**---（2）C4.5算法实现**

学生姓名： 学 号： 专业班级：

实验类型：□ 验证 ■ 综合 □ 设计 □ 创新 实验日期： 11.15 实验成绩：

**一、实验目的**

学会利用决策树C4.5算法对数据集进行分类，掌握C4.5算法的思想。

**二、实验内容**

根据决策树C4.5算法原理实现对数据集的分类，编译、运行程序并观察程序的输出，并分析实验结果，写出实验报告。

**三、实验要求**

1. 分析实验代码；

2. 分析运行结果；

3. 画出程序流程图。

**四、实验步骤**

1. 编写实验代码；

2. 编译、运行程序；

3. 观察程序的输出并分析实验结果。

**五、实验数据及处理结果**

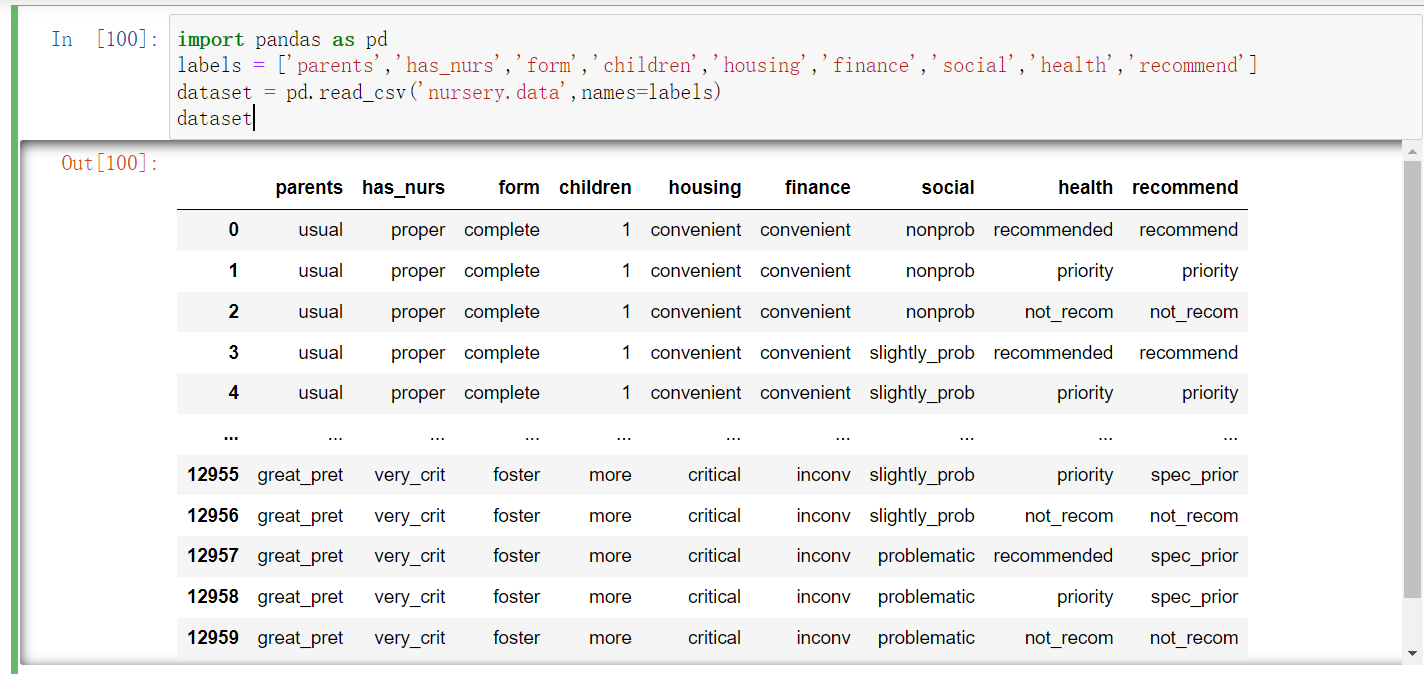
**1. 查看数据集**

托儿所数据库源于最初开发用于对托儿所应用程序进行排序的分层决策模型。20世纪80年代，斯洛文尼亚卢布尔雅那的这些学校入学人数过多，被拒绝的申请常常需要客观的解释。最终决定取决于多个子问题：父母和儿童托儿所的职业、家庭结构和财务状况以及家庭的社会和健康状况。该模型是在决策DEX专家系统外壳中开发的。

