

ASSIGNMENT-2
DATA VISUALIZATION AND PREPROCESSING

Assignment Date	24 September 2022
Student Name	ABIRAMI P.
Student Roll Number	61771921001
Maximum Marks	2 Marks

TASK-1

DOWNLOAD THE DATASET

Churn_Modelling.csv

TASK-2

LOAD THE DATABASE

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

```
df=pd.read_csv('Churn_Modelling.csv')
```

```
df.head()
```

```
df=pd.read_csv('Churn_Modelling.csv')
df.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

```
df.shape
```

```
df.shape
```

```
(10000, 14)
```

ASSIGNMENT-2

DATA VISUALIZATION AND PREPROCESSING

`df.info()`

```
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
```

`df.isnull().any()`

```
RowNumber      False
CustomerId     False
Surname         False
CreditScore    False
Geography       False
Gender         False
Age            False
Tenure         False
Balance        False
NumOfProducts  False
HasCrCard      False
IsActiveMember False
EstimatedSalary False
Exited         False
dtype: bool
```

`df.Gender.value_counts()`

```
Male      5457
Female    4543
Name: Gender, dtype: int64
```

ASSIGNMENT-2

DATA VISUALIZATION AND PREPROCESSING

```
df.Geography.unique()
```

```
array(['France', 'Spain', 'Germany'], dtype=object)
```

```
df.Age.unique()
```

```
array([42, 41, 39, 43, 44, 50, 29, 27, 31, 24, 34, 25, 35, 45, 58, 32, 38,  
       46, 36, 33, 40, 51, 61, 49, 37, 19, 66, 56, 26, 21, 55, 75, 22, 30,  
       28, 65, 48, 52, 57, 73, 47, 54, 72, 20, 67, 79, 62, 53, 80, 59, 68,  
       23, 60, 70, 63, 64, 18, 82, 69, 74, 71, 76, 77, 88, 85, 84, 78, 81,  
       92, 83])
```

```
df.Tenure.value_counts()
```

```
2      1048  
1      1035  
7      1028  
8      1025  
5      1012  
3      1009  
4       989  
9       984  
6       967  
10      490  
0       413
```

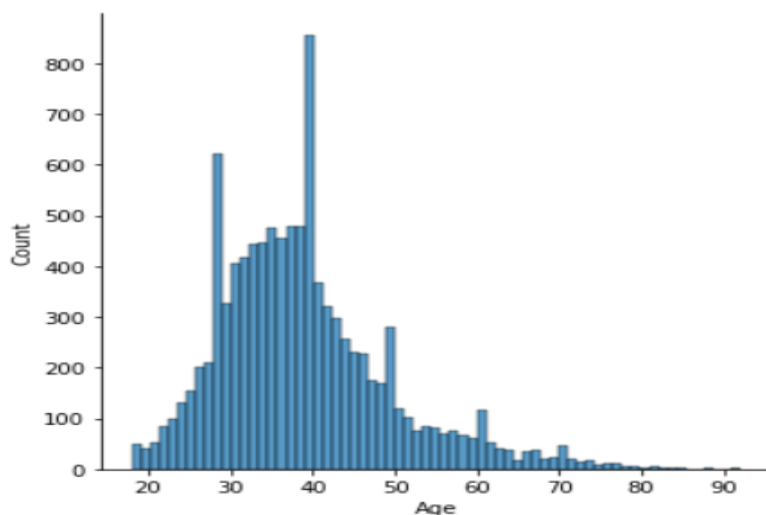
TASK-3

PERFORM BELOW VISUALIZATIONS.

- ❖ Univariate Analysis
- ❖ Bi - Variate Analysis
- ❖ Multi - Variate Analysis

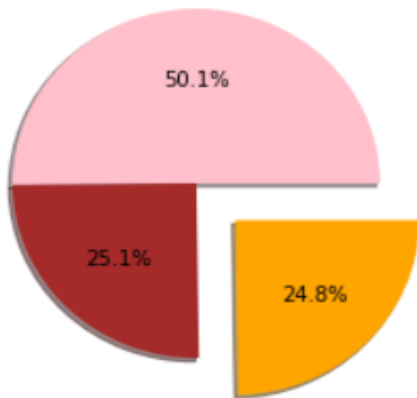
UNIVARIATE ANALYSIS

