Assignment -2

Python Programming

Data Visualization and Pre-processing

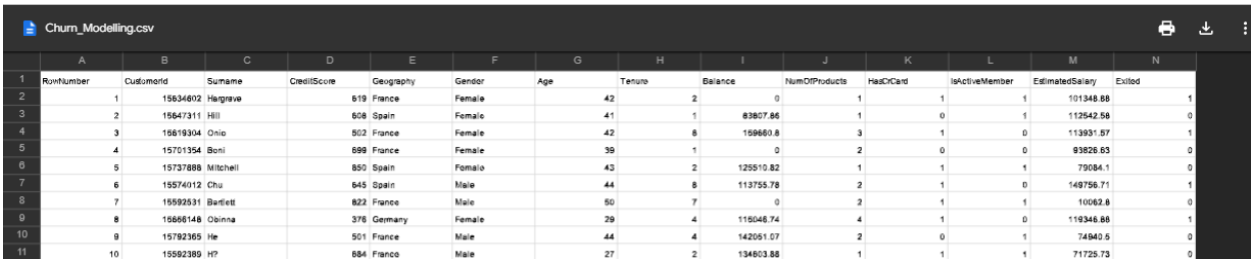
Assignment Date	27 September 2022
Student Name	R.Valliyappan
Student Roll Number	912419104036
Project	AI BASED DISCOURSEFOR BANKING INDUSTRY
Maximum Marks	2 Marks

Question-1:

Download the dataset:

Solution:

Churn_Modelling.csv



Churn_Modelling.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
2	1	15634602	Hargrave	619	France	Female	42	2	0	1	1	1	101348.88	1
3	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
4	3	15619304	Onio	502	France	Female	42	8	159660.8	3	1	0	113931.57	1
5	4	15701354	Boni	699	France	Female	39	1	0	2	0	0	93826.63	0
6	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.1	0
7	6	15574012	Chu	645	Spain	Male	44	8	113755.78	2	1	0	149756.71	1
8	7	15592531	Bartlett	822	France	Male	50	7	0	2	1	1	10062.8	0
9	8	15656148	Obinna	376	Germany	Female	29	4	115046.74	4	1	0	119346.88	1
10	9	15792365	He	501	France	Male	44	4	142051.07	2	0	1	74940.5	0
11	10	15592389	H?	684	France	Male	27	2	134603.88	1	1	1	71725.73	0

Question-2:

Load the dataset.

Solution:

from google.colab import drive
drive.mount('/content/drive').

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

import pandas as pd
import numpy as np
dataset = pd.read_csv("/content/drive/MyDrive/Churn_Modelling.csv")
dataset.head()

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

```
dataset = pd.read_csv("/content/drive/MyDrive/Churn_Modelling.csv")
dataset.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

dataset.shape

```
dataset.shape #size of dataset
```

```
(10000, 14)
```

Question-3:

Perform Below Visualizations.
- **Univariate Analysis**
- **Bi - Variate Analysis**
- **Multi - Variate Analysis**

Solution:

import matplotlib.pyplot as plt
import seaborn as sns

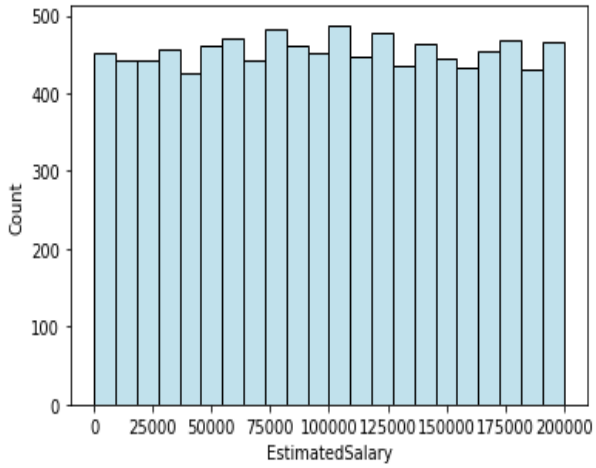
```
import matplotlib.pyplot as plt
import seaborn as sns
```

● Univariate Analysis

sns.histplot(dataset["EstimatedSalary"],color='lightblue')

