

Motivation

Last week you learned about linear regression, which predicts a number. This week, you learn about classification where your output variable y can take on only one of a small handful of possible values instead of any number in an infinite range of numbers. It turns out that linear regression is not a good algorithm for classification problems. Let's take a look at why and this will lead us into a different algorithm called logistic regression. Which is one of the most popular and most widely used learning algorithms today.

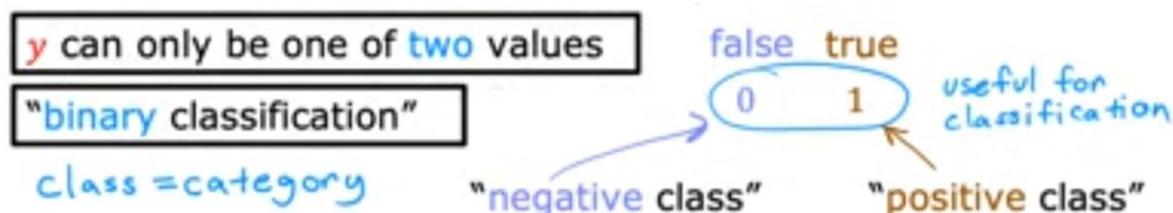
Here are some examples of classification problems recall the example of trying to figure out whether an email is spam. So the answer you want to output is going to be either a no or a yes. Another example would be figuring out if an online financial transaction is fraudulent. Fighting online financial fraud is something I once worked on and it was strangely exhilarating. Because I knew there were forces out there trying to steal money and my team's job was to stop them. So the problem is given a financial transaction.

Classification

Question	Answer " y "	
Is this email <u>spam?</u>	no	yes
Is the transaction <u>fraudulent?</u>	no	yes
Is the tumor <u>malignant?</u>	no	yes

Can your learning algorithm figure out if this transaction is fraudulent, such as what this credit card was stolen? Another example we've touched on before was trying to classify a tumor as malignant versus not. In each of these problems the variable that you want to predict can only be one of two possible values. No or yes. This type of classification problem where there are only two possible outputs is called binary classification.

Where the word binary refers to there being only two possible classes or two possible categories. In these problems I will use the terms class and category relatively interchangeably. They mean basically the same thing. By convention we can refer to these two classes or categories in a few common ways. We often designate clauses as no or yes or sometimes equivalently false or true or very commonly using the numbers zero or one. Following the common convention in computer science with zero denoting false and one denoting true.



I'm usually going to use the numbers zero and one to represent the answer y . Because that will fit in most easily with the types of learning algorithms we want to implement. But when we talk about it

will often say no or yes or false or true as well. One of the technologies commonly used is to call the false or zero class. The negative class and the true or the one class, the positive class. For example, for spam classification, an email that is not spam may be referred to as a negative example. Because the output to the question of is a spam.

The output is no or zero. In contrast, an email that has spam might be referred to as a positive training example. Because the answer to is it spam is yes or true or one to be clear, negative and positive. Do not necessarily mean bad versus good or evil versus good. It's just that negative and positive examples are used to convey the concepts of absence or zero or false vs the presence or true or one of something you might be looking for.

Such as the absence or presence of the spam illness or the spam property of an email or the absence of presence of broadening activity or absence of presence of malignancy of the tumor. Between non spam and spam emails. Which one you call false or zero and which one you call true or one is a little bit arbitrary. Often either choice could work. So, different engineer might actually swap it around and have the positive class B. The presence of a good email or the possible causes be the presence of a real financial transaction or a healthy patient.

Classification

Question	Answer "y"
Is this email <u>spam</u> ?	no yes
Is the transaction <u>fraudulent</u> ?	no yes
Is the tumor <u>malignant</u> ?	no yes

y can only be one of **two values**

"**binary classification**"

class = category

false true

useful for classification

"negative class"
≠ "bad"
absence

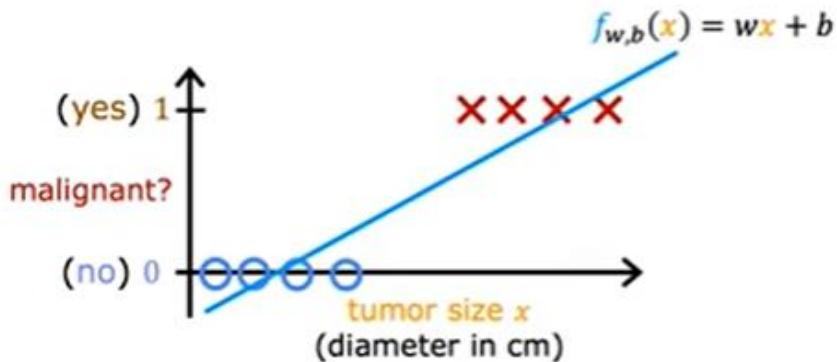
"positive class"
≠ "good"
presence

So how do you build a classification algorithm? Here's the example of a training set for classifying if the tumor is malignant. A class one, positive class, yes class or benign, class zero or negative class. I plotted both the tumor size on the horizontal axis as well as the label Y on the vertical axis.

