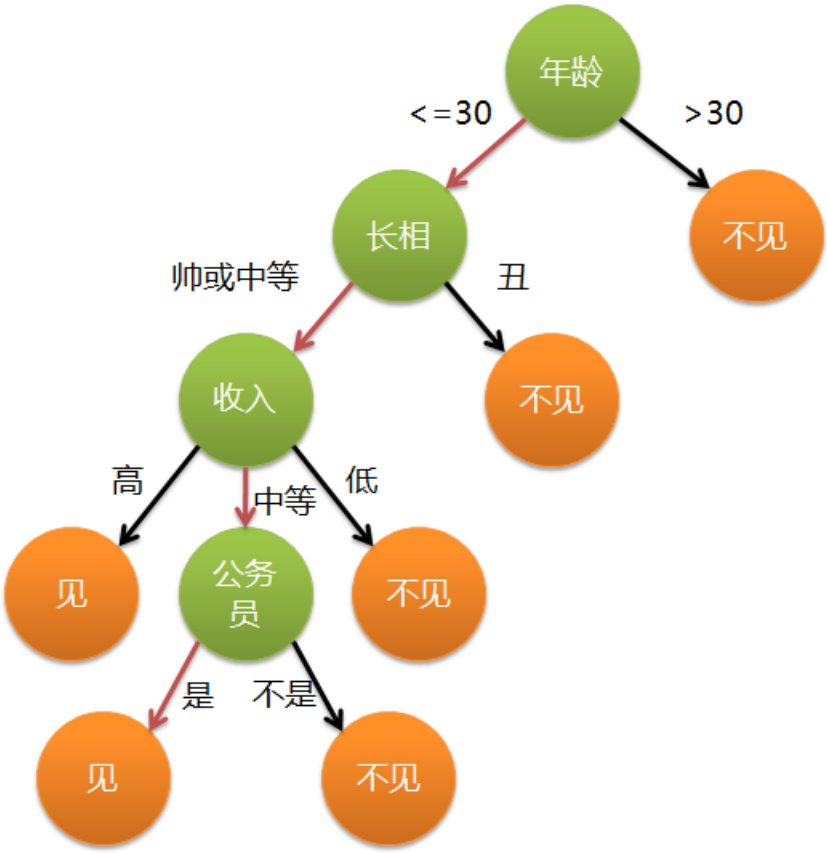
一、 决策树算法简介

我们先来看一个例子：一个女孩打算去相亲，那么自然要对现有的男性进行一个筛选，年龄太大的不见，长得丑的不见，收入太低不见，收入正常但不是公务员的也不见。那么根据这些条件，结合数据结构中学到的树结构，我们很容易得到下图：



这个图表达的就是这个女孩是否去见一个相亲对象的策略，其中绿色的节点表示一个判断条件，橙色的圆表示一个决策结果，箭头则是决策过程。

决策树是一个预测模型，他代表的是对象属性和对象值之间的一种映射关系。决策树是可以是二叉树或非二叉树。其每个非叶节点表示一个特征属性上的测试，每个分支代表这个特征属性在某个值域上的输出，而每个叶节点存放一个类别。

使用决策树进行决策的过程就是从根节点开始，测试待分类项中相应的特征属性，并按照其值选择输出分支，直到到达叶子节点，将叶子节点存放的类别作为决策结果。

总结来说：

决策树模型核心是下面几部分：

- 结点和有向边组成
- 结点有内部结点和叶结点俩种类型
- 内部结点表示一个特征，叶节点表示一个类

二、 决策树的构建

在构建决策树时，并不需要依赖专业的领域知识。例如构建一个疾病预诊断的决策树并不需要对医学有什么深入的研究。决策树使用的是属性选 们可以看到在每个绿色节点处 都会把当前的集合进行分类，也就是在某个节点处，按照某个特征属性的不同来划分构造不同的分支，同时，要确定分裂出来的子集尽量纯净，也就是让一个分裂子集中待分类的选项属于同一类别。分裂属性时按照属性值的特点可以分为三种情况：

