LECTURE 7

Decision tree classifier

Build a Decision Tree to find S,M, T

- Consider a data as shown.

Person	Gender	Height	Class
1	F	1.6	S
2	M	2.0	M
3	F	1.9	M
4	F	1.88	M
5	F	1.7	S
6	M	1.85	M
7	F	1.6	S
8	M	1.7	S
9	M	2.2	T
10	M	2.1	T
11	F	1.8	M
12	M	1.95	M
13	F	1.9	M
14	F	1.8	M
15	F	1.75	S

Attributes:

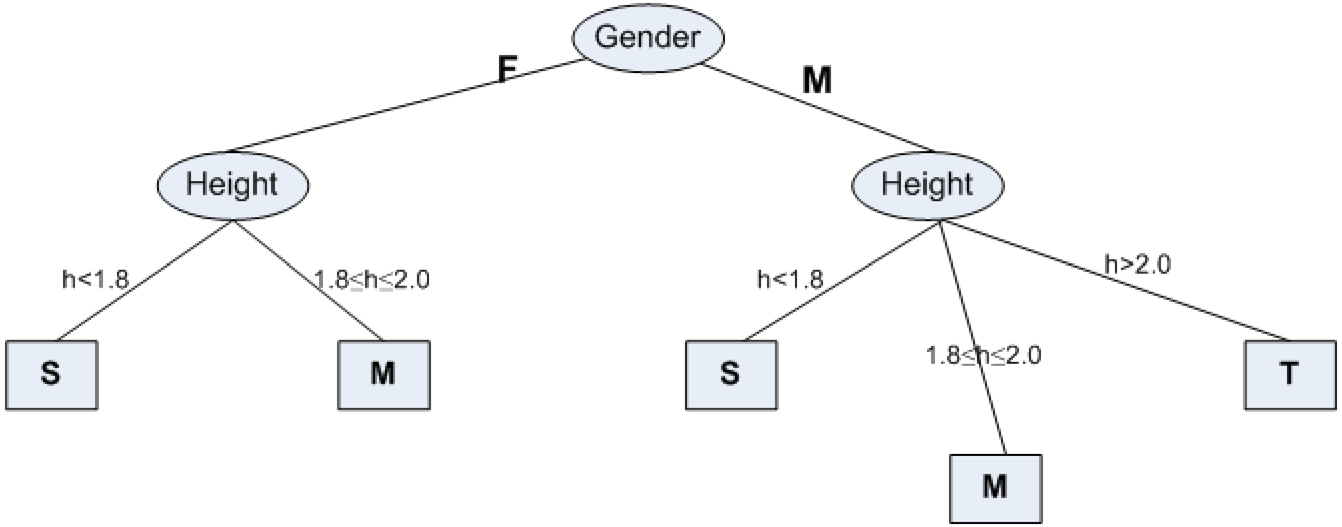
Gender = {Male(M), Female (F)} // Binary attribute
Height = {1.5, …, 2.5} // Continuous attribute

Class = {Short (S), Medium (M), Tall (T)}

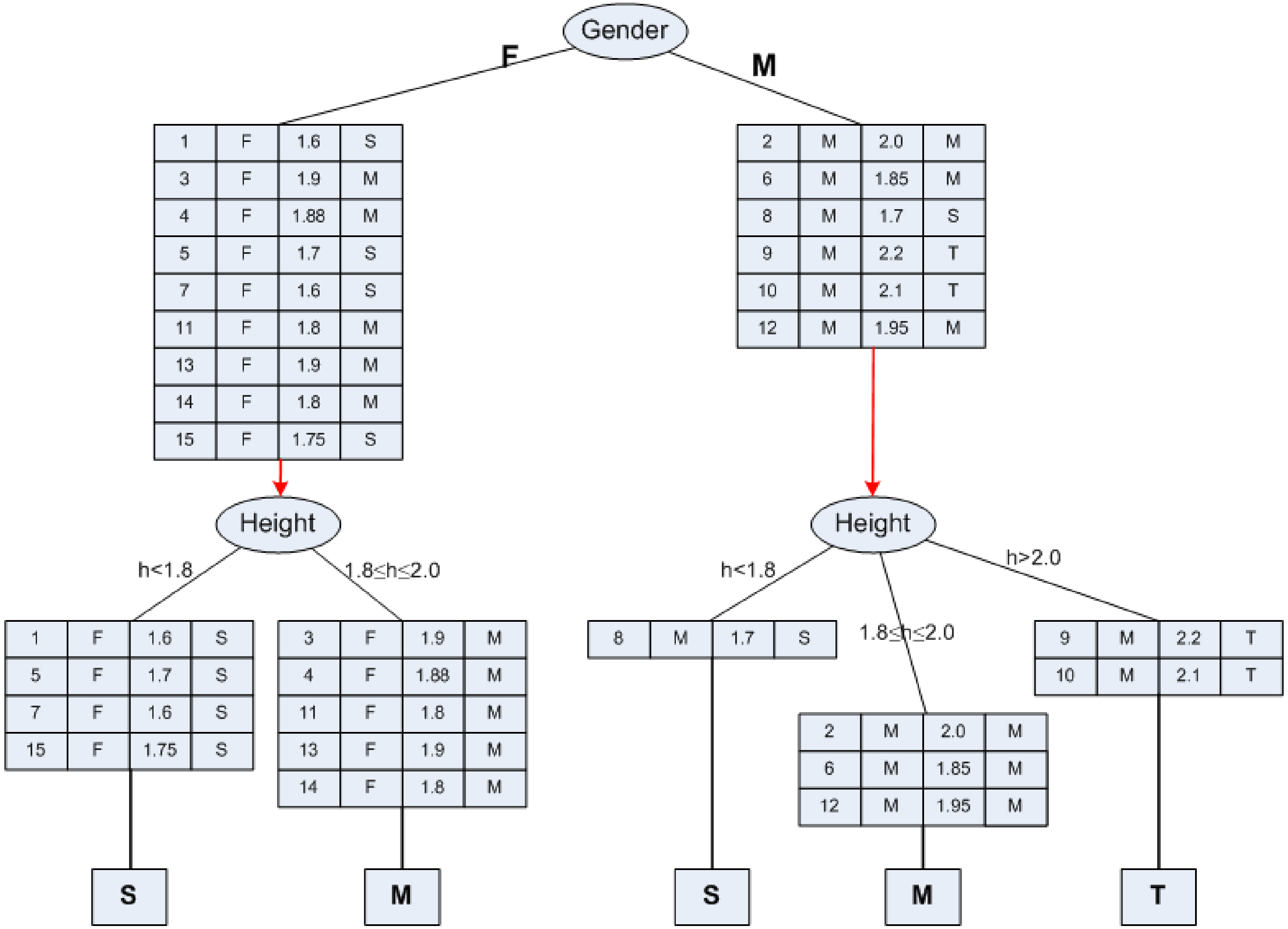
Given a person, we are to test in which class s/he belongs

Build a Decision Tree to find S,M, T

- To built a decision tree, we can select an attribute in two different orderings: <Gender, Height> or <Height, Gender>
- Further, for each ordering, we can choose different ways of splitting
- Different instances are shown in the following.
- **Approach 1 : <Gender, Height>**



Build a Decision Tree to find S,M, T



1	F	1.6	S
3	F	1.9	M
4	F	1.88	M
5	F	1.7	S
7	F	1.6	S
11	F	1.8	M
13	F	1.9	M
14	F	1.8	M
15	F	1.75	S

2	M	2.0	M
6	M	1.85	M
8	M	1.7	S
9	M	2.2	T
10	M	2.1	T
12	M	1.95	M

1	F	1.6	S
5	F	1.7	S
7	F	1.6	S
15	F	1.75	S

3	F	1.9	M
4	F	1.88	M
11	F	1.8	M
13	F	1.9	M
14	F	1.8	M

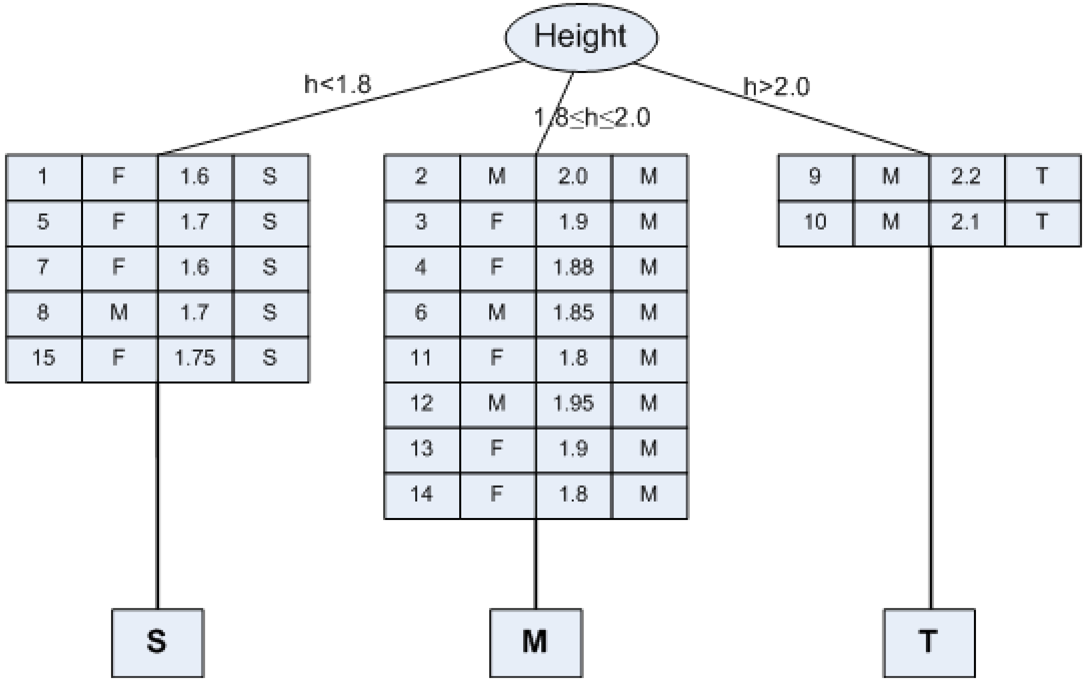
8	M	1.7	S

2	M	2.0	M
6	M	1.85	M
12	M	1.95	M

9	M	2.2	T
10	M	2.1	T

Build a Decision Tree to find S,M, T

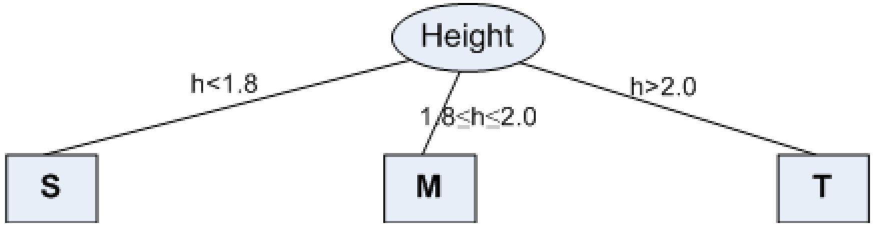
- **Approach 2 : <Height, Gender>**



1	F	1.6	S
5	F	1.7	S
7	F	1.6	S
8	M	1.7	S
15	F	1.75	S

2	M	2.0	M
3	F	1.9	M
4	F	1.88	M
6	M	1.85	M
11	F	1.8	M
12	M	1.95	M
13	F	1.9	M
14	F	1.8	M

9	M	2.2	T
10	M	2.1	T

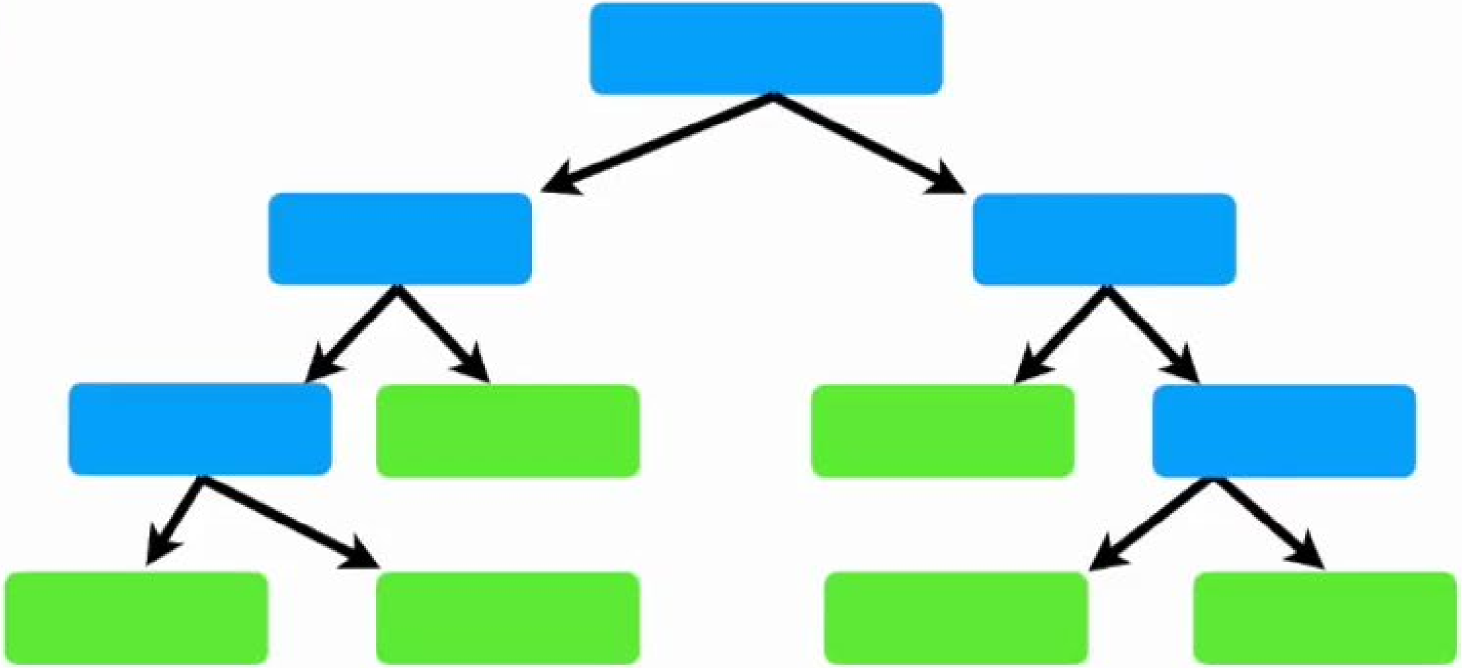


Questions??

- Which attribute choose first?
- What should be the order?
- How to decide splitting criteria?
- How to decide number of child node?
- How to measure decision tree quality?