```
sns.displot(df.CreditScore)
```

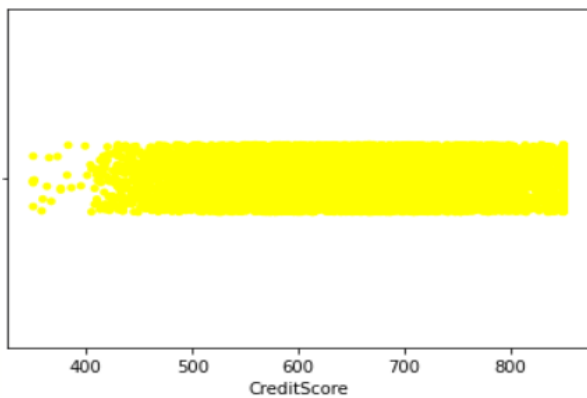


ASSIGNMENT-2
DATA VISUALIZATION AND PREPROCESSING

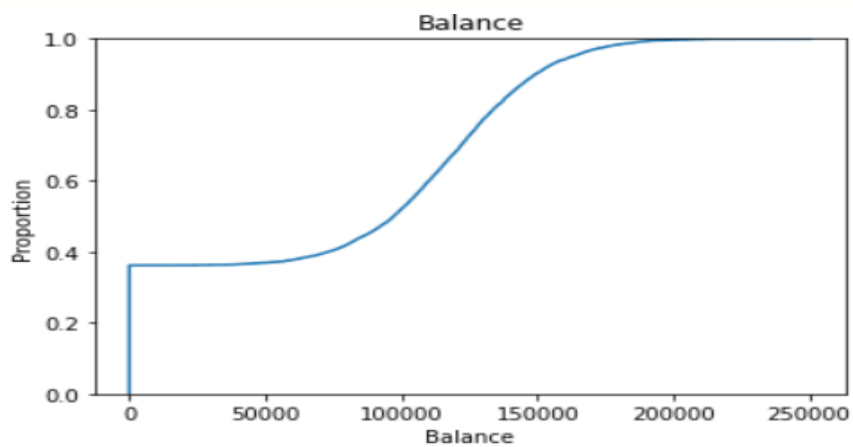
```
plt.pie(df.Geography.value_counts(),[0,0,0.3],shadow='True',autopct="%1.1f%%",colors=['pink','brown','orange']) #categorical column
```



```
sns.stripplot(df.CreditScore,color='yellow')
```



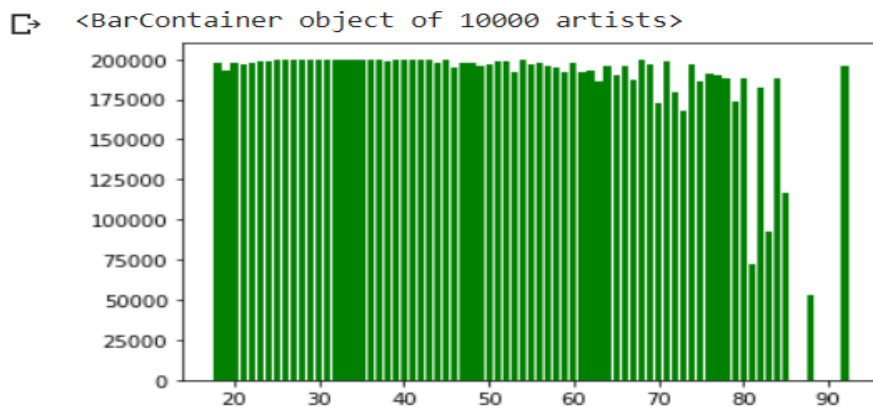
```
sns.ecdfplot(df.Balance)  
plt.title('Balance')
```



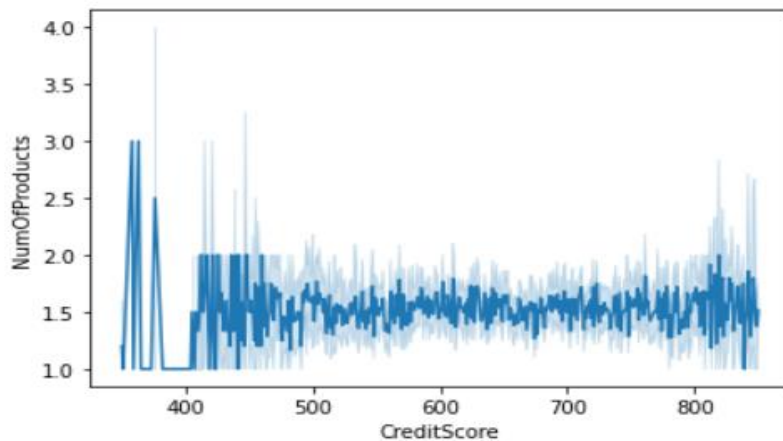
ASSIGNMENT-2
DATA VISUALIZATION AND PREPROCESSING

BI-VARIATE ANALYSIS

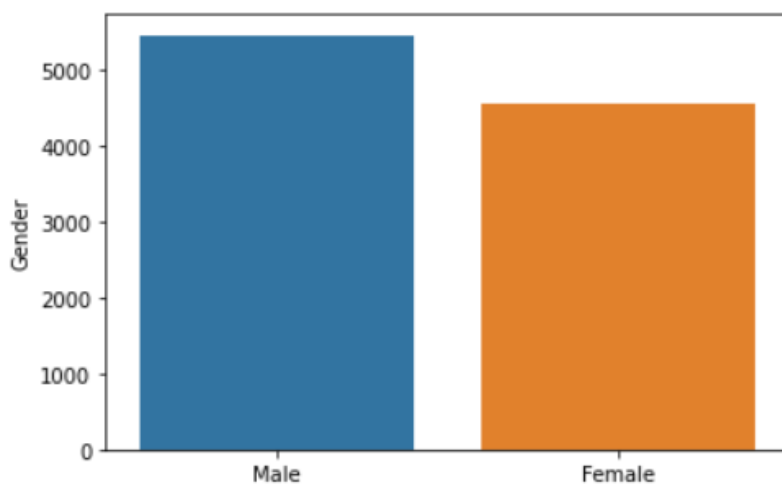
```
plt.bar(df.Age,df.EstimatedSalary,color='green')
```



