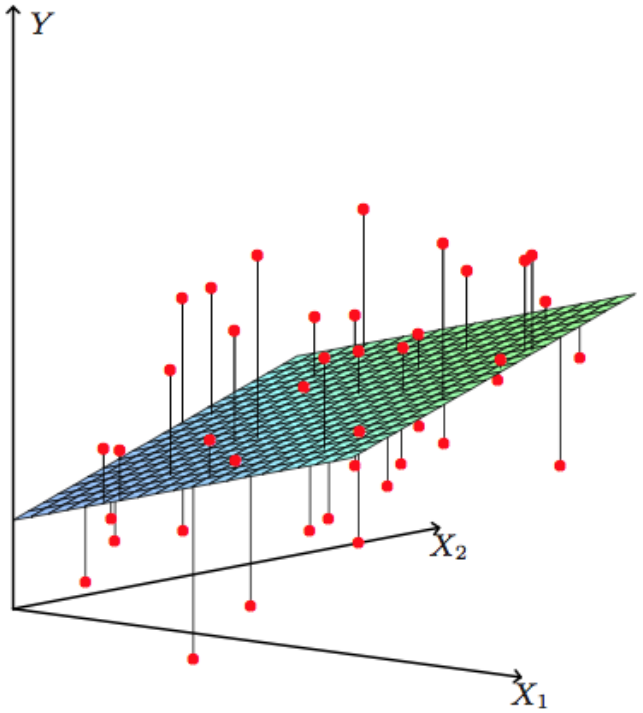
Statistical Machine Learning

Kuangnan Fang

Department of Statistics and Data Science, Xiamen University
Email: xmufkn@xmu.edu.cn



什么是决策树？

决策树本质上是一种非参数方法，因为我们不需要事先对总体的分布作任何假设。用于分类时，称呼它为分类树；用于回归时，称呼它为回归树。

例9-1： biopsy数据集：乳腺癌诊断（分类问题）。

	ID	V1	V2	V3	V4	V5	V6	V7	V8	V9	class
1	1000025	5	1	1	1	2	1	3	1	1	benign
2	1002945	5	4	4	5	7	10	3	2	1	benign
3	1015425	3	1	1	1	2	2	3	1	1	benign
...	...	...	...	...	...	...	...	...	...	...	...

什么是决策树？

例9-2： Carseats数据集：座椅销量预测（回归问题）。

	Sales	Comp-Price	Income	Advertising	Population	Price	...
1	9.5	138	73	11	276	120	...
2	11.22	111	48	16	260	83	...
3	10.06	113	35	10	269	80	...
...	...	...	...	...	...	...	...

决策树模型

决策树模型是用来描述对样本进行划分的树形结构，由节点和有向边组成。决策树包含三种节点，分别是根节点、中间节点或内节点、叶节点或终节点。

- 建模之初，全部样本组成的节点，没有入边，只有出边，称为根节点
- 不再继续分裂的节点称为树的终端节点或叶节点，没有出边，只有入边，它的个数决定了决策树的规模（size）和复杂程度
- 根节点和叶节点之外的都称作中间节点或内节点，既有入边，又有出边

决策树模型

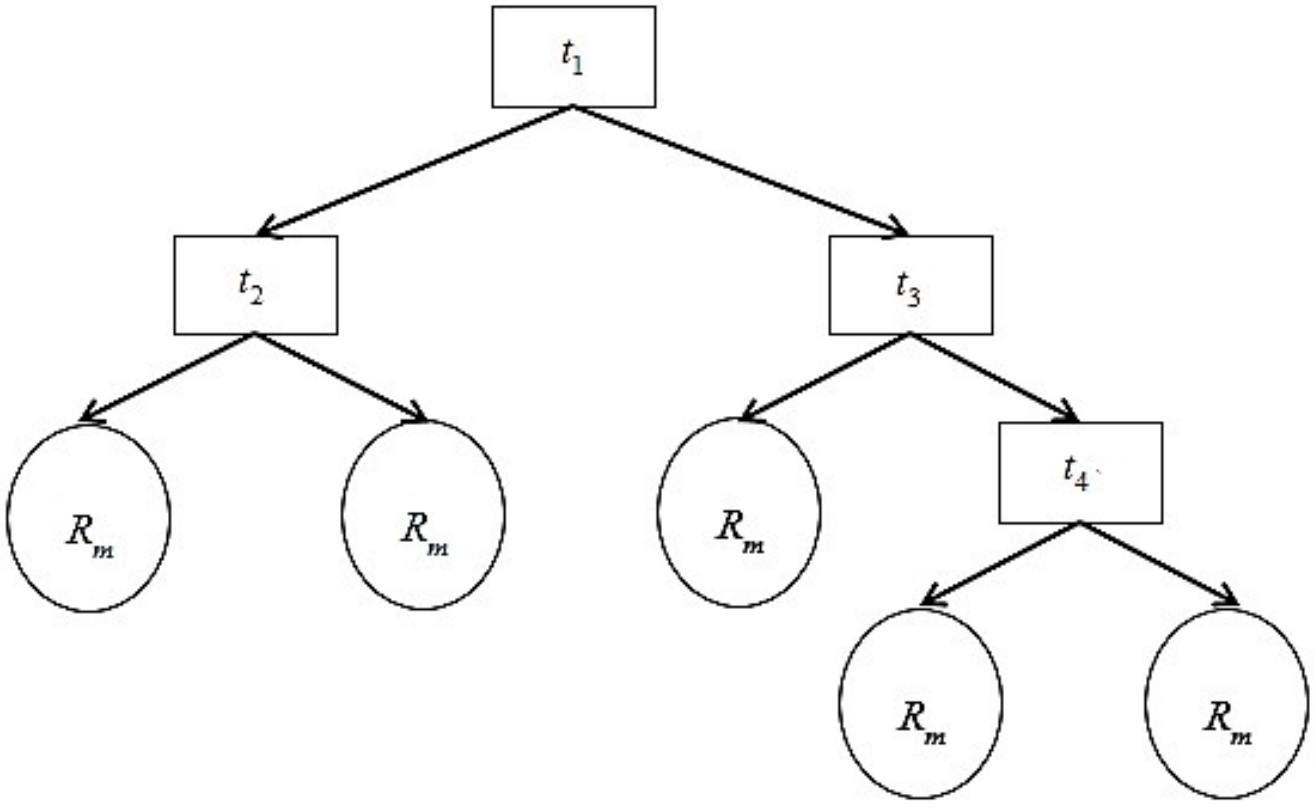


Figure 1: 决策树分割过程

决策树与if-then规则

决策树本质上是一系列if-then规则的集合，具有“互斥且完备”的性质。从根节点到叶节点的路径就是一条决策规则，路径上的内节点特征划分对应着规则的条件，叶节点的取值对应着规则的结论。

