机器学习-第七章 决策树

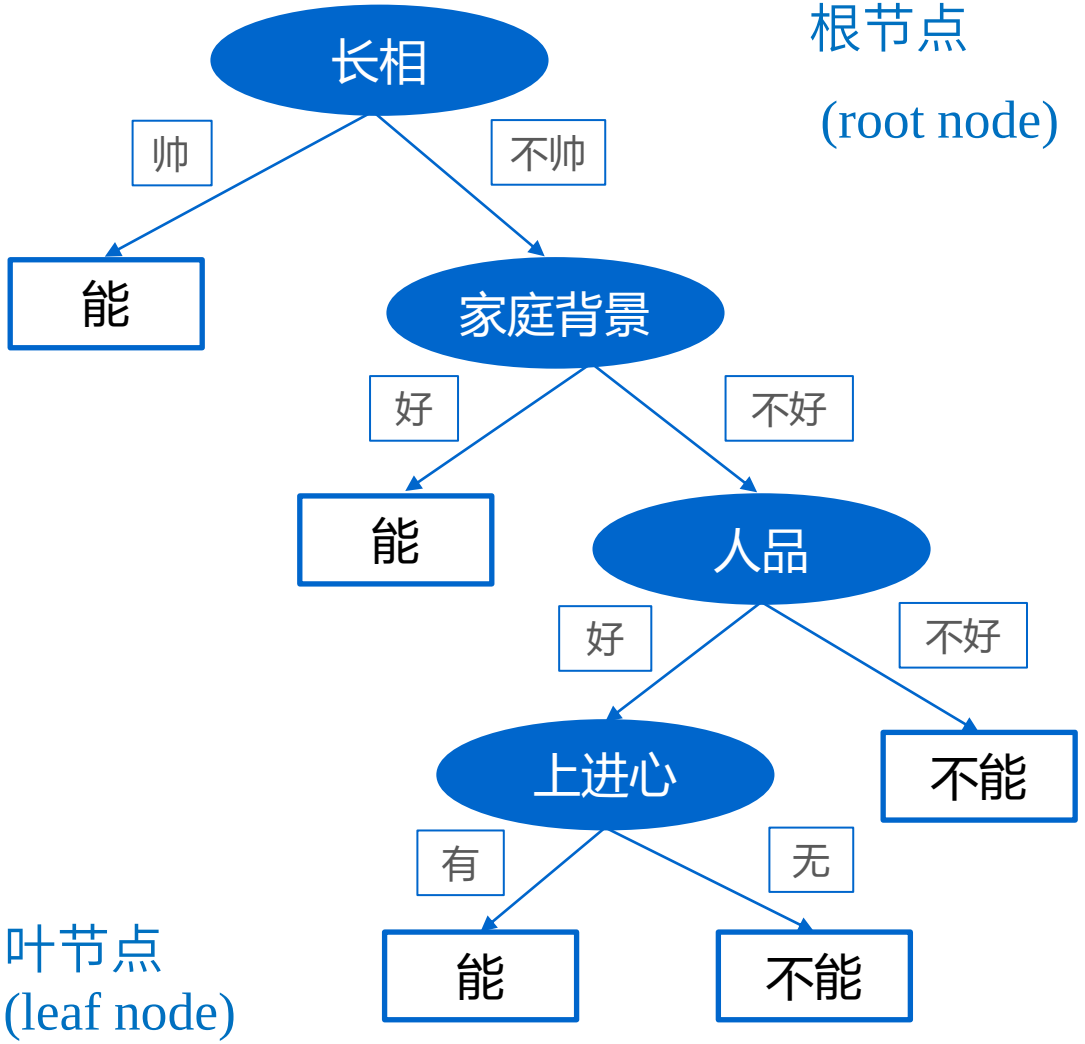
黄亮 副教授

2023年

本章目录

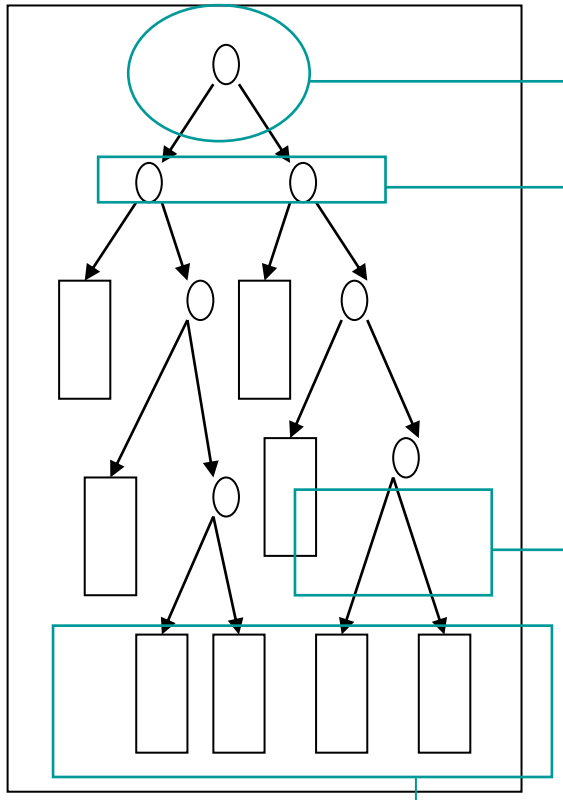
1.决策树原理

1.决策树原理



- 决策树：从训练数据中学习得出一个树状结构的模型。
- 决策树属于判别模型。
- 决策树是一种树状结构，通过做出一系列决策（选择）来对数据进行划分，这类似于针对一系列问题进行选择。
- 决策树的决策过程就是从根节点开始，测试待分类项中对应的特征属性，并按照其值选择输出分支，直到叶子节点，将叶子节点的存放的类别作为决策结果。

1.决策树原理



根节点 (root node)

非叶子节点 (non-leaf node)

(代表测试条件，对数据属性的测试)

分支 (branches) (代表测试结果)

叶节点 (leaf node)

(代表分类后所获得的分类标记)

- 决策树算法是一种归纳分类算法，它通过对训练集的学习，挖掘出有用的规则，用于对新数据进行预测。
- 决策树算法属于监督学习方法。
- 决策树归纳的基本算法是贪心算法，自顶向下来构建决策树。
- 贪心算法：在每一步选择中都采取在当前状态下最好/优的选择。
- 在决策树的生成过程中，分割方法即属性选择的度量是关键。

