

Assignment -2

Python Programming

Assignment Date	29 September 2022
Student Name	Ms.Madhumitha.M
Student Roll Number	113219071016
Maximum Marks	2 Marks

Download the Dataset

[Churn_Modelling.csv | Kaggle](#)

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

Dataset loading

Solution :

```
data = pd.read_csv(r'C:\Users\Sureeth\Desktop\Churn_Modelling.csv')
data.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
```

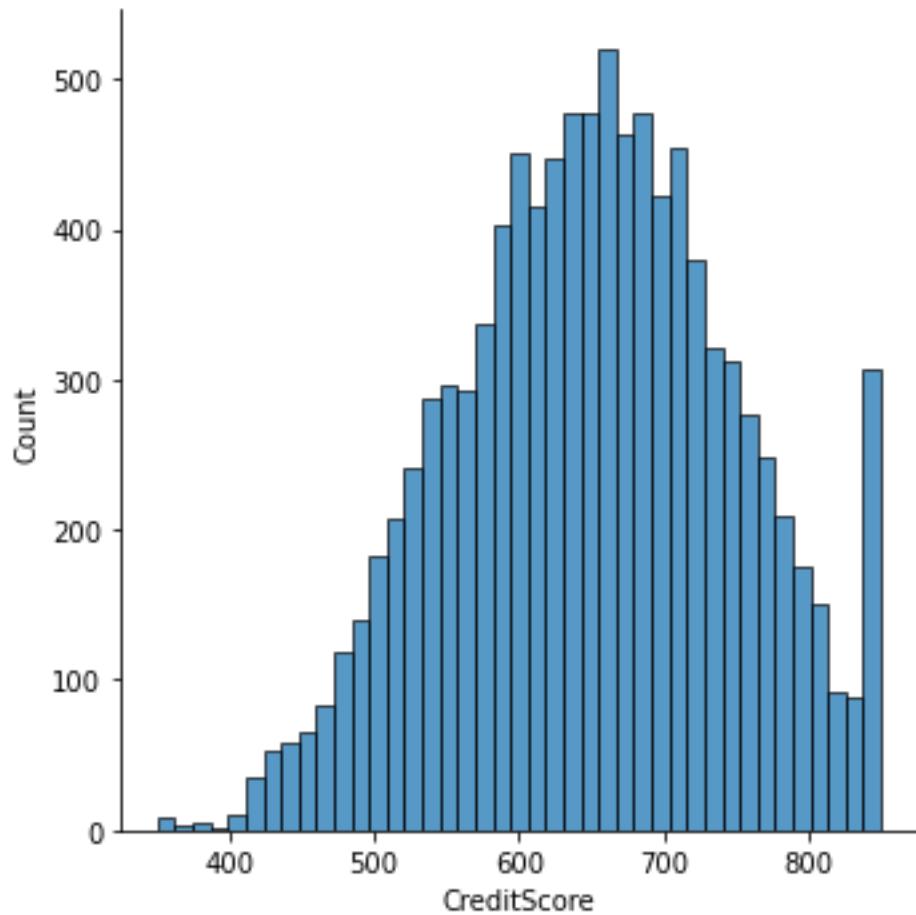
```
1      112542.58      0
2      113931.57      1
3      93826.63      0
4      79084.10      0
```

Visualizations

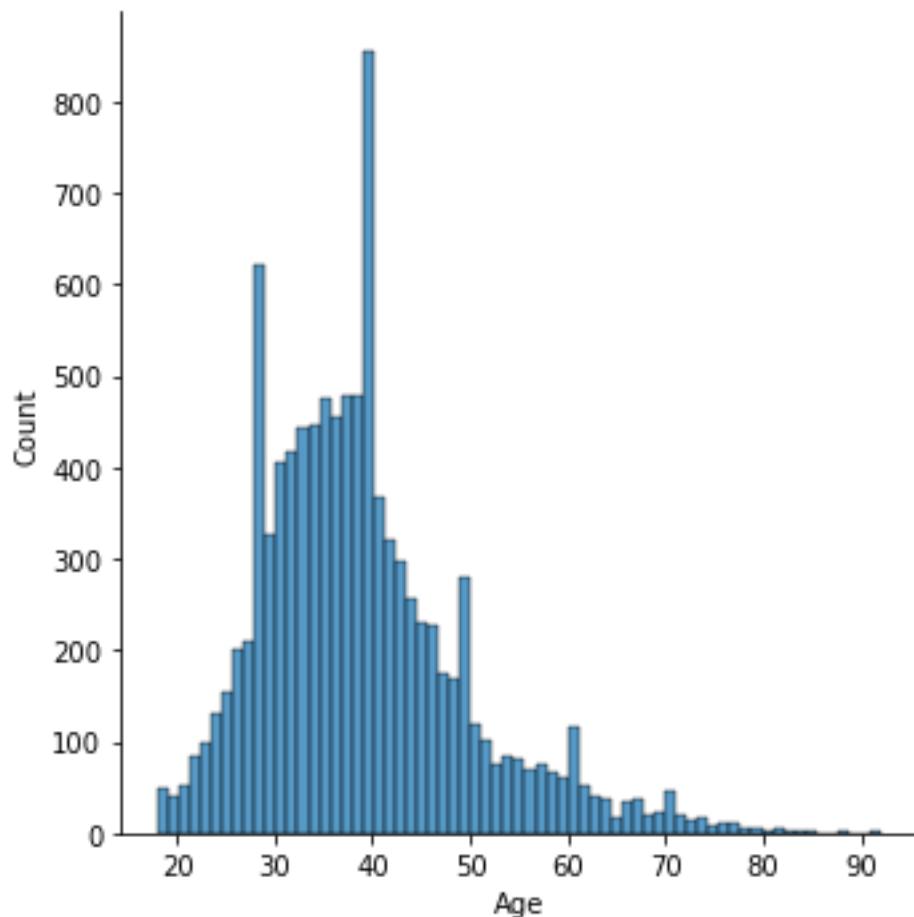
Univariate Analysis

Solution:

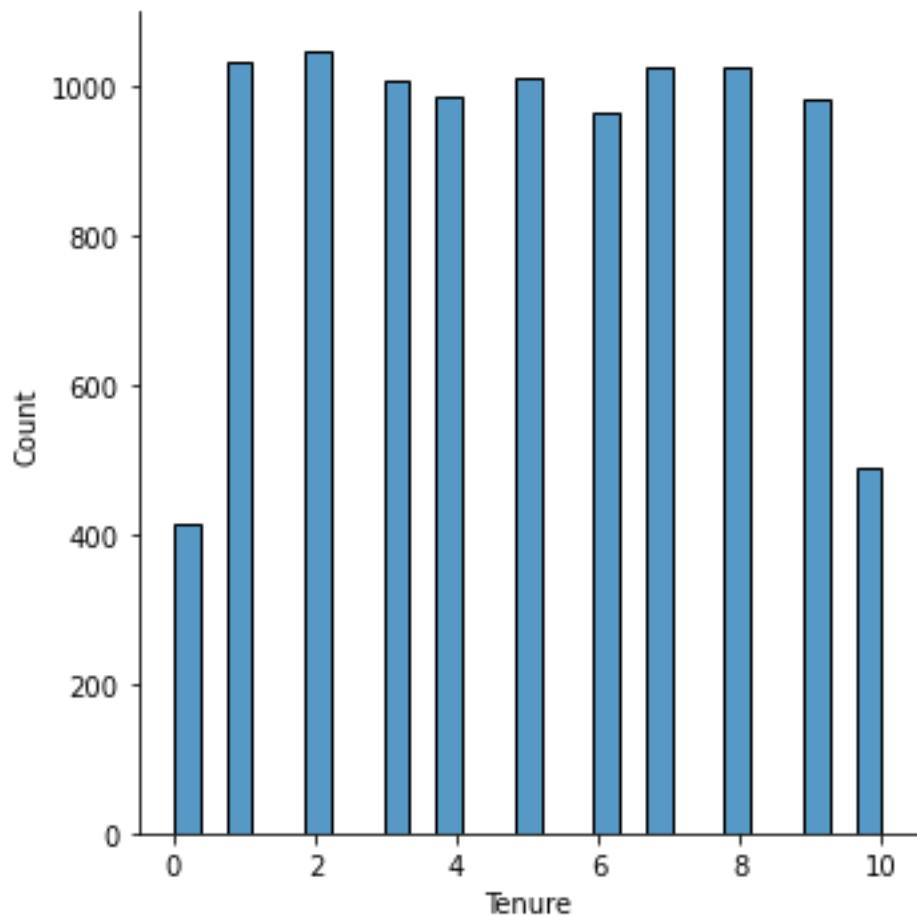
```
sns.displot(data.CreditScore)
<seaborn.axisgrid.FacetGrid at 0x26bd9c96610>
```



```
sns.displot(data.Age)
<seaborn.axisgrid.FacetGrid at 0x26bf8f28490>
```



