DECISION SUPPORT SYSTEMS

Time series in machine learning: time series clustering

Part 2

Co-funded by the
Erasmus+ Programme
of the European Union

The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein

Author: Łukasz Paśko | Academic year: 2021/2022

Objectives of the classes

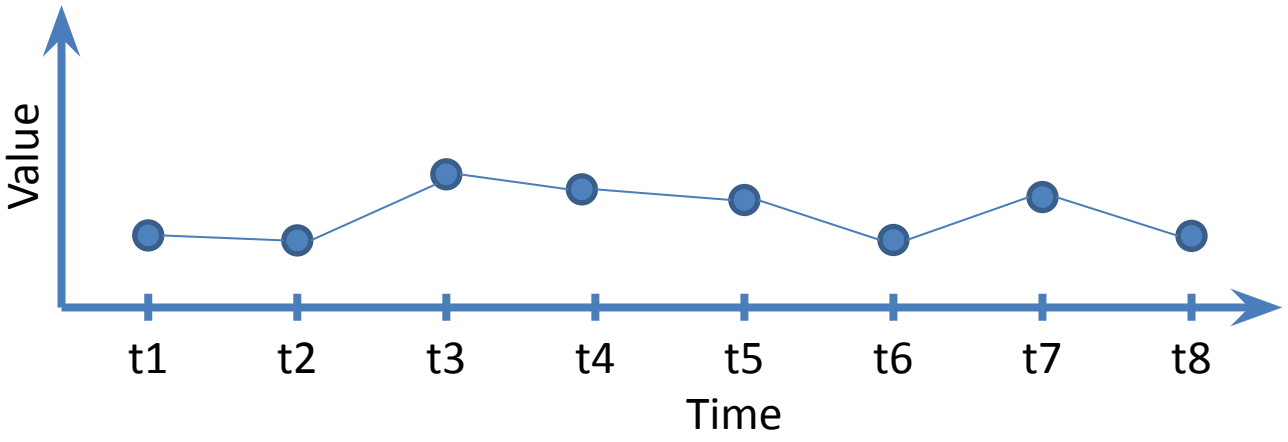
Presentation of the concept of clustering time series based on their features

Feature extraction from time series

Application of non-hierarchical methods for clustering

Time series

A sequence of data (observations) that are ordered in time (the domain is time).

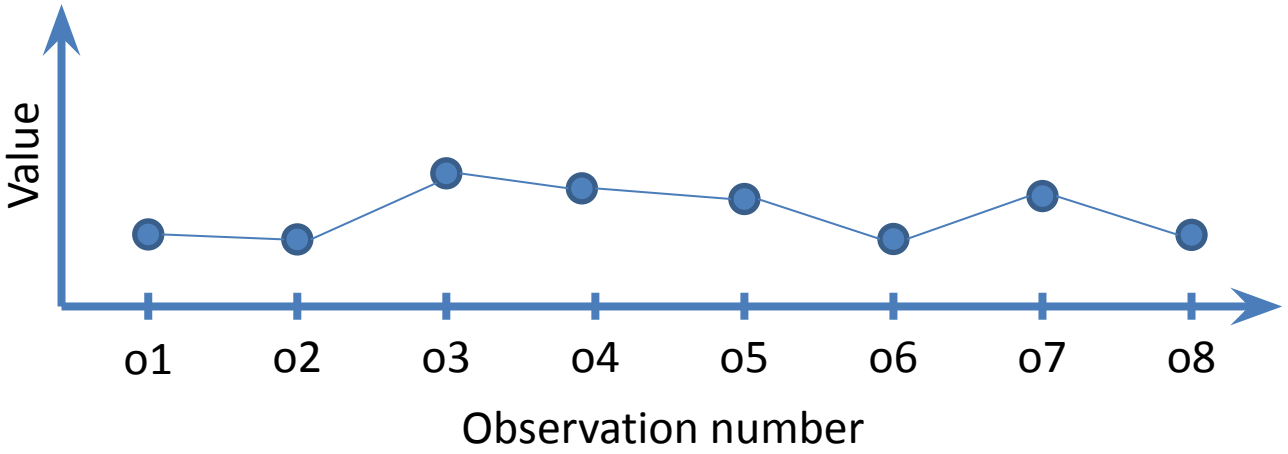


Examples:

