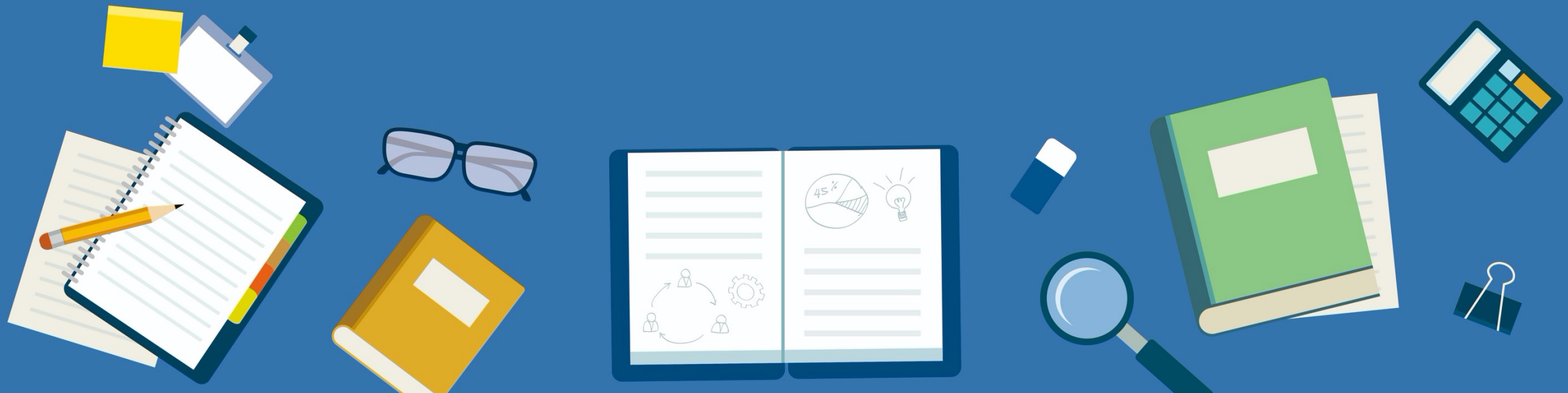
决策树

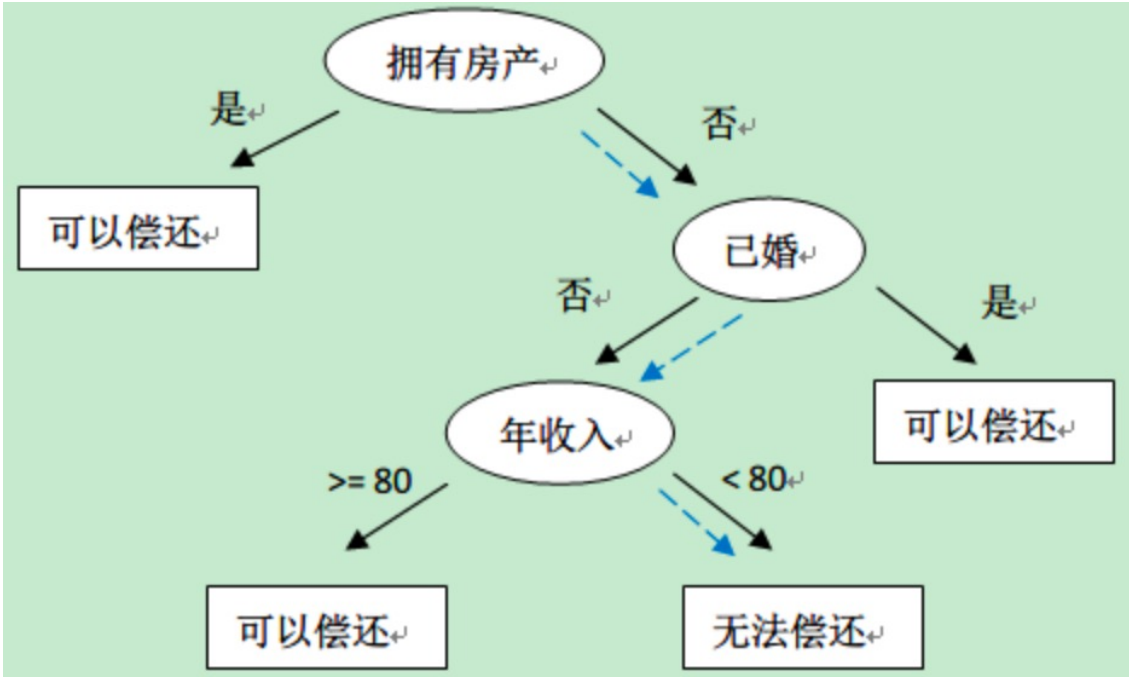


目录

目录

决策树简介

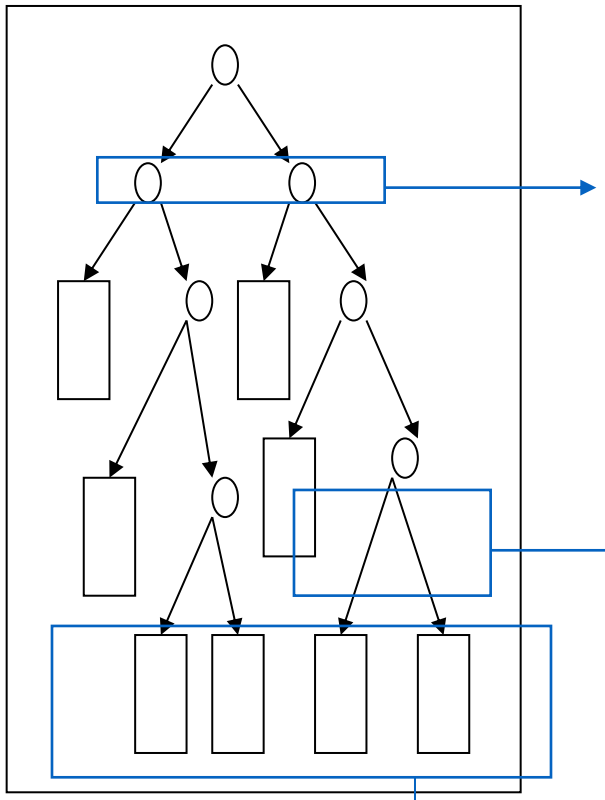
ID	拥有房产（是/否）	婚姻情况（单身，已婚，离婚）	年收入（单位：千元）	无法偿还债务（是/否）
1	是	单身	125	否
2	否	已婚	100	否
3	否	单身	70	否
4	是	已婚	120	否
5	否	离婚	95	是
6	否	已婚	60	否
7	是	离婚	220	否
8	否	单身	85	是
9	否	已婚	75	否
10	否	单身	90	是



一种树状结构的模型

决策树模型的基本概念

- 预测值
  - 分类：所在叶子结点的训练集标签进行投票，样本最多的类为预测值
  - 回归：所在叶子结点的训练集标签取平均为预测值



非叶子节点 (non-leaf node)

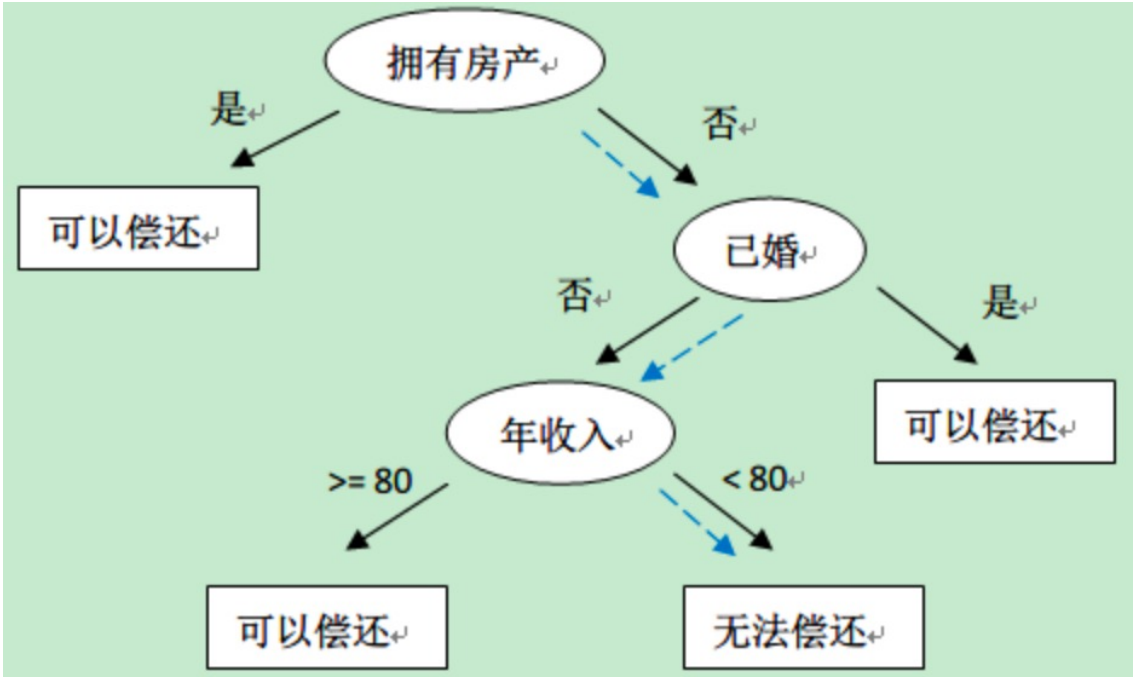
(对数据属性决策结果)

