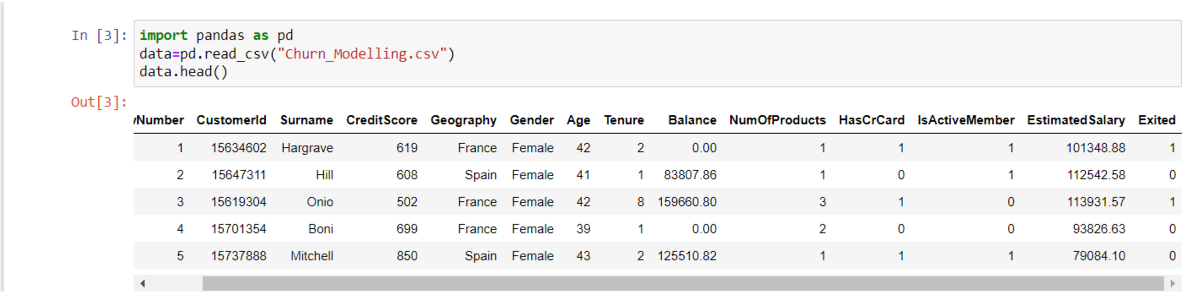
Assignment-2
Python Programming

Assignment Date	27 September 2022
Student Name	R. Subashini
Student Roll Number	820419106059
Maximum Mark	2 Mark

1. Download Data Set

2. Load the dataset

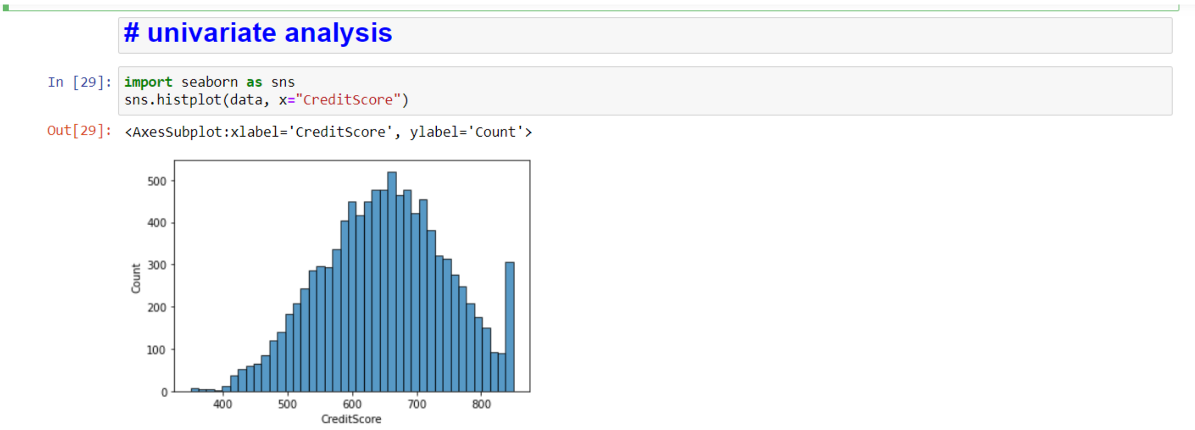


```
In [3]: import pandas as pd
        data=pd.read_csv("Churn_Modelling.csv")
        data.head()
```

Out[3]:

/Number	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
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	0
3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

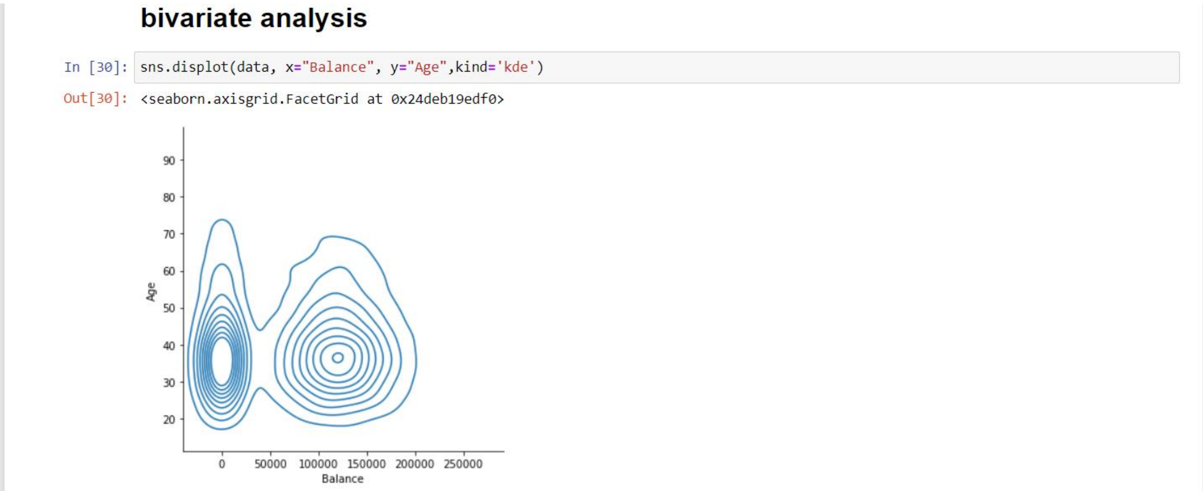
3. perform Below Visualization



univariate analysis

```
In [29]: import seaborn as sns
         sns.histplot(data, x="CreditScore")
```

```
Out[29]: <AxesSubplot:xlabel='CreditScore', ylabel='Count'>
```

