HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

IT4142E

Introduction to Data Science

Chapter 3 – part 2 : Data cleaning and pre-processing

Lecturer:

Muriel VISANI: murielv@soict.hust.edu.vn

Acknowledgements:

Khoat Than
Viet-Trung Tran

Department of Information Systems
School of Information and Communication Technology - HUST

Contents of the course

- Chapter 1: Overview
- Chapter 2: Data scraping
- Chapter 3: Data cleaning, pre-processing and integration
- Chapter 4: Introduction to Exploratory Data Analysis
- Chapter 5: Introduction to Data visualization
- Chapter 6: Introduction to Machine Learning
  - Performance evaluation
- Chapter 7: Introduction to Big Data Analysis
- Chapter 8: Applications to Image and Video Analysis

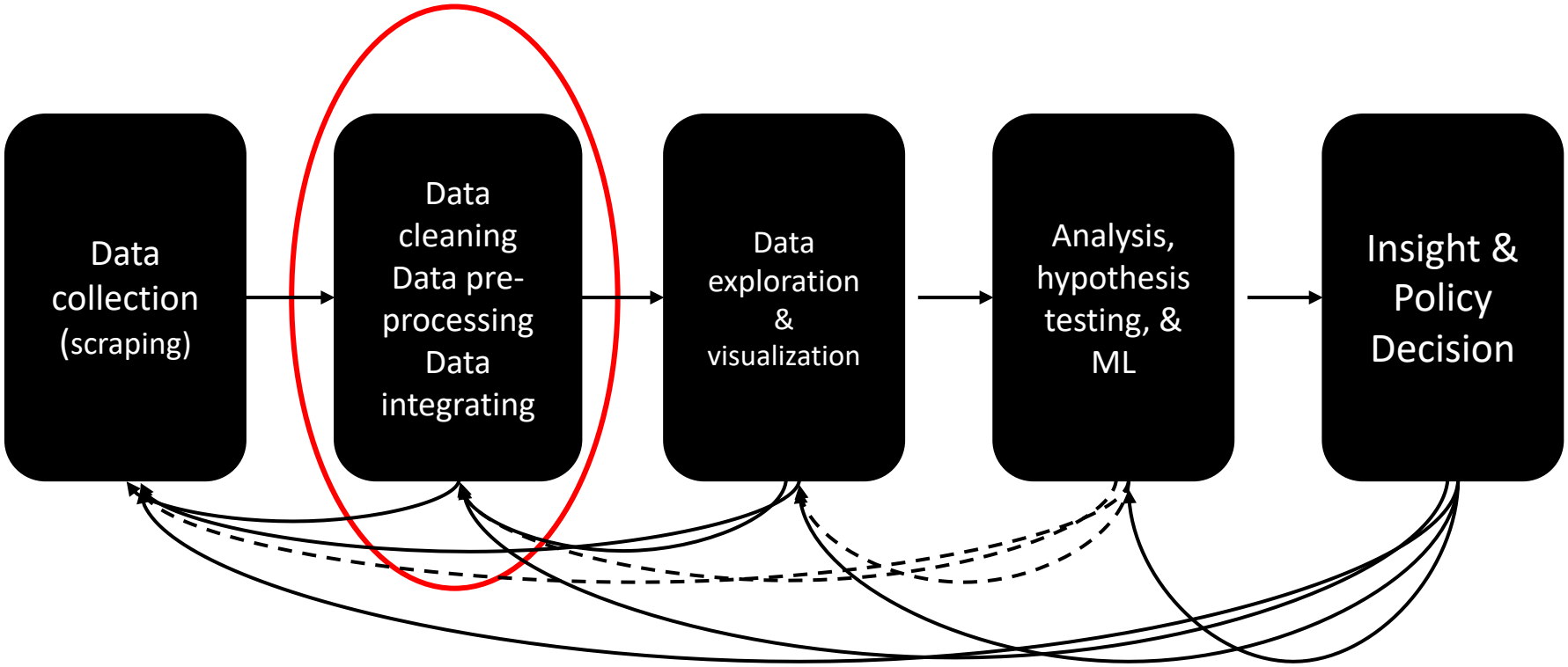
Outline

- **Chapter 3 – part 2: Data cleaning and processing**
  - Data cleaning
    - Introduction
    - Different types of data quality problems
    - How to clean the noisy / dirty data?
  - Data pre-processing
  - Technical solutions for data cleaning and pre-processing
  - Homework
  - Summary

