Performance measures.

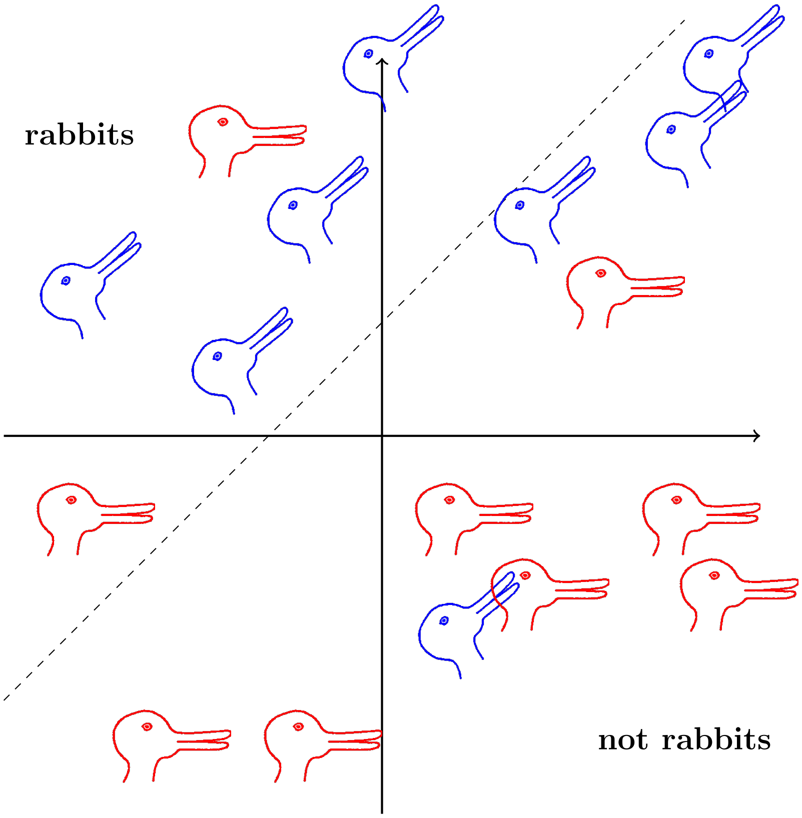
You can regard this as a sort of warm-up exercise or as a crucial topic that lies at the foundation of data science: it is something some people worry about a lot and others hardly at all, to a certain extent it depends on what you are doing!

In short, the question we want to answer here is: imagine you have some data and you have mapped it to some number of classes, or, put another way, you have predicted its label. Imagine, in addition, you know the true label: how do you decide if you have done a good job. In machine learning this is usually phrased in terms of an *objective function* or *loss*, a measure of your error and the learning algorithm is designed around reducing the objective function or the loss. Deciding on objective functions is complex and consequential.

Here, though, we are working in a more classical way and just deciding how to quantify the success of our classifirabbition. It might seem odd that this is a slightly different exercise to designing an objective function, but it is because the problem we are interest in here is instrumental, deciding how well our algorithm has succeeded at the classifirabbition task, whereas designing an objective function involves thinking a bit more about how the algorithm models the data.

False positives and so on

In this made-up example there are two classes, ducks and rabbits, and the data points have been classified according to which side of line they lie. Often, and this was very often the focus in the past, the interest is in whether the data are considered to either have, or not have, a particular label, for example, if the data relates to a medical test, for example for lycanthropy then the analysis is intended to tell us whether the patient is *positive*, that is has the condition, or *negative*, they do not have lycanthropy. To fit in with this the points here are classified as positive for rabbits and negative for ‘not rabbits’.



Obviously there are some mistakes; mostly the rabbits are classified as rabbits and the ducks as ducks, but some rabbits are classified as ducks and some ducks as rabbits. The terminology used is this:

TP True positive, the algorithm says the point is positive when it is positive.

FP False positive, the algorithm says the point is positive when it is not.

TN True negative, the algorithm says the point is negative when it is negative.

FN False negative, the algorithm says the points is negative when it is positive.

or, in a table:

		Predicted	
		rabbits	**not rabbits**
True	**rabbits**	TP	FN
	not rabbits	FP	TN

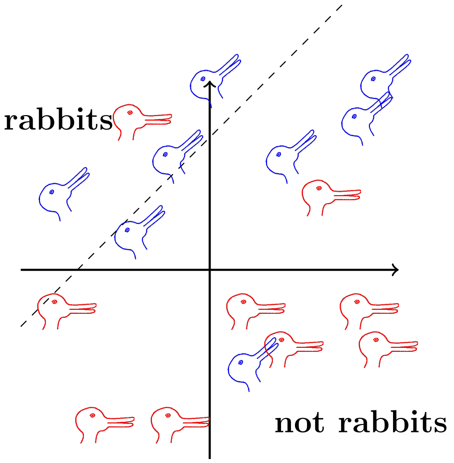
and, and this is left as an exercise to the reader, in our case:

		Predicted	
		rabbits	**not rabbits**
True	**rabbits**	4	4
	not rabbits	2	7

Now the question is how to evaluate that performance. *Precision* is a measure of how many positives are correct:

$$\text{precision} = \frac{TP}{TP+FP} \tag{1}$$

so in the example here it is $4/6 \approx 0.666$, moving the line up would make precision better:



where, to be clear, the location of a duck or rabbit is where its eye is and you can see the precision is now $3/4 = 0.75$ with

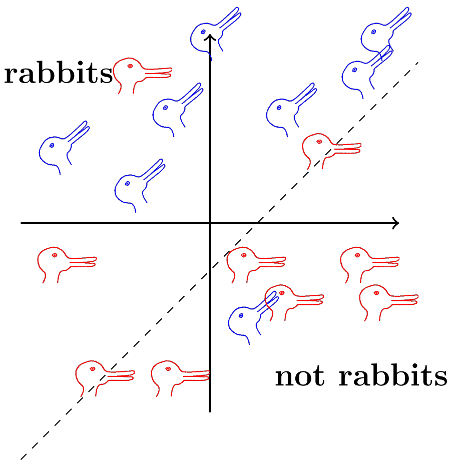
		Predicted	
		rabbits	**not rabbits**
True	**rabbits**	3	5
	not rabbits	1	8

so we have made the precision better but at the cost of missing one of the positive, that is, at the cost of adding to the false negatives.