In this example, we want to create a tree that uses **chest pain**, **good blood circulation** and **blocked artery status** to predict…

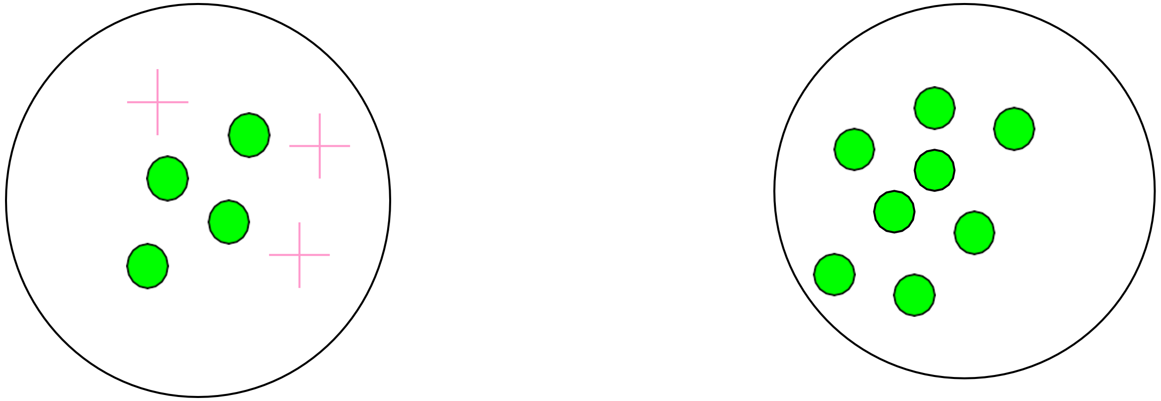
Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes
etc…	etc…	etc…	etc…



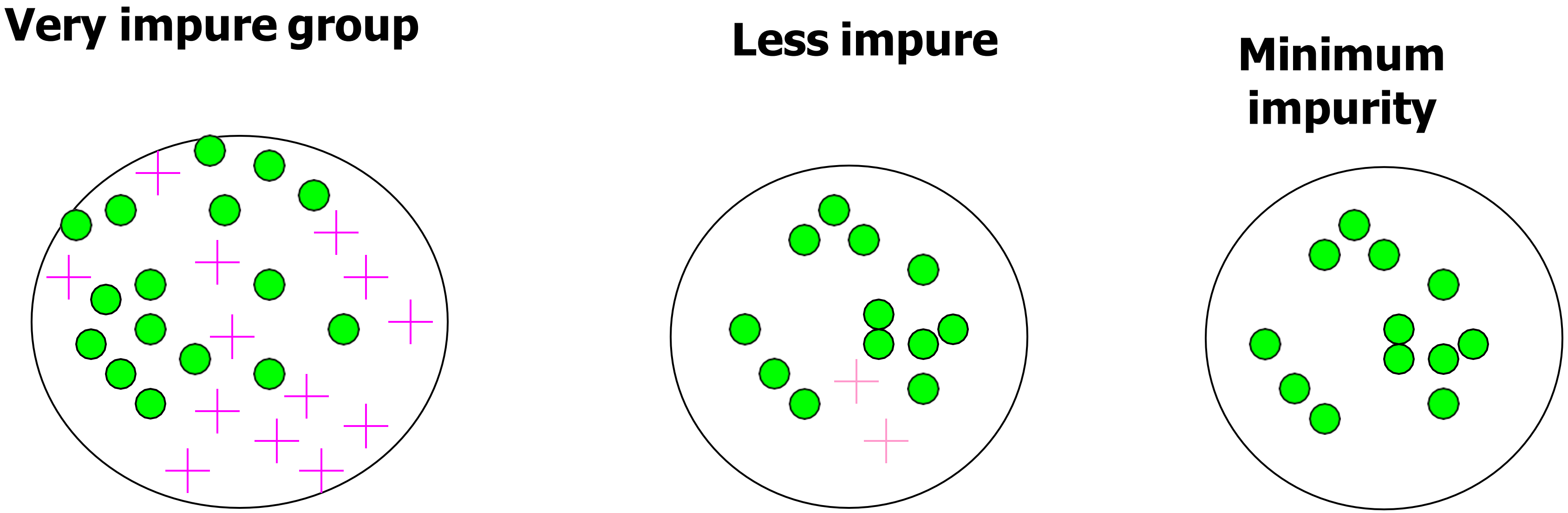
Information Gain

Impurity (informal)

- Measures the level of **impurity** in a group of examples



Impurity



Very impure group

Less impure

Minimum impurity

Information Gain

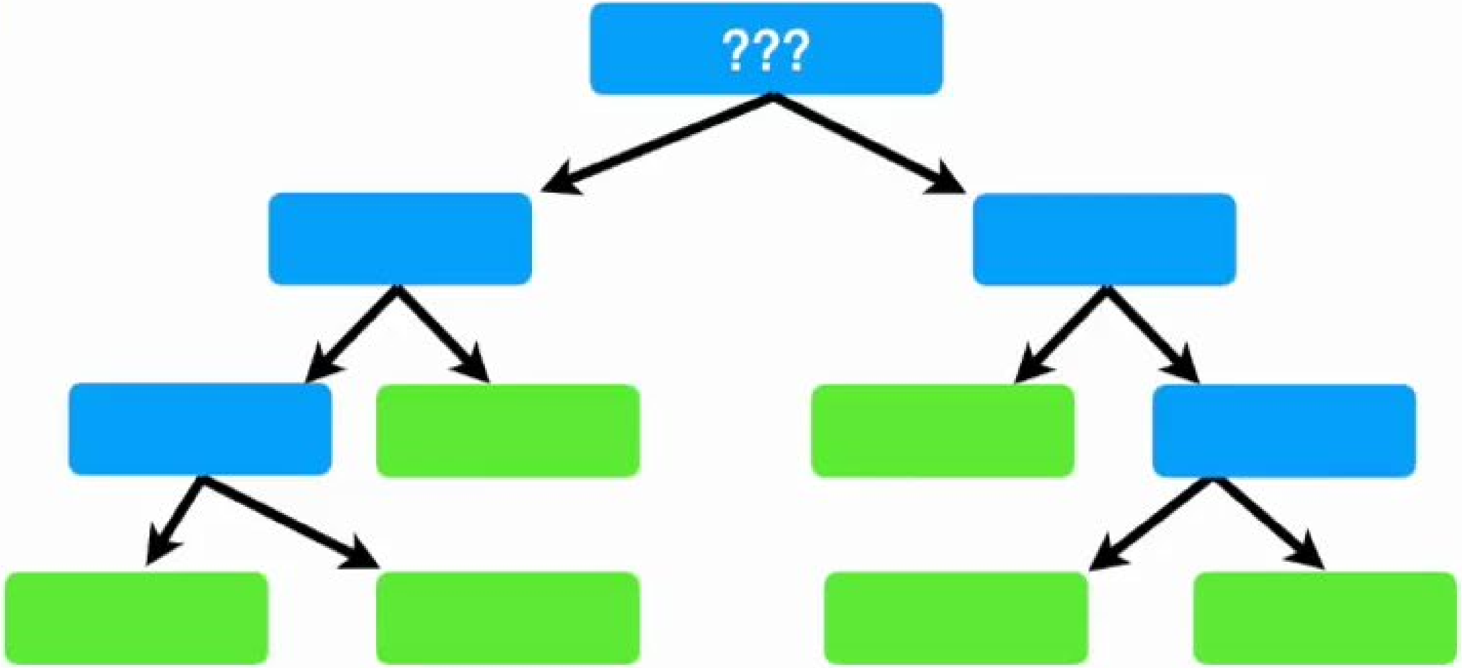
- We want to determine which attribute in a given set of training feature vectors is most useful for discriminating between the classes to be learned.

- Information gain tells us how important a given attribute of the feature vectors is.

- We will use it to decide the ordering of attributes in the nodes of a decision tree.

The first thing we want to know is whether **Chest Pain**, **Good Blood Circulation** or **Blocked Arteries** should be at the very top of our tree.

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…

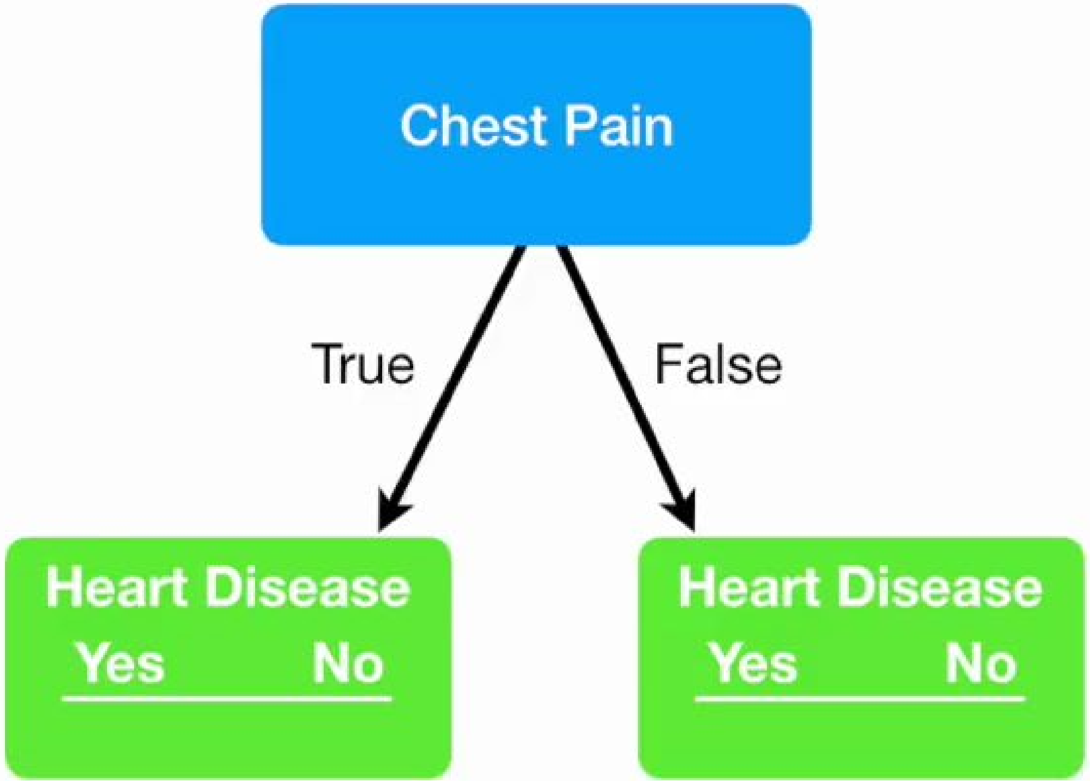


We start by looking at how well **Chest Pain** alone predicts heart disease…

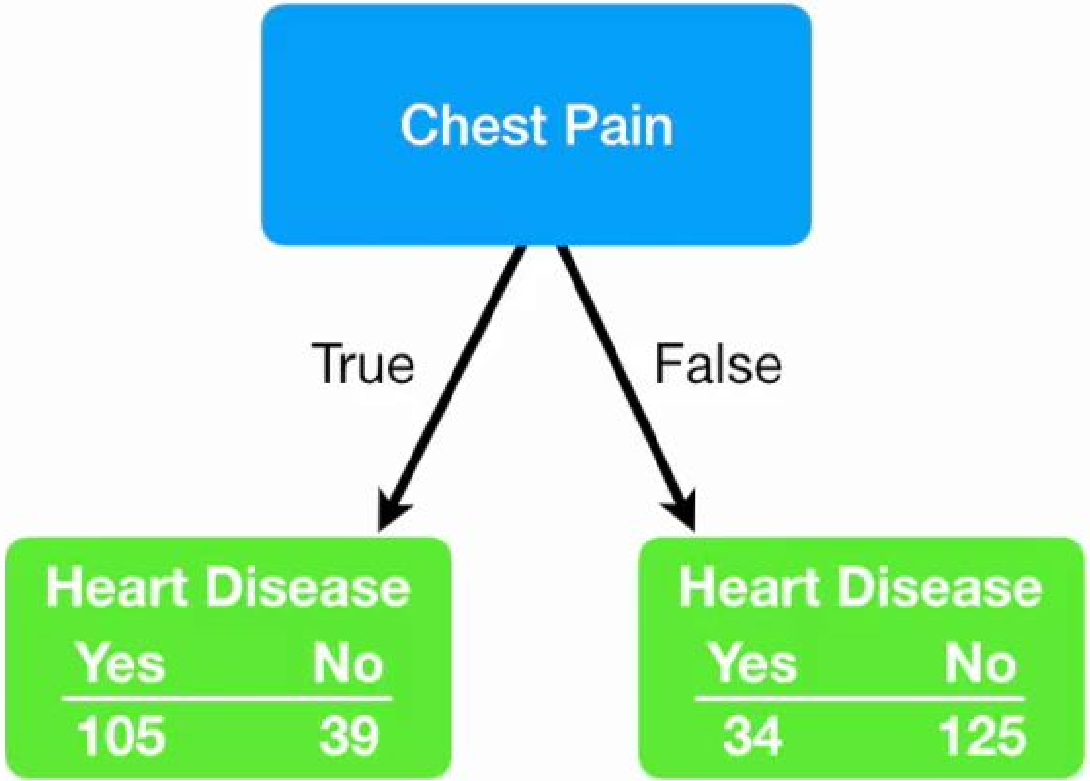
Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes
etc…	etc…	etc…	etc…

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…

Here's a little tree that only takes chest pain into account.



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…



Chest Pain

True

False

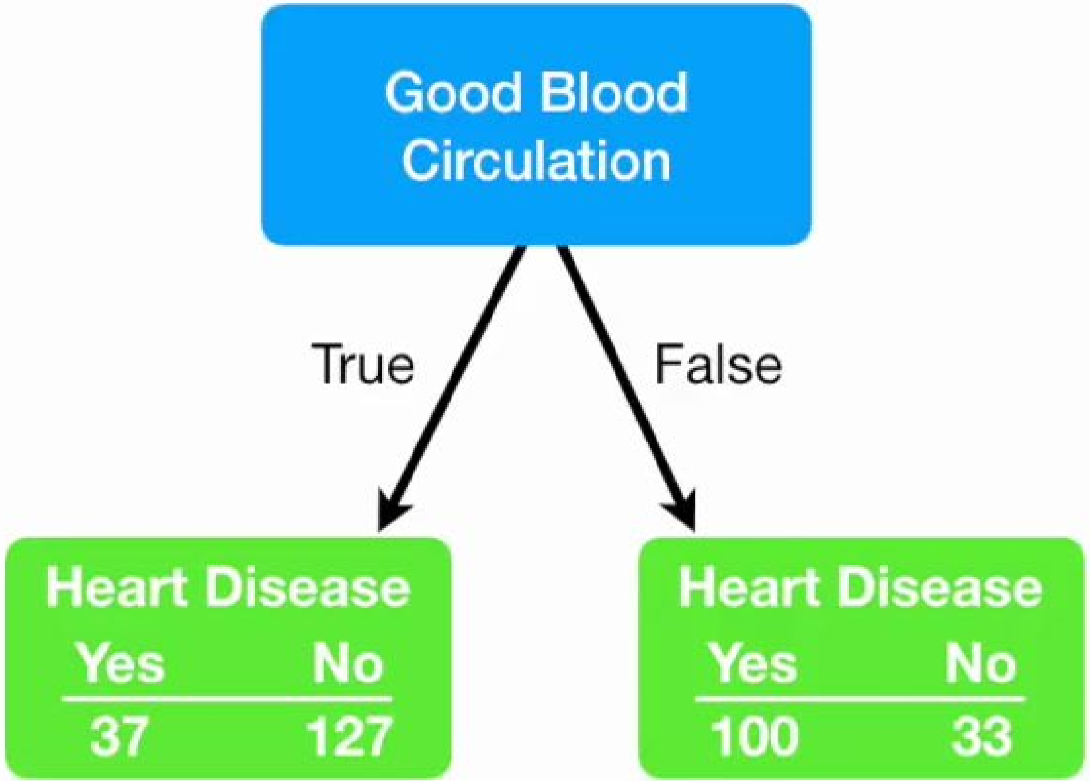
Heart Disease	
Yes	No
105	39

Heart Disease	
Yes	No
34	125

Ultimately, we look at chest pain and heart disease for all 303 patients in this study.

Now we do the exact same thing for **Good Blood Circulation**.

