法律声明

- 课程详情请咨询
  - 微信公众号：北风教育
  - 官方网址：*http://www.ibeifeng.com/*

人工智能之机器学习

决策树（Decision Tree）

主讲人：赵翌臣

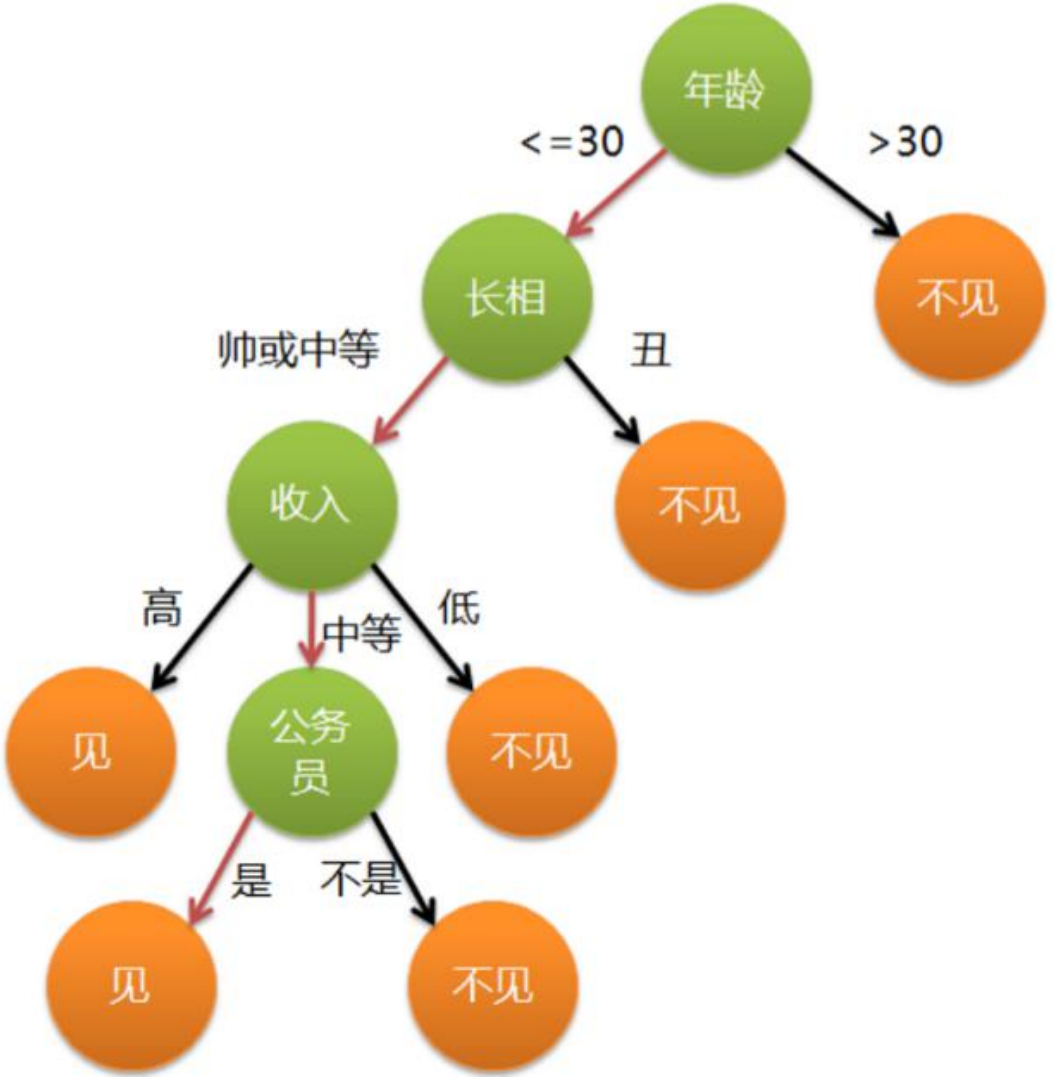
上海育创网络科技有限公司

决策树直观理解

- 凯特想在某征婚网站上寻找自己的另一半，但她发现在海量的信息中筛选出值得见面的人比较繁琐。因此，凯特想建立一个模型，让模型帮自己从众多候选者中筛选出自己有意愿见面的对象。
- 做法：从网站里随机挑选了100个人，主要观察4个特征，并给出标记（是否见面）。然后凯特训练了一个决策树模型，通过模型对征婚网站上的人进行筛选

序号	收入	年龄	公务员	长相	是否见面
1	22k	25	否	帅	是
2	7k	23	是	中等	否
3	25k	31	否	中等	否
....	....	....	....	....	....
100	15k	27	是	中等	是

决策树直观理解



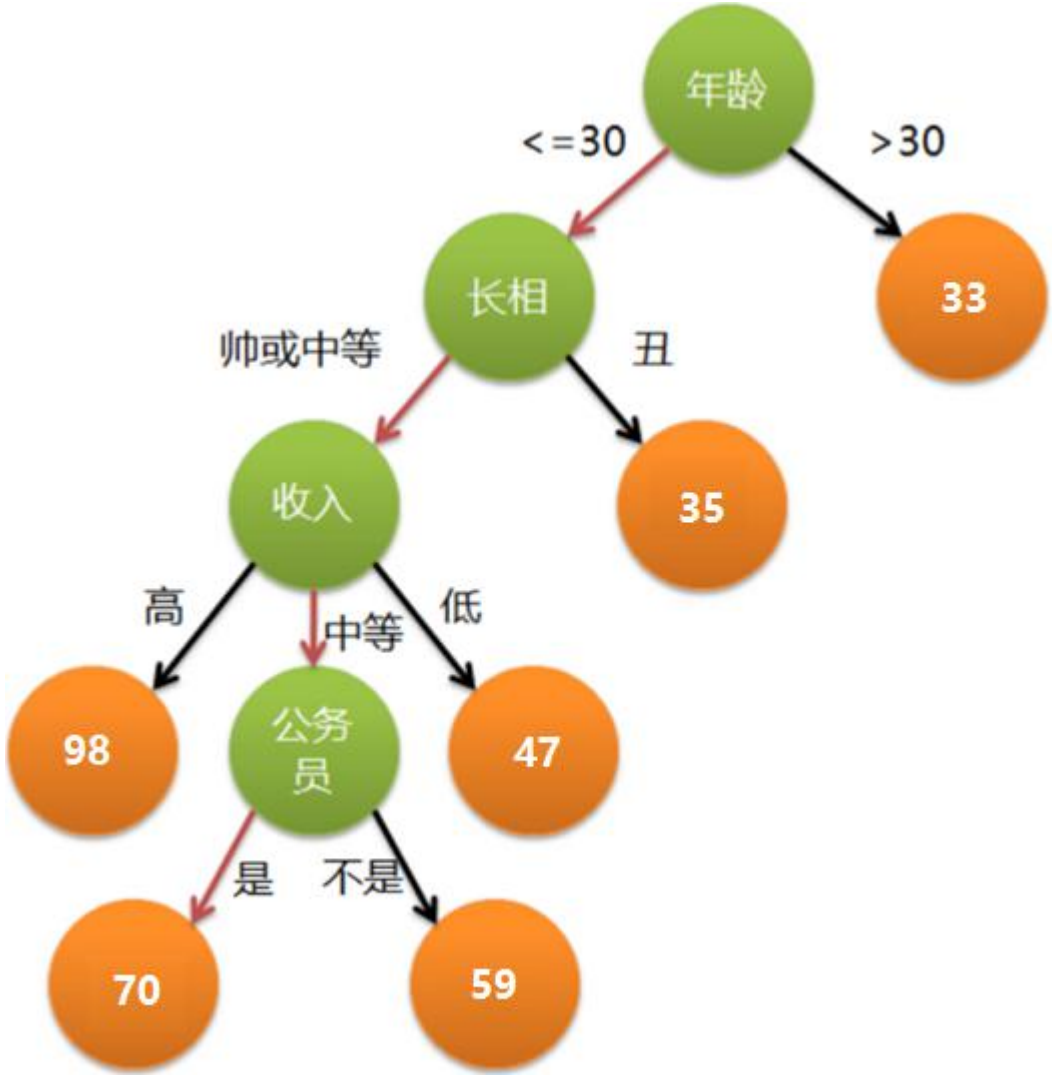
分类决策树

决策树直观理解

- 凯特想在某征婚网站上寻找自己的另一半，但她发现在海量的信息中筛选出值得见面的人比较繁琐。因此，凯特想建立一个模型，让模型帮自己从众多候选者中筛选出自己有意愿见面的对象。
- 做法：从网站里随机挑选了100个人，主要观察4个特征，并给出标记（这个男生在自己心里的打分）。然后凯特训练了一个决策树模型，通过模型对征婚网站上的人进行筛选

序号	收入	年龄	公务员	长相	打分
1	22k	25	否	帅	97
2	7k	23	是	中等	43
3	25k	31	否	中等	30
....	....	....	....	....	....
100	15k	27	是	中等	73

