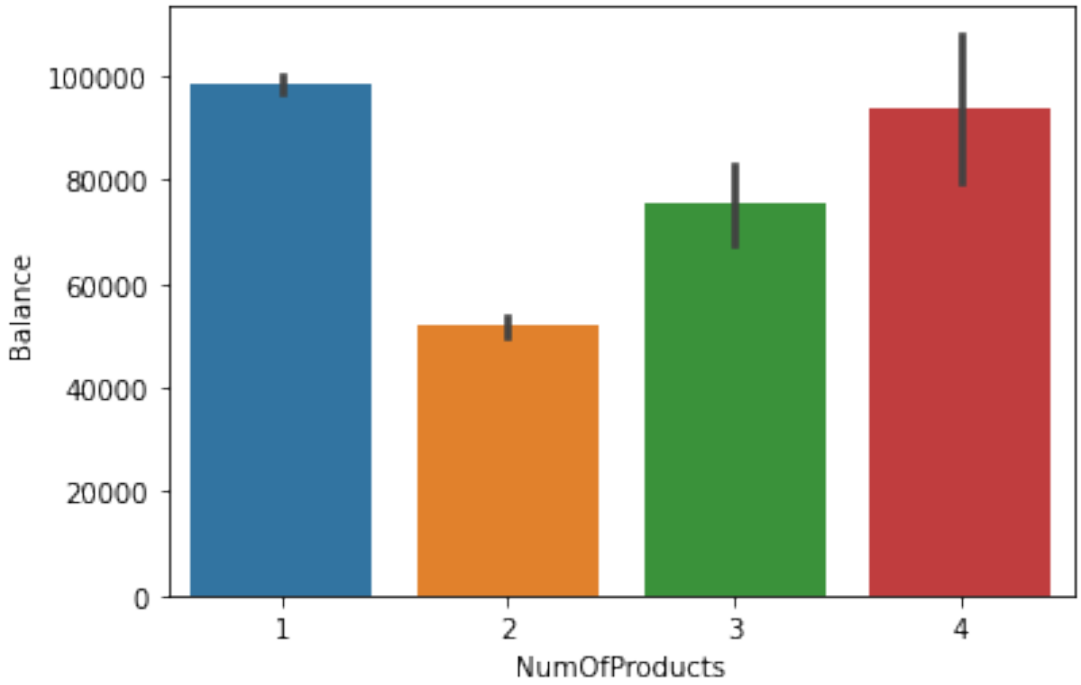
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae7deae10>
```




```
#Barplot
sns.barplot(df['NumOfProducts'],df['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
```

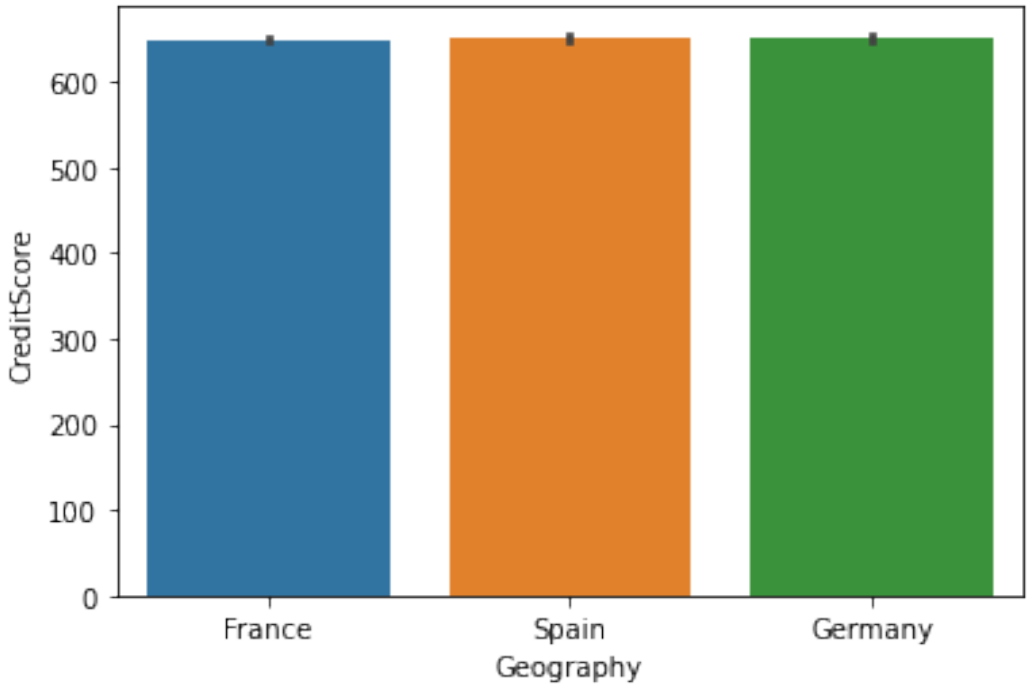
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae7cef090>
```




```
sns.barplot(df['Geography'],df['CreditScore'])
```

```
/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
```

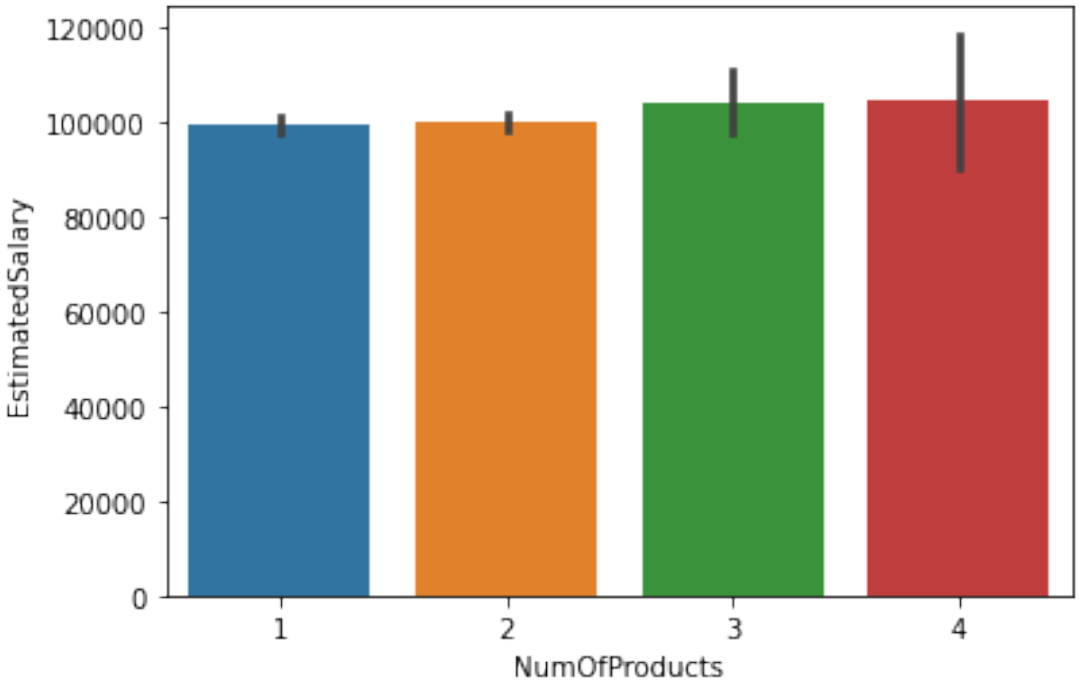
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae64e7250>
```




```
sns.barplot(df['NumOfProducts'],df['EstimatedSalary'])
```

```
/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
```

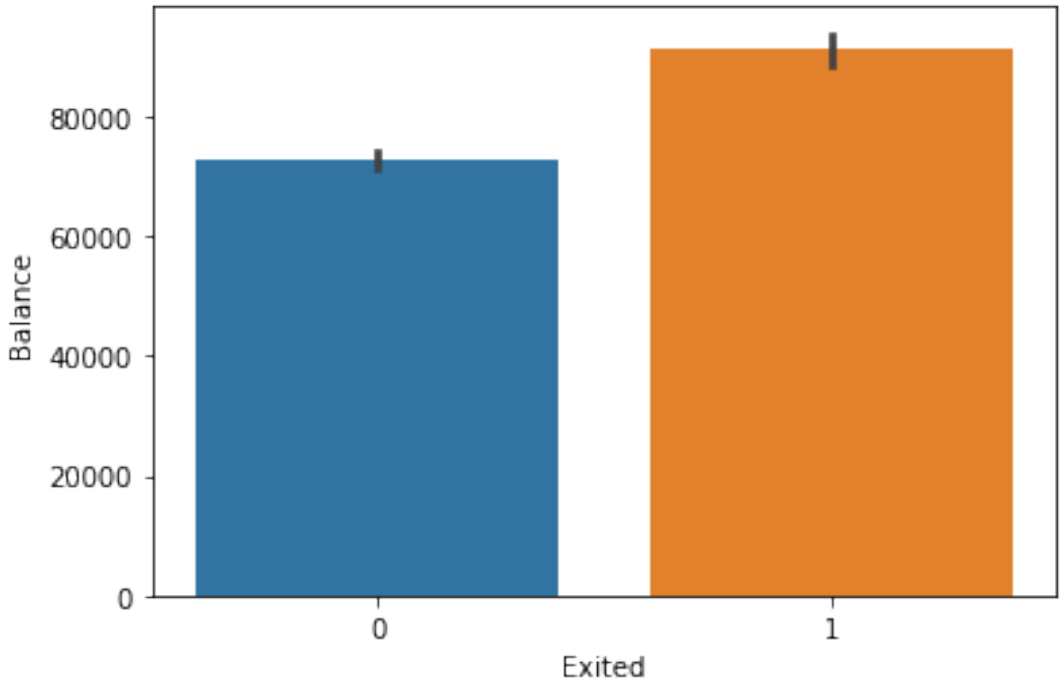
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae644e4d0>
```




```
sns.barplot(df['Exited'],df['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
```

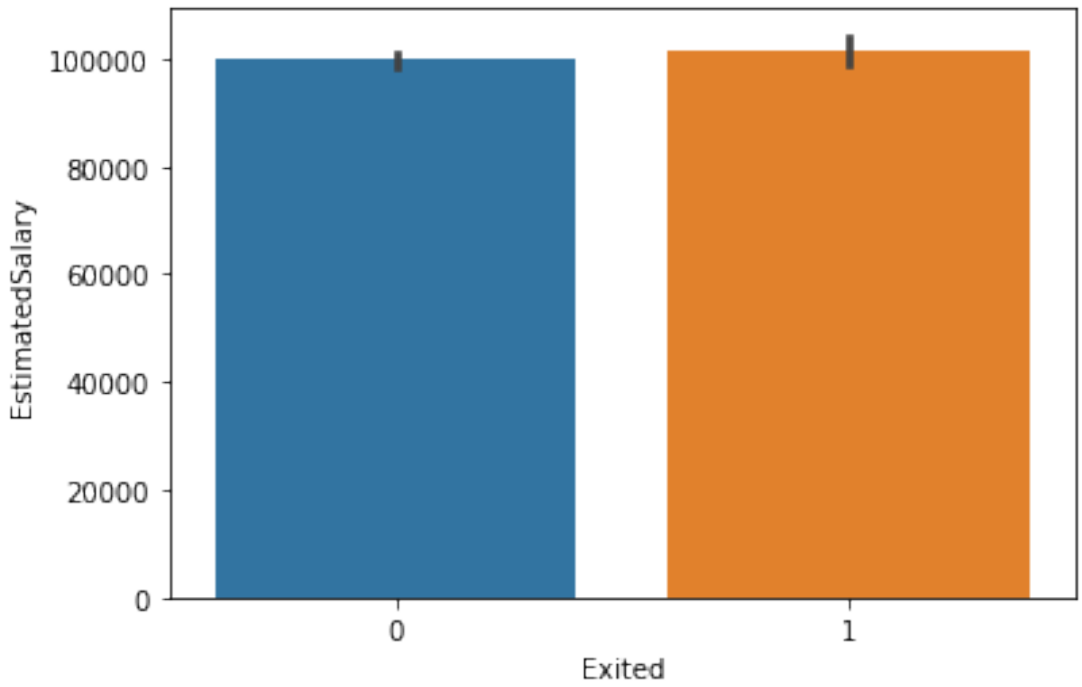
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae63c2ad0>
```




