```
import pandas as pd
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

```
data=pd.read_csv("Churn_Modelling.csv")
```

```
data.head(10)
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Ba
0	1	15634602	Hargrave	619	France	Female	42	2	
1	2	15647311	Hill	608	Spain	Female	41	1	83
2	3	15619304	Onio	502	France	Female	42	8	159
3	4	15701354	Boni	699	France	Female	39	1	
4	5	15737888	Mitchell	850	Spain	Female	43	2	125
5	6	15574012	Chu	645	Spain	Male	44	8	113
6	7	15592531	Bartlett	822	France	Male	50	7	
7	8	15656148	Obinna	376	Germany	Female	29	4	115
8	9	15792365	He	501	France	Male	44	4	142
9	10	15592389	H?	684	France	Male	27	2	134

```
data.tail(10)
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenu
9990	9991	15798964	Nkemakonam	714	Germany	Male	33	
9991	9992	15769959	Ajuluchukwu	597	France	Female	53	
9992	9993	15657105	Chukwualuka	726	Spain	Male	36	

```
#describe statistics
data.describe()
```

	RowNumber	CustomerId	CreditScore	Age	Tenure	Bala
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000
mean	5000.50000	1.569094e+07	650.528800	38.921800	5.012800	76485.889
std	2886.89568	7.193619e+04	96.653299	10.487806	2.892174	62397.405
min	1.00000	1.556570e+07	350.000000	18.000000	0.000000	0.000
25%	2500.75000	1.562853e+07	584.000000	32.000000	3.000000	0.000
50%	5000.50000	1.569074e+07	652.000000	37.000000	5.000000	97198.540
75%	7500.25000	1.575323e+07	718.000000	44.000000	7.000000	127644.240
max	10000.00000	1.581569e+07	850.000000	92.000000	10.000000	250898.090

```
data.kurt(axis=0,skipna=True)
```

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

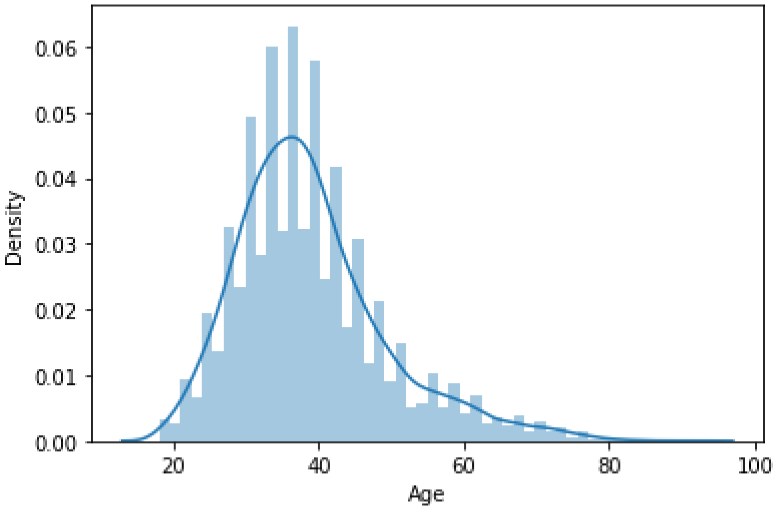
```
data.kurt(axis=1,skipna=True)
```

```
0       10.998778
1       10.997909
2       10.995886
3       10.998962
4       10.997675
          ...
9995    10.998908
9996    10.998551
```

```
9997    10.999788
9998    10.998530
9999    10.997973
Length: 10000, dtype: float64
```

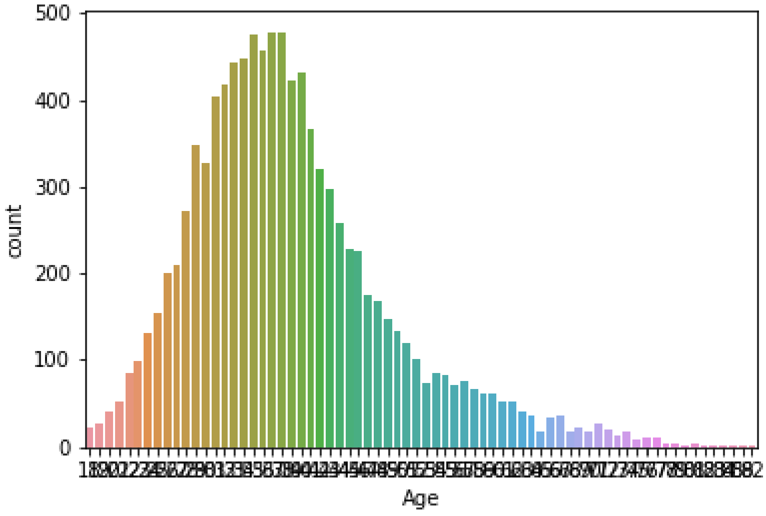
```
sns.distplot(data['Age'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbde5f7eb10>
```



