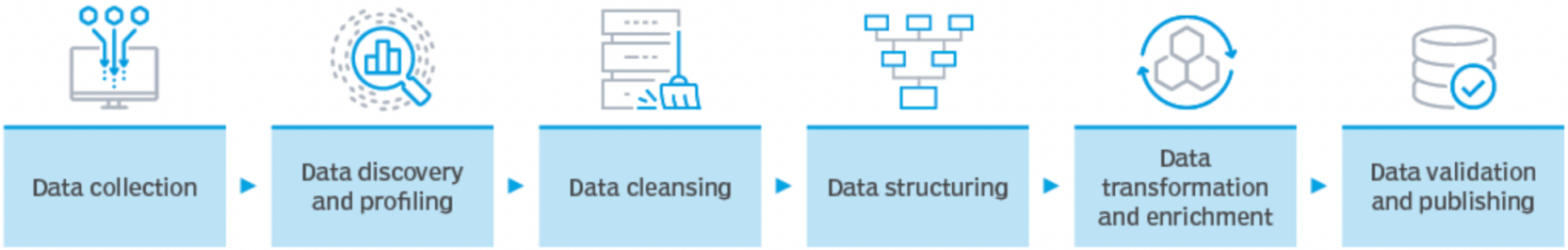
CHAPTER 2: DATA PREPARATION

Introduction to Data Science

Posts and Telecommunications Institute of Technology

Ha Noi, 2024

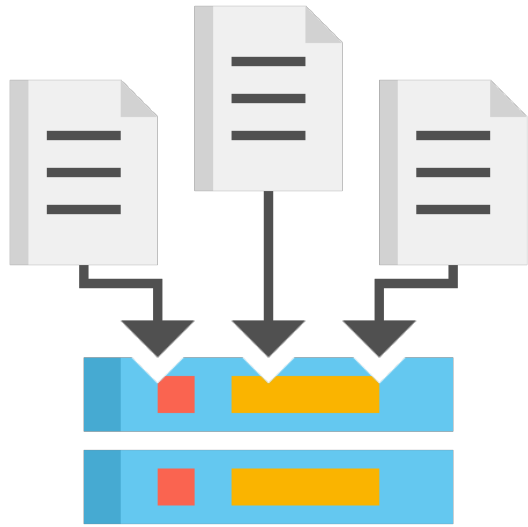
What is Data Preparation?



Data preparation is the process of preparing raw data so that it is suitable for further processing and analysis. Key steps include collecting, cleaning, validation, transformation and labeling raw data into a form suitable for machine learning (ML) algorithms and then exploring and visualizing the data. Data preparation can take up to 80% of the time spent on an ML project.