```
sns.histplot(dataset["EstimatedSalary"],color='lightblue')
```

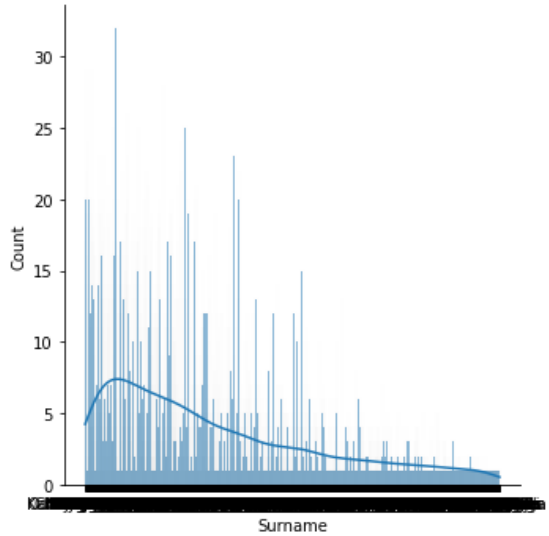
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0b83b82490>
```



sns.displot(dataset['Surname'], kde=True)

```
sns.displot(dataset['Surname'], kde=True)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f0b80893050>
```

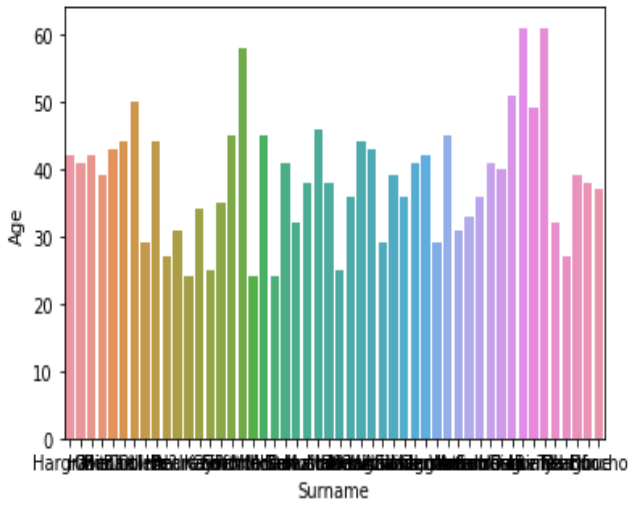


● **Bi - Variate Analysis**

sns.barplot(data=dataset.head(50), x="Surname", y="Age")

```
sns.barplot(data=dataset.head(50), x="Surname", y="Age")
```

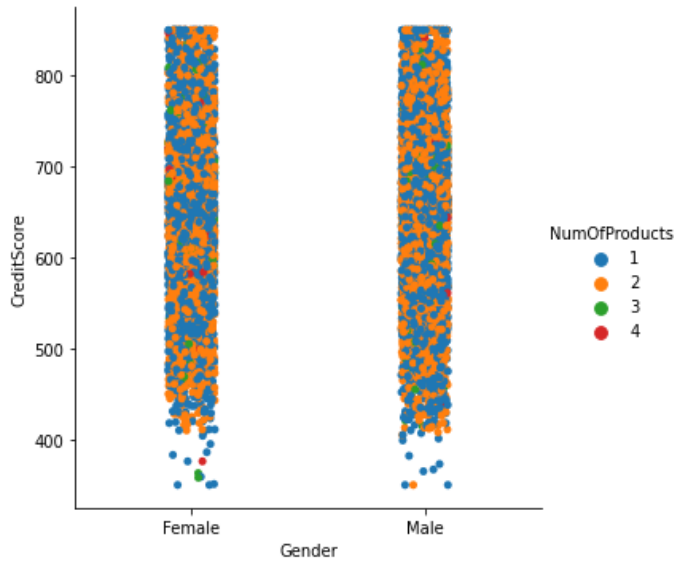
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0b7f3ab2d0>
```



sns.catplot(x='Gender', y='CreditScore', hue='NumOfProducts', data=datase)

