**Classification**

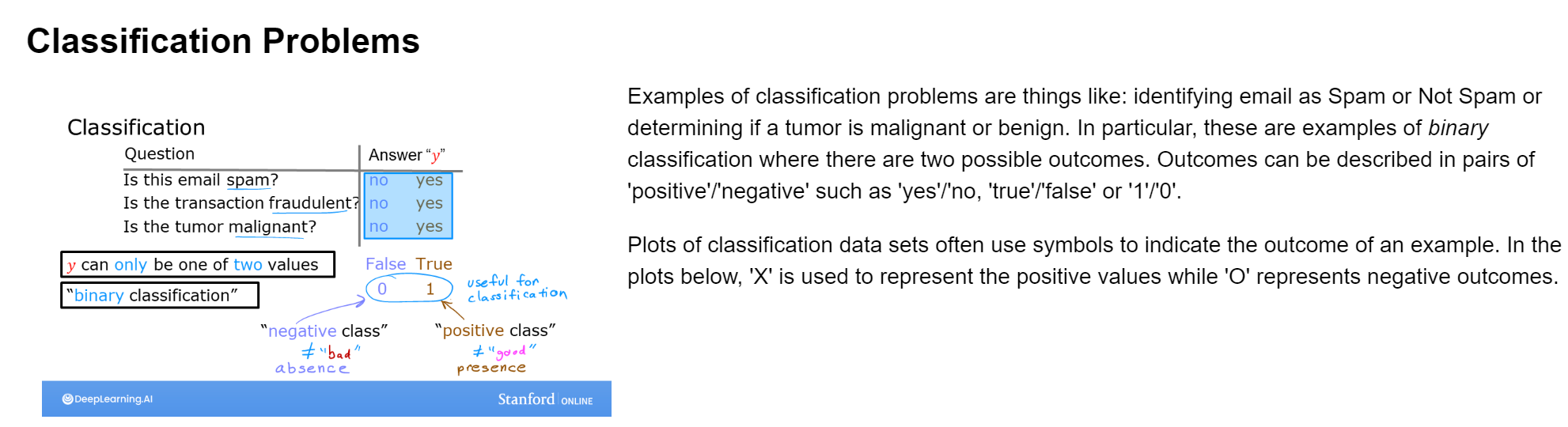
