Università degli Studi di Padova

Machine Learning
2024/2025

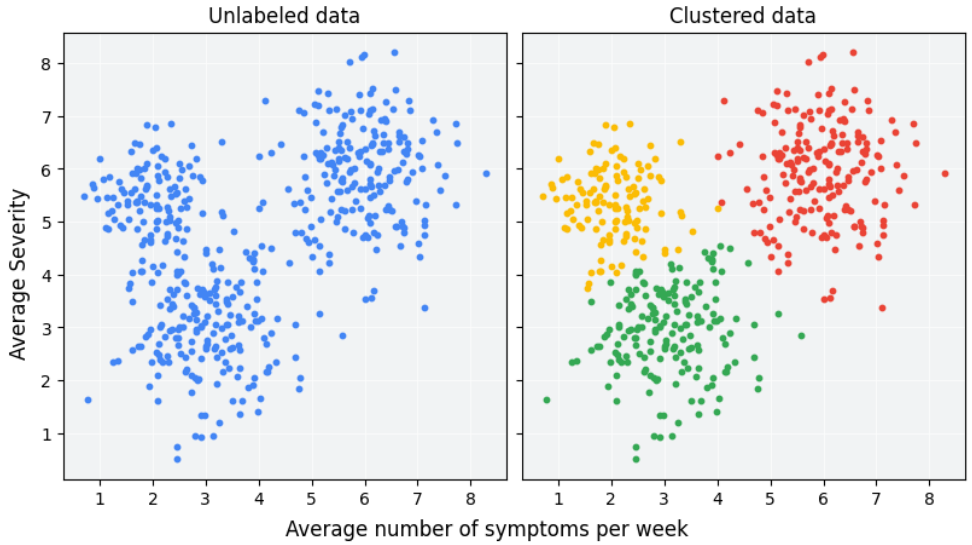
AMCO
Artificial Intelligence, Machine Learning and Control Research Group

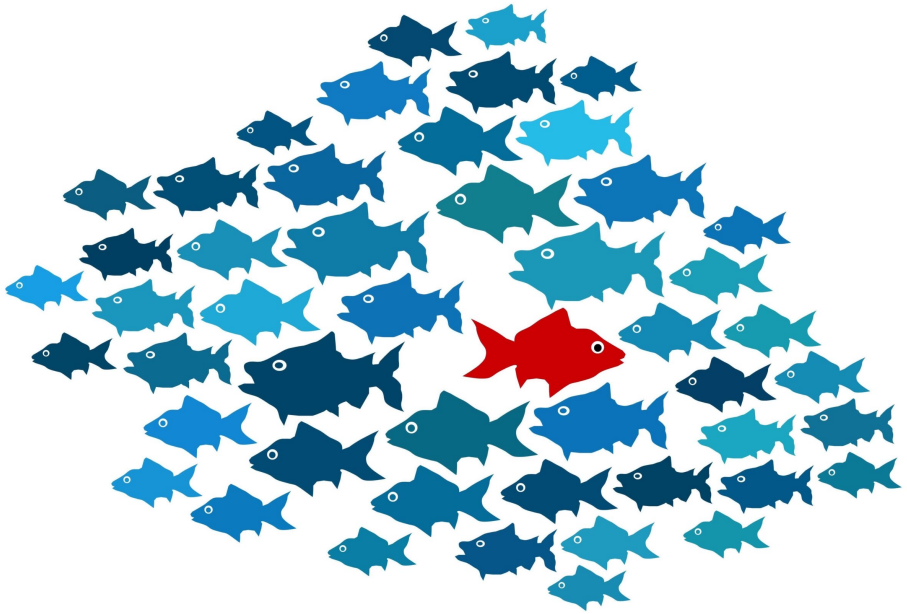
Lecture #20 Unsupervised Learning & Clustering

Gian Antonio Susto

Recap: some unsupervised learning tasks

- Clustering: finding groups into data (today!)
- Dimensionality reductions: reduce the number of features
- Density estimation: estimate the probability distribution that generates the data
- Association rule learning
- Topic modelling (for text data)
- Anomaly/Outlier detection (monday/today!)



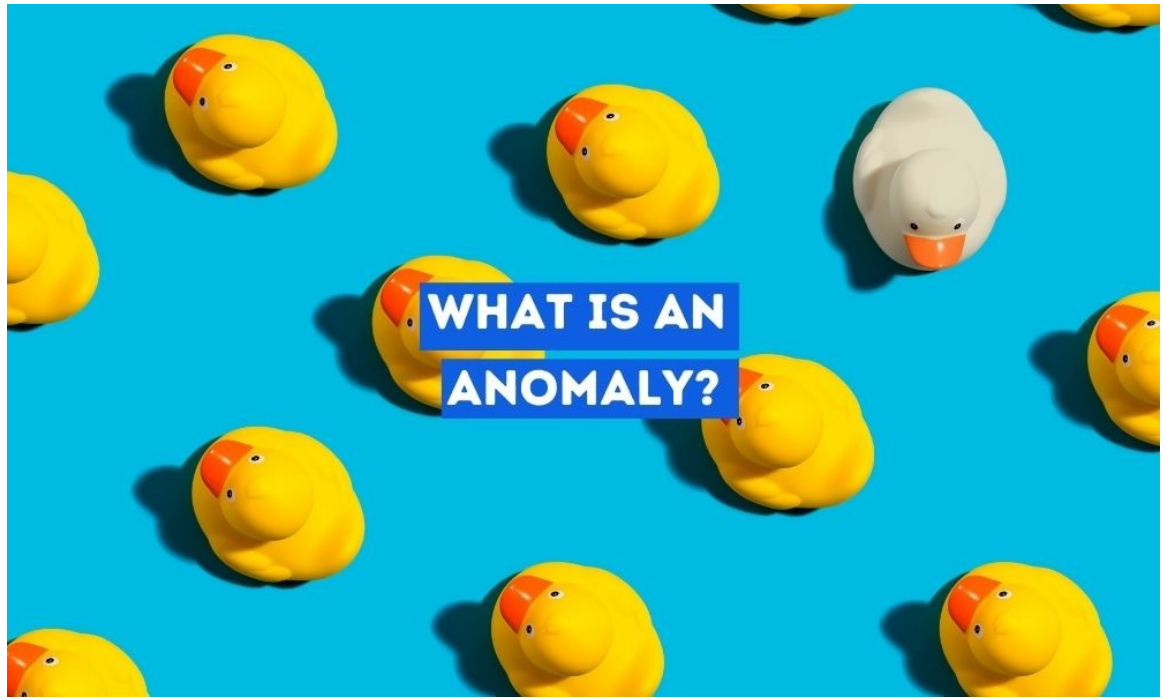


Recap: Anomaly/Outlier Detection

What is an anomaly/outlier?

'An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism' [1]

In a dataset – by definition – outliers should be few!

