Exploratory Data Analysis (P2)

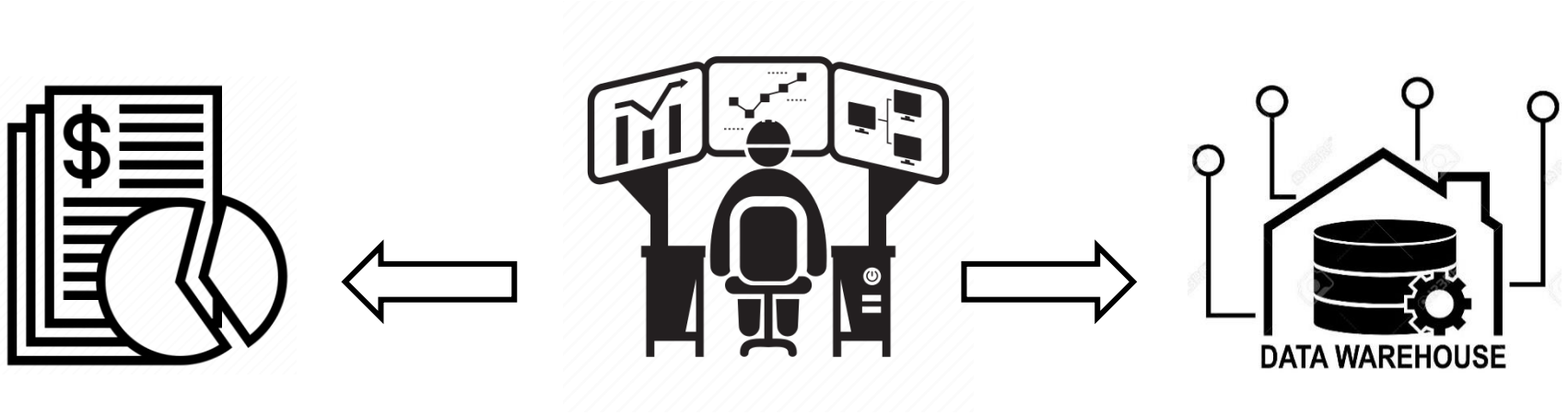
Nguyen Ngoc Thao
nnthao@fit.hcmus.edu.vn

Content outline

- Data quality
- Major tasks in Data preprocessing

Data quality

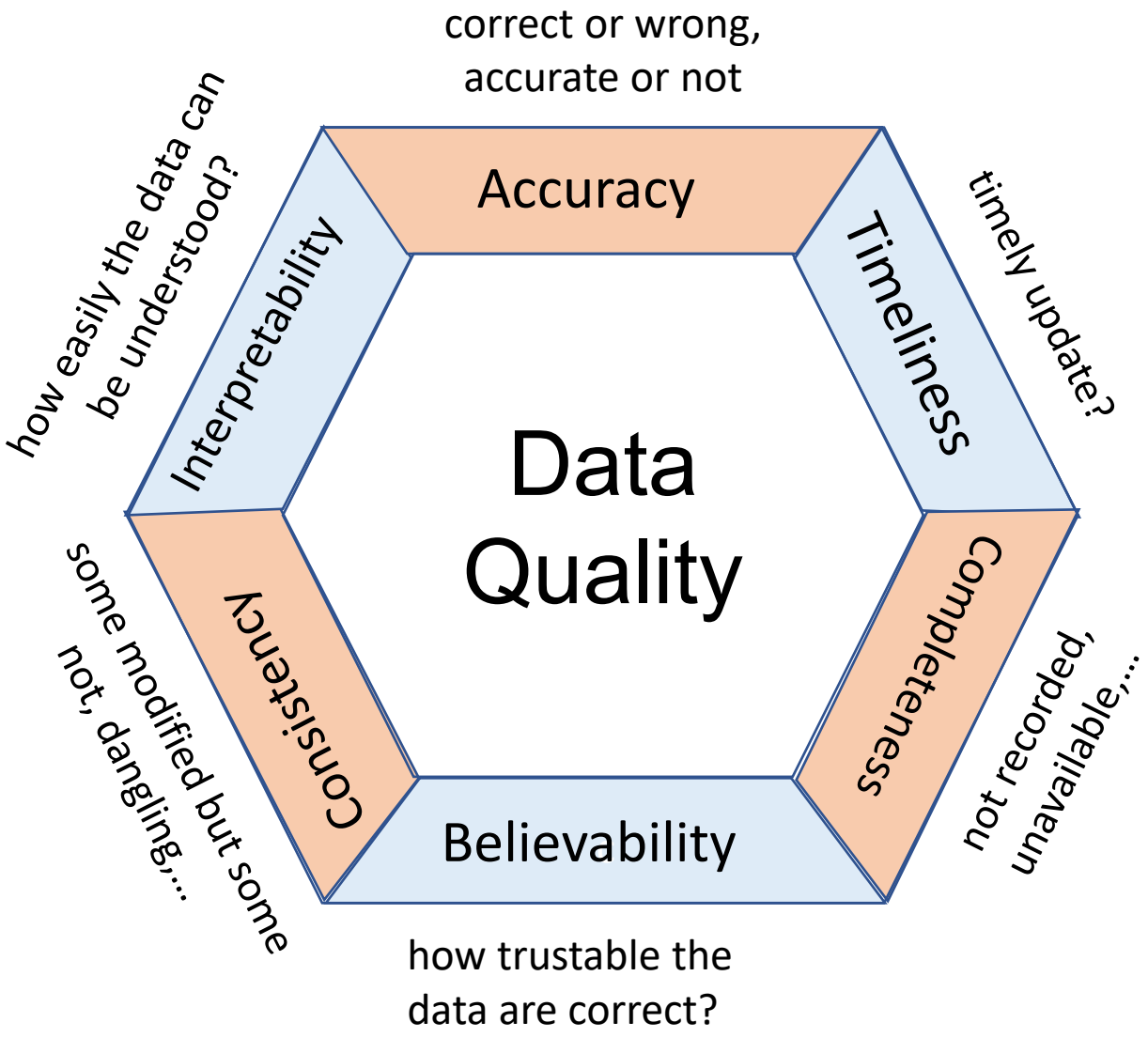
An example of data analytics



- A branch manager analyzes the sales data by inspecting the company's data warehouse to include the necessary attributes.
- HOWEVER, the data being considered has many problems
  - Information needed for the analysis has not been recorded.
  - Many errors and unusual values for some transactions have been reported.

Inaccurate, incomplete, and inconsistent data are commonplace properties of large real-world databases and data warehouses.

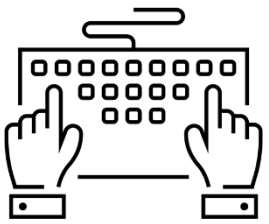
Measures of data quality



Data accuracy

- Inaccurate data means having incorrect attribute values.

Incorrect values submitted for mandatory fields

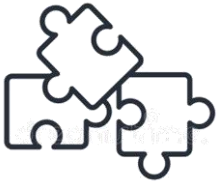


- E.g., negative weight, inappropriate range of ages, etc.
- *Disguised missing data*: many users have the same birthday, e.g., Jan 01

Faulty data collection instruments

Data transmission errors due to technology limitations

- E.g., limited buffer size for coordinating synchronized data transfer



Incorrect data may also result from inconsistencies

Data completeness

- The attributes of interest may not always be available or contain only aggregated data.
  - E.g., study the shopping habits in festive seasons while only the annual sales are available
- Many causes are leading to missing data.
  - Equipment malfunction
  - Some records are deleted due to inconsistency with other records.
  - Data is not entered due to misunderstanding.
  - Certain data may not be considered vital at the time of entry.
  - The recording of data history may have been overlooked.

Data consistency

- Inconsistencies in naming conventions or data codes
  - E.g., USA vs. US, alternative name (Bill Clinton vs. William Clinton), author name in reference: Li Fei-Fei vs. Fei-Fei, L.
- Incompatible formats for input fields
  - E.g., datetime format (dd/mm/yy vs. mm/dd/yy), rating scale ([1..5] vs. [1..10]), decimal and thousand separators
- Duplicate tuples also require data cleaning.

Data timeliness

- Suppose you are overseeing the data of monthly sales
- For a while after each month, the data stored is incomplete.
  - Several sales representatives fail to submit their sales records on time at the end of the month.
  - There are also some corrections and adjustments flowing in after the month's end.
- However, once all the data is received, it is correct.
- The month-end data are not updated in a timely fashion, harming the data quality.

Believability and Interpretability

- Suppose that a database, at one point, had several errors, all of which have since been corrected.

Believability: how trustily is the data correct?

