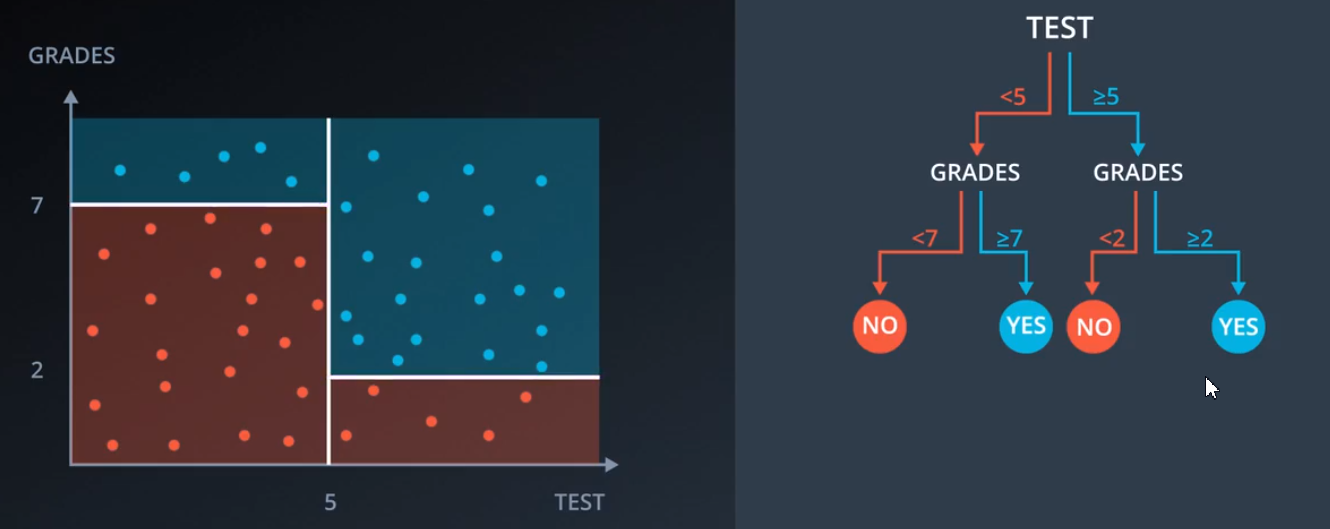
我们构建了一个具有分类特征的树，即以性别和职业进行分类。

同样我们可以建构具有连续特征的树，这是一个招生办公室获取的关于学生的两项数据，即他们在一场考试中的考试成绩和等级。

这些学生的考试成绩和等级越好，则他们被大学录取的可能性越大



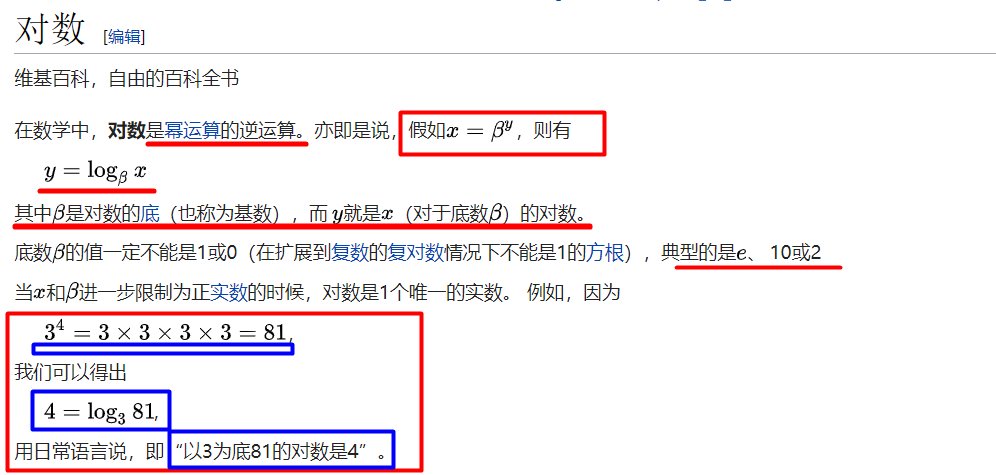
使用纵坐标切线，这意味着分离这些数据的最佳特征是考试成绩

最佳阈值是 5，因此 我们可以为决策树添加第一个结点，即 你的考试成绩是大于 等于还是小于 5。

# 00．熵定义

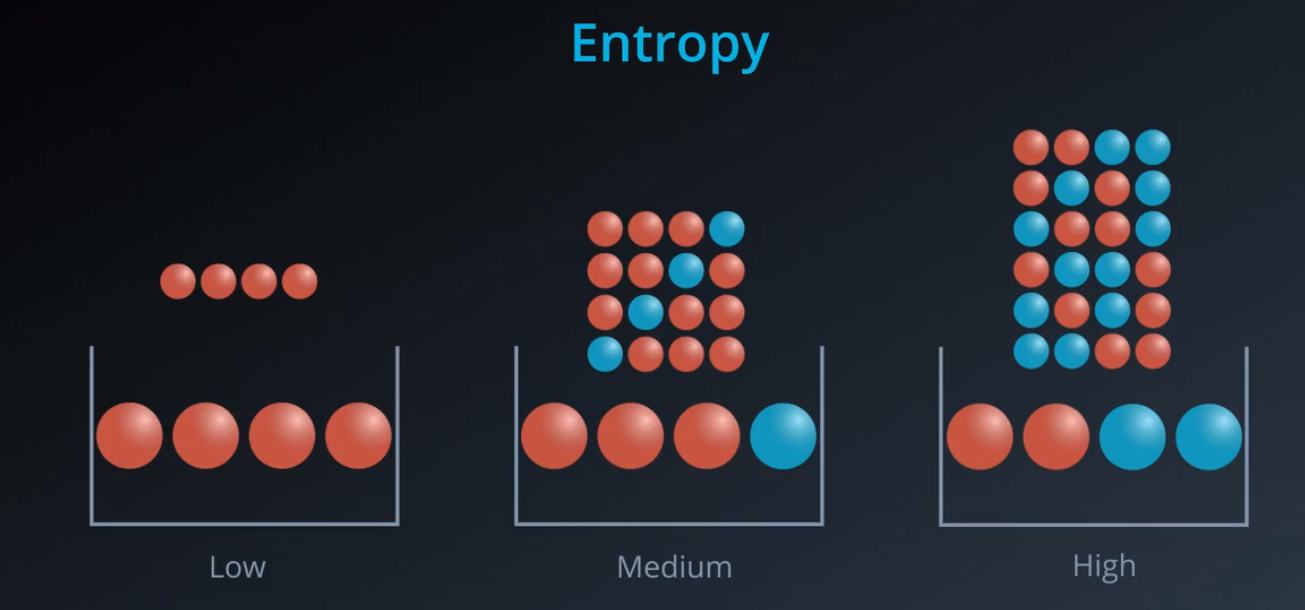
**熵**（英语：entropy）是接收的每条消息中包含的信息的***平均量***，又被称为**信息熵**、**信源熵**、**平均自信息量**。这里，“消息”代表来自 **分布** 或 **数据流中的事件、样本或特征。**

