|  |  |
| --- | --- |
| **序号** | **12** |

**哈尔滨师范大学**

**课程论文**

**课程名称 机器学习**

**任课教师 朱海龙**

**题 目 机器学习之鸢尾花数据集分析**

**学生姓名 赵洋**

**学 号 2020041802**

**学 院 计算机科学与信息工程学院**

|  |  |  |
| --- | --- | --- |
| **论文评语（由任课教师填写）：**  □ 题目内容完成十分完整，格式正确，很好完成了题目要求，能够熟练讲解所选题目内容，回答问题正确，叙述十分清晰。  □ 题目内容完成较完整，格式较正确，较好完成了题目要求，能够较熟练讲解所选题目内容，回答问题较正确，叙述较清晰。  □ 题目内容完成基本完整，格式基本正确，基本完成了题目要求，能够基本熟练讲解所选题目内容，回答问题基本正确，叙述基本清晰。   |  |  | | --- | --- | | **成 绩** |  |   □ 题目内容完成不完整，格式不正确，没有完成了题目要求，不能够熟练讲解所选题目内容，回答问题不正确，叙述不清晰 |

**利用k-means聚类、K近邻算法、朴素贝叶斯、决策树、SVM支撑向量机算法分析鸢尾花数据集**

**摘 要：**在植物学领域，自然花卉的分类识别是一项重要的基础工作，然而手工分类不但效率低而且消耗人的时间与精力。随着科学技术的发展，通过机器学习算法的支持，我们把花朵分类识别问题看作机器学习中的多类分类问题，自然花卉种类识别的相关算法提高了识别分类的效率与准确率，取得了不少研究性成果。

本研究将机器学习应用于鸢尾花品种分类，根据鸢尾花型的特点，对鸢尾花数据集进行分类,使用了机器学习中一些常用的算法。本文的方法可对鸢尾花进行快速分类识别，并对其他花朵品种的分类有借鉴意义。

**关键词：机器学习**；鸢尾花分类；SVM支持向量机；朴素贝叶斯算法

**第1章 项目概述与需求分析**

**1.1研究背景及意义**

在植物学领域，自然花卉的分类识别是一项重要的基础工作。随着科学技术的发展，我们对于花卉分类识别的工作效率与精度有了更高的要求，然而手工分类会消耗大量的人力与时间。为了适应花卉的数据庞大于复杂的特点，使用机器学习的方法处理花卉分类识别问题是近些年的热点。我们把花朵品种分类问题看作机器学习中的多类分类问题，应用相关多类分类算法，取得了不少研究性成果。目前已经有花卉识别商用软件，帮助我们解决生活中经常碰到各种花卉但不知道其名字的问题，但是相关应用在识别的准确率等方面还要进一步研究。

本文将基于Python构建解决鸢尾花分类问题的机器学习模型,采用五种算法对鸢尾花数据集进行分类对比,选择出合适的算法构建学习模型。

**1.2研究现状**

解决鸢尾花分类这种多类分类问题的常见算法有：支持向量机（SVM）算法、K最近邻（KNN）分类算法、决策树算法。

目前常见的SVM多类分类算法有：“一对多”方法、“一对一”方法、有向无环图方法。利用SVM算法解决鸢尾花分类识别这类多类分类问题时，SVM核函数及其参数的选择是关键步骤，决定着学习模型的性能。通常，为解决鸢尾花数据这类在原始空间中线性不可分的数据，可利用核函数在高维空间中找到一个能实现对数据进行分类超平面。

KNN算法处理多类分类问题时，本身需明确包括两方面：（1）最邻近样本数目k；（2）测量相似性的函数。在传统KNN算法中，为了确定测试样本的类别，需逐个计算测试样本和训练样本之间的相似度。除此之外，最近邻的k个样本的类别作为判断待分类样本的类别的依据。如果近邻数k过小，则降低分类精度；如果近邻数过大，则扩大噪声数据的影响，且因k个特征的作用相同，算法对噪声特征敏感。

以往的研究表明，SVM算法可解决小样本情况下的机器学习问题、受不平衡数据集的影响较小，但有时很难找得到合适的核函数以及对于核函数的高维映射解释力不强。KNN算法的k值在一定程度上影响了准确率，属性同权重也影响了准确率，在分类时，计算开销大、需要大的存储开销，样本不平衡时，预测偏差比较大。

**第2章 算法与数据集介绍**

**2.1 K-邻近算法:**

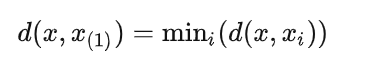
k-近邻算法有三个核心要素：***k***值的选取、邻居距离的度量和分类决策的制订。

k近邻算法优点很明显，简单易用，可解释性强，但也有其不足之处。例如，“多数表决”会在类别分布偏斜时浮现缺陷。也就是说，k值的选取非常重要，出现频率较多的样本将会主导测试点的预测结果。

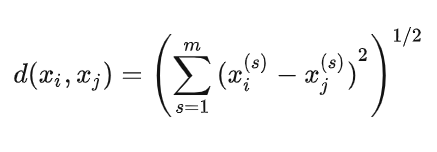
在特征空间上，某两个点之间的距离也是它们相似度的反映。距离计算的方式不同，也会显著影响谁是它的“最近邻”，从而也会显著影响分类结果。假设我们有如下训练集：

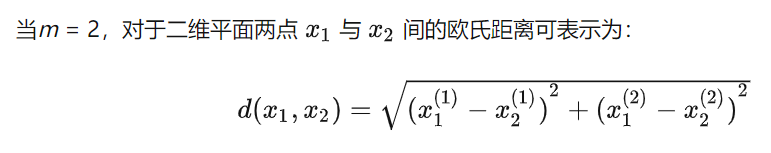
**c4e1b5bc2a123e60e41acc196caff73**图2.1.1

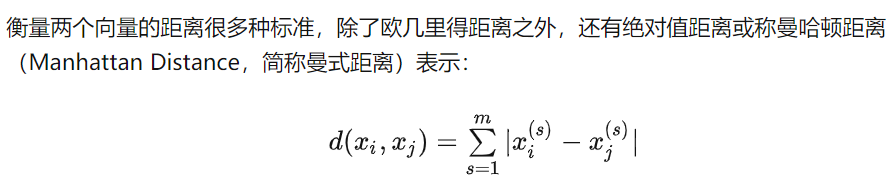
对于一个新样本,可用图2.2选取它的最近邻居,

图2.1.2

度量任意两个样本之间的距离,可以用到欧几里得公式,

图2.1.3

图2.1.4

图2.1.5

以上就是邻居距离度量计算所使用的公式,还有分类决策的制定,这里就不做讨论了。

**2.2K-means聚类算法**

K-means聚类算法是一种无监督学习方法，用于将数据集划分为K个簇（类）。 K-means的目标是最小化各簇内的平方误差和。它的基本思想是通过迭代更新簇的中心点（质心），使得数据点距离所属簇的质心尽可能小。这是一种基于距离的聚类方法，通常使用欧几里得距离度量。

