**一、课题综述**

* 1. **课题说明**

实验分工如下：

* 1. **课题目标**

针对[肥胖或心血管疾病风险](https://www.kaggle.com/datasets/aravindpcoder/obesity-or-cvd-risk-classifyregressorcluster" \t "_blank)数据集建立多种多分类模型，以预测个体的肥胖风险，并对比分析模型预测效果差异，可用于帮助医疗机构识别高风险个体，提供早期干预和预防措施，或为个体提供个性化的健康建议和干预计划。

数据集链接：https://www.kaggle.com/competitions/playground-series-s4e2

* 1. **课题数据集**

数据集来源于Kaggle，包含来自墨西哥、秘鲁和哥伦比亚的年龄在14岁至61岁之间的人群的肥胖水平数据，这些人群具有多样化的饮食习惯和身体状况。训练集和测试集已划分，测试数据不含分类目标NObeyesdad列。训练集20758个样本，测试集13849个样本，除id列外包含16个特征列，模型在测试集的泛化能力由平台提测分数反映，部分特征说明如下：

* Gender -个体性别
* family\_history\_with\_overweight - 是否有家庭成员曾患或正在患有超重
* FAVC - 高热量食物的频繁摄入
* FCVC - 蔬菜的摄入频率
* NCP - 主餐次数
* CAEC - 餐间食物的摄入
* SMOKE - 个体是否吸烟
* CH2O - 每天的水摄入量
* SCC - 个体是否监控热量摄入
* FAF - 进行体力活动的频率
* TUE - 使用科技设备的时间
* CALC - 酒精的摄入
* MTRANS - 使用的交通方式
* NObeyesdad - 肥胖水平（分类目标列）

**二、实验报告设计**

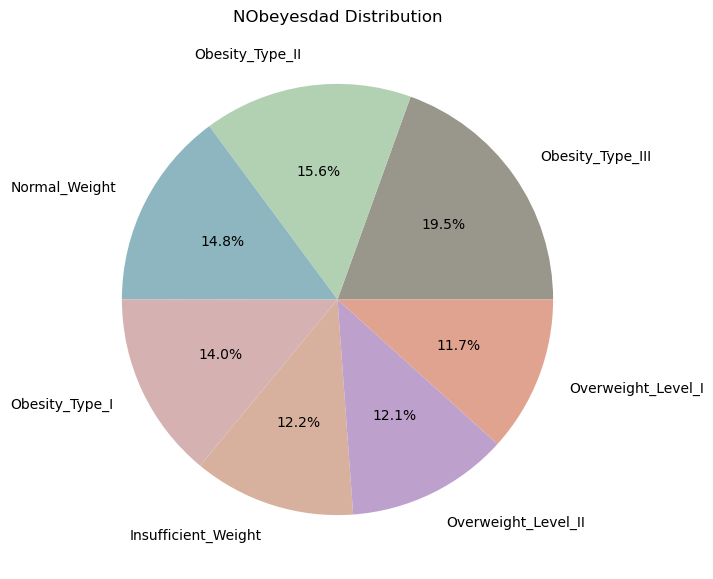
* 1. **数据准备**

**2.1.1 数据集获取与数据加载**

数据集于2024年3月在[Kaggle](https://www.kaggle.com/datasets/taweilo/taiwan-air-quality-data-20162024" \t "_new)发布，使用pandas库将其以CSV形式加载到Python3环境

