Data Preprocessing

The process of making the data more suitable for data mining

WHAT IS DATA?

- Collection of data objects and their attributes
- An attribute is a property or characteristic of an object
  - Examples: eye color of a person, temperature, etc.
  - Attribute is also known as variable, field, characteristic, or feature
- A collection of attributes describe an object
- Object is also known as record, point, case, sample, entity, or instance

Attributes

Objects

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	No
2	young	false	false	good	No
3	young	true	false	good	Yes
4	young	true	true	fair	Yes
5	young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

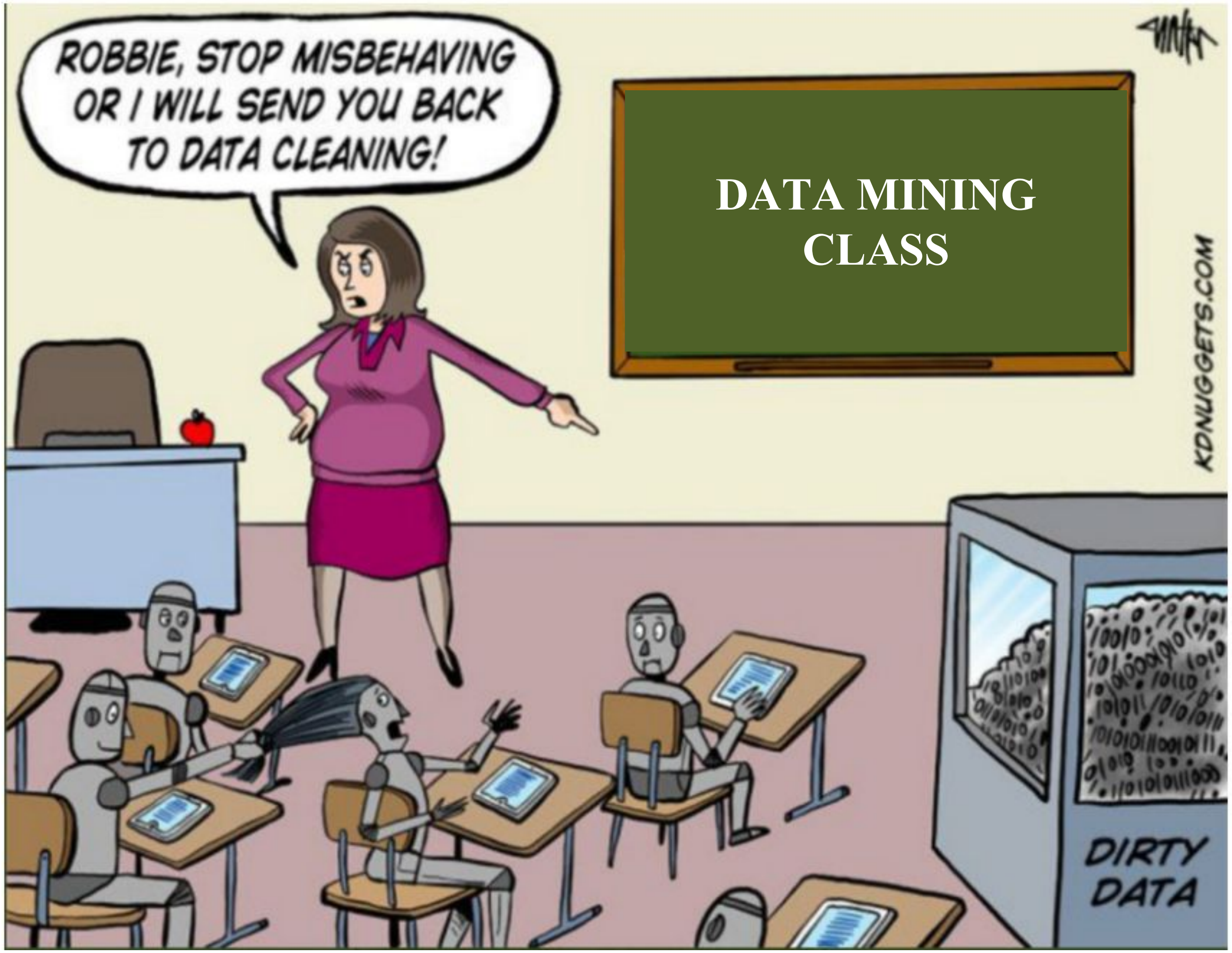


Image source: https://www.kdnuggets.com/2017/09/cartoon-machine-learning-class.html

Why Data Preprocessing?

