```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_iris
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from mlxtend.plotting import plot_confusion_matrix
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report,
precision_score, recall_score, f1_score
import warnings
warnings.filterwarnings("ignore")
%matplotlib inline

iris = load_iris()
iris.keys()

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

x.head()

x.shape, y.shape

x.info()

y.info()

x.describe()
```

```
scaler = StandardScaler()
x = scaler.fit_transform(x.values)

x_train, x_test, y_train, y_test = train_test_split(x, y.values, test_size=0.2, random_state=42)

x_train.shape, x_test.shape, y_train.shape, y_test.shape

model = GaussianNB()

model.fit(x_train, y_train)

y_pred = model.predict(x_test)

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

plot_confusion_matrix(conf_mat=cm, figsize=(5,5), show_normed=True)
plt.show()

print(f"TP value is {cm[0,0]}")
print(f"TN value is {cm[1,1] + cm[2,2]}")
print(f"FP value is {cm[0,1] + cm[0,2]}")
print(f"FN value is {cm[1,0] + cm[2,0]}")

print(f"Accuracy score is {accuracy_score(y_test, y_pred)}")

print(f"Error rate is {1 - accuracy_score(y_test, y_pred)}")

print(f"Precision score is {precision_score(y_test, y_pred, average='macro')}")

print(f"Recall score is {recall_score(y_test, y_pred, average='macro')}")
```

```
print(classification_report(y_test, y_pred))
```