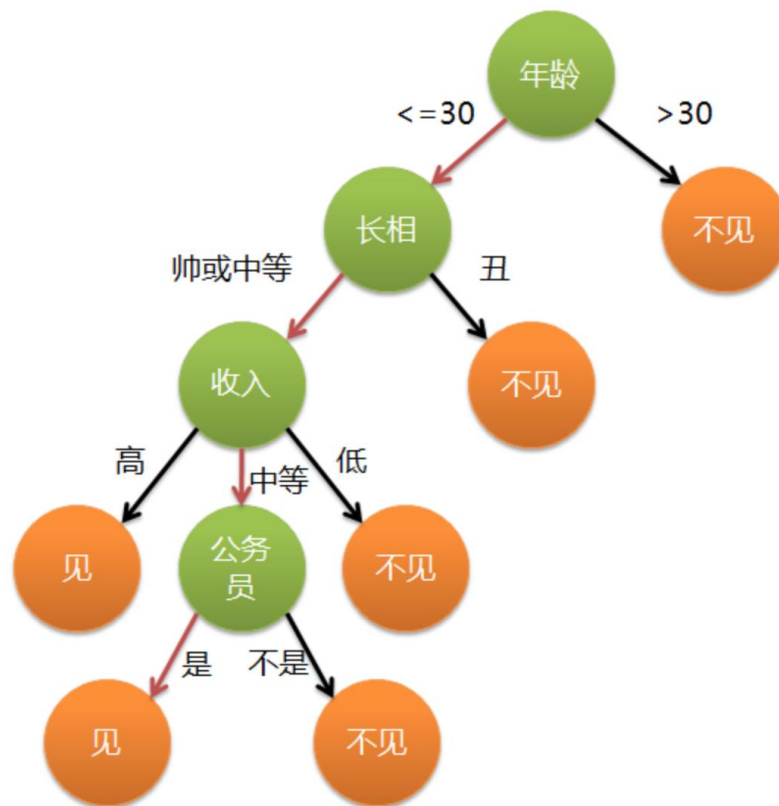


决策树

Decision Tree



比较适合分析离散数据。
如果是连续数据要先转成离散数据再做分析。





70年代后期至80年代，Quinlan开发了ID3算法。

Quinlan改进了ID3算法，称为C4.5算法。

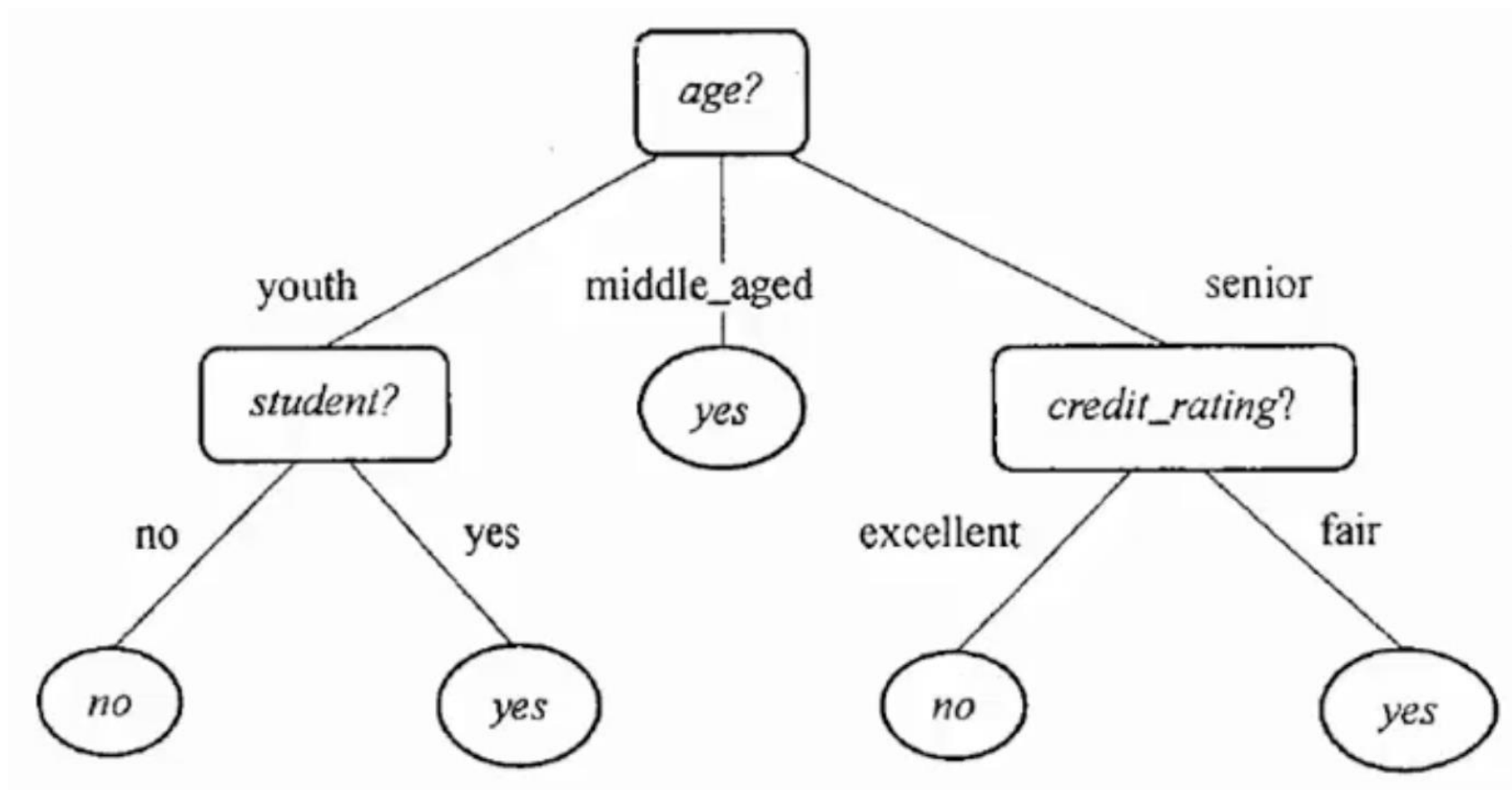
1984年，多位统计学家提出了CART算法。

例子



<i>RID</i>	<i>age</i>	<i>income</i>	<i>student</i>	<i>credit_rating</i>	<i>Class: buys_computer</i>
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

期待输出的结果



熵(entropy)概念



1948年，香农提出了“信息熵”的概念。

一条信息的信息量大小和它的不确定性有直接的关系，要搞清楚一件非常非常不确定的事情，或者是我们一无所知的事情，需要了解大量信息->信息量的度量就等于不确定性的多少。



信息熵公式：

$$H[x] = - \sum_x p(x) \log_2 p(x)$$

假如有一个普通骰子A，扔出1-6的概率都是1/6

有一个骰子B，扔出6的概率是50%，扔出1-5的概率都是10%

有一个骰子C，扔出6的概率是100%。



$$\text{骰子A} : -\left(\frac{1}{6} \times \log_2 \frac{1}{6}\right) \times 6 \approx 2.585$$

$$\text{骰子B} : -\left(\frac{1}{10} \times \log_2 \frac{1}{10}\right) \times 5 - \frac{1}{2} \times \log_2 \frac{1}{2} \approx 2.161$$

$$\text{骰子C} : -(1 \times \log_2 1) = 0$$



决策树会选择最大化信息增益来对结点进行划分。
信息增益计算：

$$Info(D) = - \sum_{i=1}^m p_i \log_2(p_i)$$

$$Info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \times Info(D_j)$$

$$Gain(A) = Info(D) - Info_A(D)$$

选择根节点-ID3算法



信息增益(Information Gain) : $\text{Gain}(A) = \text{Info}(D) - \text{Infor_A}(D)$

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

$$\text{Info}(D) = -\frac{9}{14} \log_2 \left(\frac{9}{14} \right) - \frac{5}{14} \log_2 \left(\frac{5}{14} \right) = 0.940 \text{ bits.}$$

$$\begin{aligned} \text{Info}_{age}(D) &= \frac{5}{14} \times \left(-\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} \right) \\ &\quad + \frac{4}{14} \times \left(-\frac{4}{4} \log_2 \frac{4}{4} - \frac{0}{4} \log_2 \frac{0}{4} \right) \\ &\quad + \frac{5}{14} \times \left(-\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} \right) \\ &= 0.694 \text{ bits.} \end{aligned}$$

$$\text{Gain}(age) = \text{Info}(D) - \text{Info}_{age}(D) = 0.940 - 0.694 = 0.246 \text{ bits.}$$

