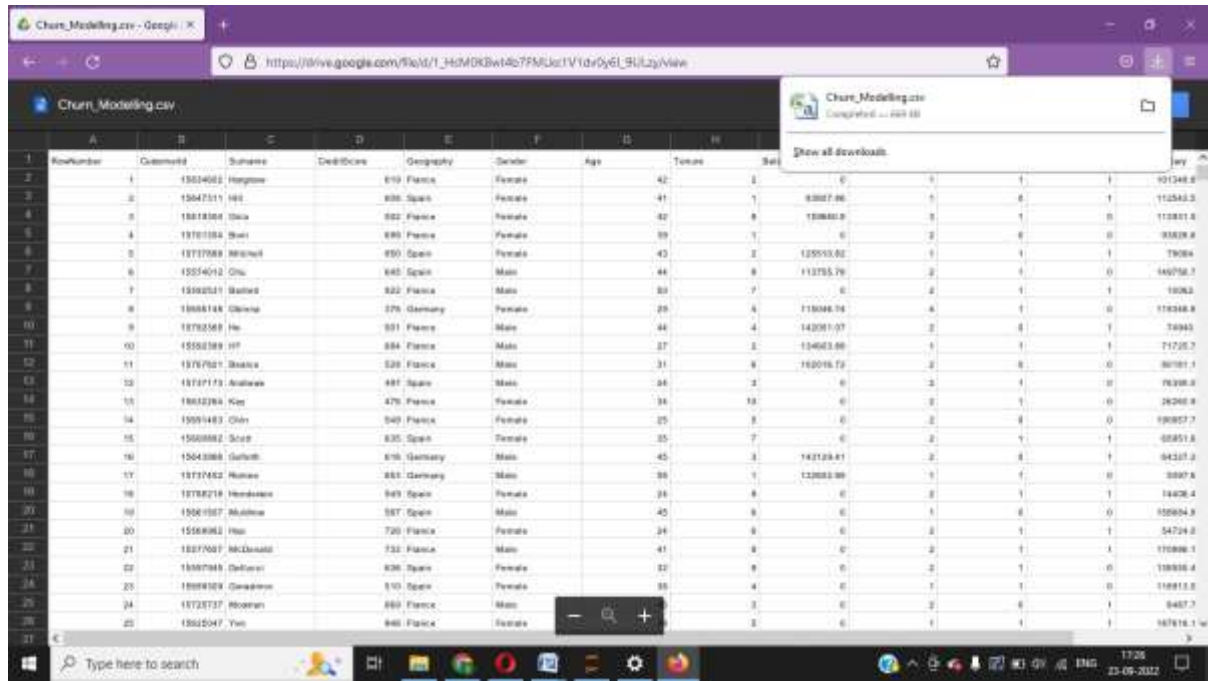


Data Visualization and Data Pre-processing

Question-1:

Download the dataset: Dataset



	A	B	C	D	E	F	G	H	I	J
	RowNumber	CustomerID	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	
1	1	15634602	Hargrove	619	France	Female	42	3	0	101348.6
2	2	15647511	Hill	606	Spain	Female	41	1	83827.86	112543.3
3	3	15681868	Isla	602	France	Female	40	8	109840.8	113801.8
4	4	15701084	Boat	695	France	Female	59	1	0	91618.8
5	5	15717689	McNeill	650	Spain	Female	43	2	120513.82	79084
6	6	15574012	Ono	645	Spain	Male	44	8	112755.76	146758.7
7	7	15682521	Battist	622	France	Male	39	7	0	18062
8	8	15688148	Glavin	576	Germany	Female	29	4	119386.76	118348.8
9	9	15782569	He	591	France	Male	44	4	142591.07	78940
10	10	15582989	HT	694	France	Male	37	2	134643.86	71725.7
11	11	15767681	Beckus	528	France	Male	31	8	162016.72	86781.1
12	12	15787173	Arakawa	497	Spain	Male	24	2	0	76398.8
13	13	15632364	Kay	476	France	Female	38	18	0	26246.8
14	14	15691483	Chen	540	France	Female	25	2	0	190827.7
15	15	15688882	Shah	635	Spain	Female	35	7	0	55851.8
16	16	15648884	Gutjahr	616	Germany	Male	45	2	142129.41	94327.2
17	17	15737452	Holzer	651	Germany	Male	58	1	122683.86	3897.8
18	18	15738219	Henderson	649	Spain	Female	38	8	0	14408.4
19	19	15661987	Mullins	587	Spain	Male	45	8	0	108604.8
20	20	15688982	Hsu	726	France	Female	28	8	0	54724.8
21	21	15277687	McDonald	732	France	Male	41	8	0	110886.1
22	22	15687848	Delacruz	636	Spain	Female	27	8	0	108855.4
23	23	15688829	Gonzalez	610	Spain	Female	55	4	0	118813.8
24	24	15725737	Roosman	689	France	Male	3	2	0	8487.7
25	25	15625047	Yeo	646	France	Female	3	0	1	167816.1

