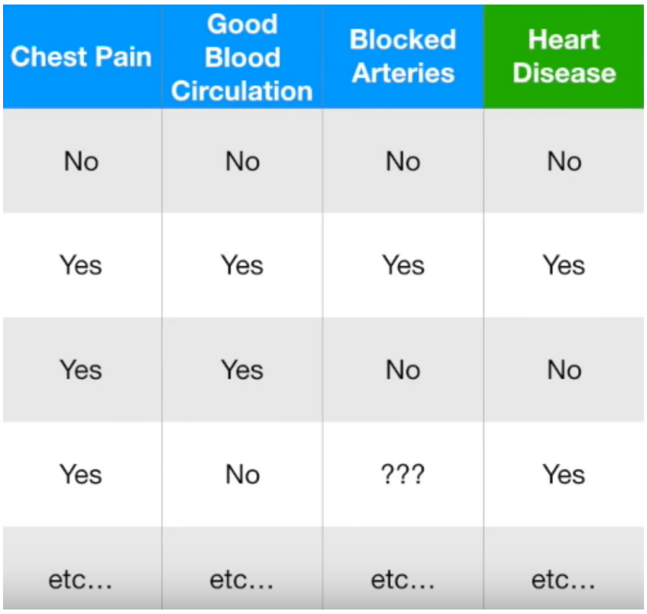
可处理的数据类型：

Numerical, Category, Multiple Choice, Rank Value

Category:

通过回答“YES/OR”的问题对样本进行分类，最终得到二分类结果

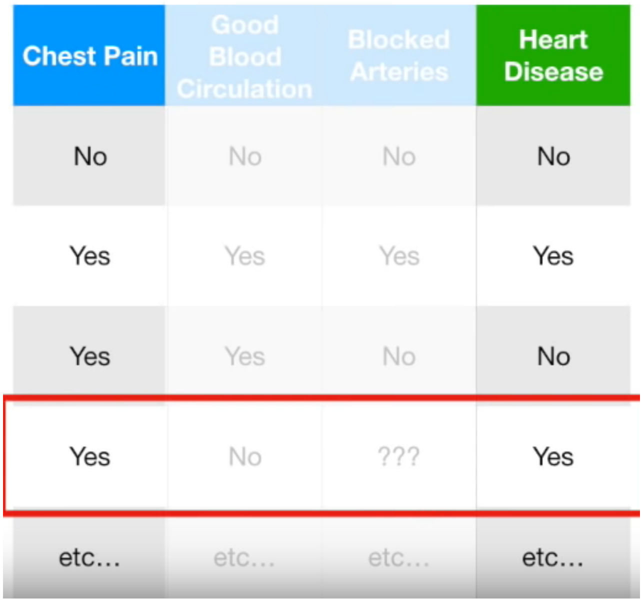


Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes
etc…	etc…	etc…	etc…

建立Decision Tree的步骤：

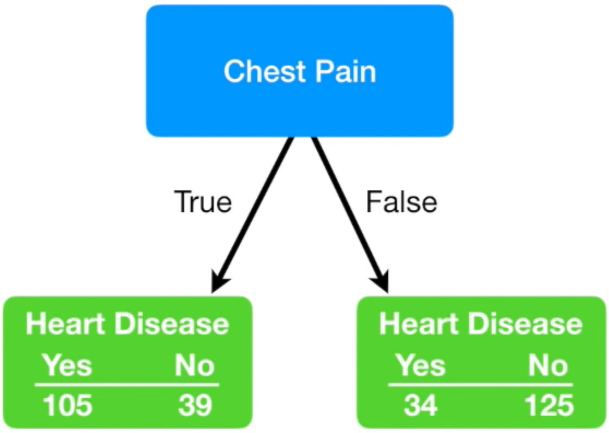
1. 统计各个特征和分类标签的关系：

对每个样本的Chest Pain属性（或其他属性）和Heart Disease的关联进行统计

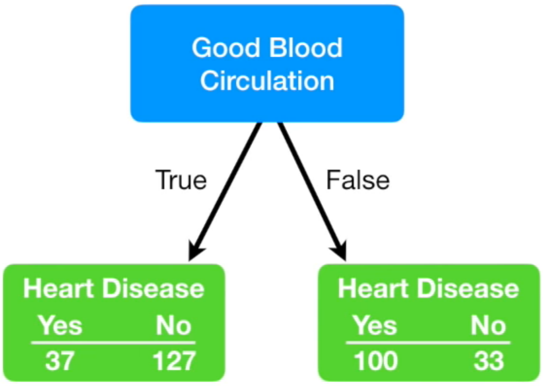


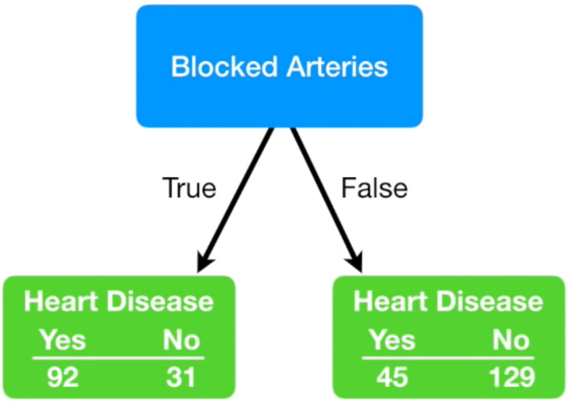
Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes
etc…	etc…	etc…	etc…

得到最终结果：



如此类推，对Good Blood Circulation, Blocked Arteries属性进行统计





*对于缺失值，跳过，不作统计

2：选择某一特征作为Root（决策标准）

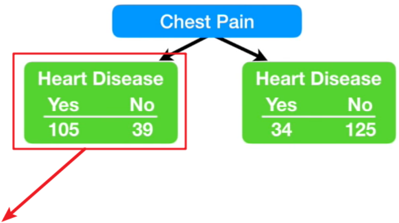
可以观察出，三个特征都没有和Heart Disease有直接关联（即True即100%为Heart Disease），则认为三个特征都是impure的。

因此，需要计算impurity。

有很多方式计算impurity，但最常用的为**Gini**。

下面给出计算Gini的例子：

（1）：计算每个leaf的Gini:

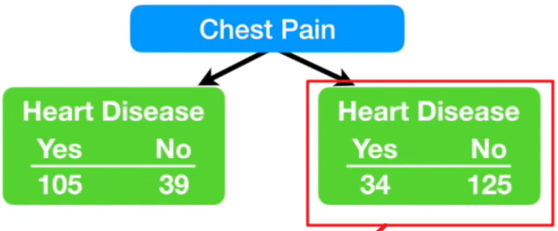


For this leaf, the Gini impurity = 1 - (the probability of “yes”)^2 - (the probability of “no”)^2

$$= 1 - \left(\frac{105}{105 + 39}\right)^2 - \left(\frac{39}{105 + 39}\right)^2$$