以Carseats数据为例：

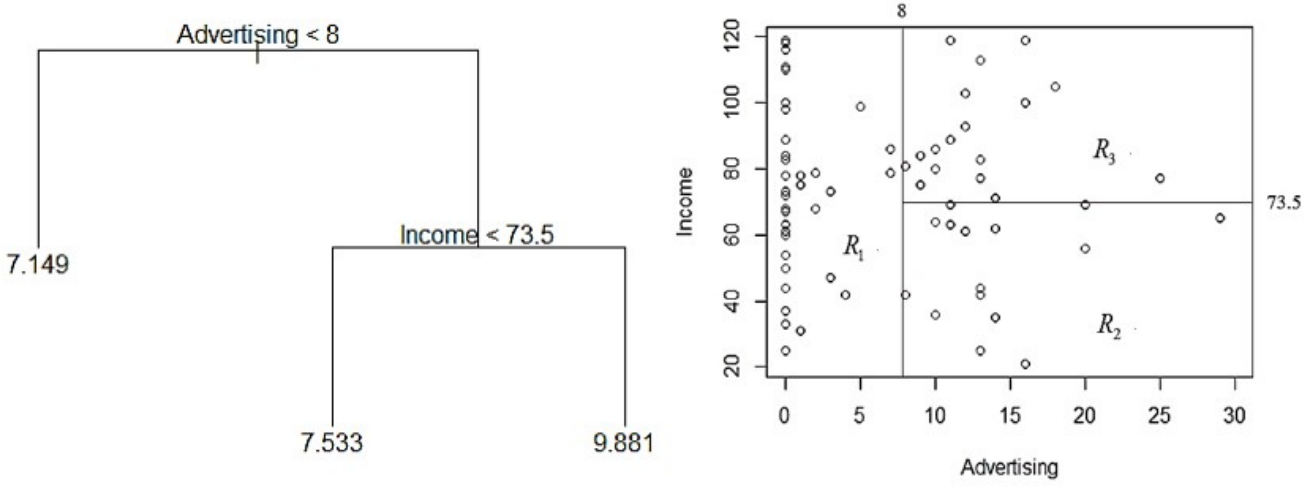
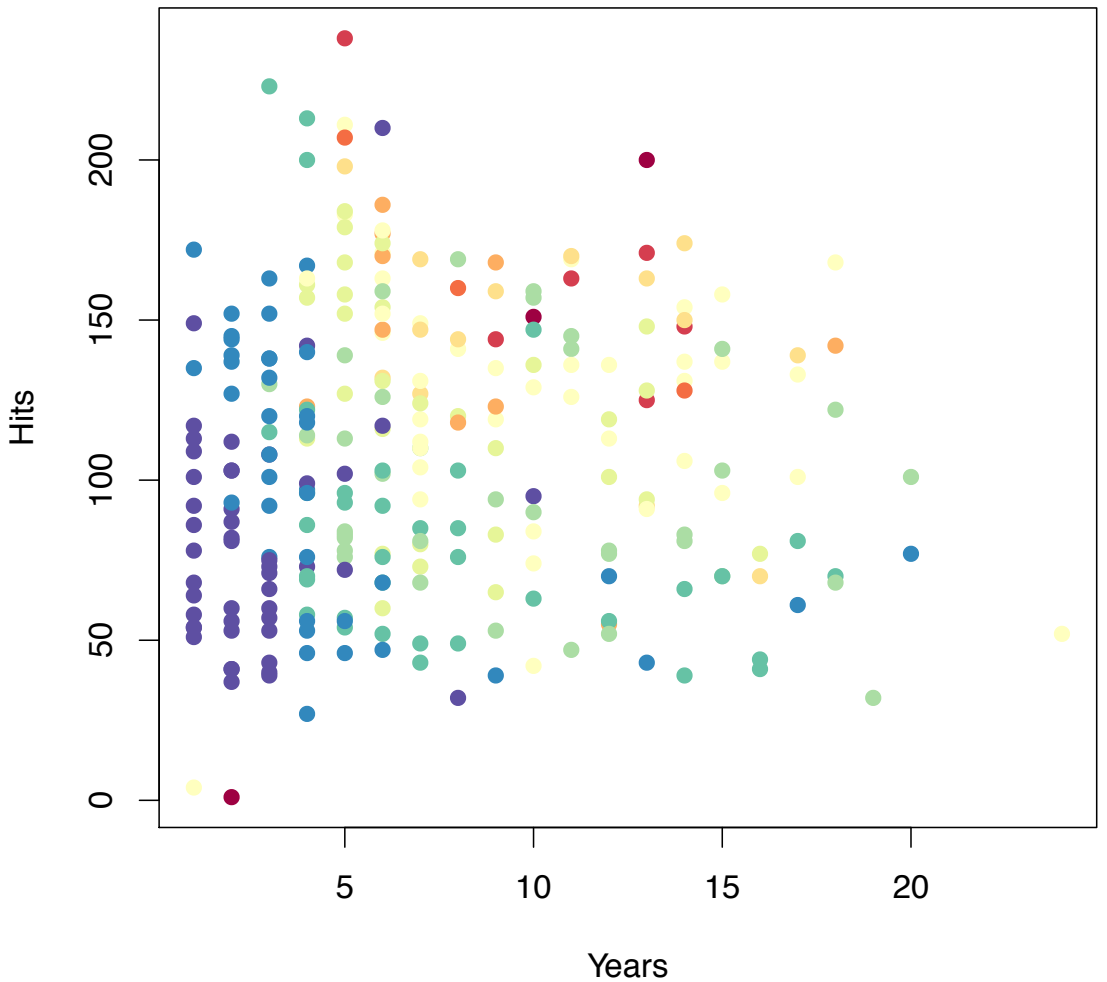


Figure 2: 左：回归决策树　右：根据左边的回归树划分特征空间

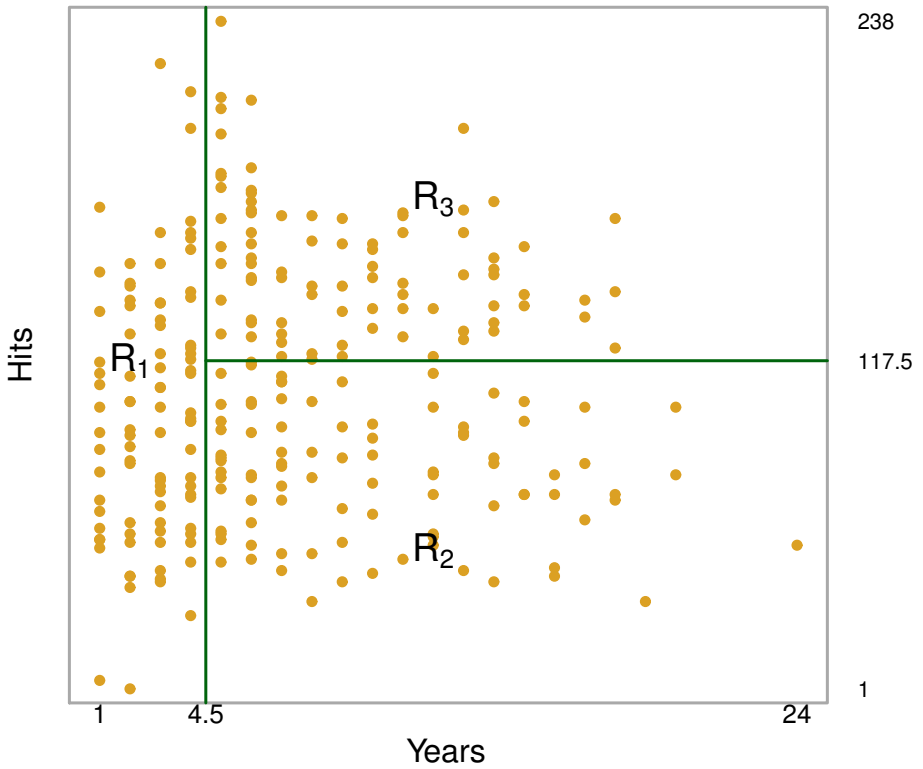
Baseball salary data: how would you stratify it?

Salary is color-coded from low (blue, green) to high (yellow,red)

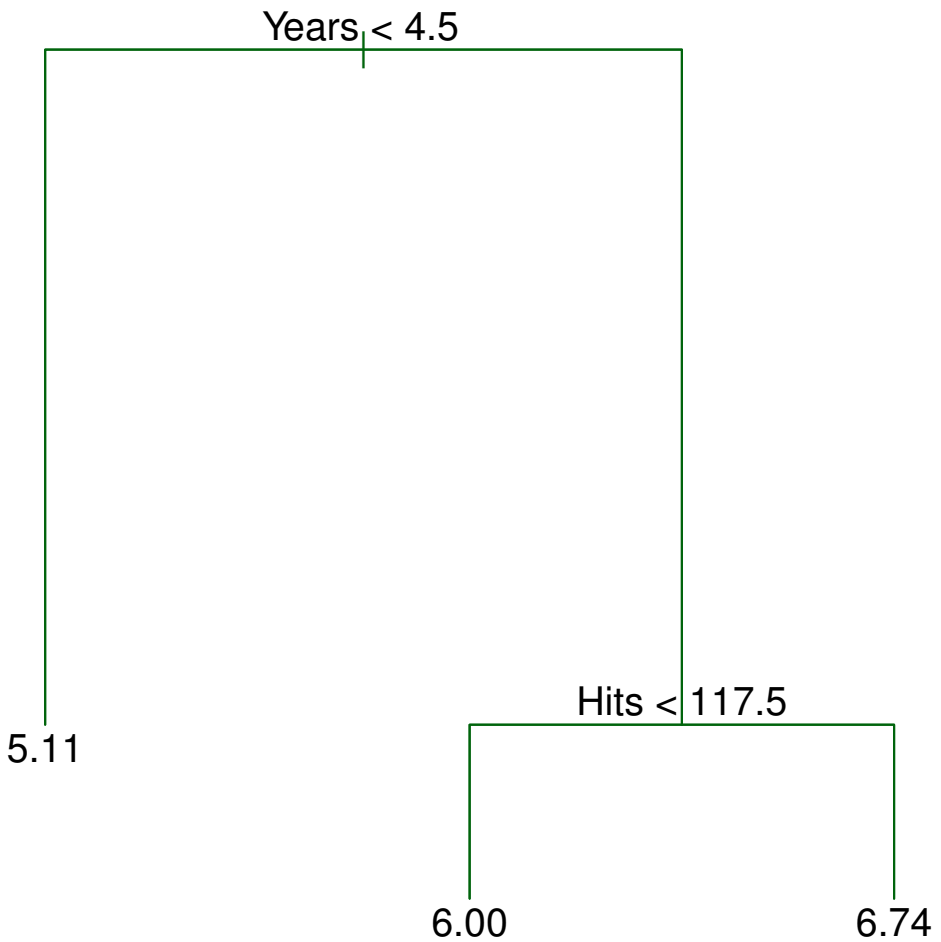


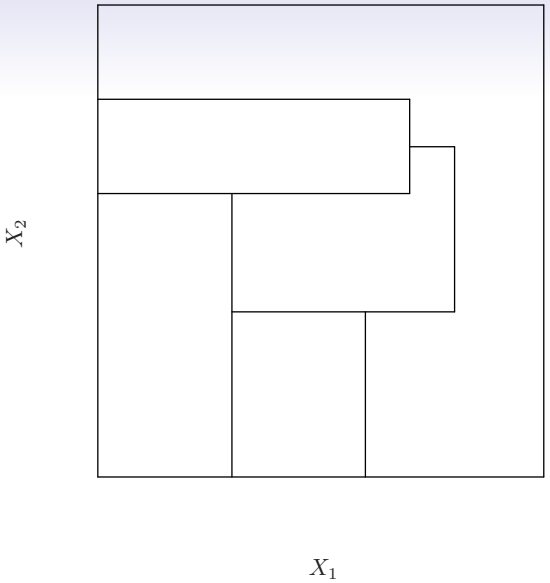
Results

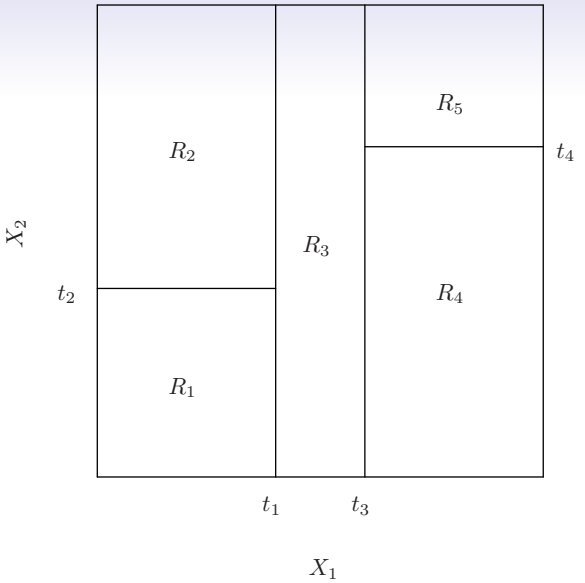
- Overall, the tree stratifies or segments the players into three regions of predictor space: $R_1 = \{\text{X} \mid \texttt{Years} < 4.5\}$, $R_2 = \{\text{X} \mid \texttt{Years} >= 4.5, \texttt{Hits} < 117.5\}$, and $R_3 = \{\text{X} \mid \texttt{Years} >= 4.5, \texttt{Hits} >= 117.5\}$.

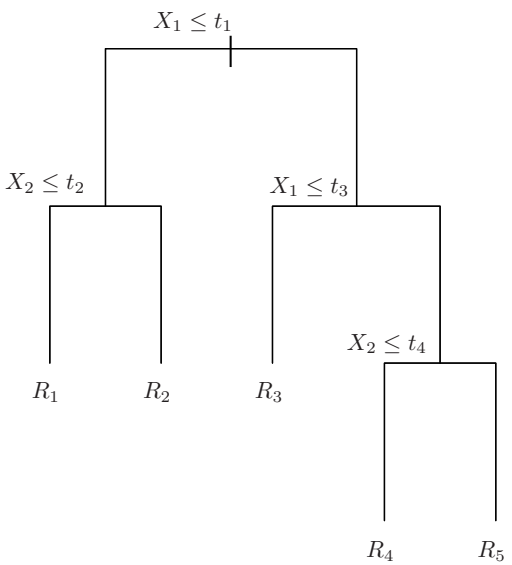


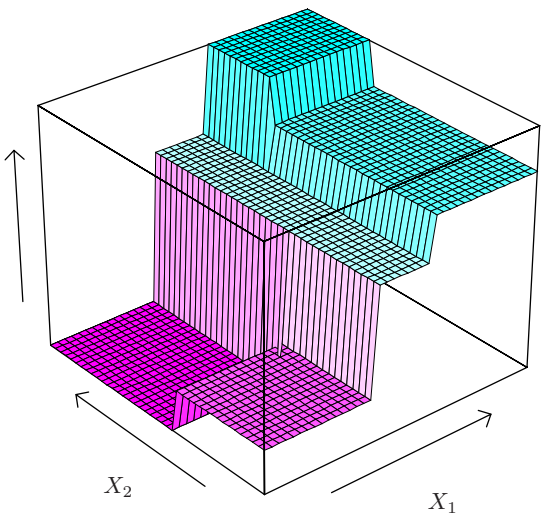
Decision tree for these data




X_2
X_1


R_1
R_2
R_3
R_4
R_5
t_1
t_2
t_3
t_4
X_2
X_1


$X_1 \leq t_1$
$X_2 \leq t_2$
$X_1 \leq t_3$
$X_2 \leq t_4$
R_1
R_2
R_3
R_4
R_5


X_2
X_1

决策树的学习

给定训练集:

$$D = ((x_1, y_1), (x_2, y_2)..., (x_n, y_n))$$

其中，$x_i = (x_{i1}, x_{i2}..., x_{ip})^T$ 表示第i 个样本，有p 个特征，y_i 可以是连续变量（回归树），也可以是离散变量（分类树）。

- 决策树的学习目标：根据给定的训练集构建一个决策树模型，希望对训练集和测试集能有较好的预测效果
- 决策树的构建过程：分裂特征选择、决策树生成和修剪枝过程（Pruning Process）

分裂特征选择

决策树分裂需要选择最优特征，在最优分裂点进行进行分裂。

- 如果对一个特征的分裂能大大提升决策树的预测效果，则说明该特征对因变量具有较好预测能力
- 如果一个特征的分裂不能提升决策树的预测效果，则称该特征对因变量不具有预测能力

如何选择最优的特征进行分裂呢？

信息增益

定义9-1（熵）:假设X是一个有限取值的离散随机变量，其概率质量函数为:$P(X=x_k)=p_k$，$k=1,2,...,K$，则随机变量X的熵为:

$$E_n(X)=-\sum_{k=1}^{K}p_k log p_k$$

- 对数以2 为底，熵的单位称作比特（bit）
- 对数以e 为底，熵的单位称作纳特（nat）
- $0\leq E_n(X)\leq logK$

信息增益

例9-3：当随机变量只取两个值，比如0,1，则X服从伯努利分布：

$$P(X=1)=p, P(x=0)=1-9$$

则X 的熵（以2 为底）为：

$$E_n(X)=-plog_2p-(1-p)log_2(1-p)$$

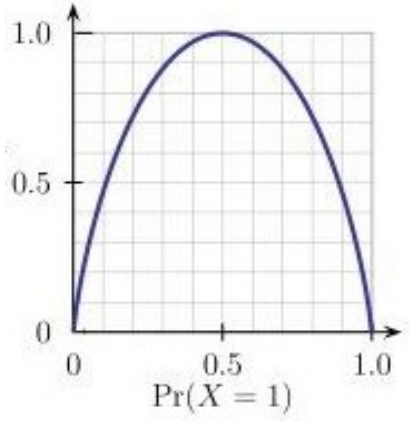


Figure 3: 伯努利分布时熵与概率关系

信息增益

定义9-2（条件熵（conditional entropy））： 假设随机向量(X,Y) 的联合概率分布为:

$$P(X=x_k,Y=y_j)=p_{kj},k=1,2,...,K,j=1,2,...,J$$

条件熵$E_n(Y|X)$ 为:

$$E_n(Y|X)=\sum_{k=1}^{K}p_kE_n(Y|X=x_k)$$

当熵和条件熵的概率由数据估计得到时，所对应的熵和条件熵分别称为经验熵（empirical entropy）和条件经验熵(empirical conditional entropy)。

信息增益

信息增益（information gain）是指已知X 信息而使Y 的信息的不确定减少的程度。

定义9-3（信息增益）： 特征X_j 对训练集D 的信息增益可以记为$g(D, X_j)$，其为D 的经验熵$E_n(D)$ 与X_j 给定下D 的经验条件熵$E_n(D|X_j)$ 之差，即:

$$g(D, X_j) = E_n(D) - E_n(D|X_j)$$

算法9-1　信息增益算法

输入：训练集D 和特征X_j

输出：信息增益$g(D, X_j)$

（1）计算训练集D 的经验熵$E_n(D)$

$$E_n(D) = -\sum_{k=1}^{K} \frac{n_k}{n} log_2 \frac{n_k}{n}$$

（2）计算特征X_j 对训练集D 的经验条件熵$E_n(D)$

$$\begin{aligned} E_n(D|X_j) &= \sum_{q=1} Q \frac{x_q}{n} E_n(D_q) \\ &= -\sum_{q=1}^{Q} \frac{n_q}{n} \sum_{k=1}^{K} \frac{n_{qk}}{n_q} log_2 \frac{n_{qk}}{n_q} \end{aligned}$$

（3）计算特征X_j的信息增益

$$g(D, X_j) = E_n(D) - E_n(D|X_j)$$

例9-4（消费贷数据）：某互联网金融公司的消费贷数据为：

ID	学历	是否有房	是否已婚	收入	是否违约
1	高中及以下	No	No	低	Yes
2	高中及以下	No	No	中等	Yes
3	高中及以下	Yes	No	中等	No
4	高中及以下	Yes	Yes	低	Yes
5	高中及以下	No	No	低	Yes
6	大学本科	No	No	低	Yes
7	大学本科	No	No	中等	Yes
8	大学本科	Yes	Yes	中等	No
9	大学本科	No	Yes	高	No
10	大学本科	No	Yes	高	No
11	研究生	No	Yes	高	No
12	研究生	No	Yes	中等	No
13	研究生	Yes	No	中等	No
14	研究生	Yes	No	高	No
15	研究生	No	No	低	Yes

因变量是违约情况，4个特征分别是借款人的学历，是否有房，是否已婚，收入情况。如何利用该训练集学习一个借款人信用预测决策树，可以对新客户的信用状况进行预测？

（1）$E_n(D) = 0.971$

（2）$E_{D,X_1} = 0.083$；$E_{D,X_2} = 0.324$；$E_{D,X_3} = 0.420$；$E_{D,X_4} = 0.363$

（3）比较各特征的信息增益，X_3（是否已婚）对训练集D的信息增益最大，所以在根节点处选择X_3作为最优特征分裂。

信息增益比率

以信息增益选择最优特征分裂，倾向于选择取值较多的特征。这样的决策树不具有泛化能力，为了解决这个问题，可以使用信息增益比率（information gain ratio）。

定义9.4（信息增益比率）：特征X_j 对训练集D 的信息增益比率$g_ratio(D, X_j)$ 为其信息增益$g(D)$ 与训练集D 关于特征X_j 的熵$E_{n_{X_j}}$ 的比值，即:

$$g\_ratio(D, X_j) = \frac{g(D, X_j)}{E_{n_{X_j}}(D)}$$

其中，$E_{n_{X_j}}(D) = -\sum_{q=1}^{Q} \frac{n_q}{n} log_2 \frac{n_q}{n}$。

连续特征

对于连续特征，常用的做法是将连续变量离散化，即将连续变量分成若干个区间，把区间看做是连续变量的取值，这样可以减少取值的数量。

二分法：给定训练集和某一连续特征X_j，X_j 在训练集D 上Q 个不同的取值，且有$X_j^1 < X_j^2 < ... < X_j^Q$，假如存在某一分裂点$t$ 将D 划分为两个子集D_t^-，D_t^+。D_t^- 包含特征$X_j < t$ 的样本，D_t^+ 包含特征$X_j \geq t$ 的样本。现在问题是分裂点t 怎么确定？可以按如下步骤：

（1）计算候选分裂点集合。计算每个取值的两两中位点集合：

$$M_j = \{\frac{X_j^q + X_j^{q+1}}{2} | 1 \leq q \leq Q-1\}$$

（2）计算候选分裂点集合M_j 中每个候选分裂点划分的不纯度的变化（比如信息增益或者信息增益比率），从中选择最优分裂点。

决策树的生成

决策树的生成算法有ID3、C4.5和CART等经典算法。

ID3算法：ID3算法的核心是在每个节点上利用信息增益准则选择最优的特征进行分裂，然后在每个子节点上又利用信息增益准则选择最优特征进行分裂，不断递归地调用以上方法，直到不能分裂为止。最终会得到一棵决策树。

算法9-2　ID3 算法

输入：训练集D 和特征集X

过程函数：TreeG(D,X)

输出：决策树T

（1）生成Tre 根节点T_{root}

（2）if D 中样本全属于同一类别C_k then

　　将T_{root} 标记为为C_k 类叶节点；return

　End if

（3）If $X=\varnothing$ OR D 中样本在X 上取值相同then

　　将T_{root} 标为叶节点，其类别标记为D 中样本最多的类；return

　End if

（4）按算法9-1 计算X 中特征对D 的信息增益，选择信息增益最大的特征X_j

　For X_j 的每一取值X_{jq} ; do

　　为T_{root} 生成一个分支；令D_q 为D 中在X_j 上取值为X_{jq} 的样本子集

　　If D_q 为空集then

　　　将分支节点标记为叶节点，其类别标记为D 中样本最多的类；return

　Else

　　　以$TreeG(D,X\{X_j)$ 为分支节点

　End if

　End for

例9-5：对例9-4中的数据，利用ID3算法构建决策树。

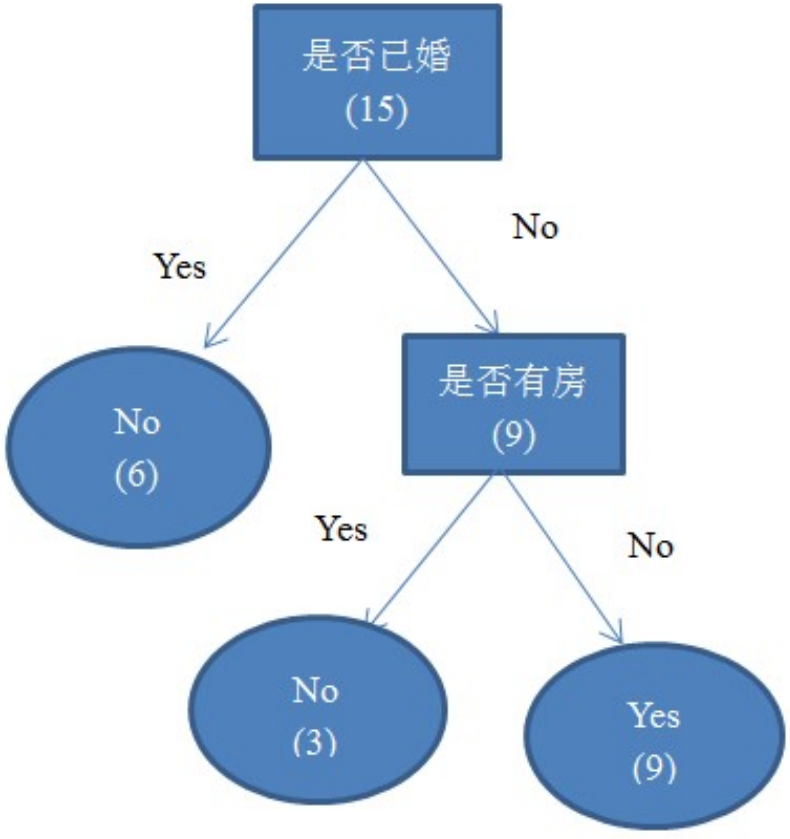


Figure 4: ID3算法决策树

决策树的生成

C4.5算法：C4.5算法与ID3算法类似，主要差异在于ID3算法的分裂特征选择是基于信息增益准则，而C4.5算法的分裂特征选择是基于信息增益比率准则。详细步骤可参考算法9-2。

CART 算法：CART算法基本思想是递归二叉分裂（recursive binary splitting）方法，在计算过程中充分利用二叉树，在一定的分割规则下将当前样本集分割为两个子样本集，使得生成的决策树的每个非叶节点都有两个分裂，这个过程又在子样本集上重复进行，直至无法再分成叶节点为止。

决策树的剪枝

决策树生成算法通过递归方式产生决策树，很有可能造成数据的过拟合，因此需要对决策树进行剪枝。

一般而言，剪枝方法有事先剪枝（pre-prune）和事后剪枝（post-prune）两种。

- 事先剪枝:在建树过程中要求每个节点的分裂使得不纯度下降超过一定阈值。
- 事后剪枝：代价复杂性剪枝（cost complexity pruning）是最常用的方法，先让树尽情生长，得到T_0 ，然后再在T_0 基础上进行修剪。

实际中常用事后剪枝!

代价复杂性剪枝

设$|T|$ 表示子树T 的叶节点数目，n_m 表示叶节点R_m 的样本量，则代价复杂性的剪枝法为：

$$C_\alpha(T) = \sum_{m=1}^{|T|} n_m Q_m(T) + \alpha|T|$$

其中，$Q_m(T)$ 为叶节点R_m 的不纯度；α 是调和参数，控制着模型对数据的拟合与模型的复杂度（树的大小）之间的平衡。

CART剪枝：一般采用事后剪枝法，常使用代价复杂性剪枝，过程如下:

（1）从决策树T_0 低端开始不断往上剪枝，直到根节点，得到一个子树（subtree）序列$\{T_0, ..., T_M\}$

（2）通过交叉验证对子树序列进行测试，选择最优子树。

分类树

当因变量为定性变量时，建立分类树模型。它的建模过程是通过基尼指数（Ginin index）选择最优分裂特征，以及根据该特征进行分裂的条件。

假设数据(x_i, y_i)，$(i = 1, 2, ..., n)$ 包含p 个t 特征和一个分类型的因变量$y \in \{1, ...K\}$，样本量为n，第k 类样本量为n_k。

定义9.4 基尼指数: 假设因变量分为K 类，样本点属于第k类的概率为p_k，则基尼指数定义为:

$$Gini(p) = \sum_{k=1}^{K} p_k(1 - p_k) = 1 - \sum_{k=1}^{K} p_k^2$$

对于给定的训练集D，其基尼指数为:

$$Gini(D) = \sum_{k=1}^{K} \hat{p_k}(1 - \hat{p_k}) = 1 - \sum_{k=1}^{K} \hat{p_k^2}, \quad \hat{p_k} = \frac{n_k}{n}$$

分类树

对于二分类的情况，若用p 表示节点m 包含其中一类的比例，则错分率、Gini指数和熵的取值分别为$1 - max(p, 1 - p)$ ，$2p(1 - p)$ ，$-plogp - (1 - p)log(1 - p)$。