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…



Good Blood Circulation

True

False

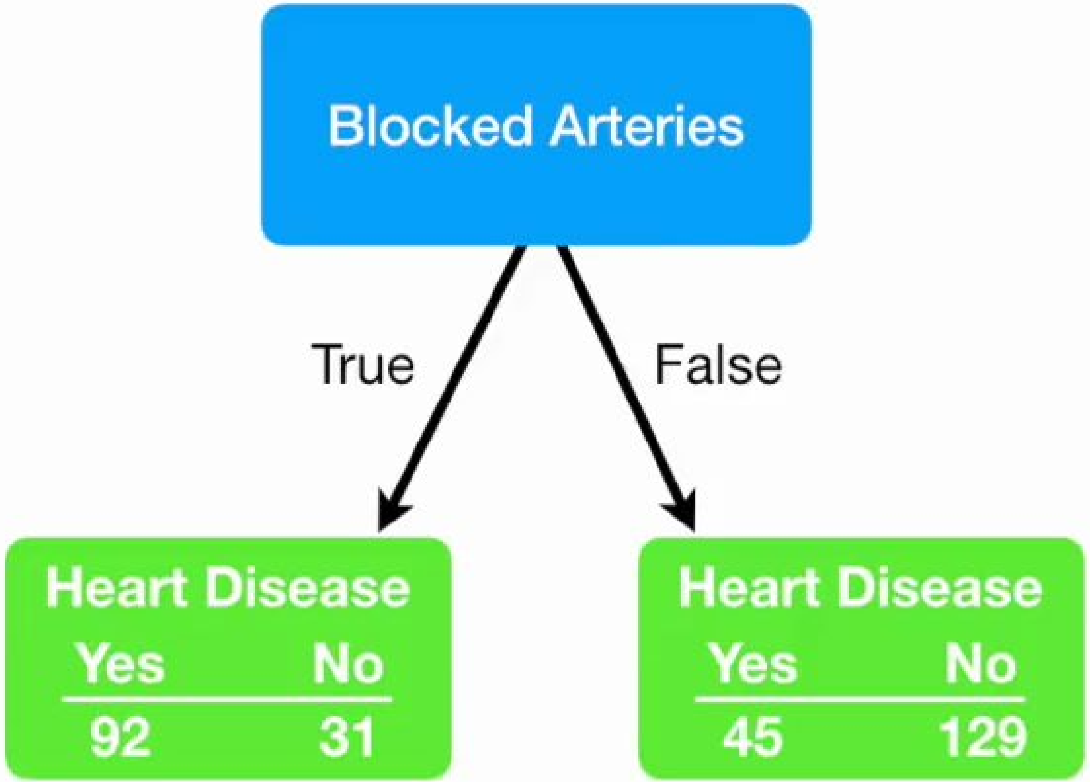
Heart Disease

Yes	No
37	127

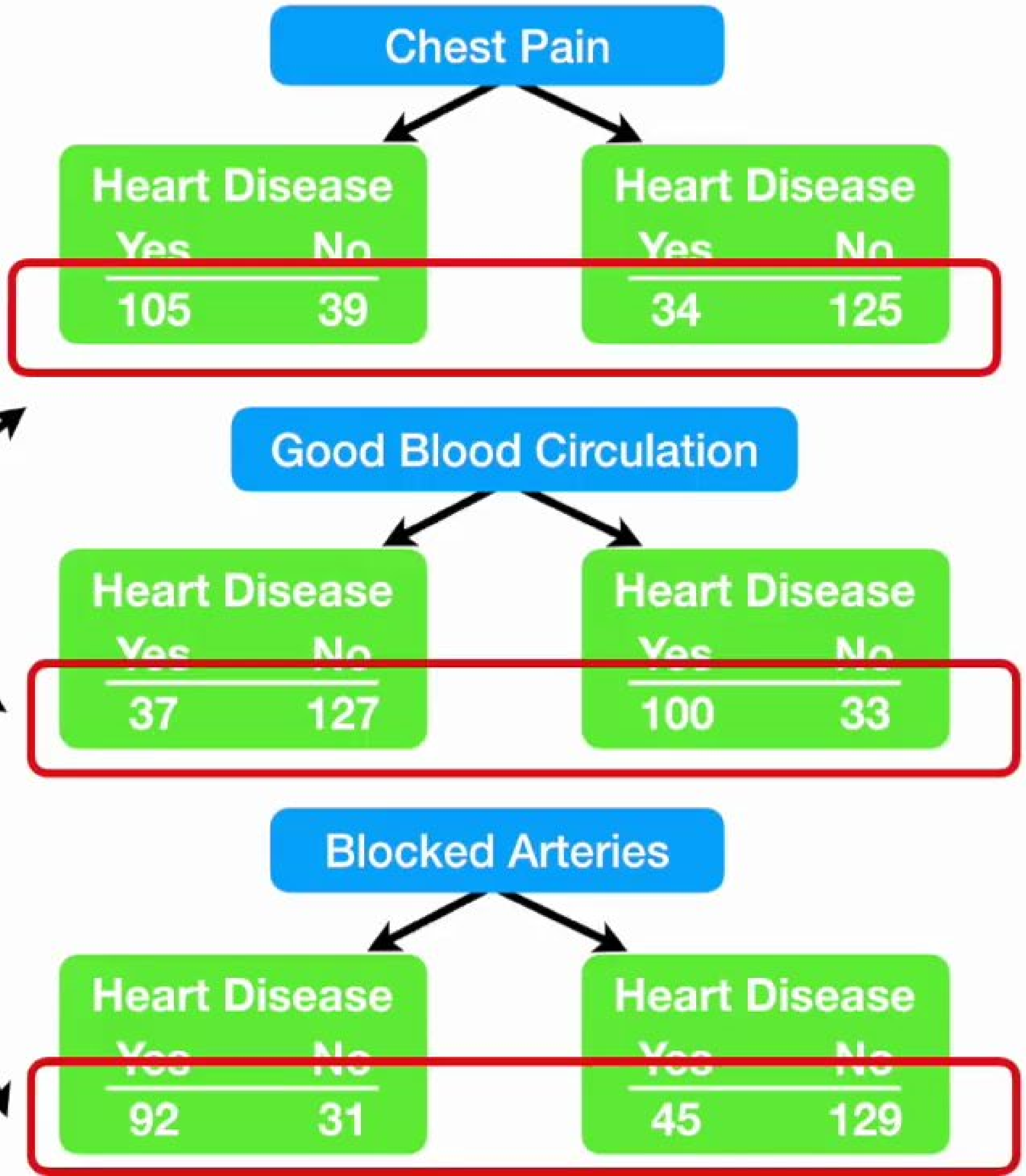
Heart Disease

Yes	No
100	33

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…



NOTE: The total number of patients with heart disease is different for Chest Pain, Good Blood Circulation and Blocked Arteries because some patients had measurements for Chest Pain, but not for Blocked Arteries, etc.



Chest Pain

Heart Disease
Yes No
105 39

Heart Disease
Yes No
34 125

Good Blood Circulation

Heart Disease
Yes No
37 127

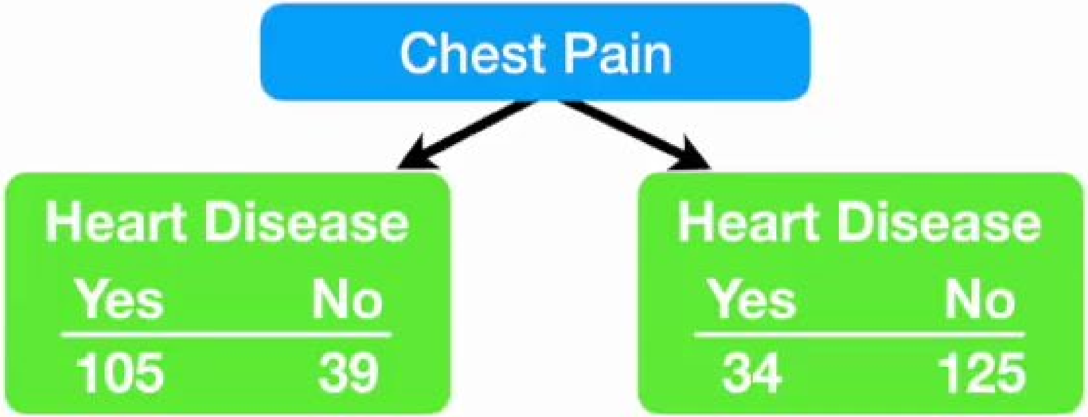
Heart Disease
Yes No
100 33

Blocked Arteries

Heart Disease
Yes No
92 31

Heart Disease
Yes No
45 129

There are a bunch of ways to measure impurity, but I'm just going to focus on a very popular one called "**Gini**".



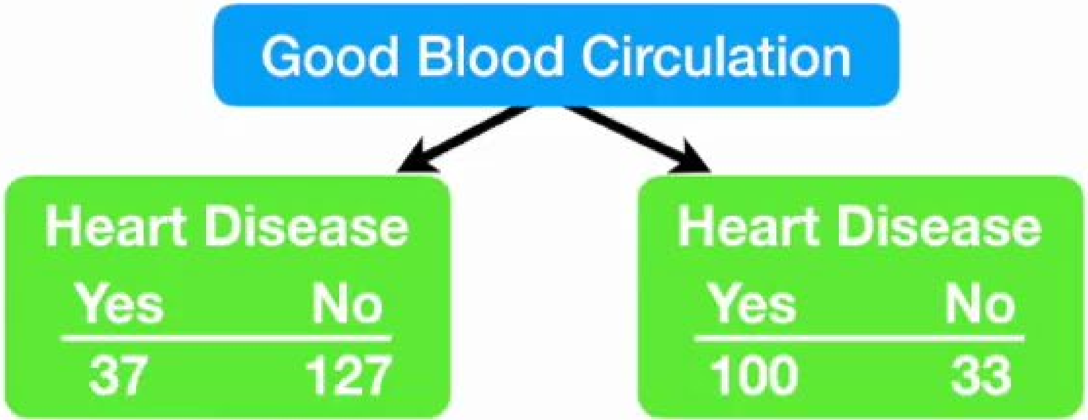
Chest Pain

Heart Disease

Yes	No
105	39

Heart Disease

Yes	No
34	125



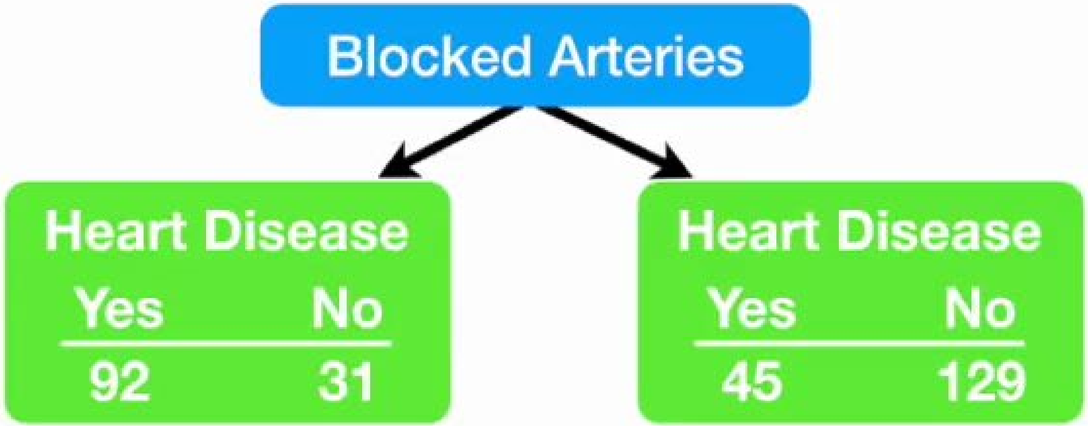
Good Blood Circulation

Heart Disease

Yes	No
37	127

Heart Disease

Yes	No
100	33



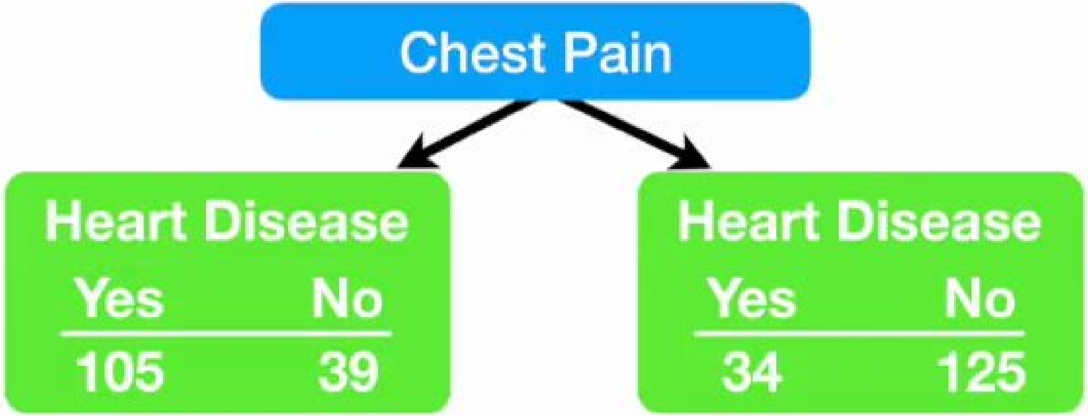
Blocked Arteries

Heart Disease

Yes	No
92	31

Heart Disease

Yes	No
45	129



Chest Pain

Heart Disease

Yes	No
105	39

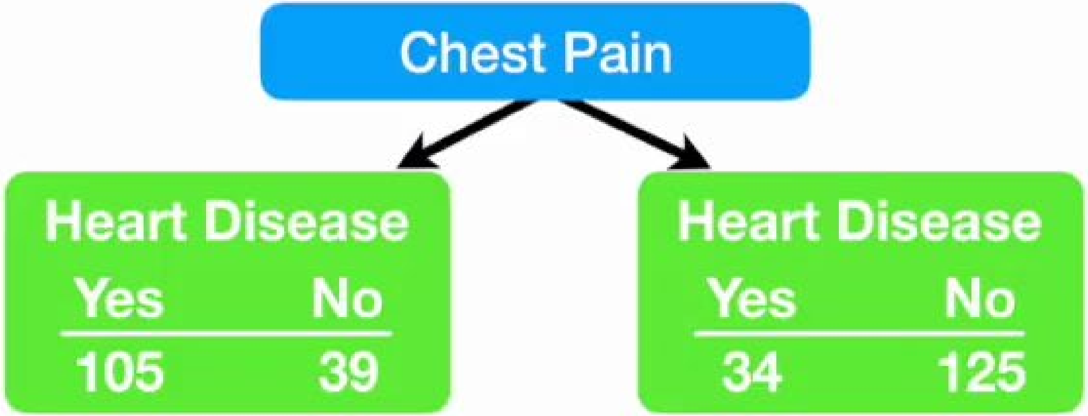
Heart Disease

Yes	No
34	125

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

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

$$= 0.395$$



Chest Pain

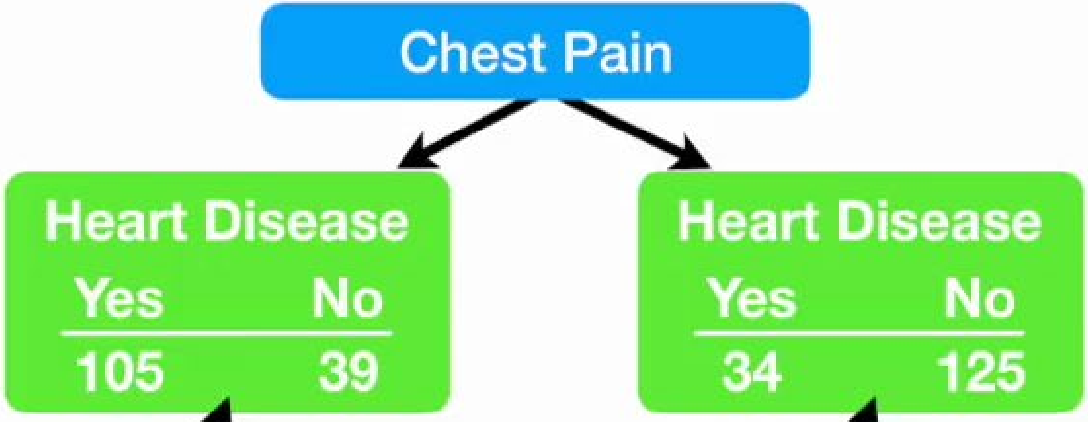
Heart Disease	
Yes	No
105	39

Heart Disease	
Yes	No
34	125

= 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



Chest Pain

Heart Disease

Yes	No
105	39

Heart Disease

Yes	No
34	125

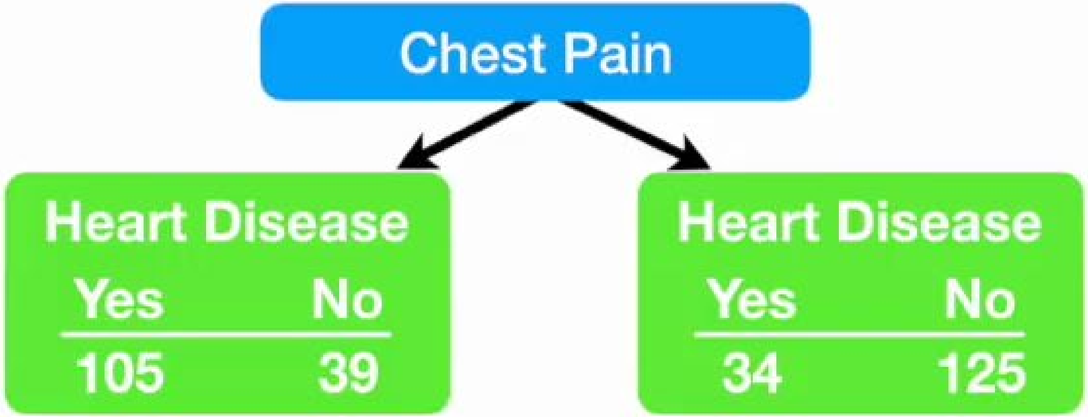
Gini impurity = 0.395

0.336

Because this leaf node represents 144 patients…

… and this leaf node represents 159 patients…

Thus, the total Gini impurity for using Chest Pain to separate patients with and without heart disease is the **weighted average of the leaf node impurities**.