= 0.395

*分母是当前分类下的样本总数

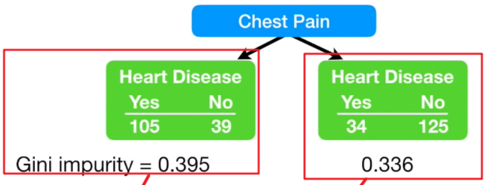


= 1 - (the probability of “yes”)^2 - (the probability of “no”)^2

$$= 1 - \left(\frac{34}{34 + 125}\right)^2 - \left(\frac{125}{34 + 125}\right)^2$$

= 0.336

（2）：以加权方式计算该特征的Gini值：



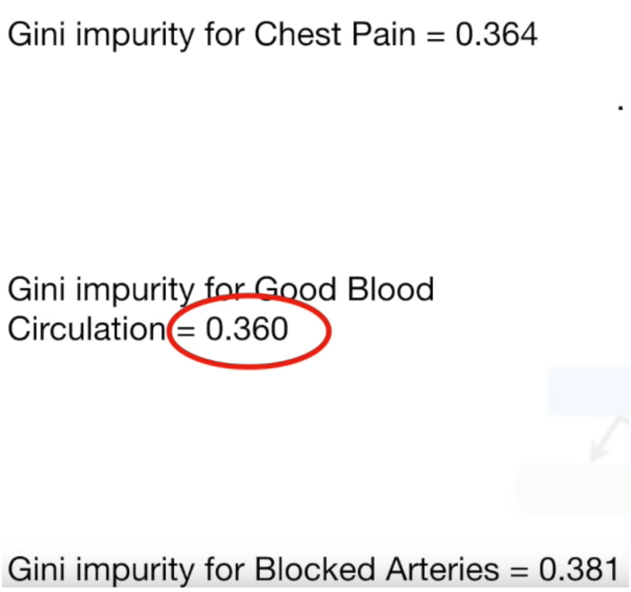
Gini impurity for Chest Pain = weighted average of Gini impurities for the leaf nodes

$$= \boxed{\left(\frac{144}{144 + 159}\right) 0.395} + \boxed{\left(\frac{159}{144 + 159}\right) 0.336}$$

= 0.364

（3）：计算所有特征的Gini值

最后，选择**Gini值最小**的特征作为决策标准



Gini impurity for Chest Pain = 0.364

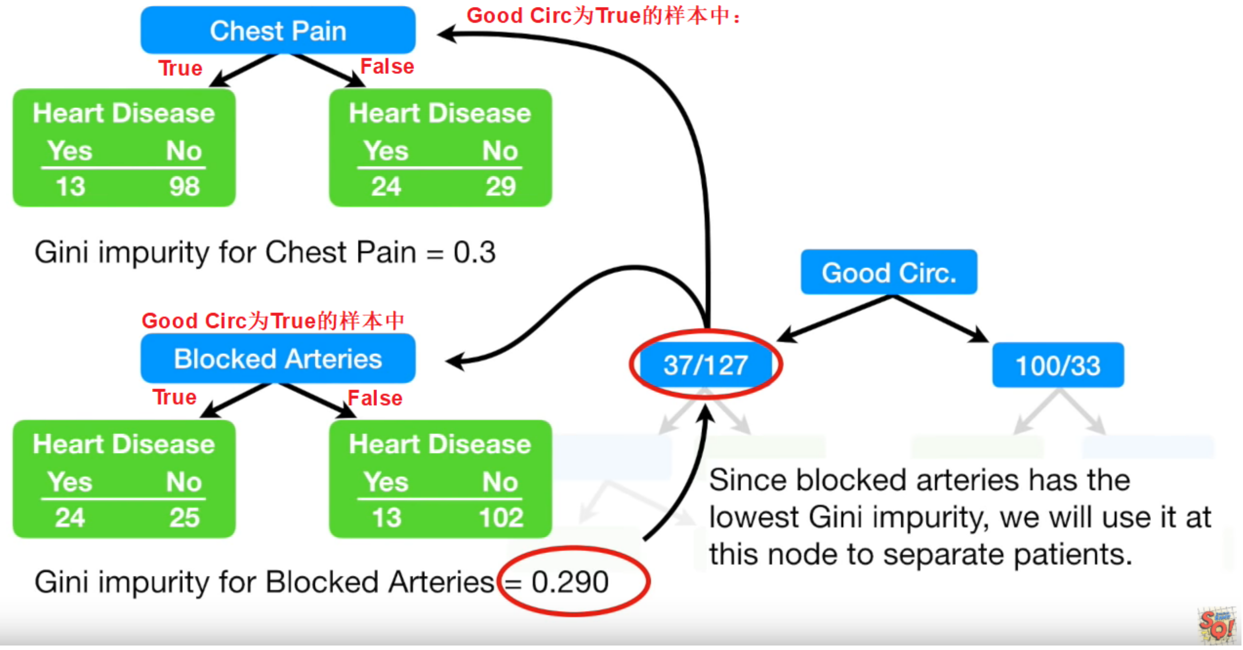
Gini impurity for Good Blood Circulation = 0.360

Gini impurity for Blocked Arteries = 0.381

3. 重复第1,2步内容

从上面可知，Good Blood Circulation作为Root Node。

再选择leaf node:



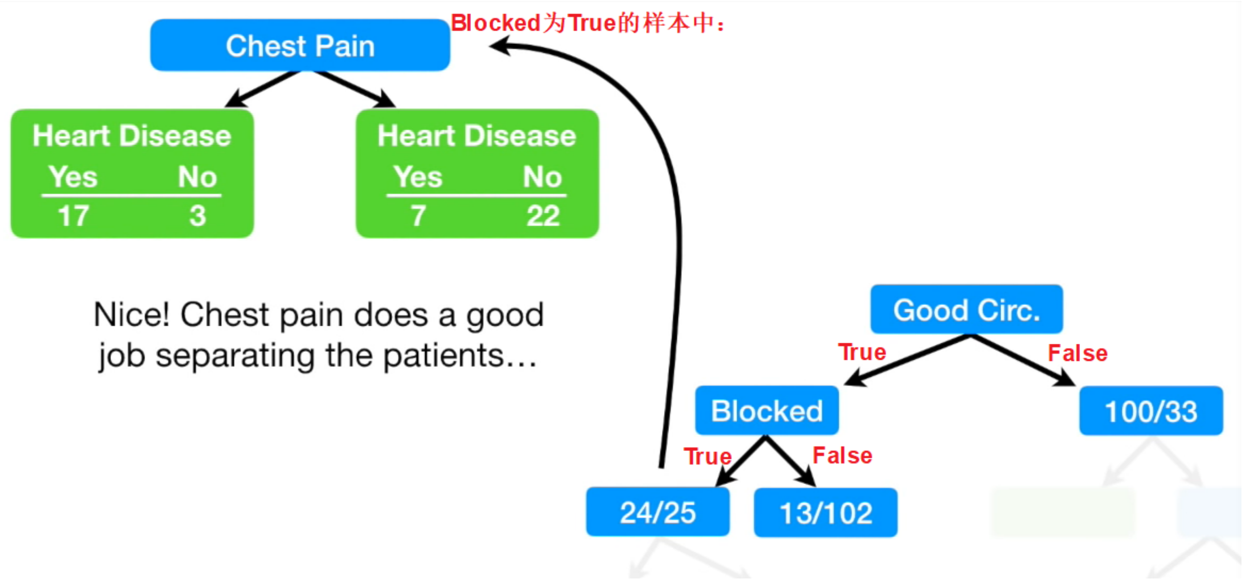
Gini impurity for Chest Pain = 0.3

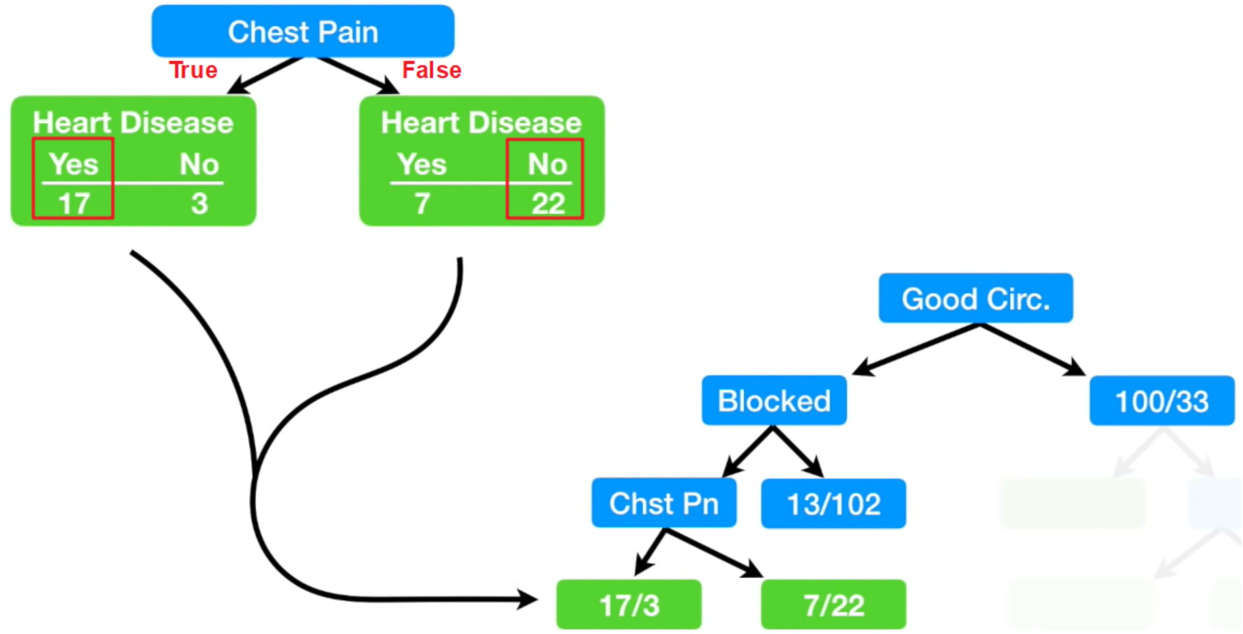
Gini impurity for Blocked Arteries = 0.290

Since blocked arteries has the lowest Gini impurity, we will use it at this node to separate patients.

Blocked Arteries的Gini值比Chest Pain少，因此选择Blocked Arteries为leaf node。

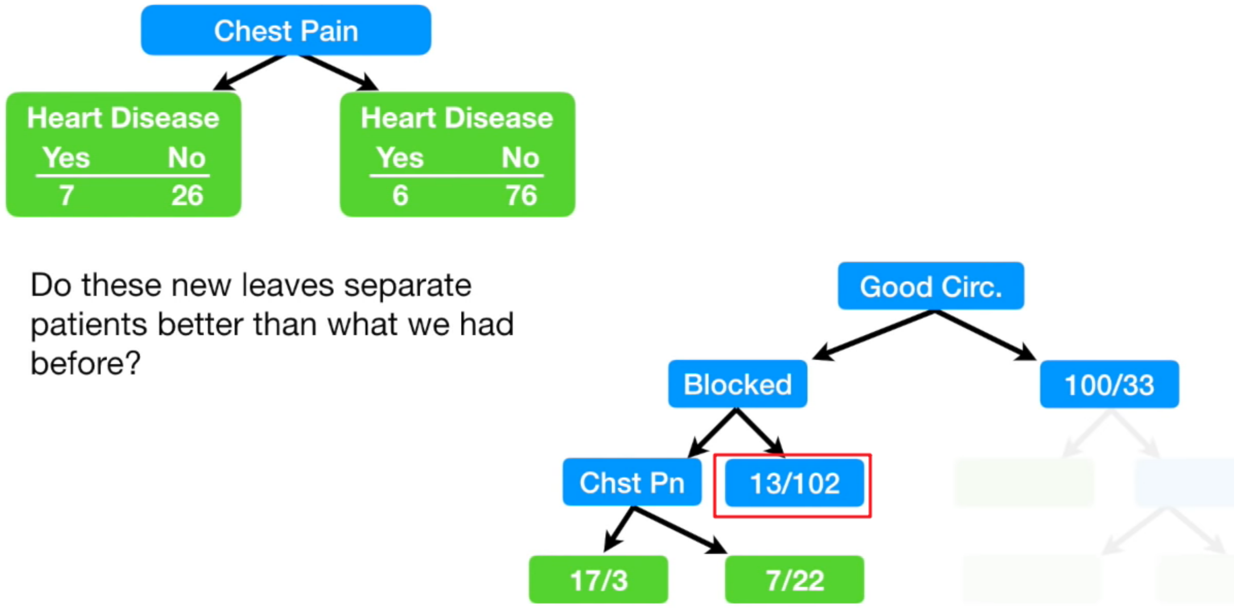
再对Blocked节点进行细分：





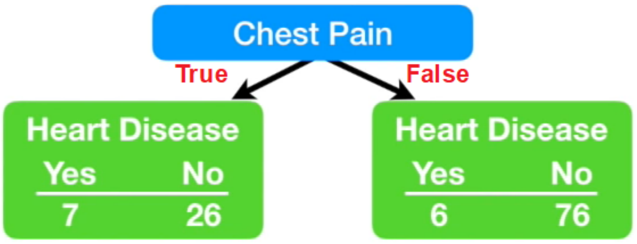
由红框中的数据可以看出，可以根据Chest Pain很好的分类出是否有Heart Disease的样本。

PS：如果对13/102的leaf node使用Chest Pain进行划分，会得到怎样的结果?



