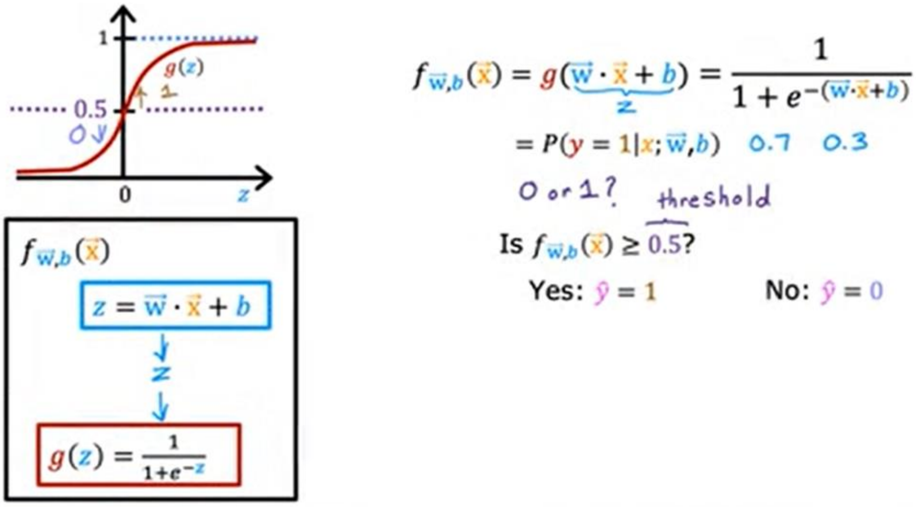
Limite de décision

To recap, here's how the logistic regression models outputs are computed in two steps. In the first step, you compute z as w.x plus b. Then you apply the Sigmoid function g to this value z. Here again, is the formula for the Sigmoid function. Another way to write this is we can say f of x is equal to g, the Sigmoid function, also called the logistic function, applied to w.x plus b, where this is of course, the value of z. If you take the definition of the Sigmoid function and plug in the definition of z, then you find that f of x is equal to this formula over here, 1 over 1 plus e to the negative z, where z is wx plus b.

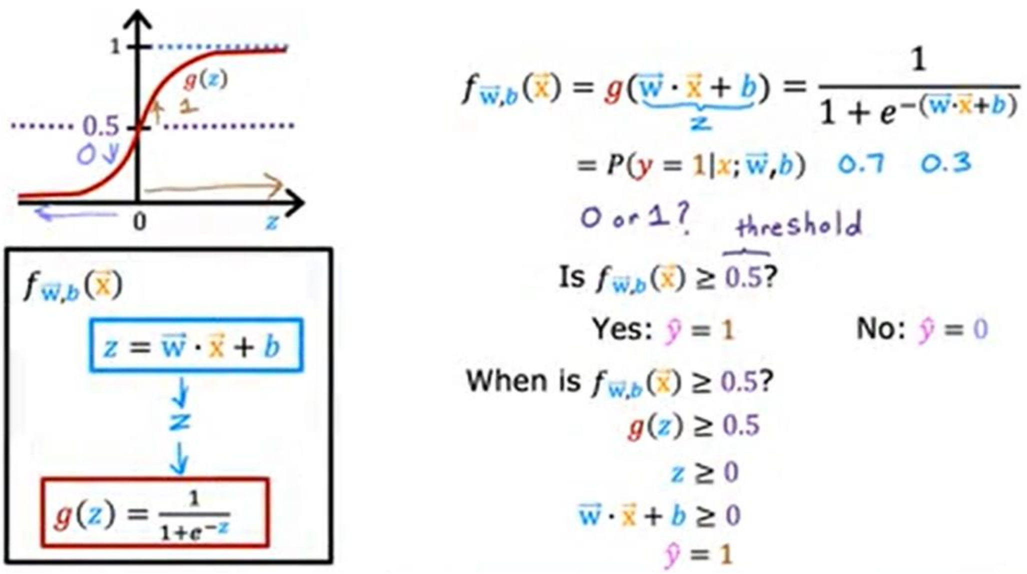
You may remember we said in the previous video that we interpret this as the probability that y is equal to 1 given x and with parameters w and b. This is going to be a number like maybe a 0.7 or 0.3. Now, what if you want to learn the algorithm to predict. Is the value of y going to be zero or one? Well, one thing you might do is set a threshold above which you predict y is one, or you set y hat to prediction to be equal to one and below which you might say y hat, my prediction is going to be equal to zero.

