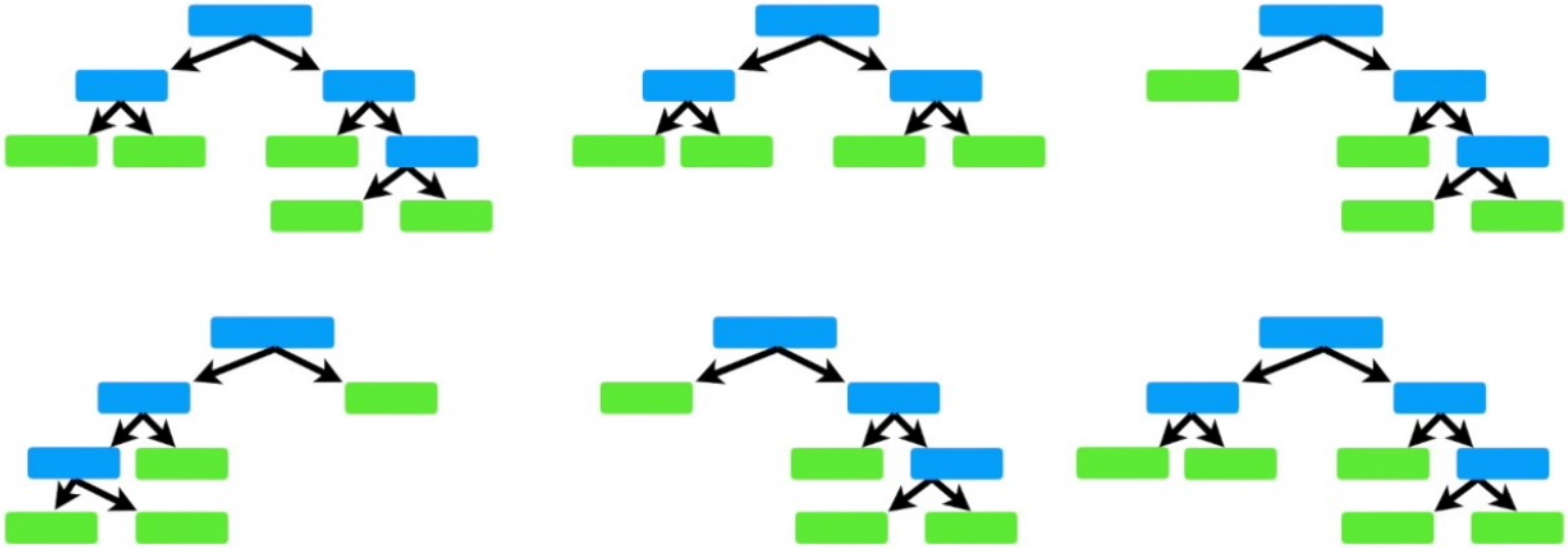
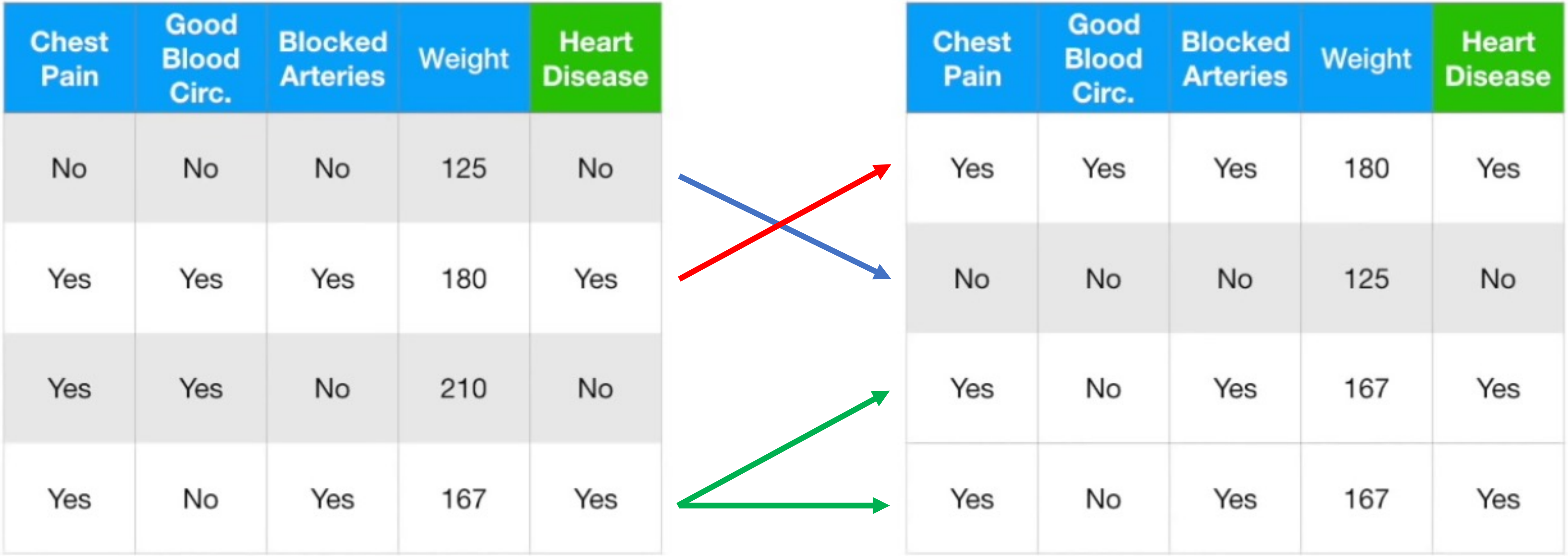
Random Forest

Idea: build a forest of trees and combine their results



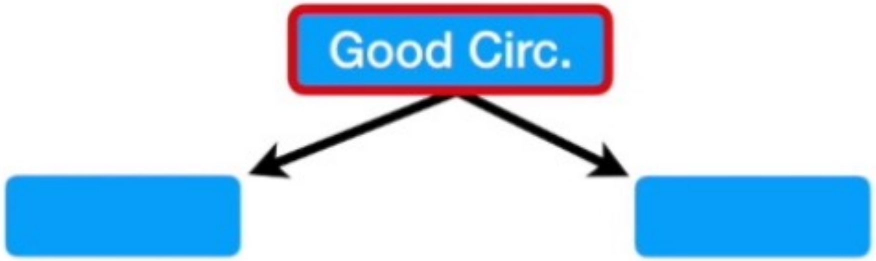
Bootstrapped datasets

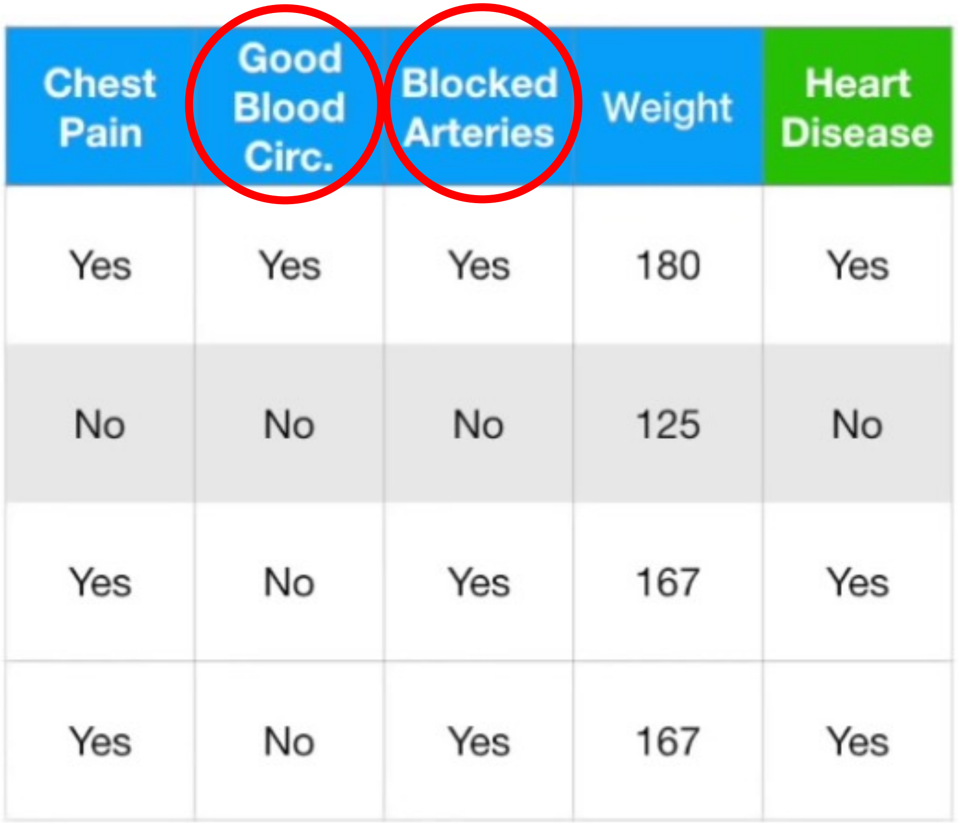


Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	No	Yes	167	Yes

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	Yes	180	Yes
No	No	No	125	No
Yes	No	Yes	167	Yes
Yes	No	Yes	167	Yes

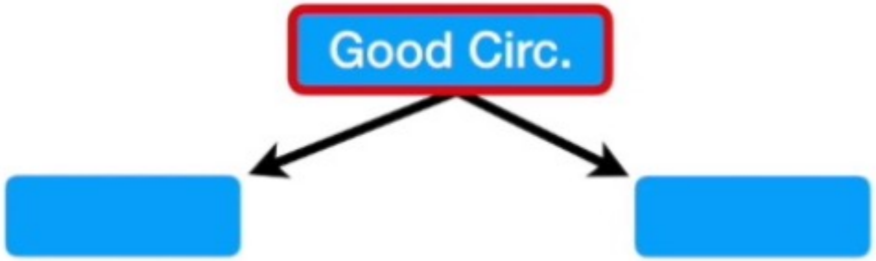
Select a subset of the features and start the procedure to build the corresponding tree

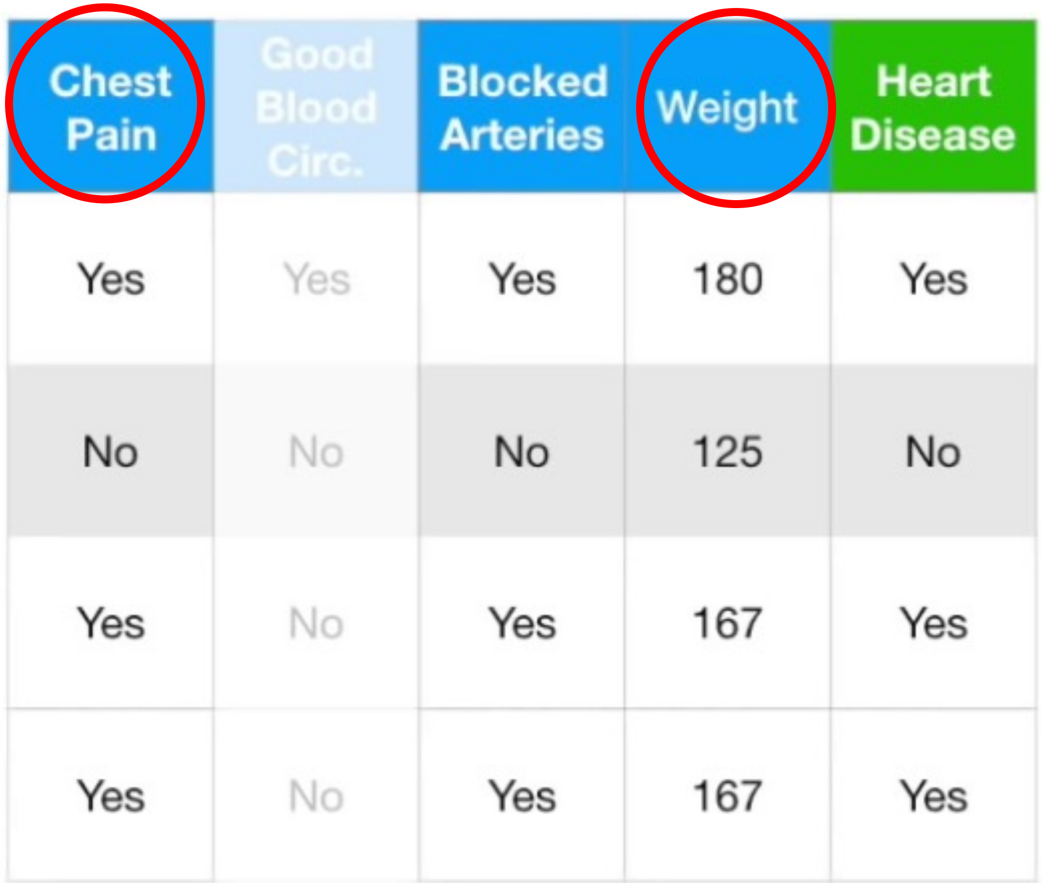




Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	Yes	180	Yes
No	No	No	125	No
Yes	No	Yes	167	Yes
Yes	No	Yes	167	Yes

