Boosting

A classic boosting method is adaboost, here we use it as an example to explain what is boosting

A classic boosting method is adaboost, here we use it as an example to explain what is boosting

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease
Yes	Yes	205	Yes
No	Yes	180	Yes
Yes	No	210	Yes
Yes	Yes	167	Yes
No	Yes	156	No
No	Yes	125	No
Yes	No	168	No
Yes	Yes	172	No

Assuming we have the above dataset

A classic boosting method is adaboost, here we use it as an example to explain what is boosting

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

Assuming we have the above dataset

Step 1: we give weights for each sample (e.g., as an initial guess, it can be $\frac{1}{total\ samples}$)

A classic boosting method is adaboost, here we use it as an example to explain what is boosting

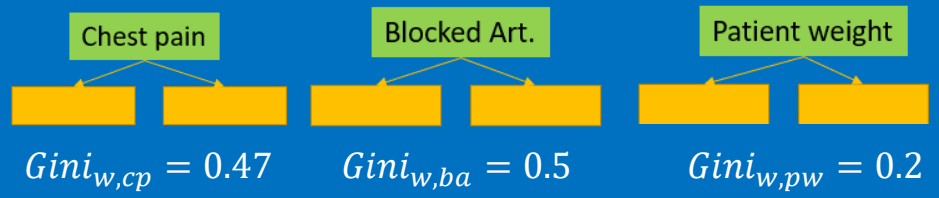
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

Assuming we have the above dataset

Step 1: we give weights for each sample (e.g., as an initial guess, it can be $\frac{1}{total\ samples}$)

Step 2: determine the "root" tree

we go through each feature, grow trees and calculate "weighted Gini index"



$Gini_{w,cp} = 0.47$ $Gini_{w,ba} = 0.5$ $Gini_{w,pw} = 0.2$

A classic boosting method is adaboost, here we use it as an example to explain what is boosting

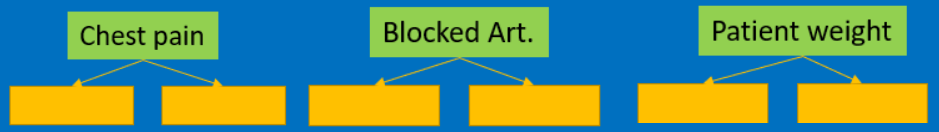
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

Assuming we have the above dataset

Step 1: we give weights for each sample (e.g., as an initial guess, it can be $\frac{1}{total\ samples}$)

Step 2: determine the "root" tree

we go through each feature, grow trees and calculate "weighted Gini index"



$Gini_{w,cp} = 0.47$ $Gini_{w,ba} = 0.5$ $Gini_{w,pw} = 0.2$

A classic boosting method is adaboost, here we use it as an example to explain what is boosting

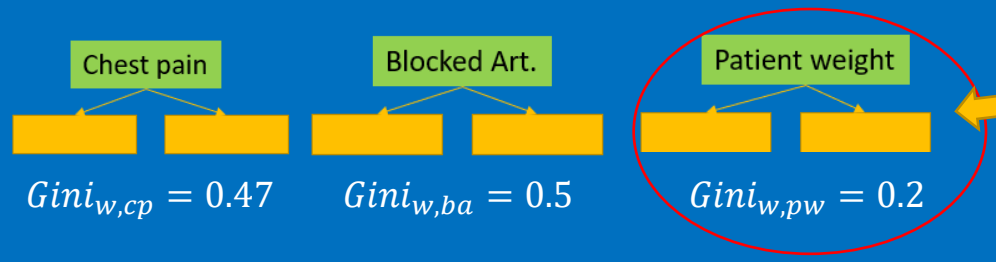
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

Assuming we have the above dataset

Step 1: we give weights for each sample (e.g., as an initial guess, it can be $\frac{1}{total\ samples}$)

Step 2: determine the "root" tree

we go through each feature, grow trees and calculate "weighted Gini index"



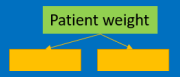
Patient weight has the smallest Gini, therefore it will be used as the "root/first tree"

