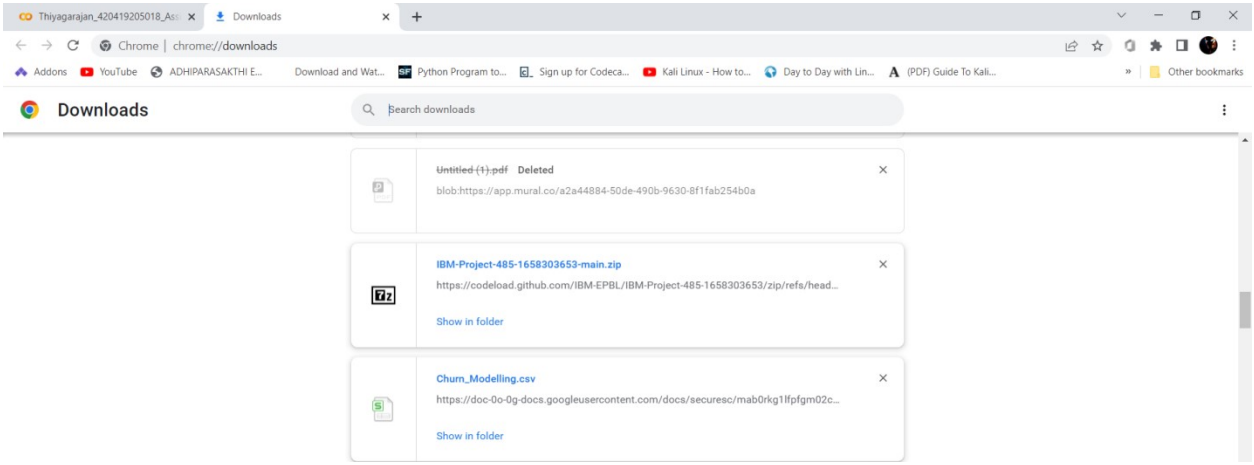
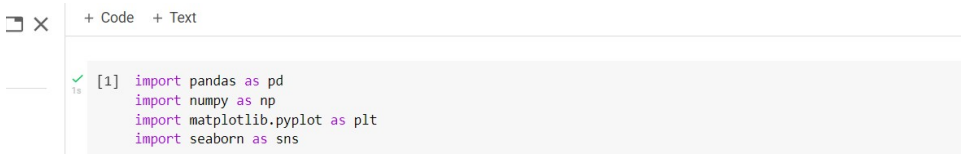
ASSIGNMENT 2

DATE	**26 SEPTEMEBR 2022.**
TEAM ID	**PNT2022TMID38674**
PROJECT NAME	**AI based discourse for Banking Industry**
NAME	**Thiyagarajan V**