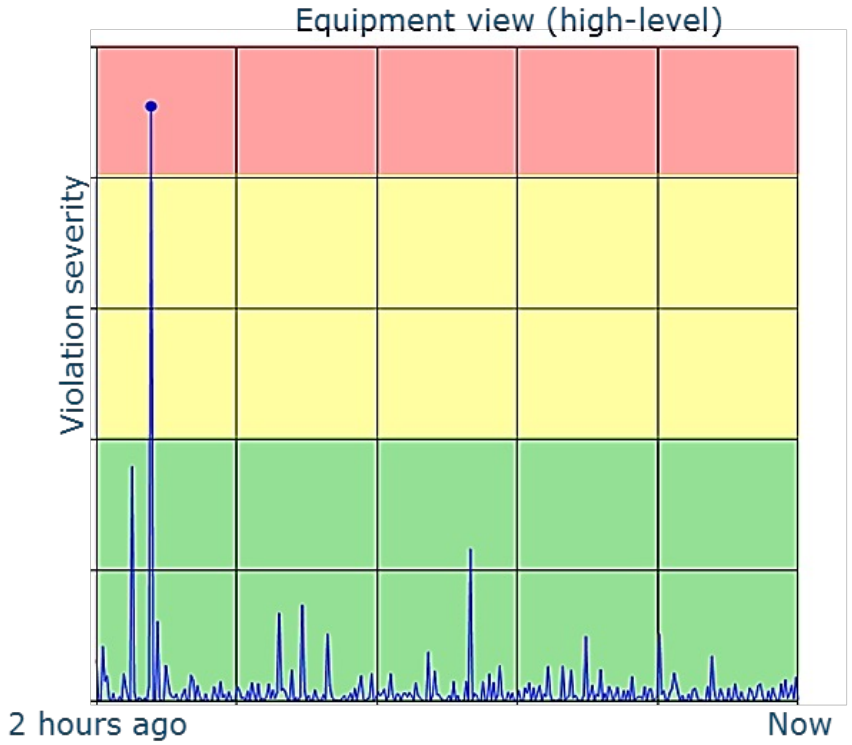


[1] D. M. Hawkins, Identification of outliers, vol. 11., Springer, 1980

Recap: Multivariate Unsupervised Anomaly Detection

Many approaches for tabular data (data where rows are observations and columns are variables) [2]:

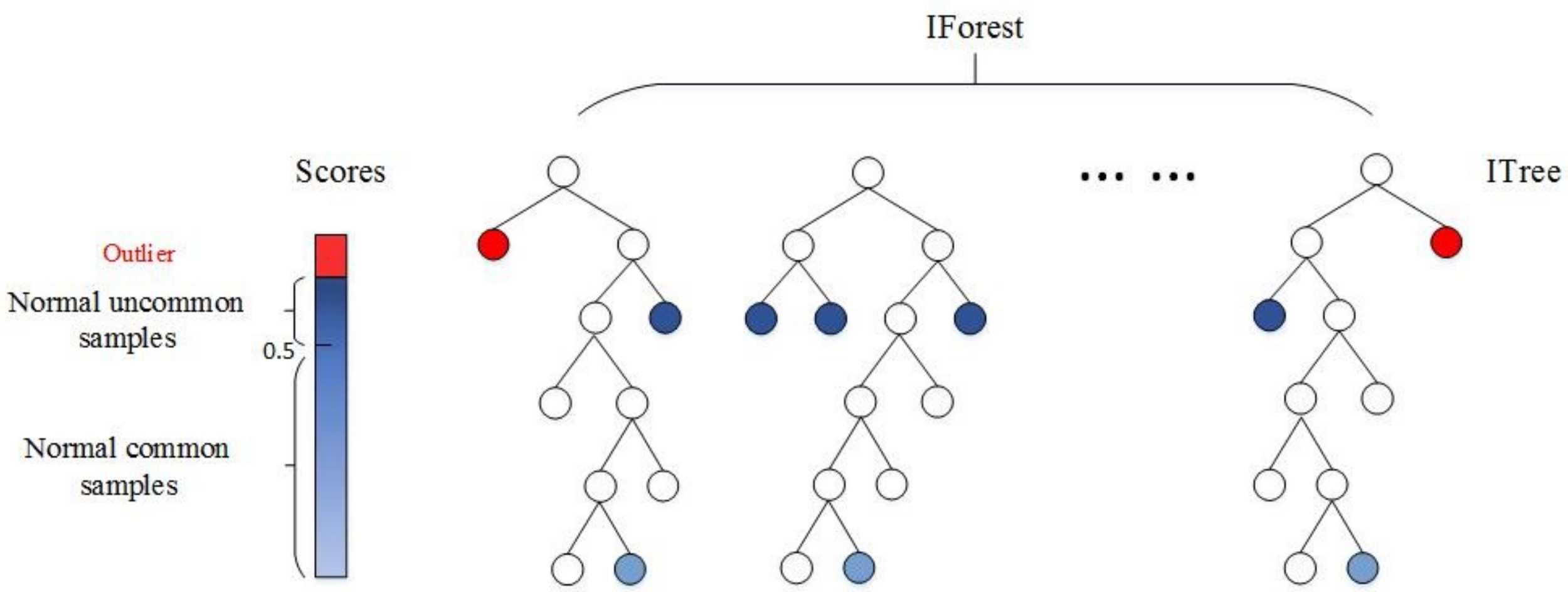
- Density-based methods (e.g. LOF, DBSCAN)
- Distance-based methods (e.g. kNN)
- Clustering-based methods (e.g. CBLOF)
- Neural Networks (e.g. Autoencoder)
- Isolation Forest
- …



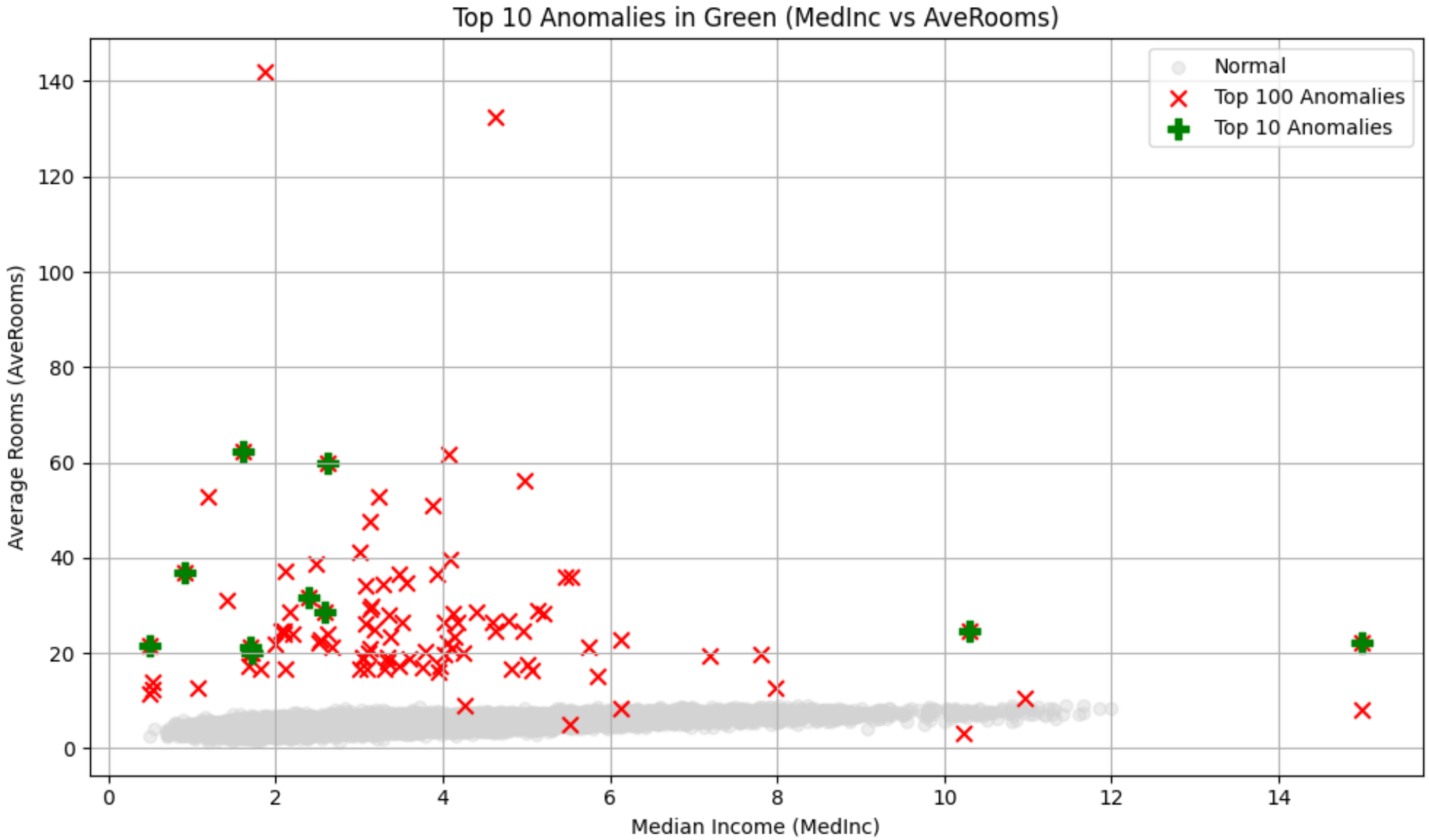
[2] PyOD (Python library for detecting outlying objects)
https://pyod.readthedocs.io/en/latest/

