人工智能之机器学习

决策树与随机森林

产品研发中心 -- 李军

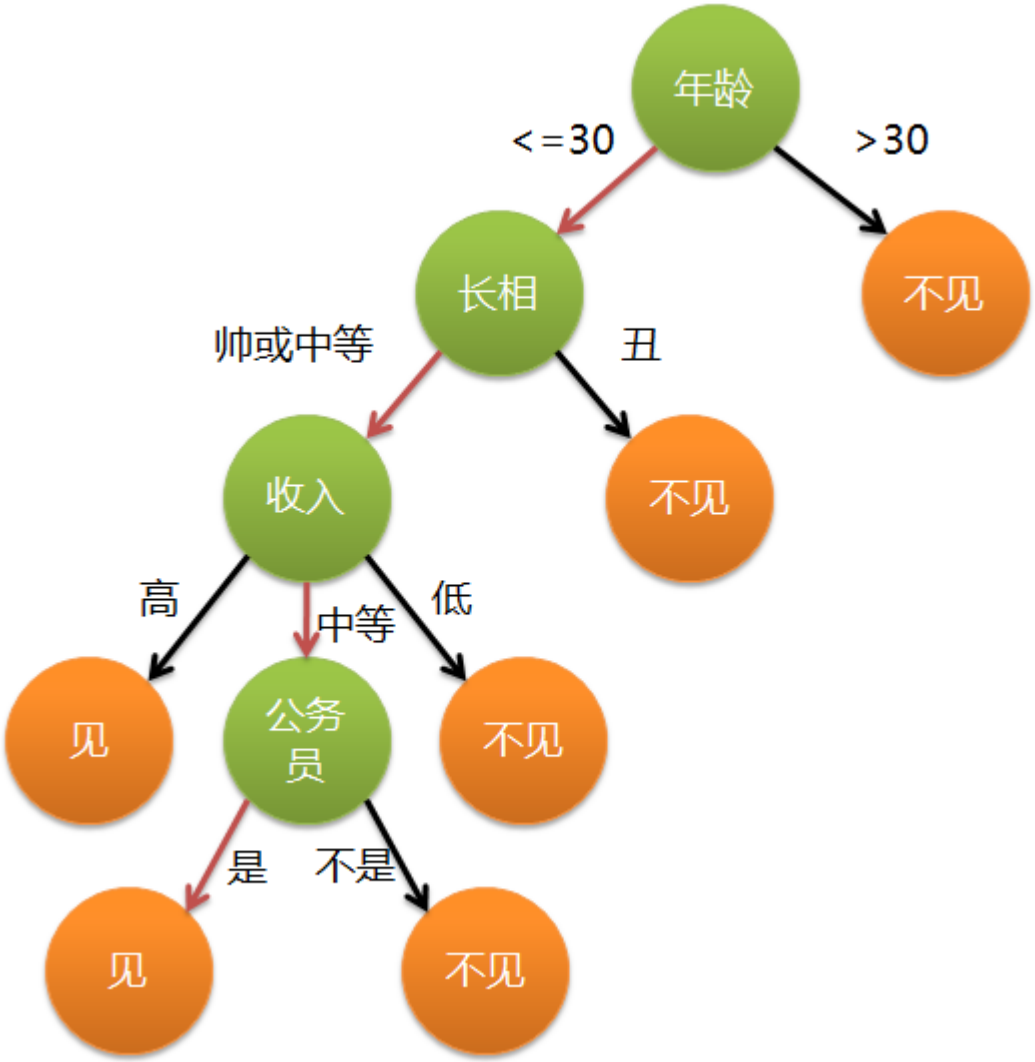
凯通科技
www.ctt-net.com

用软件重新定义世界，让世界更加智能互联

决策树简介

决策树（Decision Tree）：

- 决策树是一种树型结构，其中每个内部结点表示在一个属性上的测试，每个分支代表一个测试输出，每个叶结点代表一种类别。
- 决策树学习是以实例为基础的归纳学习。
- 决策树学习采用的是自顶向下的递归方法，其基本思想是以信息熵为度量构造一棵熵值下降最快的树，到叶子节点处的熵值为零，此时每个叶节点中的实例都属于同一类。



决策树逻辑

```
If (obj.相貌=="帅") then
    {
        If (obj.财富>=1000000000) then
            {
            print (obj.Name + "高富帅");
            }
        else
            {
            print (obj.Name + "是帅哥");
            }
    }
else
    {
        If (obj.财富>=1000000000) then
            {
            print (obj.Name + "是高富");
            }
        else
            {
            print (obj.Name + "是屌丝");
            }
    }
```

