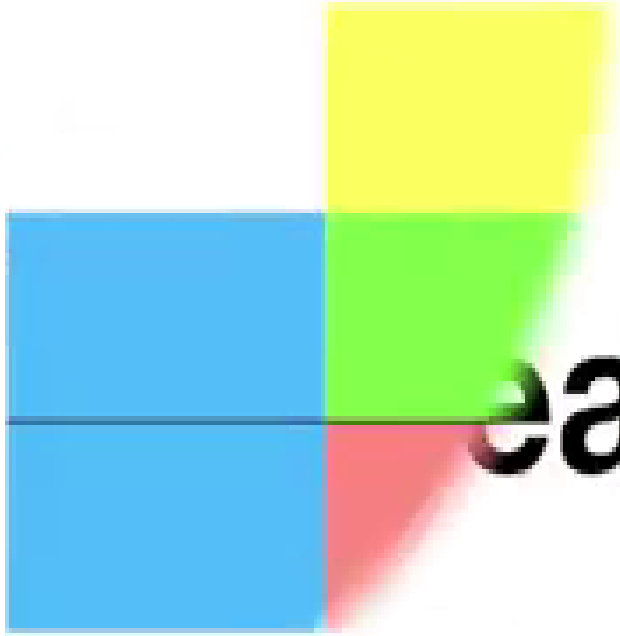
00:00:00

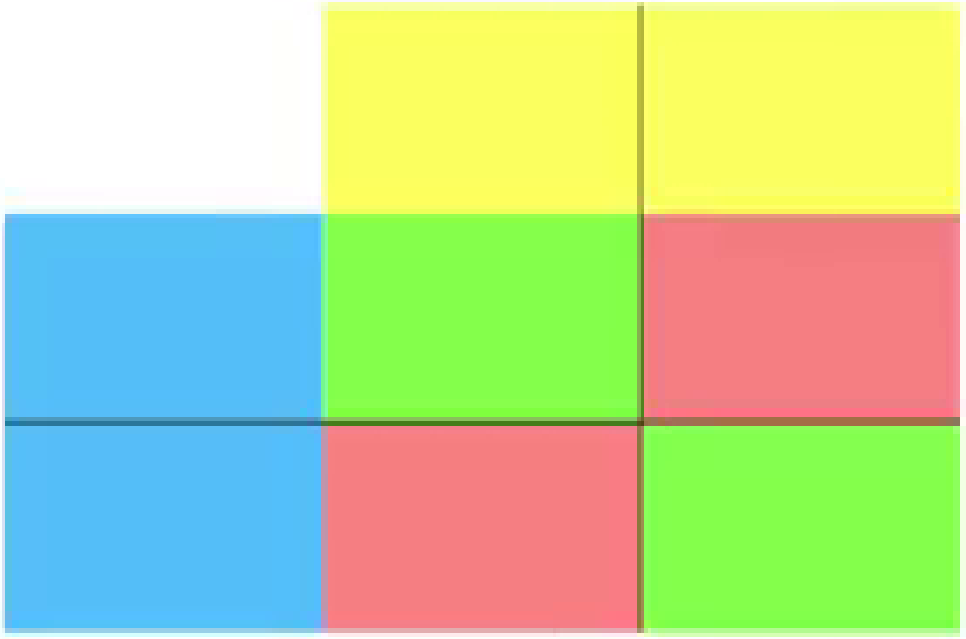
00:00:04

feel confused.



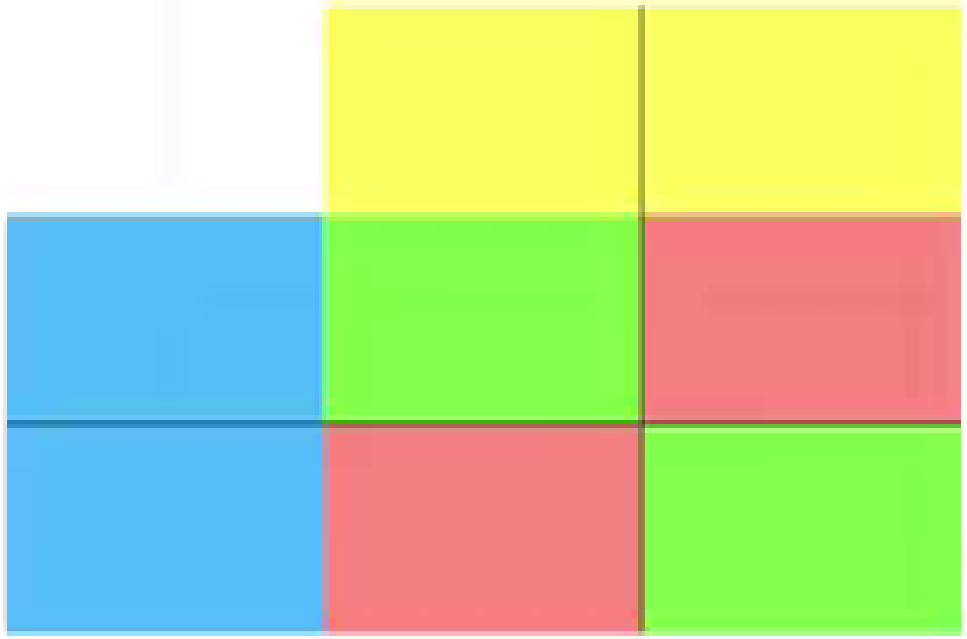
eat it!!!

00:00:07

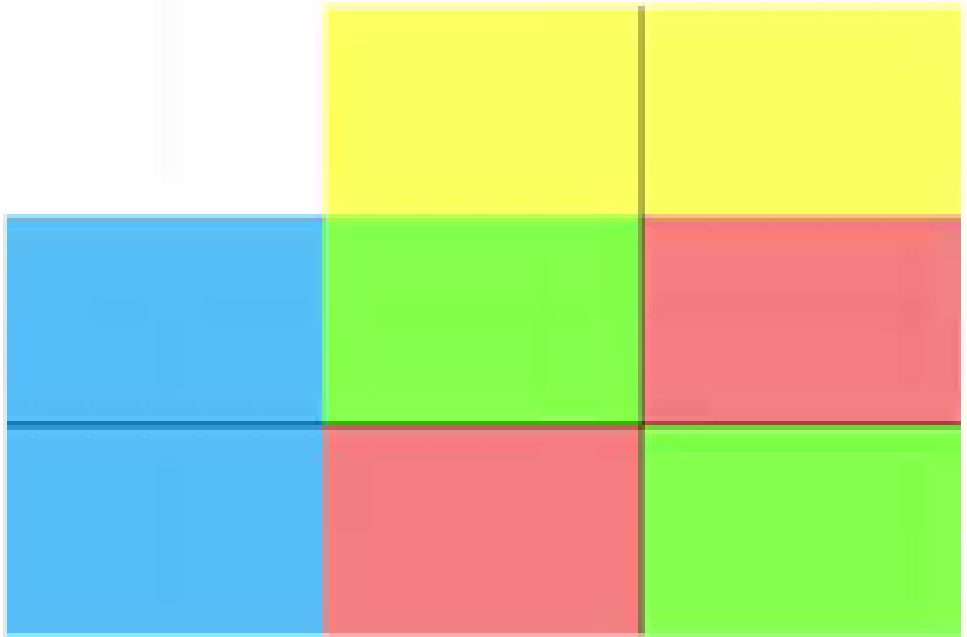


...don't sweat it!!!

00:00:12



...StatQuest is here!!!



00:00:26

Machine Learning Fundamentals: The Confusion Matrix, Clearly Explained!!!

00:00:30

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
...	...	...	...	...

Imagine that we have this medical data...

00:00:37

We've got some clinical measurements…

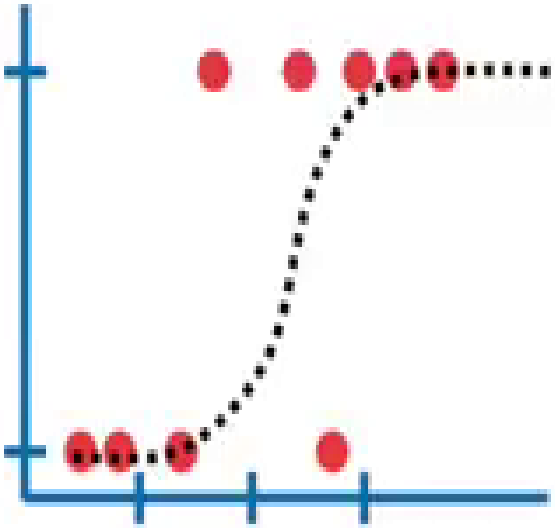
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
…	…	…	…	…

00:00:44

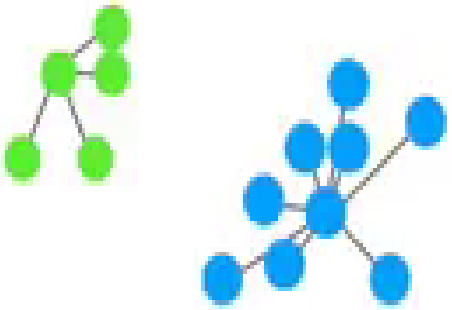
...and we want to apply a machine learning method to them to predict whether or not someone will develop heart disease.

00:01:02

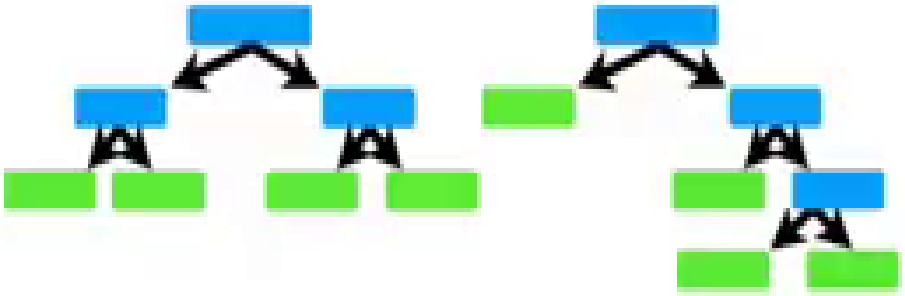
To do this, we could use **Logistic Regression**…



…or **K-Nearest Neighbors**…



…or a **Random Forest**…



…or some other method. There are tons to choose from.

How do we decide which one works best with our data?

00:01:17

We start by dividing the data into **Training** and **Testing** sets…

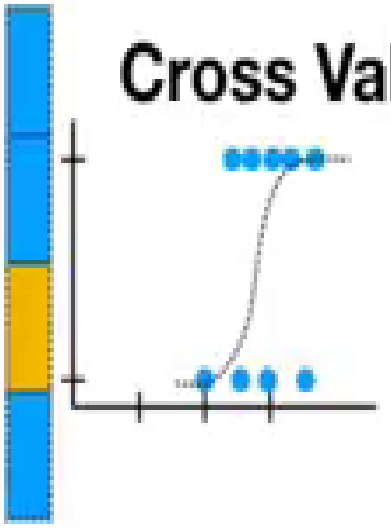
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
…				…

Training Data

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	No	210	No
…	…	…	…	…

Testing Data

NOTE: This would be an excellent opportunity to use **Cross Validation**.

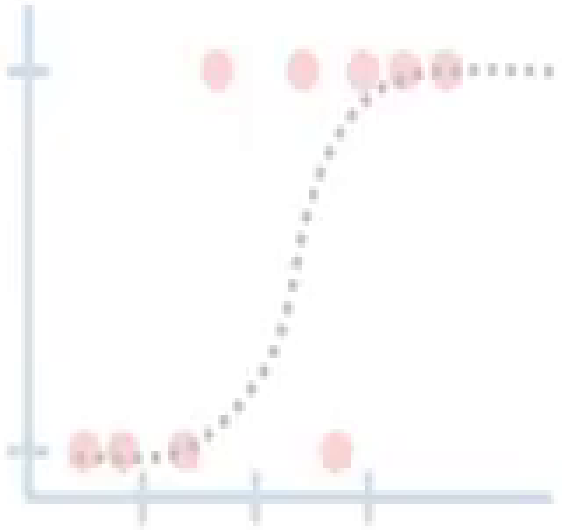


Cross Validation….

…it's no big deal!!!

00:01:23

Then we train all of the methods we're interested in with the **Training** data...

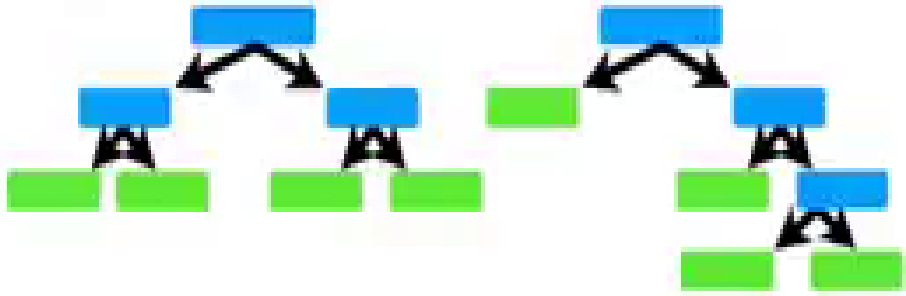


Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
...				...

Training Data

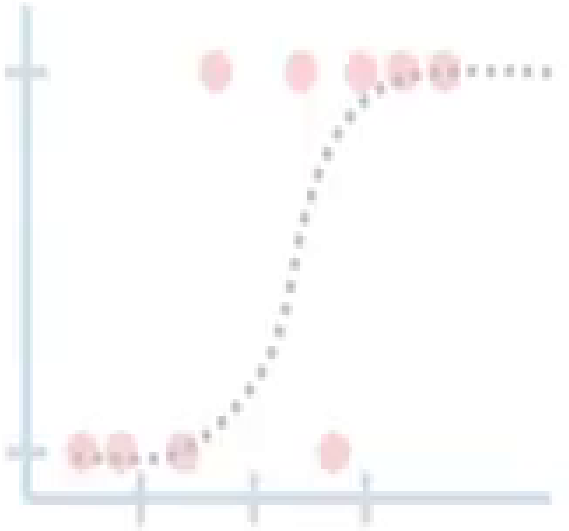
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	No	210	No
...	...	...	...	...

Testing Data



00:01:34

Now we need to summarize how each method performed on the **Testing** data.

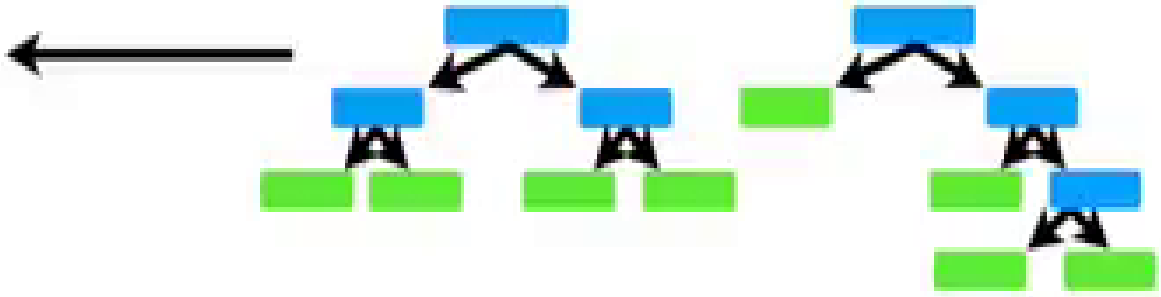


Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
...	...	...	...	...