Chest Pain

Heart Disease	
Yes	No
105	39

Heart Disease	
Yes	No
34	125

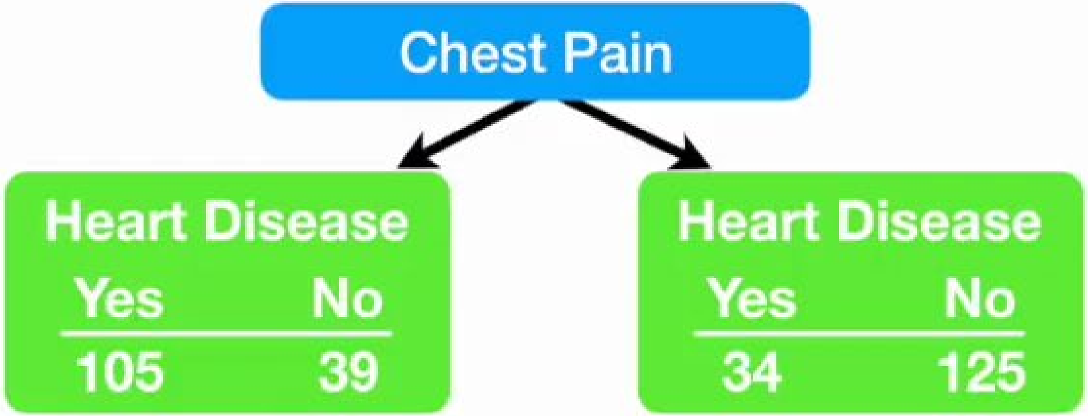
Gini impurity = 0.395 0.336

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

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

$$= 0.364$$

Gini impurity for Chest Pain = 0.364



Chest Pain

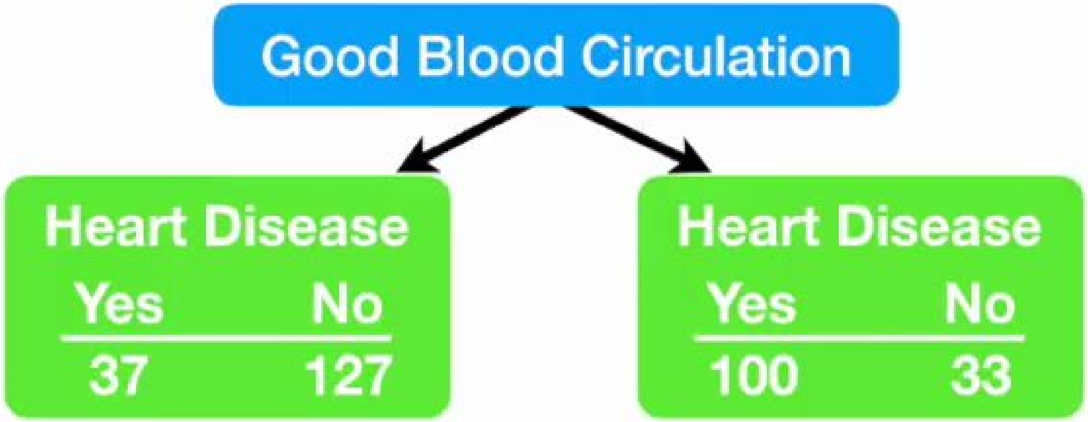
Heart Disease

Yes	No
105	39

Heart Disease

Yes	No
34	125

Gini impurity for Good Blood Circulation = 0.360



Good Blood Circulation

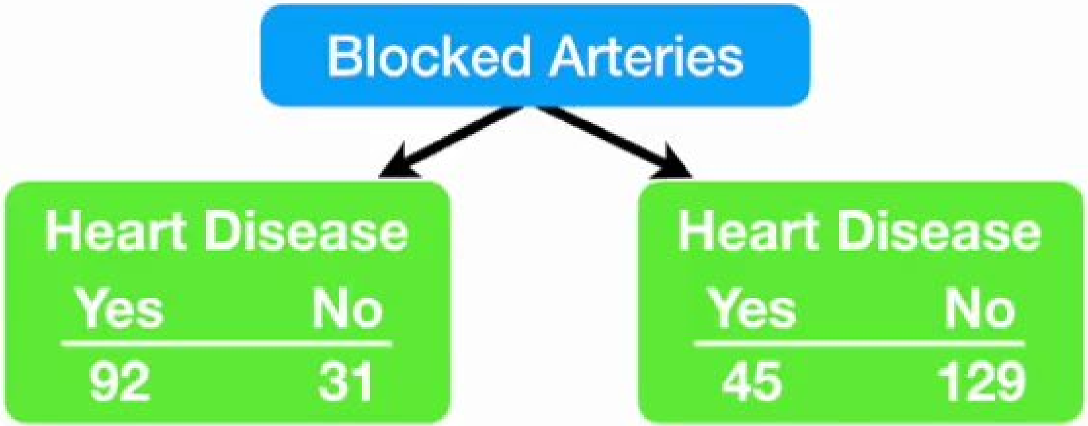
Heart Disease

Yes	No
37	127

Heart Disease

Yes	No
100	33

Gini impurity for Blocked Arteries = 0.381



Blocked Arteries

Heart Disease

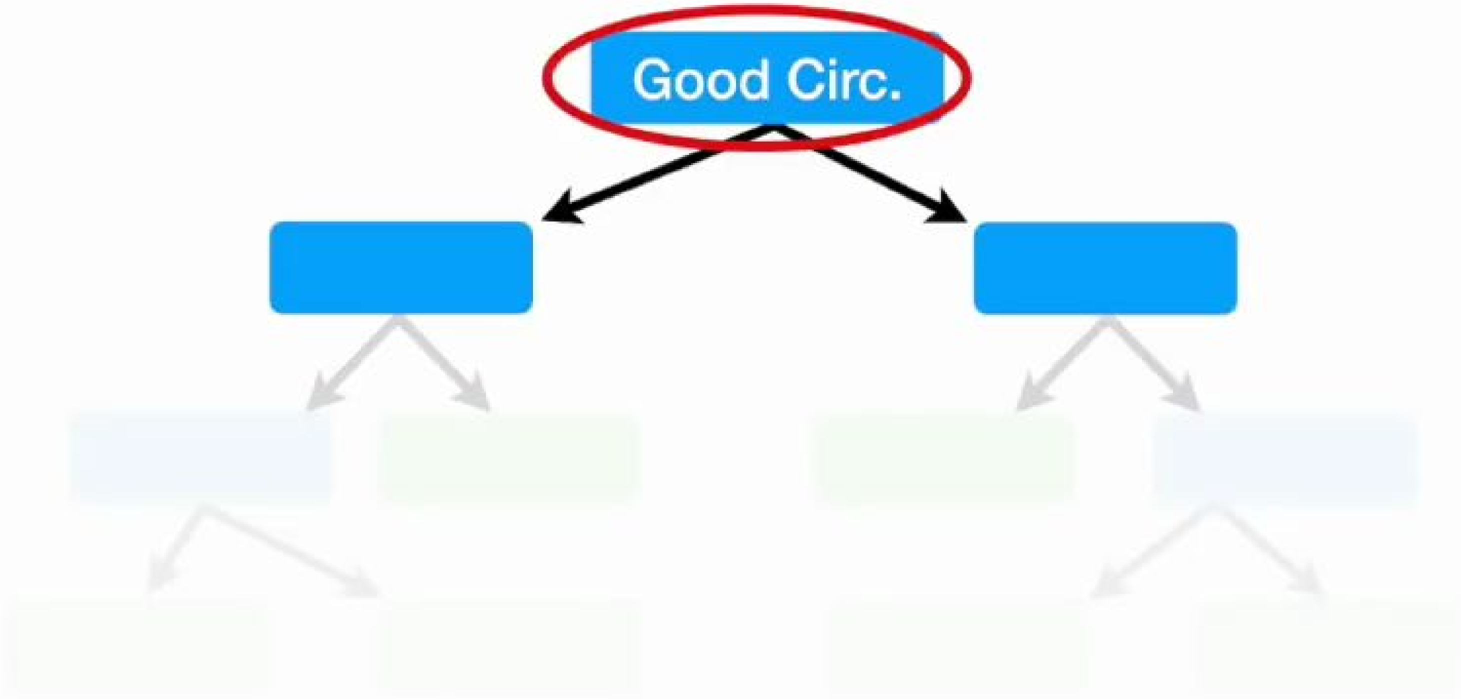
Yes	No
92	31

Heart Disease

Yes	No
45	129

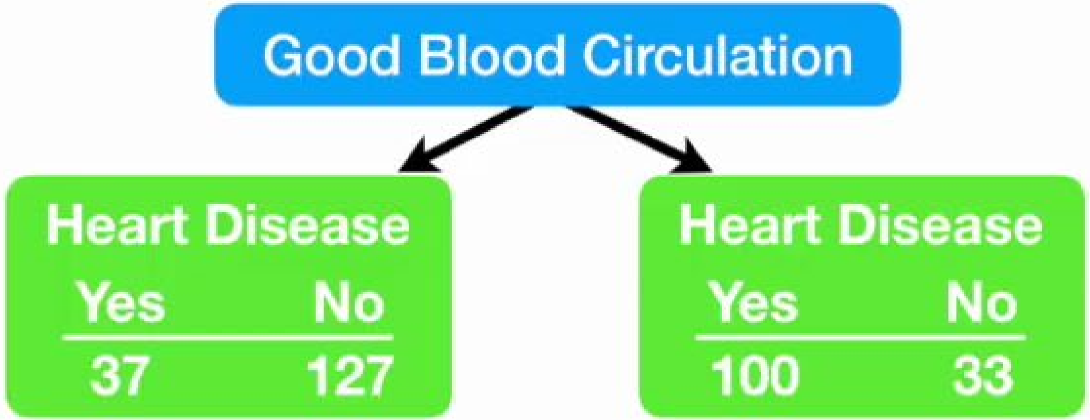
Gini impurity for Chest Pain = 0.364

…so we will use it at the root of the tree.



