

Assignment -2
PYTHON PROGRAM

Assignment Date	21 September 2022
Student Name	D.Dharshini
Student Roll Number	912619104005
Maximum Marks	2 Marks

Question-1:

Download the dataset: [Dataset](#)

Solution:

DATA PROCESSING

1.DOWNLOAD THE DATASET

Rectangular Snip

The given dataset has been downloaded successfully

2.LOAD THE DATASET

Question-2:

Load the dataset.

Solution:

2. LOAD THE DATASET

```
[ ] import numpy as np
```

```
[ ] import pandas as pd
```

```
[ ] df = pd.read_csv("Churn_Modelling.csv")
```

```
[ ] df
```

	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

```
[ ] 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

```
[ ] import numpy as np
```

```
[ ] import pandas as pd
```

```
[ ] import matplotlib.pyplot as plt
```

Question-3:

Perform Below Visualizations.

3 a) Univariate Analysis

+ Code + Text

Connect

Editing

^

3) Perform Below Visualizations. a) Univariate Analysis

```
[ ] import numpy as np
```

```
[ ] import pandas as pd
```

```
[ ] import matplotlib.pyplot as plt
```

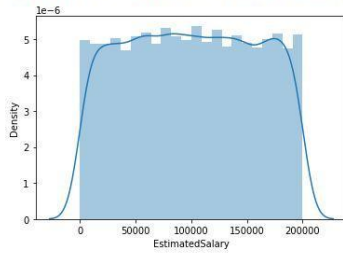
```
[ ] import seaborn as sns
```

```
[ ] data=pd.read_csv("Churn_Modelling.csv")
data.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

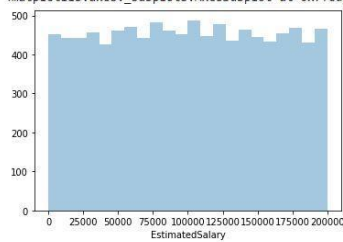
```
sns.distplot(data['EstimatedSalary'])
```

/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: 'distplot' is a deprecated function and will be removed in a future version.
warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7fed70be7490>



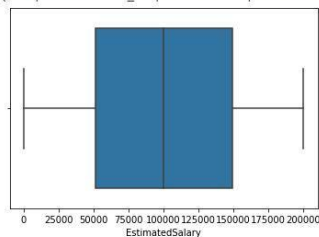
```
[ ] sns.distplot(data['EstimatedSalary'],kde=False)
```

/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: 'distplot' is a deprecated function and will be removed in a future version.
warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7fed70666250>



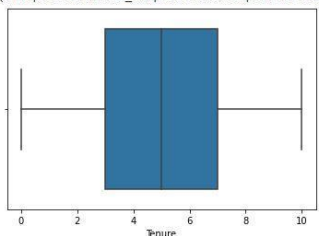
```
[ ] sns.boxplot(data['EstimatedSalary']),
```

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fed714aaf10>)



```
sns.boxplot(data['Tenure']),
```

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fed7051dc90>)



3 b) Bi - Variate Analysis

3 b) Bi - Variate Analysis

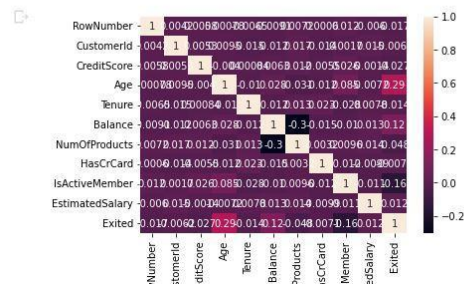
```
[ ] import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