CONTENTS

Data Collection

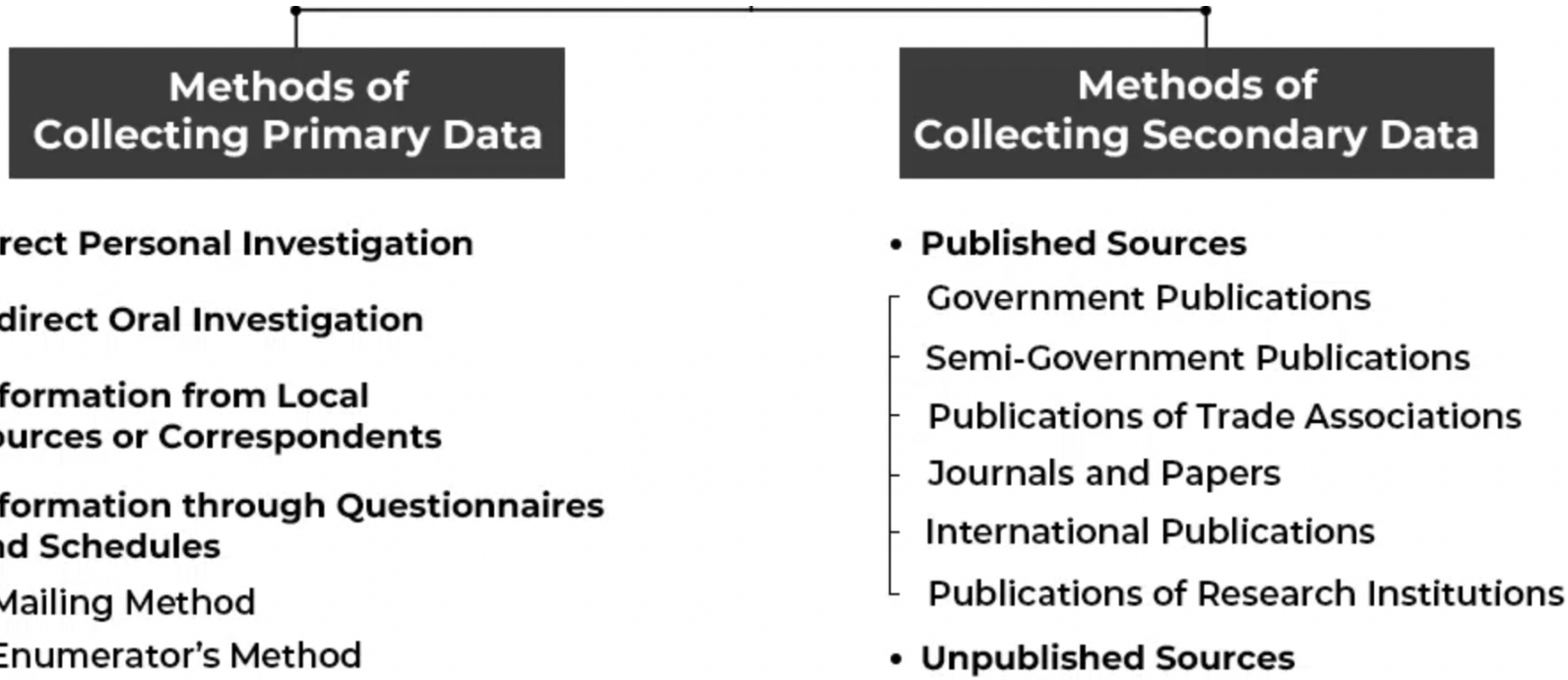


- *Collecting data* is the process of assembling all the data you need from relevant sources.
- Data resides in many data sources and has vastly different formats such as images, text, video, audio, data from social networks, online websites or other data sources.

Data Collection

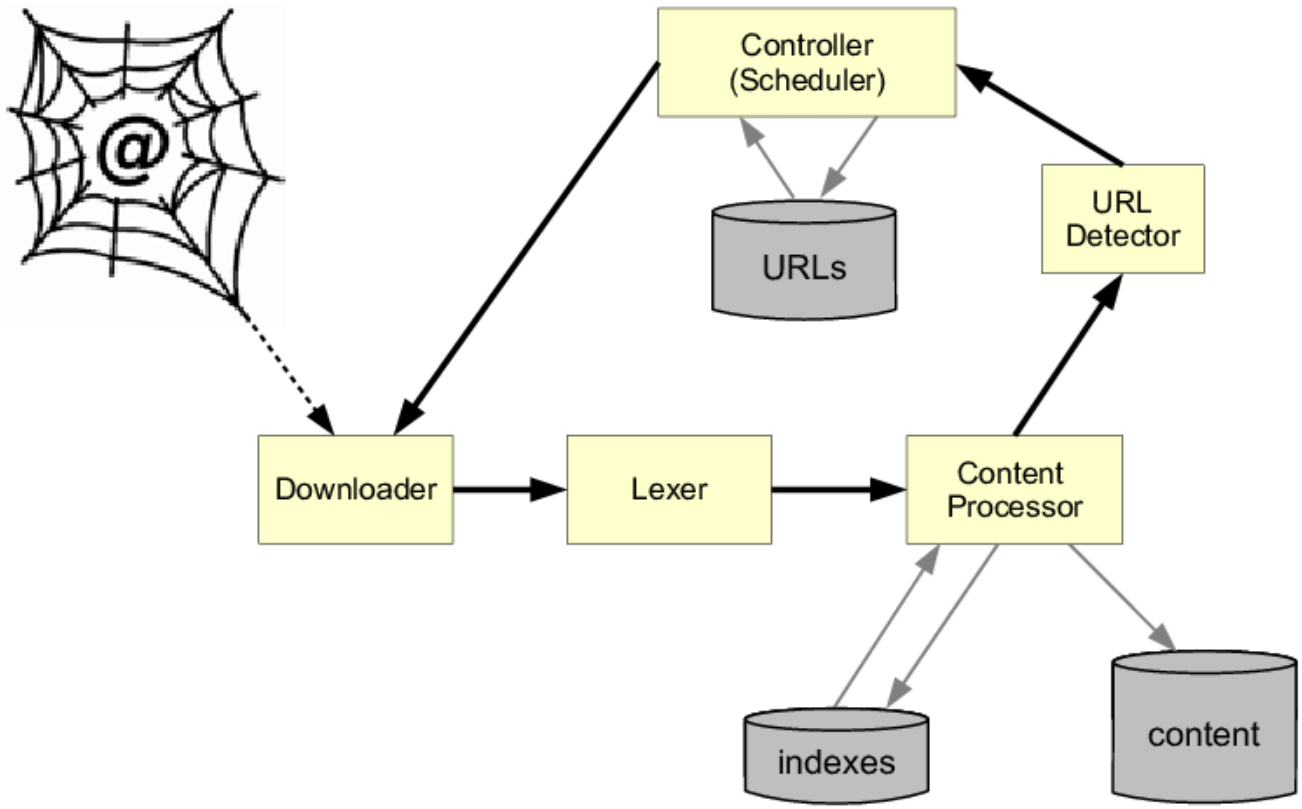
There are two different methods of collecting data:

- Primary Data Collection
- Secondary Data Collection.



Web crawler

Web crawlers are also referred to as *spiders* or *robots*, the resources on the Web are dispensed widely across globally distributed sites.



Web crawler

There are two primary types of data available on the Web that are used by mining algorithms :

- *Web content information*: This information corresponds to the Web documents and links created by users:
  - *Document data*: The document data are extracted from the pages on the World Wide Web.
  - *Linkage data*: The Web can be viewed as a massive graph, in which the pages correspond to nodes, and the linkages correspond to edges between nodes.
- *Web usage data*: This data corresponds to the patterns of user activity that are enabled by Web applications.
  - *Web transactions, ratings, and user feedback.*
  - *Web logs*: User browsing behavior is captured in the form of Web logs that are typically maintained at most Web sites.

Web crawler applications

The applications on the Web are either content- or usage-centric:

- *Content-centric applications*: search, clustering, and classification:
  - *Data mining applications*.
  - *Web crawling and resource discovery.*
  - *Web search*.
  - *Web linkage mining.*
- *Usage-centric applications*: The user activity on the Web is mined to make inferences:
  - *Recommender systems.*
  - *Web log analysis.*

A Basic Crawler Algorithm

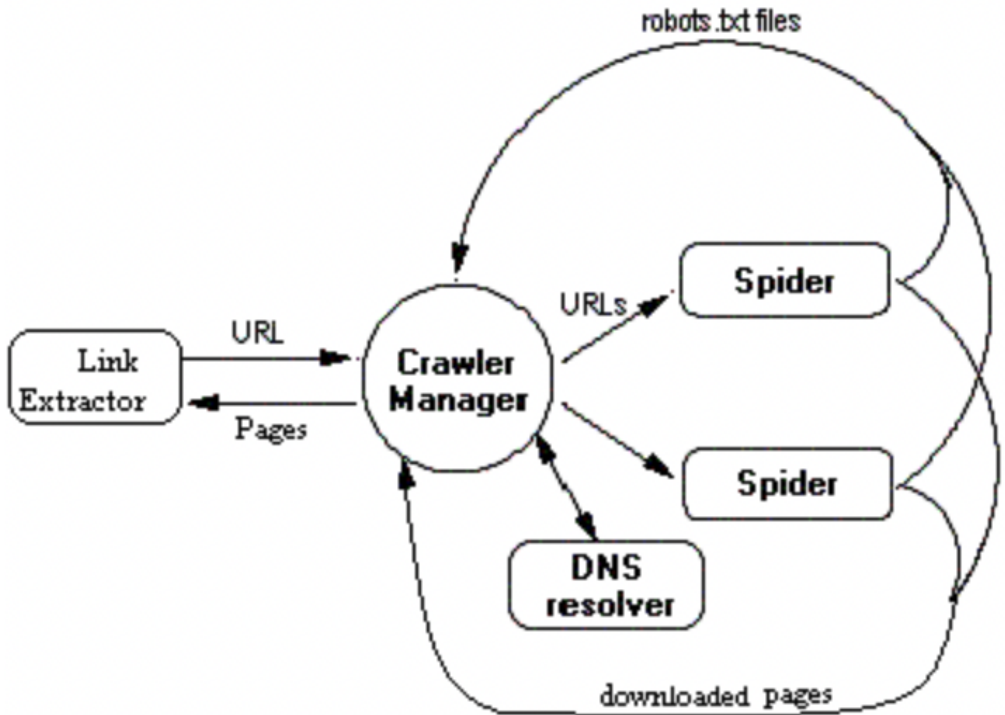
The basic crawler algorithm, described in a very general way, uses a seed set of Universal Resource Locators (URLs) S, and a selection algorithm A as the input. The algorithm A decides which document to crawl next from a current *frontier list* of URLs.

