**国家卓越工程师学院明月科创实验班**

**《概率论与数理统计》课程项目1**



**姓 名： 王子诺**

**学 号： 20235459**

**年 级： 2023级**

**班 级： 1班**

**日 期： 2025年6月25日**

重庆大学国家卓越工程师学院2024年制

目录

[概率论与数理统计项目二报告 2](#_Toc27541)

[一、 项目分析 2](#_Toc31167)

[二、 数据寻找和预处理 3](#_Toc22307)

[2.1 数据库来源 3](#_Toc26174)

[2.2 数据库预处理 4](#_Toc17993)

[2.3 数据预分类 5](#_Toc13446)

[2.4 开源数据链接 6](#_Toc23392)

[三、 代码实现 6](#_Toc31758)

[3.1 Logistic Regression方法实现 6](#_Toc23684)

[3.2 硬间隔SVM方法实现 9](#_Toc11540)

[3.3 非线性软间隔SVM（基于RBF，来源论文二） 12](#_Toc31039)

[3.4 线性软间隔SVM（基于liblinear，来源论文二中的SGD） 14](#_Toc24600)

[四、 文献阅读 17](#_Toc31405)

[4.1 论文一 17](#_Toc27134)

[4.2 论文二 18](#_Toc6543)

# 

# 概率论与数理统计项目二报告

## 项目分析

本次项目的主要目标是实现一个基于SVM（支持向量机）的分类问题，并将所得结果与Logistic Regression进行详细的数据对比分析。在构建SVM模型的过程中，我们需要分别实现硬间隔SVM和软间隔SVM两种不同的模型。具体来说，硬间隔SVM侧重于寻找一个严格的分类边界，而软间隔SVM则允许一定程度的误分类，以更好地处理实际数据中的噪声和异常点。

在数据准备阶段，我将采取多种途径进行数据收集，既包括从实际应用场景中获取的真实数据，也包括通过特定算法生成的模拟数据。为了确保模型的训练效果和评估准确性，数据集必须明确划分为训练集和测试集（或预测集）。每类样本的数量不得少于250个，这样的样本量不仅能够提供足够的数据支撑，还能确保数据集的高质量，从而为算法性能的提升奠定坚实基础。此外，我们还将按照一定比例将数据集进一步划分为训练集和验证集，以有效提升模型的泛化能力，确保其在未知数据上的表现。

在数据库建立过程中，对数据指标的选取尤为关键。我们将通过一系列科学的方法对数据指标进行细致的筛选，力求找到相关性最为合适的两个指标，以便更准确地实现分类任务。这一步骤不仅关系到模型的训练效果，更是直接影响最终分类结果的关键环节。

在实现硬间隔SVM建模时，我们选择使用线性决策边界。然而，这种看似简单直接的方法在实际应用中存在一定的局限性，因为它仅选取部分数据点作为支持向量，可能导致模型对异常点过于敏感，进而影响最终的分类结果。与之相比，Logistic Regression作为一种经典的分类算法，成为了我们自然选择的对比基准，能够更全面地考察线性决策边界的适用性和有效性，为我们提供更为可靠的对比依据。

## 数据寻找和预处理

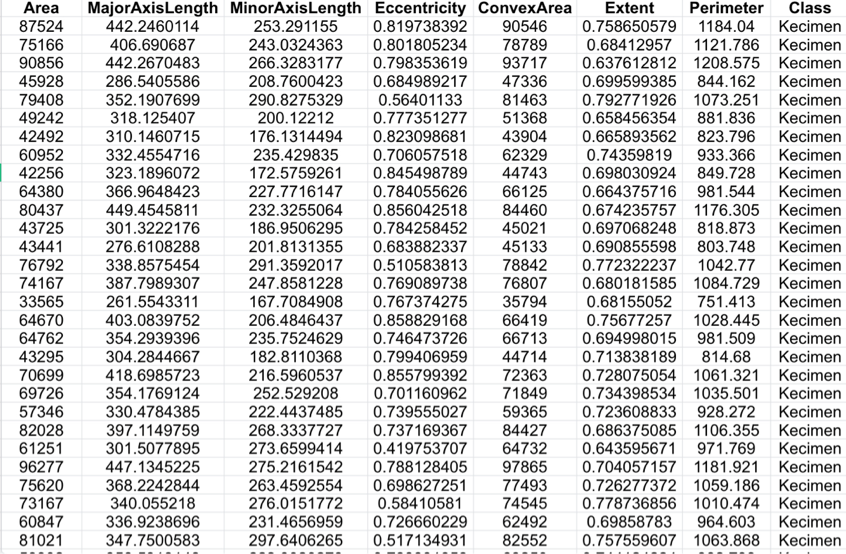
### 2.1 数据库来源

为了寻找合适的SVM数据库，我在kaggle上找到了一个关于两类葡萄干的数据集raisin\_dataset，其中有900个关于两种土耳其葡萄干（Kecimen&Besni）的多个特征的数据集，包含了面积、主X轴长度，交叉面积、含量等等信息。