```
[ ] data=pd.read_csv("Churn_Modelling.csv")
```

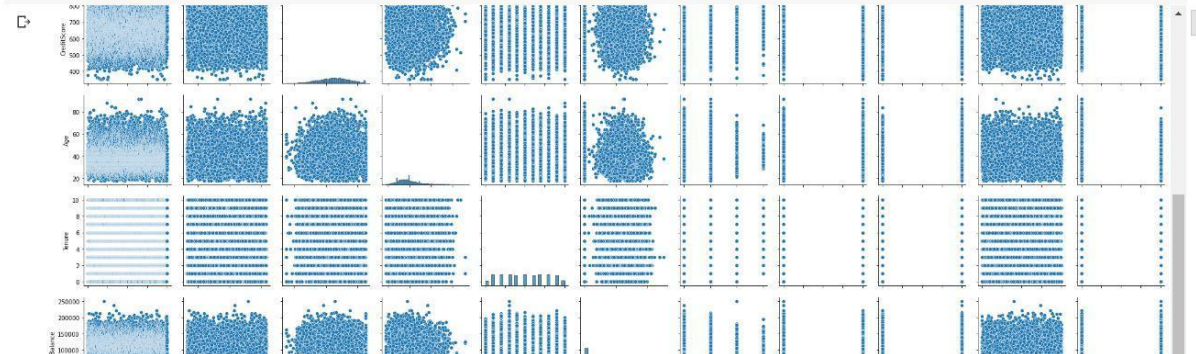
```
[ ] data.head()
```

	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

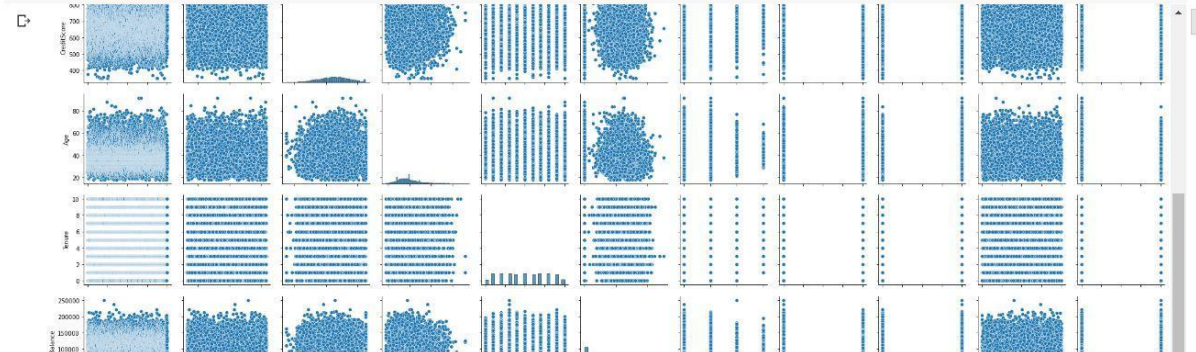
```
sns.heatmap(data.corr(),annot=True)
plt.show()
```



```
sns.pairplot(data)
plt.show()
```



```
sns.pairplot(data)
plt.show()
```



3 c) Multi - Variate Analysis

3 C)MULTI-VARIATE ANALYSIS

```
[ ] from pydoc import help
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import scale
from sklearn.decomposition import PCA
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from scipy import stats
from IPython.display import display,HTML
%matplotlib inline
np.set_printoptions(suppress=True)
pd.set_option('display.max_rows',20)
import os
print(os.listdir("../NT project/"))
```

```
FileNotFoundError                                Traceback (most recent call last)
<ipython-input-114-7c611291f384> in <module>
.....
```

```
data=pd.read_csv("Churn_Modelling.csv")
data.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

```
[ ] data.columns
```

```
Index(['RowNumber', 'CustomerId', 'Surname', 'CreditScore', 'Geography',
      'Gender', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard',
      'IsActiveMember', 'EstimatedSalary', 'Exited'],
      dtype='object')
```