Algorithm *BasicCrawler*(Seed URLs: S, Selection Algorithm: $\mathcal{A}$)
begin
  $FrontierList = S$;
  repeat
    Use algorithm $\mathcal{A}$ to select URL $X \in FrontierSet$;
    $FrontierList = FrontierList - \{X\}$;
    Fetch URL X and add to repository;
    Add all relevant URLs in fetched document X to
      end of $FrontierList$;
  until termination criterion;
end

Aspects of web crawling

Aspects of web crawling include:

- Preferential Crawlers
- Multiple Threads
- Combatting Spider Traps
- Shingling for Near Duplicate Detection



Example: Apply data web crawler

The initial step of getting data from websites involves fetching the web pages and getting meaningful structured information:

Parse HTML:

```
<html>
    <head>
        <title>A web page</title>
    </head>
    <body>
        <p id="author">Joel Grus</p>
        <p id="subject">Data Science</p>
    </body>
</html>
```

Example: Apply data web crawler

To get data from HTML, you can use the *Beautiful Soup* library which is a Python library used to parse HTML and XML, making it easier to extract information from web pages.

```
from bs4 import BeautifulSoup
import requests

# I put the relevant HTML file on GitHub. In order to fit
# the URL in the book I had to split it across two lines.
# Recall that whitespace-separated strings get concatenated.

url = ("https://raw.githubusercontent.com/"
       "joelgrus/data/master/getting-data.html")
html = requests.get(url).text
soup = BeautifulSoup(html, 'html5lib')
```

The collection will work with objects corresponding to *tags* (Tag objects) that represent the structure of an HTML page.

```
first_paragraph = soup.find('p')    # hoac chi can su dung soup
```


Example: Apply data web crawler

Tracking the U.S. Congress Problem: requires quantifying information about the National Assembly, in particular, finding all representatives who have press releases with information about the National Assembly. A page with links to all websites of US Congress representatives and delegates at h*ttps://www.house.gov/representatives* and all links to websites.
The first step is to collect all URL links from this page as follows:

```
url = "https://www.house.gov/representatives"
text = requests.get(url).text
soup = BeautifulSoup(text, "html5lib")

all_urls = [a['href']
            for a on soup('a')
            if a.has_attr('href')]

print(len(all_urls)) # Ket qua tra ve 965
```

Example: Apply data web crawler

Tracking the U.S. Congress Problem
Next, perform the following collection operations:

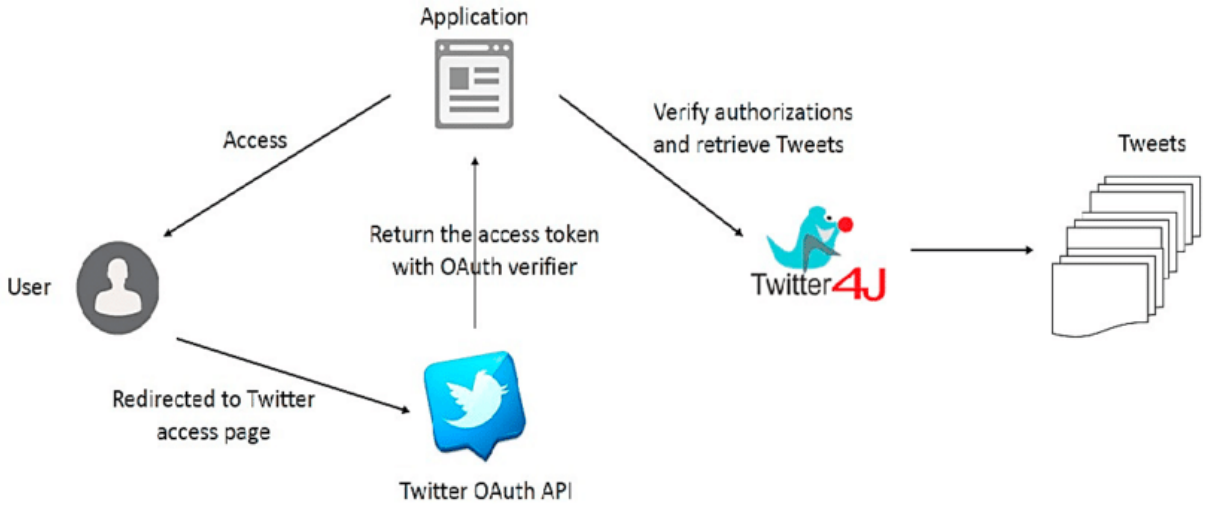
```
from typing import Dict, Set

press_releases: Dict[str, Set[str]] = {}

for house_url in good_urls:
        html = requests.get(house_url).text
        soup = BeautifulSoup(html, 'html5lib')
        pr_links = {a['href'] for a in soup('a') if 'press releases'
                                                    in a.text.lower()}
        print(f"{house_url}:{pr_links}")
        press_releases[house_url] = pr_links
```

