Data preprocessing: main tasks

1. Data cleaning: handling errors and missing values
2. Feature extraction: creating new features by combining and transforming existing ones
   - a **crucial step!** ⇒ what patterns you can find
   - application specific ⇒ understanding the domain
3. Data reduction
   - sampling
   - feature selection
   - dimension reduction by transformations

1. Data cleaning

Goal: detect & eliminate errors, missing values, duplicates, noise, sometimes outliers

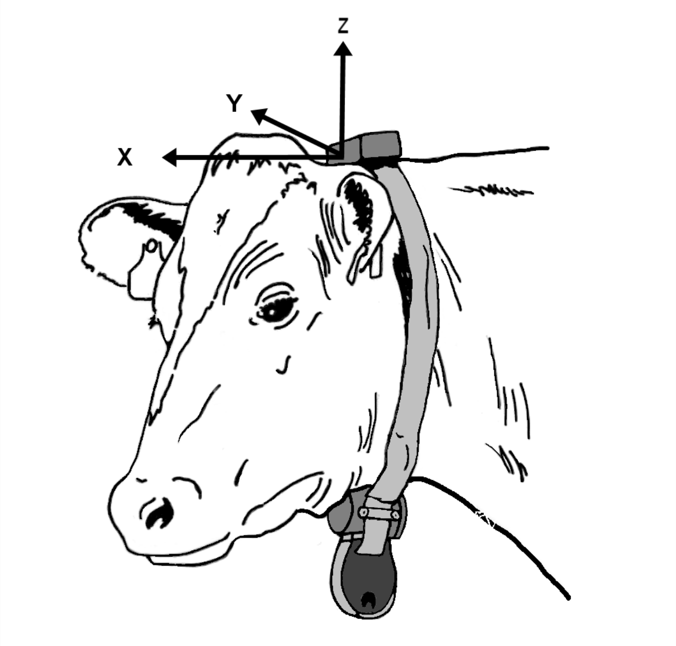
- but outliers may also reveal some interesting event!

Sources:

- automatic measuring devices may stop reading or transmit duplicates (e.g., HW failures or battery exhaustion)
- users may not want to specify (correct) information for privacy reasons
- manually entered data contains very often errors!
- automatically produced text (from scanned documents or speech) prone to errors

Real world example

Task: predict cows' activities (walking, standing, lying, ...)
Data: sequences of accelerometer measurements for time intervals when an animal performs an activity (class).