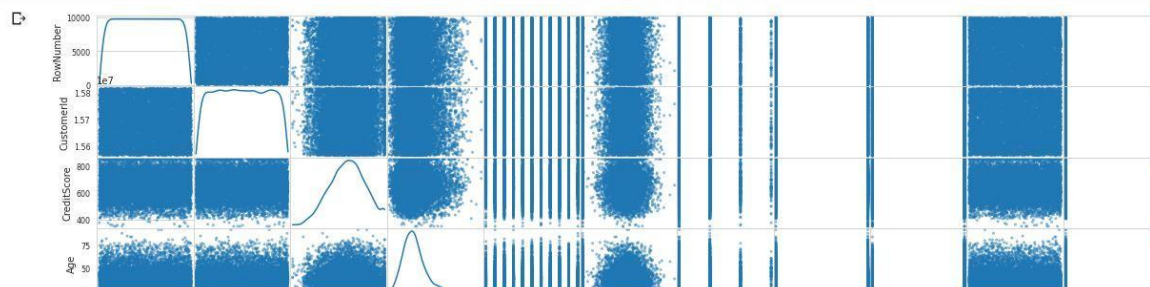
```
[ ] data.info()
```

```
[ ] data.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
```

MATRIX SCATTERPLOT

```
pd.plotting.scatter_matrix(data.loc[:, "RowNumber":"Exited"],diagonal="kde",figsize=(20,15))
plt.show()
```



Question-4:

Perform descriptive statistics on the dataset

4. DESCRIPTIVE STATISTICS

```
[ ] import numpy as np
import pandas as pd
from pandas import Series, DataFrame
import scipy
from scipy import stats
```

```
data=pd.read_csv("Churn_Modelling.csv")
data.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	15727698	Mitchell	850	Spain	Female	43	2	125510.83	1	1	1	76084.40	0

t.

```
data.sum()
```

RowNumber	50005000
CustomerId	156909405694
Surname	HargraveHillOnioBoniMitchellChuBartlettObinnaH...
CreditScore	6505288
Geography	FranceSpainFranceFranceSpainSpainFranceGermany...
Gender	FemaleFemaleFemaleFemaleFemaleMaleMaleFemaleMa...
Age	389218
Tenure	50128
Balance	764858892.88
NumOfProducts	15302
HasCrCard	7055
IsActiveMember	5151
EstimatedSalary	1000902398.81
Exited	2037
dtype: object	

```
[ ] data.sum(axis=1)
```

0	15736618.88
1	15844315.44
2	15802456.37

```
[ ] data.median()
```

RowNumber	5.000500e+03
CustomerId	1.569074e+07
CreditScore	6.520000e+02
Age	3.700000e+01
Tenure	5.000000e+00
Balance	9.719854e+04
NumOfProducts	1.000000e+00
HasCrCard	1.000000e+00
IsActiveMember	1.000000e+00
EstimatedSalary	1.001939e+05
Exited	0.000000e+00
dtype: float64	

```
[ ] data.mean()
```

RowNumber	5.000500e+03
CustomerId	1.569094e+07
CreditScore	6.505288e+02
Age	3.892180e+01
Tenure	5.012800e+00

```
data.max()

RowNumber      10000
CustomerId     15815690
Surname        Zuyeva
CreditScore    850
Geography      Spain
Gender         Male
Age           92
Tenure         10
Balance        250898.09
NumOfProducts  4
HasCrCard      1
IsActiveMember 1
EstimatedSalary 199992.48
Exited         1
dtype: object

[ ] mpg=data.EstimatedSalary
    mpg.idxmax()

6646
```

LOOKING AT SUMMARY STATISTICS THAT DESCRIBE VARIABLE DISTRIBUTION

```
[ ] data.std()

RowNumber      2886.895680
CustomerId     71936.186123
CreditScore    96.653299
Age            10.487806
Tenure         2.892174
Balance        62397.405202
NumOfProducts  0.581654
HasCrCard      0.455840
IsActiveMember 0.499797
EstimatedSalary 57510.492818
Exited         0.402769
dtype: float64

data.var()

RowNumber      8.334167e+06
CustomerId     5.174815e+09
CreditScore    9.341860e+03
```

```
[ ] num=data.NumOfProducts
    num.value_counts()

