Assignment 2

Tasks:-

1. Download the dataset: Dataset
2. Load the dataset.
3. Perform Below Visualizations.
    - Univariate Analysis
    - Bi - Variate Analysis
    - Multi - Variate Analysis
4. Perform descriptive statistics on the dataset.
5. Handle the Missing values.
6. Find the outliers and replace the outliers
7. Check for Categorical columns and perform encoding.
8. Split the data into dependent and independent variables.
9. Scale the independent variables
10. Split the data into training and testing

Mounting Drive for dataset

```
from google.colab import drive
drive.mount('/content/drive')
```

Importing libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder,MinMaxScaler
from sklearn.model_selection import train_test_split
```

Downloading and Importing the Dataset

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

data = data.iloc[:,3:]
data
```

Visualizations

1. Univariate Analysis

```
for col in data.columns:
  if(data.dtypes[col]=='int64' or data.dtypes[col]=='float64' ):
    sns.boxplot(x=data[col]).set( xlabel=col)
    plt.show()
```

2. Bi-Variate Analysis

```
sns.FacetGrid(data,hue='Exited',size=5).map(plt.scatter,"Balance","CreditScore").add_legen
plt.show()
```

3.Multivariate

```
sns.pairplot(data, hue='Exited', height=2)
```

Descriptive Analysis

```
data.describe()
```

Handling Missing Values

Since there is no null values this task is skipped

```
data.isnull().sum()
```

Finding and Removing the Outliers

Outliers are found using the univariate BOXPLOT from Task 3

```
CreditsMedian = data.loc[data['CreditScore']<400, 'CreditScore'].median()
ProdMedian = data.loc[data['NumOfProducts']>=3.5,'NumOfProducts'].median()

data.loc[data.CreditScore < 400, 'CreditScore'] = np.nan
data.fillna(CreditsMedian,inplace=True)
```

```
data.loc[data.NumOfProducts > 3, 'NumOfProducts'] = np.nan
data.fillna(ProdMedian,inplace=True)
```

Label Encoding (Categorical)

```
labelencoder = LabelEncoder()
data['Geography']= labelencoder.fit_transform(data['Geography'])
data['Gender'] = labelencoder.fit_transform(data['Gender'])
```

Seperating Dependent and Independent Values

```
independent = data.iloc[:, :-1]
dependent = data.iloc[:,-1:]
```

Scaling the Independent Variables

```
nm =MinMaxScaler()
N_independent = nm.fit_transform(independent)
```

Spliting the Train and Test Data

```
xtrain,xtest,ytrain,ytest=train_test_split(N_independent,dependent,test_size=0.3)
print(xtrain,xtest,ytrain,ytest)
```