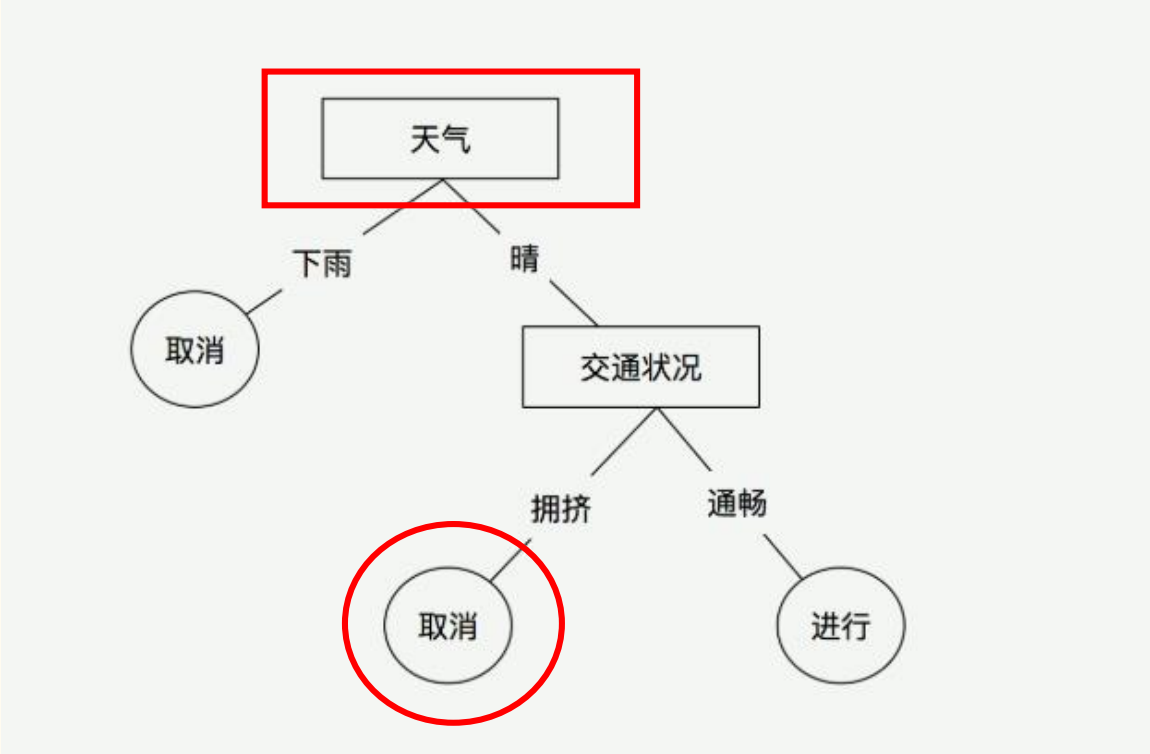
决策树原理

同学帮/视觉系：https://space.bilibili.com/202603446

1 决策树是什么?

决策树：是由一个个”决策”所组成的树，放”决策依据”的是非叶节点，放”决策结果”的是叶节点。



同学帮/视觉系：https://space.bilibili.com/202603446

ID3算法

熵：熵是描述信息的不确定度的，是随机变量不确定度的度量。熵越大，信息的不确定度越大，信息越”混乱”，越不符合决策树分类的需求。

$$\mathrm{Ent}(D)=-\sum_{k=1}^{|y|} p_k \log_2 p_k$$

pk: 第k类样本所占的比例,(k=1,2,3…|y|)

D:样本集合

2 经典的ID3算法

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
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
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
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
10	青绿	硬挺	清脆	清晰	平坦	软粘	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
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

2 经典的ID3算法

信息增益：衡量熵的变化，即在选定特征A后，数据不确定度的下降。信息增益越大，意味着这个特征的分类的能力越强，则优先选择这个特征。

$$\mathrm{Gain(D,a)} = \mathrm{Ent(D)} - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent(D^v)$$