- 可以将决策树看成一个 if – then 规则的集合
- 由决策树的根节点到叶节点的每一条路径构建一条规则
- 路径上内部节点的特征对应规则的条件，叶节点的类对应这规则的结论
- 重要特性（互斥且完备），每一个实例都被一条路径或一条规则所覆盖，且只被一条路径或一条规则所覆盖

决策树关键点

关键点：

- *决策树（decision tree）是一个树结构（可以是二叉树或非二叉树）。其每个非叶节点表示一个特征属性上的测试，每个分支代表这个特征属性在某个值域上的输出，而每个叶节点存放一个类别。*
- *构造决策树的关键性内容是进行属性选择度量，通常计算信息增益来选择树形，成熟的算法主要有：**ID3、C4.5、CART**。*
- *信息增益是特征选择中的一个重要指标，它定义为一个特征能够为分类系统带来多少信息，带来的信息越多（例如：中国梦），该特征越重要，又或者该事件发生的概率越小，则该事件的信息量越大。*

ID3算法

ID3算法的核心就是以信息增益度量属性选择，选择分裂后信息增益最大的属性进行分裂。

设D为用类别对训练元组进行的划分，则D的熵（entropy）表示为：

$$info(D) = -\sum_{i=1}^{m} p_i log_2(p_i)$$

其中pi表示第i个类别在整个训练元组中出现的概率，可以用属于此类别元素的数量除以训练元组元素总数量作为估计。
熵的实际意义表示是D中元组的类标号所需要的平均信息量。

现在我们假设将训练元组D按属性A进行划分，则A对D划分的期望信息为：

$$info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} info(D_j)$$

#分支越多的属性越分散则期望值会越小，则信息增益就越大

而信息增益即为两者的差值：

$$gain(A) = info(D) - info_A(D)$$

问题：训练出来的形状是一棵分支很多，深度很浅的树，貌似不太合理！

案例分析 – 网站账号真实性预测

根据某社区网站的抽样统计，该站10000个账号中有89%为真实账号（设为R0），11%为虚假账号（设为R1）。假定某一个账号有以下三个特征：日志密度L=s，好友密度为F=s，是否真实头像为H=yes，预测是否真实账号？

日志密度	好友密度	是否使用真实头像	账号是否真实
s	s	no	no
s	l	yes	yes
l	m	yes	yes
m	m	yes	yes
l	m	yes	yes
m	l	no	yes
m	s	no	no
l	m	no	yes
m	s	no	yes
s	s	yes	no

$$info(D) = -0.7log_20.7 - 0.3log_20.3 = 0.7 * 0.51 + 0.3 * 1.74 = 0.879$$

$$info_L(D) = 0.3 * (-\frac{0}{3}log_2\frac{0}{3} - \frac{3}{3}log_2\frac{3}{3}) + 0.4 * (-\frac{1}{4}log_2\frac{1}{4} - \frac{3}{4}log_2\frac{3}{4}) + 0.3 * (-\frac{1}{3}log_2\frac{1}{3} - \frac{2}{3}log_2\frac{2}{3}) = 0 + 0.326 + 0.277 = 0.603$$

$$gain(L) = 0.879 - 0.603 = 0.276$$

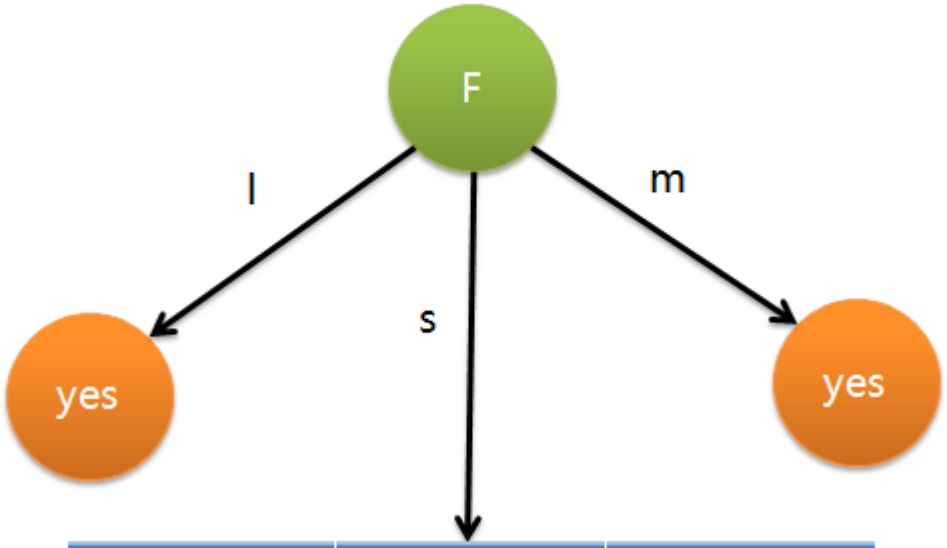
因此日志密度的信息增益是0.276。

用同样方法得到H和F的信息增益分别为0.033和0.553。

因为F具有最大的信息增益，所以第一次分裂选择F为分裂属性，分裂后的结果如下图表示：

- info(D)为真实账号样本概率7/10=0.7，为假账号样本概率3/10=0.3
- infoL(D)日志密度为I的出现概率3/10=0.3，而账号为no的概率0/3，为yes的概率3/3；为m的出现概率4/10=0.4，而账号为no的概率1/4，为yes的概率为3/4；为s的出现概率3/10=0.3,为no的概率1/3，为yes的概率2/3

案例分析 – 网站账号真实性预测



日志密度	是否使用真实头像	账号是否真实
s	no	no
m	no	no
m	no	yes
s	yes	no

在决策树构造过程中可能会出现这种情况：所有属性都作为分裂属性用光了，但有的子集还不是纯净集，即集合内的元素不属于同一类别。在这种情况下，由于没有更多信息可以使用了，一般对这些子集进行“多数表决”，即使用此子集中出现次数最多的类别作为此节点类别，然后将此节点作为叶子节点。

案例分析 – 电信客户满意度预测

根据历史用户评价数据可以建立满意度预警模型，目的就是为了预测哪些用户会给出不满意的评价，提前做些安抚与补救措施。目标变量为二分类：满意（记为0）和不满意（记为1），自变量为：故障原因、故障类型、修障时长。

客户ID	故障原因	故障类型	修障时长	满意度
01	1	5	10	1
02	1	5	14	0
03	1	5	12	1
04	2	5	16	0
05	2	5	18	1
06	2	6	22	0
07	3	6	20	1
08	3	6	24	0
09	3	6	23	1
10	3	6	25	0

$$info(D) = -\sum_{i=1}^{m} p_i log_2(p_i)$$

其中D为满意度。m=2（满意和不满意）。Pi为满意度中分别属于满意和不满意的概率。共计10条数据，满意5条，不满意5条。

$$\text{Info}(\text{满意度}) = -\frac{5}{10} * log_2\frac{5}{10} - \frac{5}{10} * log_2\frac{5}{10} = 1$$

$$\text{Info}_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} * \text{Info}(D_j)$$

其中A为故障原因，D为满意度，故障原因分别为{1、2、3}，V=3，
故障原因为1的划分中，有2个不满意和1个满意；
故障原因为2的划分中，有1个不满意和2个满意；
故障原因为3的划分中，有2个不满意和2个满意。

$$Info_{\text{故障原因}}(\text{满意度}) = \frac{3}{10} * \left(-\frac{2}{3} * log_2\frac{2}{3} - \frac{1}{3} * log_2\frac{1}{3}\right)$$
$$+ \frac{3}{10} * \left(-\frac{1}{3} * log_2\frac{1}{3} - \frac{2}{3} * log_2\frac{2}{3}\right)$$
$$+ \frac{4}{10} * \left(-\frac{2}{4} * log_2\frac{2}{4} - \frac{2}{4} * log_2\frac{2}{4}\right) = 0.165$$

信息增益Gain(故障原因)

= Info(满意度) - Info故障原因(满意度)

= 1 - 0.165 = 0.835

案例分析 – 电信客户满意度预测

$$Info_{故障类型}(满意度) = \frac{5}{10} * \left(-\frac{3}{5} * log_2\frac{3}{5} - \frac{2}{5} * log_2\frac{2}{5}\right)$$
$$+ \frac{5}{10} * \left(-\frac{2}{5} * log_2\frac{2}{5} - \frac{3}{5} * log_2\frac{3}{5}\right) = 0.205$$

故障类型的信息增益Gain(故障类型) = 1 - 0.205 = 0.795

修障时长为连续型变量，由小到大递增排序，取相邻两个值的中点作为分裂点，然后按照离散型变量计算信息增益，取其中最大的信息增益作为最终的分裂点，比如对于中点11则有两个子集（<=11和>11）

10	12	14	16	18	20	22	23	24	25
11	13	15	17	19	21	22.5	23.5	24.5	

$$Info_{修障时长\ 11}(满意度)$$
$$= \frac{1}{10} * \left(-\frac{1}{1} * log_2\frac{1}{1}\right) + \frac{9}{10} * \left(-\frac{4}{9} * log_2\frac{4}{9} - \frac{5}{9} * log_2\frac{5}{9}\right)$$

同理分别求得各个中点的信息增益，选取其中最大的信息增益作为分裂点，如取中点11，然后与故障原因和故障类型的信息增益相比较，取最大的信息增益作为第一个树叉的分支。

C4.5算法

C4.5使用信息增益率（gain ratio）的信息增益扩充（如：速度与加速度）

C4.5算法首先定义了“分裂信息”，其定义可以表示成：

$$split\_info_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} log_2(\frac{|D_j|}{|D|})$$

其中各符号意义与ID3算法相同，然后，增益率被定义为：

$$gain\_ratio(A) = \frac{gain(A)}{split\_info(A)}$$

电信客户满意度案例分析：

$$\text{SplitInfo}_{\text{故障原因}}(\text{满意度}) = -\frac{3}{10} * log_2\frac{3}{10} - \frac{3}{10} * log_2\frac{3}{10} - \frac{4}{10} * log_2\frac{4}{10} = 1.201$$

Gain(故障原因) = 0.835（前文已求得）

Gain_ratio(故障原因) = 0.835 / 1.201 = 0.695

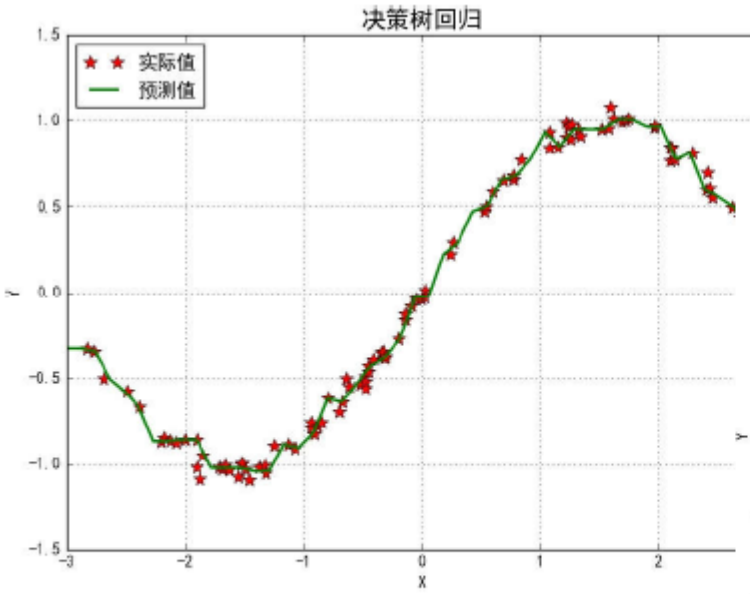
决策树总结

优点：

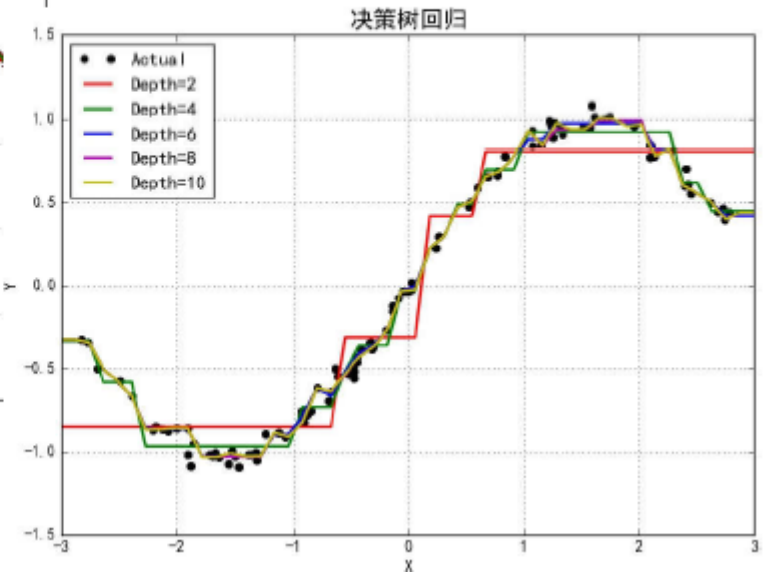
- 计算简单，易于理解，可解释性强
- 在相对短的时间内能够对大型数据源做出可行且效果良好的结果

缺点：

- 容易发生过拟合，泛化能力弱（采用剪枝、随机森林弥补）
- 对于那些各类别样本数量不一致的数据，在决策树当中，信息增益的结果偏向于那些具有更多数值的特征



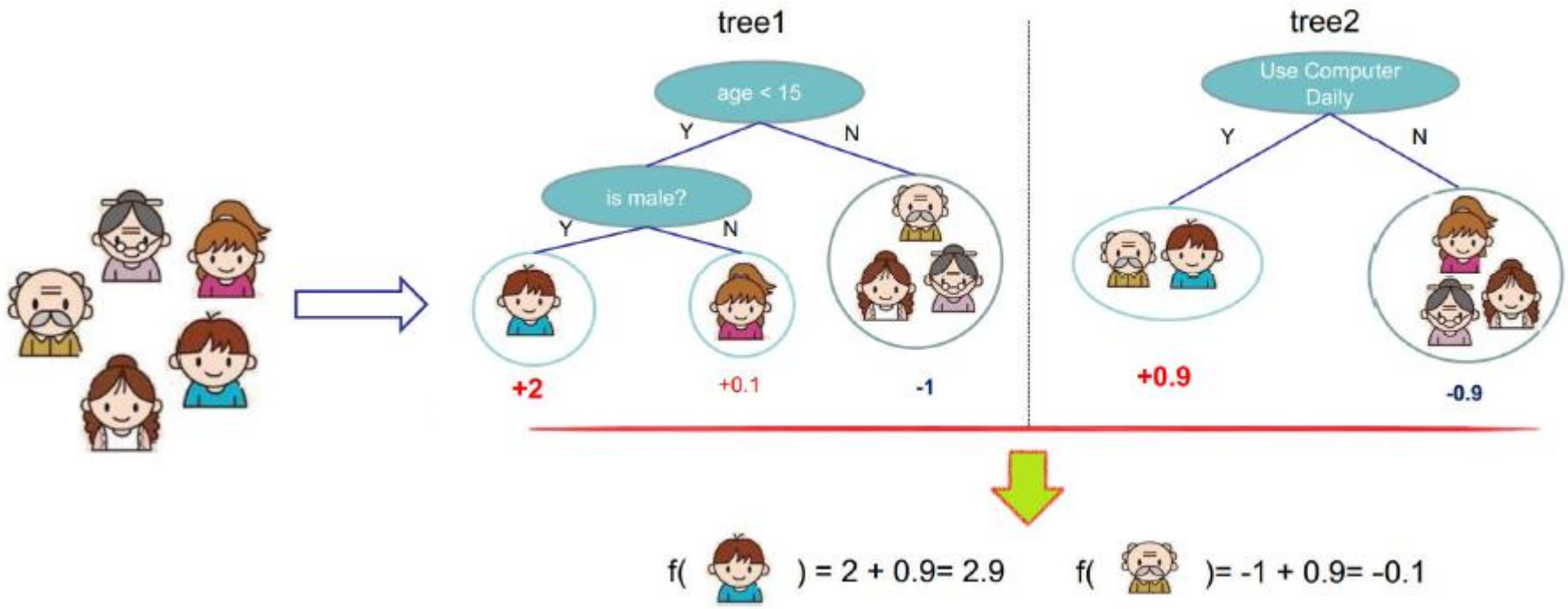
过拟合



多阶拟合

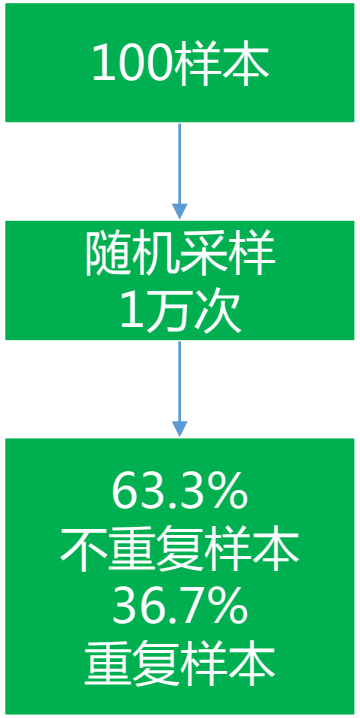
一个简单的随机森林分类

- 输入数据**x**：M个样本数据，每个数据包括年龄、性别、职业、每日使用计算机时间等
- 输出y：该样本是否喜欢计算机游戏



Bootstrap重采样

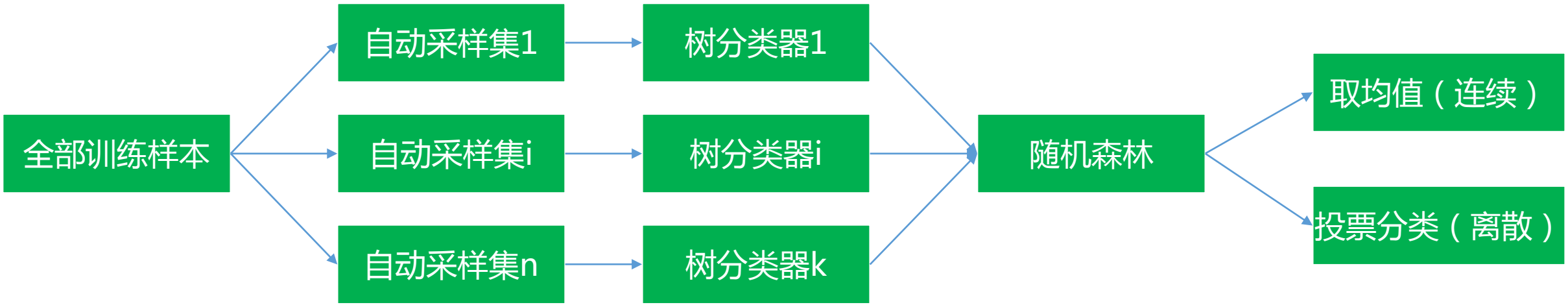
- 可以发现，Bootstrap每次约有36.79%的样本不会出现在Bootstrap所采集的样本集合中，将未参与模型训练的数据称为袋外数据OOB(Out Of Bag)。它可以用于取代测试集用于误差估计。



随机森林

随机森林策略

- 从样本集中用Bootstrap重采样选出n个样本
- 从所有属性中随机选择k个属性，选择最佳分割属性作为节点建立CART决策树
- 重复以上两步m次，即建立了m棵CART决策树
- 这m个CART决策树形成随机森林，通过投票表决结果，决定数据属于哪一类



CART算法

CART算法首先计算分裂属性的不纯度，再利用不纯度计算Gini系数，
取最小的Gini系数的属性作为树的分支

- 计算数据集不纯度

$$Gini(S) = 1 - \sum_{i=1}^{C} p_i^2$$

- 计算属性Gini指标

$$Gini_F(S) = \frac{|S_1|}{S} Gini(S_1) + \frac{|S_2|}{S} Gini(S_2)$$

随机森林案例分析

有房者	婚姻状况	年收入	拖欠贷款者
是	单身	125K	否
否	已婚	100K	否
否	单身	70K	否
是	已婚	120K	否
否	离异	95K	是
否	已婚	60K	否
是	离异	220K	否
否	单身	85K	是
否	已婚	75K	否
否	单身	90K	是

		有房	无房
拖欠贷款	否	3	3
拖欠贷款	是	0	4

$Gini(t_1)=1-(3/3)^2-(0/3)^2=0$

$Gini(t_2)=1-(4/7)^2-(3/7)^2=0.4849$

$Gini = 3/10 * 0 + 7/10 * 0.4849 = 0.343$

随机森林案例分析

对离散值如{x,y,z}，则在该属性上的划分有三种情况
({{x,y},{z}},{{x,z},y},{{y,z},x})，空集和全集的划分除外

	单身或已婚	离异
否	6	1
是	2	1

$\mathbf{Gini(t_1)=1-(6/8)^2-(2/8)^2=0.375}$

$Gini(t_2)=1-(1/2)^2-(1/2)^2=0.5$

$Gini=8/10\times0.375+2/10\times0.5=0.4$

	单身或离异	已婚
否	3	4
是	3	0

$\mathbf{Gini(t_1)=1-(3/6)^2-(3/6)^2=0.5}$

$Gini(t_2)=1-(4/4)^2-(0/4)^2=0$

$Gini=6/10\times0.5+4/10\times0=0.3$

	离异或已婚	单身
否	5	2
是	1	2

$\mathbf{Gini(t_1)=1-(5/6)^2-(1/6)^2=0.2778}$

$Gini(t_2)=1-(2/4)^2-(2/4)^2=0.5$

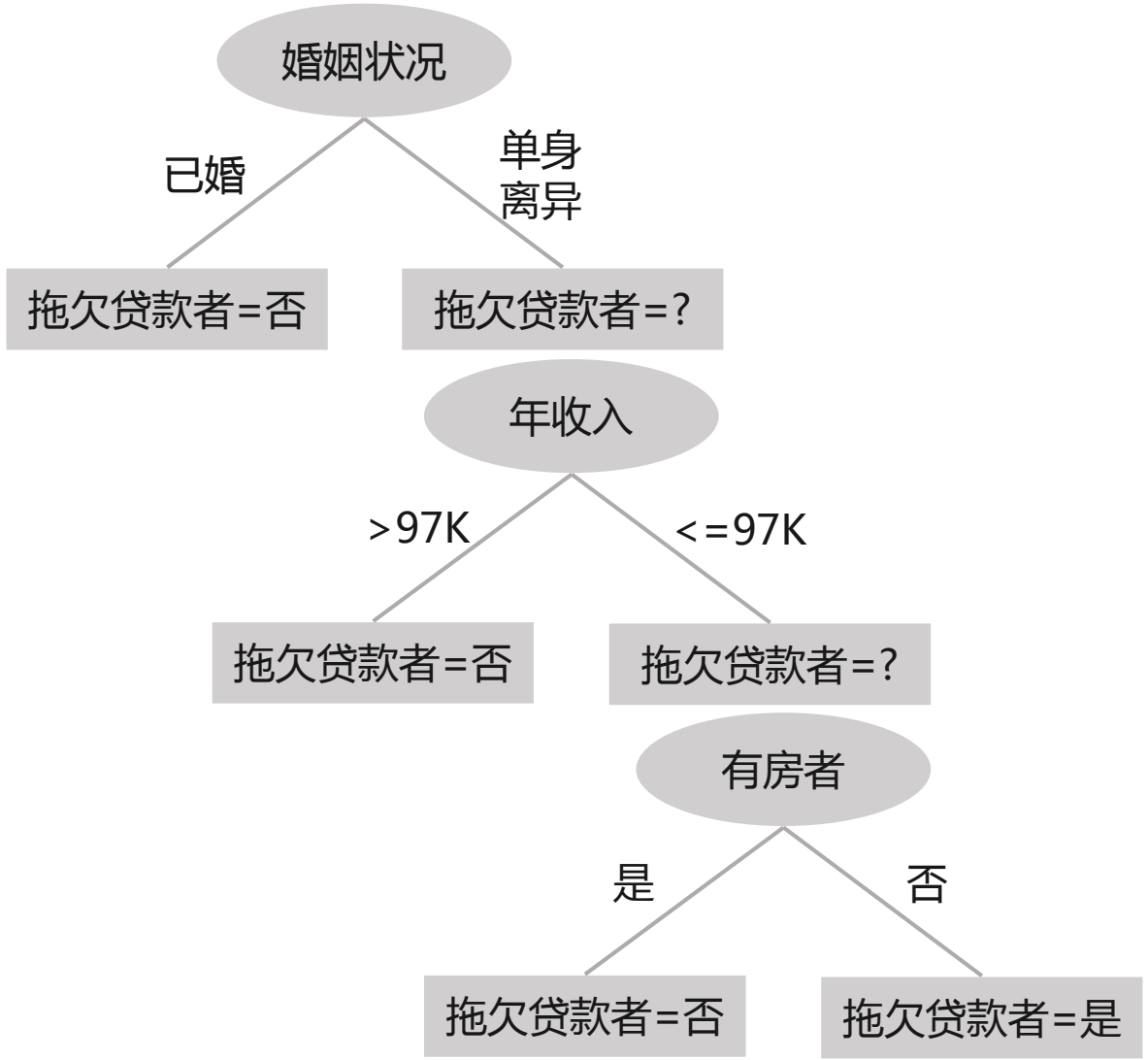
$Gini=6/10\times0.2778+4/10\times0.5=0.3667$

随机森林案例分析