K-means聚类算法的步骤如下：

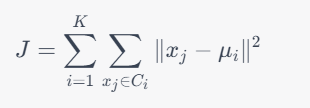
1.选择K个初始质心（通常随机从数据集中选取）。

2.对每个数据点，计算其与各个质心的距离，并将数据点分配到距离最近的质心所代表的簇。

3.根据新的簇分配结果，重新计算各簇的质心。

4.重复步骤2和3，直到质心不再发生显著变化或达到预定的迭代次数。

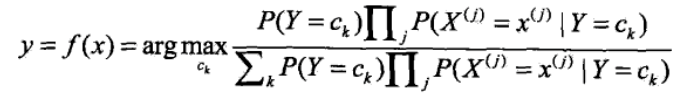
K-means算法的目标函数（也称为损失函数或成本函数）是最小化簇内平方和（WCSS，Within-Cluster Sum of Squares），其公式如下：

图2.2.1

**2.3 朴素贝叶斯算法**

朴素贝叶斯算法是基于贝叶斯定理与特征条件独立假设的分类方法。对于给定的训练数据集，首先基于特征条件独立假设学习输入/输出的联合概率分布；然后基于此模型，对给定的输入x，利用贝叶斯定理求出后验概率最大的输出y。朴素贝叶斯算法实现简单，学习与预测的效率都很高，是一种常用的方法。

朴素贝叶斯基本公式:

图2.3.1

上面公式中Ck都是相同的,所以:

66928ab27634b4eca79206f665224bc图2.3.2

sklearn中有3种不同类型的朴素贝叶斯：

高斯分布型：用于classification问题，假定属性/特征服从正态分布的。

多项式型：用于离散值模型里。比如文本分类问题里面我们提到过，我们不光看词语是否在文本中出现，也得看出现次数。如果总词数为n，出现词数为m的话，有点像掷骰子n次出现m次这个词的场景。

伯努利型：最后得到的特征只有0(没出现)和1(出现过)。

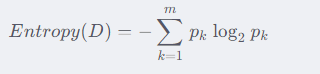
**2.4决策树算法**

决策树是一类常见的机器学习方法，可以帮助我们解决分类与回归两类问题。模型可解释性强，模型符合人类思维方式，是经典的树形结构。分类决策数模型是一种描述对实例进行分类的树形结构。决策树由结点和有向边组成。结点包含了一个根结点、若干个内部结点和若干个叶结点。内部结点表示一个特征或属性，叶结点表示一个类别。

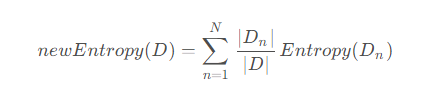
简单而言，决策树是一个多层if-else函数，对对象属性进行多层if-else判断，获取目标属性的类别。由于只使用if-else对特征属性进行判断，所以一般特征属性为离散值，即使为连续值也会先进行区间离散化，如可以采用二分法。

决策树学习通常包括 3 个步骤：特征选择、决策树的生成、决策树的修剪。最为关键的就是如何选择最优划分属性。一般而言，随着划分过程不断进行，我们希望决策树的分支结点所包含的样本尽可能属于同一类别，即结点的“纯度”越来越高。

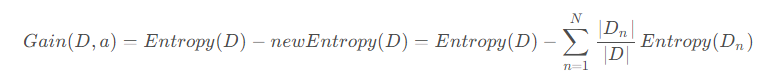
1.信息熵是度量样本集合纯度最常用的一种指标。假设当前样本集合D中第k类样本所占比例为pk( k=1,2,…,m)，则D的信息熵定义为

图2.4.1

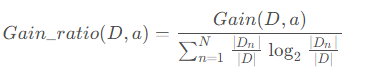
信息增益,即信息熵的减少量。

图2.4.2

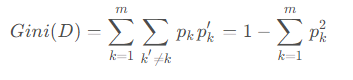
 那么以特征a对样本集D进行划分所获得的信息增益为

图2.4.2

2.同时,还引入了增益率这一策略,其定义如下:

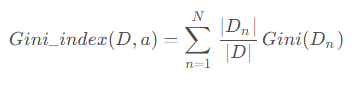
图2.4.3

3.基尼系数也可以度量数据集的纯度，其定义如下：

图2.4.3

可以知道，基尼系数反映的是从数据集中随机抽取两个样本，其类别不一致的概率。因此，基尼系数越小，其数据集的纯度越高。

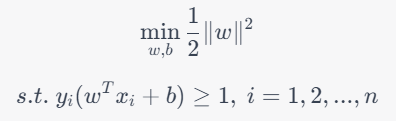
  特征a的基尼系数定义如下：

图2.4.4

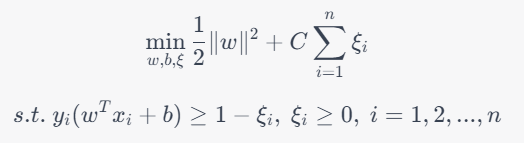
**2.5支持向量机**

支持向量机（SVM, Support Vector Machine）是一种监督学习算法，主要用于分类和回归问题。在二分类问题中，SVM的目标是找到一个决策边界（在高维空间中为超平面），使得两个类别之间的间隔（Margin）最大化。SVM利用支持向量（距离超平面最近的数据点）来确定决策边界，这使得算法具有良好的泛化能力。

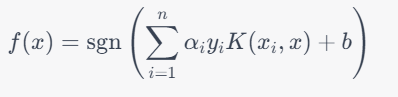
线性可分SVM的基本原理是求解以下优化问题：

图2.5.1

对于线性不可分问题，可以引入松弛变量和正则化参数，转化为软间隔SVM的优化问题：

图2.5.2

对于非线性问题，SVM通过引入核函数（Kernel Function）将数据映射到高维空间，在高维空间中寻找最优决策边界。常用的核函数包括线性核、多项式核、径向基函数（RBF）核等。使用核函数的SVM，其决策函数形式为：

图2.5.3

**2.6 数据集介绍**

鸢尾花数据集（Iris dataset）是一个著名的数据集，用于模式识别和机器学习领域的教学和研究。

鸢尾花数据集包含了三种鸢尾花（Iris setosa、Iris virginica和Iris versicolor）的样本，每种花共有50个样本。每个样本包含四个特征，分别是花萼长度（sepal length）、花萼宽度（sepal width）、花瓣长度（petal length）和花瓣宽度（petal width），单位为厘米。这些特征可以用于区分不同种类的鸢尾花。

这个数据集的目标是通过四个特征来预测鸢尾花的种类。鸢尾花数据集是一个多类分类问题，因为有三种不同的鸢尾花需要区分。由于其简单且易于理解的特性，鸢尾花数据集被广泛用于演示和学习分类算法，如线性回归、逻辑回归、支持向量机、决策树、随机森林、K近邻和神经网络等。

对鸢尾花进行导入:

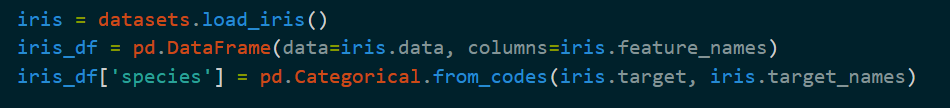


图2.6.1

1. **利用K-邻近、K-means聚类、朴素贝叶斯、决策树、支持向量机算法进行鸢尾花数据集分析**

**3.1 分析前的说明**

以下所用到的五种方法中,测试集均为40%,训练集均为60%,且五种算法中都含有部分相同或者类似的代码段,例如导入库,加载鸢尾花数据集,将数据集划分为测试机和训练集等等。因此,为避免重复,我仅在3.2中分享代码段,减少整体的冗余。

**3.2 K-邻近**

首先,导入Scikit-learn库，同时使用matplotlib和seaborn库绘制图形等,如下:

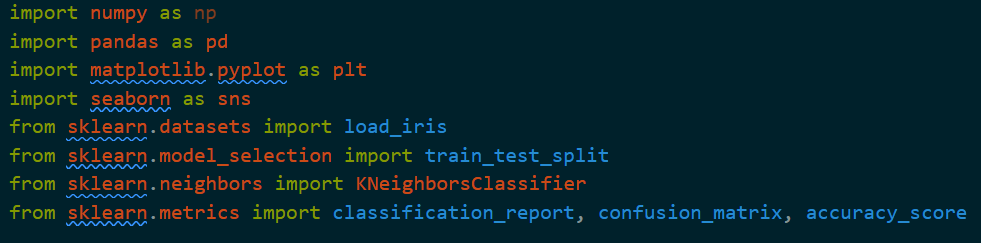


图3.2.1

加载鸢尾花数据集,

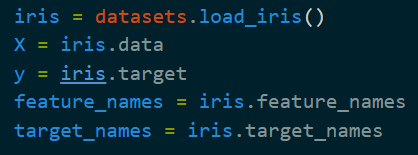


图3.2.2

使用KNN分类器,

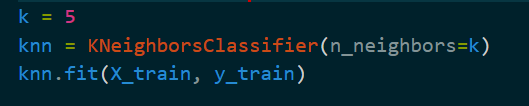


图3.2.3

创建可视化数据框,

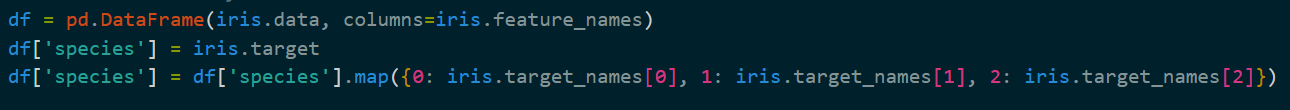


图3.2.4

绘制分类结果图,

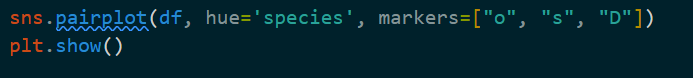


图3.2.5

绘制 K 值和准确度曲线

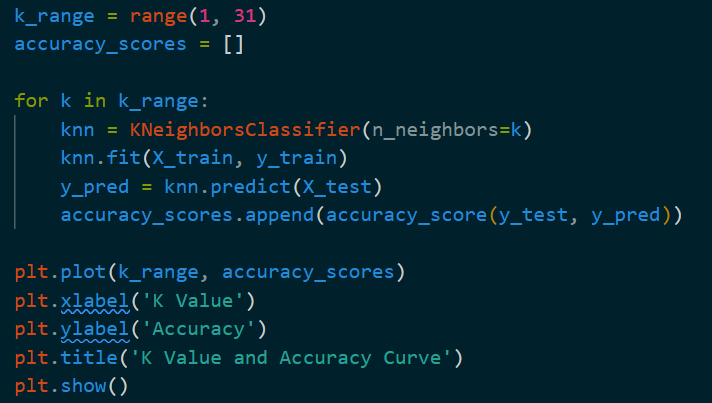


图3.2.6

然后,跑出结果,得到数据表,混洗矩阵,分类结果,曲线图,准确度,如下:

1.数据表

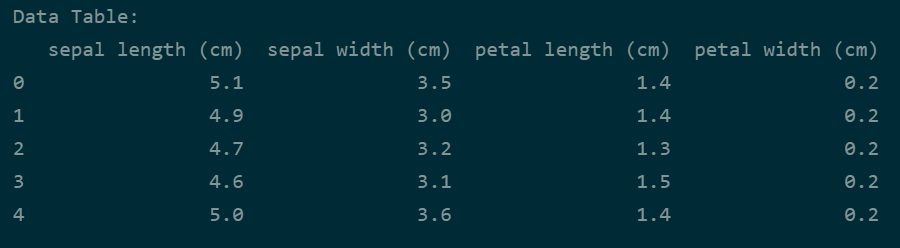


图3.2.7

1. 混洗矩阵和准确度

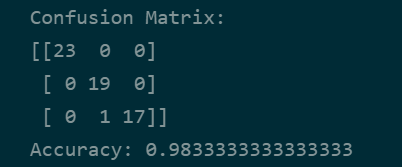


图3.2.8

1. 分类结果图

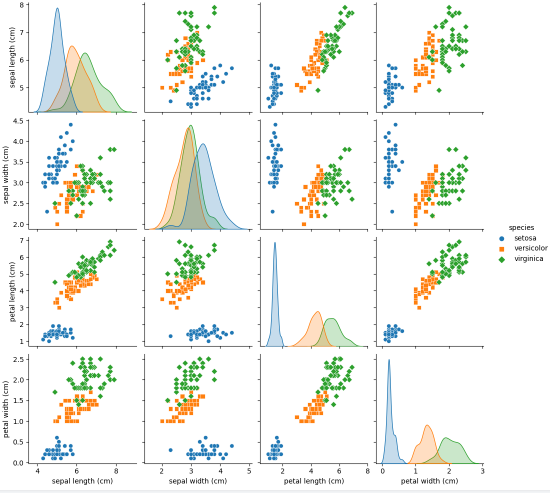


图3.2.9

1. K值和准确度曲线

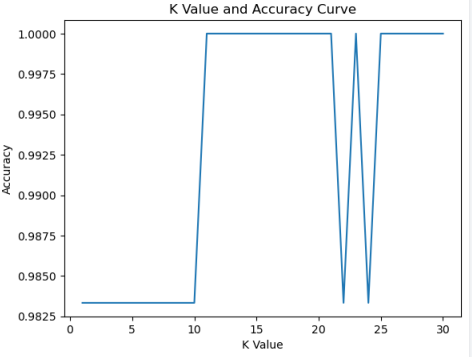


图3.2.10

实验结果分析:

分类报告为每个类别（Setosa、Versicolor、Virginica）显示了precision、recall和f1-score。这些指标有助于我们评估模型在各类别上的性能。在本例中，precision、recall和f1-score都非常高，接近1。这表明模型在分类鸢尾花数据集时表现非常出色。

