** 南昌大学实验报告**

学生姓名： 周华俊 学 号： 8008122013 专业班级： 软件工程2201班

实验类型：□ 验证■ 综合 ☑ 设计 □ 创新 实验日期： 实验成绩：

**一、实验项目名称**

分类算法

**二、实验目的**

1. 熟悉pandas、sklearn库
2. 熟悉weka平台处理与分析数据
3. 掌握决策树分类器、随机森林等

**三、实验任务**

1. 在<http://archive.ics.uci.edu/>选取课堂讲解之外的2个UCI数据，利用python开展C4.5(ID3的扩展)、随机森林、SVM等（任选一种）的数据分类分析，并与Weka上的分析做比较分析。

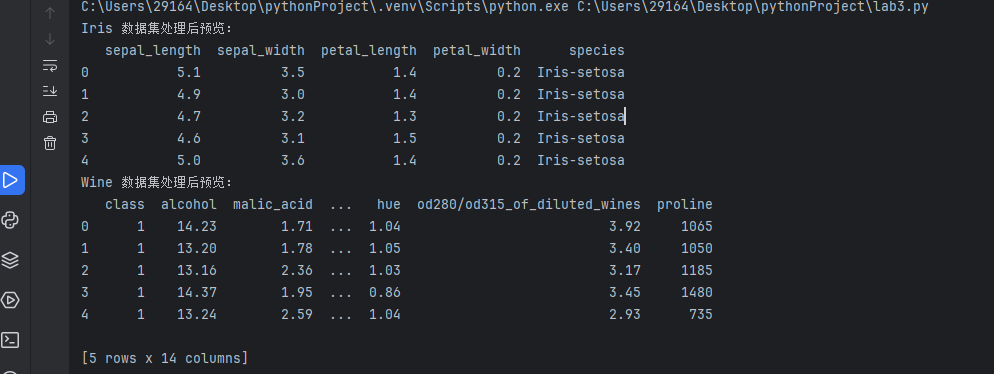
**四、主要仪器设备及耗材**

软件：pycharm。

**五、实验步骤**

1.这里选择iris和wine两个数据集，这两个是经典小型数据集，易于快速测试和对比分析，且都适用于多分类问题。加载数据集并手动列上列名然后预览

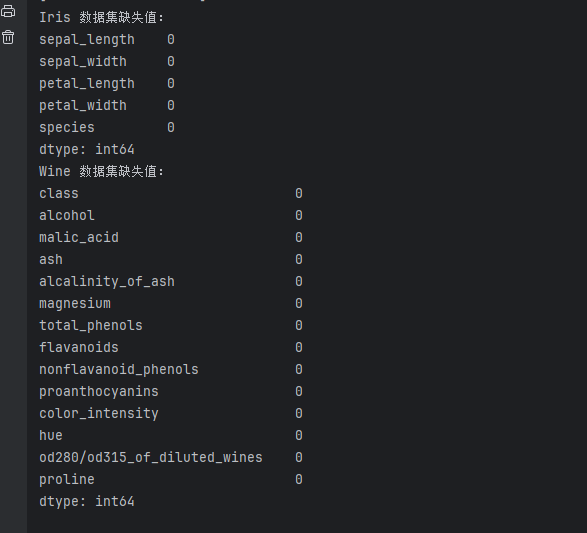
import pandas as pd  
  
# 加载 Iris 数据集  
iris\_path = r'C:\Program Files\JetBrains\fordaterevise\iris.data'  
iris\_data = pd.read\_csv(iris\_path, header=None) # 无表头加载  
# 添加列名  
iris\_columns = ['sepal\_length', 'sepal\_width', 'petal\_length', 'petal\_width', 'species']  
iris\_data.columns = iris\_columns  
print("Iris 数据集处理后预览：")  
print(iris\_data.head())  
  
# 加载 Wine 数据集  
wine\_path = r'C:\Program Files\JetBrains\fordaterevise\wine.data'  
wine\_data = pd.read\_csv(wine\_path, sep=',', header=None) # 使用逗号分隔符，无表头加载  
# 添加列名  
wine\_columns = [  
 'class', 'alcohol', 'malic\_acid', 'ash', 'alcalinity\_of\_ash', 'magnesium',  
 'total\_phenols', 'flavanoids', 'nonflavanoid\_phenols', 'proanthocyanins',  
 'color\_intensity', 'hue', 'od280/od315\_of\_diluted\_wines', 'proline'  
]  
wine\_data.columns = wine\_columns  
print("Wine 数据集处理后预览：")  
print(wine\_data.head())

