

#TEAM ID: PNT2022TMID04039

## Data Visualization and Pre-Processing

#Import the necessary libraries

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

### 2.Download and Load the dataset

```
data = pd.read_csv(r"/content/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

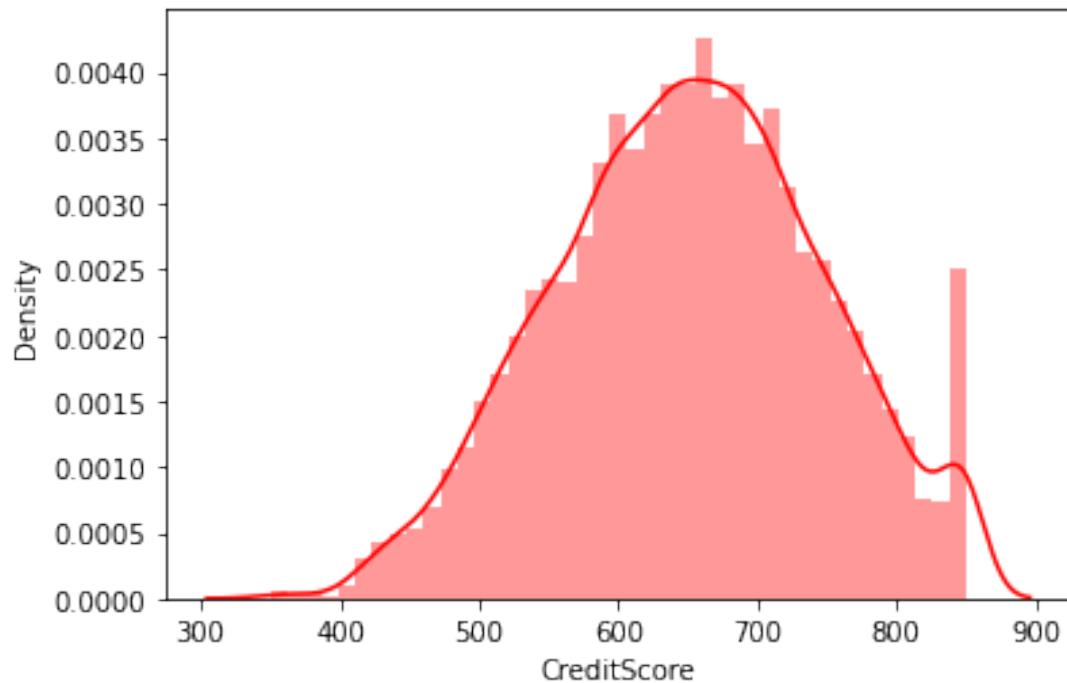
	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	\
0	2	0.00	1	1	1	
1	1	83807.86	1	0	1	
2	8	159660.80	3	1	0	
3	1	0.00	2	0	0	
4	2	125510.82	1	1	1	

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

### 3)(a)Uni-variate Analysis

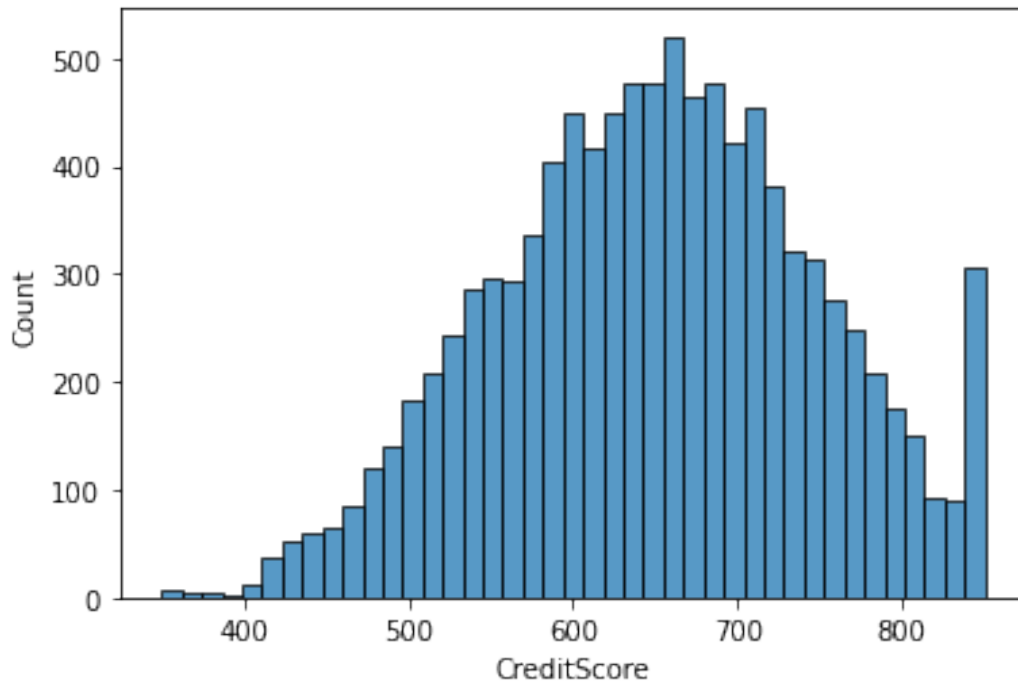
```
sns.distplot(data['CreditScore'],color="r")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f08cfbfce10>
```



```
sns.histplot(data['CreditScore'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f08d894ccd0>
```

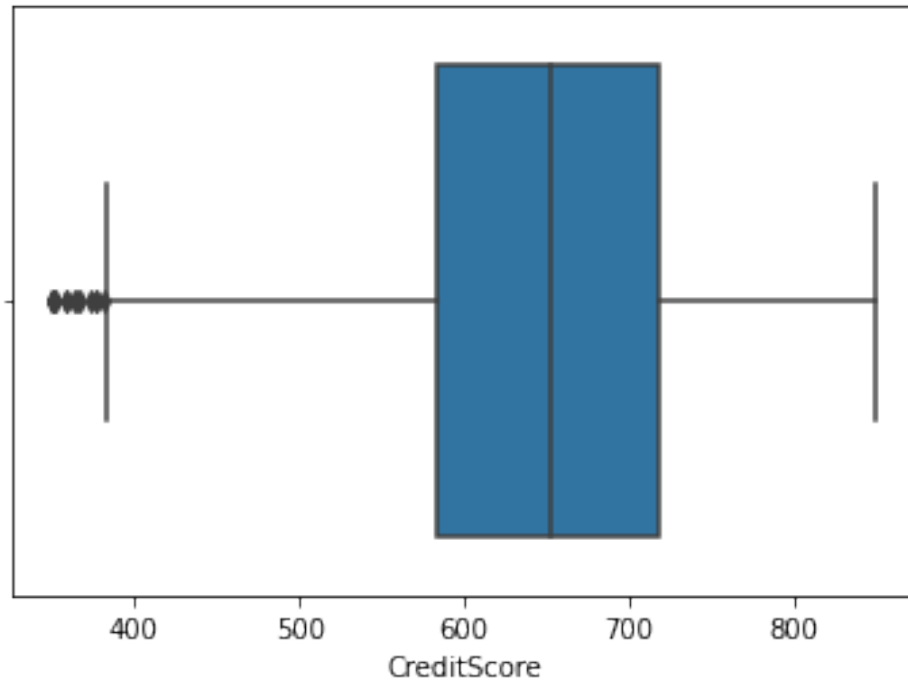


```
sns.boxplot(data['CreditScore'])
```

```
/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 0x7f08d8403350>
```

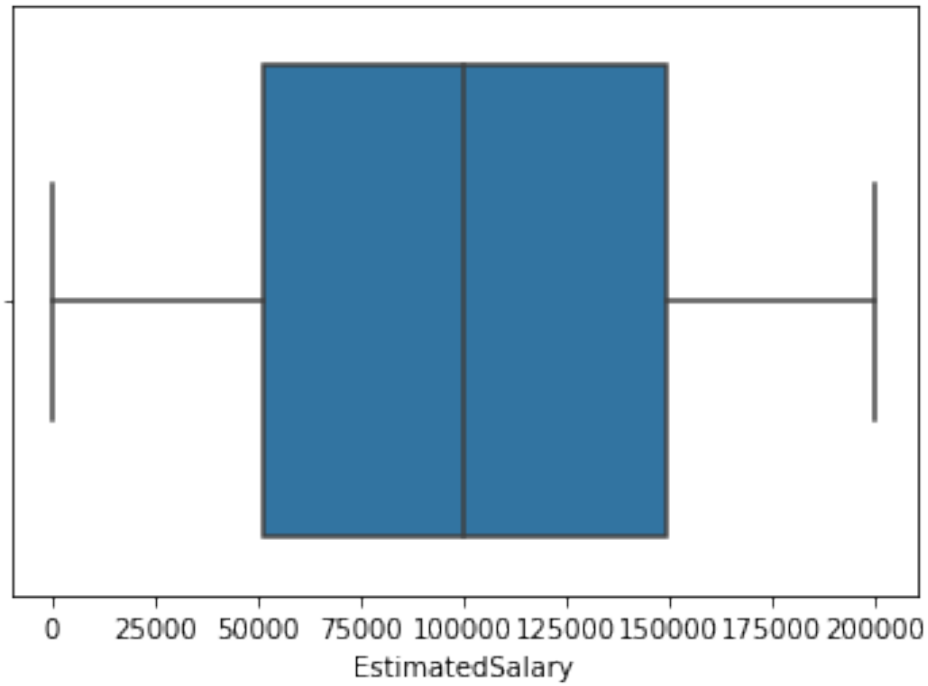