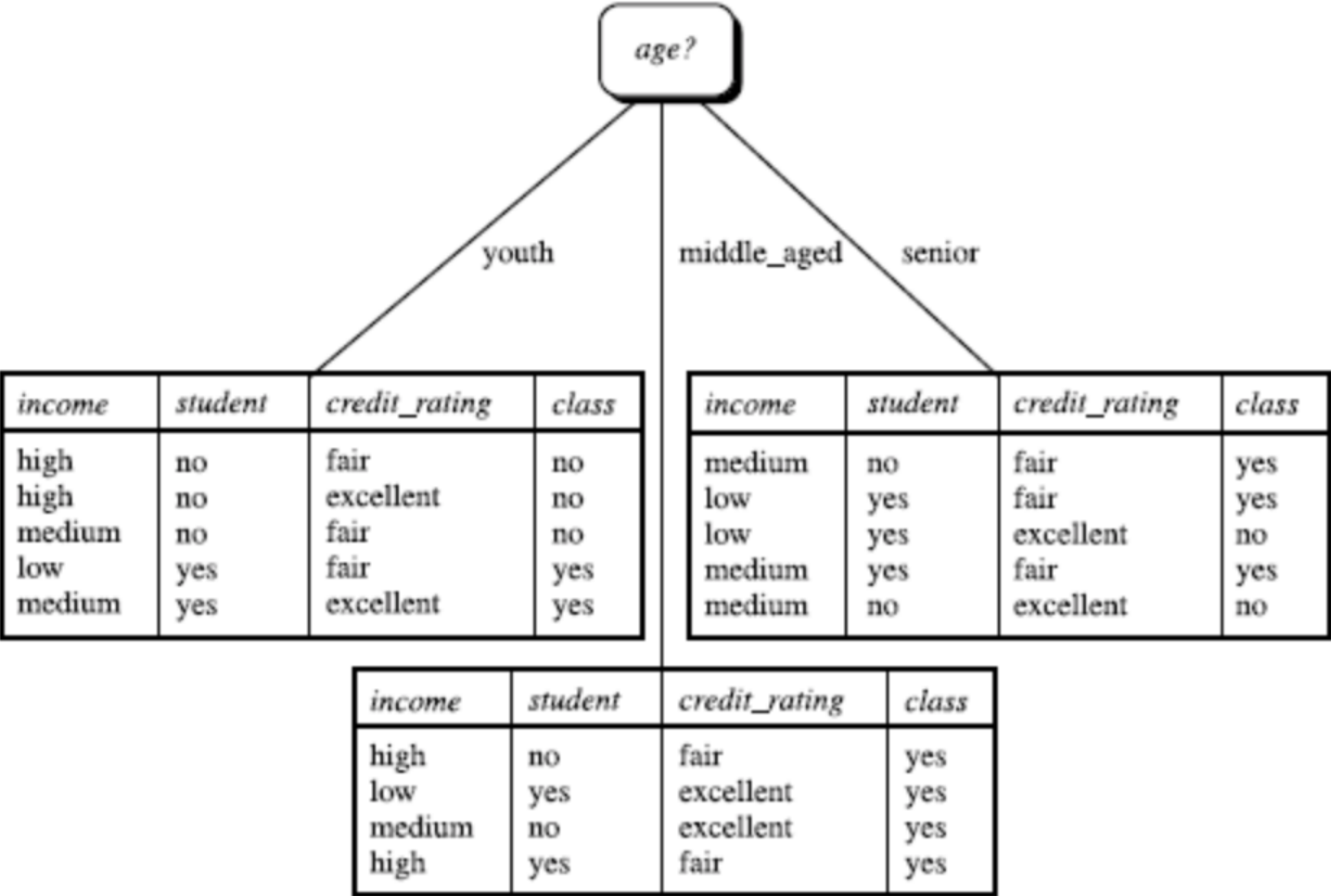
类似：

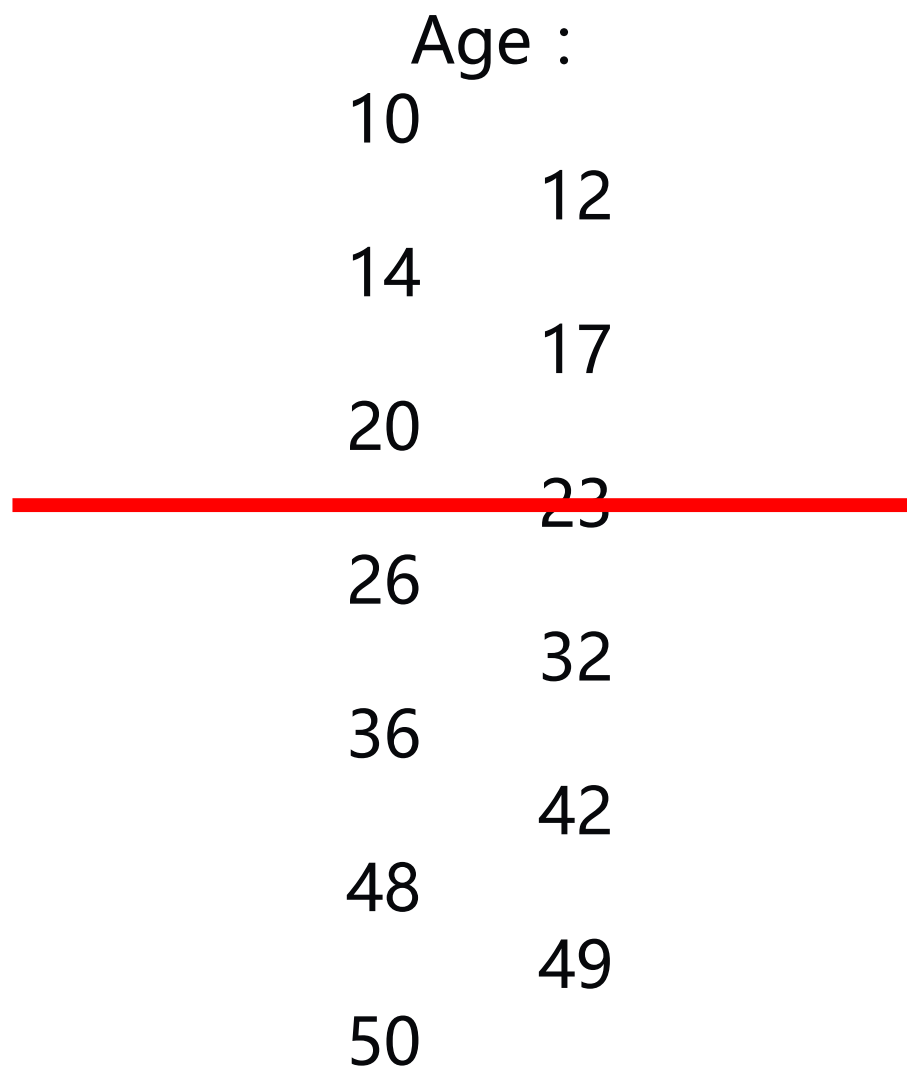
$$\text{Gain}(\text{income}) = 0.029,$$

$$\text{Gain}(\text{student}) = 0.151,$$

$$\text{Gain}(\text{credit_rating}) = 0.048$$

选择根节点-ID3算法







信息增益的方法倾向于首先选择因子数较多的变量
信息增益的改进：增益率

$$SplitInfo_A(D) = - \sum_{j=1}^v \frac{|D_j|}{|D|} \times \log_2 \left(\frac{|D_j|}{|D|} \right)$$

$$GainRate(A) = \frac{Gain(A)}{SplitInfo_A(D)}$$

决策树-例子





CART决策树的生成就是递归地构建二叉决策树的过程。
CART用基尼(Gini)系数最小化准则来进行特征选择，生成二叉树。

Gini系数计算：

$$Gini(D) = 1 - \sum_{i=1}^m p_i^2$$

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

$$\Delta Gini(A) = Gini(D) - Gini_A(D)$$



分别计算它们的Gini系数增益，取Gini系数增益值最大的属性作为决策树的根节点属性。根节点的Gini系数：

$$Gini(\text{是否拖欠贷款}) = 1 - \left(\frac{3}{10}\right)^2 - \left(\frac{7}{10}\right)^2 = 0.42$$