- Data in the real world is dirty: Lots of potentially incorrect data, e.g., instrument faulty, human or computer error, transmission error
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - e.g., occupation=“”
  - noisy: containing errors or outliers
    - e.g., Salary=“-10”
  - inconsistent: containing discrepancies in codes or names
    - e.g., Age=“42” Birthday=“03/07/1997”
    - e.g., Was rating “1,2,3”, now rating “A, B, C”
    - e.g., discrepancy between duplicate records

Why Is Data Preprocessing Important?

- **No quality data, no quality mining results!**
  - **Quality decisions must be based on quality data**
    - **e.g., duplicate or missing data may cause incorrect or even misleading statistics.**
  - **Data warehouse needs consistent integration of quality data**

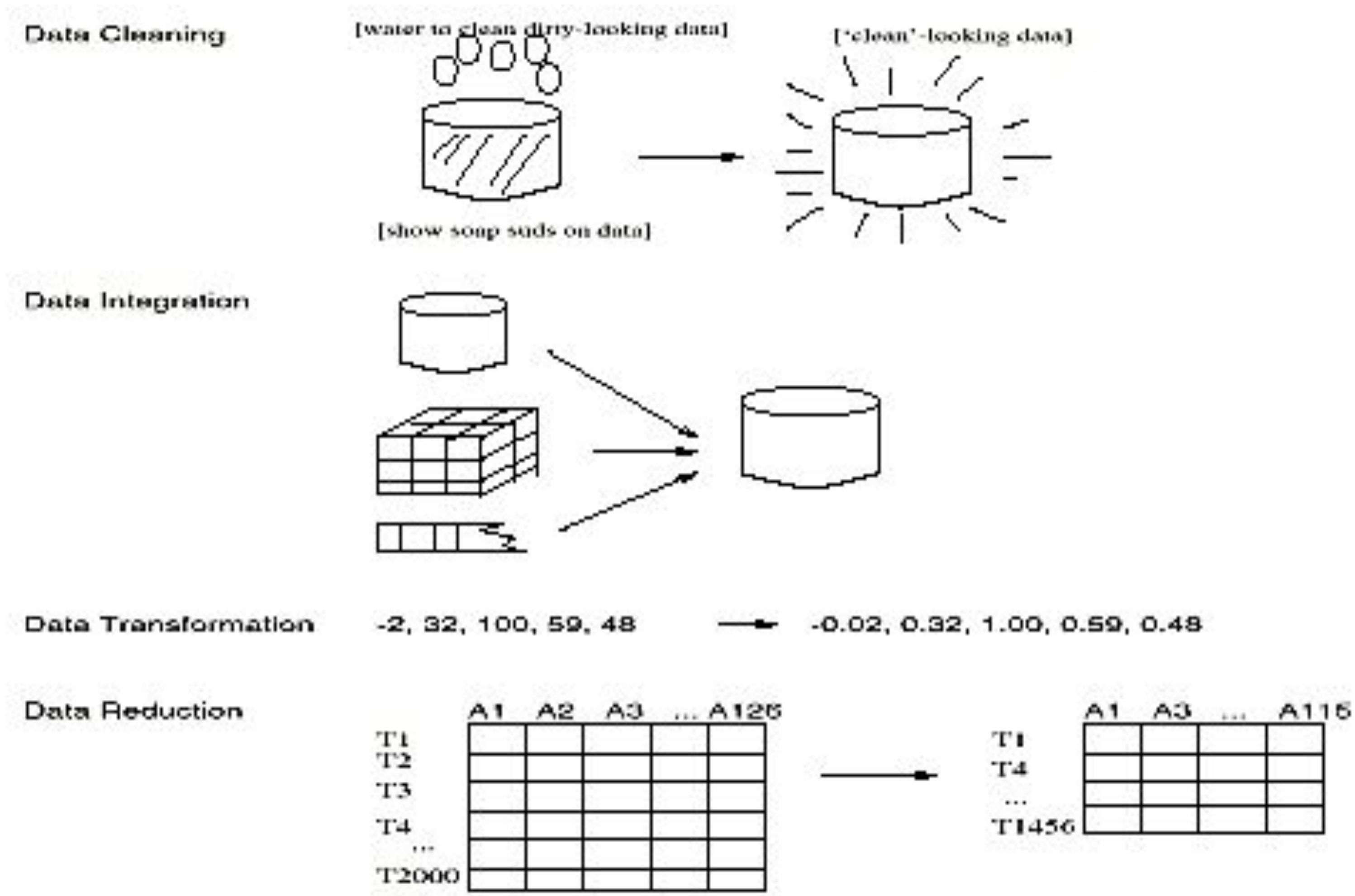
Multi-Dimensional Measure of Data Quality