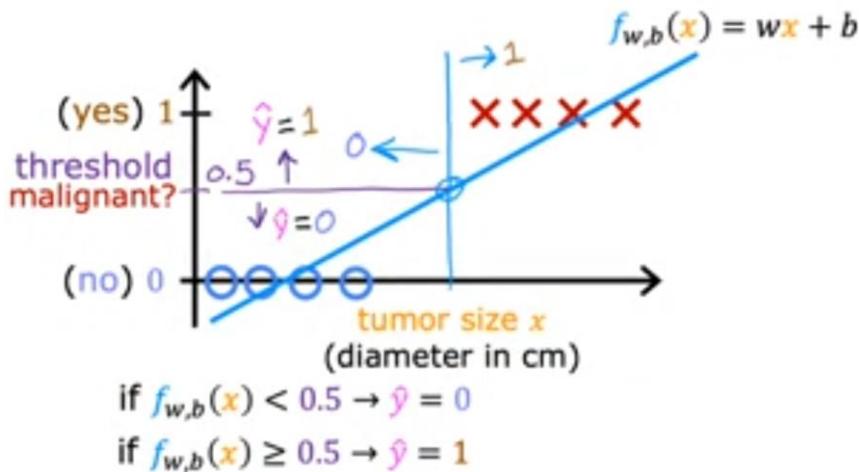
By the way, in week one, when we first talked about classification. This is how we previously visualized it on the number line except that now we're calling the classes zero.

And one and plotting them on the vertical axis. Now, one thing you could try on this training set is to apply the album you already know. Linear regression and try to fit a straight line to the data. If you do that, maybe the straight line looks like this, right?

And that's your F effects. Linear regression predicts not just the values zero and one. But all numbers between zero and one or even less than zero or greater than one. But here we want to predict categories. One thing you could try is to pick a threshold of say 0.5.



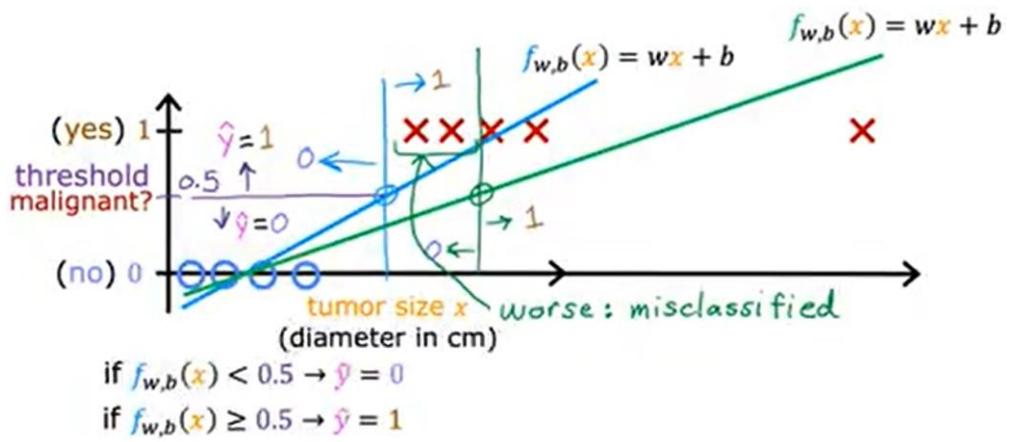
So that if the model outputs a value below 0.5, then you predict why equal zero or not malignant. And if the model outputs a number equal to or greater than 0.5, then predict Y equals one or malignant. Notice that this threshold value of 0.5 intersects the best fit straight line at this point. So if you draw this vertical line here, everything to the left ends up with a prediction of y equals zero. And everything on the right ends up with the prediction of y equals one. Now, for this particular data set it looks like linear regression could do something reasonable.



But now let's see what happens if your dataset has one more training example. This one way over here on the right. Let's also extend the horizontal axis. Notice that this training example shouldn't really change how you classify the data points. This vertical dividing line that we drew just now still makes sense as the cut off where tumors smaller than this should be classified as zero. And tumors greater than this should be classified as one.

But once you've added this extra training example on the right. The best fit line for linear regression will shift over like this. And if you continue using the threshold of 0.5, you now notice that everything to the left of this point is predicted at zero non malignant. And everything to the right of this point is predicted to be one or malignant.

This isn't what we want because adding that example way to the right shouldn't change any of our conclusions about how to classify malignant versus benign tumors. But if you try to do this with linear regression, adding this one example which feels like it shouldn't be changing anything. It ends up with us learning a much worse function for this classification problem. Clearly, when the tumor is large, we want the algorithm to classify it as malignant.



So what we just saw was linear regression causes the best fit line. When we added one more example to the right to shift over. And does the dividing line also called the decision boundary to shift over to the right. You learn more about the decision boundary in the next video, you also learn about an algorithm called logistic regression.

Where the output value of the outcome will always be between zero and one. And the average will avoid these problems that we're seeing on this slide. By the way one thing confusing about the name logistic regression is that even though it has the word of regression in it is actually used for classification. Don't be confused by the name which was given for historical reasons. It's actually used to solve binary classification problems with output label y is either zero or one.

