Team Id : PNT2022TMID42545

ASSIGNMENT_2 :- (Revathi D)

Loading The Data Set:

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

```
df = pd.read_csv("/content/Churn_Modelling.csv")
df
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	B
0	1	15634602	Hargrave	619	France	Female	42	2	
1	2	15647311	Hill	608	Spain	Female	41	1	83
2	3	15619304	Onio	502	France	Female	42	8	159
3	4	15701354	Boni	699	France	Female	39	1	
4	5	15737888	Mitchell	850	Spain	Female	43	2	125
...	...	...	...	...	...	...	...	...	
9995	9996	15606229	Obijiaku	771	France	Male	39	5	
9996	9997	15569892	Johnstone	516	France	Male	35	10	57
9997	9998	15584532	Liu	709	France	Female	36	7	
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75
9999	10000	15628319	Walker	792	France	Female	28	4	130

10000 rows × 14 columns

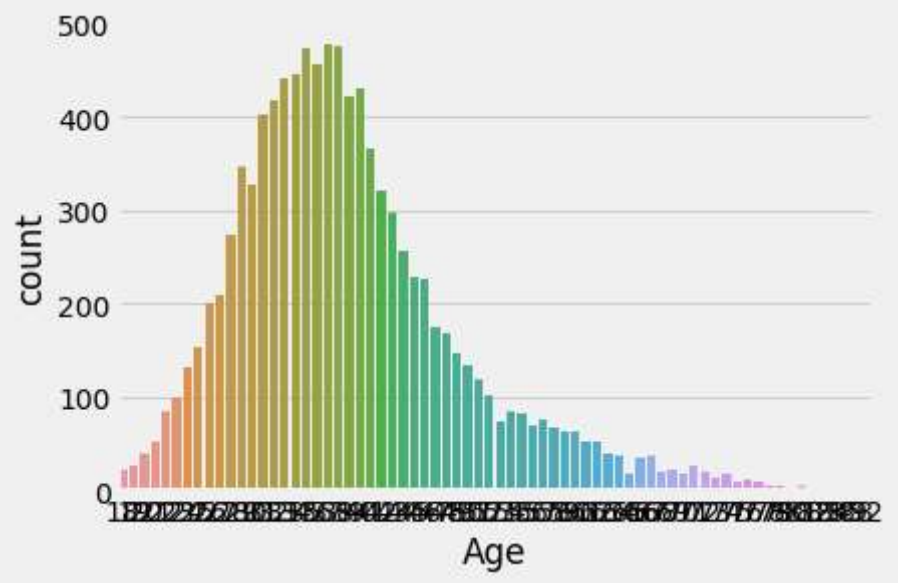
Data Visualizations -> Univariate Analysis:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
plt.style.use("fivethirtyeight")
```

```
sns.countplot(df["Age"])
```

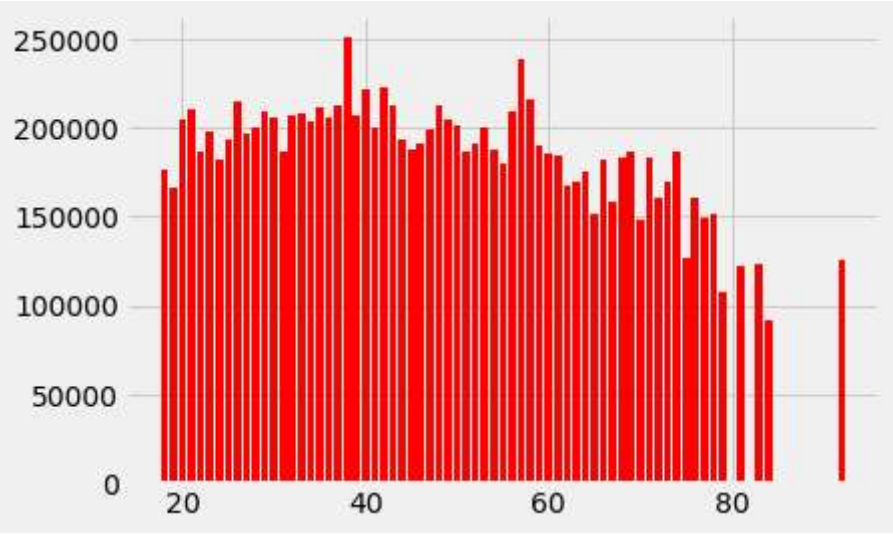
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass th
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f497484a250>
```



-> Bi - Variate Analysis :