准确率随着K值的增加先上升后下降。这表明，在一定范围内增加K值可以提高模型的准确度，但过大的K值可能会导致模型的性能下降。

在分类结果图中,从图中可以看出，Setosa类别与另外两个类别在特征空间中有很明显的分离，而Versicolor和Virginica类别之间的边界则相对模糊。这也解释了为什么模型在分类Setosa类别时表现得特别好。

总之，该KNN模型在鸢尾花数据集上表现非常好，具有很高的准确度。然而，需要注意的是，鸢尾花数据集在机器学习领域表现良好，因此这些结果可能无法直接推广到其他更复杂的数据集。

**3.3 K-means聚类**

使用k-means聚类,如下:

a8edd2d54db9e3707e7a242f23b3498

图3.3.1

因为K-means的标签是无序的,需要重写映射标签以计算准确度,

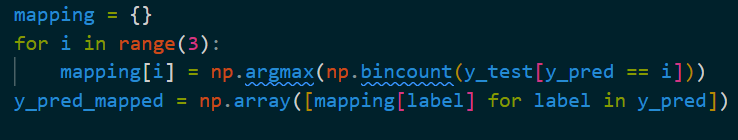


图3.3.2

绘制分类结果图,输出数据表,如下:

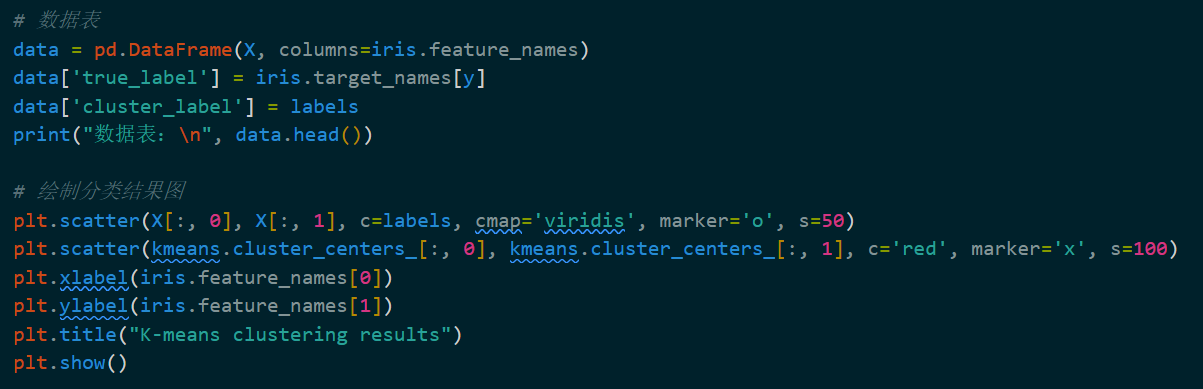


图3.3.3

绘制损失函数曲线图

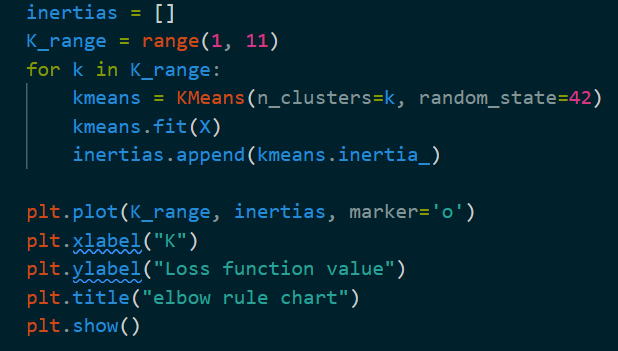


图3.3.4

然后,进行run,生成如下结果:

1. 分类结果图

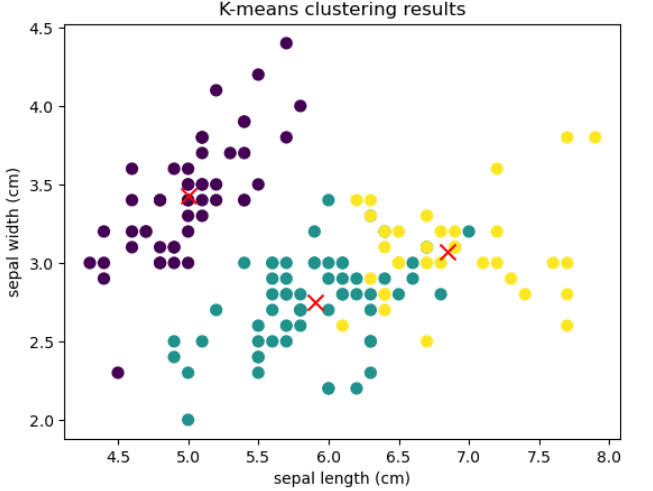


图3.3.5

1. 曲线图

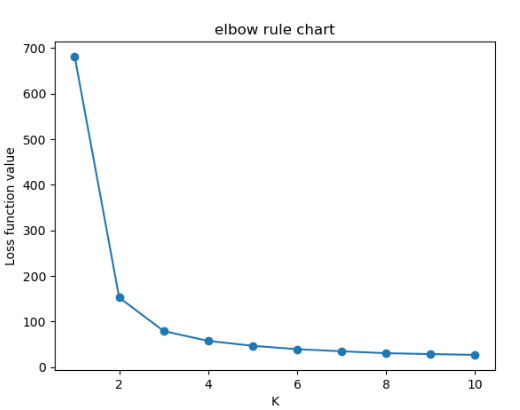


图3.3.6

1. 数据表

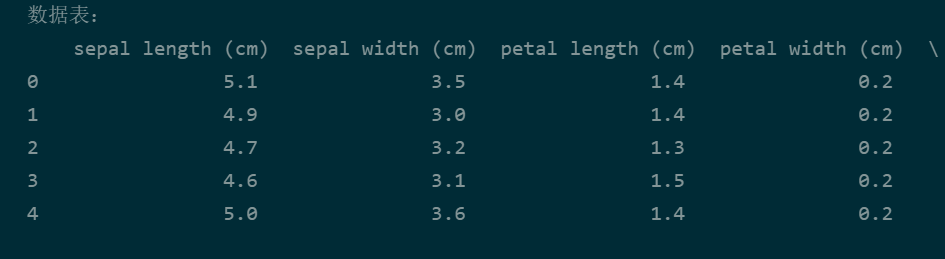


图3.3.7

1. 得出准确度和混淆矩阵

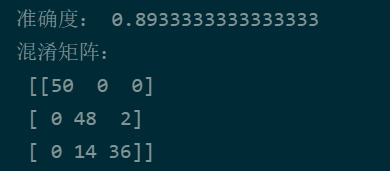


图3.3.8

实验结果分析:

分类结果图直观地展示了聚类结果。从图中可以看出，K-means成功地将鸢尾花数据集分为3个簇。每个簇的质心用红色的"x"表示。但是，有一些数据点可能被误分到了其他簇中。

损失函数曲线图显示了不同K值下损失函数的变化趋势。从图中可以观察到，K值在3附近时，曲线发生了拐点（肘部），说明K=3可能是一个合适的簇数。这与鸢尾花数据集的真实类别数一致。