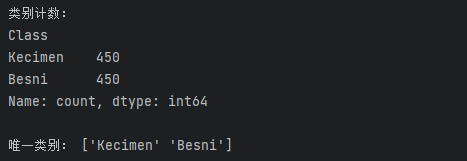
（葡萄干数据集）

这个数据集的来源是从两个品种中获得了900粒葡萄干的图像。这些图像经过了各种预处理步骤，并使用图像处理技术进行了7次形态特征提取操作。此外，还计算了每个特征的最小值、平均值、最大值和标准偏差统计信息。研究了两种葡萄干品种在特征上的分布，并将这些分布显示在图表上。



（数据集及特征）

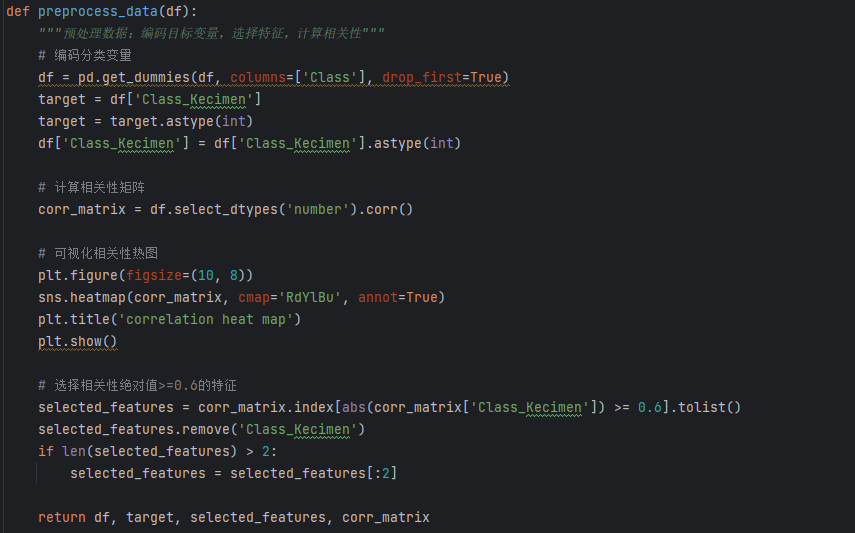
而在其中，两种类型的葡萄干数据分别有450张，我对excel的表格中的数据进行了格式统一为int64类型，以方便后续的处理。



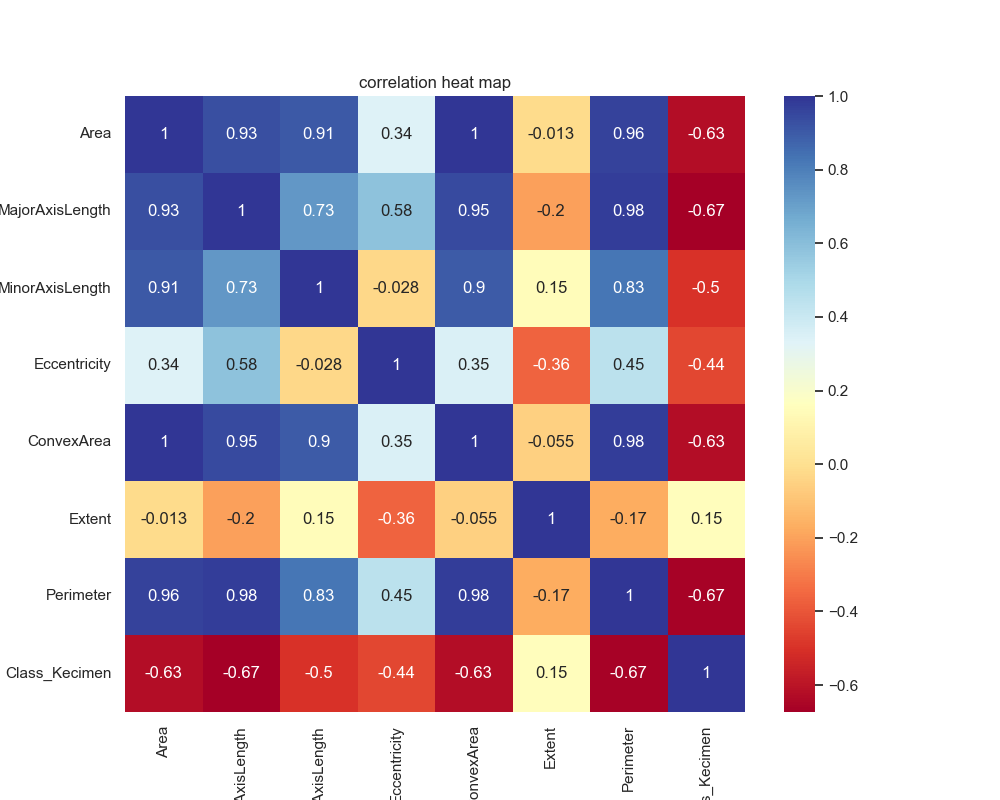
（两个类别数据的统计量）

### 2.2 数据库预处理

对于这个数据集中的很多数据特征，我需要找到最合适的数据特征来进进行模型的建立，因此，我定义一个名为 preprocess\_data的函数，对DataFrame中的 Class 列进行独热编码,将Class\_Kecimen列提取为目标变量target，使用相关性矩阵显示每对数值特征之间的相关性，值范围在 [-1, 1] 之间。最后再用相关性热图来可视化展示这些数据特征的相关性，选择相关性大于0.6的前两个特征参数来作为模型搭建的参数，最终选出来得到Area和MajorAxisLength这两个指标。



