

**2024年《机器学习》**

**工程报告**

**（完整版）**



**课 程：** 机器学习

**姓名+学号：** 党存远（2022217587）

**姓名+学号：** 汪明灿（2022217614）

**姓名+学号：** 朱屹昊（2022217579）

**姓名+学号：** 谢雨轩（2022217591）

**姓名+学号：** 张棣（2022217613）

**姓名+学号：** 陈子豪(2022217612)

**完成时间：** 2024.5.20 本项目所使用的数据集:

<https://www.kaggle.com/datasets/nelgiriyewithana/apple-quality/data>

项目源码: [DStevenFelix/-2024- (github.com)](https://github.com/DStevenFelix/-2024-)

一． 工程摘要与每人贡献

摘要：总体介绍本工程的工作。

每人分工：

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 角色  （组长，组员，独自完成） | 工作量  比例 | 负责内容 |
| 党存远 | 组长 | 30% | 完成工程报告（完整版）的撰写汇总，收集苹果质量分类的数据集；实现数据的预处理，同时完成了各类神经网络模型以及集成学习模型的构建。 |
| 朱屹昊 | 组员 | 18% | 完成随机森林模型代码并实现可视化；分析对比各个模型的分类情况。 |
| 汪明灿 | 组员 | 14% | 通过k临近完成分类任务优化超参数，再用朴素贝叶斯对数据分类并进行混淆矩阵的可视化，绘制ROC曲线来评估支持向量机（SVC）模型的分类性能。 |
| 谢雨轩 | 组员 | 14% | 完成决策树模型代码，通过交叉验证找到的最佳超参数，以及对应的最佳决策树模型。 |
| 陈子豪 | 组员 | 14% | 完成SVC的模型代码和学习曲线的生成以及最终结果的比较 |
| 张棣 | 组员 | 10% | 完成数据导入与逻辑回归的建立 |

二、研究背景与意义

苹果作为日常生活中常见的水果之一，其质量的好坏直接影响到消费者的购买意愿和食用体验。因此，对苹果质量进行准确分类具有重要意义。传统的苹果质量分类方法主要依赖于人工经验和感官判断，这种方法不仅效率低下，而且容易受到主观因素的影响。随着机器学习技术的不断发展，利用算法模型对苹果质量进行自动分类成为了可能。因此我们选择的数据集是苹果品质的数据集。该数据集包含有关一组水果的各种属性的信息，提供对其特征的深入了解。该数据集包括水果 ID、大小、重量、甜度、松脆度、多汁性、成熟度、酸度和质量等详细信息。在现代农业和食品工业中，苹果质量的监测和控制是至关重要的。随着技术的发展，传统的人工检查方法逐渐被更加高效、准确的自动化检测技术所取代。苹果质量数据集的研究背景涉及农业、食品科学、计算机视觉和机器学习等多个领域。通过自动化检测技术，可以大幅提高苹果质量检测的效率，减少人工成本和时间，同时高质量的苹果质量检测可以保证市场上的苹果符合安全标准，减少食品安全隐患。

**主要研究方法:**

数据采集和整理：收集苹果的多维度数据，如重量、颜色、大小、甜度、酸度等。

数据的预处理:对收集好的数据进行数据清洗与标准化和归一化处理。

统计分析：使用描述性统计和推论统计方法分析数据特征。

模型的构建:通过使用机器学习常见模型对数据进行分析，最终模型用于预测出苹果的质量。

**主要存在问题:**

数据质量：数据的准确性和完整性直接影响分析结果。

数据多样性：不同品种和不同生长环境的苹果数据具有较大差异性。

模型鲁棒性：模型在不同条件下的稳定性和适用性需要进一步验证。

**现有的解决方法:**

数据清洗：通过数据清洗和预处理，提升数据的质量和可用性。

多变量分析：利用多变量统计分析方法，深入挖掘数据之间的关系。

模型优化：通过交叉验证和参数调优，提升模型的预测能力和稳定性。

三、模型方法

**详细说明**本工程所使用的模型方法和理论等

1. 数据处理:

a.数据读取:使用pandas库中的read\_csv()函数来加载数据。用于从CSV文件中读取数据，并将其存储为DataFrame对象。

b.数据的展示:使用df.head()方法可以快速查看DataFrame的前几行。这有助于初步了解数据的结构和内容。通过df.info()方法展示了DataFrame的概要信息，包括每列的名称、非空值数量及字段数据类型，这有助于识别哪些列可能包含缺失值或错误的数据类型。之后通过df.describe().T生成描述性统计数据，帮助了解数值型特征的中心位置、离散度等统计量。通过.T（转置）改善显示效果。

c.数据清洗:使用df.duplicated().sum()检测并统计数据中的重复行，通过df.isnull().any(axis=1)检测含有缺失值的行。

d.特征工程:通过可视化分析（如箱形图、散点图、热力图等）来探索特征之间的关系，并识别出数据中的模式和潜在的关联。

e.数据的标准化与归一化:为了消除不同量纲的影响，对数值型特征进行了标准化或归一化处理。使用MinMaxScaler将特征缩放到[0, 1]区间，以及使用RobustScaler来减少极端值的影响。

f.数据平衡:由于数据集中“好”和“坏”标签的分布可能不均衡，使用SMOTE等技术进行过采样，以平衡类别分布。

g.降维处理：使用PCA进行数据的降维处理，pca是一种统计技术，用于在尽可能保留数据变异性的同时降低数据的维度。它在数据分析和信号处理中非常常用，可以将一组可能相关的变量转换成一组线性无关的变量，这些变量称为主成分。

1. 模型选择与训练:

a.支持向量机（SVM）：一种强大的分类器，通过找到数据点之间的最优边界来区分不同的类别。在本项目中，使用SVC（支持向量分类器）并进行了参数调优，以确定最佳的C、kernel和gamma参数。

b.决策树：决策树通过学习简单的决策规则从数据特征中推断出目标值。在本项目中，通过RandomizedSearchCV进行参数优化，以确定树的深度和分裂所需的最小样本数。

c.逻辑回归：逻辑回归是一种线性模型，用于预测二分类问题的概率。在本项目中，通过调整正则化参数、求解器等来优化模型。

d.随机森林：一种集成学习方法，它构建多个决策树并将它们的预测结果结合起来，以提高模型的稳定性和准确性。

e.K近邻（KNN）：一种基于距离的分类算法，通过查找测试数据点的K个最近邻来进行分类。

f.神经网络：

MLP多层感知机：一种经典的前馈神经网络，用于处理各种类型的数据，包括图像、文本和结构化数据。MLP通过多个隐藏层和非线性激活函数，能够学习和表示复杂的非线性关系。该模型构建的重点在于：在MLP中，数据从输入层开始，通过各隐藏层逐层传播到输出层。在每一层，神经元对输入信号进行加权求和，并通过激活函数引入非线性变换。激活函数是MLP的核心之一，负责引入非线性，增强网络表达复杂模式的能力。

卷积神经网络：一种深度学习模型，广泛应用于图像和视频处理等领域。CNN通过引入卷积层、池化层和全连接层，能够高效提取和处理数据的空间特征。卷积神经网络主要包括：卷积层(卷积层是CNN的核心组件，主要作用是提取输入数据的局部特征。卷积操作通过一个或多个卷积核在输入数据上滑动，计算局部区域的加权和。每个卷积核可以提取不同的特征，如边缘、纹理等。)，激活函数(激活函数引入非线性，使网络能够表达更复杂的特征。卷积层输出通常经过Relu激活函数),池化层(池化层用于对特征图进行下采样，减少特征图的尺寸，从而降低计算复杂度并提高模型的抗噪性。常用的池化方法包括最大池化和平均池化。最大池化选取局部区域的最大值，平均池化则计算局部区域的平均值。),全连接层(用于综合前面提取的特征并进行分类或回归任务。全连接层中的每个神经元与前一层的所有神经元相连，类似于传统的多层感知机),损失函数(衡量模型预测值与真实值之间的差异，是模型优化的目标。对于分类任务，常用的损失函数是交叉熵损失),以及反向传播(用于计算损失函数对每个参数的梯度，并使用优化算法更新权重和偏置。)

残差网络: 一种深度神经网络架构，特别适用于构建非常深的神经网络模型。ResNet 通过引入了所谓的“残差模块”或“跳跃连接”来解决深度网络训练中常见的梯度消失和梯度爆炸问题。

监督自编码器:由编码器和解码器两部分构成，编码器将高维输入数据压缩到低维潜在空间表示，解码器则将这个潜在表示重建回原始数据。自编码器的目标是最小化输入数据与重建数据之间的差异，通常使用均方误差作为损失函数,由于自编码器是一种无监督的学习方法因此在编码器中加入了分类任务。

g.集成学习: 种通过结合多个基学习器的预测结果来提高整体模型性能的方法。集成学习方法通过将多个弱学习器组合成一个强学习器，从而在准确性和稳健性上优于单一模型。在我构建的集成学习模型中我使用基于投票分类器的基模型结合了SVM，决策树以及K临近，对于本次的二分类我使用了逻辑回归作为元模型进行训练训练。

h.朴素贝叶斯：一种基于贝叶斯定理的分类器，它假设特征之间相互独立，适用于大量特征的数据集。

四、系统设计

系统的**详细**设计，系统流程，系统的每一步的具体流程，例如，如何处理语料，如何训练模型，如何测试模型，最后对模型进行评估等。

对于该数据集，我们的目的旨在机器学习模型准确地分类苹果的质量。该系统整合了数据预处理、特征工程、模型训练、测试和评估等步骤，使用多种模型来实现高精度的分类效果。以下是这个系统的构建:

**系统设计**

1. 目标

设计一个能够预测苹果品质的机器学习系统。

2. 功能

a.数据导入与预处理

b.特征工程

c.可视化分析

d.模型训练与评估

e.预测与结果展示

3. 特点

a.使用多种数据预处理技术提高模型性能。

b.应用多种机器学习算法进行比较和选择。

c.采用混淆矩阵，准确率以及交叉验证,学习曲线评估模型稳定性和性能。

**系统内容:**

1: 数据导入与预处理

a .导入数据集：df = pd.read\_csv(r"./apple\_quality.csv")

b.检查数据基本信息：df.info() 和 df.describe()

c.清洗数据：删除重复项、处理缺失值、去除无关特征A\_id。

d.数据类型转换：确保Acidity字段为浮点数。

2: 特征工程

a.创建新特征Label，将Quality字段转换为数值型。

3: 可视化分析

b.使用Seaborn和Matplotlib库对数据进行可视化，包括直方图、箱形图、散点图、小提琴图和热力图。

4: 数据预处理

a.标准化处理：使用MinMaxScaler和StandardScaler。

b.异常值处理：使用RobustScaler和IQR方法。

c.数据平衡：使用SMOTE技术处理不平衡数据集。

5: 划分数据集

a.将数据集划分为训练集和测试集。

6: 降维处理

a.使用PCA技术进行降维。

7: 模型训练

a.训练多种模型，包括SVM、决策树、随机森林、KNN、神经网络等，同时使用RandomizedSearchCV进行参数调优。

8: 模型评估

a.使用准确率、混淆矩阵、ROC曲线和AUC值评估模型性能，应用交叉验证和学习曲线评估模型稳定性。

9: 预测与结果展示

使用训练好的模型对测试集进行预测，可视化展示模型的决策边界。

**具体实现:**

1、首先进行数据预处理：

a.数据的加载与检查:首先从CSV文件中加载苹果质量数据，使用Pandas进行初步的数据查看，包括基本信息（如列名称、数据类型和统计描述）。通过df.info()和df.describe()方法，我们能够获取关于数据集的初步统计概览，例如各特征的计数、均值、标准差等。

b.之后进行数据的清洗操作:我们的系统数据清洗的步骤包括处理缺失值、删除重复项以及转换数据类型。例如，将'Acidity'列从字符串转换为浮点数，将'Quality'列的分类标签（如"good"、"bad"）转换为数值型标签。这一步骤是通过自定义的clean\_data()函数实现的，确保数据的一致性和适用性。

c.最后在数据处理阶段进行了数据的可视化:使用Seaborn和Matplotlib库对清洗后的数据进行可视化，通过直方图以及散点图展示各特征的分布情况。

2、之后进行不平衡数据的处理以及划分

a.使用SMOTE在少数类的样本之间插值生成新的样本。这有助于平衡类别，改善模型在少数类上的预测性能。

b.特征缩放:使用归一化，确保所有特征在相同的尺度上，使得模型训练更加稳定和快速。

c.将清洗和预处理后的数据集分为训练集和测试集

3、进行数据的降维处理

a.通过PCA对数据进行降维处理以便适合模型

1. 选择合适的模型进行训练

在本项目中，我们选用了**支持向量机 (SVM)，决策树，逻辑回归，随机森林，K近邻 (KNN)，神经网络，卷积神经网络 (CNN)，残差网络，监督自编码器，集成学习，朴素贝叶斯**等模型。

a.SVC，我们使用RandomizedSearchCV进行参数调优，得到最优参数best\_params，使用SVC类创建SVC模型实例best\_svc\_model，使用训练集X\_train和y\_train对模型进行训练，使用测试集X\_test进行预测，得到结果svc\_predicted。最后通过打印混淆矩阵svc\_conf\_matrix，计算并打印准确率svc\_acc\_score最后打印分类报告classification\_report，决策树，逻辑回归以及随机森林，KNN都采用类似的方法进行训练以及模型评估。

b.神经网络，我们使用Tenserflow定义神经网络模型结构，包括多层感知机和卷积神经网络，使用Sequential或MLPClassifier创建神经网络模型实例，通过训练集对模型进行训练，使用测试集进行预测，得到结果，最后打印出测试集准确率。

最后通过对比我们发现SVC在测试集上的准确率较高，使用交叉验证等技术来训练模型，确保模型在未见数据上的泛化能力,

5、结果的可视化和解释:

a.通过ROC曲线，混淆矩阵以及学习曲线进行最终结果的可视化以及解释。

五．实验结果分析、对比和讨论

对实验结果进行分析说明，测试一些课程中演示的样例，根据结果说明为什么对或者为什么错等具体分析；对不同模型和参数的对比进行分析；最后对系统提出改进方案等。

**1、数据导入:**

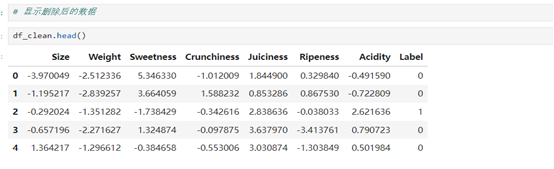
使用df（DataFrame）用于处理和分析结构化数据。它类似于电子表格或数据库表格，可以将数据组织成行和列的形式，让我们审阅数据更加方便。由行索引和列标签组成，可以用于存储不同类型的数据（整数、浮点数、字符串等）。通过调用df中的方法达到导入数据效果。

