第 11 章

1：

决策节点通过分支连接在一起,连接路径自根节点向下,直至叶节点终止.从根节点开始,在决策节点进行属性测试,每个可能的结果产生一个分支.每个分支要么与另一个决策节点相连,要么到达一个终止叶节点.

①当前节点样本数小于允许分裂的最小样本数

②任何分裂不能导致子节点的样本数小于叶子节点最少样本数

③当前节点不纯度小于分裂的不纯度阈值

④当前节点的深度大于等于数的最大深度。

简述 CART 算法的基本原理,写出其最佳拆分判断的数学表达式,解释其中每个符号的物理意义,及整个公式的物理意义.答:CART 算法将训练数据集中具有相同目标属性值的记录递归地划分为一些记录子集,对所有可用的变量和所有可能存在的划分值进行穷举搜索划分.按照标准选择最优的划分为每个决策节点构建树.最佳拆分判断的数学表达式:

t_L:节点 t 的左子节点，t_R:节点 t 的右子节点,P_L=(t_L 的记录数)/训练集的记录数 P_R=(t_R 的记录数)/训练集的记录数，P(j|t_L)=(在 t_L 处,j 类的记录数)/节点 t 的记录数,P(j|t_R)=(在 t_R 处,j 类的记录数)/节点 t 的记录数,在节点 t 处所有可能的划分中,最佳划分是按照 Φ(s|t)方法得到的最大值.

$$Z_{dbta}=\frac{\left(p_1-p_2\right)}{\sqrt{p_{\text{pooled}}\left(1-p_{\text{pooled}}\right)\left(1/n_1-1/n_2\right)}}$$

2：

不可以；解决方法：将此连续变量分为连续的几段，通过划分成区域对其进行分类。

连续变量是不可分的，必须将其转化位离散变量。

3：

错，决策树在每个节点都追求建立一组尽可能“纯”的叶节点，也就是在一个特定的叶节点中记录具有相同的分类。

$$Z_{dbta}=\frac{\left(p_1-p_2\right)}{\sqrt{p_{\text{pooled}}\left(1-p_{\text{pooled}}\right)\left(1/n_1-1/n_2\right)}}$$

由这个公式我们可以可以获知，叶子节点是不可分的最小的单元，叶子内的节点必须具有相同的属性。

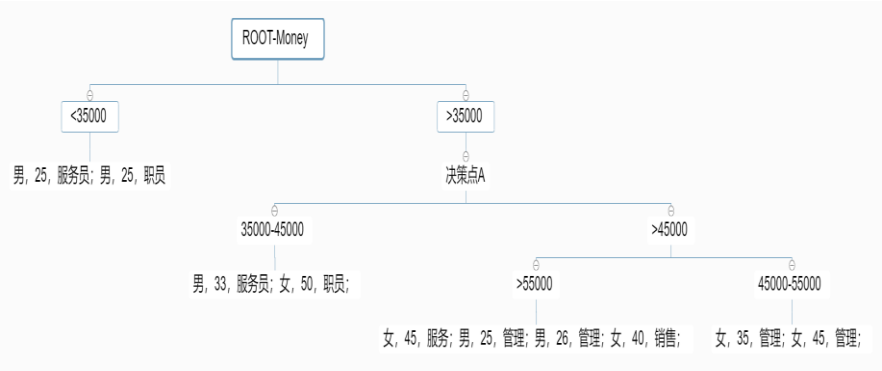
4：

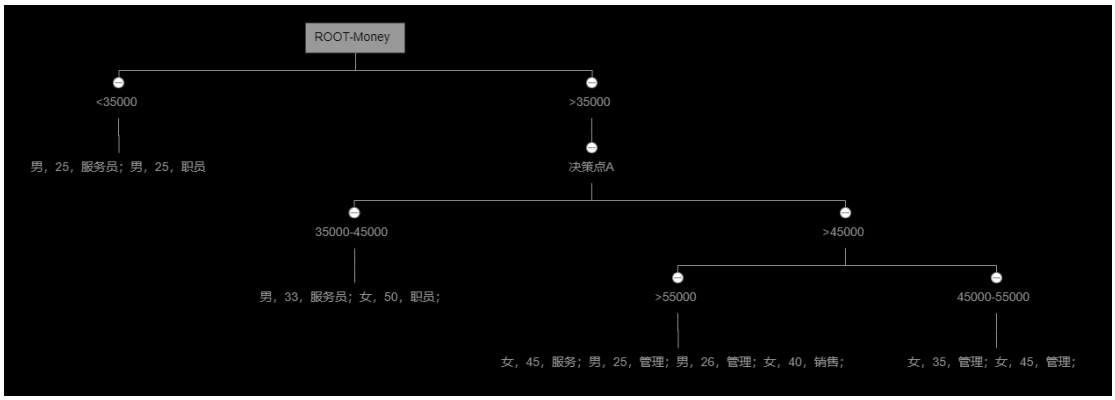
二叉树：二叉树的划分限制为二元，对所有可用变量和所有可能存在的划分值进行穷举搜索划分，在每个节点找到最佳的划分。二叉树是每个节点最多有两个子树的树结构。通常子树被称作“左子树”（leftsubtree）和“右子树”（rightsubtree）。二叉树常被用于实现二叉查找树和二叉堆。二叉树的每个结点至多只有二棵子树(不存在度大