（数据处理部分代码）



（相关性热图）

### 2.3 数据预分类

在此之后，每次进行模型训练之前，我还要划分数据集和验证集，从sklearn.model\_selection中导入train\_test\_split函数，引入随机数种子，按照4:1的比例去随机划分，从而使得模型能够具有泛化性。

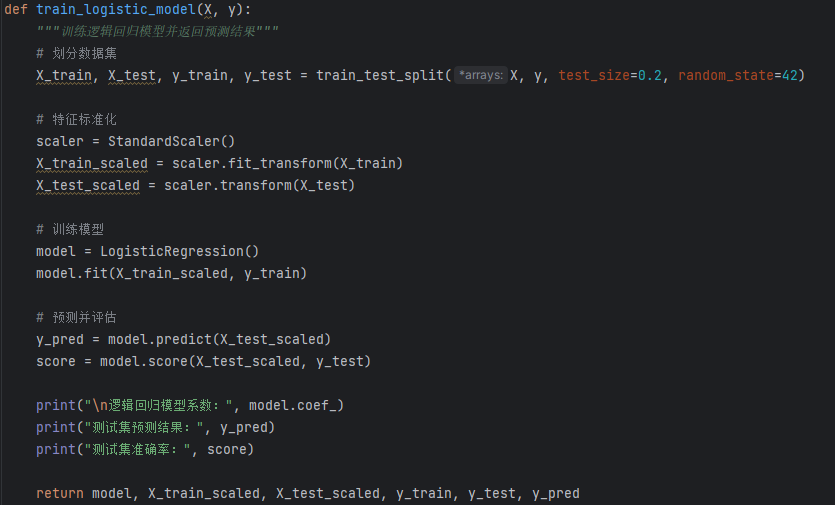
### 2.4 开源数据链接

开源数据链接： <https://www.kaggle.com/datasets/muratkokludataset/raisin-dataset/data>

## 代码实现

### 3.1 Logistic Regression方法实现

#### 3.1.1 代码构架

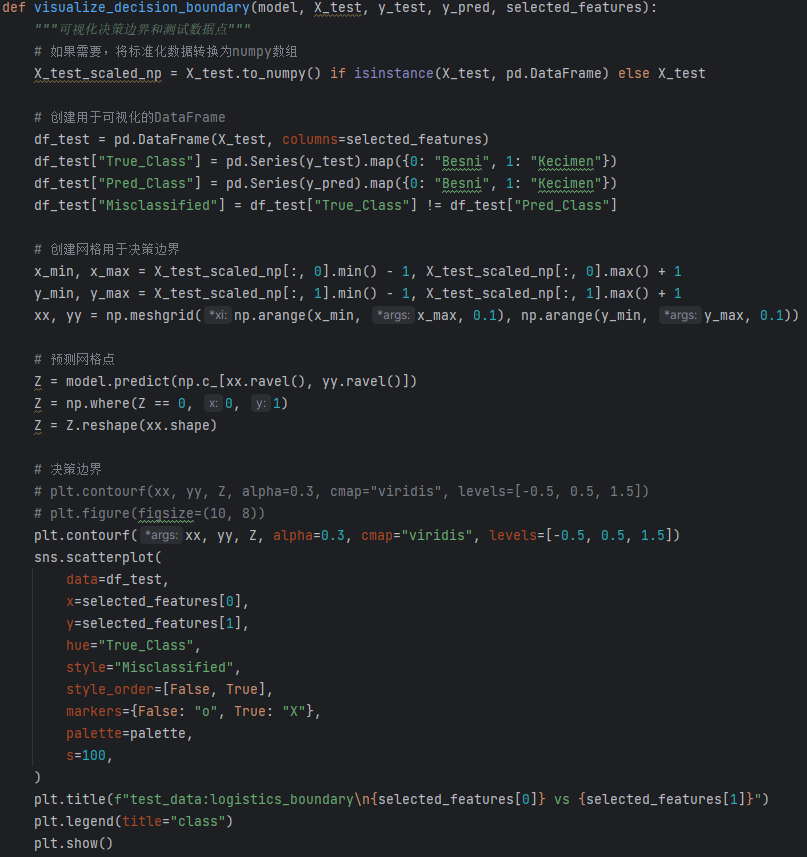


（Logistic回归代码）

我在使用logistic回归方法进行分类时，首先划分为训练集和测试集。随后，为了提高模型的训练效率和预测准确性，必须对训练集的特征进行标准化处理。标准化过程包括计算每个特征的均值和标准差，并将所有特征值转换为均值为0、标准差为1的标准化数据。这一步骤不仅有助于消除不同特征量纲的影响，还能使模型更容易收敛。标准化后的训练集和测试集特征分别存储在X\_train\_scaled和X\_test\_scaled变量中，以便后续使用。