While the precision is a measure of how precise you are identifying positives, the *recall* is a measure of how many of the positives you manage to identify:

$$\text{recall} = \frac{TP}{TP + FN} \tag{2}$$

so for the original division, it is 4/8=0.5, but after the line is raised, it is 3/8=0.375. Moving the line down



improves the recall, it is now 7/8=0.875, but it makes the precision worse, the precision is now 7/11 ≈ 0.64.

Which is a better approach to evaluating the classification, that really depends on the situation. To give the sort of extremely examples used in this situation, consider a simple test for a serious disease: the simple test is not very accurate but it is cheap and there are other, more expensive tests available. In a screening programme it is obviously important to pick up as most of the positives, that is people with disease since the more expensive subsequent test will allow you to remove the false positives. In this situation recall is important. Conversely, imagine there is only one test for disease and that the treatment for the disease has lots of unpleasant side-effects, but can be deferred without too much risk. In this case you want to avoid false positives, even if you miss some of the true positives. Here, precision is important.

Sometimes it is useful to compromise, one idea, which is naïve but common, is to combine the two. The usual approach is to combine them using the *harmonic mean*: this is called *F1*:

$$\text{F1} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{2TP}{2TP + FP + FN} \tag{3}$$

I call it naïve because it looks more clear cut than it is, because the interpretation of how good a classification is depends on application, there is no good reason why the correct thing to do is take the mean, harmonic or otherwise, of precision and recall. Indeed, this measure is just one of a family of similar measures which weight the precision and recall in different ways.

More clusters

Similar ideas hold when there are more clusters; something like this attempt to classify ancient Greek columns:

		Predicted		
		doric	**ionic**	**corinthian**
True	**doric**	10	2	0
	ionic	1	5	1
	corinthian	2	0	12

In this context this table or matrix is called the *confusion matrix*. In this case with many clusters one of the most common measures is the *accuracy*, which basically counts how often the classification is correct:

$$\text{accuracy} = \frac{\text{correct predictions}}{\text{the number of data points}} \tag{4}$$

and, of course, the correct predictions are the red numbers:

		Predicted		
		doric	**ionic**	**corinthian**
True	**doric**	10	2	0
	ionic	1	5	1
	corinthian	2	0	12

so accuracy is 27/33.

You can also extend precision and recall to the confusion matrix, there are different approachs to this, one being *macro-averaging*. Basically, say for precision, you calculate a precision for each column. For doric this would be measure of what fraction of things in the doric column are actually doric columns, here I realise I have picked a weirdly bad example, the first use of column meaning the column of the confusion matrix, the second meaning the tubular piece of stone we are looking at is in the doric style.

		Predicted		
		doric	**ionic**	**corinthian**
True	**doric**	10	2	0
	ionic	1	5	1
	corinthian	2	0	12

Here then the precision for doric is 10/13. Now the macro-average precision is just the average precision for all three columns:

$$\text{macro-average precision} = \frac{1}{3}\left(\frac{10}{13}+\frac{5}{7}+\frac{12}{13}\right) \tag{5}$$

The same thing can be done for recall: to work out recall for doric we want to know how many of all the doric columns have been classified as doric:

		Predicted		
		doric	**ionic**	**corinthian**
True	**doric**	10	2	0
	ionic	1	5	1
	corinthian	2	0	12

Now the recall is doric is 10/12 and the macro-average is

$$\text{macro-average recall} = \frac{1}{3}\left(\frac{10}{12}+\frac{5}{7}+\frac{12}{14}\right) \tag{6}$$

The *micro-averaging* strategy, in contrast, reduces the clustering to a positive versus negative problem, for example

		Predicted	
		ionic	**not ionic**
True	**ionic**	5	7
	not ionic	2	24

and then applies the original definitions of precision and recall to this new, smaller, precision matrix. Of course, there is lots of different ways to do this, like

		Predicted	
		doric	**not doric**
True	**doric**	10	2
	not doric	3	18

or even

		Predicted	
		doric or ionic	**neither doric nor ionic**
True	**doric or ionic**	18	2
	neither doric not ionic	1	12

This is by no means all the micro-averages and, of course, for more classes there will be even more possible micro-averages.

1 Summary

A classic problem classifies data as positive or negative. In this case there are four results, true positive, true negatives, false positive and false negative. Here we consider two ways of evaluating the classification based on these, the precision, which quantifies what fraction of the points classified as positive are really positive and the recall, which quantifies what fraction of all the positive points are identified as positive. F1 is also defined, it is the harmonic mean of precision and recall. For multi-class problems one measure of accuracy, the fraction of points that are correctly classifies. In these problems we look at the macro-average of precision and recall, for these you work out the precision or recall for each class and then average. For the micro-average, you reduce the table to a positive versus negative problem by lumping some categories together as positive and the rest as negative; you then work out precision or recall as before.