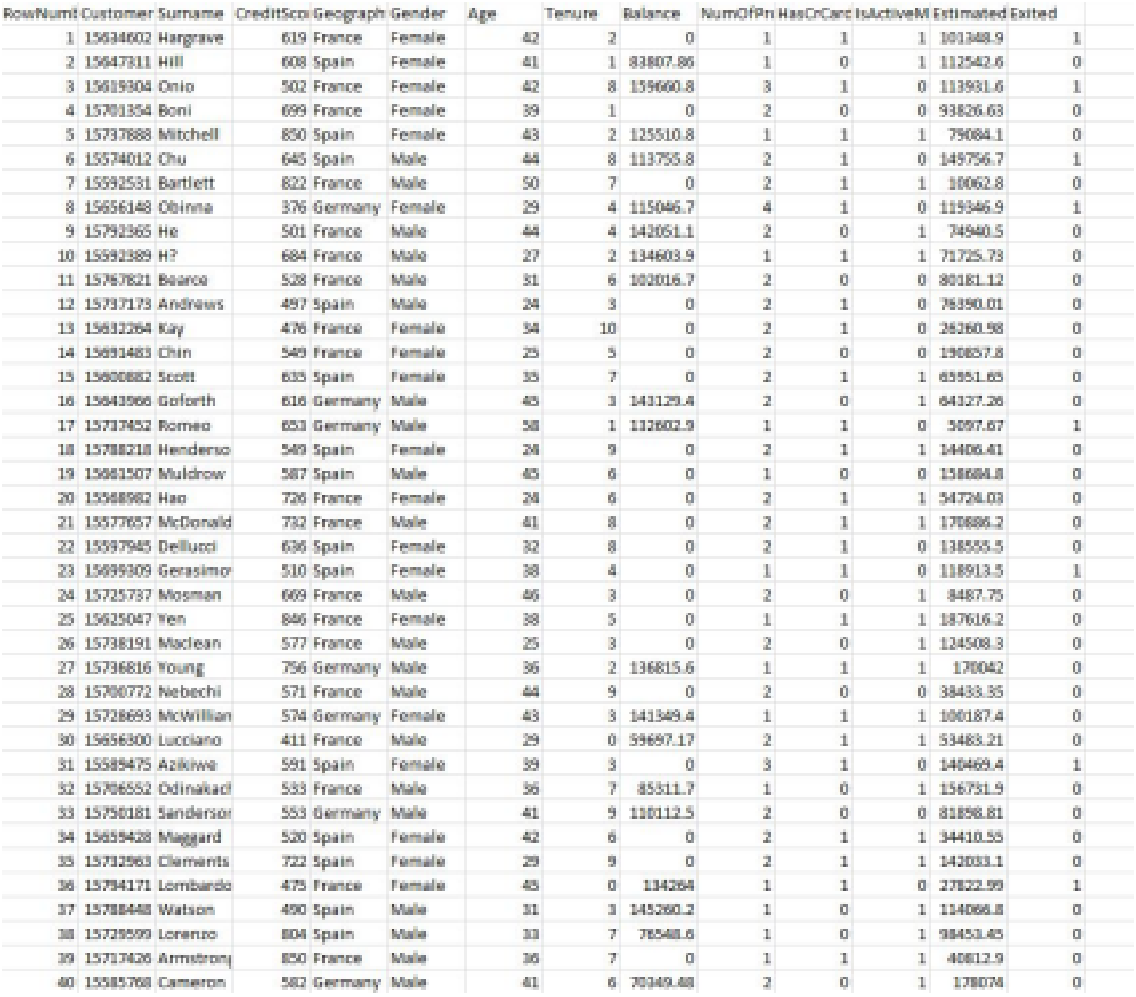
Data Visualization and Pre-processing
Assignment -2

Project Name	AI BASED DISCOURSE FOR BANKING INDUSTRY
Student Name	DARWIN ARUN DOSS I
Student Roll no	720819205008
Maximum Marks	2 Marks

Question-1.Download dataset

Solution:

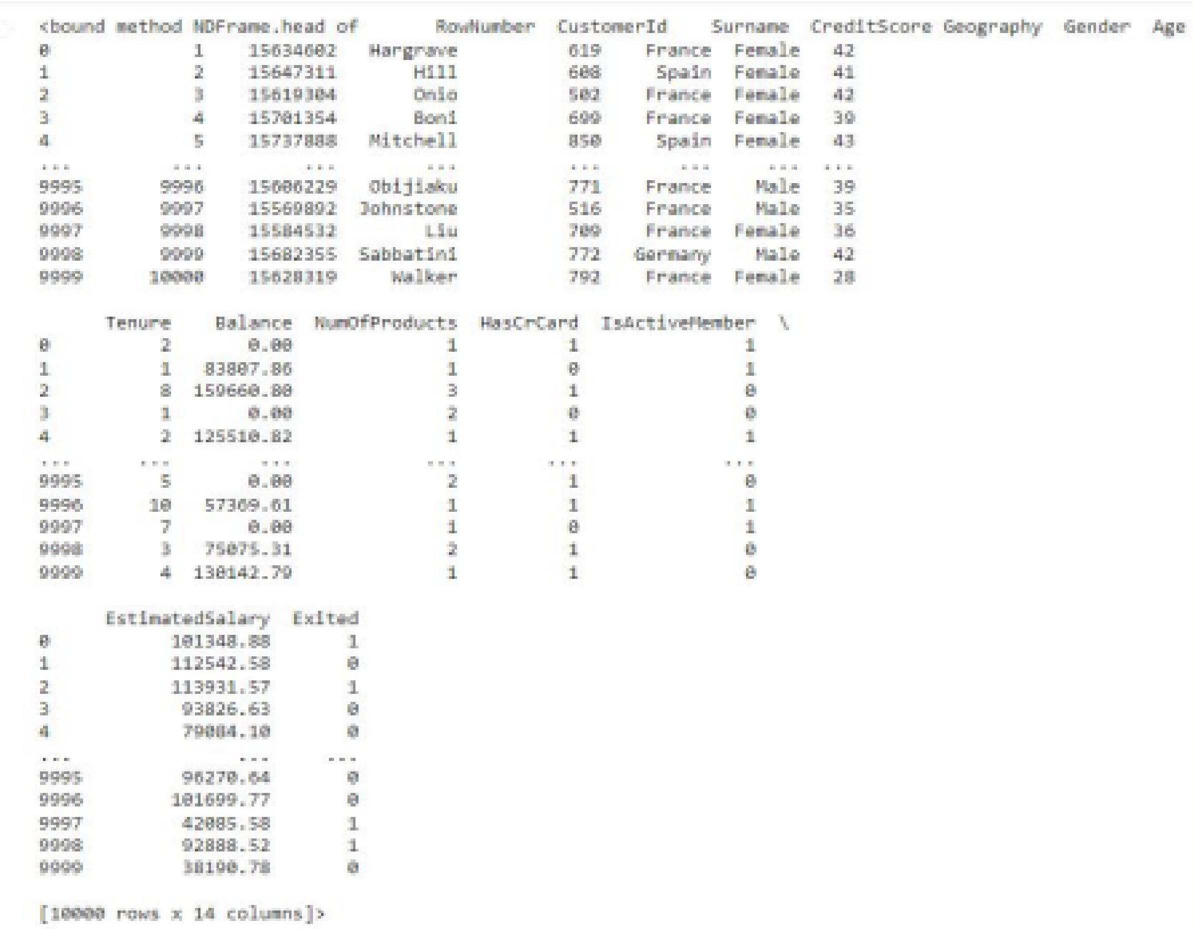


RowNumt	Customer	Surname	CreditSco	Geograph	Gender	Age	Tenure	Balance	NumOfPr	HasCrCarc	IsActiveM	Estimated	Exited
1	15634602	Hargrave	619	France	Female	42	2	0	1	1	1	101348.9	1
2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.6	0
3	15619304	Onio	502	France	Female	42	8	159660.8	3	1	0	113931.6	1
4	15701354	Boni	699	France	Female	39	1	0	2	0	0	93826.63	0
5	15737888	Mitchell	850	Spain	Female	43	2	125510.8	1	1	1	79084.1	0
6	15574012	Chu	645	Spain	Male	44	8	113755.8	2	1	0	149756.7	1
7	15592531	Bartlett	822	France	Male	50	7	0	2	1	1	10062.8	0
8	15656148	Obinna	376	Germany	Female	29	4	115046.7	4	1	0	119346.9	1
9	15792365	He	501	France	Male	44	4	142051.1	2	0	1	74940.5	0
10	15592389	H?	684	France	Male	27	2	134603.9	1	1	1	71725.73	0
11	15767821	Bearce	528	France	Male	31	6	102016.7	2	0	0	80181.12	0
12	15737173	Andrews	497	Spain	Male	24	3	0	2	1	0	76390.01	0
13	15632264	Kay	476	France	Female	34	10	0	2	1	0	26260.98	0
14	15691483	Chin	549	France	Female	25	5	0	2	0	0	190857.8	0
15	15600882	Scott	635	Spain	Female	35	7	0	2	1	1	65951.65	0
16	15643966	Goforth	616	Germany	Male	45	3	143129.4	2	0	1	64327.26	0
17	15737452	Romeo	653	Germany	Male	58	1	132602.9	1	1	0	5097.67	1
18	15788218	Henderso	549	Spain	Female	24	9	0	2	1	1	14406.41	0
19	15661507	Muldrow	587	Spain	Male	45	6	0	1	0	0	158684.8	0
20	15568982	Hao	726	France	Female	24	6	0	2	1	1	54724.03	0
21	15577657	McDonald	732	France	Male	41	8	0	2	1	1	170886.2	0
22	15597945	Dellucci	636	Spain	Female	32	8	0	2	1	0	138555.5	0
23	15699309	Gerasimo	510	Spain	Female	38	4	0	1	1	0	118913.5	1
24	15725737	Mosman	669	France	Male	46	3	0	2	0	1	8487.75	0
25	15625047	Yen	846	France	Female	38	5	0	1	1	1	187616.2	0
26	15738191	Maclean	577	France	Male	25	3	0	2	0	1	124508.3	0
27	15736816	Young	756	Germany	Male	36	2	136815.6	1	1	1	170042	0
28	15700772	Nebechi	571	France	Male	44	9	0	2	0	0	38433.35	0
29	15728693	McWillian	574	Germany	Female	43	3	141349.4	1	1	1	100187.4	0
30	15656300	Lucciano	411	France	Male	29	0	59697.17	2	1	1	53483.21	0
31	15589475	Azikiwe	591	Spain	Female	39	3	0	3	1	0	140469.4	1
32	15706552	Odinakacl	533	France	Male	36	7	85311.7	1	0	1	156731.9	0
33	15750181	Sandersoi	553	Germany	Male	41	9	110112.5	2	0	0	81898.81	0
34	15659428	Maggard	520	Spain	Female	42	6	0	2	1	1	34410.55	0
35	15732963	Clements	722	Spain	Female	29	9	0	2	1	1	142033.1	0
36	15794171	Lombardo	475	France	Female	45	0	134264	1	1	0	27822.99	1
37	15788448	Watson	490	Spain	Male	31	3	145260.2	1	0	1	114066.8	0
38	15729599	Lorenzo	804	Spain	Male	33	7	76548.6	1	0	1	98453.45	0
39	15717426	Armstron	850	France	Male	36	7	0	1	1	1	40812.9	0
40	15585768	Cameron	582	Germany	Male	41	6	70349.48	2	0	1	178074	0