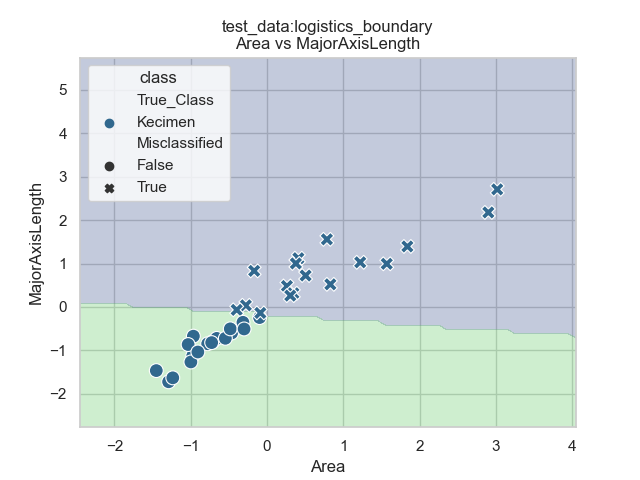
为了全面评估模型的性能，需要利用训练好的模型对测试集进行预测，并将预测结果存储在y\_pred变量中。此外，通过计算模型的score（通常指准确率或其他评估指标），可以量化模型在测试集上的表现，从而对模型的泛化能力有一个直观的认识。这一系列的评估步骤不仅有助于判断模型的优劣，还能为进一步的模型优化提供重要参考。

#### 3.1.2 可视化结果显示



（绘制决策边界）

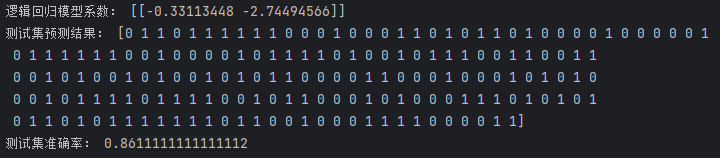
该代码中我标记预测错误的样本（True 表示真实类别与预测类别不一致，False 表示预测正确）。然后使用网格的方法进行决策边界，从而得到特定的数据点是否为kecimen（其实否也就是Besni类的）。正确分类的点用圆形（"o"），错误分类的点用叉（"X"）。



（Logistic\_决策边界结果图）

#### 3.1.3 模型评估

最终模型的测试结果为：



模型系数反映了每个特征对目标变量（即分类结果）的贡献程度。第一个特征的系数为-0.33113448，负值意味着当该特征值增加时，模型预测目标类别为1（例如“Kecimen”）的概率会降低，更倾向于预测类别0（例如“Besni”）。第二个特征的系数为-2.74494566，较大的负值显示出该特征对分类的影响更为显著，且增加该特征值将显著降低预测类别1的概率。

测试集准确率为 86.11%，表示模型在测试集上的预测正确率，具有较强的分类能力，即正确分类的样本占总样本的比例。也就是说，在数据集平衡的前提下，这个准确率表明模型能够较好地区分两类数据。

### 3.2 硬间隔SVM方法实现

#### 3.2.1 代码构架



(硬间隔SVM代码实现）

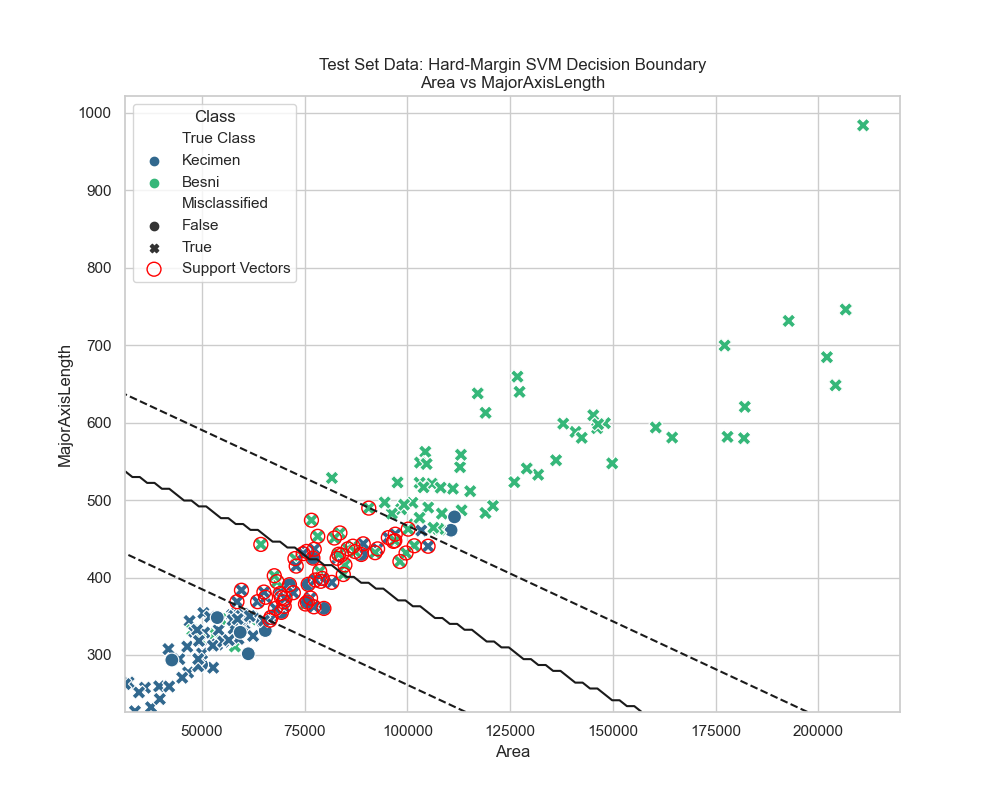
这里写一个函数train\_svm\_model用于训练一个硬间隔支持向量机（SVM）模型并返回预测结果。也是使用 train\_test\_split 将输入特征 X 和目标变量 y 划分为训练集，然后通过 StandardScaler 对训练集和测试集特征进行标准化处理，以确保特征在相同尺度上。接着，函数尝试使用 LinearSVC（C=1e10，用一个非常大的值来模拟硬间隔 SVM）在标准化后的训练数据上训练模型，若数据线性不可分则捕获异常并返回空值。训练完成后，模型在测试集上进行预测，输出模型系数、预测结果和准确率。最后，函数返回训练好的模型、原始和标准化后的测试集与训练集特征、训练集和测试集目标变量、预测结果以及标准化器对象。此函数适用于二分类任务，特别强调硬间隔 SVM 的应用，但要求数据线性可分。

