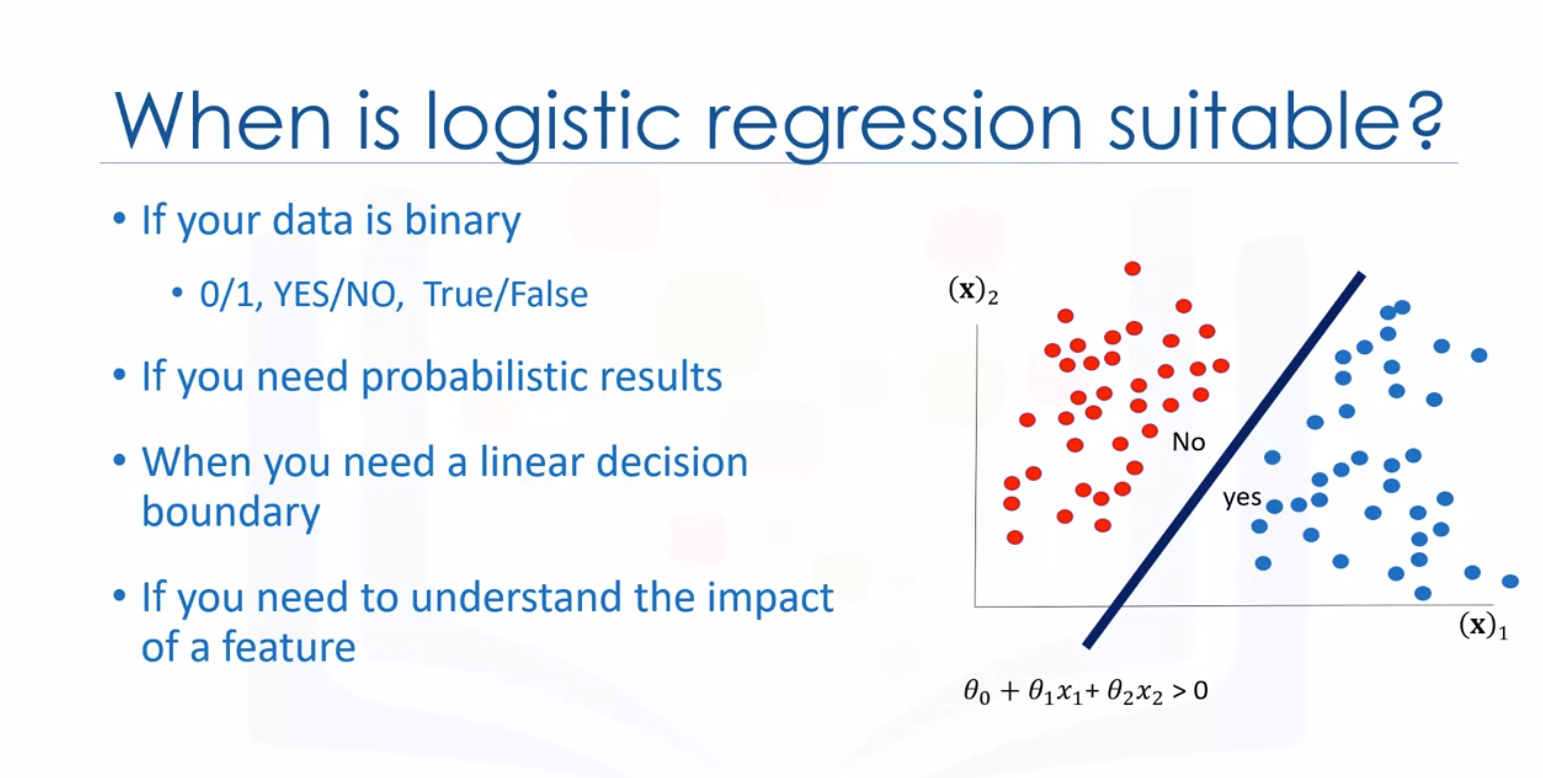


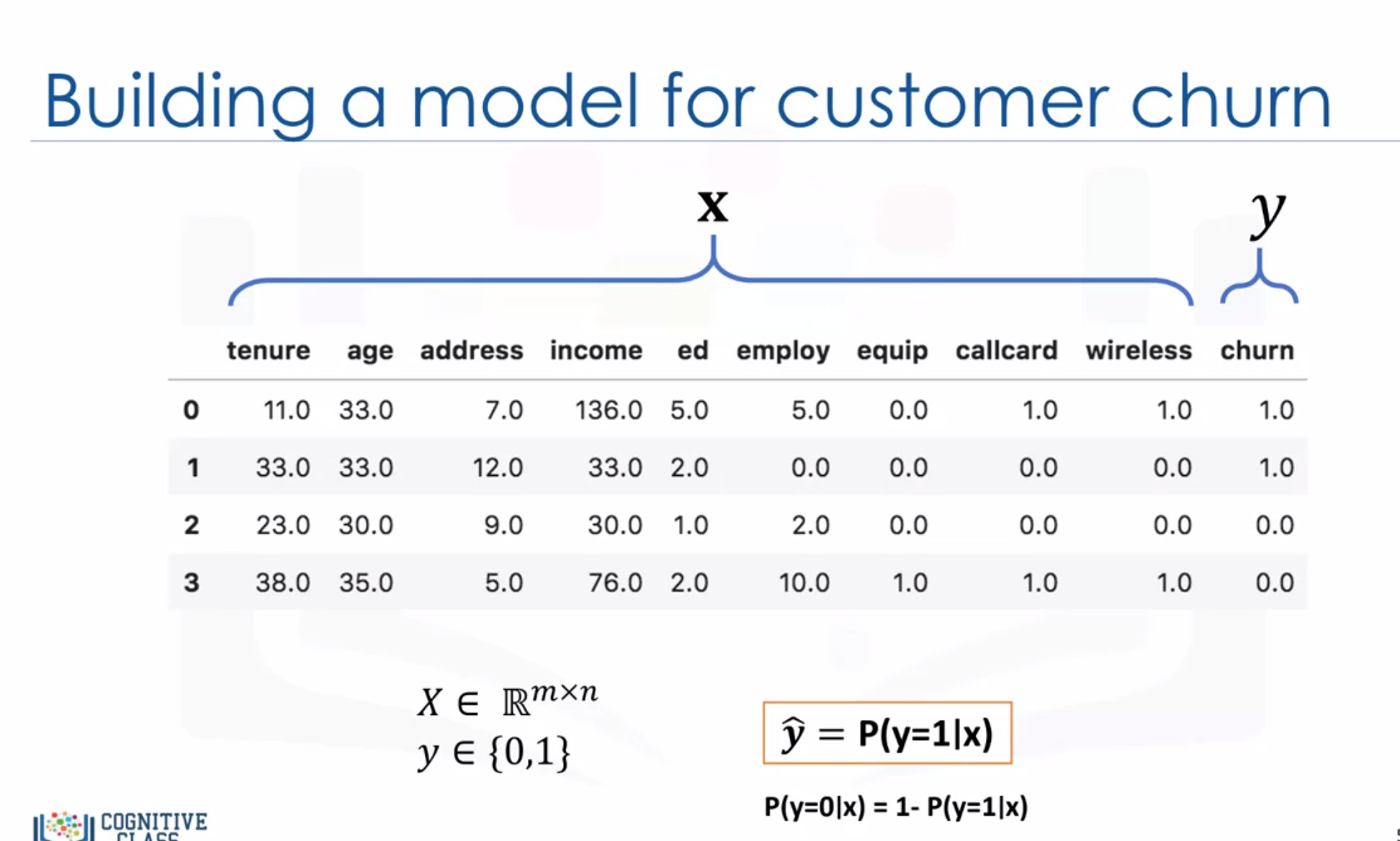
Logistic regression is a type of classification algorithm, is analogous to linear regression but tries to predict a categorical or discrete target field instead of a numeric one.

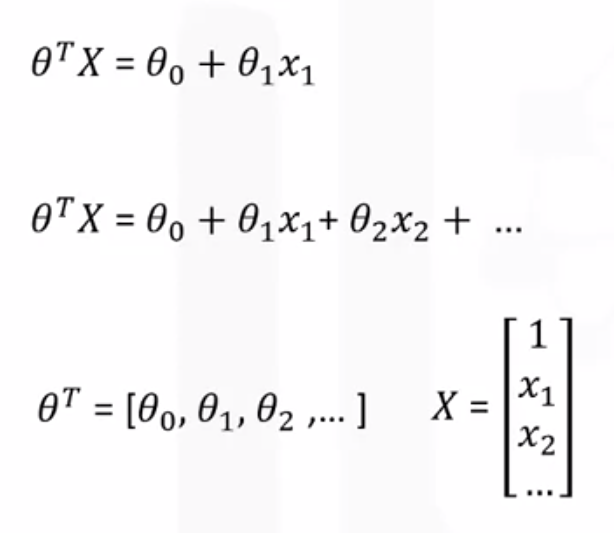
In logistic regression dependent variables should be continuous.