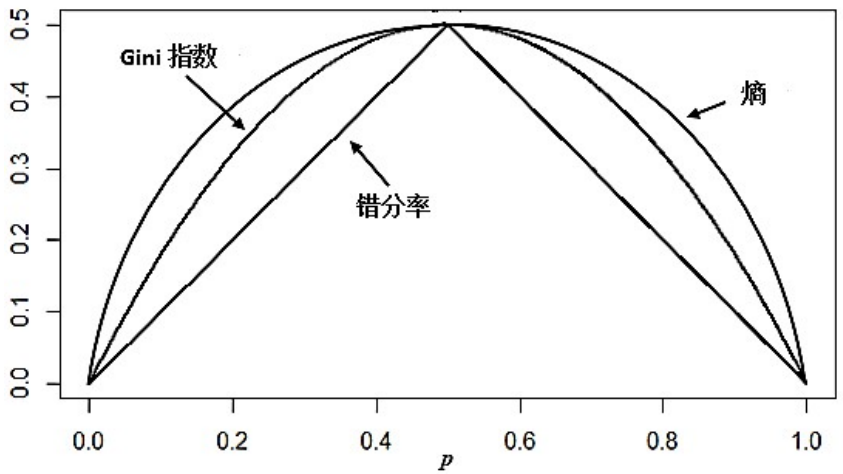


Figure 5: 衡量节点不纯度的三种指标

算法9-3：分类树算法

1. 采用递归二叉分裂法在训练集中生成一棵分类树。最优的分裂方案是使得M 个节点的不纯度减少达到最小，其中，衡量节点不纯度的指标有：

$$\text{错分率：}\frac{1}{n_m}\sum_{x_i\in R_m} I(y_i \neq k(m)) = 1 - p_{m,\hat{k}(m)}$$

$$\text{Gini指数：}\sum_{k\neq k'} \hat{p}_{m,k}\hat{p}_{m,k'} = \sum_{k=1}^{K} p_{\hat{m},k}(1-\hat{p}_{m,k})$$

$$\text{熵：}-\sum_{k=1}^{K} \hat{p}_{m,k} log \hat{p}_{m,k}$$

2. 剪枝。

1）对大树进行代价复杂性剪枝，得到一系列最优子树，子树是α 的函数；

2）利用K 折交叉验证选择α ，找出此α 对应的子树即可。

3. 预测。令$\hat{p}_{m,k} = \frac{1}{n_m}\sum_{x_i\in R_m} I(y_i = k)$ 表示在节点m 中第k 类样本点的比例，则预测节点m 的类别为：

$$k(m) = arg\, \max_k \hat{p}_{m,k}$$

即节点m 中类别最多的一类。

分类树

例9-5：考虑biopsy 数据集。该数据集包含699 个病人乳腺肿瘤切片的诊断信息，随机抽取500 个样本作为训练集，可以训练得到一棵含有10 个叶节点的分类树。

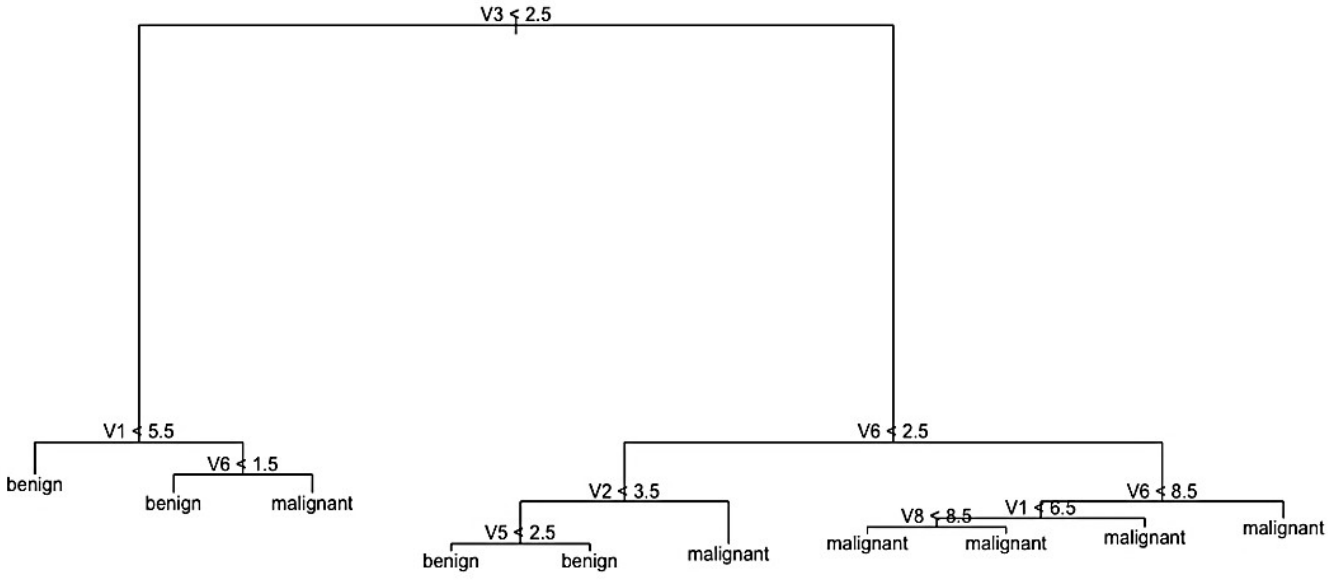


Figure 6: 未剪枝的树

分类树

对树剪枝，当叶节点个数为4 时，交叉验证误差达到最低。当$V3 < 2.5$，病人类别预测为benign；当$V3 \geq 2.5$ 且$V6 \geq 2.5$，预测为malignant；当$V3 \geq 2.5$ 且$V6 < 2.5$，$V2 < 3.5$，预测为benign，反之预测为malignant

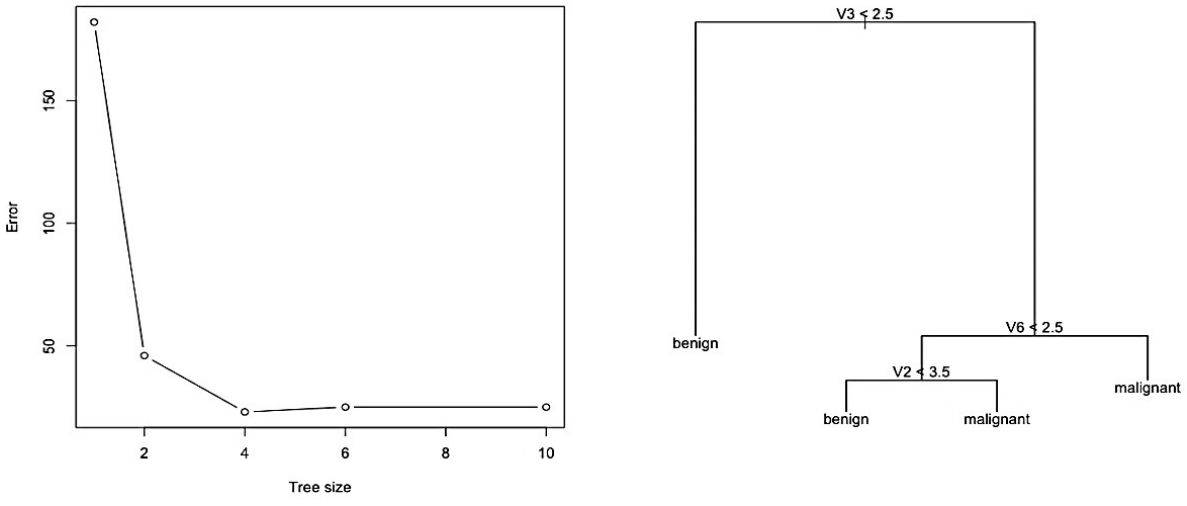


Figure 7: 根据交叉验证误差最小化剪枝的树

回归树

当因变量为连续型变量时，建立回归树模型。对于回归树，分裂准则一般可以用样本残差平方和。在预测上，将落在该叶节点的观测点的平均值作为该叶节点预测值。

假设数据(x_i, y_i)，$(i = 1, 2, ..., n)$ 包含p 个t 特征和一个连续型的因变量y ，样本量为n。假设把数据分成M 个区域（或称为节点），第m 个区域R_m 的样本量为n_m $(m = 1, 2, ..., M)$，且有一个固定的输出值c_m。回归树模型可表示为:

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

回归树

当特征空间的划分确定后，区域R_m 上的c_m 的最优估计值$\hat{c}_m$ 是R_m 上所有样本对应的因变量的均值，即:

$$\hat{c}_m = ave(y_i|c_i \in R_m) = \frac{1}{n_m}\sum_{i \in R_m} y_i$$

如何对特征空间进行划分？

启发式方法：选定第j 个变量x_j 和它的分裂点s ，并定义两个区域：

$$R_1(j,s) = \{x|x_j \leq s\}, R_2(j,s) = \{x|x_j > s\}$$

然后通过求解：

$$\underset{j,s}{min}[\underset{c_1}{min}\sum_{x_i \in R_1(j,s)}(y_i - c_1)^2 + \underset{c_2}{min}\sum_{x_i \in R_2(j,s)}(y_i - c_2)^2]$$

寻找最优分裂变量x_j 和最优分裂点s。

算法9-4：回归树算法

1. 采用递归二叉分裂法在训练集中生成一棵分类树。最优的分裂方案是使得M 个节点的不纯度减少达到最小，其中，衡量节点不纯度的指标Q_m 为样本残差平方和的平均：

$$Q_T(T) = \frac{1}{n_m} \sum_{x_i \in R_m} (y_i - y_{R_m})^2$$

（1）使用启发式方法选择最优分裂变量x_j 和最优分裂点s。

（2）用选定的(j, s) 划分区域并决定相应的输出值

$$R_1(j, s) = \{x | x_j \leq s\}; R_2(j, s) = \{x | x_j > s\}$$

$$\hat{c_m} = \frac{1}{n_m} \sum_{i \in R_m} y_i, m = 1, 2$$

（3）对子区域递归重复步骤（1）、（2），直到满足条件停止 （4）将特征空间划分成M 个区域，得到回归树

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

算法9-4：回归树算法

2. 剪枝

（1）对树进行代价复杂性剪枝，得到一些列最优子树，子树是a 的函数

（2）利用K 折交叉验证选择a，找出此a 对应的子树即可

3. 预测。区域划分好后，可以确定某一给定预测数据所属的区域，并用这一区域的训练集的平均响应值对其进行预测:

$$y_{R_m} = \frac{1}{n_m} \sum_{x_i \in R_m} y_i$$

回归树

例9-6: Carseats数据集是一个包含400个不同商店的儿童座椅销售情况的模拟数据集，随机抽取300个样本作为训练集，可以训练得到一棵含有15个叶节点的回归树