- Values of certain parameters recorded every second by sensors during operation of a certain device (e.g., speed, vibration),
- The temperature in the production hall recorded every 10 minutes,
- The number of products produced during each working shift,
- Daily electricity consumption,
- The number of employees absent each day,
- Altitude above the ground during flight measured every second,
- ECG,
- Weekly store revenue,
- Daily bank account balance,
- Poland's unemployment rate by quarter,
- Stock market index values at the end of each day,
- and many more...

"Non-time" series

A string of data (observations) that are ordered (the order of the data is important).



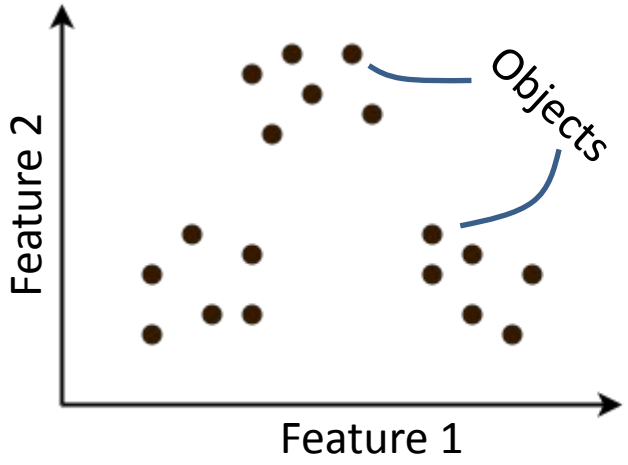
All of the time series analysis methods and techniques that will appear in today's class can also be used for "non-time" series.

Clustering

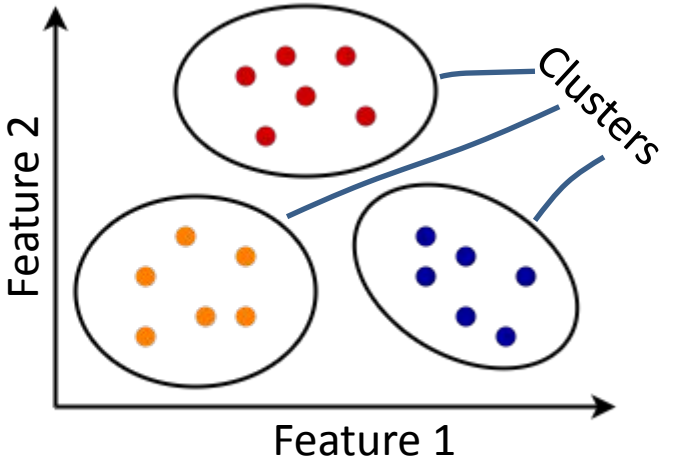
The task of finding clusters, also called groups, in a set of objects.

The goal: objects included in the same cluster must be similar to each other in terms of the characteristics under study, while objects from different clusters should be as different from each other as possible. Clustering is otherwise known as clustering or patternless classification.

