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

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data = pd.read_csv("banknotes.csv")

data.shape

(1372, 5)

data.size

6860

data.head()

   Variance  Skewness  Curtosis  Entropy  Class
0   3.62160    8.6661   -2.8073 -0.44699      0
1   4.54590    8.1674   -2.4586 -1.46210      0
2   3.86600   -2.6383    1.9242  0.10645      0
3   3.45660    9.5228   -4.0112 -3.59440      0
4   0.32924   -4.4552    4.5718 -0.98880      0

X = data.drop('Class', axis=1)
y = data['Class']

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size
=0.20)

X.head()

   Variance  Skewness  Curtosis  Entropy
0   3.62160    8.6661   -2.8073 -0.44699
1   4.54590    8.1674   -2.4586 -1.46210
2   3.86600   -2.6383    1.9242  0.10645
3   3.45660    9.5228   -4.0112 -3.59440
4   0.32924   -4.4552    4.5718 -0.98880

y.head()

0    0
1    0
2    0
3    0
4    0
Name: Class, dtype: int64

X.shape
```

```
(1372, 4)
```

```
y.shape
```

```
(1372,)
```

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='linear')
svclassifier.fit(X_train, y_train)
```

```
SVC(kernel='linear')
```

```
y_predict = svclassifier.predict(X_test)
```

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_predict))
print(classification_report(y_test,y_predict))
```

```
[[139   3]
 [  2 131]]
              precision    recall  f1-score   support

           0       0.99      0.98      0.98       142
           1       0.98      0.98      0.98       133

    accuracy                           0.98       275
   macro avg       0.98      0.98      0.98       275
weighted avg       0.98      0.98      0.98       275
```

```
X_test_one = np.array([[3.62160, 8.6661, -2.8073, -0.44699]])  # Keep
it as a row with four features
y_predict1 = svclassifier.predict(X_test_one)
print( y_predict1)
```

```
[0]
```

```
C:\Users\MGM\anaconda3\Lib\site-packages\sklearn\base.py:493:
UserWarning: X does not have valid feature names, but SVC was fitted
with feature names
  warnings.warn(
```

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='rbf')
svclassifier.fit(X_train, y_train)
```

```
SVC()
```

```
y_predict = svclassifier.predict(X_test)
```

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_predict))
print(classification_report(y_test,y_predict))
```

```
[[142   0]
 [  0 133]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       142
           1       1.00      1.00      1.00       133

    accuracy                           1.00       275
   macro avg       1.00      1.00      1.00       275
weighted avg       1.00      1.00      1.00       275
```

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='sigmoid')
svclassifier.fit(X_train, y_train)
```

```
SVC(kernel='sigmoid')
```

```
y_predict = svclassifier.predict(X_test)
```

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_predict))
print(classification_report(y_test,y_predict))
```

```
[[95 47]
 [49 84]]
              precision    recall  f1-score   support

           0       0.66      0.67      0.66       142
           1       0.64      0.63      0.64       133

    accuracy                           0.65       275
   macro avg       0.65      0.65      0.65       275
weighted avg       0.65      0.65      0.65       275
```

```
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 , random_state=0)
```

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='linear')
svclassifier.fit(X_train, y_train)
```

```
SVC(kernel='linear')
```

```
y_predict = svclassifier.predict(X_test)
```

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_predict))
print(classification_report(y_test,y_predict))
```

```
[[227   5]
 [  0 180]]
              precision    recall  f1-score   support

           0       1.00      0.98      0.99       232
           1       0.97      1.00      0.99       180

    accuracy                           0.99       412
   macro avg       0.99      0.99      0.99       412
weighted avg       0.99      0.99      0.99       412
```

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='rbf')
svclassifier.fit(X_train, y_train)
```

```
SVC()
```

```
y_predict = svclassifier.predict(X_test)
```

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_predict))
print(classification_report(y_test,y_predict))
```

```
[[231   1]
 [  0 180]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       232
           1       0.99      1.00      1.00       180

    accuracy                           1.00       412
   macro avg       1.00      1.00      1.00       412
weighted avg       1.00      1.00      1.00       412
```

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='sigmoid')
svclassifier.fit(X_train, y_train)
```

```
SVC(kernel='sigmoid')
```

```
y_predict = svclassifier.predict(X_test)
```

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_predict))
print(classification_report(y_test,y_predict))
```

```
[[165  67]
 [ 68 112]]
              precision    recall  f1-score   support
```

```
           0       0.71      0.71      0.71       232
           1       0.63      0.62      0.62       180

    accuracy                           0.67       412
   macro avg       0.67      0.67      0.67       412
weighted avg       0.67      0.67      0.67       412
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size
=0.35 , random_state=0)
```

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='linear')
svclassifier.fit(X_train, y_train)
```

```
SVC(kernel='linear')
```

```
y_predict = svclassifier.predict(X_test)
```

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_predict))
print(classification_report(y_test,y_predict))
```

```
[[264   6]
 [  0 211]]
              precision    recall  f1-score   support

           0       1.00      0.98      0.99       270
           1       0.97      1.00      0.99       211

    accuracy                           0.99       481
   macro avg       0.99      0.99      0.99       481
weighted avg       0.99      0.99      0.99       481
```

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='rbf')
svclassifier.fit(X_train, y_train)
```

```
SVC()
```

```
y_predict = svclassifier.predict(X_test)
```

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_predict))
print(classification_report(y_test,y_predict))
```

```
[[269   1]
 [  0 211]]
              precision    recall  f1-score   support
```

```
           0       1.00      1.00      1.00       270
           1       1.00      1.00      1.00       211

    accuracy                           1.00       481
   macro avg       1.00      1.00      1.00       481
weighted avg       1.00      1.00      1.00       481
```

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='sigmoid')
svclassifier.fit(X_train, y_train)
```

```
SVC(kernel='sigmoid')
```

```
y_predict = svclassifier.predict(X_test)
```

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_predict))
print(classification_report(y_test,y_predict))
```

```
[[193  77]
 [ 76 135]]
              precision    recall  f1-score   support

           0       0.72      0.71      0.72       270
           1       0.64      0.64      0.64       211

    accuracy                           0.68       481
   macro avg       0.68      0.68      0.68       481
weighted avg       0.68      0.68      0.68       481
```