A common choice would be to pick a threshold of 0.5 so that if f of x is greater than or equal to 0.5, then predict y is one. We write that prediction as y hat equals 1, or if f of x is less than 0.5, then predict y is 0, or in other words, the prediction y hat is equal to 0.



Now, let's dive deeper into when the model would predict one. In other words, when is f of x greater than or equal to 0.5. We'll recall that f of x is just equal to g of z. So f is greater than or equal to 0.5 whenever g of z is greater than or equal to 0.5. But when is g of z greater than or equal to 0.5? Well, here's a Sigmoid function over here. So g of z is greater than or equal to 0.5 whenever z is greater than or equal to 0. That is whenever z is on the right half of this axis.

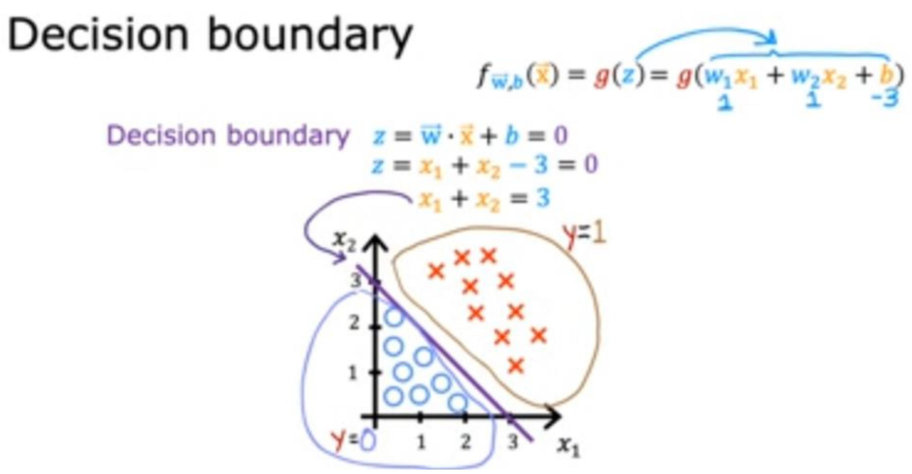
Finally, when is z greater than or equal to zero? Well, z is equal to w.x plus b, so z is greater than or equal to zero whenever w.x plus b is greater than or equal to zero. To recap, what you've seen here is that the model predicts 1 whenever w.x plus b is greater than or equal to 0. Conversely, when w.x plus b is less than zero, the algorithm predicts y is 0.



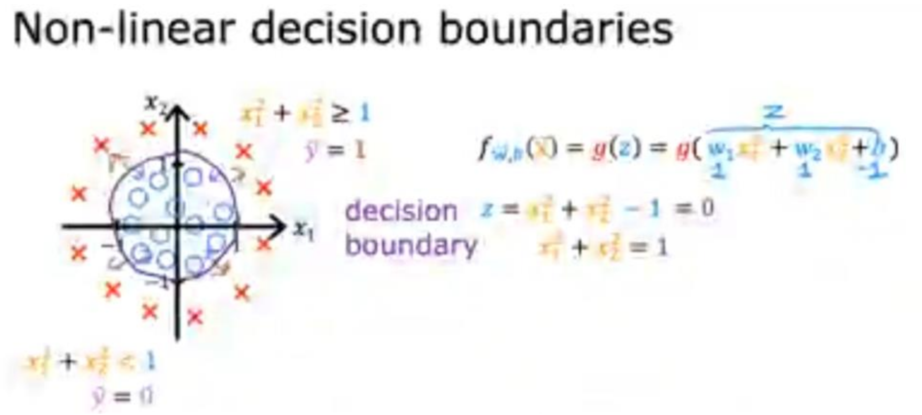
Given this, let's now visualize how the model makes predictions. I'm going to take an example of a classification problem where you have two features, x1 and x2 instead of just one feature. Here's a training set where the little red crosses denote the positive examples and the little blue circles denote negative examples. The red crosses corresponds to y equals 1, and the blue circles correspond to y equals 0.