1    5084
2    4590
3     266
4       60
Name: NumOfProducts, dtype: int64

[ ] data.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  1.566570e+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
max     9999.99999  1.569094e+07  850.000000  92.000000  10.000000  250898.090000  4.000000  1.00000  1.000000  199992.480000  1.000000
```

Question-5:

Handle the Missing values

5.HANDLE MISSING VALUE

```
[ ] import pandas as pd
```

```
[ ] data=pd.read_csv("Churn_Modelling.csv")
data.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

```
[ ] data.shape
```

```
[ ] data.shape
```

(10000, 14)

```
data.isnull()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	False	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	False
2	False	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	False
4	False	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	False
9996	False	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	False

10000 rows x 14 columns

```
[ ] data.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
```

```
[ ] data.isnull().sum().sum()
```

0

FILLING NULL VALUES

```
df=data.fillna(value=0)
df
```

	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
...
9995	9996	15606229	Obijaku	771	France	Male	39	5	0.00	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	101699.77	0
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	42085.58	1

10000 rows x 14 columns

```
[ ] df.isnull().sum().sum()
```

0

```
[ ] df1=data.fillna(value=5)
df1
```

	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
...

FILLING NULL VALUES WITH A PREVIOUS VALUE

```
[ ] df2=data.fillna(method='pad')
df2
```

	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
...
9995	9996	15606229	Obijaku	771	France	Male	39	5	0.00	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	101699.77	0

FILLING NULL VALUES WITH A PREVIOUS VALUE

```
[ ] df2=data.fillna(method='pad')
df2
```

	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
...
9995	9996	15606229	Obijaku	771	France	Male	39	5	0.00	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	101699.77	0

```
[ ] df2.isnull().sum().sum()
```

0

```
[ ] #filling NULL values with the next value
df3=data.fillna(method='bfill')
df3
```

	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
...
9995	9996	15606229	Obijaku	771	France	Male	39	5	0.00	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	101699.77	0
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	42085.58	1
9998	9999	15669855	Costantini	735	Germany	Male	40	8	75275.84	0	1	0	68298.58	1

DROPPING NULL VALUES

```
df4=data.dropna()
df4
```

	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
...
9995	9996	15606229	Obijaku	771	France	Male	39	5	0.00	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	101699.77	0
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	42085.58	1
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	92888.52	1
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	38190.78	0

```
df5=data.dropna(how='any')
df5
```

	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
...
9995	9996	15606229	Obijaku	771	France	Male	39	5	0.00	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	101699.77	0
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	42085.58	1
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	92888.52	1
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	38190.78	0

replace()

```
import numpy as np
df6=df.replace(to_replace=np.nan,value=8763)
df6
```

	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
...
9995	9996	15606229	Obijaku	771	France	Male	39	5	0.00	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	101699.77	0
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	42085.58	1

interpolate()

```
data['EstimatedSalary']=data['EstimatedSalary'].interpolate(method='linear')
data
```

	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
...
9995	9996	15606229	Obijaku	771	France	Male	39	5	0.00	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	101699.77	0
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	42085.58	1
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	92888.52	1
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	38190.78	0

Question-6:

Find the outliers and replace the outliers

6.FIND THE OUTLIERS AND REPLACE THE OUTLIERS

```
[ ] import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

```
[ ] data=pd.read_csv("Churn_Modelling.csv")
data1=data["CreditScore"]
outliers=[]
def detect_outliers(data):
    threshold=3
    mean=np.mean(data)
    std=np.std(data)
    for i in data:
        z_score=(i-mean)/std
        if np.abs(z_score)>threshold:
            outliers.append(z_score)
    return outliers
```

```
[ ] outlier_pt=detect_outliers(data1)
```

+ Code

+ Text

▶ outlier_pt

```
[ ] outlier_pt=detect_outliers(data1)
```

```
[ ] outlier_pt
```