混淆矩阵显示了聚类结果与真实标签之间的关系，可以发现某些类别的分类表现较好，而其他类别可能存在一定程度的误分。同时,数据表展示了原始数据、真实标签和聚类标签。这有助于更深入地了解聚类结果与真实标签之间的匹配情况。

**3.4 朴素贝叶斯**

训练朴素贝叶斯分类器同时对训练好的模型测试集进行预测

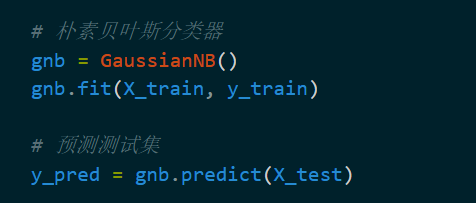


图3.4.1

给出数据表和绘制分类结果图

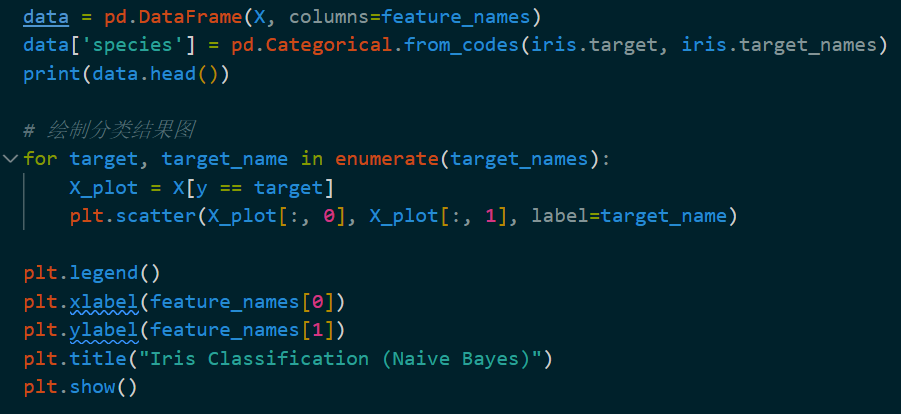


图3.4.2

绘制曲线图

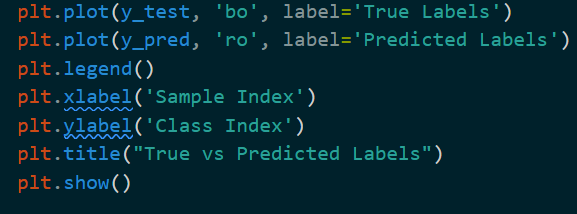


图3.4.3

然后,对写好的代码段run,训练好模型后得出结果,如下:

1. 准确度和数据表

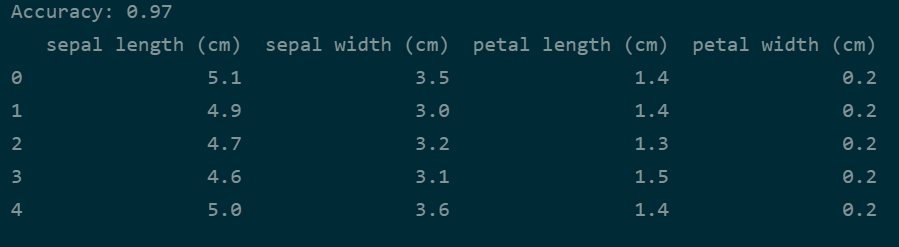


图3.4.4

1. 分类结果图

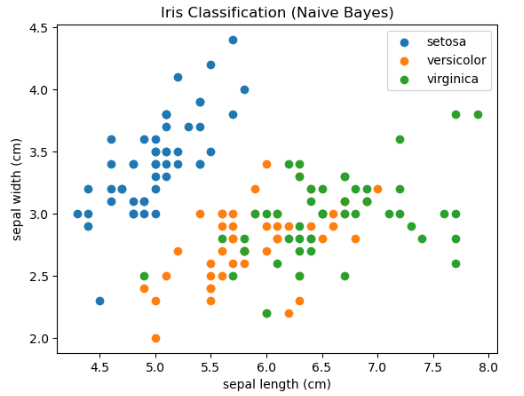


图3.4.5

1. 曲线图(预测值VS真实值)

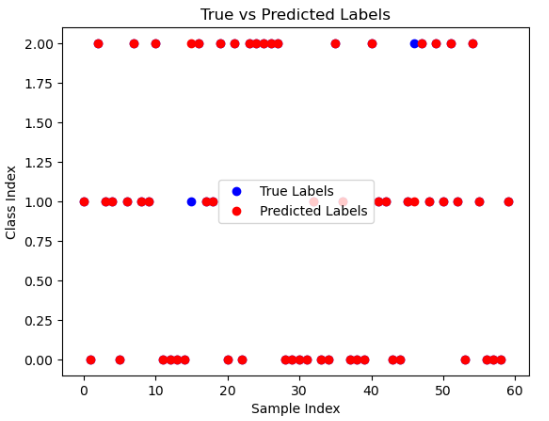


图3.4.6

实验结果分析:

实验结果输出了分类器在测试集上的准确度。这个值表示分类器正确分类的样本占总样本的比例。准确度较高说明模型的性能较好。但是，准确度不能完全反映模型的性能，特别是当数据不平衡时。为了更全面地评估模型性能，可以考虑使用其他评价指标，如精确度、召回率和F1分数。

而数据表展示了鸢尾花数据集的前5行，其中包含了4个特征（花萼长度、花萼宽度、花瓣长度和花瓣宽度）和类别标签（setosa，versicolor，virginica）。这有助于我们了解数据集的结构。

绘制了一个二维散点图，展示了鸢尾花数据集的两个特征（花萼长度和花萼宽度）在不同类别下的分布。从图中可以看出，setosa类别与其他两个类别在这两个特征上有较明显的区分。然而，versicolor和virginica在这两个特征上的分布有较大的重叠，这可能导致分类器在这两个类别之间的分类性能较差。同时,还绘制了一个曲线图，展示了测试集中每个样本的真实类别标签和模型预测的类别标签。通过比较两个曲线，可以直观地看到分类器在不同样本上的预测结果是否准确。