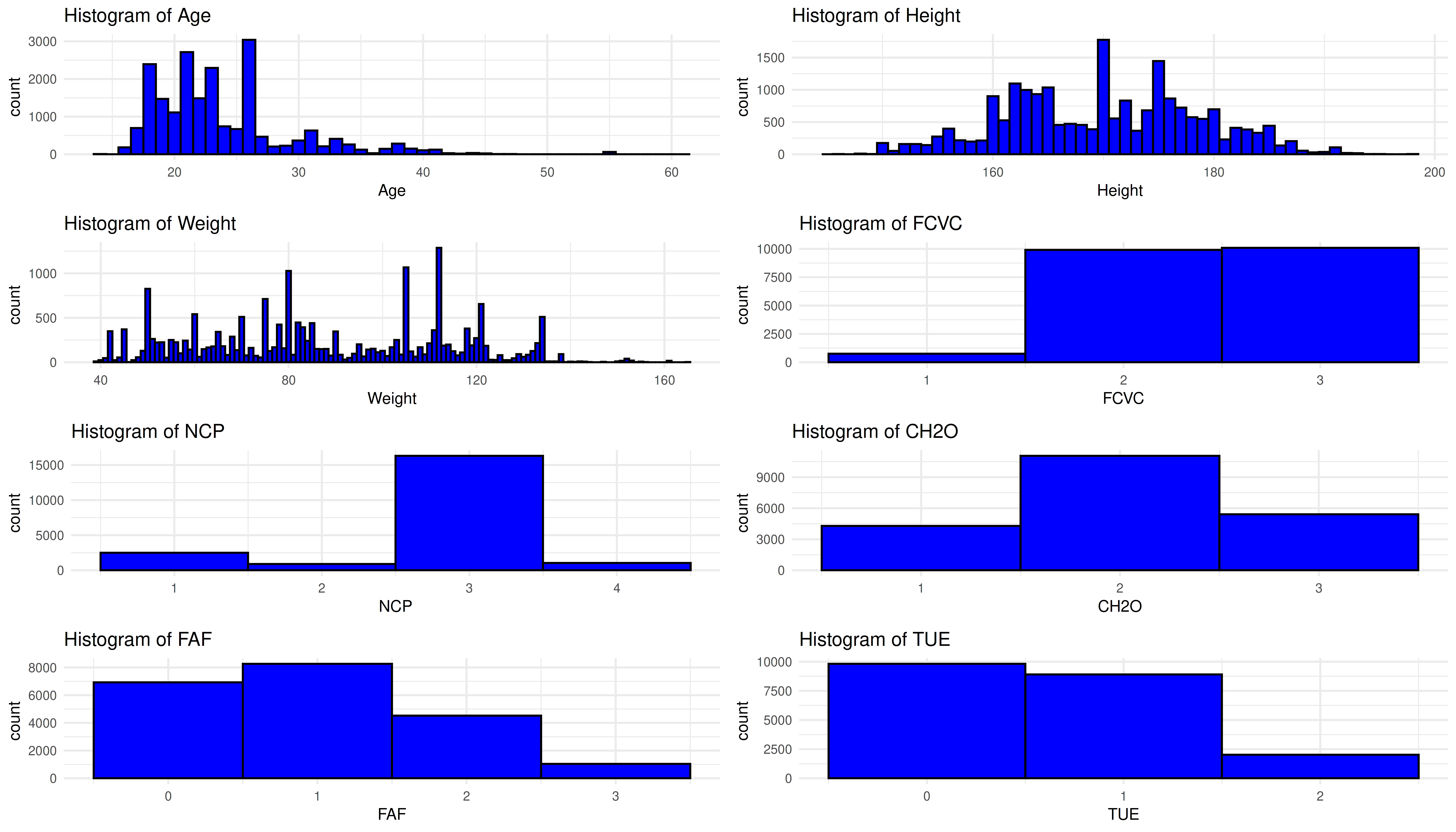
**2.1.2 数据探索性可视化**

* 最终分类标签及其分布（共7个）



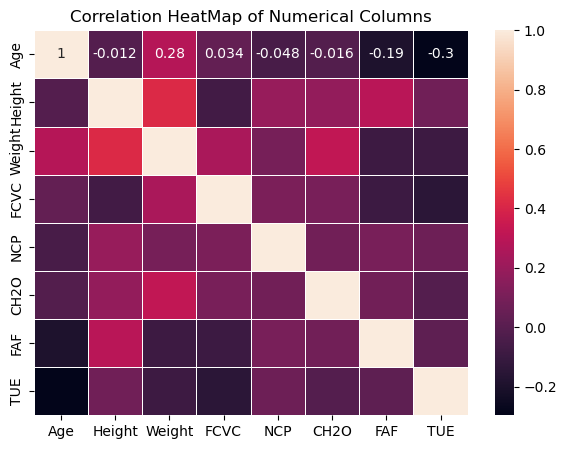
各类数量基本均匀，无明显少数类

* 特征取值分布：



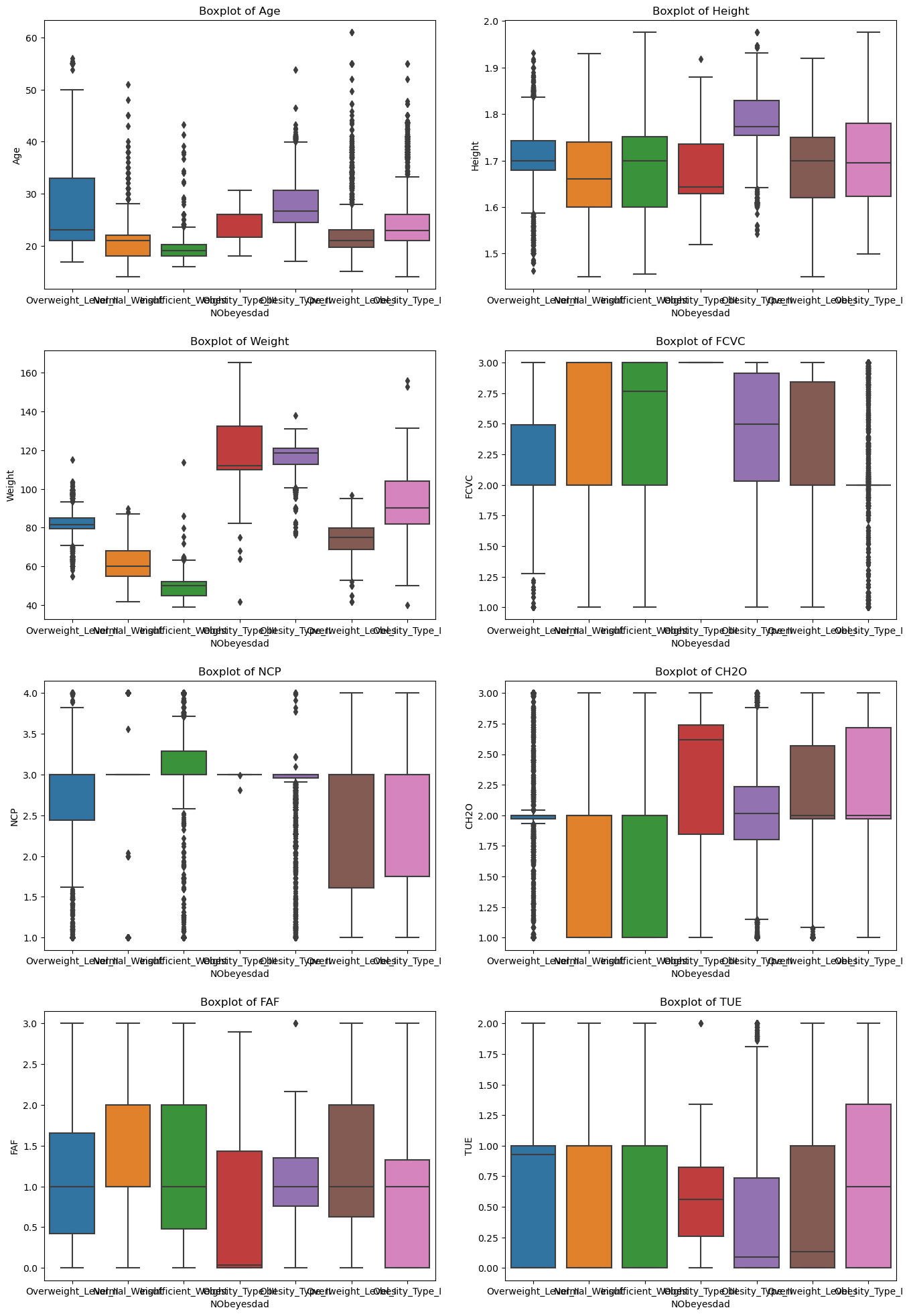
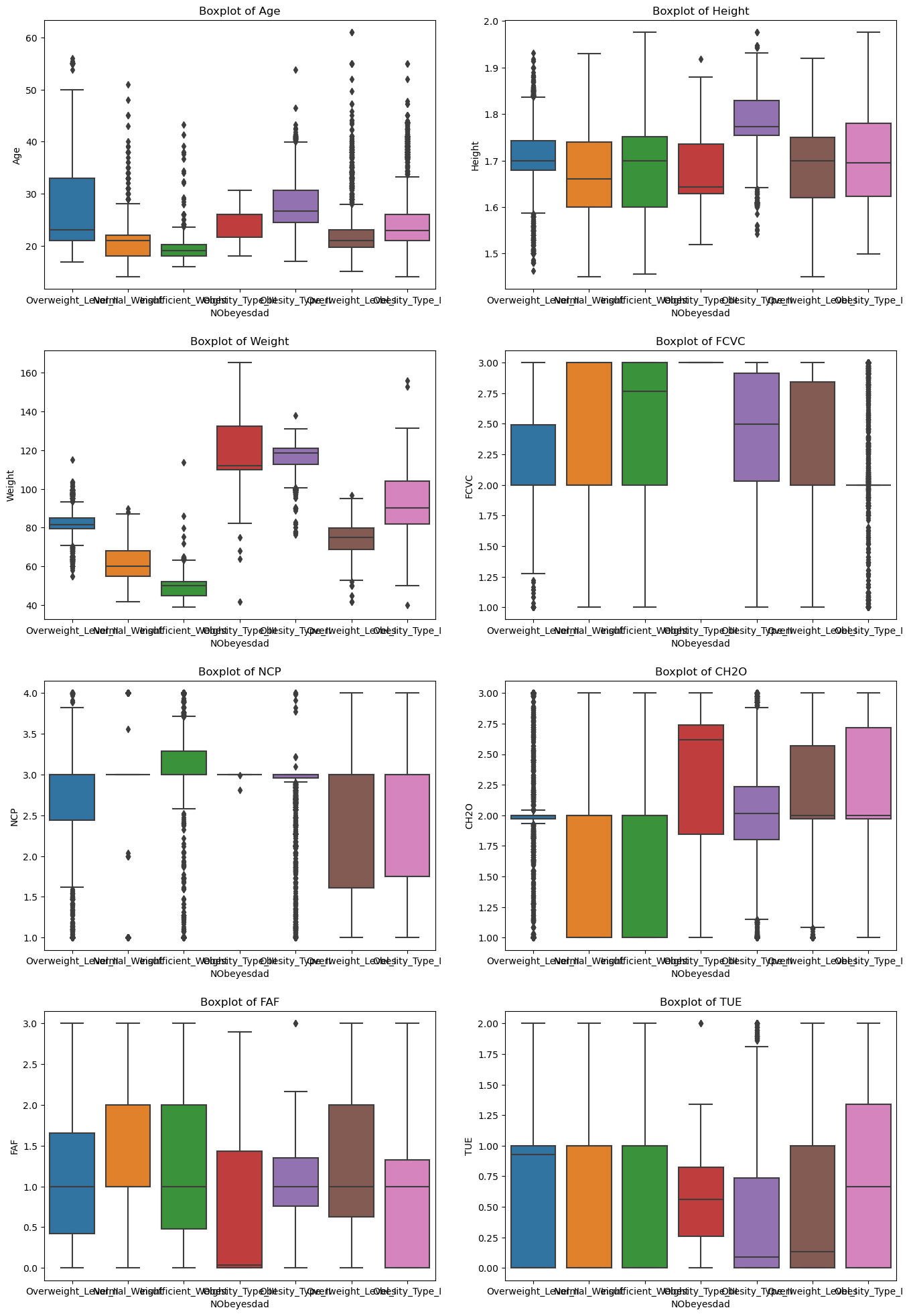
* 特征相关性分析

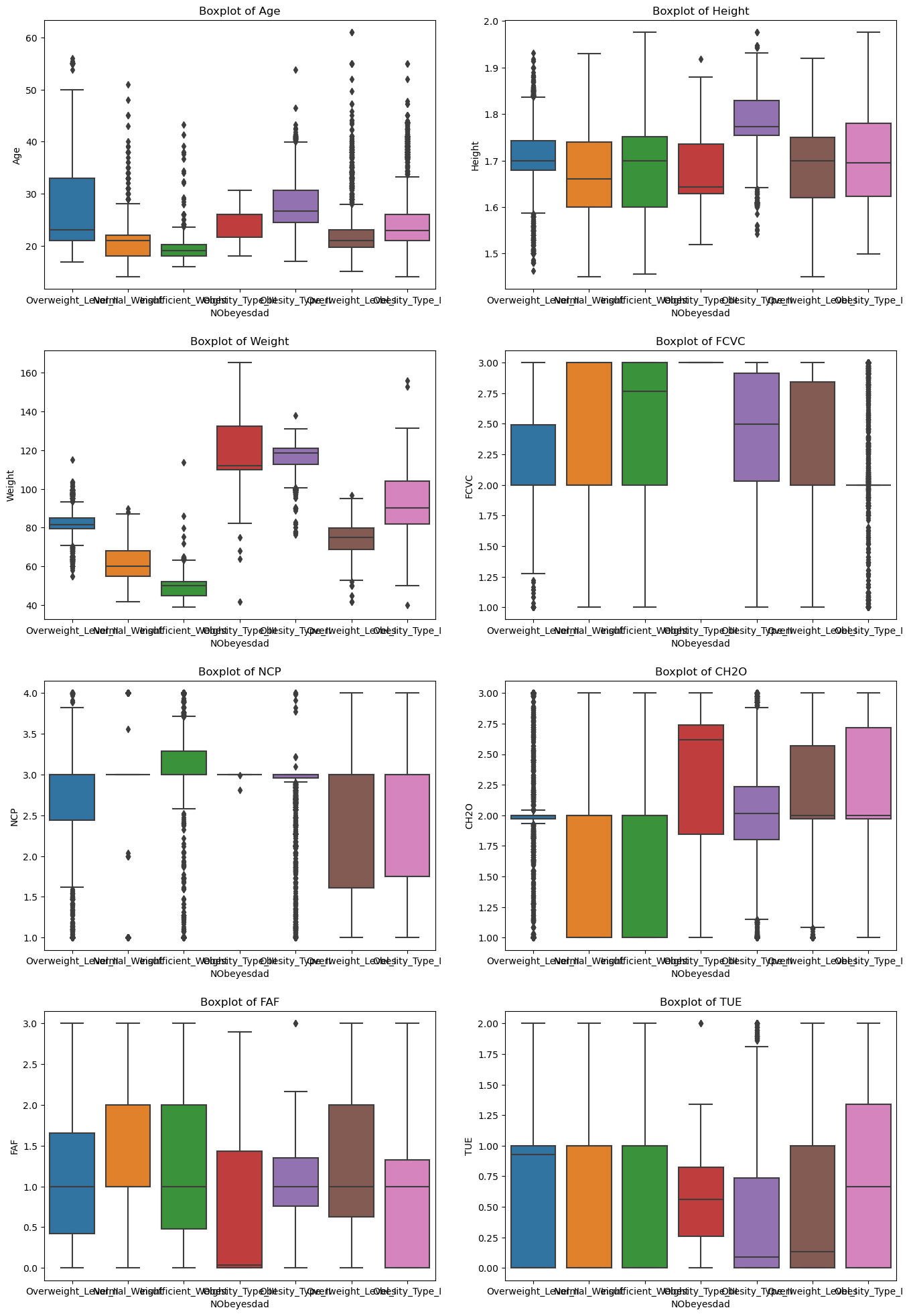
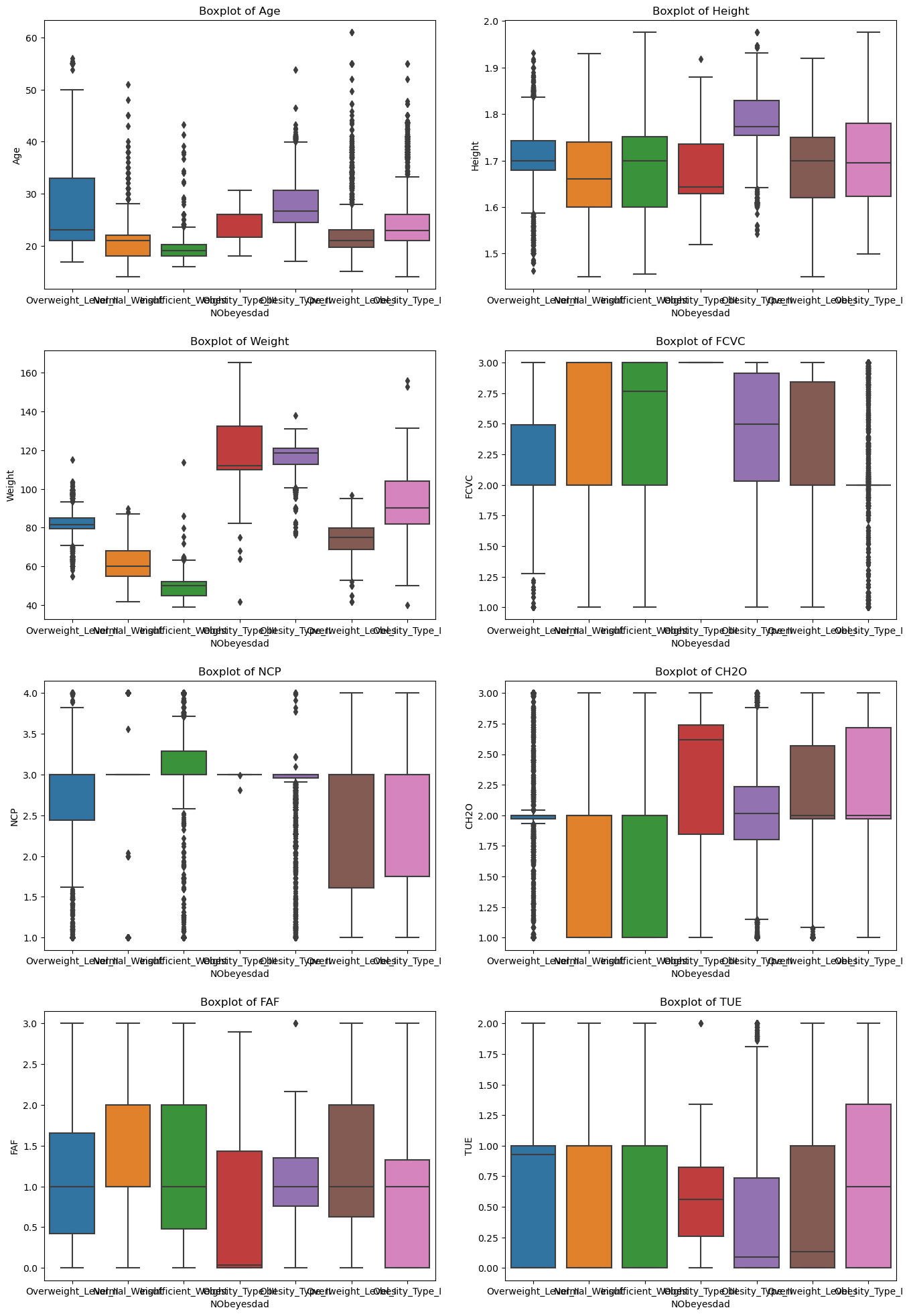