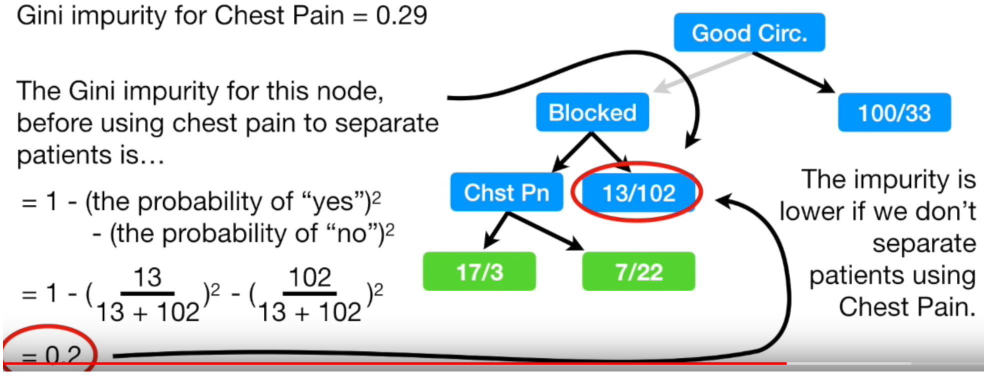
Do these new leaves separate patients better than what we had before?

对13/102使用Chest Pain进行分类后，得到的样本划分和Gini值为**0.29**：



Gini impurity for Chest Pain = 0.29

但是，如果不使用Chest Pain进行划分，直接保留13/102的数据，Gini值为**0.2**：



Gini impurity for Chest Pain = 0.29

The Gini impurity for this node, before using chest pain to separate patients is…

= 1 - (the probability of “yes”)^2
- (the probability of “no”)^2

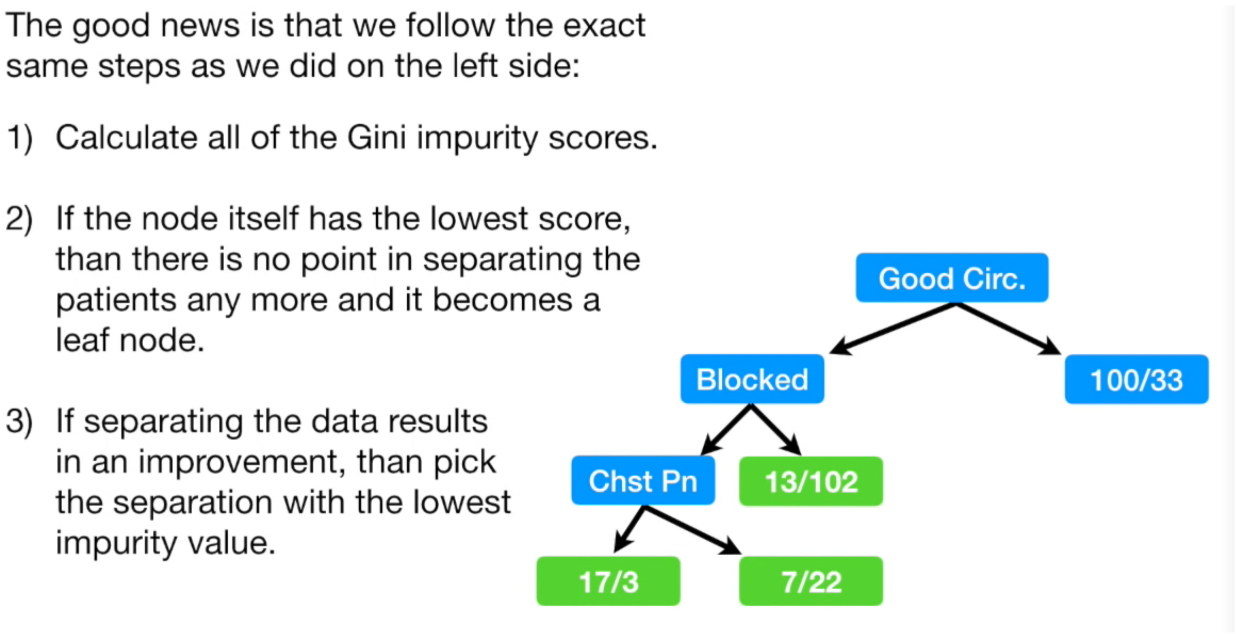
$$= 1 - \left(\frac{13}{13 + 102}\right)^2 - \left(\frac{102}{13 + 102}\right)^2$$

= 0.2

The impurity is lower if we don’t separate patients using Chest Pain.

比划分后的0.29还要低。因此，保留原分类更好。

接着对构建右子树：

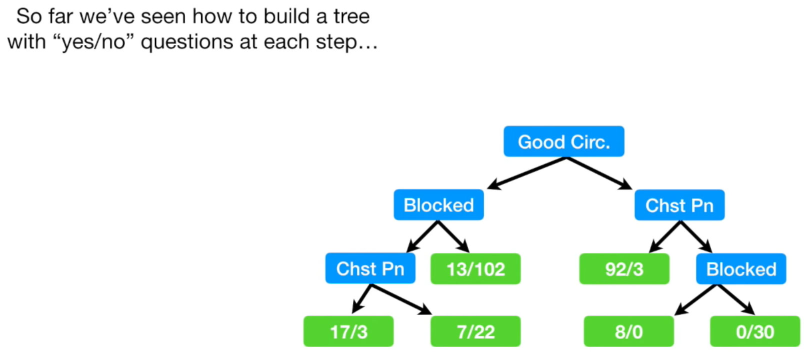


The good news is that we follow the exact same steps as we did on the left side:

1) Calculate all of the Gini impurity scores.

2) If the node itself has the lowest score, than there is no point in separating the patients any more and it becomes a leaf node.

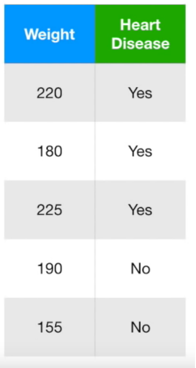
3) If separating the data results in an improvement, than pick the separation with the lowest impurity value.