输出结果如下：  


1. 数据预处理

2.1 检查缺失值

检查 Iris 和 Wine 数据集是否存在缺失值，并根据需要进行处理。可以发现数据集并无缺失值



2.2 数据标准化

from sklearn.preprocessing import StandardScaler  
# 对数据进行标准化（对于非类别特征）  
scaler = StandardScaler()  
# 对 Iris 数据集进行标准化（特征列）  
iris\_data\_scaled = iris\_data.iloc[:, :-1] # 去掉目标列  
iris\_data\_scaled = scaler.fit\_transform(iris\_data\_scaled)  
# 对 Wine 数据集进行标准化（特征列）  
wine\_data\_scaled = wine\_data.iloc[:, 1:] # 去掉目标列  
wine\_data\_scaled = scaler.fit\_transform(wine\_data\_scaled)

1. 训练模型

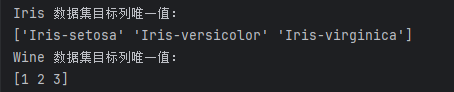
这里我选随机森林算法进行训练

# 对 Iris 数据集训练并评估模型  
rf\_iris.fit(iris\_data\_scaled, y\_iris)  
y\_pred\_iris = rf\_iris.predict(iris\_data\_scaled)  
accuracy\_iris = accuracy\_score(y\_iris, y\_pred\_iris)  
print(f"Iris 数据集的随机森林分类准确率：{accuracy\_iris \* 100:.2f}%")  
  
# 对 Wine 数据集训练并评估模型  
rf\_wine.fit(wine\_data\_scaled, y\_wine)  
y\_pred\_wine = rf\_wine.predict(wine\_data\_scaled)  
accuracy\_wine = accuracy\_score(y\_wine, y\_pred\_wine)  
print(f"Wine 数据集的随机森林分类准确率：{accuracy\_wine \* 100:.2f}%")

结果如下：

看到这个结果我直接心头一惊，担心是不是哪里弄错了，比如数据导入时出了问题，于是选择查看数据集目标唯一值

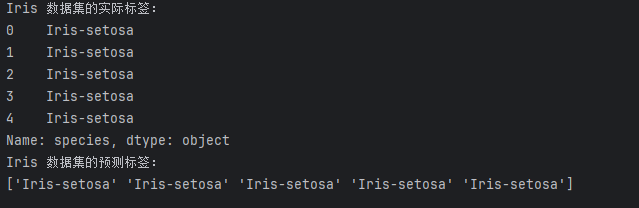
输出如下：



发现没有问题

然后进行交叉验证和对判定过程进行输出





发现没有问题，最后觉得是因为Iris 数据集是非常简单且易于分类的，它具有明显的特征区分，不同类别（Iris-setosa, Iris-versicolor, Iris-virginica）之间的特征差异非常大。尤其是有些特征，如花瓣长度和花萼长度，能够很好地区分不同的物种，同时Wine 数据集同样具有非常清晰的分类特征，每个类别（如不同的葡萄酒类型）有很好的区分，最后导致了100%的结果

1. Weka

4.1 首先将data文件转换为weka兼容的arff文件

代码部分如下：

from scipy.io import arff

import pandas as pd

# 准备 Iris 数据集

iris\_path = r'C:\Program Files\JetBrains\fordaterevise\iris.data'

iris\_data = pd.read\_csv(iris\_path, header=None)

iris\_columns = ['sepal\_length', 'sepal\_width', 'petal\_length', 'petal\_width', 'species']

iris\_data.columns = iris\_columns

# 保存为 .arff 文件

arff.dump('iris.arff',

iris\_data.values,

relation="iris",

names=iris\_data.columns)

print("Iris 数据集已保存为 iris.arff")

# 准备 Wine 数据集

wine\_path = r'C:\Program Files\JetBrains\fordaterevise\wine.data'

wine\_data = pd.read\_csv(wine\_path, header=None)

wine\_columns = [

'class', 'alcohol', 'malic\_acid', 'ash', 'alcalinity\_of\_ash', 'magnesium',

'total\_phenols', 'flavanoids', 'nonflavanoid\_phenols', 'proanthocyanins',

'color\_intensity', 'hue', 'od280/od315\_of\_diluted\_wines', 'proline'

]