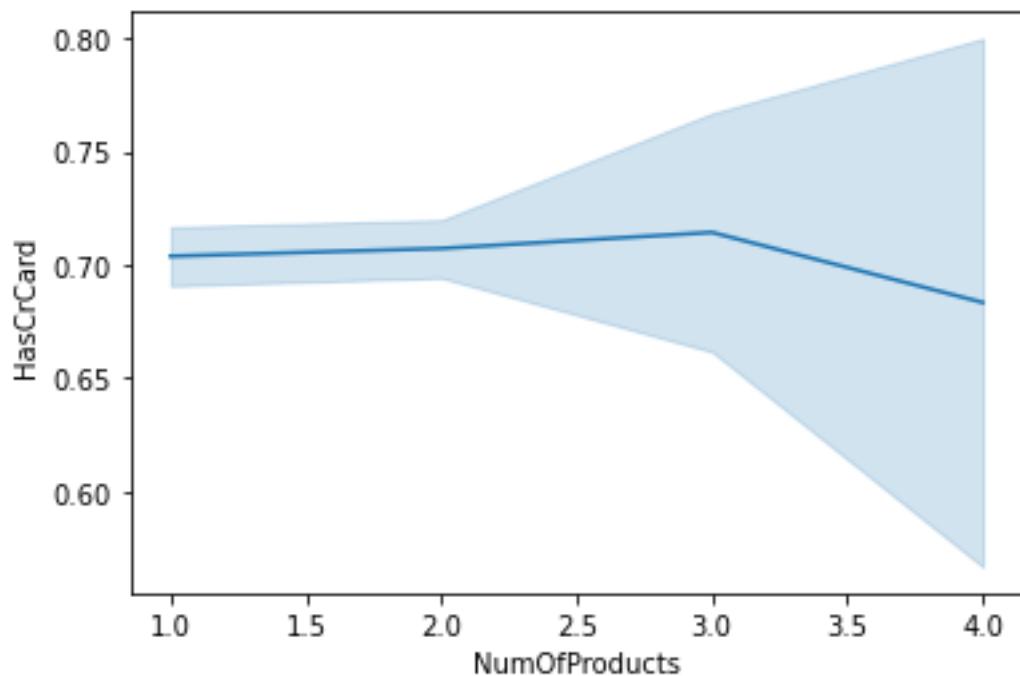
```
sns.displot(data.Tenure)  
<seaborn.axisgrid.FacetGrid at 0x26bf6cd5f70>
```



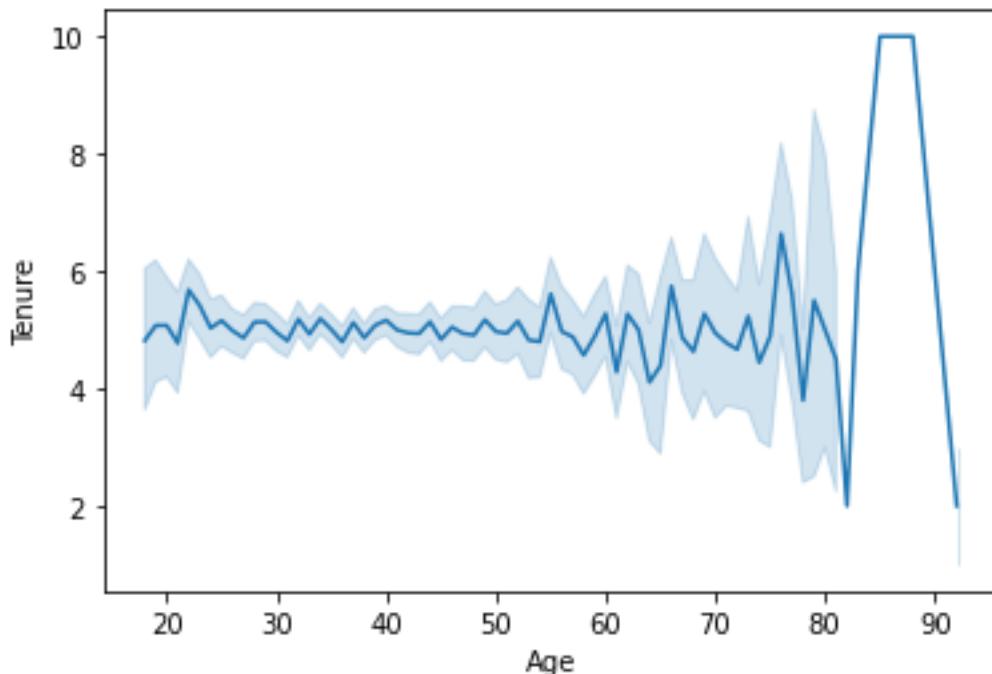
Bi-Variate Analysis

Solution:

```
sns.lineplot(x=data.NumOfProducts, y=data.HasCrCard)  
<AxesSubplot:xlabel='NumOfProducts', ylabel='HasCrCard'>
```



```
sns.lineplot(x=data.Age, y=data.Tenure)
<AxesSubplot:xlabel='Age', ylabel='Tenure'>
```



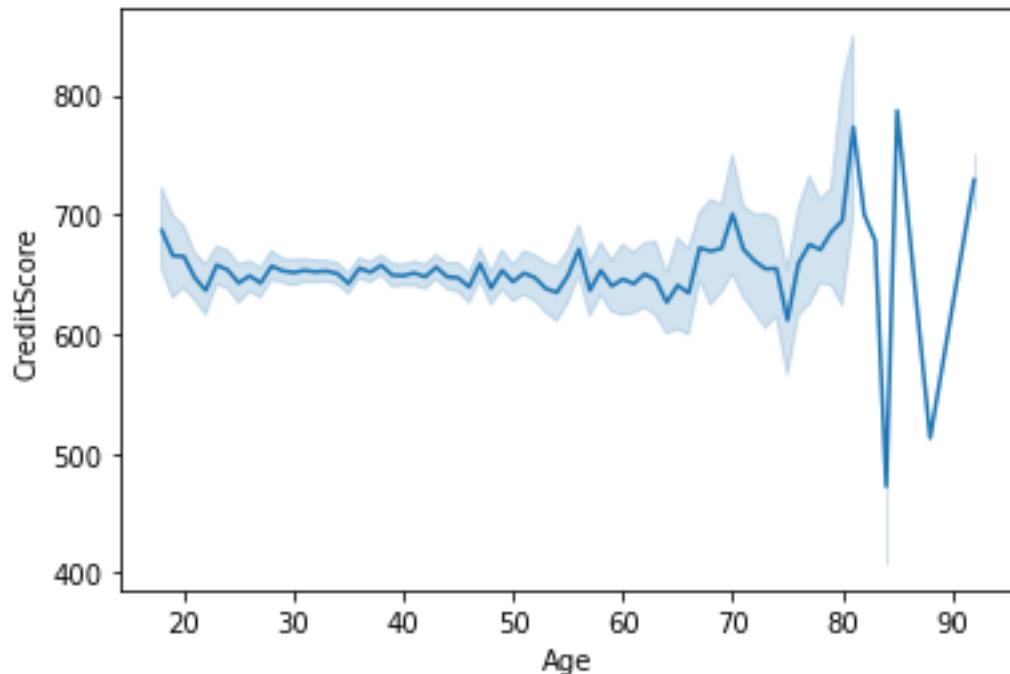
```
sns.lineplot(data.Age,data.CreditScore)
```

```
C:\Users\vijay\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
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.
```

```
    warnings.warn(
```

```
<AxesSubplot:xlabel='Age', ylabel='CreditScore'>
```

