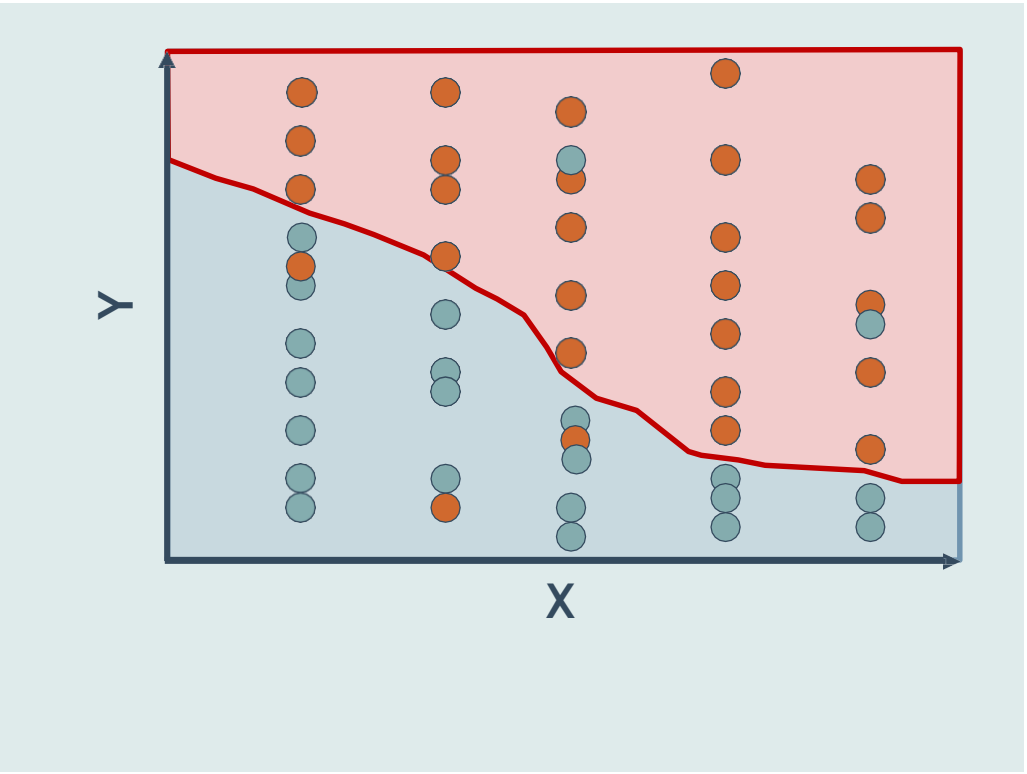
# 第11章 决策树

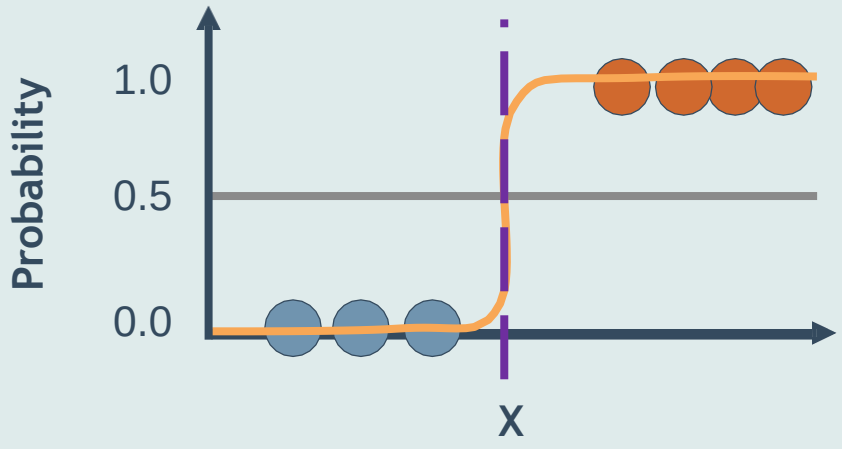
# 不同分类器的特点




**K近邻：**

- 模型就是训练数据
- 拟合训练数据很快
  - 只是存储数据
- 预测比较慢
  - 需要计算大量的距离
- 判定边界较灵活

# 不同分类器的特点




$$y_\beta(x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x+\varepsilon)}}$$

逻辑回归：

- 模型就是参数
- 拟合训练数据可能较慢
  - 必须找到最优参数
- 预测较快
  - 计算期望值
- 判定边界较简单，缺乏灵活性

# 决策树介绍

**J. Ross Quinlan**
**罗斯.昆兰**
**1943-**

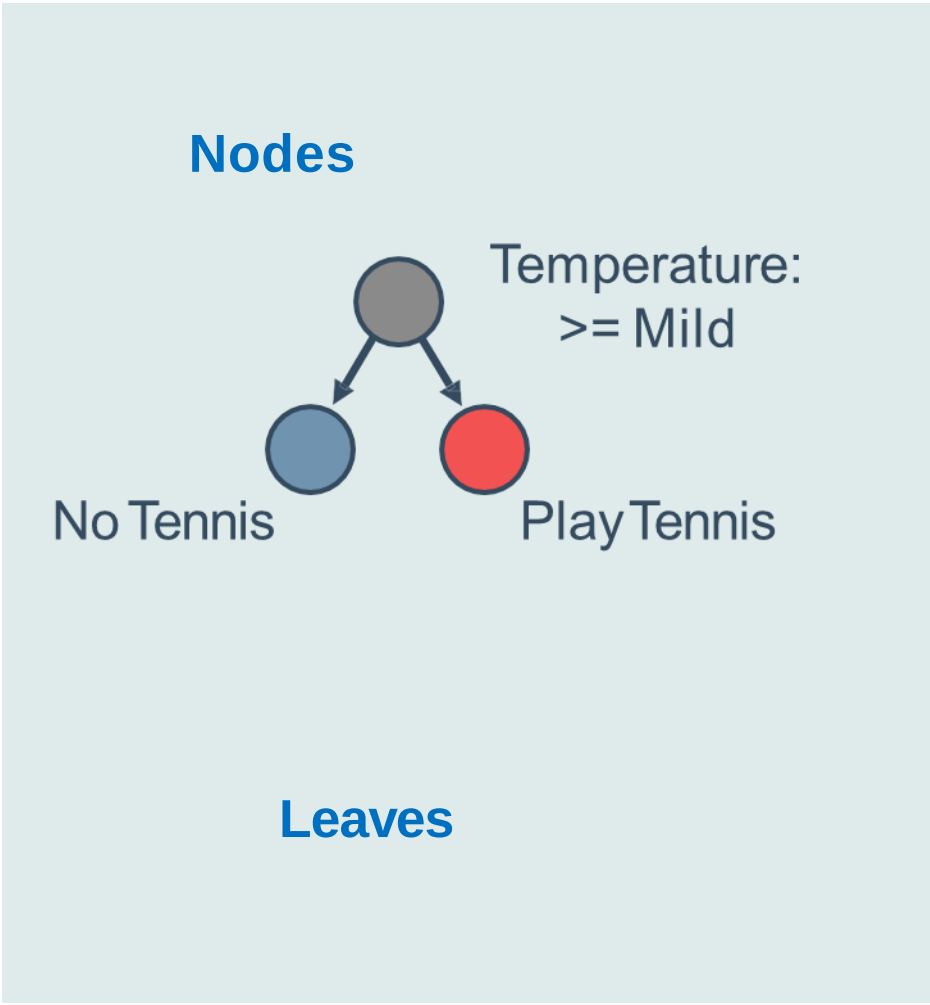
**决策树（decision tree）算法**起源于E.B.Hunt等人于1966年发表的论文“experiments in Induction”，但真正让决策树成为机器学习主流算法的还是Quinlan（罗斯.昆兰）（2011年获得了数据挖掘领域最高奖KDD创新奖），昆兰在1979年提出了ID3算法，掀起了决策树研究的高潮。现在最常用的决策树算法是C4.5是昆兰在1993年提出的。现在有了商业应用新版本是C5.0。

# 决策树介绍

| Day | Outlook | Temperature | Humidity | Wind | PlayTennis |
|---|---|---|---|---|---|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | Normal | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

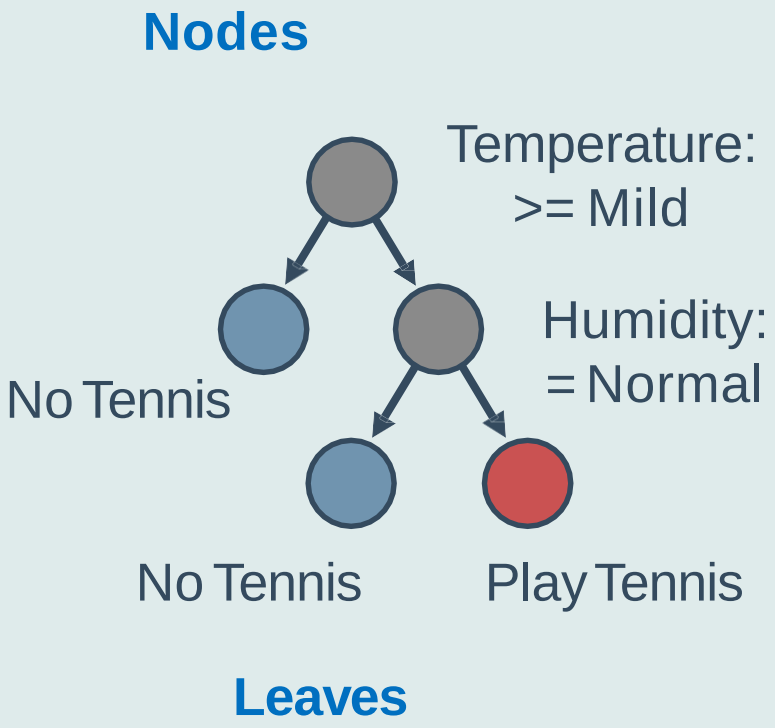
# 决策树介绍

- 想要根据temperature, humidity, wind, outlook来预测是否打网球
- 使用特征来划分数据，进而预测结果



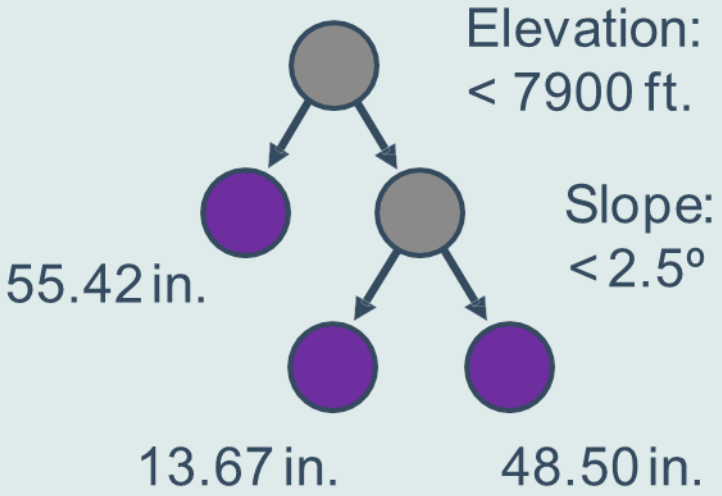
# 决策树介绍

- 想要根据temperature, humidity, wind, outlook来预测是否打网球
- 使用特征来划分数据，进而预测结果
- 预测类别结果的决策树