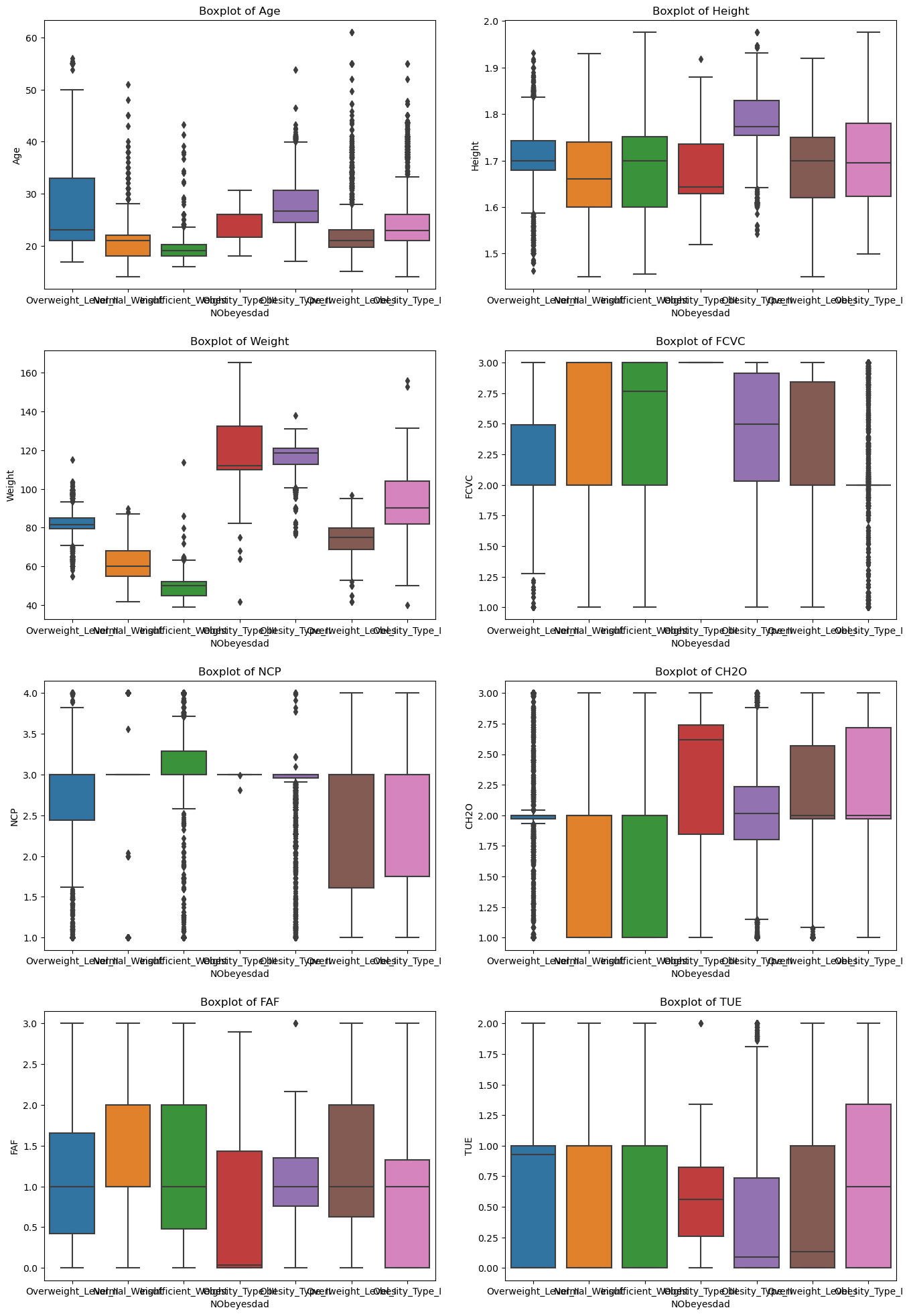


Age 和 Weigh、Height 和 Weight 之间有较强正相关关系，其他特征之间的相关性相对较弱，没有显著相关性。

* 不同类别下特征分布箱线图：







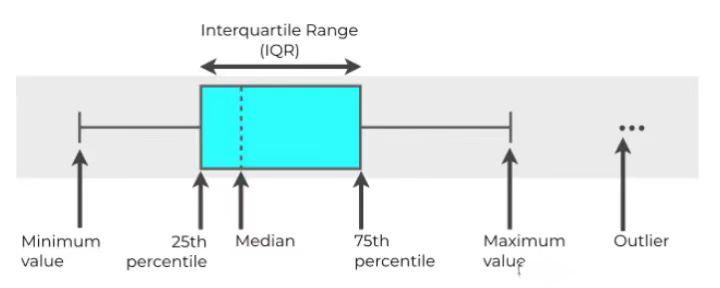
* 1. **数据预处理**

**2.2.1 初步探索**

数据集无缺失值、重复值，无需处理缺失值，无需删除重复值；分离id列

**2.2.2 异常值处理**

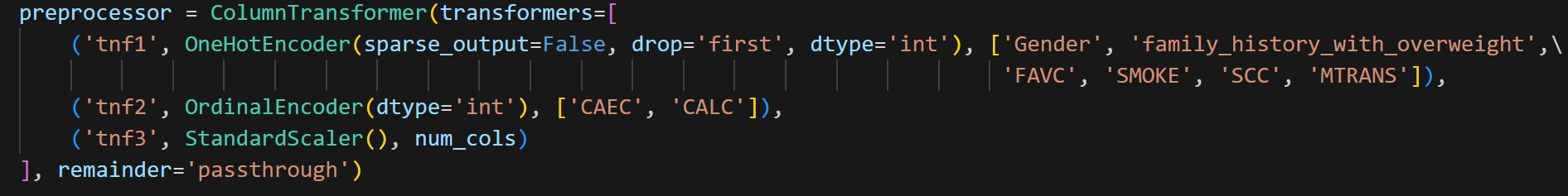
数据中存在一些极端值或异常值，会误导模型训练和预测。使用IQR方法剔除异常样本，提升数据可靠性。



**2.2.3 将目标分类标签转为数值**

考虑到算法兼容性等因素，将预测类别NObeyesdad转为数值型

**2.2.4 特征编码**



对无序的分类变量，使用独热编码转换为二进制向量，以便模型处理

对有序的分类变量，使用有序编码保留类别间的顺序关系。如CAEC和 CALC按顺序编码为 0, 1, 2, ...

**2.2.5 特征标准化**

为避免数值特征的量纲差异导致模型产生偏差，标准化数值型特征，转换成具有零均值和单位方差的标准正态分布形式，使模型更好地收敛

**2.2.6 数据集划分**

训练集和测试集约按3：2划分，分别用于模型训练和验证泛化能力。

随机抽取训练集中20%的数据作为验证集。

通过以上预处理将原始数据转化为一个清晰、规范且易于模型处理的数据集。过程中，清理异常值确保数据质量，标准化保证数据一致性，为后续模型训练和预测奠定了坚实基础，使结果具有更高可信度和可解释性。

* 1. **模型搭建**

**2.3.1 模型评估指标**

* **准确率：**

准确率是正确预测的样本数占总样本数的比例，本数据集各类别样本数量较均衡，可使用准确率评估模型性能

* **F1 Score：**

F1 Score可以平衡精确率和召回率，可以作为较好的评价指标

* **ROC曲线和AUC：**

本实验是多分类问题，采用**一对多**方式计算并绘制ROC曲线:对于每个类别k构建一个二分类问题，将该类别视为正类，其余所有类别视为负类,为每个类别分别绘制ROC曲线和计算AUC。

由于数据集不存在类别不平衡问题，采用**微平均**方法汇总AUC值：将所有类别或类别对的真阳性率（TPR）和假阳性率（FPR）合并在一起，然后计算总的ROC曲线和AUC。

* **混淆矩阵：**

显示模型预测结果与实际结果的对比情况的二维矩阵

**2.3.2 KNN模型**

KNN在多分类问题中通过找到最近的K个邻居，通过投票机制确定测试样本的类别，实现相对简单。

**K值选择：**训练过程中，模型K值选为5。

**距离度量：**模型允许在欧氏距离和曼哈顿距离之间进行选择。通过在构造函数中添加 metric 参数指定要使用的距离度量，并在 \_predict 方法中依据选择进行相应计算。

**数据标准化：**前述预处理已完成，对 KNN 模型尤其重要，因为距离度量对不同量纲的特征非常敏感。

**加权投票机制：**为改进传统 KNN 中简单多数投票的局限性，模型引入了加权投票机制，根据距离赋予每个邻居不同的权重。距离越近的邻居对结果的影响越大，从而提升预测的准确性。为了计算权重，对每个邻居的距离取倒数，使得距离越小的邻居权重越大，最终根据每个类别的权重和确定预测类别。

