Assignment -2

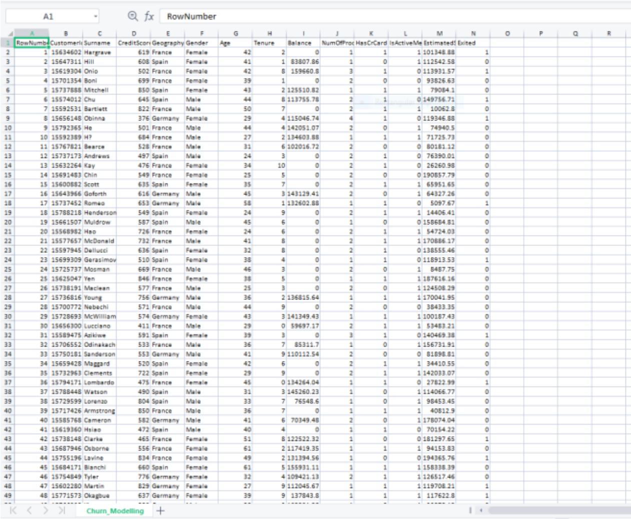
DATA VISUALISATION AND PRE-PROCESSING

Assignment Date	28 September 2022
Student Name	GOKULRAJ S
Student Roll Number	612419104011
Maximum Marks	2 Marks

Question-1:

Download the Dataset

SOLUTION:



A1 fx RowNumber

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	RowNumbe	Customerl	Surname	CreditScor	Geography	Gender	Age	Tenure	Balance	NumOfProc	HasCrCard	IsActiveMe	Estimated!	Exited
2	1	15634602	Hargrave	619	France	Female	42	2	0	1	1	1	101348.88	1
3	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
4	3	15619304	Onio	502	France	Female	42	8	159660.8	3	1	0	113931.57	1
5	4	15701354	Boni	699	France	Female	39	1	0	2	0	0	93826.63	0
6	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.1	0
7	6	15574012	Chu	645	Spain	Male	44	8	113755.78	2	1	0	149756.71	1
8	7	15592531	Bartlett	822	France	Male	50	7	0	2	1	1	10062.8	0
9	8	15656148	Obinna	376	Germany	Female	29	4	115046.74	4	1	0	119346.88	1
10	9	15792365	He	501	France	Male	44	4	142051.07	2	0	1	74940.5	0
11	10	15592389	H?	684	France	Male	27	2	134603.88	1	1	1	71725.73	0
12	11	15767821	Bearce	528	France	Male	31	6	102016.72	2	0	0	80181.12	0
13	12	15737173	Andrews	497	Spain	Male	24	3	0	2	1	0	76390.01	0
14	13	15632264	Kay	476	France	Female	34	10	0	2	1	0	26260.98	0
15	14	15691483	Chin	549	France	Female	25	5	0	2	0	0	190857.79	0
16	15	15600882	Scott	635	Spain	Female	35	7	0	2	1	1	65951.65	0
17	16	15643966	Goforth	616	Germany	Male	45	3	143129.41	2	0	1	64327.26	0
18	17	15737452	Romeo	653	Germany	Male	58	1	132602.88	1	1	0	5097.67	1
19	18	15788218	Henderson	549	Spain	Female	24	9	0	2	1	1	14406.41	0
20	19	15661507	Muldrow	587	Spain	Male	45	6	0	1	0	0	158684.81	0
21	20	15568982	Hao	726	France	Female	24	6	0	2	1	1	54724.03	0
22	21	15577657	McDonald	732	France	Male	41	8	0	2	1	1	170886.17	0
23	22	15597945	Dellucci	636	Spain	Female	32	8	0	2	1	0	138555.46	0
24	23	15699309	Gerasimov	510	Spain	Female	38	4	0	1	1	0	118913.53	1
25	24	15725737	Mosman	669	France	Male	46	3	0	2	0	1	8487.75	0
26	25	15625047	Yen	846	France	Female	38	5	0	1	1	1	187616.16	0
27	26	15738191	Maclean	577	France	Male	25	3	0	2	0	1	124508.29	0
28	27	15736816	Young	756	Germany	Male	36	2	136815.64	1	1	1	170041.95	0
29	28	15700772	Nebechi	571	France	Male	44	9	0	2	0	0	38433.35	0
30	29	15728693	McWilliam	574	Germany	Female	43	3	141349.43	1	1	1	100187.43	0
31	30	15656300	Lucciano	411	France	Male	29	0	59697.17	2	1	1	53483.21	0
32	31	15589475	Azikiwe	591	Spain	Female	39	3	0	3	1	0	140469.38	1
33	32	15706552	Odinakach	533	France	Male	36	7	85311.7	1	0	1	156731.91	0
34	33	15750181	Sanderson	553	Germany	Male	41	9	110112.54	2	0	0	81898.81	0
35	34	15659428	Maggard	520	Spain	Female	42	6	0	2	1	1	34410.55	0
36	35	15732963	Clements	722	Spain	Female	29	9	0	2	1	1	142033.07	0
37	36	15794171	Lombardo	475	France	Female	45	0	134264.04	1	1	0	27822.99	1
38	37	15788448	Watson	490	Spain	Male	31	3	145260.23	1	0	1	114066.77	0
39	38	15729599	Lorenzo	804	Spain	Male	33	7	76548.6	1	0	1	98453.45	0
40	39	15717426	Armstrong	850	France	Male	36	7	0	1	1	1	40812.9	0
41	40	15585768	Cameron	582	Germany	Male	41	6	70349.48	2	0	1	178074.04	0
42	41	15619360	Hsiao	472	Spain	Male	40	4	0	1	1	0	70154.22	0
43	42	15738148	Clarke	465	France	Female	51	8	122522.32	1	0	0	181297.65	1
44	43	15687946	Osborne	556	France	Female	61	2	117419.35	1	1	1	94153.83	0
45	44	15755196	Lavine	834	France	Female	49	2	131394.56	1	0	0	194365.76	1
46	45	15684171	Bianchi	660	Spain	Female	61	5	155931.11	1	1	1	158338.39	0
47	46	15754849	Tyler	776	Germany	Female	32	4	109421.13	2	1	1	126517.46	0
48	47	15602280	Martin	829	Germany	Female	27	9	112045.67	1	1	1	119708.21	1
49	48	15771573	Okagbue	637	Germany	Female	39	9	137843.8	1	1	1	117622.8	1