Training Data

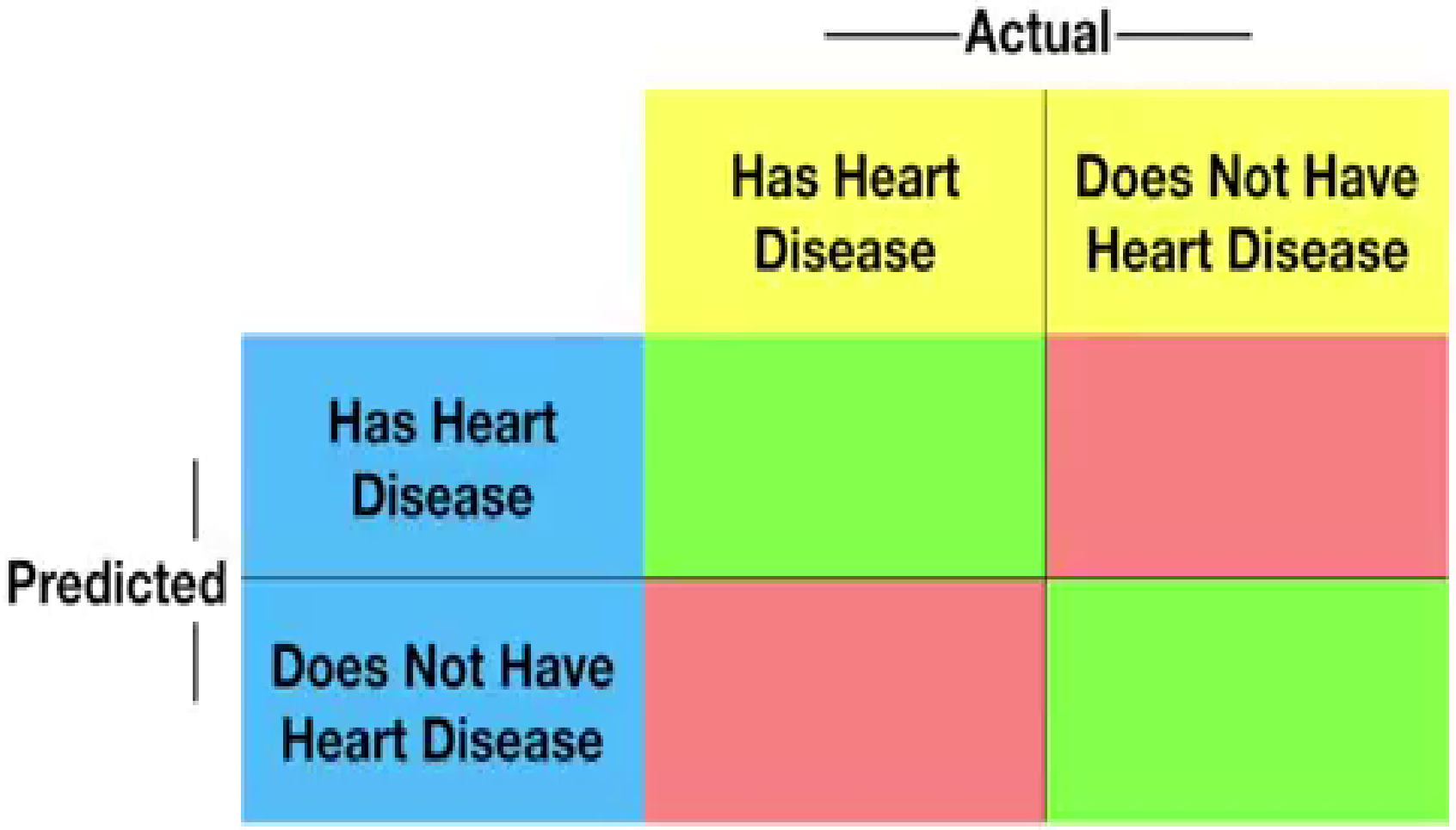
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	No	210	No
...	...	...	...	...

Testing Data

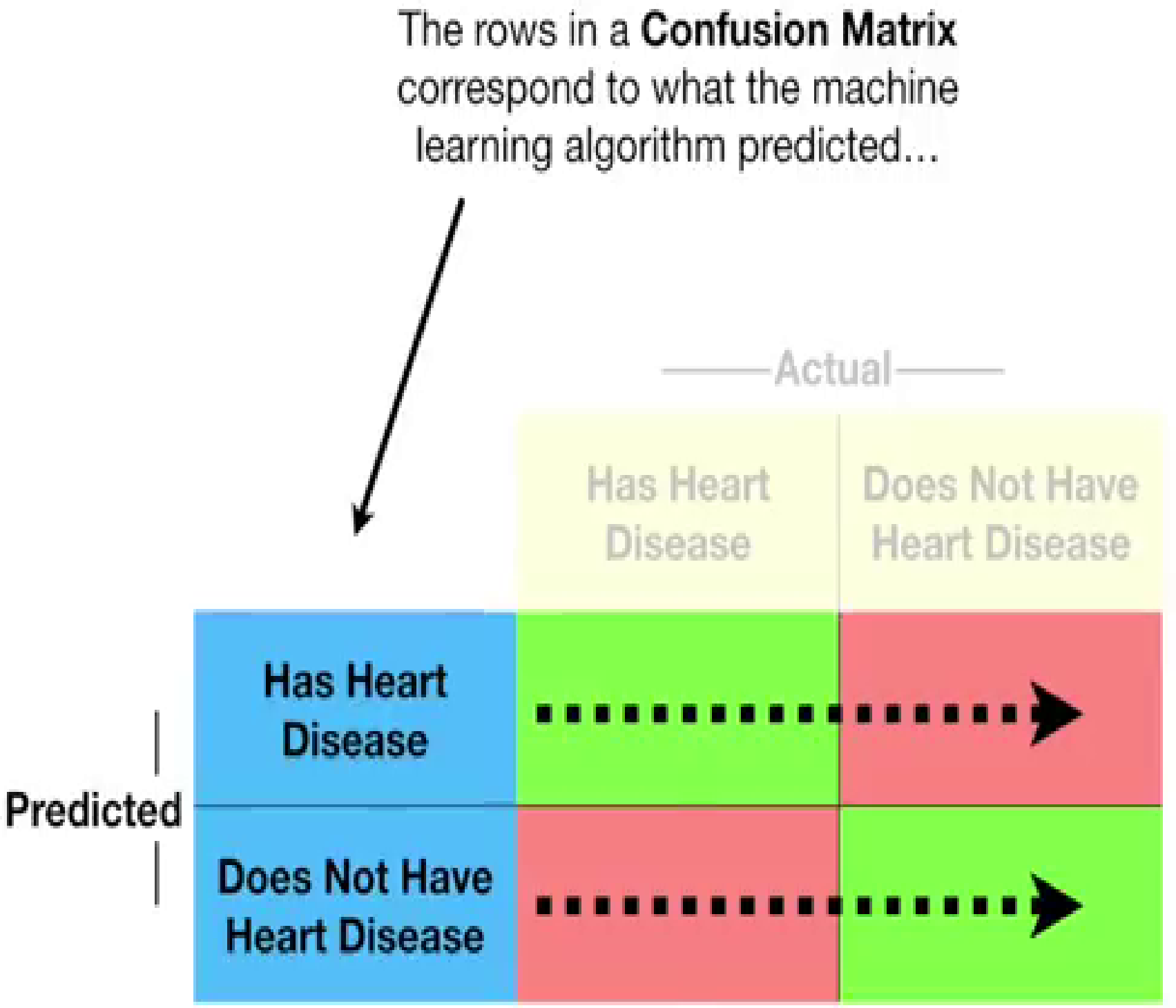


00:01:41

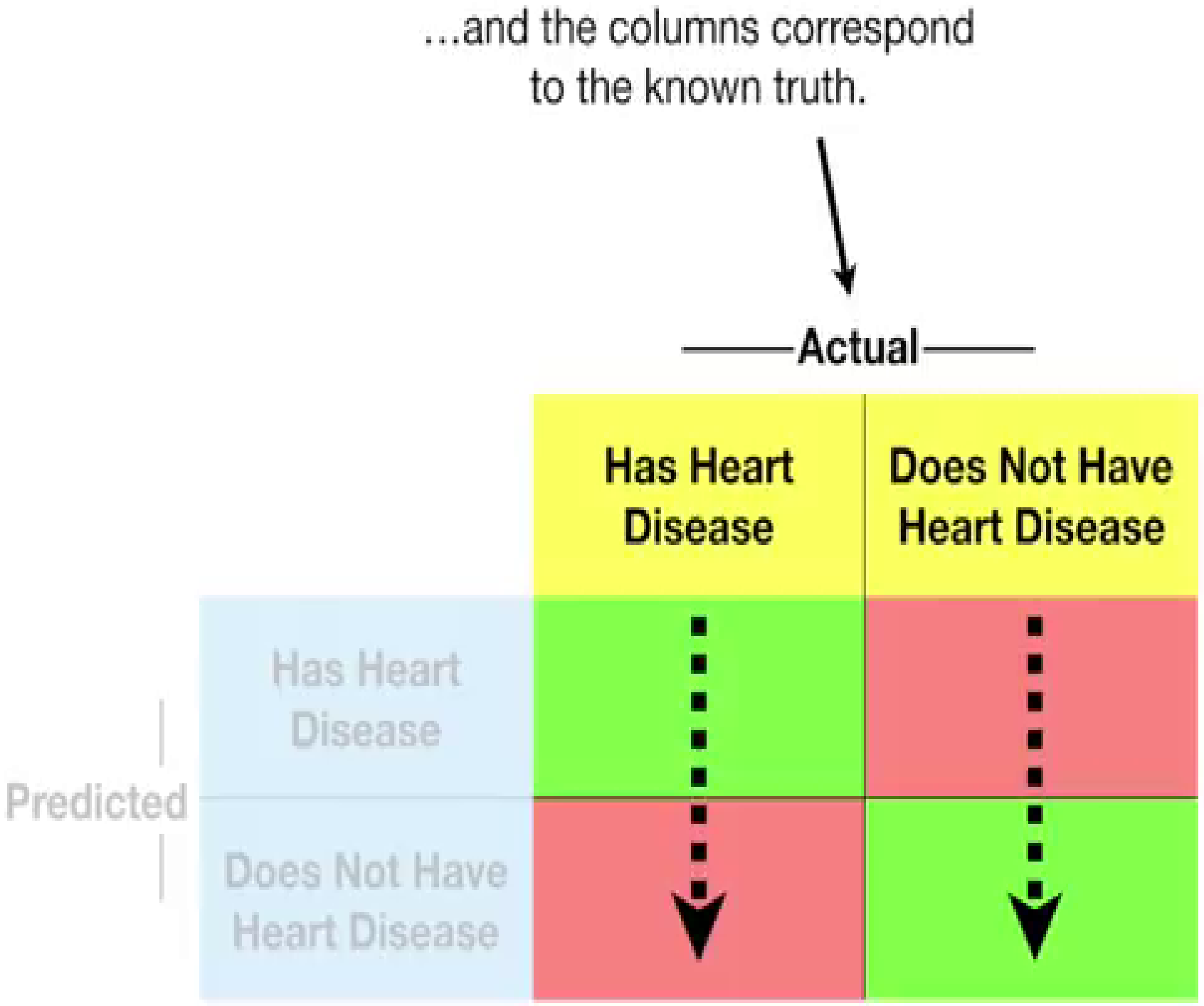
One way to do this is by creating a **Confusion Matrix** for each method.



00:01:47

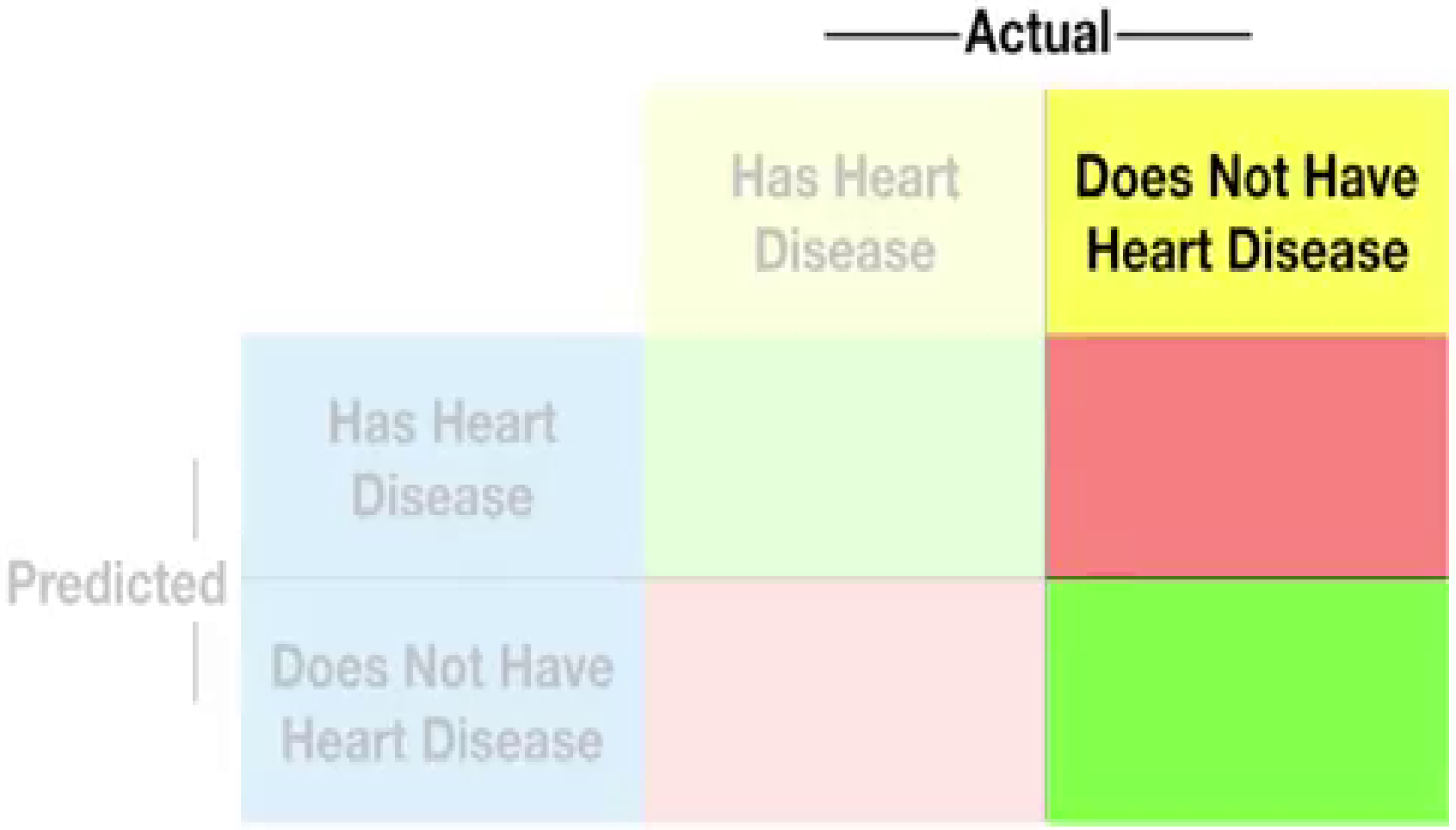

The rows in a **Confusion Matrix** correspond to what the machine learning algorithm predicted...
Actual
Has Heart Disease
Does Not Have Heart Disease
Predicted
Has Heart Disease
Does Not Have Heart Disease