A classic boosting method is adaboost, here we use it as an example to explain what is boosting

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

Assuming we have the above dataset

Step 1: we give weights for each sample (e.g., as an initial guess, it can be $\frac{1}{total\ samples}$)

Step 2: determine the “root” tree 

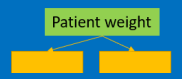
Step 3: determine how much the above tree will contribute to the final results

A classic boosting method is adaboost, here we use it as an example to explain what is boosting

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

Assuming we have the above dataset

Step 1: we give weights for each sample (e.g., as an initial guess, it can be $\frac{1}{total\ samples}$)

Step 2: determine the "root" tree 

Step 3: determine how much the above tree will contribute to the final results

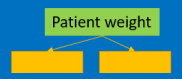
- Calculating the weighted prediction error for the tree, which is 0.125 (see Adaboost tutorial for the details)

A classic boosting method is adaboost, here we use it as an example to explain what is boosting

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

Assuming we have the above dataset

Step 1: we give weights for each sample (e.g., as an initial guess, it can be $\frac{1}{total\ samples}$)

Step 2: determine the "root" tree 

Step 3: determine how much the above tree will contribute to the final results

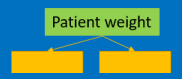
- Calculating the weighted prediction error for the tree, which is 0.125 (see Adaboost tutorial for the details)
- Calculating the "amount of say" for the tree, which is 0.97 (see Adaboost tutorial for the details)

A classic boosting method is adaboost, here we use it as an example to explain what is boosting

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

Assuming we have the above dataset

Step 1: we give weights for each sample (e.g., as an initial guess, it can be $\frac{1}{total\ samples}$)

Step 2: determine the "root" tree 

Step 3: determine how much the above tree will contribute to the final results ("amount of say")

- Calculating the weighted prediction error for the tree, which is 0.125 (see Adaboost tutorial for the details)
- Calculating the "amount of say" for the tree, which is 0.97 (see Adaboost tutorial for the details)

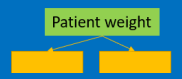
Step 4: Update the sample weight depending on the "amount of say"

A classic boosting method is adaboost, here we use it as an example to explain what is boosting

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	San wei	Sample weight
Yes	Yes	205	Yes	1/	0.05
No	Yes	180	Yes	1/	0.05
Yes	No	210	Yes	1/	0.05
Yes	Yes	167	Yes	1/	0.33
No	Yes	156	No	1/	0.05
No	Yes	125	No	1/	0.05
Yes	No	168	No	1/	0.05
Yes	Yes	172	No	1/	0.05

Assuming we have the above dataset

Step 1: we give weights for each sample (e.g., as an initial guess, it can be $\frac{1}{total\ samples}$)

Step 2: determine the "root" tree 

Step 3: determine how much the above tree will contribute to the final results ("amount of say")

- Calculating the weighted prediction error for the tree, which is 0.125 (see Adaboost tutorial for the details)
- Calculating the "amount of say" for the tree, which is 0.97 (see Adaboost tutorial for the details)

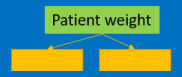
Step 4: Update the sample weight depending on the "amount of say"

A classic boosting method is adaboost, here we use it as an example to explain what is boosting

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sam wei	Sample weight
Yes	Yes	205	Yes	1/	0.05
No	Yes	180	Yes	1/	0.05
Yes	No	210	Yes	1/	0.05
Yes	Yes	167	Yes	1/	0.33
No	Yes	156	No	1/	0.05
No	Yes	125	No	1/	0.05
Yes	No	168	No	1/	0.05
Yes	Yes	172	No	1/	0.05

Assuming we have the above dataset

Step 1: we give weights for each sample (e.g., as an initial guess, it can be $\frac{1}{total\ samples}$)

Step 2: determine the "root" tree 

Step 3: determine how much the above tree will contribute to the final results ("amount of say")