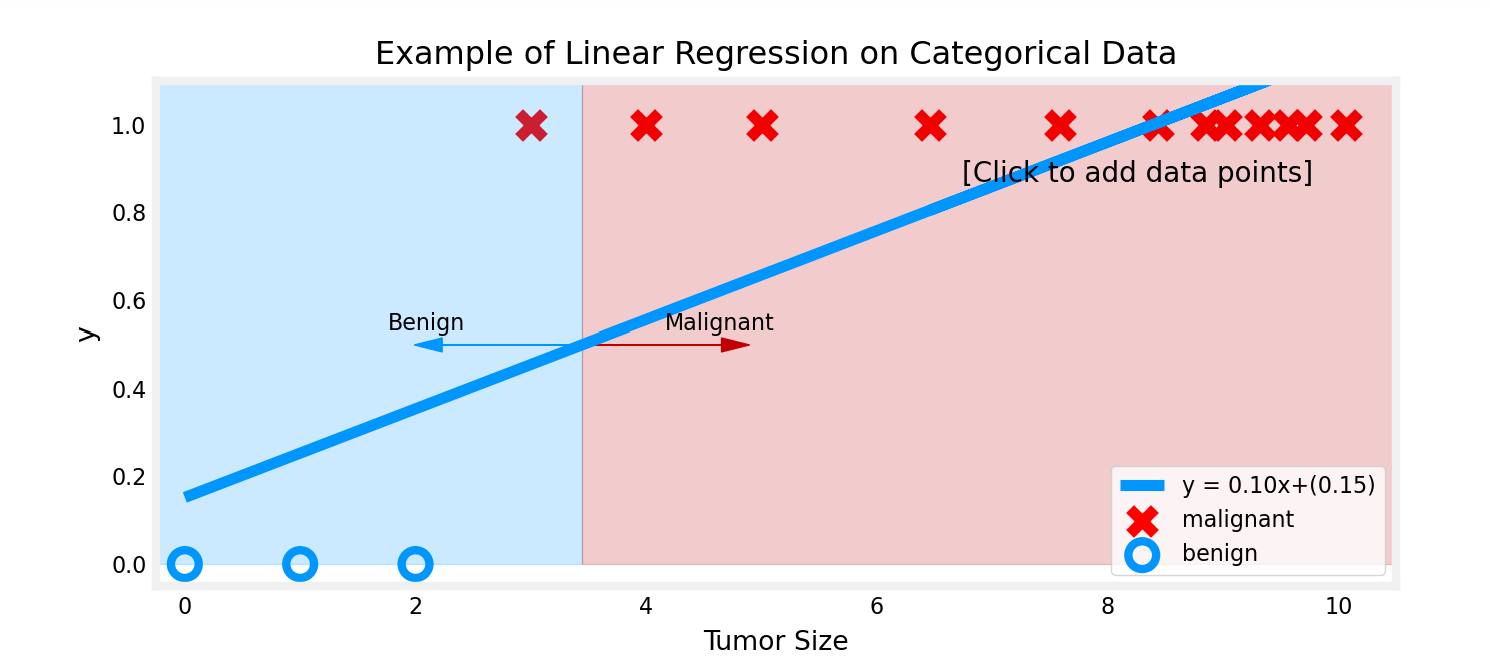
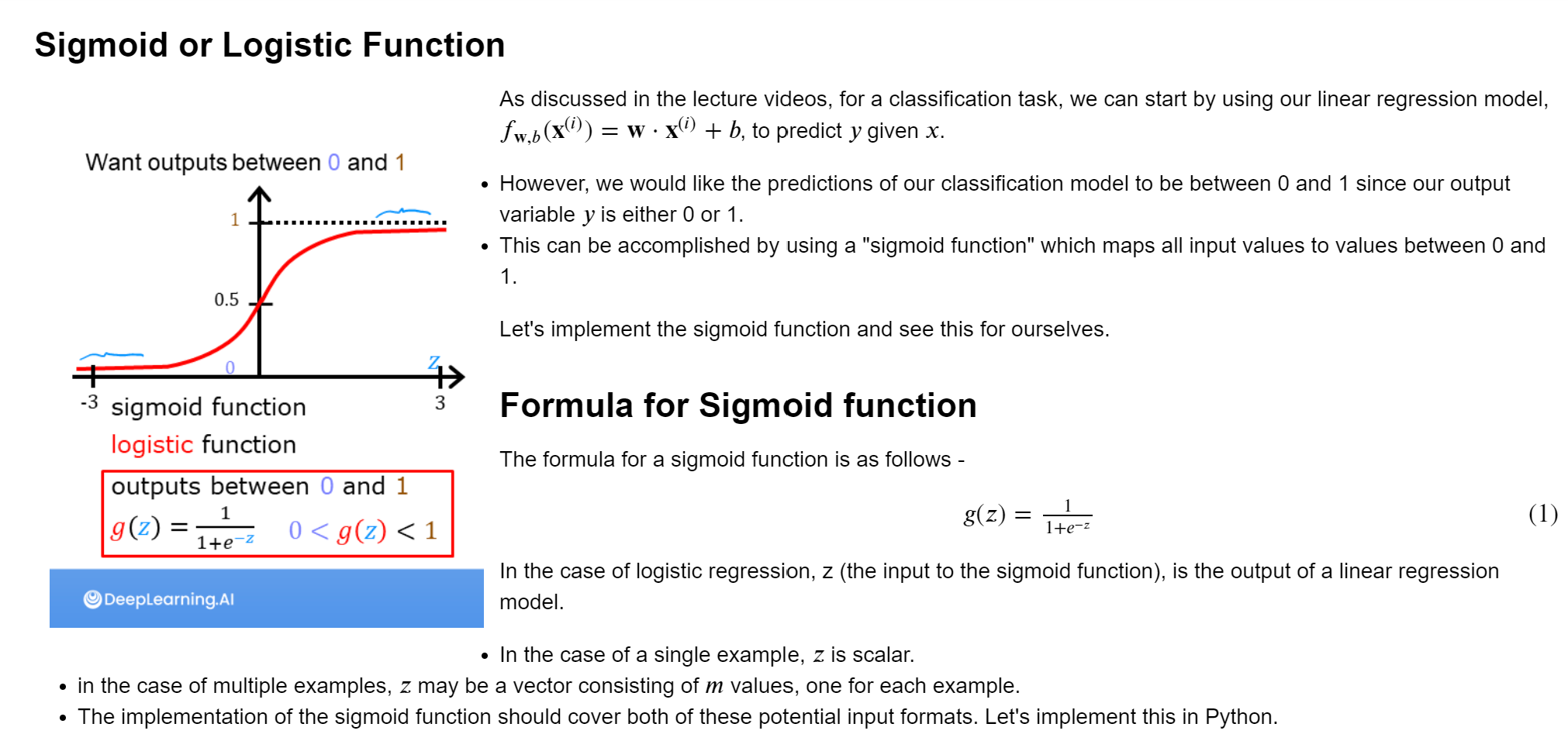


