**一、课题综述**

* 1. **课题说明**

任务划分：

-Data Prepare：

-Data Preprocess：

-Model Construct：

-Train & Test：

-Plot Result：

-Optimize & Review：彭坤宇

* 1. **课题目标**

本课题使用的数据集来自于数据分析与数据挖掘竞赛Kaggle，该竞赛为数据科学领域著名的国际性赛事之一。课题使用的数据集为学生压力分析数据集，这个数据集包含大约20个对学生压力影响最大的特征。课题的目标为对于目标数据集，搭建相应的传统机器学习模型和深度神经网络模型，完成基于机器学习的分类任务，并能取得较优的分类性能。

* 1. **课题数据集**

本课题使用的数据集为来自Kaggle竞赛的“Student Stress Factors”([Student Stress Factors: A Comprehensive Analysis (kaggle.com)](https://www.kaggle.com/datasets/rxnach/student-stress-factors-a-comprehensive-analysis/data))，该数据集包含大约20个对学生压力影响最大的特征。这些特征是在科学上考虑了5个主要因素，它们是心理、生理、社会、环境和学术因素。

其中一些是：

心理因素 => 'anxiety\_level'、'self\_esteem'、'mental\_health\_history'、'depression'

生理因素 => 'headache'、'blood\_pressure'、'sleep\_quality'、'breathing\_problem'

环境因素 => 'noise\_level'、'living\_conditions'、'safety'、'basic\_needs'，

学术因素 => 'academic\_performance'、'study\_load'、'teacher\_student\_relationship'、'future\_career\_concerns'，

社会因素 => 'social\_support'、'peer\_pressure'、'extracurricular\_activities'、'bullying'。

此外，对学生的压力程度进行判定，分为0，1，2三个类别，并以此进行分类任务。

**二、实验报告设计**

* 1. **数据准备**

经过筛选，最后选择非图片数据集Student Stress Factors，数据集来源为kaggle。运用该数据集可以根据大学生不同的压力来源对其进行压力分级，有一定实践意义。

* 1. **数据预处理**

数据集已经经过kaggle预处理，不存在缺失值和异常值，因此可以省略剔除异常值和补充缺失值的过程。

将数据集分为自变量（特征，x）和因变量（目标，y）：借助train\_test\_split函数完成测试集和训练集的划分

对于不同模型，数据预处理不同：

逻辑回归模型：不进行进一步处理，直接将X\_train, X\_test, y\_train, y\_test作为输入

朴素贝叶斯模型，XGBoost：借助pd.get\_dummies(y\_train)将y\_train转换为one-hot编码。

* 1. **模型搭建**

模型1：逻辑回归模型

**\_\_init\_\_**：

初始化函数，用于初始化模型的输入维度 input\_dim、类别数量 num\_classes 和学习率 lr。然后调用 build 方法构建模型。

**build ：**

构建模型，创建 logistic regression 模型、损失函数（交叉熵损失函数）和优化器（随机梯度下降优化器）。其中使用了自行构建的logistic regression模型，包含init函数和forward方法。

**load\_data ：**

将输入的数据 X 和标签 y 划分为训练集和测试集，并将它们转换为 PyTorch 张量。

**train ：**

使用随机梯度下降进行优化，通过多次迭代（epoch）来更新模型参数。每个 epoch 中，计算模型的预测值 y\_pred，然后计算交叉熵损失，并进行反向传播更新模型参数。损失值被保存在列表 losses 中，并在每 100 个 epoch 打印一次。最后返回包含所有轮次损失值的列表。

**test ：**

测试模型的方法。使用测试集进行测试，计算模型在测试集上的准确率。同时，获取模型对测试集的预测标签、真实标签和预测的正类概率分数。最终返回这些结果。

模型2：xgboost模型

使用one-hot编码后y.shape[1]=3，即具有0，1，2三个标签。对于三个标签进行独立建模，这是在处理多标签分类问题时的一种典型方式。

构建模型xgb = XGB(n\_estimators=2, max\_depth=2, reg\_lambda=1, min\_child\_weight=1, objective='logistic')

