Chapter 9 –
Classification and Regression Trees

Instructor: Zach Zhizhong ZHOU,
Shanghai Jiao Tong University
主讲教师：周志中，上海交通大学

Data Mining for Business Intelligence

Shmueli, Patel & Bruce

Trees and Rules 树和规则

Goal: Classify or predict an outcome based on a set of predictors 目标：根据一系列预测因子对结果进行分类或者预测

The output is a set of rules 输出是一系列规则

Example:

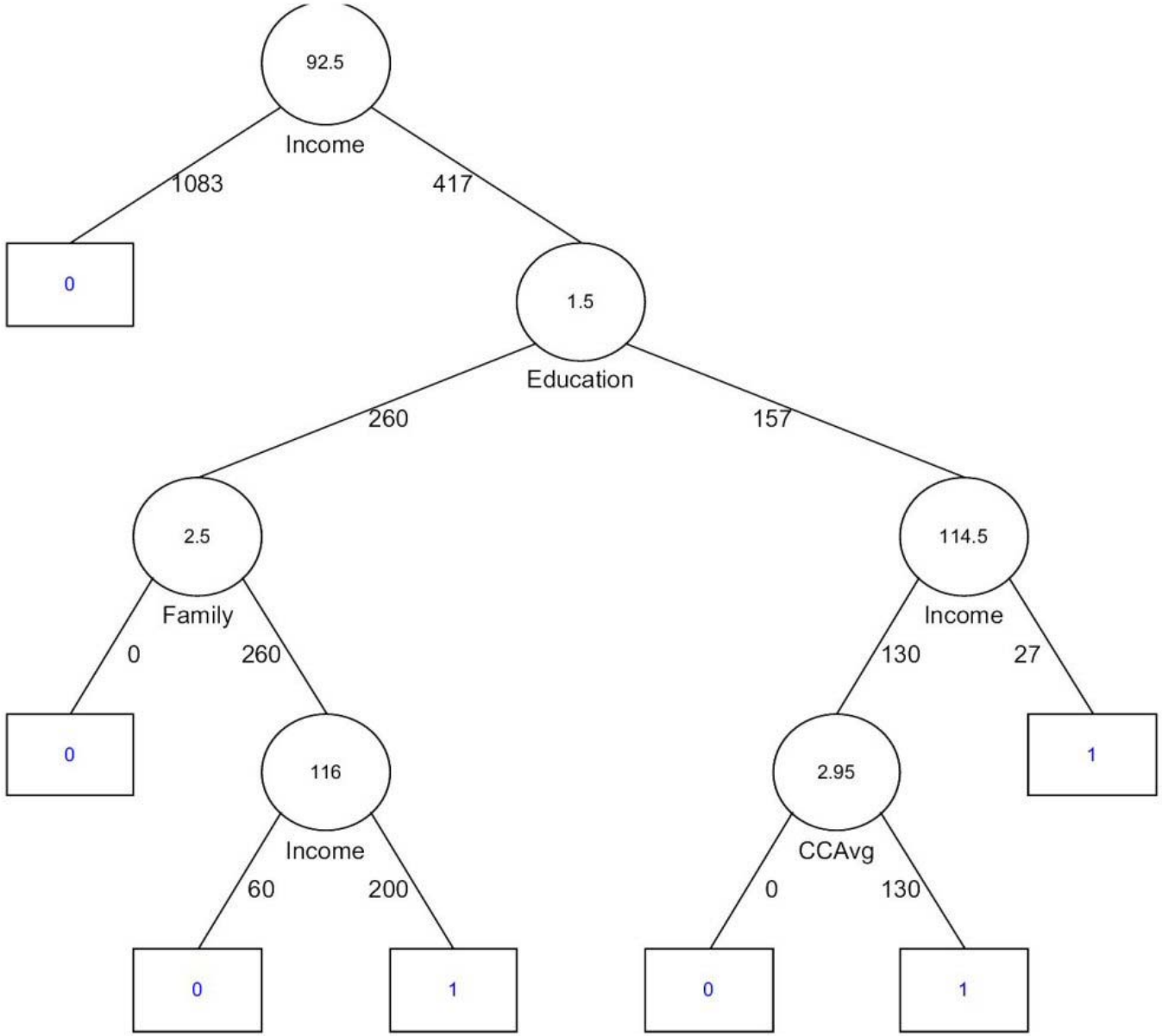
Goal: classify a record as “will accept credit card offer” or “will not accept” 目标：将一个记录归类为“将会接受信用卡优惠”或者“不会接受信用卡优惠”

Rule might be “IF (Income > 92.5) AND (Education < 1.5) AND (Family <= 2.5) THEN Class = 0 (nonacceptor) 规则可能是如果收入大于92.5并且教育大于1.5并且家庭小于或等于2.5，那么将记录归类为0 （不接受信用卡）

