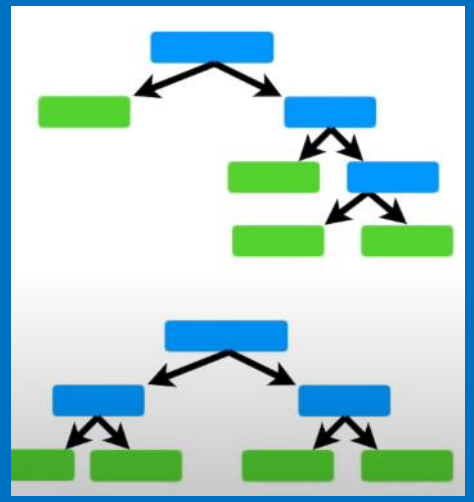
AdaBoost

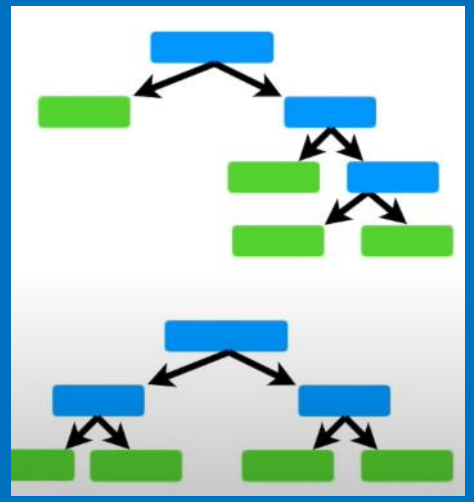
Difference between random forest and Adaboost

- In RF, each time you grow a tree, the tree is a full size tree



Difference between random forest and Adaboost

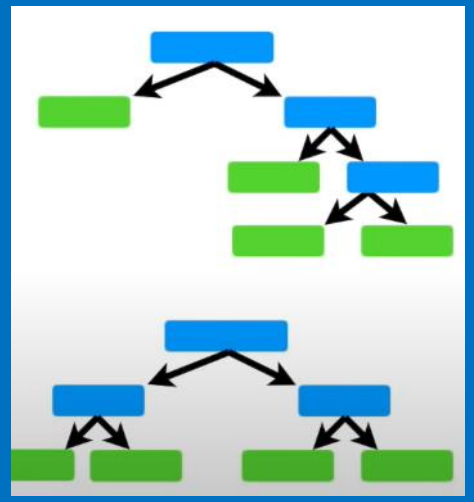
- In RF, each time you grow a tree, the tree is a full size tree



Some tree might be bigger than others, but there is no pre-defined tree depth

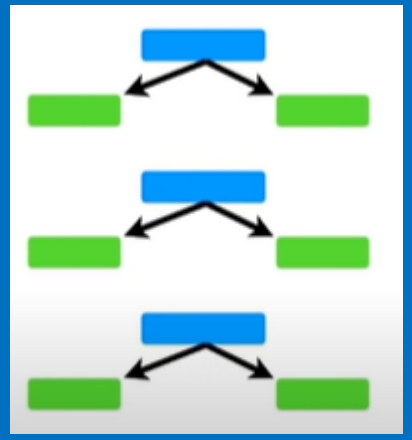
Difference between random forest and Adaboost

- In RF, each time you grow a tree, the tree is a full size tree



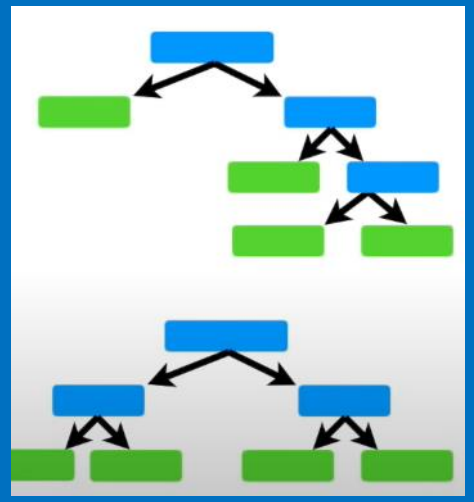
Some tree might be bigger than others, but there is no pre-defined tree depth

- In Adaboost, each time you grow a tree, the tree is just a node and two leaves



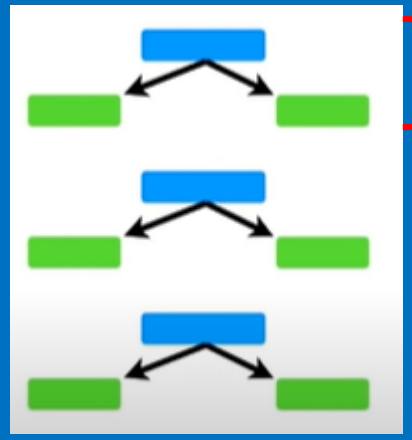
Difference between random forest and Adaboost

- In RF, each time you grow a tree, the tree is a full size tree



Some tree might be bigger than others, but there is no pre-defined tree depth

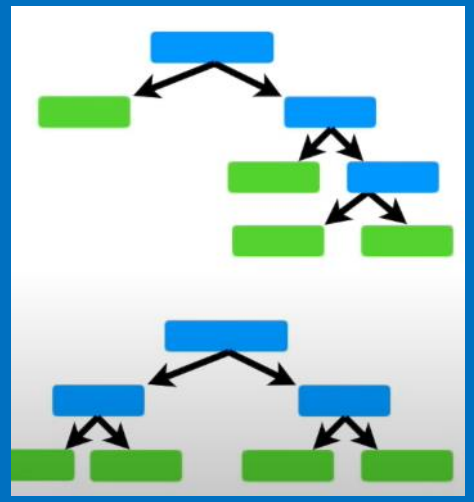
- In Adaboost, each time you grow a tree, the tree is just a node and two leaves



A tree with just one node and 2 leaves is called a “stump”

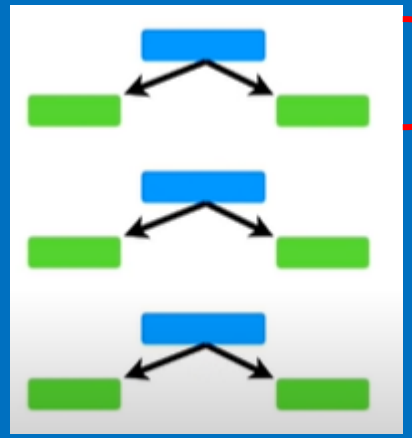
Difference between random forest and Adaboost

- In RF, each time you grow a tree, the tree is a full size tree



Some tree might be bigger than others, but there is no pre-defined tree depth

- In Adaboost, each time you grow a tree, the tree is just a node and two leaves

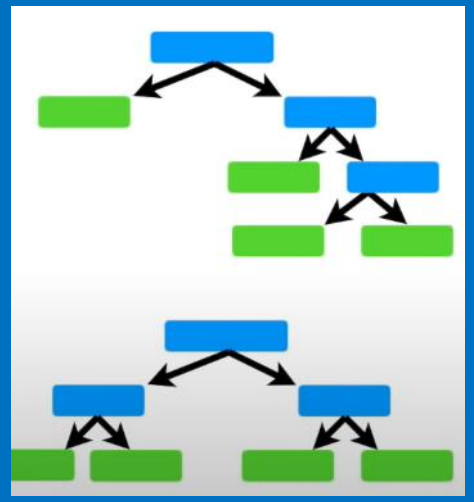


A tree with just one node and 2 leaves is called a “stump”

So Adaboost is really a “forest of stumps” rather than trees

Difference between random forest and Adaboost

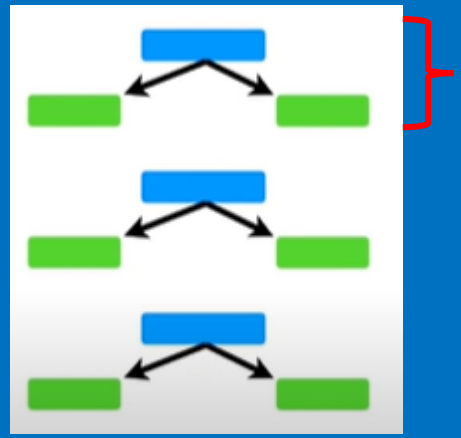
- In RF, each time you grow a tree, the tree is a full size tree



Some tree might be bigger than others, but there is no pre-defined tree depth

- In RF, each tree has an equal vote on the final classification

- In Adaboost, each time you grow a tree, the tree is just a node and two leaves

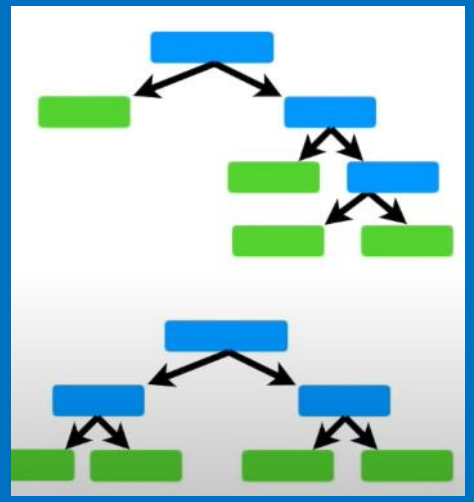


A tree with just one node and 2 leaves is called a “stump”

So Adaboost is really a “forest of stumps” rather than trees

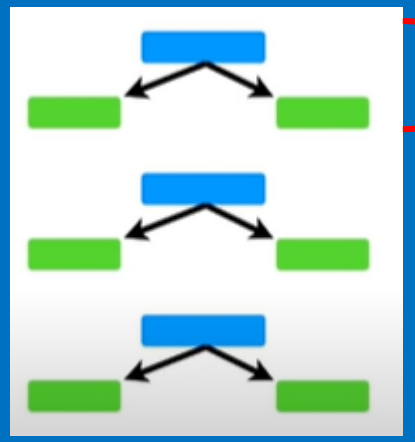
Difference between random forest and Adaboost