Before clustering



After clustering



Time series clustering

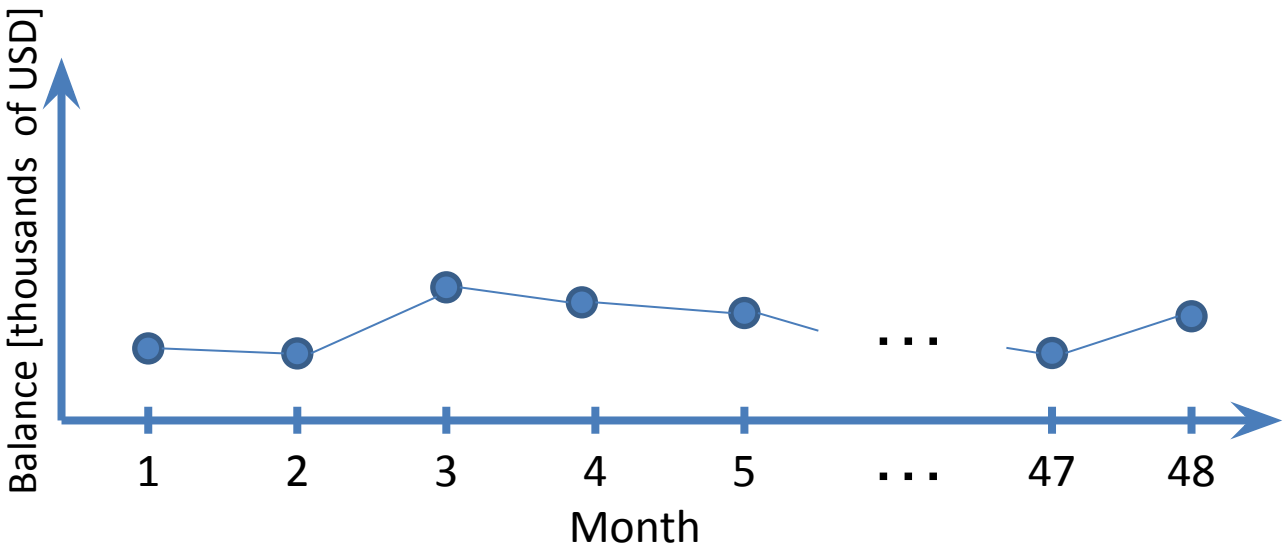
Time series clustering is an important task because:

- Facilitates the discovery of patterns present in the data - generating clusters for a certain set of data makes it easier to understand the structure of the data, allows you to more easily spot anomalies or other types of patterns present in the time series,
- Makes it easier to review large amounts of data - datasets containing time series can be huge (especially nowadays), making it difficult for an analyst to review such a large amount of data,
- Clustering is the most commonly used exploratory technique - it is part of the more complex tasks of:
  - Discovering rules (patterns) in time series,
  - Classification of time series,
  - Detection of anomalies in time series.

The task

Clustering of bank customers

Customer clustering (dividing customers into groups) is important for various industries and institutions. It is used, among others, in banking, where customer clustering improves, for example marketing campaigns or risk management.