Question-2.Load the dataset

Solution:

```
import numpy as np
import pandas as pd
import seaborn as sns
importmatplotlib.pyplot as plt
import sklearn
data = pd.read_csv(r'Churn_Modelling.csv')
df.head
```



```
<bound method NDFrame.head 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   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]>
```

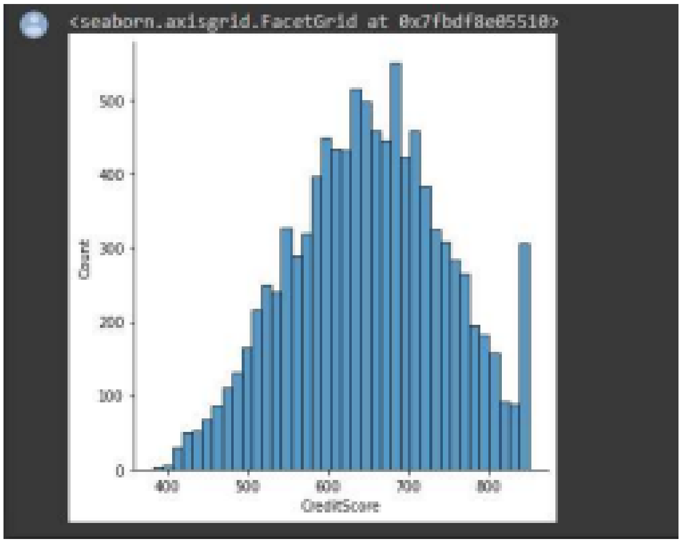
Question-3.Perform Below Visualizations. Perform

Below Visualizations.

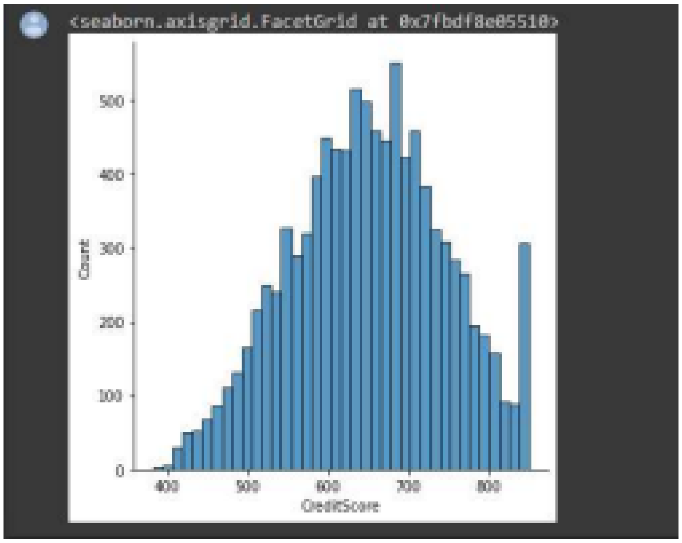
3.1 Univariate Analysis

Solution:

sns.displot(data['CreditScore'])

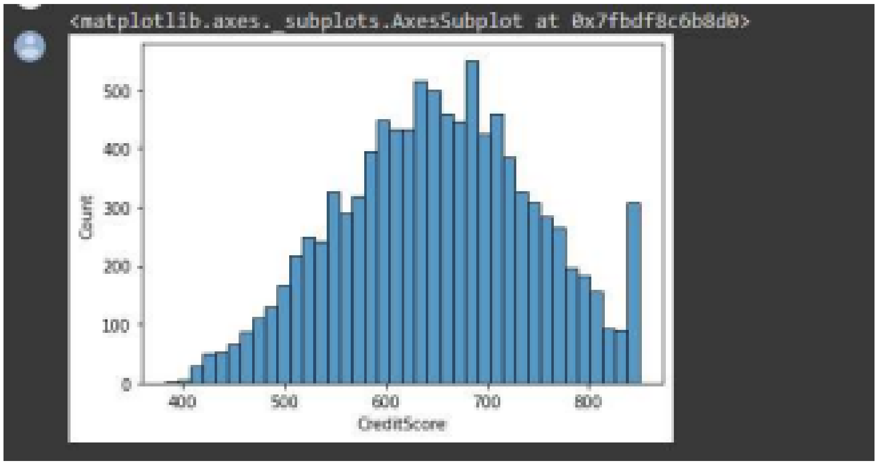


```
<seaborn.axisgrid.FacetGrid at 0x7fbdf8e05510>
```

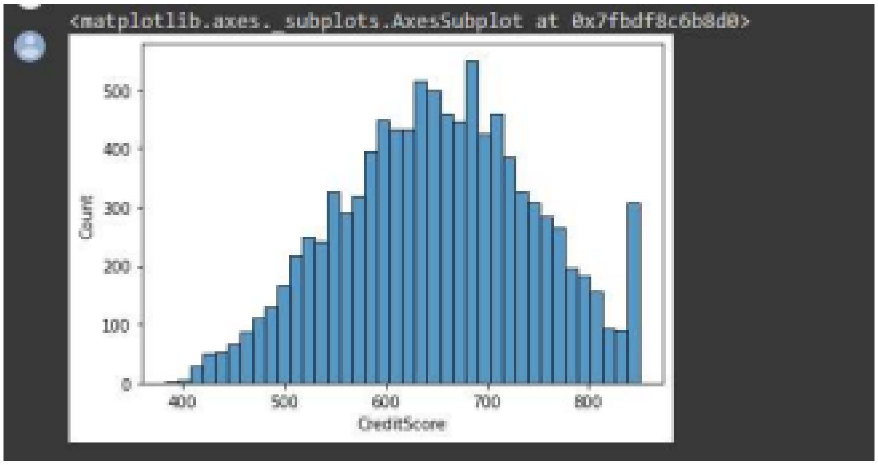


```
<seaborn.axisgrid.FacetGrid at 0x7fbdf8e05510>
```

sns.histplot(data['CreditScore'])