分支 (branches)

(对数据属性的决策)

叶节点 (leaf node)

(最终的预测结果)

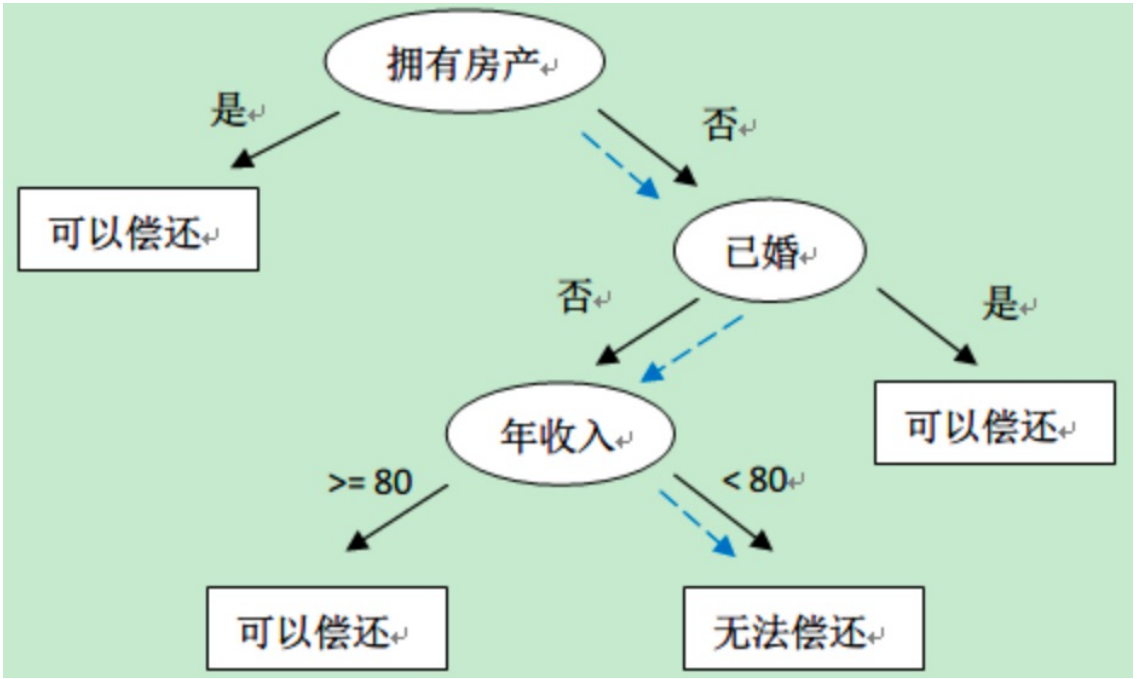


目录

决策树建树流程

- createBranch函数伪代码：

```
def createBranch():
    检测数据集中的每个子项是否属于同一个分类:
    if so:
        生成叶子结点
    else:
        寻找划分数据集的最好特征，并使用最好特征划分数据集
        for每个划分的子集
            调用函数createBranch()并返回结果到分支节点中
        return 分支节点
```



目录

决策树的划分

划分方式	模型
信息增益	ID3
信息增益率	C4.5
Gini系数	CART

决策树的划分：基尼系数

- 数据集D的纯度可以用Gini系数来度量：

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

- 基尼系数越小说明数据纯度越高
- 示例：
- A = [0,0,0,0,0, 1,1,1,1,1]　　$\mathrm{Gini}(\mathrm{A}) = 1 - (0.5^2 + 0.5^2) = 0.5$
- B = [0,0,0,0,0, 0,0,0,0,1]　　$\mathrm{Gini}(\mathrm{B}) = 1 - (0.9^2 + 0.1^2) = 0.18$

0表示可以偿还债务，
1表示无法偿还债务

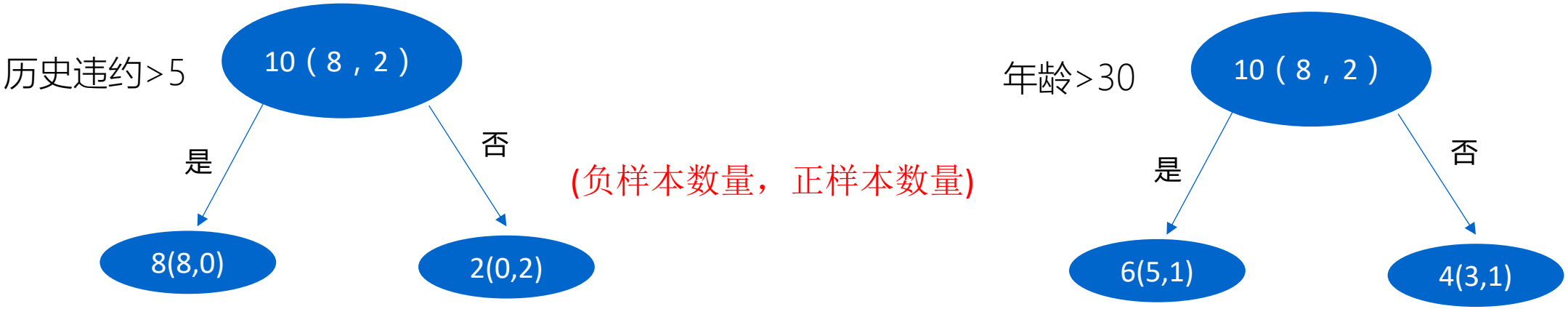
决策树的划分：CART树

- 在特征A下，将数据划分成两类，一类是D1,一类是D2,那么在特征A下的基尼系数为：

$$Gini(D,A)=\frac{|D_1|}{|D|}Gini(D_1)+\frac{|D_2|}{|D|}Gini(D_2)$$

- 选择划分后基尼系数最小的属性作为最优划分属性

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



$$\mathrm{Gini}(D,\text{历史违约}>5)=\frac{8}{10}\times\left[1-\left(\left(\frac{8}{8}\right)^2+\left(\frac{0}{8}\right)^2\right)\right]+\frac{2}{10}\times\left[1-\left(\left(\frac{2}{2}\right)^2+\left(\frac{0}{2}\right)^2\right)\right]=0$$

$$\mathrm{Gini}(D,\text{年龄}>30)=\frac{6}{10}\times\left[1-\left(\left(\frac{5}{6}\right)^2+\left(\frac{1}{6}\right)^2\right)\right]+\frac{4}{10}\times\left[1-\left(\left(\frac{3}{4}\right)^2+\left(\frac{1}{4}\right)^2\right)\right]=0.3167$$

决策树的划分：CART树

```
def createBranch():
    检测数据集中的每个子项是否属于同一个分类:
    if so:
        生成叶子结点
    else:
      寻找划分数据集的最好特征，并使用最好特征划分数据集
      for每个划分的子集
          调用函数createBranch()并返回结果到分支节点中
      return 分支节点
```

