1. 决策树生成是一个递归的过程，有三种递归返回条件

2. ID3算法采用信息增益进行属性划分，但是对取值数目较多的属性有偏好

3. C4.5采用信息增益率进行属性划分，但是对取值数目较少的属性有偏好。因此C4.5采用启发式，先用信息增益筛选出高于平均值的属性，然后再用增益率选最优

4. CART(二叉树)通过最小化基尼指数进行属性划分。基尼指数反映了从数据集D中随机抽取两个样本，其类别标记不一致的概率。基尼数越小，D的纯度越高

5. 决策树防止过拟合

    1. 预剪枝：决策树生成中，对每一个节点进行属性划分前，要进行"泛化能力提升"判断，如果划分后可以提升， 就划，不然直接把它标为叶子节点。会有欠拟合的风险
    2. 后剪枝：决策树生成完毕，自底向上考察每一个非叶子节点，进行"泛化能力提升'判断，若不通过，则把该节点为根的子树变成一个叶节点。欠/过拟合风险小，但是开销大
    3. 如何进行"泛化能力提升"判断，可以用验证集

6. 连续属性

    1. 连续属性，可以采用二分法，选取划分点（两个相邻值的平均），将所有样本分为两部分，
    2. n个样本，就有n-1个候选划分点
    3. 划分点的选取，可以通过信息增益等方式
    4. 连续属性可以多次用于属性划分！

7. 缺失值怎么处理

    1. 如何属性划分。为每个样本赋权重（初始化为1），求出某一属性a中无缺失值的加权比例，推广信息增益，在无缺失值的集合上求，之后乘上这个比例
    2. 确定了被划分的属性，怎么分配样本？
        1. 若样本有正常值，则的样本正常分配给对应叶节点。
        2. 若为缺失值，样本x同时划入所有子节点，但是样本权重按比例减小（比例为非缺失值中各取值的占比）

8. 多变量决策树（斜决策树）

    1. 单变量决策树在高维空间形成决策边界，是轴平行的。较好的可解释性。若学习目标复杂，也可以通过多段划分近似
    2. 每个非叶子节点不仅是针对某一属性，而是对属性的线性组合的测试，类似于一个个线性分类器，形成倾斜的边界

9. ID3 C4.5 CART比较

    1. ID3使用信息增益，对取值数目较多的属性有偏好，不能处理连续属性，不能处理缺失值，不剪枝
    2. C4.5使用信息增益率，对取值数目较少的属性有偏好，能处理连续属性，能处理缺失值，支持剪枝
    3. CART是二叉树，单变量分割（每次只选一个属性而不是多个），特征双化（因为二叉树，遍历所有组合），要剪枝，可以用于分类或者回归
        1. CART分类树，使用GINI指数
        2. CART回归树，和分类有两点不同：

    1. 特征选择与属性划分，不是根据基尼指数，而是均方差。即对于任意划分特征A，对应的任意划分点s两边划分成的数据集D1和D2，求出使D1和D2各自集合的均方差最小，同时D1和D2的均方差之和最小所对应的特征和特征值划分点。
    2. 预测过程，分类树直接选择叶节点频率最高的类别，回归树则是用叶子节点的中位数或者均值作为输出。
4. 样本的一点波动都会导致树结构剧变，可以引入随机森林