INTERQUANTILE RANGE

```
▶ sorted(data1)
```

```
↗ 351,  
   358,  
   359,  
   363,  
   365,  
   367,  
   373,  
   376,  
   376,  
   382,  
   383,  
   386,  
   395,  
   399,  
   401,  
   404,  
   405,
```

```
521,  
521,  
521,  
521,  
521,  
521,  
521,  
521,  
521,  
521,  
...]
```

```
[ ] quantile1,quantile3=np.percentile(data1,[25,75])
```

```
[ ] print(quantile1,quantile3)
```

```
584.0 718.0
```

```
[ ] iqr_value=quantile3-quantile1  
print(iqr_value)
```

```
134.0
```

```
[ ] lower_bound_val=quantile1-(1.5*iqr_value)
```

```
▶ quantile1,quantile3=np.percentile(data1,[25,75])
```

```
[ ] print(quantile1,quantile3)
```

```
584.0 718.0
```

```
[ ] iqr_value=quantile3-quantile1  
    print(iqr_value)
```

```
134.0
```

```
[ ] lower_bound_val=quantile1-(1.5*iqr_value)  
    upper_bound_val=quantile3+(1.5*iqr_value)
```

```
[ ] print(lower_bound_val,upper_bound_val)
```

```
383.0 919.0
```

7. CHECK FOR CATEGORICAL COLUMNS AND PERFORM ENCODING

Question-7:

Check for Categorical columns and perform encoding.

7. CHECK FOR CATEGORICAL COLUMNS AND PERFORM ENCODING

```
[ ] import pandas as pd
import numpy as np
import seaborn as sns
%matplotlib inline
```

METHOD I

```
[ ] data=pd.read_csv("Churn_Modelling.csv")
NEW_DataM1=data
data1=pd.get_dummies(NEW_DataM1["Gender"])
```

```
[ ] data1.head()
```

```
      Female  Male
0          1     0
1          1     0
```

```
2          1     0
3          1     0
4          1     0
```

```
NEW_DataM1.drop('Gender',axis='columns')
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	43	2	125510.82	1	1	1	79084.10	0
...
9995	9996	15606229	Obijaku	771	France	39	5	0.00	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	516	France	35	10	57369.61	1	1	1	101699.77	0
9997	9998	15584532	Liu	709	France	36	7	0.00	1	0	1	42085.58	1
9998	9999	15682355	Sabbatini	772	Germany	42	3	75075.31	2	1	0	92888.52	1

```
9999 10000 15628319 Walker 792 France 28 4 130142.79 1 1 0 38190.78 0
10000 rows x 13 columns
```

[+ Code](#) [+ Text](#)

```
[ ] NEW_DataM1["Male"]=data1["Male"].to_list()
NEW_DataM1["Female"]=data1["Female"].to_list()
```

```
[ ] NEW_DataM1
```

	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
...
9995	9996	15606229	Obijaku	771	France	Male	39	5	0.00	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	101699.77	0
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	42085.58	1

9997	9996	15684532	Liu	769	France	Female	30	7	0.00	1	0	1	42685.30	1	0
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	92888.52	1	1
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	38190.78	0	0

10000 rows x 16 columns

```
[ ] NEW_DataM1.head(2)
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Male
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1	0
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0	0

METHOD II

```
[ ] from sklearn.preprocessing import LabelEncoder
```

```
[ ] data=pd.read_csv("Churn_Modelling.csv")
l3=LabelEncoder()
label=l3.fit_transform(data["Gender"])
```

```
[ ] from sklearn.preprocessing import LabelEncoder
```

```
data=pd.read_csv("Churn_Modelling.csv")
l3=LabelEncoder()
label=l3.fit_transform(data["Gender"])
```

+ Code

+ Text

```
[ ] l3.classes_
```

```
array(['Female', 'Male'], dtype=object)
```

```
[ ] Data=NEW_DataM1.drop("Gender",axis='columns')
Data
```