Example: Apply data web crawler

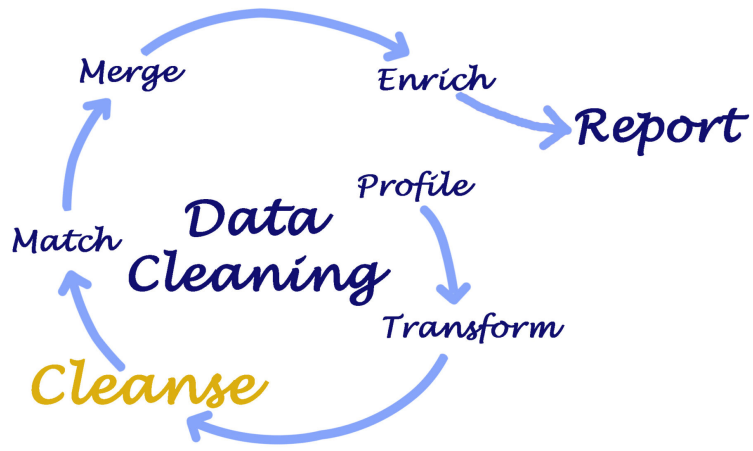
Using APIs: Read examples of Using Twitter APIs.



Note: Collecting data requires long-term storage instead of just saving it while running in the *list* variable, so it is necessary to store this data into a file or some database to be able to use the data.

Data Cleaning

Data cleaning is important because the data collection process often contains errors. There are several causes of missing values or errors during data collection.