1. Download the dataset: Dataset



2. Load the dataset.

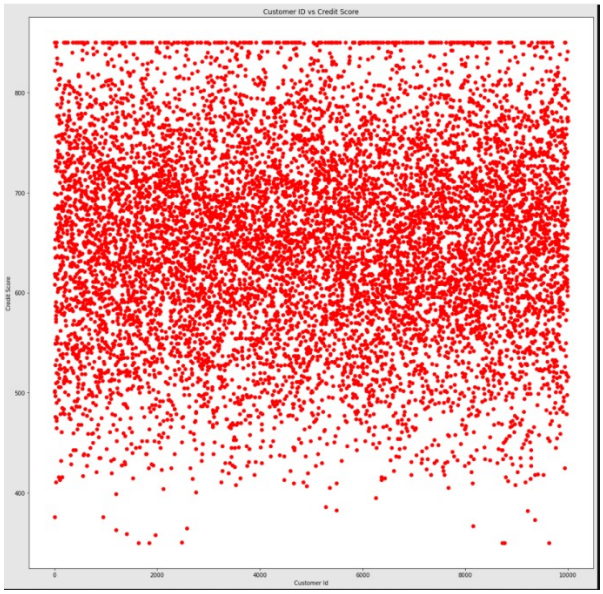


3. Perform Below Visualizations.

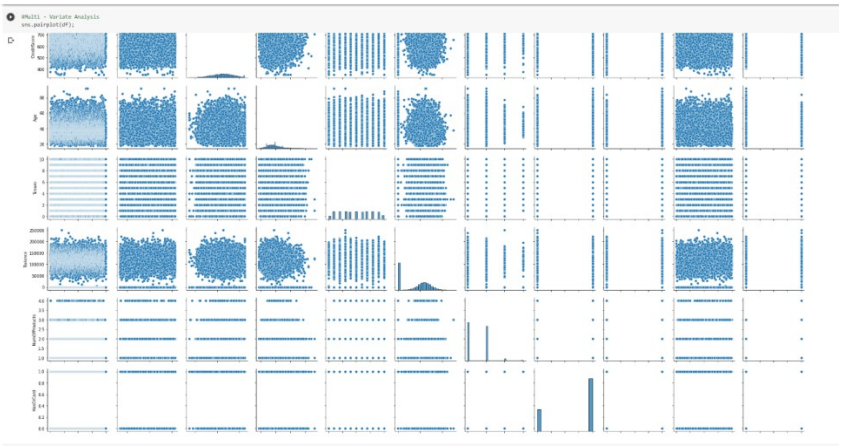
Univariate Analysis

Bi - Variate Analysis

```
#Bi - Variate Analysis
x=df['RowNumber']
y=df['CreditScore']
plt.figure(figsize=(18, 18))
plt.title("Customer ID vs Credit Score")
plt.xlabel("Customer Id")
plt.ylabel("Credit Score")
plt.scatter(x,y,c='red')
```



Multi - Variate Analysis



4. Perform descriptive statistics on the dataset.

```
[ ]  #Multi - Variate Analysis
     sns.pairplot(df);
```

```
#Perform descriptive statistics on the dataset
df.describe()
```

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.00000	10000.000000	10000.000000	10000.000000
mean	5000.50000	1.569094e+07	650.528800	38.921800	5.012800	76485.889288	1.530200	0.70550	0.515100	100090.239881	0.203700
std	2886.89568	7.193619e+04	96.653299	10.487806	2.892174	62397.405202	0.581654	0.45584	0.499797	57510.492818	0.402769

5. Handle the Missing values.

```
[4]
max  10000.00000  1.581569e+07  850.000000  92.000000  10.000000  250898.090000  4.000000  1.00000  1.000000  199992.480000  1.000000
```

```
[5] #Handle the Missing values
    df.isnull().sum()
```

```
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
```

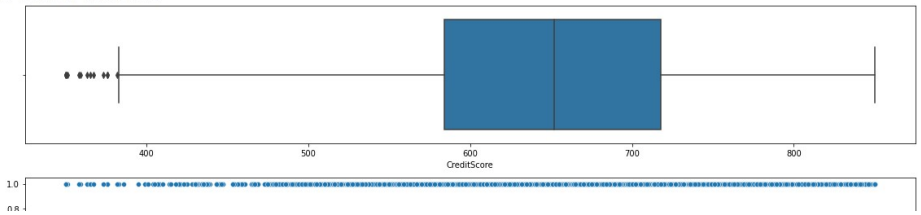
6. Find the outliers and replace the outliers

```
+ Code  + Text

[7] #Find the outliers and replace the outliers
    #Checking for outliers
    def box_scatter(data, a, b):
        fig, (a1, a2) = plt.subplots(nrows=2, ncols=1, figsize=(16,6))
        sns.boxplot(data=data, x=a, ax=a1)
        sns.scatterplot(data=data, x=a,y=b,ax=a2)

box_scatter(df,'CreditScore','Exited');
plt.tight_layout()
print(f"# of Bivariate Outliers: {len(df.loc[df['CreditScore'] < 400])}")
```

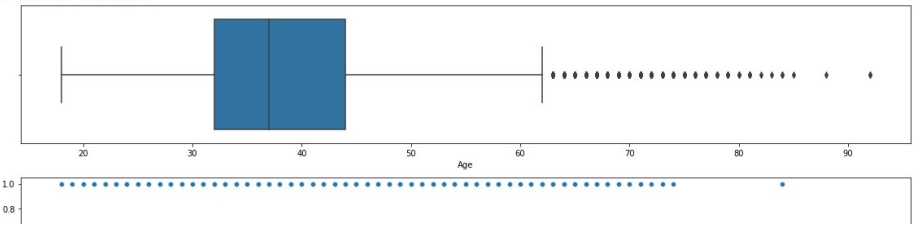
```
# of Bivariate Outliers: 19
```




```
+ Code  + Text

box_scatter(df,'Age','Exited');
plt.tight_layout()
print(f"# of Bivariate Outliers: {len(df.loc[df['Age'] > 87])}")
```

