姓名和学号：李胜志 2210180232

章节名称：决策树算法的鸢尾花数据集分类应用

知识目标：掌握使用决策树算法构建模型并对鸢尾花数据集进行分类。

能力目标：能够使用决策算法训练模型，构建决策树模型。

素质目标：养成分析问题、事前规划的良好习惯。

知识重点：使用决策算法构建决策树模型并对模型进行评估；计算决策树的最佳深度。

知识难点：计算决策树的最佳深度。

使用决策树算法对Sklearn自带的鸢尾花数据集进行分类。

1. 确定决策树的最佳深度值。

from sklearn import datasets

from sklearn.model\_selection import train\_test\_split

import numpy as np

mydata = datasets.load\_iris()

x = mydata.data[:, 2:4]

y = mydata.target

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=50, random\_state=0)

from sklearn.tree import DecisionTreeClassifier

# 决策树深度和预测准确率计算

depth = np.arange(1, 15)

scores = []

for i in depth:

model = DecisionTreeClassifier(criterion='entropy', max\_depth=i)

model.fit(x\_train, y\_train)

score = model.score(x\_test, y\_test)

scores.append(score)

index = np.argmax(scores) # 获取预测准确率最大值的下标

print(depth[index]) # 得到预测准确率最大值对应的深度值

运行结果：3

1. 但深度为3时，建立决策树模型并评估。

from matplotlib import pyplot as plt

plt.plot(depth, scores,'o-')

plt.rcParams['font.sans-serif'] = 'Microsoft Yahei'

plt.xlabel('决策树深度')

plt.ylabel('预测准确率')

# 构建深度为3的决策树模型

model = DecisionTreeClassifier(criterion='entropy', max\_depth=3)

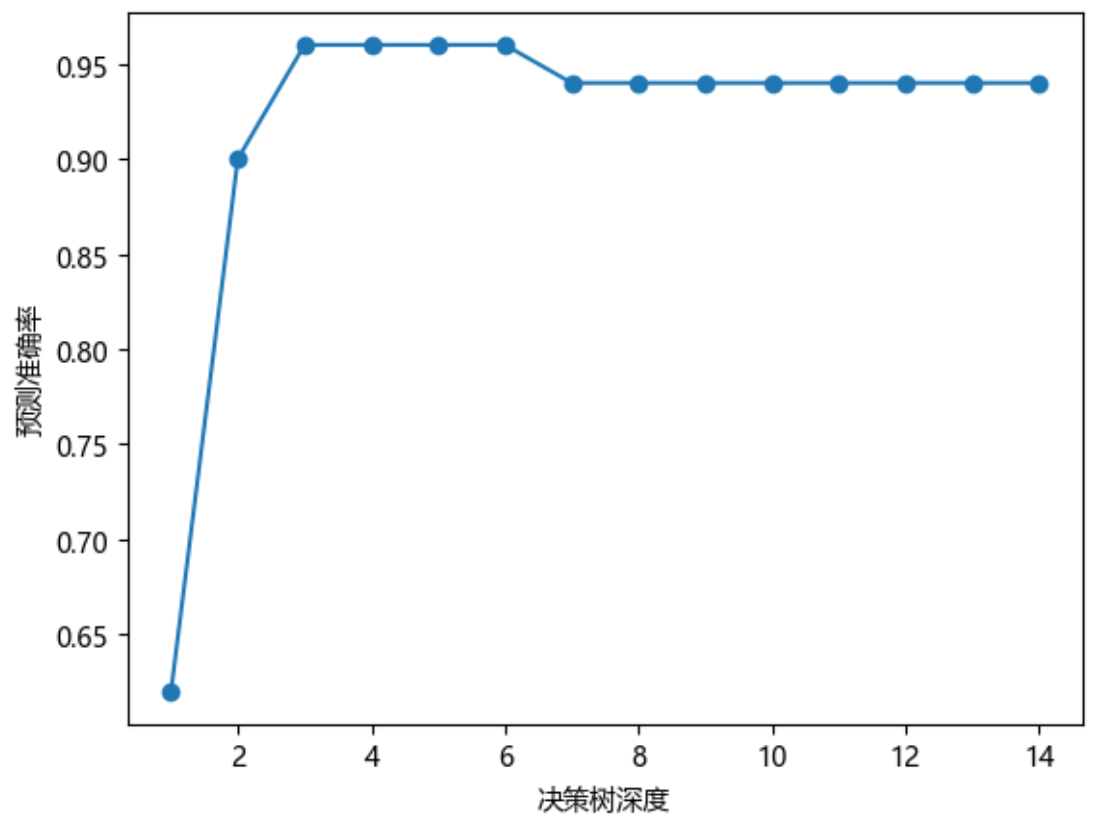
model.fit(x\_train, y\_train)

# 模型评估

score = model.score(x\_test, y\_test)

print(score)

运行结果：0.96



1. 绘制可视化图像，显示分类结果。

from matplotlib.colors import ListedColormap

N, M = 500, 800 # 定义网格采样点

t1 = np.linspace(0, 8, N) # 生成采样点的横坐标值

t2 = np.linspace(0, 5, M) # 生成采样点的纵坐标值

x1, x2 = np.meshgrid(t1, t2) # 生成网格采样点

x\_new = np.stack((x1.flat,x2.flat), axis=1) # 将采样点作为测试点

y\_predict = model.predict(x\_new) # 预测测试点的值

y\_hat = y\_predict.reshape(x1.shape) # 设置与 x1 相同的形状

iris\_cmap = ListedColormap(["#ACF080", "#A0A0FF", "#FF8080"]) # 设置分类界面的颜色

# 绘制分类界面

plt.pcolormesh(x1, x2, y\_hat, cmap = iris\_cmap)

plt.scatter(x[y==0, 0], x[y==0, 1], c="b", marker="o", s=60) # 绘制标签值为0 的样本点

plt.scatter(x[y==1, 0], x[y==1, 1], c="r", marker="^", s=60) # 绘制标签值为1 的样本点

plt.scatter(x[y==2, 0], x[y==2, 1], c="g", marker=".", s=60) # 绘制标签值为2 的样本点

plt.rcParams["font.sans-serif"] = "Microsoft Yahei"

plt.xlabel("花瓣长度")

plt.ylabel("花瓣宽度")

plt.show()

运行结果：