各参数具体解释为

**\_estimators=2：**表示要构建的弱学习器（通常是决策树）的数量。在这里意味着会构建两棵树。

**max\_depth=2：**表示每棵决策树的最大深度。这里意味着每棵树最多有 2 层。

r**eg\_lambda=1：**L2 正则化的强度。正则化有助于控制模型的复杂性，防止过拟合。这里设置为 1，表示正则化项的权重。

**min\_child\_weight=1**：定义叶子节点中最小的样本权重和。这控制树的生长，避免分裂节点时过于细致划分。这里设置为 1。

**objective='logistic'**：表示使用逻辑回归作为目标函数。它希望最小化负对数似然损失，从而使得模型的输出更接近真实标签的概率。

模型3：朴素贝叶斯模型

与xgboost一样，对标签进行独立建模

**\_\_init\_\_：**

初始化函数，接收训练集的特征 X\_train 和对应的类别标签 y\_train。计算训练集中每个类别（0和1）的占比，即先验概率 P(类别)。

**getFrequency：**

在数据集 data 中，计算特征 feature 的值为 value 的样本占整体样本的比例。该方法用于计算条件概率 P(特征|类别) 和先验概率 P(特征)。

**predict ：**

遍历测试集中的每个样本，对于每个样本，遍历其特征。

对于每个特征，计算在类别为 0 和 1 时的条件概率 P\_feature\_label0 和 P\_feature\_label1。

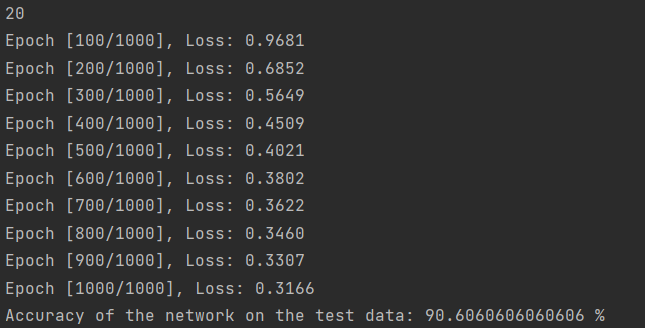
