

**实 验 报 告**

**（ 2024 / 2025 学年 第 一 学期 ）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | Python程序设计（混合式）（JS170101S） | | | | | |
| 实验名称 | 实验三：K-近邻分类与K-均值聚类实验 | | | | | |
| 实验时间 | 2024 | 年 | 11 | 月 | 29 | 日 |
| 指导单位 | 计算机学院、软件学院、网络空间安全学院 | | | | | |
| 指导教师 | 杨尚东 | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 学生姓名 | 袁浩宽 | 班级学号 | B23042424 |
| 学院(系) | 计算机学院、软件学院、网络空间安全学院 | 专 业 | 数据科学与大数据技术 |

| **实 验 报 告** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **实验名称** | **K-近邻分类与K-均值聚类实验** | | | **指导教师** | 杨尚东 |
| **实验类型** | **验证** | **实验学时** | **2** | **实验时间** | 2学时 |
| 1. **实验目的和要求**    1. 掌握K-近邻分类的基本原理。    2. 学会使用Python语言实现K-近邻分类算法。    3. 掌握K-均值聚类的基本原理。    4. 学会使用Python语言实现K-均值聚类算法。 | | | | | |
| 二、**实验环境(实验设备)**  硬件: 微型计算机  软件: Windows / MacOS / Linux 操作系统、Python3.7或3.8 | | | | | |
| 1. **实验原理及内容** 2. 编写代码，使用K近邻分类算法对训练数据（实验二题目2训练数据.txt）进行模型训练，计算在验证集（训练数据后20%）上的正确率、精准率和召回率。  * 数据集说明   该数据集需要通过特征数据分析，来预测二分类的结果。训练数据集中的二分类结果已经给出，为最后一列   * 数据集信息   本数据集用于估计信用卡的违约概率，训练数据包括27997个样本。   * 数据属性   与实验二相同  要求：  （1）使用sklearn实现  （2）自定义编码实现，程序运行时间角度与（1）进行比较。（选做）  from sklearn.neighbors import KNeighborsClassifier  from sklearn.metrics import accuracy\_score, precision\_score, recall\_score  from sklearn.model\_selection import train\_test\_split  import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  train\_data = pd.read\_csv('data/experiment\_2/题目2训练数据.txt', header=None, delim\_whitespace=True)  test\_data = pd.read\_csv('data/experiment\_2/题目2测试数据.txt',header=None,delim\_whitespace=True)  X = train\_data.iloc[:, :-1]  y = train\_data.iloc[:, -1]  X\_test = test\_data.iloc[:,:]  X\_train, X\_val, y\_train, y\_val = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  model = KNeighborsClassifier(n\_neighbors=2,algorithm='brute',weights='distance')  model.fit(X\_train, y\_train)  y\_val\_pred = model.predict(X\_val)  accuracy = accuracy\_score(y\_val, y\_val\_pred)  precision = precision\_score(y\_val, y\_val\_pred)  recall = recall\_score(y\_val, y\_val\_pred)  print(**f**'Accuracy: {accuracy**:.4f**}')  print(**f**'Precision: {precision**:.4f**}')  print(**f**'Recall: {recall**:.4f**}')  y\_test = model.predict(X\_test).T  np.savetxt('result/prediction3-1.txt', y\_test, fmt='%.2f')  print(X\_test)  print(y\_test)  d = np.concatenate((X\_test, y\_test.reshape(-1, 1)), axis=1)  d1 = np.array([i for i in d if i[-1] == 1])  d0 = np.array([i for i in d if i[-1] == 0])  xfeature, yfeature = 0, 13  plt.figure(figsize=(8, 6))  plt.scatter(d1[:,xfeature],d1[:,yfeature],label='y = 1',alpha=0.5,s=1)  plt.scatter(d0[:,xfeature],d0[:,yfeature],label='y = 0',alpha=0.5,s=1)  plt.xlabel(**f**'feature {xfeature}')  plt.ylabel(**f**'feature {yfeature}')  plt.legend()  plt.title(**f**'visualization feature {xfeature} & feature {yfeature}')  plt.savefig("img/problem3-1.png")  plt.show()  ’’’  Accuracy: 0.6913  Precision: 0.2970  Recall: 0.2821  RESULT:    ’’’   1. 编写代码，利用K-均值聚类和PCA降维（from sklearn.decomposition import PCA，请在网络检索使用方法）方法，实现对书上手写数字数据集“arab\_digits\_training.txt”的第2-785列特征数据进行聚类，聚类中心数量选10。   要求：  （1）使用sklearn实现，  （2）使用matplotlib，在PCA降维后的空间使用不同颜色画出每个类别的样本，并用该类别中类标出现次数最多的类别标注出该类的聚类中心。（选做）  import numpy as np  import matplotlib.pyplot as plt  from sklearn.decomposition import PCA  *# from scipy.stats import mode*  **def** distance(x1, x2):      return np.sqrt(np.sum((x1 - x2) \*\* 2))  **def** classify\_one\_example(x, centers):      min\_i = 0      min\_dist = float('inf')      for i in range(len(centers)):          center = centers[i]          dist = distance(x, center)          if dist < min\_dist:              min\_i = i              min\_dist = dist      return min\_i, min\_dist  **def** re\_classify\_examples(X, example\_bags, centers):      for example\_bag in example\_bags:          