Also called CART, Decision Trees, or just Trees 称之为CART或者决策树或者树。

Rules are represented by tree diagrams 数的枝叶表示规则

Trees and Rules 树和规则



Key Ideas 关键思想

Recursive partitioning: Repeatedly split the records into two parts so as to achieve maximum homogeneity within the new parts
递归划分：重复将记录划为2部分，尽可能确保每部分的记录有最大的同质性。

Pruning the tree: Simplify the tree by pruning peripheral branches to avoid overfitting 修剪枝叶：通过修剪边缘枝叶简化树结构，防止过度拟合。

Recursive Partitioning 递归划分

Recursive Partitioning Steps 递归划分步骤

- Pick one of the predictor variables, x_i 挑出一个预测变量 x_i

- Pick a value of x_i, say s_i, that divides the training data into two (not necessarily equal) portions 找出预测变量的某个值 s_i 用来把训练数据集分成2部分（数据量不一定相等）。

- Measure how "pure" or homogeneous each of the resulting portions are 衡量每个部分的记录的"纯度"或者同质性

  "Pure" = containing records of mostly one class

  "纯度"=包含的绝大多数记录是一种类别。

Recursive Partitioning Steps 递归划分步骤

- Algorithm tries different values of x_i, and s_i to maximize purity in initial split 算法尝试不同的x_i, 和 s_i 的取值，最大化初次划分结果的“纯度”

- After you get a “maximum purity” split, repeat the process for a second split, and so on 当你得到最大纯度的划分结果，重复以上过程继续进行划分。

Example: Riding Mowers 例子: 刈草车

- Goal: Classify 24 households as owning or not owning riding mowers 目标：对24个家庭进行分类判断他们是否拥有刈草车。

- Predictors = Income, Lot Size 预测变量：收入、房屋面积

Income	Lot_Size	Ownership
60.0	18.4	owner
85.5	16.8	owner
64.8	21.6	owner
61.5	20.8	owner
87.0	23.6	owner
110.1	19.2	owner
108.0	17.6	owner
82.8	22.4	owner
69.0	20.0	owner
93.0	20.8	owner
51.0	22.0	owner
81.0	20.0	owner
75.0	19.6	non-owner
52.8	20.8	non-owner
64.8	17.2	non-owner
43.2	20.4	non-owner
84.0	17.6	non-owner
49.2	17.6	non-owner
59.4	16.0	non-owner
66.0	18.4	non-owner
47.4	16.4	non-owner
33.0	18.8	non-owner
51.0	14.0	non-owner
63.0	14.8	non-owner

How to split 如何划分?

- Order records according to one variable, say lot size
  数据根据一个变量比如房屋面积进行排序。

- Find midpoints between successive values

  E.g. first midpoint is 14.4 (halfway between 14.0 and 14.8)

- Divide records into those with lotsize > 14.4 and those < 14.4

- After evaluating that split, try the next one, which is 15.4 (halfway between 14.8 and 16.0)

Note: Categorical Variables 注意：类别型变量

- Examine all possible ways in which the categories can be split. 检查所有可能的类别型变量划分方式

- E.g., categories A, B, C can be split 3 ways 如果有3个类别，那么它们可以采用以下划分方式:

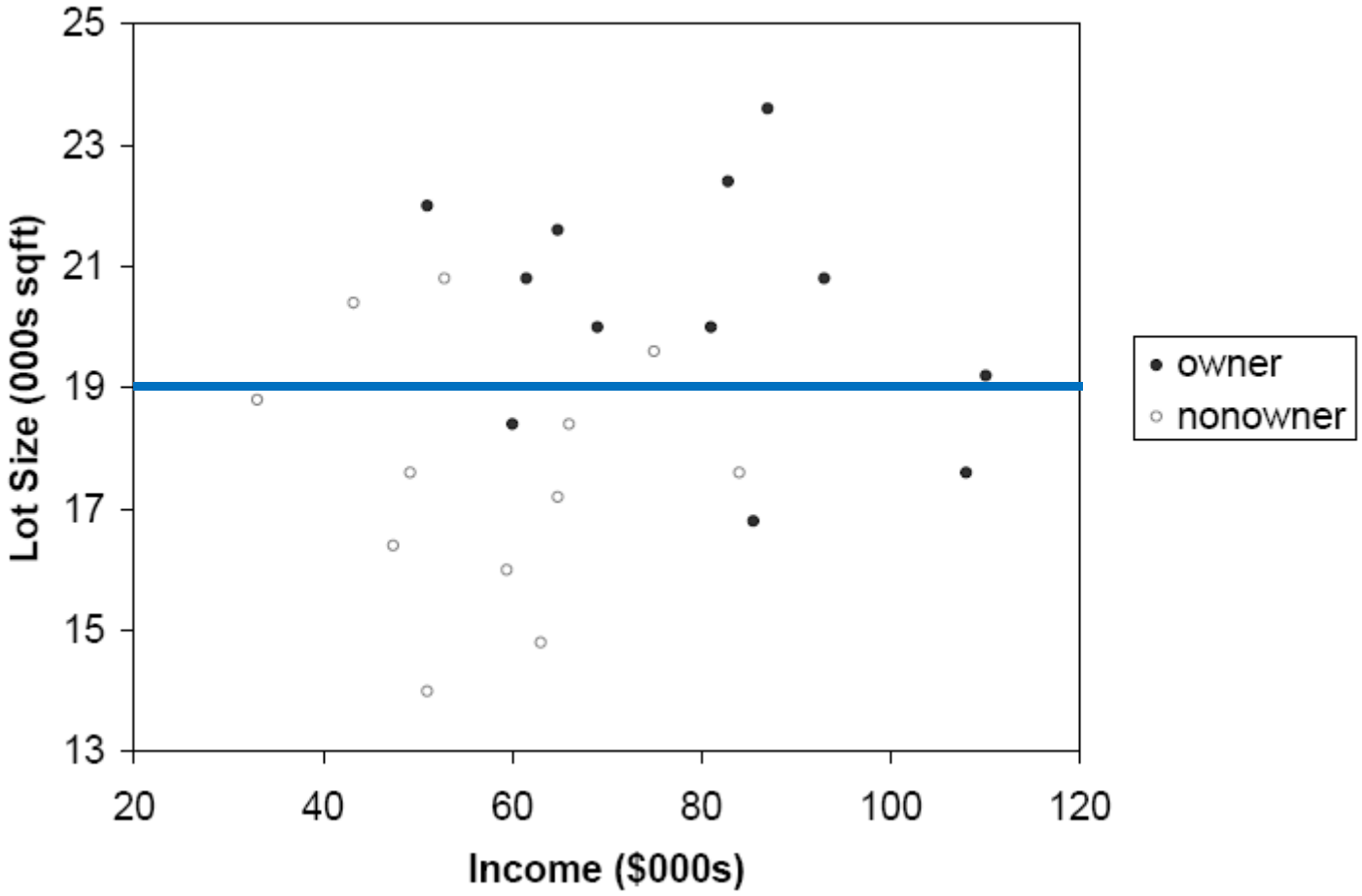
{A} and {B, C}

{B} and {A, C}

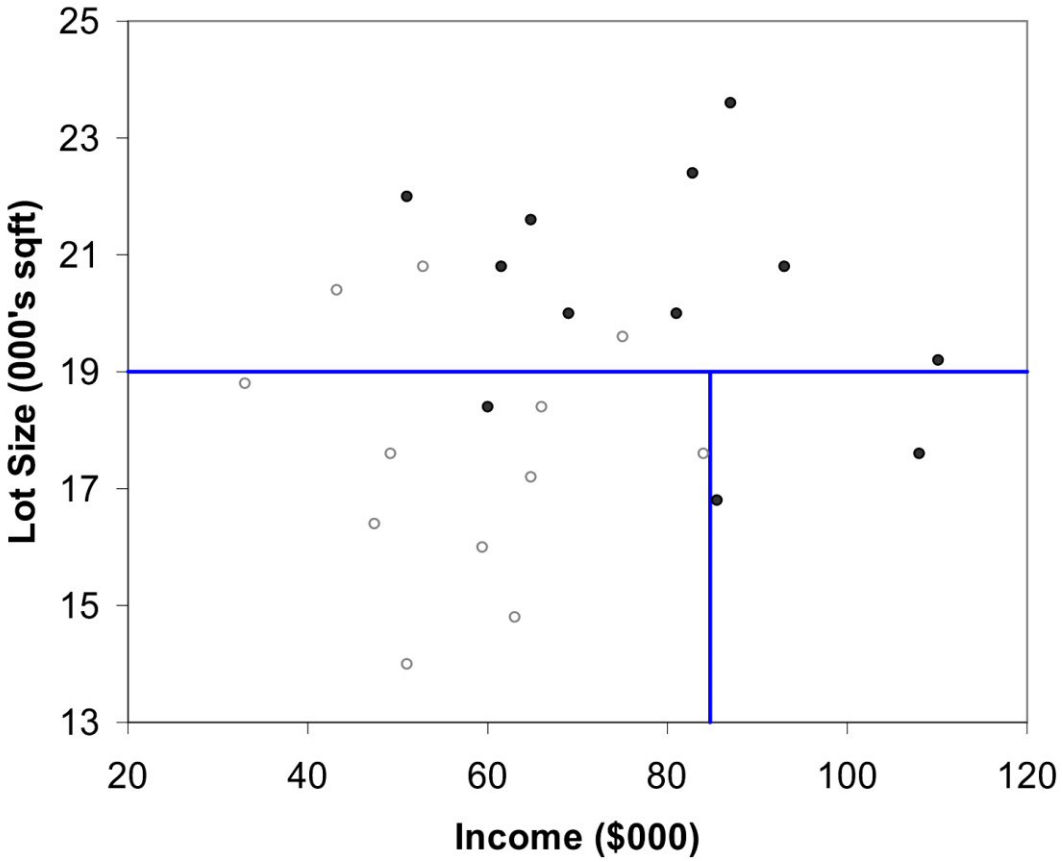
{C} and {A, B}

- With many categories, # of splits becomes huge 如果有很多类别，那么划分的可能性就会非常多。

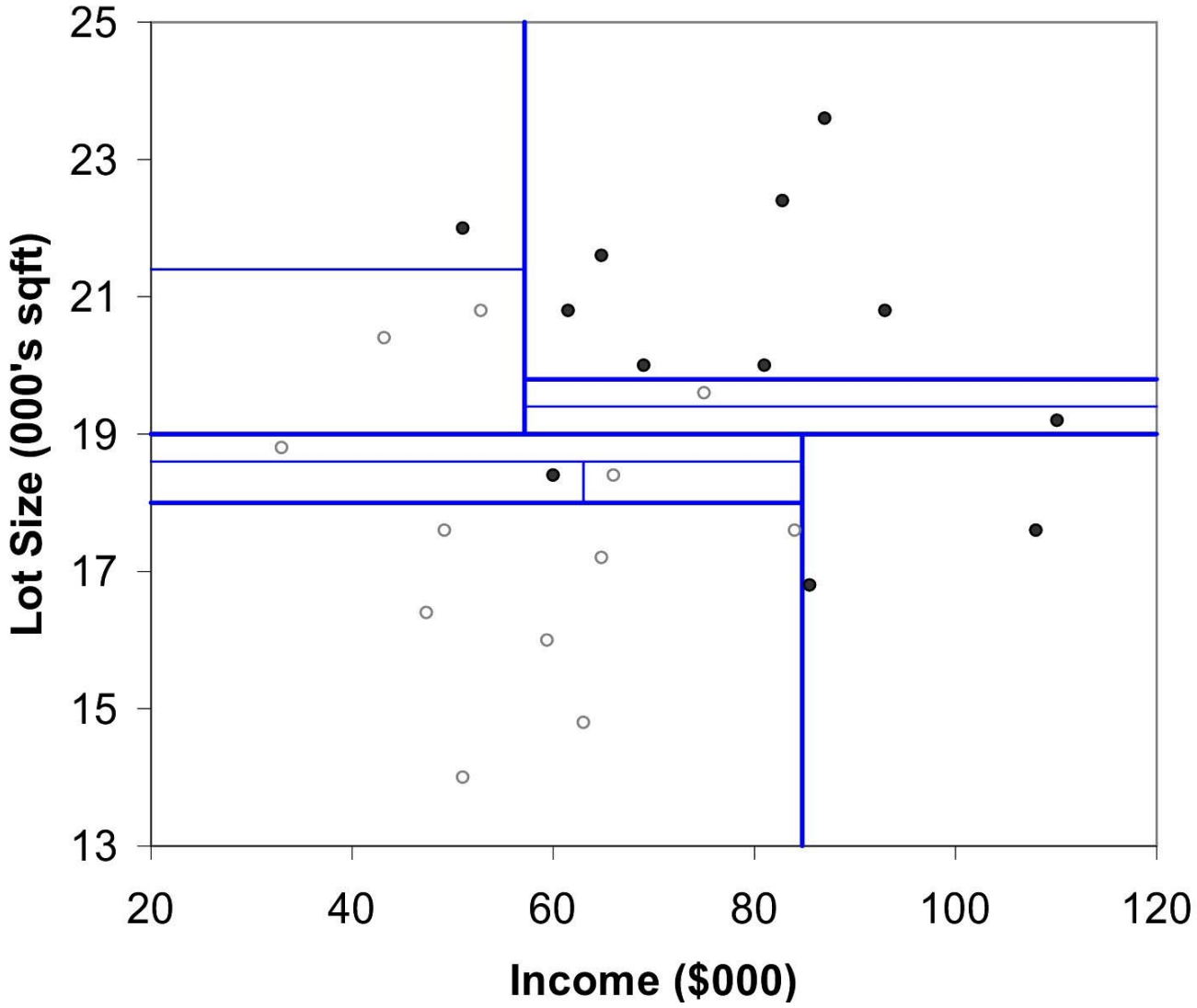
The first split: Lot Size = 19,000



Second Split: Income = $84,000



After All Splits



Measuring Impurity 衡量不纯度

Gini Index 基尼系数

Gini Index for rectangle A containing m records 在包含m个记录的矩形A中的基尼系数:

$$I(A) = 1 - \sum_{k=1}^{m} p_k^2$$

p = proportion of cases in rectangle A that belong to class k （ p_k 代表矩形A中属于类别k的百分比）

- I(A) = 0 when all cases belong to same class
- Max value when all classes are equally represented (= 0.50 in binary case)

Note: XLMiner uses a variant called “delta splitting rule”

Entropy 信息熵

$$entropy\,(A) = -\sum_{k=1}^{m} p_k \log_2(p_k)$$

p = proportion of cases (out of m) in rectangle A that belong to class k

- Entropy ranges between 0 (most pure) and $\log_2(m)$ (equal representation of classes)

Impurity and Recursive Partitioning
不纯度与递归划分

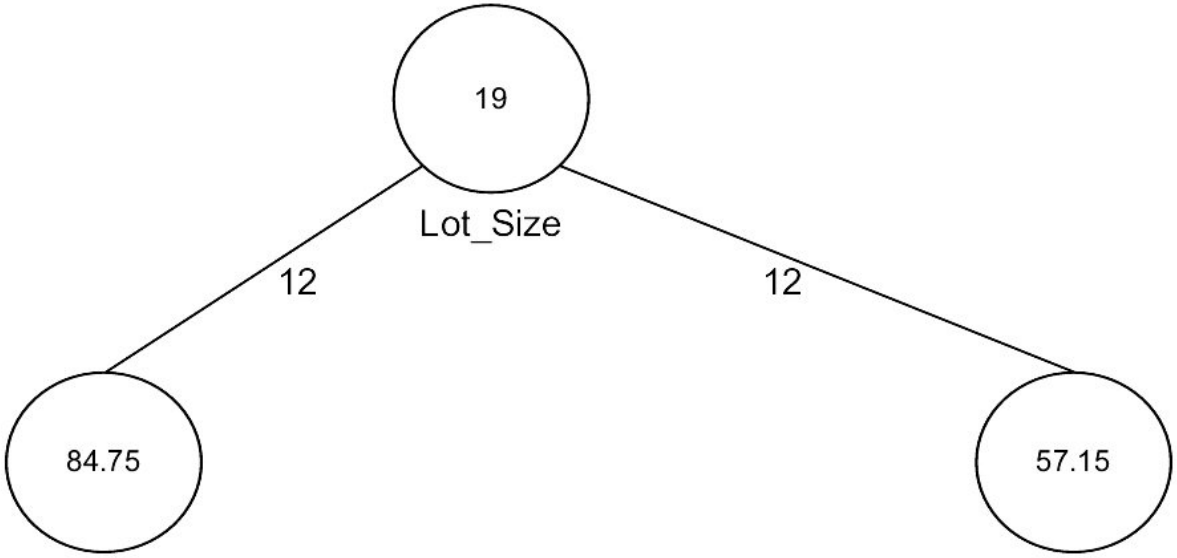
- Obtain overall impurity measure (weighted avg. of individual rectangles) 得到总的不纯度指标（每个矩形不纯度的加权平均值）

- At each successive stage, compare this measure across all possible splits in all variables 在每个后续阶段，比较所有可能的划分方式下的取值。

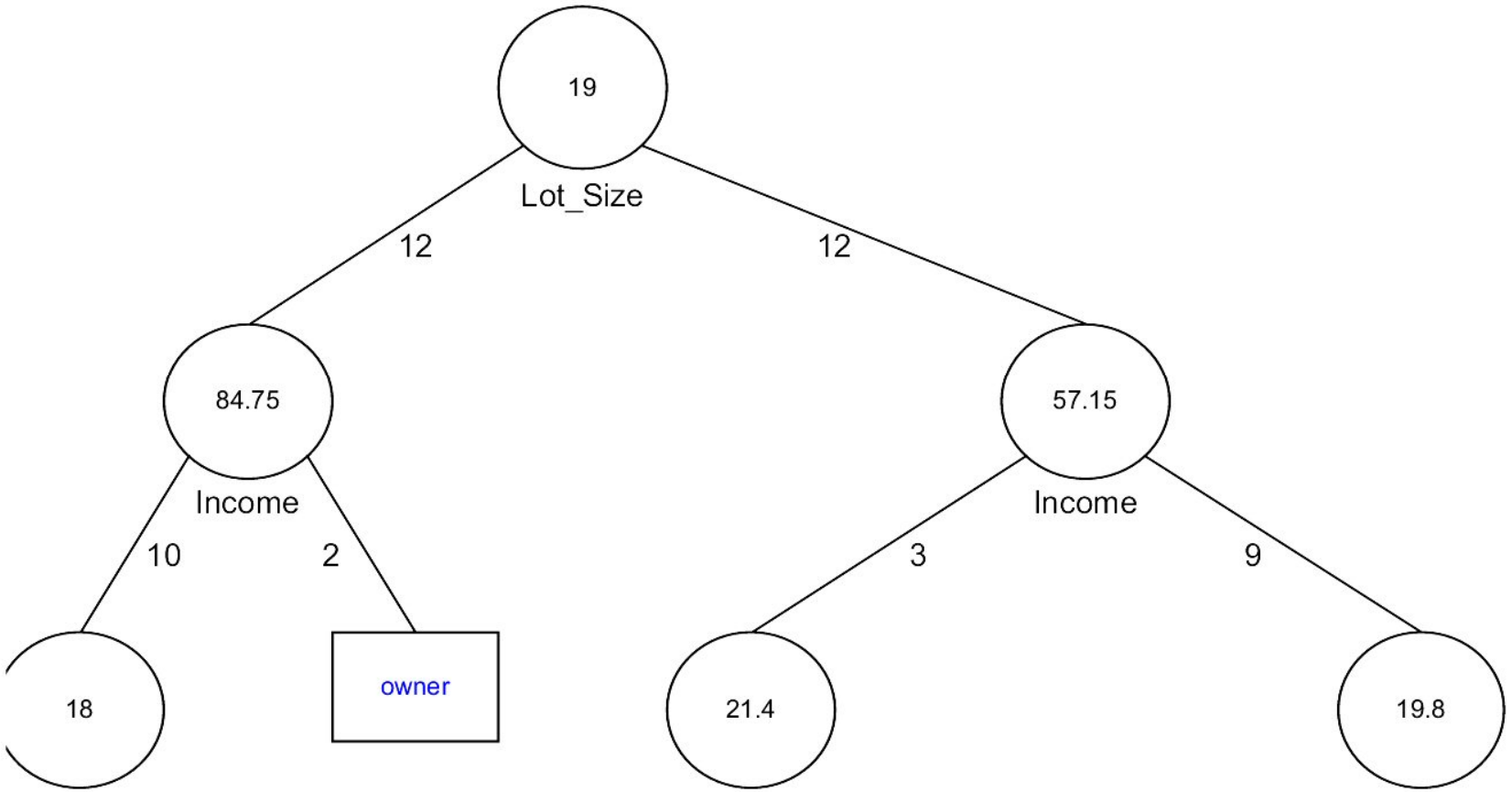
- Choose the split that reduces impurity the most 选择能够最大降低总的不纯度的划分方法。

- Chosen split points become nodes on the tree 划分出来的点成为树上的节点。

First Split – The Tree



Tree after three splits



Tree Structure 树的结构

- Split points become nodes on tree (circles with split value in center) 划出的点成为树上的节点（中间有切割值的圆圈）

- Rectangles represent “leaves” (terminal points, no further splits, classification value noted) 矩形代表树叶（终点，没有进一步的划分，注明了分类取值）

- Numbers on lines between nodes indicate # cases 线条旁边的数字代表记录数目。

- Read down tree to derive rule 从树的上面往下读可以得到规则：

  E.g., If lot size < 19, and if income > 84.75, then class = “owner”

Determining Leaf Node Label
决定树叶节点的标签

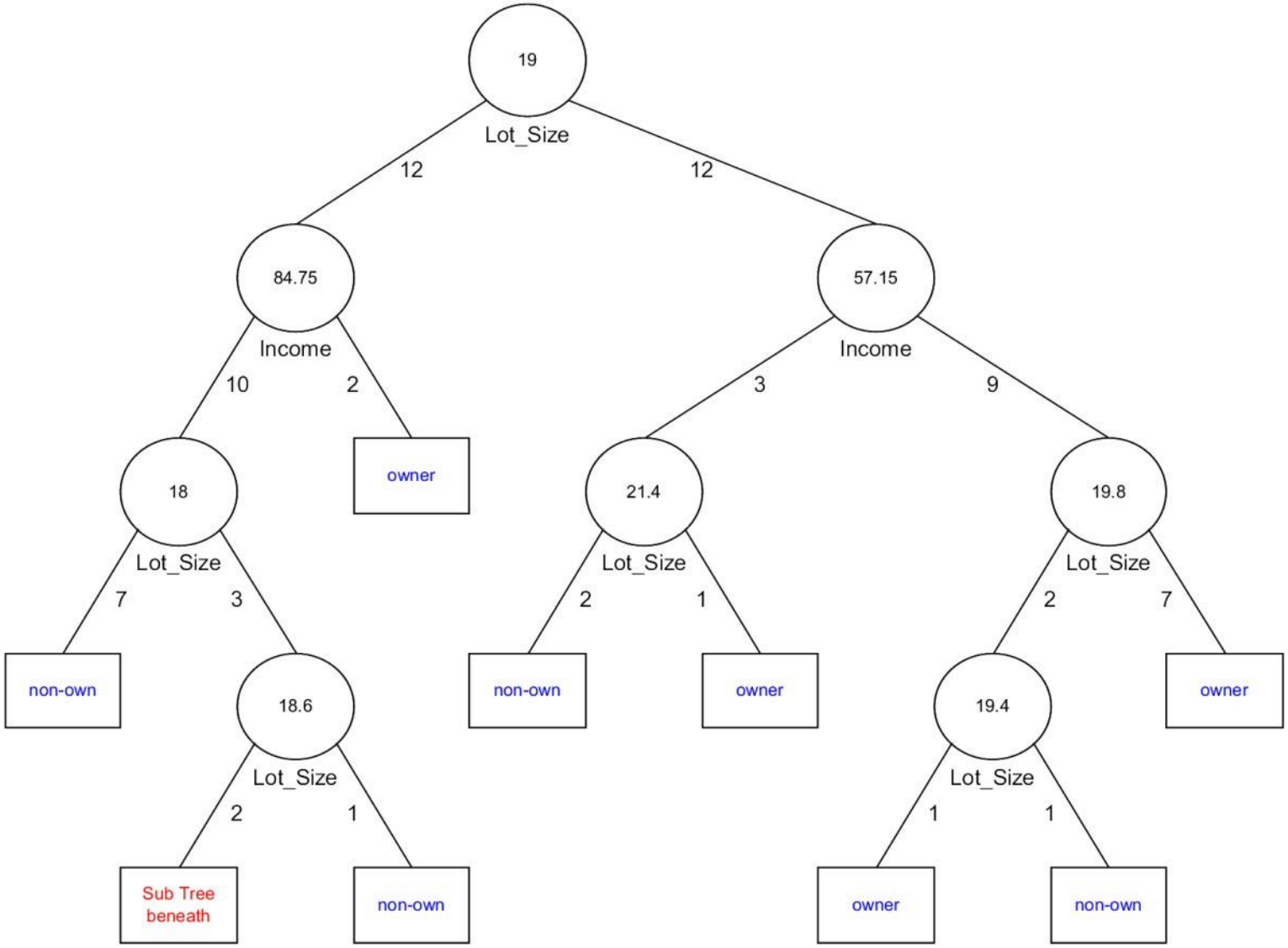
- Each leaf node label is determined by "voting" of the records within it, and by the cutoff value 叶子节点的标签由叶子内的记录投票并按截值决定。

