In [1]:
```python
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import confusion_matrix,accuracy_score
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```python
data=pd.read_csv("D:\madhu\Churn_Modelling.csv")
data.head()
```

Out[2]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

In [4]:
```python
data.tail()
```

Out[4]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
9995	9996	15606229	Obijiaku	771	France	Male	39	5	0.00	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	101699.77	0
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	42085.58	1
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	92888.52	1
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	38190.78	0

In [5]:
```python
data.describe()
```

Out[5]:

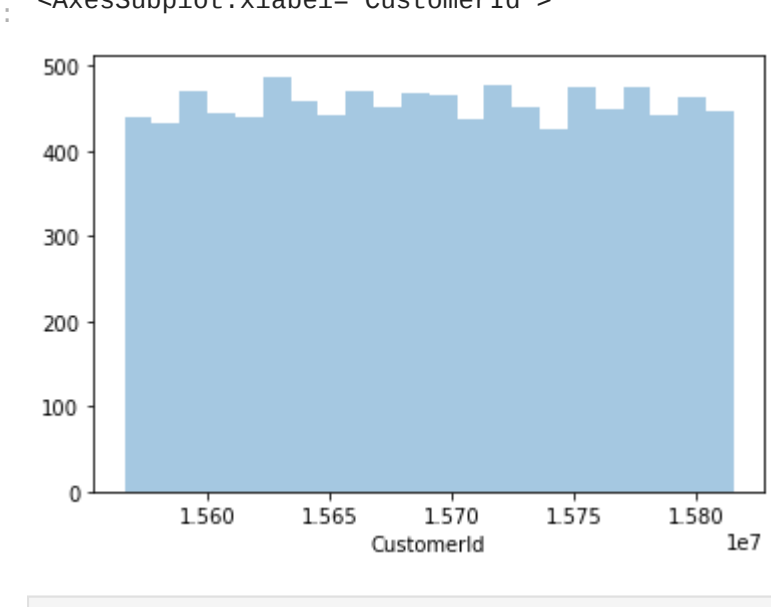
	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.00000	10000.000000	10000.000000	10000.000000
mean	5000.50000	1.569094e+07	650.528800	38.921800	5.012800	76485.889288	1.530200	0.70550	0.515100	100090.239881	0.203700
std	2886.89568	7.193619e+04	96.653299	10.487806	2.892174	62397.405202	0.581654	0.45584	0.499797	57510.492818	0.402769
min	1.00000	1.556570e+07	350.000000	18.000000	0.000000	0.000000	1.000000	0.00000	0.000000	11.580000	0.000000
25%	2500.75000	1.562853e+07	584.000000	32.000000	3.000000	0.000000	1.000000	0.00000	0.000000	51002.110000	0.000000
50%	5000.50000	1.569074e+07	652.000000	37.000000	5.000000	97198.540000	1.000000	1.00000	1.000000	100193.915000	0.000000
75%	7500.25000	1.575323e+07	718.000000	44.000000	7.000000	127644.240000	2.000000	1.00000	1.000000	149388.247500	0.000000
max	10000.00000	1.581569e+07	850.000000	92.000000	10.000000	250898.090000	4.000000	1.00000	1.000000	199992.480000	1.000000