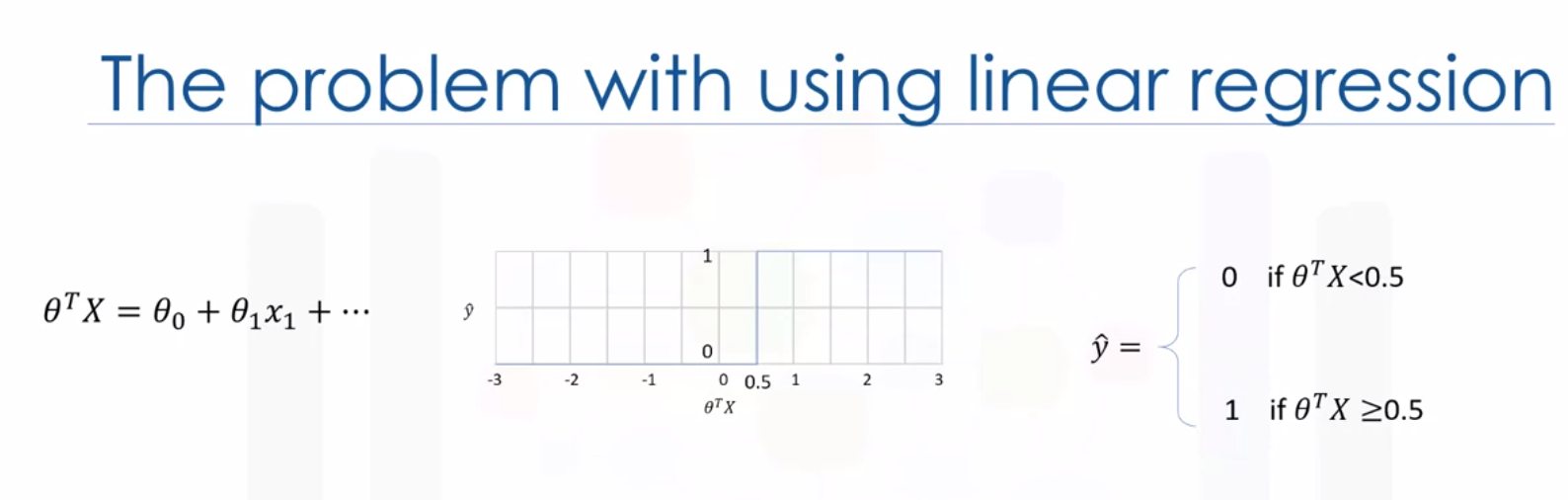
If categorical, they should be dummy or indicator coded. This means we have to transform them to some continuous value.

Please note that logistic regression can be used for both binary classification and multi-class classification.