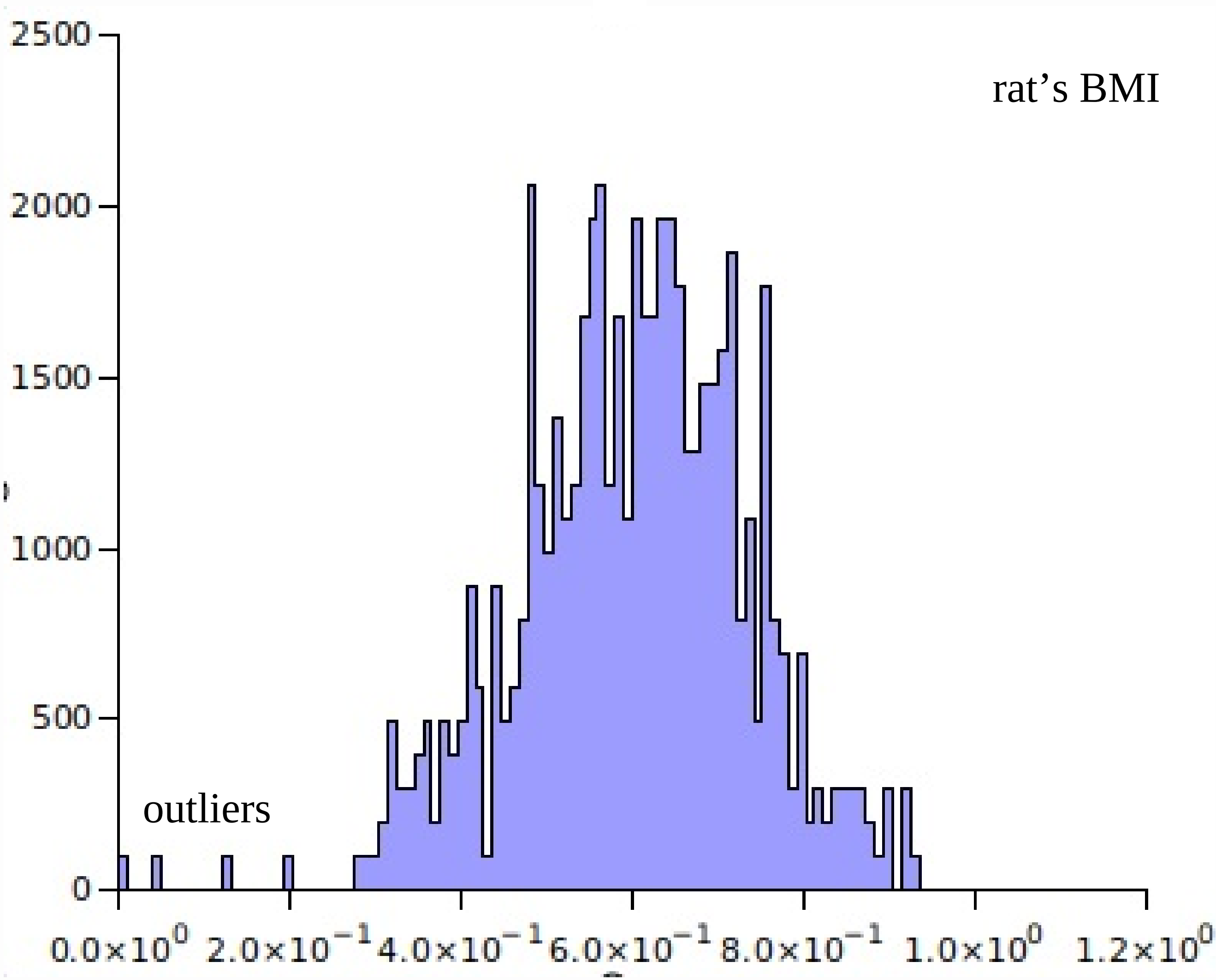


- faulty devices
- lack of calibration
- transmission breaks
- noise
- individual fluctuations
- **errors in human annotation**

Errors and inconsistencies: strategies

- check inconsistencies between different data sources
  - e.g., name spelling
- use domain knowledge
  - known ranges of values (age cannot be 800 yrs)
  - known relationships (if country='USA', city$\neq$'Sanghai')
- check outliers and extreme values (error candidates)
  - not errors, if they have a reasonable explanation
- data smoothing reduces noise and random fluctuations
  - e.g., scaling, discretization, dimension reduction
- use robust methods in the modelling phase

Example: outliers may reveal errors



Missing values: strategies

If possible, **replace with correct values**. Otherwise,

- if a feature has many missing values, **prune the feature**
- if a record has many missing value, **prune the record**
- **impute** missing values
  - mean or median of the feature (among all or similar records/nearest neighbours)
  - predict the missing value using other features (e.g., random forests imputation)
  - **Warning!** Imputation may have a strong effect the results!
- use a **modelling technique that allows missing values** (just replace with special values like “NA”)

2. Feature extraction methods

- scaling and normalization: numerical → numerical
- discretization: numerical → categorical
- binarization: categorical → binary (0/1)
- creating similarity graphs: any type → graph
- transformations for dimension reduction: create new less redundant features and keep the best ones
  - both feature extraction + data reduction

Scaling and normalization

Problem: Features with large magnitudes often dominate
$\Rightarrow$ transform to the same scale or standardize distributions

- **min-max scaling**:

$$y = \frac{x - \min(x)}{\max(x) - \min(x)} \quad \text{(new range } [0, 1])$$

- **mean normalization**:

$$y = \frac{x - mean(x)}{\max(x) - \min(x)} \quad \text{(new range } [-1, 1], mean(y) = 0)$$

- **Beware!** outliers can affect a lot!