# 预测连续值的回归树

- 例如：使用喜马拉雅山脉的坡度和高度
- 预测平均降水量（连续值）
- 叶子节点的值是其所有成员的平均值

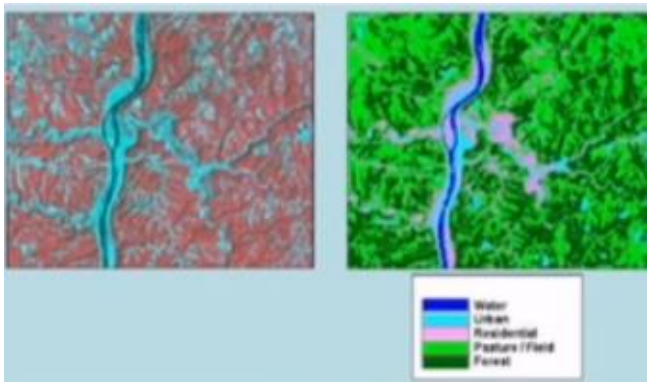


# 决策树应用举例


金融

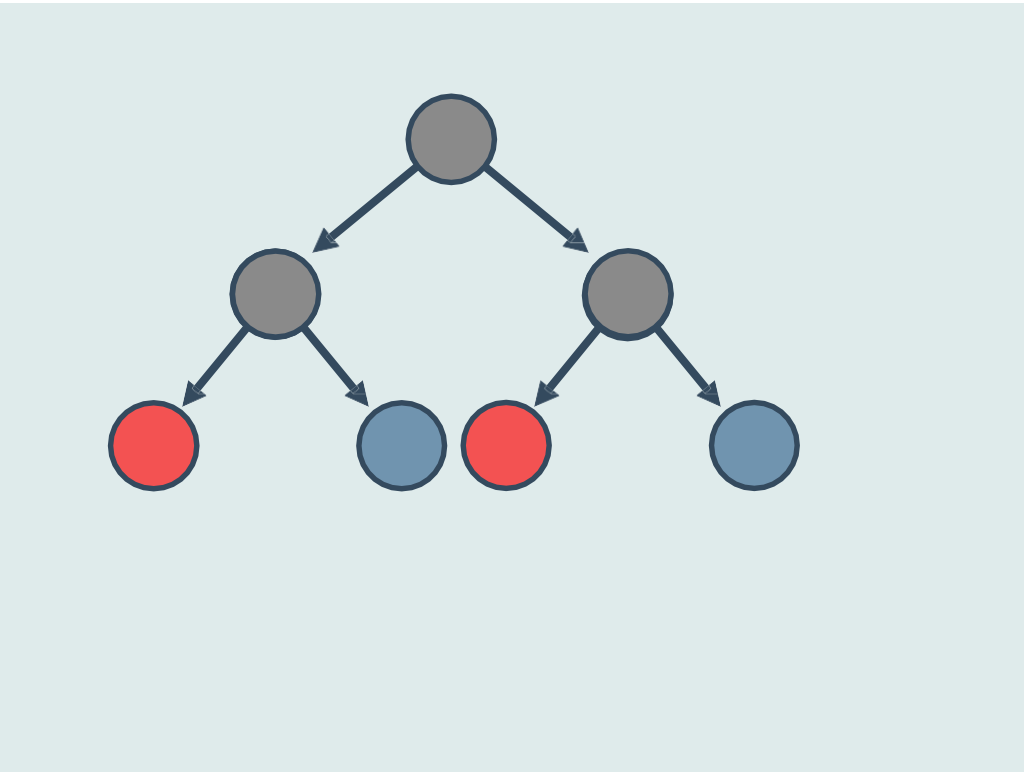

保险


医疗



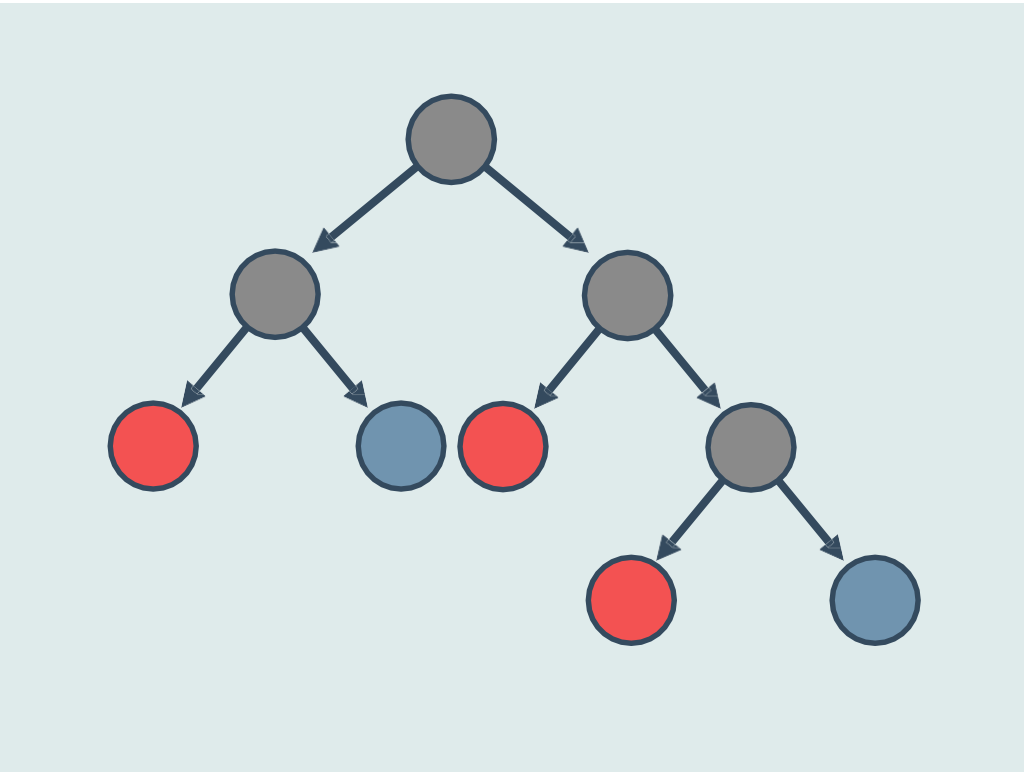
遥感

# 创建一个决策树



- 选取一个特征，把数据分成两部分，构成一个二叉树
- 继续选取特征，划分数据

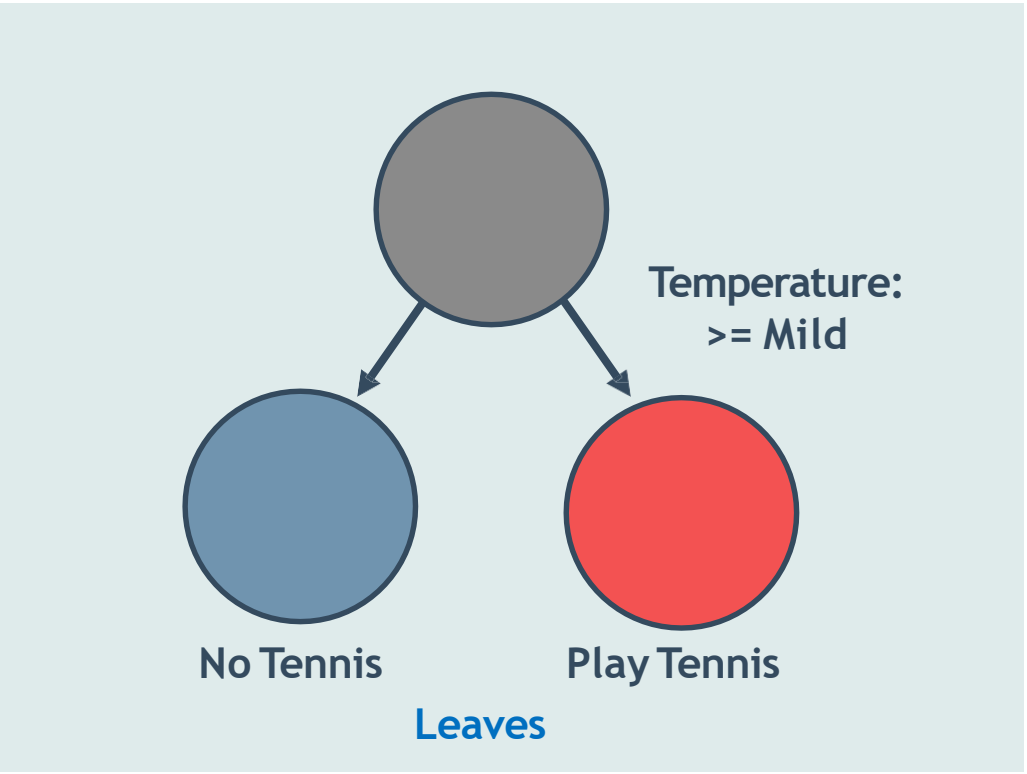
# 何时停止分裂



直到：

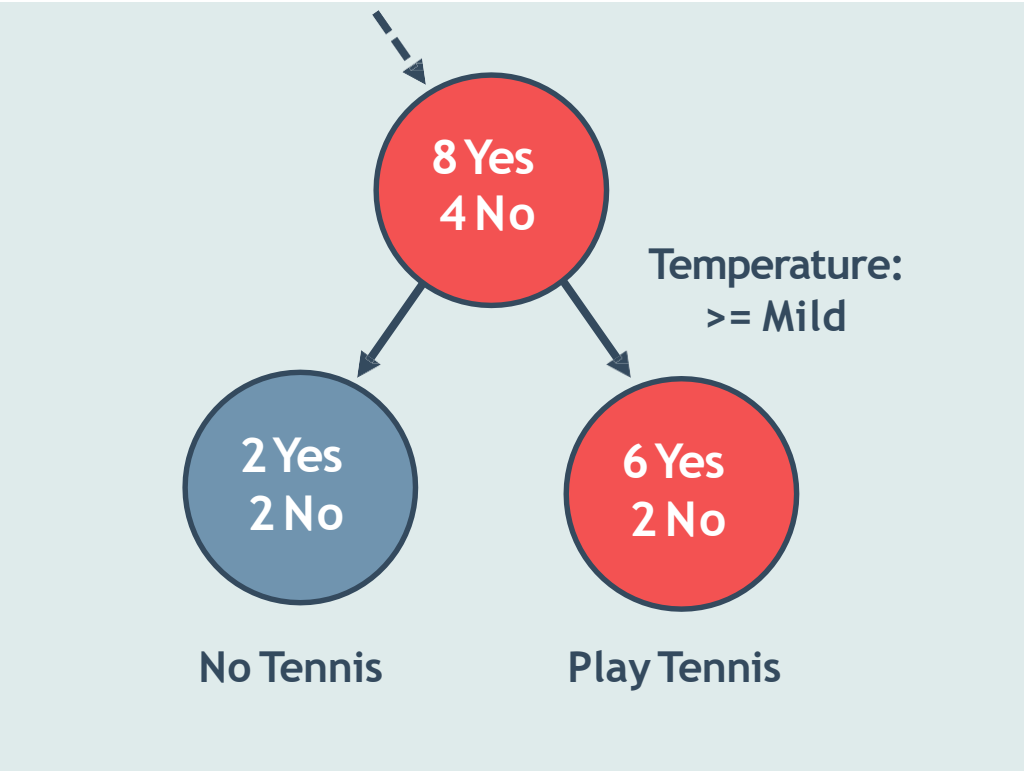
- 叶子节点纯了——仅包含一类实例
- 达到最大深度
- 达到某一性能指标

# 创建最优决策树




- 使用贪婪搜索：每一步寻找最优划分
- 什么是最优划分？
- 最大化不纯度减小量的划分
- 如何度量不纯度？

# 基于分类错误的划分



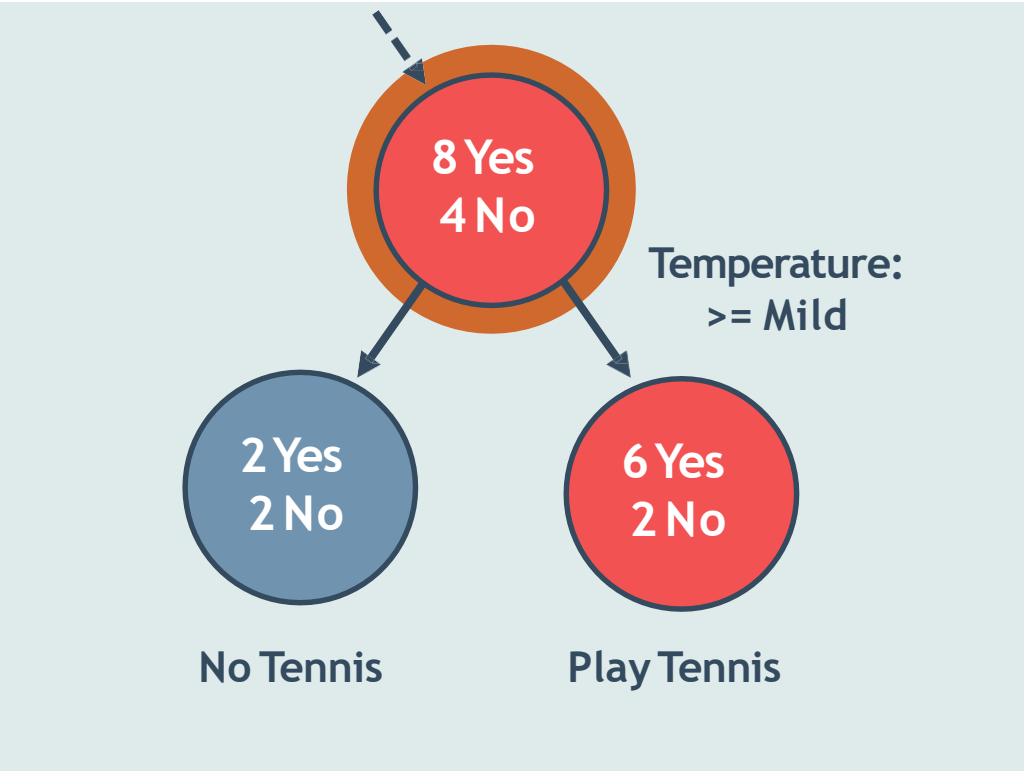

分类错误公式：