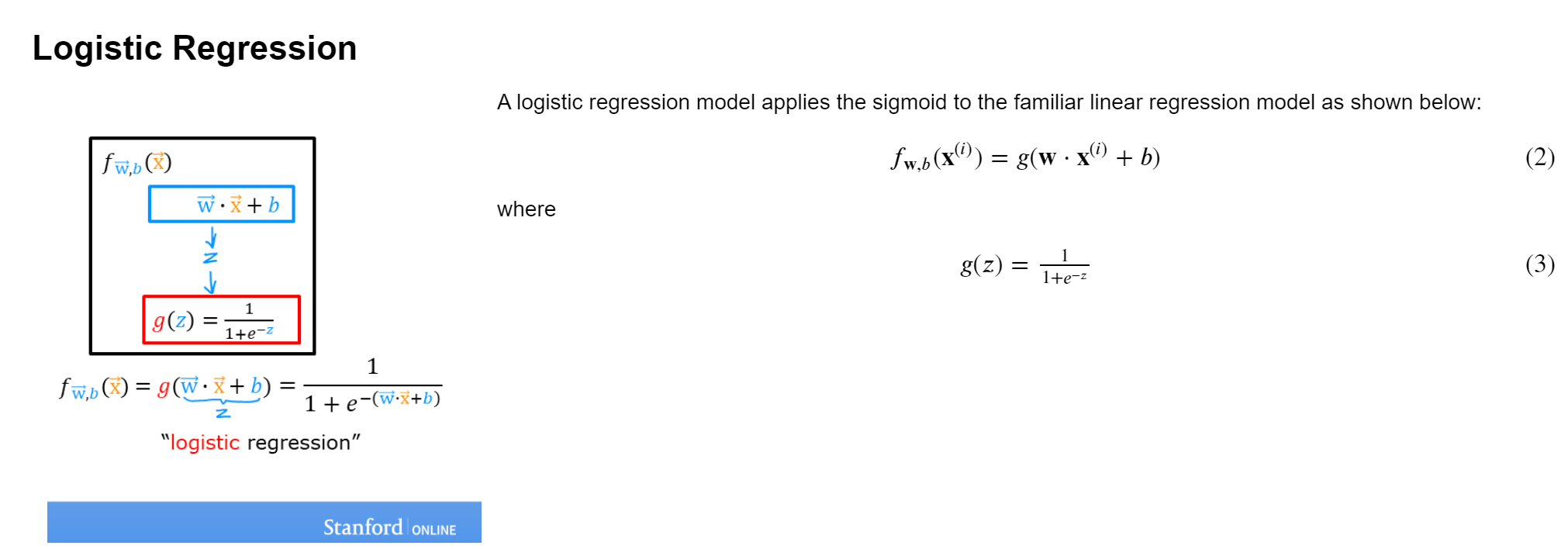
Figure 1 Classifications problems

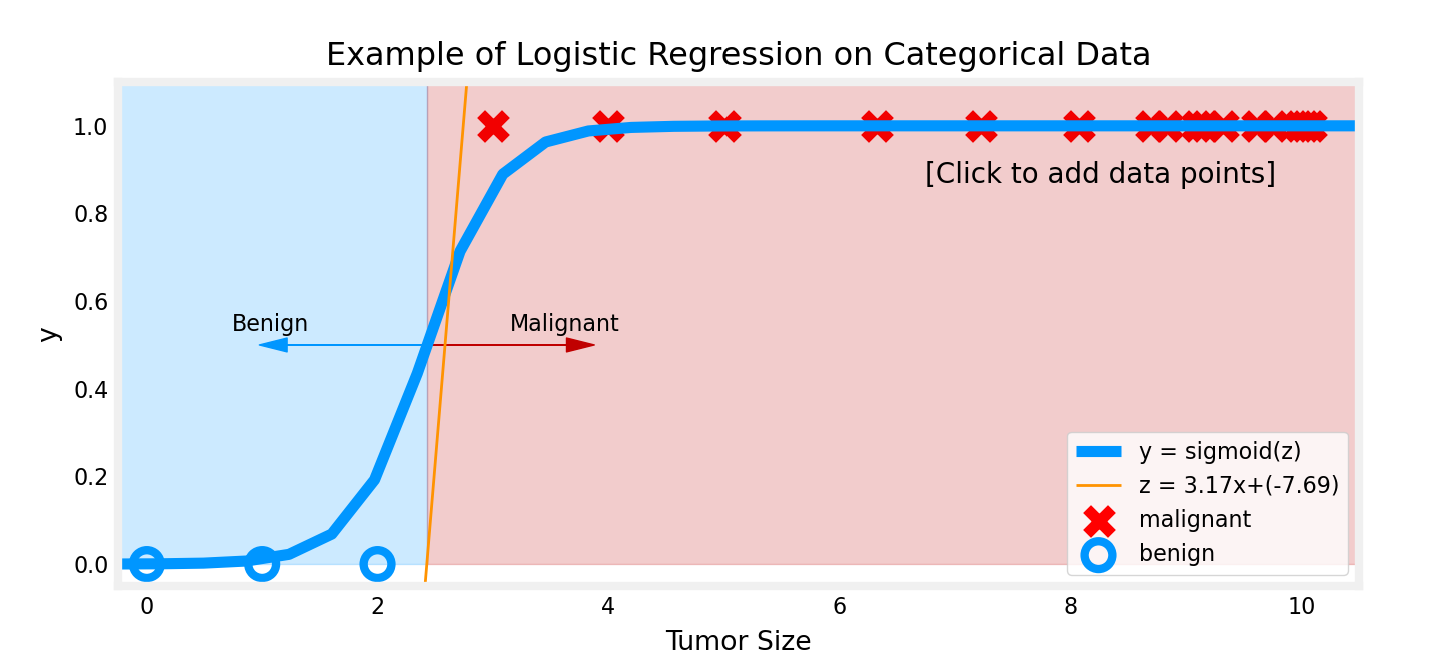


The example above demonstrates that the linear model is insufficient to model categorical data. The smallest malignant tumor is consider benign despite being bigger than the biggest benign tumor.