00:01:52

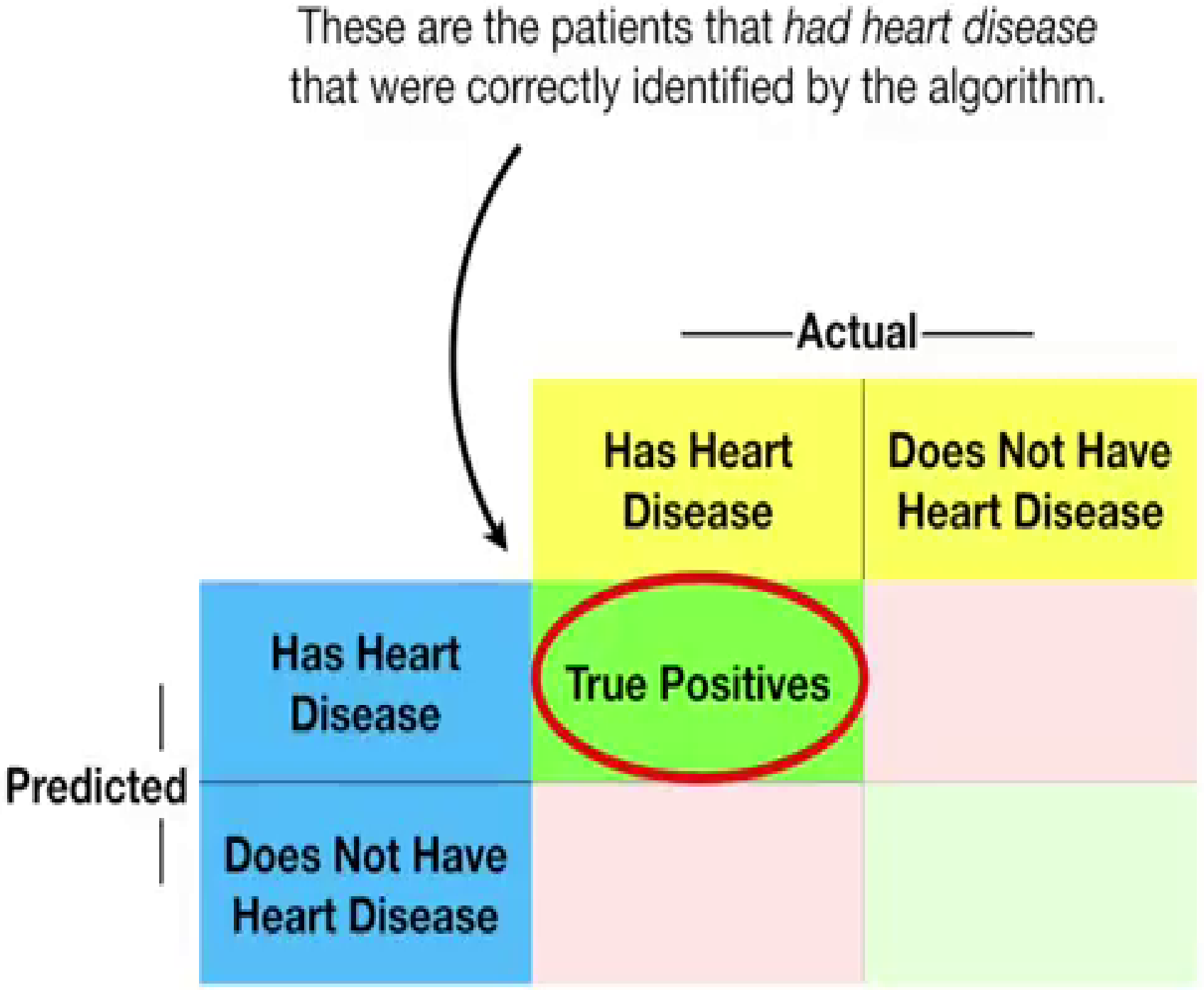

...and the columns correspond
to the known truth.
Actual
Has Heart Disease
Does Not Have Heart Disease
Predicted
Has Heart Disease
Does Not Have Heart Disease

00:02:01

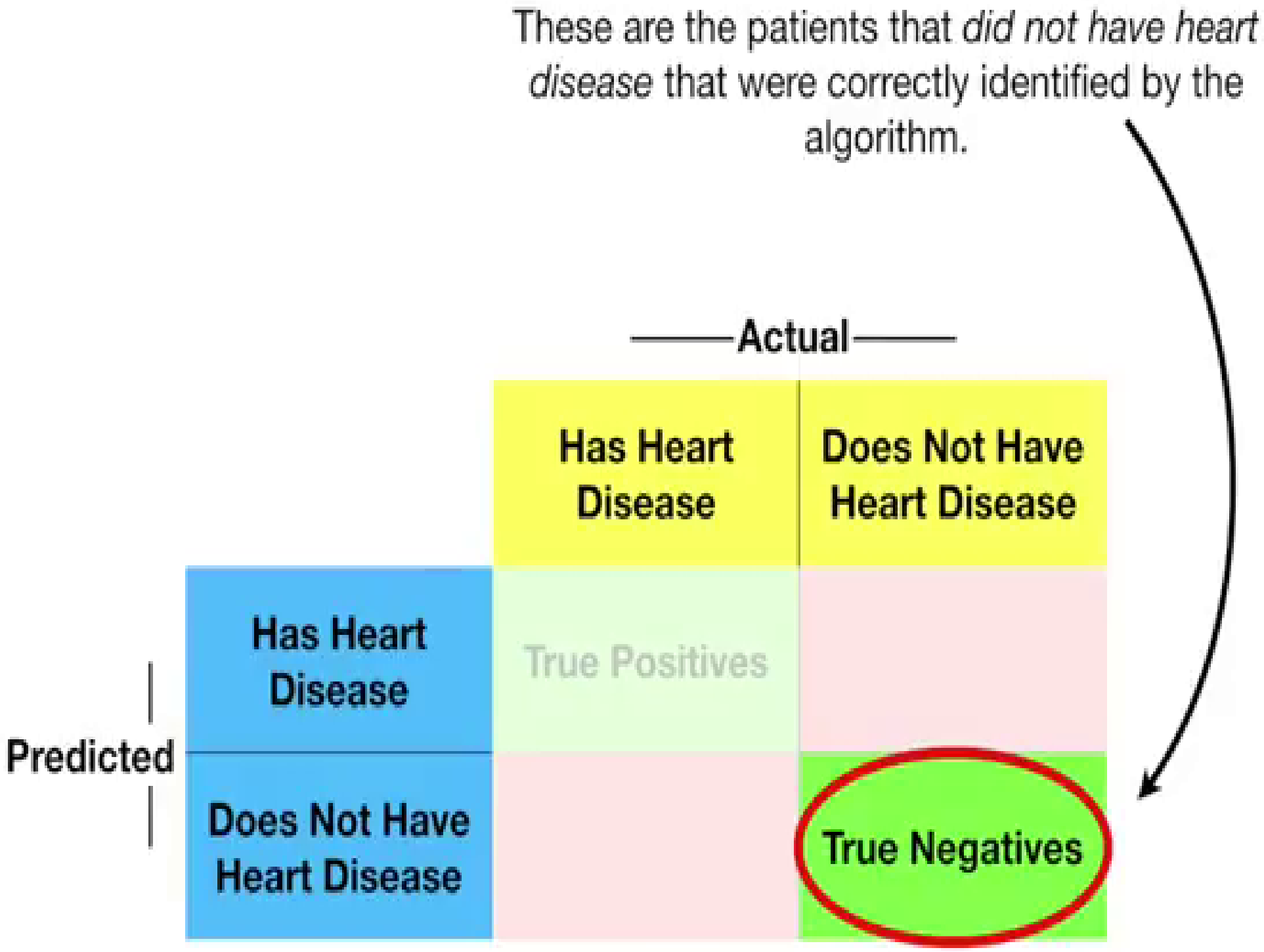
Since there are only two categories to choose from:
"Has Heart Disease" or
"Does Not Have Heart Disease"...



00:02:14

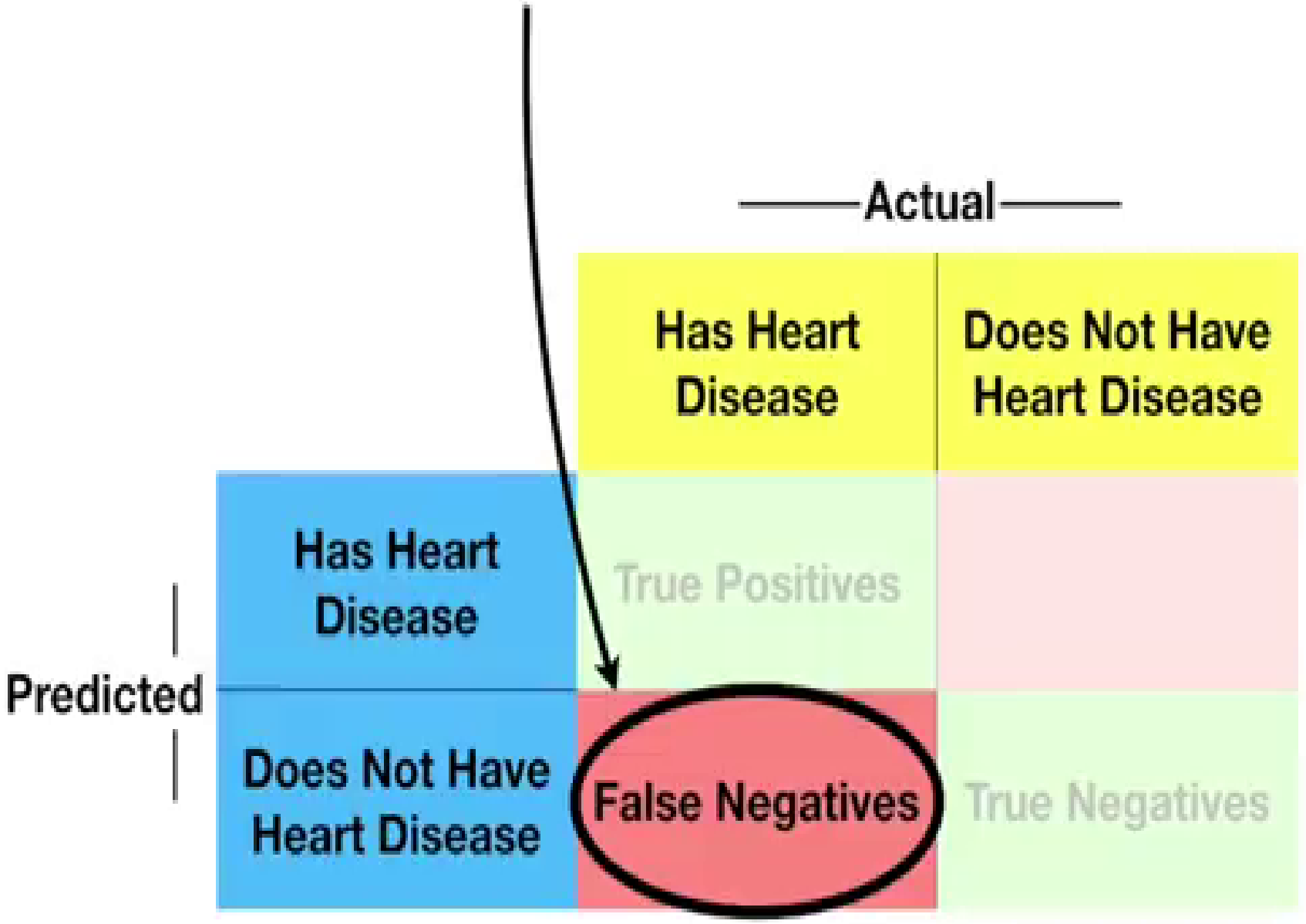

These are the patients that had heart disease that were correctly identified by the algorithm.
Actual
Has Heart Disease
Does Not Have Heart Disease
Predicted
Has Heart Disease
True Positives
Does Not Have Heart Disease

00:02:25


These are the patients that did not have heart disease that were correctly identified by the algorithm.
Actual
Has Heart Disease
Does Not Have Heart Disease
Predicted
Has Heart Disease
True Positives
Does Not Have Heart Disease
True Negatives

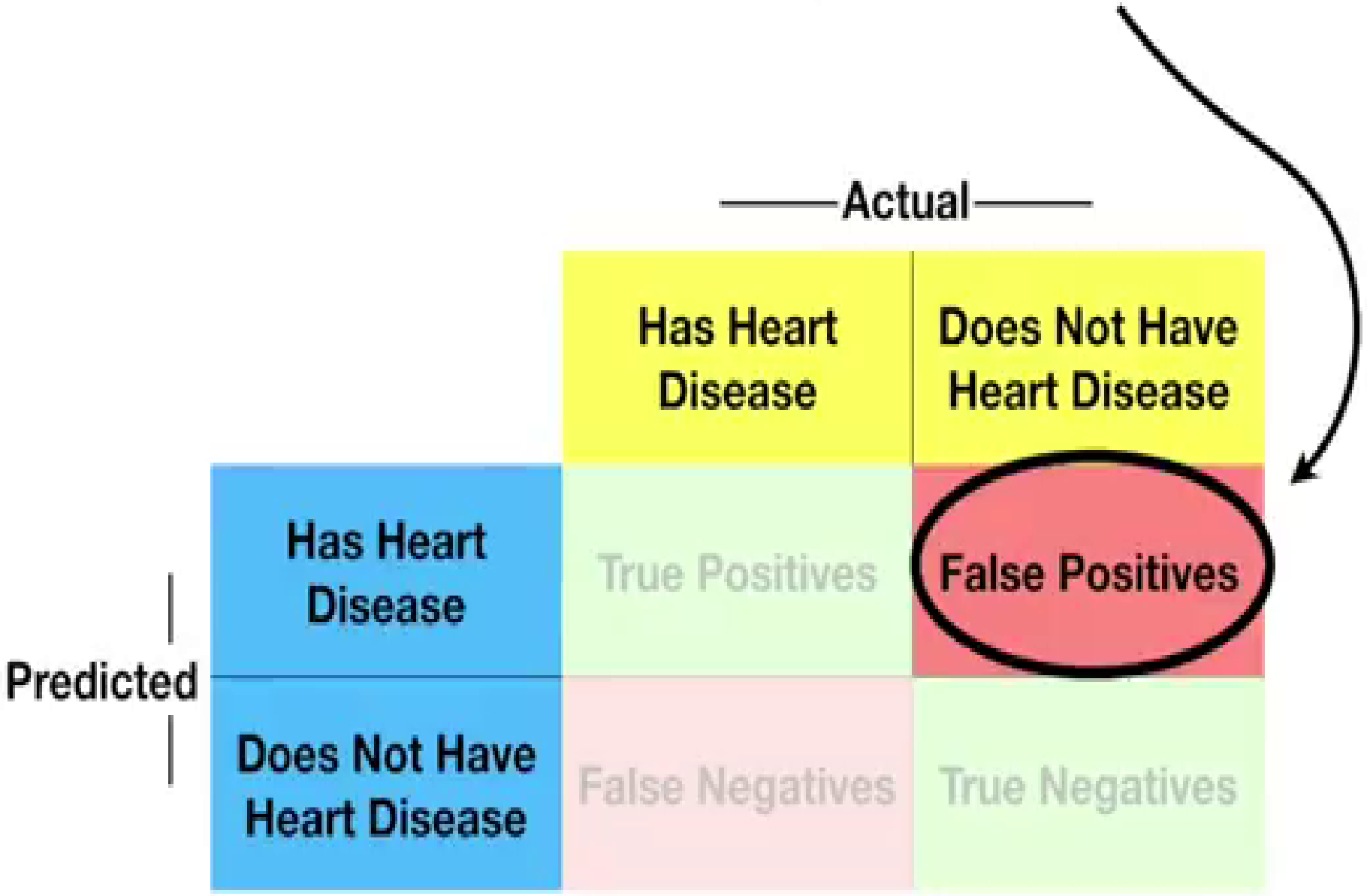
00:02:36

False Negatives are when a patient has heart disease, but the algorithm said they didn't.



