ASSIGNMENT -2

```
In [22]: import pandas as pd
         import numpy as np
         import seaborn as sns
         import matplotlib.pyplot as plt
         import sklearn
```

```
In [5]: data=pd.read_csv("C:/Users/MANOHARI/Downloads/Churn_Modelling.csv")
        data.head()
```

Out[5]:

|	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Ba
0	1	15634602	Hargrave	619	France	Female	42	2	
1	2	15647311	Hill	608	Spain	Female	41	1	838
2	3	15619304	Onio	502	France	Female	42	8	1596
3	4	15701354	Boni	699	France	Female	39	1	
4	5	15737888	Mitchell	850	Spain	Female	43	2	1255

UNIVARIATE ANALYSIS

```
In [6]: sns.distplot(data['CreditScore'])
```

```
C:\Users\MANOHARI\anaconda3\lib\site-packages\seaborn\distributions.py:26
19: FutureWarning: `distplot` is a deprecated function and will be remove
d in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-le
vel function for histograms).
  warnings.warn(msg, FutureWarning)
```
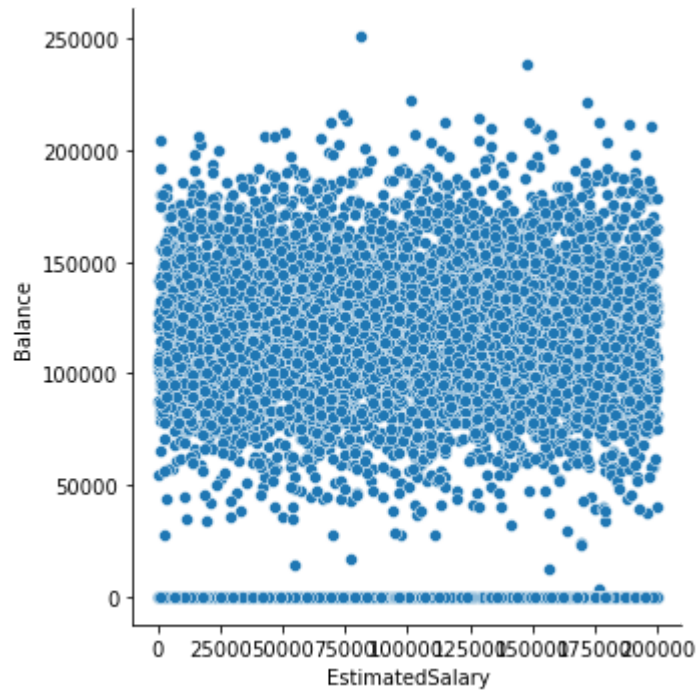
Out[6]: `<AxesSubplot:xlabel='CreditScore', ylabel='Density'>`
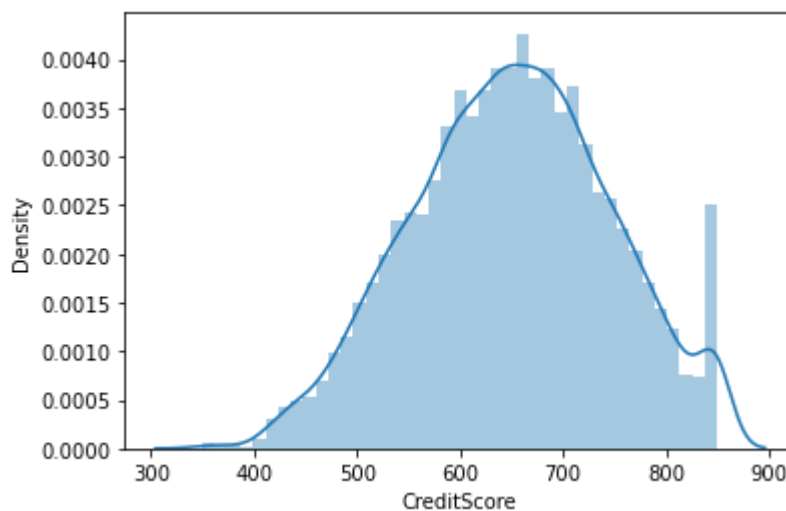
BIVARIATE ANALYSIS

```
In [7]: sns.relplot(x='EstimatedSalary',y="Balance", data=data)
```
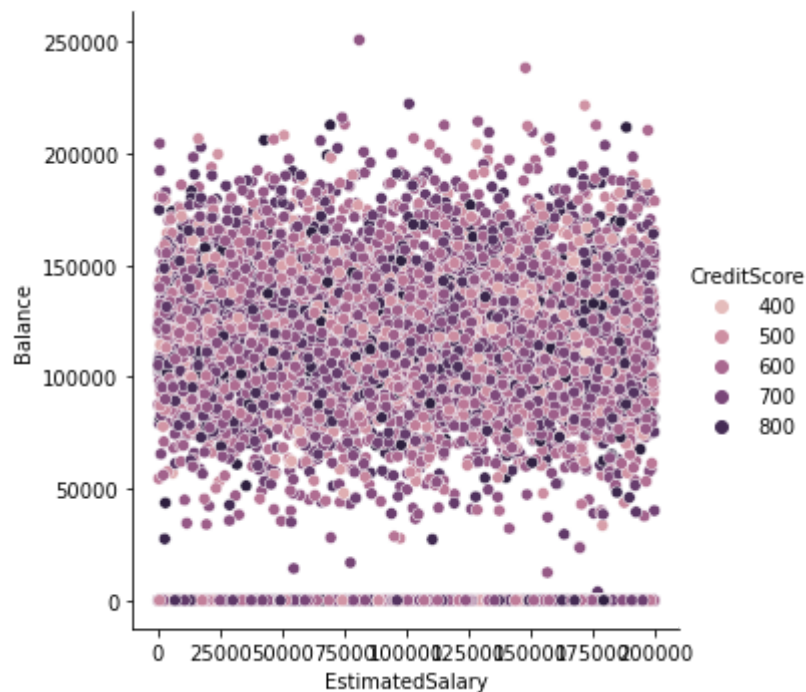
`<seaborn.axisgrid.FacetGrid at 0x20e8a05a1c0>`

