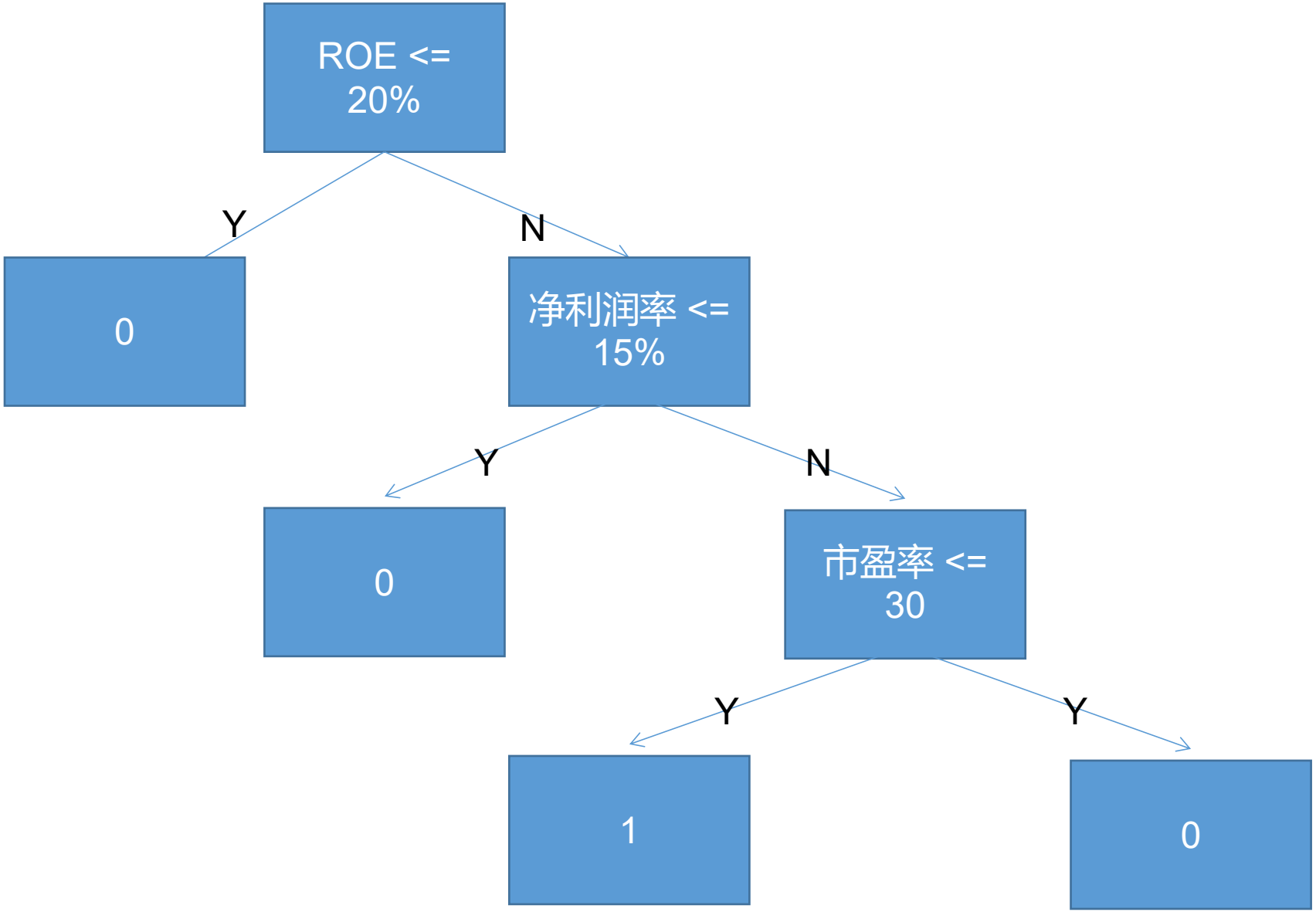
决策树 （Decision Tree）

使用决策树进行决策



通过学习生成决策树

财务质量	市值	周期性	性质	表现
差	大市值	周期	国有	上涨
中	小市值	非周期	民营	上涨
中	大市值	非周期	国有	上涨
好	大市值	周期	国有	上涨
差	大市值	非周期	民营	上涨
差	中市值	非周期	民营	上涨
中	中市值	非周期	民营	上涨
中	大市值	非周期	国有	上涨
好	小市值	周期	国有	下跌
差	中市值	周期	国有	下跌
好	小市值	周期	民营	下跌
中	中市值	周期	国有	下跌
好	小市值	非周期	民营	下跌
差	小市值	周期	民营	下跌

信息熵 （Entropy）

信息熵

$$Ent = -\sum p_i \log(p_i)$$

衡量随机变量的不确定性

随机变量 X 代表小明考试是否及格（0：不及格；1：及格）

如果小明学习成绩很差，考试几乎不可能及格，则 Ent = ？

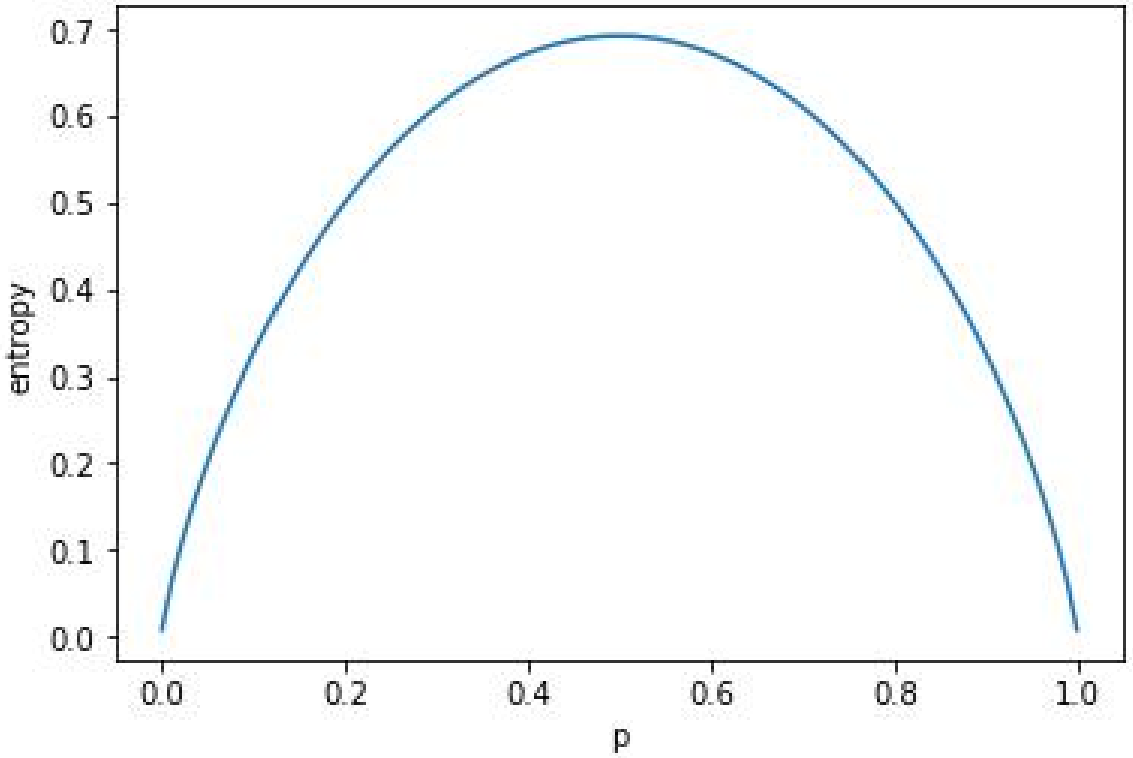
Ent = - 0log0 - 1log1 = 0

如果小明学习成绩一般，及格可能性为50%，则 Ent = ？

Ent = - （1/2）log（1/2）- （1/2）log（1/2） = 0.6931

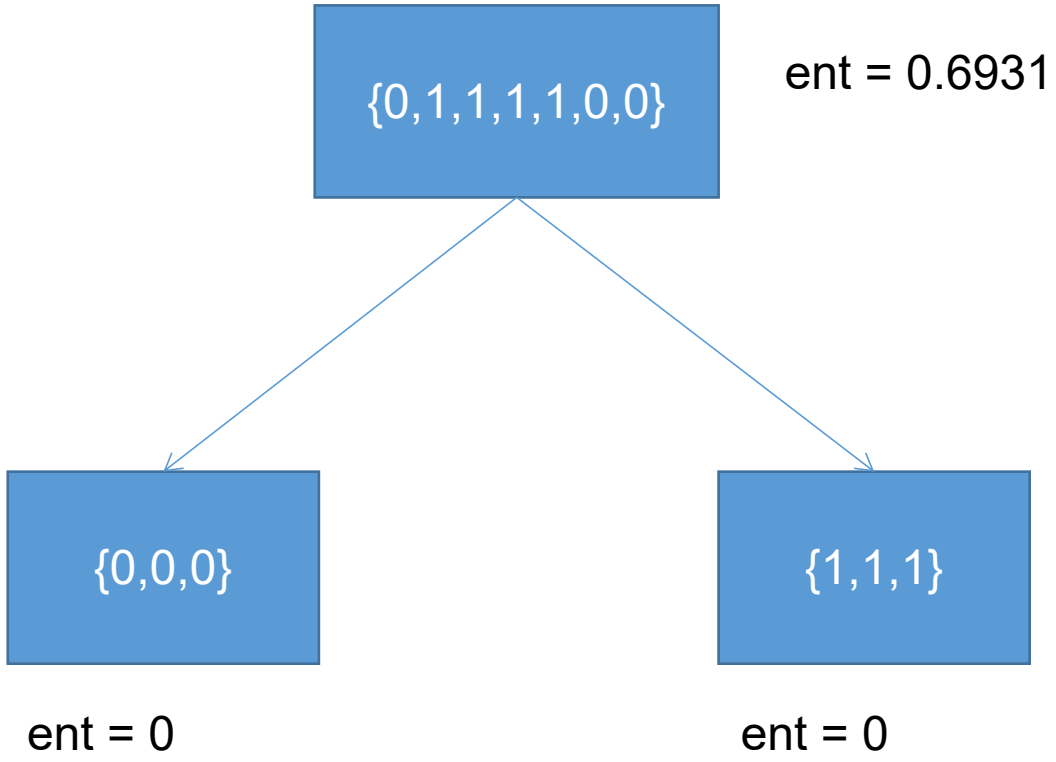
信息熵

X ~ Bernoulli (p)

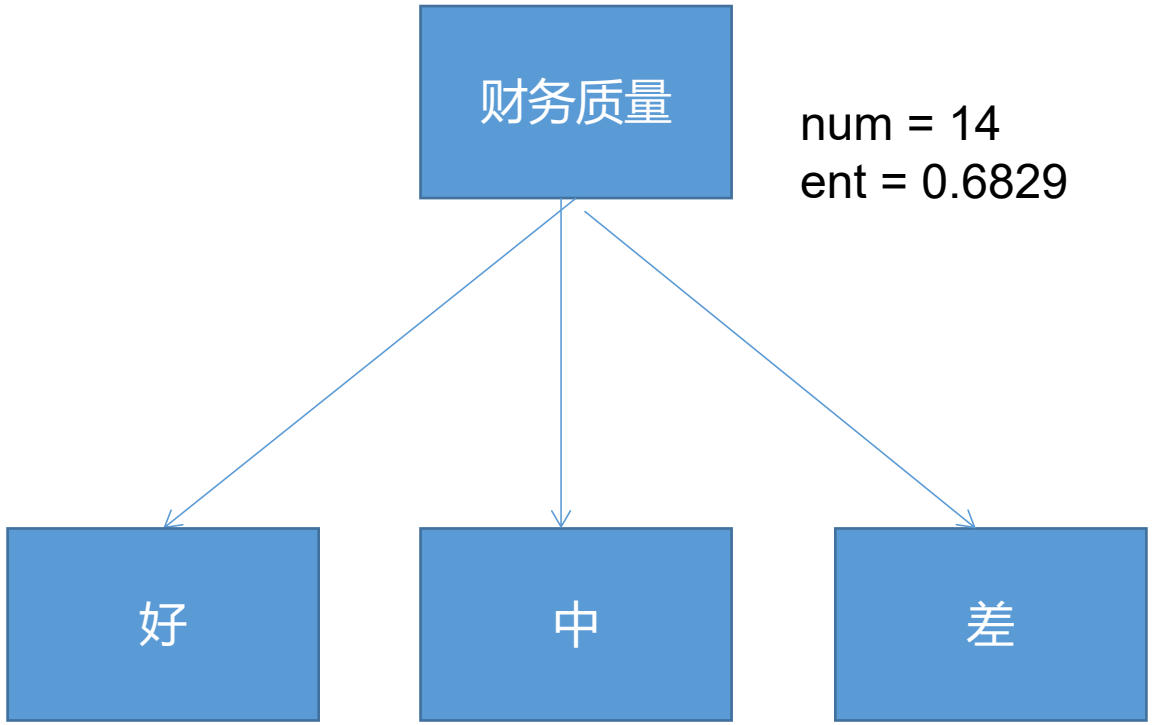


生成决策树

最优划分



信息增益



$$Ent(D) = 0.6829$$

$$\sum \frac{|D_i|}{|D|} Ent(D_i)$$

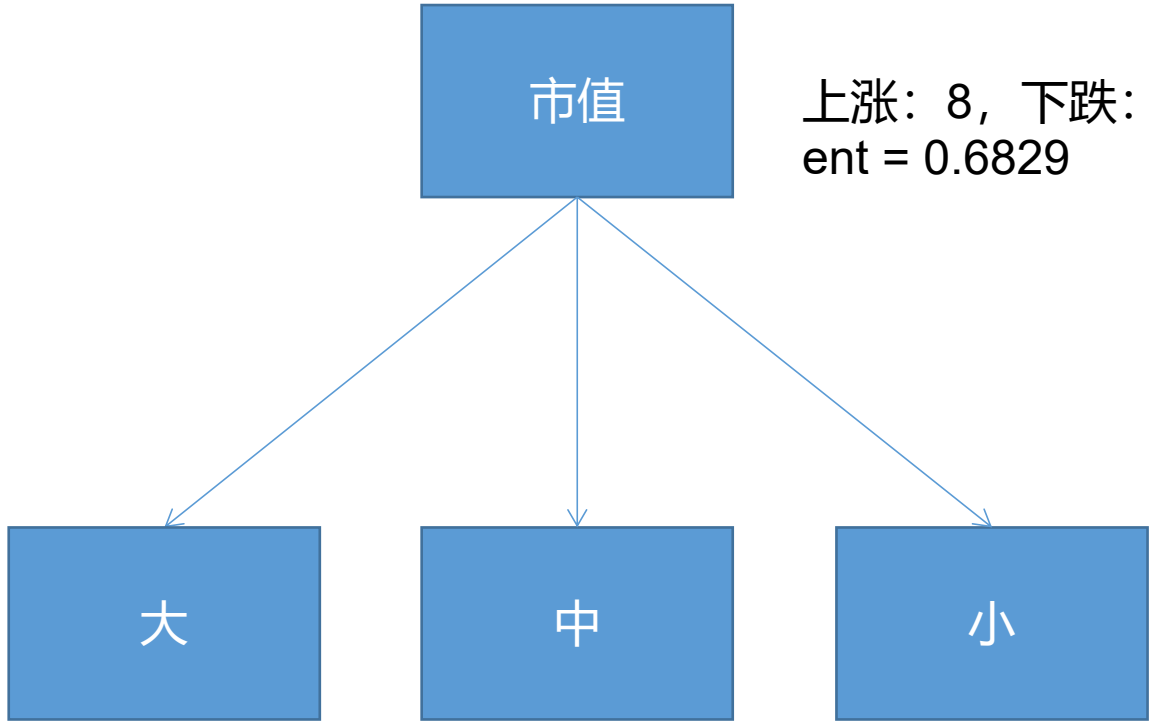
$$= \frac{4}{14} \times 0.5623 + \frac{5}{14} \times 0.5004 + \frac{5}{14} \times 0.6730$$

$$= 0.5797$$

$$Gain(D, \text{"财务质量"}) = 0.6829 - 0.5797 = 0.1032$$

最优划分属性

Gain(D, “财务质量”) = 0.6829 - 0.5797 = 0.1032
Gain(D, “市值”) = 0.6829 - 0.3767 = 0.3062
Gain(D, “周期性”) = 0.6829 - 0.5041 = 0.1788
Gain(D, “性质”) = 0.6829 - 0.6829 = 0

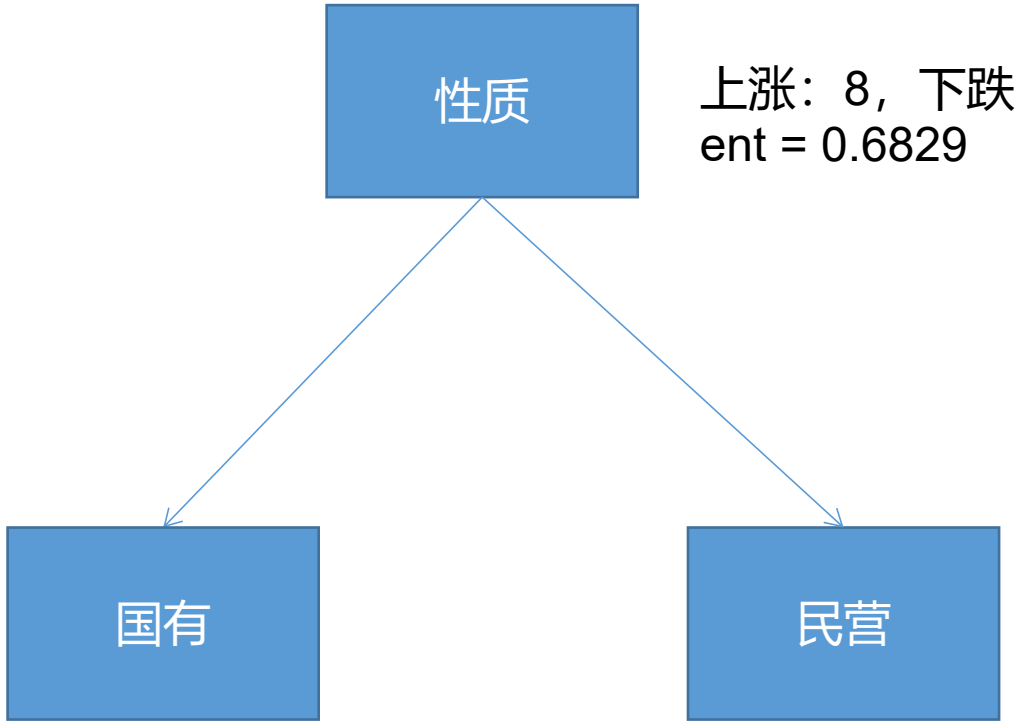


上涨：8，下跌：6
ent = 0.6829

上涨：5, 下跌：0
ent = 0

上涨：2, 下跌：2
ent = 0.6931

上涨：1, 下跌：4
ent = 0.5004

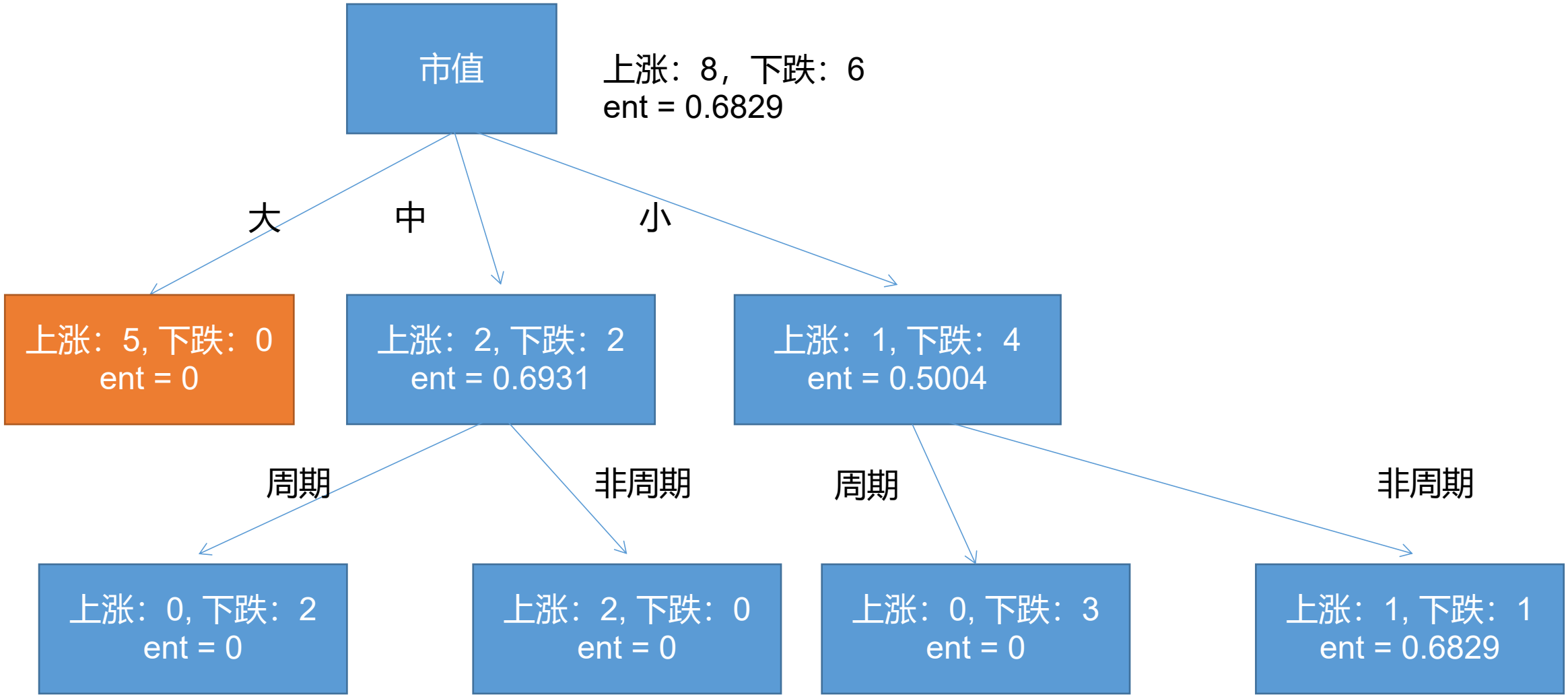


上涨：8，下跌：6
ent = 0.6829

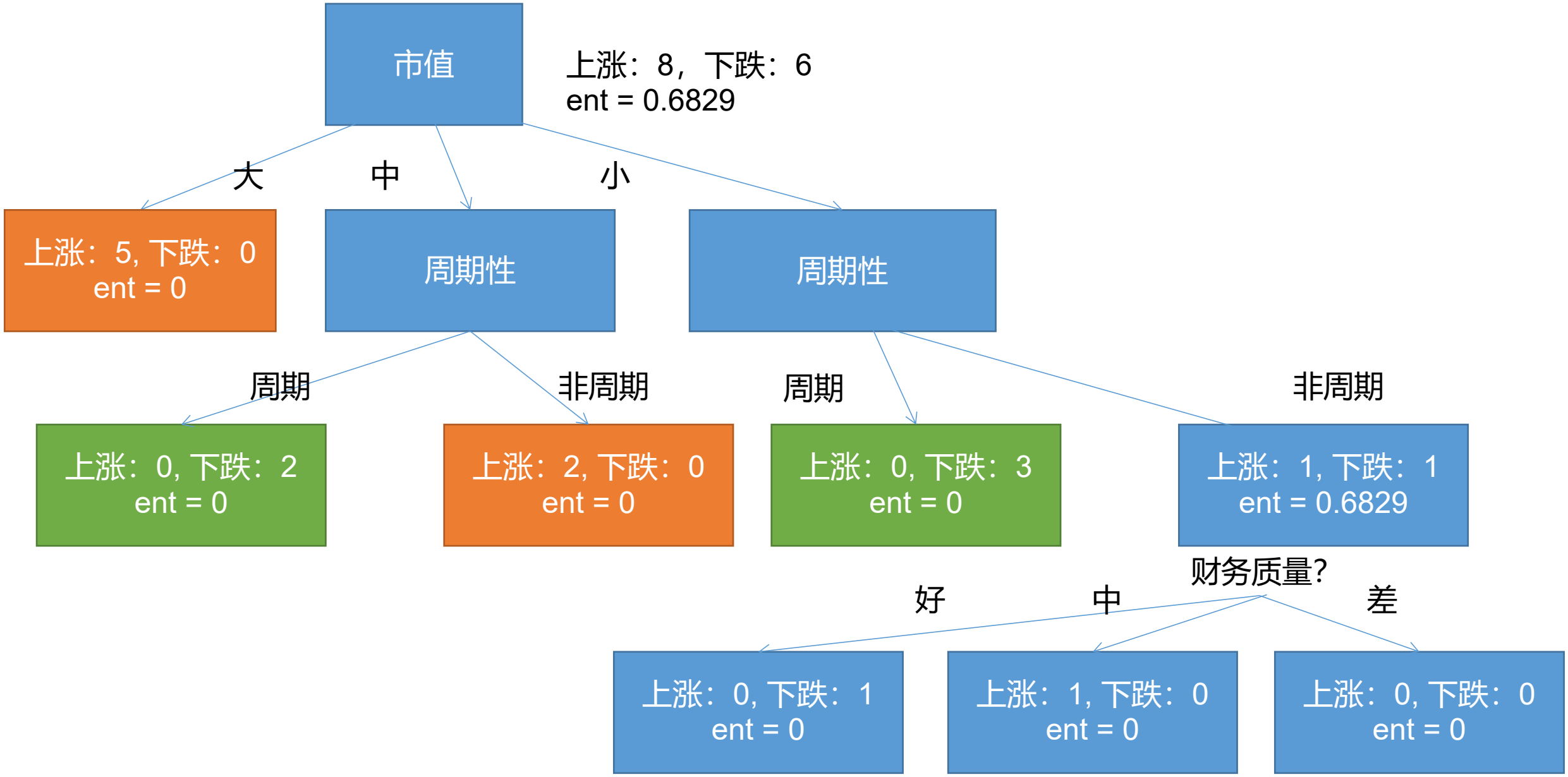
上涨：4, 下跌：3
ent = 0.6829

上涨：4, 下跌：3
ent = 0.6829

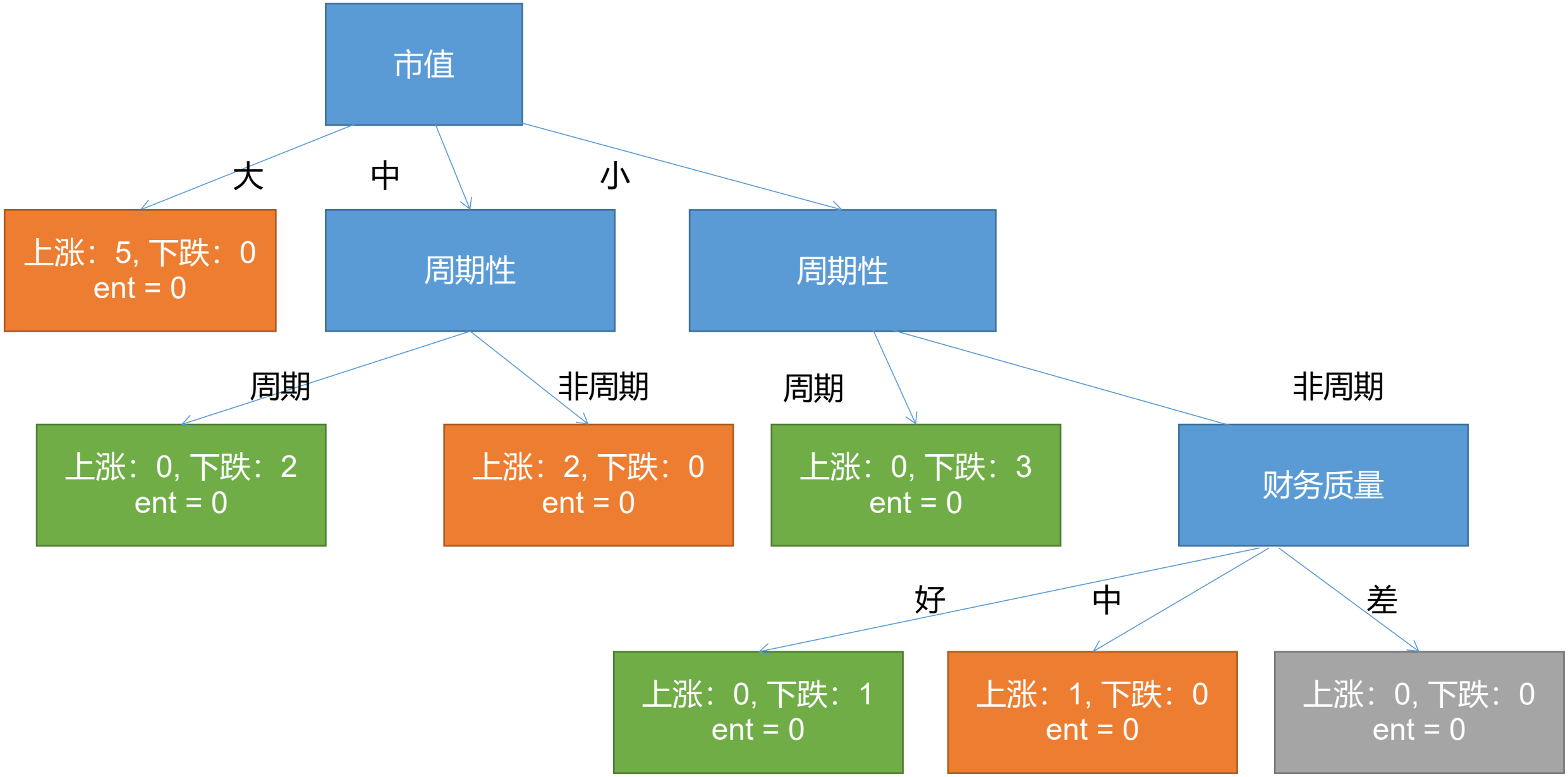
继续划分


市值
上涨：8，下跌：6
ent = 0.6829
大
中
小
上涨：5, 下跌：0
ent = 0
上涨：2, 下跌：2
ent = 0.6931
上涨：1, 下跌：4
ent = 0.5004
周期
非周期
周期
非周期
上涨：0, 下跌：2
ent = 0
上涨：2, 下跌：0
ent = 0
上涨：0, 下跌：3
ent = 0
上涨：1, 下跌：1
ent = 0.6829

继续划分


市值
上涨：8，下跌：6
ent = 0.6829
大
中
小
上涨：5, 下跌：0
ent = 0
周期性
周期性
周期
非周期
周期
非周期
上涨：0, 下跌：2
ent = 0
上涨：2, 下跌：0
ent = 0
上涨：0, 下跌：3
ent = 0
上涨：1, 下跌：1
ent = 0.6829
财务质量?
好
中
差
上涨：0, 下跌：1
ent = 0
上涨：1, 下跌：0
ent = 0
上涨：0, 下跌：0
ent = 0

最终生成的决策树

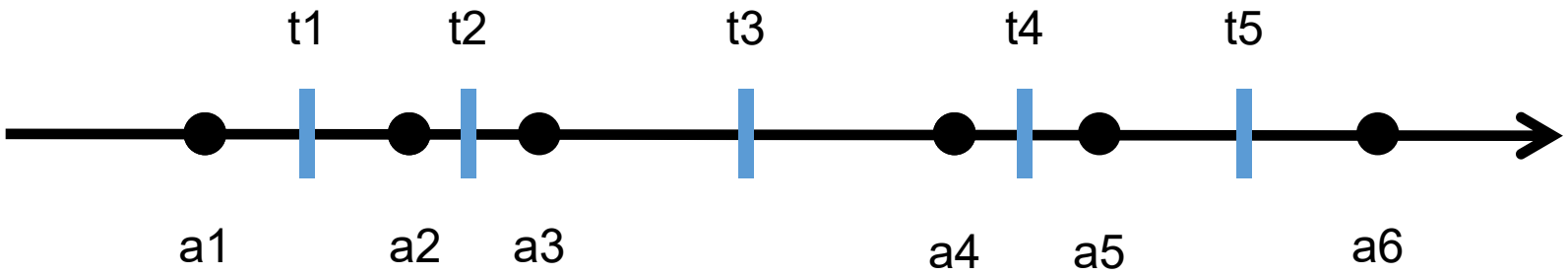

市值
大
中
小
上涨：5, 下跌：0
ent = 0
周期性
周期性
周期
非周期
周期
非周期
上涨：0, 下跌：2
ent = 0
上涨：2, 下跌：0
ent = 0
上涨：0, 下跌：3
ent = 0
财务质量
好
中
差
上涨：0, 下跌：1
ent = 0
上涨：1, 下跌：0
ent = 0
上涨：0, 下跌：0
ent = 0

何时终止继续划分?

1. 纯节点：当前节点中的样本属于同一类别
2. 空节点：当前节点中无样本数据
3. 已使用了所有的属性
4. 所有属性上的划分均无法降低信息熵

处理数值型特征

bi-partition



$$Gain(D,a)$$

$$= \max_{t_i} \text{Gain}(\text{D}, \text{a}, \text{t}_\text{i})$$

$$= \max_{t_i} \left\{ Ent(D) - \left(\frac{|D(a < t_i)|}{|D|} Ent(D(a < t_i)) \right) + \frac{|D(a \geq t_i)|}{|D|} Ent(D(a \geq t_i)) \right\}$$

bi-partition

标签型数据 $Gain(D,a)=\max_{a_i} \mathrm{Gain}(\mathrm{D},\mathrm{a},\mathrm{a_i})$

数值型数据 $Gain(D,a)=\max_{t_i} \mathrm{Gain}(\mathrm{D},\mathrm{a},\mathrm{t_i})$

基尼系数（Gini Index）

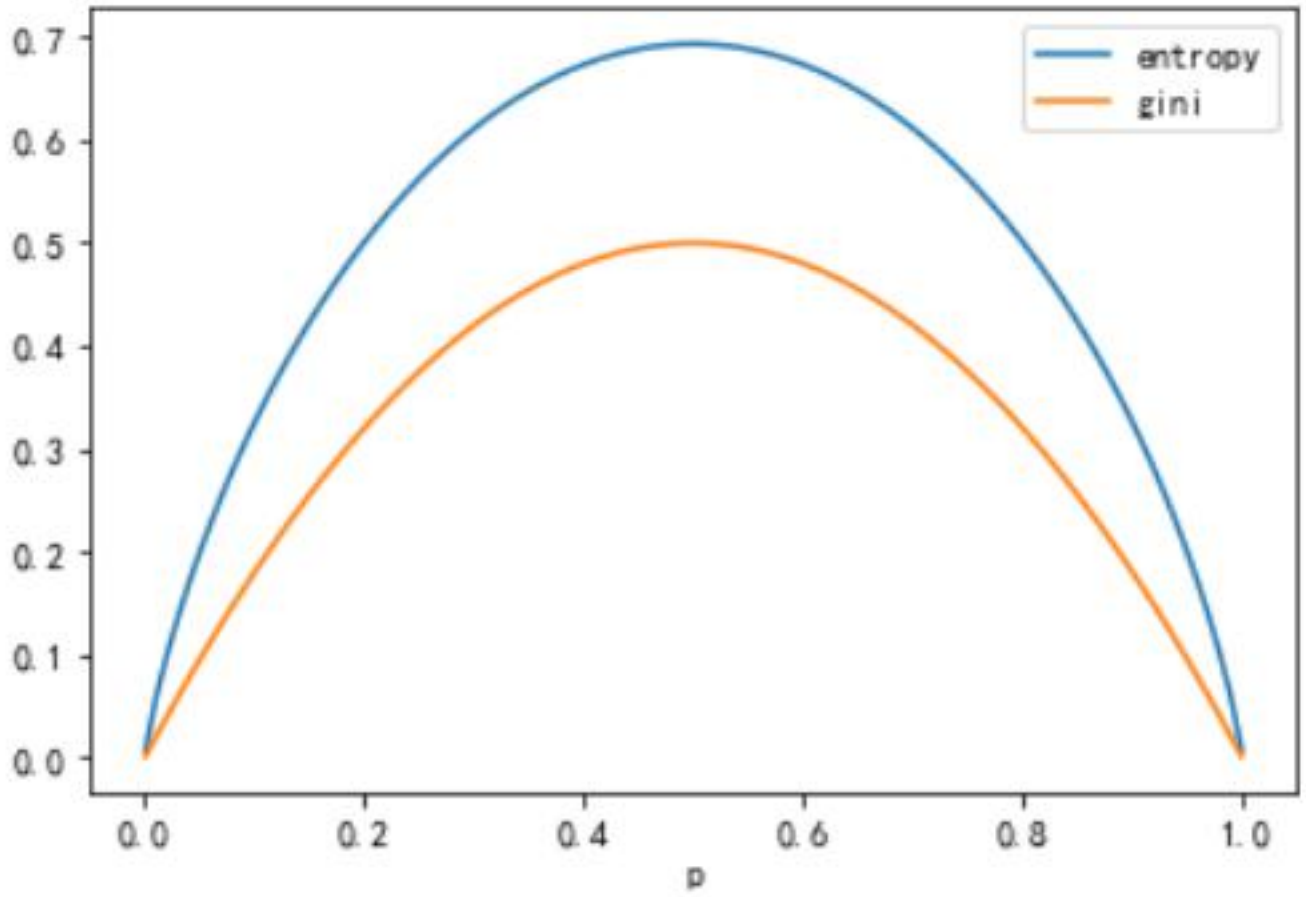
基尼系数

$$\text{Gini} = 1 - \sum p_i^2$$

与信息熵类似，基尼系数也可以反应一个随机变量的不确定程度

基尼系数 vs 信息熵

X ~ Bernoulli (p)



使用 sklearn 生成决策树

使用 sklearn 生成决策树

sklearn.tree.DecisionTreeClassifier

关键参数：

- criterion = 'gini'："gini' or 'entropy'

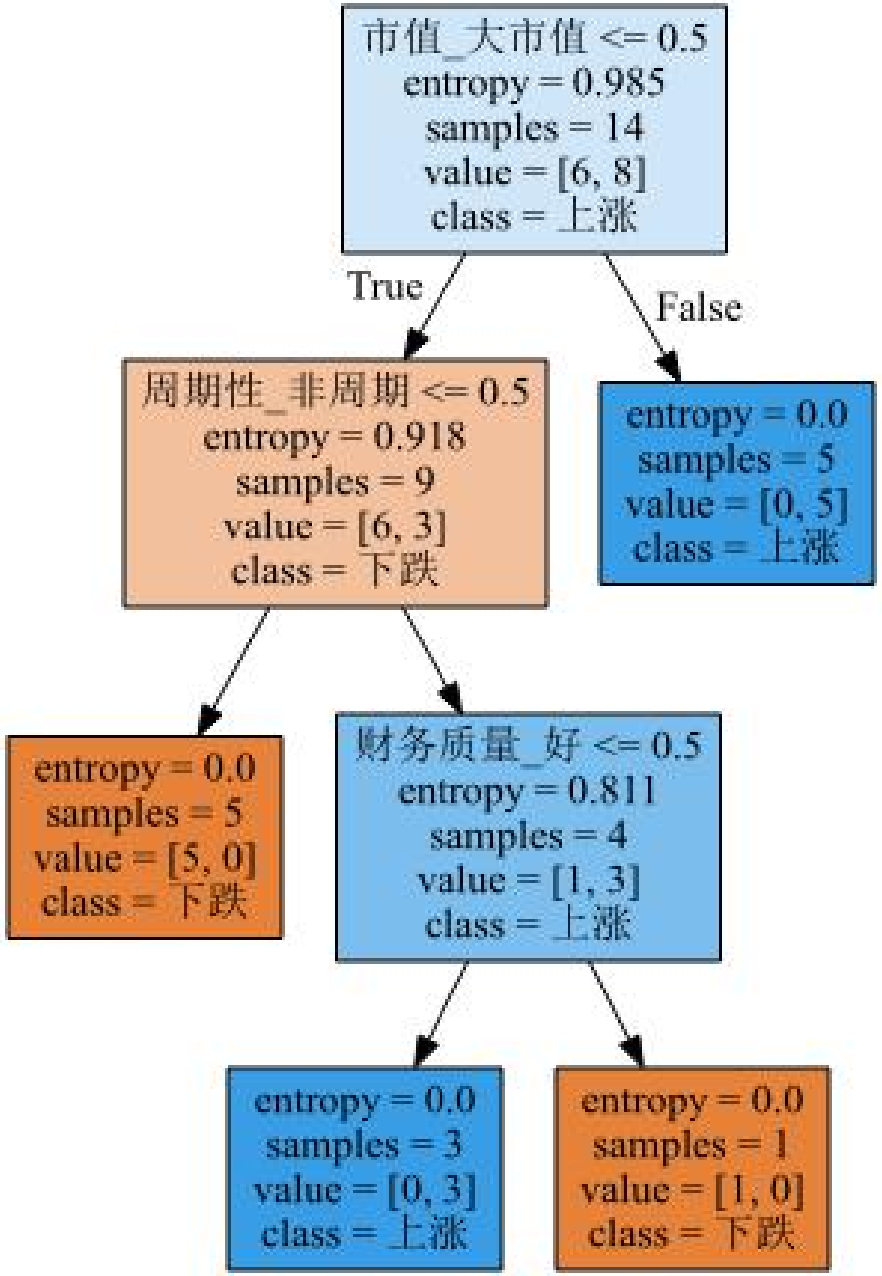
- max_depth = None：用于控制树的深度

- min_impurity_decrease = 0：只有当不纯度减低超过这一阈值时才进行划分

使用 sklearn 生成决策树

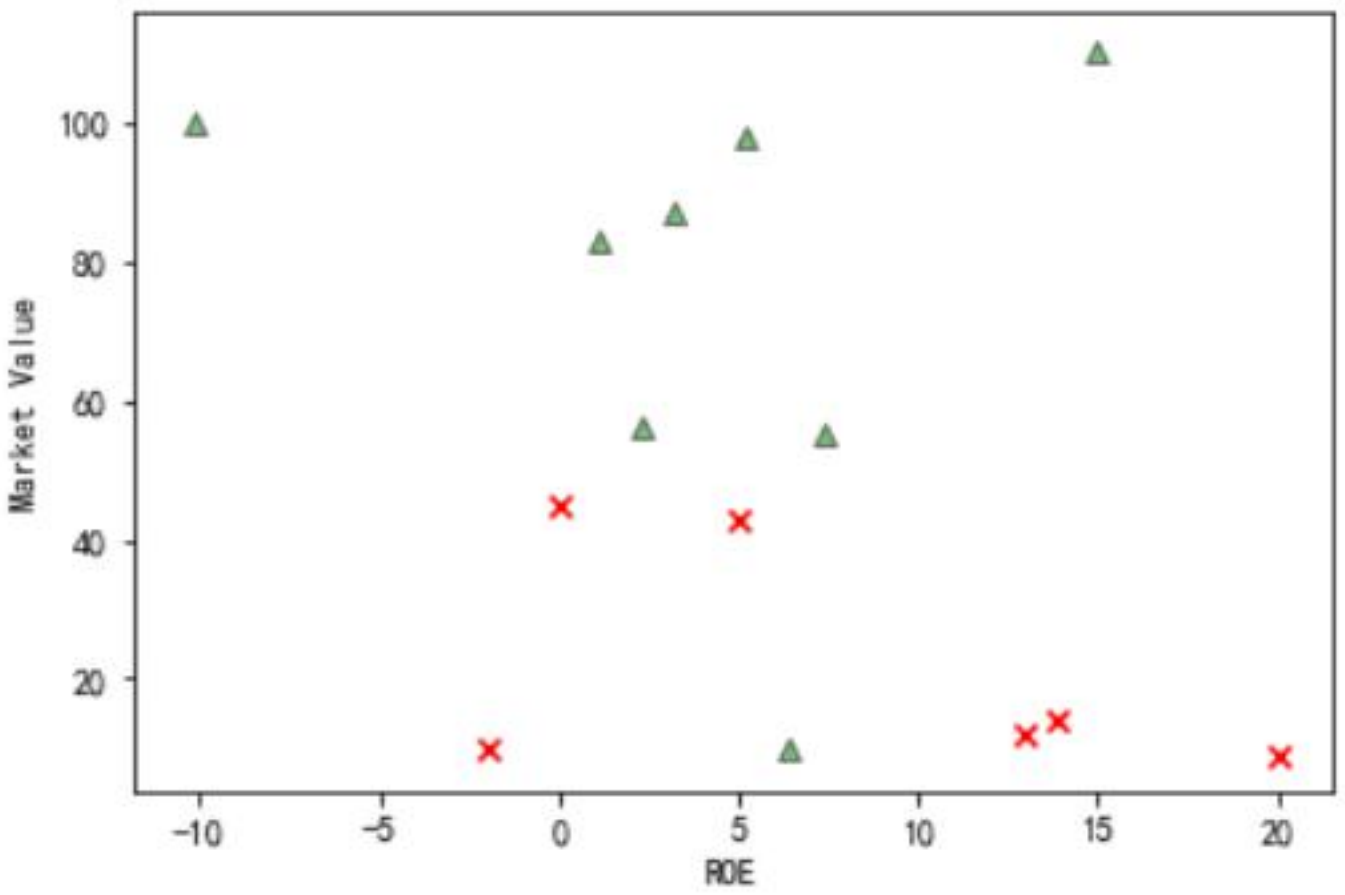
- sklearn 中的树是 “二叉树”
- sklearn 仅能处理数值型数据 （One-hot encoding）

财务质量	市值	周期性	性质	表现
差	大市值	周期	国有	上张
中	小市值	非周期	民营	上张
中	大市值	非周期	国有	上张
好	大市值	周期	国有	上张
差	大市值	非周期	民营	上张
差	中市值	非周期	民营	上张
中	中市值	非周期	民营	上张
中	大市值	非周期	国有	上张
好	小市值	周期	国有	下跌
差	中市值	周期	国有	下跌
好	小市值	周期	民营	下跌
中	中市值	周期	国有	下跌
好	小市值	非周期	民营	下跌
差	小市值	周期	民营	下跌

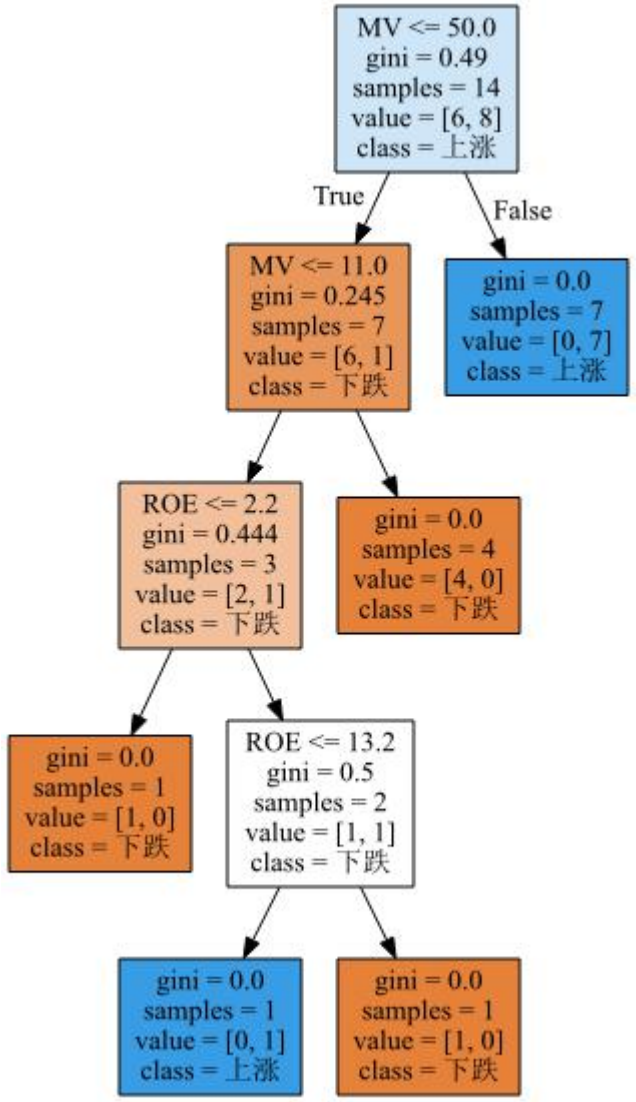


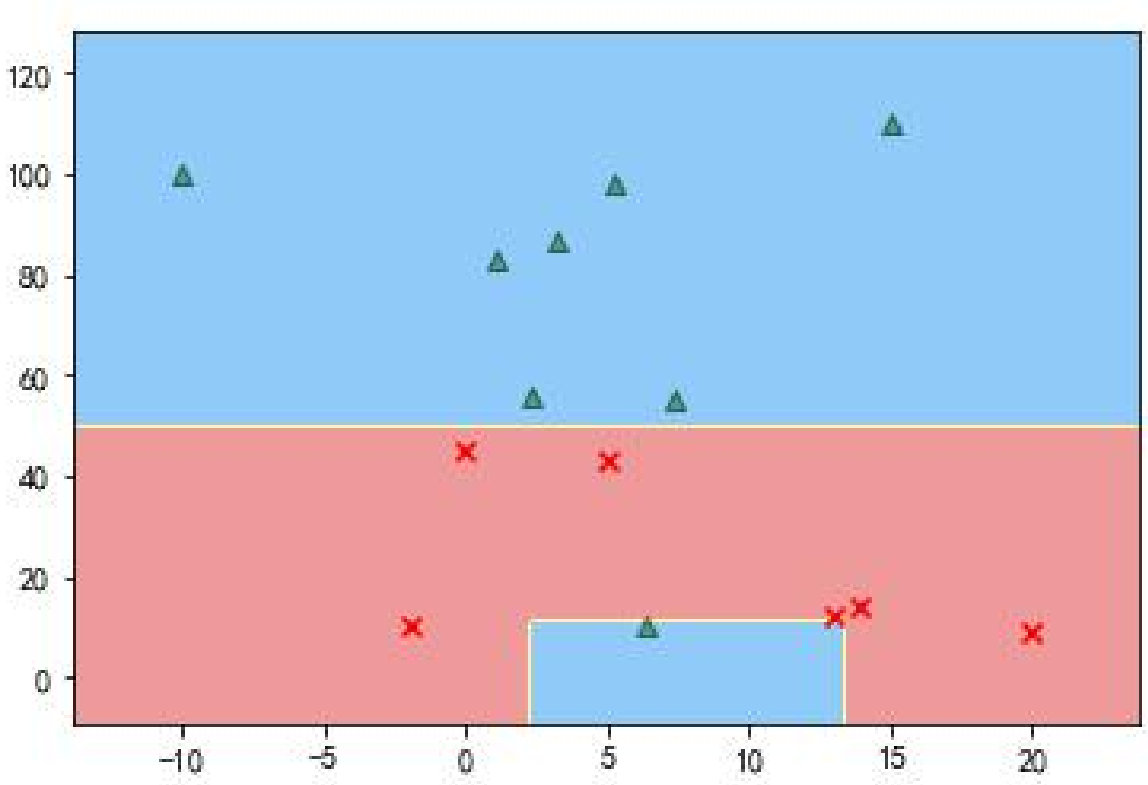
处理数值型数据的例子

净资产收益率	市值	表现
-10.1	100	上涨
6.4	10	上涨
5.2	98	上涨
15	110	上涨
1.1	83	上涨
2.3	56	上涨
7.4	55	上涨
3.2	87	上涨
20	9	下跌
0	45	下跌
13.9	14	下跌
5	43	下跌
13	12	下跌
-2	10	下跌



处理数值型数据的例子

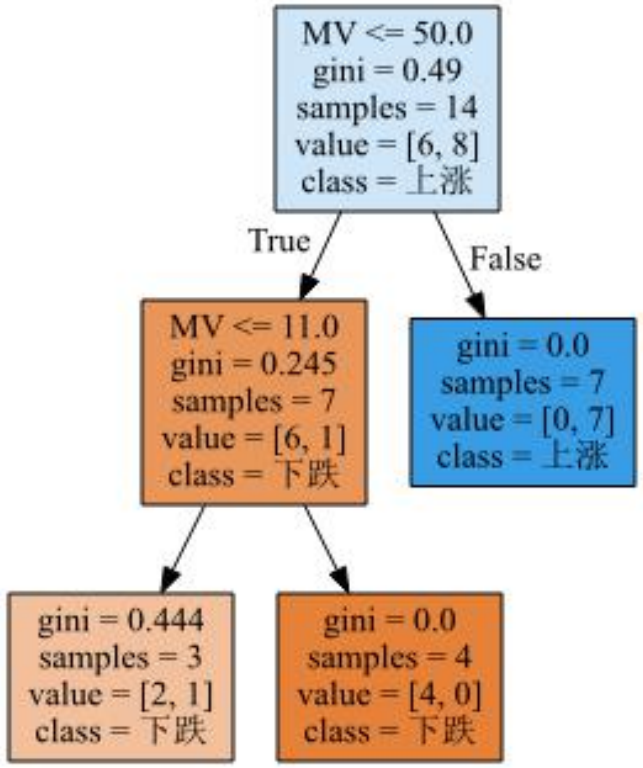




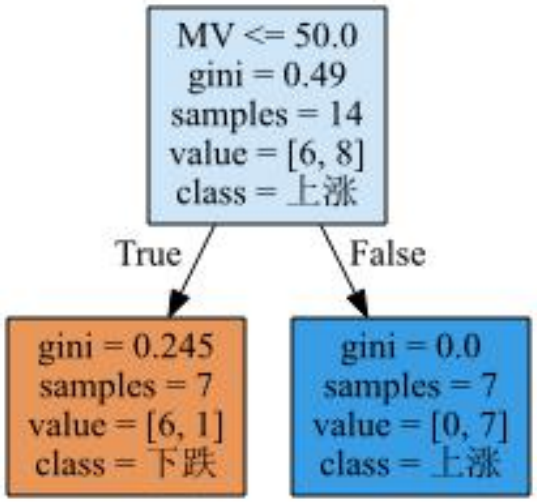
overfitting

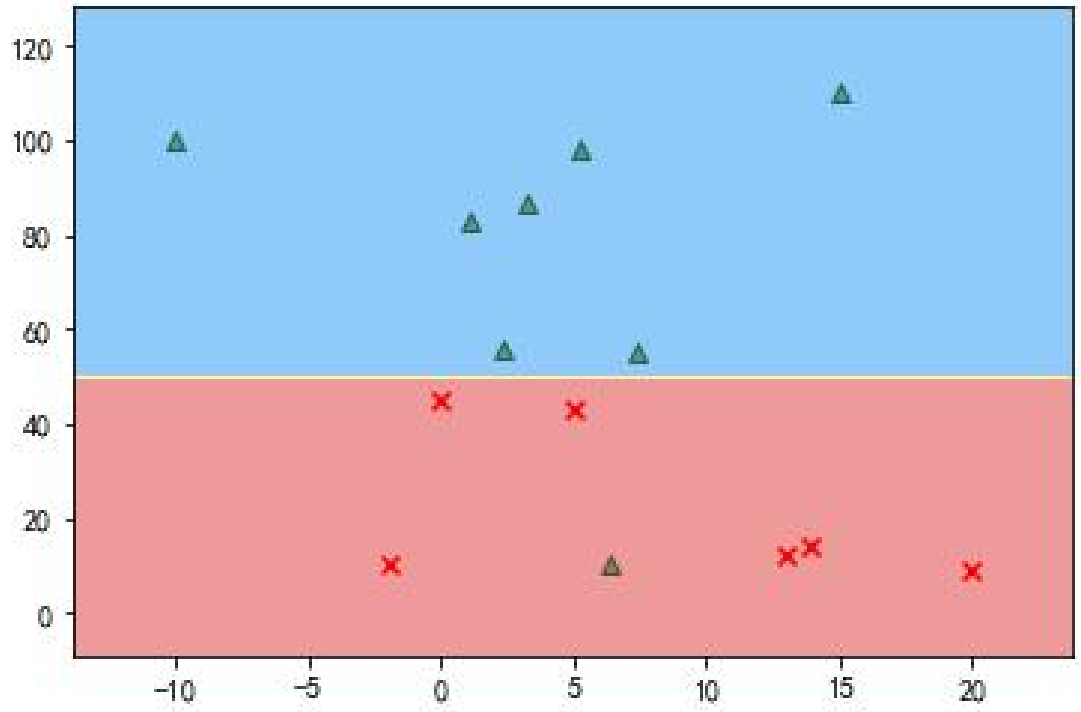
控制树的复杂度

max_depth = 2
min_impurity_decrease = 0

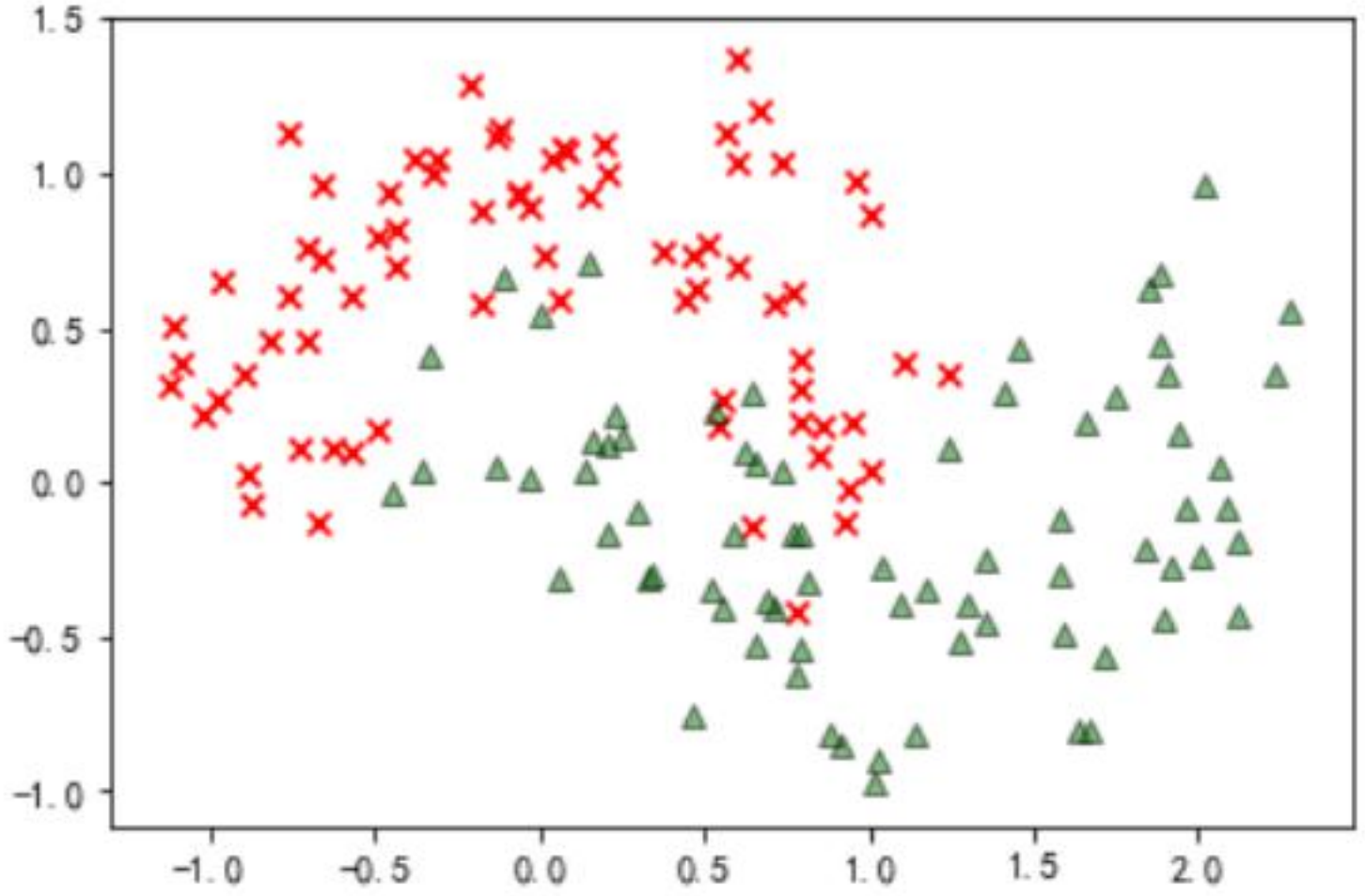


max_depth = 2
min_impurity_decrease = 0.1

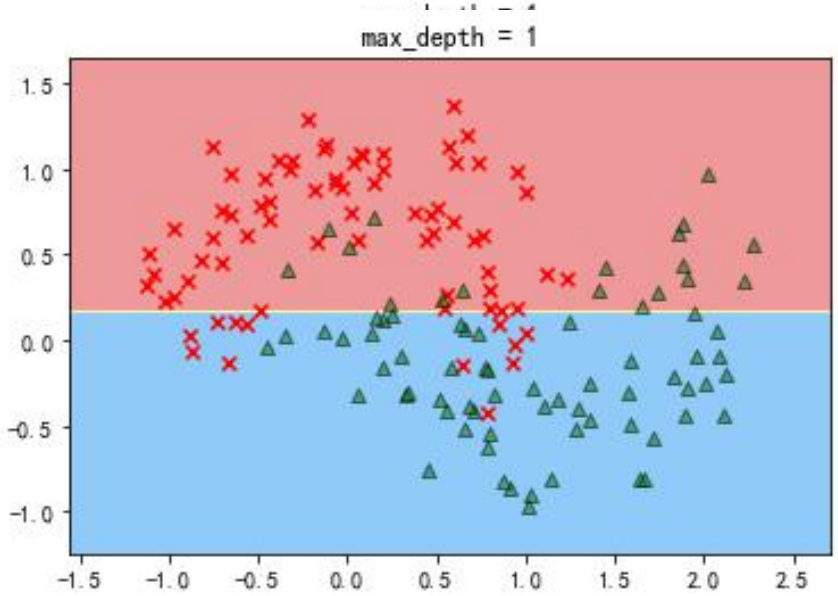


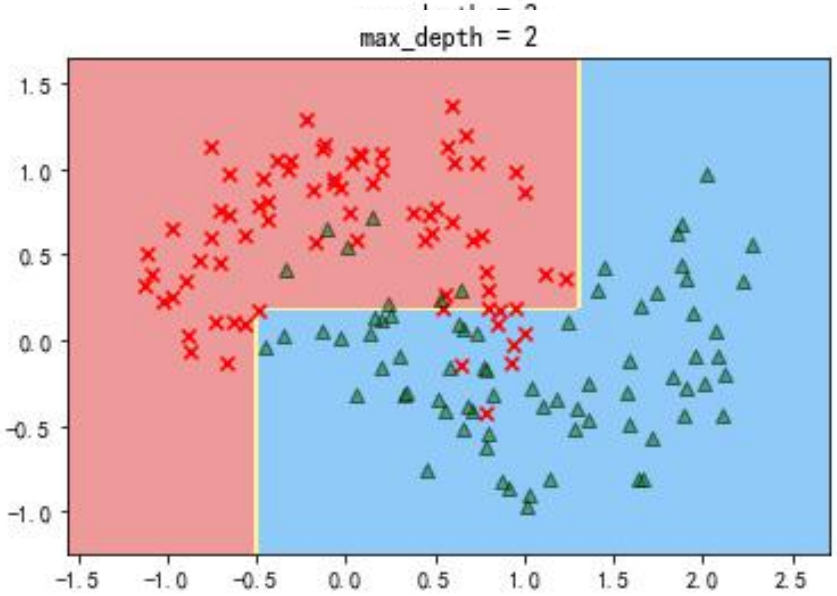


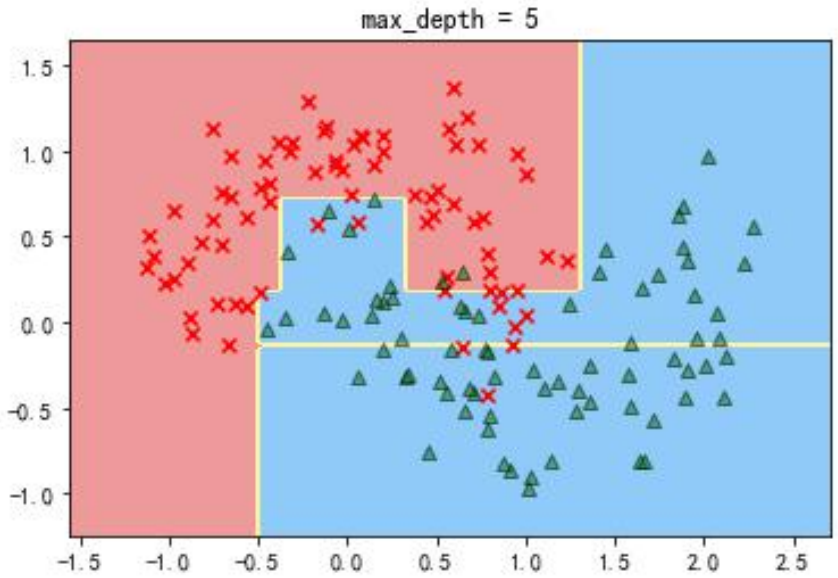
处理数值型数据的例子 2

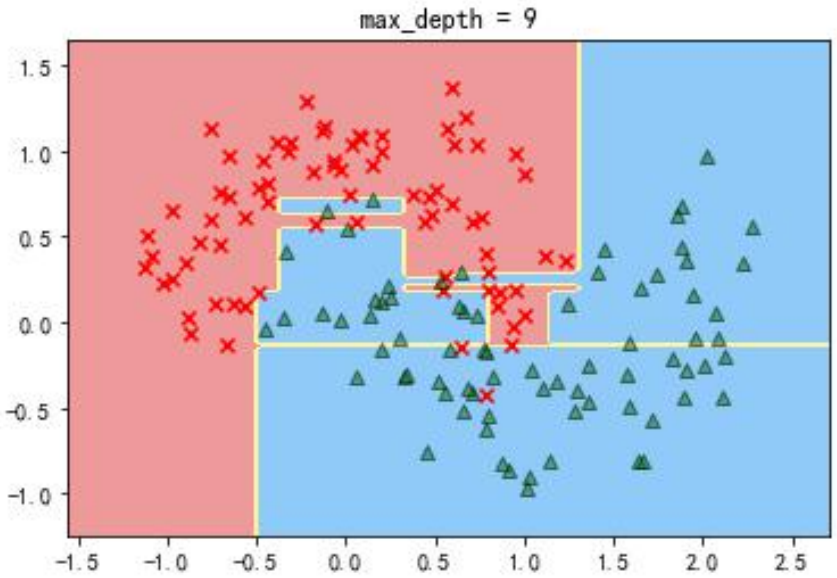


处理数值型数据的例子 2



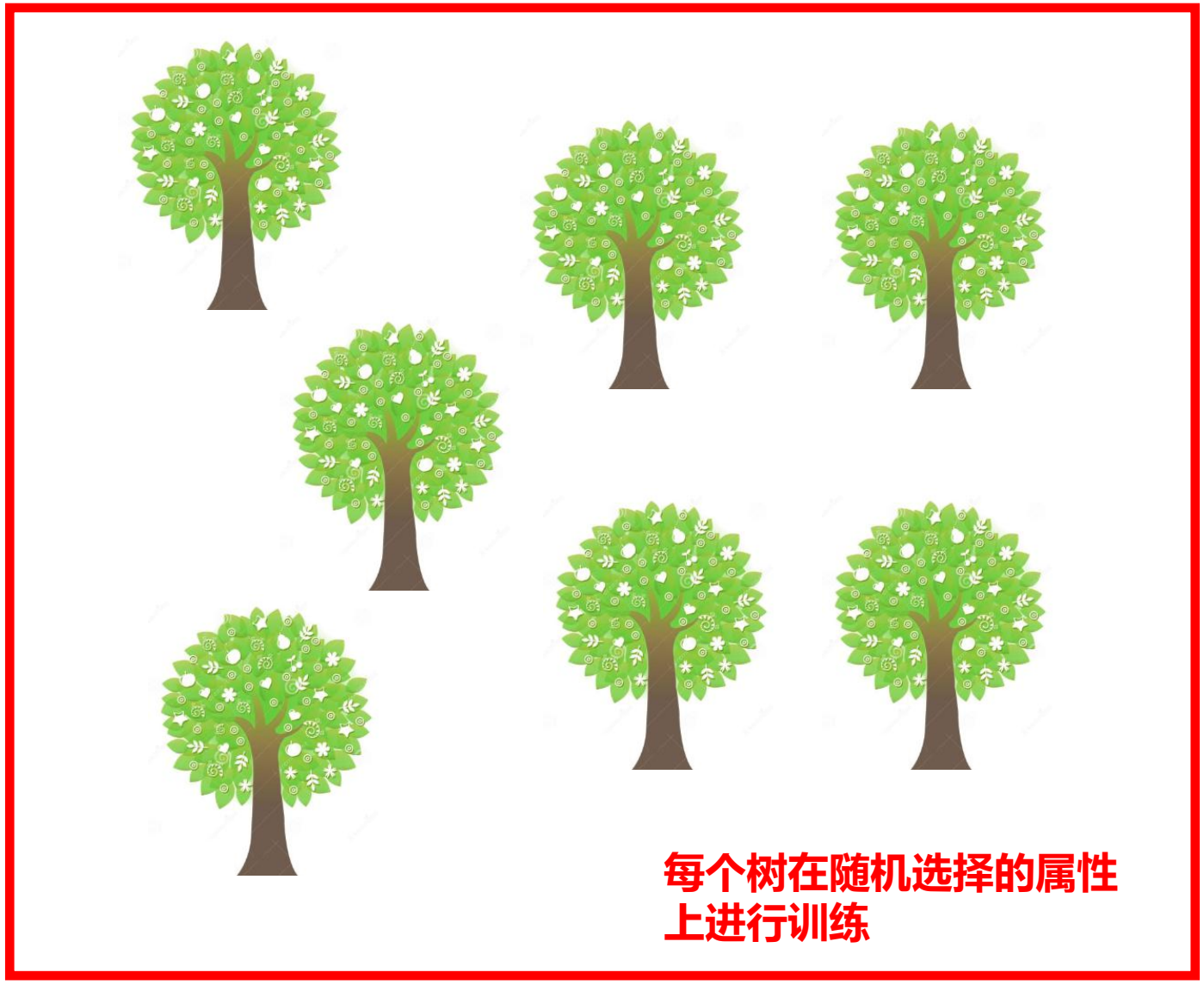






随机森林（Random Forest)

随机森林



投票决定
最终分类结果

使用随机森林

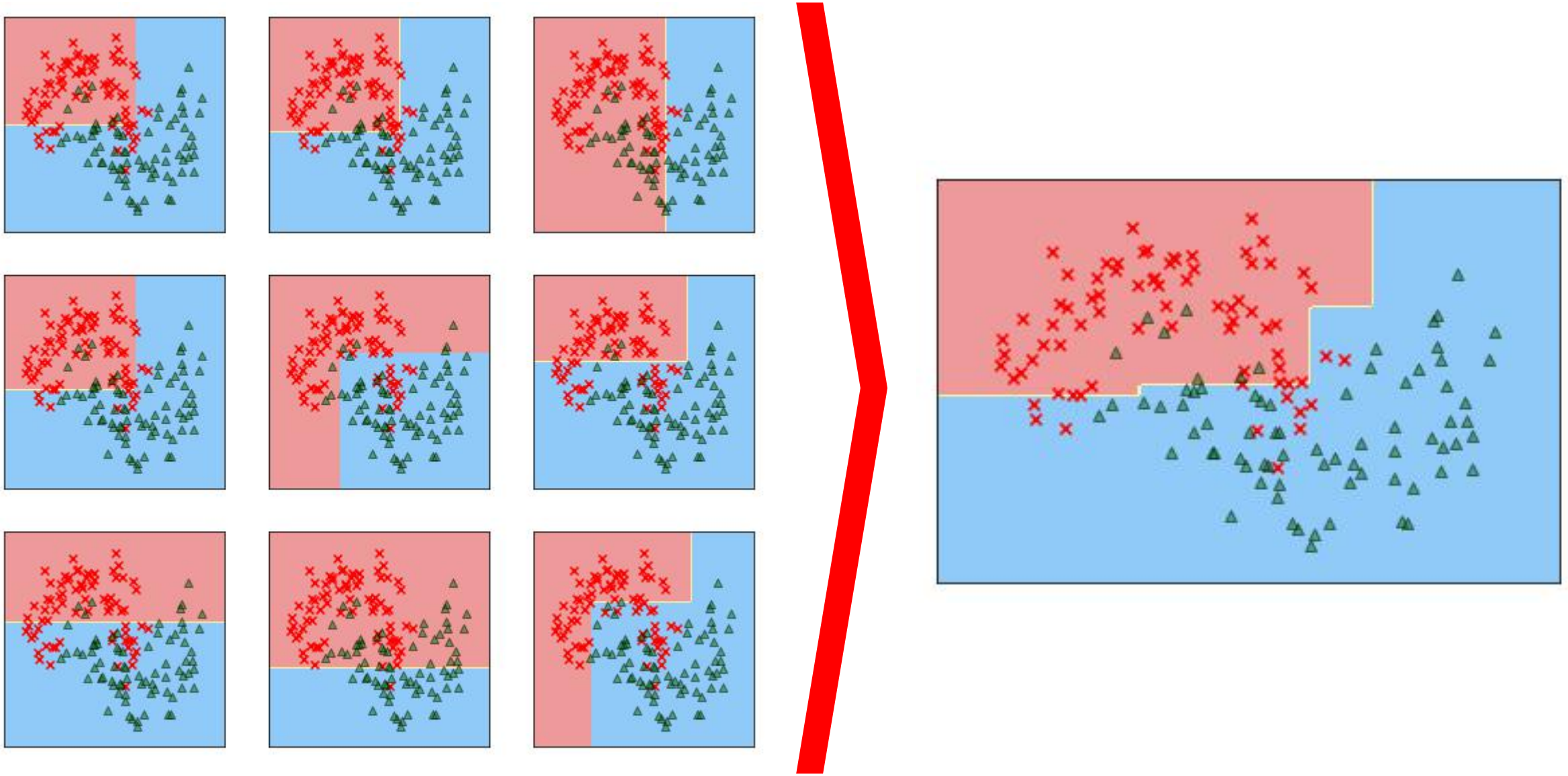
sklearn.ensemble.RandomForestClassifier

关键参数：

- n_estimators = 10：森林中树的数量

- max_depth = None：用于控制树的深度

使用随机森林

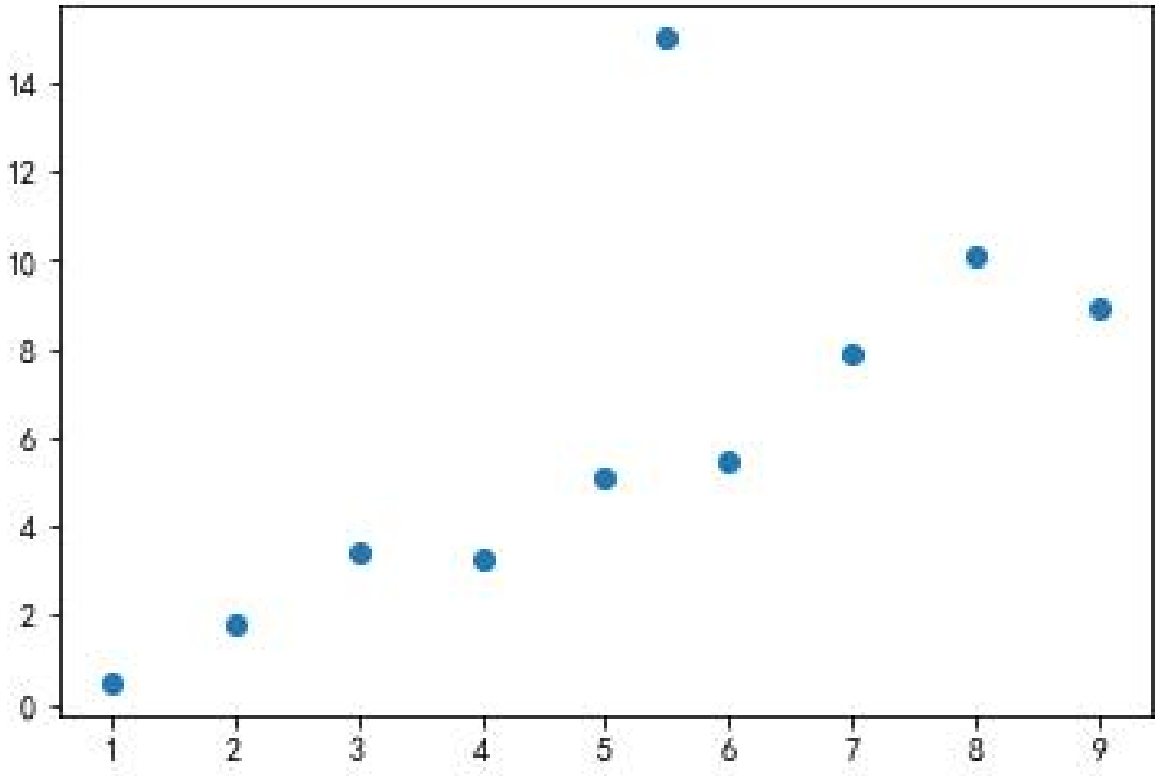


回归树

节点不纯度

分类树	回归树
entropy gini	mse

使用回归树



max_depth = 1

会得到什么样的树?

使用回归树

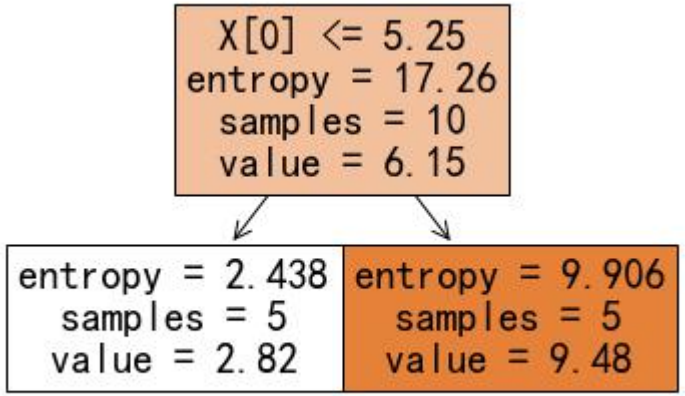
sklearn.tree.DecisionTreeRegressor

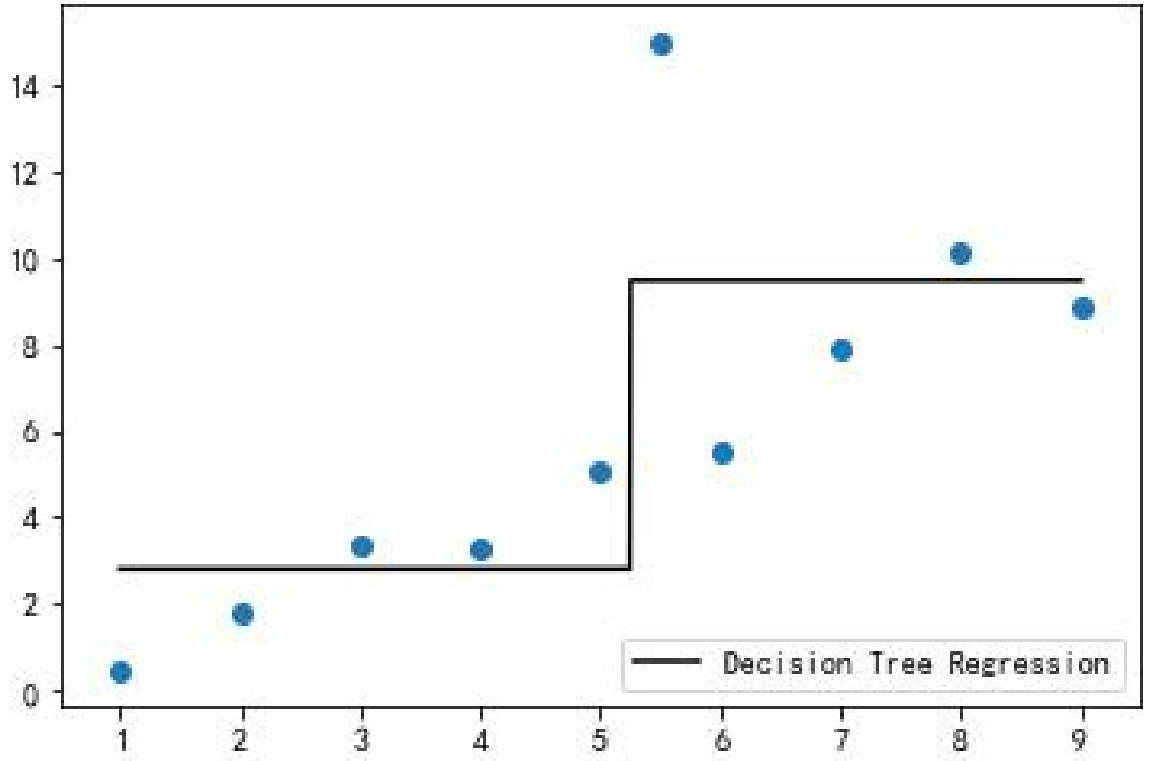
关键参数：

- max_depth = None
- min_impurity_decrease = 0

使用回归树

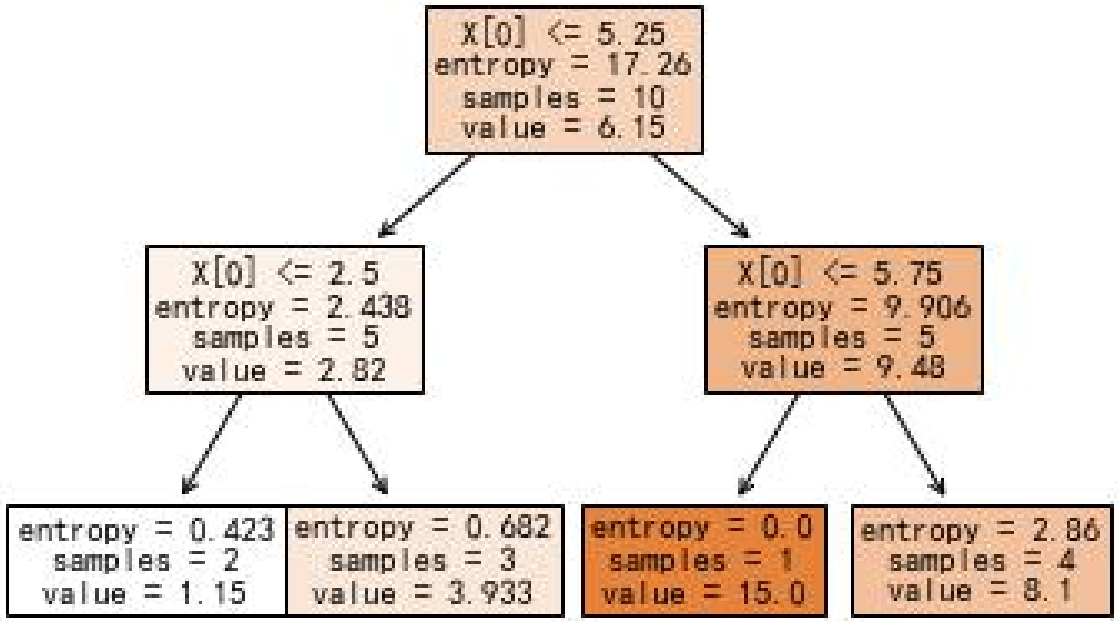
max_depth = 1

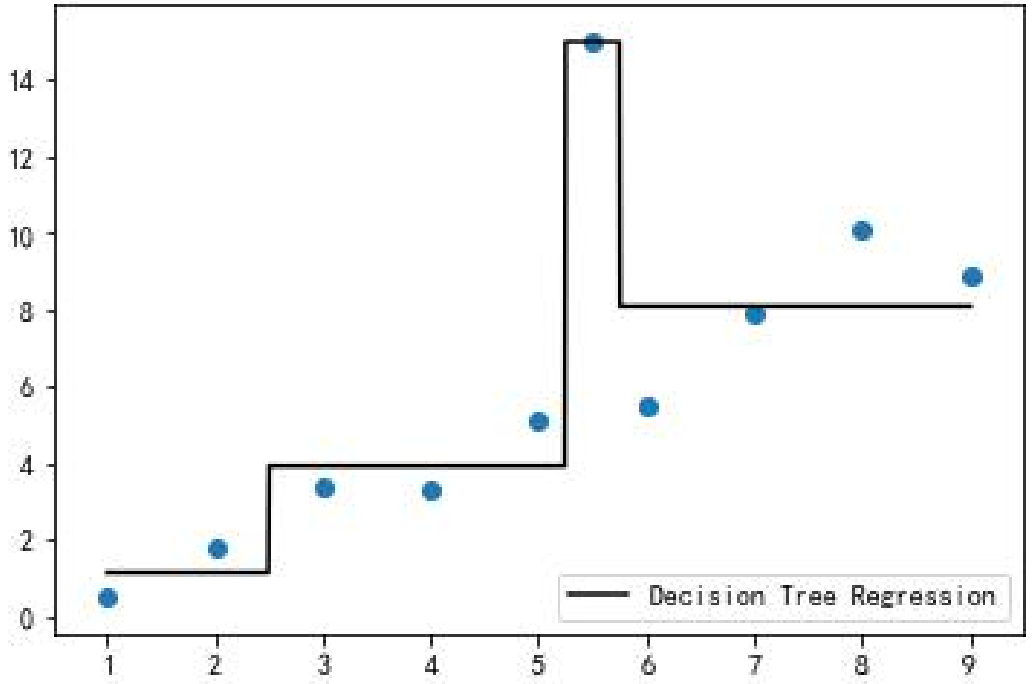




使用回归树

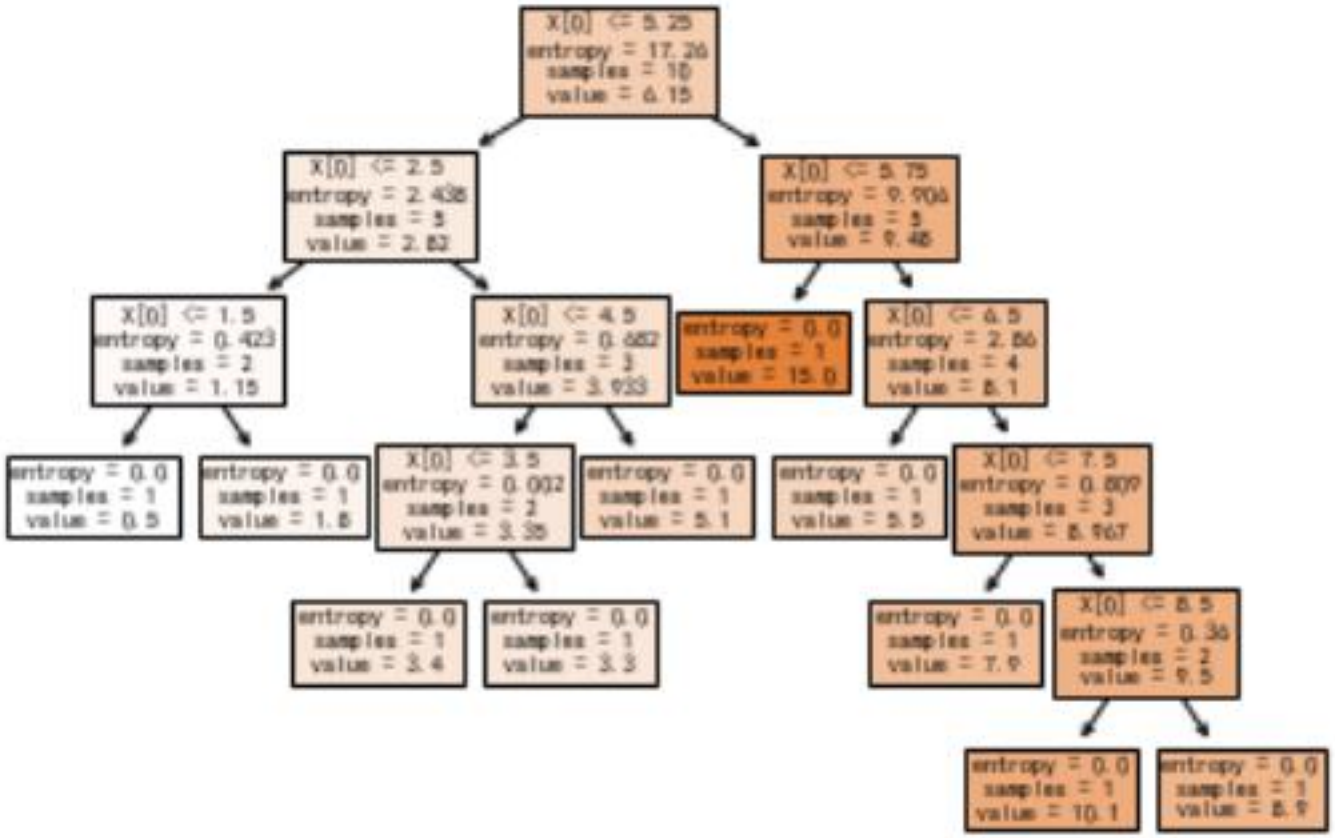
max_depth = 2

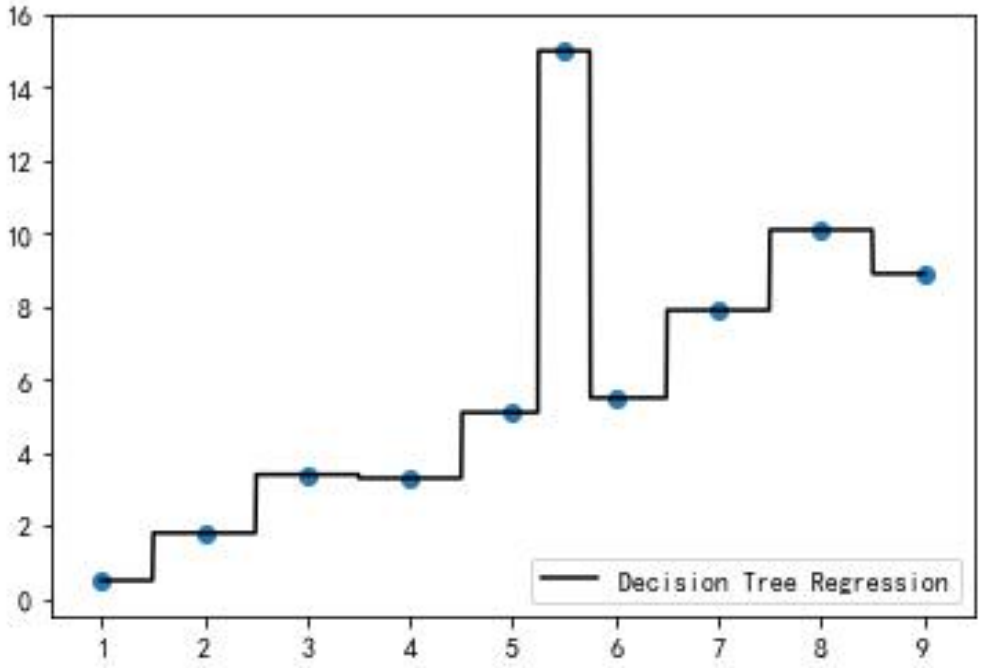




使用回归树

max_depth = None





决策树的优缺点

优点	缺点
易于理解、易于解读 原始数据不需要进行 scaling	泛化能力较差（需要和集成方法结合使用） 决策树的生成基于贪心算法（不保证全局最优）