$$E(t) = 1 - \max_i [p(i|t)]$$

节点

类别出现的概率

# 基于分类错误的划分



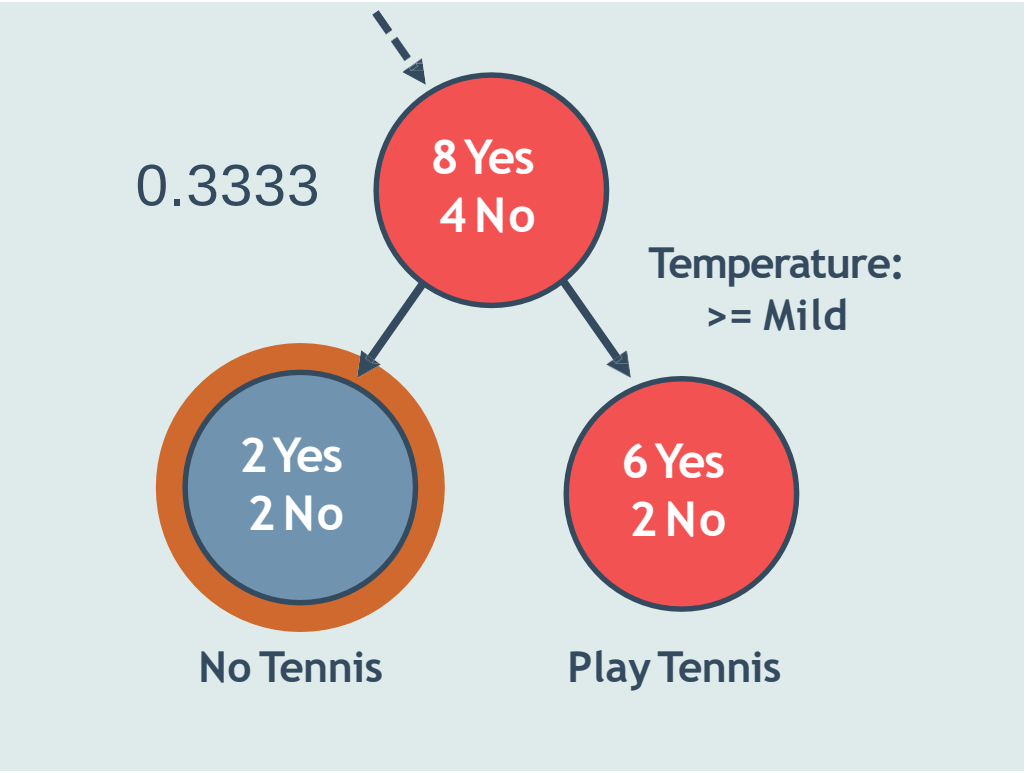

分类错误公式：

$$E(t) = 1 - \max_i [p(i|t)]$$

划分前的分类错误：

1 – 8/12 = 0.3333

# 基于分类错误的划分



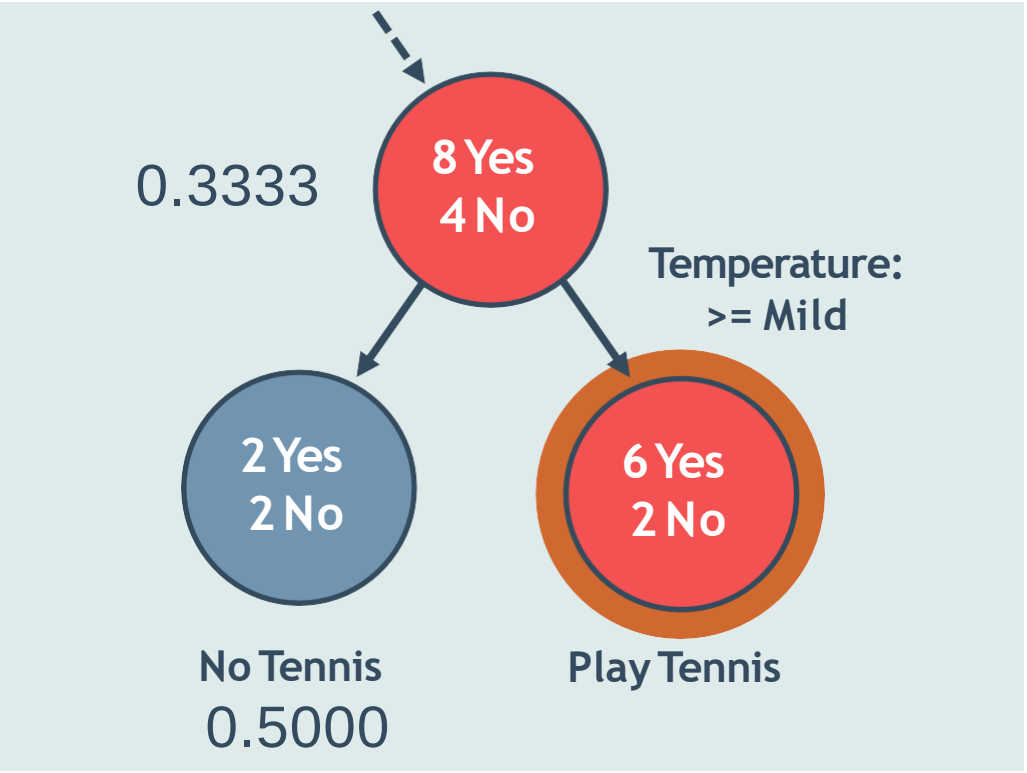

分类错误公式：

$$E(t) = 1 - \max_i [p(i|t)]$$

划分后左边的分类错误：

1 – 2/4 = 0.5000

# 基于分类错误的划分



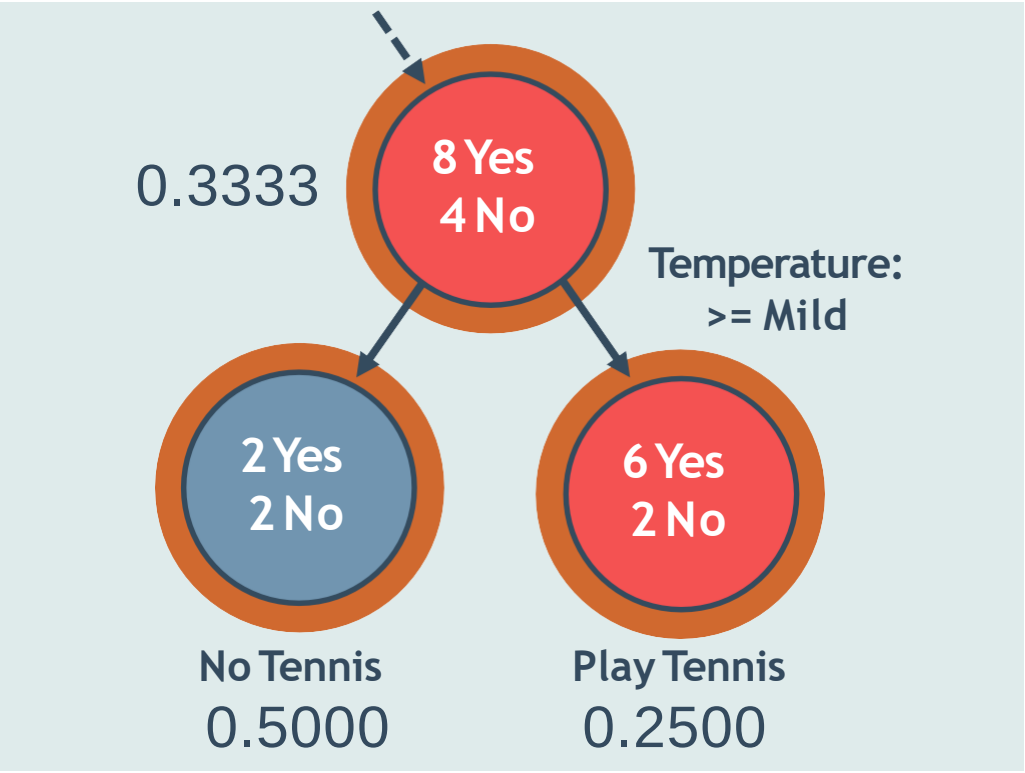

分类错误公式：

$$E(t) = 1 - \max_i [p(i|t)]$$

划分后右边的分类错误：

1 – 6/8 = 0.2500

# 基于分类错误的划分




分类错误公式：

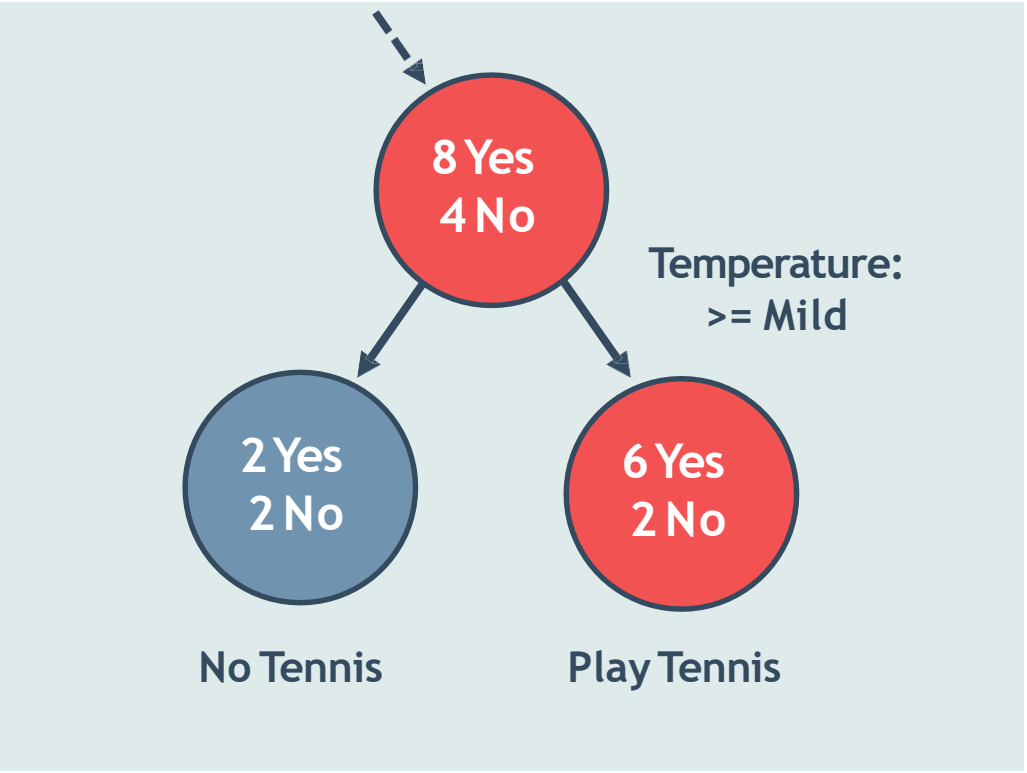
$$E(t) = 1 - \max_i [p(i|t)]$$

分类错误的变化：

$0.3333 - 4/12 * 0.5000 - 8/12 * 0.2500$