```
sns.barplot(df['Exited'],df['EstimatedSalary'])
```

```
/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
```

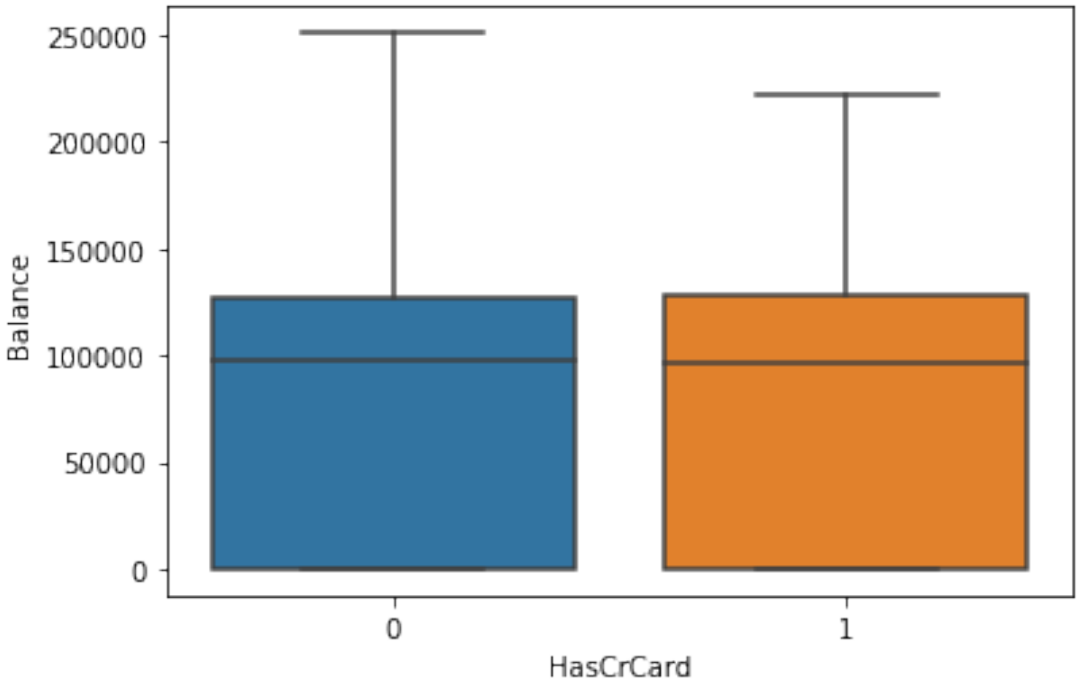
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae63a5610>
```




```
sns.boxplot(df['HasCrCard'],df['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
```

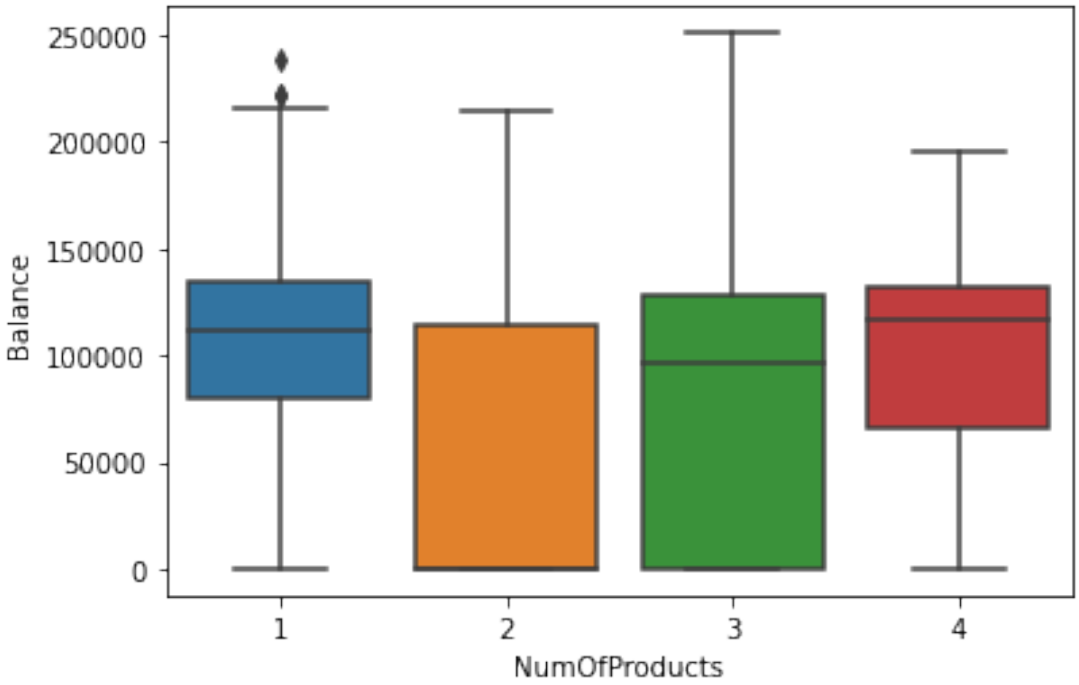
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae639ec10>
```




```
sns.boxplot(df['NumOfProducts'],df['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
```

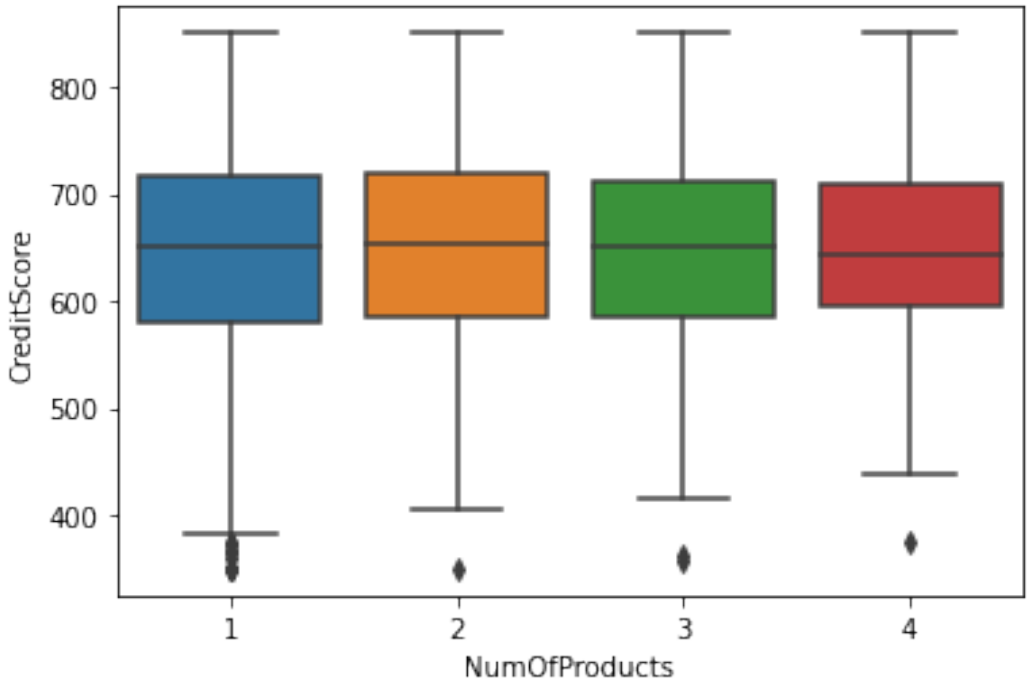
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae62f78d0>
```




```
sns.boxplot(df['NumOfProducts'],df['CreditScore'])
```

```
/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
```

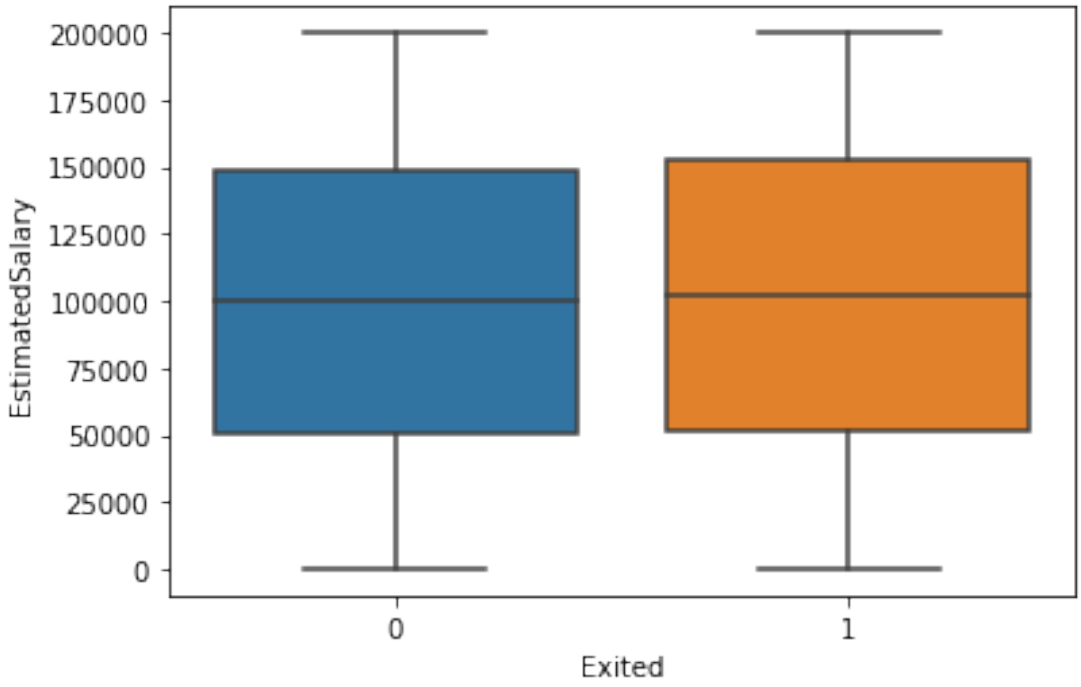
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae6286a10>
```




```
sns.boxplot(df['Exited'],df['EstimatedSalary'])
```

```
/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
```

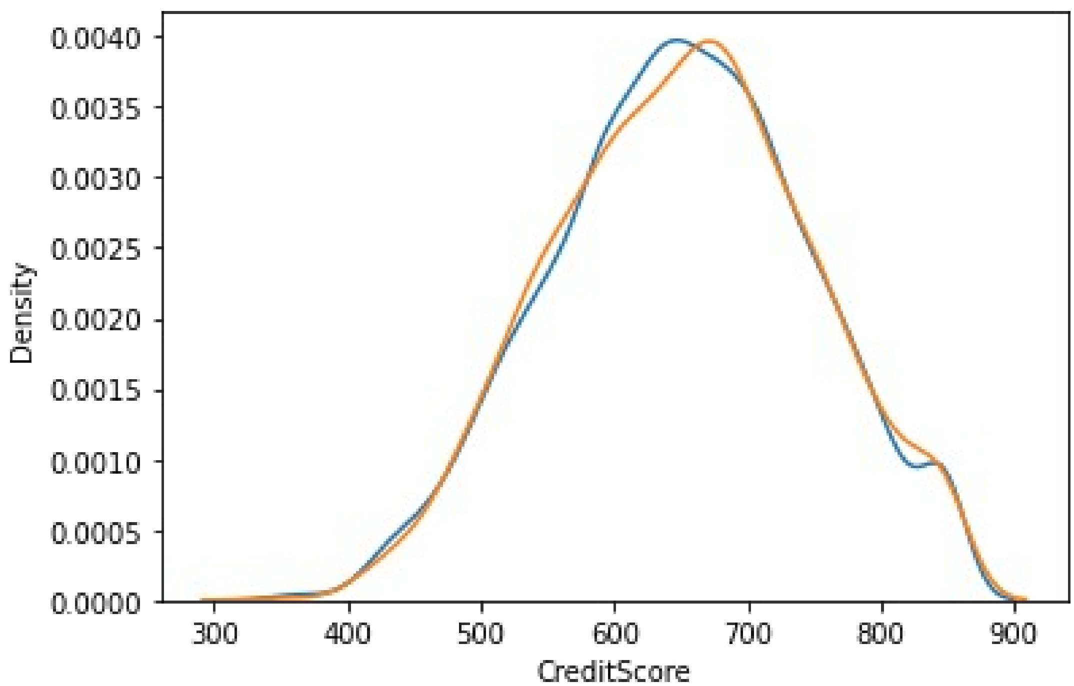
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae6156d90>
```




```
#Distplot
sns.distplot(df[df['HasCrCard']==1]['CreditScore'],hist=False)
sns.distplot(df[df['HasCrCard']==0]['CreditScore'],hist=False)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `kdeplot` (an axes-
level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `kdeplot` (an axes-
level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
```

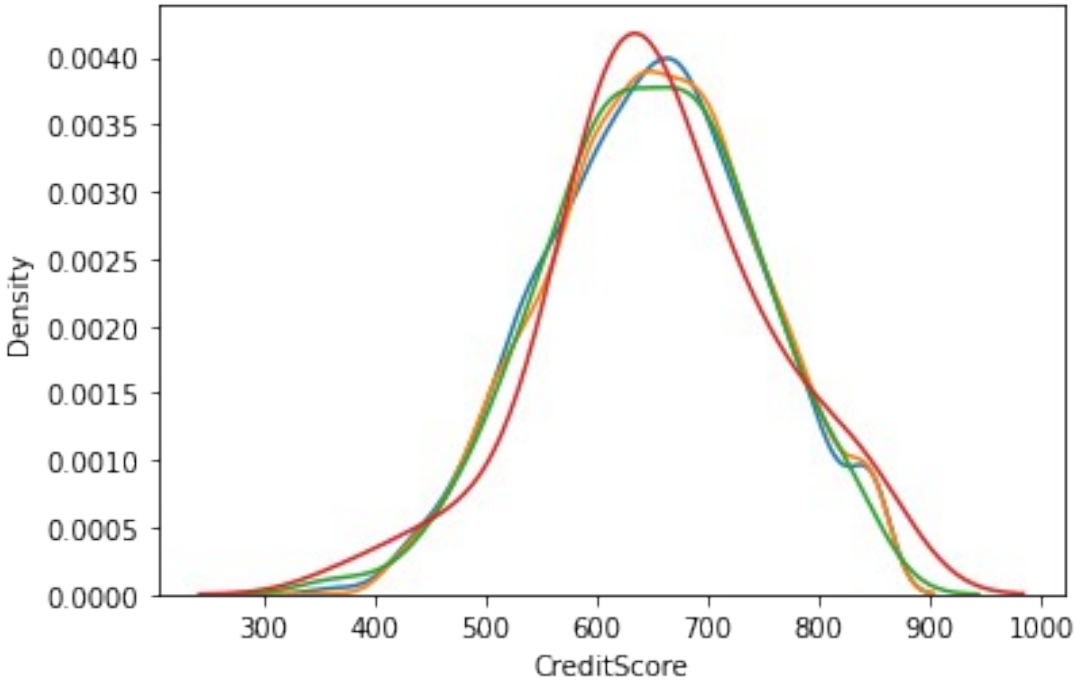
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae607a510>
```