#### 3.2.2 可视化结果显示



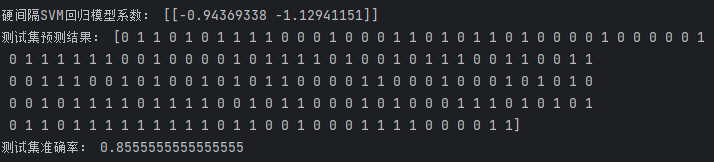
(可视化硬间隔SVM代码）

此处使用函数 visualize\_decision\_boundary\_for\_svm 用于可视化硬间隔支持向量机（SVM）模型在测试集上的决策边界、支持向量和数据点。基于原始（未标准化）测试集特征 X\_test，创建包含真实类别（"Besni" 或 "Kecimen"）、预测类别和误分类标记的 DataFrame。然后，生成一个基于两个特征(selected\_features)的二维网格，标准化网格点后使用模型预测类别以绘制决策边界（直线）和间隔边界（w\*x+b=±1）。再绘制测试集数据点，颜色表示真实类别，形状区分正确和错误分类（圆形为正确，叉形为错误）。此外，函数识别测试集中的支持向量，以红色空心圆标记。最后，生成包含决策边界、间隔、数据点和支持向量的图表，直观展示 SVM 的分类效果和支持向量位置。



（硬间隔SVM分类结果）

#### 3.2.3 模型评估分析



硬间隔支持向量机（SVM）模型在测试集上的准确率为 85.56%，表明其在二分类任务（中具有较好的分类性能，但略低于逻辑回归模型的 86.11%。模型系数为 [[-0.94369338 -1.12941151]]，显示两个特征对分类的贡献较为接近，第二个特征的影响略大于第一个，且负系数表明特征值增加时倾向于预测类别 0。相比逻辑回归，SVM 的准确率稍低，可能因其严格的线性分离要求未完全适应数据分布。