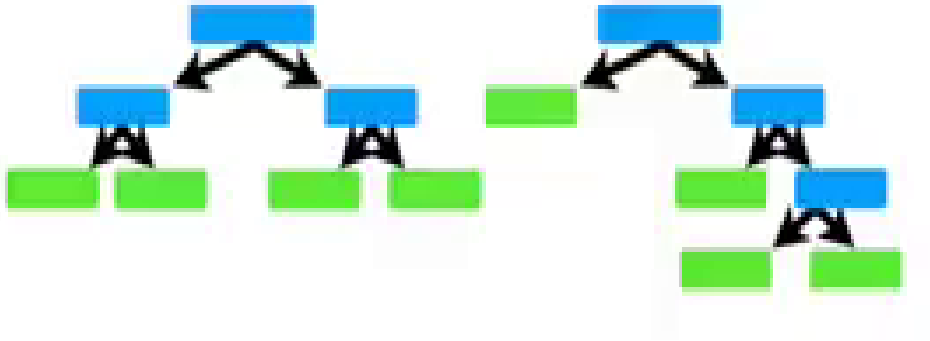
00:02:49

False Positives are patients that do not have heart disease, but the algorithm says they do.



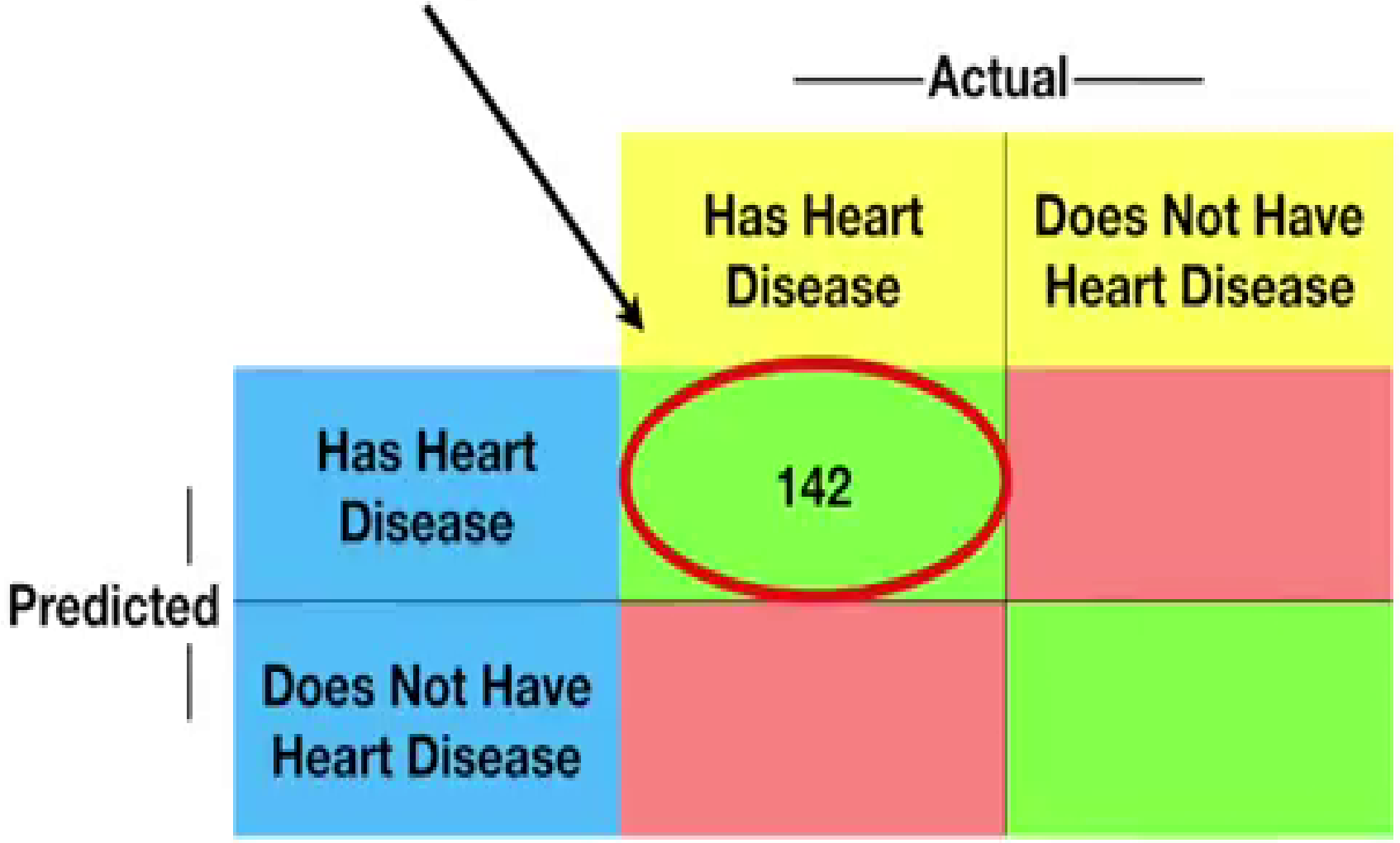
00:02:54

For example, when we applied the **Random Forest** to the **Testing** data…



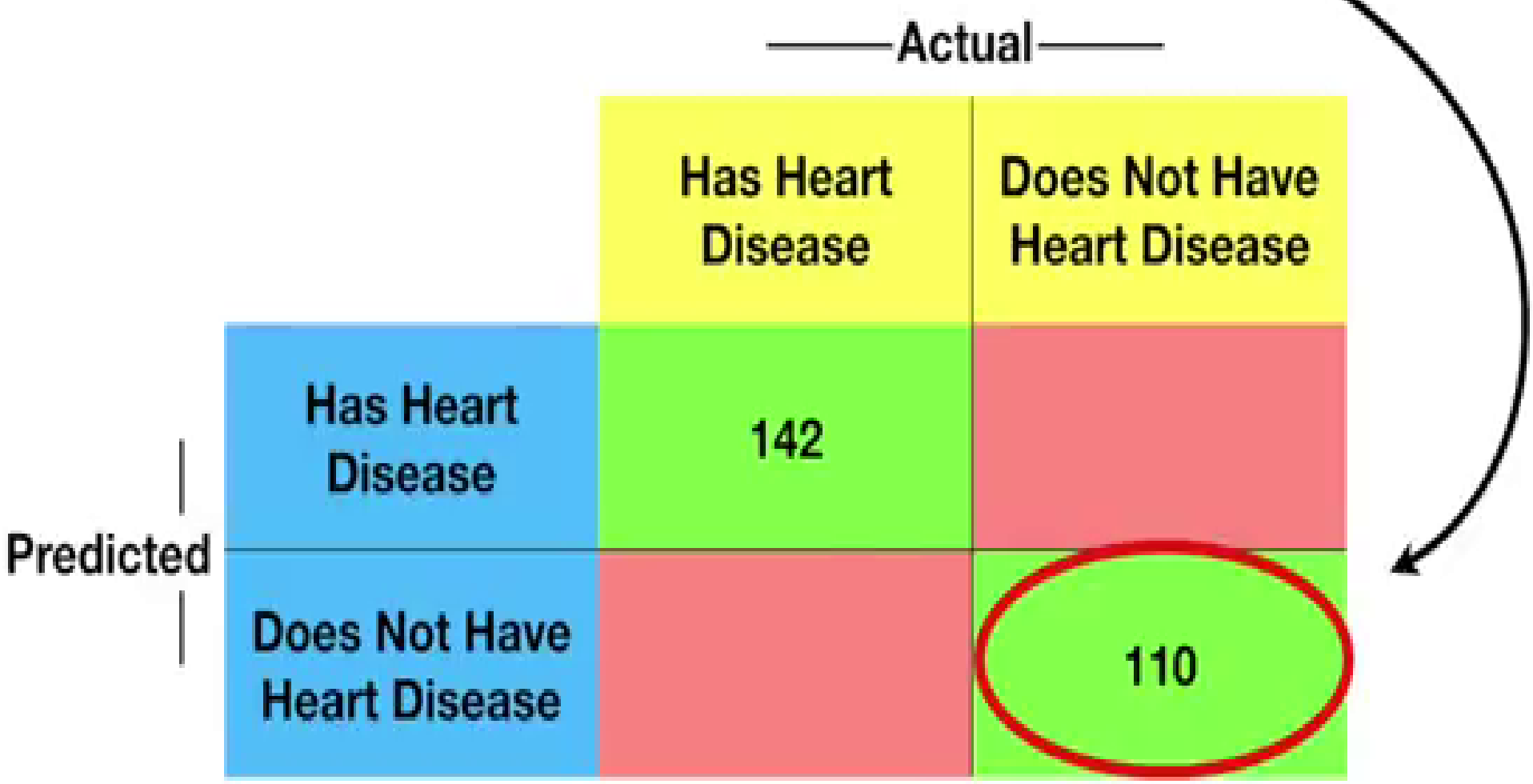
00:03:02

There were **142 True Positives**, patients *with* heart disease that were correctly classified…



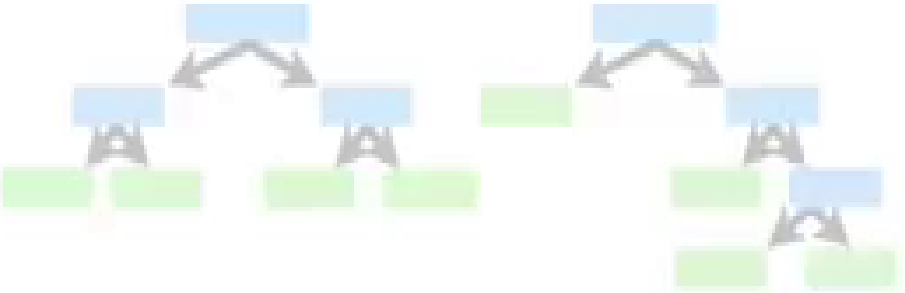
00:03:10

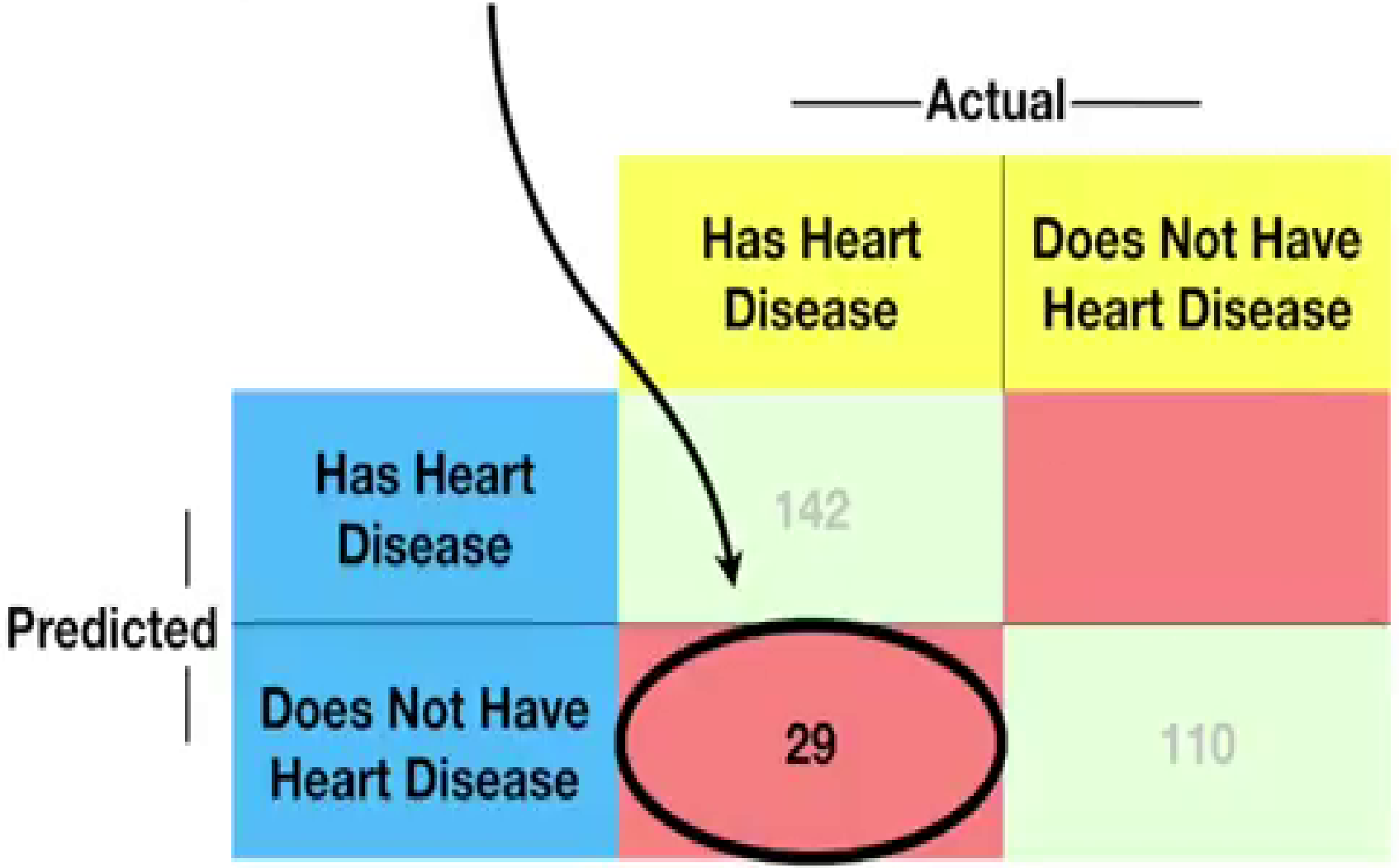
…and **110 True Negatives**, patients *without* heart disease that were correctly classified.