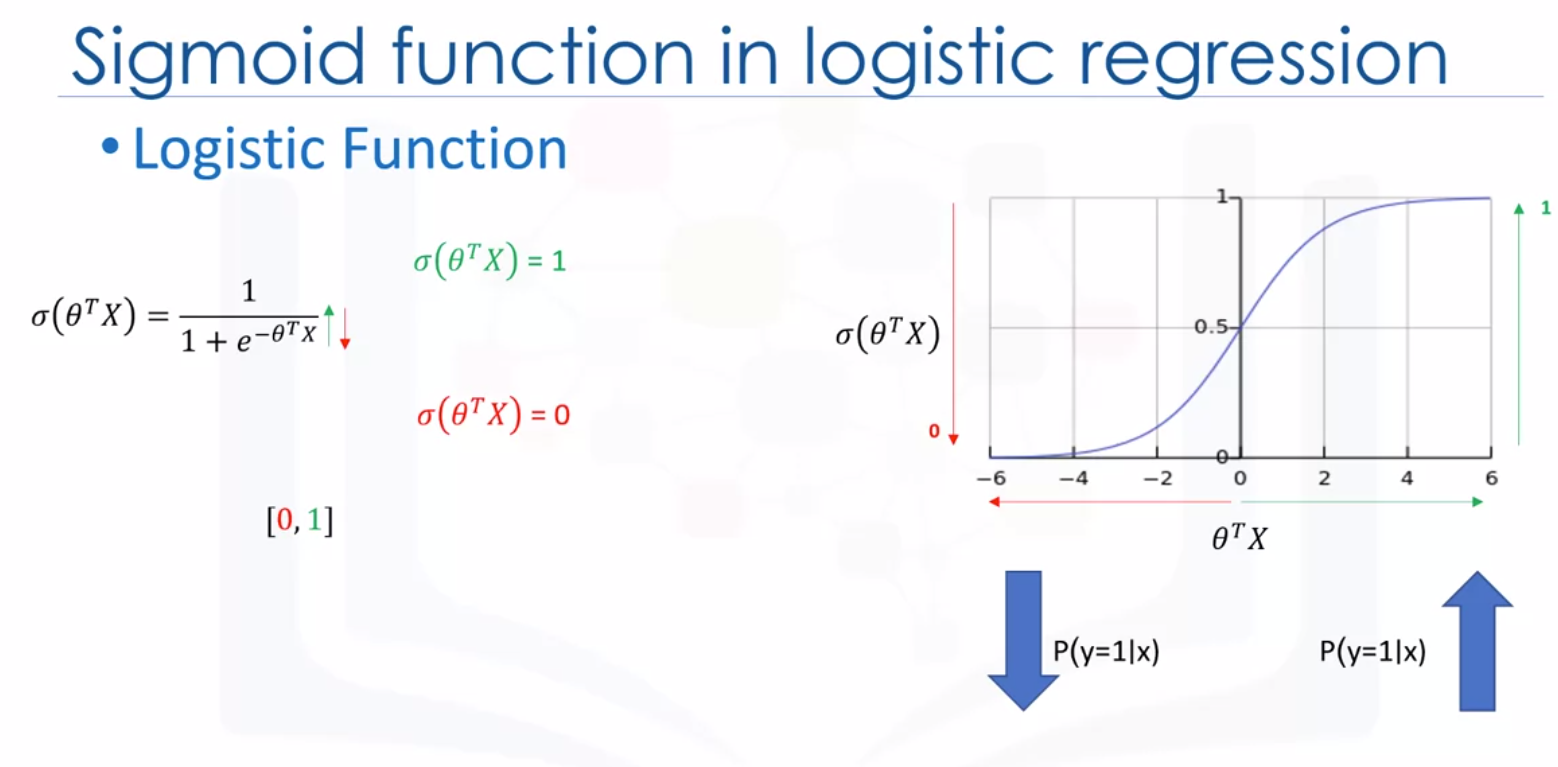


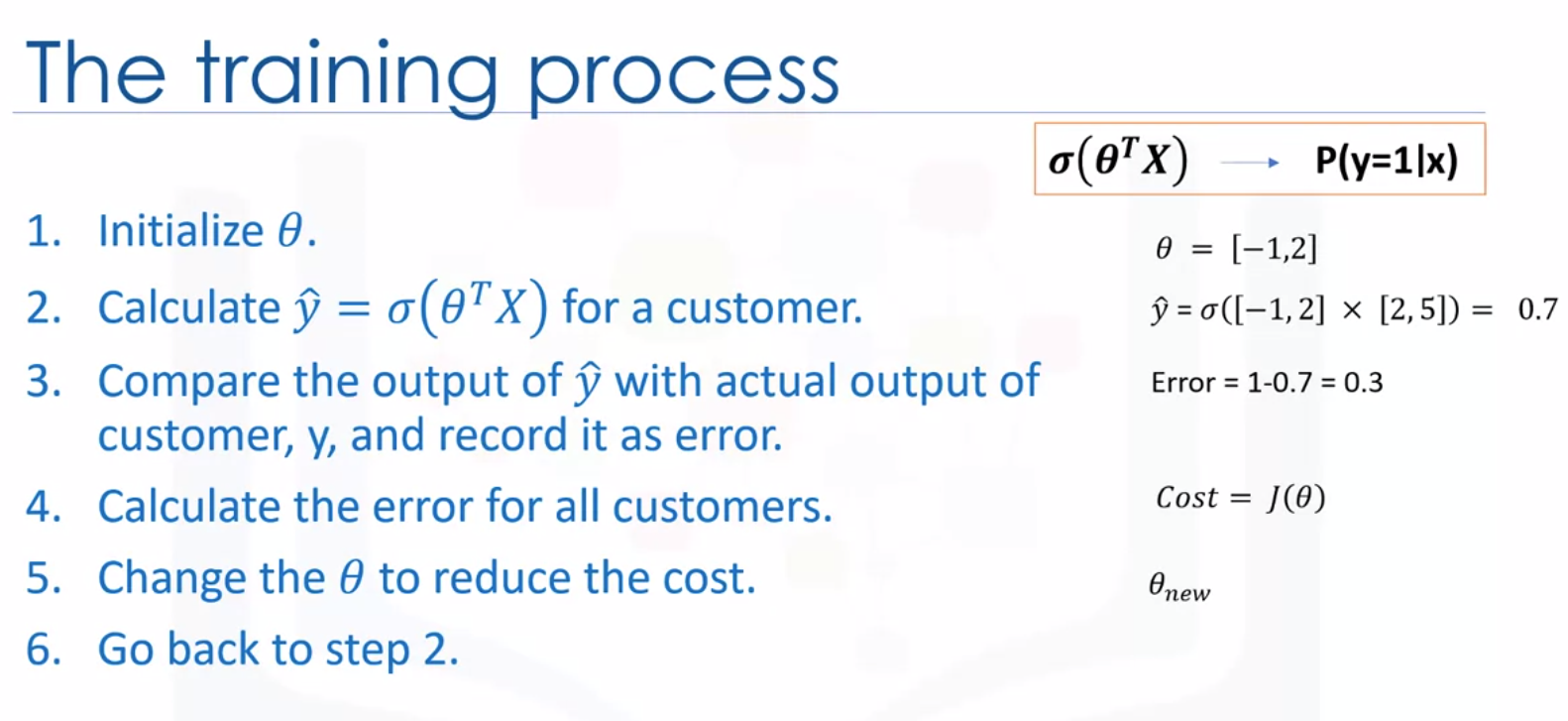






Linear regression in classification problem: you can’t properly measure the probability of a case belonging to a class. No difference with output of 1 and 1000.