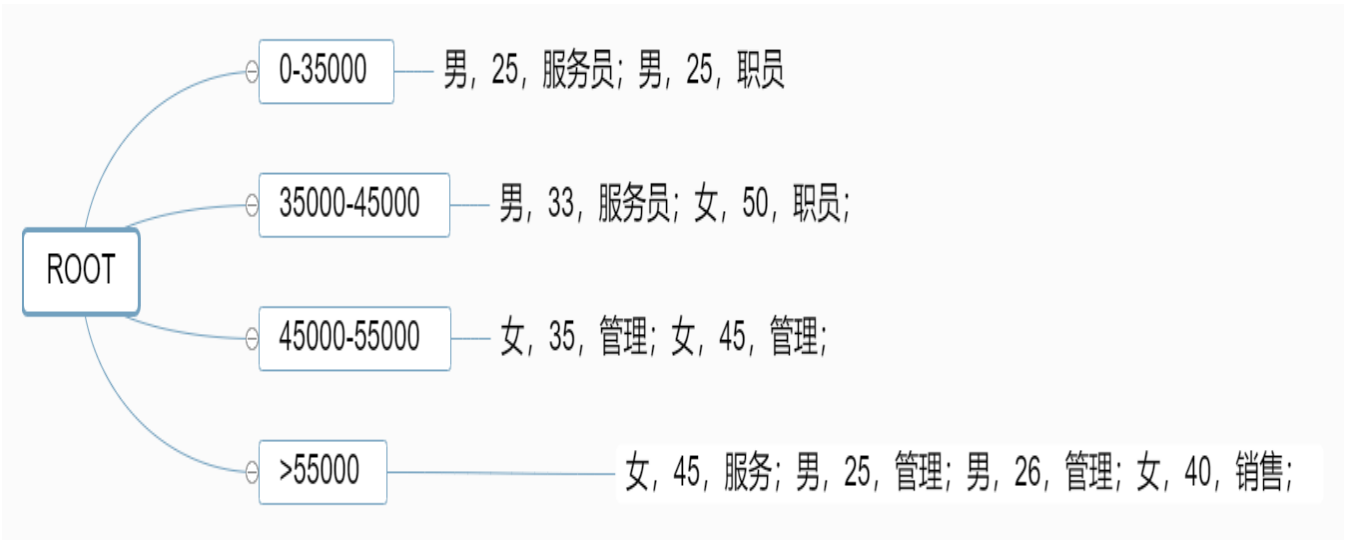
于 2 的结点)，二叉树的子树有左右之分，次序不能颠倒。二叉树的第 i 层至多有 2^{i-1}个结点；深度为 k 的二叉树至多有 2^k-1 个结点；对任何一棵二叉树 T，如果其终端结点数为 n_0，度为 2 的结点数为 n_2，则 n_0=n_2+1。一棵深度为 k，且有 2^k-1 个节点称之为满二叉树；深度为 k，有 n 个节点的二叉树，当且仅当其每一个节点都与深度为 k 的满二叉树中，序号为 1 至 n 的节点对应时，称之为完全二叉树。但是最终可能并非所有叶节点都一定属于同一个类别，会导致一定程度的分类错误，