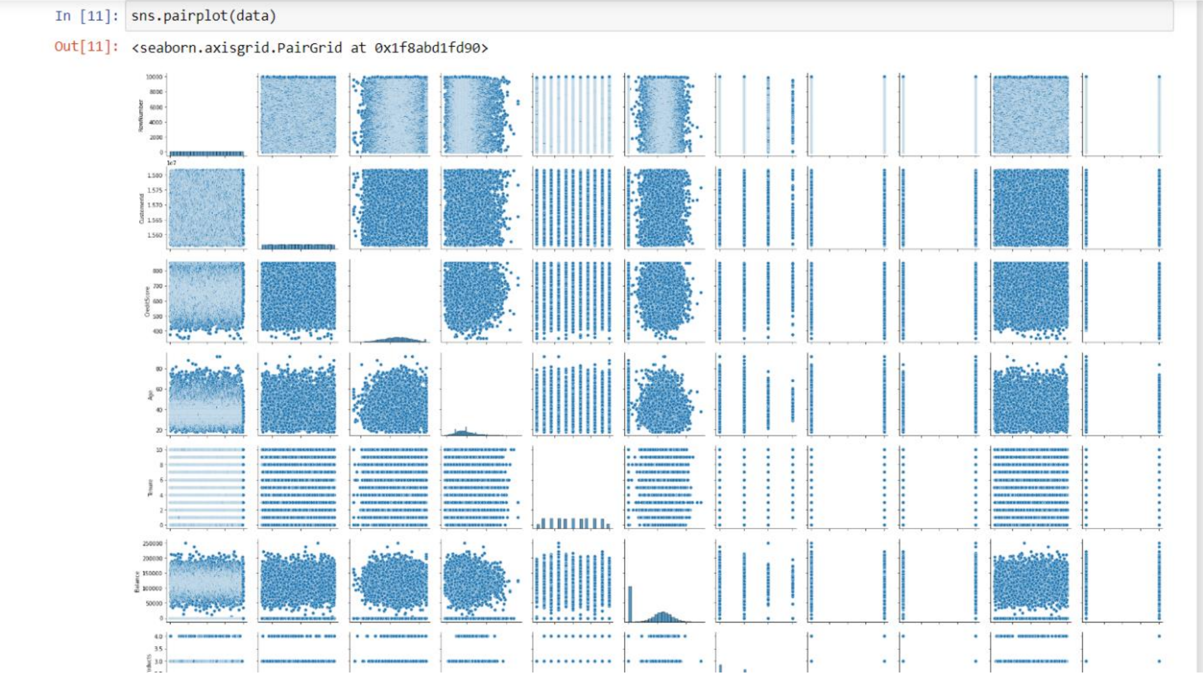


bivariate analysis

```
In [30]: sns.displot(data, x="Balance", y="Age",kind='kde')
```

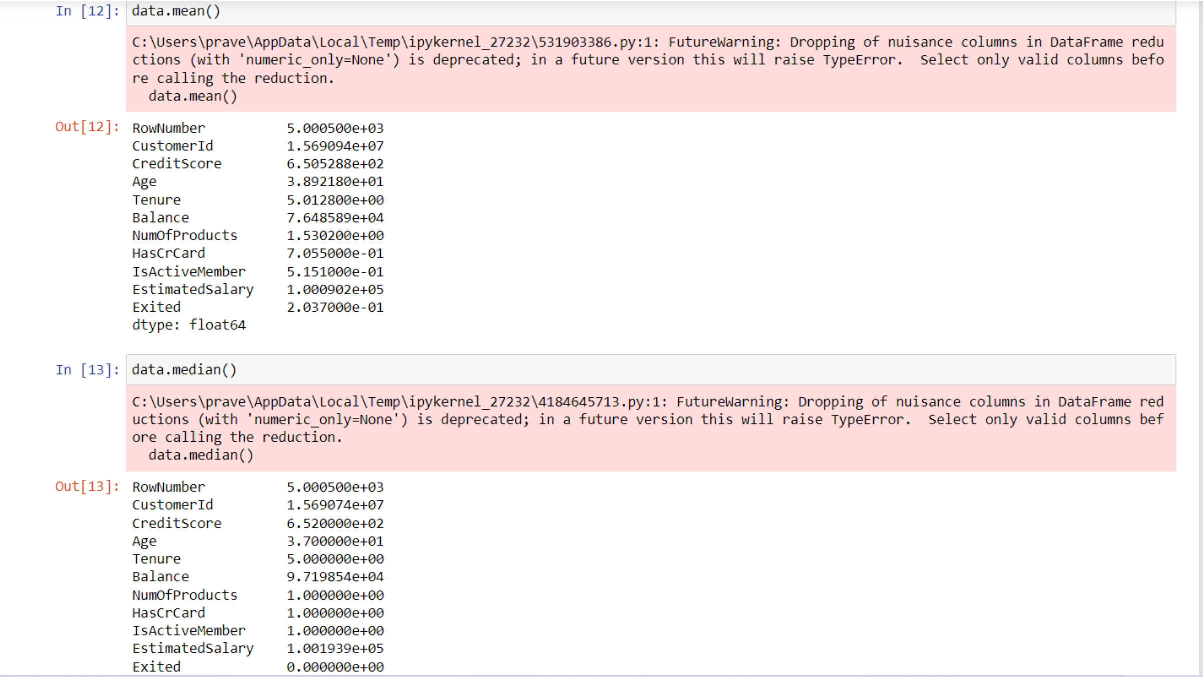
```
Out[30]: <seaborn.axisgrid.FacetGrid at 0x24deb19edf0>
```

Assignment-2
Python Programming

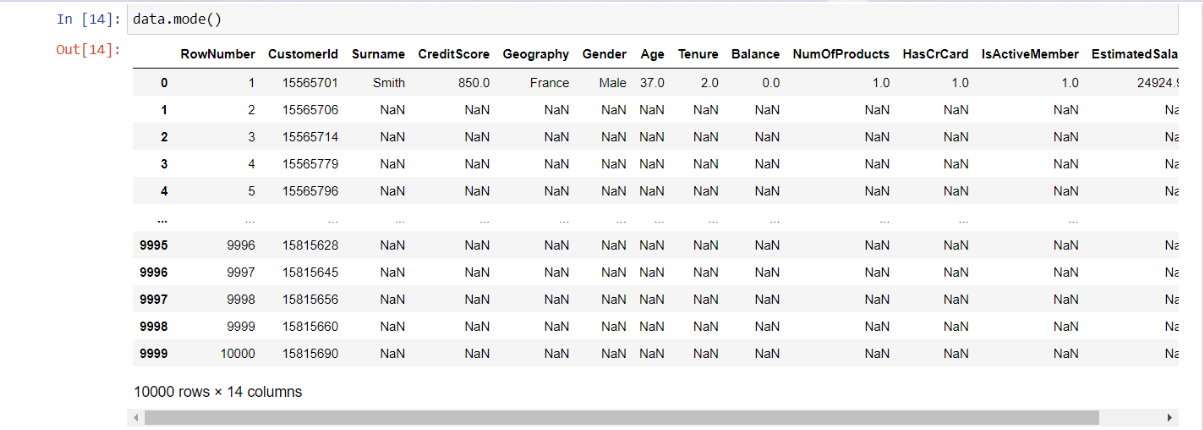


```
In [11]: sns.pairplot(data)

Out[11]: <seaborn.axisgrid.PairGrid at 0x1f8abd1fd90>
```

4. Perform descriptive Statistic on the dataset.



```
In [12]: data.mean()

C:\Users\prave\AppData\Local\Temp\ipykernel_27232\531903386.py:1: FutureWarning: Dropping of nuisance columns in DataFrame redu
ctions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns befo
re calling the reduction.
  data.mean()

Out[12]: 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 [13]: data.median()

C:\Users\prave\AppData\Local\Temp\ipykernel_27232\4184645713.py:1: FutureWarning: Dropping of nuisance columns in DataFrame red
uctions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns bef
ore calling the reduction.
  data.median()

Out[13]: 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
```

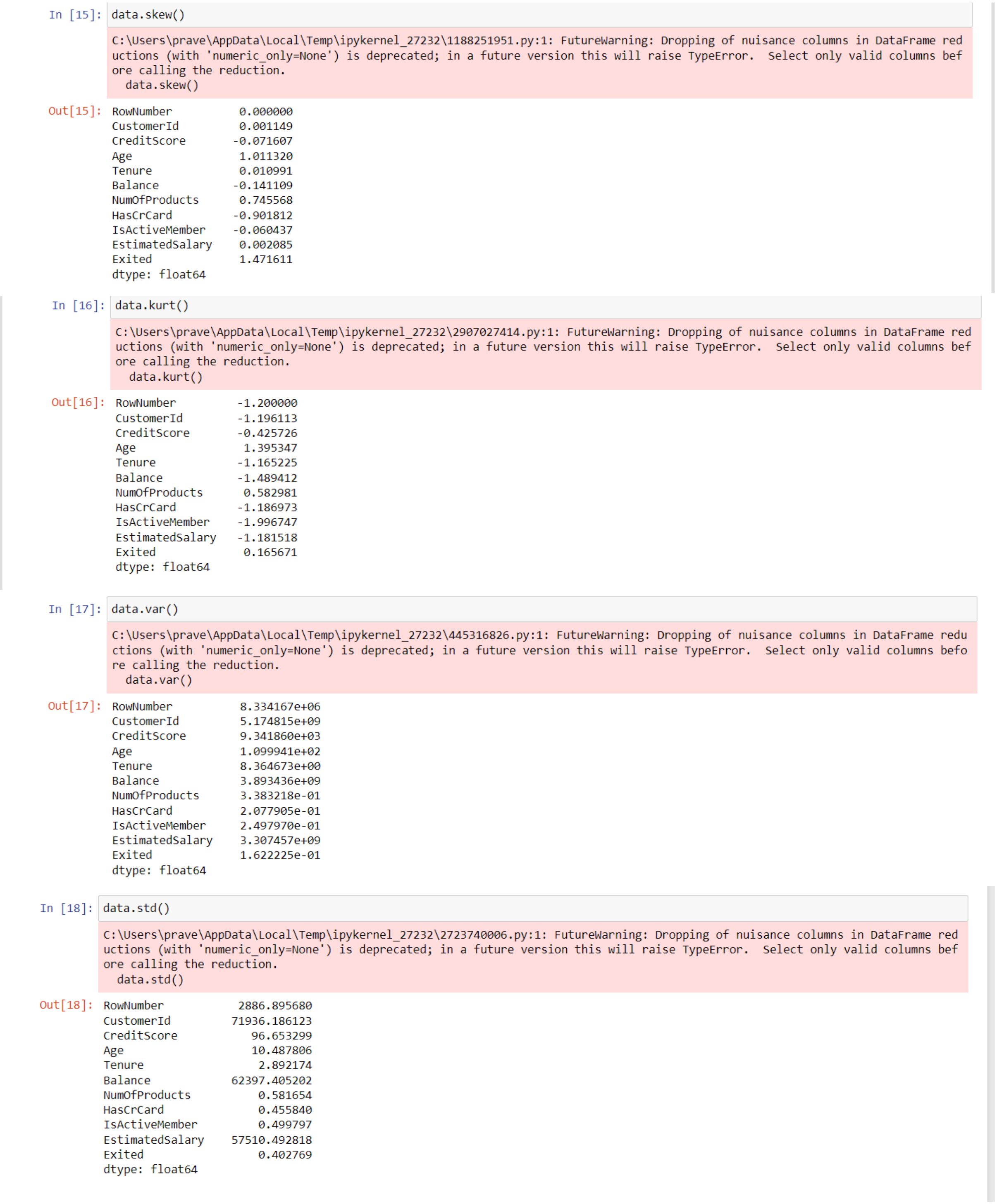


```
In [14]: data.mode()

Out[14]:
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSala
0	1	15565701	Smith	850.0	France	Male	37.0	2.0	0.0	1.0	1.0	1.0	24924.!
1	2	15565706	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
2	3	15565714	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
3	4	15565779	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
4	5	15565796	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
...	...	...	...	...	...	...	...	...	...	...	...	...	
9995	9996	15815628	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
9996	9997	15815645	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
9997	9998	15815656	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
9998	9999	15815660	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
9999	10000	15815690	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na

10000 rows × 14 columns

Assignment-2
Python Programming



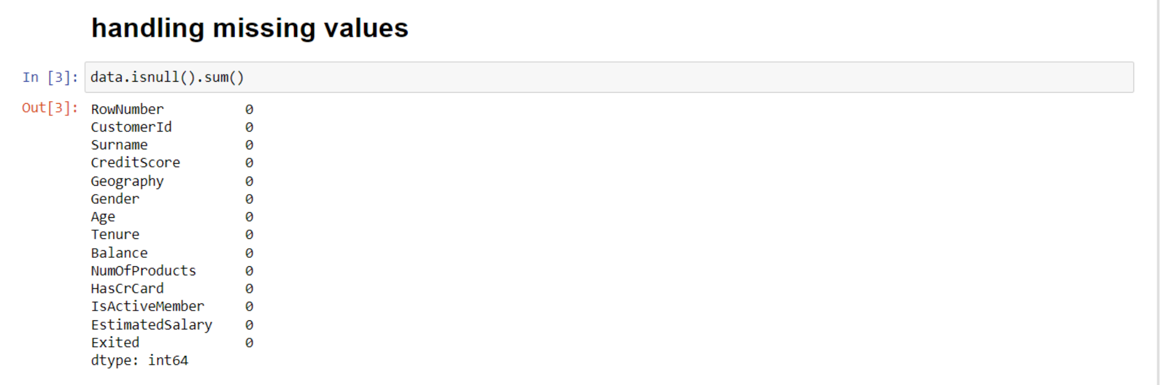
```
In [15]: data.skew()

C:\Users\prave\AppData\Local\Temp\ipykernel_27232\1188251951.py:1: FutureWarning: Dropping of nuisance columns in DataFrame red
uctions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns bef
ore calling the reduction.
  data.skew()

Out[15]: 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 [16]: data.kurt()

C:\Users\prave\AppData\Local\Temp\ipykernel_27232\2907027414.py:1: FutureWarning: Dropping of nuisance columns in DataFrame red
uctions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns bef
ore calling the reduction.
  data.kurt()

Out[16]: 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 [17]: data.var()

C:\Users\prave\AppData\Local\Temp\ipykernel_27232\445316826.py:1: FutureWarning: Dropping of nuisance columns in DataFrame redu
ctions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns befo
re calling the reduction.
  data.var()

Out[17]: 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 [18]: data.std()

C:\Users\prave\AppData\Local\Temp\ipykernel_27232\2723740006.py:1: FutureWarning: Dropping of nuisance columns in DataFrame red
uctions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns bef
ore calling the reduction.
  data.std()

Out[18]: 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
```

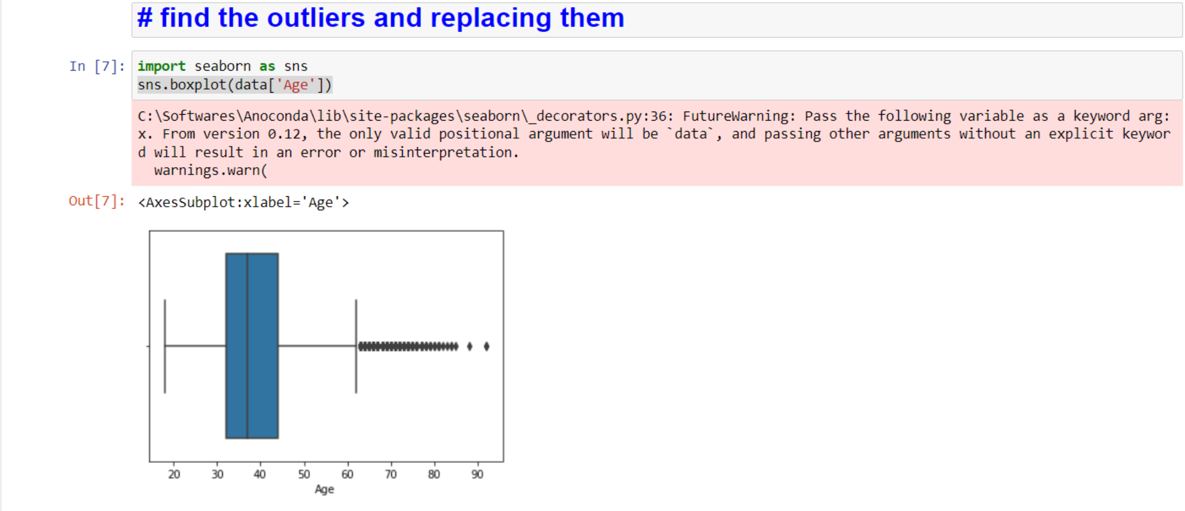
5. Handle the Missing Values.



handling missing values

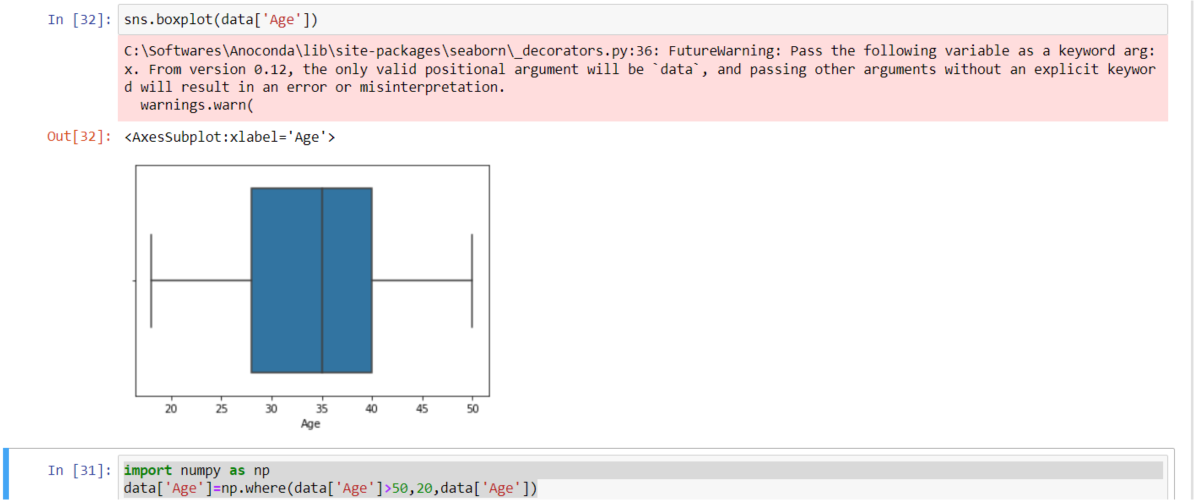
```
In [3]: data.isnull().sum()

Out[3]: 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
```

6. Find the outliers and replace the outliers



find the outliers and replacing them

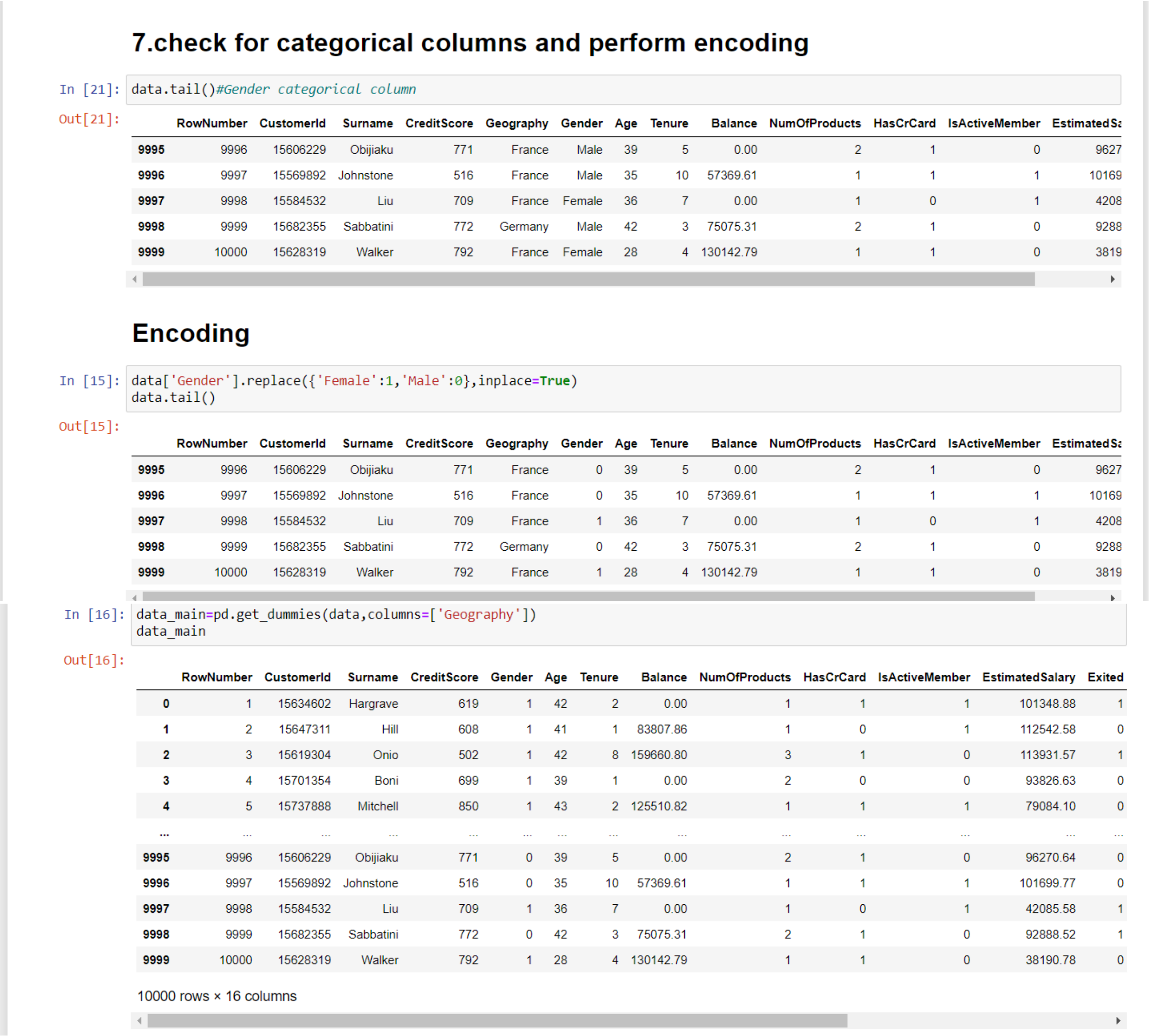
```
In [7]: import seaborn as sns
        sns.boxplot(data['Age'])

        C:\Softwares\Anoconda\lib\site-packages\seaborn\_decorators.py:36: 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 keywor
        d will result in an error or misinterpretation.
          warnings.warn(