- Calculating the weighted prediction error for the tree, which is 0.125 (see Adaboost tutorial for the details)
- Calculating the "amount of say" for the tree, which is 0.97 (see Adaboost tutorial for the details)

Step 4: Update the sample weight depending on the "amount of say"

Step 5: Go back to "step 2", using the updated weight, and repeat the processes between "step 2" and "step4"

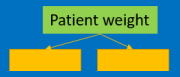
A classic boosting method is adaboost, here we use it as an example to explain what is boosting

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	0.05
No	Yes	180	Yes	0.05
Yes	No	210	Yes	0.05
Yes	Yes	167	Yes	0.33
No	Yes	156	No	0.05
No	Yes	125	No	0.05
Yes	No	168	No	0.05
Yes	Yes	172	No	0.05

Assuming we have the above dataset

Step 1: we give weights for each sample (e.g., as an initial guess, it can be $\frac{1}{total\ samples}$)

Step 2: determine the "root" tree

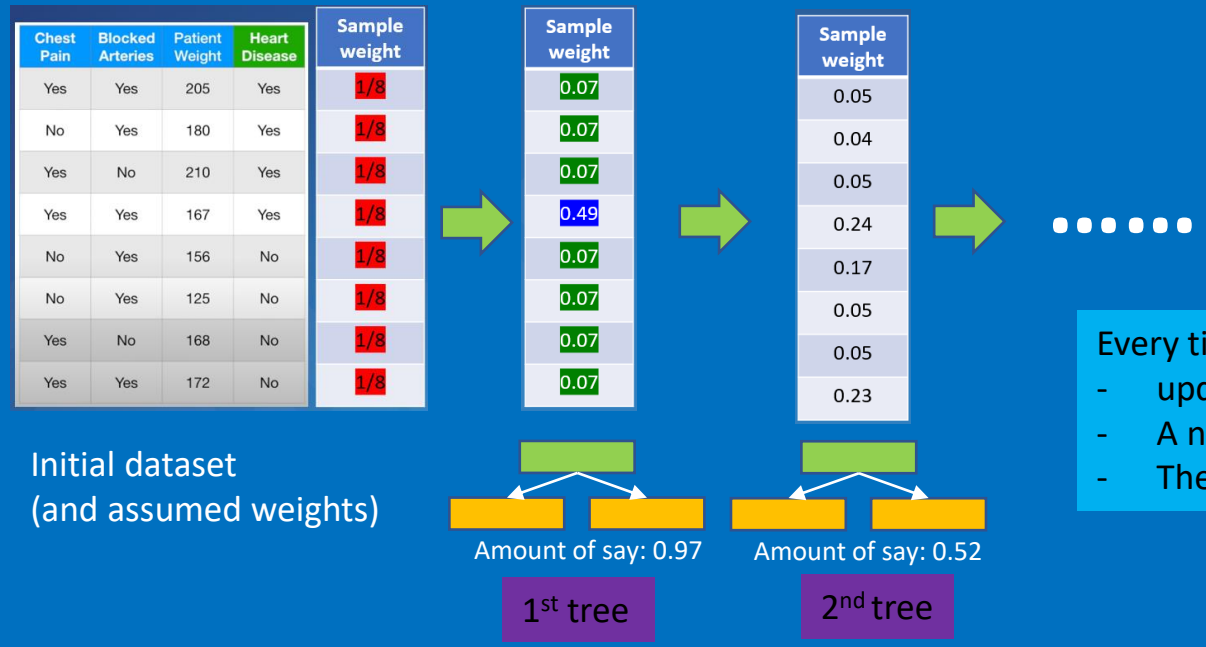


Step 3: determine how much the above tree will contribute to the final results ("amount of say")

- Calculating the weighted prediction error for the tree, which is 0.125 (see Adaboost tutorial for the details)
- Calculating the "amount of say" for the tree, which is 0.97 (see Adaboost tutorial for the details)

Step 4: Update the sample weight depending on the "amount of say"

Step 5: Go back to "step 2", using the updated weight, and repeat the processes between "step 2" and "step4"



Initial dataset
(and assumed weights)