同时计算先验概率 P\_feature。

最后，根据朴素贝叶斯分类器的原理，计算在类别为 0 和 1 时的概率，并根据概率大小决定样本的预测类别。

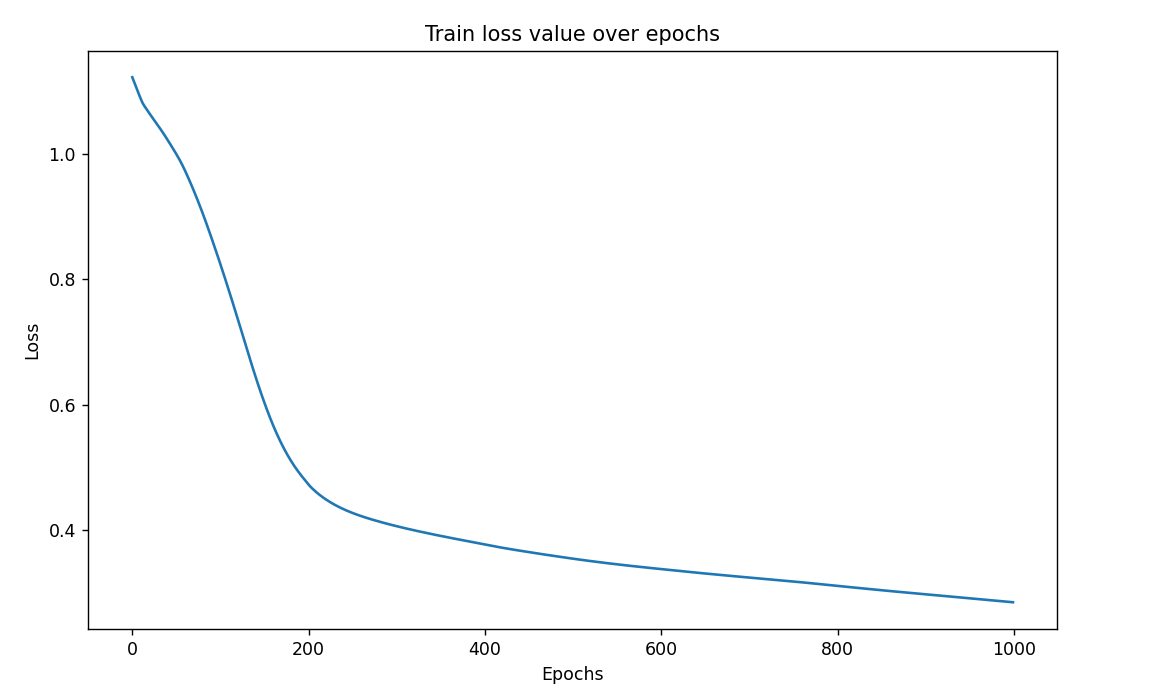
* 1. **模型训练测试**

模型一：逻辑回归模型

训练过程：

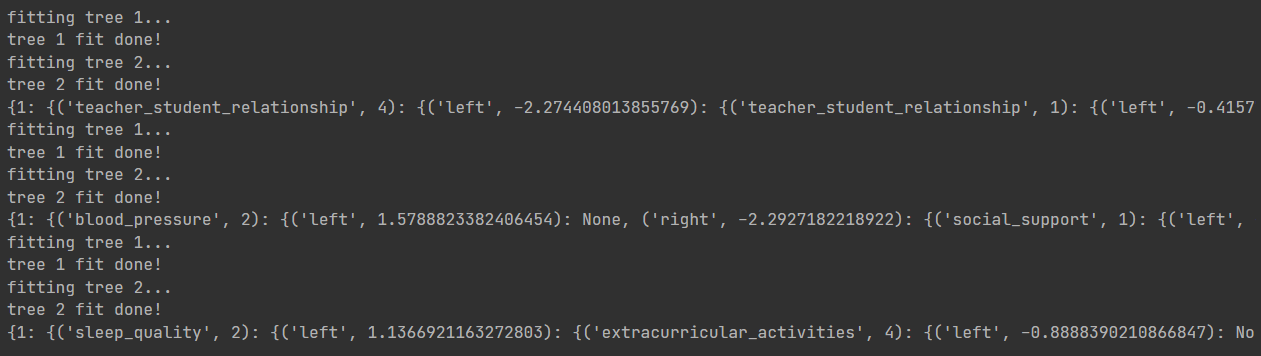


Loss曲线

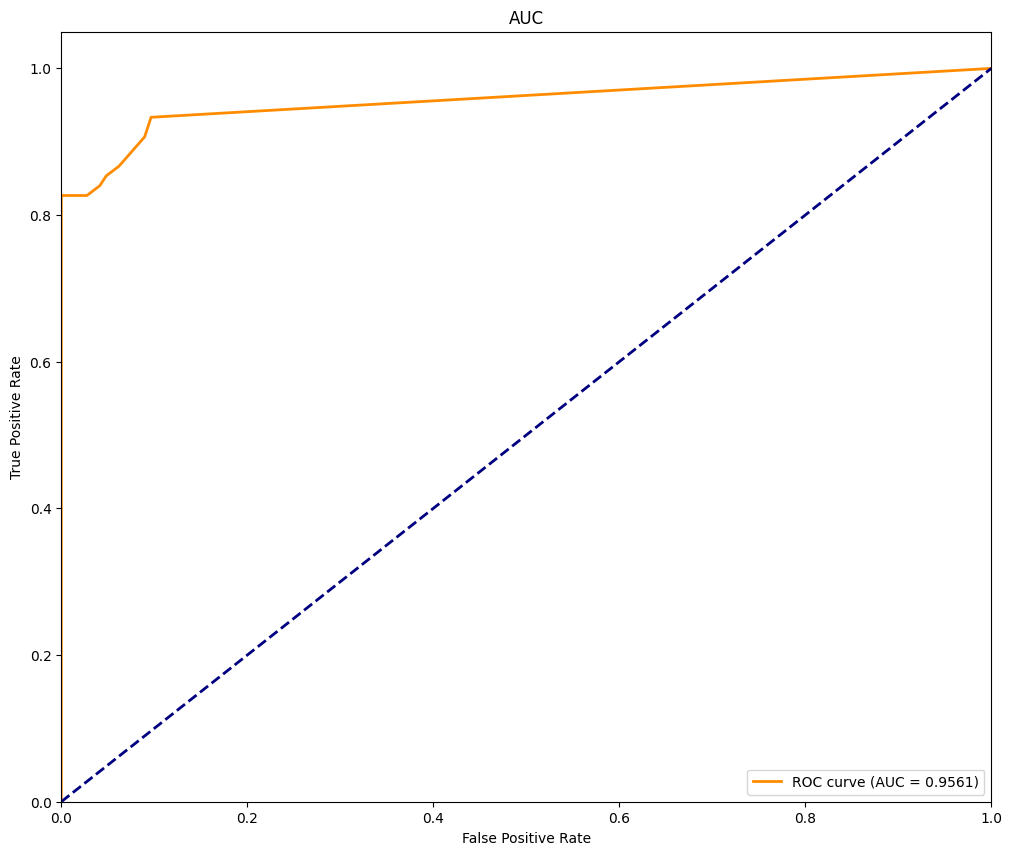
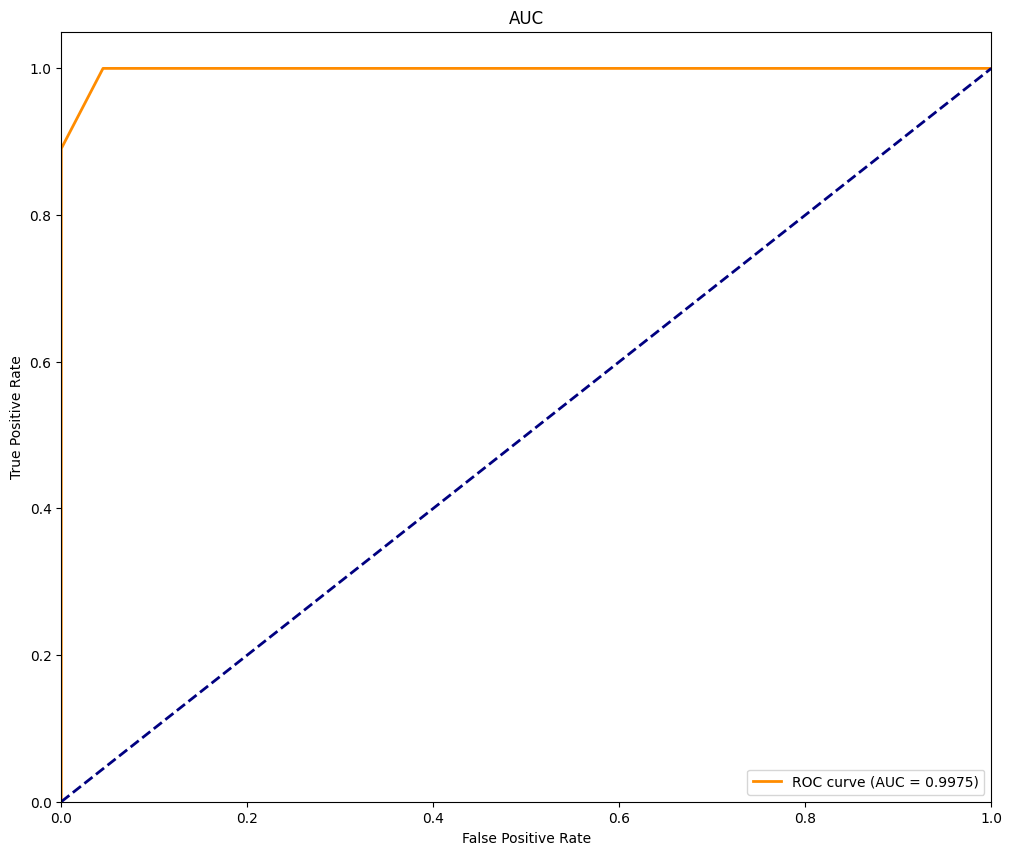
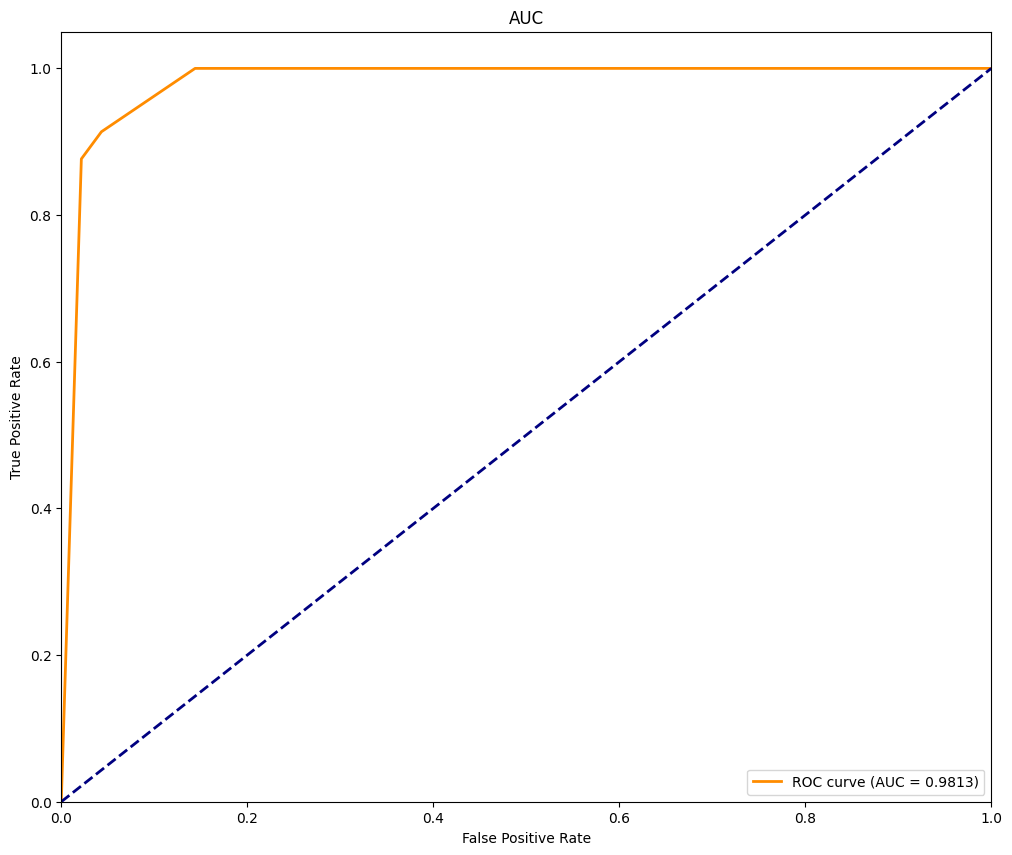