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

9995999615606229Obijaku771France3950.0021096270.6401

9996999715569892Johnstone516France351057369.61111101699.7701

9997999815584532Liu709France3670.0010142085.5810

9998999915682355Sabbatini772Germany42375075.3121092888.5211

99991000015628319Walker792France284130142.7911038190.7800

10000 rows x 15 columns

<

>

[] Data["Gender"] = label

Data

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

▶

>

▶ Data["Gender"] = label

Data

⌵

	RowNumber	CustomerId	Surname	CreditScore	Geography	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Male	Female
0	1	15634602	Hargrave	619	France	42	2	0.00	1	1	1	101348.88	1	0	
1	2	15647311	Hill	608	Spain	41	1	83807.86	1	0	1	112542.58	0	0	
2	3	15619304	Onio	502	France	42	8	159660.80	3	1	0	113931.57	1	0	
3	4	15701354	Boni	699	France	39	1	0.00	2	0	0	93826.63	0	0	
4	5	15737888	Mitchell	850	Spain	43	2	125510.82	1	1	1	79084.10	0	0	
...
9995	9996	15606229	Obijaku	771	France	39	5	0.00	2	1	0	96270.64	0	1	
9996	9997	15569892	Johnstone	516	France	35	10	57369.61	1	1	1	101699.77	0	1	
9997	9998	15584532	Liu	709	France	36	7	0.00	1	0	1	42085.58	1	0	
9998	9999	15682355	Sabbatini	772	Germany	42	3	75075.31	2	1	0	92888.52	1	1	
9999	10000	15628319	Walker	792	France	28	4	130142.79	1	1	0	38190.78	0	0	

10000 rows x 16 columns

Question-8:

Split the data into dependent and independent variables.

8.SPLIT THE DATA INTO DEPENDENT AND INDEPENDENT VARIABLES

```
[ ] import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

```
[ ] data=pd.read_csv("Churn_Modelling.csv")
```

```
X=data.iloc[:,2:9]
X
```

	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance
0	Hargrave	619	France	Female	42	2	0.00
1	Hill	608	Spain	Female	41	1	83807.86
2	Onio	502	France	Female	42	8	159660.80
3	Boni	699	France	Female	39	1	0.00
4	Mitchell	850	Spain	Female	43	2	125510.82
...
9995	Obijiaku	771	France	Male	39	5	0.00
9996	Johnstone	516	France	Male	35	10	57369.61
9997	Liu	709	France	Female	36	7	0.00
9998	Sabbatini	772	Germany	Male	42	3	75075.31
9999	Walker	792	France	Female	28	4	130142.79

10000 rows x 7 columns

```
[ ] Y=data.iloc[:,9]
Y
```

```
0      1
1      1
2      3
3      2
4      1
...
9995   2
9996   1
9997   1
9998   2
9999   1
```

Question-9:

Scale the independent variables

```
Name: NumOfProducts, Length: 10000, dtype: int64
```

9. SCALE THE INDEPENDENT VARIABLES

```
[ ] import numpy as np
import pandas as pd
from pandas import Series, DataFrame
import matplotlib.pyplot as plt
from pylab import rcParams
import seaborn as sb
import scipy
import sklearn
from sklearn import preprocessing
from sklearn.preprocessing import scale
```

```
[ ] %matplotlib inline
rcParams['figure.figsize']=5,4
sb.set_style('whitegrid')
```

Normalizing and transforming features with MinMaxScaler() and fit_transform()

```
[ ] data=pd.read_csv("Churn_Modelling.csv")
```

Normalizing and transforming features with MinMaxScaler() and fit_transform()