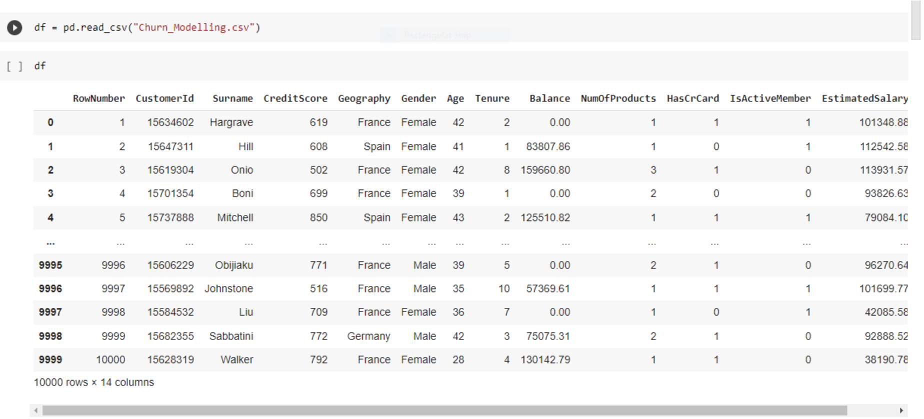
Churn_Modelling

Question-2:

Loading dataset

SOLUTION:

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



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

```
[ ] df
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9995	9996	15606229	Obijiaku	771	France	Male	39	5	0.00	2	1	0	96270.64
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	101699.77
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	42085.58
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	92888.52
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	38190.78

10000 rows × 14 columns

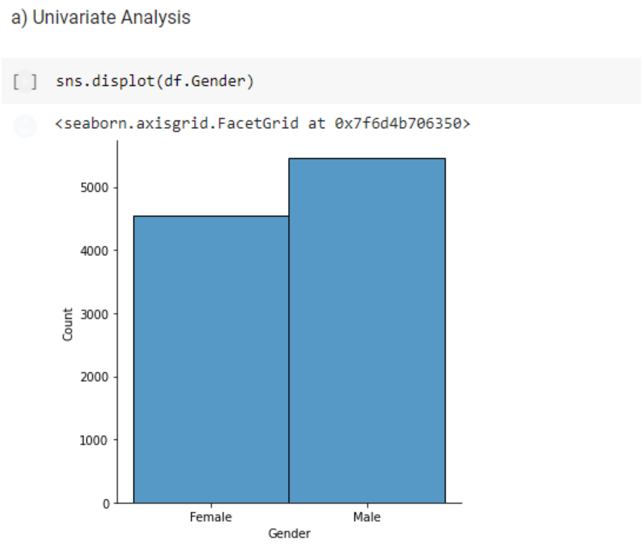
Question-3:

1.Visualizations

a) Univariate Analysis

SOLUTION:

```
sns.displot(df.Gender)
```



a) Univariate Analysis

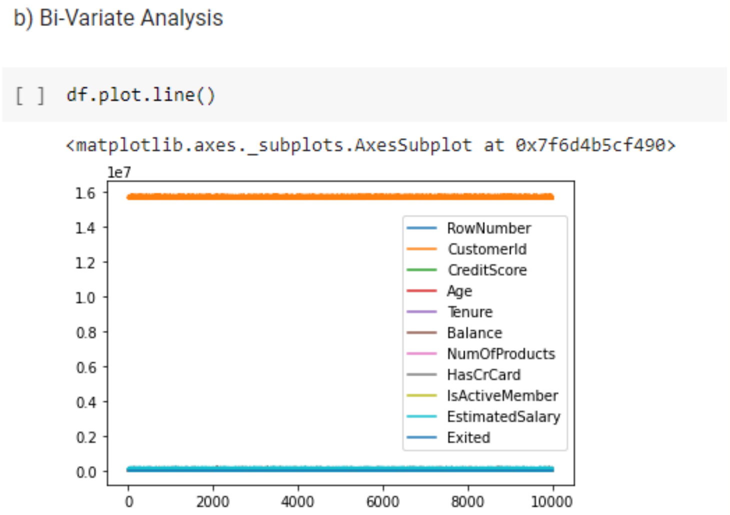
```
[ ] sns.displot(df.Gender)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f6d4b706350>
```

b) Bi-Variate Analysis

SOLUTION:

```
df.plot.line()
```



b) Bi-Variate Analysis

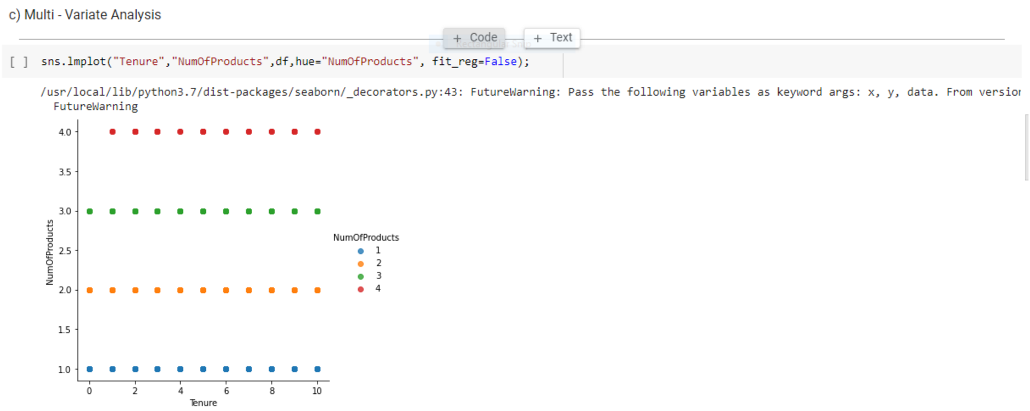
```
[ ] df.plot.line()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6d4b5cf490>
```

c) Multi - Variate Analysis

SOLUTION:

```
sns.lmplot("Tenure","NumOfProducts",df,hue="NumOfProducts", fit_reg=False);
```



c) Multi - Variate Analysis

```
[ ] sns.lmplot("Tenure","NumOfProducts",df,hue="NumOfProducts", fit_reg=False);
```

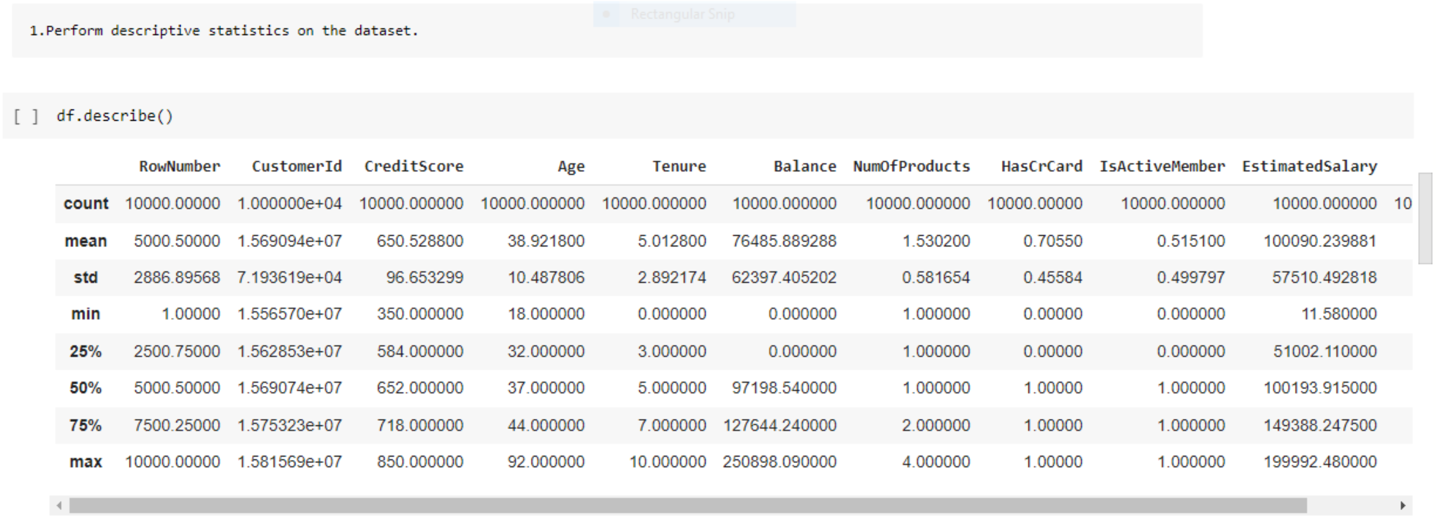
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y, data. From versio
  FutureWarning
```

Question-4:

Perform descriptive statistics on the dataset.