MULTIVARIATE ANALYSIS

```python
sns.relplot(x='EstimatedSalary',y="Balance", hue = "CreditScore", data=da
```

`<seaborn.axisgrid.FacetGrid at 0x20e89448b80>`

DESCRIPTIVE STATISTICS

SUM OF EXITED

```
In [9]:  data['Exited'].sum()

Out[9]:  2037
```

Average of Age

```
In [10]:  data['Age'].mean()

Out[10]:  38.9218
```

Standard Deviation of Estimated salary

```
In [11]:  data['EstimatedSalary'].std()

Out[11]:  57510.49281769822
```

Median of Estimated Salary

```
In [12]:  data['EstimatedSalary'].median()

Out[12]:  100193.915
```

Maximum and Minimum of CreditScores

```
In [13]:  print("max=",data['CreditScore'].max(),"min=",data['CreditScore'].min())

          max= 850 min= 350
```

```
In [14]:  data.isnull().sum()
          #No null values
          data['CreditScore'].fillna(data['CreditScore'].mean())
          #filling null values with mean

Out[14]:  0         619
          1         608
          2         502
          3         699
          4         850
                   ...
          9995      771
          9996      516
          9997      709
          9998      772
          9999      792
          Name: CreditScore, Length: 10000, dtype: int64
```

```
In [15]:  sns.boxplot(data['CreditScore'])

          C:\Users\MANOHARI\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
          FutureWarning: Pass the following variable as a keyword arg: x. From vers
          ion 0.12, the only valid positional argument will be `data`, and passing
          other arguments without an explicit keyword will result in an error or mi
          sinterpretation.
            warnings.warn(

Out[15]:  <AxesSubplot:xlabel='CreditScore'>
```