```
plt.bar(df["Age"],df["Balance"],color='r')
```

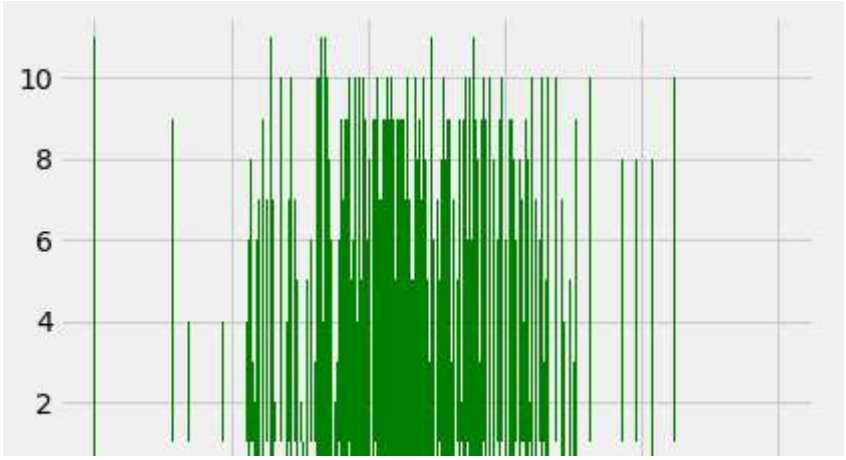
```
<BarContainer object of 10000 artists>
```



-> Multi - Variate Analysis :

```
plt.bar(df["Balance"], df["Tenure"], df["Age", df["IsActiveMember"] ,color='g')
```

```
<BarContainer object of 10000 artists>
```



Descriptive Statistics:

```
df.describe(include="all" )
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age
count	10000.00000	1.000000e+04	10000	10000.000000	10000	10000	10000.000000
unique	NaN	NaN	2932	NaN	3	2	NaN
top	NaN	NaN	Smith	NaN	France	Male	NaN
freq	NaN	NaN	32	NaN	5014	5457	NaN
mean	5000.50000	1.569094e+07	NaN	650.528800	NaN	NaN	38.921800
std	2886.89568	7.193619e+04	NaN	96.653299	NaN	NaN	10.487806
min	1.00000	1.556570e+07	NaN	350.000000	NaN	NaN	18.000000
25%	2500.75000	1.562853e+07	NaN	584.000000	NaN	NaN	32.000000
50%	5000.50000	1.569074e+07	NaN	652.000000	NaN	NaN	37.000000
75%	7500.25000	1.575323e+07	NaN	718.000000	NaN	NaN	44.000000
max	10000.00000	1.581569e+07	NaN	850.000000	NaN	NaN	92.000000

Handling Missing Values:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
```

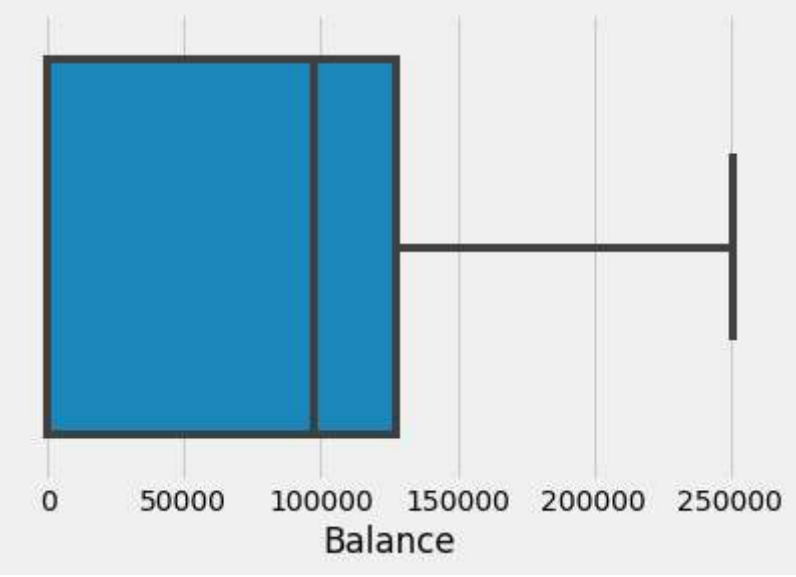
```
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   RowNumber        10000 non-null  int64  
 1   CustomerId       10000 non-null  int64  
 2   Surname          10000 non-null  object 
 3   CreditScore      10000 non-null  int64  
 4   Geography        10000 non-null  object 
 5   Gender           10000 non-null  object 
 6   Age              10000 non-null  int64  
 7   Tenure           10000 non-null  int64  
 8   Balance          10000 non-null  float64
 9   NumOfProducts    10000 non-null  int64  
 10  HasCrCard        10000 non-null  int64  
 11  IsActiveMember   10000 non-null  int64  
 12  EstimatedSalary  10000 non-null  float64
 13  Exited           10000 non-null  int64  
dtypes: float64(2), int64(9), object(3)
memory usage: 1.1+ MB
```

```
#No missing values to handle so moving on to next..
```

Finding Outliners : (Checked Only For Balance Column)

```
sns.boxplot(df["Balance"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4905610bd0>
```




```
#No outliners found..
```

Encoding For Categorial Values: (Label Encoding)