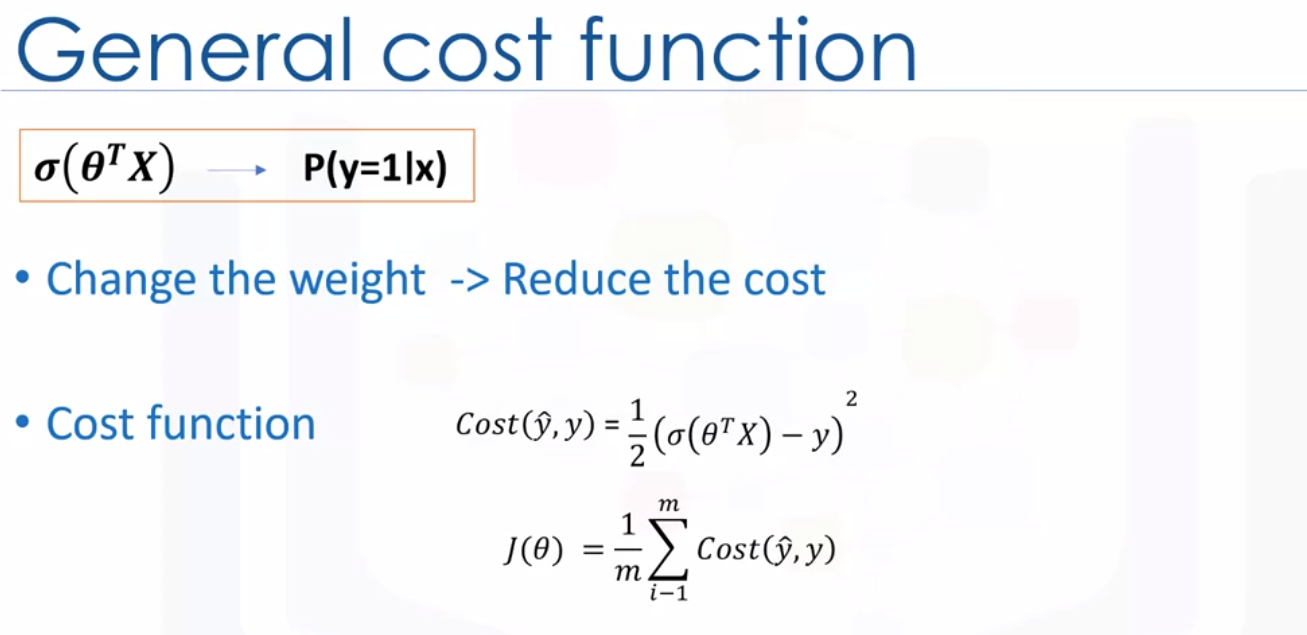


1. Initialize theta vector with random values as with most ML algorithms.

6. Go back to step 2, then we start another iteration and calculate the cost of model again. We keep doing those steps over and over, changing the values of theta each time, until the cost is low enough.

There are different ways to change the values of Theta, but one of the most popular ways is gradient descent.

There are various ways to stop iterations, but essentially you stop training by calculating the accuracy of your model and stop it when it’s satisfactory.



Using the derivative of the cost function, we can find how to change the parameters to reduce the cost or rather the error.