1.决策树原理

决策树的特点

优点：

- 推理过程容易理解，计算简单，可解释性强。
- 比较适合处理有缺失属性的样本。
- 可自动忽略目标变量没有贡献的属性变量，也为判断属性变量的重要性，减少变量的数目提供参考。

缺点：

- 容易造成过拟合，需要采用剪枝操作。
- 忽略了数据之间的相关性。
- 对于各类别样本数量不一致的数据，信息增益会偏向于那些更多数值的特征。

1.决策树原理

决策树的三种基本类型

建立决策树的关键，即在当前状态下选择哪个属性作为分类依据。根据不同的目标函数，建立决策树主要有一下三种算法： ID3(Iterative Dichotomiser)、C4.5、CART(Classification And Regression Tree)。

算法	支持模型	树结构	特征选择	连续值处理	缺失值处理	剪枝	特征属性多次使用
ID3	分类	多叉树	信息增益	不支持	不支持	不支持	不支持
C4.5	分类	多叉树	信息增益率	支持	支持	支持	不支持
CART	分类 回归	二叉树	基尼指数 均方差	支持	支持	支持	支持

2.ID3算法

2. ID3算法

ID3 算法

- ID3 算法最早是由罗斯昆（J. Ross Quinlan）于1975年提出的一种决策树构建算法，算法的核心是“信息熵”，期望信息越小，信息熵越大，样本纯度越低。。
- ID3 算法是以信息论为基础，以信息增益为衡量标准，从而实现对数据的归纳分类。
- ID3 算法计算每个属性的信息增益，并选取具有最高增益的属性作为给定的测试属性。

2.ID3算法

ID3 算法

其大致步骤为：

1. 初始化特征集合和数据集合；
2. 计算数据集合信息熵和所有特征的条件熵，选择信息增益最大的特征作为当前决策节点；
3. 更新数据集合和特征集合（删除上一步使用的特征，并按照特征值来划分不同分支的数据集合）；
4. 重复 2，3 两步，若子集值包含单一特征，则为分支叶子节点。

信息熵

信息熵 $$H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} log_2 \frac{|C_k|}{|D|}$$

K是类别，D是数据集，C_k是类别K下的数据集

右边类别数据：

数量	是	否	信息熵
15	9	6	0.971

$$H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} log_2 \frac{|C_k|}{|D|}$$

$$= -\frac{9}{15} log_2 \frac{9}{15} - \frac{6}{15} log_2 \frac{6}{15} = 0.971$$

	年龄	有工作	有房子	信用	类别
0	青年	否	否	一般	否
1	青年	否	否	好	否
2	青年	是	否	好	是
3	青年	是	是	一般	是
4	青年	否	否	一般	否
5	中年	否	否	一般	否
6	中年	否	否	好	否
7	中年	是	是	好	是
8	中年	否	是	非常好	是
9	中年	否	是	非常好	是
10	老年	否	是	非常好	是
11	老年	否	是	好	是
12	老年	是	否	好	是
13	老年	是	否	非常好	是
14	老年	否	否	一般	否

信息熵

按年龄划分

年龄	数量	是	否	信息熵
青年	5	2	3	0.9710
中年	5	3	2	0.9710
老年	5	4	1	0.7219

A_1	年龄
A_2	有工作
A_3	有房子
A_4	信用

$$H(D|A_1 = 青年) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.971$$

$$H(D|A_1 = 中年) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.971$$

$$H(D|A_1 = 老年) = -\frac{4}{5}log_2\frac{4}{5} - \frac{1}{5}log_2\frac{1}{5} = 0.7219$$

	年龄	有工作	有房子	信用	类别
0	青年	否	否	一般	否
1	青年	否	否	好	否
2	青年	是	否	好	是
3	青年	是	是	一般	是
4	青年	否	否	一般	否
5	中年	否	否	一般	否
6	中年	否	否	好	否
7	中年	是	是	好	是
8	中年	否	是	非常好	是
9	中年	否	是	非常好	是
10	老年	否	是	非常好	是
11	老年	否	是	好	是
12	老年	是	否	好	是
13	老年	是	否	非常好	是
14	老年	否	否	一般	否

条件熵

条件熵 $$H(D|A)=\sum_{i=1}^{n}\frac{|D_i|}{|D|}H(D_i)$$

A是特征，i是特征取值

$$H(D|\text{年龄})=\sum_{i=1}^{n}\frac{|D_i|}{|D|}H(D_i)$$

$$=\frac{5}{15}\times 0.971+\frac{5}{15}\times 0.971+\frac{5}{15}\times 0.7219$$

$$=0.8897$$

	年龄	有工作	有房子	信用	类别
0	青年	否	否	一般	否
1	青年	否	否	好	否
2	青年	是	否	好	是
3	青年	是	是	一般	是
4	青年	否	否	一般	否
5	中年	否	否	一般	否
6	中年	否	否	好	否
7	中年	是	是	好	是
8	中年	否	是	非常好	是
9	中年	否	是	非常好	是
10	老年	否	是	非常好	是
11	老年	否	是	好	是
12	老年	是	否	好	是
13	老年	是	否	非常好	是
14	老年	否	否	一般	否

信息增益

信息增益 $g(D,A) = H(D) - H(D|A)$

$$H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} log_2 \frac{|C_k|}{|D|} = 0.971$$

$$H(D|A_1) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} H(D_i) = 0.8897$$

$g(D,A_1) = H(D) - H(D|A_1)$ =0.971- 0.8897 = 0.0813

A_1	年龄
A_2	有工作
A_3	有房子
A_4	信用

	年龄	有工作	有房子	信用	类别
0	青年	否	否	一般	否
1	青年	否	否	好	否
2	青年	是	否	好	是
3	青年	是	是	一般	是
4	青年	否	否	一般	否
5	中年	否	否	一般	否
6	中年	否	否	好	否
7	中年	是	是	好	是
8	中年	否	是	非常好	是
9	中年	否	是	非常好	是
10	老年	否	是	非常好	是
11	老年	否	是	好	是
12	老年	是	否	好	是
13	老年	是	否	非常好	是
14	老年	否	否	一般	否

