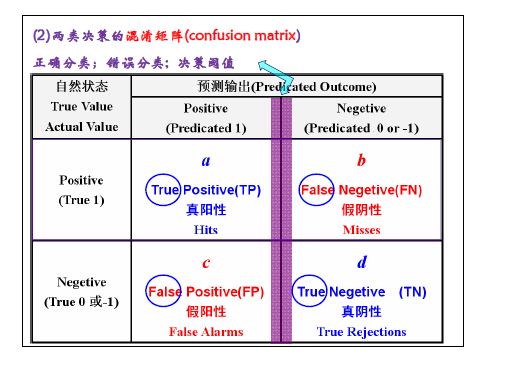
某个类别的，利用混淆矩阵评价分类模型的性能，查准率 查全率 f1分值，各个类别平均预测的正确率

混淆矩阵：confusion matrix

混淆矩阵的每一列代表了预测类别 ，每一列的总数表示预测为该类别的数据的数目；每一行代表了数据的真实归属类别 ，每一行的数据总数表示该类别的数据实例的数目。





混淆矩阵举例：

TP = True Postive = 真阳性； FP = False Positive = 假阳性

FN = False Negative = 假阴性； TN = True Negative = 真阴性

比如我们一个模型对15个样本进行预测，然后结果如下

注： 0 为阳性 1 为阴性

预测值：1    1    1    1    1    0    0    0    0    0    1    1    1    0    1

真实值：0    1    1    0    1    1    0    0    1    0    1    0    1    0    0

|  |  |  |  |
| --- | --- | --- | --- |
|  | 预测值：0 | 预测值：1 | 总计 |
| 真实值：0 | 4 TP | 4 FN | 8 |
| 真实值：1 | 2 FP | 5 TN | 7 |
| 总计 | 6 | 9 |  |

准确度(Accuracy) = (TP+TN) / (TP+TN+FN+TN) (即对角线上的值，就是阳性和阴性真实值跟预测值都一样的 / 所有的样本数)

在上面的例子中，准确度 = (5+4) / 15 = 0.6

精度(precision, 或者PPV, positive predictive value) = TP / (TP + FP)

在上面的例子中，精度 =4/（4+2）=0.667

F1-值(F1-score) = 2\*TP / (2\*TP+FP+FN)

（F1值 = 正确率 \* 召回率 \* 2 / (正确率 + 召回率)）

在上面的例子中，F1-值 = 2\*4/（2\*4+2+4）

/\*对角线上的数据为预测数据跟真实数据一样\*/

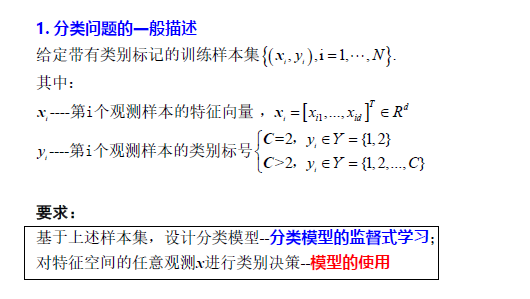
为0的准确率：

4/（4+2）

为1的准确率：

5（5+4）

如果是分类问题 什么是分类，对分类做定义 举几个例子



按照不同的特征对数据进行划分成不同的类别。

二分类：比如说根据人脸识别出男性或者女性

多分类：

iris数据集，根据数据的特征，把该样本划分为不同的品种。

/\*具体看分值多少，分值多的话，就多写点，少的话，就简写\*/

猫狗兔识别技术：，譬如有一个1000个样本的训练集，是1000张照片，里面有200张是猫，200张是狗，600张是兔子，一共分成三类。我们将每个照片向量化后，加上它的标签

“猫”——“0”

“狗”——“1”

“兔子”——“2”

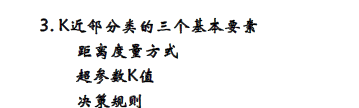
这相当于一个x和y的对应关系，把它们输入到训练集去训练（但是这个地方的标签0、1、2并不是实数定义，而是离散化的标签定义）。经过多轮训练之后，分类器将逻辑关系调整到了一个相对稳定的程度，然后用这个分类器再对这200张猫，200张狗，600张兔子进行分类的时候。发现：

200张猫的图片中，有180张可以正确识别为猫，而有20张误判为狗。

200张狗的图片可以全部判断正确为狗。

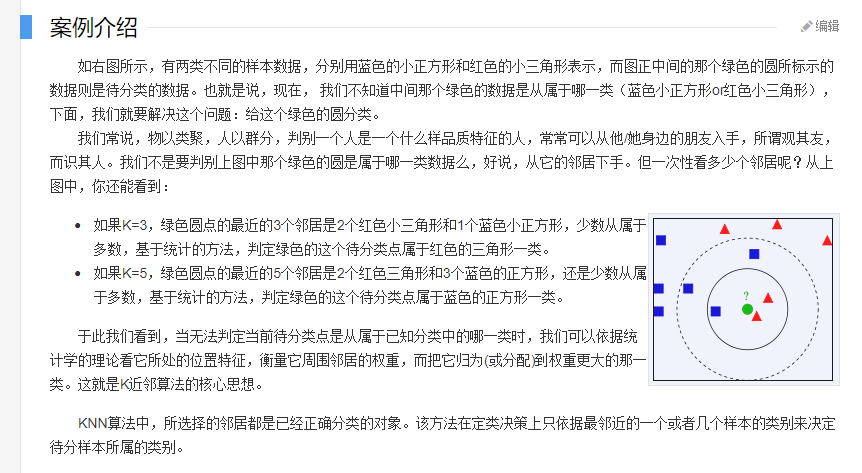
600张兔子的图片中，有550张可以正确识别为兔子，还有30张被误判为猫，20张误判为狗。

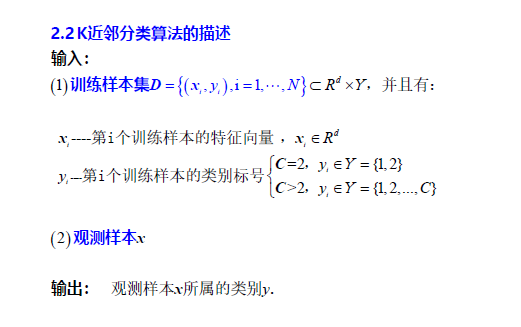
K近邻模型 给了已知标签的样本集 对任何一个样本集进行预测给了k紧邻模型 预测的过程 K代表什么 什么因素会影响K金林模型的分类性能

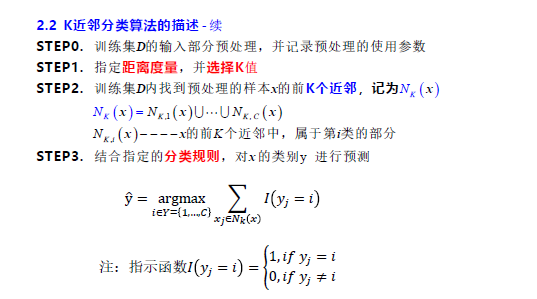


K近邻，就是画圈的，在规定的范围内，那个类别的数据离样本数据的个数多，就预测该样本为那个类别

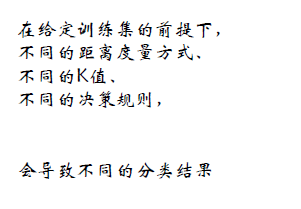
K最近邻(k-Nearest Neighbor，KNN)分类算法，是一个理论上比较成熟的方法，也是最简单的机器学习算法之一。该方法的思路是：在特征空间中，如果一个样本附近的k个最近(即特征空间中最邻近)样本的大多数属于某一个类别，则该样本也属于这个类别。





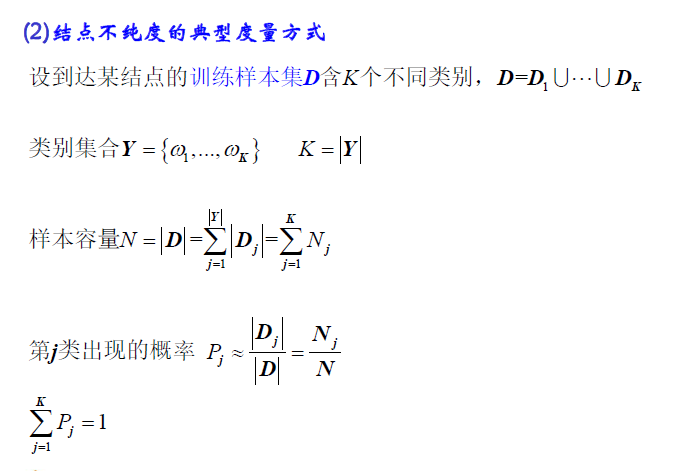


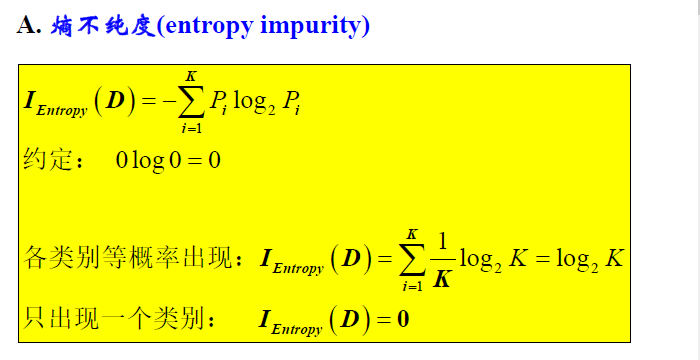
什么因素会影响K金林模型的分类性能 ？



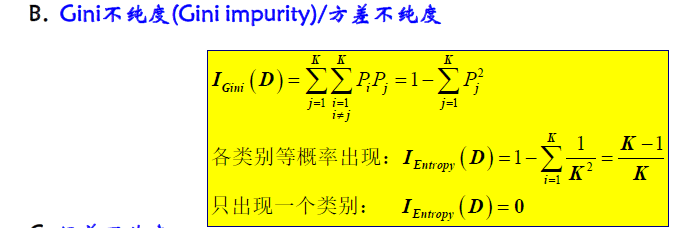
假定在树的构建过程中，怎么用训练样本集来评价当前三种 节点不纯度的方法 三种方法都计算一下不纯度



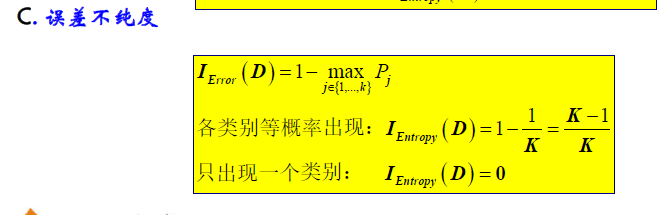


One 

Two



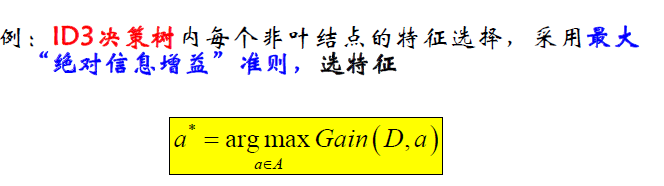
Three

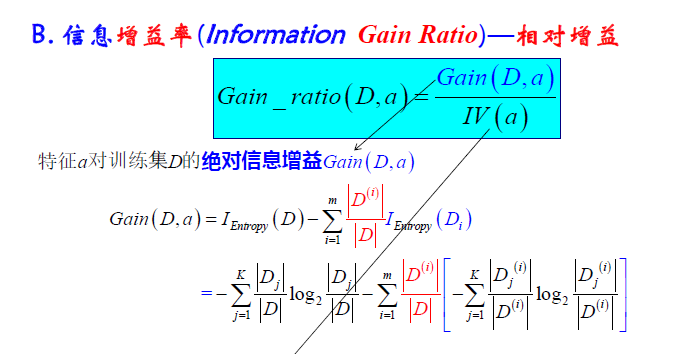


如果现在节点已经选好了特征，后面已经分成了几个分支 计算一下这种划分 决策树的绝对增益 用信息商，三种规则 节点不纯度的三种计算方式

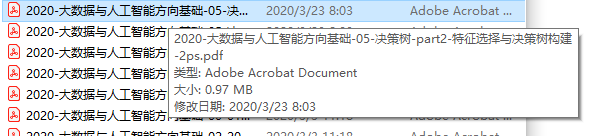
假定节点已经分裂了 怎么计算节点的绝对增益

绝对增益





(没学好，跳过)具体的知识点应该在这个ppt中



两个集成模型，一个随机森林，一个bagging树

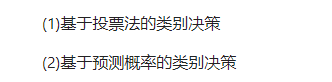
模型评价 混淆矩阵评价模型 根据混淆矩阵计算别的指标

（<https://gitee.com/lijianmin1/statistical_learning/blob/master/%E5%86%B3%E7%AD%96%E6%A0%91/Bagging_and_%E9%9A%8F%E6%9C%BA%E6%A3%AE%E6%9E%97.ipynb>）

bagging使用了boostrap的思想，从m个样本的训练集中有放回地抽取m次，获得第一个样本集，用于训练第一个基学习器，以此类推可获得k个样本集供基学习器训练。这里训练得到的模型都是各个独立的，最后，用测试数据，经过每个模型预测，最后采用投票的方式获得预测的结果。

其实，随机森林就是bagging树演化过来的，不同的是，在训练数据的时候，bagging使用的样本的所有特征，随机森林，可以随机的抽取特征进行训练（可以自己设定使用多少特征），后面的其实都是一样的。

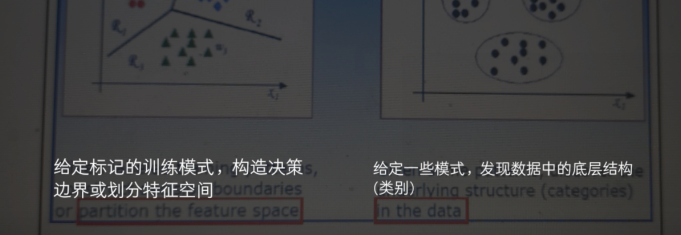
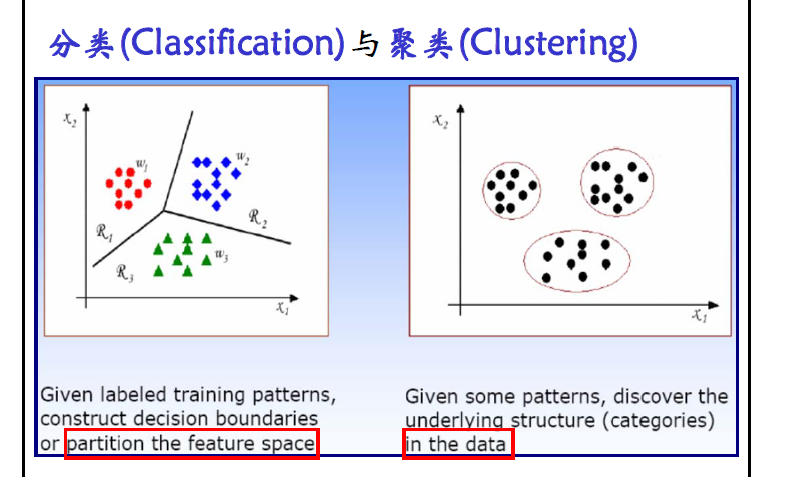
模型的评价：

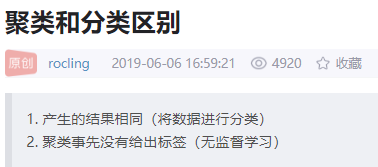


投票法就是，比如一个测试样本，根据每个模型所得到的结果（0,1），进行投票，看看那个类别得到的票多，就预测这个样本是那个类别。

预测概率的决策（其实就是投票法的一种变形，根据每个模型预测这个样本可能是每个类别的概率（0~1之间的数）），投票累加概率和，那个类别的所得票数多，就预测这个样本为那个类别。）

要了解大数据与人工智能 知道 什么是分类 什么是聚类 能够简单的一两句举例说明 做一个定义



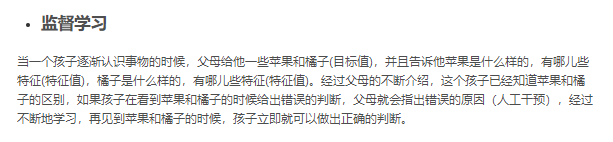


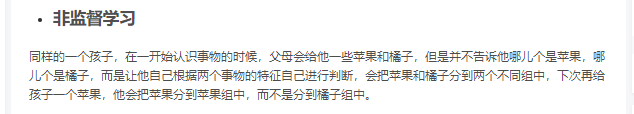


什么是监督式学习 非监督式学习 举例说明

监督学习和非监督学习的差别之一就在于：有没有目标值的差别

      而另一个区别就在于：学习过程有没有人工干预





监督式用于分类和回归模型的学习 需要训练集学习模型的时候 需要样本的输入部分和别的部分

聚类 采用没有标签的数据集完成 没有用到其他的

（对哒，基本就是这个意思）

模型的评价 两类方式 以分类为例，分类比较多对于分类来说 有两大类评价的方式 假定用训练集写好了分类模型 让模型对测试集挨个测试

得到混淆矩阵 基于混淆矩阵的模型评价 混淆矩阵：两类别的分类问题 两个类别同时都是我们关注的类别 比如性别分类 多类别的分类 每个都是关注的

特定的分类 比如人脸检测 给定两个或者两个以上的内容 同时感兴趣的类别 这样的分类模型得到的混淆矩阵 c行c列的，混淆矩阵每一个元素统计的是样本的数量

，

基于混淆矩阵做的评价 算出来参与决策的样本数量，利用混淆矩阵能够估计测试样本集总体预测的错误率和正确率 主对角线的元素都是正确（预测出来的结果跟真实结构一样）的样本数量，

估计某一个特定类别的查准率查全率以及f1分值。算出每一个类别的正确率 c个类别的预测正确率的平均值

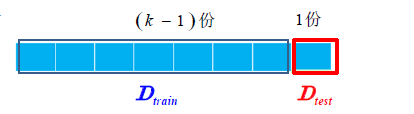
（重复了，大概就是这意思）

**分类是已知标签的，聚类是未知标签的！**

**K折交叉验证：（可用来进行模型选择或模型评价）**

①单轮K折交叉验证

把数据集D随机打乱，均分成K等份。依次从头到尾遍历每个等份中的样本，并以该等份中的样本作为测试集，以其他所有等份中的样本作为训练集，评估本轮交叉验证的正确率(或错误率)。完成遍历后，以K轮交叉验证的平均正确率(或错误率)作为最终交叉验证的总体正确率（或错误率）。



②多轮K折交叉验证即将单轮K折交叉验证重复多次。

K折交叉验证可能涉及到分层随机打乱

分层随机打乱：在保证每一折中每类样本数量一致的前提下进行样本的随机打乱。

**三种监督式学习模型（监督式即样本标签已知的情况下进行模型学习）**

**监督式学习：基于已知标签的数据集进行模型的学习，基于该模型对未知样本的输出做出预测。**

**（考分类问题的概率比较大）**

1. **KNN模型（K近邻模型）**

流程：

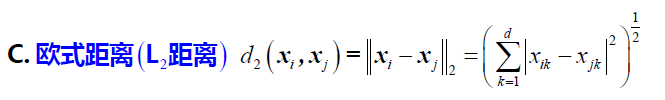
1. 对训练样集D进行标准化预处理，并记录预处理的使用参数。

标准化预处理通常采用0均值、1方差的方式：

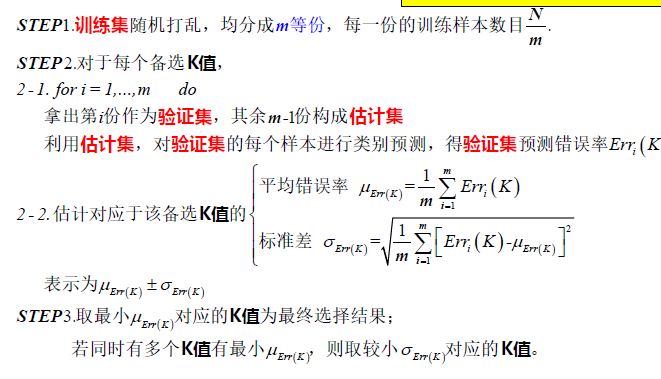
对于每个样本，用该样本每个特征的值减去该特征的均值除以该特征的标准差。

1. 指定距离度量，并选择K值。

距离度量通常采用欧氏距离：（欧式距离就是通常意义上说的距离）



K值的选择通常用到m折交叉验证(跟之前说的K折交叉验证一样)