- **Measures for data quality: A multidimensional view**
  - **Accuracy: correct or wrong, accurate or not**
  - **Completeness: not recorded, unavailable, …**
  - **Consistency: some modified but some not, dangling, …**
  - **Timeliness: timely update?**
  - **Believability: how trustable the data are correct?**
  - **Interpretability: how easily the data can be understood?**

Knowledge discovery from data

- **Major Tasks in Data Preprocessing**
  - **Data cleaning**
    - **Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies**
  - **Data integration**
    - **where multiple data sources may be combined**
  - **Data reduction**
    - **Dimensionality reduction, Data compression**
  - **Data transformation**
    - **where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations**

Major Tasks in Data Preprocessing



Data Cleaning

- **Data cleaning tasks**
  - **Fill in missing values**
  - **Identify outliers and smooth out noisy data**
  - **Correct inconsistent data**

Incomplete (Missing) Data

- Data is not always available
  - E.g., many tuples have no recorded value for several attributes, such as customer income in sales data
- Missing data may be due to
  - equipment malfunction
  - inconsistent with other recorded data and thus deleted
  - data not entered due to misunderstanding
  - certain data may not be considered important at the time of entry
  - not register history or changes of the data
- Missing data may need to be inferred

How to Handle Missing Data?

- **Ignore the tuple: usually done when class label is missing (when doing classification—not effective in certain cases)**
- **Fill in the missing value manually: tedious + infeasible?**
- **Fill in it automatically with**
  - **a global constant : e.g., “unknown”, a new class?!**
  - **the attribute mean**
  - **the attribute mean for all samples belonging to the same class: smarter**
  - **the most probable value: inference-based such as Bayesian formula or decision tree**

How to Handle Missing Data?

	col1	col2	col3	col4	col5
0	2	5.0	3.0	6	NaN
1	9	NaN	9.0	0	7.0
2	19	17.0	NaN	9	NaN

mean() →

	col1	col2	col3	col4	col5
0	2.0	5.0	3.0	6.0	7.0
1	9.0	11.0	9.0	0.0	7.0
2	19.0	17.0	6.0	9.0	7.0

Noisy Data

- **Noise: random error in a measured variable**
- **Incorrect attribute values may be due to**
  - **faulty data collection instruments**
  - **data entry problems**
  - **data transmission problems**
  - **technology limitation**
  - **inconsistency in naming convention**

How to Handle Noisy Data?

- **Binning**
  - first sort data and partition into (equal-frequency) bins
  - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- **Regression**
  - smooth by fitting the data into regression functions
- **Clustering**
  - detect and remove outliers
- **Combined computer and human inspection**
  - detect suspicious values and check by human (e.g., deal with possible outliers)
- `

