Assignment -2 Data Visualization and Pre-processing

Assignment Date	17 October 2022
Student Name	Krishnamoorthy B
Student Roll Number	722819104065
Maximum Marks	2 Marks

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

Result Screenshots:

In [2]:	<pre># Step - 1 : importing the required libraries import pandas import numpy</pre>													
In [3]:	<pre># Step - 2 : Loading the dataset data_frame = pandas.read_csv(r"F:\Naalaiya Thiran\Churn_Modelling.csv")</pre>													
In [4]:	data_fra	ime												
Out[4]:	Ro	wNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	Estimated
	0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101
	1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112
	2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	110
	3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	90
	4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79
	9995	9996	15606229	Obijiaku	771	France	Male	39	5	0.00	2	1	0	96
	9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	10
	9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	42
	9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	92
	9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	38

```
In [5]: # Step - 3 : Visualizations
  In [6]: import seaborn
            from matplotlib import pyplot
  In [7]: # Univariate Analysis
            seaborn.histplot(data = data_frame['Exited'], kde = True)
            pyplot.show()
               8000
                7000
                6000
                5000
             4000
                3000
               2000
               1000
                     0.0
                             0.2
                                               0.6
                                                        0.8
  In [8]: # Bivariate Analysis
            seaborn.scatterplot(data = data_frame, x = 'CreditScore', y = 'Balance')
            pyplot.show()
              250000
              200000
           150000
100000
              50000
                  0
                         400
                                                            800
                                           600
In [11]: # Multivariate analysis
seaborn.scatterplot(data = data_frame, x = 'EstimatedSalary', y = 'Balance', hue = 'Exited')
          pyplot.show()
              250000
              200000
           150000
100000
              50000
```

```
In [12]: # Step - 4 : Handle missing values
           data_frame.isnull().sum()
 Out[12]: RowNumber
           {\tt CustomerId}
                               0
           Surname
                               0
           CreditScore
           Geography
                               0
           Gender
                               0
           Age
           Tenure
           Balance
           NumOfProducts
                               0
           HasCrCard
                               0
           IsActiveMember
           EstimatedSalary
           Exited
           dtype: int64
 In [13]: # Step - 5 : checking for categorical columns and performing encoding
           data_frame.head(1)
 Out[13]:
              RowNumber Customerid Surname CreditScore Geography Gender Age Tenure Balance NumOfProducts HasCrCard IsActiveMember EstimatedSalary
                       1 15634602 Hargrave
                                                                            42
                                                                                          0.0
                                                                                                                                           101348.88
                                                            France Female
 In [14]: input_data = data_frame.iloc[:, 3:13]
           result_data = data_frame.iloc[:, 13:14]
In [15]: input_data
Out[15]:
                CreditScore Geography Gender Age Tenure
                                                          Balance NumOfProducts HasCrCard IsActiveMember EstimatedSalary
             0
                       619
                                              42
                                                      2
                                                             0.00
                                                                                                               101348.88
                               France Female
                                                                                         0
                                                                                                               112542.58
             1
                       608
                                Spain Female
                                              41
                                                       1 83807.86
                                                                              1
                                                                                                       1
                                                                                                               113931.57
                       502
                                                                              3
                                                                                                       0
                               France Female
                                              42
                                                      8 159660.80
                                                                              2
                                                                                                       0
                                                                                         0
             3
                       699
                               France Female
                                              39
                                                             0.00
                                                                                                                93826.63
                       850
                                Spain Female
                                              43
                                                      2 125510.82
                                                                                                                79084.10
                                                                                                       0
           9995
                       771
                               France
                                        Male
                                              39
                                                             0.00
                                                                                                                96270.64
           9996
                       516
                               France
                                       Male
                                              35
                                                      10 57369.61
                                                                              1
                                                                                         1
                                                                                                       1
                                                                                                               101699.77
           9997
                       709
                                              36
                                                             0.00
                                                                                                                42085.58
                               France Female
           9998
                       772
                                       Male
                                              42
                                                      3 75075.31
                                                                              2
                                                                                                       0
                                                                                                                92888.52
                             Germany
           9999
                                                      4 130142.79
                                                                                                                38190.78
                       792
                               France Female
          10000 rows × 10 columns
```

In [16]: result_data

```
Out[16]:
              Exited
                  0
                  0
          9995
                  0
          9996
                  0
          9997
          9998
          9999
                  0
         10000 rows × 1 columns
In [17]: # converting dataframe into arrays
         input_data = input_data.values
         result_data = result_data.values
In [18]: input_data
[709, 'France', 'Female', ..., 0, 1, 42085.58],
                [772, 'Germany', 'Male', ..., 1, 0, 92888.52],
[792, 'France', 'Female', ..., 1, 0, 38190.78]], dtype=object)
In [19]: result data
Out[19]: array([[1],
                [0],
                [1],
                ...,
                [1],
                [1],
                [0]], dtype=int64)
In [20]: # finding out unique categorical values
         data_frame['Gender'].unique()
Out[20]: array(['Female', 'Male'], dtype=object)
In [21]: data_frame['Geography'].unique()
Out[21]: array(['France', 'Spain', 'Germany'], dtype=object)
In [25]: #Step-6: Applying encoding
```

```
In [22]: from sklearn.preprocessing import OneHotEncoder
         from sklearn.compose import ColumnTransformer
In [23]: ct = ColumnTransformer([("oh",OneHotEncoder(),[1,2])],remainder = "passthrough")
         input_data = ct.fit_transform(input_data)
In [26]: input_data[0:10, 1:2]
Out[26]: array([[0.0],
                 [0.0],
                 [0.0],
                 [0.0],
                 [0.0],
                 [0.0],
                 [0.0],
                 [1.0],
                 [0.0],
                 [0.0]], dtype=object)
In [27]: input_data.shape
Out[27]: (10000, 13)
In [36]: result_data.shape
Out[36]: (10000, 1)
In [37]: #Step-7: Splitting data into train and test data
 In [29]: from sklearn.model_selection import train_test_split
           input_data_train,input_data_test,result_data_train,result_data_test = train_test_split(input_data,result_data,test_size = 0.3,rar
           4
 In [30]: input_data_train
 Out[30]: array([[1.0, 0.0, 0.0, ..., 1, 1, 55796.83],
                  [1.0, 0.0, 0.0, ..., 1, 0, 19823.02],
                  [1.0, 0.0, 0.0, \ldots, 0, 1, 13848.58],
                  [1.0, 0.0, 0.0, ..., 1, 0, 181429.87],
                  [0.0, 0.0, 1.0, ..., 1, 1, 148750.16],
                  [0.0, 1.0, 0.0, ..., 1, 0, 118855.26]], dtype=object)
 In [34]: input_data_train.shape
 Out[34]: (7000, 13)
 In [32]: input data test
 Out[32]: array([[0.0, 1.0, 0.0, ..., 1, 1, 192852.67],
                  [1.0, 0.0, 0.0, ..., 1, 0, 128702.1],
                  [0.0, 0.0, 1.0, ..., 1, 1, 75732.25],
                  [1.0, 0.0, 0.0, ..., 1, 1, 167400.29],
                  [1.0, 0.0, 0.0, ..., 1, 1, 70849.47],
                  [0.0, 1.0, 0.0, ..., 1, 1, 33759.41]], dtype=object)
 In [33]: input_data_test.shape
```