序号	是否有房	婚姻状况	年收入	是否拖欠贷款
1	yes	single	125K	no
2	no	married	100K	no
3	no	single	70K	no
4	yes	married	120K	no
5	no	divorced	95K	yes
6	no	married	60K	no
7	yes	divorced	220K	no
8	no	single	85K	yes
9	no	married	75K	no
10	no	single	90K	yes



根据是否有房来进行划分时，Gini系数增益计算：
(左子节点代表yes，右子节点代表no)

$$Gini(\text{左子节点}) = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2 = 0$$

$$Gini(\text{右子节点}) = 1 - \left(\frac{3}{7}\right)^2 - \left(\frac{4}{7}\right)^2 = 0.4898$$

$$\Delta\{\text{是否有房}\} = 0.42 - \frac{7}{10} \times 0.4898 - \frac{3}{10} \times 0 = 0.077$$

	是否拖欠贷款
Yes	3
No	7

		是否有房	
		N1(Yes)	N2(No)
是否拖欠贷款	Yes	0	3
	No	3	4



根据婚姻状况来进行划分时，Gini系数增益计算：

- {married} | {single,divorced}
- {single} | {married,divorced}
- {divorced} | {single,married}

当分组为{married} | {single,divorced}时：

$$\Delta\{\text{婚姻状况}\} = 0.42 - \frac{4}{10} \times 0 - \frac{6}{10} \times \left[1 - \left(\frac{3}{6}\right)^2 - \left(\frac{3}{6}\right)^2\right] = 0.12$$

当分组为{single} | {married,divorced} 时：

$$\Delta\{\text{婚姻状况}\} = 0.42 - \frac{4}{10} \times 0.5 - \frac{6}{10} \times \left[1 - \left(\frac{1}{6}\right)^2 - \left(\frac{5}{6}\right)^2\right] = 0.053$$

当分组为{divorced} | {single,married} 时：

$$\Delta\{\text{婚姻状况}\} = 0.42 - \frac{2}{10} \times 0.5 - \frac{8}{10} \times \left[1 - \left(\frac{2}{8}\right)^2 - \left(\frac{6}{8}\right)^2\right] = 0.02$$



根据年收入来进行划分时，Gini系数增益计算：

是否拖欠贷款	no	no	no	yes	yes	yes	no	no	no	no
年收入	60	70	75	85	90	95	100	120	125	220
相邻值中点	65	72.5	80	87.7	92.5	97.5	110	122.5	172.5	
Gini 系数增益	0.02	0.045	0.077	0.003	0.02	0.12	0.077	0.045	0.02	

例如当面对年收入为60和70这两个值时，我们算得其中间值为65。倘若以中间值65作为分割点，于是则得Gini系数增益为：

$$\Delta(\text{年收入}) = 0.42 - \frac{1}{10} \times 0 - \frac{9}{10} \times \left[1 - \left(\frac{6}{9}\right)^2 - \left(\frac{3}{9}\right)^2 \right] = 0.02$$



根据计算知道，三个属性划分根节点的增益最大的有两个：年收入属性和婚姻状况，他们的增益都为0.12。可以随机选择一个作为根结点。如假我们选择婚姻状况作为根结点。接下来，使用同样的方法，分别计算剩下的属性，其中根结点的Gini系数为：

$$Gini(\text{是否拖欠贷款}) = 1 - \left(\frac{3}{6}\right)^2 - \left(\frac{3}{6}\right)^2 = 0.5$$