- In RF, each time you grow a tree, the tree is a full size tree



Some tree might be bigger than others, but there is no pre-defined tree depth

- In Adaboost, each time you grow a tree, the tree is just a node and two leaves



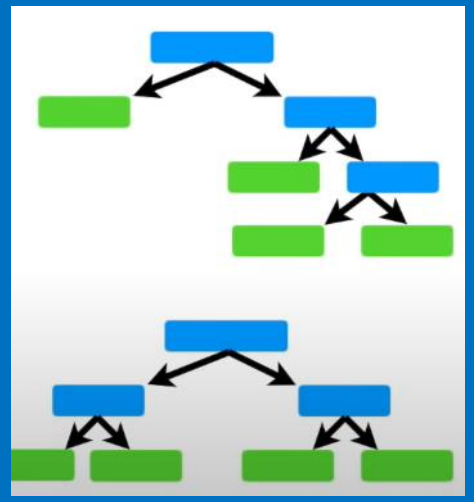
A tree with just one node and 2 leaves is called a “stump”

So Adaboost is really a “forest of stumps” rather than trees

- In RF, each tree has an equal vote on the final classification
- In Adaboost, some stumps get more say in the final classification/regression than others

Difference between random forest and Adaboost

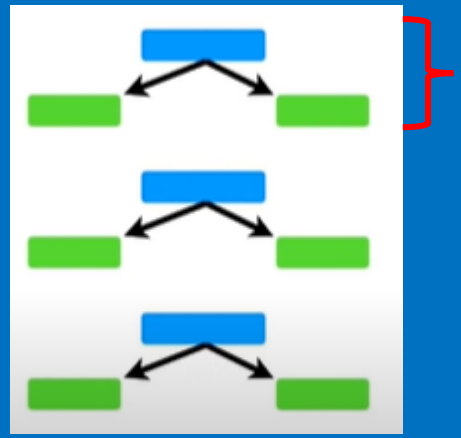
- In RF, each time you grow a tree, the tree is a full size tree



Some tree might be bigger than others, but there is no pre-defined tree depth

- In RF, each tree has an equal vote on the final classification
- In RF, each tree is made independently of the others

- In Adaboost, each time you grow a tree, the tree is just a node and two leaves



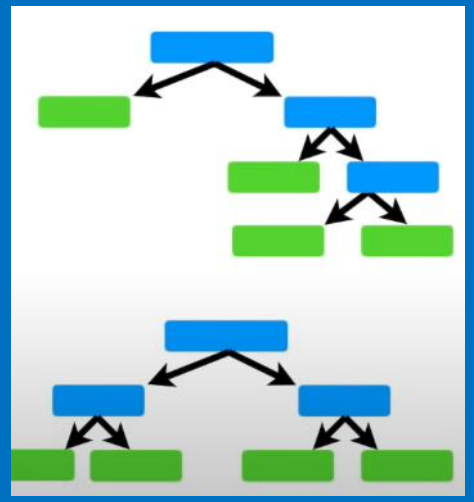
A tree with just one node and 2 leaves is called a “stump”

So Adaboost is really a “forest of stumps” rather than trees

- In Adaboost, some stumps get more say in the final classification/regression than others

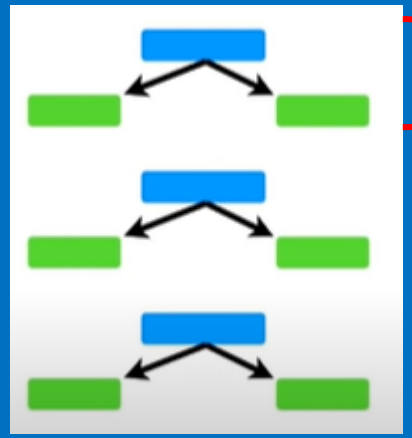
Difference between random forest and Adaboost

- In RF, each time you grow a tree, the tree is a full size tree



Some tree might be bigger than others, but there is no pre-defined tree depth

- In Adaboost, each time you grow a tree, the tree is just a node and two leaves



A tree with just one node and 2 leaves is called a "stump"

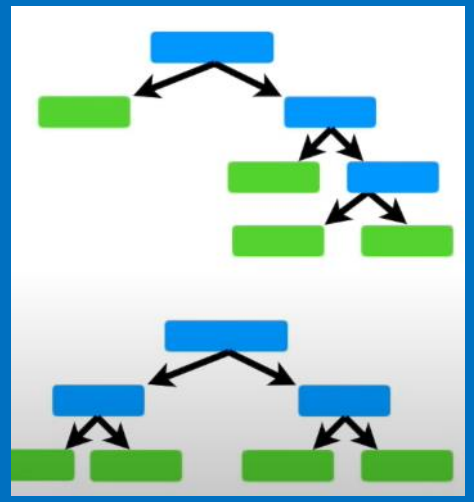
So Adaboost is really a "forest of stumps" rather than trees

- In RF, each tree has an equal vote on the final classification
- In RF, each tree is made independently of the others (so it does not matter which tree you grow first …)

- In Adaboost, some stumps get more say in the final classification/regression than others

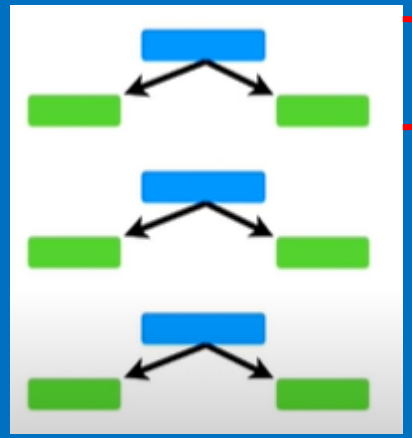
Difference between random forest and Adaboost

- In RF, each time you grow a tree, the tree is a full size tree



Some tree might be bigger than others, but there is no pre-defined tree depth

- In Adaboost, each time you grow a tree, the tree is just a node and two leaves



A tree with just one node and 2 leaves is called a "stump"

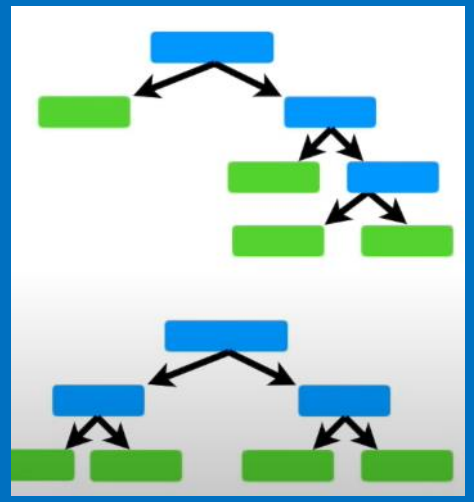
So Adaboost is really a "forest of stumps" rather than trees

- In RF, each tree has an equal vote on the final classification
- In RF, each tree is made independently of the others (so it does not matter which tree you grow first …)

- In Adaboost, some stumps get more say in the final classification/regression than others
- In Adaboost, the order of growing trees are important

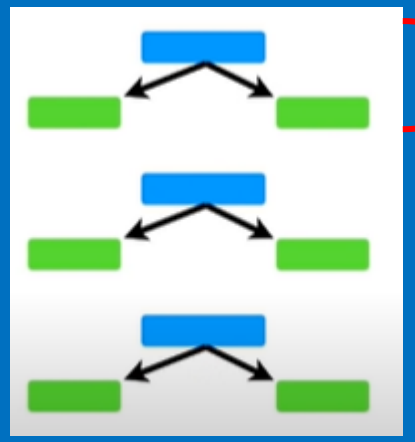
Difference between random forest and Adaboost

- In RF, each time you grow a tree, the tree is a full size tree



Some tree might be bigger than others, but there is no pre-defined tree depth

- In Adaboost, each time you grow a tree, the tree is just a node and two leaves



A tree with just one node and 2 leaves is called a "stump"

So Adaboost is really a "forest of stumps" rather than trees

- In RF, each tree has an equal vote on the final classification
- In RF, each tree is made independently of the others (so it does not matter which tree you grow first …)

- In Adaboost, some stumps get more say in the final classification/regression than others
- In Adaboost, the order of growing trees are important (e.g., the errors of the 1st tree affects how the 2nd tree would be made)

How to build a Adaboost

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

First, let’s look at the above example dataset

How to build a Adaboost

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

First, let's look at the above example dataset

We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

How to build a Adaboost

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
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	?