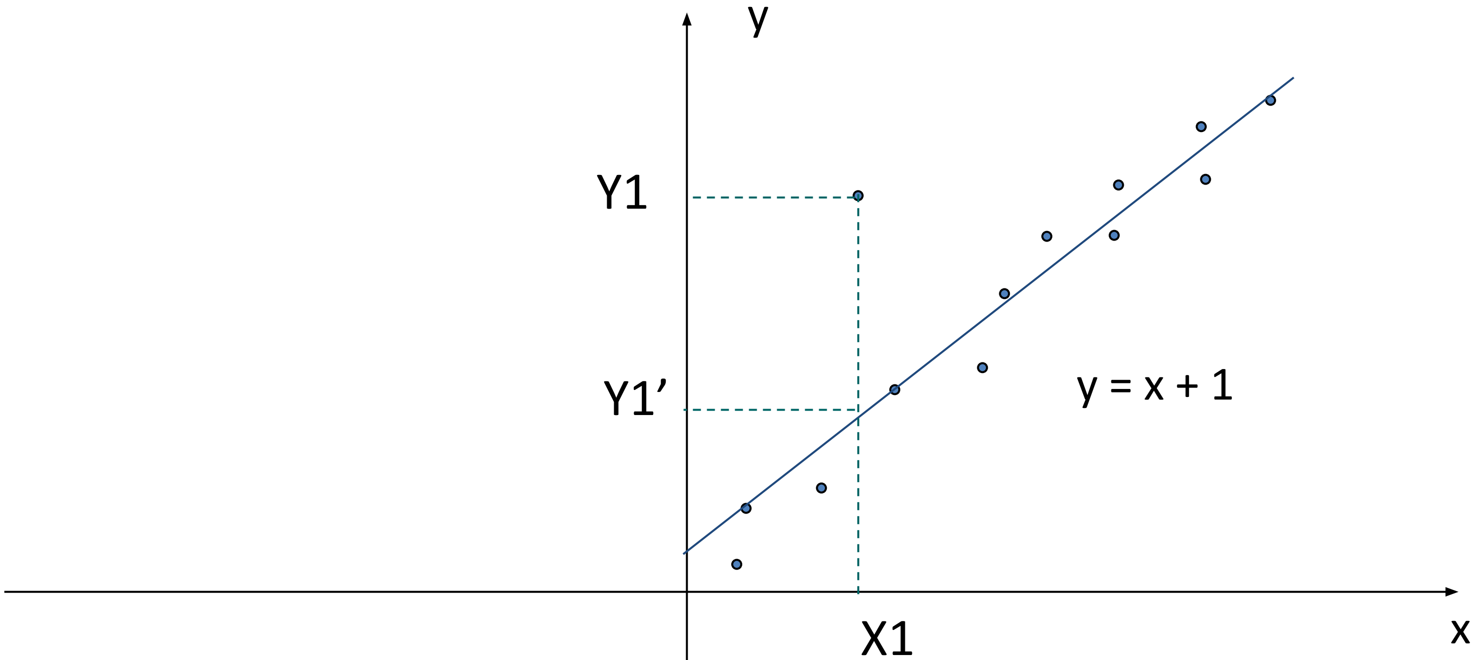
Sorted data for *price* (in dollars): 4, 8, 15, 21, 21, 24, 25, 28, 34

Partition into (equal-frequency) bins:
Bin 1: 4, 8, 15
Bin 2: 21, 21, 24
Bin 3: 25, 28, 34

