In [62]:
```
import pandas as pd
import numpy as np
```

In [63]:
```
df=pd.read_csv("../../../Bigdata Files/Credit Score Classification Dataset.csv")
df
```

Out[63]:

	Age	Gender	Income	Education	Marital Status	Number of Children	Home Ownership	Credit Score
0	25	Female	50000.0	Bachelor's Degree	Single	0	Rented	High
1	30	Male	100000.0	Master's Degree	Married	2	Owned	High
2	35	Female	75000.0	Doctorate	Married	1	Owned	High
3	40	Male	125000.0	High School Diploma	Single	0	Owned	High
4	45	Female	100000.0	Bachelor's Degree	Married	3	Owned	High
...	...	...	...	...	...	...	...	...
159	29	Female	27500.0	High School Diploma	Single	0	Rented	Low
160	34	Male	47500.0	Associate's Degree	Single	0	Rented	Average
161	39	Female	62500.0	Bachelor's Degree	Married	2	Owned	High
162	44	Male	87500.0	Master's Degree	Single	0	Owned	High
163	49	Female	77500.0	Doctorate	Married	1	Owned	High

164 rows × 8 columns

In [64]:
```
df.head()
```

Out[64]:

	Age	Gender	Income	Education	Marital Status	Number of Children	Home Ownership	Credit Score
0	25	Female	50000.0	Bachelor's Degree	Single	0	Rented	High
1	30	Male	100000.0	Master's Degree	Married	2	Owned	High
2	35	Female	75000.0	Doctorate	Married	1	Owned	High
3	40	Male	125000.0	High School Diploma	Single	0	Owned	High
4	45	Female	100000.0	Bachelor's Degree	Married	3	Owned	High

In [65]:
```
df.tail()
```

Out[65]:

	Age	Gender	Income	Education	Marital Status	Number of Children	Home Ownership	Credit Score
159	29	Female	27500.0	High School Diploma	Single	0	Rented	Low
160	34	Male	47500.0	Associate's Degree	Single	0	Rented	Average
161	39	Female	62500.0	Bachelor's Degree	Married	2	Owned	High
162	44	Male	87500.0	Master's Degree	Single	0	Owned	High
163	49	Female	77500.0	Doctorate	Married	1	Owned	High

In [66]:
```
df.describe()
```

Out[66]:

	Age	Income	Number of Children
count	164.000000	155.000000	164.000000
mean	37.975610	83951.612903	0.652439
std	8.477289	32187.194669	0.883346
min	25.000000	25000.000000	0.000000
25%	30.750000	58750.000000	0.000000
50%	37.000000	82500.000000	0.000000
75%	45.000000	105000.000000	1.000000
max	53.000000	162500.000000	3.000000

In [67]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 164 entries, 0 to 163
Data columns (total 8 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Age                 164 non-null    int64  
 1   Gender              158 non-null    object 
 2   Income              155 non-null    float64
 3   Education           164 non-null    object 
 4   Marital Status      164 non-null    object 
 5   Number of Children  164 non-null    int64  
 6   Home Ownership      164 non-null    object 
 7   Credit Score        164 non-null    object 
dtypes: float64(1), int64(2), object(5)
memory usage: 10.4+ KB
```

In [68]:
```
df.isna().sum()
```

Out[68]:
```
Age                   0
Gender                6
Income                9
Education             0
Marital Status        0
Number of Children    0
Home Ownership        0
Credit Score          0
dtype: int64
```

In [69]:
```
df["Gender"].fillna(df["Gender"].mode()[0],inplace=True)
```

In [70]:
```
df["Gender"].mode()[0]
```

Out[70]:
```
'Female'
```

In [71]:
```
df["Income"].mean()
```

Out[71]:
```
83951.6129032258
```

In [72]:
```
df["Income"].fillna(df["Income"].mean(),inplace=True)
```

In [73]:
```
df.isna().sum()
```

Out[73]:
```
Age                   0
Gender                0
```