### 3.3 非线性软间隔SVM（基于RBF，来源论文二）

#### 3.3.1 代码构架

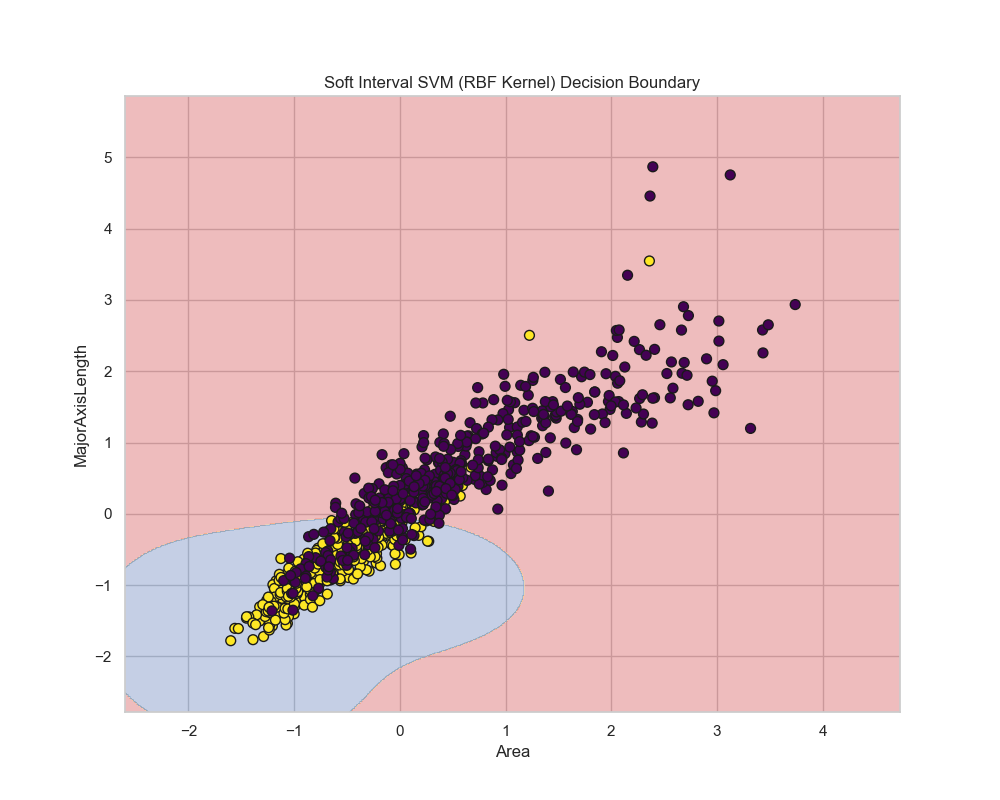


（基于RBF的软间隔SVM代码实现）

函数 train\_soft\_margin\_svm 实现了一个基于径向基函数（RBF）核的软间隔支持向量机（SVM）分类器，用于处理非线性可分数据。首先通过 train\_test\_split 将特征矩阵 X 和目标变量 y 划分为训练集和测试集。随后标准化特征，确保特征尺度一致，因为 SVM 对尺度敏感。接着，函数初始化一个 SVC 模型，采用 RBF 核，通过参数 C（默认 1.0）控制误分类惩罚的强度，gamma（默认 "scale"）调节核函数的宽度，从而实现软间隔 SVM 的非线性分类。模型在标准化后的训练集上拟合后，对测试集进行预测，并输出准确率、分类报告（包含精确率、召回率、F1 分数）和混淆矩阵以全面评估性能。函数返回训练好的模型、标准化器和测试集准确率。这种方式通过 RBF 核将数据映射到高维空间，结合软间隔机制（允许一定误分类）实现灵活的非线性分类。

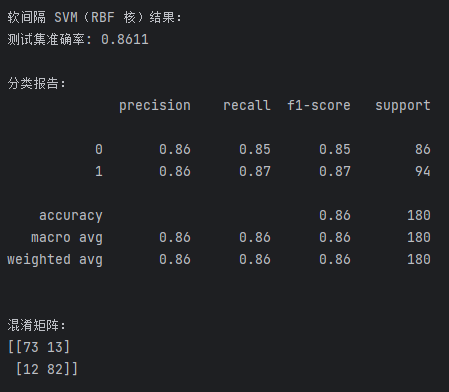
#### 3.3.2 可视化结果显示

我使用了一个函数 plot\_decision\_boundary 来进行可视化基于非线性支持向量机（SVM）模型（如使用 RBF 核的软间隔 SVM）的二维特征空间决策边界。它首先检查输入特征矩阵 X 是否包含正好两个特征，接着，使用提供的 scaler 对特征数据 X 进行标准化。再基于标准化后的特征范围，生成一个步长 0.01的二维网格，并对网格点进行模型预测，生成决策边界区域 Z。最后，绘制决策边界（使用红黄蓝颜色映射），并用 plt.scatter 绘制标准化后的数据点。展示非线性 SVM 的分类边界和数据分布，直观分析模型的分类效果。