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

How to build a Adaboost

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

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

At the start, all the samples get the same weight: $\frac{1}{total\ samples}$, in this case, it is 1/8, which means all the samples are equally important

How to build a Adaboost

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

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"

How to build a Adaboost

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

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"

This is done by finding the variable (Chest pain, Blocked arteries and patient weight) that does the best job to indicate "heart disease"

How to build a Adaboost

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

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"

This is done by finding the variable (Chest pain, Blocked arteries and patient weight) that does the best job to indicate "heart disease"

We use Gini-index to do this (like any other decision tree method)

How to build a Adaboost

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

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

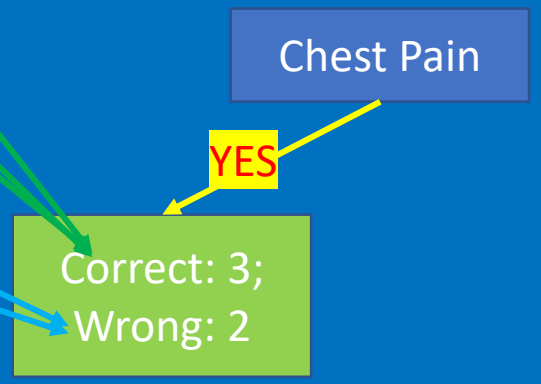
Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"

This is done by finding the variable (Chest pain, Blocked arteries and patient weight) that does the best job to indicate "heart disease"

We use Gini-index to do this (like any other decision tree method)

For the "Chest Pain", when "Heart Disease==true", three samples are correctly labelled, while two are not



How to build a Adaboost

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

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

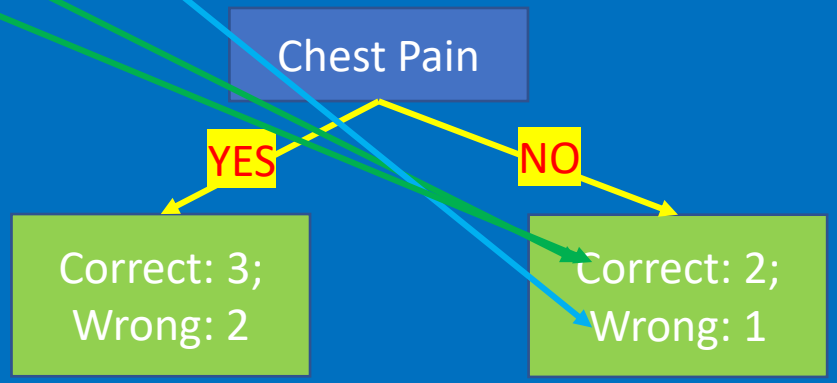
Step 2: grow our first "stump"

This is done by finding the variable (Chest pain, Blocked arteries and patient weight) that does the best job to indicate "heart disease"

We use Gini-index to do this (like any other decision tree method)

For the "Chest Pain", when "Heart Disease==true", three samples are correctly labelled, while two are not

For the "Chest Pain", when "Heart Disease==false", 2 samples are correctly labelled, while 1 are not



How to build a Adaboost

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

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

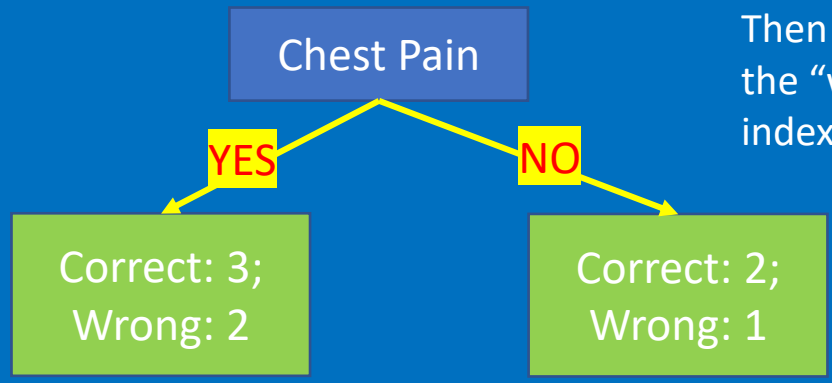
- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"

This is done by finding the variable (Chest pain, Blocked arteries and patient weight) that does the best job to indicate "heart disease"

We use Gini-index to do this (like any other decision tree method)



Then we can calculate the "weighted" Gini-index for "Chest Pain"

How to build a Adaboost

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

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

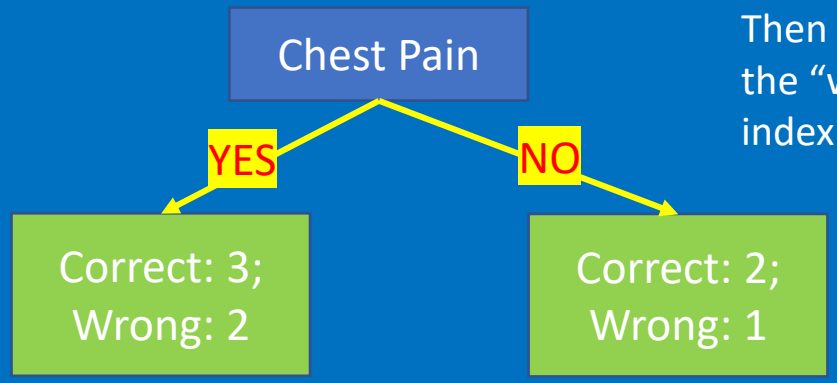
- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"

This is done by finding the variable (Chest pain, Blocked arteries and patient weight) that does the best job to indicate "heart disease"

We use Gini-index to do this (like any other decision tree method)



Then we can calculate the "weighted" Gini-index for "Chest Pain"

Note that here all the sample weight is the same, so when we calculate the Gini they can be cancelled

How to build a Adaboost

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

$G = 1 - P_{yes}{}^2 - P_{no}{}^2$

First, let's look at the above example dataset

We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

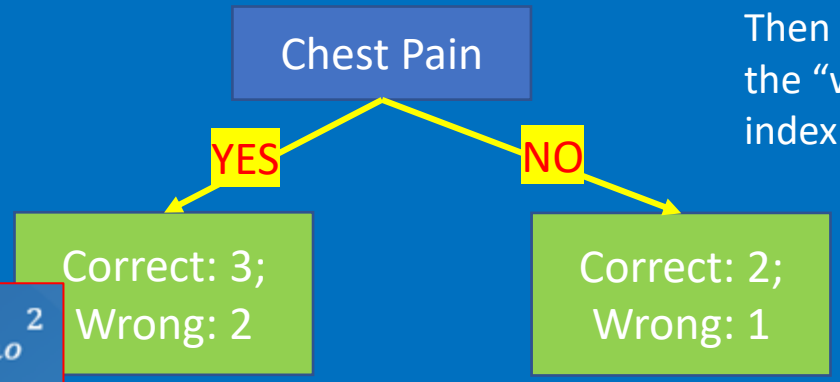
- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"

This is done by finding the variable (Chest pain, Blocked arteries and patient weight) that does the best job to indicate "heart disease"

We use Gini-index to do this (like any other decision tree method)



Then we can calculate the "weighted" Gini-index for "Chest Pain"

Note that here all the sample weight is the same, so when we calculate the Gini they can be cancelled

$$G = 1 - \left[\frac{\frac{1}{8}\times 1 + \frac{1}{8}\times 1 + \frac{1}{8}\times 1}{(\frac{1}{8}\times 1 + \frac{1}{8}\times 1 + \frac{1}{8}\times 1) + (\frac{1}{8}\times 1 + \frac{1}{8}\times 1)}\right]^2 - \left[\frac{\frac{1}{8}\times 1 + \frac{1}{8}\times 1}{(\frac{1}{8}\times 1 + \frac{1}{8}\times 1 + \frac{1}{8}\times 1) + (\frac{1}{8}\times 1 + \frac{1}{8}\times 1)}\right]^2 = 0.47$$

How to build a Adaboost

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

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

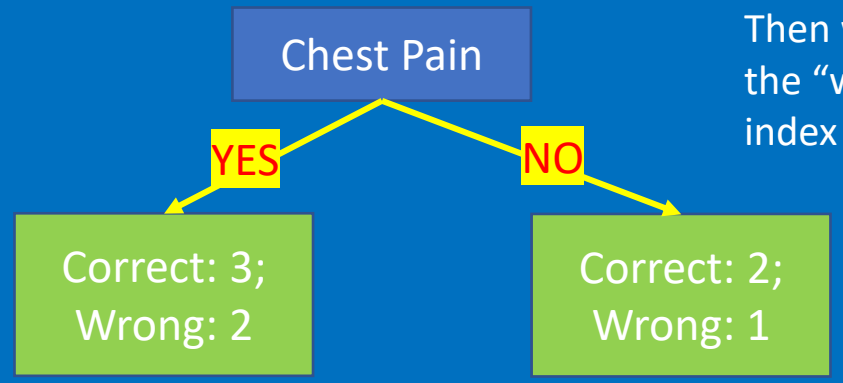
- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"

This is done by finding the variable (Chest pain, Blocked arteries and patient weight) that does the best job to indicate "heart disease"

We use Gini-index to do this (like any other decision tree method)



Then we can calculate the "weighted" Gini-index for "Chest Pain"

$$Gini_{Chest-pain} = 0.47$$

Note that here all the sample weight is the same, so when we calculate the Gini they can be cancelled

How to build a Adaboost

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

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

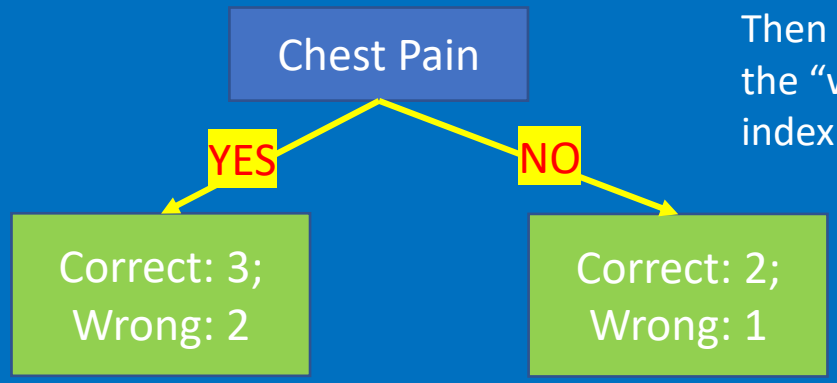
- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"

This is done by finding the variable (Chest pain, Blocked arteries and patient weight) that does the best job to indicate "heart disease"

We use Gini-index to do this (like any other decision tree method)



Then we can calculate the "weighted" Gini-index for "Chest Pain"

$$Gini_{Chest-pain} = 0.47$$

Then we can calculate the Gini-index for all the predictors

Note that here all the sample weight is the same, so when we calculate the Gini they can be cancelled

How to build a Adaboost

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

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"

This is done by finding the variable (Chest pain, Blocked arteries and patient weight) that does the best job to indicate "heart disease"

We use Gini-index to do this (like any other decision tree method)

$$Gini_{Chest-pain} = 0.47$$

$$Gini_{Blocked\ Arteries} = 0.5$$

$$Gini_{Patient\ Weight} = 0.2$$

How to build a Adaboost

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

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"