**对数**



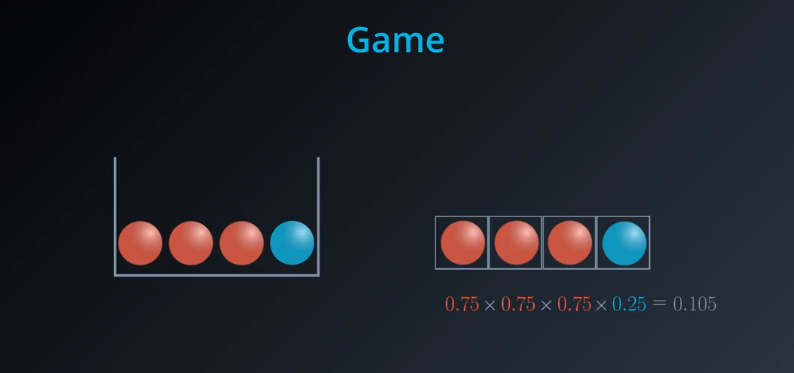
# 07．熵的理解

概率解释（熵就是球移动的自由程度）



集合越稳固或越具有同类性，其熵越低，反之亦然。

计算3红1蓝的出现的概率



球的颜色为三红一蓝,取出第一个球为红色的概率是多少？

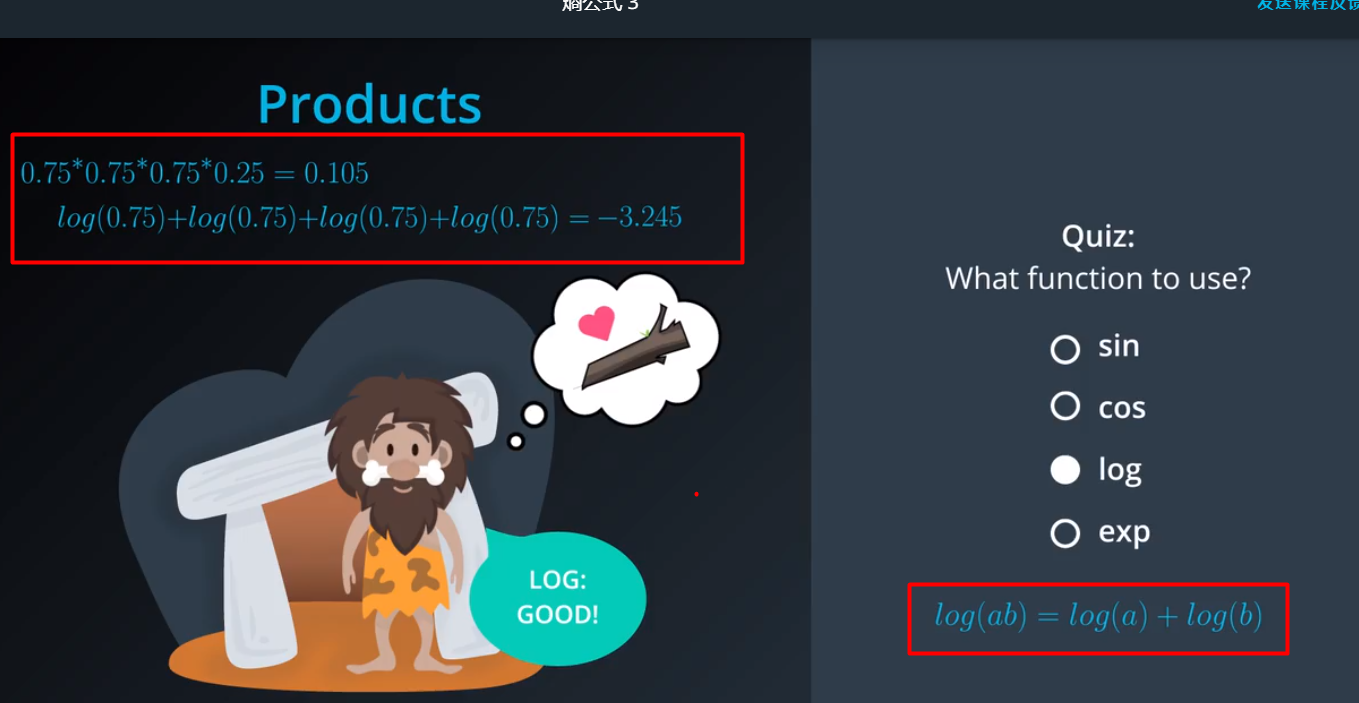
是四分之三或 0.75 ，因为桶里的四个球有三个红的,取出第二 三个球为红色的概率也是 0.75