混淆矩阵

模型二：xgboost模型



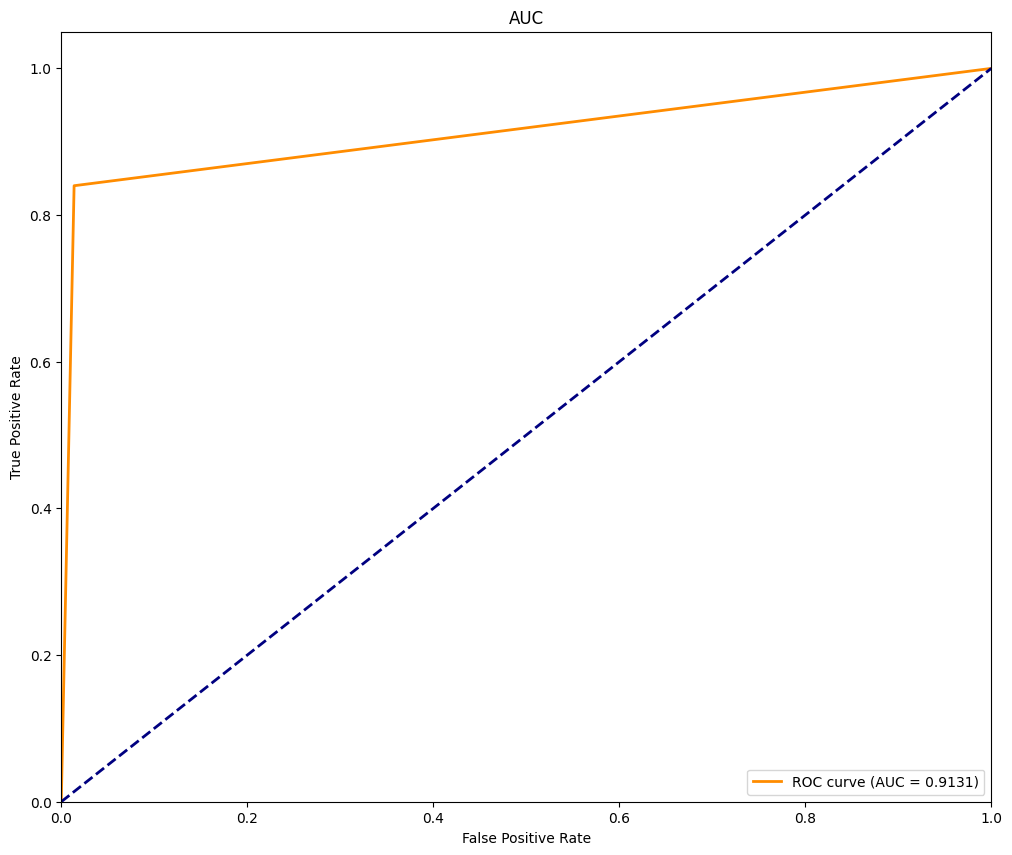