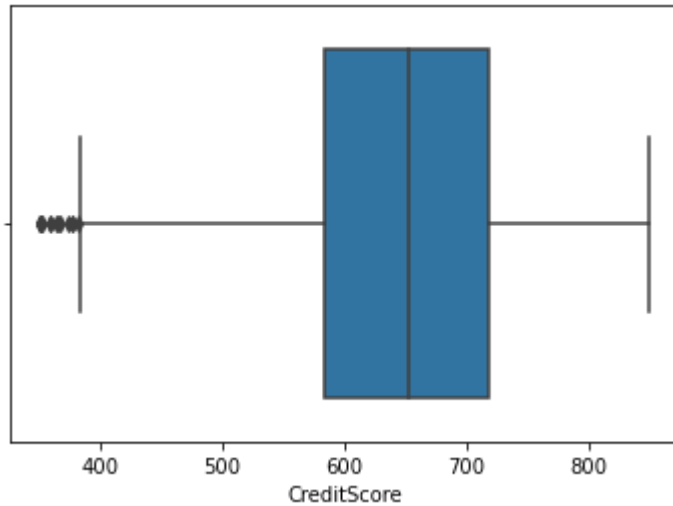
REMOVING AND REPLACING OUTLIERS

In [16]:
```python
percentile25 = data['CreditScore'].quantile(0.25)
percentile75 = data['CreditScore'].quantile(0.75)
iqr=percentile75-percentile25
upper_limit = percentile75 + 1.5 * iqr
lower_limit = percentile25 - 1.5 * iqr

data[data['CreditScore'] > upper_limit]
data[data['CreditScore'] < lower_limit]

new_df = data[(data['CreditScore'] < upper_limit) & (data['CreditScore']>
new_df.shape
```

Out[16]: (9984, 14)

In [17]:
```python
sns.boxplot(new_df['CreditScore'])
```

C:\Users\MANOHARI\anaconda3\lib\site-packages\seaborn_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From vers
ion 0.12, the only valid positional argument will be `data`, and passing
other arguments without an explicit keyword will result in an error or mi
sinterpretation.
  warnings.warn(

Out[17]: <AxesSubplot:xlabel='CreditScore'>

```
In [18]:  new_df_cap = data.copy()
          new_df_cap['CreditScore'] = np.where(
              new_df_cap['CreditScore'] > upper_limit,
              upper_limit,
              np.where(
                  new_df_cap['CreditScore'] < lower_limit,
                  lower_limit,
                  new_df_cap['CreditScore']
              )
          )
          sns.boxplot(new_df_cap['CreditScore'])
```
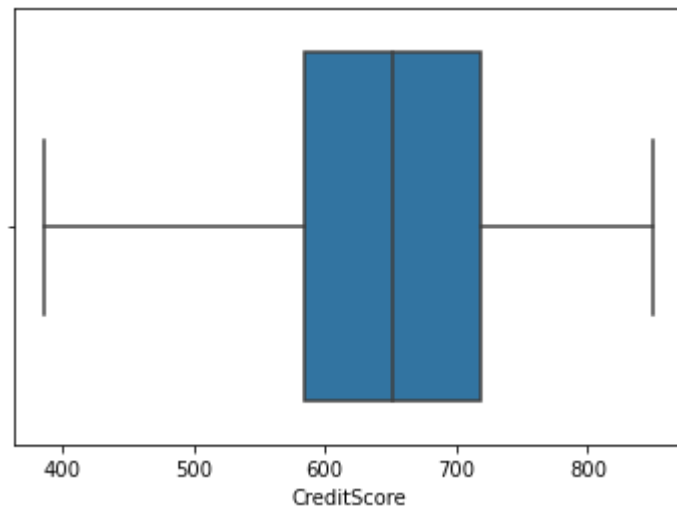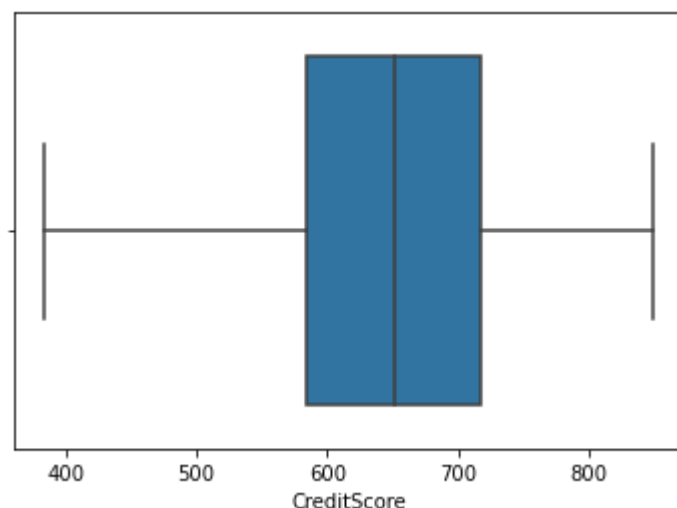
C:\Users\MANOHARI\anaconda3\lib\site-packages\seaborn_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From vers
ion 0.12, the only valid positional argument will be `data`, and passing
other arguments without an explicit keyword will result in an error or mi
sinterpretation.
  warnings.warn(