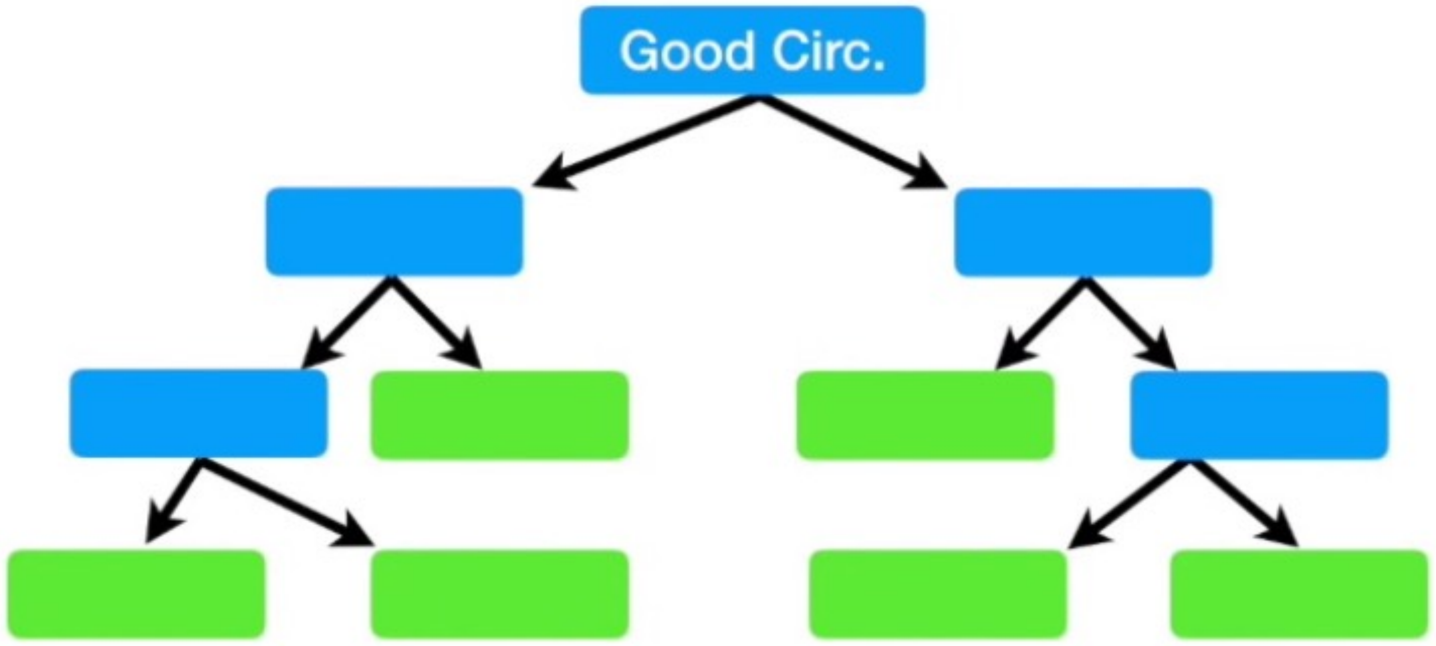
Continue building the tree by considering the remaining features





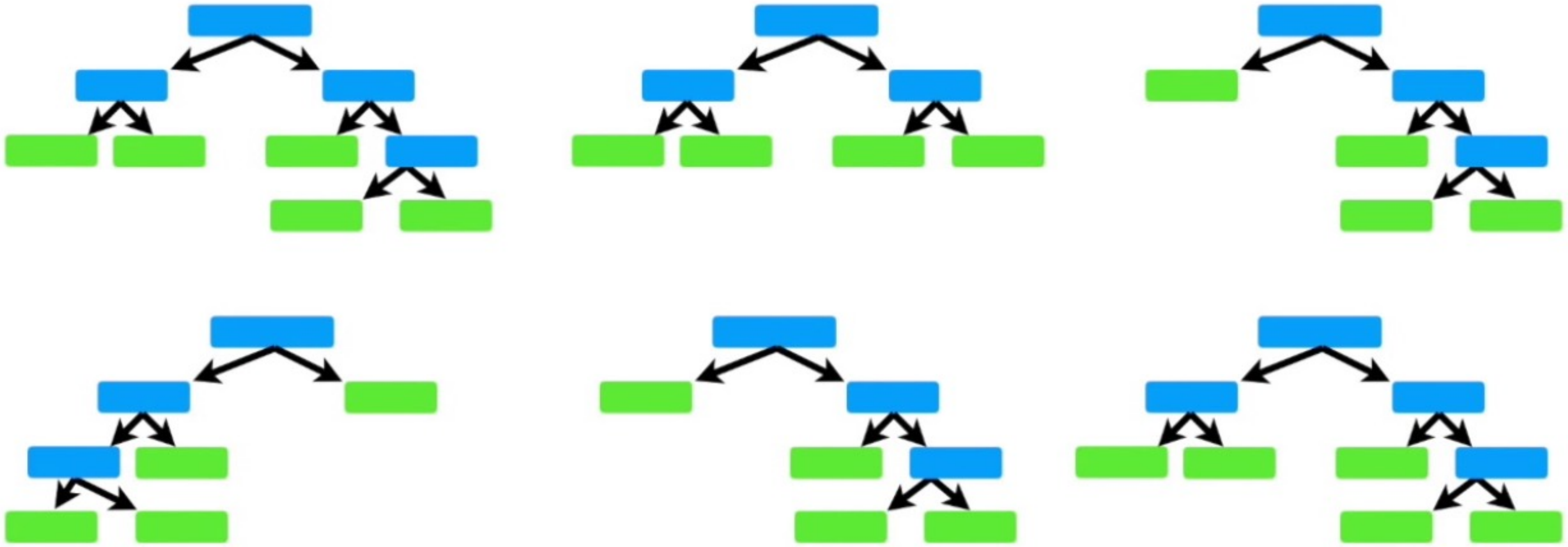
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	Yes	180	Yes
No	No	No	125	No
Yes	No	Yes	167	Yes
Yes	No	Yes	167	Yes

Build the final tree

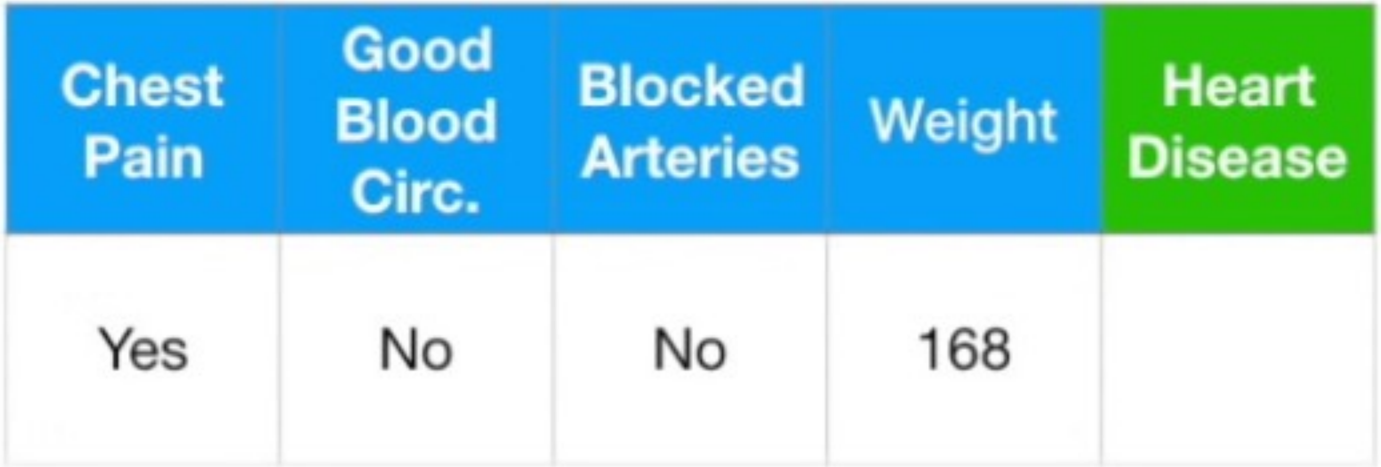


Buid the tree only considering a random subset of features at each step.

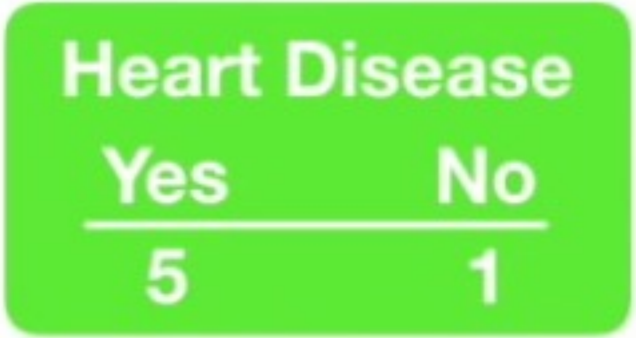
Create a forest of trees from bootstrapped datasets



Classify a new sample with «bagging» (bootstrapping + aggregation)