SOLUTION:

```
df.describe()
```



1.Perform descriptive statistics on the dataset.

```
[ ] df.describe()
```

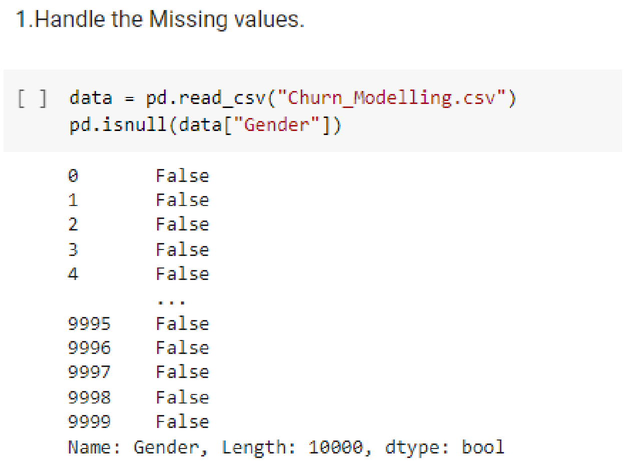
	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.00000	10000.000000	10000.000000	10
mean	5000.50000	1.569094e+07	650.528800	38.921800	5.012800	76485.889288	1.530200	0.70550	0.515100	100090.239881	
std	2886.89568	7.193619e+04	96.653299	10.487806	2.892174	62397.405202	0.581654	0.45584	0.499797	57510.492818	
min	1.00000	1.556570e+07	350.000000	18.000000	0.000000	0.000000	1.000000	0.00000	0.000000	11.580000	
25%	2500.75000	1.562853e+07	584.000000	32.000000	3.000000	0.000000	1.000000	0.00000	0.000000	51002.110000	
50%	5000.50000	1.569074e+07	652.000000	37.000000	5.000000	97198.540000	1.000000	1.00000	1.000000	100193.915000	
75%	7500.25000	1.575323e+07	718.000000	44.000000	7.000000	127644.240000	2.000000	1.00000	1.000000	149388.247500	
max	10000.00000	1.581569e+07	850.000000	92.000000	10.000000	250898.090000	4.000000	1.00000	1.000000	199992.480000	

Question-5:

Handle the Missing values.

SOLUTION:

```
data = pd.read_csv("Churn_Modelling.csv")
pd.isnull(data["Gender"])
```



1.Handle the Missing values.

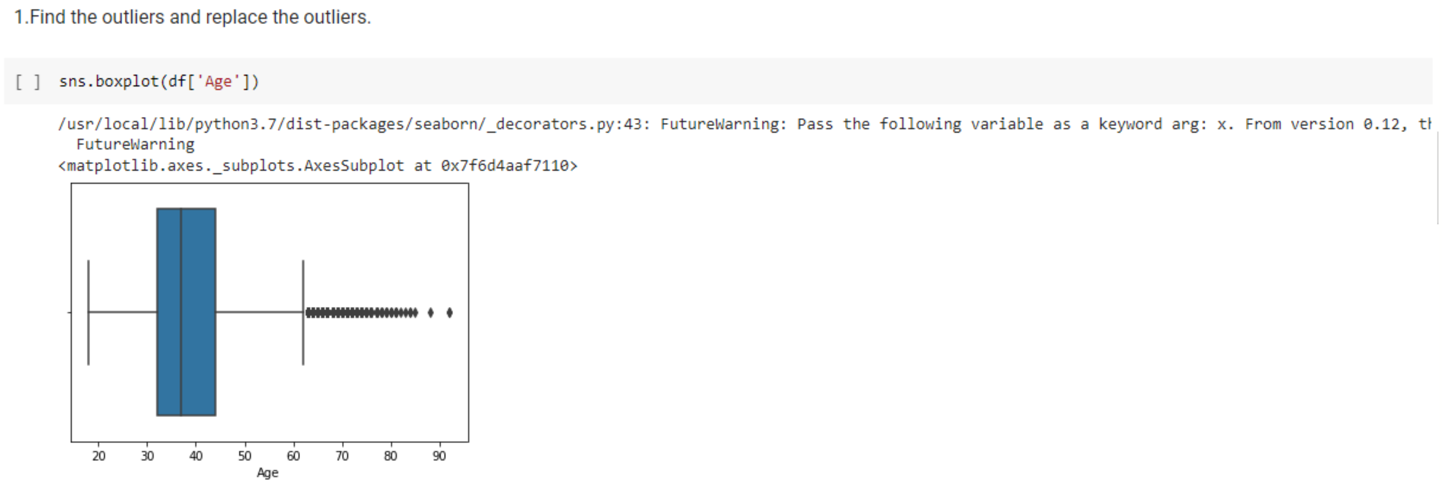
```
[ ]  data = pd.read_csv("Churn_Modelling.csv")
     pd.isnull(data["Gender"])

     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
```

Question-6:

Find the outliers and replace the outliers.