Smoothing by bin means:
Bin 1: 9, 9, 9
Bin 2: 22, 22, 22
Bin 3: 29, 29, 29

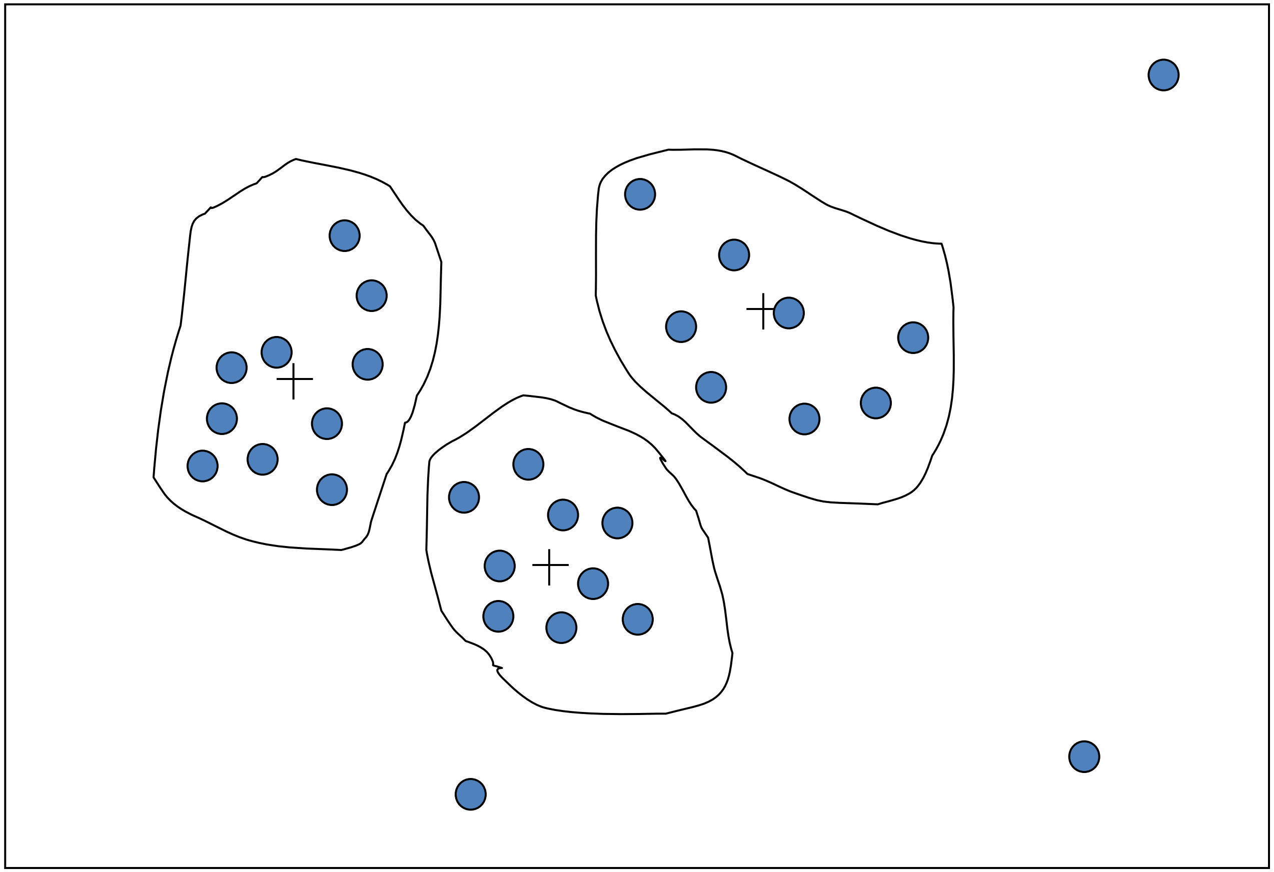
Smoothing by bin boundaries:
Bin 1: 4, 4, 15
Bin 2: 21, 21, 24
Bin 3: 25, 25, 34

Regression



- Linear regression (best line to fit two variables)
- Multiple linear regression (more than two variables, fit to a multidimensional surface

Cluster Analysis



Data Integration

- **Data integration:**
  - **Combines data from multiple sources into a coherent store**
- **Schema integration: e.g., A.cust-id $\equiv$ B.cust-#**
  - **Integrate metadata from different sources**
- **Entity identification problem:**
  - **Identify real world entities from multiple data sources, e.g., Bill Clinton = William Clinton**
- **Detecting and resolving data value conflicts**
  - **For the same real world entity, attribute values from different sources are different**
  - **Possible reasons: different representations, different scales, e.g., metric vs. British units**

Handling Redundancy in Data Integration

- **Redundant data occur often when integration of multiple databases**
  - **Object identification: The same attribute or object may have different names in different databases**
  - **Derivable data: One attribute may be a "derived" attribute in another table, e.g., annual revenue**
- **Redundant attributes may be able to be detected by correlation analysis**