- E.g., the past errors had caused many problems for sales department users → they no longer trust the data

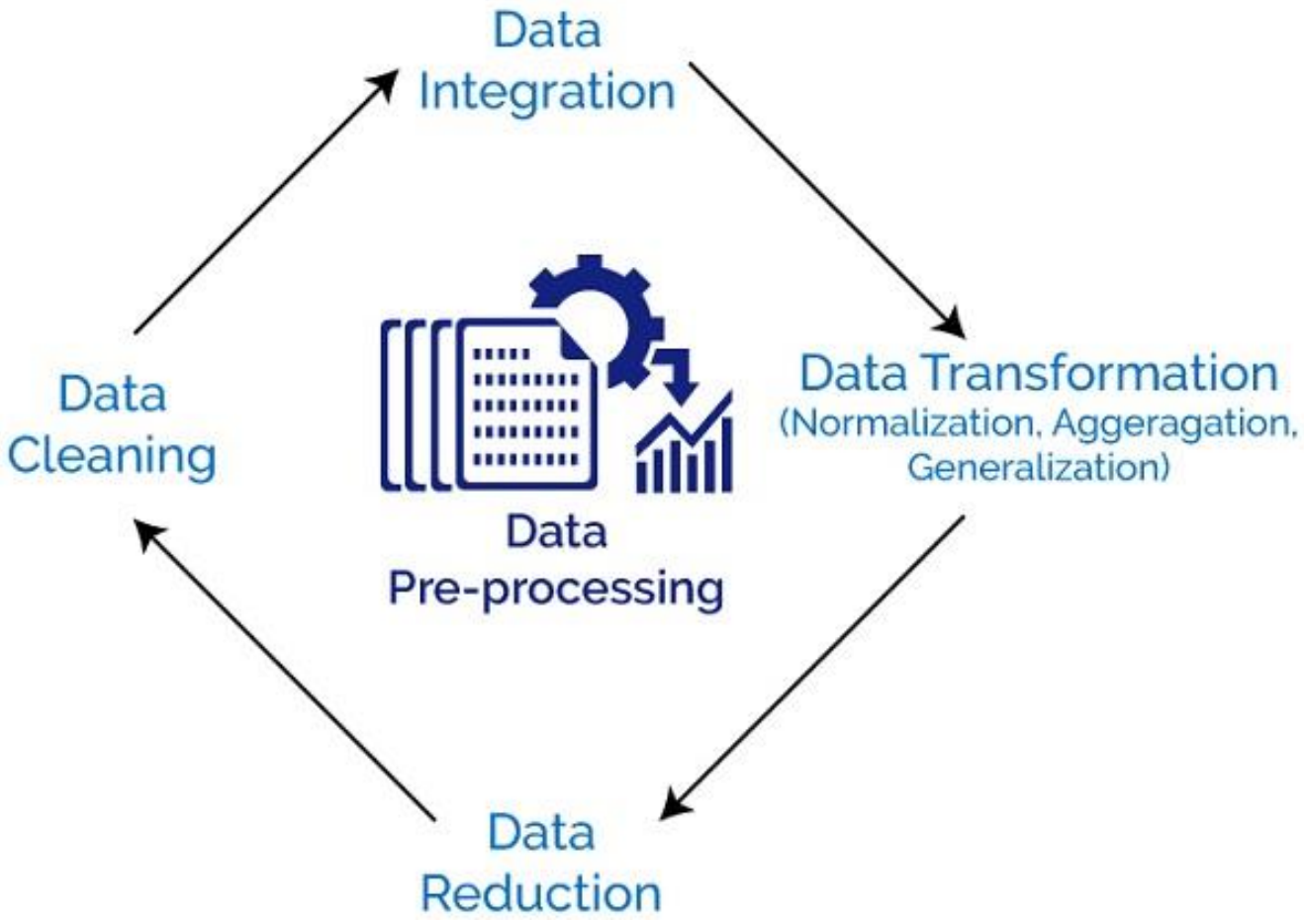
Interpretability: how easily is the data interpreted?

- E.g., the data use many accounting codes → the sales department does not know how to figure out

Data quality is subjective

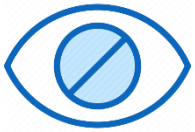
- Data quality depends on the intended use of the data.
- Two users may assess the quality of a database differently.
- Consider a database in which some customer addresses are outdated or incorrect, yet overall, 80% of them are accurate.
  - A marketing analyst considers the database to be accurate enough for target marketing purposes.
  - However, a sales manager may consider the data inaccurate.

Major tasks in Data preprocessing



Data cleaning

How to handle missing data?



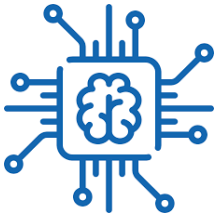
Ignore the tuple

- Usually done when class label is missing
- Not effective when the percentage of missing values per attribute varies considerably

Fill in the missing value manually

- Tedious and infeasible

Fill in it automatically with



- A global constant, e.g., “unknown” or a new class
- The attribute mean (for all samples of the same class)
- The most probable value: Bayesian approach or decision tree

How to handle noisy data?

Binning and smoothing

- First sort data and partition into (equal-frequency) bins
- Then smooth each bin by its mean, median, or boundary, etc.

Regression

- Smooth by fitting the data into regression functions

Clustering

- Detect and remove outliers

Hybrid

- Suspicious values are detected by computers and checked by human

Binning and smoothing: An example

- Consider the following sorted data points

4	8	15	21	21	24	25	28	34

- Partition into equal-frequency bins

Bin 1:	4	8	15
Bin 2:	21	21	24
Bin 3:	25	28	34

- Smooth the bins

Bin 1:	9	9	9
Bin 2:	22	22	22
Bin 3:	29	29	29

By means

Bin 1:	8	8	8
Bin 2:	21	21	21
Bin 3:	28	28	28

By medians

Bin 1:	4	4	15
Bin 2:	21	21	24
Bin 3:	25	25	34

By bin boundaries

Quiz 01: Binning and Smoothing

1. Consider the following 1D data series, which includes 15 data points sorted in ascending order.

   21, 25, 27, 29, 32, 36, 36, 48, 67, 80, 84, 85, 89, 92, 97

   - Apply equal-frequency and equal-width binnings to the given data series to obtain three bins of data points.
   - Apply smoothing on the bins obtained from equal-width binning.

2. How to perform binning and smoothing, without coding from scratch, in **Python**?

   For each available function, show the result for the above data.

Data integration

Entity identification problem

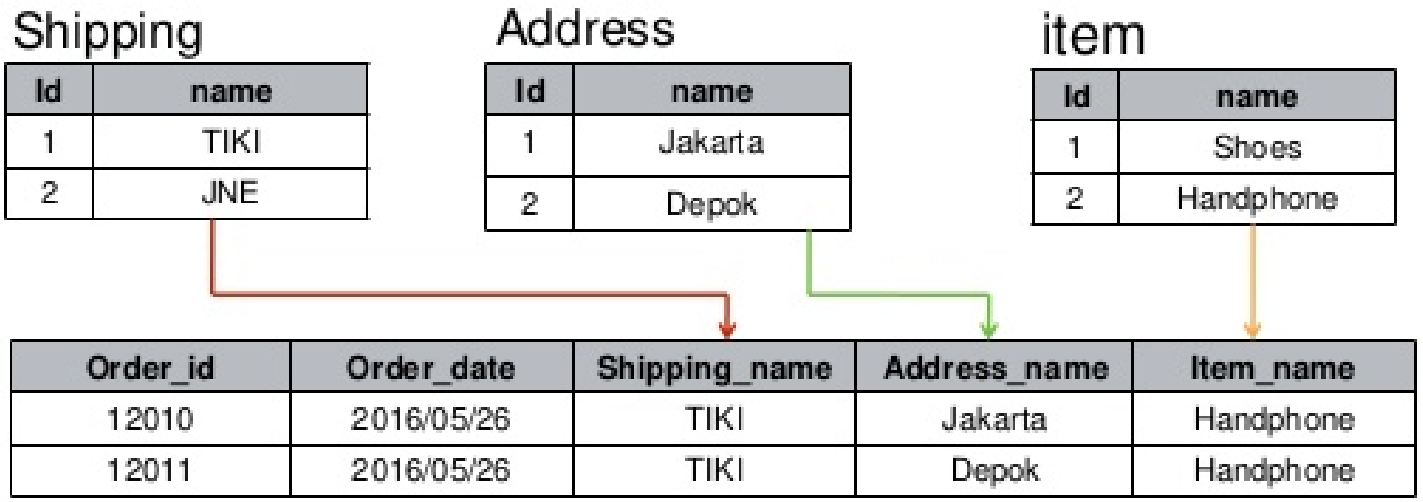
- Entity identification problem arises during integration.
- Identify real world entities from multiple data sources
  - Differences in representation, scaling, or encoding
  - E.g., metric units in British system and other systems, currencies, grading scheme between schools, time format, etc.
- Matching attributes from one database to another following the ontological structure.
  - An attribute in one system is recorded at, say, a lower abstraction level than the “same” attribute in another.
  - E.g., “Total sales” may refer to one branch or to all stores in a region.
- Careful integration helps improve mining speed and quality.

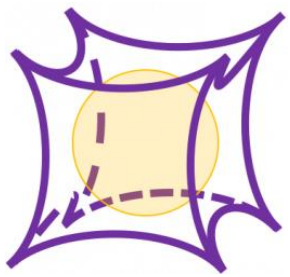
How to handle redundancy?

- Redundant data often occur when integrating databases.
- Object identification: The same attribute or object may have different names in various databases.
  - E.g., the occupation information may be stored in column “job” of the first database and column “career” of the second database.
- Derivable data: An attribute is derived from other attributes.
  - E.g., the annual revenue is the sum of monthly revenues.
- Correlation analysis helps detect attributes having the similar trends $\rightarrow$ keep one attribute and discard the others.