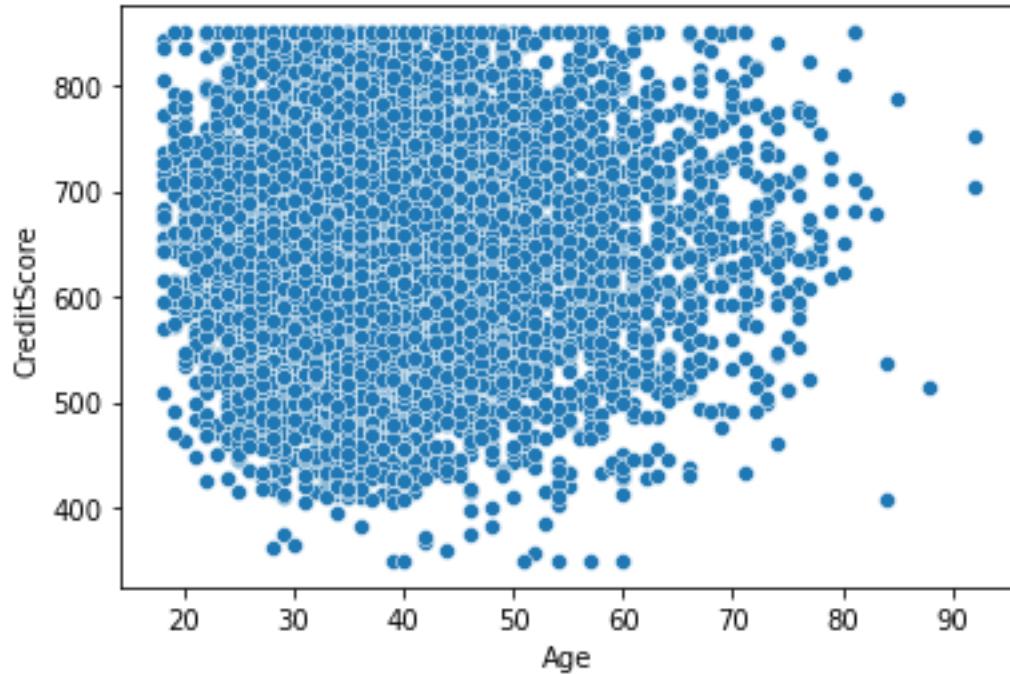


```
sns.scatterplot(data.Age,data.CreditScore)
```

```
C:\Users\vijay\anaconda3\lib\site-packages\seaborn\_decorators.py:36:  
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.
```

```
    warnings.warn(
```

```
<AxesSubplot:xlabel='Age', ylabel='CreditScore'>
```

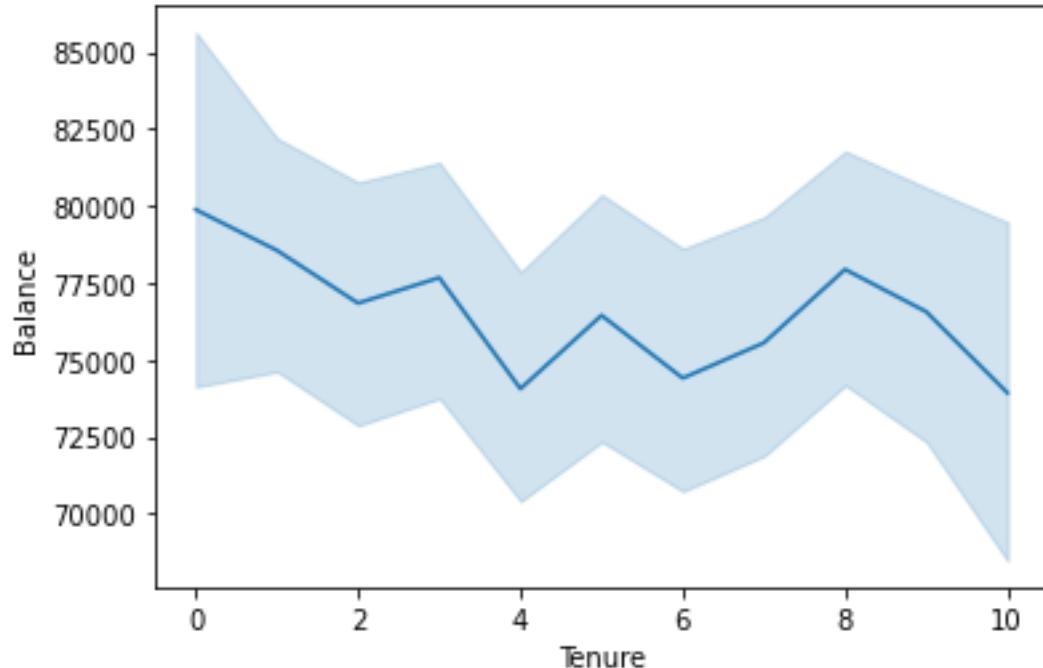


```
sns.lineplot(data.Tenure,data.Balance)
```

```
C:\Users\vijay\anaconda3\lib\site-packages\seaborn\_decorators.py:36:  
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.
```

```
warnings.warn(
```

```
<AxesSubplot:xlabel='Tenure', ylabel='Balance'>
```

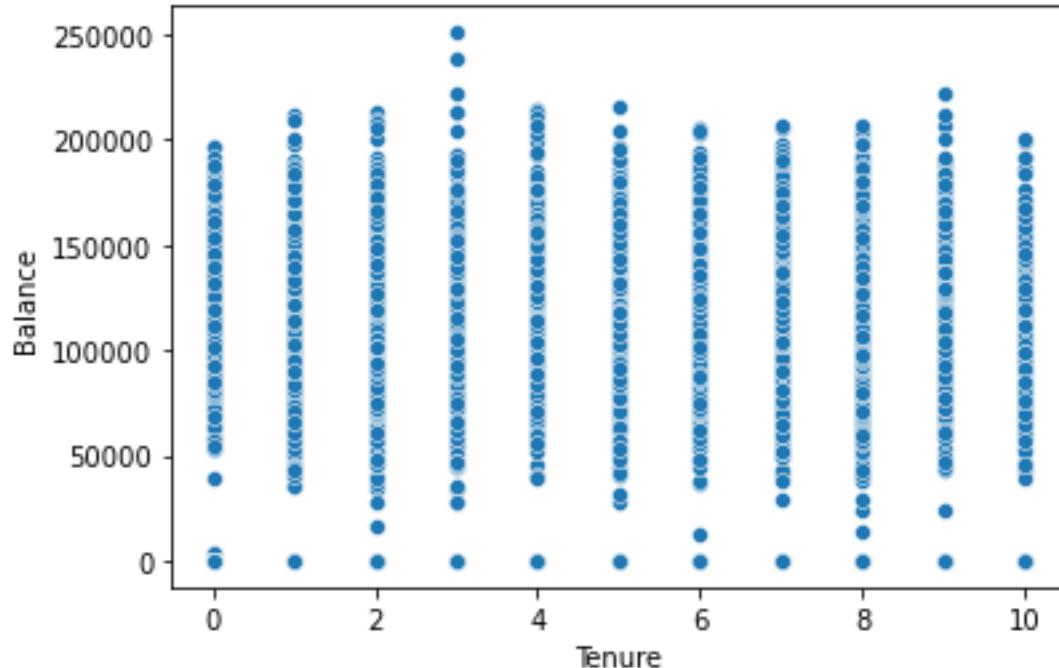


```
sns.scatterplot(data.Tenure,data.Balance)
```

```
C:\Users\vijay\anaconda3\lib\site-packages\seaborn\_decorators.py:36:  
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.
```

```
warnings.warn(
```

```
<AxesSubplot:xlabel='Tenure', ylabel='Balance'>
```