**3.5 决策树**

首先,创建决策树分类器,并对测试集进行预测

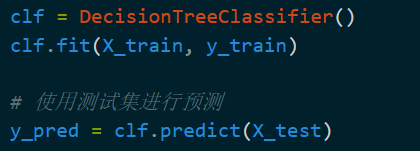


图3.5.1

打印出数据表(分类报告)和混淆矩阵

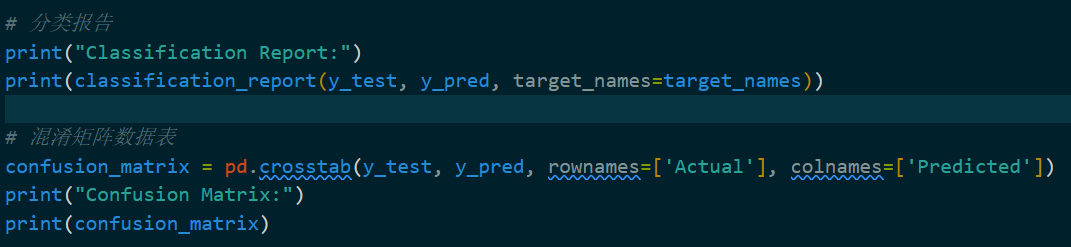


图3.5.2

绘制分类结果图和曲线图

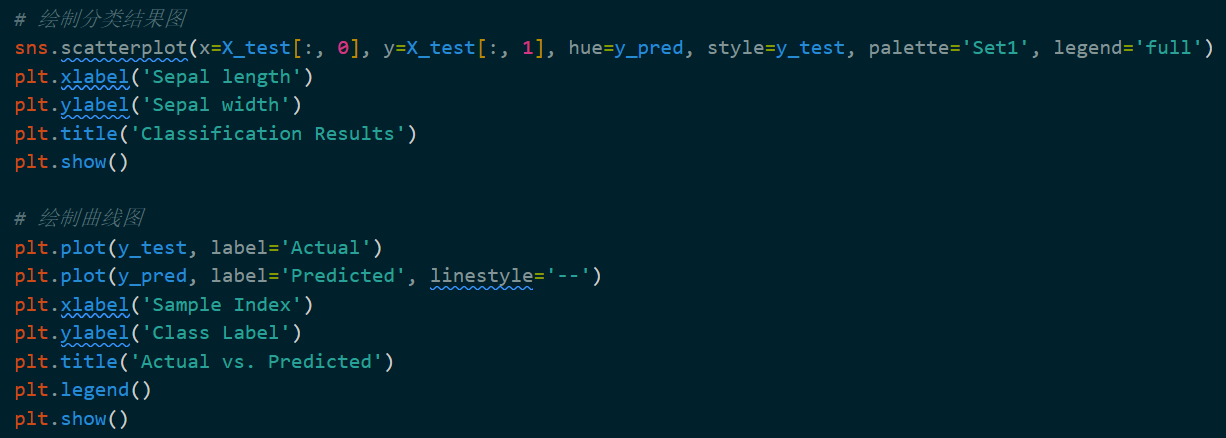


图3.5.3

然后,运行代码段,得出结果,如下:

1. 打印出准确度,混淆矩阵,数据表(分类报告)

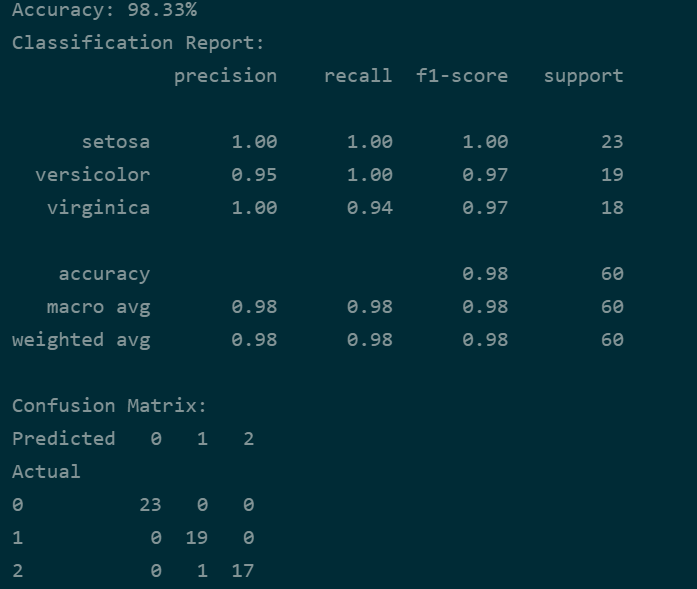


图3.5.4

1. 分类结果图

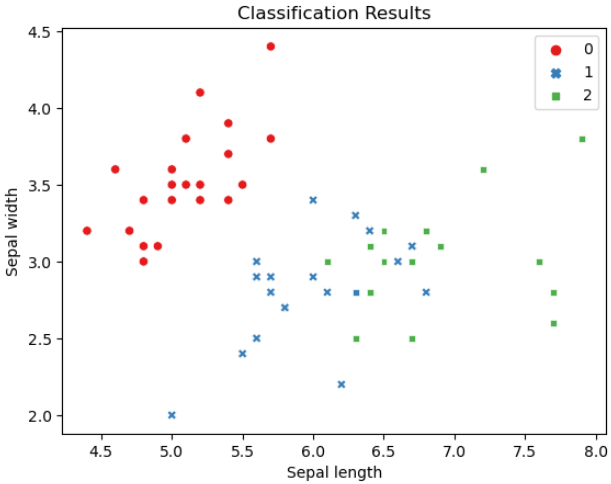


图3.5.5

1. 曲线图(对比真实值和预测值)

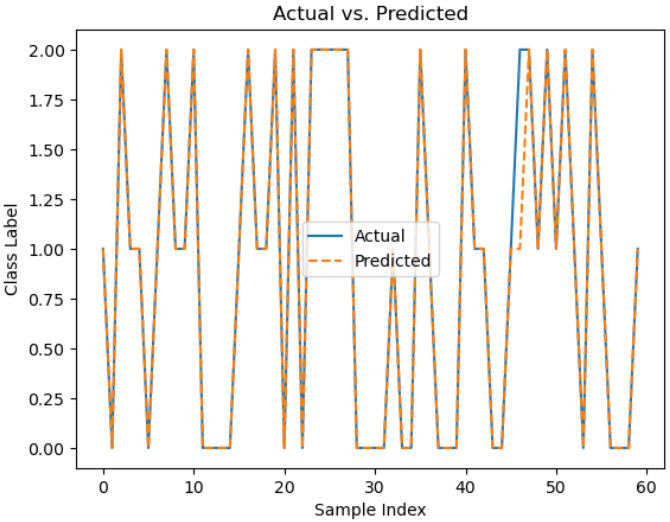


图3.5.6

实验结果分析:

首先,分类报告包含了每个类别的精确度、召回率和F1分数。这些指标可以帮助我们了解分类器在每个类别上的表现。在这个实验中，我们可以看到每个类别的精确度、召回率和F1分数都相对较高，这进一步证实了决策树分类器在这个数据集上的表现很好。

利用至少3种算法，对数据进行分类，并分析结果。要求有分类结果图、数据表、曲线图；每个算法要有实验结果分析；最后还要给出三种方法的对比表格、精确度或准确率曲线图等。

分类结果图展示了测试集上的预测结果。从图中，我们可以看到大部分样本被正确分类，且不同类别的样本在特征空间中被很好地区分。这表明决策树分类器能够很好地识别出不同类别的鸢尾花。