univariate analysis

In [6]:
```python
#histogram
sns.distplot(data['CustomerId'],kde=False)
```

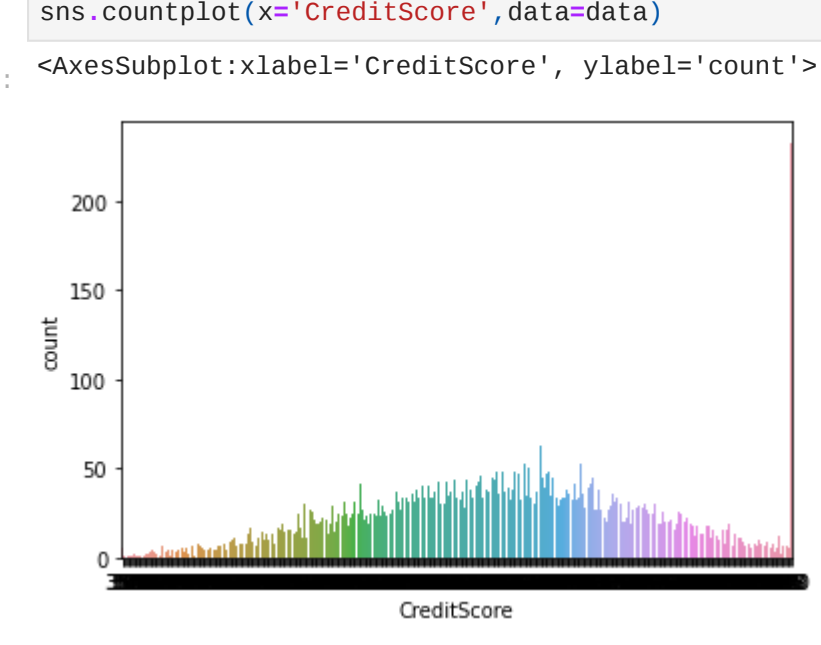
```
C:\anaconda\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Pleas
e adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[6]: <AxesSubplot:xlabel='CustomerId'>



In [7]:
```python
#count plot
sns.countplot(x='CreditScore',data=data)
```

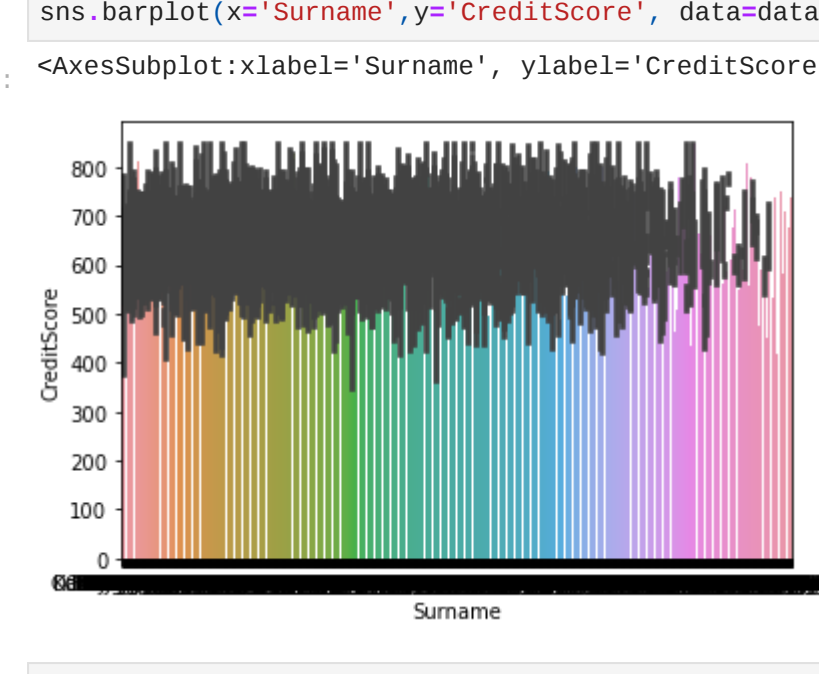
Out[7]: <AxesSubplot:xlabel='CreditScore', ylabel='count'>



Bi-Variate Analysis

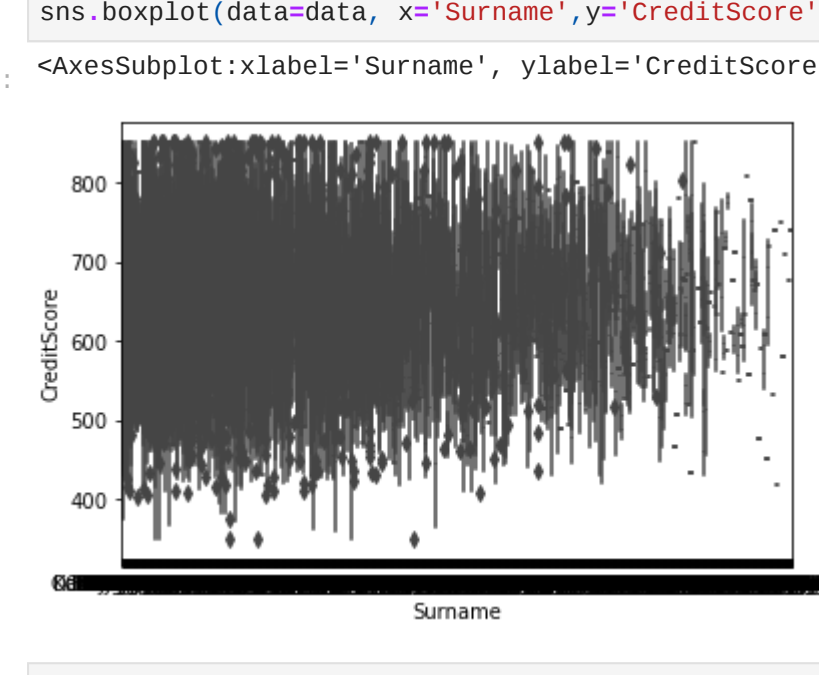
In [12]:
```python
#bar chart
sns.barplot(x='Surname',y='CreditScore', data=data)
```

