## 附录

# 导入所需库

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.datasets import load\_iris

from sklearn.tree import DecisionTreeClassifier, plot\_tree

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

from sklearn.metrics import accuracy\_score, confusion\_matrix, ConfusionMatrixDisplay

# 1. 加载鸢尾花数据集

iris = load\_iris()

X = iris.data  # 特征矩阵

y = iris.target  # 目标变量

feature\_names = iris.feature\_names  # 特征名称

class\_names = iris.target\_names  # 类别名称

# 2. 划分训练集和测试集（70%训练，30%测试）

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# 3. 构建决策树分类器

clf = DecisionTreeClassifier(criterion='gini', max\_depth=3, random\_state=42)

clf.fit(X\_train, y\_train)  # 模型训练

# 4. 预测结果

y\_pred = clf.predict(X\_test)  # 在测试集上进行预测

# 5. 评估模型性能

accuracy = accuracy\_score(y\_test, y\_pred)  # 计算准确率

print(f"测试集准确率: {accuracy \* 100:.2f}%")

# 混淆矩阵

cm = confusion\_matrix(y\_test, y\_pred)

disp = ConfusionMatrixDisplay(confusion\_matrix=cm, display\_labels=class\_names)

disp.plot(cmap=plt.cm.Blues)

plt.title("Confusion Matrix")

plt.show()

# 6. 可视化决策树结构

plt.figure(figsize=(12, 8))

plot\_tree(clf,

          feature\_names=feature\_names,

          class\_names=class\_names,

          filled=True,

          rounded=True)

plt.title("Decision Tree Visualization")

plt.show()

# 7. 特征重要性分析

importances = clf.feature\_importances\_

for i, feature in enumerate(feature\_names):

    print(f"{feature}: {importances[i]:.4f}")

# 绘制特征重要性图

plt.figure(figsize=(8, 6))

plt.bar(feature\_names, importances, color="skyblue")

plt.title("Feature Importances")

plt.xlabel("Features")

plt.ylabel("Importance Score")

plt.xticks(rotation=45)

plt.show()