Numerical data: How a model ingests data using feature vectors

Until now, we've given you the impression that a model acts directly on the rows of a dataset; however, models actually ingest data somewhat differently.

For example, suppose a dataset provides five columns, but only two of those columns (`b` and `d`) are features in the model. When processing the example in row 3, does the model simply grab the contents of the highlighted two cells (3b and 3d) as follows?

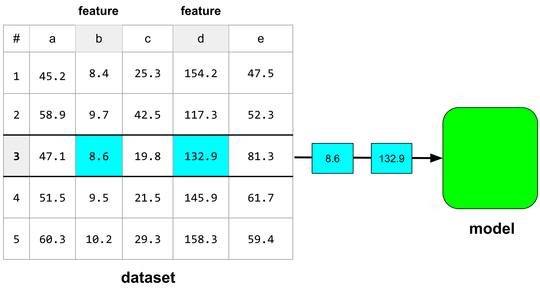

Figure 1. *Not exactly how a model gets its examples.*

In fact, the model actually ingests an array of floating-point values called a **feature vector**. You can think of a feature vector as the floating-point values comprising one example.

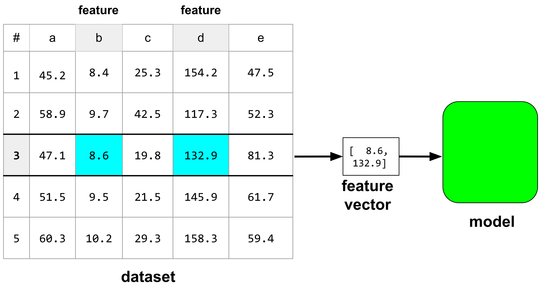

Figure 2. *Closer to the truth, but not realistic.*

However, feature vectors seldom use the dataset's *raw values*. Instead, you must typically process the dataset's values into representations that your model can better learn from. So, a more realistic feature vector might look something like this:

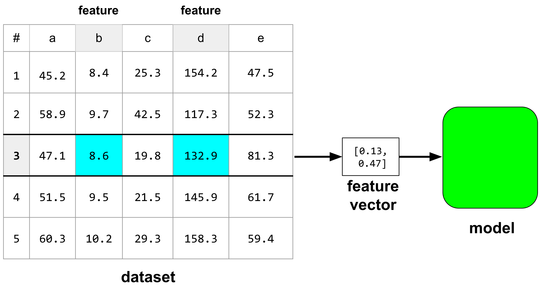

Figure 3. *A more realistic feature vector.*

Wouldn't a model produce better predictions by training from the *actual* values in the dataset than from *altered* values? Surprisingly, the answer is no.

You must determine the best way to represent raw dataset values as trainable values in the feature vector. This process is called **feature engineering**, and it is a vital part of machine learning. The most common feature engineering techniques are:

- **Normalization**: Converting numerical values into a standard range.
- **Binning** (also referred to as **bucketing**): Converting numerical values into buckets of ranges.

This unit covers normalizing and binning. The next unit, Working with categorical data, covers other forms of **preprocessing**, such as converting non-numerical data, like strings, to floating point values.

Every value in a feature vector must be a floating-point value. However, many features are naturally strings or other non-numerical values. Consequently, a large part of feature engineering is representing non-numerical values as numerical values. You'll see a lot of this in later modules.

Key terms:

- Binning
- Bucketing
- Feature engineering
- Feature vector
- Normalization
- Preprocessing

Help Center

Previous
Introduction (3 min)

Next
First steps (5 min)

Was this helpful?
Send feedback