Data Reduction Strategies

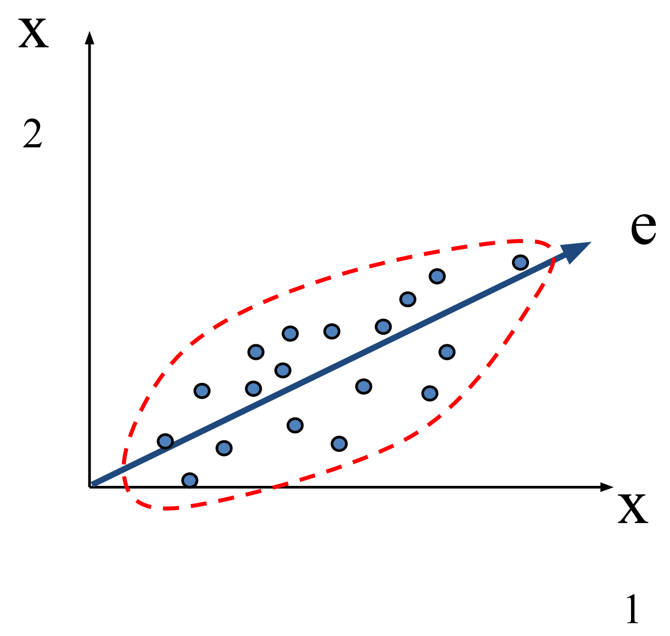
- **Data reduction: Obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results**
- **Why data reduction? — A database/data warehouse may store terabytes of data. Complex data analysis may take a very long time to run on the complete data set.**
- **Data reduction strategies**
  - **Dimensionality reduction, e.g., remove unimportant attributes**
    - **Principal Components Analysis (PCA)**
    - **Feature subset selection, feature creation**
  - **Data compression**

Data Reduction : Dimensionality Reduction

- Curse of dimensionality
  - When dimensionality increases, data becomes increasingly sparse
  - Density and distance between points, which is critical to clustering, outlier analysis, becomes less meaningful
  - The possible combinations of subspaces will grow exponentially
- Dimensionality reduction
  - Avoid the curse of dimensionality
  - Help eliminate irrelevant features and reduce noise
  - Reduce time and space required in data mining
  - Allow easier visualization
- Dimensionality reduction techniques
  - Principal Component Analysis
  - feature selection

Principal Component Analysis (PCA)

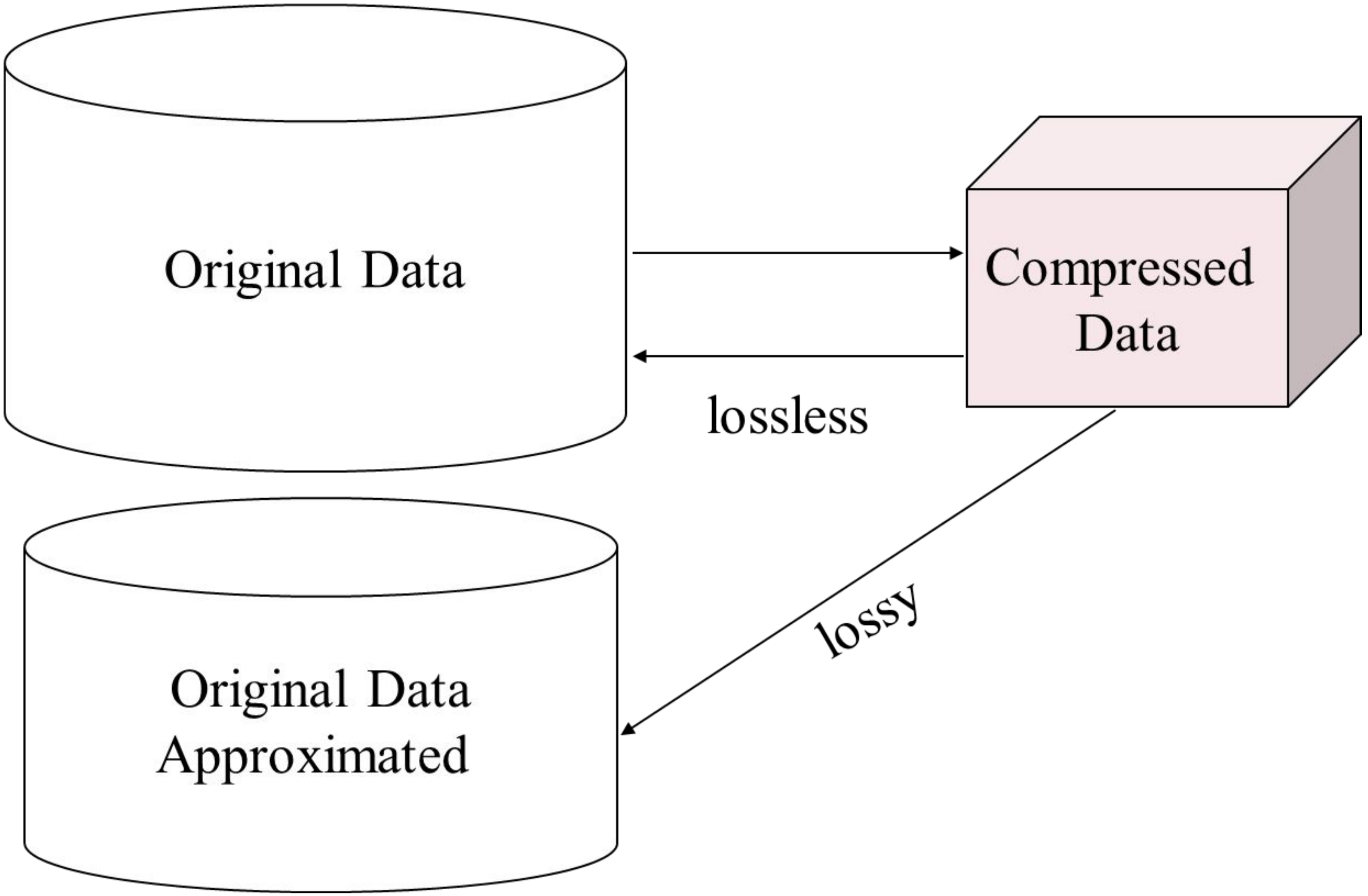
- **Find a projection that captures the largest amount of variation in data**
- **The original data are projected onto a much smaller space, resulting in dimensionality reduction. We find the eigenvectors of the covariance matrix, and these eigenvectors define the new space**



