

**学生实验实习报告册**

|  |  |
| --- | --- |
| 学年学期： | 2023-2024 学年 🞏春■秋学期 |
| 课程名称： | Python程序设计与应用 |
| 学生学院： | 自动化学院 |
| 专业班级： | 08052102 |
| 学生学号： | 2021212981 |
| 学生姓名： | 王忠全 |
| 联系电话： | 1508669338 |

**重庆邮电大学教务处制**

|  |  |  |  |
| --- | --- | --- | --- |
| **实验测验名称** | **Python在科学计算和可视化中的应用实验** | | |
| **地点** | C318 | **时间** | 第九周9-12节 |
| **9-12指导教师** | 张 焱 | **成绩** |  |
| **一、实验目的**  1、了解科学计算的基本概念；  2、 了解数据可视化的概念；  3、 运用科学计算库进行矩阵分析和数值运算；  4、 了解图像的矩阵表示和处理；  5、 运行数据绘图库进行简单图表绘制；  6、 掌握SQLite数据库的创建、以及增删改查操作。 | | | |
| **二、实验测验所需仪器设备和软件**  Pychram图标 的图像结果  Pycharm 2022.2.3  环境：Python 3.8 | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **三、实验题目**   1. 使用KNN分类算法实现根据身高和体重对体型分类。 2. 使用k-means聚类算法进行分类。 3. 使用matplotlib绘制折线图对龟兔赛跑中兔子和乌龟的行走轨迹进行可视化。 4. 科学坐标图绘制：根据给定的数据绘制阻尼衰减曲线图。 5. 多级雷达图绘制：根据给定的任务能力数据，绘制多级雷达图。 6. 在sqlite3数据库“Student.db”中，新建一张名为“userinfo”的表，并插入以下记录后，将王五的Email修改更新为Wangwu@163.com后，将所有的记录打印输出。  |  |  |  |  | | --- | --- | --- | --- | | StuNumber | Name | ClassNumber | Email | | 20190001 | 张三 | C01 | Zhangsan@163.com | | 20190002 | 李四 | C02 | Lisi@163.com | | 20190003 | 王五 | C03 | Wangwu@qq.com | | 20190004 | 小明 | C04 | Xiaoming@qq.com |   **接下来分别对每个题目进行分析：**   1. **【使用KNN分类算法实现根据身高和体重对体型分类】**     KNN（K-Nearest Neighbors）分类算法是一种基于实例的监督学习算法，用于对未知样本进行分类，其基本思想是通过测量不同特征之间的距离来判断样本的类别。  算法过程：  准备数据集：收集已经标记好类别的样本数据集。  选择K值：确定要考虑的最近邻居的数量K。K可以是任意正整数，通常通过交叉验证或其他方法来确定最佳的K值。  计算距离：对于待分类的样本，计算它与训练集中每个样本之间的距离。通常使用欧氏距离、曼哈顿距离等作为距离度量方式。  选择最近的K个邻居：根据计算得到的距离，选择与待分类样本距离最近的K个样本作为最近邻居。  确定类别：根据K个最近邻居的类别，采取多数表决的方式确定待分类样本的类别。即，选择K个最近邻居中出现次数最多的类别作为待分类样本的类别。  输出结果：将待分类样本归类到确定的类别中。    对于使用KNN算法对身高和体重进行分类，就想到首先需要采集数据，这里我列举了5x10一共50人的身高和体重，每一位的身高体重给他定义了一个标签“偏瘦”“标准”“偏胖”，于是得到了数据集。  然后调用sklearn.neighbors里面的KneighborsClassifier分类器，实例化后训练数据，得到模型后进行测试预测，输入三个测试人的身高和体重，观察其预测结果。  **2、【使用k-means聚类算法进行分类】**  K-means  ——物以类聚，人以群分。    沿用第一题的身高体重数据,对身高和体重进行聚类，这里就是需要去了解学习sklearn.cluster 库里面的 Kmeans包，就是专门用来进行k-means聚类算法进行分类。  使用Kmeans算法训练数据集，之后遍历循环得到分类后的不同类的点，将同一类的点赋予相同颜色，最后使用matplotlib.pyplot进行图像可视化。  **3、【使用matplotlib绘制折线图对龟兔赛跑中兔子和乌龟的行走轨迹进行可视化】**    我们大家儿时都喜欢的故事书《伊索寓言》就曾经讲过这样一个故事—龟兔赛跑：比赛开始后，兔子飞快地冲出了起点，远远领先于乌龟，他看到自己的优势，就觉得胜利已经近在眼前。于是，兔子决定在路边的树下休息片刻，等待乌龟迎头赶上时再继续比赛。  而乌龟虽然行动缓慢，但他一直坚持不懈地朝目标前进。他没有停下来休息，而是一步一步地向前移动。当乌龟走到兔子停下的树下时，他看到兔子正在打盹，于是他默默地继续前行。  当兔子醒来时，他发现乌龟已经接近终点了。他慌忙起身，全力冲刺，但已经来不及了。乌龟以他稳定的速度跑到终点，成为了比赛的胜利者。  我们使用Python中的matplotlib库，对这个故事进行形象的表示，那就需要模拟乌龟和兔子各自的行走轨迹，再绘制折线图，设置图表的格式等。  **4、【科学坐标图绘制：根据给定的数据绘制阻尼衰减曲线图】**    所谓阻尼衰减最简单的就是带阻尼的弹簧震荡衰减，这在我们自动化学院的专业核心课程《自动控制原理》就有详细的说明，现在就尝试用Python的numpy库结合matplotlib库进行绘制。  生成等间距离散的点，形成x轴，再用一个表达式表示y轴，就是我们的阻尼衰减曲线，最后使用pyplot绘出即可。  **5、【多级雷达图绘制：根据给定的任务能力数据，绘制多级雷达图】**    雷达图就像一张蜘蛛网，它对于数据的表示以及可视化有很好的表现，但是其绘制过程也相对于比较复杂，并且雷达图运用在两个及两个以上的对比数据中有很直观的展现。  这里我们对两位同学大学里面的五门课程进行雷达图显示，“高等数学”“大学物理”“大学体育”“C语言”“线性代数”，以此用雷达图直观的来进行比较。  **6、【在sqlite3数据库设计】**    在“Student.db”中，新建一张名为“userinfo”的表，并插入以下记录后，将王五的Email修改更新为Wangwu@163.com后，将所有的记录打印输出。  SQLite是一种嵌入式关系型数据库管理系统（RDBMS），它被设计成轻量级、快速且易于使用。SQLite不像传统的客户端/服务器数据库系统，它将整个数据库作为一个文件存储在主机文件系统中。SQLite的应用范围广泛，特别适合小规模项目、个人应用和移动应用等场景。它可以与多种编程语言集成，并提供了丰富的API，如C/C++、Python、Java等。无需独立的数据库服务器，SQLite可以直接在应用程序内部进行数据存储和管理，使得数据访问更加高效和便捷。  数据库的基本操作：增删改查，在sqlite3中都能实现。 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **四、实验结果及分析讨论（含原始数据记录等）**   1. **【使用KNN分类算法实现根据身高和体重对体型分类】**      1. # 使用KNN分类算法实现根据（身高和体重）对体型进行分类 3. **from** sklearn.neighbors **import** KNeighborsClassifier    # 导入KNN分类算法包 5. My\_tall\_weight\_train = [ 6. [180, 50], [180, 65], [180, 90], [180, 95], [180, 55], 7. [168, 57], [176, 65], [163, 48], [182, 80], [159, 51], 8. [175, 90], [165, 50], [185, 100], [160, 45], [180, 95], 9. [170, 55], [190, 105], [155, 40], [176, 85], [162, 48], 10. [172, 60], [195, 110], [158, 42], [182, 94], [168, 52], 11. [178, 80], [150, 35], [188, 98], [157, 43], [173, 68], 12. [165, 37], [188, 70], [175, 80], [183, 88], [162, 45], 13. [176, 75], [168, 58], [158, 41], [179, 82], [170, 50], 14. [187, 97], [152, 38], [169, 62], [181, 92], [172, 56], 15. [191, 103], [157, 44], [174, 78], [164, 49], [184, 90] 16. ] 18. My\_tall\_weight\_biaoqian\_train = [ 19. '偏瘦', '标准', '偏胖', '偏胖', '偏瘦', 20. '标准', '标准', '偏瘦', '偏胖', '偏瘦', 21. '偏胖', '偏瘦', '偏胖', '偏瘦', '偏胖', 22. '标准', '偏胖', '偏瘦', '偏胖', '偏瘦', 23. '标准', '偏胖', '偏瘦', '偏胖', '标准', 24. '偏胖', '偏瘦', '偏胖', '偏瘦', '标准', 25. '偏瘦', '标准', '标准', '偏胖', '偏瘦', 26. '标准', '偏瘦', '偏胖', '偏瘦', '标准', 27. '偏胖', '偏瘦', '偏胖', '偏瘦', '偏胖', 28. '标准', '偏瘦', '偏胖', '偏瘦', '偏胖' 29. ] 31. # 实例化knn对象 32. knn = KNeighborsClassifier(n\_neighbors=5) 33. # n\_neighbors:就是k-NN的k的值，选取最近的k个点。 35. knn.fit(My\_tall\_weight\_train, My\_tall\_weight\_biaoqian\_train) 36. # knn.fit(x\_train.y\_train)使用k-近邻算法对训练数据集x\_train和对应的标签y\_rain进行拟合。 38. My\_test = [[172, 40], [172, 50], [172, 60]] 40. My\_pre = knn.predict(My\_test) 41. **print**(My\_pre)     对于使用KNN算法对身高和体重进行分类，这里我列举了5x10一共50人的身高和体重，每一位的身高体重给他定义了一个标签“偏瘦”“标准”“偏胖”，于是得到了数据集然后调用sklearn.neighbors里面的KneighborsClassifier分类器，实例化后训练数据，得到模型后进行测试预测，分别输入三个测试人的身高和体重，观察预测效果，[172,40][172,50][172,60] ，预测为['偏瘦' '标准' '偏胖']，符合实验预期，实验结果正确。  **2、【使用k-means聚类算法进行分类】**     1. # k-means聚类算法进行分类  （可以沿用第一题的身高体重数据 | 或者是手写数据集MINIST） 3. **from** sklearn.cluster **import** KMeans 4. **import** matplotlib.pyplot as plt 5. # plt绘图显示中文名 6. plt.rcParams['font.sans-serif'] = ['SimHei'] 8. # 定义身高体重数据 9. My\_tall\_weight\_train = [ 10. [180, 50], [180, 65], [180, 90], [180, 95], [180, 55], 11. [168, 57], [176, 65], [163, 48], [182, 80], [159, 51], 12. [175, 90], [165, 50], [185, 100], [160, 45], [180, 95], 13. [170, 55], [190, 105], [155, 40], [176, 85], [162, 48], 14. [172, 60], [195, 110], [158, 42], [182, 94], [168, 52], 15. [178, 80], [150, 35], [188, 98], [157, 43], [173, 68], 16. [165, 37], [188, 70], [175, 80], [183, 88], [162, 45], 17. [176, 75], [168, 58], [158, 41], [179, 82], [170, 50], 18. [187, 97], [152, 38], [169, 62], [181, 92], [172, 56], 19. [191, 103], [157, 44], [174, 78], [164, 49], [184, 90] 20. ] 22. My\_kmeans = KMeans(n\_clusters=3) 23. # n\_clusters 生成的聚类数 25. My\_kmeans.fit(My\_tall\_weight\_train)    # 训练 27. cls1 = [My\_tall\_weight\_train[i] **for** i **in** range(len(My\_kmeans.labels\_)) **if** My\_kmeans.labels\_[i] == 0] 28. cls2 = [My\_tall\_weight\_train[i] **for** i **in** range(len(My\_kmeans.labels\_)) **if** My\_kmeans.labels\_[i] == 1] 29. cls3 = [My\_tall\_weight\_train[i] **for** i **in** range(len(My\_kmeans.labels\_)) **if** My\_kmeans.labels\_[i] == 2] 31. plt.scatter([x[0] **for** x **in** cls1], [y[1] **for** y **in** cls1], color = 'red') 32. plt.scatter([x[0] **for** x **in** cls2], [y[1] **for** y **in** cls2], color = 'blue') 33. plt.scatter([x[0] **for** x **in** cls3], [y[1] **for** y **in** cls3], color = 'green') 35. plt.xlabel('身高') 36. plt.ylabel('体重') 37. plt.title("使用K-Means进行聚类") 38. plt.show()     沿用第一题的身高体重数据,对身高和体重进行聚类，这里就是需要去了解学习sklearn.cluster 库里面的 Kmeans包，就是专门用来进行k-means聚类算法进行分类。  使用Kmeans算法训练数据集，之后遍历循环得到分类后的不同类的点，将同一类的点赋予相同颜色，最后使用matplotlib.pyplot进行图像可视化。如图所示，使用K-means聚类成了3类，红色类绿色类，蓝色类，实验结果与预期相符，实验结果正确。  **3、【使用matplotlib绘制折线图对龟兔赛跑中兔子和乌龟的行走轨迹进行可视化】**     1. # 使用matplotlib绘制折现图，对龟兔赛跑中兔子和乌龟行走轨迹进行可视化 3. **import** matplotlib.pyplot as plt 4. # plt绘图显示中文名 5. plt.rcParams['font.sans-serif'] = ['SimHei'] 7. # 乌龟行走轨迹 8. turtle\_path = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16] 9. # 兔子行走轨迹 10. rabbit\_path = [0, 2, 4, 6, 8, 8, 8, 8, 8, 8, 8, 8, 8, 10, 12, 14, 16] 12. # 绘制折线图 13. plt.plot(turtle\_path, label='乌龟')  # 绘制乌龟行走轨迹 14. plt.plot(rabbit\_path, label='兔子')  # 绘制兔子行走轨迹 16. # 设置图表标题、坐标轴名称、图例等属性 17. plt.title('---龟兔赛跑轨迹---') 18. plt.xlabel('时间（s）') 19. plt.ylabel('距离（m）') 20. plt.legend()    # 线例 22. # 显示图形 23. plt.show()     可以观察到如图，兔子先是快速跑了一段距离，发现远远超过乌龟后停止运动休息起来，发现被乌龟超过后，再跑，可是乌龟一直都在很努力的匀速直线运动，坚持不懈一直到达终点。  