查阅资料可知，每个样本共有八个属性，分别为parents、has\_nurs、form、children、housing、finance、social、health。各属性取值如下表：

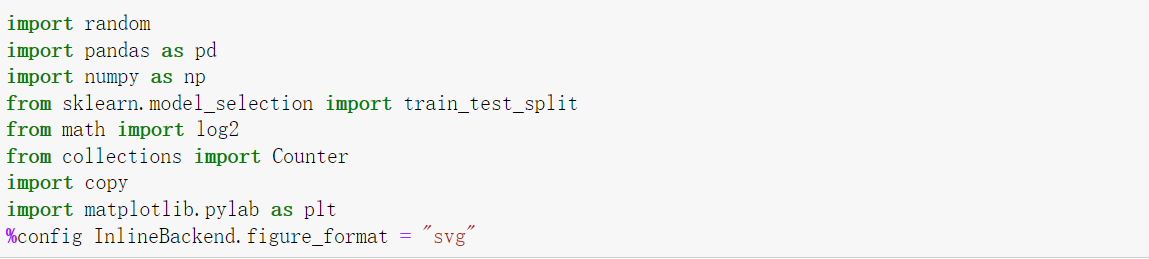
|  |  |
| --- | --- |
| 属性 | 取值 |
| parents | usual, pretentious, great\_pret |
| has\_nurs | proper, less\_proper, improper, critical, very\_crit |
| form | complete, completed, incomplete, foster |
| children | 1, 2, 3, more |
| housing | convenient, less\_conv, critical |
| finance | convenient, inconv |
| social | non-prob, slightly\_prob, problematic |
| health | recommended, priority, not\_recom |

使用Jupyter Notebook查看数据集。代码如下如下：

 由上图可知数据集共包含12960个数据样本，每个样本共有8个属性(最后一列为分类属性)，这与上文数据集介绍保持一致。本次实验正是根据这8个属性预测给定样本属于哪种类别。