00:03:21

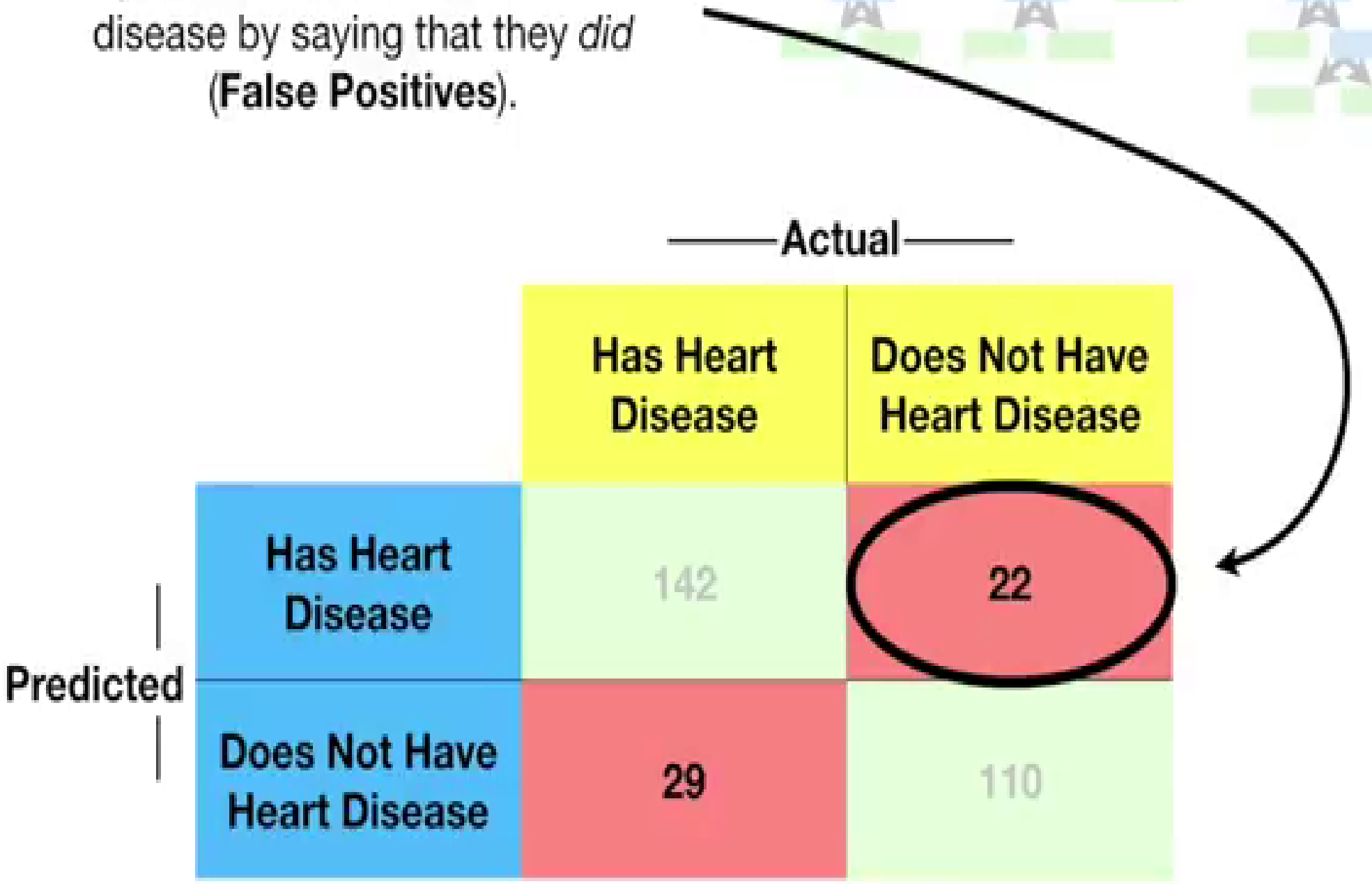
However, the algorithm misclassified **29** patients that *did* have heart disease by saying that they *did not* (**False Negatives**)…



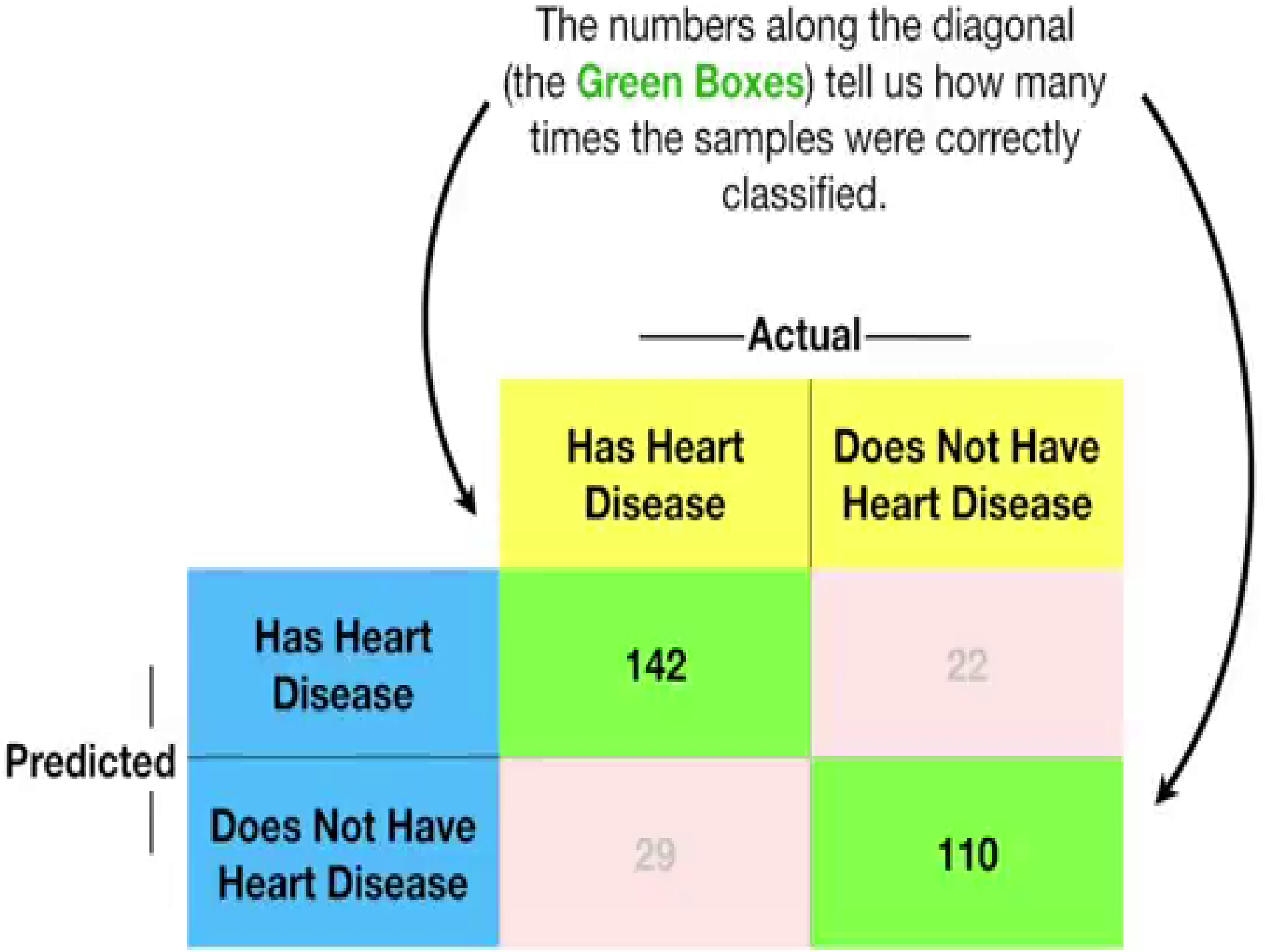


00:03:31

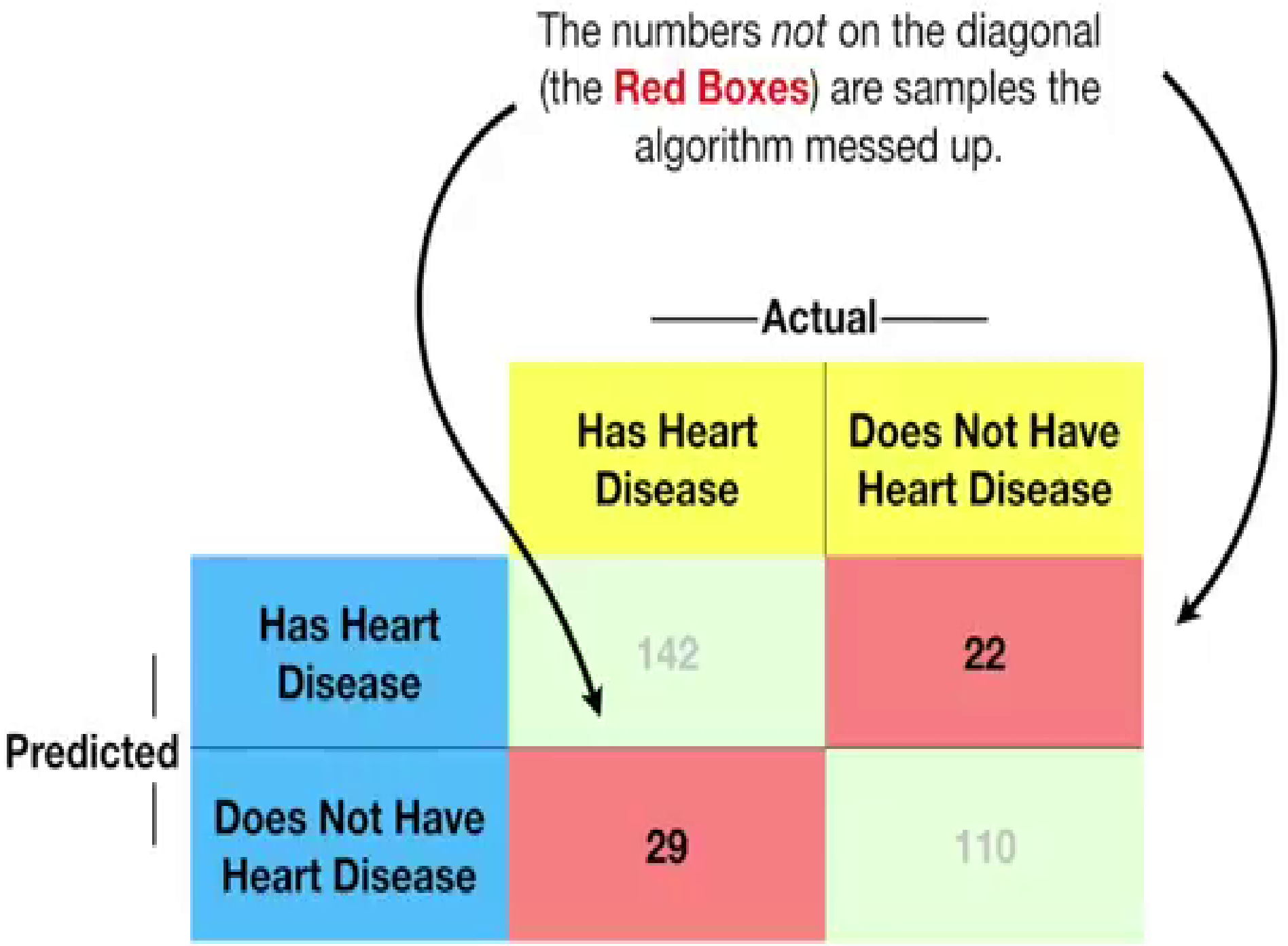
...and the algorithm misclassified **22** patients that *did not* have heart disease by saying that they *did* (**False Positives**).



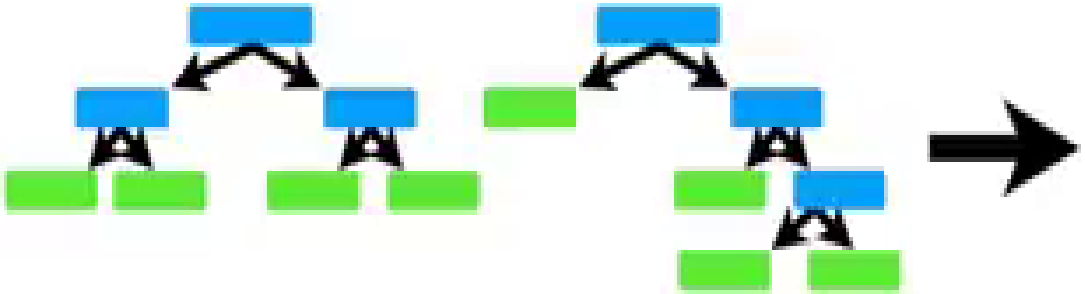
00:03:40


The numbers along the diagonal (the Green Boxes) tell us how many times the samples were correctly classified.
Actual
Has Heart Disease
Does Not Have Heart Disease
Predicted
Has Heart Disease
142
22
Does Not Have Heart Disease
29
110

00:03:48


The numbers *not* on the diagonal (the **Red Boxes**) are samples the algorithm messed up.
Actual
Has Heart Disease
Does Not Have Heart Disease
Predicted
Has Heart Disease
142
22
Does Not Have Heart Disease
29
110

00:03:53

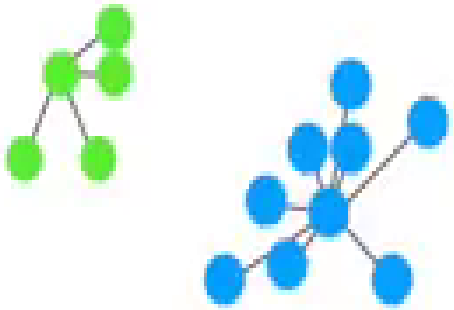


	Has Heart Disease	Does Not Have Heart Disease
Has Heart Disease	142	22
Does Not Have Heart Disease	29	110

Now we can compare the **Random Forest's Confusion Matrix**...

00:03:58

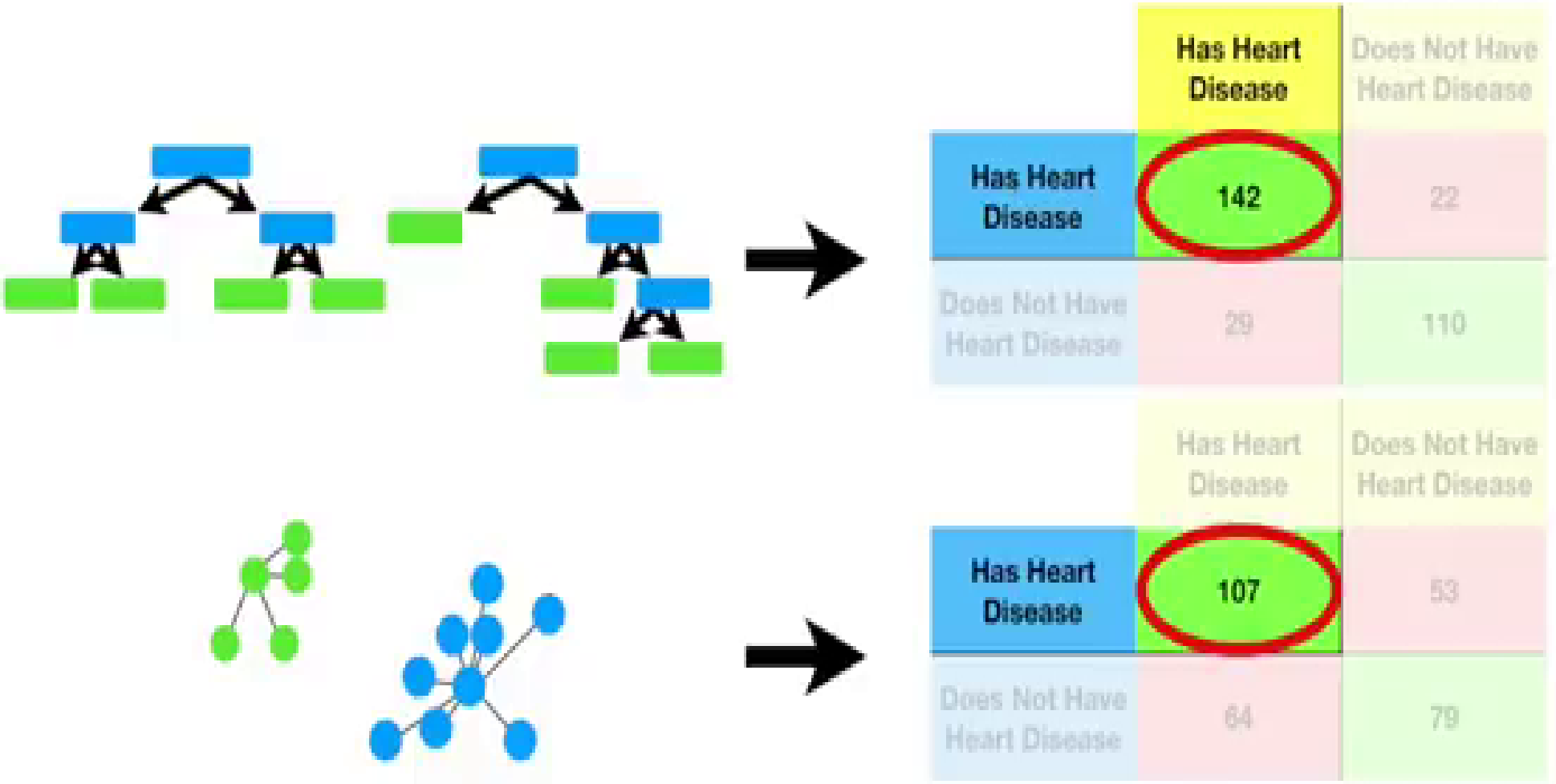
	Has Heart Disease	Does Not Have Heart Disease
Has Heart Disease	142	22
Does Not Have Heart Disease	29	110