浓密树（一般树）：浓密树将分类变量的值进行划分，准确率较高。但是可能会导致过度浓密的树，使得相当多的叶节点仅包含很少的记录。

5：6：







[{"data":{"id":"d9586144c0de","created":1607952765,"text":"ROOT-Money","background":"#ffffff","color":"#000000","font-size":24},"children":[{"data":{"id":"c7yaqbd2kps0","created":1608536867016,"text":"<35000","layout":null,"background":"#ffffff","color":"#000000","font-size":24},"children":[{"data":{"id":"c7yaqy56lww0","created":1608536916605,"text":"男，25，服务员；男，25，职员","layout":null,"background":"#ffffff","color":"#000000","font-size":24},"children":[]}]},{"data":{"id":"c7yaqvahi5s0","created":1608536910397,"text":">35000","layout":null,"background":"#ffffff","color":"#000000","font-size":24},"children":[{"data":{"id":"c7yarr5fqy80","created":1608536979747,"text":"决策点 A","layout_right_offset":{"x":-9,"y":5},"layout":null,"background":"#ffffff","color":"#000000","font-size":24},"children":[{"data":{"id":"c7yas8z5ec00","created":16085

37018549,"text":"35000-
45000","layout":null,"background":"#ffffff","color":"#000000","fo
nt-
size":24},"children":[{"data":{"id":"c7yas9hu4nc0","created":1608
537019679,"text":" 男， 33， 服务员； 女， 50， 职员；
","layout":null,"background":"#ffffff","color":"#000000","font-
size":24},"children":[]}]},{"data":{"id":"c7yasadonrc0","created":16
08537021605,"text":">45000","layout":null,"background":"#ffffff
","color":"#000000","font-
size":24},"children":[{"data":{"id":"c7yatfpwnds0","created":16085
37111592,"text":">55000","layout":null,"background":"#ffffff","c
olor":"#000000","font-
size":24},"children":[{"data":{"id":"c7yatyqcu7s0","created":16085
37152978,"text":"女，45，服务；男，25，管理；男，26，管理；女，
40 ， 销 售 ；
","layout":null,"background":"#ffffff","color":"#000000","font-
size":24},"children":[]}]},{"data":{"id":"c7yatl7wbs00","created":16
08537123564,"text":"45000-
55000","layout":null,"background":"#ffffff","color":"#000000","fo
nt-
size":24},"children":[{"data":{"id":"c7yaulczz6g0","created":16085
37202236,"text":" 女， 35， 管 理； 女， 45， 管 理；

","layout":null,"background":"#ffffff","color":"#000000","font-size":24},"children":[]}]}]}]}]}]}]

7：

分类与回归树：每个节点能找到最佳的划分，但是最终可能并非所有叶节点都一定属于同一个类别，会导致一定程度的分类错误。CART 算法的二分法可以简化决策树的规模，提高生成决策树的效率。分裂：分裂过程是一个二叉递归划分过程，其输入和预测特征既可以是连续型的也可以是离散型的，CART 没有停止准则，会一直生长下去；剪枝：采用代价复杂度剪枝，从最大树开始，每次选择训练数据熵对整体性能贡献最小的那个分裂节点作为下一个剪枝对象，直到只剩下根节点。CART 会产生一系列嵌套的剪枝树，需要从中选出一颗最优的决策树；树选择：用单独的测试集评估每棵剪枝树的预测性能（也可以用交叉验证）。C4.5 为多叉树，运算速度慢，CART 为二叉树，运算速度快；C4.5 只能分类，CART 既可以分类也可以回归；CART 使用 Gini 系数作为变量的不纯度量，减少了大量的对数运算；CART 采用代理测试来估计缺失值，而 C4.5 以不同概率划分到不同节点中；CART 采用“基于代价复杂度剪枝”方法进行剪枝，而 C4.5 采用悲观剪枝方法。

C4.5 决策树：将分类变量的值进行划分，准确率较高。但是可能会导致过度浓密的树，使得相当多的叶节点仅包含很少的记录。:C4.5 算法使用信息增益或熵降低的概念来选择最优划分.假设有一个候选

划分 S,将训练数据集 T 分成若干子集 T1,T2,...,TK;H(T)为 T 的信息熵;pi 代表子集 Ti 的记录比例.

C4.5 算法最大的特点是克服了 ID3 对特征数目的偏重这一缺点，引入信息增益率来作为分类标准。

- 引入悲观剪枝策略进行后剪枝；
- 引入信息增益率作为划分标准；
- 将连续特征离散化，假设 n 个样本的连续特征 A 有 m 个取值，C4.5 将其排序并取相邻两样本值的平均数共 m-1 个划分点，分别计算以该划分点作为二元分类点时的信息增益，并选择信息增益最大的点作为该连续特征的二元离散分类点；