**华中科技大学**

**计算机科学与技术学院**

**《机器学习》结课报告**



专 业： 计算机科学与技术

班 级： CS2306

学 号： U202315712

姓 名： 张致远

成 绩：

指导教师： 张腾

**完成日期：2025 年 5 月 30 日**

目录

[1. 实验要求 2](#_Toc131770776)

[2. 算法设计与实现 2](#_Toc131770777)

[3. 实验环境与平台 2](#_Toc131770778)

[4. 结果与分析 2](#_Toc131770779)

[5. 个人体会 2](#_Toc131770780)

Adaboost算法实现

# 实验要求

大作业要求分别实现以对数几率回归和决策树桩为基分类器的 AdaBoost

算法，其中对数几率回归请参考课件 5.3 节，决策树桩指只有一层的决策树，决

策树请参考课件第 12 章，AdaBoost 算法请参考讲义。

实现以上算法时不允许调用已实现好的任何机器学习模型库。

你的代码需读取 data.csv 及 targets.csv 两个文件，并输出在不同数目基

分类器条件下的 10 折交叉验证的预测结果至 experiments/base#\_fold#.csv，

以供评测。基分类器数目取 1，5，10，100 这四种数值。输入样例，输出样

例及评测代码详见提供的压缩包。

对预测结果所在文件命名格式说明如下：基分类器数目为 x 对应的预测文

件为 basex\_fold1.csv~basex\_fold10.csv，1~10 指的是用作测试集的子集编

号。每个预测文件分成两列，第一列为样例的序号（序号从 1 开始），第二列

为该样例的预测标记。评测时子文件夹 experiments 会建立好，请不要在你的

代码中强行建立此文件夹以免出错。如果对文件名和格式还有疑问，可以参考

压缩包中提供的输出样例和评测代码。

# 算法设计与实现

1. 决策树桩的实现

决策树桩作为AdaBoost算法中的基础分类器，

首先初始化样本权重，权重是一个形状为(n\_sample,)的数组，样本权重决定了每个样本在计算分类误差时的重要性。

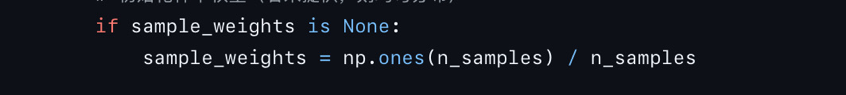


图1

接下来遍历所有特征，对每个特征计算所有唯一值的中点作为候选阈值，对 每个候选阈值，尝试两种分裂方向（polarity=1或-1），再计算每种分裂的加权错误率。

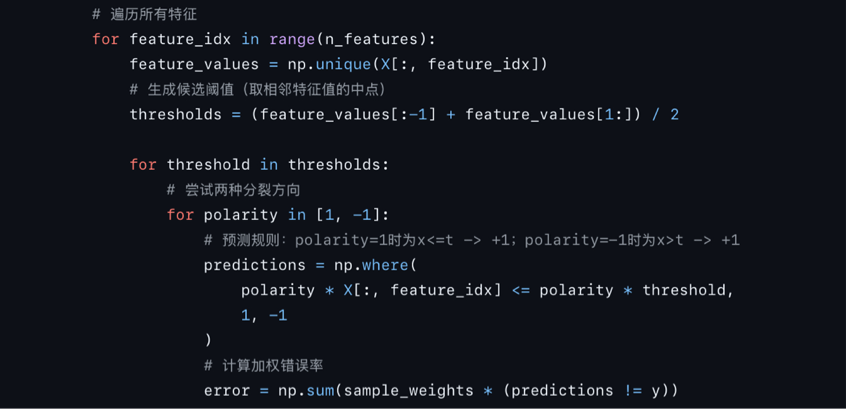


图2

预测时，当分类方向polarity=1时，x<=1预测+1，否则预测-1，分裂方向为-1时则相反。

1. LogisticRegresson实现

实现sigmoid函数将线性组合映射到（0,1）概率空间

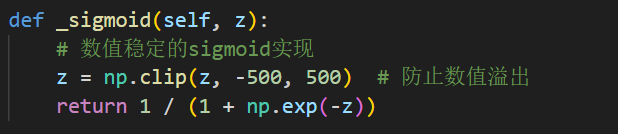


图3

计算交叉熵损失作为梯度下降优化的loss

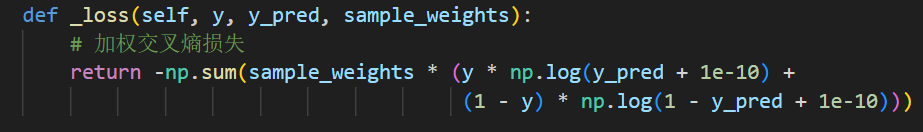


图4

最后进行梯度下降优化：

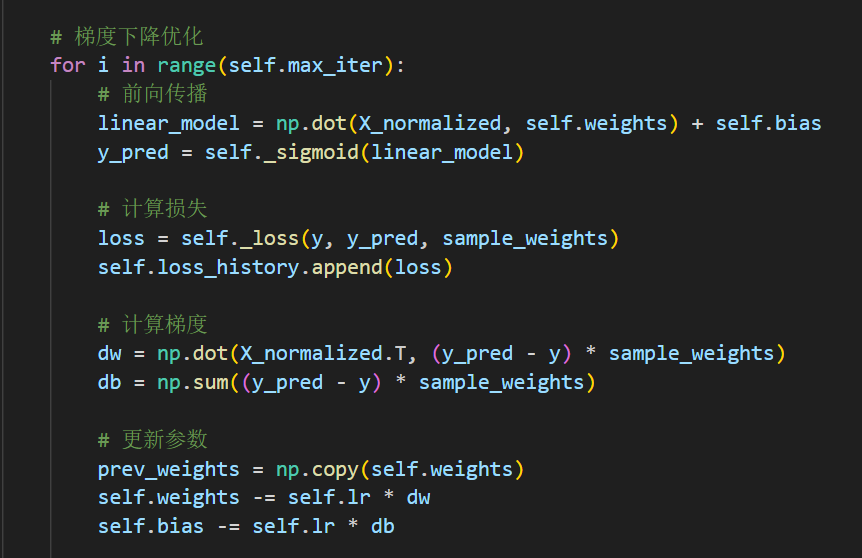


图5

1. AdaBoost的实现

adaboost算法通过组合多个弱分类器来构造强分类器，通过多轮裁判，每一轮调整样本权重，使后续分类器更关注之前错误分类的样本，最终预测是所有弱分类器的加权投票结果