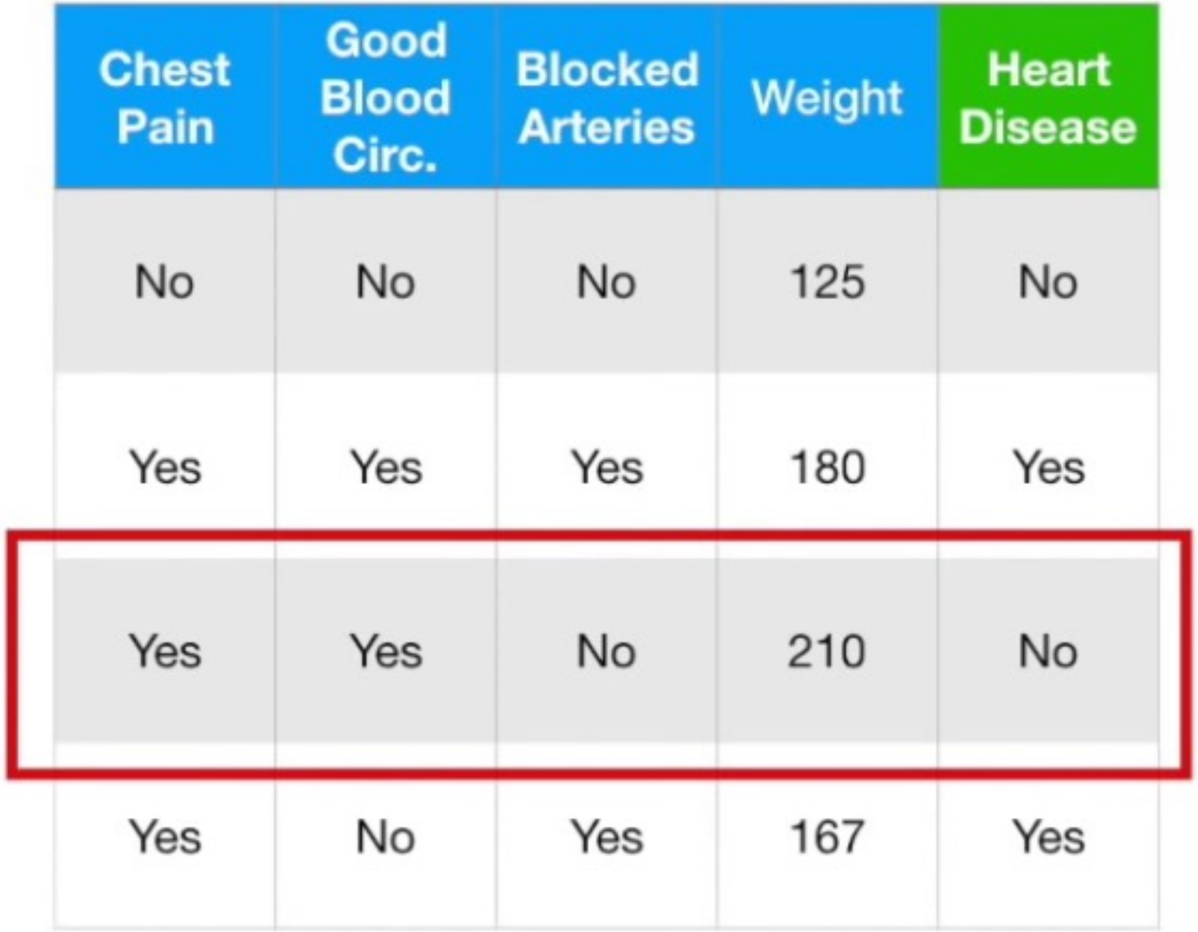


Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	No	No	168	



Heart Disease	
Yes	No
5	1

Evaluate the accuracy of a Random Forest: Out-of-Bag Error



Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	No	Yes	167	Yes

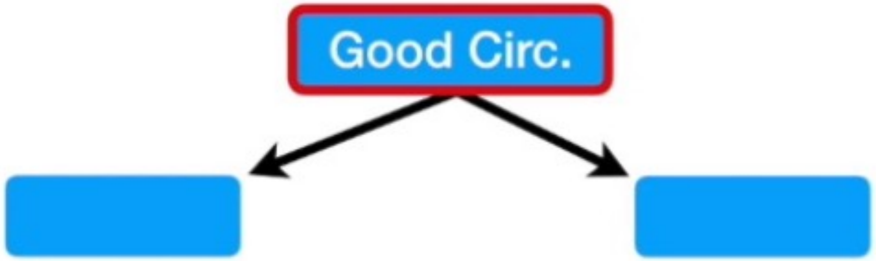
- For each tree

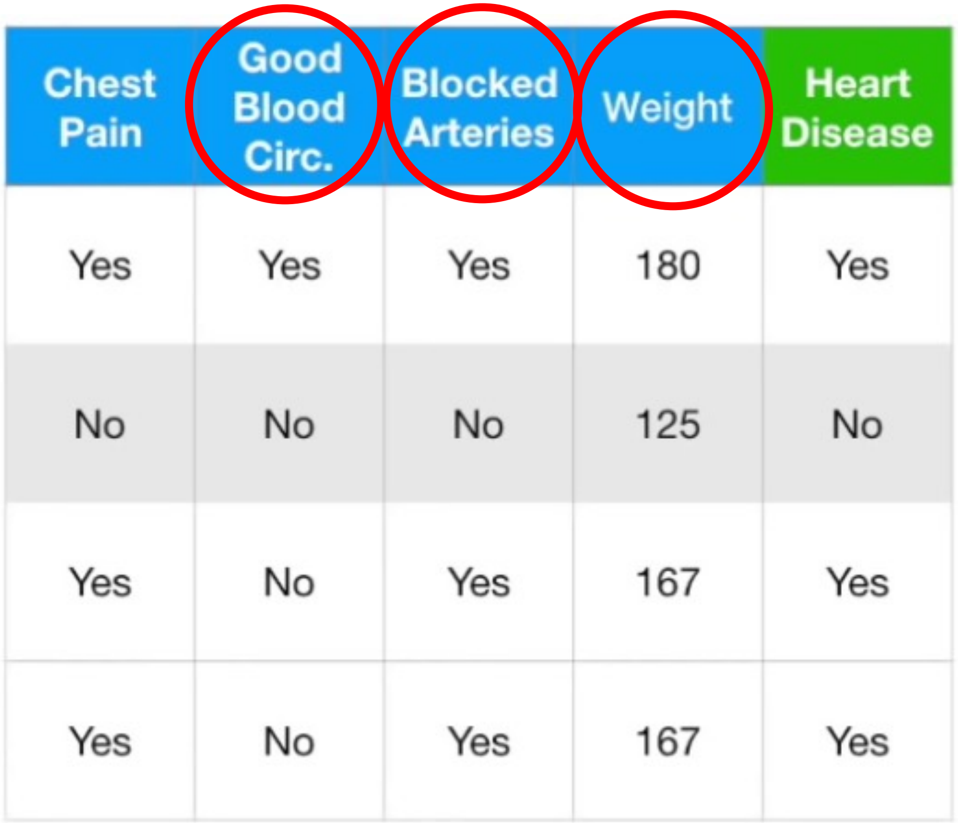
1. Create the Out-of-Bag Dataset
2. Evaluate the misclassification error on the tree by using the Out-of-Bag Dataset

- Return the proportion of misclassified samples in the Out-of-Bag Datasets overall

Better Random Forests

Select a subset of the features and start the procedure to build the corresponding tree **and the corresponding forest**



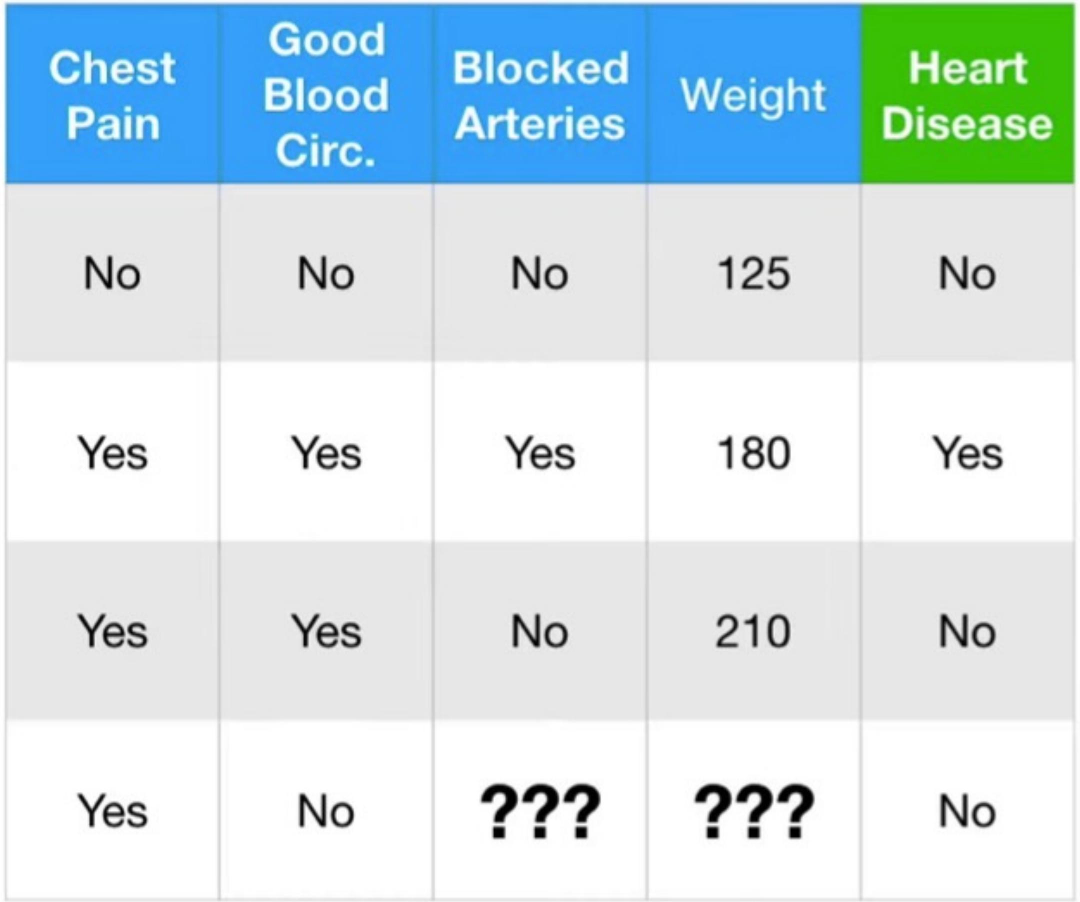


Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	Yes	180	Yes
No	No	No	125	No
Yes	No	Yes	167	Yes
Yes	No	Yes	167	Yes

Missing values

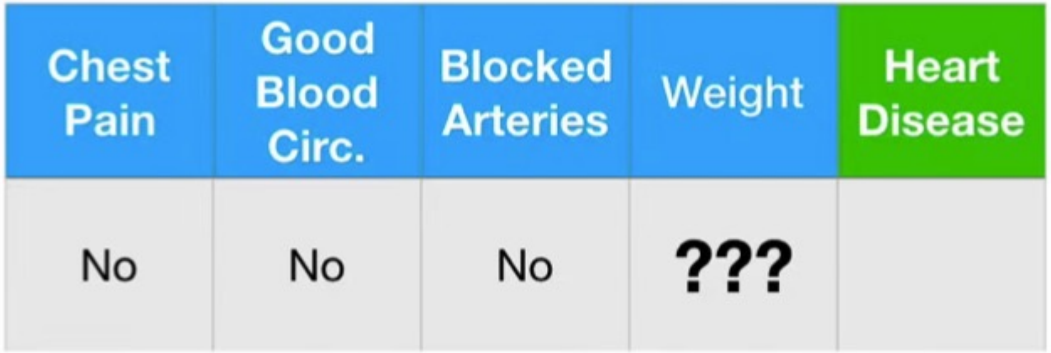
Missing Data

In the original dataset



Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	No	???	???	No

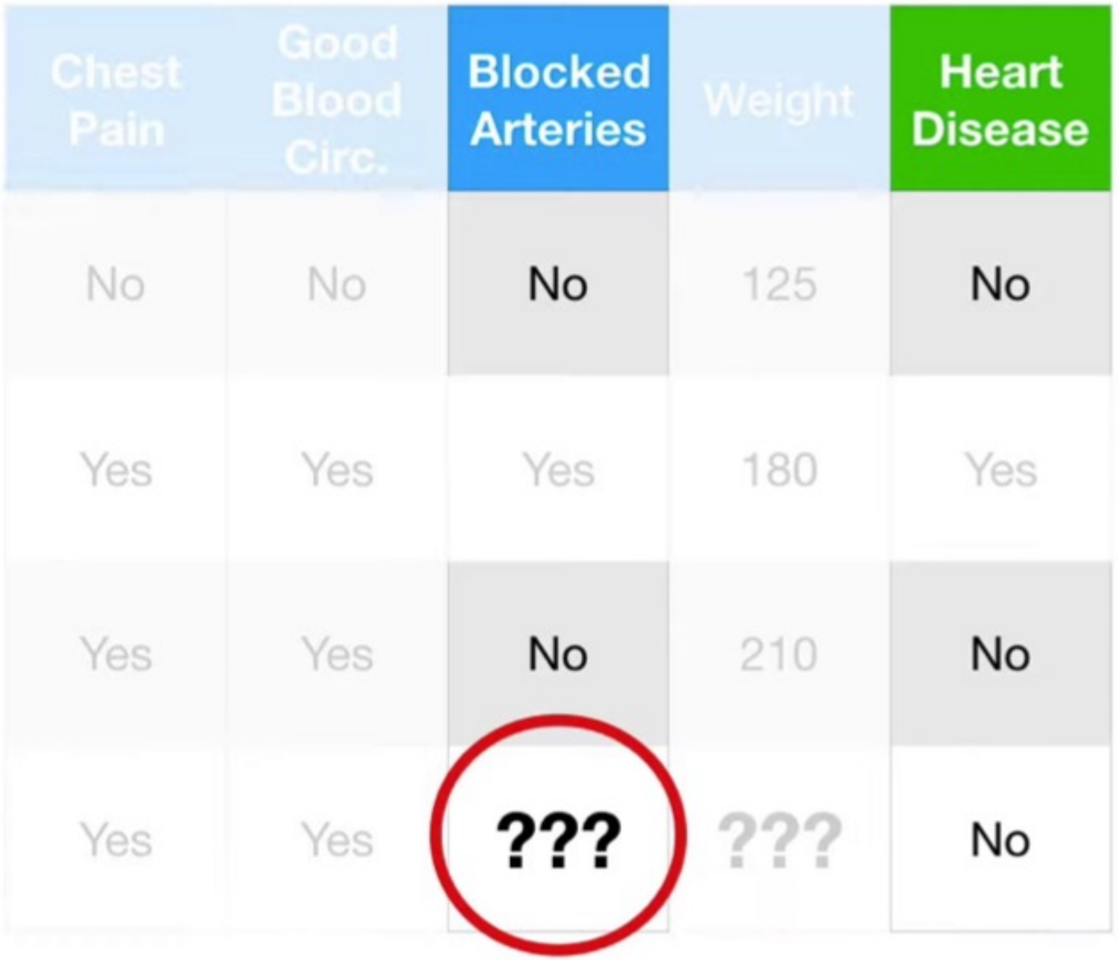
In the new sample



Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	???	

Missing values in the dataset

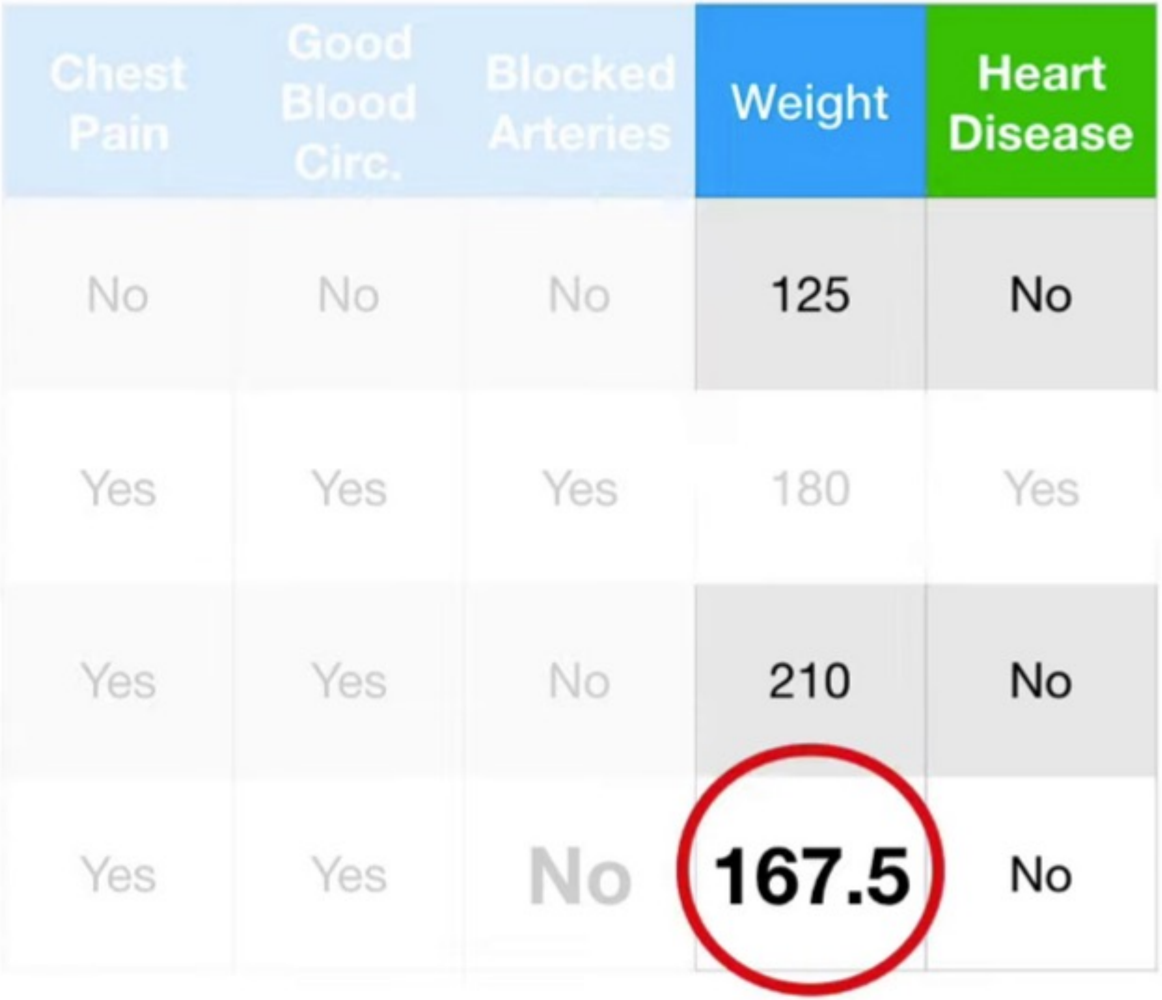
Initial guess - Categorical



Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	Yes	???	???	No

Consider the most common value found in the other samples that have NO as value for Heart Disease