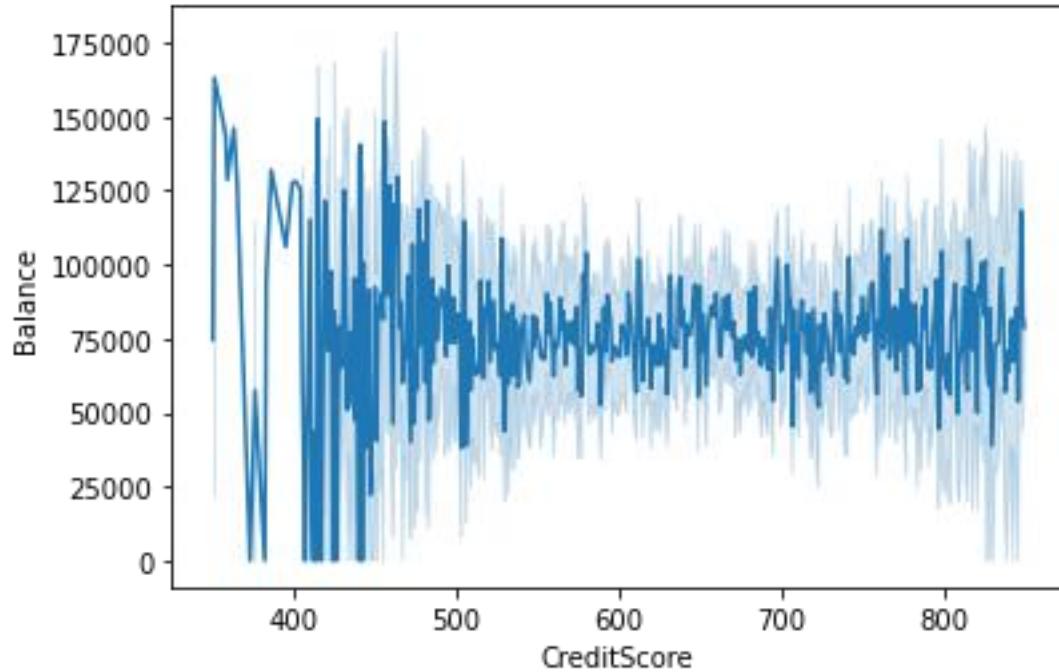


```
sns.lineplot(data.CreditScore,data.Balance)
```

```
C:\Users\vijay\anaconda3\lib\site-packages\seaborn\_decorators.py:36:  
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.
```

```
warnings.warn(
```

```
<AxesSubplot:xlabel='CreditScore', ylabel='Balance'>
```

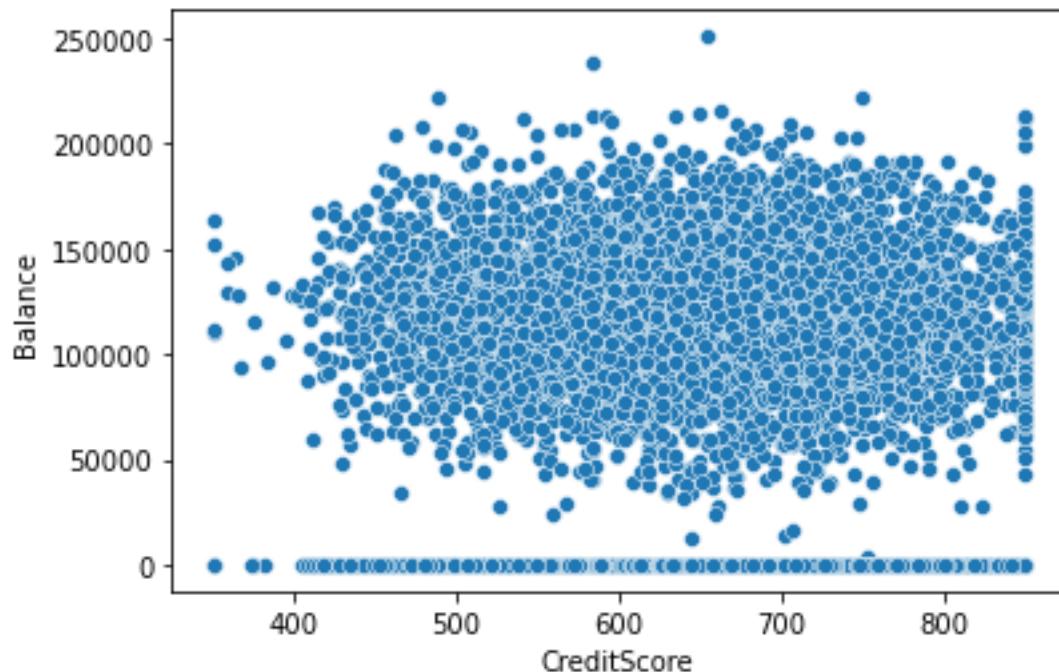


```
sns.scatterplot(data.CreditScore,data.Balance)
```

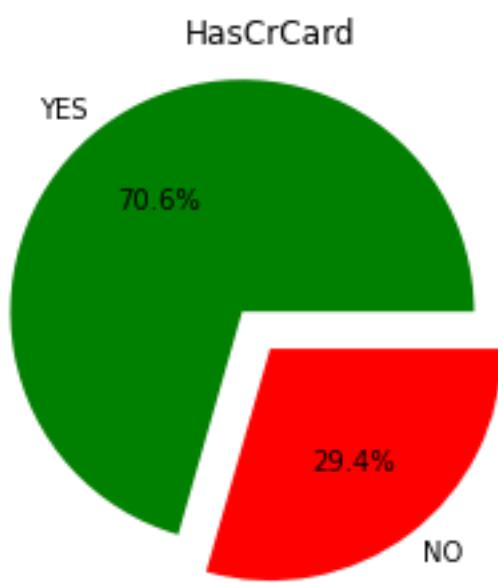
```
C:\Users\vijay\anaconda3\lib\site-packages\seaborn\_decorators.py:36:  
FutureWarning: Pass the following variables as keyword args: x, y. From  
version 0.12, the only valid positional argument will be `data`, and passing  
other arguments without an explicit keyword will result in an error or  
misinterpretation.
```

```
warnings.warn(
```

```
<AxesSubplot:xlabel='CreditScore', ylabel='Balance'>
```



```
plt.pie(data.HasCrCard.value_counts(), [0.2,0], labels=['YES','NO'], autopct="%1.1f%", colors=['green','red'])
plt.title('HasCrCard')
Text(0.5, 1.0, 'HasCrCard')
```



```
data.HasCrCard.value_counts()
```

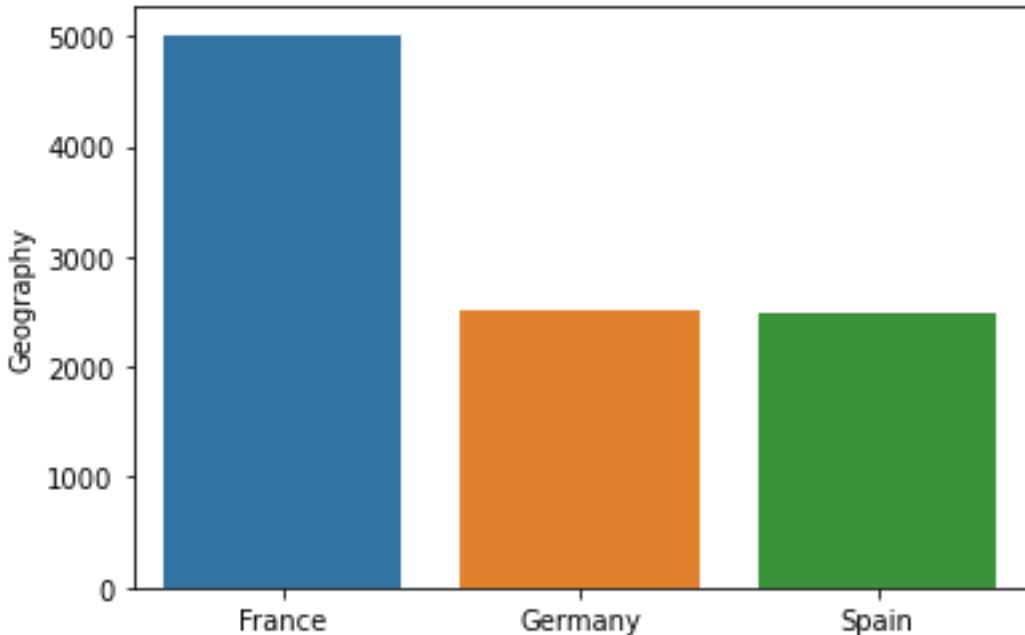
```
1    7055
0    2945
Name: HasCrCard, dtype: int64

sns.barplot(data.Geography.value_counts().index,data.Geography.value_counts())

C:\Users\vijay\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
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.

warnings.warn(
```

```
<AxesSubplot:ylabel='Geography'>
```

