四 川 大 学 计 算 机 学 院、软 件 学 院

实 验 报 告

学号：2022141460176 姓名：杨一舟 专业：计算机科学与技术

课程名称	数据挖掘导引	实验课时	
实验项目	使用 KNN 算法进行数据挖掘实验	实验时间	2024.5.20
实验目的	基于 IRIS 数据集通过编程实现 KNN 算法的实践		
实验环境	Visual studio Code		
实验内容（算法、程序、步骤和方法）	**一、实验步骤** KNN（K-Nearest Neighbors）算法实验步骤： **数据准备：** 收集带有标签的训练样本数据集，其中每个样本都有对应的特征和标签，本实验中选用 IRIS 数据集。 **选择 K 值：** 确定要考虑的邻居数量 K，通常通过交叉验证来选择最佳的 K 值。本实验中选用 K 值为 3 **计算距离：** 对于要预测的每个未知样本，计算其与训练集中所有已知样本的距离。常用的距离度量包括欧氏距离、曼哈顿距离等。 **选择邻居：** 根据距离选择与未知样本最近的 K 个邻居。		

进行投票：

对于选定的 K 个邻居，根据其标签进行投票，选择出现次数最多的类别作为未知样本的预测类别。

评估模型：

使用测试集评估模型的性能，计算准确率、召回率、F1 分数等指标。

优化和调整：

根据评估结果调整 K 值或进行其他优化操作，以提高模型性能。

二、实验源代码

```
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix

# 加载 iris 数据集
iris = datasets.load_iris()
X = iris.data
y = iris.target
feature_names = iris.feature_names
target_names = iris.target_names

# 划分数据集为训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 数据标准化
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
```

```
X_test = scaler.transform(X_test)


# 初始化 KNN 分类器，这里我们设定 K=3
knn = KNeighborsClassifier(n_neighbors=3)


# 训练模型
knn.fit(X_train, y_train)


# 预测测试集
y_pred = knn.predict(X_test)


# 打印分类报告和混淆矩阵
print(classification_report(y_test, y_pred, target_names=target_names))
print(confusion_matrix(y_test, y_pred))


# 绘制两个特征之间的散点图以及决策边界
# 注意：KNN 没有明确的决策边界，但我们可以绘制测试集样本和它们的预测结果
plt.figure(figsize=(10, 8))


# 绘制训练集样本
for i, c, label in zip(range(3), ('red', 'green', 'blue'), target_names):
    plt.scatter(X_train[y_train == i, 0], X_train[y_train == i, 1],
                c=c, label=label, alpha=0.6, edgecolor='black')


# 绘制测试集样本并用不同颜色表示预测结果
for i, c, label in zip(range(3), ('orange', 'cyan', 'magenta'), target_names):
    plt.scatter(X_test[y_test == i, 0], X_test[y_test == i, 1],
                c=c, label=f'Predicted {label}', alpha=0.8, marker='s', edgecolor='black')


# 添加图例
plt.legend()
plt.xlabel(feature_names[0])
plt.ylabel(feature_names[1])
plt.title('K-NN Classification of Iris Dataset')
plt.show()
```

（接上）实验内容（算法、程序、步骤和方法）

三、实验结果

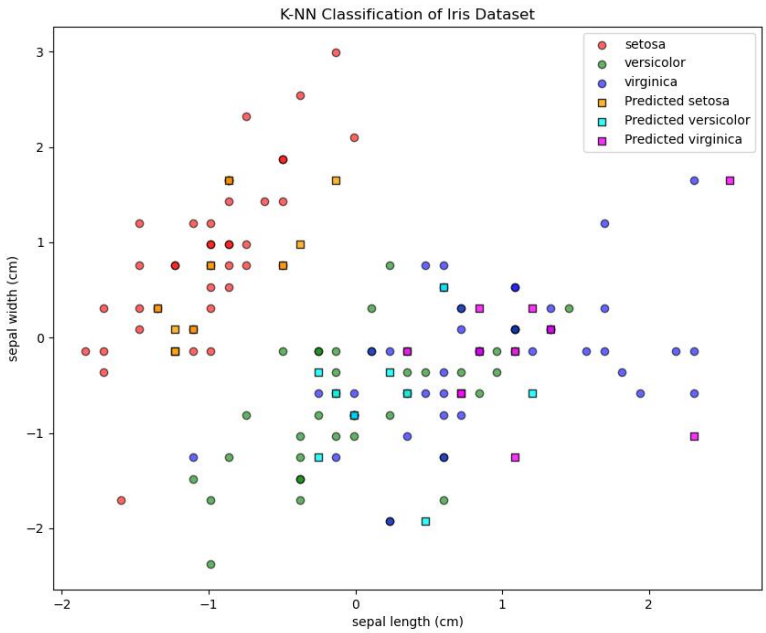
```
(mountaintorch) C:\Users\MountainMist\Desktop\2024春\数据挖掘\数据挖掘导引期末报告>python K近邻算法.py
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        10
  versicolor       1.00      1.00      1.00         9
   virginica       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30

[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
```

以上为分类报告和混淆矩阵



以上为结果可视化展示

数据记录和计算	实验结果如上述所示 结果分析： 本算法中将 IRIS 数据集分为了训练集与测试集两部分，其中圆点表示训练集中各个数据点依据 KNN 算法进行分类产生的不同类别，方点表示测试集中各个数据点的预测类别结果。
结　论（结　果）	成功完成了 KNN 算法的设计与实践
小　　结	在此次 KNN 算法实践中，我深入了解了其基于实例的学习方式和邻近性度量的重要性。KNN 算法简单直观，对于分类问题表现出色，尤其在数据分布明确且样本量较大的情况下。然而，它对于高维数据和不平衡数据集的处理能力有限，需结合实际情况进行选择和优化。
指导老师评　　议	成绩评定：　　指导教师签名：

实验报告说明

专业实验中心

实验名称　要用最简练的语言反映实验的内容。如验证某程序、定律、算法，可写成“验证×××”；分析×××。

实验目的　目的要明确，要抓住重点，可以从理论和实践两个方面考虑。在理论上，验证定理、公式、算法，并使实验者获得深刻和系统的理解，在实践上，掌握使用实验设备的技能技巧和程序的调试方法。一般需说明是验证型实验还是设计型实验，是创新型实验还是综合型实验。

实验环境　实验用的软硬件环境（配置）。

实验内容（算法、程序、步骤和方法）　这是实验报告极其重要的内容。这部分要写明依据何种原理、定律算法、或操作方法进行实验，要写明经过哪几个步骤。还应该画出流程图（实验装置的结构示意图），再配以相应的文字说明，这样既可以节省许多文字说明，又能使实验报告简明扼要，清楚明白。

数据记录和计算　指从实验中测出的数据以及计算结果。

结论（结果）　即根据实验过程中所见到的现象和测得的数据，作出结论。

小结　对本次实验的体会、思考和建议。

备注或说明　可写上实验成功或失败的原因，实验后的心得体会、建议等。

注意：

- 实验报告将记入实验成绩；
- 每次实验开始时，交上一次的实验报告，否则将扣除此次实验成绩。