Tuple duplication

- Duplication should also be detected at the tuple level.
  - E.g., two or more identical tuples for a given unique data entry case.
- The use of denormalized tables (often done to improve performance by avoiding joins operation) is also a reason.
  - E.g., a purchase order database contains a purchaser's name and address instead of a key to this information in a purchaser database

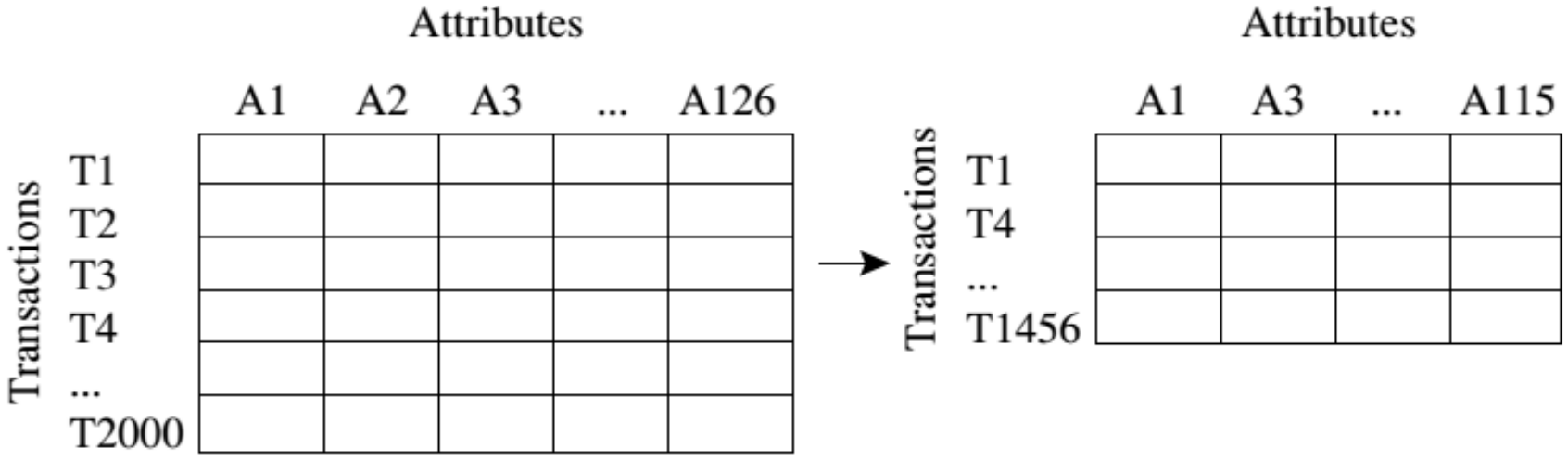




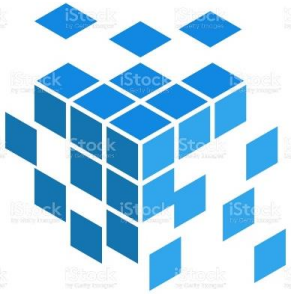
Data reduction

Why data reduction?

- A data collection stores terabytes of data → complex data analysis on the entire dataset may take a long time.
- Data reduction reduces the dataset in volume to achieve (almost) the same analytical results.



Why data reduction?



Avoid the curse of dimensionality

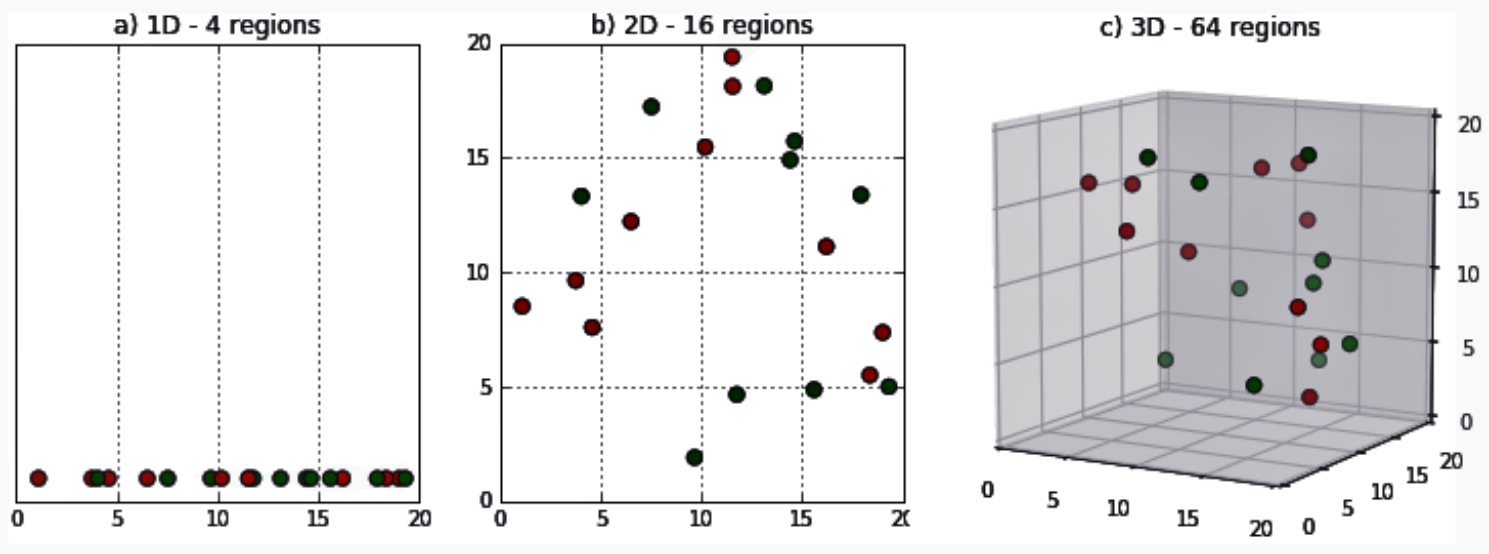
Eliminate irrelevant features and reduce noise

Reduce time and space required in data mining

Allow easier visualization

Curse of dimensionality

- As the dimensionality increases, the volume of the space increases so fast that the available data become sparse.



- The data must grow exponentially with the dimensionality to obtain a reliable result.

Data reduction techniques

Dimensionality reduction

- Data encoding schemes are applied for a compressed representation of the original data.

Numerosity reduction

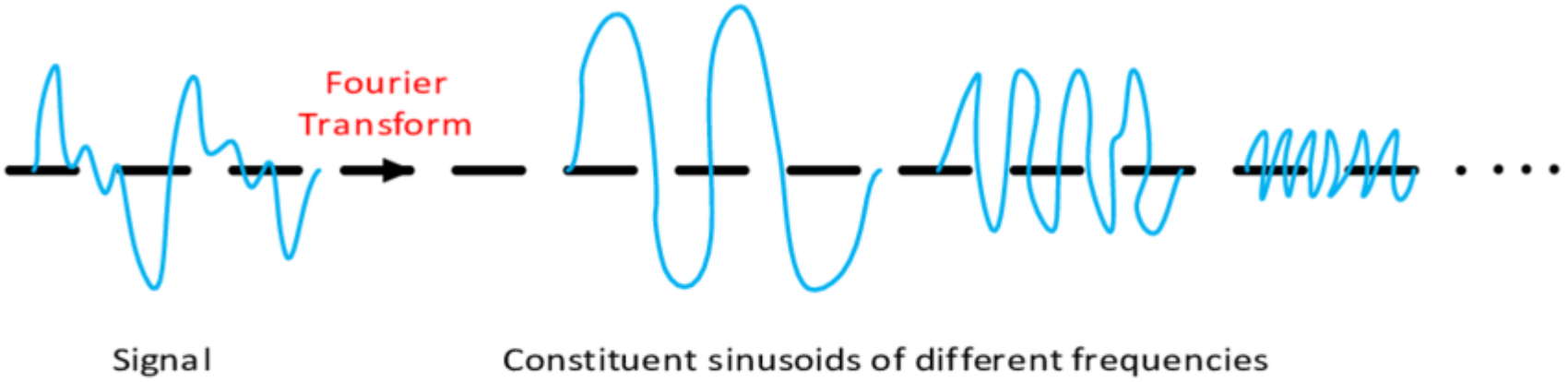
- Data volume is reduced by choosing alternative, smaller forms of data representation

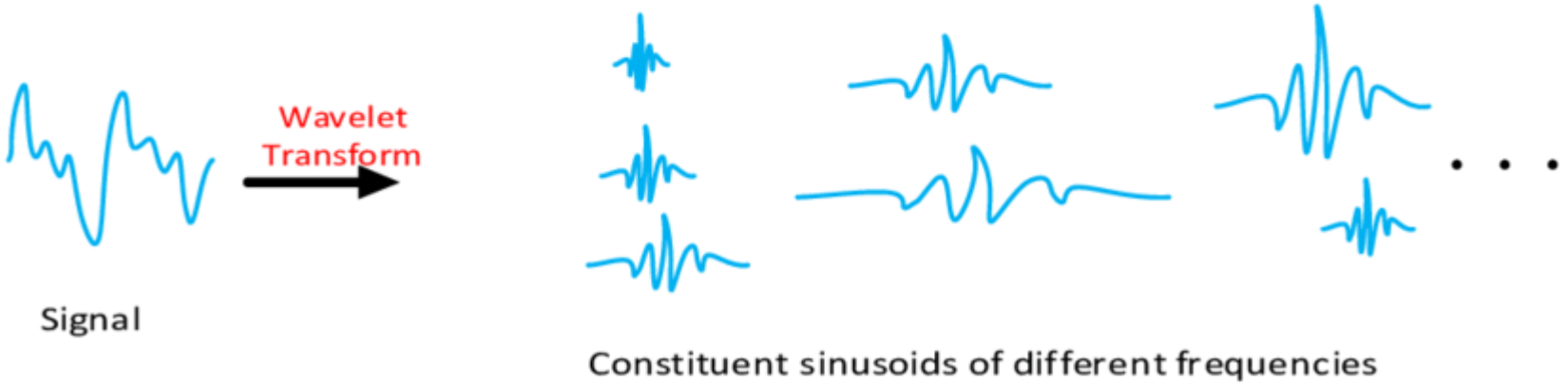
Data compression

- The data is encoded using fewer bits than the original representation.