$= 0$

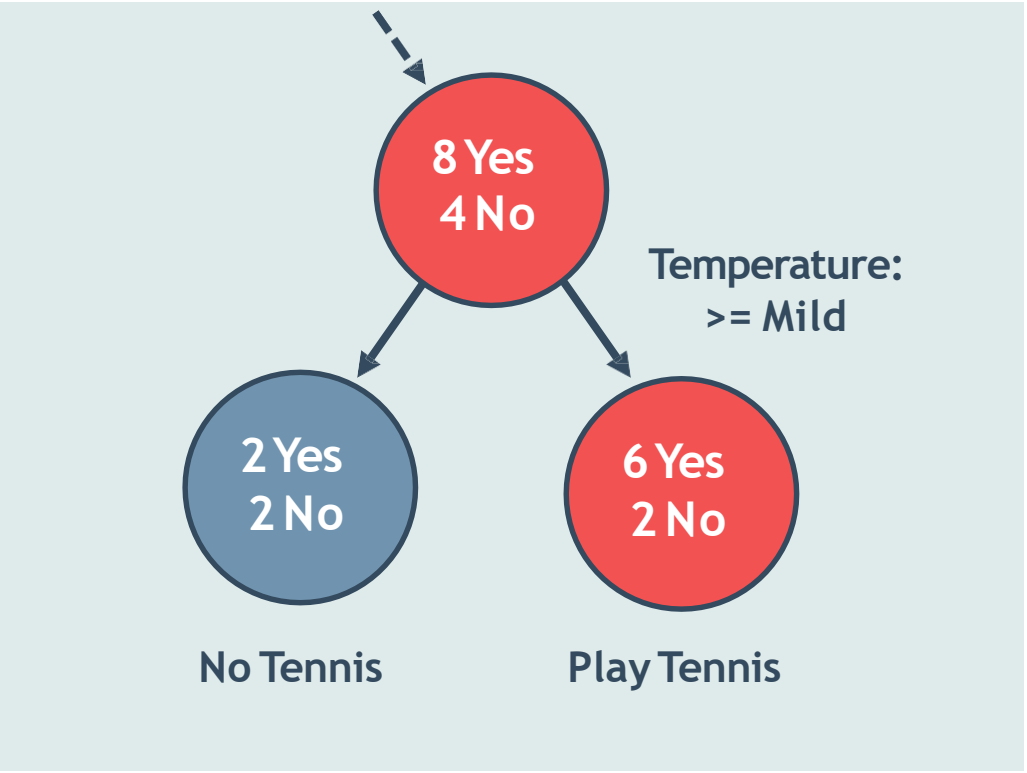
# 基于分类错误的划分




- 使用分类错误，分裂停止
- 问题：叶子节点仍然不是同质的
- 尝试另外一个性能指标？

# 基于熵的划分

信息熵：度量样本集合纯度最常用的一种指标

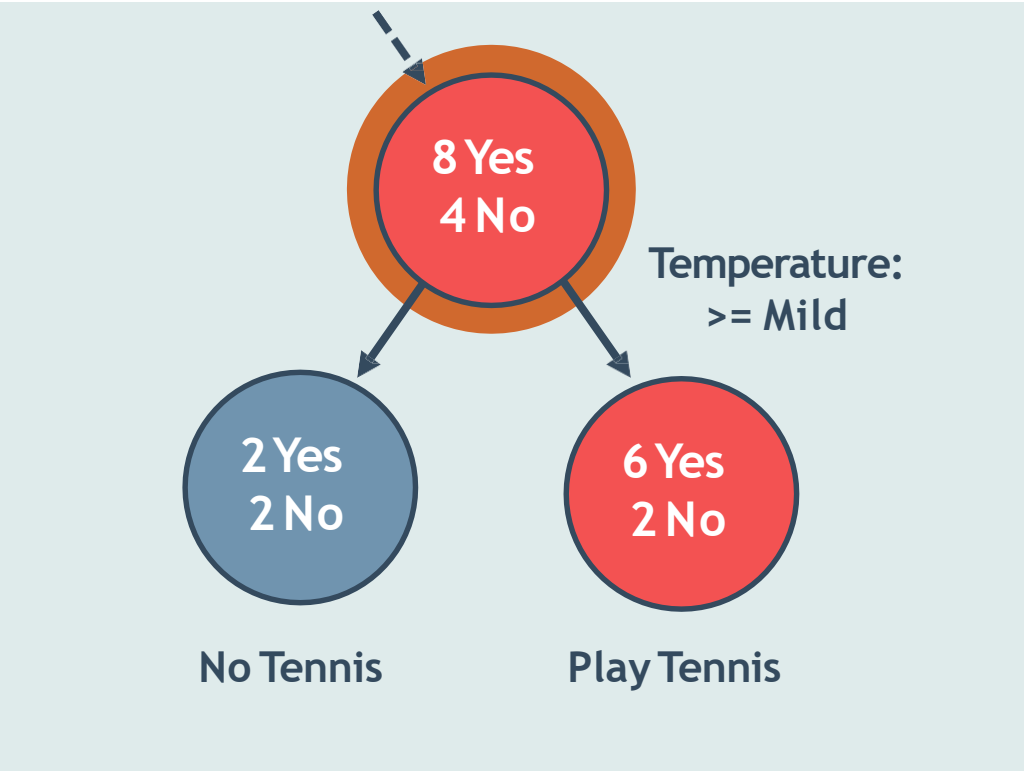



熵的公式：

$$H(t) = -\sum_{i=1}^{n} p(i|t) log_2[p(i|t)]$$

Ent($t$)或Ent($D$)，
$D$为对应的数据集

# 基于熵的划分



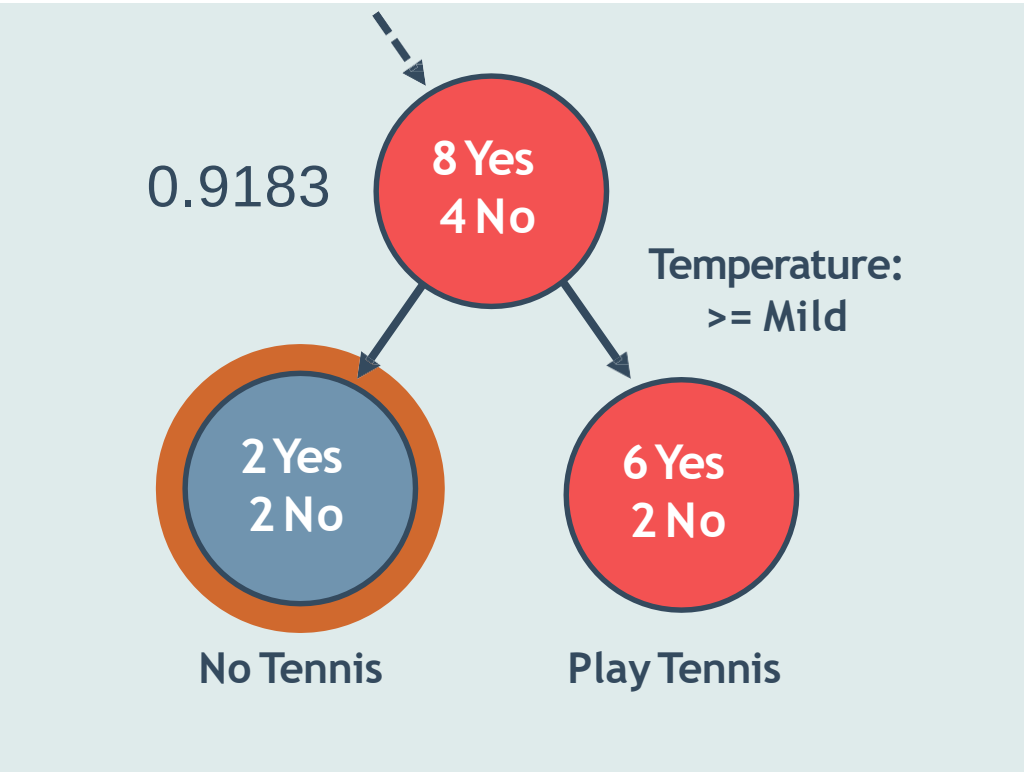

熵的公式:

$$H(t) = -\sum_{i=1}^{n} p(i|t) log_2[p(i|t)]$$

划分前的熵:

$-8/12*log_2(8/12)-4/12*log_2(4/12)$
$= 0.9183$

# 基于熵的划分




熵的公式:

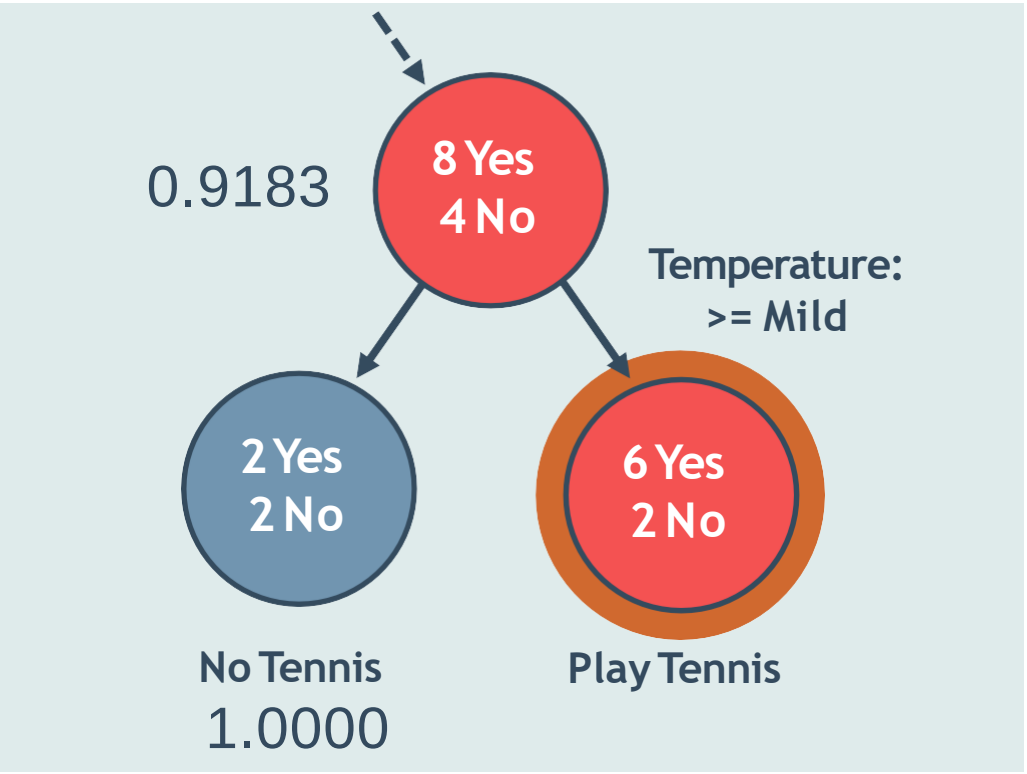
$$H(t) = -\sum_{i=1}^{n} p(i|t) log_2[p(i|t)]$$