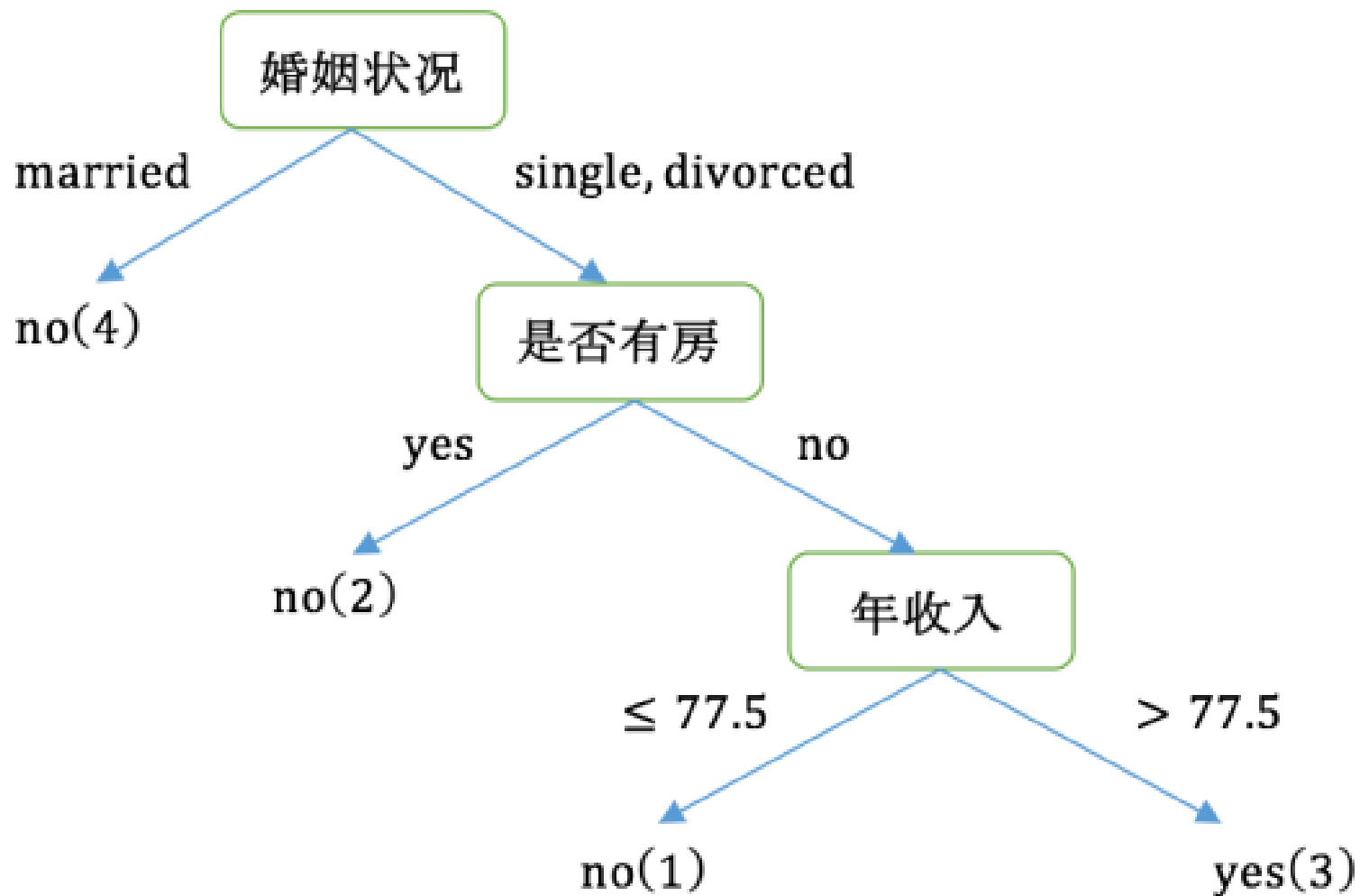
与前面的计算过程类似，对于是否有房属性，可得：

$$\Delta\{\text{是否有房}\} = 0.5 - \frac{4}{6} \times \left[1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2\right] - \frac{2}{6} \times 0 = 0.25$$