那么 我们取出的第四个球为蓝色的概率为多少呢？

是四分之一 因为四个球中只有一个蓝球

由于我们每次记录取出的球的颜色之后会将球放回,所以这**四个事件是完全独立的**

因此，三红一蓝的概率是四个事件概率的积，既0.105 。



第一个问题是

假如我们有一千个球 我们得到一千个球的积 这个积总位于零和一之间 有可能非常小？

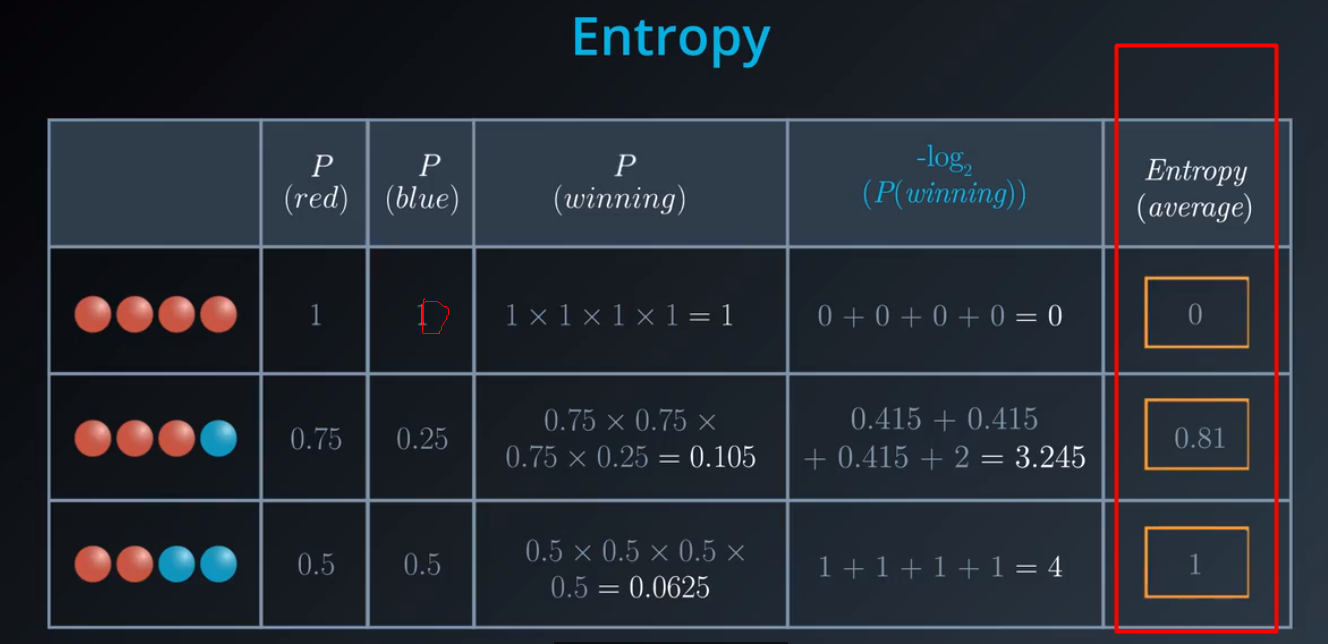
第二个问题是

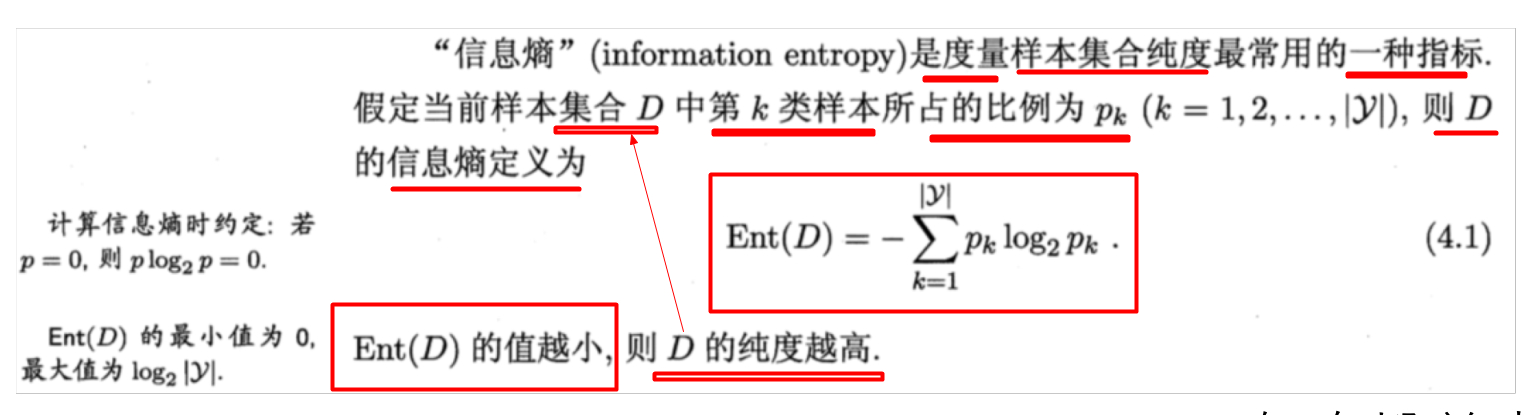
稍微改变其中一个因素 就会对最后的积产生极大的影响 我们需要更可控的因素 还有什么比积更好呢？