$\mathrm{D^v}$：属性a中取值为$\mathrm{a^v}$的集合

ID3算法缺陷

（1）不支持连续特征

（2）采用信息增益大的特征优先建立决策树的节点。在相同条件下，取值比较多的特征比取值少的特征信息增益大。

（3）不支持缺失值处理

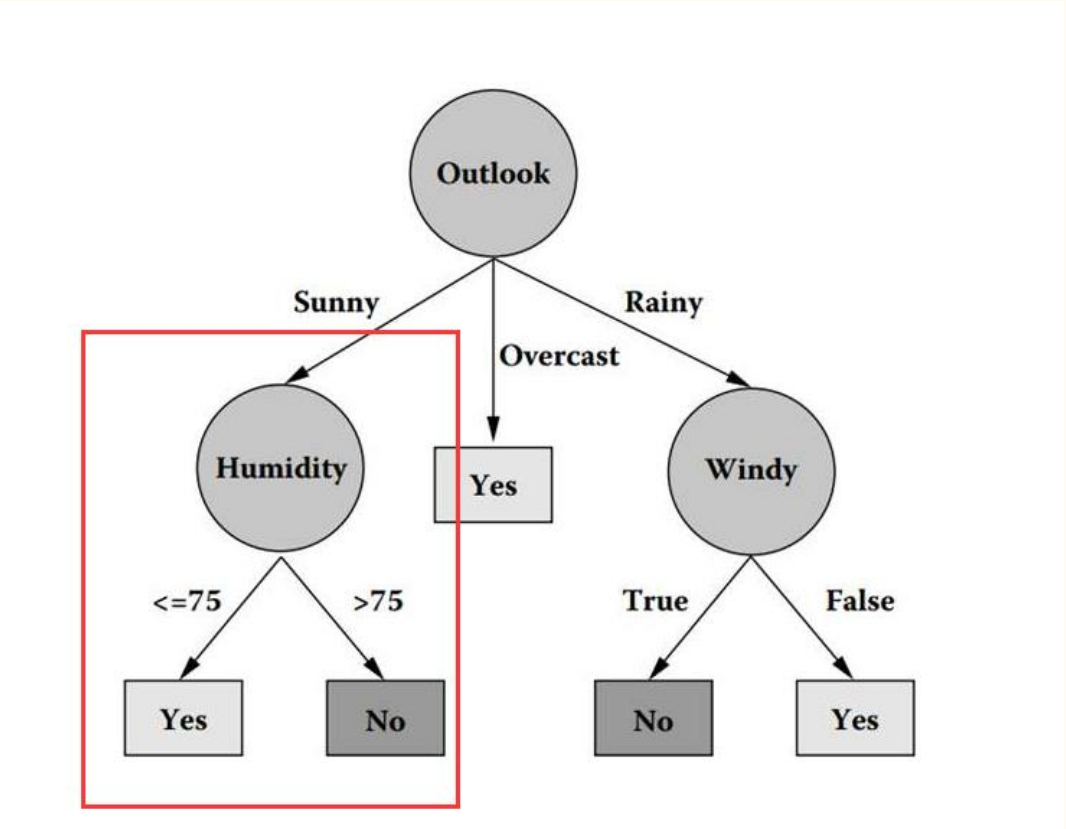
（4）没有应对过拟合的策略

C4.5算法

连续特征：C4.5的思路是将连续的特征离散化。

$$\{a^1, a^2 \ldots\ldots a^n\}$$

$$\mathrm{T}_a = \{\frac{a^i + a^{i+1}}{2} | 1 \le i \le n-1\}$$



3 改进的C4.5算法

信息增益比：针对信息增益偏向于取值比较多的特点而提出

$$\text{Gain\_ratio(D,a)}=\frac{Gain(D,a)}{IV(a)}$$

$$Iv(a)=-\sum_{v=1}^{V}\frac{|D^v|}{|D|}\log_2\frac{|D^v|}{|D|}$$

其中IV(a)称为属性a的固有值

3 改进的C4.5算法

缺失值处理

1.如何在属性值缺失的情况下进行划分属性选择

2.在给定划分属性的情况下，若样本在该属性上缺失，如何对样本进行

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

3 改进的C4.5算法

$$\rho = \frac{\sum_{x \epsilon D^{\sim}} \omega_{\mathrm{x}}}{\sum_{x \epsilon D} \omega_{\mathrm{x}}}$$