```
sns.catplot(x='Gender', y='CreditScore', hue='NumOfProducts', data=dataset)
```

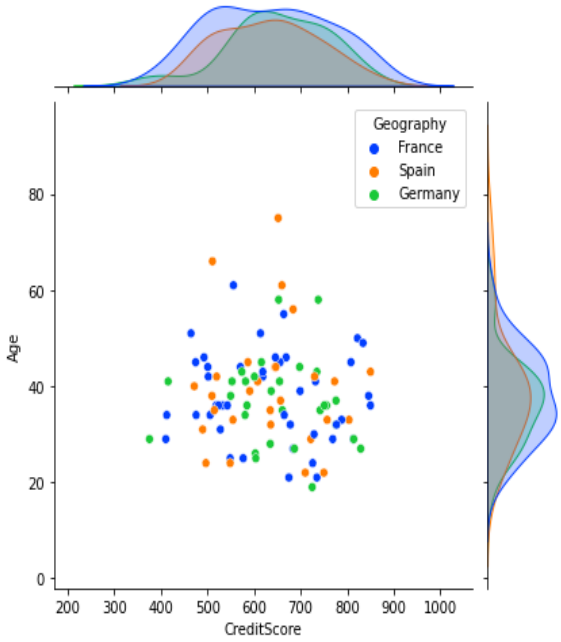
```
<seaborn.axisgrid.FacetGrid at 0x7f0b7cace850>
```



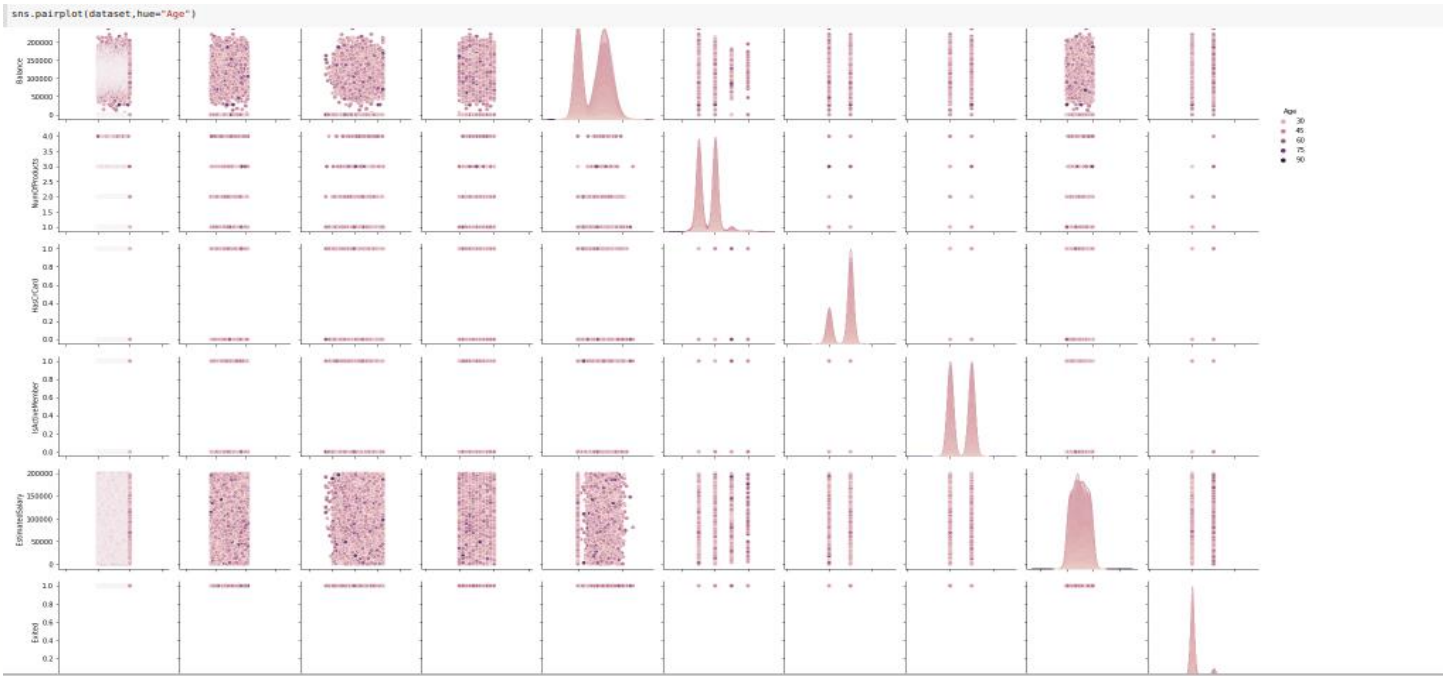
● Multi - Variate Analysis

sns.jointplot(x='CreditScore',y='Age',data=dataset.head(100),
palette='bright',hue='Geography');