**2. C.5算法实现**

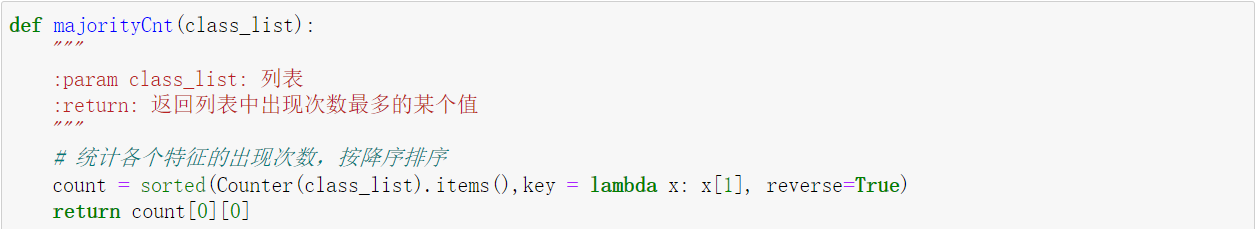
代码实现如下：





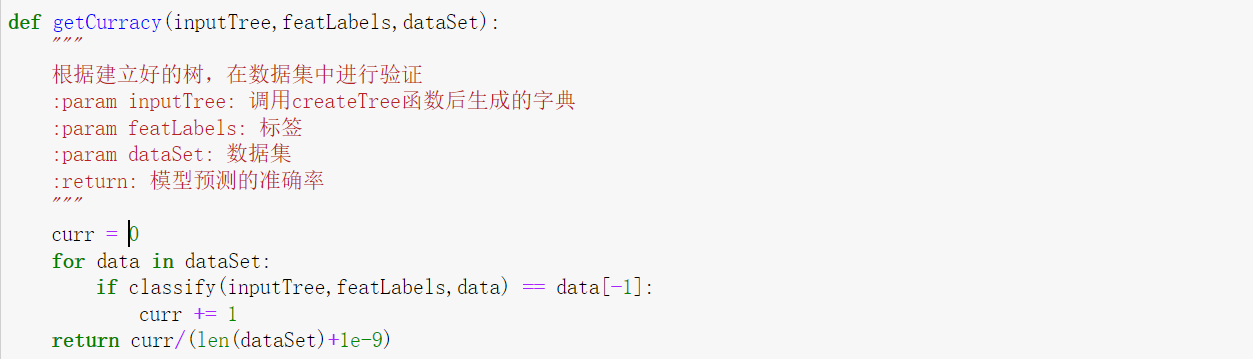


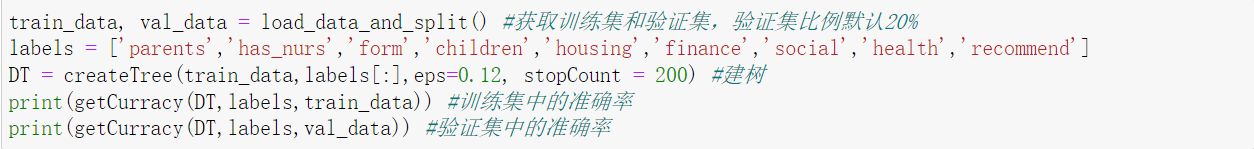












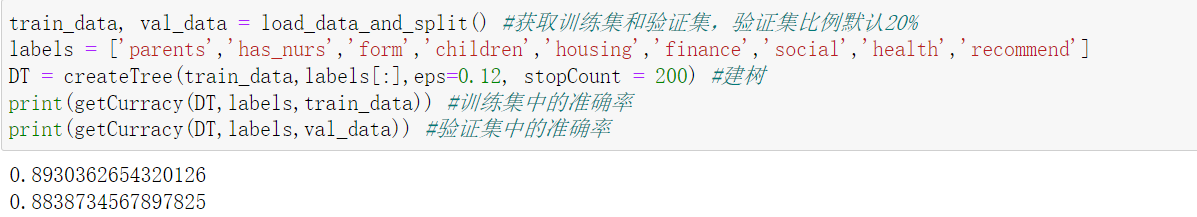
上述程序的流程图如下：



注意到，由于本次实验所采用的数据集较为庞大，所以在生成决策树时进行了预剪枝。分别设置了数据集样本数阈值以及信息增益率阈值两个阈值限制树的生长。在递归建树的过程中，如果数据集的样本数目不大于数据集样本数阈值，则采用众数投票法选择类标签生成叶子节点从而抑制生长。同理，如果该数据集划分后最大的信息增益率小于信息增益率阈值，则采用众数投票法选择类标签生成叶子节点，停止建树递归返回。预剪枝可以防止生成过于复杂且过度拟合，但是需要摸索到合适参数。