**Logistic regression**







Unlike linear regression, when big tumor size examples are added, the model still make correct prediction on the smallest malignant tumor.

* We interpret the output of the model f\_w,b(x) ss the probability that y=1 given x and parameterized by w and b.
  + Therefore, to get a final prediction (y=0 or y=1) from the logistic regression model, we can use the following heuristic -

if fw,b(x) >= 0.5, predict y=1

if fw,b(x) < 0.5, predict y=0

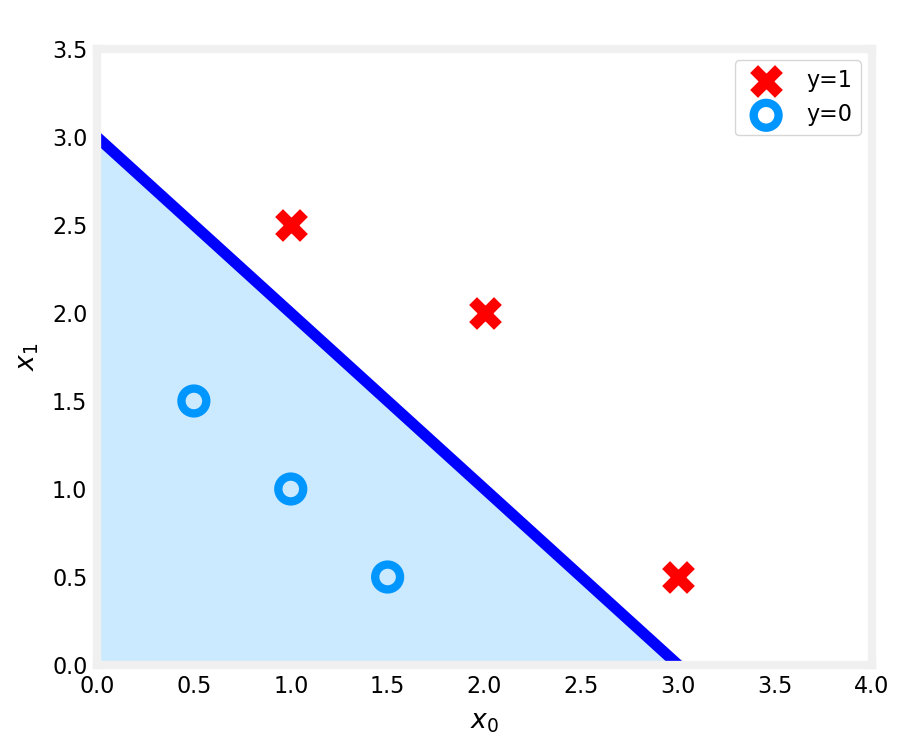
* g(z) >= 0.5 for z >= 0
* For a logistic regression model, z=w⋅x+b. Therefore,
  + if w⋅x+b >= 0, the model predicts y=1
  + if w⋅x+b < 0, the model predicts y=0

**Plotting decision boundary**

Now, let's go back to our example to understand how the logistic regression model is making predictions.

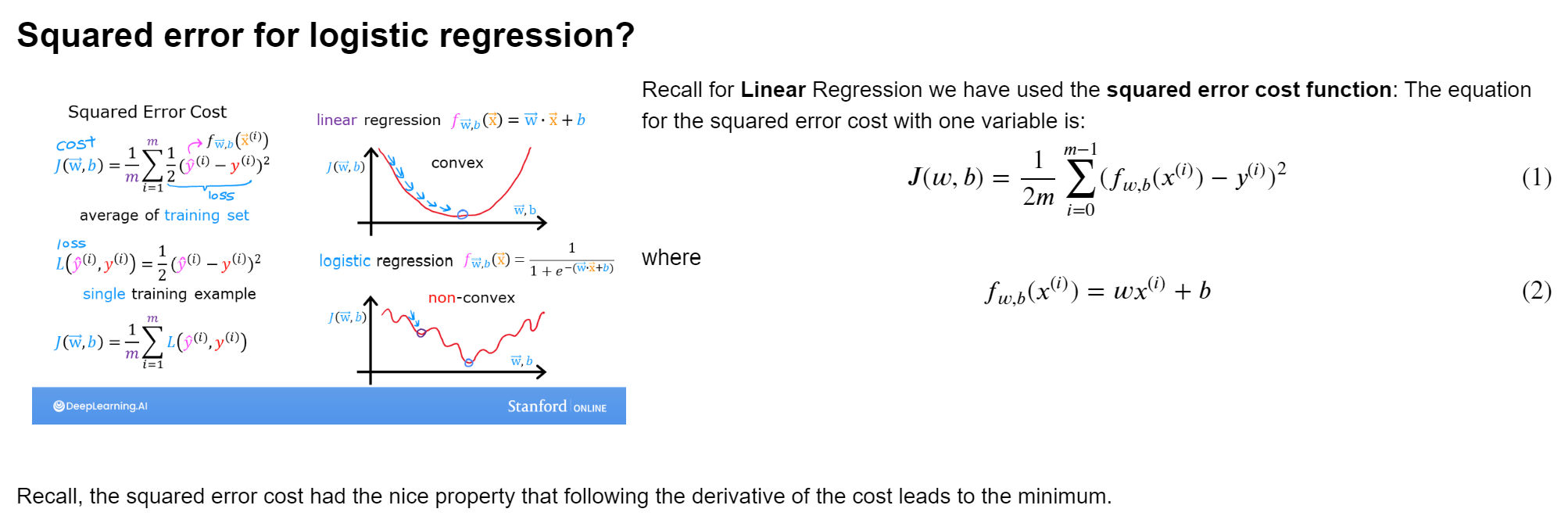
* Our logistic regression model has the form : f(x)=g(−3+x0+x1)
* From what you've learnt above, you can see that this model predicts :
  + y=1 if -3 + x\_0 + x\_1 >= 0