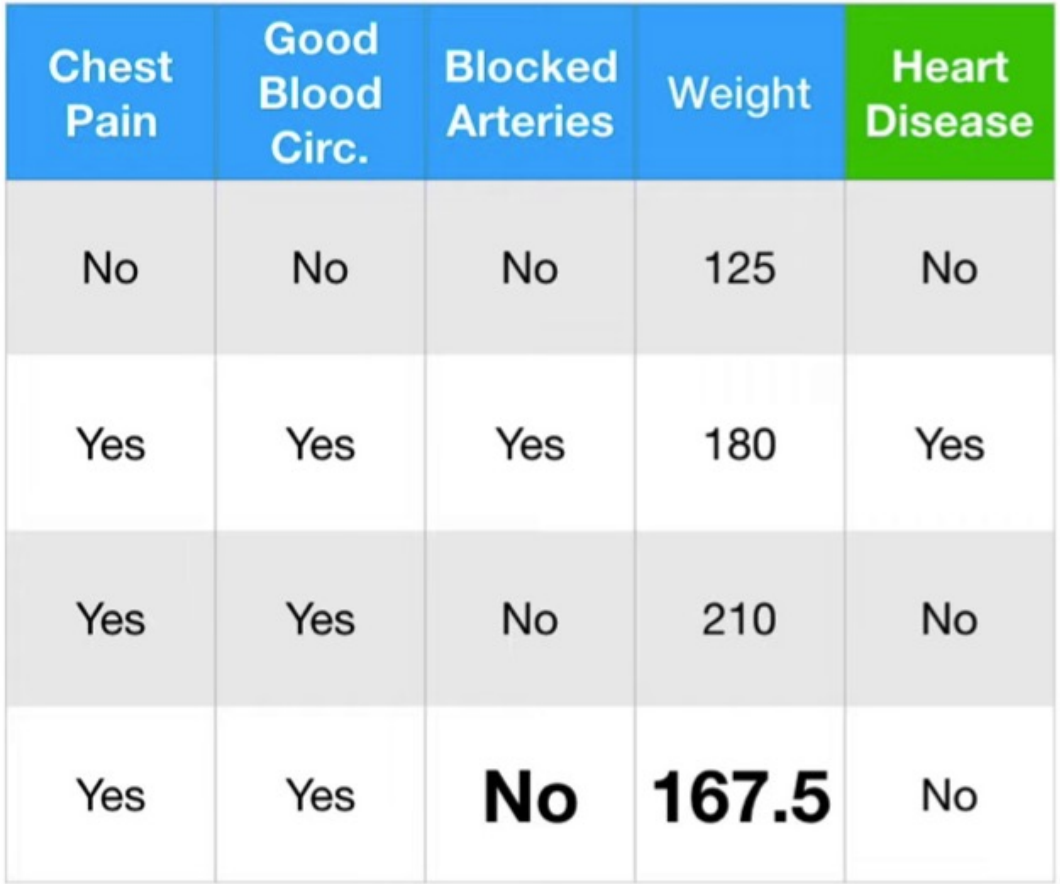
Initial Guess - Numerical



Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	Yes	No	167.5	No

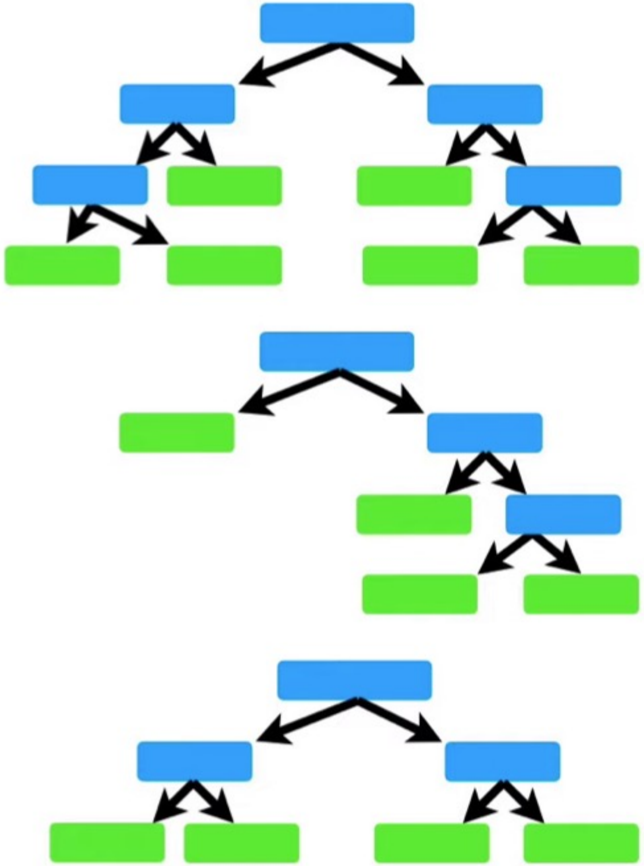
Consider the median value among the ones found in the other samples that have NO as value for Heart Disease

Value refinement



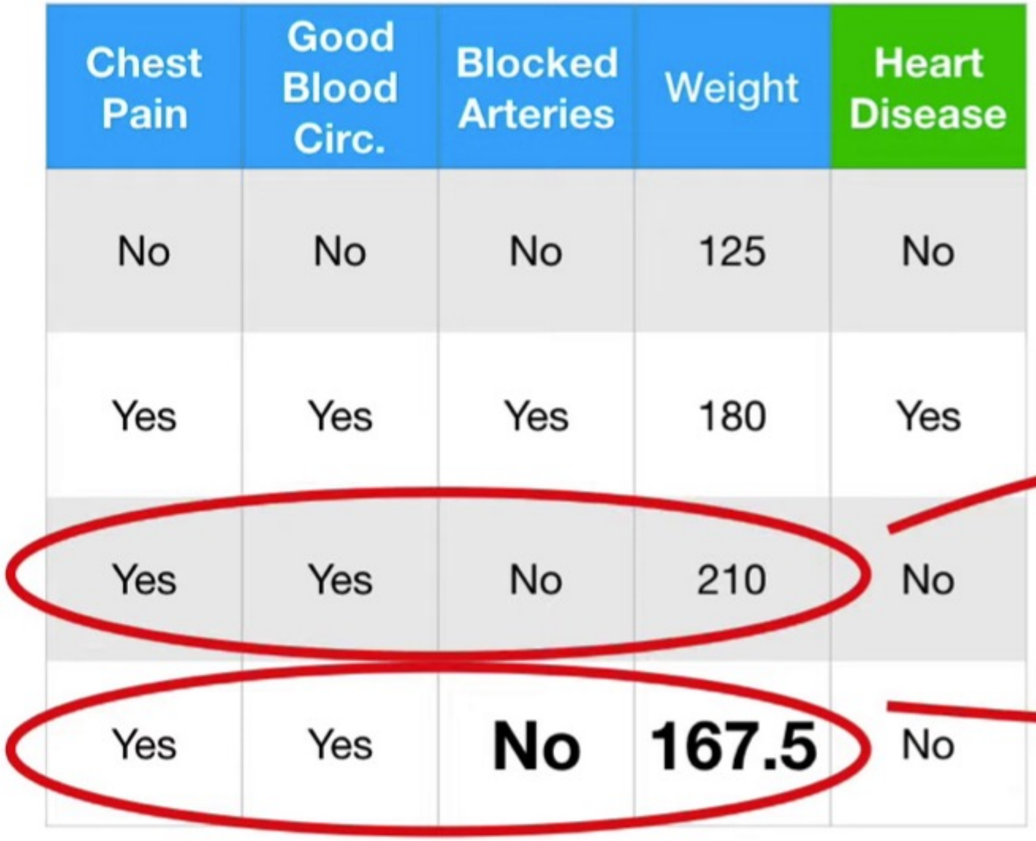
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	Yes	**No**	**167.5**	No

1. Build a Random Forest

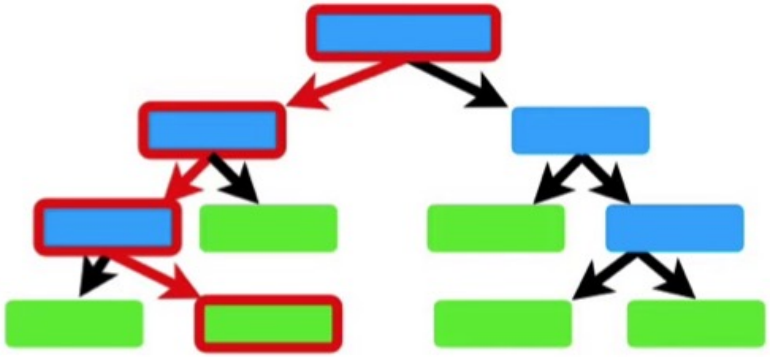


Value refinement

2. Run the data through the forest by considering each tree



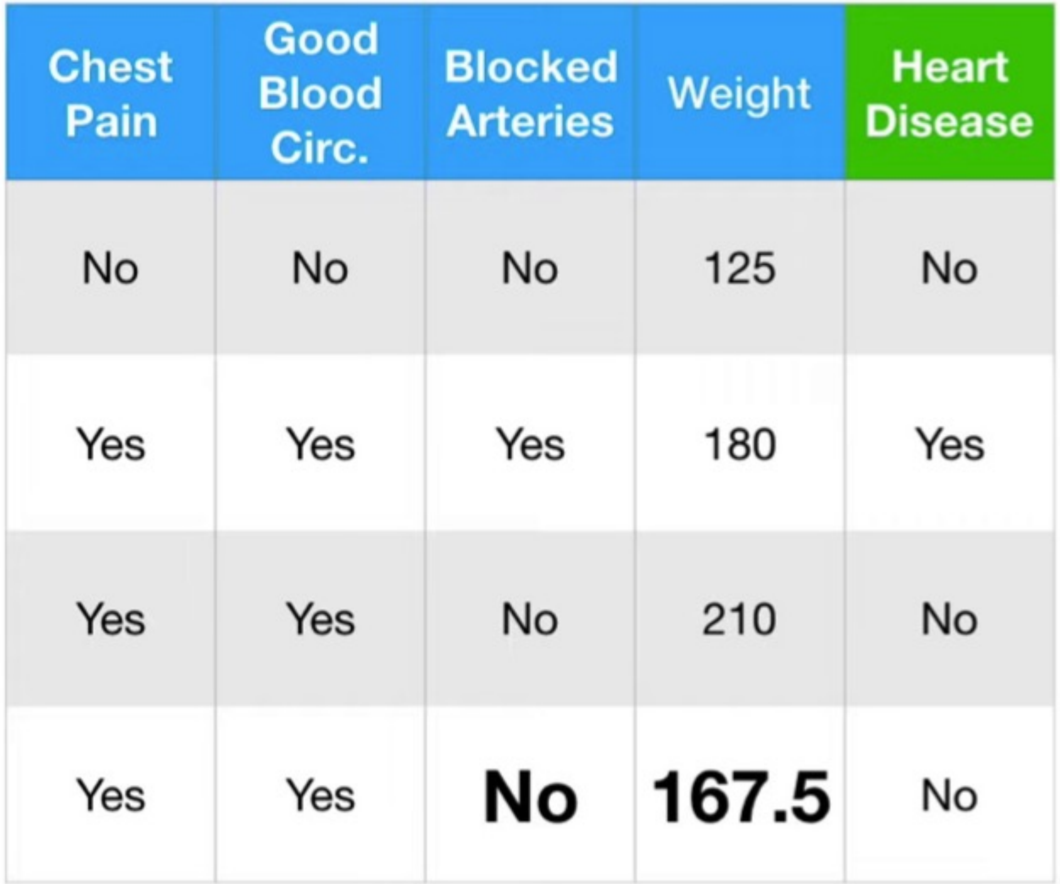
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	Yes	**No**	**167.5**	No



Sample 3 and 4 end up at the same leaf node.