wine\_data.columns = wine\_columns

# 保存为 .arff 文件

arff.dump('wine.arff',

wine\_data.values,

relation="wine",

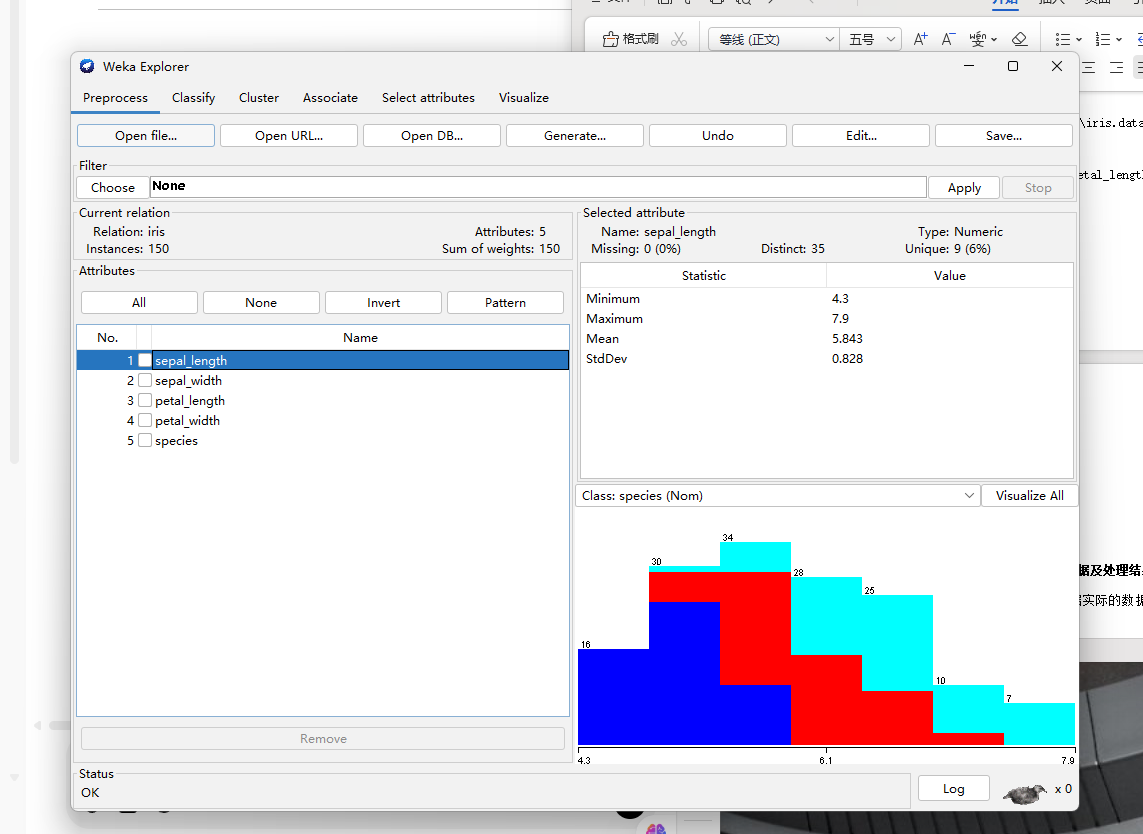
names=wine\_data.columns)

print("Wine 数据集已保存为 wine.arff")