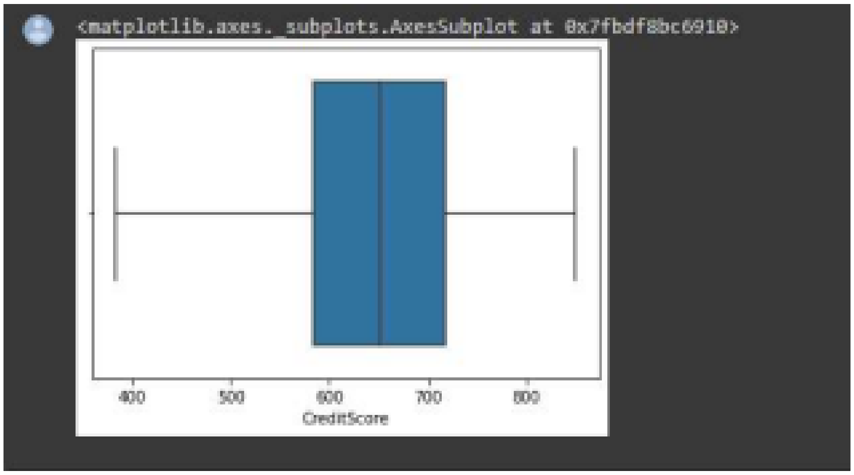


```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbdf8c6b8d0>
```

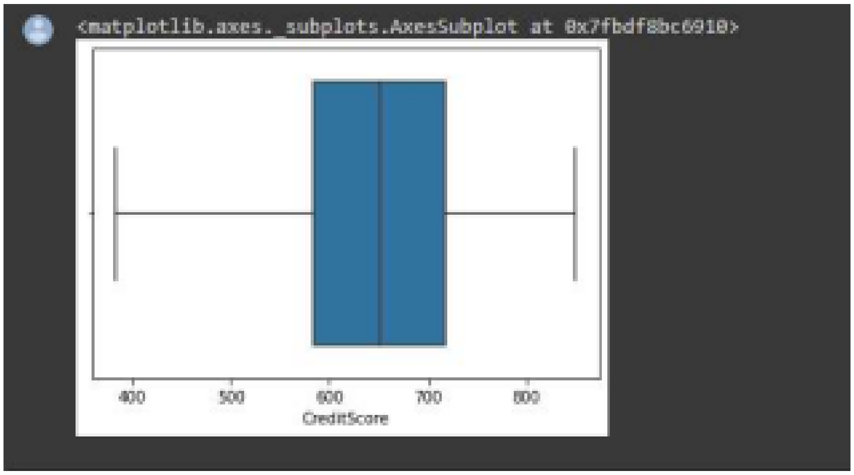


```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbdf8c6b8d0>
```

sns.boxplot(x = data['CreditScore'])

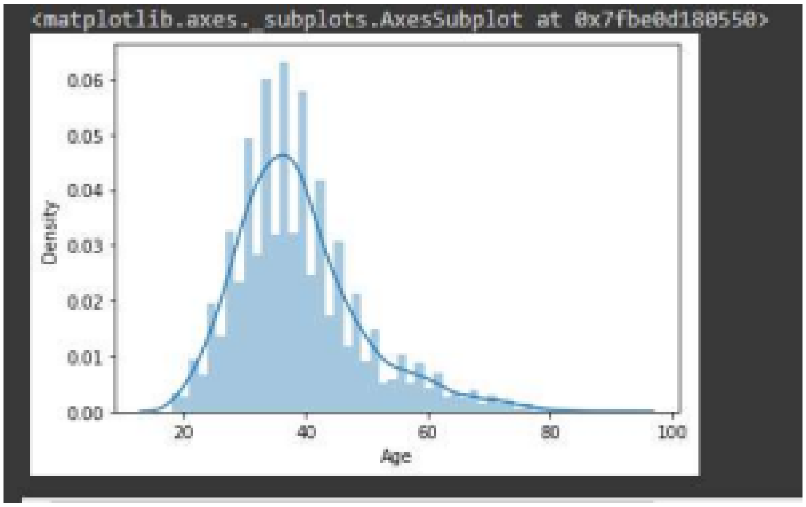


```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbdf8bc6910>
```

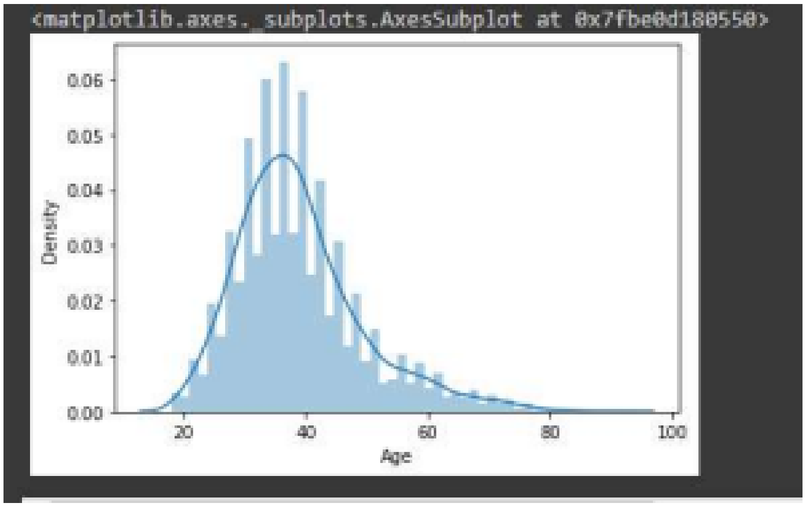


```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbdf8bc6910>
```

sns.distplot(data['Age'])