决策树直观理解



回归决策树

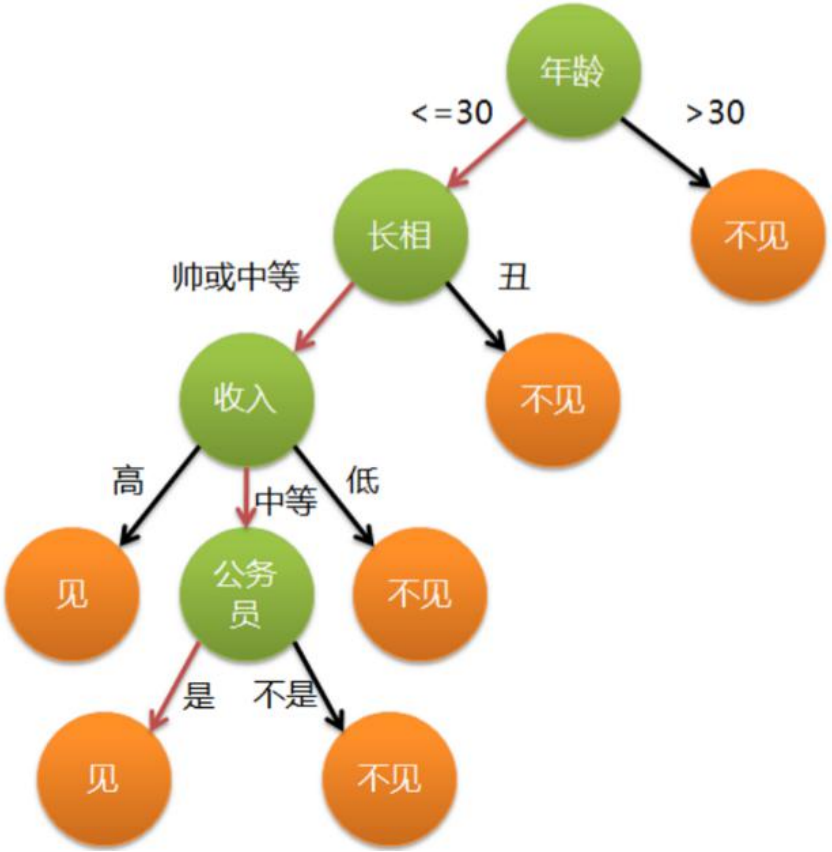
什么是决策树（Decision tree）？

- 决策树结构
  - 结点（node）：内部结点和叶结点。内部结点=》？；叶结点=》？
  - 有向边
- 分类决策树
  - 叶结点=》类别
- 回归决策树
  - 叶结点=》实数

特征 标签

决策树的使用

- 决策树的构建
  - 暂略
- 决策树的预测
  - 从根结点开始，对实例的某一特征进行测试，根据测试结果将实例分配到其子结点，如此递归地对实例进行判断并分配，直至达到叶结点。
  - 分类：输出得票最多的类
  - 回归：输出样本标签的均值
- 备注
  - 可以将决策树看成一个if-else规则的集合。每一个实例只被一条路径所覆盖

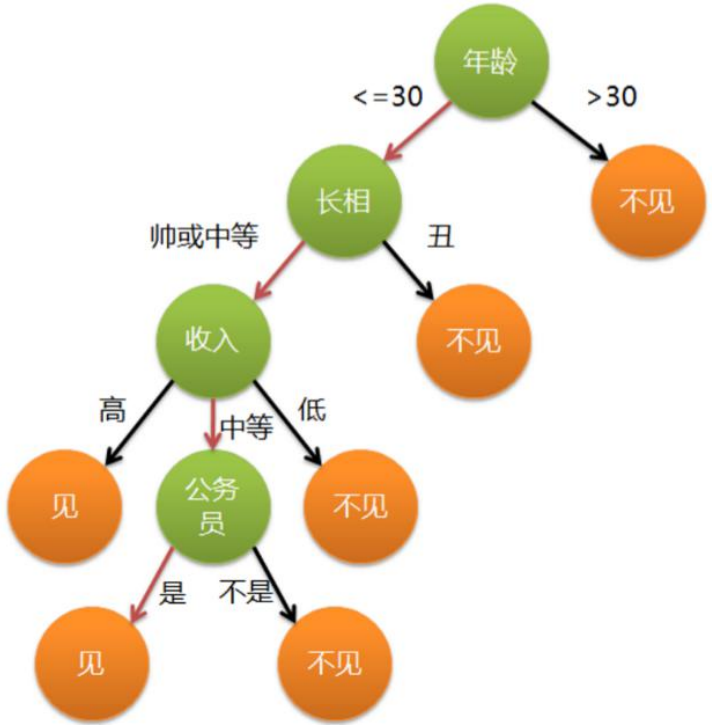


决策树介绍

- 从历史上来说，决策树的起源非常早，甚至在有机器学习之前就有决策树了，因为决策树模仿的是人类在做决策的过程。在医药、商业的应用里，如何让电脑模仿人类做决策是一个很早的人工智能问题
- 决策树的构建包含了很多前人的巧思（启发式），但是为什么要这样做并没有什么理论保证，决策树的构建算法有上百种，没有人能说哪个好哪个不好，某些决策树算法流行起来，这些都是历史的选择。在Spark中实现了ID3算法和CART算法的决策树，我们将会重点讲解。

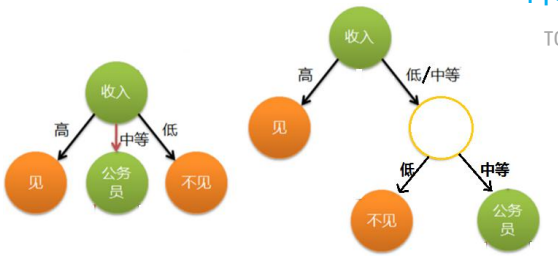
如果让你去建一棵决策树，你怎么建?

- 第一步：确定评价决策树好坏的指标（用什么？）
- 第二步：列举出所有可能的决策树，然后计算各自的指标，从而选出最优的一棵树
- 缺点是什么?

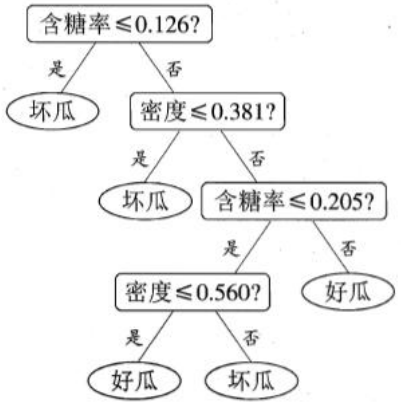


在测试集上的表现
从所有可能的决策树中选取最优的DT是一个NP-hard问题

启发式学习

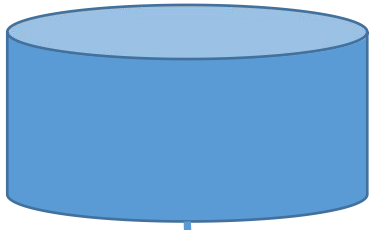


- 贪心策略
  - 贪心策略举例
  - 思想：确定贪心指标，在候选方案集合中执行一个让贪心指标最大的方案。不会从全局最优的角度思考问题，近似求解，这个解可能是次优解（sub-optimal）



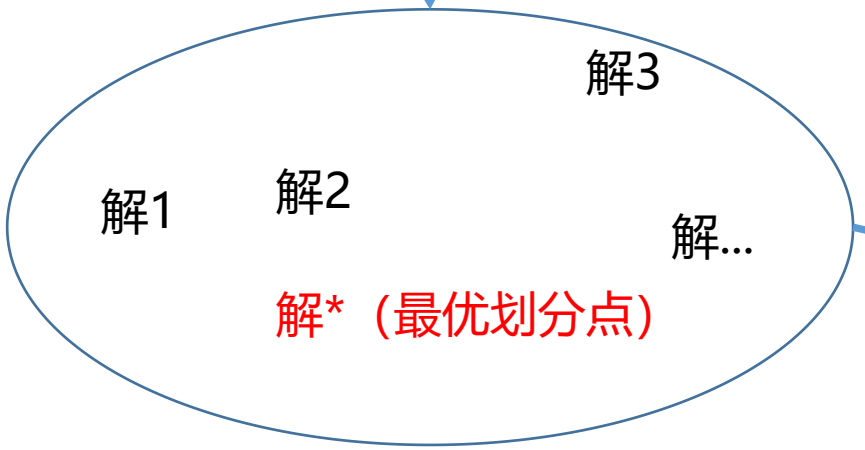
- 决策树的贪心指标
  - 信息增益（后面谈）
- 候选方案集合（连续特征，离散特征，标签）
  - 连续特征：比如年龄中<10岁，<20岁，<30岁等等都可以是划分点。假设：5种方案
  - 离散特征：
    - 多叉树：直接按照不同特征值划分即可，比如长相里的丑，中等，帅。1种方案
    - 二叉树：丑/中等帅，丑中等/帅，丑帅/中等，都是划分点。3种方案
  - 连续特征和离散特征划分点取并集，5+1或5+3种方案

