# 决策树算法

决策树算法是一种有监督、非参数、简单、高效的机器学习算法。相对于前面章节中介绍的无监督学习方法，决策树算法由于充分利用了响应变量的信息，所以能够很好地克服噪声的问题，在分类及预测方面效果更佳。决策树的决策边界为矩形，所以对于真实决策也为矩形的样本数据集有着很好的预测效果。此外，决策树算法以树形展示分类结果，在结果的展示方面比较直观，所以在实务中应用较为广泛。

# 目录

CONTENTS

PART 01

# 决策树算法的基本原理

# 决策树算法的基本原理

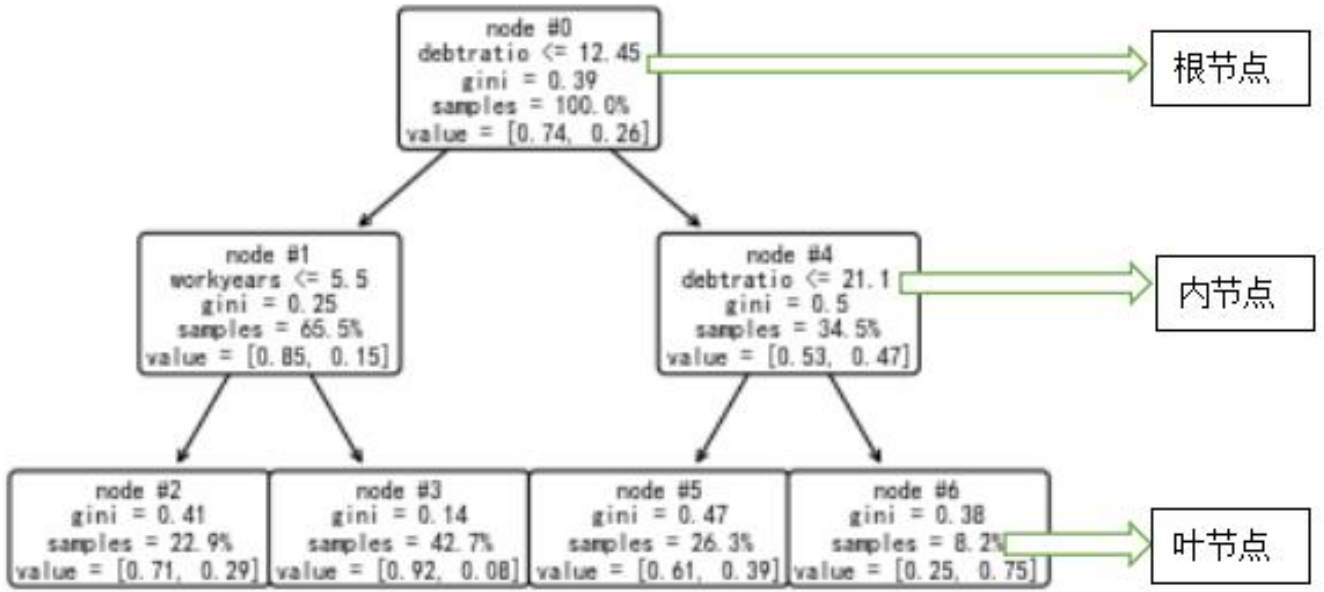
决策树算法借助于树的分支结构构建模型。

如果是用于分类问题，则为分类树；

如果是用于回归问题，则为回归树。

# 决策树算法的基本原理

该例子为通过决策树判断个人信用卡客户是否出现违约。在图13.1中，最上面的一个点是根节点，最下面的各个点是叶节点，其他的点都是内节点（本例中展示的决策树内节点只有一层，但实务中可能有很多层都属于内节点）。本例中根节点为0号（node #0），样本示例全集中未违约客户和违约客户的占比分别为0.74、0.26。




在样本示例全集中，如果信用卡客户的债务率debtratio <= 12.45，那么就会被分到1号节点，1号节点未违约客户和违约客户的占比分别为0.85、0.15；如果信用卡客户的债务率debtratio > 12.45，那么就会被分到4号节点，4号节点未违约客户和违约客户的占比分别为0.53、0.47。然后在4号节点中，如果信用卡客户的债务率debtratio <= 21.1，那么就会被分到5号节点，5号节点未违约客户和违约客户的占比分别为0.61、0.39；如果信用卡客户的债务率debtratio > 21.1，那么就会被分到6号节点，6号节点未违约客户和违约客户的占比分别为0.25、0.75，需要引起高度重视。

# 决策树算法的基本原理

如果是分类树，叶节点将类别占比最大的类别作为该叶节点的预测值；如果是回归树，叶节点将节点内所有样本响应变量实际值的平均值作为该叶节点的预测值。

从原理的角度，决策树本质上就是依次选取最为合适的特征向量，按照特征向量的具体取值，不断对特征空间进行矩形分割的过程，因为每一次切割都是直线，所以其决策边界为矩形。在分割空间时，决策树执行的是一种自上而下的贪心算法，即每次仅选择一个变量按照变量临界值进行分割，该变量及其临界值都是当前步骤下，能够实现局部最优的分割变量和分割临界值，并未从全盘考虑整体最优。

# 决策树算法的基本原理

一般来说，大部分机器学习都需要将特征变量标准化，以便让特征之间的比较可以在同一个量纲上进行。但是对于决策树算法而言，从数据构建过程来看，不纯度函数的计算和比较都是单特征的，所以决策树算法不需要对特征变量进行标准化处理。

综上所述，决策树的分类规则非常容易理解，准确率也比较高，尤其是针对实际决策边界为矩形的情形，而且不需要了解背景知识就可以进行分类，是一个非常有效的算法。因其简单、有效，也成为了与朴素贝叶斯估计方法并驾齐驱的两大流行机器学习算法。

# 特征变量选择及其临界值确定方法

决策树生长的过程，其实就是按照某一特征变量对响应变量进行分类，将样本示例全集不断按照响应变量分类进行分割的过程。那么应该按照什么的样的规则进行分割，或者说按照什么样的规则使树生长，才能取得最好的分类效果？一言以蔽之，就是要使得分裂生长后同一样本子集内的相似性程度（或称“纯度”）越高越好，或者说样本示例全集的“不纯度”通过切割样本的方式下降越多越好。这在本质上就是特征变量选择及其临界值确定的问题，常用的方法包括信息增益（Information gain）、增益比率（gain ratio）、基尼指数（Gini index），对应于ID3、C4.5和CART三种决策树算法。下面，我们逐一介绍下这几种方法。

# 一、信息熵

在介绍信息增益和增益比率之前，首先需要了解信息熵的概念。信息熵本质上是一种节点不纯度函数，用来衡量样本集的混乱程度，如果样本示例全集为 D，响应变量一共有 K 个类别，$p_k$是第 k 类样本的占比，则该样本集的信息熵为：

$$Ent(D) = -\sum_{k=1}^{K} p_k log_2 p_k$$

公式中的$log_2$表示以 2 为底的对数，分布越集中，样本集中的样本越属于同一类别，集合的混乱程度就越小，或者所集合的纯度越高；一个极端的情形是，所有样本只有 1 个类别（k=1），那么就不存在混乱的问题了（$p_k = 1$），样本集的信息熵也就为 0（$log_2 p_k = 0$）；同样的，分布越分散，，样本集的信息熵越大，说明集合的混乱程度越大，或者说集合的纯度越小。

# 二、信息增益（Information gain）

ID3 决策树算法基于信息增益（Information gain），在选择特征变量时的基本标准是，通过选择该特征向量对数据集进行划分，可以使得样本集的信息增益最大。设样本示例全集为 D，某特征变量为 a，样本集可以通过该特征变量将响应变量划分为 v 个类别，那么信息增益即为：

$$Gain(D,a) = Ent(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent(D^v)$$

