```
import numpy as np
import pandas as pd

df=pd.read_csv('/content/Credit Score Classification Dataset.csv')
df.head()
```

	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

```
df.tail()
```

	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
				Bachelor's				

```
df.drop_duplicates(inplace=True)
```

```
df.drop('Gender',axis=1,inplace=True)
```

```
df.isna().sum()
```

```
Age                  0
Income               9
Education            0
Marital Status       0
Number of Children   0
Home Ownership       0
Credit Score         0
dtype: int64
```

```
df.dtypes
```

```
Age                    int64
Income                 float64
Education              object
Marital Status         object
Number of Children     int64
Home Ownership         object
Credit Score           object
dtype: object
```

```
df['Income'].fillna(df['Income'].mean(),inplace=True)
```

```
df['Credit Score'].value_counts()
```

```
High       80
Average    23
Low         9
Name: Credit Score, dtype: int64
```

```
from sklearn.preprocessing import LabelEncoder,MinMaxScaler,StandardScaler

lb=LabelEncoder()
df['Education']=lb.fit_transform(df['Education'])
df['Marital Status']=lb.fit_transform(df['Marital Status'])
df['Home Ownership']=lb.fit_transform(df['Home Ownership'])
df['Credit Score']=lb.fit_transform(df['Credit Score'])
```

```
x=df.iloc[:,:-1].values
y=df.iloc[:,-1].values
```

```
from sklearn.model_selection import train_test_split
xtr,xts,ytr,yts=train_test_split(x,y,test_size=0.30,random_state=42)
```

```
min=MinMaxScaler()
min.fit(xtr)
xtr=min.transform(xtr)
xts=min.transform(xts)
```

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix,accuracy_score,classification_repor

knn=KNeighborsClassifier()
mult=MultinomialNB()
sv=SVC()

lst=[knn,mult,sv]
```

```
for i in lst:
  print(i)
  print('****************')
  i.fit(xtr,ytr)
  ypr=i.predict(xts)
  print(confusion_matrix(yts,ypr))
  print('******************')
  print(accuracy_score(yts,ypr))
  print('******************')
  print(classification_report(yts,ypr))
```

```
KNeighborsClassifier()
****************
[[ 3  0  0]
 [ 2 25  0]
 [ 4  0  0]]
******************
0.8235294117647058
******************
              precision    recall  f1-score   support

           0       0.33      1.00      0.50         3
           1       1.00      0.93      0.96        27
           2       0.00      0.00      0.00         4

    accuracy                           0.82        34
   macro avg       0.44      0.64      0.49        34
weighted avg       0.82      0.82      0.81        34

MultinomialNB()
****************
[[ 3  0  0]
 [ 2 25  0]
 [ 4  0  0]]
******************
0.8235294117647058
******************
              precision    recall  f1-score   support

           0       0.33      1.00      0.50         3
           1       1.00      0.93      0.96        27
           2       0.00      0.00      0.00         4

    accuracy                           0.82        34
   macro avg       0.44      0.64      0.49        34
weighted avg       0.82      0.82      0.81        34

SVC()
****************
[[ 3  0  0]
 [ 2 25  0]
 [ 4  0  0]]
******************
0.8235294117647058
******************
              precision    recall  f1-score   support

           0       0.33      1.00      0.50         3
```

```
              0       0.33      1.00      0.50         3
              1       1.00      0.93      0.96        27
              2       0.00      0.00      0.00         4

    accuracy                           0.82        34
   macro avg       0.44      0.64      0.49        34
weighted avg       0.82      0.82      0.81        34

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py
  _warn_prf(average, modifier, msg_start, len(result))
```

Colab paid products - Cancel contracts here