|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 姓名： 张晨龙 | | 学号：1925050069 | 专业年级：2019 | 班级：软件工程1班 |
| 分组： 无 | | 实验室： 曾宪梓楼503 | 指导教师：李霞 | 实验日期：2022.3.21 |
| **实验的准备阶段** | **课程名称** | 机器学习及与智能数据处理 | | |
| **实验名称** | KNN算法分类及预测 | | |
| **实验目的** | 1. 掌握最KNN算法的数学原理及推导过程 2. 编程实现KNN算法，并应用于具体案例 | | |
| **实验内容** | 1、编写代码，实现对iris数据集的KNN算法分类及预测，要求：  （1）数据集划分为测试集占20%；  （2）n\_neighbors=5；  （3）评价模型的准确率；  （4）使用模型预测未知种类的鸢尾花。  2、改进模型，要求：  （1）数据集划分采用10折交叉验证；  （2）寻找最优的n\_neighbors值（在5-10之间）；  （3）使用新的模型预测未知种类的鸢尾花。  待预测未知数据：  X1=[[1.5 , 3 , 5.8 , 2.2], [6.2 , 2.9 , 4.3 , 1.3]] | | |
| **实验类型**  （打☑） | □验证性 □演示性 🗹设计性 □综合性 | | |
| **实验的重点、难点** | KNN算法总n\_neighbors值的寻找 | | |
| **实验环境** | Python 3.6  Numpy | | |
| **实验的实施阶段** | **实验步骤及实验结果** | 1. **编写代码，实现对iris数据集的KNN算法分类及预测**   *# 导入数据集和模型所需的包* **from** sklearn.datasets **import** load\_iris **from** sklearn.neighbors **import** KNeighborsClassifier **as** KNN **from** sklearn.model\_selection **import** train\_test\_split   **def** main():  *# 加载鸢尾花数据集的数据和标签* X, y = load\_iris(return\_X\_y=**True**)  *# 按照 训练集: 测试集 = 8: 2划分数据集* X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y,test\_size=0.2)  *# 定义模型* model = KNN(n\_neighbors=5)  *# 用训练集拟合模型* model.fit(X\_train,y\_train)  *# 得到测试集的准确率* test\_acc = model.score(X\_test,y\_test)  print(**"鸢尾花测试集中,模型Accuracy 为： %f"** % (test\_acc))  *# 加载预测的数据X1* X1 = [[1.5, 3, 5.8, 2.2], [6.2, 2.9, 4.3, 1.3]]  *# 用模型进行预测并得到预测结果* predict = model.predict(X1)  *# 获得X1中两个鸢尾花预测得到的索引所对应的类别* X1\_classes = [load\_iris().target\_names[idx] **for** idx **in** predict]  print(**"两个鸢尾花的预测结果为：[1.5, 3, 5.8, 2.2] ：{0}, [6.2, 2.9, 4.3, 1.3] ：{1}"**.format(\*X1\_classes))   **if** \_\_name\_\_ == **'\_\_main\_\_'**:  main()  **实验1结果：**     1. **改进模型进行预测：**   *# 导入数据集和模型所需的包* **from** sklearn.datasets **import** load\_iris **from** sklearn.neighbors **import** KNeighborsClassifier **as** KNN **from** sklearn.model\_selection **import** cross\_val\_score *# 定义每次迭代的训练函数* **def** train\_one\_epoch(X,y,model):  model.fit(X, y)  scores = cross\_val\_score(model, X, y, cv=10).mean()  **return** scores  **def** main():  *# 定义迭代次数和超参数n\_neighbors* epochs = 6  n\_neighbors = 5  *# 加载鸢尾花数据集的数据和标签* X, y = load\_iris(return\_X\_y=**True**)  *# 定义用于记录最佳准确率、最佳模型和最佳n\_neighbors的参数* best\_score = 0  best\_model = **None** best\_n = 5  *# 进行迭代* **for** iter **in** range(epochs):  *# 定义当前的KNN模型* cur\_model = KNN(n\_neighbors=n\_neighbors)  *# 训练模型* cur\_score = train\_one\_epoch(X,y,cur\_model)  *# 判断得分是否超过最佳得分， 如果是则替换最佳模型* **if** cur\_score > best\_score:  best\_n = n\_neighbors  best\_model = cur\_model  best\_score = cur\_score  print(**"epoch {0}, 目前的 n\_neighbors {1}, 10折交叉验证的平均准确率为{2}"**.format(iter+1, n\_neighbors, cur\_score))  *# 每次迭代n\_neighbors += 1* n\_neighbors += 1  print(**"迭代结束"**)  print(**"最佳模型的n\_neighbors 为 {0}, 10折交叉验证的平均准确率为 {1}"**.format(best\_n,best\_score))  *# 定义用于测试的数据* X1 = [[1.5, 3, 5.8, 2.2], [6.2, 2.9, 4.3, 1.3]]  *# 用最佳模型进行预测并得到预测结果* predict = best\_model.predict(X1)  *# 获得X1中两个鸢尾花预测得到的索引所对应的类别* X1\_classes = [load\_iris().target\_names[idx] **for** idx **in** predict]  print(**"两个鸢尾花的预测结果为：[1.5, 3, 5.8, 2.2] ：{0}, [6.2, 2.9, 4.3, 1.3] ：{1}"**.format(\*X1\_classes))   **if** \_\_name\_\_ == **'\_\_main\_\_'**:  main()  **实验2结果：** | | |
| **实验结果的处理阶段** | **实验结果的分析与总结** | 通过本次实验我学会了使用KNearestNeighbors 算法来进行分类和预测。学会了算法的本质原理和训练过程，并在课程中学习了使用sklearn.preprocessing 包来进行简单的数据预处理的操作。通过实验找到了鸢尾花数据集中的最佳n\_neighbors参数，模型取得了较好的测试结果。学会了使用K折交叉验证划分数据集，从而验证模型的性能。 | | |