实验报告（6）

学号：202202206341 姓名：王子俊

1. **实验名称： 支持向量机**
2. **实验目的**
3. 理解关联分析
4. 理解Apriori算法
5. 了解apyori库
6. 了解pandas库
7. **实验内容**

本实验使用 Iris 数据集，采用支持向量机（SVM）分类器进行模型训练。首先加载并可视化数据集，随后划分训练集和测试集，训练模型并进行预测，最后评估模型准确率和分类性能，输出混淆矩阵与分类报告，结果表明模型表现优秀。

1. **实验步骤**
2. 数据集准备：加载 Iris 数据集并进行基本数据可视化，分析数据集的信息。
3. 模型训练：使用 train\_test\_split 划分数据集，采用支持向量机（SVM）进行模型训练。
4. 模型预测：使用训练好的模型进行预测，并计算准确率。
5. 结果分析：通过混淆矩阵和分类报告分析模型性能。
6. **实验结果与分析**

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.datasets import load\_iris

# 读取iris数据集

iris = load\_iris()

data = pd.DataFrame(data=iris.data, columns=iris.feature\_names)

data['species'] = iris.target\_names[iris.target]

# 查看数据集的基本信息

print("数据集基本信息：")

print(data.info())

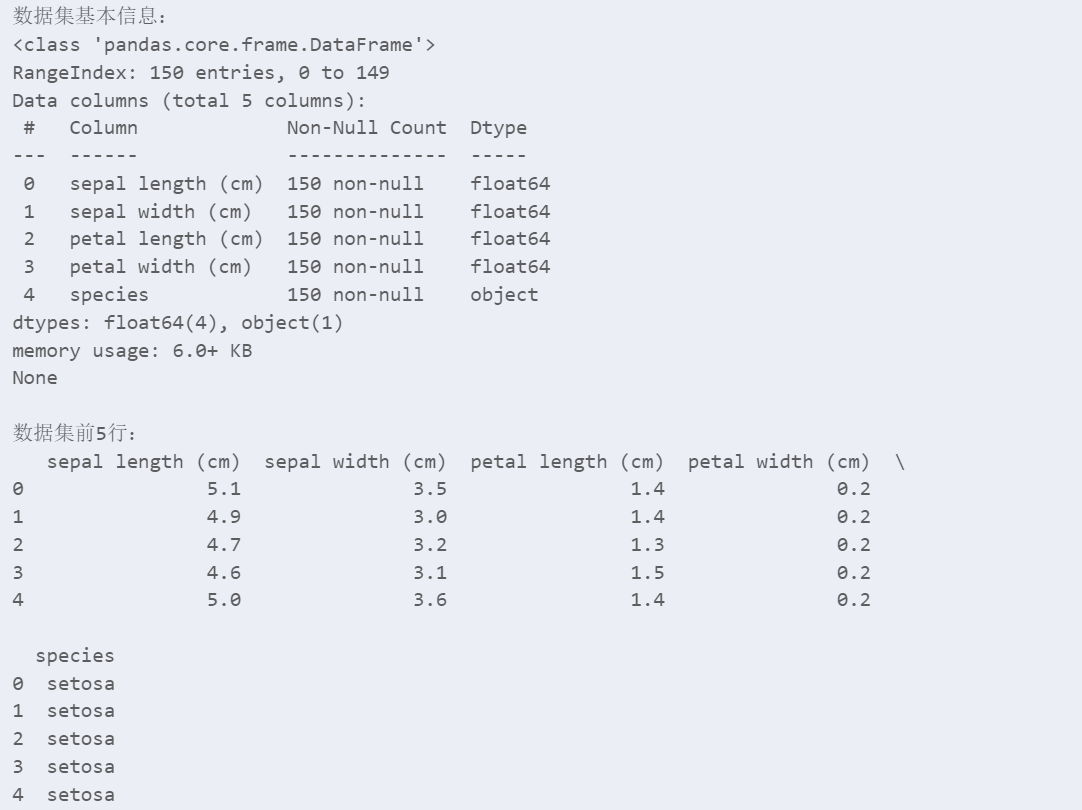
print("\n数据集前5行：")