example\_bag.clear()      new\_cost = 0      for x in X:          index, dist = classify\_one\_example(x, centers)          example\_bags[index].append(x)          new\_cost += dist      return new\_cost / len(X)  **def** cal\_centers(example\_bags, centers):      for i in range(len(example\_bags)):          if len(example\_bags[i]) > 0:              centers[i] = np.mean(example\_bags[i], axis=0)  **def** kmeans(X, K, max\_iter=5000, tol=1e-20):      centers = np.array([X[i] for i in range(K)])      example\_bags = [[] for \_ in range(K)]      cost = float('inf')      for i in range(max\_iter):          new\_cost = re\_classify\_examples(X, example\_bags, centers)          cal\_centers(example\_bags, centers)            if i % 10 == 0:              print(**f**"Iteration {i + 1}, Cost: {new\_cost**:.4f**}")          if np.abs(new\_cost - cost) < tol:              print(**f**"Stopped at iteration {i + 1}")              break          cost = new\_cost      return centers, example\_bags  **def** accuracy(labels\_true, labels\_pred):      return np.sum(labels\_true == labels\_pred) / len(labels\_true)  data = np.loadtxt("data/number\_data/arab\_digits\_training.txt", delimiter="\t")  X = data[:, 1:]  y = data[:, 0]  pca = PCA(n\_components=2)  X\_pca = pca.fit\_transform(X)  K = 10  centers, example\_bags = kmeans(X\_pca, K)  labels\_pred = np.array([np.argmin([distance(x, center) for center in centers]) for x in X\_pca])  acc = accuracy(y, labels\_pred)  print(**f**"Accuracy: {acc**:.4f**}")  plt.figure(figsize=(8, 6))  for i in range(K):      example\_bags[i] = np.array(example\_bags[i])      plt.scatter(example\_bags[i][:, 0], example\_bags[i][:, 1], label=**f**'Cluster {i}', s=5)  centers = np.array(centers)  plt.scatter(centers[:, 0], centers[:, 1], c='black', marker='+', s=200, label='Centroids')  plt.title('K-Means Clustering on PCA-reduced Data', fontsize=16)  plt.xlabel('Principal Component 1', fontsize=12)  plt.ylabel('Principal Component 2', fontsize=12)  plt.legend()  plt.grid(True)  plt.savefig("img/problem3-2.png")  plt.show()  ’’’  RESULT:    ’’’ | | | | | |
| **四、实验小结**（包括问题和解决方法、心得体会、意见与建议、实验出错信息及解决方案等）  (一)实验中遇到的主要问题及解决方法  在进行K-均值聚类时，由于K值的选择和初始化聚类中心可能影响结果的稳定性，初期聚类结果并不理想。通过调整K值和改进算法中的聚类中心更新策略，最终得到较为合理的聚类结果。在此过程中，利用PCA对数据进行降维，可视化聚类效果，帮助我们更直观地理解各类数据点的分布情况。  （二）实验心得  通过这次实验，我对K-近邻分类和K-均值聚类算法有了更加深入的理解，尤其是在实际应用中如何结合PCA等降维技术来优化数据处理和可视化。同时，在手写数字数据集的聚类实验中，投票机制在确认每个聚类中心代表的数字类别时起到了至关重要的作用，使得结果更加符合实际分类标准。  在实验过程中，我深刻体会到数据预处理和算法参数调优对实验结果的重要性。尤其是聚类算法，由于其无监督性质，聚类结果容易受到初始条件和迭代次数的影响，因此需要更加细致的调参与验证。  总体而言，本次实验不仅加深了我对K-近邻和K-均值聚类算法的理解，也使我认识到无监督学习和监督学习之间的区别，以及如何在具体任务中选择合适的评价指标和优化方法。  （三）意见与建议（没有可省略）  建议可以让实验报告成为一个开放式的课题，但是有一个baseline，比如限定使用基本的算法之类，然后让所有同学在此基础上进行优化。根据最终的算法的跑出效果，可视化效果，以及前期的数据分析处理水平经行一定的打分。 | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. **支撑毕业要求指标点**   （一）思想、素质教育目标  目标1.1 在教学过程中，激发学生自豪感与爱国情怀，鼓励学生通过努力学习掌握先进科学技术，服务国家，回馈社会。  目标1.2 通过Python编程、人工智能主流算法等知识的学习，增强学生的科技强国意识，**帮助其树立正确的价值观。**  （二）知识教学目标  目标2.1 使学生掌握Python语言编程。  目标2.2 使学生掌握人工智能基本算法及Python语言实现方法。  （三）能力教学目标  目标3.1 使学生掌握Python语言编程方法，具有使用Python语言进行编程的能力。  目标3.2 使学生掌握线性回归、逻辑斯蒂分类、K-近邻分类、K-均值聚类、神经网络等人工智能基本算法，使学生具有使用Python语言实现人工智能基本算法的能力，对实际应用中的问题进行设计研究方案的能力。 | | | | | | |
| **六、指导教师评语** | | | | | | |
| **评**  **分**  **细**  **则** | **评分项** | **优秀** | **良好** | **中等** | **合格** | **不合格** |
| **遵守实验室规章制度** |  |  |  |  |  |
| **学习态度** |  |  |  |  |  |
| **算法思想准备情况** |  |  |  |  |  |
| **程序设计能力** |  |  |  |  |  |
| **解决问题能力** |  |  |  |  |  |
| **课题功能实现情况** |  |  |  |  |  |
| **算法设计合理性** |  |  |  |  |  |
| **算法效能评价** |  |  |  |  |  |
| **回答问题准确度** |  |  |  |  |  |
| **报告书写认真程度** |  |  |  |  |  |
| **内容详实程度** |  |  |  |  |  |
| **文字表达熟练程度** |  |  |  |  |  |
| **其它评价意见** |  | | | | |
| **本次实验能力达成评价（总成绩）** |  | **批阅人** |  | **日期** |  |