

```
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

from sklearn.datasets import load_iris

# Load the Iris dataset

iris_data = load_iris()

x = pd.DataFrame(iris_data['data'], columns=iris_data['feature_names'])
y = pd.DataFrame(iris_data['target'], columns=['target'])

# 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)
```