Recap: Isolation Forest

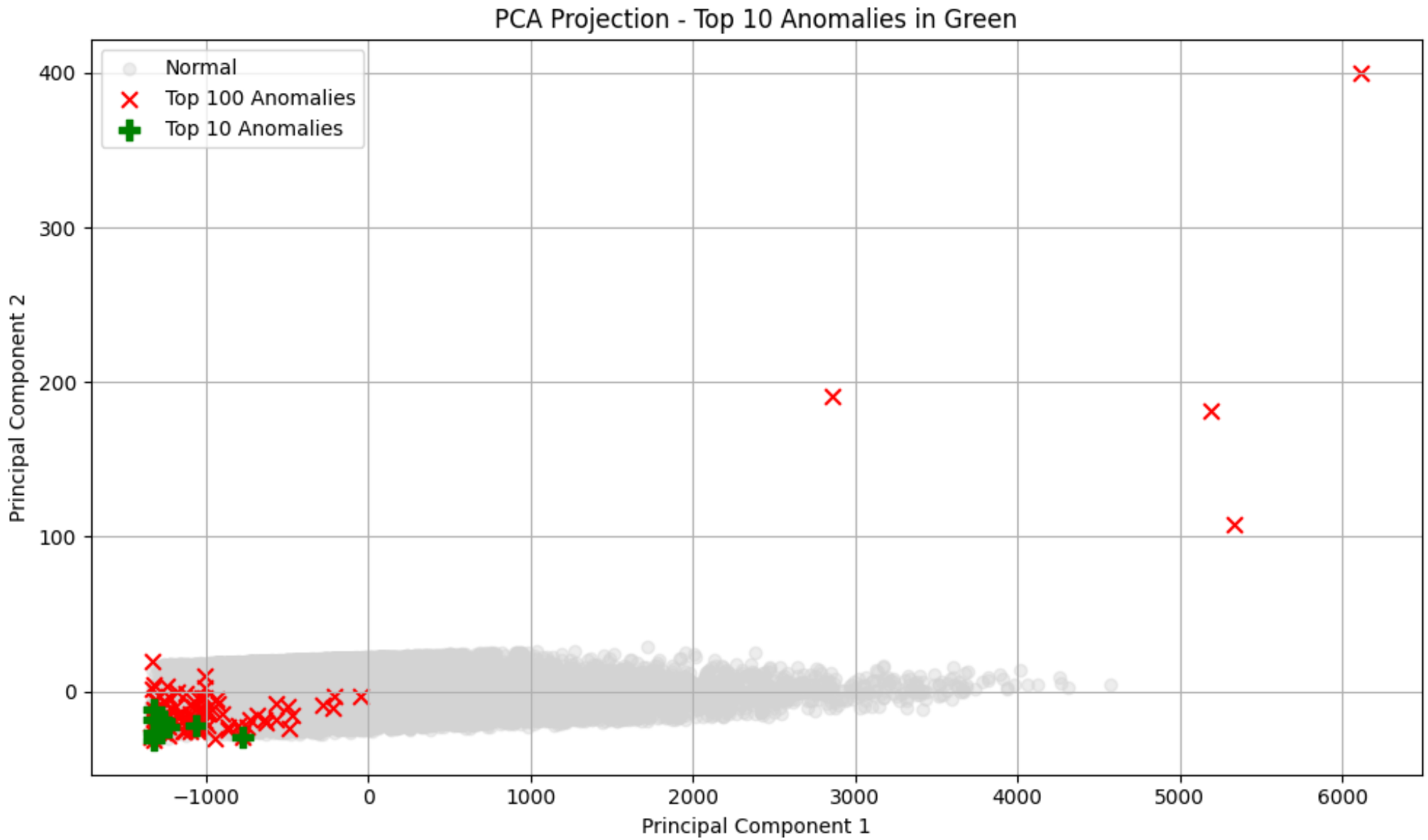
An ensemble approach: anomaly score computed as mean of the depth over the various isolation trees



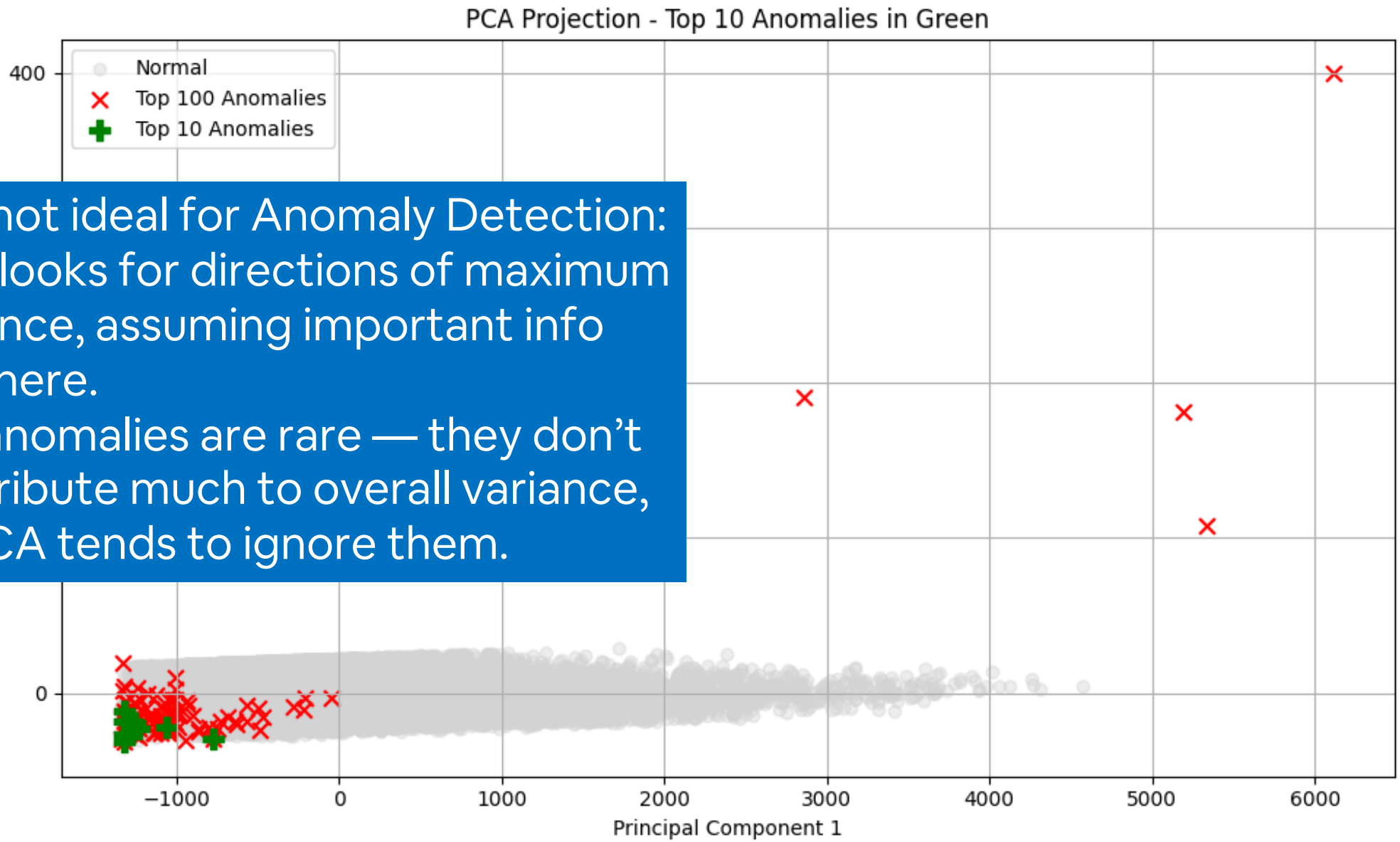
Recap: California housing dataset



Recap: California housing dataset



Recap: California housing dataset



PCA is not ideal for Anomaly Detection:

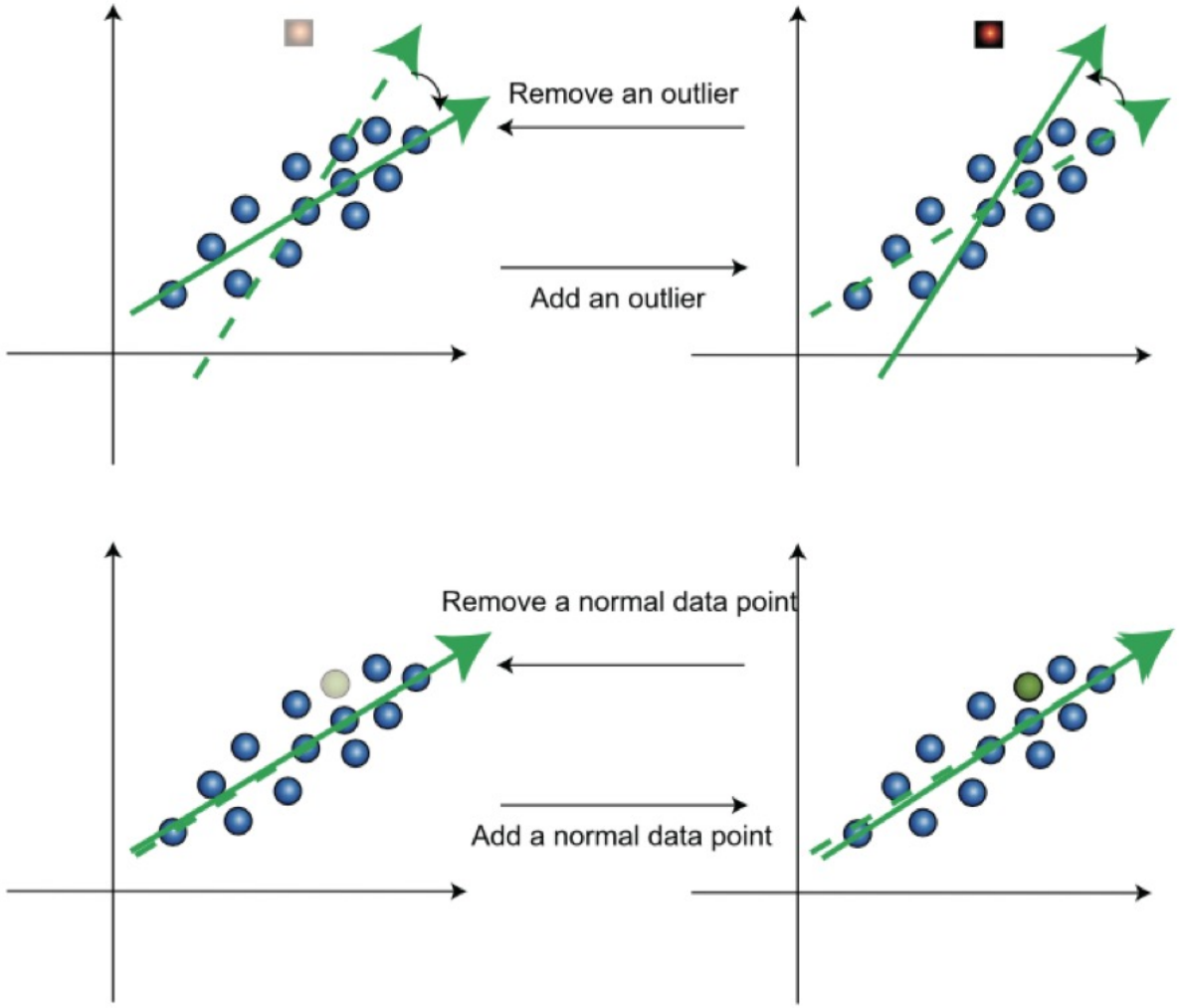
- PCA looks for directions of maximum variance, assuming important info lies there.
- But anomalies are rare — they don't contribute much to overall variance, so PCA tends to ignore them.

Oversampling PCA: osPCA

Idea:

- principal directions represent “normal” attributes
- outliers change directions of principal components

To make it effective, each data sample, when checked if it changed the direction of the first PC, is oversampled (replicated many times)

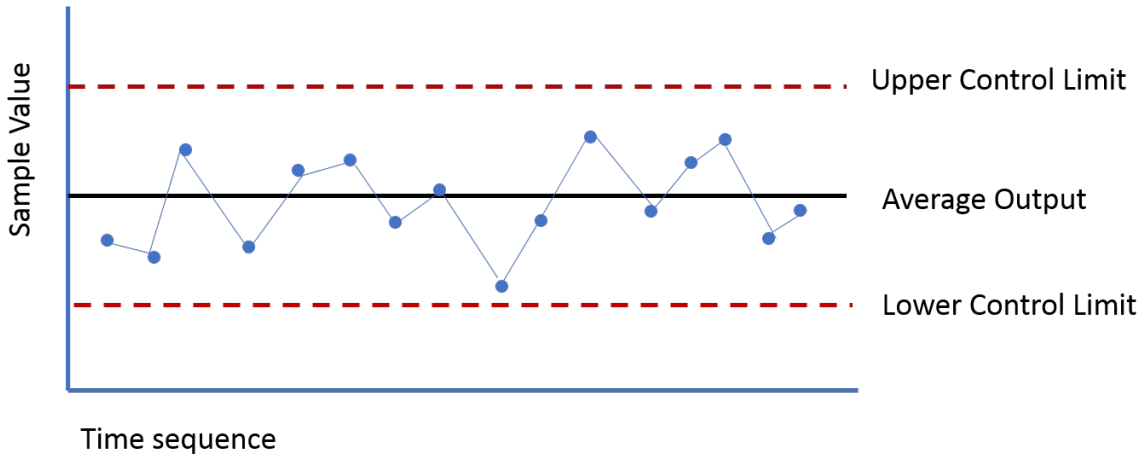


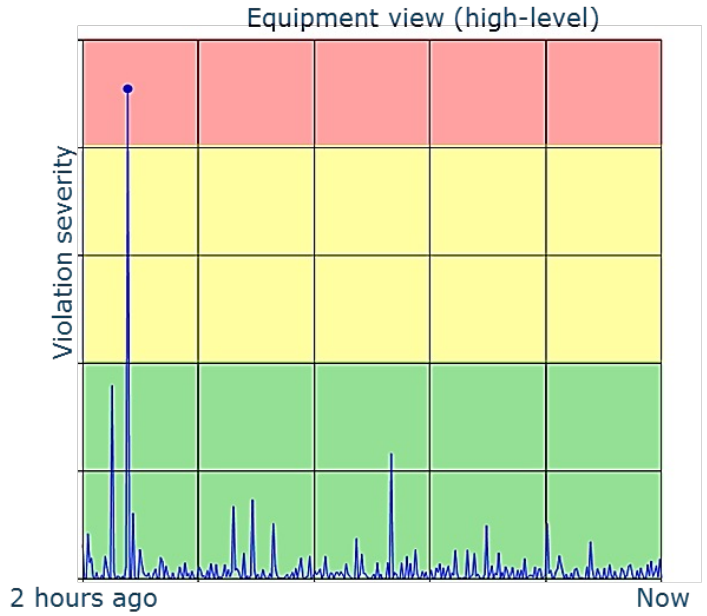
Lee, Yuh-Jye, Yi-Ren Yeh, and Yu-Chiang Frank Wang. "Anomaly detection via online oversampling principal component analysis.”, 2013.

