1. （必填）自己提出的问题的理解（罗列全部）：

提出的问题1：决策树如何形成？

讨论后的理解：

首先先计算数据集中元素的属性的熵值，根据熵值大小一次加入树中，这样可以形成唯一的一棵树

提出的问题2：为什么说构建好的决策树问题是一个NP完全问题？

讨论后的理解：

问题的答案，是无法直接计算得到的，只能通过间接的“猜算”来得到结果。这也就是非确定性问题。而这些问题的通常有个算法，它不能直接告诉你答案是什么，但可以告诉你，某个可能的结果是[正确的答案](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=158477110&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)还是错误的，但是这类问题的时间复杂度一般是呈指数式增长的。这个可以告诉你“猜算”的答案正确与否的算法，假如可以在多项式时间内算出来，就叫做多项式非确定性问题。从所有可能的决策树选择最优的决策时是NP完全问题，是说最优决策树无法在多项式时间里计算出来。决策树得到的结果不是全局最优的，而是在每一个节点做的局部最优决策得到的。

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）：

问题3：预剪枝和后剪枝各有什么优劣？

* 自己的理解：
* 预剪枝

预剪枝会使得决策树的很多分支没有展开，也就是没有继续分类下去，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开销。但是另一方面，有些分支的当前划分虽不能提升泛化性能、甚至可能导致泛化性能暂时下降，但是在其基础上进行的后续划分有可能导致性能显著提升。预剪枝基于’贪心’本质，也就是能多剪枝就多剪枝，使得预剪枝策略给决策树带来了欠拟合的风险。

* 后剪枝

后剪枝决策树通常比预剪枝决策树保留了更多的分支。一般情况下，后剪枝决策树的欠拟合风险很小，泛化性能往往由于预剪枝决策树，但是后剪枝过程是在生成完全决策树后进行的，并且要自下往上地对树中的非叶子节点逐一进行考察计算，因此训练时间的开销比为剪枝和预剪枝决策树都要大得多。

问题4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？

自己的理解：我认为，在剪枝后，可能有些元素因为极为特殊（比如这一片区域中它和周围的许多点都有不同属性）而遭到剪枝。当多次剪枝后，就有可能出现集合不完全覆盖的情况.

1. （必填）读书计划

1、本周完成的内容章节：如3.1-3.4

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）

本章主要介绍了监督学习中决策树的构建。决策树的构建是一个复杂的NP完全问题，我们可以找到较优解却无法找到最优解，我们可以通过计算熵值来对它进行信息优化。同时，我们也可以运用预剪枝来避免过度拟合，运用后剪枝来避免欠拟合。