

南京理工大学泰州科技学院

课程：数据挖掘与可视化

Lab01 实验报告

专业班级 20 信管

学 号 2009120162

姓 名 周智涵

指导教师 叶志鹏

一、实验目标

- 回顾 Python 基础知识
- 掌握 Python 数据读写
- 掌握 Python 面向对象的概念

- 掌握 Python 模块与包的概念

二、实验要求

- 自学并巩固 Python 基础知识。
- 学术诚信，抄袭零分。
- 3 月 26 日晚 12 点之前完成，请参考相关超时惩罚机制，特殊情况除外。
- 电子报告，格式工整（代码不建议大面积截图），pdf 提交。可参考 Word，Markdown, Latex 编辑器等，可参考模板。
- 报告内容，包括清晰的描述实验步骤的结果以及结合理论课内容解释说明（为什么会有这样的结果，what, how, why）。贴上重要代码与图片截图。
- 将代码，实验报告，图片结果，数据等一并打包成压缩包 (.zip)，文件夹与实验报告命名为班级 _ 姓名 _ 学号，通过学习通上传。

三、实验环境

- 设备规格：AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz ， 64 位操作系统，基于 x64 的处理器
- 实验软件：解释器：Python3.11.1 Pycharm
- 实验使用的 python 库：Numpy, collections, random

四、任务

4.1 Iris 数据集的数据分析 50'

本部分将对 Iris 数据集进行可视化并分析，请完成以下几个任务，数据集 iris.data 在压缩包中。

- 读取 iris.data 数据集到内存中，并存储为列表命名为 iris_list，里面的元素可以是自定义对象，也可以是 2 维列表，并输出 iris.data 有多少个样本个数，以及有多少种类别。10'

```
1. with open("iris.data", "r") as file:
2.     iris_list = []
3.     # ["sepal_length", "sepal_width", "petal_length", "petal_width", "species"]
4.     for line in file:
5.         line = line.replace("\n", "") # 去除掉换行
6.         col = line.split(",") # 将数据拆分为列表
7.         iris_list.append(col) # 添加进 iris_list
8.
9. print("共有样本个数: " + str(len(iris_list)))
10. species = Counter(row[4] for row in iris_list) # 调用 counter 函数方法统计类别
```

- 按照第一列属性 (sepal length 花萼长度) 将上步操作得到的列表升序排序并打印结果。

5'

```
1. print(species)
2. sorted_ls = sorted(iris_list, key=lambda x: x[0]) # 以第一列为关键字排序
3. print(sorted_ls)
sorted()默认为升序排序
```

- 实现一个 Python 函数，能够实现对 iris_list 的有放回随机抽样，函数参数为抽样列表 data，抽样个数 number，并测试打印结果。10'

- 实现一个 Python 函数，能够实现对 iris_list 的无放回随机抽样，函数参数为抽样列表 data，抽样个数 number，并测试打印结果。10'

```
1. def sampling_with_replacement(data, number): # 有放回
2.     return random.choices(data, k=number)
3.
4.
5. def sampling_without_replacement(data, number): # 无放回
6.     return random.sample(data, k=number)
7.
8.
9. print(sampling_with_replacement(iris_list, 10))
10. print(sampling_without_replacement(iris_list, 10))
```

- 统计 iris 各列属性均值，方差，标准差，中位数并打印输出。15'

```
1. iris_list = [lst[:-1] for lst in iris_list] # 去除最后一列名称
```

```
2. float_iris_list = [] # 将由于 split, 字符串转换为浮点数
3. for tmp in iris_list:
4.     for item in tmp:
5.         float_iris_list.append(float(item))
6.
7. float_iris_list = [float_iris_list[i:i + 4] for i in range(0, len(float_iris_list), 4)] # 切片每四个分一组
8. print(float_iris_list)
9. iris_list = np.array(float_iris_list)
10. mean = np.mean(iris_list, axis=0) # 平均数
11. variance = np.var(iris_list, axis=0) # 方差
12. std_deviation = np.std(iris_list, axis=0) # 标准差
13. median = np.median(iris_list, axis=0) # 中位数
14. print(mean)
15. print(variance)
16. print(std_deviation)
17. print(median)
```

