```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.model selection import train test split
from \ sklearn.linear\_model \ import \ Logistic Regression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
iris = datasets.load_iris()
X = iris.data
y = iris.target
X = X[y != 2]
y = y[y != 2]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
model = LogisticRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
\verb|print("\nConfusion Matrix:\n", confusion_matrix(y\_test, y\_pred))| \\
print("\nClassification Report:\n", classification_report(y_test, y_pred))
X_vis = X_train[:, :2]
model_vis = LogisticRegression().fit(X_vis, y_train)
x_{min}, x_{max} = X_{vis}[:, 0].min() - 1, <math>X_{vis}[:, 0].max() + 1
y_min, y_max = X_vis[:, 1].min() - 1, X_vis[:, 1].max() + 1
xx, yy = np.meshgrid(np.linspace(x_min, x_max, 200),
                     np.linspace(y_min, y_max, 200))
Z = model_vis.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)
plt.contourf(xx, yy, Z, alpha=0.3)
plt.scatter(X_vis[:, 0], X_vis[:, 1], c=y_train, edgecolors='k')
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.title("Logistic Regression Decision Boundary")
plt.show()
```

accuracy

macro avg

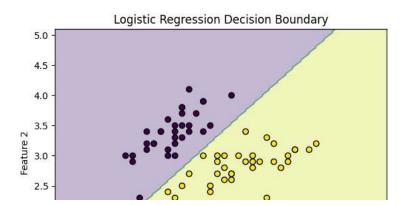
weighted avg

1.00

1.00

1.00

1.00



1.00

1.00

1.00

30

30

30