Out[12]: <AxesSubplot:xlabel='Surname', ylabel='CreditScore'>



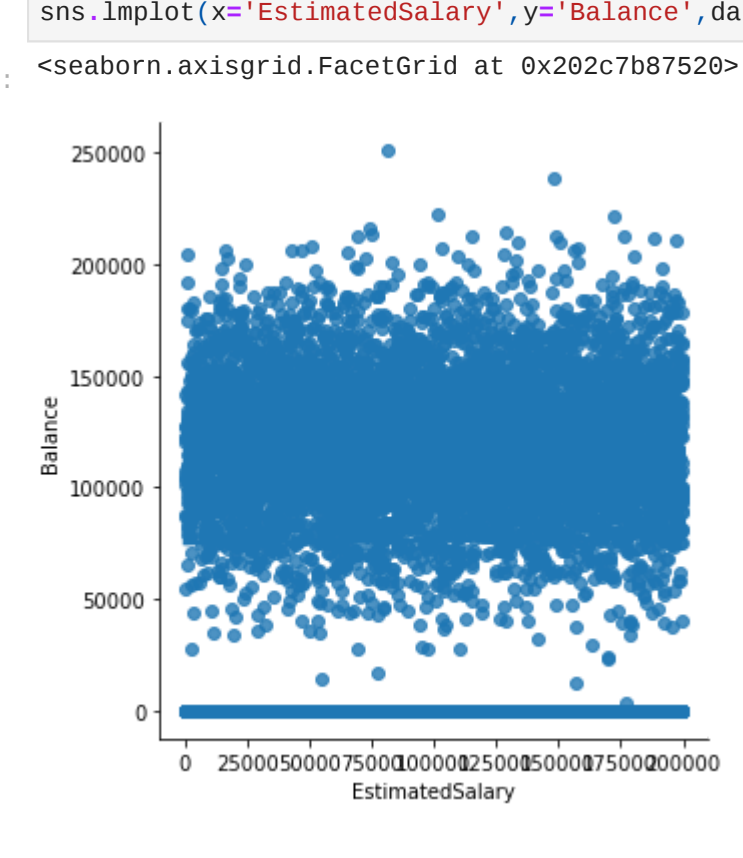
In [13]:
```python
#box plot
sns.boxplot(data=data, x='Surname',y='CreditScore')
```

Out[13]: <AxesSubplot:xlabel='Surname', ylabel='CreditScore'>



In [14]:
```python
#regression plot
sns.lmplot(x='EstimatedSalary',y='Balance',data=data)
```

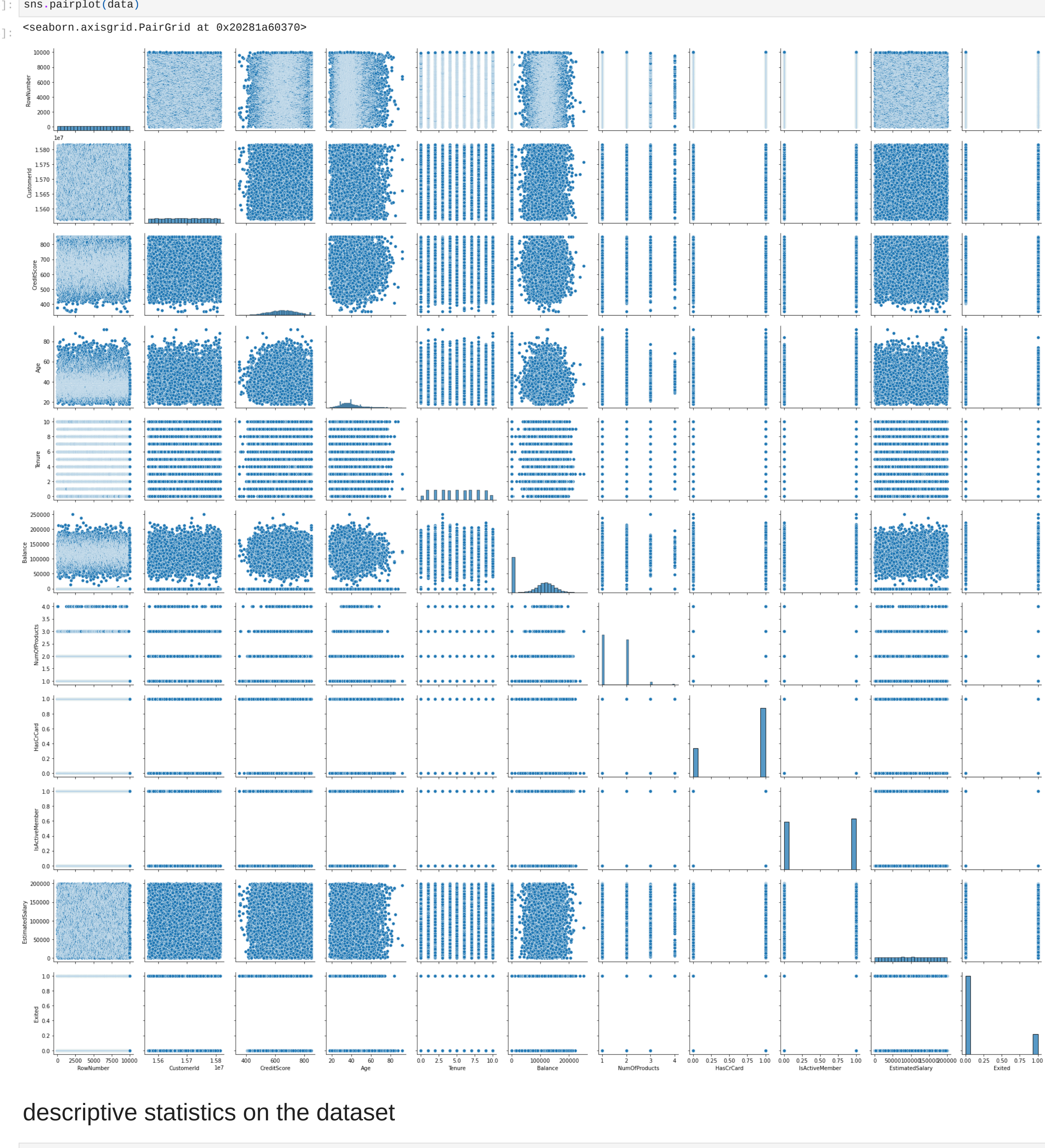
Out[14]: <seaborn.axisgrid.FacetGrid at 0x202c7b87520>



Multivariate Analysis

In [15]:
```python
sns.pairplot(data)
```

Out[15]: <seaborn.axisgrid.PairGrid at 0x20281a60370>



descriptive statistics on the dataset

In [16]:
```python
data=pd.read_csv("D:\madhu\Churn_Modelling.csv")
data.head()
```

Out[16]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

In [17]:
```python
data.tail()
```

Out[17]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
9995	9996	15606229	Obijiaku	771	France	Male	39	5	0.00	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	101699.77	0
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	42085.58	1
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	92888.52	1
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	38190.78	0

In [18]:
```python
data.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
```

In [19]:
```python
#mean
data.mean()
```