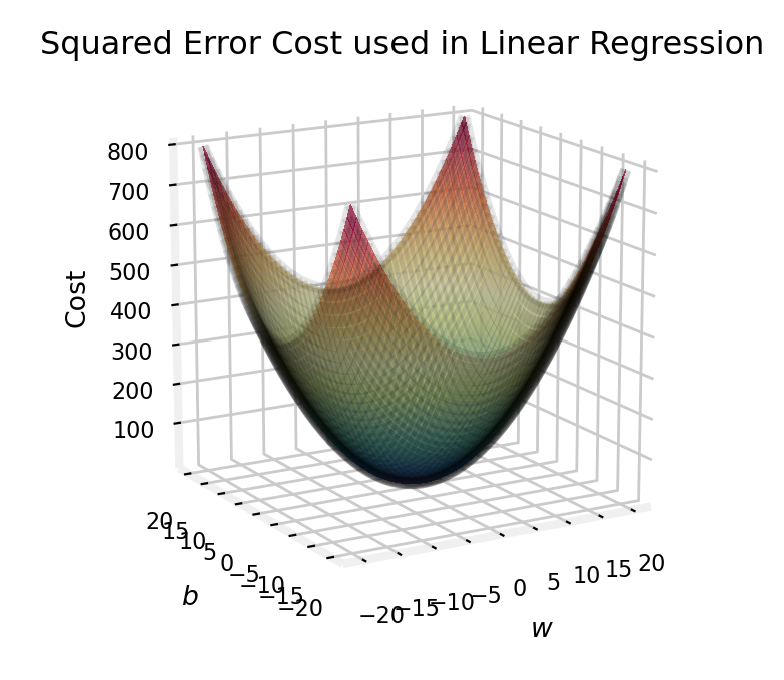
Let's see what this looks like graphically. We'll start by plotting -3 + x\_0 + x\_1 = 0, which is equivalent to x\_1 = 3 – x\_0.



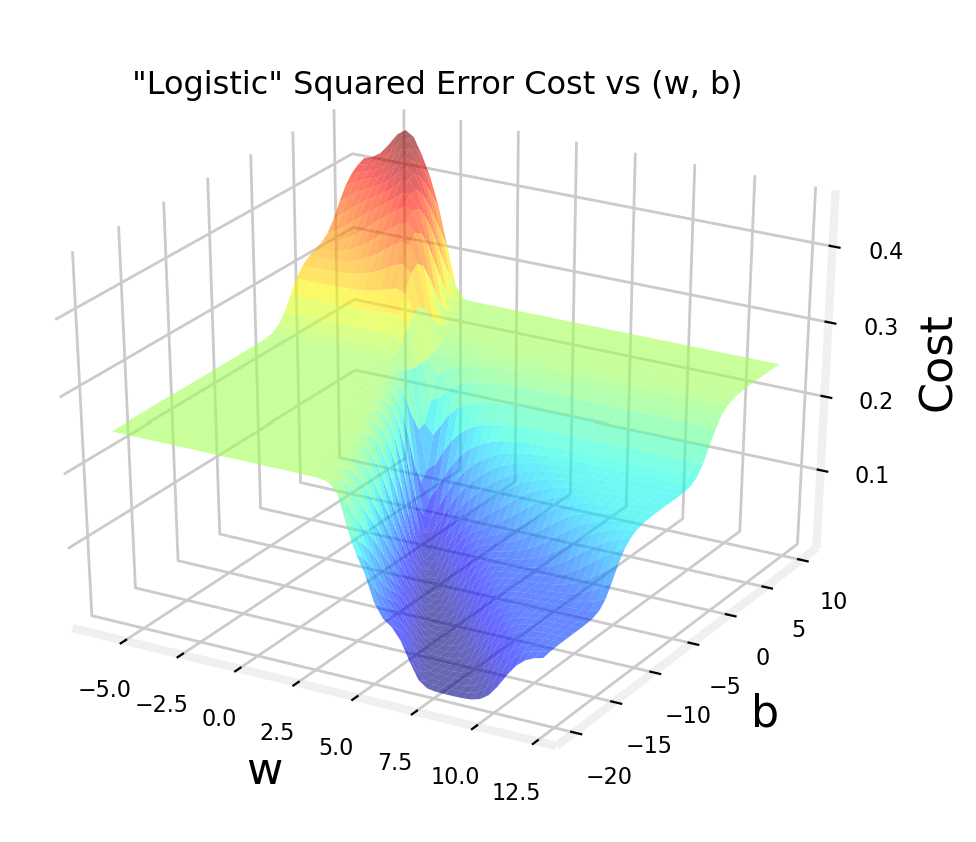
In the plot the blue line represents de decision boundary -3 + x\_0 + x\_1 = 0. What it bellow is classified as y = 0 and what is above is classified as y = 1. This line is known as the decision boundary. As seen before, by using more complex polynomial terms (ex : f(x) = g(x\_0^2 + x\_1 − 1), we can come up with more complex non-linear decision boundaries.