Out[7]: <AxesSubplot:xlabel='Age'>
```



```
In [32]: sns.boxplot(data['Age'])

         C:\Softwares\Anoconda\lib\site-packages\seaborn\_decorators.py:36: 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 keywor
         d will result in an error or misinterpretation.
           warnings.warn(

Out[32]: <AxesSubplot:xlabel='Age'>
```

```
In [31]: import numpy as np
         data['Age']=np.where(data['Age']>50,20,data['Age'])
```

7. Check for Categorical Columns and encoding



7.check for categorical columns and perform encoding

In [21]:
```
data.tail()#Gender categorical column
```

Out[21]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSa
9995	9996	15606229	Obijiaku	771	France	Male	39	5	0.00	2	1	0	9627
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	10169
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	4208
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	9288
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	3819

Encoding

In [15]:
```
data['Gender'].replace({'Female':1,'Male':0},inplace=True)
data.tail()
```

Out[15]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSa
9995	9996	15606229	Obijiaku	771	France	0	39	5	0.00	2	1	0	9627
9996	9997	15569892	Johnstone	516	France	0	35	10	57369.61	1	1	1	10169
9997	9998	15584532	Liu	709	France	1	36	7	0.00	1	0	1	4208
9998	9999	15682355	Sabbatini	772	Germany	0	42	3	75075.31	2	1	0	9288
9999	10000	15628319	Walker	792	France	1	28	4	130142.79	1	1	0	3819

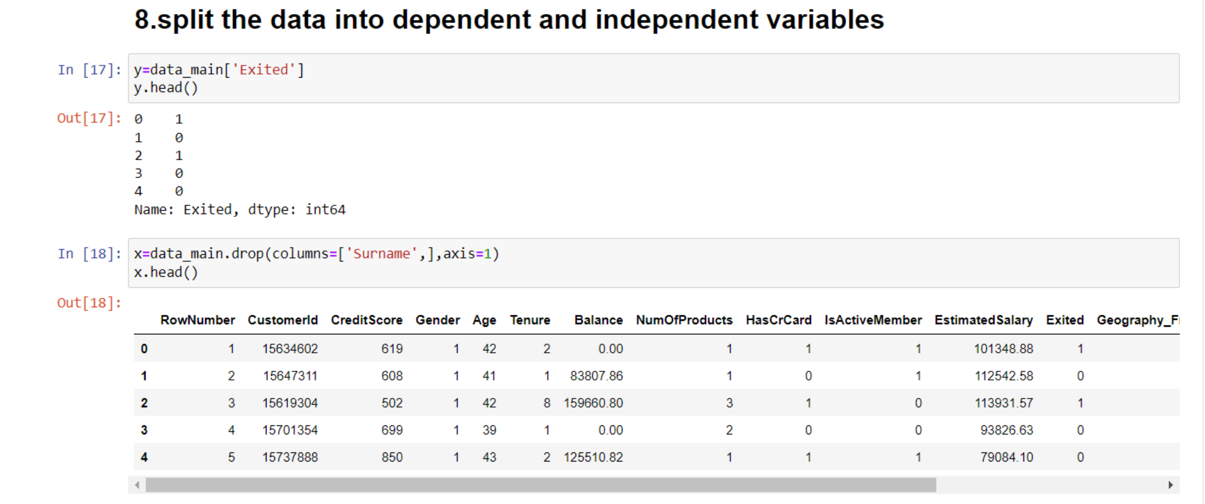
In [16]:
```
data_main=pd.get_dummies(data,columns=['Geography'])
data_main
```

Out[16]:

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

10000 rows × 16 columns

8. Split the data into dependent and independent variables.



8.split the data into dependent and independent variables

In [17]:
```
y=data_main['Exited']
y.head()
```

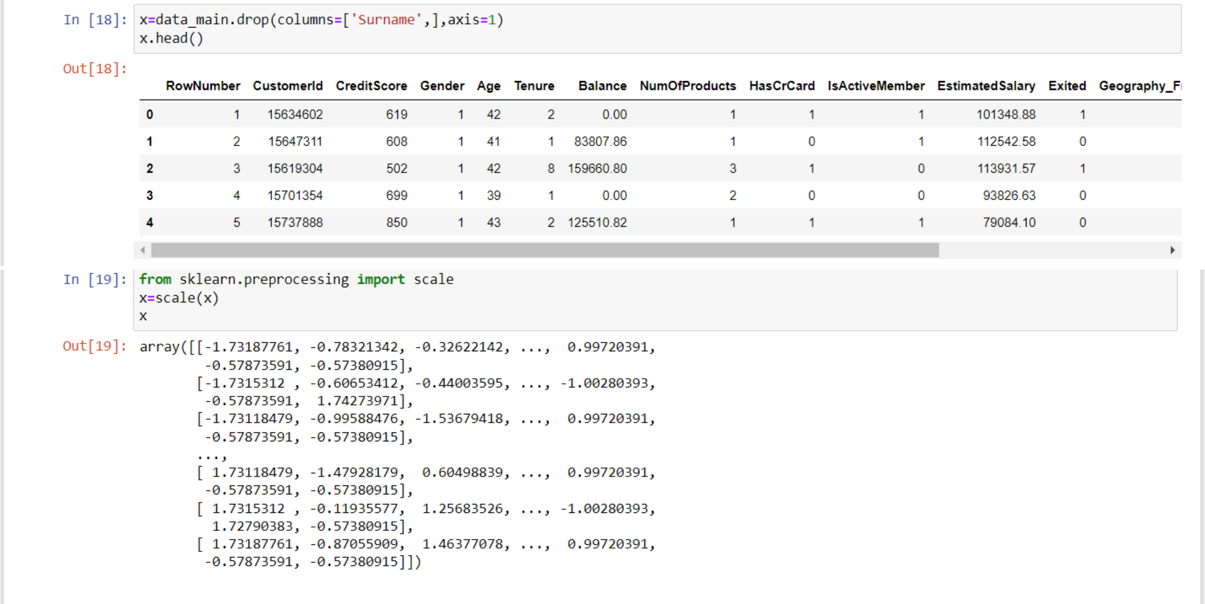
Out[17]:
```
0    1
1    0
2    1
3    0
4    0
Name: Exited, dtype: int64
```

In [18]:
```
x=data_main.drop(columns=['Surname',],axis=1)
x.head()
```

Out[18]:

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

9. Scale the independent variables