```
C:\Users\michael\AppData\Local\Temp\ipykernel_15964\4148090336.py:3: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=N
one') is deprecated; in a future version this will raise TypeError.  Select only valid columns before calling the reduction.
  data.mean()
```

Out[19]:
```
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
```

In [20]:
```python
# median
data.median()
```

```
C:\Users\michael\AppData\Local\Temp\ipykernel_15964\2060464946.py:2: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=N
one') is deprecated; in a future version this will raise TypeError.  Select only valid columns before calling the reduction.
  data.median()
```

Out[20]:
```
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
```

In [21]:
```python
#mode
data.mode()
```

Out[21]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15565701	Smith	850.0	France	Male	37.0	2.0	0.0	1.0	1.0	1.0	24924.92	0.0
1	2	15565706	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	3	15565714	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	4	15565779	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	5	15565796	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9995	9996	15815628	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9996	9997	15815645	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9997	9998	15815656	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9998	9999	15815660	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9999	10000	15815690	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

10000 rows × 14 columns

handle the missing values

In [22]:
```python
data.isna()
```

Out[22]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9995	False	False	False	False	False	False	False	False	False	False	False	False	False	False
9996	False	False	False	False	False	False	False	False	False	False	False	False	False	False
9997	False	False	False	False	False	False	False	False	False	False	False	False	False	False
9998	False	False	False	False	False	False	False	False	False	False	False	False	False	False
9999	False	False	False	False	False	False	False	False	False	False	False	False	False	False

10000 rows × 14 columns

In [24]:
```python
#skewness
data.skew()
```

```
C:\Users\michael\AppData\Local\Temp\ipykernel_15964\683194226.py:3: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=No
ne') is deprecated; in a future version this will raise TypeError.  Select only valid columns before calling the reduction.
  data.skew()
```

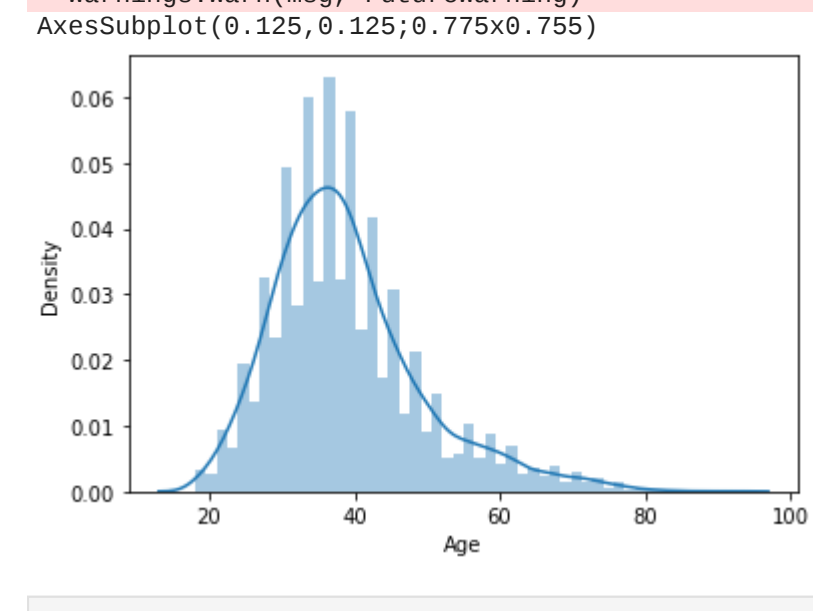
Out[24]:
```
RowNumber          0.000000
CustomerId         0.001149
CreditScore       -0.071607
Age                1.011320
Tenure             0.010991
Balance           -0.141109
NumOfProducts      0.745568
HasCrCard         -0.901812
IsActiveMember    -0.060437
EstimatedSalary    0.002085
Exited             1.471611
dtype: float64
```

In [26]:
```python
data.isna().any()
```

Out[26]:
```
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
```

In [27]:
```python
print(sns.distplot(data['Age']))
```