Attribute Subset Selection

- **Another way to reduce dimensionality of data**
- **Redundant attributes**
  - **Duplicate much or all of the information contained in one or more other attributes**
  - **E.g., purchase price of a product and the amount of sales tax paid**
- **Irrelevant attributes**
  - **Contain no information that is useful for the data mining task at hand**
  - **E.g., students' ID is often irrelevant to the task of predicting students' GPA**

Data Compression

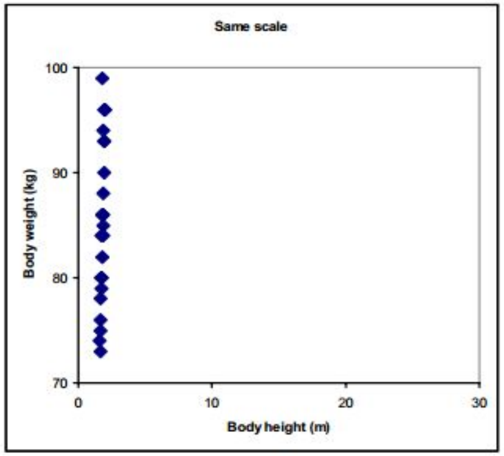


Data Transformation

- **A function that maps the entire set of values of a given attribute to a new set of replacement values s.t. each old value can be identified with one of the new values**
- **Methods**
  - **Smoothing: Remove noise from data (binning, clustering, regression)**
  - **Normalization: Scaled to fall within a smaller, specified range**
    - **min-max normalization**
    - **z-score normalization**
    - **normalization by decimal scaling**