首先初始化样本权重

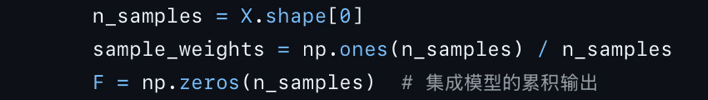


图6

然后迭代训练弱分类器，计算分类器权重，更新样本权重

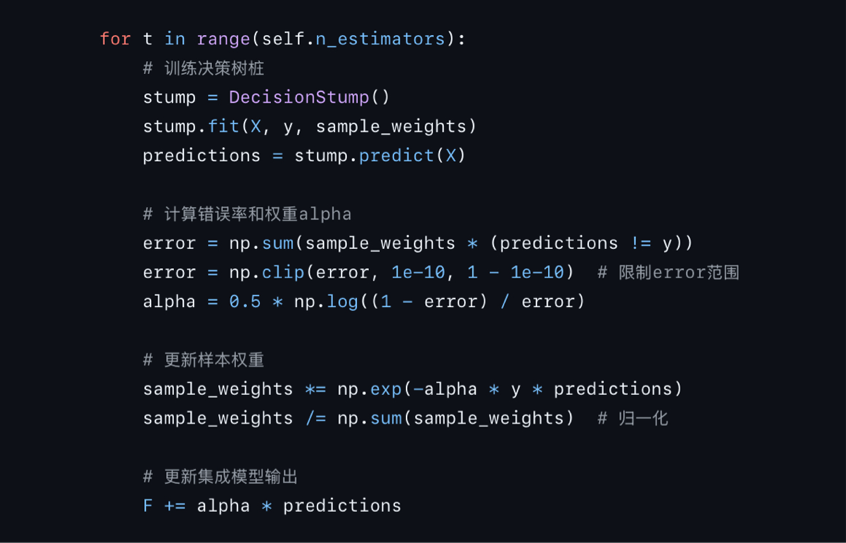


图7

预测时让所有基分类器同时预测，如何在根据此前训练得到的投票加权合成最后的预测。

# 实验环境与平台

CPU 12th Gen Intel(R) Core(TM) i5-12500H

Windows11操作系统

Python 3.8

# 结果与分析

采用1,5,10,100个决策树桩作为基分类器，取十折中九折作为训练集，一折作为训练集，训练、测试十次后取准确率的平均值，共得到四组准确率。

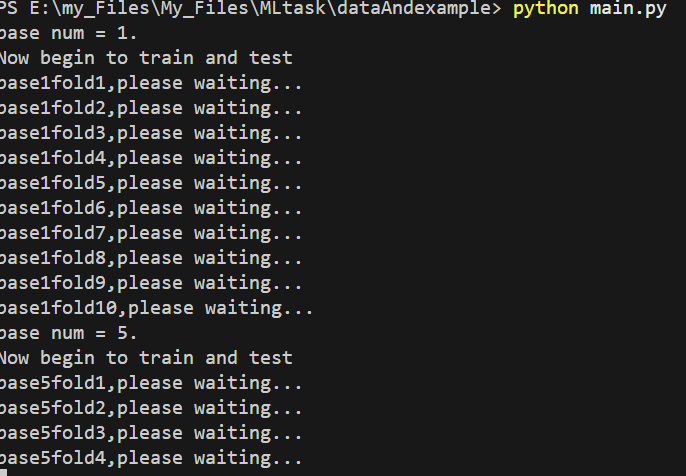


图8

测试评估结果如下

|  |  |
| --- | --- |
| 基分类器数 | 平均准确率/% |
| 1 | 59.8 |
| 5 | 62.4 |
| 10 | 75.3 |
| 100 | 76.1 |

表1

在adaboost文件中使用sklearn中的决策树代替朴素决策树，结果较好，于是查询资料优化决策树：

1. 采用基尼不纯度作为分割标准，更敏感于类别分布的变化更适合处理类别不平衡的数据。

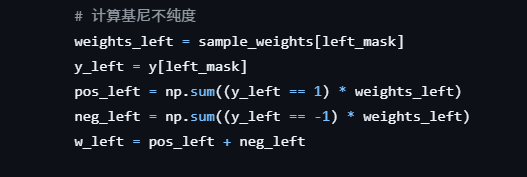


图9

1. 先排序特征值，再选择分割点，处理了重复特征值的情况，确保只在实际可分割的点上进行评估与样本权重配合更好(保持权重与样本的顺序一致性)

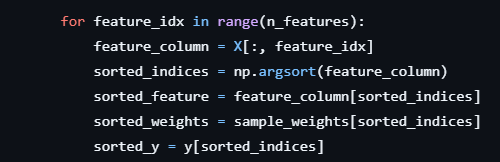