Every time repeat this process, we will get
- updated "sample weights"
- A new tree
- The "amount of say" for the tree

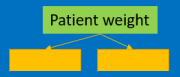
A classic boosting method is adaboost, here we use it as an example to explain what is boosting

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	0.05
No	Yes	180	Yes	0.05
Yes	No	210	Yes	0.05
Yes	Yes	167	Yes	0.33
No	Yes	156	No	0.05
No	Yes	125	No	0.05
Yes	No	168	No	0.05
Yes	Yes	172	No	0.05

Assuming we have the above dataset

Step 1: we give weights for each sample (e.g., as an initial guess, it can be $\frac{1}{total\ samples}$)

Step 2: determine the "root" tree

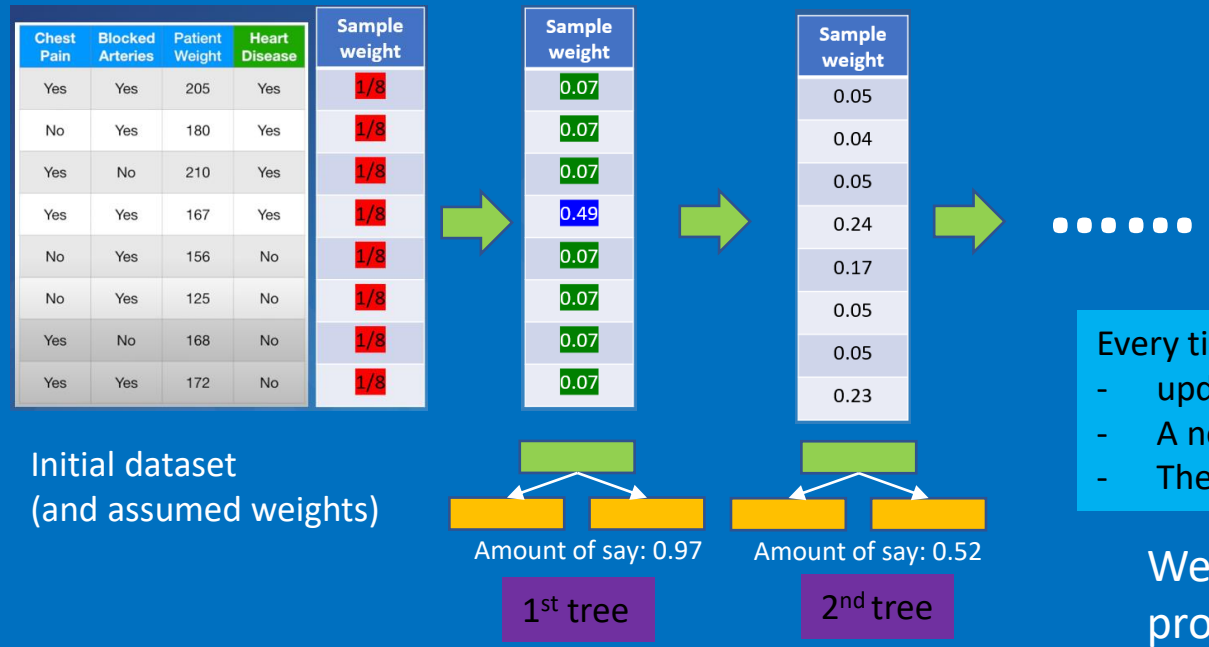


Step 3: determine how much the above tree will contribute to the final results ("amount of say")

- Calculating the weighted prediction error for the tree, which is 0.125 (see Adaboost tutorial for the details)
- Calculating the "amount of say" for the tree, which is 0.97 (see Adaboost tutorial for the details)

Step 4: Update the sample weight depending on the "amount of say"

Step 5: Go back to "step 2", using the updated weight, and repeat the processes between "step 2" and "step4"



Initial dataset (and assumed weights)

1st tree

2nd tree

Every time repeat this process, we will get
- updated "sample weights"
- A new tree
- The "amount of say" for the tree