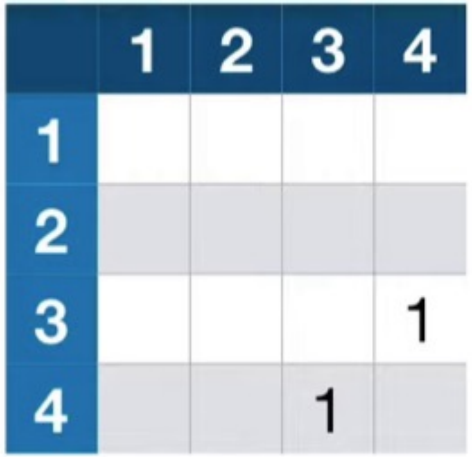
This seems they are similar.

Value refinement



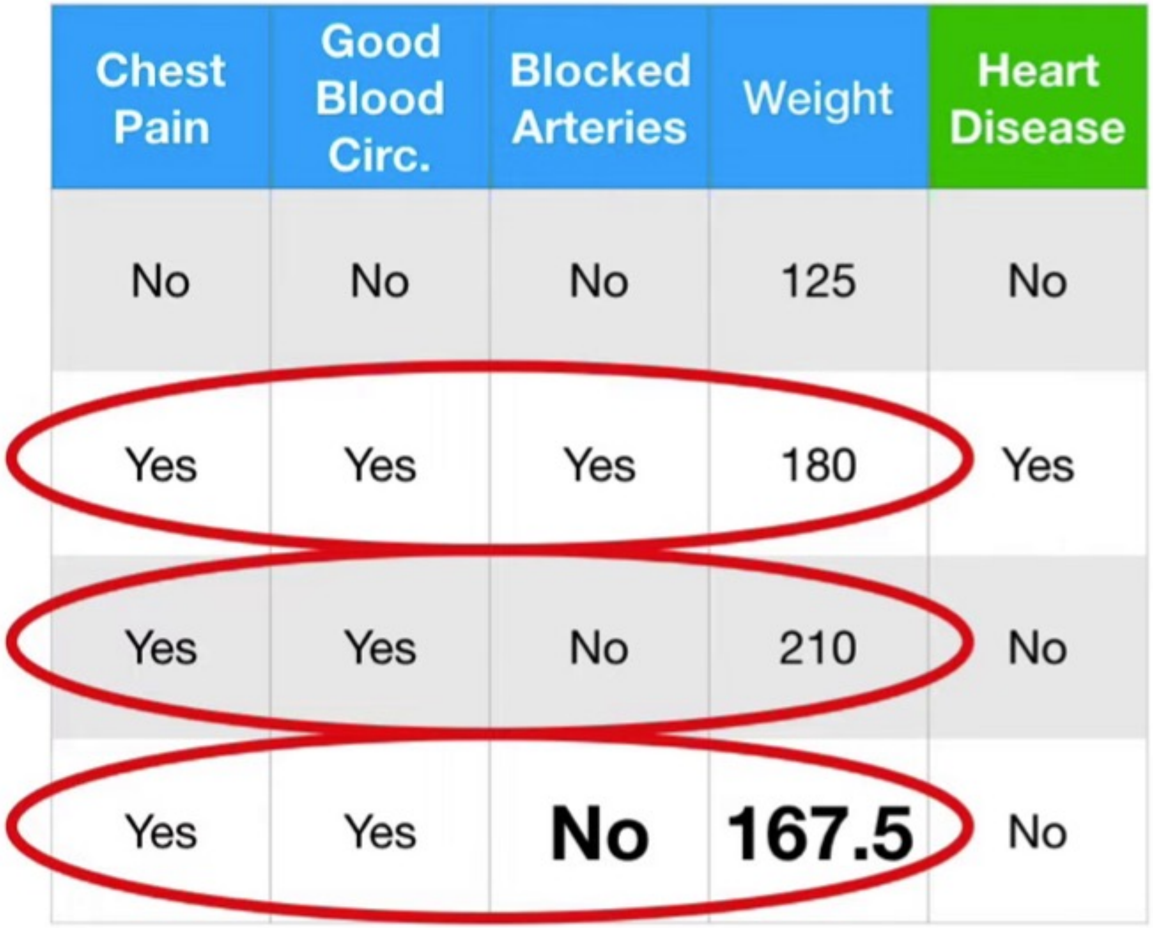
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	Yes	**No**	**167.5**	No

Keep track of the similarity via a proximity matrix

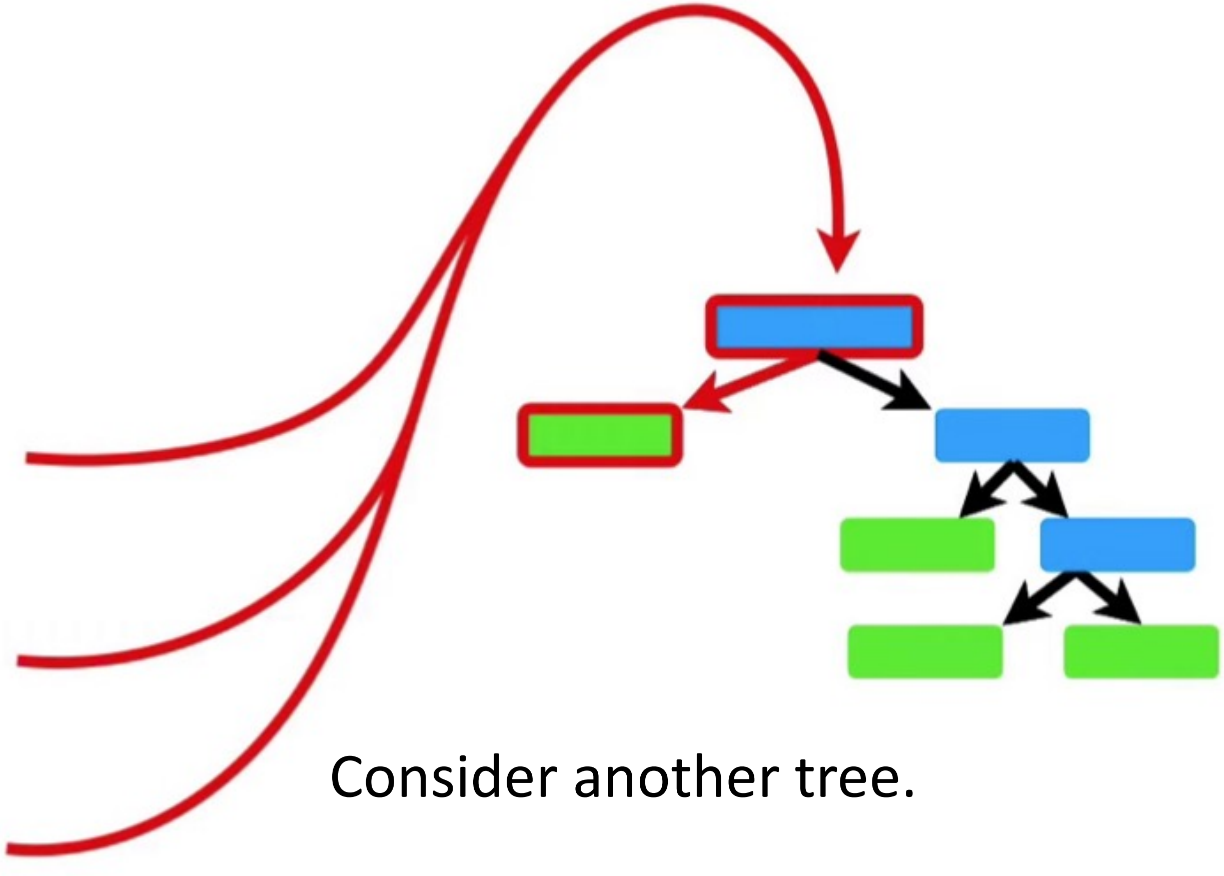


	1	2	3	4
1				
2				
3				1
4			1	

Value refinement



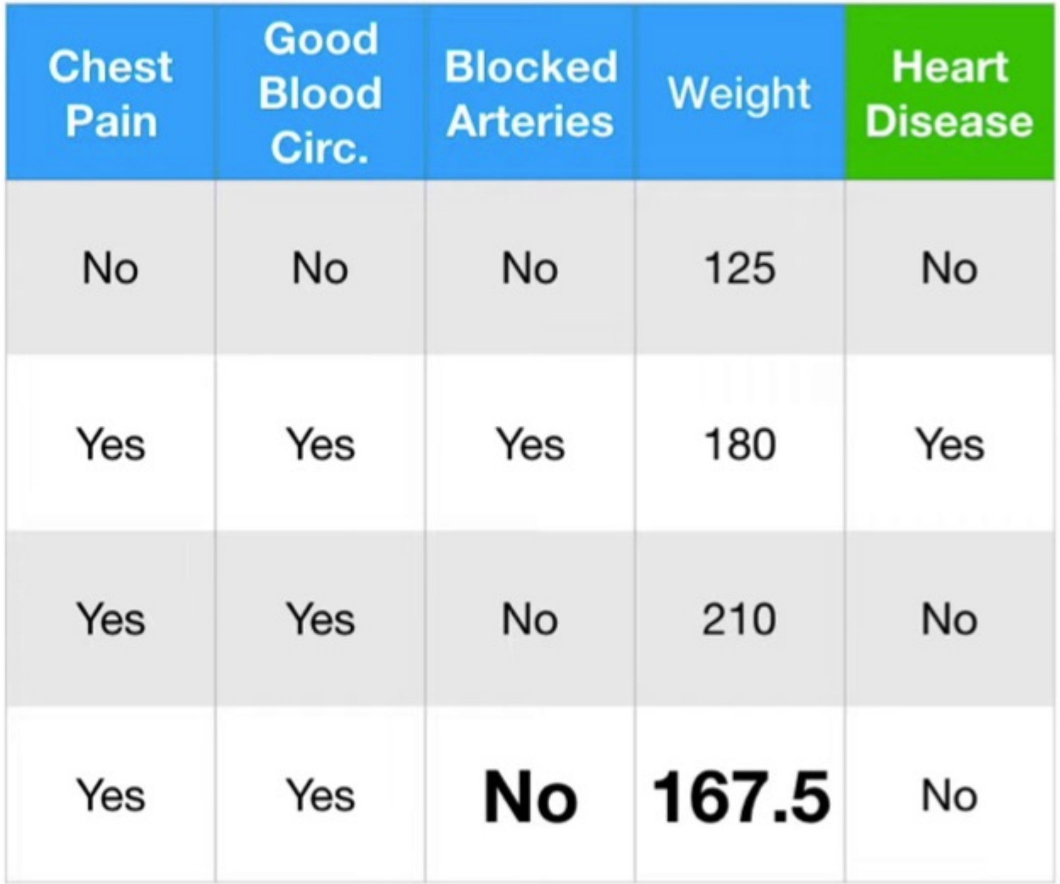
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	Yes	**No**	**167.5**	No



Consider another tree.

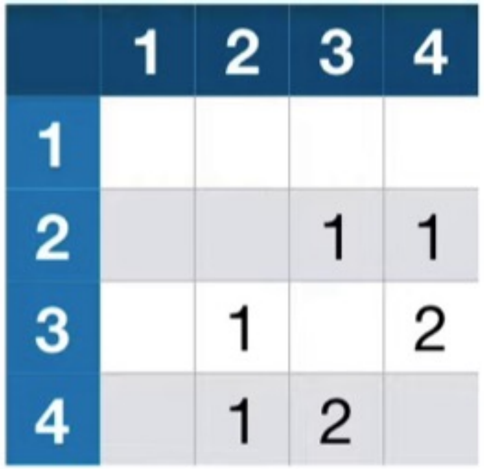
Sample 2, 3 and 4 end up at the same leaf node.