Gini impurity for Good Blood Circulation = 0.360

Gini impurity for Blocked Arteries = 0.381



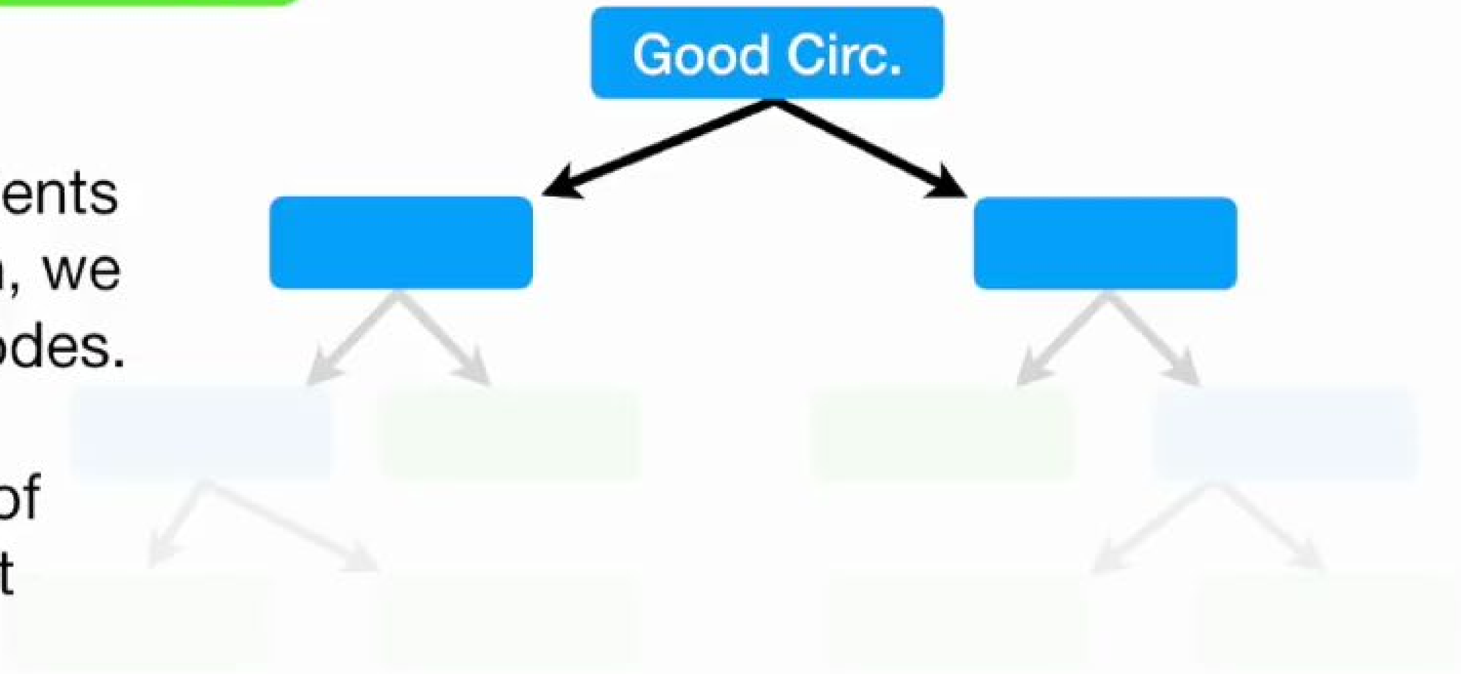
Good Blood Circulation

Heart Disease

Yes	No
37	127

Heart Disease

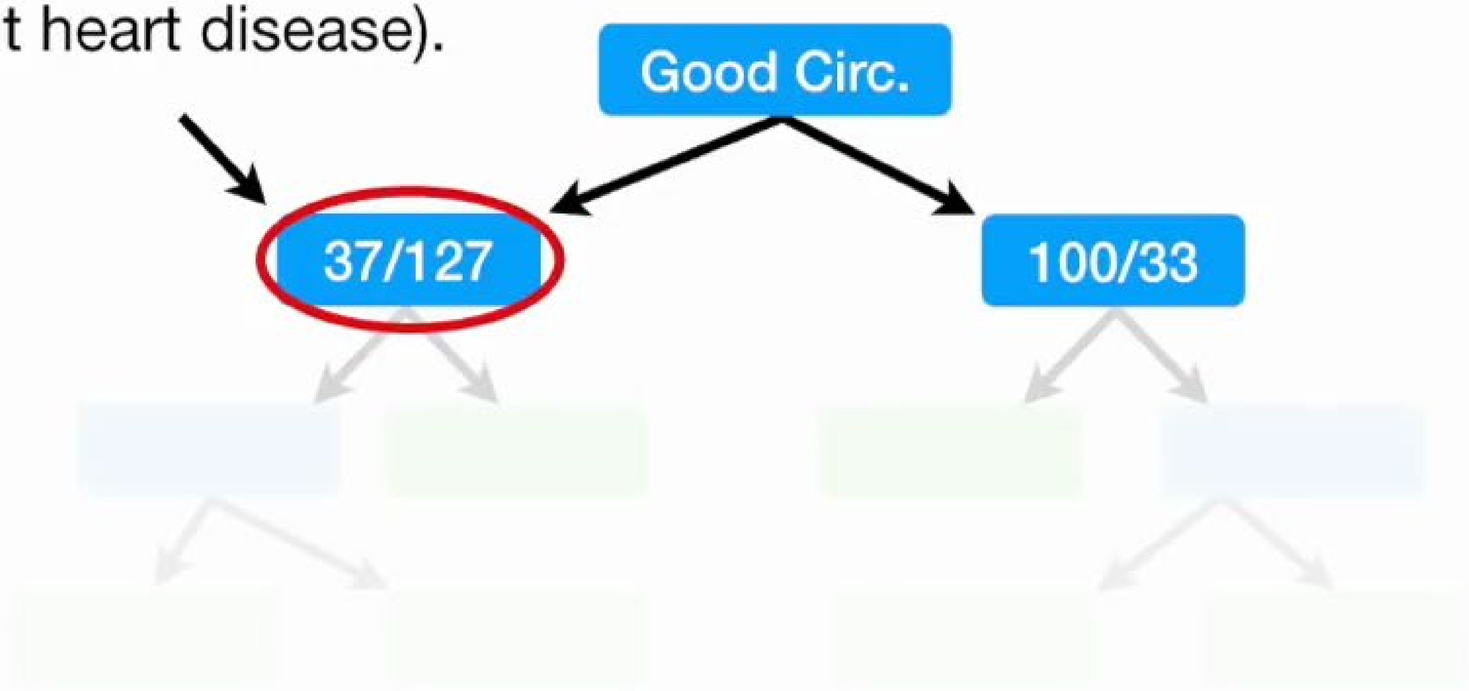
Yes	No
100	33

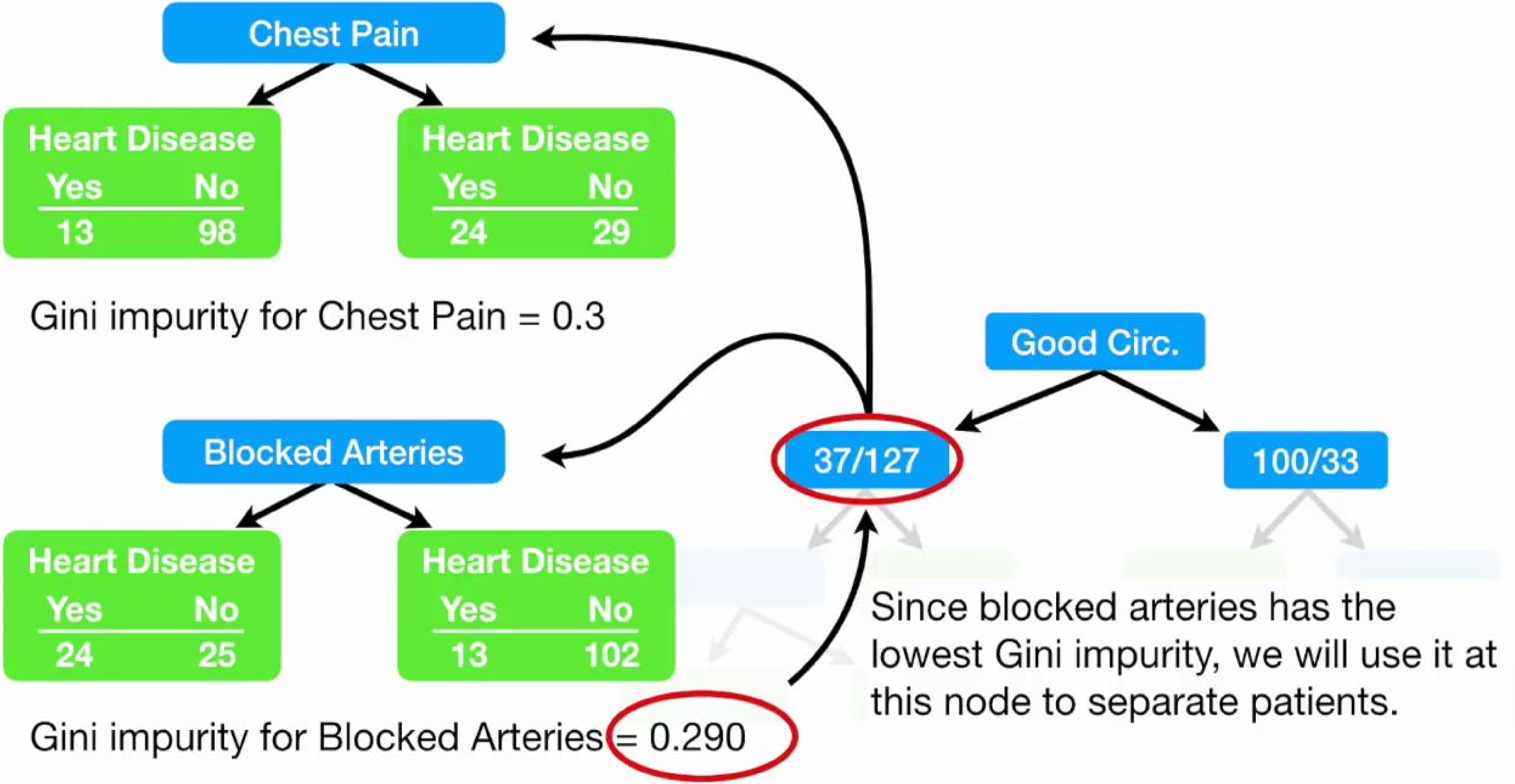


When we divided all of the patients using **Good Blood Circulation**, we ended up with "impure" leaf nodes.

Each leaf contained a mixture of patients with and without Heart Disease.

Now we need to figure how well **chest pain** and **blocked arteries** separate these 164 patients (37 with heart disease and 127 without heart disease).





Chest Pain

Heart Disease	
Yes	No
13	98

Heart Disease	
Yes	No
24	29

Gini impurity for Chest Pain = 0.3

Good Circ.

37/127

100/33

Blocked Arteries

Heart Disease	
Yes	No
24	25

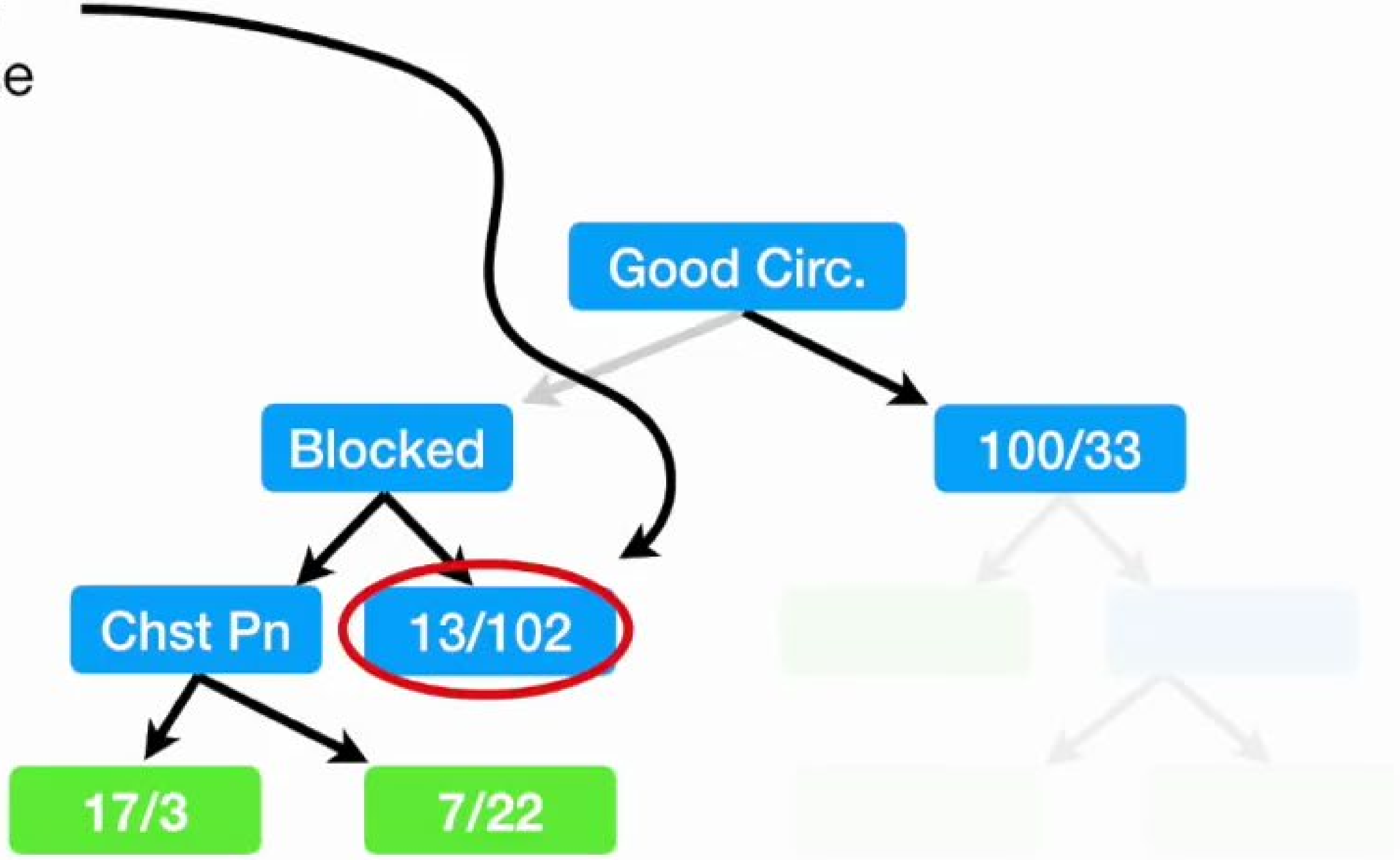
Heart Disease	
Yes	No
13	102

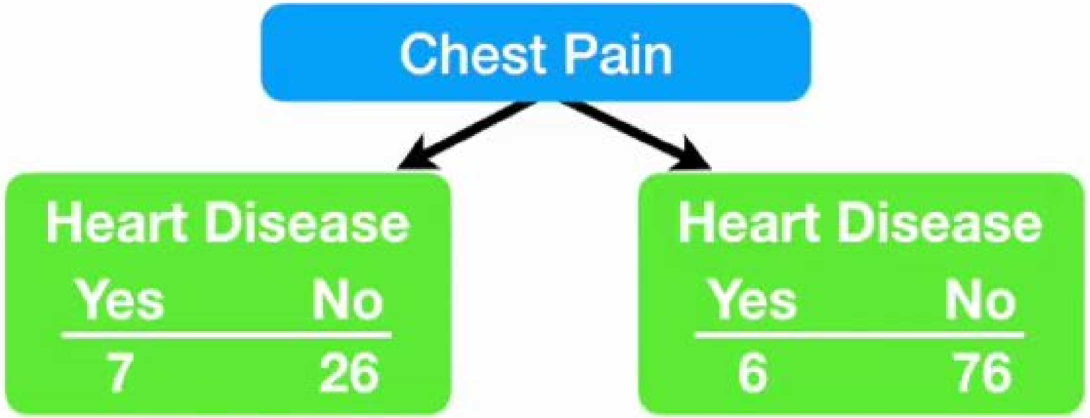
Gini impurity for Blocked Arteries = 0.290

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

Now let’s see what happens when we use chest pain to divide these 115 patients (13 with heart disease and 102 without).

NOTE: The vast majority of the patients in this node (89%) don’t have heart disease.



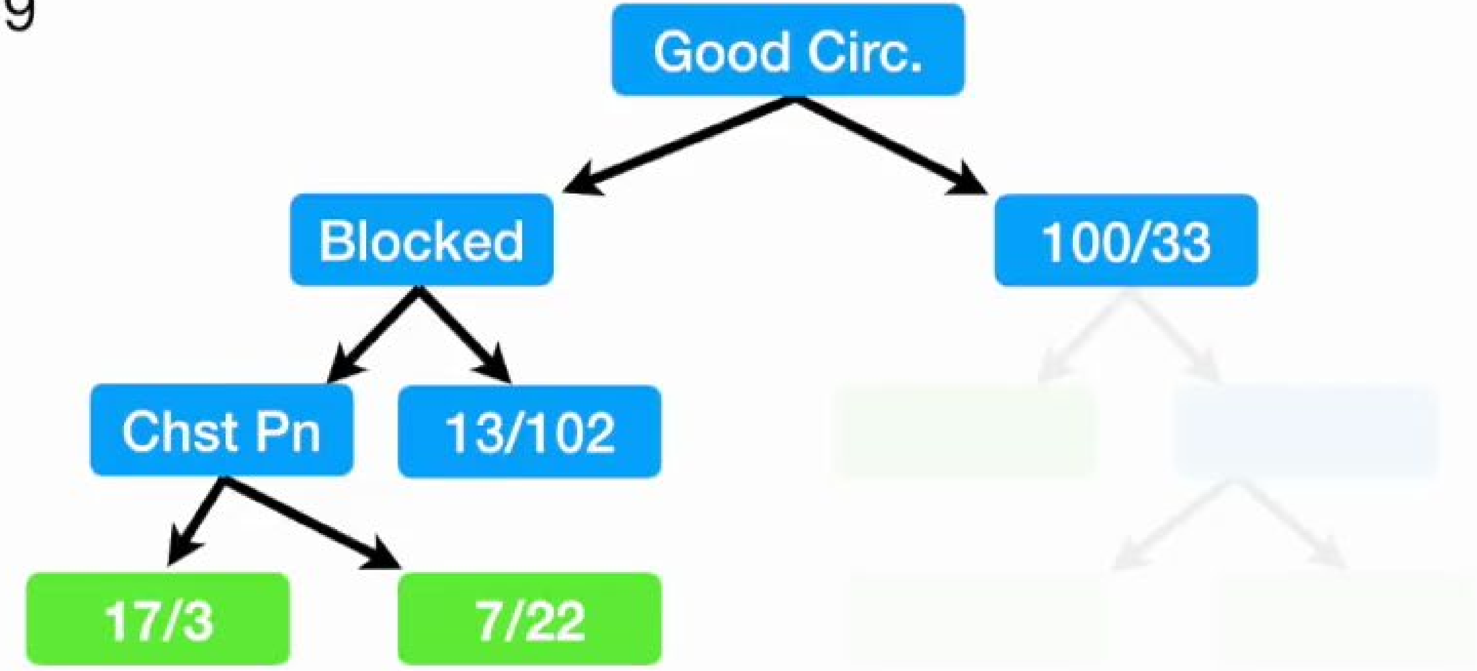


Chest Pain

Heart Disease	
Yes	No
7	26

Heart Disease	
Yes	No
6	76

Gini impurity for Chest Pain = 0.29



Good Circ.

