

```

import numpy as np
from sklearn.datasets import load_iris
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

def sigmoid(z):
    return 1.0 / (1.0 + np.exp(-z))

def log_reg(X, y, num_iter=200, lr=0.001):
    weights = np.zeros(X.shape[1])
    for _ in range(num_iter):
        z = np.dot(X, weights)
        h = sigmoid(z)
        grad = np.dot(X.T, (h - y)) / y.shape[0]
        weights -= grad * lr
    return weights

iris = load_iris()
X = iris.data[:, :2]
y = (iris.target != 0) * 1
sc = StandardScaler()
X_std = sc.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_std, y, test_size=0.4)
weights = log_reg(X_train, y_train)
y_pred = sigmoid(np.dot(X_test, weights)) > 0.5
print(np.mean(y_pred == y_test))

```

0.9333333333333333