**Logistic Regression, Logistic Loss**

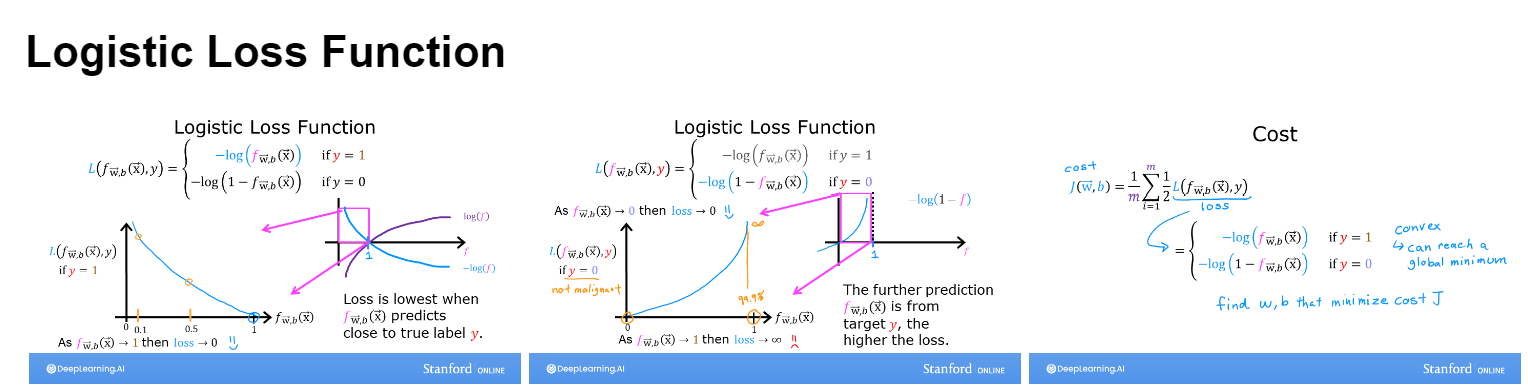




This cost function worked well for linear regression, it is natural to consider it for logistic regression as well. However, as the slide above points out, f\_wb(x) now has a non-linear component, the sigmoid function : f\_w,b(x(i)) = sigmoid(w \* x(i) + b). Let's try a squared error cost on the example from an earlier lab, now including the sigmoid.