**2.3.3 SVM支持向量机**

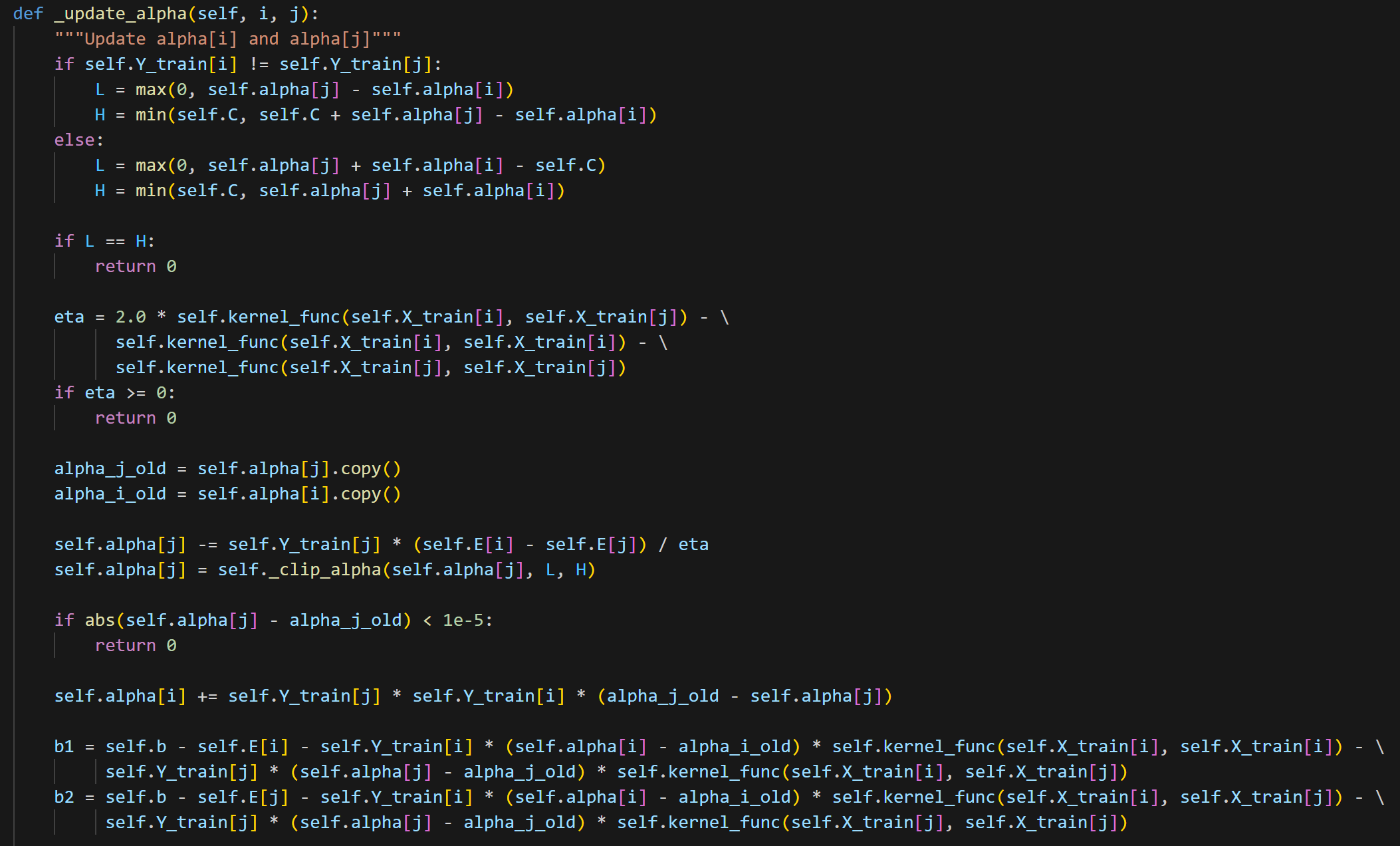
支持向量机（SVM）是一种监督学习算法。分类问题中，SVM 的主要目标是在特征空间中找到一个最优超平面，使不同类别的样本被这个超平面以最大间隔分开。本次实验手动实现了SVM的底层代码，由于数据集不是线性可分，实现朴素的SVM后引入软间隔，并引入核函数，**手动实现了线性核函数、多项式核函数、径向基核函数（RBF）的SVM算法**，同时由于RBF作为核函数是模型计算复杂度过高导致训练时间过长，使用了**手动实现的SMO算法**进行优化。

线性核：

多项式核：

RBF函数核：

**SMO算法：**序列最小优化（SMO）算法将复杂的二次规划问题分解成一系列简单的子问题，从而显著提高SVM训练速度，核心在于每次选择两个拉格朗日乘子进行优化，而不是一次性优化所有乘子。部分代码实现如下：



**2.3.4 随机森林**

随机森林是由多个决策树组成的集合模型，基本结构搭建包含以下几步：

* 数据抽样：使用Bagging方法数据采样
* 决策树的构建与训练：对每个训练集训练一棵决策树。在树中每个节点分裂时，只从一部分特征中选取最佳特征进行分裂，从而增加模型的差异性，减少了各树之间的关联。
* 集成预测：对每一个新的输入样本，随机森林的每颗回归树都会给出一个预测值Ti(x)，最终的预测结果是所有树预测值的平均值：

**2.3.5 DNN模型**

深度神经网络（DNN）是一种多层神经网络，广泛用于分类任务。DNN通过多层非线性变换学习输入数据的复杂特征表示，从而实现高效分类。

* 网络结构：

**输入层：**一个具有64个神经元的全连接层，使用Sigmoid激活函数，输入特征数量由数据集X的特征数量决定。

**隐藏层：**共9个全连接层，神经元数量从64递增至512再递减至16，激活函数主要使用ReLU，最后一个隐藏层使用ReLU激活函数。

**Dropout层：**在每个全连接层后都跟有一个Dropout层，丢弃率从0.05递增至0.15再递减至0.05，用于减少过拟合。

**输出层：**一个全连接层，神经元数量等于目标变量NObeyesdad的唯一值数量，使用Softmax激活函数，用于多分类问题。

**优化器和损失函数：**使用Adam优化器进行模型训练，损失函数采用稀疏分类交叉熵：

**2.3.6 LightGBM**

LightGBM是一个实现GBDT梯度提升树算法的框架，选择了基于histogram的决策树算法，每次选取最大增益的节点进行分裂，训练速度较块、内存消耗较低、准确率较高。

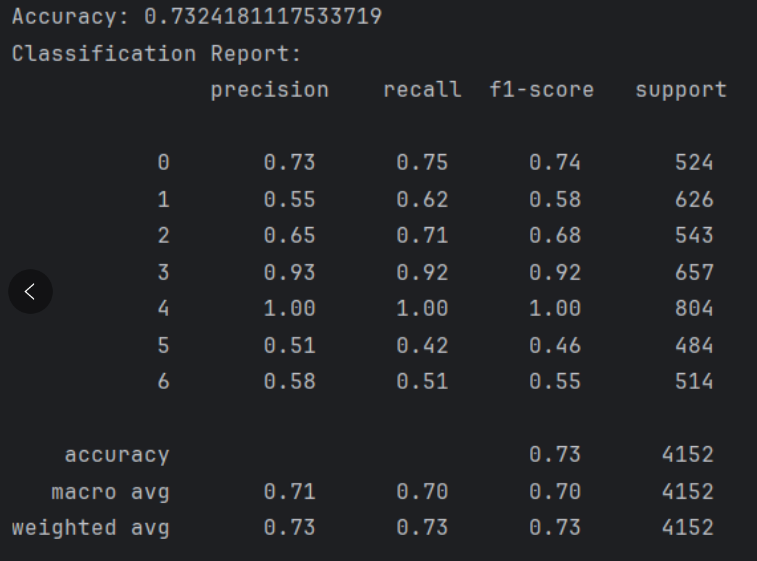
* 1. **模型训练测试**

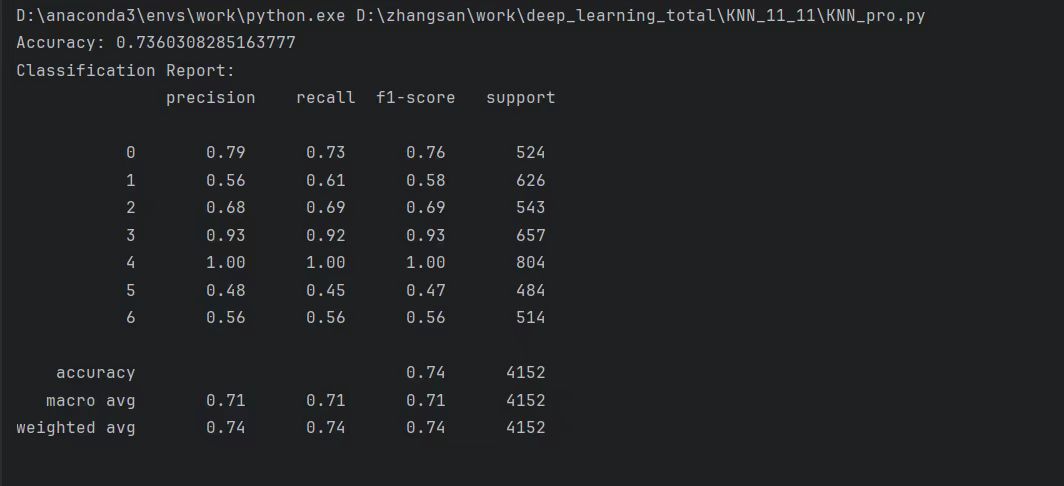
**2.4.1 KNN**

|  |  |  |  |
| --- | --- | --- | --- |
| 验证集准确率 | 验证集F1分数 | 验证集ROC AUC | 测试集准确率 |
| 0.74 | 0.74 | 0.88 | 0.7360 |

训练过程中各项指标数据：

调库实现（左）、手写实现（右）：

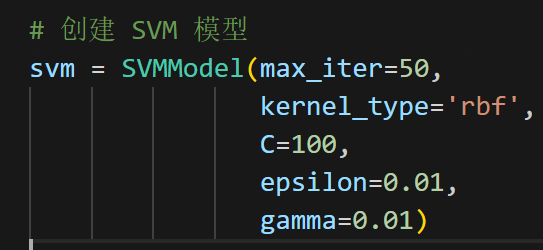


****

**2.4.2 SVM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 核函数选择 | 验证集准确率 | 验证集F1分数 | 验证集ROC AUC | 测试集准确率 |
| 线性核 | 0.75 | 0.75 | 0.75 | 0.7487 |
| 多项式核 | 0.83 | 0.83 | 0.83 | 0.8331 |
| RBF核 | 0.88 | 0.88 | 0.88 | 0.8820 |

其中模型参数设置如下（以rbf为例）：

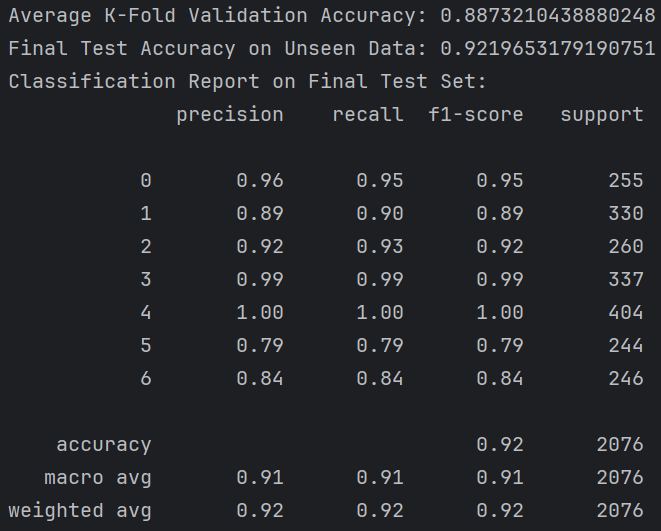


其中：

* max\_iter：算法在停止之前允许的最大迭代次数，如果在达到最大迭代次数前没有收敛，会提前终止
* kernel\_type：核函数类别
* C：正则化参数，控制对误分类样本的惩罚程度，较大意味着对误分类的容忍度较低，可能导致过拟合；较小则模型会更加平滑，可能导致欠拟合
* Epsilon：收敛阈值，当两次迭代之间的目标函数变化小于epsilon时认为已经收敛，可以停止迭代
* Gamma: RBF核函数的参数，控制高斯核宽度，较大意味着较窄，模型更关注局部信息，可能过拟合；较小意味着较宽，模型更平滑，可能欠拟合