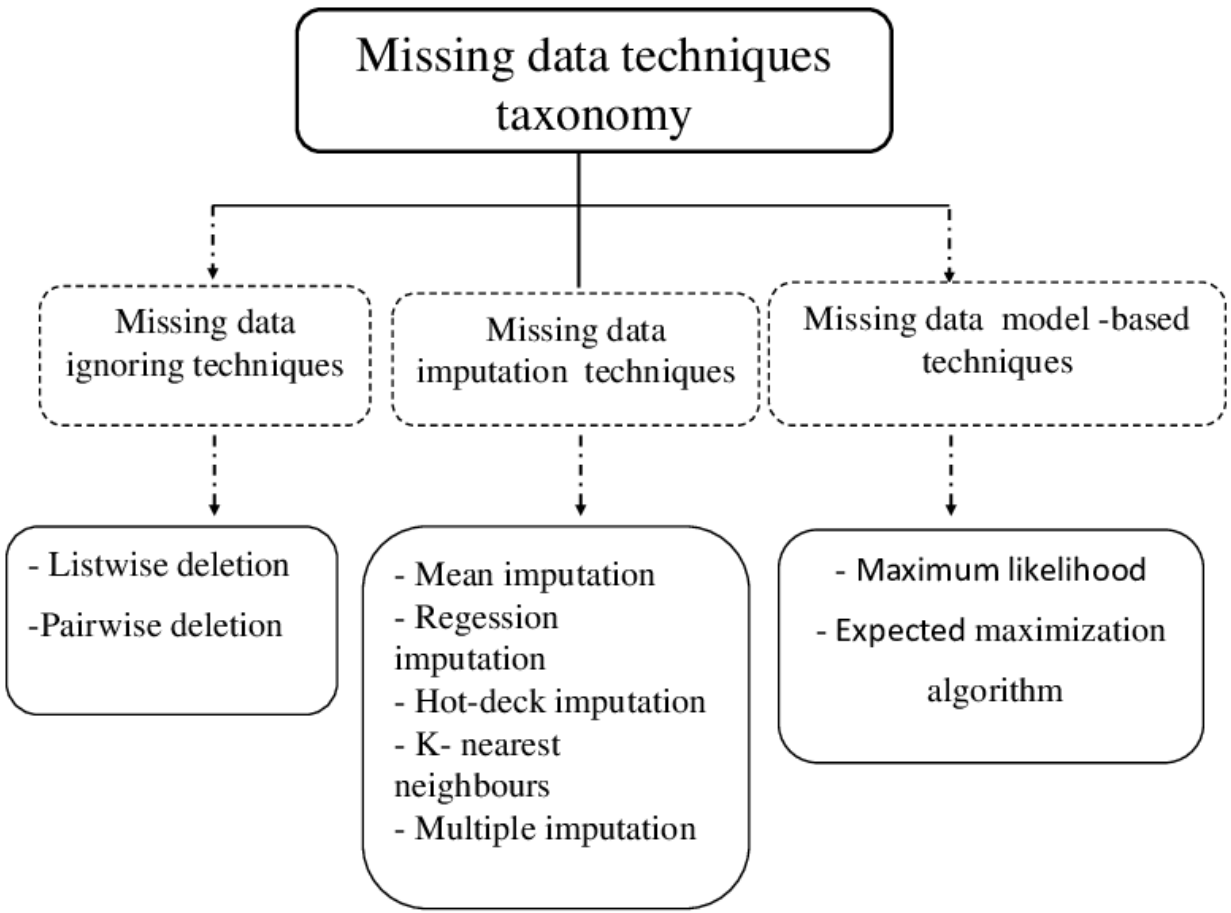
Data Clearning

Important aspects of data cleansing:

- *Handling Missing Values:*
  - Many values in the data may be unknown and missing.
  - The process of estimating missing values is also called *imputation*.
- *Handling Incorrect Values:*
  - In cases where information comes from multiple sources, it may not be consistent.
  - Eliminating contradictions is part of the analysis process.
  - Data points that do not conform to the distribution of the remaining data are often noisy as outliers

Handling Missing Values

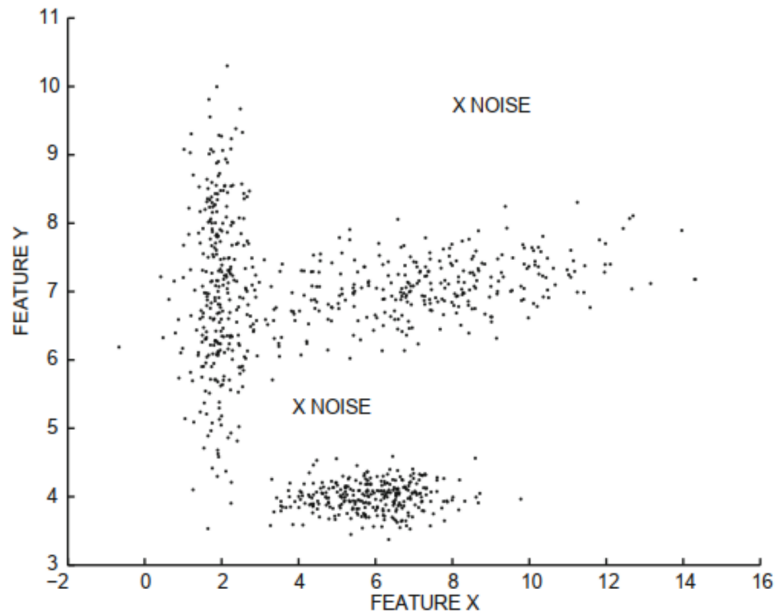
Three techniques for handling missing values:



Handling Incorrect Values

Some of the main methods used to handle incorrect or inconsistent values are as follows:

- Detect data inconsistencies
- Domain knowledge
- Data-centric approach.



Data normalization

- Scaling changes the range of the data
- Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. It is also known as Min-Max scaling.