```
sns.boxplot(data['EstimatedSalary'])
```

```
/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
```

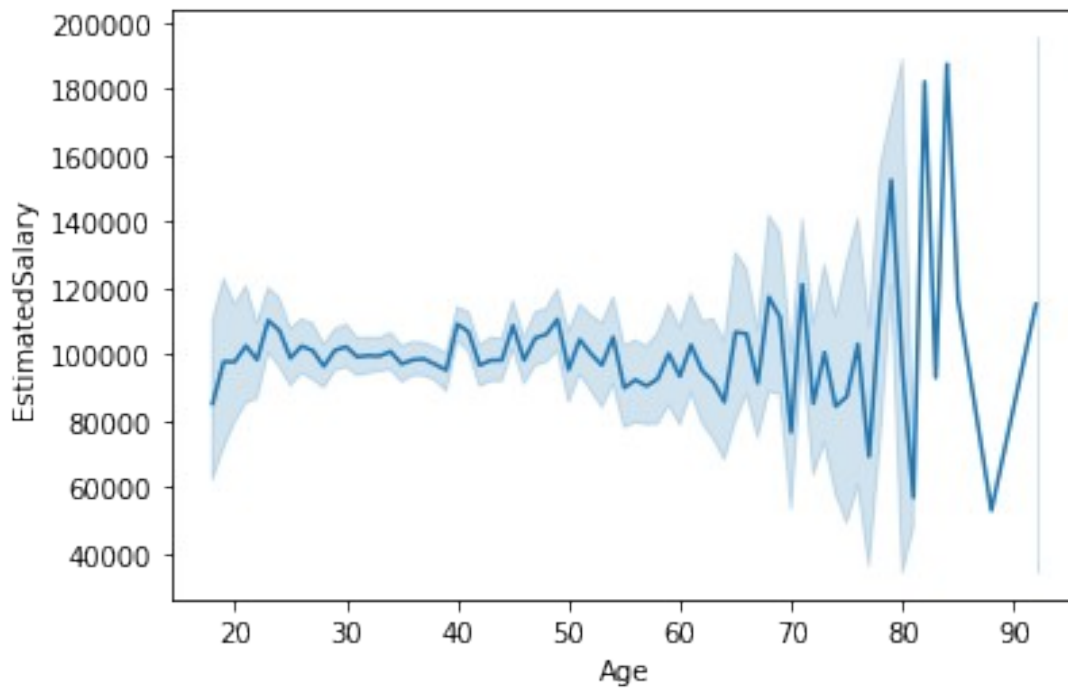
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f08d83791d0>
```



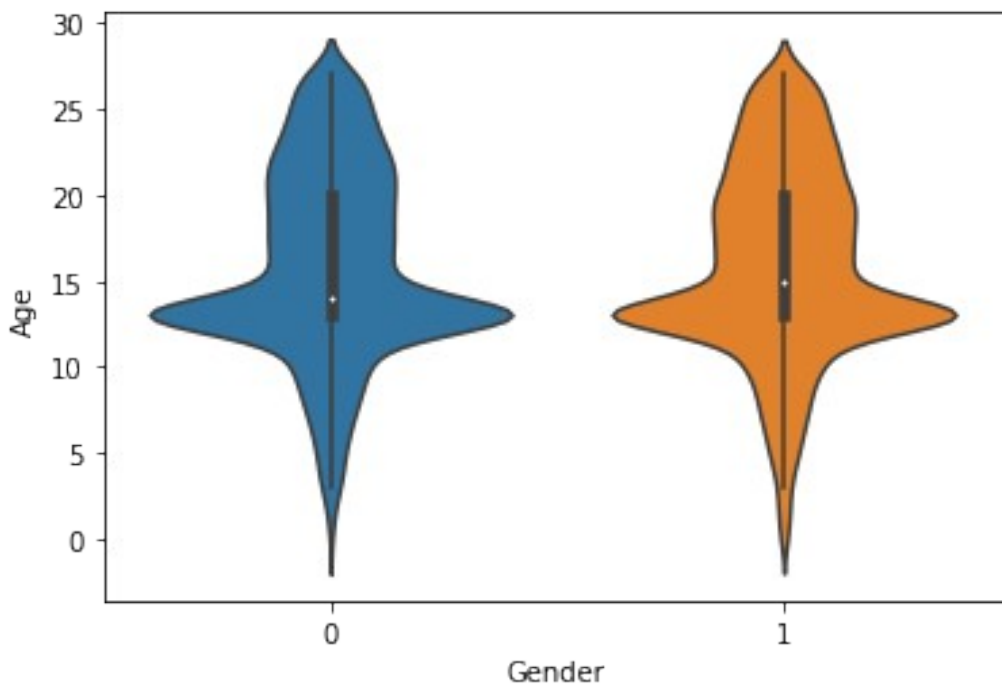
### 3)(B)Bi-variate Analysis

```
import warnings
warnings.filterwarnings("ignore")

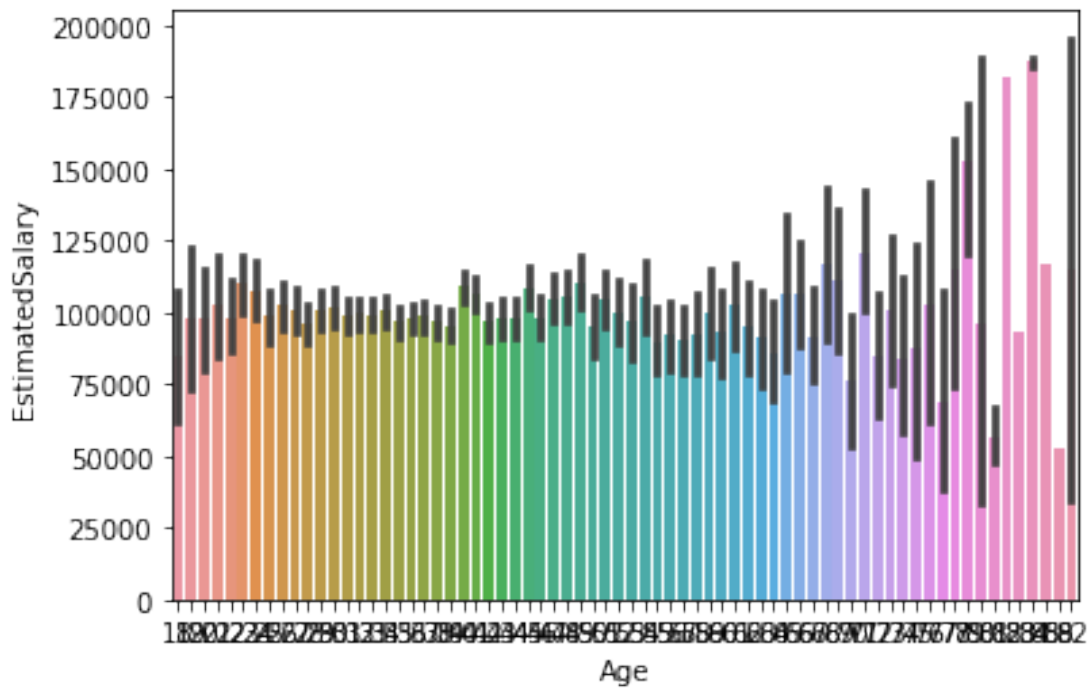
sns.lineplot(data['Age'], data['EstimatedSalary'])
<matplotlib.axes._subplots.AxesSubplot at 0x7f08d82f8290>
```



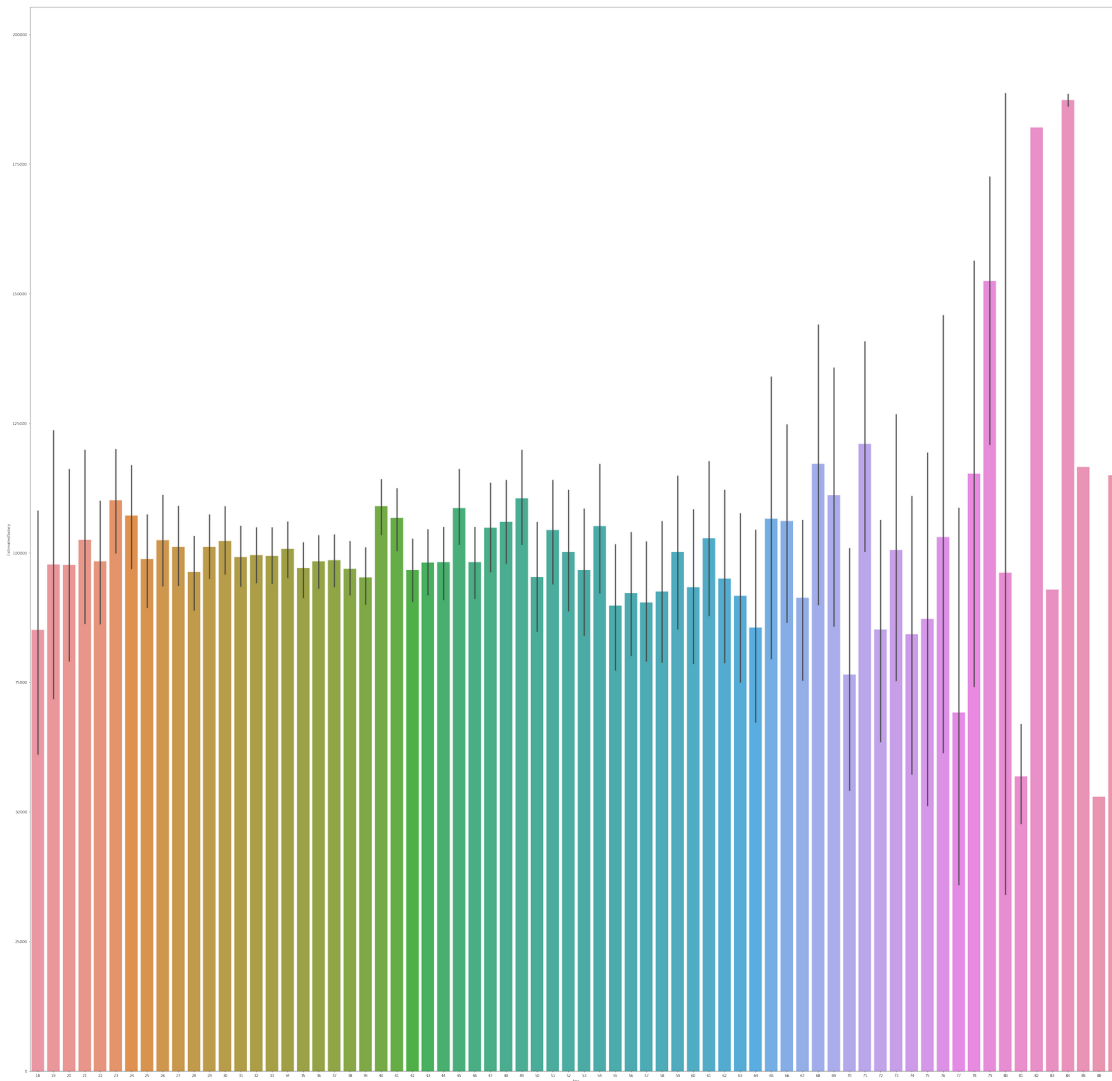
```
import seaborn as sns
sns.violinplot(y = data['Age'], x = data['Gender'])
<matplotlib.axes._subplots.AxesSubplot at 0x7f08c7535990>
```



```
sns.barplot(data['Age'], data['EstimatedSalary'])
<matplotlib.axes._subplots.AxesSubplot at 0x7f08d82e0bd0>
```

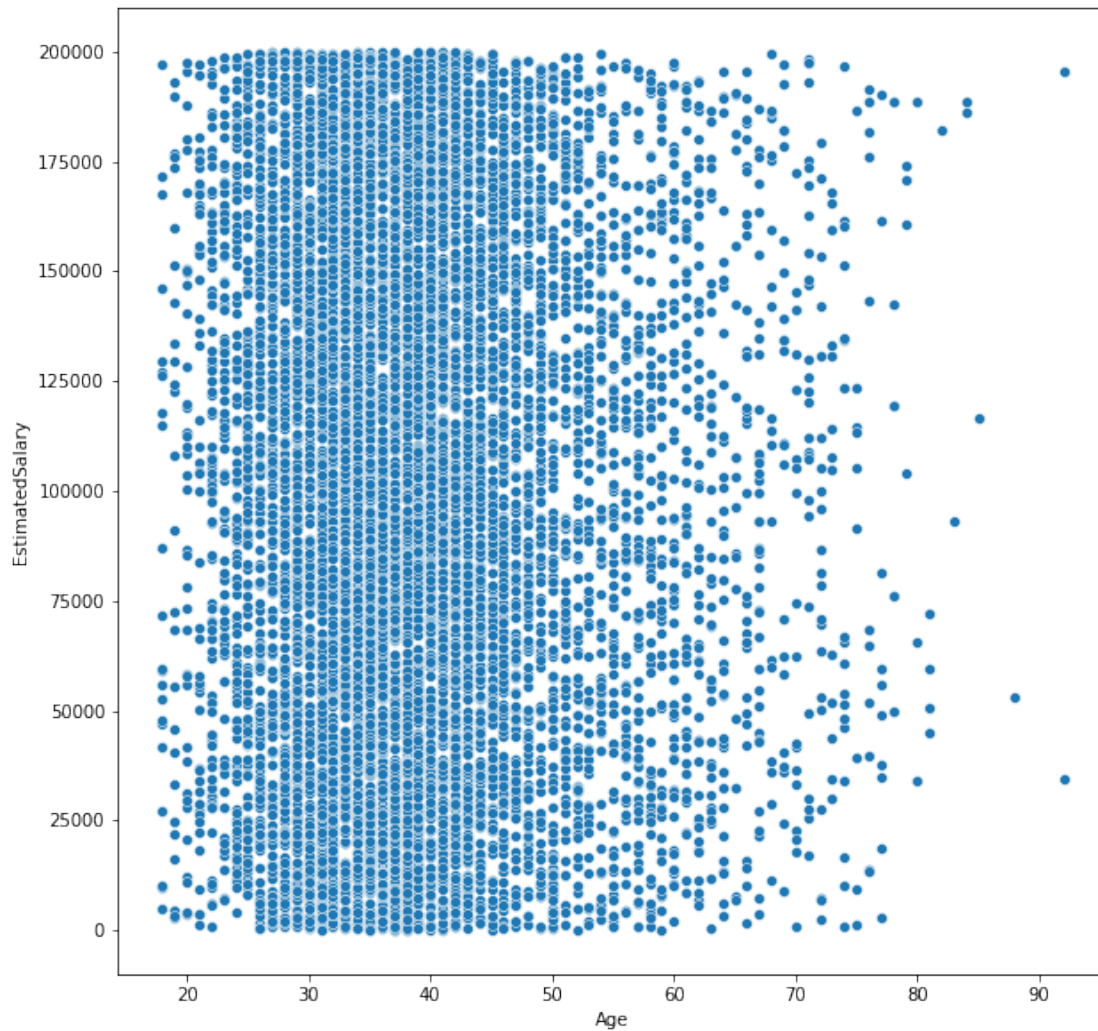


```
plt.figure(figsize=(50,50))
sns.barplot(data['Age'], data['EstimatedSalary'])
<matplotlib.axes._subplots.AxesSubplot at 0x7f08d7fd5e90>
```



```
plt.figure(figsize=(10,10))
sns.scatterplot(data['Age'], data['EstimatedSalary'])
<matplotlib.axes._subplots.AxesSubplot at 0x7f08d7fdb450>
```

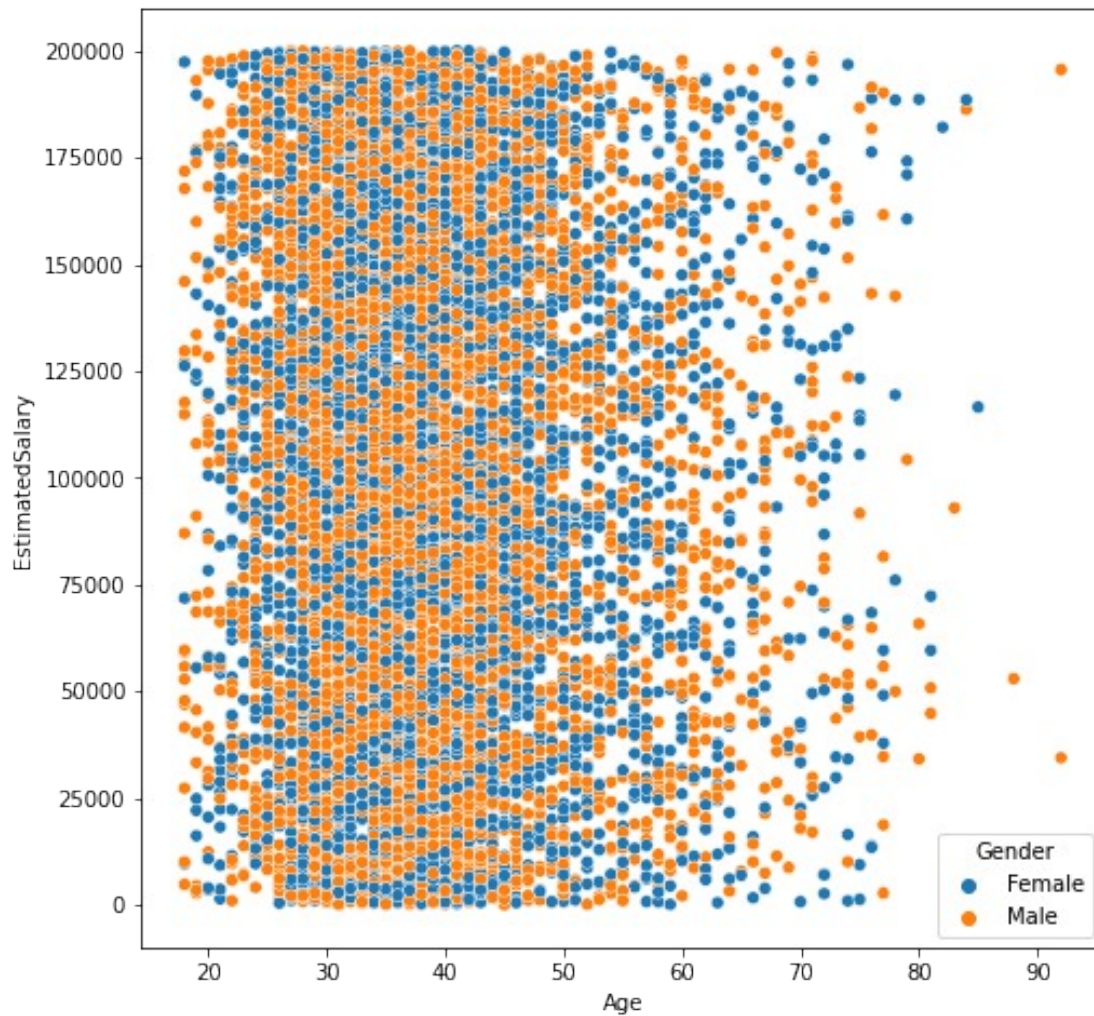




### 3)(C)Multi-Variate Analysis

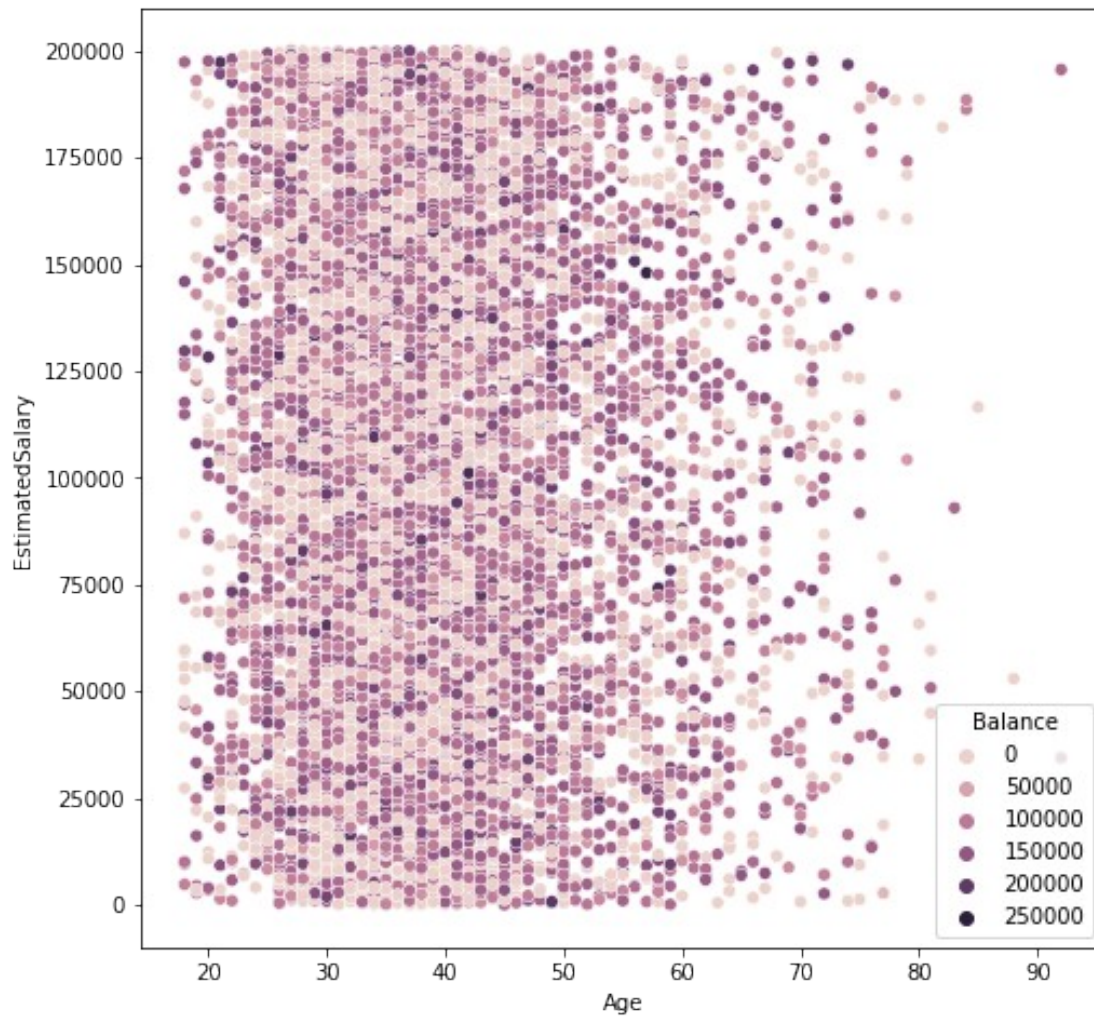
```
plt.figure(figsize=(8,8))  
sns.scatterplot(data['Age'], data['EstimatedSalary'], hue =  
data['Gender'])
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f08d7c67d90>



```
plt.figure(figsize=(8,8))
sns.scatterplot(data['Age'], data['EstimatedSalary'], hue =
data['Balance'])
```

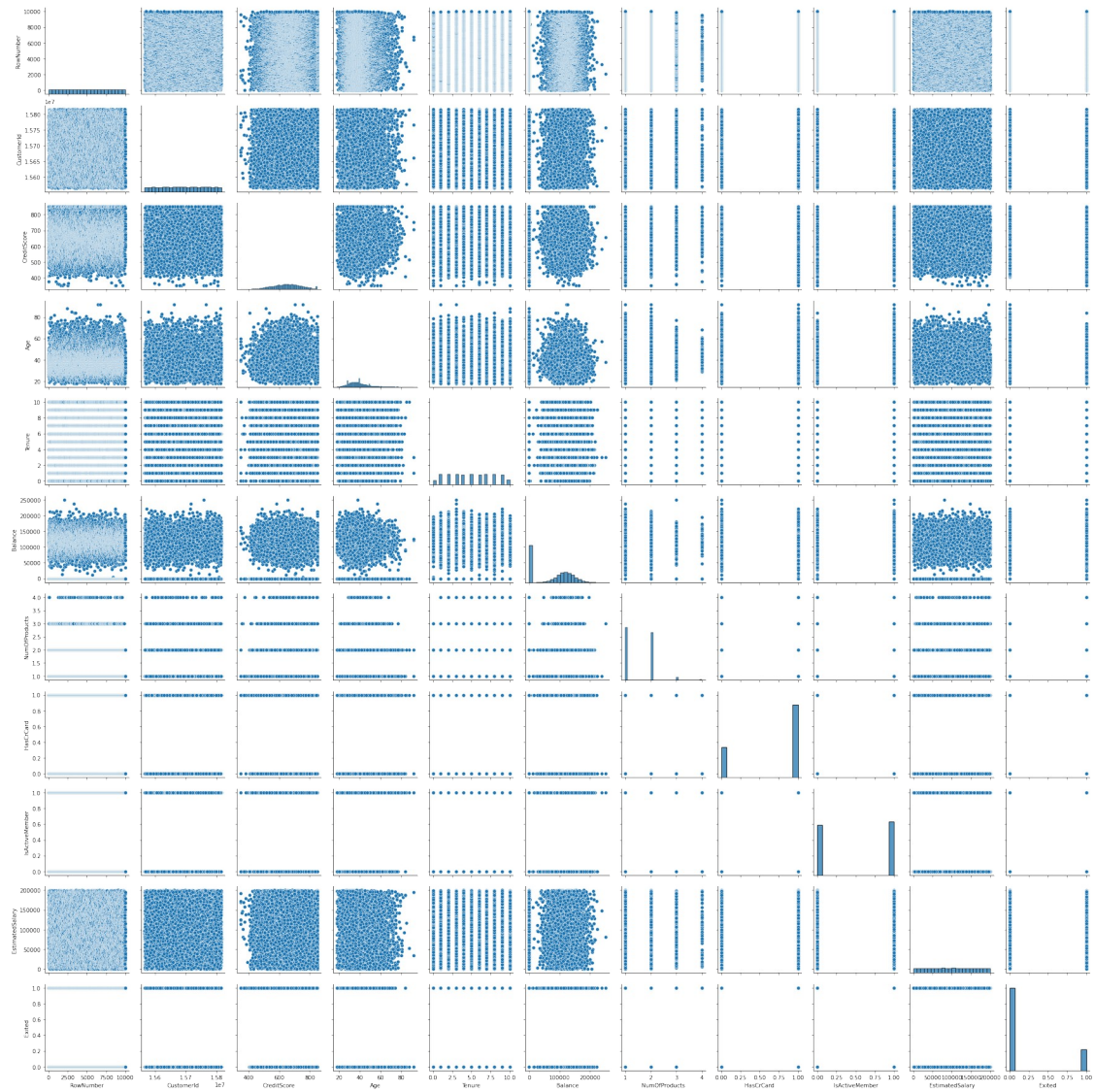
<matplotlib.axes.\_subplots.AxesSubplot at 0x7f08d637b1d0>



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

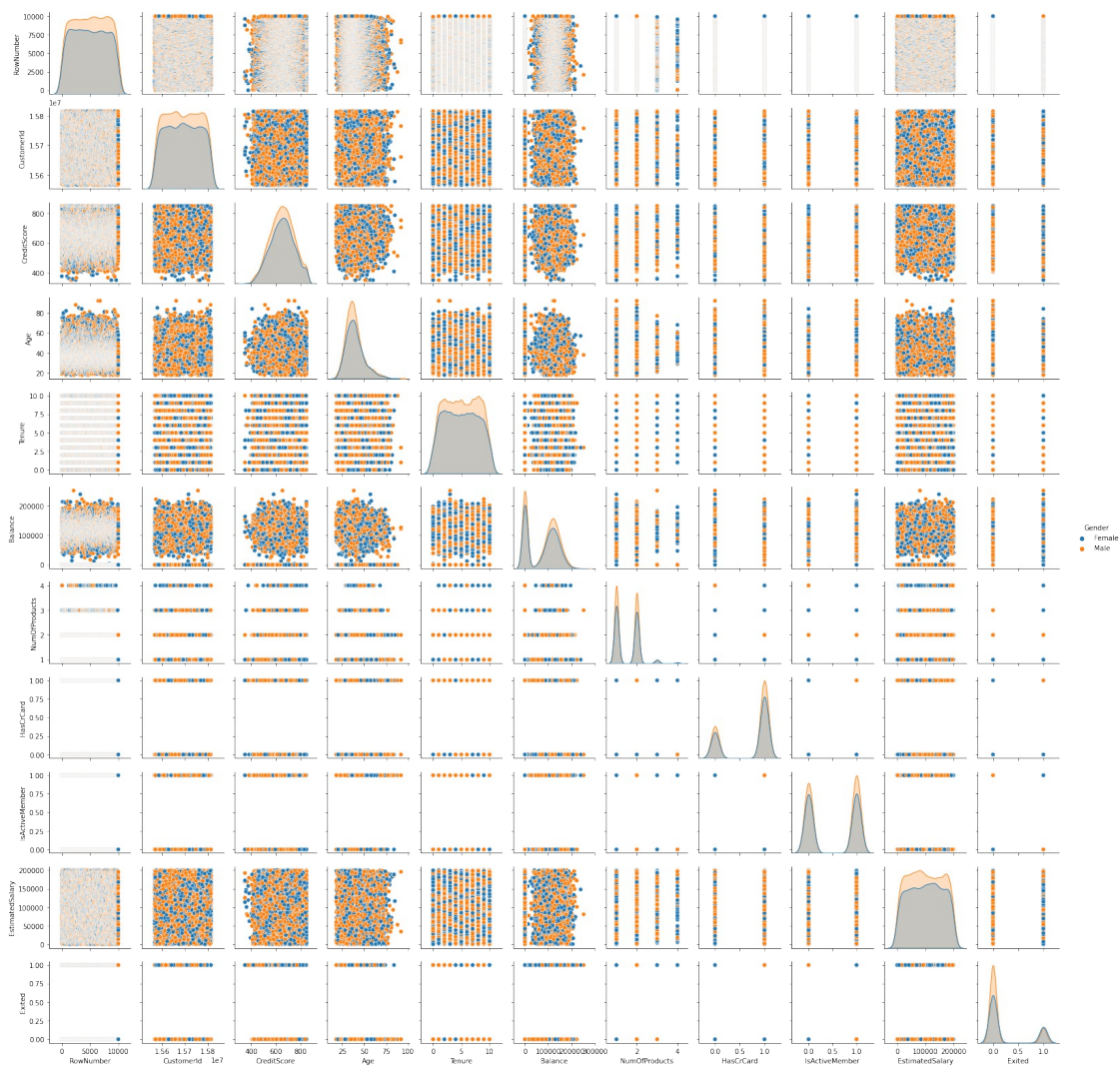
```
<seaborn.axisgrid.PairGrid at 0x7f08d63ed250>
```





```
sns.pairplot(df,hue="Gender",size=2)
```

```
<seaborn.axisgrid.PairGrid at 0x7f08c721f350>
```



```
data.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

```

```

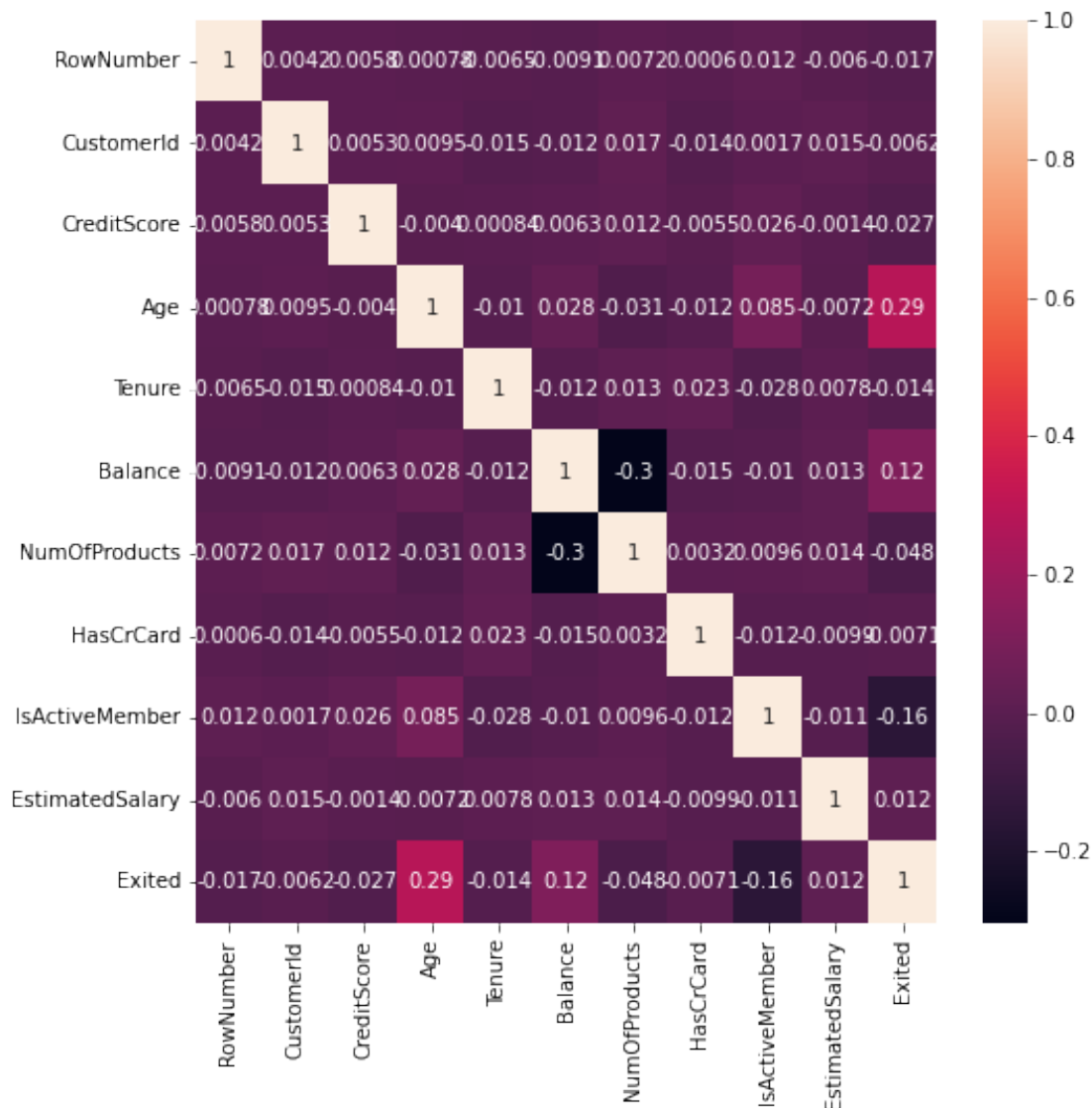
plt.figure(figsize=(8,8))
sns.heatmap(data.corr(), annot = True)

```

```

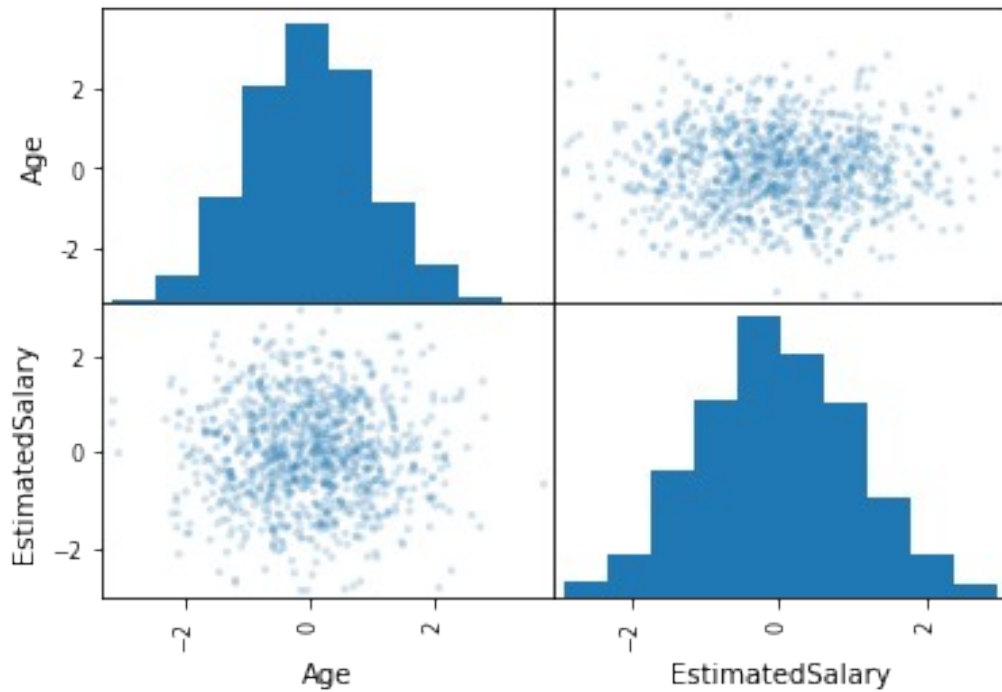
<matplotlib.axes._subplots.AxesSubplot at 0x7f08d5329850>

```



```
df = pd.DataFrame(np.random.randn(1000, 2),
columns=['Age','EstimatedSalary'])
pd.plotting.scatter_matrix(df, alpha=0.2)
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at
0x7f08d0675410>,
      <matplotlib.axes._subplots.AxesSubplot object at
0x7f08d069e990>],
      [<matplotlib.axes._subplots.AxesSubplot object at
0x7f08d0656f90>,
      <matplotlib.axes._subplots.AxesSubplot object at
0x7f08d06195d0>]],
      dtype=object)
```



#### 4)Descriptive Statistics

data.mean()

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	

data.median()

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
```

```
data.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]

data.var()

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

data.std

<bound method NDFrame.\_add\_numeric\_operations.<locals>.std 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	1	83807.86	1	0	1	1
2	8	159660.80	3	1	0	0
3	1	0.00	2	0	0	0
4	2	125510.82	1	1	1	1
...	...	...	...	...	...	...
9995	5	0.00	2	1	0	0
9996	10	57369.61	1	1	1	1
9997	7	0.00	1	0	1	1
9998	3	75075.31	2	1	0	0
9999	4	130142.79	1	1	0	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]>

data.describe()

	RowNumber	CustomerId	CreditScore	Age
Tenure \				
count	10000.000000	1.000000e+04	10000.000000	10000.000000
mean	5000.500000	1.569094e+07	650.528800	38.921800
std	2886.89568	7.193619e+04	96.653299	10.487806
min	1.000000	1.556570e+07	350.000000	18.000000
25%	2500.750000	1.562853e+07	584.000000	32.000000
50%	5000.500000	1.569074e+07	652.000000	37.000000
75%	7500.250000	1.575323e+07	718.000000	44.000000
max	10000.000000	1.581569e+07	850.000000	92.000000

	Balance	NumOfProducts	HasCrCard	IsActiveMember \
count	10000.000000	10000.000000	10000.000000	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

```
data['Age'].unique()
```

```
array([42, 41, 39, 43, 44, 50, 29, 27, 31, 24, 34, 25, 35, 45, 58, 32,
      38,
      46, 36, 33, 40, 51, 61, 49, 37, 19, 66, 56, 26, 21, 55, 75, 22,
      30,
      28, 65, 48, 52, 57, 73, 47, 54, 72, 20, 67, 79, 62, 53, 80, 59,
      68,
      23, 60, 70, 63, 64, 18, 82, 69, 74, 71, 76, 77, 88, 85, 84, 78,
      81,
      92, 83])
```

```
data['Gender'].unique()
```

```
array(['Female', 'Male'], dtype=object)
```

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

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

```
Name: Age, Length: 70, dtype: int64
```

```
data.max()
```

```

RowNumber      10000
CustomerId     15815690
Surname        Zuyeva
CreditScore    850
Geography      Spain
Gender         Male
Age            92
Tenure         10
Balance        250898.09
NumOfProducts  4
HasCrCard      1
IsActiveMember 1
EstimatedSalary 199992.48
Exited         1
dtype: object

```

## 5)Handle Missing Values

```
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

```
data.shape
```

```
(10000, 14)
```

```
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
```

```
data.isnull()
```

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

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

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

[10000 rows x 14 columns]

```
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

```

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

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender
Age	\					

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

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

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

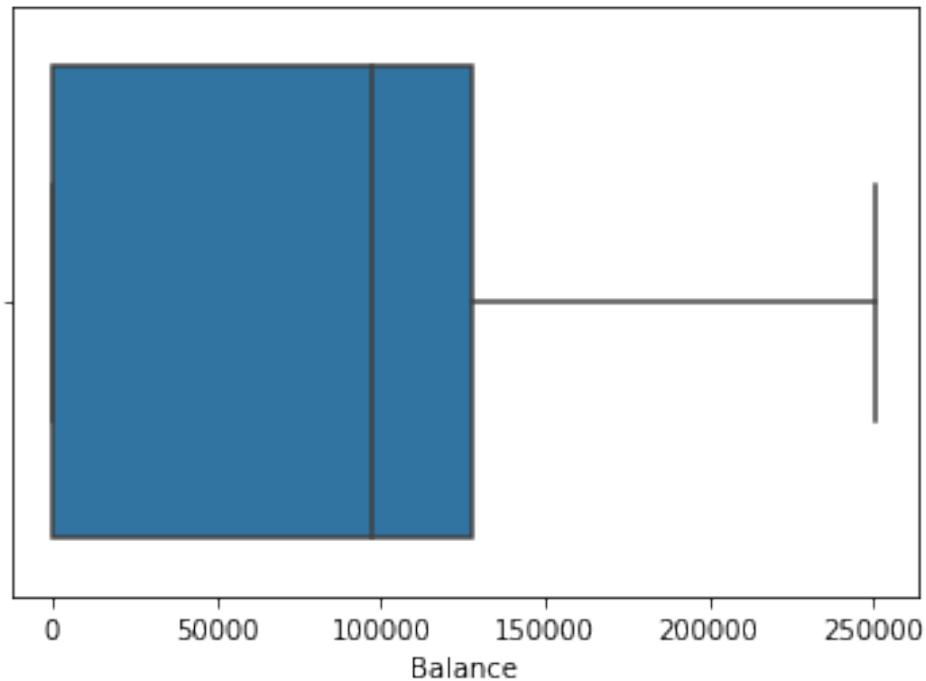
[10000 rows x 14 columns]



## 6)Outliers

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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f08d7c3e750>
```



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

	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				

```
iqr = q.iloc[0] - q.iloc[1]
```

```
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
```

```
u = q.iloc[0] + (1.5*iqr)
u
```

```
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
```

```
l = q.iloc[1] - (1.5*iqr)
l
```

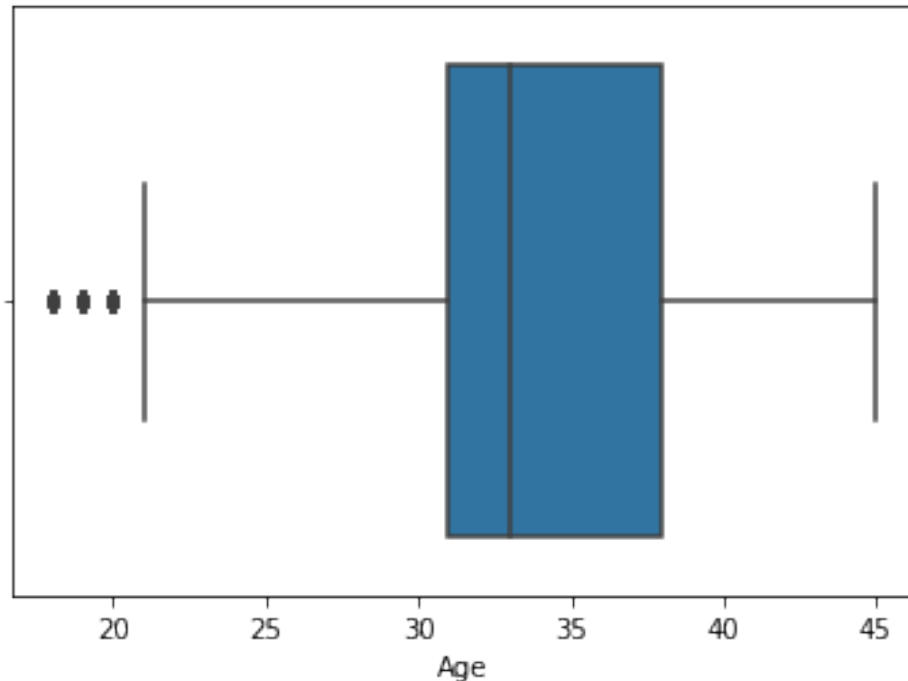
```
RowNumber      -4.998500e+03
CustomerId      1.544147e+07
CreditScore     3.830000e+02
Age             1.400000e+01
Tenure          -3.000000e+00
Balance         -1.914664e+05
NumOfProducts   -5.000000e-01
HasCrCard        -1.500000e+00
IsActiveMember   -1.500000e+00
EstimatedSalary -9.657710e+04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

Handling outliers

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

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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f08d04a8550>
```



## 7)Check for Categorical columns and perform encoding

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
le = LabelEncoder()
oneh = OneHotEncoder()
data['Gender'] = le.fit_transform(data['Gender'])

from sklearn.preprocessing import LabelEncoder, OneHotEncoder
le = LabelEncoder()
oneh = OneHotEncoder()
data['Gender'] = le.fit_transform(data['Gender'])

data.head()
```

	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

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

```
data['Age'] = le.fit_transform(data['Age'])
data['Geography'] = le.fit_transform(data['Geography'])
```

```
data.head()
```

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

	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

## 8) Split the data into dependent and independent variables

```
X=data.iloc[:,0:10]
```

```
X
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender
Age \						
0	1	15634602	Hargrave	619	0	0
24						
1	2	15647311	Hill	608	2	0
23						
2	3	15619304	Onio	502	0	0
24						
3	4	15701354	Boni	699	0	0
21						
4	5	15737888	Mitchell	850	2	0
25						
...	...	...	...	...	...	...
...						
9995	9996	15606229	Obijiaku	771	0	1
21						
9996	9997	15569892	Johnstone	516	0	1
17						
9997	9998	15584532	Liu	709	0	0
18						
9998	9999	15682355	Sabbatini	772	1	1
24						
9999	10000	15628319	Walker	792	0	0
10						

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

[10000 rows x 10 columns]

Y=data.iloc[:,10]

Y

0	1
1	0
2	1
3	0
4	1
...	...
9995	1

```

9996    1
9997    0
9998    1
9999    1
Name: HasCrCard, Length: 10000, dtype: int64

y = data['EstimatedSalary']
y
0      101348.88
1      112542.58
2      113931.57
3       93826.63
4       79084.10
...
9995       96270.64
9996      101699.77
9997       42085.58
9998       92888.52
9999       38190.78
Name: EstimatedSalary, Length: 10000, dtype: float64

```

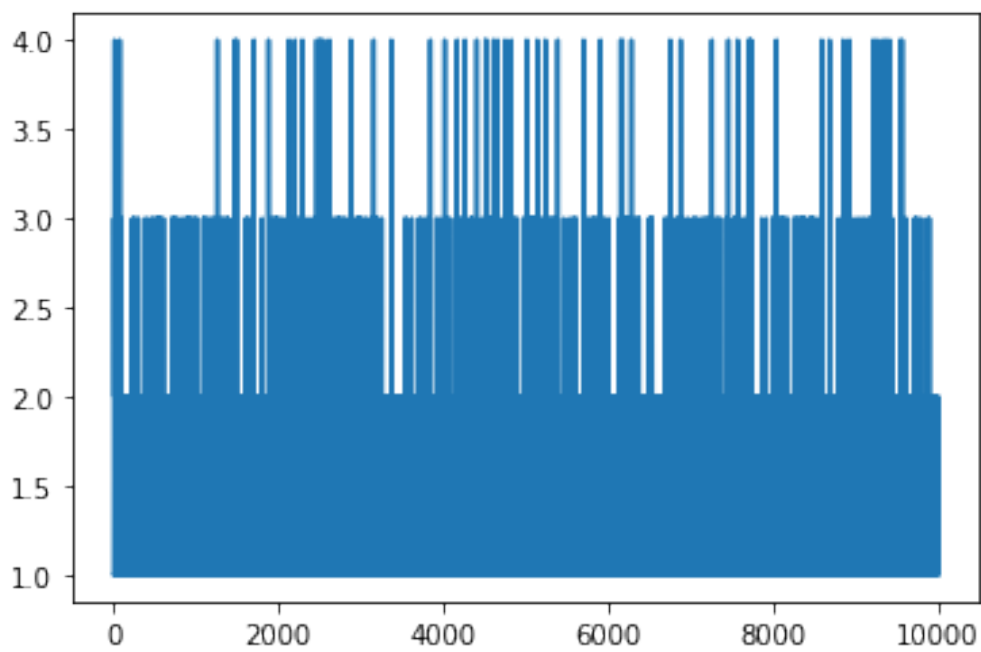
## 9)Scale the independent variables

```

EstimatedSalary=data.NumOfProducts
plt.plot(EstimatedSalary)

```

```
[<matplotlib.lines.Line2D at 0x7f08d020d5d0>]
```

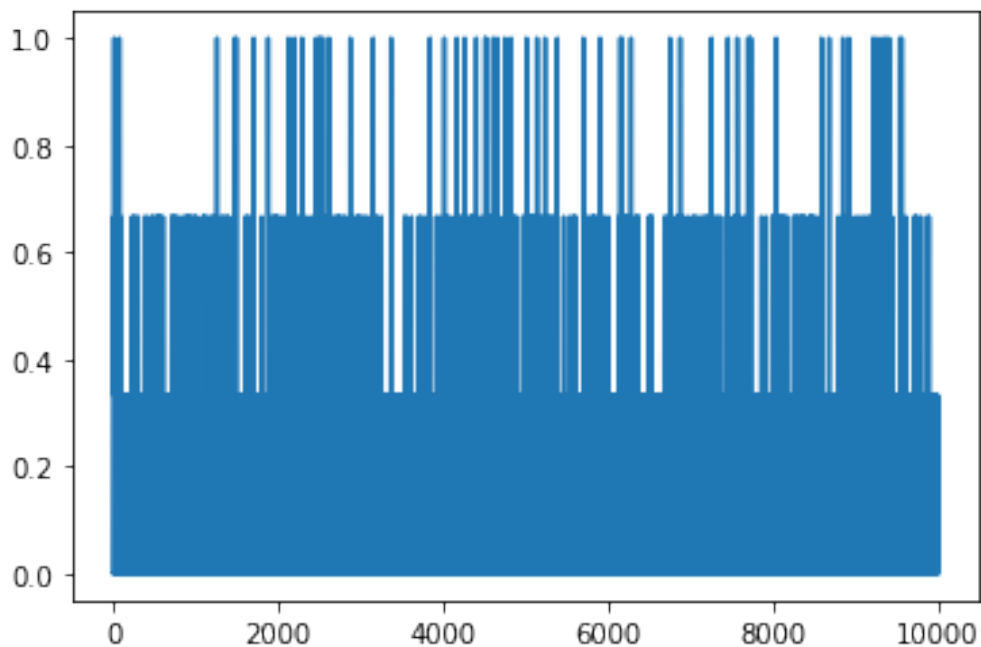


```
data[['EstimatedSalary']].describe()
```

	EstimatedSalary
count	10000.000000
mean	100090.239881
std	57510.492818
min	11.580000
25%	51002.110000
50%	100193.915000
75%	149388.247500
max	199992.480000

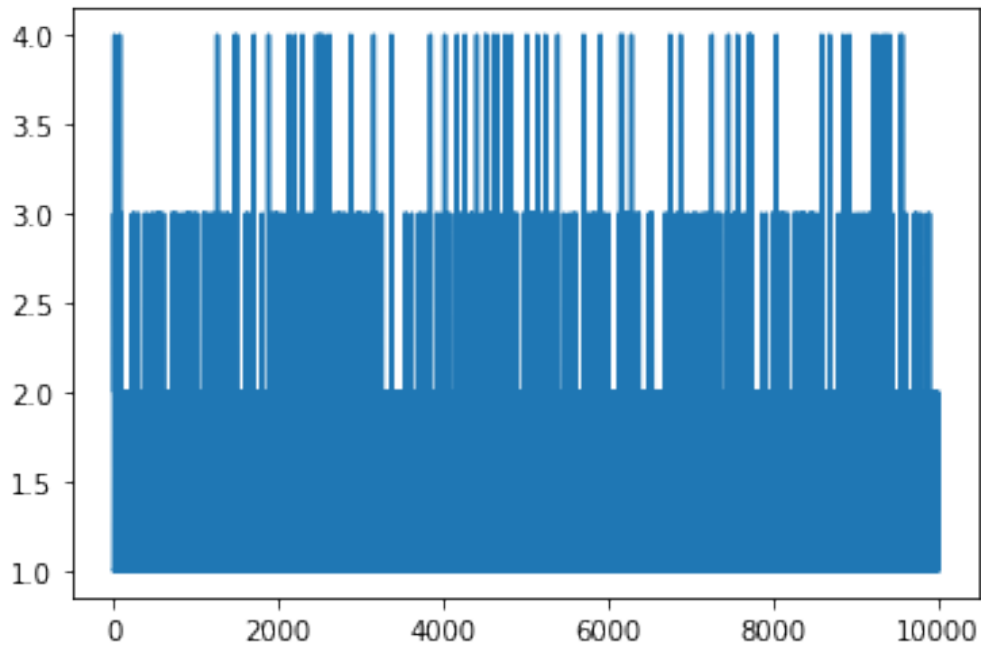
```
from sklearn import preprocessing
from sklearn.preprocessing import scale
EstimatedSalary_matrix=EstimatedSalary.values.reshape(-1,1)
scaled=preprocessing.MinMaxScaler()
scaled_EstimatedSalary=scaled.fit_transform(EstimatedSalary_matrix)
plt.plot(scaled_EstimatedSalary)
```

```
[<matplotlib.lines.Line2D at 0x7f08d01fb5d0>]
```



```
std_EstimatedSalary=scale(EstimatedSalary,axis=0,with_mean=False,with_
std=False)
plt.plot(std_EstimatedSalary)
```

```
[<matplotlib.lines.Line2D at 0x7f08d015eed0>]
```

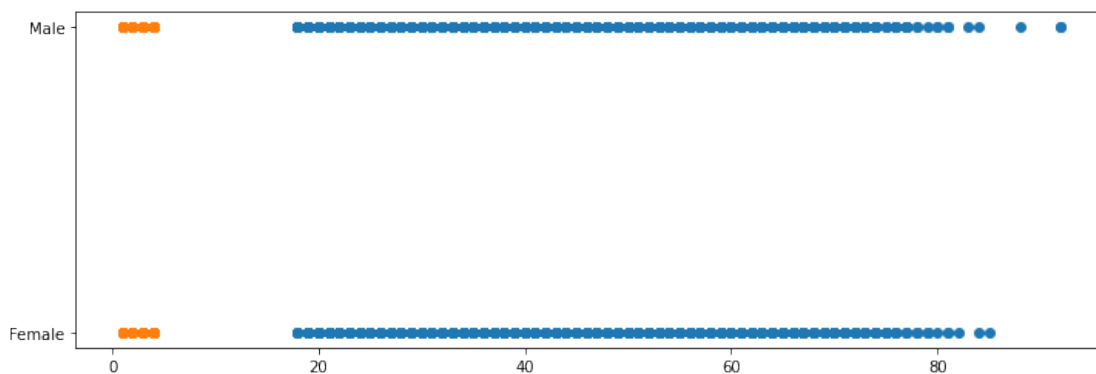


```
x = df[['Age', 'NumOfProducts']].values
y = df['Gender'].values
```

```
fig, ax = plt.subplots(figsize=(12, 4))
```

```
ax.scatter(x[:,0], y)
ax.scatter(x[:,1], y)
```

```
<matplotlib.collections.PathCollection at 0x7f08c4e6ae50>
```



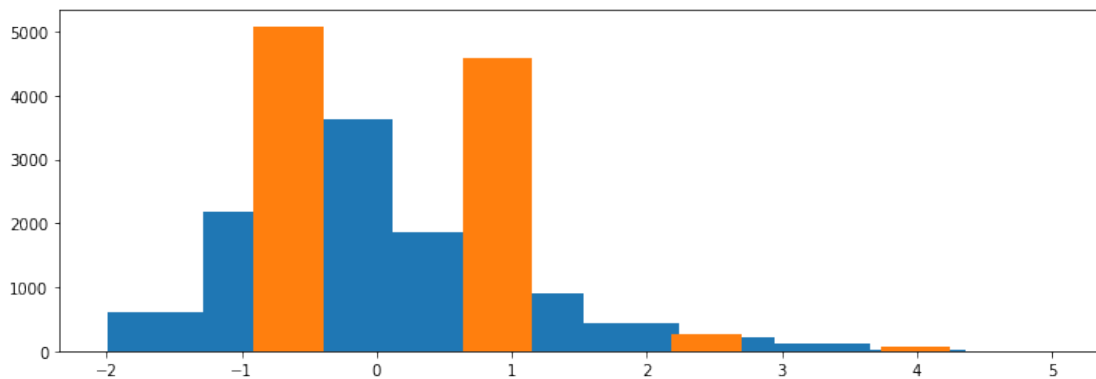
```
from sklearn.preprocessing import StandardScaler
fig, ax = plt.subplots(figsize=(12, 4))
```

```
scaler = StandardScaler()
x_std = scaler.fit_transform(x)
```

```
ax.hist(x_std[:,0])
ax.hist(x_std[:,1])
```



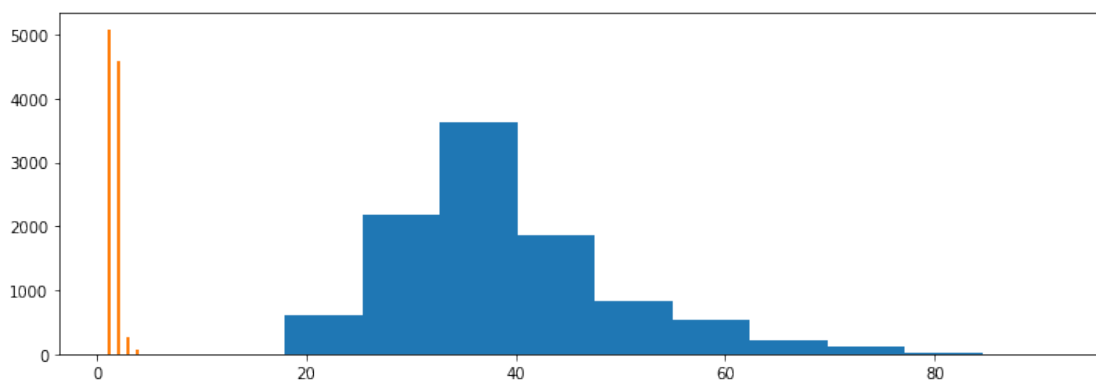
```
(array([5084.,    0.,    0., 4590.,    0.,    0., 266.,    0.,    0.,
        60.]),
 array([-0.91158349, -0.39578748,  0.12000854,  0.63580456,
        1.15160057,
        1.66739659,  2.18319261,  2.69898862,  3.21478464,
        3.73058066,
        4.24637668])),
<a list of 10 Patch objects>)
```



```
fig, ax = plt.subplots(figsize=(12, 4))
```

```
ax.hist(x[:,0])
ax.hist(x[:,1])
```

```
(array([5084.,    0.,    0., 4590.,    0.,    0., 266.,    0.,    0.,
        60.]),
 array([1. , 1.3, 1.6, 1.9, 2.2, 2.5, 2.8, 3.1, 3.4, 3.7, 4. ]),
<a list of 10 Patch objects>)
```



## 10) Split the data into training and testing

```
x=np.array(data["Surname"]).reshape(-1,1)
```

```
x.shape
```

```
(10000, 1)
```

```

y=np.array(data["EstimatedSalary"])
y.shape

(10000,)

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30)

x_train
array([[ 'Gardner'],
       [ 'Stevenson'],
       [ 'Costa'],
       ...,
       [ 'McCawley'],
       [ 'Miller'],
       [ 'Capon']], dtype=object)

x_train.shape

(7000, 1)

x_test
array([[ 'Duncan'],
       [ "O'Brien"],
       [ 'Hunt'],
       ...,
       [ 'Chiang'],
       [ 'Ferguson'],
       [ 'Wimble']], dtype=object)

x_test.shape

(3000, 1)

y_train
array([ 47848.56, 125518.32,  43174.49, ..., 198914.8 , 109794.31,
        52796.31])

y_test
array([ 86410.28, 177025.79, 101455.07, ..., 110114.38, 106918.67,
        54865.92])

```