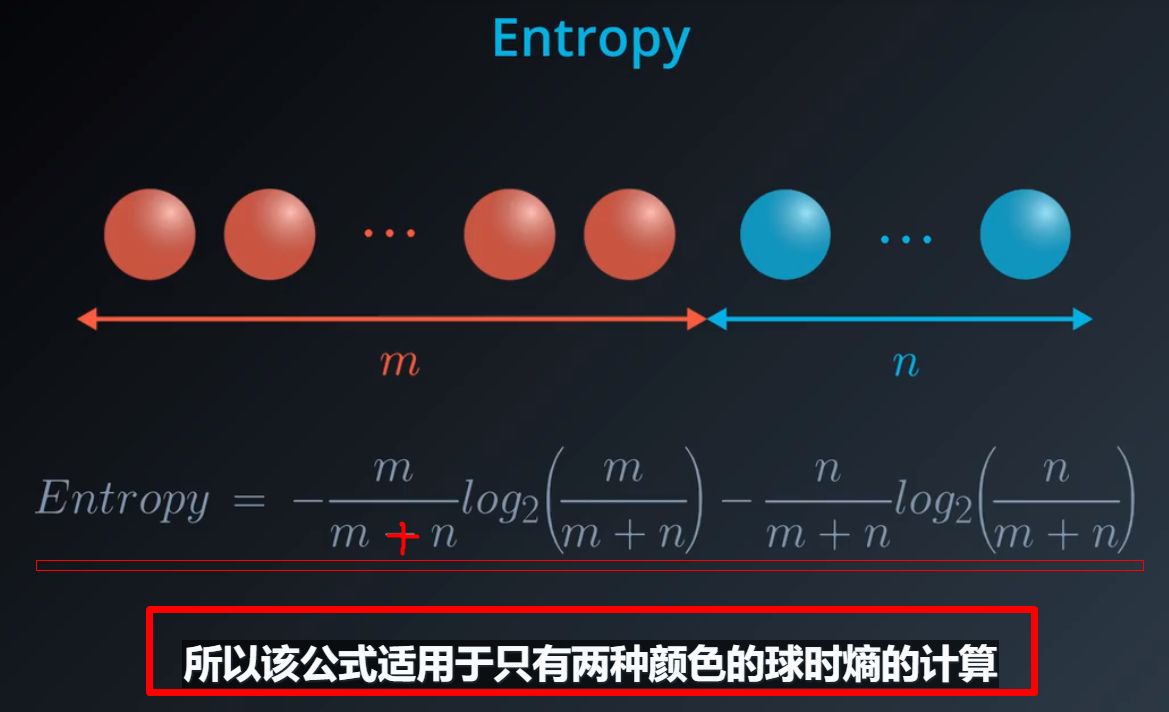
最好的方式为：将积改为和

# 10．熵公式

**熵的计算公式**







总结：熵越大，集合纯度越低，预测概率就越低。

# 12．信息增益

信息增益

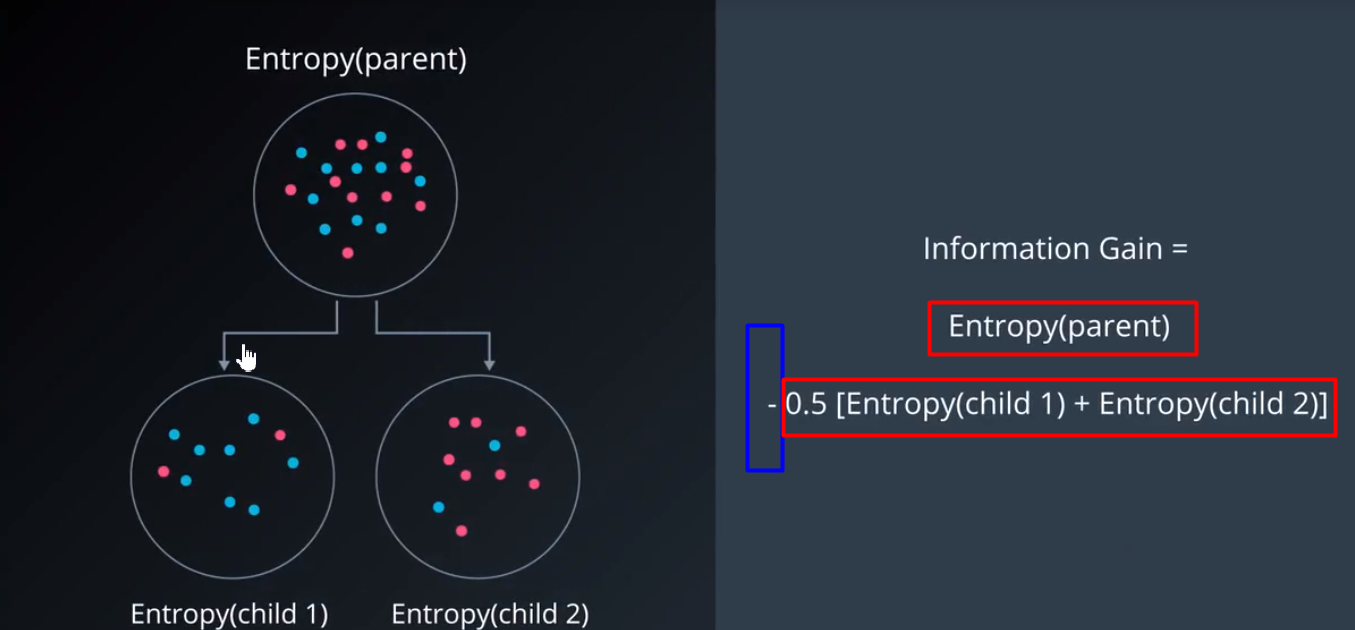
在决策树中的每一个结点处

我们可以计算父结点处数据的熵

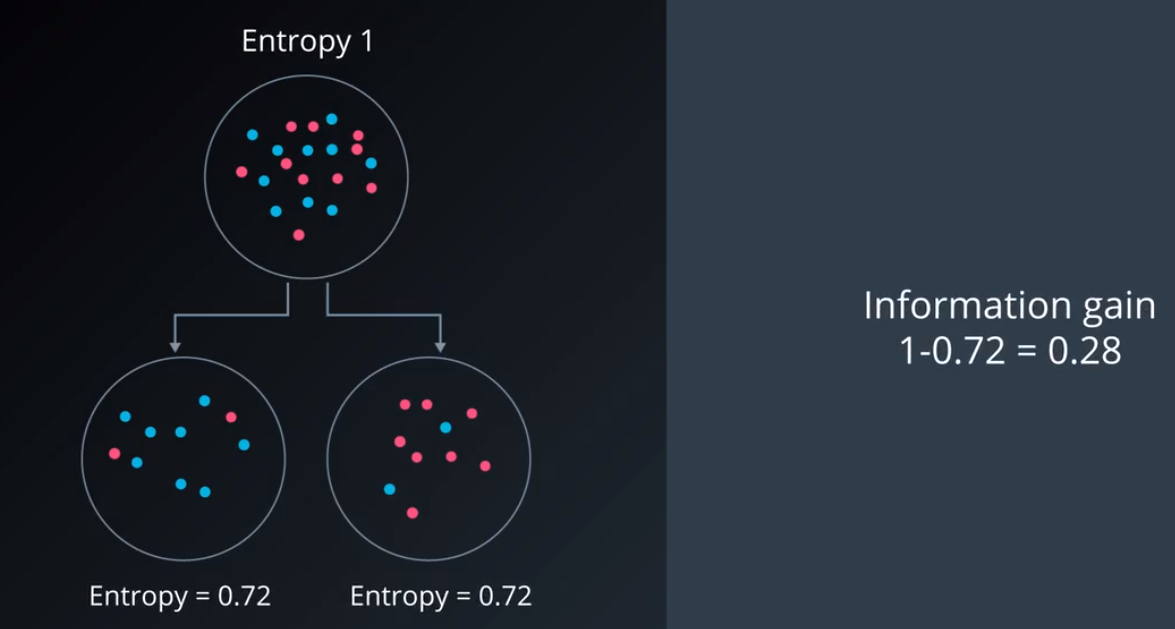
然后计算两个子结点的熵

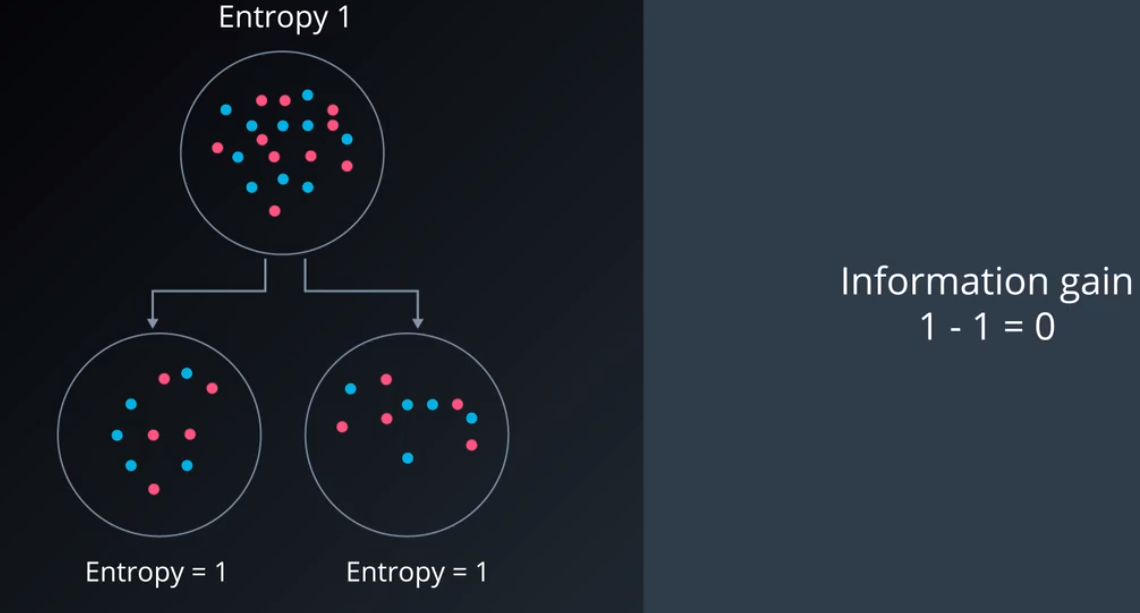