Question-2:

Load the dataset.

Solution:

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

data = pd.read_csv("E://Churn_Modelling.csv")
data.head()
```

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
data = pd.read_csv("E://Churn_Modelling.csv")
data.head()
```

```
Out[1]:
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary
0	1	15634602	Hargrave	819	France	Female	42	2	0.00	1	1	1	101348.88
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58
2	3	15619304	Oniz	502	France	Female	42	8	159660.80	3	1	0	113931.57
3	4	15701354	Boni	689	France	Female	39	1	0.00	2	0	0	83826.63
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10

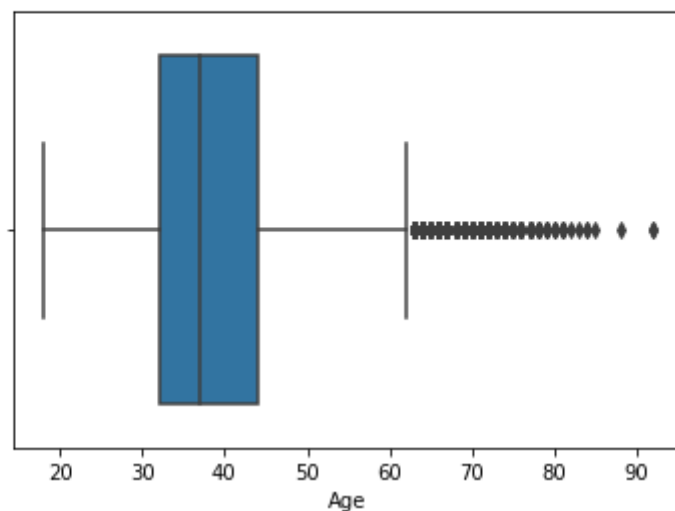
Question-3:

Perform Below Visualizations.

- Univariate Analysis