So far we've seen how to build a tree with "yes/no" questions at each step…

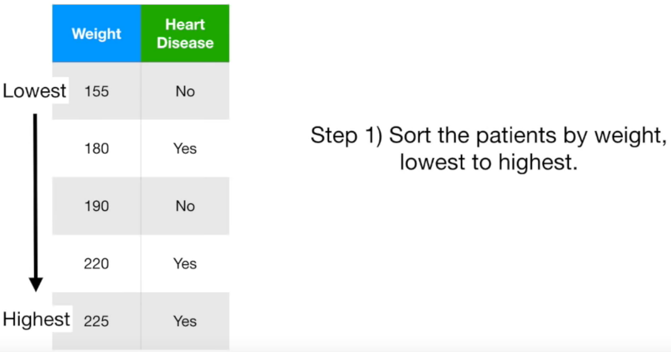
如何处理数值型特征值?

E.g:



Weight	Heart Disease
220	Yes
180	Yes
225	Yes
190	No
155	No

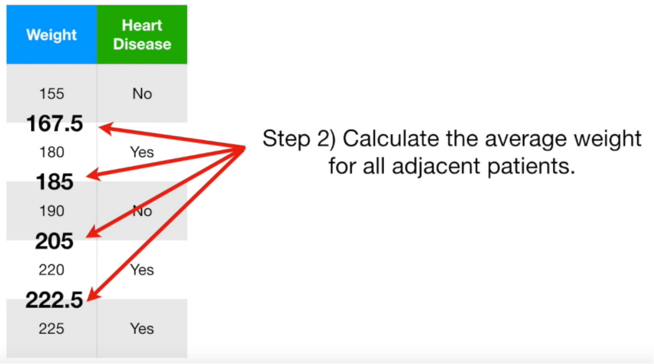
（1）将特征值由小到大排序



	Weight	Heart Disease
Lowest	155	No
	180	Yes
	190	No
	220	Yes
Highest	225	Yes

Step 1) Sort the patients by weight, lowest to highest.

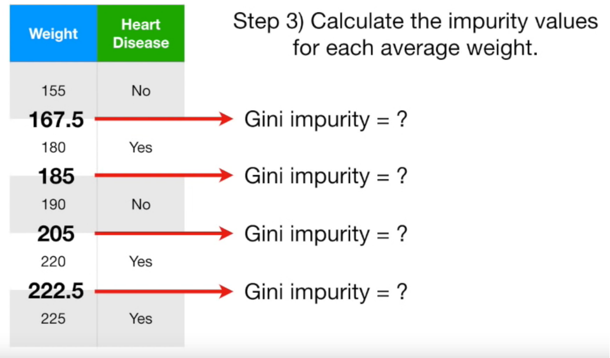
（2）计算相邻两个特征值的平均值



Weight	Heart Disease
155	No
167.5	
180	Yes
185	
190	No
205	
220	Yes
222.5	
225	Yes

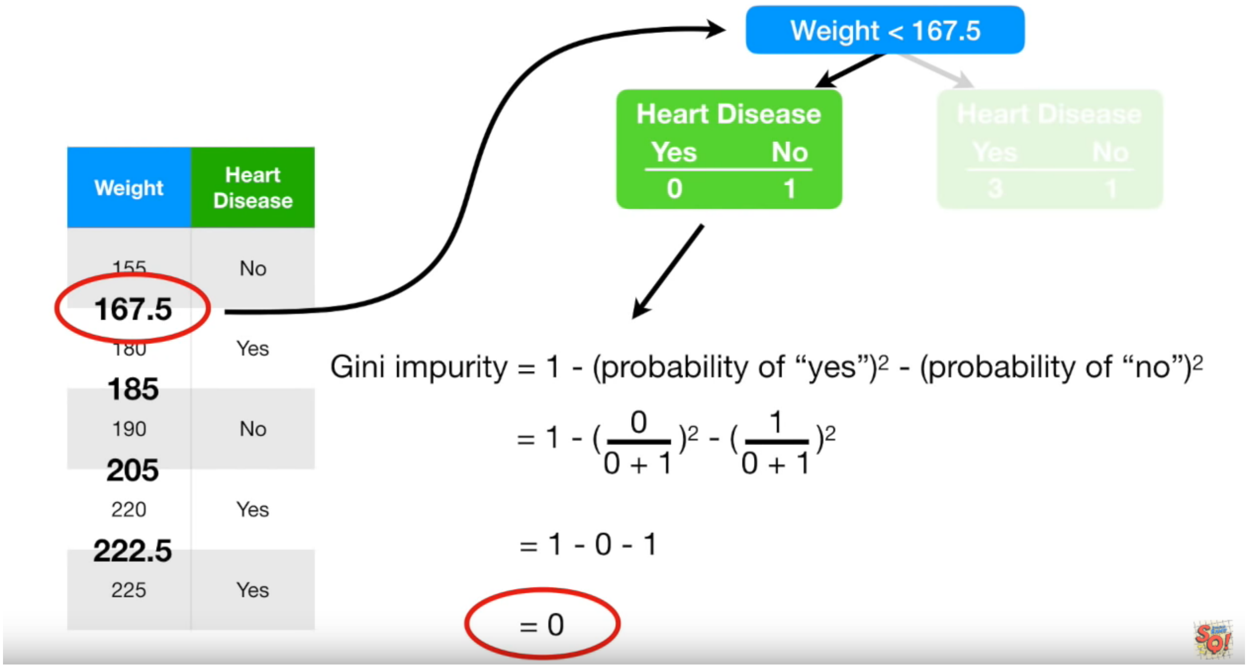
Step 2) Calculate the average weight for all adjacent patients.

（3）计算每个平均值的Gini值



Step 3) Calculate the impurity values for each average weight.

Weight	Heart Disease	
155	No	
167.5		Gini impurity = ?
180	Yes	
185		Gini impurity = ?
190	No	
205		Gini impurity = ?
220	Yes	
222.5		Gini impurity = ?
225	Yes	