划分后左边的熵:

$-2/4*log_2(2/4)-2/4*log_2(2/4)$

= 1.0000

# 基于熵的划分




熵的公式:

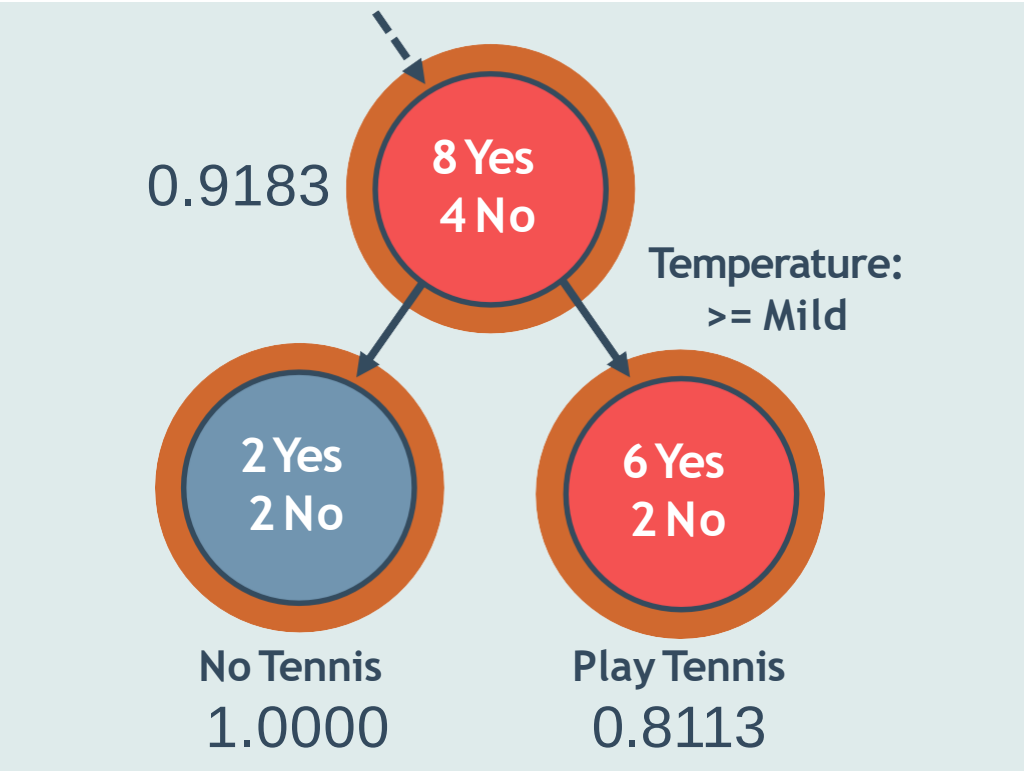
$$H(t) = -\sum_{i=1}^{n} p(i|t) log_2[p(i|t)]$$

划分后右边的熵:

$-6/8 * log_2(6/8) - 2/8 * log_2(2/8)$

= 0.8113

# 基于熵的划分



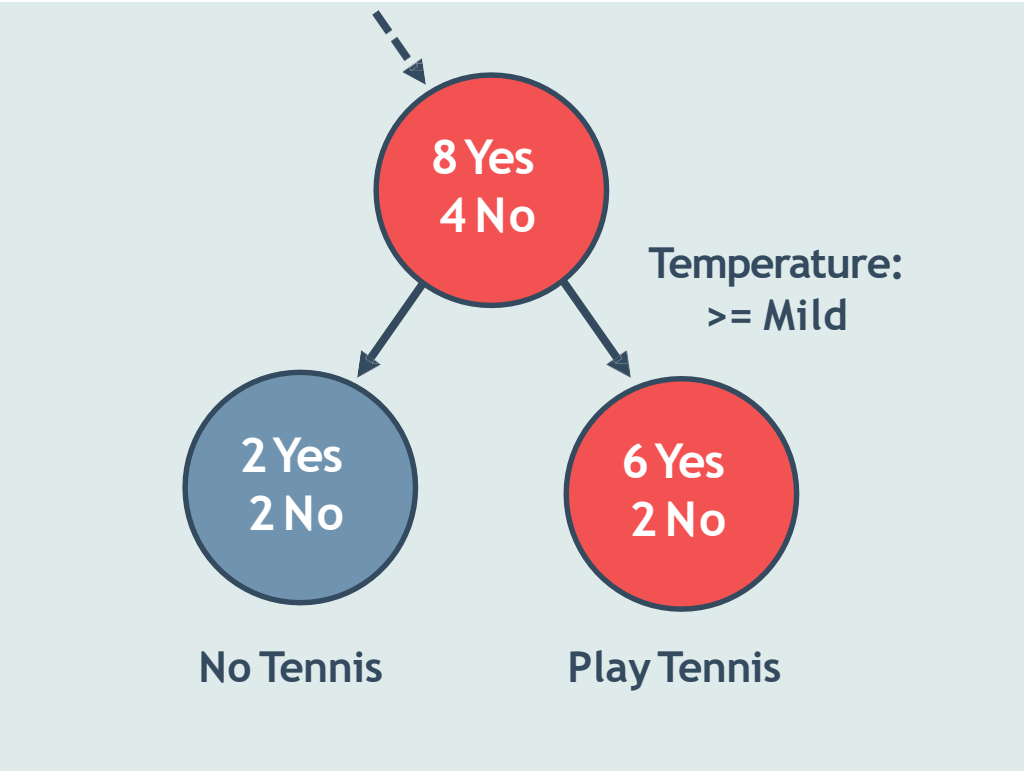

## 熵的公式

$$H(t) = -\sum_{i=1}^{n} p(i|t) log_2[p(i|t)]$$

## 熵的变化：

0.9183 – 4/12∗1.0000 – 8/12∗0.8113
= 0.0441

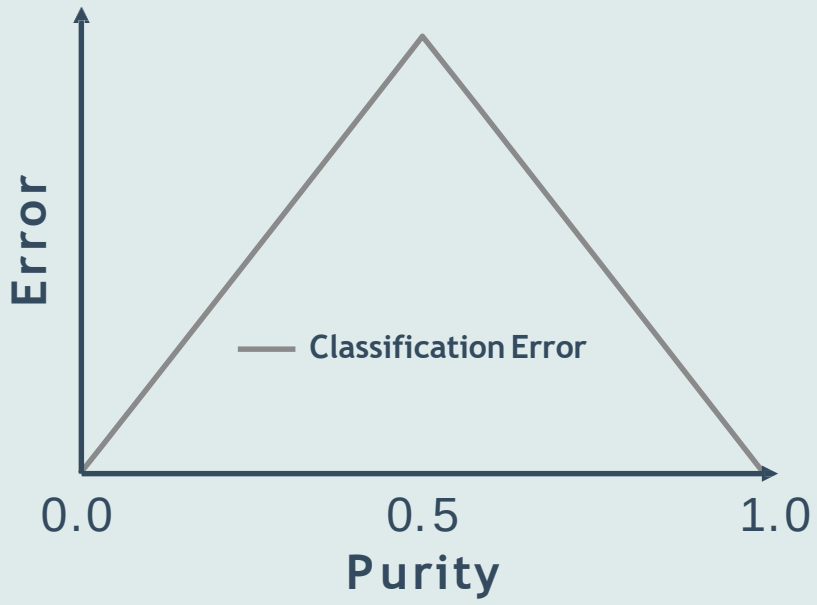
# 基于熵的划分




- 基于熵的划分允许继续分裂下去
- 最终达到叶子节点同质的目标
- 为什么熵可以达到这一目标，而分类错误不行？

# 分类错误 vs 熵

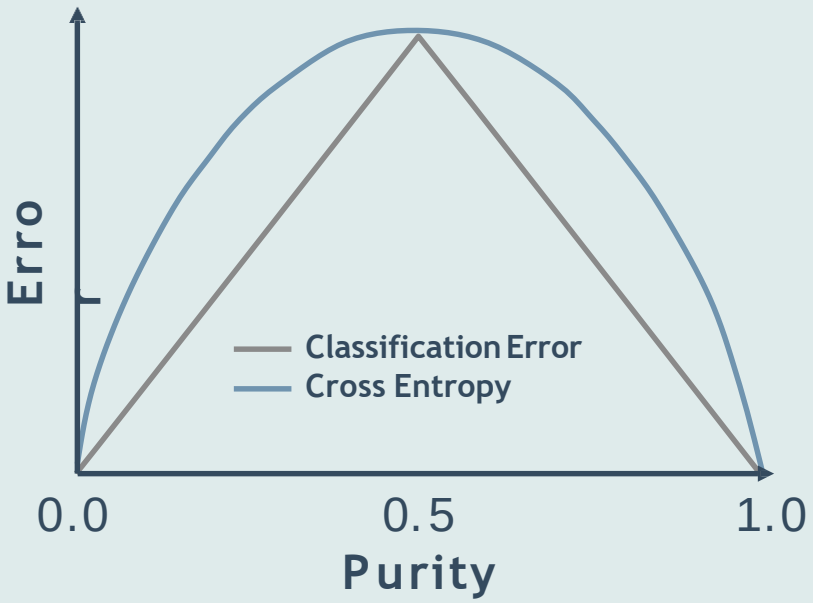
- 分类错误是一个平坦函数，在中心点达到最大值
- 中心点表示的是0.5/0.5的划分
- 分类指标偏向于远离中心点的结果