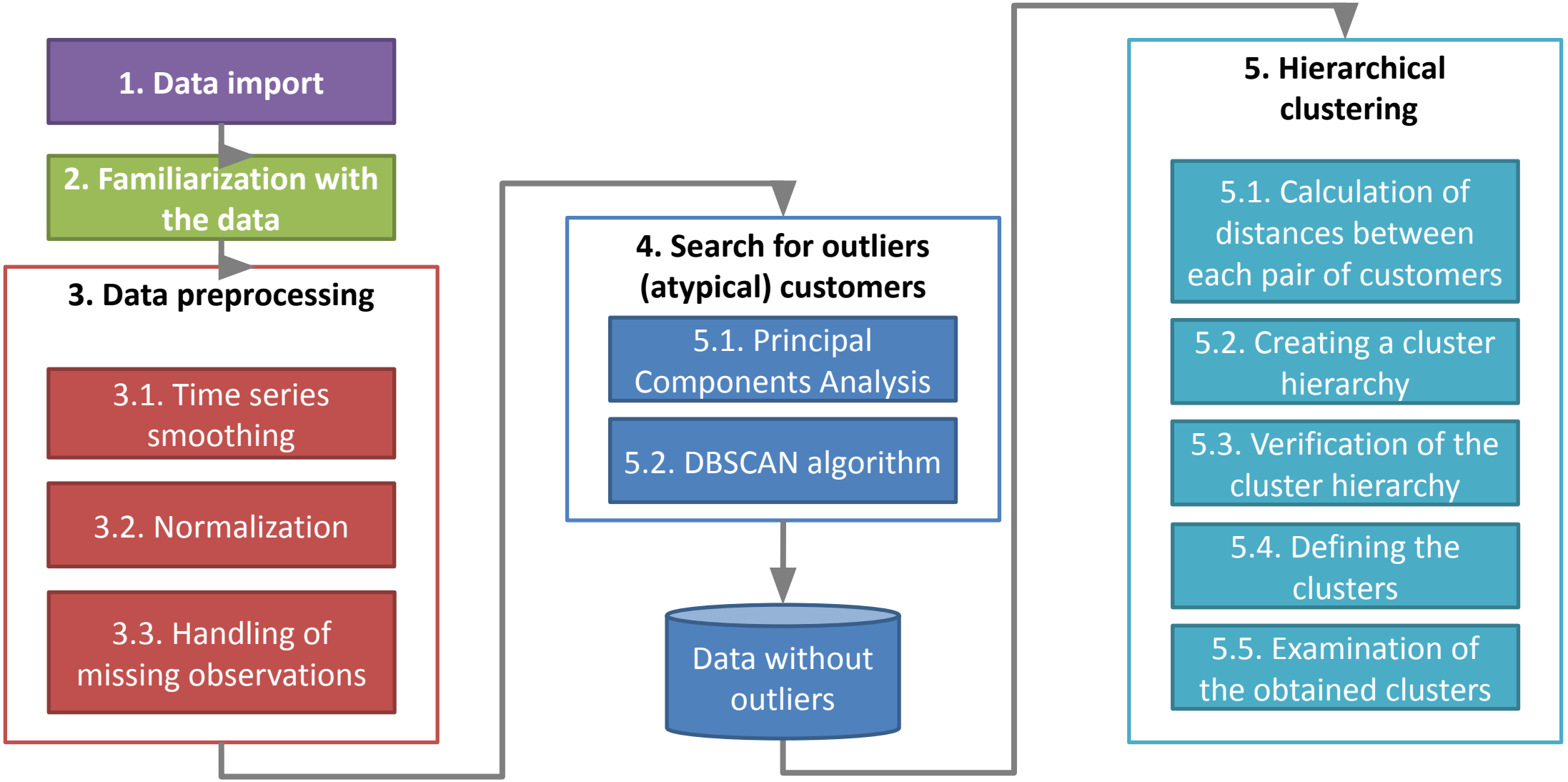
Each customer is described by **48** numerical values - the balance of the bank account (balance) at the end of the month.

The data covers **4** years.