Mathematical transform

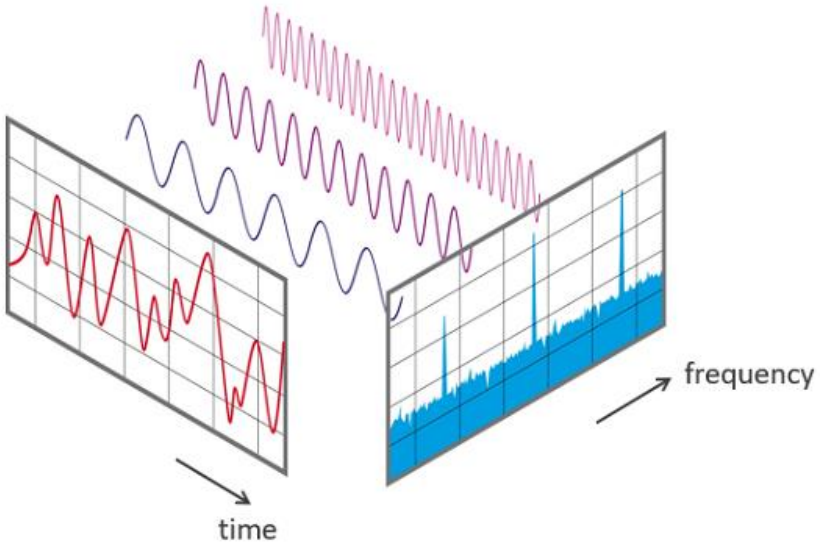
- Map the data to a new space and store only a small fraction of the strongest of the signal coefficients

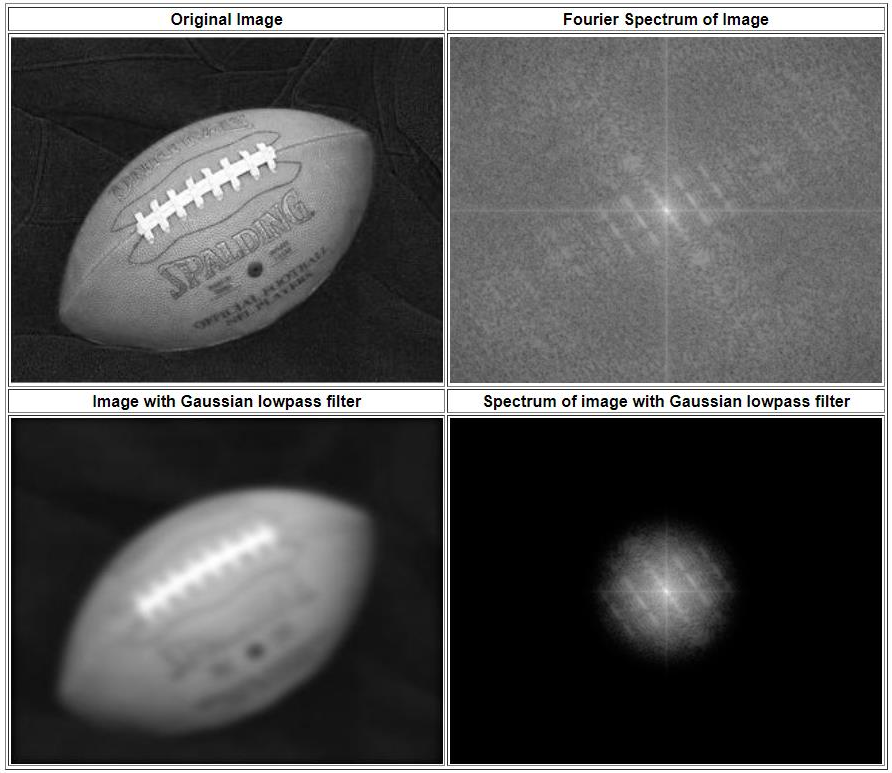




Mathematical transform: DFT

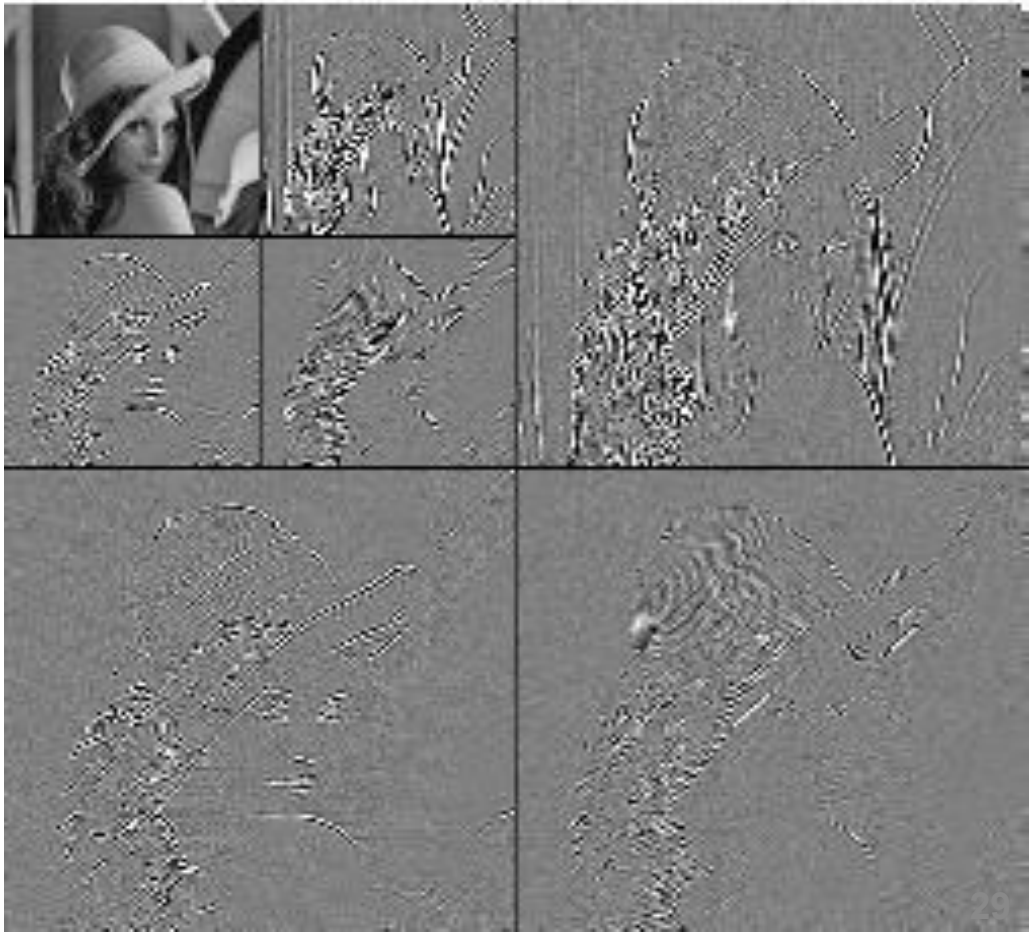
- Discrete Fourier transform decomposes a function in time domain into the frequency one
  - E.g., decompose an audio wave in the time domain into its constituent frequencies and volume (amplitude)