```
Income                0
Education             0
Marital Status        0
Number of Children    0
Home Ownership        0
Credit Score          0
dtype: int64
```

In [74]:
```
from sklearn.preprocessing import LabelEncoder
```

In [75]:
```
encoder=LabelEncoder()
df["Gender"]=encoder.fit_transform(df["Gender"])
df["Education"]=encoder.fit_transform(df["Education"])
df["Marital Status"]=encoder.fit_transform(df["Marital Status"])
df["Home Ownership"]=encoder.fit_transform(df["Home Ownership"])
```

In [76]:
```
df.head()
```

Out[76]:

	Age	Gender	Income	Education	Marital Status	Number of Children	Home Ownership	Credit Score
0	25	0	50000.0	1	1	0	1	High
1	30	1	100000.0	4	0	2	0	High
2	35	0	75000.0	2	0	1	0	High
3	40	1	125000.0	3	1	0	0	High
4	45	0	100000.0	1	0	3	0	High

In [77]:
```
df.isna().sum()
```

Out[77]:
```
Age                   0
Gender                0
Income                0
Education             0
Marital Status        0
Number of Children    0
Home Ownership        0
Credit Score          0
dtype: int64
```

In [78]:
```
X=df.drop(["Gender","Credit Score"],axis=1).values
```

In [79]:
```
X
```

Out[79]:
```
array([[2.50000000e+01, 5.00000000e+04, 1.00000000e+00, 1.00000000e+00,
        0.00000000e+00, 1.00000000e+00],
       [3.00000000e+01, 1.00000000e+05, 4.00000000e+00, 0.00000000e+00,
        2.00000000e+00, 0.00000000e+00],
       [3.50000000e+01, 7.50000000e+04, 2.00000000e+00, 0.00000000e+00,
        1.00000000e+00, 0.00000000e+00],
       [4.00000000e+01, 1.25000000e+05, 3.00000000e+00, 1.00000000e+00,
        0.00000000e+00, 0.00000000e+00],
       [4.50000000e+01, 1.00000000e+05, 1.00000000e+00, 0.00000000e+00,
        3.00000000e+00, 0.00000000e+00],
       [5.00000000e+01, 1.50000000e+05, 4.00000000e+00, 0.00000000e+00,
        0.00000000e+00, 0.00000000e+00],
       [2.60000000e+01, 4.00000000e+04, 0.00000000e+00, 1.00000000e+00,
        0.00000000e+00, 1.00000000e+00],
       [3.10000000e+01, 6.00000000e+04, 1.00000000e+00, 1.00000000e+00,
        0.00000000e+00, 1.00000000e+00],
       [3.60000000e+01, 8.00000000e+04, 4.00000000e+00, 0.00000000e+00,
        2.00000000e+00, 0.00000000e+00],
       [4.10000000e+01, 1.05000000e+05, 2.00000000e+00, 1.00000000e+00,
```

```
 0.00000000e+00, 0.00000000e+00],
[4.60000000e+01, 9.00000000e+04, 3.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[5.10000000e+01, 1.35000000e+05, 1.00000000e+00, 0.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[2.70000000e+01, 3.50000000e+04, 3.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.20000000e+01, 5.50000000e+04, 0.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.70000000e+01, 7.00000000e+04, 1.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[4.20000000e+01, 9.50000000e+04, 4.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[4.70000000e+01, 8.50000000e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[5.20000000e+01, 1.25000000e+05, 3.00000000e+00, 0.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[2.80000000e+01, 3.00000000e+04, 0.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.30000000e+01, 5.00000000e+04, 3.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.80000000e+01, 6.50000000e+04, 1.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[4.30000000e+01, 8.00000000e+04, 4.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[4.80000000e+01, 7.00000000e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[5.30000000e+01, 1.15000000e+05, 0.00000000e+00, 0.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[2.90000000e+01, 2.50000000e+04, 3.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.40000000e+01, 4.50000000e+04, 0.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.90000000e+01, 6.00000000e+04, 1.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[4.40000000e+01, 7.50000000e+04, 4.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[4.90000000e+01, 6.50000000e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[2.50000000e+01, 5.50000000e+04, 1.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.00000000e+01, 1.05000000e+05, 4.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[3.50000000e+01, 8.00000000e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[4.00000000e+01, 1.30000000e+05, 3.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[4.50000000e+01, 1.05000000e+05, 1.00000000e+00, 0.00000000e+00,
 3.00000000e+00, 0.00000000e+00],
[5.00000000e+01, 1.55000000e+05, 4.00000000e+00, 0.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[2.60000000e+01, 4.50000000e+04, 0.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.10000000e+01, 6.50000000e+04, 1.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.60000000e+01, 8.50000000e+04, 4.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[4.10000000e+01, 1.10000000e+05, 2.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[4.60000000e+01, 9.50000000e+04, 3.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[5.10000000e+01, 1.40000000e+05, 1.00000000e+00, 0.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[2.70000000e+01, 3.75000000e+04, 3.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.20000000e+01, 5.75000000e+04, 0.00000000e+00, 1.00000000e+00,
```

