

实验 6 基于决策树算法的学员 Python 水平判断实习指导书

一、实验基本信息

项目	内容
实验名称	基于决策树算法的学员 Python 水平判断
授课课程	大数据分析
授课班级	工业工程 2241 班
实验学时	2 学时（90 分钟）
实验类型	算法应用型实验
实验目标	1. 掌握决策树模型对分类数据的训练流程；2. 熟练通过交互式输入完成学员 Python 水平判断；3. 培养数据输入严谨性、结果验证细致性的科研品质。

二、实验原理

决策树是基于“if - else”规则的监督学习分类算法，通过递归分裂特征（问卷问题答案）生成树形结构，最终叶节点对应“Python 水平”类别（如编程高手、门外汉等）。本实验以问卷答案为特征，训练决策树模型，实现对学员 Python 水平的自动化判断。

三、实验环境准备

1. 软件要求

1. 操作系统: Windows 10/11 (64 位) ;
2. 软件套件: Anaconda3 (2024.02 及以上版本) ;
3. 开发工具: Spyder (Anaconda 自带, 确保 IPython 控制台可正常运行代码) ;
4. 依赖库: `scikit-learn` (决策树模型)、`numpy` (数据处理) 。

2. 预检查步骤

打开 Spyder, 在 IPython 控制台输入以下代码, 验证库是否正常导入:

```
import sklearn
import numpy as np
from sklearn.tree import DecisionTreeClassifier
print("Scikit-learn 版本: ", sklearn.__version__)
print("NumPy 版本: ", np.__version__)
print("库导入成功! ")
```

若提示 `ModuleNotFoundError`, 打开 Anaconda Prompt 执行:

```
conda install scikit-learn numpy
```

四、实验内容与操作步骤

模块 1: 问卷数据与模型训练

在 Spyder 中新建文件, 输入以下代码:

```
from sklearn import tree
import numpy as np
# 问卷问题
questions = (
    '《Python 程序设计基础 (第 2 版) 》',
    '《Python 程序设计基础与应用》',
```

```

'《大数据的 Python 基础》',
'《Python 程序设计开发宝典》',
'《中学生可以这样学 Python》',
'《玩转 Python 轻松过二级》',
'《Python 编程基础与案例集锦（中学版）》',
'微信公众号“Python 小屋”的免费资料',
)
# 样本答案：每个列表对应一个学员的问卷回答（0=没看过，1=部分看过，2=大部分可以看懂，3=没压力）
answers = [
    [3,0,0,0,0,3,3,1],
    [3,0,0,0,0,3,3,2],
    [3,0,0,0,0,3,3,3],
    [0,3,0,0,0,0,0,3],
    [0,3,0,0,0,0,0,2],
    [0,3,0,0,0,0,0,1],
    [0,0,3,0,0,3,0,3],
    [0,0,3,0,0,3,0,2],
    [0,0,3,0,0,3,0,1],
    [0,0,0,3,0,0,3,3],
    [0,0,0,3,0,0,3,2],
    [0,0,0,3,0,0,3,1],
    [0,0,0,0,3,3,0,3],
    [0,0,0,0,3,3,0,2],
    [0,0,0,0,3,3,0,1],
]
# Python 水平标签
labels = ['编程高手','中极选手','高级选手','门外汉','初级选手','中级选手','编程高手','高级选手',
'初级选手','编程高手','高级选手','初级选手','编程高手','高级选手','初级选手']
# 初始化并训练决策树分类器
clf = DecisionTreeClassifier()

```

```
clf = clf.fit(answers, labels)
```

模块 2：交互式 Python 水平预测

继续添加代码，实现用户交互与预测：

```
yourAnswer = []

print("\n=== 请回答以下问卷（输入 0-3 的整数，0=没看过，1=部分看过，2=大部分可以看懂，3=没压力） ===")

for i in range(len(questions)):
    while True:
        try:
            answer = int(input(f"问题{i+1}: {questions[i]}? "))
            yourAnswer.append(answer)
            break
        except:
            print("输入无效，请重新输入！")

prediction = clf.predict(np.array(yourAnswer).reshape(1, -1))[0]
print(f"\n 你的 Python 水平预测为： {prediction}")
```

模块 3：决策树可视化（可选）

若需可视化决策树（直观查看决策逻辑），添加以下代码：

```
import matplotlib.pyplot as plt
plt.figure(figsize=(15, 10))
tree.plot_tree(
    clf,
    feature_names=questions,
    class_names=clf.classes_,
    filled=True,
    rounded=True
```

```
)  
plt.title("Python 水平判断决策树", fontsize=14)  
plt.savefig("python_skill_tree.png", dpi=300, bbox_inches="tight")  
plt.show()
```

五、实验注意事项

- 1. 输入有效性：问卷输入必须为 0 - 3 的整数，代码通过 try - except 拦截无效输入（如字母、符号等），确保数据符合模型要求。
- 2. 数据一致性：answers（样本答案）与 labels（Python 水平标签）的长度需严格一致，否则模型训练会报错。
- 3. 交互流程：按问卷问题依次输入，每道题输入后立即进入下一题，确保交互逻辑顺畅。

六、实验报告要求

- 1. 报告结构：包含“实验目的”“实验原理”“实验步骤与结果（附代码截图、交互预测过程截图）”“问题与解决方法”“实验总结”。
- 2. 关键截图：需附“问卷输入过程（含无效输入拦截提示）”“Python 水平预测结果”截图，标注清晰且为本人操作。
- 3. 问题与解决：记录输入错误（如误输字母、超出 0-3 范围的数字）时的拦截提示及重新输入的过程。

七、实验考核标准

考核维度	考核要点（总分 100 分）	分值
代码一致性	数据定义、模型训练、交互预测代码与教材完全一致（语法错误扣 5 分 / 处）	40
交互有效性	输入校验生效（无效输入被拦截），预测结果合理	30

报告质量	结构完整，截图标注规范，“问题与解决”真实详细	20
科研品质	输入操作严谨，无随意无效输入，体现细致性	10