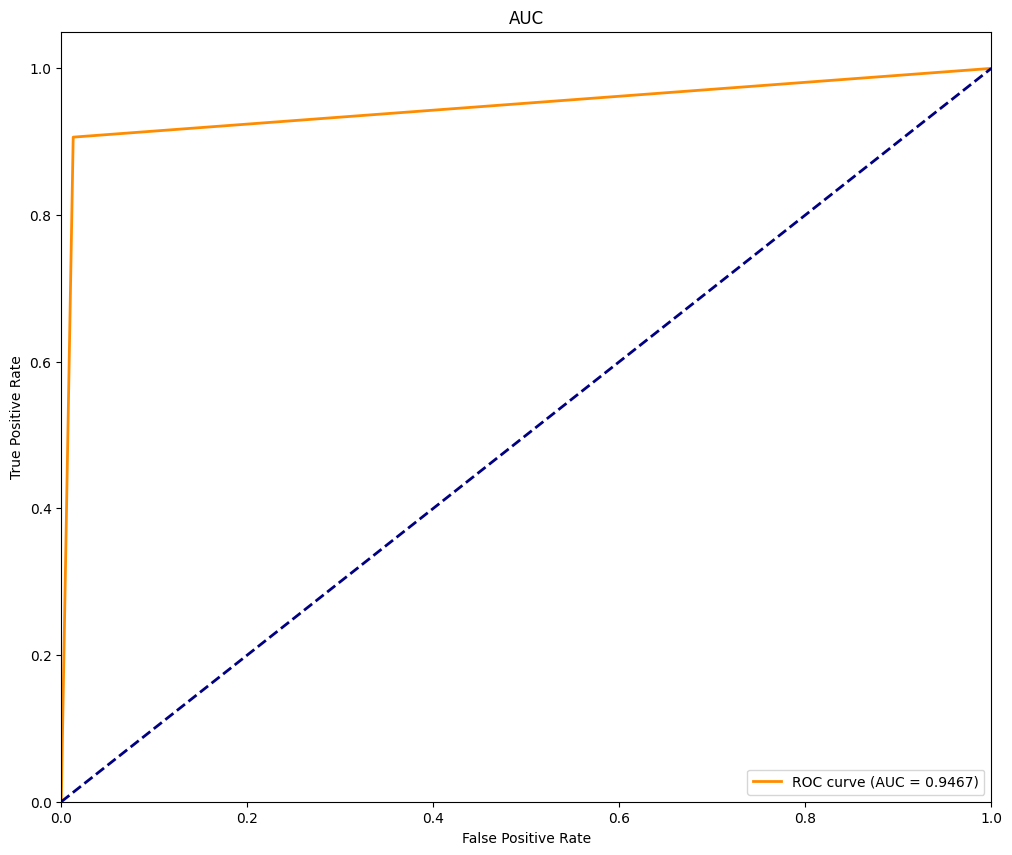
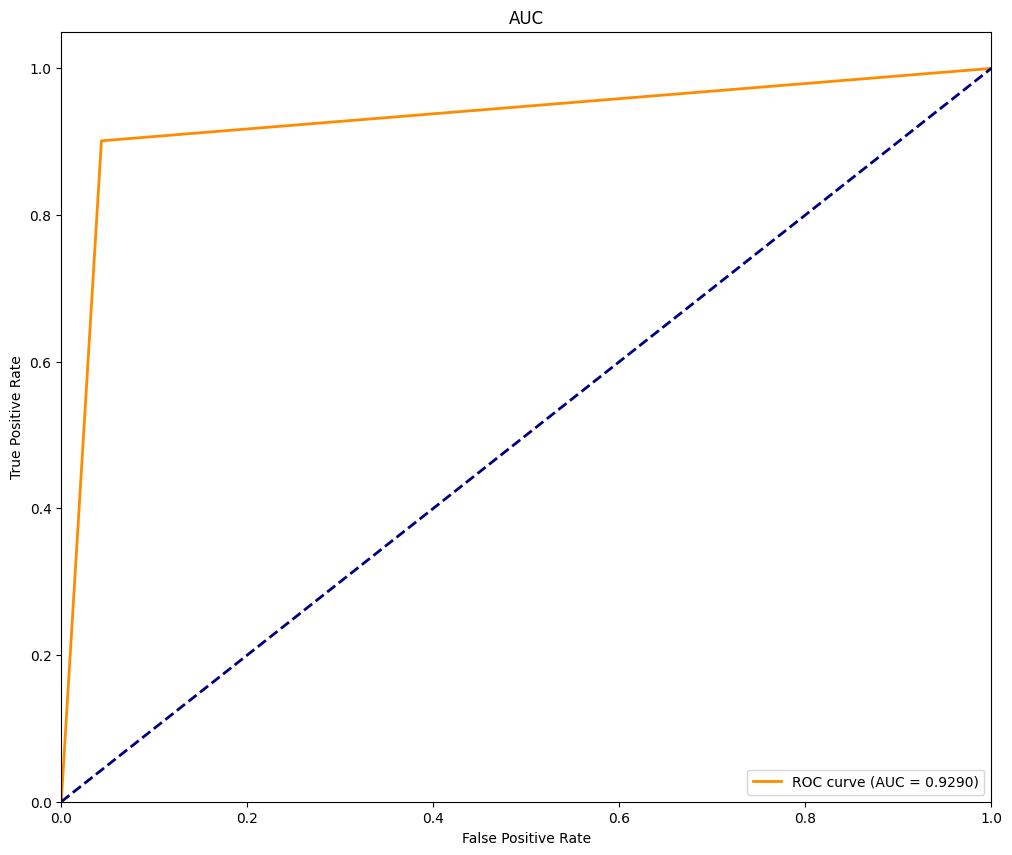
对于三个标签各自的auc曲线