Univariate Control Chart vs Unsupervised AD

To enable Decision Making information should be:

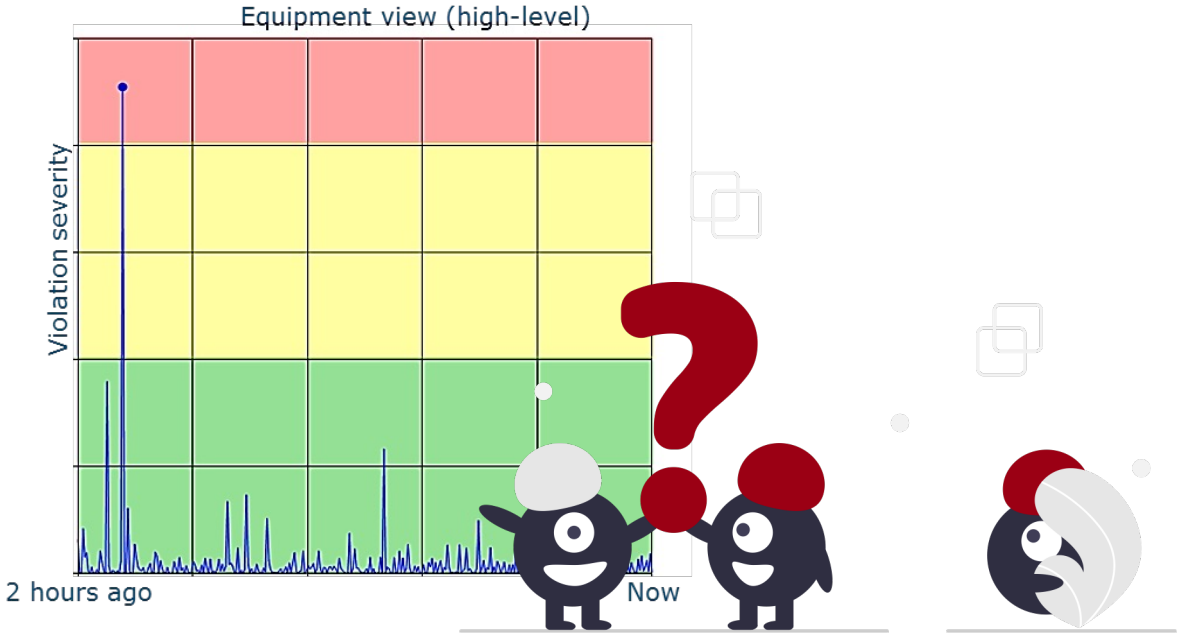
✗	complete
✗	concise
✓	interpretable

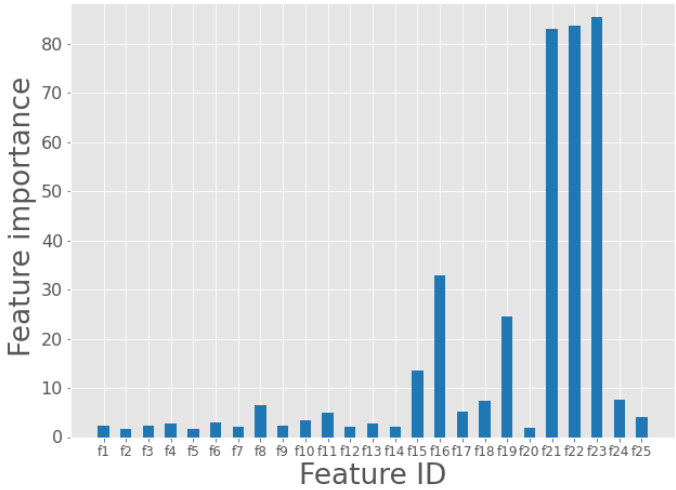




I’ve got the Anomaly Score: now, what?

- Thanks to the Anomaly score users are alerted of potential anomalous situation, however it is up to them to discover potential troubles
- It would be nice to ease the Root Cause Analysis to provide additional information, like feature rankings…



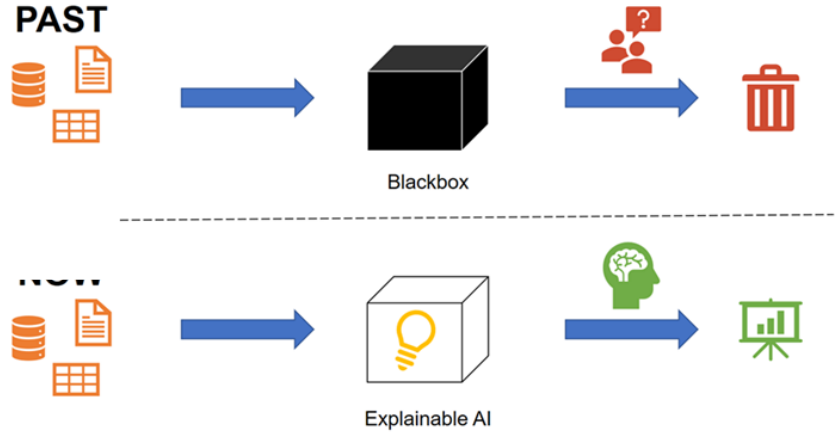


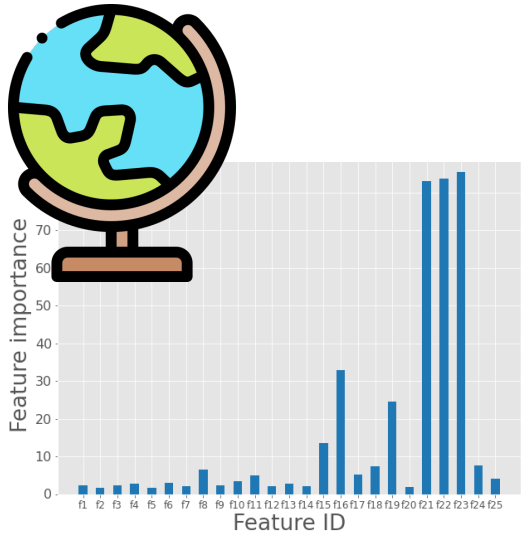
Depth-based Isolation Forest Feature Importance (DIFFI) [2]

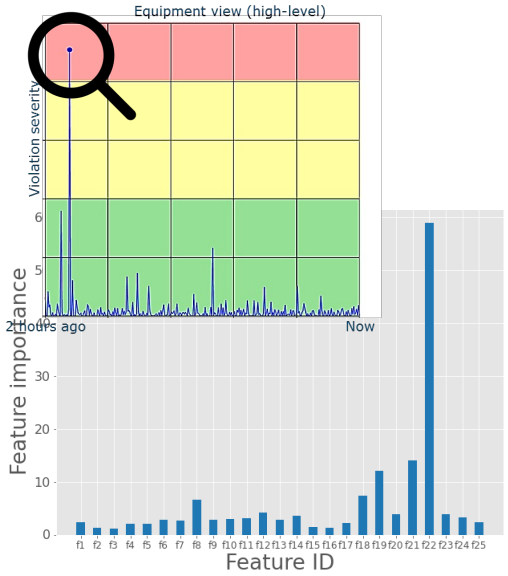
DIFFI is an eXplainable Artificial Intelligente (XAI) approach designed for the Isolation Forest