```
sns.jointplot(x='CreditScore',y='Age',data=dataset.head(100),palette='bright',hue='Geography');
```



sns.pairplot(dataset,hue="Age")



Question-4:

Perform descriptive statistics on the dataset.

Solution:

dataset.describe()

```
dataset.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

dataset.mean()

```
dataset.mean()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reduction
  """Entry point for launching an IPython kernel.
RowNumber          5.000500e+03
CustomerId         1.569094e+07
CreditScore        6.505288e+02
Age                3.892180e+01
Tenure             5.012800e+00
Balance            7.648589e+04
NumOfProducts      1.530200e+00
HasCrCard          7.055000e-01
IsActiveMember     5.151000e-01
EstimatedSalary    1.000902e+05
Exited             2.037000e-01
dtype: float64
```

dataset.median()

```
dataset.median()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance columns in
  """Entry point for launching an IPython kernel.
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
```

Question-5:

Handle the Missing values.

Solution:

dataset.isnull()

```
dataset.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
9998	False	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	False

10000 rows × 14 columns

dataset.isnull().sum()

```
dataset.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
```

Question- 6

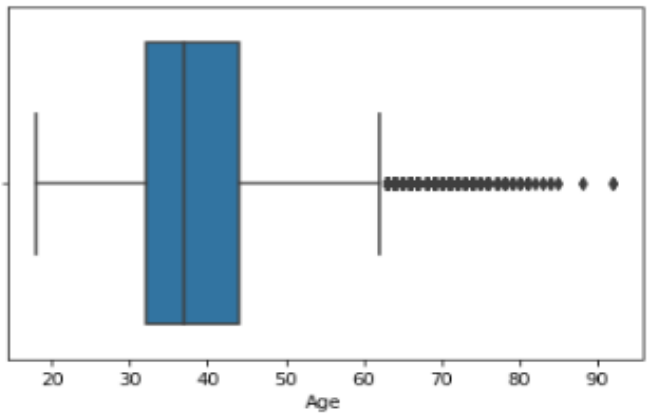
Find the outliers and replace the outliers.

Solution:

sns.boxplot(dataset['Age'])

```
sns.boxplot(dataset['Age'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e9eb10b10>
```



Q1=dataset.Age.quantile(0.25)
Q3=dataset.Age.quantile(0.75)
Q1,Q3

```
Q1=dataset.Age.quantile(0.25)
Q3=dataset.Age.quantile(0.75)
Q1,Q3
```

```
(32.0, 44.0)
```

IQR=Q3-Q1
IQR

```
IQR=Q3-Q1
IQR
```

```
12.0
```

lower_limit=Q1-1.5*IQR
upper_limit=Q3+1.5*IQR
lower_limit,upper_limit

```
lower_limit=Q1-1.5*IQR
upper_limit=Q3+1.5*IQR
lower_limit,upper_limit
```

```
(14.0, 62.0)
```

new_dataset=dataset[(dataset.Age>lower_limit)&(dataset.Age<upper_limit-3)]
new_dataset.head()

```
new_dataset=dataset[(dataset.Age>lower_limit)&(dataset.Age<upper_limit-3)]
new_dataset.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

Question-7

Check for Categorical columns and perform encoding.

