决策树

目录

Contents

学习目标

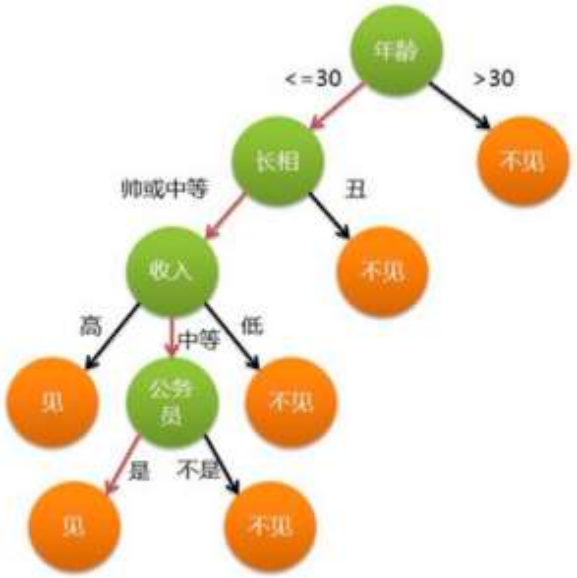
Learning Objectives

1. 理解决策树算法的基本思想
2. 知道构建决策树的步骤

决策树简介 - 生活中的决策树

- 女孩相亲的决策树

女儿：多大年纪了？
母亲：26。
女儿：长的帅不帅？
母亲：挺帅的。
女儿：收入高不？
母亲：不算很高，中等情况。
女儿：是公务员不？
母亲：是，在税务局上班呢。
女儿：那好，我去见见。



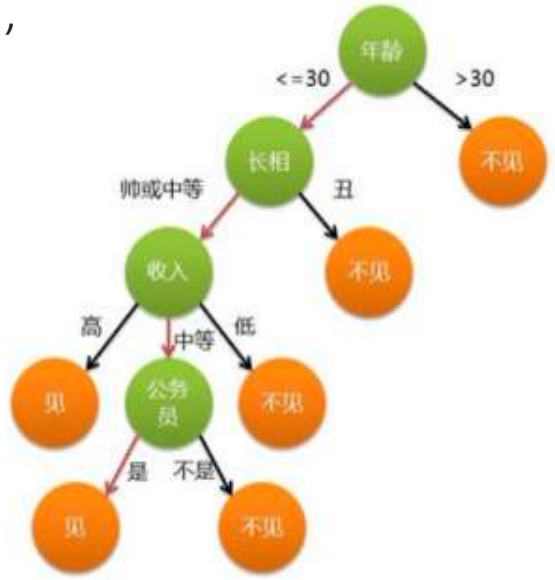
年龄	长相	收入	是否公务员	预测值
28	帅	高	否	?
30	丑	高	是	?
39	帅	中等	否	?

决策树简介

- 决策树是一种树形结构

  树中每个内部节点表示一个特征上的判断，每个分支代表一个判断结果的输出，

  每个叶子节点代表一种分类结果



- 决策树的建立过程

  1. 特征选择：选取有较强分类能力的特征。

  2. 决策树生成：根据选择的特征生成决策树。

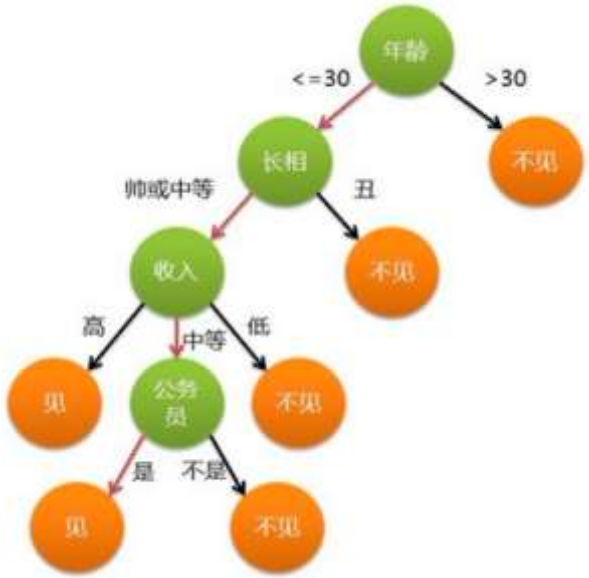
  3. 决策树也易过拟合，采用剪枝的方法缓解过拟合。

思考

1、女孩相亲过程中，更看重哪些特征呢？

年龄 > 长相 > 收入 > 是否公务员

2、机器学习算法中，选择数据集中的哪些特征进行分裂，会更好呢？



目录

Contents

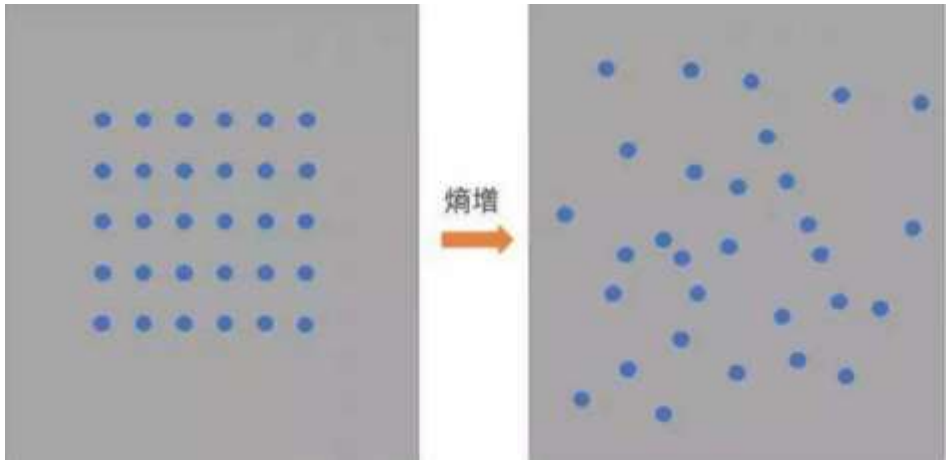
学习目标

Learning Objectives

1. 理解信息熵的意义
2. 理解信息增益的作用
3. 知道ID3树的构建流程

信息熵

- 熵 Entropy： 信息论中代表随机变量不确定度的度量
  - 熵越大，数据的不确定性越高，信息就越多
  - 熵越小，数据的不确定性越低



- 举个栗子：

1.数据 α：ABCDEFGH

2.数据 β：AAAABBCD

数据α 包含了 8 种信息，数据 β 包含了 4 种信息，
特征α 的信息熵大于特征 β 的信息熵。

信息熵

- 计算方法
  - H(x) = - $\sum_{i=0}^{n} P(x_i) \log_2 P(x_i)$
  - 其中 $P(x_i)$ 表示数据中类别出现的概率，H(x) 表示信息的信息熵值

- 栗子1-1：计算数据 α（ABCDEFGH） 信息熵，其中A、B、C、D、E、F、G、H 出现的概率为：1/8
  - H(α) = - $\sum_{i=0}^{n} P(x_i) \log_2 P(x_i)$ = $(-\frac{1}{8}\log_2\frac{1}{8}) * 8 = 3$

- 栗子1-2：计算数据 β（AAAABBCD） 信息熵

  其中A 出现的概率为1/2，B 出现的概率为1/4，C、D 出现的概率为1/8
  - H(β) = - $\sum_{i=0}^{n} P(x_i) \log_2 P(x_i)$ = $(-\frac{1}{2}\log_2\frac{1}{2}) + (-\frac{1}{4}\log_2\frac{1}{4}) + (-\frac{1}{8}\log_2\frac{1}{8}) * 2$ = $\frac{1}{2}*1 + \frac{1}{4}*2 + \frac{1}{8}*3*2 = 1.75$

ID3决策树-信息熵

- 栗子2-1：假如数据集有三个类别，分别占比为：{⅓, ⅓, ⅓}，信息熵：
  - H = - $\sum_{i=0}^{n} P(x_i) \log_2 P(x_i)$ = $(-\frac{1}{3}\log_2\frac{1}{3}) * 3 = 1.0986$
- 栗子2-2：假如数据集有三个类别，分别占比为：{1/10, 2/10, 7/10}，信息熵：
  - H = - $\sum_{i=0}^{n} P(x_i) \log_2 P(x_i)$ = $(-\frac{1}{10}\log_2\frac{1}{10}) + (-\frac{2}{10}\log_2\frac{2}{10}) + (-\frac{7}{10}\log_2\frac{7}{10})$ = 0.8018
- 栗子2-3：假如数据集有三个类别，分别占比为：{1, 0, 0}，信息熵：
  - H = - $\sum_{i=0}^{n} P(x_i) \log_2 P(x_i)$ = $(-1\log_2 1)$ = 0

信息增益

- 概念

  特征a对训练数据集D的信息增益$Gain(D, a)$或g(D，a)，定义为集合D的熵H(D)与特征a给定条件下D的熵H(D | a)之差。

- 数学公式　　$Gain(D, a)$ = H(D)-H(D | a)

  信息增益 = 熵 - 条件熵

- 条件熵

  $H(D \mid a) = \sum_{v=1}^{n} \frac{D^V}{D} H(D^V) = \sum_{v=1}^{n} \frac{D^V}{D} \sum_{k=1}^{k} \frac{C^{kV}}{D^V} log \frac{C^{kV}}{D^V}$

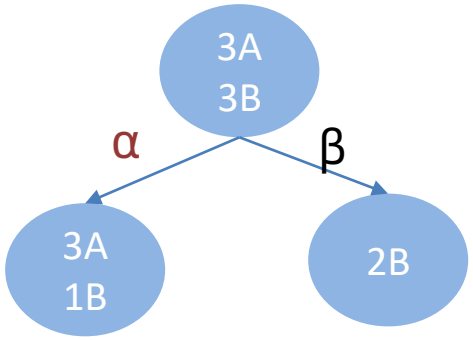
信息增益

特征a	目标值
α	A
α	A
β	B
α	A
β	B
α	B

已知6个样本，根据特征a：

α 部分对应的目标值为： AAAB

β 部分对应的目标值为：BB



- 条件为α的熵： $-\frac{3}{4}\log_2\frac{3}{4}-\frac{1}{4}\log_2\frac{1}{4}=0.81$
- 条件为β的熵： $-\frac{2}{2}\log_2\frac{2}{2}=0$
- 条件熵：# α 部分占了 4/6，β 部分占了 2/6

$$\frac{4}{6}*0.81+\frac{2}{6}*0=0.54$$

$Gain(D,a)$ = H(D)-H(D| a)

- 熵： $-\frac{3}{6}\log_2\frac{3}{6}-\frac{3}{6}\log_2\frac{3}{6}=1$
- 信息增益：熵 - 条件熵： 1.0 - 0.54 = 0.46

ID3决策树构建流程

1. 计算每个特征的信息增益
2. 使用信息增益最大的特征将数据集 拆分为子集
3. 使用该特征（信息增益最大的特征）作为决策树的一个节点
4. 使用剩余特征对子集重复上述（1，2，3）过程

ID3决策树

- 已知　某一个论坛客户流失率数据
- 需求　考察性别、活跃度特征哪一个特征对流失率的影响更大
- 分析
  - ✓ 15条样本：5个正样本、10个负样本
  - ✓ 1 计算熵
  - ✓ 2 计算性别信息增益
  - ✓ 3 计算活跃度信息增益
  - ✓ 4 比较两个特征的信息增益

uin	gender	act_info	is_lost
1	男	高	0
2	女	中	0
3	男	低	1
4	女	高	0
5	男	高	0
6	男	中	0
7	男	中	1
8	女	中	0
9	女	低	1
10	女	中	0
11	女	高	0
12	男	低	1
13	女	低	1
14	男	高	0
15	男	高	0

ID3决策树- 信息增益案例

- 计算熵　　$H(D)= (-\frac{5}{15}\log_2\frac{5}{15}) + (-\frac{10}{15}\log_2\frac{10}{15}) = 0.9812$
- 计算性别条件熵(a="性别")

$$H(D, 性别) = \sum_{v=1}^{n}\frac{D^V}{D}H(D^V)$$

$$= (\frac{8}{15})(-\frac{3}{8}\log_2\frac{3}{8} - \frac{5}{8}\log_2\frac{5}{8}) + (\frac{7}{15})(-\frac{2}{7}\log_2\frac{2}{7} - \frac{5}{7}\log_2\frac{5}{7})$$

- 计算性别信息增益(a="性别")

$$Gain(D, a) = H(D) - H(D \mid a)$$

$$= 0.9812 - (\frac{8}{15})(-\frac{3}{8}\log_2\frac{3}{8} - \frac{5}{8}\log_2\frac{5}{8}) + (\frac{7}{15})(-\frac{2}{7}\log_2\frac{2}{7} - \frac{5}{7}\log_2\frac{5}{7}) = 0.0064$$

	positive	negative	汇总
整体	5	10	15
男性	3	5	8
女性	2	5	7
高	0	6	6
中	1	4	5
低	4	0	4

ID3决策树- 信息增益案例

- 计算活跃度条件熵(a=“活跃度")

$$H(D, 活跃度) = \sum_{v=1}^{n} \frac{D^V}{D} H(D^V)$$

$$= (\frac{6}{15})(0) + (\frac{5}{15})(-\frac{1}{5}\log_2\frac{1}{5} - \frac{4}{5}\log_2\frac{4}{5}) + (\frac{4}{15})(0)$$

- 计算活跃度信息增益(a=“活跃度")

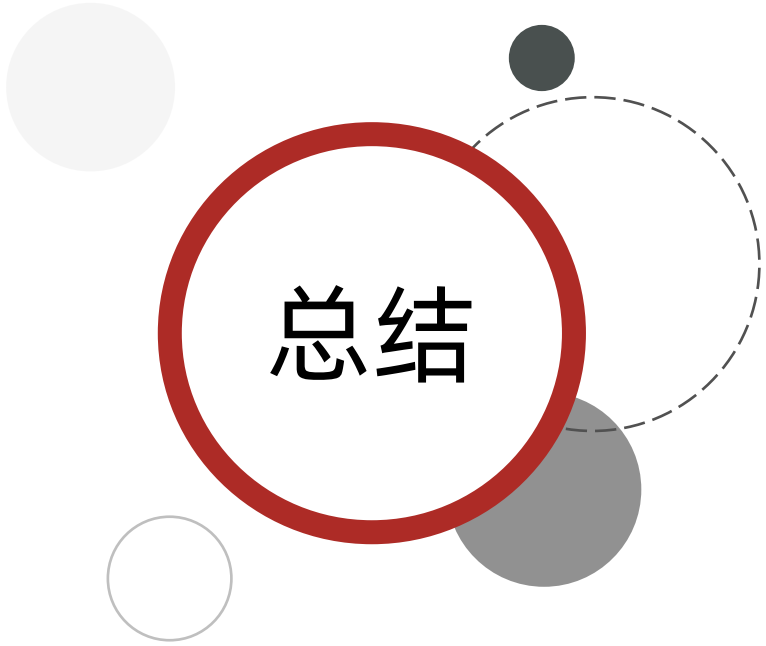
$$Gain(D, a) = H(D) - H(D \mid a)$$

$$= 0.9812 - (\frac{6}{15})(0) + (\frac{5}{15})(-\frac{1}{5}\log_2\frac{1}{5} - \frac{4}{5}\log_2\frac{4}{5}) + (\frac{4}{15})(0) = 0.6776$$

uin	gender	act_info	is_lost
1	男	高	0
2	女	中	0
3	男	低	1
4	女	高	0
5	男	高	0
6	男	中	0
7	男	中	1
8	女	中	0
9	女	低	1
10	女	中	0
11	女	高	0
12	男	低	1
13	女	低	1
14	男	高	0
15	男	高	0

	positive	negative	汇总
整体	5	10	15
男性	3	5	8
女性	2	5	7
高	0	6	6
中	1	4	5
低	4	0	4

结论：活跃度的信息增益比性别的信息增益大，对用户流失的影响比性别大。



1 决策树是什么?

决策树是一种树形结构，树中每个内部节点表示一个特征上的判断，每个分支代表一个判断结果的输出，每个叶子节点代表一种分类结果

2 信息熵?

在信息论中代表随机变量不确定度的度量

3.信息增益 ?

由于特征a而使得对数据D的分类不确定性减少的程度。

4. ID3树的构建流程

1.计算每个特征的信息增益
2.使用信息增益最大的特征将数据集 拆分为子集
3.使用信息增益最大的特征作为决策树的一个节点
4.重复上述步骤

练习

1、下列关于决策树的概念描述错误的是？

A）决策树算法需要构建树结构

B）决策树上的每一个主节点代表一个判断条件

C）决策树上的每一个叶节点代表一种分类结果

D）通过决策树不能明确特征的重要性程度

答案解析：决策树要根据特征重要性进行分裂

答案：D

2、下列关于熵和信息熵的描述错误的是？

A）熵越大，系统的混乱程度越小

B）信息熵是用来描述信息的完整性和有序性的

C）信息的有序状态越一致、数据越集中，信息熵越小，反之越大

答案解析：信息越混乱熵越大

答案：A

练习

3、下列关于信息增益的描述正确的是？（**多选**）

A）表达的是在用某个特征对数据集进行分裂

B）是ID3算法中的核心

C）需要消除的不确定性越大，信息增益越小，表示这个特征越不重要

D）对类别数较多的特征比较青睐

答案：ABD

目录

Contents

学习目标

Learning Objectives

1. 理解信息增益率的意义
2. 知道C4.5树的构建方法

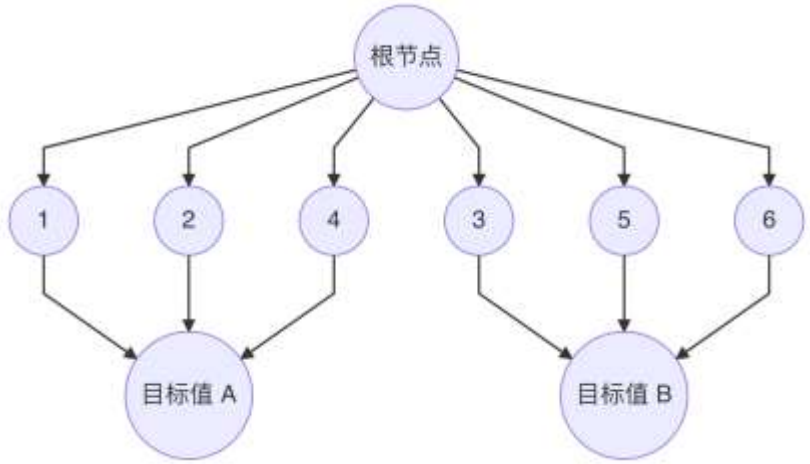
C4.5决策树

- ID3树的不足

  偏向于选择种类多的特征作为分裂依据

- 举个栗子

特征b	特征a	目标值
1	α	A
2	α	A
3	β	B
4	α	A
5	β	B
6	α	B



特征a有2个取值

特征b有6个取值

1 特征a作为分裂特征，会构造一棵深度为2的决策树，该树的预测准确率可能非常高

2 但整棵树过于依赖少数的特征（只根据少数特征进行学习），导致过拟合

信息增益率

- 信息增益率 = 信息增益 /特征熵
- 计算方法

Gain_Ratio(D, a) = $\frac{Gain(D,a)}{IV(a)}$ ，

- Gain_Ratio(D, a) 信息增益率
- IV(a) = - $\sum_{v=1}^{n} \frac{D^V}{D} Ent(\frac{D^V}{D})$ ， IV特征熵

- 信息增益率的本质
  - 特征的信息增益 ➗ 特征的内在信息
  - 相当于对信息增益进行修正，增加一个惩罚系数
  - 特征取值个数较多时，惩罚系数较小；特征取值个数较少时，惩罚系数较大。
  - 惩罚系数：数据集D以特征a作为随机变量的熵的倒数

信息增益率

- 已知数据集

特征b	特征a	目标值
1	α	A
2	α	A
3	β	B
4	α	A
5	β	B
6	α	B

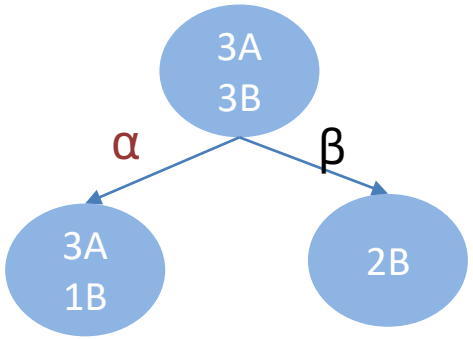
- 需求：求特征a、b的信息增益率

- 特征a的信息增益率：

1 信息增益： $(-\frac{3}{6}\log_2\frac{3}{6}-\frac{3}{6}\log_2\frac{3}{6})-(\frac{4}{6}*(-\frac{3}{4}\log_2\frac{3}{4}-\frac{1}{4}\log_2\frac{1}{4})-\frac{2}{6}*(-0))=1-0.54=0.46$

2 IV信息熵： $-\frac{4}{6}\log_2\frac{4}{6}-\frac{2}{6}\log_2\frac{2}{6}=0.92$

3 信息增益率：信息增益/信息熵=0.46/0.92=0.5



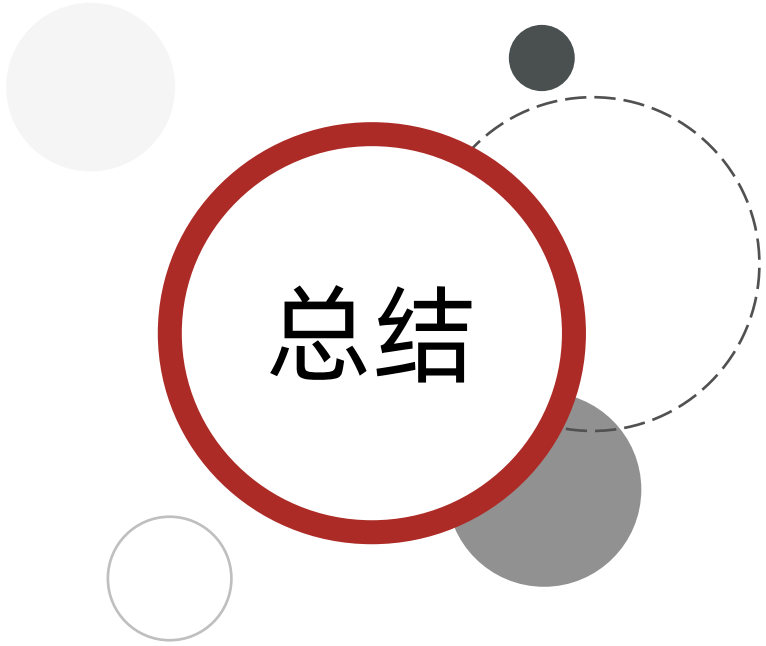
- 特征b的信息增益率：

1 信息增益： $-\frac{3}{6}\log_2\frac{3}{6}-\frac{3}{6}\log_2\frac{3}{6}-6*0=1$

2 IV信息熵： $-\frac{1}{6}\log_2\frac{1}{6}*6=2.58$

3 信息增益率：信息增益/信息熵=1/2.58=0.39

- 结论：特征a的信息增益率大于特征b的信息增益率，根据信息增益率，应该选择特征a作为分裂特征



1 信息增益率的作用

- 信息增益偏向于选择种类多的特征作为分裂依据
- 缓解ID3树中存在的不足

2 信息增益率

- 信息增益率 = 信息增益 /特征熵
- 相当于对信息增益进行修正，增加一个惩罚系数

练习

1、下列关于信息增益率的说法错误的是？

A 能有效缓解信息增益所带来的弊端

B 是在信息增益的基础上除以当前特征的固有值

C 是C4.5算法中的核心

D 倾向于选择取值多的特征

答案：D

2、在机器学习中，信息增益率是用来衡量什么的？

A 属性对分类的贡献

B 分类的准确度

C 样本的数量

D 计算机存储空间的使用率

答案：A

目录

Contents

学习目标

Learning Objectives

1. 理解基尼指数的作用
2. 知道cart构建的特征选择方法
3. 知道分类决策树的特点

CART决策树（Classification and Regression Tree）

Cart模型是一种决策树模型，它即可以用于分类，也可以用于回归。

Cart回归树使用平方误差最小化策略，

Cart分类生成树采用的基尼指数最小化策略。

CART分类树

基尼值Gini（D）：从数据集D中随机抽取两个样本，其类别标记不一致的概率。
Gini（D）值越小，数据集D的纯度越高。

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

基尼指数Gini_index（D）：选择使划分后基尼系数最小的属性作为最优化分属性。

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

注意：

1.信息增益（ID3）、信息增益率值越大（C4.5），则说明优先选择该特征。

2.基尼指数值越小（CART），则说明优先选择该特征。

CART分类树

- 已知：是否拖欠贷款数据。
- 需求：计算各特征的基尼指数，选择最优分裂点

序号	是否有房	婚姻状况	年收入(K)	是否拖欠贷款
1	yes	single	125	no
2	no	married	100	no
3	no	single	70	no
4	yes	married	120	no
5	no	divorced	95	yes
6	no	married	60	no
7	yes	divorced	220	no
8	no	single	85	yes
9	no	married	75	no
10	no	Single	90	Yes

CART决策树

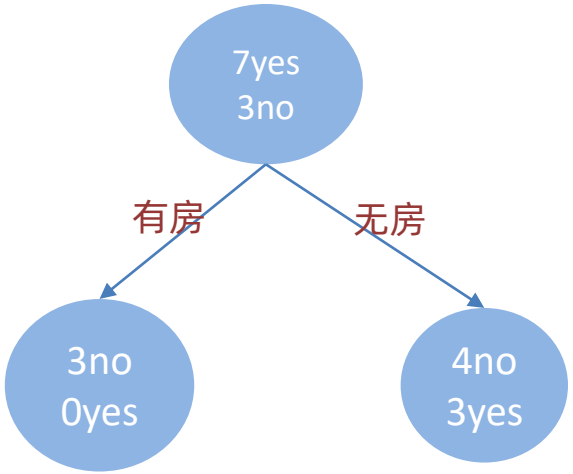
- **是否有房**

有房子的基尼值： **有房子有 1、4、7 共计三个样本，对应：3个no、0个yes**

$$Gini(\text{是否有房，yes}) = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2 = 0$$

无房子的基尼值：**无房子有 2、3、5、6、8、9、10 共七个样本，对应：4个no、3个yes**

$$\text{Gini}(\text{是否有房，no}) = 1 - \left(\frac{3}{7}\right)^2 - \left(\frac{4}{7}\right)^2 = 0.4898$$



基尼系数：**第一部分样本占了总样本的 3/10、第二部分样本占了总样本的 7/10**

$$\text{Gini\_}index(D, \text{是否有房}) = \frac{7}{10} * 0.4898 + \frac{3}{10} * 0 = 0.343$$

CART决策树 - CART树原理举例

- **婚姻状况**

1 计算 {married} 和 {single,divorced} 情况下的基尼指数

- 结婚的基尼值，有 2、4、6、9 共 4 个样本，并且对应目标值全部为 no：

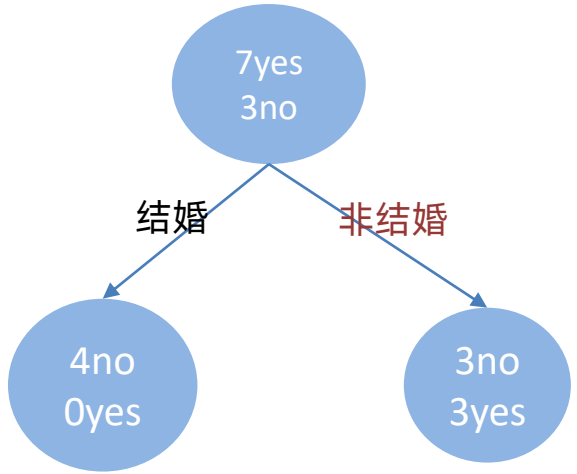
$$\text{Gini}(D,\ \text{married}) = 0$$

- 不结婚的基尼值，有 1、3、5、7、8、10 共 6 个样本，并且对应 3 个 no，3 个 yes

$$\text{Gini}(D,\ \text{single,divorced}) = 1 - \left(\frac{3}{6}\right)^2 - \left(\frac{3}{6}\right)^2 = 0.5$$

- 以 married 作为分裂点的基尼指数：

$$\text{Gini\_index}(D,\ \text{married}) = \frac{4}{10} * 0 + \frac{6}{10} * \left[1 - \left(\frac{3}{6}\right)^2 - \left(\frac{3}{6}\right)^2\right] = 0.3$$



2 计算 {single} | {married,divorced} 情况下基尼指数

$$\text{Gini\_index}(D,\ \text{婚姻状况}) = \frac{4}{10} * 0.5 + \frac{6}{10} * \left[1 - \left(\frac{1}{6}\right)^2 - \left(\frac{5}{6}\right)^2\right] = 0.367$$

3 计算 {divorced} | {single,married} 情况下基尼指数

$$\text{Gini\_index}(D,\ \text{婚姻状况}) = \frac{2}{10} * 0.5 + \frac{8}{10} * \left[1 - \left(\frac{2}{8}\right)^2 - \left(\frac{6}{8}\right)^2\right] = 0.4$$

最终：该特征的基尼值为 0.3，并且预选分裂点为：{married} 和 {single,divorced}

CART决策树 - CART树原理举例

- **年收入**

1 先将数值型属性升序排列，以相邻中间值作为待确定分裂点：

是否拖欠贷款	no	no	no	yes	yes	yes	no	no	no	no
年收入	60	70	75	85	90	95	100	120	125	220
相邻值中点	/	65	72.5	80	87.7	92.5	97.5	110	122.5	172.5	/

待确定的分裂点为：65、72.5、80、87.7、92.5、97.5、110、122.5、172.5

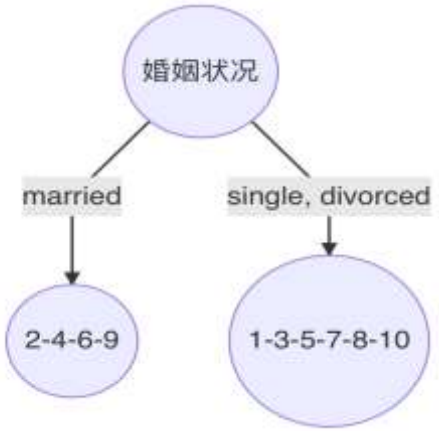
2 以年收入 65 将样本分为两部分，计算基尼指数

$$\text{节点为 } 65 \text{ 时 }:\{\text{年收入}\} = \frac{1}{10} * 0 - \frac{9}{10} * \left[1 - \left(\frac{6}{9}\right)^2 - \left(\frac{3}{9}\right)^2\right] = 0.4$$

3 以此类推计算所有分割点的基尼指数，最小的基尼指数为 0.3

是否拖欠贷款	no	no	no	yes	yes	yes	no	no	no	no
年收入	60	70	75	85	90	95	100	120	125	220
相邻值中点	/	65	72.5	80	87.7	92.5	97.5	110	122.5	172.5	/
Gini_index	/	0.4	0.375	0.343	0.417	0.4	0.3	0.343	0.375	0.4	/

CART决策树 - CART树原理举例



第1轮结果

以是否有房作为分裂点的基尼指数为：0.343

以婚姻状况为分裂特征、以 married 作为分裂点的基尼指数为：0.3

以年收入作为分裂特征、以 97.5 作为分裂点的的基尼指数为：0.3

第2轮

1 样本 2、4、6、9 样本的类别都是 no，已经达到最大纯度

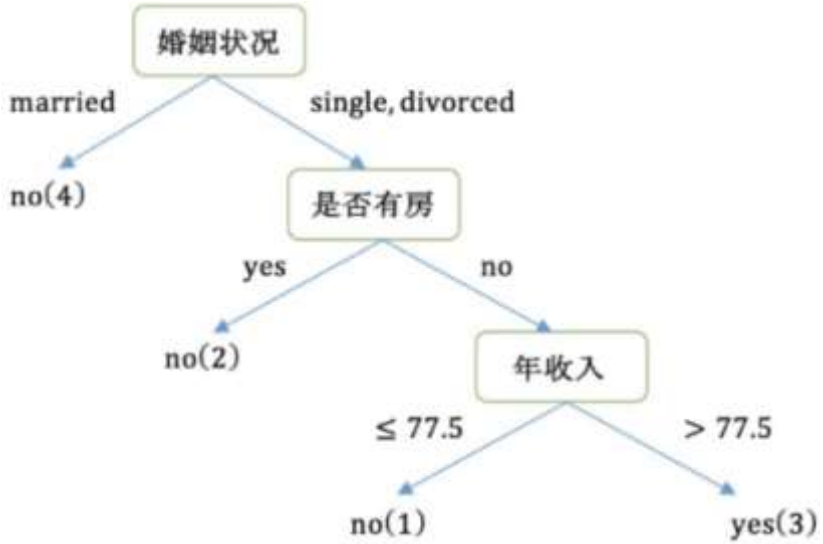
所以，该节点不需要再继续分裂。

2 样本 1、3、5、7、8、10 样本中仍然包含 4 个 no，2 个 yes

该节点并未达到要求的纯度，需要继续划分。

3 右子树的数据集变为： 1、3、5、7、8、10，在该数据集中计算

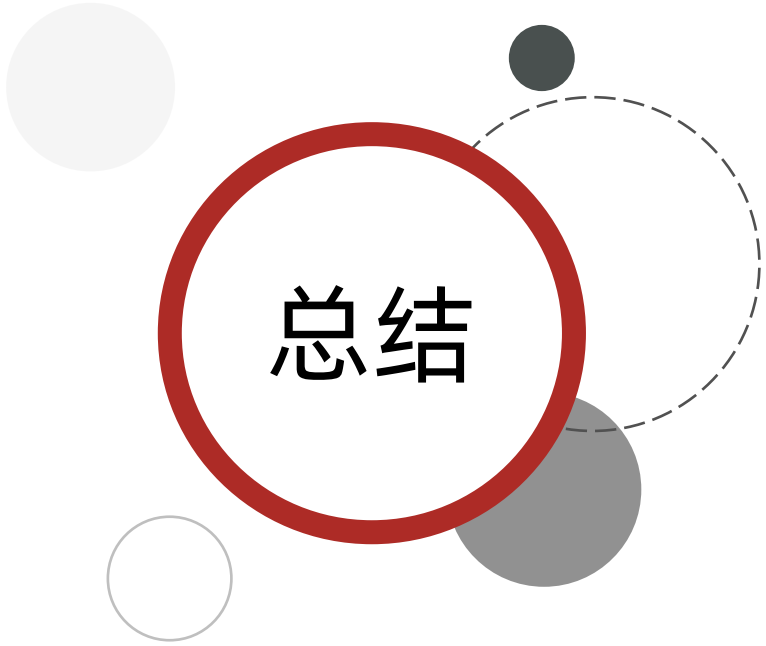
不同特征的基尼指数，选择基尼指数最小的特征继续分裂。



重复上述过程，直到构建完成整个决策树

三种分类树的对比

名称	提出时间	分支方式	特点
ID3	1975	信息增益	1.ID3只能对离散属性的数据集构成决策树 2.倾向于选择取值较多的属性
C4.5	1993	信息增益率	1.缓解了ID3分支过程中总喜欢偏向选择值较多的属性 2.可处理连续数值型属性，也增加了对缺失值的处理方法 3.只适合于能够驻留于内存的数据集,大数据集无能为力
CART	1984	基尼指数	1.可以进行分类和回归，可处理离散属性，也可以处理连续属性 2.采用基尼指数，计算量减小 3.一定是二叉树



1. CART决策树的作用？

   分类和回归

2. 基尼指数的作用？

   特征筛选，基尼指数值越小，则说明优先选择该特征。

练习

1、下列关于cart树的说法正确的是？

A) 基尼指数值越大，则说明优先选择该特征

B) 基尼指数是CART算法中用于划分属性的重要依据

C) 基尼指数的计算使用到了自然对数

D) CART算法不能用于回归场景

答案：B

目录

Contents

学习目标

Learning Objectives

1. 知道分类决策树API函数
2. 完成泰坦尼克号生存预测的案例

CART决策案例 -泰坦尼克号乘客生存预测

- **决策树API介绍**

sklearn.tree.DecisionTreeClassifier(criterion='gini', max_depth=None,random_state=None)

- Criterion： 特征选择标准

  "gini"或"entropy"，前者代表基尼系数，后者代表信息增益。默认"gini"，即CART算法

- min_samples_split：内部节点再划分所需最小样本数
- min_samples_leaf：叶子节点最少样本数
- max_depth：决策树最大深度

CART决策案例 -泰坦尼克号乘客生存预测

- **案例背景**
  - 1912年4月15日，在它的首航中，泰坦尼克号在与冰山相撞后沉没，在2224名乘客和机组人员中造成1502人死亡。
  - 造成海难失事的原因之一是乘客和机组人员没有足够的救生艇。尽管幸存生存有一些运气因素，但有些人比其他人更容易生存，例如妇女，儿童和上流社会。
  - 有了遇难和幸运数据，运用机器学习工具来预测哪些乘客可幸免于悲剧。

- **数据情况**
  - 数据集中的特征包括票的类别，是否存活，乘坐班次，年龄，登陆，home.dest，房间，船和性别等
  - 乘坐班是指乘客班（1，2，3），是社会经济阶层的代表
  - age数据存在缺失

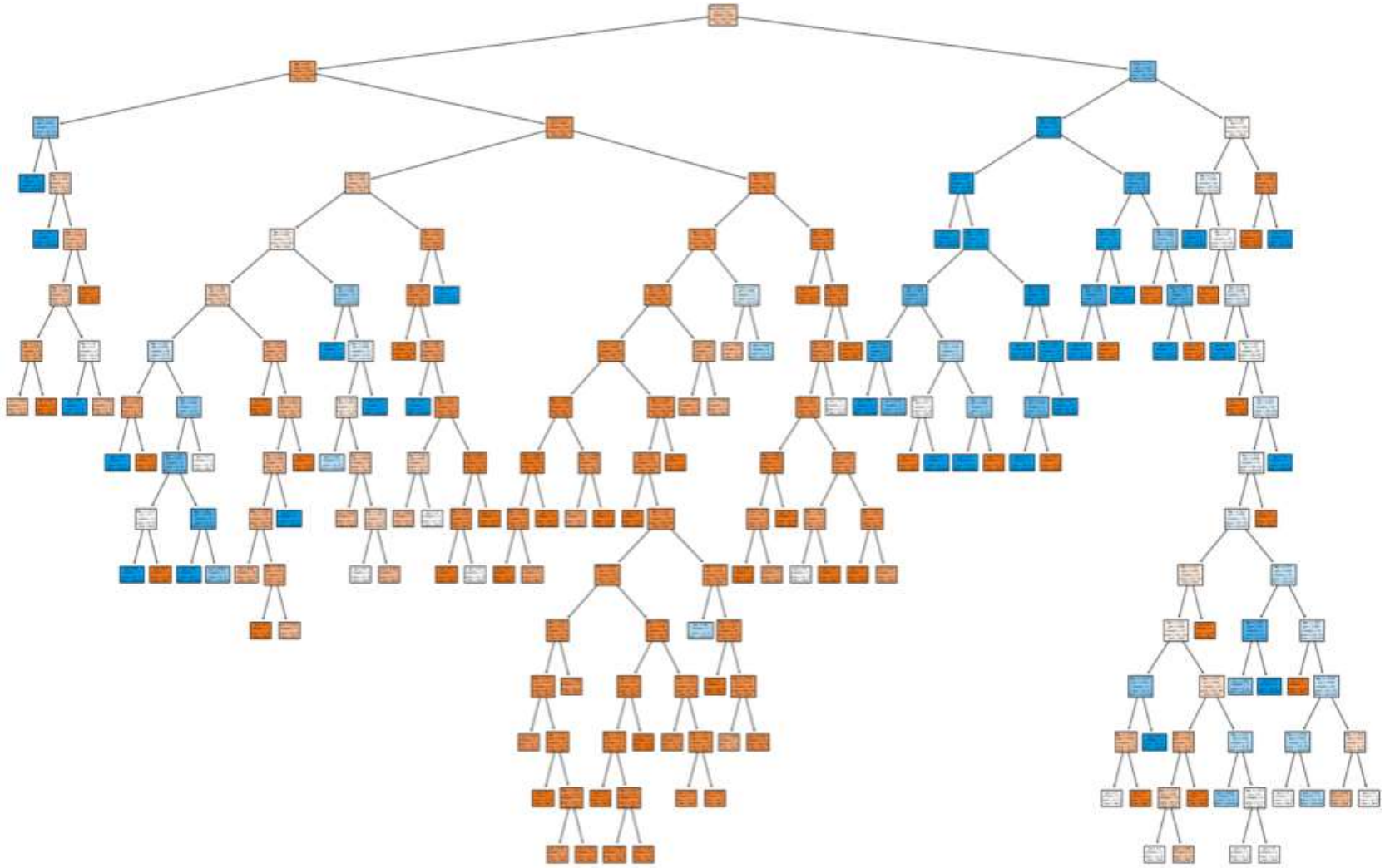
	A	B	C	D	E	F	G	H	I	J	K	L
1	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
2	1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
3	2	1	1	, Mrs. John Bradley (Florence Brigg	female	38	1	0	PC 17599	71.2833	C85	C
4	3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 310128	7.925		S
5	4	1	1	elle, Mrs. Jacques Heath (Lily May	female	35	1	0	113803	53.1	C123	S
6	5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		S
889	888	1	1	Graham, Miss. Margaret Edith	female	19	0	0	112053	30	B42	S
890	889	0	3	inston, Miss. Catherine Helen "Car	female		1	2	W./C. 6607	23.45		S
891	890	1	1	Behr, Mr. Karl Howell	male	26	0	0	111369	30	C148	C
892	891	0	3	Dooley, Mr. Patrick	male	32	0	0	370376	7.75		Q

CART决策案例 -泰坦尼克号乘客生存预测

```
# 1.导入依赖包
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree

def dm04_泰坦尼克():
    # 2.读数据到内存并预处理
    # 2.1 读取数据
    taitan_df = pd.read_csv("./data/titanic/train.csv")
    print(taitan_df.head())     # 查看前5条数据
    print(taitan_df.info)       # 查看特性信息
    # 2.2 数据处理，确定x y
    x = taitan_df[['Pclass', 'Age', 'Sex']]
    y = taitan_df['Survived']
    # 2.3 缺失值处理
    x['Age'].fillna(x['Age'].mean(), inplace=True)
    print('x-->1', x.head(10))
    # 2.4 pclass类别型数据,需要转数值one-hot编码
    x = pd.get_dummies(x)
    print('x-->2', x.head(10))
    # 2.5 数据集划分
    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20, random_state=33)
```

```
    # 3.训练模型,实例化决策树模型
    estimator = DecisionTreeClassifier()
    estimator.fit(x_train, y_train)

    # 4.模型预测
    y_pred = estimator.predict(x_test)

    # 5.模型评估
    # 5.1 输出预测准确率
    myret = estimator.score(x_test, y_test)
    print('myret-->\n', myret)

    # 5.2 更加详细的分类性能
    myreport = classification_report(y_pred, y_test, target_names=['died', 'survived'])
    print('myreport-->\n', myreport)

    # 5.3 决策树可视化
    plot_tree(estimator,
              max_depth=10,
              filled=True,
              feature_names=['Pclass', 'Age', 'Sex_female', 'Sex_male'],
              class_names=['died', 'survived'] )
    plt.show()
```

泰坦尼克号乘客生存预测



目录

Contents

学习目标

Learning Objectives

1. 了解回归决策树的构建原理
2. 能利用回归决策树API解决问题

CART 回归决策树

- CART 回归树和 CART 分类树的不同之处在于
  - CART 分类树预测输出的是一个离散值，CART 回归树预测输出的是一个连续值
  - CART 分类树使用基尼指数作为划分、构建树的依据，CART 回归树使用平方损失
  - 分类树使用叶子节点多数类别作为预测类别，回归树则采用叶子节点里均值作为预测输出

- CART 回归树的平方损失

$$\text{Loss}(y, f(x)) = (f(x) - y)^2$$

- 栗子：根据平方损失，构建CART 回归树
  - 已知：数据集只有 1 个特征 x, 目标值值为 y

x	1	2	3	4	5	6	7	8	9	10
y	5.56	5.7	5.91	6.4	6.8	7.05	8.9	8.7	9	9.05

  - 分析：因只有 1 个特征，所以只需选择该特征的最优划分点，并不需要计算其他特征。

CART回归决策树

◆ 1 先将特征 x 的值排序，并取相邻元素均值作为待划分点，如下图所示：

s	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5

◆ 2 计算每一个划分点的平方损失，例如：划分点1.5 的平方损失计算过程为：

1. R1 为 小于 1.5 的样本个数，样本数量为：1，其输出值为：5.56

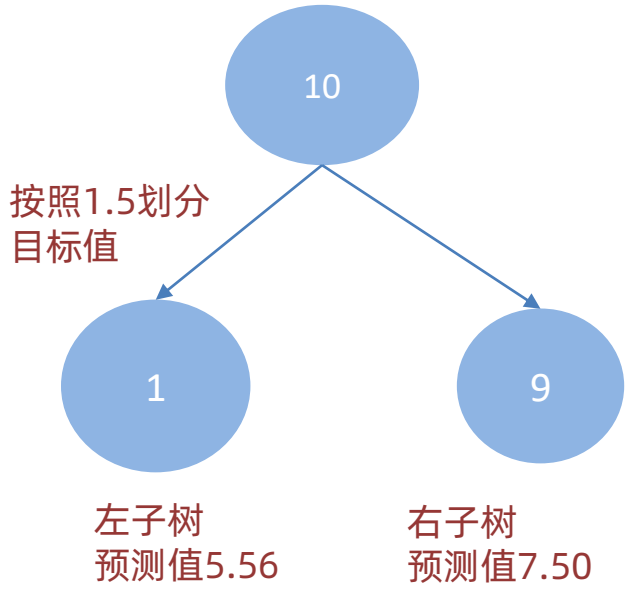
$$R1 = 5.56$$

2. R2 为 大于 1.5 的样本个数，样本数量为：9，其输出值为：

$$R2 = (5.7 + 5.91 + 6.4 + 6.8 + 7.05 + 8.9 + 8.7 + 9 + 9.05)/9 = 7.50$$

3. 该划分点的平方损失：

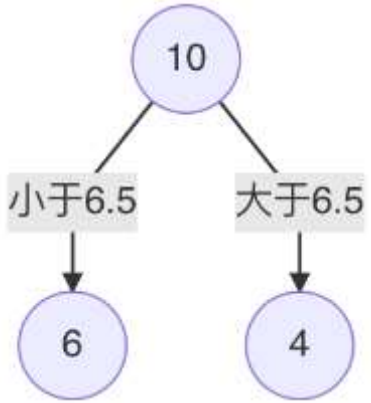
$$L(1.5) = (5.56 - 5.56)^2 + [(5.7 - 7.5)^2 + (5.91 - 7.5)^2 + \ldots + (9.05 - 7.5)^2] = 0 + 15.72 = 15.72$$



CART 回归决策树

◆ 3 以此方式计算 2.5、3.5... 等划分点的平方损失，结果如下所示：

s	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5
m(s)	15.72	12.07	8.36	5.78	3.91	**1.93**	8.01	11.73	15.74



◆ 4 当划分点 s=6.5时,m(s) 最小。所以第1个划分变量：特征为 X, 切分点为 6.5

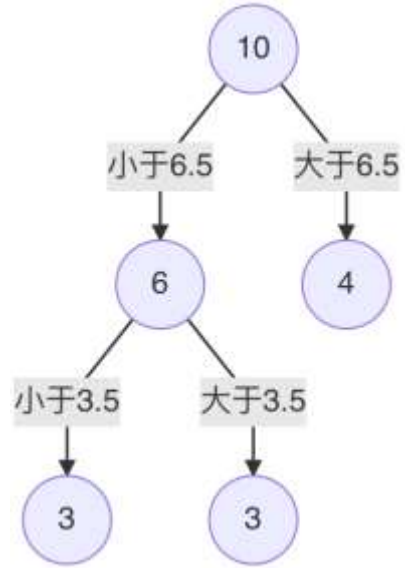
◆ 5 对左子树的 6 个节点计算每个划分点的平方式损失，找出最优划分点

x	1	2	3	4	5	6
y	5.56	5.7	5.91	6.4	6.8	7.05

s	1.5	2.5	3.5	4.5	5.5
c1	5.56	5.63	5.72	5.89	6.07
c2	6.37	6.54	6.75	6.93	7.05

eg：以x=1.5作为切分点，左子树c1输出为5.56，右子树c2输出 (5.7+5.91+6.4+6.8+7.05)/5 = 6.37

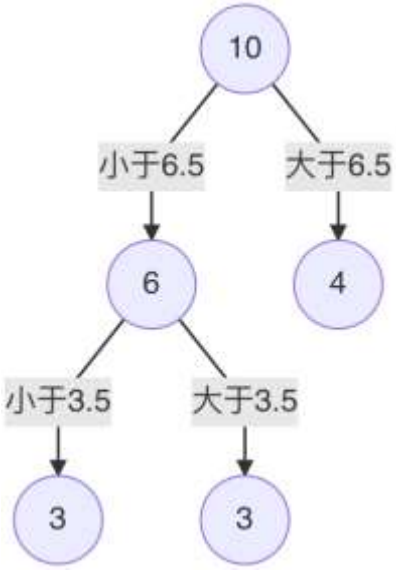
s	1.5	2.5	3.5	4.5	5.5
m(s)	1.3087	0.754	0.2771	0.4368	1.0644



◆ 6 s=3.5时，m(s) 最小，所以左子树继续以 3.5 进行分裂

CART回归决策树

- **7 假设**在生成**3个区域**之后停止划分，以上就是最终回归树。
  - 每一个叶子节点的输出为：挂在该节点上的所有样本均值。



CART 回归树构建过程小结

- 1 选择一个特征，将该特征的值进行排序，取相邻点计算均值作为**待划分点**
- 2 根据所有划分点，将数据集分成两部分：R1、R2
- 3 R1 和 R2 两部分的平方损失相加作为该切分点平方损失
- 4 取最小的平方损失的划分点，作为当前特征的划分点
- 5 以此计算其他特征的最优划分点、以及该划分点对应的损失值
- 6 在所有的特征的划分点中，选择出最小平方损失的划分点，作为当前树的分裂点

CART决策树 - 案例：线性回归与回归决策树对比

- 已知数据

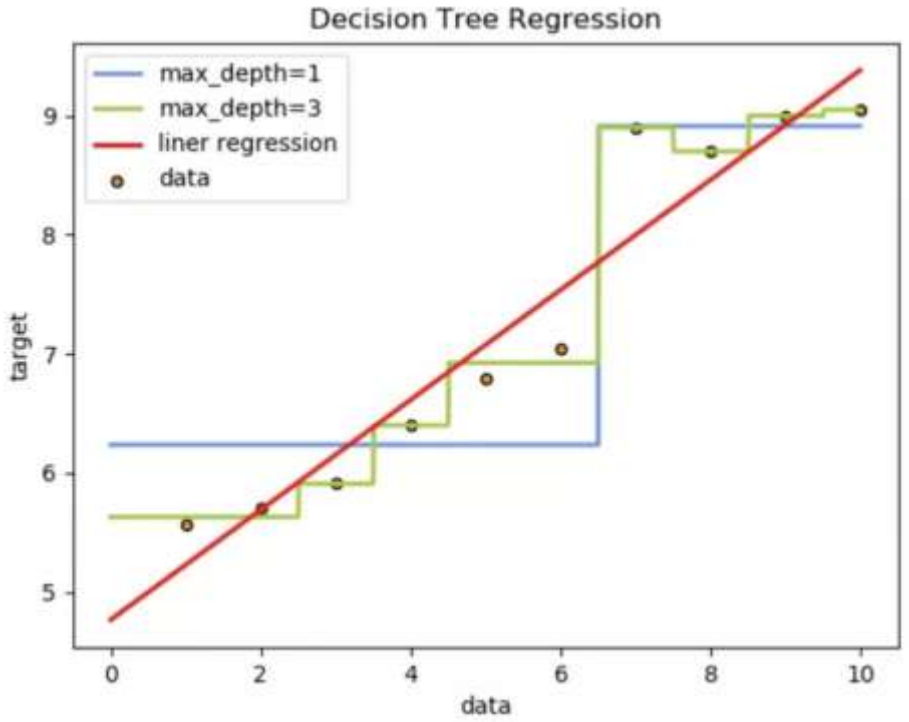
x	1	2	3	4	5	6	7	8	9	10
y	5.56	5.7	5.91	6.4	6.8	7.05	8.9	8.7	9	9.05

- 需求

分别训练线性回归、回归决策树模型，并预测对比

- 分析

训练模型，并使用1000个[0.0, 10]之间的数据，让模型预测，画出预测值图线



从预测效果来看：

1、线性回归是一条直线

2、决策树是曲线

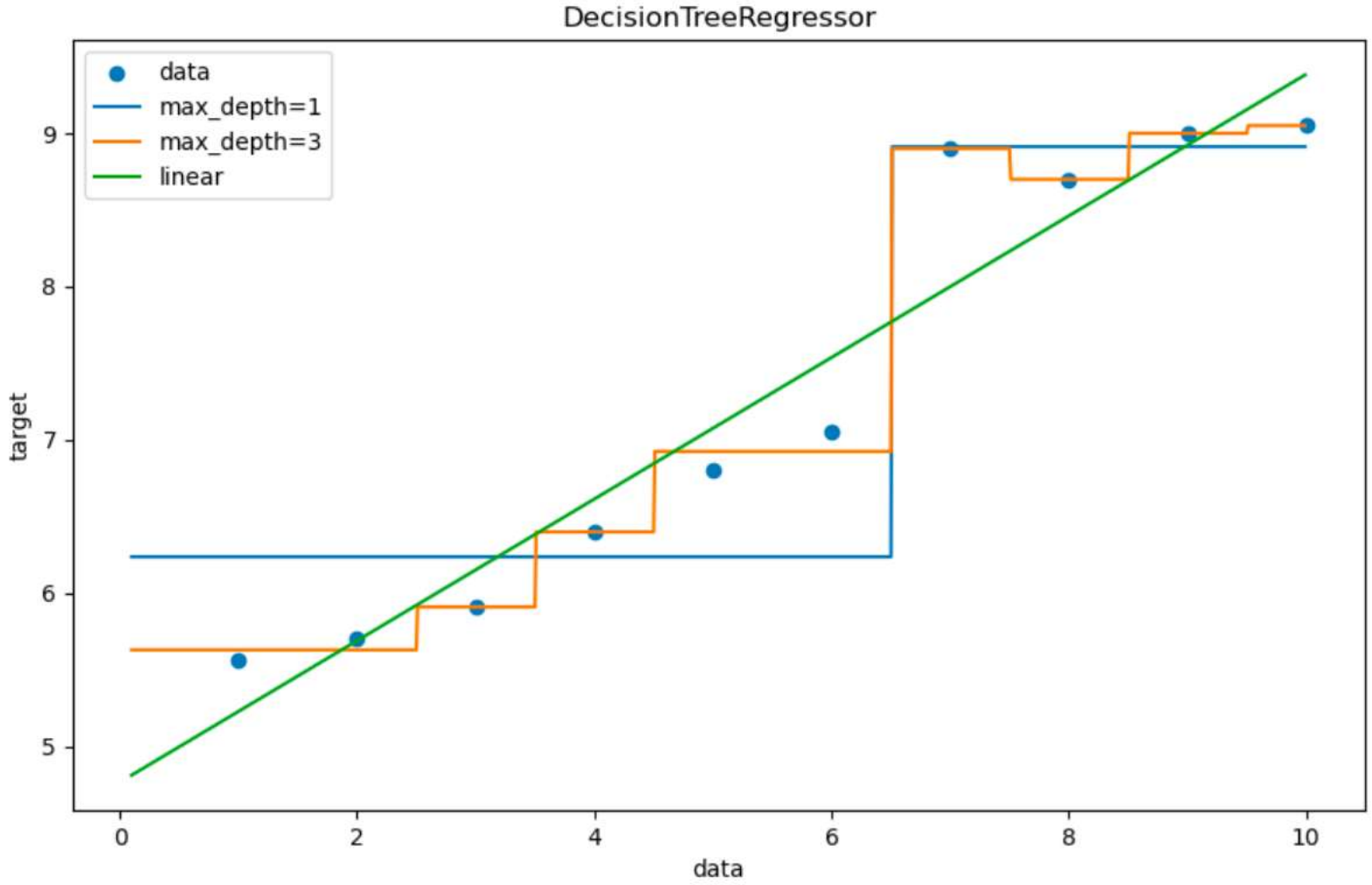
3、树的拟合能力是很强的，易过拟合

案例：线性回归与回归决策树对比

```
# 1.导入依赖包
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeRegressor  # 回归决策树
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

def dm01_回归分类():
    # 2.准备数据
    x = np.array(list(range(1,11))).reshape(-1, 1)
    y = np.array([5.56, 5.70, 5.91, 6.40, 6.80, 7.05, 8.90, 8.70, 9.00, 9.05])
    print('x-->', x)
    print('y-->', y)

    # 3.模型训练，实例化模型
    model1 = DecisionTreeRegressor(max_depth=1)
    model2 = DecisionTreeRegressor(max_depth=3)
    model3 = LinearRegression()
    model1.fit(x, y)
    model2.fit(x, y)
    model3.fit(x, y)
```

```
    # 4.模型预测# 等差数组-按照间隔
    x_test = np.arange(0.0, 10.0, 0.01).reshape(-1, 1)
    y_pre1 = model1.predict(x_test)
    y_pre2 = model2.predict(x_test)
    y_pre3 = model3.predict(x_test)
    print(y_pre1.shape, y_pre2.shape, y_pre3.shape)

    # 5.结果可视化
    plt.figure(figsize=(10, 6), dpi=100)
    plt.scatter(x, y, label='data')

    plt.plot(x_test, y_pre1, label='max_depth=1')  # 深度1层
    plt.plot(x_test, y_pre2, label='max_depth=3')  # 深度3层
    plt.plot(x_test, y_pre3, label='linear')
    plt.xlabel('data')
    plt.ylabel('target')
    plt.title('DecisionTreeRegressor')
    plt.legend()
    plt.show()
```

案例：线性回归与回归决策树对比



练习

1.以下哪项是回归决策树说法正确的是（多选）？

A. 可以处理非线性关系

B. 可以处理多个输入变量

C. 可以处理分类和回归问题

D. 可以避免过拟合

答案为：AB

2.在回归决策树中，如何选择最佳的分割点？

A. 根据信息增益选择

B. 根据基尼指数选择

C. 根据均方误差选择

D. 根据叶节点样本数量选择

答案为：C

目录

Contents

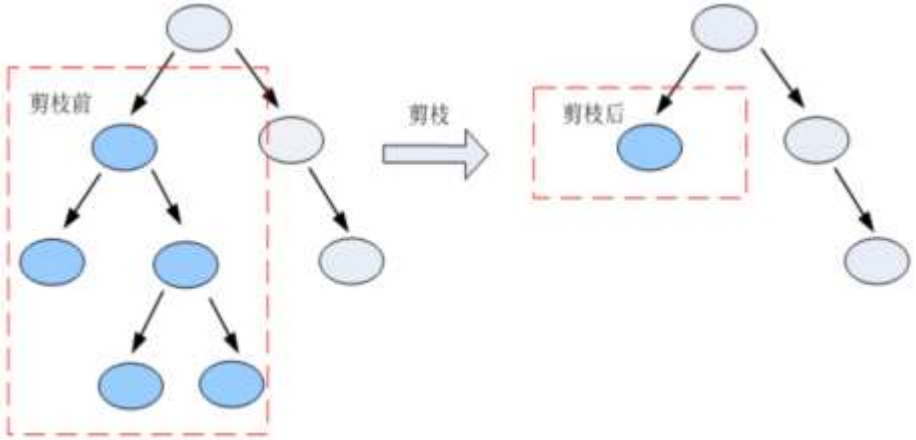
学习目标

Learning Objectives

1. 知道什么是剪枝
2. 理解剪枝的作用
3. 知道常用剪枝方法
4. 了解不同剪枝方法的优缺点

决策树正则化 - 剪枝

- **为什么要剪枝？**
  - 决策树剪枝是一种防止决策树过拟合的一种正则化方法；提高其泛化能力。
- **剪枝**
  - 把子树的节点全部删掉，使用用叶子节点来替换



- **剪枝方法**

1.预剪枝

指在决策树生成过程中，对每个节点在划分前先进行估计，若当前节点的划分不能带来决策树泛化性能提升，则停止划分并将当前节点标记为叶节点;

2.后剪枝

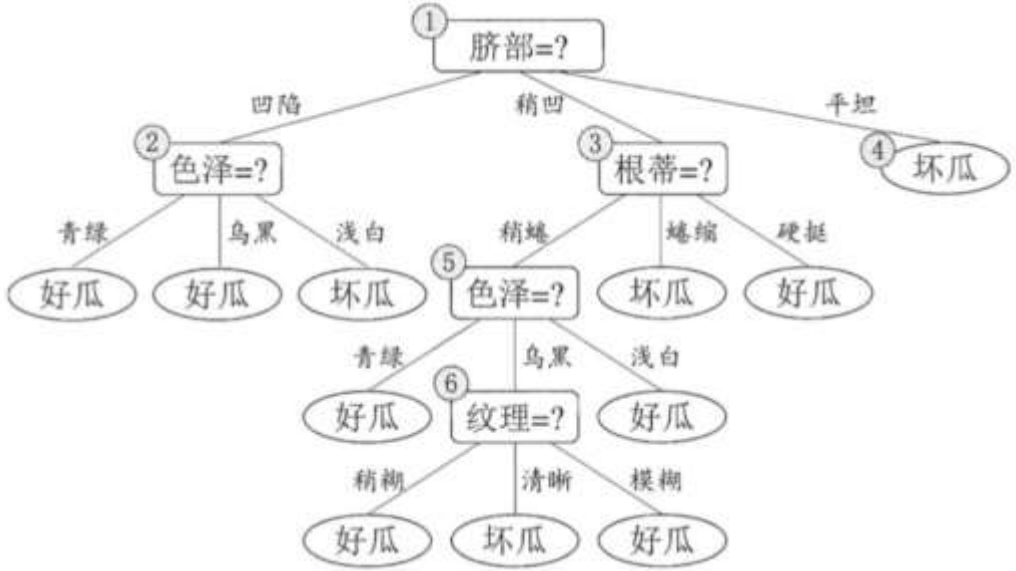
是先从训练集生成一棵完整的决策树，然后自底向上地对非叶节点进行考察，若将该节点对应的子树替换为叶节点能带来决策树泛化性能提升，则将该子树替换为叶节点。