DIFFI provides a variable ranking for:

- Global Explainability (ie. what variables are important for the whole Isolation Foreset model)
- Local Explainability (ie. what variables are important for a particular prediction)

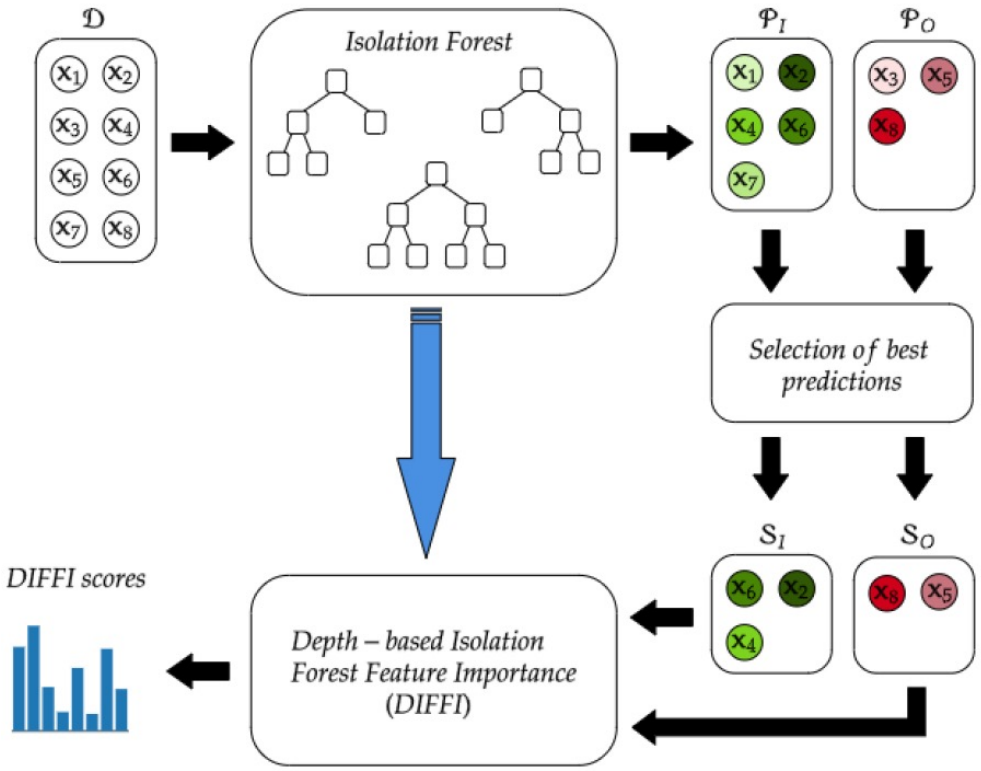






[2] Carletti, M., Terzi, M., & Susto, G. A. (2023). Interpretable Anomaly Detection with DIFFI: Depth-based feature importance of Isolation Forest. *Engineering Applications of Artificial Intelligence*, *119*, 105730.

Depth-based Isolation Forest Feature Importance (DIFFI) [2]



DIFFI provides a variable ranking that:

- Does not require true labels (other XAI approaches do!)
- Low computational cost
- No tuning

IDEA: mark a feature as "important" if

- it induces isolation of outliers at small depths (i.e. near the root)
- At the same time, does not contribute to the isolation of inliers

[2] Carletti, M., Terzi, M., & Susto, G. A. (2023). Interpretable Anomaly Detection with DIFFI: Depth-based feature importance of Isolation Forest. *Engineering Applications of Artificial Intelligence*, *119*, 105730.

For technical details

Timeframe

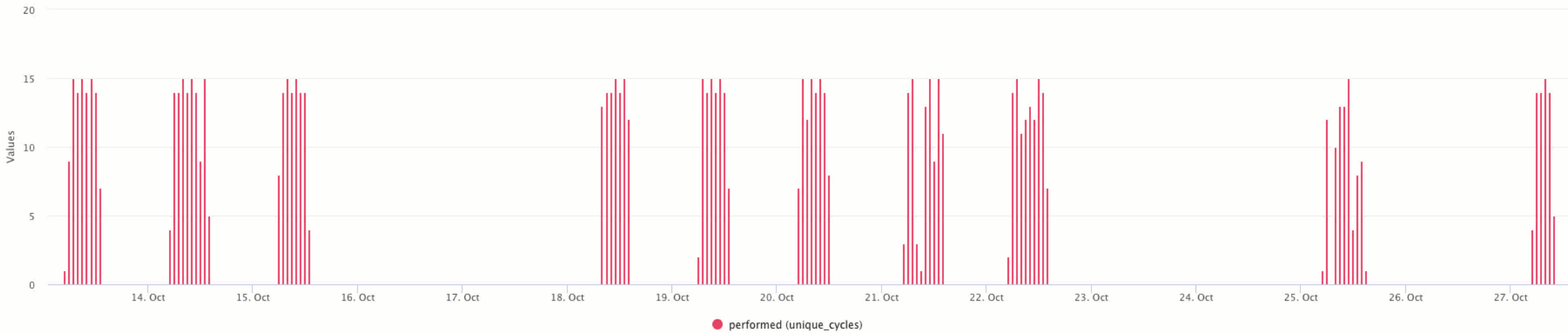
All Time

Capacity

Capacity is measured in number of cycles.

980	980	0	0
Total Cycles	Performed Cycles	Stopped Cycles	Estop Cycles

Total Hourly Capacity

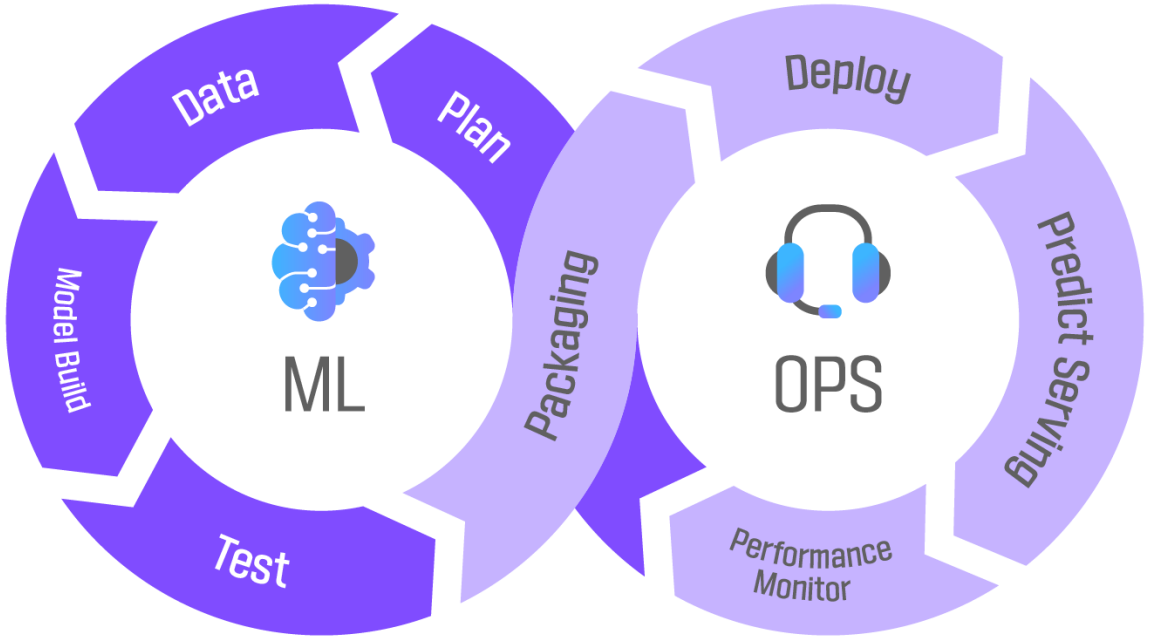


Duration

Duration is the sum of Working Time, Stopped Time, Downtime and Waiting Time.

