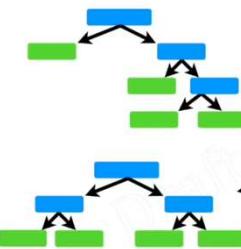
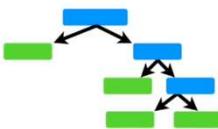


BOOSTING

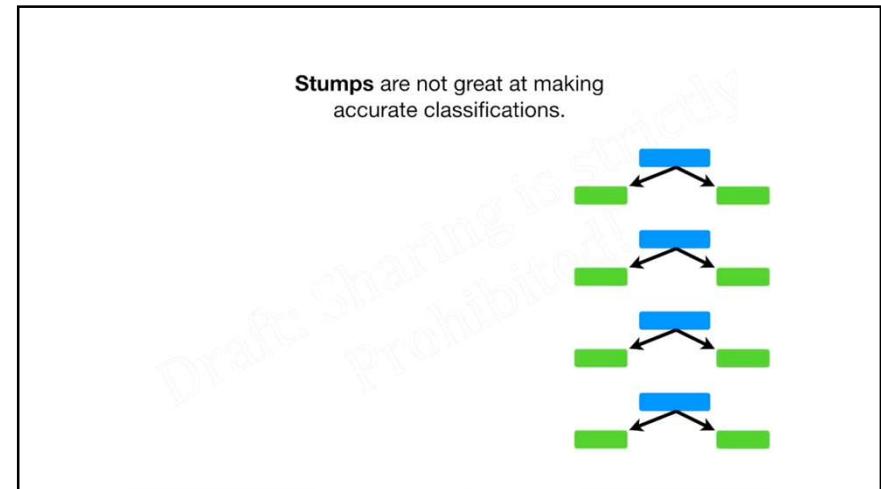
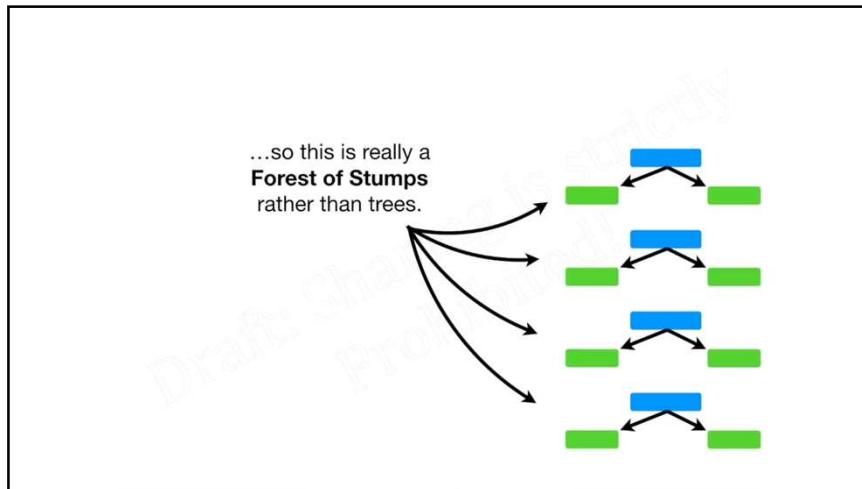
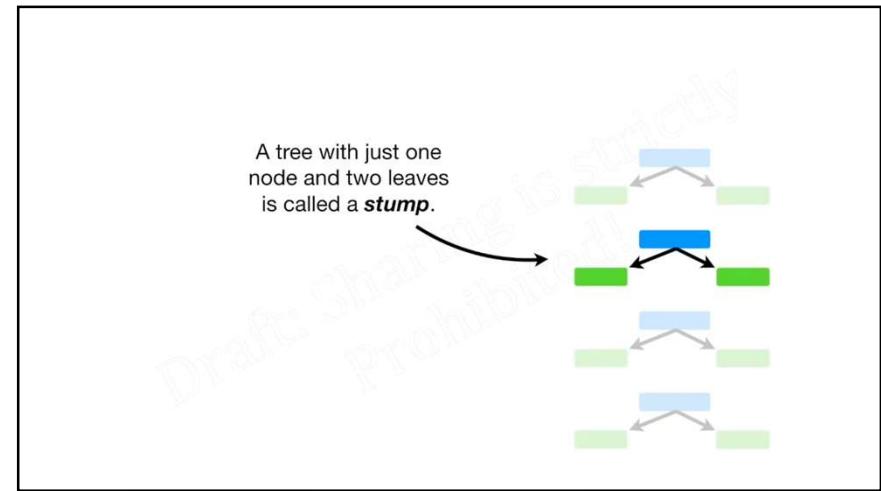
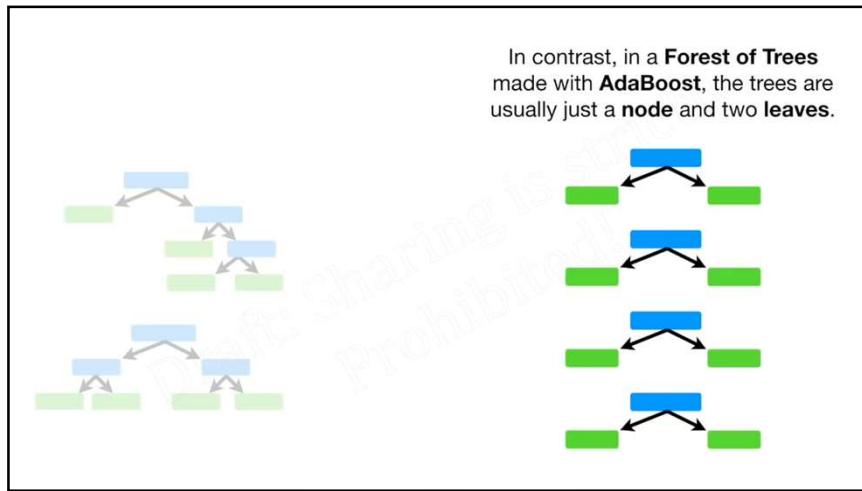
- AdaBoost
- Gradient Boost
- XGBoost

ADABOOST

In a **Random Forest**, each time you make a tree, you make a full sized tree.



Some trees might be bigger than others, but there is no predetermined maximum depth.



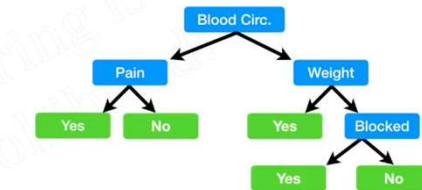
For example, if we were using this data to determine if someone had heart disease or not...

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



...then a full sized **Decision Tree** would take advantage of all 4 variables that we measured (**Chest Pain**, **Blood Circulation**, **Blocked Arteries** and **Weight**) to make a decision...

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



...but a **Stump** can only use one variable to make a decision.

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



...but a **Stump** can only use one variable to make a decision.

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



Thus, **Stumps** are technically "weak learners".

...but a **Stump** can only use one variable to make a decision.

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

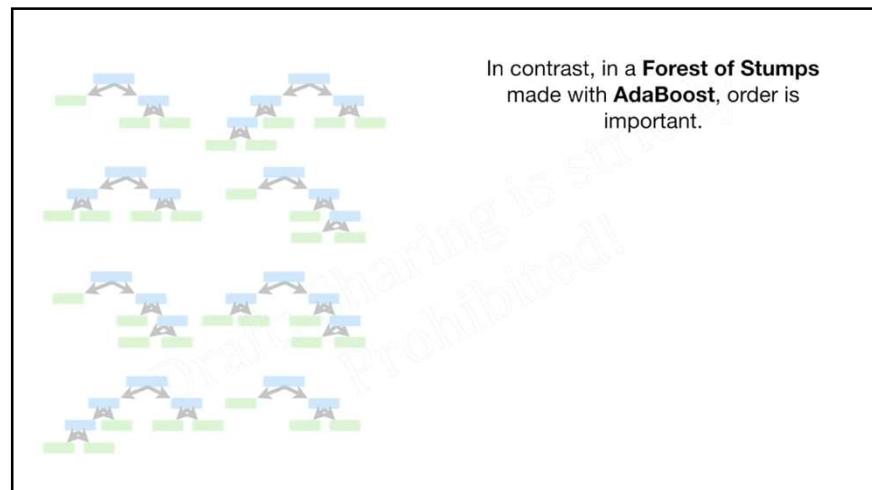
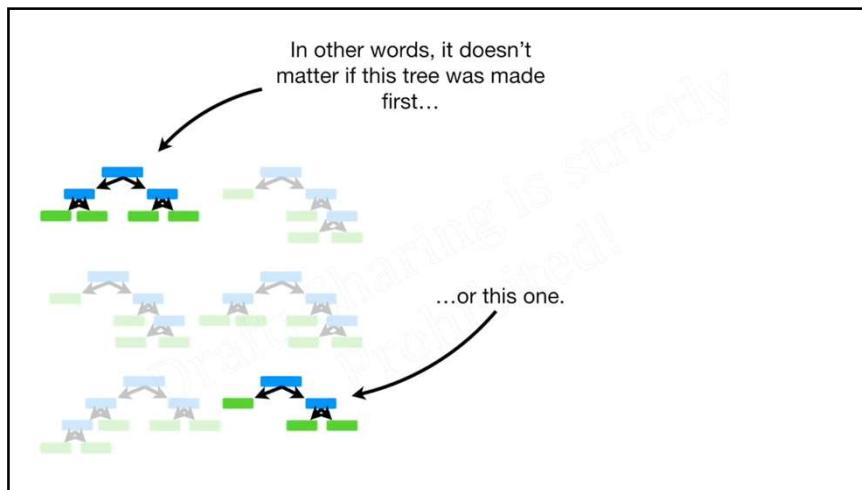
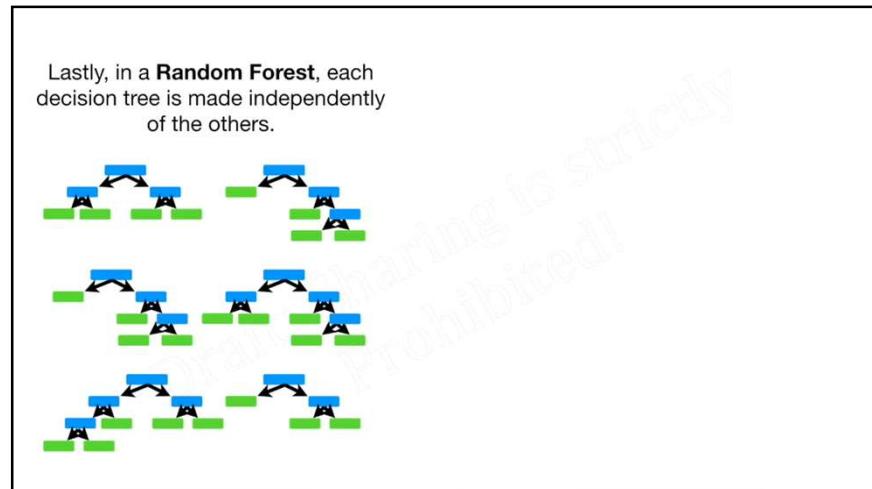
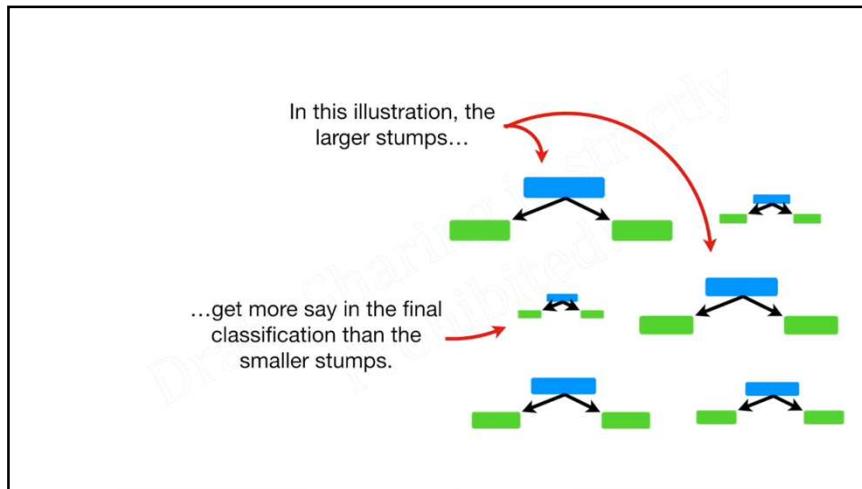
Thus, **Stumps** are technically “weak learners”. However, that’s the way **AdaBoost** likes it, and it’s one of the reasons why they are so commonly combined.