ID3算法

缺点

1. ID3 没有剪枝策略，容易过拟合；
2. 信息增益准则对**可取值数目较多的特征有所偏好**，类似“编号”的特征其信息增益接近于 1；
3. 只能用于处理离散分布的特征；
4. 没有考虑缺失值。

3.C4.5算法

3.C4.5算法

C4.5 算法

C4.5 算法是 Ross 对 ID3 算法的改进。

- 用信息增益率来选择属性。ID3选择属性用的是子树的信息增益，而C4.5用的是信息增益率。
- 在决策树构造过程中进行剪枝。
- 对非离散数据也能处理。
- 能够对不完整数据进行处理。

信息增益率

信息增益率 $g_R(D,A)=\dfrac{g(D,A)}{IV(A)}$

其中，$IV(A)=-\sum_{i=1}^{n}\dfrac{|D_i|}{|D|}log_2\dfrac{|D_i|}{|D|}$，$n$是特征$A$的取值个数

$g(D,A_1)=H(D)-H(D|A_1)$ =0.971- 0.8897 = 0.0813

$$IV(A_1)=-\sum_{i=1}^{3}\frac{|D_i|}{|D|}log_2\frac{|D_i|}{|D|}$$

$$=\frac{5}{15}\times 1.585+\frac{5}{15}\times 1.585+\frac{5}{15}\times 1.585=1.585$$

$$g_R(D,A_1)=\frac{g(D,A_1)}{IV(A_1)}=\frac{0.0813}{1.585}=0.0513$$

备注：信息增益 $g(D,A)=H(D)-H(D|A)$

	年龄	有工作	有房子	信用	类别
0	青年	否	否	一般	否
1	青年	否	否	好	否
2	青年	是	否	好	是
3	青年	是	是	一般	是
4	青年	否	否	一般	否
5	中年	否	否	一般	否
6	中年	否	否	好	否
7	中年	是	是	好	是
8	中年	否	是	非常好	是
9	中年	否	是	非常好	是
10	老年	否	是	非常好	是
11	老年	否	是	好	是
12	老年	是	否	好	是
13	老年	是	否	非常好	是
14	老年	否	否	一般	否

C4.5的剪枝

过拟合的原因：

- 为了尽可能正确分类训练样本，节点的划分过程会不断重复直到不能再分，这样就可能对训练样本学习的“太好”了，把训练样本的一些特点当做所有数据都具有的一般性质，从而导致过拟合。

通过剪枝处理去掉一些分支来降低过拟合的风险。

- 剪枝的基本策略有“预剪枝”（prepruning）和“后剪枝”（post-pruning）

C4.5的剪枝

预剪枝（prepruning）

预剪枝不仅可以降低过拟合的风险而且还可以减少训练时间，但另一方面它是基于“贪心”策略，会带来欠拟合风险。

训练集

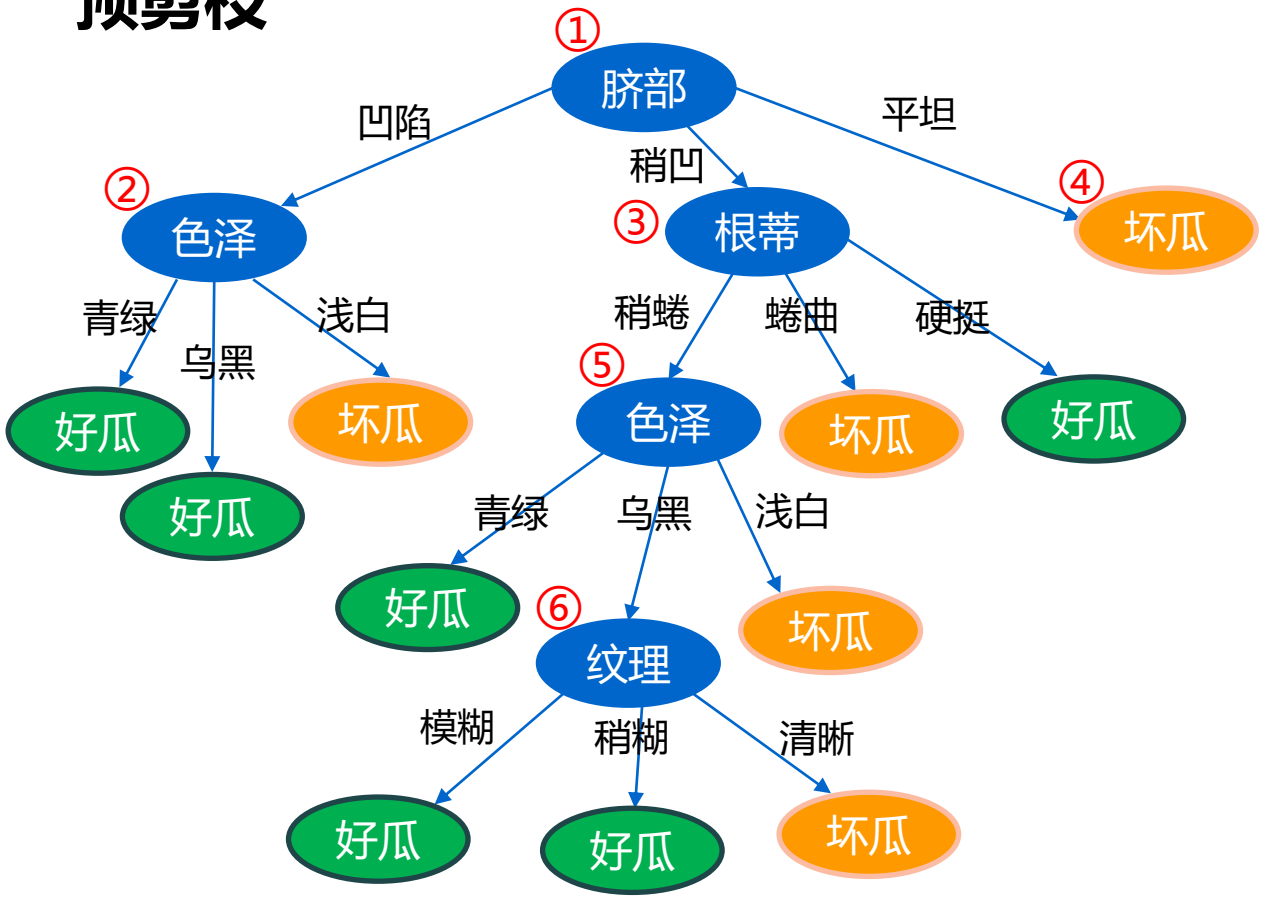
编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