```
In [3]: sns.boxplot(data.Age)
```

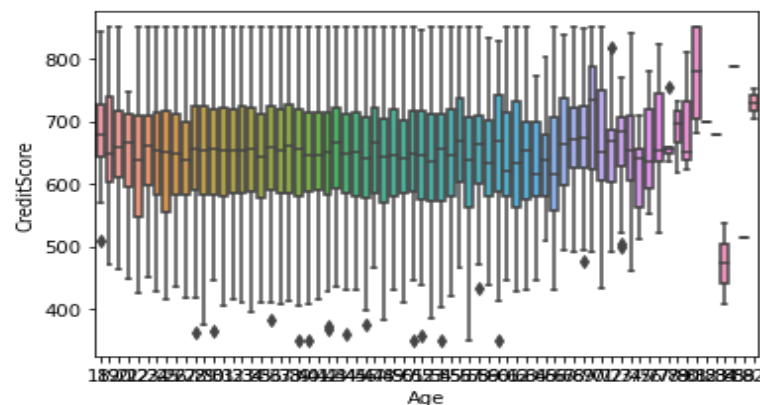
```
Out[3]: <AxesSubplot:xlabel='Age'>
```



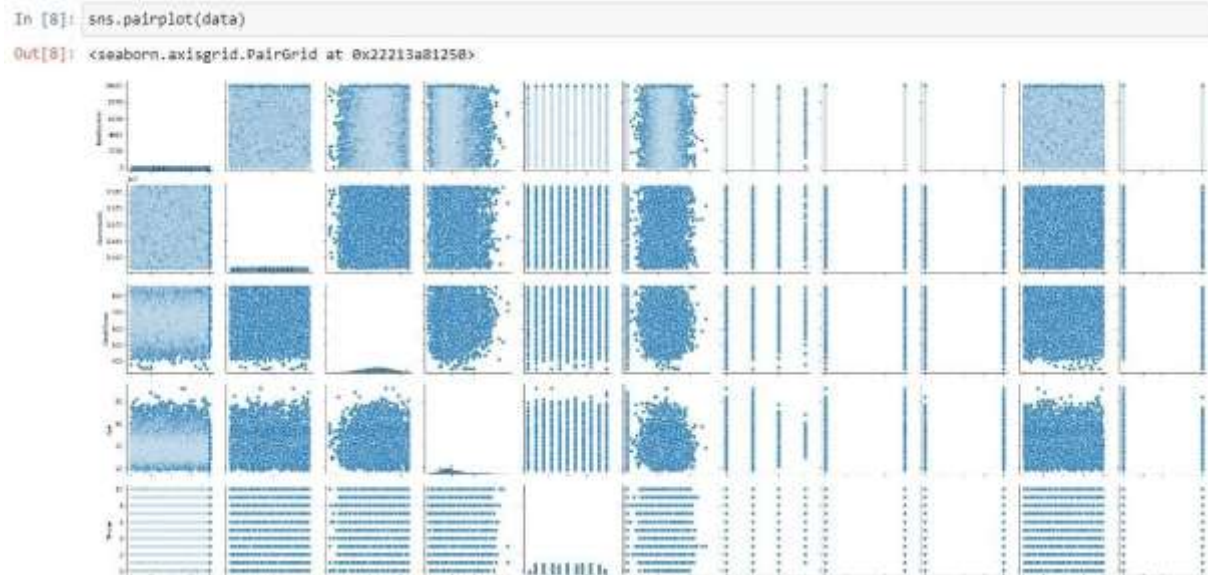
- Bi - Variate Analysis

```
In [7]: sns.boxplot(y=data.CreditScore,x=data.Age)
```

```
Out[7]: <AxesSubplot:xlabel='Age', ylabel='CreditScore'>
```



- Multi - Variate Analysis



Question-4:

Perform descriptive statistics on the dataset.

```
In [9]: data['NumOfProducts'].mean()
```

```
Out[9]: 1.5302
```

```
In [10]: data['EstimatedSalary'].median()
```

```
Out[10]: 100193.915
```

```
In [11]: data['Tenure'].mode()
```

```
Out[11]: 0    2
dtype: int64
```

```
In [13]: data.kurt()
```

```
Out[13]: RowNumber      -1.200000
CustomerId    -1.196113
CreditScore   -0.425726
Age           1.395347
Tenure        -1.165225
Balance       -1.489412
NumOfProducts  0.582981
HasCrCard     -1.186973
IsActiveMember -1.996747
EstimatedSalary -1.181518
Exited        0.165671
dtype: float64
```

```
In [16]: data.var()
```

```
Out[16]: RowNumber      8.334167e+06  
CustomerId    5.174815e+09  
CreditScore   9.341860e+03  
Age           1.099941e+02  
Tenure        8.364673e+00  
Balance       3.893436e+09  
NumOfProducts 3.383218e-01  
HasCrCard     2.077905e-01  
IsActiveMember 2.497970e-01  
EstimatedSalary 3.307457e+09  
Exited        1.622225e-01  
dtype: float64
```

```
In [17]: data.std()
```

```
Out[17]: 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
```

Question-5:

Handle the Missing values.

```
In [18]: data.isna().any()
```

```
Out[18]: 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
```

```
In [19]: data.isna().sum()
```

```
Out[19]: 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
```

```
In [22]: data['EstimatedSalary'].fillna(data['EstimatedSalary'].mean(),inplace=True)
data
```

```
Out[22]:
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	10134
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	11254
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	11393
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	9382
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	7908
...
9995	9996	15606229	Obijaku	771	France	Male	39	5	0.00	2	1	0	9627
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	10169
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	4208
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	9288
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	3819

10000 rows x 14 columns

Question-6:

Find the outliers and replace the outliers

```
In [23]: Q1=data.Age.quantile(0.25)
Q2=data.Age.quantile(0.75)
IQR=Q2-Q1
print(IQR)
```

```
12.0
```

```
In [24]: data=data[~((data.Age<(Q1-1.5*IQR))|(data.Age>(Q2+1.5*IQR)))]
data
```

```
Out[24]:
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	10134
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	11254
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	11393
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	9382
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	7908
...
9995	9996	15606229	Obijaku	771	France	Male	39	5	0.00	2	1	0	9627
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	10169
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	4208
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	9288
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	3819

9641 rows x 14 columns

Question-7:

Check for Categorical columns and perform encoding.