In a **Random Forest**, each tree has an equal vote on the final classification.

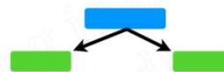
This tree's vote... ...is worth just as much as this tree's vote...

...or this tree's vote.

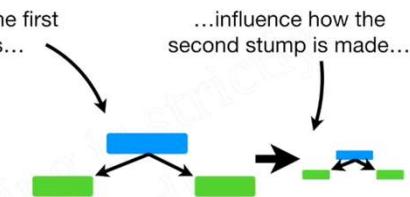
In contrast, in a **Forest of Stumps** made with **AdaBoost**, some stumps get more say in the final classification than others.



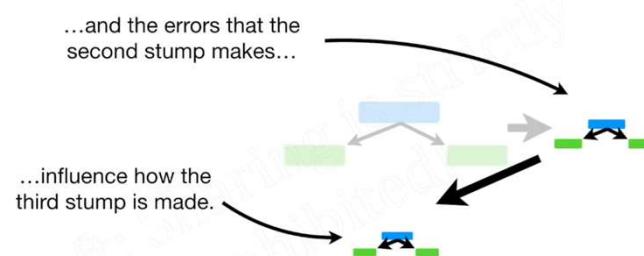
The errors that the first stump makes...



The errors that the first stump makes...



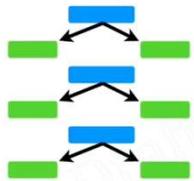
...and the errors that the second stump makes...



To review, the three ideas behind **AdaBoost** are...

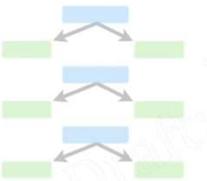
To review, the three ideas behind **AdaBoost** are...

- 1) AdaBoost combines a lot of "weak learners" to make classifications. The weak learners are almost always **stumps**.

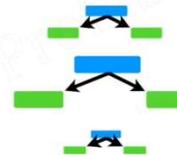


To review, the three ideas behind **AdaBoost** are...

- 1) AdaBoost combines a lot of "weak learners" to make classifications. The weak learners are almost always stumps.

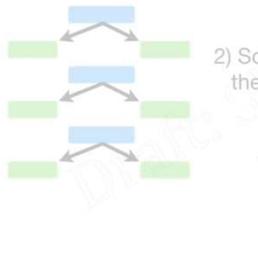


- 2) Some stumps get more say in the classification than others.

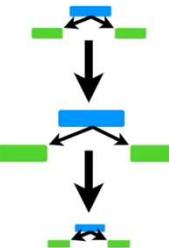


To review, the three ideas behind **AdaBoost** are...

- 1) AdaBoost combines a lot of "weak learners" to make classifications. The weak learners are almost always stumps.



- 3) Each **stump** is made by taking the previous **stump's** mistakes into account.



Now let's dive into the nitty gritty detail of how to create a **Forest of Stumps** using **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, we'll start with some data.

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

We create a **Forest of Stumps** with **AdaBoost** to predict if a patient has heart disease.

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

We will make these predictions based on a patient's **Chest Pain** and **Blocked Artery** status and their **Weight**.

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	

The first thing we do is give each sample a weight that indicates how important it is to be correctly classified.

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

At the start, all samples get the same weight...

$$\frac{1}{\text{total number of samples}} = \frac{1}{8}$$

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

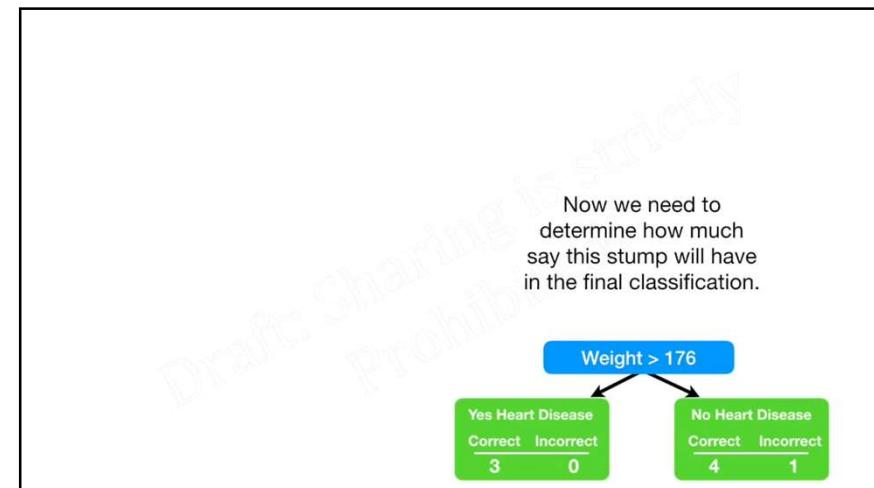
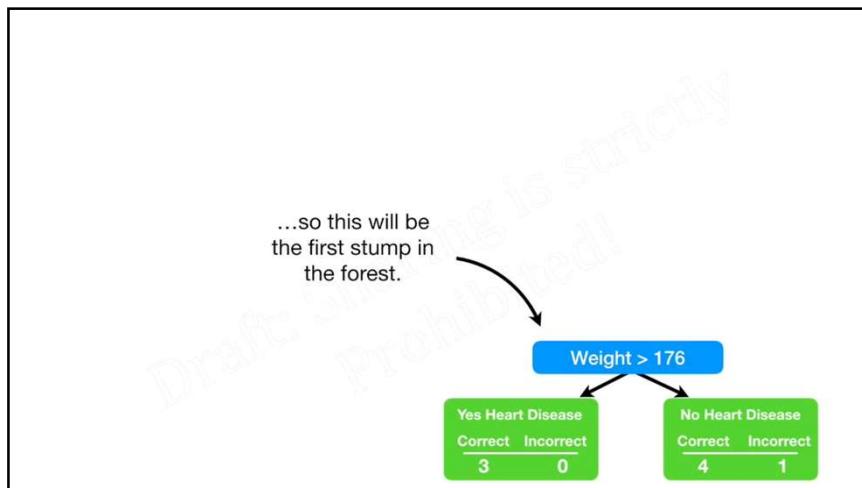
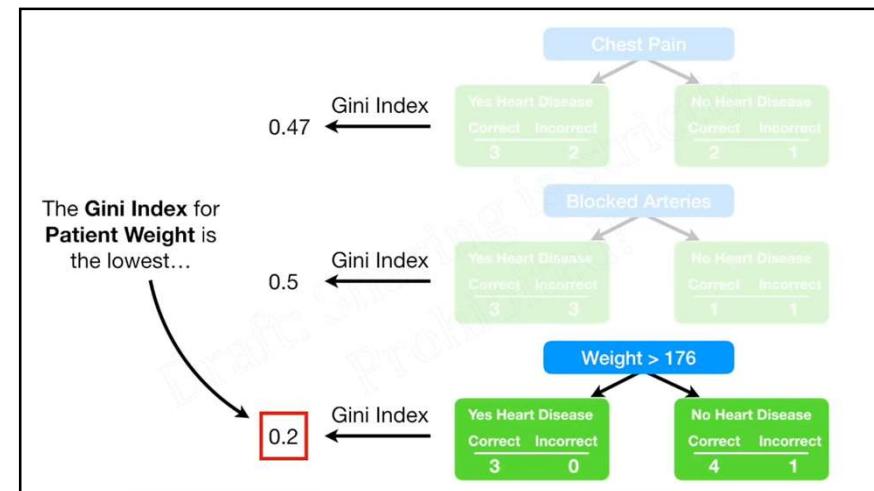
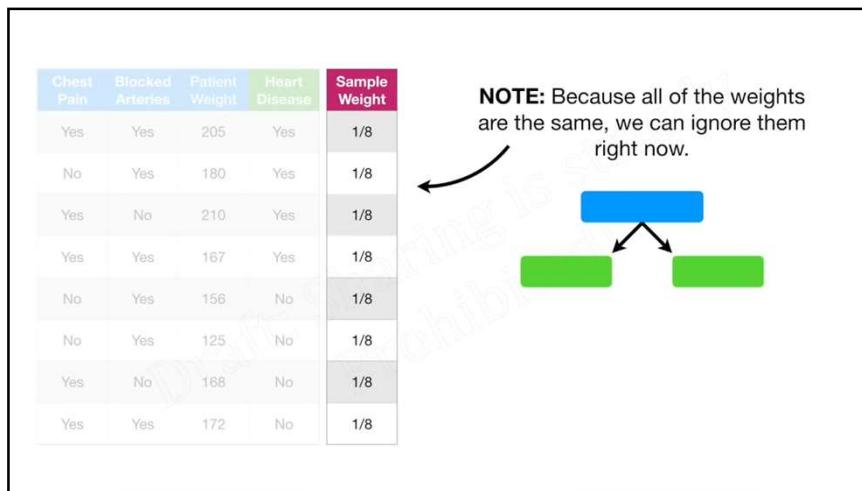
However, after we make the first stump, these weights will change in order to guide how the next stump is created.

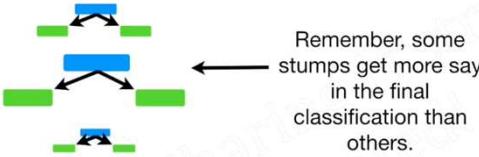
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

Now we need to make the first **stump** in the forest.

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

This is done finding the variable, **Chest Pain**, **Blocked Arteries** or **Patient Weight**, that does the best job classifying the samples.





Weight > 176			
Yes Heart Disease		No Heart Disease	
Correct	Incorrect	Correct	Incorrect
3	0	4	1

We determine how much say a stump has in the final classification based on how well it classified the samples.

Weight > 176			
Yes Heart Disease		No Heart Disease	
Correct	Incorrect	Correct	Incorrect
3	0	4	1

This stump made one error.

Weight > 176			
Yes Heart Disease		No Heart Disease	
Correct	Incorrect	Correct	Incorrect
3	0	4	1

This patient, who weighs less than 176, has heart disease, but the stump says they do not.

