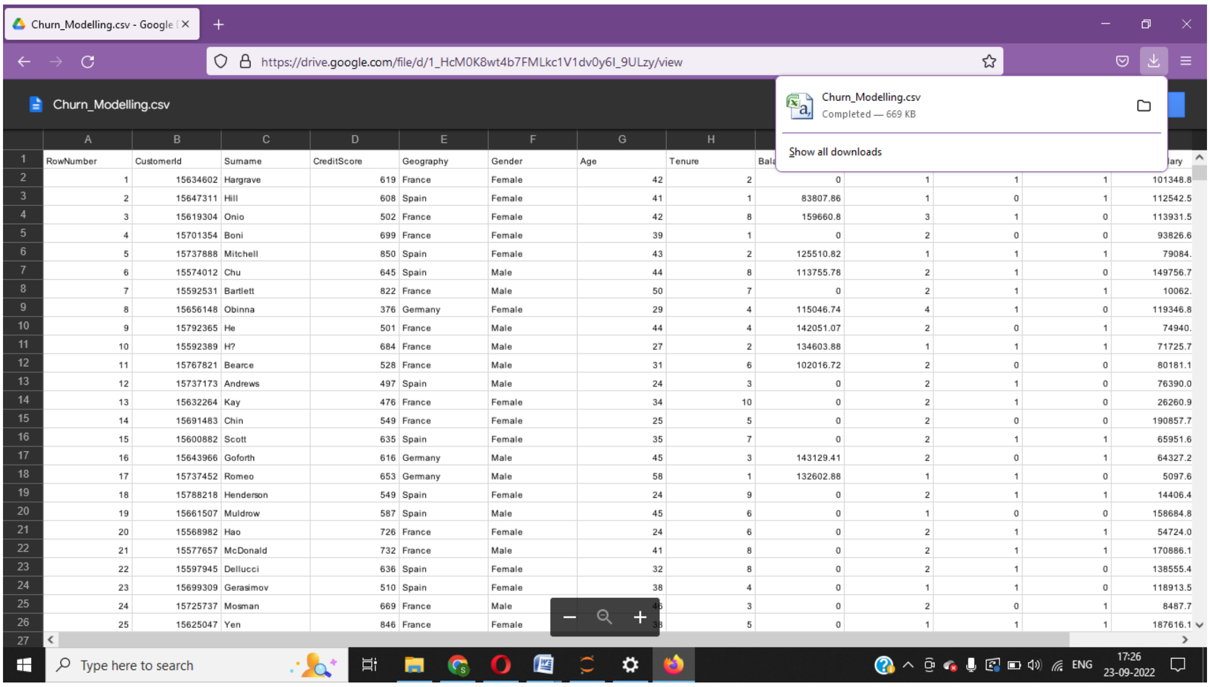
Assignment -2
Data Visualization and Data Pre-processing

Assignment Date	17 September 2022
Student Name	Logeshkumar R
Student Roll Number	727719EUCS074
Maximum Marks	2 Marks

Question-1:

Download the dataset: Dataset



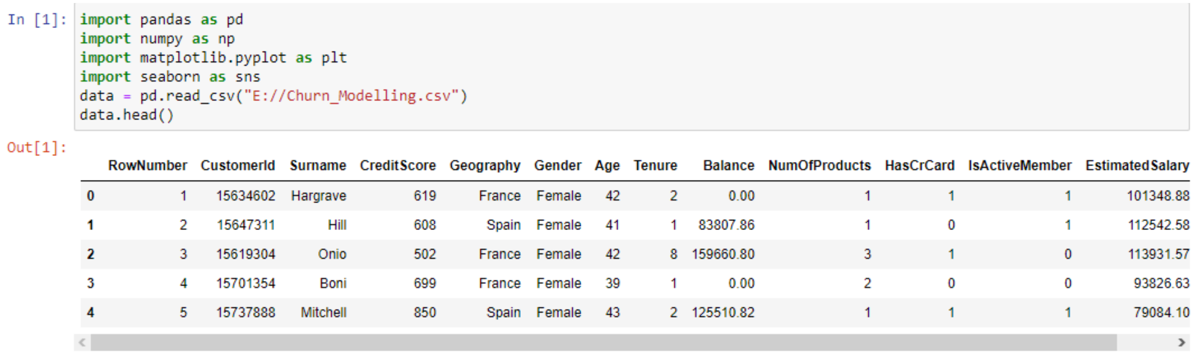
1	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Bala				lary
2	1	15634602	Hargrave	619	France	Female	42	2	0	1	1	1	101348.8
3	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.5
4	3	15619304	Onio	502	France	Female	42	8	159660.8	3	1	0	113931.5
5	4	15701354	Boni	699	France	Female	39	1	0	2	0	0	93826.6
6	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.
7	6	15574012	Chu	645	Spain	Male	44	8	113755.78	2	1	0	149756.7
8	7	15592531	Bartlett	822	France	Male	50	7	0	2	1	1	10062.
9	8	15656148	Obinna	376	Germany	Female	29	4	115046.74	4	1	0	119346.8
10	9	15792365	He	501	France	Male	44	4	142051.07	2	0	1	74940.
11	10	15592389	H?	684	France	Male	27	2	134603.88	1	1	1	71725.7
12	11	15767821	Bearce	528	France	Male	31	6	102016.72	2	0	0	80181.1
13	12	15737173	Andrews	497	Spain	Male	24	3	0	2	1	0	76390.0
14	13	15632264	Kay	476	France	Female	34	10	0	2	1	0	26260.9
15	14	15691483	Chin	549	France	Female	25	5	0	2	0	0	190857.7
16	15	15600882	Scott	635	Spain	Female	35	7	0	2	1	1	65951.6
17	16	15643966	Goforth	616	Germany	Male	45	3	143129.41	2	0	1	64327.2
18	17	15737452	Romeo	653	Germany	Male	58	1	132602.88	1	1	0	5097.6
19	18	15788218	Henderson	549	Spain	Female	24	9	0	2	1	1	14406.4
20	19	15661507	Muldrow	587	Spain	Male	45	6	0	1	0	0	158684.8
21	20	15568982	Hao	726	France	Female	24	6	0	2	1	1	54724.0
22	21	15577657	McDonald	732	France	Male	41	8	0	2	1	1	170886.1
23	22	15597945	Dellucci	636	Spain	Female	32	8	0	2	1	0	138555.4
24	23	15699309	Gerasimov	510	Spain	Female	38	4	0	1	1	0	118913.5
25	24	15725737	Mosman	669	France	Male	46	3	0	2	0	1	8487.7
26	25	15625047	Yen	846	France	Female	38	5	0	1	1	1	187616.1
27													