**2、数据处理:**

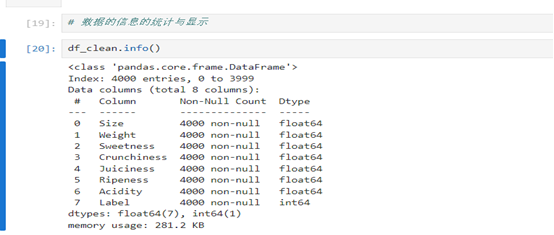
数据清洗



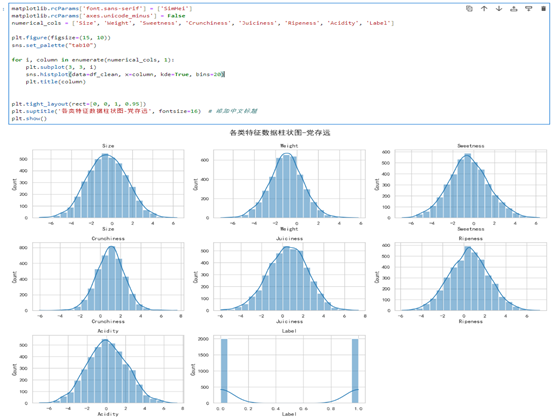
删除特征后的数据显示



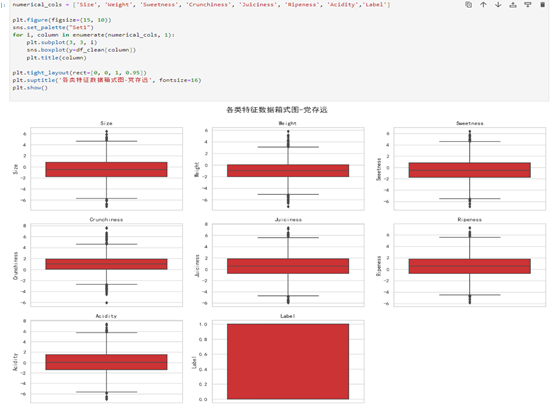
数据特征的显示



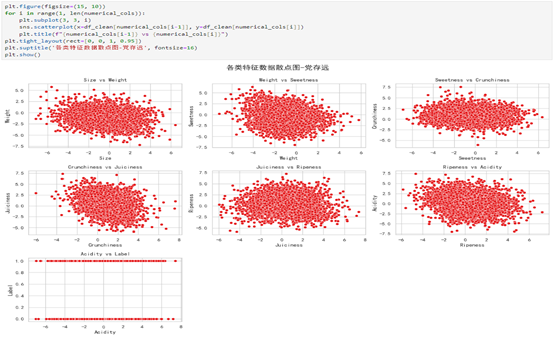
各类数据特征的柱状图显示



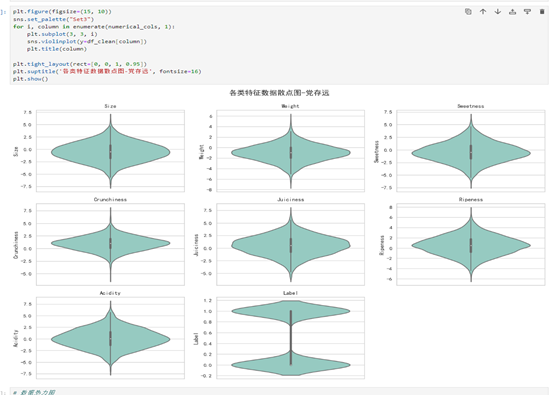
各类数据箱式图



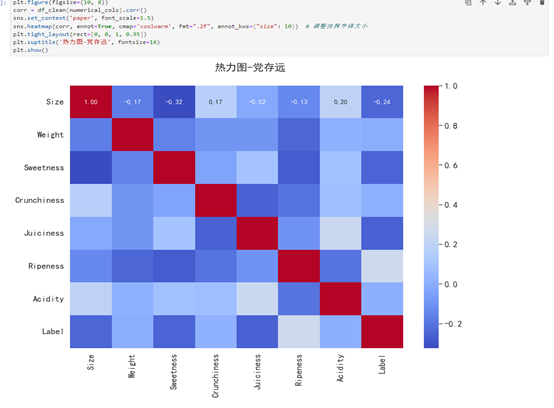
各类数据散点图



各类数据小提琴图



数据热力图



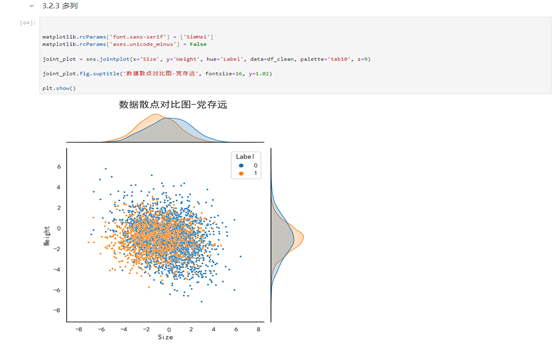
数据特征箱式对比图



数据散点图



数据对比散点图



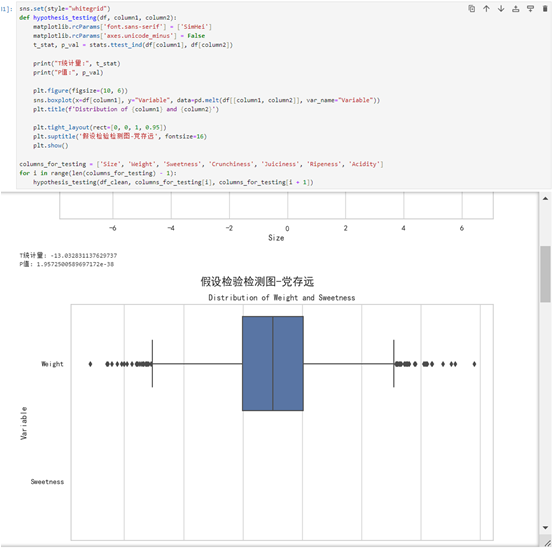
相关性热力图

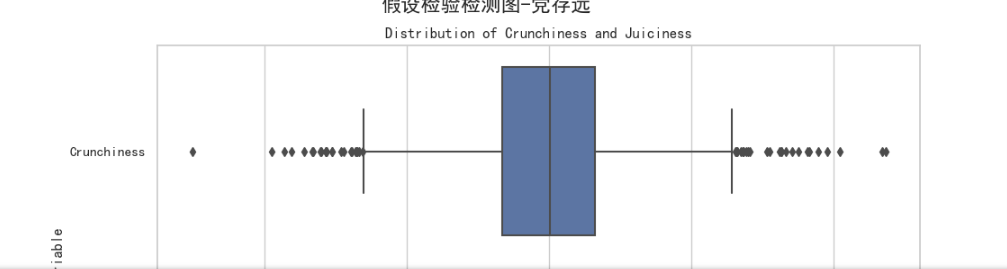


