

## ASSIGNMENT 2

ASSIGNMENT DATE	24 SEPTEMBER 2022
STUDENT NAME	HARI @ ARJUN.K
STUDENT ROLL NUMBER	2019504525
MAXIMUM MARKS	2 MARKS

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

*#2.Load the dataset*

```
df = pd.read_csv(r"C:\Users\Karthikeyan\Downloads\
Churn_Modelling.csv")
```

```
df.head()
```

	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

	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	

	EstimatedSalary	Exited
0	101348.88	1
1	112542.58	0
2	113931.57	1
3	93826.63	0
4	79084.10	0

*#4.descriptive statistics*

```
df.describe()
```

	RowNumber	CustomerId	CreditScore	Age
Tenure \				
count	10000.00000	1.000000e+04	10000.000000	10000.000000
mean	5000.50000	1.569094e+07	650.528800	38.921800
std	2886.89568	7.193619e+04	96.653299	10.487806

## ASSIGNMENT 2

2.892174

## ASSIGNMENT 2

min	1.00000	1.556570e+07	350.000000	18.000000
0.000000				
25%	2500.75000	1.562853e+07	584.000000	32.000000
3.000000				
50%	5000.50000	1.569074e+07	652.000000	37.000000
5.000000				
75%	7500.25000	1.575323e+07	718.000000	44.000000
7.000000				
max	10000.00000	1.581569e+07	850.000000	92.000000
10.000000				

	Balance	NumOfProducts	HasCrCard	IsActiveMember	\
count	10000.000000	10000.000000	10000.00000	10000.000000	
mean	76485.889288	1.530200	0.70550	0.515100	
std	62397.405202	0.581654	0.45584	0.499797	
min	0.000000	1.000000	0.00000	0.000000	
25%	0.000000	1.000000	0.00000	0.000000	
50%	97198.540000	1.000000	1.00000	1.000000	
75%	127644.240000	2.000000	1.00000	1.000000	
max	250898.090000	4.000000	1.00000	1.000000	

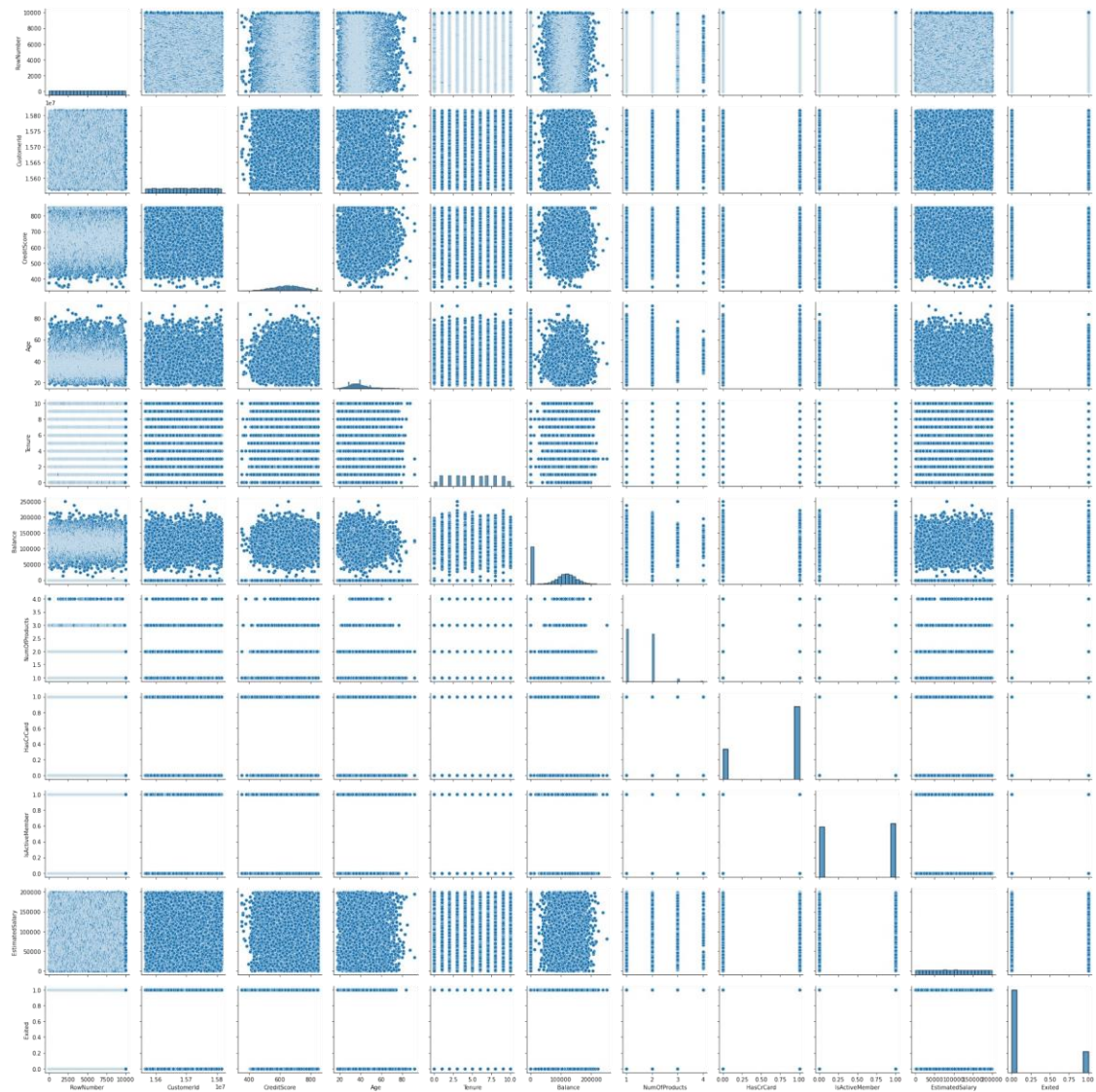
	EstimatedSalary	Exited
count	10000.000000	10000.000000
mean	100090.239881	0.203700
std	57510.492818	0.402769
min	11.580000	0.000000
25%	51002.110000	0.000000
50%	100193.915000	0.000000
75%	149388.247500	0.000000
max	199992.480000	1.000000