Question-2:

Load the dataset.

Solution:

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

data = pd.read_csv("E://Churn_Modelling.csv")
data.head()
```



```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        data = pd.read_csv("E://Churn_Modelling.csv")
        data.head()
```

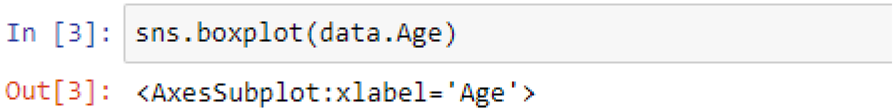
Out[1]:

	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

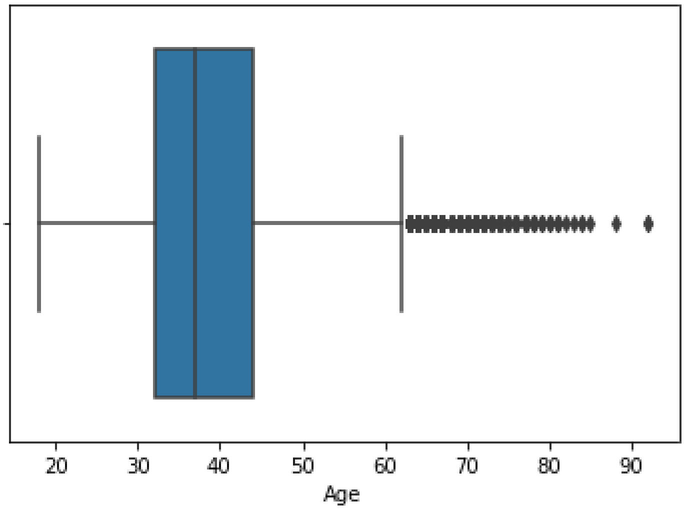
Question-3:

Perform Below Visualizations.

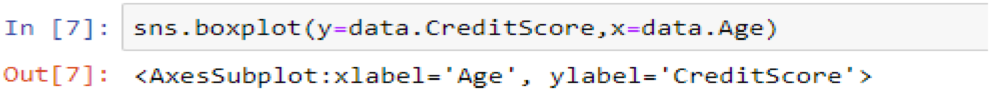
● Univariate Analysis



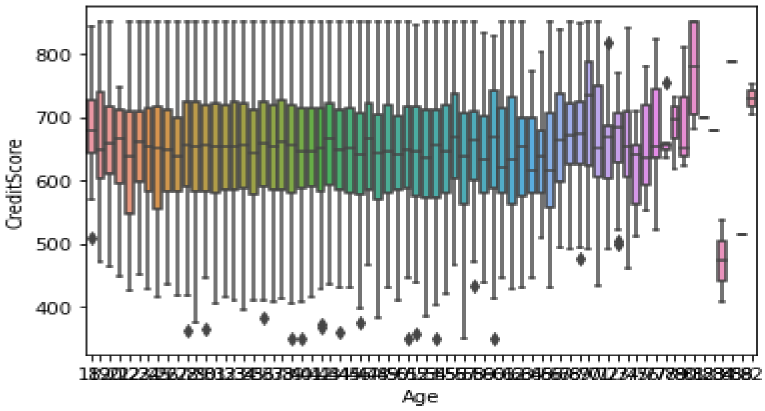
```
In [3]: sns.boxplot(data.Age)
Out[3]: <AxesSubplot:xlabel='Age'>
```



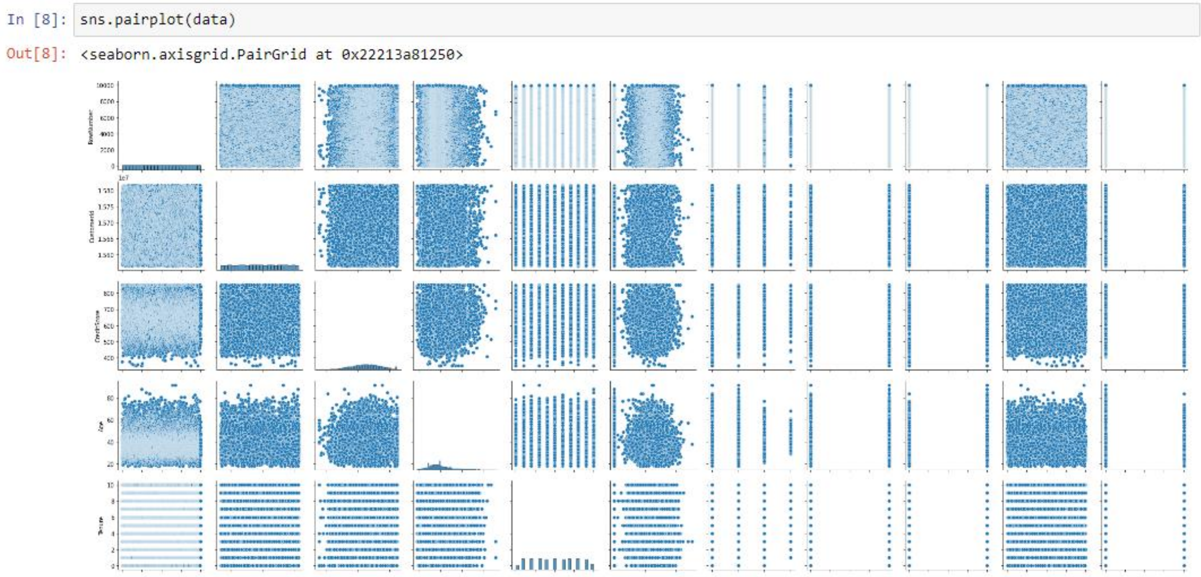
● Bi - Variate Analysis