剪枝思想

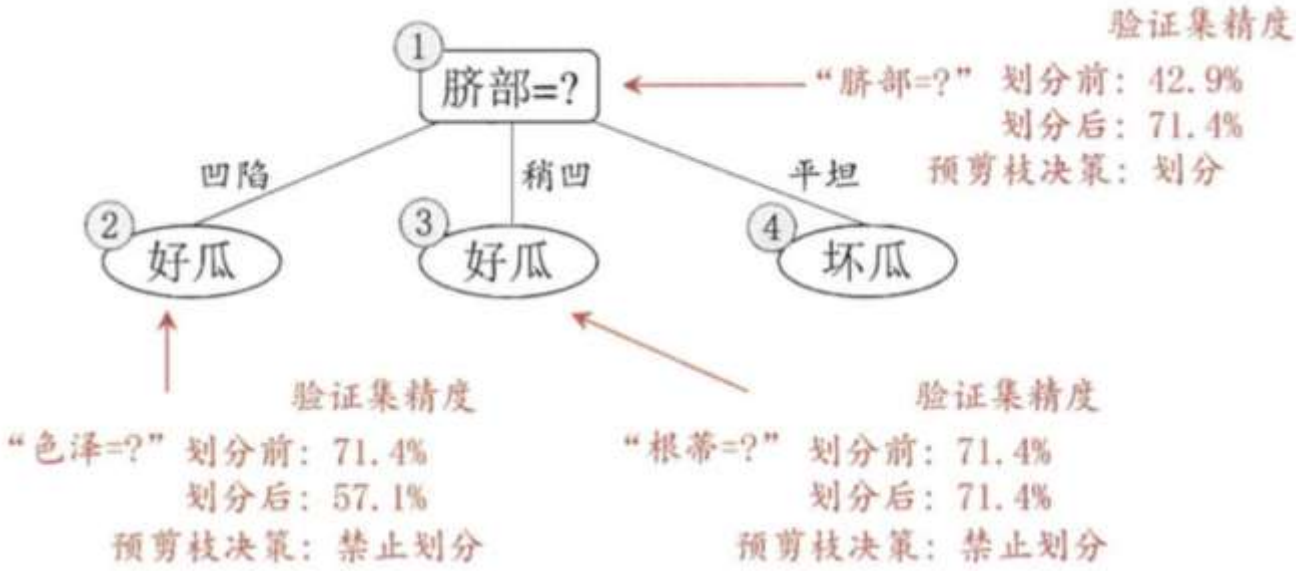
- **已知 训练集和验证集**

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否

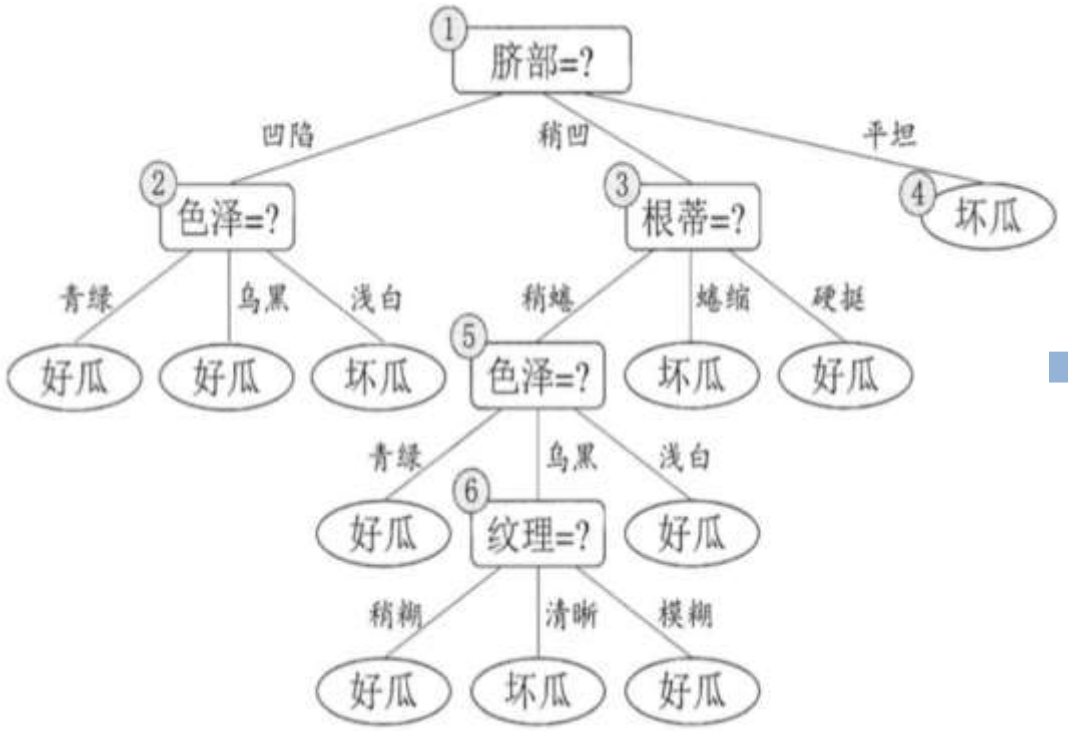


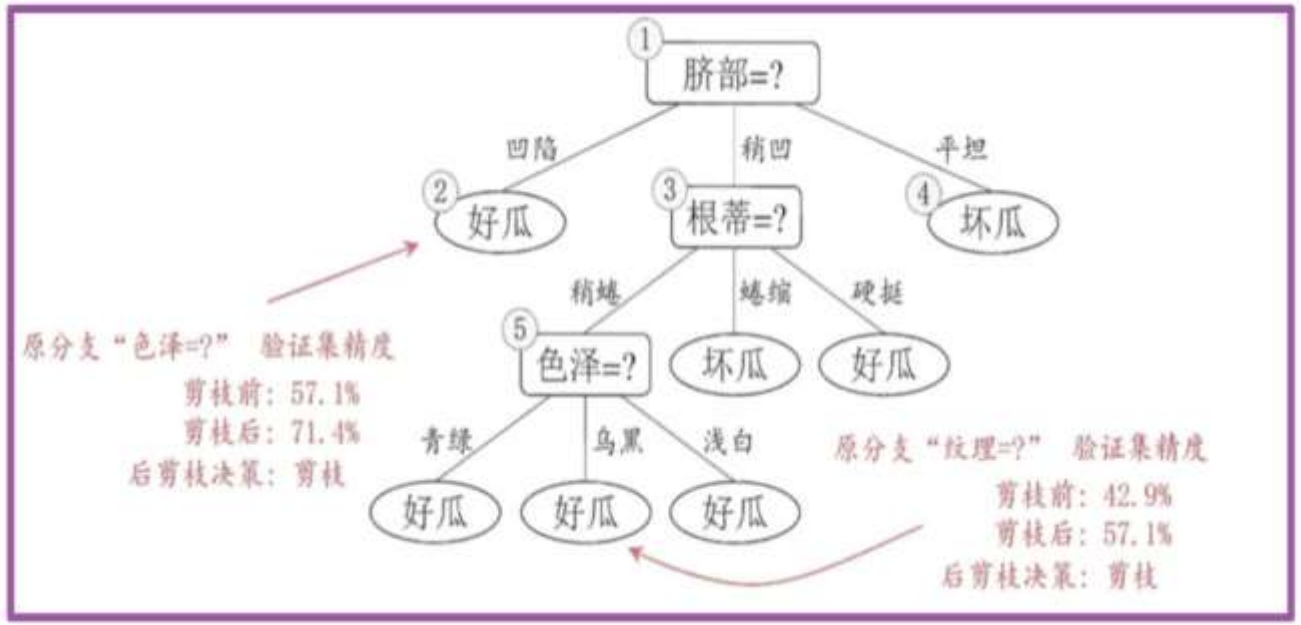
预剪枝



基于预剪枝策略从上表数据所生成的决策树如上图所示，其验证集精度为 71.4%.

决策树正则化 - 剪枝的基本工作原理 - 后剪枝





先利用训练集完整的生成一颗树，有6个内部节点。分别考察这些节点作为叶子节点模型的准确率，若准确率上升，则剪掉，否则保留。

决策树正则化 - 剪枝技术对比

预剪枝

优点

预剪枝使决策树的很多分支没有展开，不单降低了过拟合风险，还显著减少了决策树的训练、测试时间开销

缺点

有些分支的当前划分虽不能提升泛化性能，但后续划分却有可能导致性能的显著提高；预剪枝决策树也带来了欠拟合的风险

后剪枝

优点

比预剪枝保留了更多的分支。一般情况下，后剪枝决策树的欠拟合风险很小，泛化性能往往优于预剪枝

缺点

后剪枝先生成，后剪枝。自底向上地对树中所有非叶子节点进行逐一考察，训练时间开销比未剪枝的决策树和预剪枝的决策树都要大得多。

总结

1 剪枝？

把叶子节点、子节点删掉，用更大的叶子节点（子树）替换叶子节点

2 预剪枝和后剪枝？

1.预剪枝

指在决策树生成过程中，对每个节点在划分前先进行估计，若当前节点的划分不能带来决策树泛化性能提升，则停止划分并将当前节点标记为叶节点;

2.后剪枝

是先从训练集生成一棵完整的决策树，然后自底向上地对非叶节点进行考察，若将该节点对应的子树替换为叶节点能带来决策树泛化性能提升，则将该子树替换为叶节点。

练习

1、下列关于剪枝的描述正确的是？（多选）

A）剪枝是为了防止模型产生过拟合

B）常用的剪枝方法有预剪枝和后剪枝

C）预剪枝是提前设定树在构建过程中的限制参数，一边生成一边验证效果

D）后剪枝需要等树构建完成后再遍历节点

答案解析：A剪枝正则化一种方法ok B正确 C正确 D描述正确

答案：ABCD


黑马程序员
www.itheima.com
传智教育旗下高端IT教育品牌