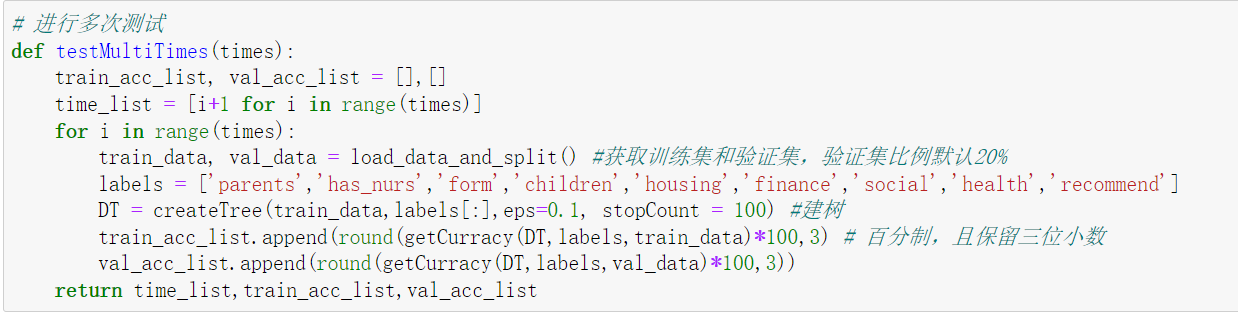
1. **运行结果**

运行上述程序，结果如下：

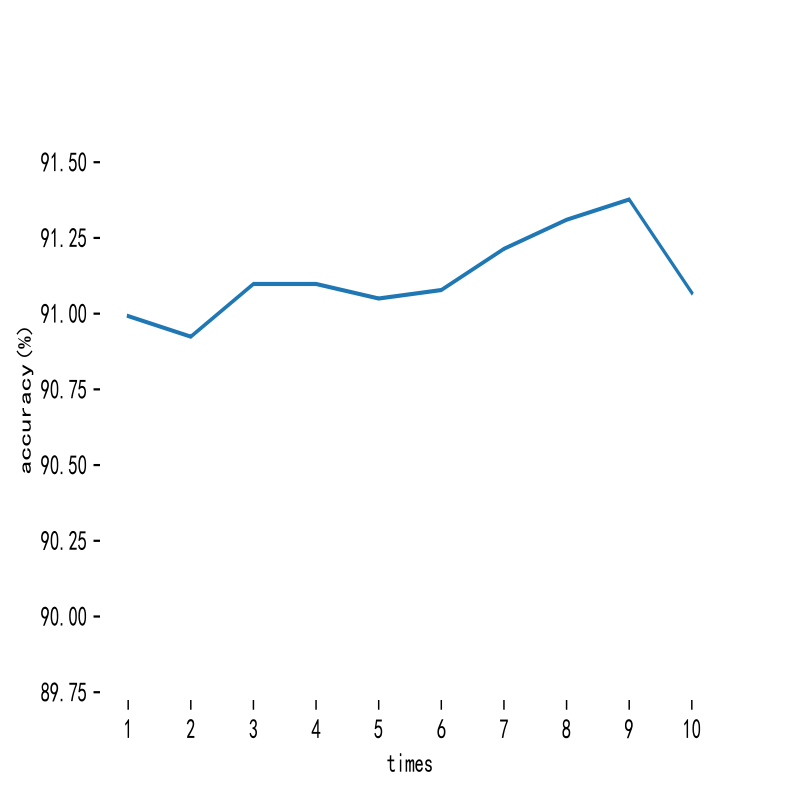


可以看到在设置信息增益率阈值为0.12，样本数量阈值为200的情况下，模型在训练集上准确率为89.3%，验证集上准确率为88.3%。两者相差不大，说明模型并未造成过拟合。至于准确率较低，则可能是参数未调整好。

注意到划分训练集和验证集时，程序会将数据集打乱并进行随机采样，所以每次运行结果不尽相同。为了观察模型的稳定性，下面进行多次测试并进行结果可视化。代码及运行结果如下：