对于连续值处理引进“分裂点”的思想，假设样本集中某个属性共n个连续值，则有n-1个分裂点，每个“分裂点”为相邻两个连续值的均值 (a[i] + a[i+1]) / 2。

	60		70		75		85		90		95		100		120		125		220	
	65		72		80		87		92		97		110		122		172			
	≤	>	≤	>	≤	>	≤	>	≤	>	≤	>	≤	>	≤	>	≤	>		
是	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0		
否	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1		
Gini	0.400		0.375		0.343		0.417		0.400		0.310		0.343		0.375		0.400			

随机森林案例分析


婚姻状况
已婚
单身
离异
拖欠贷款者=否
拖欠贷款者=?
年收入
>97K
<=97K
拖欠贷款者=否
拖欠贷款者=?
有房者
是
否
拖欠贷款者=否
拖欠贷款者=是

投票机制

- 简单投票机制
  - 一票否决(一致表决)
  - 少数服从多数
    - 有效多数(加权)
  - 阈值表决
- 贝叶斯投票机制

案例实践 – 鸢尾花种类预测

```
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
5.4,3.7,1.5,0.2,Iris-setosa
4.8,3.4,1.6,0.2,Iris-setosa
4.8,3.0,1.4,0.1,Iris-setosa
4.3,3.0,1.1,0.1,Iris-setosa
5.8,4.0,1.2,0.2,Iris-setosa
5.7,4.4,1.5,0.4,Iris-setosa
5.4,3.9,1.3,0.4,Iris-setosa
5.1,3.5,1.4,0.3,Iris-setosa
5.7,3.8,1.7,0.3,Iris-setosa
5.1,3.8,1.5,0.3,Iris-setosa
5.4,3.4,1.7,0.2,Iris-setosa
5.1,3.7,1.5,0.4,Iris-setosa
4.6,3.6,1.0,0.2,Iris-setosa
5.1,3.3,1.7,0.5,Iris-setosa
4.8,3.4,1.9,0.2,Iris-setosa
5.0,3.0,1.6,0.2,Iris-setosa
5.0,3.4,1.6,0.4,Iris-setosa
5.2,3.5,1.5,0.2,Iris-setosa
5.2,3.4,1.4,0.2,Iris-setosa
4.7,3.2,1.6,0.2,Iris-setosa
4.8,3.1,1.6,0.2,Iris-setosa
```