This is done by finding the variable (Chest pain, Blocked arteries and patient weight) that does the best job to indicate "heart disease"

We use Gini-index to do this (like any other decision tree method)

$$Gini_{Chest-pain} = 0.47$$

$$Gini_{Blocked\ Arteries} = 0.5$$

$$Gini_{Patient\ Weight} = 0.2$$

The Gini-index for "patient weight" is the lowest, so there is where we start growing "stumps"

How to build a Adaboost

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

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"

This is done by finding the variable (Chest pain, Blocked arteries and patient weight) that does the best job to indicate "heart disease"

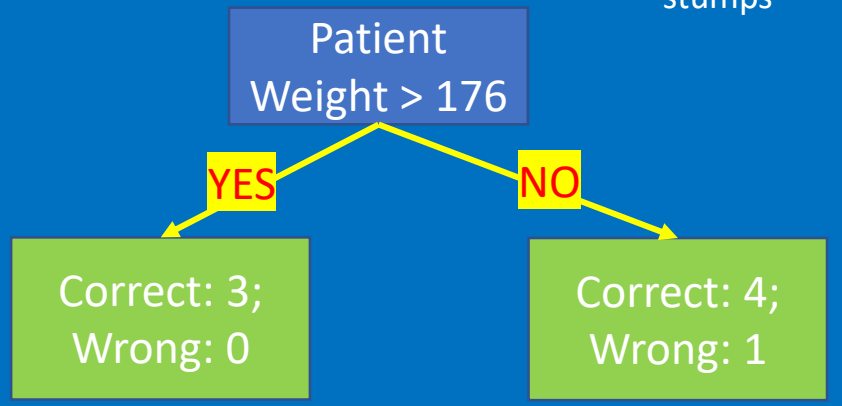
We use Gini-index to do this (like any other decision tree method)

$$Gini_{Chest-pain} = 0.47$$

$$Gini_{Blocked\ Arteries} = 0.5$$

$$Gini_{Patient\ Weight} = 0.2$$

The Gini-index for "patient weight" is the lowest, so there is where we start growing "stumps"



How to build a Adaboost

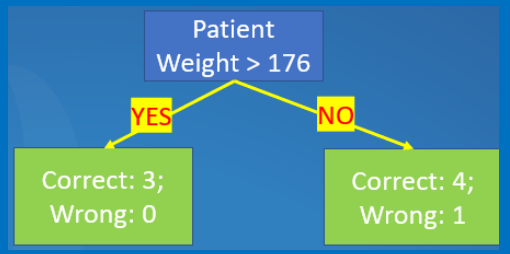
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"



Step 3: We need to determine how much this stump will say in the final results

How to build a Adaboost

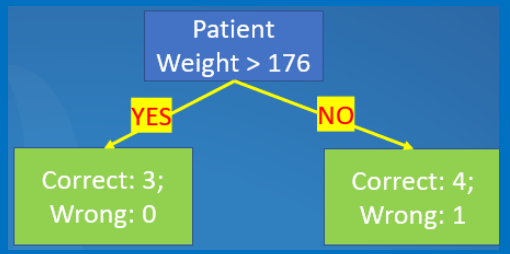
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"



Step 3: We need to determine how much this stump will say in the final results

Step 3.1: This is defined by the error of this "stump"

How to build a Adaboost

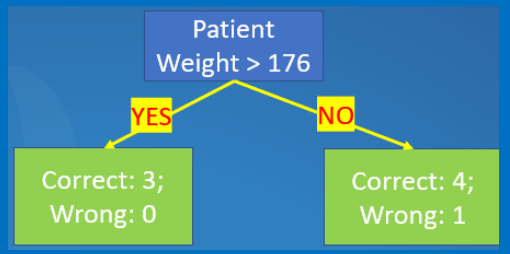
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

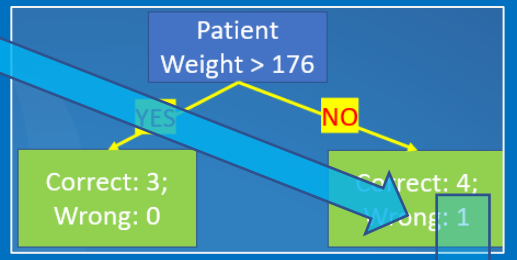
Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"



Step 3: We need to determine how much this stump will say in the final results

Step 3.1: This is defined by the error of this "stump"



This stump makes one error ~ for one sample, it "predicts" that the patient should not have disease but in fact he/she does

How to build a Adaboost

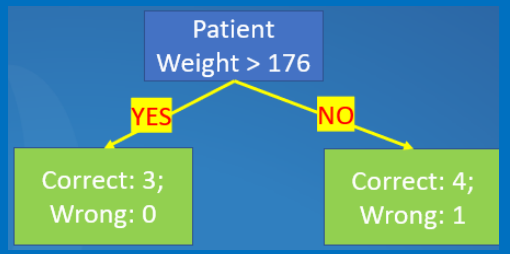
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

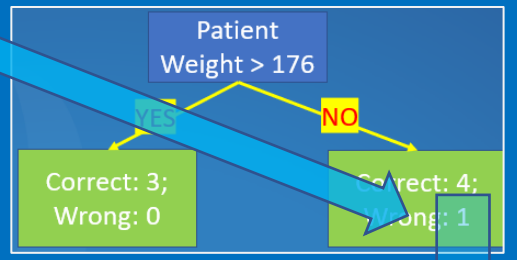
Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"



Step 3: We need to determine how much this stump will say in the final results

Step 3.1: This is defined by the error of this "stump"



This stump makes one error ~ for one sample, it "predicts" that the patient should not have disease but in fact he/she does

the "total error" for this "stump" is: "weights" x "wrong prediction"

How to build a Adaboost

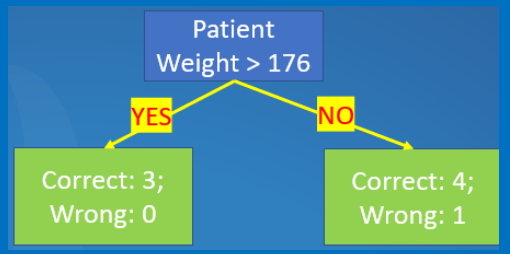
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

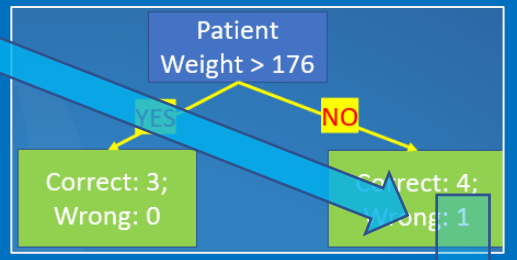
Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"



Step 3: We need to determine how much this stump will say in the final results

Step 3.1: This is defined by the error of this "stump"



This stump makes one error ~ for one sample, it "predicts" that the patient should not have disease but in fact he/she does

the "total error" for this "stump" is: "weights" x "wrong prediction"

In this case it is 1/8 x 1 = 1/8

How to build a Adaboost

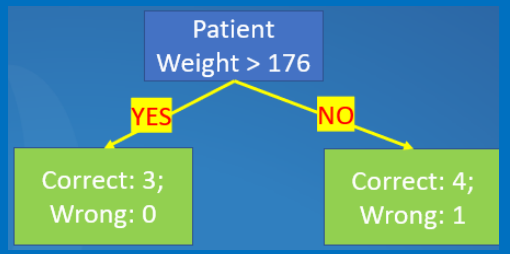
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

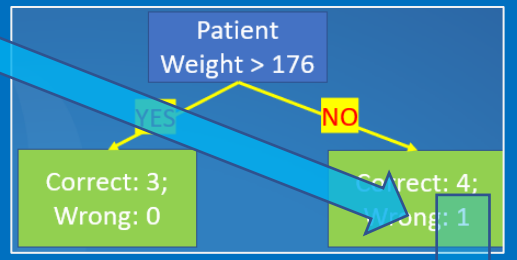
Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"



Step 3: We need to determine how much this stump will say in the final results

Step 3.1: This is defined by the error of this "stump"



This stump makes one error ~ for one sample, it "predicts" that the patient should not have disease but in fact he/she does

the "total error" for this "stump" is: "weights" x "wrong prediction"

In this case it is 1/8 x 1 = 1/8

Because all the weights are added up to 1, so "total error" will always be between 0 (prefect "stump") and 1

How to build a Adaboost

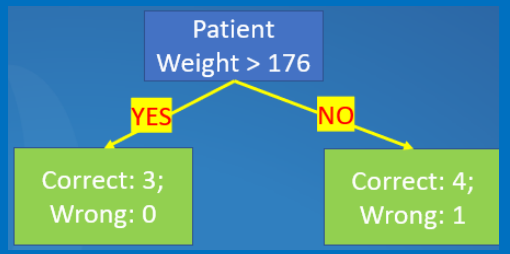
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"



Step 3: We need to determine how much this stump will say in the final results

Step 3.1: Total error for first "stump" = 1/8

Step 3.2: The amount of "say" for first "stump" is

$$Amount\ of\ say = \frac{1}{2}\log\left(\frac{1 - total\ error}{total\ error}\right)$$

How to build a Adaboost

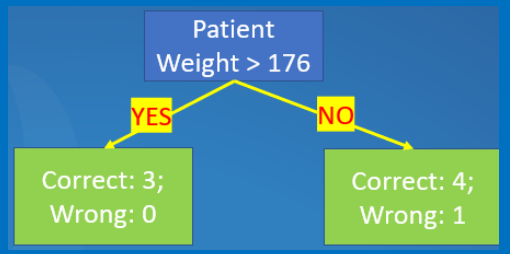
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"