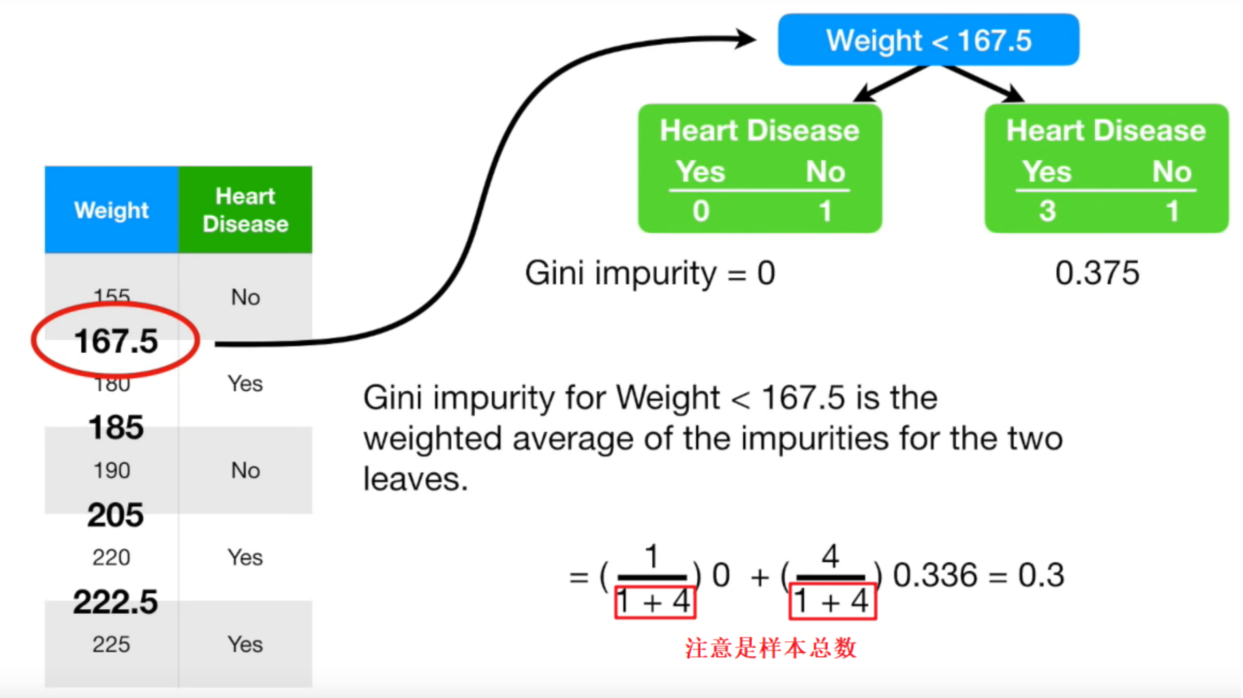
Weight	Heart Disease
155	No
167.5	
180	Yes
185	
190	No
205	
220	Yes
222.5	
225	Yes

Gini impurity = 1 - (probability of “yes”)^2 - (probability of “no”)^2

$$= 1 - \left(\frac{0}{0+1}\right)^2 - \left(\frac{1}{0+1}\right)^2$$

$$= 1 - 0 - 1$$

$$= 0$$



Weight	Heart Disease
155	No
167.5	
180	Yes
185	
190	No
205	
220	Yes
222.5	
225	Yes

Weight < 167.5

Heart Disease	
Yes	No
0	1

Heart Disease	
Yes	No
3	1

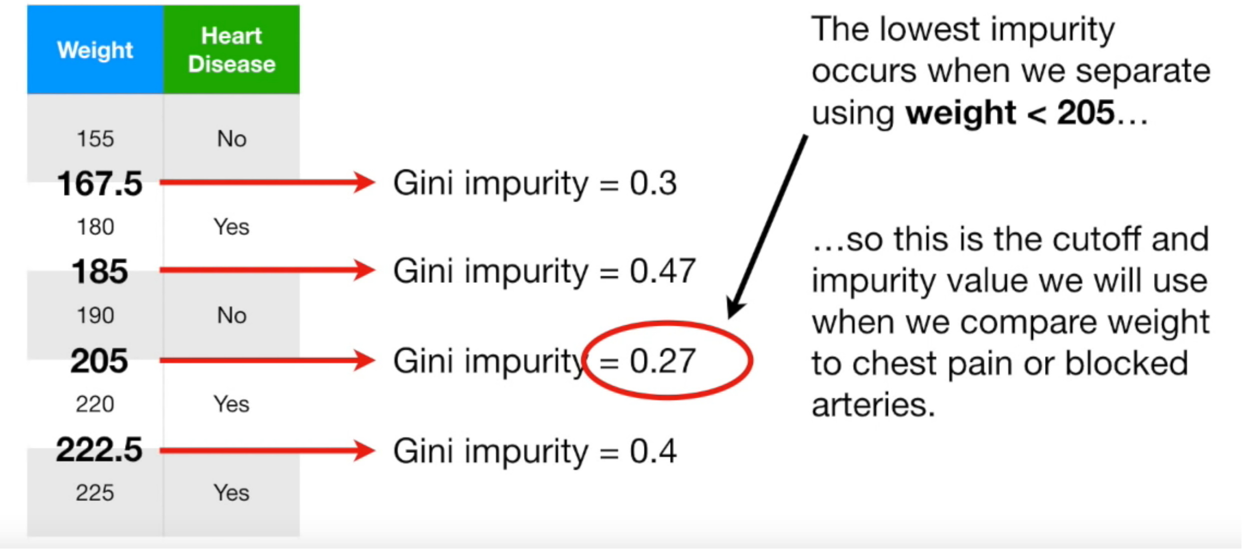
Gini impurity = 0

0.375

Gini impurity for Weight < 167.5 is the weighted average of the impurities for the two leaves.

$$= \left(\frac{1}{\boxed{1+4}}\right) 0 + \left(\frac{4}{\boxed{1+4}}\right) 0.336 = 0.3$$

注意是样本总数



Weight	Heart Disease
155	No
167.5	
180	Yes
185	
190	No
205	
220	Yes
222.5	
225	Yes