Goals of this chapter

Goal	Description of the goal
M1	**Understand and be able to design and manage the systems which are based on Data Science (DS)**
M1.1	Identify and understand the components of the systems based on DS
M1.2	Identify, compare, and categorize the data types and systems in practice
M1.3	Be able to design systems based on DS in their future organizations

Recall: insight-driven DS methodology



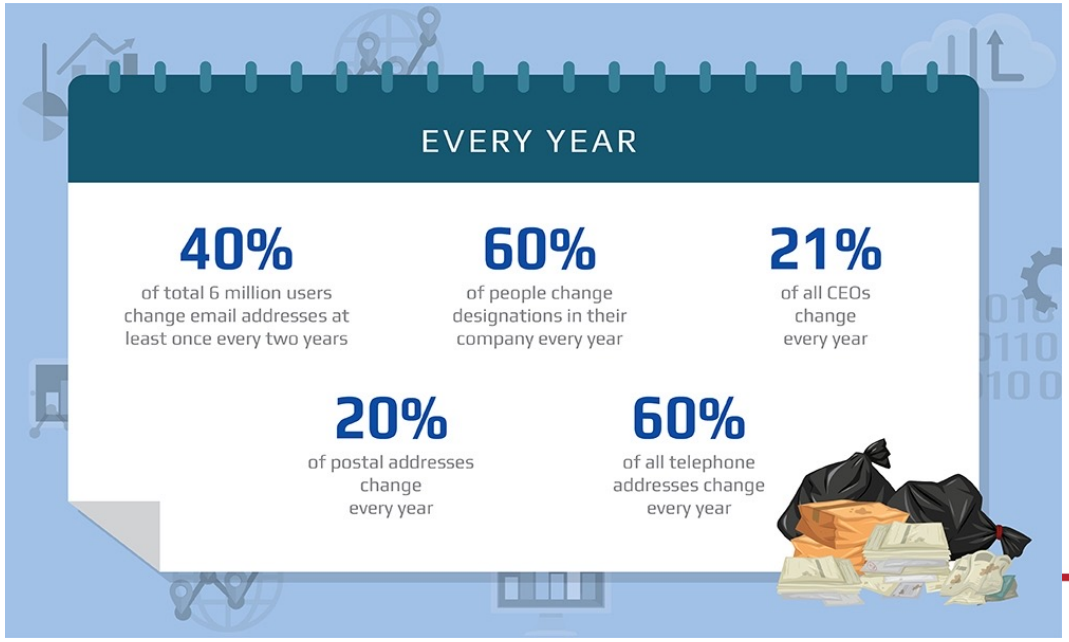
- Data cleaning and pre-processing must be done **before** data integration
- But, you need to understand data integration before cleaning or processing the data

Data cleaning

Introduction

Why data cleaning?

- Data in real world is **dirty**
  - Incomplete
  - Noisy
  - Inconsistent
  - Duplicate records
- Why is data dirty?