实验结果与预期相符，实验结果正确。  **4、【科学坐标图绘制：根据给定的数据绘制阻尼衰减曲线图】**     1. # 科学坐标图绘制：根据给定的数据绘制阻尼衰减曲线图 3. **import** numpy as np 4. **import** matplotlib.pyplot as plt 5. pi = np.pi 7. # 生成离散的点 8. My\_x = np.linspace(0, 10, 100) 10. # 阻尼衰减曲线 11. My\_y = np.exp( - My\_x) \* np.cos(2 \* pi \* My\_x) 12. **print**(My\_y) 14. plt.plot(My\_x, My\_y) 15. plt.show()     如图生成了震荡衰减曲线和《自动控制原理》所学知识相符合，实验结果与预期相符，实验结果正确。  **5、【多级雷达图绘制：根据给定的任务能力数据，绘制多级雷达图】**     1. # 多级雷达图绘制：根据给定任务能力数据，绘制多级雷达图 3. **import** numpy as np 4. **import** matplotlib.pyplot as plt 5. pi = np.pi 6. # plt绘图显示中文名 7. plt.rcParams['font.sans-serif'] = ['SimHei'] 9. # 使用字典存储 10. My\_Score = [{'高等数学': 93, '大学物理': 80, '大学体育': 60, 'C语言': 98, '线性代数': 85}, 11. {'高等数学': 80, '大学物理': 65, '大学体育': 90, 'C语言': 80, '线性代数': 85}] 12. data\_len = len(My\_Score[0])    # 五维度 13. **print**(data\_len) 15. angles = np.linspace(0, 2 \* pi, data\_len, endpoint=False)    # 把360度分成5个，不包括最后值 17. lab = [key **for** key **in** My\_Score[0].keys()]    # 获取五个维度的标签 18. **print**(lab) 19. My\_score\_evey = [[v **for** v **in** socre.values()] **for** socre **in** My\_Score]    # 获取各课成绩 20. **print**(My\_score\_evey) 22. My\_score\_evey\_1 = np.concatenate((My\_score\_evey[0], [My\_score\_evey[0][0]]))    # 获取第一个 23. **print**(My\_score\_evey\_1) 25. My\_score\_evey\_2 = np.concatenate((My\_score\_evey[1], [My\_score\_evey[1][0]]))    # 获取第二个 26. **print**(My\_score\_evey\_2) 28. angles = np.concatenate((angles, [angles[0]])) 29. lab = np.concatenate((lab, [lab[0]])) 30. **print**(angles) 31. **print**(lab) 33. My\_fig = plt.figure(figsize=(8, 6), dpi=100) 34. ax = plt.subplot(111, polar=True) 36. ax.plot(angles, My\_score\_evey\_1, color='red') 37. ax.plot(angles, My\_score\_evey\_2, color='blue') 39. ax.set\_thetagrids(angles \* 180 / pi, lab) 40. ax.set\_theta\_zero\_location('N') 42. ax.set\_rlabel\_position(270) 44. plt.show()     根据雷达图，我们可以看到两位同学的“高等数学”“大学物理”“大学体育”“C语言”“线性代数”成绩对比情况。直观且清除的看到了雷达效果，实验结果与预期相符，实验结果正确。  **6、【在sqlite3数据库设计】**     1. # 在sqlite3数据库“Student.db”中，新建一张名为“userinfo”的表，并插入以下记录后， 2. # 将王五的Email修改更新为Wangwu@163.com后，将所有的记录打印输出。 