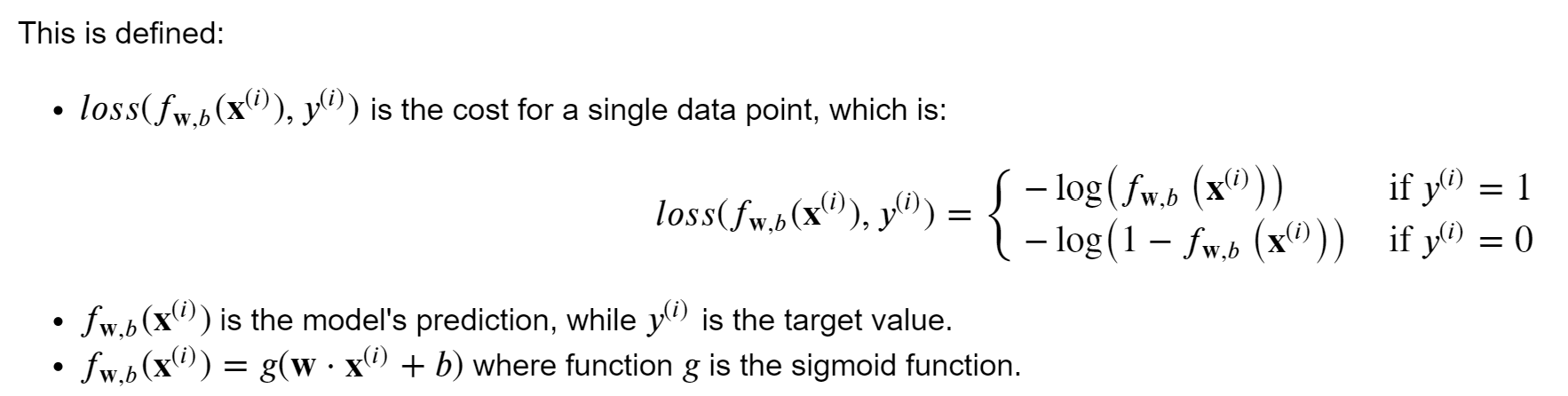


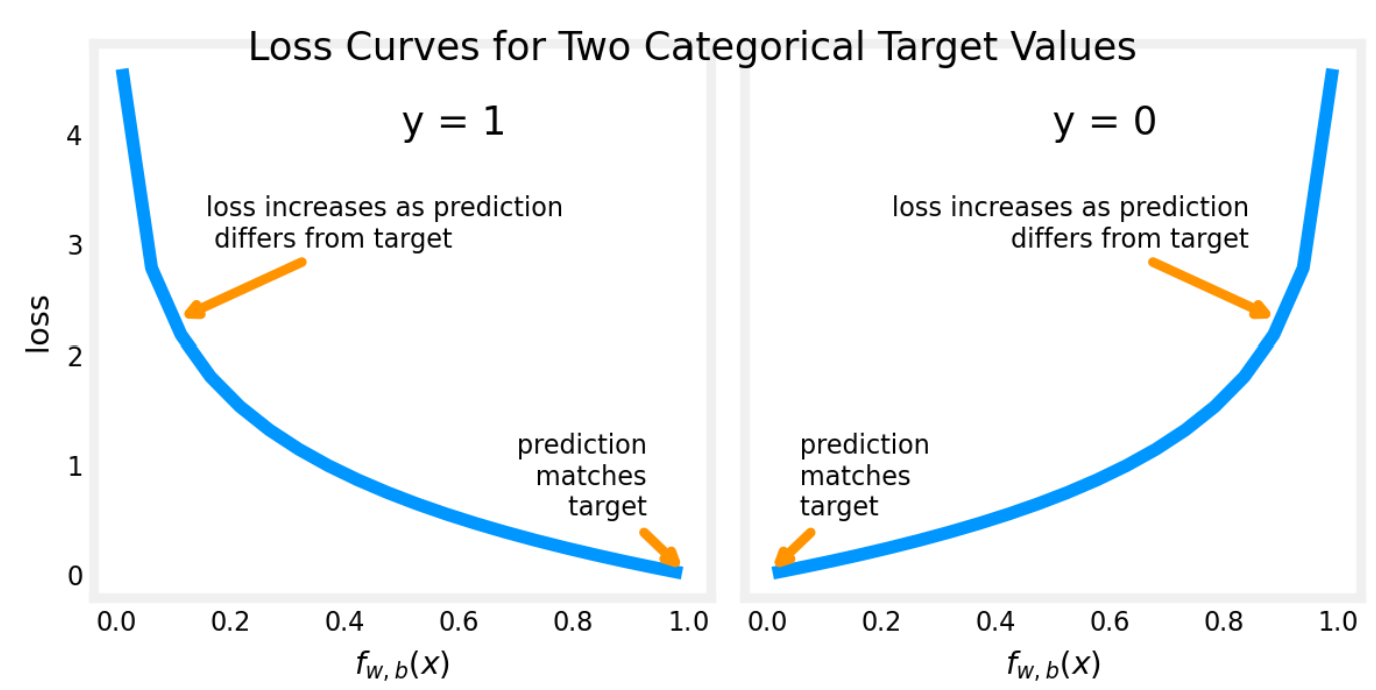
While this produces a pretty interesting plot, the surface above not nearly as smooth as the 'soup bowl' from linear regression. Logistic regression requires a cost function more suitable to its non-linear nature. This starts with a Loss function.

