ASSIGNMENT – 02

DATA VISUALIZATION AND PRE-PROCESSING

Assignment Date	19-09-2022
Student Name	VALLIAMMAI S
Student Roll Number	311519106105
Maximum Marks	2

1. UNIVARIATE ANALYSIS

+ Code + Text

```
[1] import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    import math
    raw_data=pd.read_csv("Churn_Modelling.csv")
    print(raw_data)
    sns.countplot(x="Geography",data=raw_data)
```

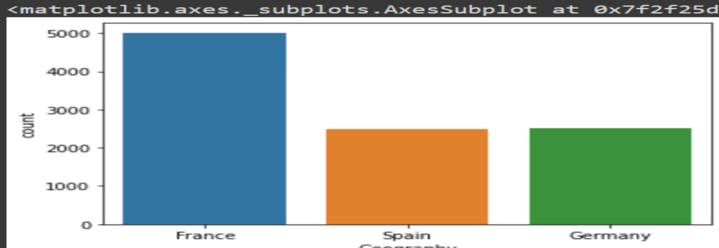
```
      RowNumber  CustomerId    Surname  CreditScore Geography  Gender  Age  \
0             1    15634602   Hargrave          619    France  Female   42
1             2    15647311       Hill          608     Spain  Female   41
2             3    15619304       Onio          502    France  Female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

+ Code + Text

```
[1]       Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
     0         2       0.00              1          1               1
     1         1   83807.86              1          0               1
     2         8  159660.80              3          1               0
     3         1       0.00              2          0               0
     4         2  125510.82              1          1               1
     ...     ...        ...            ...        ...             ...
     9995      5       0.00              2          1               0
     9996     10   57369.61              1          1               1
     9997      7       0.00              1          0               1
     9998      3   75075.31              2          1               0
     9999      4  130142.79              1          1               0

           EstimatedSalary  Exited
     0           101348.88       1
     1           112542.58       0
     2           113931.57       1
     3            93826.63       0
     4            79084.10       0
     ...               ...     ...
     9995         96270.64       0
     9996        101699.77       0
     9997         42085.58       1
     9998         92888.52       1
     9999         38190.78       0

     [10000 rows x 14 columns]
```

+ Code + Text

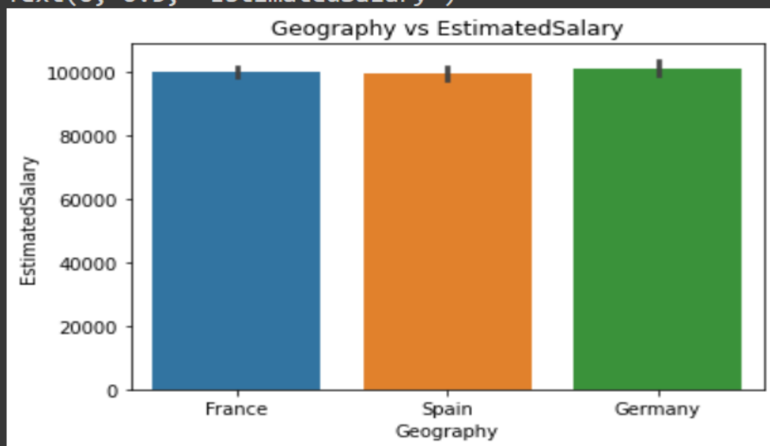
```
[1]  9997         42085.58       1
     9998         92888.52       1
     9999         38190.78       0

     [10000 rows x 14 columns]
     <matplotlib.axes._subplots.AxesSubplot at 0x7f2f25dbb810>
```



2. BIVARIATE ANALYSIS

```
[2] sns.barplot(x="Geography",y="EstimatedSalary",data=raw_data)
    plt.title('Geography vs EstimatedSalary')
    plt.xlabel('Geography')
    plt.ylabel('EstimatedSalary')
```