```
In [7]: sns.boxplot(y=data.CreditScore,x=data.Age)
Out[7]: <AxesSubplot:xlabel='Age', ylabel='CreditScore'>
```



- Multi - Variate Analysis

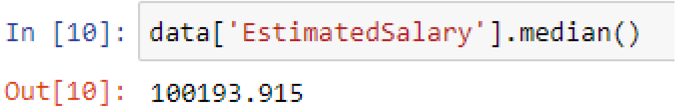


```
In [8]: sns.pairplot(data)

Out[8]: <seaborn.axisgrid.PairGrid at 0x22213a81250>
```

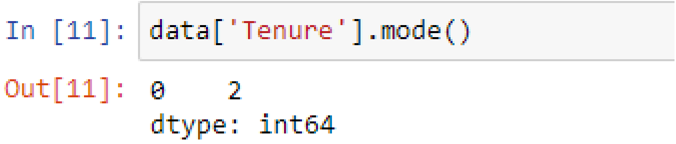
Question-4:

Perform descriptive statistics on the dataset.

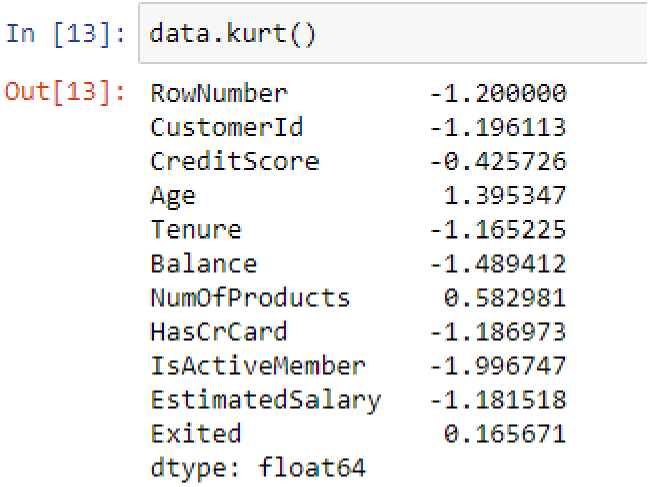
```
In [9]: data['NumOfProducts'].mean()

Out[9]: 1.5302
```



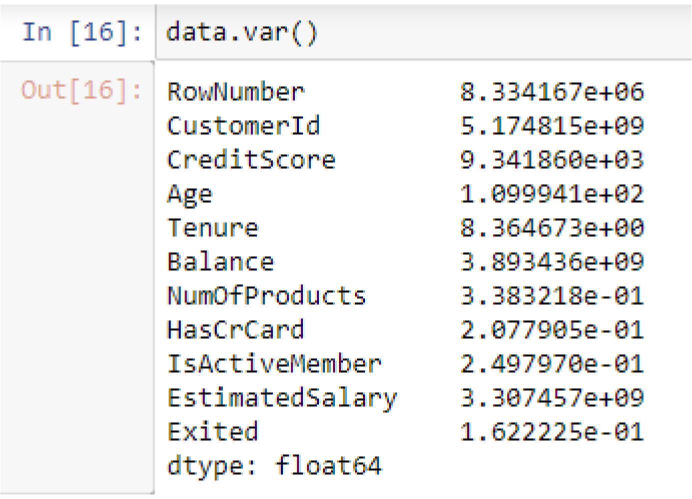
```
In [10]: data['EstimatedSalary'].median()