```
sns.lineplot(df.CreditScore,df.NumOfProducts)
```



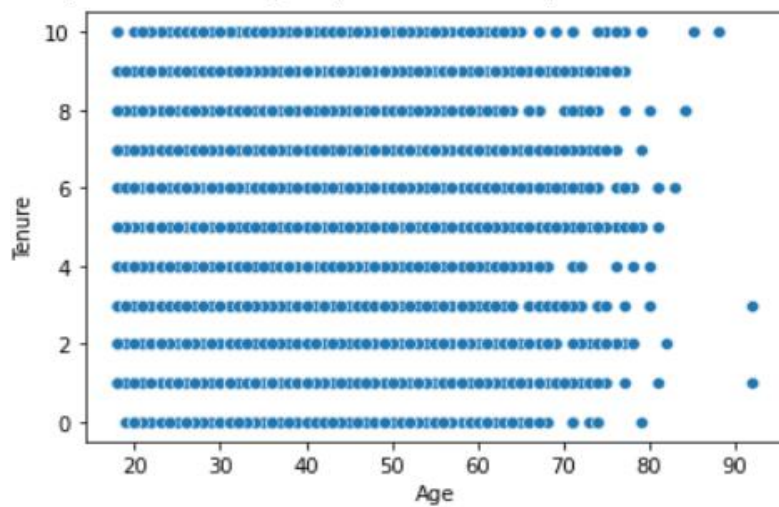
```
sns.barplot(df.Gender.value_counts().index,df.Gender.value_counts())
```



ASSIGNMENT-2

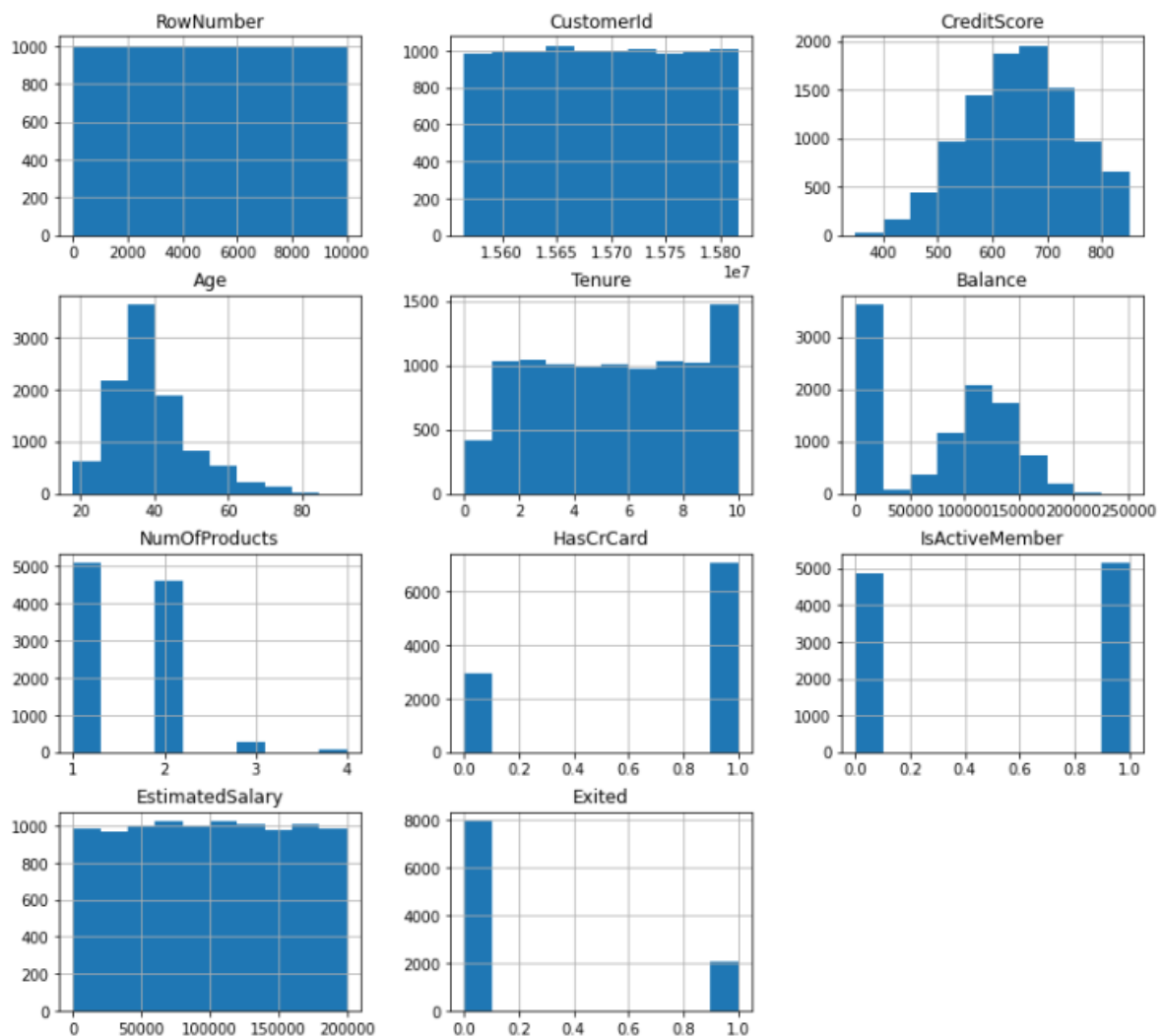
DATA VISUALIZATION AND PREPROCESSING