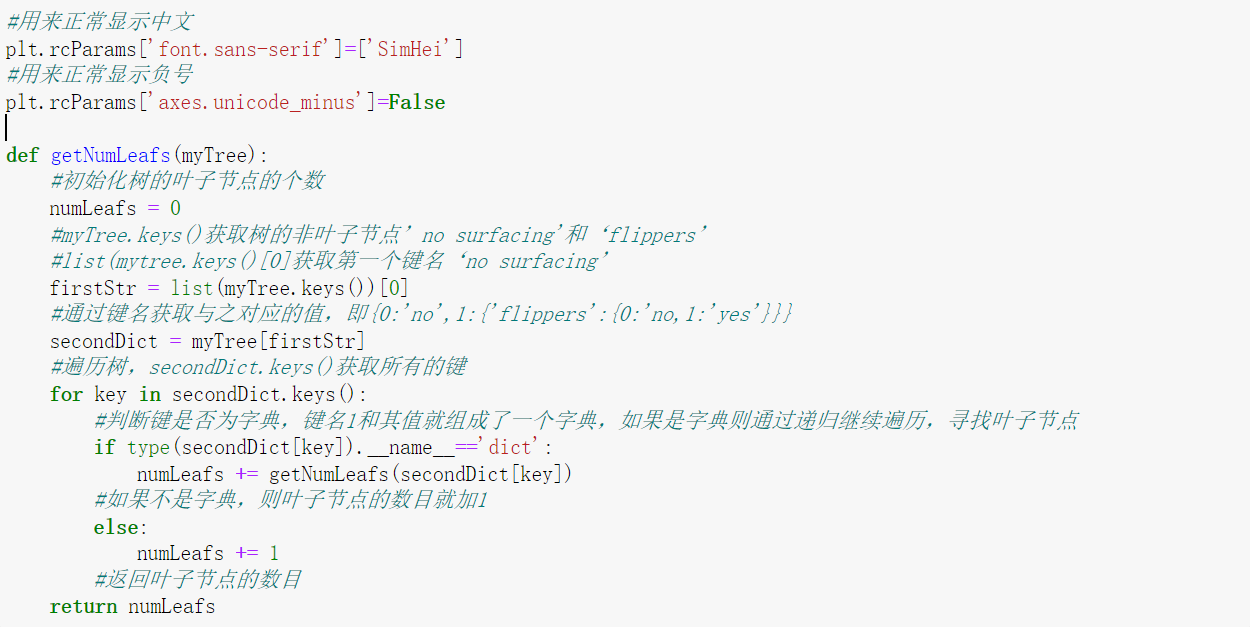




可以发现尽管每次运行测试集和验证集均不相同，但是模型在验证集上获得的准确率均保持在90%以上，且曲线稳定。而模型在验证集上所获准确率拨动较大，但与验证集上的准确率无明显差距，仍然可以接受。说明模型能够达到基本要求。

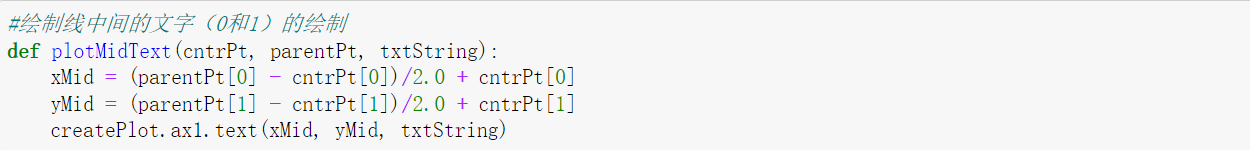
1. **结果可视化**

上述程序中，存储树的结构为字典并且是嵌套字典。如果想要观察算法的运行细节，这无疑是困难的。为了将树的结构更加直观地展现出来，下面对树结构进行可视化处理。代码如下：