Out[10]: 100193.915
```



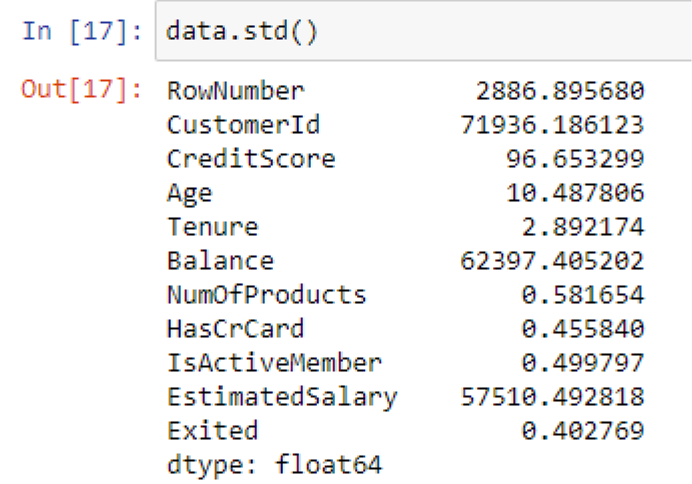
```
In [11]: data['Tenure'].mode()

Out[11]: 0    2
         dtype: int64
```



```
In [13]: data.kurt()

Out[13]: 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
```



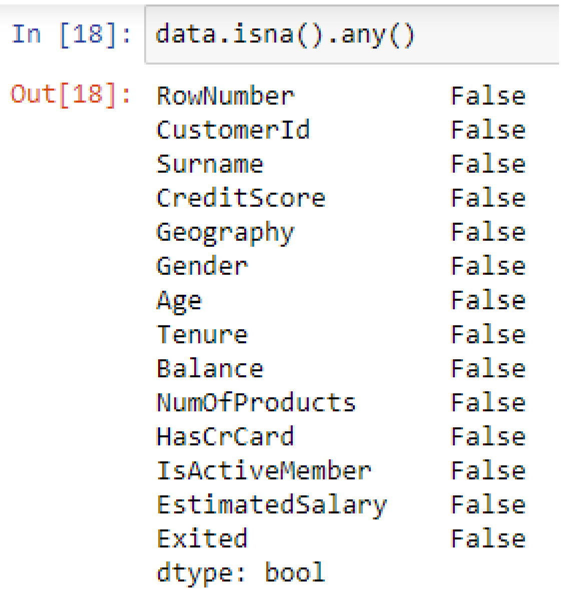
```
In [16]: data.var()

Out[16]: 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
```



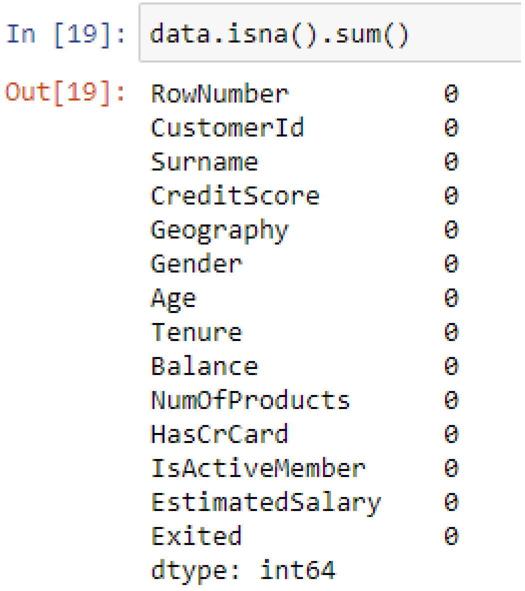
```
In [17]: data.std()