```
Text(0, 0.5, 'EstimatedSalary')
```

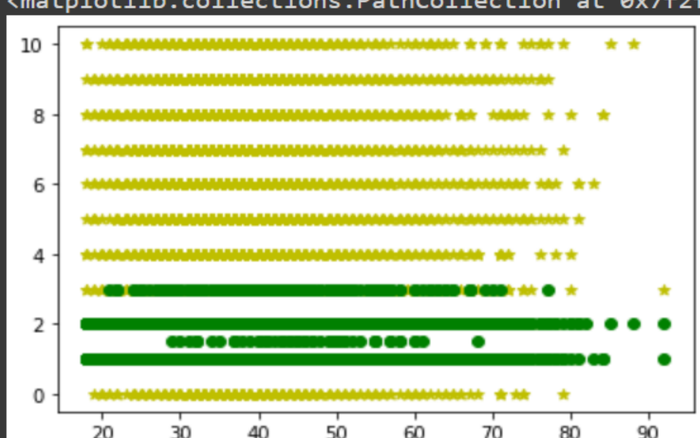


+ Code + Text

3. MULTIVARIATE ANALYSIS

```
[15] import matplotlib.pyplot as plt
     x1= raw_data.Age
     y1 = raw_data.Tenure
     x2 = raw_data.Age
     y2=raw_data.NumOfProducts
     plt.scatter(x1,y1,color='y',marker="*")
     plt.scatter(x2,y2,color='g')
```

```
<matplotlib.collections.PathCollection at 0x7f2f24cc8250>
```



4.DESCRIPTIVE STATISTICS

[5]
```
print(raw_data['CreditScore'].describe())
```

```
count    10000.000000
mean       650.528800
std         96.653299
min        350.000000
25%        584.000000
50%        652.000000
75%        718.000000
max        850.000000
Name: CreditScore, dtype: float64
```

5.MISSING VALUES

[6]
```
raw_data.isnull()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary
0	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9995	False	False	False	False	False	False	False	False	False	False	False	False	False
9996	False	False	False	False	False	False	False	False	False	False	False	False	False
9997	False	False	False	False	False	False	False	False	False	False	False	False	False
9998	False	False	False	False	False	False	False	False	False	False	False	False	False
9999	False	False	False	False	False	False	False	False	False	False	False	False	False

EstimatedSalary	Exited
False	False
False	False
False	False
False	False
False	False
...	...
False	False
False	False
False	False
False	False
False	False

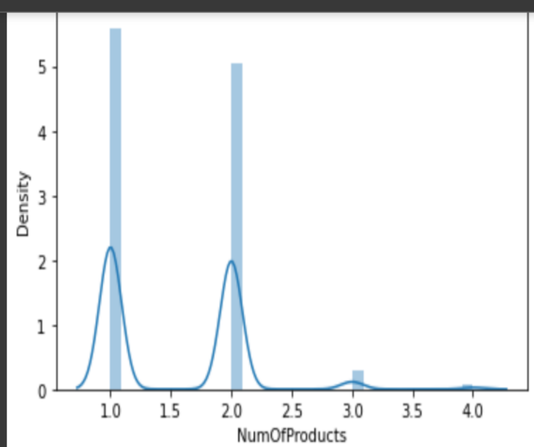
6.OUTLIERS

```
[7] sns.distplot(raw_data["NumOfProducts"])
    ul=raw_data["NumOfProducts"].mean()+3*raw_data["NumOfProducts"].std()
    ll=raw_data["NumOfProducts"].mean()-3*raw_data["NumOfProducts"].std()
    raw_data.loc[(raw_data["NumOfProducts"]>ul) | (raw_data["NumOfProducts"]<ll)]
```

6279	6280	15608338	Chiemenam	757	Spain	Female	55	9	117294.12	4	1	0
6750	6751	15690546	Riley	618	France	Female	42	2	0.00	4	0	0
6875	6876	15665283	Brookes	610	France	Female	57	7	72092.95	4	0	1
7257	7258	15648681	Voronoff	747	France	Female	47	5	139914.60	4	0	1
7457	7458	15668889	Galgano	665	Germany	Female	43	2	116322.27	4	1	0
7567	7568	15750545	Chidiebere	629	France	Male	44	5	0.00	4	0	0
7698	7699	15691513	Dawkins	592	France	Male	60	9	0.00	4	1	1
7724	7725	15673591	Oluchukwu	842	France	Male	44	3	141252.18	4	0	1
7729	7730	15681007	Yen	850	France	Female	35	2	128548.49	4	1	0
8041	8042	15701439	Fanucci	698	Spain	Female	50	1	0.00	4	1	0

+ Code + Text

[7]