*#3.visualizations*

`sns.pairplot(df) # multivariate`

`<seaborn.axisgrid.PairGrid at 0x1f91a9b9d90>`

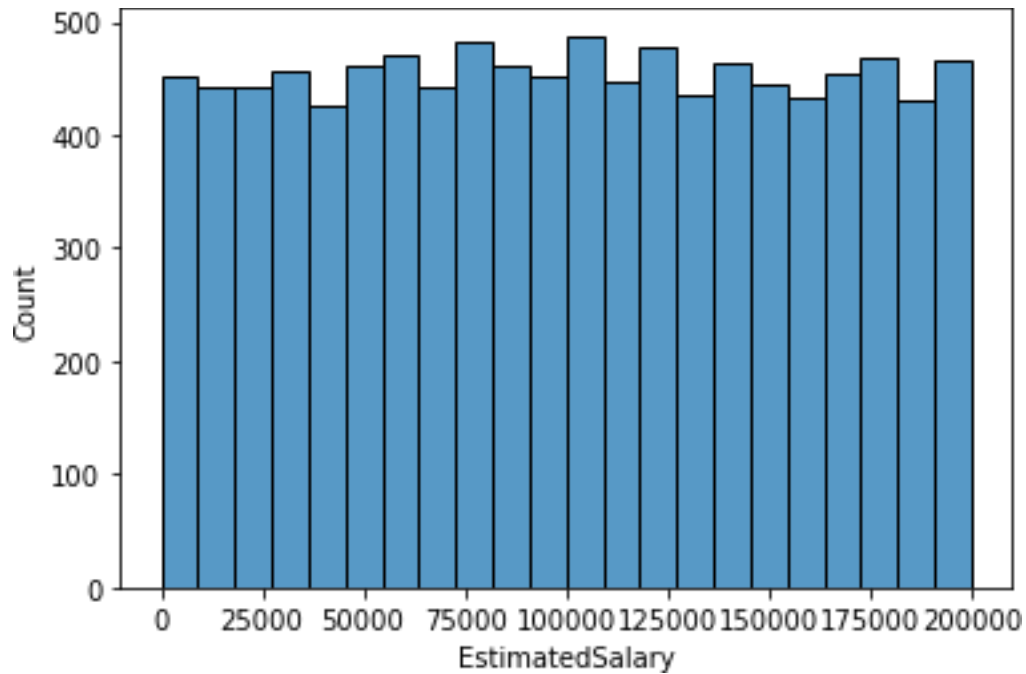
## ASSIGNMENT 2



```
sns.histplot(df["EstimatedSalary"]) # univariate
```

