Practical No-6

Date of Conduction : **Date of Checking:**

Data Analytics III

1. Implement Simple Naïve Bayes classification algorithm using Python/R on iris.csv dataset.
2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.

Python Code

```
import pandas as pd
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, f1_score

# Load the Iris dataset
iris_data = pd.read_csv('iris.csv')

# Separate features (X) and target variable (y)
X = iris_data.iloc[:, :-1]
y = iris_data.iloc[:, -1]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Create a Naïve Bayes model (Gaussian Naïve Bayes for continuous features)
model = GaussianNB()

# Train the model
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Compute Confusion Matrix
conf_matrix = confusion_matrix(y_test, y_pred)

# Compute Performance Metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')  # weighted
precision for multi-class
recall = recall_score(y_test, y_pred, average='weighted')  # weighted
recall for multi-class
f1 = f1_score(y_test, y_pred, average='weighted')  # weighted F1 score for
multi-class

# Print the results
print("Confusion Matrix:")
print(conf_matrix)
print("\nAccuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1)
```

OUTPUT

```
"C:\Users\Ram Kumar Solanki\PycharmProjects\pythonProject\venv\Scripts\python.exe"
"C:\Users\Ram Kumar Solanki\PycharmProjects\MBA_BFS\main.py"
Confusion Matrix:
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]

Accuracy: 1.0
Precision: 1.0
Recall: 1.0
F1 Score: 1.0

Process finished with exit code 0
```