```
 0.00000000e+00, 1.00000000e+00],
[3.70000000e+01, 7.25000000e+04, 1.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[4.20000000e+01, 1.00000000e+05, 4.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[4.70000000e+01, 9.00000000e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[5.20000000e+01, 1.30000000e+05, 3.00000000e+00, 0.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[2.80000000e+01, 3.25000000e+04, 0.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.30000000e+01, 5.25000000e+04, 3.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.80000000e+01, 6.75000000e+04, 1.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[4.30000000e+01, 9.25000000e+04, 4.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[4.80000000e+01, 8.25000000e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[5.30000000e+01, 1.22500000e+05, 0.00000000e+00, 0.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[2.90000000e+01, 2.75000000e+04, 3.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.40000000e+01, 4.75000000e+04, 0.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.90000000e+01, 6.25000000e+04, 1.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[4.40000000e+01, 8.75000000e+04, 4.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[4.90000000e+01, 7.75000000e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[2.50000000e+01, 5.75000000e+04, 1.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.00000000e+01, 1.12500000e+05, 4.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[3.50000000e+01, 8.50000000e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[2.50000000e+01, 6.00000000e+04, 1.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.00000000e+01, 1.17500000e+05, 4.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[3.50000000e+01, 9.00000000e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[4.00000000e+01, 1.42500000e+05, 3.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[4.50000000e+01, 1.10000000e+05, 1.00000000e+00, 0.00000000e+00,
 3.00000000e+00, 0.00000000e+00],
[5.00000000e+01, 1.60000000e+05, 4.00000000e+00, 0.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[2.60000000e+01, 4.75000000e+04, 0.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.10000000e+01, 6.75000000e+04, 1.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.60000000e+01, 9.00000000e+04, 4.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[4.10000000e+01, 1.15000000e+05, 2.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[4.60000000e+01, 9.75000000e+04, 3.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[5.10000000e+01, 1.45000000e+05, 1.00000000e+00, 0.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[2.70000000e+01, 3.75000000e+04, 3.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.20000000e+01, 5.75000000e+04, 0.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.70000000e+01, 7.50000000e+04, 1.00000000e+00, 0.00000000e+00,
```

 2.00000000e+00, 0.00000000e+00],
[4.20000000e+01, 1.05000000e+05, 4.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[4.70000000e+01, 9.50000000e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[5.20000000e+01, 1.35000000e+05, 3.00000000e+00, 0.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[2.80000000e+01, 3.25000000e+04, 0.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.30000000e+01, 5.25000000e+04, 3.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.80000000e+01, 6.75000000e+04, 1.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[4.30000000e+01, 9.25000000e+04, 4.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[4.80000000e+01, 8.50000000e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[5.30000000e+01, 1.25000000e+05, 0.00000000e+00, 0.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[2.90000000e+01, 2.75000000e+04, 3.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.40000000e+01, 4.75000000e+04, 0.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.90000000e+01, 6.25000000e+04, 1.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[4.40000000e+01, 8.75000000e+04, 4.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[4.90000000e+01, 7.75000000e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[2.50000000e+01, 5.75000000e+04, 1.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.00000000e+01, 1.12500000e+05, 4.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[3.50000000e+01, 8.50000000e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[2.50000000e+01, 6.25000000e+04, 1.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.00000000e+01, 1.17500000e+05, 4.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[3.50000000e+01, 9.00000000e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[4.00000000e+01, 1.42500000e+05, 3.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[4.50000000e+01, 1.15000000e+05, 1.00000000e+00, 0.00000000e+00,
 3.00000000e+00, 0.00000000e+00],
[5.00000000e+01, 1.62500000e+05, 4.00000000e+00, 0.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[2.60000000e+01, 5.00000000e+04, 0.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.10000000e+01, 7.00000000e+04, 1.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.60000000e+01, 9.50000000e+04, 4.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[4.10000000e+01, 1.20000000e+05, 2.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[4.60000000e+01, 1.02500000e+05, 3.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[5.10000000e+01, 1.50000000e+05, 1.00000000e+00, 0.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[2.70000000e+01, 3.75000000e+04, 3.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.20000000e+01, 5.75000000e+04, 0.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.70000000e+01, 7.75000000e+04, 1.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[4.20000000e+01, 1.10000000e+05, 4.00000000e+00, 1.00000000e+00,