Value refinement



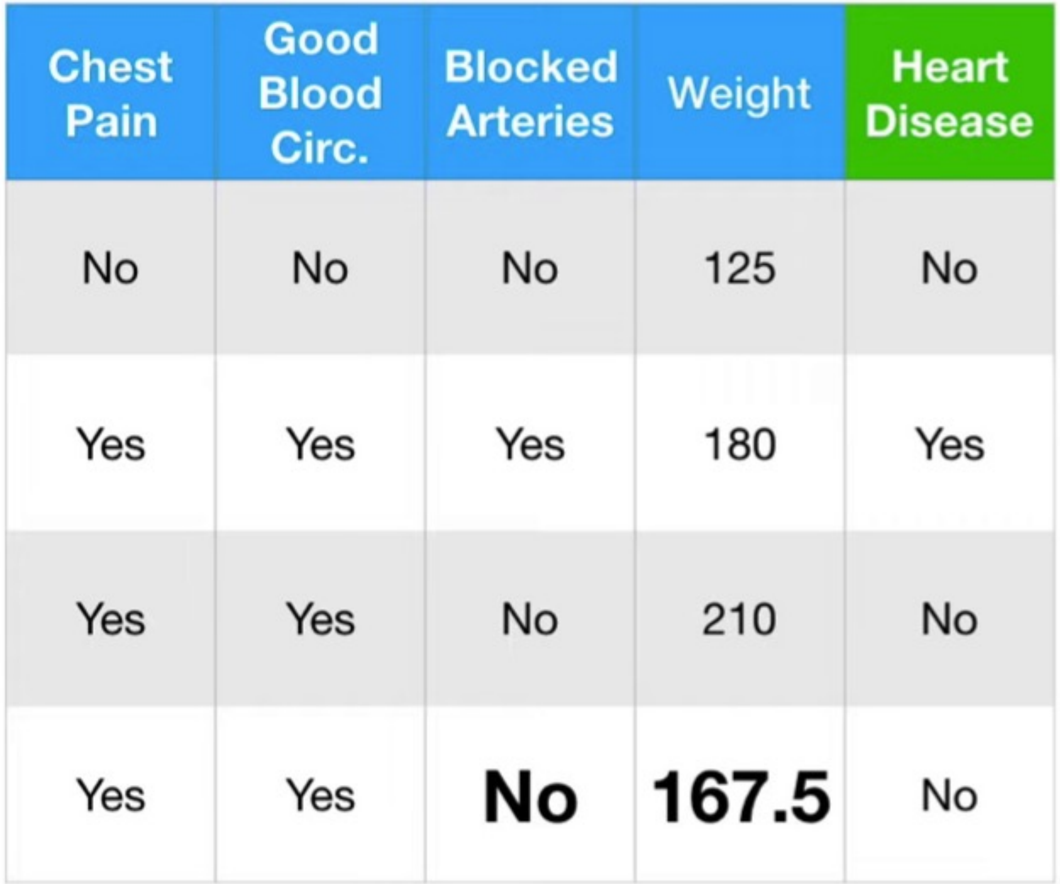
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	Yes	**No**	**167.5**	No

Update the proximity matrix



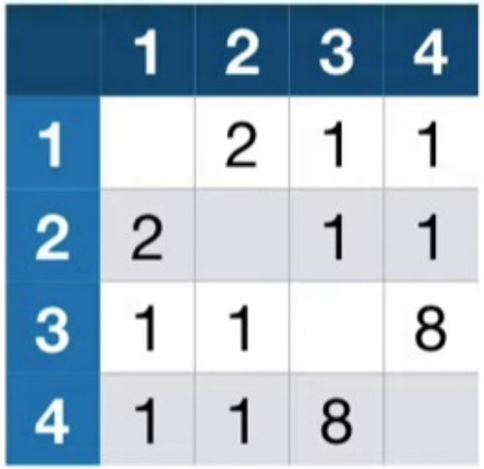
	1	2	3	4
1				
2			1	1
3		1		2
4		1	2	

Value refinement

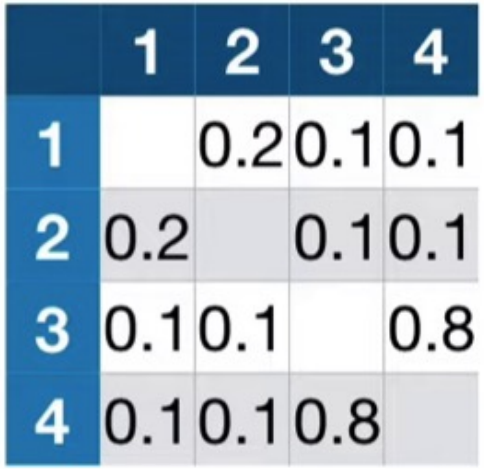


Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	Yes	**No**	**167.5**	No

Update the proximity matrix with respect to all the trees and then normalize with respect to the total number of trees

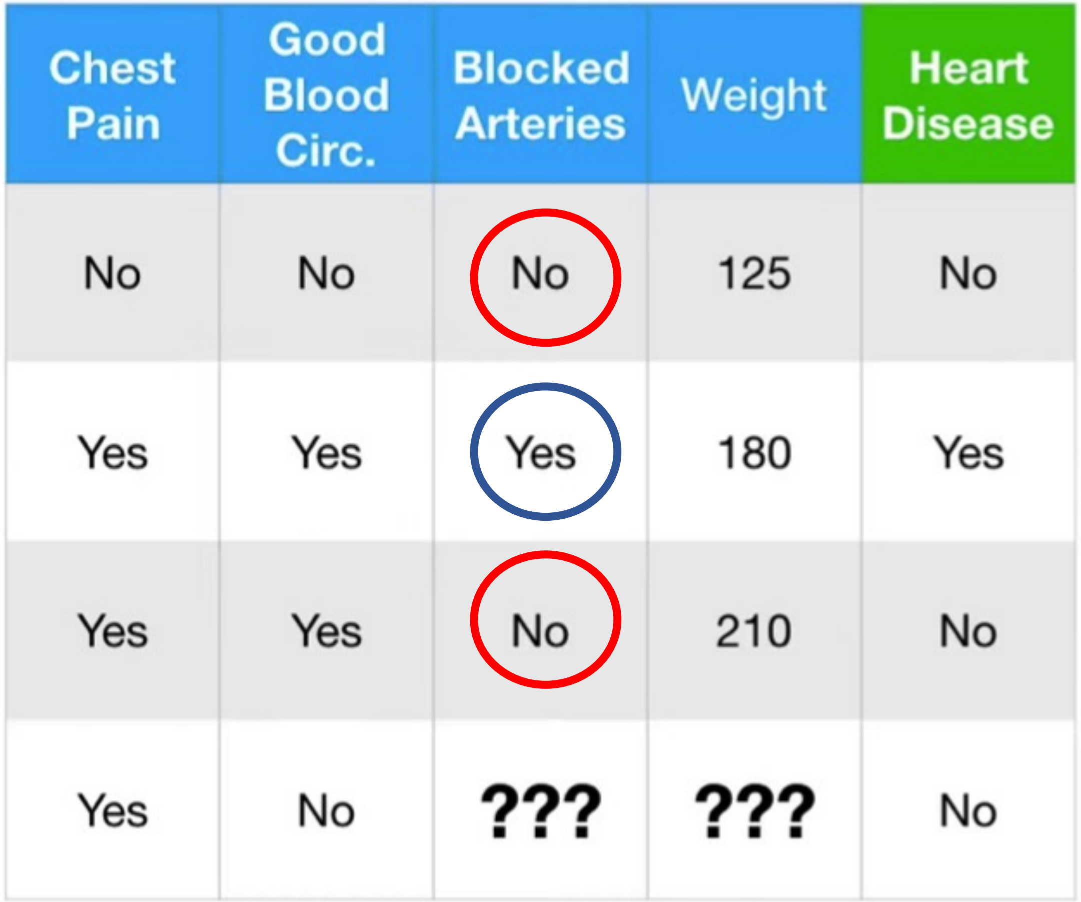


	1	2	3	4
1		2	1	1
2	2		1	1
3	1	1		8
4	1	1	8	



	1	2	3	4
1		0.2	0.1	0.1
2	0.2		0.1	0.1
3	0.1	0.1		0.8
4	0.1	0.1	0.8	

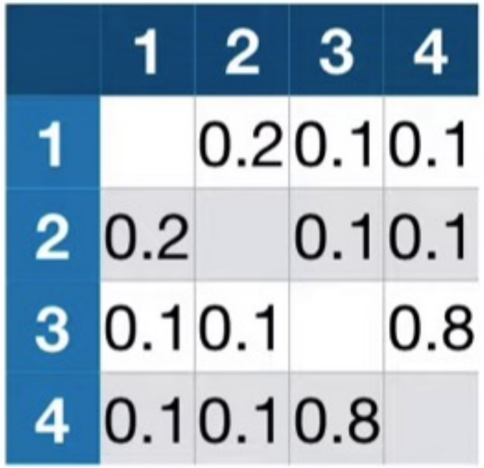
Value refinement



Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	No	???	???	No

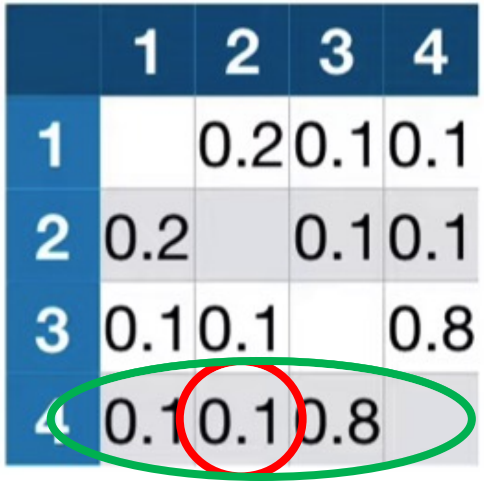
Use the proximity matrix to compute the value