对于年收入属性则有：

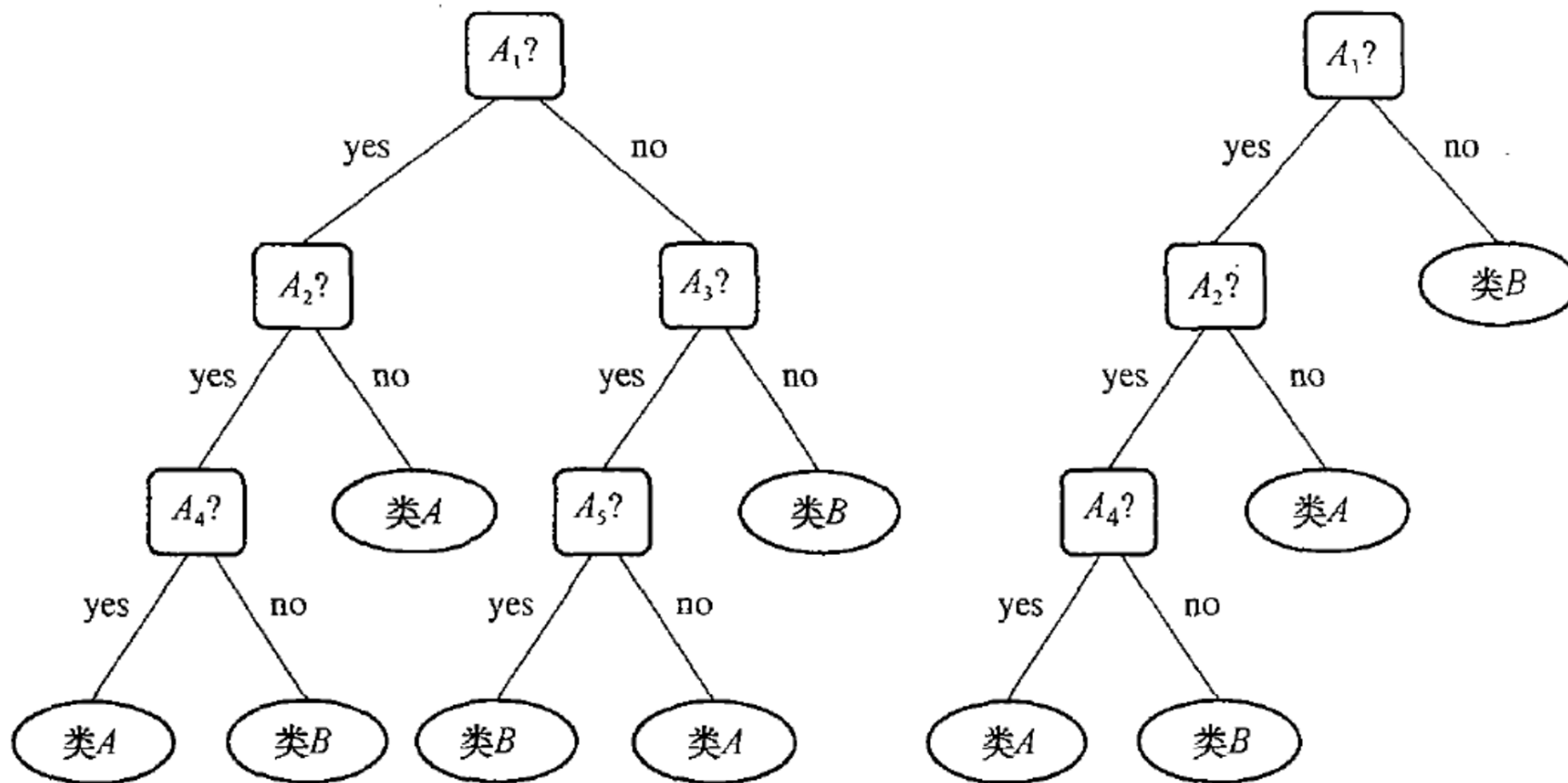
是否拖欠贷款	no	yes	yes	yes	no	no
年收入	70	85	90	95	125	220
相邻值中点	77.5	87.7	92.5	110	172.5	
Gini 系数增益	0.1	0.25	0.05	0.25	0.1	