**2.4.3 随机森林**

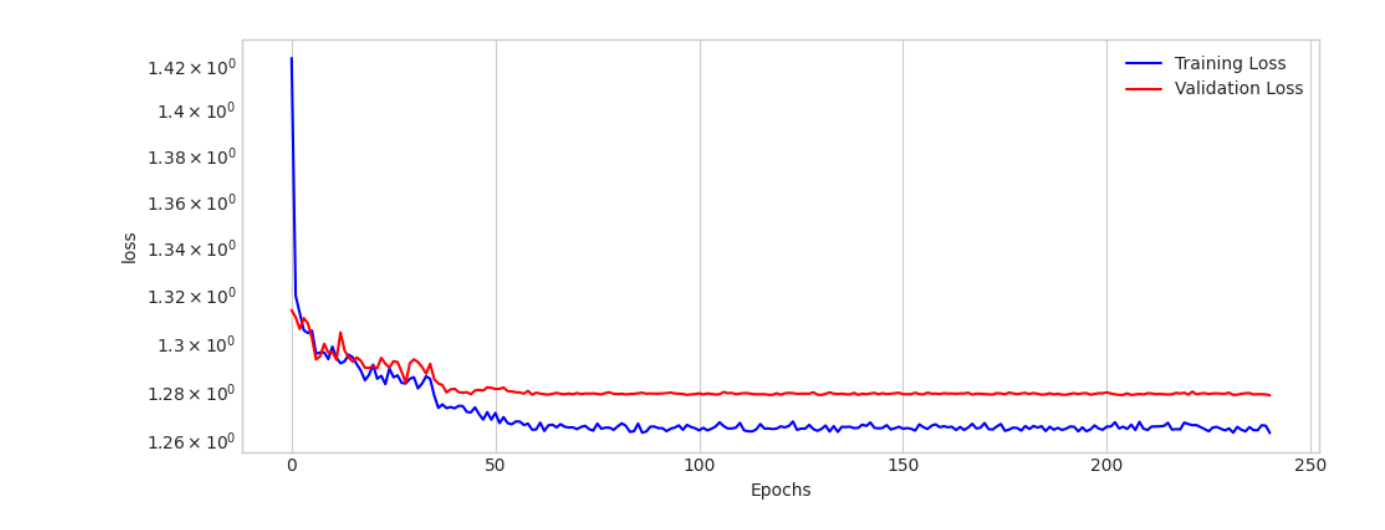
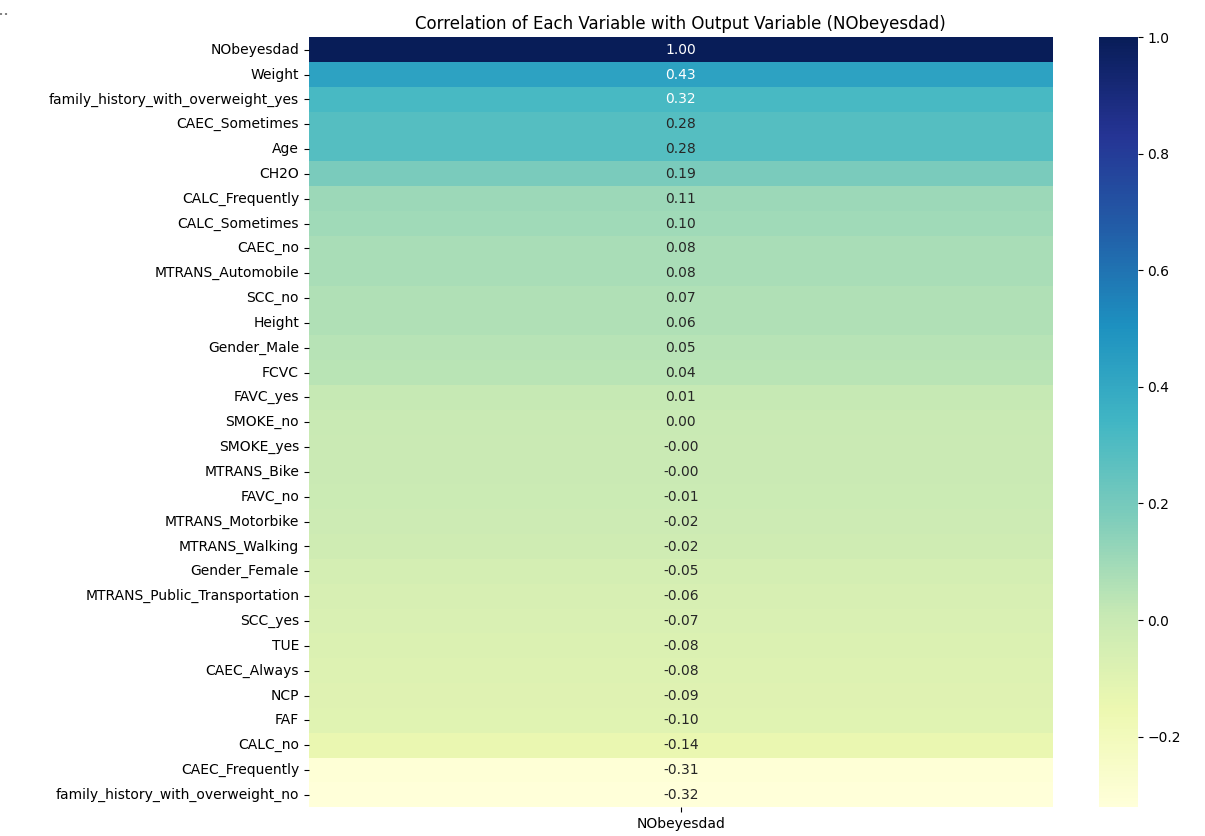
训练过程中使用5折交叉验证，经过多轮不同数据清洗方法选择及模型调整，最终各项指标如下（运行时间基于不同电脑性能可能存在差异）：



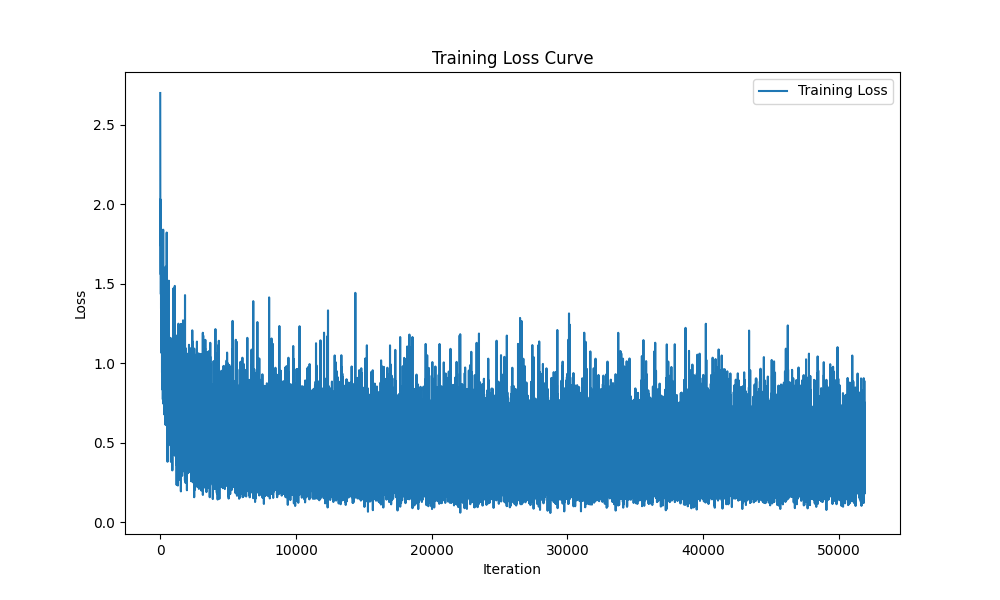
**2.4.4 DNN深度神经网络**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实现方式 | 验证集准确率 | 验证集F1分数 | 验证集ROC AUC | 测试集准确率 |
| 调库实现 | 0.9059 | 0.8969 | 0.9408 | 0.9066 |
| 手动实现 | 0.88 | 0.88 | 0.88 | 0.8786 |

**调库实现训练过程可视化：**



**手动实现训练过程可视化：**

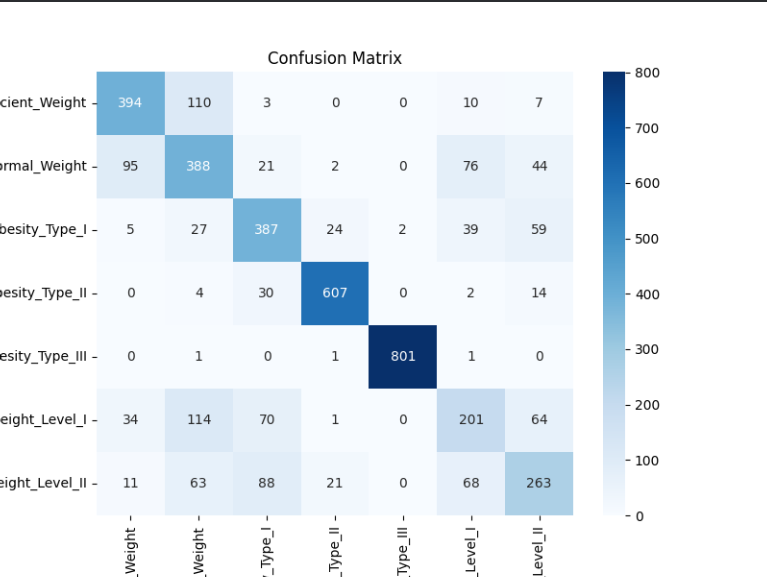


**2.4.5 LightGBM**

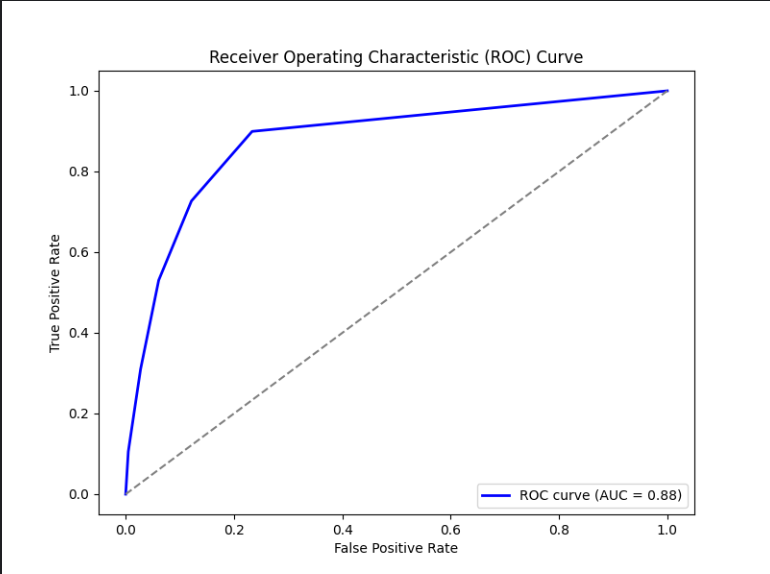
|  |  |  |  |
| --- | --- | --- | --- |
| 验证集准确率 | 验证集F1 | 验证集ROC AUC | 测试集准确率 |
| 0.91 | 0.91 | 0.91 | 0.99 |