Out[17]: 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
```

Question-5:

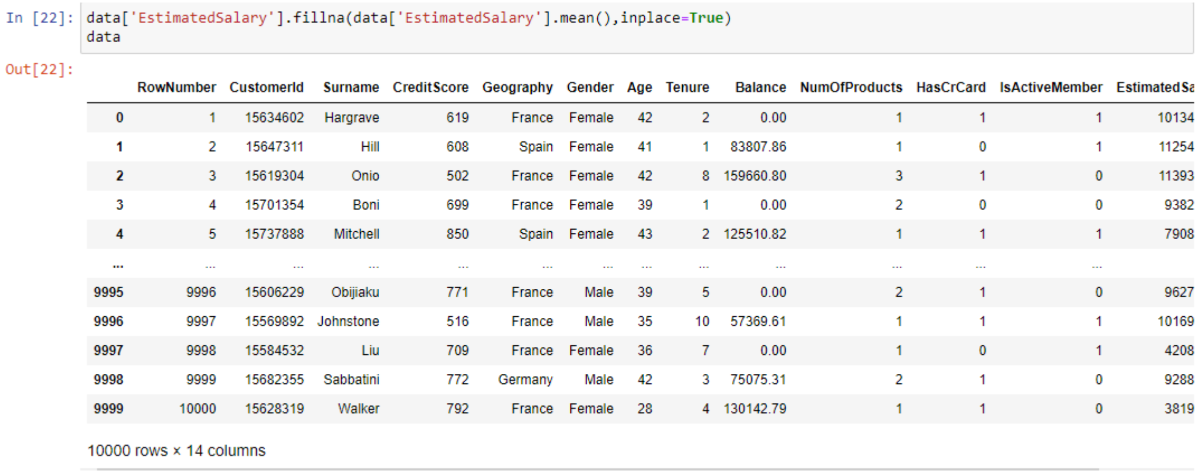
Handle the Missing values.



```
In [18]: data.isna().any()

Out[18]: 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
```



```
In [19]: data.isna().sum()

Out[19]: 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
```



```
In [22]: data['EstimatedSalary'].fillna(data['EstimatedSalary'].mean(),inplace=True)
         data
```

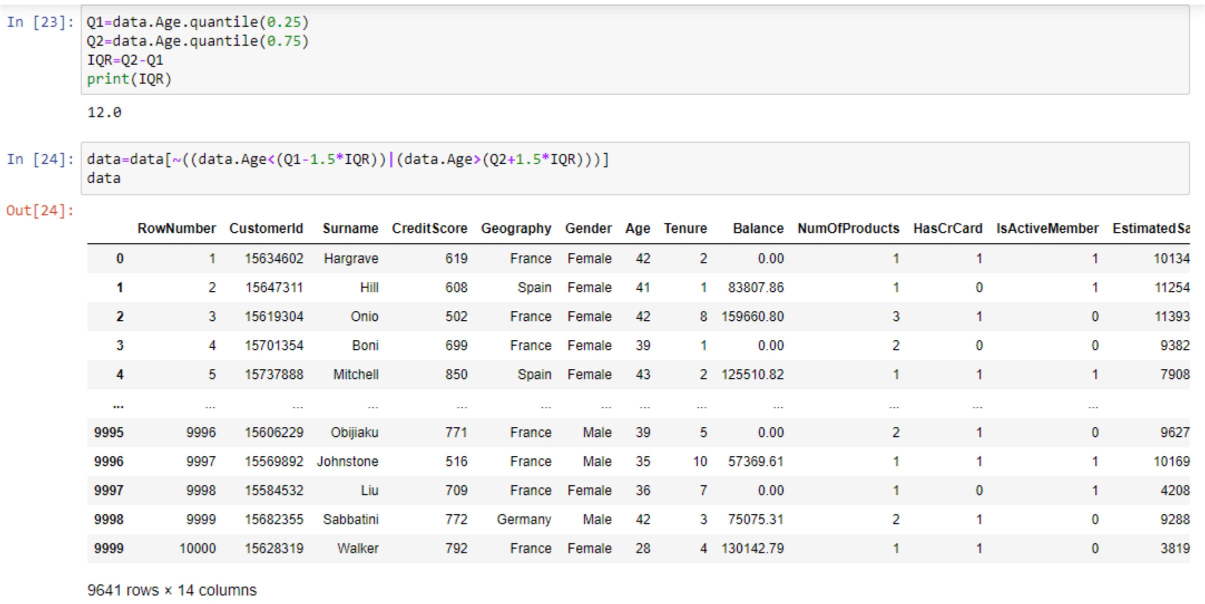
Out[22]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSa
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	10134
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	11254
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	11393
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	9382
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	7908
...	...	...	...	...	...	...	...	...	...	...	...	...	
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

10000 rows × 14 columns

Question-6:

Find the outliers and replace the outliers



```
In [23]: Q1=data.Age.quantile(0.25)
         Q2=data.Age.quantile(0.75)
         IQR=Q2-Q1
         print(IQR)