各自的预测准确率为0.9561，0.9975，0.9813

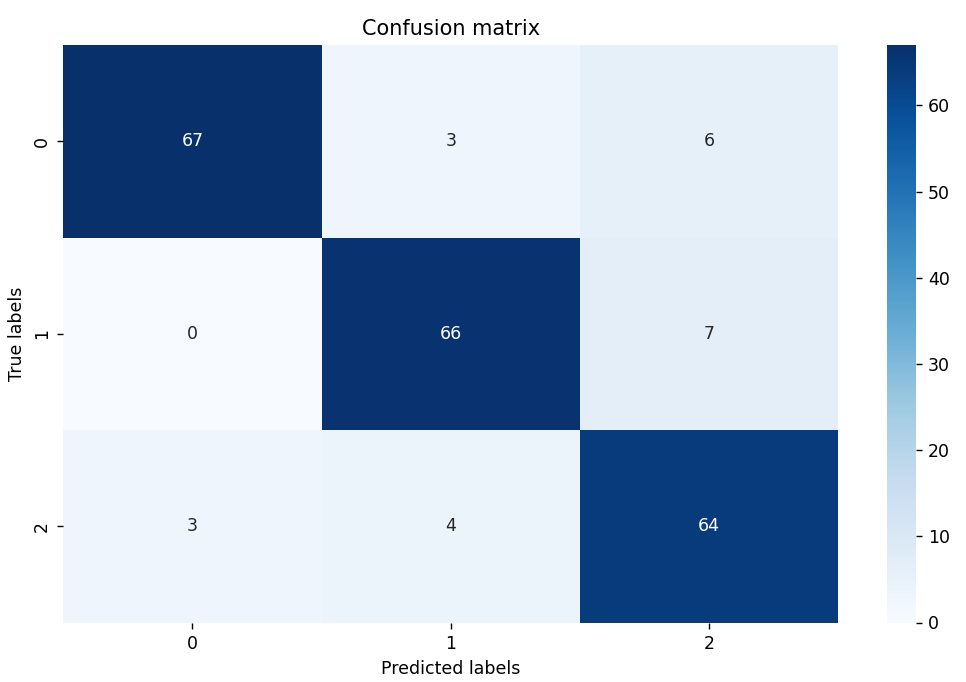
模型三：朴素贝叶斯回归

对于三个标签各自的auc曲线

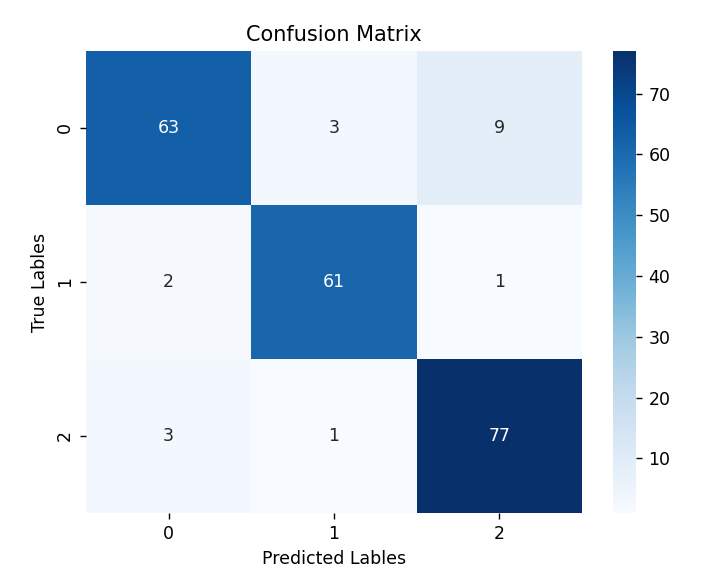
  

各自的预测准确率为0.9131，0.9476，0.9290

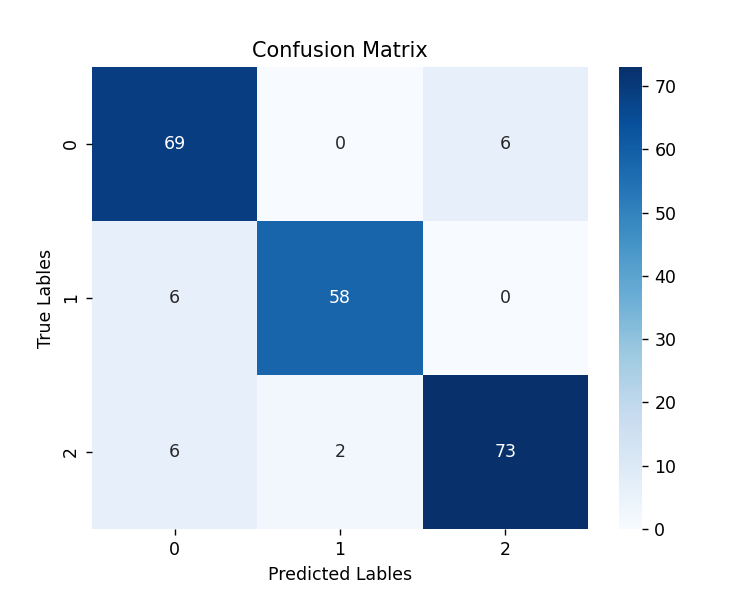
* 1. **结果可视化**



实验1混淆矩阵



实验2混淆矩阵



实验3混淆矩阵

模型1预测准确率：



模型2预测准确率：



模型3预测准确率：



* 1. **分析和优化**

**分析：**

预测准确率：

XGBOOST性能最好，朴素贝叶斯其次，逻辑回归最差，但由于测试集较小，所以可以认为预测效果相近。

计算速度：

朴素贝叶斯与逻辑回归方法较快，xgboost方法由于需要构建较为复杂的结构，所以计算复杂度较高，计算速度较慢。

参数：

xgboost参数较多，相对较难使用，而朴素贝叶斯与逻辑回归参数较少，易于使用。

输出结果：

逻辑回归可以输出样本属于某个类别的概率，帮助使用者进行辅助判断。Xgboost可以输出特征的重要性分数，帮助雷杰模型对于任务的贡献。

数据类型：

朴素贝叶斯和逻辑回归方法要数据之间具有独立性，若各特征之间具有复杂关系，则预测效果较差。

**优化：**

选择朴素贝叶斯进行优化

优化点1：