验证集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否

C4.5的剪枝

预剪枝



基于表生成未剪枝的决策树

剪枝策略

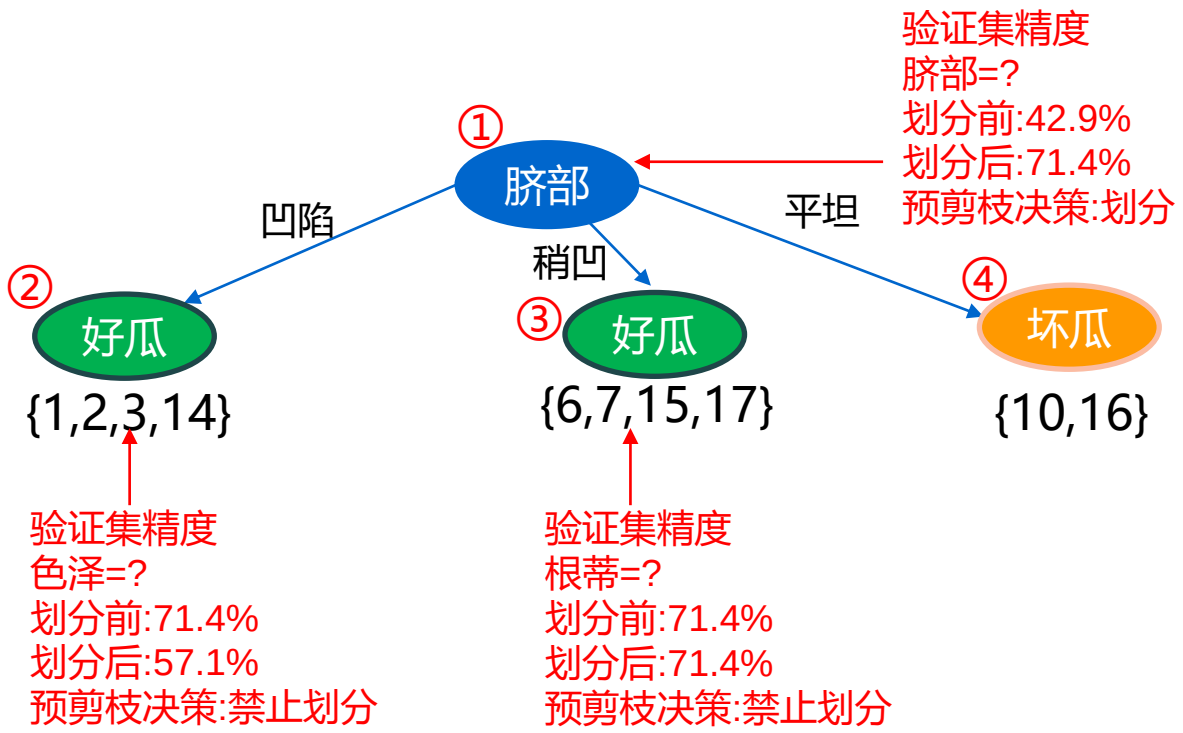
在节点划分前来确定是否继续增长，及早停止增长

主要方法有：

- 节点内数据样本低于某一阈值；
- 所有节点特征都已分裂；
- 节点划分前准确率比划分后准确率高。

C4.5的剪枝

预剪枝



预剪枝的决策树

剪枝策略

在节点划分前来确定是否继续增长，及早停止增长

主要方法有：

- 节点内数据样本低于某一阈值；
- 所有节点特征都已分裂；
- 节点划分前准确率比划分后准确率高。

C4.5的剪枝

后剪枝

- 在已经生成的决策树上进行剪枝，从而得到简化版的剪枝决策树。

- 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，后剪枝的欠拟合风险更小，泛化性能往往优于预剪枝决策树。

训练集

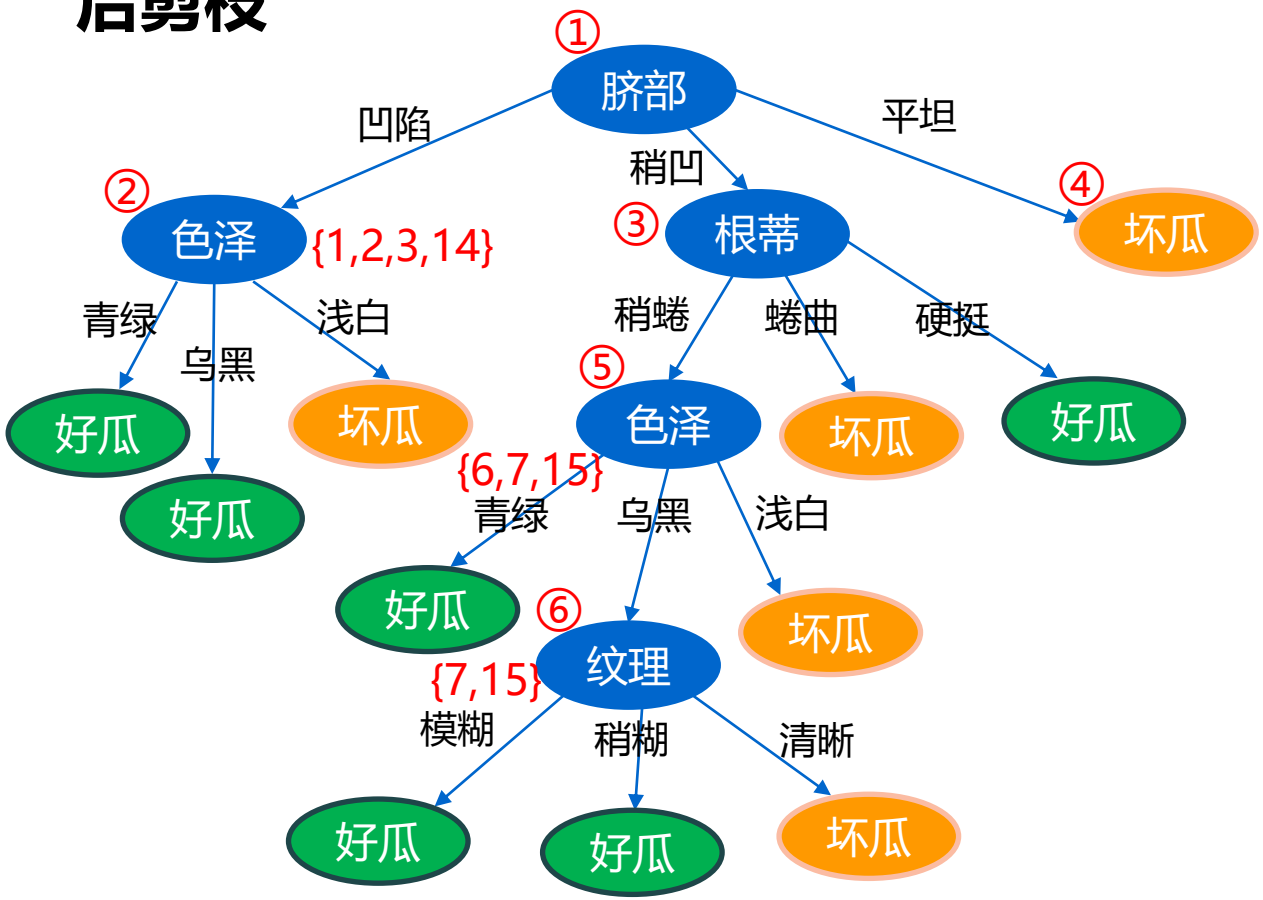
编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

