数据挖掘

第5章 决策树分类

2024 年 10 月

南京大学
NANJING UNIVERSITY

目录

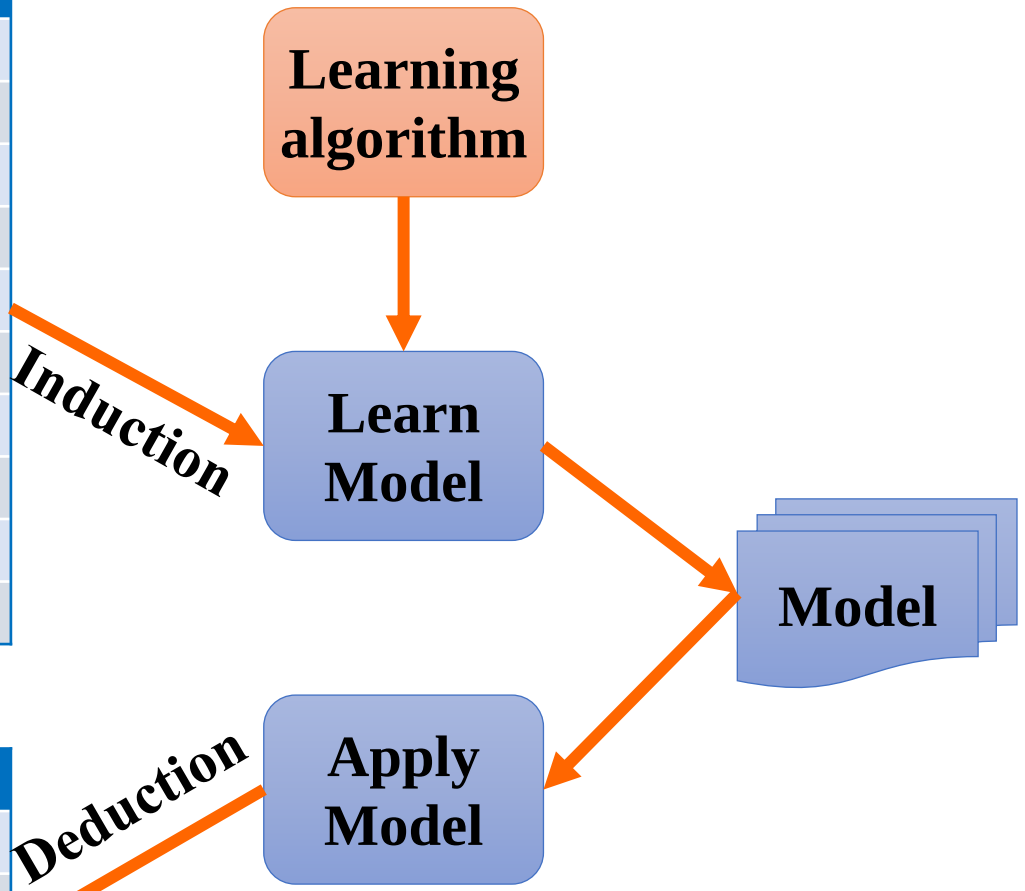
分类

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



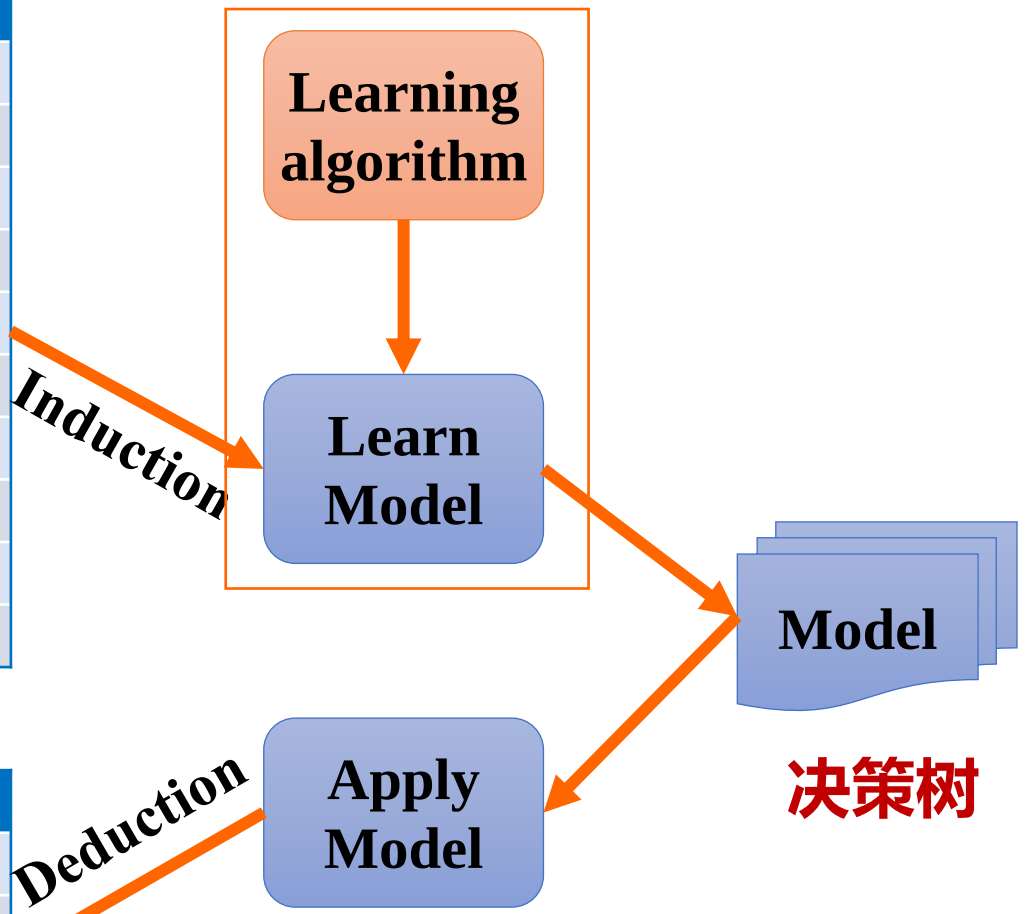
决策树分类

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



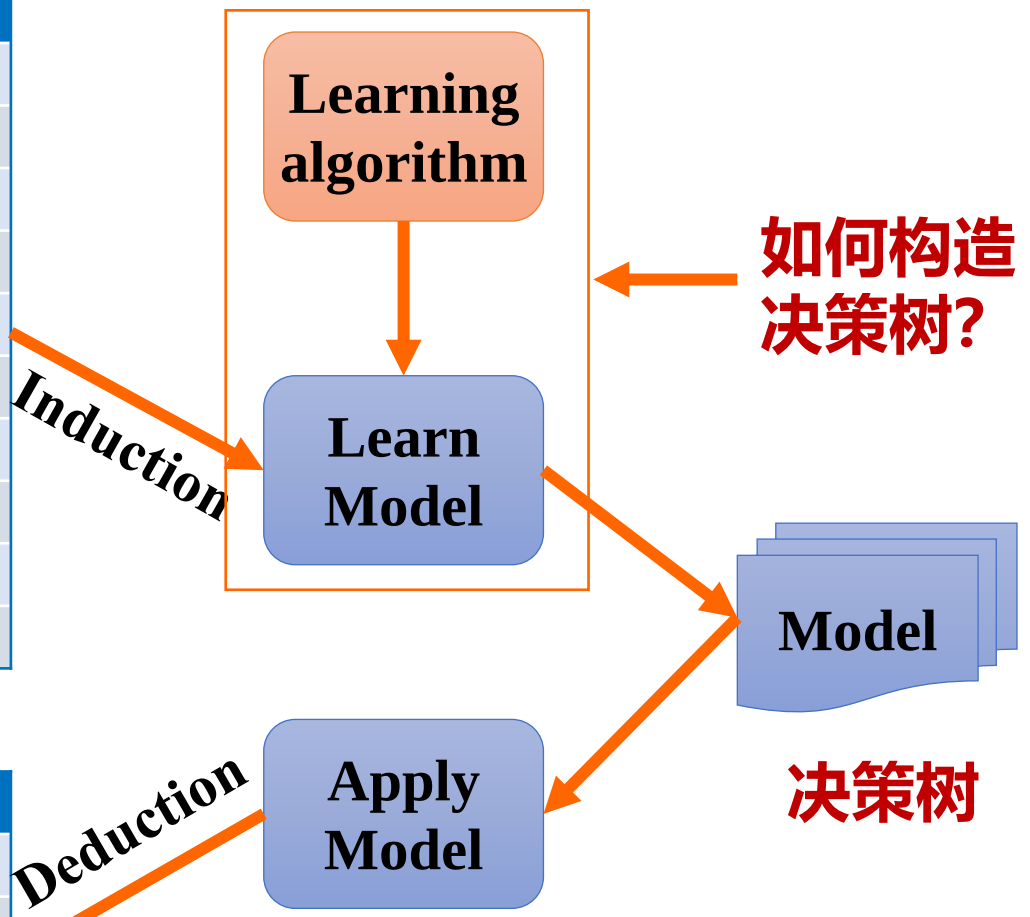
决策树分类

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

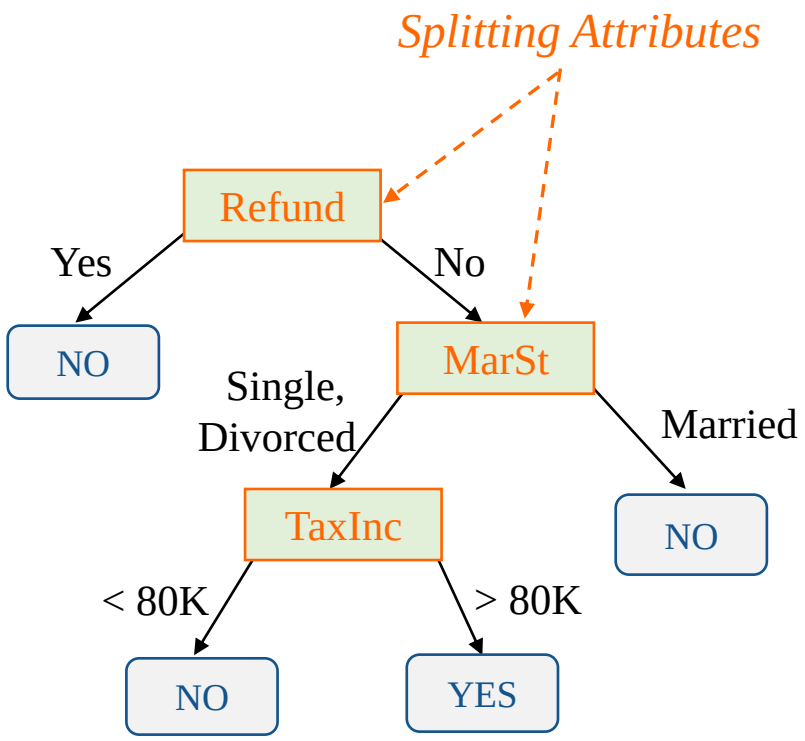


南京大学
NANJING UNIVERSITY

目录

引言

- 树状结构，可以很好的对数据进行分类；
- 决策树的根节点到叶节点的每一条路径构建一条规则；
- 具有互斥且完备的特点，即每一个样本均被且只能被一条路径所覆盖；
- 只要提供的数据量足够庞大真实，通过数据挖掘模式，就可以构造决策树。

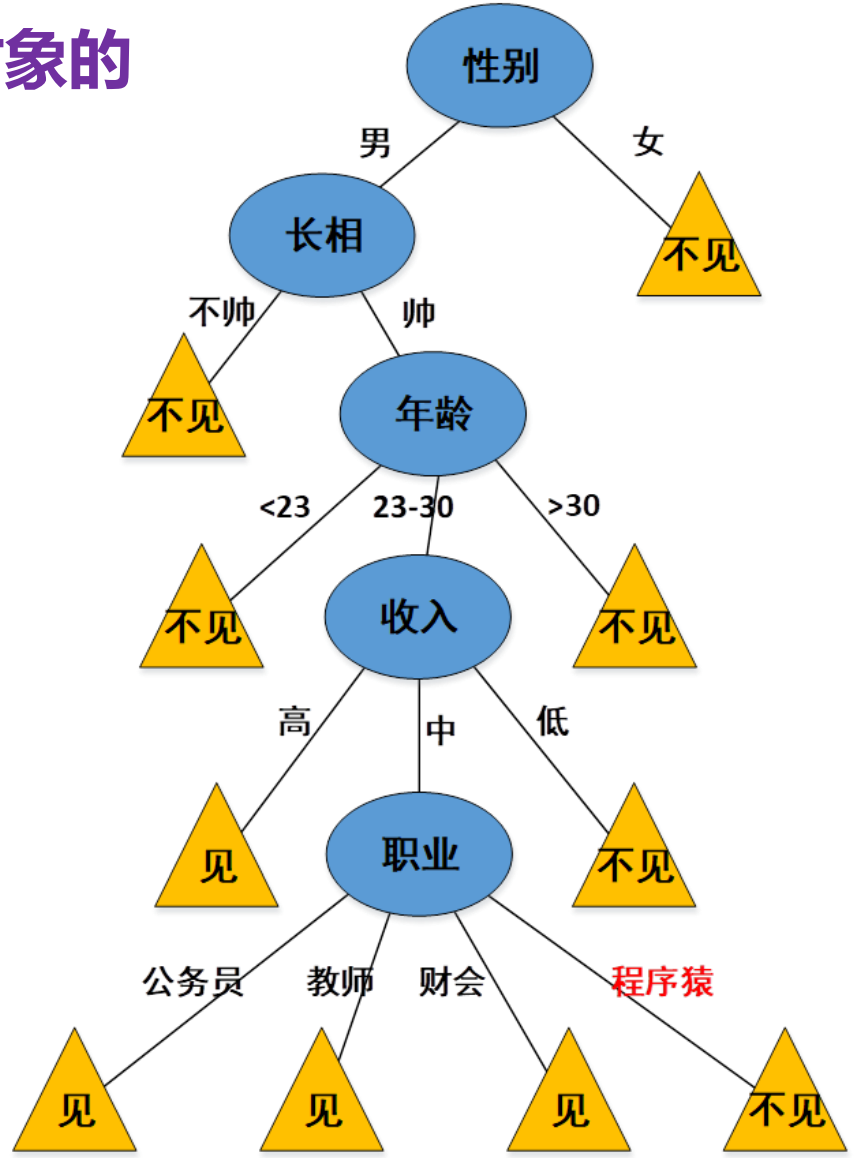


决策树基本思想

- 相亲的常见场景
  - 母亲：尼美，给你介绍个男朋友吧。
  - 尼美：多大年纪了?
  - 母亲：26。
  - 尼美：长的怎么样?
  - 母亲：挺帅的。
  - 尼美：收入高不?
  - 母亲：不算很高，中等情况。
  - 尼美：是公务员不?
  - 母亲：是，在税务局上班呢。
  - 尼美：那好，我去见见。

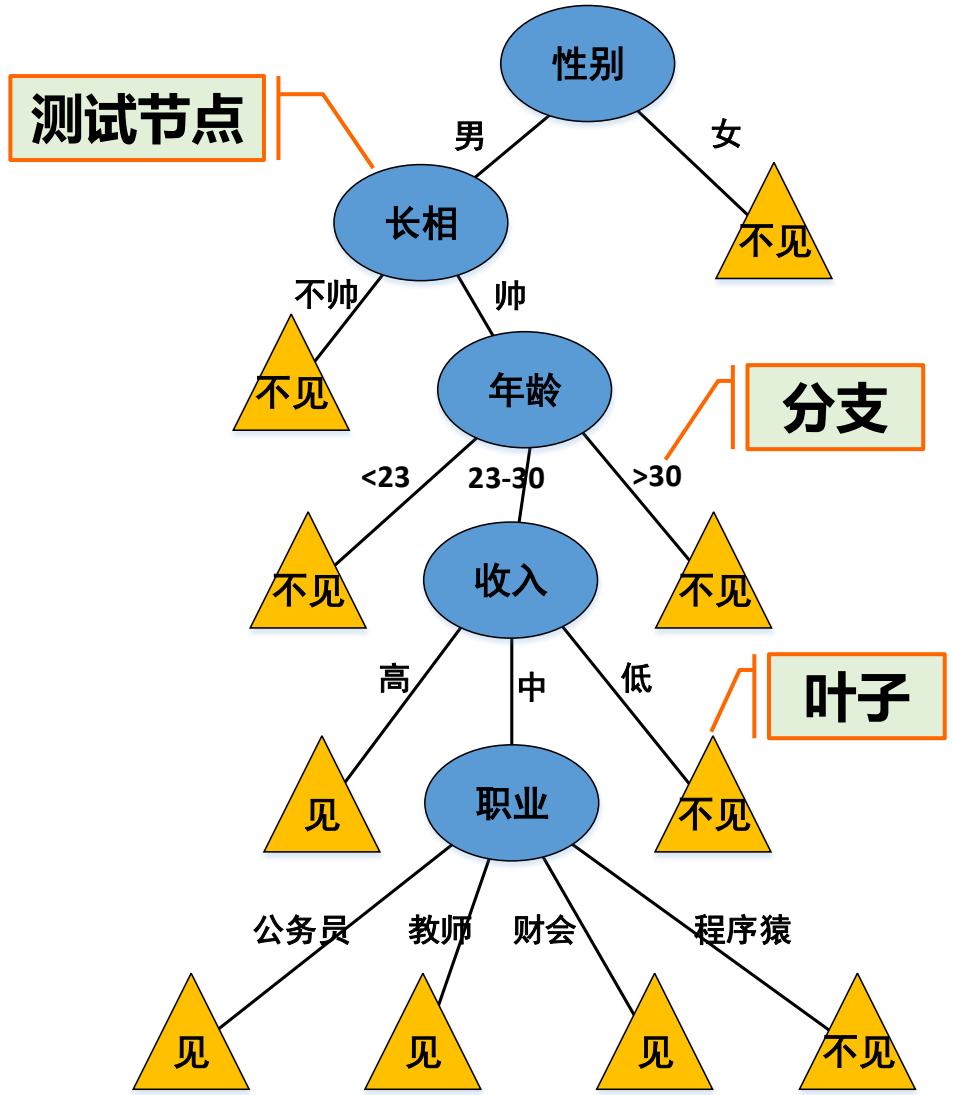
决策树基本思想

- 尼美（女，23岁，企业白领）是如何选择相亲对象的
  - 尼美对相亲对象的属性建模
    - <性别，长相，年龄，收入，职业>
  - 尼美心中对相亲对象筛选过程
    - 性别：当然不能是女的
    - 长相：要帅的
    - 年龄：比自己大但小于30
    - 收入：中等或以上
    - 职业：收入中等则要稳定体面
  - 尼美根据属性作出决定
    - 见 or 不见



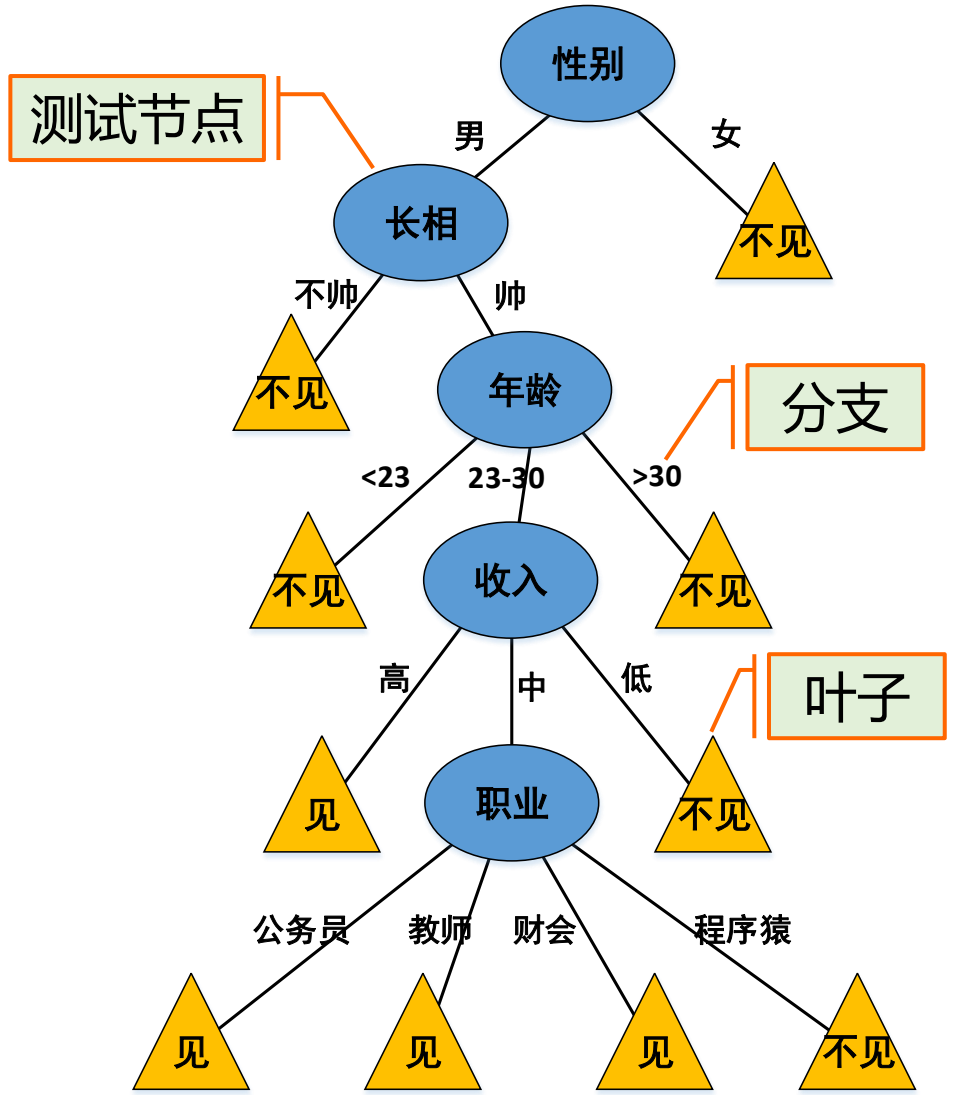
决策树基本思想

- 相亲公司分析了尼美相亲判断过程的基本组成
  - 测试结点
    - 表示某种作为判断条件的属性
  - 分支
    - 根据条件属性取值选取的路径
  - 叶子
    - 使判断终止的结论



决策树基本思想

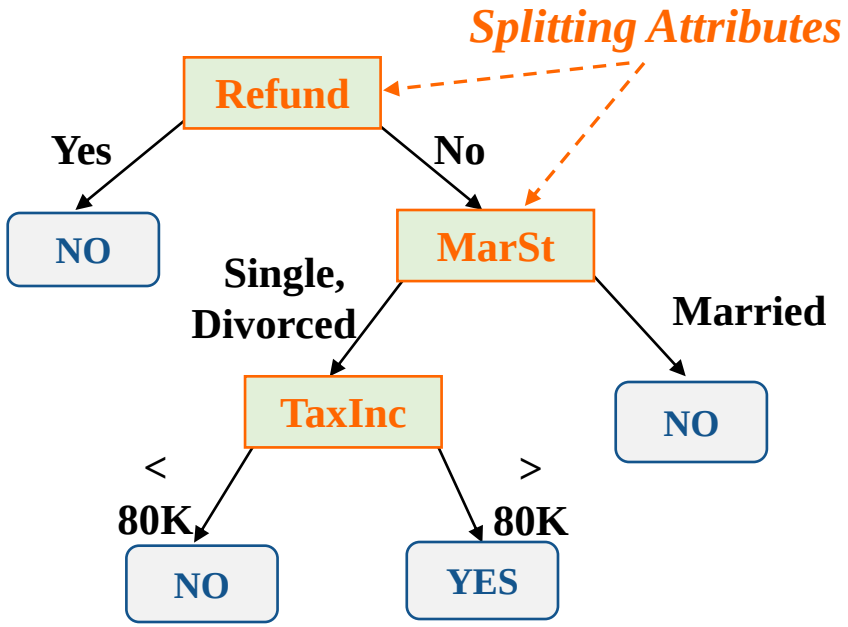
- **相亲公司分析了尼美相亲判断过程的基本组成**
  - **测试结点**
    - 表示某种作为判断条件的属性
  - **分支**
    - 根据条件属性取值选取的路径
  - **叶子**
    - 使判断终止的结论
- **尼美做选择时，其实用的是决策树**
  - **关键在于决策树如何构造**



一个决策树的例子

Tid	Refund	Marital Status	Taxable Income	Cheat
	categorical	categorical	continuous	class
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

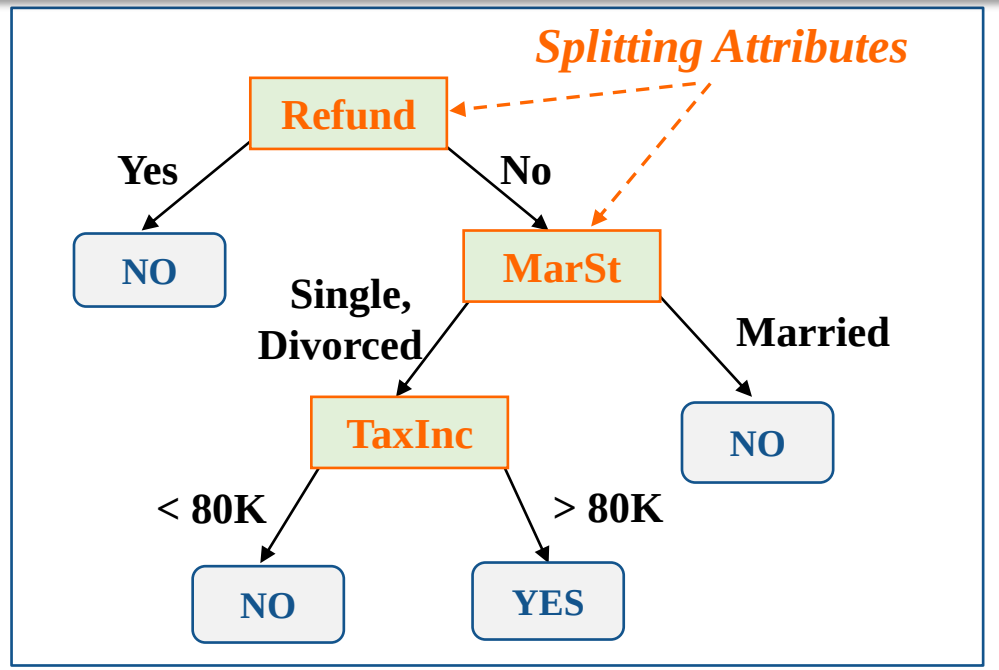
训练数据

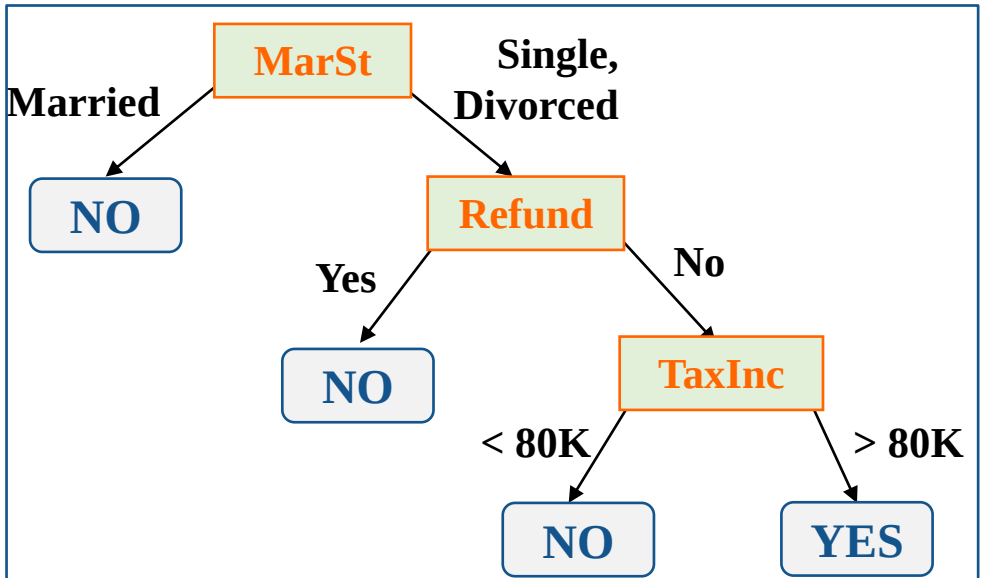


模型：决策树

一个决策树的例子（Different?）

Tid	Refund	Marital Status	Taxable Income	Cheat
		categorical	continuous	class
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

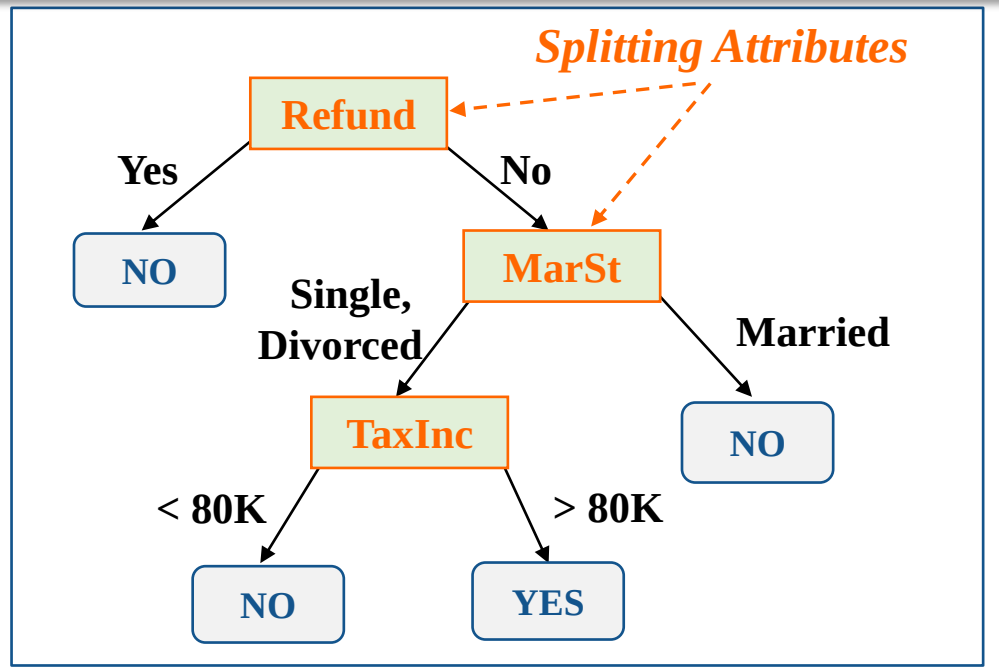


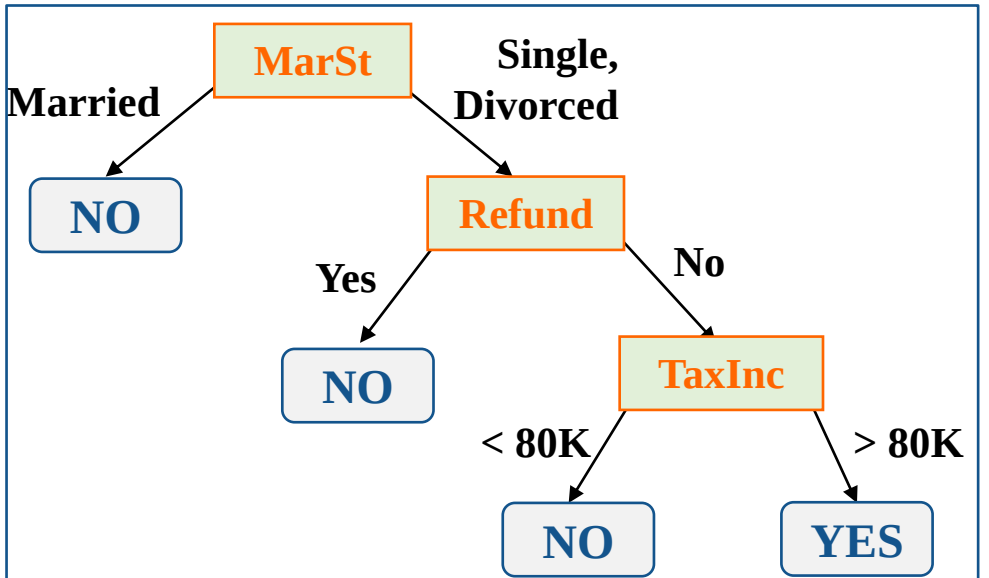


一个决策树的例子（Different?）

Tid	Refund	Marital Status	Taxable Income	Cheat
	categorical	categorical	continuous	class
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

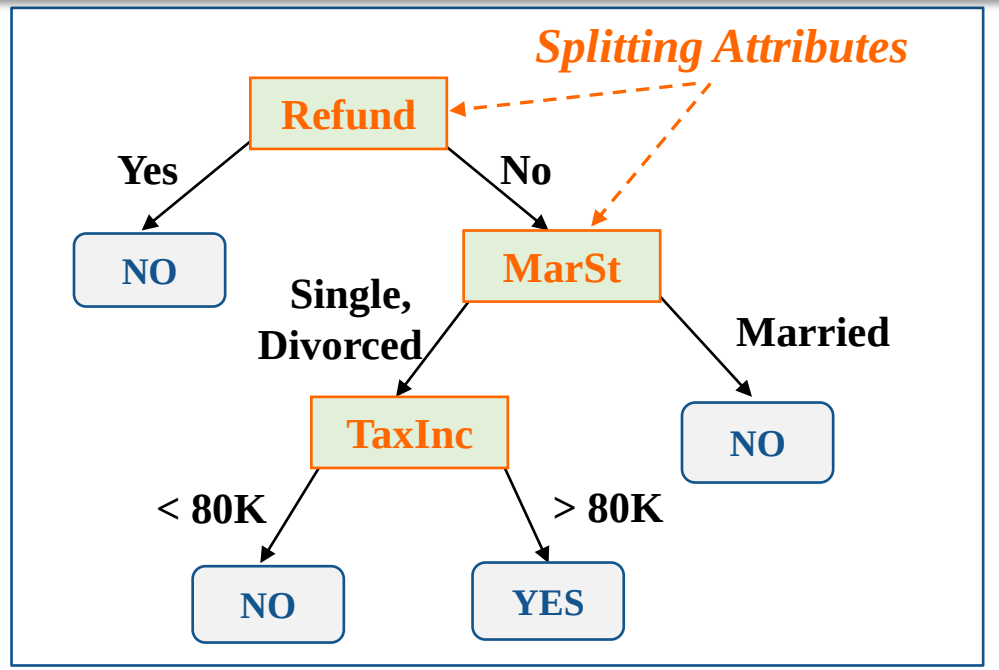
两棵决策树的属性划分顺序不一样

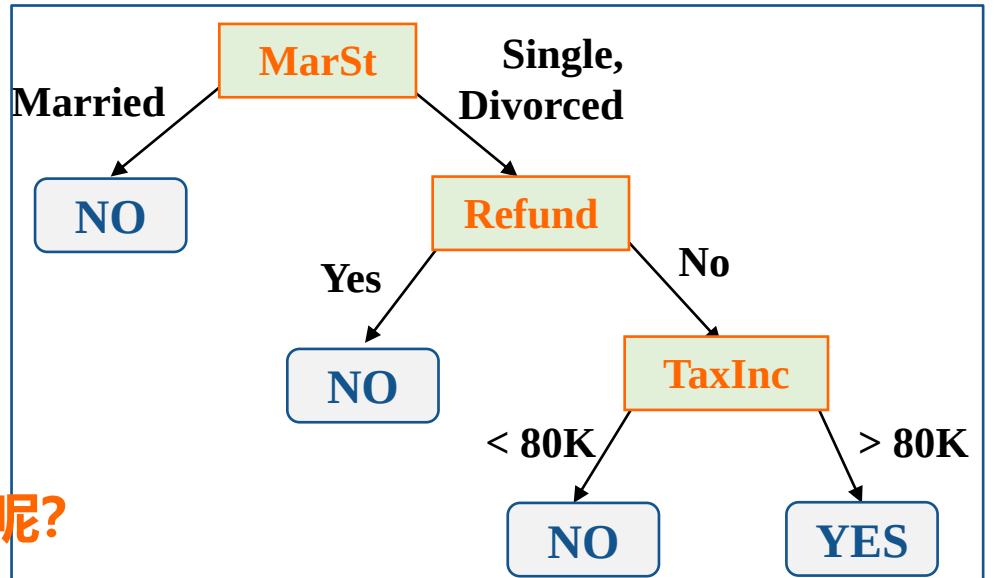




一个决策树的例子（Different?）

Tid	Refund	Marital Status	Taxable Income	Cheat
	categorical	categorical	continuous	class
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





两棵决策树的属性划分顺序不一样

到底构造哪课决策树分类效果最好呢?

南京大学
NANJING UNIVERSITY

目录

构造决策树

- 有许多决策树算法：
  - Hunt算法 （1966）
  - 信息增益——Information gain （ID3）
  - 增益比率——Gain ratio（ID3, C4.5）
  - 基尼指数——Gini index (CART, SLIQ)

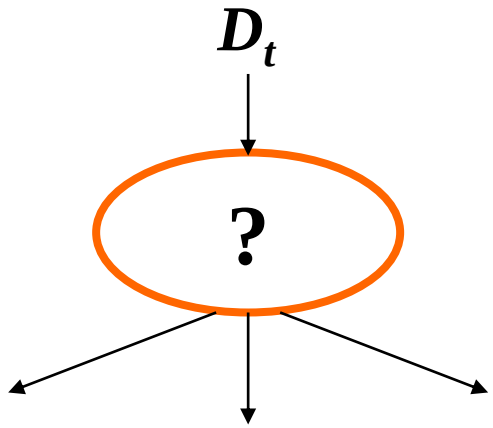
Hunt 算法

设D_t是与节点t相关联的训练记录集，

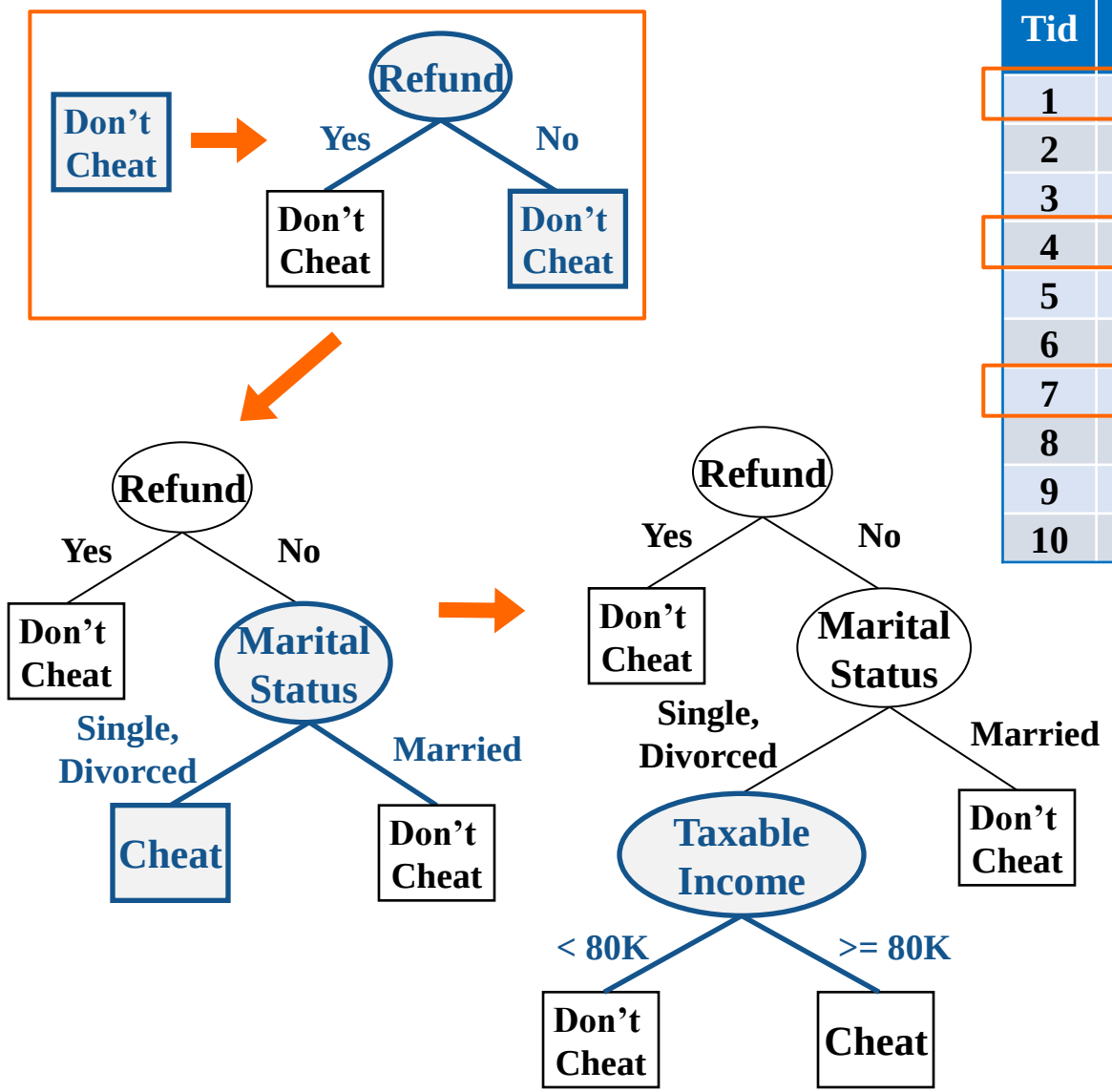
算法步骤：

- **如果D_t 中所有记录都属于同一个类y_t，则t是叶节点，用y_t标记**
- **如果D_t 中包含属于多个类的记录，则选择一个属性测试条件，将记录划分成较小的子集**
- **对于测试条件的每个输出，创建一个子结点，并根据测试结果将Dt中的记录分布到子结点中。然后，对于每个子结点，递归地调用该算法**

Tid	Refund	Marital Status	Taxable Income	Cheat
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



Hunt 算法

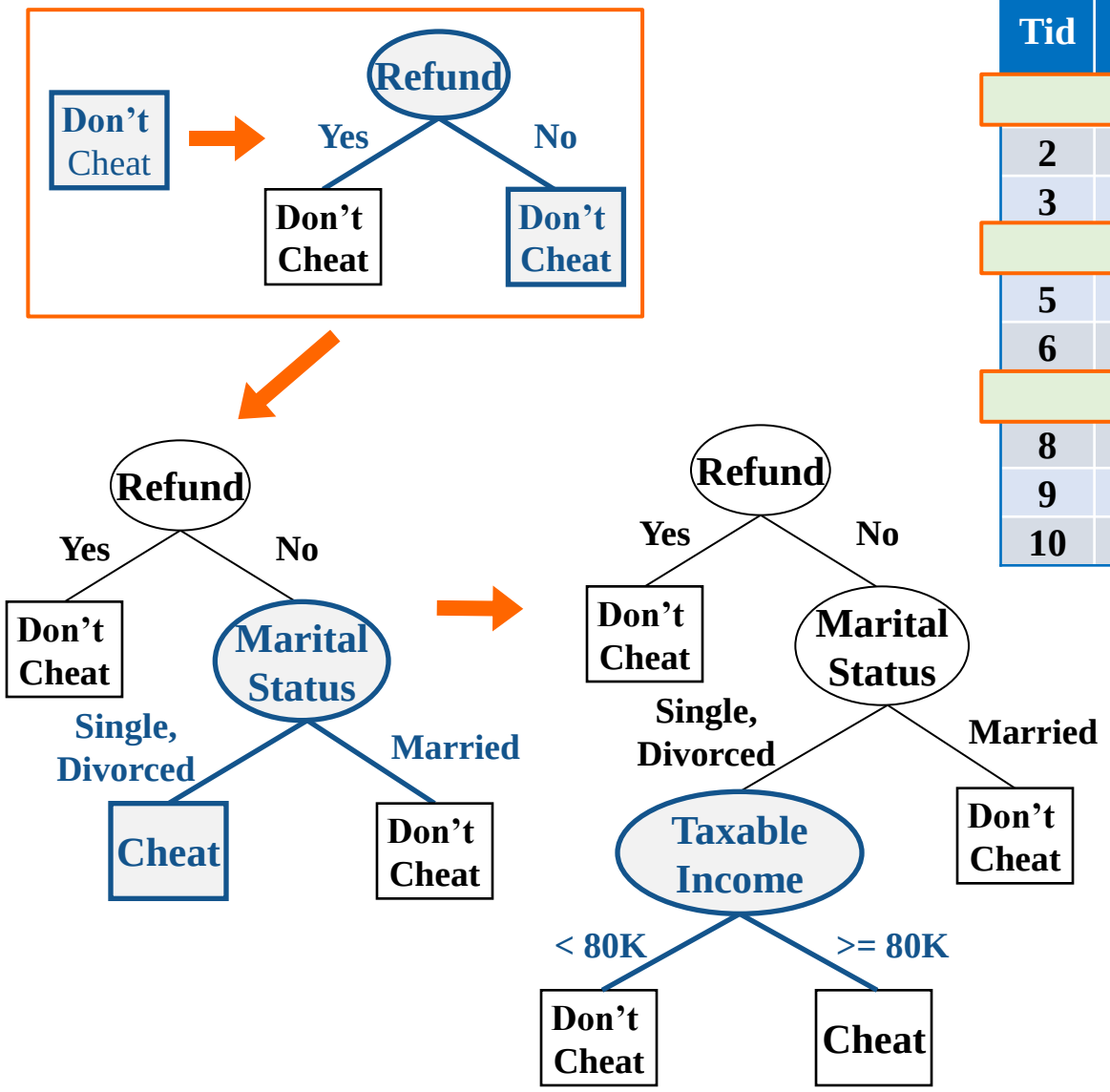


Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

设D_t是与节点t相关联的训练记录集，算法步骤：

- 如果D_t 中所有记录都属于同一个类y_t，则t是叶节点，用y_t标记
- 如果D_t 中包含属于多个类的记录，则**选择一个属性测试条件**，将记录划分成较小的子集
- 对于测试条件的每个输出，创建一个子结点，并根据测试结果将D_t中的记录分布到子结点中。然后，对于每个子结点，递归地调用该算法

Hunt 算法

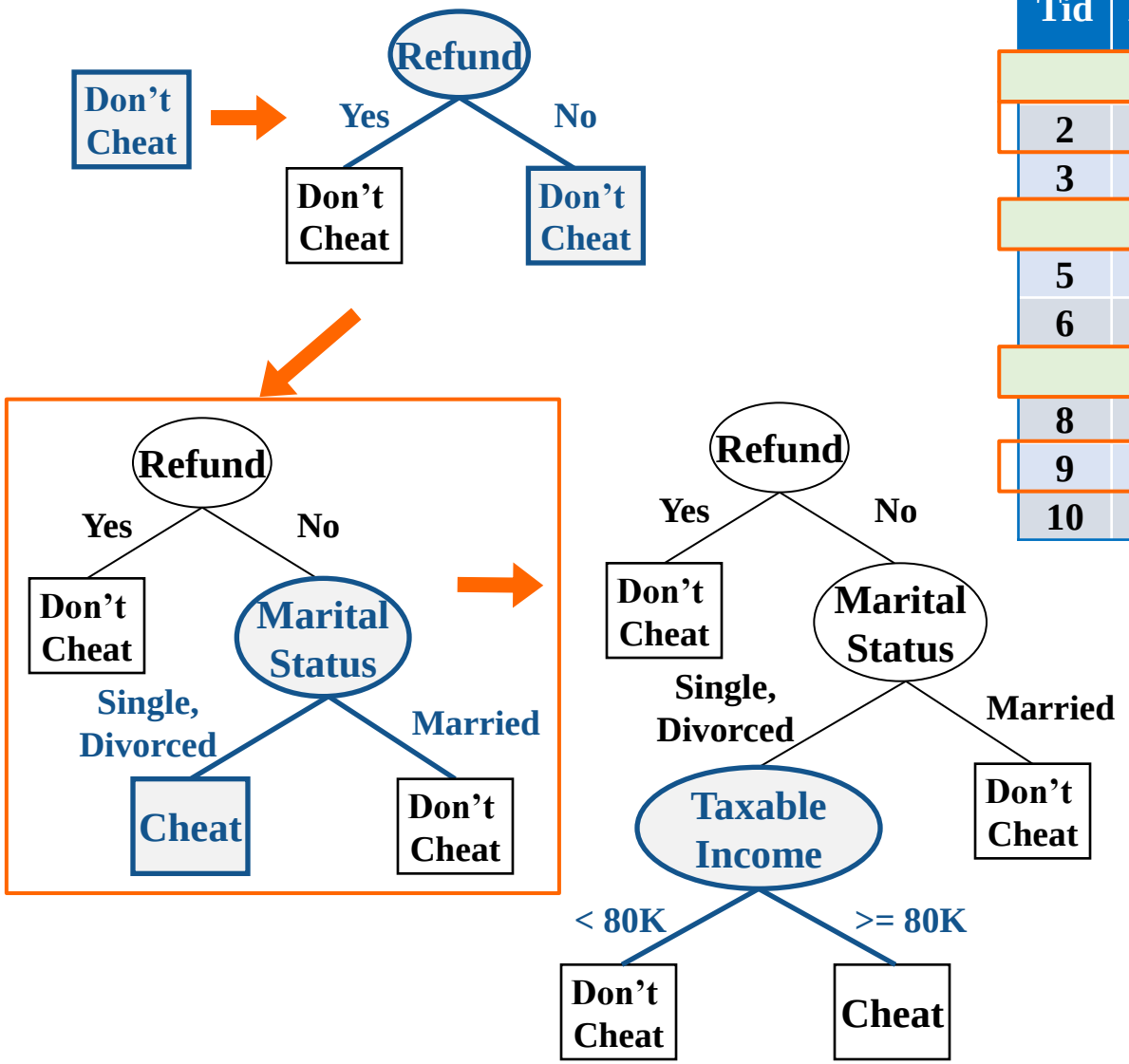


Tid	Refund	Marital Status	Taxable Income	Cheat
2	No	Married	100K	No
3	No	Single	70K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

设D_t是与节点t相关联的训练记录集，算法步骤：

- 如果D_t 中所有记录都属于同一个类y_t，则t是叶节点，用y_t标记
- 如果D_t 中包含属于多个类的记录，则**选择一个属性测试条件**，将记录划分成较小的子集
- 对于测试条件的每个输出，创建一个子结点，并根据测试结果将D_t中的记录分布到子结点中。然后，对于每个子结点，递归地调用该算法

Hunt 算法

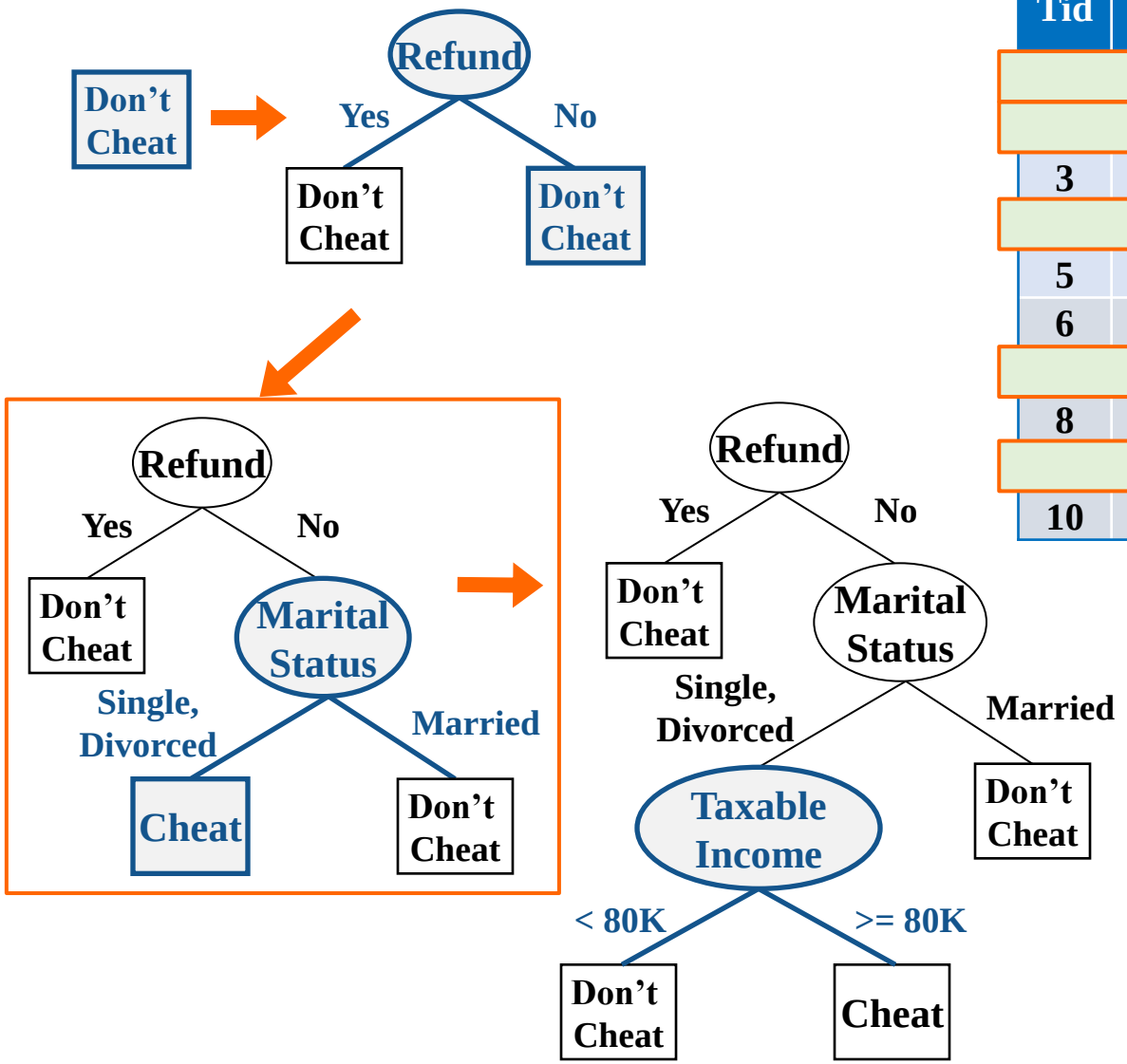


Tid	Refund	Marital Status	Taxable Income	Cheat
2	No	Married	100K	No
3	No	Single	70K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

设D_t是与节点t相关联的训练记录集，算法步骤：

- 如果D_t 中所有记录都属于同一个类y_t，则t是叶节点，用y_t标记
- 如果D_t 中包含属于多个类的记录，则**选择一个属性测试条件**，将记录划分成较小的子集
- 对于测试条件的每个输出，创建一个子结点，并根据测试结果将D_t中的记录分布到子结点中。然后，对于每个子结点，递归地调用该算法

Hunt 算法

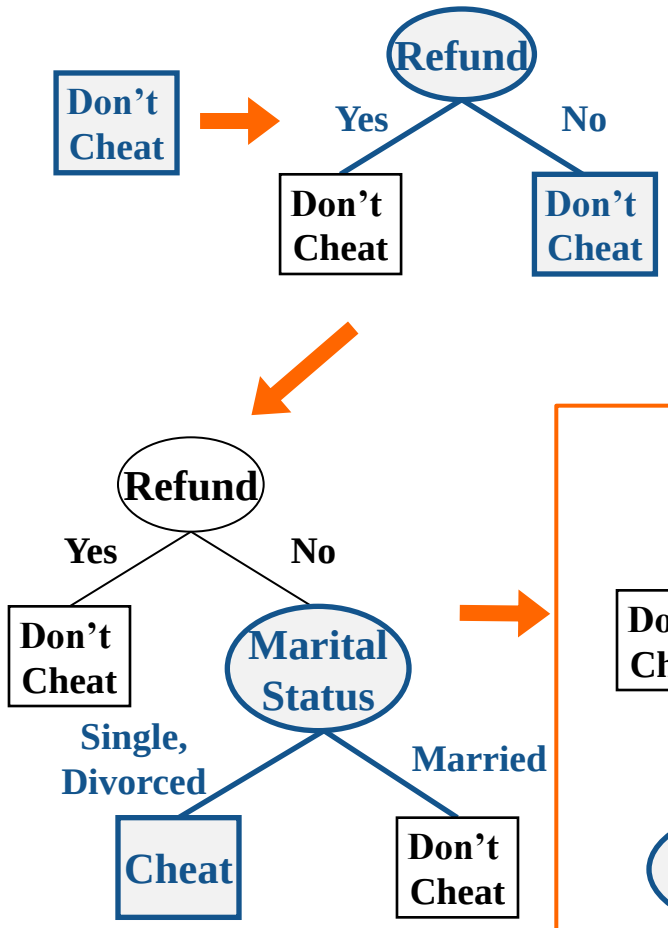


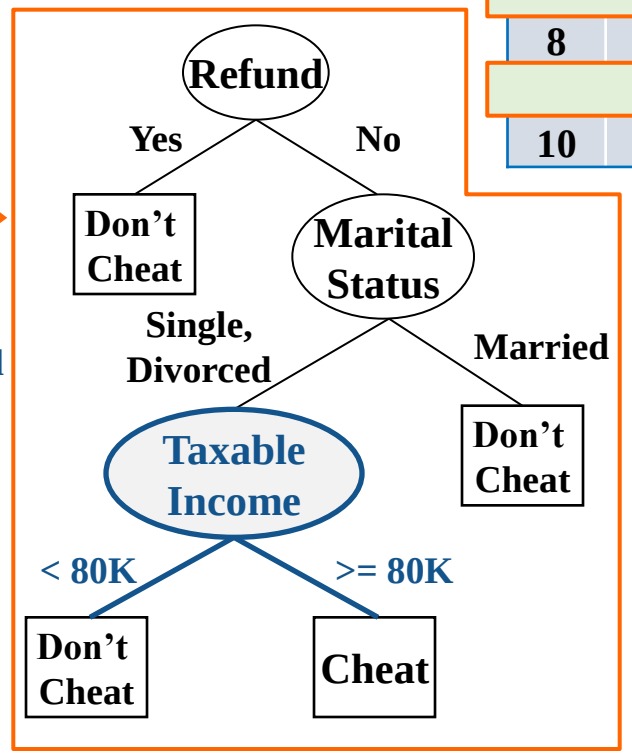
Tid	Refund	Marital Status	Taxable Income	Cheat
3	No	Single	70K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
8	No	Single	85K	Yes
10	No	Single	90K	Yes

设D_t是与节点t相关联的训练记录集，算法步骤：

- 如果D_t 中所有记录都属于同一个类y_t，则t是叶节点，用y_t标记
- 如果D_t 中包含属于多个类的记录，则**选择一个属性测试条件**，将记录划分成较小的子集
- 对于测试条件的每个输出，创建一个子结点，并根据测试结果将D_t中的记录分布到子结点中。然后，对于每个子结点，递归地调用该算法

Hunt 算法





Tid	Refund	Marital Status	Taxable Income	Cheat
3	No	Single	70K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
8	No	Single	85K	Yes
10	No	Single	90K	Yes

设D_t是与节点t相关联的训练记录集，算法步骤：

- 如果D_t 中所有记录都属于同一个类y_t，则t是叶节点，用y_t标记
- 如果D_t 中包含属于多个类的记录，则**选择一个属性测试条件**，将记录划分成较小的子集
- 对于测试条件的每个输出，创建一个子结点，并根据测试结果将D_t中的记录分布到子结点中。然后，对于每个子结点，递归地调用该算法

构造决策树

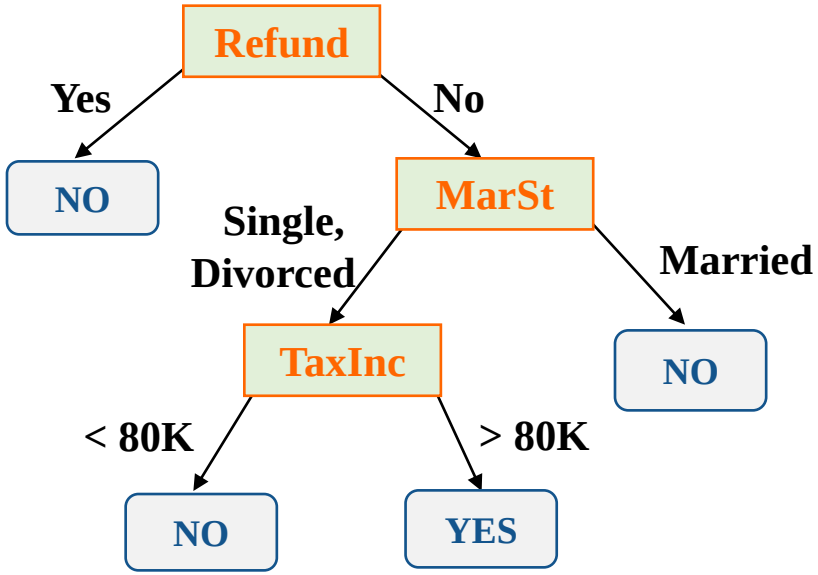
- Hunt算法采用贪心策略构建决策树
  - 在选择划分数据的属性时，采取一系列局部最优决策来构造决策树。
- 决策树归纳的设计问题
  - 如何分裂训练记录?
    - 怎样为不同类型的属性指定测试条件？
    - 怎样评估每种测试条件？
  - 如何停止分裂过程?

Tid	Refund	Marital Status	Taxable Income	Cheat
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

怎样为不同类型的属性指定测试条件?

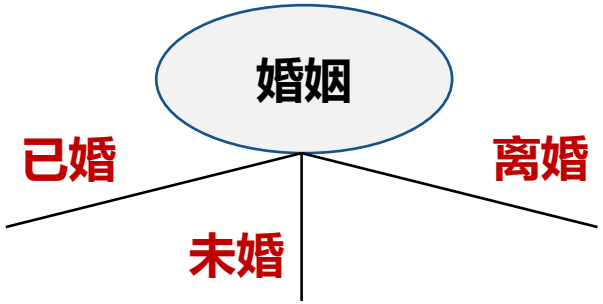
- 依赖于属性的类型
  - 标称
  - 序数
  - 连续
- 依赖于划分的路数
  - 多路划分
  - 二元划分

Tid	Refund	Marital Status	Taxable Income	Cheat
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



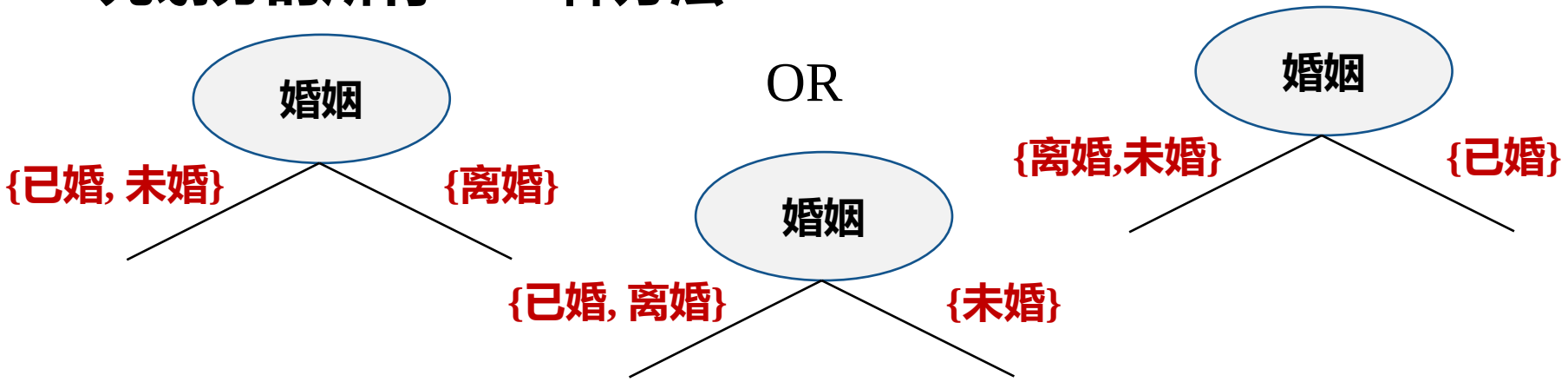
基于标称属性的分裂

- **多路划分：划分数（输出数）取决于该属性不同属性值的个数**



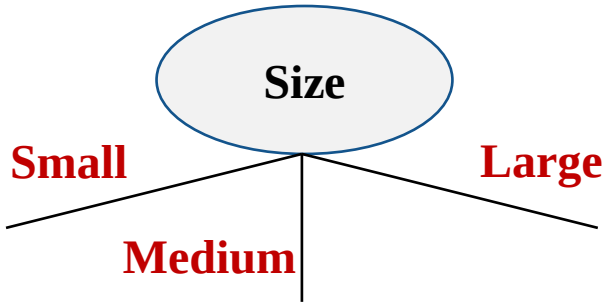
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- **二元划分：划分数为2，这种划分要考虑创建k个属性值的二元划分的所有$2^{k-1}-1$种方法**

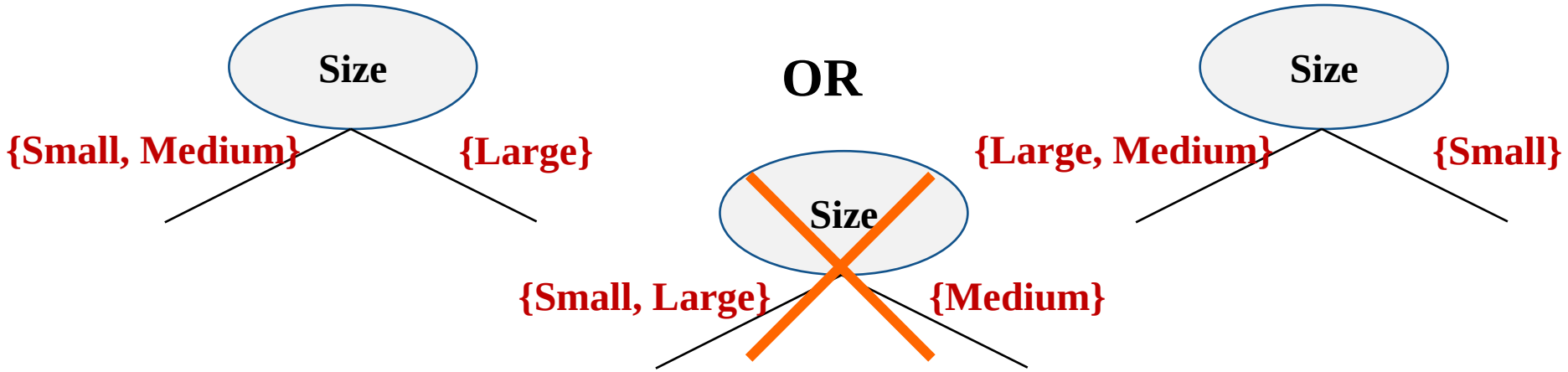


基于序数属性的划分

- **多路划分：划分数（输出数）取决于该属性不同属性值的个数**

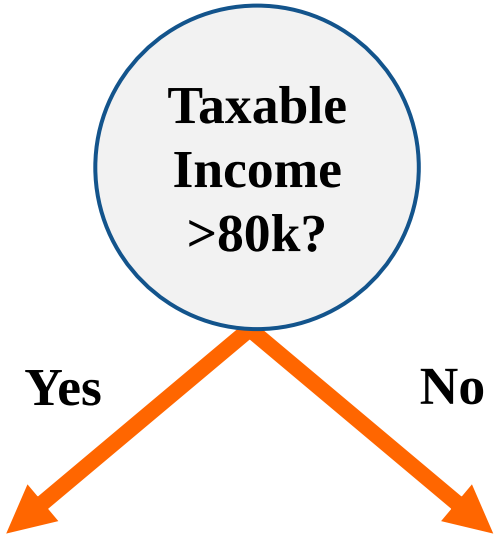


- **二元划分：划分数为2，这种划分要考虑创建k个属性值的二元划分的所有$2^{k-1}-1$种方法**

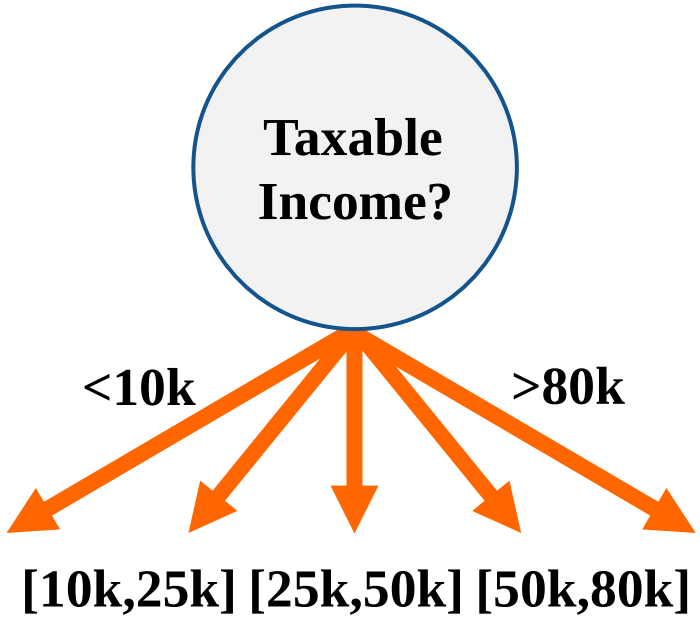


基于连续属性的划分

- **二元划分：** （A<v）or （A≥v）
  - 考虑所有的划分点，选择一个**最优划分点v**
- **多路划分：** $v_i \le A < v_{i+1}$ （i=1,…,k）



Binary split



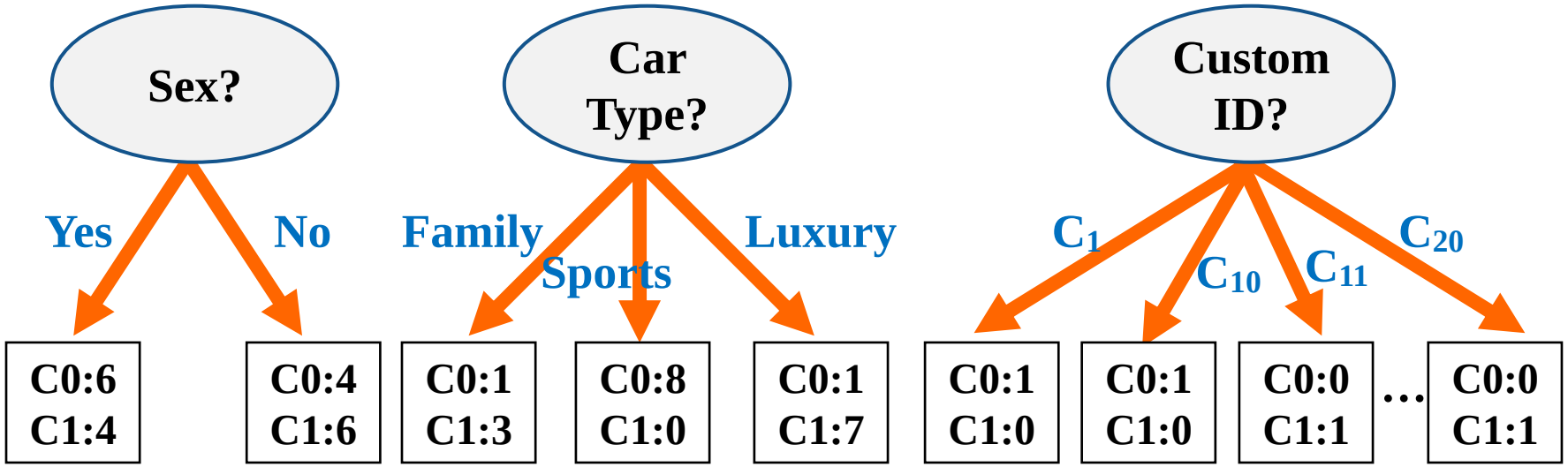
Multi-way split

决策树

- 决策树归纳的设计问题
  - 如何分裂训练记录
    - 怎样为不同类型的属性指定测试条件?
    - 怎样评估每种测试条件?
  - 如何停止分裂过程

怎样选择最佳划分?

- 在划分前：10 个记录 class 0, 10 个记录 class 1



怎样选择最佳划分?

- 选择最佳划分的度量通常是根据划分后子节点纯性的程度。

  纯性的程度越高，类分布就越倾斜，划分结果越好。

  - 结点纯性的度量:

C0:5 C1:5	C0:9 C1:1
纯性小 (不纯性大)	纯性大

顾客数据

训练集如右图所示：

根据训练集数据建立决策树，并判断顾客：

(青年，低收入，无游戏爱好，中等信用度)

是否有购买电脑的倾向

id	年龄	收入	爱好	信用	购买
1	青	高	否	中	否
2	青	高	否	优	否
3	中	高	否	中	是
4	老	中	否	中	是
5	老	低	是	中	是
6	老	低	是	优	否
7	中	低	是	优	是
8	青	中	否	中	否
9	青	低	是	中	是
10	老	中	是	中	是
11	青	中	是	优	是
12	中	中	否	优	是
13	中	高	是	中	是
14	老	中	否	优	否

Entropy 基于熵

$$Entropy(S) = -\sum_{i=1}^{C} p_i \log(p_i)$$

不确定性

- 熵值越高，数据越混乱
- 熵值越低，数据越纯

p_i: the proportion of instances in the dataset that take the i^{th} target value

c1	0
c2	6

数据纯度高

$$Entropy(t) = -\sum_j p(j \mid t) \log p(j \mid t)$$

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

c1	3
c2	3

数据混乱

$$Entropy(t) = -\sum_j p(j \mid t) \log p(j \mid t)$$

P(C1) = 3/6 = 1/2 P(C2) = 3/6 = 1/2

Entropy = – 0.5 log 0.5 – 0.5 log 0.5 = 1

Entropy 基于熵 —— 信息增益算法ID3

$$Entropy(S) = -\sum_{i=1}^{C} p_i \log(p_i)$$

p_i: the proportion of instances in the dataset that take the i^{th} target value

购买的比例为： 9/14

不购买的比例为： 5/14

顾客数据的熵值：

$$Entropy(S) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940$$

$$Gain(S, A) = Entropy(S) - \sum_{v \in A} \frac{|S_v|}{|S|} Entropy(S_v)$$

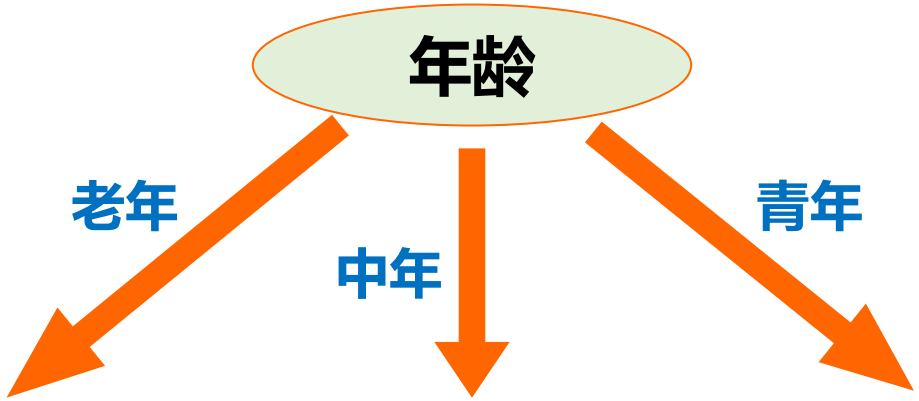
S_v: the subset of S where attribute A takes the value v.

id	年龄	收入	爱好	信用	购买
1	青	高	否	中	否
2	青	高	否	优	否
3	中	高	否	中	是
4	老	中	否	中	是
5	老	低	是	中	是
6	老	低	是	优	否
7	中	低	是	优	是
8	青	中	否	中	否
9	青	低	是	中	是
10	老	中	是	中	是
11	青	中	是	优	是
12	中	中	否	优	是
13	中	高	是	中	是
14	老	中	否	优	否

Entropy 基于熵 —— 信息增益算法ID3

1、假设以年龄为树的根节点

id	年龄	收入	爱好	信用	购买
1	青	高	否	中	否
2	青	高	否	优	否
3	中	高	否	中	是
4	老	中	否	中	是
5	老	低	是	中	是
6	老	低	是	优	否
7	中	低	是	优	是
8	青	中	否	中	否
9	青	低	是	中	是
10	老	中	是	中	是
11	青	中	是	优	是
12	中	中	否	优	是
13	中	高	是	中	是
14	老	中	否	优	否



id	收入	爱好	信用	购买
4	中	否	中	是
5	低	是	中	是
6	低	是	优	否
10	中	是	中	是
14	中	否	优	否

id	收入	爱好	信用	购买
1	高	否	中	否
2	高	否	优	否
8	中	否	中	否
9	低	是	中	是
11	中	是	优	是

id	收入	爱好	信用	购买
3	高	否	中	是
7	低	是	优	是
12	中	否	优	是
13	高	是	中	是

Entropy 基于熵 —— 信息增益算法ID3

$$Gain(S,A) = Entropy(S) - \sum_{v \in A} \frac{|S_v|}{|S|} Entropy(S_v)$$

原始数据分类所需的期望信息：

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

按照年龄分类所需的期望信息：

$$Info_{age}(D) = \frac{5}{14}I(2,3) + \frac{4}{14}I(0,4) + \frac{5}{14}I(3,2) = 0.694$$

id	年龄	收入	爱好	信用	购买
1	青	高	否	中	否
2	青	高	否	优	否
3	中	高	否	中	是
4	老	中	否	中	是
5	老	低	是	中	是
6	老	低	是	优	否
7	中	低	是	优	是
8	青	中	否	中	否
9	青	低	是	中	是
10	老	中	是	中	是
11	青	中	是	优	是
12	中	中	否	优	是
13	中	高	是	中	是
14	老	中	否	优	否

id	收入	爱好	信用	购买
4	中	否	中	是
5	低	是	中	是
6	低	是	优	否
10	中	是	中	是
14	中	否	优	否

id	收入	爱好	信用	购买
1	高	否	中	否
2	高	否	优	否
8	中	否	中	否
9	低	是	中	是
11	中	是	优	是

id	收入	爱好	信用	购买
3	高	否	中	是
7	低	是	优	是
12	中	否	优	是
13	高	是	中	是

Entropy 基于熵 —— 信息增益算法ID3

原始数据分类所需的期望信息：

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

按照年龄分类所需的期望信息：

$$Info_{age}(D) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(3,2) = 0.694$$

信息增益：

$$Gain(S, A) = Entropy(S) - \sum_{v \in A} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

Entropy 基于熵 —— 信息增益算法ID3

相似的

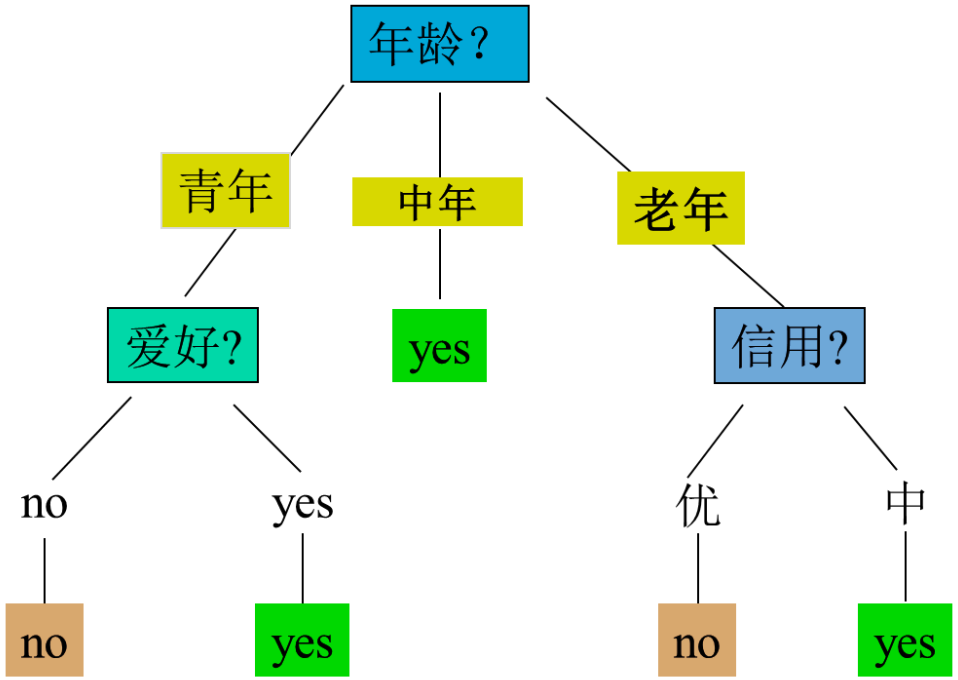
$$Gain(age) = 0.246$$
$$Gain(income) = 0.029$$
$$Gain(fancy) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

id	年龄	收入	爱好	信用	购买
1	青	高	否	中	否
2	青	高	否	优	否
3	中	高	否	中	是
4	老	中	否	中	是
5	老	低	是	中	是
6	老	低	是	优	否
7	中	低	是	优	是
8	青	中	否	中	否
9	青	低	是	中	是
10	老	中	是	中	是
11	青	中	是	优	是
12	中	中	否	优	是
13	中	高	是	中	是
14	老	中	否	优	否

Entropy 基于熵 —— 信息增益算法ID3

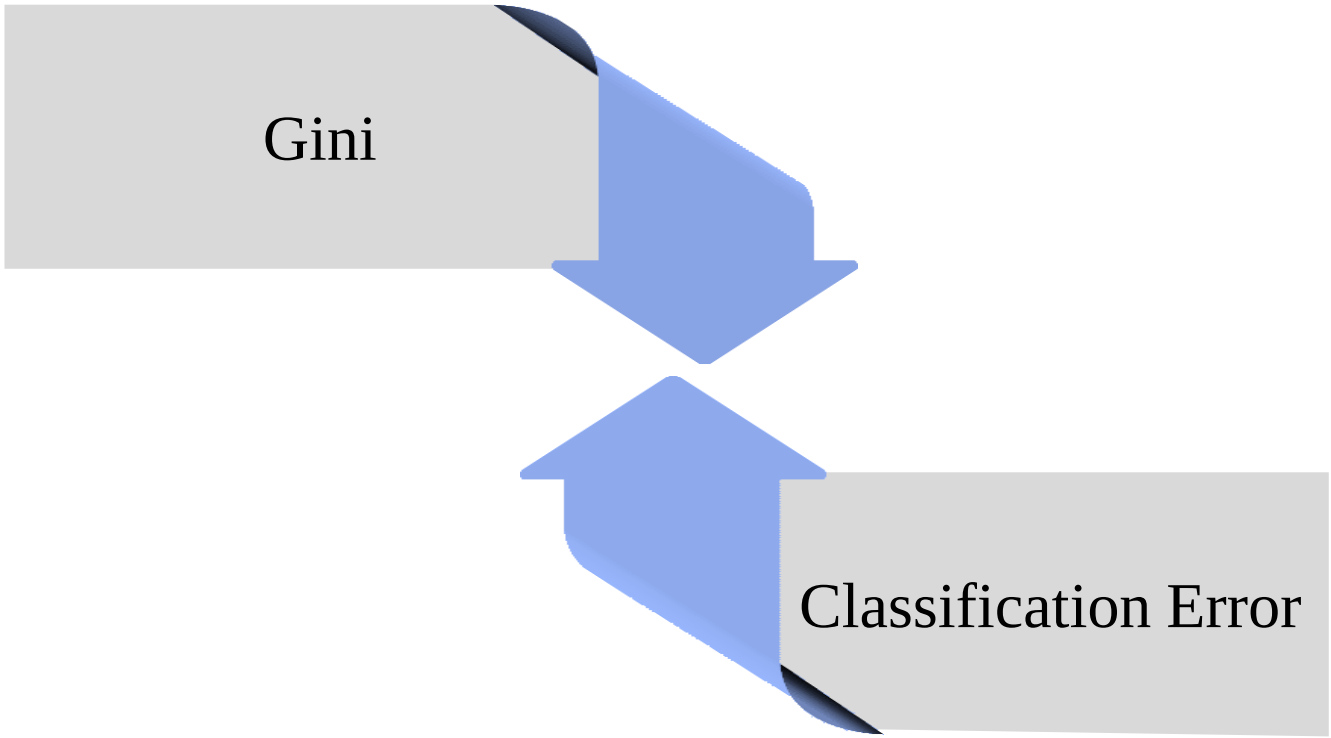
相似的

$$Gain(age) = 0.246$$
$$Gain(income) = 0.029$$
$$Gain(fancy) = 0.151$$
$$Gain(credit\_rating) = 0.048$$



id	年龄	收入	爱好	信用	购买
1	青	高	否	中	否
2	青	高	否	优	否
3	中	高	否	中	是
4	老	中	否	中	是
5	老	低	是	中	是
6	老	低	是	优	否
7	中	低	是	优	是
8	青	中	否	中	否
9	青	低	是	中	是
10	老	中	是	中	是
11	青	中	是	优	是
12	中	中	否	优	是
13	中	高	是	中	是
14	老	中	否	优	否

其它结点纯性的测量



纯性的测量：GINI

- **给定结点t的Gini值计算：**

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

(p(j | t) 是在结点t中，类j发生的概率)

- **当类分布均衡时，Gini值达到最大值 (1 - 1/nc)**
- **相反当只有一个类时，Gini值达到最小值0，纯性越大**

c1	0
c2	6

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

P(C1) = 0/6 = 0　　P(C2) = 6/6 = 1

Gini = 1 – P(C1)^2 – P(C2)^2 = 1 – 0 – 1 = 0

c1	3
c2	3

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

P(C1) = 3/6 = 1/2　　P(C2) = 3/6 = 1/2

Gini = 1 – P(C1)^2 – P(C2)^2 = 1 – 1/4 – 1/4 = 1/2

基于 Classification Error的划分

- **给定结点t的Classification Error值计算：**

$$Error(t) = 1 - \max_i P(i \mid t)$$

  - **当类分布均衡时，Gini值达到最大值 (1 - 1/nc)**
  - **相反当只有一个类时，Gini值达到最小值0，纯性越大**

c1	0
c2	6

P(C1) = 0/6 = 0　　P(C2) = 6/6 = 1

Error = 1 – max (0, 1) = 1 – 1 = 0

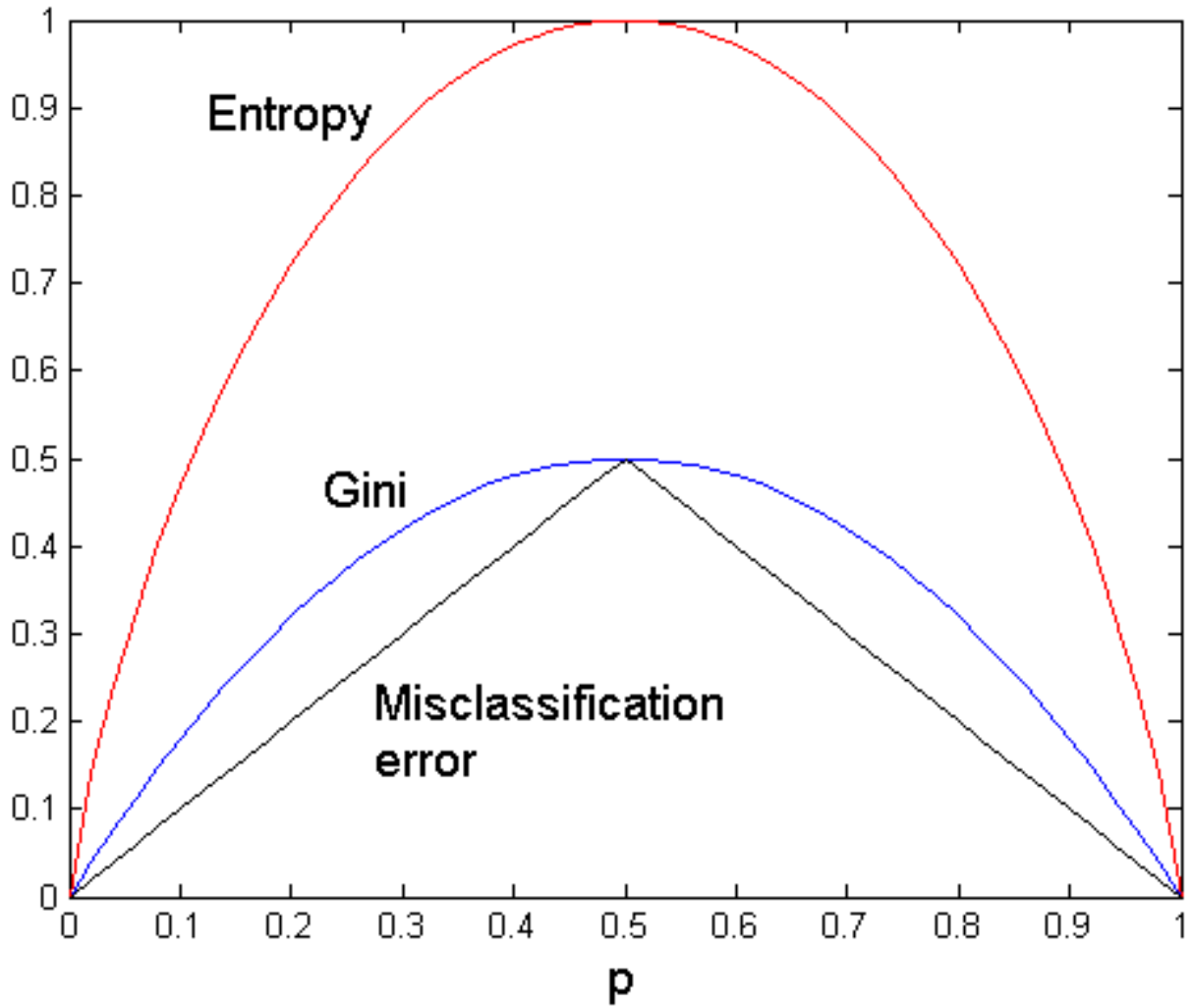
c1	3
c2	3

P(C1) = 1/6　　P(C2) = 5/6

Error = 1 – max (1/6, 5/6) = 1 – 5/6 = 1/6

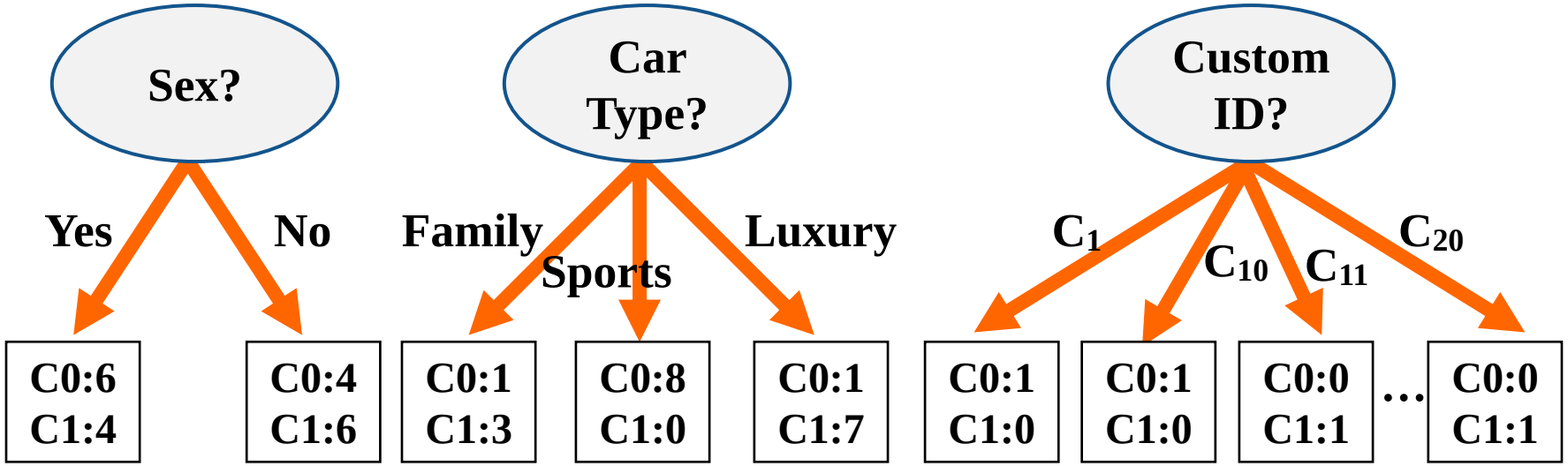
纯性度量之间的比较

- **二元分类问题:**



思考，哪棵树子节点纯性最高?

在划分前：10 个记录 class 0, 10 个记录 class 1



Entropy Bias

基于熵和Gini指标，会趋向于具有大量不同值的划分：

利用雇员id产生更纯的划分，但它却毫无用处。

考虑增益率（Gain Ratio）C4.5算法

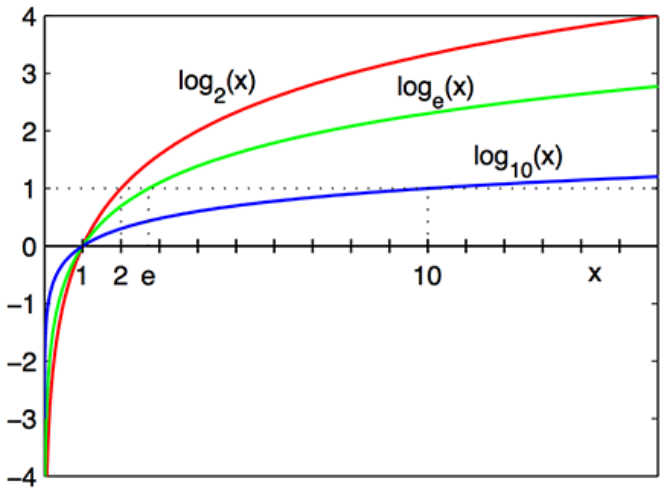
解决该问题的策略有两种：

- 限制测试条件只能是二元划分
- 使用增益率，K越大，SplitINFO越大，增益率被平衡。

id	年龄	收入	爱好	信用	购买
1	青	高	否	中	否
2	青	高	否	优	否
3	中	高	否	中	是
4	老	中	否	中	是
5	老	低	是	中	是
6	老	低	是	优	否
7	中	低	是	优	是
8	青	中	否	中	否
9	青	低	是	中	是
10	老	中	是	中	是
11	青	中	是	优	是
12	中	中	否	优	是
13	中	高	是	中	是
14	老	中	否	优	否

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO}$$

$$SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$



考虑增益率（Gain Ratio）C4.5算法

$$Gain(age) = 0.246$$

$$Gain(income) = 0.029$$

$$Gain(fancy) = 0.151$$

$$Gain(credit\_rating) = 0.048$$

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO}$$

$$SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

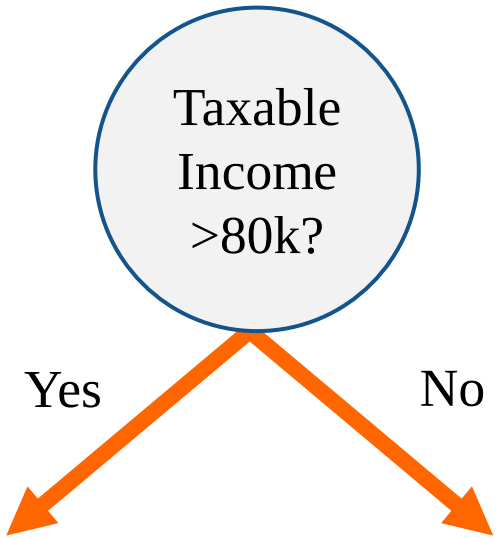
id	年龄	收入	爱好	信用	购买
1	青	高	否	中	否
2	青	高	否	优	否
3	中	高	否	中	是
4	老	中	否	中	是
5	老	低	是	中	是
6	老	低	是	优	否
7	中	低	是	优	是
8	青	中	否	中	否
9	青	低	是	中	是
10	老	中	是	中	是
11	青	中	是	优	是
12	中	中	否	优	是
13	中	高	是	中	是
14	老	中	否	优	否

$$SplitInfo_{income}(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right) = 1.557$$

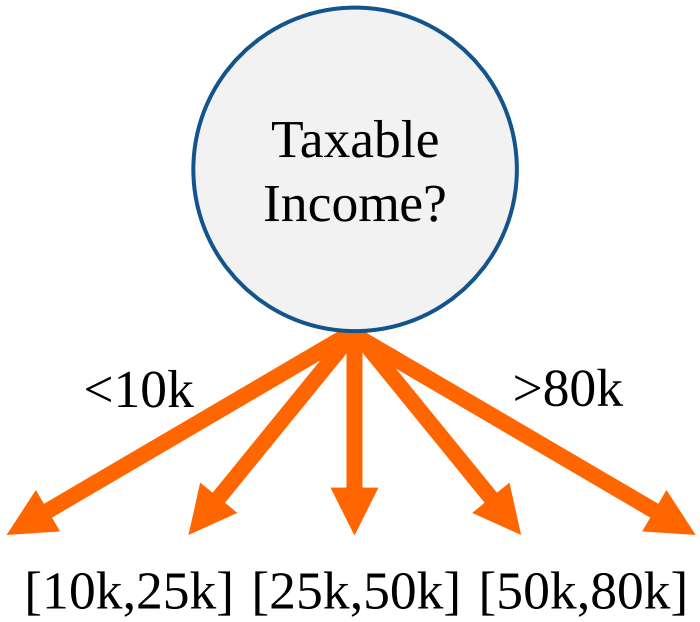
gain_ratio(income) = 0.029/1.557 = 0.019

数据是连续的怎么办

- **二元划分：** $(A<v)$ or $(A\geq v)$
  - 考虑所有的划分点，选择一个**最优划分点v**
- **多路划分：** $v_i \leq A < v_{i+1}$ $(i=1,\ldots,k)$



Binary split



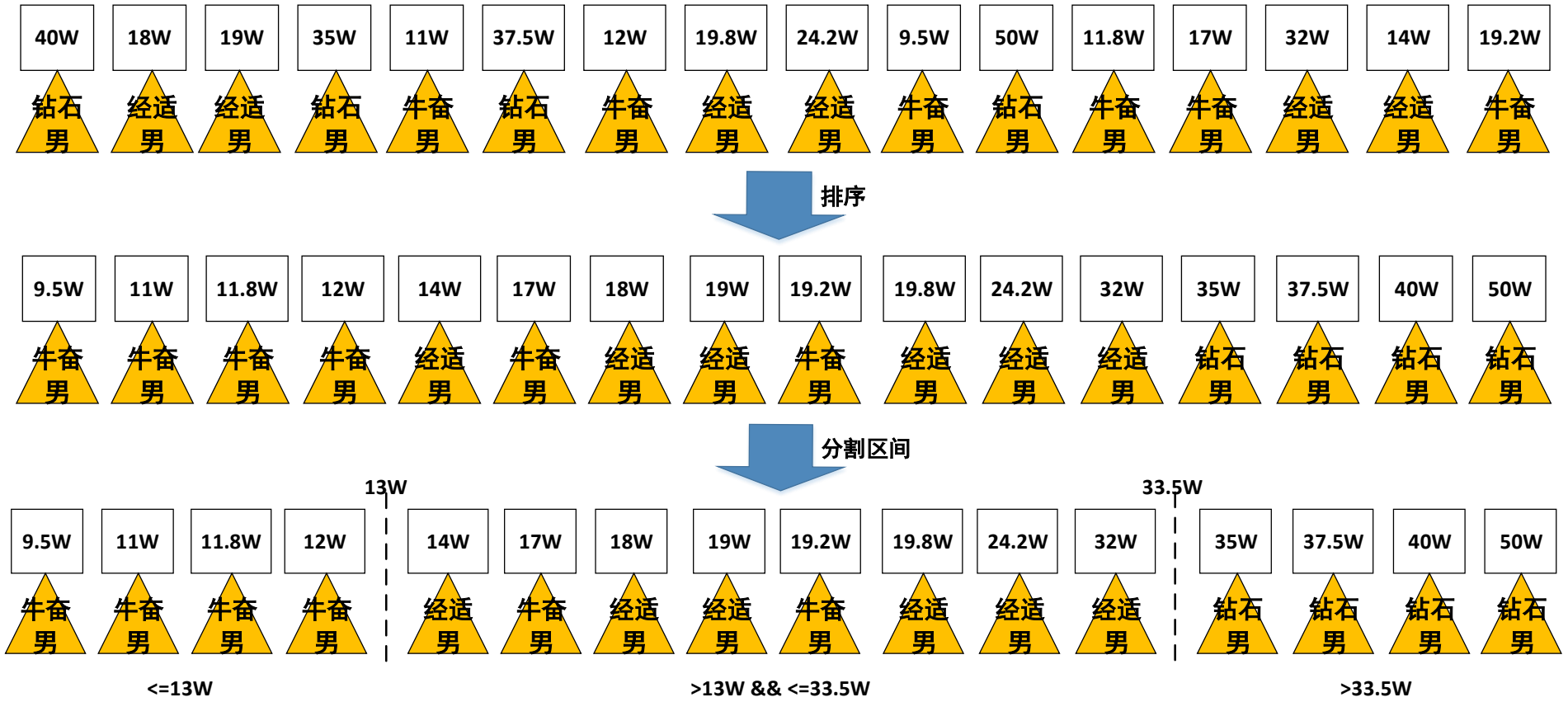
Multi-way split

一个例子

序号	姓名	职业分类	职位评级	收入	有房有车	债务情况	评级
1	A	金融	A类	40W	1	低	钻石男
2	B	IT	A类	18W	3	高	经适男
3	C	行政	A类	19W	2	低	经适男
4	D	司法	A类	35W	0	低	钻石男
5	E	行政	B类	11W	3	中	牛奋男
6	F	金融	B类	37.5W	3	低	钻石男
7	G	IT	B类	12W	2	中	牛奋男
8	H	司法	A类	19.8W	2	低	经适男
9	J	行政	A类	24.2W	0	低	经适男
10	K	教育	C类	9.5W	3	低	牛奋男
11	L	司法	A类	50W	3	中	钻石男
12	M	教育	C类	11.8W	2	低	牛奋男
13	N	IT	B类	17W	0	低	牛奋男
14	P	教育	A类	32W	2	中	经适男
15	Q	教育	C类	14W	2	低	经适男
16	R	IT	B类	19.2W	2	高	牛奋男

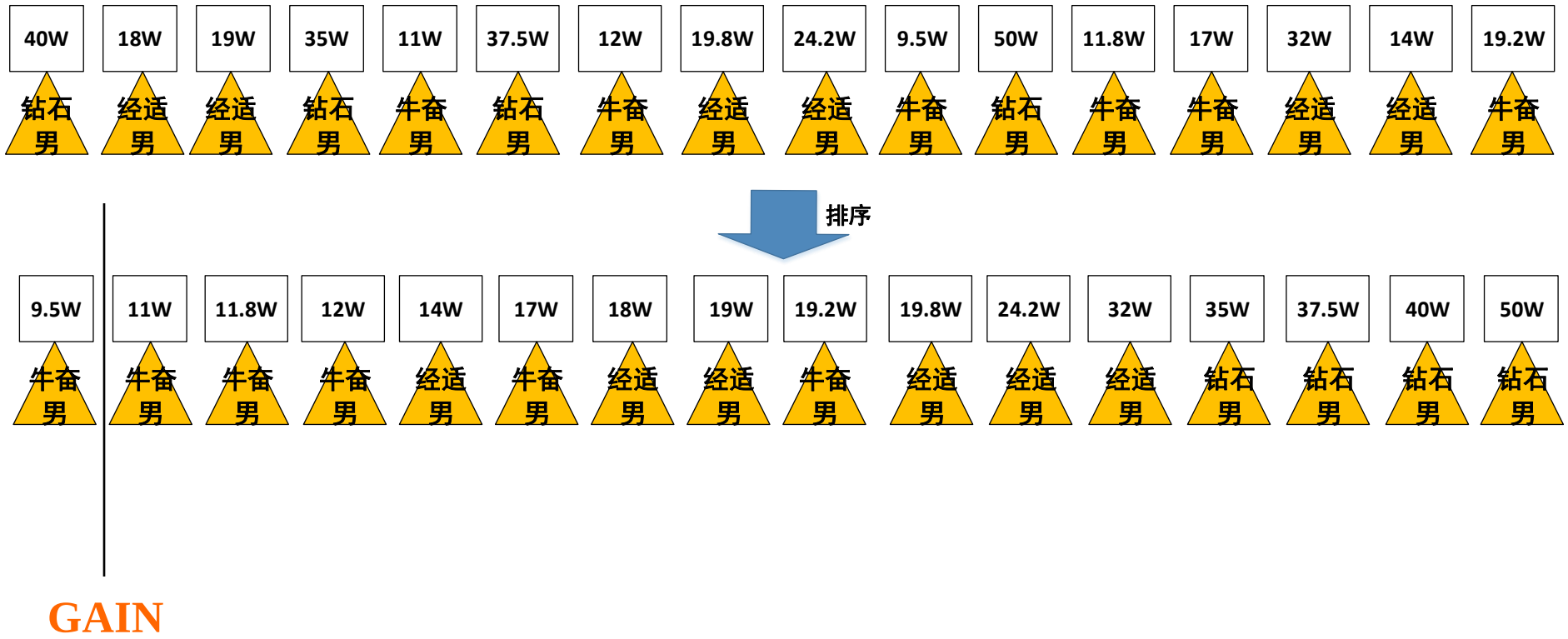
多路划分 —— 连续变量的离散化处理

- 收入是个连续变量，分割成离散区间



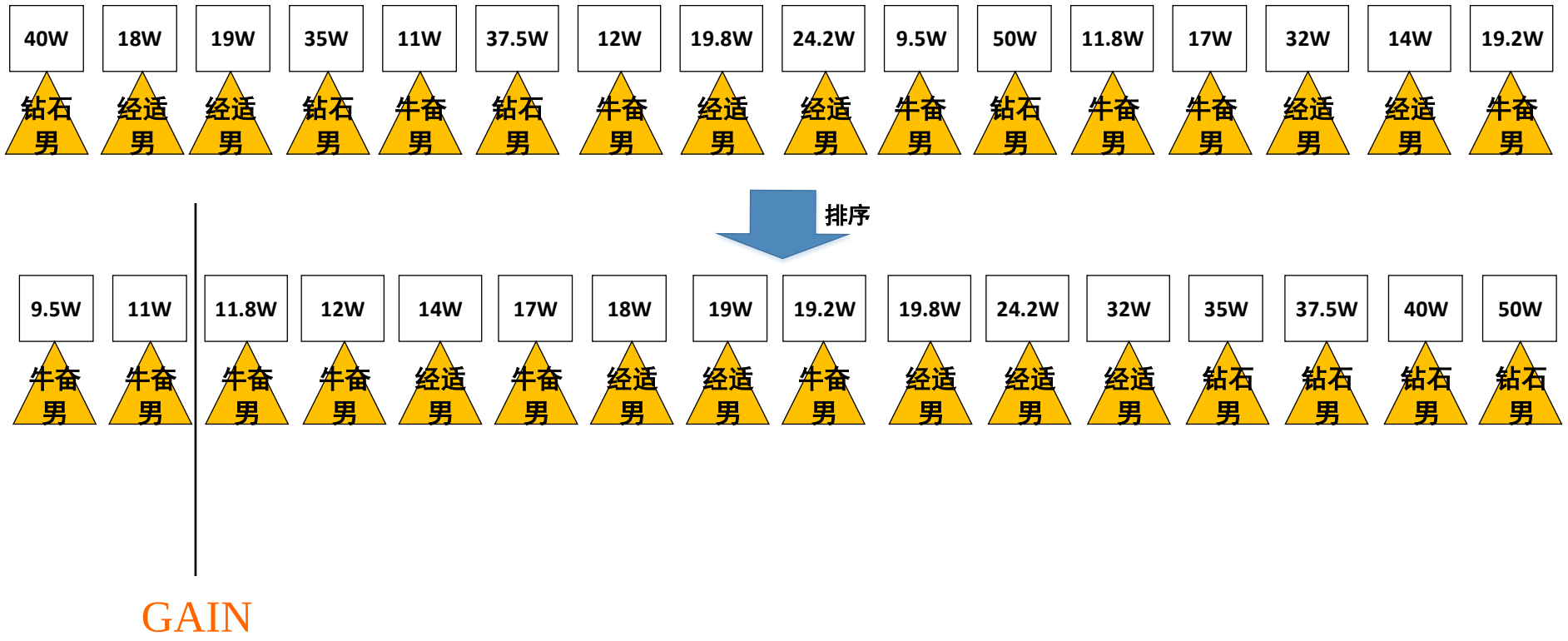
二元划分 —— 选择最佳划分点

- **分割区间的策略**
  - 从最小值开始建立分割区间，开始计算各自的信息增益，选择信息增益最大的一个分割区间作为最佳划分点



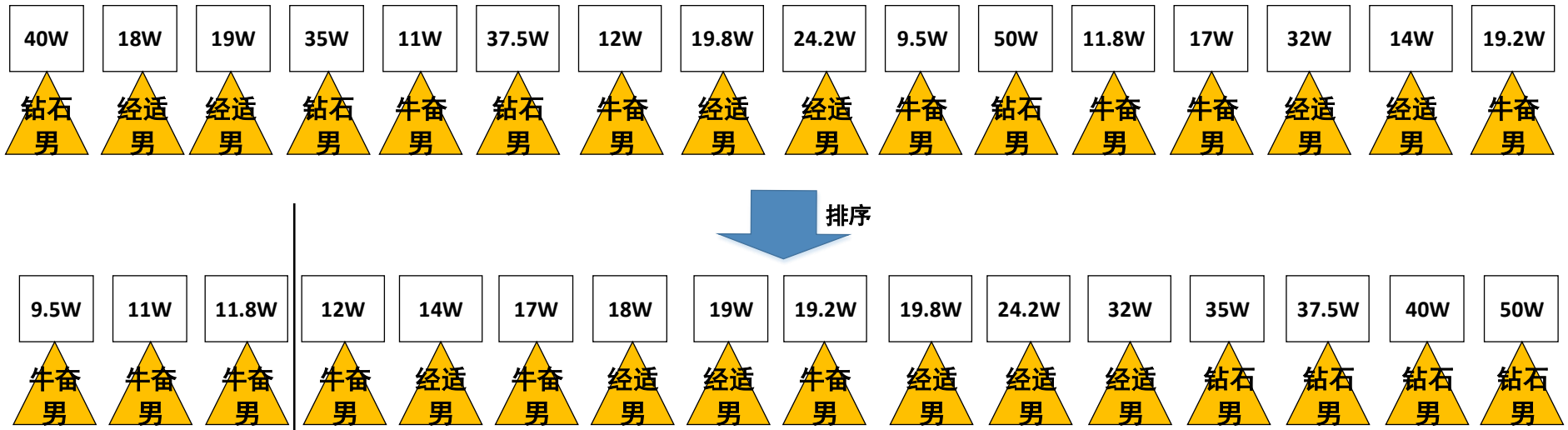
二元划分 —— 选择最佳划分点

- **分割区间的策略**
  - 从最小值开始建立分割区间，开始计算各自的信息增益，选择信息增益最大的一个分割区间作为最佳划分点



二元划分 —— 选择最佳划分点

- **分割区间的策略**
  - 从最小值开始建立分割区间，开始计算各自的信息增益，选择信息增益最大的一个分割区间作为最佳划分点

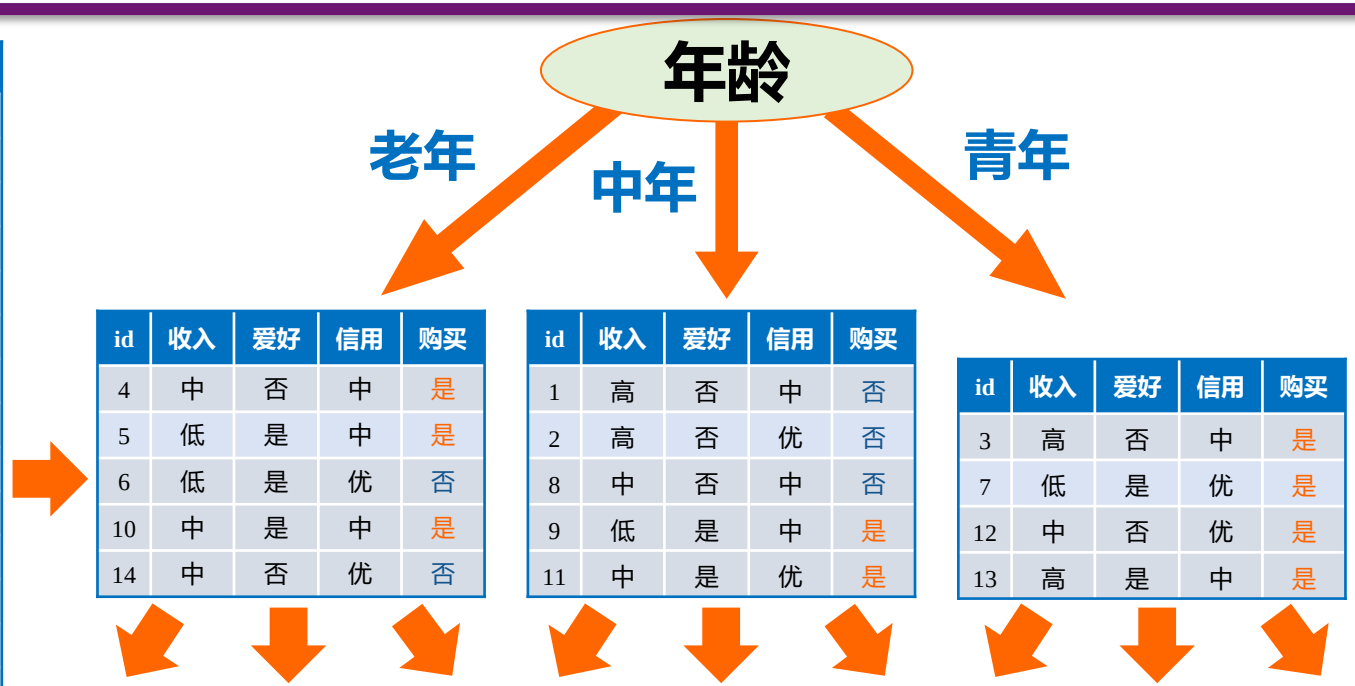


构造决策树

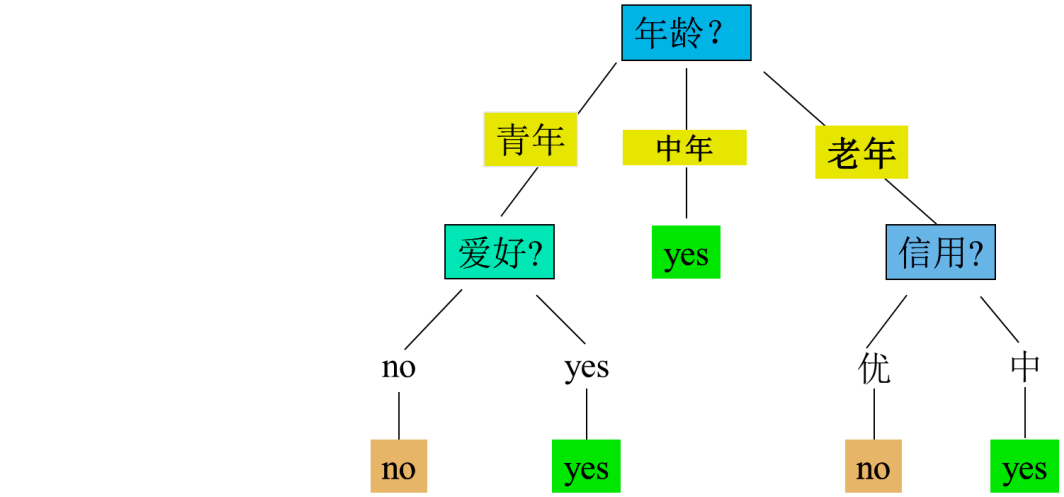
- Hunt算法采用贪心策略构建决策树
  - 在选择划分数据的属性时，采取一系列局部最优决策来构造决策树。
- 决策树归纳的设计问题
  - 如何分裂训练记录
    - 怎样为不同类型的属性指定测试条件?
    - 怎样评估每种测试条件?
  - 如何停止分裂过程

构造决策树

id	年龄	收入	爱好	信用	购买
1	青	高	否	中	否
2	青	高	否	优	否
3	中	高	否	中	是
4	老	中	否	中	是
5	老	低	是	中	是
6	老	低	是	优	否
7	中	低	是	优	是
8	青	中	否	中	否
9	青	低	是	中	是
10	老	中	是	中	是
11	青	中	是	优	是
12	中	中	否	优	是
13	中	高	是	中	是
14	老	中	否	优	否



采用信息增益等准则继续往下分裂，直到数据都属于同一类



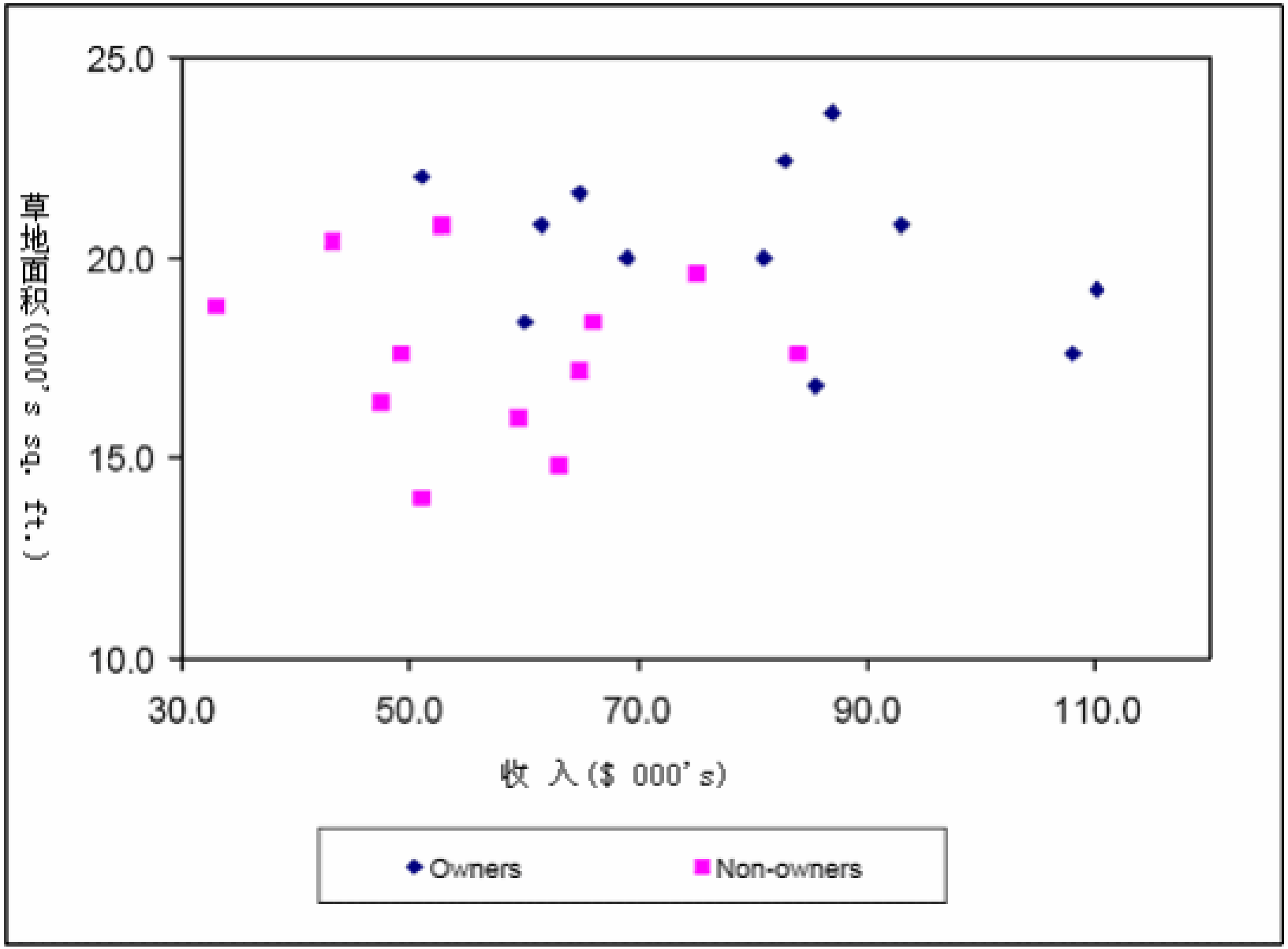
构造决策树 —— 一个例子

割草机制造商意欲发现一个把城市中的家庭分成那些愿意购买乘式割草机和不愿意购买的两类的方法。在这个城市的家庭中随机抽取24个家庭作为样本。

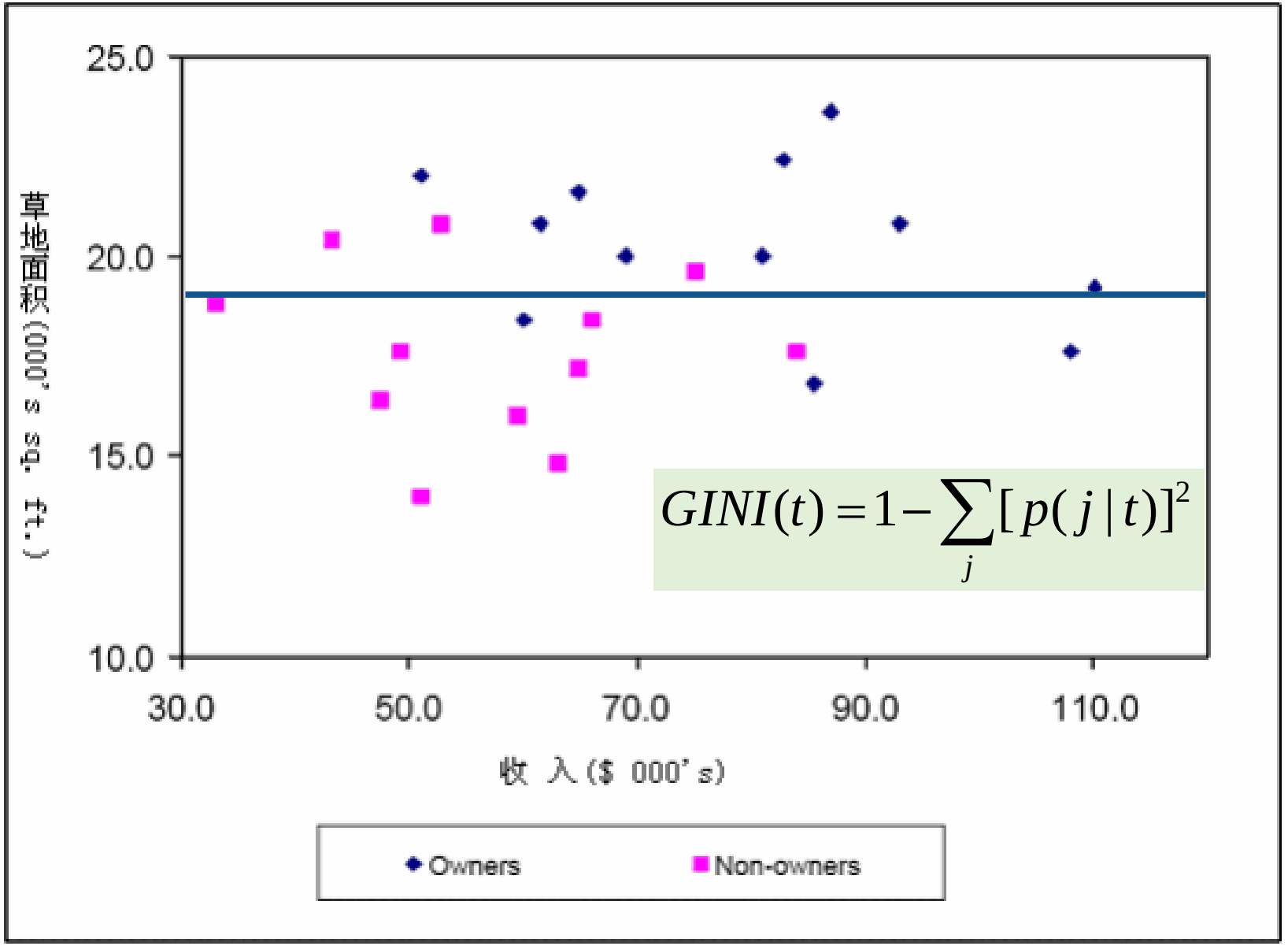
- 自变量是收入和草地面积
- 类别变量是：拥有和没有割草机。

id	收入	草地面积	拥有
1	60	18.4	是
2	85.5	16.8	是
3	64.8	21.6	是
4	61.5	20.8	是
5	87	23.6	是
6	110.1	19.2	是
7	108	17.6	是
17	84	17.6	否
18	49.2	17.6	否
19	59.4	16	否
20	66	18.4	否
21	47.4	16.4	否
22	33	18.8	否
23	51	14	否
24	63	14.8	否

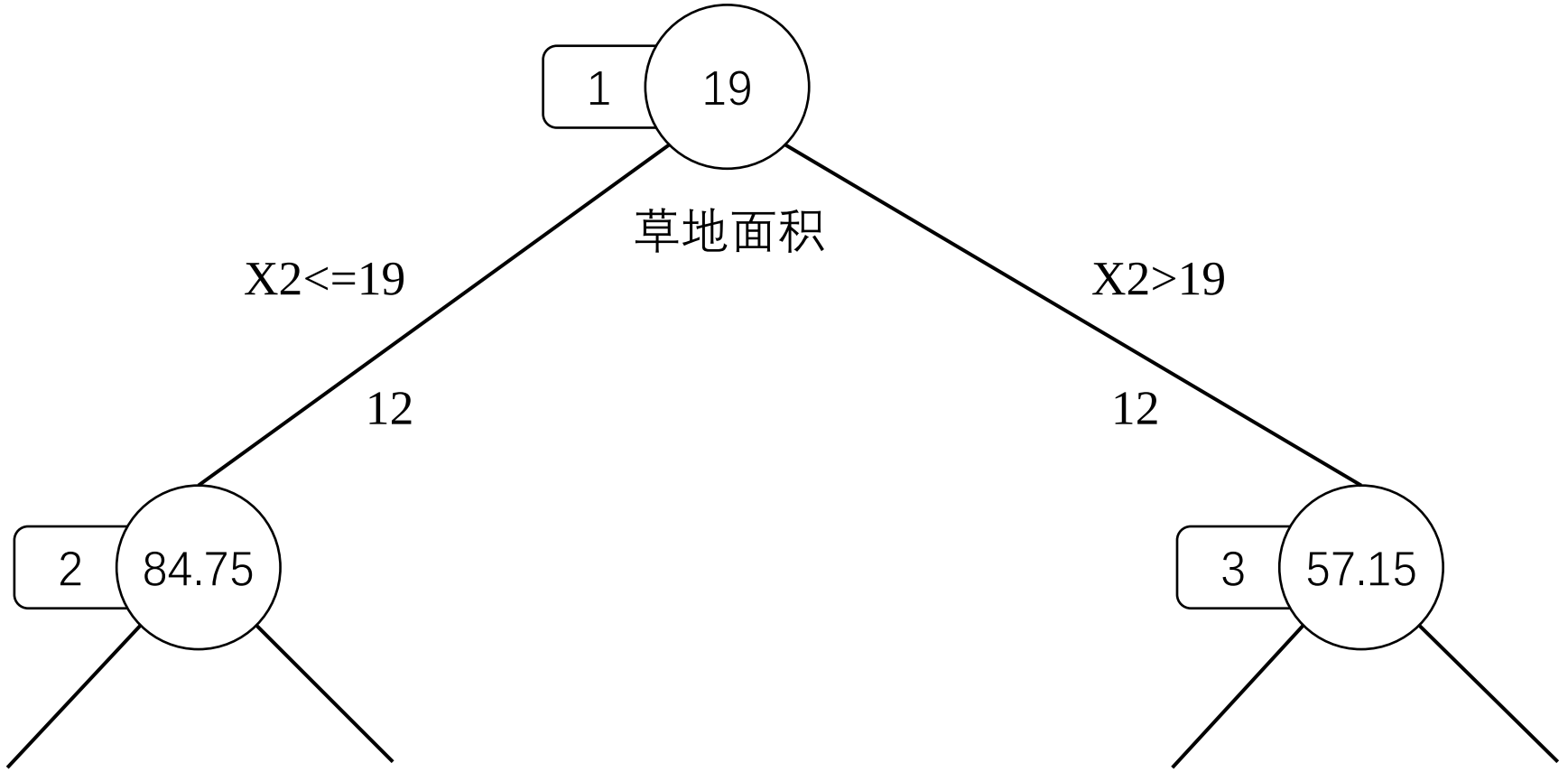
构造决策树 —— 一个例子


25.0
20.0
15.0
10.0
草地面积(000's sq. ft.)
30.0
50.0
70.0
90.0
110.0
收 入($ 000's)
Owners
Non-owners

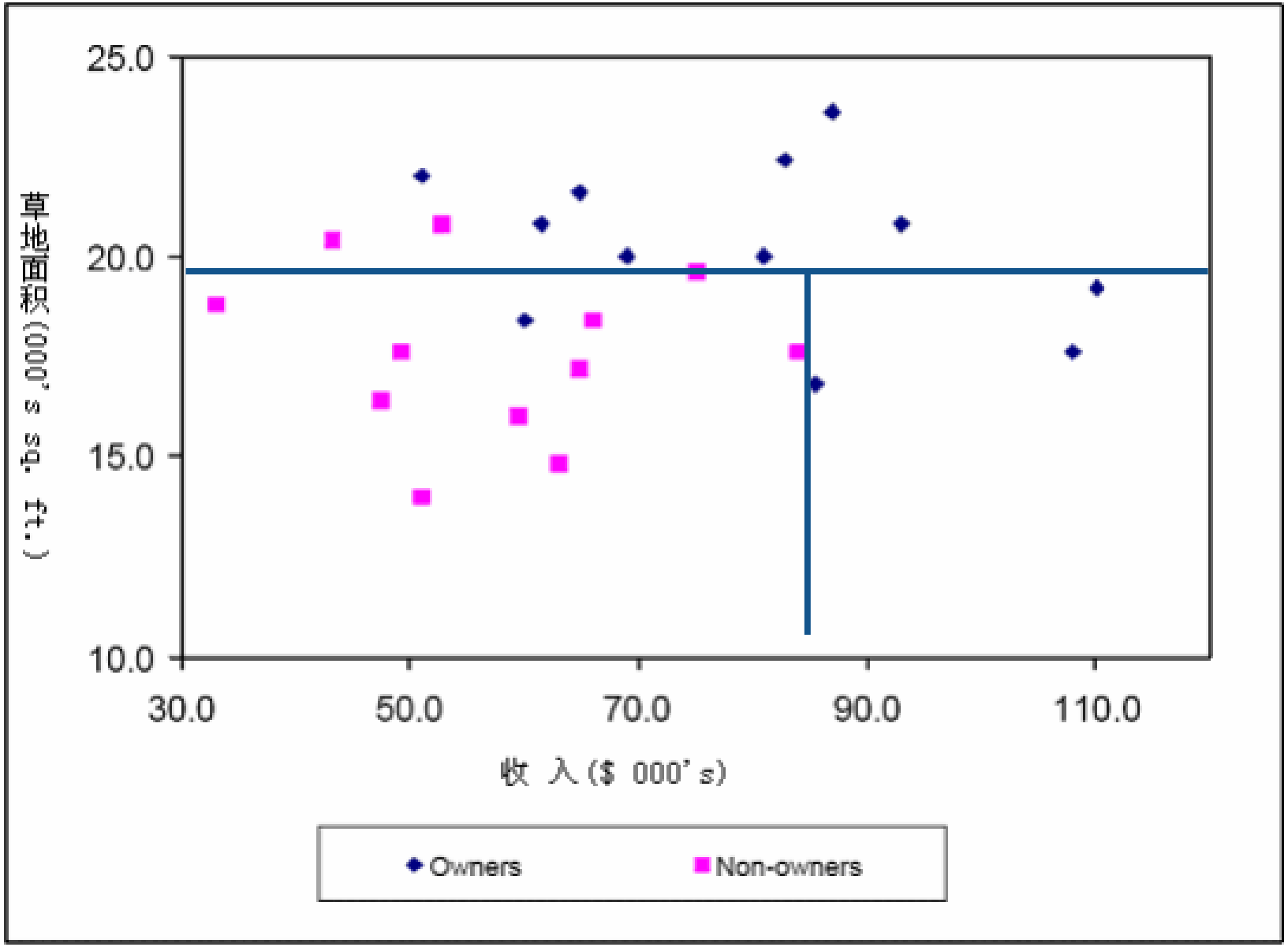
构造决策树 —— 一个例子


25.0
20.0
15.0
10.0
草地面积(000's sq. ft.)
30.0
50.0
70.0
90.0
110.0
收 入($ 000's)
Owners
Non-owners
$GINI(t) = 1 - \sum_j [p(j|t)]^2$

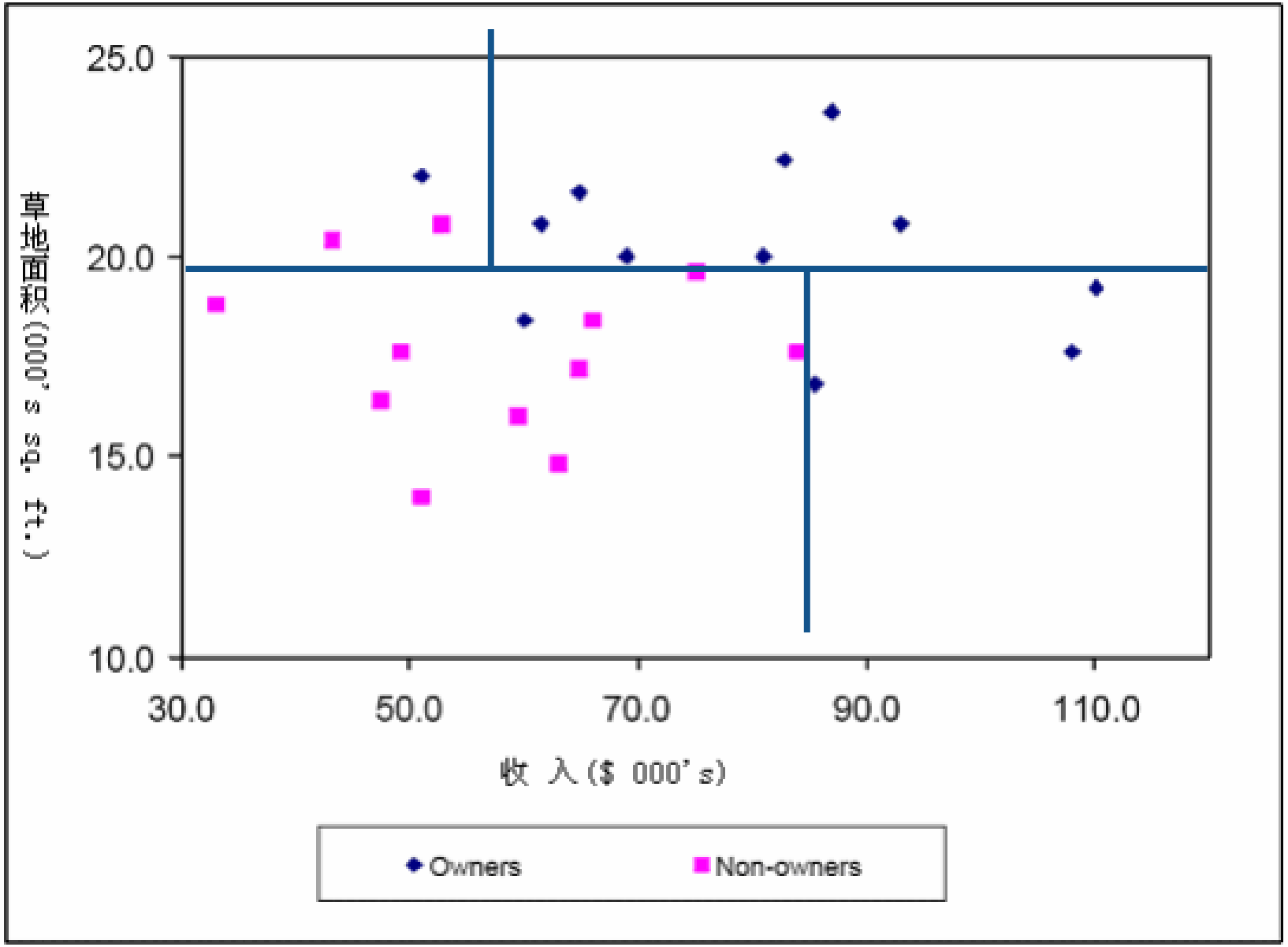
构造决策树 —— 一个例子



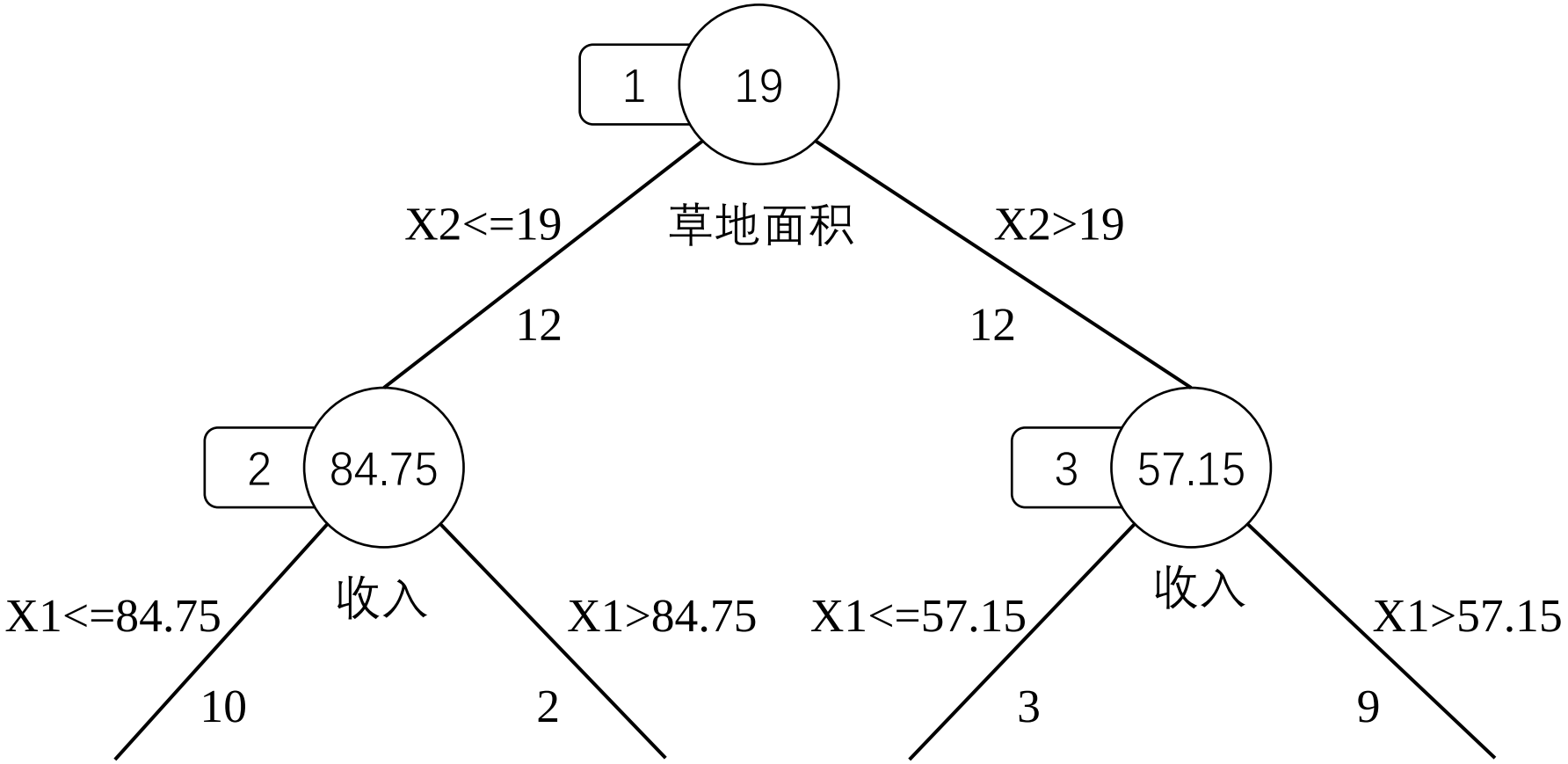
构造决策树 —— 一个例子


25.0
20.0
15.0
10.0
30.0
50.0
70.0
90.0
110.0
草地面积(000's sq. ft.)
收 入($ 000's)
Owners
Non-owners

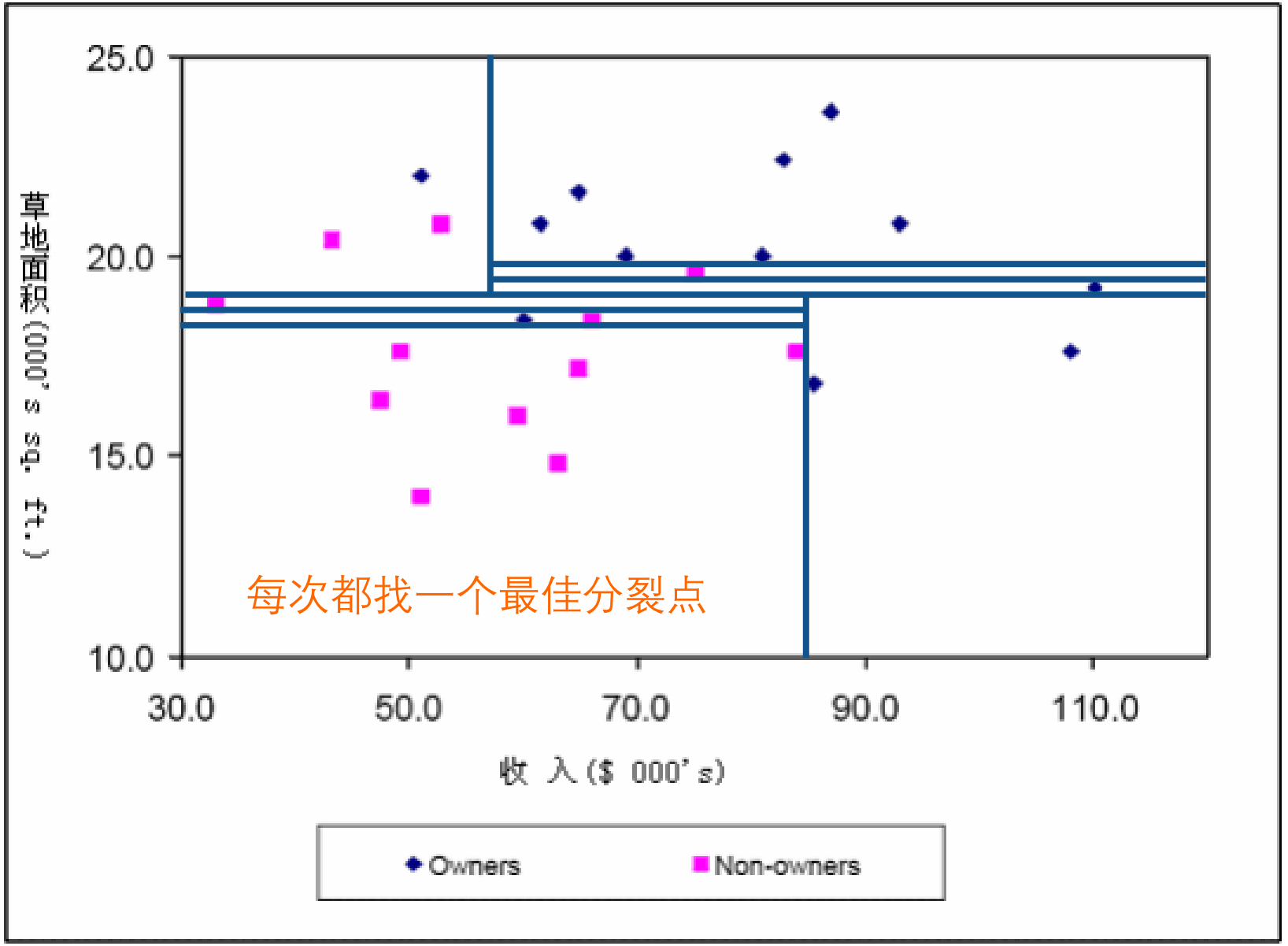
构造决策树 —— 一个例子


25.0
20.0
15.0
10.0
草地面积(000's sq. ft.)
30.0
50.0
70.0
90.0
110.0
收 入($ 000's)
Owners
Non-owners

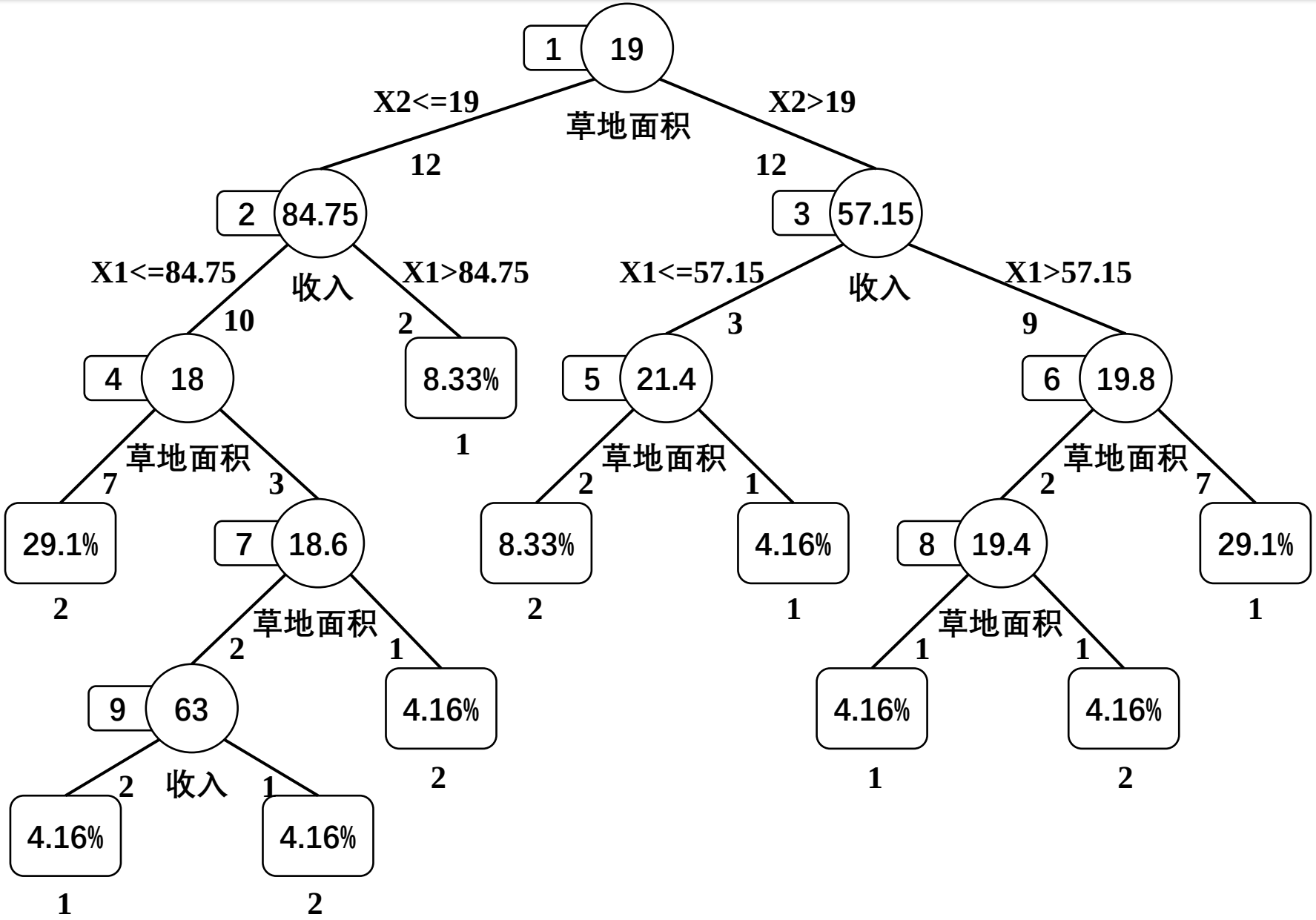
构造决策树 —— 一个例子


1
19
X2<=19
草地面积
X2>19
12
12
2
84.75
3
57.15
X1<=84.75
收入
X1>84.75
X1<=57.15
收入
X1>57.15
10
2
3
9

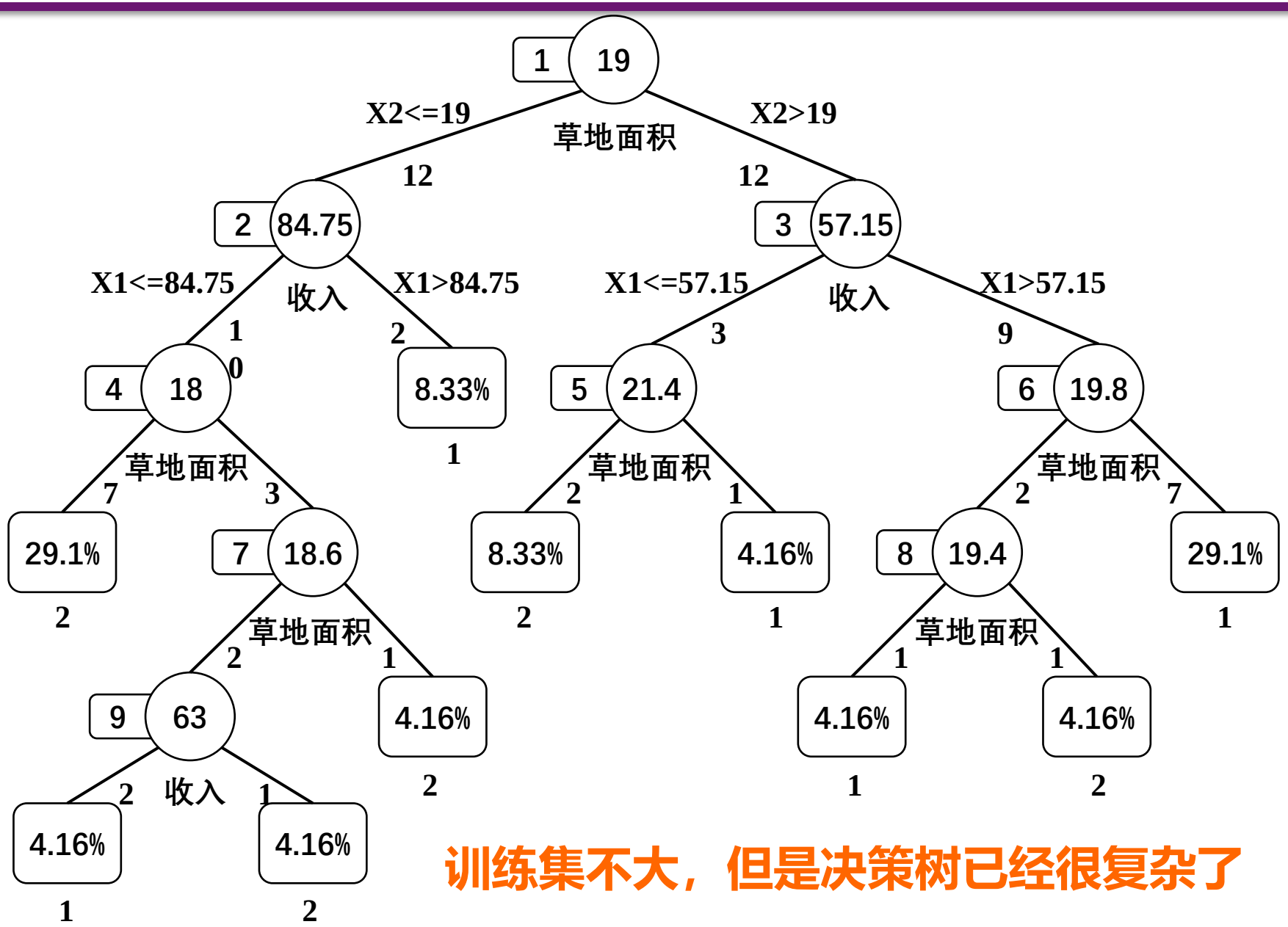
构造决策树 —— 一个例子


25.0
20.0
15.0
10.0
草地面积(000's sq. ft.)
30.0
50.0
70.0
90.0
110.0
收 入($ 000's)
每次都找一个最佳分裂点
Owners
Non-owners

构造决策树 —— 一个例子（CART算法）



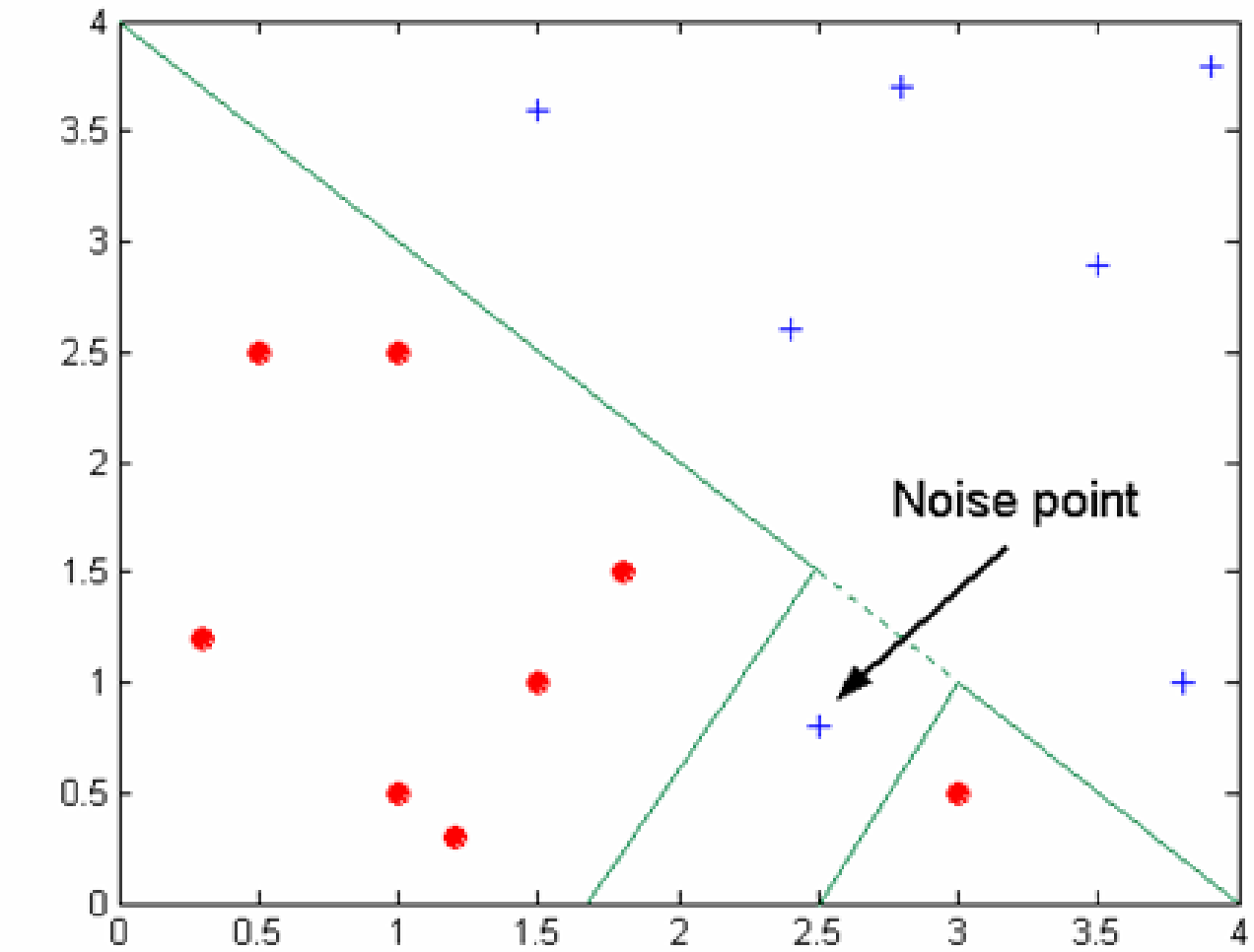
构造决策树 —— 一个例子（CART算法）



训练集不大，但是决策树已经很复杂了

构造决策树 —— 复杂的决策树带来过拟合问题

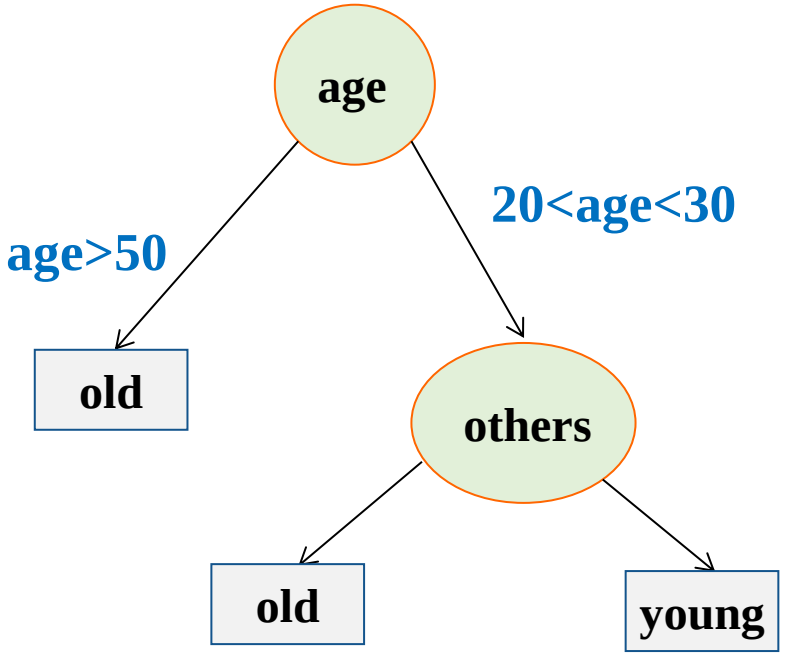
- 过拟合问题



构造决策树 —— 复杂的决策树带来过拟合问题

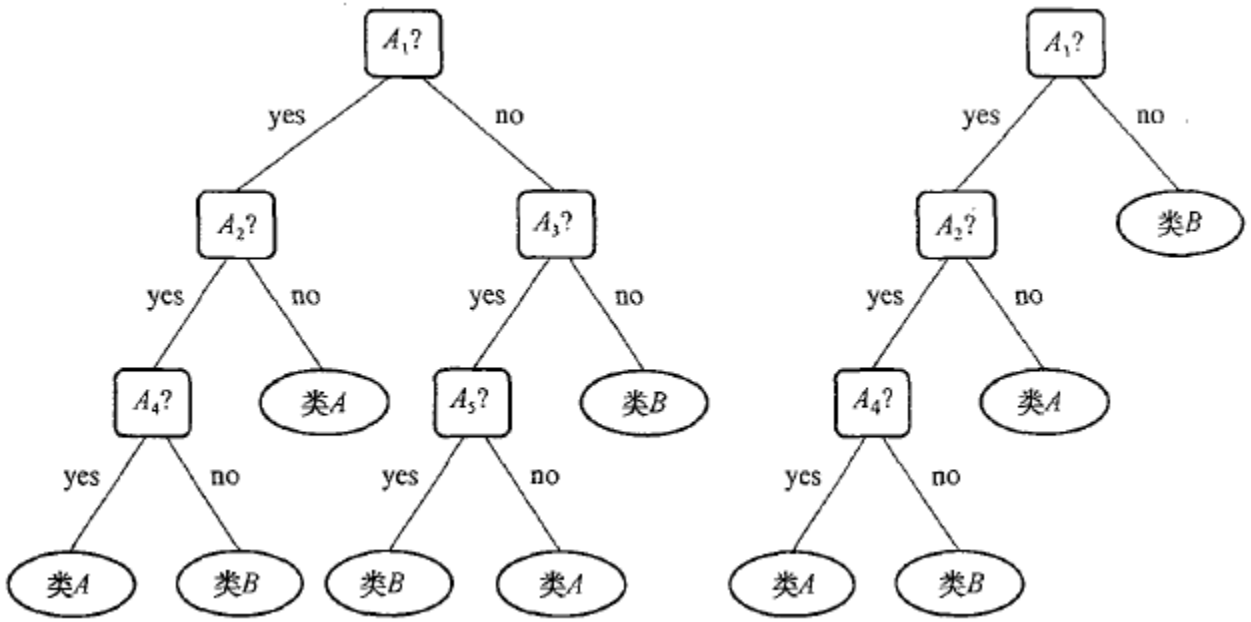
- 过拟合问题

age	class
>50	old
>50	old
>50	old
20<age<30	old
20<age<30	young
20<age<30	young
20<age<30	young



构造决策树 —— 剪枝方法

- Two approaches to avoid overfitting
  - Prepruning:如果划分带来的信息增益、Gini指标等低于阈值，或元组数目低于阈值，则停止这次划分
  - Postpruning：从完全生长的树中剪去树枝——得到一个逐步修剪树【提示：度量分类器性能】



一棵未剪枝的决策树和它剪枝后的版本

总结

- 特点:
  - 决策树是一种构建分类（回归）模型的非参数方法
  - 不需要昂贵的的计算代价
  - 决策树相对容易解释
  - 决策树是学习离散值函数的典型代表
  - 决策数对于噪声的干扰具有相当好的鲁棒性
  - 冗余属性不会对决策树的准确率造成不利影响
  - 数据碎片问题：随着树的生长，可能导致叶结点记录数太少，对于叶结点代表的类，不能做出具有统计意义的判决
  - 子树可能在决策树中重复多次，使决策树过于复杂
  - 决策树无法学习特征之间的线性关系，难以完成特征构造