**Changing optimization parameters**

It's time to get your hands dirty with optimization. You'll now try optimizing a model at a very low learning rate, a very high learning rate, and a "just right" learning rate. You'll want to look at the results after running this exercise, remembering that a low value for the loss function is good.

For these exercises, we've pre-loaded the predictors and target values from your previous classification models (predicting who would survive on the Titanic). You'll want the optimization to start from scratch every time you change the learning rate, to give a fair comparison of how each learning rate did in your results. So we have created a function get\_new\_model() that creates an unoptimized model to optimize.

**Instructions**

* Import SGD from keras.optimizers.
* Create a list of learning rates to try optimizing with called lr\_to\_test. The learning rates in it should be .000001, 0.01, and 1.
* Using a for loop to iterate over lr\_to\_test:
  + Use the get\_new\_model() function to build a new, unoptimized model.
  + Create an optimizer called my\_optimizer using the SGD() constructor with keyword argument lr=lr.
  + Compile your model. Set the optimizer parameter to be the SGD object you created above, and because this is a classification problem, use 'categorical\_crossentropy' for the loss parameter.
  + Fit your model using the predictors and target.