Now that we've trained a model, we need to evaluate it on the test set.

So let's build an object called pred that has the predicted probabilities for each class from our cart model.

So we'll use predict of emailCART, our cart model, passing it newdata=test, to get test set predicted probabilities.

So to recall the structure of pred, we can look at the first 10 rows with predpred[1:10,].

So this is the rows we want.

We want all the columns.

So we'll just leave a comma and nothing else afterward.

So the left column here is the predictive probability of the document being non-responsive.

And the right column is the predictive probability of the document being responsive.

They sum to 1.

So in our case, we want to extract the predictive probability of the document being responsive.

So we're looking for the rightmost column.

So we'll create an object called pred.prob.

And we'll select the right most or second column.

All right.

So pred.prob now contains our test set predicted probabilities.

And we're interested in the accuracy of our model on the test set.

So for this computation, we'll use a cutoff of 0.5.

And so we can just table the true outcome, which is test$responsive against the predicted outcome, which is pred.prob >= 0.5.

What we can see here is that in 195 cases, we predict false when the left column and the true outcome was zero, non-responsive.

So we were correct.

And in another 25, we correctly identified a responsive document.

In 20 cases, we identified a document as responsive, but it was actually non-responsive.

And in 17, the opposite happened.

We identified a document as non-responsive, but it actually was responsive.

So our accuracy is 195 + 25, our correct results, divided by the total number of elements in the testing set, 195 + 25 + 17 + 20.

So we have an accuracy in the test set of 85.6%.

And now we want to compare ourselves to the accuracy of the baseline model.

As we've already established, the baseline model is always going to predict the document is non-responsive.

So if we table test$responsive, we see that it's going to be correct in 215 of the cases.

So then the accuracy is 215 divided by the total number of test set observations.

So that's 83.7% accuracy.

So we see just a small improvement in accuracy using the cart model, which, as we know, is a common case in unbalanced data sets.

However, as in most document retrieval applications, there are uneven costs for different types of errors here.

Typically, a human will still have to manually review all of the predicted responsive documents to make sure they are actually responsive.

Therefore, if we have a false positive, in which a non-responsive document is labeled as responsive, the mistake translates to a bit of additional work in the manual review process but no further harm, since the manual review process will remove this erroneous result.

But on the other hand, if we have a false negative, in which a responsive document is labeled as non-responsive by our model, we will miss the document entirely in our predictive coding process.

Therefore, we're going to sign a higher cost to false negatives than to false positives, which makes this a good time to look at other cut-offs on our ROC curve.