启发式构建决策树过程



数据集

每个特征的划分点都是一个候选解

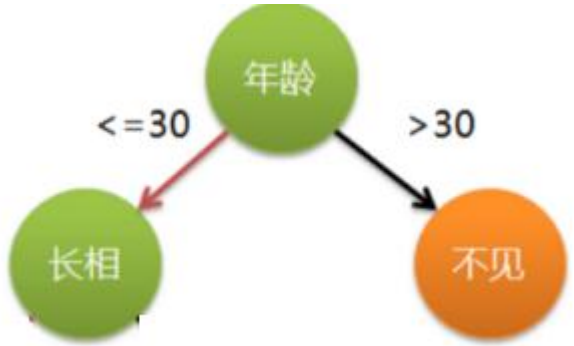


全体划分点候选集
（连续特征和离散特征划分点取并集）

贪心策略

通过最大化贪心指标，寻找最优解

构建决策树



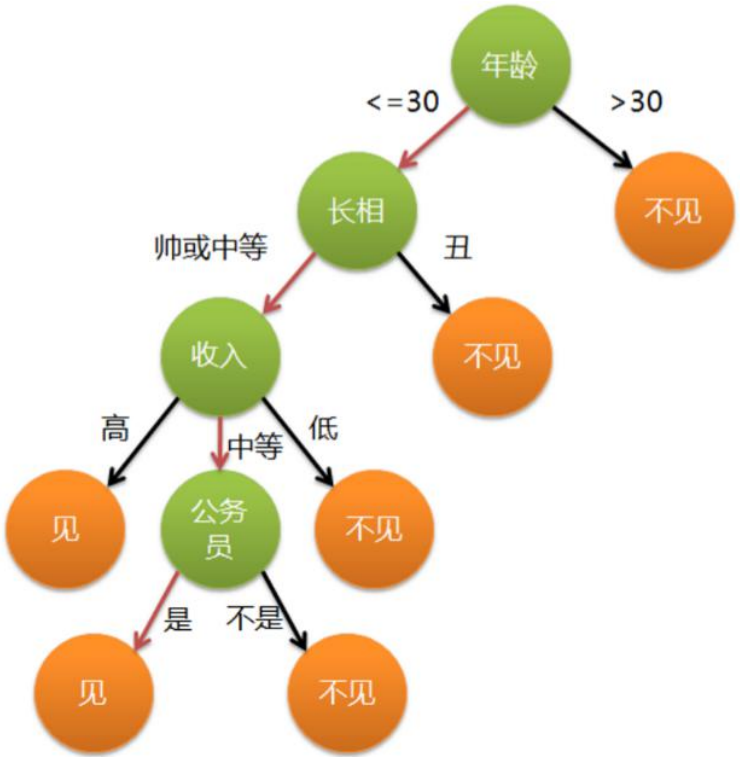
启发式学习的两个问题

- 这个贪心指标怎么设计?
- 怎么确定全体划分点候选集?

从模型复杂度角度出发，设计贪心指标

- 比如有以下数据集，请构建一棵决策树

序号	颜色（特征一）	类型（特征二）	标记
1	黄	A	是
2	红	B	否
3	蓝	C	否
4	黄	B	是

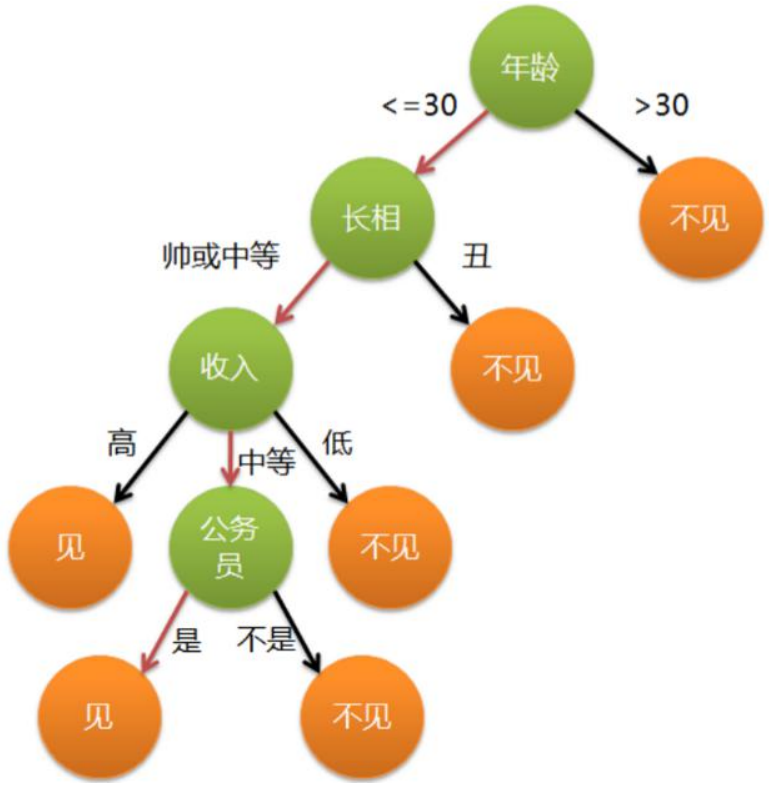


- 思考：为什么要在每个结点上费老大劲去选择最优划分点呢？随便选呗？

假设有一个训练集，有4个特征A、B、C、D；标记={0,1}。我们发现：无论ABC取什么，标记都和D的有关系，也就是说，D是最主要的因素。如果用D作为划分特征的话，我们的决策树将会十分精致（模型即简单又准确）；如果没选D，那么模型可能会变得复杂，还会增加计算量。

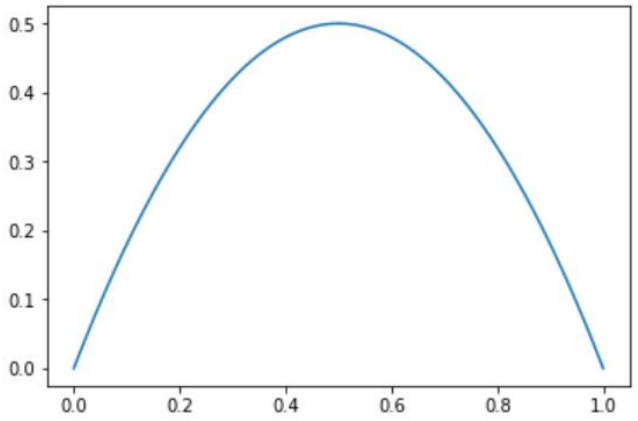
贪心指标与建树方法

- 建树的方法论
  - 让叶结点尽早变得更纯
- 衡量不纯度的指标
  - 树结点内样本标签不一致的程度
- 衡量变纯程度的指标（信息增益）
  - 划分前的不纯度为a，比如划分后产生两个结点，它们的不纯度分别为b和c，可以对其加权为B和C，那信息增益=a-(B+C)
- 建树过程
  - 遍历全体划分点候选集，选择信息增益最大的划分点构建决策树，递归执行

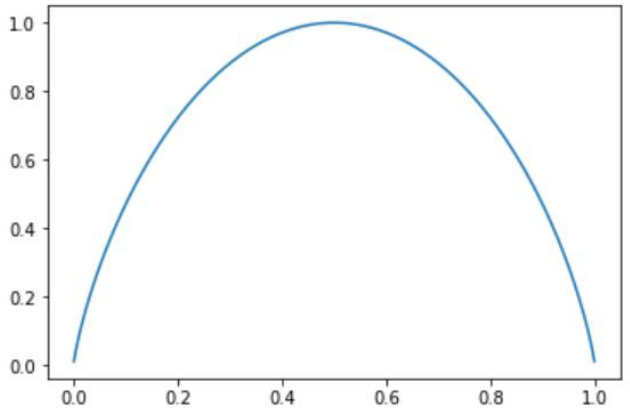


衡量不纯度的指标

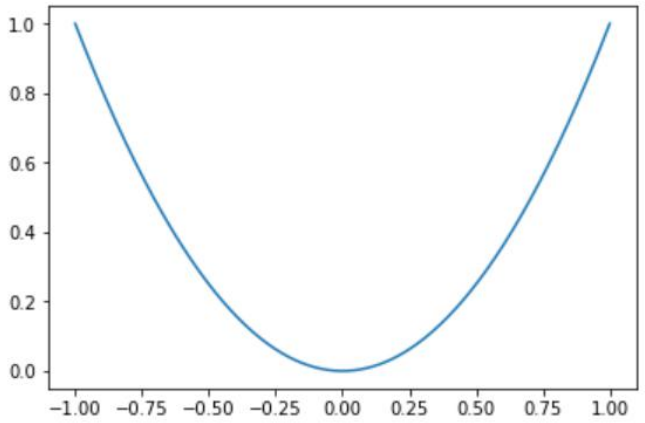
	Impurity	Task	Formula	Description
基尼不纯度：	Gini impurity	Classification	$\sum_{i=1}^{C} f_i(1-f_i)$	f_i is the frequency of label i at a node and C is the number of unique labels.
香农熵：	Entropy	Classification	$\sum_{i=1}^{C} -f_i log(f_i)$	f_i is the frequency of label i at a node and C is the number of unique labels.
均方误差：	Variance	Regression	$\frac{1}{N}\sum_{i=1}^{N}(y_i-\mu)^2$	y_i is label for an instance, N is the number of instances and μ is the mean given by $\frac{1}{N}\sum_{i=1}^{N} y_i$.