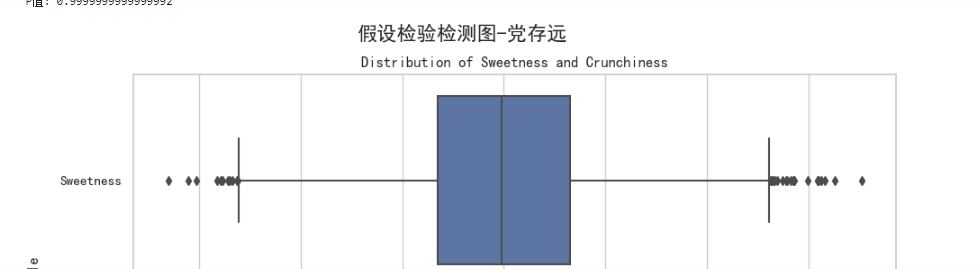
异常处理



假设检验图







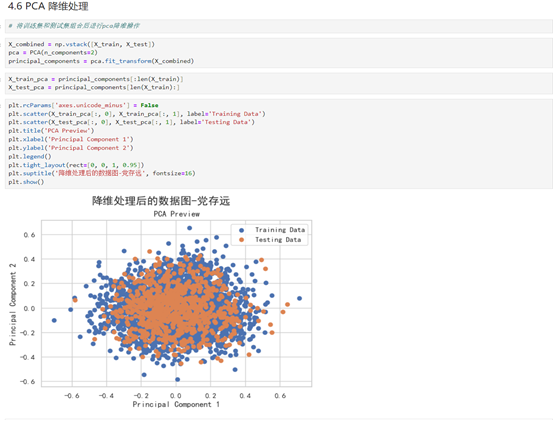
不平衡数据处理



标准化



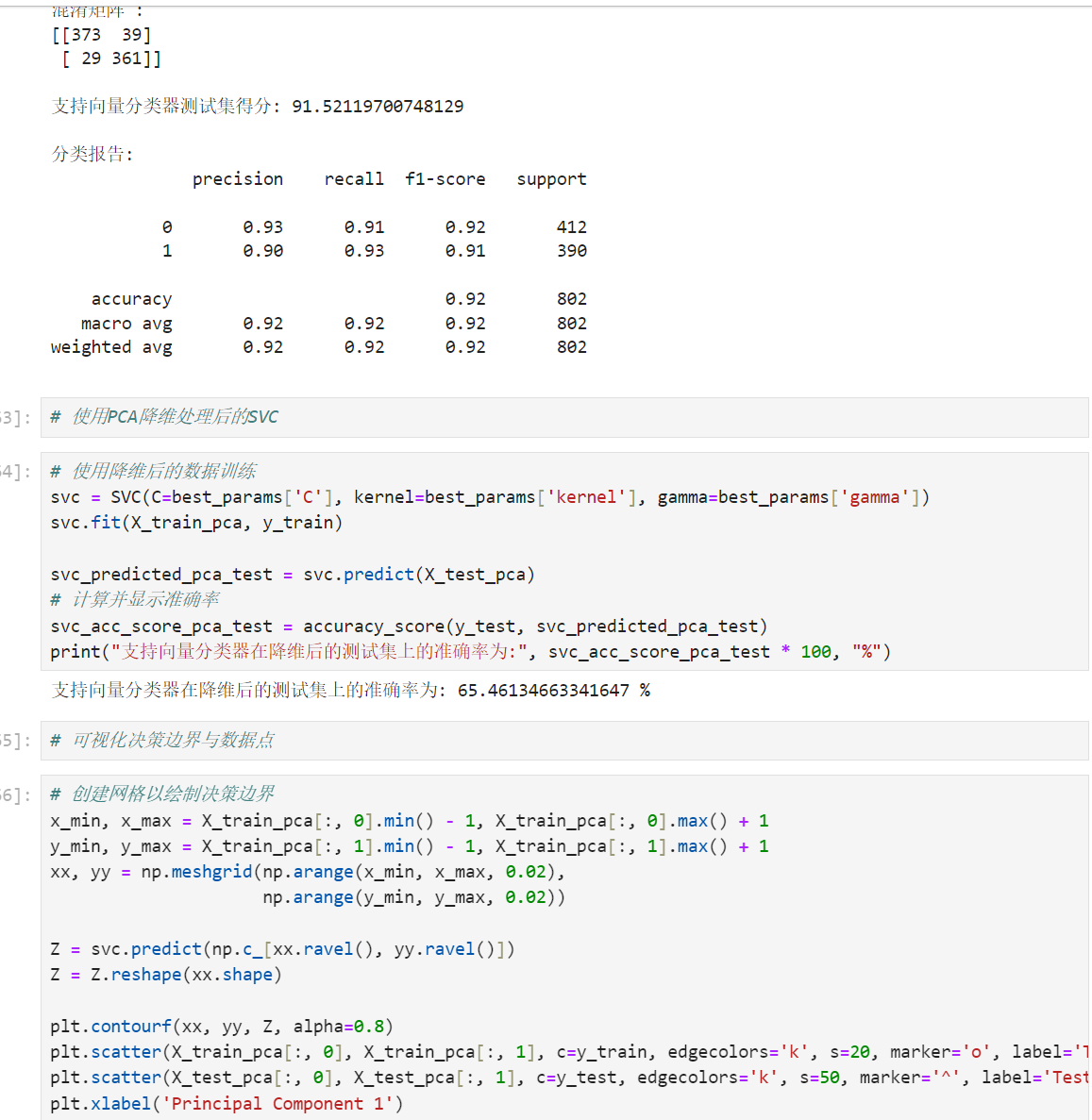
降维处理后的数据图

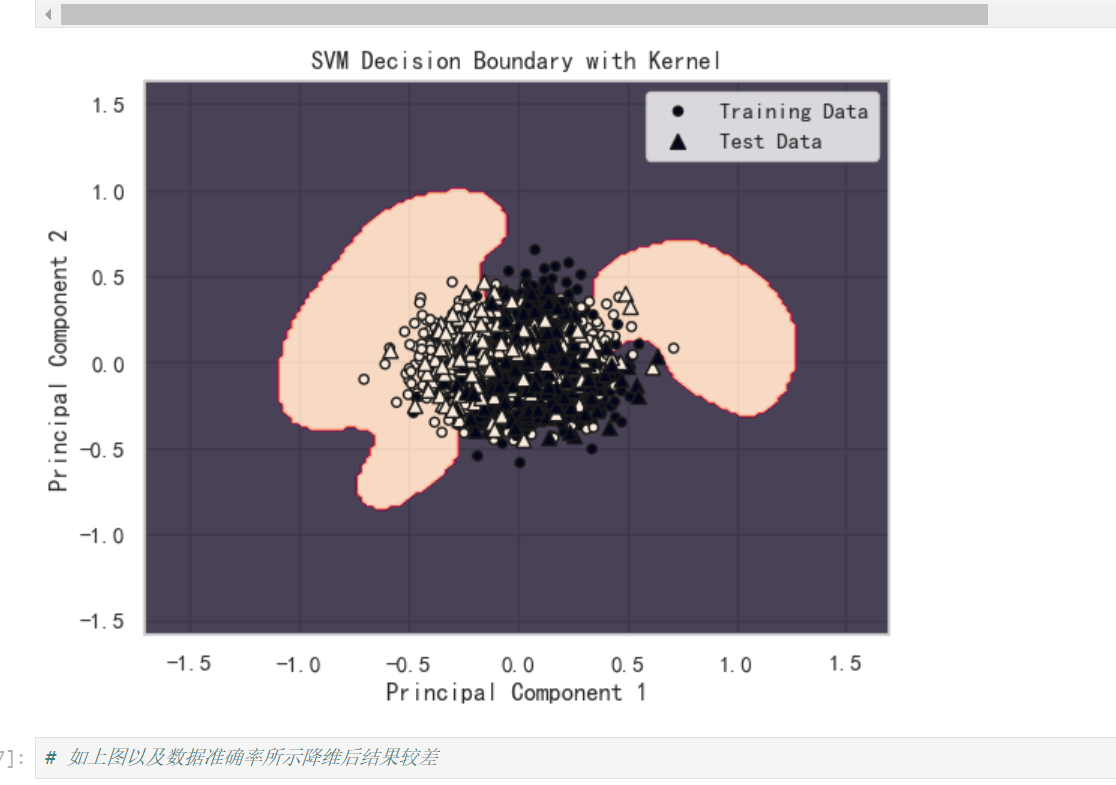


**3、模型构建**

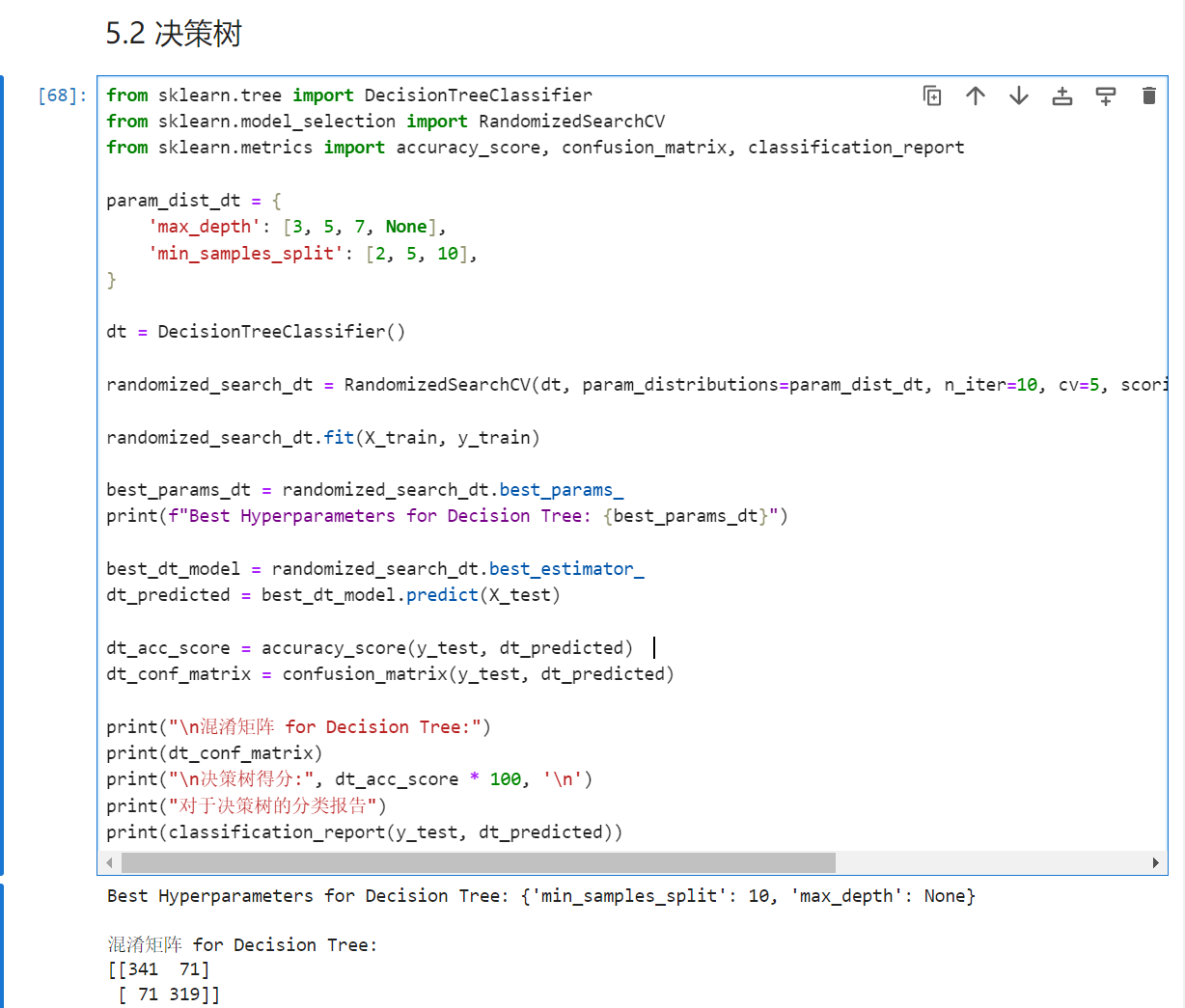
SVC

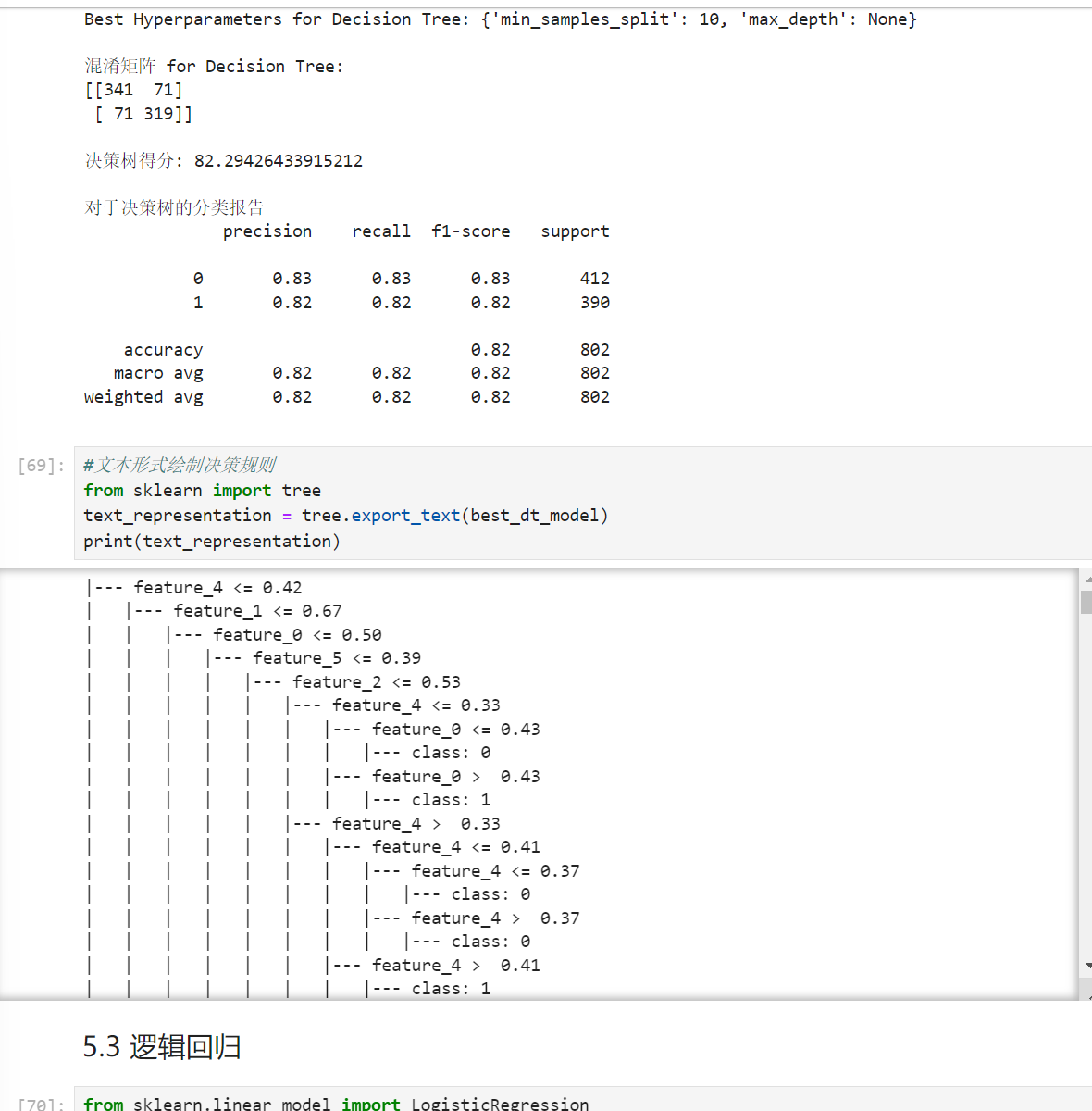






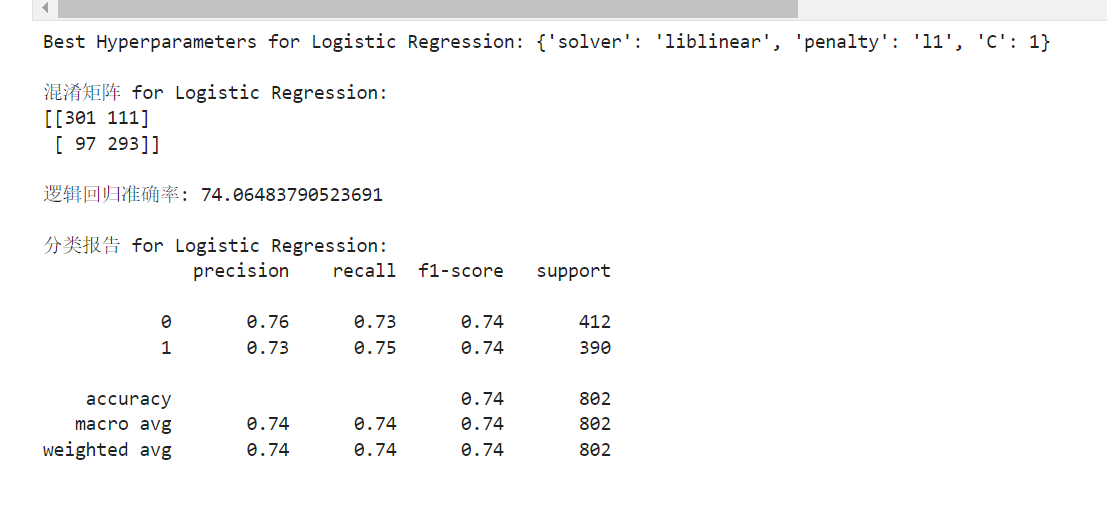
决策树





逻辑回归

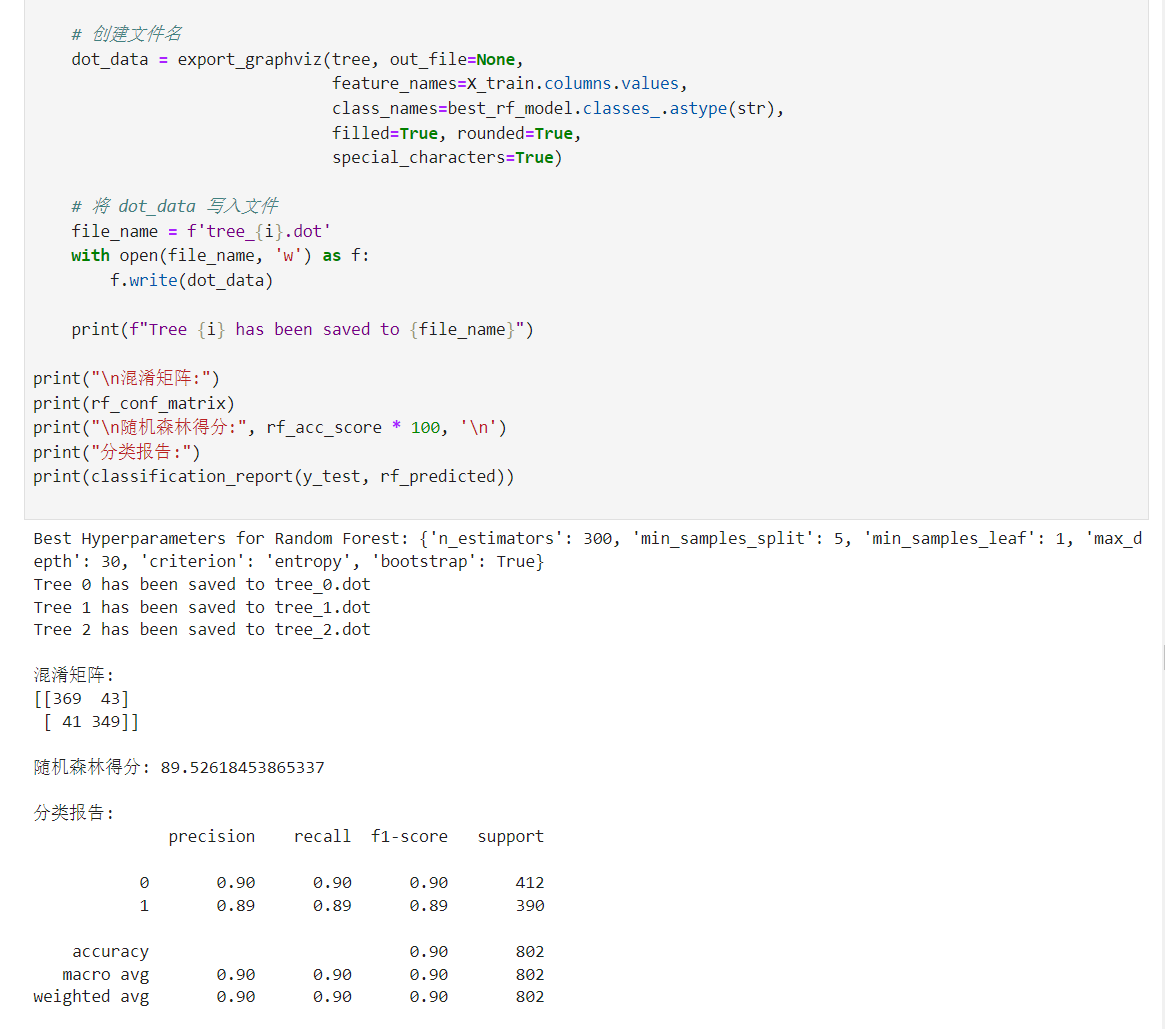


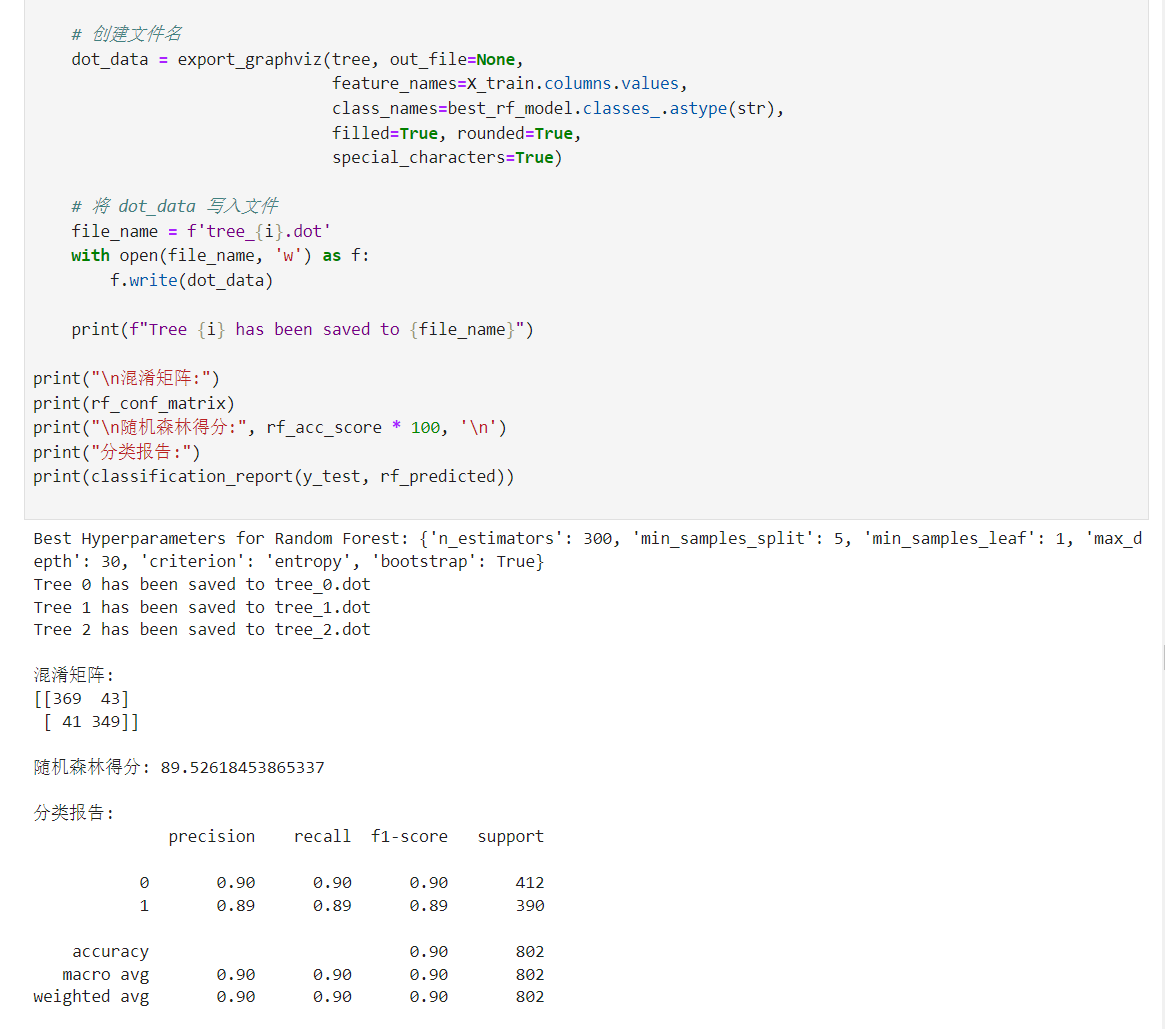


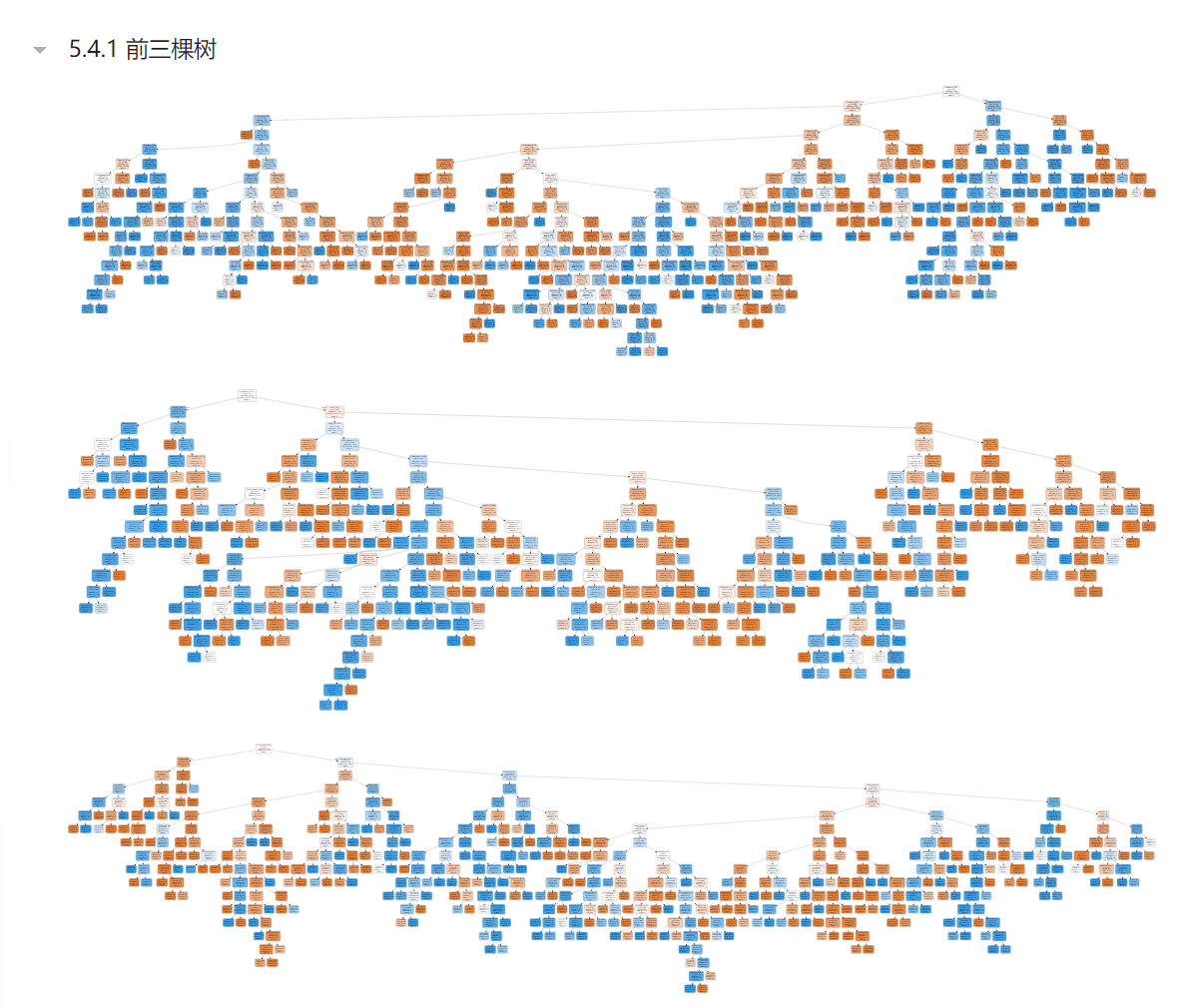