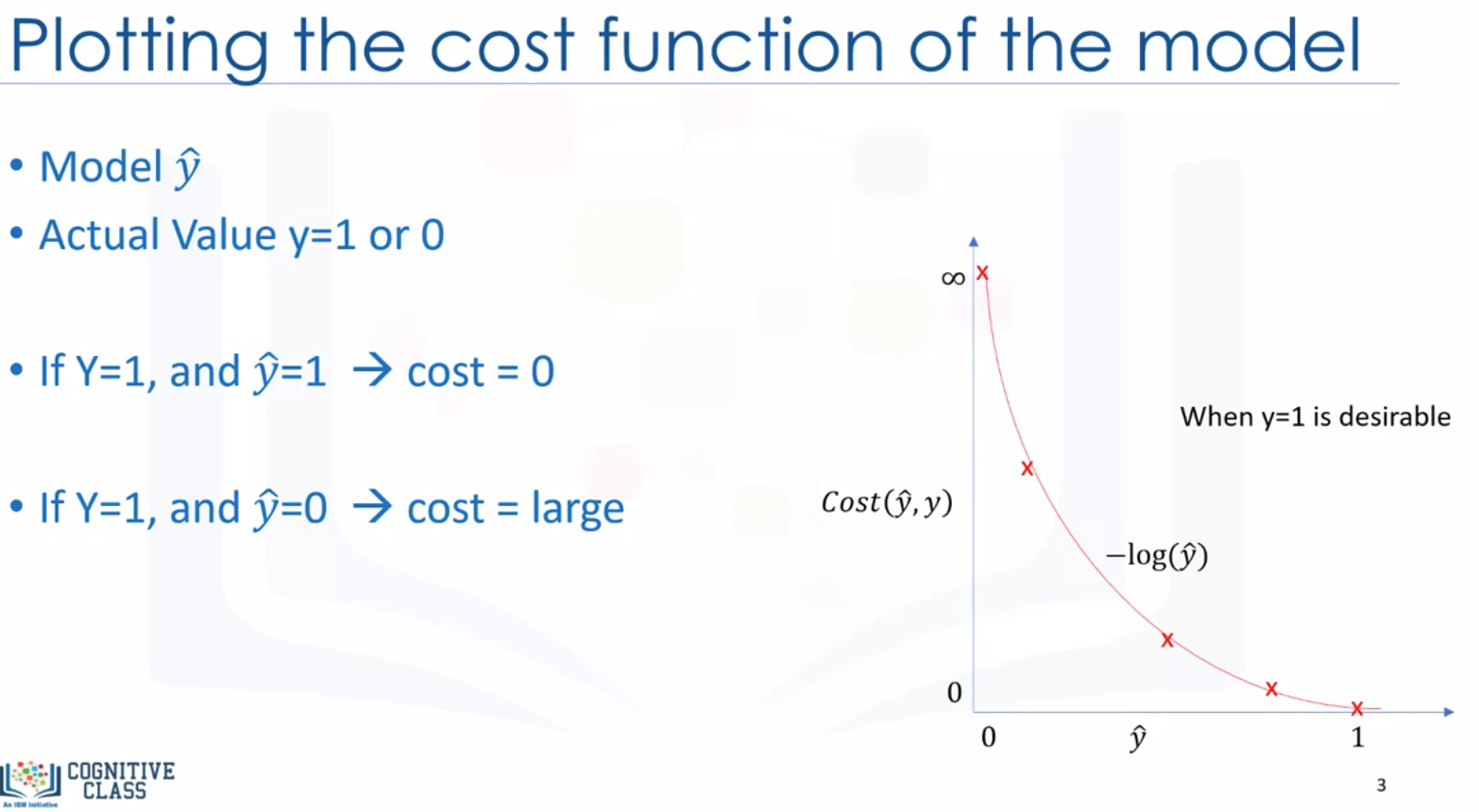
Usually the square of this equation is used because of the possibility of the negative result ad for the sake of simplicity, half of this value is considered as the cost function through the derivative process.

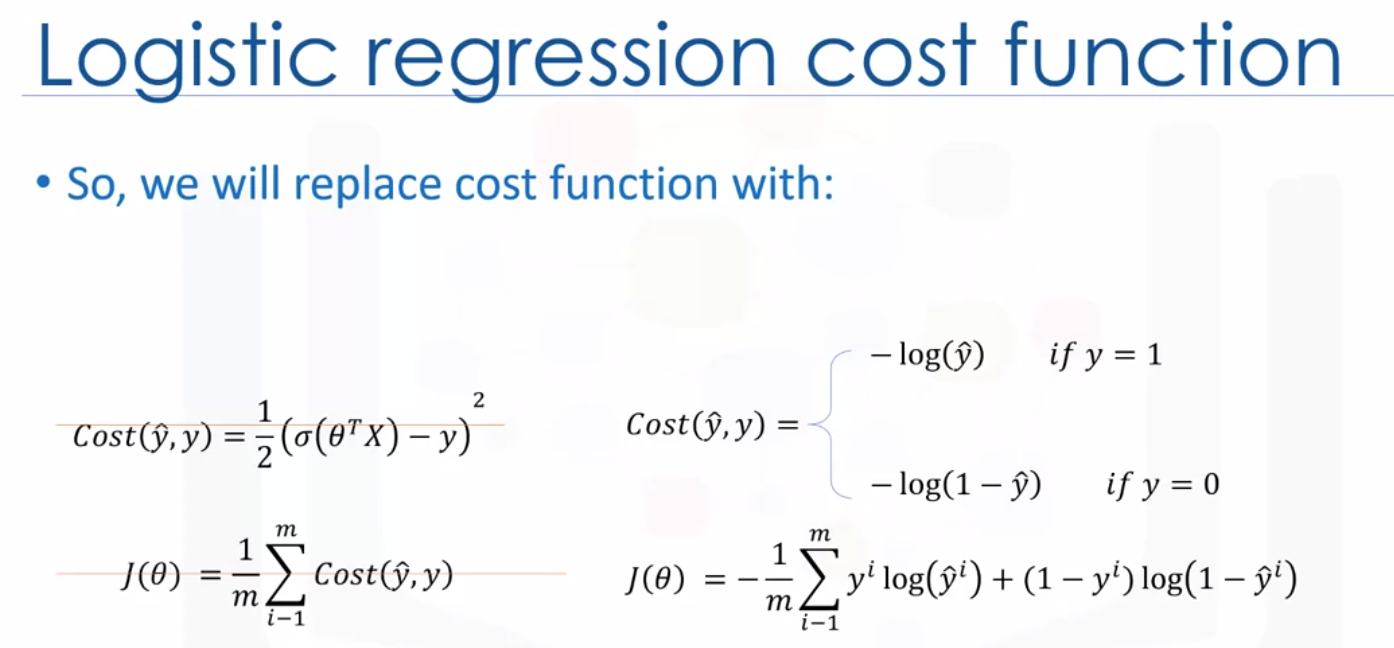
How do we find or set the best weights or parameters that minimize this cost function?

