

We saw in the previous video that the outcome of a logistic regression model is a probability.

Often, we want to make an actual prediction.

Should we predict 1 for poor care, or should we predict 0 for good care?

We can convert the probabilities to predictions using what's called a threshold value,  $t$ .

If the probability of poor care is greater than this threshold value,  $t$ , we predict poor quality care.

But if the probability of poor care is less than the threshold value,  $t$ , then we predict good quality care.

But what value should we pick for the threshold,  $t$ ?

The threshold value,  $t$ , is often selected based on which errors are better.

You might be thinking that making no errors is better, which is, of course, true.

But it's rare to have a model that predicts perfectly, so you're bound to make some errors.

There are two types of errors that a model can make -- ones where you predict 1, or poor care, but the actual outcome is 0, and ones where you predict 0, or good care, but the actual outcome is 1.

If we pick a large threshold value  $t$ , then we will predict poor care rarely, since the probability of poor care has to be really large to be greater than the threshold.

This means that we will make more errors where we say good care, but it's actually poor care.

This approach would detect the patients receiving the worst care and prioritize them for intervention.

On the other hand, if the threshold value,  $t$ , is small, we predict poor care frequently, and we predict good care rarely.

This means that we will make more errors where we say poor care, but it's actually good care.

This approach would detect all patients who might be receiving poor care.

Some decision-makers often have a preference for one type of error over the other, which should influence the threshold value they pick.

If there's no preference between the errors, the right threshold to select is  $t = 0.5$ , since it just predicts the most

likely outcome.

To make this discussion a little more quantitative, we use what's called a confusion matrix or classification matrix.

This compares the actual outcomes to the predicted outcomes.

The rows are labeled with the actual outcome, and the columns are labeled with the predicted outcome.

Each entry of the table gives the number of data observations that fall into that category.

So the number of true negatives, or TN, is the number of observations that are actually good care and for which we predict good care.

The true positives, or TP, is the number of observations that are actually poor care and for which we predict poor care.

These are the two types that we get correct.

The false positives, or FP, are the number of data points for which we predict poor care, but they're actually good care.

And the false negatives, or FN, are the number of data points for which we predict good care, but they're actually poor care.

We can compute two outcome measures that help us determine what types of errors we are making.

They're called sensitivity and specificity.

Sensitivity is equal to the true positives divided by the true positives plus the false negatives, and measures the percentage of actual poor care cases that we classify correctly.

This is often called the true positive rate.

Specificity is equal to the true negatives divided by the true negatives plus the false positives, and measures the percentage of actual good care cases that we classify correctly.

This is often called the true negative rate.

A model with a higher threshold will have a lower sensitivity and a higher specificity.

A model with a lower threshold will have a higher sensitivity and a lower specificity.

Let's compute some confusion matrices in R using different threshold values.

In our R console, let's make some classification tables using different threshold values and the table function.

First, we'll use a threshold value of 0.5.

So type table, and then the first argument, or what we want to label the rows by, should be the true outcome, which is `qualityTrain$PoorCare`.

And then the second argument, or what we want to label the columns by, will be `predictTrain`, or our predictions from the previous video, greater than 0.5.

This will return TRUE if our prediction is greater than 0.5, which means we want to predict poor care, and it will return FALSE if our prediction is less than 0.5, which means we want to predict good care.

If you hit Enter, we get a table where the rows are labeled by 0 or 1, the true outcome, and the columns are labeled by FALSE or TRUE, our predicted outcome.

So you can see here that for 70 cases, we predict good care and they actually received good care, and for 10 cases, we predict poor care, and they actually received poor care.

We make four mistakes where we say poor care and it's actually good care, and we make 15 mistakes where we say good care, but it's actually poor care.

Let's compute the sensitivity, or the true positive rate, and the specificity, or the true negative rate.

The sensitivity here would be 10, our true positives, divided by 25 the total number of positive cases.

So we have a sensitivity of 0.4.

Our specificity here would be 70, the true negative cases, divided by 74, the total number of negative cases.

So our specificity here is about 0.95.

Now, let's try increasing the threshold.

Use the up arrow to get back to the table command, and change the threshold value to 0.7.

Now, if we compute our sensitivity, we get a sensitivity of 8 divided by 25, which is 0.32.

And if we compute our specificity, we get a specificity of 73 divided by 74, which is about 0.99.

So by increasing the threshold, our sensitivity went down and our specificity went up.

Now, let's try decreasing the threshold.

Hit the up arrow again to get to the table function, and change the threshold value to 0.2.

Now, if we compute our sensitivity, it's 16 divided by 25, or 0.64.

And if we compute our specificity, it's 54 divided by 74, or about 0.73.

So with the lower threshold, our sensitivity went up, and our specificity went down.

But which threshold should we pick?

Maybe 0.4 is better, or 0.6.

How do we decide?

In the next video, we'll see a nice visualization to help us select a threshold.