scikit-learn库中的LogisticRegression类进行逻辑回归模型的训练和评估。定义逻辑回归模型的参数网格param\_dist\_lr，其中包含了C（正则化参数）、penalty（正则化类型）和solver（求解方法）等参数的不同取值。再创建一个LogisticRegression对象lr，设置了最大迭代次数为1000。使用RandomizedSearchCV类创建一个随机搜索对象randomized\_search\_lr用于在给定的参数网格param\_dist\_lr上进行参数搜索。参数n\_iter表示要进行的随机搜索迭代次数，cv表示交叉验证的折数，scoring表示评估指标，random\_state表示随机种子，n\_jobs表示并行运行的作业数。随机搜索对象randomized\_search\_lr对训练数据X\_train和y\_train进行拟合。打印出最佳超参数，获取最佳模型best\_lr\_model，使用测试数据X\_test进行预测，得到预测结果。计算逻辑回归的准确率lr\_acc\_score和混淆矩阵lr\_conf\_matrix。打印逻辑回归的分类报告，其中包括精确率、召回率、F1分数等评估指标。

随机森林



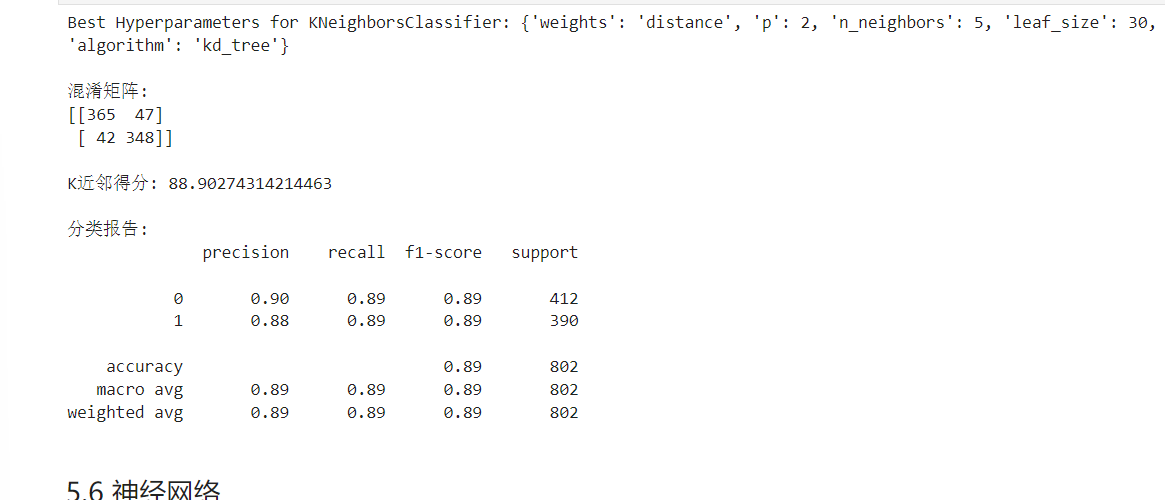