Step 3: We need to determine how much this stump will say in the final results

Step 3.1: Total error for first "stump" = 1/8

Step 3.2: The amount of "say" for first "stump" is

$$Amount\ of\ say = \frac{1}{2}\log\left(\frac{1 - total\ error}{total\ error}\right)$$

In this case it is 0.97

How to build a Adaboost

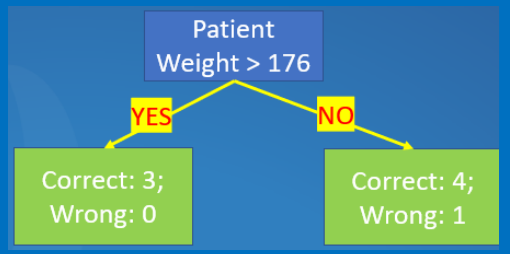
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"



Step 3: We need to determine how much this stump will say in the final results. In this case the "amount of say" is *0.97*

Step 4: Update sample weights

How to build a Adaboost

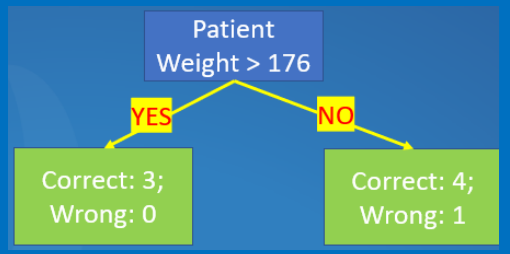
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

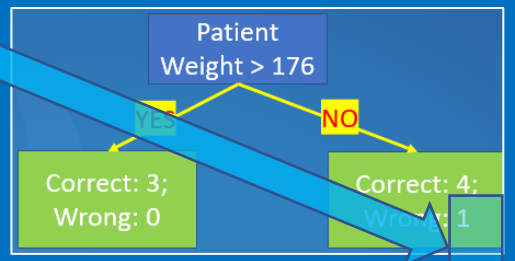
Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"



Step 3: We need to determine how much this stump will say in the final results. In this case the "amount of say" is *0.97*

Step 4: Update sample weights



In the first stump, this sample is not correctly labelled, so we need to increase its "importance" (by increasing its sample weight) so the next stump can pay more attention on it

How to build a Adaboost

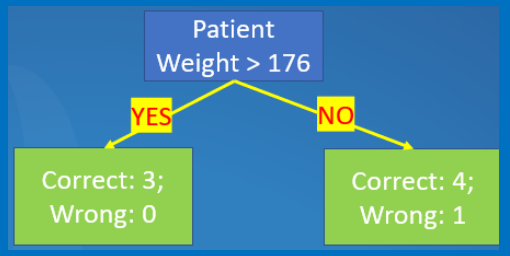
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

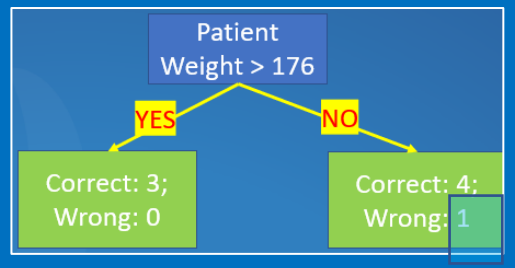
Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"



Step 3: We need to determine how much this stump will say in the final results. In this case the "amount of say" is *0.97*

Step 4: Update sample weights



New sample weight to be increased to

$$W_{new} = W_{old} \times e^{amount\ of\ say}$$

In this case it is

$$W_{new} = \frac{1}{8} \times e^{0.97} = 0.33$$

How to build a Adaboost

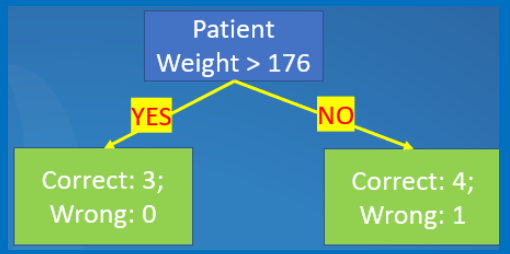
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	0.33
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

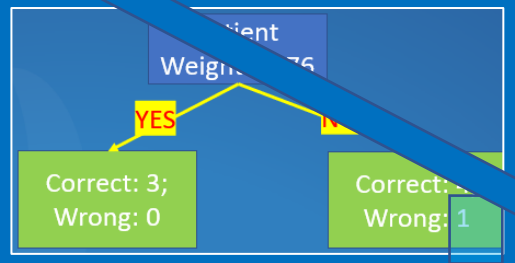
Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"



Step 3: We need to determine how much this stump will say in the final results. In this case the "amount of say" is *0.97*

St 4: Update sample weights



New sample weight to be increased to

$$W_{new} = W_{old} \times e^{amount\ of\ say}$$

In this case it is

$$W_{new} = \frac{1}{8} \times e^{0.97} = 0.33$$

How to build a Adaboost

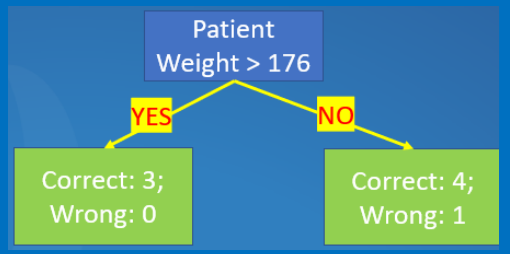
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	0.33
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"



Step 3: We need to determine how much this stump will say in the final results. In this case the "amount of say" is *0.97*

Step 4: Update sample weights

Now we need to reduce the rest sample weights (the samples are correctly labelled)

How to build a Adaboost

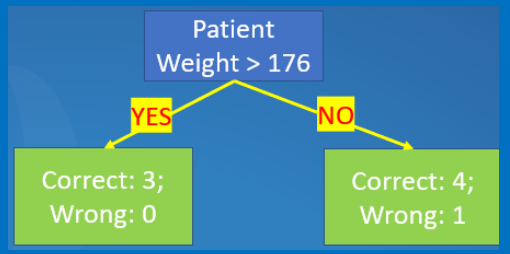
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	0.33
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"



Step 3: We need to determine how much this stump will say in the final results. In this case the "amount of say" is *0.97*

Step 4: Update sample weights

Now we need to reduce the rest sample weights (the samples are correctly labelled)

The formula to decrease the sample weight is:

$$W_{new} = W_{old} \times e^{-amount\ of\ say}$$

How to build a Adaboost

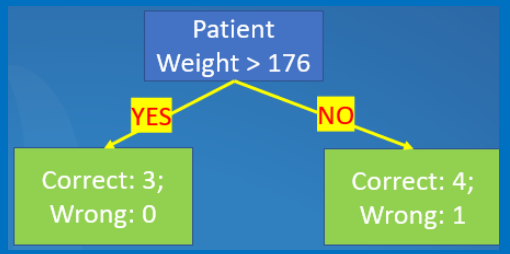
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	0.33
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"



Step 3: We need to determine how much this stump will say in the final results. In this case the "amount of say" is *0.97*

Step 4: Update sample weights

Now we need to reduce the rest sample weights (the samples are correctly labelled)

The formula to decrease the sample weight is:

$$W_{new} = W_{old} \times e^{-amount\ of\ say}$$

In this case the new weights are 0.05

How to build a Adaboost

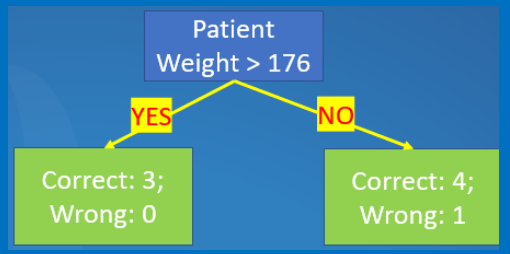
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	0.05
No	Yes	180	Yes	0.05
Yes	No	210	Yes	0.05
Yes	Yes	167	Yes	0.33
No	Yes	156	No	0.05
No	Yes	125	No	0.05
Yes	No	168	No	0.05
Yes	Yes	172	No	0.05

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"



Step 3: We need to determine how much this stump will say in the final results. In this case the "amount of say" is *0.97*

Step 4: Update sample weights

The increased sample weight is 0.33

The decreased sample weight is 0.05

How to build a Adaboost

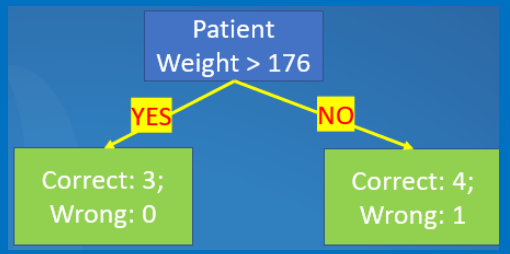
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	0.05
No	Yes	180	Yes	0.05
Yes	No	210	Yes	0.05
Yes	Yes	167	Yes	0.33
No	Yes	156	No	0.05
No	Yes	125	No	0.05
Yes	No	168	No	0.05
Yes	Yes	172	No	0.05

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"



Step 3: We need to determine how much this stump will say in the final results. In this case the "amount of say" is *0.97*

Step 4: Update sample weights

The increased sample weight is 0.33

The decreased sample weight is 0.05

The we normalize the sample weights so they can be added up to 1.0

How to build a Adaboost

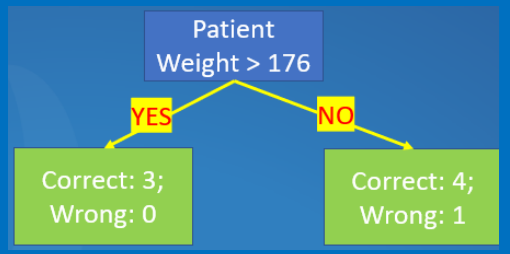
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	0.07
No	Yes	180	Yes	0.07
Yes	No	210	Yes	0.07
Yes	Yes	167	Yes	0.49
No	Yes	156	No	0.07
No	Yes	125	No	0.07
Yes	No	168	No	0.07
Yes	Yes	172	No	0.07