Gini impurity = 0.3

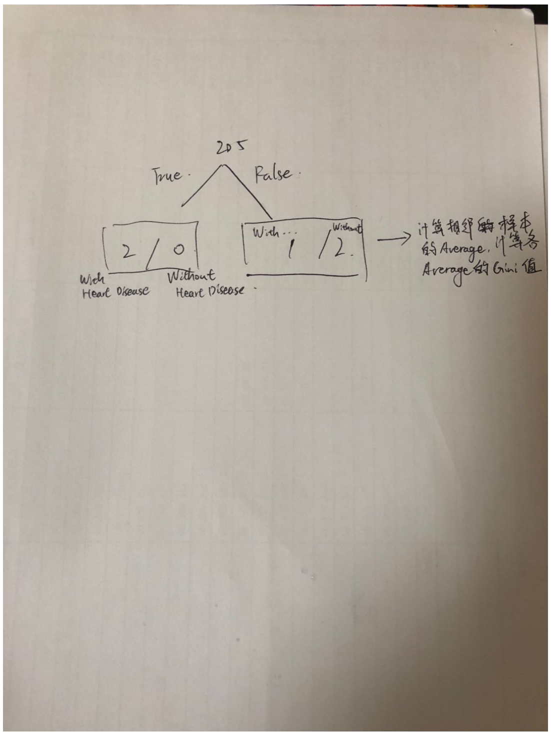
Gini impurity = 0.47

Gini impurity = 0.27

Gini impurity = 0.4

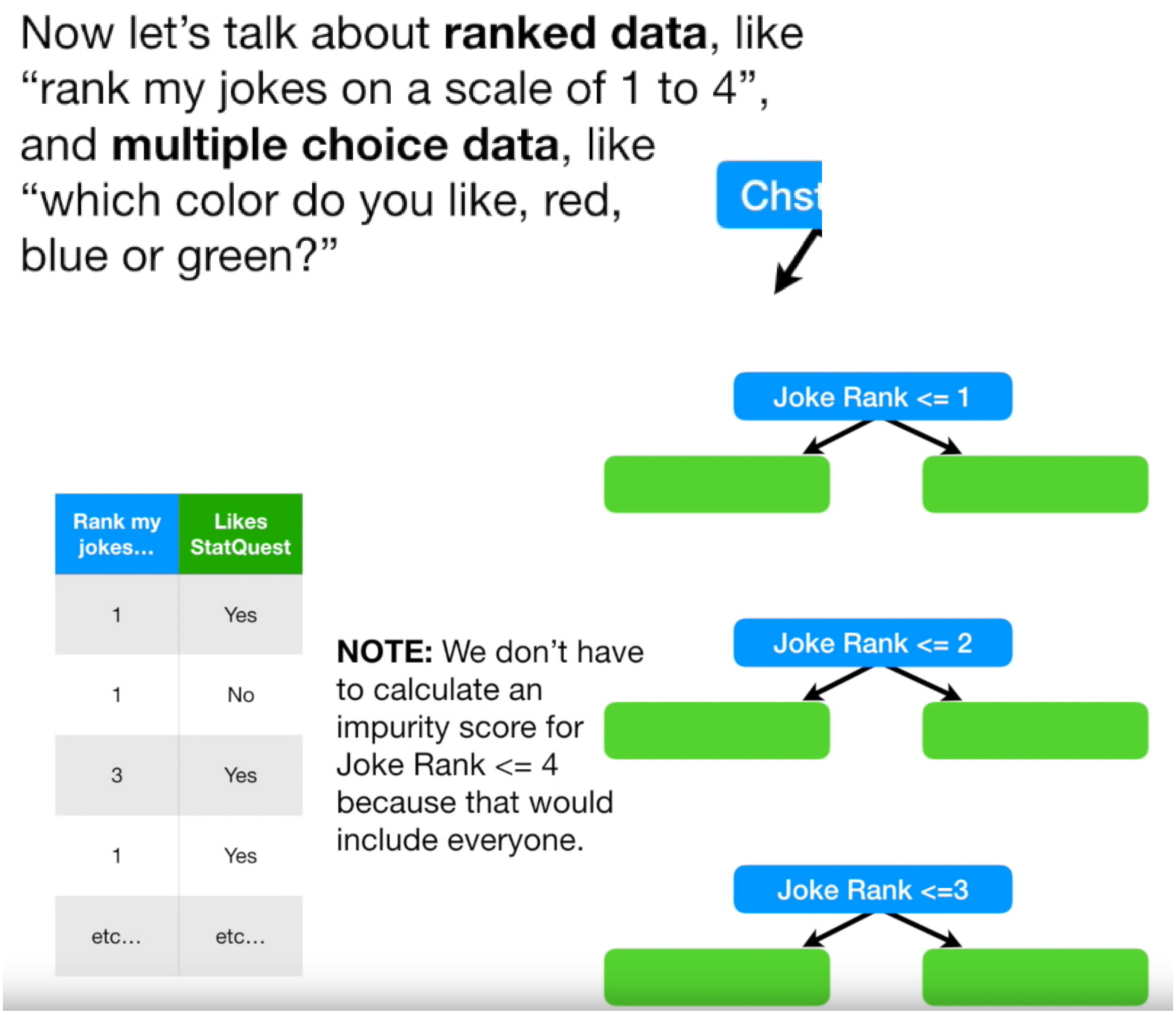
The lowest impurity occurs when we separate using **weight < 205**…

…so this is the cutoff and impurity value we will use when we compare weight to chest pain or blocked arteries.



计算相邻的样本的Average，计算各Average的Gini值

如何处理Rank data或multiple choice data?

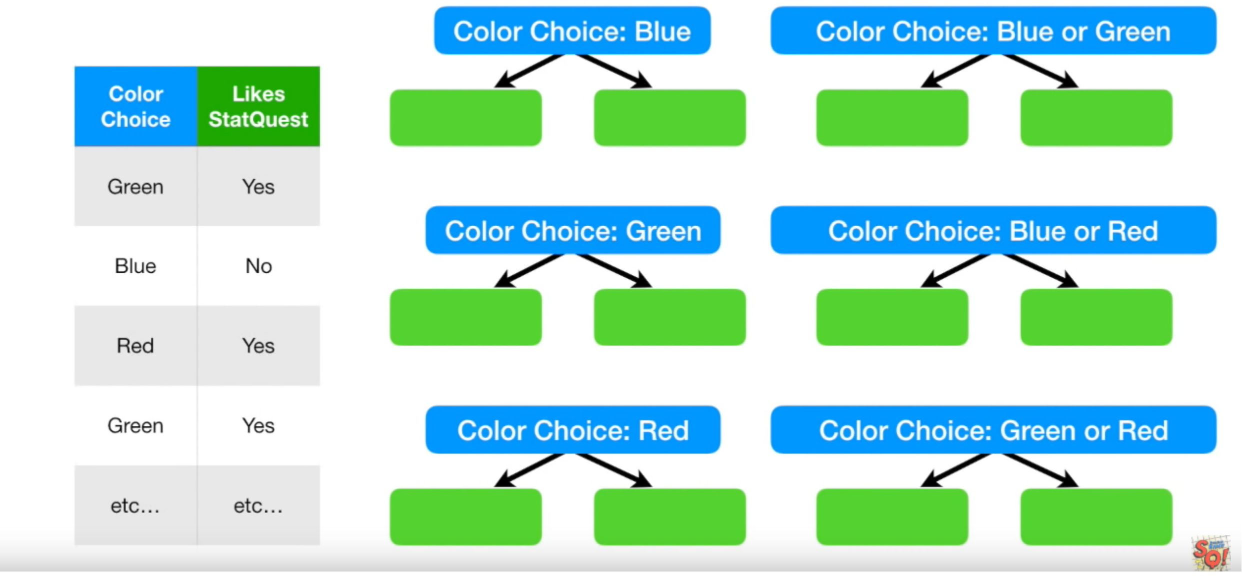


Now let's talk about **ranked data**, like "rank my jokes on a scale of 1 to 4", and **multiple choice data**, like "which color do you like, red, blue or green?"

Rank my jokes...	Likes StatQuest
1	Yes
1	No
3	Yes
1	Yes
etc...	etc...

