The math behind AdaBoost

Let's assign to each data point an initial weight of one.

- Now, let's fit our **first** Learner. Before we wanted to minimize the number of errors. Now we want to minimize the sum of weights of the incorrectly classified points which as of now is the same. If we add the weights of the correctly classified points, we get a 7 and 3 for the incorrectly classified points.

- So, let's weigh the incorrectly classified points a bit more. How much? Well, let's make it enough to have this model be a 50-50 model. So, if we take these three points and weight them by a factor of 7/3 or 2.33, our model now has 7 as the sum of the weights of the correctly classified points and also 7 as the sum of the weights of the incorrectly classified points. So, this model now is lame, and we need a second one.

- Our **second** model will be this one that fits the newly weighted data best. For this one, we can see that if we add the weights of the correctly classified points, we get 11, and for the incorrectly classified ones, we get 3 again. So, let's weigh the incorrectly classified points. By how much?

- Well, enough to make this model a 50-50 model again. Notice that if we weigh these three errors each by a factor of 11/3 or 3.66, the sum of the weights of the incorrectly classified points is 11. So, our model is just a 50-50 model again. That's lame. Let's fit a third model in this newly weighted data.

- Our **third** model has the added weight of the correctly classified points? Now it's 19 and still 3 for the incorrect ones.

Quiz Question

Why does AdaBoost apply weighting to misclassified points?

a. In order to have these points removed from the dataset since they are misclassified
b. In subsequent models, classifiers can focus on the misclassified samples more.
c. All weights between the prediction classes should equal zero.