Compute the frequency
Yes = 1/3
No = 2/3

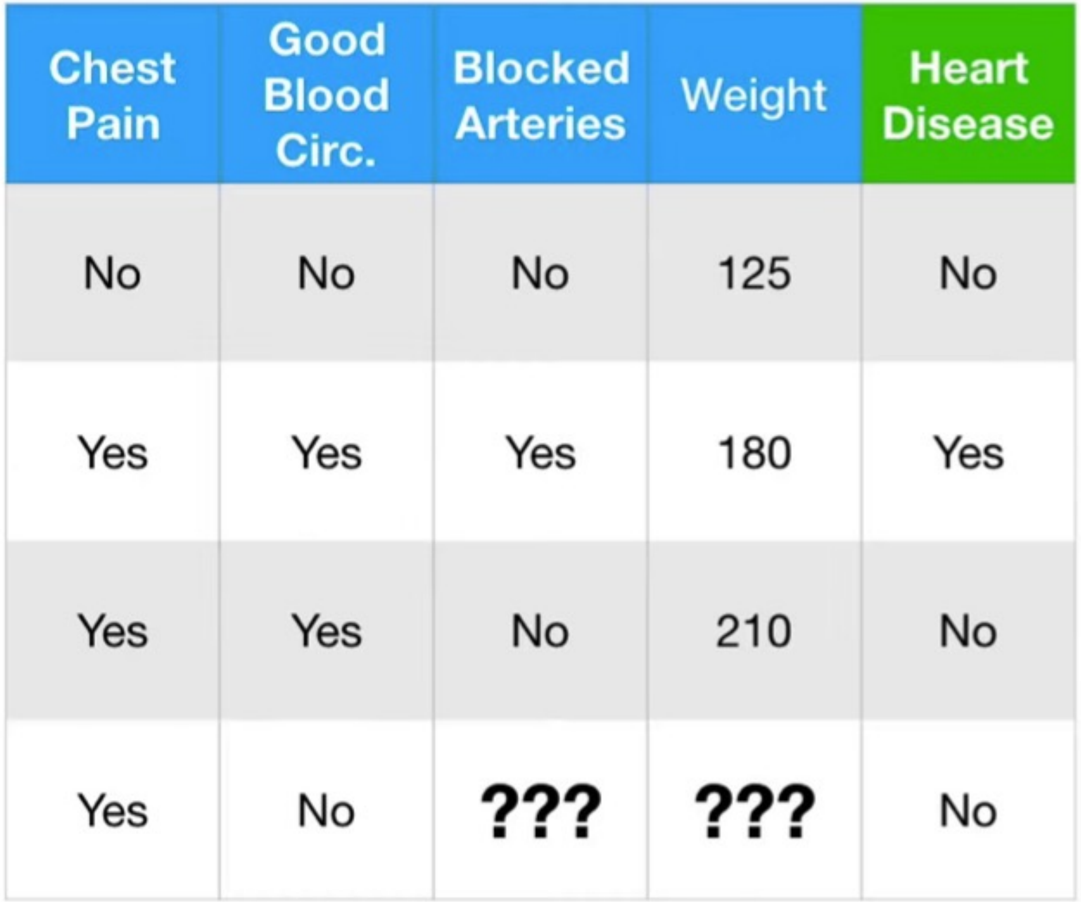


	1	2	3	4
1		0.2	0.1	0.1
2	0.2		0.1	0.1
3	0.1	0.1		0.8
4	0.1	0.1	0.8	

Value refinement



	1	2	3	4
1		0.2	0.1	0.1
2	0.2		0.1	0.1
3	0.1	0.1		0.8
4	0.1	0.1	0.8	

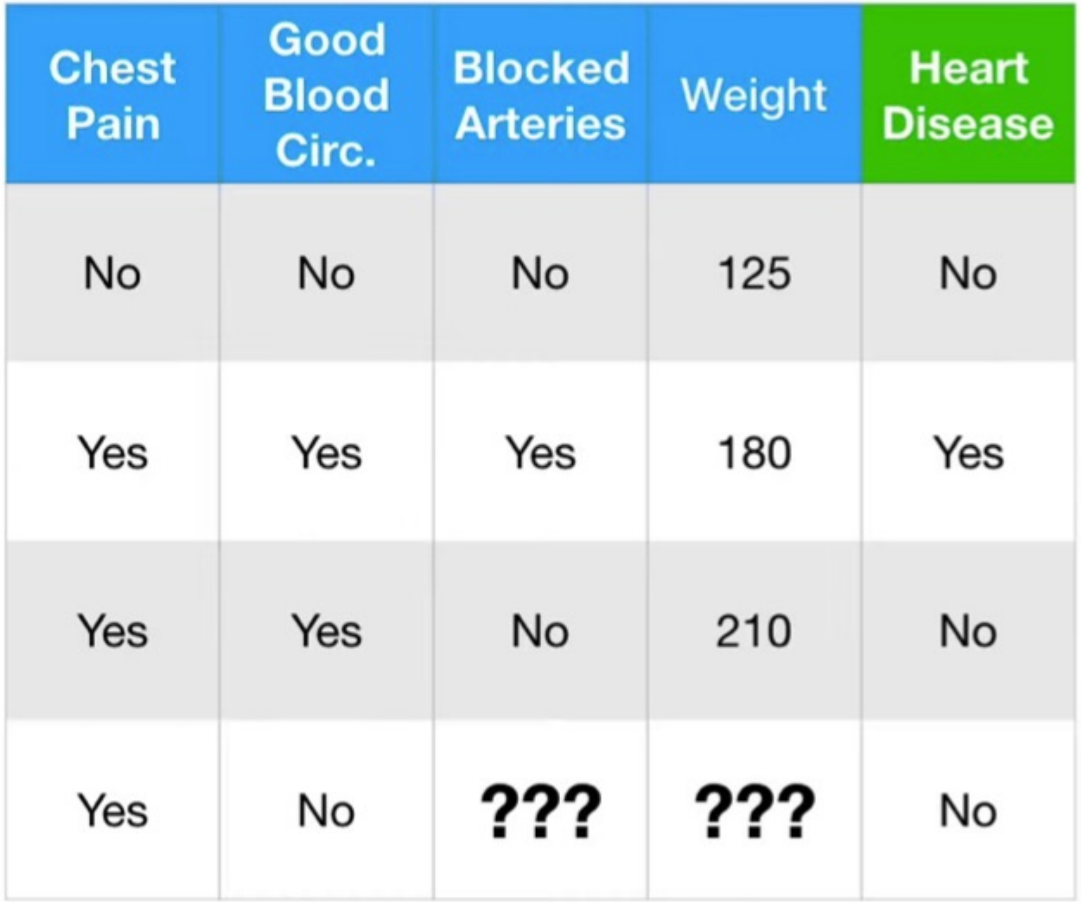


Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	No	???	???	No

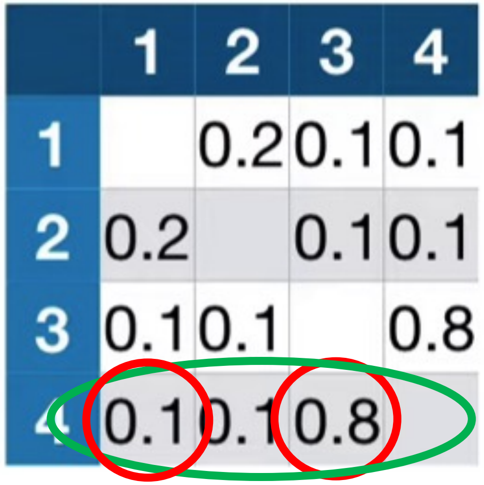
$$\text{Yes} = \frac{1}{3} \cdot proximity^{YES}$$

$$= \frac{1}{3} \cdot \frac{0.1}{01. + 0.1 + 0.8} = 0.03$$

Value refinement



Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	No	???	???	No

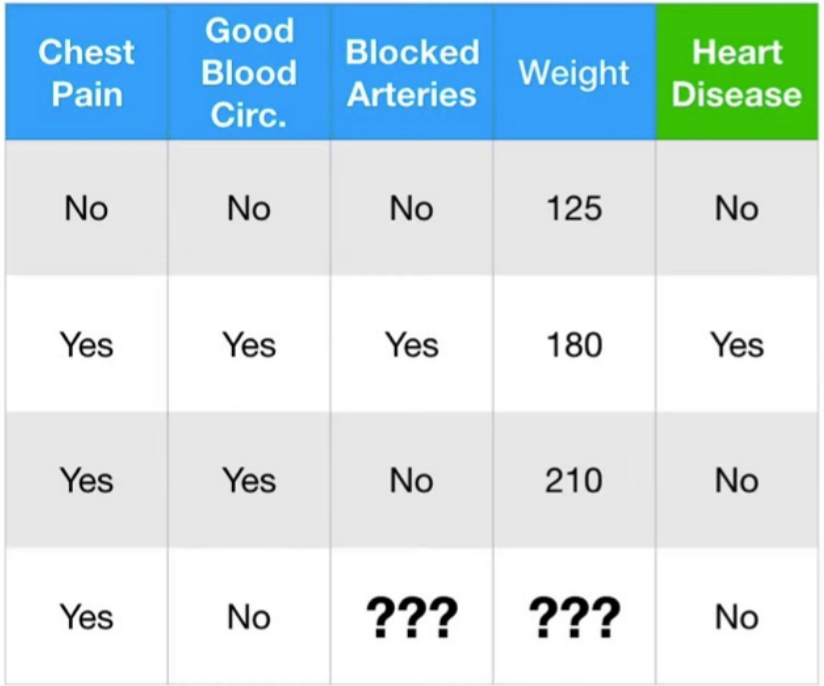


	1	2	3	4
1		0.2	0.1	0.1
2	0.2		0.1	0.1
3	0.1	0.1		0.8
4	0.1	0.1	0.8	

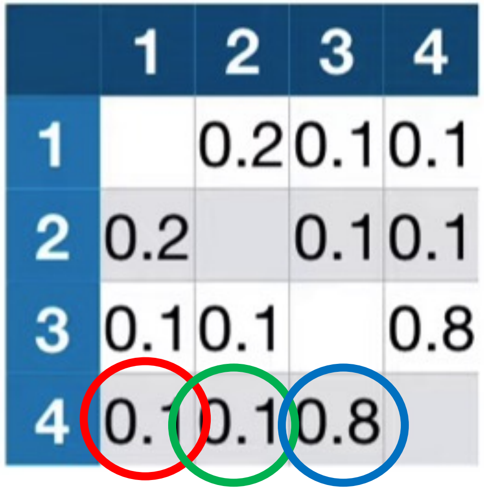
$$\text{No} = \frac{2}{3}\cdot proximity^{YES}$$

$$= \frac{2}{3}\cdot\frac{0.1+0.8}{01.+0.1+0.8} = 0.6$$

Value refinement



Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	No	???	???	No

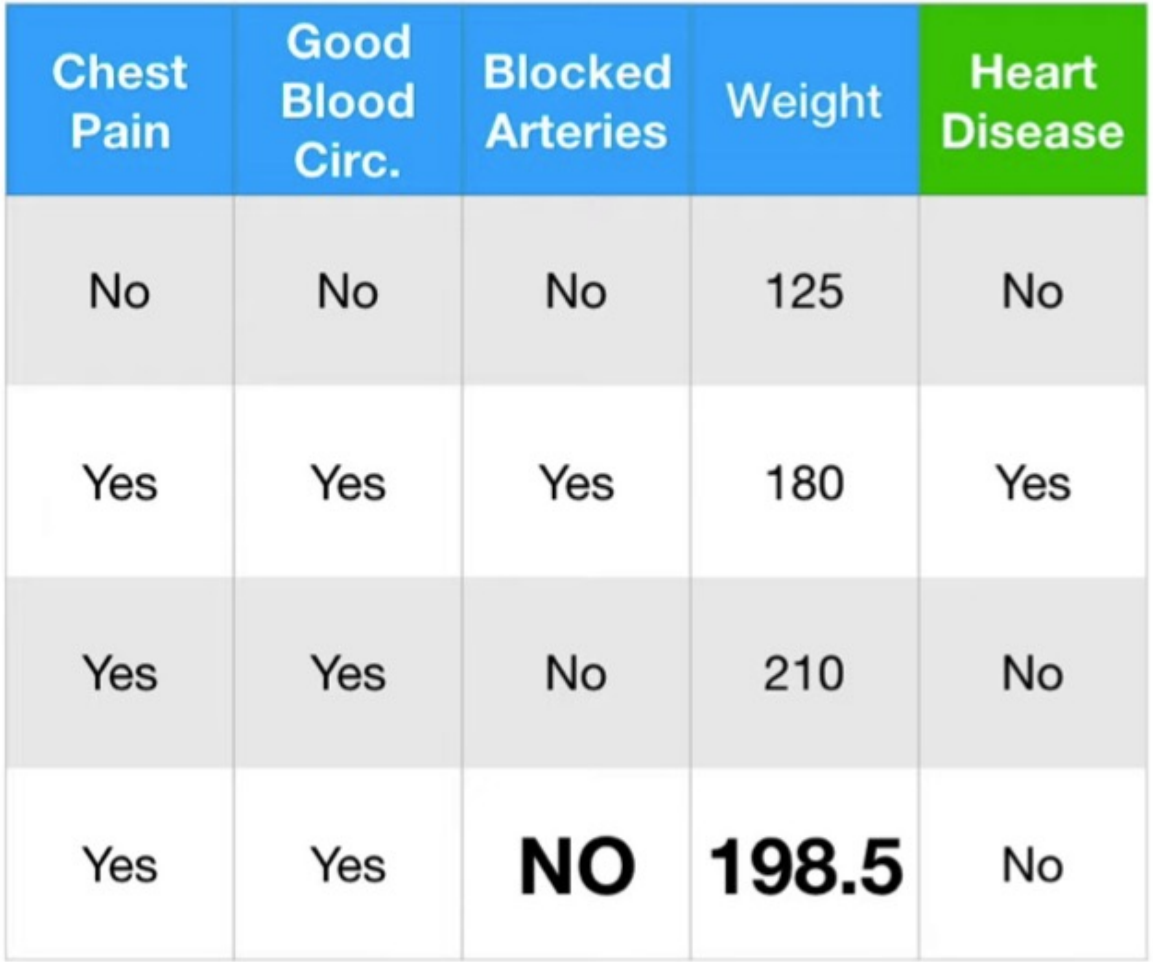


	1	2	3	4
1		0.2	0.1	0.1
2	0.2		0.1	0.1
3	0.1	0.1		0.8
4	0.1	0.1	0.8	

Compute the weighted average

$$w = 125 \cdot \frac{0.1}{0.1 + 0.1 + 0.8} + 180 \cdot \frac{0.1}{0.1 + 0.1 + 0.8} + 210 \cdot \frac{0.8}{0.1 + 0.1 + 0.8} = 198.5$$

Value refinement



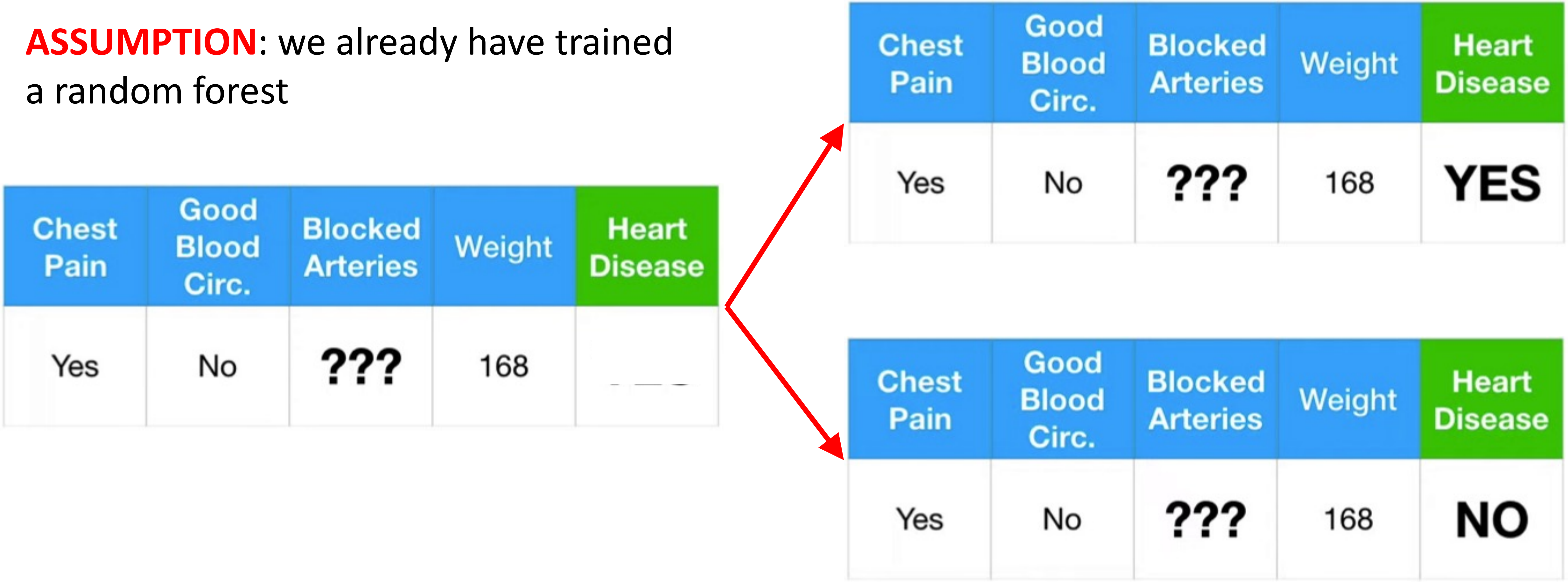
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	Yes	**NO**	**198.5**	No

- Build a new Random Forest
- Repeat all the previous steps until until the missing values converge