父结点的熵与子结点熵 平均值之间 的差值

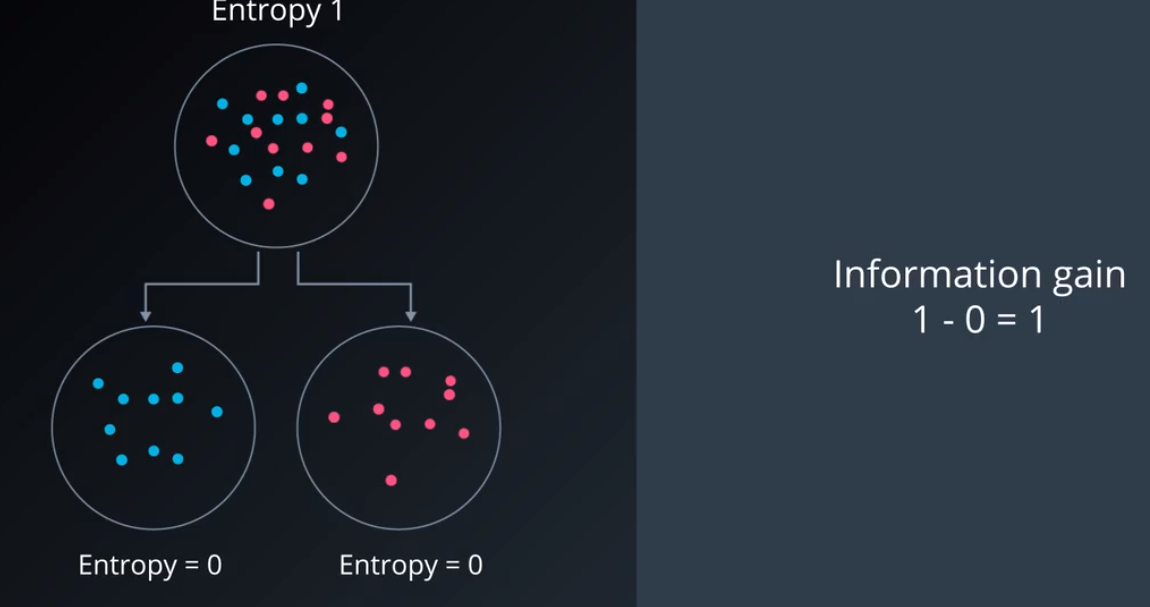
即为信息增益

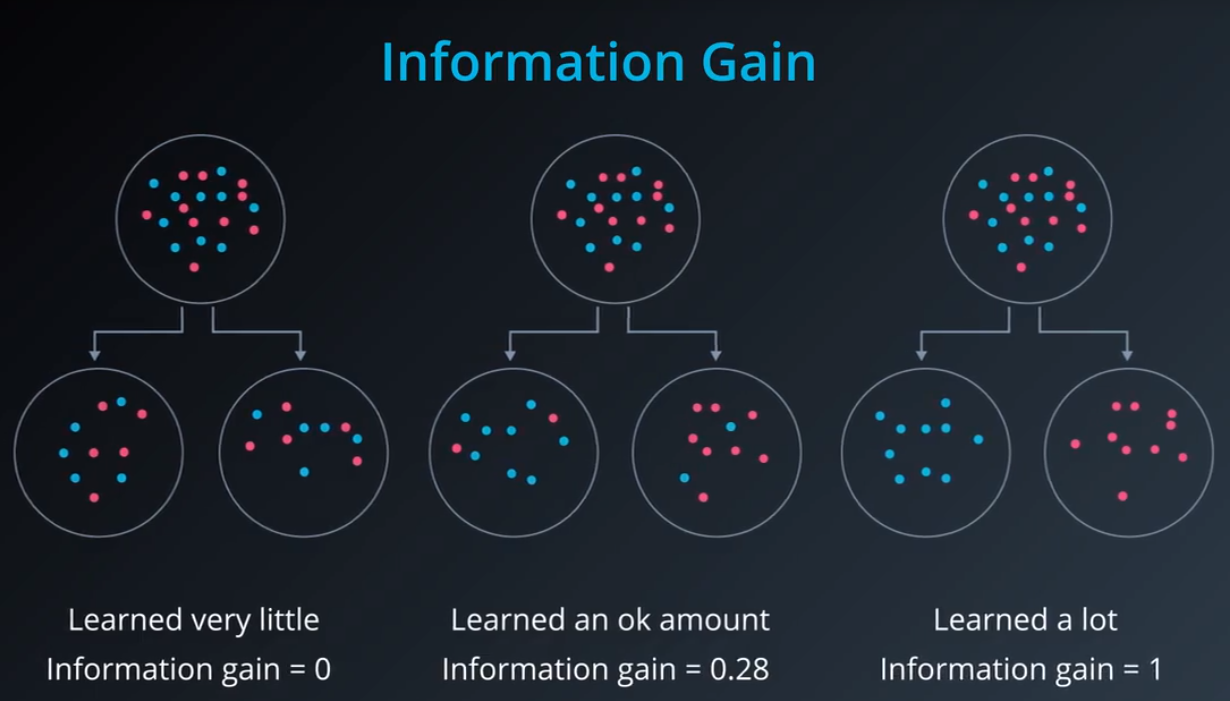


计算方法









第一种分法完全没用，分割后的数据与原数据都包含相似的红点和蓝点，对于我们了解数据没有帮助

第二种分法就比较好了，我们可以把大多数蓝点分在一边，而把大多数红点分在另一边

所以 我们对于数据有了一定了解。

第三种分法就更棒了，它把所有的蓝点分在一边，并把所有的红点分在另一边