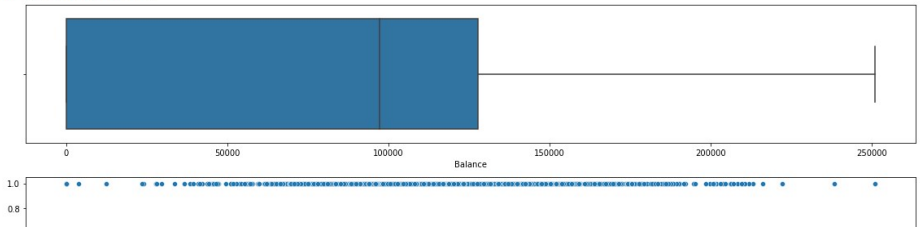
```
# of Bivariate Outliers: 3
```



+ Code + Text

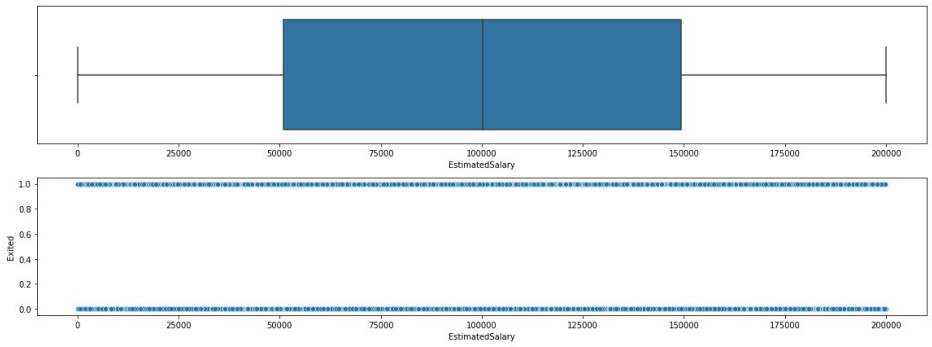
```
box_scatter(df,'Balance','Exited');
plt.tight_layout()
print(f"# of Bivariate Outliers: {len(df.loc[df['Balance'] > 220000])}")
```

```
# of Bivariate Outliers: 4
```



+ Code + Text

```
[11] box_scatter(df,'EstimatedSalary','Exited');
     plt.tight_layout()
```



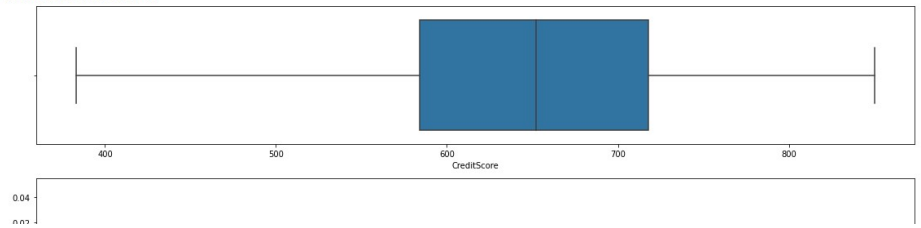

```
#Removing outliers
for i in df:
```

+ Code + Text

```
[12]
```

```
box_scatter(df,'CreditScore','Exited');
plt.tight_layout()
print(f"# of Bivariate Outliers: {len(df.loc[df['CreditScore'] < 400])}")
```