$$\rho_k^{\sim} = \frac{\sum_{x \epsilon D_k^{\sim}} \omega_{\mathrm{x}}}{\sum_{x \epsilon D^{\sim}} \omega_{\mathrm{x}}}$$

$$\gamma_k^{\sim} = \frac{\sum_{x \epsilon D_{\mathrm{v}}^{\sim}} \omega_{\mathrm{x}}}{\sum_{x \epsilon D^{\sim}} \omega_{\mathrm{x}}}$$

$$Gain(D, a) = \rho \times Gain(D^{\sim}, a)$$
$$= \rho \times (Ent(D^{\sim}) - \sum_{v=1}^{V} \gamma_v^{\sim} Ent(D_V^{\sim}))$$

$$Ent(D^{\sim}) = -\sum_{k=1}^{|y|} \rho_k^{\sim} \log_2 \rho_k^{\sim}$$

同学帮/视觉系：https://space.bilibili.com/202603446

C4.5算法缺陷

（1）剪枝的算法有非常多，C4.5的剪枝方法有优化的空间

（2）C4.5生成的是多叉树，很多时候，在计算机中二叉树模型会比多叉树运算效率高。如果采用二叉树，可以提高效率

（3）C4.5只能用于分类

（4）C4.5使用了熵模型，里面有大量的耗时的对数运算,如果是连续值还有大量的排序运算

CART算法

CART假设决策树是二叉树，并且可以分类也可以回归，而且用基尼系数代替了熵模型进行特征选择，也提供了优化的剪枝策略

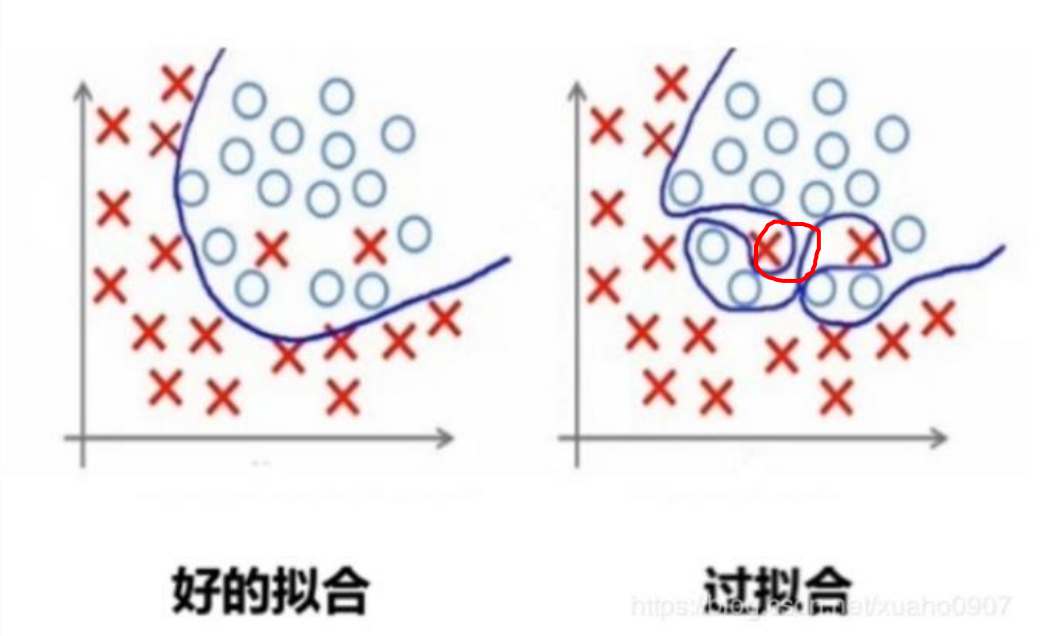
$$\text{Gini(D)}=\sum_{k=1}^{|y|}\sum_{k'!=k}p_k p_{k'}=1-\sum_{k=1}^{|y|}p_k^2$$