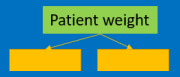
We stop this iterative processes until the "amount of say" is small enough

A classic boosting method is adaboost, here we use it as an example to explain what is boosting

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sam wei	Sample weight
Yes	Yes	205	Yes	1/	0.05
No	Yes	180	Yes	1/	0.05
Yes	No	210	Yes	1/	0.05
Yes	Yes	167	Yes	1/	0.33
No	Yes	156	No	1/	0.05
No	Yes	125	No	1/	0.05
Yes	No	168	No	1/	0.05
Yes	Yes	172	No	1/	0.05

Assuming we have the above dataset

Step 1: we give weights for each sample (e.g., as an initial guess, it can be $\frac{1}{total\ samples}$)

Step 2: determine the "root" tree 

Step 3: determine how much the above tree will contribute to the final results ("amount of say")

- Calculating the weighted prediction error for the tree, which is 0.125 (see Adaboost tutorial for the details)
- Calculating the "amount of say" for the tree, which is 0.97 (see Adaboost tutorial for the details)

Step 4: Update the sample weight depending on the "amount of say"

Step 5: Go back to "step 2", using the updated weight, and repeat the processes between "step 2" and "step4"

Chest Pain	Blocked Arteries	Patient Weight
Yes	Yes	205

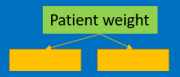
When we have a testing data

A classic boosting method is adaboost, here we use it as an example to explain what is boosting

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sam wei	Sample weight
Yes	Yes	205	Yes	1/	0.05
No	Yes	180	Yes	1/	0.05
Yes	No	210	Yes	1/	0.05
Yes	Yes	167	Yes	1/	0.33
No	Yes	156	No	1/	0.05
No	Yes	125	No	1/	0.05
Yes	No	168	No	1/	0.05
Yes	Yes	172	No	1/	0.05

Assuming we have the above dataset

Step 1: we give weights for each sample (e.g., as an initial guess, it can be $\frac{1}{total\ samples}$)

Step 2: determine the "root" tree 

Step 3: determine how much the above tree will contribute to the final results ("amount of say")

- Calculating the weighted prediction error for the tree, which is 0.125 (see Adaboost tutorial for the details)
- Calculating the "amount of say" for the tree, which is 0.97 (see Adaboost tutorial for the details)

Step 4: Update the sample weight depending on the "amount of say"

Step 5: Go back to "step 2", using the updated weight, and repeat the processes between "step 2" and "step4"

Chest Pain	Blocked Arteries	Patient Weight
Yes	Yes	205

When we have a testing data

w_1  + w_2  w_n 

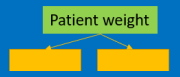
We use the testing data to go through all the trees

A classic boosting method is adaboost, here we use it as an example to explain what is boosting

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	0.05
No	Yes	180	Yes	0.05
Yes	No	210	Yes	0.05
Yes	Yes	167	Yes	0.33
No	Yes	156	No	0.05
No	Yes	125	No	0.05
Yes	No	168	No	0.05
Yes	Yes	172	No	0.05

Assuming we have the above dataset

Step 1: we give weights for each sample (e.g., as an initial guess, it can be $\frac{1}{total\ samples}$)

Step 2: determine the "root" tree 

Step 3: determine how much the above tree will contribute to the final results ("amount of say")

- Calculating the weighted prediction error for the tree, which is 0.125 (see Adaboost tutorial for the details)
- Calculating the "amount of say" for the tree, which is 0.97 (see Adaboost tutorial for the details)

Step 4: Update the sample weight depending on the "amount of say"

Step 5: Go back to "step 2", using the updated weight, and repeat the processes between "step 2" and "step4"

Chest Pain	Blocked Arteries	Patient Weight
Yes	Yes	205

When we have a testing data

w_1  + w_2  w_n 

We use the testing data to go through all the trees

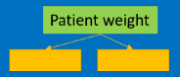
- When the tree output is "YES", we take it as "1".
- When the tree output is "NO", we take it as "0"