Blocked

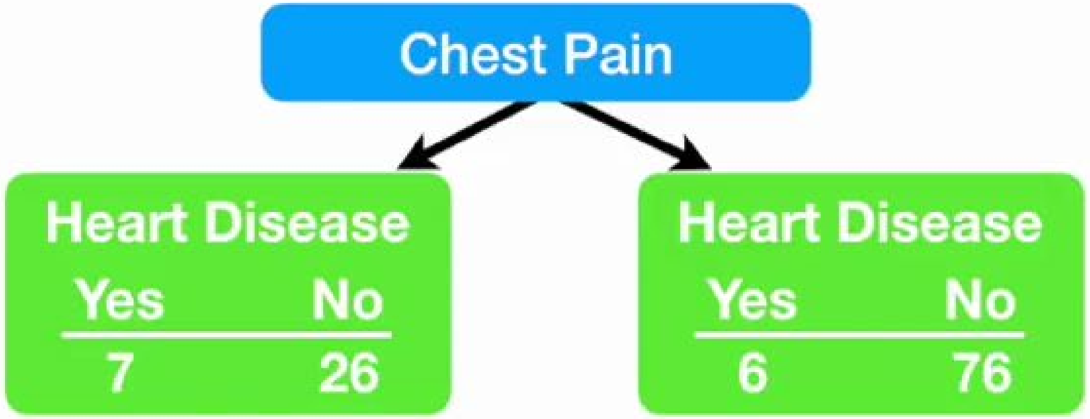
100/33

Chst Pn

13/102

17/3

7/22

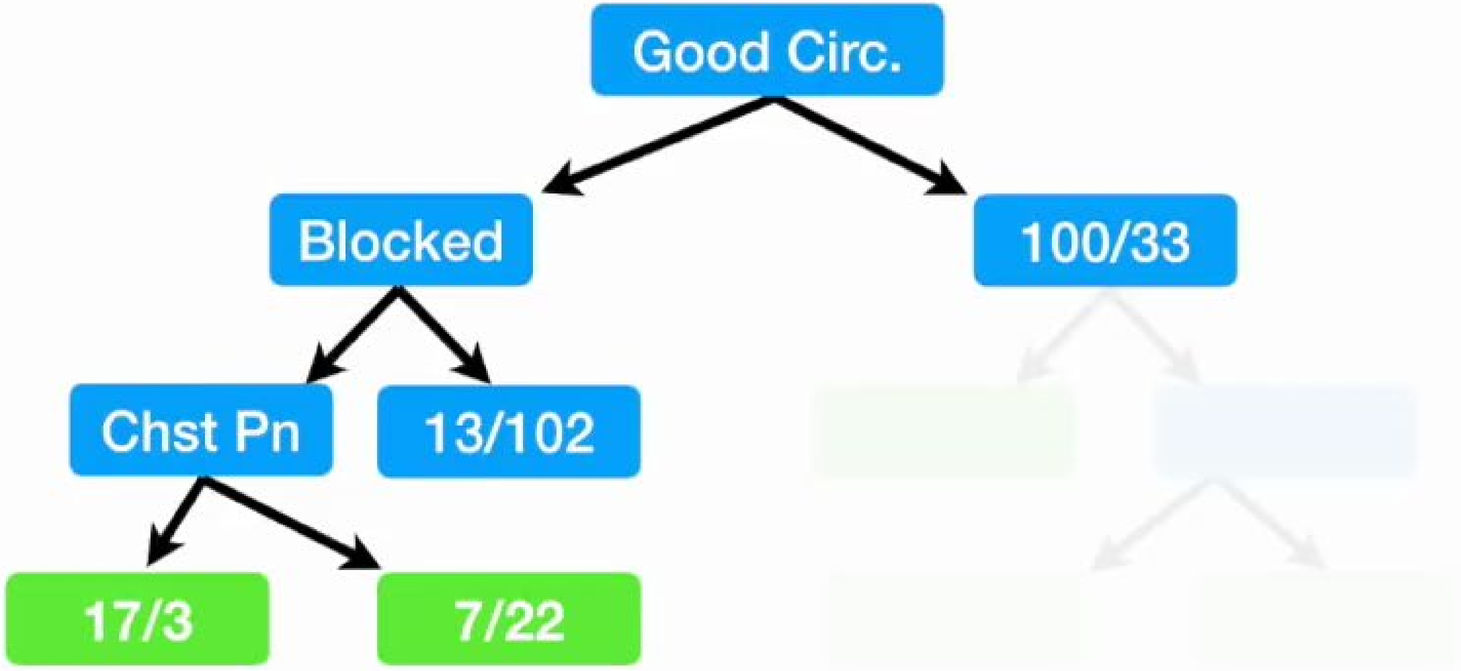


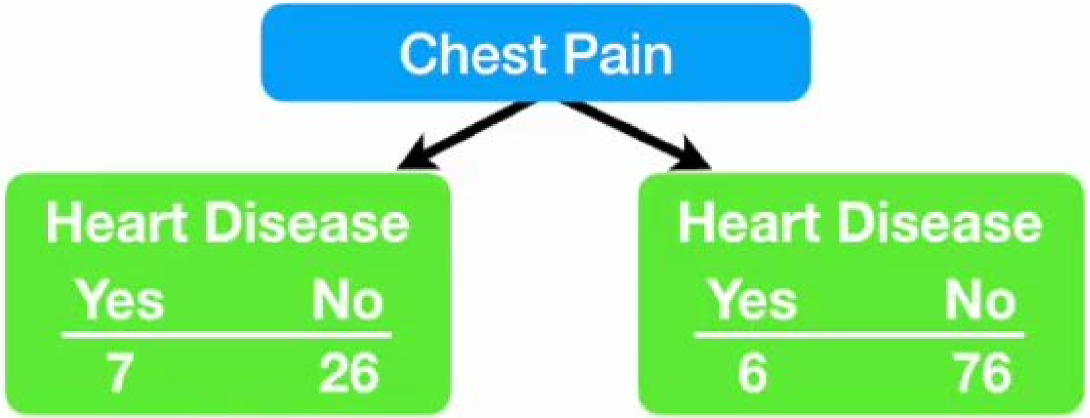
Chest Pain

Heart Disease Yes	Heart Disease No
7	26

Heart Disease Yes	Heart Disease No
6	76

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





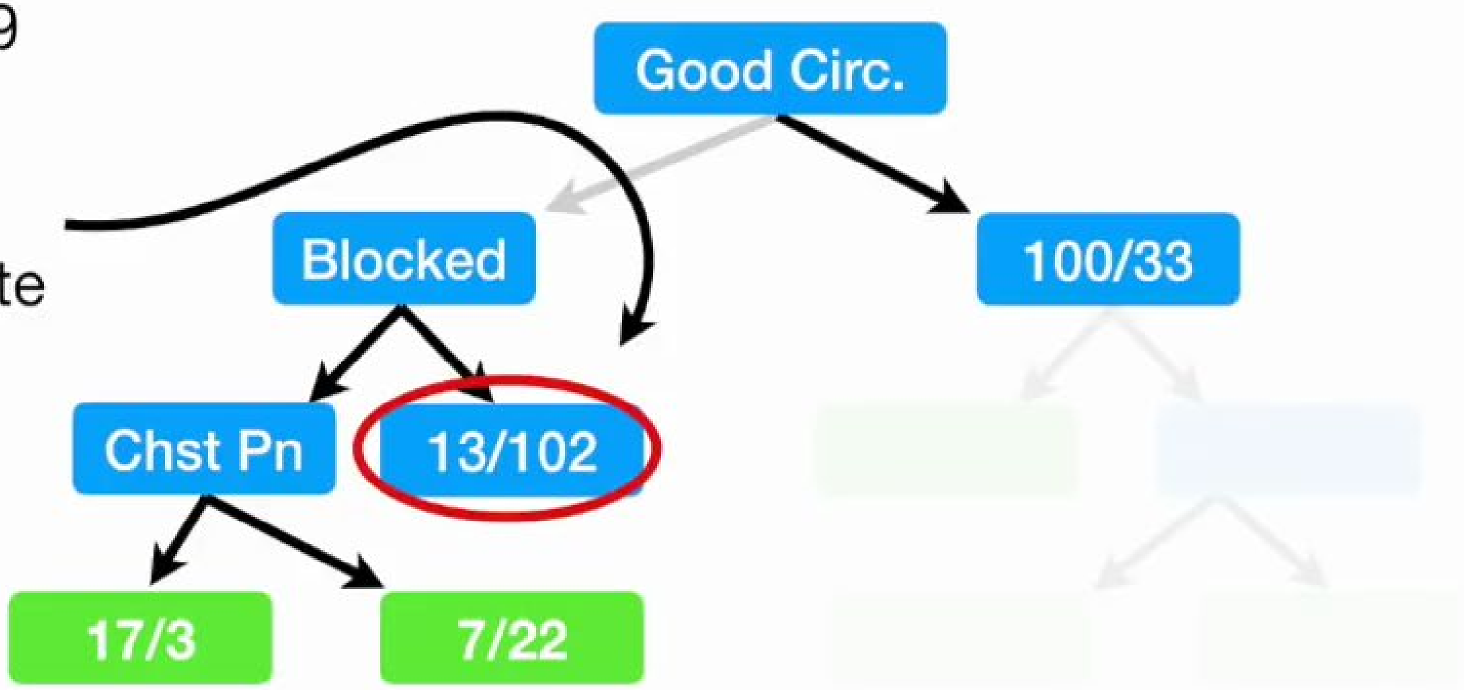
Chest Pain

Heart Disease	
Yes	No
7	26

Heart Disease	
Yes	No
6	76

Gini impurity for Chest Pain = 0.29

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



Good Circ.

Blocked

100/33

Chst Pn

13/102

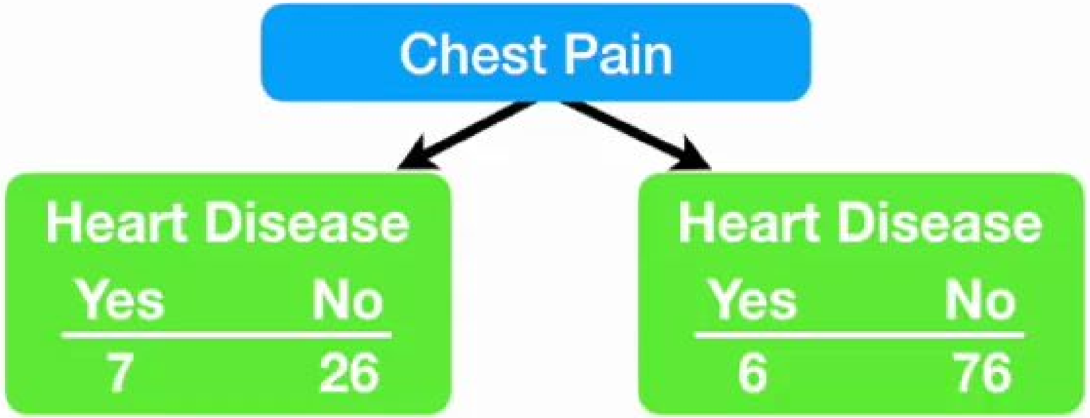
17/3

7/22

= 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



Chest Pain

Heart Disease

Yes	No
7	26

Heart Disease

Yes	No
6	76

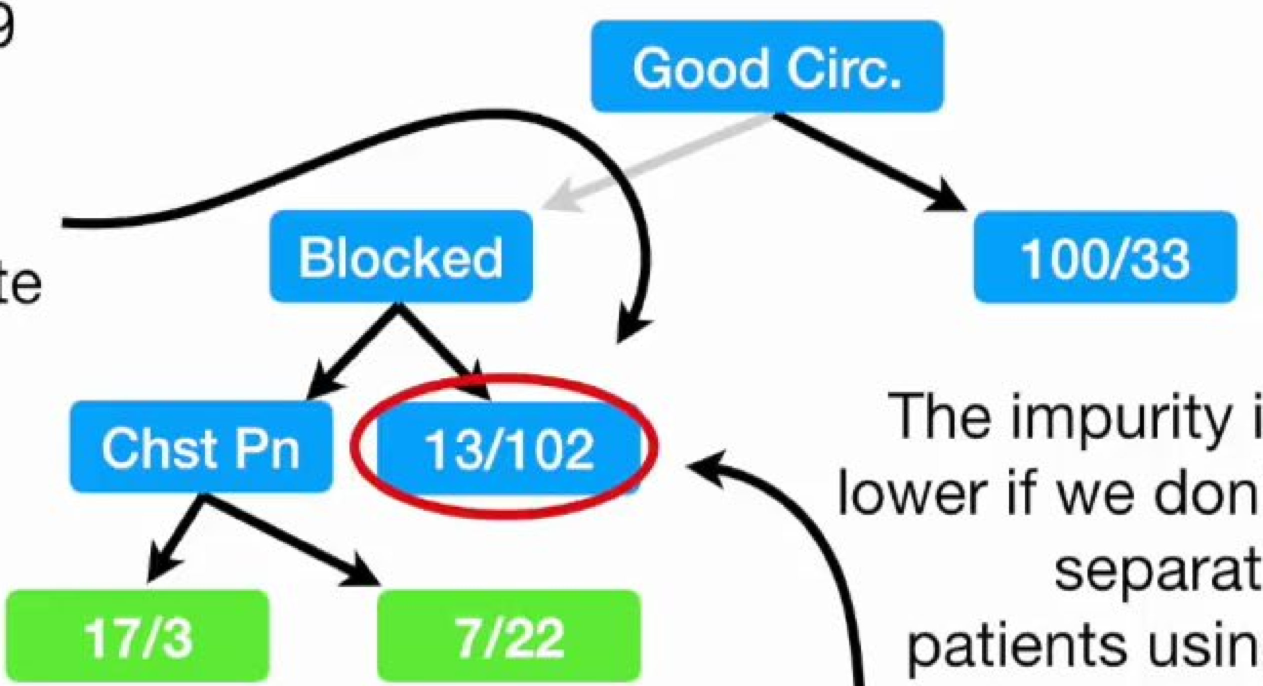
Gini impurity for Chest Pain = 0.29

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

$$= 1 - (\text{the probability of “yes”})^2 - (\text{the probability of “no”})^2$$

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

$$\boxed{= 0.2}$$



Good Circ.

Blocked

100/33

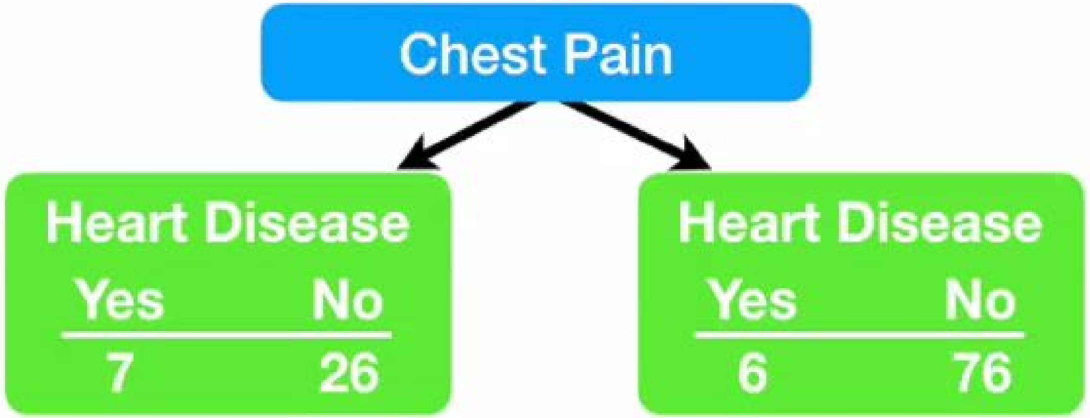
Chst Pn

13/102

17/3

7/22

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



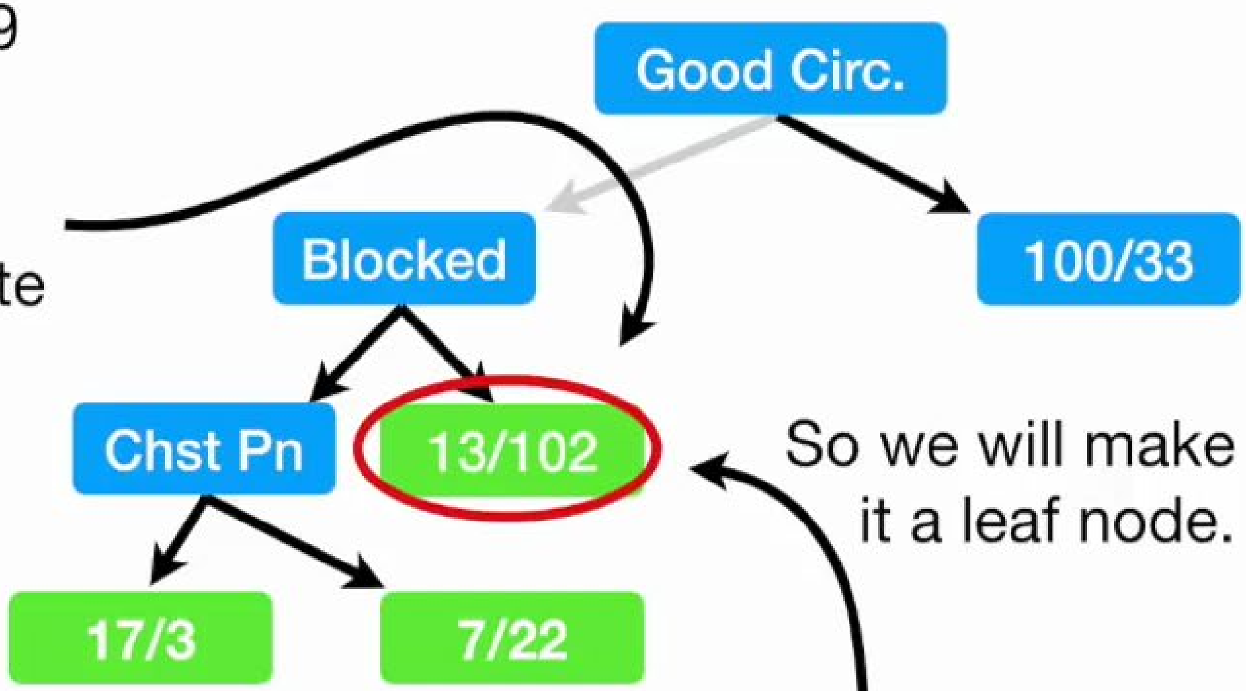
Gini impurity for Chest Pain = 0.29

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

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

$$\boxed{= 0.2}$$



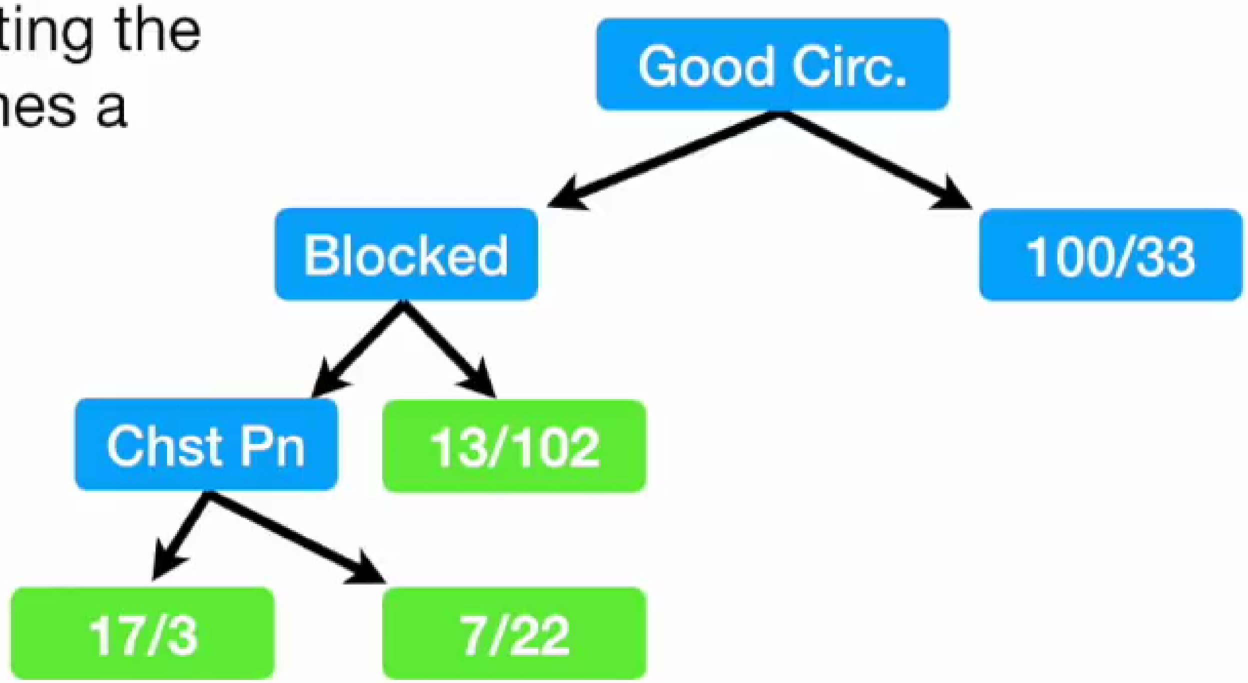
So we will make it a leaf node.