```
from sklearn.preprocessing import LabelEncoder as le
```

```
df['Surname'] = le.fit_transform(df['Surname'], df["Surname"])
df
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Bal
0	1	15634602	0	619	France	Female	42	2	
1	2	15647311	0	608	Spain	Female	41	1	838
2	3	15619304	0	502	France	Female	42	8	1596
3	4	15701354	0	699	France	Female	39	1	
4	5	15737888	0	850	Spain	Female	43	2	1255
...	...	...	...	...	...	...	...	...	
9995	9996	15606229	1	771	France	Male	39	5	
9996	9997	15569892	1	516	France	Male	35	10	573
9997	9998	15584532	0	709	France	Female	36	7	
9998	9999	15682355	1	772	Germany	Male	42	3	750
9999	10000	15628319	0	792	France	Female	28	4	1301

10000 rows × 14 columns

```
df['Geography'] = le.fit_transform(df['Geography'], df["Geography"])
df
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Bal
0	1	15634602	0	619	0	Female	42	2	
1	2	15647311	0	608	2	Female	41	1	838
2	3	15619304	0	502	0	Female	42	8	1596
3	4	15701354	0	699	0	Female	39	1	
4	5	15737888	0	850	2	Female	43	2	1255
...	...	...	...	...	...	...	...	...	
9995	9996	15606229	1	771	0	Male	39	5	
9996	9997	15569892	1	516	0	Male	35	10	573
9997	9998	15584532	0	709	0	Female	36	7	
9998	9999	15682355	1	772	1	Male	42	3	750
9999	10000	15628319	0	792	0	Female	28	4	1301

10000 rows × 14 columns

```
df['Gender'] = le.fit_transform(df['Gender'], df["Gender"])
df
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Bal
0	1	15634602	0	619	0	0	42	2	
1	2	15647311	0	608	2	0	41	1	838
2	3	15619304	0	502	0	0	42	8	1596
3	4	15701354	0	699	0	0	39	1	
4	5	15737888	0	850	2	0	43	2	1255
...	...	...	...	...	...	...	...	...	
9995	9996	15606229	1	771	0	1	39	5	
9996	9997	15569892	1	516	0	1	35	10	573
9997	9998	15584532	0	709	0	0	36	7	
9998	9999	15682355	1	772	1	1	42	3	750
9999	10000	15628319	0	792	0	0	28	4	1301

10000 rows × 14 columns

Dependent And Independent Varibles :

```
#As here the Exited columnn is the dependent/target column which is y !!

x = df.iloc[:,0:12]
x
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Bal
0	1	15634602	0	619	0	0	42	2	
1	2	15647311	0	608	2	0	41	1	838
2	3	15619304	0	502	0	0	42	8	1596

```
y = df["Exited"]
y
```

```
0       1
1       0
2       1
3       0
4       0
       ..
9995    0
9996    0
9997    1
9998    1
9999    0
Name: Exited, Length: 10000, dtype: int64
```

Training And Testing :

```
from sklearn.model_selection import train_test_split

xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size = 0.3, random_state=10)

xtrain
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Bal
5663	5664	15765287	0	850	0	0	38	2	
2840	2841	15714057	1	528	2	1	40	4	

Scaling :

```
from sklearn.preprocessing import MinMaxScaler
```

3442	3443	15784150	0	850	1	0	30	8	1548

```
nm = MinMaxScaler()
```

9372	9373	15775826	1	677	0	1	30	1	781

```
n_xtrain = nm.fit_transform(xtrain)
n_xtrain
```

```
array([[0.56631326, 0.7983751 , 0.        , ..., 0.33333333, 1.        ,
        0.        ],
       [0.28395679, 0.59344198, 1.        , ..., 0.33333333, 1.        ,
        0.        ],
       [0.454991  , 0.95089686, 0.        , ..., 0.33333333, 1.        ,
        0.        ],
       ...,
       [0.13432687, 0.5016241 , 0.        , ..., 0.33333333, 1.        ,
        0.        ],
       [0.72934587, 0.60231855, 0.        , ..., 0.        , 1.        ,
        0.        ],
       [0.12882577, 0.49272753, 1.        , ..., 0.        , 1.        ,
        0.        ]])
```

```
n_xtest = nm.fit_transform(xtest)
n_xtest
```

```
array([[0.09371874, 0.38570863, 0.        , ..., 0.        , 0.        ,
        1.        ],
       [0.93568714, 0.22474162, 0.        , ..., 0.        , 1.        ,
        0.        ],
       [0.22934587, 0.94253122, 0.        , ..., 0.        , 1.        ,
        0.        ],
       ...,
       [0.75415083, 0.67871418, 1.        , ..., 0.        , 1.        ,
        0.        ],
       [0.52760552, 0.05567626, 0.        , ..., 0.33333333, 0.        ,
        1.        ],
       [0.96559312, 0.82983031, 1.        , ..., 0.33333333, 0.        ,
        0.        ]])
```

Colab paid products - Cancel contracts here

0s completed at 13:51