* 1. **结果可视化（优化后的最终结果）**
* **KNN:**

混淆矩阵：



ROC曲线：

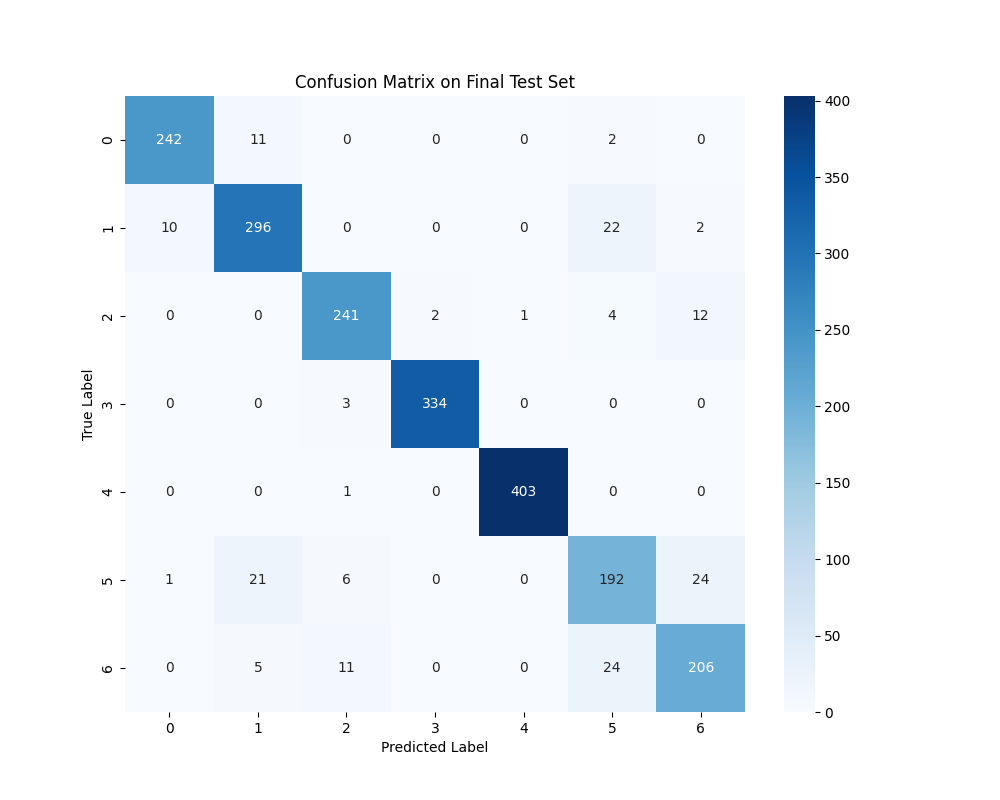


平台测试集提交分数：

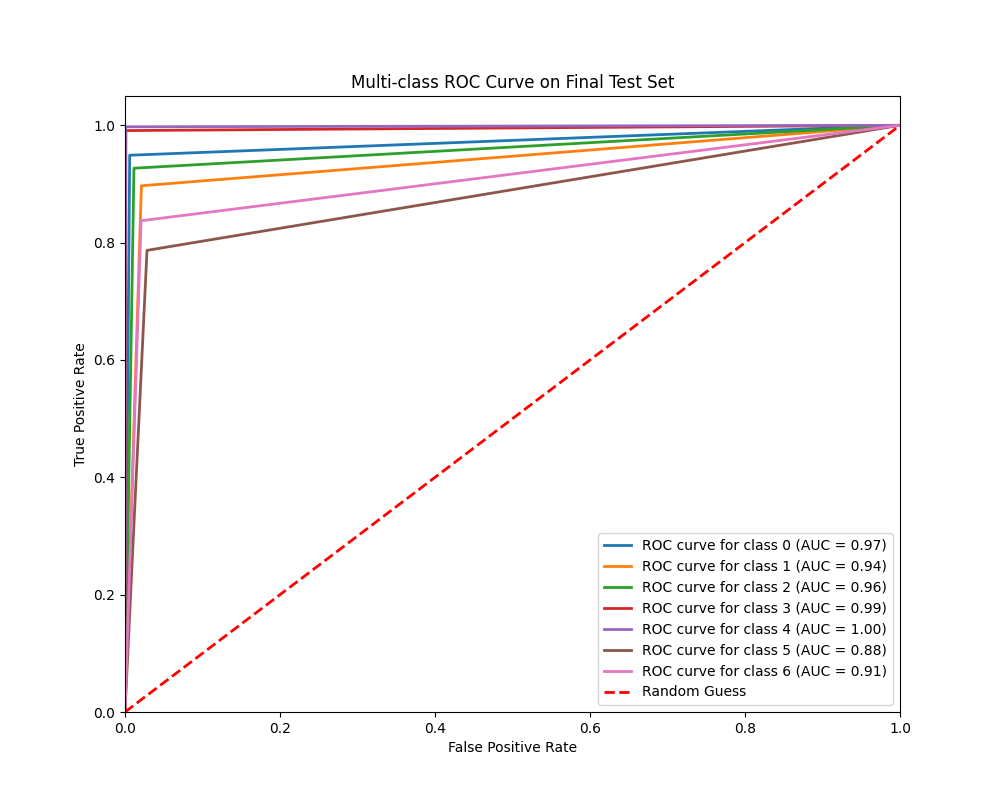


* **随机森林：**

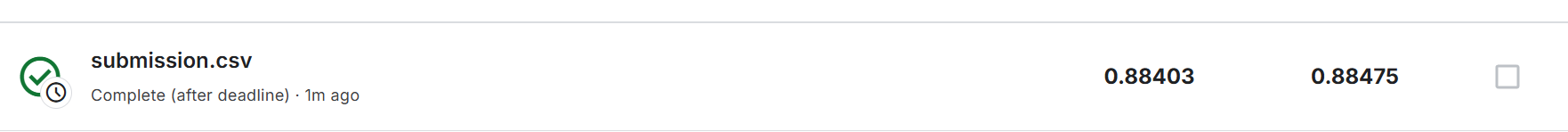
混淆矩阵：



ROC曲线：

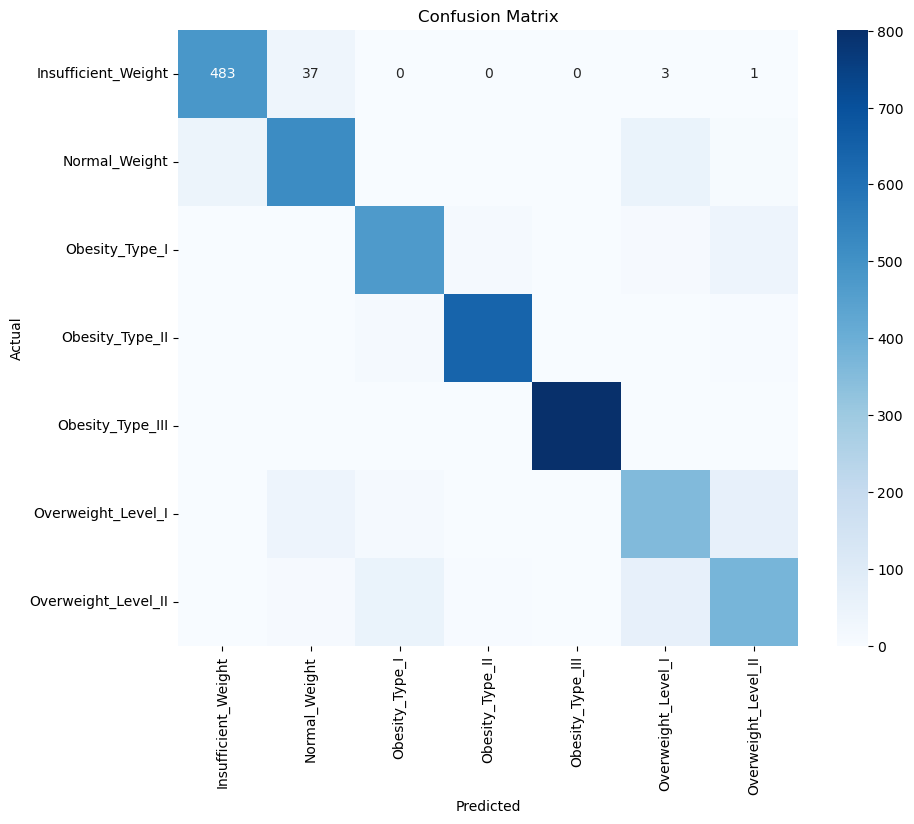


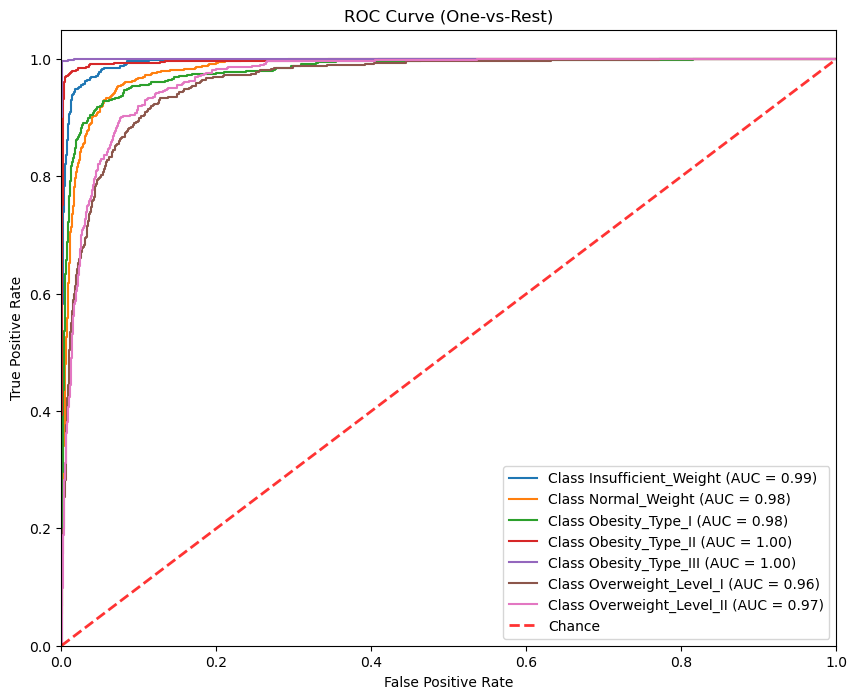
平台测试集提交分数：



* **SVM（rbf）：**

混淆矩阵、ROC曲线：



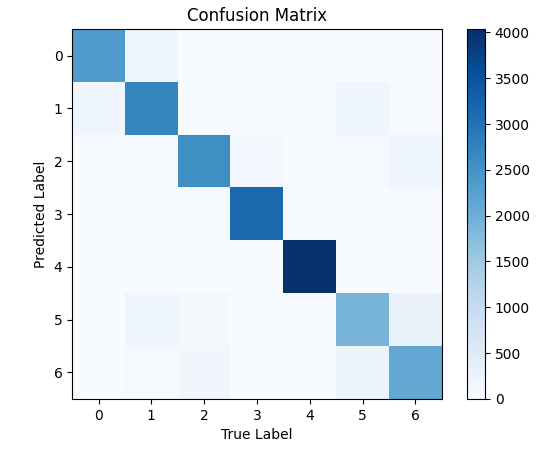


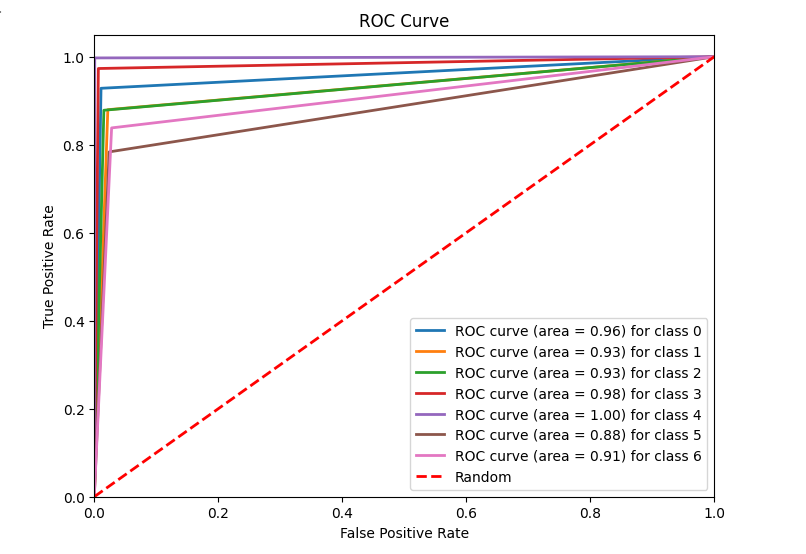
平台测试集提交分数：



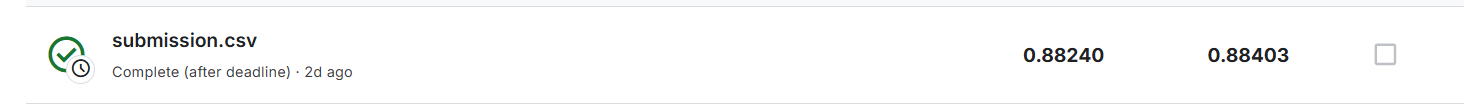
* **DNN调库实现：**

混淆矩阵：

ROC曲线：

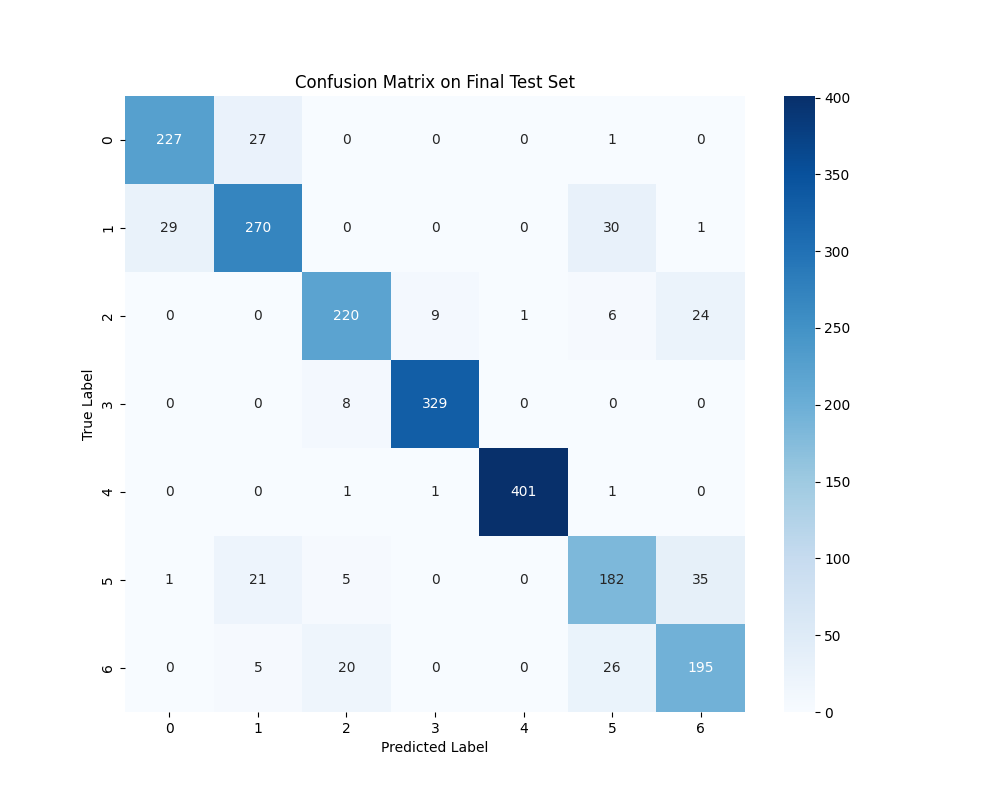


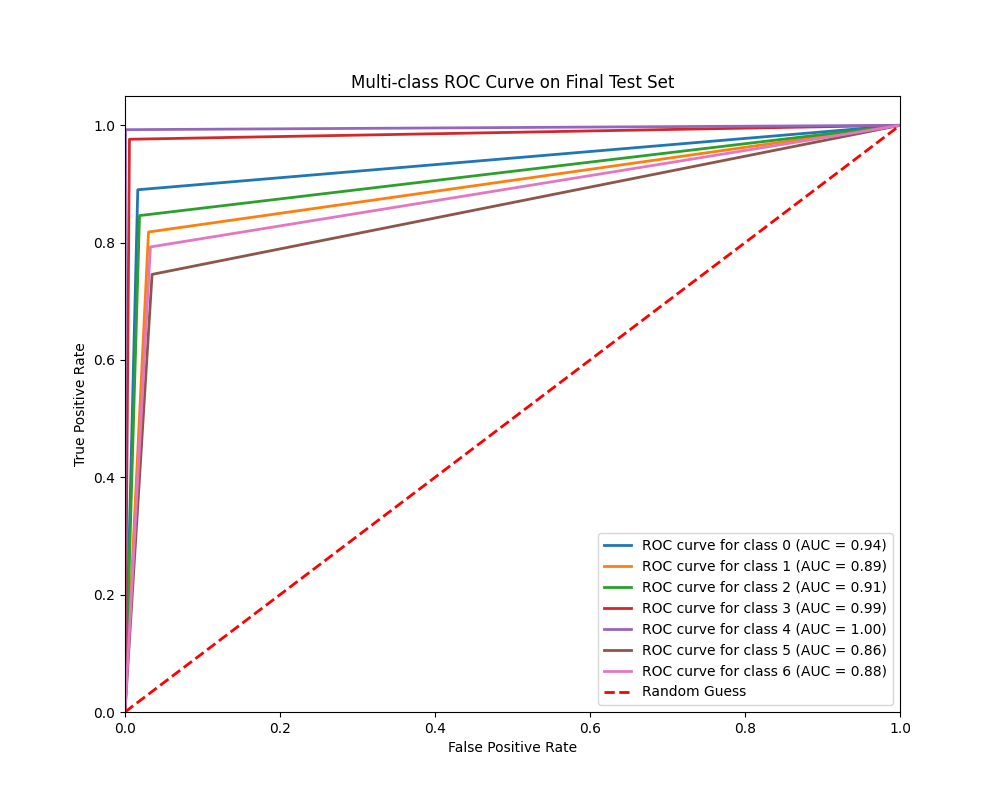
平台测试集提交分数：



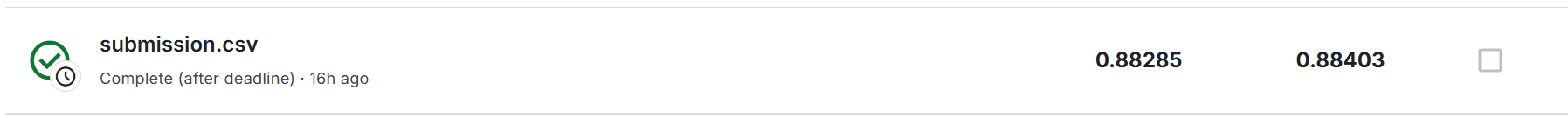
* **DNN手动复现：**

混淆矩阵：

ROC曲线：

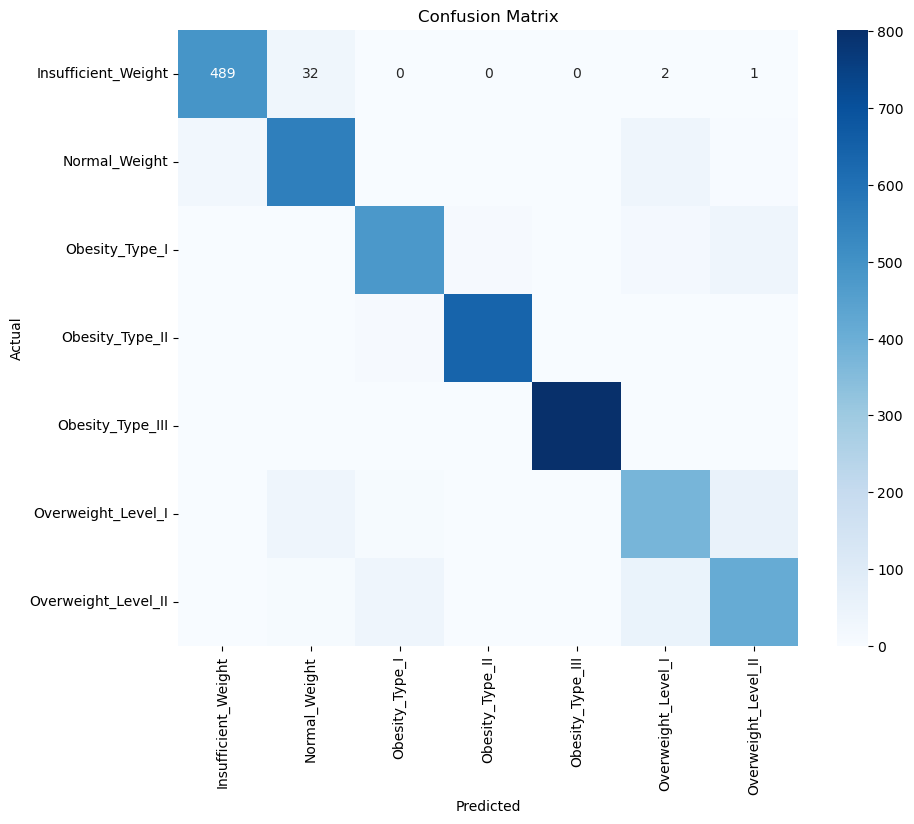


平台测试集提交分数：

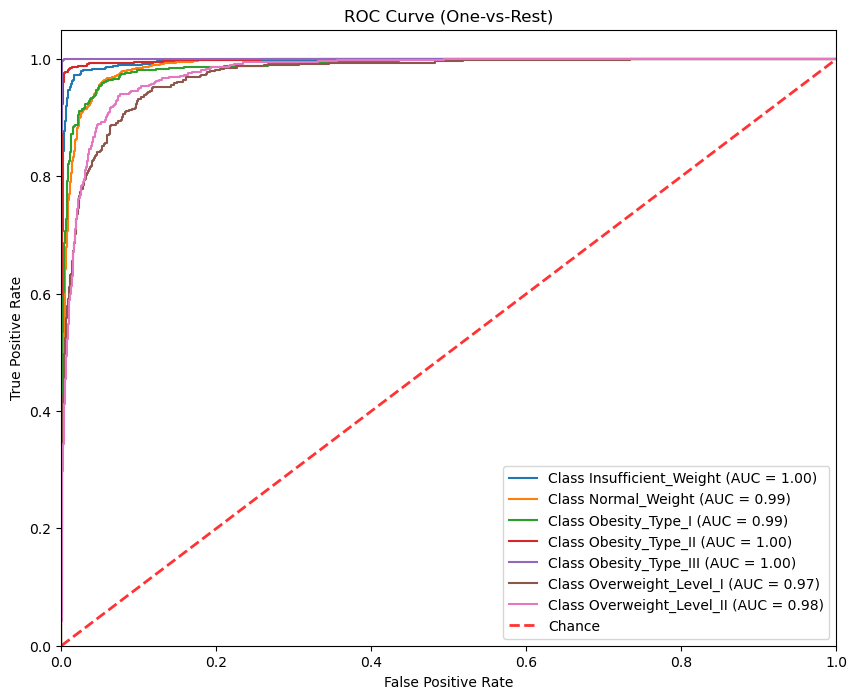


* **LightGBM：**

混淆矩阵：



ROC曲线：



平台提交分数：



* 1. **分析和优化**

**2.6.1 模型结果对比分析**

最终各模型评价指标对比如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模型\指标 | 验证集准确率 | 验证集F1 Score | 验证集ROC AUC | 测试集提交分数 |
| KNN（手写） | 0.74 | 0.74 | 0.88 | 0.7360 |
| SVM（手写线性） | 0.75 | 0.75 | 0.75 | 0.7487 |
| SVM（手写多项式） | 0.83 | 0.83 | 0.83 | 0.8331 |
| SVM（手写rbf） | 0.88 | 0.88 | 0.88 | 0.8820 |
| RF（手写） | 0.92 | 0.92 | 0.92 | 0.8840 |
| DNN（调库） | 0.9096 | 0.8969 | 0.94 | 0.8824 |
| DNN（手写） | 0.88 | 0.88 | 0.88 | 0.8828 |
| LGBM | 0.91 | 0.91 | 0.91 | 0.9032 |