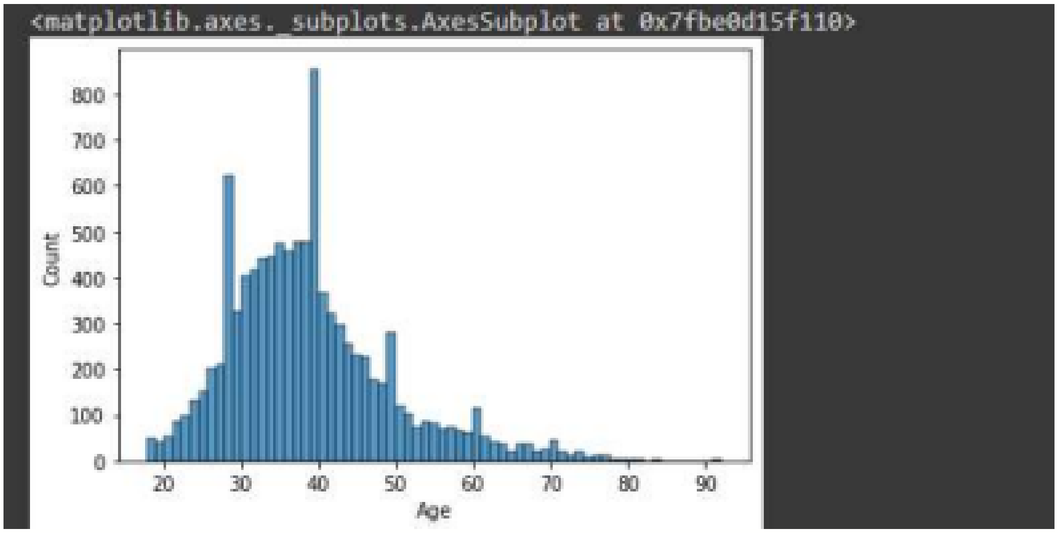


```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbe0d180550>
```

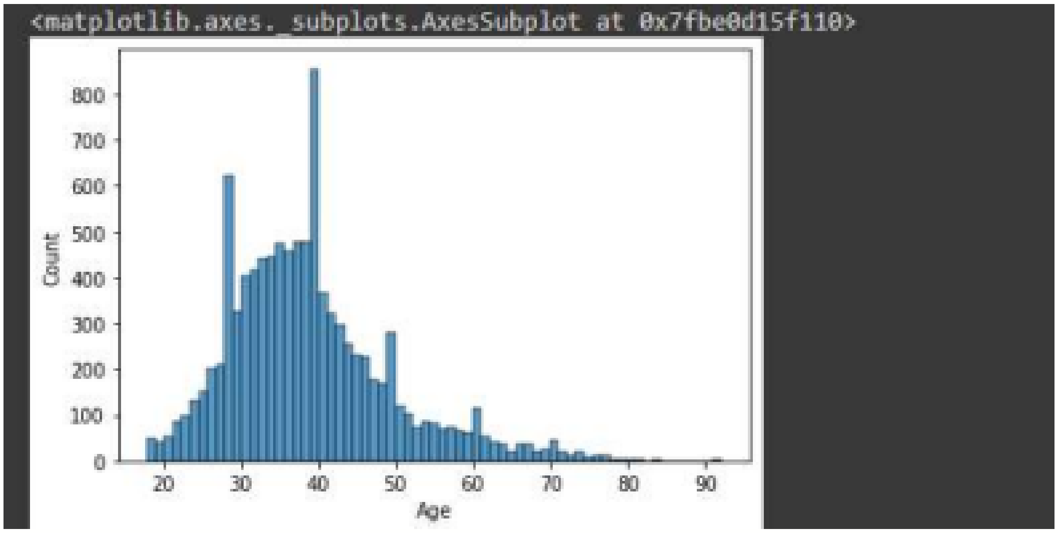


```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbe0d180550>
```

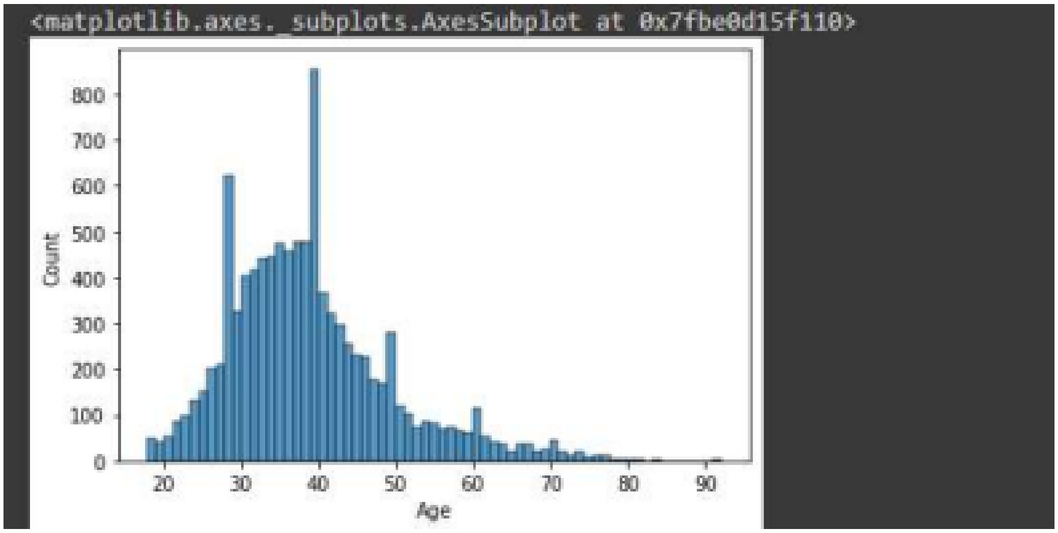
```
sns.histplot(data['Age'])
```



```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbe0d15f110>
```

