1. （必填）自己提出的问题的理解（罗列全部）：
2. 提出的问题1：预剪枝和后剪枝各有什么优势？

讨论后的理解：预剪枝使得很多分支没有展开，这不仅降低了过拟合的风险，还显著减少了决策树的训练时间开销和测试时间。但是，有些分支虽当前不能提升泛化性。甚至可能导致泛化性暂时降低，但在其基础上进行后续划分却有可能导致显著提高，因此预剪枝的这种贪心本质，给决策树带来了欠拟合的风险。

后剪枝表示先从训练集中生成一颗完整决策树。从决策树的底部往上进行剪枝。后剪枝通常比预剪枝保留更多的分支，其欠拟合风险很小，因此后剪枝的泛化性能往往由于预剪枝决策树。但后剪枝过程是从底往上裁剪，因此其训练时间开销比前剪枝要大。

1. 提出的问题2：为什么说剪枝后的规则集合可能不再是互不相交且完全覆盖的？

讨论后的理解：剪枝可以是自顶向上或者自底向下的，如果生成了规则，可以剪一开始节点的规则。那么剪开始节点的限制条件可以导致该分支在二维空间上代表的区域扩张，和其他区域重合。在三叉树里面剪去一个树的分支，可以使该树不完全覆盖

还有一种理解是剪枝会删去某些属性，导致属性不完全被规则覆盖。

1. （必填）别人提出的问题的理解（选择几个问题罗列，并给出理解）：
2. 问题3：什么是离散属性，什么是连续属性？

自己的理解：离散属性就像投硬币，只有0或1，连续属性就比如人的身高，是一个连续的值，有无限种可能。

1. 问题4：选择信息增益率相比选择信息增益作为混杂函数的优势在哪里？

自己的理解：信息增益特征可以选取的值过多，导致出现偏差。信息增益率还是要比信息增益可靠的多，修正了这一偏袒性。

1. （必填）读书计划

1、本周完成的内容章节：如3.1-3.5

2、下周计划：2.1-2.5

四、（选做）读书摘要及理解或伪代码的具体实现（读书摘要、伪代码的具体实现代码等可以写到这个部分）

1、读书摘要及理解（选做）

决策树（Decision Tree）是一种简单但是广泛使用的分类器。通过训练数据构建决策树，可以高效的对未知的数据进行分类。决策树有两大优点：1）决策树模型可以读性好，具有描述性，有助于人工分析；2）效率高，决策树只需要一次构建，反复使用，每一次预测的最大计算次数不超过决策树的深度。

决策树构建的基本步骤如下：

1. 开始，所有记录看作一个节点

2. 遍历每个变量的每一种分割方式，找到最好的分割点

3. 分割成两个节点N1和N2

4. 对N1和N2分别继续执行2-3步，直到每个节点足够“纯”为止

决策树的变量可以有两种：1） 数字型（Numeric）：变量类型是整数或浮点数，如前面例子中的“年收入”。用“>=”，“>”,“<”或“<=”作为分割条件（排序后，利用已有的分割情况，可以优化分割算法的时间复杂度）。

2） 名称型（Nominal）：类似编程语言中的枚举类型，变量只能重有限的选项中选取，比如前面例子中的“婚姻情况”，只能是“单身”，“已婚”或“离婚”。使用“=”来分割。

如何评估分割点的好坏？如果一个分割点可以将当前的所有节点分为两类，使得每一类都很“纯”，也就是同一类的记录较多，那么就是一个好分割点。比如上面的例子，“拥有房产”，可以将记录分成了两类，“是”的节点全部都可以偿还债务，非常“纯”；“否”的节点，可以偿还贷款和无法偿还贷款的人都有，不是很“纯”，但是两个节点加起来的纯度之和与原始节点的纯度之差最大，所以按照这种方法分割。构建决策树采用贪心算法，只考虑当前纯度差最大的情况作为分割点。