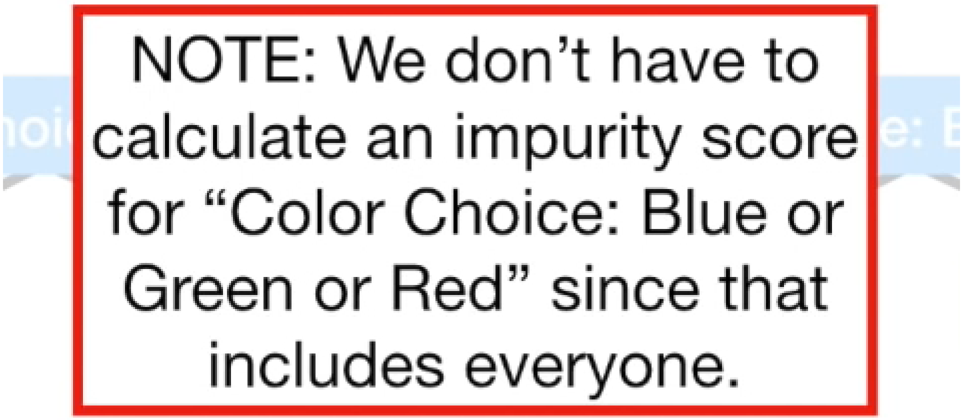
NOTE: We don't have to calculate an impurity score for Joke Rank <= 4 because that would include everyone.

1代表最无趣，4代表最有趣



Color Choice	Likes StatQuest
Green	Yes
Blue	No
Red	Yes
Green	Yes
etc…	etc…

Color Choice: Blue
Color Choice: Blue or Green
Color Choice: Green
Color Choice: Blue or Red
Color Choice: Red
Color Choice: Green or Red

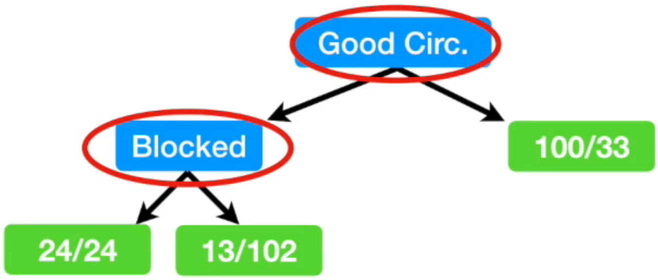


NOTE: We don’t have to calculate an impurity score for “Color Choice: Blue or Green or Red” since that includes everyone.

Feature Selection and Missing Data

Feature Selection:

决策树有时可以起到feature selection的作用：



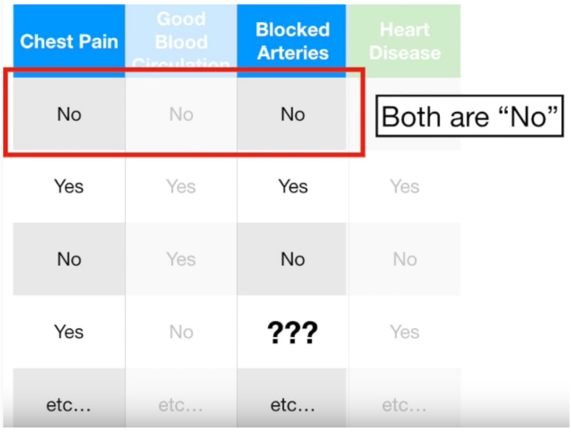
某些情况下（比如使用Chest Pain再进行细分时，Gini值没有减少），当前左子树只有Blocked这一feature，相当于实现了feature selection。

可以设置一个关于Gini值的threshold，当Gini值达到该threshold时才构建子树，这样可以解决over fit的问题。

Handling Missing Data:

对于YES/NO的问题，可以根据其他最相关的feature作出假设，或根据label作出假设。

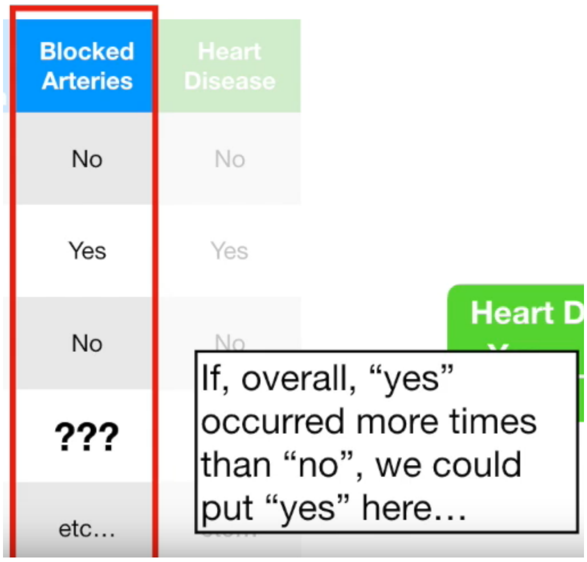
E.g:



Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	
Yes	Yes	Yes	Yes
No	Yes	No	No
Yes	No	???	Yes
etc…	etc…	etc…	etc…

Both are "No"

Chest Pain和Blocked Arteries几乎相同



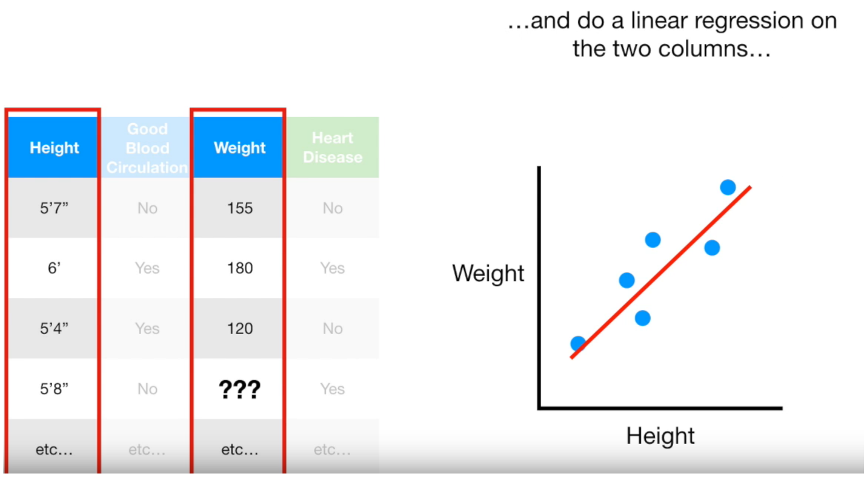
Blocked Arteries	Heart Disease
No	No
Yes	Yes
No	No
???	
etc…	etc…

If, overall, "yes" occurred more times than "no", we could put "yes" here…

Blocked Arteries和Heart Disease几乎相同

对于数值型特征值，可以选择median value，average value，或找出最相关的feature，使用linear regression作出预测。

E.g:



…and do a linear regression on the two columns…

Height	Good Blood Circulation	Weight	Heart Disease
5’7”	No	155	No
6’	Yes	180	Yes
5’4”	Yes	120	No
5’8”	No	???	Yes
etc…	etc…	etc…	etc…

Height和Weight关联性最强，根据Height对Weight作出预测。