Out[18]: <AxesSubplot:xlabel='CreditScore'>

ENCODING AND CATEGORICAL COLUMNS

```
In [20]:  data['Gender'].replace({'Male':0, 'Female':1}, inplace=True)
          data
```

Out[20]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure
0	1	15634602	Hargrave	619	France	1	42	2
1	2	15647311	Hill	608	Spain	1	41	1
2	3	15619304	Onio	502	France	1	42	8
3	4	15701354	Boni	699	France	1	39	1
4	5	15737888	Mitchell	850	Spain	1	43	2
...	...	...	...	...	...	...	...	...
9995	9996	15606229	Obijiaku	771	France	0	39	5
9996	9997	15569892	Johnstone	516	France	0	35	10
9997	9998	15584532	Liu	709	France	1	36	7
9998	9999	15682355	Sabbatini	772	Germany	0	42	3
9999	10000	15628319	Walker	792	France	1	28	4

10000 rows × 14 columns

INDEPENDENT AND DEPENDENT VARIABLES

In [21]:
```python
x=data.loc[:, ['CreditScore', 'HasCrCard','IsActiveMember']]
y=data.loc[:,'Exited']
```

TRAIN AND TEST DATA

In [23]:
```python
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.4, r
```

In [24]:
```python
x_train
```

Out[24]:

	CreditScore	HasCrCard	IsActiveMember
7809	579	1	1
5279	652	1	1
3279	652	1	0
8984	645	1	0
8466	613	0	0
...	...	...	...
9225	594	1	1
4859	794	1	1
3264	738	1	0
9845	590	1	1
2732	623	1	0

6000 rows × 3 columns

```
In [25]:  x_test
```

Out[25]:

	CreditScore	HasCrCard	IsActiveMember
9394	597	1	1
898	523	1	0
2398	706	1	1
5906	788	0	0
2343	706	1	1
...	...	...	...
4758	493	1	0
9914	496	1	0
7067	746	0	1
4578	691	0	0
4202	526	0	0

4000 rows × 3 columns

```
In [26]:  y_test
```

```
Out[26]:  9394    0
          898     1
          2398    0
          5906    0
          2343    0
                 ..
          4758    0
          9914    0
          7067    0
          4578    0
          4202    0
          Name: Exited, Length: 4000, dtype: int64
```

```
In [27]:  y_train
```

```
Out[27]:  7809    1
          5279    0
          3279    0
          8984    0
          8466    0
                 ..
          9225    0
          4859    0
          3264    0
          9845    0
          2732    1
          Name: Exited, Length: 6000, dtype: int64
```