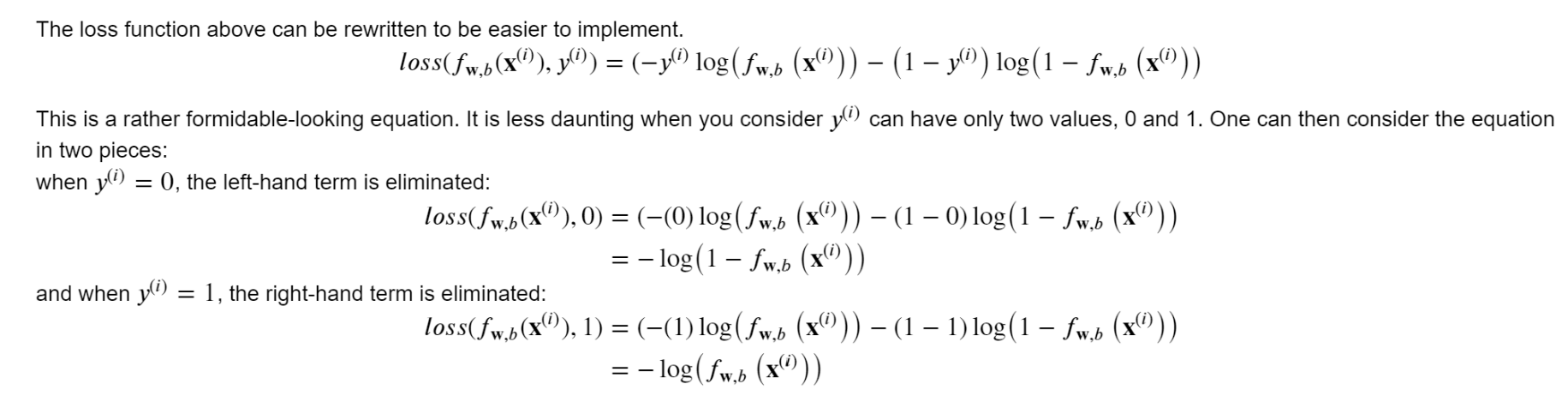


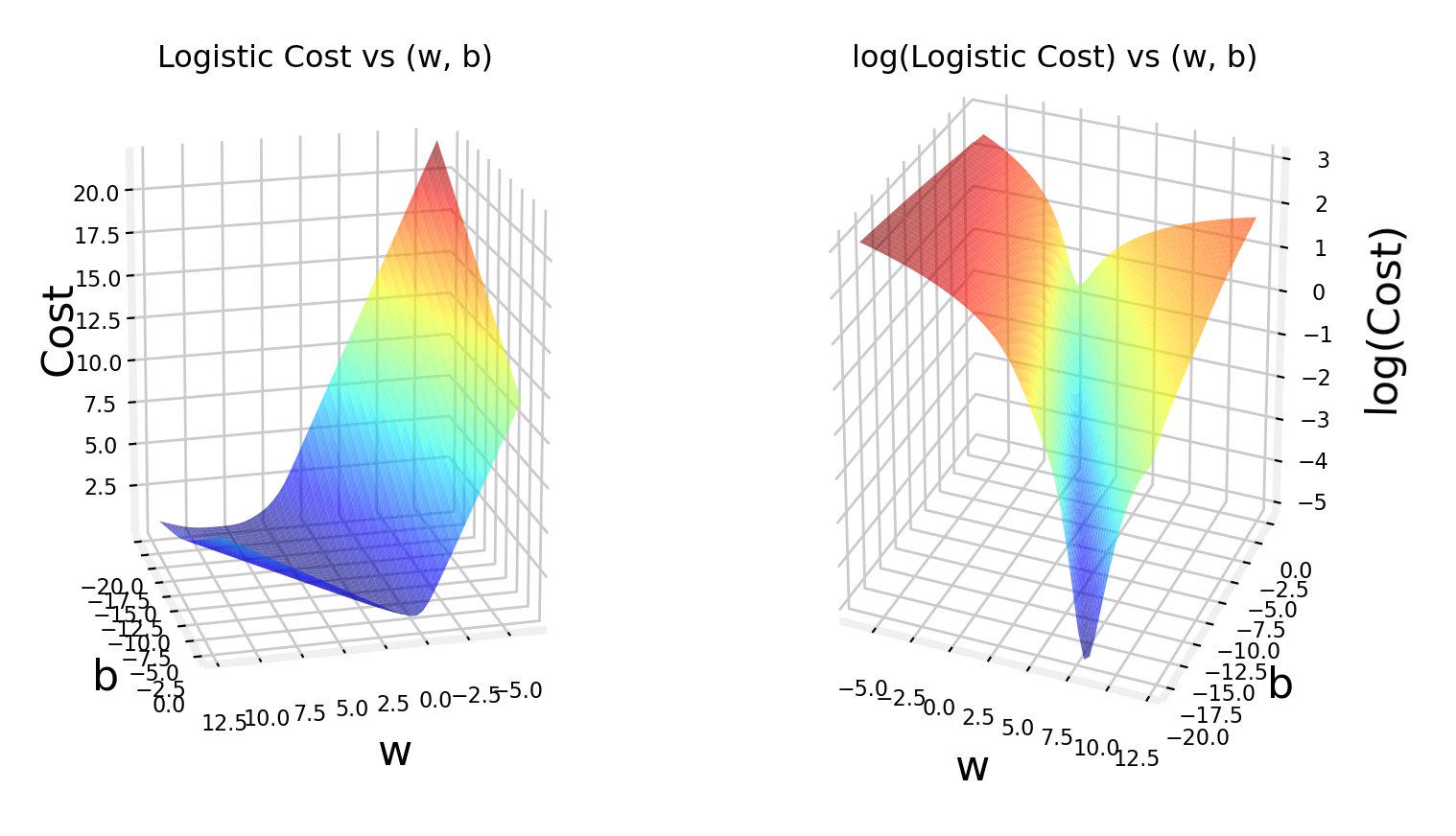
The goal is to have the same loss value when the prediction differs with the same value in comparison with the target value. When the target value is y=1 and f(x) is close to 1, the loss

- log(f(x)) is close to 0. When the target value is y=0 and f(x) is close to 0, loss is – log(1 - f(x)) is also close to 0. The loss value is symmetric with respect to the difference between the prediction and the target value.









Now we have a smooth curve well suited for gradient descent. Both the cost and the log of the cost are plotted to highlight that even when the cost value is low, the slope of the curve is still declining.