Solution:

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
dataset['Geography'] = le.fit_transform(dataset['Geography'])
dataset['Gender']  = le.fit_transform(dataset['Gender'])

dataset.head(10)
```

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
dataset['Geography'] = le.fit_transform(dataset['Geography'])
dataset['Gender']  = le.fit_transform(dataset['Gender'])

dataset.head(10)
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	228	0	0	42	2	0	1	1	1	101348.88	1
1	2	15647311	Hill	217	2	0	41	1	743	1	0	1	112542.58	0
2	3	15619304	Onio	111	0	0	42	8	5793	3	1	0	113931.57	1
3	4	15701354	Boni	308	0	0	39	1	0	2	0	0	93826.63	0
4	5	15737888	Mitchell	459	2	0	43	2	3696	1	1	1	79084.10	0
5	6	15574012	Chu	254	2	1	44	8	2674	2	1	0	149756.71	1
6	7	15592531	Bartlett	431	0	1	50	7	0	2	1	1	10062.80	0
7	8	15656148	Obinna	8	1	0	29	4	2781	4	1	0	119346.88	1
8	9	15792365	He	110	0	1	44	4	4962	2	0	1	74940.50	0
9	10	15592389	H?	293	0	1	27	2	4450	1	1	1	71725.73	0

Question-8
Split the data into dependent and independent variables.

Solution:

```
y = dataset['CreditScore'] #dependent
x = dataset.drop(columns = ['CreditScore'],axis = 1) #independent
x.head()
```

```
y = dataset['CreditScore'] #dependent
x = dataset.drop(columns = ['CreditScore'],axis = 1) #independent
x.head()
```

	RowNumber	CustomerId	Surname	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	0	0	42	2	0	1	1	1	101348.88	1
1	2	15647311	Hill	2	0	41	1	743	1	0	1	112542.58	0
2	3	15619304	Onio	0	0	42	8	5793	3	1	0	113931.57	1
3	4	15701354	Boni	0	0	39	1	0	2	0	0	93826.63	0
4	5	15737888	Mitchell	2	0	43	2	3696	1	1	1	79084.10	0

Question-9
Scale the independent variables

Solution:

```
names = ['RowNumber','CustomerId','Geography','Gender','Age','Tenure','Balance','NumOfProducts','HasCrCard','IsActiveMember','EstimatedSalary','Exited']
```

```
names = ['RowNumber','CustomerId','Geography','Gender','Age','Tenure','Balance','NumOfProducts','HasCrCard','IsActiveMember','EstimatedSalary','Exited']
```

```
from sklearn.preprocessing import scale
x = scale(x[names])
x
```

```
from sklearn.preprocessing import scale

x = scale(x[names])
x
```

```
array([[-1.73187761, -0.78321342, -0.90188624, ...,  0.97024255,
         0.02188649,  1.97716468],
       [-1.7315312 , -0.60653412,  1.51506738, ...,  0.97024255,
         0.21653375, -0.50577476],
       [-1.73118479, -0.99588476, -0.90188624, ..., -1.03067011,
         0.2406869 ,  1.97716468],
       ...,
       [ 1.73118479, -1.47928179, -0.90188624, ...,  0.97024255,
        -1.00864308,  1.97716468],
       [ 1.7315312 , -0.11935577,  0.30659057, ..., -1.03067011,
        -0.12523071,  1.97716468],
       [ 1.73187761, -0.87055909, -0.90188624, ..., -1.03067011,
        -1.07636976, -0.50577476]])
```

```
x = pd.DataFrame(x,columns = names)
x.head()
```

	RowNumber	CustomerId	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	-1.731878	-0.783213	-0.901886	-1.095988	0.293517	-1.041760	-1.225848	-0.911583	0.646092	0.970243	0.021886	1.977165
1	-1.731531	-0.606534	1.515067	-1.095988	0.198164	-1.387538	0.117350	-0.911583	-1.547768	0.970243	0.216534	-0.505775
2	-1.731185	-0.995885	-0.901886	-1.095988	0.293517	1.032908	1.333053	2.527057	0.646092	-1.030670	0.240687	1.977165
3	-1.730838	0.144767	-0.901886	-1.095988	0.007457	-1.387538	-1.225848	0.807737	-1.547768	-1.030670	-0.108918	-0.505775
4	-1.730492	0.652659	1.515067	-1.095988	0.388871	-1.041760	0.785728	-0.911583	0.646092	0.970243	-0.365276	-0.505775

Question-10

Split the data into training and testing

Solution:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 0)
x_train
```

