**Scaling up to multiple data points**

You've seen how different weights will have different accuracies on a single prediction. But usually, you'll want to measure model accuracy on many points. You'll now write code to compare model accuracies for two different sets of weights, which have been stored as weights\_0 and weights\_1.

input\_data is a list of arrays. Each item in that list contains the data to make a single prediction. target\_actuals is a list of numbers. Each item in that list is the actual value we are trying to predict.

In this exercise, you'll use the mean\_squared\_error() function from sklearn.metrics. It takes the true values and the predicted values as arguments.

You'll also use the preloaded predict\_with\_network() function, which takes an array of data as the first argument, and weights as the second argument.

**Instructions**

* Import mean\_squared\_error from sklearn.metrics.
* Using a for loop to iterate over each row of input\_data:
  + Make predictions for each row with weights\_0 using the predict\_with\_network() function and append it to model\_output\_0.
  + Do the same for weights\_1, appending the predictions to model\_output\_1.
* Calculate the mean squared error of model\_output\_0 and then model\_output\_1 using the mean\_squared\_error() function. The first argument should be the actual values (target\_actuals), and the second argument should be the predicted values (model\_output\_0 or model\_output\_1).