A classic boosting method is adaboost, here we use it as an example to explain what is boosting

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	0.05
No	Yes	180	Yes	0.05
Yes	No	210	Yes	0.05
Yes	Yes	167	Yes	0.33
No	Yes	156	No	0.05
No	Yes	125	No	0.05
Yes	No	168	No	0.05
Yes	Yes	172	No	0.05

Assuming we have the above dataset

Step 1: we give weights for each sample (e.g., as an initial guess, it can be $\frac{1}{total\ samples}$)

Step 2: determine the "root" tree 

Step 3: determine how much the above tree will contribute to the final results ("amount of say")

- Calculating the weighted prediction error for the tree, which is 0.125 (see Adaboost tutorial for the details)
- Calculating the "amount of say" for the tree, which is 0.97 (see Adaboost tutorial for the details)

Step 4: Update the sample weight depending on the "amount of say"

Step 5: Go back to "step 2", using the updated weight, and repeat the processes between "step 2" and "step4"

Chest Pain	Blocked Arteries	Patient Weight
Yes	Yes	205

When we have a testing data

w_1  + w_2  w_n 

We use the testing data to go through all the trees

- When the tree output is "YES", we take it as "1".
- When the tree output is "NO", we take it as "0"

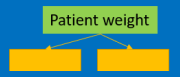
The tree output is multiplied by the "amount of say" w_i

A classic boosting method is adaboost, here we use it as an example to explain what is boosting

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sam wei	Sample weight
Yes	Yes	205	Yes	1/	0.05
No	Yes	180	Yes	1/	0.05
Yes	No	210	Yes	1/	0.05
Yes	Yes	167	Yes	1/	0.33
No	Yes	156	No	1/	0.05
No	Yes	125	No	1/	0.05
Yes	No	168	No	1/	0.05
Yes	Yes	172	No	1/	0.05

Assuming we have the above dataset

Step 1: we give weights for each sample (e.g., as an initial guess, it can be $\frac{1}{total\ samples}$)

Step 2: determine the "root" tree 

Step 3: determine how much the above tree will contribute to the final results ("amount of say")

- Calculating the weighted prediction error for the tree, which is 0.125 (see Adaboost tutorial for the details)
- Calculating the "amount of say" for the tree, which is 0.97 (see Adaboost tutorial for the details)

Step 4: Update the sample weight depending on the "amount of say"

Step 5: Go back to "step 2", using the updated weight, and repeat the processes between "step 2" and "step4"

Chest Pain	Blocked Arteries	Patient Weight
Yes	Yes	205

When we have a testing data

w_1  + w_2  w_n 

We use the testing data to go through all the trees

- When the tree output is "YES", we take it as "1".
- When the tree output is "NO", we take it as "0"

The tree output is multiplied by the "amount of say" w_i

If the final result is larger than zero, then we say the final result is "YES"

A classic boosting method is adaboost, here we use it as an example to explain what is boosting

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	0.05
No	Yes	180	Yes	0.05
Yes	No	210	Yes	0.05
Yes	Yes	167	Yes	0.33
No	Yes	156	No	0.05
No	Yes	125	No	0.05
Yes	No	168	No	0.05
Yes	Yes	172	No	0.05

Assuming we have the above dataset

Step 1: we give weights for each sample (e.g., as an initial guess, it can be $\frac{1}{total\ samples}$)

Step 2: determine the "root" tree 

Step 3: determine how much the above tree will contribute to the final results ("amount of say")

- Calculating the weighted prediction error for the tree, which is 0.125 (see Adaboost tutorial for the details)
- Calculating the "amount of say" for the tree, which is 0.97 (see Adaboost tutorial for the details)

Step 4: Update the sample weight depending on the "amount of say"

Step 5: Go back to "step 2", using the updated weight, and repeat the processes between "step 2" and "step4"

Chest Pain	Blocked Arteries	Patient Weight
Yes	Yes	205

When we have a testing data

w_1  + w_2  w_n 

We use the testing data to go through all the trees

- When the tree output is "YES", we take it as "1".
- When the tree output is "NO", we take it as "0"