Standardization or z-score normalization

If the distribution is normal:

$$z = \frac{x - mean(x)}{stdev(x)}$$

$$mean(z) = 0$$

$$stdev(z) = 1$$

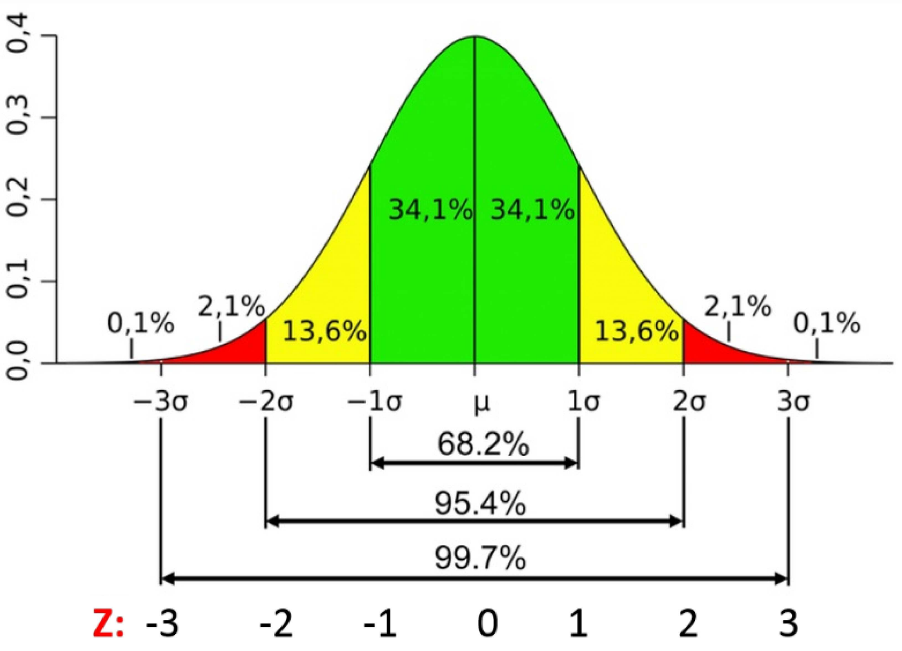


image source:
`https://sphweb.bumc.bu.edu/otlt/MPH-Modules/PH717-QuantCore/PH717-Module6-RandomError/PH717-Module6-RandomError5.html`

Discretization: numerical $\rightarrow$ categorical

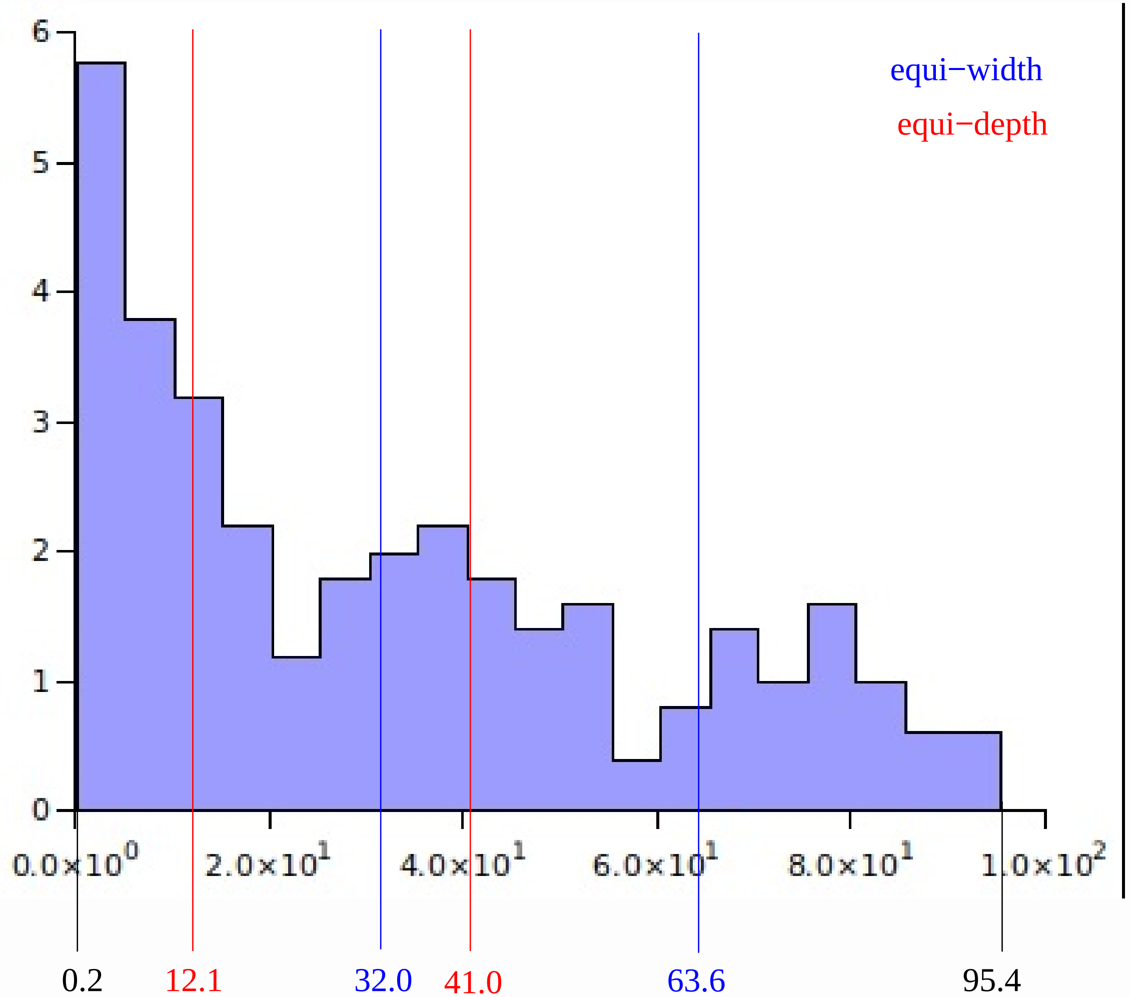
- divide the numerical range into intervals (bins) + give labels to bins
- temperature could be discretized as $T < 0°$C cold, $0 \leq T < 15°$C cool, $T \geq 15°$C warm
- **binarization**: a special case when the new variable is binary (true/false or 1/0)
- e.g., frost=1, if $T < 0$ and frost=0 otherwise
- Note! Also categorical variables can be binarized
  - eye-colour={blue, brown, green, grey} $\Rightarrow$ blue-eyed=1, if eye-colour=blue, and 0 otherwise

Some discretization methods

- Equi-width discretization
  - equally wide bins
  - good if uniform distribution
- Equi-depth (equal frequency)
  - each bin has an equal number of records
- Many supervised methods if class labels available
- Visual/manual: often best results, but can be worksome

Example: internet users/100 people in countries

Equi-width or equi-depth wouldn't present natural groups



Discretization: benefits and limitations

+ good way to handle mixed data
+ removes noise and individual variation

$\Rightarrow$ it is often worth of analyzing a discretized version of purely numerical data

  + less noise, clearer patterns
  + more efficient algorithms
  + discrete patterns may help to choose the right modelling method also for numerical data

- loses some information
- optimal discretization difficult! (optimal discretization of one variable may depend on other variables)

Useful type transport: any type → similarity graph

- idea: present **pairwise similarities** among closest neighbours by a neighbourhood/similarity graph
- suitable for **any data type** if the distance/similarity function can be defined
- for any application based on the notion of similarity/distances
  - e.g., clustering, recommendations based on similarity
- enables use of numerous network algorithms
- Beware: can be time consuming for large data! (brute force $\mathcal{O}(n^2)$, n=number of objects)

Constructing nearest neighbour graph (idea)

Given objects $O_1, \ldots, O_n$, a distance measure d and a user-defined parameter ϵ or K.

1. create a node for each O_i
2. create an edge between a pair near/similar objects:
   i) if $d(O_i, O_j) \leq \epsilon \Rightarrow$ undirected edge $O_i - O_j$ **or**
   ii) if O_j is among K nearest neighbours of $O_i \Rightarrow$ directed edge $O_i \rightarrow O_j$ (direction can be ignored)
3. give weights to edges reflecting similarity, e.g.,

$$w_{ij} = e^{-d(O_i,O_j)^2/t^2} \quad \text{(heat kernel, } t \text{ user-defined)}$$

3. Data reduction: approaches

1. sampling (select a subset of records)
2. feature selection (select a subset of features)
   - application specific!
   - **filtering** methods: prune features before modelling
   - **wrapper** methods: use modelling (e.g., clustering) to evaluate goodness of feature sets
   - **hybrid** methods: candidates by filtering + evaluation by modelling
3. dimension reduction
   - by axis rotation (PCA, SVD)
   - with type transformation

Main messages

- careful with data types
- careful with preprocessing (data often dirty!)
- feature extraction has a strong effect

Reading for lecture 1:

Book Ch 1 and Ch 2 except 2.4.3–2.4.4