基尼指数



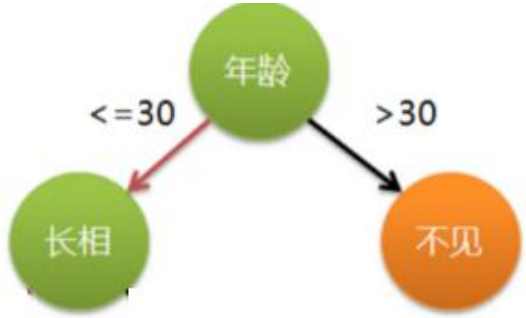
香农熵



方差

衡量不纯度的指标与信息增益

- **基尼指数（分类）：**
  - 两次抽取，拿到两个不同类别实例的概率，如果结点中实例是纯的，那么基尼指数=0
- **香农熵（分类）：**
  - 刻画不确定程度，如果结点中实例是纯的，那么香农熵=0
- **均方误差（回归）：**
  - 结点内的均方误差
- 信息增益
  - 信息增益 = 原来结点的不纯度 – 子结点不纯度的和。



$$IG(D,s) = Impurity(D) - \frac{N_{left}}{N} Impurity(D_{left}) - \frac{N_{right}}{N} Impurity(D_{right})$$

信息增益与决策树算法

- 根据不同的衡量结点不纯的指标，一些无聊的大人非要说使用了不同的算法，并纷纷给这些构建决策树的算法起了名字。
- **ID3算法：**基于香农熵增益，缺点：会偏爱取值较多的特征
  - 香农熵增益 = 结点的香农熵 – 子结点香农熵的带权和
- **C4.5算法：**基于香农熵增益比，缺点：计算复杂度高
  - 香农熵增益比 = 参数*香农熵增益　　ps：某特征的特征值种类越多，那么参数越　。
- **CART分类算法：**基于基尼指数增益
  - 基尼指数增益 = 结点的基尼指数 – 子结点的基尼指数的带权和
- **CART回归算法：**基于方差增益
  - 方差增益 = 结点的方差 – 子结点的方差的带权和

启发式学习的两个问题

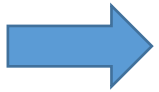
- 这个贪心指标怎么设计？
  - 第一步：设计衡量结点不纯度的指标
  - 第二步：设计衡量变纯程度的指标（信息增益，贪心指标）
  - 第三步：遍历全体划分点候选集，使用能让信息增益最大的划分点构建决策树。递归构建树，直到达到停止条件，完成整棵决策树的构建
- 怎么确定全体划分点候选集？

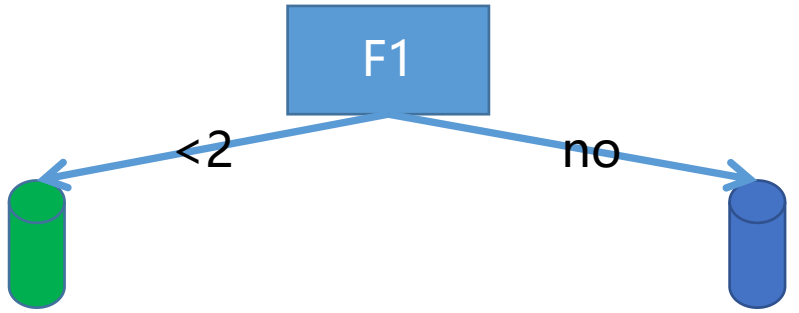
划分候选集

- 连续特征F1有10种取值：1、2、3、4、5、6、7、8、9、10
- 离散特征F2有 4 种取值：A、B、C、D

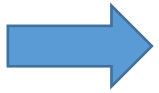
连续特征

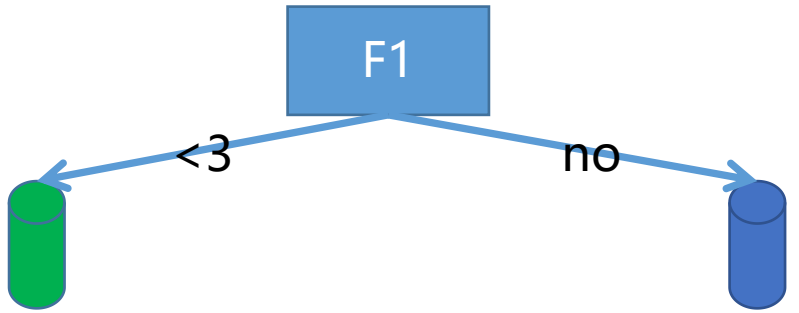
1、|2、3、4、5、6、7、8、9、10



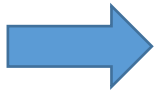


1、2、|3、4、5、6、7、8、9、10



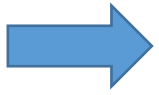


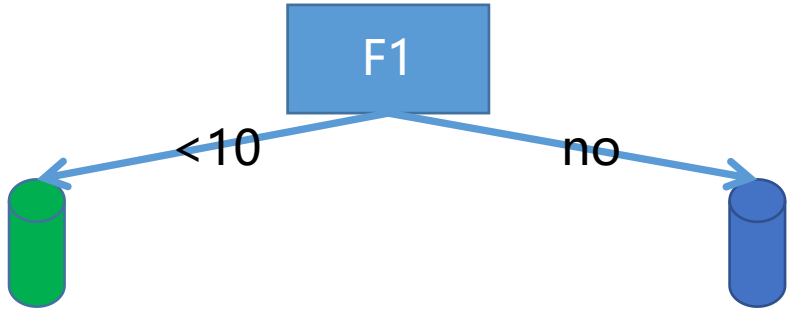
……



……

1、2、3、4、5、6、7、8、9、|10

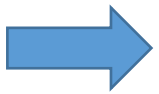


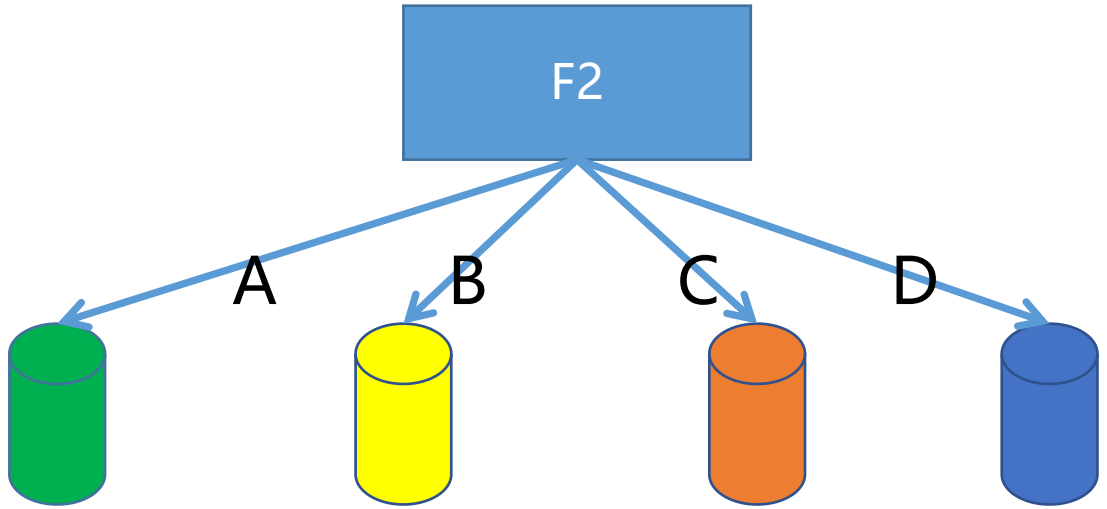


splits={ F1<2, F1<3, F1<4, F1<5, F1<6, F1<7, F1<8, F1<9, F1<10 }

离散特征——多叉树分法

A、B、C、D

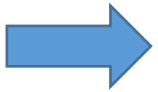


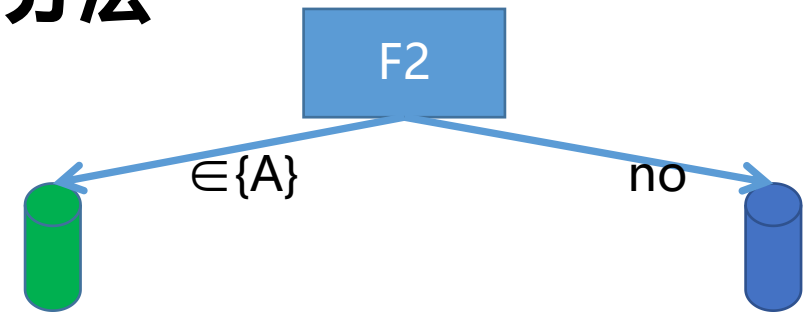


ID3算法采用的是多叉树分法

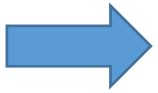
离散特征——二叉树分法

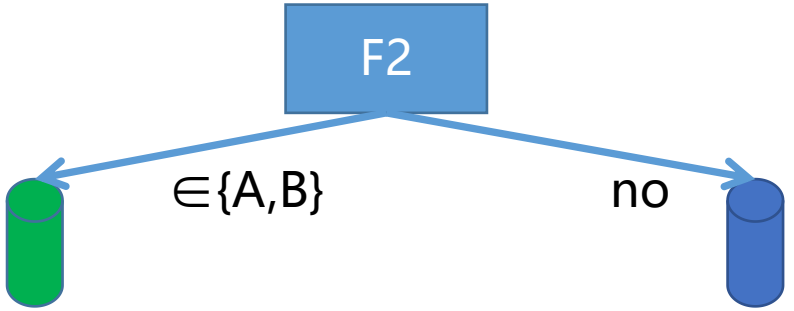
A、|B、C、D



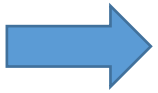


A、B、|C、D



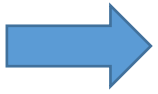


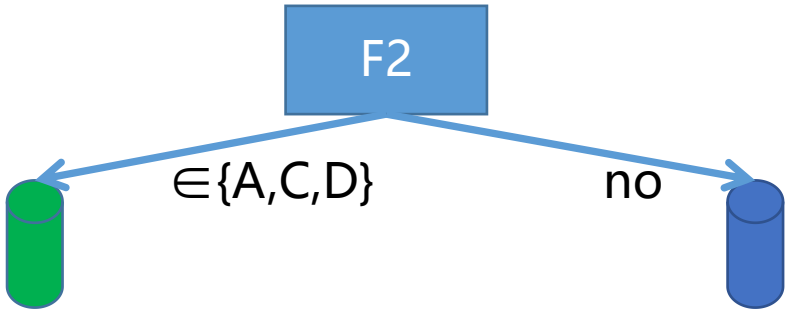
……



…… len(splits)=2^(M-1)-1=7

A、C、D、|B





CART树采用的二叉树分法

splits={ A|.. ， AB|.. ， AC|.. ， AD|.. ， ABC|.. ， ABD|.. ， ACD|.. }

启发式学习的两个问题搞定

- 这个贪心指标怎么设计？
  - 第一步：设计衡量结点不纯度的指标
  - 第二步：设计衡量变纯程度的指标（信息增益，贪心指标）
  - 第三步：遍历全体划分点候选集，选择信息增益最大的划分点构建决策树。反复执行构建树的枝干，直到达到停止条件，完成整棵决策树的构建
- 怎么确定全体划分点候选集？
  - 连续特征：先按照特征值排序，“见缝插针”，5个划分点
  - 离散特征
    - 多叉树：1个划分点
    - 二叉树： 2^(M-1)-1个划分点
  - 连续特征和离散特征划分点取并集