```
C:\anaconda\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Pleas
e 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)
AxesSubplot(0.125,0.125;0.775x0.755)
```



In [28]:
```python
data.kurt()
```

```
C:\Users\michael\AppData\Local\Temp\ipykernel_15964\2907027414.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=N
one') is deprecated; in a future version this will raise TypeError.  Select only valid columns before calling the reduction.
  data.kurt()
```

Out[28]:
```
RowNumber         -1.200000
CustomerId        -1.196113
CreditScore       -0.425726
Age                1.395347
Tenure            -1.165225
Balance           -1.489412
NumOfProducts      0.582981
HasCrCard         -1.186973
IsActiveMember    -1.996747
EstimatedSalary   -1.181518
Exited             0.165671
dtype: float64
```

In [29]:
```python
data.var()
```

```
C:\Users\michael\AppData\Local\Temp\ipykernel_15964\445316826.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=No
ne') is deprecated; in a future version this will raise TypeError.  Select only valid columns before calling the reduction.
  data.var()
```

Out[29]:
```
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
```

In [30]:
```python
data.std()
```

```
C:\Users\michael\AppData\Local\Temp\ipykernel_15964\2723740006.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=N
one') is deprecated; in a future version this will raise TypeError.  Select only valid columns before calling the reduction.
  data.std()
```

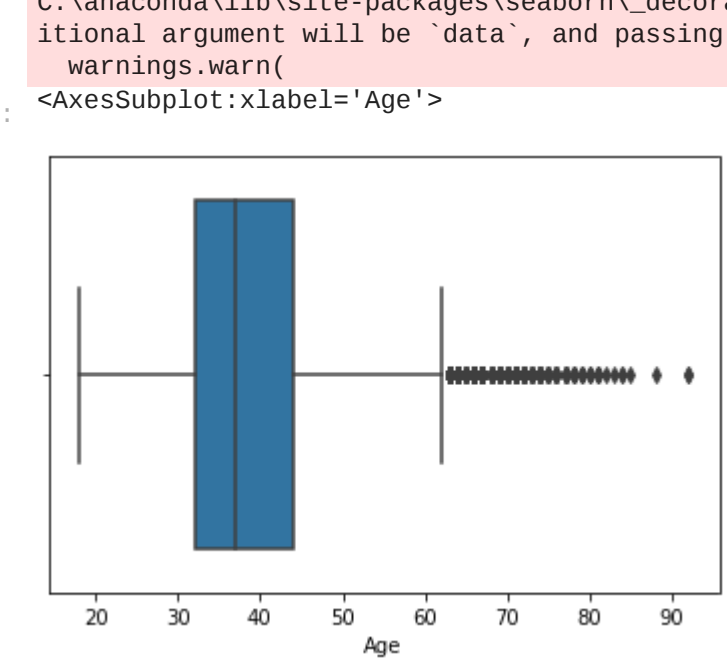
Out[30]:
```
RowNumber           2886.895680
CustomerId         71936.186123
CreditScore           96.653299
Age                   10.487806
Tenure                 2.892174
Balance            62397.405202
NumOfProducts          0.581654
HasCrCard              0.455840
IsActiveMember         0.499797
EstimatedSalary    57510.492818
Exited                 0.402769
dtype: float64
```

find the outlier and replace the outliers

In [31]:
```python
sns.boxplot(data['Age'])
```

```
C:\anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid pos
itional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[31]: <AxesSubplot:xlabel='Age'>



In [32]:
```python
qnt=data.quantile(q=(0.30,0.45))
qnt
```

Out[32]:

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0.30	3000.70	15641363.90	598.7	33.0	3.0	0.000	1.0	1.0	0.0	60736.079	0.0
0.45	4500.55	15678399.85	639.0	36.0	5.0	87621.897	1.0	1.0	0.0	90057.258	0.0

In [33]:
```python
iqr =qnt.loc[0.45]-qnt.loc[0.30] #iqr calculation
iqr
```

Out[33]:
```
RowNumber           1499.850
CustomerId         37035.950
CreditScore           40.300
Age                    3.000
Tenure                 2.000
Balance            87621.897
NumOfProducts          0.000
HasCrCard              0.000
IsActiveMember         0.000
EstimatedSalary    29321.179
Exited                 0.000
dtype: float64
```

In [34]:
```python
#lower extreme values
lower=qnt.loc[0.30]-1.5*iqr
lower
```

Out[34]:
```
RowNumber          7.509250e+02
CustomerId         1.558581e+07
CreditScore        5.382500e+02
Age                2.850000e+01
Tenure             0.000000e+00
Balance           -1.314328e+05
NumOfProducts      1.000000e+00
HasCrCard          1.000000e+00
IsActiveMember     0.000000e+00
EstimatedSalary    1.675431e+04
Exited             0.000000e+00
dtype: float64
```

In [35]:
```python
#upper extreme values
upper=qnt.loc[0.45]+1.5*iqr
upper
```

Out[35]:
```
RowNumber          6.750325e+03
CustomerId         1.573395e+07
CreditScore        6.994500e+02
Age                4.050000e+01
Tenure             8.000000e+00
Balance            2.190547e+05
NumOfProducts      1.000000e+00
HasCrCard          1.000000e+00
IsActiveMember     0.000000e+00
EstimatedSalary    1.340390e+05
Exited             0.000000e+00
dtype: float64
```