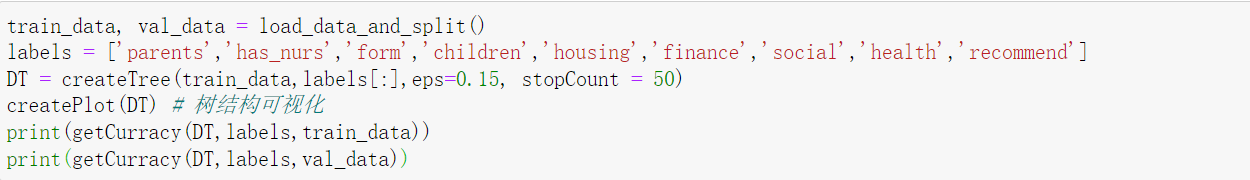


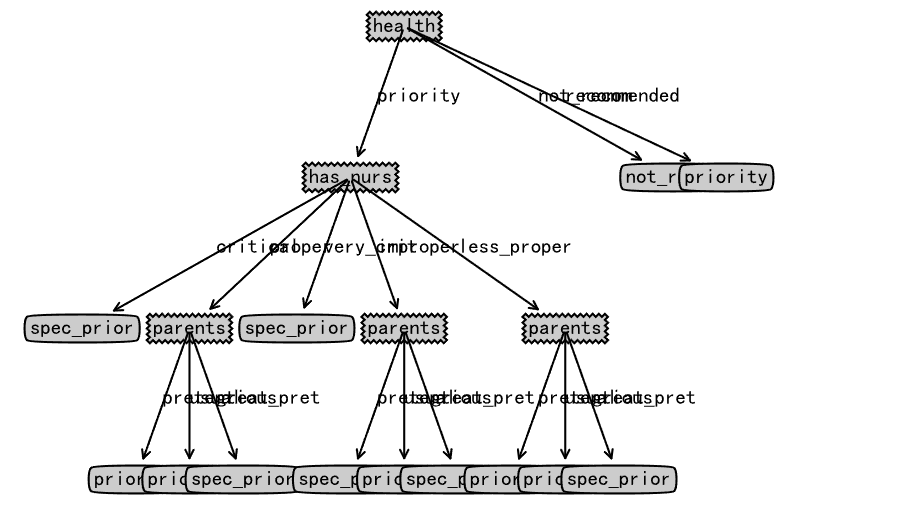




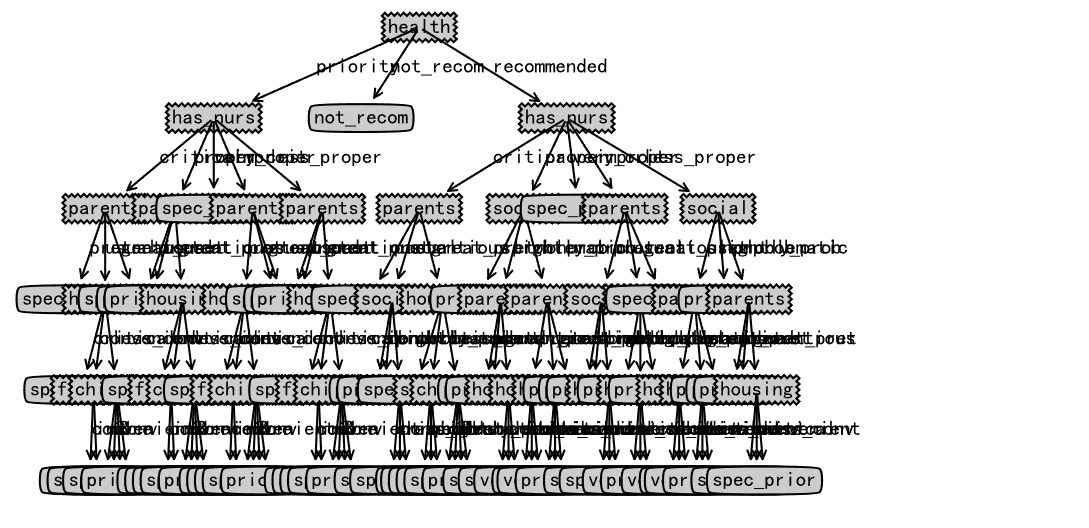
代码参考自《机器学习实战》(人民邮电出版社 Peter Harrington著)。由于该书在讲解决策树时同样使用的字典结构存储树，于是只要保证程序存储树的结构为字典，便能直接调用作者实现的函数进行结果可视化，节省大量时间。