演示建立一颗分类树的过程

- 现有训练集如下，请使用ID3算法训练一个决策树模型，对未来的西瓜的优劣做预测

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

演示建立一颗分类树的过程

- 在决策树开始学习时，根结点包含D中所有样例，|Y|=2，其中正例占p_1=8/17，反例占p_2=9/17。于是根结点的香农熵为：

$$\mathrm{Ent}(D) = -\sum_{k=1}^{2} p_k \log_2 p_k$$

$$= -(\frac{8}{17}\log_2\frac{8}{17} + \frac{9}{17}\log_2\frac{9}{17}) = 0.998$$

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

$$IG(D,s) = Impurity(D) - \frac{N_{left}}{N} Impurity(D_{left}) - \frac{N_{right}}{N} Impurity(D_{right})$$

演示建立一颗分类树的过程

- 全体划分点候选集：{色泽=？，根蒂=？…，触感=？}
- 思考：len(全体划分点候选集)=?
- 先计算“色泽”，根据色泽可以将数据集D分为3个子集：
- D^1：{1,4,6,10,13,17}（正例p1=3/6，反例占p2=3/6）
- D^2：{2,3,7,8,9,15}（正例p1=4/6，反例占p2=2/6）
- D^3：{5,11,12,14,16}（正例p1=1/5，反例占p2=4/5）

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

$$\mathrm{Ent}(D^1) = -(\frac{3}{6}\log_2\frac{3}{6} + \frac{3}{6}\log_2\frac{3}{6}) = 1.000$$

$$\mathrm{Ent}(D^2) = -(\frac{4}{6}\log_2\frac{4}{6} + \frac{2}{6}\log_2\frac{2}{6}) = 0.918$$

$$\mathrm{Ent}(D^3) = -(\frac{1}{5}\log_2\frac{1}{5} + \frac{4}{5}\log_2\frac{4}{5}) = 0.722$$

演示建立一颗分类树的过程

- 计算使用“色泽”划分数据集后的信息增益：

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

$$= 0.998 - (\frac{6}{17}\times 1.000 + \frac{6}{17}\times 0.918 + \frac{5}{17}\times 0.722) = 0.109$$

- 类似地，计算出使用其他属性划分数据集后的信息增益：

$\text{Gain}(D,\text{根蒂}) = 0.143$　　$\text{Gain}(D,\text{敲声}) = 0.141$　　$\text{Gain}(D,\text{触感}) = 0.006$

$\text{Gain}(D,\text{纹理}) = 0.381$　　$\text{Gain}(D,\text{脐部}) = 0.289$

- 显然，选择“纹理”划分后信息增益最大，于是，通过“纹理”划分数据集，各分支结点包含样例子集的情况是：

编程1分类——计算色泽属性的信息增益（离散特征）

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

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

$$= 0.998 - (\frac{6}{17}\times 1.000 + \frac{6}{17}\times 0.918 + \frac{5}{17}\times 0.722) = 0.109$$

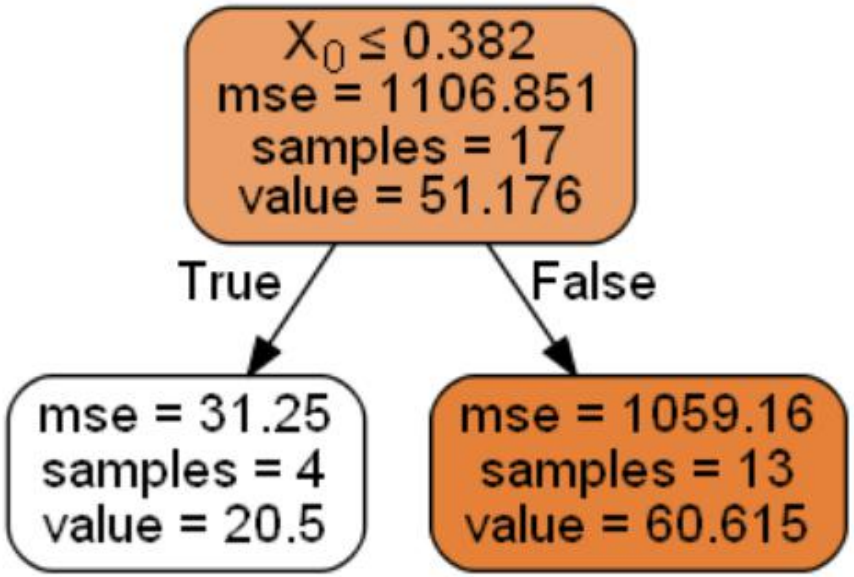
编程2分类——计算密度属性的信息增益（连续特征）

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

$$\text{Gain}(D, \text{密度}) = 0.262$$

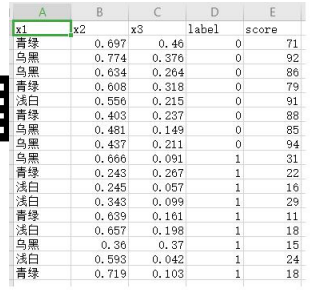
编程3回归——计算密度属性的信息增益（连续特征）

x1	x2	x3	label	score
青绿	0.697	0.46	0	71
乌黑	0.774	0.376	0	92
乌黑	0.634	0.264	0	86
青绿	0.608	0.318	0	79
浅白	0.556	0.215	0	91
青绿	0.403	0.237	0	88
乌黑	0.481	0.149	0	85
乌黑	0.437	0.211	0	94
乌黑	0.666	0.091	1	31
青绿	0.243	0.267	1	22
浅白	0.245	0.057	1	16
浅白	0.343	0.099	1	29
青绿	0.639	0.161	1	11
浅白	0.657	0.198	1	18
乌黑	0.36	0.37	1	15
浅白	0.593	0.042	1	24
青绿	0.719	0.103	1	18



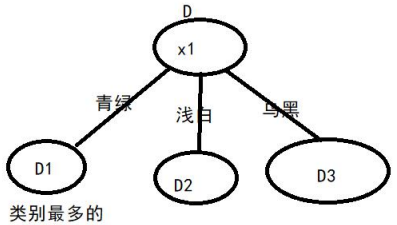
决策树梳理

x1	x2	x3	label	score
青绿	0.697	0.46	0	71
乌黑	0.774	0.376	0	92
乌黑	0.634	0.264	0	86
青绿	0.608	0.318	0	79
浅白	0.556	0.215	0	91
青绿	0.403	0.237	0	88
乌黑	0.481	0.149	0	85
乌黑	0.437	0.211	0	94
乌黑	0.666	0.091	1	31
青绿	0.243	0.267	1	22
浅白	0.245	0.057	1	16
浅白	0.343	0.099	1	29
青绿	0.639	0.161	1	11
浅白	0.657	0.198	1	18
乌黑	0.36	0.37	1	15
浅白	0.593	0.042	1	24
青绿	0.719	0.103	1	18