$$E(t) = 1 - \max_i [p(i|t)]$$

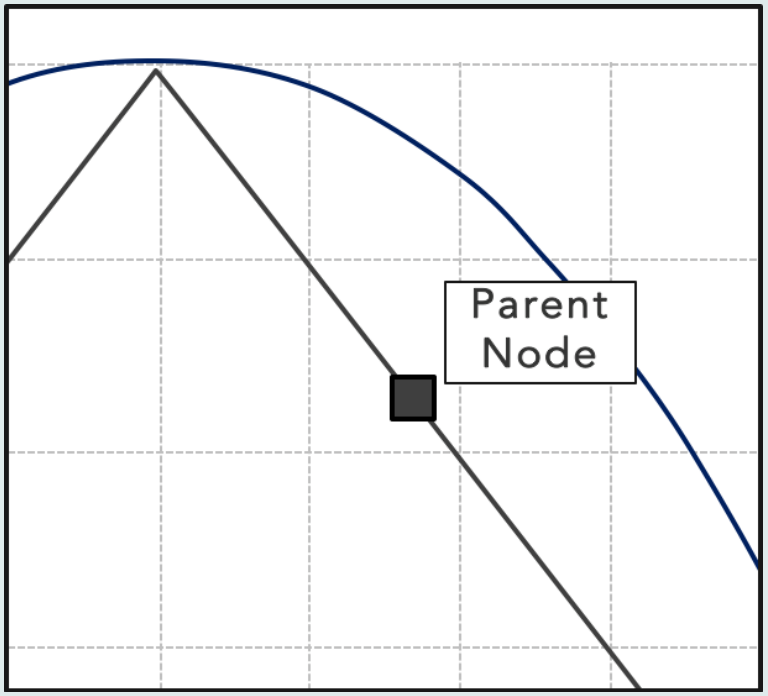
# 分类错误 vs 熵

- 熵具有相同的最大值，但是弯曲的曲线
- 曲度使得分裂可以继续到叶子节点纯了为止
- 为什么？



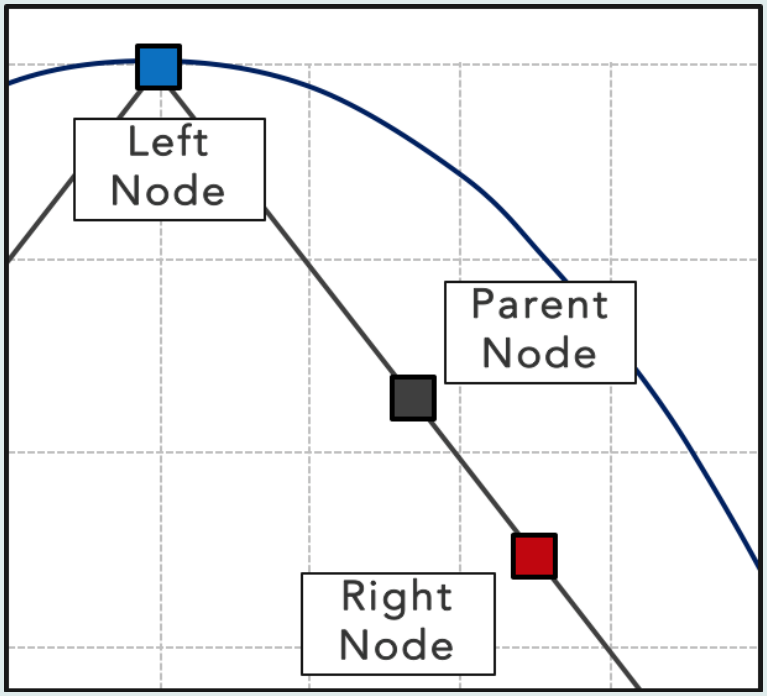

$$H(t) = -\sum_{i=1}^{n} p(i|t) log_2[p(i|t)]$$

# 分裂带来的信息增益



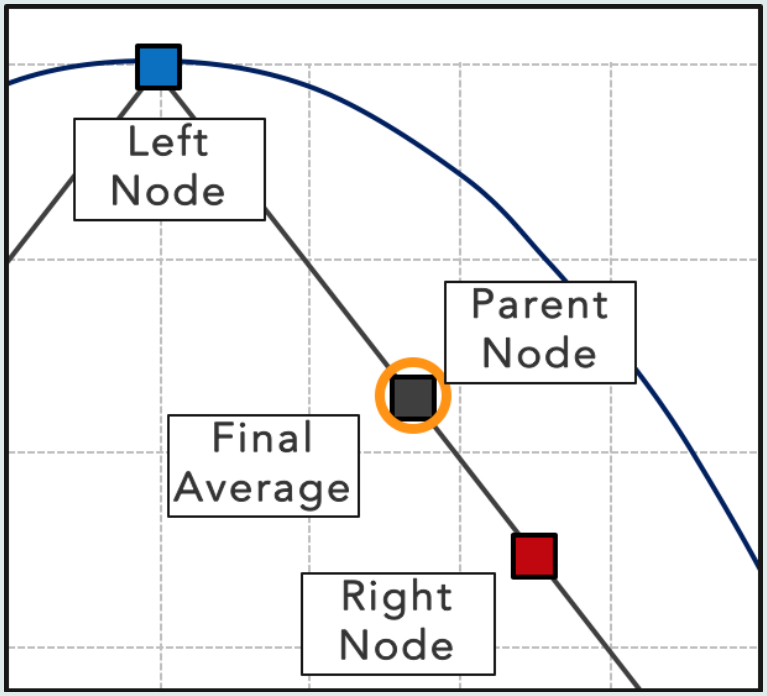

- 使用分类错误，函数是平坦的

# 分裂带来的信息增益



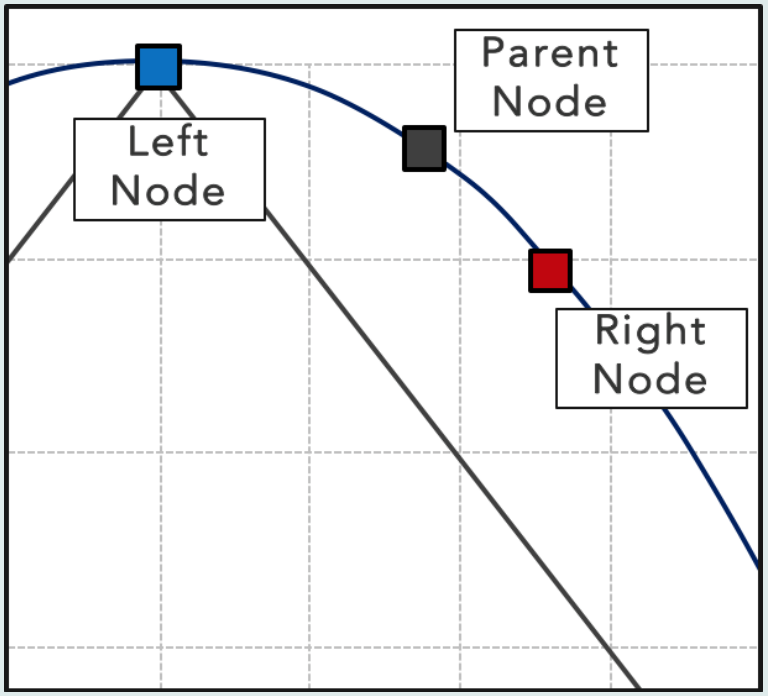

- 使用分类错误，函数是平坦的

# 分裂带来的信息增益



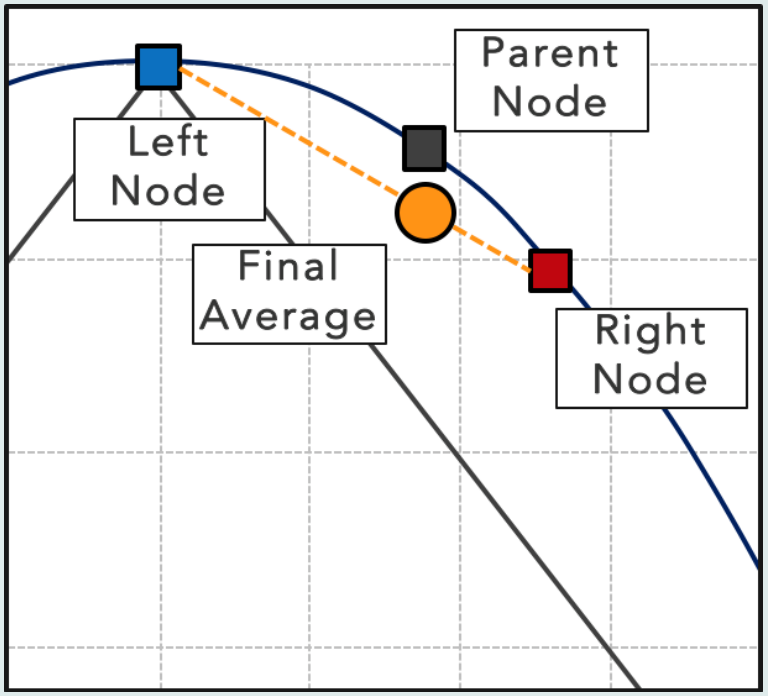

- 使用分类错误，函数是平坦的
- 最终的平均分类错误很有可能与父节点的分类错误相等
- 从而导致提前停止

# 分裂带来的信息增益




- 使用熵，函数有个“鼓包”

# 分裂带来的信息增益




- 使用熵，函数有个“鼓包”
- 使得子节点的平均熵少于父节点的熵
- 从而产生信息增益，使得分裂可以继续

# 信息增益(information gain)

假定离散属性$a$有$n$个可能的取值$\{a_1, a_2, \ldots, a_n\}$，若使用$a$来对样本集$D$进行划分，则会产生$n$个分支节点，其中第$i$个分支节点包含了$D$中所有在属性$a$上取值为$a_i$的样本，记为$D_i$,则

$$Gain(D, a) = Ent(D) - \sum_{i=1}^{n} \frac{|D_i|}{|D|} Ent(D_i)$$

# 信息增益(information gain)

| 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 |

# 信息增益率(information gain ratio)

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

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

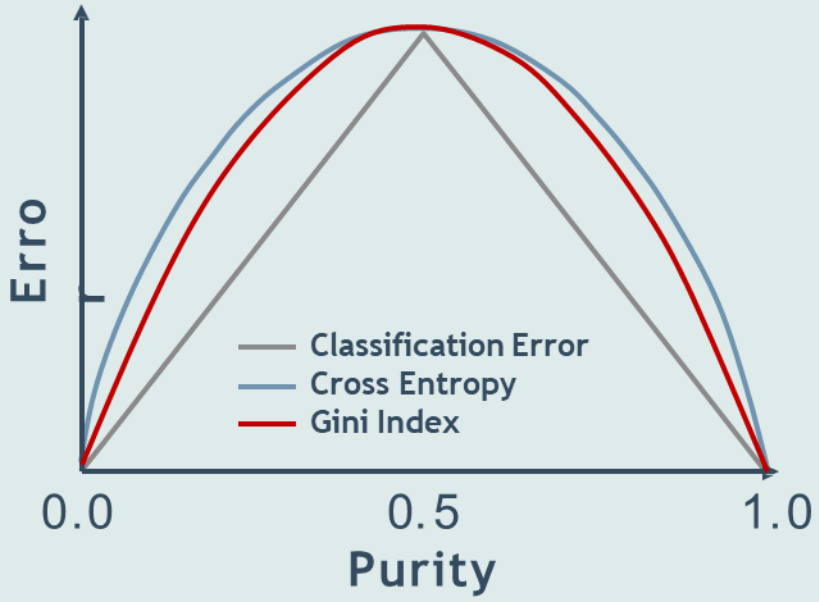
属性$a$的固有值

# 信息增益率(information gain ratio)

| 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 |

# 基尼指数

- 实际中，常使用基尼指数做分裂
- 其函数类似于熵---也有“鼓包”
- 没有对数

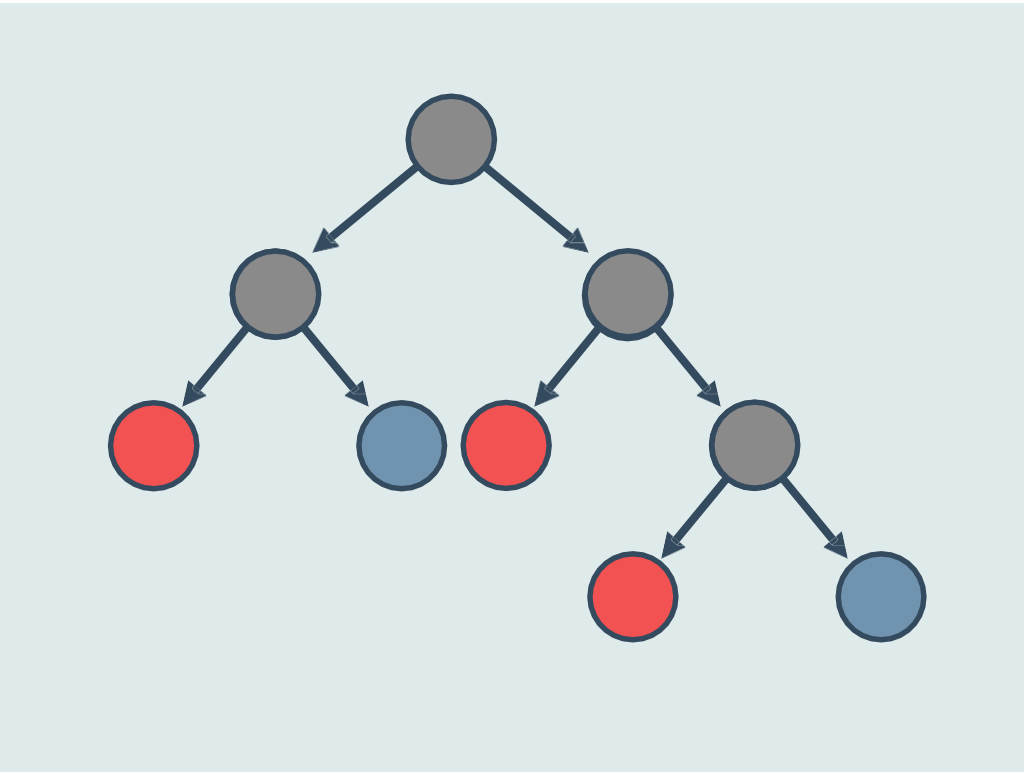



$$Gini(\mathrm{t}) = \sum_{i=1}^{n} p(i|t)[1 - p(i|t)] = 1 - \sum_{i=1}^{n} p(i|t)^2$$

