Practical 6 – Naïve Bayes

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score

df = pd.read_csv("iris.csv")
X = df.iloc[:, :-1]
y = df.iloc[:, -1]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
nb_classifier = GaussianNB()
nb_classifier.fit(X_train, y_train)
y_pred = nb_classifier.predict(X_test)
cm = confusion_matrix(y_test, y_pred)
TN, FP, FN, TP = cm.ravel() if cm.shape == (2,2) else (None, None, None, None)
accuracy = accuracy_score(y_test, y_pred)
error_rate = 1 - accuracy
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
print("Confusion Matrix:")
print(cm)
print(f"True Negatives (TN): {TN}")
print(f"False Positives (FP): {FP}")
print(f"False Negatives (FN): {FN}")
print(f"True Positives (TP): {TP}")
print(f"Accuracy: {accuracy:.4f}")
print(f"Error Rate: {error_rate:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
```

Output :

Confusion Matrix:

[[10 0 0]

 [0 9 0]

 [0 0 11]]

True Negatives (TN): None

False Positives (FP): None

False Negatives (FN): None

True Positives (TP): None

Accuracy: 1.0000

Error Rate: 0.0000

Precision: 1.0000

Recall: 1.0000