Building a Simple Classifier

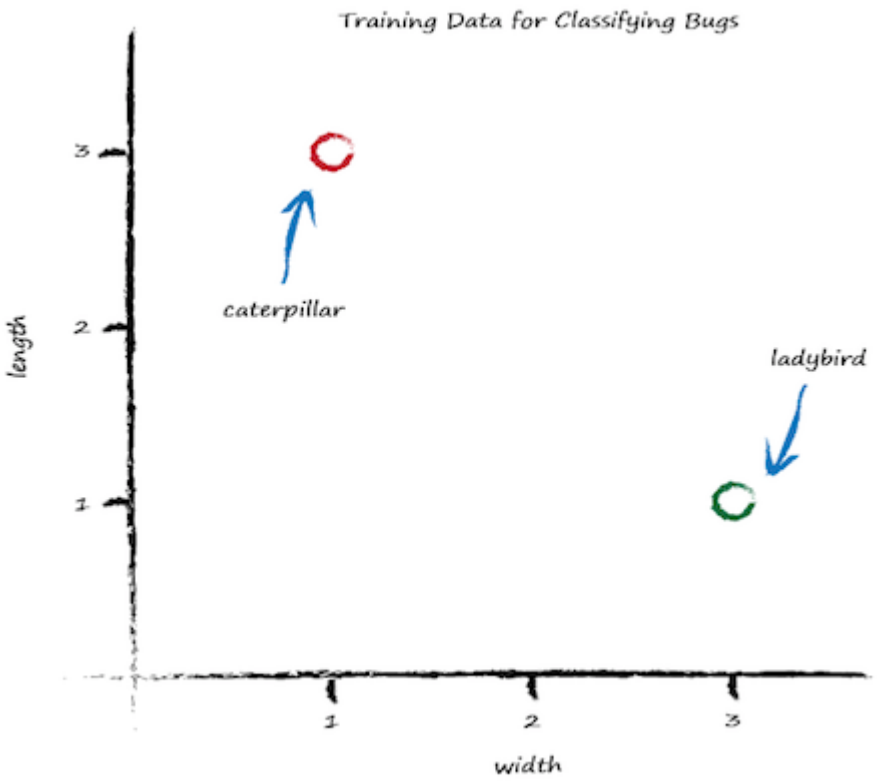
In this lesson, we will learn how to build a linear classifier with the help of some examples which we call "training data".

We want to train our linear classifier to correctly classify bugs as ladybirds or caterpillars. We saw above this is simply about refining the slope of the dividing line that separates the two groups of points on a plot of big width and height. How do we do this? Rather than develop some mathematical theory up front, let's try to feel our way forward by trying to do it. We'll understand the mathematics better that way. We do need some examples to learn from. The following table shows two examples, just to keep this exercise simple.

Example	Width	Length	Bug
1	3.0	1.0	ladybird
2	1.0	3.0	caterpillar

We have an example of a bug which has width 3.0 and length 1.0, which we know is a ladybird. We also have an example of a bug which is longer at 3.0 and thinner at 1.0, which is a caterpillar. This is a set of examples which we know to be the truth. It is these examples which will help refine the slope of the classifier function. Examples of truth used to teach a predictor, or a classifier is called the *training data*.

Let's plot these two training data examples. Visualising data is often very helpful to get a better understanding of it, a feel for it, which isn't easy to get just by looking at a list or table of numbers.

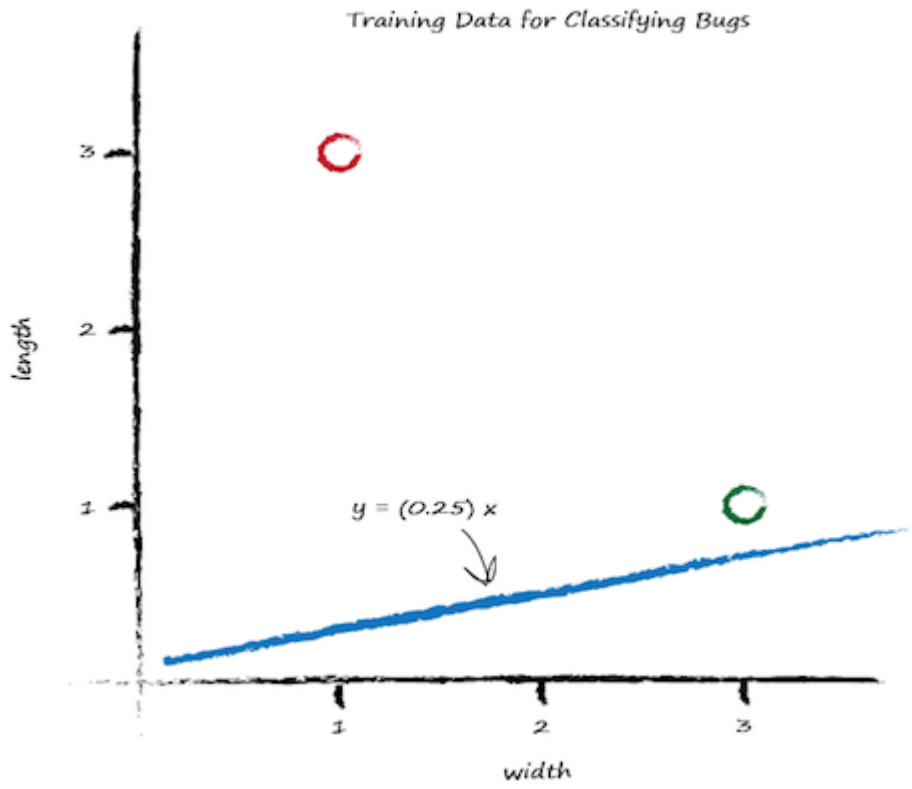


Let's start with a random dividing line, just to get started somewhere. Looking back at our kilometers to miles predictor, we had a linear function whose parameter we adjusted. We can do the same here because the dividing line is a straight line:

$$y = Ax$$

We have deliberately used the names *y* and *x* instead of length and width, because strictly speaking, the line is not a predictor here. It doesn't convert width to length like we previously converted kilometers to miles. Instead, it is a dividing line, a classifier. You may also notice that this $y = Ax$ is simpler than the fuller form for a straight line $y = Ax + B$. We've deliberately kept this garden bug scenario as simple as possible. Having a non-zero **B** simple means the line doesn't go through the origin of the graph, which doesn't add anything useful to our scenario.

We saw before that the parameter **A** controls the slope of the line. The larger **A** is, the larger the slope. Let's go for $A = 0.25$ to get started. The dividing line is $y = 0.25x$. Let's plot this line on the same plot of training data to see what it looks like:



Well, we can see that the line $y = 0.25x$ isn't a good classifier already without the need to do any calculations. The line doesn't divide the two types of bug. We can't say "if the bug is above the line then it is a caterpillar" because the ladybird is above the line too. So intuitively we need to move the line up a bit. We'll resist the temptation to do this by looking at the plot and drawing a suitable line. We want to see if we can find a repeatable recipe to do this, a series of computer instructions, which computer scientists call an *algorithm*.