```
sns.distplot(df[df['NumOfProducts']==1]['CreditScore'],hist=False)
sns.distplot(df[df['NumOfProducts']==2]['CreditScore'],hist=False)
sns.distplot(df[df['NumOfProducts']==3]['CreditScore'],hist=False)
sns.distplot(df[df['NumOfProducts']==4]['CreditScore'],hist=False)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `kdeplot` (an axes-
level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `kdeplot` (an axes-
level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `kdeplot` (an axes-
level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `kdeplot` (an axes-
level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
```

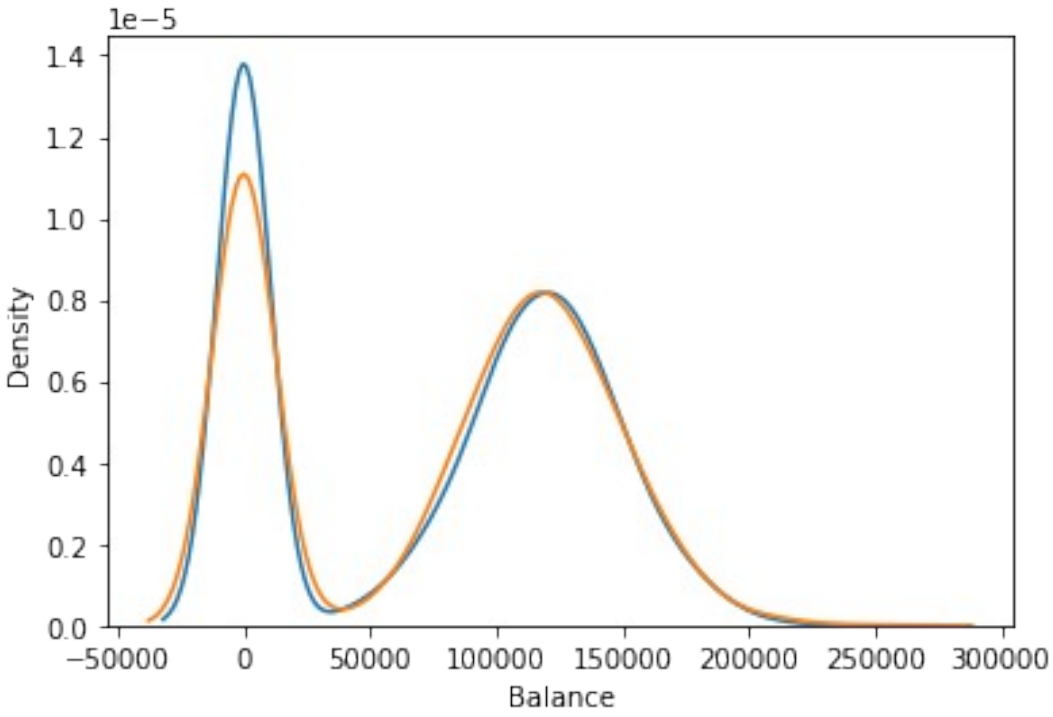
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae7f22e10>
```




```
sns.distplot(df[df['HasCrCard']==1]['Balance'],hist=False)
sns.distplot(df[df['HasCrCard']==0]['Balance'],hist=False)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `kdeplot` (an axes-
level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `kdeplot` (an axes-
level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
```

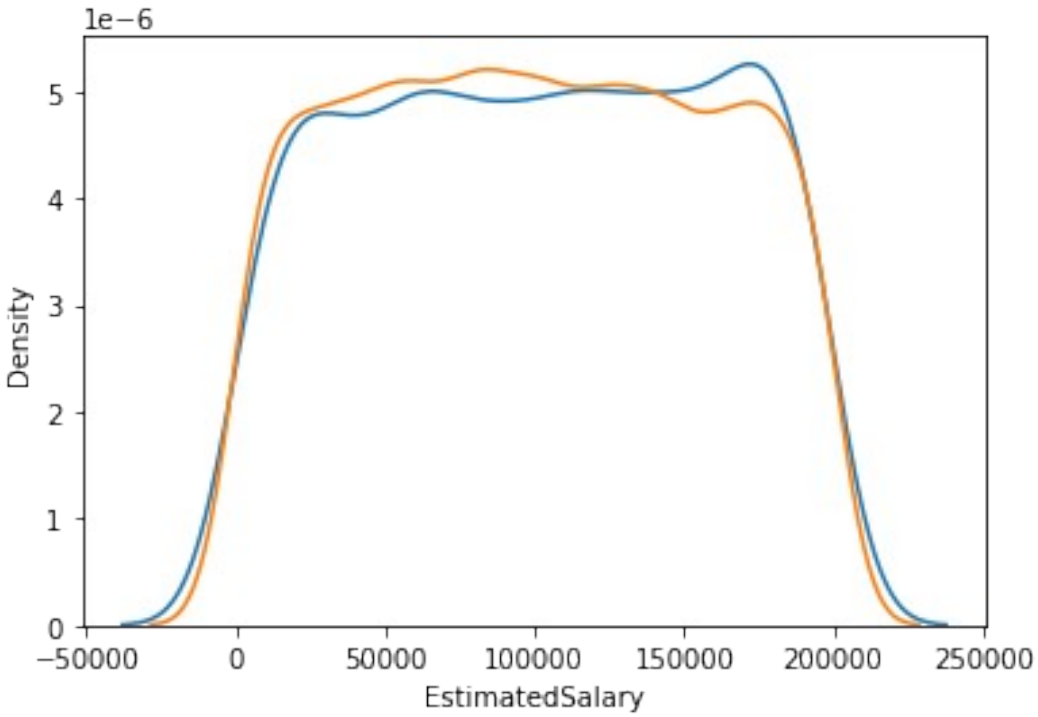
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae5fd73d0>
```




```
sns.distplot(df[df['Exited']==1]['EstimatedSalary'],hist=False)
sns.distplot(df[df['Exited']==0]['EstimatedSalary'],hist=False)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `kdeplot` (an axes-
level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `kdeplot` (an axes-
level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
```

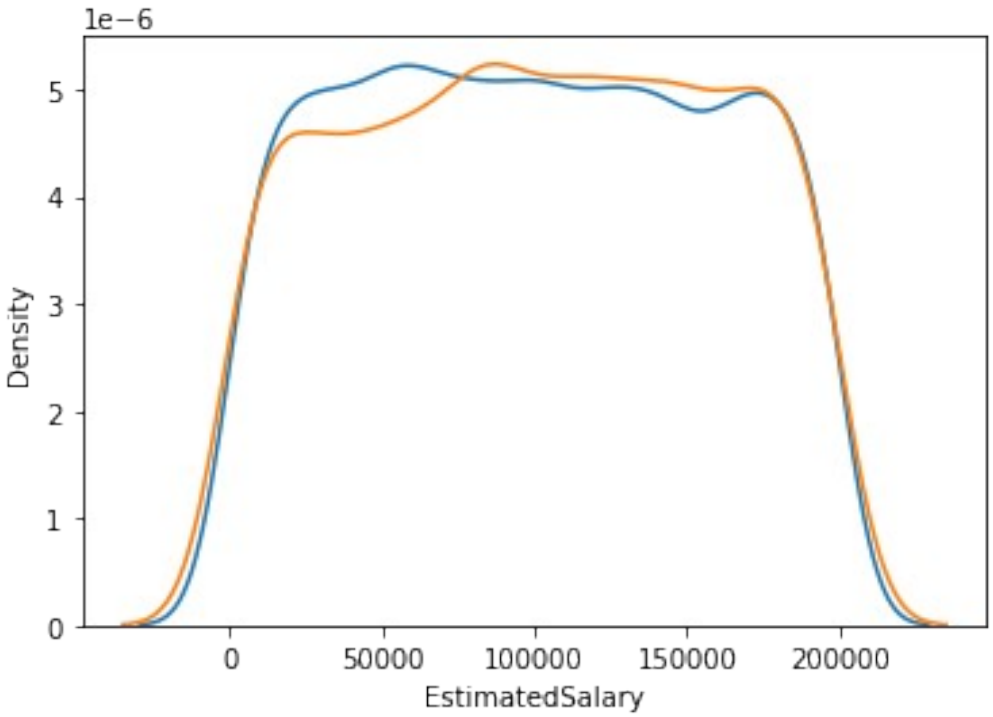
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae6486cd0>
```




```
sns.distplot(df[df['HasCrCard']==1]['EstimatedSalary'],hist=False)
sns.distplot(df[df['HasCrCard']==0]['EstimatedSalary'],hist=False)
```

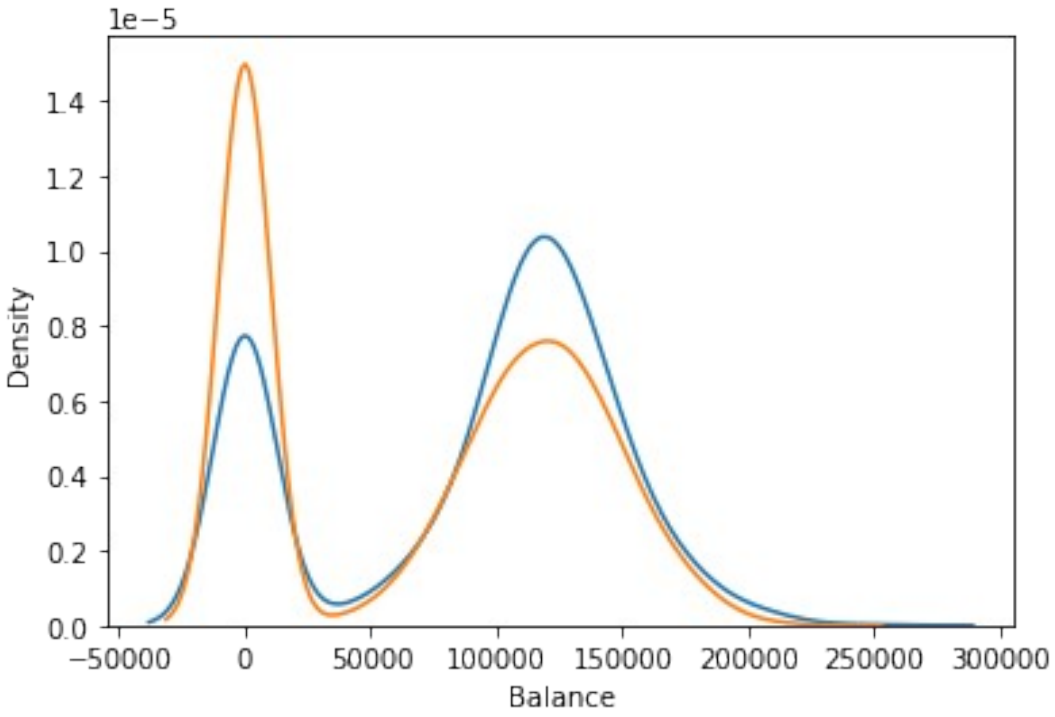
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `kdeplot` (an axes-
level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `kdeplot` (an axes-
level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae5f02850>
```