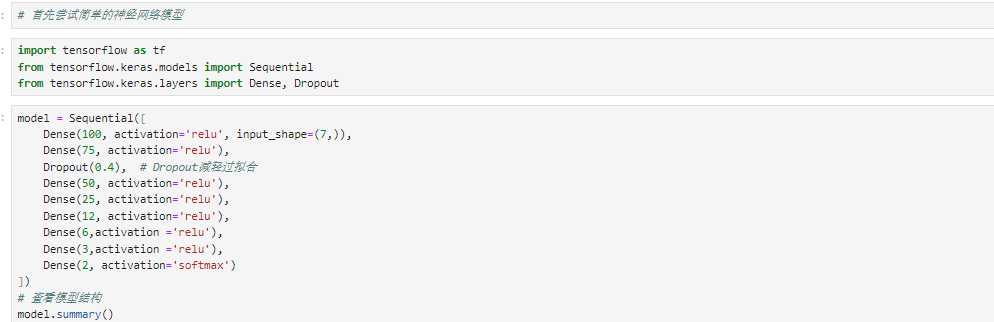


KNN

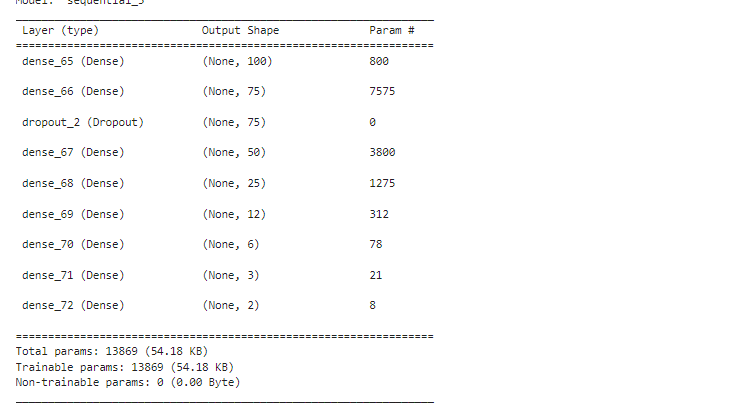




MLP前馈神经网络



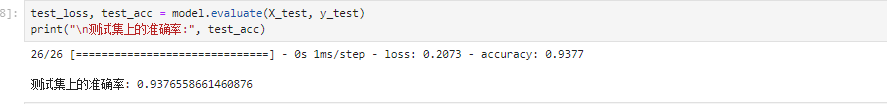
神经网络结构示意



模型训练过程



准确率

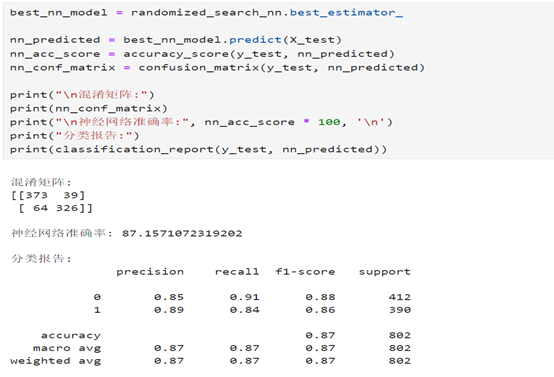


如图所示普通前馈神经网络在测试集上的准确率达到了0.92 可以看出该神经网络的泛化能力较强，由于该种较为简单的神经网络层数和结点数量多，进行递归的次数多，同时使用了Dropout层减少过拟合保证了该模型较优，该类模型适用于像本项目种所用的数据集的特征数量较少但是数据量较大的情况。

MLPClassifier



准确率



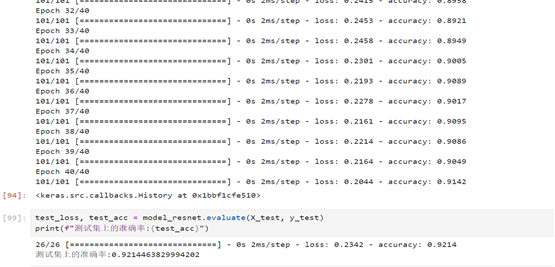
通过测试集的准确率可以发现，该种模型相较于MLP前馈神经网络，模型的效果变差，这是由于我们小组所选择的数据集类型是表格形数据类型，一般该种类型下，最为适合的模型是MLP前馈神经网络。