Replacing Outliers

```
[8] import statistics
    mean=statistics.mean(raw_data.NumOfProducts)
    raw_data['NumOfProducts']=np.where(raw_data['NumOfProducts']==4,mean,raw_data['NumOfProducts'])
    print(raw_data.NumOfProducts[9])
```

```
1.0
```

7.CATEGORICAL COLUMNS AND ENCODING

```
[9] print(pd.Categorical(raw_data.Gender))
```

```
['Female', 'Female', 'Female', 'Female', 'Female', ..., 'Male', 'Male', 'Female', 'Male', 'Female']
Length: 10000
Categories (2, object): ['Female', 'Male']
```

8.Encoding

```
[10] from sklearn.preprocessing import LabelEncoder,OneHotEncoder
     x=raw_data.iloc[:,:].values
     labelencoder_x=LabelEncoder()
     x[:,5]=labelencoder_x.fit_transform(x[:,5])
     y=pd.DataFrame(x)
     print(y)
```

```
         0         1          2    3        4  5   6   7          8    9  10  \
0        1  15634602   Hargrave  619   France  0  42   2        0.0  1.0   1
1        2  15647311       Hill  608    Spain  0  41   1   83807.86  1.0   0
2        3  15619304       Onio  502   France  0  42   8   159660.8  3.0   1
3        4  15701354       Boni  699   France  0  39   1        0.0  2.0   0
4        5  15737888   Mitchell  850    Spain  0  43   2  125510.82  1.0   1
...    ...       ...        ...  ...      ... ..  ..  ..        ...  ...  ..
9995  9996  15606229   Obijiaku  771   France  1  39   5        0.0  2.0   1
9996  9997  15569892  Johnstone  516   France  1  35  10   57369.61  1.0   1
9997  9998  15584532        Liu  709   France  0  36   7        0.0  1.0   0
9998  9999  15682355  Sabbatini  772  Germany  1  42   3   75075.31  2.0   1
9999 10000  15628319     Walker  792   France  0  28   4  130142.79  1.0   1

     11         12 13
0     1  101348.88  1
1     1  112542.58  0
```

```
[10]
3        4  15701354       Boni  699   France  0  39   1        0.0  2.0   0
4        5  15737888   Mitchell  850    Spain  0  43   2  125510.82  1.0   1
...    ...       ...        ...  ...      ... ..  ..  ..        ...  ...  ..
9995  9996  15606229   Obijiaku  771   France  1  39   5        0.0  2.0   1
9996  9997  15569892  Johnstone  516   France  1  35  10   57369.61  1.0   1
9997  9998  15584532        Liu  709   France  0  36   7        0.0  1.0   0
9998  9999  15682355  Sabbatini  772  Germany  1  42   3   75075.31  2.0   1
9999 10000  15628319     Walker  792   France  0  28   4  130142.79  1.0   1

     11         12 13
0     1  101348.88  1
1     1  112542.58  0
2     0  113931.57  1
3     0   93826.63  0
4     1    79084.1  0
...  ..        ... ..
9995  0   96270.64  0
9996  1  101699.77  0
9997  1   42085.58  1
9998  0   92888.52  1
9999  0   38190.78  0

[10000 rows x 14 columns]
```

9.Split the data into dependent and independent variables

```
[11] print("Dependent variables")
     x= raw_data.iloc[ : ,[4,5,10,11,13]].values
     print(x)
     print("Independent variables")
     y= raw_data.iloc[ : ,[1,2,3,6,7,8,9,12]].values
     print(y)
```

```
Dependent variables
[['France' 'Female' 1 1 1]
 ['Spain' 'Female' 0 1 0]
 ['France' 'Female' 1 0 1]
 ...
 ['France' 'Female' 0 1 1]
 ['Germany' 'Male' 1 0 1]
 ['France' 'Female' 1 0 0]]
Independent variables
[[15634602 'Hargrave' 619 ... 0.0 1.0 101348.88]
 [15647311 'Hill' 608 ... 83807.86 1.0 112542.58]
 [15619304 'Onio' 502 ... 159660.8 3.0 113931.57]
 ...
 [15584532 'Liu' 709 ... 0.0 1.0 42085.58]
 [15682355 'Sabbatini' 772 ... 75075.31 2.0 92888.52]
 [15628319 'Walker' 792 ... 130142.79 1.0 38190.78]]
```