```
sns.distplot(df[df['Exited']==1]['Balance'],hist=False)
sns.distplot(df[df['Exited']==0]['Balance'],hist=False)
```

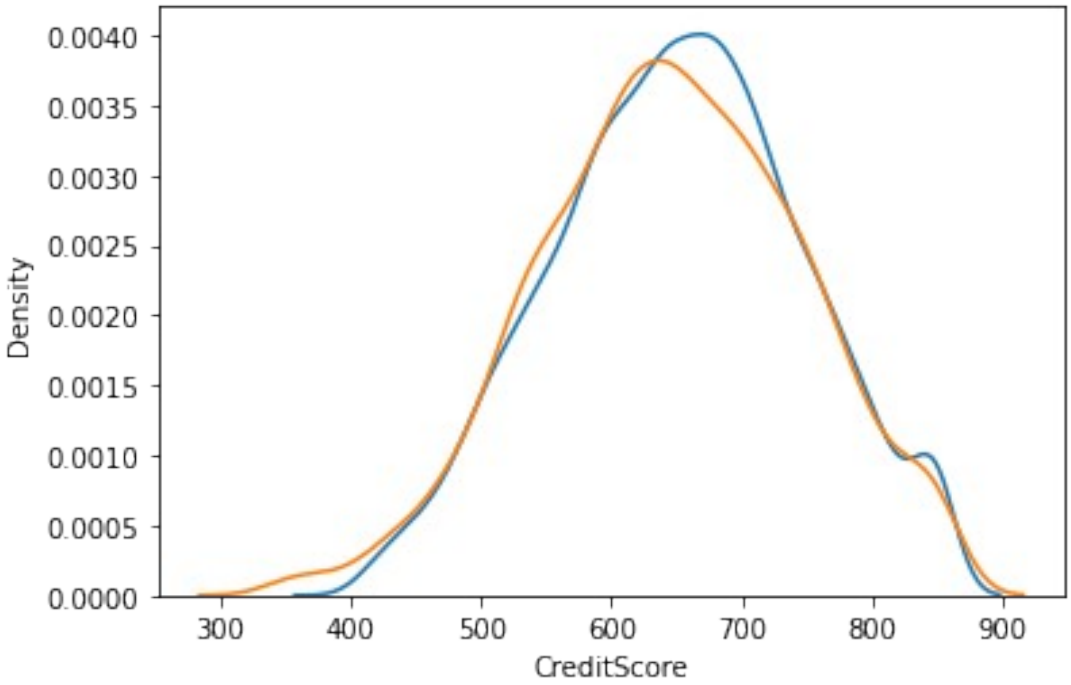
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `kdeplot` (an axes-
level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `kdeplot` (an axes-
level function for kernel density plots).
  warnings.warn(msg, FutureWarning)

<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae5eac390>
```




```
sns.distplot(df[df['Exited']==0]['CreditScore'],hist=False)
sns.distplot(df[df['Exited']==1]['CreditScore'],hist=False)
```

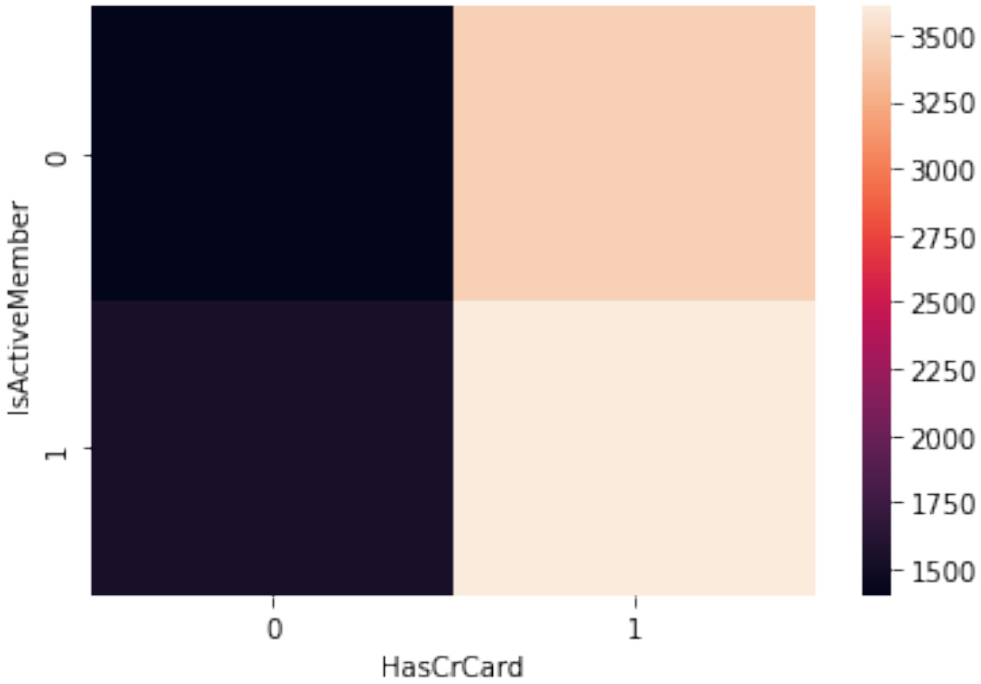
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `kdeplot` (an axes-
level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `kdeplot` (an axes-
level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae5e01710>
```



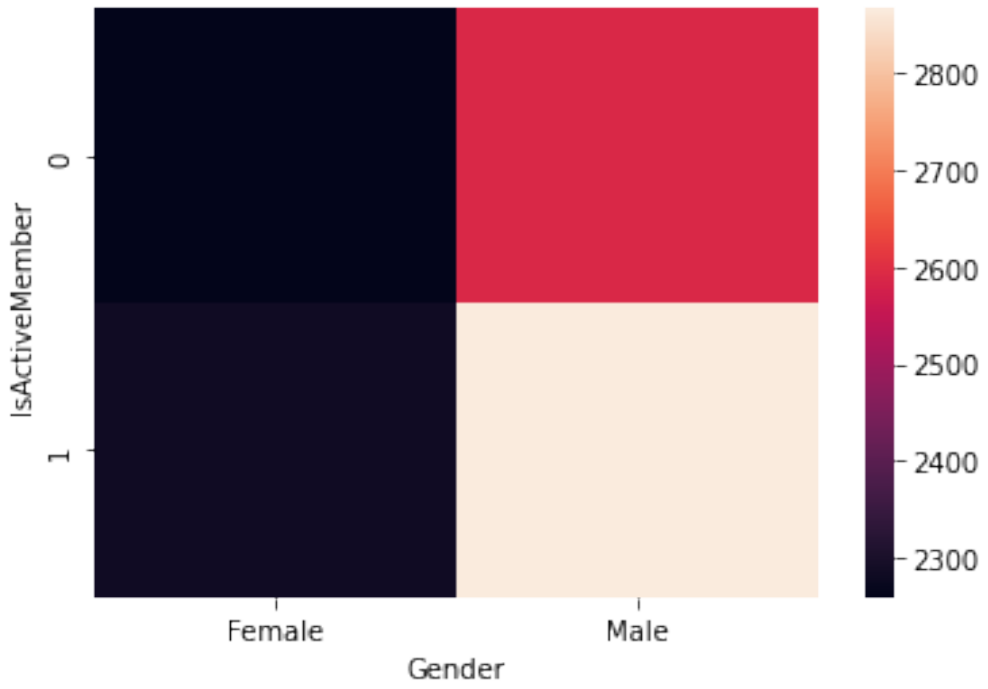

```
sns.heatmap(pd.crosstab(df['IsActiveMember'],df['HasCrCard']))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae5dde350>
```



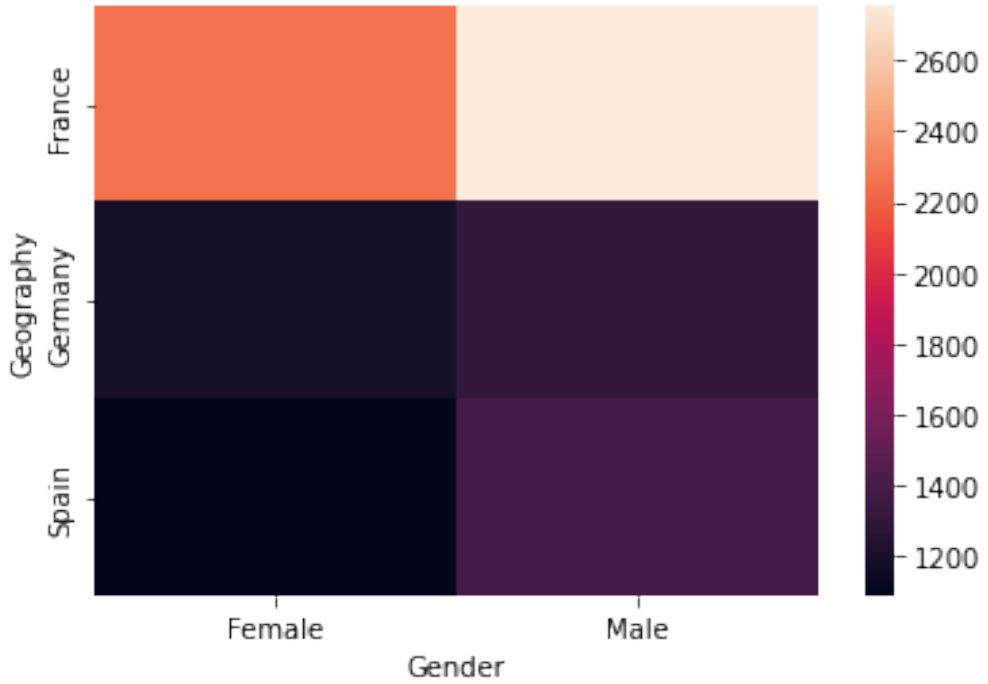

```
sns.heatmap(pd.crosstab(df['IsActiveMember'],df['Gender']))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae5cd8590>
```



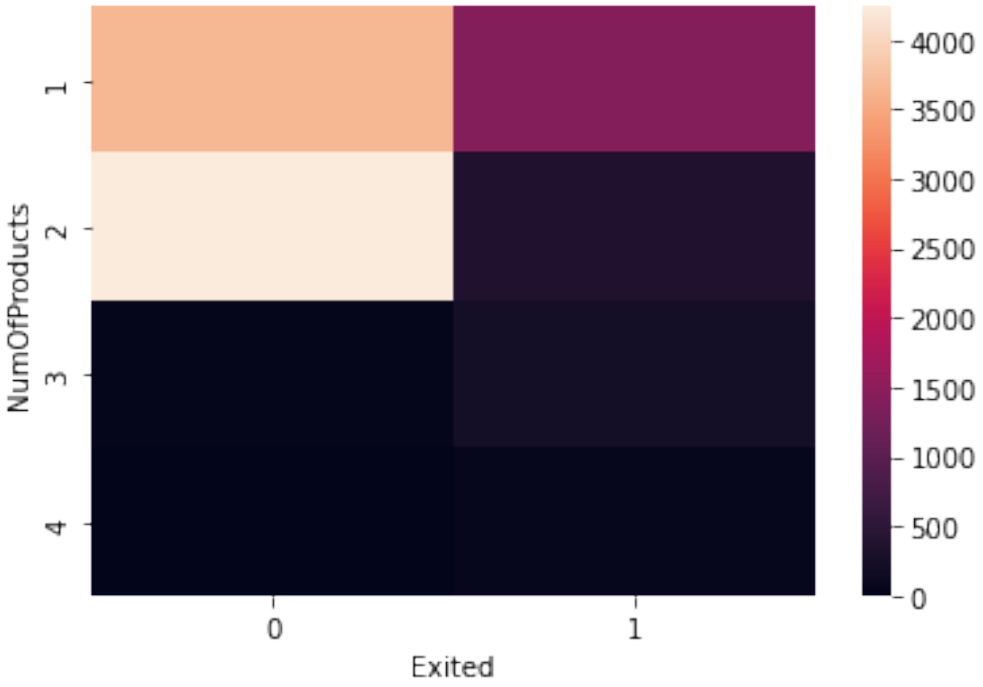

```
sns.heatmap(pd.crosstab(df['Geography'],df['Gender']))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae5bfe4d0>
```




```
sns.heatmap(pd.crosstab(df['NumOfProducts'],df['Exited']))
```

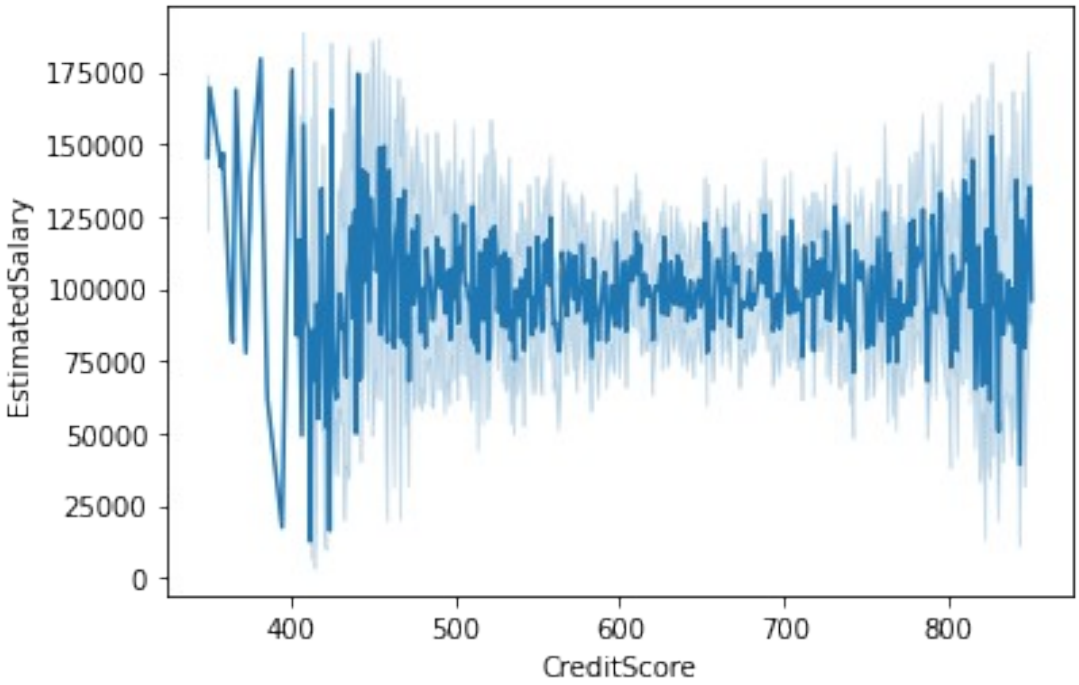
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae5ba0210>
```




```
#lineplot
sns.lineplot(df['CreditScore'],df['EstimatedSalary'])
```

```
/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
```

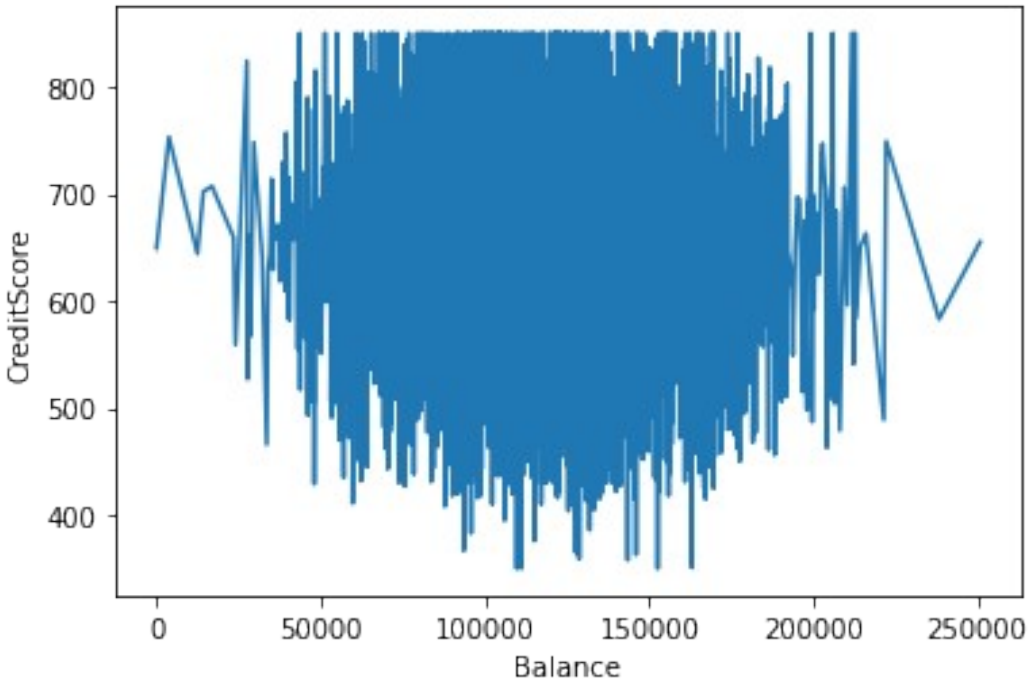
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae5abfd90>
```




```
sns.lineplot(df['Balance'],df['CreditScore'])
```

```
/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
```

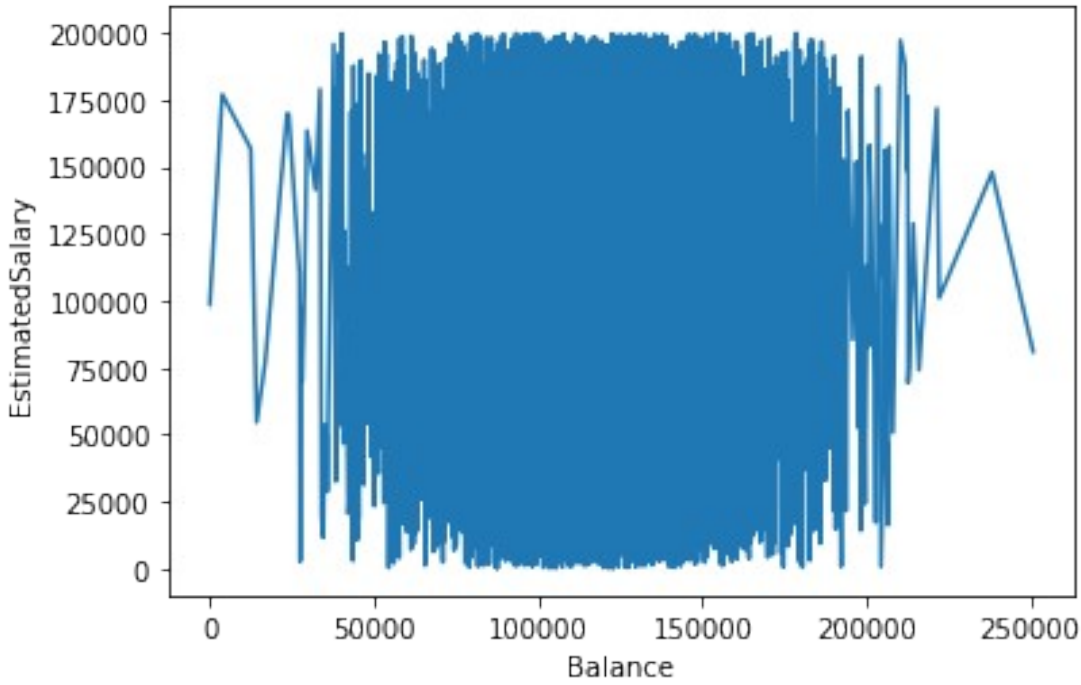
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae5945c90>
```




```
sns.lineplot(df['Balance'],df['EstimatedSalary'])
```

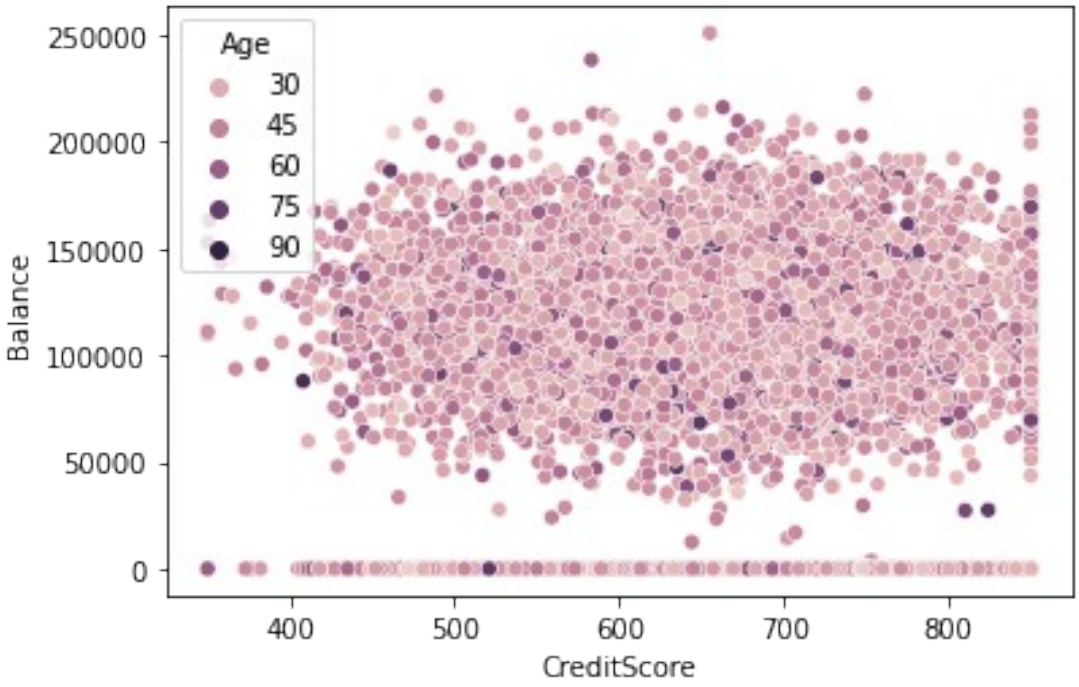
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae5a42650>
```




```
#Multivariate Analysis

sns.scatterplot(df['CreditScore'],df['Balance'],hue=df['Age'],color='P
ink')
```

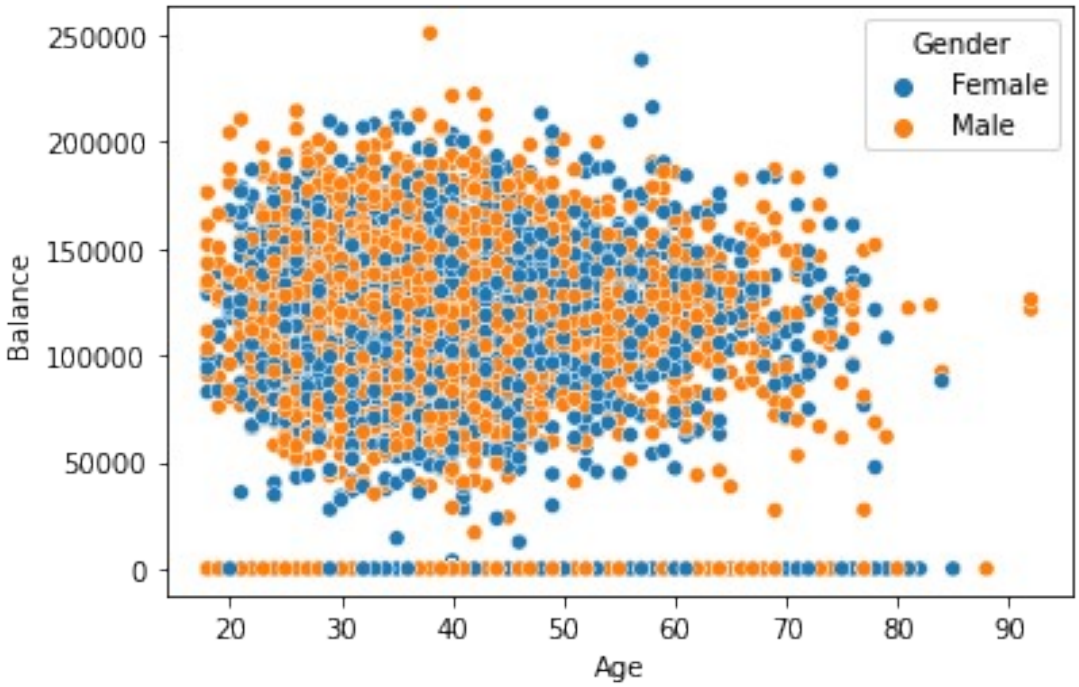
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.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

<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae59e0590>
```




```
sns.scatterplot(df['Age'],df['Balance'],hue=df['Gender'],color='Pink')
```

```
/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
```

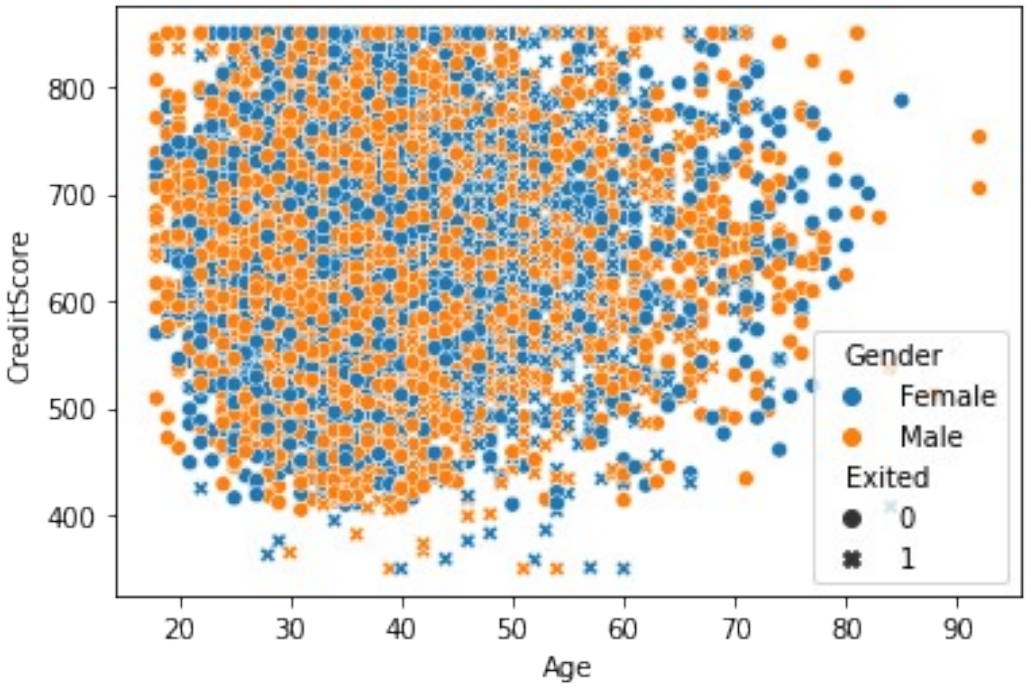
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae58e2f90>
```




```
sns.scatterplot(df['Age'],df['CreditScore'],hue=df['Gender'],style=df[
'Exited'])
```

```
/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
```

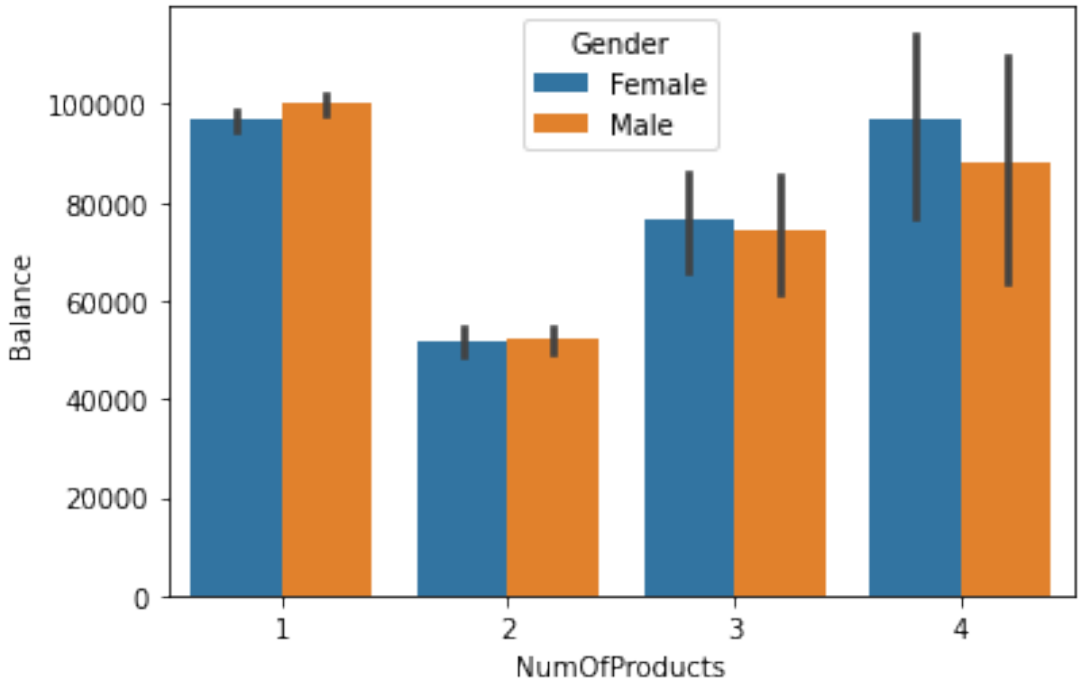
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae58dc390>
```




```
#Barplot
sns.barplot(df['NumOfProducts'],df['Balance'],hue=df['Gender'])
```

```
/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
```

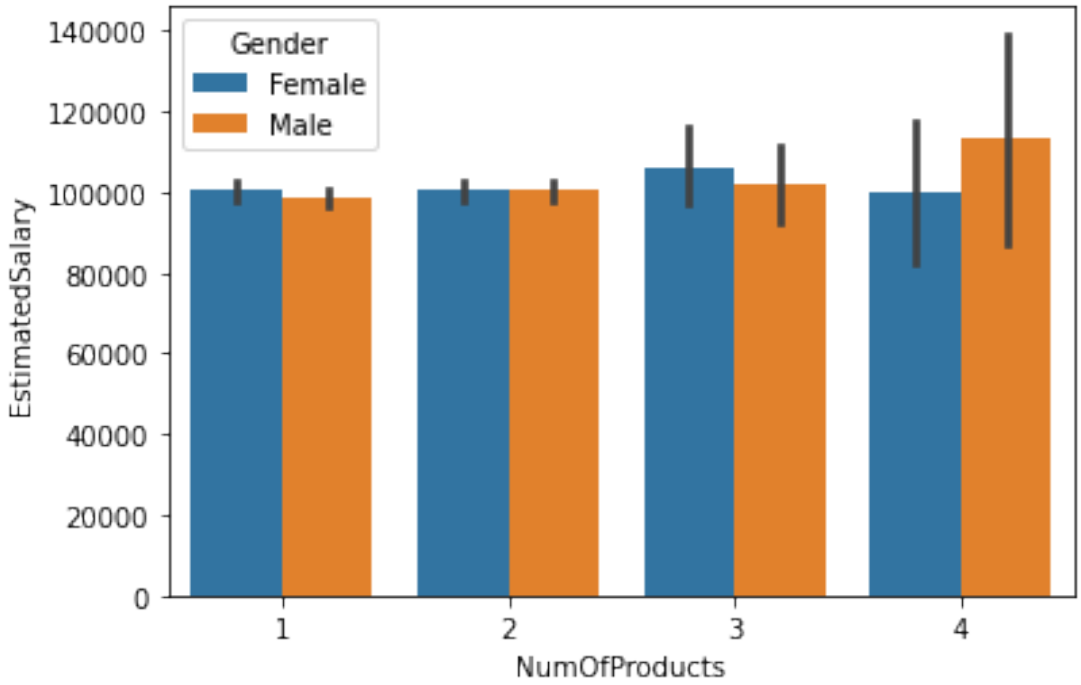
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae5ea53d0>
```




```
sns.barplot(df['NumOfProducts'],df['EstimatedSalary'],hue=df['Gender']
)
```

```
/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
```

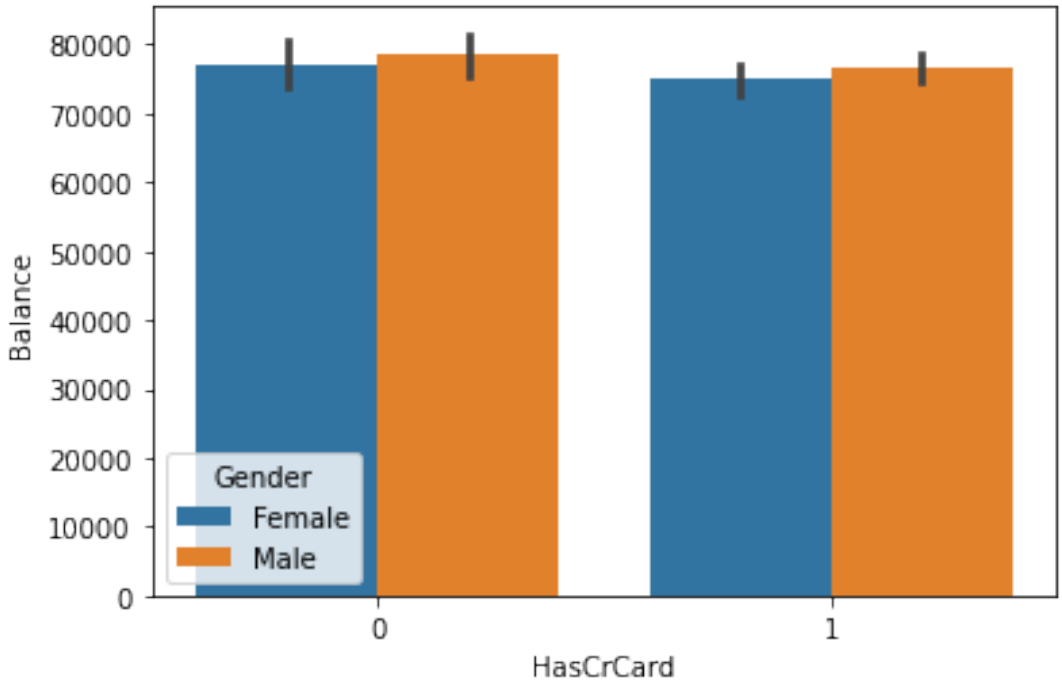
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae5d9d110>
```




```
sns.barplot(df['HasCrCard'],df['Balance'],hue=df['Gender'])
```

```
/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
```

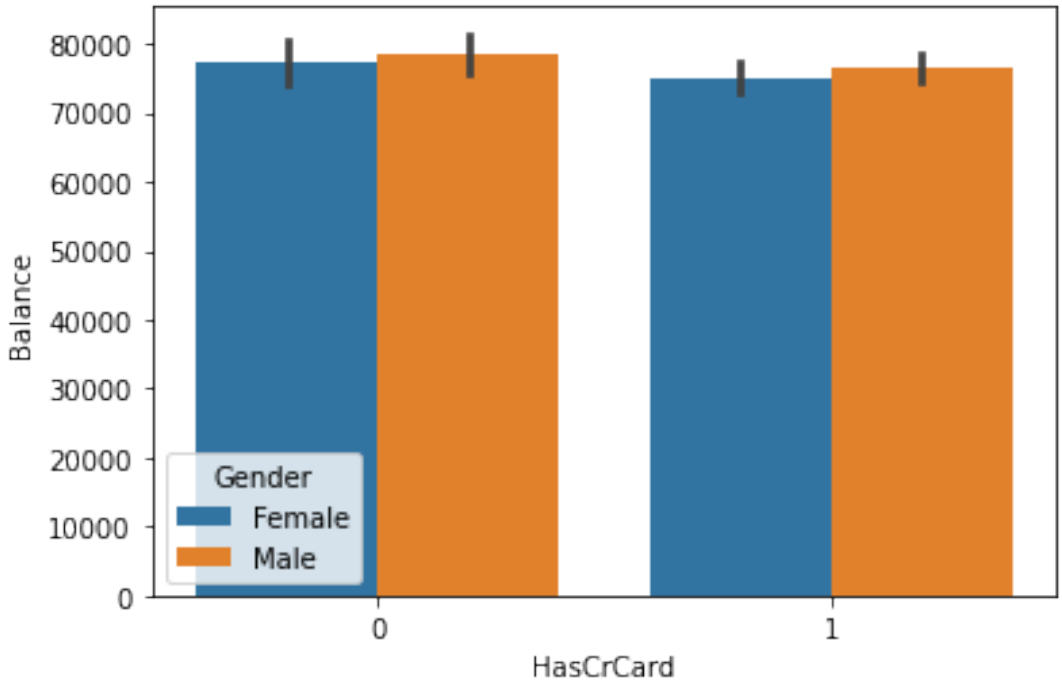
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae57227d0>
```




```
sns.barplot(df['HasCrCard'],df['Balance'],hue=df['Gender'])
```

```
/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
```

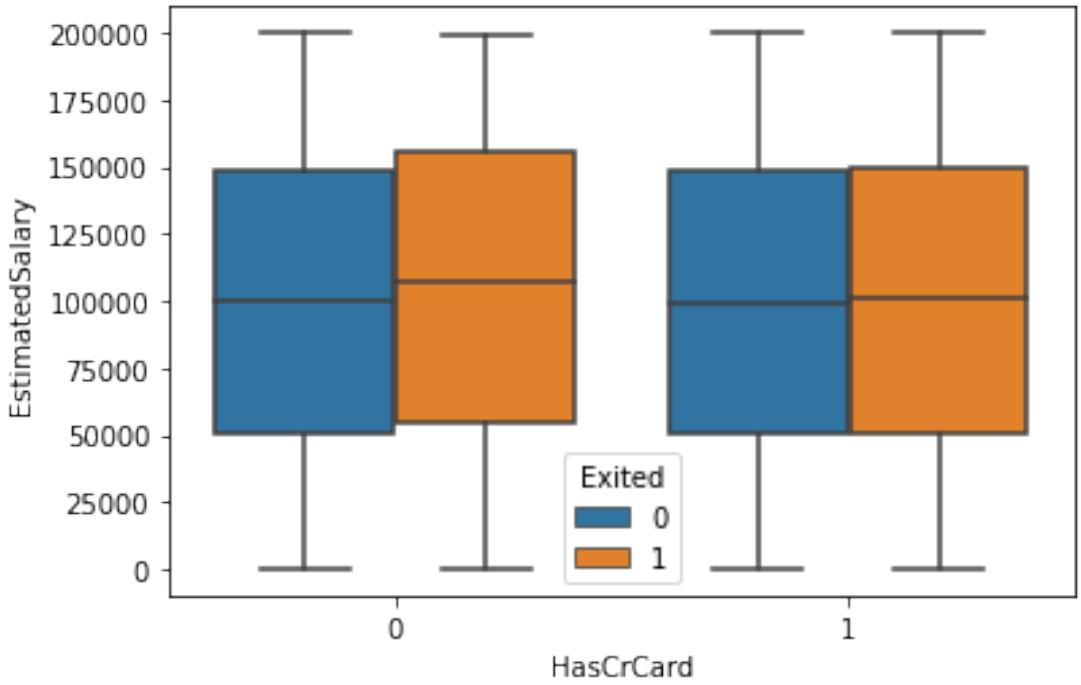
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae572a0d0>
```




```
#Boxplot
sns.boxplot(df['HasCrCard'],df['EstimatedSalary'],hue=df['Exited'])
```

```
/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
```