卷积神经网络



由图可以看到我创建了一个非常简单的卷积神经网络，通过设立卷积层，展平层以及全连接层和输出层来实现模型的搭建，并且在全连接层中利用relu作为激活函数。

训练过程



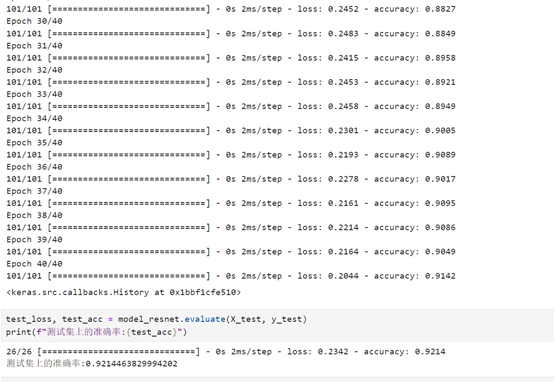
通过图片信息可以看到经过多次训练，数据集在卷积神经网络中的准确率较高。

残差网络



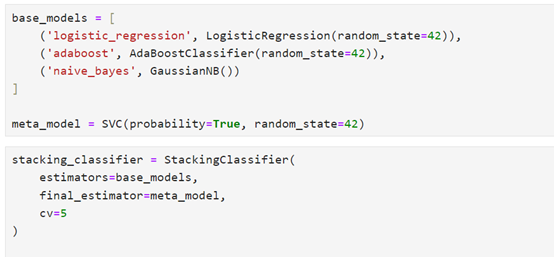
在构建残差网络的过程中，我首先创建了输入层并且设置了网络的输入层，之后我设计了全连接层和残差块部分，使用Relu作为激活函数。

模型的训练



观察模型在数据集上的运行结果可以看到，本模型在测试集上的准确率较好，说明该模型的表现较优，分类能力较强。

集成学习模型构建

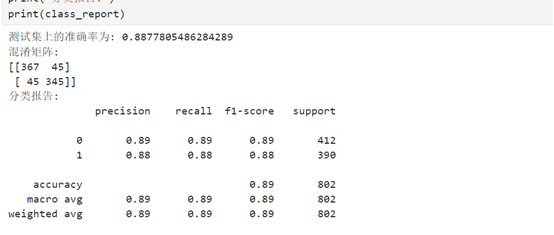


通过堆叠分类器来构建模型，使用逻辑回归,朴素贝叶斯以及adaboost作为堆叠分类器中的元模型，对于该二分类问题，选择逻辑回归作为基模型进行训练。

训练过程



准确率



分析该模型导致模型的准确率发生变化的原因是基模型个和元模型。

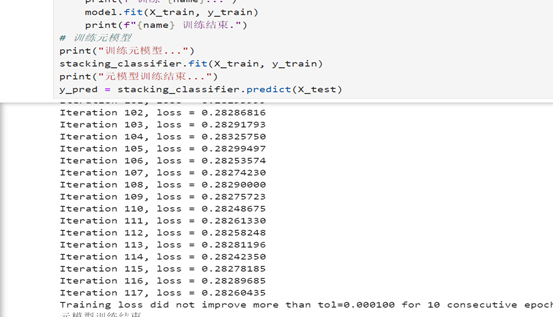
由于我使用的逻辑回归，朴素贝叶斯不适合该类数据集。尝试使用不同的元模型和基模型进行修改和测试。