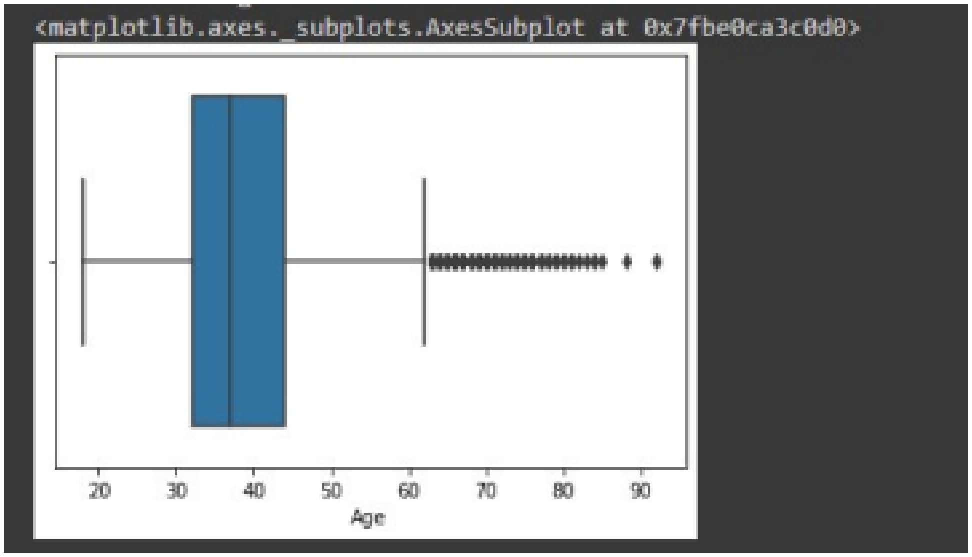


```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbe0d15f110>
```

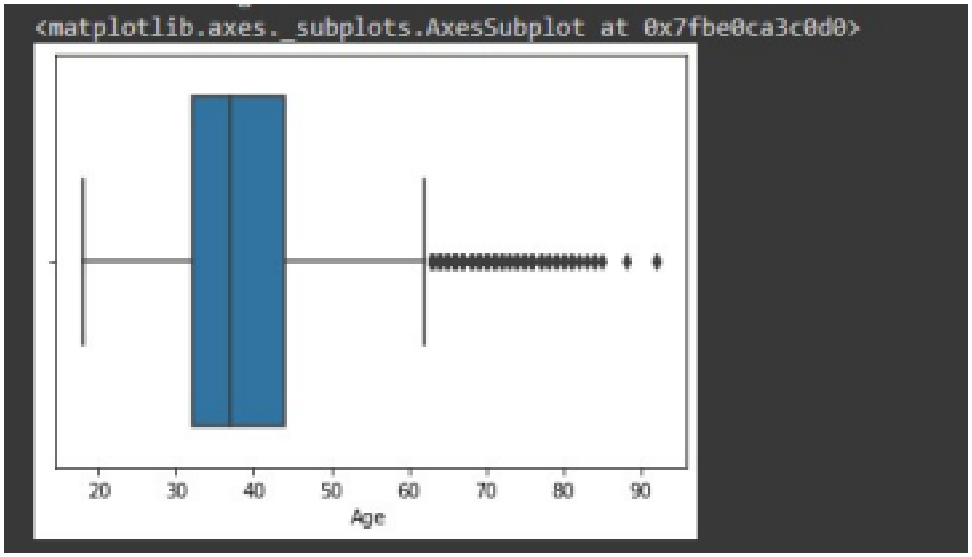


```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbe0d15f110>
```

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



```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbe0ca3c0d0>
```



```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbe0ca3c0d0>
```

3.2 Bivariate Analysis

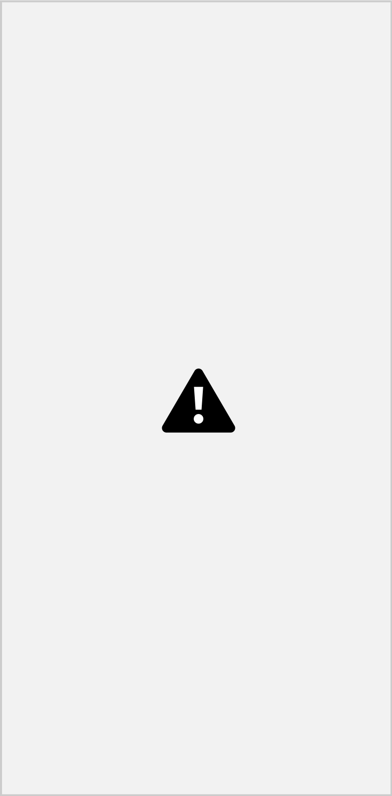
Solution:

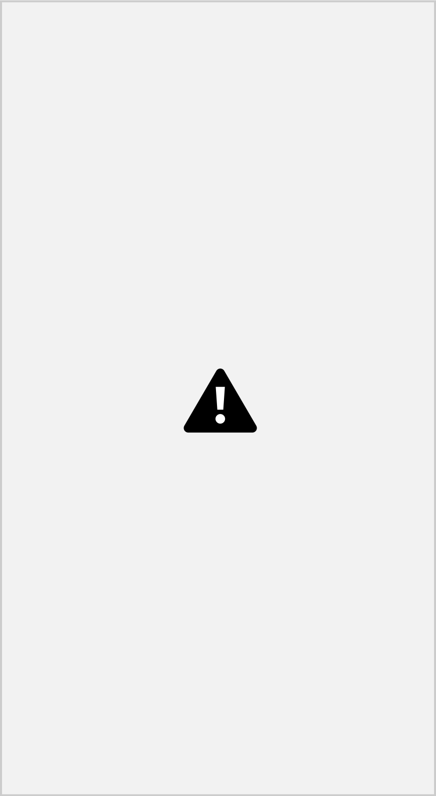
```
plt.figure(figsize=(7,7))
sns.lineplot(data = data, x = 'Tenure', y = 'CreditScore')
```

```
plt.figure(figsize=(10,10))
sns.barplot(data = data, x = 'CreditScore',
                                 'CreditScore', y ='EstimatedSalary')
```