最后构建的CART





预剪枝后剪枝





优点：

小规模数据集有效

缺点：

处理连续变量不好

类别较多时，错误增加的比较快

不能处理大量数据

决策树

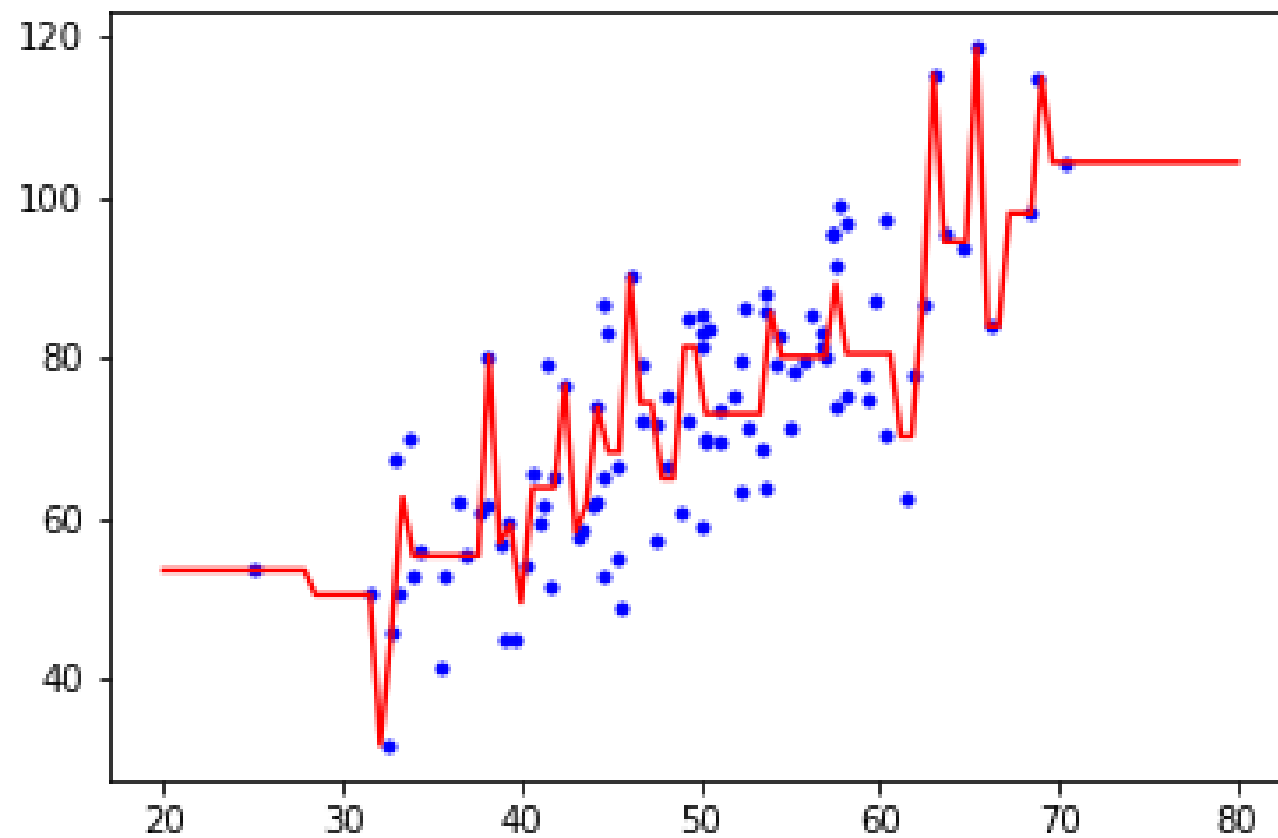


决策树-线性二分类



决策树-非线性二分类





回归树



回归树-预测房价





「关注 AI MOOC 官方公众号」

这里持续分享
Python、机器学习和深度学习的
各种资源干货及
有趣又实用的硬知识



覃秉丰

AI MOOC创始人

网易计算机视觉微专业核心讲师

机器学习、深度学习多年开发经验

受邀为中国移动、国家电网、华夏银行、
太平洋保险等世界五百强企业总部做AI内训