其中的 Ent(D)是指样本示例全集的信息熵，Ent($D^v$)是指每个子类别的信息熵，$\frac{D^v}{D}$是指每个子样本集占样本示例全集的比例，信息增益的含义就是用样本示例全集的信息熵减去每个子类别信息熵的加权和。在样本示例全集信息熵 Ent(D)既定保持不变的前提下，我们需要做的就是要找到能使

$$\sum_{v=1}^{V} \frac{|D^v|}{|D|} \mathrm{Ent}(D^v)$$

最小的特征变量 a，从而使得信息增益最大。用更好理解的语言来解释，就是要通过特征变量的选择，在整体上使得类别内部的纯度“高高益善”。

# 二、信息增益（Information gain）

在信息增益算法下，决策树倾向于选择取值类别较多的特征变量，比如设样本示例全集容量为 n，那么在极端情形下，按照某特征变量划分，样本集可以通过该特征变量将响应变量划分为 n 个类别，即每个样本观测值都属于 1 类并构成 1 个子样本集，而每个子样本集因为只有 1 个样本，其子类别的信息熵 $\mathrm{Ent}(D^v)$肯定为 0，$\sum_{v=1}^{V}\frac{|D^v|}{|D|}\mathrm{Ent}(D^v)$也就等于 0，信息增益也就最大化了。

# 三、增益比率（gain ratio）

C4.5 决策树算法基于增益比率（gain ratio），在选择特征变量时的基本标准是，每次决策树分裂时，都选择增益比率（gain ratio）最大的特征进行划分。设样本示例全集为 D，某特征变量为 a，样本集可以通过该特征变量将响应变量划分为 v 个类别，那么增益比率即为：

$$Gain\_ratio(D,a) = \frac{Gain(D,a)}{H(a)}$$

，其中

$$H(a) = -\sum_{v=1}^{V} \frac{|D^v|}{|D|} \log_2 \frac{|D^v|}{|D|}$$

Gain(D,a)即为前面介绍的信息增益，而 H(a)则为样本示例全集 D 关于特征变量 a 的取值熵或固有值，在 H(a)中，V 的值越大，H(a)就会越大。所以 C4.5 决策树算法相当于对 ID3 进行了改进，在一定程度程度上解决了“决策树倾向于选择取值类别较多的特征变量”的问题。

# 四、基尼指数（Gini index）

CART 决策树算法基于基尼指数（Gini index）在选择特征变量时的基本标准是，每次决策树分裂时，都选择基尼指数（gain ratio）最小的特征进行划分。。基尼指数是指从样本集中随机抽取两个样本示例，这两个示例类别不一致的概率，是衡量样本示例全集纯度的另一种方式。如果样本示例全集为 D，响应变量一共有 K 个类别，$p_k$ 是第 k 类样本的占比，则该样本集的基尼指数为：

$$Gini(D) = \sum_{k=1}^{K} \sum_{k' \neq k} p_k p_{k'} = 1 - \sum_{k=1}^{K} p_k^2$$

公式中的 $\sum_{k=1}^{K} p_k^2$ 表示从样本集中随机抽取两个样本示例，两个示例类别一致的概率，基尼指数越小表明数据集 D 的中同一类样本的数量越多，或者说纯度越高。如果某特征变量为 a，样本集可以通过该特征变量将响应变量划分为 v 个类别，则该样本集的基尼指数为：

$$Gini(D,a) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} Gini(D^v)$$

# 四、基尼指数（Gini index）

CART 决策树是二分类树，每次分裂生长时，将当前节点的样本集分成两部分，包括属性变量取值为 $v$ 或者不为 $v$，样本示例全集 D 划分为 $D^v$ 和 $\widetilde{D}^v$，则样本示例全集的基尼指数为：

$$Gini(D,a,v)=\frac{|D^v|}{|D|}Gini(D^v)+\frac{|\widetilde{D}^v|}{|D|}Gini(\widetilde{D}^v)$$

在 CART 决策树方法下，会找到使得 $Gini(D,a,v)$ 最小的 $a$ 和 $v$，然后按照特征变量 $a$ 将样本示例全集分为取值为 $v$ 以及取值不为 $v$ 两部分，形成二叉树。

事实上，ID3、C4.5 和 CART 三种决策树算法的本质是一致的，不论是基于什么样的特征变量选择方法，其思路都是针对每个特征变量，寻找其最优临界值，基于最优临界值计算采纳该特征变量时可实现的样本子集纯度的改进幅度（不纯度的下降幅度）；遍历各特征变量后，选择可以使得样本子集纯度的改进幅度（不纯度的下降幅度）最大的特征变量，及其最优的临界值作为生长分裂标准。

# 决策树的剪枝

在决策树的生长过程中，如果我们不加限制，那么决策树就会尽情生长下去，造成决策树分支过多。最为极端的情形就是生长到最后每个样本示例都成为一个节点，那么可以预见的就是势必造模型产生过拟合线性，泛化能力不足。所以为达到应有的泛化能力或预测效果，我们有必要对决策树进行“剪枝（pruning）”处理，使其达到一定程度后能够停止生长。

剪枝有分预剪枝（prepruning）和后剪枝(post-pruning)两种。预剪枝的基本思路是“边构造边剪枝”，在树的生长过程中设定一个指标，如果达到该指标，或者说当前节点的划分不能带来决策树泛化性能的提升，决策树就会停止生长并将当前节点标记为叶节点。

# 决策树的剪枝

后剪枝的基本思路是“构造完再剪枝”，首先让决策树尽情生长，从训练集中生成一棵完整的决策树，一直到叶节点都有最小的不纯度值，然后自底向上遍历所有非叶节点，若将该节点对应的子树直接替换成叶节点能带来决策树泛化能力的提升，则将该子树替换成叶节点，达到剪枝的效果。

在两张剪枝方法的选择方面，预剪枝存在一定局限性，因为在树的生长过程中，很多时候虽然当前的划分会导致测试集准确率降低，但如果能够继续生长，在之后的划分中，准确率可能会有显著上升；而一旦停止分支，使得节点N成为叶节点，就断绝了其后继节点进行“好”的分支操作的任何可能性，所以预剪枝容易造成欠拟合。

# 决策树的剪枝

后剪枝则较好的克服了这一局限性，而且可以充分利用全部训练集的信息而无需保留部分样本用于交叉验证，所以优势较为明显。但是后剪枝是在构建完全决策树之后进行的，并且要自底向上自底向上遍历所有非叶节点，所以其计算时间、计算量要远超预剪枝方法，尤其是针对大样本数据集的时候。实务中，针对小样本数据集，后剪枝方法是首选；针对大样本数据集，用户需要权衡预测效果和计算量。

# 包含剪枝决策树的损失函数

**一、分类树**

针对分类树，其节点分裂准则为“节点不纯度下降最大化+剪枝惩罚项”，假定某决策树有$|T|$个叶节点，$N_t$为叶节点 t 的样本示例个数，包含剪枝决策树的损失函数即为：

$$C_\alpha(T) = \sum_{t=1}^{|T|} N_t H_t(T) + \alpha|T|$$

函数公式中，$\sum_{t=1}^{|T|} N_t H_t(T)$表示误差大小，衡量模型的拟合程度，$\alpha|T|$表示模型复杂程度，$\alpha \geqslant 0$为惩罚项（正则化系数）。

其中$H_t(T) = -\sum_k \frac{N_{tk}}{N_t} log_2 \frac{N_{tk}}{N_t}$，即为叶节点 t 的信息熵，$N_{tk}$为叶节点 t 中分类为 k 的样本示例个数。

我们需要做的就是使得决策树的损失函数最小化，大体上叶节点$|T|$越多，单个叶节点 t 的信息熵越低，模型整体的拟合程度越好，但是其模型复杂程度$\alpha|T$也会上升，于是就需要在模型拟合程度和模型复杂程度之间进行平衡，找到最优点。

# 包含剪枝决策树的损失函数

## 二、回归树

针对回归树，其节点分裂准则为“最小化残差平方和+剪枝惩罚项”，假定某决策树有$|T|$个叶节点，在叶节点t共有k个样本示例，包含剪枝决策树的损失函数即为：

$$C_a(T) = \sum_{t=1}^{|T|} H_t(T) + \alpha|T|$$

其中$H_t(T) = \sum_k (y_k - \hat{y_k})^2$，$H_t(T)$即为所有样本示例的残差平方和（残差为实际值与预测值之差）。

与分类树相同，我们需要做的就是使得决策树的损失函数最小化，大体上叶节点$|T|$越多，单个叶节点t的残差平方和越小，模型整体的拟合程度越好，但是其模型复杂程度$\alpha|T$也会上升，于是就需要在模型拟合程度和模型复杂程度之间进行平衡，找到最优点。

# 变量重要性

决策树算法是一种典型的非参数算法，这也意味着在其模型中不包含类似于回归系数之类的参数存在，难以直接评价特征变量对于响应变量的影响程度。实务中，用户常遇到的一个问题就是，在决策树模型中采纳的诸多特征变量之间，重要性排序是怎样的?

一个非常明确但又容易被误解的事实就是，并非先采用的特征变量就必然是贡献最大的，而是应该通过计算因采纳该变量引起的残差平方和（或信息增益、信息增益率、基尼指数等指标）变化的幅度来进行排序，残差平方和下降或基尼指数下降越多、信息增益或信息增益率提升越多，说明变量在决策树模型中越为重要。

PART 02

# 数据准备

# 数据准备

本节我们以“数据13.1”和“数据13.2”为例进行讲解。“数据13.1”记录的是某商业银行个人信用卡客户信用状况，变量包括这些信用卡客户是否发生违约（credit）、年龄（age）、受教育程度（education）、工作年限（workyears）、居住年限（resideyears）、年收入水平（income）、债务收入比（debtratio）、信用卡负债（creditdebt）、其他负债（otherdebt）。是否发生违约（credit）分为两个类别：“0”表示“未发生违约”，“1”表示“发生违约”；受教育程度（education）分为五个类别：“2”表示“初中”，“3”表示“高中及中专”，“4”表示“大学本专科”，“5”表示“硕士研究生”，“6”表示“博士研究生”。

**针对“数据13.1”的决策树模型，我们以是否发生违约（credit）为响应变量，以年龄（age）、受教育程度（education）、工作年限（workyears）、居住年限（resideyears）、年收入水平（income）、债务收入比（debtratio）、信用卡负债（creditdebt）、其他负债（otherdebt）为特征变量，使用分类决策树算法进行拟合。**

# 数据准备

“数据13.2”的案例数据是一些上市商业银行大股东持股量和其净资产收益率，变量包括这些商业银行净资产收益率（roe）、第一大股东的持股量（top1）、前五大股东的持股量（top5）、前十大股东的持股量（top10）、第一大股东持股量的平方项（stop1）、前五大股东持股量的平方项（stop5）、前十大股东持股量的平方项（stop10）。

**针对“数据13.2”的决策树模型，我们以“净资产收益率（roe）”为响应变量，以第一大股东的持股量（top1）、前五大股东的持股量（top5）、前十大股东的持股量（top10）、第一大股东持股量的平方项（stop1）、前五大股东持股量的平方项（stop5）、前十大股东持股量的平方项（stop10）为特征变量，使用回归决策树算法进行拟合。**

