# 模式识别作业——西瓜数据集3.0αAdaboost算法训练

# 实验环境

## 数据集的选取

本次实验选用的是周志华编著的机器学习教材89页的西瓜数据集3.0α

## 编程工具的选取

本实验使用python语言，由于本实验中模型的训练消耗时间不多，代码复杂度也较低，因此jupyter notebook在训练模型中交互式运行的优势并没有那么大，故采用pycharm编程环境。

## 库的选取

本实验采用sklearn库中的函数训练模型

## 操作系统

操作系统为Windows10操作系统，版本为19043.1645

# 主要函数

我首先尝试了自己编程实现一个Adaboost算法，但是由于本次数据集有多个特征，限于本人编程水平有限，我并没有找到合理的选择下一步选取的特征的方法（尝试了计算GINI系数与信息增益比，但是可能我对过程理解不够细致导致程序出现了小BUG），因此也就放弃了自己编程实现，而是采用sklearn库中的adaboost算法。

## 训练部分

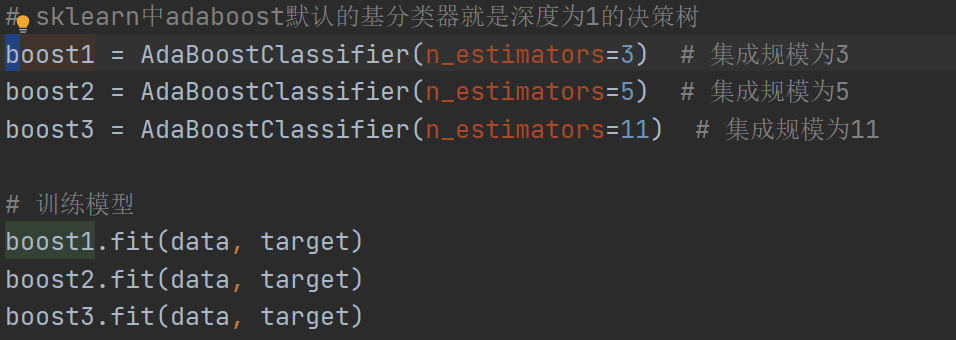
### 数据集导入

首先导入数据集，由于数据集规模不大，我在主函数中手动定义了数据集。考虑到sklearn中Adaboost算法的输入是np数组类型的标签和数据，我采用字典的形式存储数据集。字典的键是”data”和”target”，字典的值是np数组。

### 模型训练

有了数据集后，就可以给模型喂数据了。Sklearn中提供的Adaboost算法接口是AdaboostClassifier(base\_estimator=None, \*, n\_estimators=50, learning\_rate=1.0, algorithm='SAMME.R', random\_state=None)。第一个参数是基学习器，默认是深度为1的决策树。n\_estimators是基学习器数量，本实验中创建三个AdaboostClassifier对象，学习器数目分别为3,5,11。Learningrate是学习率。后边两个参数分别是采用的算法和随机数种子。