# 基尼指数(Gini index)

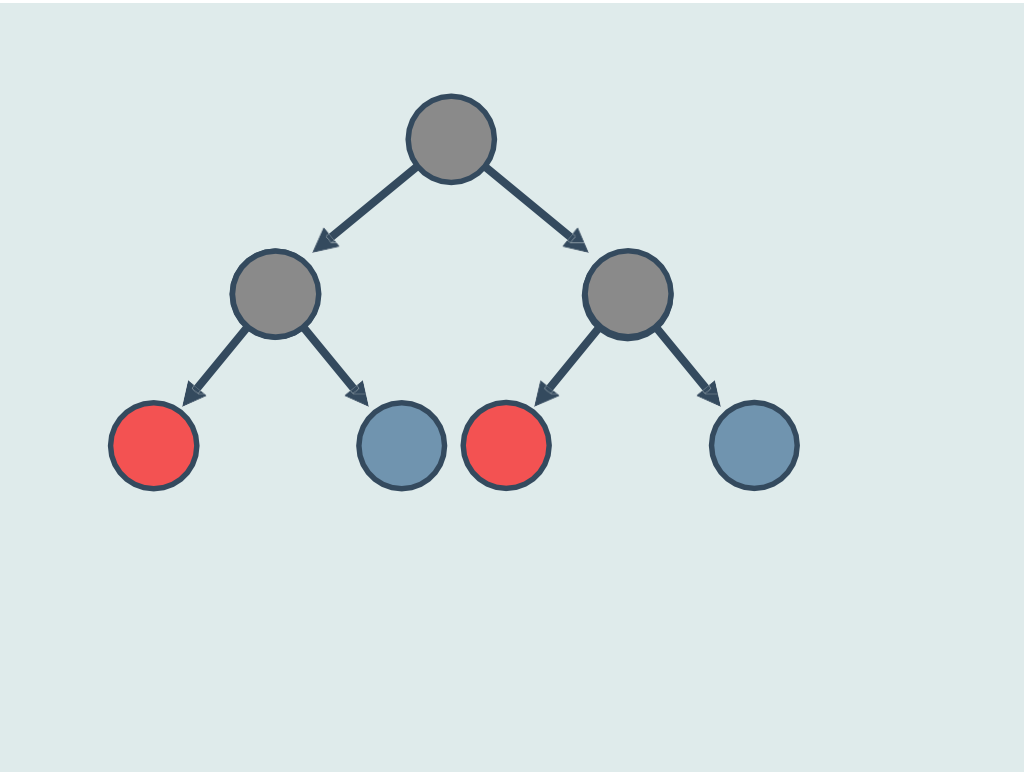
| 编号 | 色泽 | 根蒂 | 敲声 | 纹理 | 脐部 | 触感 | 好瓜 |
|---|---|---|---|---|---|---|---|
| 1 | 青绿 | 蜷缩 | 浊响 | 清晰 | 凹陷 | 硬滑 | 是 |
| 2 | 乌黑 | 蜷缩 | 沉闷 | 清晰 | 凹陷 | 硬滑 | 是 |
| 3 | 乌黑 | 蜷缩 | 浊响 | 清晰 | 凹陷 | 硬滑 | 是 |
| 4 | 青绿 | 蜷缩 | 沉闷 | 清晰 | 凹陷 | 硬滑 | 是 |
| 5 | 浅白 | 蜷缩 | 浊响 | 清晰 | 凹陷 | 硬滑 | 是 |
| 6 | 青绿 | 稍蜷 | 浊响 | 清晰 | 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 |

# 决策树高方差（high variance）



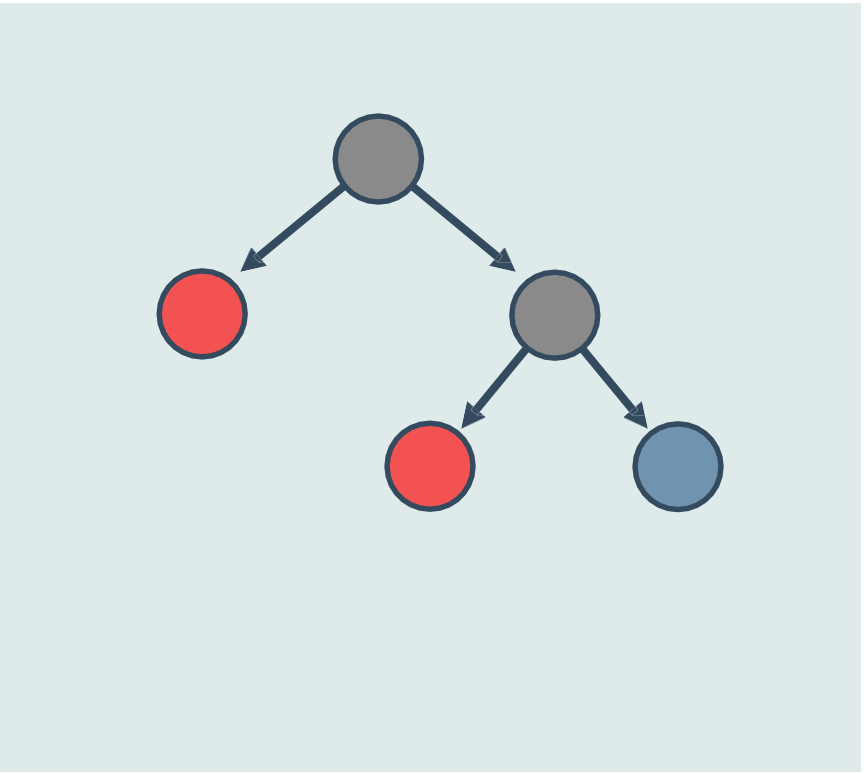
- 问题：决策树容易过拟合
- 数据微小的变化能对预测结果产生较大的影响 ---high variance

# 修剪决策树



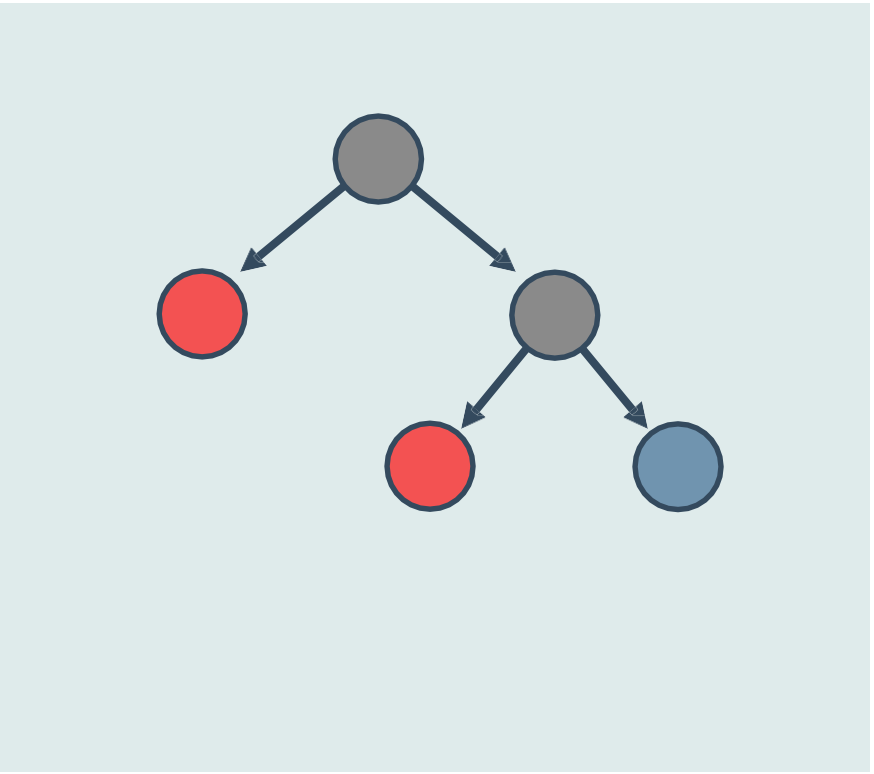
- 问题：决策树容易过拟合
- 数据微小的变化能对预测结果产生较大的影响 ---high variance
- 解决方案: 修剪决策树
  - 预剪枝
  - 后剪枝

# 预剪枝



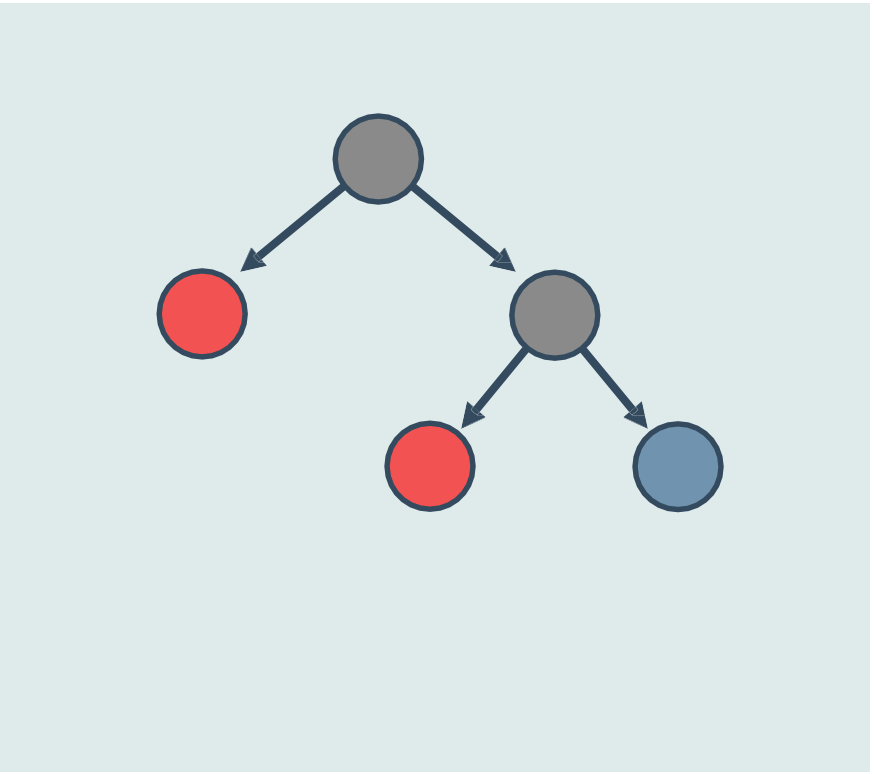
- 在决策树构建过程中，依据预先设定的条件，提前终止树的生长。
- Scikit-Learn中：
  - 决策树的最大深度（max_depth）
  - 决策树的最大叶子数（max_leaf_nodes）
  - 可分裂节点应包含的最少样例数（min_samples_split）
  - 叶节点应包含的最少样例数（min_samples_leaf）
  - 不纯度减少的最小量（min_impurity_decrease）

# 后剪枝



- 在决策树构建完成之后进行剪枝，得到一棵简化的树。
- 自底向上地考察每个非叶节点，如果将其子树剪去，成为一个叶节点，能带来决策树泛化性能提升，则将该子树替换为叶节点
- 错误率降低剪枝 (reduced-error pruning, REP)