```
sns.countplot(data["Age"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbde5969ad0>
```



```
data.skew(axis=0,skipna=True)
```

```
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
```

```
data.skew(axis=1,skipna=True)
```

```
     0       3.316373
     1       3.316193
     2       3.315777
     3       3.316411
     4       3.316145
               ...
     9995    3.316399
     9996    3.316325
     9997    3.316581
     9998    3.316321
     9999    3.316207
     Length: 10000, dtype: float64
```

```
data.isnull().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
```

```
data.isnull().sum()
```

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

```
data.duplicated()
```

```
     0       False
     1       False
```

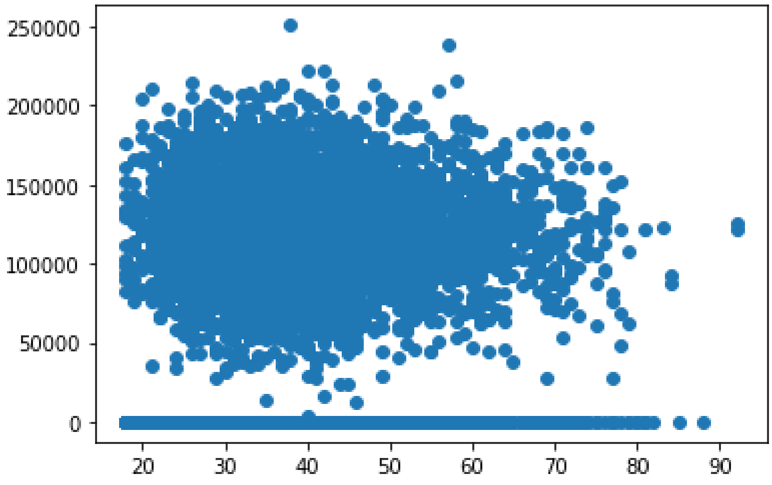
```
2       False
3       False
4       False
        ...
9995    False
9996    False
9997    False
9998    False
9999    False
Length: 10000, dtype: bool
```

```
data.duplicated().sum()
```

```
0
```

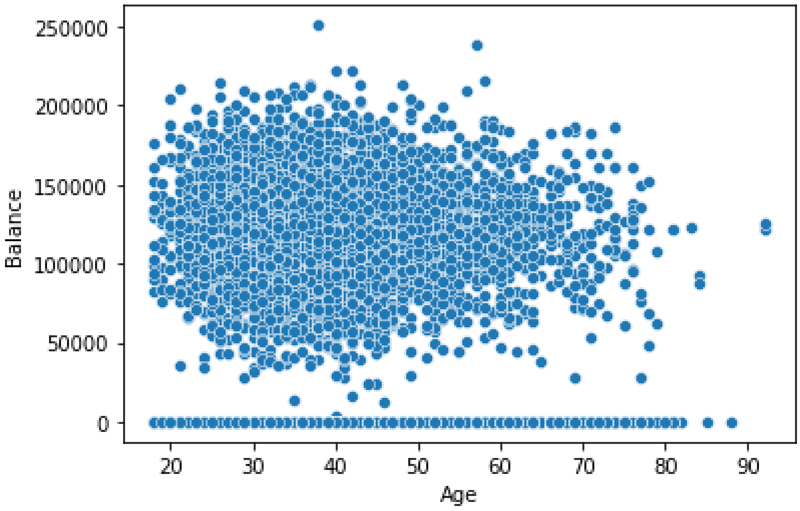
```
plt.scatter(data.Age,data.Balance)
```