SOLUTION:

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

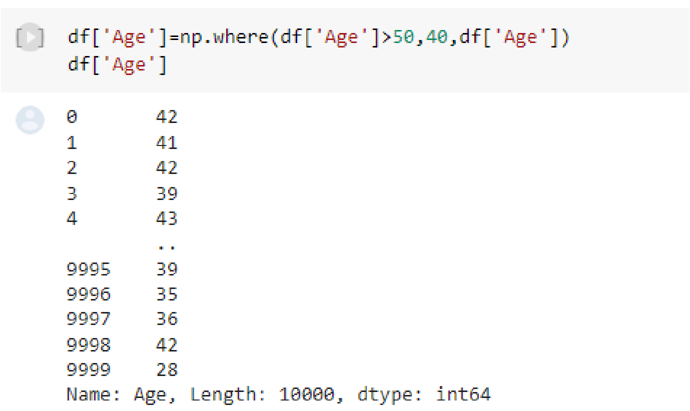


1.Find the outliers and replace the outliers.

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

    /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, tl
      FutureWarning
    <matplotlib.axes._subplots.AxesSubplot at 0x7f6d4aaf7110>
```

SOLUTION:

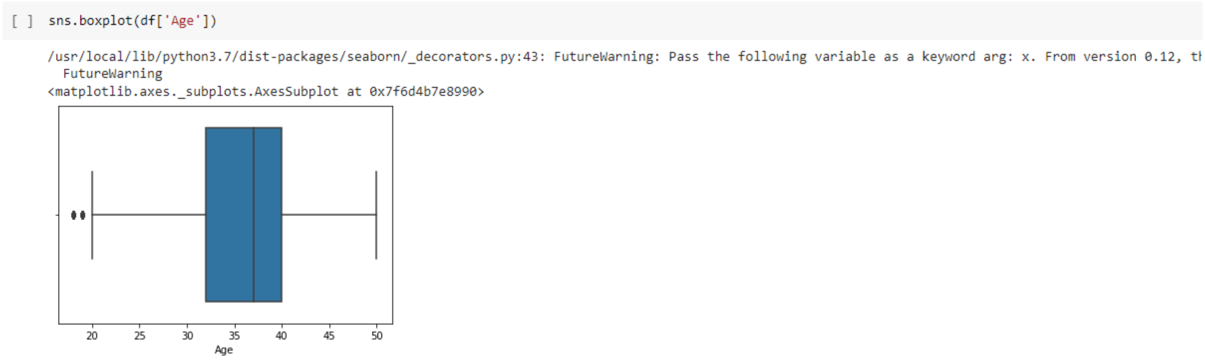
```
df['Age']=np.where(df['Age']>50,40,df['Age'])
df['Age']
```



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

    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
```

SOLUTION:

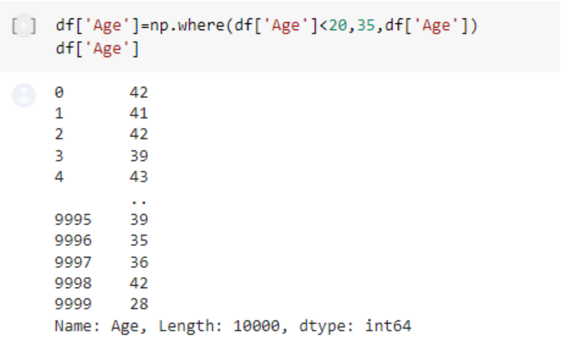
```
sns.boxplot(df['Age'])
```



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

    /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, th
      FutureWarning
    <matplotlib.axes._subplots.AxesSubplot at 0x7f6d4b7e8990>
```

SOLUTION:

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



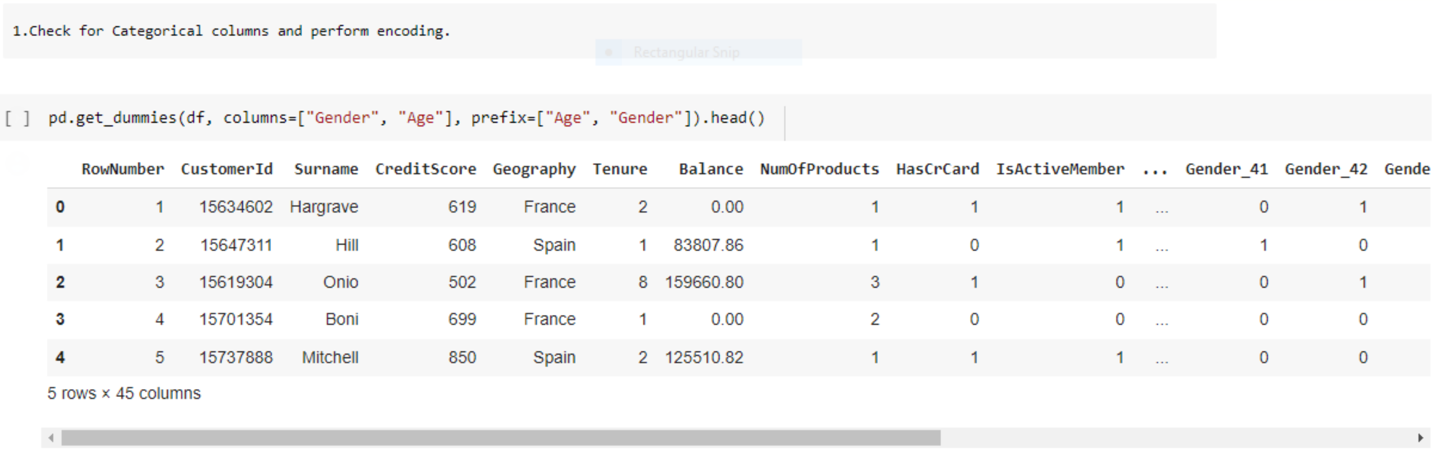
```
[ ] df['Age']=np.where(df['Age']<20,35,df['Age'])
    df['Age']