各模型的结果分析如下：

**2.6.2 KNN结果分析**

为改进传统KNN中简单多数投票的局限性，模型引入**加权投票机制**，根据距离赋予每个邻居不同权重。距离越近对结果影响越大，从而提升预测准确性。为计算权重，对每个邻居的距离取倒数（并加小常数避免除零错误），距离越小的邻居权重越大。最终根据每个类别的权重和来确定预测类别。

**KNN在高维数据中效果较差原因分析：**

* 高维空间中数据点间的距离容易变得相似导致邻居难以区分，SVM等模型可通过核方法处理高维数据，效果较好。
* 噪声敏感：KNN对噪声数据敏感，少量异常点会对结果造成很大影响。
* 特征选择：KNN对特征选择敏感，不同特征的尺度和相关性都会影响距离的计算，导致分类效果不佳。

**2.6.3 SVM结果分析与优化**

手写实现SVM时最初只实现了线性核函数的SVM，测试集和验证集准确率都约0.75，有较大改进空间，分析原因可能是数据集线性不可分，故考虑更换核函数与交叉验证两个优化方向。加入5折交叉验证后准确率提升至0.85。

* Rbf核函数的SVM：实现过程中主要遇到了参数选择和计算效率的问题，线性核函数只需要调整正则化参数C，而RBF核函数除了C还需要调整核函数的参数gamma,最初手动调参，优化过程中通过交叉验证确定最佳值，模型准确率有所提升。
* 由于RBF核函数需要计算每个样本对之间的距离，计算成本远高于线性核函数的情况，不做优化的情况下模型训练需要接近10小时，故采用SMO算法进行时间方面的优化。
* 经过验证，原始数据集线性不可分，优化后的预测准确率约0.88，效果有一定提升。

**2.6.4 随机森林结果分析与优化**

* **模型性能分析：**

混淆矩阵：从混淆矩阵中可看出大部分类别的分类效果较好，尤其是类别3和4，正确率接近完美。然而类别5和6出现一定的分类错误，特别是类别5被预测为1和6的次数较多，说明模型对这两类的识别不够精确，导致混淆。

分类报告：分类报告中的精确率、召回率和F1分数表明类别4表现最佳，F1分数达到1.00，分类完全正确。类别5、6的F1分数较低（0.79和0.84），说明在这些类别上模型的表现相对不好。

K折交叉验证：平均验证准确率0.887，表明模型在验证集上的表现较稳定，但仍有提升空间。

ROC曲线：多分类ROC曲线显示了不同类别的AUC值。类别4、3的AUC分数接近1.00，表现好；类别5的 AUC分数为0.88，较其他类别低，再次验证了类别5在模型中的辨识度不高。

* **模型优化方向**

特征选择和工程：进一步分析特征重要性，若存在对类别5和6的识别有贡献的特征，可以加大权重或进行特征工程增强。

增加决策树数量：当前模型n\_estimators为10。适当增加决策树数量为12，可进一步提升模型的泛化能力，提高对类别5和6的预测准确性。

深度调优：模型当前max\_depth为10，可对不同的max\_depth进行网格搜索或随机搜索以找到最佳深度。

最小样本分裂数：当前设置为2，可以适当增大min\_samples\_split值，以减少噪声带来的误差。