```
[ ] data=pd.read_csv("Churn_Modelling.csv")
```

```
[ ] data.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

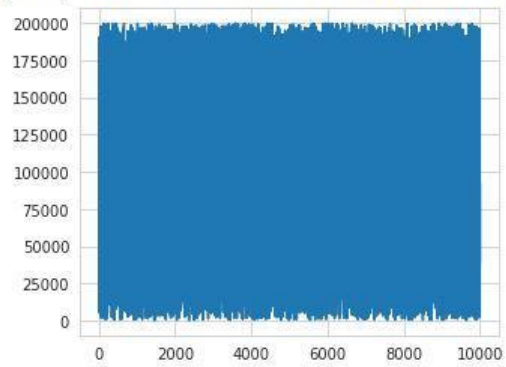
```
[ ] tenure=data.EstimatedSalary
plt.plot(tenure)
```

```
[<matplotlib.lines.Line2D at 0x7fed680f7490>]
200000
175000
```



```
plt.plot(tenure)
```

```
[<matplotlib.lines.Line2D at 0x7fed680f7490>]
```



```
[ ] data[['Tenure']].describe()
```

Tenure	
count	10000.000000
mean	5.012800
std	2.892174

min	0.000000
25%	3.000000
50%	5.000000
75%	7.000000
max	10.000000

```
tenure_matrix=tenure.values.reshape(-1,1)
scaled=preprocessing.MinMaxScaler()
scaled_tenure=scaled.fit_transform(tenure_matrix)
plt.plot(scaled_tenure)
```

```
[<matplotlib.lines.Line2D at 0x7fed680e1750>]
```

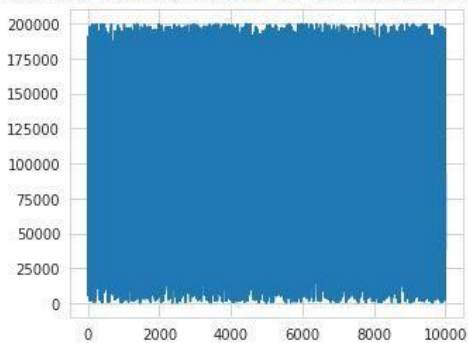


+ Code

+ Text

```
[ ] std_tenure=scale(tenure,axis=0,with_mean=False,with_std=False)
plt.plot(std_tenure)
```

```
[<matplotlib.lines.Line2D at 0x7fed68046b90>]
```



+ Code

+ Text

10. SPLIT THE DATA INTO TRAINING AND TESTING

Question-10:

Split the data into training and testing

10. SPLIT THE DATA INTO TRAINING AND TESTING

```
[ ] import pandas as pd
data=pd.read_csv("Churn_Modelling.csv")
```

```
data.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	10000.000000	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	0.000000	0.000000	11.580000	0.000000
25%	2500.75000	1.562853e+07	584.000000	32.000000	3.000000	0.000000	1.000000	0.000000	0.000000	51002.110000	0.000000
50%	5000.50000	1.569074e+07	652.000000	37.000000	5.000000	97198.540000	1.000000	1.000000	1.000000	100193.915000	0.000000
75%	7500.25000	1.575323e+07	718.000000	44.000000	7.000000	127644.240000	2.000000	1.000000	1.000000	149388.247500	0.000000
max	10000.00000	1.581569e+07	850.000000	92.000000	10.000000	250898.090000	4.000000	1.000000	1.000000	199992.480000	1.000000

```
[ ] import numpy as np
```

```
[ ] x=np.array(data["CustomerId"]).reshape(-1,1)
x.shape
```

```
(10000, 1)
```

[+ Code](#)
[+ Text](#)

```
[ ] y=np.array(data["EstimatedSalary"])
y.shape
```

```
(10000,)
```

```
[ ] print(y)
```

```
[101348.88 112542.58 113931.57 ... 42085.58 92888.52 38190.78]
```

```
[ ] print(type(x))
```

```
<class 'numpy.ndarray'>
```

```
[ ] from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30)
```

```
[ ] x_train.shape
```

```
(7000, 1)
```

```
[ ] x_test.shape
```

```
(3000, 1)
```

```
[ ] y_train.shape
```

```
(7000,)
```

```
[ ] y.shape
```

```
(10000,)
```

```
[ ] print(y_train.shape)
```

```
(7000,)
```

```
[ ] print(y_test.shape)
```

```
(3000,)
```