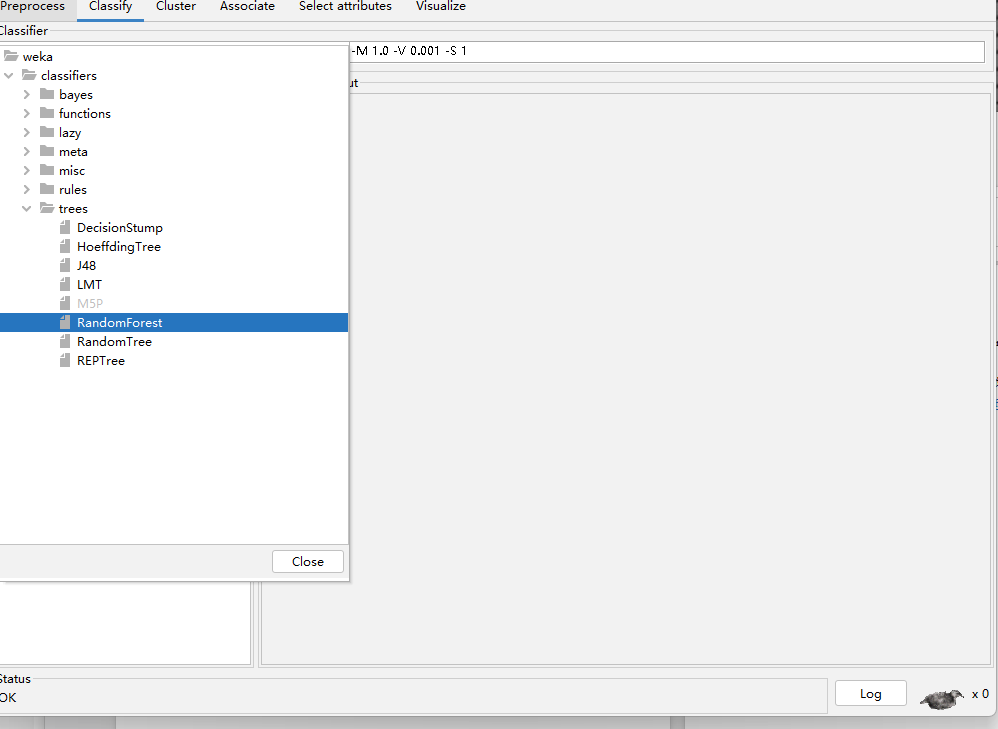
2.2

在weka导入文件

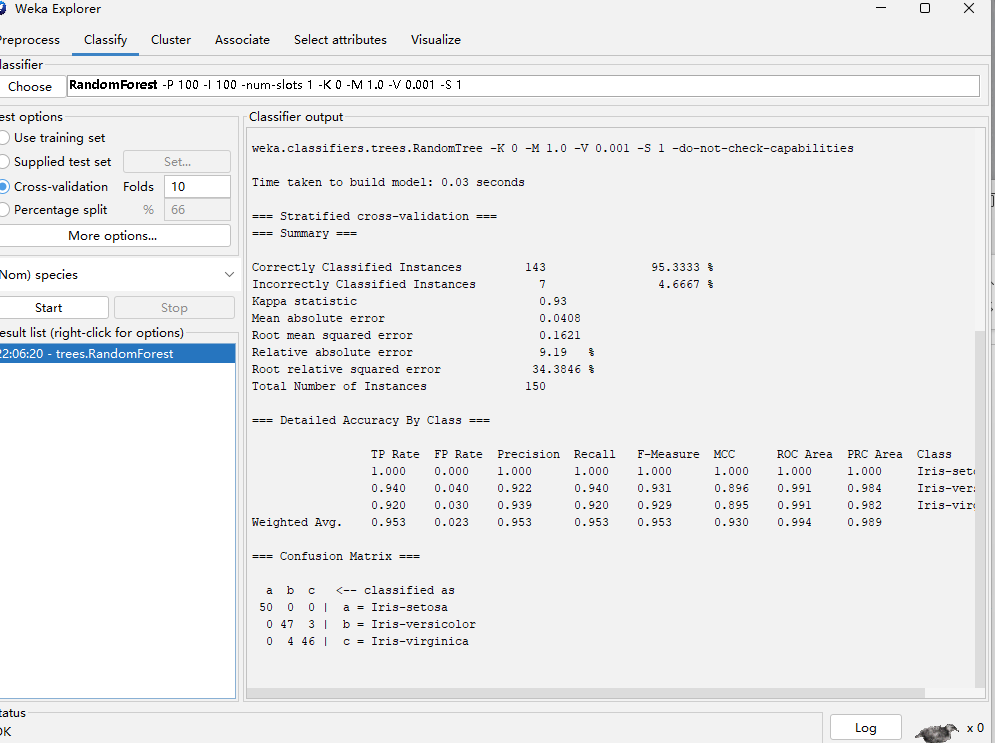
Preprocess中查看数据情况并且进行预处理



选择模型

）

运行



可知结果一致

总的来说，Python 和 Weka 各有优势：Python 通过强大的编程能力和丰富的库（如 sklearn）提供高度灵活的数据处理、模型训练和可视化功能，适合处理复杂任务和大规模数据，但需要一定的编程基础；而 Weka 以直观的图形界面和内置算法支持快速操作与评估，适合中小型数据集和教学场景，但在灵活性和复杂任务处理上有所不足。Python 更适合开发与研究，Weka 则是快速验证与教学的理想工具。

1. **思考讨论题或体会或对改进实验的建议**

思考讨论题和体会：

观察到在对Iris和Wine数据集进行分类时，准确率达到了100%。这种结果可能是由于数据集本身具有明显的特征区分性，导致模型能够轻松地区分不同类别。这种情况对于真实世界数据集的泛化能力会有什么影响？

通过比较Python和Weka工具的优劣势，你提到Python适合处理复杂任务和大规模数据，而Weka则适合中小型数据集和教学场景。进一步思考如何结合两者的优势，例如在大规模数据处理时先用Python进行数据预处理和特征工程，然后在Weka中快速尝试不同算法模型。

对改进实验的建议：

在实验中，可以尝试使用交叉验证等技术来更全面地评估模型的性能，以避免过拟合和提高模型的泛化能力。

考虑对数据进行更多的特征工程处理，如特征选择、特征变换等，以提高模型的准确性和稳定性。

在比较Python和Weka工具时，可以尝试探索更多复杂的算法、调参方法和模型评估指标，以深入了解不同工具在机器学习任务中的表现。

**八、参考资料**

1. Pang-Ning Tan, Michael Steinbach etc. Introduction to Data Mining, Second Edition, Pearson, 2019.
2. 李锐，李鹏，曲亚东等译. 机器学习实战. 高等教育出版社.2013.
3. <http://www.cse.msu.edu/~ptan/dmbook/software/>.（课本实验代码）