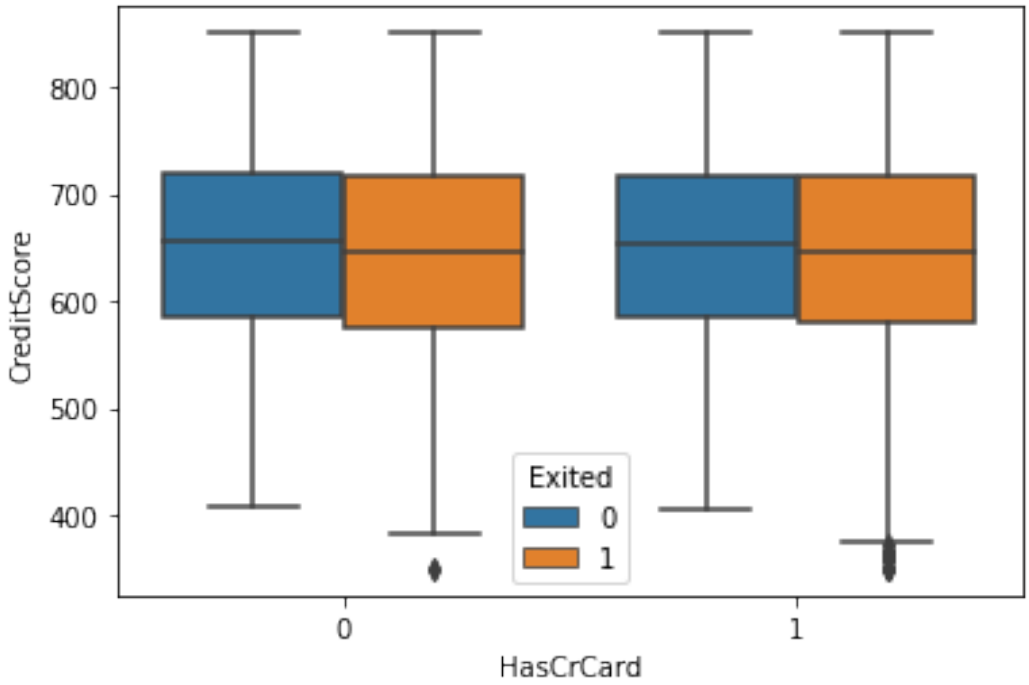
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae55ef790>
```




```
sns.boxplot(df['HasCrCard'],df['CreditScore'],hue=df['Exited'])
```

```
/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
```

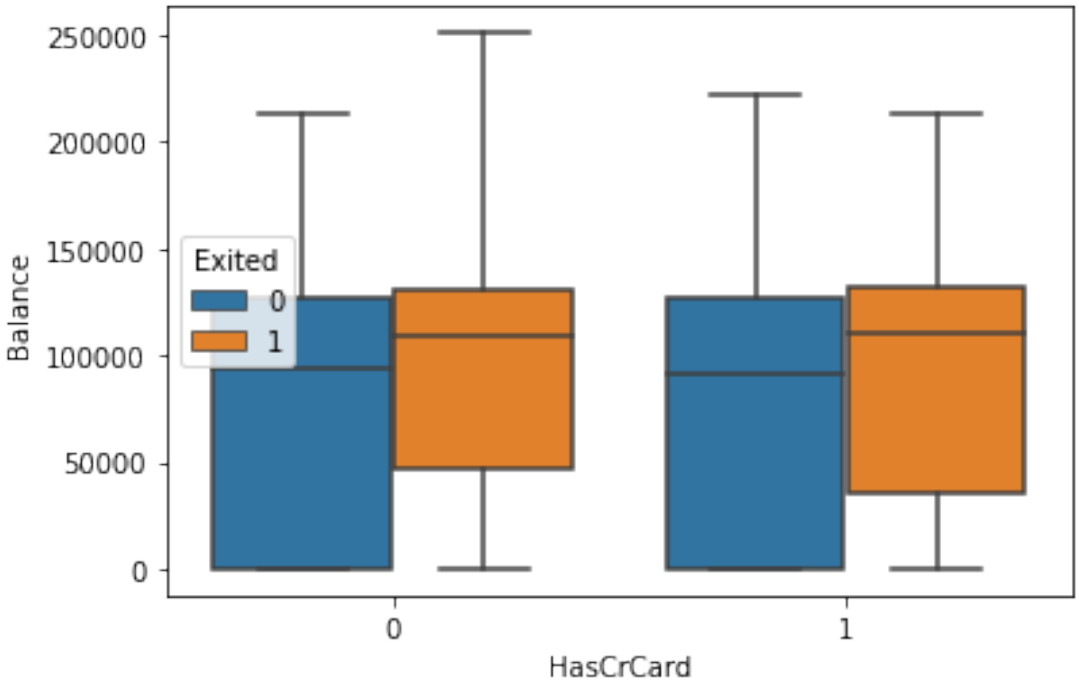
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae556c610>
```




```
sns.boxplot(df['HasCrCard'],df['Balance'],hue=df['Exited'])
```

```
/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
```

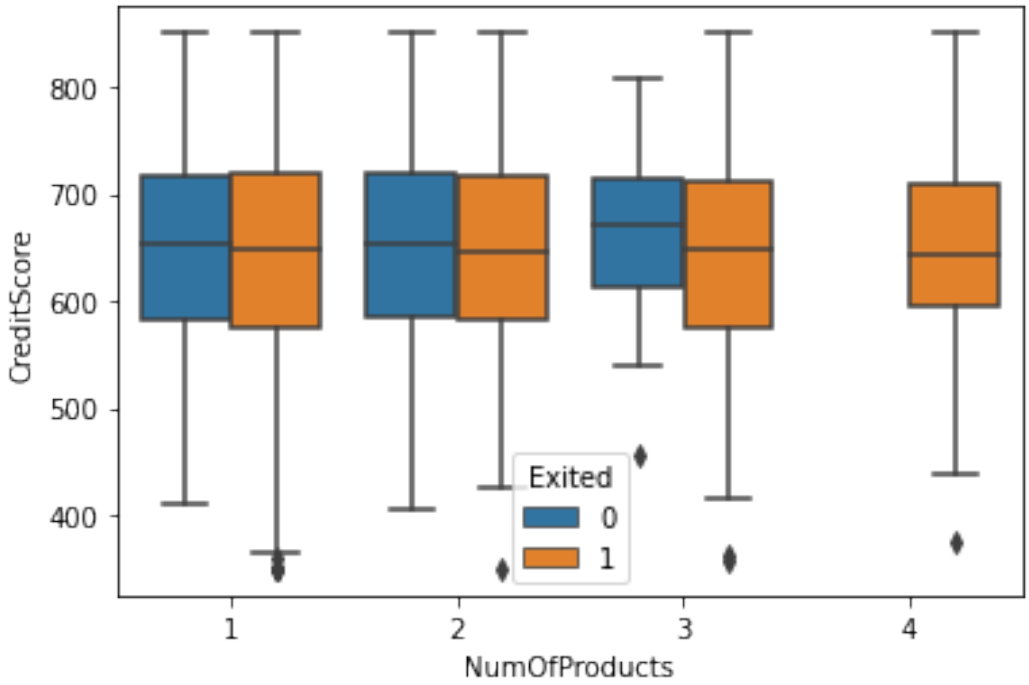
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae5405610>
```




```
sns.boxplot(df['NumOfProducts'],df['CreditScore'],hue=df['Exited'])
```

```
/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
```

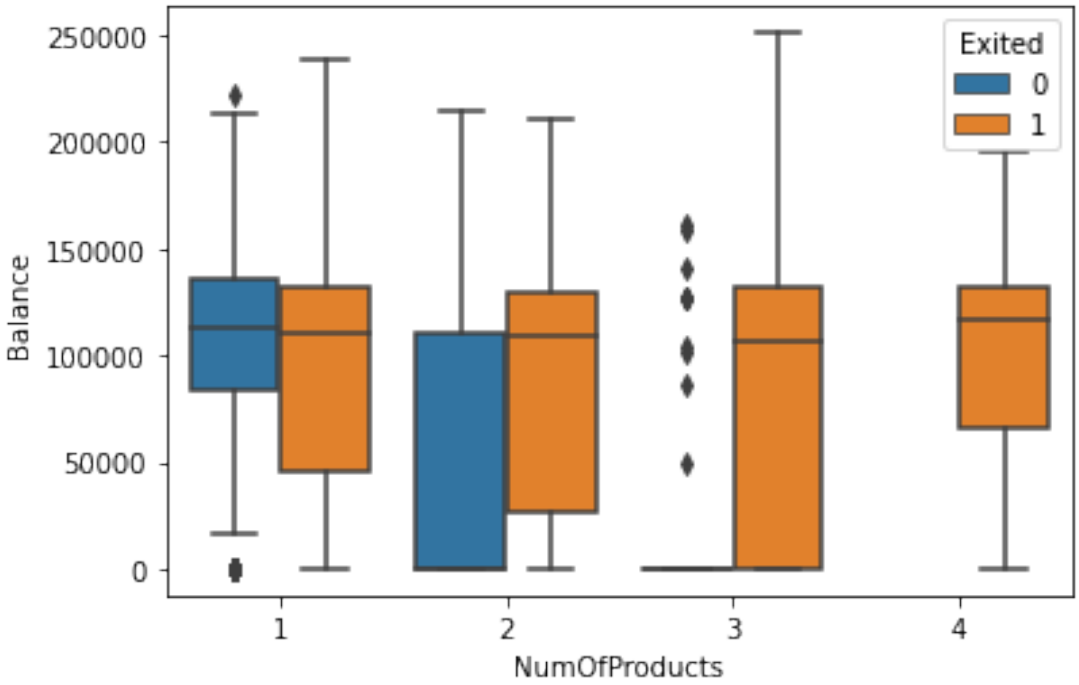
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae5333910>
```




```
sns.boxplot(df['NumOfProducts'],df['Balance'],hue=df['Exited'])
```

```
/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
```

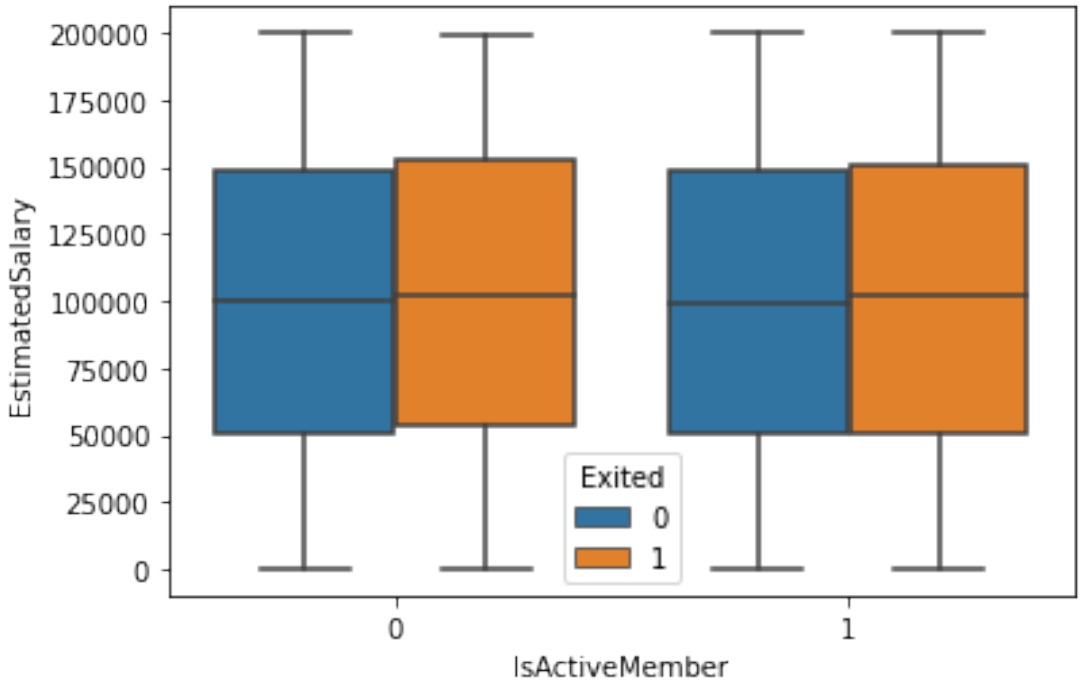
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae5243450>
```




```
sns.boxplot(df['IsActiveMember'],df['EstimatedSalary'],hue=df['Exited'
])
```