1. 属性值是离散的且不要求生成二叉决策树，则按照该属性的每种属性值进行划分；
2. 属性值是离散的且要求生成二叉决策树，则使用该属性划分的一个子集来测试，把数据按照属于该子集和不属于该子集，划分为两类；

3. 属性值是连续的，则确定一个值作为分裂点，按照大于该数值和小于等于该数值，生成两个分支。

生成决策树的流程一般来说可以分为三步，如下图所示

特征选择时需要从训练数据中选取一个特征来作为当前节点进行属性分裂，根据选取特征的方法不同就产生了多种的决策树算法；

决策树生成时，依据选择的特征评估标准，从上到下递归地生成子节点，直到数据不可分为止；

决策树容易产生过拟合现象，也就是决策树太过于“茂盛”，节点太多，因此就需要进行剪枝。剪枝的策略对决策树的正确率影响非常大，常用的剪枝方法一般有前置剪枝和后置剪枝。

前置裁剪是在构建决策树的过程时，提前停止。这会导致切分节点的条件设置过于苛刻，导致决策树很短小。结果就是决策树无法达到最优。实践证明这中策略无法得到较好的结果。

后置裁剪则是决策树构建好后，才开始裁剪。采用两种方法：

1）用单一叶节点代替整个子树，叶节点的分类采用子树中最主要的分类；

2）将一个子树完全替代另外一颗子树。

后置裁剪的问题在于计算效率，部分节点计算后就被裁剪会导致浪费。

三、 数据纯度函数

选择特征的目标是让分类后的数据集比较纯净，为了衡量数据集的纯净度，就需要用到数据纯度函数。

1. 信息熵

在概率论中，信息熵给了我们一种度量不确定性的方式，是用来衡量随机变量不确定性的，熵就是信息的期望值。若待分类的事物可能划分在 N 类中，分别是x_1，x_2，$\cdots x_{n,}$ 每一种取到的概率分别是p_1，p_2，$\cdots p_{n,}$ 那么数据集 D 的熵定义为

$$H(D) = -\sum_{i=1}^{|n|} p_i \log p_i$$

其中 0≤ $H(D) \leq \log(n)$

熵越大，随机变量的不确定性就越大。当随机变量只有两个值的时候，例如 P(X=1)=p，P(X=0)=1-p 时，熵为

$$H(P) = -p \log_2 p - (1-p) \log_2 (1-p)$$

2. 信息增益

特征对训练数据集 D 的信息增益 g（D,A），定义为集合 D 的经验熵 H(D)与特征 A 给定条件 D 的经验条件熵 H(D|A)之差。

3. 基尼指数

分类问题中，假设有 K 个类，样本点属于第 k 类的概率为p_k，则概率分布的基尼指数定义为

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

对于二分类问题，样本属于第一个类的概率是 p，则概率分布的基尼指数为

$$Gini(p) = 2p(1 - p)$$

对于给定的样本集合 D，其基尼指数为

$$Gini(D) = 1 - \sum_{k=1}^{K}\left(\frac{|C_k|}{|D|}\right)^2$$

其中C_k是 D 中属于第 k 类的样本子集，K 是类的个数。

如果样本集合 D 根据特征 A 是否取某一可能值 a 被分割为D_1和D_2两部分，则在特征 A 的条件下，集合 D 的基尼指数定义为

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

基尼指数表示集合 D 的不确定性，基尼指数越大，样本集合的不确定性也就越大。

四、 决策树构建算法

常用的决策树生成算法主要有以下三种：

1. ID3 算法

ID3 算法的核心思想是在决策树的各个结点上应用信息增益准则来选择特征并构建决策树。流程为：

- 从根结点开始计算所有可能的特征的信息增益，选择信息增益最大的特征作为该结点的特征并据此建立子结点
- 对子结点递归地调用上述方法构建出决策树
- 直到所有特征的信息增益都非常小，或者是没有特征可以选择时就结束，并且获得最终的决策树。