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

Weight > 176			
Yes Heart Disease		No Heart Disease	
Correct	Incorrect	Correct	Incorrect
3	0	4	1

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

The **Total Error** for a stump is the sum of the weights associated with the *incorrectly* classified samples.

```

graph TD
    Root[Weight > 176] --> Yes1[Yes Heart Disease  
Correct: 3  
Incorrect: 0]
    Root --> No1[No Heart Disease  
Correct: 4  
Incorrect: 1]
  
```

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

Thus, in this case, the **Total Error** is 1/8.

```

graph TD
    Root[Weight > 176] --> Yes2[Yes Heart Disease  
Correct: 3  
Incorrect: 0]
    Root --> No2[No Heart Disease  
Correct: 4  
Incorrect: 1]
  
```

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

NOTE: Because all of the **Sample Weights** add up to **1**, **Total Error** will always be between **0**, for a perfect stump, and **1**, for a horrible stump.

```

graph TD
    Root[Weight > 176] --> Yes3[Yes Heart Disease  
Correct: 3  
Incorrect: 0]
    Root --> No3[No Heart Disease  
Correct: 4  
Incorrect: 1]
  
```

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

We use the **Total Error** to determine **Amount of Say** this stump has in the final classification with the following formula:

```

graph TD
    Root[Weight > 176] --> Yes4[Yes Heart Disease  
Correct: 3  
Incorrect: 0]
    Root --> No4[No Heart Disease  
Correct: 4  
Incorrect: 1]
  
```

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

We use the **Total Error** to determine **Amount of Say** this stump has in the final classification with the following formula:

$$\text{Amount of Say} = \frac{1}{2} \log\left(\frac{1 - \text{Total Error}}{\text{Total Error}}\right)$$

```

graph TD
    A[Weight > 176] --> B[Yes Heart Disease  
Correct | Incorrect  
3 | 0]
    A --> C[No Heart Disease  
Correct | Incorrect  
4 | 1]
  
```

We can draw a graph of the **Amount of Say** by plugging in a bunch of numbers between 0 and 1 for **Total Error**.

$$\text{Amount of Say} = \frac{1}{2} \log\left(\frac{1 - \text{Total Error}}{\text{Total Error}}\right)$$

```

graph TD
    A[Weight > 176] --> B[Yes Heart Disease  
Correct | Incorrect  
3 | 0]
    A --> C[No Heart Disease  
Correct | Incorrect  
4 | 1]
  
```

The **Blue Line** tells us the **Amount of Say** for **Total Error** values between 0 and 1.

$$\text{Amount of Say} = \frac{1}{2} \log\left(\frac{1 - \text{Total Error}}{\text{Total Error}}\right)$$

```

graph TD
    A[Weight > 176] --> B[Yes Heart Disease  
Correct | Incorrect  
3 | 0]
    A --> C[No Heart Disease  
Correct | Incorrect  
4 | 1]
  
```

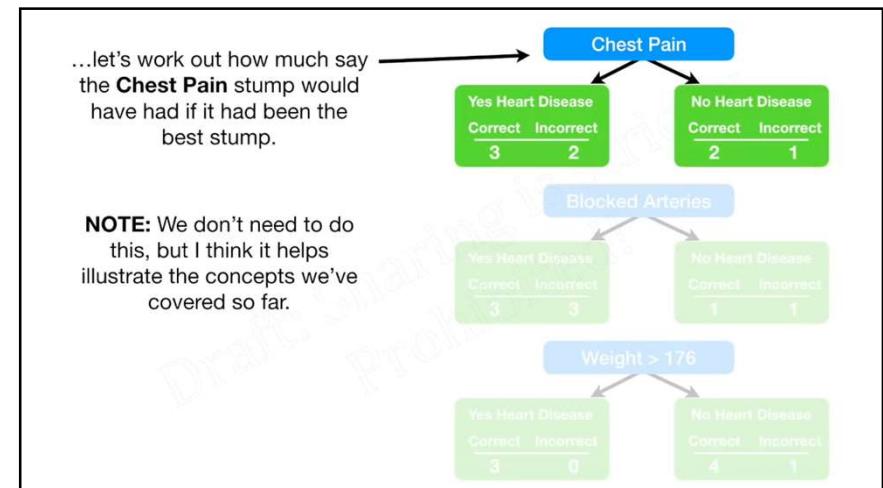
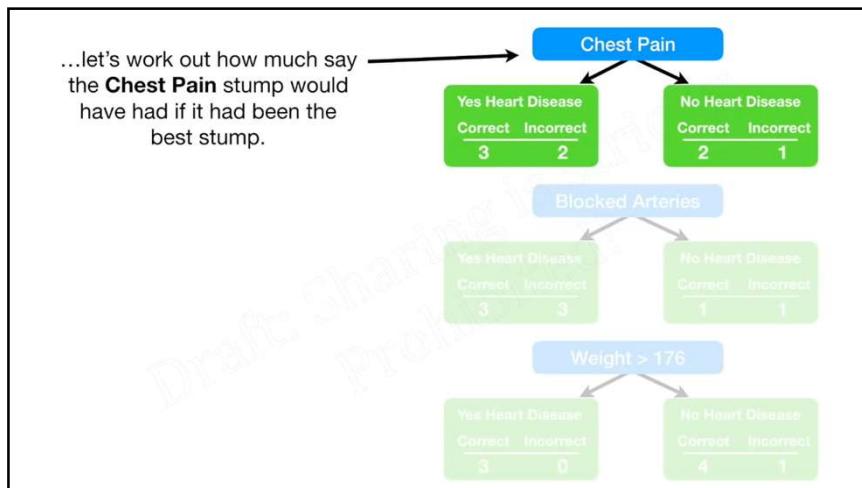
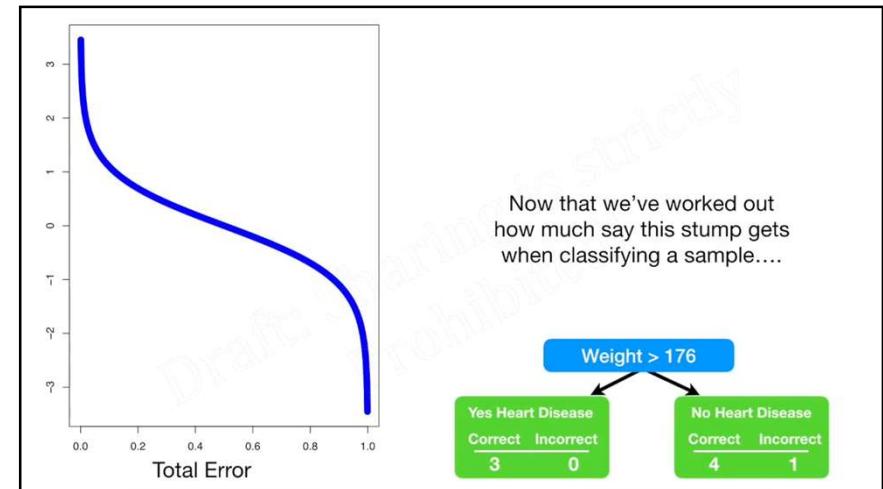
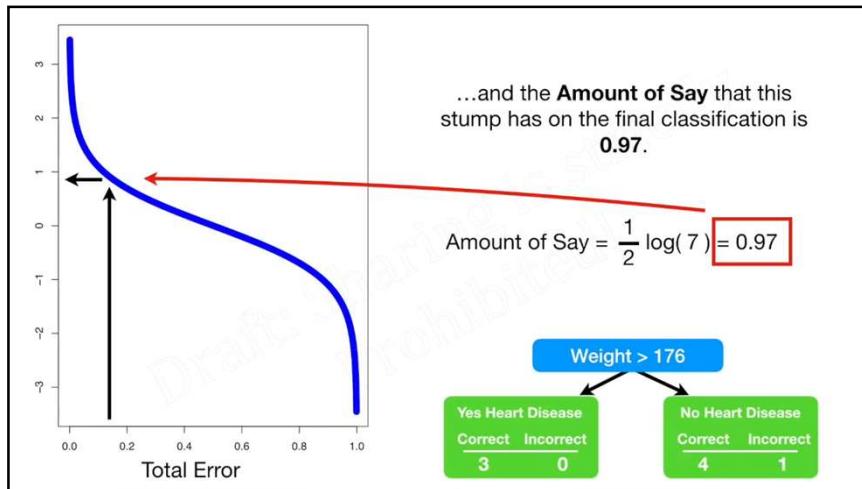
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

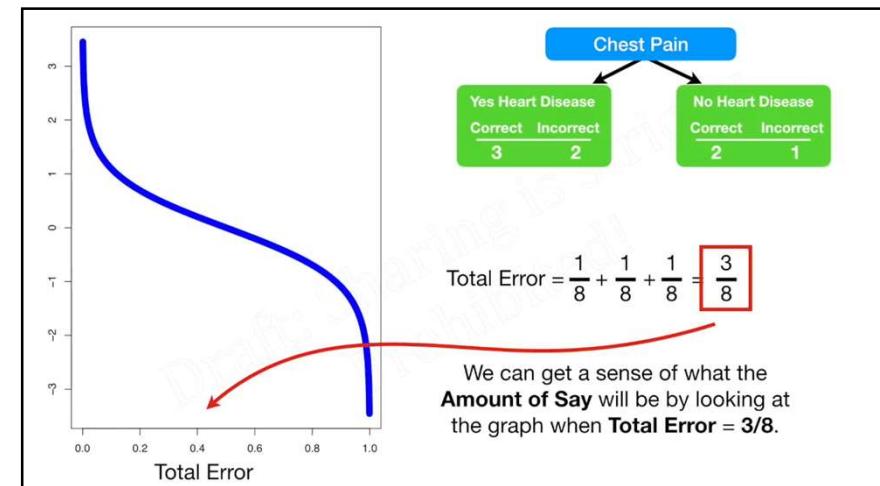
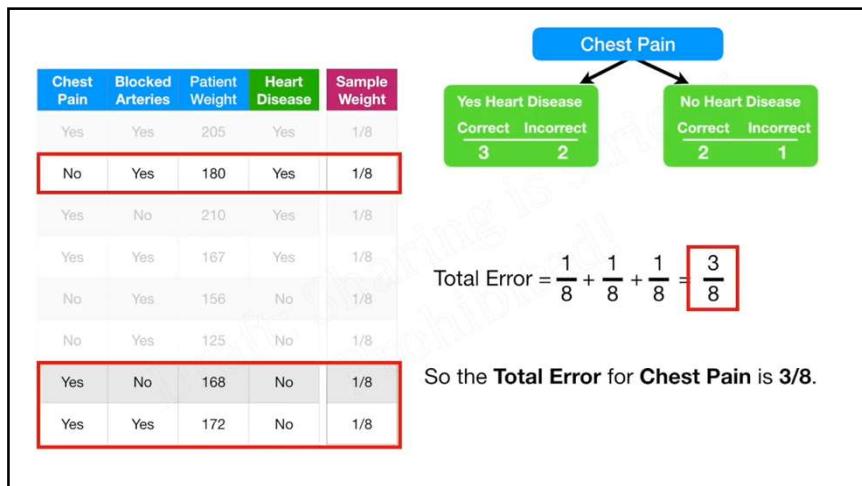
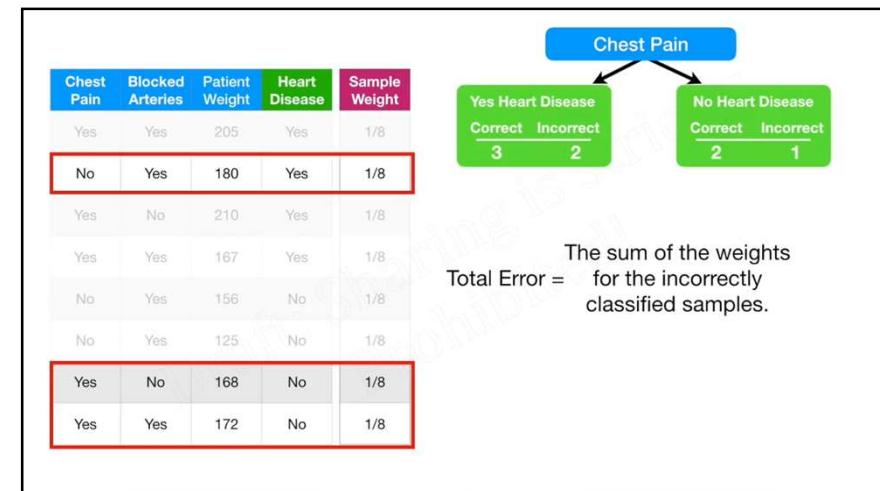
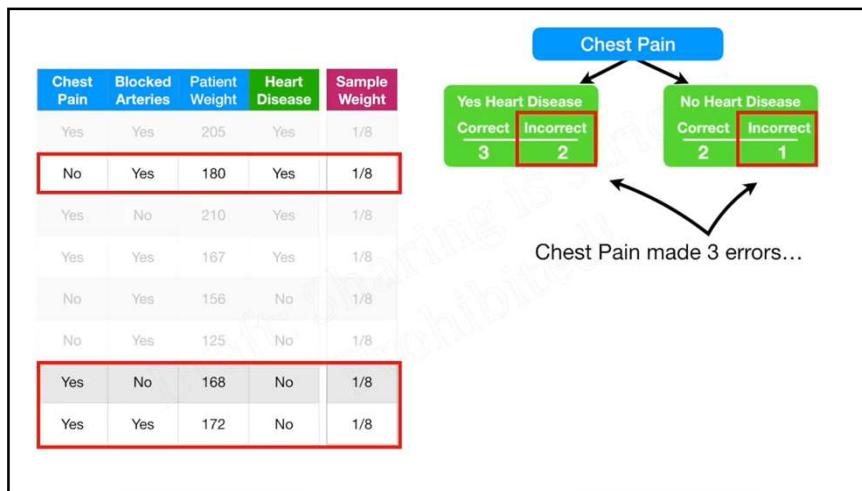
With **Patient Weight > 176**, the **Total Error** is **1/8**, so we just plug and chug...