3. **import** sqlite3 5. conn = sqlite3.connect('Student.db') 6. cur = conn.cursor() 8. cur.execute("DROP TABLE IF EXISTS userinfo") 10. sql\_text\_1 = '''''CREATE TABLE userinfo 11. ( 12. StuNumber NUMBER, 13. Name TEXT, 14. ClassNumber TEXT, 15. Email TEXT 16. ) 17. ''' 18. cur.execute(sql\_text\_1) 20. sql\_text\_2 = "INSERT INTO userinfo VALUES (20190001,'张三', 'C01', 'Zhangsan@163.com')" 21. cur.execute(sql\_text\_2) 23. data = [ 24. (20190002, '李四', 'C02', 'Lisi@163.com'), 25. (20190003, '王五', 'C03', 'Wangwu@qq.com'), 26. (20190004, '小明', 'C04', 'Xiaoming@qq.com') 27. ] 28. cur.executemany('INSERT INTO userinfo VALUES(?,?,?,?)', data) 29. conn.commit() 31. cur.execute('SELECT \* FROM userinfo') 32. rows = cur.fetchall() 33. **for** row **in** rows: 34. **print**(row) 36. **print**('------------------------------------------') 37. cur.execute("UPDATE userinfo SET Email = 'Wangwu@163.com' WHERE Name = '王五'") 38. cur.execute('SELECT \* FROM userinfo') 39. rows = cur.fetchall() 40. **for** row **in** rows: 41. **print**(row)     可以看到在目录下生成了Student.db数据库，并且在终端显示以下结果：  (20190001, '张三', 'C01', 'Zhangsan@163.com')  (20190002, '李四', 'C02', 'Lisi@163.com')  (20190003, '王五', 'C03', 'Wangwu@qq.com')  (20190004, '小明', 'C04', 'Xiaoming@qq.com')  ------------------------------------------  (20190001, '张三', 'C01', 'Zhangsan@163.com')  (20190002, '李四', 'C02', 'Lisi@163.com')  (20190003, '王五', 'C03', 'Wangwu@163.com')  (20190004, '小明', 'C04', 'Xiaoming@qq.com')  与实验要求：  在sqlite3数据库“Student.db”中，新建一张名为“userinfo”的表，并插入以下记录后，将王五的Email修改更新为Wangwu@163.com后，将所有的记录打印输出。   |  |  |  |  | | --- | --- | --- | --- | | StuNumber | Name | ClassNumber | Email | | 20190001 | 张三 | C01 | Zhangsan@163.com | | 20190002 | 李四 | C02 | Lisi@163.com | | 20190003 | 王五 | C03 | Wangwu@qq.com | | 20190004 | 小明 | C04 | Xiaoming@qq.com |   一致，实验结果符合预期，实验结果正确。 |

|  |
| --- |
| **五、实验测验总结与体会**  问题一：学会了KNN算法，领悟了sklearn包的作用，对于分类上的经典算法有了更深入的了解，训练数据集，测试数据集，加强了Python在数据处理与分类上的运用。  问题二：学会了K-Means算法，同样运用在聚类分类上的巨大作用，训练再测试，加强了Python的运用。  问题三：学会了Python中重要的matplotlib绘图包的运用，绘制折线图，运用到实际。  问题四：学会了科学坐标图的绘制，运用matplotlib绘图包结合我们所学知识《自动控制控制原理》绘制了阻尼衰减曲线。  问题五：学会了matplotlib绘图包中的雷达图绘制，雷达图可以清楚直观的对比两组数据。  问题六：学会了一种嵌入式关系型数据库SQLite，在Python中运用sqlite3实现了简单的增删改查操作，对数据库进行实际操作。  致谢：  这是最后一次Python实验课程了，感谢Python实验课程张老师的引导与授予知识，感谢Python理论课程韩老师的孜孜教诲，Python真是一门推开新世界的大门，感谢老师们的教导！  我关于Python学习过程的博客也在不断更新中，感谢老师们授予的知识：https://blog.csdn.net/Akaxi1?type=blog |
| **六、评阅意见**  评阅人签字：  评阅日期： |