Step 1 : Downloading the dataset of chum_modelling(csv format)

Step 2 : Importing the libraries and loading the dataset

```
import numpy as np
import pandas as pd
```

```
data = pd.read_csv('/content/Churn_Modelling.csv')
```

```
file = pd.DataFrame(data)
file
file.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balan
0	1	15634602	Hargrave	619	France	Female	42	2	0.
1	2	15647311	Hill	608	Spain	Female	41	1	83807.
2	3	15619304	Onio	502	France	Female	42	8	159660.
3	4	15701354	Boni	699	France	Female	39	1	0.
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.

```
file['HasCrCard'] = file['HasCrCard'].astype('category')
```

```
file['IsActiveMember'] = file['IsActiveMember'].astype('category')
file['Exited'] = file['Exited'].astype('category')
```

```
file = file.drop(columns=['RowNumber', 'CustomerId', 'Surname'])
#Removing the RowNumer,CustomerId and Surname column labels from the dataset
```

```
file.head()
#Displaying the data without first three columns
```

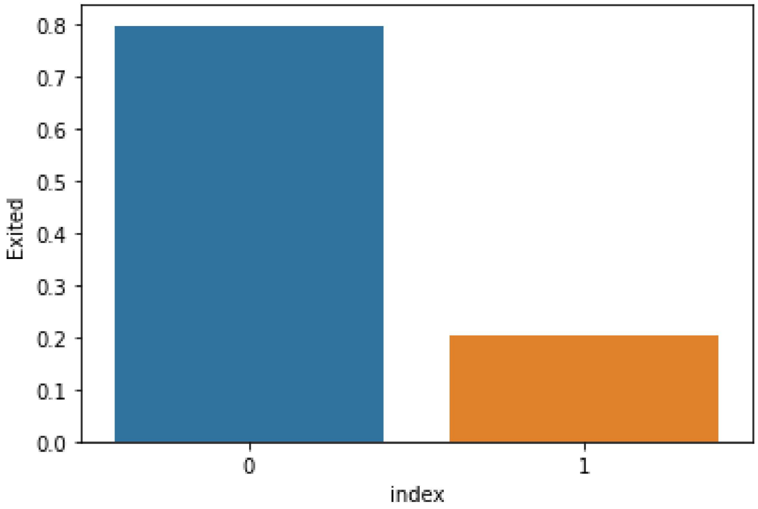
	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsA
0	619	France	Female	42	2	0.00	1	1	

3.Performing the visualizations

- Uni-variate Analysis
- Bi-variate Analysis
- Multi-variate Analysis

```
#Importing seabron library
import seaborn as sns
depth = file['Exited'].value_counts(normalize=True).reset_index()
sns.barplot(data=depth,x='index',y='Exited')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fdc53fb4890>
```



```
depth
```

	index	Exited
0	0	0.7963
1	1	0.2037

From the above relation analysis, it can be said as the data is imbalanced For to correct this, we are processing and visualizing the data using matplotlib libary below

```
import matplotlib.pyplot as plt
```

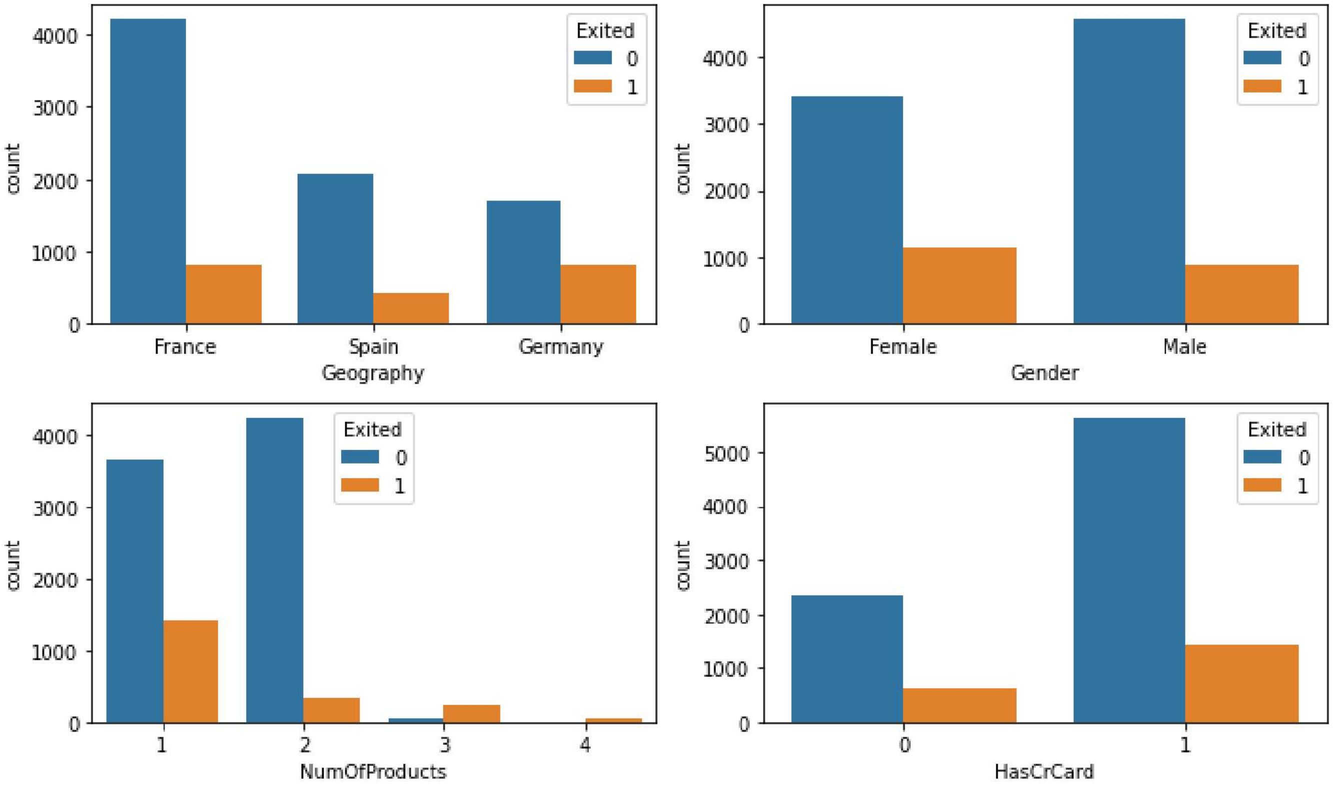
```
categorical = file.drop(columns=['CreditScore', 'Age', 'Tenure', 'Balance', 'EstimatedSalary'
rows = int(np.ceil(categorical.shape[1] / 2)) - 1