分类：
不纯度：基尼不纯度（CART分类算法）——二叉树
香农熵（ID算法）——多叉树
IG = 根结点不纯度 - 子结点不纯度的带权和 （信息增益，贪心指标）

候选方案

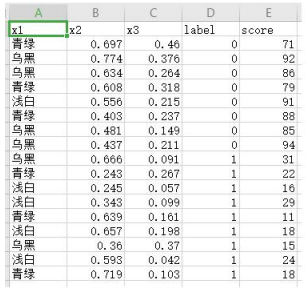


类别最多的

S S S
1 16 16

候选方案数=1+16+16

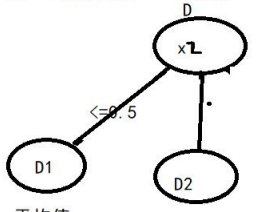
2^(M-1)-1



分类：
不纯度：基尼不纯度（CART分类算法）——二叉树
香农熵（ID算法）——多叉树
IG = 根结点不纯度 - 子结点不纯度的带权和 （信息增益，贪心指标）

回归：
不纯度：均方差 CART回归——二叉树

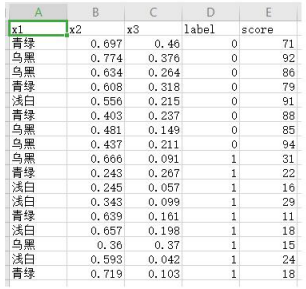
候选方案



平均值

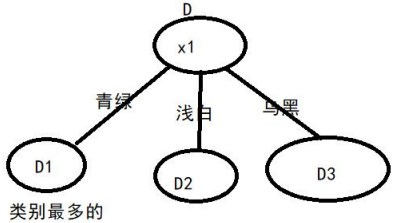
S S S
2^(M-1)-1 16 16

候选方案数=1+16+16



分类：
不纯度：基尼不纯度（CART分类算法）——二叉树
香农熵（ID算法）——多叉树
IG = 根结点不纯度 - 子结点不纯度的带权和 （信息增益，贪心指标）

候选方案



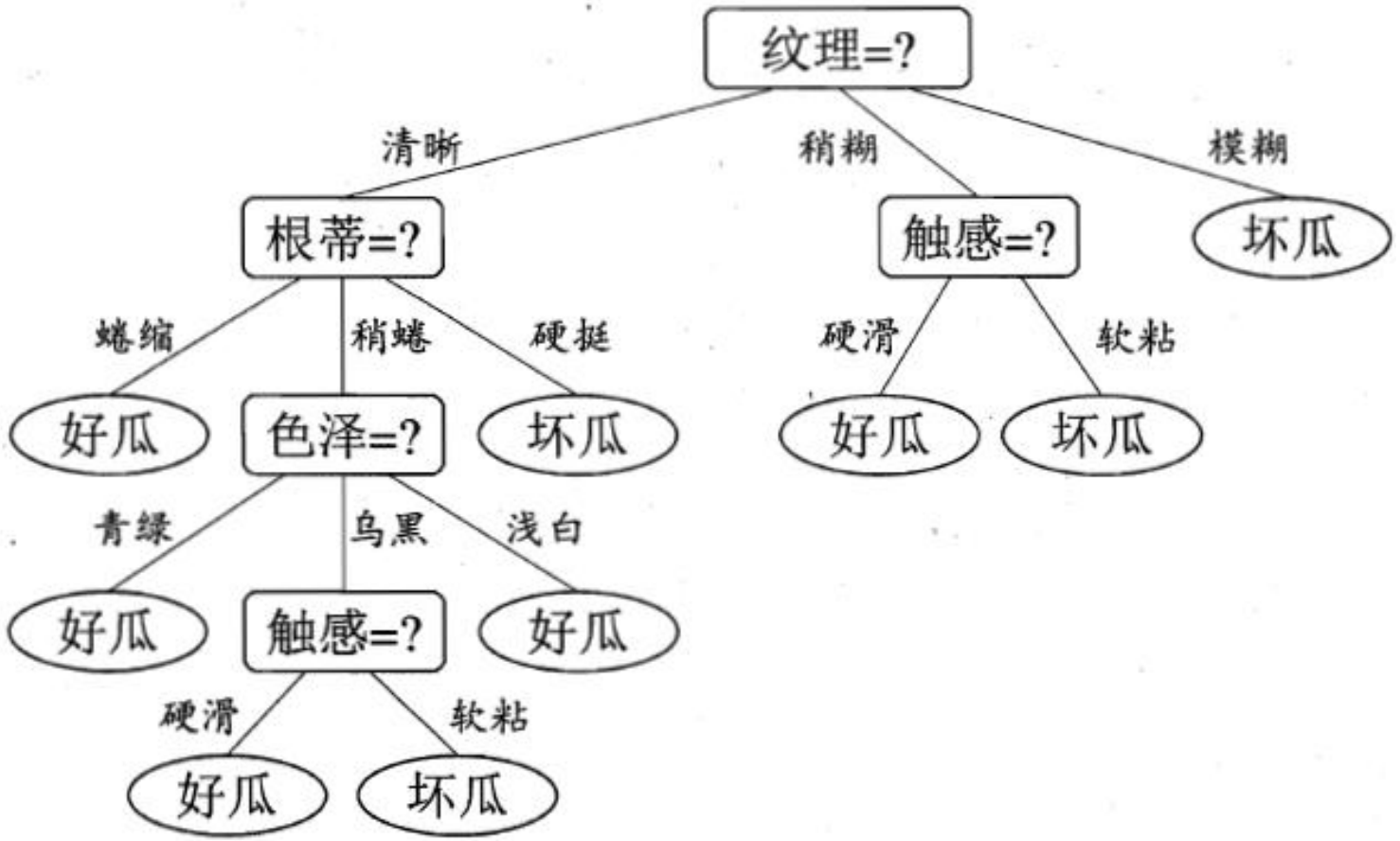
类别最多的

S S S
1 16 16

候选方案数=1+16+16

演示建立一颗分类树的过程

- 每次建树之后，按照特征值将样本分配到对应结点，重复，便可得到完整的决策树



构建决策树伪代码

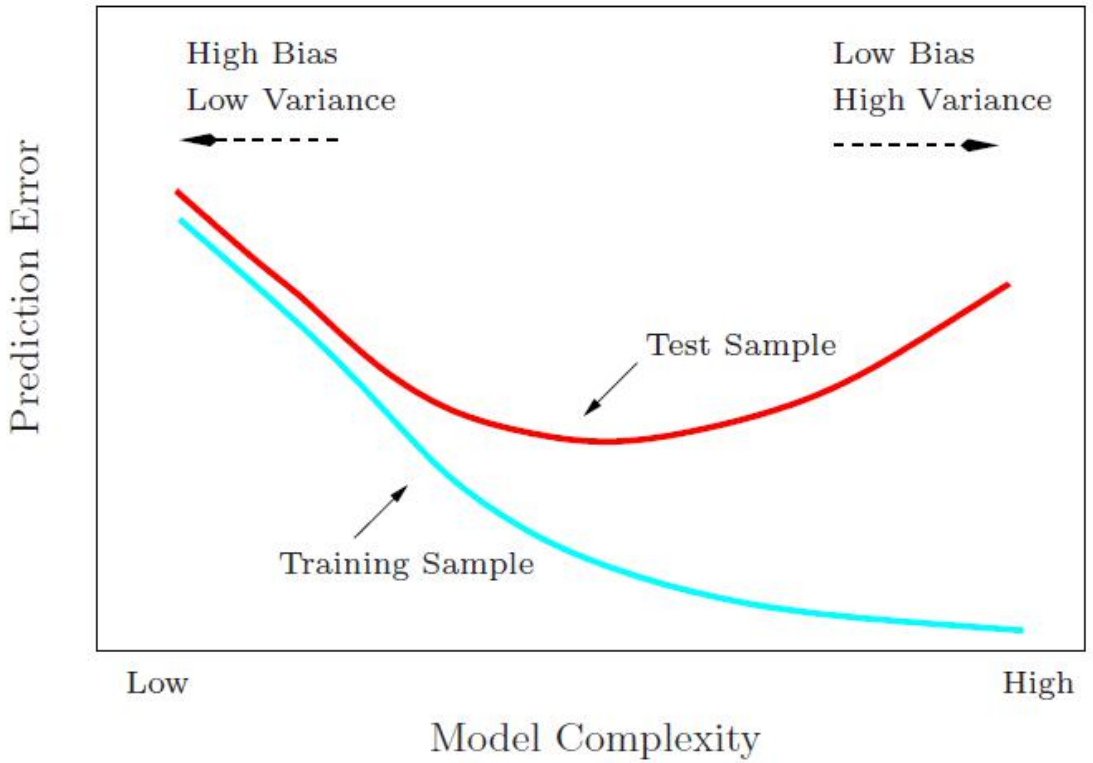
```
def buildTree：
        初始化根结点
        确定全体划分点候选集
        for //反复建树
                for (i<-splits){ //试探性结点分裂
                        按照特征将样本转移到对应子结点
                        记录划分点和对应的信息增益
                }
                选择信息增益最大的划分点
                进行结点分裂
                if 满足停止构造的条件
                        break
        return 树
```

决策树的剪枝

- 为什么要剪枝?
  - 宽泛来讲：机器学习模型都会有过拟合风险，在训练集上表现的过于优秀，而泛化能力差，因此机器学习模型都有对应的防止过拟合的策略，决策树的策略就是剪枝
  - 具体来讲：决策树生成算法会递归地建造决策树的分支，直到结点都很纯为止。这样产生的决策树往往对训练数据的分类很准确，但失去泛化能力，即发生过拟合了，因此，可通过主动去掉一些分支来降低过拟合的风险，提高模型泛化能力

关于过拟合的思考

- 如何判断你的模型是否过拟合？
- 答：可以看训练误差和测试误差的关系，测试误差越小，说明模型泛化能力越强。



剪枝方法

- 周志华
  - 预剪枝：在决策树生成过程中，考察该轮划分是否会提高泛化性能，如果不能提高泛化性能就不划分
    - 优点：性能开销小、速度快
    - 缺点：可能会欠拟合，错过最优解
  - 后剪枝：先训练一棵完整的树，然后自底向上地进行考察，如果删除该划分能提高泛化性能就删除
    - 优点：欠拟合风险小
    - 缺点：性能开销大、速度慢

剪枝方法

- 李航
  - 构建了一个目标函数：代价函数 + 正则化项
  - 效果：类似于后剪枝

$$C_\alpha(T) = C(T) + \alpha |T| \tag{5.14}$$

式(5.14)中，$C(T)$表示模型对训练数据的预测误差，即模型与训练数据的拟合程度，$|T|$表示模型复杂度，参数$\alpha \geq 0$控制两者之间的影响．较大的α促使选择较简单的模型（树），较小的α促使选择较复杂的模型（树）．$\alpha = 0$意味着只考虑模型与训练数据的拟合程度，不考虑模型的复杂度．

剪枝方法

- Spark
  - ①树的深度等于超参数maxDepth
  - ②没有分裂点能带来大于minInfoGain的信息增益
  - ③没有分裂点能让划分后的结点上至少有minInstances个实例
  - 效果：类似于？剪枝
  - 优点：性能开销小，求解速度快
  - 缺点：可能会欠拟合