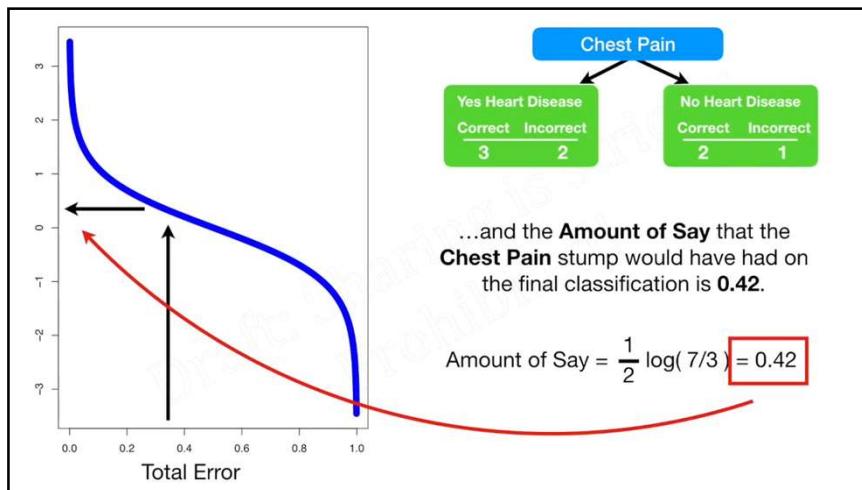
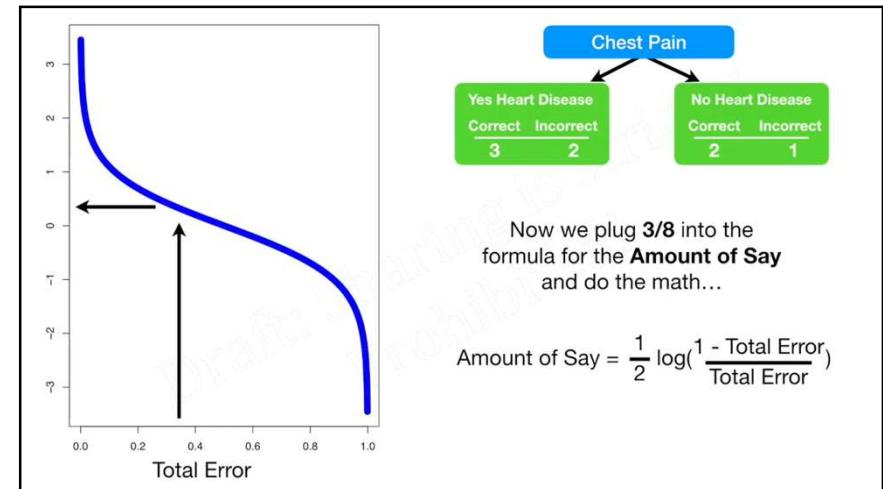
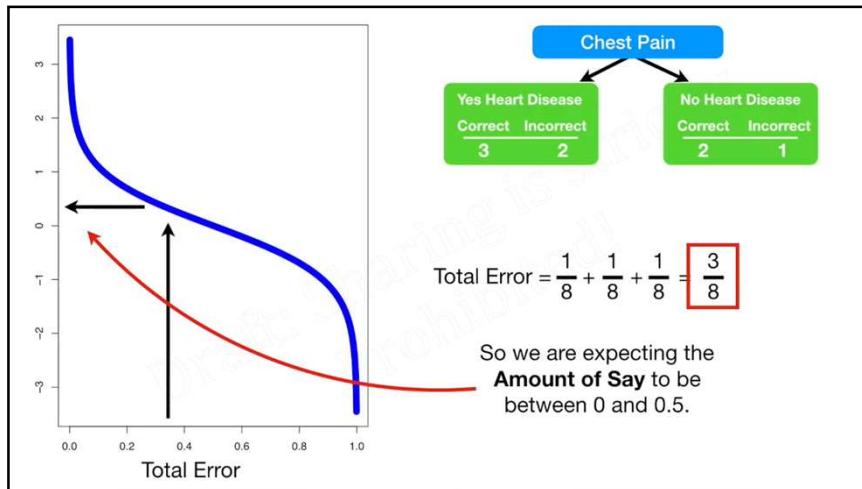
$$\text{Amount of Say} = \frac{1}{2} \log\left(\frac{1 - \text{Total Error}}{\text{Total Error}}\right)$$

```

graph TD
    A[Weight > 176] --> B[Yes Heart Disease  
Correct | Incorrect  
3 | 0]
    A --> C[No Heart Disease  
Correct | Incorrect  
4 | 1]
  
```







Now we know how the **Sample Weights** for the *incorrectly* classified samples are used to determine the **Amount of Say** each stump gets.

A diagram of a decision tree with three nodes. The top node is blue and labeled 'Chest Pain'. It branches into two green nodes: 'Yes Heart Disease' and 'No Heart Disease'. Each green node further branches into two green leaf nodes: 'Correct' and 'Incorrect'. Arrows indicate the flow from the top node through the branches to the leaf nodes.

Now we need to learn how to modify the weights so that the next stump will take the errors that the current stump made into account.



Now we need to learn how to modify the weights so that the next stump will take the errors that the current stump made into account.



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

Let's go back to the first stump that we made.

Weight > 176

Yes Heart Disease

Correct Incorrect
3 0

No Heart Disease

Correct Incorrect
4 1

When we created this stump, all of the **Sample Weights** were the same...

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

Weight > 176

Yes Heart Disease

Correct Incorrect
3 0

No Heart Disease

Correct Incorrect
4 1



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

Let's start by increasing the **Sample Weight** for the *incorrectly classified sample*.

Weight > 176

Yes Heart Disease	
Correct	Incorrect
3	0

No Heart Disease	
Correct	Incorrect
4	1

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

New Sample = sample weight $\times e^{\text{amount of say Weight}}$

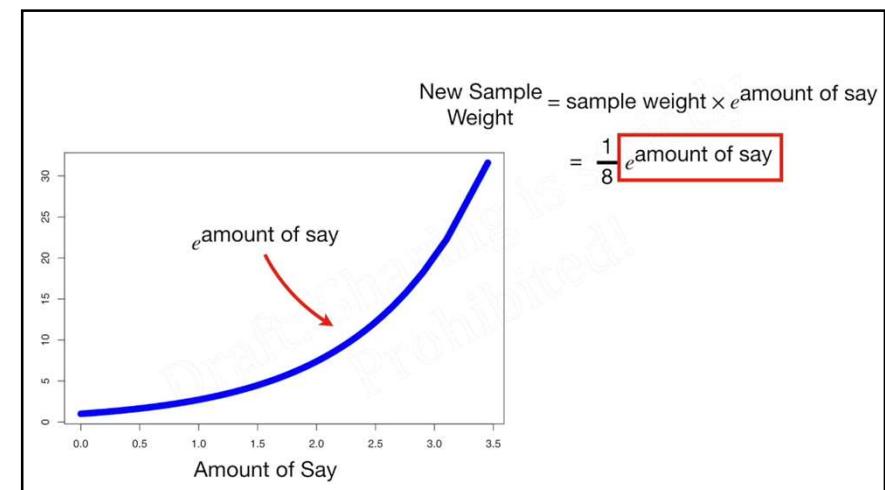
This is the formula we will use to *increase* the **Sample Weight** for the sample that was *incorrectly classified*.