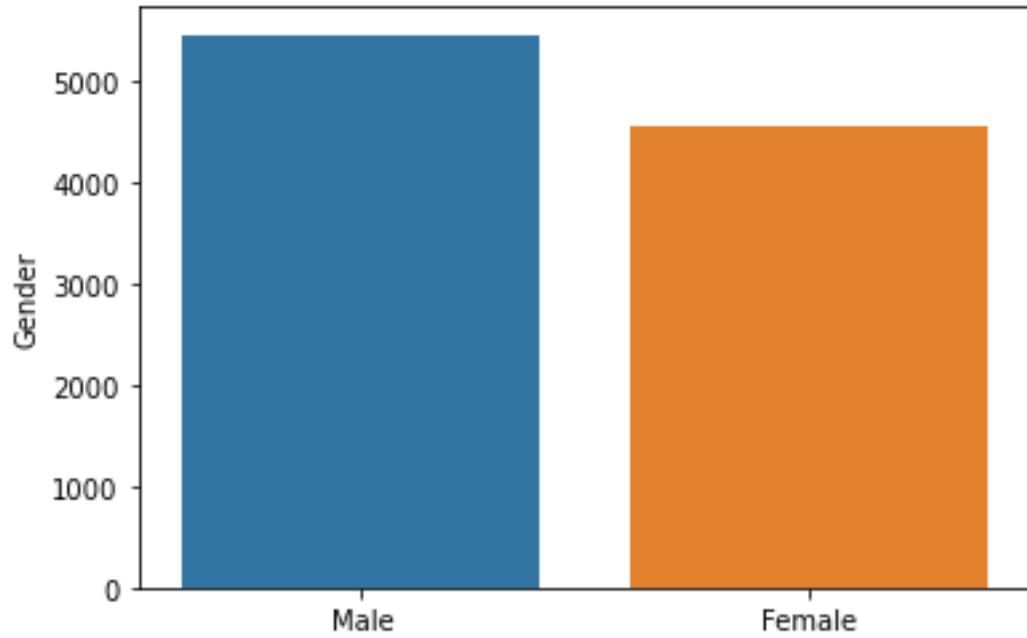


```
sns.barplot(data.Gender.value_counts().index,data.Gender.value_counts())
```

```
C:\Users\vijay\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
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.

warnings.warn(
```