树构建完成后，调用createPlot()函数将树结构以图像的形式展现出来。如下：

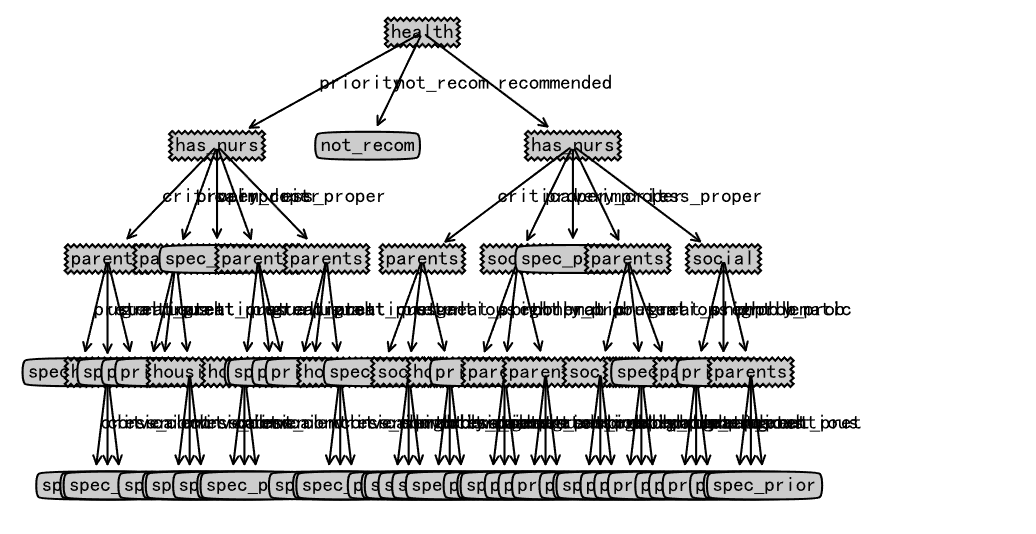




当然，树的结构复杂程序取决于参数的调整。如同样的训练集与验证集，将增益率阈值设置为0.1，其他参数不变，树结构如下：



将增益率阈值设置为0.1，样本数量阈值设置为200，树结构如下：



事实上，调整样本数量阈值相当于限制树的深度，而调整信息增益率阈值则相当于限制树的分支数量。合理的信息增益率阈值和样本数量阈值将会极大地化简树结构，同时在训练集上的准确率减低，但是泛化能力提高，可以有效地防止过拟合。过小的阈值将会让树的结构过于复杂，使用该模型进行运行和预测消耗的时间较长，同时存在过拟合风险。同样地，过大的阈值参数则会过度地化简树结构，使得模型的预测能力大大降低，欠拟合风险极大地增加。在实际应用中，需要结合具体数据集调整参数以获得适当的参数。

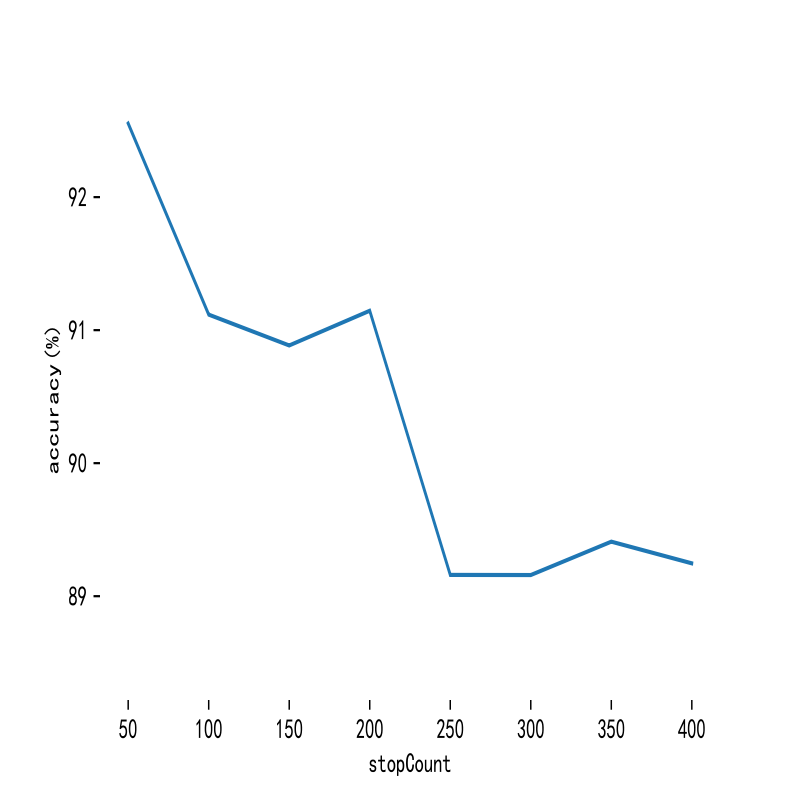
1. **参数调整**

上文提到，合适的参数对模型的评估能力至关重要。为了获取好的模型参数，需要反复调整参数进行实验，对比实验结果以筛选出令人满意的答案。下面便通过控制变量法，改变模型中的样本数量阈值以及信息增益率阈值，观察实验结果。 调整样本数量阈值。





实验结果如下：



实验中设置样本数量阈值从50变化到400，共进行8次实验。通过实验结果可以发现，样本数量阈值大于150后，训练集和验证集的准确率下降较快。样本数量阈值为50与150时，模型均能获得较高的准确率。

