Unlike categorical data, neural network models can interpret and evaluate all forms of numeric data. In other words, if our input data has no categorical data types, it can be provided to a neural network model in its raw form. Even though a neural network *can* train on raw numerical data, it does not mean that it *should* train on raw data. There are many reasons why a raw numeric variable is insufficient for use when training a neural network model, such as:

* Raw data often has outliers or extreme values that can artificially inflate a variable's importance.
* Numerical data can be measured using different units across a dataset—such as time versus temperature, or length versus volume.
* The distribution of a variable can be skewed, leading to misinterpretation of the central tendency.

If we use raw numeric data to train a neural network model, there is a chance that the neural network model will perform adequately. However, there is a far greater probability that the neural network model will interpret the raw numerical data inappropriately, which will yield an inadequate model. Thankfully, we can minimize this risk by standardizing (also commonly referred to as normalization) the numerical data prior to training.

**REWIND**

Scikit-learn's StandardScaler module standardizes numerical data such that a variable is rescaled to a mean of 0 and standard deviation of 1.

If we use the StandardScaler module to standardize our numerical variables, we reduce the overall likelihood that outliers, variables of different units, or skewed distributions will have a negative impact on a model's performance.