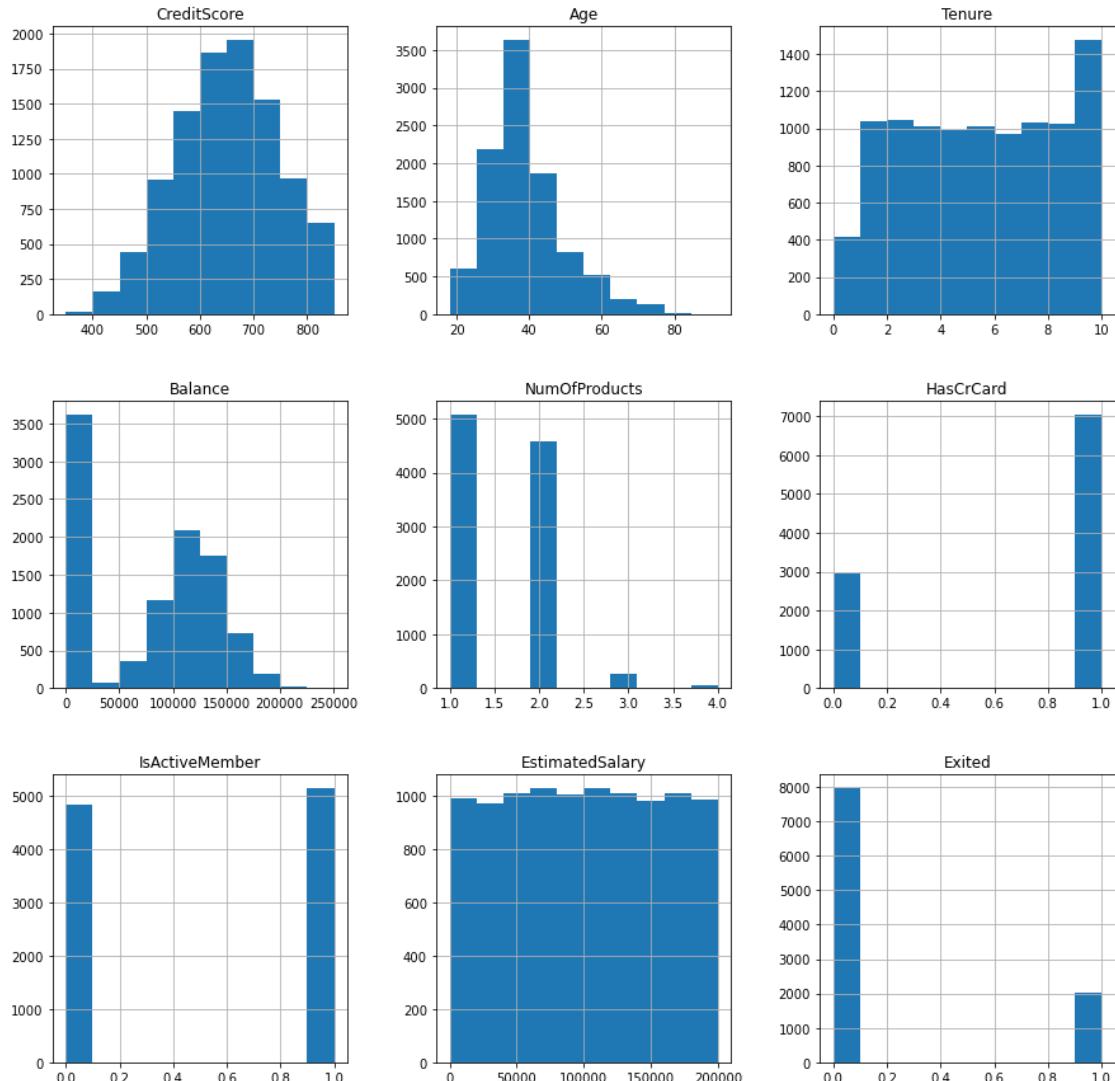
```
<AxesSubplot:ylabel='Gender'>
```



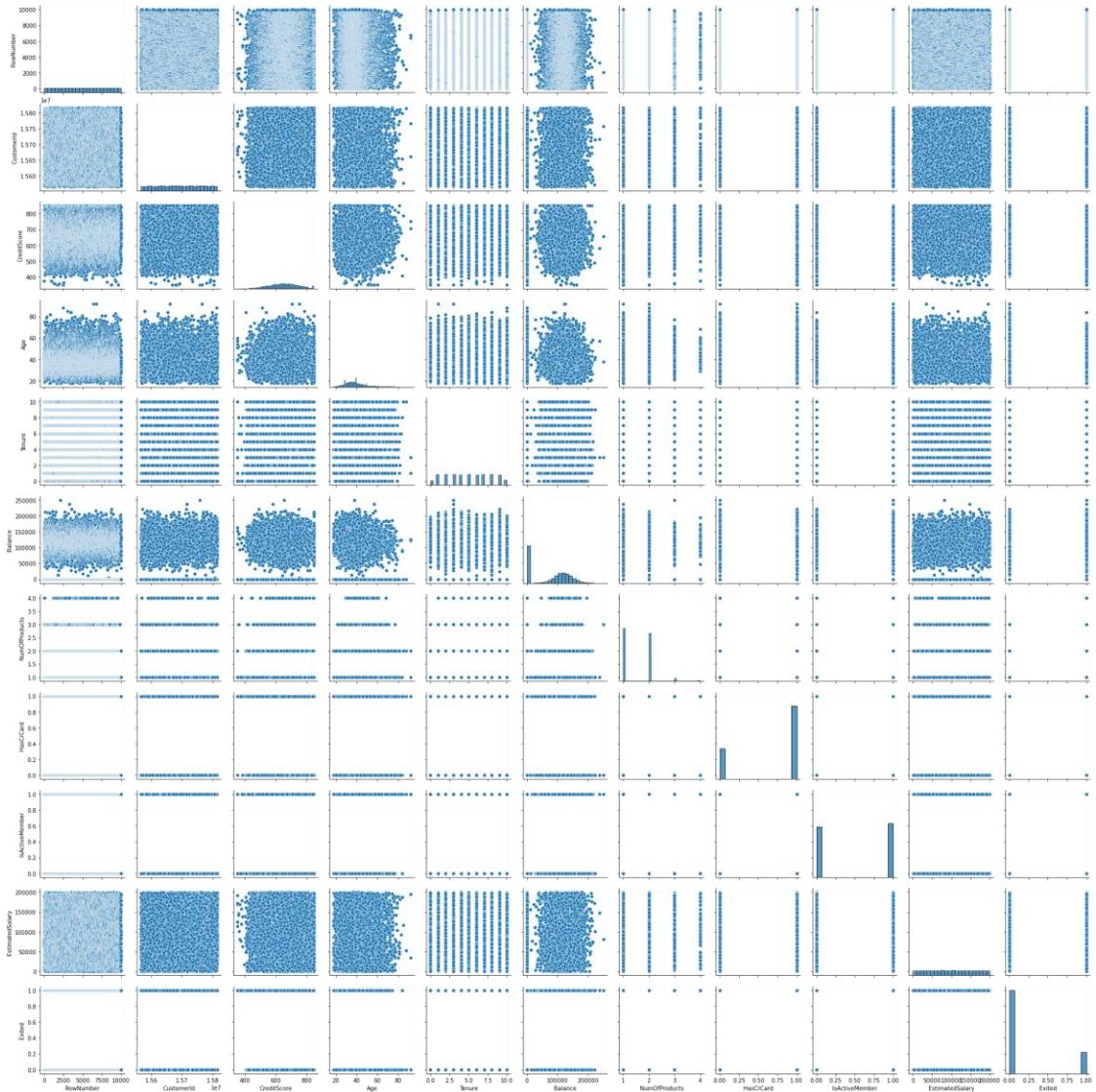
Multi-Variate Analysis

```
data.hist(figsize=(15,15))

array([[<AxesSubplot:title={'center':'CreditScore'}>,
       <AxesSubplot:title={'center':'Age'}>,
       <AxesSubplot:title={'center':'Tenure'}>],
      [<AxesSubplot:title={'center':'Balance'}>,
       <AxesSubplot:title={'center':'NumOfProducts'}>,
       <AxesSubplot:title={'center':'HasCrCard'}>],
      [<AxesSubplot:title={'center':'IsActiveMember'}>,
       <AxesSubplot:title={'center':'EstimatedSalary'}>,
       <AxesSubplot:title={'center':'Exited'}>]], dtype=object)
```

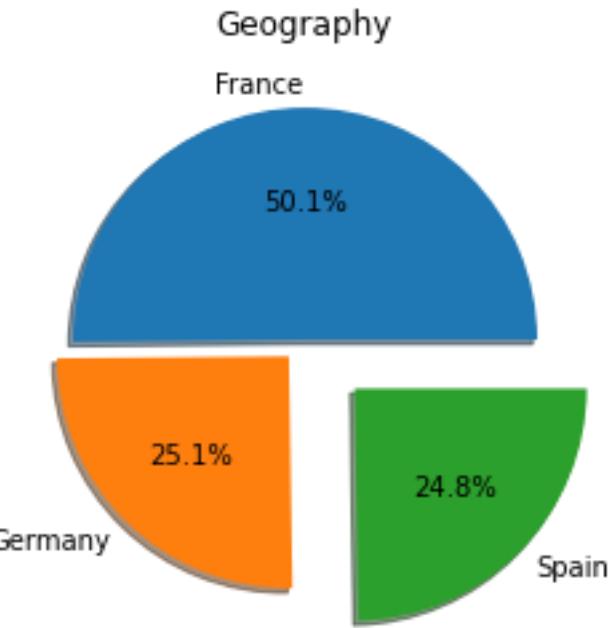


```
sns.pairplot(data)  
<seaborn.axisgrid.PairGrid at 0x26bf6d1e070>
```



```
plt.pie(data.Geography.value_counts(),[0,0.1,0.3],shadow=True,labels=['France','Germany','Spain'],autopct="%1.1f%")
plt.title('Geography')
```

```
Text(0.5, 1.0, 'Geography')
```



Descriptive statistics on the dataset

`data.describe()`

	RowNumber	CustomerId	CreditScore	Age	Tenure	\
count	10000.00000	1.000000e+04	10000.00000	10000.00000	10000.00000	
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	350.00000	18.00000	0.00000	
25%	2500.75000	1.562853e+07	584.00000	32.00000	3.00000	
50%	5000.50000	1.569074e+07	652.00000	37.00000	5.00000	
75%	7500.25000	1.575323e+07	718.00000	44.00000	7.00000	
max	10000.00000	1.581569e+07	850.00000	92.00000	10.00000	
	Balance	NumOfProducts	HasCrCard	IsActiveMember		\
count	10000.00000	10000.00000	10000.00000	10000.00000		
mean	76485.889288	1.530200	0.70550	0.515100		
std	62397.405202	0.581654	0.45584	0.499797		
min	0.00000	1.00000	0.00000	0.00000		
25%	0.00000	1.00000	0.00000	0.00000		
50%	97198.540000	1.00000	1.00000	1.00000		
75%	127644.240000	2.00000	1.00000	1.00000		
max	250898.090000	4.00000	1.00000	1.00000		
	EstimatedSalary	Exited				
count	10000.00000	10000.00000				
mean	100090.239881	0.203700				
std	57510.492818	0.402769				
min	11.580000	0.000000				

```
25%      51002.110000      0.000000
50%      100193.915000      0.000000
75%      149388.247500      0.000000
max      199992.480000      1.000000

data.Geography.unique()

array(['France', 'Spain', 'Germany'], dtype=object)

data.Gender.value_counts()

Male      5457
Female    4543
Name: Gender, dtype: int64

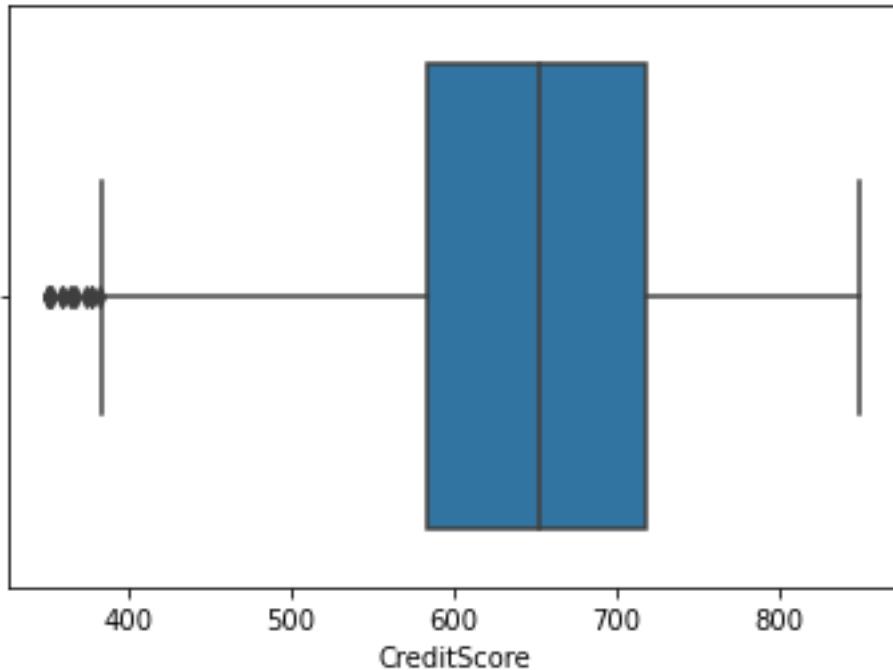
data.Geography.value_counts()

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

Handling the missing data and outliers