Missing values in the new samples

Create two copies of the sample with missing data

ASSUMPTION: we already have trained a random forest

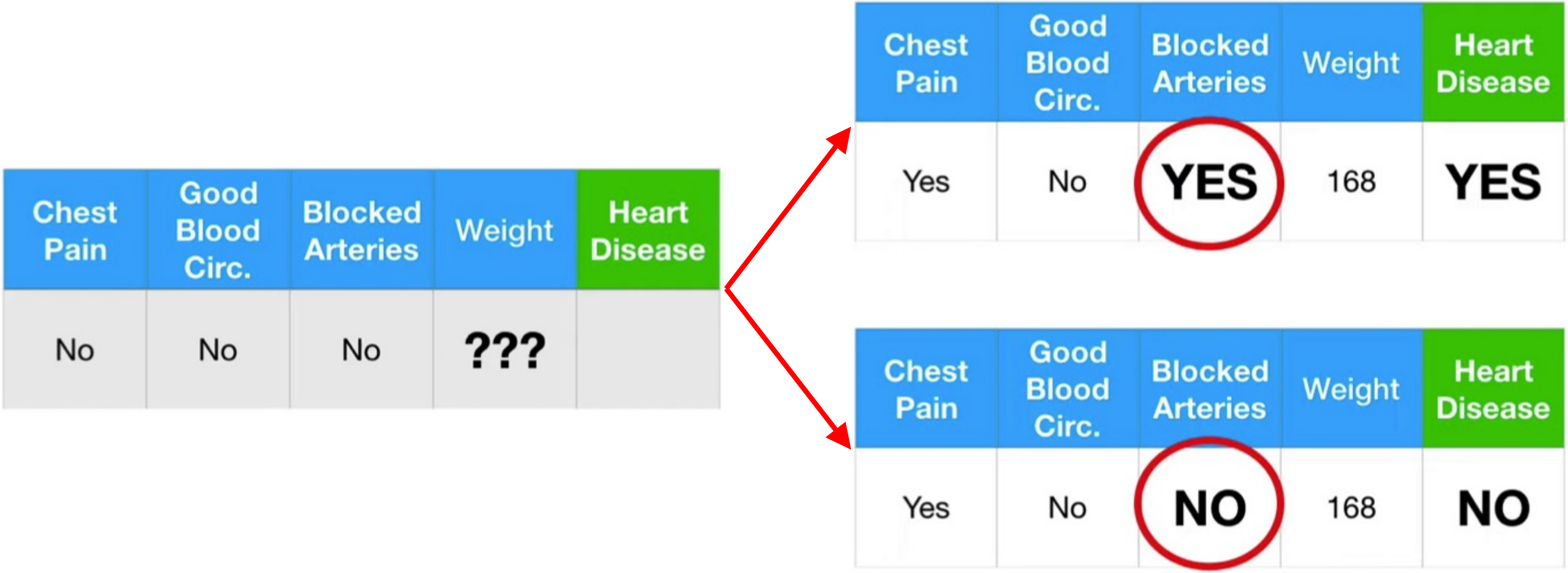


Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	No	???	168	

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	No	???	168	YES

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	No	???	168	NO

Use the method for missing values in the dataset to guess a value

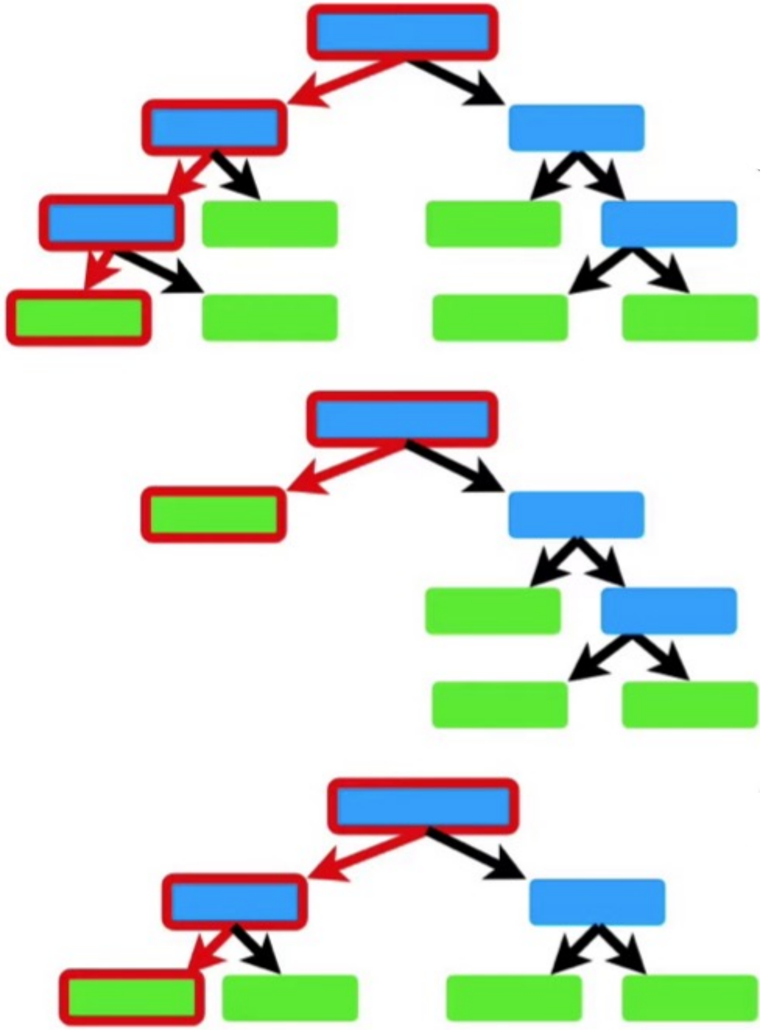


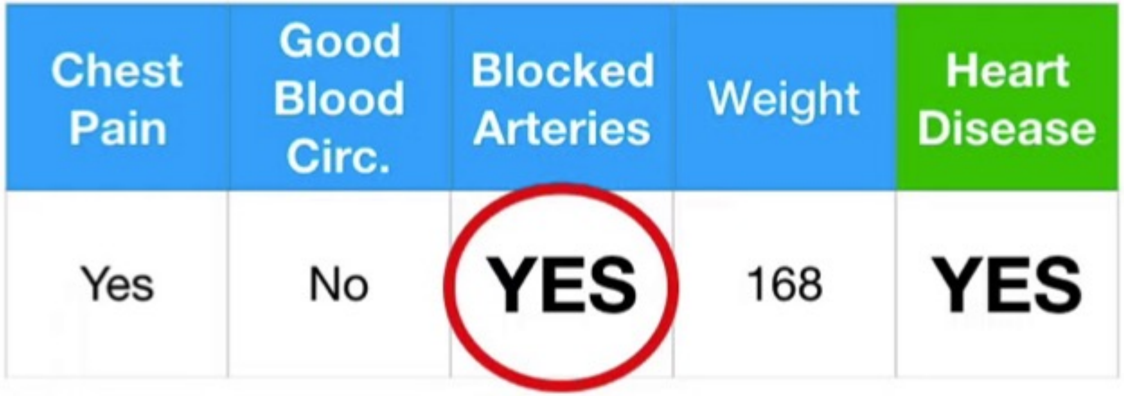
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	???	

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	No	YES	168	YES

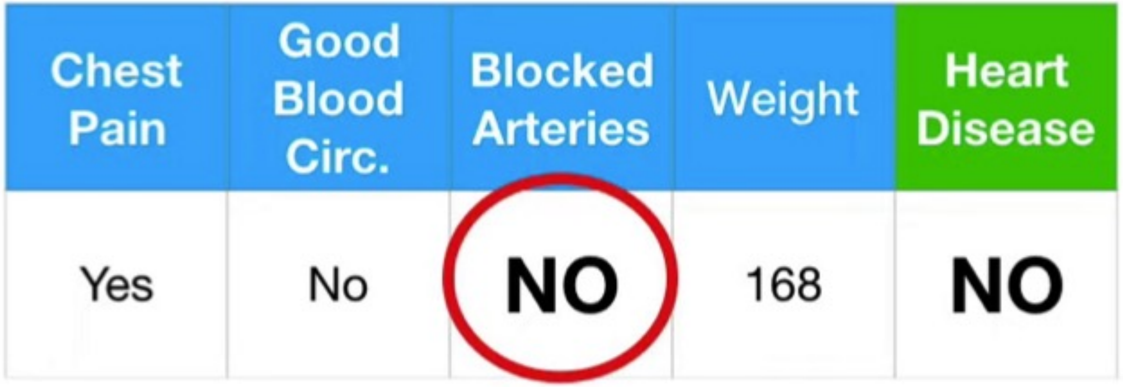
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	No	NO	168	NO

Run the two samples in the random forest



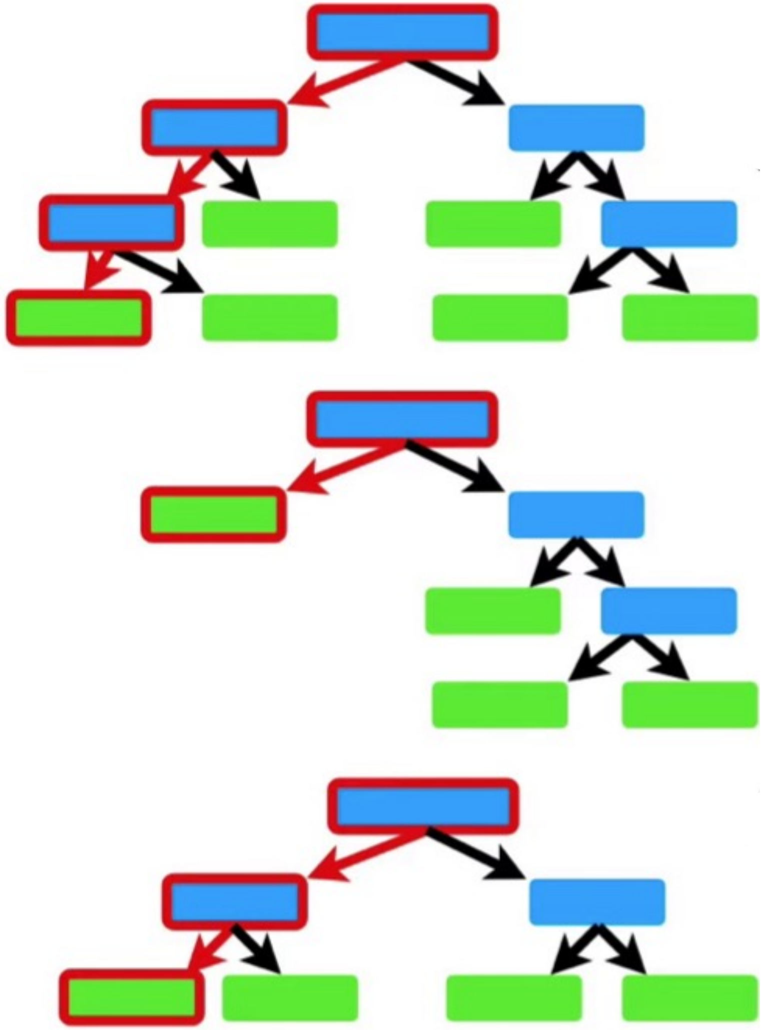


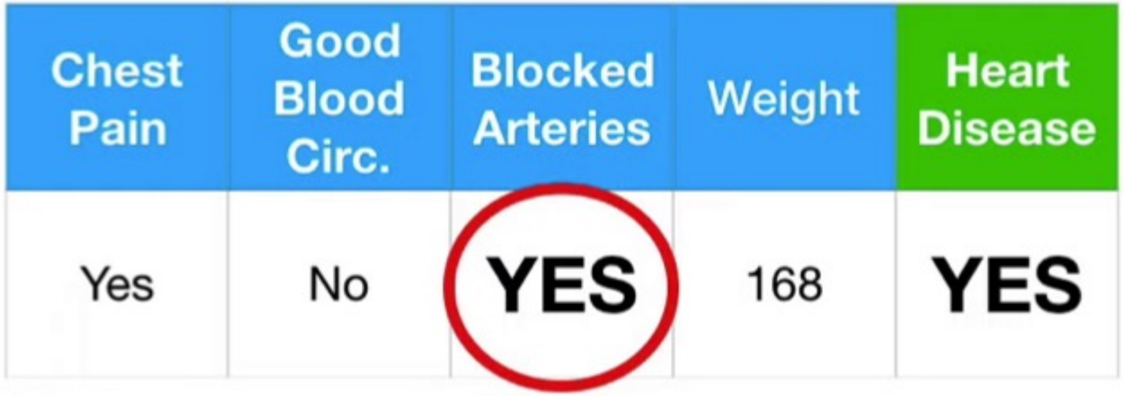
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	No	YES	168	YES



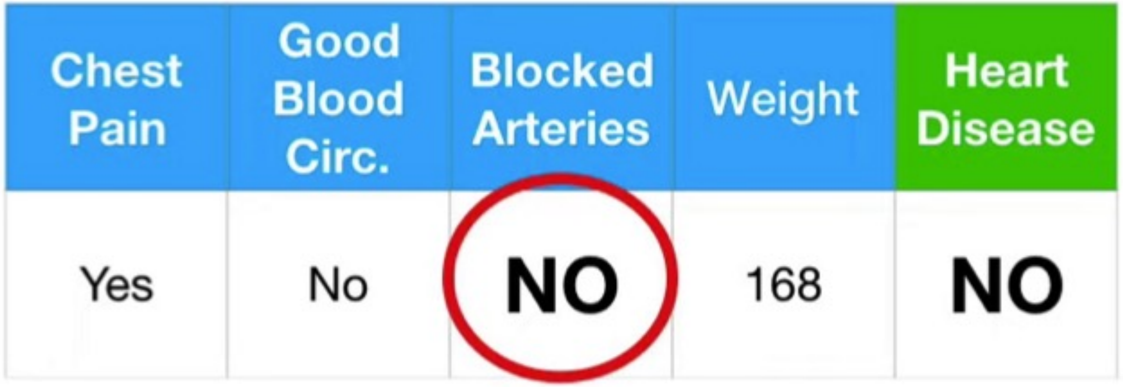
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	No	NO	168	NO

Count all the times the two samples are correctly labelled





Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	No	YES	168	YES



Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	No	NO	168	NO