实验结果:

```
C:\Users\19089\AppData\Local\Programs\Python\Python311\python.exe C:\Users\19089
共有样本个数：150
Counter({'Iris-setosa': 50, 'Iris-versicolor': 50, 'Iris-virginica': 50})
[['4.3', '3.0', '1.1', '0.1', 'Iris-setosa'], ['4.4', '2.9', '1.4', '0.2', 'Iris-setosa'], ['4.7', '3.4', '1.6', '0.2', 'Iris-setosa'], ['4.8', '3.0', '1.4', '0.1', 'Iris-setosa'], ['4.9', '3.1', '1.5', '0.1', 'Iris-setosa'], ['5.0', '3.6', '1.4', '0.1', 'Iris-setosa'], ['5.4', '4.4', '1.5', '0.4', 'Iris-setosa'], ['5.0', '3.5', '1.6', '0.6', 'Iris-setosa'], ['4.4', '3.0', '1.4', '0.4', 'Iris-setosa'], ['4.9', '3.6', '1.4', '0.1', 'Iris-setosa'], ['4.7', '3.2', '1.6', '0.4', 'Iris-setosa'], ['4.8', '3.1', '1.6', '0.2', 'Iris-setosa'], ['5.2', '3.7', '1.4', '0.3', 'Iris-setosa'], ['5.2', '3.4', '1.6', '0.4', 'Iris-setosa'], ['5.3', '3.6', '1.5', '0.2', 'Iris-setosa'], ['4.8', '3.4', '1.6', '0.2', 'Iris-setosa'], ['4.7', '3.2', '1.4', '0.2', 'Iris-setosa'], ['4.6', '3.2', '1.4', '0.2', 'Iris-setosa'], ['4.5', '2.3', '1.3', '0.3', 'Iris-setosa'], ['4.4', '2.9', '1.4', '0.2', 'Iris-setosa']]
```

```
[['5.2', '4.1', '1.5', '0.1', 'Iris-setosa'], ['6.3', '3.3', '6.0', '2.5', 'Iris-virginica'], ['6.7', '2.5', '5.8', '1.8', 'Iris-virginica'], ['6.9', '3.1', '5.1', '2.3', 'Iris-virginica'], ['6.5', '3.0', '5.8', '2.2', 'Iris-virginica'], ['5.0', '2.3', '3.3', '1.0', 'Iris-versicolor'], ['4.4', '2.9', '1.4', '0.2', 'Iris-setosa'], ['6.9', '3.1', '5.4', '2.1', 'Iris-virginica'], ['5.7', '4.4', '1.5', '0.4', 'Iris-setosa'], ['6.9', '3.1', '5.4', '2.1', 'Iris-virginica'], ['5.4', '3.4', '1.5', '0.4', 'Iris-setosa'], ['7.7', '2.8', '6.7', '2.0', 'Iris-virginica'], ['5.1', '3.8', '1.9', '0.4', 'Iris-setosa'], ['4.6', '3.6', '1.0', '0.2', 'Iris-setosa'], ['4.7', '3.2', '1.3', '0.2', 'Iris-setosa'], ['5.1', '3.5', '1.4', '0.3', 'Iris-setosa'], ['6.6', '3.0', '4.4', '1.4', 'Iris-versicolor'], ['5.1', '3.8', '1.6', '0.2', 'Iris-setosa'], ['5.7', '3.8', '1.7', '0.3', 'Iris-setosa'], ['5.7', '2.8', '4.1', '1.3', 'Iris-versicolor']]
```

```
[5.84333333 3.054      3.75866667 1.19866667]
[0.68112222 0.18675067 3.09242489 0.57853156]
[0.82530129 0.43214658 1.75852918 0.76061262]
[5.8 3. 4.35 1.3 ]
```

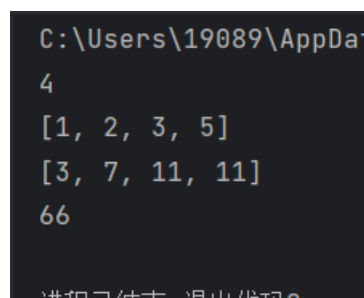
4.2 实现 *Vector* 类 50'

- 实现 `Vector` 类，并完成 `__init__(self, data)` 方法。10'
- 实现 `__len__(self)` 特殊方法，能够通过 `len(vector)` 获取到向量的维度。10'

- 实现 `__str__(self)` 特殊方法，能够通过 `print(vector)` 获取到向量的元素。10'
- 实现向量的加法运算，如 `vec3 = vec1 + vec2`。10'
- 实现向量的内积运算， $scale = \vec{x1} * \vec{x2} = x^T 1 * x2$ 。10'

```
1. class Vector:
2.     def __init__(self, data):
3.         self.data = data
4.
5.     def __len__(self):
6.         return len(self.data)
7.
8.     def __str__(self):
9.         return str(self.data)
10.
11.    def __add__(self, other):
12.        if len(self.data) != len(other.data):
13.            raise ValueError("向量必须是同一维度")
14.        result = []
15.        for i in range(len(self.data)):
16.            result.append(self.data[i] + other.data[i])
17.        return Vector(result)
18.
19.    def dot(self, other):
20.        if len(self.data) != len(other.data):
21.            raise ValueError("向量必须是同一维度")
22.        result = 0
23.        for i in range(len(self.data)):
24.            result += self.data[i] * other.data[i]
25.        return result
26.
27.
28. X = Vector([1, 2, 3, 5])
29. Y = Vector([2, 5, 8, 6])
30. print(len(X))
31. print(X)
32. print(X + Y)
33. print(X.dot(Y))
```

实验结果:



```
C:\Users\19089\AppData
4
[1, 2, 3, 5]
[3, 7, 11, 11]
66
进程已结束 退出代码0
```

五、实验结尾

以上便是此次实验的全部过程。做实验期间，遇到过多种问题，例如字符串不能参与运算，对 numpy 了解不多，科学计算的部分，基本上是边搜边做。向量相加和内积，一开始未注意到需要两向量维度一样，经过查询才改正。

。

2023/3/15 13: 20

周智涵