```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
```

```
data = pd.read_csv("bill_authentication.csv")
print(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)
print(x)
```

```
      Variance  Skewness  Curtosis  Entropy
0      3.62160   8.66610   -2.8073 -0.44699
1      4.54590   8.16740   -2.4586 -1.46210
2      3.86600  -2.63830    1.9242  0.10645
3      3.45660   9.52280   -4.0112 -3.59440
4      0.32924  -4.45520    4.5718 -0.98880
...        ...       ...       ...      ...
1367   0.40614   1.34920   -1.4501 -0.55949
1368  -1.38870  -4.87730    6.4774  0.34179
1369  -3.75030 -13.45860   17.5932 -2.77710
1370  -3.56370  -8.38270   12.3930 -1.28230
1371  -2.54190  -0.65804    2.6842  1.19520

[1372 rows x 4 columns]
```

```
y = data['Class']
print(y)
```

```
0       0
1       0
2       0
3       0
4       0
       ..
1367    1
1368    1
1369    1
1370    1
1371    1
Name: Class, Length: 1372, dtype: int64
```

```
from sklearn.model_selection import train_test_split
```

```
# Test Size = 20%
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20)
```

```
model = SVC(kernel='rbf')
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
```

```
print("Linear Kernel SVM Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", cm)
```

```
Linear Kernel SVM Accuracy: 1.0
Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00       158
           1       1.00      1.00      1.00       117

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

Confusion Matrix:
 [[158   0]
 [  0 117]]
```

```
# Test Size = 30


x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.30)


model = SVC(kernel='rbf')
model.fit(x_train, y_train)
y_pred = model.predict(x_test)


print("Linear Kernel SVM Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", cm)
```

```
Linear Kernel SVM Accuracy: 1.0
Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00       216
           1       1.00      1.00      1.00       196

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

Confusion Matrix:
 [[216   0]
 [  0 196]]
```

```
# Test Size = 35


x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.30)


model = SVC(kernel='rbf')
model.fit(x_train, y_train)
y_pred = model.predict(x_test)


print("Linear Kernel SVM Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", cm)
```

```
Linear Kernel SVM Accuracy: 0.9975728155339806
Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00       245
           1       0.99      1.00      1.00       167

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

Confusion Matrix:
 [[244   1]
 [  0 167]]
```

```
# Test Size = 40


x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.40)


model = SVC(kernel='rbf')
model.fit(x_train, y_train)
y_pred = model.predict(x_test)


print("Linear Kernel SVM Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", cm)
```

```
Linear Kernel SVM Accuracy: 0.994535519125683
Classification Report:
               precision    recall  f1-score   support

           0       1.00      0.99      0.99       296
           1       0.99      1.00      0.99       253
```

```
    accuracy                           0.99       549
   macro avg       0.99      0.99      0.99       549
weighted avg       0.99      0.99      0.99       549

Confusion Matrix:
 [[293   3]
 [  0 253]]
```