图10

采用新的决策树桩在进行测试

|  |  |
| --- | --- |
| 基分类器数 | 平均准确率/% |
| 1 | 71.2 |
| 5 | 86.3 |
| 10 | 86.7 |
| 100 | 86.7 |

可以看出有较大程度的优化。

对数几率回归的结果如下：

|  |  |
| --- | --- |
| 基分类器数 | 平均准确率/% |
| 1 | 88.4 |
| 5 | 88.8 |
| 10 | 88.9 |
| 100 | 89.1 |

# 5. 个人体会

通过本次实验，我对AdaBoost算法及其实现有了更深入的理解，同时也积累了宝贵的实践经验。在实现AdaBoost算法时，我深刻体会向SOTA模型学习的重要性。最初使用简单的决策树桩作为基分类器时，模型性能较差。通过优化决策树桩的分割标准（如采用基尼不纯度）和特征值处理方式（排序后选择分割点），模型的准确率显著提升。这让我意识到，即使是弱分类器，其设计和优化也能对整体模型性能产生重大影响。

AdaBoost的核心思想是通过调整样本权重，使模型更关注难以分类的样本。在实现过程中，我观察到随着迭代次数的增加，样本权重的分布会逐渐偏向那些被错误分类的样本，从而帮助模型逐步改进。这一机制让我对集成学习的“集思广益”有了更直观的认识。

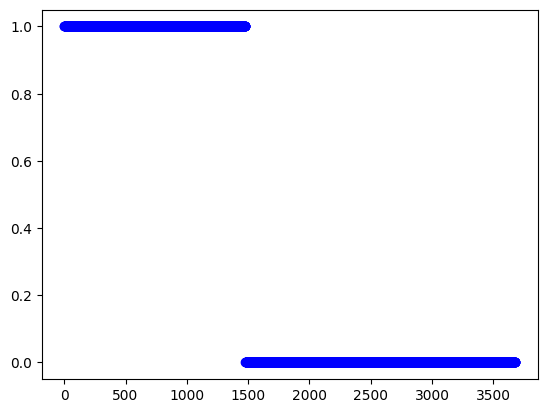


图11. 3680条数据的分布可视化

还可以改进的是训练集和测试集的排布（如图8），因为题目所测试用的target.csv文件中的顺序不能改变，所以可能还是会造成一些问题，因为正例和负例的分布极不均衡，有的训练集中只有正例或者负例。