```
sns.scatterplot(df.Age,df.Tenure)
```



MULTI-VARIATE ANALYSIS

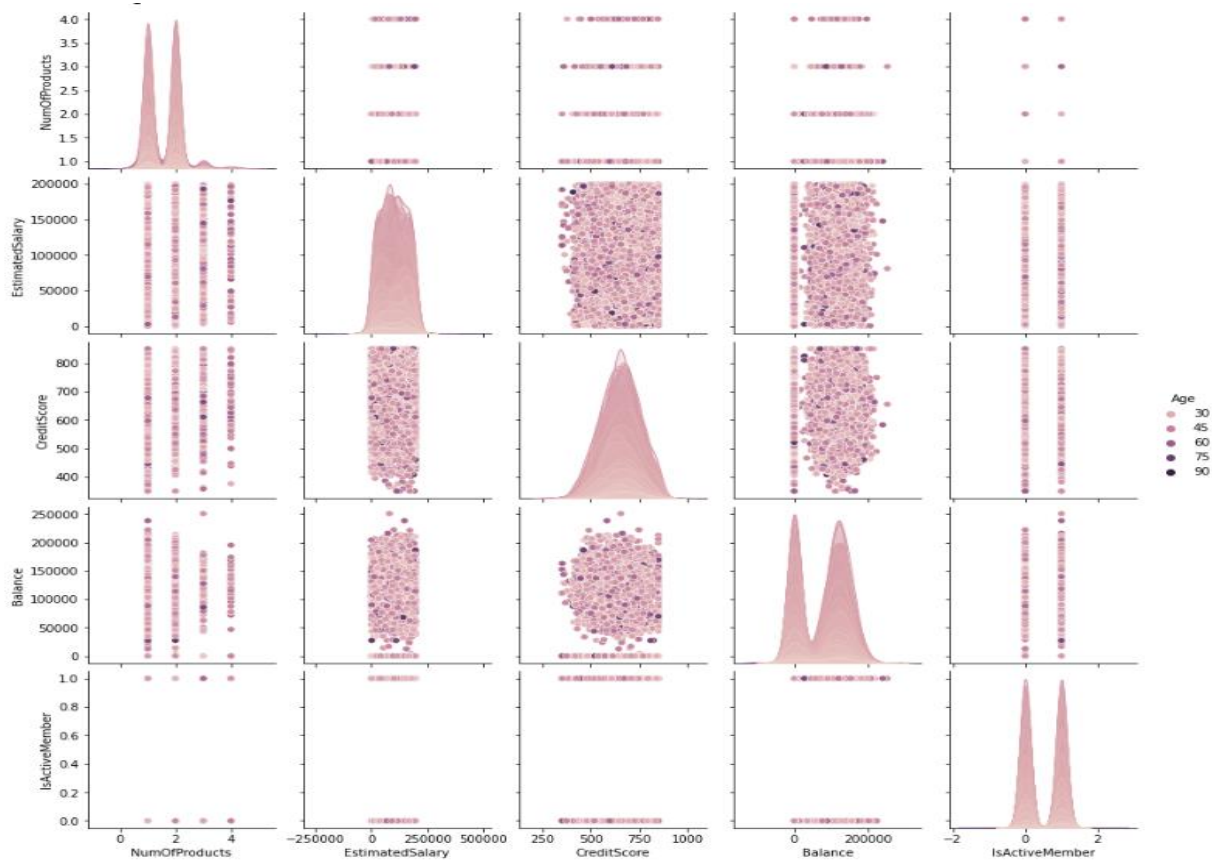
```
multi=df.hist(figsize=(13,12))
```



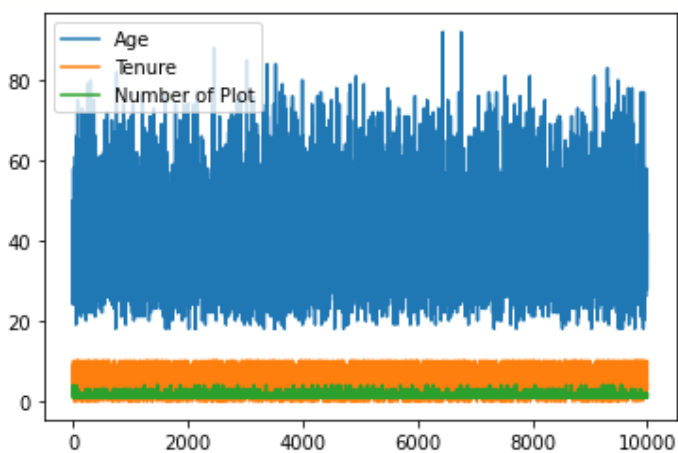
ASSIGNMENT-2

DATA VISUALIZATION AND PREPROCESSING