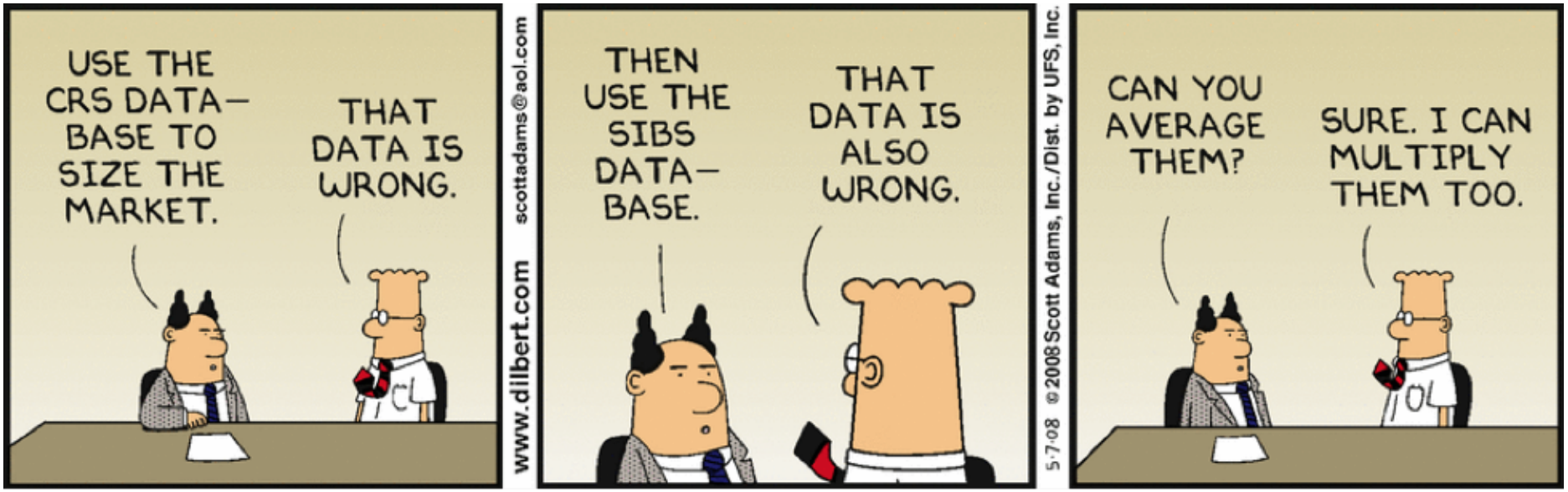
[Source: https://datafloq.com]

Why is data dirty?

- Data in real world is dirty
  - **Incomplete** (e.g name = “”)
    - Lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - Different considerations between the time when the data was collected and when it is analyzed
    - Human/hardware/software bugs
  - **Noisy** (e.g. salary =‘-10k’)
    - Containing errors or outliers
    - Faulty data collection instruments
    - Human error at data entry
    - Error in data transmission
  - **Inconsistent** (e.g., Age=“20” Birthday=“02/02/2000”)
    - Different data sources
    - Functional dependency violation
  - **Redundant**