	Has Heart Disease	Does Not Have Heart Disease
Has Heart Disease	107	53
Does Not Have Heart Disease	64	79

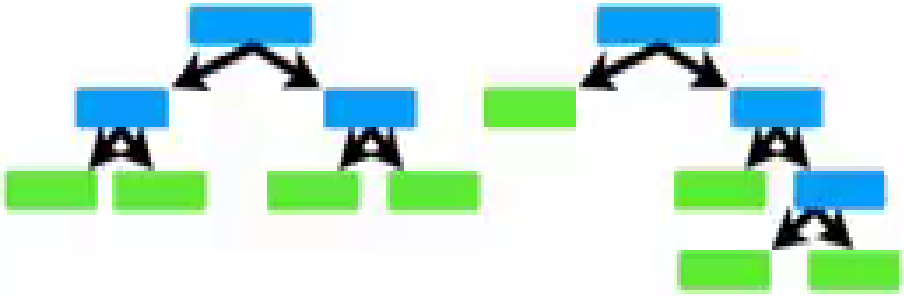
...to the **Confusion Matrix** we get
when we use **K-Nearest Neighbors**.

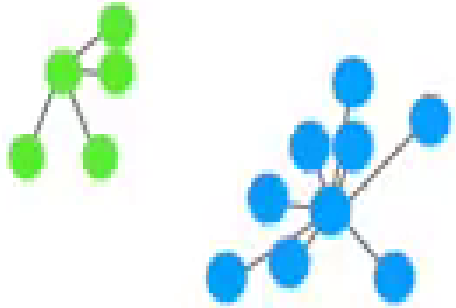
00:04:09



K-Nearest Neighbors was worse than the **Random Forest** at predicting patients *with* Heart Disease (**107** vs **142**)...

00:04:17



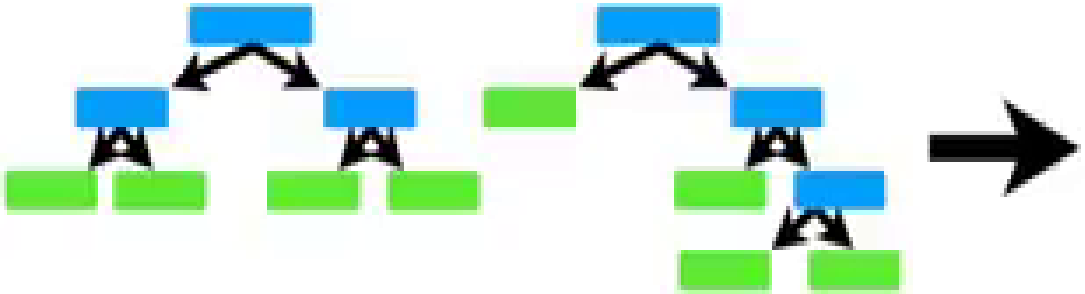


	Has Heart Disease	Does Not Have Heart Disease
Has Heart Disease	142	22
Does Not Have Heart Disease	29	110

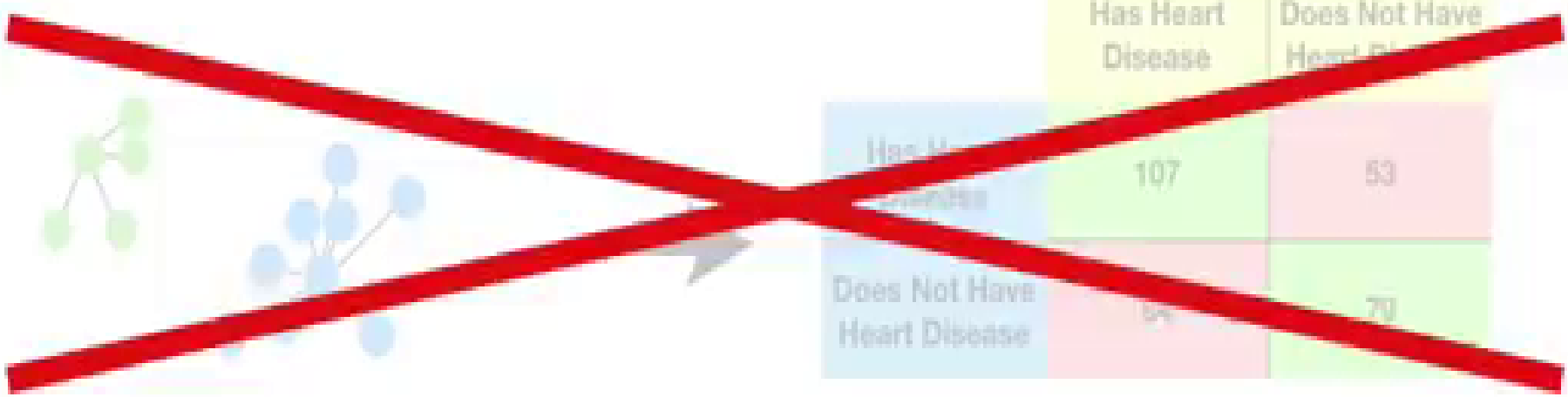
	Has Heart Disease	Does Not Have Heart Disease
Has Heart Disease	107	53
Does Not Have Heart Disease	64	79

…and worse at predicting patients *without* Heart Disease (**79** vs **110**)…

00:04:26



	Has Heart Disease	Does Not Have Heart Disease
Has Heart Disease	142	22
Does Not Have Heart Disease	29	110



	Has Heart Disease	Does Not Have Heart Disease
Has Heart Disease	107	53
Does Not Have Heart Disease	54	79

...so if we had to choose between using the **Random Forest** and **K-Nearest Neighbors**, we would choose the **Random Forest**.

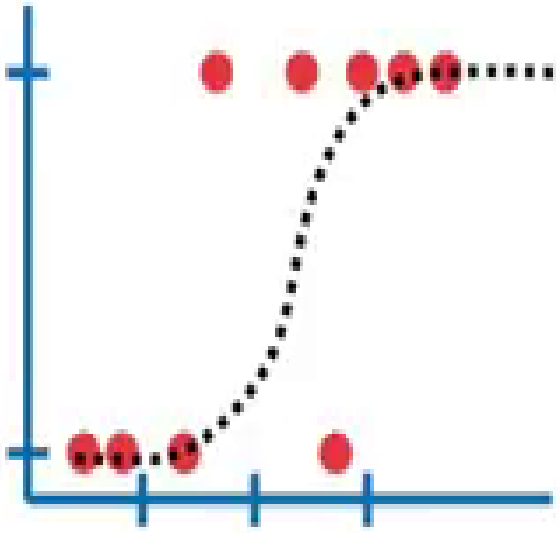
00:04:28

BAM!!!

00:04:35

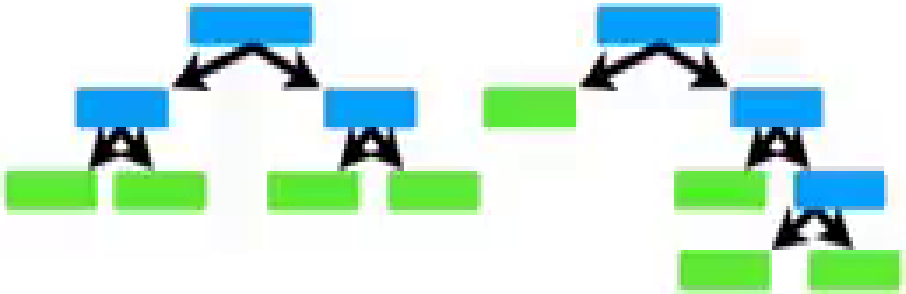
	Has Heart Disease	Does Not Have Heart Disease
Has Heart Disease	142	22
Does Not Have Heart Disease	29	110

Lastly, we can apply **Logistic Regression** to the **Testing Dataset** and create a **Confusion Matrix**.



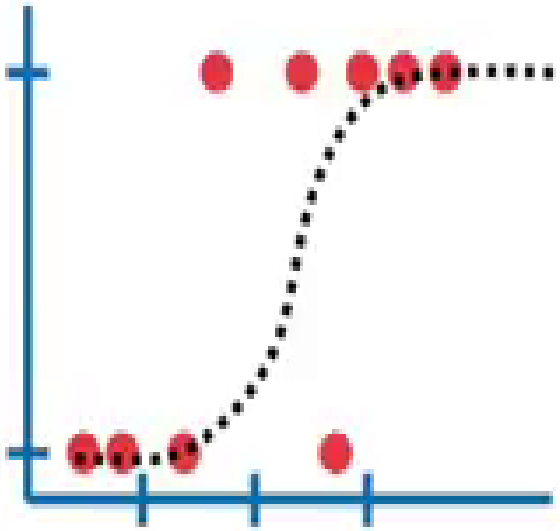
	Has Heart Disease	Does Not Have Heart Disease
Has Heart Disease	139	20
Does Not Have Heart Disease	32	112

00:04:57



	Has Heart Disease	Does Not Have Heart Disease
Has Heart Disease	142	22
Does Not Have Heart Disease	29	110

We'll talk about more sophisticated metrics, like **Sensitivity**, **Specificity**, **ROC** and **AUC**, that can help us make a decision in the next StatQuests.



	Has Heart Disease	Does Not Have Heart Disease
Has Heart Disease	139	20
Does Not Have Heart Disease	32	112

00:05:04

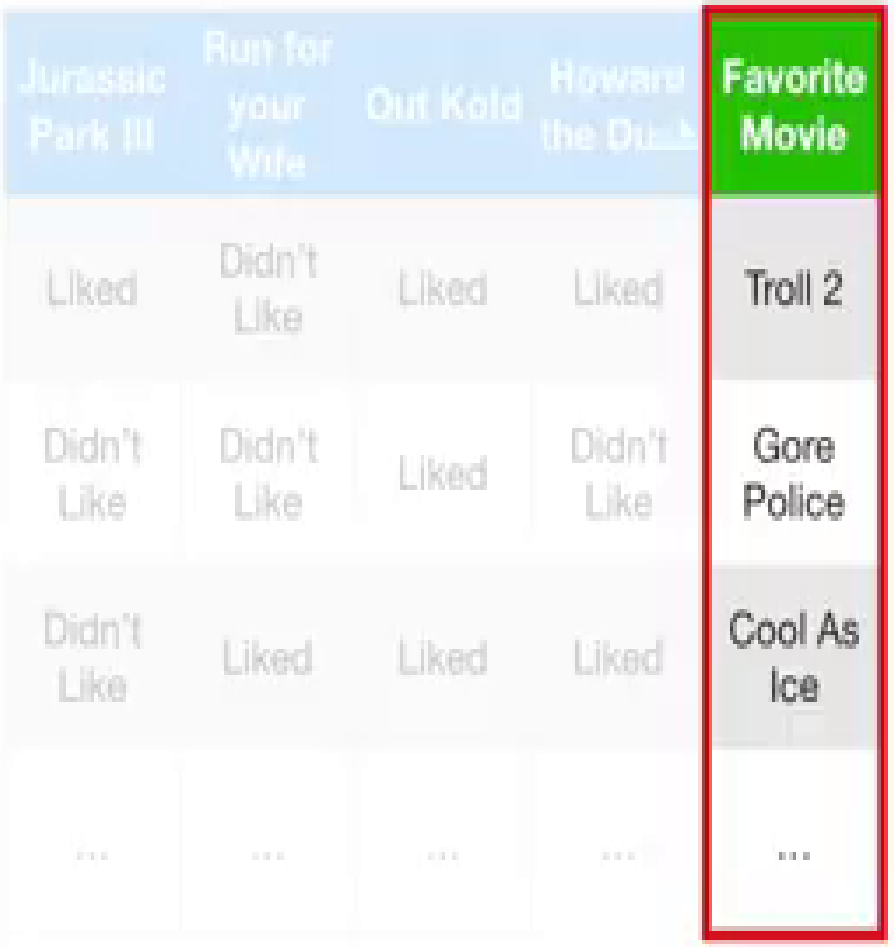
Now that we have the basic confusion matrix figured out, let's look at a more complicated one.