验证集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否

C4.5的剪枝

后剪枝



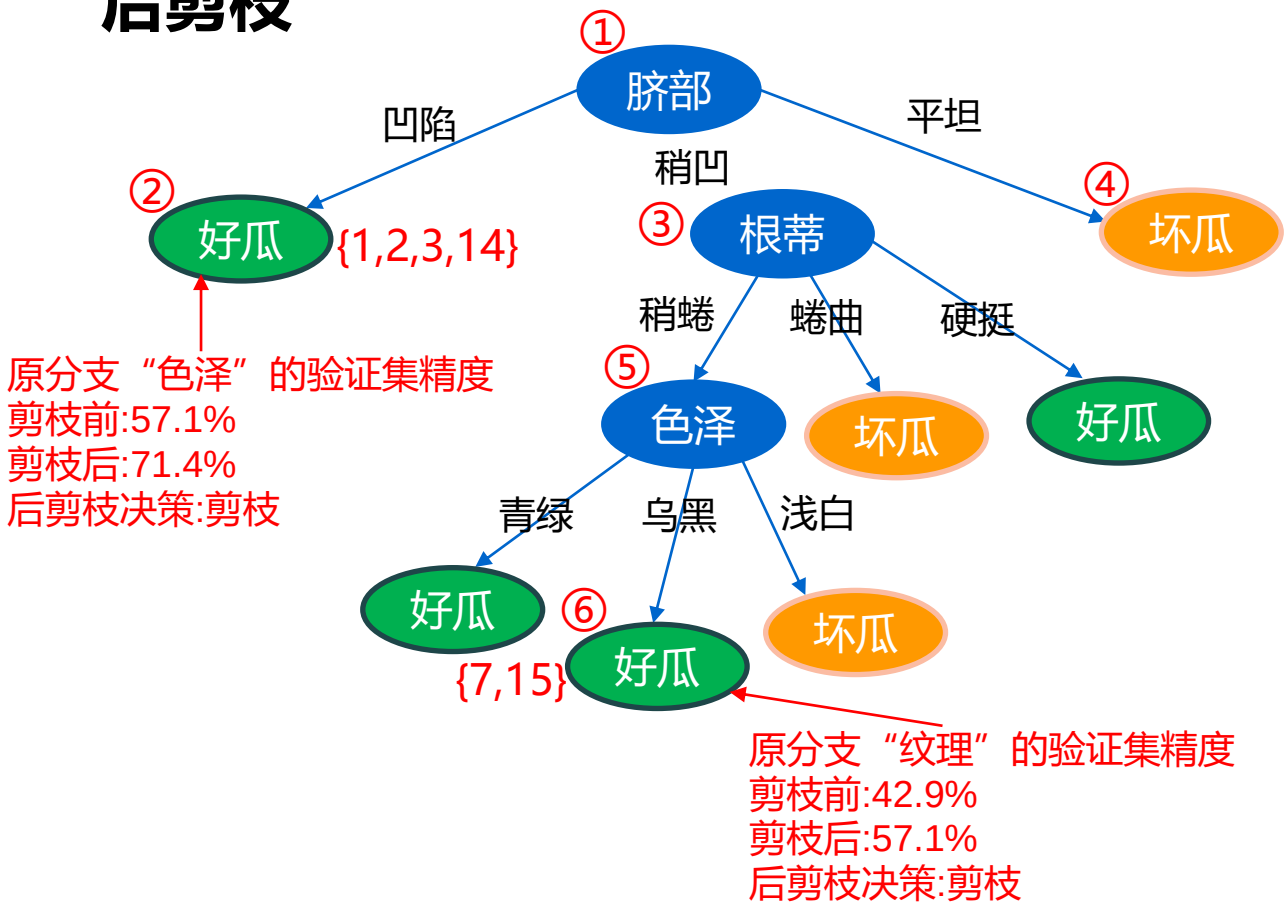
基于表生成未剪枝的决策树

剪枝方法

- 在已经生成的决策树上进行剪枝，从而得到简化版的剪枝决策树。
- C4.5 采用的悲观剪枝方法，用递归的方式从低往上针对每一个非叶子节点，评估用一个最佳叶子节点去代替这课子树是否有益。如果剪枝后与剪枝前相比其错误率是保持或者下降，则这棵子树就可以被替换掉。C4.5 通过训练数据集上的错误分类数量来估算未知样本上的错误率。
- 后剪枝决策树的欠拟合风险很小，泛化性能往往优于预剪枝决策树。