$$\text{Gini\_index(D,a)}=\sum_{v=1}^{V}\frac{|D^v|}{|D|}Gini(D^v)$$

$$a_*=argmin\ Gini_{index(D,a)}, a\in \text{A}$$

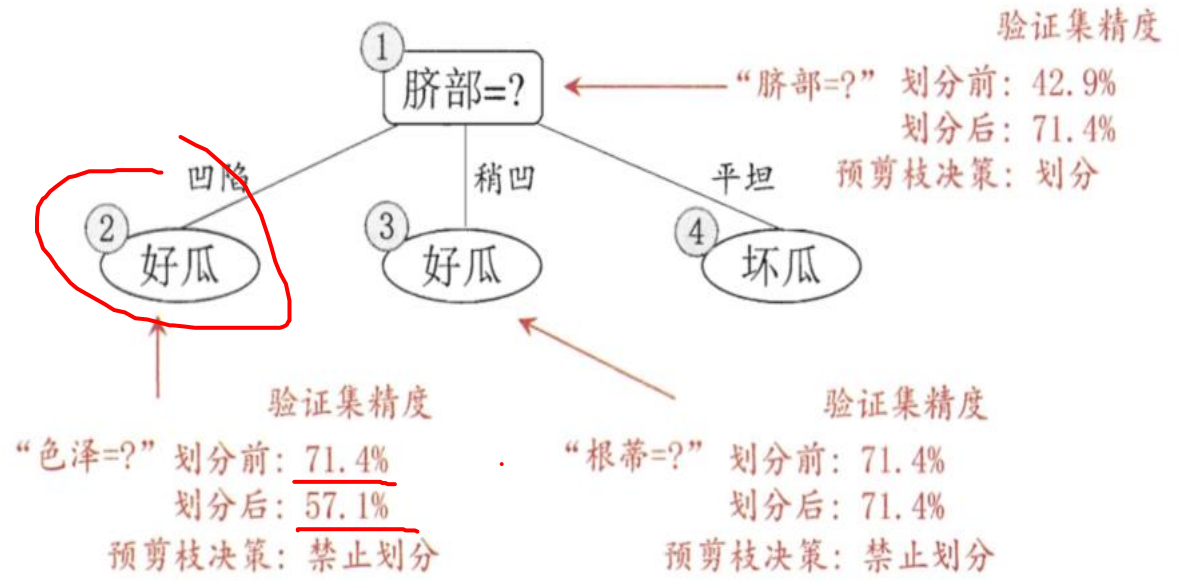
5 决策树的泛化能力