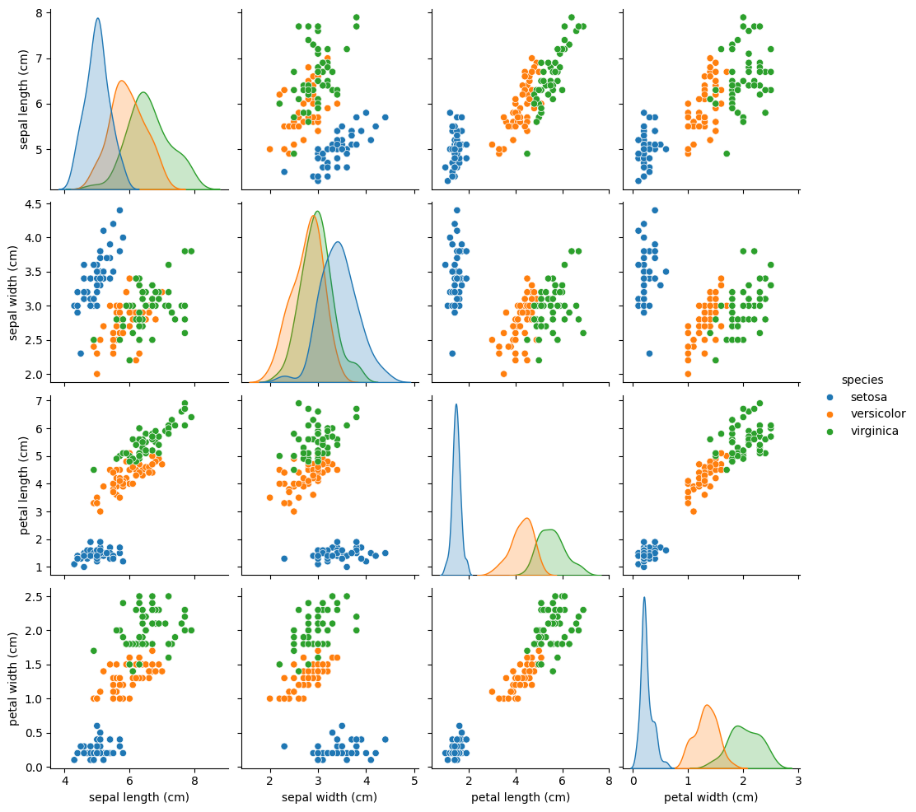
print(data.head())

# 数据可视化 - pairplot

sns.pairplot(data, hue='species')

plt.show()





from sklearn.model\_selection import train\_test\_split

from sklearn.svm import SVC

# 划分数据集，80%用于训练，20%用于测试

X = data.iloc[:, :-1]  # 特征

y = data['species']  # 标签

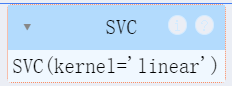
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=41)

# 初始化支持向量机分类器

model = SVC(kernel='linear')  # 使用线性核

# 训练模型

model.fit(X\_train, y\_train)



from sklearn**.**metrics import accuracy\_score

# *进行预测*

y\_pred = model**.**predict(X\_test)

# *评估模型准确性*

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"模型预测准确率：{accuracy:.2f}")



from sklearn**.**metrics import confusion\_matrix, classification\_report

# *输出混淆矩阵*

cm = confusion\_matrix(y\_test, y\_pred)

print("\n混淆矩阵：")

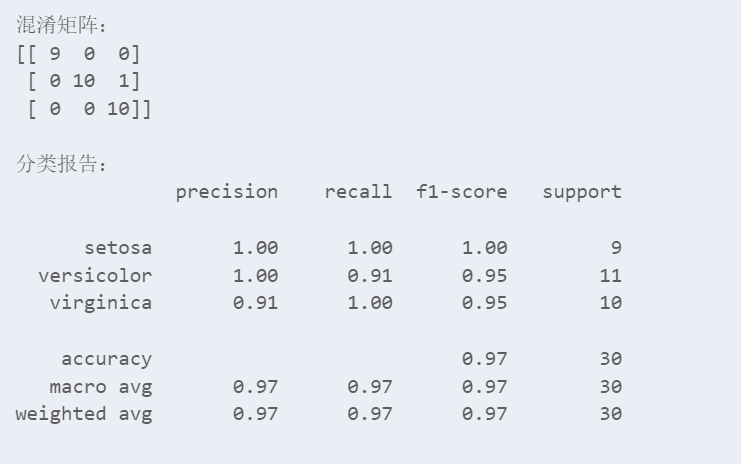
print(cm)

# *输出分类报告*

report = classification\_report(y\_test, y\_pred)

print("\n分类报告：")

print(report)



**高斯核训练**

# *训练模型*

model**.**fit(X\_train, y\_train)

# *模型预测*

y\_pred = model**.**predict(X\_test)

# *评估模型准确性*

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"模型预测准确率：{accuracy:.2f}")

# *结果分析*

cm = confusion\_matrix(y\_test, y\_pred)