**2.6.5 DNN结果分析与优化**

初始模型使用了5个隐层，并且层数逐渐减少。而改进后的模型使用了更多的层次（共9个隐层），深度为64、128、256、512等逐渐增大的结构，再逐渐减小。这种逐步加深再缩小的设计可以帮助网络更好地提取特征，并在不同的层次学习不同抽象级别的特征。

**Dropout 层：**增加了多个 Dropout 层，在不同层次中使用了不同的 Dropout 比率（如0.05、0.1、0.15），可以帮助进一步防止过拟合，并增强模型的泛化能力。

在训练过程中逐步降低学习率（0.0001、0.00005、0.00001等）。这有助于在模型收敛时进一步精细化权重更新，使得模型可以更精确地找到最优解。

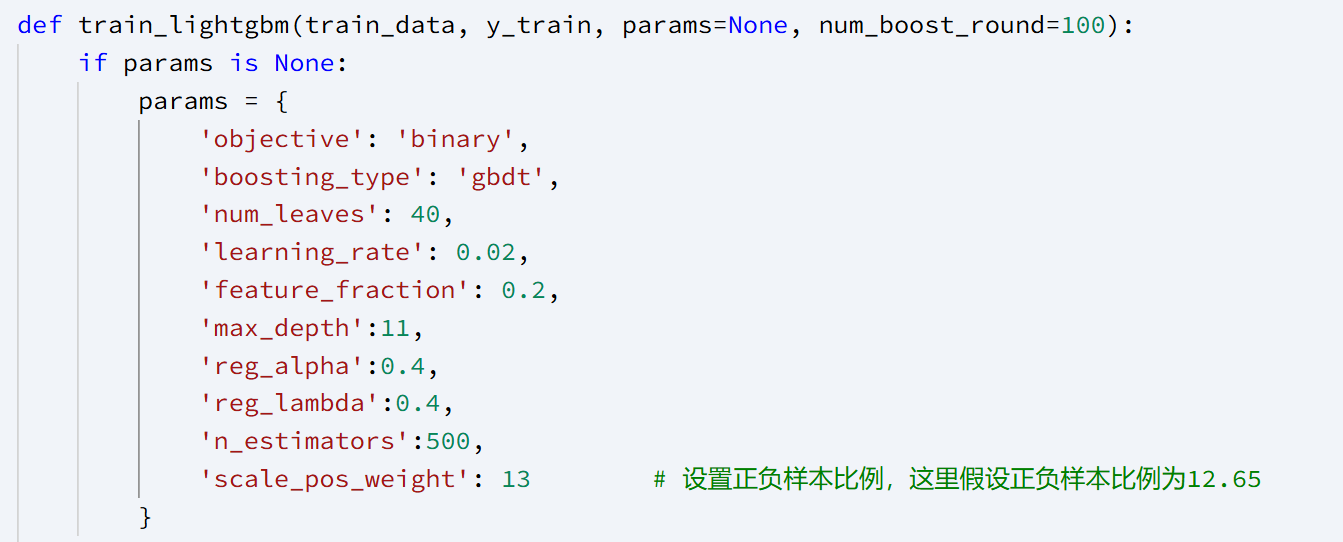
**引入5折交叉验证**，并且通过**调整不同的学习率**来验证模型的性能，比单次训练和验证更能代表模型的真实表现。

对于复现模型，其在整体分数上与调库模型相差不多，之间的差异几乎可以忽略，但由于复现模型并没有做系统的优化，使得在运行速度上大幅落后于调库模型。

**2.6.6 LightGBM结果分析与优化**

LightGBM在本次任务中分类结果较好，可能是由于其每次选择分裂增益最大的叶子节点进行分裂，从而更快地达到最优解，另外它自动将连续特征分桶，减少了内存占用和计算时间，同时保持了模型的准确性。

为了防止过拟合，在模型中引入L1和L2正则化惩罚项，并将树的层数设置的较深，叶子节点也没有设置的太高：



后续优化可以使用模型融合的策略，如使用多个不同的模型（LightGBM、XGBoost、RF等）集成，通过投票或加权平均提高分类性能；或使用Stacking方法，通过堆叠多个模型，将前一个模型的预测结果作为下一个模型的输入特征，进一步提升模型的性能。平台上的高分模型可证明该优化思路的可行性。

**三、实验总结**

1. **总结**

本次分类实验中，我们手写构建了KNN、SVM、随机森林、DNN以及LightGBM五种模型，并采用了准确率、F1 Score、ROC曲线和AUC等指标来评估模型性能，其中Lightgbm、SVM、随机森林、DNN分类模型在多分类肥胖风险预测任务中表现较好。

总体来说：KNN模型在处理高维数据和噪声数据方面表现不佳，SVM模型通过选择合适的核函数和参数调优能够处理复杂的非线性关系，但计算成本较高；随机森林模型通过增加模型的差异性和泛化能力表现出色，但在某些特定类别上仍需优化；DNN模型通过多层非线性变换和正则化技术表现出色，但手动实现的模型需要进一步优化；LightGBM模型通过高效的算法和正则化技术表现出最强的分类能力和最高的泛化能力。我们也对SVM、随机森林、DNN的结果进行了重点分析与优化，手动实现模型的底层代码也让我们对原理的理解更为深刻。

未来的工作可以进一步探索模型融合和高级优化技术，以提升模型的整体性能。通过本次实验，我们掌握了不同分类模型的使用方法、优缺点及适用场景，深化了对机器学习模型构建、优化及评估的理解，为未来更复杂的建模工作奠定了基础。