fig, axes = plt.subplots(nrows=rows, ncols=2, figsize=(10,6))
axes = axes.flatten()

#Generating subplots
for row in range(rows):
    cols = min(2, categorical.shape[1] - row*2)
    for col in range(cols):
        col_name = categorical.columns[2 * row + col]
        ax = axes[row*2 + col]

        sns.countplot(data=categorical, x=col_name, hue="Exited", ax=ax);

plt.tight_layout()
```



4.Performing Descriptive Statistics method for to explain the features on the dataset

[ ] ↳ *2 cells hidden*

5. Handling and checking for any missing values

[ ] ↳ *3 cells hidden*

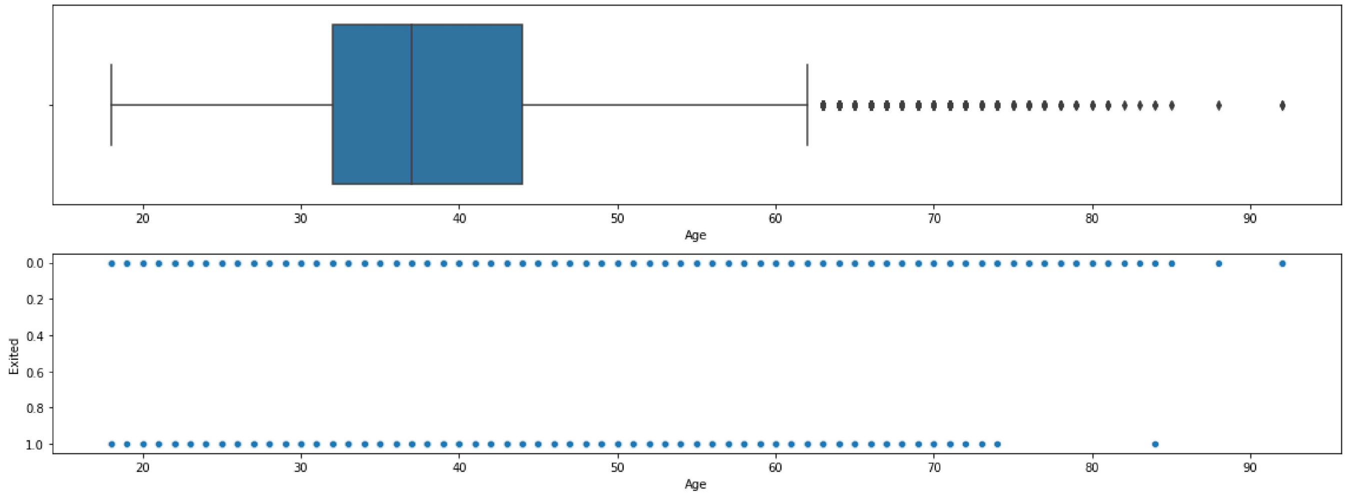
6. Finding the outliers and replacing the outliers

[ ] ↳ *2 cells hidden*

19 outliers in the analysis view

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

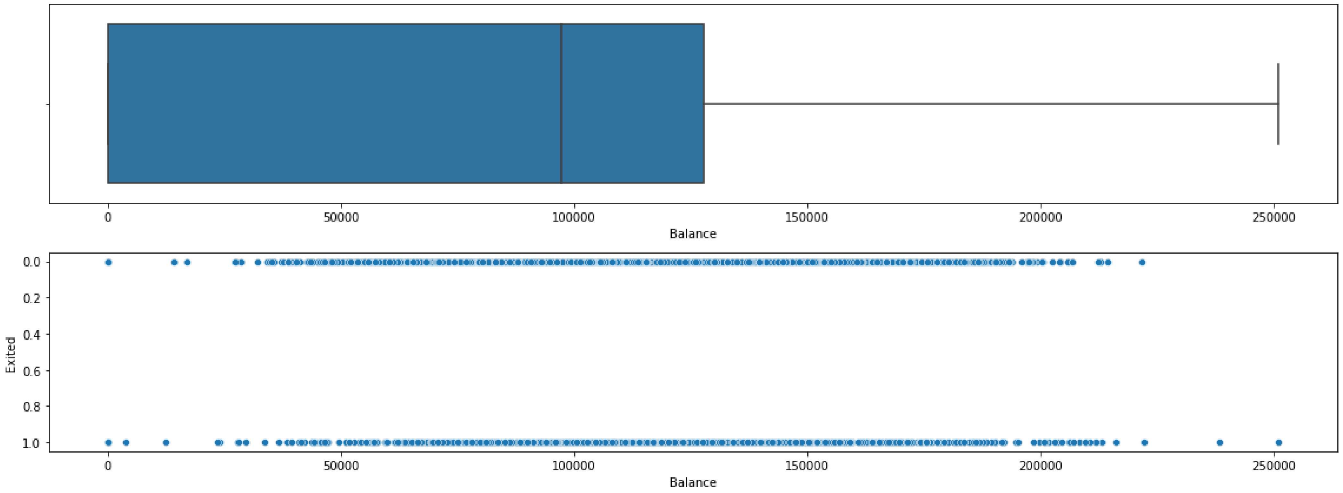
```
# of Bivariate Outliers: 3
```



3 outliers in the above analysis view

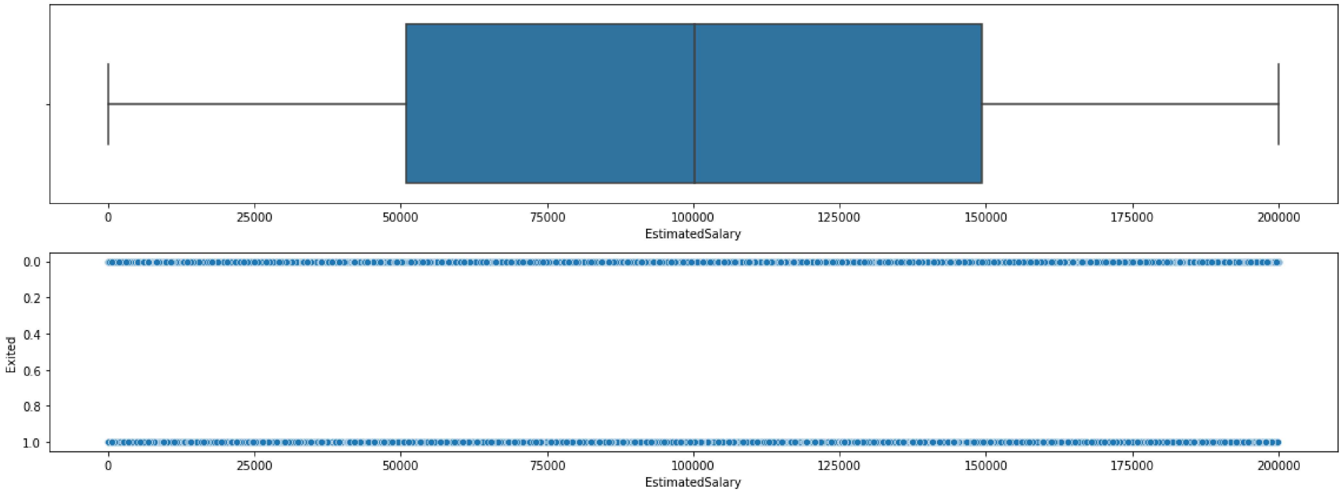
```
box_scatter(file,'Balance','Exited');
plt.tight_layout()
print(f"# of Bivariate Outliers: {len(file.loc[file['Balance'] > 220000])}")
```

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



We can see that there are four outliers above

```
box_scatter(file,'EstimatedSalary','Exited');
plt.tight_layout()
```



Removing the outliers

[ ] ↳ *4 cells hidden*

After removing outliers, boxplot can visualized as below

[ ] ↳ *3 cells hidden*

7. Checking for categorical columns and performing label encoding

Label encoding is performed to convert the labels into numerical(binary digits) values

[ ] ↳ *1 cell hidden*

8. Splitting the data into dependent and independent variables

[ ] ↳ *2 cells hidden*

9. Scaling the independent variables

It can done using StandardScaler from **sciket-learn** framework

[ ] ↳ *1 cell hidden*

10. Splitting the data into training and testing

[ ] ↳ *6 cells hidden*