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"



Step 3: We need to determine how much this stump will say in the final results. In this case the "amount of say" is *0.97*

Step 4: Update sample weights

The normalized increased sample weight is 0.49

The normalized decreased sample weight is 0.07

How to build a Adaboost

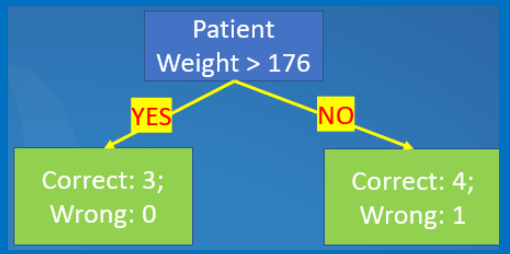
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	0.07
No	Yes	180	Yes	0.07
Yes	No	210	Yes	0.07
Yes	Yes	167	Yes	0.49
No	Yes	156	No	0.07
No	Yes	125	No	0.07
Yes	No	168	No	0.07
Yes	Yes	172	No	0.07

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"



Step 3: We need to determine how much this stump will say in the final results. In this case the "amount of say" is *0.97*

Step 4: Update sample weights

The normalized increased sample weight is 0.49

The normalized decreased sample weight is 0.07

Step 5: Then we can grow the second tree but using the updated sample weight

How to build a Adaboost

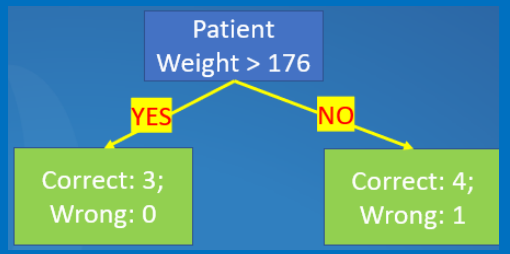
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	0.07
No	Yes	180	Yes	0.07
Yes	No	210	Yes	0.07
Yes	Yes	167	Yes	0.49
No	Yes	156	No	0.07
No	Yes	125	No	0.07
Yes	No	168	No	0.07
Yes	Yes	172	No	0.07

First, let's look at the above example dataset
We want to use AdaBoost to predict if a patient has "Heart Disease" depending on:

- *"Chest pain"*
- *"Blocked Arteries"*
- *"Patient Weight"*

Step 1: giving each sample a weight that indicates how important they are

Step 2: grow our first "stump"



Step 3: We need to determine how much this stump will say in the final results. In this case the "amount of say" is *0.97*

Step 4: Update sample weights

The normalized increased sample weight is 0.49

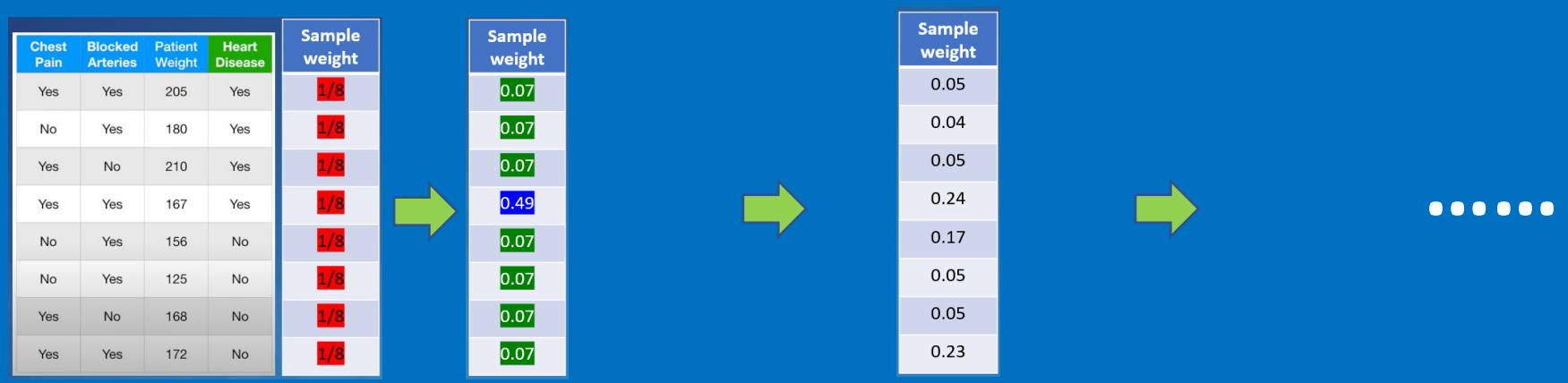
The normalized decreased sample weight is 0.07

Step 5: Then we can grow the second tree but using the updated sample weight

……

How to use a Adaboost

So from last step, we've created many "stumps", each has the "amount of say"



Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight	Sample weight	Sample weight
Yes	Yes	205	Yes	1/8	0.07	0.05
No	Yes	180	Yes	1/8	0.07	0.04
Yes	No	210	Yes	1/8	0.07	0.05
Yes	Yes	167	Yes	1/8	0.49	0.24
No	Yes	156	No	1/8	0.07	0.17
No	Yes	125	No	1/8	0.07	0.05
Yes	No	168	No	1/8	0.07	0.05
Yes	Yes	172	No	1/8	0.07	0.23

Initial dataset
(and assumed weights)

First stump

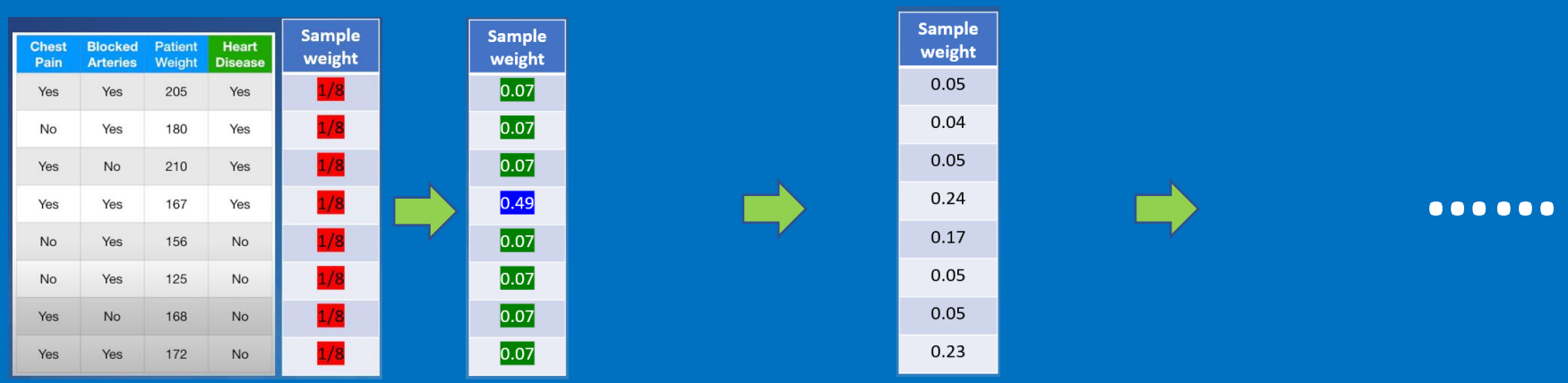
Amount of say: 0.97

2nd stump

Amount of say: 0.52

How to use a Adaboost

So from last step, we've created many "stumps", each has the "amount of say"



Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample weight	Sample weight	Sample weight
Yes	Yes	205	Yes	1/8	0.07	0.05
No	Yes	180	Yes	1/8	0.07	0.04
Yes	No	210	Yes	1/8	0.07	0.05
Yes	Yes	167	Yes	1/8	0.49	0.24
No	Yes	156	No	1/8	0.07	0.17
No	Yes	125	No	1/8	0.07	0.05
Yes	No	168	No	1/8	0.07	0.05
Yes	Yes	172	No	1/8	0.07	0.23

Initial dataset
(and assumed weights)

First stump

Amount of say: 0.97

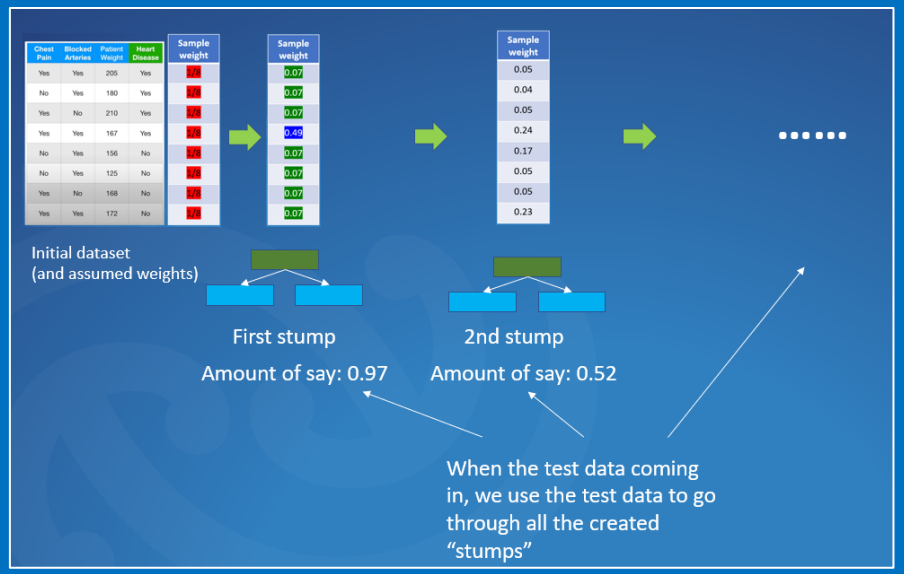
2nd stump

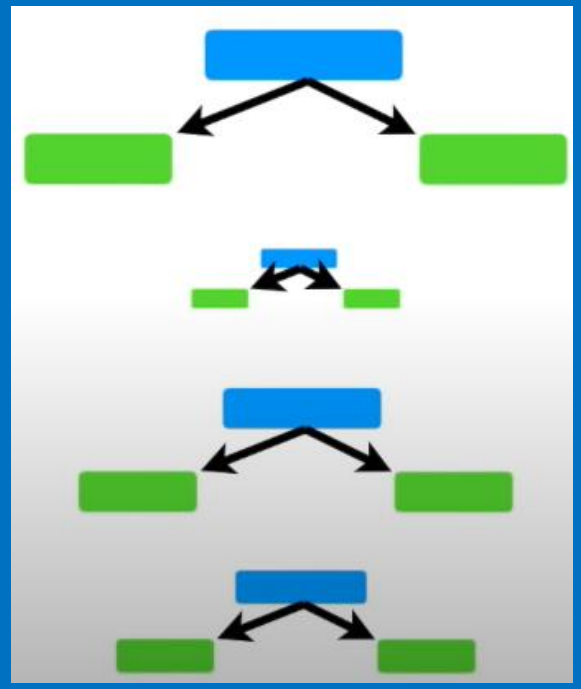
Amount of say: 0.52

When the test data coming in, we use the test data to go through all the created "stumps"