```
In [18]: x=data_main.drop(columns=['Surname',],axis=1)
         x.head()
```

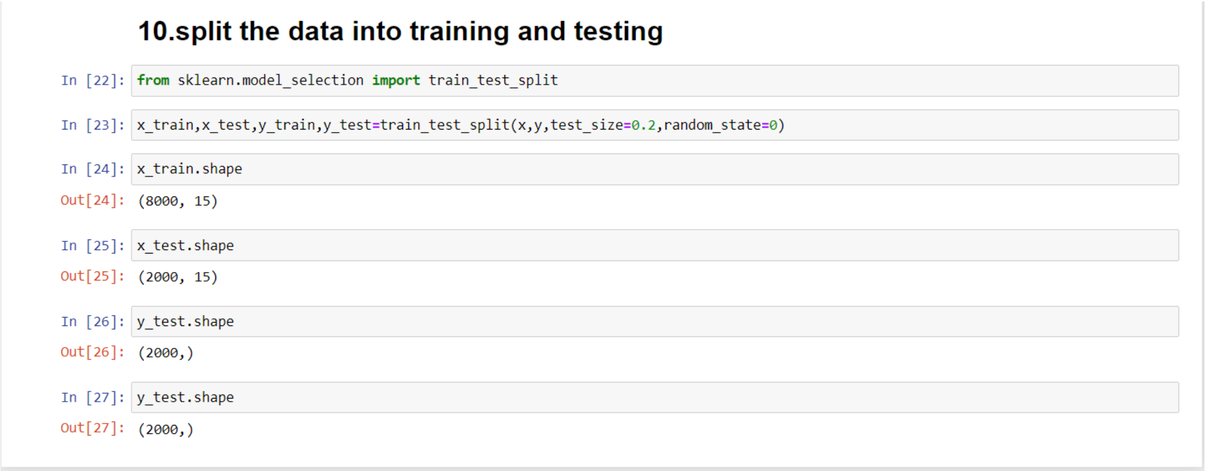
Out[18]:

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

```
In [19]: from sklearn.preprocessing import scale
         x=scale(x)
         x
```

```
Out[19]: array([[-1.73187761, -0.78321342, -0.32622142, ...,  0.99720391,
                 -0.57873591, -0.57380915],
                [-1.7315312 , -0.60653412, -0.44003595, ..., -1.00280393,
                 -0.57873591,  1.74273971],
                [-1.73118479, -0.99588476, -1.53679418, ...,  0.99720391,
                 -0.57873591, -0.57380915],
                ...,
                [ 1.73118479, -1.47928179,  0.60498839, ...,  0.99720391,
                 -0.57873591, -0.57380915],
                [ 1.7315312 , -0.11935577,  1.25683526, ..., -1.00280393,
                  1.72790383, -0.57380915],
                [ 1.73187761, -0.87055909,  1.46377078, ...,  0.99720391,
                 -0.57873591, -0.57380915]])
```

10. Split the data into training and testing



10.split the data into training and testing

```
In [22]: from sklearn.model_selection import train_test_split

In [23]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)

In [24]: x_train.shape

Out[24]: (8000, 15)

In [25]: x_test.shape

Out[25]: (2000, 15)

In [26]: y_test.shape

Out[26]: (2000,)

In [27]: y_test.shape

Out[27]: (2000,)
```