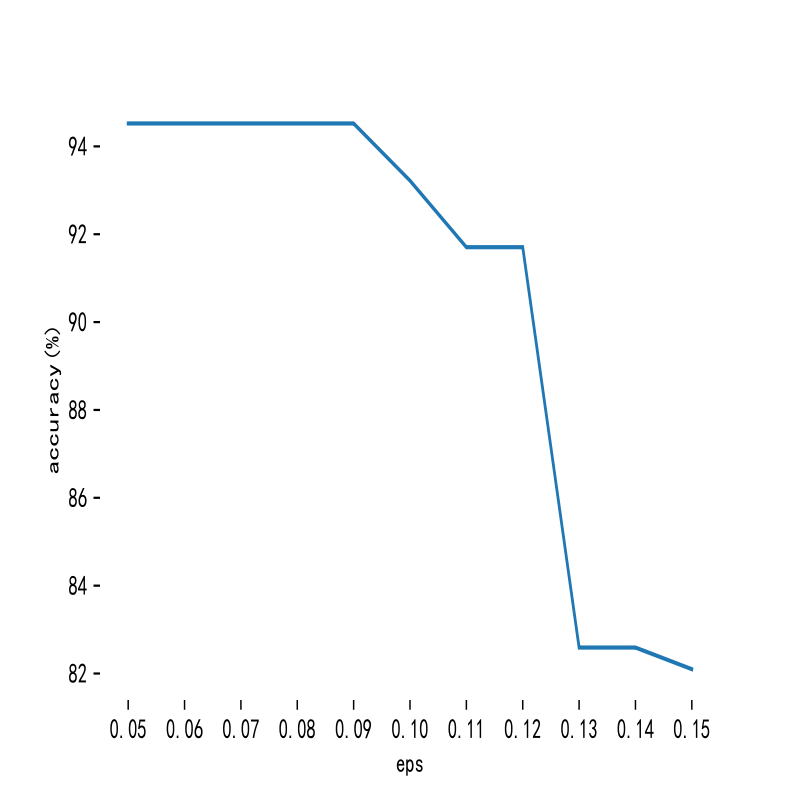
事实上，从整体来看样本数量阈值增大，势必会造成模型在训练集上的准确率下降，但同时其泛化能力提高，在验证集上的准确率可能会有所上升。这也是引入样本数量阈值的原因。

调整信息增益率阈值。





实验结果如下：



可以发现，当阈值小于0.09时，模型的准确率并无明显变化。在0.09后阈值开始出现拨动。当阈值大于0.12时，模型准确率大幅下降。说明大部分数据划分后的信息增益率都是不大于0.12。当阈值大于0.1时，验证集的准确率开始大于训练集，说明信息增益阈值设置位0.1能够有效提高模型的泛化能力。所以信息增益阈值选择0.1较为合适。

综上，样本数量阈值选择150，信息增益阈值选择0.1，模型能获得一个理想的预测效果。

**六、实验总结**

本次实验先是根据算法流程使用python搭建了一个自己的C4.5算法。为了防止模型过于复杂且发生过拟合现象，算法在生成决策树时进行了预剪枝来提高模型的泛化能力。具体做法为引入两个超参数----样本数量阈值以及信息增益率阈值。前者限制树的深度，后者限制树的分支数。在完成模型搭建后，初步实验获取了89%的准确率。说明程序实现了基本功能。

但是，由于存储树的结构为字典，想要查看生成结果的具体细节是一件困难的事情。为了解决这个问题，本文通过微调《机器学习实战》中对树的可视化代码同时结合网络上相关代码，实现了对树的可视化，同时也发现两个超参数参数对树结构的影响是巨大的。为了获得更好的分类效果以及一棵简约的树，又对模型的参数进行了调整。通过多次实验结构的比对，发现样本数量阈值选择150，信息增益率阈值选择0.1，模型的泛化能力较强，且生成的决策树简单高效。

当然，本次实验也存在不足。比如由于本人编程能力以及时间有限，未能实现决策树的后剪枝，只是在决策树的生成阶段进行了预剪枝。由于预剪枝对数据集是十分敏感的，所以模型的可拓展性也因此受到了限制。同时，实验也只选择了准确率一个指标来评判模型及选择参数，未能从多角度分析模型，存在一定的片面性。同时，也未将此模型与其他模型对比，难以客观地评判模型的好坏。

**七、参考资料**

《数据挖掘原理与实践 蒋盛益，李霞，郑琪编著》

《机器学习实战 Peter Harrington著》