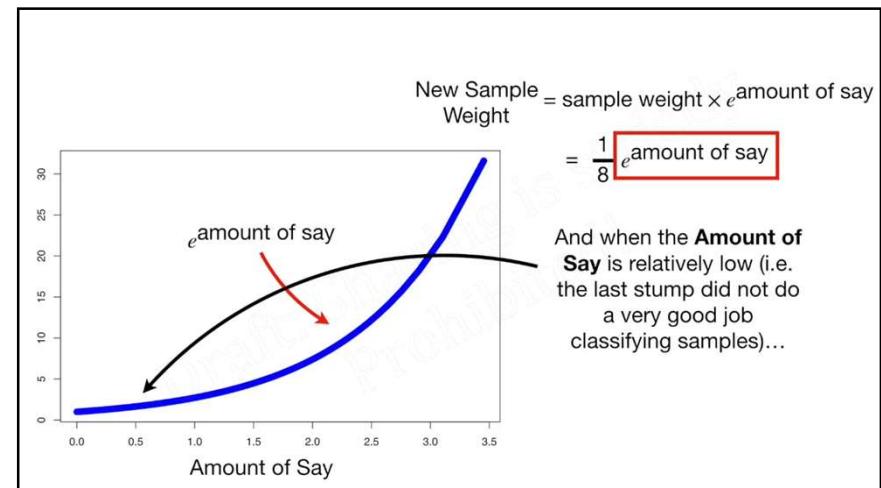
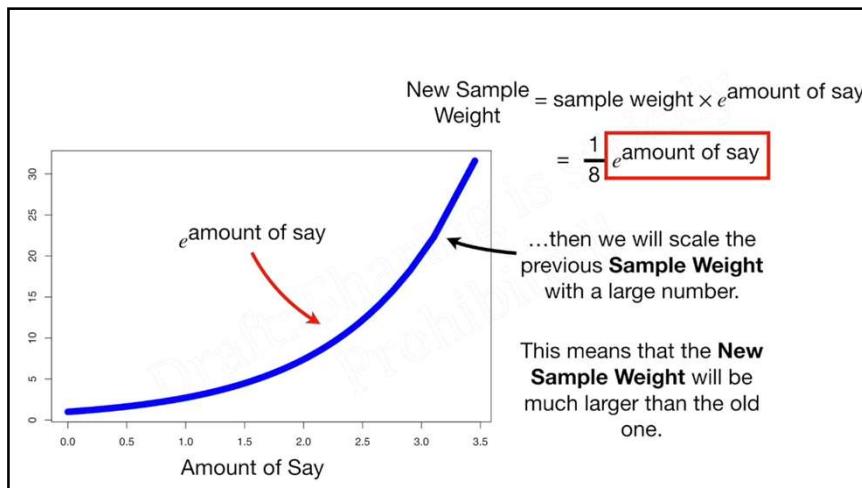
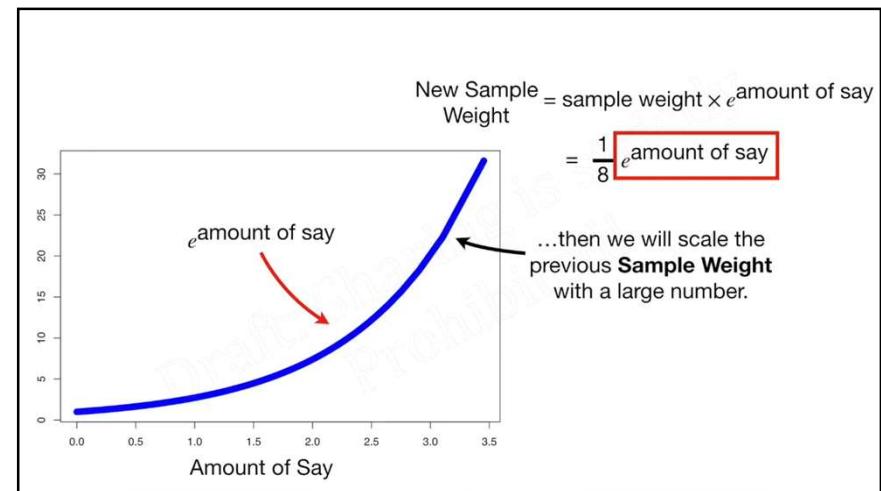
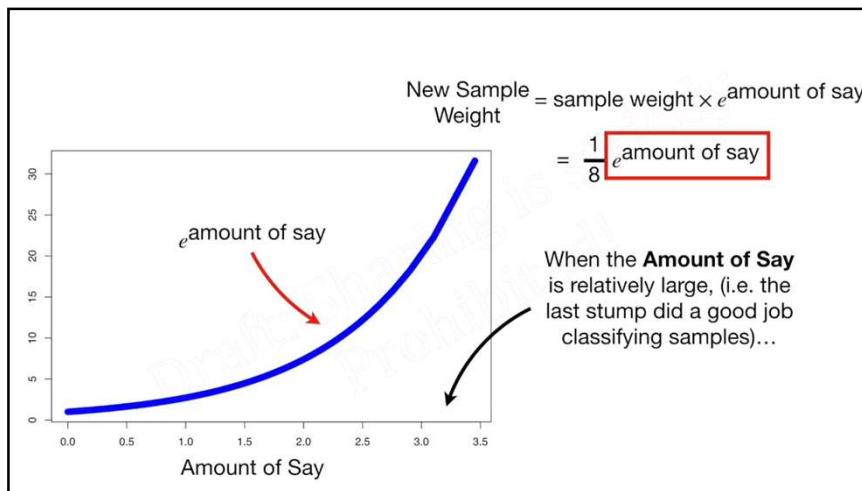
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

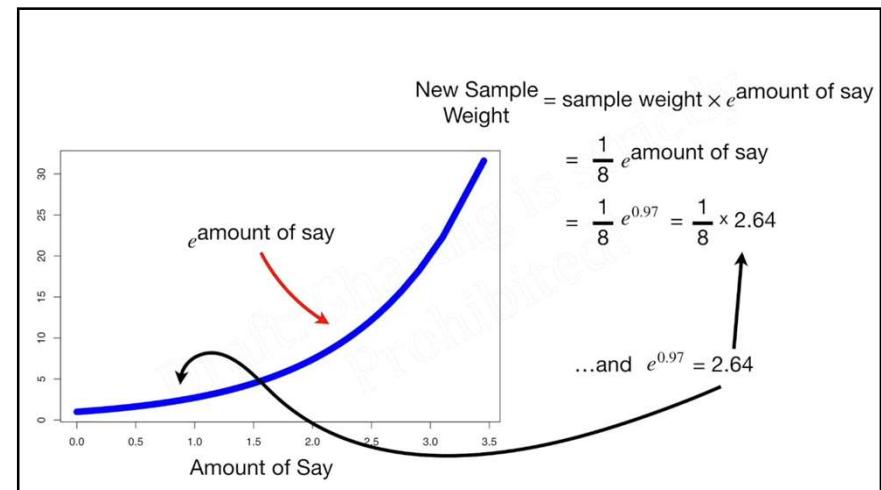
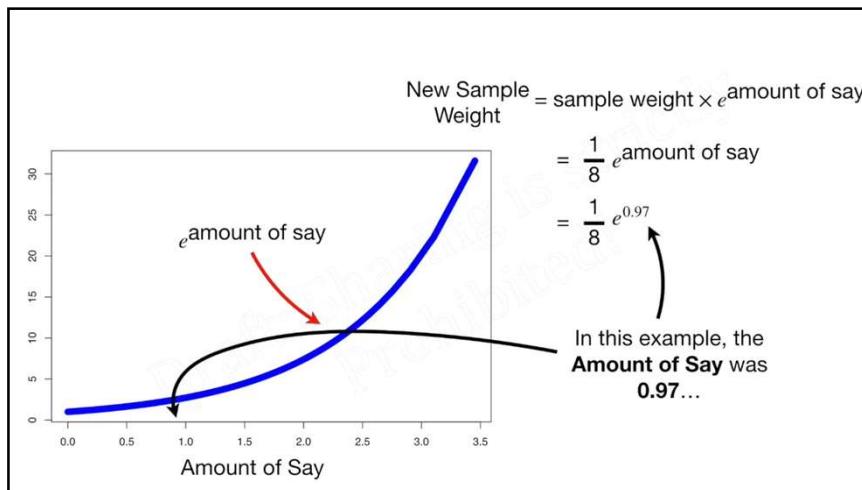
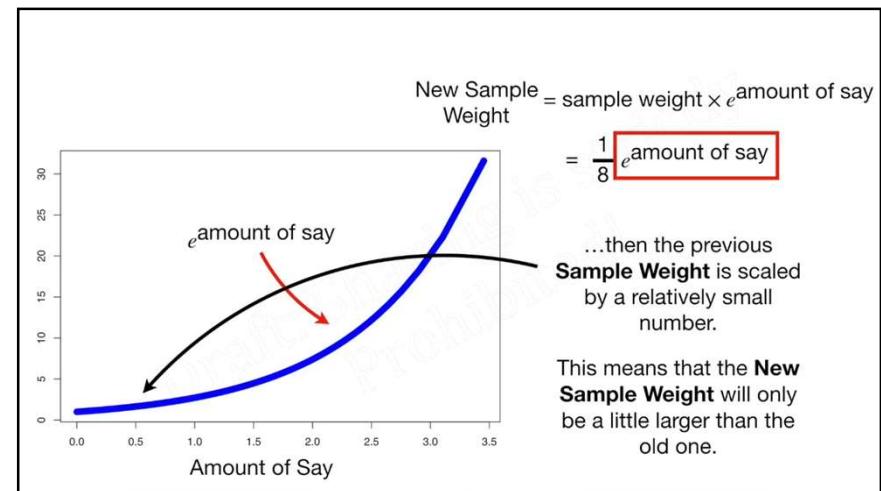
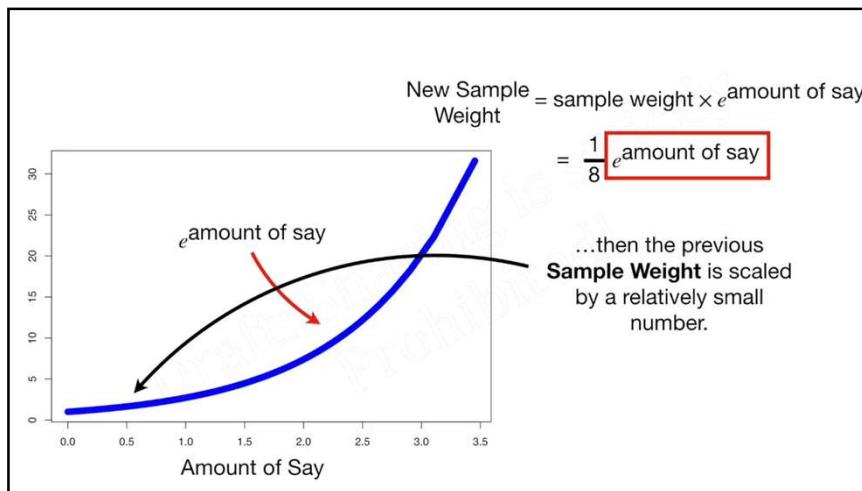
New Sample = sample weight $\times e^{\text{amount of say Weight}}$

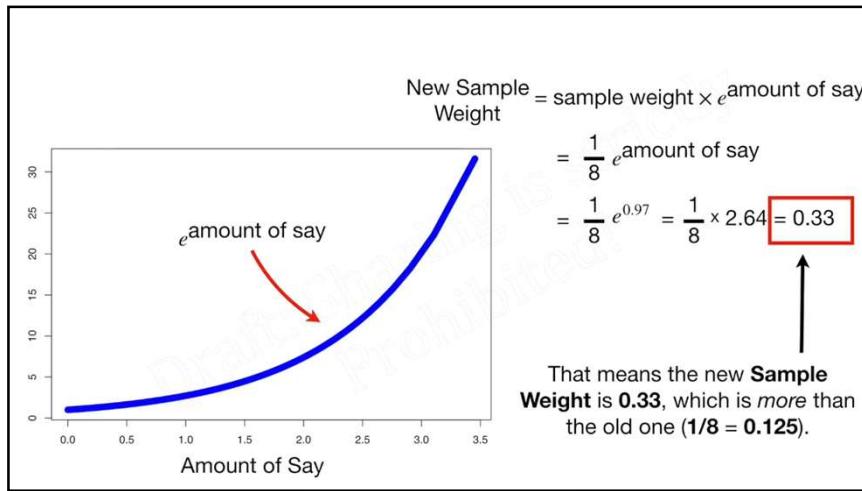
$= \frac{1}{8} e^{\text{amount of say}}$

We plug in the **Sample Weight** from the last stump...









Now we need to *decrease* the **Sample Weights** for all of the *correctly classified samples*.