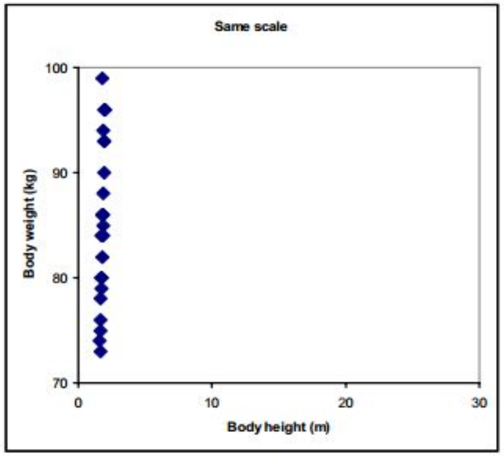
Data Transformation

Height (m)	**1.8**	**1.61**	**1.68**	**1.75**	**1.74**	**1.67**	**1.72**	**1.98**	**1.92**	**1.7**	**1.77**	**1.92**
Weight (kg)	**86**	**74**	**73**	**84**	**79**	**78**	**80**	**96**	**90**	**80**	**86**	**93**
Height (m)	**1.6**	**1.85**	**1.87**	**1.94**	**1.89**	**1.89**	**1.86**	**1.78**	**1.75**	**1.8**	**1.68**	
Weight (kg)	**75**	**84**	**85**	**96**	**94**	**86**	**88**	**99**	**80**	**82**	**76**	

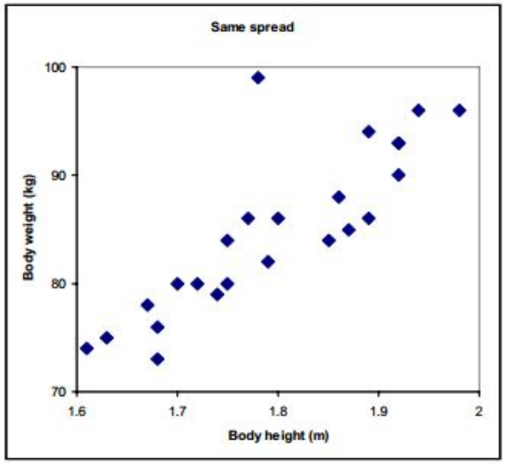


Data Transformation

- **We can see that the data points only spread in the vertical direction because body weight has much larger numerical range than body height.**

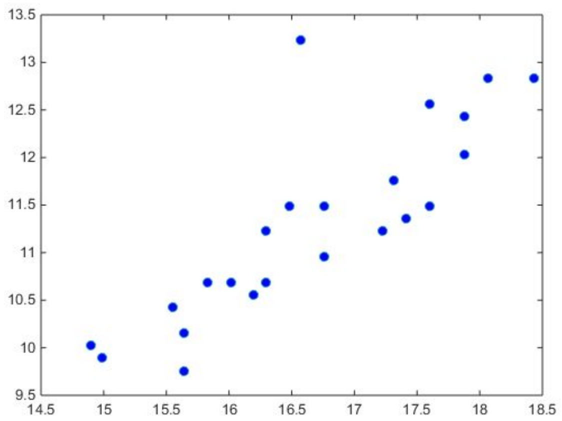


- **Let's zoom the Figure**
  - **There is strong correlation between body height and body weight, except for one outlier in the data.**



Data Transformation

- **Solution:**
  - **Scaling : In order to give both variable, body weight and height, equal weight in the data, we standardized (scaling or weighting) them.**
  - **There are may ways, but the most common techniques are**
    - **min-max normalization**
    - **z-score normalization**
    - **normalization by decimal scaling**



min-max normalization

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

- The minimum value is 8
- The maximum value is 20

Assume, we want to scale data between 0 and 1,

- The new min is 0
- The new max is 1

A	A
8	0
20	1
10	0.16
15	0.58

Normalization

- **Z-score normalization (μ: mean, σ: standard deviation):**

$$v' = \frac{v - \mu_A}{\sigma_A}$$

  - **Ex. Let μ = 54,000, σ = 16,000. Then $73,000 is mapped to**

$$\frac{73{,}600 - 54{,}000}{16{,}000} = 1.225$$

- **Normalization by decimal scaling**

$$v' = \frac{v}{10^j}$$

Where j is the smallest integer such that Max(|v'|) < 1

Let the input data is: -10, 201, 301, -401, 501, 601, 701

To normalize the above data,

Step 1: *Maximum absolute value in given data(m): 701*

Step 2: *Divide the given data by 1000 (i.e j=3)*

Result: *The normalized data is: -0.01, 0.201, 0.301, -0.401, 0.501, 0.601, 0.701*

Acknowledgements

- **Lecture slides modified from**
  - **Data Mining: Concepts and Techniques (3rd ed.), Jiawei Han, Micheline Kamber, and Jian Pei**