CS500-Data Science Tools and Technique
Data Quality and Data Preprocessing

Bahar Ali
PhD Scholar,
National University Of Computer and Emerging Sciences,
Peshawar.

Contents

Data Quality

- Poor data quality negatively affects many data processing efforts
- The most important point is that poor data quality is an unfolding disaster
- Poor data quality costs extra for every data science/ data mining project.

Data Quality

- What kinds of data quality problems?
- How can we detect problems with the data?
- What can we do about these problems?

- Examples of data quality problems:
  - ❑ Noise and outliers
  - ❑ Missing values
  - ❑ Duplicate data

Data Quality

Noise

- For objects, noise is an extraneous object
- For attributes, noise refers to modification of original values
- Noise is anything that is not the "true" signal.
- It may have values close to your true signal.

Outliers

- Outliers are data objects with characteristics that are considerably different than most of the other data objects in the data set
    - The vast majority of time outliers are noise but sometimes a data point that is true signal can be an outlier
    - Outliers are the goal of our analysis i.e. Credit card fraud and Intrusion detection

Data Quality

Missing values

- Reasons for missing values:
  - Information is not collected (e.g., people decline to give their age)
  - Attributes may not be applicable to all cases (e.g., annual income is not applicable to children)

- Handling missing values:
  - Eliminate data objects or variables
  - Estimate missing values (Add mean value of the attribute)

Data Quality

Duplicate data

- Data set may include data objects that are duplicates
- Major issue when merging data from heterogeneous sources
- Examples:
  - ❑ Same person with multiple email addresses

Data Preprocessing

- Real-world data is often incomplete, inconsistent, lacking in certain behaviors or trends, and is likely to contain many errors
- Data preprocessing ensures
  - Accuracy: The quality of being correct.
  - Completeness: Data meets the expectations
  - Consistency: Keeping information uniform in one dataset with another dataset at the same point in time.
  - Timeliness: Timely updated
  - Believability: Trustable
  - Interpretability: Easily understandable

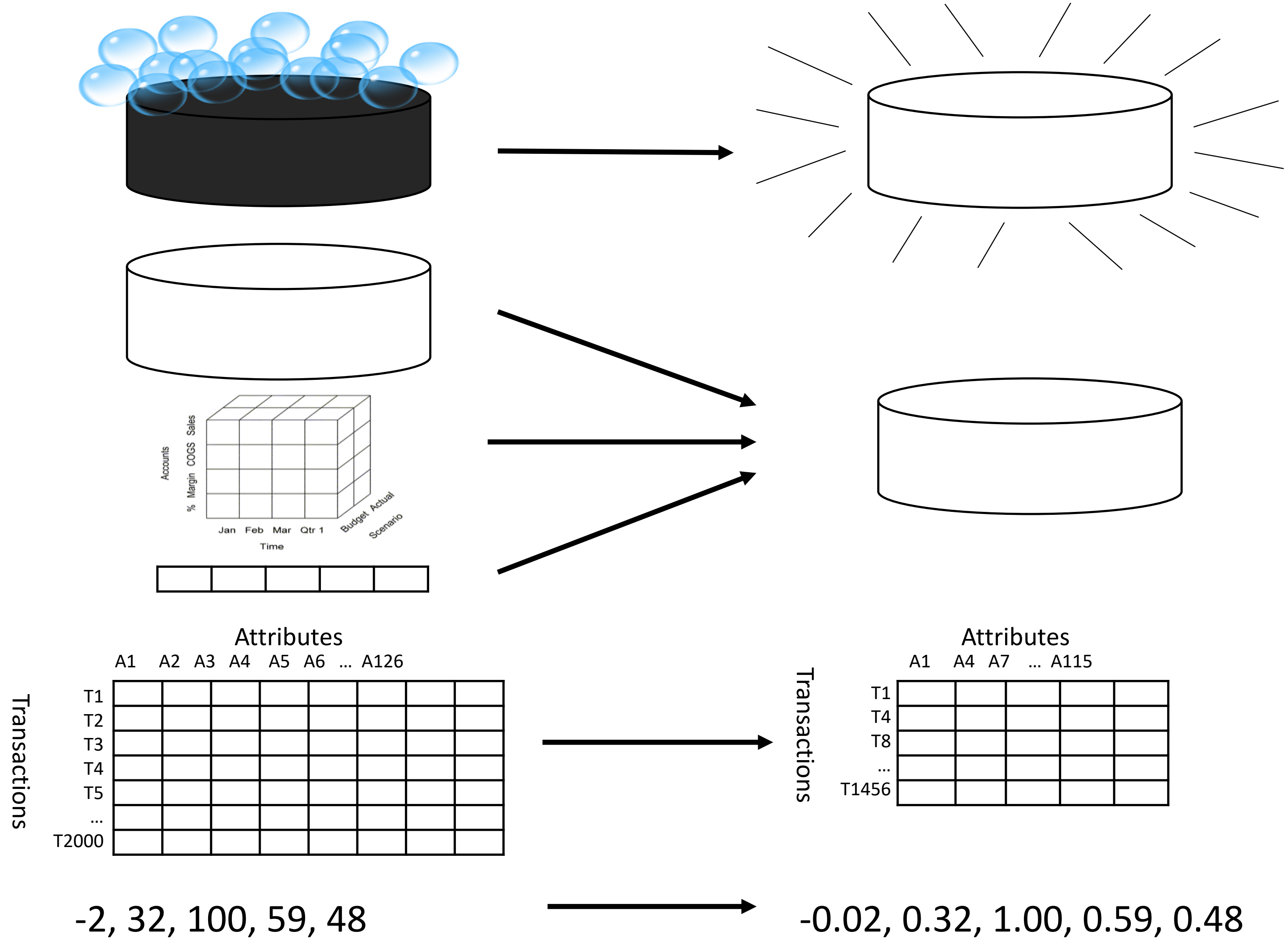
Forms of Data Preprocessing

1. Data cleaning

2. Data integration

3. Data reduction

4. Data transformation



1. Data Cleaning

- Data in the real world is;
  - Incomplete (Missing Data): lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - e.g., Occupation=“ ” (missing data)
  - Noisy: containing noise, errors, or outliers
    - e.g., Salary=“−10” (an error)
  - Inconsistent: containing discrepancies in codes or names
    - Age=“42”, Birthday=“03/07/2000”
    - Duplicate records not matching

1.1. Handling Incomplete (Missing) Data

- Ignoring the tuple: Not very effective unless a tuple contains several attributes with missing values
- Filling in the missing value manually: tedious, infeasible
- Filling in it automatically with
  - A global constant: e.g. ∞
  - The attribute mean (Very Effective)
  - The most probable value: inference-based such as a Bayesian formula or decision tree
    - e.g., the annual salary of a person can be inferred using his occupation and age

1.2. Handling Noisy Data

- Binning
  1. Partitioning the data into bins after sorting
  2. Smoothing by bin means, by bin median, by bin boundaries, etc.
- Regression
  - Fitting the data into regression functions
- Clustering
  - Detecting and removing outliers
- Combined computer and human inspection
  - Detecting suspicious values and checking by human

Binning

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 median:

  Bin 1: 8, 8, 8

  Bin 2: 21, 21, 21

  Bin 3: 28, 28, 28

- Smoothing by bin boundaries:

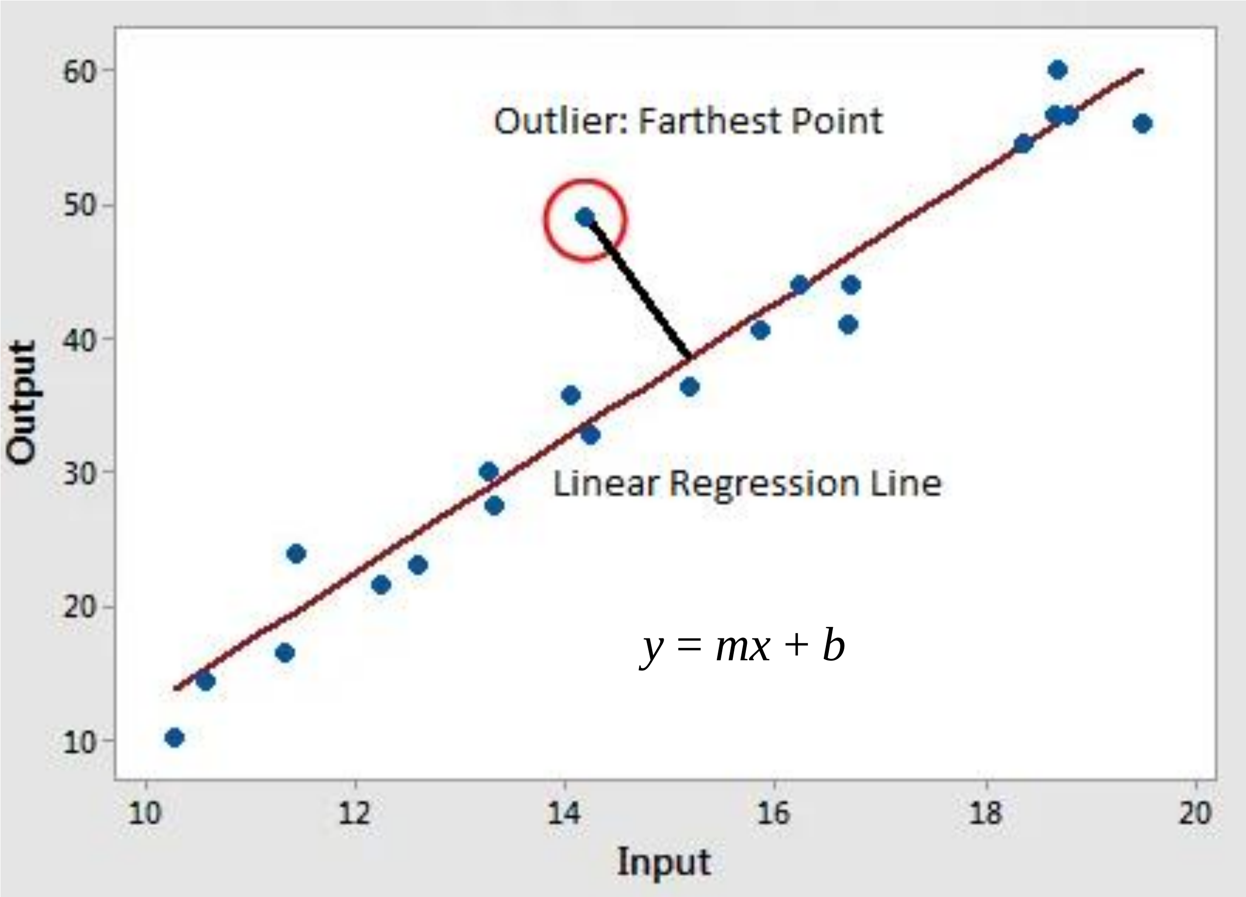
  Bin 1: 4, 4, 15

  Bin 2: 21, 21, 24

  Bin 3: 25, 25, 34

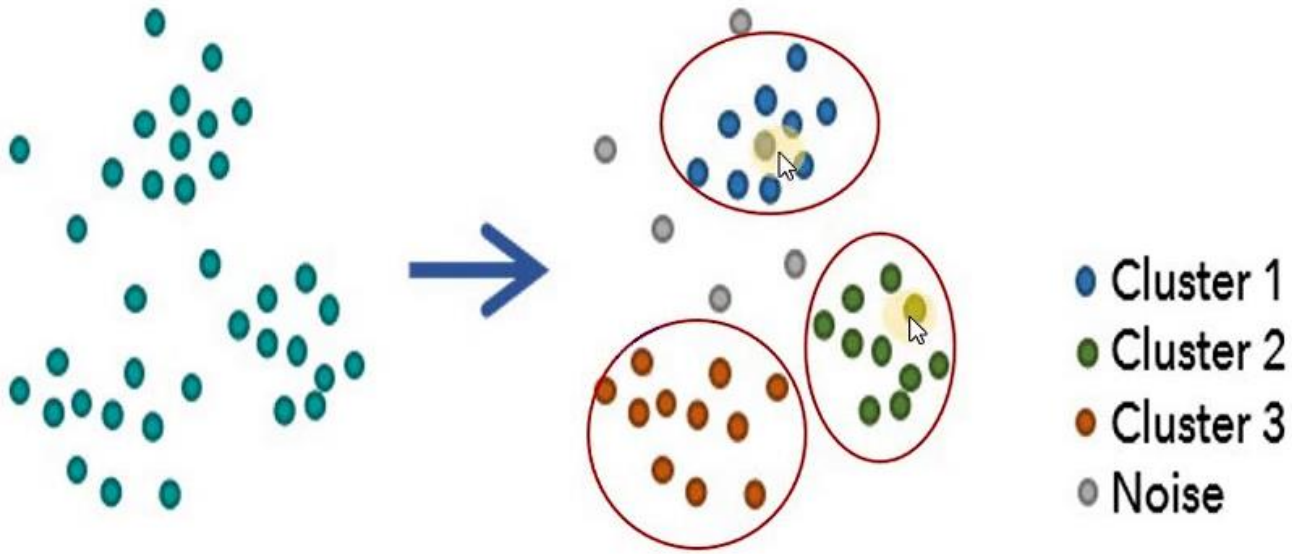
Regression

- A technique which is used to fit an equation to a dataset.



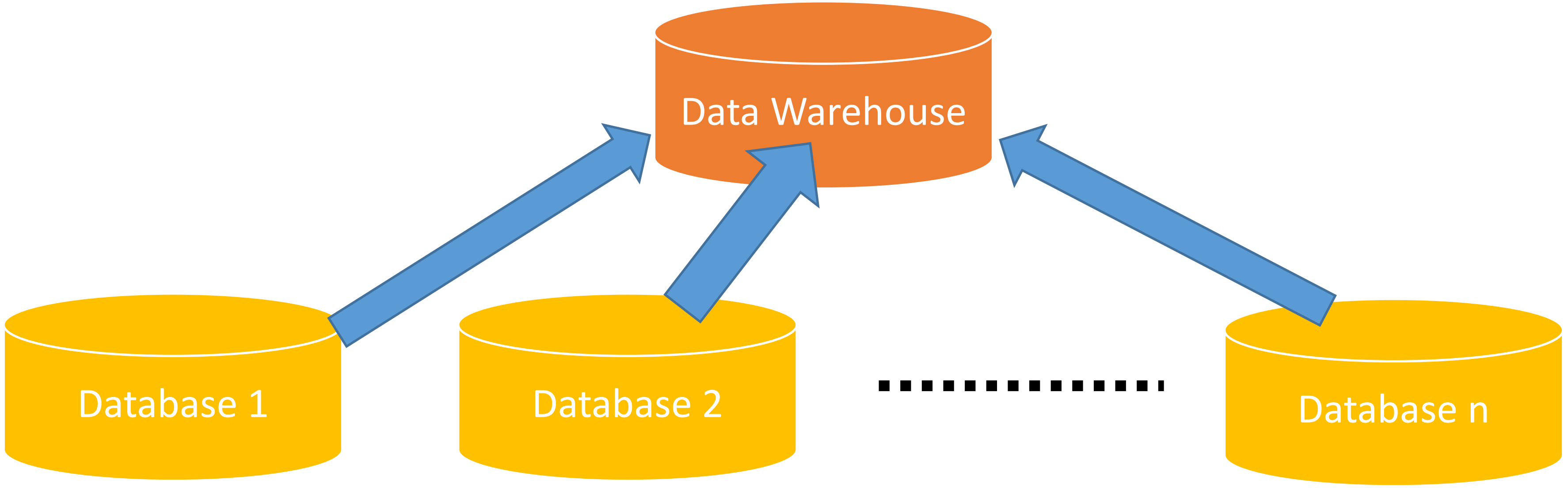
Clustering

- Clustering divides the data points into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups



2. Data Integration

- Data integration is combining data from multiple sources into a coherent data store, as in data warehouses



- There are a number of issues: schema integration, entity resolution, and redundancy/ inconsistency

2. Data Integration

- Schema integration
  - Integrating data from multiple sources with heterogeneous schemas
    - e.g customer-id ≡ customer# ?
- Entity resolution
  - Identifying the matching records from multiple sources (i.e., those corresponding to the same real-world entity)
    - e.g., Jae-Gil Lee, Jae Gil Lee, Jae G. Lee, and Jae Lee correspond to the same person
- Redundancy/ Inconsistency
  - Finding the true value of an attribute
    - e.g., The price of a book might vary at different stores

3. Data Reduction

- A database/data warehouse may store petabytes of data, and complex data analysis may take a very long time to run on the complete data set

- **Data reduction** is obtaining a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results

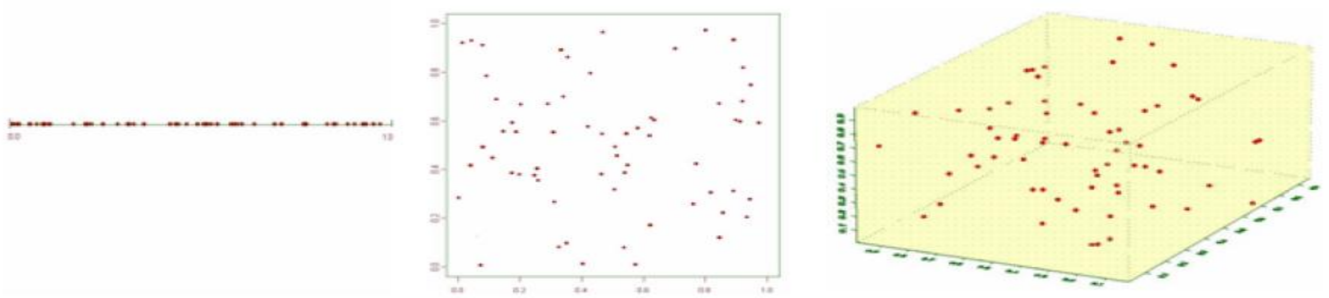
3. Data Reduction

Strategies

- Dimensionality reduction
  - Wavelet transform
  - Principal components analysis (PCA)
  - Feature subset selection, feature creation
- Numerosity reduction (some simply call it data reduction)
  - Regression
  - Histograms, clustering, sampling
  - Data cube aggregation
- Data compression

3.1 Dimensionality Reduction:

- **Curse of dimensionality**
  - When dimensionality increases, data becomes increasingly sparse in the space that it occupies
  - Definitions of density and distance between points, which is critical for clustering and outlier detection, become less meaningful



1-dimension 2-dimension 3-dimension

3.1 Dimensionality Reduction

- Purposes
  - To avoid curse of dimensionality
  - To reduce the amount of time and memory required by data mining algorithms
  - To allow data to be more easily visualized
  - Possibly, to help eliminate irrelevant features or reduce noise
- Techniques
  - Wavelet transform
  - Principal component analysis (PCA)
  - Feature selection
    - Removing Redundant features
    - Removing Irrelevant features

3.1 Dimensionality Reduction
Optimum method

- Recursive Feature elimination:
  - It is a greedy optimization algorithm which aims to find the best performing feature subset.
  - It repeatedly creates models and keeps aside the best or the worst performing feature at each iteration.
  - It constructs the next model with the remaining features until all the features are used.
  - It then ranks the features based on the order of their elimination.

3.1 Dimensionality Reduction
Heuristics in Feature Selection

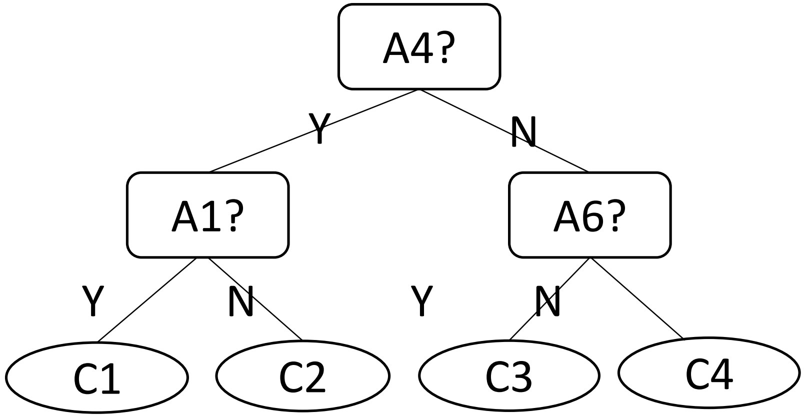
- Stepwise forward selection
  - The forward selection approach, also called the greedy approach
  - Repeatedly picking the best single-attribute
    - It starts with empty data set
    - Try adding each feature
    - Estimate classification/regression accuracy for adding each feature
    - Select features that give maximum improvement
    - Stop when there is no significant improvement

3.1 Dimensionality Reduction
Heuristics in Feature Selection

- Stepwise backward elimination
  - Gradually eliminating the features that affect the performance the least
  - Repeatedly eliminating the worst attribute
    - Starts with the full feature set
    - Try removing features
    - Remove the least significant feature at each iteration which improves the performance of the model
    - Stop when there is no significant improvement

- Decision tree induction
  - Using the attributes appeared in the decision tree
  - Attribute with the highest information gain are included

3.1 Dimensionality Reduction
Heuristics in Feature Selection

Forward selection	Backward elimination	Decision tree
Initial attribute set: {A1, A2, A3, A4, A5, A6} Initial reduced set: {} $\Rightarrow$ {A1} $\Rightarrow$ {A1, A4} $\Rightarrow$ Reduced attribute set: {A1, A4, A6}	Initial attribute set: {A1, A2, A3, A4, A5, A6} $\Rightarrow$ {A1, A3, A4, A5, A6} $\Rightarrow$ {A1, A4, A5, A6} $\Rightarrow$ Reduced attribute set: {A1, A4, A6}	Initial attribute set: {A1, A2, A3, A4, A5, A6}  $\Rightarrow$ Reduced attribute set: {A1, A4, A6}

3.2 Numerosity Reduction

- Reducing the data volume by choosing alternative, smaller forms of data representation
- Parametric methods (e.g., regression)
  - Assume the data fits in some model, estimate the model parameters, store only the parameters, and discard the data (except possible outliers)
  - e.g., linear regression ($y = mx + b$)
- Non-parametric methods
  - Do not assume models
  - Histograms, clustering, sampling

3.2 Numerosity Reduction

Regression Analysis

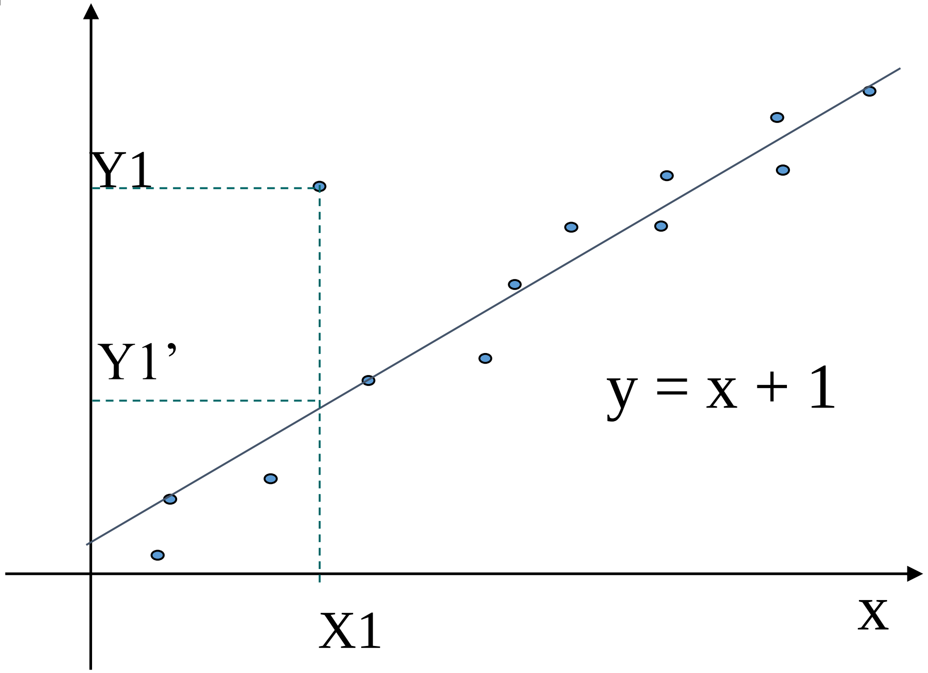
- Regression analysis includes any techniques for the modeling and analysis of numerical data consisting of values of a ***dependent*** variable and of one or more ***independent*** variables

- The parameters are estimated so as to give a "**best fit**" of the data

- Most commonly, the best fit is evaluated by using the **least squares method**, but other criteria have also been used

3.2 Numerosity Reduction

Linear Regression



- $y = mx + b$
  - Two regression coefficients, m and b, specify the line and are to be estimated by using the data at hand

3.2 Numerosity Reduction

Sampling

- Definition: selection of a **subset** of individual observations within a population of individuals intended to yield some knowledge about the population
- Key principle: choosing a **representative** subset of the data
  - Simple random sampling may have very poor performance in the presence of skewed data

3.2 Numerosity Reduction

Types of Sampling

- Simple random sampling
  - There is an equal probability of selecting any particular item
- Sampling without replacement
  - Once an object is selected, it is removed from the population
- Sampling with replacement
  - A selected object is not removed from the population
    - In sampling with replacement, the same object can be picked up more than once
- Stratified sampling
  - The population is divided into non-overlapping groups (i.e., strata)

3.2 Numerosity Reduction

Stratified Sampling

- In general, the size of the sample in each stratum is taken in proportion to the size of the stratum ← called **proportional allocation**
  - e.g., suppose that in a company there are staff members as below, and we are asked to sample **40** staffs

Male, full-time	90
Male, part-time	18
Female, full-time	9
Female, part-time	63
Total	180

= (90 x 40) / 180 = 20

= (18 x 40) / 180 = 4

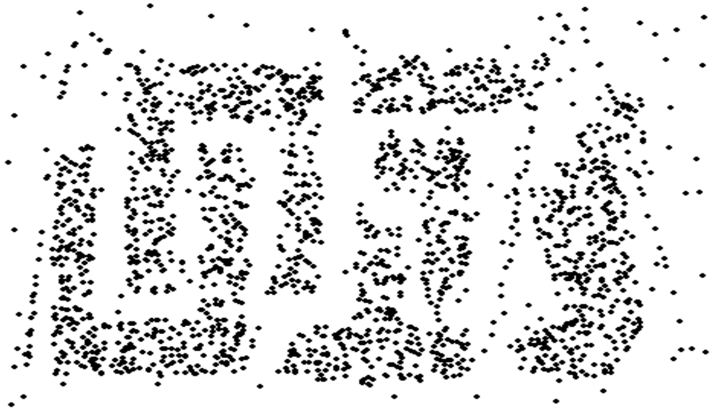
= (9 x 40) / 180 = 2

= (63 x 40) / 180 = 14

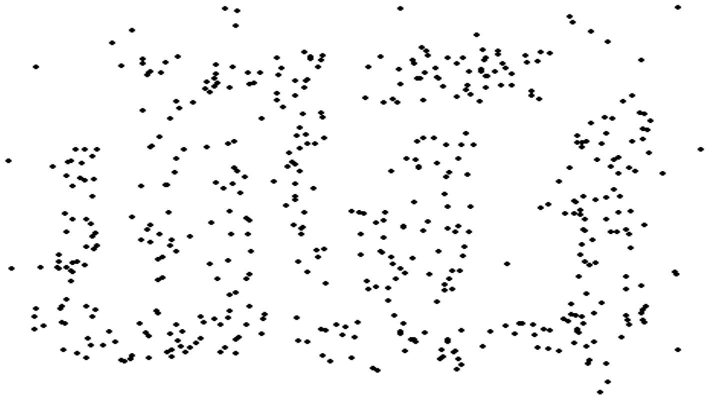
3.2 Numerosity Reduction

Sampling Size

8000 points



2000 Points



500 Points

3.2 Numerosity Reduction

Data Cube Aggregation

- The lowest level of a data cube (base cuboid)
  - The aggregated data for an individual entity of interest
  - e.g., the amount of sales per day

- Multiple levels of aggregation in data cubes
  - Further reducing the size of data to deal with
  - e.g., day → week → month → quarter → year

- Referencing appropriate levels
  - Using the smallest representation which is enough to solve the task

3.3 Data Compression

- Data compression is the process of encoding information using fewer bits than the original representation would use

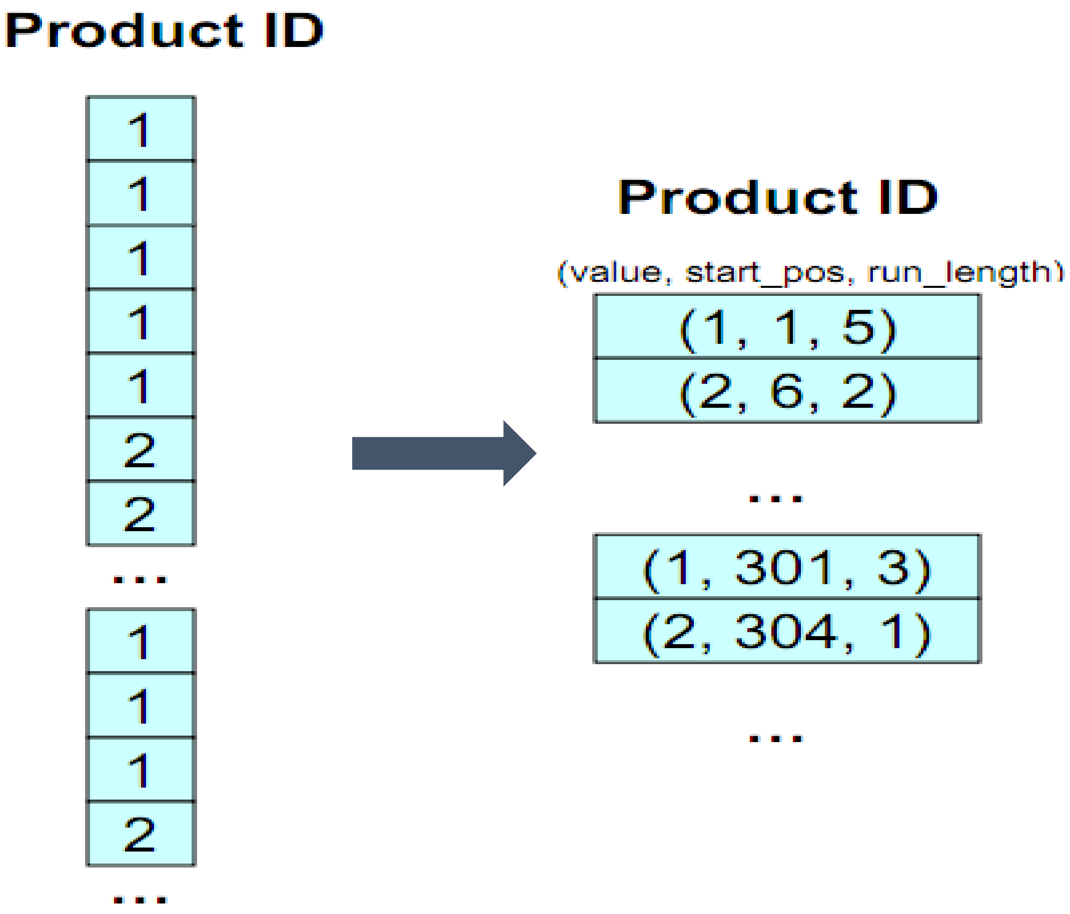
- It was originally developed for reducing the data size, but it is important also for **improving the query performance**
  - Operations can be performed ***directly*** on compressed data, and the amount of disk I/O's is much reduced

- Almost all data warehousing systems compress data when loading the data

3.3 Data Compression

Run-Length Encoding

- Runs of data (i.e., sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run



4. Data Transformation

- Definition
  - A function that maps the entire set of values of a given attribute to a new set of values such that each old value can be identified with one of the new values
- Methods
  - Normalization
    - Min-max normalization
    - Mean normalization
    - Standardization (Z-score normalization)
  - Discretization: Concept hierarchy climbing
    - A **concept hierarchy** defines a sequence of mappings from a set of low-level concepts to higher-level, more general concept.
    - i.e. replacing low level concepts (such as numerical values for age) by high level concepts (such as baby, child, teenager, young, adult, old)

4. Data Transformation

Normalization:

The goal of normalization is to change the values of numeric columns in the dataset to a common scale.

person_name	Salary	Year_of_experience	Expected Position Level
Aman	100000	10	2
Abhinav	78000	7	4
Ashutosh	32000	5	8
Dishi	55000	6	7
Abhishek	92000	8	3
Avantika	120000	15	1
Ayushi	65750	7	5

The attributes salary and year_of_experience are on different scale and hence attribute salary can take high priority over attribute year_of_experience in the model.

4. Data Transformation

Normalization:

- Min-max normalization [0, 1]

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

- Mean normalization [-0.5, 0.5]

$$x' = \frac{x - \text{average}(x)}{\max(x) - \min(x)}$$

- Standardization (Z-score normalization) [-3, 3]

$$x' = \frac{x - \text{average}(x)}{\sigma}$$

4. Data Transformation
Data Discretization

- Definition
  - Reducing the number of values for a given continuous attribute by dividing the range of the attribute into intervals and replacing actual data values with the interval labels
- Purposes
  - To find informative cut-off points in the data
  - To enable the use of some learning algorithms
  - To reduce the data size