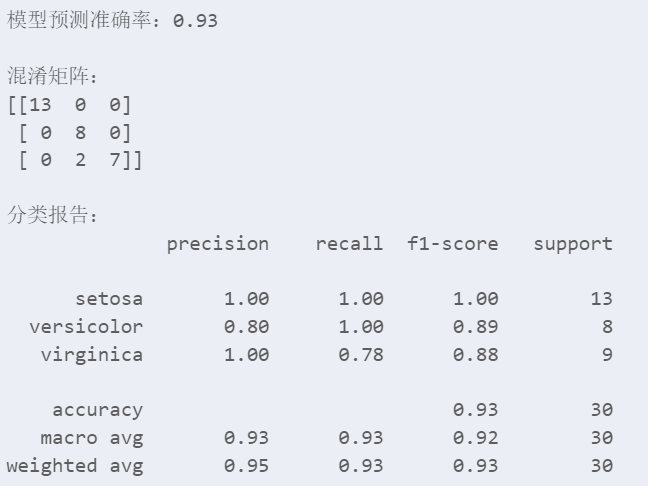
print("\n混淆矩阵：")

print(cm)

report = classification\_report(y\_test, y\_pred)

print("\n分类报告：")

print(report)



**实验总结**

本次实验比较了使用线性核（Linear Kernel）和高斯核（RBF Kernel）在支持向量机（SVM）分类器上的表现。实验基于 Iris 数据集，评估了每种核函数的准确性及其在不同类别上的性能。

线性核结果：

准确率：97%

混淆矩阵：

[[ 9 0 0]

[ 0 10 1]

[ 0 0 10]]

**分类报告**：表现较为均衡，特别是在 setosa 和 virginica 类别上，均达到了较高的精度、召回率和 F1 分数。对于 versicolor 类别，精度和 F1 分数相对较高，但召回率略有下降（0.91）。

高斯核结果：

准确率：93%

混淆矩阵：

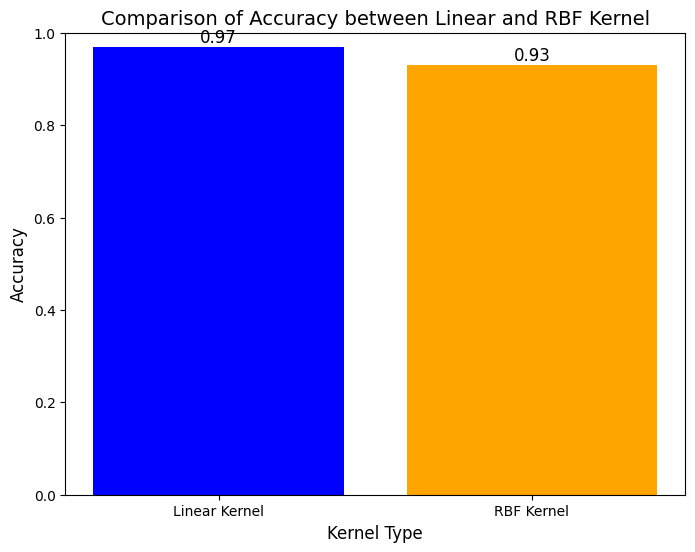
[[13 0 0]

[ 0 8 0]

[ 0 2 7]]

**分类报告**：在 setosa 类别上，精度和召回率均为 1.00，表现完美。但在 versicolor 和 virginica 类别上，准确率有所下降，特别是在 virginica 类别，召回率为 0.78，导致整体准确率低于线性核模型。

**对比分析：**

****

**准确率**：线性核模型的准确率略高（97% vs. 93%），表明线性核在此数据集上的表现优于高斯核。

**精度与召回率**：线性核在 versicolor 类别上表现较为均衡（精度 1.00，召回率 0.91），而高斯核在 versicolor 和 virginica 类别的表现不如线性核，特别是 virginica 类别，召回率下降较为显著（从 1.00 降至 0.78）。

**类别平衡**：高斯核对于类别之间的平衡影响较大，尤其在较小类别样本（如 virginica）的表现上有所下降。相比之下，线性核在所有类别上的表现更为稳定。

**结论：**

尽管高斯核在某些类别上的表现较好（如 setosa），但线性核在整体准确率和各类别平衡性方面表现更优。对于此数据集，线性核似乎是一个更好的选择，而高斯核则在处理某些特定类别时可能出现过拟合或不均衡的情况。

1. **实验总结**

在这次实验中，我学习了如何使用支持向量机（SVM）分类器，特别是线性核和高斯核（RBF核）对数据集进行建模与评估。通过对比两种核的性能，我理解了不同核函数对模型准确率和类别平衡的影响。线性核在此数据集上表现更好，准确率较高，而高斯核虽然对某些类别表现较好，但在其他类别上出现了准确率下降的问题。通过混淆矩阵和分类报告的分析，我进一步认识到模型在不同类别的精度、召回率和F1分数的表现差异。未来可以进一步探索不同的核函数对其他数据集的适用性，并尝试优化模型参数，如调节高斯核的超参数（例如C和gamma），以提高模型的泛化能力。此外，随着深度学习和集成学习方法的快速发展，我也计划尝试使用这些更先进的模型来解决类似的分类问题，进一步提升模型的性能与稳定性。