         12.0
```

```
In [24]: data=data[~((data.Age<(Q1-1.5*IQR))|(data.Age>(Q2+1.5*IQR)))]
         data
```

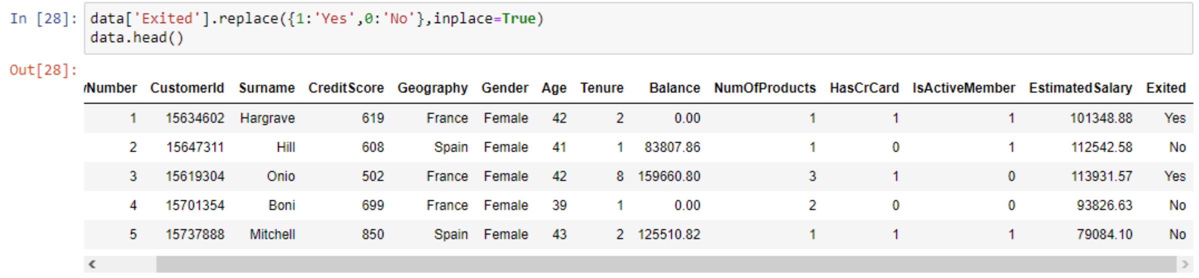
Out[24]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSa
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	10134
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	11254
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	11393
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	9382
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	7908
...	...	...	...	...	...	...	...	...	...	...	...	...	
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

9641 rows × 14 columns

Question-7:

Check for Categorical columns and perform encoding.



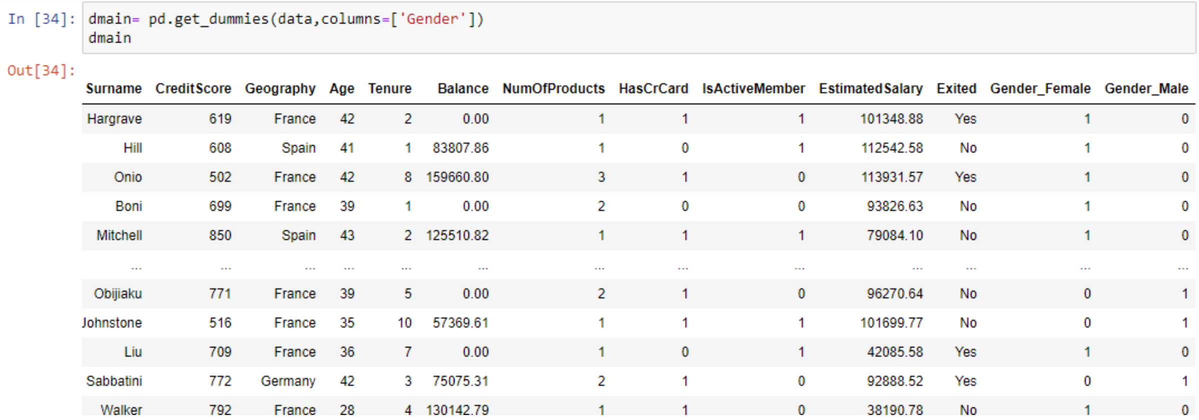
```
In [28]: data['Exited'].replace({1:'Yes',0:'No'},inplace=True)
         data.head()
```

Out[28]:

/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	Yes
2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	No
3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	Yes
4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	No
5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	No

Question-8:

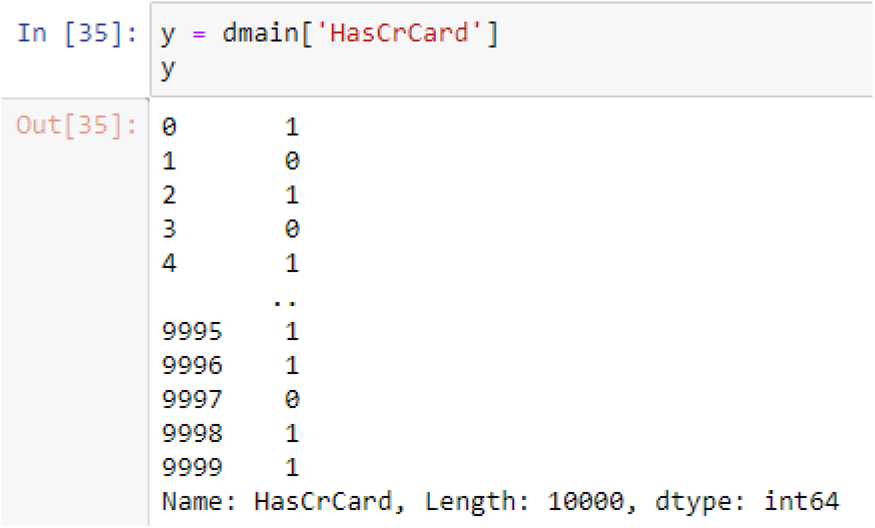
Split the data into dependent and independent variables.



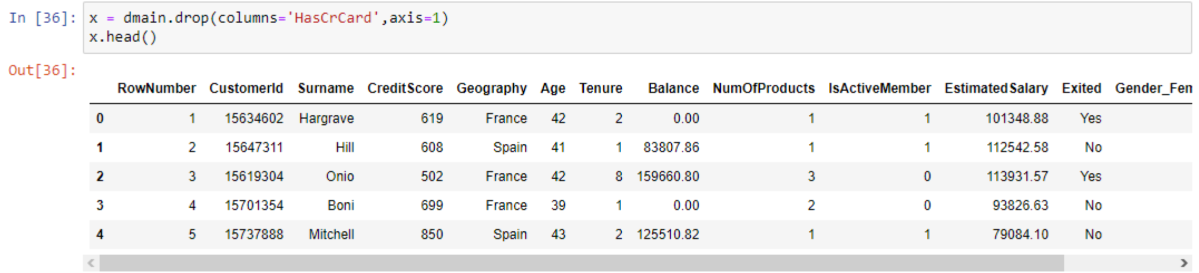
```
In [34]: dmain= pd.get_dummies(data,columns=['Gender'])
         dmain
```

Out[34]:

Surname	CreditScore	Geography	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Gender_Female	Gender_Male
Hargrave	619	France	42	2	0.00	1	1	1	101348.88	Yes	1	0
Hill	608	Spain	41	1	83807.86	1	0	1	112542.58	No	1	0
Onio	502	France	42	8	159660.80	3	1	0	113931.57	Yes	1	0
Boni	699	France	39	1	0.00	2	0	0	93826.63	No	1	0
Mitchell	850	Spain	43	2	125510.82	1	1	1	79084.10	No	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...
Obijiaku	771	France	39	5	0.00	2	1	0	96270.64	No	0	1
Johnstone	516	France	35	10	57369.61	1	1	1	101699.77	No	0	1
Liu	709	France	36	7	0.00	1	0	1	42085.58	Yes	1	0
Sabbatini	772	Germany	42	3	75075.31	2	1	0	92888.52	Yes	0	1
Walker	792	France	28	4	130142.79	1	1	0	38190.78	No	1	0