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 0)
```

```
x_train
```

	RowNumber	CustomerId	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
7681	0.928899	-0.797032	-0.901886	0.912419	-0.564665	-1.041760	1.117213	0.807737	0.646092	0.970243	-0.770218	1.977165
9031	1.396553	0.714314	-0.901886	0.912419	0.007457	0.687130	-1.225848	0.807737	0.646092	-1.030670	-1.395767	-0.505775
3691	-0.453278	0.963450	-0.901886	-1.095988	3.535540	-0.004426	1.354191	-0.911583	-1.547768	0.970243	-1.499656	-0.505775
202	-1.661903	-1.250707	1.515067	0.912419	1.056346	-0.004426	-1.225848	-0.911583	-1.547768	0.970243	0.800862	1.977165
5625	0.216680	-0.385174	-0.901886	-1.095988	2.009882	0.687130	1.070229	-0.911583	0.646092	0.970243	0.512497	-0.505775
...	...	...	...	...	...	...	...	...	...	...	...	...
9225	1.463756	-1.473777	0.306591	-1.095988	-0.660018	-0.350204	0.698607	0.807737	0.646092	0.970243	1.093273	-0.505775
4859	-0.048671	-0.609314	1.515067	-1.095988	-1.613554	-0.350204	0.608299	-0.911583	0.646092	0.970243	0.133249	-0.505775
3264	-0.601195	-1.620525	-0.901886	0.912419	-0.373958	-0.004426	1.358909	0.807737	0.646092	-1.030670	1.414415	-0.505775
9845	1.678530	-0.374039	1.515067	-1.095988	-0.087897	1.378686	-1.225848	0.807737	0.646092	0.970243	0.846147	-0.505775
2732	-0.785485	-1.364118	0.306591	-1.095988	0.865639	-1.387538	0.506303	-0.911583	0.646092	-1.030670	0.326305	1.977165

7000 rows × 12 columns

x_test

```
x_test
```

	RowNumber	CustomerId	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
9394	1.522299	-1.045250	0.306591	-1.095988	-0.373958	1.032908	0.875323	-0.911583	0.646092	0.970243	1.613046	-0.505775
898	-1.420801	-0.503813	-0.901886	-1.095988	0.102810	-1.041760	0.424422	-0.911583	0.646092	-1.030670	0.497532	1.977165
2398	-0.901186	-0.793292	1.515067	-1.095988	0.293517	1.032908	0.302927	-0.911583	0.646092	0.970243	-0.423561	-0.505775
5906	0.314021	0.760190	-0.901886	0.912419	-0.660018	-0.350204	0.570464	-0.911583	-1.547768	-1.030670	-0.186439	-0.505775
2343	-0.920239	1.042107	0.306591	0.912419	-0.087897	-0.004426	1.387129	0.807737	0.646092	0.970243	0.618560	-0.505775
...	...	...	...	...	...	...	...	...	...	...	...	...
4004	-0.344851	0.661806	1.515067	-1.095988	0.198164	-0.350204	-1.225848	0.807737	-1.547768	0.970243	0.826264	-0.505775
7375	0.822897	-0.723866	-0.901886	0.912419	3.630893	0.341352	0.071162	0.807737	0.646092	0.970243	-0.769654	-0.505775
9307	1.492162	-0.146464	-0.901886	0.912419	0.102810	-1.041760	1.466728	0.807737	0.646092	0.970243	1.170455	-0.505775
8394	1.175889	-1.292287	-0.901886	0.912419	2.868064	1.724464	1.257616	-0.911583	0.646092	0.970243	-0.508468	-0.505775
5233	0.080887	-1.385388	0.306591	-1.095988	0.960993	-0.350204	0.197777	-0.911583	0.646092	0.970243	-1.153427	1.977165

3000 rows × 12 columns

x_test.shape

```
x_test.shape
```

```
(3000, 12)
```

y_train

```
y_train
```

```
7681     641
9031     541
3691     590
202      516
5625     508
        ...
9225     594
4859     794
3264     738
9845     590
2732     623
Name: CreditScore, Length: 7000, dtype: int64
```

y_test

```
y_test
```

```
9394     597
898      523
2398     706
5906     788
2343     706
        ...
4004     530
7375     639
9307     685
8394     692
5233     731
Name: CreditScore, Length: 3000, dtype: int64
```

y_test.shape

```
y_test.shape
```

```
(3000,)
```