# 机器学习——python——决策树

决策树是一种基本的分类与回归方法，它可以看作if-then规则的集合，也可以认为是定义在特征空间与类空间上的条件概率分布。将决策树转换成if-then规则的过程如下：

1. 由决策树的根节点到叶节点的每一条路径构建一条规则；
2. 路径内部结点的特征对应规则的条件；
3. 叶节点的类对应规则的结论.

决策树学习算法主要由三部分构成：

1. 特征选择
2. 决策树生成
3. 决策树的剪枝

**优**

**- 易于理解和解释，甚至比线性回归更直观；**

**- 与人类做决策思考的思维习惯契合；**

**- 模型可以通过树的形式进行可视化展示；**

**- 可以直接处理非数值型数据，不需要进行哑变量的转化，甚至可以直接处理含缺失值的数据；**

**缺**

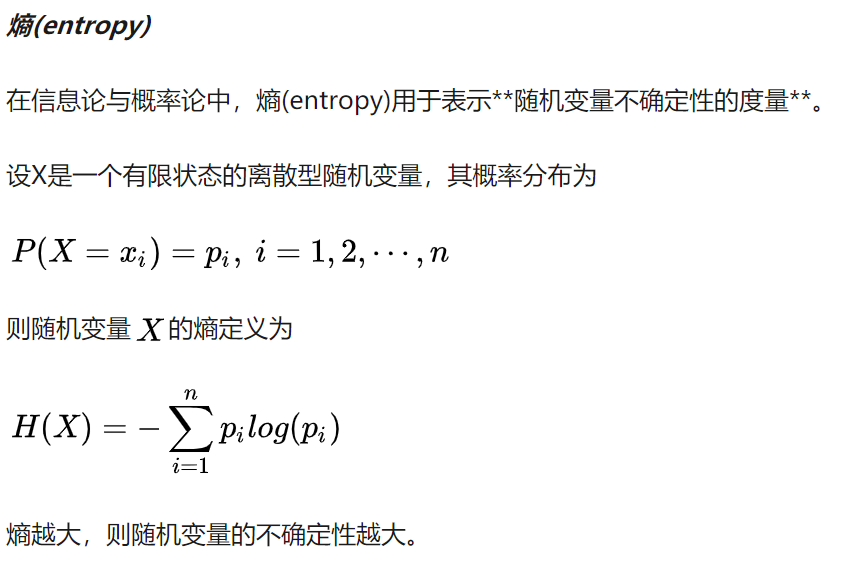
**- 对于有大量数值型输入和输出的问题，决策树未必是一个好的选择；**

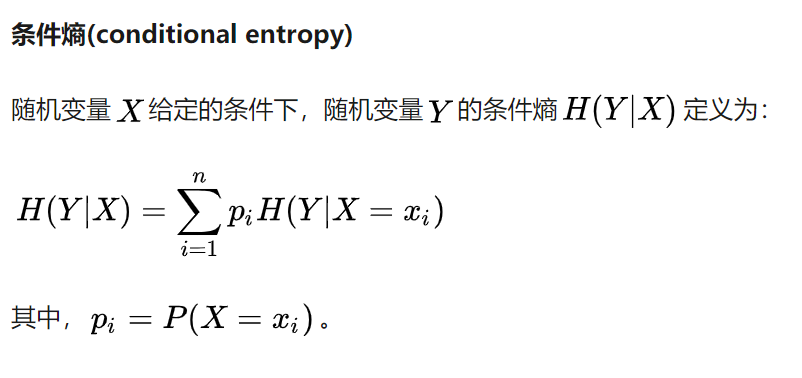
**- 特别是当数值型变量之间存在许多错综复杂的关系，如金融数据分析；**

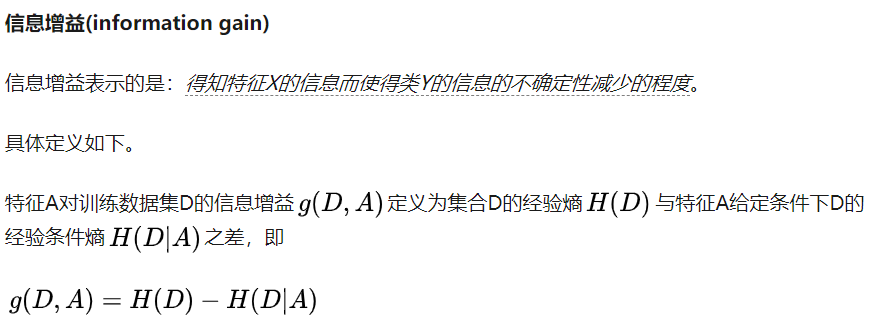
**- 决定分类的因素取决于更多变量的复杂组合时；**

**- 模型不够稳健，某一个节点的小小变化可能导致整个树会有很大的不同。**

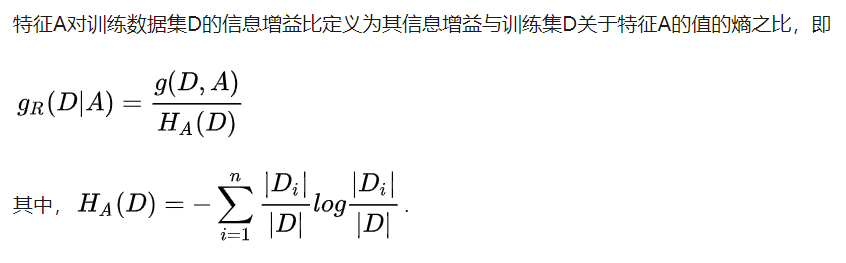
## 一、特征选择

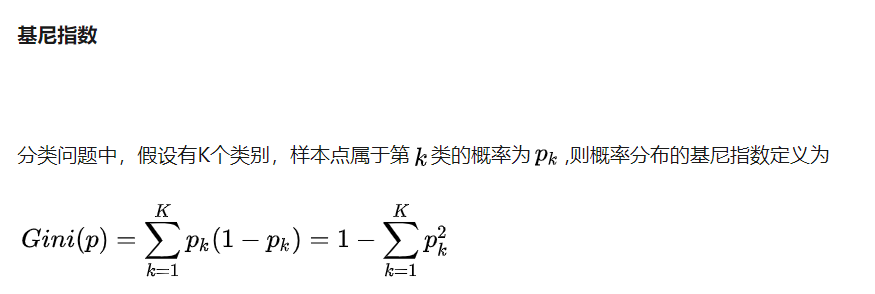












## 二、决策树的生成

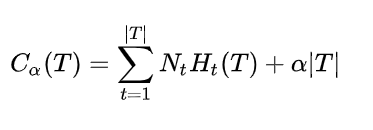
**常用的算法有：ID3，C4.5，CART。相应的准则分别是信息增益，信息增益比和基尼指数。**

ID3：从根节点开始，对结点计算所有可能特征的信息增益，选择信息增益最大的特征作为结点的特征，并由该特征的不同取值构建子节点；对子节点递归地调用以上方法，构建决策树；直到所有特征的信息增益均很小或者没有特征可选时为止。