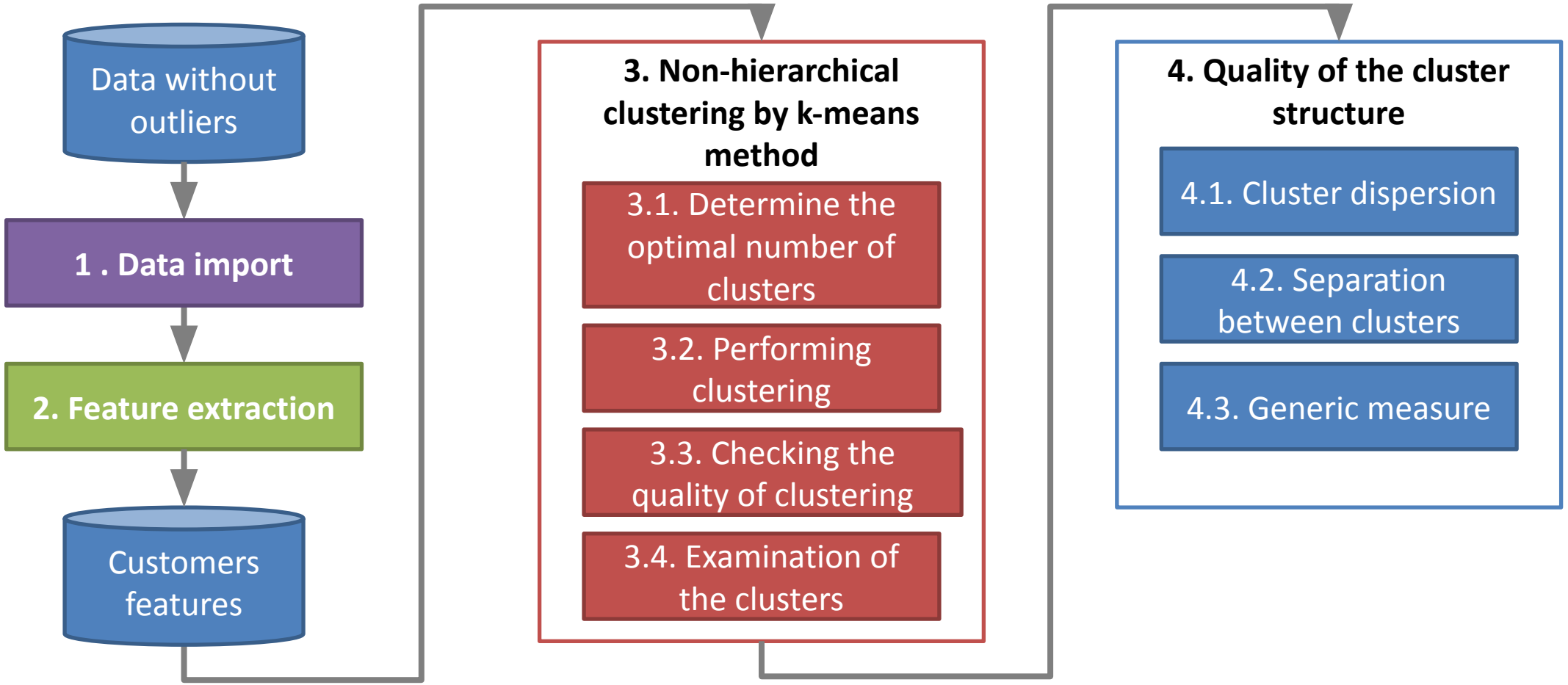
We have **5869** customers in the dataset, **or 5869 time series**.

We are looking for groups consisting of similar customers (similar in terms of their account balance in consecutive months).