- Records within each leaf node are from the training data 叶子节点中的记录从训练数据集而来。

- Default cutoff=0.5 means that the leaf node's label is the majority class. 叶子节点标签的截值默认值是0.5，意味着投票是简单多数原则。

- Cutoff = 0.75: requires majority of 75% or more "1" records in the leaf to label it a "1" node 如果截值是0.75则需要75%的多数票决定一个叶子的标签。

Tree after all splits



The Overfitting Problem
过度拟合问题

Stopping Tree Growth
停止树的生长

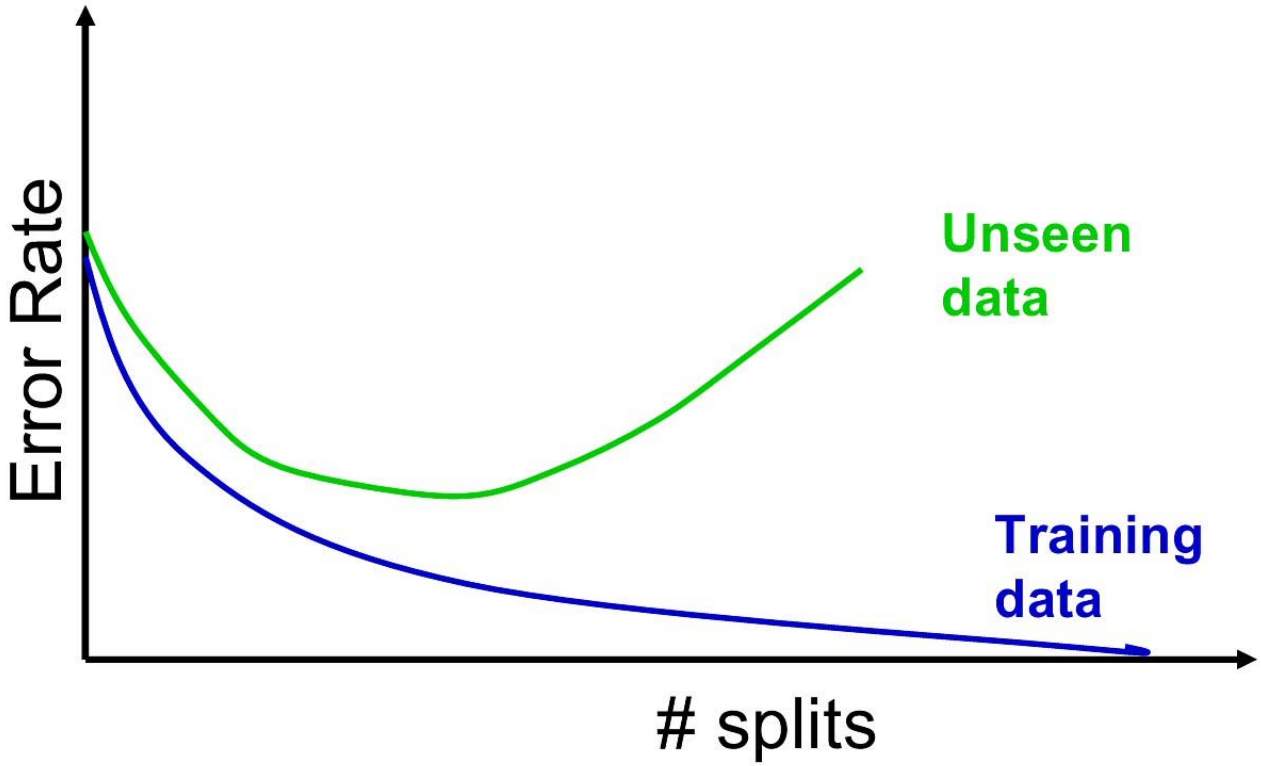
- Natural end of process is 100% purity in each leaf
通常在叶子纯度达到100%时划分过程停止。

- This overfits the data, which end up fitting noise in the data 这使数据过度拟合，导致最终结果是数据中的噪声也被拟合。

