REPRESENTATION

Data comes in in various forms — need to extract feature vectors. This is feature engineering and takes ~70% of your time.

Examples:

— String → One-hot encoding: map each unique string to an integer (a "coefficient"). This is sparse categorical data. (BUT REPRESENT w/ BOOL VEC!)

(multi-hot: can have multiple present)

vocabulary {"a", "b", "c"} → 0, 1, 2; other (out-of-vocab bucket) → 3

binary vector for cat data: ⟨{1,0}, {1,0}, {1,0}, {1,0}⟩ → 0, 1, 2, 3

(too inefficient? use sparse rep)

What are good features?

- Clear and obvious meaning
- Mostly non-zero in the dataset
- No "magic values" (e.g. -1 → bool + int)
- Feature values shouldn't change over time
- Should not have extreme outliers (e.g. upstream instability)
  - ↳ Cap or transform or scale (e.g. log(_)) (cap = 1000)

Binning trick: map continuous → categorical

e.g. latitude not linearly related to home price, but mapping to city, nbhd, etc. might

✡ KNOW YOUR DATA!

- visualize (e.g. histograms, etc.)
- debug (dupes? missing vals? outliers? test/training/val sets?)
- monitor over time

Scrubbing incoming (possibly bad) data by writing a program.

- Remove bad examples
  - Omitted values
  - Duplicates
  - Bad labels or values (tricky to detect)
- Detect aggregate problems w/ stats + viz
  - Histograms
  - Max/min, stdev, etc.
  - List of most common discrete vals

Rules

1. Know how you expect your data to look
2. Verify that the data meets these expectations
3. Check training data agrees w/ other sources (e.g. dashboards)

★ GOOD ML RELIES ON GOOD DATA!