实际操作也很简单，仅需下边两步就可以基本实现我们的任务要求，接下来就是可视化。

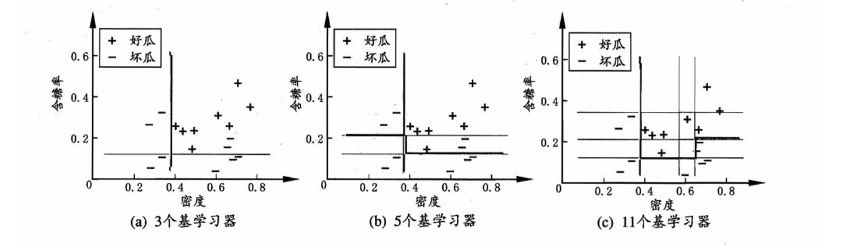


## 可视化部分

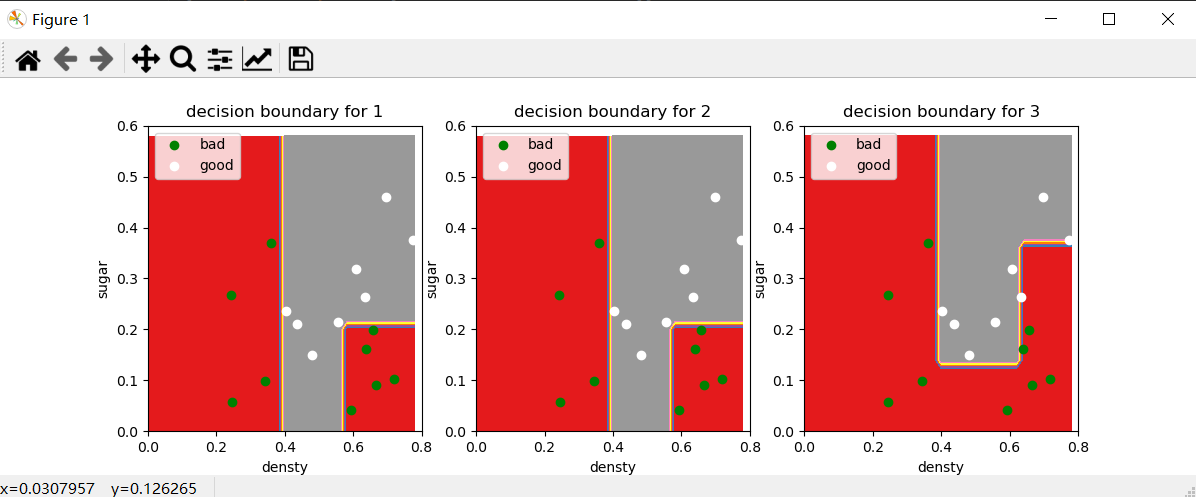
这部分思路也很清晰，大致思路如下：

1. 设置坐标最小值、最大值与间隔。
2. 然后依次遍历上述三个模型boost1，boost2，boost3，按列叠加两个矩阵并添加到label\_set当中。
3. 由于我们有3张图，因此需要3个坐标轴，创建这三个坐标轴。
4. 然后循环3次，每次画一张图。画图具体方法：为0代表坏瓜，用绿色表示，1代表好瓜，用白色表示。
5. 最后plt.show()显示3张图即可。

## 实验结果分析

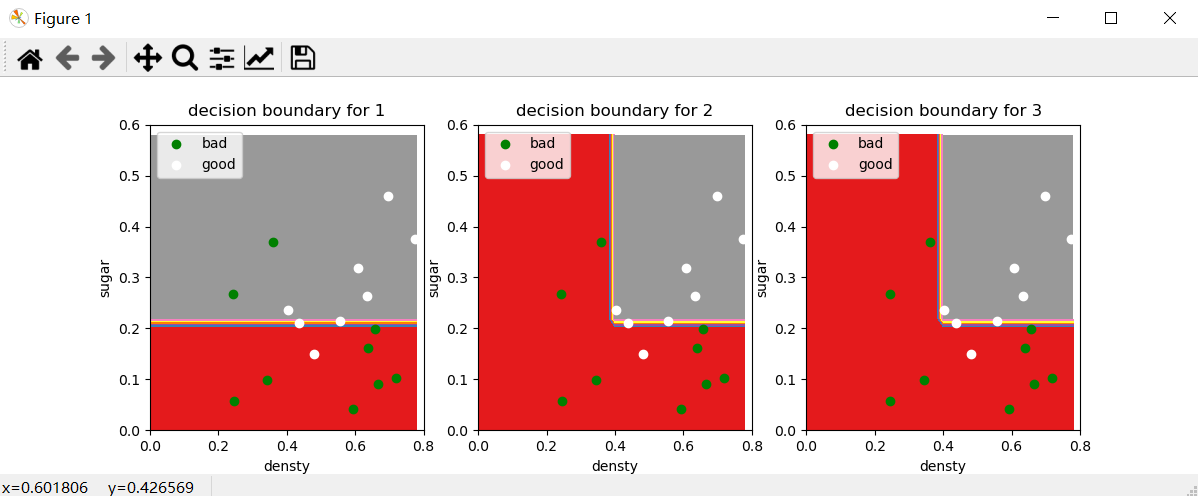


上图为书上给出的3个学习器训练效果，可以看出随着学习器数目增加分类结果也确实趋于准确。下面给出我的训练结果。

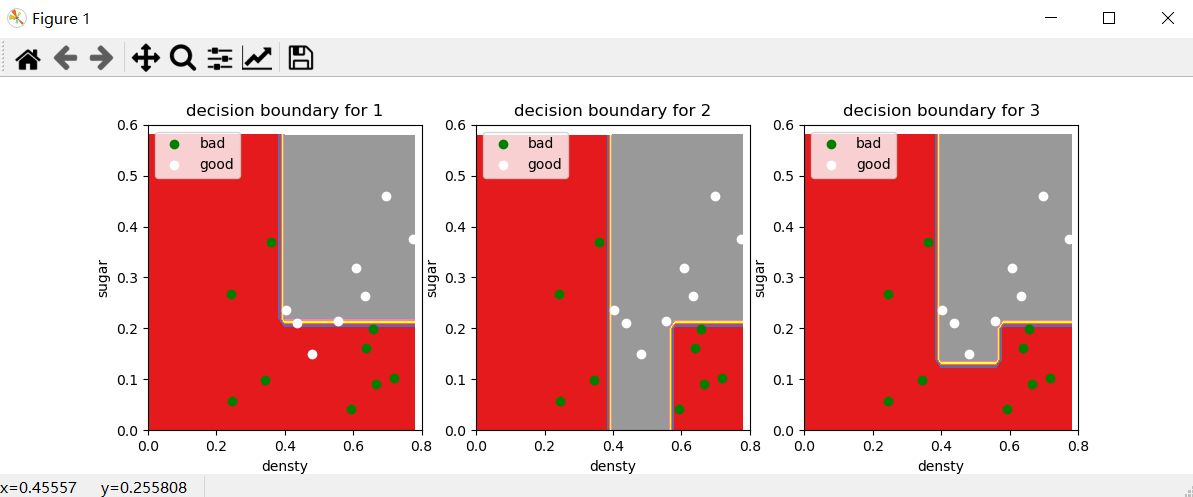


可以看出当学习器数目增加到11时，我的训练效果和书中给出的基本是一致的，当学习器数目为3或5时，我的训练效果要好于书中给出效果。究其原因，可能是选择的划分标签不同，也可能是sklearn的Adaboost采用了某些优化方法，使得学习器数目较小时也可以有相对较好的学习效果。

