```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.naive_bayes import BernoulliNB
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix
from sklearn import metrics
from sklearn import datasets
```

```
dataset = datasets.load_iris()
X = dataset.data[:, :2]
y = dataset.target
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=0)
```

```
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
classifier = BernoulliNB()
classifier.fit(X_train, y_train)
```

```
BernoulliNB()
```

```
y_pred = classifier.predict(X_test)
```

```
cm = confusion_matrix(y_test, y_pred)
```

```
confusion_matrix=cm
FP = confusion_matrix.sum(axis=0) - np.diag(confusion_matrix)
FN = confusion_matrix.sum(axis=1) - np.diag(confusion_matrix)
TP = np.diag(confusion_matrix)
TN = confusion_matrix.sum() - (FP + FN + TP)
```

```
ACC = (TP+TN)/(TP+FP+FN+TN)
```

```
ERR = 1- ACC
```

```
PRC = TP/(TP+FP)
```

```
RCL = TP/(TP+FN)
```

```
print('True Positives (TP): ', TP[0])
print('False Positives (FP): ', FP[0])
```

```
print('False Positives (FP): ', FP[0])
print('True Negatives (TN): ', TN[0])
print('False Negatives (FN): ', FN[0])
print('Accuracy: ', ACC[0])
print('Error Rate: ', ERR[0])
print('Precision: ', PRC[0])
print('Recall: ', RCL[0])
```

```
True Positives (TP):  9
False Positives (FP):  0
True Negatives (TN):  19
False Negatives (FN):  2
Accuracy:  0.9333333333333333
Error Rate:  0.06666666666666665
Precision:  1.0
Recall:  0.8181818181818182
```

✓ 0s completed at 8:49 PM