Mathematical transform: Wavelet

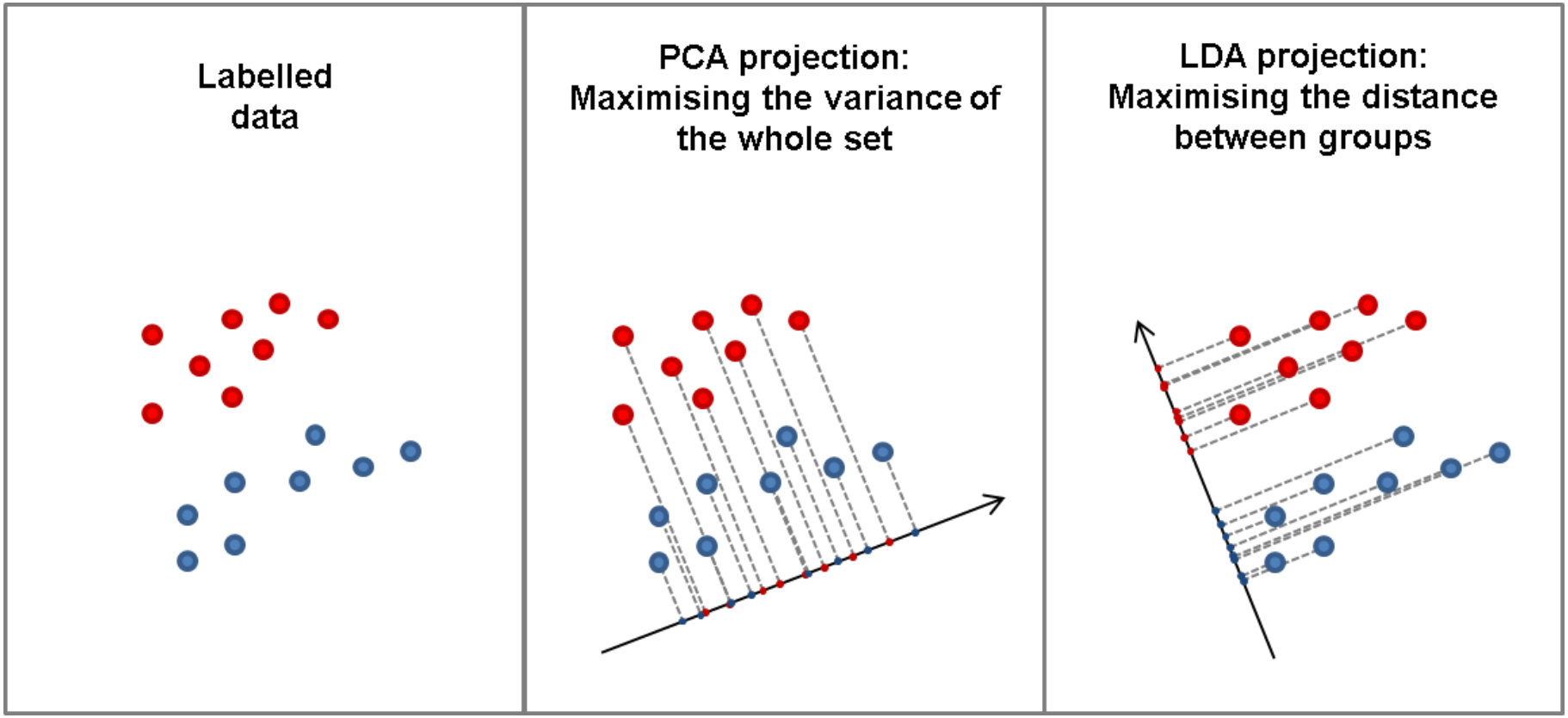
- Wavelet transform: decompose a (n-dimensional) signal into different frequency sub-bands



- Preserve relative distance between objects at different levels of resolution
- Allow natural clusters to become more distinguishable
- Used for image compression

Subspace transform

- PCA and LDA both look for linear combinations of variables which best explain the data.

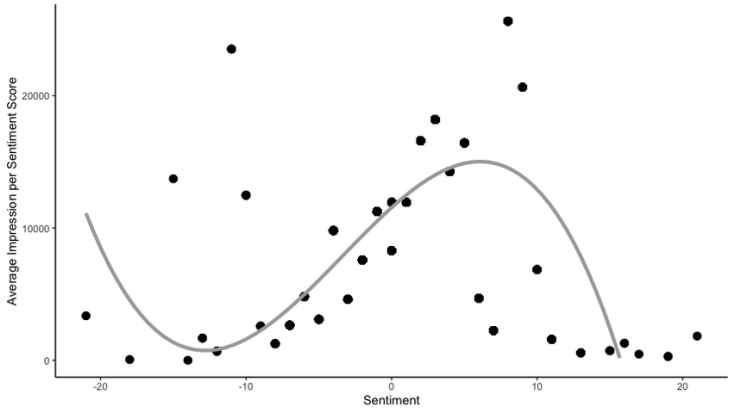


Attribute subset selection

- A way to remove redundant and/or irrelevant attributes
  - The purchase price of a product already includes the amount of sales tax paid $\rightarrow$ redundant
  - Predicting a student's GPA does not require his full name $\rightarrow$ irrelevant
- There are 2^d possible attribute combinations of d attributes.

Parametric numerosity reduction

- Parametric methods stores only the model's parameters, while discarding the original data (except possible outliers)

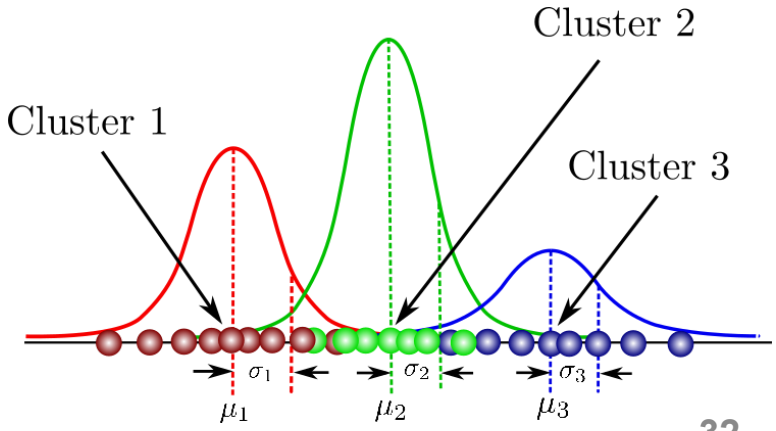


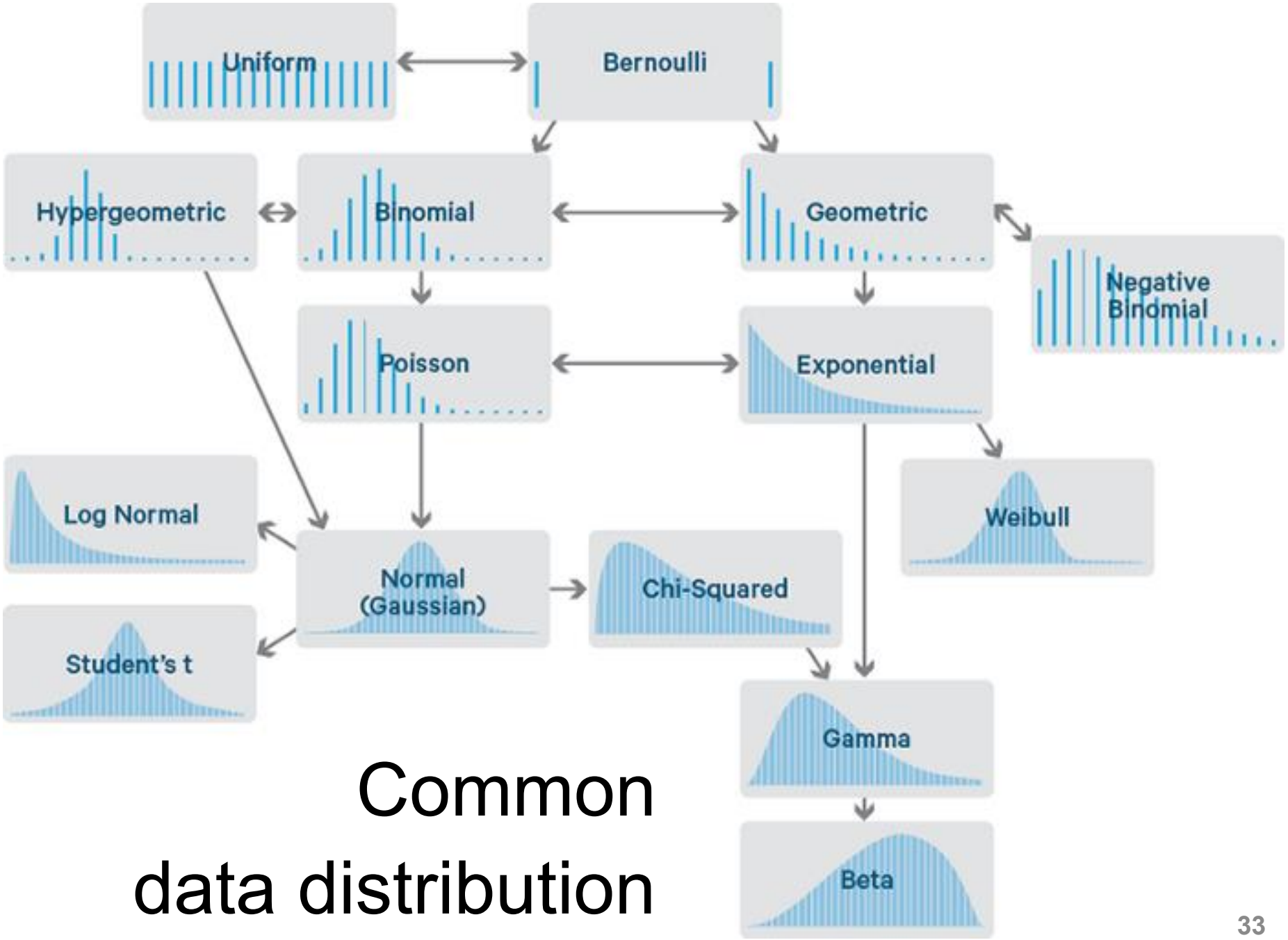
Regression analysis

- The parameters are estimated to give a "best fit" of the data
- The best fit is evaluated by using the least squares method or other criteria

Gaussian mixture model

- All data points are assumed to come from a mixture of Gaussian distributions
- The best fit is estimated by using the Expectation-Maximization algorithm.