A reminder of the previous task



Course of the task

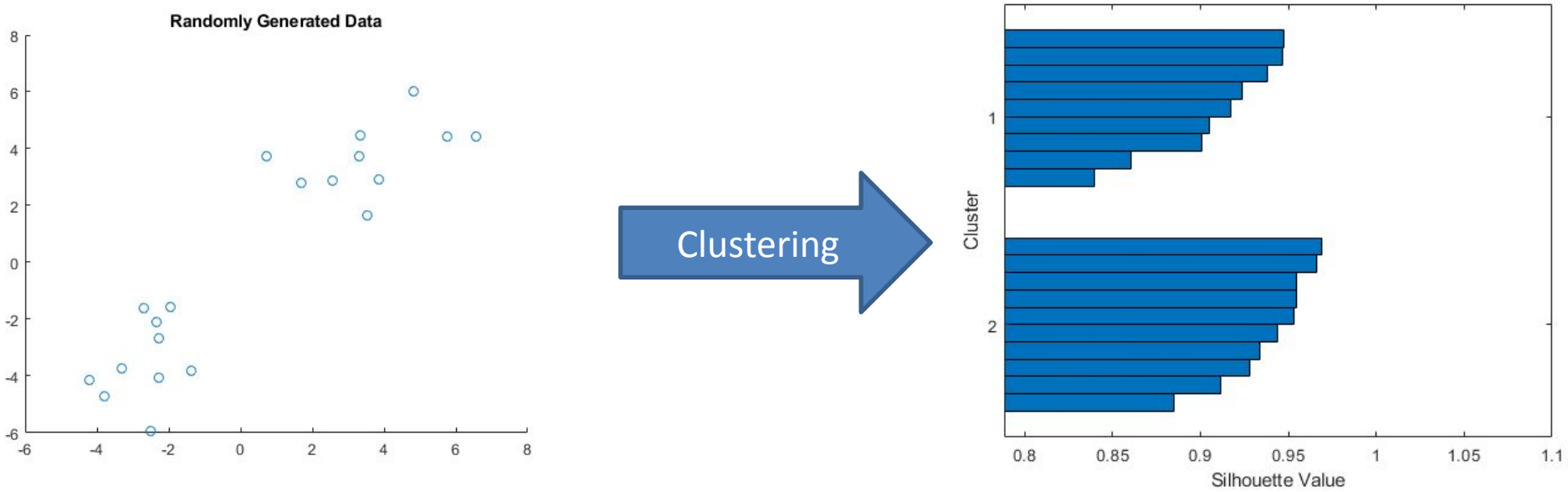


Evalclusters function

Uses one of four criteria to determine the optimal number of clusters:

- Calinsky-Harabasz index - to maximize,
- Davies-Bouldin index - to minimize,
- Gap statistic - to maximize,
- Outline measure (silhouette) - to maximize.

The average value of the outline for the entire dataset can be used to determine the quality of clustering. The outline can be visualized using an outline graph (silhouette function).

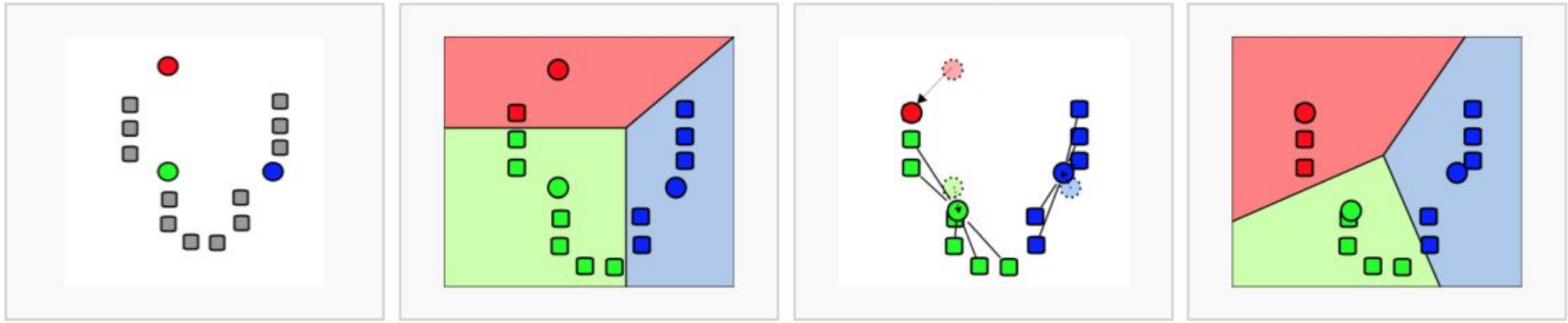


Non-hierarchical clustering

The most popular non-hierarchical clustering algorithm is the k-means algorithm.

- The final result of the algorithm (i.e., the resulting clusters) is heavily influenced by the initial random division of objects into clusters.
- The algorithm is not guaranteed to find the optimal division of objects (it may get stuck in the local optimum).
- In view of the above points, it is recommended to apply this algorithm repeatedly with different initial distributions of objects into clusters.
- It is necessary to determine k (the number of classes) before even running the algorithm.

Like hierarchical methods, the k-means algorithm is best used for spherical or convex clusters in data that are compact and clearly separated from each other. In addition, the results of these methods are affected by outlier observations and noise in the data.



Thank You for your attention!