The logistic regression model will make predictions using this function f of x equals g of z, where z is now this expression over here, w1x1 plus w2x2 plus b, because we have two features x1 and x2. Let's just say for this example that the value of the parameters are w1 equals 1, w2 equals 1, and b equals negative 3. Let's now take a look at how logistic regression makes predictions. In particular, let's figure out when wx plus b is greater than equal to 0 and when wx plus b is less than 0.

To figure that out, there's a very interesting line to look at, which is when wx plus b is exactly equal to 0. It turns out that this line is also called the decision boundary because that's the line where you're just almost neutral about whether y is 0 or y is 1. Now, for the values of the parameters w_1, w_2, and b that we had written down above, this decision boundary is just x_1 plus x_2 minus 3. When is x_1 plus x_2 minus 3 equal to 0?



Well, that will correspond to the line x_1 plus x_2 equals 3, and that is this line shown over here. This line turns out to be the decision boundary, where if the features x are to the right of this line, logistic regression would predict 1 and to the left of this line, logistic regression with predicts 0. In other words, what we have just visualize is the decision boundary for logistic regression when the parameters w_1, w_2, and b are 1,1 and negative 3. Of course, if you had a different choice of the parameters, the decision boundary would be a different line.



Now let's look at a more complex example where the decision boundary is no longer a straight line. As before, crosses denote the class y equals 1, and the little circles denote the class y equals 0. Earlier last week, you saw how to use polynomials in linear regression, and you can do the same in logistic regression. This set z to be w_1, x_1 squared plus w_2, x_2 squared plus b. With this choice of features, polynomial features into a logistic regression.

F of x, which equals g of z, is now g of this expression over here. Let's say that we ended up choosing w_1 and w_2 to be 1 and b to be negative 1. Z is equal to 1 times x_1 squared plus 1 times x_2 squared minus 1. The decision boundary, as before, will correspond to when z is equal to 0. This expression will be equal to 0 when x_1 squared plus x_2 squared is equal to 1.

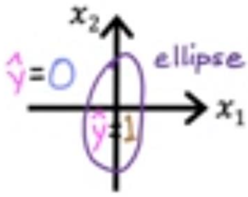
If you plot on the diagram on the left, the curve corresponding to x_1 squared plus x_2 squared equals 1, this turns out to be the circle. When x_1 squared plus x_2 squared is greater than or equal to 1, that's this area outside the circle and that's when you predict y to be 1. Conversely, when x_1 squared plus x_2 squared is less than 1, that's this area inside the circle and that's when you predict y to be 0.

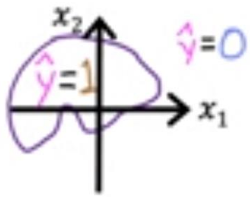
Can we come up with even more complex decision boundaries than these? Yes, you can. You can do so by having even higher-order polynomial terms. Say z is w_1, x_1 plus w_2, x_2 plus w_3, x_1 squared plus w_4, x_1, x_2 plus w_5, x_2 squared. Then it's possible you can get even more complex decision boundaries. The model can define decision boundaries, such as this example, an ellipse just like this, or with a different choice of the parameters.

You can even get more complex decision boundaries, which can look like functions that maybe looks like that. So this is an example of an even more complex decision boundary than the ones we've seen previously. This implementation of logistic regression will predict y equals 1 inside this shape and outside the shape will predict y equals 0. With these polynomial features, you can get very complex decision boundaries.

In other words, logistic regression can learn to fit pretty complex data. Although if you were to not include any of these higher-order polynomials, so if the only features you use are x_1, x_2, x_3, and so on, then the decision boundary for logistic regression will always be linear, will always be a straight line.

Non-linear decision boundaries





$$f_{\vec{w},b}(\vec{x}) = g(z) = g(w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_1x_2 + w_5x_2^2 + w_6x_1^3 + \cdots + b)$$