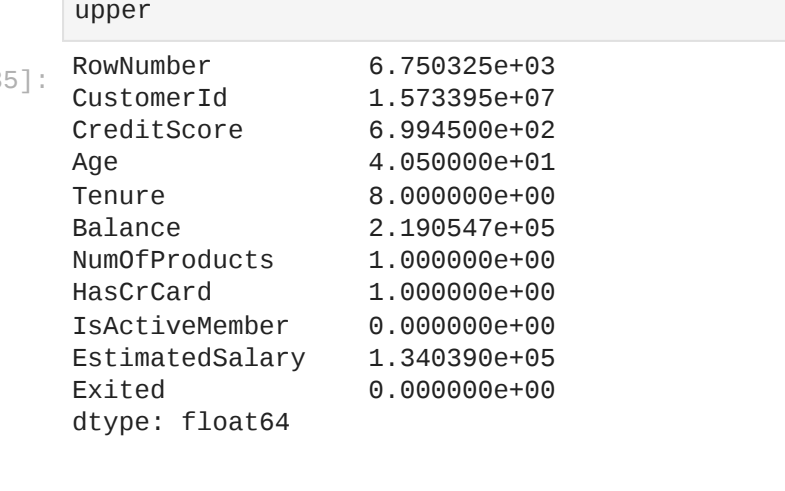
Replacing the outlier

In [130]:
```python
data['Age']=np.where(data['Age']>45,31,data['Age'])
```

In [136]:
```python
sns.boxplot(data['Age'])
```

```
C:\anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid pos
itional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[136]: <AxesSubplot:xlabel='Age'>



Check for Categorical columns and perform encoding.

In [46]:
```python
data=pd.read_csv("D:\madhu\Churn_Modelling.csv")
data.head()
```

Out[46]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

In [47]:
```python
#manually handling the categorical columns
data['Gender'].replace( {'Female':1,'Male':0},inplace=True)
data['Geography'].replace( {'France':1,'Spain':0},inplace=True)
data.head()
```

Out[47]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	1	1	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	0	1	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	1	1	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	1	1	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	0	1	43	2	125510.82	1	1	1	79084.10	0

Split the data into dependent and independent variables.

In [48]:
```python
#target variable
y=data['Balance']
y.head()
```

Out[48]:
```
0         0.00
1     83807.86
2    159660.80
3         0.00
4    125510.82
Name: Balance, dtype: float64
```

In [49]:
```python
#independent
x=data.drop(columns=['Balance'],axis=1)
x.head()
```

Out[49]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	1	1	42	2	1	1	1	101348.88	1
1	2	15647311	Hill	608	0	1	41	1	1	0	1	112542.58	0
2	3	15619304	Onio	502	1	1	42	8	3	1	0	113931.57	1
3	4	15701354	Boni	699	1	1	39	1	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	0	1	43	2	1	1	1	79084.10	0

In [50]:
```python
data=pd.get_dummies(data,columns=['Balance'])
data.head()
```

Out[50]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	NumOfProducts	HasCrCard	...	Balance_212692.97	Balance_212696.32	Balance_212778.2	Balance_213146.2
0	1	15634602	Hargrave	619	1	1	42	2	1	1	...	0	0	0	0
1	2	15647311	Hill	608	0	1	41	1	1	0	...	0	0	0	0
2	3	15619304	Onio	502	1	1	42	8	3	1	...	0	0	0	0
3	4	15701354	Boni	699	1	1	39	1	2	0	...	0	0	0	0
4	5	15737888	Mitchell	850	0	1	43	2	1	1	...	0	0	0	0

5 rows × 6395 columns

Scale the independent variables

In [ ]:
```python
from sklearn.preprocessing import scale
```

In [ ]:
```python
x=scale(x)
```

In [51]:
```python
x
```

Out[51]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	1	1	42	2	1	1	1	101348.88	1
1	2	15647311	Hill	608	0	1	41	1	1	0	1	112542.58	0
2	3	15619304	Onio	502	1	1	42	8	3	1	0	113931.57	1
3	4	15701354	Boni	699	1	1	39	1	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	0	1	43	2	1	1	1	79084.10	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9995	9996	15606229	Obijiaku	771	1	0	39	5	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	516	1	0	35	10	1	1	1	101699.77	0
9997	9998	15584532	Liu	709	1	1	36	7	1	0	1	42085.58	1
9998	9999	15682355	Sabbatini	772	Germany	0	42	3	2	1	0	92888.52	1
9999	10000	15628319	Walker	792	1	1	28	4	1	1	0	38190.78	0

10000 rows × 13 columns

In [52]:
```python
x.mean()
```

```
C:\Users\michael\AppData\Local\Temp\ipykernel_15964\3791599204.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=N
one') is deprecated; in a future version this will raise TypeError.  Select only valid columns before calling the reduction.
  x.mean()
```

Out[52]:
```
RowNumber          5.000500e+03
CustomerId         1.569094e+07
CreditScore        6.505288e+02
Gender             4.543000e-01
Age                3.892180e+01
Tenure             5.012800e+00
NumOfProducts      1.530200e+00
HasCrCard          7.055000e-01
IsActiveMember     5.151000e-01
EstimatedSalary    1.000902e+05
Exited             2.037000e-01
dtype: float64
```

In [53]:
```python
x.std()
```

```
C:\Users\michael\AppData\Local\Temp\ipykernel_15964\2178744708.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=N
one') is deprecated; in a future version this will raise TypeError.  Select only valid columns before calling the reduction.
  x.std()
```

Out[53]:
```
RowNumber           2886.895680
CustomerId         71936.186123
CreditScore           96.653299
Gender                 0.497932
Age                   10.487806
Tenure                 2.892174
NumOfProducts          0.581654
HasCrCard              0.455840
IsActiveMember         0.499797
EstimatedSalary    57510.492818
Exited                 0.402769
dtype: float64
```

Split the data into training and testing

In [261]:
```python
from sklearn.model_selection import train_test_split
```

In [262]:
```python
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=0)
```

In [263]:
```python
x_train.shape
```

Out[263]: (8000, 13)

In [264]:
```python
x_test.shape
```

Out[264]: (2000, 13)

In [265]:
```python
y_train.shape
```

Out[265]: (8000,)

In [266]:
```python
y_test.shape
```

Out[266]: (2000,)

In [ ]: