决策树(Decision Tree)：通俗易懂

作者：writer 来源：新浪博客

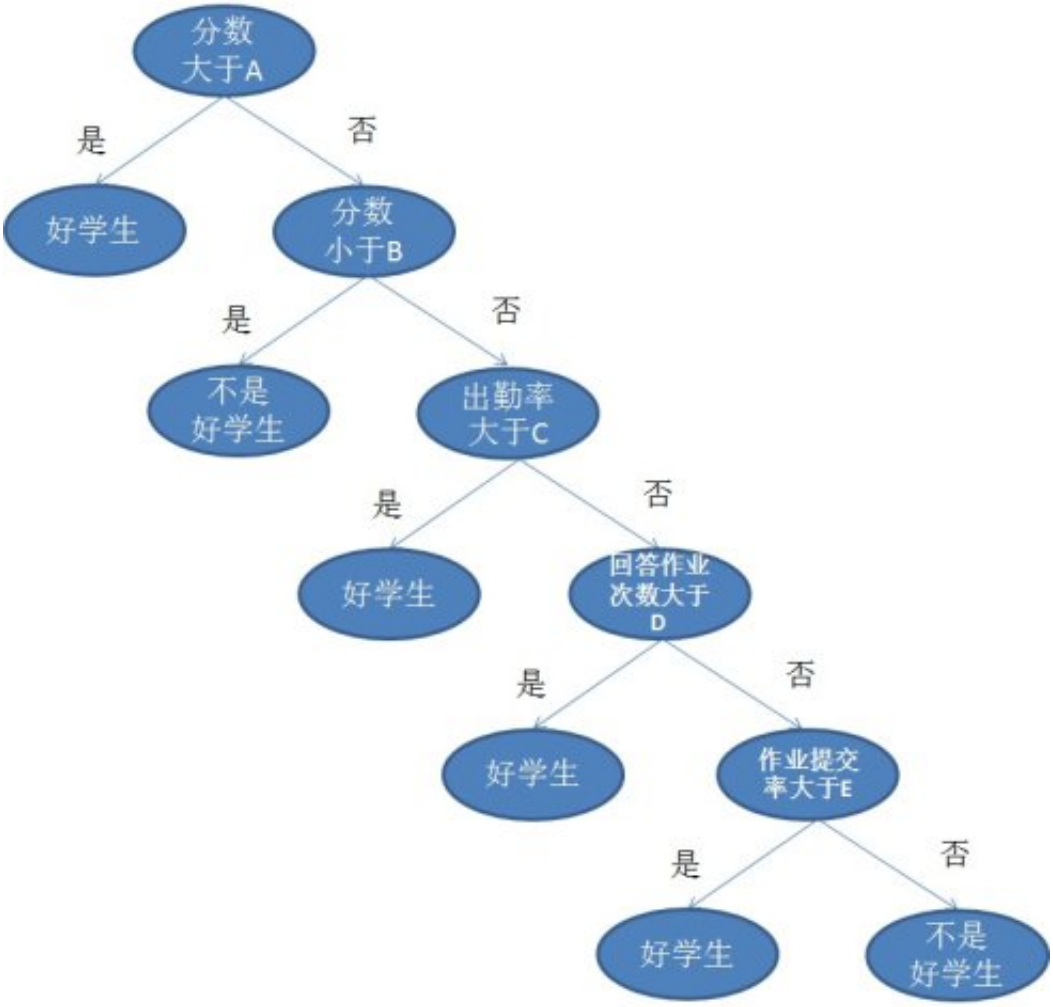
本文原地址：https://www.iikx.com/news/statistics/816.html

决策树是一种机器学习的方法。决策树的生成算法有ID3, C4.5和C5.0等。决策树是一种树形结构，其中每个内部节点表示一个属性上的判断，每个分支代表一个判断结果的输出，最后每个叶节点代表一种分类结果。决策树是一种十分常用的分类方法，需要监管学习(有教师的Supervised Learning)，监管学习就是给出一堆样本，每个样本都有一组属性和一个分类结果，也就是分类结果已知，那么通过学习这些样本得到一个决策树，这个决策树能够对新的数据给出正确的分类。这里通过一个简单的例子来说明决策树的构成思路：

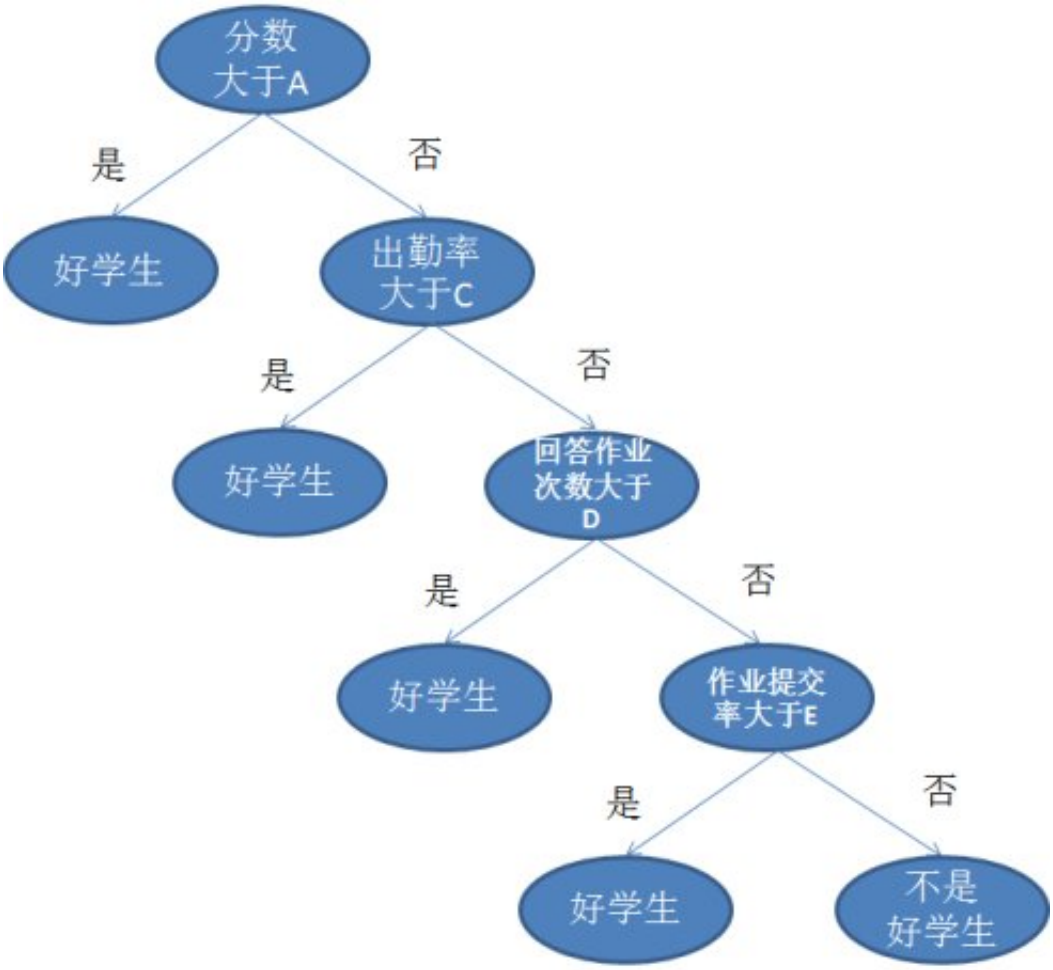
给出如下的一组数据，一共有十个样本(学生数量)，每个样本有分数，出勤率，回答问题次数，作业提交率四个属性，最后判断这些学生是否是好学生。最后一列给出了人工分类结果。

学生编号	分数	出勤率	回答问题次数	作业提交率	分类:是否好学生
1	99	80%	5	90%	是
2	89	100%	6	100%	是
3	69	100%	7	100%	否
4	50	60%	8	70%	否
5	95	70%	9	80%	否
6	98	60%	10	80%	是
7	92	65%	11	100%	是
8	91	80%	12	85%	是
9	85	80%	13	95%	是
10	85	91%	14	98%	是

然后用这一组附带分类结果的样本可以训练出多种多样的决策树，这里为了简化过程，我们假设决策树为二叉树，且类似于下图：



通过学习上表的数据，可以A，B，C，D，E的具体值，而A，B，C，D，E则称为阈值。当然也可以有和上图完全不同的树形，比如下图这种的：



所以**决策树的生成**主要分以下两步，这两步通常通过学习已经知道分类结果的样本来实现。

1. 节点的分裂：一般当一个节点所代表的属性无法给出判断时，则选择将这一节点分成2个

子 节点(如不是二叉树的情况会分成n个子节点)