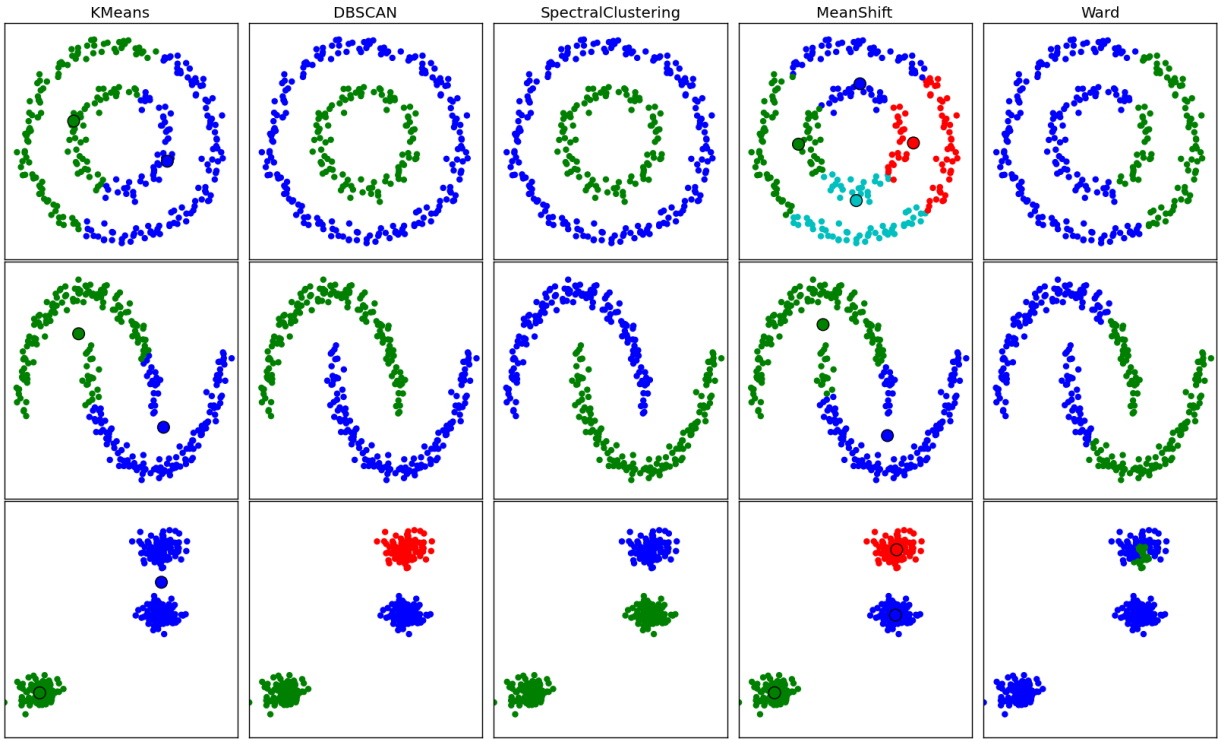




Common data distribution

Why data distribution is important?

- Correctly identifying the data distribution helps apply suitable data analysis or machine learning methods.



Clustering methods work differently on three exemplar data distributions.

Image credit:
ogrisel.github.io

Non-parametric numerosity reduction

- Non-parametric methods do not assume models.
- A histogram divides the data into buckets and stores the average sum for each bucket
- Equal-width (distance) binning
  - Divide the range into N intervals of equal width, $W = (B - A)/N$
    - where A and B are the lowest and highest values of the attribute
  - Outliers may dominate presentation, skewed data is not handled well
- Equal-depth (frequency) binning
  - Divide the range into N intervals, each containing approximately same number of samples $\rightarrow$ good data scaling

Histogram analysis: An example

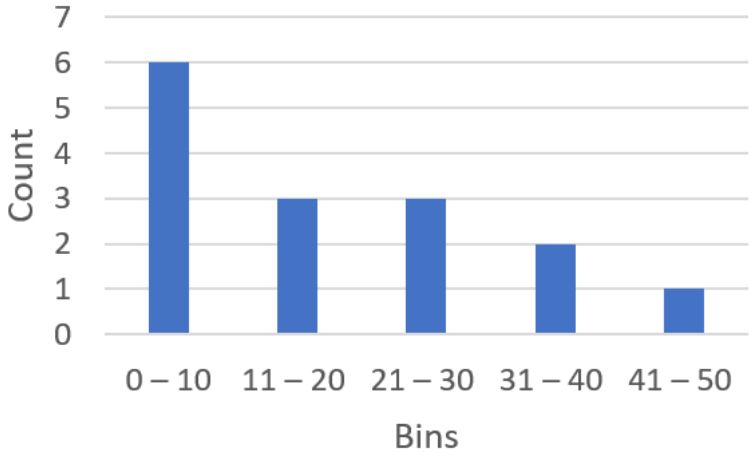
- Consider the values aside

0, 2, 5, 8, 8, 10, 15, 15, 20, 25, 25, 30, 35, 40, 49

- Partition the above data into 5 bins

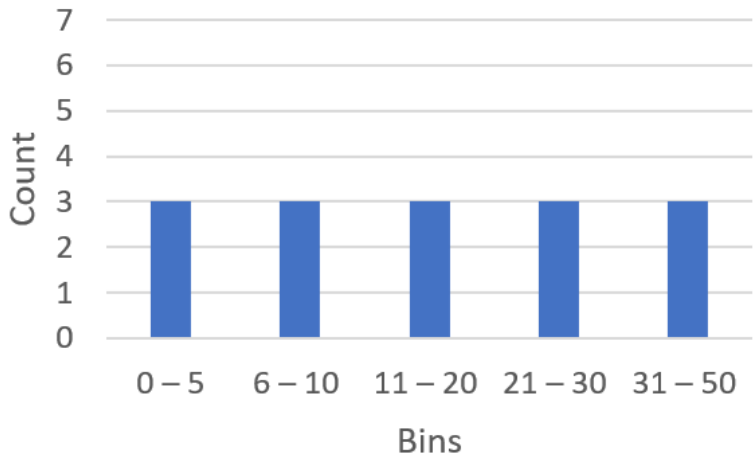
Equal-width binning

	Bin range	Values
Bin 1	0 – 10	0, 2, 5, 8, 8, 10
Bin 2	11 – 20	15, 15, 20
Bin 3	21 – 30	25, 25, 30
Bin 4	31 – 40	35, 40
Bin 5	41 – 50	49



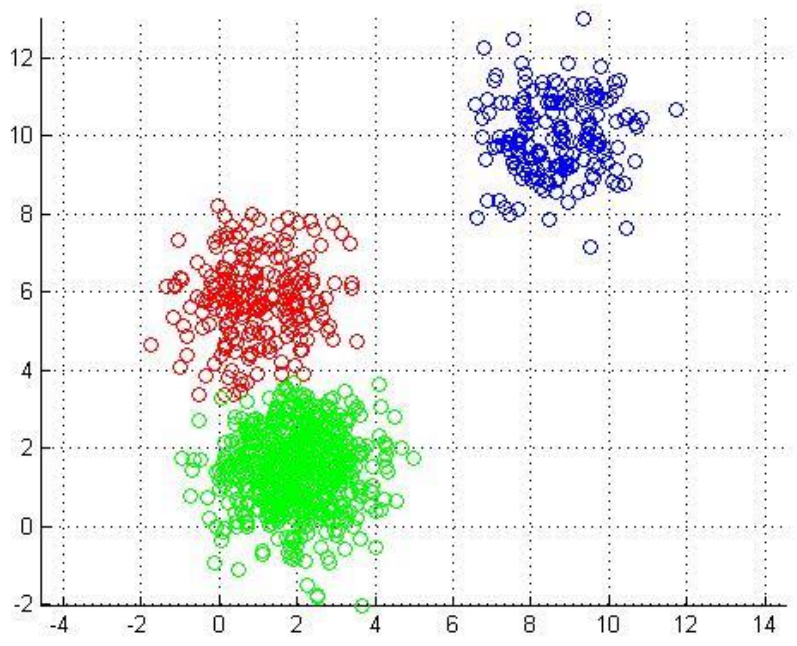
Equal-frequency binning

	Bin range	Values
Bin 1	0 – 5	0, 2, 5
Bin 2	6 – 10	8, 8, 10
Bin 3	11 – 20	15, 15, 20
Bin 4	21 – 30	25, 25, 30
Bin 5	31 – 50	35, 40, 49