**总结**

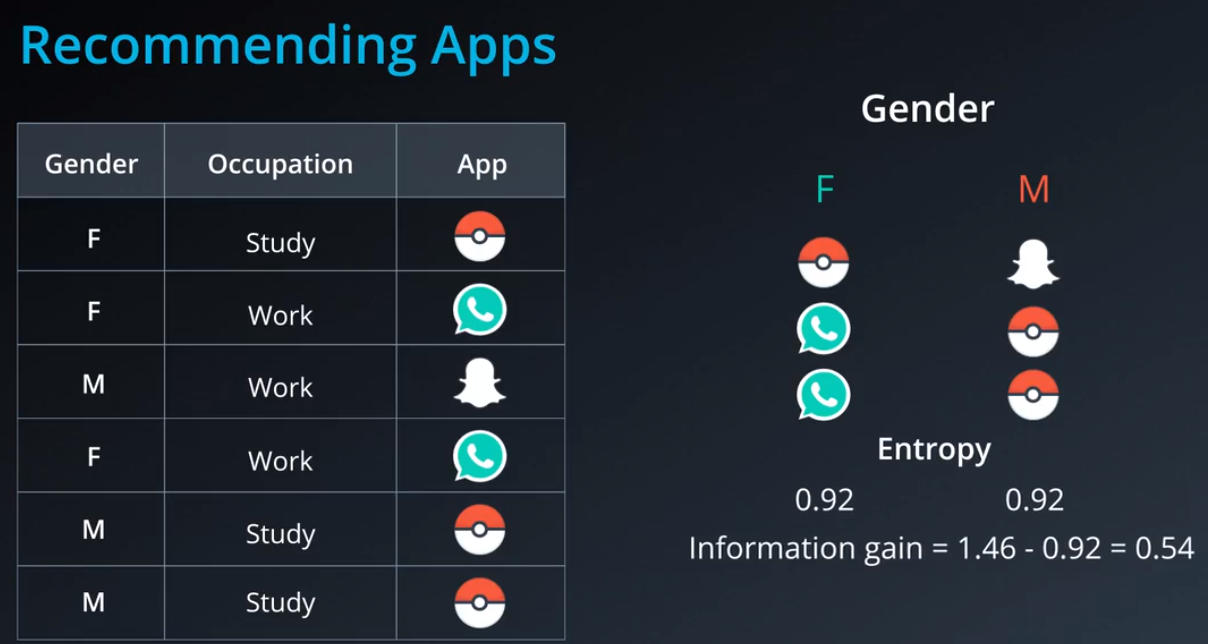
以上是三种分法以及其信息增益值，如果决策树必须做出选择，它将选择第三种分法，因为第三种分法提供的信息增益最大。

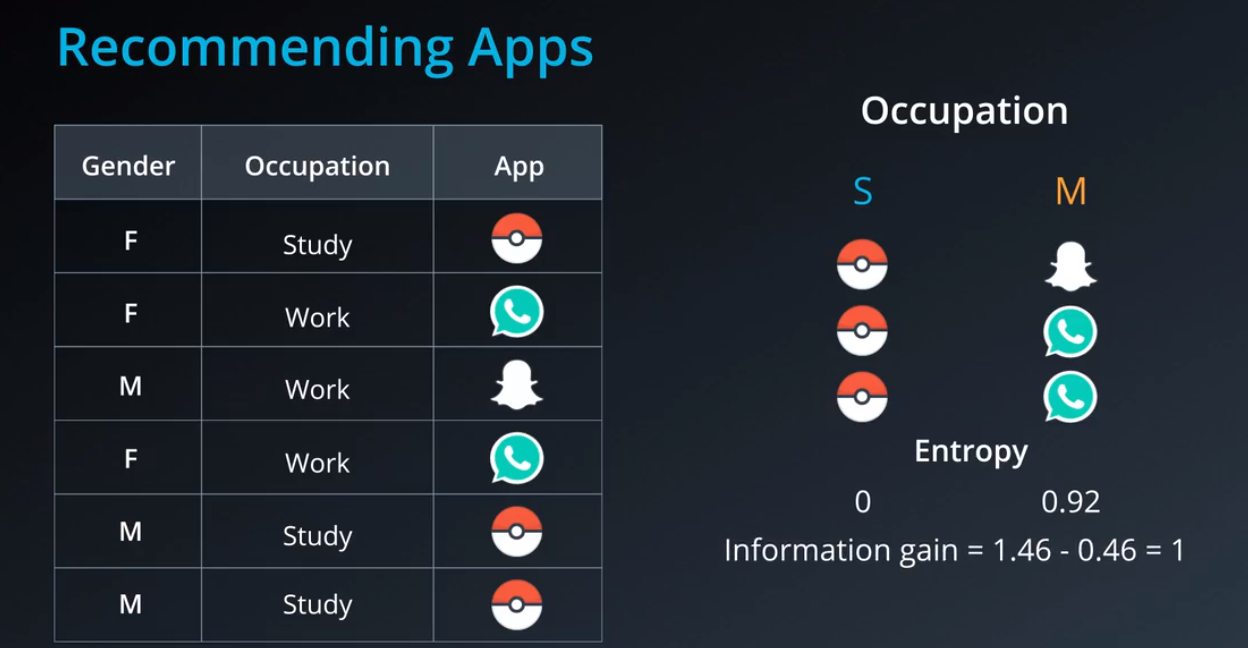
# 13. 建立决策树

计算父节点（APP）的熵



计算 子节点Gender 的熵

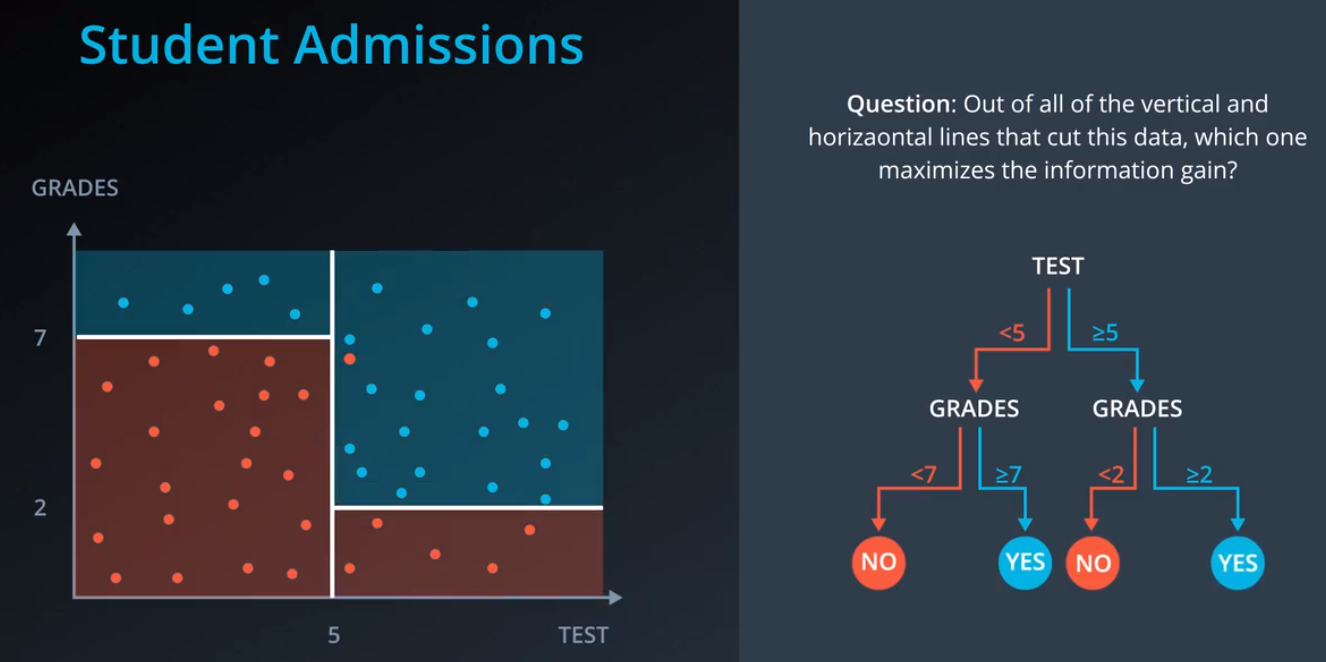




总结：按性别分，信息增益为0.54、按职业分，信息增益为1。

因此决策树 选择 信息增益大的按职（Occupation）进行分类.