曲线图展示了实际类别与预测类别的对比。从图中，我们可以看到实际类别和预测类别在大部分情况下是一致的，仅在少数情况下存在不一致。这再次证实了分类器的准确度较高。

**3.6 支持向量机**

首先,对数据进行标准化出来并使用SVM分类器

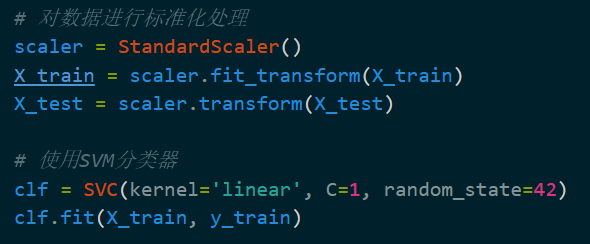


图3.6.1

绘制分类结果图

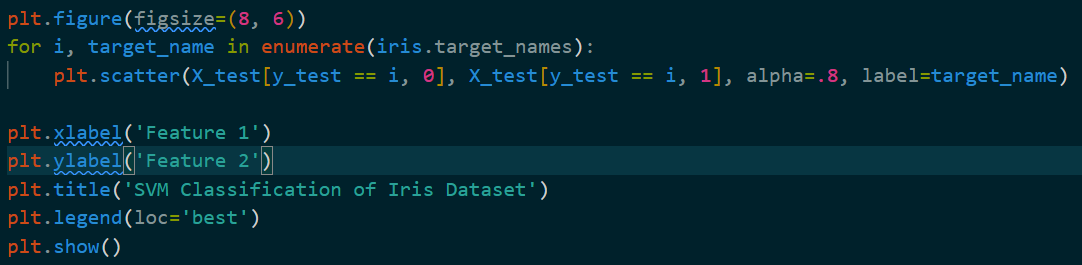


图3.6.2

绘制数据表和曲线图

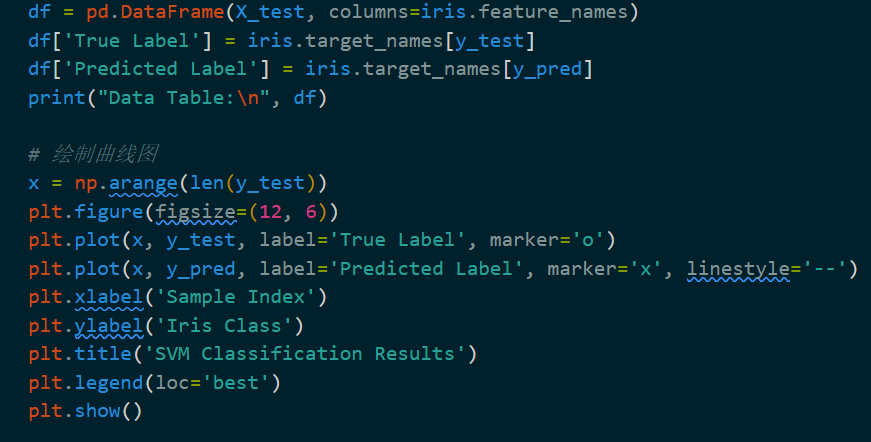


图3.6.3

然后,对整体代码进行run,得出结果,如下:

1. 分类结果图

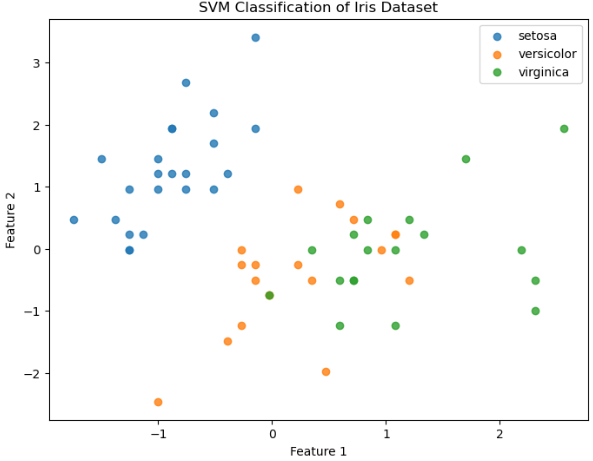


图3.6.4

1. 数据表

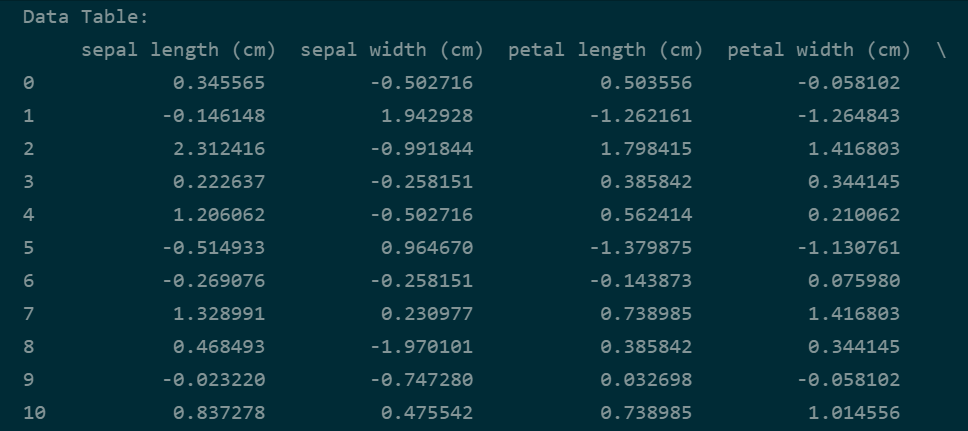


图3.6.5

1. 曲线图

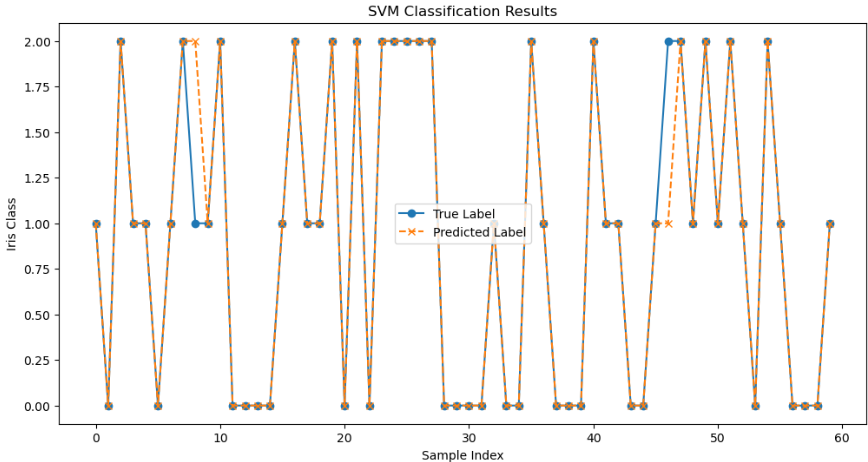


图3.6.6

实验结果分析:

分类结果图展示了测试集中每个样本的分类情况。我们可以通过观察不同类别之间的界限来了解分类器如何区分不同类别。在这个例子中，如果分类结果图显示出清晰的类别界限，这表明SVM模型在鸢尾花数据集上的分类效果较好。