```
 0.00000000e+00, 0.00000000e+00],
[4.70000000e+01, 8.39516129e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[5.20000000e+01, 8.39516129e+04, 3.00000000e+00, 0.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[2.80000000e+01, 8.39516129e+04, 0.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.30000000e+01, 8.39516129e+04, 3.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.80000000e+01, 8.39516129e+04, 1.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[4.30000000e+01, 8.39516129e+04, 4.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[4.80000000e+01, 8.39516129e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[5.30000000e+01, 8.39516129e+04, 0.00000000e+00, 0.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[2.90000000e+01, 8.39516129e+04, 3.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.40000000e+01, 4.75000000e+04, 0.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.90000000e+01, 6.25000000e+04, 1.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[4.40000000e+01, 8.75000000e+04, 4.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[4.90000000e+01, 7.75000000e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[2.50000000e+01, 5.75000000e+04, 1.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.00000000e+01, 1.12500000e+05, 4.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[3.50000000e+01, 8.50000000e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[2.50000000e+01, 6.00000000e+04, 1.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.00000000e+01, 1.17500000e+05, 4.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[3.50000000e+01, 9.00000000e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[2.80000000e+01, 7.50000000e+04, 1.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.30000000e+01, 8.20000000e+04, 4.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[3.10000000e+01, 9.50000000e+04, 2.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[2.60000000e+01, 5.50000000e+04, 1.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[3.20000000e+01, 8.50000000e+04, 4.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[2.90000000e+01, 6.80000000e+04, 2.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[3.40000000e+01, 1.05000000e+05, 1.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 1.00000000e+00],
[2.50000000e+01, 5.50000000e+04, 1.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.00000000e+01, 1.05000000e+05, 4.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[3.50000000e+01, 8.00000000e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[4.00000000e+01, 1.30000000e+05, 3.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[4.50000000e+01, 1.05000000e+05, 1.00000000e+00, 0.00000000e+00,
 3.00000000e+00, 0.00000000e+00],
[5.00000000e+01, 1.55000000e+05, 4.00000000e+00, 0.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[2.60000000e+01, 4.50000000e+04, 0.00000000e+00, 1.00000000e+00,
```