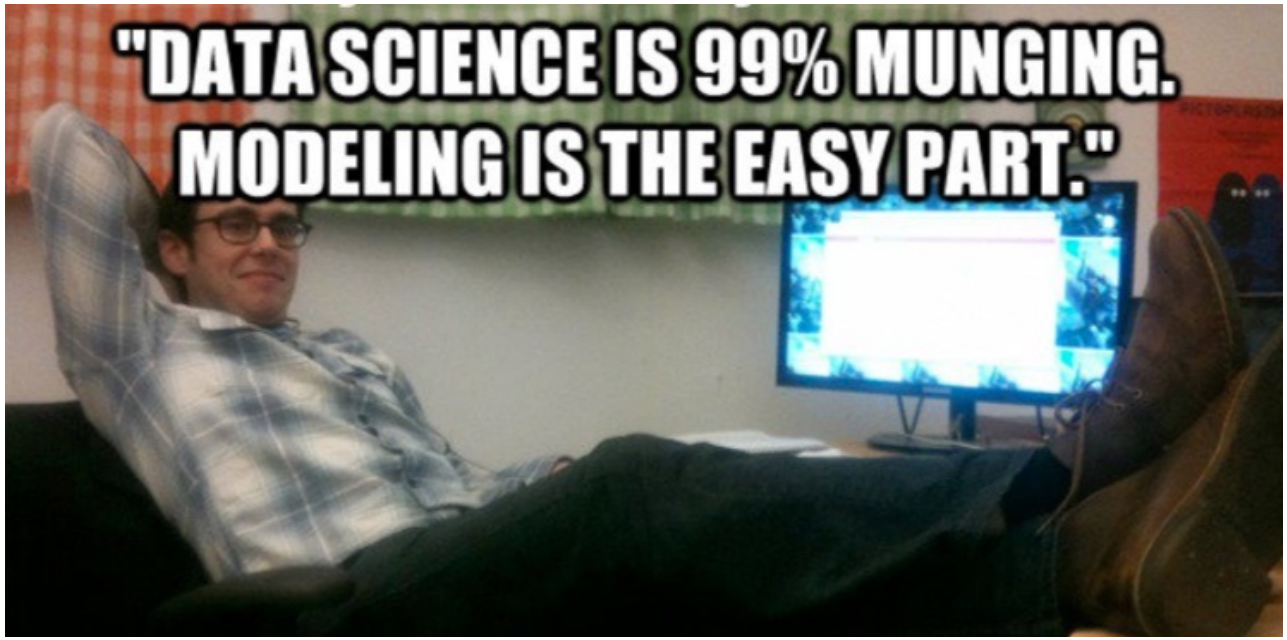
No quality data, no quality DS

- Quality decisions must be based on quality data
- Dirty data may cause incorrect / misleading statistics
  - Some consider that, in the US, up to 40% of business objectives that failed, failed because of the use of dirty data



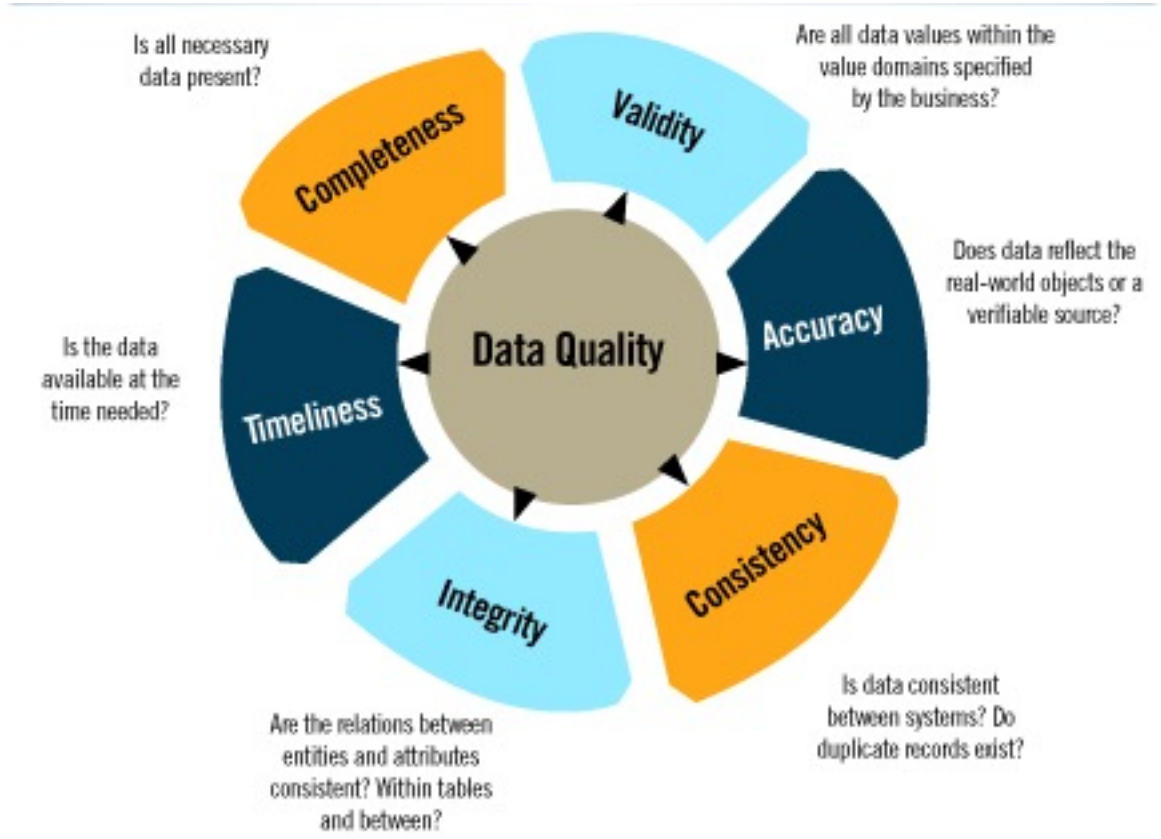
Data preprocessing is costly

- Data extraction, cleaning, and transformation comprises the majority of the work of building a data warehouse

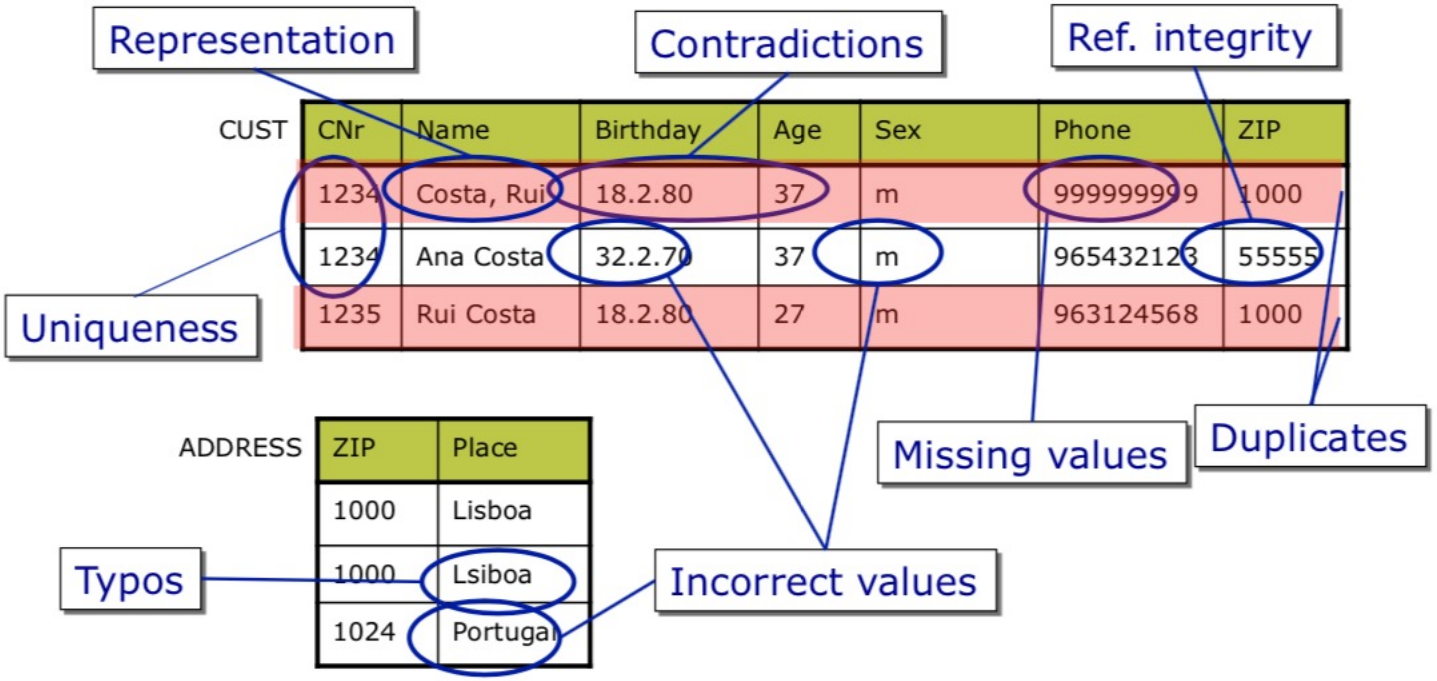


Data quality dimensions

- “Even though quality cannot be defined, you know what it is.” Robert Pirsig



Data quality problems



Data cleaning

Different types of data quality problems

Taxonomy of data quality problems

- Value-level
  - At the level of a cell in the data table
- Attribute (value-set) level
  - At the level of a column in the data table
- Record level
  - At the level of a row in the data table
- Table (relation) level
  - At the level of the whole data table
- Multiple tables level
  - At the level of multiple data tables

Value level

- Missing value: value not filled
  - Ex: birthdate=‘’
- Syntax violation: value does not satisfy the syntax rule defined for the attribute
  - Ex: zipcode=27655-175; syntactical rule: xxxx-xxx
- Spelling error
  - Ex: city=‘Hnoi’, instead of ‘Hanoi’
- Domain violation: value does not belong to the valid domain set
  - Ex: age=240; age:{0,120}
- …

Attribute and record levels

- Attribute (value-set) level
  - Existence of synonyms: attribute takes different values, but with the same meaning
    - Ex: Cold: chilly, freezing, frosty.
  - Existence of homonyms: same word used with diff meanings
    - Ex: Right = correct OR direction opposite of left
  - Uniqueness violation: unique attribute takes the same value more than once
    - Ex: two clients have the same ID number
  - Integrity constraint violation (on the column)
    - Ex: sum of the values of a percentage attribute among all records is more than 100
      - Example: in the capstone project, for one task, the sum of the % of work of all students (records) involved should be 100%
- Record level
  - Integrity constraint violation (on the row)
    - Ex: total price of a product is different from price plus taxes

Table (relation) level

- Heterogeneous data representations: different ways of representing the same real world entity
  - Ex: name = ‘John Smith’; name = ‘Smith, John’
- Functional dependency violation
  - Ex: (2765-175, ‘Estoril’) and (2765-175, ‘Oeiras’)
- Existence of **approximate** duplicates
  - Ex: (1, André Fialho, 12634268) and (2, André Pereira Fialho, 12634268)!
- Integrity constraint violation (on the table)
  - Ex: sum of all employees salaries + bonuses is superior to the max budget established

Multiple tables level

- Heterogeneous data representations
  - Ex: one table stores meters, another stores inches
- Existence of synonyms
- Existence of homonyms
- Different granularities: same real world entity represented with diff. granularity levels
  - Ex: age:{0-30,31-60,>60};age:{0-25,26-40, 40-65, >65}
- Referential integrity violation
- Existence of duplicates, or approximate duplicates
- Integrity constraint violation

Data cleaning

How to clean the noisy / dirty data?

Methodology for data cleaning

- Extraction of the individual attributes that are relevant
- Correction of data quality problems at the value level
  - Missing values, syntax violation, etc
- Correction of data quality problems at the attribute (value-set) level and record level
  - Synonyms, homonyms, uniqueness violation, integrity constraint violation, etc
- Correction of data quality problems at relation level
  - Violation of functional dependencies (inconsistent data), redundant data, etc
- Correction of data quality problems problems at multiple relations level
  - Referential integrity violation (inconsistent data), redundant data, etc
- User feedback
  - To solve instances of data quality problems that cannot be addressed by automatic methods
- Effectiveness of the data cleaning and transformation process must be always measured for a sample of the data set

Focus on 4 data cleaning tasks

- 4 important tasks of Data cleaning
  1. Manage missing data
     - Either ignore it, OR…
     - … Fill in the missing values
       - Value-level problem

  2. Manage noisy data
     - Identify noisy data
     - Correct noisy data
       - Value-level or record-level problem

  3. Manage redundant data
     - Identify redundant data
       - Record-level problem, attribute-level problem, table-level problem, or multi-table level tables
     - Remove redundancy

  4. Manage inconsistent data
     - Identify inconsistent data
       - Record-level problem, attribute-level problem, table-level problem, or multi-table level tables
     - Correct inconsistent data

1. Manage missing data

- Different possible ways to manage missing data
  - Ignore the tuple: usually done when class label is missing (assuming the task is classification—not effective in certain cases)
  - Fill in the missing value manually: tedious + infeasible?
    - Use a global constant to fill in the missing value: e.g., “unknown”, a new class?!
    - Use the attribute mean to fill in the missing value
    - Use the attribute mean for all samples of the same class to fill in the missing value: smarter
    - Use the most probable value to fill in the missing value:modal value or inference based on linear regression, Bayes formula, regression tree, etc…
    - Often, we manage missing data based on a semi-automatic procedure
      - Computer + manual intervention

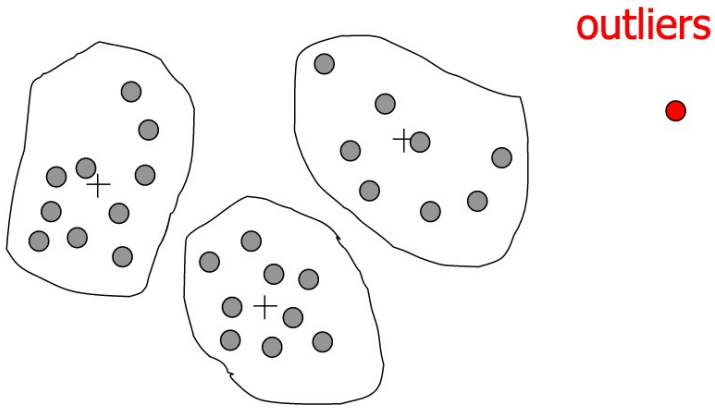
2. Manage noisy data – Identification

- Noisy data are mostly constraint violations and outliers
  - Noisy data can be at the value level (*e.g.* univariate outliers), or they can be at the record level (*e.g.* multivariate outliers)

- Example of very simple methods for detecting outliers:
  - https://towardsdatascience.com/ways-to-detect-and-remove-the-outliers-404d16608dba

- Managing outliers thanks to clustering:
  - Some clustering methods can be used to simultaneously detect and remove outliers (see chapter 6)

- Once noisy data is detected, it can be handled similarly to missing data, or using specific methods (see next slide)



2. Manage noisy data – Correction

- On top of the same methods for handling missing values:
  - Binning method for data smoothing: effective when there are too many variations due to noise in the attribute's values
    - First, sort data and partition into (equi-depth) bins
    - Then, one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
    - https://www.geeksforgeeks.org/python-binning-method-for-data-smoothing/#:~:text=Binning%20method%20is%20used%20to,values%2C%20they%20perform%20local%20smoothing.
    - Example:
      - Sorted data for price (in dollars): 4, 8, 9, 12, 15, 21, 21, 21, 24, 25, 26, 26, 28, 29, 34
      - Partition into (equi-depth) bins: (size = 5)
        - 4, 8, 9, 12, 15, 21, 21, 21, 24, 25, 26, 26, 28, 29, 34
      - Smoothing by bin median:
        - Values in bin 1 -> 9, 9, 9, 9, 9
        - Values in bin 2 -> 21, 21, 21, 21, 21
        - Values in bin 3 -> 28, 28, 28, 28, 28

3. Manage redundant data

- Redundant data can be problems at the record-level, at the attribute-level, at the table-level, or at the multi-table level
- Redundancy often occurs when integrating multiple databases
- Avoiding redundancies will improve data mining speed
- Avoiding redundancies will improve data mining quality (depending on the methods used)
  - Some methods for data mining are very effective, yet very sensitive to redundancy
    - For instance, SVM with RBF kernel and Euclidean distance

3. Manage redundant data

- Possible solutions to detect redundant data:
  - Usually using semi-automatic tools
    - Difficulties
      - 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 from monthly revenue
    - Possible solution:
      - Redundant attributes may be detected by correlation analysis (see chapter 4)
    - To correct redundant data, we ignore some attributes in order to get rid of redundancy

4. Manage inconsistent data

- Inconsistent data can be problems at the record-level, at the attribute-level, at the table-level, or at the multi-table level
- Inconsistency often occurs when integrating multiple databases
- Avoiding inconsistencies will obviously improve the quality of data mining…
- … and decision-making!

4. Manage inconsistent data

- Possible solutions
  - Manual correction using external references
  - Semi-automatic correction, using various tools
    - To detect violation of known functional dependencies and data constraints
    - To correct inconsistent data, we replace it by the values from the most reliable source

Data pre-processing

Data pre-processing

- Some of the tasks explained previously can be considered either as data cleaning, or data pre-processing
  - *e.g.* correcting redundant or noisy data
  - It does not really matter… as long as you don't forget to do it!

- Some important steps of data pre-processing
  - Data wrapping: https://en.wikipedia.org/wiki/Wrapper_(data_mining)
  - Data normalization: scaled to fall within a small, specified range
    - min-max normalization
    - z-score normalization *a.k.a.* standardization)
    - normalization by decimal scaling
  - Data reduction: to obtain a reduced representation of the data
  - Data aggregation: summarization
    - *E.g*. average, median, etc.
  - Computing the derived attributes that you need
  - …

Data normalization

- Min-max normalization: transformation maps the values of a variable to a new range [NewMin, NewMax]

$$x'_i = \frac{x_i - OriginalMin}{OriginalMax - OriginalMin} \times (NewMax - NewMin) + NewMin$$

- Z-score normalization: (μ_A: mean, σ_A : standard deviation of the attribute A to normalize):

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

- Normalization by decimal scaling: ensures the range is between -1 and 1
  - with n the number of digits of the maximum absolute value:

$$x'_i = \frac{x_i}{10^n}$$

Data reduction

- Goal: obtain a reduced representation of the data
  - much smaller in volume yet producing the same (or almost the same) data mining results

- Dimensionality reduction
  - Feature selection (*e.g.* SFFS, SFBS, genetic algorithm)
  - Feature extraction (*e.g.* PCA analysis)

- Data Compression
  - Convert text to numbers
  - Data clustering

- Discretization
  - Convert continuous data to categories
  - *E.g.* age=1 if in [0,12]; 2 if in [12-25], etc…

Technical solutions for data cleaning and pre-processing

Many tools for pre-processing

- Open Refine
- Trifacta Wrangler
- Python libraries

My personal suggestions (all free)

Open Refine and Trifacta Wrangler are more user-friendly

With Python libraries, you are more in control

- Tableau (and its free version Tableau Public)
- TabDrake TIBCO
- Clarity
- Winpure Data
- LadderData
- Cleaner
- Cloudingo
- Reifier
- IBM Infosphere Quality Stage

Demo on OpenRefine

- OpenRefine was formerly Google Refine
- It is a desktop application, not a web-service
  - Your sensitive data is (supposedly) safe
- Demo
  - https://youtu.be/B70J_H_zAWM
- More information on
  - https://guides.library.illinois.edu/openrefine

Homework

Main homework

- Practical using Python libraries
- 5-days challenge: https://www.kaggle.com/rtatman/data-cleaning-challenge-handling-missing-values
    - Day 1: Handling missing values
    - Day 2: Scaling and normalization
    - Day 3: Parsing dates
    - Day 4: Character encodings
    - Day 5: Inconsistent Data Entry

- Libraries used:
    - Pandas
    - Numpy
    - Chardet
    - Datetime
    - Fuzzywuzzy
    - Seaborn
    - Scipy
    - Mlxtend.preprocessing
    - Matplotlib.pyplot

Other homework for this lecture

- Data cleaning
  - Learn some simple methods to remove outlier data
  - https://towardsdatascience.com/ways-to-detect-and-remove-the-outliers-404d16608dba
- Data pre-processing
  - Learn more about OpenRefine on
  - https://guides.library.illinois.edu/openrefine

Summary

Summary

- You have learnt about
  - Data cleaning
  - Data pre-processing
- You have had demos / tutorials on
  - Openrefine
- You will realize homework using Python libraries
  - This will help you for the capstone project (especially for topic 1 on EDA, but also possibly for topic 2 on ML)

Questions

ĐẠI HỌC BÁCH KHOA

25 YEARS ANNIVERSARY SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attention!!!