在 ID3 算法中，输入为训练数据集 D，特征集 A，和阈值ε；输出为决策树 T。该算法的具体实现步骤如下：

1. 若 D 中所有实例属于同一类C_k，则 T 为单结点树，并将类C_k作为该结点的类标记，返回 T；
2. 若 A 为空集，则 T 为单结点树，并且把 D 中实例数最大的类C_k作为该结点的类标记，返回 T；
3. 若 A 不空，则按照算法

$$H(D) = -\sum_{i=1}^{|n|} p_i \log p_i$$

计算 A 中各特征对 D 的信息增益，选择信息增益最大的特征A_g；

4. 若A_g的信息增益小于阈值ε，则置 T 为单结点树，并把 D 中实例数最大的类C_k作为该结点的类标记，返回 T；
5. 否则，对A_g的每一个可能值a_i，按照$A_g = a_i$把 D 分割成为若干个非空子集D_i，把D_i中最大的类作为标记，构建子结点，由结点和其子结点构成树 T 并返回；
6. 对于第 i 个子结点，以D_i为训练集，以$A - \{A_g\}$作为特征集，递归调用步骤 1-5，并得到和返回子树。

2. C4.5 算法

C4.5 算法和 ID3 很类似，在生成的时候是采用信息增益比来选择特征的。C4.5 算法输入为训练数据集 D，特征集 A，和阈值ε；输出为决策树 T。该算法的具体实现步骤如下：

1. 若 D 中所有实例属于同一类C_k，则 T 为单结点树，并将类C_k作为该结点的类标记，返回 T；
2. 若 A 为空集，则 T 为单结点树，并且把 D 中实例数最大的类C_k作为该结点的类标记，返回 T；
3. 若 A 不空，则按照算法

$$g_R(D, A) = \frac{g(D, A)}{H(D)}$$

计算 A 中各特征对 D 的信息增益比，选择信息增益比最大的特征A_g；

4. 若A_g的信息增益小于阈值ε，则置 T 为单结点树，并把 D 中实例数最大的类C_k作为该结点的类标记，返回 T；
5. 否则，对A_g的每一个可能值a_i，按照$A_g = a_i$把 D 分割成为若干个非空子集D_i，把D_i中最大的类作为标记，构建子结点，由结点和其子结点构成树 T 并返回；
6. 对于第 i 个子结点，以D_i为训练集，以$A - \{A_g\}$作为特征集，递归调用步骤 1-5，并得到和返回子树。

3. CART 算法

CART 是在给定输入随机变量 X 的条件下输出随机变量 Y 的条件概率分布的学习方法。该算法利用基尼指数，假定决策树是二叉树，输入为训练数据集 D，停止计算的条件；输出为决策树 T。该算法的具体实现步骤如下：

1. 设结点的训练数据集为 D，计算现有特征对该数据集的基尼指数。对每一个特征 A，对其可能的取值 a，根据样本点对 A=a 的测试把 D 分割成为D_1和D_2两个部分，并且利用公式

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

计算 A=a 时的基尼指数；

2. 在所有可能的特征 A 以及其所有可能的切分点 a 中，选择基尼指数最小的特征以及其对应的切分点作为最优特征和最优切分点。按照

最优特征和最优切分点从现结点生成两个子结点，并且把训练数据集按照特征分配到两个子结点中；

3. 对两个子结点递归地调用步骤 1 和 2，直到满足停止条件；
4. 生成并返回 CART 决策树。

该算法的停止条件是结点中样本的个数小于预定阈值，或者样本集的基尼指数小于预定阈值，或者是没有更多的特征。

五、 决策树的优缺点

1. 优点

- 决策树易于理解和实现。人们在通过解释后都有能力去理解决策树所表达的意义。
- 对于决策树，数据的准备往往是简单或者是不必要的.其他的技术往往要求先把数据一般化，比如去掉多余的或者空白的属性。
- 能够同时处理数据型和常规型属性。其他的技术往往要求数据属性的单一。
- 是一个白盒模型，如果给定一个观察的模型，那么根据所产生的决策树很容易推出相应的逻辑表达式。
- 易于通过静态测试来对模型进行评测。表示有可能测量该模型的可信度。
- 在相对短的时间内能够对大型数据源做出可行且效果良好的结果。

2. 缺点

- 对于那些各类别样本数量不一致的数据，在决策树当中信息增益的结果偏向于那些具有更多数值的特征。
- 可规模性一般，连续变量需要划分成离散变量，容易过拟合。