# 后剪枝

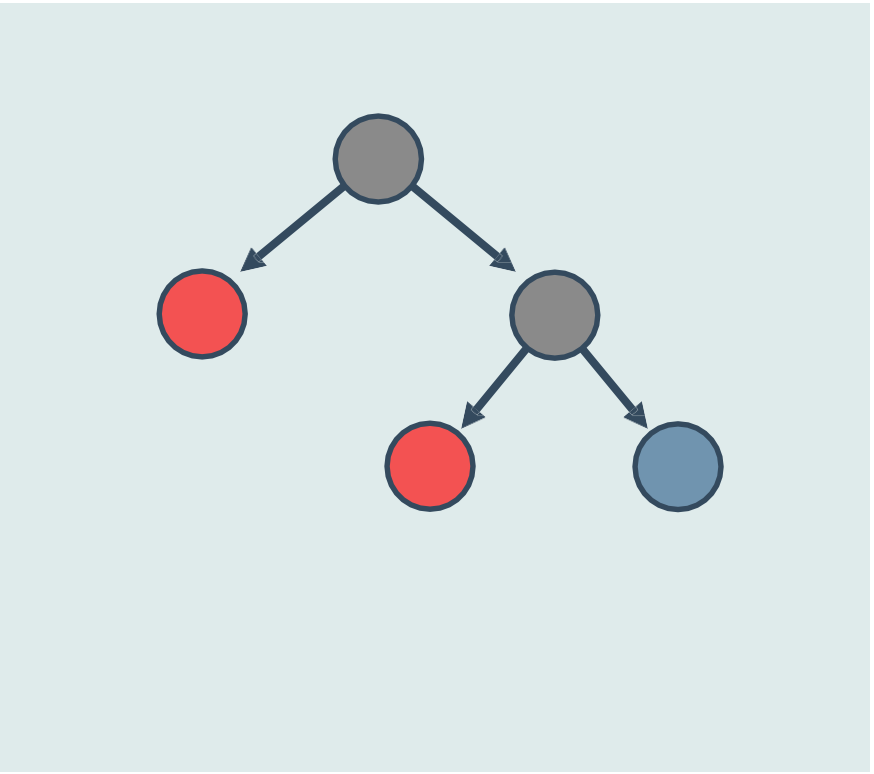


- Scikit-Learn从0.22版本开始实现了代价复杂度剪枝（cost-complexity, CCP）策略

$$\alpha = \frac{C(t) - C(T_t)}{|T_t| - 1}$$

  - $|T_t|$为子树中叶节点的个数
  - $C(T_t)$和$C(t)$分别是剪枝前后该子树的预测错误（或者所有叶节点的不纯度之和）
- 计算树中每个非叶节点的α 值，然后循环剪掉具有最小α 值的子树，直到最小α 值大于用户预先给定的参数值ccp_alpha 为止。

# 预剪枝vs. 后剪枝



- 一般情形下，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。但后剪枝训练时间比未剪枝和预剪枝决策树都要大得多。

# 连续与缺失值

连续值处理

给定样本集$D$和连续属性$a$，假定$a$在$D$上有$n$个不同的取值，从小到大排序后记为$\{a_1, a_2, \ldots, a_n\}$，若使用划分点$t$来对其进行划分，将$D$分为子集$D_t^-$和$D_t^+$（$D_t^-$：在属性$a$上取值$\leq t$，$D_t^+$：在属性$a$上取值$> t$）。

# 连续与缺失值

连续值处理

对相邻的属性取值$a_i$与$a_{i+1}$来说，在区间$[a_i,a_{i+1})$中取任意值所产生的划分结果相同。

把区间$[a_i,a_{i+1})$的中位点$\frac{a_i+a_{i+1}}{2}$作为候选划分点，则可参照离散属性值一样来考察这些划分点。

# 连续与缺失值

## 连续值处理

选取最优的划分点进行样本集合的划分，

假设 $T_a = \left\{ \frac{a_i + a_{i+1}}{2} \middle| 1 \le i \le n-1 \right\}$，则

$$Gain(D,a) = \max_{t \in T_a} Gain(D,a,t)$$

$$= \max_{t \in T_a} \left( Ent(D) - \sum_{\lambda \in \{-,+\}} \frac{|D_t^\lambda|}{|D|} Ent(D_t^\lambda) \right)$$

# 连续与缺失值

| 编号 | 色泽 | 根蒂 | 敲声 | 纹理 | 脐部 | 触感 | 密度 | 含糖率 | 好瓜 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 青绿 | 蜷缩 | 浊响 | 清晰 | 凹陷 | 硬滑 | 0.697 | 0.460 | 是 |
| 2 | 乌黑 | 蜷缩 | 沉闷 | 清晰 | 凹陷 | 硬滑 | 0.774 | 0.376 | 是 |
| 3 | 乌黑 | 蜷缩 | 浊响 | 清晰 | 凹陷 | 硬滑 | 0.634 | 0.264 | 是 |
| 4 | 青绿 | 蜷缩 | 沉闷 | 清晰 | 凹陷 | 硬滑 | 0.608 | 0.318 | 是 |
| 5 | 浅白 | 蜷缩 | 浊响 | 清晰 | 凹陷 | 硬滑 | 0.556 | 0.215 | 是 |
| 6 | 青绿 | 稍蜷 | 浊响 | 清晰 | 稍凹 | 软粘 | 0.403 | 0.237 | 是 |
| 7 | 乌黑 | 稍蜷 | 浊响 | 稍糊 | 稍凹 | 软粘 | 0.481 | 0.149 | 是 |
| 8 | 乌黑 | 稍蜷 | 浊响 | 清晰 | 稍凹 | 硬滑 | 0.437 | 0.211 | 是 |
| 9 | 乌黑 | 稍蜷 | 沉闷 | 稍糊 | 稍凹 | 硬滑 | 0.666 | 0.091 | 否 |
| 10 | 青绿 | 硬挺 | 清脆 | 清晰 | 平坦 | 软粘 | 0.243 | 0.267 | 否 |
| 11 | 浅白 | 硬挺 | 清脆 | 模糊 | 平坦 | 硬滑 | 0.245 | 0.057 | 否 |
| 12 | 浅白 | 蜷缩 | 浊响 | 模糊 | 平坦 | 软粘 | 0.343 | 0.099 | 否 |
| 13 | 青绿 | 稍蜷 | 浊响 | 稍糊 | 凹陷 | 硬滑 | 0.639 | 0.161 | 否 |
| 14 | 浅白 | 稍蜷 | 沉闷 | 稍糊 | 凹陷 | 硬滑 | 0.657 | 0.198 | 否 |
| 15 | 乌黑 | 稍蜷 | 浊响 | 清晰 | 稍凹 | 软粘 | 0.360 | 0.370 | 否 |
| 16 | 浅白 | 蜷缩 | 浊响 | 模糊 | 平坦 | 硬滑 | 0.593 | 0.042 | 否 |
| 17 | 青绿 | 蜷缩 | 沉闷 | 稍糊 | 稍凹 | 硬滑 | 0.719 | 0.103 | 否 |

# 连续与缺失值

缺失值处理

- 如何在属性值缺失的情况下进行划分属性的选择?
- 给定划分属性，若样本在该属性上的值缺失，如何对样本进行划分?

# 连续与缺失值

缺失值处理

给定样本集$D$和属性$a$，令$\tilde{D}$表示$D$中在属性$a$上没有缺失值的样本子集。对于第一个问题，根据$\tilde{D}$来判断属性$a$的优劣。

# 连续与缺失值

缺失值处理

假定属性$a$在$D$上有$n$个可取值$\{a_1, a_2, \ldots, a_n\}$，令$\tilde{D}_i$表示$\tilde{D}$中在属性$a$上取值为$a_i$的样本子集，$\tilde{D}_k$表示$\tilde{D}$中属于第$k$类($k$=1,2,…,K)的样本子集。则$\tilde{D}_i = \cup_{i=1}^{n} \tilde{D}_i$，$\tilde{D}_i = \cup_{k=1}^{K} \tilde{D}_k$。

# 连续与缺失值

缺失值处理

假设我们为每个样本$x$赋予一个权重$\omega_x$，并定义：

$$\rho = \frac{\sum_{x \in \tilde{D}} \omega_x}{\sum_{x \in D} \omega_x}$$

$$\tilde{p}_k = \frac{\sum_{x \in \tilde{D}_k} \omega_x}{\sum_{x \in \tilde{D}} \omega_x} \quad (1 \leq k \leq K)$$

$$\tilde{r}_i = \frac{\sum_{x \in \tilde{D}_i} \omega_x}{\sum_{x \in \tilde{D}} \omega_x} \quad (1 \leq k \leq K)$$

# 连续与缺失值

缺失值处理

$$Gain(D,a) = \rho \times Gain\left(\widetilde{D},a\right)$$
$$= \rho \times \left( Ent\left(\widetilde{D}\right) - \sum_{i=1}^{n} Ent\left(\widetilde{D}_i\right) \right)$$

$$Ent\left(\widetilde{D}\right) = -\sum_{k=1}^{K} \tilde{p}_k \log_2 \tilde{p}_k$$

# 连续与缺失值

缺失值处理

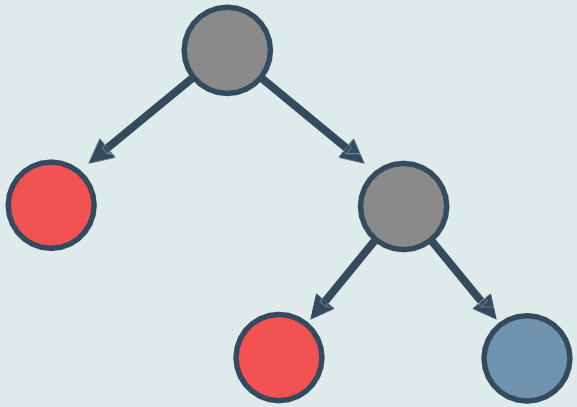
对于第二个问题，若样本$x$在划分属性$a$上取值已知，则将$x$划入与其取值对应的子节点，且样本权值在子节点中保持为$\omega_x$。

若样本$x$在划分属性$a$上取值已未知，则将$x$同时划入所有子节点，且样本权值在与属性$a_i$对应的子节点中调整为$\tilde{r}_i \cdot \omega_x$。

# 连续与缺失值

| 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 |

# 决策树的优点



- 容易实现和解释
  - “if … then … else”逻辑
- 可以处理任何数据类型
  - 类别型，连续数值
- 无需数据预处理和缩放
- 易于追溯和倒推

| | |
|---|---|
| criterion | **测试条件选择标准**<br>DecisionTreeClassifier的缺省值是“gini”，即用基尼指数作为衡量指标，也可以是“entropy”，即使用熵作为衡量指标；<br>DecisionTreeRegressor的缺省值是“mse”，即使用均方差作为衡量指标，也可以是“mae”，即使用平均绝对值误差作为衡量指标。 |
| splitter | **测试条件选择策略**<br>缺省值是“best”，即选取最优划分条件，也可以是“random”，表示随机选取划分条件。 |
| max_depth | **决策树的最大深度**<br>缺省值是没有深度限制。设置树的最大深度是为了防止过拟合。 |
| min_samples_split | **节点可分裂的最少样例数**<br>缺省值是2。一个节点可以进一步分裂必须最少包含min_samples_split个样例为了防止过拟合，可以增大此值。 |
| min_samples_leaf | **叶子节点的最少样例数**<br>缺省值是1。一个叶子节点必须最少包含min_samples_leaf个样例。如果增大此值，可以及早停止过于细分叶子节点，防止过拟合。 |
| max_features | **选择测试条件可考虑的最大特征数**<br>缺省值是没有最大特征数的限制，即可以考虑数据集中的所有特征。减少考虑的特征数，一来可以减少决策树的生成时间；二来可以增大决策树的随机性，有利于提升随机森林等集成学习模型的效果。 |
| max_leaf_nodes | **最大叶节点个数**<br>缺省值是不限制叶节点的个数。它和树的最大深度类似，可以防止过拟合。 |
| min_impurity_decrease | **最小不纯度减少量**<br>如果用某一测试条件划分节点带来的不纯度的减少量小于这个阈值，则不用此测试条件划分该节点。 |