缺失值处理

- 可以使用之前的方法
- 决策树独有的一种方法：在周志华的书中有对于缺失值处理的算法（课后作业）

决策树模型优缺点

- **优点：**
  - 易于解释
  - 处理类别特征，其他的技术往往要求数据属性的单一
  - 延展到多分类
  - 不需要特征放缩
  - 能捕获非线性关系和特征间的影响
- **缺点：**
  - 寻找最优的决策树是一个NP-hard的问题，只能通过启发式方法求次优解
  - 决策树会因为样本发生一点点的改动，就会导致树结构的剧烈改变
  - 如果某些离散特征的特征值种类多，生成决策树容易偏向于这些特征
  - 有些比较复杂的关系，决策树很难学习，比如异或

编程——决策树综合案例之森林植被类型预测

- 数据集：
  - https://archive.ics.uci.edu/ml/datasets/covertype
- 解释：
  - 该数据集记录了美国科罗拉多州不同地块的森林植被类型。每个样本包含了描述每块土地的若干特征，包括海拔、坡度、到水源的距离、遮阳情况和土壤类型，并且随同给出了地块的已知森林植被类型。我们需要总共54 个特征中的其余各项来预测森林植被类型

Data Set Characteristics:	Multivariate	Number of Instances:	581012	Area:	Life
Attribute Characteristics:	Categorical, Integer	Number of Attributes:	54	Date Donated	1998-08-01
Associated Tasks:	Classification	Missing Values?	No	Number of Web Hits:	185453

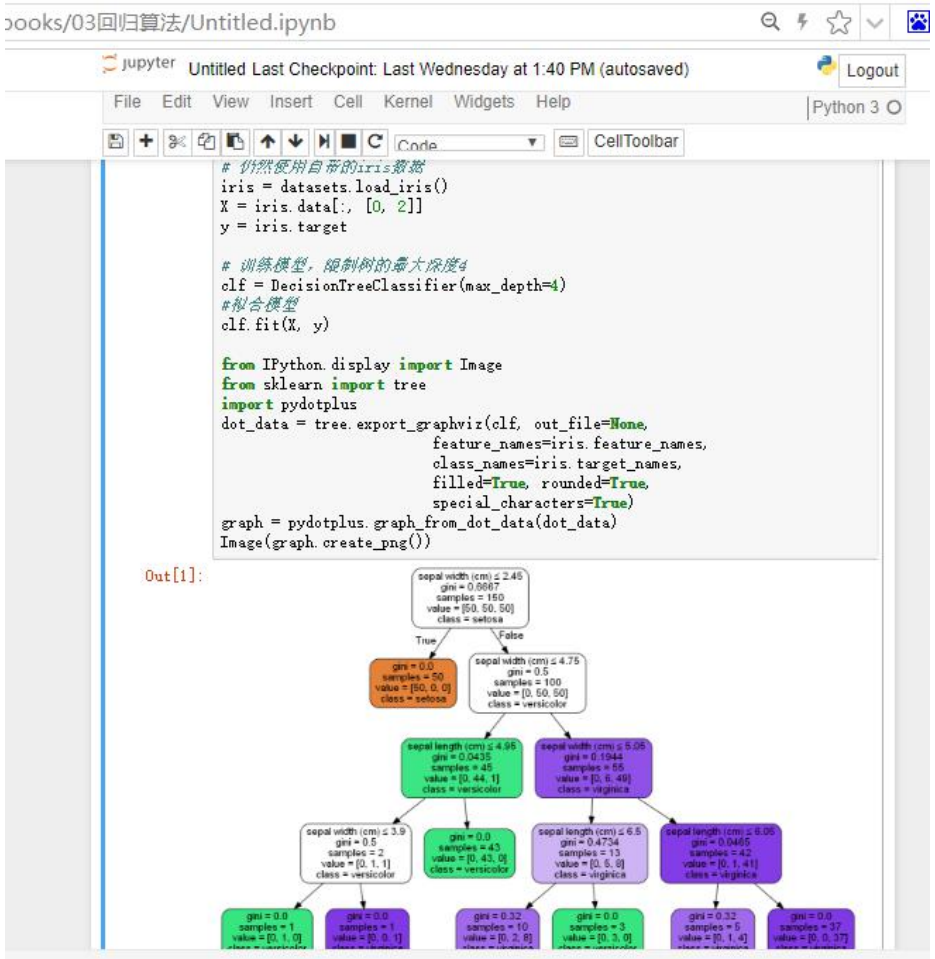
编程——决策树回归案例之共享单车租赁数量预测

- This dataset contains the hourly and daily count of rental bikes between years 2011 and 2012 in Capital bikeshare system with the corresponding weather and seasonal information.
  - 数据下载 http://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset

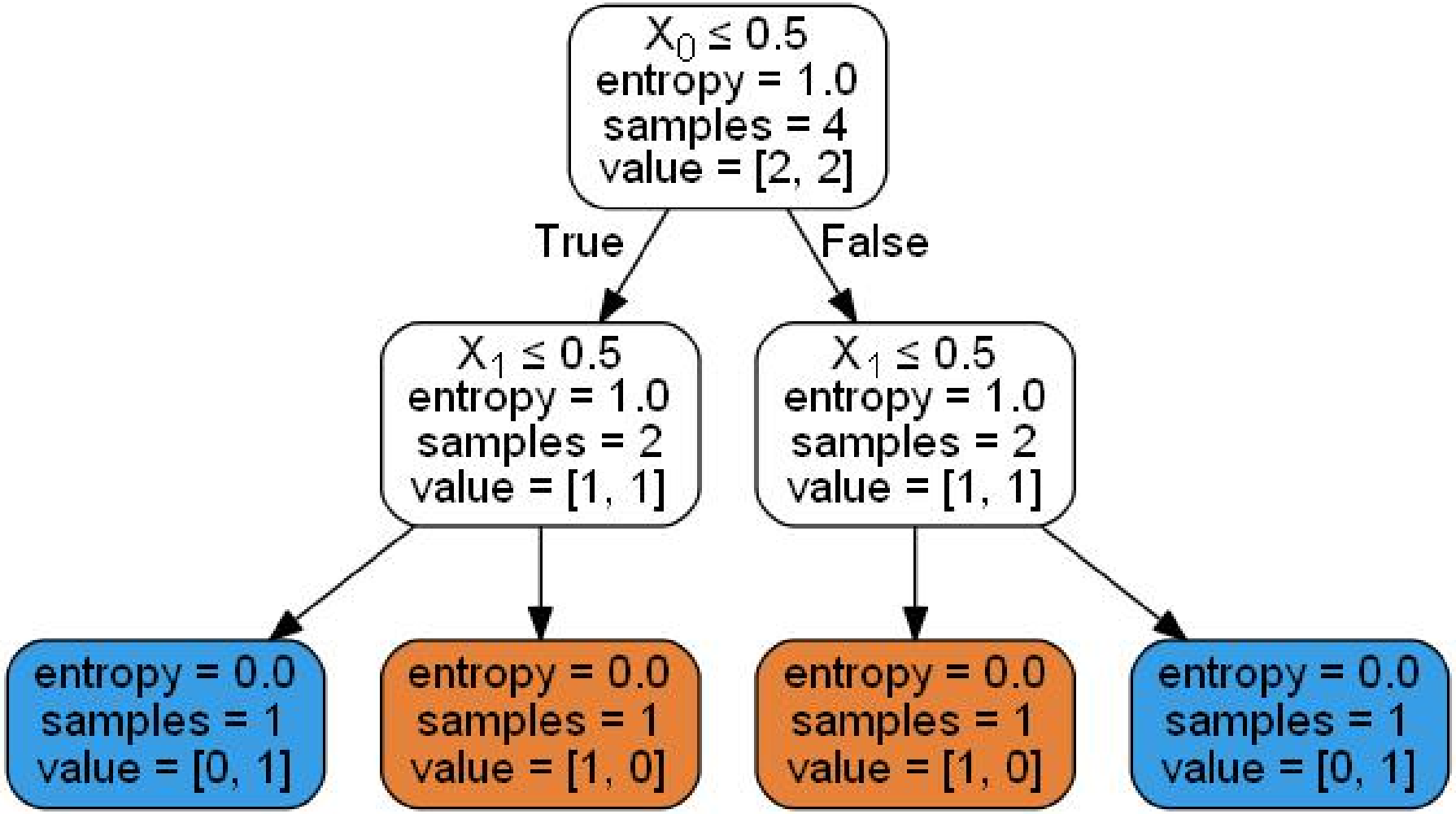
Data Set Characteristics:	Univariate	Number of Instances:	17389	Area:	Social
Attribute Characteristics:	Integer, Real	Number of Attributes:	16	Date Donated	2013-12-20
Associated Tasks:	Regression	Missing Values?	N/A	Number of Web Hits:	232895

决策树可视化

- 下载graphviz-2.38安装并配置环境变量
- 安装pydotplus包，建议离线安装
  - https://pypi.org/project/pydotplus/#files
- 重启jupyter



决策树异或


$X_0 \leq 0.5$
entropy = 1.0
samples = 4
value = [2, 2]
True
False
$X_1 \leq 0.5$
entropy = 1.0
samples = 2
value = [1, 1]
$X_1 \leq 0.5$
entropy = 1.0
samples = 2
value = [1, 1]
entropy = 0.0
samples = 1
value = [0, 1]
entropy = 0.0
samples = 1
value = [1, 0]
entropy = 0.0
samples = 1
value = [1, 0]
entropy = 0.0
samples = 1
value = [0, 1]