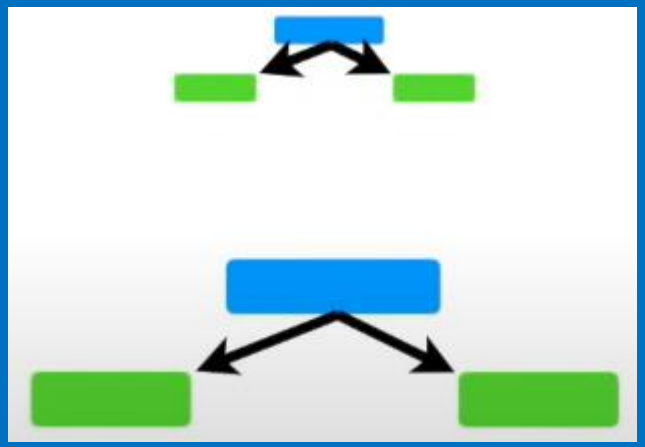
How to use a Adaboost

So from last step, we've created many "stumps", each has the "amount of say"





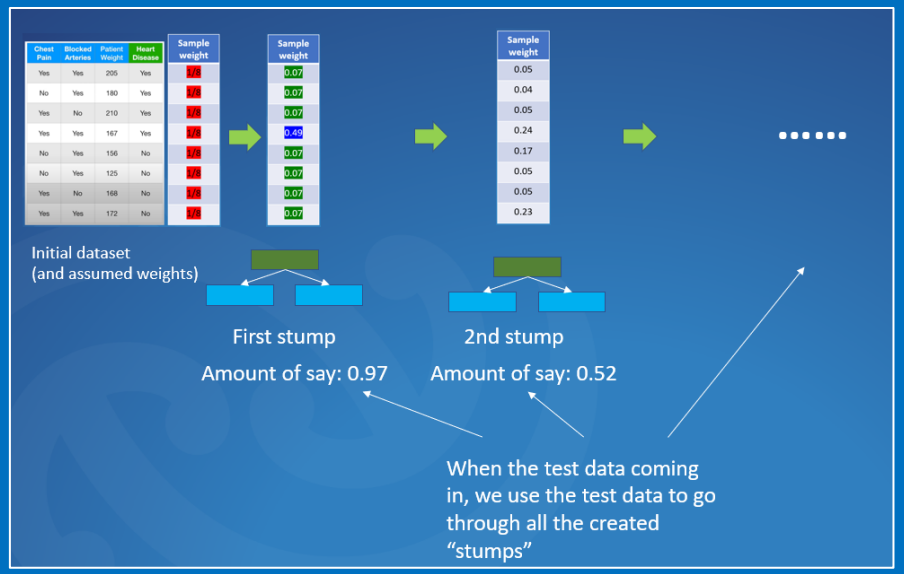
Some "stumps" will give "heart disease==True"



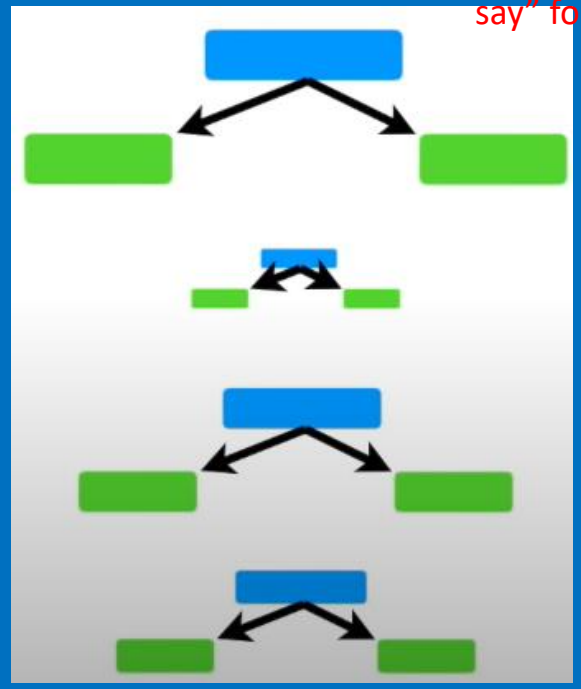
Some "stumps" will give "heart disease==False"

How to use a Adaboost

So from last step, we've created many "stumps", each has the "amount of say"



These are "amount of say" for each "stumps"



0.93

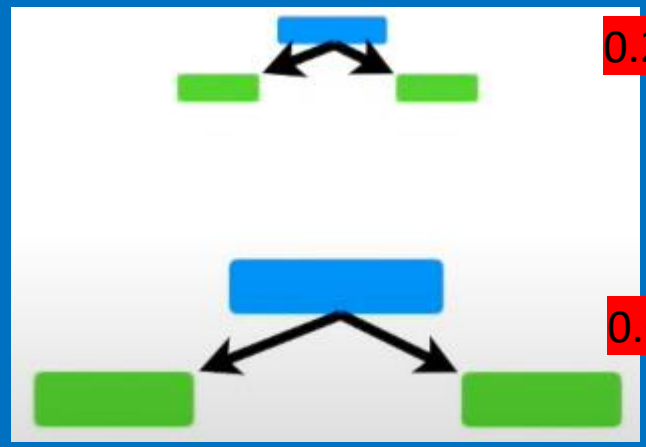
0.23

0.55

0.5

Some "stumps" will give "heart disease==True"

These are "amount of say" for each "stumps"



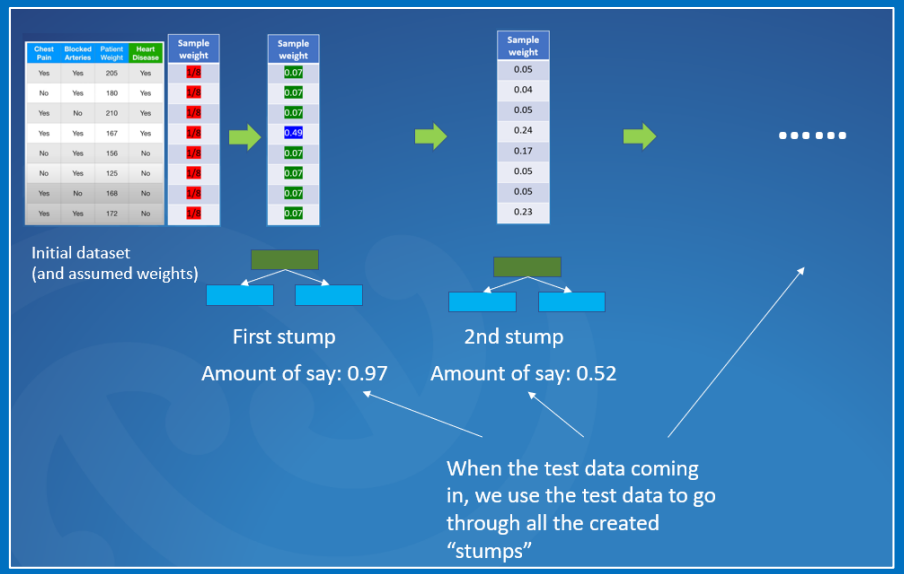
0.27

0.61

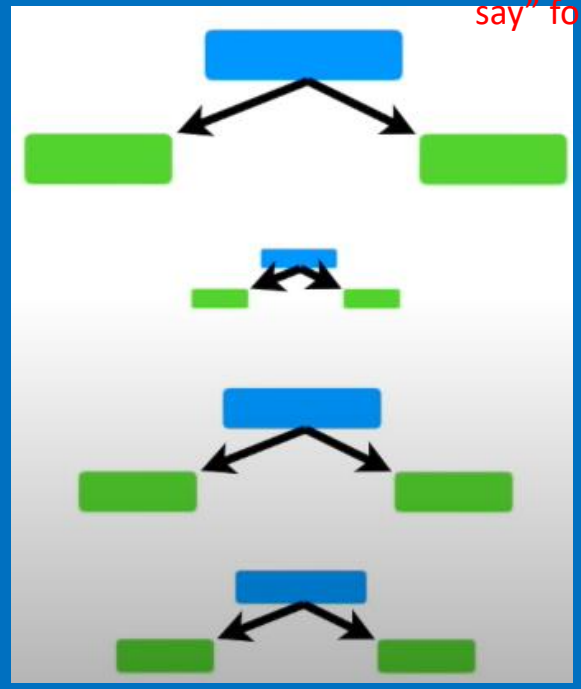
Some "stumps" will give "heart disease==False"

How to use a Adaboost

So from last step, we've created many "stumps", each has the "amount of say"



These are "amount of say" for each "stumps"