```
sns.boxplot(data.CreditScore)

C:\Users\vijay\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
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.
    warnings.warn(
<AxesSubplot:xlabel='CreditScore'>
```



```
q1=data.CreditScore.quantile(0.25)
q3=data.CreditScore.quantile(0.75)
```

```
IQR=q3-q1
```

```
upper_limit= q3 + 1.5*IQR
lower_limit= q1 - 1.5*IQR

print("Upper limit :",upper_limit)
print("Lower limit :",lower_limit)
```

```
Upper limit : 919.0
Lower limit : 383.0
```

```
data.median()
```

```
C:\Users\vijay\AppData\Local\Temp\ipykernel_2108\4184645713.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.

data.median()
```

CreditScore	652.000
Age	37.000
Tenure	5.000
Balance	97198.540
NumOfProducts	1.000
HasCrCard	1.000
IsActiveMember	1.000
EstimatedSalary	100193.915

```

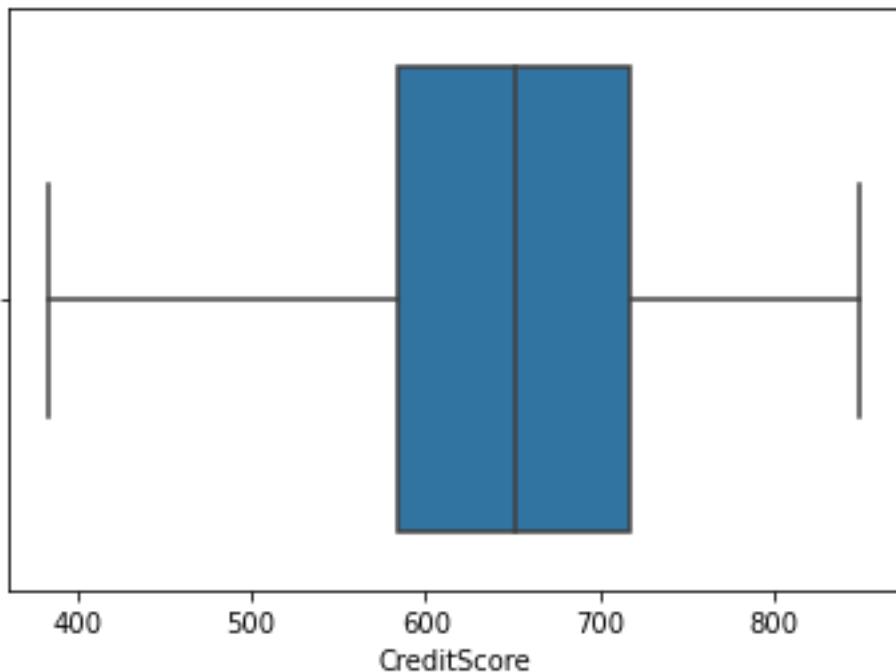
Exited          0.000
dtype: float64

data['CreditScore']=
np.where(data['CreditScore']<lower_limit,6.520000e+02,data['CreditScore'])

sns.boxplot(data.CreditScore)

C:\Users\vijay\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
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.
    warnings.warn(
<AxesSubplot:xlabel='CreditScore'>

```



Label Encoding

```

from sklearn.preprocessing import LabelEncoder

le=LabelEncoder()

data.Gender=le.fit_transform(data.Gender)

data.head(10)

```

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	\
0	619.0	France	0	42	2	0.00		1
1	608.0	Spain	0	41	1	83807.86		1

```

2      502.0    France      0   42      8  159660.80      3
3      699.0    France      0   39      1     0.00      2
4      850.0    Spain       0   43      2  125510.82      1
5      645.0    Spain       1   44      8  113755.78      2
6      822.0    France      1   50      7     0.00      2
7      652.0    Germany     0   29      4  115046.74      4
8      501.0    France      1   44      4  142051.07      2
9      684.0    France      1   27      2  134603.88      1

```

	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	1	101348.88	1
1	0	1	112542.58	0
2	1	0	113931.57	1
3	0	0	93826.63	0
4	1	1	79084.10	0
5	1	0	149756.71	1
6	1	1	10062.80	0
7	1	0	119346.88	1
8	0	1	74940.50	0
9	1	1	71725.73	0

One hot encoding

```

data_main=pd.get_dummies(data,columns=[ 'Geography' ])
data_main.head(15)

```

	CreditScore	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	\
0	619.0	0	42	2	0.00	1	1	
1	608.0	0	41	1	83807.86	1	0	
2	502.0	0	42	8	159660.80	3	1	
3	699.0	0	39	1	0.00	2	0	
4	850.0	0	43	2	125510.82	1	1	
5	645.0	1	44	8	113755.78	2	1	
6	822.0	1	50	7	0.00	2	1	
7	652.0	0	29	4	115046.74	4	1	
8	501.0	1	44	4	142051.07	2	0	
9	684.0	1	27	2	134603.88	1	1	
10	528.0	1	31	6	102016.72	2	0	
11	497.0	1	24	3	0.00	2	1	
12	476.0	0	34	10	0.00	2	1	
13	549.0	0	25	5	0.00	2	0	
14	635.0	0	35	7	0.00	2	1	

	IsActiveMember	EstimatedSalary	Exited	Geography_France	\
0	1	101348.88	1	1	
1	1	112542.58	0	0	
2	0	113931.57	1	1	
3	0	93826.63	0	1	
4	1	79084.10	0	0	

5	0	149756.71	1	0
6	1	10062.80	0	1
7	0	119346.88	1	0
8	1	74940.50	0	1
9	1	71725.73	0	1
10	0	80181.12	0	1
11	0	76390.01	0	0
12	0	26260.98	0	1
13	0	190857.79	0	1
14	1	65951.65	0	0

	Geography_Germany	Geography_Spain
0	0	0
1	0	1
2	0	0
3	0	0
4	0	1
5	0	1
6	0	0
7	1	0
8	0	0
9	0	0
10	0	0
11	0	1
12	0	0
13	0	0
14	0	1