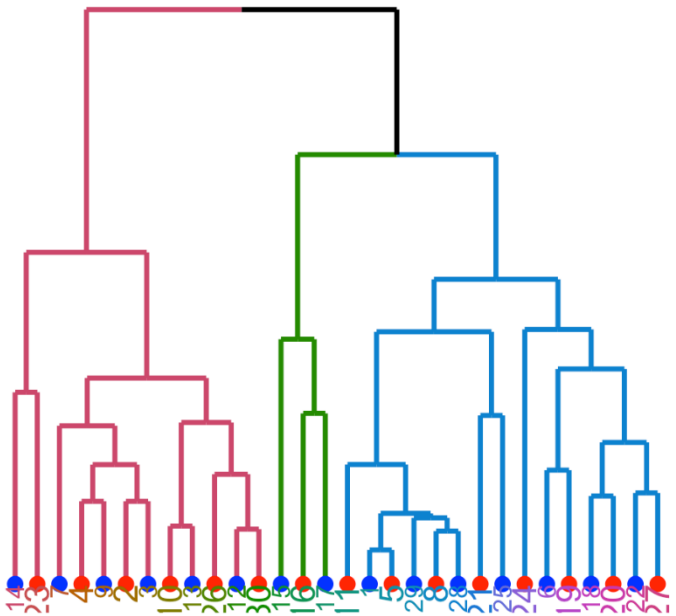


Clustering

- Partition the data into clusters based on similarity, and store cluster representation (e.g., centroid and diameter) only
- Very effective if data is clustered but not if data is “smeared”



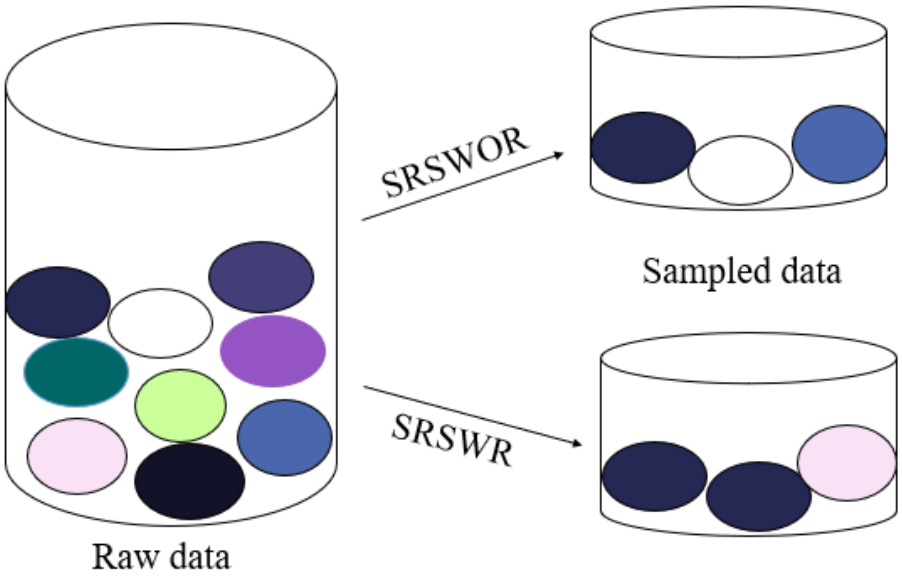
Distance-based clustering



Hierarchical clustering

Random sampling

- Choose a representative subset s of the whole dataset D
- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Simple random sampling: any item can be selected with an equal probability $\rightarrow$ poor performance in skewed data

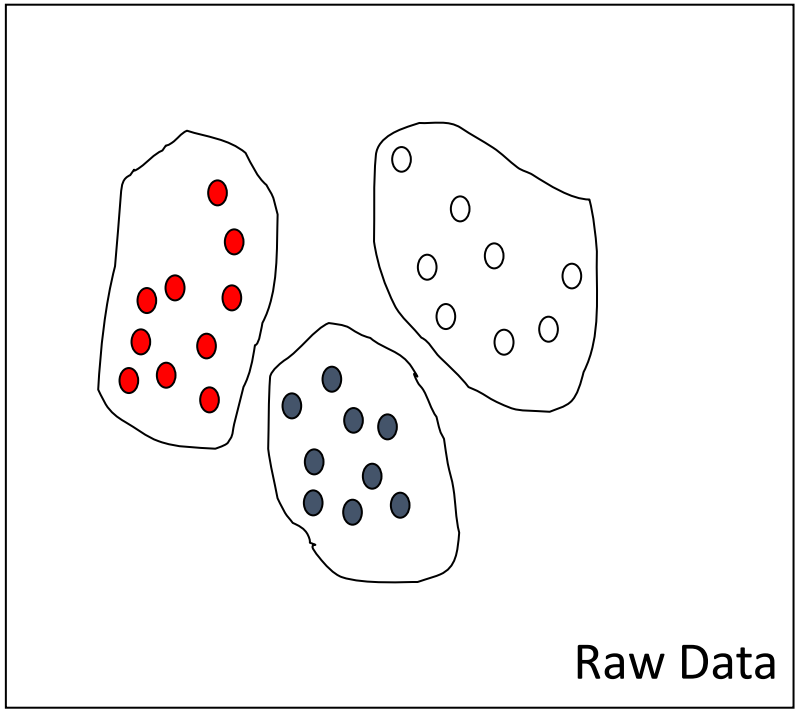


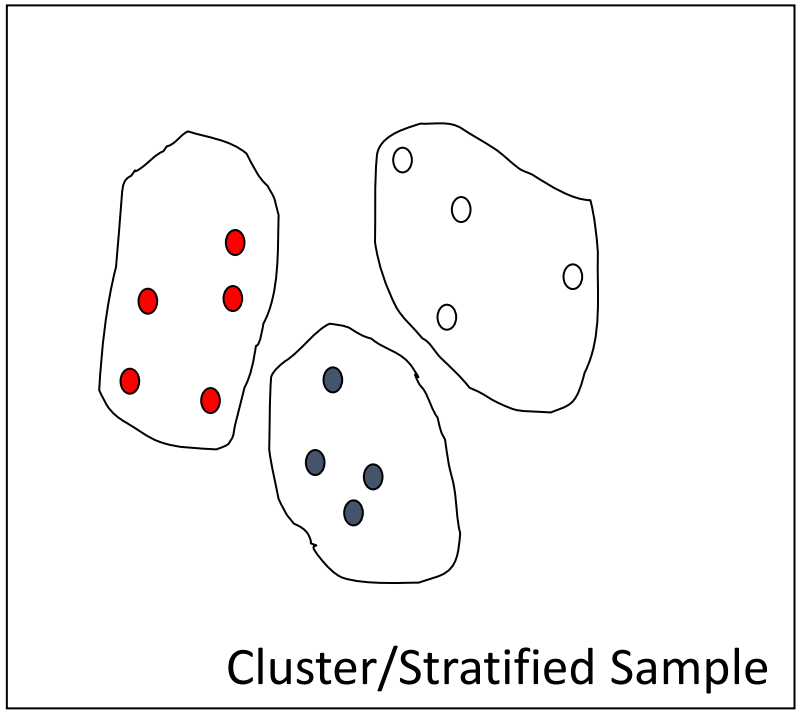
Once an object is selected, it is removed from the population

A selected object is not removed from the population

Stratified random sampling

- Partition the dataset and draw samples from each partition proportionally → good for skewed data



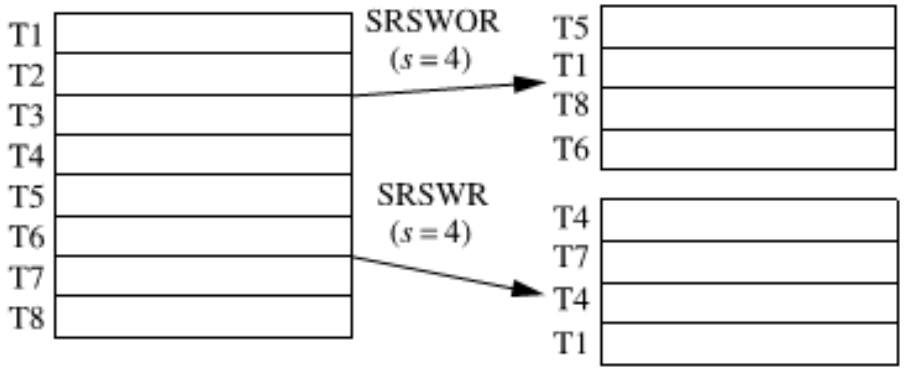


Sampling: An example

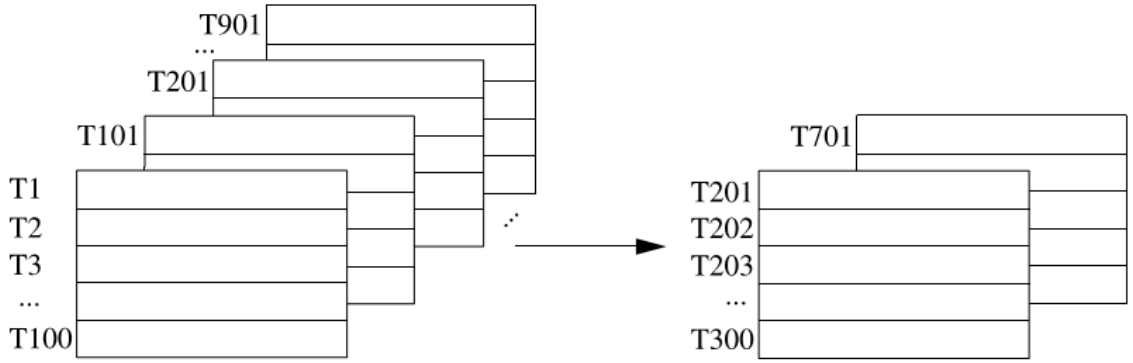
Startified sample
(according to *age*)