```
sns.pairplot(data=df[['Geography','Age','NumOfProducts','EstimatedSalary','CreditScore','Balance','IsActiveMember']],hue='Age',kind='scatter')
```



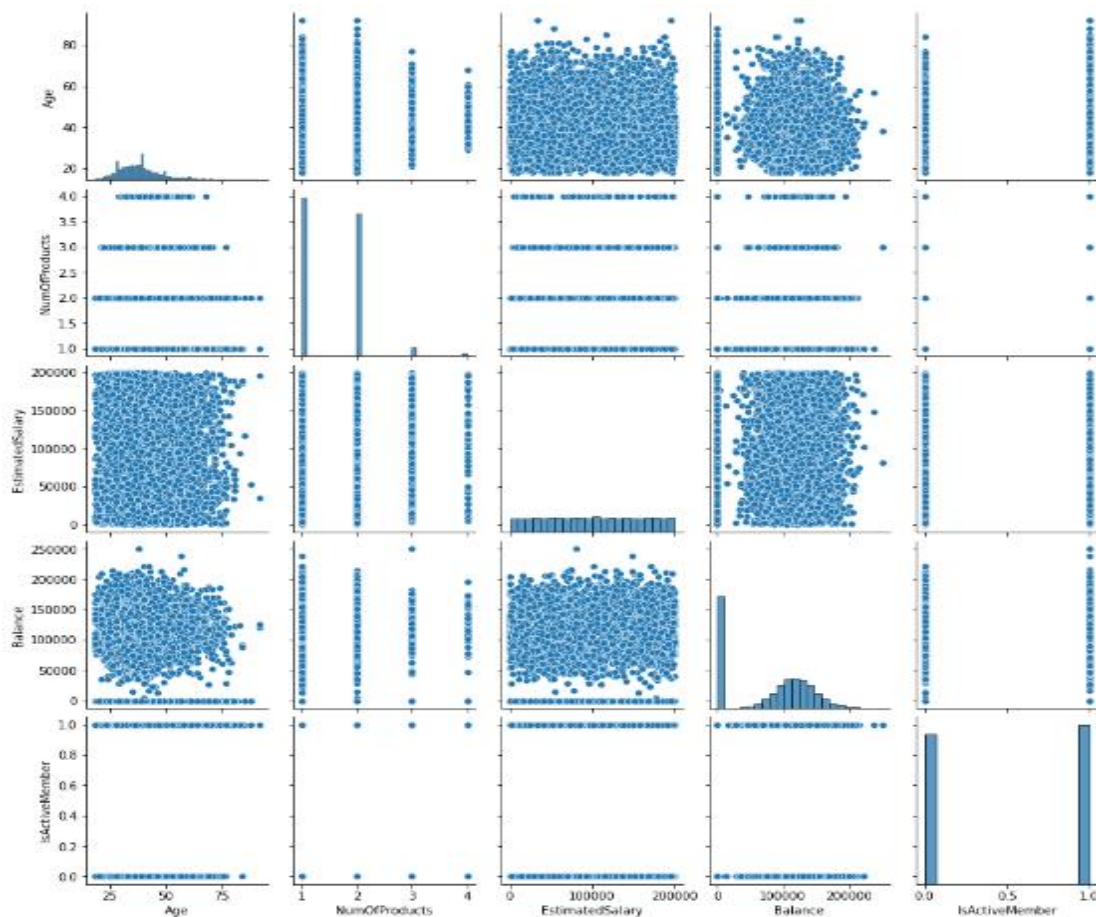
```
df.Age.plot()
df.Tenure.plot()
df.NumOfProducts.plot()
plt.legend(['Age','Tenure','Number of Plot'])
```



ASSIGNMENT-2

DATA VISUALIZATION AND PREPROCESSING

```
sns.pairplot(data=df[['Age','NumOfProducts','EstimatedSalary','Balance','IsActiveMember']],diag_ki
nd='hist')
```



TASK-4

DESCRIPTIVE STATISTICS

```
df.describe()
```

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.00000	10000.000000	10000.000000	10000.000000
mean	5000.50000	1.569094e+07	650.528800	38.921800	5.012800	76485.889288	1.530200	0.70550	0.515100	100090.239881	0.203700
std	2886.89568	7.193619e+04	96.653299	10.487806	2.892174	62397.405202	0.581654	0.45584	0.499797	57510.492818	0.402769
min	1.00000	1.556570e+07	350.000000	18.000000	0.000000	0.000000	1.000000	0.00000	0.000000	11.580000	0.000000
25%	2500.75000	1.562853e+07	584.000000	32.000000	3.000000	0.000000	1.000000	0.00000	0.000000	51002.110000	0.000000
50%	5000.50000	1.569074e+07	652.000000	37.000000	5.000000	97198.540000	1.000000	1.00000	1.000000	100193.915000	0.000000
75%	7500.25000	1.575323e+07	718.000000	44.000000	7.000000	127644.240000	2.000000	1.00000	1.000000	149388.247500	0.000000
max	10000.00000	1.581569e+07	850.000000	92.000000	10.000000	250898.090000	4.000000	1.00000	1.000000	199992.480000	1.000000

ASSIGNMENT-2

DATA VISUALIZATION AND PREPROCESSING

TASK-5

HANDLE THE MISSING DATA

