

Name	SWETHA.A
Register Number	720419104029
Team Size	4
Team ID	PNT2022TMID43503

Assignment -II

1) Importing

In []:

```
import pandas as pd
```

```
import numpy as np
```

```
import seaborn as sns
```

```
from matplotlib import pyplot as plt
```

```
import warnings
```

```
warnings.filterwarnings('ignore')
```

2.Load the Dataset

In []:

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

In [43]:

```
data
```

Out[43]:

	Row	Cust	Sur	Cred	Geo	Ge	A	Te	Bal	NumO	Has	IsActiv	Estima	Ex
	Num	omer	na	itSco	grap	nd	g	nu	anc	fProdu	CrC	eMem	tedSal	ite
	ber	Id	me	re	hy	er	e	re	e	cts	ard	ber	ary	d

0	1	0.27 5616	Har gra ve	619	Fran ce	Fe ma le	4 2	2	0.00	1	1	1	101348 .88	1
1	2	0.32 6454	Hill	608	Spai n	Fe ma le	4 1	1	838 07.8 6	1	0	1	112542 .58	0
2	3	0.21 4421	Oni o	502	Fran ce	Fe ma le	4 2	8	159 660. 80	3	1	0	113931 .57	1
3	4	0.54 2636	Bon i	699	Fran ce	Fe ma le	3 9	1	0.00	2	0	0	93826. 63	0
4	5	0.68 8778	Mit chel l	850	Spai n	Fe ma le	4 3	2	125 510. 82	1	1	1	79084. 10	0
...
9 9 9 5	9996	0.16 2119	Obi jiak u	771	Fran ce	Ma le	3 9	5	0.00	2	1	0	96270. 64	0
9 9 9 6	9997	0.01 6765	Joh nsto ne	516	Fran ce	Ma le	3 5	10	573 69.6 1	1	1	1	101699 .77	0
9 9	9998	0.07 5327	Liu	709	Fran ce	Fe ma le	3 6	7	0.00	1	0	1	42085. 58	1

97														
9998	9999	0.466637	Sab bati ni	772	Ger man y	Ma le	42	3	75075.31	2	1	0	92888.52	1
9999	10000	0.250483	Wal ker	792	Fran ce	Fe ma le	28	4	130142.79	1	1	0	38190.78	0

10000 rows × 14 columns

3. Visualizations

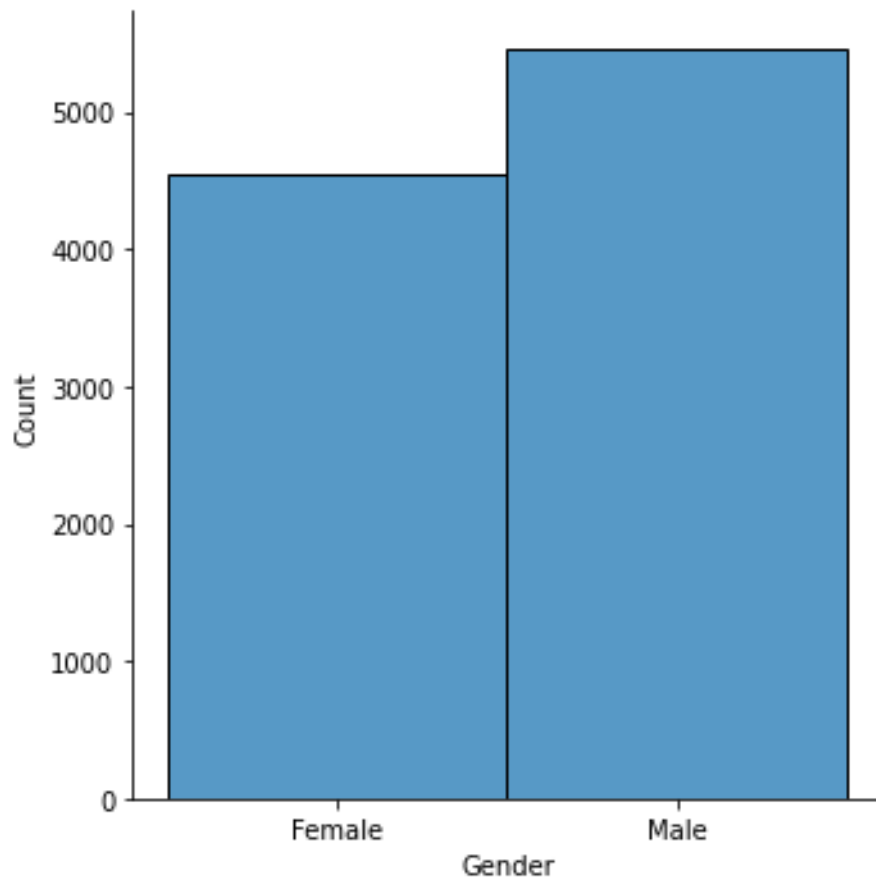
a) Univariate Analysis

In [44]:

```
sns.displot(data.Gender)
```

Out[44]:

<seaborn.axisgrid.FacetGrid at 0x7f80cb07c690>



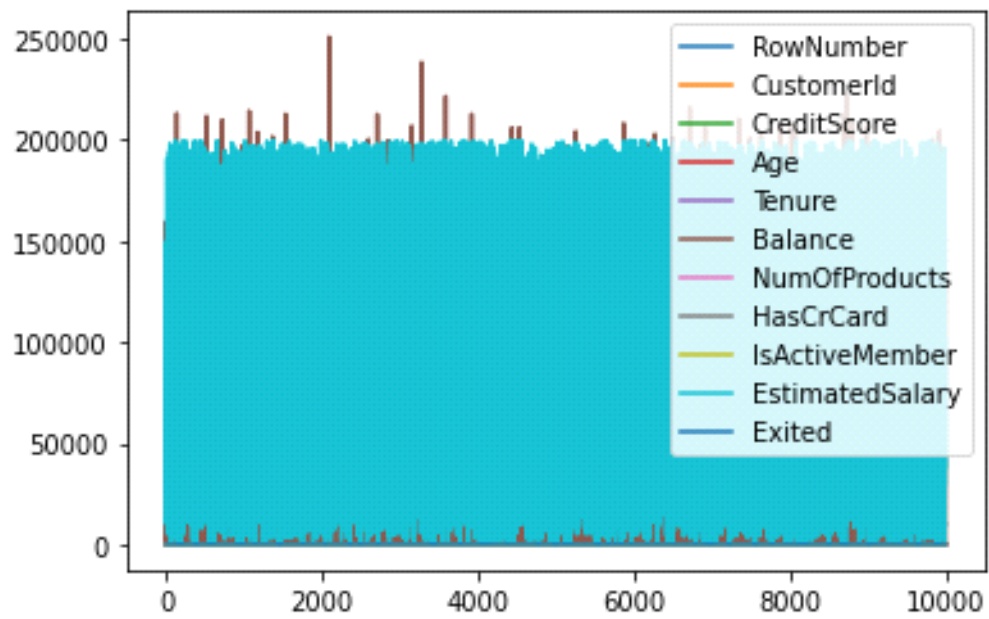
B)Bi-Variate Analysis

In [45]:

```
data.plot.line()
```

Out[45]:

<matplotlib.axes._subplots.AxesSubplot at 0x7f80cb9a8a50>



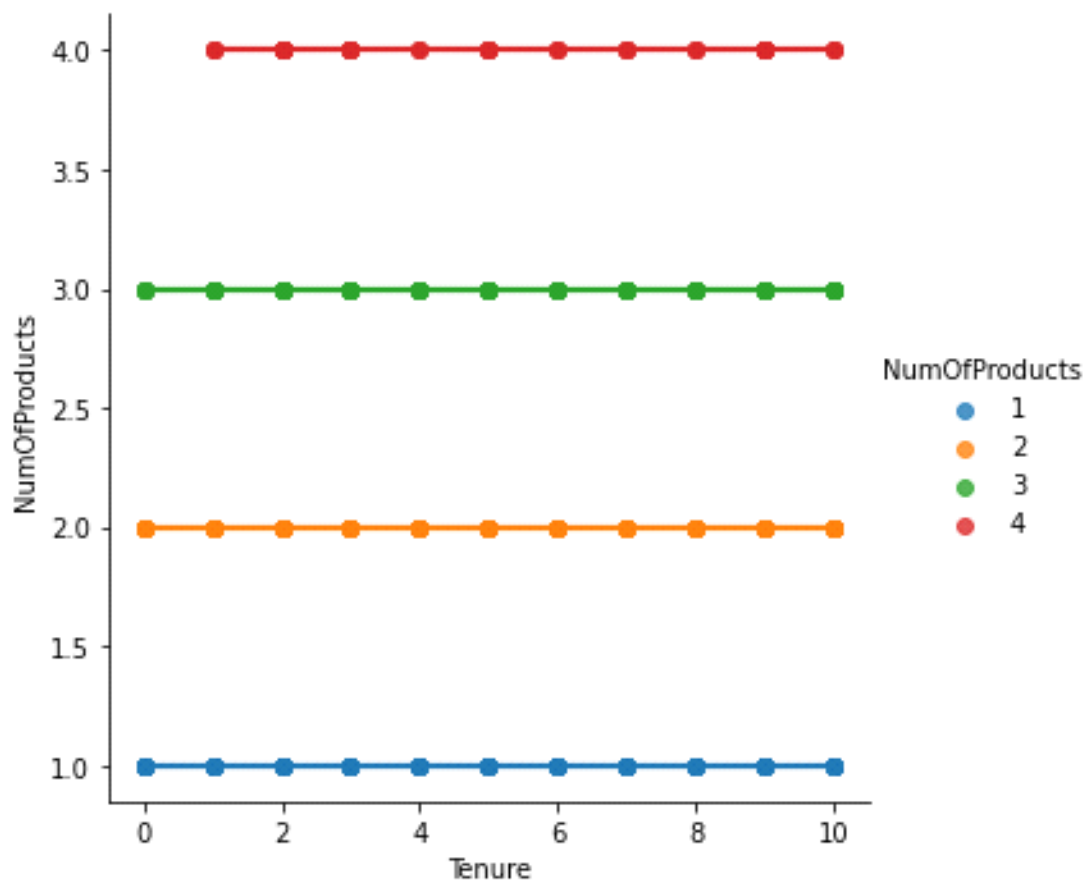
C)Multi - Variate Analysis

In [46]:

```
sns.lmplot("Tenure", "NumOfProducts", data, hue="NumOfProducts")
```

Out[46]:

<seaborn.axisgrid.FacetGrid at 0x7f80cb95fe10>



4) Perform descriptive statistics on the dataset.

In [47]:

```
data.describe()
```

Out[47]:

	Row Num ber	Custo merId	Credi tScore	Age	Tenur e	Balanc e	NumOf Produc ts	HasC rCar d	IsActiv eMemb er	Estimat edSalar y	Exite d
co un t	10000 .0000 0	10000. 00000 0	10000. 00000 0	10000. 00000 0	10000. 00000 0	10000. 000000	10000.0 00000	10000 .0000 0	10000.0 00000	10000.0 00000	10000. 00000 0
m ea n	5000. 50000	0.5009 80	650.52 8800	36.533 900	5.0128 00	76485. 889288	1.53020 0	0.705 50	0.51510 0	100090. 239881	0.2037 00

std	2886. 89568	0.2877 57	96.653 299	6.4738 43	2.8921 74	62397. 405202	0.58165 4	0.455 84	0.49979 7	57510.4 92818	0.4027 69
min	1.000 00	0.0000 00	350.00 0000	20.000 000	0.0000 00	0.0000 00	1.00000 0	0.000 00	0.00000 0	11.5800 00	0.0000 00
25%	2500. 75000	0.2513 20	584.00 0000	32.000 000	3.0000 00	0.0000 00	1.00000 0	0.000 00	0.00000 0	51002.1 10000	0.0000 00
50%	5000. 50000	0.5001 70	652.00 0000	37.000 000	5.0000 00	97198. 540000	1.00000 0	1.000 00	1.00000 0	100193. 915000	0.0000 00
75%	7500. 25000	0.7501 64	718.00 0000	40.000 000	7.0000 00	127644 .24000 0	2.00000 0	1.000 00	1.00000 0	149388. 247500	0.0000 00
max	10000 .0000 0	1.0000 00	850.00 0000	50.000 000	10.000 000	250898 .09000 0	4.00000 0	1.000 00	1.00000 0	199992. 480000	1.0000 00

5)Handle the Missing values.

In []:

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

```
pd.isnull(data["Gender"])
```

Out[]:

0 False

1 False

2 False

3 False

4 False

...

9995 False

9996 False

9997 False

9998 False

9999 False

Name: Gender, Length: 10000, dtype: bool

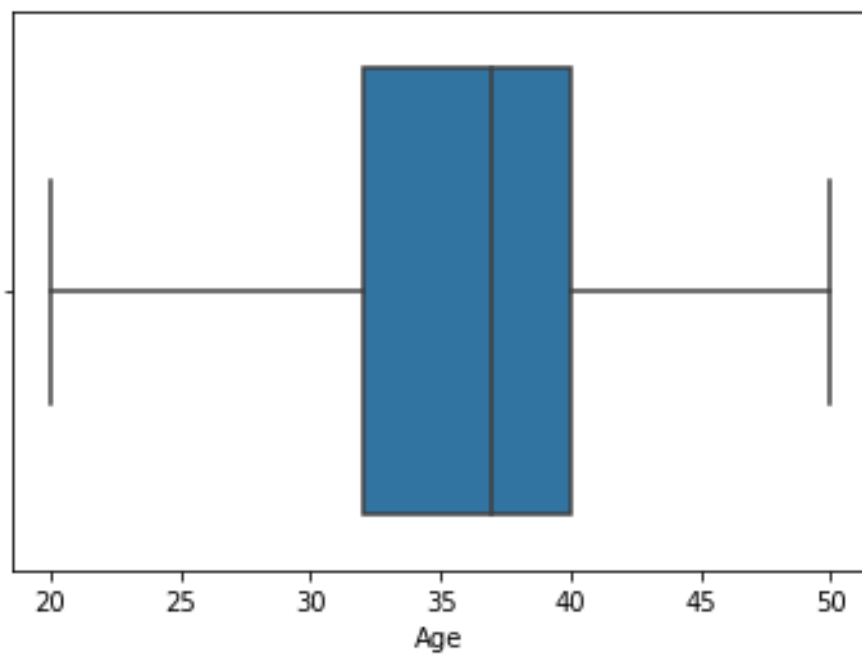
6) Find the outliers and replace the outliers

In [48]:

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

Out[48]:

<matplotlib.axes._subplots.AxesSubplot at 0x7f80caeafc50>



In [28]:

```
data['Age']=np.where(data['Age']>50,40,data['Age'])
```

```
data['Age']
```

Out[28]:

0 42

1 41

2 42

3 39

4 43

..

9995 39

9996 35

9997 36

9998 42

9999 28

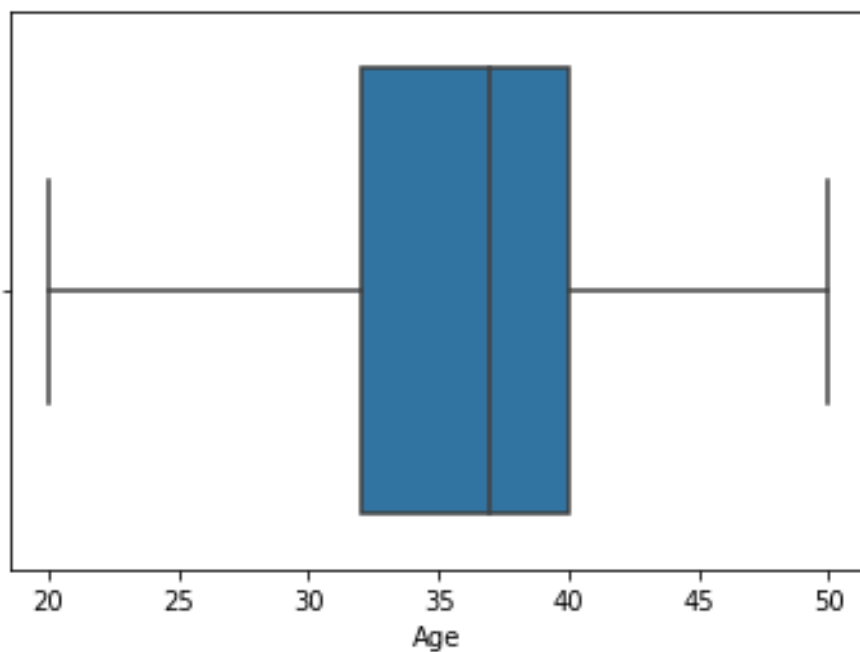
Name: Age, Length: 10000, dtype: int64

In [49]:

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

Out[49]:

<matplotlib.axes._subplots.AxesSubplot at 0x7f80cb95fc10>



In [34]:

```
data['Age']=np.where(data['Age']<20,35,data['Age'])
```

data['Age']

Out[34]:

0 42

1 41

2 42

3 39

	R o w N u m be r	C us to m e r I d	S u r n a m e	C re di tS co re	G eo gr a p h y	T e n u r e	B al a n c e	Nu m Of Pr od uct s	H as C r C ar d	Is Ac tiv eM em ber	.	G e n d er — 4 1	G e n d er — 4 2	G e n d er — 4 3	G e n d er — 4 4	G e n d er — 4 5	G e n d er — 4 6	G e n d er — 4 7	G e n d er — 4 8	G e n d er — 4 9	G e n d er — 5 0	
	0	1	0. 27 56 16 rave	H ar g ra v e	61 9	Fr an ce	2	0. 0 0	1	1	1	.	.	.	0	1	0	0	0	0	0	0
	1	2	0. 32	H il l	60 8	S pa in	1	8 3 8 0	1	0	1	.	.	.	1	0	0	0	0	0	0	0

		64 54					7. 8 6														
2	3	0. 21 44 21	O ni o	50 2	Fr an ce	8	1 5 9 6 6 0. 8 0	3	1	0	.	0	1	0	0	0	0	0	0	0	0
3	4	0. 54 26 36	B o ni	69 9	Fr an ce	1	0. 0 0	2	0	0	.	0	0	0	0	0	0	0	0	0	0
4	5	0. 68 87 78	M it c h el l	85 0	S pa in	2	1 2 5 5 1 0. 8 2	1	1	1	.	0	0	1	0	0	0	0	0	0	0

5 rows × 45 columns

8) Split the data into dependent and independent variables.

A) Split the data into Independent variables.

In [37]:

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

```
print(X)
```

```
[[1 15634602 'Hargrave' ... 1 1 101348.88]
[2 15647311 'Hill' ... 0 1 112542.58]
[3 15619304 'Onio' ... 1 0 113931.57]
...
[9998 15584532 'Liu' ... 0 1 42085.58]
[9999 15682355 'Sabbatini' ... 1 0 92888.52]
[10000 15628319 'Walker' ... 1 0 38190.78]]
```

B) Split the data into Dependent variables.

In [38]:

```
Y = data.iloc[:, -1].values
print(Y)
```

```
[1 0 1 ... 1 1 0]
```

9) Scale the independent variables

In [39]:

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
data[["CustomerId"]] = scaler.fit_transform(data[["CustomerId"]])
```

In [40]:

```
print(data)
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age \
0	1	0.275616	Hargrave	619	France	Female	42
1	2	0.326454	Hill	608	Spain	Female	41
2	3	0.214421	Onio	502	France	Female	42
3	4	0.542636	Boni	699	France	Female	39
4	5	0.688778	Mitchell	850	Spain	Female	43
...
9995	9996	0.162119	Obijiaku	771	France	Male	39
9996	9997	0.016765	Johnstone	516	France	Male	35

9997	9998	0.075327	Liu	709	France	Female	36
9998	9999	0.466637	Sabbatini	772	Germany	Male	42
9999	10000	0.250483	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]

10) Split the data into training and testing

In [42]:

```
from sklearn.model_selection import train_test_split

train_size=0.8

X = data.drop(columns = ['Tenure']).copy()
y = data['Tenure']

X_train, X_rem, y_train, y_rem = train_test_split(X,y, train_size=0.8)

test_size = 0.5

X_valid, X_test, y_valid, y_test = train_test_split(X_rem,y_rem, test_size=0.5)

print(X_train.shape), print(y_train.shape)

print(X_valid.shape), print(y_valid.shape)

print(X_test.shape), print(y_test.shape)

(8000, 13)

(8000,)

(1000, 13)

(1000,)

(1000, 13)

(1000,)

Out[42]:

(None, None)
```