```
In [35]: y = dmain['HasCrCard']
         y

Out[35]: 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
```



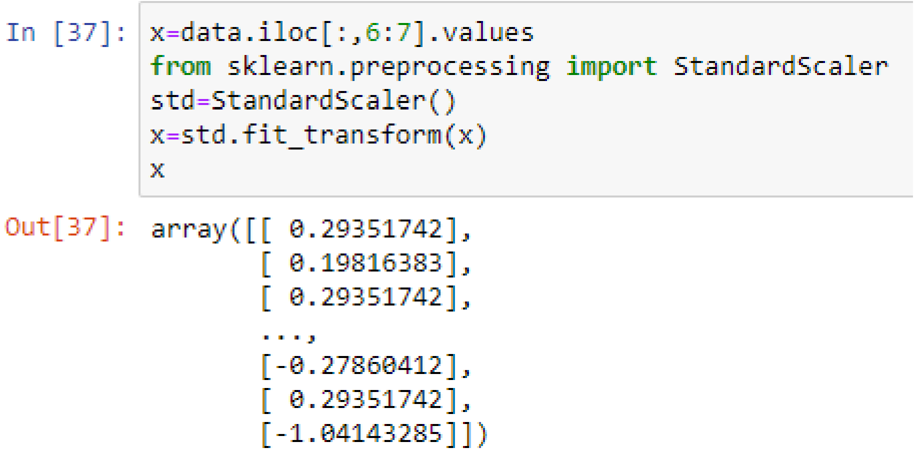
```
In [36]: x = dmain.drop(columns='HasCrCard',axis=1)
         x.head()
```

Out[36]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Age	Tenure	Balance	NumOfProducts	IsActiveMember	EstimatedSalary	Exited	Gender_Fen
0	1	15634602	Hargrave	619	France	42	2	0.00	1	1	101348.88	Yes	
1	2	15647311	Hill	608	Spain	41	1	83807.86	1	1	112542.58	No	
2	3	15619304	Onio	502	France	42	8	159660.80	3	0	113931.57	Yes	
3	4	15701354	Boni	699	France	39	1	0.00	2	0	93826.63	No	
4	5	15737888	Mitchell	850	Spain	43	2	125510.82	1	1	79084.10	No	

Question-9:

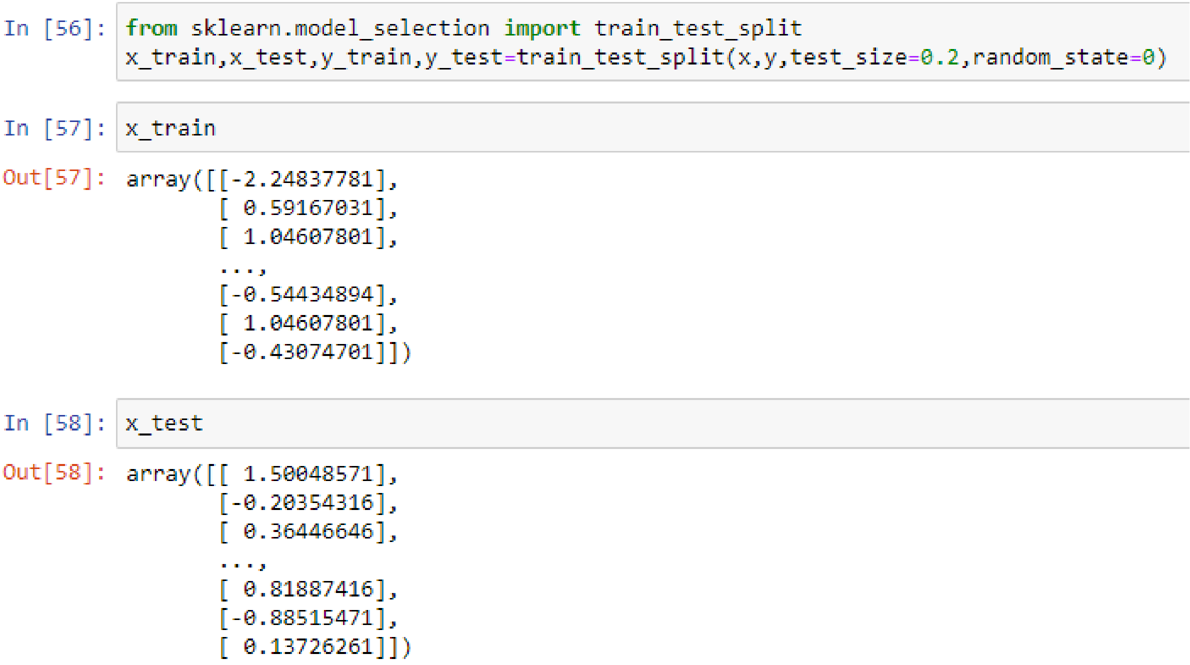
Scale the independent variables



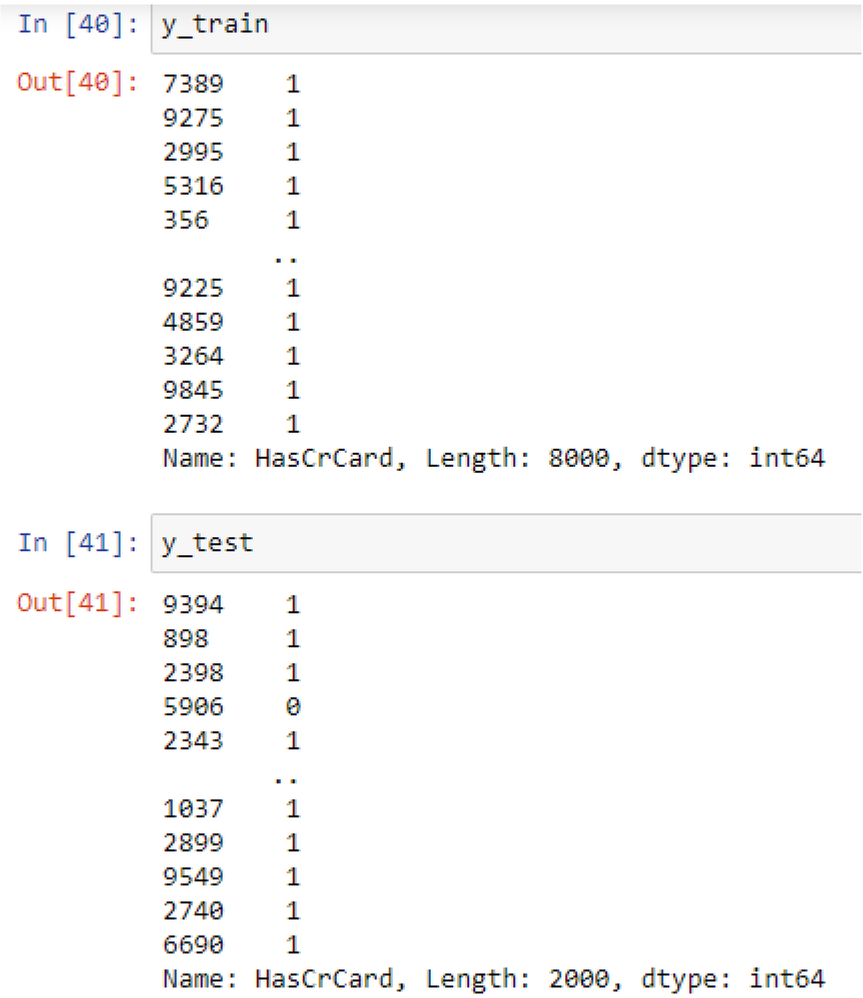
```
In [37]: x=data.iloc[:,6:7].values
         from sklearn.preprocessing import StandardScaler
         std=StandardScaler()
         x=std.fit_transform(x)
         x

Out[37]: array([[ 0.29351742],
                [ 0.19816383],
                [ 0.29351742],
                ...,
                [-0.27860412],
                [ 0.29351742],
                [-1.04143285]])
```

Question-10:

Split the data into training and testing



```
In [56]: 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)

In [57]: x_train

Out[57]: array([[-2.24837781],
                [ 0.59167031],
                [ 1.04607801],
                ...,
                [-0.54434894],
                [ 1.04607801],
                [-0.43074701]])

In [58]: x_test

Out[58]: array([[ 1.50048571],
                [-0.20354316],
                [ 0.36446646],
                ...,
                [ 0.81887416],
                [-0.88515471],
                [ 0.13726261]])
```



```
In [40]: y_train
```

```
Out[40]: 7389    1
         9275    1
         2995    1
         5316    1
         356     1
                ..
         9225    1
         4859    1
         3264    1
         9845    1
         2732    1
         Name: HasCrCard, Length: 8000, dtype: int64
```

```
In [41]: y_test
```

```
Out[41]: 9394    1
         898     1
         2398    1
         5906    0
         2343    1
                ..
         1037    1
         2899    1
         9549    1
         2740    1
         6690    1
         Name: HasCrCard, Length: 2000, dtype: int64
```