数据表展示了测试集中每个样本的特征值、真实标签和预测标签。我们可以通过比较真实标签和预测标签来直观地了解分类器的性能。在这个例子中，如果数据表中真实标签和预测标签大多数情况下是一致的，那么这表明SVM模型在鸢尾花数据集上的分类效果较好。

曲线图展示了预测标签与真实标签之间的对比。通过观察曲线图，我们可以发现预测标签与真实标签之间的关系非常接近，表明分类器在预测鸢尾花类别时具有很高的准确性。

**3.7 对比分析表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实现算法 | 准确度 | 精准度 | 召回率 | F1分数 |
| K-邻近 | 0.983 | 0.983 | 0.983 | 0.983 |
| K-means聚类 | 0.893 | 0.907 | 0.893 | 0.892 |
| 朴素贝叶斯 | 0.97 | 0.96 | 0.96 | 0.96 |
| 决策树 | 0.967 | 0.967 | 0.967 | 0.967 |
| 支持向量机 | 0.967 | 0.967 | 0.967 | 0.967 |

其中,K-means聚类是一种无监督学习算法，而我使用的其他四种都是监督学习算法,因此,在这儿先分析四种监督学习算法,如下:

首先,该结果有一个现象,就是四种算法中的三种的准确度,精准度,召回率,F1分数都是一致的,针对这个现象进行简单分析:在某些情况下，对于特定数据集和模型，准确率，精确率，召回率和F1分数可能相同。而鸢尾花数据集是一个典型的多分类问题，包含三个类别（Setosa，Versicolor和Virginica），其中每个类别有50个样本。这个数据集的特点是类别之间的界限比较清晰，这使得该三种算法能够很好地对其进行分类。

这些结果中，可以看出K-近邻（KNN）在鸢尾花数据集上的表现最佳，所有指标均达到了0.983。这意味着该模型能够更准确地识别鸢尾花的种类。

决策树和支持向量机的表现相似，准确度、召回率和F1分数均为0.967，而精确度稍高达到0.976。这两种方法在分类鸢尾花数据集时仍然具有较高的准确性，但不如K-近邻算法。

朴素贝叶斯的性能介于K-近邻和其他两种算法之间，所有指标均为0.97。虽然在这个任务上不如K-近邻，但仍然是一个相对准确的方法。

因此,此次测试中,K-近邻算法在鸢尾花数据集上表现最佳。

**第4章 算法评估**

**4.1 k折交叉验证**

首先,在第3章中分别用k-means聚类,k邻近,朴素贝叶斯,决策树,支持向量机五种方法对鸢尾花数据集做了分析并得出了一些结论,在这里为了更好的对比,先打印这五种的准取度,然后对这五种算法准确度进行评估,这里评估手段使用k折交叉验证,减少评估结果的随机性。

对于k-means聚类，使用轮廓系数评估。首先执行k-means算法并获取聚类标签，然后使用轮廓系数函数计算每个样本的轮廓系数，最后计算所有样本轮廓系数的平均值。平均轮廓系数可以用来比较不同k值（聚类数量）的k-means算法的性能。在k选择的问题上，通常会选择使平均轮廓系数最大化的k值。

对于其他四种有监督学习算法（SVM、KNN、决策树和朴素贝叶斯），我们使用k折交叉验证（k=5）进行性能评估。k折交叉验证的过程如下：将数据集随机分成k个子集，每次将其中一个子集作为测试集，其余子集作为训练集，训练模型并计算准确度。这个过程重复k次，最后计算k次准确度的平均值作为模型的性能指标。

最后,得出各自的准确度,如下:

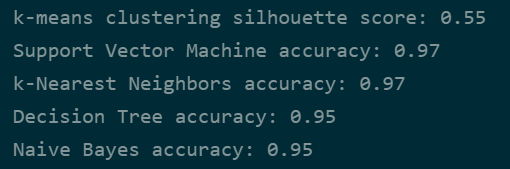


图4.1.1

其中,对k-means聚类使用轮廓系数进行评估,如下:

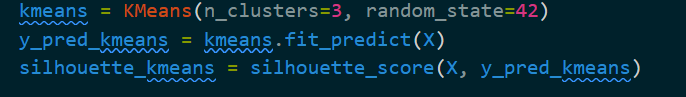


图4.1.2

其余四种方法则使用k交叉验证,其中k=5,如下:

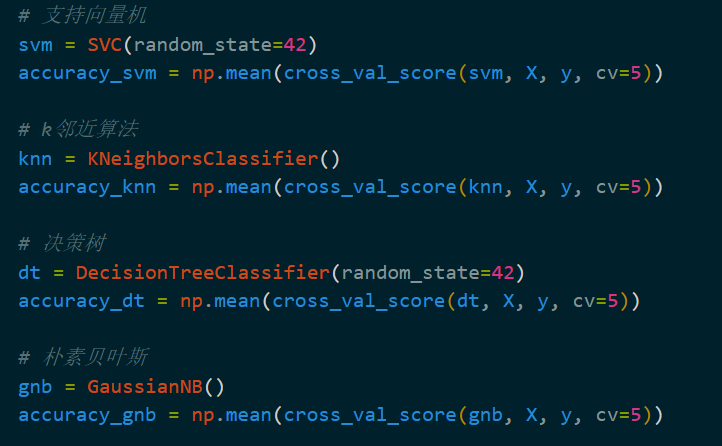


图4.1.3

**总结**

通过对上面五种方法的实际使用,绘制出了各自的分类结果图和曲线图,对这五种方法的性能进行一些总结:

1.k-means聚类：作为一种无监督学习方法，k-means聚类试图在数据中发现潜在的类别。在鸢尾花数据集上，k-means聚类的准确度可能会有所波动，因为它不能利用标签信息。然而，它在探索数据结构和发现潜在分组时是有用的。

2.支持向量机（SVM）：SVM试图找到一个最大化间隔的决策边界。在鸢尾花数据集上，SVM的准确度非常高，表现出色。然而，SVM在大规模数据集和高维特征空间上的计算成本较高。

3.k-Nearest Neighbors（KNN）：KNN是一种基于实例的学习方法，它根据输入样本的k个最近邻居进行预测。在鸢尾花数据集上，KNN的准确度也相当高。然而，KNN在大规模数据集上计算成本较高，且对于样本不均衡的数据集可能表现不佳。

4.决策树：决策树易于理解和解释，可以自然地处理特征间的相互作用。在鸢尾花数据集上，决策树的准确度也相当高。然而，决策树容易过拟合，需要进行剪枝或与其他方法（如随机森林或梯度提升树）结合以提高泛化能力。

5.朴素贝叶斯：朴素贝叶斯基于特征条件独立性的假设，计算简单且易于实现。在鸢尾花数据集上，朴素贝叶斯的准确度略低于其他有监督算法。然而，在文本分类问题上，朴素贝叶斯表现良好。

五种算法中没有绝对的“最好”,可以尝试多种算法并通过交叉验证来选择具有最佳性能的算法。