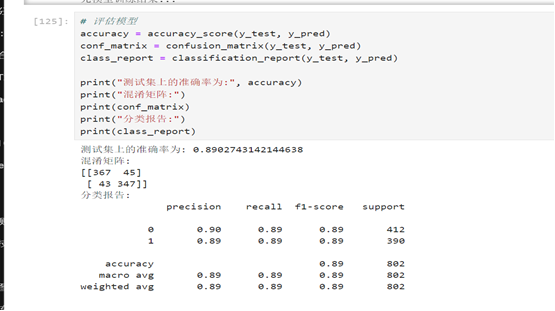
集成学习模型构建



该模型使用随机森林，朴素贝叶斯以及神经网络作为基模型，对于二分类使用用逻辑回归作为元模型。

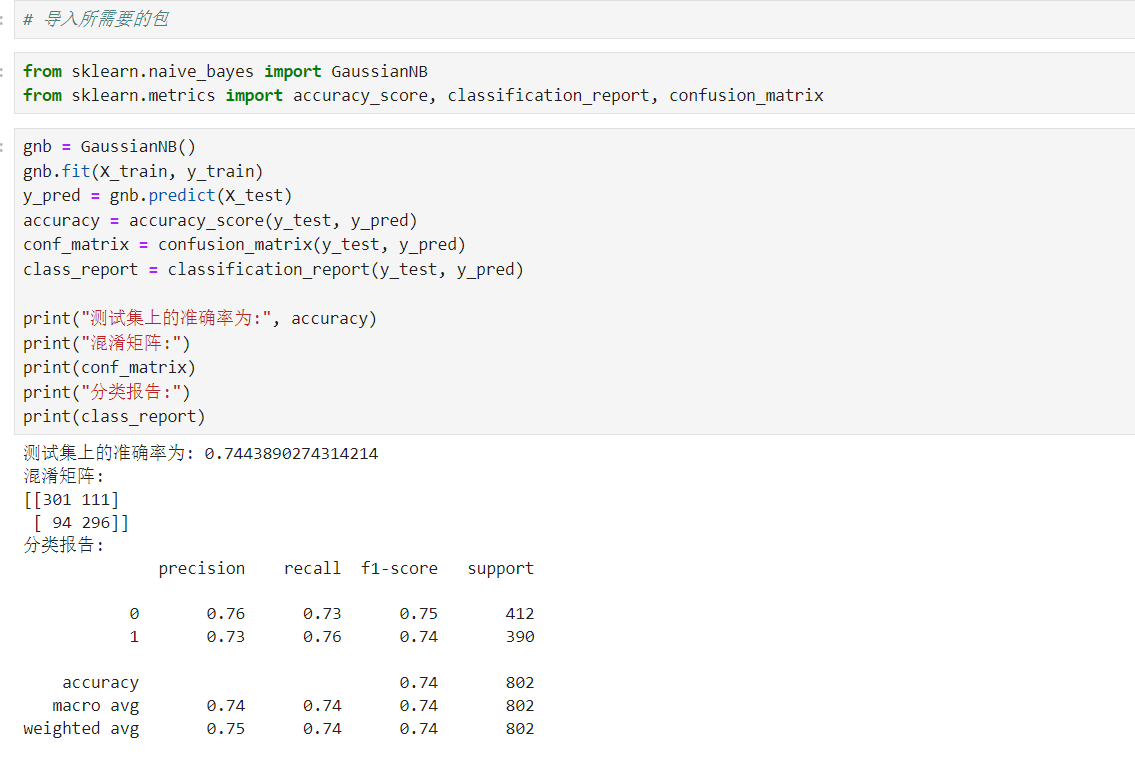
集成学习训练过程及准确率



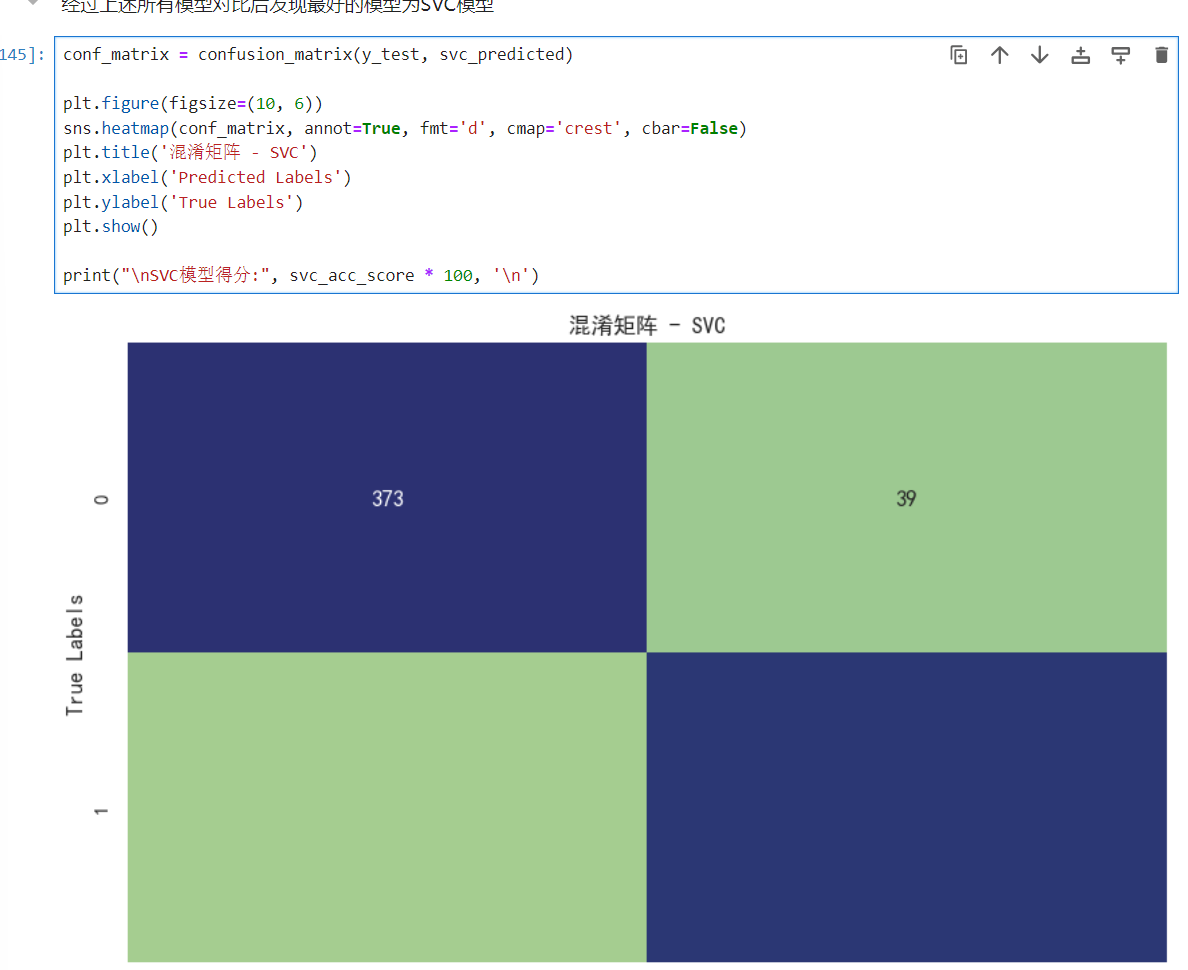


如图所示，通过更改基模型提升了模型的准确率，原因可能是神经网络，辑回归更适合作为训练二分类的基模型类型。

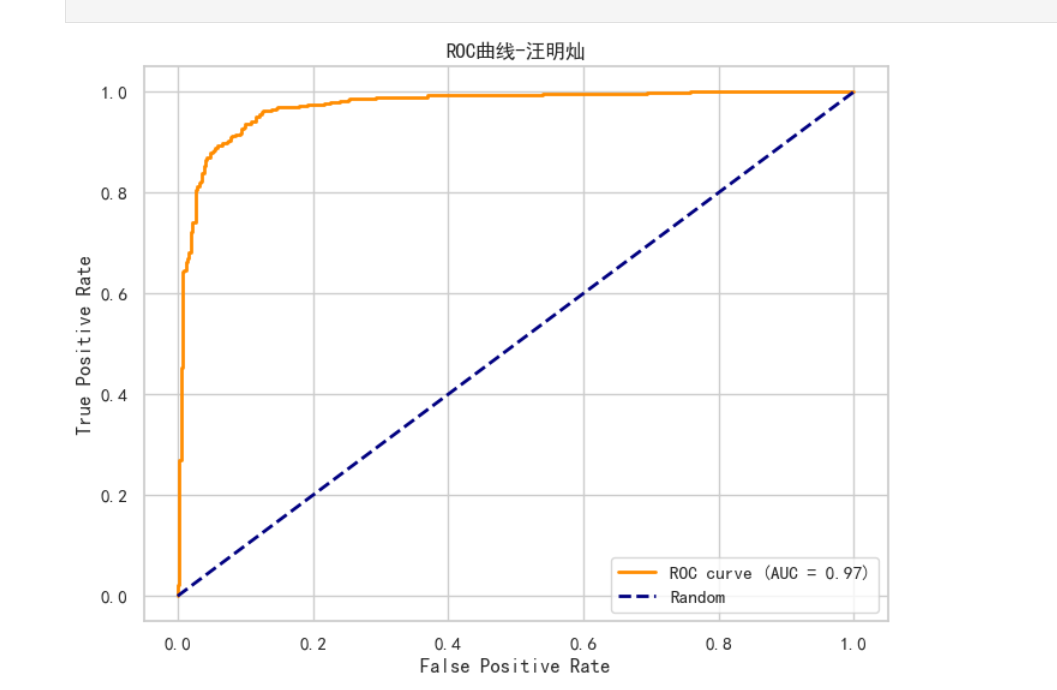
朴素贝叶斯

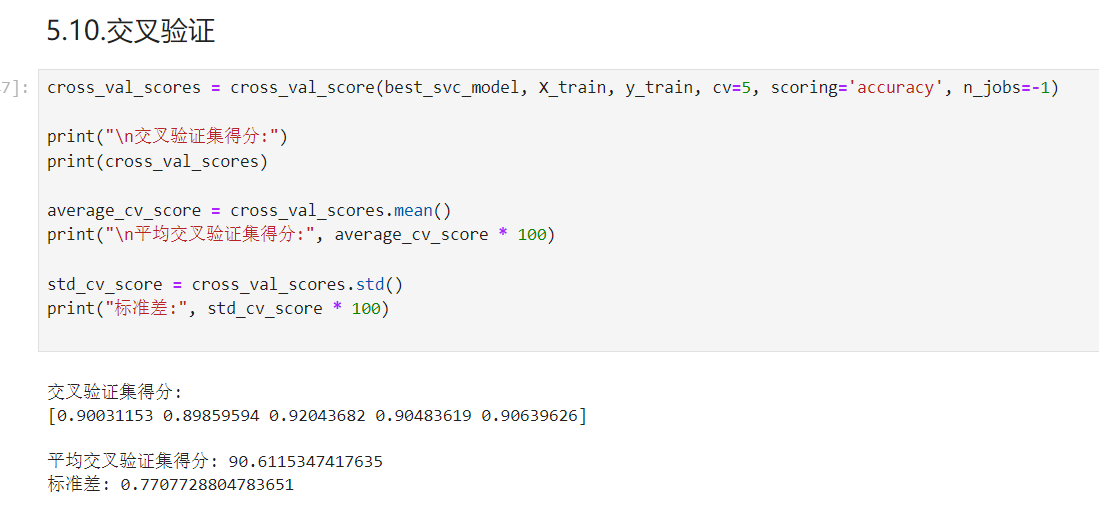


1. **实验结果分析与处理:**

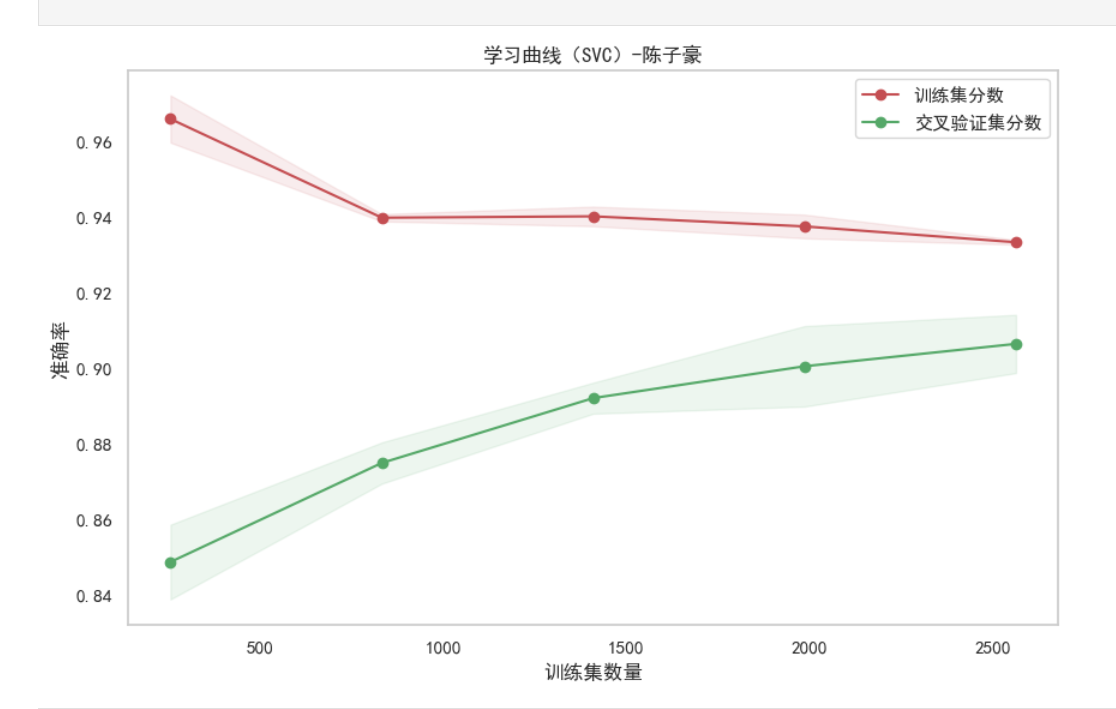
****











注意：代码输出的所有可视化结果图中必须加中文标题，并在标题结尾加上“实现该代码的学生姓名”，如“**多种算法ROC曲线对比图-张三**”。

六．其他

通过这次项目，我们对机器学习有了更加深刻的理解。通过这次项目了解到了数据处理对于项目的后续进程的重要性，数据的质量直接影响到模型训练的结果，合适的数据清洗和特征处理是成功应用机器学习算法的前提。同时数据处理时需要通过可视化对数据的特征进行更深入的了解。在数据清洗过程中，我学到了处理缺失值和异常值的重要性。在本项目中，我选择移除了包含缺失值的行，因为数据集足够大，这样处理不会导致信息的严重丢失。此外，移除不必要的列也是清洗的一部分，因为这些列对于建立模型并没有帮助，反而可能引入噪声。在实际工作中，正确的处理缺失值和无关特征对提升数据质量至关重要，可以防止模型学到错误的关联，提高模型的泛化能力。在进行数据处理时，我们尝试了多种方法并对比了它们的效果。这个过程是一个不断试错和学习的过程。例如，在标准化数据时，我们比较了MinMaxScaler和RobustScaler的效果，选择了最适合当前数据集的方法，之后了解到不同的模型有着不同的假设前提和适用场景，并不是越复杂的模型就一定会在测试集上表现更为优异，了解并选择合适的模型对于解决特定问题至关重要，对于不同场景一定要选择不同的模型进行训练否则会适得其反导致模型效果变差。最后，通过本次项目，不仅提升了自己在数据处理、模型构建和评估的技能，也加深了对机器学习理论和实践应用的理解，特别是在模型评估和选择上，我们学到了如何根据实际问题灵活调整策略和方法，比如在神经网络的搭建中，如何通过修改神经元数量以及神经网络的层数和激活函数的种类，迭代的次数来修改模型以得到表现更加优秀的模型。

七．对本门课的感想、意见和建议

希望听到你的心声。

对于本课程我们认为内容丰富，涉及层面广，同时课程难度也较大，需要认真学习并且通过不断的练习才能对机器学习有更深入的了解。

希望本课程老师能够讲的更加细致。