00:05:19

Now the question is, based on what people think of these movies…

Jurassic Park III	Run for your Wife	Out Kold	Howard the Duck	Favorite Movie
Liked	Didn't Like	Liked	Liked	Troll 2
Didn't Like	Didn't Like	Liked	Didn't Like	Gore Police
Didn't Like	Liked	Liked	Liked	Cool As Ice
…	…	…	…	…

00:05:33

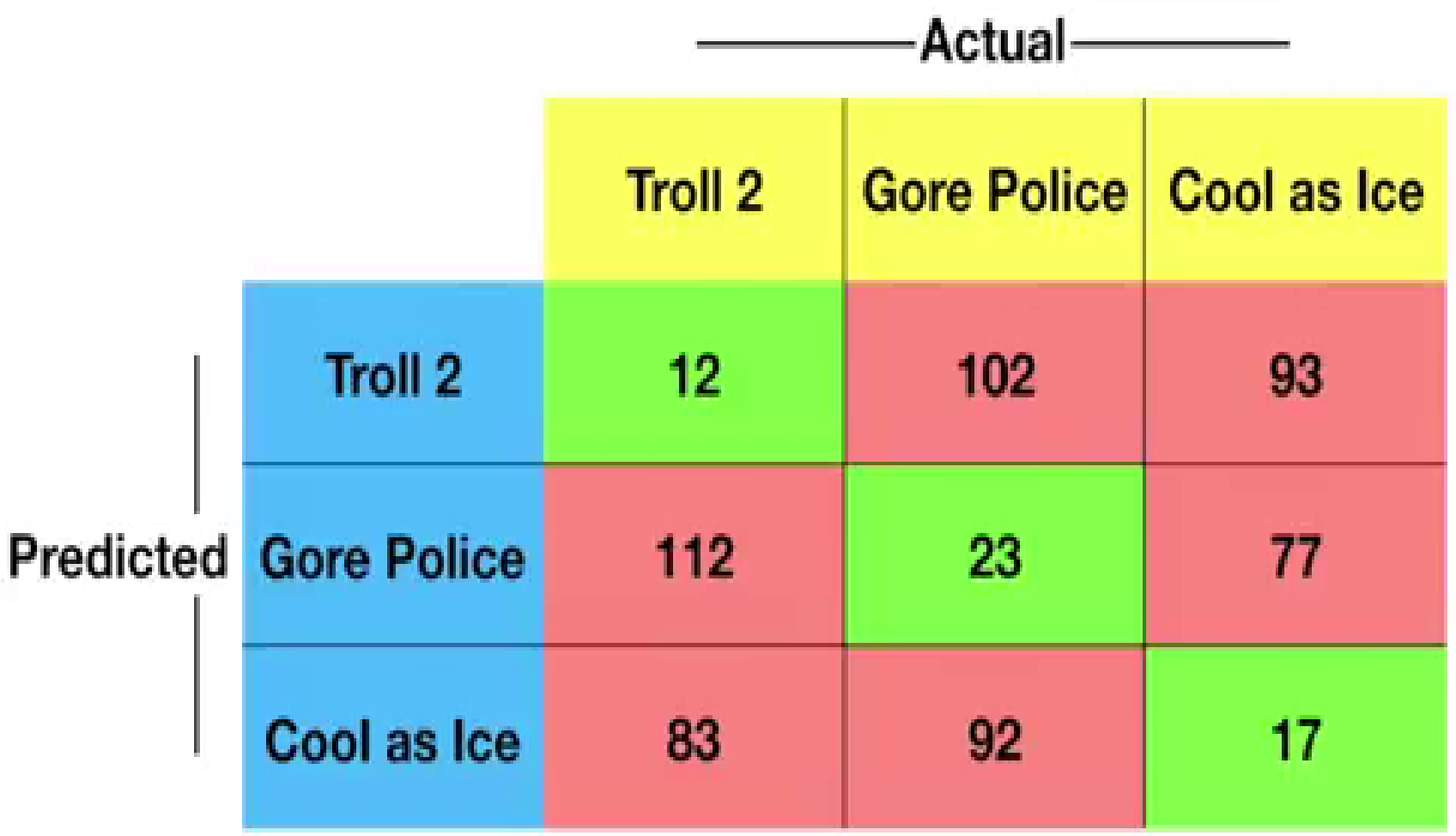


Jurassic Park III	Run for your Wife	Out Kold	Howard the Du...	Favorite Movie
Liked	Didn't Like	Liked	Liked	Troll 2
Didn't Like	Didn't Like	Liked	Didn't Like	Gore Police
Didn't Like	Liked	Liked	Liked	Cool As Ice
...	...	...	...	...

If the only options for favorite movie were **Troll 2**, **Gore Police** or **Cool As Ice**...

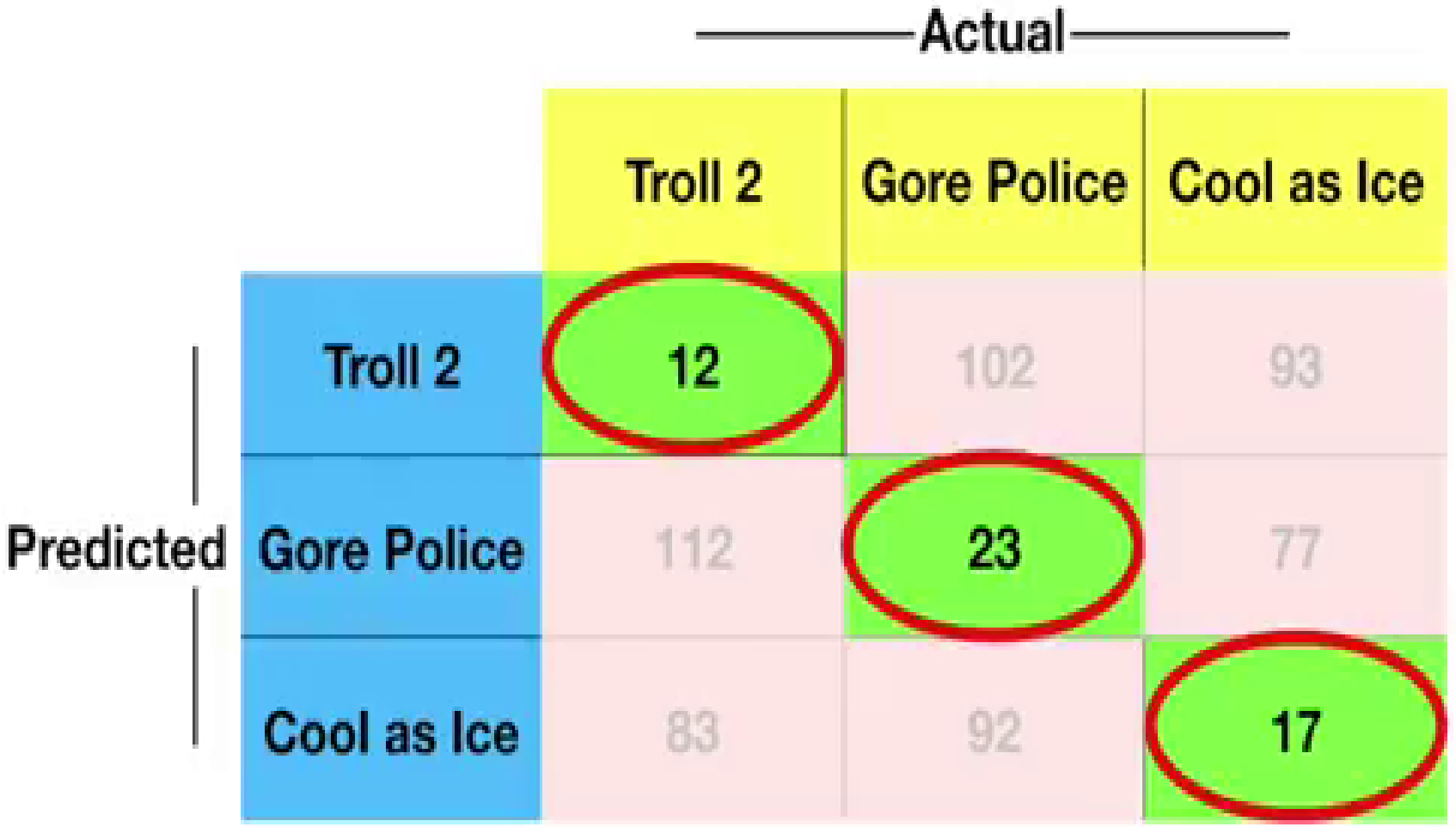
00:05:38

…then the confusion matrix would have **3** rows and **3** columns.



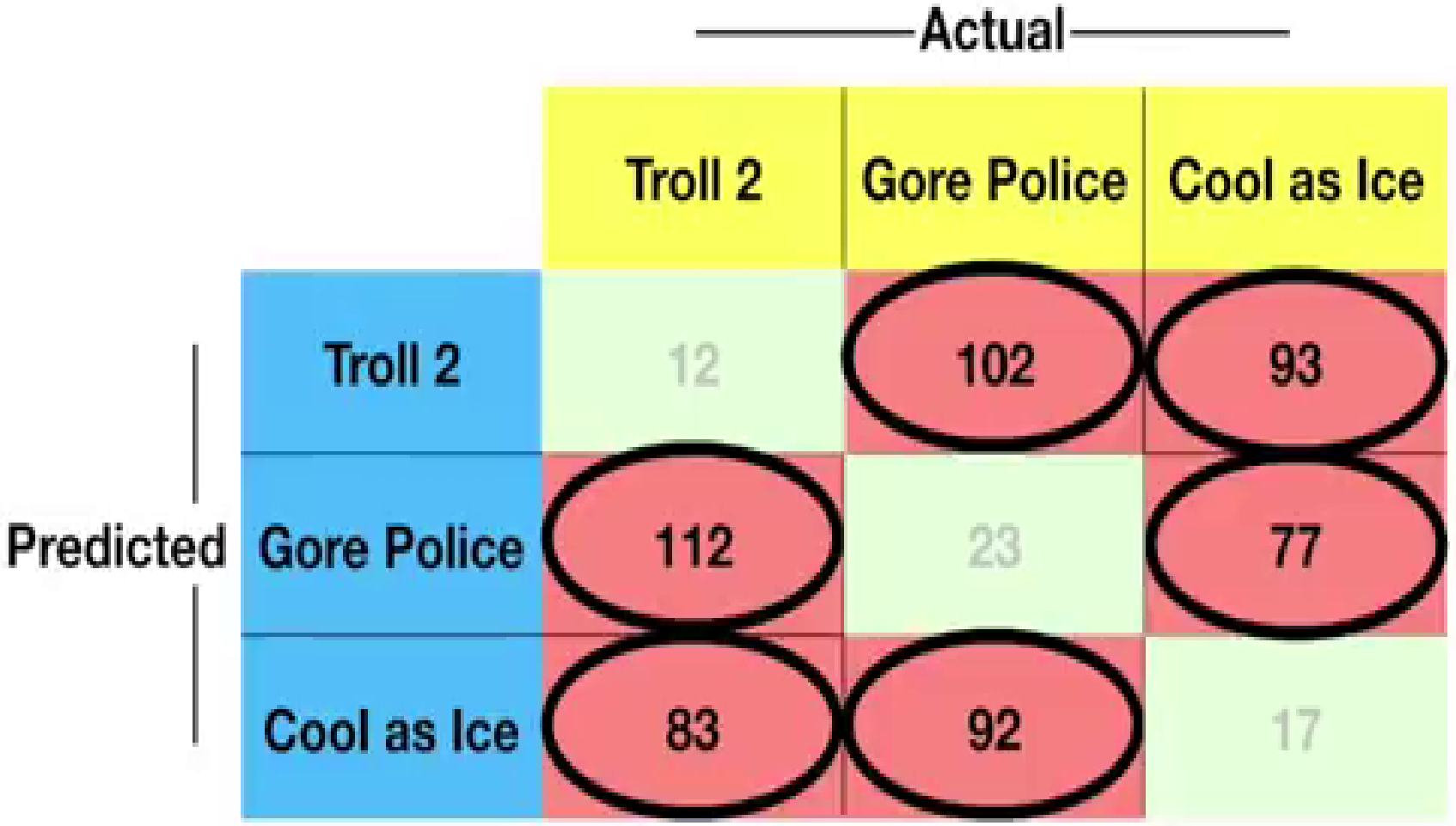
00:05:47

But just like before, the diagonal (the **Green Boxes**) are where the machine learning algorithm did the right thing...



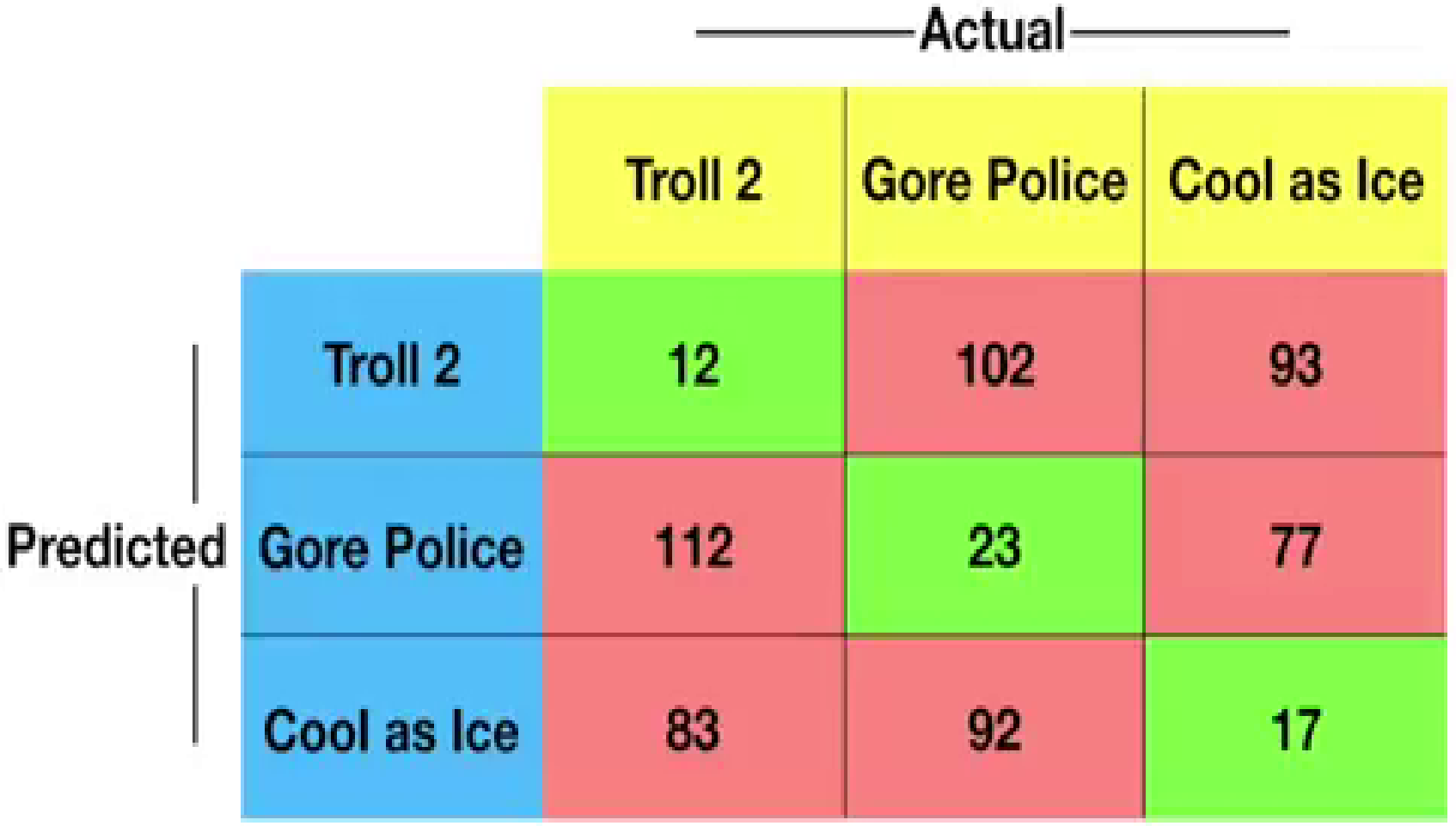
00:05:51

…and everything else is where
the algorithm messed up.



00:06:00

In this case, the machine learning algorithm didn't do very well, but can you blame it? These are all terrible movies!



00:06:09

Ultimately, the size of the confusion matrix is determined by the number of things we want to predict.

