

数据挖掘

第六章 决策树分类

目录

1. 基本概念
2. 决策树
3. 决策树构建

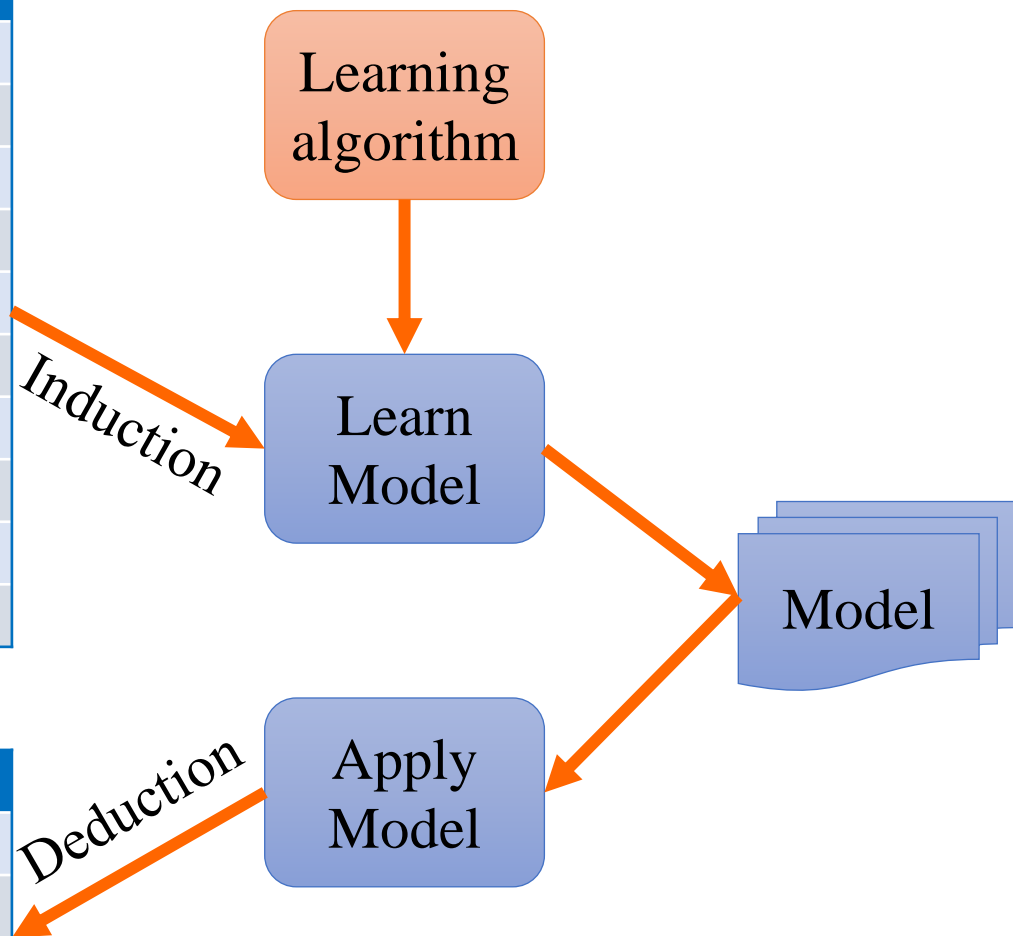
分类

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



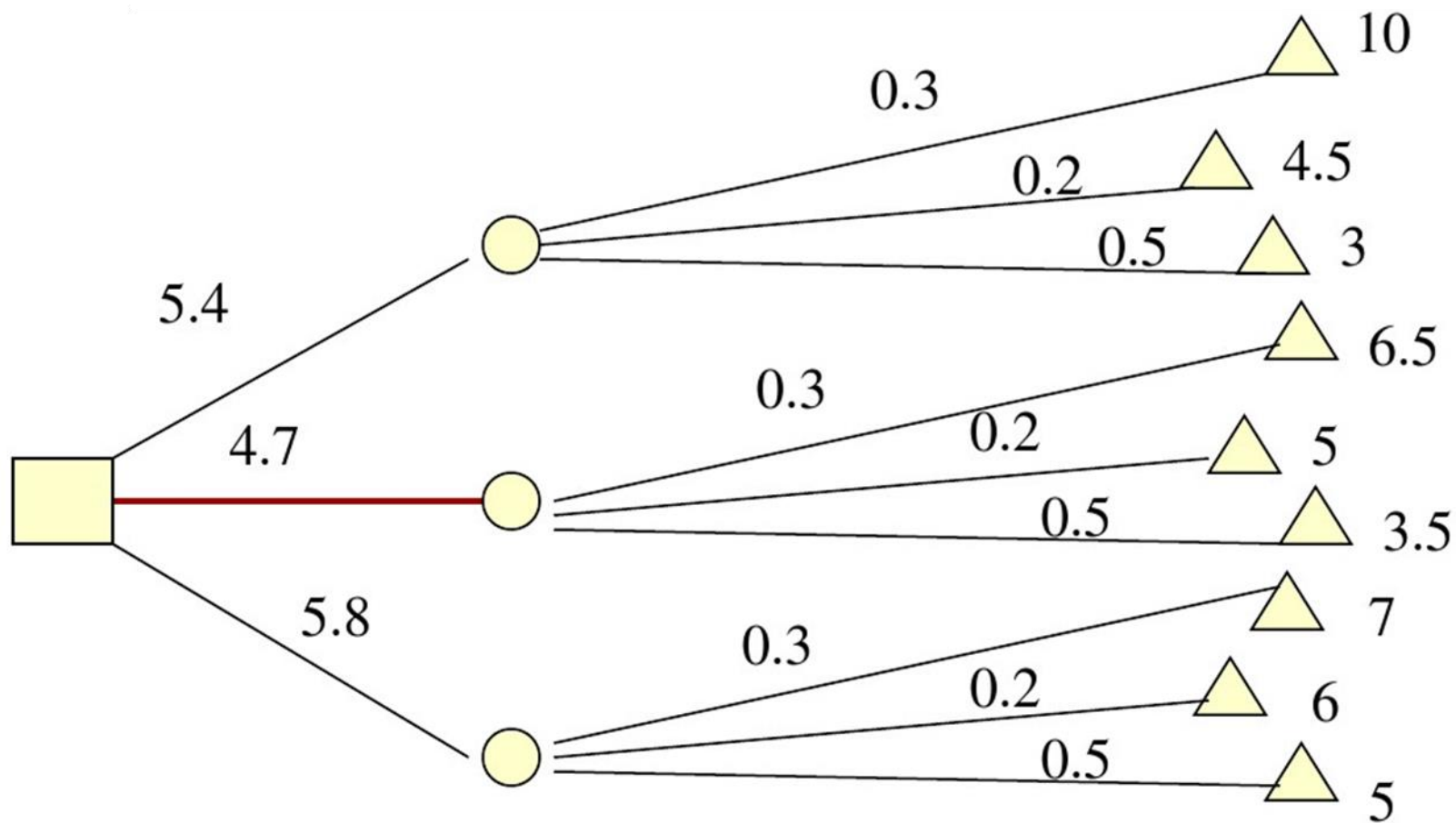
目录

1. 基本概念
2. 决策树
3. 决策树构建

2.1 引言

- 决策树是一种直观的**风险型决策**问题求解法。
 - 例p342：某部队接到上级命令，要求在最短的时间内赶到140 千米以外的某个山口阻击长途奔袭的敌人。可供选择的行军路线有**1、2、3**条道路。
 - 这三条道路刚刚遭到敌军的空袭，据估计每条道路受到“严重破坏”、“一般破坏”和“轻度破坏”的概率分别为**0.3、0.2、0.5**。在受到不同程度破坏的条件下，部队通过各条道路所需的时间分别近似为**10、4.5、3；6.5、5、3.5；7、6、5**小时。
 - 问该部队应选择哪一条路线作为行军路线？

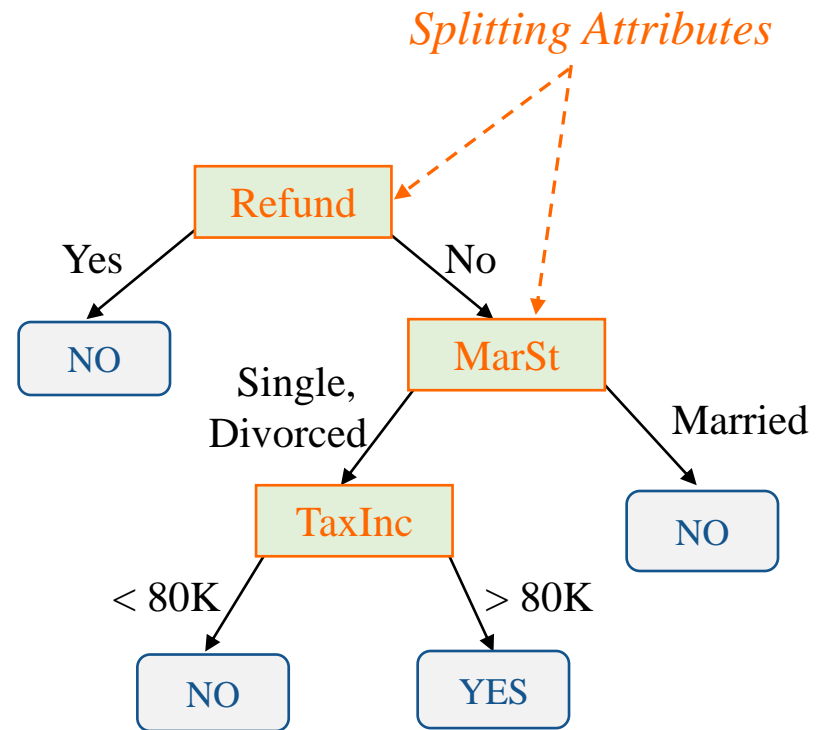
2.1 引言



决策分支画成图形很像一棵树的枝干，故称决策树

2.1 引言

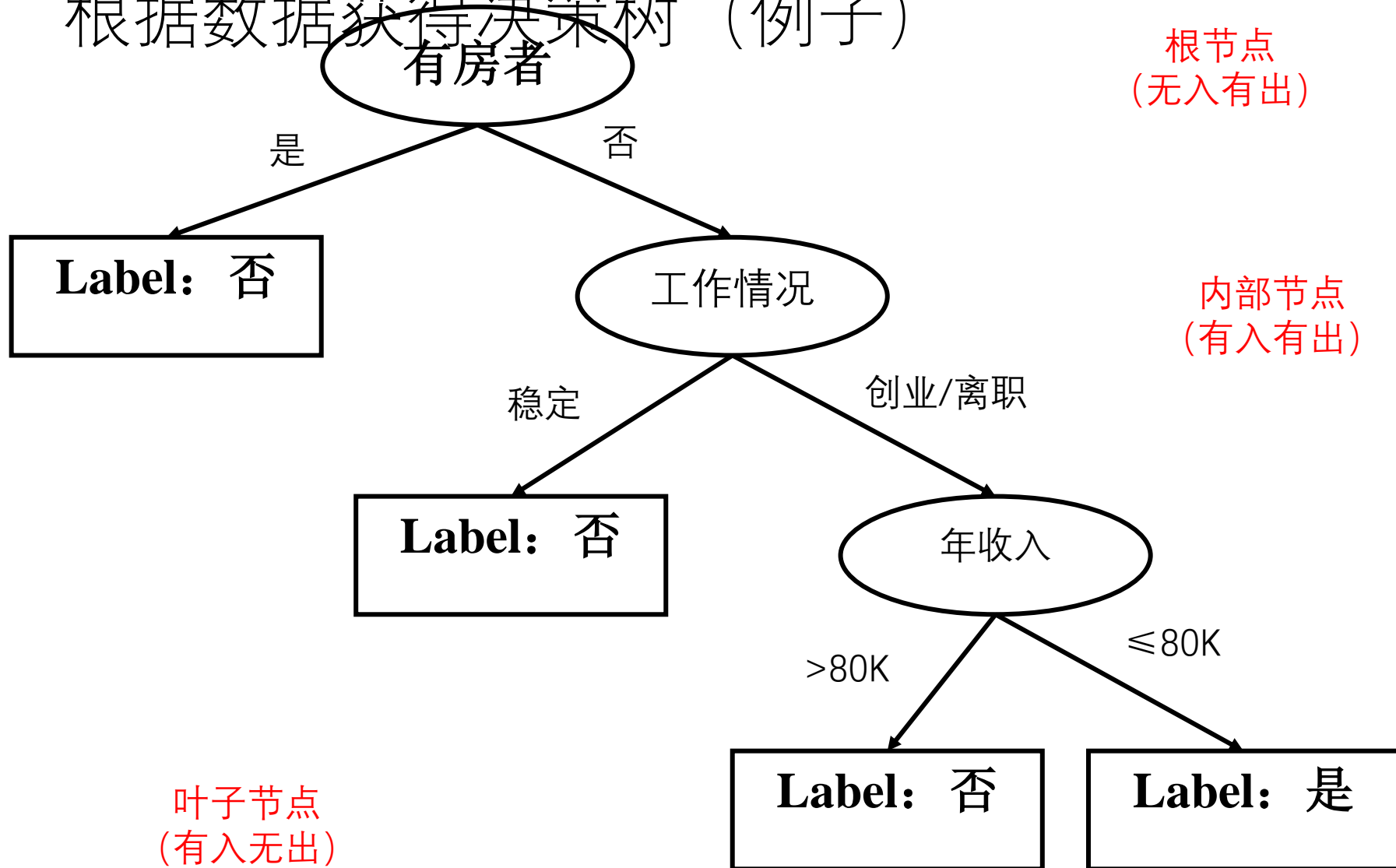
- 树状结构，可以很好的对数据进行分类；
- 决策树的根节点到叶节点的每一条路径构建一条规则；
- 具有互斥且完备的特点，即每一个样本均被且只能被一条路径所覆盖；
- 只要提供的数据量足够庞大真实，通过数据挖掘模式，就可以构造决策树。



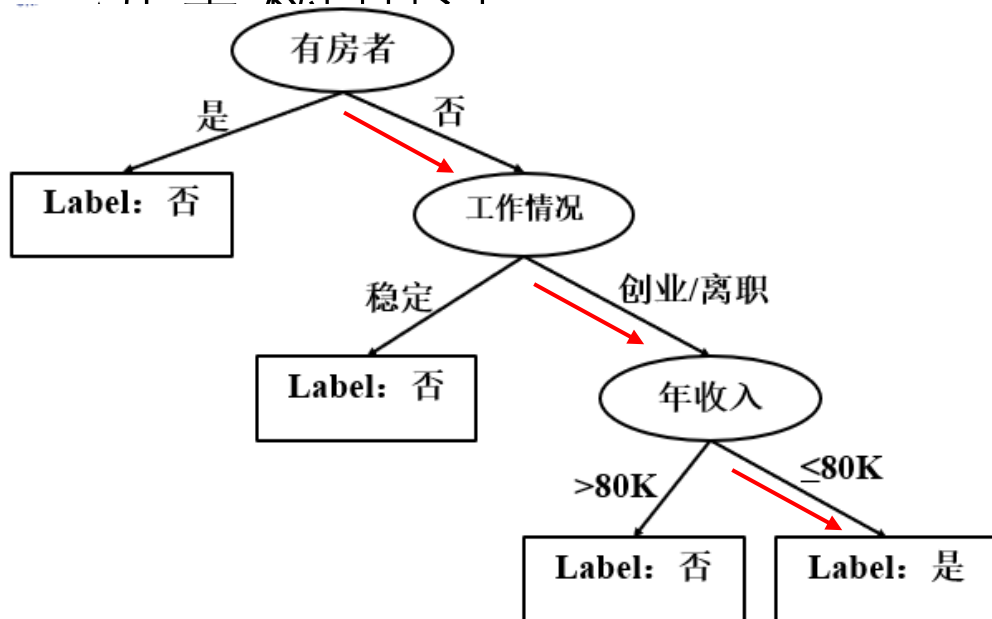
什么是决策树

• 月	Id	有房者	工作情况	年收入	拖欠贷款者
	1	是	创业	125K	否
	2	否	稳定	100K	否
	3	否	创业	70K	否
	4	是	稳定	120K	否
	5	否	离职	95K	是
	6	否	稳定	60K	否
	7	是	离职	220K	否
	8	否	创业	85K	是
	9	否	稳定	75K	否
	10	否	创业	90K	是

根据数据获得决策树 (例子)



决策树模型



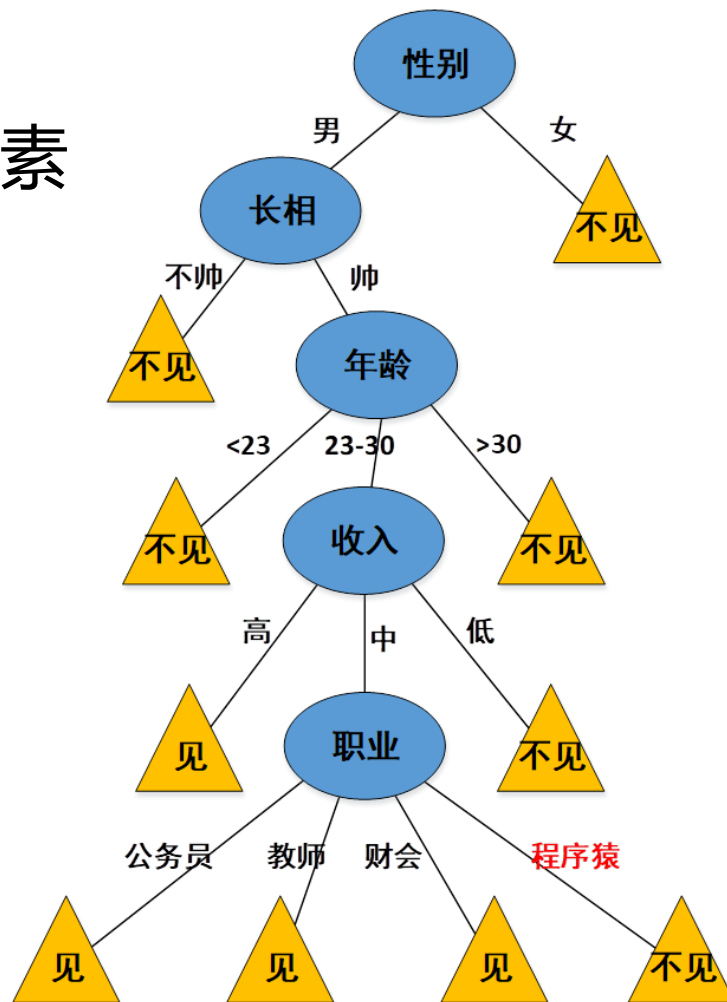
用于预测新加入数据
的目标值

Id	有房者	工作情况	年收入	拖欠贷款者
16	否	创业	70K	???



一个决策树包括如下哪些要素

- ☒ A 测试节点
- ☒ B 分支
- ☒ C 叶子



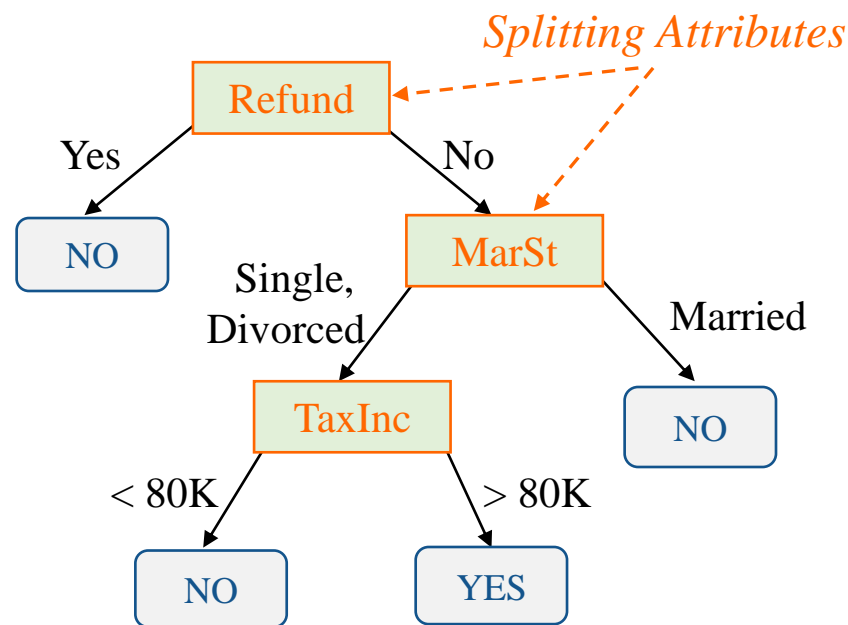
提交

一个决策树的例子 (Different?)

categorical categorical continuous class

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

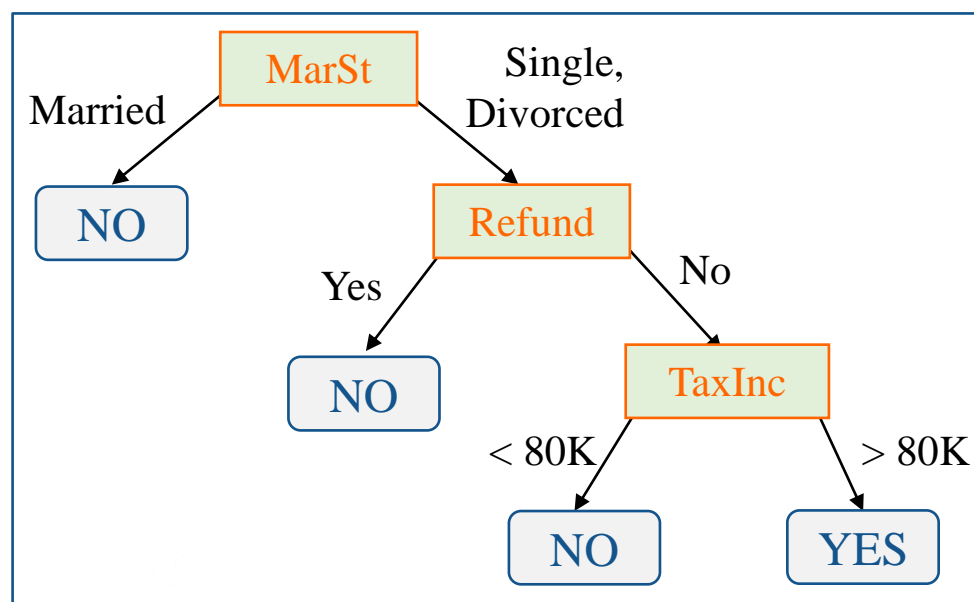
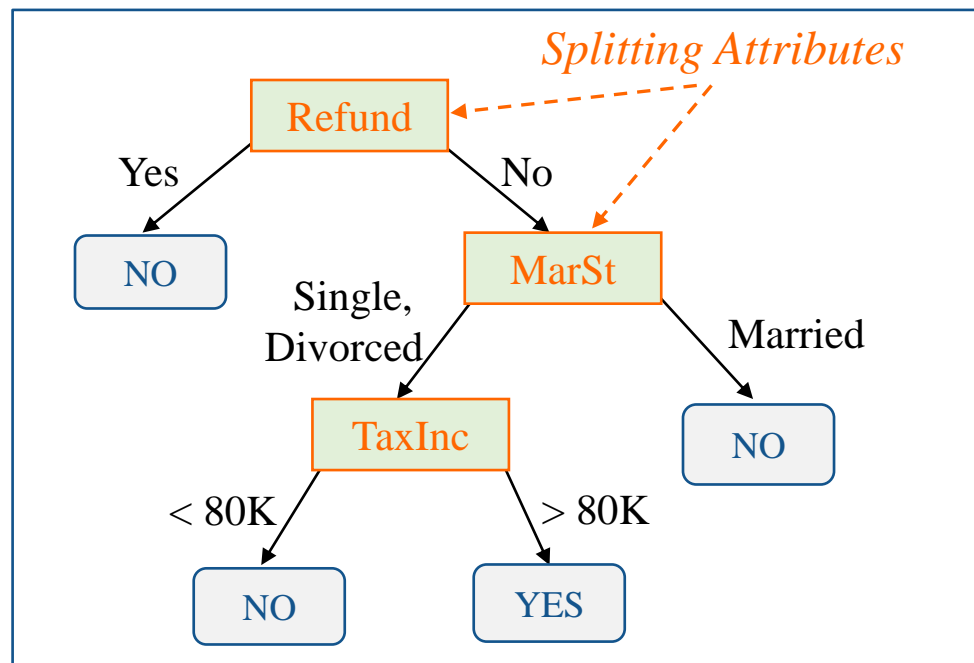
训练数据



模型: 决策树

一个决策树的例子 (Different?)

Tid	Refund	Marital Status	Taxable Income	Cheat
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?)

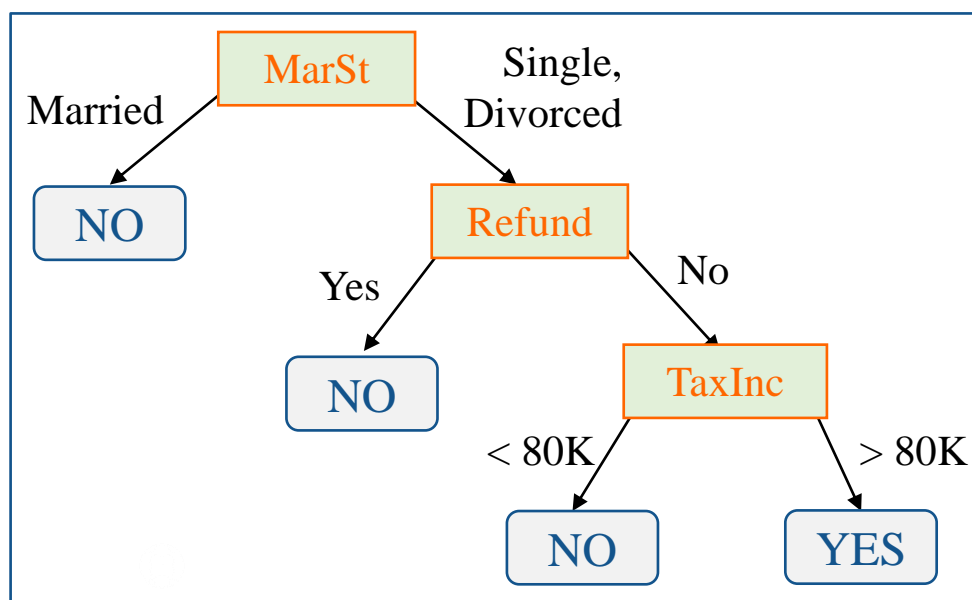
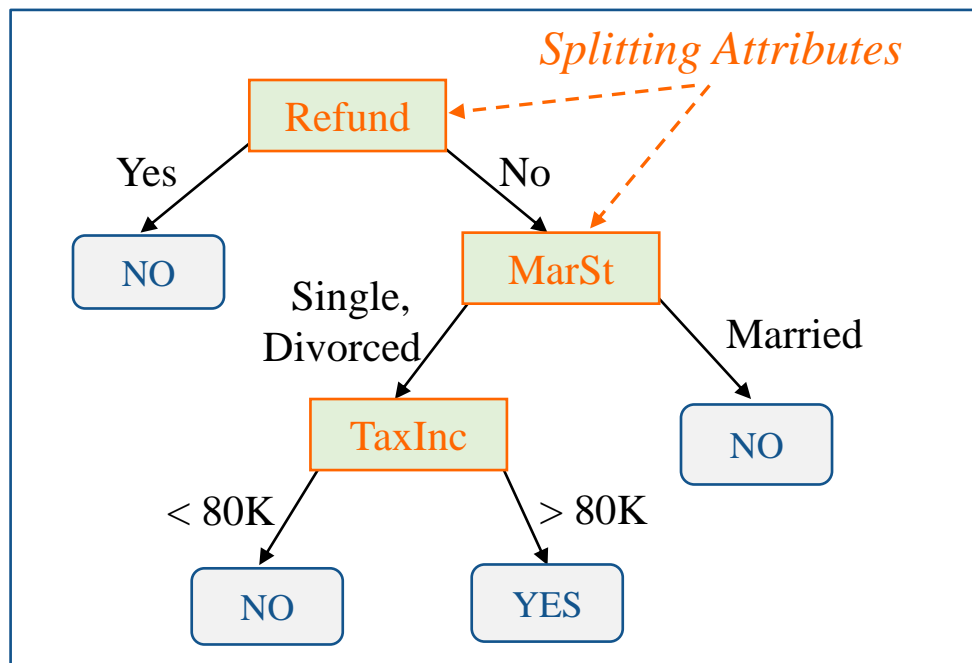
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

categorical

categorical

continuous

class



两棵决策树的属性划分顺序不一样

一个决策树的例子 (Different?)

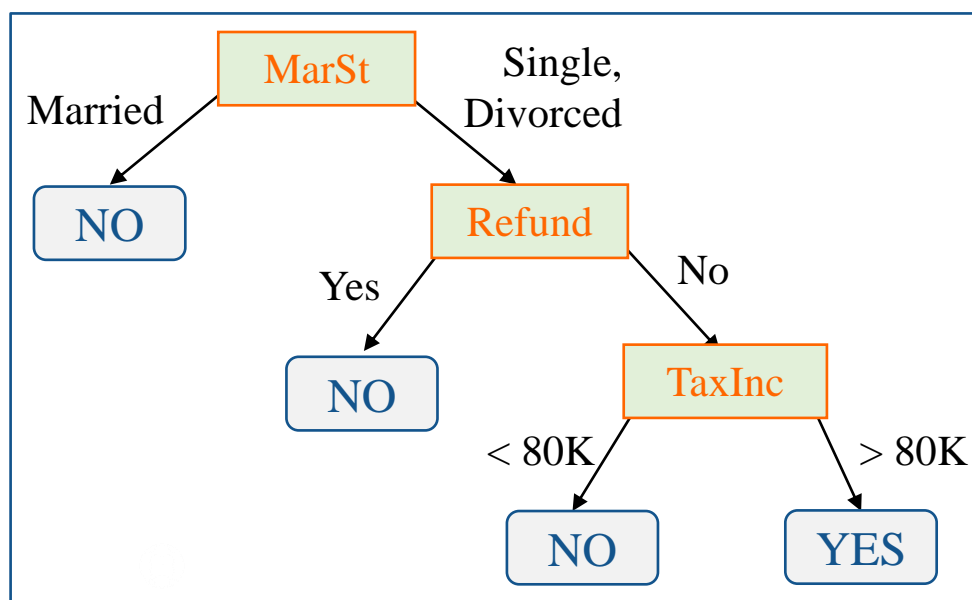
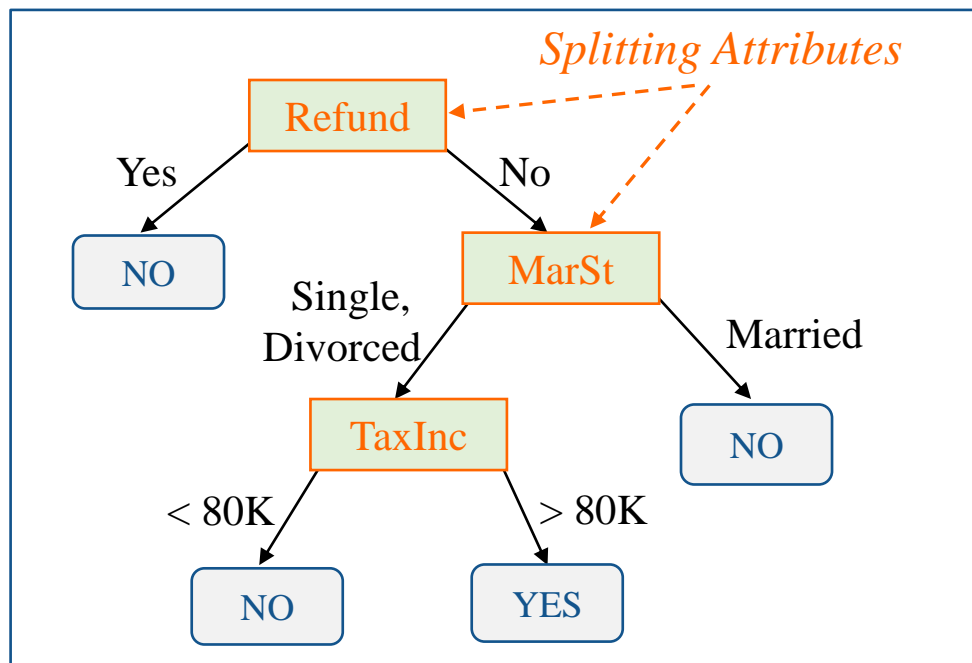
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

categorical

categorical

continuous

class



两棵决策树的属性划分顺序不一样
到底构造哪棵决策树分类效果最好呢?

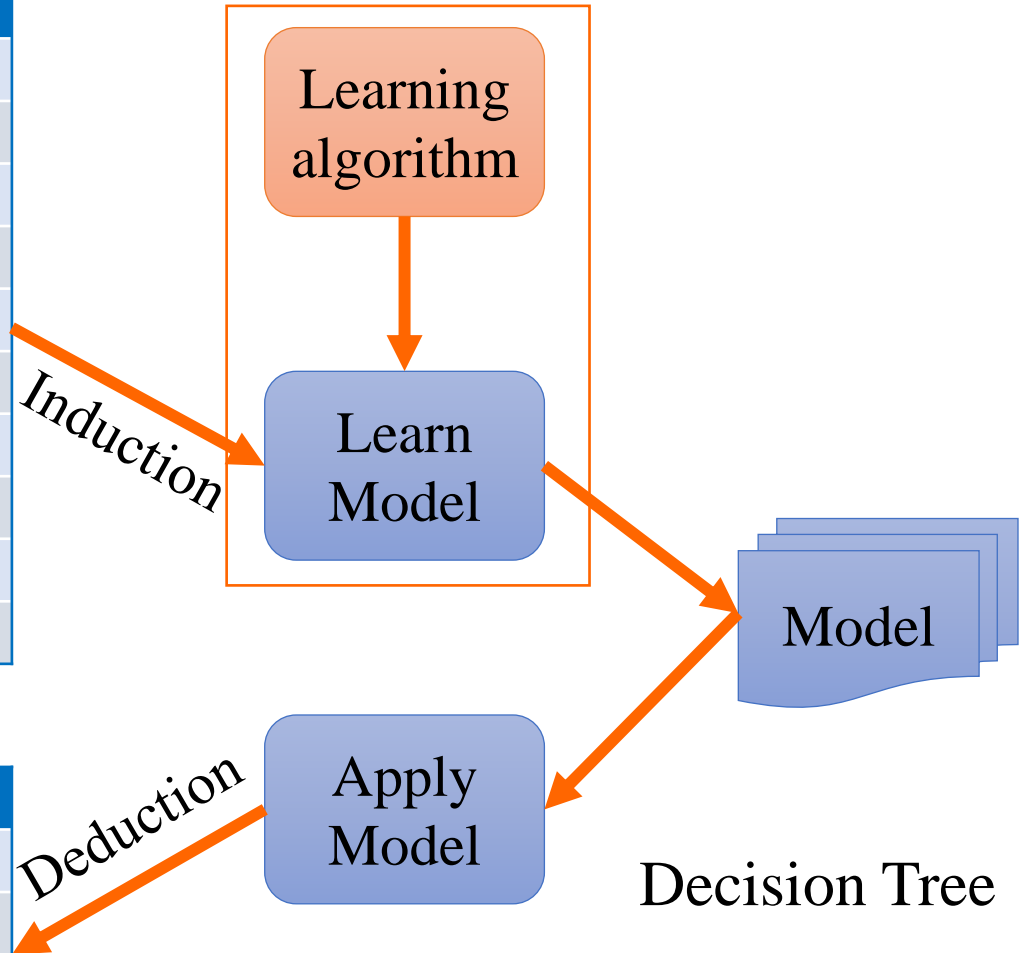
决策树分类

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



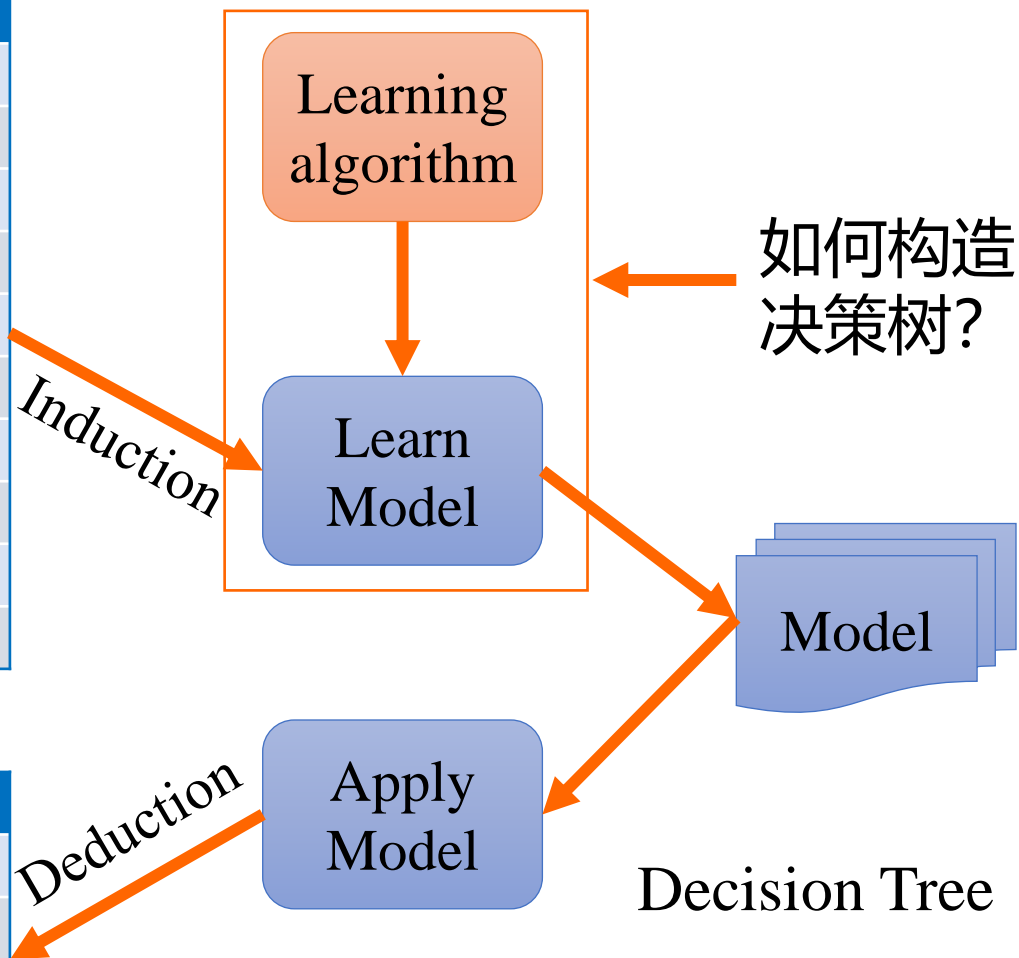
决策树分类

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



目录

1. 基本概念
2. 决策树
3. 决策树构建

3 构造决策树

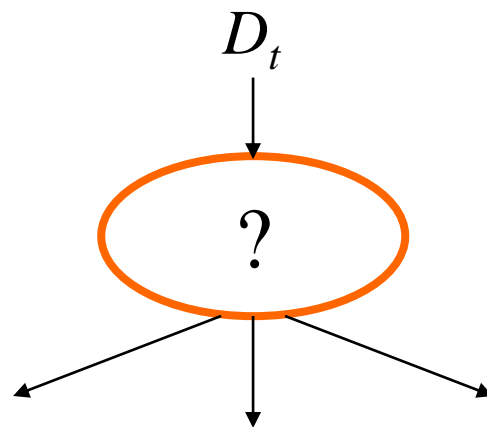
- 有许多决策树算法：
 - **Hunt算法**
 - **信息增益——Information gain** (ID3)
 - **增益比率——Gain ration** (ID3, C4.5)
 - **基尼指数——Gini index** (SLIQ, SPRINT)

3.1 Hunt 算法

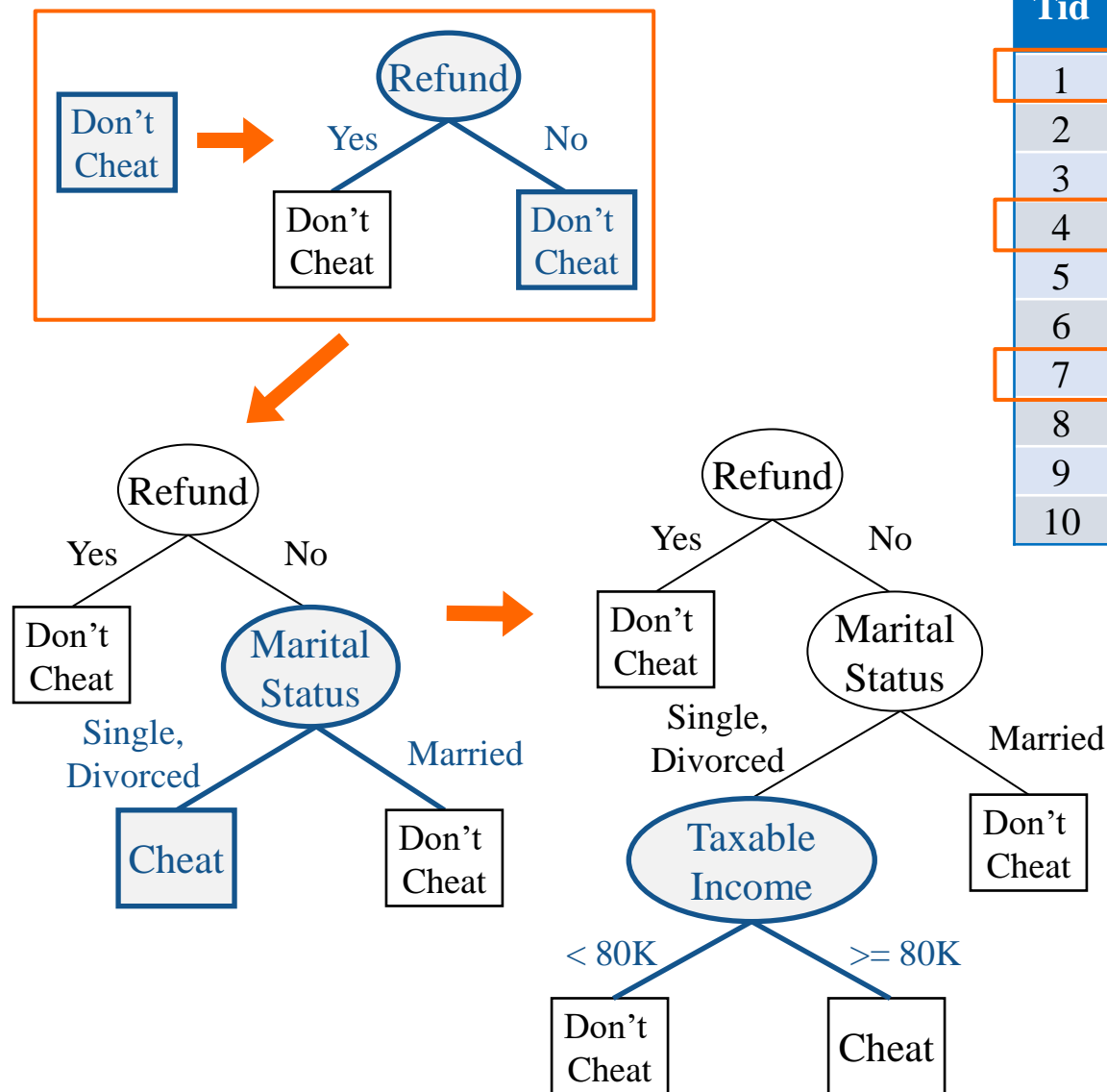
设 D_t 是与节点 t 相关联的训练记录集，
算法步骤：

- 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 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

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



3.1 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：

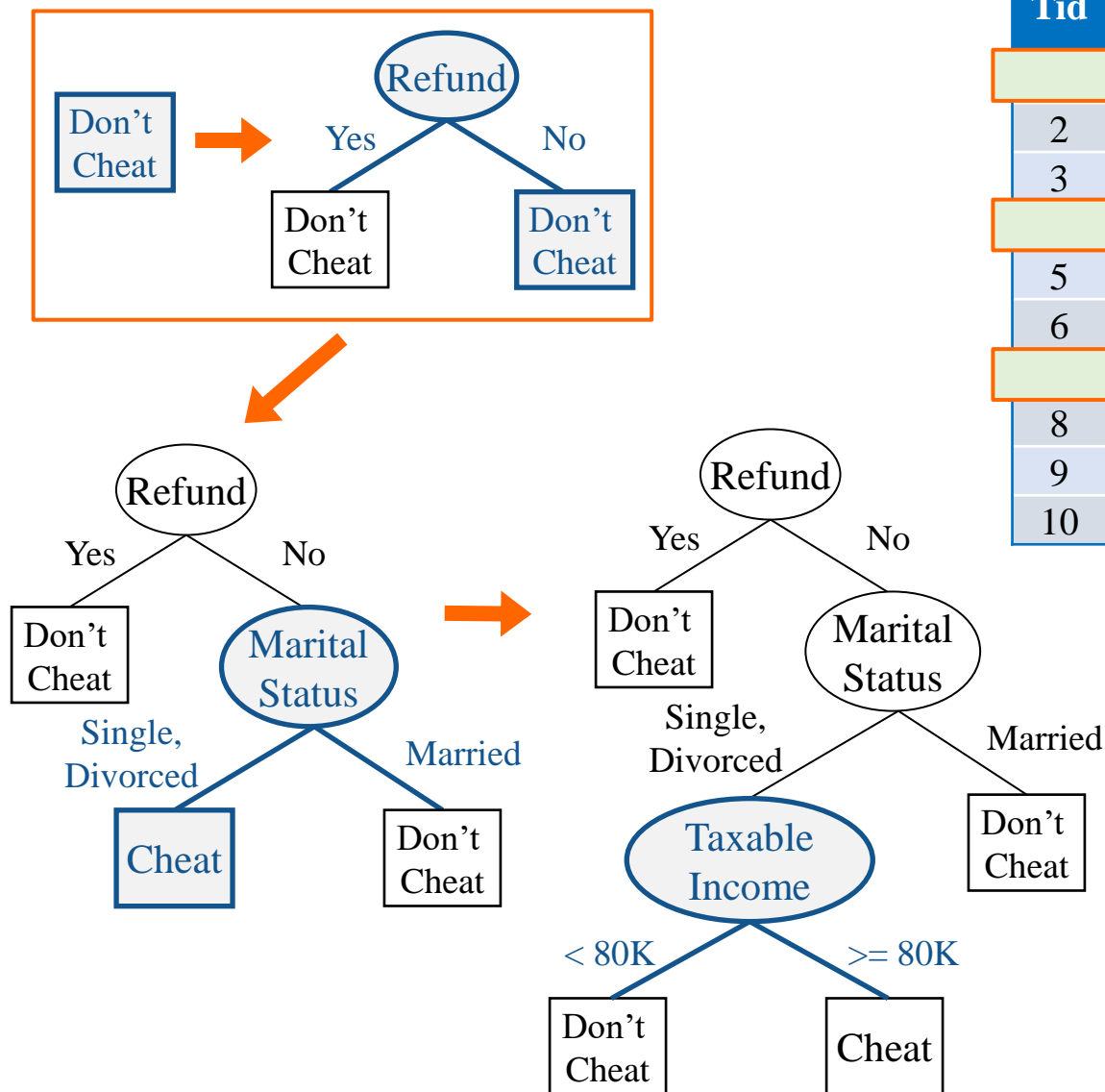
- 如果 D_t 中所有记录都属于同一个类 y_t ，则 t 是叶节点，用 y_t 标记
- 如果 D_t 中包含属于多个类的记录，则**选择一个属性测试条件**，将记录划分成较小的子集
- 对于测试条件的每个输出，创建一个子结点，并根据测试结果将 D_t 中的记录分布到子结点中。然后，对于每个子结点，递归地调用该算法

3.1 Hunt 算法

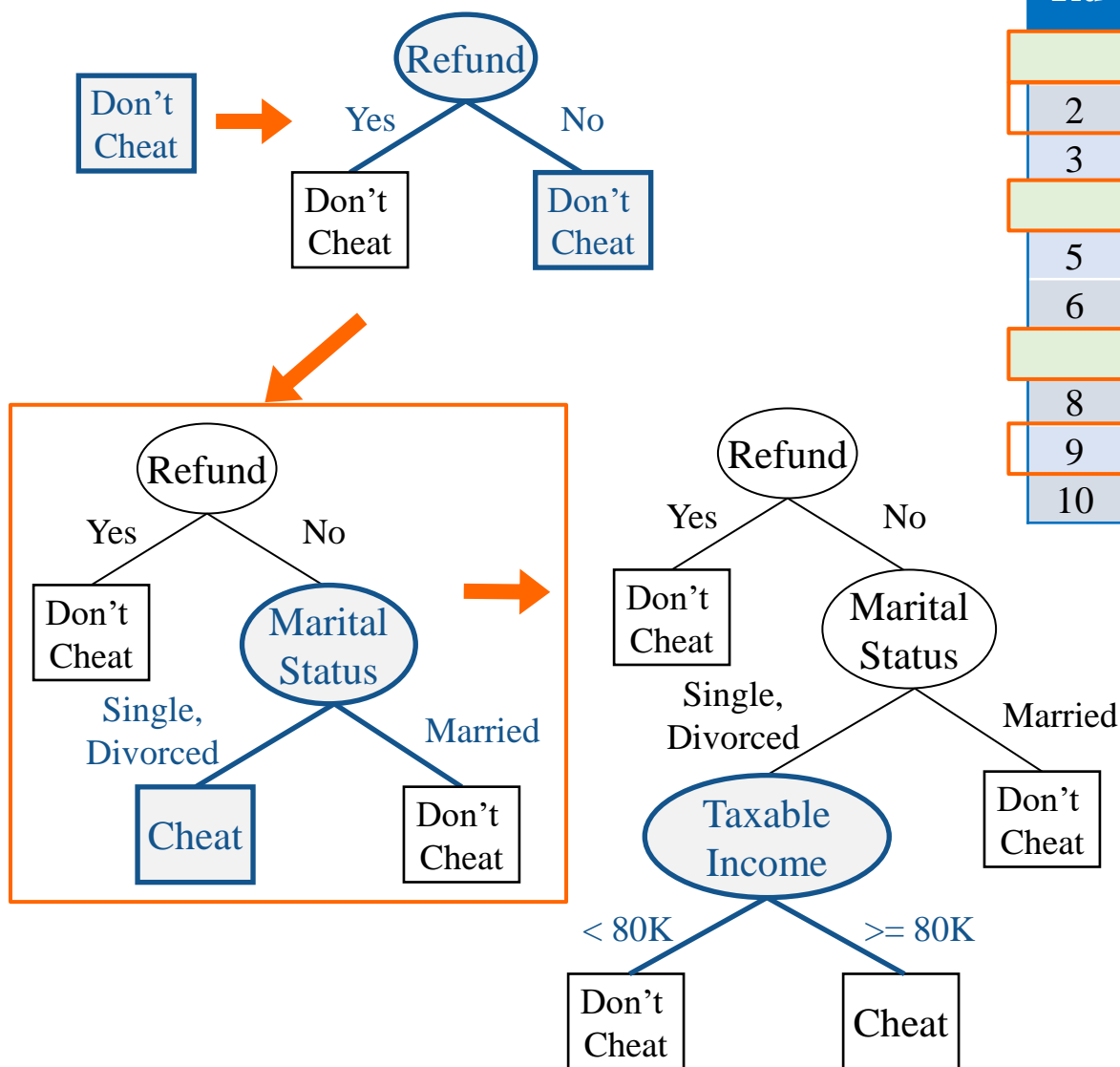
Tid	Refund	Marital Status	Taxable Income	Cheat
2	No	Married	100K	No
3	No	Single	70K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

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



3.1 Hunt 算法

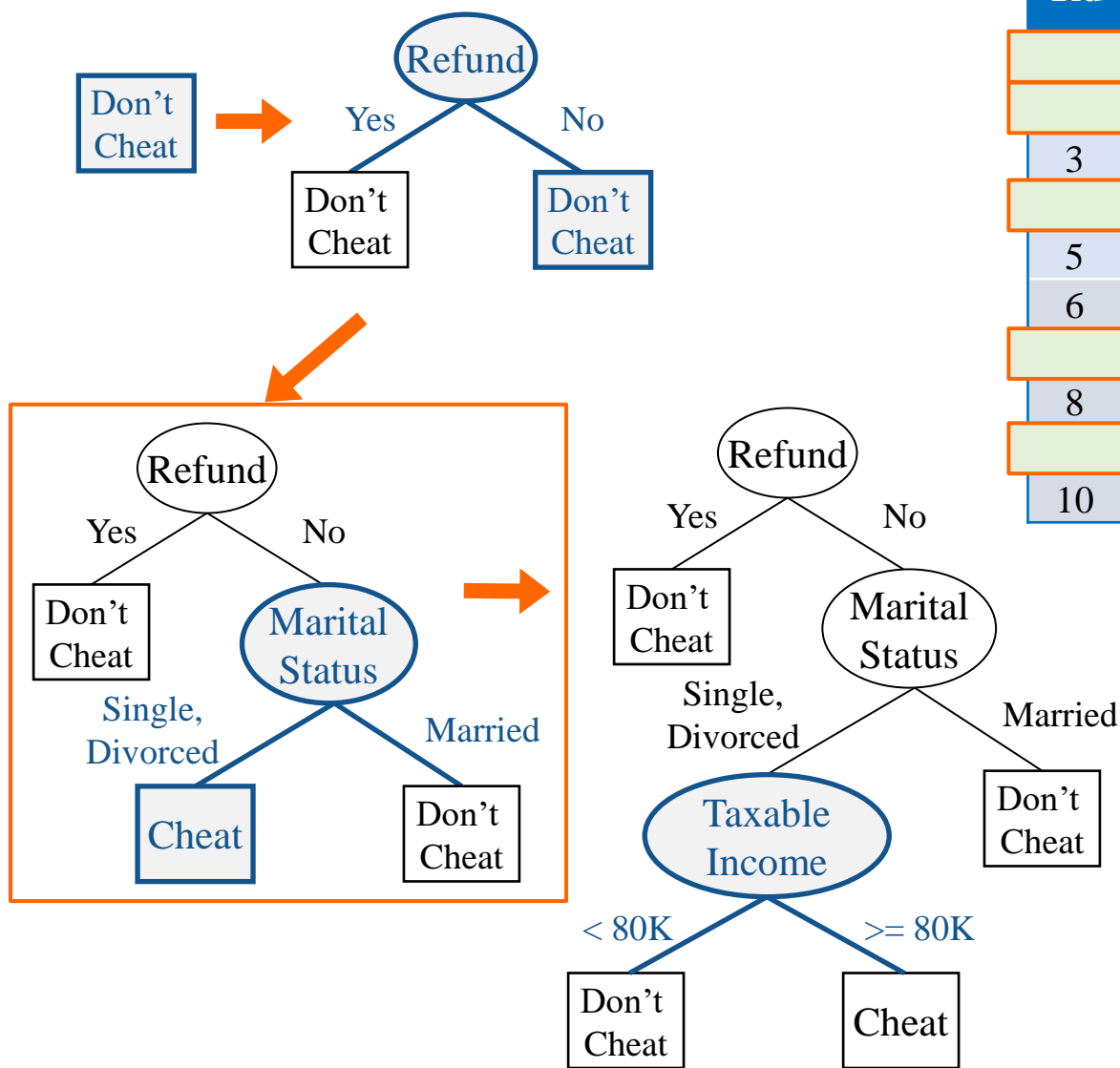


Tid	Refund	Marital Status	Taxable Income	Cheat
2	No	Married	100K	No
3	No	Single	70K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

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

3.1 Hunt 算法

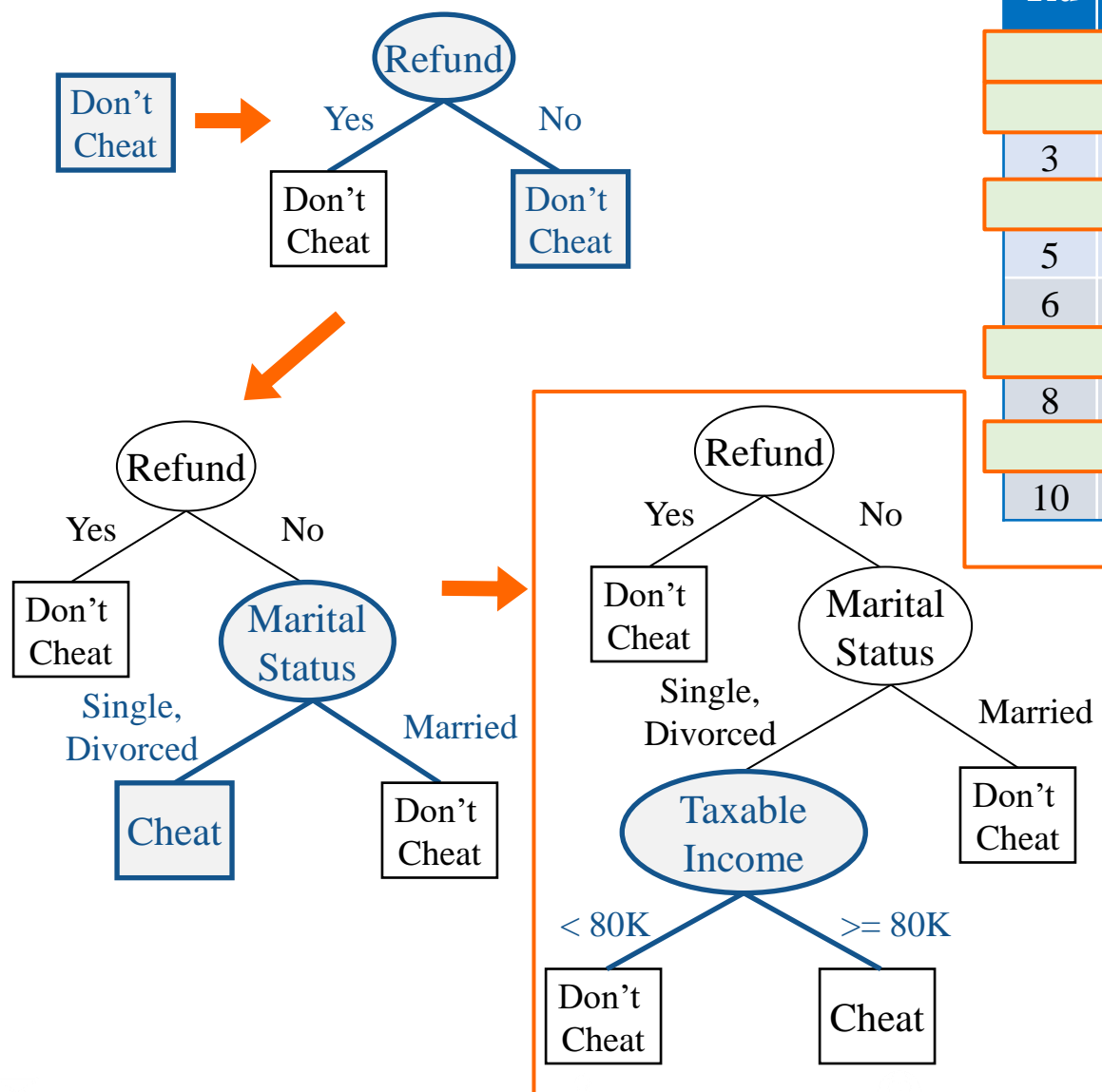


Tid	Refund	Marital Status	Taxable Income	Cheat
3	No	Single	70K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
8	No	Single	85K	Yes
10	No	Single	90K	Yes

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

3.1 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

3.2 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

3.3 怎样为不同类型的属性指定测试条件？

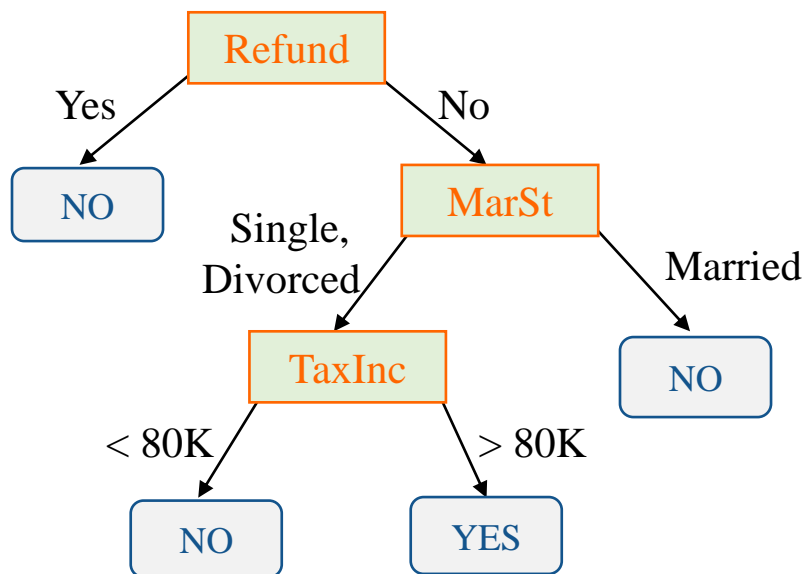
- 依赖于属性的类型

- 标称
- 序数
- 连续

- 依赖于划分的路数

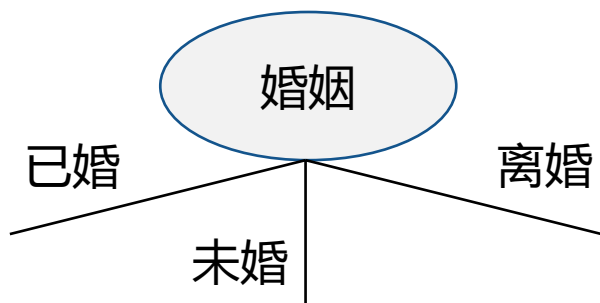
- 多路划分
- 二元划分

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



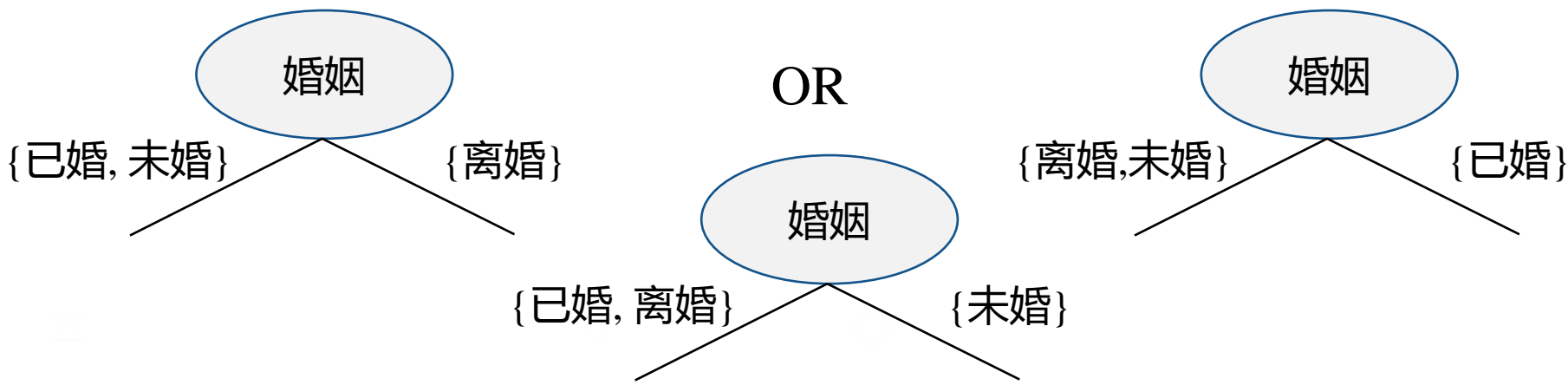
3.3.1 基于标称属性的分裂

- **多路划分：**划分数（输出数）取决于该属性不同属性值的个数



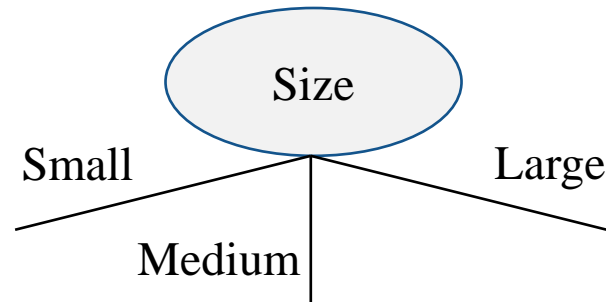
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- **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 $2^k - 1$ 种方法

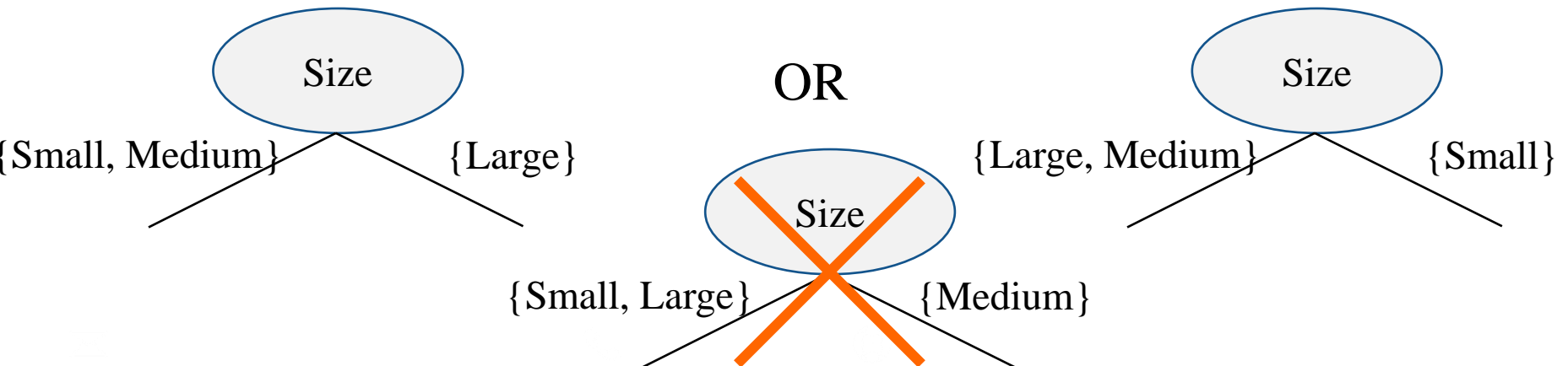


3.3.2 基于序数属性的划分

- **多路划分：**划分数（输出数）取决于该属性不同属性值的个数

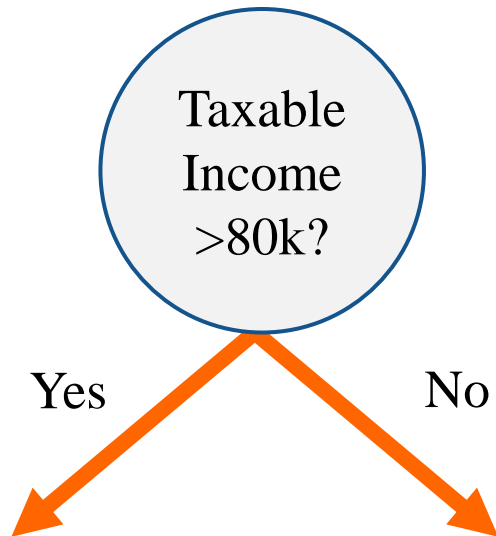


- **二元划分：**划分数为2，这种划分要考虑创建k个属性值的二元划分的所有 $2^{k-1}-1$ 种方法

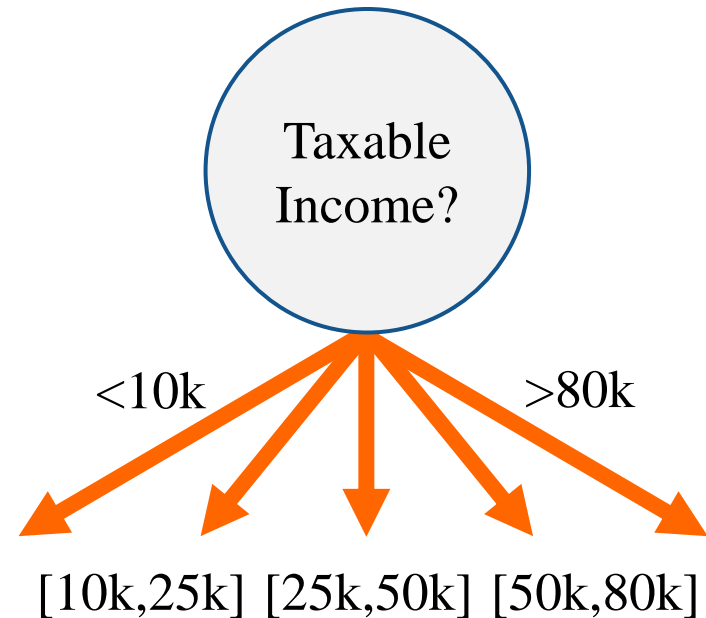


3.3.3 基于连续属性的划分

- **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:** $v_i \leq A < v_{i+1}$ ($i=1, \dots, k$)



Binary split



Multi-way split

3.4 决策树

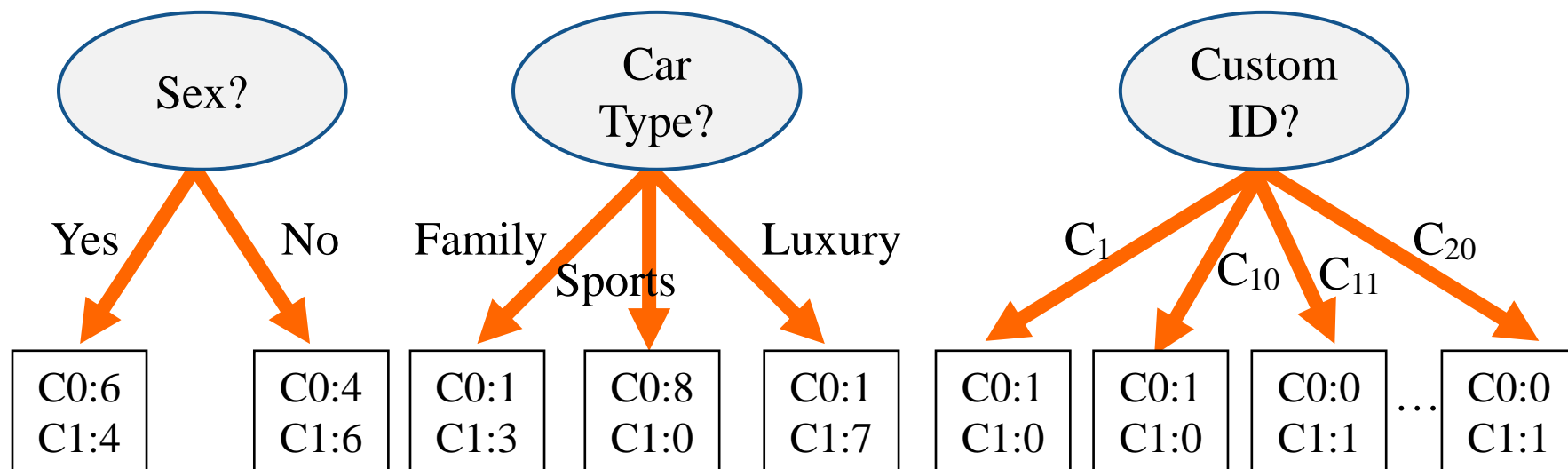
设 D_t 是与节点 t 相关联的训练记录集，算法步骤：

- 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
-
- 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

3.4 怎样选择最佳划分？

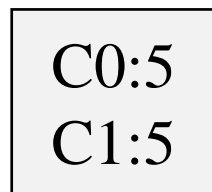
在划分前：10 个记录 class 0,

10 个记录 class 1

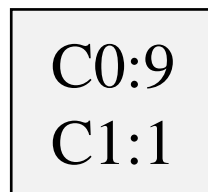


3.4 怎样选择最佳划分？

- 选择最佳划分的度量通常是根据划分后子结点纯性的程度。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：



纯性小
(不纯性大)



纯性大

3.4 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

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

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

A= [填空1] B= [填空2]

C= [填空3] D= [填空4]

c1	0
c2	6

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

$$P(C1) = 0/6 = \text{A} \quad P(C2) = 6/6 = \text{B}$$

$$Entropy = -0 \log 0 - 1 \log 1 = -0 - 0 = 0$$

c1	3
c2	3

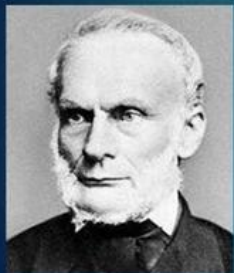
$$Entropy(t) = -\sum_j p(j|t) \log p(j|t)$$

$$P(C1) = 3/6 = \text{C} \quad P(C2) = 3/6 = \text{D}$$

$$Entropy = -0.5 \log 0.5 - 0.5 \log 0.5 = 1$$

作答

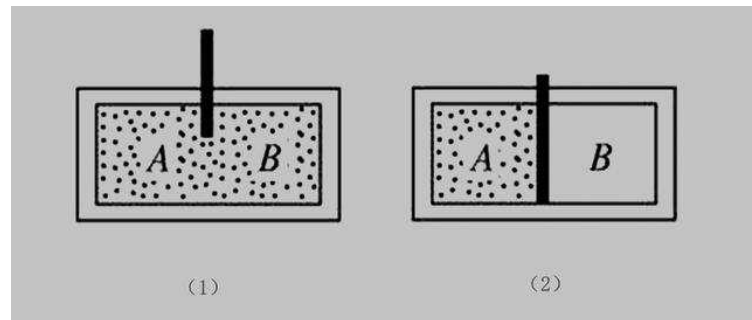
熵 (Entropy) 的科学含义



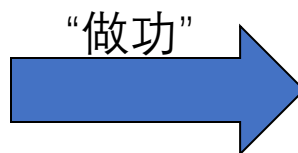
鲁道夫·克劳修斯 (Rudolf Clausius)

1850-1865年之间正式提出热力学第二定律和热力学熵的概念。

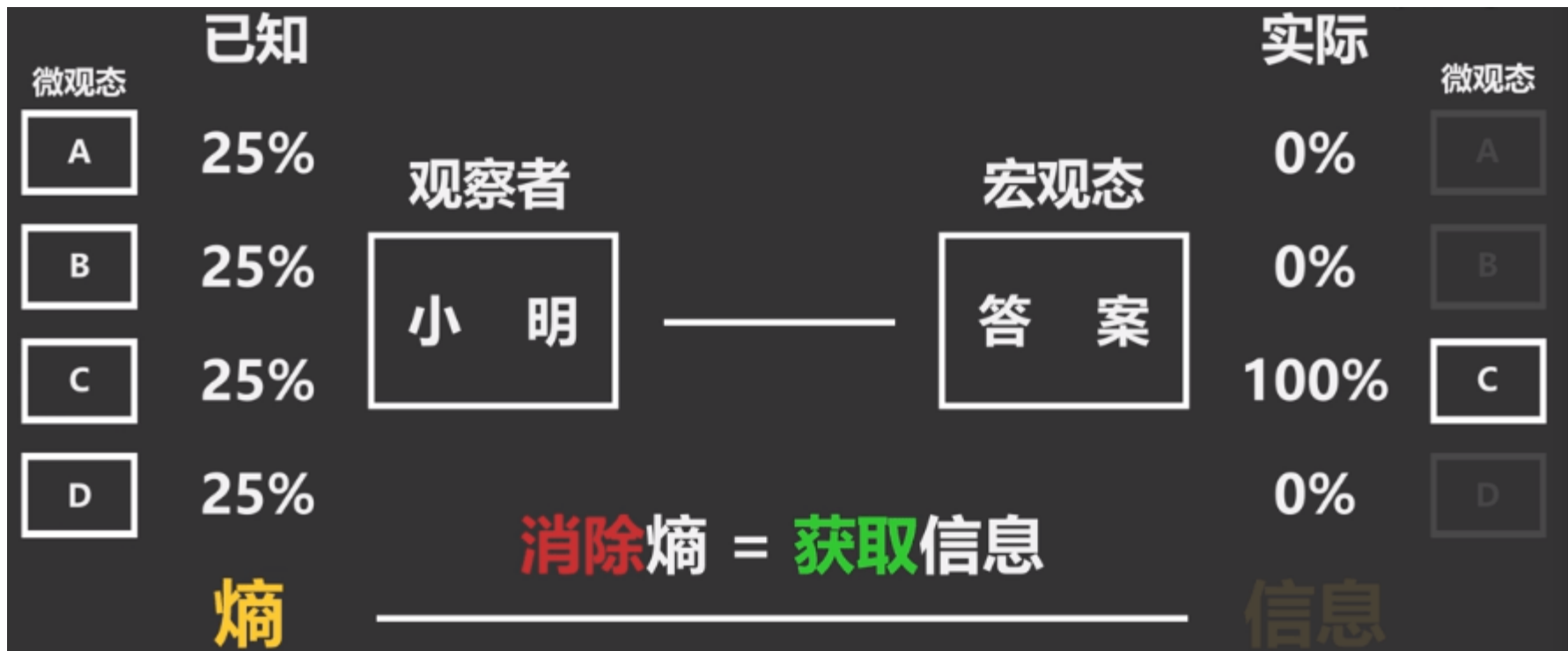
熵在经典力学上的含义更容易理解，即世界上没有永动机，最终会走向平衡静止，即熵死。



混乱 \Leftrightarrow 熵高



有序 \Leftrightarrow 熵低



信息熵

信息： $I = -\log_2 p(x_i)$

信息熵 $H = -\sum_{i=1}^n p(x_i) \log_2 p(x_i)$

信息增益: 分类前的信息减去分类后的信息熵

3.4.1 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

c1	3
c2	3

数据混乱

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

$$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$$

$$Entropy = -0 \log 0 - 1 \log 1 = -0 - 0 = 0$$

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

$$P(C1) = 3/6 = 1/2 \quad P(C2) = 3/6 = 1/2$$

$$Entropy = -0.5 \log 0.5 - 0.5 \log 0.5 = -0 - 0 = 0$$

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

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

购买的比例为: [填空1] / 14

不购买的比例为: [填空2] / 14

顾客数据的熵值: [填空3]

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

作答

3.4.1 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：

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

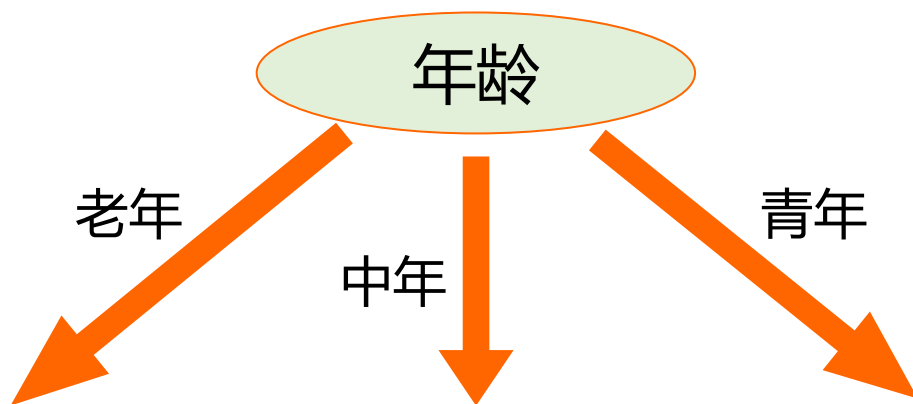
$$Gain(S, A) = Entropy(S) - \sum_{v \in A} \frac{|S_v|}{|S|} Entropy(S_v)$$

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

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

3.4.1 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

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

原始数据分类所需的期望信息：

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

按照年龄分类所需的期望信息：

$$Info_{age}(D) = \frac{5}{14} E(2,3) + \frac{4}{14} E(0,4) + \frac{5}{14} E(3,2) = \text{[填空1]}$$

作答

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

3.4.1 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：

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

按照年龄分类所需的期望信息：

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

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

3.4.1 Entropy 基于熵 —— 信息增益算法ID3

原始数据分类所需的期望信息：

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

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：

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

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

3.4.1 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

3.4.1 Entropy 基于熵 —— 信息增益算法ID3

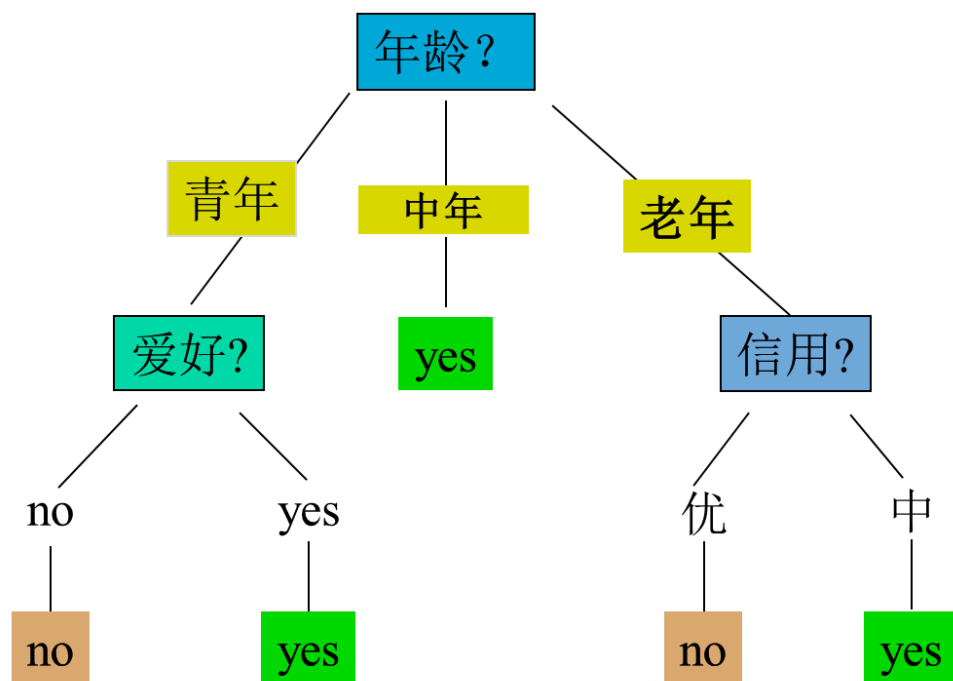
相似的

$$Gain(age) = 0.246$$

$$Gain(income) = 0.029$$

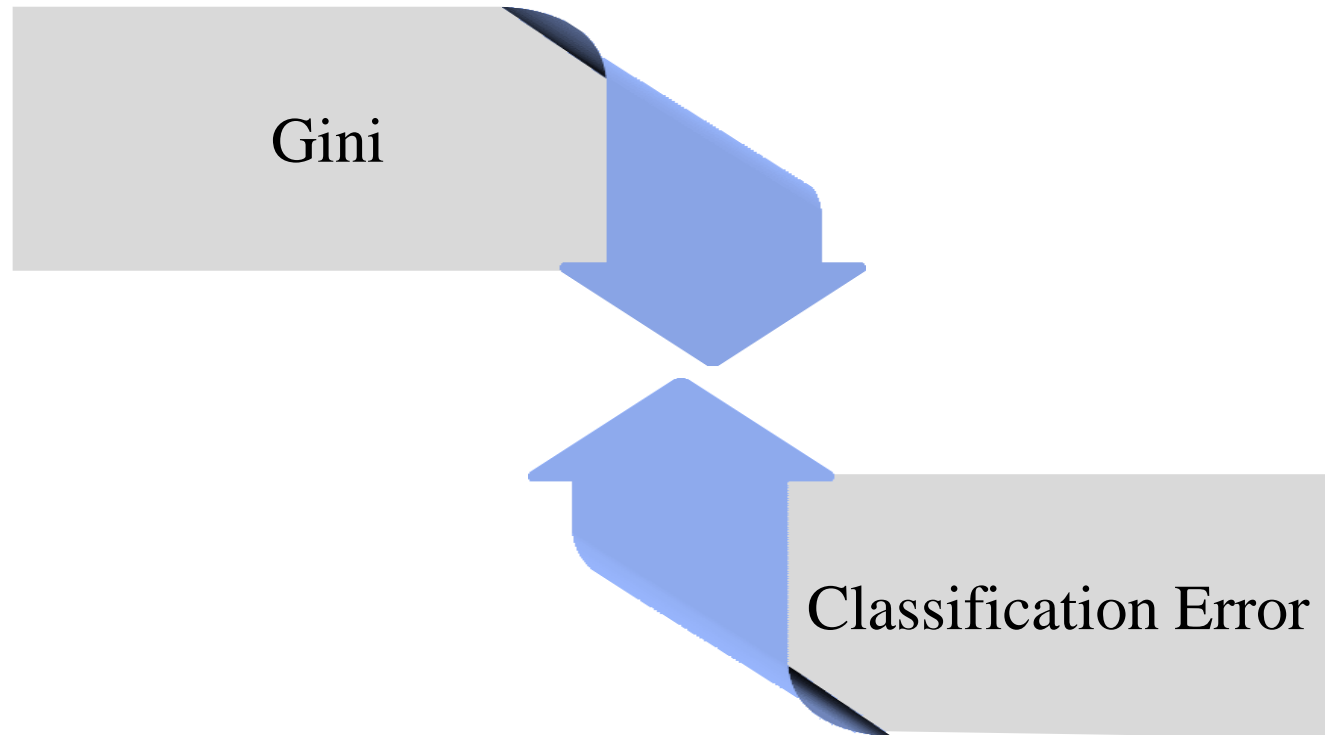
$$Gain(fancy) = 0.151$$

$$Gain(credit_rating) = 0.048$$



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

3.4.2 其它结点纯性的测量



正常使用填空题需3.0以上版本雨课堂

- 给定结点t的Gini值计算：

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

($p(j|t)$) 是在结点t中，类j发生的概率)

A= [填空1]

B= [填空2]

C= [填空3]

D= [填空4]

- 当类分布均衡时，Gini值达到最大值 ($1 - 1/n_c$)
- 相反当只有一个类时，Gini值达到最小值0，纯度越大

c1	0
c2	6

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

$$P(C1) = 0/6 = \text{A} \quad P(C2) = 6/6 = \text{B}$$

$$Gini = 1 - P(C1)^2 - P(C2)^2 = 1 - 0 - 1 = 0$$

c1	3
c2	3

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

$$P(C1) = 3/6 = \text{C} \quad P(C2) = 3/6 = \text{D}$$

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

作答

3.4.2 纯性的测量：GINI

- 给定结点t的Gini值计算：

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

($p(j|t)$ 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

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

$$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$$

$$Gini = 1 - P(C1)^2 - P(C2)^2 = 1 - 0 - 1 = 0$$

c1	3
c2	3

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

$$P(C1) = 3/6 = 1/2 \quad P(C2) = 3/6 = 1/2$$

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

- 给定结点t的Classification Error值计算：

A= [填空1]

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

B= [填空2]

C= [填空3]

- 当类分布均衡时，Error值达到最大值 $(1 - 1/n_c)$

D= [填空4]

- 相反当只有一个类时，Error值达到最小值0，纯度越大

c1	0
c2	6

$$P(C1) = 0/6 = \text{A} \quad P(C2) = 6/6 = \text{B}$$

$$Error = 1 - \max(0, 1) = 1 - 1 = 0$$

c1	1
c2	5

$$P(C1) = \text{C} \quad P(C2) = \text{D}$$

$$Error = 1 - \max(1/6, 5/6) = 1 - 5/6 = 1/6$$

作答

3.4.2 基于 Classification Error的划分

- 给定结点t的Classification Error值计算：

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

- 当类分布均衡时，Error值达到最大值 $(1 - 1/n_c)$
- 相反当只有一个类时，Error值达到最小值0，纯度越大

c1	0
c2	6

$$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$$

$$Error = 1 - \max(0, 1) = 1 - 1 = 0$$

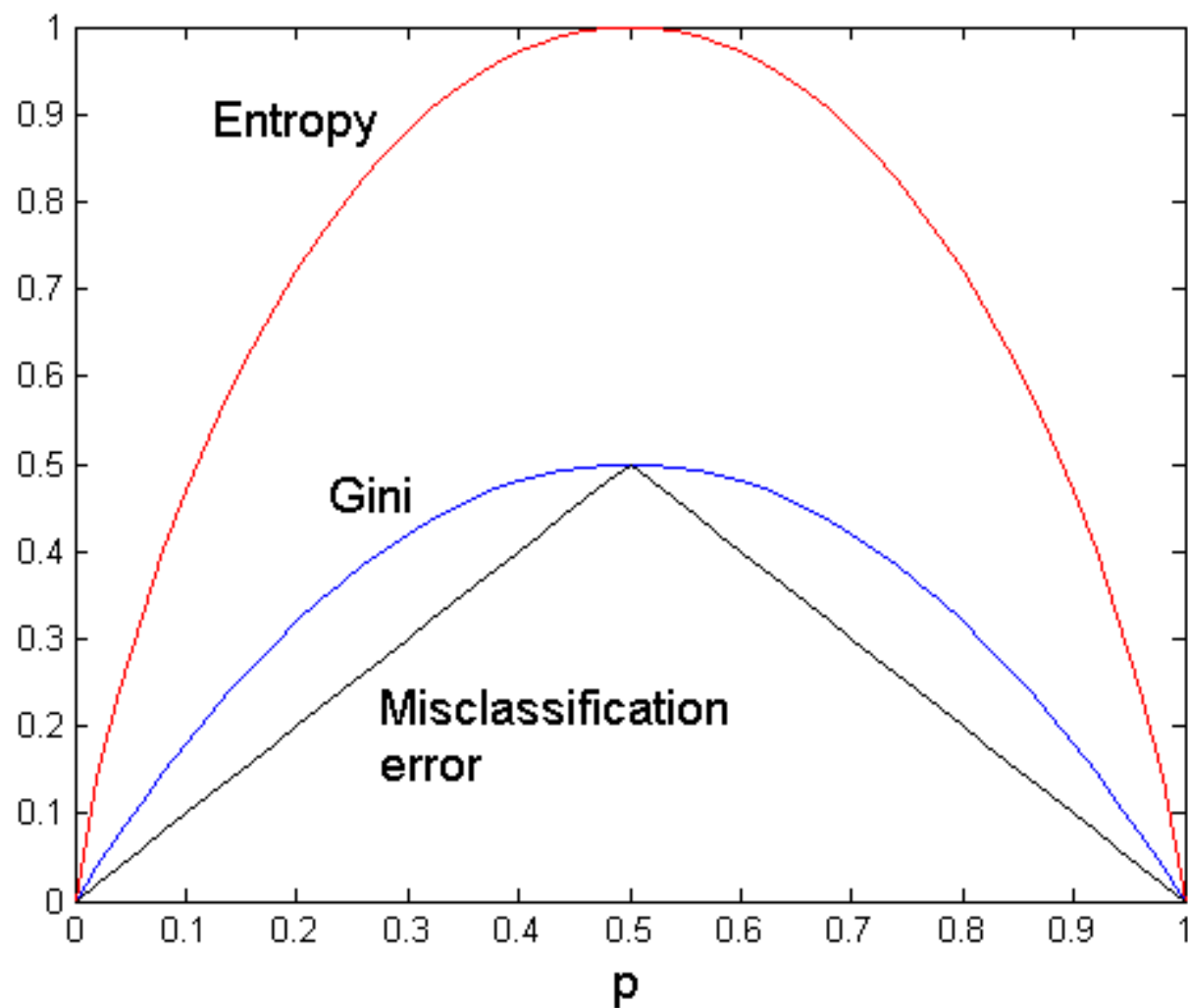
c1	1
c2	5

$$P(C1) = 1/6 \quad P(C2) = 5/6$$

$$Error = 1 - \max(1/6, 5/6) = 1 - 5/6 = 1/6$$

3.4.2 纯度度量之间的比较

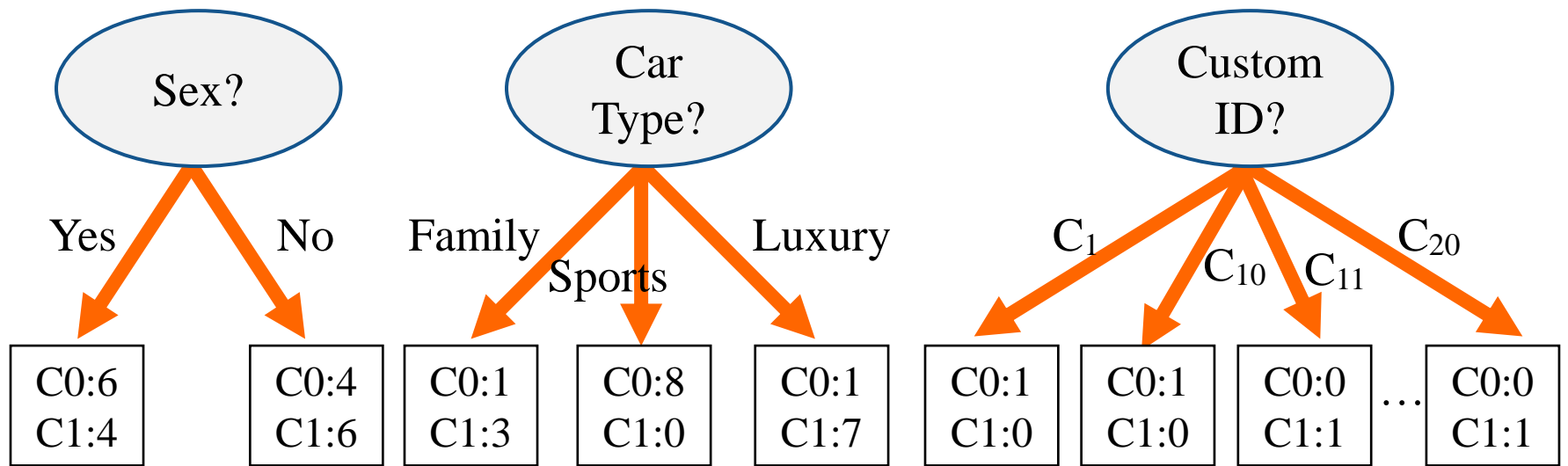
- 二元分类问题:



3.4.3 思考，哪棵树子节点纯度最高？

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如：
利用雇员id产生更纯的划分，但它却毫无用处。

3.4.3 考虑增益率 (Gain Ratio) C4.5算法

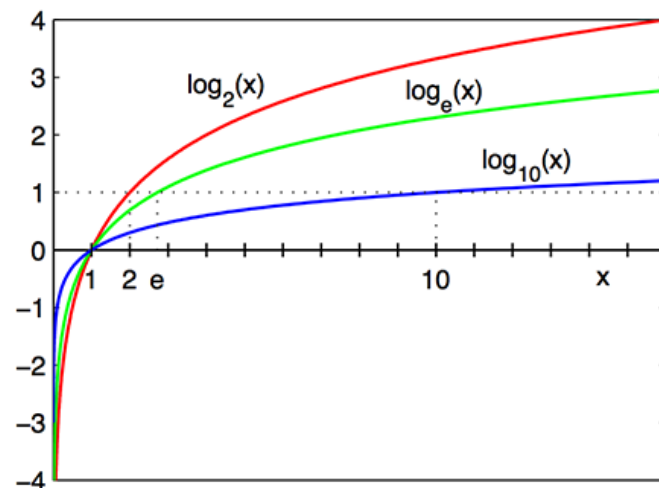
解决该问题的策略有两种：

- 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

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

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



$$Gain(age) = 0.246$$

$$Gain(income) = 0.029$$

$$Gain(fancy) = 0.151$$

$$Gain(credit_rating) = 0.048$$

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

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

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

$$SplitInfo_{income}(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right) = \text{[填空1]}$$

3.4.3 考虑增益率 (Gain Ratio) C4.5算法

$$Gain(age) = 0.246$$

$$Gain(income) = 0.029$$

$$Gain(fancy) = 0.151$$

$$Gain(credit_rating) = 0.048$$

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

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

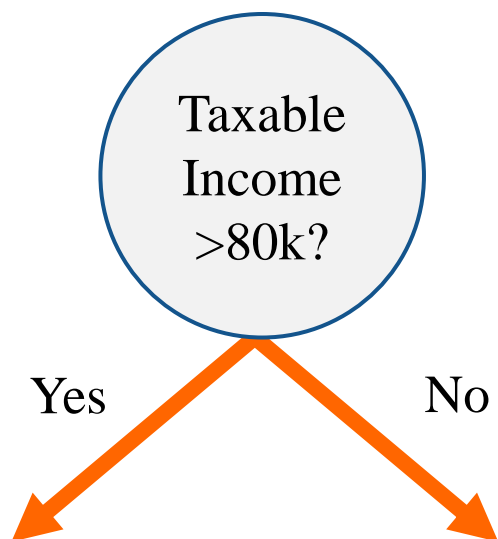
id	年龄	收入	爱好	信用	购买
1	青	高	否	中	否
2	青	高	否	优	否
3	中	高	否	中	是
4	老	中	否	中	是
5	老	低	是	中	是
6	老	低	是	优	否
7	中	低	是	优	是
8	青	中	否	中	否
9	青	低	是	中	是
10	老	中	是	中	是
11	青	中	是	优	是
12	中	中	否	优	是
13	中	高	是	中	是
14	老	中	否	优	否

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

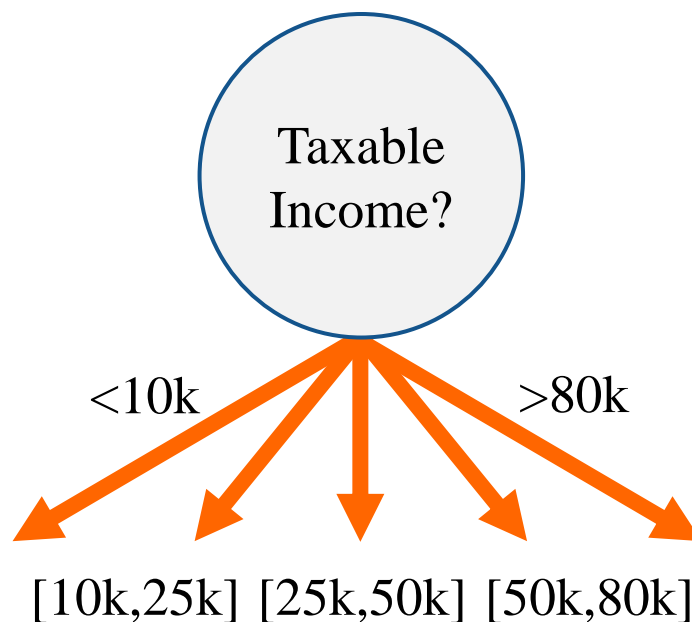
$$gain_ratio(income) = 0.029 / 1.557 = 0.019$$

3.4.4 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:** $v_i \leq A < v_{i+1}$ ($i=1, \dots, k$)



Binary split



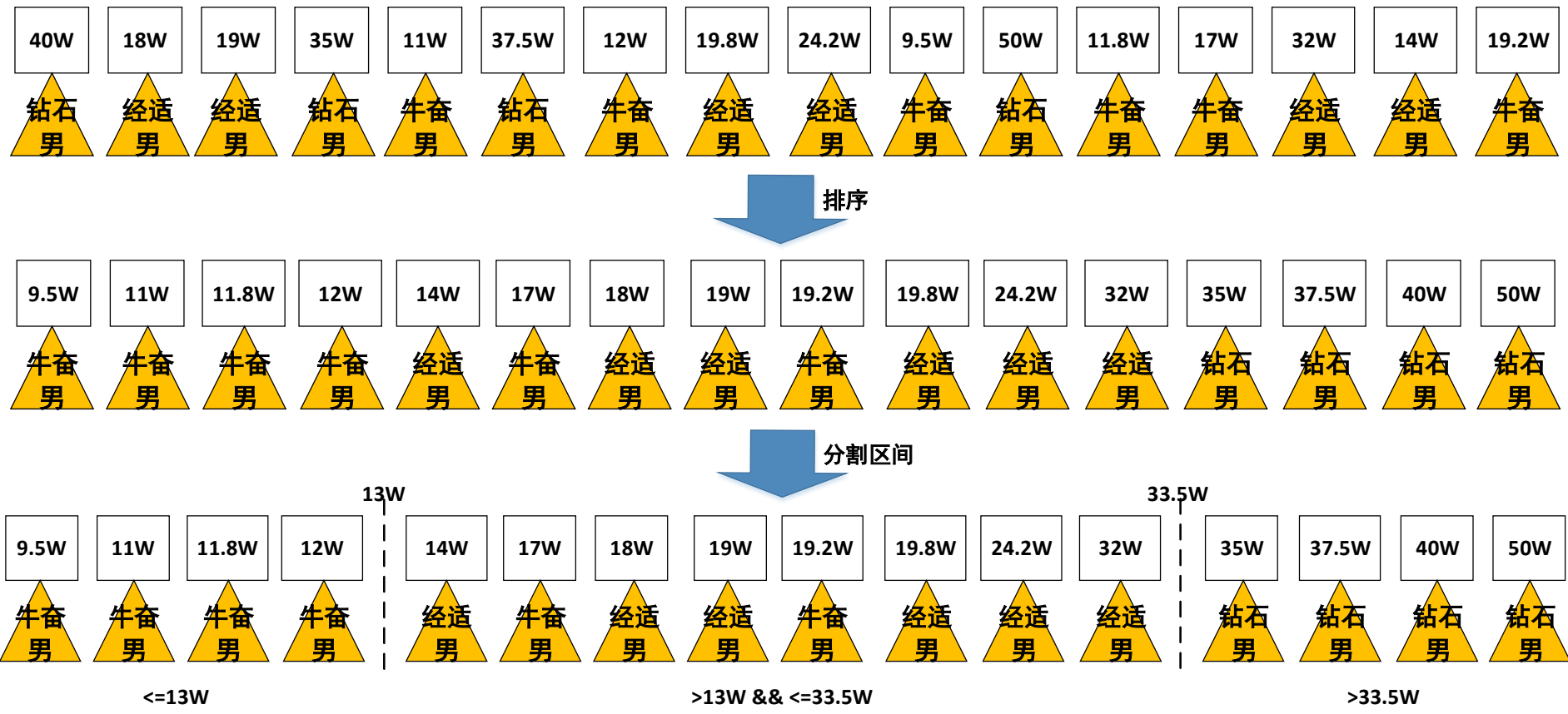
Multi-way split

3.4.4 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

3.4.4 多路划分 —— 连续变量的离散化处理

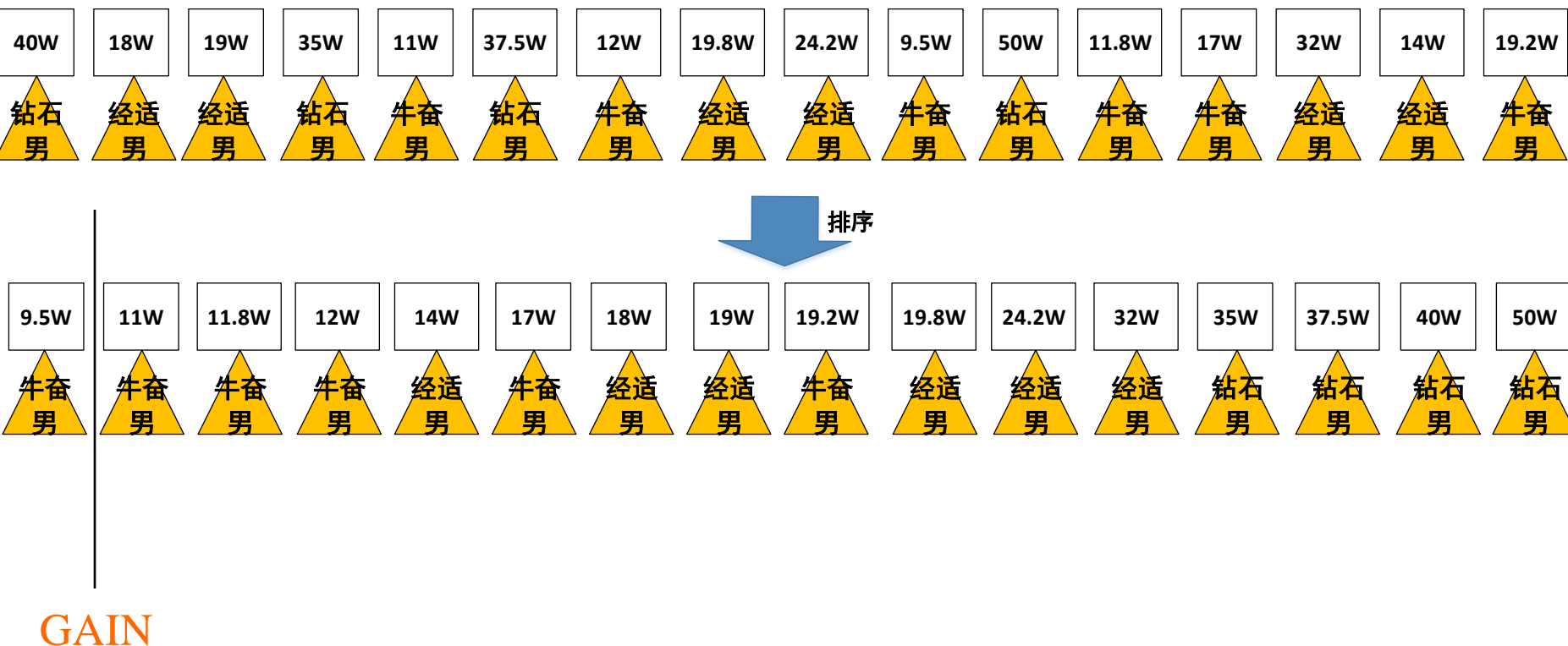
- 收入是个连续变量，分割成离散区间



3.4.4 二元划分 —— 选择最佳划分点

- 分割区间的策略

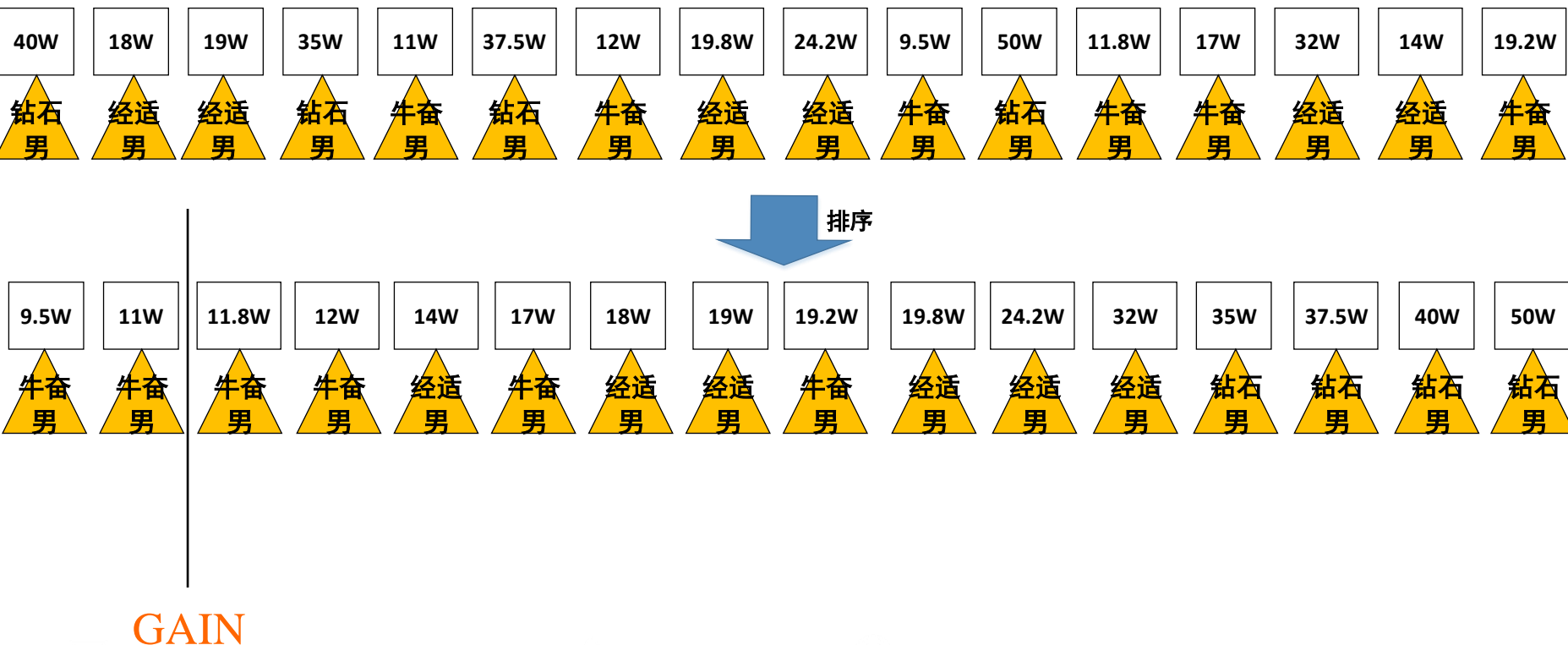
- 从最小值开始建立分割区间，开始计算各自的信息增益，选择**信息增益最大**的一个分割区间作为最佳划分点



3.4.4 二元划分 —— 选择最佳划分点

- 分割区间的策略

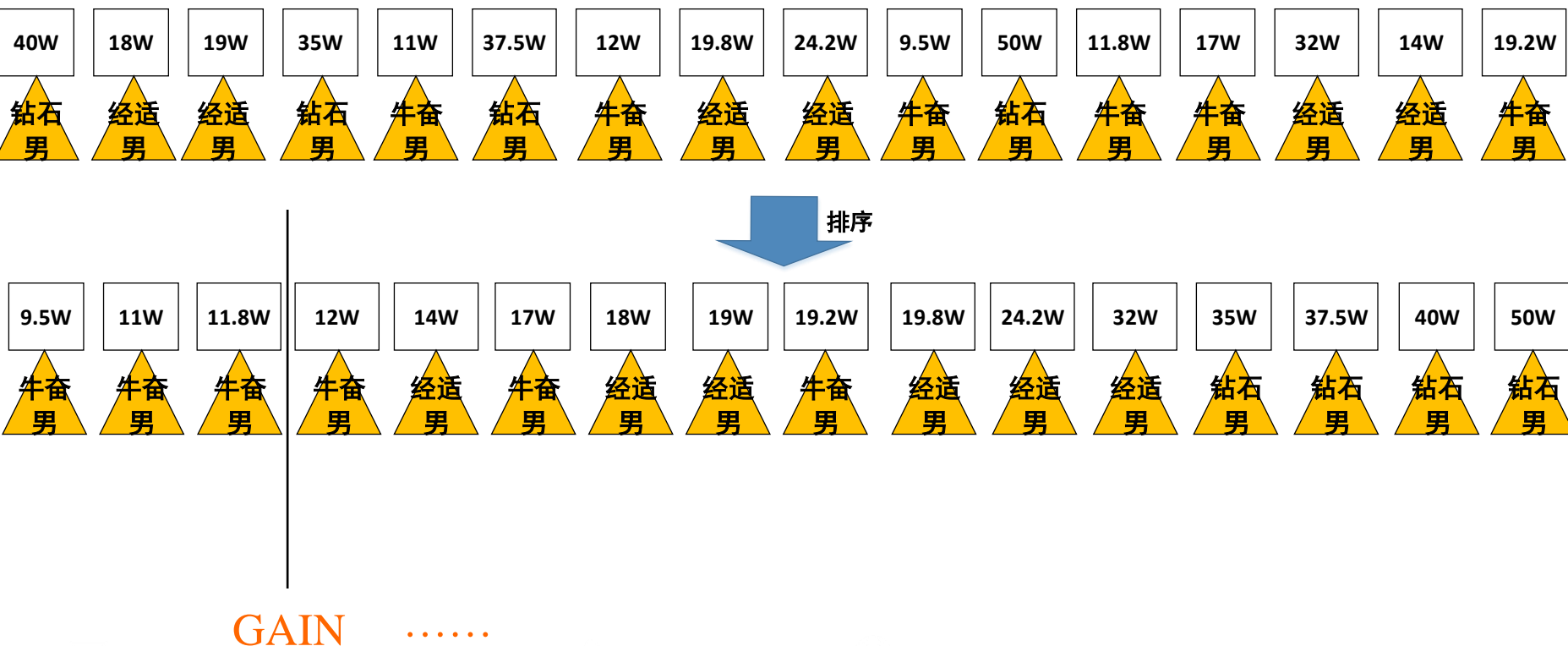
- 从最小值开始建立分割区间，开始计算各自的信息增益，选择**信息增益最大**的一个分割区间作为最佳划分点



3.4.4 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



采用决策树分类算法，连续数据如何处理？

- ☒ A 连续数据离散化
- ☒ B 选择最佳划分点分裂
- ☐ C 连续数据每2个值之间形成分裂

作答

决策树特征构造适合采用如下哪种方法

☐ A 单调变换

☒ B 线性组合

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

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

作答

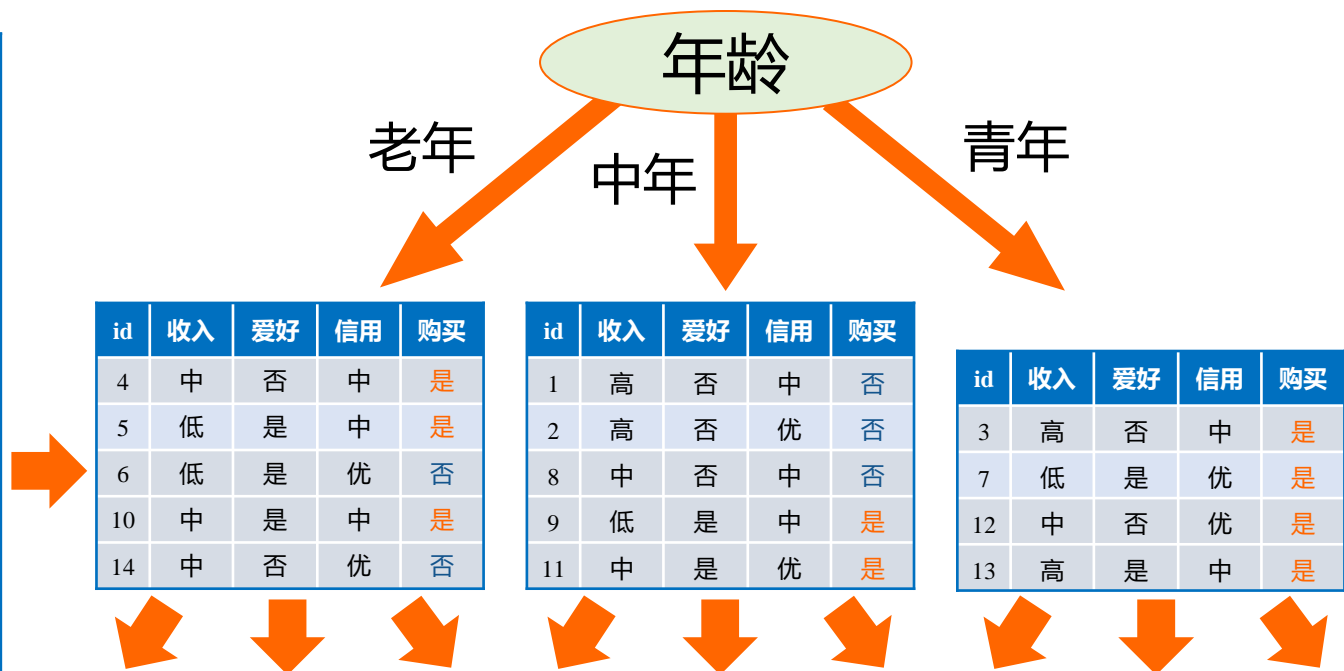
3.5 构造决策树

设 D_t 是与节点 t 相关联的训练记录集，算法步骤：

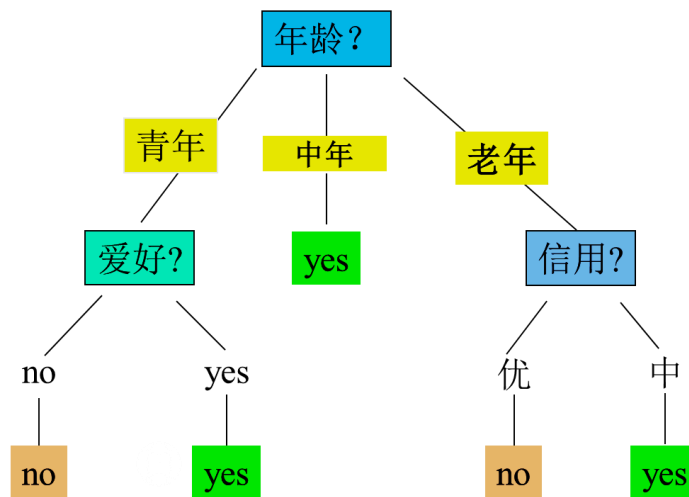
- 如果 D_t 中所有记录都属于同一个类 y_t ，则 t 是叶节点，用 y_t 标记
 - 如果 D_t 中包含属于多个类的记录，则选择一个属性测试条件，将记录划分成较小的子集
 - 对于测试条件的每个输出，创建一个子结点，并根据测试结果将 D_t 中的记录分布到子结点中。然后，对于每个子结点，递归地调用该算法
-
- Hunt算法采用贪心策略构建决策树
 - 在选择划分数据的属性时，采取一系列局部最优决策来构造决策树。
 - 决策树归纳的设计问题
 - 如何分裂训练记录
 - 怎样为不同类型的属性指定测试条件？
 - 怎样评估每种测试条件？
 - 如何停止分裂过程

3.5 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



3.5 构造决策树

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

3.5.1 构造决策树——一个例子

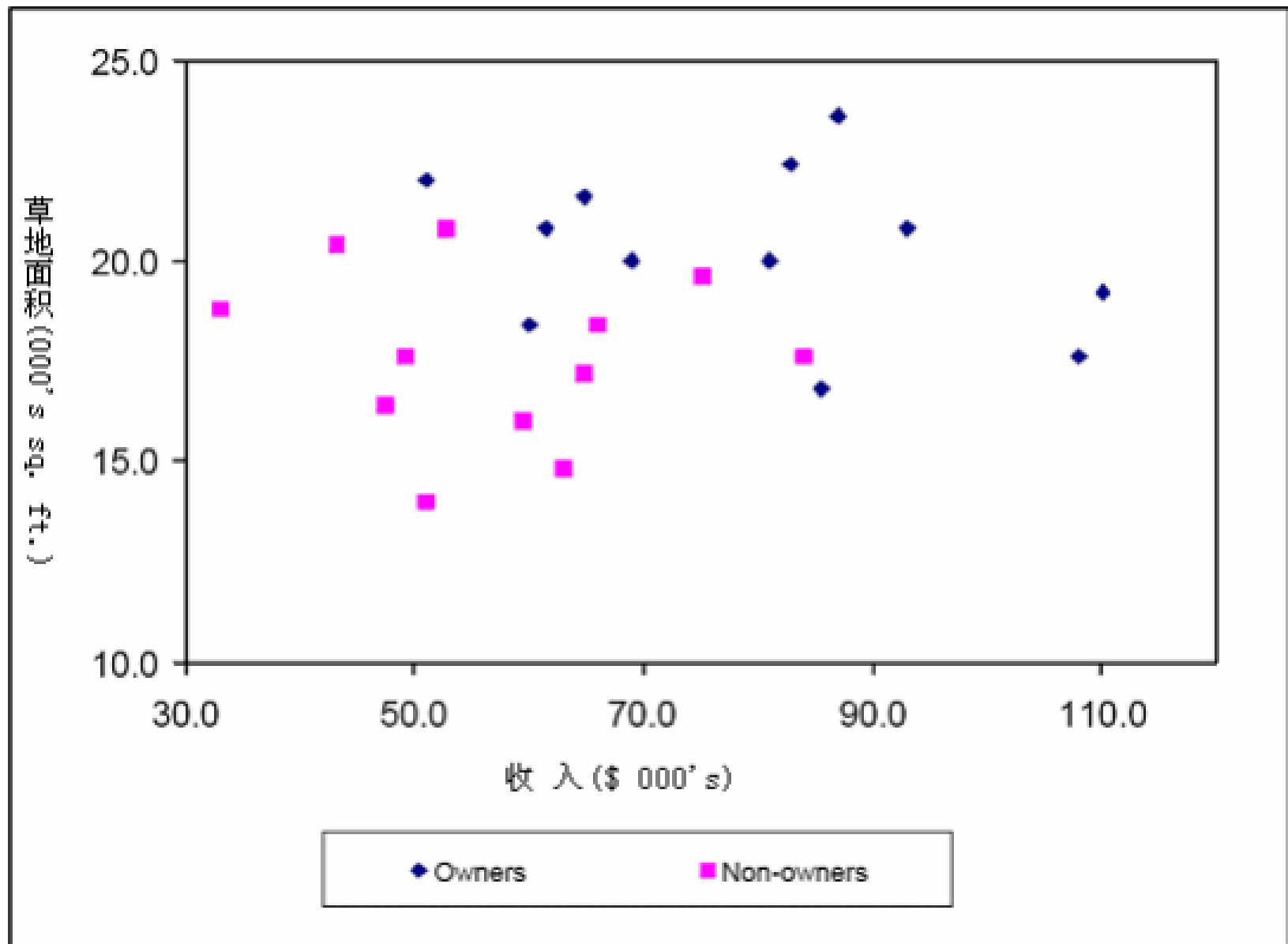
割草机制造商意欲发现一个把城市中的家庭分成那些愿意购买乘式割草机和不愿意购买的两类的方法。在这个城市的家庭中随机抽取24个非拥有者的家庭作为样本。

自变量是收入和草地面积

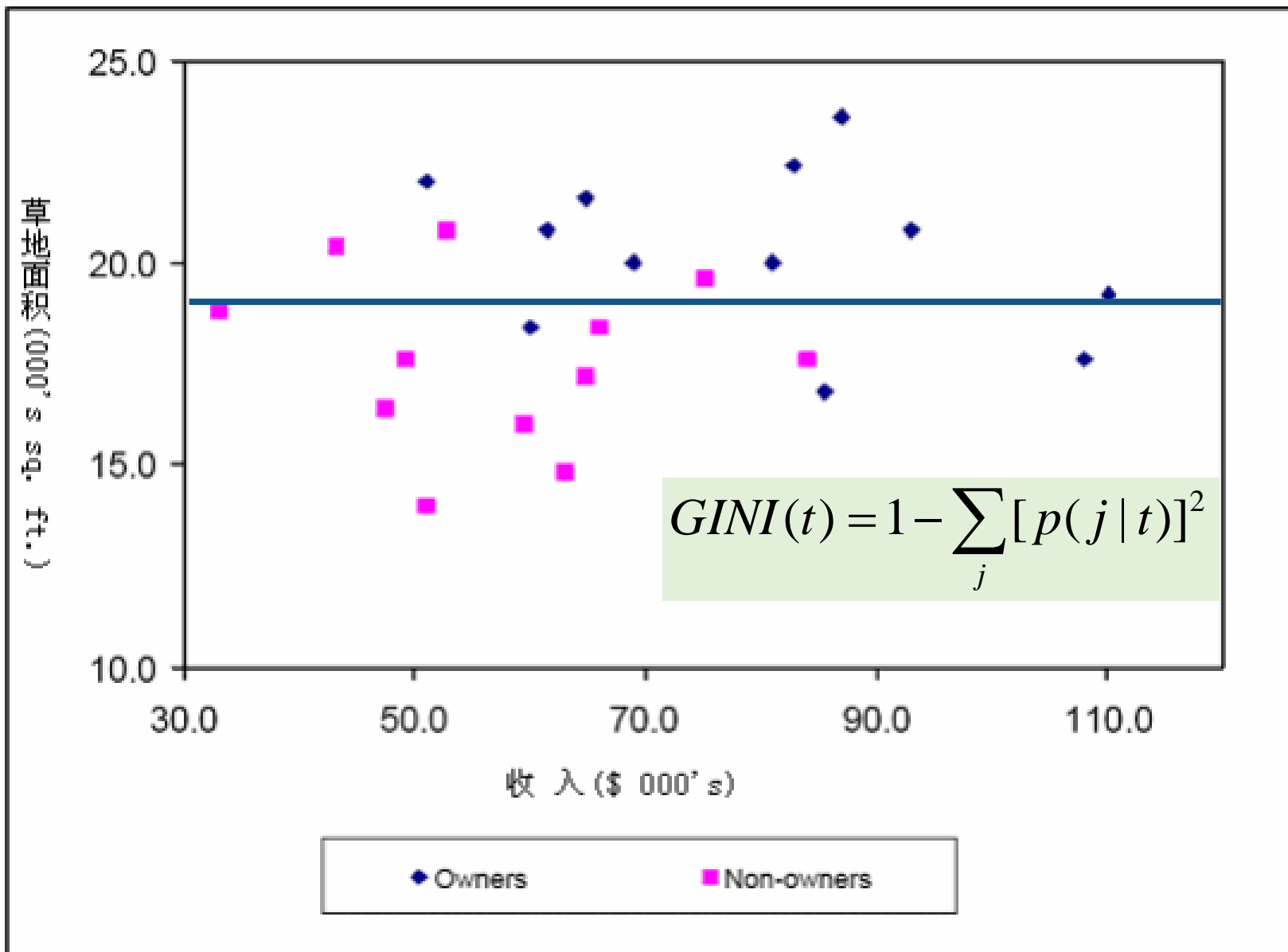
类别变量是：拥有和没有割草机。

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

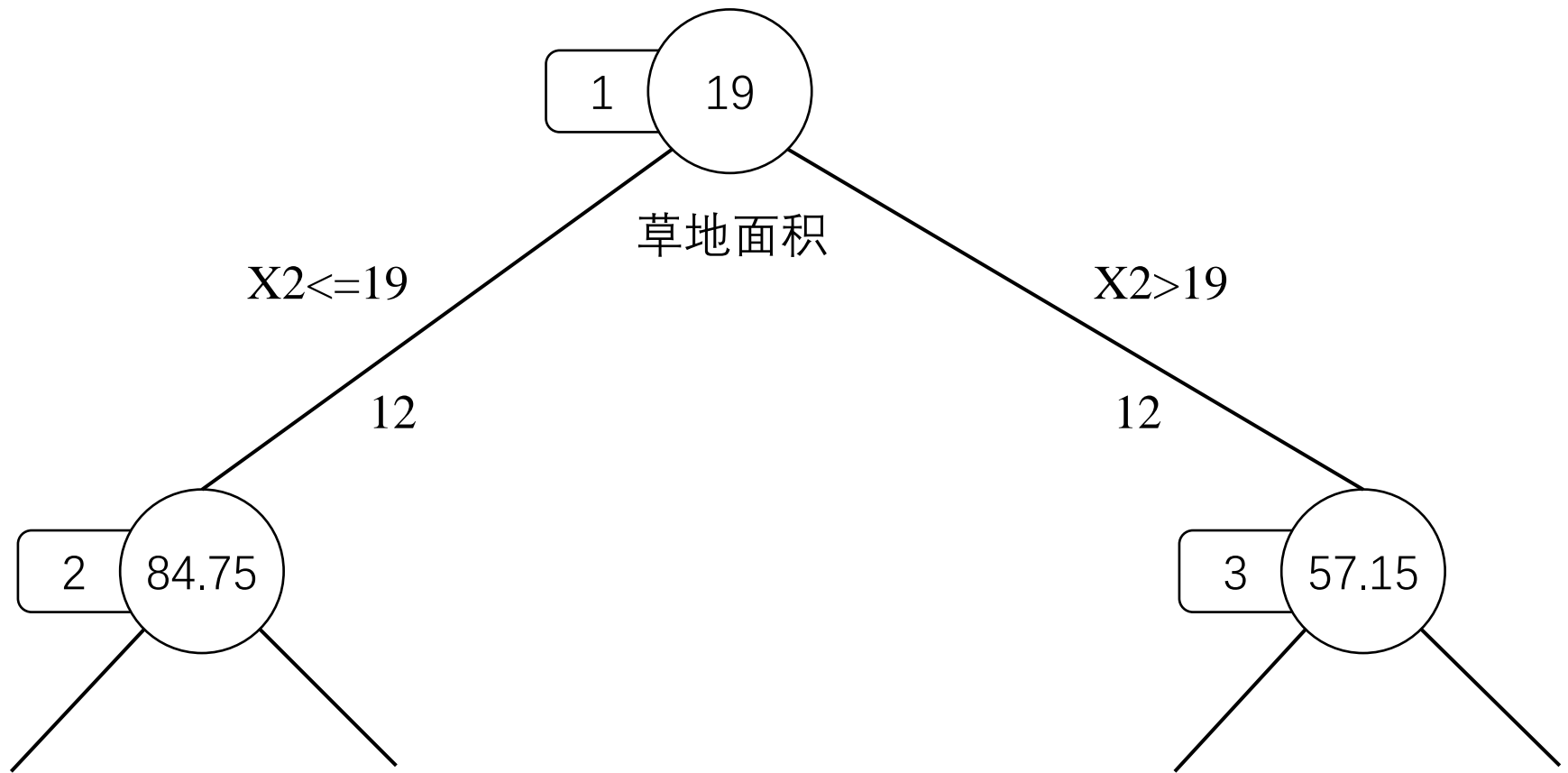
3.5.1 构造决策树 —— 一个例子



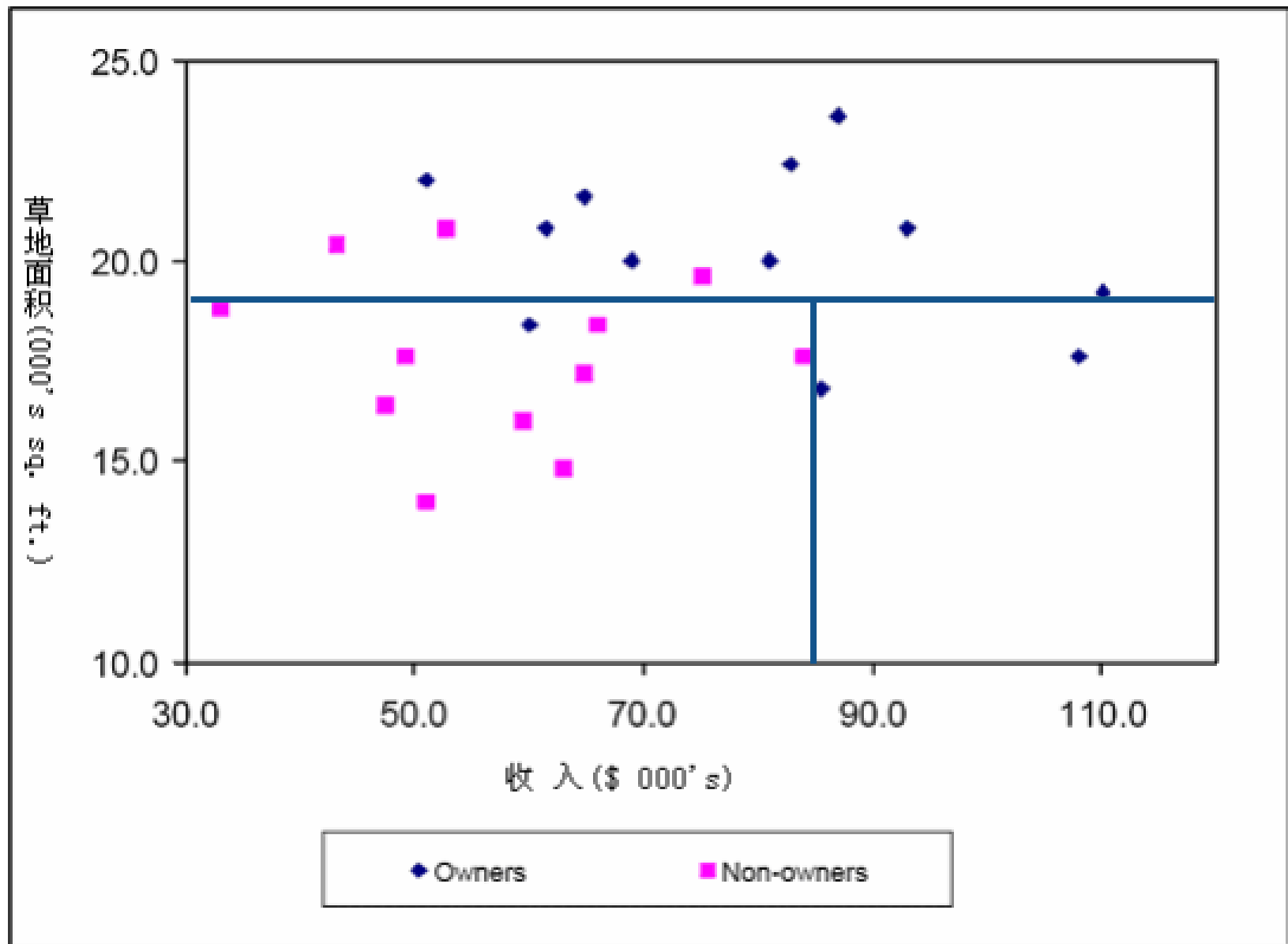
3.5.1 构造决策树——一个例子



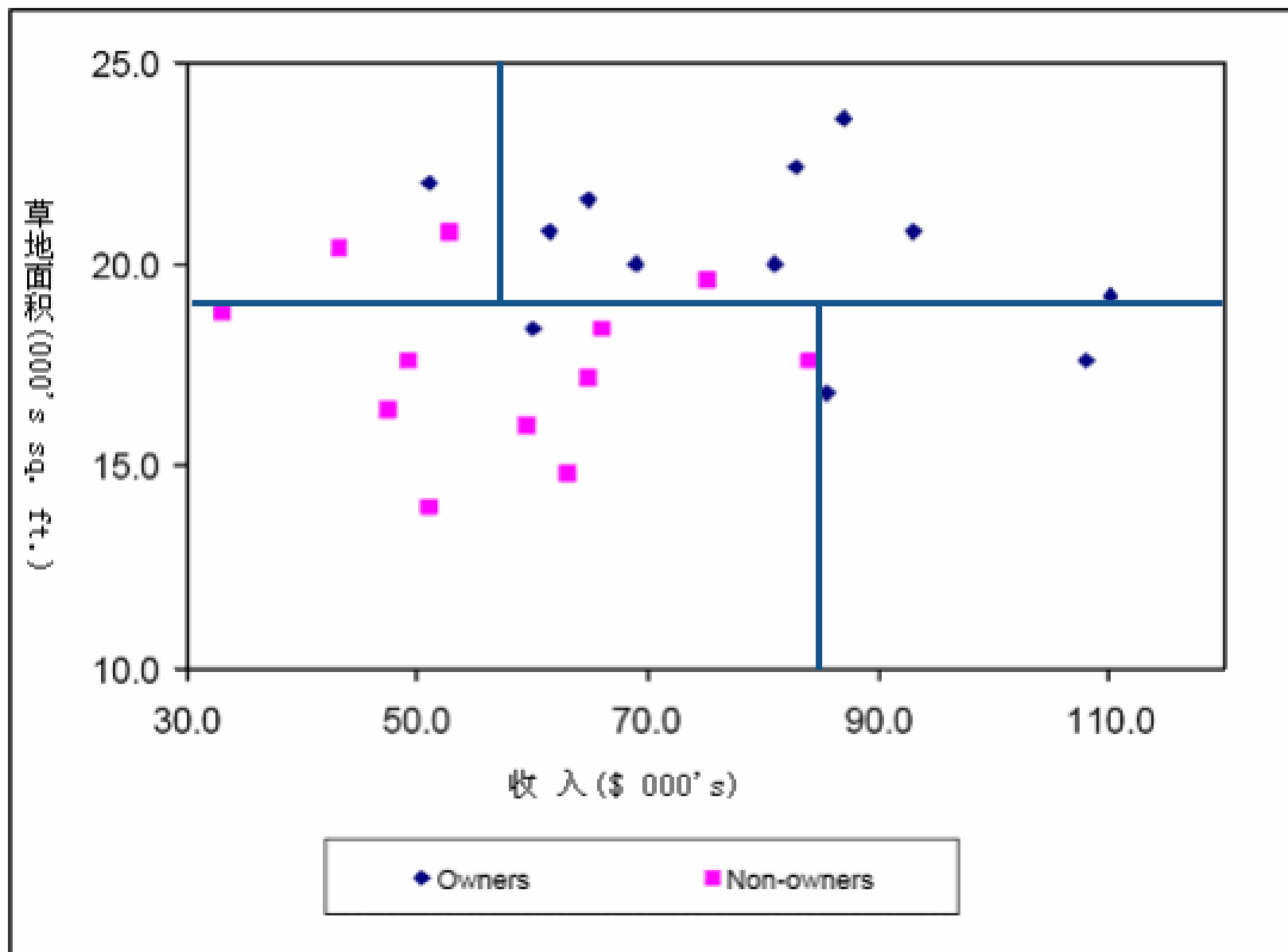
3.5.1 构造决策树——一个例子



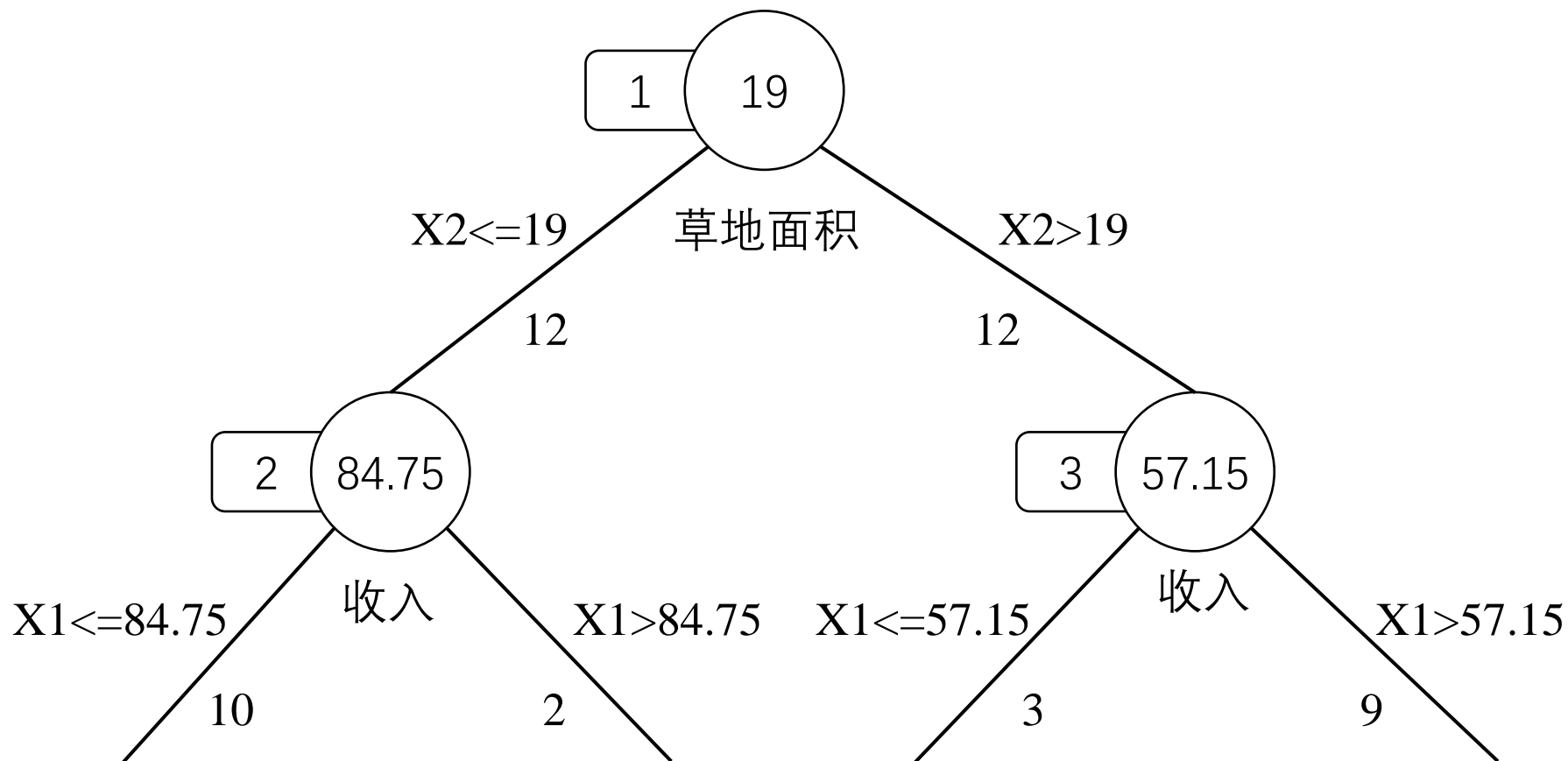
3.5.1 构造决策树 —— 一个例子



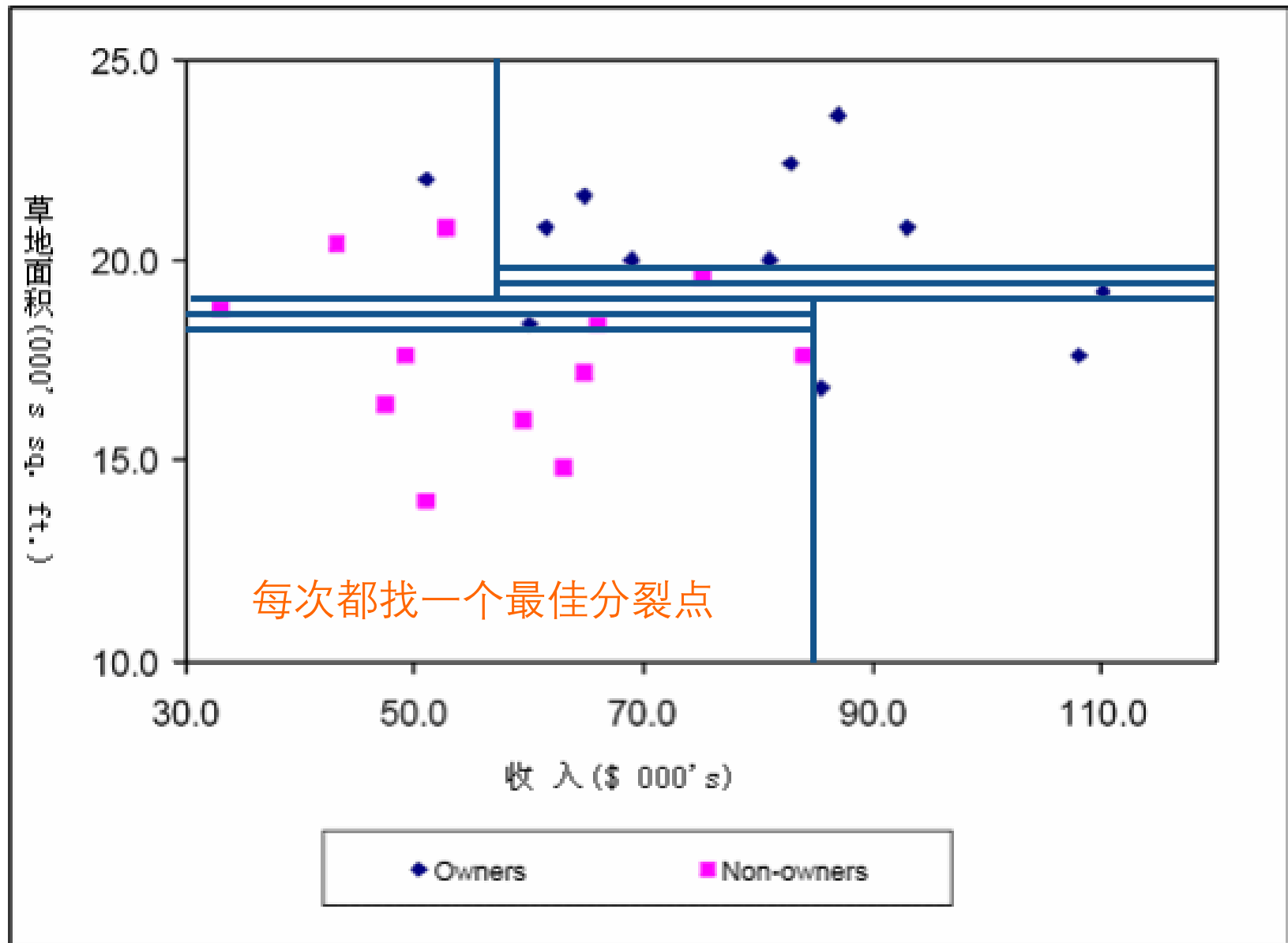
3.5.1 构造决策树——一个例子



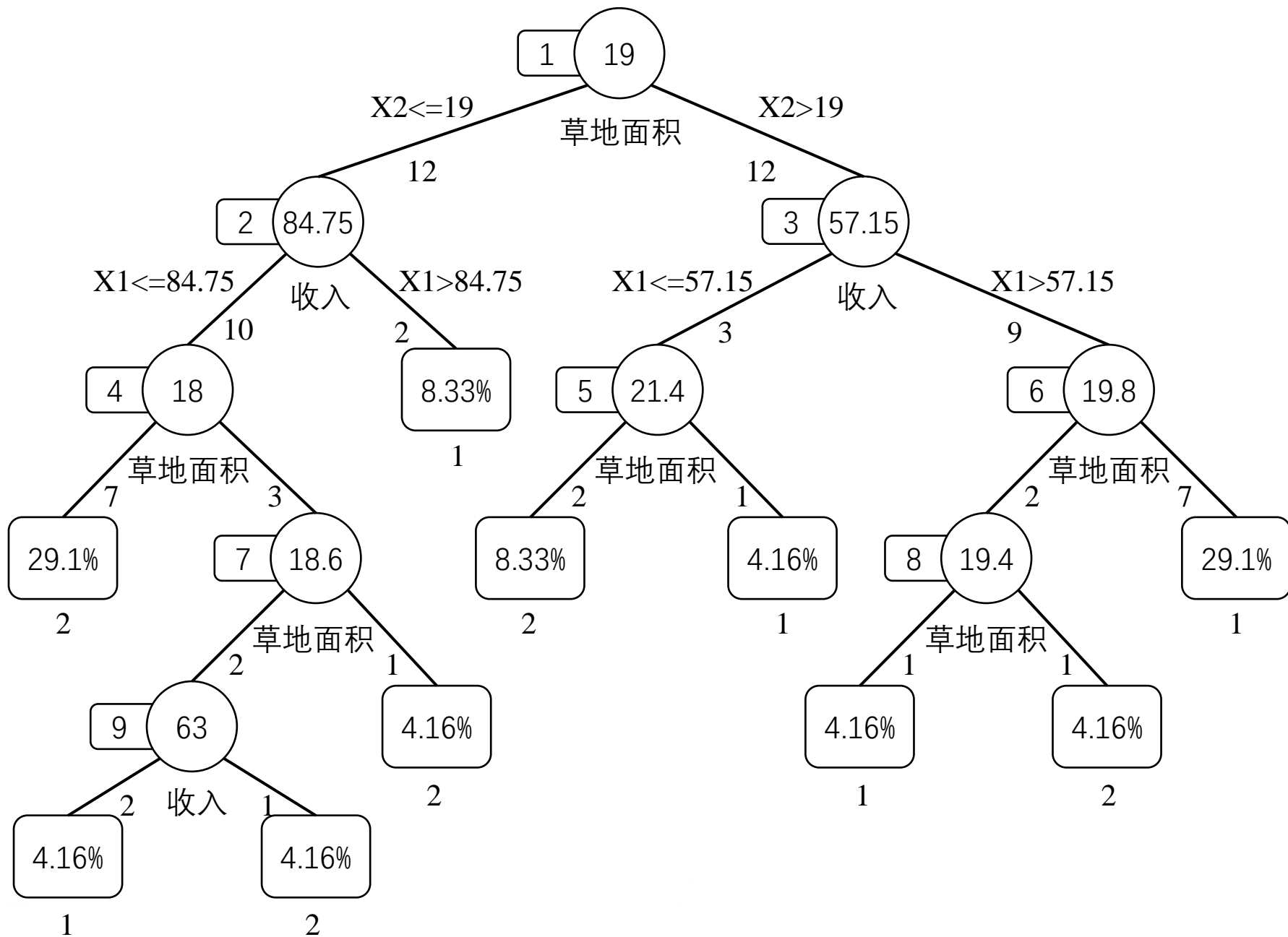
3.5.1 构造决策树 —— 一个例子



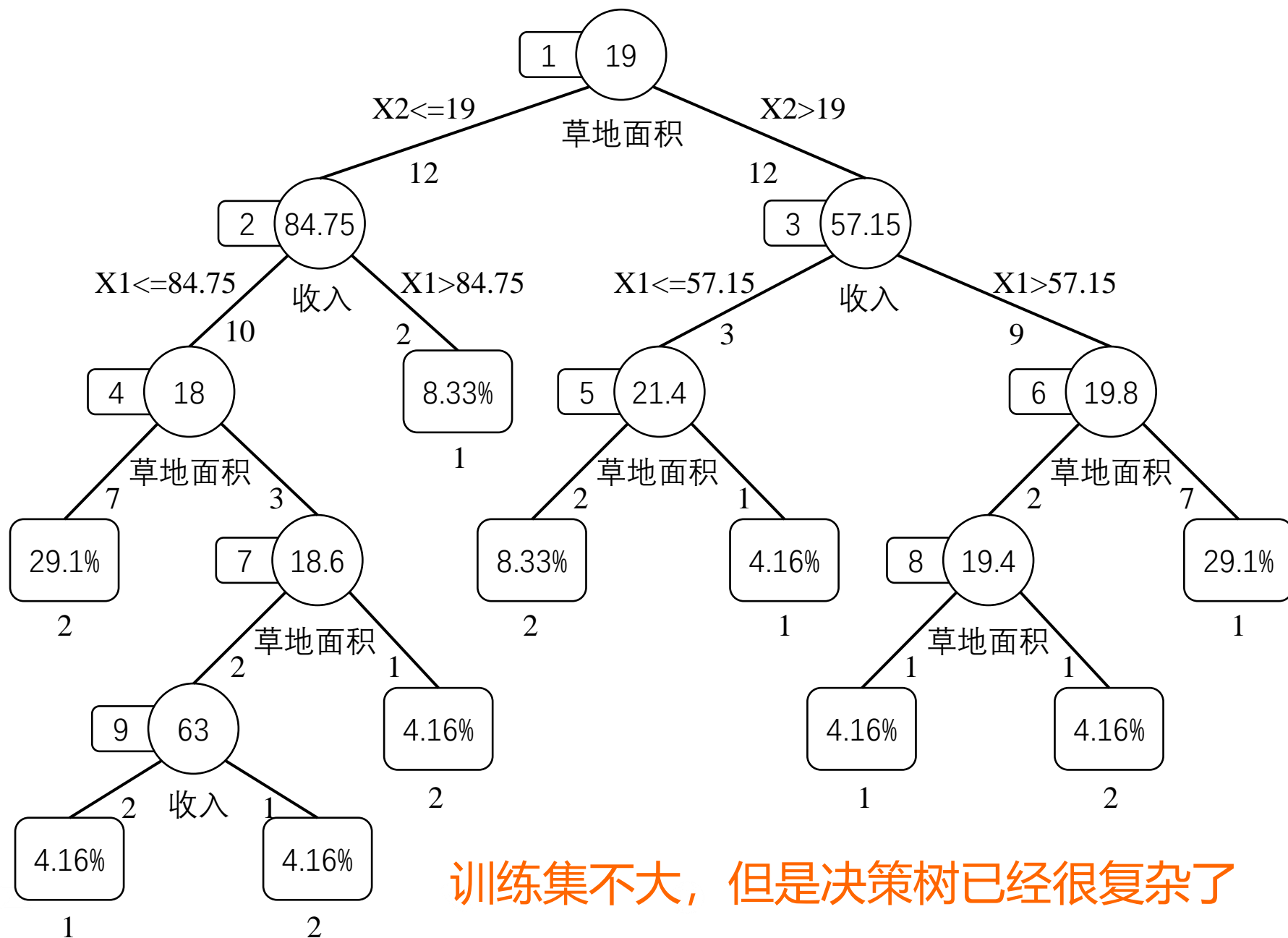
3.5.1 构造决策树 —— 一个例子



3.5.1 构造决策树——一个例子（CART算法）

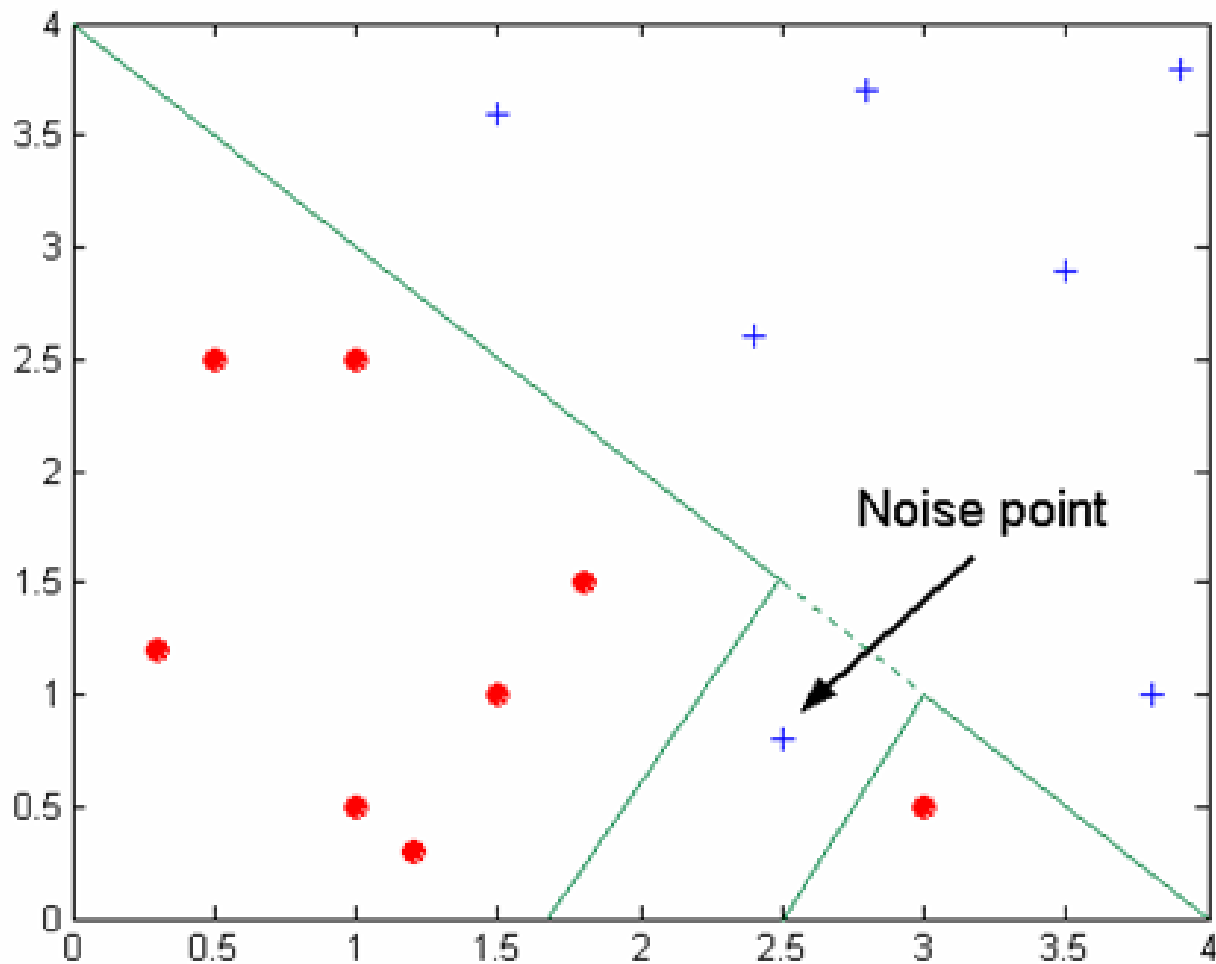


3.5.1 构造决策树——一个例子（CART算法）



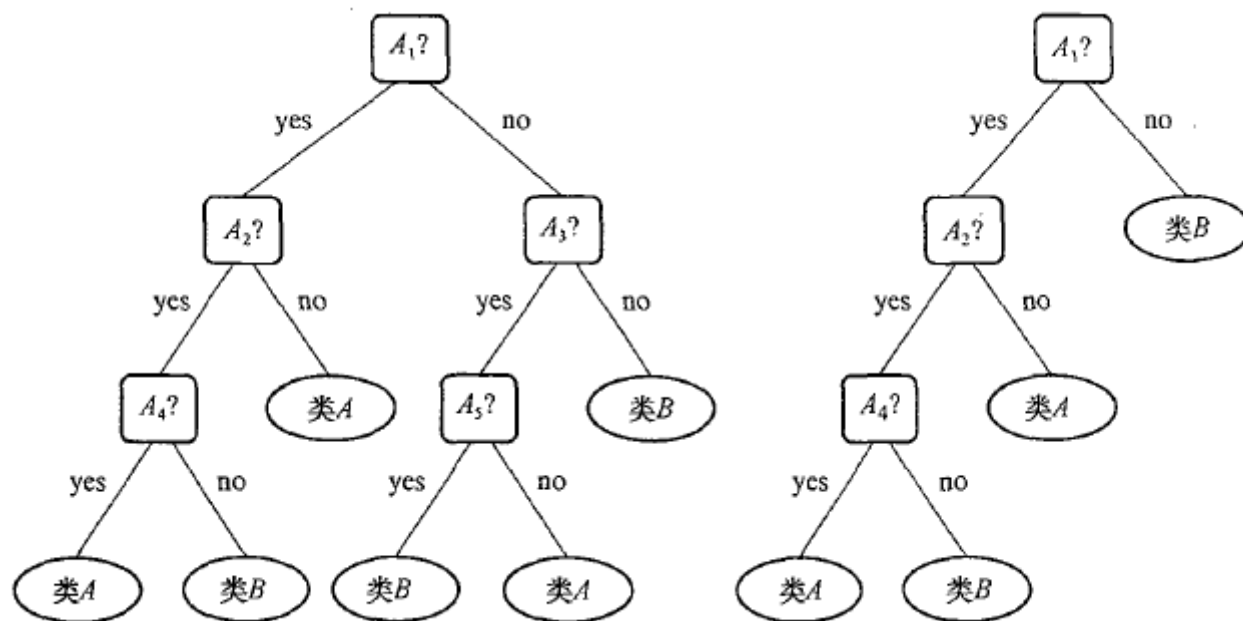
3.5.1 构造决策树 —— 复杂的决策树带来过拟合问题

- 过拟合问题



3.5.2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

下列说法正确的是

- ☐ A 过拟合是由于训练集多，模型过于简单
- ☒ B 过拟合是由于训练集少，模型过于复杂
- ☒ C 欠拟合是由于训练集多，模型过于简单
- ☐ D 欠拟合是由于训练集少，模型过于简单

提交

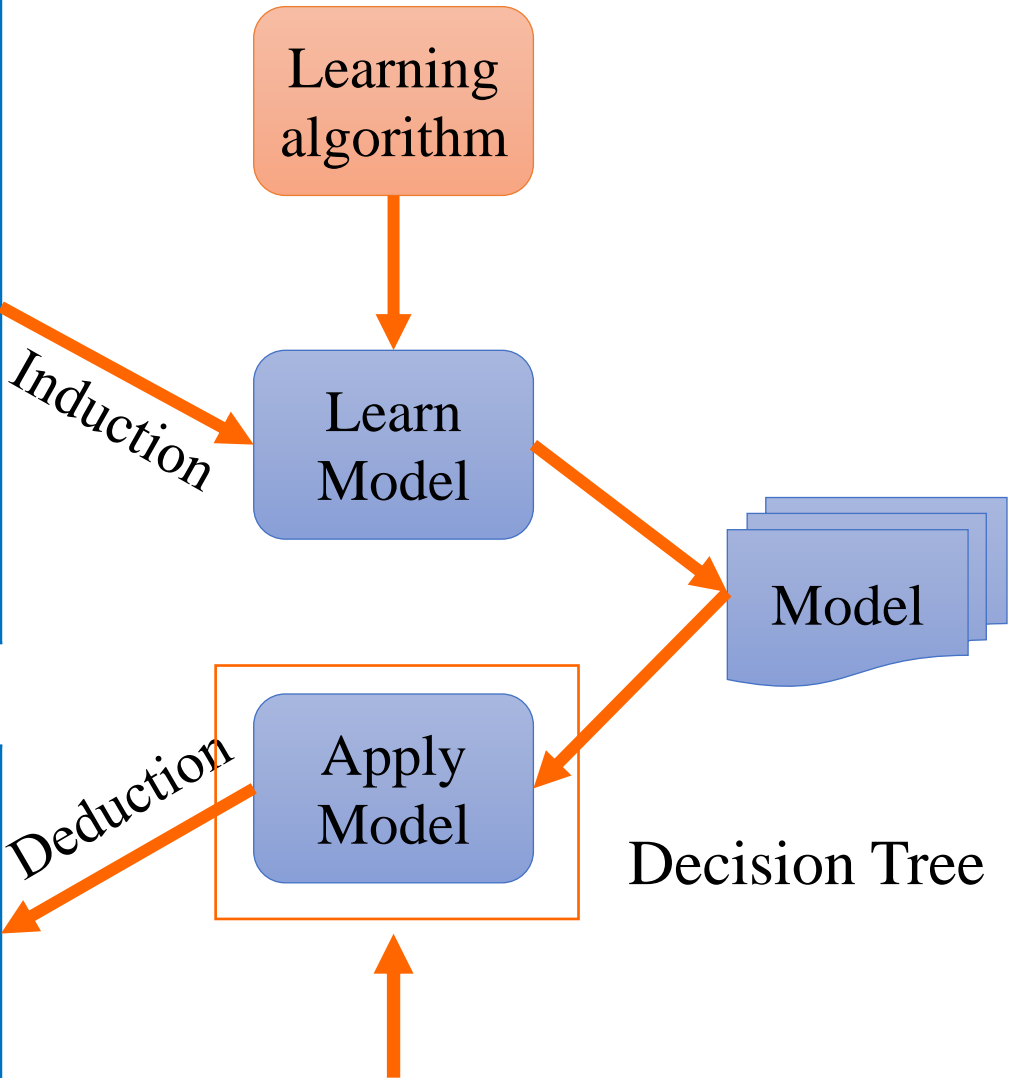
3.6 决策树分类任务

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	?

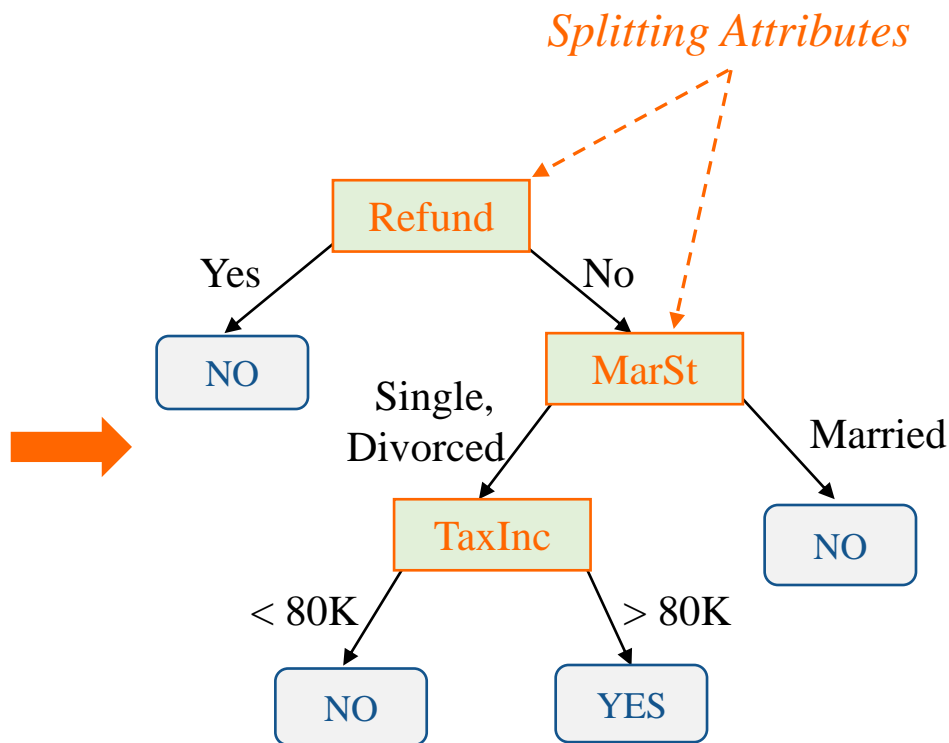
Test Set



3.6 决策树分类任务

categorical		categorical		continuous	class
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	

训练数据



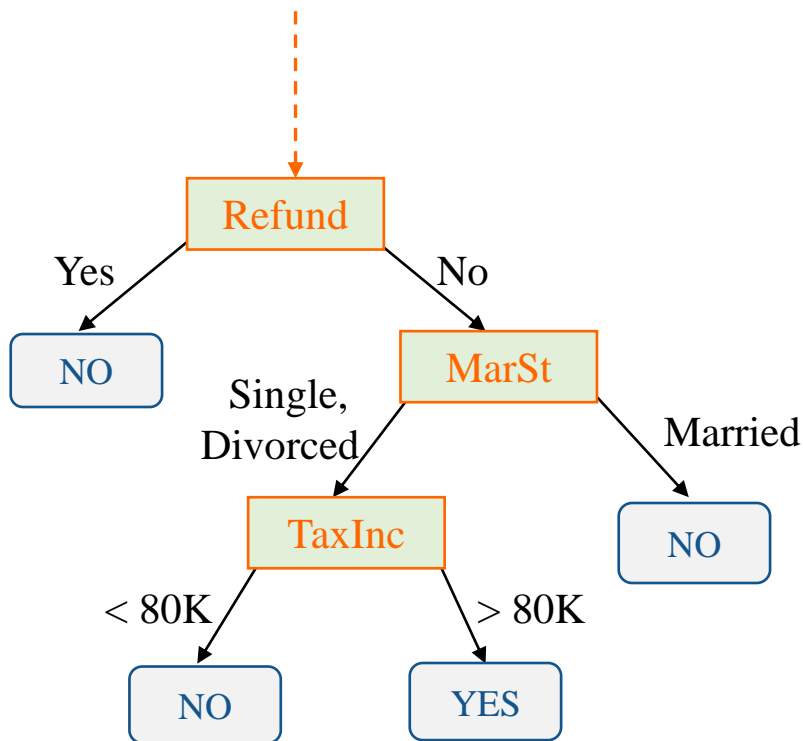
模型：决策树

3.6 决策树分类任务

测试数据

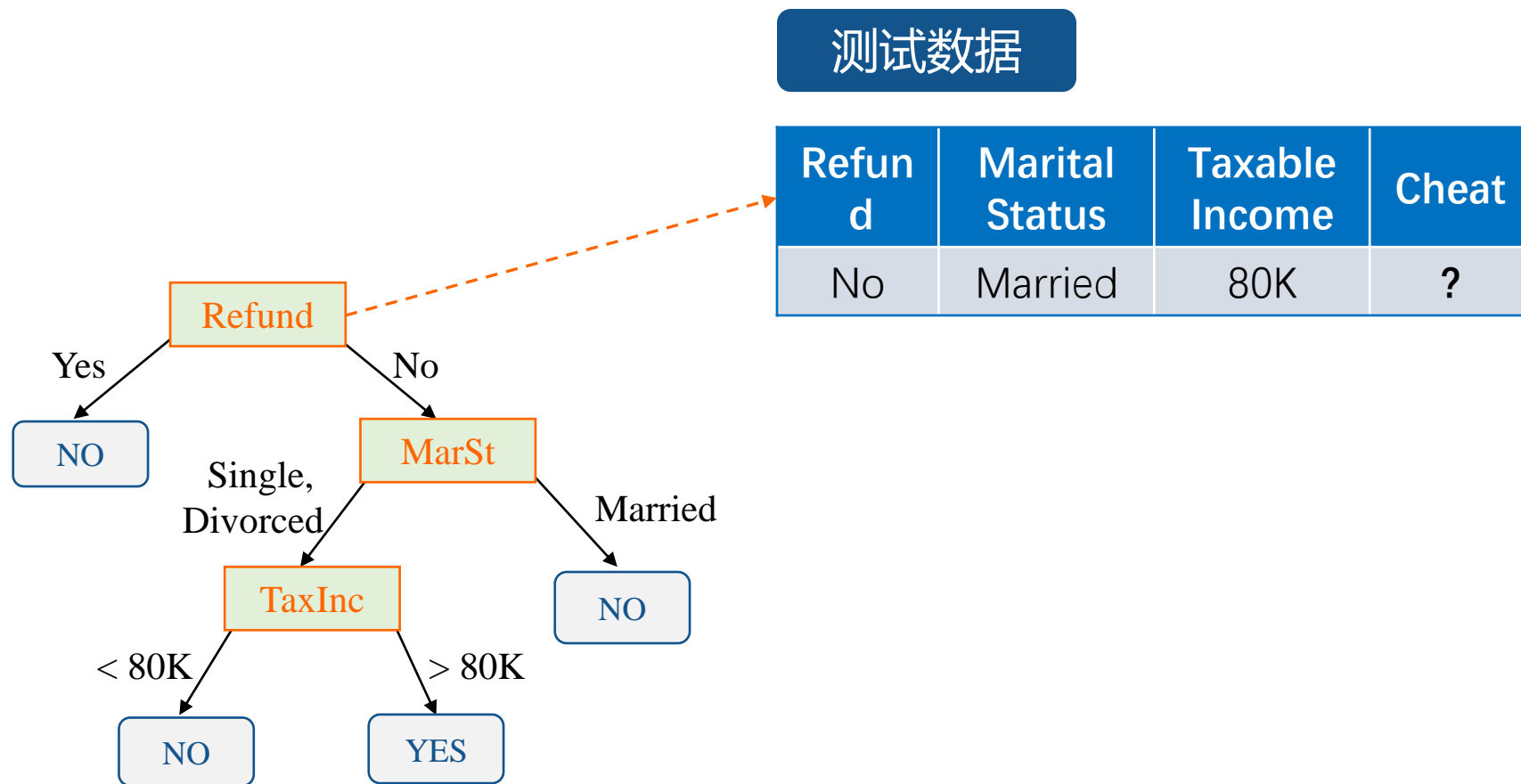
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Start from the root of tree.



从决策树的根节点到叶节点的一条路径就形成了相应对象的类别测试，决策树可以很容易转换为分类规则

3.6 决策树分类任务

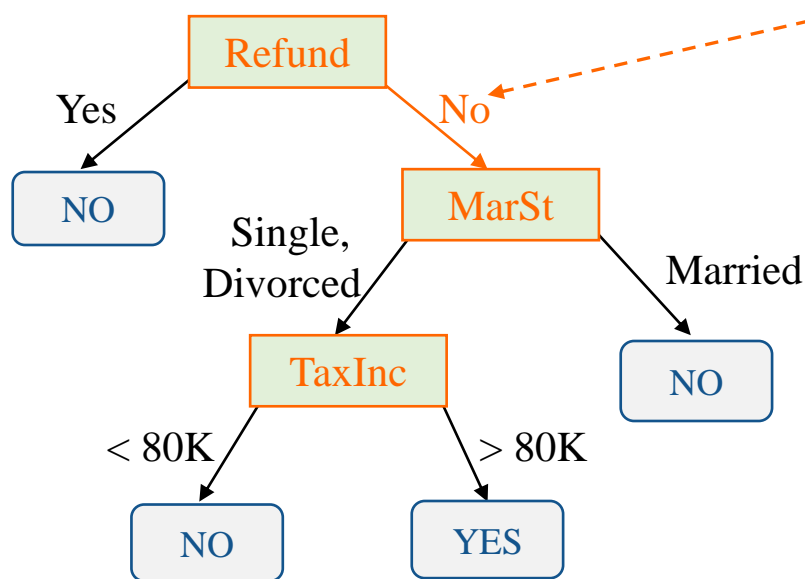


从决策树的根节点到叶节点的一条路径就形成了相应对象的类别测试，决策树可以很容易转换为分类规则

3.6 决策树分类任务

测试数据

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

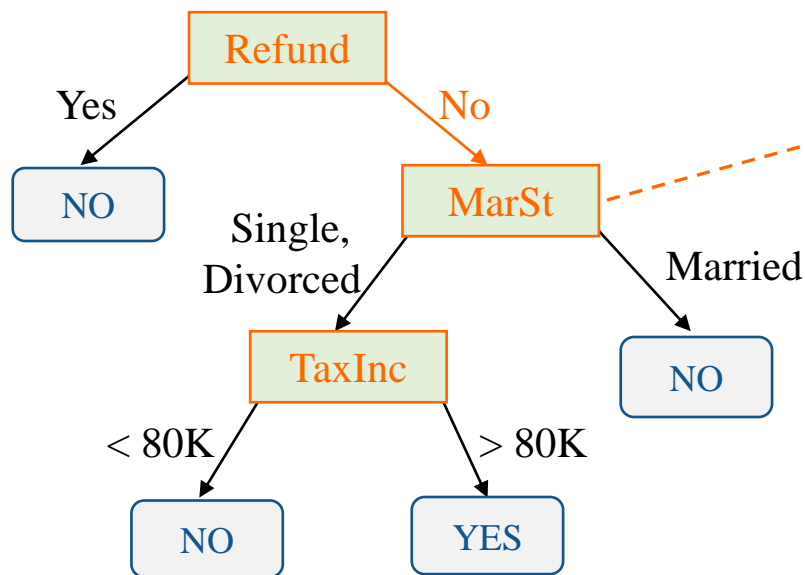


从决策树的根节点到叶节点的一条路径就形成了相应对象的类别测试，决策树可以很容易转换为分类规则

3.6 决策树分类任务

测试数据

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

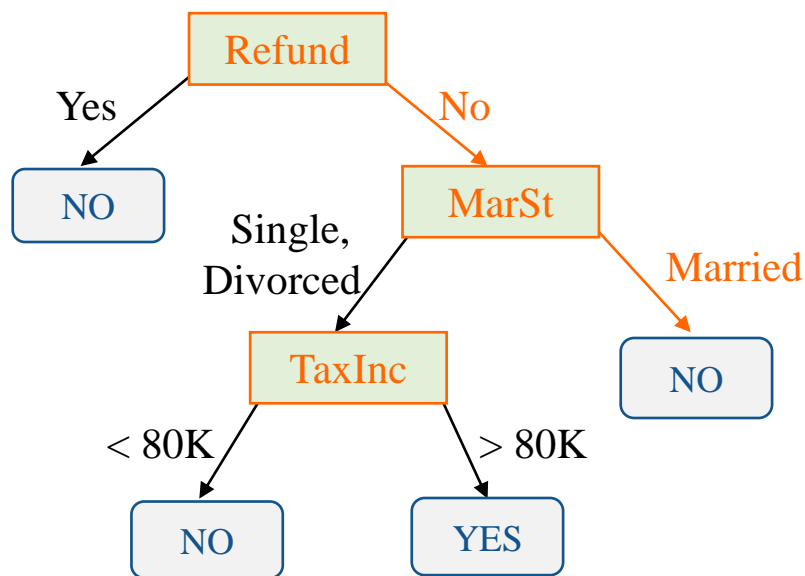


从决策树的根节点到叶节点的一条路径就形成了相应对象的类别测试，决策树可以很容易转换为分类规则

3.6 决策树分类任务

测试数据

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

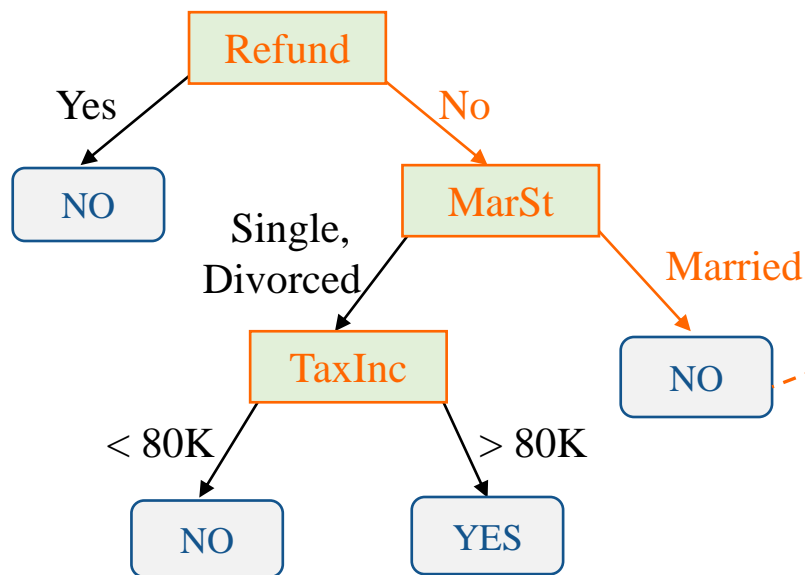


从决策树的根节点到叶节点的一条路径就形成了相应对象的类别测试，决策树可以很容易转换为分类规则

3.6 决策树分类任务

测试数据

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Assign Cheat to “No”

从决策树的根节点到叶节点的一条路径就形成了相应对象的类别测试，决策树可以很容易转换为分类规则

- 特点:

- 决策树是一种构建分类模型的非参数方法
- 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：随着数的生长，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：**特征构造**

下列说法正确的是

- ☐ A 决策树算法不能很好的解决冗余属性的问题
- ☐ B 决策树算法对噪声敏感
- ☐ C 决策树可以学习特征之间的线性关系
- ☒ D 决策树对特征的单调变换不敏感

提交

题目来源&内容

DataCastle平台 神策杯” 2018高校算法大师赛

神策杯
2018高校算法大师赛

MacBook Pro 等你来拿 现场 offer 你提不起

赛题背景：

神策数据推荐系统是基于神策分析平台的智能推荐系统。本次竞赛是模拟业务场景，以新闻文本的核心词提取为目的，最终结果达到提升推荐和用户画像的效果。

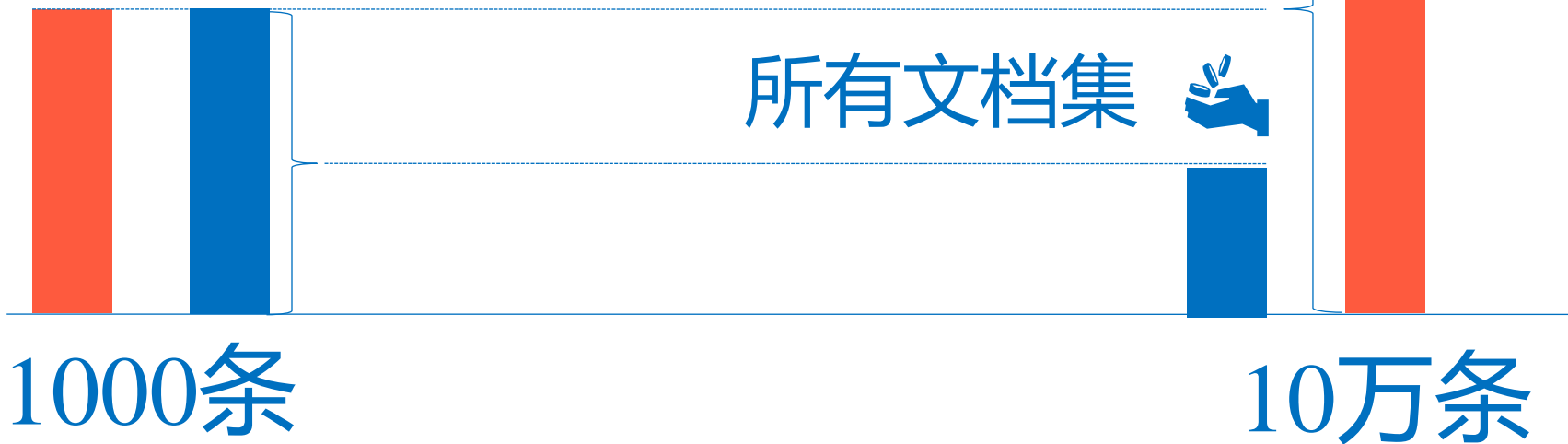
赛题内容：

以已标注关键词的1000篇文档为训练集，训练出一个“关键词提取”的模型，来提取10万篇文档的关键词。

评分原则：

选手上交对10万篇文档的标注结果，每篇文档标注两个关键词，官方在10万篇选取1000篇作为评分依据，每篇命中一个关键词记0.5分，命中两个记1分。

训练集



- all_docs.txt, 108295篇资讯文章数据, 数据格式为: ID 文章标题 文章正文, 中间由\001分割。
- train_docs_keywords.txt, 1000篇文章的关键词标注结果, 数据格式为: ID 关键词列表, 中间由\t分割。
- 所标注的文档和评分文档关键词数量大于1小于5。

all_docs.txt

ID	标题	正文
D083417	LOL：faker和恩静的前世今生恩静要结婚了，那飞科变捞的原因？	近来李哥仿佛又开始替补了，的确今年锻练的锅真的很大，算了不说了，我们聊点开心的。Faker和恩静...
D026238	可爱担当吴芊盈甜美笑容感染全场蓄力绽放非凡魅力惹人爱	近日，SDT娱乐练习生吴芊盈在新一期的《创造101》中，表现优异展现了不凡实力，成功晋级...
D066225	生一个孩子和生两个孩子有哪些区别？	虽然二胎开放了，但是有些家庭却坚持一个好，而有些家庭政策积极响应，生了二胎。一胎家庭和....
D000212	复仇者联盟3：无限战争结局，如何影响漫威影集神盾局特工	《甄嬛传》想必很多人已经二刷三刷，剧中5位小主的命运差异好大，剧...
D011909	【NCT127成员介绍】谁都认证的拥有克里斯马的队长泰容君！	13日，TOWER_官方推特公开泰容相关宣传。[#NCT127]成员介绍谁都...

ID	label
D083417	LOL,faker,恩静
D026238	吴芊盈,创造101
D066225	一胎，二胎
D000212	复仇者联盟3,无限战争,漫威,神盾局特工
D011909	NCT127,泰荣君

ID	类别
1-40000	娱乐新闻
40001-44060	体育新闻
44061-54060	健康新闻
54061-64060	军事新闻
64061-74060	正文文本
74061-84060	教育新闻
98296-108295	饮食菜谱

求解思路

简单规则

根据训练集的观察结果，制定简单规则，预测所有文档级。（无监督）

二分类

将关键词提取问题转化为二分类的问题，对每个词判断其是否为关键词的概率。

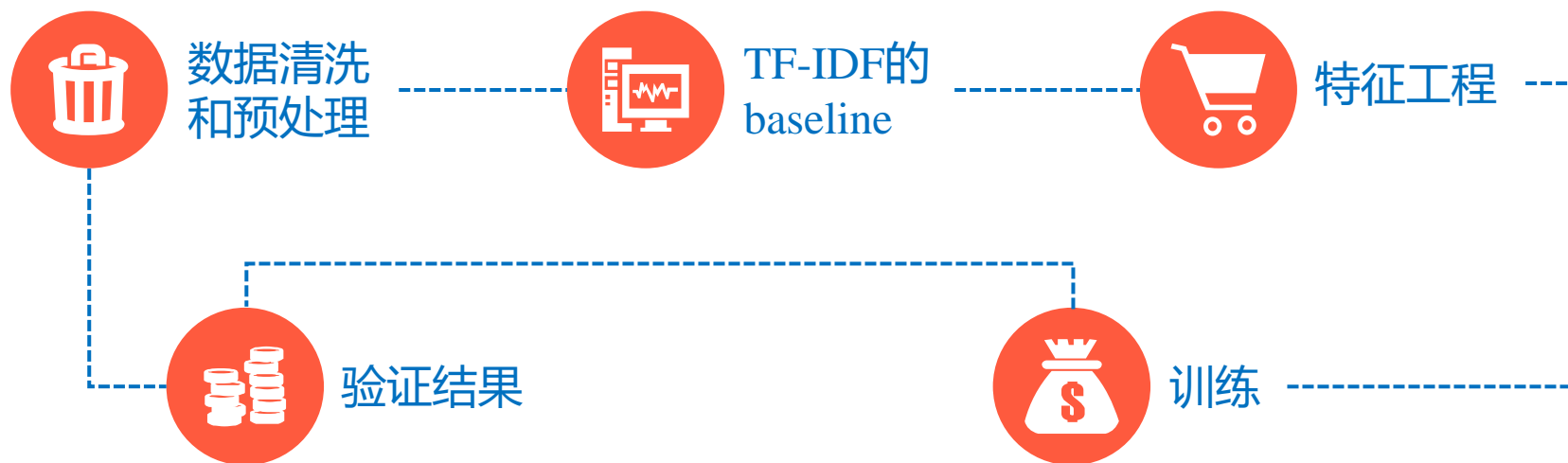
word2vec+神经网络

将文档和对应关键词表示为向量，利用神经网络。进行预测。

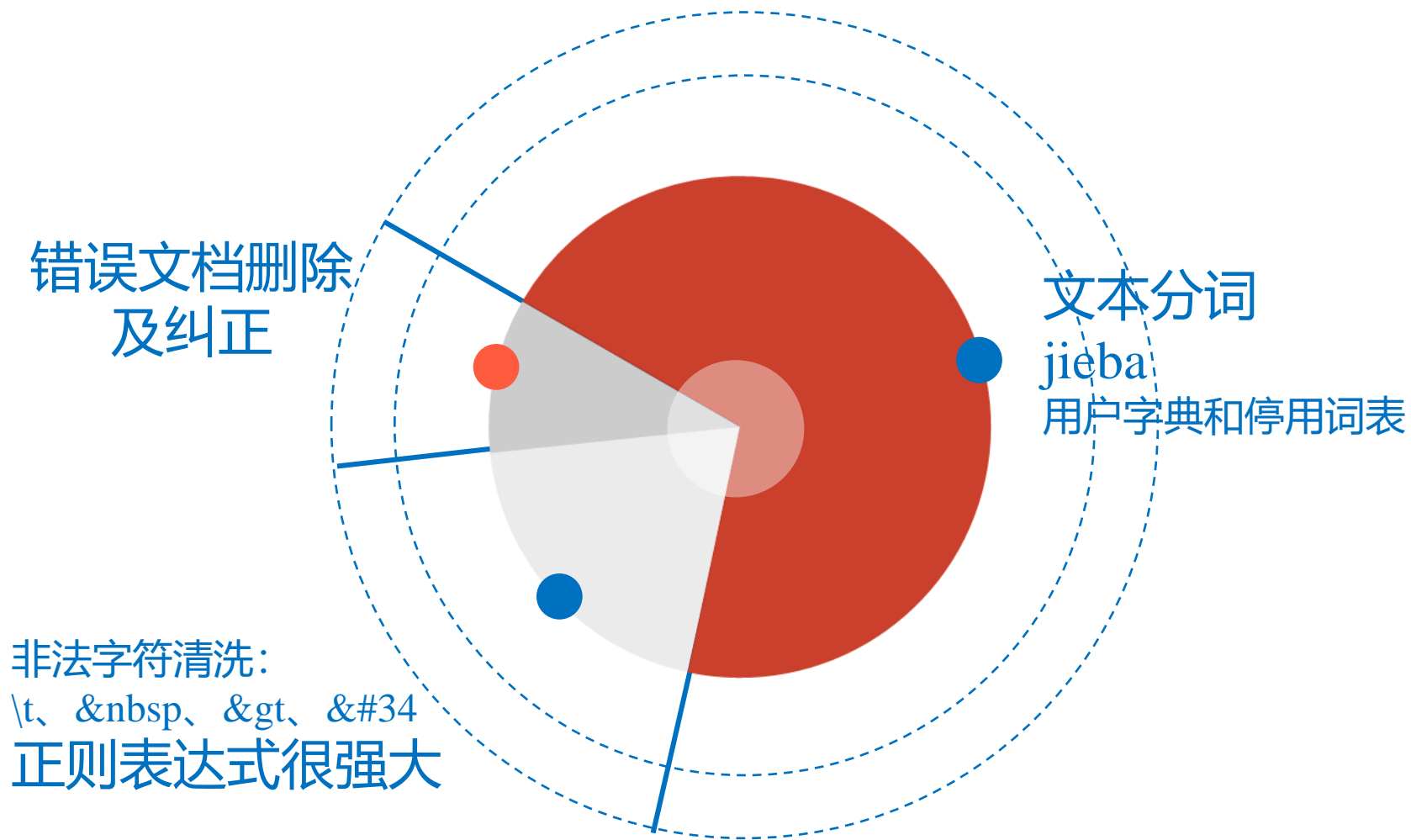
详细过程

外部依赖

pandas、numpy、jieba、jieba.analyse、re、math、sklearn、lightgbm、collections、tqdm

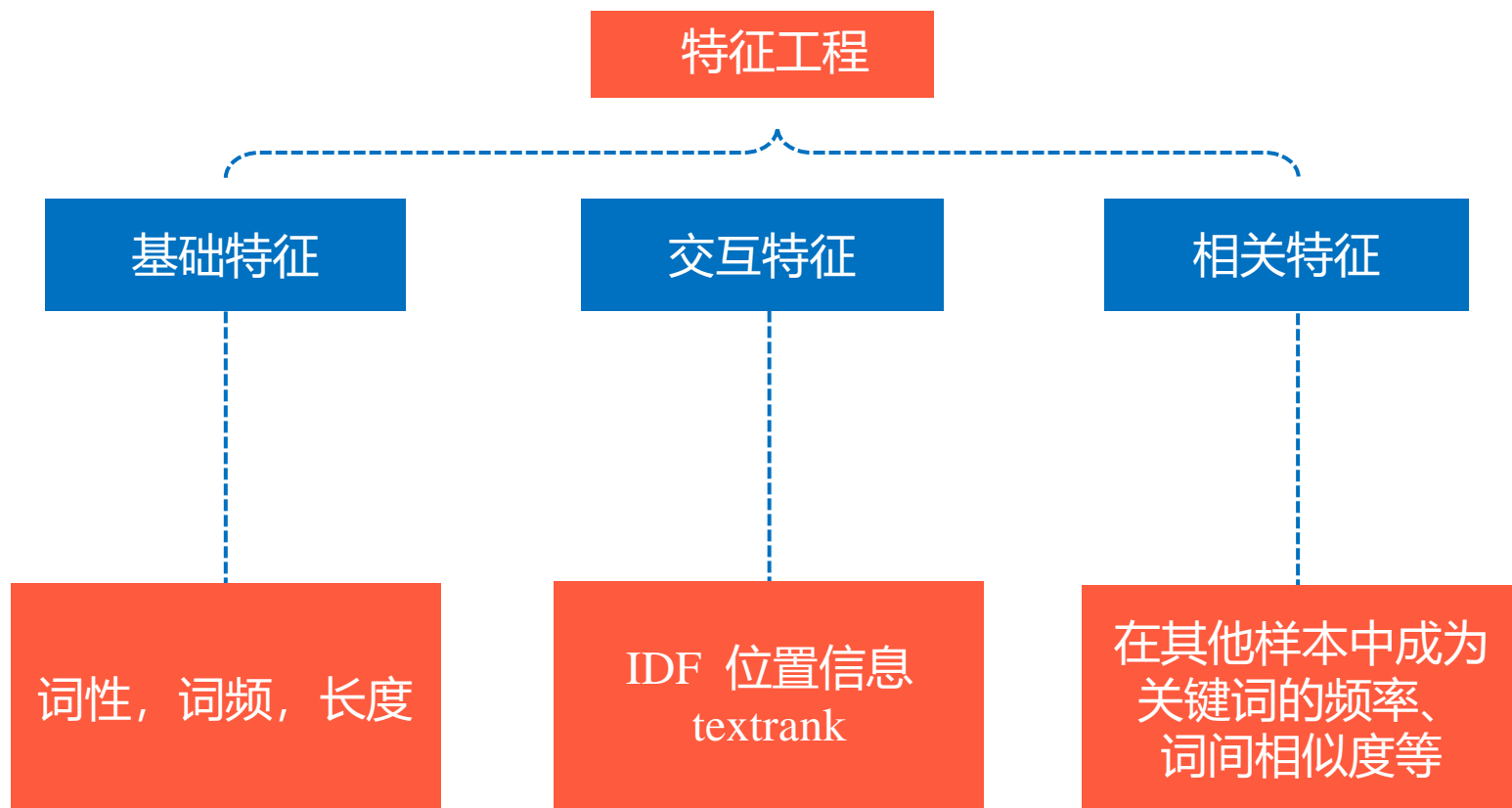


清洗及预处理



特征工程

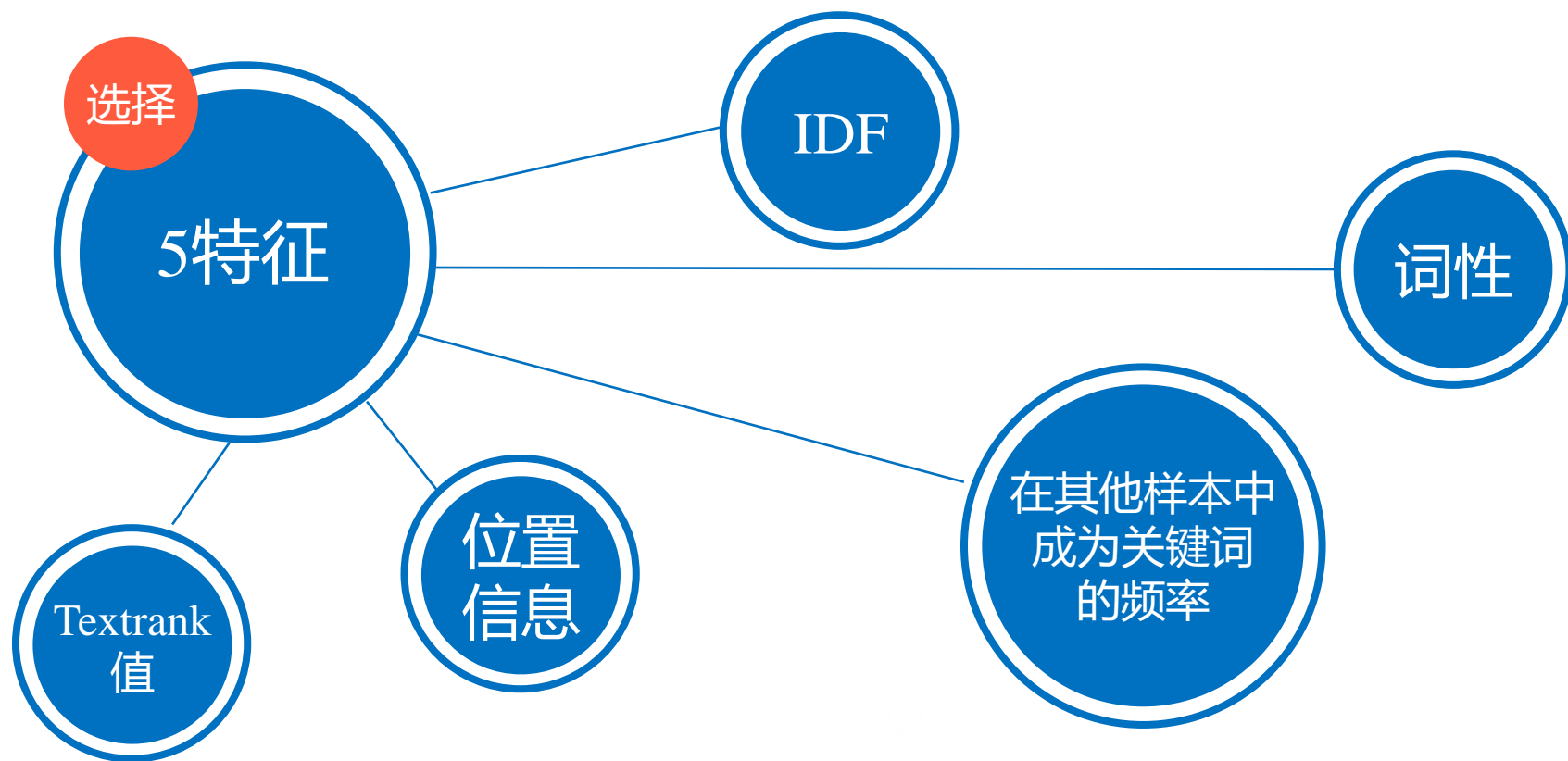
心得：特征工程的好坏程度直接决定了结果的上限



特征选择

采用sklearn.feature_selection SelectKBest、 ExtraTreesClassifier 决策树分析

其他：L1、L2正则化、循环提取



数据挖掘竞赛案例1

< 地点推荐系统 >

- 竞赛背景

移动数据

基于地点推荐技术

熟悉周遭环境

提升地点的影响力

参赛数据

地图信息

A	B	C	D	E	F	G
地点ID	纬度	经度	所在城市	粗类别	细类别	
107780	22.29743	114.1726	overseas	公共机构	公寓/小区/里弄	
70990	31.13684	121.4226	shanghai			
132379	31.23218	121.3976	shanghai			
38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆	
104522	27.7	85.33333	overseas			
91784	13.73705	100.5604	overseas	交通/住宿	地铁站/轻轨站	
97543	34.52023	112.9788	zhengzhou			
2996	31.18514	121.428	shanghai	商店/生活	时尚服饰	
96184	31.17635	121.5073	shanghai			
33986	30.71455	121.3366	shanghai			
84982	1.29695	103.8523	overseas	学校/教育	大学/研究所/专科院	
41797	31.27782	121.3654	shanghai			
60801	31.1731	121.4908	shanghai			

用户信息

	A	B	C	D
1	用户ID	地点ID	前往次数	
2	7263	112417	1	
3	7263	112416	1	
4	7262	112413	1	
5	7262	112412	1	
6	7262	112411	1	
7	7262	112410	1	
8	7261	112408	1	
9	7261	112407	1	

参赛要求

就训练集数据中的每一位用户，
各推荐50个不同的用户感兴趣的地点。

评分标准

平均截断召回率

$$\text{Recall} = \frac{1}{M} \sum_u \frac{|V_u \cap S_u|}{V_u}$$

- 协同过滤算法

一、基于用户的协同过滤算法

二、基于物品的协同过滤算法

● 实验过程

1 数据预处理

地图信息

A	B	C	D	E	F	G
地点ID	纬度	经度	所在城市	粗类别	细类别	
107780	22.29743	114.1726	overseas	公共机构	公寓/小区/里弄	
70990	31.13684	121.4226	shanghai			
132379	31.23218	121.3976	shanghai			
38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆	
104522	27.7	85.33333	overseas			
91784	13.73705	100.5604	overseas	交通/住宿	地铁站/轻轨站	
97543	34.52023	112.9788	zhengzhou			
2996	31.18514	121.428	shanghai	商店/生活	时尚服饰	
96184	31.17635	121.5073	shanghai			
33986	30.71455	121.3366	shanghai			
84982	1.29695	103.8523	overseas	学校/教育	大学/研究所/专科院	
41797	31.27782	121.3654	shanghai			
60801	31.1731	121.4908	shanghai			

用户信息

	A	B	C	D
1	用户ID	地点ID	前往次数	
2	7263	112417	1	
3	7263	112416	1	
4	7262	112413	1	
5	7262	112412	1	
6	7262	112411	1	
7	7262	112410	1	
8	7261	112408	1	
9	7261	112407	1	

根据经纬度聚类，将连续数据离散化

● 实验过程

1 数据预处理

用户ID	地点ID	纬度	经度	地点	粗类别	细类别	
592	38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆	
2761	38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆	
4266	38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆	
4608	38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆	
6598	38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆	
7531	38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆	
13255	38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆	
13693	38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆	
17482	38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆	
23743	38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆	
25023	38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆	

- 实验过程

- 2 计算相关度

皮尔逊相关系数
(Pearson correlation coefficient)

$$r(X, Y) = \frac{Cov(X, Y)}{\sqrt{Var[X] Var[Y]}}$$

$Cov(X, Y)$ 为X与Y的协方差

$Var[X]$ 为X的方差, $Var[Y]$ 为Y的方差

● 实验过程

3 设置参数

```
class recommender:
    # data: 数据集, 这里指users
    # k: 表示得出最相近的k的近邻
    # metric: 表示使用计算相似度的方法
    # n: 表示推荐place的个数
    def __init__(self, data, k=10, metric='pearson', n=50):
        #数据集data (用user), pearson矩阵, 推荐数为i+1, 最近邻为3
        #如果推荐数少于50, 可能是邻居数不够
        self.k = k
        self.n = n
        self.username2id = {}
        self.userid2name = {}
        self.productid2name = {}
```

设置K近邻的相关参数

```
# 推荐算法的主体函数
def recommend(self, user_id):
    # 定义一个字典, 用来存储推荐的地点和分数
    recommendations = {}
    # 计算出user与所有其他用户的相似度, 返回一个list
    nearest = self.computeNearestNeighbor(user_id)
    # print nearest

    userRatings = self.data[user_id] #打分=数据中的userid
    # print userRatings
    totalDistance = 0.0
    # 得住最近的k个近邻的总距离
    for i in range(self.k):
        totalDistance += nearest[i][1]
    if totalDistance == 0.0:
        totalDistance = 1.0
```









对相似度进行排序计算

● 实验过程

4 输出结果

```
id= 22242
placeid_list: ['1354', '800', '14975', '55000', '17763', '1367', '57514', '72703', '120387', '10155', '29
near_list: [('6231', 1.00000000000000475), ('9673', 1.0000000000000002), ('37061', 1.0000000000000002), ('
id= 24848
placeid_list: ['4878', '11787', '36328', '36332', '116644', '42053', '510', '55444', '52522', '55883', '4
near_list: [('48227', 1.0000000000000002), ('47316', 1.0), ('42469', 0.9999999999999999), ('30514', 0.99
id= 26802
placeid_list: ['4418', '3151', '1478', '11445', '25095', '30981', '1498', '20279', '1529', '2242', '525',
near_list: [('8147', 1.0000000002665401), ('20203', 1.0000000002665401), ('20444', 1.0000000002665401),
```

● 实验总结

4	-		lhtloviewzx	0	1	2016-11-26 21:50
5	-		148	0	2	2016-10-16 10:03
6	-		rw_personal	0	3	2016-10-22 13:11
7	-		huaming	0.00026	1	2016-11-22 14:30
8	-		beautiful	0.00082	1	2016-11-08 23:06
9	-		testmm	0.00092	1	2016-11-19 20:19
10	-		NUDT丁兆云DM	0.00166	9	2017-11-27 06:34
11	-		yshbjut	0.04306	6	2016-11-07 18:17