DataSci 400

lesson 4: dealing with numeric data

Seth Mottaghinejad

today's agenda

- types of numeric columns
- two types of functions
  - **summarization** function
  - **transformation** functions
- common transformations for **feature engineering**:
  - binning, trimming, normalizing
- special transformations
  - RFM (recency, frequency, monetary)

types of numeric columns

- **floats**: mathematicians call it a **continuous variable**
  - in `pandas` we use `float64` (or `float32` etc.)
- **integers**: mathematicians call it a **discrete variable**
  - in `pandas` we use `int64` (or `int32` etc.)
- **datetimes**: this is more like a construct, behind the scenes datetimes are like integer columns with the counter starting at some date in the past like `1960-01-01 00:00:00`
  - in `pandas` we use `datetime64`
  - the **difference** between two `datetime64` is a `timedelta[n]` type

summary vs transformation functions

- a **summary** function takes in **an array of numbers** and returns **a single number** (or several sometimes)
    - **average:** `mean([4, 5, 6])` → `5`
- a **transformation** function takes in **an array of numbers** and returns **an array of numbers of the same size** where we assume a 1-1 correspondence between elements
    - **log:** `log([4, 5, 6])` → `[1.386, 1.610, 1.792]`
    - **round:** `log([1.386, 1.610, 1.792])` → `[1.4, 1.7, 1.8]`
    - **lag:** `log([4, 5, 6])` → `[NA, 4, 5]`

types of numeric columns

- feature engineering is all about **applying transformation** functions to the data to create **new features from existing features**
- for numeric features three common transformations are
    - **binning:** take a numeric feature and return a categorical feature based on what **pre-defined interval** the number is in
    - **trimming:** remove outliers from the data by replacing values that exceed a threshold with the threshold or NA
    - **normalization:** rescale columns in the data so they all have the same **scale** (range of values)

two common ways to normalize

- **Z-normalization** makes most values fall between -2 and 2

$$x \to \frac{x - \mathrm{mean}(x)}{\mathrm{std}(x)}$$

- **min-max normalization** forces all the values between 0 and 1

$$x \to \frac{x - \min(x)}{\max(x) - \min(x)}$$

- **some (but not all)** ML algorithms require normalized data

special transformations: RFM example

- because datasets can be very different depending on industry, there are some data transformations that are **industry-specific**
- in **retail** for example, where we have **times series** data of customer purchases we can extract RFM features
    - **R is for recency:** how long ago was the most recent purchase?
    - **F is for frequency:** how many purchases where made in the last week (or month, or whatever window you choose)?
    - **M is for monetary:** how much in total was spent in the last week (or month, or whatever window you choose)?

date	customer	purchased
2020-01-01	XYZ	$50
2020-01-03	XYZ	$23
2020-02-11	XYZ	$35
2020-01-02	ABC	$50
2020-01-02	ABC	$23
2020-01-29	ABC	$35
⋮	⋮	⋮

date	customer	purchased	R	F	M
2020-01-01	XYZ	$50	NA	NA	NA
2020-01-03	XYZ	$23	2 days	2	$73
2020-01-11	XYZ	$35	8 days	1	$35
2020-01-02	ABC	$65	NA	NA	NA
2020-01-03	ABC	$25	1 day	2	$90
2020-01-29	ABC	$35	26 days	1	$35
⋮	⋮	⋮	⋮	⋮	⋮

the end