Why detecting anomalies? For monitoring of a productive solution (3/3)

- Once we deploy a Machine Learning model in production, is not over!
- For example, we should monitor if the underlying data are consistent with what we had in training
- Anomaly Detection can be useful for monitoring! If we have outliers, we can avoid trusting a single prediction or we can start with re-train
- This is part of MLOps (Machine Learning operations) principles



How to evaluate an Unsupervised Anomaly Detection system if we are in unsupervised settings?

- That’s the true drawback of unsupervised approaches!
- If you can't get ground truth labels, try to get approximate labels or insights:

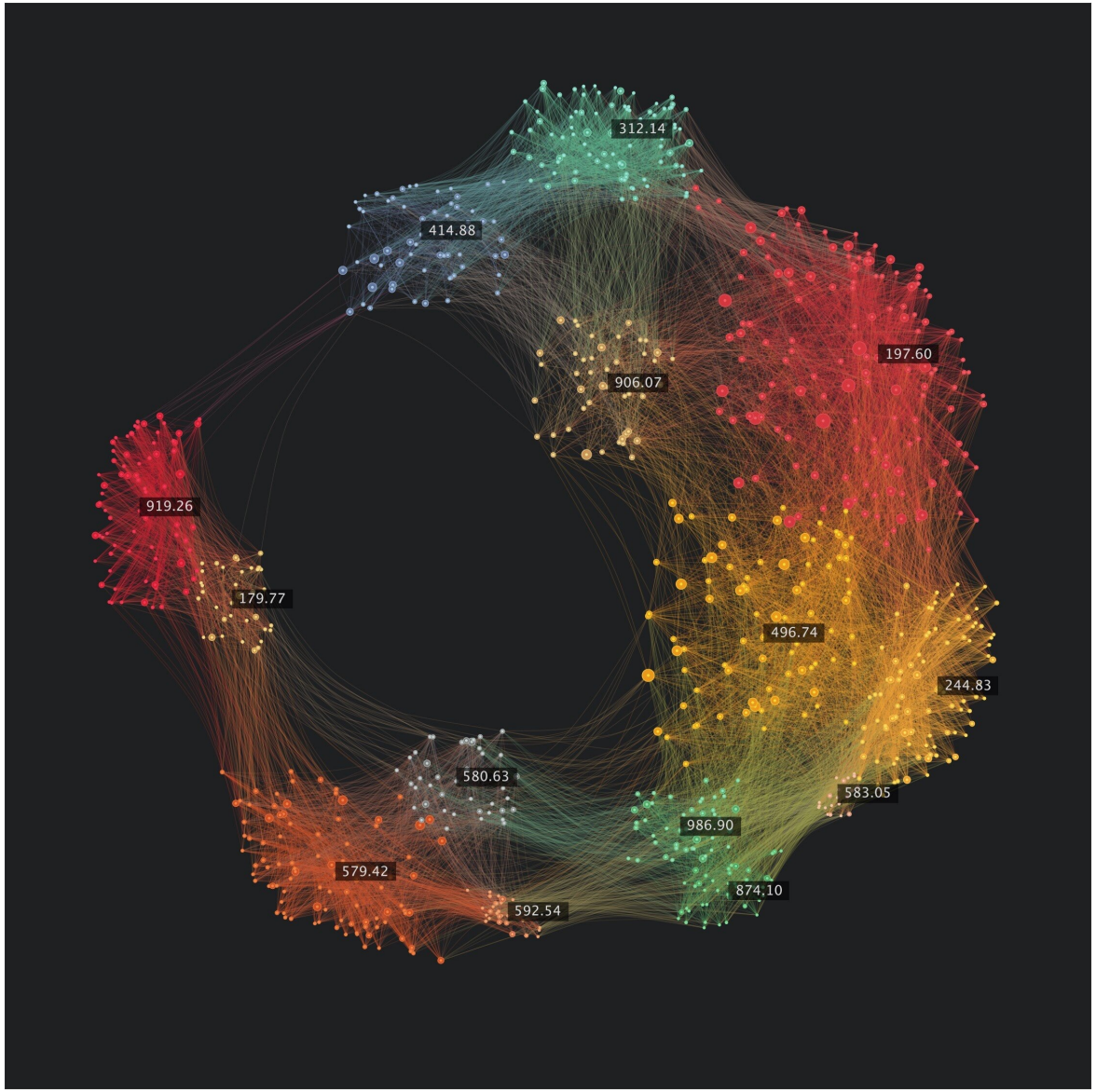
(i) Domain experts: Ask them to review top *N* anomalies flagged by your model. Are they meaningful?

(ii) Known anomaly cases: Use events like system failures, alerts, or log anomalies as partial labels.

(iii) Synthetic injection: Add known anomalies to the dataset (e.g. noise, out-of-distribution points) and check if your model finds them.

What is clustering?

Clustering is a type of unsupervised learning where the goal is to group similar data points together based on their features — without using any labels or predefined categories.



✅ What clustering does:

- Finds patterns or structure in your data
- Organizes data into clusters: groups where members are more similar to each other than to members of other groups

Why Clustering? As preprocessing step (1/2)

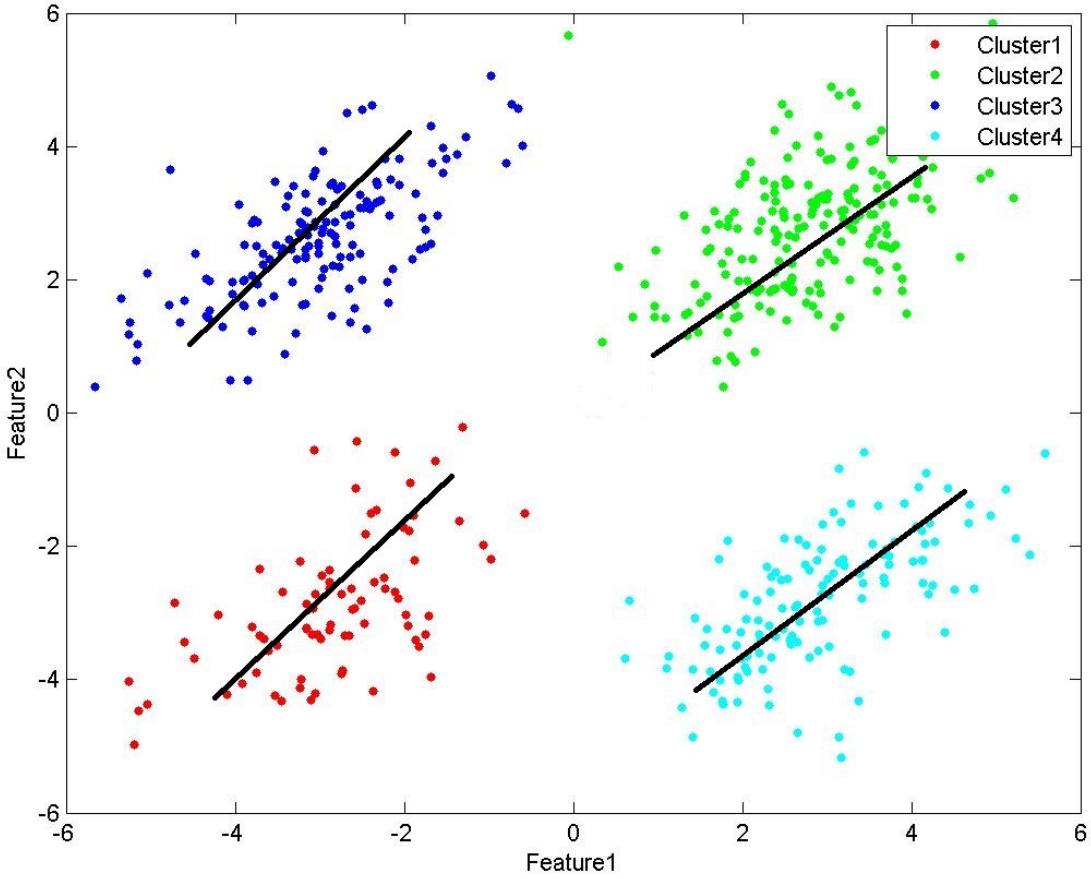
When data is heterogeneous or complex, clustering may be a good idea!