过拟合问题：所建立的决策树模型在训练样本中表现得过于优越，导致在验证数据集以及测试数据集中表现不佳



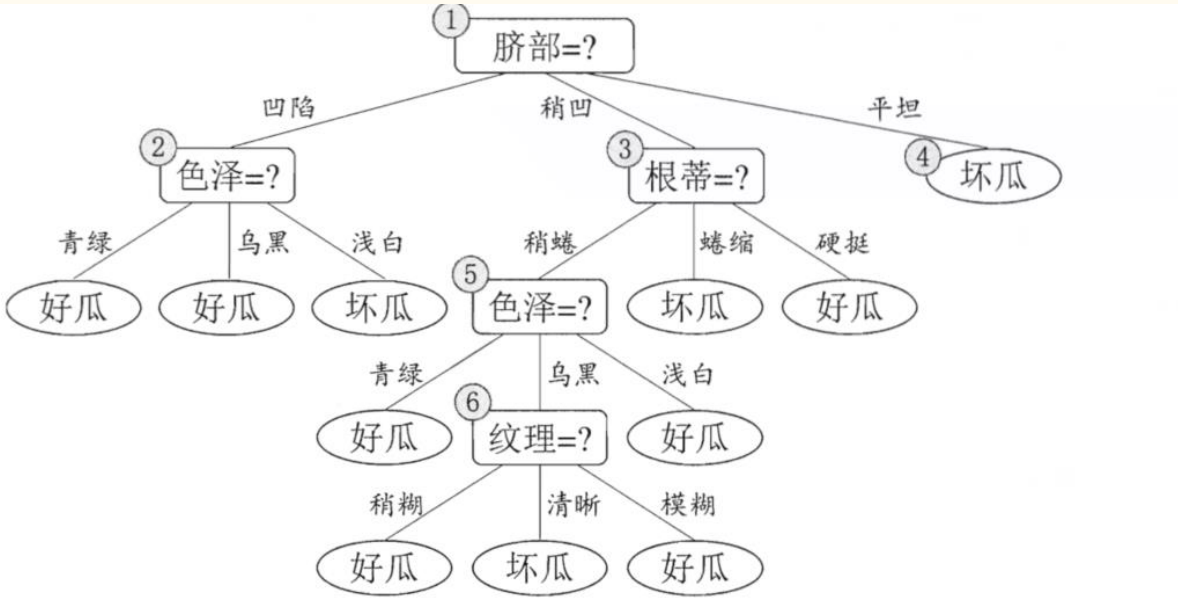
5 决策树的泛化能力

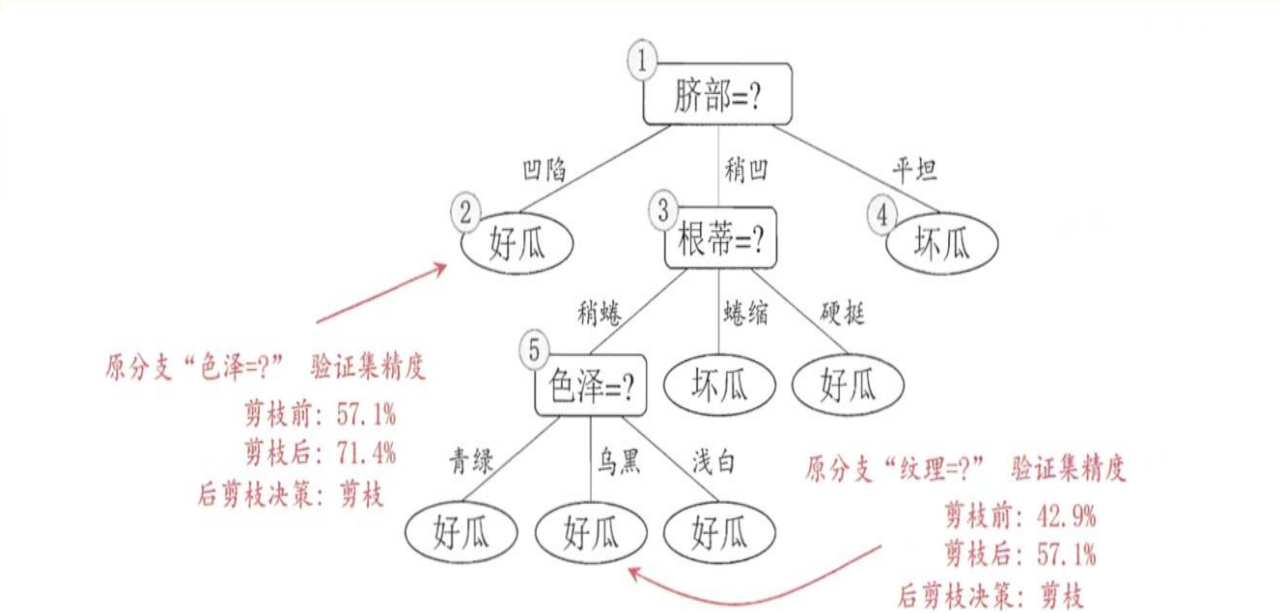
预剪枝



5 决策树的泛化能力

后剪枝





同学帮/视觉系：https://space.bilibili.com/202603446