```
df1=pd.read_csv('Churn_Modelling.csv')
df1.head()
```

RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

```
df1.shape
```

```
(10000, 14)
```

```
df1.isnull().any()
```

```
RowNumber      False
CustomerId      False
Surname         False
CreditScore     False
Geography       False
Gender          False
Age             False
Tenure          False
Balance         False
NumOfProducts   False
HasCrCard       False
IsActiveMember  False
EstimatedSalary False
Exited          False
dtype: bool
```

```
df1.isnull().sum()
```

```
RowNumber      0
CustomerId      0
Surname         0
CreditScore     0
Geography       0
Gender          0
Age             0
Tenure          0
Balance         0
NumOfProducts   0
HasCrCard       0
IsActiveMember  0
EstimatedSalary 0
Exited          0
dtype: int64
```

ASSIGNMENT-2

DATA VISUALIZATION AND PREPROCESSING

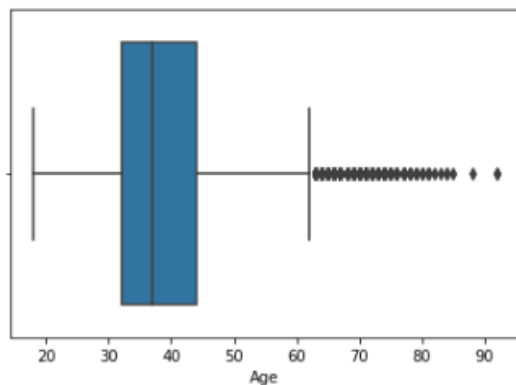
```
df1.info()

<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
```

TASK-6

OUTLIERS REPLACEMENT

```
sns.boxplot(df1.Age)
```



```
q1=df1.Age.quantile(0.25)
```

```
q3=df1.Age.quantile(0.75)
```

```
IQR = q3-q1
```

```
upper_limit=q3 + 1.5 * IQR
```

```
upper_limit
```

```
upper_limit
```

```
62.0
```

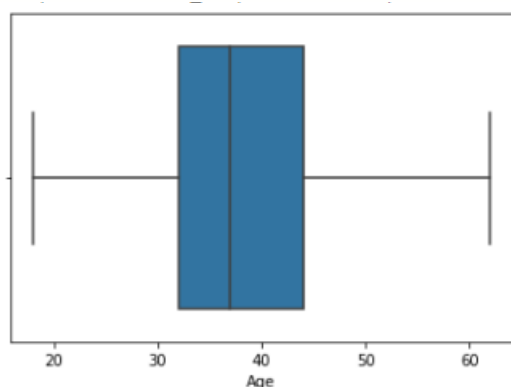
ASSIGNMENT-2

DATA VISUALIZATION AND PREPROCESSING

```
df1.describe()
```

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.00000	10000.000000	10000.000000	10000.000000
mean	5000.50000	1.569094e+07	650.528800	38.921800	5.012800	76485.889288	1.530200	0.70550	0.515100	100090.239881	0.203700
std	2886.89568	7.193619e+04	96.653299	10.487806	2.892174	62397.405202	0.581654	0.45584	0.499797	57510.492818	0.402769
min	1.00000	1.556570e+07	350.000000	18.000000	0.000000	0.000000	1.000000	0.00000	0.000000	11.580000	0.000000
25%	2500.75000	1.562853e+07	584.000000	32.000000	3.000000	0.000000	1.000000	0.00000	0.000000	51002.110000	0.000000
50%	5000.50000	1.569074e+07	652.000000	37.000000	5.000000	97198.540000	1.000000	1.00000	1.000000	100193.915000	0.000000
75%	7500.25000	1.575323e+07	718.000000	44.000000	7.000000	127644.240000	2.000000	1.00000	1.000000	149388.247500	0.000000
max	10000.00000	1.581569e+07	850.000000	92.000000	10.000000	250898.090000	4.000000	1.00000	1.000000	199992.480000	1.000000

```
sns.boxplot(df1.Age)
```



TASK-7

CHECK FOR CATEGORICAL COLUMN AND PERFORM ENCODING
LABEL ENCODING FOR GENDER COLUMN

```
df1.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

```
from sklearn.preprocessing import LabelEncoder
```

```
le = LabelEncoder()
```

```
df1.Gender=le.fit_transform(df1.Gender)
```

```
df1.head()
```

ASSIGNMENT-2

DATA VISUALIZATION AND PREPROCESSING

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	0	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	0	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	0	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	0	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	0	43	2	125510.82	1	1	1	79084.10	0