CART假设决策树是一个二叉树，它通过递归地二分每个特征，将特征空间划分为有限个单元，并在这些单元上确定预测的概率分布。CART算法中，对于回归树，采用的是平方误差最小化准则；对于分类树，采用基尼指数最小化准则。

## 三、决策树的剪枝

由于生成的决策树存在过拟合问题，需要进行剪枝以优化决策树。往往是从已经生成的树上剪掉一些叶结点及以上的子树，并将器父结点或者其根结点作为新的叶结点，从而简化生成的决策树。决策树的剪枝通过极小化决策树整体的损失函数来实现。在提高信息增益的基础上，通过对模型的复杂度T施加惩罚，便得到了损失函数的定义：



具体的算法如下：

1. 计算每个节点的经验熵；

2. 递归地从树的叶节点向上回缩，如果将某一个父节点的所有叶节点合并，能够使得其损失函数减小，则进行剪枝，将父节点变成新的叶节点；

3. 返回2，直到不能继续合并。

## 四、实战

在这个例子中，一个人将尝试决定他/她是否应该参加喜剧节目。例中人物每次在镇上举办喜剧节目时都进行注册，并注册一些关于喜剧演员的信息，并且还登记了他/她是否去过。基于此数据集，Python 可以创建决策树，这个决策树可用于决定是否值得参加任何新的演出。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Age** | **Experience** | **Rank** | **Nationality** | **Go** |
| 36 | 10 | 9 | UK | NO |
| 42 | 12 | 4 | USA | NO |
| 23 | 4 | 6 | N | NO |
| 52 | 4 | 4 | USA | NO |
| 43 | 21 | 8 | USA | YES |
| 44 | 14 | 5 | UK | NO |
| 66 | 3 | 7 | N | YES |
| 35 | 14 | 9 | UK | YES |
| 52 | 13 | 7 | N | YES |
| 35 | 5 | 9 | N | YES |
| 24 | 3 | 5 | USA | NO |
| 18 | 3 | 7 | UK | YES |
| 45 | 9 | 9 | UK | YES |

结果说明：