```
<matplotlib.collections.PathCollection at 0x7fbde572c9d0>
```



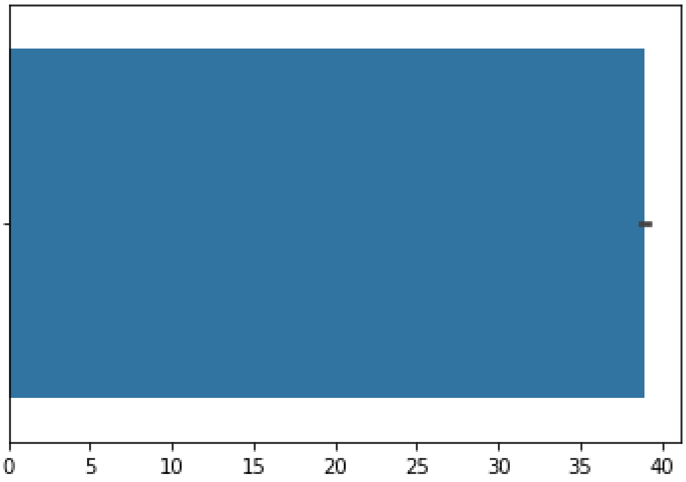
```
sns.scatterplot(data.Age,data.Balance)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbde5692590>
```



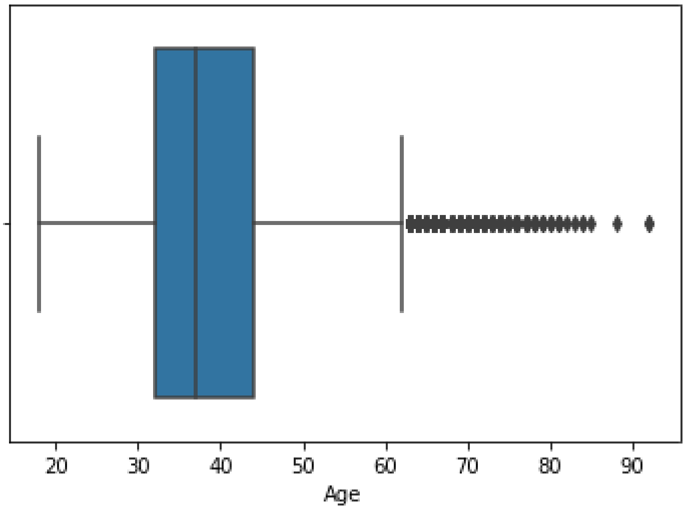
```
sns.barplot(data['Age'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbde567a790>
```



```
sns.boxplot(data['Age'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbde55d7050>
```

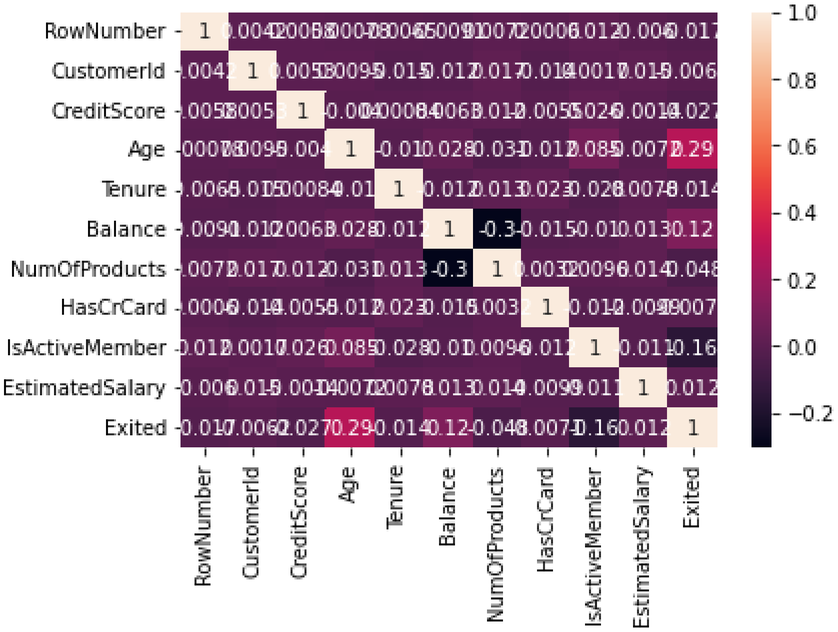


```
data.corr()
```

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	N
RowNumber	1.000000	0.004202	0.005840	0.000783	-0.006495	-0.009067	

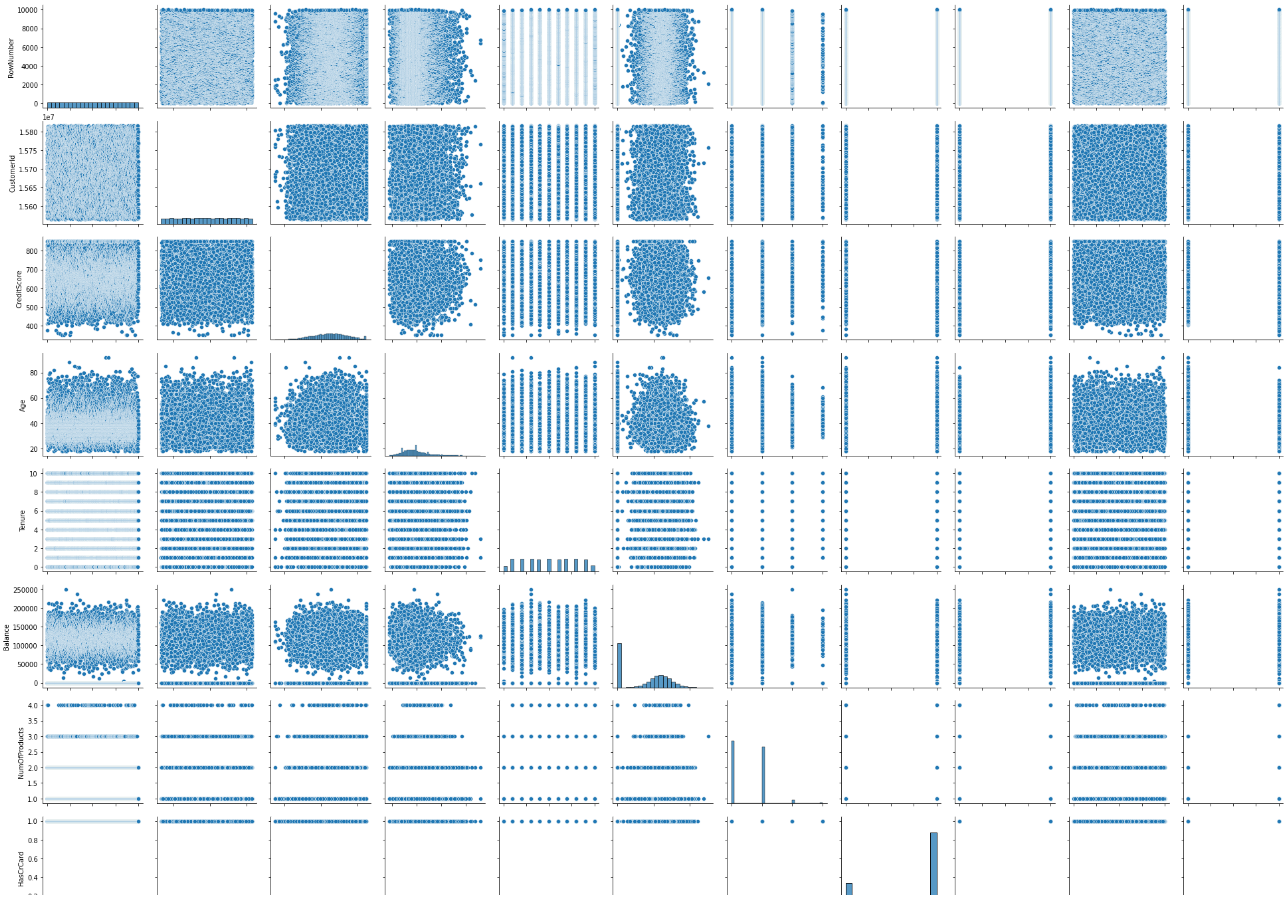
```
sns.heatmap(data.corr(),annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbde55d0650>
```



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

```
<seaborn.axisgrid.PairGrid at 0x7fbde2b92f10>
```



```
from scipy.stats import spearmanr
```

```
corr=spearmanr(data)
corr
```

```
       [8.55178468e-001, 5.95071292e-001, 0.00000000e+000,
        5.03861020e-001, 2.33332702e-002, 8.30304249e-001,
        8.90508036e-001, 8.74364323e-002, 9.36216720e-001,
        8.39475437e-002, 3.71469037e-001, 8.90503148e-001,
        2.38243578e-001, 2.72106138e-001],
       [6.07981798e-001, 5.50722932e-001, 5.03861020e-001,
        0.00000000e+000, 5.41558890e-001, 7.63332662e-001,
        4.25266703e-001, 9.09790109e-001, 5.69634028e-001,
        2.08874884e-001, 7.03844602e-001, 1.52541637e-002,
        9.01602674e-001, 1.98609526e-002],
       [3.11698199e-001, 5.46203060e-001, 2.33332702e-002,
        5.41558890e-001, 0.00000000e+000, 8.37437458e-001,
        4.06537979e-004, 7.06678685e-001, 2.01668047e-023,
        9.38702072e-001, 4.70093788e-001, 6.57076013e-001,
        9.84458653e-001, 1.08256524e-007],
       [6.88261457e-002, 7.93006618e-001, 8.30304249e-001,
        7.63332662e-001, 8.37437458e-001, 0.00000000e+000,
        2.89407525e-003, 1.31173411e-001, 1.76909716e-001,
        1.98811127e-001, 5.64246762e-001, 2.41686809e-002,
        4.08370570e-001, 1.25850456e-026],
       [9.62034639e-001, 3.80283664e-001, 8.90508036e-001,
        4.25266703e-001, 4.06537979e-004, 2.89407525e-003,
        0.00000000e+000, 2.98157345e-001, 8.65526378e-004,
        4.60240532e-009, 1.26581605e-001, 6.74797620e-005,
        8.07912562e-001, 4.60367975e-243],
       [4.88086885e-001, 1.31785022e-001, 8.74364323e-002,
        9.09790109e-001, 7.06678685e-001, 1.31173411e-001,
        2.98157345e-001, 0.00000000e+000, 3.41506861e-001,
        1.96808492e-001, 2.53904935e-002, 4.13650739e-003,
        4.36732384e-001, 1.62203448e-001],
       [3.67465405e-001, 1.63585747e-001, 9.36216720e-001,
        5.69634028e-001, 2.01668047e-023, 1.76909716e-001,
        8.65526378e-004, 3.41506861e-001, 0.00000000e+000,
        1.12319427e-231, 3.25429744e-001, 2.50330560e-001,
        2.38918636e-001, 7.64706959e-029],
       [4.06300660e-001, 5.36514208e-002, 8.39475437e-002,
        2.08874884e-001, 9.38702072e-001, 1.98811127e-001,
        4.60240532e-009, 1.96808492e-001, 1.12319427e-231,
        0.00000000e+000, 6.99615740e-001, 1.03295766e-001,
        2.08799333e-001, 2.85374243e-036],
       [9.52261425e-001, 1.60847582e-001, 3.71469037e-001,
        7.03844602e-001, 4.70093788e-001, 5.64246762e-001,
        1.26581605e-001, 2.53904935e-002, 3.25429744e-001,
        6.99615740e-001, 0.00000000e+000, 2.35441825e-001,
        3.15383179e-001, 4.75414918e-001],
       [2.28461236e-001, 8.66447868e-001, 8.90503148e-001,
        1.52541637e-002, 6.57076013e-001, 2.41686809e-002,
        6.74797620e-005, 4.13650739e-003, 2.50330560e-001,
        1.03295766e-001, 2.35441825e-001, 0.00000000e+000,
        2.51464473e-001, 1.34826852e-055],
       [5.48097586e-001, 1.27389774e-001, 2.38243578e-001,
        9.01602674e-001, 9.84458653e-001, 4.08370570e-001,
        8.07912562e-001, 4.36732384e-001, 2.38918636e-001,
        2.08799333e-001, 3.15383179e-001, 2.51464473e-001,
        0.00000000e+000, 2.27067756e-001],
       [9.75106276e-002, 5.31116466e-001, 2.72106138e-001,
        1.98609526e-002, 1.08256524e-007, 1.25850456e-026,
        4.60367975e-243, 1.62203448e-001, 7.64706959e-029,
```

```
import statsmodels.api as sm
```

```
x=data[["EstimatedSalary"]]
y=data["CreditScore"]
```

```
model=sm.OLS(y,x)
result=model.fit()
result.summary()
```

OLS Regression Results

Dep. Variable:	CreditScore	**R-squared (uncentered):**	0.735
Model:	OLS	**Adj. R-squared (uncentered):**	0.735
Method:	Least Squares	**F-statistic:**	2.779e+04
Date:	Wed, 09 Nov 2022	**Prob (F-statistic):**	0.00
Time:	04:26:15	**Log-Likelihood:**	-72429.
No. Observations:	10000	**AIC:**	1.449e+05
Df Residuals:	9999	**BIC:**	1.449e+05

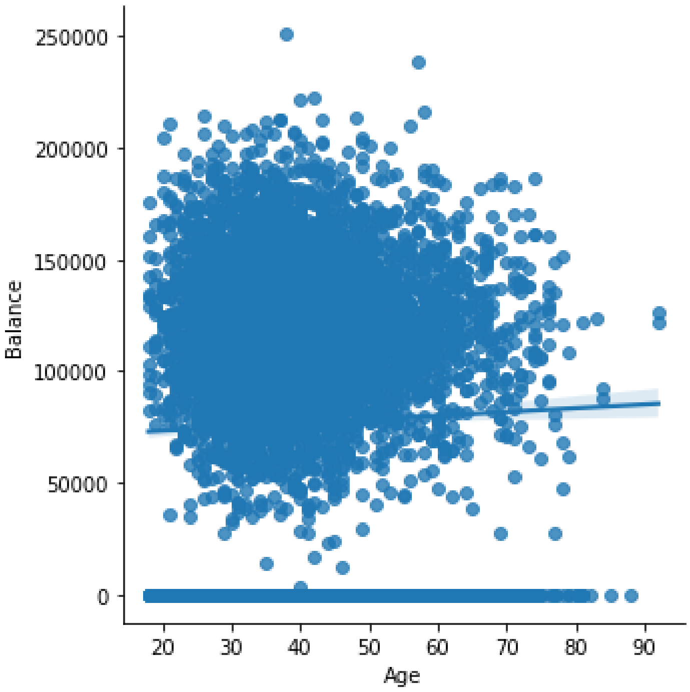
```
from sklearn.preprocessing import scale
x=scale(x)
x
```

```
array([[ 0.02188649],
       [ 0.21653375],
       [ 0.2406869 ],
       ...,
       [-1.00864308],
       [-0.12523071],
       [-1.07636976]])
```

[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.

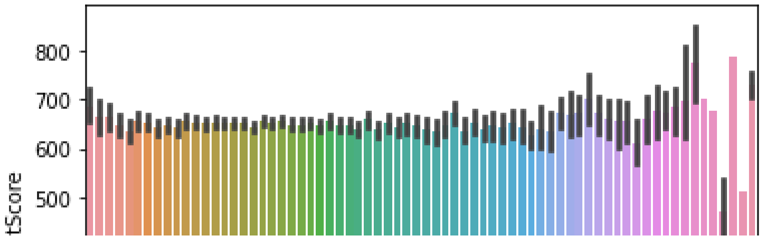
```
sns.lmplot(x='Age',y='Balance',data=data)
```

```
<seaborn.axisgrid.FacetGrid at 0x7fbdd31031d0>
```



```
sns.barplot(x="Age",y="CreditScore",data=data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbdd2ff7550>
```



```
qnt = data.quantile(q=[0.75,0.25])
qnt
```

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	Ha
0.75	7500.25	15753233.75	718.0	44.0	7.0	127644.24	2.0	
0.25	2500.75	15628528.25	584.0	32.0	3.0	0.00	1.0	

```
iqr=qnt.loc[0.75]-qnt.loc[0.25]
iqr
```

```
RowNumber            4999.5000
CustomerId         124705.5000
CreditScore           134.0000
Age                    12.0000
Tenure                  4.0000
Balance            127644.2400
NumOfProducts           1.0000
HasCrCard               1.0000
IsActiveMember          1.0000
EstimatedSalary     98386.1375
Exited                  0.0000
dtype: float64
```

```
upper= qnt.loc[0.75]+1.5*iqr
upper
```

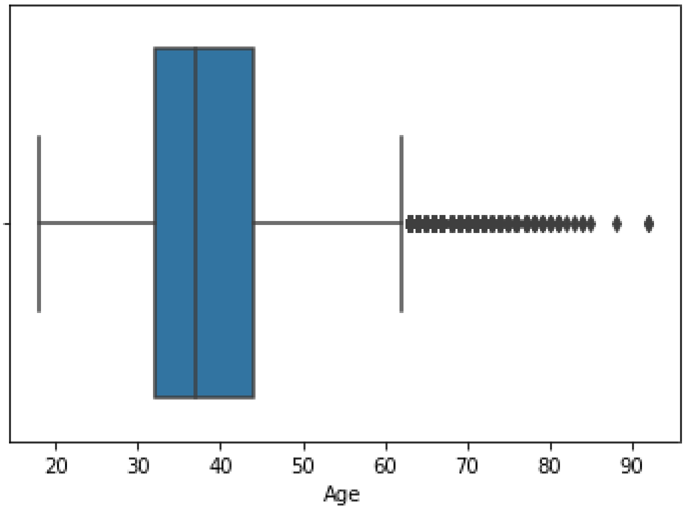
```
RowNumber          1.499950e+04
CustomerId         1.594029e+07
CreditScore        9.190000e+02
Age                6.200000e+01
Tenure             1.300000e+01
Balance            3.191106e+05
NumOfProducts      3.500000e+00
HasCrCard          2.500000e+00
IsActiveMember     2.500000e+00
EstimatedSalary    2.969675e+05
Exited             0.000000e+00
dtype: float64
```

```
lower= qnt.loc[0.25]-1.5*iqr
lower
```

```
RowNumber         -4.998500e+03
CustomerId         1.544147e+07
CreditScore        3.830000e+02
Age                1.400000e+01
Tenure            -3.000000e+00
Balance           -1.914664e+05
NumOfProducts     -5.000000e-01
HasCrCard         -1.500000e+00
IsActiveMember    -1.500000e+00
EstimatedSalary   -9.657710e+04
Exited             0.000000e+00
dtype: float64
```

```
sns.boxplot(data["Age"])
```

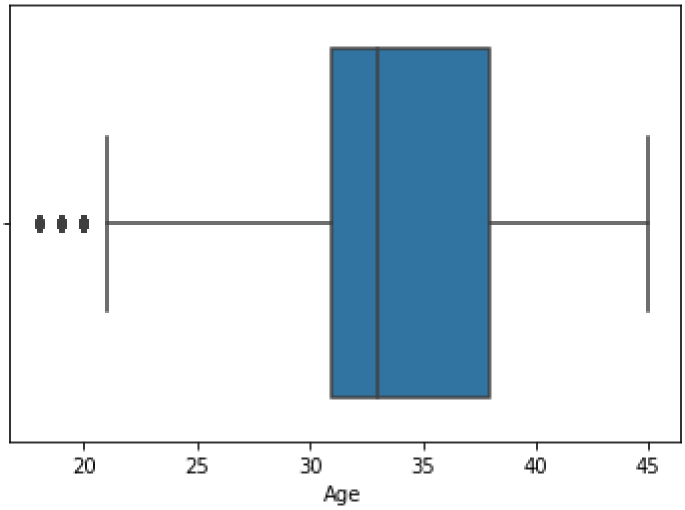
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbdd2cfd610>
```



```
data["Age"]= np.where(data["Age"]>45,31,data["Age"])
```

```
sns.boxplot(data["Age"])
```

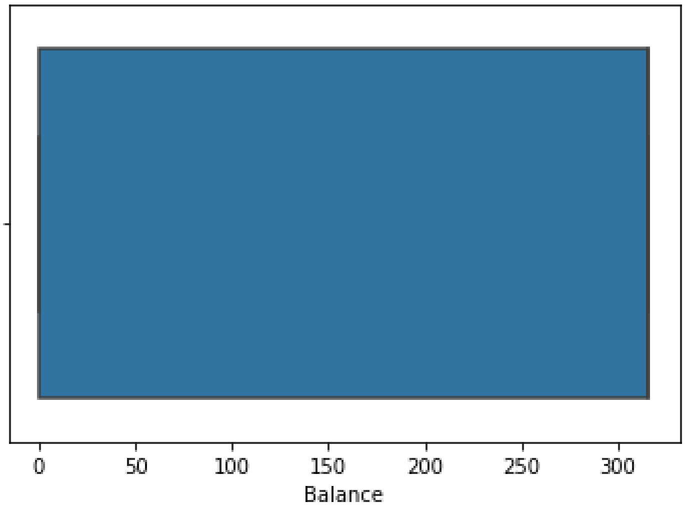
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbdd2cd1b50>
```



```
data["Balance"]= np.where(data["Balance"]>618,316,data["Balance"])
```

```
sns.boxplot(data["Balance"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbdd2c33810>
```



```
data.head()
```

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

```
data.head(10)
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Bal
0	1	15634602	Hargrave	619	France	0	42	2	
1	2	15647311	Hill	608	Spain	0	41	1	3
2	3	15619304	Onio	502	France	0	42	8	3
3	4	15701354	Boni	699	France	0	39	1	
4	5	15737888	Mitchell	850	Spain	0	43	2	3
5	6	15574012	Chu	645	Spain	1	44	8	3
6	7	15592531	Bartlett	822	France	1	31	7	
7	8	15656148	Obinna	376	Germany	0	29	4	3
8	9	15792365	He	501	France	1	44	4	3
9	10	15592389	H?	684	France	1	27	2	3

```
data["HasCrCard"].replace({1:"yes",0:"no"},inplace = True)
```

```
data.head(10)
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Bal
0	1	15634602	Hargrave	619	France	0	42	2	
1	2	15647311	Hill	608	Spain	0	41	1	3
2	3	15619304	Onio	502	France	0	42	8	3
3	4	15701354	Boni	699	France	0	39	1	
4	5	15737888	Mitchell	850	Spain	0	43	2	3
5	6	15574012	Chu	645	Spain	1	44	8	3
6	7	15592531	Bartlett	822	France	1	31	7	
7	8	15656148	Obinna	376	Germany	0	29	4	3
8	9	15792365	He	501	France	1	44	4	3
9	10	15592389	H?	684	France	1	27	2	3

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

```
data["Age"]=le.fit_transform(data["Age"])
```

```
data.Age.unique()
```

```
array([24, 23, 21, 25, 26, 13, 11,  9,  6, 16,  7, 17, 27, 14, 20, 18, 15,
       22, 19,  1,  8,  3,  4, 12, 10,  2,  5,  0])
```

```
x=data.iloc[:,0:13].values
x
```

```
array([[1, 15634602, 'Hargrave', ..., 'yes', 1, 101348.88],
       [2, 15647311, 'Hill', ..., 'no', 1, 112542.58],
       [3, 15619304, 'Onio', ..., 'yes', 0, 113931.57],
       ...,
       [9998, 15584532, 'Liu', ..., 'no', 1, 42085.58],
       [9999, 15682355, 'Sabbatini', ..., 'yes', 0, 92888.52],
       [10000, 15628319, 'Walker', ..., 'yes', 0, 38190.78]], dtype=object)
```

```
y=data.iloc[:,13:14].values
y
```

```
array([[1],
       [0],
       [1],
       ...,
       [1],
       [1],
       [0]])
```

```
data.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Bal
0	1	15634602	Hargrave	619	France	0	24	2	
1	2	15647311	Hill	608	Spain	0	23	1	3
2	3	15619304	Onio	502	France	0	24	8	3
3	4	15701354	Boni	699	France	0	21	1	
4	5	15737888	Mitchell	850	Spain	0	25	2	3

```
from sklearn.preprocessing import  OneHotEncoder
```

```
ohe= OneHotEncoder()
```

```
z=ohe.fit_transform(x[:,0:14]).toarray()
z
```

```
array([[1., 0., 0., ..., 0., 0., 0.],
       [0., 1., 0., ..., 0., 0., 0.],
       [0., 0., 1., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]])
```

```
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,random_state=0)
```

```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

```
((8000, 13), (2000, 13), (8000, 1), (2000, 1))
```

```
x_train
```

```
array([[7390, 15676909, 'Mishin', ..., 'yes', 0, 163830.64],
       [9276, 15749265, 'Carslaw', ..., 'yes', 1, 57098.0],
       [2996, 15582492, 'Moore', ..., 'yes', 0, 185630.76],
       ...,
       [3265, 15574372, 'Hoolan', ..., 'yes', 0, 181429.87],
       [9846, 15664035, 'Parsons', ..., 'yes', 1, 148750.16],
       [2733, 15592816, 'Udokamma', ..., 'yes', 0, 118855.26]],
      dtype=object)
```

```
x_test
```

```
array([[9395, 15615753, 'Upchurch', ..., 'yes', 1, 192852.67],
       [899, 15654700, 'Fallaci', ..., 'yes', 0, 128702.1],
       [2399, 15633877, 'Morrison', ..., 'yes', 1, 75732.25],
       ...,
       [9550, 15772604, 'Chiemezie', ..., 'yes', 0, 141533.19],
       [2741, 15787699, 'Burke', ..., 'yes', 1, 11276.48],
       [6691, 15579223, 'Niu', ..., 'yes', 0, 192950.6]], dtype=object)
```

```
y_train
```

```
array([[0],
       [0],
       [0],
       ...,
       [0],
       [0],
       [1]])
```

```
y_test
```

```
array([[0],
       [1],
       [0],
       ...,
       [0],
       [0],
       [0]])
```

```
from sklearn.preprocessing import scale
```

```
x=data["CreditScore"]
S=scale(x)
S
```

```
array([-0.32622142, -0.44003595, -1.53679418, ...,  0.60498839,
        1.25683526,  1.46377078])
```

```
y=data["Age"]
y
```

```
0       24
1       23
2       24
3       21
4       25
        ..
9995    21
9996    17
9997    18
9998    24
```

```
    9999    10
    Name: Age, Length: 10000, dtype: int64
```

```
x=data.drop(data["Age"],axis=0)
x
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Ter
28	29	15728693	McWilliams	574	Germany	0	25	
29	30	15656300	Lucciano	411	France	1	11	
30	31	15589475	Azikiwe	591	Spain	0	21	
31	32	15706552	Odinakachukwu	533	France	1	18	
32	33	15750181	Sanderson	553	Germany	1	23	
...	...	...	...	...	...	...	...	
9995	9996	15606229	Obijiaku	771	France	1	21	
9996	9997	15569892	Johnstone	516	France	1	17	
9997	9998	15584532	Liu	709	France	0	18	
9998	9999	15682355	Sabbatini	772	Germany	1	24	
9999	10000	15628319	Walker	792	France	0	10	

9972 rows × 14 columns

```
y=data.iloc[:,-1].values
y
```

```
    array([1, 0, 1, ..., 1, 1, 0])
```

```
data=pd.DataFrame({"Age":[1,2,np.nan],"CreditScore":[1,np.nan,np.nan],"Balance":[1,2,3]})
data
```

	Age	CreditScore	Balance
0	1.0	1.0	1
1	2.0	NaN	2
2	NaN	NaN	3

```
data.isnull().any()
```

```
    Age             True
    CreditScore     True
    Balance        False
    dtype: bool
```

```
data.isnull().sum()
```

```
Age            1
CreditScore    2
Balance        0
dtype: int64
```

```
data.dropna()
```

	Age	CreditScore	Balance
0	1.0	1.0	1

```
data.dropna(axis=1)
```

	Balance
0	1
1	2
2	3

```
data["Age"].mean()
```

```
1.5
```

✓ 0s completed at 10:15 AM ● ×