    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
```

Question-7:

Check for Categorical columns and perform encoding.

SOLUTION:

```
pd.get_dummies(df, columns=["Gender", "Age"], prefix=["Age", "Gender"]).hea
```

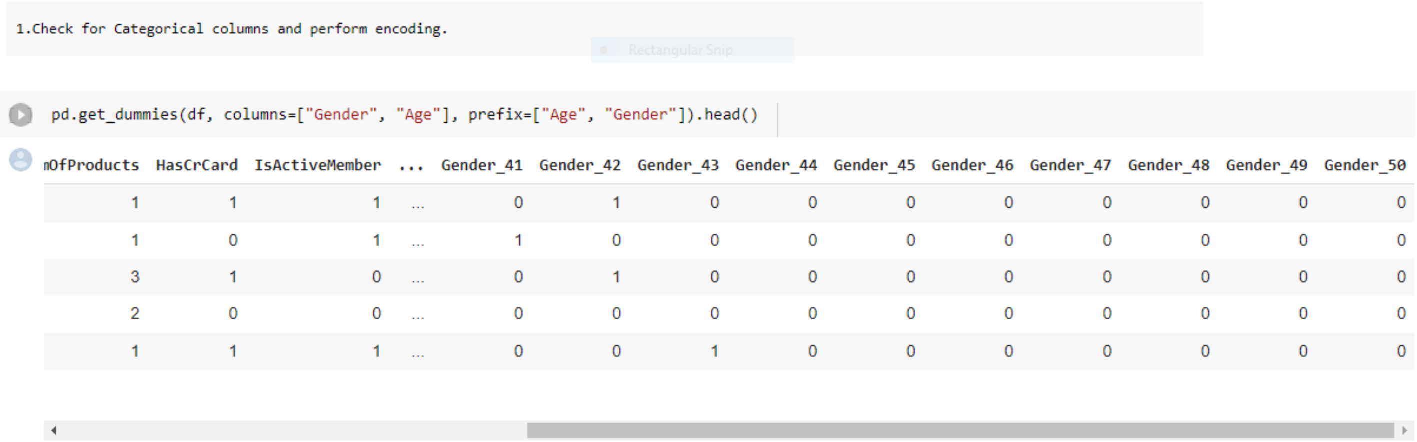


1.Check for Categorical columns and perform encoding.

```
[ ] pd.get_dummies(df, columns=["Gender", "Age"], prefix=["Age", "Gender"]).head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	...	Gender_41	Gender_42	Gende
0	1	15634602	Hargrave	619	France	2	0.00	1	1	1	...	0	1	
1	2	15647311	Hill	608	Spain	1	83807.86	1	0	1	...	1	0	
2	3	15619304	Onio	502	France	8	159660.80	3	1	0	...	0	1	
3	4	15701354	Boni	699	France	1	0.00	2	0	0	...	0	0	
4	5	15737888	Mitchell	850	Spain	2	125510.82	1	1	1	...	0	0	

5 rows × 45 columns



1.Check for Categorical columns and perform encoding.

```
pd.get_dummies(df, columns=["Gender", "Age"], prefix=["Age", "Gender"]).head()
```

nOfProducts	HasCrCard	IsActiveMember	...	Gender_41	Gender_42	Gender_43	Gender_44	Gender_45	Gender_46	Gender_47	Gender_48	Gender_49	Gender_50
1	1	1	...	0	1	0	0	0	0	0	0	0	0
1	0	1	...	1	0	0	0	0	0	0	0	0	0
3	1	0	...	0	1	0	0	0	0	0	0	0	0
2	0	0	...	0	0	0	0	0	0	0	0	0	0
1	1	1	...	0	0	1	0	0	0	0	0	0	0