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.

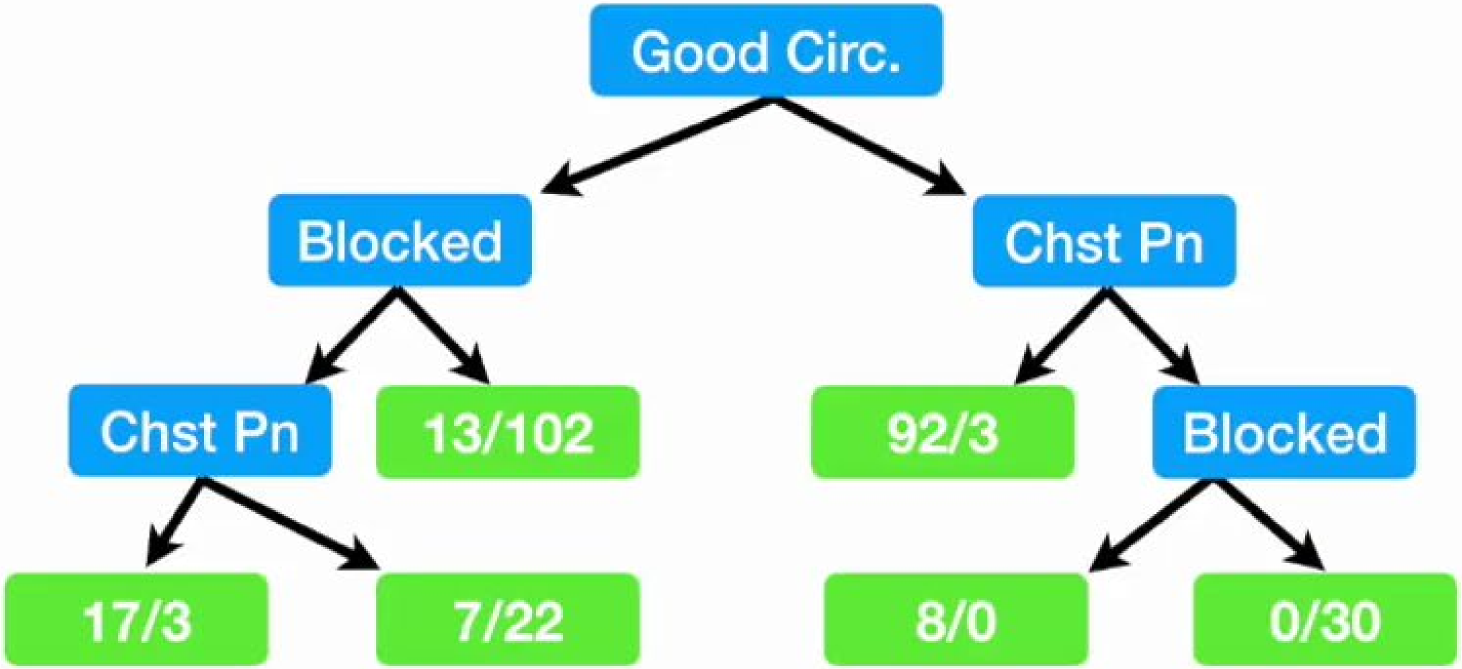
2) If the node itself has the lowest score, than there is no point in separating the patients any more and it becomes a leaf node.

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…

…but what if we have numeric data, like patient weight?

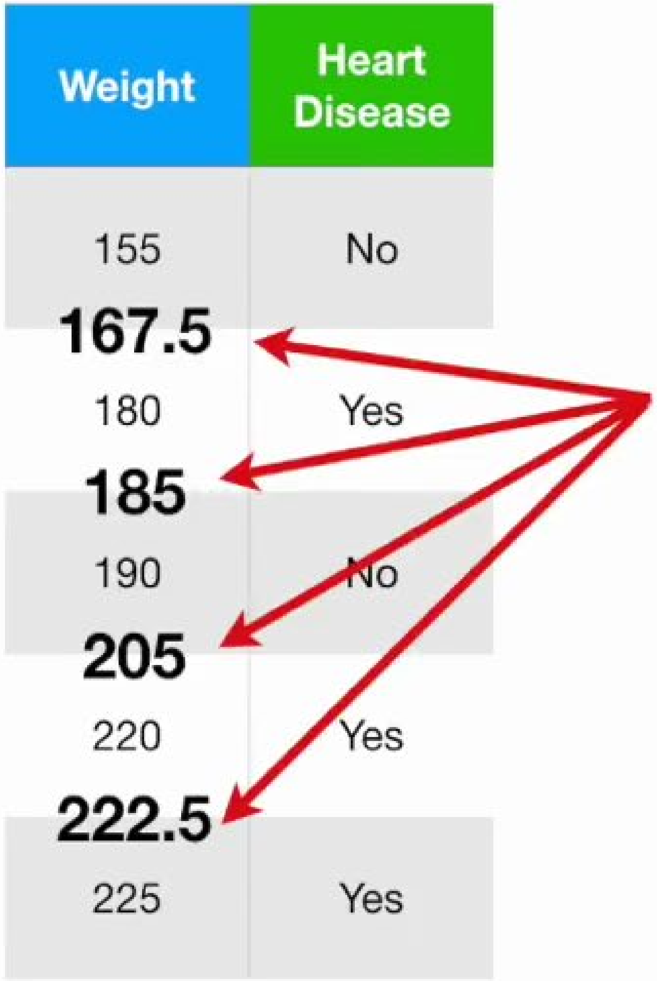


Weight	Heart Disease
220	Yes
180	Yes
225	Yes
190	No
155	No

How do we determine what’s the best weight to use to divide the patients?

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

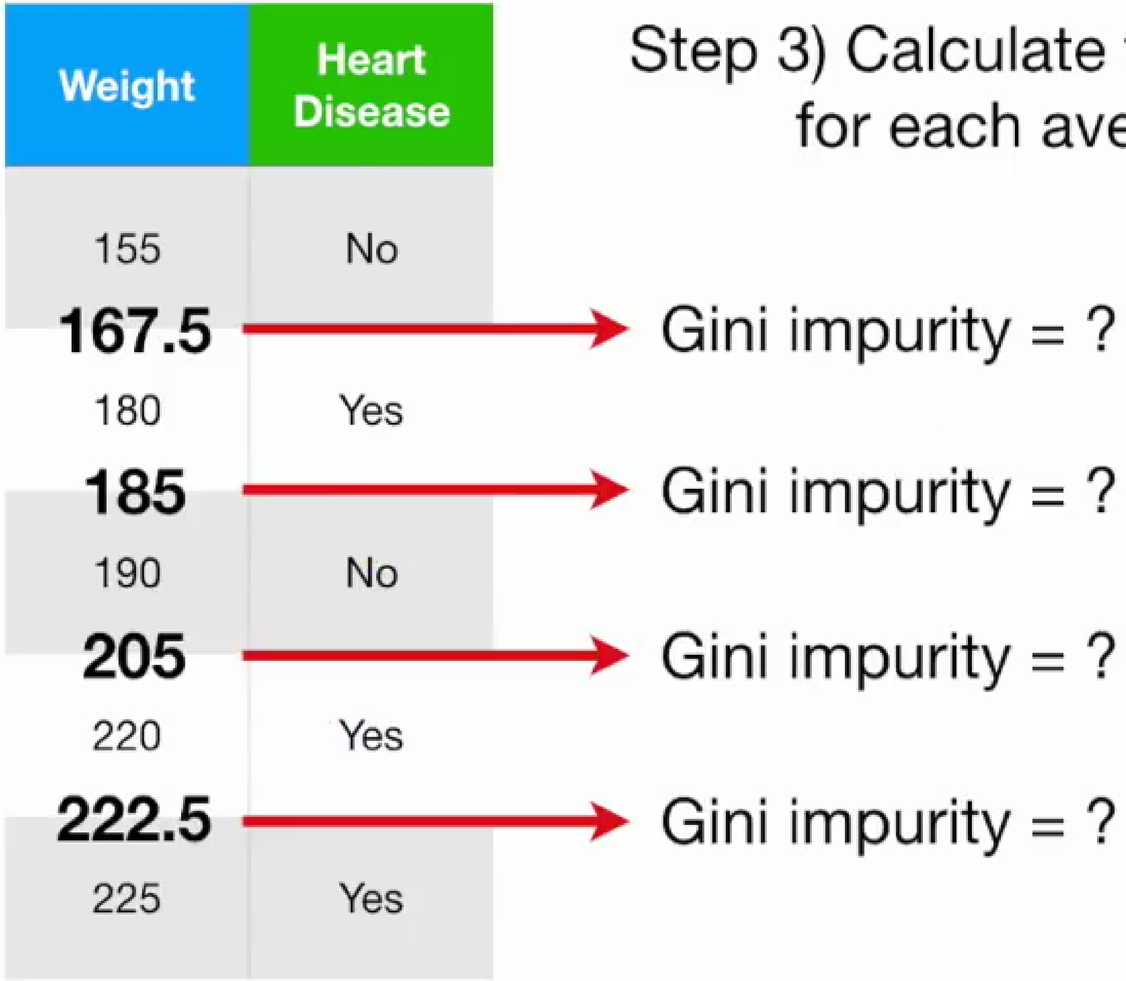
Step 1) Sort the patients by weight, lowest to highest.



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.

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



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

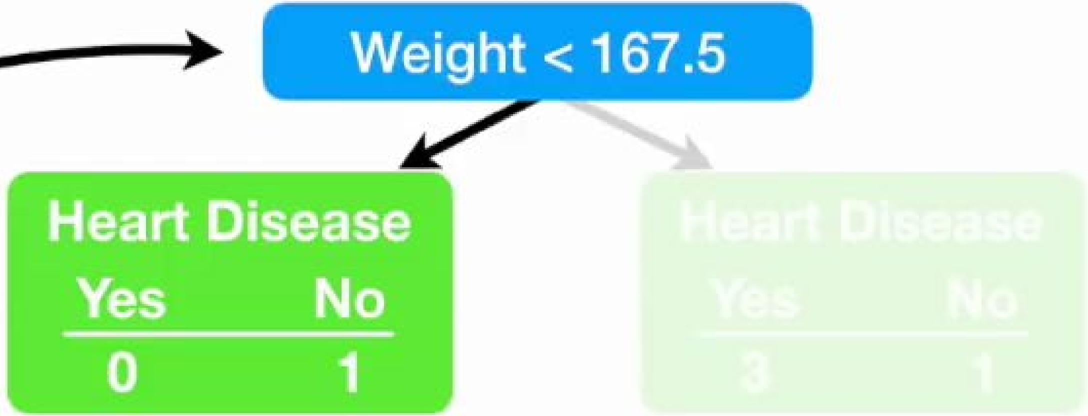
167.5 → Gini impurity = ?

185 → Gini impurity = ?

205 → Gini impurity = ?

222.5 → Gini impurity = ?

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



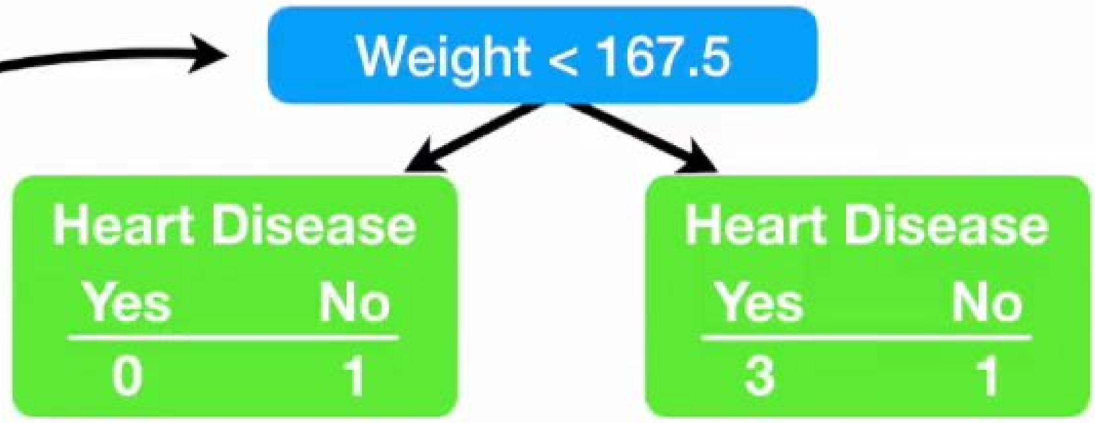
$$\text{Gini impurity} = 1 - (\text{probability of “yes”})^2 - (\text{probability of “no”})^2$$