2. 阈值的确定：选择适当的阈值使得分类错误率最小 (Training Error)。

比较常用的决策树有ID3，C4.5和CART(Classification And Regression Tree)，CART的分类效果一般优于其他决策树。下面介绍具体步骤。

ID3:
由增熵(Entrophy)原理来决定那个做父节点，那个节点需要分裂。对于一组数据，熵越大说明分类结果越好。

比如上表中的4个属性：单一地通过以下语句分类：

1. 分数小于70为【不是好学生】：分错1个

2. 出勤率大于70为【好学生】：分错3个

3. 问题回答次数大于9为【好学生】：分错2个

4. 作业提交率大于80%为【好学生】：分错2个

最后发现 分数小于70为【不是好学生】这条分错最少，也就是熵最大，所以应该选择这条为父节点进行树的生成，当然分数也可以选择大于71，大于72等等，出勤率也可以选择小于60，65等等，总之会有很多类似上述1~4的条件，最后选择分类错最少即熵最大的那个条件。而当分裂父节点时道理也一样，分裂有很多选择，针对每一个选择，与分裂前的分类错误率比较，留下那个提高最大的选择，即熵增益最大的选择。

C4.5：通过对ID3的学习可以知道ID3存在一个问题

，那就是越细小的分割分类错误率越小，所以ID3会越分越细，比如以第一个属性为例：设阈值小于70可将样本分为2组，但是分错了1个。如果设阈值小于70，再加上阈值等于95，那么分错率降到了0，但是这种分割显然只对训练数据有用，对于新的数据没有意义，这就是所说的过度学习(Overfitting)。分割太细了，训练数据的分类可以达到0错误率，但是因为新的数据和训练数据不同，所以面对新的数据分错率反倒上升了。决策树是通过分析训练数据，得到数据的统计信息，而不是专为训练数据量身定做。就比如给男人做衣服，叫来10个人做参考，做出一件10个人都能穿的衣服，然后叫来另外5个和前面10个人身高差不多的，这件衣服也能穿。但是当你为10个人每人做一件正好合身的衣服，那么这10件衣服除了那个量身定做的人，别人都穿不了。所以为了避免分割太细，c4.5对ID3进行了改进，C4.5中，增加的熵要除以分割太细的代价，这个比值叫做信息增益率，显然分割太细分母增加，信息增益率会降低。除此之外，其他的原理和ID3相同。

CART：分类回归树

CART是一个二叉树，也是回归树，同时也是分类树，CART的构成简单明了。

CART只能将一个父节点分为2个子节点。CART用GINI指数来决定如何分裂：

GINI指数：总体内包含的类别越杂乱，GINI指数就越大(跟熵的概念很相似)。

a. 比如出勤率大于70%这个条件将训练数据分成两组：大于70%里面有两类：【好学生】和【不是好学生】，而小于等于70%里也有两类：【好学生】和【不是好学生】。

b. 如果用分数小于70分来分：则小于70分只有【不是好学生】一类，而大于等于70分有【好学生】和【不是好学生】两类。

比较a和b，发现b的凌乱程度比a要小，即GINI指数b比a小，所以选择b的方案。以此为例，将所有条件列出来，选择GINI指数最小的方案，这个和熵的概念很类似。

CART还是一个回归树，回归解析用来决定分布是否终止。理想地说每一个叶节点里都只有一个类别时分类应该停止，但是很多数据并不容易完全划分，或者完全划分需要很多次分裂，必然造成很长的运行时间，所以CART可以对每个叶节点里的数据分析其均值方差，当方差小于一定值可以终止分裂，以换取计算成本的降低。

CART和ID3一样，存在偏向细小分割，即过度学习(过度拟合的问题)，为了解决这一问题，对特别长的树进行剪枝处理，直接剪掉。

以上的决策树训练的时候，一般会采取Cross-Validation法：比如一共有10组数据：

第一次. 1到9做训练数据， 10做测试数据

第二次. 2到10做训练数据，1做测试数据

第三次. 1，3到10做训练数据，2做测试数据，以此类推

做10次，然后大平均错误率。这样称为 10 foldsCross-Validation。

比如 3 foldsCross-Validation 指的是数据分3份，2份做训练，1份做测试。