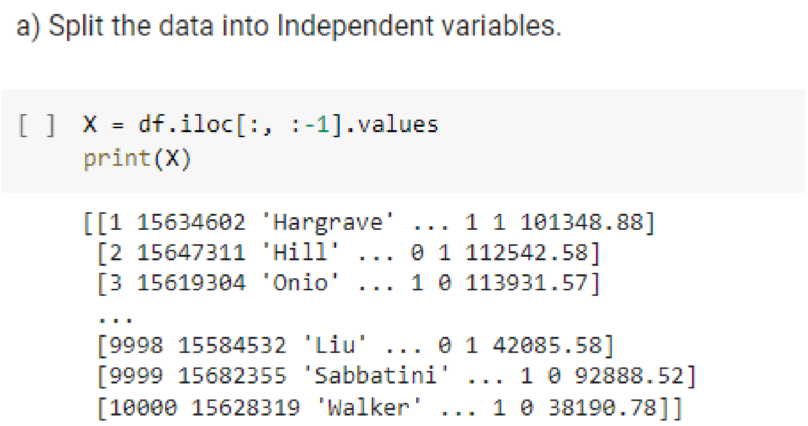
Question-8:

. Split the data into dependent and independent variables.

a) Split the data into Independent variables.

SOLUTION:

```
X = df.iloc[:, :-1].values
print(X)
```



a) Split the data into Independent variables.

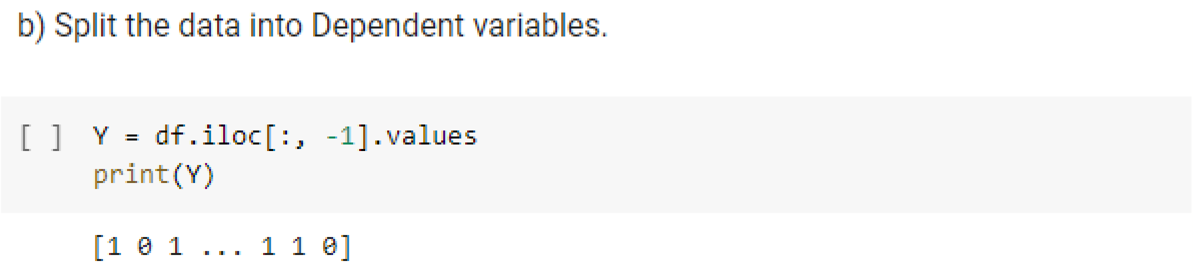
```
[ ]  X = df.iloc[:, :-1].values
     print(X)
```

```
[[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

SOLUTION:

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



b) Split the data into Dependent variables.

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

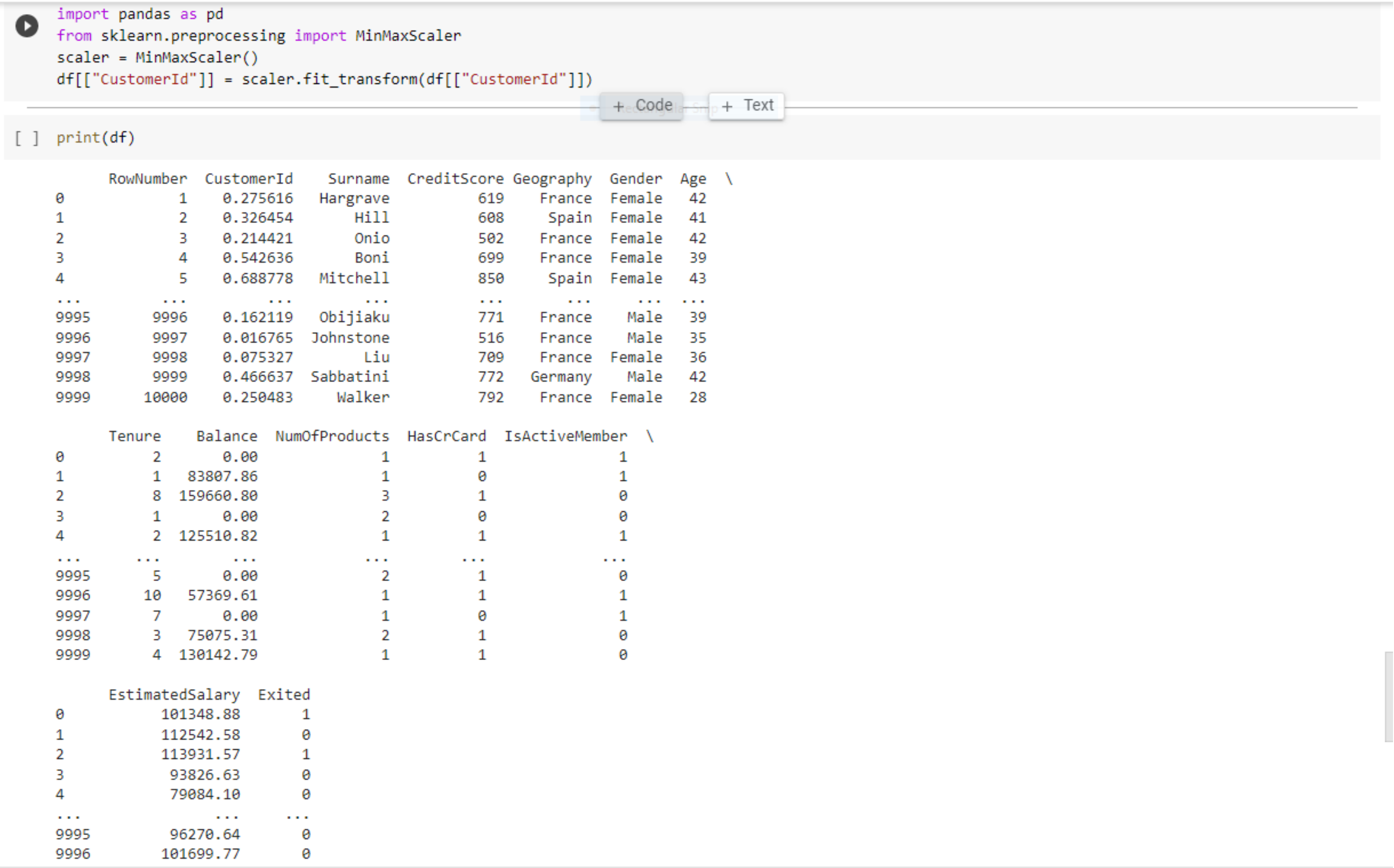
```
[1 0 1 ... 1 1 0]
```

Question-9:

Scale the independent variables

SOLUTION:

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



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