10.Scale the independent variables

```
[12] from sklearn.preprocessing import MinMaxScaler
     min_max_scaler = MinMaxScaler()
     print("Scaled Independent Variable CreditScore")
     raw_data[["CreditScore"]] = min_max_scaler.fit_transform(raw_data[["CreditScore"]])
     print(raw_data)
```

```
Scaled Independent Variable CreditScore
      RowNumber  CustomerId    Surname  CreditScore Geography  Gender  Age  \
0             1    15634602   Hargrave        0.538    France  Female   42
1             2    15647311       Hill        0.516     Spain  Female   41
2             3    15619304       Onio        0.304    France  Female   42
3             4    15701354       Boni        0.698    France  Female   39
4             5    15737888   Mitchell        1.000     Spain  Female   43
...         ...         ...        ...          ...       ...     ...  ...
9995       9996    15606229   Obijiaku        0.842    France    Male   39
9996       9997    15569892  Johnstone        0.332    France    Male   35
9997       9998    15584532        Liu        0.718    France  Female   36
9998       9999    15682355  Sabbatini        0.844   Germany    Male   42
9999      10000    15628319     Walker        0.884    France  Female   28

      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0          2       0.00            1.0          1               1
1          1   83807.86            1.0          0               1
```

```
[12]        Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
     0           2       0.00            1.0          1               1
     1           1   83807.86            1.0          0               1
     2           8  159660.80            3.0          1               0
     3           1       0.00            2.0          0               0
     4           2  125510.82            1.0          1               1
     ...       ...        ...            ...        ...             ...
     9995        5       0.00            2.0          1               0
     9996       10   57369.61            1.0          1               1
     9997        7       0.00            1.0          0               1
     9998        3   75075.31            2.0          1               0
     9999        4  130142.79            1.0          1               0

           EstimatedSalary  Exited
     0           101348.88       1
     1           112542.58       0
     2           113931.57       1
     3            93826.63       0
     4            79084.10       0
     ...               ...     ...
     9995         96270.64       0
     9996        101699.77       0
     9997         42085.58       1
     9998         92888.52       1
     9999         38190.78       0

     [10000 rows x 14 columns]
```

11.Split the data into training and testing

```
[13] import pandas as pd
     from sklearn.linear_model import LinearRegression
     from sklearn.model_selection import train_test_split
     X = raw_data.iloc[:, :-1]
     y = raw_data.iloc[:, -1]
     X_train, X_test, y_train, y_test = train_test_split(
         X, y, test_size=0.05, random_state=0)
     print(X_train)
     print(X_test)
     print(y_train)
     print(y_test)
```

```
     ...        ...       ...        ...         ...      ...     ...  ...
     8938      8939  15722409    Ritchie       0.686    Spain    Male   47
     9291      9292  15679804   Esquivel       0.572   France    Male   36
     491        492  15699005     Martin       0.720   France  Female   41
     2021      2022  15795519   Vasiliev       0.732  Germany  Female   18
     4299      4300  15711991  Chiawuotu       0.530   France    Male   30

           Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
     9394       8  131101.04            1.0          1               1
     898        2  102967.41            1.0          1               0
     2398       8   95386.82            1.0          1               1
```

```
[13]
      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
9394       8  131101.04            1.0          1               1
898        2  102967.41            1.0          1               0
2398       8   95386.82            1.0          1               1
5906       4  112079.58            1.0          0               0
2343       5  163034.82            2.0          1               1
...      ...        ...            ...        ...             ...
8938       8  107604.66            1.0          1               1
9291       5  117559.05            2.0          1               1
491        2  156067.05            1.0          1               1
2021       3  128743.80            1.0          0               0
4299       8       0.00            2.0          0               0

      EstimatedSalary
9394        192852.67
898         128702.10
2398         75732.25
5906         89368.59
2343        135662.17
...               ...
8938         80149.27
9291        111573.30
491           9983.88
2021        197322.13
4299          3183.15
```

```
[13]
[500 rows x 13 columns]
799     0
1069    1
8410    0
9436    0
5099    1
       ..
9225    0
4859    0
3264    0
9845    0
2732    1
Name: Exited, Length: 9500, dtype: int64
9394    0
898     1
2398    0
5906    0
2343    0
       ..
8938    0
9291    0
491     0
2021    0
4299    0
Name: Exited, Length: 500, dtype: int64
```