连续特征决策树分类：

首先想一下其中的细节

最根本的 是要想清楚所有可能的水平切法和垂直分法

看看哪种分法的熵最大

然后在建立决策树时反复迭代

这里我们可以看到 我们的垂直切线在横坐标 5 处

水平切线在纵坐标 7 处

最后一道水平切线在纵坐标 2 处

最终就得到了有两部分数据的决策树

# 14.随机森林



决策树中潜在的问题，比如说我们有一张极大的表格 其中行列很多，然后我们建立了决策树，如上图

然后我们得到了如下结论:

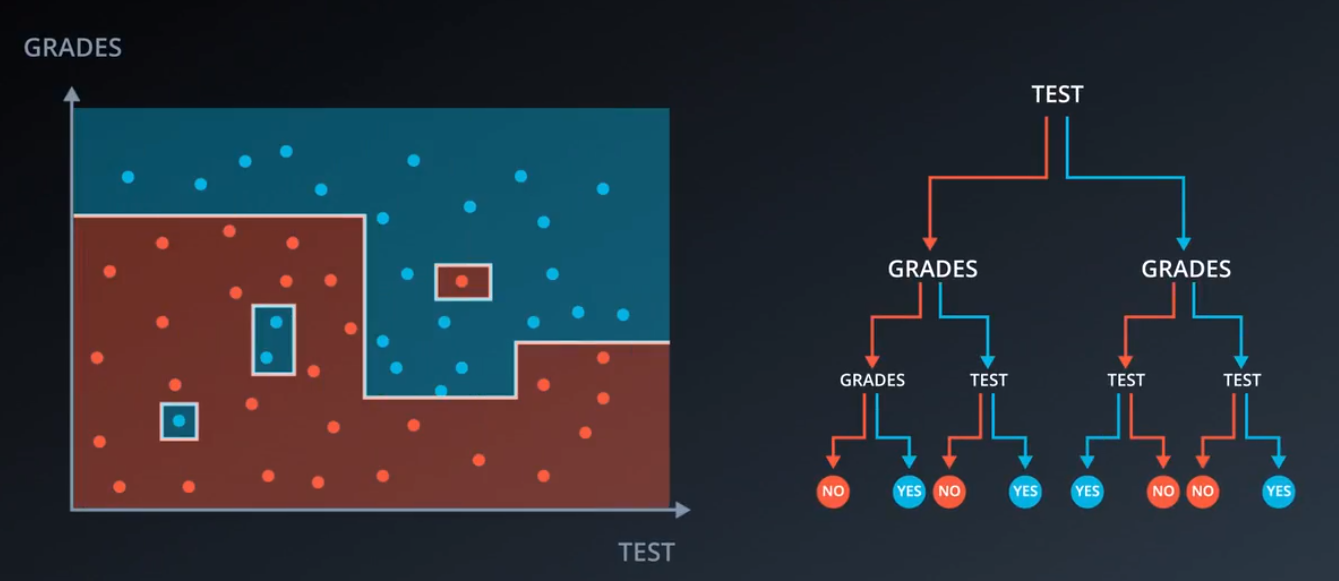
如果一位客户为男性 年龄在 15 到 25 之间 居住在美国

使用 Android 还在上学，喜欢网球和披萨 但不喜欢长时间在沙滩上长散步

那么他很可能下载 Pokemon Go

这种做法很不理想 看起来似乎只是在记忆数据

这种情况称为过拟合 决策树经常会过拟合



如果我们选用连续特征也会出现这种问题，决策树有许多结点 它最终会呈现给我们多个几乎与点相接的小方块，这些小方块将不同颜色的点分开，这也是过拟合现象 因为它对数据不具备普适性。

解决方法：

可以这样操作，随机从数据中挑选几列,并根据这些列建构决策树,然后随机选取其它几列,再次构建决策树，然后让决策树进行选择，当我们有新的数据时，比如说出现了一个新用户