The tree output is multiplied by the "amount of say" w_i

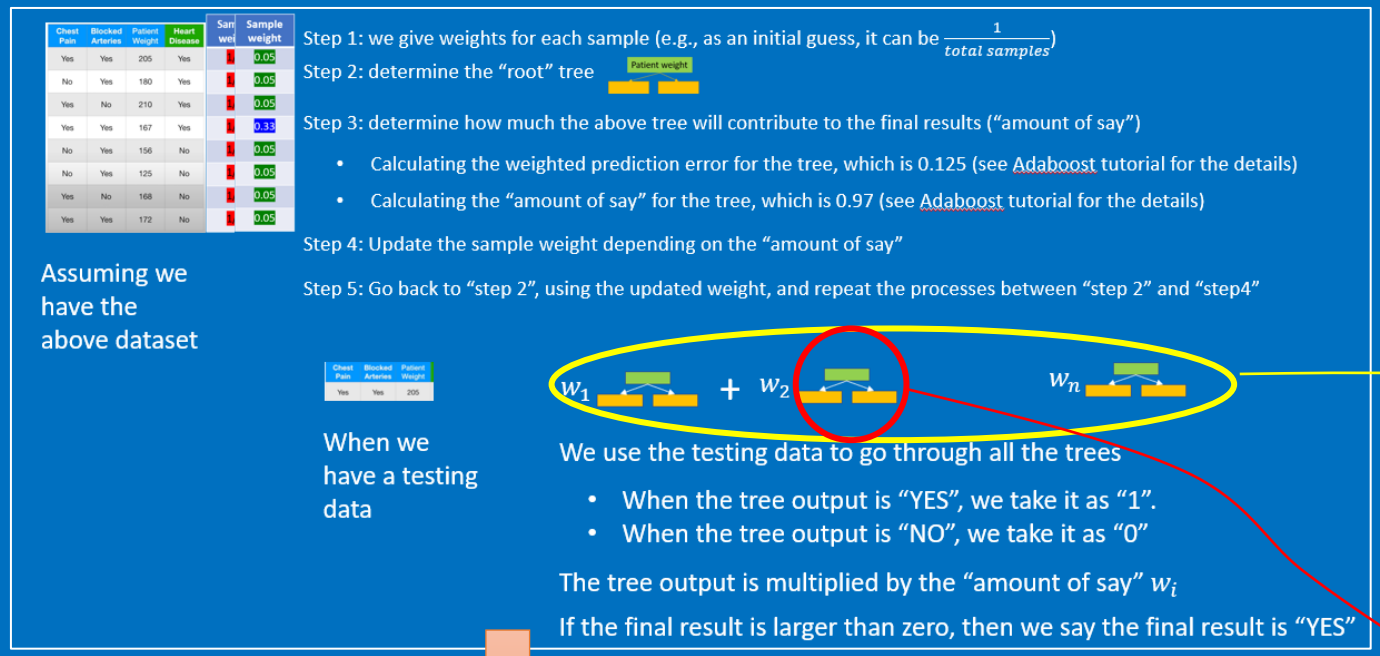
If the final result is larger than zero, then we say the final result is "YES"

What is boosting ?
Boosting is an ensemble learning method that combines a set of weak learners into a strong learner to minimize training errors.

In boosting, the sample data is fitted with a model and then trained sequentially—that is, each model tries to compensate for the weaknesses of its predecessor (so models are interconnected).

With each iteration, the weak rules from each individual classifier are combined to form one, strong prediction rule.

A classic boosting method is adaboost, here we use it as an example to explain what is boosting



Individual tree

Final combined tree

What is boosting ?
Boosting is an ensemble learning method that combines a set of weak learners into a strong learner to minimize training errors.

In boosting, the sample data is fitted with a model and then trained sequentially—that is, each model tries to compensate for the weaknesses of its predecessor (so models/trees are interconnected).

With each iteration, the weak rules from each individual classifier are combined to form one, strong prediction rule.