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

$$= 1 - 0 - 1$$

$$\boxed{= 0}$$

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

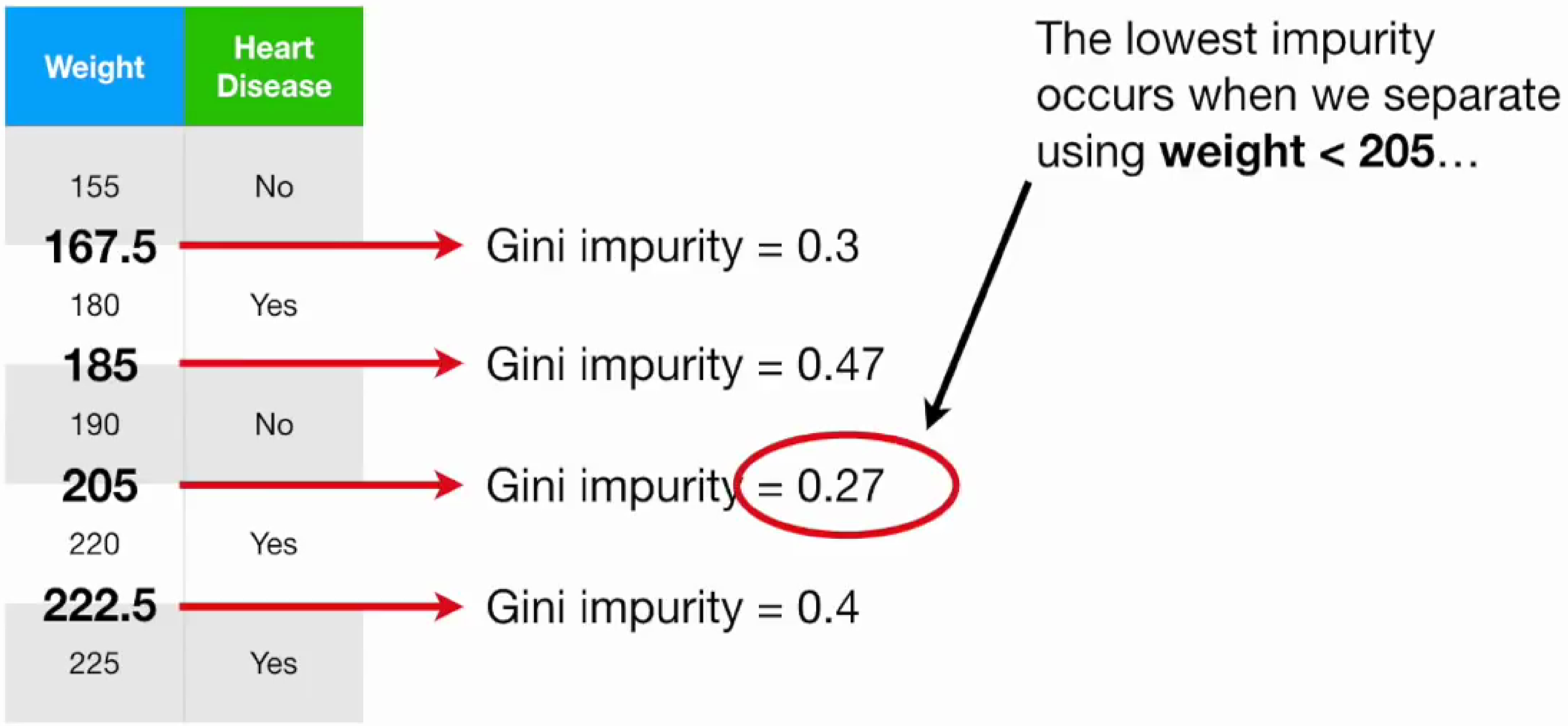


Gini impurity = 0

0.375

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

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



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

167.5 → Gini impurity = 0.3

185 → Gini impurity = 0.47

205 → Gini impurity = 0.27

222.5 → Gini impurity = 0.4

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

Measure of Impurity: GINI

- Gini Index for a given node t :

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

(NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

- Maximum (1 - 1/n_c) when records are equally distributed among all classes, implying least interesting information
- Minimum (0.0) when all records belong to one class, implying most interesting information

Examples for computing GINI

$$GINI(t) = 1 - \sum_{j} [p(j \mid t)]^2$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Gini = 1 – P(C1)^2 – P(C2)^2 = 1 – 0 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Gini = 1 – (1/6)^2 – (5/6)^2 = 0.278

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Gini = 1 – (2/6)^2 – (4/6)^2 = 0.444

Splitting Based on GINI

- Used in CART
- When a node p is split into k partitions (children), the quality of split is computed as,

$$\boxed{GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)}$$

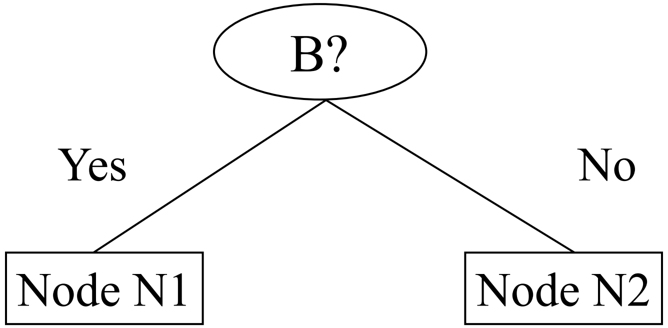
where, n_i = number of records at child i,

n = number of records at node p.

Binary Attributes: Computing GINI Index

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

- Splits into two partitions
- Effect of Weighing partitions:
  - Larger and Purer Partitions are sought for.



	Parent
C1	**6**
C2	**6**
Gini = 0.500	

Gini(N1)
= 1 – $(5/7)^2$ – $(2/7)^2$
= 0.408

Gini(N2)
= 1 – $(1/5)^2$ – $(4/5)^2$
= 0.32

	N1	N2
C1	**5**	**1**
C2	**2**	**4**
Gini=?		

Gini(Children)
= 7/12 * 0.408 +
5/12 * 0.32
= 0.371

Categorical Attributes: Computing Gini Index

- For each distinct value, gather counts for each class in the dataset
- Use the count matrix to make decisions

Multi-way split

	CarType		
	Family	Sports	Luxury
C1	1	2	1
C2	4	1	1
Gini	0.393		

Two-way split
(find best partition of values)

	CarType	
	{Sports, Luxury}	{Family}
C1	3	1
C2	2	4
Gini	0.400	

	CarType	
	{Sports}	{Family, Luxury}
C1	2	2
C2	1	5
Gini	0.419	

Alternative Splitting Criteria based on INFO

- Entropy at a given node t:

$$\boxed{Entropy(t) = -\sum_{j} p(j \mid t) \log p(j \mid t)}$$

(NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

- Measures homogeneity of a node.
  - Maximum ($\log n_c$) when records are equally distributed among all classes implying least information
  - Minimum (0.0) when all records belong to one class, implying most information
- Entropy based computations are similar to the GINI index computations

Examples for computing Entropy

$$Entropy(t) = -\sum_{j} p(j\,|\,t)\log_{2} p(j\,|\,t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0   P(C2) = 6/6 = 1

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6   P(C2) = 5/6

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (1/6) = 0.65

C1	**2**
C2	**4**

P(C1) = 2/6   P(C2) = 4/6

Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

Splitting Based on INFO...

- Information Gain:

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

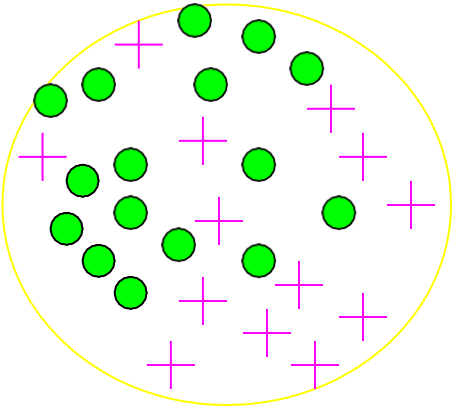
Parent Node, p is split into k partitions;

n_i is number of records in partition i

- Measures Reduction in Entropy achieved because of the split. Choose the split that achieves most reduction (maximizes GAIN)
- Used in ID3 and C4.5
- Disadvantage: Tends to prefer splits that result in large number of partitions, each being small but pure.

Entropy: a common way to measure impurity

- Entropy = $\sum_i - p_i \log_2 p_i$



p_i is the probability of class i

Compute it as the proportion of class i in the set.

- Entropy comes from information theory. The higher the entropy the more the information content.

What does that mean for learning from examples?

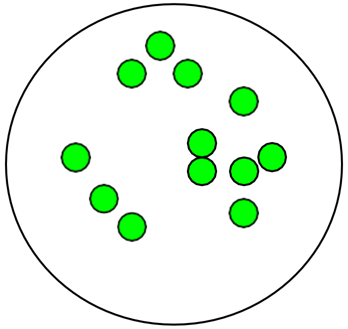
Based on slide by Pedro Domingos

2-Class Cases:

- What is the entropy of a group in which all examples belong to the same class?
  - entropy = - 1 $\log_2 1$ = 0

not a good training set for learning

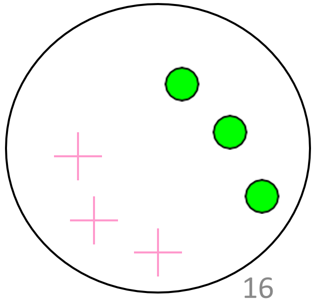
Minimum impurity



- What is the entropy of a group with 50% in either class?
  - entropy = -0.5 $\log_2 0.5$ – 0.5 $\log_2 0.5$ =1

good training set for learning

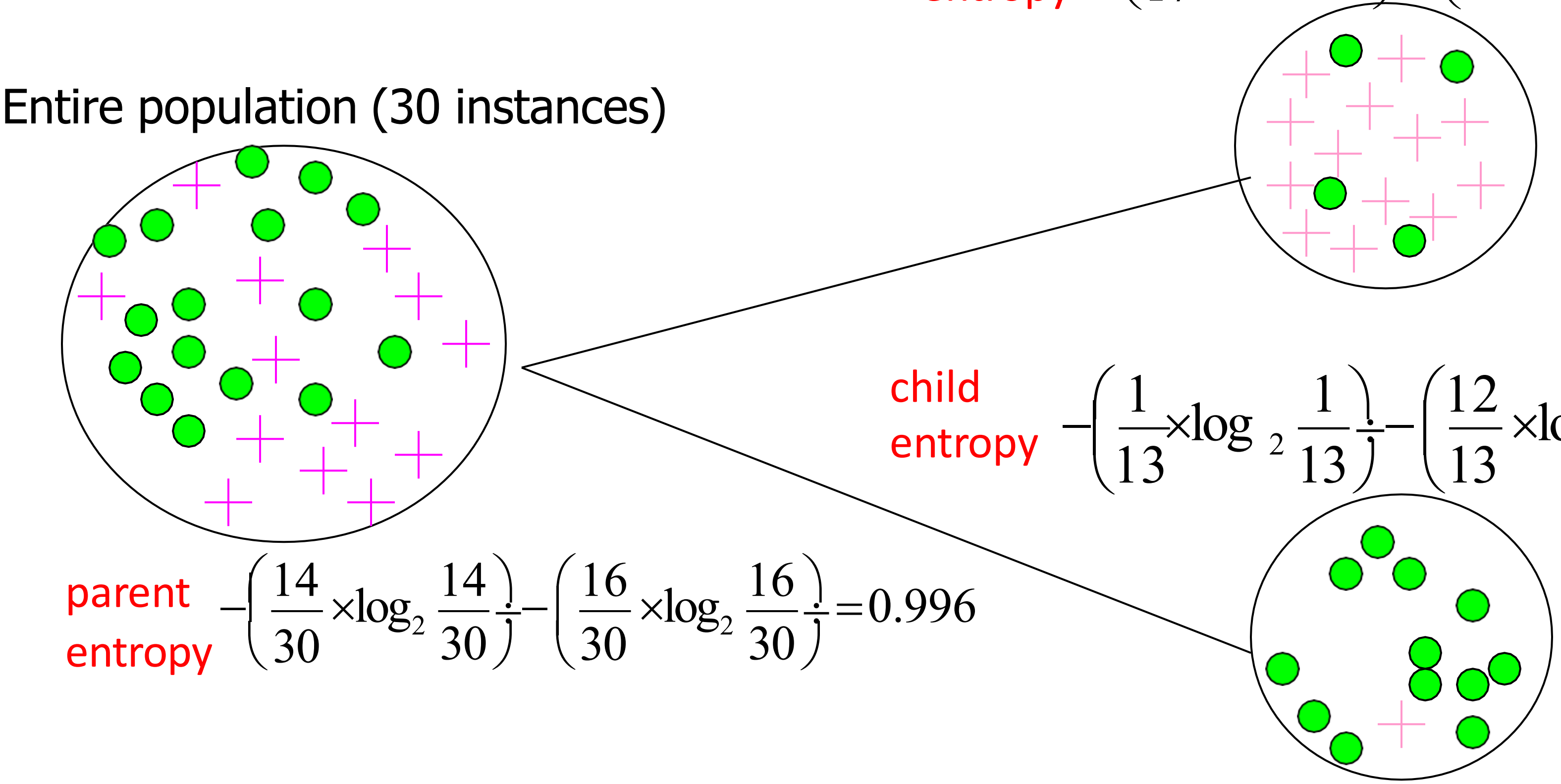
Maximum impurity



Calculating Information Gain

Information Gain = entropy(parent) – [average entropy(children)]

child entropy $-\left(\frac{13}{17}\times\log_2\frac{13}{17}\right)-\left(\frac{4}{17}\times\log_2\frac{4}{17}\right)=0.787$



child entropy $-\left(\frac{1}{13}\times\log_2\frac{1}{13}\right)-\left(\frac{12}{13}\times\log_2\frac{12}{13}\right)=0.391$

parent entropy $-\left(\frac{14}{30}\times\log_2\frac{14}{30}\right)-\left(\frac{16}{30}\times\log_2\frac{16}{30}\right)=0.996$

(Weighted) Average Entropy of Children = $\left(\frac{17}{30}\times0.787\right)+\left(\frac{13}{30}\times0.391\right)=0.615$

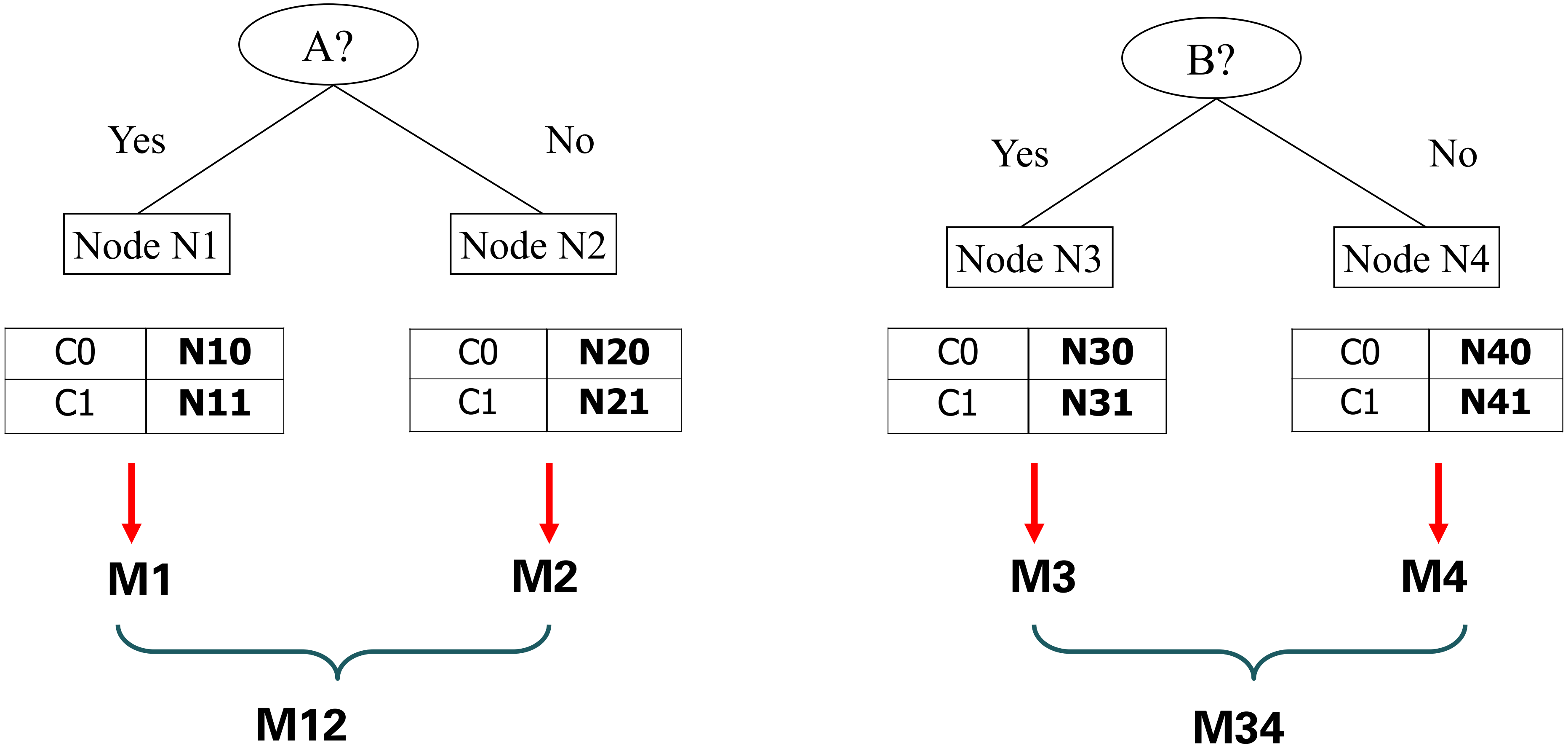
Information Gain= 0.996 - 0.615 = 0.38

How to Find the Best Split

Before Splitting:

C0	**N00**
C1	**N01**

→ **M0**



C0	**N10**
C1	**N11**

C0	**N20**
C1	**N21**

C0	**N30**
C1	**N31**

C0	**N40**
C1	**N41**

M1 **M2** **M3** **M4**

M12 **M34**

Gain = M0 – M12 vs M0 – M34