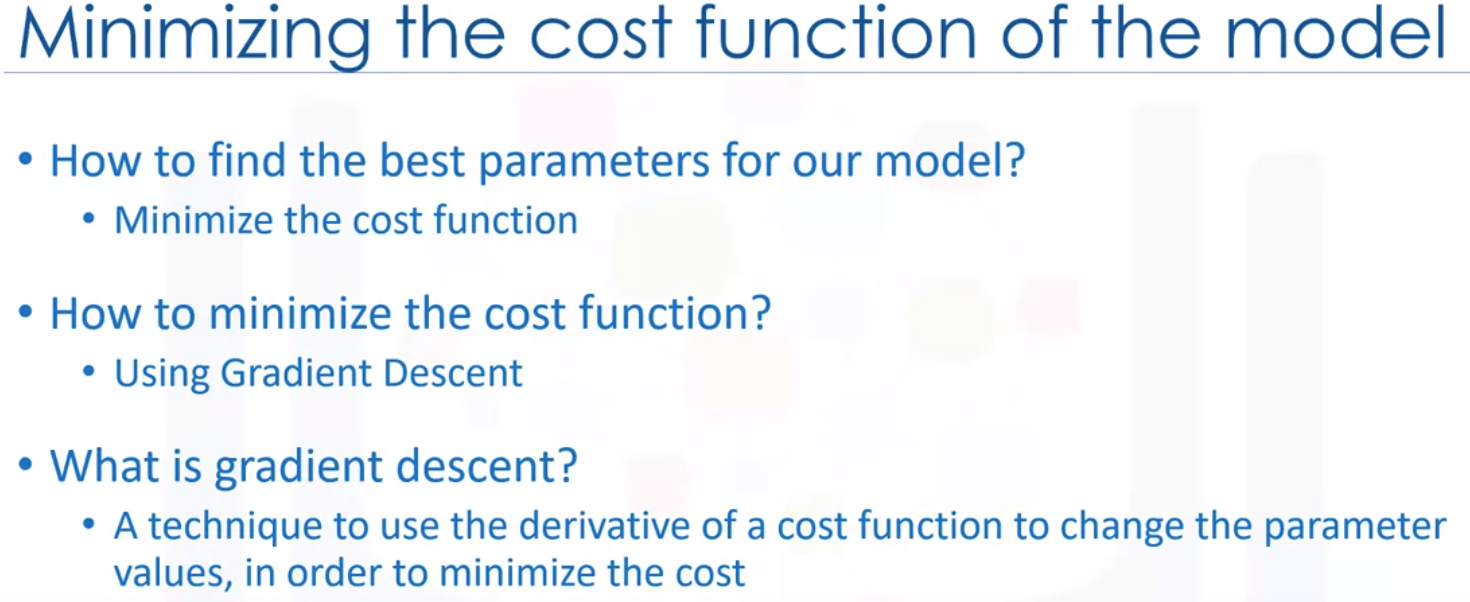
We should calculate the minimum point of this cost function and it will show us the best parameters for our model.

Although we can find the minimum point of a function using the derivative of a function, there’s not an easy way to find the global minimum point for such an equation.