# 载入分析所需要的模块和函数

在进行分析之前，我们首先载入分析所需要的模块和函数，读取数据集并进行观察。

示例

参阅教材内容

PART 03

# 分类问题决策树算法示例

# 变量设置及数据处理

示例

参阅教材内容

# 未考虑成本-复杂度剪枝的决策树分类算法模型

一、 当分裂准则为信息熵时

示例

参阅教材内容

# 未考虑成本-复杂度剪枝的决策树分类算法模型

二、 当分裂准则为基尼指数时

示例

参阅教材内容

# 考虑成本-复杂度剪枝的决策树分类算法模型

示例

参阅教材内容

# 绘制图形观察"叶节点总不纯度随alpha值变化情况"

示例

参阅教材内容

# 绘制图形观察“节点数和树的深度随alpha值变化情况”

示例

参阅教材内容

# 绘制图形观察“训练样本和测试样本的预测准确率随alpha值变化情况”

示例

参阅教材内容

# 通过10折交叉验证法寻求最优alpha值

示例

参阅教材内容

# 决策树特征变量重要性水平分析

示例

参阅教材内容

# 绘制ROC曲线

示例

参阅教材内容

# 运用两个特征变量绘制决策树算法决策边界图

示例

参阅教材内容

PART 04

# 回归问题决策树算法示例

# 变量设置及数据处理

示例

参阅教材内容

# 未考虑成本-复杂度剪枝的决策树回归算法模型

示例

参阅教材内容

# 考虑成本-复杂度剪枝的决策树回归算法模型

示例

参阅教材内容

# 绘制图形观察"叶节点总均方误差随alpha值变化情况"

示例

参阅教材内容

# 绘制图形观察“节点数和树的深度随alpha值变化情况”

示例

参阅教材内容

# 绘制图形观察“训练样本和测试样本的拟合优度随alpha值变化情况”

示例

参阅教材内容

# 通过10折交叉验证法寻求最优alpha值并开展特征变量重要性水平分析

示例

参阅教材内容

# 最优模型拟合效果图形展示

示例

参阅教材内容

# 构建线性回归算法模型进行对比

示例

参阅教材内容

PART 05

# 习　题

# 习题

1、使用使用“数据5.1”数据文件（详情已在第5章中介绍），把响应变量设定为“V1征信违约记录”，将其他变量作为特征变量，具体包括“V2资产负债率”、“V6主营业务收入”、“V7利息保障倍数”、“V13银行负债”、“V9其他渠道负债”，构建决策树分类算法模型。

（1）变量设置及数据处理

（2）构建未考虑成本-复杂度剪枝的决策树分类算法模型

（3）构建考虑成本-复杂度剪枝的决策树分类算法模型

（4）绘制图形观察"叶节点总不纯度随alpha值变化情况"

# 习题

(5) 绘制图形观察“节点数和树的深度随alpha值变化情况”

(6) 绘制图形观察“训练样本和测试样本的预测准确率随alpha值变化情况”

(7) 通过10折交叉验证法寻求最优alpha值

(8) 开展决策树特征变量重要性水平分析

(9) 绘制ROC曲线

(10) 运用两个特征变量绘制决策树算法决策边界图

# 习题

2、使用“数据4.3”数据文件（详情已在第4章习题部分中介绍），Profit contribution为利润贡献度，作为响应变量；Net interest income为净利息收入、Intermediate income为中间业务收入、Deposit and finance daily为日均存款加理财之和，均作为特征变量，构建决策树回归算法模型。

（1）变量设置及数据处理

（2）构建未考虑成本-复杂度剪枝的决策树回归算法模型

（3）构建考虑成本-复杂度剪枝的决策树回归算法模型

（4）绘制图形观察"叶节点总均方误差随alpha值变化情况"

# 习题

(5) 绘制图形观察“节点数和树的深度随alpha值变化情况”

(6) 绘制图形观察“训练样本和测试样本的拟合优度随alpha值变化情况”

(7) 通过10折交叉验证法寻求最优alpha值并开展特征变量重要性水平分析

(8) 最优模型拟合效果图形展示

(9) 构建线性回归算法模型进行对比。

# 感谢聆听

THANKS