```
 0.00000000e+00, 1.00000000e+00],
[3.10000000e+01, 6.50000000e+04, 1.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.60000000e+01, 8.50000000e+04, 4.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[4.10000000e+01, 1.10000000e+05, 2.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[4.60000000e+01, 9.50000000e+04, 3.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[5.10000000e+01, 1.40000000e+05, 1.00000000e+00, 0.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[2.70000000e+01, 3.75000000e+04, 3.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.20000000e+01, 5.75000000e+04, 0.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.70000000e+01, 7.25000000e+04, 1.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[4.20000000e+01, 1.00000000e+05, 4.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[4.70000000e+01, 9.00000000e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[5.20000000e+01, 1.30000000e+05, 3.00000000e+00, 0.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[2.80000000e+01, 3.25000000e+04, 0.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.30000000e+01, 5.25000000e+04, 3.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.80000000e+01, 6.75000000e+04, 1.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[4.30000000e+01, 9.25000000e+04, 4.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[4.80000000e+01, 8.25000000e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00],
[5.30000000e+01, 1.22500000e+05, 0.00000000e+00, 0.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[2.90000000e+01, 2.75000000e+04, 3.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.40000000e+01, 4.75000000e+04, 0.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 1.00000000e+00],
[3.90000000e+01, 6.25000000e+04, 1.00000000e+00, 0.00000000e+00,
 2.00000000e+00, 0.00000000e+00],
[4.40000000e+01, 8.75000000e+04, 4.00000000e+00, 1.00000000e+00,
 0.00000000e+00, 0.00000000e+00],
[4.90000000e+01, 7.75000000e+04, 2.00000000e+00, 0.00000000e+00,
 1.00000000e+00, 0.00000000e+00]])
```

In [80]:
```
y=df["Credit Score"].values
```

In [81]:
```
y
```

Out[81]:
```
array(['High', 'High', 'High', 'High', 'High', 'High', 'Average',
       'Average', 'High', 'High', 'High', 'High', 'Low', 'Average',
       'High', 'High', 'High', 'High', 'Low', 'Average', 'High', 'High',
       'High', 'High', 'Low', 'Average', 'High', 'High', 'High',
       'Average', 'High', 'High', 'High', 'High', 'High', 'Average',
       'Average', 'High', 'High', 'High', 'High', 'Low', 'Average',
       'High', 'High', 'High', 'High', 'Low', 'Average', 'High', 'High',
       'High', 'High', 'Low', 'Average', 'High', 'High', 'High',
       'Average', 'High', 'High', 'Average', 'High', 'High', 'High',
       'High', 'High', 'Average', 'Average', 'High', 'High', 'High',
       'High', 'Low', 'Average', 'High', 'High', 'High', 'High', 'Low',
       'Average', 'High', 'High', 'High', 'High', 'Low', 'Average',
       'High', 'High', 'High', 'Average', 'High', 'High', 'Average',
       'High', 'High', 'High', 'High', 'High', 'Average', 'Average',
       'High', 'High', 'High', 'High', 'Low', 'Average', 'High', 'High',
```

```
       'High', 'High', 'Low', 'Average', 'High', 'High', 'High', 'High',
       'Low', 'Average', 'High', 'High', 'High', 'Average', 'High',
       'High', 'Average', 'High', 'High', 'Average', 'High', 'High',
       'Average', 'High', 'Average', 'High', 'Average', 'High', 'High',
       'High', 'High', 'High', 'Average', 'Average', 'High', 'High',
       'High', 'High', 'Low', 'Average', 'High', 'High', 'High', 'High',
       'Low', 'Average', 'High', 'High', 'High', 'High', 'Low', 'Average',
       'High', 'High', 'High'], dtype=object)
```

In [82]:
```
from sklearn.model_selection import train_test_split
```

In [83]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=4
```

In [84]:
```
from sklearn.preprocessing import StandardScaler
```

In [85]:
```
scaler=StandardScaler()
scaler.fit(X_train)
X_train=scaler.transform(X_train)
X_test=scaler.transform(X_test)
```

In [86]:
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
```

In [87]:
```
knn=KNeighborsClassifier(n_neighbors=9)
nb=GaussianNB()
sv=SVC()
lst_models=[knn,nb,sv]
```

In [88]:
```
from sklearn.metrics import classification_report
for i in lst_models:
    i.fit(X_train,y_train)
    y_pred=i.predict(X_test)
    print("*"*100)
    print(i)
    print("*"*100)
    print(classification_report(y_test,y_pred))
```

```
****************************************************************************************************
************
KNeighborsClassifier(n_neighbors=9)
****************************************************************************************************
************
              precision    recall  f1-score   support

     Average       0.67      0.73      0.70        11
        High       0.94      0.97      0.95        32
         Low       0.80      0.57      0.67         7

    accuracy                           0.86        50
   macro avg       0.80      0.76      0.77        50
weighted avg       0.86      0.86      0.86        50

****************************************************************************************************
************
GaussianNB()
****************************************************************************************************
************
              precision    recall  f1-score   support
```

```
              precision    recall  f1-score   support

     Average       0.90      0.82      0.86        11
        High       0.94      0.97      0.95        32
         Low       1.00      1.00      1.00         7

    accuracy                           0.94        50
   macro avg       0.95      0.93      0.94        50
weighted avg       0.94      0.94      0.94        50

**************************************************************************************************
************
SVC()
**************************************************************************************************
************
              precision    recall  f1-score   support

     Average       0.69      0.82      0.75        11
        High       0.94      0.97      0.95        32
         Low       1.00      0.57      0.73         7

    accuracy                           0.88        50
   macro avg       0.88      0.79      0.81        50
weighted avg       0.89      0.88      0.88        50
```

In [ ]: