

好的，我将为你准备一套包含10道题目的高难度决策树练习题，涵盖信息增益、信息增益率、基尼系数、剪枝等知识点，题型包括选择题、计算题和构建决策树题。每道题将附带详细解析，帮助你深入理解相关概念。稍后我会整理好后反馈给你。

# 决策树高难度练习题

## 1. 选择题：以下关于信息增益的说法，错误的是：

- A. 信息增益衡量了使用某一属性划分数据集所获得的熵减少量。  
B. 信息增益倾向于选择取值范围较多的属性进行划分。  
C. 信息增益不受属性可能取值个数的影响。  
D. ID3算法在选择划分属性时使用信息增益准则。

正确答案： C

详细解析：

- **A：正确。** 信息增益定义为划分前后**信息熵**的差值，表示因使用某属性划分数据所**减少的不确定性**。公式上，信息增益 = 划分前数据集的熵 - 划分后各子集熵的加权和。
- **B：正确。** 信息增益存在**偏好取值较多的属性**的问题。因为取值越多的属性可以将数据集划分得更细，使每个子集更纯，从而熵降低更多，信息增益往往变大。
- **C：错误。** 实际上，信息增益会受到属性取值个数的影响：取值种类很多的属性往往得到更高的信息增益。这也是为什么需要信息增益率来校正这种偏好。
- **D：正确。** 经典ID3决策树算法选择划分属性时使用信息增益最大化的准则（选信息增益最高的属性作为节点）。

## 2. 计算题：给定如下表所示的数据集，类别标注为“是”（正类）和“否”（负类）。请计算：(1) 使用属性 X 划分时的信息增益；(2) 使用属性 Y 划分时的信息增益。并指出哪个属性具有更高的信息增益，从而更适合作为决策树根节点的划分属性。

正确答案： 属性 X 的信息增益更高，应选择 X 作为根节点划分属性。

详细解析：

1. **计算数据集的熵：** 数据集共有10个样本，其中正类5个、负类5个。熵的计算公式为

$Ent(D) = -\sum_i p_i \log_2 p_i$ 。这里  $p_{是} = 5/10 = 0.5$ ,  $p_{否} = 5/10 = 0.5$ 。因此数据集的初始熵为：

$$Ent(D) = -0.5\log_2(0.5) - 0.5\log_2(0.5) = -0.5(-1) - 0.5(-1) = 1.0\text{位}。Ent(D) = -0.5\log_2(0.5) - 0.5\log_2(0.5)$$

2. **计算属性 X 的信息增益：** 属性 X 有取值 A、B、C。根据表中统计：

- X=A 时有4个样本（全为“是”，3是、1否）。该子集熵  $Ent_{X=A} = -\frac{3}{4}\log_2 \frac{3}{4} - \frac{1}{4}\log_2 \frac{1}{4} \approx 0.811$ 。
- X=B 时有3个样本（1是、2否）。该子集熵  $Ent_{X=B} = -\frac{1}{3}\log_2 \frac{1}{3} - \frac{2}{3}\log_2 \frac{2}{3} \approx 0.918$ 。
- X=C 时有3个样本（0是、3否）。该子集熵  $Ent_{X=C} = -\frac{0}{3}\log_2 \frac{0}{3} - \frac{3}{3}\log_2 \frac{3}{3} = 0$ （纯负类熵为0）。

计算划分后的**加权熵**：

$$Ent(D | X) = 4/10 Ent_{X=A} + 3/10 Ent_{X=B} + 3/10 Ent_{X=C} \approx 0.4(0.811) + 0.3(0.918) + 0.3(0) = 0.3244$$

则属性 X 的信息增益：

$$Gain(D, X) = Ent(D) - Ent(D | X) \approx 1.0000 - 0.5998 = 0.4002。Gain(D, X) = Ent(D) - Ent(D|X) \approx 1.0$$

3. **计算属性 Y 的信息增益：** 属性 Y 有取值 P、Q。根据表中统计：

- Y=P 时有6个样本（其中4是、2否），子集熵  $Ent_{Y=P} = -\frac{4}{6}\log_2 \frac{4}{6} - \frac{2}{6}\log_2 \frac{2}{6} \approx 0.918$ 。
- Y=Q 时有4个样本（其中1是、3否），子集熵  $Ent_{Y=Q} = -\frac{1}{4}\log_2 \frac{1}{4} - \frac{3}{4}\log_2 \frac{3}{4} \approx 0.811$ 。

加权熵：

$$Ent(D | Y) = 6/10 Ent_{Y=P} + 4/10 Ent_{Y=Q} \approx 0.6(0.918) + 0.4(0.811) = 0.5508 + 0.3244 = 0.8752。Ent(D$$

属性 Y 的信息增益：

$$Gain(D, Y) = Ent(D) - Ent(D | Y) \approx 1.0000 - 0.8752 = 0.1248。Gain(D, Y) = Ent(D) - Ent(D|Y) \approx 1.0$$

4. **比较与选择属性：** 属性 X 的信息增益约为 0.4002，显著高于属性 Y 的 0.1248。因此，按照ID3算法应选择属性 X 作为根节点进行划分。直观上看，属性 X 的划分使得数据集纯度提升更多，这与我们计算的结果一致。属性 X 作为根节点划分后，其子节点熵较低，信息增益最大。

### 3. 选择题：以下关于信息增益率的说法，错误的是：

- A. 信息增益率 = 信息增益 ÷ 属性的固有值(Intrinsic Value)，用于惩罚取值过多的属性。  
B. 信息增益率在一定程度上减少了信息增益对取值数目多的属性的偏好。  
C. C4.5算法总是直接选择信息增益率数值最高的属性进行划分。  
D. C4.5算法采用信息增益率作为划分属性的评价指标。

**正确答案：** C

**详细解析：**

- **A：** 正确。信息增益率的定义即为**信息增益除以属性的“划分信息”**（属性值熵）。公式上，增益率  $GainRatio(D, a) = \frac{Gain(D, a)}{IV(a)}$ ，其中  $IV(a) = -\sum_{v \in \text{值}(a)} \frac{|D_v|}{|D|} \log_2 \frac{|D_v|}{|D|}$ 。这一比值用于抵消信息增益对取值多属性的偏好。
- **B：** 正确。由于分母“固有值”IV随着属性取值种类增多而增大，“增益率”会相对降低那些取值很多但信息增益一般的属性评分，从而在一定程度上**平衡信息增益的偏好**。不过需要注意增益率本身又会偏向取值较少的属性，因此实际算法会做出修正。
- **C：** 错误。C4.5算法并非简单选择最高增益率的属性。为了避免增益率偏爱取值很少但信息增益很低的属性，C4.5通常先从候选中**筛选信息增益高于平均值的属性**，再从中选择增益率最高的。因此并不是“不考虑信息增益只看增益率”。
- **D：** 正确。C4.5决策树在划分属性时确实采用信息增益率作为主要衡量指标之一。它相对于ID3的信息增益准则进行了改进，用以得到更好的划分属性选择。

### 4. 计算题：假设有一个二分类数据集，共有10个样本（正类和负类样本各5个）。属性 M 有10个可能取值（每个样本的 M 值都不相同，因此用 M 划分后每个样本各自成为一个叶节点）；属性 P 是一个二值属性，将数据划分为两部分：一部分包含7个样本（其中5个是正类，2个是负类），另一部分包含3个样本（其中0个是正类，3个是负类）。

请计算：使用属性 M 划分的数据集的信息增益和信息增益率，以及使用属性 P 划分的数据集的信息增益和信息增益率。根据计算结果，分别指出ID3算法和C4.5算法会选择哪一个属性作为划分依据，并解释原因。

**正确答案：** ID3算法将选择属性 M 进行划分；C4.5算法将选择属性 P 进行划分。

**详细解析：**

1. **属性 M 的信息增益：** 初始数据集熵  $Ent(D) = 1.0$  位（因为正负各占一半）。属性 M 将每个样本单独分入 10 个子集，每个子集纯度为 100%（只有一个样本）。因此划分后各子集熵均为 0，**加权熵**  $Ent(D|M) = 0$ 。

◦ 信息增益  $Gain(D, M) = Ent(D) - Ent(D|M) = 1.0 - 0 = 1.0$  位。

2. **属性 M 的信息增益率：** 由于 M 有 10 个取值，每个子集占数据集比例为 1/10。属性 M 的固有值 (Intrinsic Value) 计算为：

$$IV(M) = -\sum_{i=1}^{10} \frac{1}{10} \log_2 \frac{1}{10} = -10 \times \frac{1}{10} \log_2 (0.1) = -\log_2 (0.1) = \log_2 (10) \approx 3.3219 \text{ 位}。 IV(M) = -\sum_{i=1}^{10} \frac{1}{10} \log_2 \frac{1}{10}$$

◦ 增益率  $GainRatio(D, M) = \frac{Gain(D, M)}{IV(M)} \approx \frac{1.0}{3.3219} \approx 0.3015$ 。

3. **属性 P 的信息增益：** 属性 P 将数据划分为两个子集： $D_1$  有 7 个样本（5 是、2 否）， $D_2$  有 3 个样本（0 是、3 否）。首先计算各子集熵：

$$Ent(D_1) = -\frac{5}{7} \log_2 \frac{5}{7} - \frac{2}{7} \log_2 \frac{2}{7} \approx 0.863 \text{ 位}。$$

$$Ent(D_2) = -\frac{0}{3} \log_2 \frac{0}{3} - \frac{3}{3} \log_2 \frac{3}{3} = 0 \text{ 位}（D_2 \text{ 全为负类}）。$$

划分后的加权熵：

$$Ent(D|P) = \frac{7}{10} Ent(D_1) + \frac{3}{10} Ent(D_2) \approx 0.7(0.863) + 0.3(0) = 0.604。 Ent(D|P) = \frac{7}{10} Ent(D_1) + \frac{3}{10} Ent(D_2)$$

◦ 信息增益  $Gain(D, P) = Ent(D) - Ent(D|P) \approx 1.0 - 0.604 = 0.396$  位。

4. **属性 P 的信息增益率：** 计算属性 P 的固有值：P 有两个取值，子集比例分别为 0.7 和 0.3：

$$IV(P) = -[0.7\log 20.7 + 0.3\log 20.3] \approx -[-0.360 - 0.521] = 0.881\text{位}。IV(P) = -[0.7\log_2 0.7 + 0.3\log_2 0.3] \approx$$

- 增益率  $GainRatio(D, P) = \frac{Gain(D, P)}{IV(P)} \approx \frac{0.396}{0.881} \approx 0.450$ 。

5. **比较与算法选择：** 属性 M 虽然信息增益最高（1.0，比P的0.396大得多），但其增益率仅约0.3015，明显低于属性 P 的增益率0.450。原因是M的取值过于分散，划分信息IV很大，削弱了其增益率评分。

- 按照 **ID3算法**（以信息增益最大化为准则），会选择属性 M，因为  $Gain(D, M) > Gain(D, P)$ 。ID3 不考虑取值数量偏好，因此倾向于选择把数据“几乎记住”的属性 M。
- 按照 **C4.5算法**（以信息增益率为准则），会选择属性 P，因为  $GainRatio(D, P) > GainRatio(D, M)$ 。增益率规则偏向能有效划分且不会把数据集过度碎片化的属性。C4.5认识到属性 M 虽然记忆了训练集，但泛化能力存疑，因此会选属性 P 获得更高的增益率。该结果体现了信息增益率减轻多值属性偏好的作用：ID3选择了取值最多的 M，而C4.5改选了 P 来避免过拟合。

## 5. 选择题：以下关于基尼系数（基尼指数）的说法，错误的是：

- A. 基尼指数越小，表示数据集纯度越高。
- B. CART决策树算法使用基尼指数作为划分属性的标准。
- C. 计算基尼指数不涉及对数运算，因此比基于熵的划分计算更高效。
- D. 对于二分类任务，当两类样本各占一半时，基尼指数达到1。

**正确答案：** D

**详细解析：**

- **A：** 正确。基尼指数（Gini）用于衡量数据的不纯度，计算公式为  $Gini(D) = 1 - \sum_k p_k^2$ 。当数据集纯度越高（越偏向单一类别）时， $p_k$  中一个值接近1，其平方接近1，Gini值就越小；纯度最高（全属单一类）时  $Gini = 0$ 。反之，Gini值越大表示数据越混杂。
- **B：** 正确。CART决策树在分类问题中通常采用基尼指数作为划分标准，即每次选择**基尼指数最小**（分裂后不纯度最低）的属性进行划分。这与ID3/C4.5使用熵/增益的准则不同，但本质都是追求子节点纯度提升。
- **C：** 正确。基尼指数的计算不涉及对数，主要是平方和运算，相对信息熵的计算**更加简单**。在实际实现中，使用基尼指标选择划分的决策树计算速度会略快一些，这是其优势之一。
- **D：** 错误。对于二分类，当两类样本各50%时达到纯度最低，此时  $p_1 = p_2 = 0.5$ ，基尼指数  $= 1 - (0.5^2 + 0.5^2) = 0.5$ 。0.5是二分类问题下Gini的最大值，而不是1。基尼指数取值范围一般在0（最纯）到0.5（1 - 1/k，k为类别数，当各类均匀分布时达到最大值）。因此选项D表述错误，50%对50%时Gini应为0.5而非1。

## 6. 计算题：下表是一个包含6个样本的数据集，其中属性 A 和 B 均为二值属性，类别为二分类（“是”表示正类，“否”表示负类）。请计算按照属性 A 划分和按照属性 B 划分时的基尼指数（Gini指数），并指出在基尼指数准则下哪个属性是更优的划分选择。

样本	A	B	类别
1	0	0	是
2	0	1	是
3	0	1	否
4	1	0	否
5	1	1	是
6	1	1	否

**正确答案：** 按照属性 A 划分的基尼指数较低，因此属性 A 是更优的划分选择。

**详细解析：**

- **总体初始基尼指数：** 数据集共有正类3个、负类3个，各占一半。初始  $Gini(D) = 1 - (0.5^2 + 0.5^2) = 0.5$ 。
- **按 A 划分：**
  - A=0 子集包含样本1、2、3（2个“是”，1个“否”），该子集  $Gini = 1 - ((\frac{2}{3})^2 + (\frac{1}{3})^2) = 1 - (0.444 + 0.111) = 0.445$ 。

- A=1 子集包含样本4、5、6 (1个“是”，2个“否”)，该子集  

$$Gini = 1 - \left(\left(\frac{1}{3}\right)^2 + \left(\frac{2}{3}\right)^2\right) = 1 - (0.111 + 0.444) = 0.445。$$
- 划分后的加权基尼指数：

$$Gini(D | A) = 36(0.445) + 36(0.445) = 0.445。Gini(D|A) = \frac{3}{6}(0.445) + \frac{3}{6}(0.445) = 0.445。$$

• **按 B 划分：**

- B=0 子集包含样本1、4 (各1个“是”和“否”)， $Gini = 1 - (0.5^2 + 0.5^2) = 0.5。$
- B=1 子集包含样本2、3、5、6 (2个“是”，2个“否”)， $Gini = 1 - (0.5^2 + 0.5^2) = 0.5。$
- 加权基尼指数：

$$Gini(D | B) = 26(0.5) + 46(0.5) = 0.5。Gini(D | B) = \frac{2}{6}(0.5) + \frac{4}{6}(0.5) = 0.5。$$

- **结果分析：** 属性 A 划分后的加权Gini指数为0.445，低于属性 B 划分后的0.5，说明按 A 划分能够使子集纯度提升更多（不纯度降低更多）。因此在CART算法中会选择属性 A 进行划分，使得基尼指数最小化。相比之下，属性 B 的划分未能降低整体Gini（两组仍各有50%正负），效果较差。

## 7. 选择题：以下关于决策树预剪枝 (pre-pruning) 的说法中，错误的是：

- A. 预剪枝通过在决策树训练过程中提前停止继续分裂节点，来防止过拟合现象。  
 B. 常用的预剪枝策略包括限制决策树最大深度、设置信息增益最小阈值、要求叶节点最小样本数等。  
 C. 预剪枝可能错过一些有预测价值的细分，从而导致模型欠拟合。  
 D. 预剪枝需要先生成完整决策树，再利用验证集剪去效果不佳的分支。

**正确答案：** D

**详细解析：**

- **A：** 正确。预剪枝是在决策树生长过程中提前停止划分的一系列技术总称，其目的正是**防止过拟合**。当某次划分不满足预设的条件（如增益不足等），算法就终止该分支的继续分裂，将当前节点作为叶节点。这样可以避免树变得过深、对训练集噪声建模过度。
- **B：** 正确。常见的预剪枝策略包括：设定树的**最大深度**上限、设定**信息增益/增益率的最小阈值**（低于此值则不再分裂）、要求节点**最小样本数**（若节点样本过少则不再分裂）等。这些策略都会在构建过程中提前停止进一步的节点分裂，从而达到剪枝的效果。
- **C：** 正确。预剪枝的潜在缺点是可能过早停止了划分，**遗漏一些有用模式**。例如某次划分的信息增益略低于阈值但实际有意义，被预剪枝阻止，则模型可能因此欠拟合，无法捕捉潜在结构。预剪枝在减少过拟合的同时，也存在让模型偏差增大的风险。
- **D：** 错误。选项描述的是**后剪枝**的过程。预剪枝并不生成完整树后再剪，而是在**生成过程中**就停止分裂。利用验证集评估子树并剪除的是后剪枝（如减枝技术中的**减少错误剪枝**等）。预剪枝不需要生成完整树，也通常不需要专门的验证集，而是在训练过程中依据预先设定的准则直接决定是否分裂节点。

## 8. 选择题：以下关于决策树后剪枝 (post-pruning) 的说法，错误的是：

- A. 后剪枝在决策树完全生长之后，再对树进行简化以降低模型复杂度。  
 B. 后剪枝通常利用验证集来评估剪枝后的决策树泛化性能。  
 C. CART算法的代价复杂度剪枝(CCP)是一种典型的后剪枝方法。  
 D. 后剪枝需要在决策树生长时设定提前停止的准则（如限定最大树深度）。

**正确答案：** D

**详细解析：**

- **A：** 正确。后剪枝指让决策树先完全生长（尽可能细地划分，拟合训练数据），然后再自底向上**剪掉不必要的枝叶**，以简化模型、防止过拟合。它是在完整树生成“后”进行的剪枝操作。
- **B：** 正确。大多数后剪枝方法需要使用**独立的验证集**或者通过交叉验证，来评估剪枝对模型泛化性能的影响。例如，**减少错误剪枝**(Reduced Error Pruning) 会检查如果剪去某子树后在验证集上的错误率是否不升高，若是则执行剪枝。这样可确保剪枝不会损害模型在未见数据上的表现。
- **C：** 正确。CART决策树采用的**代价复杂度剪枝**(Cost-Complexity Pruning)是后剪枝的经典代表。它通过引入一个复杂度惩罚项 $\alpha$ ，衡量模型复杂度与训练误差之间的综合指标，然后剪去能降低这一指标的子树，从而找到最优子树大小。这个过程是在整棵树训练完毕后进行的。

- **D：错误。**选项D描述的实际上是预剪枝的做法（在树生长时设置限制条件）。后剪枝**不在生长阶段作硬性限制**，而是在生成完整树后再做优化处理。因此，后剪枝不需要在构建时设阈值限制深度或增益，而是先允许树充分生长，再根据泛化效果裁减枝叶。

**9. 计算题：在一棵已生成的决策树中，有一个子树覆盖了验证集中的30个样本。该子树（未剪枝时）对这30个验证样本中有6个分类错误。如果将整棵子树剪枝为一个叶节点（该叶节点预测类别为子树训练数据中的多数类），则这个叶节点对上述30个验证样本中有8个预测错误。根据减少错误剪枝（Reduced Error Pruning）的策略，请判断是否应该剪除该子树，并给出理由。**

**正确答案：**不应剪除该子树。

**详细解析：**根据减少错误剪枝原则，**只有当剪枝后验证集错误数不增加时才执行剪枝**。具体来看：

- 不剪枝子树时，验证集错误数为6；剪枝成叶节点后，错误数为8，**错误率变高**。剪枝使得在验证集上误分类从20%上升到26.7%，性能变差。
- 因为剪枝后错误数增加，说明该子树对泛化有意义，**剪掉它会降低模型泛化性能**。因此不应剪除该子树。相反，如果剪枝后的错误数≤剪枝前（例如剪枝后也是6或更少），则可以考虑剪枝以简化模型而不损失精度。这个例子中剪枝会损失信息，故保持子树更佳。

**10. 构建题：如下表是一个小型数据集，包含三个二值属性 A、B、C 和二分类的类别标记（“是”/“否”）。请使用 ID3 算法（信息增益准则）构建一棵决策树，能够完全正确分类该数据集的样本。请给出决策树的构建过程（如每次选择的划分属性）以及最终生成的树结构或决策规则。构建过程中请一直划分直到所有叶节点纯度达到100%或无可用属性为止。**

样本	A	B	C	类别
1	0	0	0	是
2	0	0	1	是
3	0	1	0	是
4	0	1	1	否
5	1	0	0	否
6	1	0	1	否
7	1	1	0	否
8	1	1	1	是

**正确答案：**（决策树结构如下）

- A = 0:
  - B = 0: 类别 = 是
  - B = 1:
    - C = 0: 类别 = 是
    - C = 1: 类别 = 否
- A = 1:
  - B = 0: 类别 = 否
  - B = 1:
    - C = 0: 类别 = 否
    - C = 1: 类别 = 是

**详细解析：**

1. **根节点划分属性选择：**数据集初始熵  $Ent(D) = 1$  位（正负各4个）。计算各属性的信息增益：

- $Gain(D, A)$ : 按 A 划分为两组, A=0子集(样本1-4)有3个“是”、1个“否”, A=1子集(样本5-8)有1个“是”、3个“否”。熵分别约  $Ent(A = 0) = 0.811$ 、 $Ent(A = 1) = 0.811$ , 加权熵 0.811, 信息增益  $Gain(D, A) \approx 0.189$ 。
  - $Gain(D, B)$ : 按 B 划分为两组, 各组都是2“是”/2“否” (见表中B=0行和B=1行各有2是2否), 熵均为1, 加权熵仍为1, 信息增益  $Gain(D, B) = 0$ 。
  - $Gain(D, C)$ : 按 C 划分两组, 各组也是2“是”/2“否” (C=0组与C=1组各2是2否), 信息增益  $Gain(D, C) = 0$ 。
- 因此属性 A 信息增益最大, 选择 **A 作为根节点** 进行划分。

2. **划分 A=0 子集**: A=0组 (样本1-4) 有3是、1否, 需继续划分。剩余属性: B、C。计算该子集上B、C的信息增益:

- 在 A=0 子集中, 按 B 划分: B=0得到样本1、2 (2是0否, 纯), B=1得到样本3、4 (1是1否)。信息增益  $Gain_{A0}(B) = Ent(A = 0) - [\frac{2}{4}Ent(B = 0) + \frac{2}{4}Ent(B = 1)] = 0.811 - [0.5 * 0 + 0.5 * 1] = 0.811 - 0.5 = 0.311$ 。
  - 按 C 划分: C=0得到样本1、3 (2是0否, 纯), C=1得到样本2、4 (1是1否)。计算可知  $Gain_{A0}(C)$  也约为 0.311 (与按B划分情况对称)。
- 增益相等情况下可任选其一 (假设算法按属性名顺序选择 B)。于是**选择属性 B 划分 A=0 子集**。
- 划分结果: 对于 A=0: B=0 分支包含样本1、2 (全是“是”) 成为叶节点, 类别判定为“是”; B=1 分支包含样本3、4 (1是1否, 非纯), 需继续划分。

3. **划分 A=0, B=1 子集**: 剩余属性仅有 C。对子集(样本3、4)按 C 划分:

- C=0 分支包含样本3 (“是”), C=1 分支包含样本4 (“否”)。两个分支均纯, 成为叶节点。于是得到: 在 A=0且B=1情况下, **C=0时类别=是, C=1时类别=否**。

4. **划分 A=1 子集**: A=1组 (样本5-8) 有1是、3否, 需继续划分。剩余属性: B、C。类似地计算该组的信息增益: 由于数据分布对称, 可知按 B 划分信息增益约0.311, 按 C 划分信息增益约0.311 (在样本5-8中, B、C 划分效果相同, 各自能将该组3:1的分布划分得一边纯、一边1:1)。假定仍按属性顺序选择 **B 划分 A=1 子集**。

- 划分结果: 对于 A=1: B=0 分支包含样本5、6 (0是2否, 纯负类) 成为叶节点, 类别判定为“否”; B=1 分支包含样本7、8 (1否1是) 需继续划分。

5. **划分 A=1, B=1 子集**: 剩余属性为 C。对子集(样本7、8)按 C 划分:

- C=0 分支含样本7 (“否”), C=1 分支含样本8 (“是”), 均为纯叶节点。于是: 在 A=1且B=1情况下, **C=0时类别=否, C=1时类别=是**。

6. **构建结果**: 综合以上拆分, 决策树构建完成, 各叶节点均已纯。最终决策树结构可描述为:

- 如果 **A=0 且 B=0**, 则类别判断为“是” (对应样本1、2);
  - 如果 **A=0 且 B=1 且 C=0**, 则类别=“是” (样本3); 如果 **A=0 且 B=1 且 C=1**, 则类别=“否” (样本4)。
  - 如果 **A=1 且 B=0**, 则类别=“否” (样本5、6);
  - 如果 **A=1 且 B=1 且 C=0**, 则类别=“否” (样本7); 如果 **A=1 且 B=1 且 C=1**, 则类别=“是” (样本8)。
- 这一规则集可对应到上面给出的树结构, 每条路径从根到叶描述了一个决策条件组合和最终分类。该决策树成功将所有训练样本正确分类, 且每个叶节点纯度达到100%。