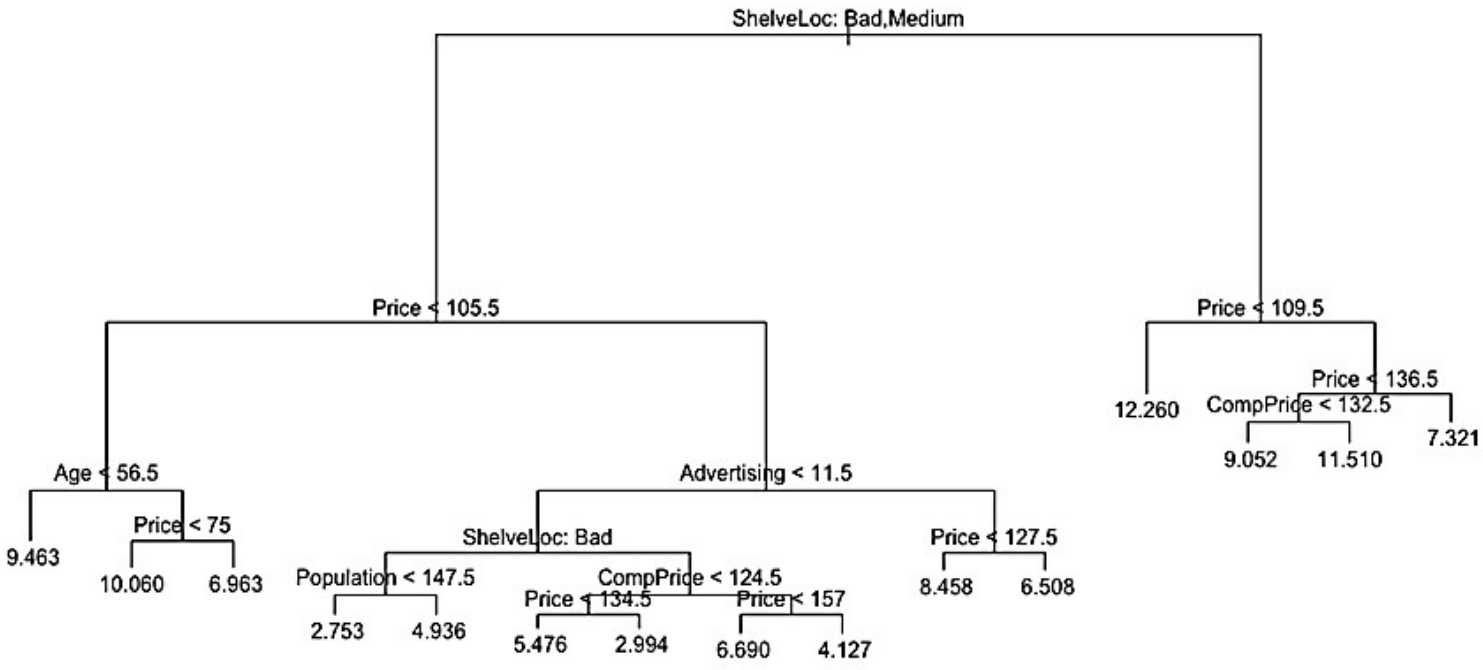


Figure 8: 未剪枝的树

回归树

同样根据交叉验证结果对其剪枝，最终得到一棵含有7个叶节点的树。

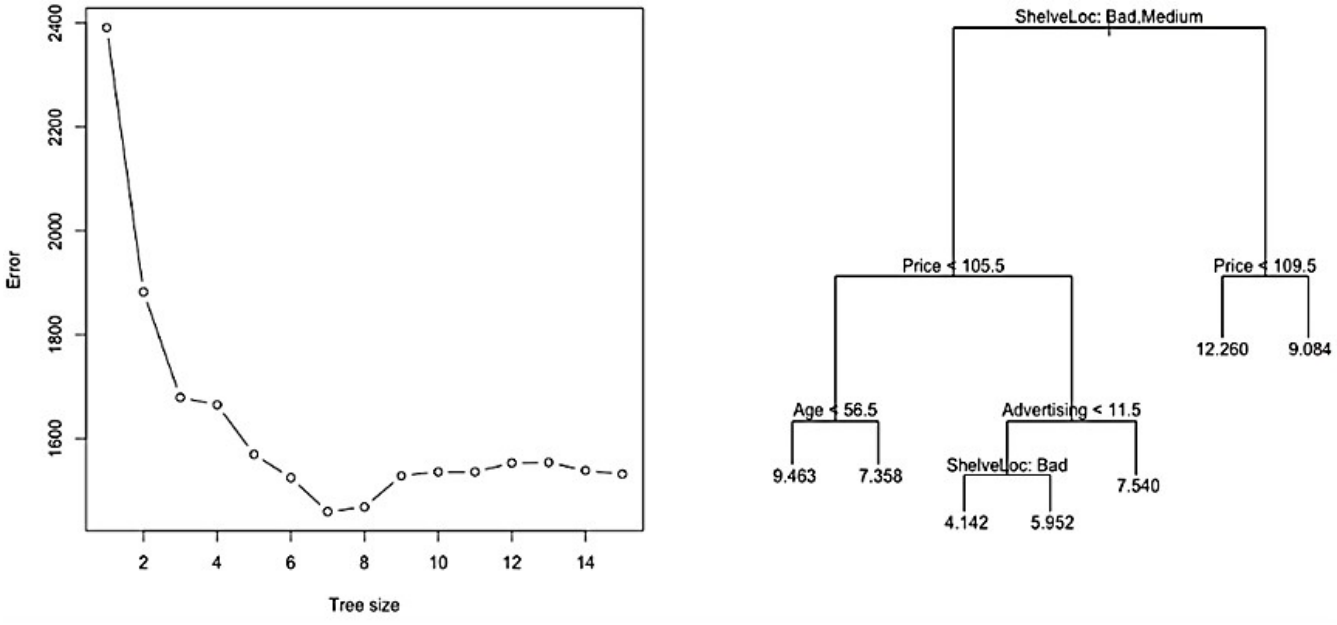


Figure 9: 根据交叉验证误差最小化剪枝的树

树的优缺点

树的优点：

- 易理解、解释性强
- 不需要任何先验假设
- 与传统的回归和分类方法相比，更接近人的大脑决策模式
- 用图形展示，可视化效果好，非专业人士也可轻松解释
- 可以直接处理定性的特征，不需像线性回归那样将定性变量转换成虚拟变量。

树的缺点：

- 决策树方差大、不稳定
- 数据很小的扰动可能得到完全不同的分裂结果，有可能是完全不同的决策树

python 案例分析

考虑数据heart.data（可以从R 语言的glmpath 包获取），该数据集描述了462 个南非人的身体健康状况指标，用来研究哪些因素对是否患有心脏病有影响。其中，因变量y 是一个二分类变量，代表是否患有冠心病，自变量共9 个，包括sbp（血压），tobacco（累计烟草量），ldl（低密度脂蛋白胆固醇），adiposity（肥胖），famhist（是否有心脏病家族史，定性变量），typea（型表现），obesity（过度肥胖），alcohol（当前饮酒），age （年龄）。

python 案例分析

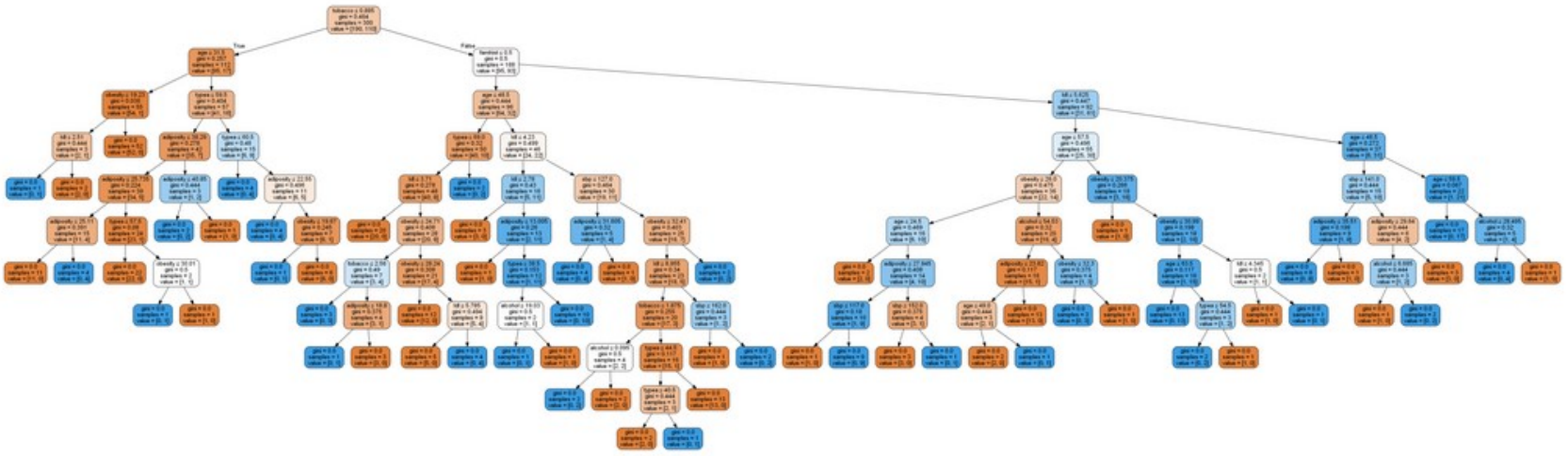


Figure 10: heart.data数据集的未剪枝的树

python 案例分析

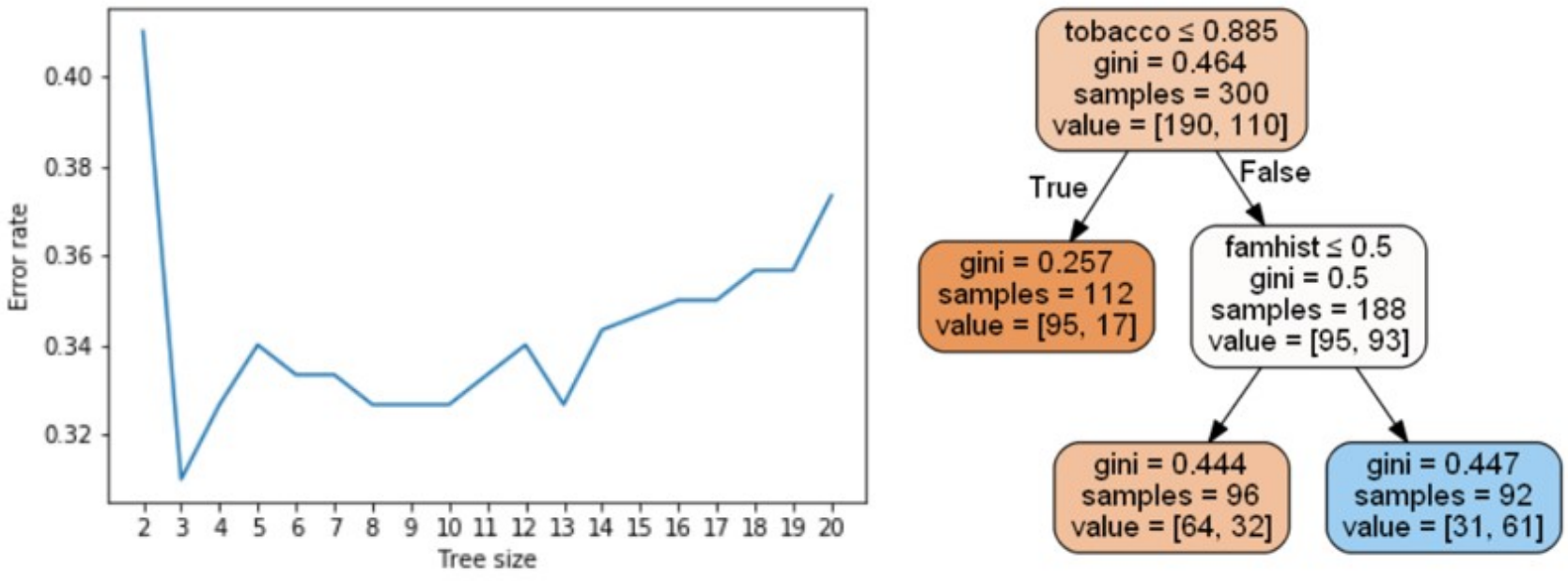


Figure 11: heart.data数据集的剪枝后的树