（基于软间隔SVM的可视化结果）

#### 3.3.3 模型评估

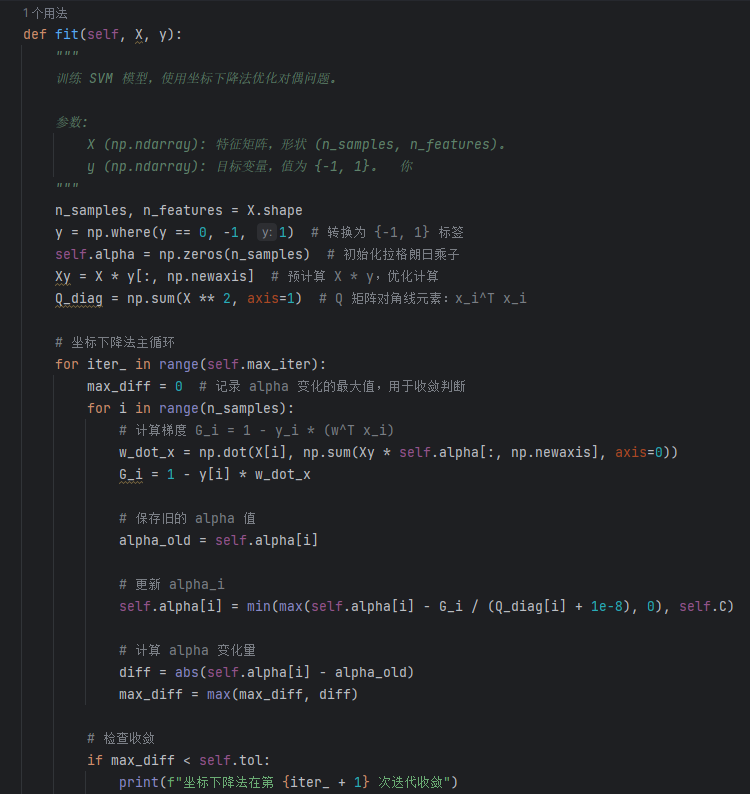


（基于RBF核的软间隔SVM模型结果）

逻辑回归模型和硬间隔支持向量机（SVM）模型在测试集上的准确率分别为 86.11% 和 85.56%，表现接近，但逻辑回归略优。逻辑回归的模型系数（[-0.33113448, -2.74494566]）显示第二个特征对分类的影响远大于第一个。两者均假设数据线性可分，但逻辑回归的稍高准确率可能表明其对数据的拟合更灵活。硬间隔 SVM 的较低准确率可能受限于其严格的线性分离要求，若数据存在噪声或非线性结构，性能可能受影响。综合来看，逻辑回归在当前数据集上表现略优且计算成本较低，但若数据线性可分性较差，可考虑软间隔 SVM 或非线性核函数以提升性能。

### 3.4 线性软间隔SVM（基于liblinear，来源论文二中的SGD）

#### 3.4.1 代码实现

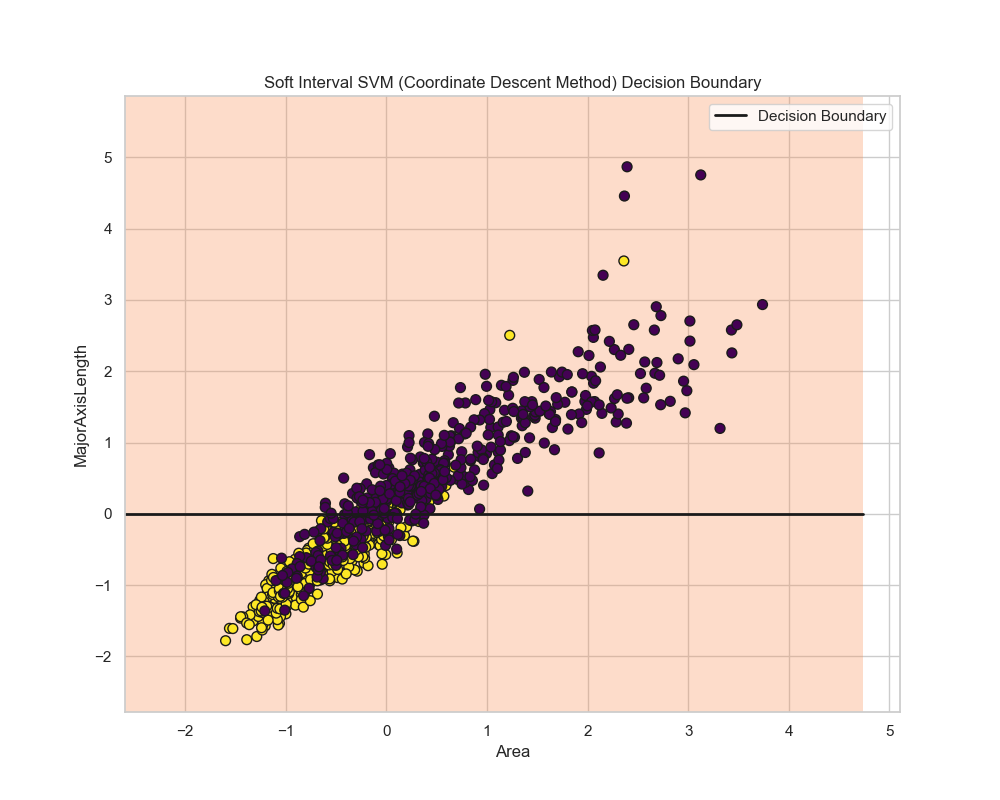


（基于liblinear的软间隔SVM代码实现）

此处通过设计函数 train\_coordinate\_descent\_svm 实现了一个基于坐标下降法的软间隔SVM模型，通过自定义的 CoordinateDescentSVM 类优化对偶问题。CoordinateDescentSVM 类通过迭代更新拉格朗日乘子（alpha），计算权重向量 w 和偏置项 b，并识别支持向量，允许部分误分类（通过正则化参数 C 控制）。模型在标准化后的训练集上拟合后，对测试集进行预测，并输出准确率、分类报告和混淆矩阵以评估性能。返回值为训练好的模型、标准化器和准确率。之所以采用这种方法是因为可以通过坐标下降法来高效优化线性 SVM，适合线性可分或接近线性可分的数据。

#### 3.4.2 可视化结果显示

我在可视化显示结果时使用了函数 plot\_decision\_boundary，用于显示基于坐标下降法训练的软间隔线性支持向量机（SVM）模型在二维特征空间中的决策边界和数据点分布。使用提供的 scaler 对特征进行标准化后，基于标准化特征的范围生成一个高分辨率（步长 0.01）的二维网格，并通过模型预测网格点类别以绘制分类区域。分类区域通过 plt.contourf 以红黄蓝颜色映射（透明度 0.3）显示，颜色由目标变量 y 决定。函数还利用模型的权重 w 和偏置 b 计算并绘制明确的决策边界直线（w1\*x1 + w2\*x2 + b = 0），以黑色实线表示，直观展示线性 SVM 的分类效果。