- Overfitting leads to low predictive accuracy of new data 过度拟合导致在新数据上预测准确度低。

- Past a certain point, the error rate for the validation data starts to increase 划分次数超过一定数值后，验证数据集的错误率开始增加。

Full Tree Error Rate



CHAID

CHAID, older than CART, uses chi-square statistical test to limit tree growth 比CART更老的CHAID使用chi-square统计检验来限制树的生长。

Splitting stops when purity improvement is not statistically significant 当纯度的增长在统计上不显著时，停止划分过程。

Pruning 修剪枝叶

- CART lets tree grow to full extent, then prunes it back CART让树长到最茂盛的程度然后修剪它的枝叶。

- Idea is to find that point at which the validation error begins to rise 思路是找到使验证数据集的错误率开始上升的枝叶数目。

- Generate successively smaller trees by pruning leaves 逐步通过修剪枝叶生成较小的树。

- At each pruning stage, multiple trees are possible 在每个步骤存在多种可能的树。

- Use *cost complexity* to choose the best tree at that stage 使用*成本加复杂度*来选择每个步骤最好的树。

Cost Complexity 成本加复杂度

$$CC(T) = Err(T) + \alpha L(T)$$

CC(T) = cost complexity of a tree 一棵树的CC

Err(T) = proportion of misclassified records 被错误分类的记录的百分率

a = penalty factor attached to tree size (set by user)
用户设置的针对树大小的惩罚因子

Among trees of given size, choose the one with lowest CC 给定树的大小，选择一棵使CC达到最小的树。

- Do this for each size of tree 为每个给定大小的树做这件事情。

Using Validation Error to Prune
使用验证错误率来修剪

Pruning process yields a set of trees of different sizes and associated error rates 修剪过程得到一系列大小不同的树和相应的错误率。

Two trees of interest: 两种引起兴趣的树：

- Minimum error tree 最小错误率决策树
  - Has lowest error rate on validation data 在验证数据集中有最低的错误率
- Best pruned tree 最佳修剪决策树
  - Smallest tree within one std. error of min. error 最小的决策树，其错误率在最小错误率的一个标准误差内。
  - This adds a bonus for simplicity/parsimony 这在错误率基础上加上了对简单性或者简约的奖励。

Error rates on pruned trees

# Decision Nodes	% Error Training	% Error Validation	
41	0	2.133333	
40	0.04	2.2	
39	0.08	2.2	
38	0.12	2.2	
37	0.16	2.066667	
36	0.2	2.066667	
35	0.2	2.066667	
14	1.16	1.533333	
13	1.16	1.6	
12	1.2	1.6	
11	1.2	1.466667	<-- Min. Err. Tree
10	1.6	1.666667	
9	2.2	1.666667	
8	2.2	1.866667	
7	2.24	1.866667	
6	2.24	1.6	<-- Best Pruned Tree
5	4.44	1.8	
4	5.08	2.333333	
3	5.24	3.466667	

Regression Trees 回归树

Regression Trees for Prediction
回归树用于预测

- Used with continuous outcome variable 用于连续型结果变量

- Procedure similar to classification tree 计算过程与分类树类似。

- Many splits attempted, choose the one that minimizes impurity 尝试多个划分方案，找到一个方案使不纯性最小。

Differences from CT 与分类树的不同

- Prediction is computed as the average of numerical target variable in the rectangle (in CT it is majority vote) 预测值是矩形内记录的目标变量的平均值（在分类树中是多数表决结果）

- Impurity measured by sum of squared deviations from leaf mean 不纯性以距离叶节点均值的离差平方和作为评判标准。

- Performance measured by RMSE (root mean squared error) 预测表现用RMSE（均方根差）衡量。

Advantages of trees 树的优点

- Easy to use, understand 易于使用和理解
- Produce rules that are easy to interpret & implement 产生的规则容易解释和应用
- Variable selection & reduction is automatic 变量选择和排除过程自动完成
- Do not require the assumptions of statistical models 不需要统计模型的假设
- Can work without extensive handling of missing data 即使未能广泛处理缺失数据，我们也可以使用树

Disadvantages 树的缺点

- May not perform well where there is structure in the data that is not well captured by horizontal or vertical splits 如果数据结构不能很好地用横切或者纵切捕捉到，那么树的表现是不好的。

- Since the process deals with one variable at a time, no way to capture interactions between variables 计算过程在每个步骤只处理一个预测变量，因此不能捕捉到预测变量之间的交互关系。

Summary 总结

- Classification and Regression Trees are an easily understandable and transparent method for predicting or classifying new records 分类树和回归树是易于理解且明晰的方法，可对新的数据记录进行分类或者预测。
- A tree is a graphical representation of a set of rules 一个树可以在图形上使用一系列规则表示出来。
- Trees must be pruned to avoid over-fitting of the training data　树必须进行修剪以防止对训练数据集的过度拟合。
- As trees do not make any assumptions about the data structure, they usually require large samples 因为树未对数据结构做任何假设，因此它们通常需要大样本进行训练。