1. 在训练集D内找到预处理的样本x的前K个近邻。

就是找到距离当前样本最近的K个近邻。

1. 结合指定的分类规则，对x的类别进行预测。

结合最近邻的K个样本的类别，来决策当前x的类别。

决策方式：①多数表决，即近邻的K个样本中哪一类的样本最多就预测为哪一类。

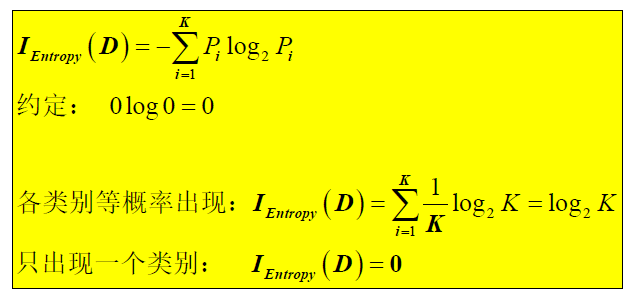
②基于距离的加权投票。在最近邻的K个样本中，距离当前样本越近的 样本权重越大。

1. **决策树模型**

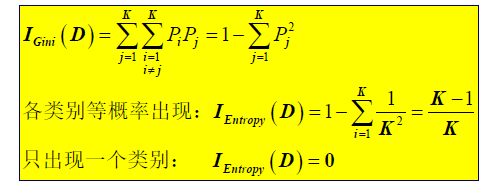
**决策树模型分为分类树（用于预测类别）和回归树（用于实值函数的回归）。**

**度量不纯度的方法：（这个只能记住公式）**

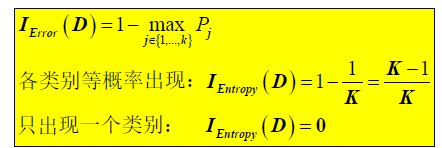
1. 熵不纯度



1. Gini不纯度



1. 误差不纯度

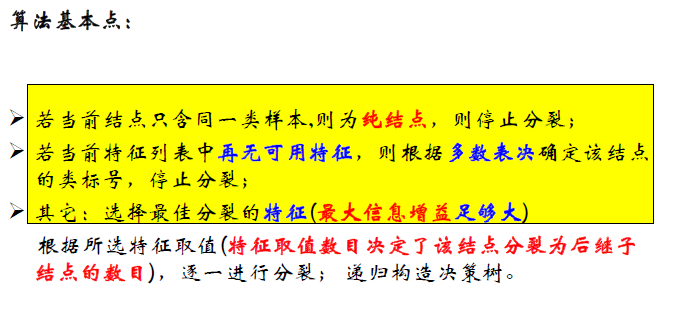


**基于不纯度的节点特征选择（具体得记公式，有点复杂跳过了）**

1. 绝对信息增益
2. 信息增益比
3. 基尼指数

三种决策树：

①ID3决策树：基于最大绝对信息增益的特征选择原则。（只能用于分类）



只适用于离散型或者非数值型特征描述的样本集，不处理缺失信息、不涉及剪枝操作。

②C4.5决策树：基于最大增益率的特征选择原则。（只能用于分类）

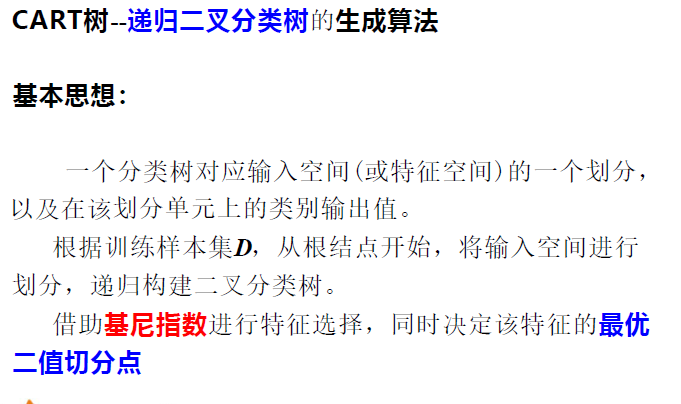
相比ID3决策树：可以对连续数值特征进行处理、可以对缺失值进行处理、首先让树充分生长，然后利用分枝的统计显著性来实现剪枝（后剪枝）。

对于连续数值型的特征采用：二分法。基于最大绝对信息增益找最佳切分点，基于信息增益比选择最佳分裂特征。

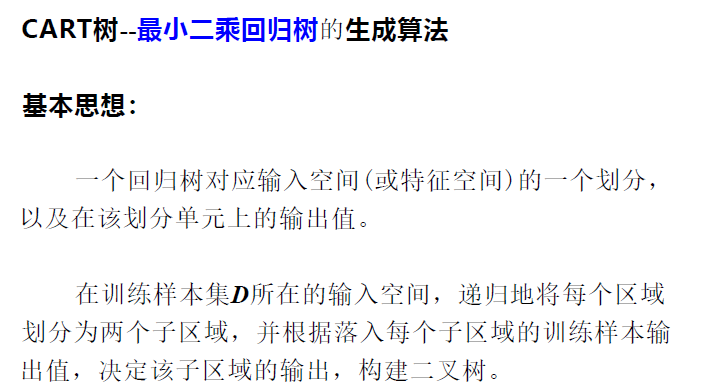
③CART决策树：基于划分后基尼指数最小的特征选择原则。（既能用于分类也可用于回归）

CART树一定是二叉树！

CART分类树算法：



CART回归树算法：



过拟合：具有较低训练误差的模型，其泛化误差可能高于具有较高训练误差的模型，这种情况称为模型过拟合(过学习)。  
欠拟合：决策树规模很小时，训练和检验误差都很大， 这种情况为模型的欠拟合(欠学习)，原因是模型尚未学习到数据的真实结构。

剪枝：

1. 预剪枝：在完全拟合整个训练集之前就停止决策树的生长。
2. 后剪枝：初始阶段--决策树按照最大规模生长。剪枝阶段--修剪完全增长的决策树。