```
/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
```

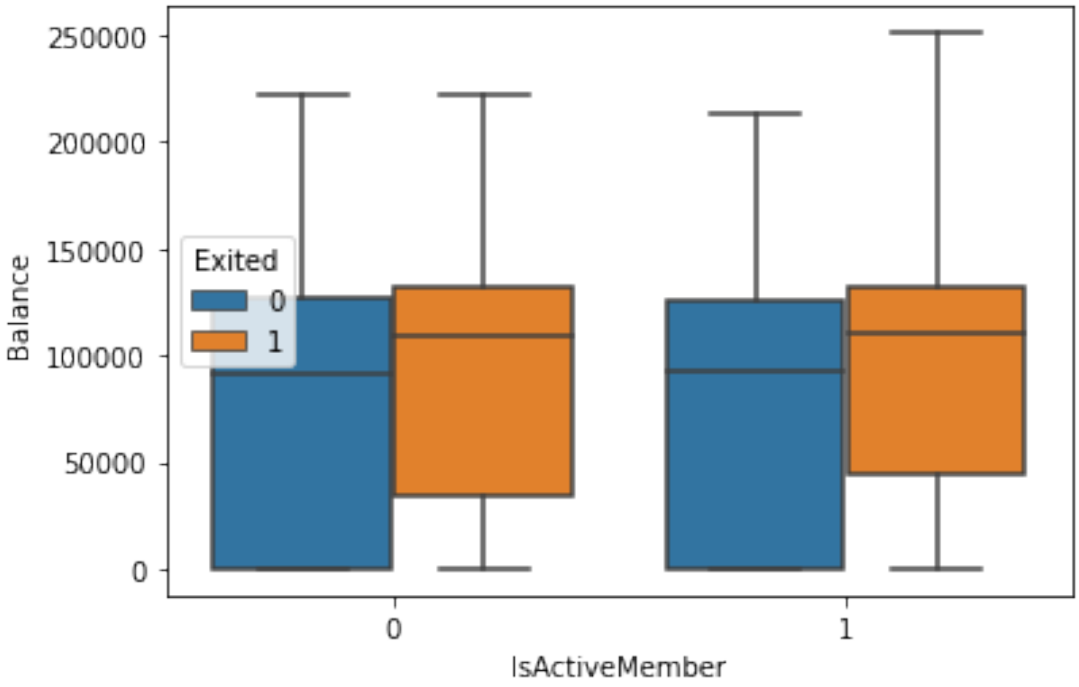
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae5137810>
```




```
sns.boxplot(df['IsActiveMember'],df['Balance'],hue=df['Exited'])
```

```
/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 0x7f4ae5076310>
```




```
#Descriptive Statistics
df.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
```

```
df.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
```

```
df.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]

df.kurt

<bound method NDFrame._add_numeric_operations.<locals>.kurt of
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.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
```

```
df.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

df.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

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
```

```
#Handle Missing Value
df.isna()
```

```
      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.isna().any()

RowNumber          False
CustomerId         False
Surname            False
CreditScore        False
Geography          False
Gender             False
Age                False
Tenure             False
Balance            False
NumOfProducts      False
```

```
HasCrCard          False
IsActiveMember     False
EstimatedSalary    False
Exited             False
dtype: bool
```

```
#Handling Outliers and replacing them
quant=df.quantile(q=(0.75,0.25))
quant
```

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

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

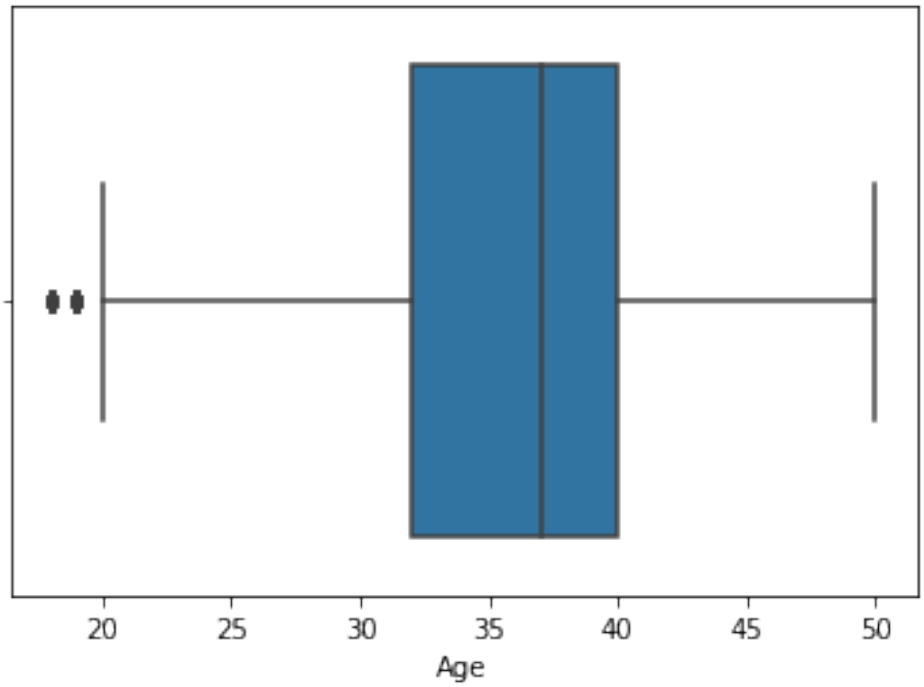
```
df['Age']=np.where(df['Age']>50,40,df['Age'])
#removing the outliers where the age>60
df['Age']
```

```
0       42
1       41
2       42
3       39
4       43
        ..
9995    39
9996    35
9997    36
9998    42
9999    28
Name: Age, Length: 10000, dtype: int64
```

```
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
```

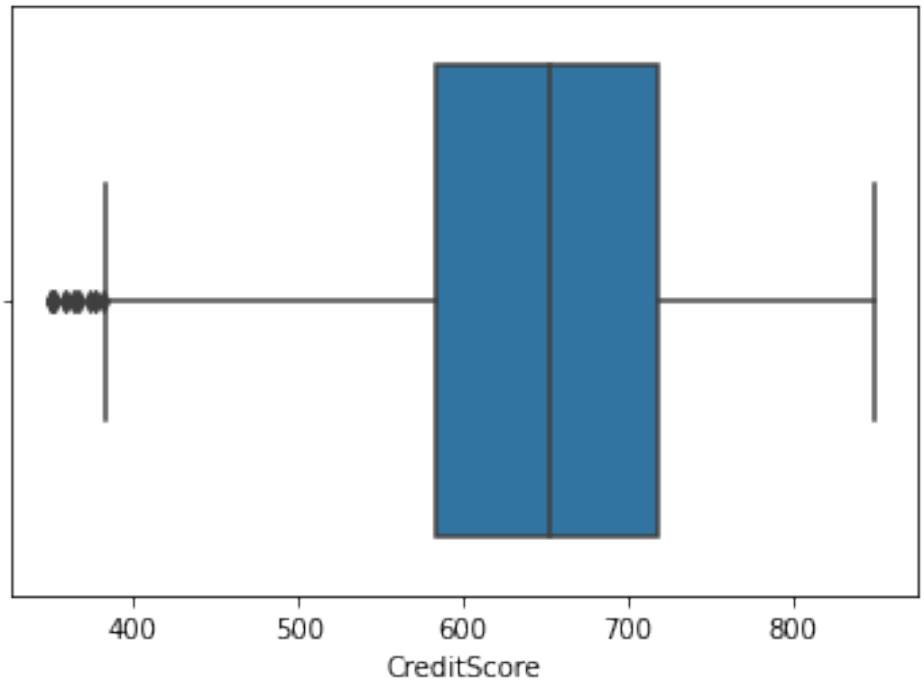
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae4052790>
```




```
df['Age']=np.where(df['Age']<20,35,df['Age']) #removing the outliers where age<20

#CreditScore

sns.boxplot(df['CreditScore'])
```

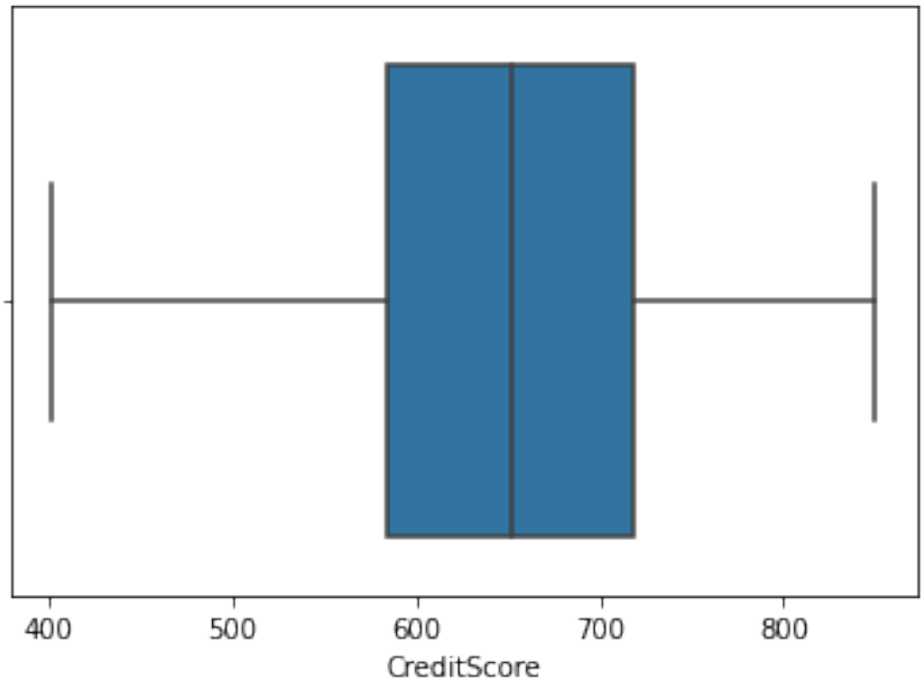
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ae3fb5f50>
```




```
df['CreditScore']=np.where(df['CreditScore']<400,600,df['CreditScore']
)
sns.boxplot(df['CreditScore'])
```

```
/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 0x7f4ae3f7f7d0>
```




```
#Performing Encoding
df['Gender'].replace({'Female':0,'Male':1},inplace=True)
df.head(10)
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age
0	1	15634602	Hargrave	619	France	0	42
1	2	15647311	Hill	608	Spain	0	41
2	3	15619304	Onio	502	France	0	42
3	4	15701354	Boni	699	France	0	39
4	5	15737888	Mitchell	850	Spain	0	43
5	6	15574012	Chu	645	Spain	1	44
6	7	15592531	Bartlett	822	France	1	50
7	8	15656148	Obinna	376	Germany	0	29
8	9	15792365	He	501	France	1	44
9	10	15592389	H?	684	France	1	27

	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
5       8  113755.78              2          1               0
6       7       0.00              2          1               1
7       4  115046.74              4          1               0
8       4  142051.07              2          0               1
9       2  134603.88              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
5        149756.71       1
6         10062.80       0
7        119346.88       1
8         74940.50       0
9         71725.73       0
```

```
data=pd.get_dummies(df,columns=['Geography'])
data
```

```
      RowNumber  CustomerId    Surname  CreditScore  Gender  Age
Tenure  \
0             1    15634602   Hargrave          619       0   42
2
1             2    15647311       Hill          608       0   41
1
2             3    15619304       Onio          502       0   42
8
3             4    15701354       Boni          699       0   39
1
4             5    15737888   Mitchell          850       0   43
2
...         ...         ...        ...          ...     ...  ...     .
..
9995       9996    15606229   Obijiaku          771       1   39
5
9996       9997    15569892  Johnstone          516       1   35
10
9997       9998    15584532        Liu          709       0   36
7
9998       9999    15682355  Sabbatini          772       1   42
3
9999      10000    15628319     Walker          792       0   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]
```

```
#Spliting independent and dependent variables
y=data['Exited']
x=data.drop(columns=['Exited'],axis=1)
x.head()
```

```
   RowNumber  CustomerId   Surname  CreditScore  Gender  Age
Tenure  \
0          1    15634602  Hargrave          619       0   42       2
```

```
1          2    15647311      Hill          608       0   41       1

2          3    15619304      Onio          502       0   42       8

3          4    15701354      Boni          699       0   39       1

4          5    15737888  Mitchell          850       0   43       2


     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

   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

y.head()

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

x=x.drop(columns=['Surname'],axis=1)
x=x.drop(columns=['RowNumber'],axis=1)
x

      CustomerId  CreditScore  Gender  Age  Tenure    Balance
NumOfProducts  \
0       15634602          619       0   42       2       0.00
1
1       15647311          608       0   41       1   83807.86
1
2       15619304          502       0   42       8  159660.80
3
```

```
3       15701354          699       0   39       1       0.00
2
4       15737888          850       0   43       2  125510.82
1
...          ...          ...     ...  ...     ...        ...
...
9995    15606229          771       1   39       5       0.00
2
9996    15569892          516       1   35      10   57369.61
1
9997    15584532          709       0   36       7       0.00
1
9998    15682355          772       1   42       3   75075.31
2
9999    15628319          792       0   28       4  130142.79
1

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

      Geography_Germany  Geography_Spain
0                     0                0
1                     0                1
2                     0                0
3                     0                0
4                     0                1
...                 ...              ...
9995                  0                0
9996                  0                0
9997                  0                0
9998                  1                0
9999                  0                0

[10000 rows x 13 columns]
```

```
#Scaling the independent Variable
x=scale(x)
x
```

```
array([[-0.78321342, -0.32622142, -1.09598752, ...,  0.99720391,
        -0.57873591, -0.57380915],
       [-0.60653412, -0.44003595, -1.09598752, ..., -1.00280393,
        -0.57873591,  1.74273971],
       [-0.99588476, -1.53679418, -1.09598752, ...,  0.99720391,
        -0.57873591, -0.57380915],
       ...,
       [-1.47928179,  0.60498839, -1.09598752, ...,  0.99720391,
        -0.57873591, -0.57380915],
       [-0.11935577,  1.25683526,  0.91241915, ..., -1.00280393,
         1.72790383, -0.57380915],
       [-0.87055909,  1.46377078, -1.09598752, ...,  0.99720391,
        -0.57873591, -0.57380915]])
```

```
x.mean()
```

```
5.437291642570367e-16
```

```
x.std()
```

```
0.9999999999999999
```

```
#Spliting 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=1)
x_train.shape
```

```
(8000, 13)
```

```
x_test.shape
```

```
(2000, 13)
```

```
y_train.shape
```

```
(8000,)
```

```
y_test.shape
```

```
(2000,)
```