将原有的概率变为对数，由于对数化了之后在[0.1]上值的变化更大了，这样就消除了概率计算下溢的风险。log和原曲线增减性相同，且在同样的位置取到极值点，所以这样的替换是可行的。且在源代码中改动较少，对概率取对数即可。

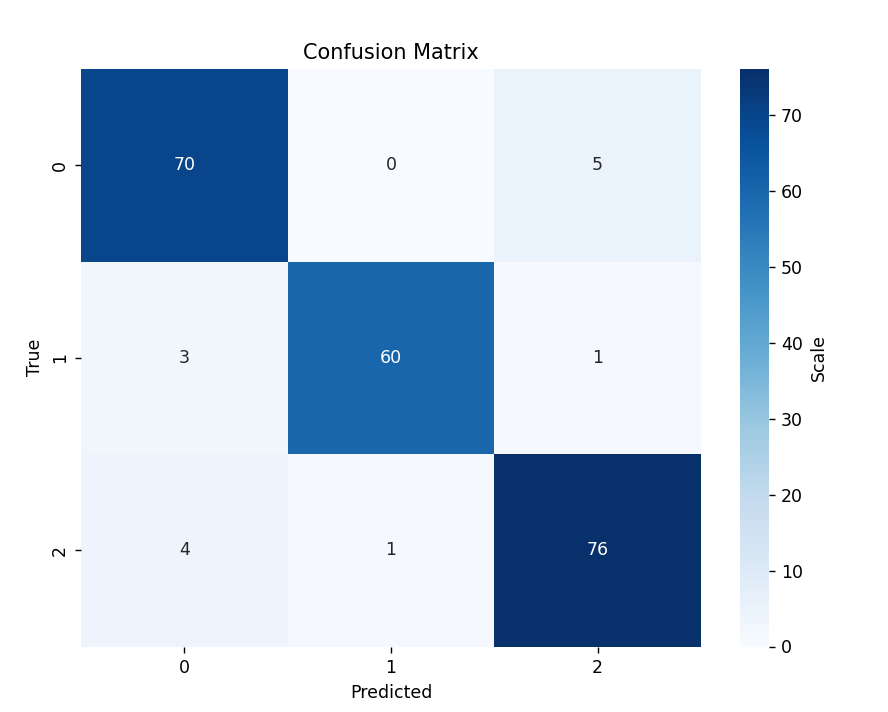
优化点2：

加入拉普拉斯平滑（Laplace smoothing）。在概率计算中，如果某个事件在训练数据中从未出现，那么其概率会被计算为零，这可能导致在分类时出现问题。

对于计算概率的部分，我在每个概率的计算中都加上了一个很小的值（1e-10），这是拉普拉斯平滑的一种实现方式。这个很小的值确保了在概率计算中不会出现零的情况，避免了取对数时出现问题。

拉普拉斯平滑的意义在于保证每个事件的概率都不会被认为是零，从而避免了零概率带来的问题。这对于朴素贝叶斯分类器的性能和稳定性都有积极的影响，特别是在处理较小的数据集或者某些特征值在某个类别下未出现的情况。

加入后预测效果：



准确率：

1. **总结**

本次实验为课程的第二次实验，对机器学习的分类任务有了一定的体会。整个实验流程大致分为Data Prepare、Data Preprocess、Model Construct、Train & Test、Plot Result、Optimize & Review六个步骤，每个步骤都至关重要不可或缺。

分类任务的关键在于通过通过大量数据学习训练样本特征，从而准确预测测试样本所属的类别。模型的合适程度可以通过评估指标来判断，包括准确率、精确度、召回率、F1 分数等。在朴素贝叶斯模型的性能优化上，我们小组也做了一定的深入思考和研究，这让我们受益匪浅。