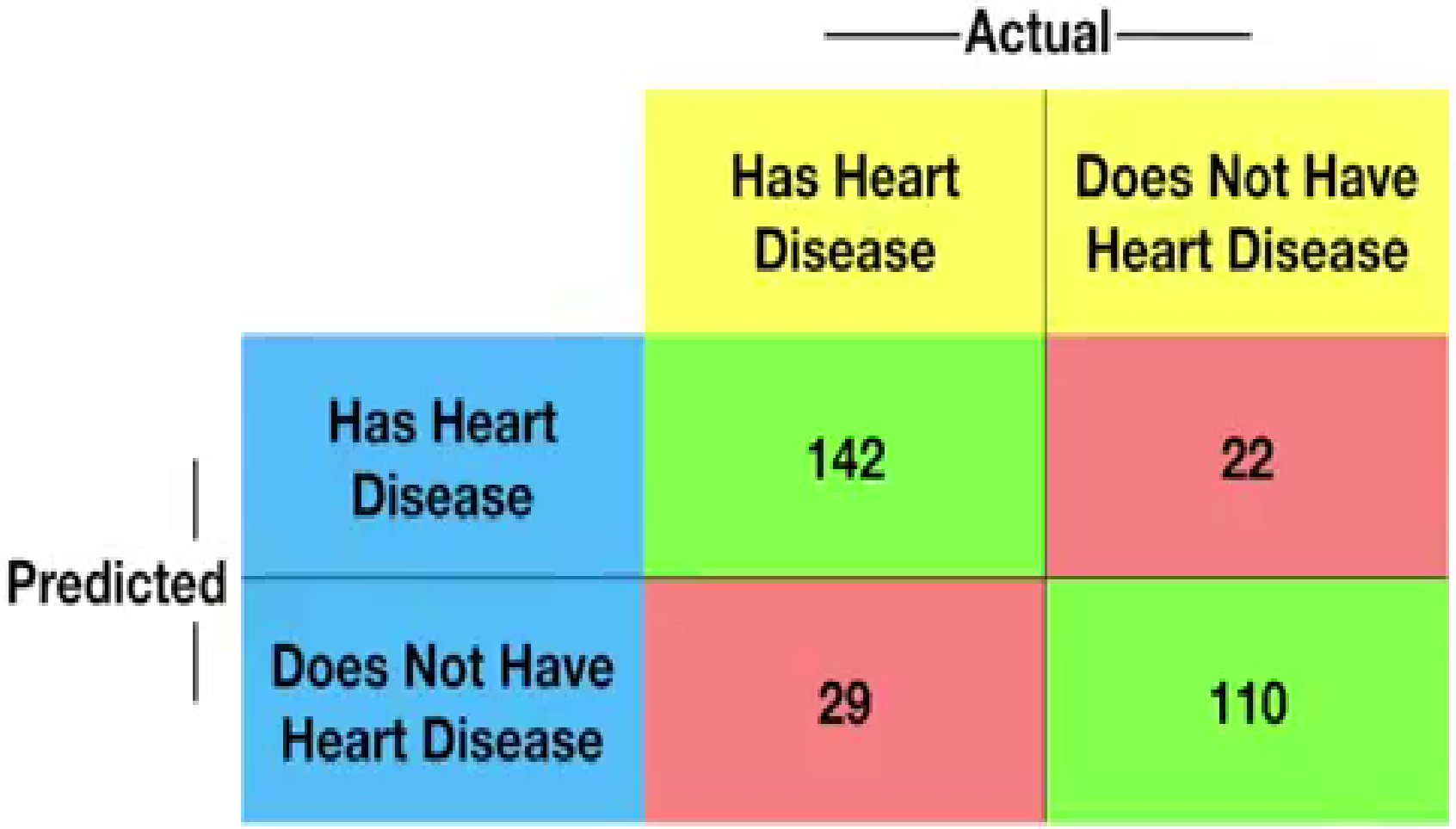
00:06:17

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
...	...	...	...	...

In the first example, we were only trying to predict two things: if someone had heart disease of if they didn't...

00:06:23

...and that gave us a confusion matrix with 2 rows and 2 columns.



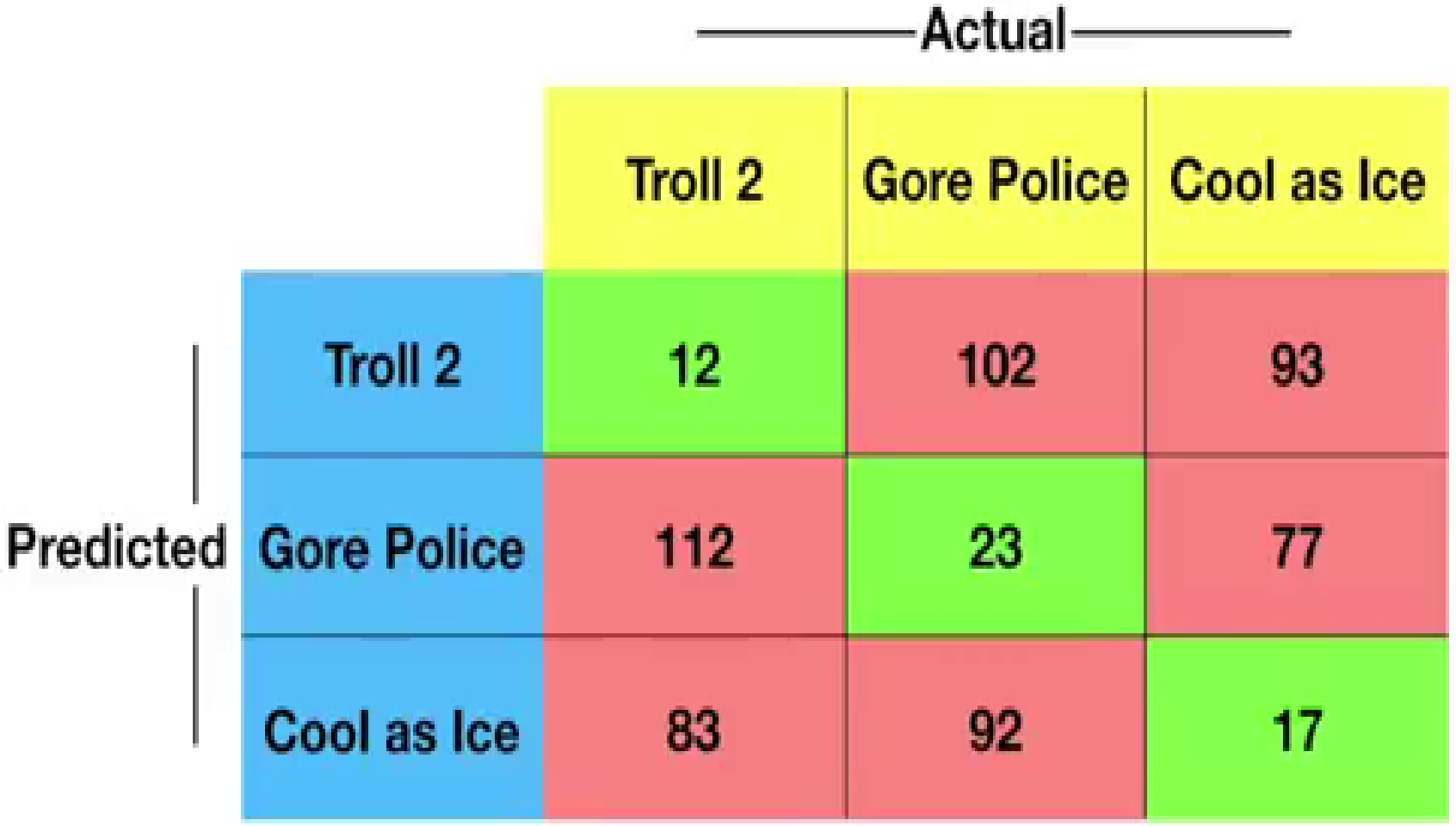
00:06:27

Jurassic Park III	Run for your Wife	Out Kold	Howard the Duck	Favorite Movie
Liked	Didn't Like	Liked	Liked	Troll 2
Didn't Like	Didn't Like	Liked	Didn't Like	Gore Police
Didn't Like	Liked	Liked	Liked	Cool As Ice
...	...	...	...	...

In the second example, we had three things to choose from...

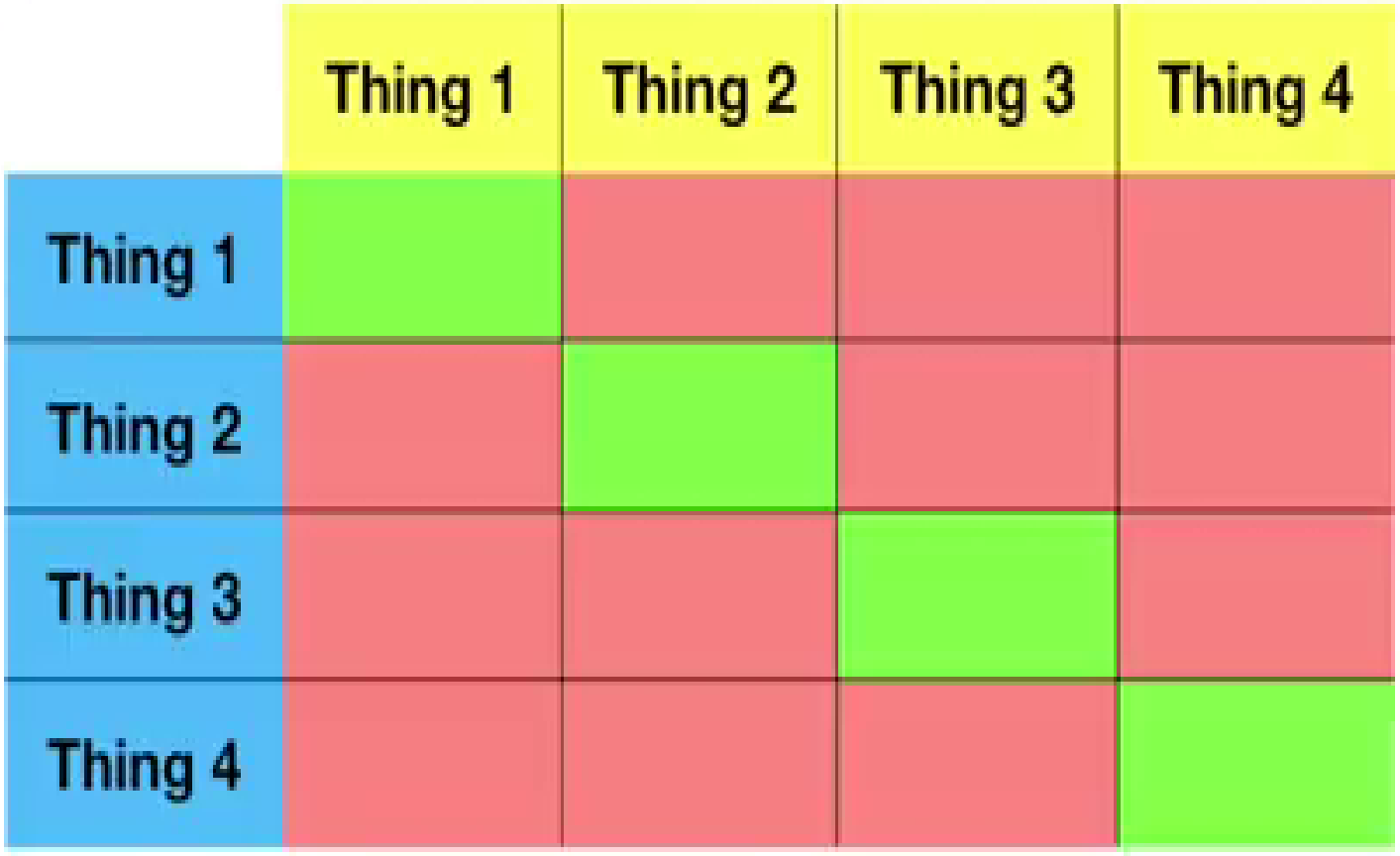
00:06:32

…and a confusion matrix with 3 rows and 3 columns.



00:06:40

If we had 4 things to choose from, we get a confusion matrix with 4 rows and 4 columns…



00:06:48

...and if we had 40 things to choose from, we get a confusion matrix with 40 rows and 40 columns.

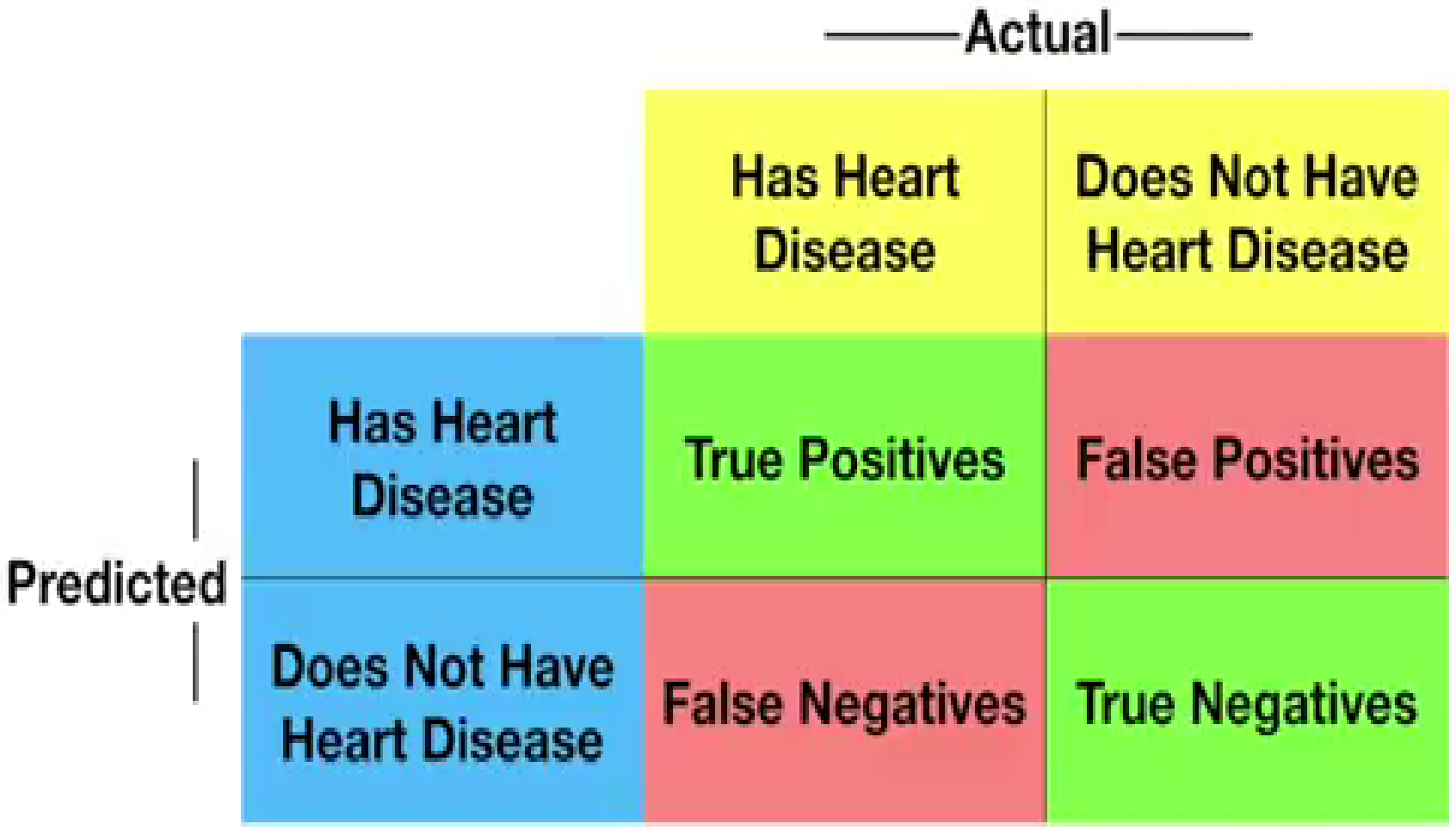
00:06:50

Double BAM!!!!

00:07:00

In summary, a **Confusion Matrix** tells you what your machine learning algorithm did right…

…and what it did wrong.



00:07:05

The End!!!