THANK YOU

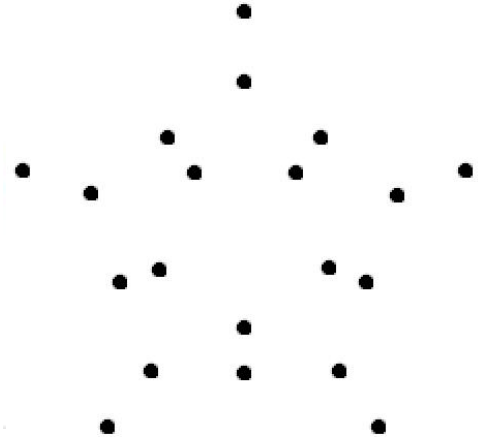
上海育创网络科技有限公司

P问题、NP问题、NPC问题、NP-Hard问题

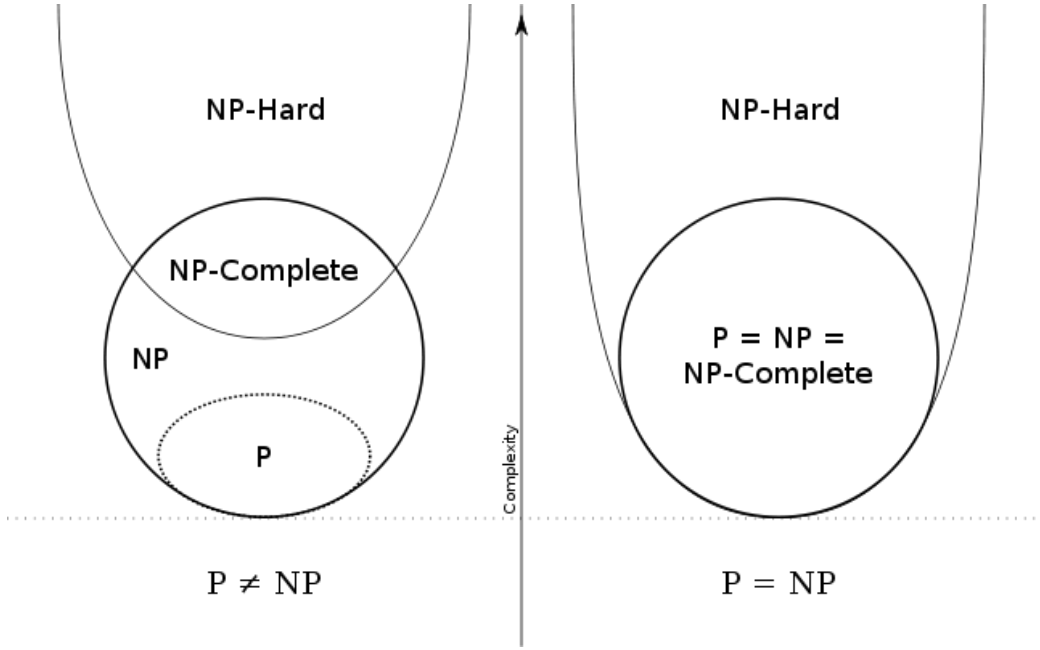
- 时间复杂度
  - 介绍：它不是衡量一个程序解决问题需要花多少时间的；而是衡量当问题规模扩大后，程序求得结果需要的时间增长得有多快的。
  - 多项式级的复杂度：
    - 包含：O(1)，O(n)，O(n^k)，O(lg(n))，O(nlg(n))
    - 举例：打印dict[“小明”] =>O(1)；打印List中的一个“小明”元素 =>O(n)。
  - 非多项式级的复杂度：
    - 包含：O(k^n)，O(n!)
    - 举例：打印n个数的全排列 =>O(n!)。

P问题、NP问题、NPC问题、NP-Hard问题

- P问题（Polynomial Problem）
  - 如果一个问题的求解复杂度是多项式级的，就称这个问题是P问题
  - 举例：打印一个List中的最大值
- NP问题（Non-Deterministic Polynomial Problem）
  - 不是非P类问题（N不是not的意思）
  - 如果一个问题的求解复杂度不能确定是多项式级别的，但是能在多项式时间内验证某个答案是不是该问题的一个解，就称这个问题是NP问题
  - 举例：哈密尔顿回路，给一堆点，问能否找到一条经过每个点一次且恰好一次（不遗漏也不重复）最后又返回到出发点的路

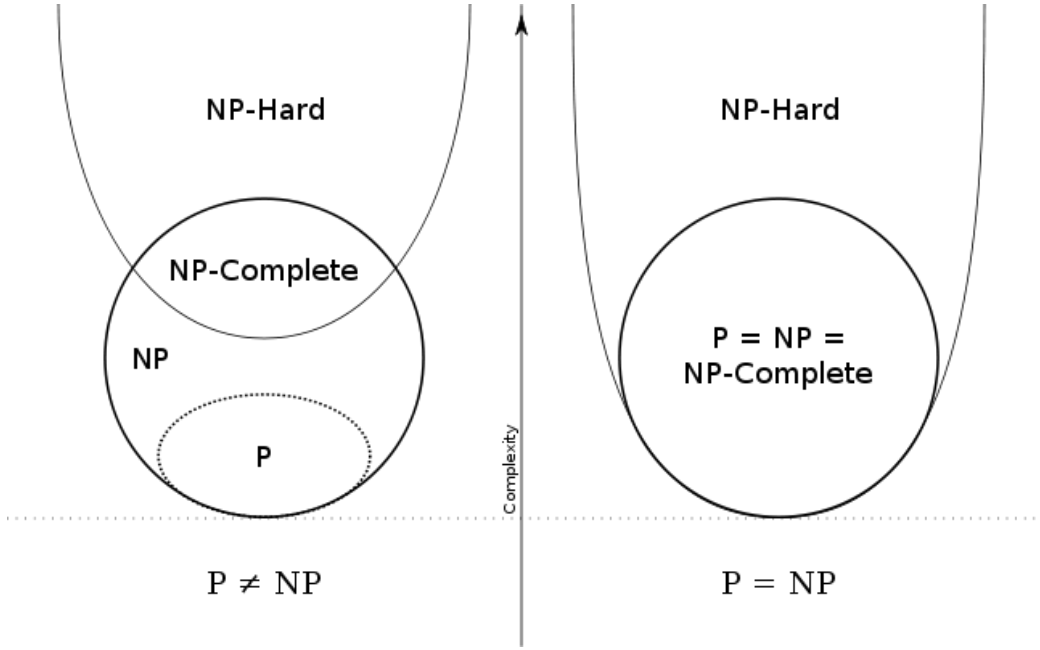


P问题、NP问题、NPC问题、NP-Hard问题



- 因此，科学家们极力地探索NP问题是否=P问题，即NP=P？（NP问题能否在多项式时间内求解），如果能证明NP=P那么计算机就能帮我们做更多的任务了。
- 美国Clay数学研究所于2000年在巴黎法兰西学院宣布：对七个“千僖年数学难题”进行悬赏，其中“千僖难题”的首位就是“NP=P?”，足见其显赫地位和无穷魅力。

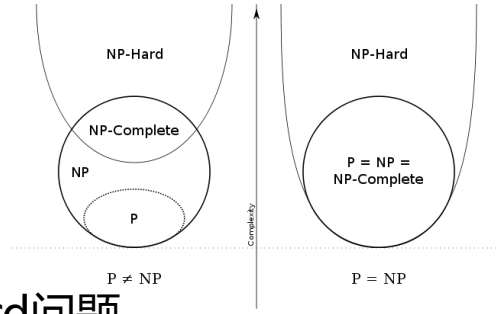
P问题、NP问题、NPC问题、NP-Hard问题



- 遗憾的是，至今没有人能证明NP=P；反而人们开始相信， NP=P不成立，即存在至少一个不可能有多项式级复杂度的算法的NP问题。人们如此坚信NP ≠P是有原因的，就是在研究NP问题的过程中找出了一类非常特殊的NP问题叫做 NPC问题（ Non-deterministic Polynomial complete problem ）， NPC问题的特点是**不可能有多项式级复杂度的解法。**

P问题、NP问题、NPC问题、NP-Hard问题

- NPC问题：
  - 如果一个问题是NP问题，并且所有的NP问题都可以在多项式时间内归约到它，那它就是NPC问题。
  - 举例：背包问题、哈密尔顿回路问题、旅行推销员问题、子图同构问题等 https://baike.baidu.com/item/NPC%E9%97%AE%E9%A2%98/8698778



- NP-hard问题：
  - 如果一个问题（不一定是NP问题），所有的NP问题都可以归约到它，那它就是NP-hard问题。
  - P问题属于NP问题，NPC问题属于NP问题；NPC问题同时属于NP-hard问题，是NP与NP-hard的交集
- 归约
  - 如果问题A的求解难度≤问题B的求解难度，那么问题A就能归约到问题B；比如：解一元一次方程的问题可以归约到解一元二次方程的问题。
  - 为了证明某个NP问题A实际上是NPC问题，证明者必须用一个已知的NPC问题归约到A。
- 注意
  - 当强调一个问题只有非多项式级复杂度解法时，不要说它是一个NP问题，应该说它是一个NP-hard问题

假设离散特征值有{A,B,C,D}

A|...	AB|...	...|A
B|...	AC|...	...|B
C|...	AD|...	...|C
D|...	BC|...	...|D
	BD|...	
	CD|...	

划分方案数S正好多了2倍，因此除以2，即

$$S = (C_4^1 + C_4^2 + C_4^3)/2 = (4+6+4)/2 = 7$$

假设有m个特征值，划分方案数 $S = (C_m^1 + C_m^2 + \cdots + C_m^{m-1})/2$

根据公式 $(a+b)^m = C_m^0 a^m b^0 + C_m^1 a^{m-1} b^1 + \cdots + C_m^m a^0 b^m$

则 $(1+1)^m = C_m^0 + C_m^1 + \cdots + C_m^m$

因此 $S = (2^m - C_m^0 - C_m^m)/2 = (2^m - 2)/2 = 2^{m-1} - 1$