C4.5的剪枝

后剪枝



后剪枝的决策树

剪枝方法

- 在已经生成的决策树上进行剪枝，从而得到简化版的剪枝决策树。
- C4.5 采用的悲观剪枝方法，用递归的方式从低往上针对每一个非叶子节点，评估用一个最佳叶子节点去代替这课子树是否有益。如果剪枝后与剪枝前相比其错误率是保持或者下降，则这棵子树就可以被替换掉。C4.5 通过训练数据集上的错误分类数量来估算未知样本上的错误率。
- 后剪枝决策树的欠拟合风险很小，泛化性能往往优于预剪枝决策树。

C4.5的缺点

缺点

- 剪枝策略可以再优化；
- C4.5 用的是多叉树，用二叉树效率更高；
- C4.5 只能用于分类；
- C4.5 使用的熵模型拥有大量耗时的对数运算，连续值还有排序运算；
- C4.5 在构造树的过程中，对数值属性值需要按照其大小进行排序，从中选择一个分割点，所以只适合于能够驻留于内存的数据集，当训练集大得无法在内存容纳时，程序无法运行。

4.CART算法

01 决策树原理

02 ID3算法

03 C4.5算法

04 CART算法

4.CART算法

CART

- Classification and Regression Tree (CART) 是决策树的一种。
- 用基尼指数来选择属性（分类），或用均方差来选择属性（回归）。
- 顾名思义，CART算法既可以用于创建分类树，也可以用于创建回归树，两者在构建的过程中稍有差异。
- 如果目标变量是离散的，称为分类树。
- 如果目标变量是连续的，称为回归树。

CART算法-分类

基尼指数　　分类时用基尼指数来选择属性

$Gini(D,A)$表示经过A =a分割后集合D的不确定性。

$$Gini(D,A)=\frac{|D_1|}{|D|}Gini(D_1)+\frac{|D_2|}{|D|}Gini(D_2)$$

$$Gini(p)=\sum_{k=1}^{K}p_k(1-p_k)=1-\sum_{k=1}^{K}p_k^2$$

回顾信息熵：

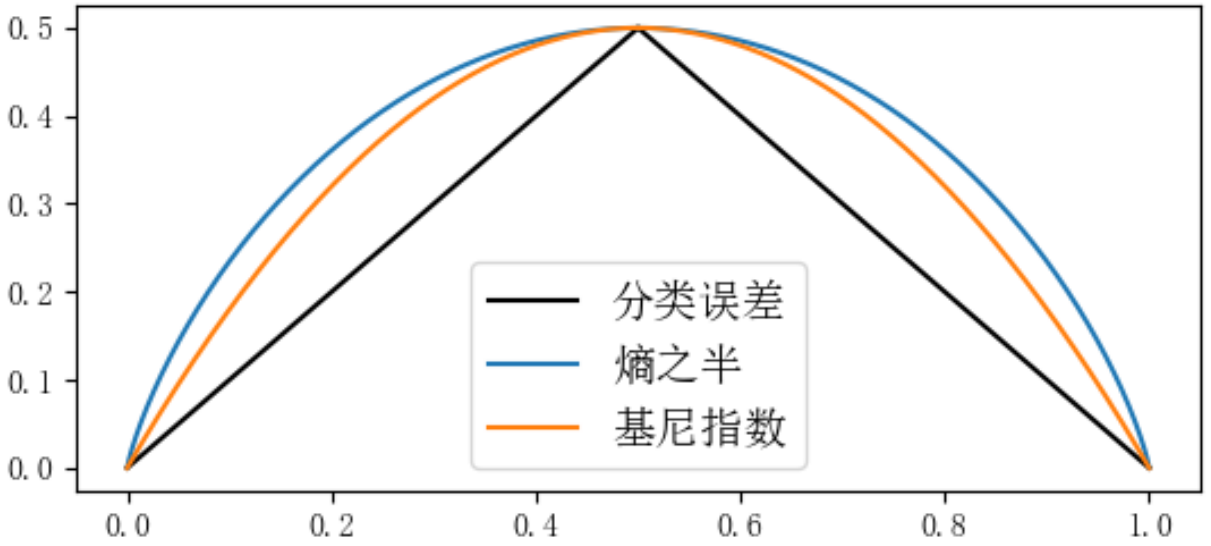
$$H(D)=-\sum_{k=1}^{K}\frac{|C_k|}{|D|}log_2\frac{|C_k|}{|D|}$$

在$p_k=1$处泰勒一阶展开：

$$\ln p_k \approx \ln 1+\frac{1}{1}(p_k-1)=p_k-1$$

分类误差：

$$error(p)=\begin{cases}p, & p<0.5\\ 1-p, & p\geq 0.5\end{cases}$$



CART算法-分类

基尼指数　　分类时用基尼指数来选择属性

$Gini(D,A)$表示经过A =a分割后集合D的不确定性。

$$Gini(D,A)=\frac{|D_1|}{|D|}Gini(D_1)+\frac{|D_2|}{|D|}Gini(D_2)$$

$$Gini(p)=\sum_{k=1}^{K}p_k(1-p_k)$$

$$Gini(D,A_1=\text{青年})=\frac{5}{15}\times\left(2\times\frac{2}{5}\times\left(1-\frac{2}{5}\right)\right)+\frac{10}{15}\times\left(2\times\frac{7}{10}\times\left(1-\frac{7}{10}\right)\right)=0.44$$