```
In [28]: data['Exited'].replace({1:'Yes',0:'No'},inplace=True)
data.head()
```

```
Out[28]:
```

Number	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	Yes
2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	No
3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	Yes
4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	No
5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	No

Question-8:

Split the data into dependent and independent variables.

```
In [34]: dmain= pd.get_dummies(data,columns=['Gender'])
dmain
```

```
Out[34]:
```

Surname	CreditScore	Geography	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Gender_Female	Gender_Male
Hargrave	619	France	42	2	0.00	1	1	1	101348.88	Yes	1	0
Hill	608	Spain	41	1	83807.86	1	0	1	112542.58	No	1	0
Onio	502	France	42	8	159660.80	3	1	0	113931.57	Yes	1	0
Boni	699	France	39	1	0.00	2	0	0	93826.63	No	1	0
Mitchell	850	Spain	43	2	125510.82	1	1	1	79084.10	No	1	0
...
Obijaku	771	France	39	5	0.00	2	1	0	98270.64	No	0	1
Johnstone	516	France	35	10	57369.61	1	1	1	101699.77	No	0	1
Liu	709	France	36	7	0.00	1	0	1	42085.58	Yes	1	0
Sebbastini	772	Germany	42	3	75075.31	2	1	0	82888.52	Yes	0	1
Walker	792	France	28	4	130142.79	1	1	0	38190.78	No	1	0

```
In [35]: y = dmain['HasCrCard']
y
```

```
Out[35]:
```

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

Name: HasCrCard, Length: 10000, dtype: int64

```
In [36]: x = dmain.drop(columns='HasCrCard',axis=1)
x.head()
```

```
Out[36]:
```

RowNumber	CustomerId	Surname	CreditScore	Geography	Age	Tenure	Balance	NumOfProducts	IsActiveMember	EstimatedSalary	Exited	Gender_Fem
0	1	15634602	Hargrave	619	France	42	2	0.00	1	1	101348.88	Yes
1	2	15647311	Hill	608	Spain	41	1	83807.86	1	1	112542.58	No
2	3	15619304	Onio	502	France	42	8	159660.80	3	0	113931.57	Yes
3	4	15701354	Boni	699	France	39	1	0.00	2	0	93826.63	No
4	5	15737888	Mitchell	850	Spain	43	2	125510.82	1	1	79084.10	No

Question-9:

Scale the independent variables

```
In [37]: x=data.iloc[:,6:7].values
         from sklearn.preprocessing import StandardScaler
         std=StandardScaler()
         x=std.fit_transform(x)
         x
```

```
Out[37]: array([[ 0.29351742],
                [ 0.19816383],
                [ 0.29351742],
                ...,
                [-0.27860412],
                [ 0.29351742],
                [-1.04143285]])
```

Question-10:

Split the data into training and testing

```
In [56]: 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)
```

```
In [57]: x_train
```

```
Out[57]: array([[ -2.24837781],
                [ 0.59167031],
                [ 1.04607801],
                ...,
                [-0.54434894],
                [ 1.04607801],
                [-0.43074701]])
```

```
In [58]: x_test
```

```
Out[58]: array([[ 1.50048571],
                [-0.20354316],
                [ 0.36446646],
                ...,
                [ 0.81887416],
                [-0.88515471],
                [ 0.13726261]])
```

```
In [40]: y_train
```

```
Out[40]: 7389    1
          9275    1
          2995    1
          5316    1
          356     1
          ..
          9225    1
          4859    1
          3264    1
          9845    1
          2732    1
          Name: HasCrCard, Length: 8000, dtype: int64
```

```
In [41]: y_test
```

```
Out[41]: 9394    1
          898     1
          2398    1
          5906    0
          2343    1
          ..
          1037    1
          2899    1
          9549    1
          2740    1
          6690    1
          Name: HasCrCard, Length: 2000, dtype: int64
```