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

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

New Sample Weight = sample weight $\times e^{-\text{amount of say}}$

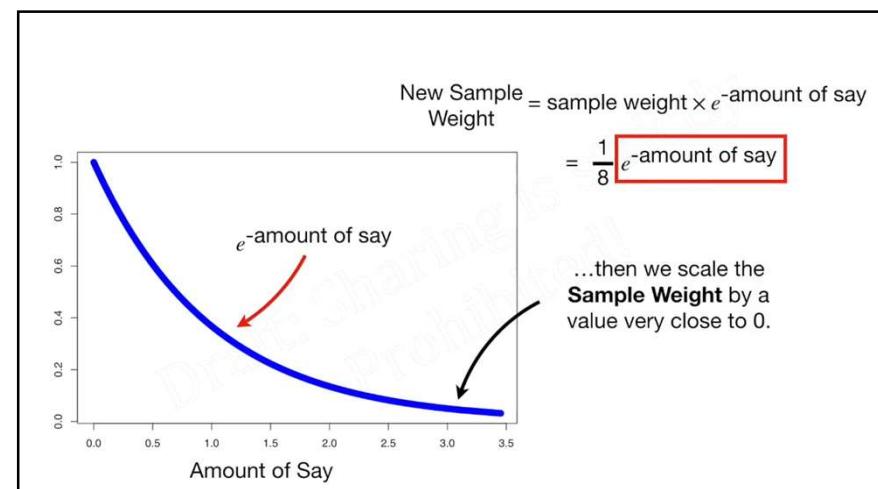
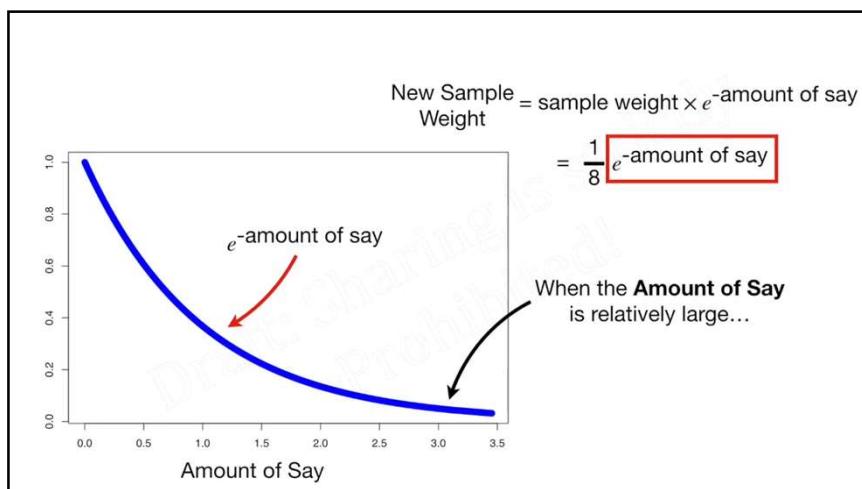
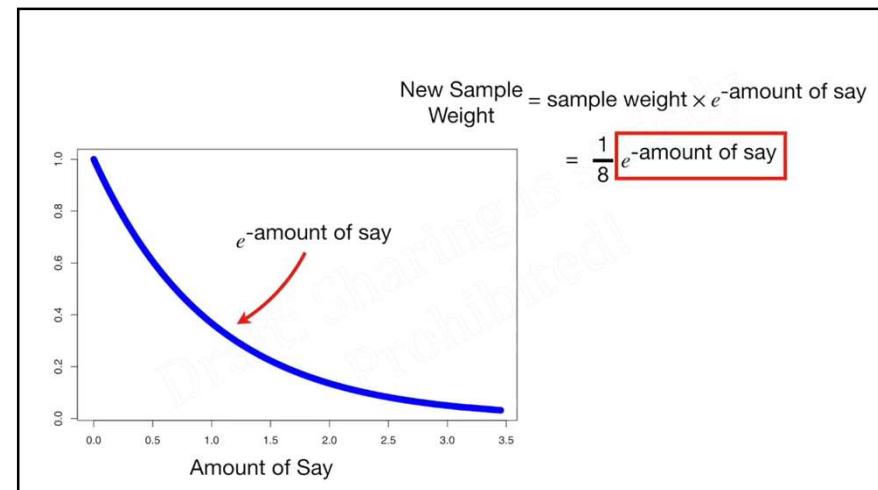
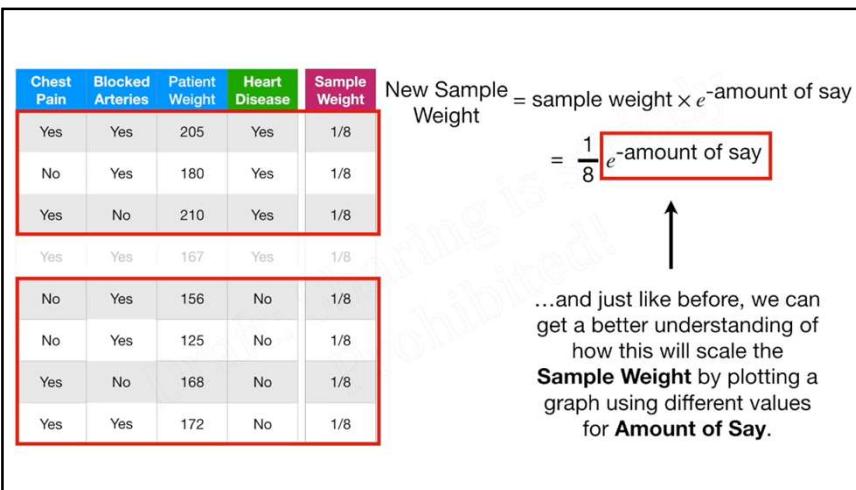
This is the formula we will use to *decrease* the **Sample Weights**.

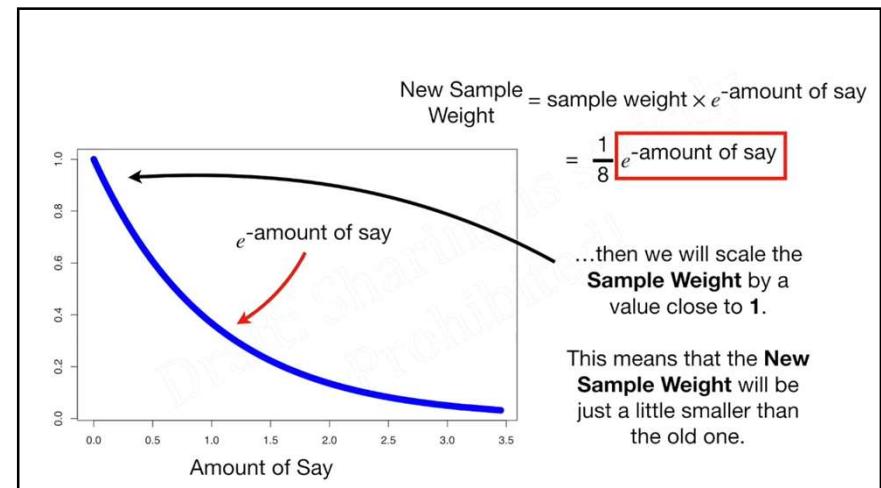
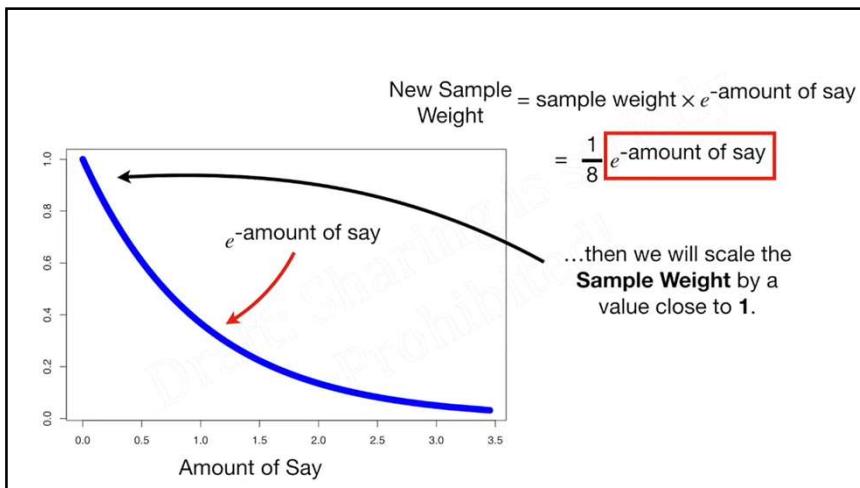
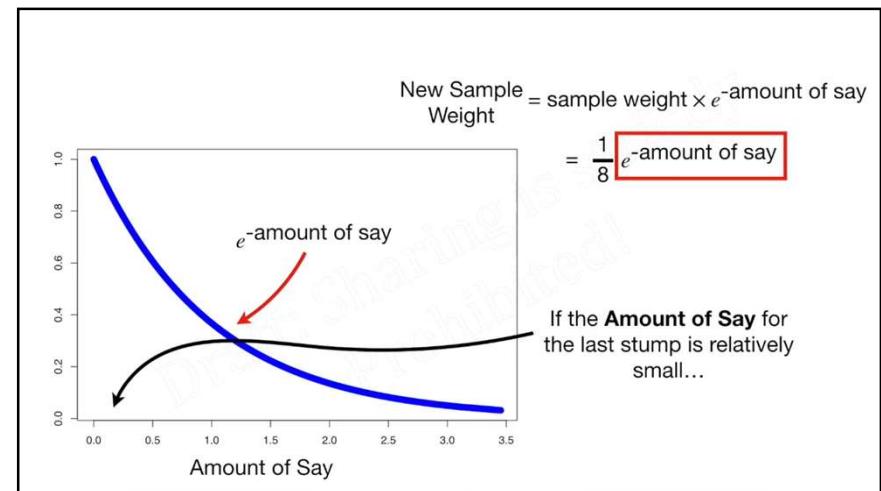
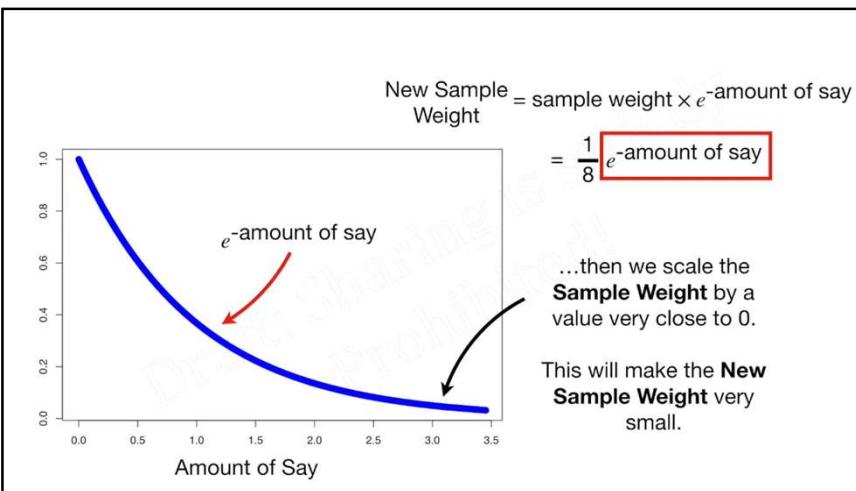
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