$Gini(D,A_1=\text{中年})=0.48$

$Gini(D,A_1=\text{老年})=0.44$

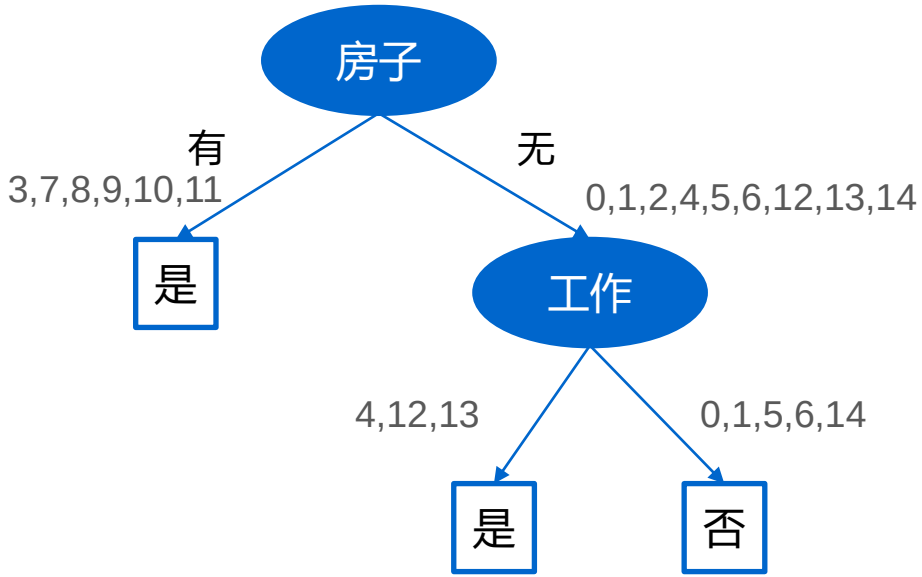
$Gini(D,A_2=\text{是})=0.32$

$Gini(D,A_3=\text{是})=0.27$

$Gini(D,A_4=\text{非常好})=0.36$

$Gini(D,A_4=\text{好})=0.47$

$Gini(D,A_4=\text{一般})=0.32$



	年龄	有工作	有房子	信用	类别
0	青年	否	否	一般	否
1	青年	否	否	好	否
2	青年	是	否	好	是
3	青年	是	是	一般	是
4	青年	否	否	一般	否
5	中年	否	否	一般	否
6	中年	否	否	好	否
7	中年	是	是	好	是
8	中年	否	是	非常好	是
9	中年	否	是	非常好	是
10	老年	否	是	非常好	是
11	老年	否	是	好	是
12	老年	是	否	好	是
13	老年	是	否	非常好	是
14	老年	否	否	一般	否

CART算法-回归

用均方差来选择属性

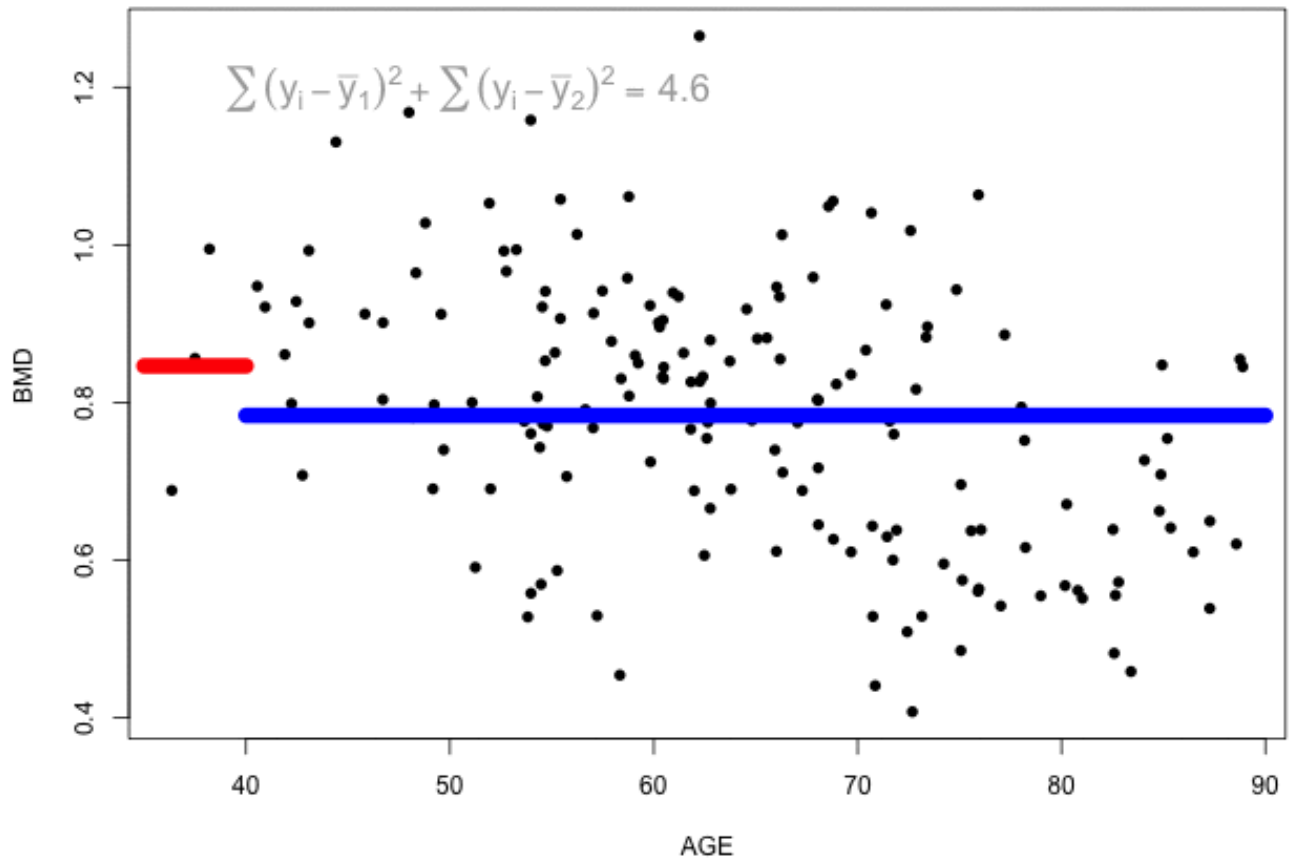
对于连续值的处理，CART 分类树采用均方差的大小来度量特征的各个划分点。对于任意划分特征 A，对应的任意划分点s 两边划分成的数据集 D_1和D_2 ，求出使 D_1和D_2各自集合的均方差最小，同时 D_1和D_2的均方差之和最小所对应的特征和特征值划分点。表达式为：

$$\min_{a,s}[\min_{c_1}\sum_{x_i\in D_1}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in D_2}(y_i-c_2)^2]$$

其中，c_1为D_1数据集的样本输出均值，c_2为D_2 数据集的样本输出均值。

CART算法-回归

用均方差来选择属性 $\min_{a,s}[\min_{c_1}\sum_{x_i\in D_1}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in D_2}(y_i-c_2)^2]$