ONE HOT ENCODING FOR GEOGRAPHY COLUMN

```
df_main=pd.get_dummies(df1,columns=['Geography'])
df_main.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Geography_France	Geography_Germany	Geography_Spain
0	1	15634602	Hargrave	619	0	42	2	0.00	1	1	1	101348.88	1	1	0	0
1	2	15647311	Hill	608	0	41	1	83807.86	1	0	1	112542.58	0	0	0	1
2	3	15619304	Onio	502	0	42	8	159660.80	3	1	0	113931.57	1	1	0	0
3	4	15701354	Boni	699	0	39	1	0.00	2	0	0	93826.63	0	1	0	0
4	5	15737888	Mitchell	850	0	43	2	125510.82	1	1	1	79084.10	0	0	0	1

CORRELATION

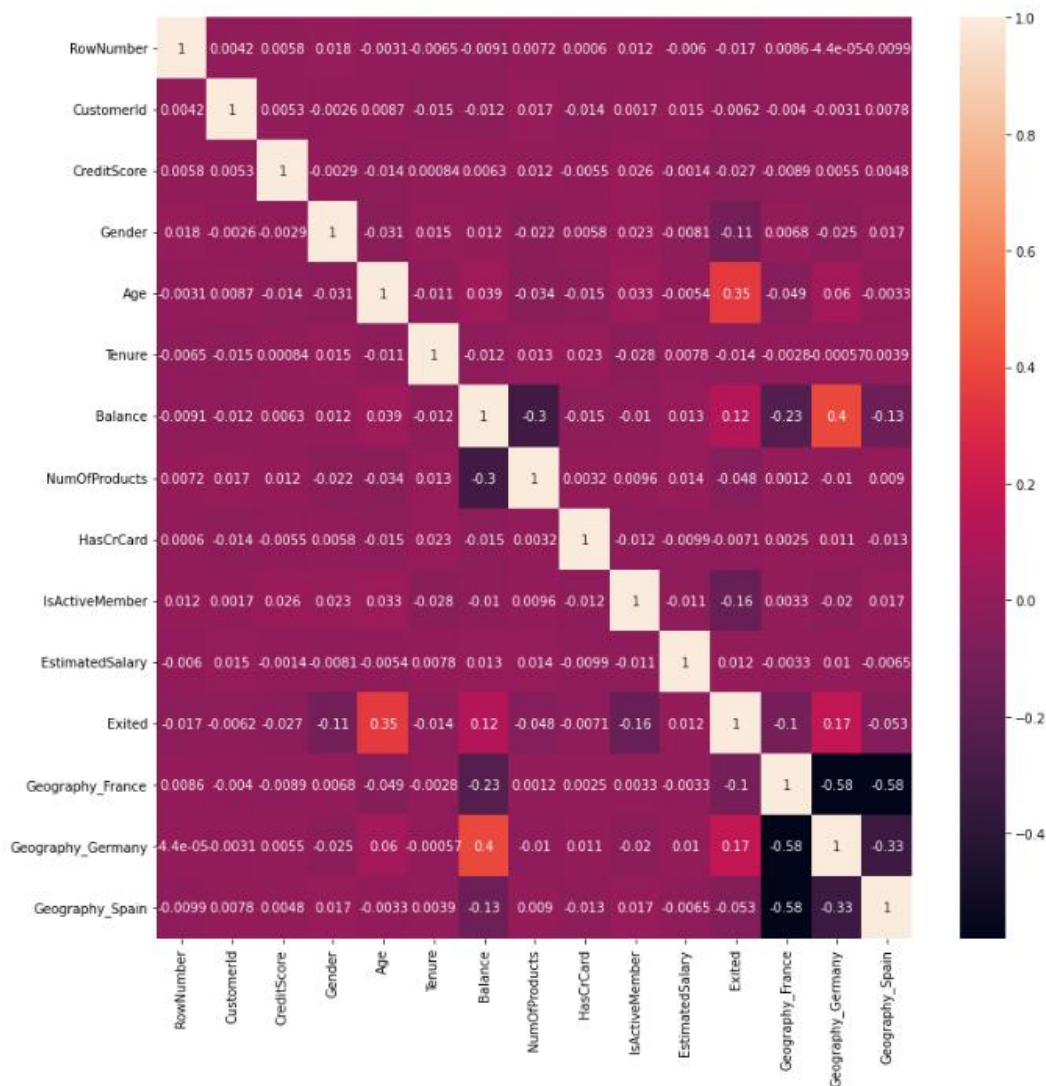
```
df_main.corr()
```

	RowNumber	CustomerId	CreditScore	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Geography_France	Geography_Germany	Geography_Spain
RowNumber	1.000000	0.004202	0.005840	0.018196	-0.003079	-0.006495	-0.009067	0.007246	0.000599	0.012044	-0.005988	-0.016571	0.008590	-0.000044	-0.009905
CustomerId	0.004202	1.000000	0.005308	-0.002641	0.008700	-0.014883	-0.012419	0.016972	-0.014025	0.001665	0.015271	-0.006248	-0.004049	-0.003097	0.007800
CreditScore	0.005840	0.005308	1.000000	-0.002857	-0.013854	0.000842	0.006268	0.012238	-0.005458	0.025651	-0.001384	-0.027094	-0.008928	0.005538	0.004780
Gender	0.018196	-0.002641	-0.002857	1.000000	-0.030740	0.014733	0.012087	-0.021859	0.005766	0.022544	-0.008112	-0.106512	0.006772	-0.024628	0.016889
Age	-0.003079	0.008700	-0.013854	-0.030740	1.000000	-0.010722	0.039293	-0.033586	-0.015448	0.033451	-0.005418	0.349269	-0.048858	0.059633	-0.003299
Tenure	-0.006495	-0.014883	0.000842	0.014733	-0.010722	1.000000	-0.012254	0.013444	0.022583	-0.028362	0.007784	-0.014001	-0.002848	-0.000567	0.003868
Balance	-0.009067	-0.012419	0.006268	0.012087	0.039293	-0.012254	1.000000	-0.304180	-0.014858	-0.010084	0.012797	0.118533	-0.231329	0.401110	-0.134892
NumOfProducts	0.007246	0.016972	0.012238	-0.021859	-0.033586	0.013444	-0.304180	1.000000	0.003183	0.009612	0.014204	-0.047820	0.001230	-0.010419	0.009039
HasCrCard	0.000599	-0.014025	-0.005458	0.005766	-0.015448	0.022583	-0.014858	0.003183	1.000000	-0.011866	-0.009933	-0.007138	0.002467	0.010577	-0.013480
IsActiveMember	0.012044	0.001665	0.025651	0.022544	0.033451	-0.028362	-0.010084	0.009612	-0.011866	1.000000	-0.011421	-0.156128	0.003317	-0.020486	0.016732
EstimatedSalary	-0.005988	0.015271	-0.001384	-0.008112	-0.005418	0.007784	0.012797	0.014204	-0.009933	-0.011421	1.000000	0.012097	-0.003332	0.010297	-0.006482
Exited	-0.016571	-0.006248	-0.027094	-0.106512	0.349269	-0.014001	0.118533	-0.047820	-0.007138	-0.156128	0.012097	1.000000	-0.104955	0.173488	-0.052667
Geography_France	0.008590	-0.004049	-0.008928	0.006772	-0.048858	-0.002848	-0.231329	0.001230	0.002467	0.003317	-0.003332	-0.104955	1.000000	-0.580359	-0.575418
Geography_Germany	-0.000044	-0.003097	0.005538	-0.024628	0.059633	-0.000567	0.401110	-0.010419	0.010577	-0.020486	0.010297	0.173488	-0.580359	1.000000	-0.332084
Geography_Spain	-0.009905	0.007800	0.004780	0.016889	-0.003299	0.003868	-0.134892	0.009039	-0.013480	0.016732	-0.006482	-0.052667	-0.575418	-0.332084	1.000000

ASSIGNMENT-2

DATA VISUALIZATION AND PREPROCESSING