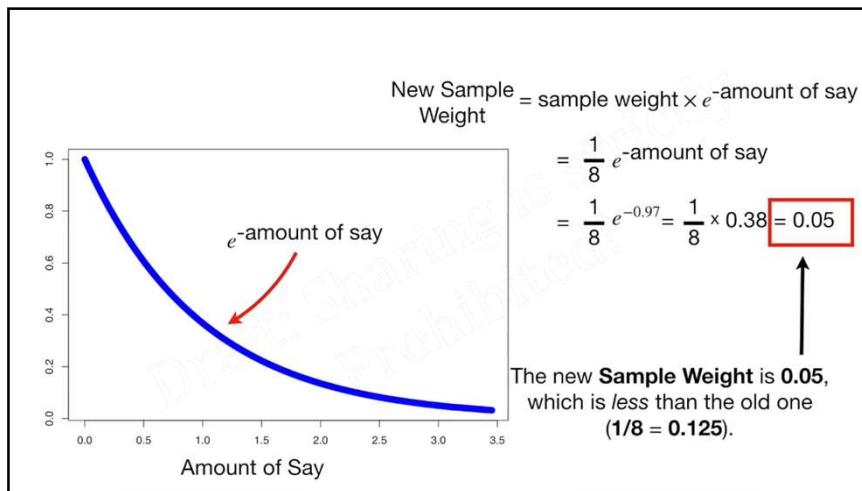
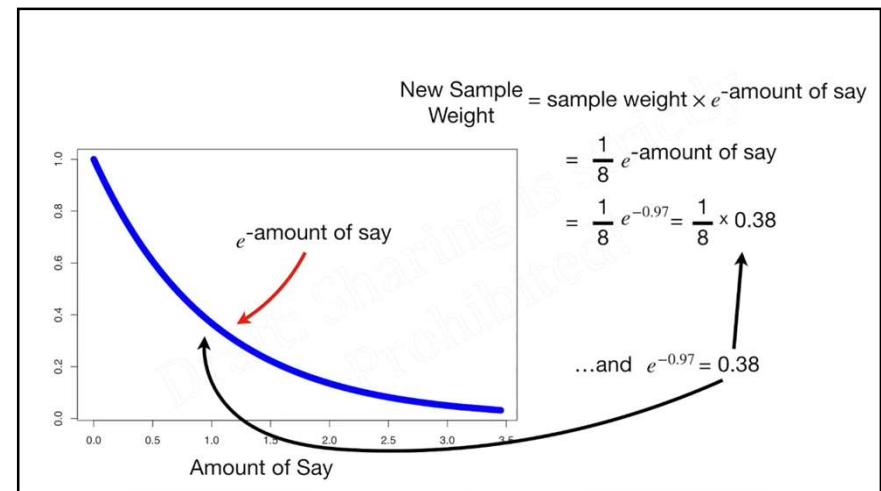
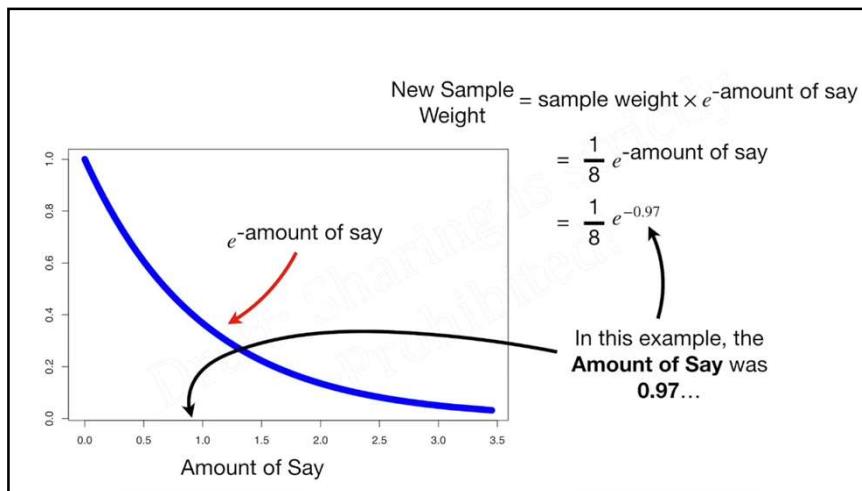
New Sample Weight = sample weight $\times e^{-\text{amount of say}}$

$$= \frac{1}{8} e^{-\text{amount of say}}$$

Just like before, we plug in the **Sample Weight**...







We will keep track of the New Sample Weights in this column...

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight	New 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	

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight	New 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	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	

We plug in **0.33** for the sample that was *incorrectly classified...*

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight	New Weight
Yes	Yes	205	Yes	1/8	0.05
No	Yes	180	Yes	1/8	0.05
Yes	No	210	Yes	1/8	0.05
Yes	Yes	167	Yes	1/8	0.33
No	Yes	156	No	1/8	0.05
No	Yes	125	No	1/8	0.05
Yes	No	168	No	1/8	0.05
Yes	Yes	172	No	1/8	0.05

All of the other samples get **0.05**.

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight	New Weight
Yes	Yes	205	Yes	1/8	0.05
No	Yes	180	Yes	1/8	0.05
Yes	No	210	Yes	1/8	0.05
Yes	Yes	167	Yes	1/8	0.33
No	Yes	156	No	1/8	0.05
No	Yes	125	No	1/8	0.05
Yes	No	168	No	1/8	0.05
Yes	Yes	172	No	1/8	0.05

Now we need to normalize the **New Sample Weights** so that they will add up to 1.

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight	New Weight
Yes	Yes	205	Yes	1/8	0.05
No	Yes	180	Yes	1/8	0.05
Yes	No	210	Yes	1/8	0.05
Yes	Yes	167	Yes	1/8	0.33
No	Yes	156	No	1/8	0.05
No	Yes	125	No	1/8	0.05
Yes	No	168	No	1/8	0.05
Yes	Yes	172	No	1/8	0.05

Right now, if you add up the **New Sample Weights**, you get **0.68**.

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight	New Weight	Norm. Weight
Yes	Yes	205	Yes	1/8	0.05	0.07
No	Yes	180	Yes	1/8	0.05	0.07
Yes	No	210	Yes	1/8	0.05	0.07
Yes	Yes	167	Yes	1/8	0.33	0.49
No	Yes	156	No	1/8	0.05	0.07
No	Yes	125	No	1/8	0.05	0.07
Yes	No	168	No	1/8	0.05	0.07
Yes	Yes	172	No	1/8	0.05	0.07

So we divide each **New Sample Weight** by **0.68** to get the normalized values.

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight	New Weight	Norm. Weight
Yes	Yes	205	Yes	1/8	0.05	0.07
No	Yes	180	Yes	1/8	0.05	0.07
Yes	No	210	Yes	1/8	0.05	0.07
Yes	Yes	167	Yes	1/8	0.33	0.49
No	Yes	156	No	1/8	0.05	0.07
No	Yes	125	No	1/8	0.05	0.07
Yes	No	168	No	1/8	0.05	0.07
Yes	Yes	172	No	1/8	0.05	0.07

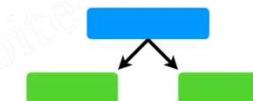
Now, when we add up the **New Sample Weights**, we get **1** (plus or minus a little rounding error).

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

Now we just transfer the **Normalized Sample Weights** to the **Sample Weights** column, since those are what we will use for the next stump.

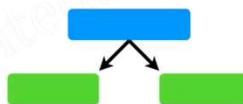
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Yes	Yes	205	Yes	0.07
No	Yes	180	Yes	0.07
Yes	No	210	Yes	0.07
Yes	Yes	167	Yes	0.49
No	Yes	156	No	0.07
No	Yes	125	No	0.07
Yes	No	168	No	0.07
Yes	Yes	172	No	0.07

Now we can use the modified **Sample Weights** to make the second **stump** in the forest.



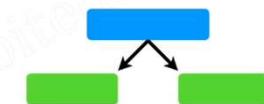
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Yes	Yes	205	Yes	0.07
No	Yes	180	Yes	0.07
Yes	No	210	Yes	0.07
Yes	Yes	167	Yes	0.49
No	Yes	156	No	0.07
No	Yes	125	No	0.07
Yes	No	168	No	0.07
Yes	Yes	172	No	0.07

In theory, we could use the **Sample Weights** to calculate **Weighted Gini Indexes** to determine which variable should split the next stump.



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

The **Weighted Gini Index** would put more emphasis on correctly classifying this sample (the one that was misclassified by the last stump), since this sample has the largest **Sample Weight**.



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

Alternatively, instead of using a **Weighted Gini Index**, we can make a new collection of samples that contains duplicate copies of the samples with the largest **Sample Weights**.

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	Yes	167		
No	Yes	156		
No	Yes	125	No	0.07
Yes	No	168	No	0.07
Yes	Yes	172	No	0.07

So we start by making a new, but empty, dataset that is the same size as the original...

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight	Chest Pain	Blocked Arteries	Patient Weight	Heart Disease
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				

Then we pick a random number between 0 and 1...

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight	Chest Pain	Blocked Arteries	Patient Weight	Heart Disease
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				

...and we see where that number falls when we use the **Sample Weights** like a distribution.

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight	Chest Pain	Blocked Arteries	Patient Weight	Heart Disease
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				

If the number is between 0 and 0.07, then we would put this sample into the new collection of samples...

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight	Chest Pain	Blocked Arteries	Patient Weight	Heart Disease
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				

...and if the number is between 0.07 and 0.14 ($0.07 + 0.07 = 0.14$), then we would put this sample into the new collection of samples...

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

...and if the number is between **0.14** and **0.21**
(**0.14 + 0.07 = 0.21**), then we would put this sample into the new collection of samples...

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

...and if the number is between **0.21** and **0.70**
(**0.21 + 0.49 = 0.70**), then we would put this sample into the new collection of samples...

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

← etc.

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

For example, imagine the first number I picked was **0.72**...

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

...then I would put this sample into my new collection of samples...

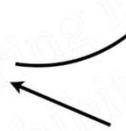


Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Yes	Yes	205	Yes	0.07
No	Yes	180	Yes	0.07
Yes	No	210	Yes	0.07
Yes	Yes	167	Yes	0.49
No	Yes	156	No	0.07
No	Yes	125	No	0.07
Yes	No	168	No	0.07
Yes	Yes	172	No	0.07

Then I pick another random number and get **0.42...**

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

...and I would put this sample into my new collection of samples...



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

Then I pick **0.83...**

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

...and I would put this sample into my new collection of samples...

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease
No	Yes	156	No
Yes	Yes	167	Yes
No	Yes	125	No

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

...and I would put this sample into my new collection of samples...

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	0.07
No	Yes	156	No	0.07
No	Yes	125	No	0.07
Yes	No	168	No	0.07
Yes	Yes	172	No	0.07

NOTE: This is the second time that we have added this particular sample to the new collection of samples.

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease
No	Yes	156	No
Yes	Yes	167	Yes
No	Yes	125	No

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

We then continue to pick random numbers and add samples to the new collection until the new collection is the same size as the original.

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		
Yes	No	168		
Yes	Yes	172		

Ultimately, this sample was added to the new collection of samples 4 times, reflecting its larger **Sample Weight**.

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

Now we get rid of the original samples...

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

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

...and use the new collection of samples.

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

Lastly, we give all the samples equal **Sample Weights**, just like before.

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

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

However, that doesn't mean the next stump will not emphasize the need to correctly classify these samples.

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

Because these samples are all the same, they will be treated as a block, creating a large penalty for being misclassified.

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

Now we go back to the beginning and try to find the stump that does the best job classifying the new collection of samples.

Chest Pain



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

Now we go back to the beginning and try to find the stump that does the best job classifying the new collection of samples.