CART算法-回归

预测方式

对于决策树建立后做预测的方式，上面讲到了 CART 分类树采用叶子节点里概率最大的类别作为当前节点的预测类别。

而回归树输出不是类别，它采用的是用最终叶子的均值或者中位数来预测输出结果。

CART剪枝

CART算法采用一种“基于代价复杂度的剪枝”方法进行**后剪枝**，这种方法会生成一系列树，每个树都是通过将前面的树的某个或某些子树替换成一个叶节点而得到的，这一系列树中的最后一棵树仅含一个用来预测类别的叶节点。然后用一种成本复杂度的度量准则来判断哪棵子树应该被一个预测类别值的叶节点所代替。

这种方法需要使用一个单独的测试数据集来评估所有的树，根据它们在测试数据集熵的分类性能选出最佳的树。

CART算法

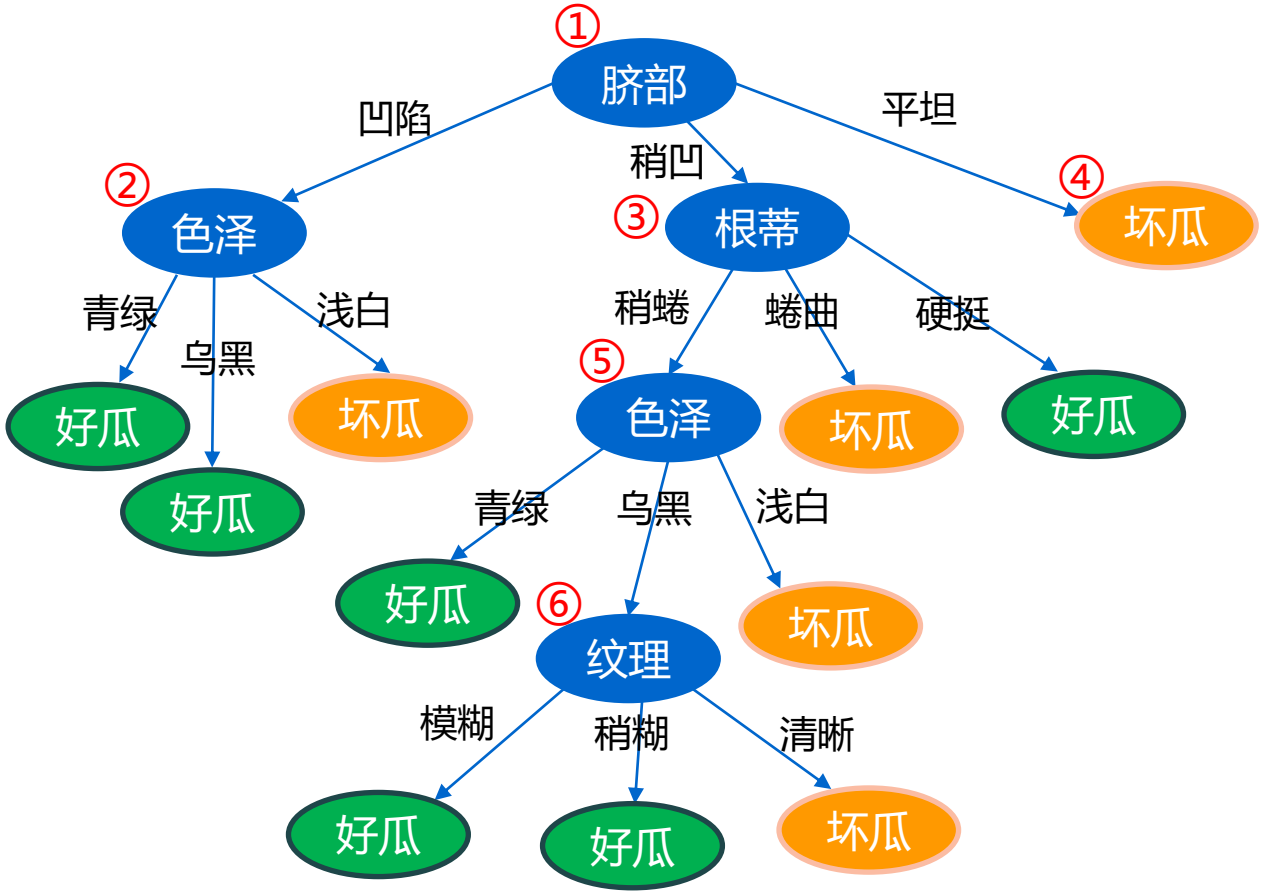
CART剪枝

具体流程:

(1)计算每一个结点的条件熵

(2)递归的从叶子节点开始往上遍历,减掉叶子节点,然后判断损失函数的值是否减少,如果减少,则将父节点作为新的叶子节点

(3)重复(2),直到完全不能剪枝.



决策树差异总结

- **划分标准的差异**：ID3 使用信息增益偏向特征值多的特征，C4.5 使用信息增益率克服信息增益的缺点，偏向于特征值小的特征，CART 使用基尼指数克服 C4.5 需要求 log 的巨大计算量，偏向于特征值较多的特征。
- **使用场景的差异**：ID3 和 C4.5 都只能用于分类问题，CART 可以用于分类和回归问题；ID3 和 C4.5 是多叉树，速度较慢，CART 是二叉树，计算速度很快；
- **样本数据的差异**：ID3 只能处理离散数据且缺失值敏感，C4.5 和 CART 可以处理连续性数据且有多种方式处理缺失值；从样本量考虑的话，小样本建议 C4.5、大样本建议 CART。C4.5 处理过程中需对数据集进行多次扫描排序，处理成本耗时较高，而 CART 本身是一种大样本的统计方法，小样本处理下泛化误差较大；
- **样本特征的差异**：ID3 和 C4.5 层级之间只使用一次特征，CART 可多次重复使用特征；
- **剪枝策略的差异**：ID3 没有剪枝策略，C4.5 是通过悲观剪枝策略来修正树的准确性，而 CART 是通过代价复杂度剪枝。

参考文献

1. 《统计学习方法》，清华大学出版社，李航著，2019年出版
2. 《机器学习》，清华大学出版社，周志华著，2016年出版
3. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag, 2006
4. 《人工智能基础》，中国大学慕课，彭涛


谢 谢!