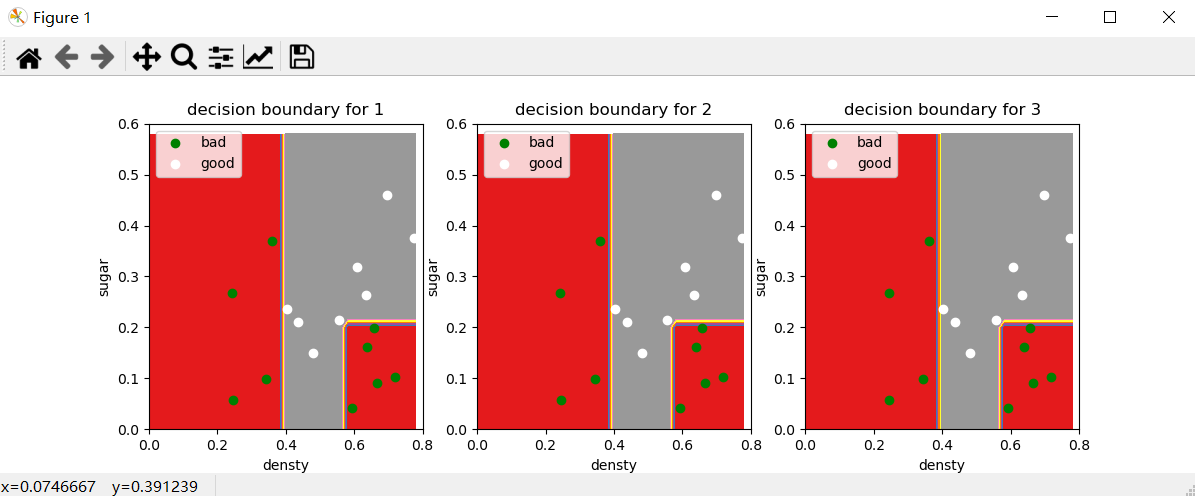
据此我又改变了参数进行测试，图如下



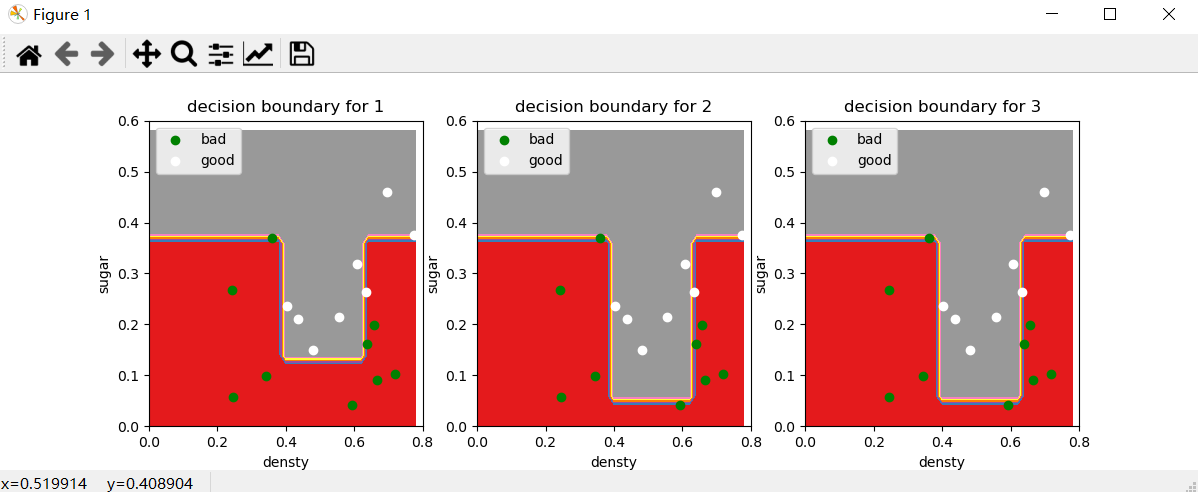
1、Learning\_rate = 0.1



2、Learning\_rate = 0.5



3、Learning\_rate = 1.5



4、n\_estimator = 15 / 50 / 100

从上图1,2,3可以看出，默认学习率1在这个案例中是不错的选择，太小或太大都会导致最终结果不收敛。

上图4可以看出，n\_estimator大于50后可以说是已经完全收敛了，更大的基学习器数模不会对最终效果产生太大影响。同时也可以推断出，对于本次这种简单的数据集，基学习器数目并不是越大越好，例如学习器数目>50后的训练效果其实不如n = 11时的训练效果，可能有些过拟合。

# 总结与感悟

通过本次实验，我对sklearn库，matplotlib库以及numpy库都有了加深的了解，同时也尝试了自己编写一些算法，虽然失败了，不过我认为这个过程也加深了我对于adaboost算法的了解。