Chest Pain



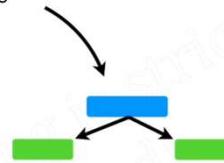
Blocked Arteries



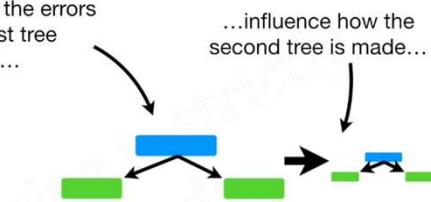
Patient Weight



So that is how the errors
that the first tree
makes...

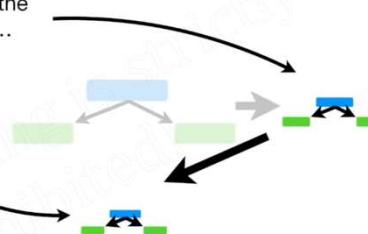


So that is how the errors
that the first tree
makes...

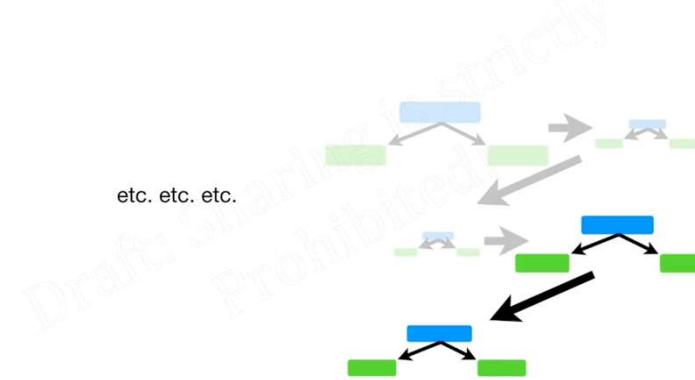


...and the errors that the
second tree makes...

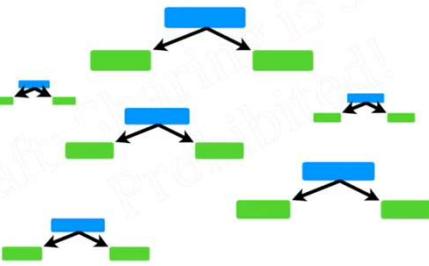
...influence how the
third tree is made.



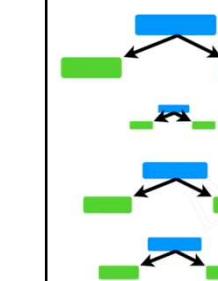
etc. etc. etc.



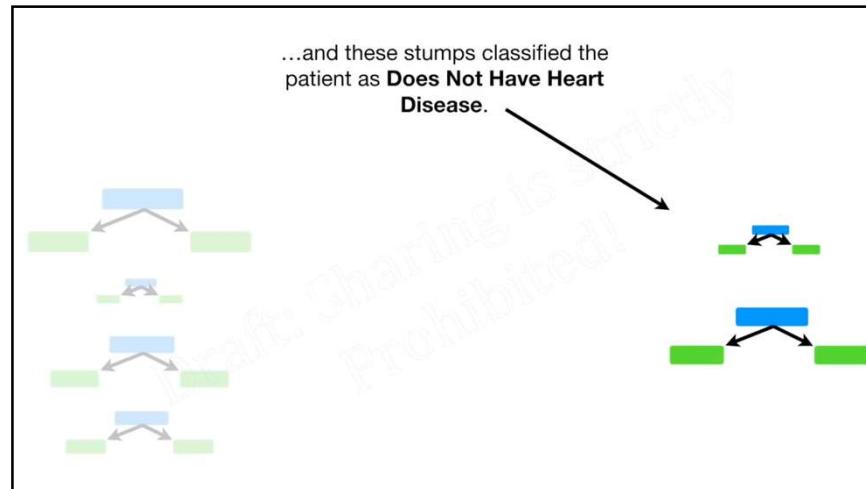
Now we need to talk about how a forest of stumps created by **AdaBoost** makes classifications...



Imagine that these stumps classified a patient as **Has Heart Disease**...



...and these stumps classified the patient as **Does Not Have Heart Disease**.



These are the **Amounts of Say** for these stumps...

Has Heart Disease

Amount of Say

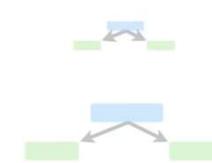
→ 0.97

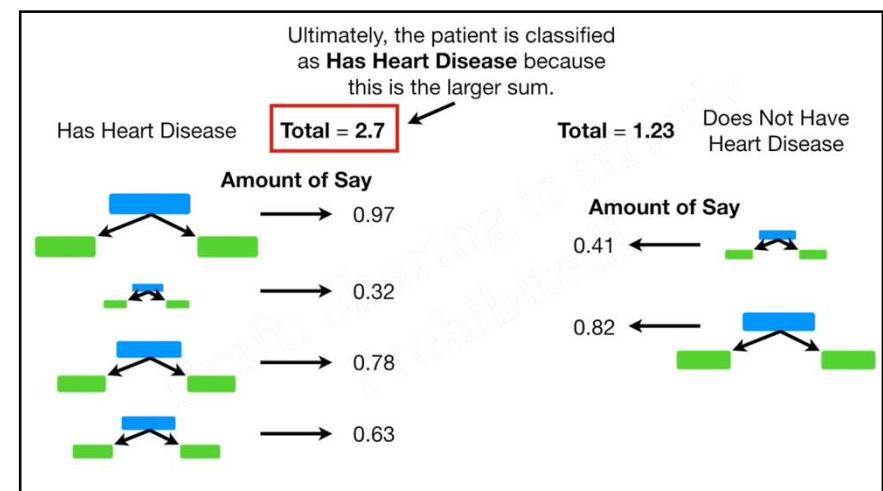
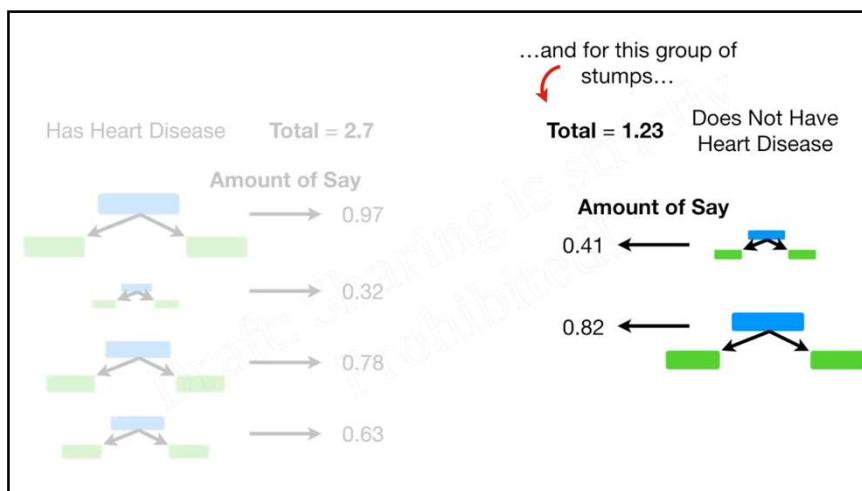
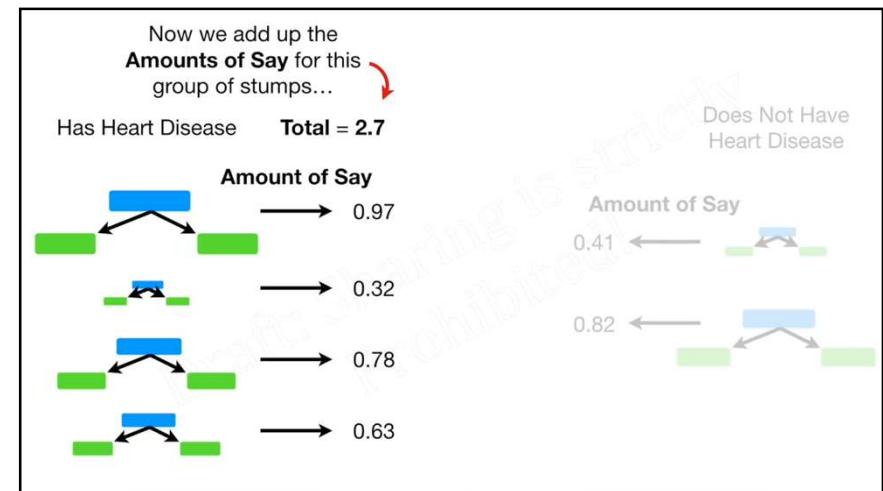
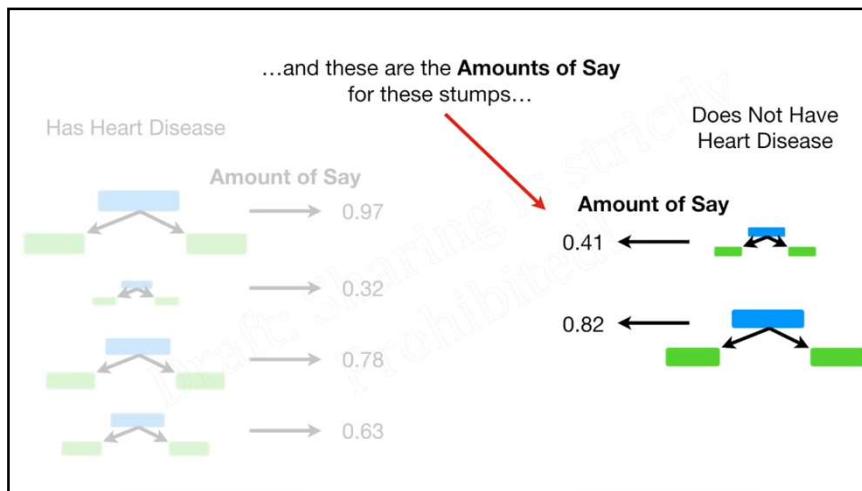
→ 0.32

→ 0.78

→ 0.63

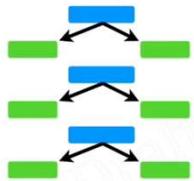
Does Not Have Heart Disease





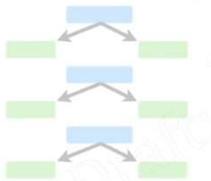
To review, the three ideas behind **AdaBoost** are...

- 1) AdaBoost combines a lot of "weak learners" to make classifications. The weak learners are almost always **stumps**.

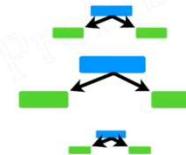


To review, the three ideas behind **AdaBoost** are...

- 1) AdaBoost combines a lot of "weak learners" to make classifications. The weak learners are almost always stumps.

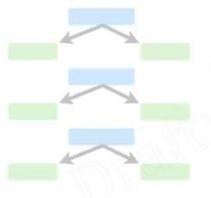


- 2) Some stumps get more say in the classification than others.

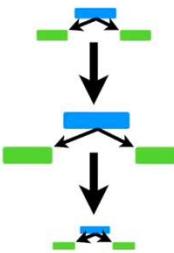


To review, the three ideas behind **AdaBoost** are...

- 1) AdaBoost combines a lot of "weak learners" to make classifications. The weak learners are almost always stumps.



- 3) Each **stump** is made by taking the previous **stump's** mistakes into account.



THANK YOU

StarQuest with Joachim Starmer
https://www.youtube.com/watch?v=LsKxG1dLYA
https://www.youtube.com/watch?v=3CC4N4z3GJc