Standardization:

$$x' = \frac{x - \bar{x}}{\sigma}$$

Mean Normalization:

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

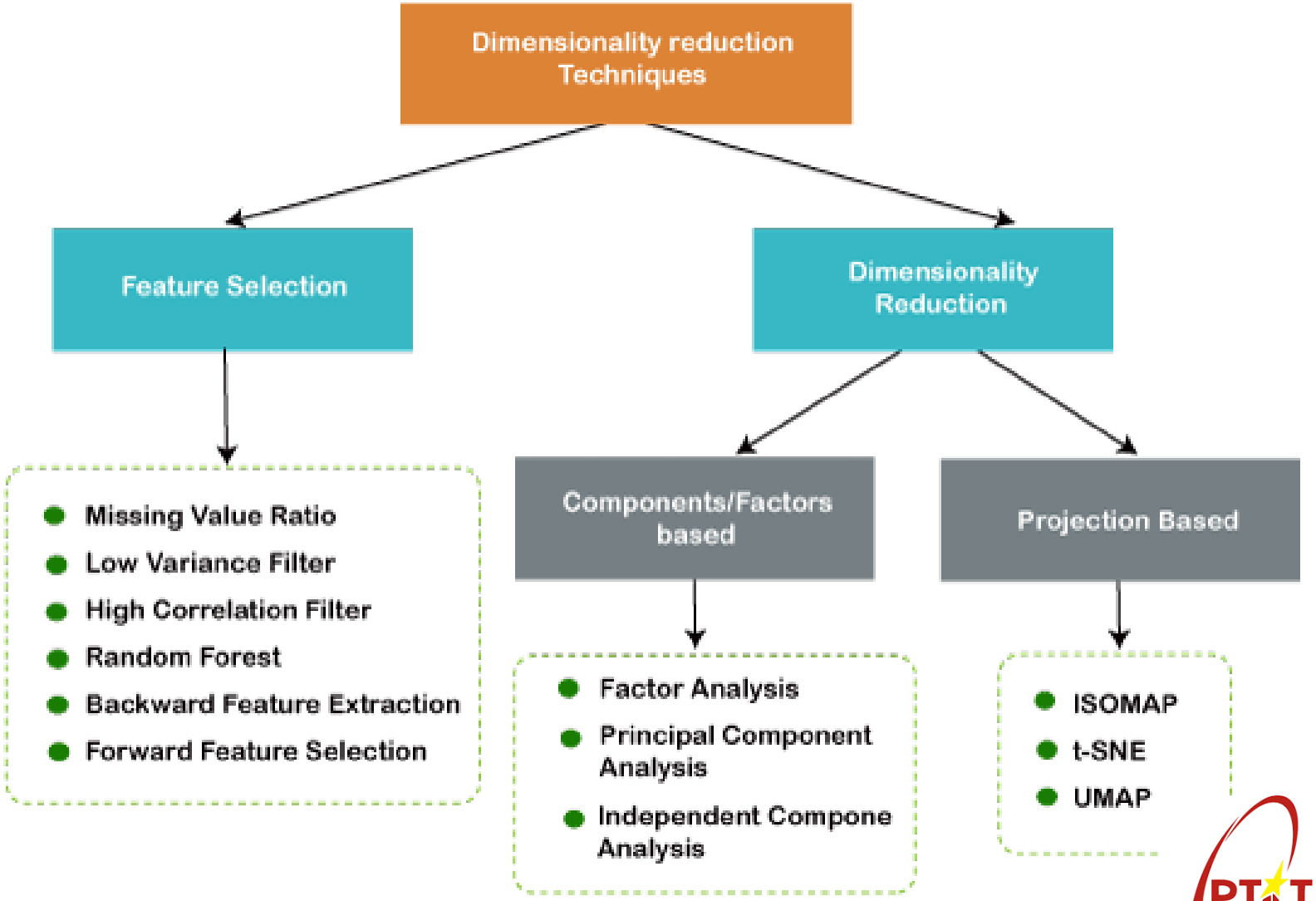
Min-Max Scaling:

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

Dimensionality Reduction and Data Transformation

- The purpose is to represent data more compactly.
- Applying complex and computationally expensive algorithms is much easier.
- Reducing data dimensionality can be reduced in the number of rows (number of records) or in the number of columns (number of dimensions).
- Reducing data dimensionality leads to information loss.
- Various types of data dimensionality reduction are used in different applications:
  - Sampling data
  - Featured selection
  - Reduce data dimensionality
  - Transform data

Dimensionality Reduction techniques



Data Transformation