就只需让所有的决策树做出预测 并选取结果中显示最多的。

比如 这些决策树觉得

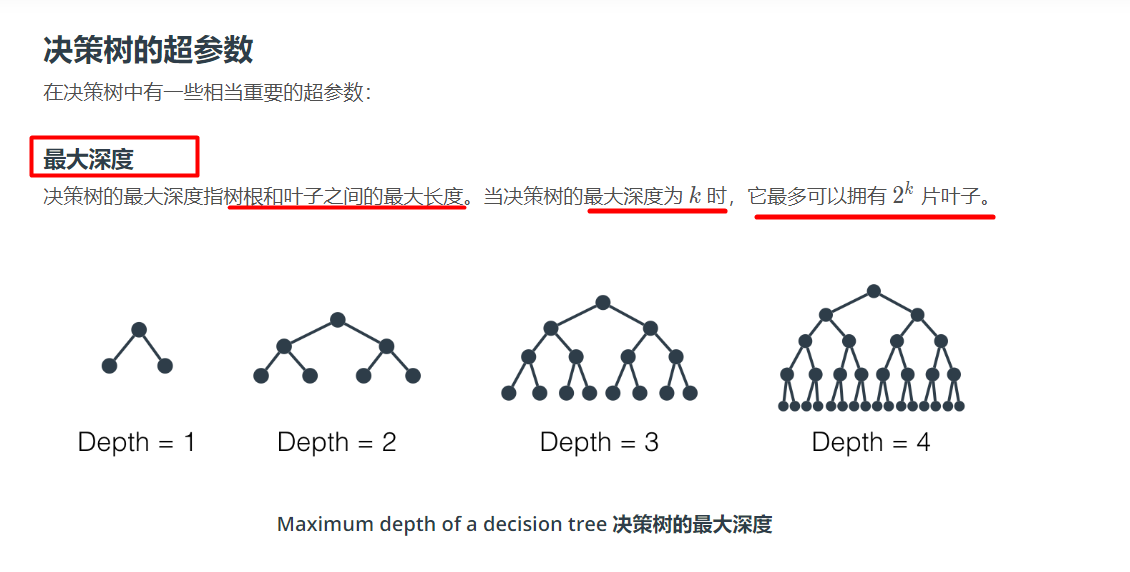
这个人会下载 Snapchat WhatsApp 和 WhatsApp

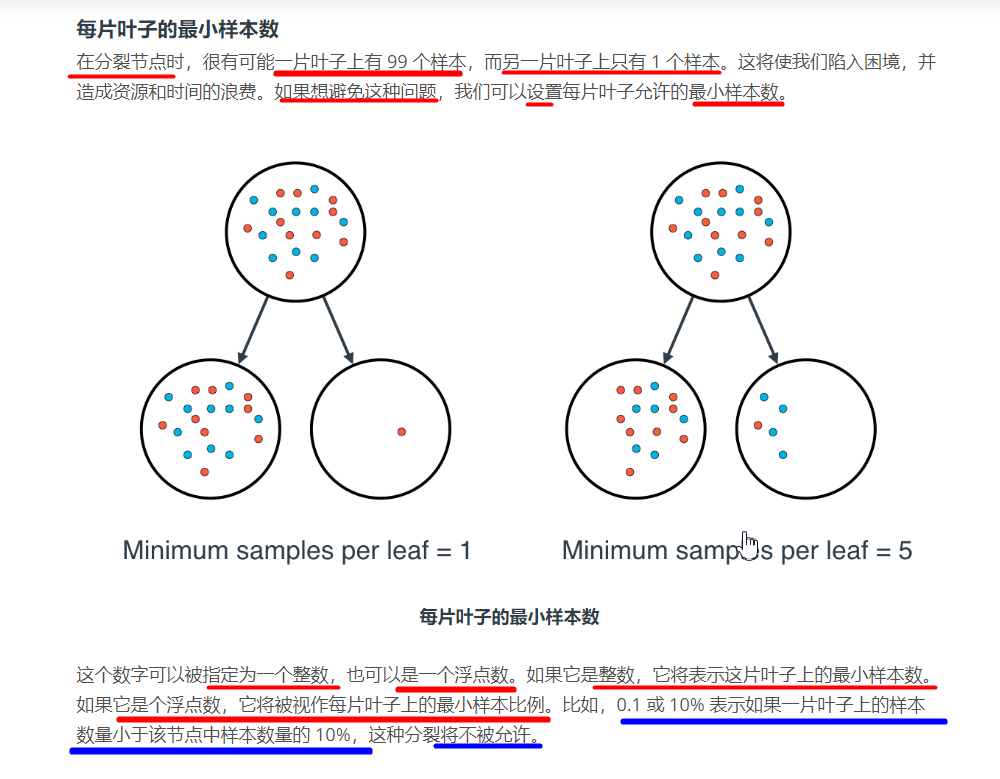
因此 决策树的集成会推荐此人下载 WhatsApp

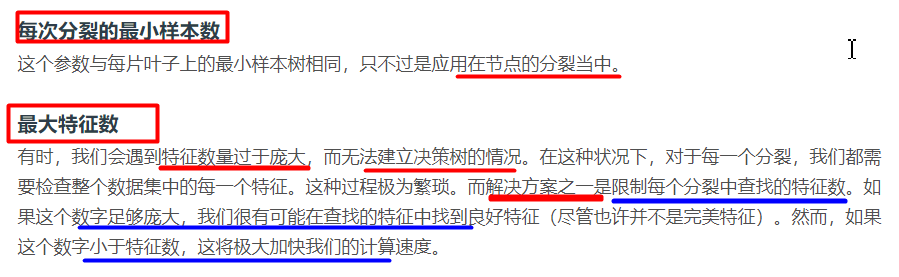
由于我们利用随机的列建构的多个决策树做出了预测

这种方法称为随机森林

# 15．超参数

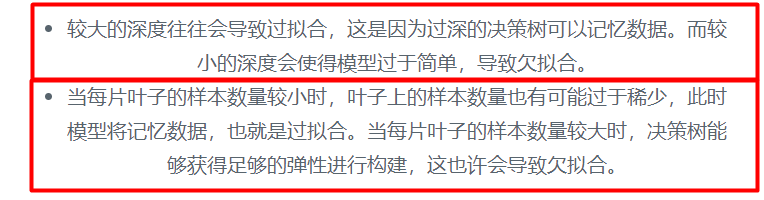






特征：

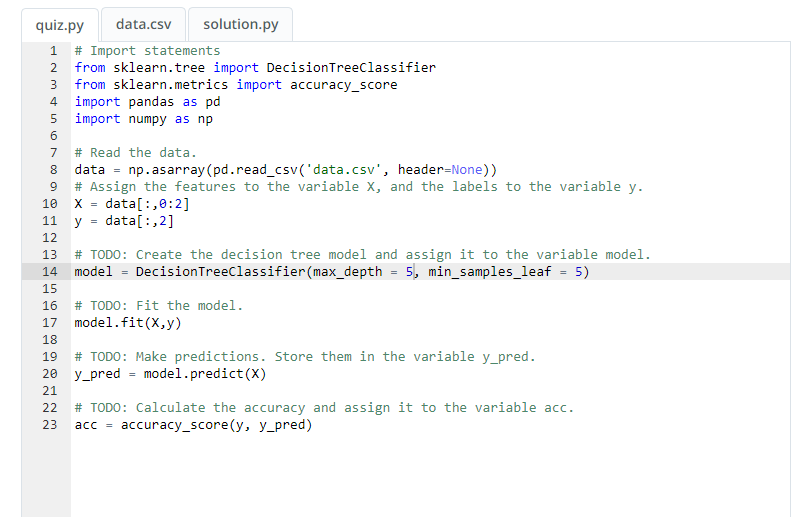


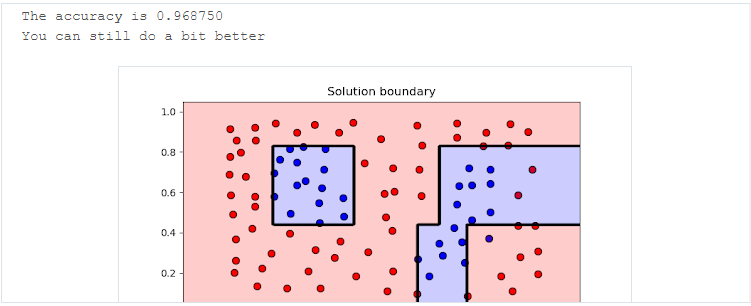


# 16．Sklearn中的决策树









总结：

1. 由于概率不好计算，特将概率积的方式 转换为 熵的和。
2. 熵越大，集合的纯度越低，数据越无法分类。
3. 信息增益，是父熵 – 1/2（子熵1 – 子熵2）。
4. 信息增益越大，数据分类效果越好，信息增益用于决策树