（Liblinear算法的软间隔SVM的线性决策边界）

#### 3.4.3 模型评估分析

自定义坐标下降法实现的软间隔线性 SVM 模型在测试集上的准确率仅为 52.22%，表现较差。相比逻辑回归（86.11%）、硬间隔 SVM（85.56%）和 RBF 核软间隔 SVM（86.11%），该模型性能显著较差，需调整 C 值、增加迭代次数或考虑非线性核以改善分类效果。可能此数据集并不是特别合适该方法。

## 文献阅读

### 4.1 论文一

#### 4.1.1 论文简介

《Fast SVM classifier for large-scale classification problems》，发表在《Information Sciences》期刊（2023年，第642卷，119136号），作者为Huajun Wang、Genghui Li和Zhenkun Wang。论文提出了一种新的支持向量机（SVM）模型，称为截断平方铰链损失支持向量机，旨在解决大规模分类问题中的计算复杂度和鲁棒性问题。

#### 4.1.2 论文分析

这篇论文提出了一种新的SVM模型，通过引入截断平方铰链损失函数来提高模型的鲁棒性和稀疏性，同时降低计算成本。其实就是提出了一种新的损失函数，文章中写成，也就是截断平方铰链损失函数。

文章中是这样定义这个函数的：

其中 ，表示分类误差，从鲁棒性角度分析，当r>1 时，损失固定为1，限制了异常值对模型的过度影响，比传统铰链损失和平方铰链损失更鲁棒。

而从稀疏性的角度来分析，当 r<0 时，损失为0，减少了支持向量的数量，从而提高了模型的稀疏性。

那么在有了新的损失函数以后，就可以提出新的SVM模型，也就是新的优化问题变成了：

这就是这个文章中所提出的学习机，那么我们就有了正则化参数，其中x和y分别是样本特征和标签。通过以上步骤，我们就完整地构建了一套有创新性的SVM算法。

这篇论文在创新方面，不仅在于提出了一种新的loss函数去构建SVM算法，还提出了一种新的交替方向乘子法（-ADMM），结合工作集策略，通过随机划分训练集为工作集和非工作集，降低计算复杂度。证明了算法的全局收敛性（公式4.2），确保其收敛到近端驻点（即局部最优解）。算法在每次迭代中只处理部分数据，计算复杂度低，适合大规模数据集。并通过与八种其他SVM求解器（包括HSVM、SSVM、PSVM等）的比较，验证了-SVM的优越性，即使用较少的支持向量，以及较短的训练时间，就能鲁棒性较好，准确率较高的结果。

### 4.2 论文二

#### 4.2.1 论文简介

《SVM-SMO-SGD: A hybrid-parallel support vector machine algorithm using sequential minimal optimization with stochastic gradient descent》，发表在《Parallel Computing》期刊（2022年，第113卷，102955）。作者为Gizen Mutlu和Çiğdem İnan Acı，提出了一种结合支持向量机（SVM）、顺序最小优化（SMO）和随机梯度下降（SGD）的混合并行算法，称为SVM-SMO-SGD，旨在优化SVM在大规模数据集上的时间和资源消耗，同时保持高精度。

#### 4.2.2 论文分析

这篇文章是为了处理当数据量极具增加时，普通SVM所需时间大大增加的问题，因此提出一种混合算法SVM-SMO-SGD，通过结合SMO（优化SVM权重计算）和SGD（加速权重更新），在CPU和GPU上实现更高效的并行计算，减少时间和资源消耗，同时保持SVM的分类精度。在算法优化方面，主要体现在SVM与SMO的结合上。

基于传统的SVM，其目标是找到一个最优超平面，用于最大化分类边界。在非线性情况下，SVM通过核函数（如高斯径向基函数，RBF）将数据映射到高维空间，优化问题可以表示为以下二次规划形式：

有约束条件：

其中C为正则化参数，这是基于简单的SVM理论。而根据这样一个RBF的高斯核函数，我们可以进行非线性的软间隔SVM分类器，这也就是3.3中实现的一个分类情况。

而如果我们使用SMO来优化这个模型，让其在计算机中能够更快地进行计算，就可以通过分解QP问题来解决SVM的优化问题。每次迭代选择两个拉格朗日乘子（和）进行优化，而保持其他乘子不变，以满足线性等式约束 。如此，我们就可以更新我们的参数：

这里的E是误差，也就是根据误差去更新拉格朗日乘子，从而达到优化的目的。

而使用随机梯度下降（SGD）通过随机选择样本更新权重向量w ，解决SVM的原始形式优化问题。SVM的损失函数在论文中表述为

同时也通过SGD进行权重更新规则:

在这里是学习率，一个常在神经网络中出现的参数，这里决定了参数受影响的程度。SGD通过随机选择样本进行梯度更新，在牺牲部分精度的前提下降低了计算成本，适合大规模数据集。