```
print(df)
```

```
      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
```

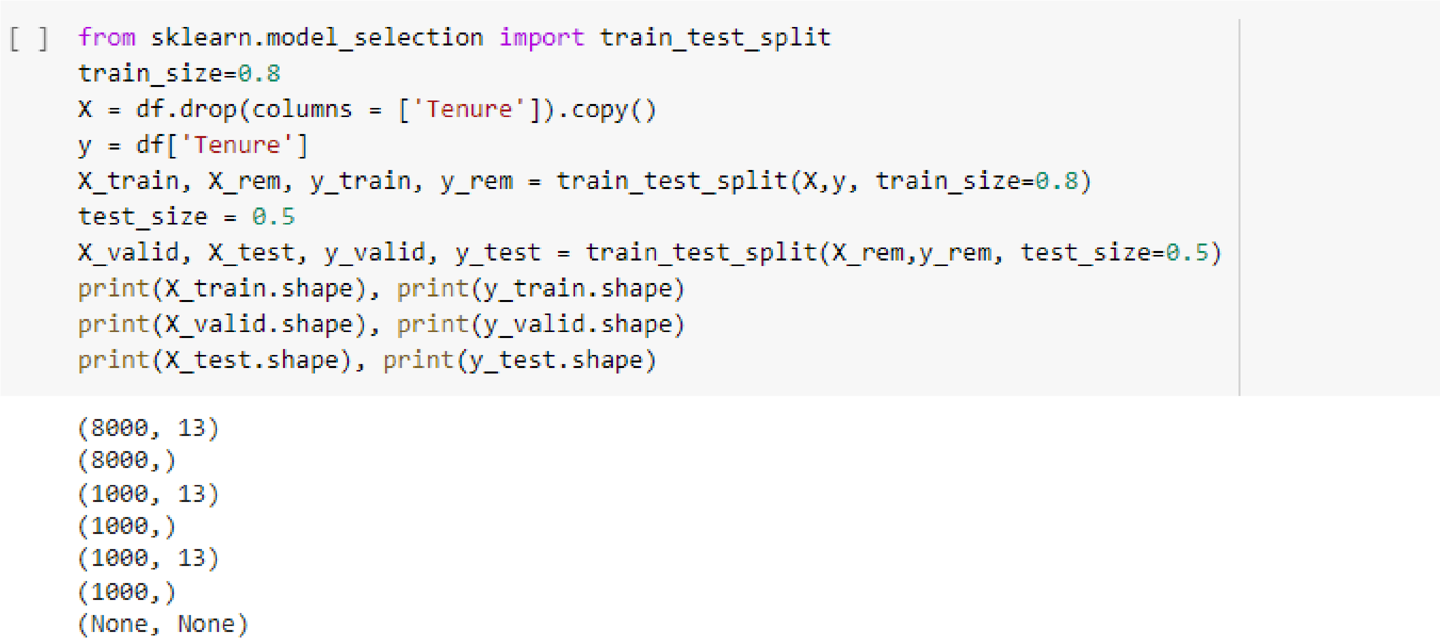
Question-10:

Split the data into training and testing

1. List item

SOLUTION:

```
from sklearn.model_selection import train_test_split

train_size=0.8
X = df.drop(columns = ['Tenure']).copy()
y = df['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)
```



```
from sklearn.model_selection import train_test_split
train_size=0.8
X = df.drop(columns = ['Tenure']).copy()
y = df['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,)
(None, None)
```