0.93

0.23

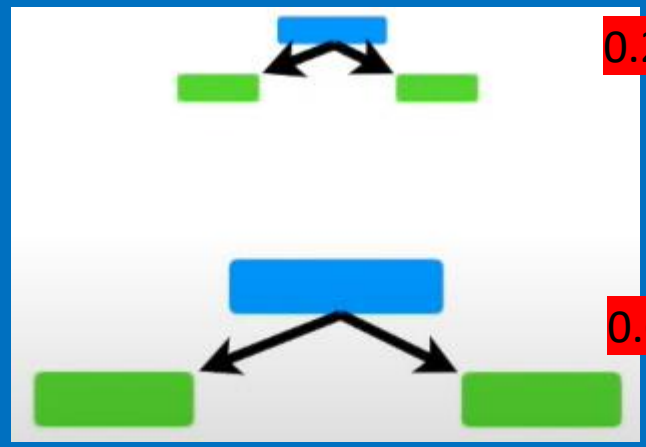
0.55

0.5

Some "stumps" will give "heart disease==True"

Total of "amount of say" 2.13

These are "amount of say" for each "stumps"



0.27

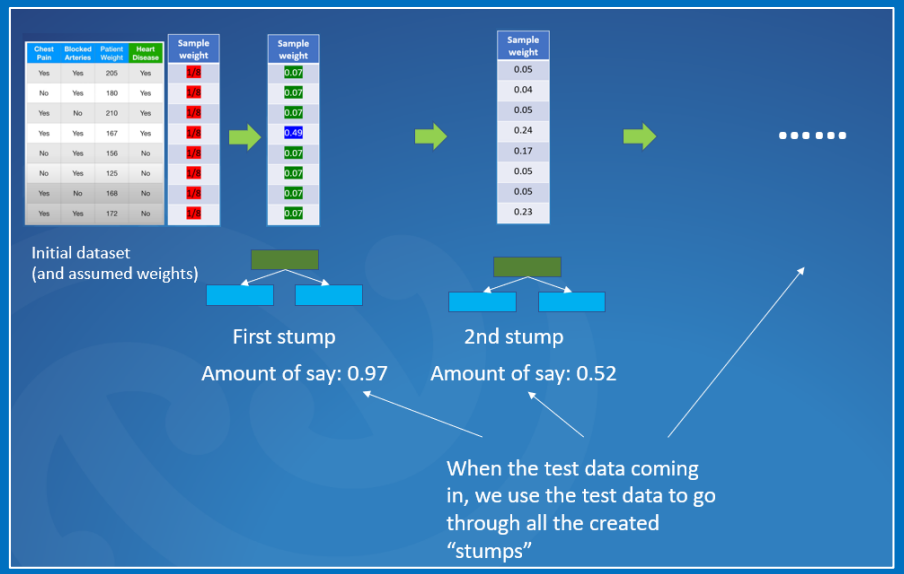
0.61

Some "stumps" will give "heart disease==False"

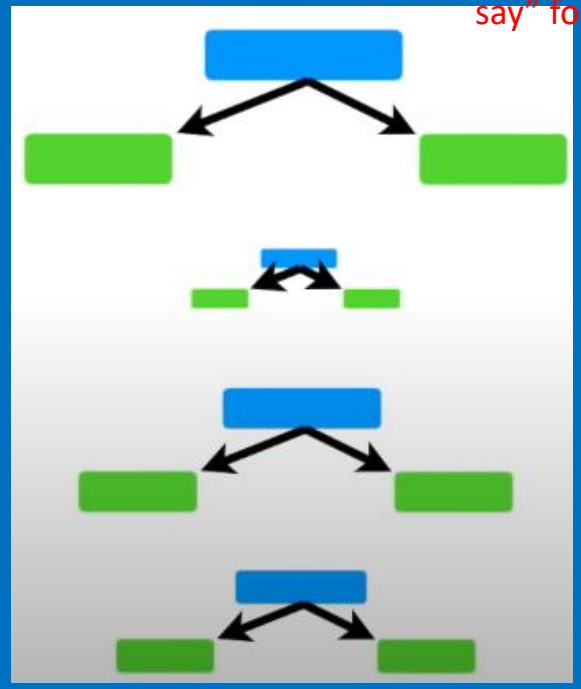
0.88

How to use a Adaboost

So from last step, we've created many "stumps", each has the "amount of say"



These are "amount of say" for each "stumps"



0.93

0.23

0.55

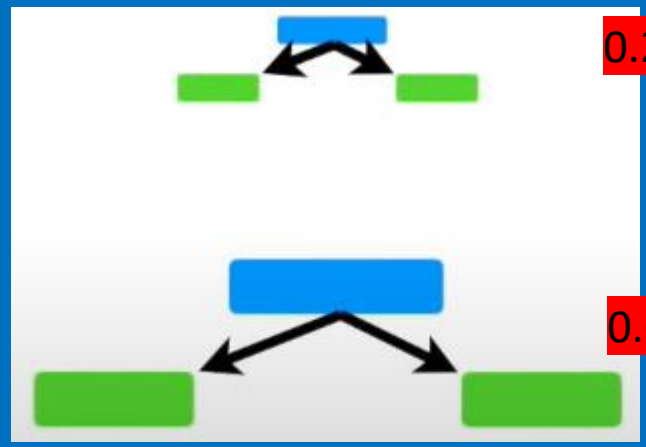
0.5

Some "stumps" will give "heart disease==True"

Total of "amount of say"

2.13

These are "amount of say" for each "stumps"



0.27

0.61

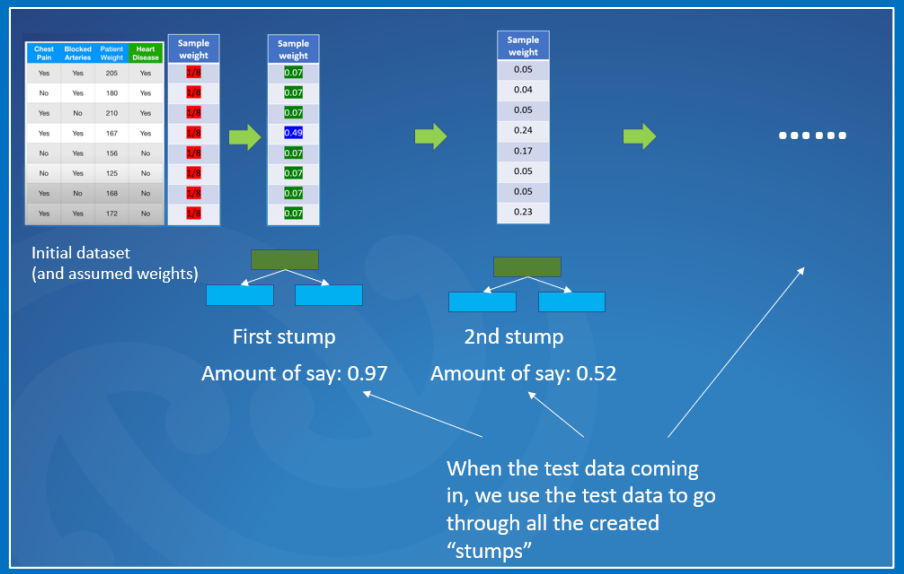
Some "stumps" will give "heart disease==False"

0.88

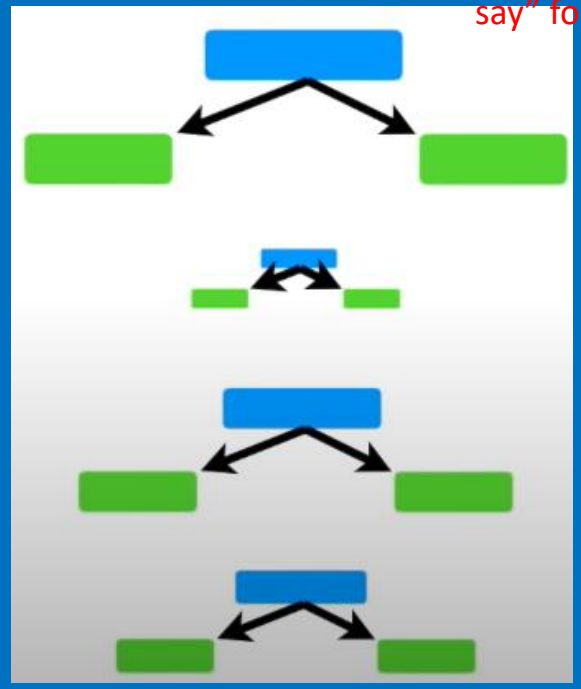
Patients has heart disease has higher "amount of say"

How to use a Adaboost

So from last step, we've created many "stumps", each has the "amount of say"



These are "amount of say" for each "stumps"



0.93

0.23

0.55

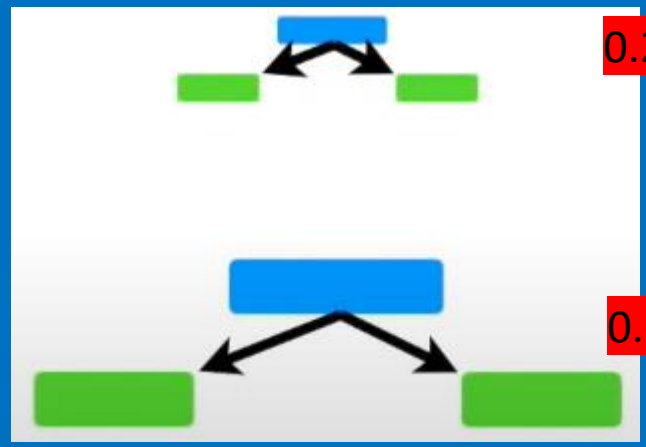
0.5

Some "stumps" will give "heart disease==True"

Total of "amount of say"

2.13

These are "amount of say" for each "stumps"



0.27

0.61

Some "stumps" will give "heart disease==False"

Patients has heart disease has higher "amount of say"

So the classification result is "heart disease==True"