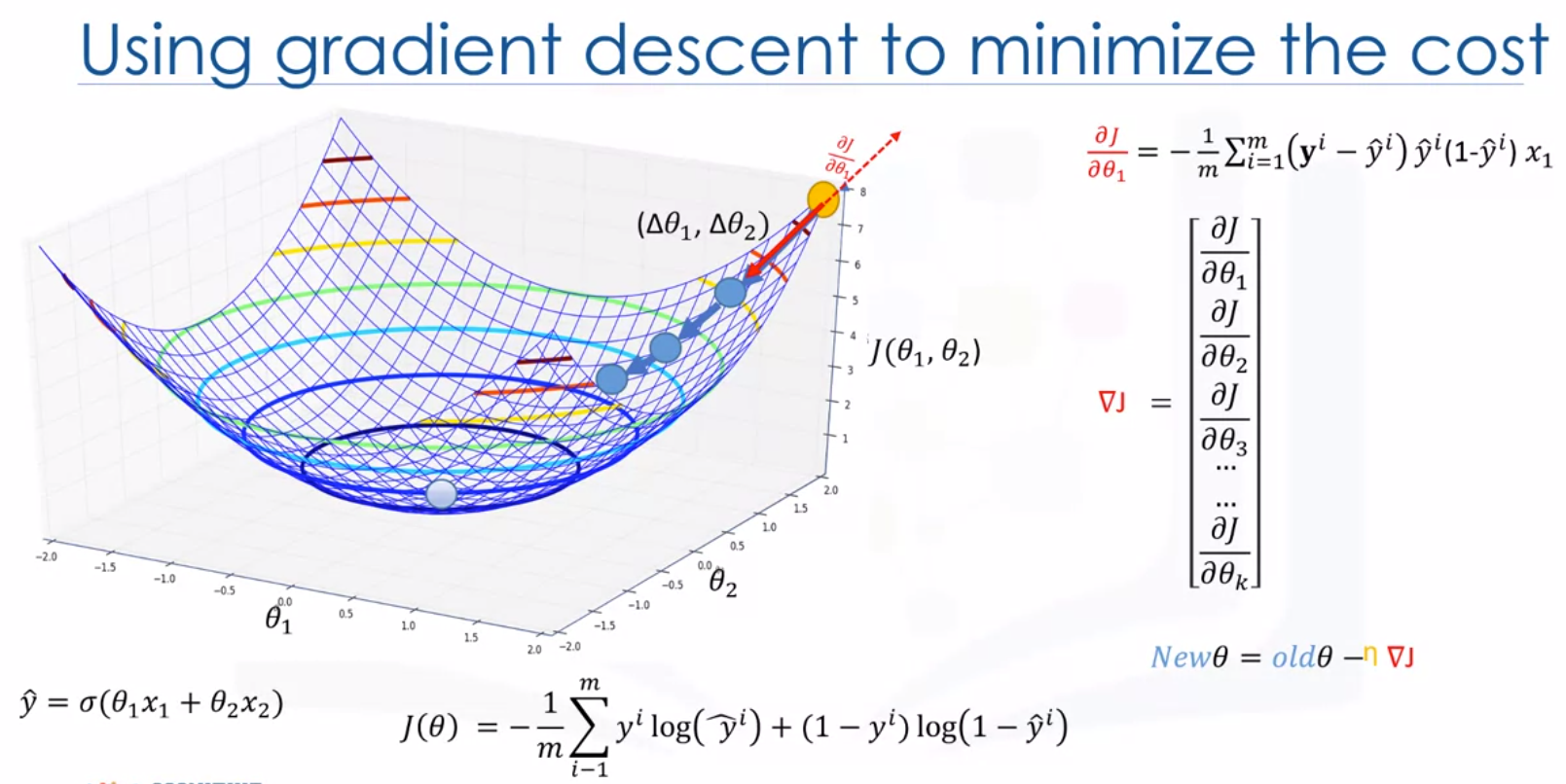
Well, we find another cost function instead, one which has the same behavior but is easier to find its minimum point.







Generally, gradient descent is an iterative approach to finding the minimum of a function. Specifically in our case, gradient descent is a technique to use the derivative of a cost function to change the parameter values to minimize the cost or error.



You can select random parameter values that locate a point on the bowl. You can think of our starting point being the yellow point.

You change the parameters by delta theta1 and delta theta2, and take one step on the surface.

The steeper the slope, the further we can step, and we can keep taking steps.

As we approach the lowest point, the slope diminishes, so we can take smaller steps until we reach a flat surface. This is the minimum point of our cure and the optimum theta1, theta2.

To find the direction and size of these steps, in other words, to find how to update the parameters, you should calculate the gradient of the cost function at that point. The gradient is the slope of the surface at every point and the direction of the gradient is the direction of the greatest uphill.

If you select a random point on this surface, for example the yellow point, and take the partial derivative of J of theta with respect to each parameter at that point, it gives you the slope of the move for each parameter at that point.

Now, if we move in the opposite direction of that slope, it guarantees that we go down in the error curve.

Gradient descent takes increasingly smaller steps towards the minimum with each iteration.

So in a nutshell, this equation returns the slope of that point and we should update the parameter in the opposite direction of the slope. A vector of all these slopes is the gradient vector, and we use this vector to change or update all the parameters.

We take the previous values of the parameters and subtract the error derivative. This results in the new parameters for theta that we know will decrease the cost.

Also we multiple the gradient value by a constant values Mu, which is called the learning rate.

Learning rate gives us additional control on how fast we move on the surface. In sum, gradient descent is like taking steps in the current direction of the slope, and the learning rate is like the length of the step you take.

Notice that it’s an iterative operation and in each iteration we update the parameters and minimize the cost until the algorithm converge is on an acceptable minimum.