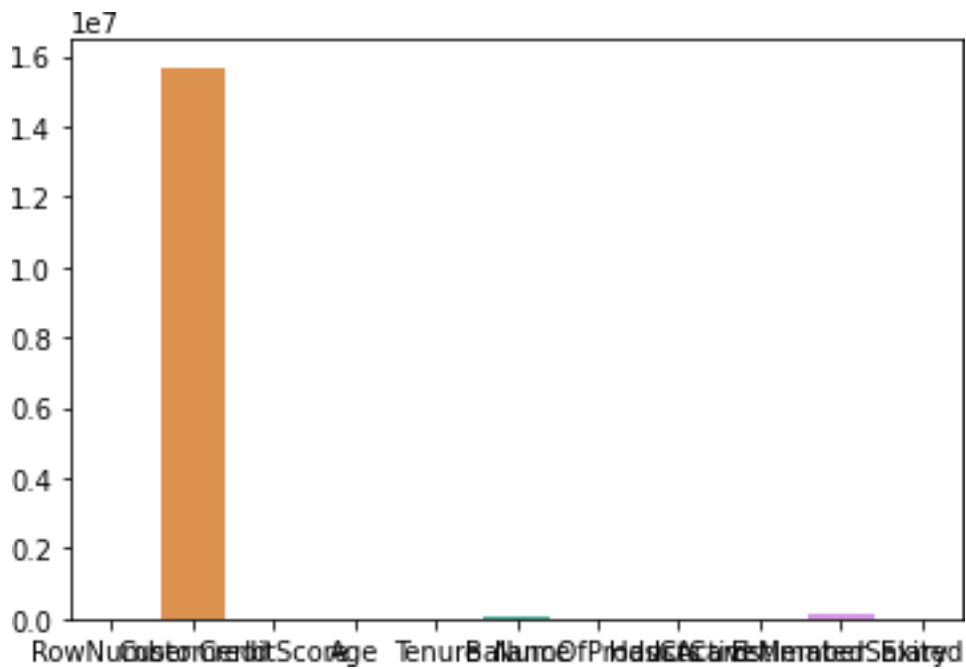
```
<AxesSubplot:xlabel='EstimatedSalary', ylabel='Count'>
```

## ASSIGNMENT 2



```
sns.barplot(data=df) # bivariate
```

```
<AxesSubplot:>
```



```
#8.independent and dependent variables
```

```
x=df.iloc[:,3:13].values  
y=df.iloc[:,13:14].values
```

## ASSIGNMENT 2

```
print(x.shape)
print(y.shape)

(10000, 10)
(10000, 1)

#7.categorical data

from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer([("oh", OneHotEncoder(),
[1,2])],remainder="passthrough")
x=ct.fit_transform(x)

x.shape

(10000, 13)

#10.split data into training and testing

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
print(x_train.shape,x_test.shape)

(8000, 13) (2000, 13)

print(y_train.shape,y_test.shape)

(8000, 1) (2000, 1)

#9.scale independent variables

from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.transform(x_test)

print(x_train.shape,x_test.shape)

(8000, 13) (2000, 13)

#5.handle missing values

df.isnull().sum()

RowNumber      0
CustomerId     0
Surname        0
CreditScore    0
Geography     0
Gender         0
Age           0
Tenure        0
```

## ASSIGNMENT 2

```
Balance          0
NumOfProducts    0
HasCrCard         0
IsActiveMember   0
EstimatedSalary  0
Exited            0
dtype: int64
```

```
#6.outliers
```

```
df.skew()
```

```
C:\Users\Karthikeyan\AppData\Local\Temp\
ipykernel_3748\1665899112.py:1: FutureWarning: Dropping of nuisance
columns in DataFrame reductions (with 'numeric_only=None') is
deprecated; in a future version this will raise TypeError. Select
only valid columns before calling the reduction.
```

```
df.skew()
```

```
RowNumber          0.000000
CustomerId          0.001149
CreditScore       -0.071607
Age                1.011320
Tenure             0.010991
Balance           -0.141109
NumOfProducts      0.745568
HasCrCard          -0.901812
IsActiveMember    -0.060437
EstimatedSalary    0.002085
Exited             1.471611
dtype: float64
```