训练样本：

Iris.data

属性包括：

花萼长度，花萼宽度

花瓣长度，花瓣宽度

类别分类：

Iris-setosa

Iris-versicolor

Iris-virginica

Iris_DecisionTree.py

*（**决策树**）*

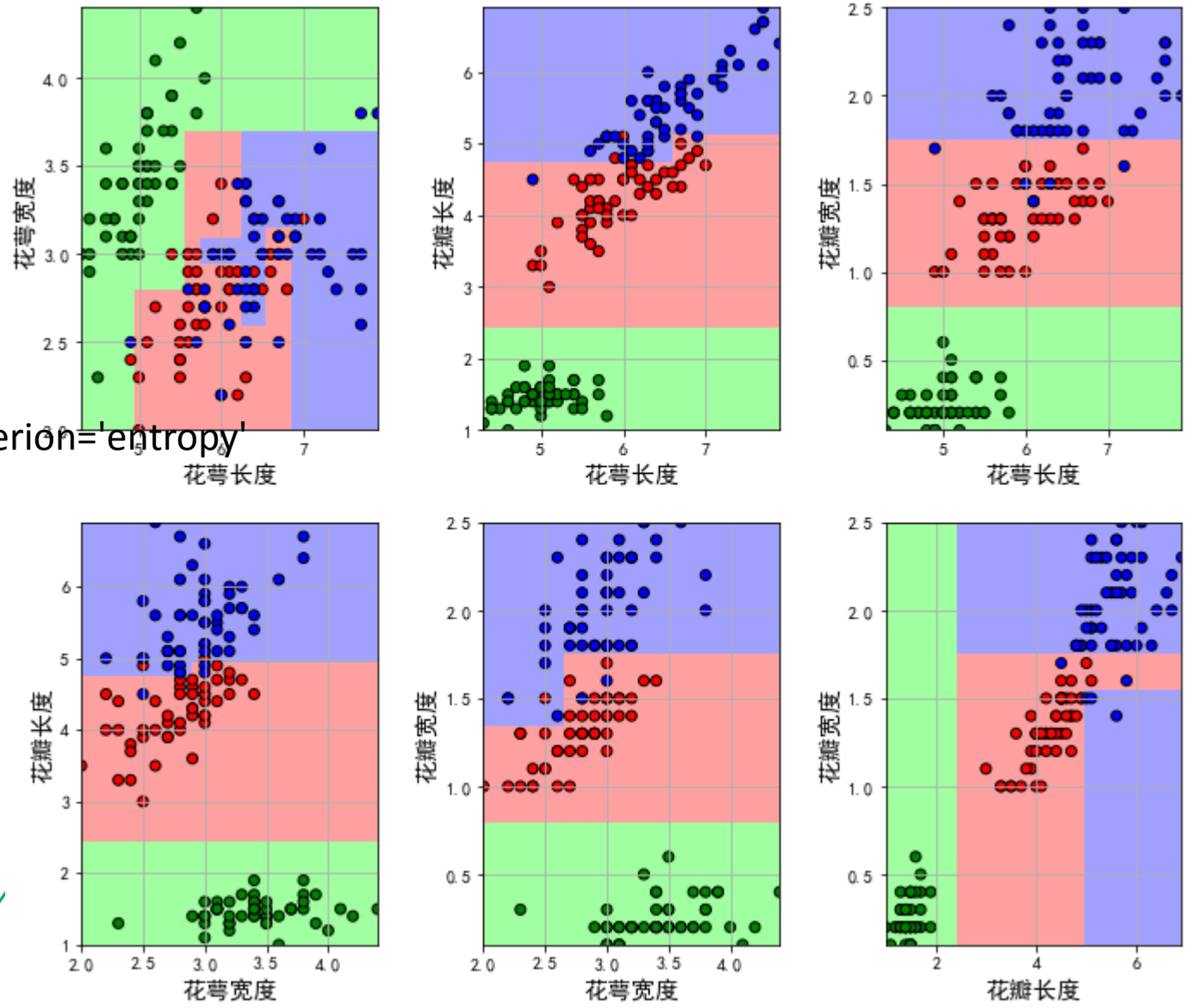
Iris_DecisionTree_Enum.py

*（**决策树 + 图形**）*

Iris_RandomForest_Enum.py

*（**随机森林 + 图形**）*

criterion='entropy'



感谢您的聆听！

Thank you for your time!

凯通科技
www.ctt-net.com

用软件重新定义世界，让世界更加智能互联