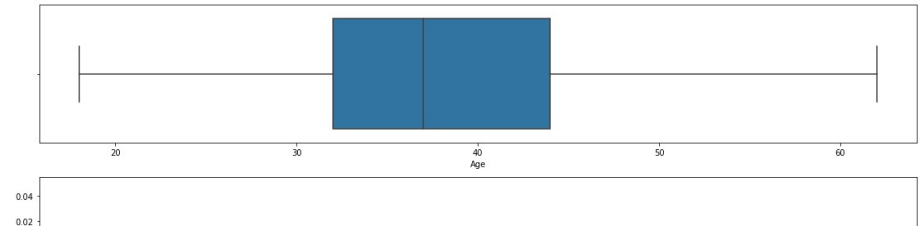
```
# of Bivariate Outliers: 19
```



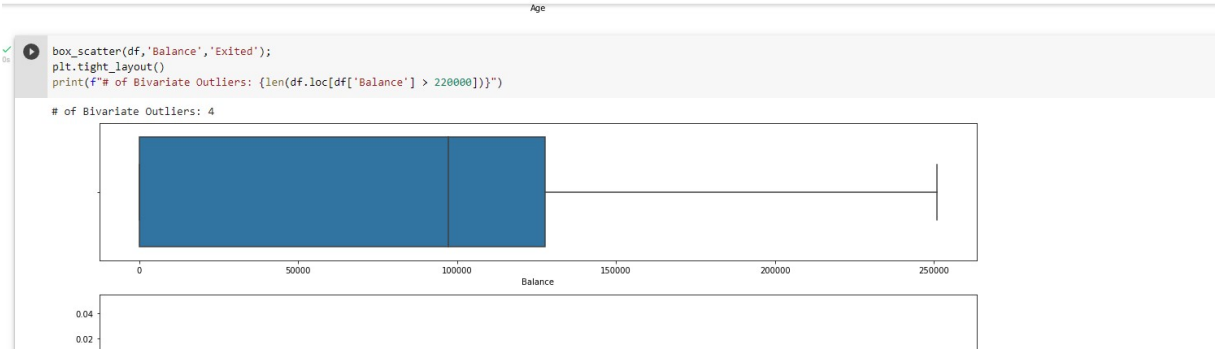
+ Code + Text

```
[ ] box_scatter(df,'Age','Exited');
    plt.tight_layout()
    print(f"# of Bivariate Outliers: {len(df.loc[df['Age'] > 87])}")
```

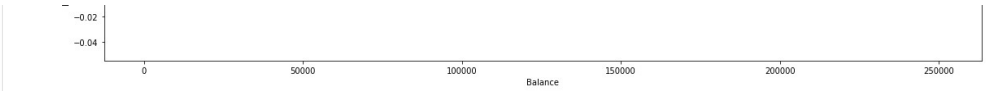
```
# of Bivariate Outliers: 0
```



+ Code + Text



7. Check for Categorical columns and perform encoding.



8. Split the data into dependent and independent variables.

```
[15] #Check for Categorical columns and perform encoding.
     from sklearn.preprocessing import LabelEncoder
     encoder=LabelEncoder()
     for i in df:
```

9. Scale the independent variables

```
[16] #Split the data into dependent and independent variables.
     #Dependent Variable
     x= df.iloc[:, -2].values
```

10. Split the data into training and testing

+ Code + Text

```
[18] #Split the data into training and testing
     from sklearn.model_selection import train_test_split
     x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.5,random_state=0)

[20] x_train

     array([177772.03, 128052.29,  84033.35, ..., 181429.87, 148750.16,
            118855.26])

[21] x_test

     array([192852.67, 128702.1 ,  75732.25, ..., 137041.26,  32996.89,
            125305.34])

[22] y_train

     array([[[3.3400000e+02, 1.5728669e+07, 1.4260000e+03, ..., 1.0000000e+00,
              1.0000000e+00, 0.0000000e+00],
             [6.3920000e+03, 1.5784099e+07, 5.6300000e+02, ..., 1.0000000e+00,
              1.0000000e+00, 0.0000000e+00],
             [1.0000000e+00, 0.0000000e+00],
             [9.8460000e+03, 1.5664035e+07, 2.1220000e+03, ..., 2.0000000e+00,
              1.0000000e+00, 1.0000000e+00],
             [2.7330000e+03, 1.5592816e+07, 2.6780000e+03, ..., 1.0000000e+00,
              1.0000000e+00, 0.0000000e+00]])

[ ] y_test

     array([[[9.3950000e+03, 1.5615753e+07, 2.6910000e+03, ..., 1.0000000e+00,
              1.0000000e+00, 1.0000000e+00],
             [8.9900000e+02, 1.5654700e+07, 8.4600000e+02, ..., 1.0000000e+00,
              1.0000000e+00, 0.0000000e+00],
             [2.3990000e+03, 1.5633877e+07, 1.8570000e+03, ..., 1.0000000e+00,
```