```
data_main.corr()
```

	CreditScore	Gender	Age	Tenure	Balance	\
CreditScore	1.000000	-0.003613	-0.001992	-0.000650	0.007074	
Gender	-0.003613	1.000000	-0.027544	0.014733	0.012087	
Age	-0.001992	-0.027544	1.000000	-0.009997	0.028308	
Tenure	-0.000650	0.014733	-0.009997	1.000000	-0.012254	
Balance	0.007074	0.012087	0.028308	-0.012254	1.000000	
NumOfProducts	0.012293	-0.021859	-0.030680	0.013444	-0.304180	
HasCrCard	-0.003942	0.005766	-0.011721	0.022583	-0.014858	
IsActiveMember	0.023596	0.022544	0.085472	-0.028362	-0.010084	
EstimatedSalary	0.001619	-0.008112	-0.007201	0.007784	0.012797	
Exited	-0.018298	-0.106512	0.285323	-0.014001	0.118533	
Geography_France	-0.009889	0.006772	-0.039208	-0.002848	-0.231329	
Geography_Germany	0.005748	-0.024628	0.046897	-0.000567	0.401110	
Geography_Spain	0.005681	0.016889	-0.001685	0.003868	-0.134892	

	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary
\				
CreditScore	0.012293	-0.003942	0.023596	0.001619
Gender	-0.021859	0.005766	0.022544	-0.008112
Age	-0.030680	-0.011721	0.085472	-0.007201

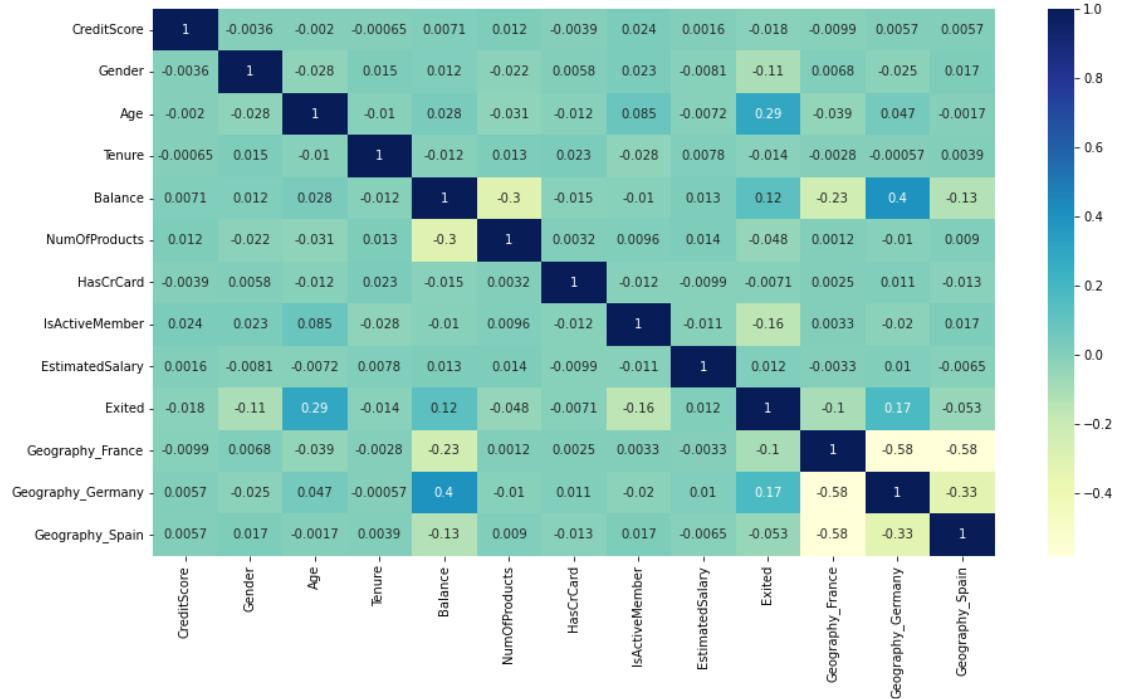
Tenure	0.013444	0.022583	-0.028362	0.007784
Balance	-0.304180	-0.014858	-0.010084	0.012797
NumOfProducts	1.000000	0.003183	0.009612	0.014204
HasCrCard	0.003183	1.000000	-0.011866	-0.009933
IsActiveMember	0.009612	-0.011866	1.000000	-0.011421
EstimatedSalary	0.014204	-0.009933	-0.011421	1.000000
Exited	-0.047820	-0.007138	-0.156128	0.012097
Geography_France	0.001230	0.002467	0.003317	-0.003332
Geography_Germany	-0.010419	0.010577	-0.020486	0.010297
Geography_Spain	0.009039	-0.013480	0.016732	-0.006482

	Exited	Geography_France	Geography_Germany	\
CreditScore	-0.018298	-0.009889	0.005748	
Gender	-0.106512	0.006772	-0.024628	
Age	0.285323	-0.039208	0.046897	
Tenure	-0.014001	-0.002848	-0.000567	
Balance	0.118533	-0.231329	0.401110	
NumOfProducts	-0.047820	0.001230	-0.010419	
HasCrCard	-0.007138	0.002467	0.010577	
IsActiveMember	-0.156128	0.003317	-0.020486	
EstimatedSalary	0.012097	-0.003332	0.010297	
Exited	1.000000	-0.104955	0.173488	
Geography_France	-0.104955	1.000000	-0.580359	
Geography_Germany	0.173488	-0.580359	1.000000	
Geography_Spain	-0.052667	-0.575418	-0.332084	

	Geography_Spain
CreditScore	0.005681
Gender	0.016889
Age	-0.001685
Tenure	0.003868
Balance	-0.134892
NumOfProducts	0.009039
HasCrCard	-0.013480
IsActiveMember	0.016732
EstimatedSalary	-0.006482
Exited	-0.052667
Geography_France	-0.575418
Geography_Germany	-0.332084
Geography_Spain	1.000000

```
plt.figure(figsize=(15,8))
sns.heatmap(data_main.corr(), annot=True, cmap="YlGnBu")
```

<AxesSubplot:>



```
data_main.corr().Exited.sort_values(ascending=False)
```

Exited	1.000000
Age	0.285323
Geography_Germany	0.173488
Balance	0.118533
EstimatedSalary	0.012097
HasCrCard	-0.007138
Tenure	-0.014001
CreditScore	-0.018298
NumOfProducts	-0.047820
Geography_Spain	-0.052667
Geography_France	-0.104955
Gender	-0.106512
IsActiveMember	-0.156128

Name: Exited, dtype: float64

```
data_main.head()
```

	CreditScore	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	\
0	619.0	0	42	2	0.00		1	1
1	608.0	0	41	1	83807.86		1	0
2	502.0	0	42	8	159660.80		3	1
3	699.0	0	39	1	0.00		2	0
4	850.0	0	43	2	125510.82		1	1
	IsActiveMember	EstimatedSalary	Exited	Geography_France	\			
0	1	101348.88	1		1			
1	1	112542.58	0		0			

```

2          0      113931.57      1      1
3          0      93826.63      0      1
4          1      79084.10      0      0

```

	Geography_Germany	Geography_Spain
0	0	0
1	0	1
2	0	0
3	0	0
4	0	1

Spilting of data for Training and Testing

Dependent variable

```

y=data_main['Exited']
print(y)

0      1
1      0
2      1
3      0
4      0
..
9995    0
9996    0
9997    1
9998    1
9999    0
Name: Exited, Length: 10000, dtype: int64

```

Independent variable

```

X=data_main.drop(columns=['Exited'],axis=1)
X.head(10)

```

	CreditScore	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	\
0	619.0	0	42	2	0.00	1	1	
1	608.0	0	41	1	83807.86	1	0	
2	502.0	0	42	8	159660.80	3	1	
3	699.0	0	39	1	0.00	2	0	
4	850.0	0	43	2	125510.82	1	1	
5	645.0	1	44	8	113755.78	2	1	
6	822.0	1	50	7	0.00	2	1	
7	652.0	0	29	4	115046.74	4	1	
8	501.0	1	44	4	142051.07	2	0	
9	684.0	1	27	2	134603.88	1	1	

```

IsActiveMember  EstimatedSalary  Geography_France  Geography_Germany \
0              1             101348.88           1                 0
1              1             112542.58           0                 0
2              0             113931.57           1                 0
3              0             93826.63            1                 0
4              1             79084.10            0                 0
5              0             149756.71           0                 0
6              1             10062.80            1                 0
7              0             119346.88           0                 1
8              1             74940.50            1                 0
9              1             71725.73            1                 0

Geography_Spain
0              0
1              1
2              0
3              0
4              1
5              1
6              0
7              0
8              0
9              0

```

Scaling

```

from sklearn.preprocessing import scale

x_scaled=pd.DataFrame(scale(X),columns=X.columns)
x_scaled.head()

CreditScore    Gender        Age      Tenure     Balance  NumOfProducts \
0   -0.332983 -1.095988  0.293517 -1.041760 -1.225848       -0.911583
1   -0.447572 -1.095988  0.198164 -1.387538  0.117350       -0.911583
2   -1.551792 -1.095988  0.293517  1.032908  1.333053       2.527057
3    0.500391 -1.095988  0.007457 -1.387538 -1.225848       0.807737
4    2.073384 -1.095988  0.388871 -1.041760  0.785728       -0.911583

HasCrCard  IsActiveMember  EstimatedSalary  Geography_France \
0    0.646092       0.970243      0.021886       0.997204
1   -1.547768       0.970243      0.216534      -1.002804
2    0.646092      -1.030670      0.240687       0.997204
3   -1.547768      -1.030670     -0.108918       0.997204
4    0.646092       0.970243     -0.365276      -1.002804

Geography_Germany  Geography_Spain
0      -0.578736      -0.573809
1      -0.578736      1.742740
2      -0.578736     -0.573809

```

3	-0.578736	-0.573809
4	-0.578736	1.742740

Train Test Split

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test =
train_test_split(x_scaled,y,test_size=0.3,random_state=0)

X_train.shape
(7000, 12)

y_train.shape
(7000,)

X_test.shape
(3000, 12)

y_test.shape
(3000,)
```