T38	youth
T256	youth
T307	youth
T391	youth
T96	middle_aged
T117	middle_aged
T138	middle_aged
T263	middle_aged
T290	middle_aged
T308	middle_aged
T326	middle_aged
T387	middle_aged
T69	senior
T284	senior

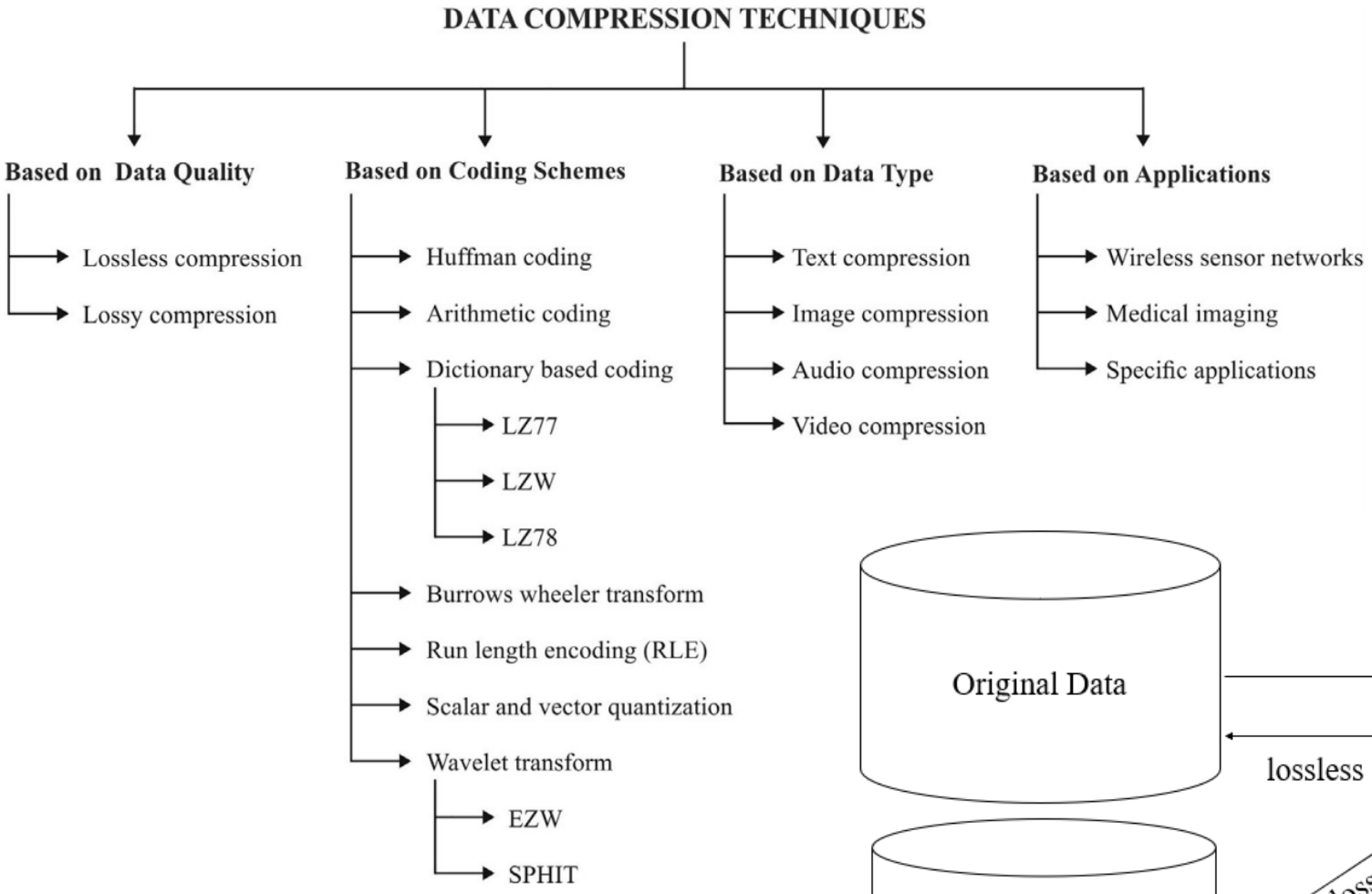
T38	youth
T391	youth
T117	middle_aged
T138	middle_aged
T290	middle_aged
T326	middle_aged
T69	senior

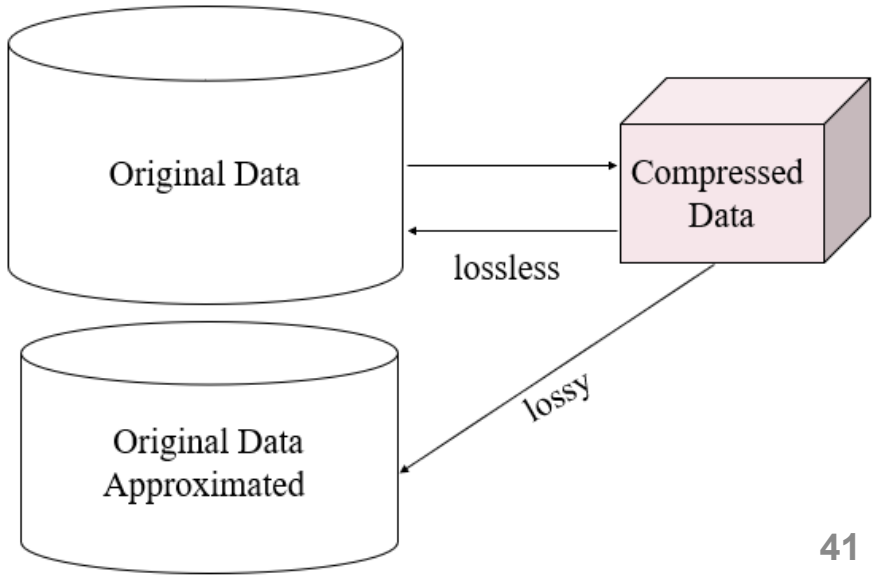


Cluster sample
($s = 2$)



Data compression





Data transformation

Data normalization

- Let A be a numeric attribute with n observed values, $v_1, .., v_n$
- Min-max normalization

$$v_i' = \frac{v_i - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

  - where max_A, new_max_A, min_A and new_min_A are the original and modified maximum and minimum values of attribute A, respectively.

- Decimal scaling: $v_i' = \frac{v_i}{10^j}$
  - where j is the smallest integer such that $\max(|v_i'|) < 1$
  - Move the decimal point of values of A, in which the number of decimal points moved depends on the maximum absolute value of A

Data normalization

- Z-score normalization: $v_i' = \frac{v_i - \mu_A}{\sigma_A}$
  - Where μ_A and σ_A are the mean and standard deviation, respectively, of attribute A
- A variation that is more robust to outliers: replace σ_A by mean absolute deviation of A

$$s_A = \frac{1}{n}(|v_1 - \mu_A| + |v_2 - \mu_A| + \cdots + |v_n - \mu_A|)$$

Data normalization: An example

- Consider the following sorted points

4	8	15	21	21	24	25	28	34

- Perform min-max normalization to the new range [-1, 1]
  - $min = 4, max = 34, new\_min = -1, new\_max = 1$

-1	-0.733	-0.267	0.133	0.133	0.333	0.4	0.6	1

- Perform decimal scaling normalization
  - $max = 34 \rightarrow j = 2$

0.04	0.08	0.15	0.21	0.21	0.24	0.25	0.28	0.34

- Perform Z-score normalization
  - $mean = 20, std = 8.994$

-1.779	-1.334	-0.556	0.111	0.111	0.445	0.556	0.889	1.557

Quiz 02: Data normalization

1. Consider the following 1D data series, which includes 10 data points sorted in ascending order.

   5, 12, 18, 23, 35, 42, 50, 55, 63, 70

   Transform the value 35 to a new value using

   - Min-max normalization with the range [-1.0, 1.0]
   - z-score normalization
   - Decimal scaling

2. How to perform normalization, without coding from scratch, in **Python**?

   For each available function, show the result for the above data.

Data discretization

- The range of a continuous attribute is divided into intervals, whose labels are used to replace actual data values.
- It aims to reduce data size or prepare for further analysis.
- Typical methods can be applied recursively, e.g., clustering and histogram-binning.

References

- Jiawei Han, Micheline Kamber, and Jian Pei, 2011. Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann Publishers Inc. Chapter 2 and Chapter 3.


...the end.