```
plt.figure(figsize=(13,13))
sns.heatmap(df_main.corr(),annot=True)
```



TASK-8

SPLIT THE DATA INTO DEPENDENT AND INDEPENDENT VARIABLES

#y - target columns

#X - predicting columns

y=df_main['Surname']

y

```
0      Hargrave
1      Hill
2      Onio
3      Boni
4      Mitchell
...
9995   Obijiaku
9996   Johnstone
9997   Liu
9998   Sabbatini
9999   Walker
Name: Surname, Length: 10000, dtype: object
```

ASSIGNMENT-2

DATA VISUALIZATION AND PREPROCESSING

```
X=df_main.drop(columns=['Surname'],axis=1)
X.head()
```

	RowNumber	CustomerId	CreditScore	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Geography_France	Geography_Germany	Geography_Spain
0	1	15634602	619	0	42	2	0.00	1	1	1	101348.88	1	1	0	0
1	2	15647311	608	0	41	1	83807.86	1	0	1	112542.58	0	0	0	1
2	3	15619304	502	0	42	8	159660.80	3	1	0	113931.57	1	1	0	0
3	4	15701354	699	0	39	1	0.00	2	0	0	93826.63	0	1	0	0
4	5	15737888	850	0	43	2	125510.82	1	1	1	79084.10	0	0	0	1

TASK-9

SCALE THE INDEPENDENT VARIABLES

```
from sklearn.preprocessing import scale
x_scaled=pd.DataFrame(scale(X),columns=X.columns)
x_scaled.head()
```

	RowNumber	CustomerId	CreditScore	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Geography_France	Geography_Germany	Geography_Spain
0	-1.731878	-0.783213	-0.326221	-1.095988	0.457039	-1.041760	-1.225848	-0.911583	0.646092	0.970243	0.021886	1.977165	0.997204	-0.578736	-0.573809
1	-1.731531	-0.606534	-0.440036	-1.095988	0.342361	-1.387538	0.117350	-0.911583	-1.547768	0.970243	0.216534	-0.505775	-1.002804	-0.578736	1.742740
2	-1.731185	-0.995885	-1.536794	-1.095988	0.457039	1.032908	1.333053	2.527057	0.646092	-1.030670	0.240687	1.977165	0.997204	-0.578736	-0.573809
3	-1.730838	0.144767	0.501521	-1.095988	0.113004	-1.387538	-1.225848	0.807737	-1.547768	-1.030670	-0.108918	-0.505775	0.997204	-0.578736	-0.573809
4	-1.730492	0.652659	2.063884	-1.095988	0.571717	-1.041760	0.785728	-0.911583	0.646092	0.970243	-0.365276	-0.505775	-1.002804	-0.578736	1.742740

TASK-10

SPLIT THE 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_scaled,y,test_size=0.3,random_state=0)
```

```
X_train.shape
```

```
(7000, 15)
```

```
X_test.shape
```

```
(3000, 15)
```

```
y_train.shape
```

```
(7000,)
```

```
y_test.shape
```

```
(3000,)
```