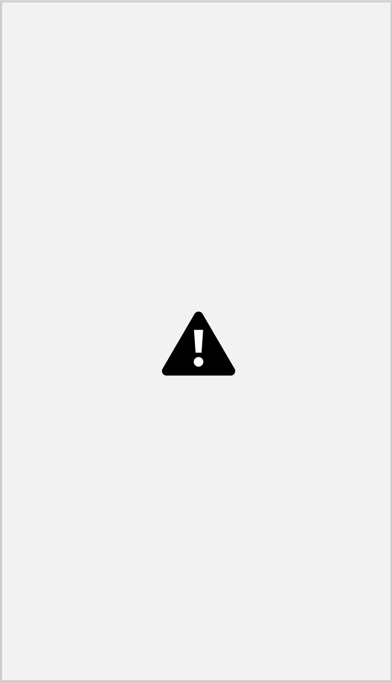
```
plt.figure(figsize=(10,10))
```

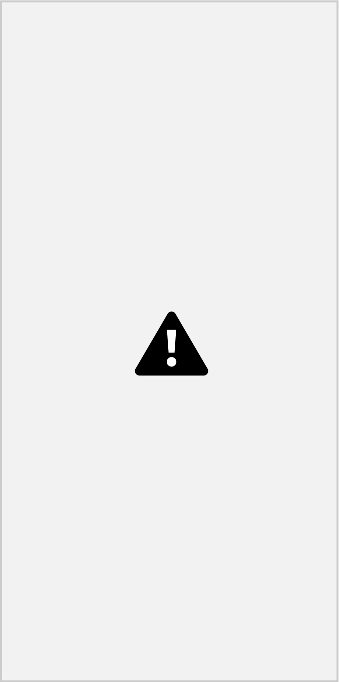
```
sns.barplot(data = data, x = 'CreditScore',
                                            y = 'Tenure')
```





```
plt.figure(figsize=(10,10))
sns.lineplot(data['Age'], data['EstimatedSalary'])
```





```
plt.figure(figsize=(17,17))
sns.barplot(data['Age'], data['EstimatedSalary'])
```

```
data['EstimatedSalary'])
```

```
sns.scatterplot(data = data, x = 'CreditScore
                              'CreditScore', y = 'Age')
```

3.3 Multivariate Analysis

Solution:

```
sns.scatterplot(data = data, x = 'CreditScore', 'CreditScore', y =

                              'Balance', hue = 'Gender')
```

```
sns.scatterplot(data['Tenure'], data['CreditScore'], hue = data['Gender'])
```

sns.scatterplot(data['Tenure'], data['CreditScore'], hue = data['Gender'])

```
sns.scatterplot(data['Age'], data['Balance'],
                                data['Balance'], hue = data['Gender'])
```

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

Question-4.Perform descriptive statistics on the dataset. Perform descriptive statistics on the dataset.

Solution:

```
data.mean(numeric_only = True)
```

```
data.median(numeric_only = True)
```

```
data['CreditScore'].mode()
```

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

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

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

```
data.std(numeric_only=True)
```

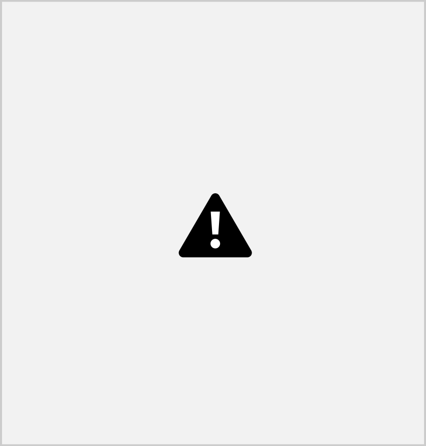
```
data.describe()
```

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

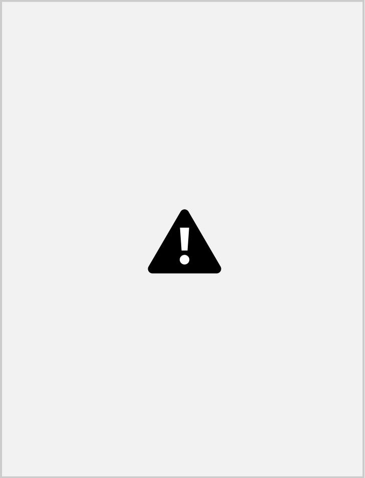
Question-5.Handle the Missing values.

Solution:

```
data.isnull().any()
```



data.isnull().sum()



Question-6.Find the outliers and replace the outliers

Solution:

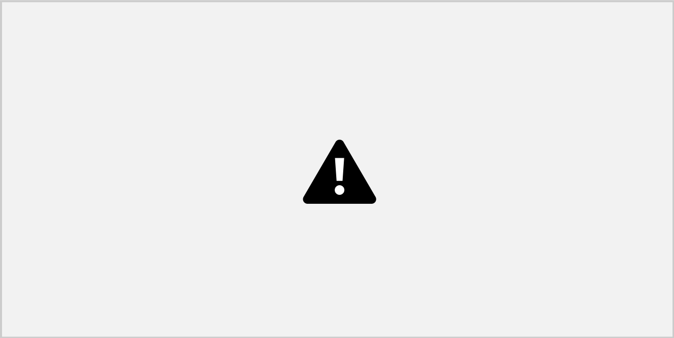
sns.boxplot(data['CreditScore'])#Outlier detection - box plot

```
fig, ax = plt.subplots(figsize = (5,3)) #Outlier detection - Scatter
plot  ax.scatter(data['Balance'], data['Exited'])

# x-axis label
ax.set_xlabel('Balance')

# y-axis label
ax.set_ylabel('Exited')
plt.show()

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

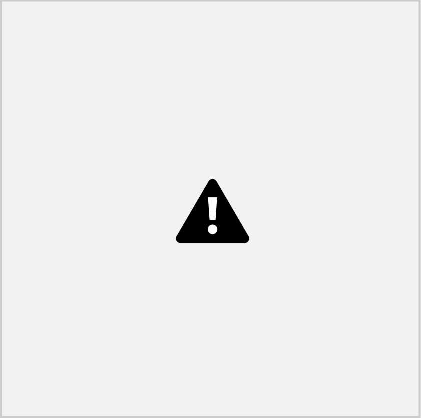


```
from scipy import stats #Outlier detection – zscore
```

```
zscore = np.abs(stats.zscore(data['CreditScore']))
print(zscore)
print('No. of Outliers : ', np.shape(np.where(zscore>3)))
```

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

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



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

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

```
Q1 = data['EstimatedSalary'].quantile(0.25) #Outlier detection - IQR
Q3 = data['EstimatedSalary'].quantile(0.75)
iqr = Q3 - Q1
print(iqr)
upper=Q3 + 1.5 * iqr
lower=Q1 - 1.5 * iqr
count = np.size(np.where(data['EstimatedSalary'] >upper))
count = count + np.size(np.where(data['EstimatedSalary'] <lower))
print('No. of outliers : ', count)
```

```
data['CreditScore'] = np.where(np.logical_or(data['CreditScore']>900,
data['CreditScore']<383), 65 0, data['CreditScore'])
sns.boxplot(data['CreditScore'])
```

```
upper = data.Age.mean() + (3 * data.Age.std()) #Outlier detection - 3
sigma  lower = data.Age.mean() - (3 * data.Age.std())
columns = data[ ( data['Age'] > upper ) | ( data['Age']<lower ) ]
print('Upper range : ', upper)
print('Lower range : ', lower)
print('No. of Outliers : ', len(columns))
```

```
columns = ['EstimatedSalary', 'Age', 'Balance', 'NumOfProducts', 'Tenure', 'CreditScore']
#After outl ier removal

for i in columns:
 Q1 = data[i].quantile(0.25)
 Q3 = data[i].quantile(0.75)
 iqr = Q3 - Q1
 upper=Q3 + 1.5 * iqr
 lower=Q1 - 1.5 * iqr
 count = np.size(np.where(data[i] >upper))
 count = count + np.size(np.where(data[i] <lower))
 print('No. of outliers in ', i, ' : ', count)
```

Question-7. Check for Categorical columns and perform encoding

Solution:

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

Question-8.Split the data into dependent and independent variables split the data in X and Y

Solution:

```
x # independent values ( inputs)
x = data.iloc[:, 0:13]
```

```
y # dependent values (output)
y = data['Exited']
```

Question-9.Scale the independent variables

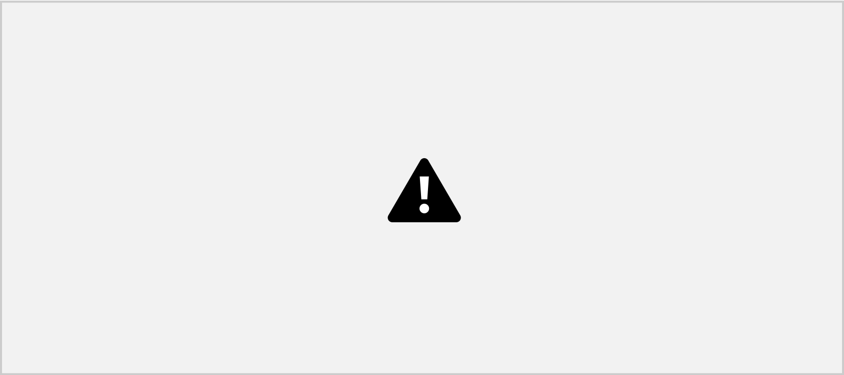
Solution:

```
from sklearn.preprocessing import StandardScaler, MinMaxScaler
sc = StandardScaler()
x_scaled = sc.fit_transform(x)
x_scaled
```

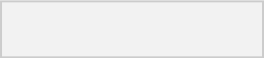
Question-10.Split x and y into Training and Testing

Solution:

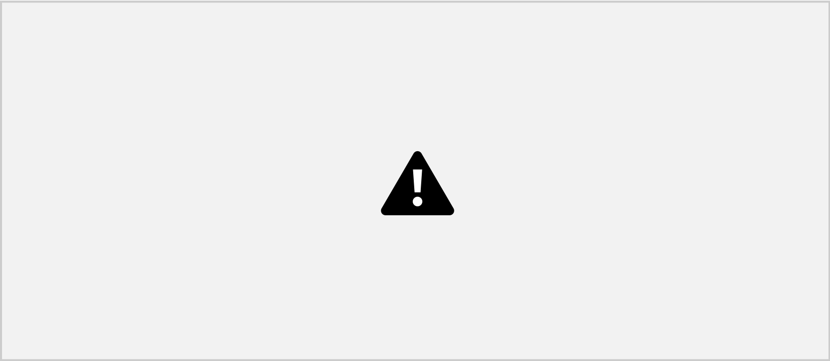
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x_scaled, y, test_size = 0.3, random_state = 0)
x_train
```



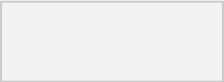
x_train.shape



x_test



x_test.shape



y_train

y_test