- 最好特征：在候选属性集合中，划分后基尼系数最小的属性

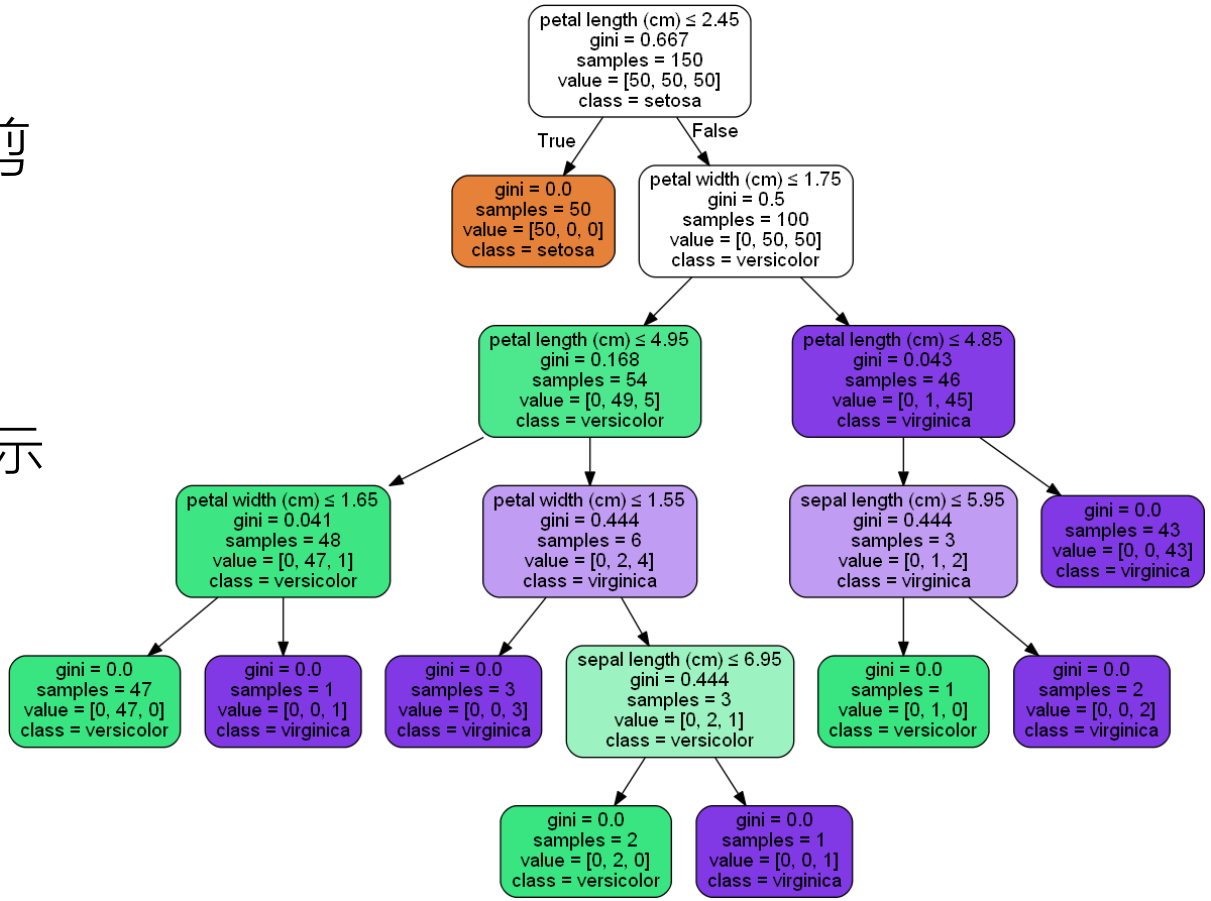
目录

决策树的剪枝

- 训练过程中，决策树完全生长，很容易造成过拟合，需要进行剪枝提高泛化能力
- 预剪枝：在构建决策树的过程中，提前停止
  - max_depth, min_sample
- 后剪枝：决策树构建好后，对它进行裁剪
- 损失函数：

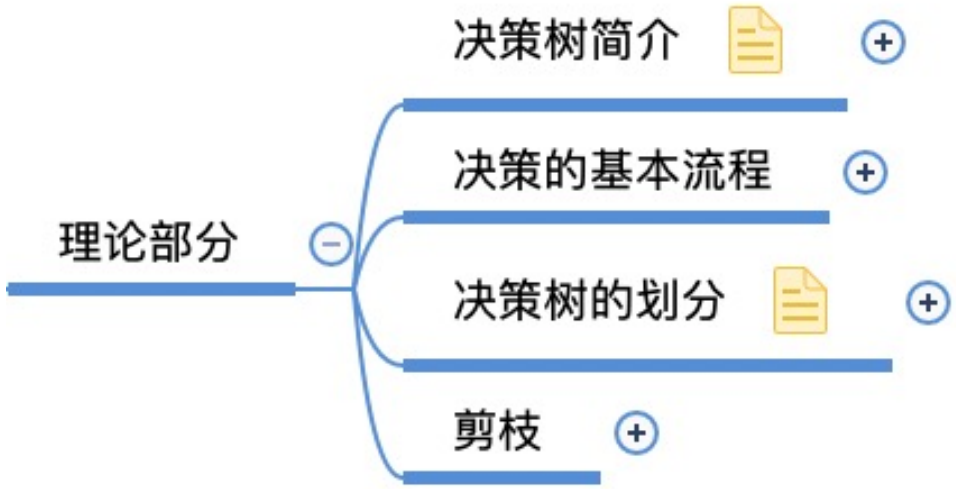
$$C_\alpha(T) = C(T) + \alpha|T|$$

- C(T)表示子树的预测误差，分类树用基尼系数表示
- |T|表示子树T的叶子节点个数
- α是正则化参数，α越大剪枝越厉害



目录

理论部分的总结



- 决策树的优点：
  - 推理过程容易理解，计算简单，可解释性强。
- 决策树的缺点：
  - 容易造成过拟合，需要采用剪枝操作。

目录