Local/multiple models:

- One model per cluster may dramatically increase performances!

Drawbacks:

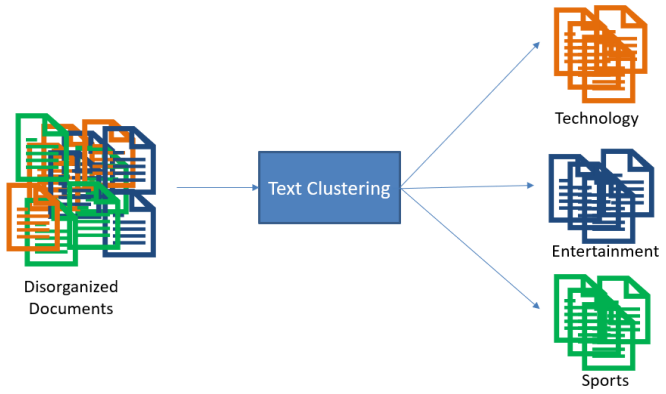
- More complexity to handle
- We are reducing the amount of data for each model, which can lead to lower performances if data are few in number

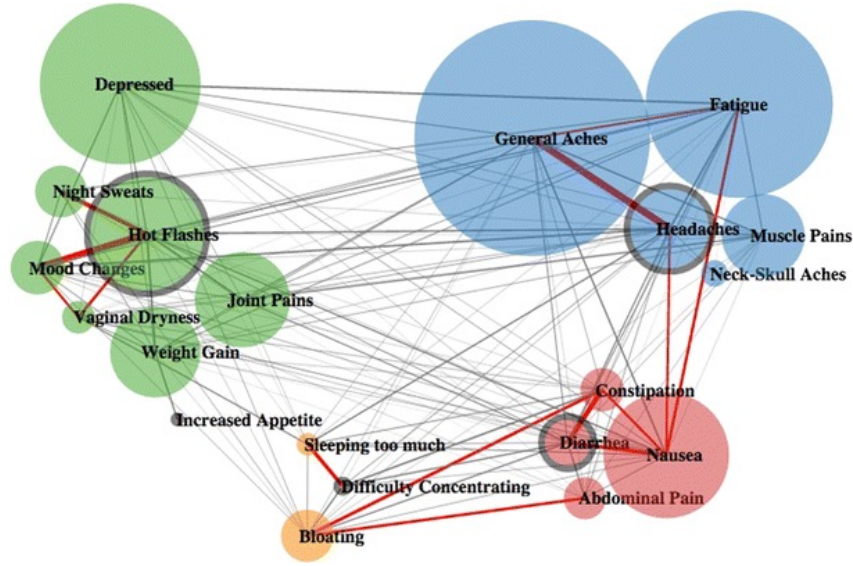


Why Clustering? As the final objective (2/2)

In many cases, clustering is a fundamental task by itself:

- Digital marketing/adverting (customer segmentation)
- Document organization (topic modelling)
- Web & user analytics (website user behaviour)
- Manufacturing (more later)
- …





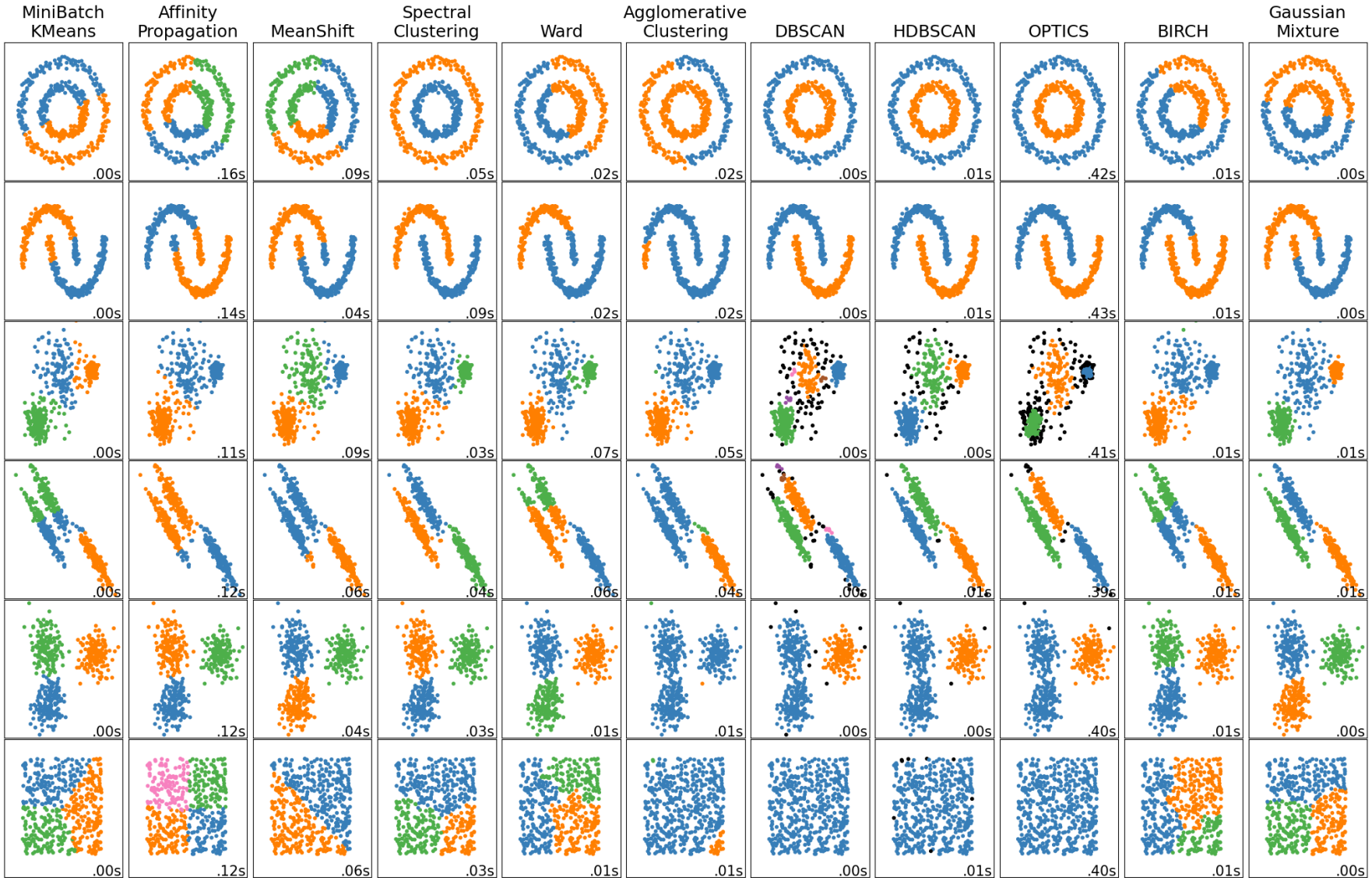
Unsupervised learning: either the final task or a preprocessing step!

As already seen with anomaly detection, also clustering is relevant both for pre-processing and as the final goal of a Machine Learning project!

Many approaches for clustering!

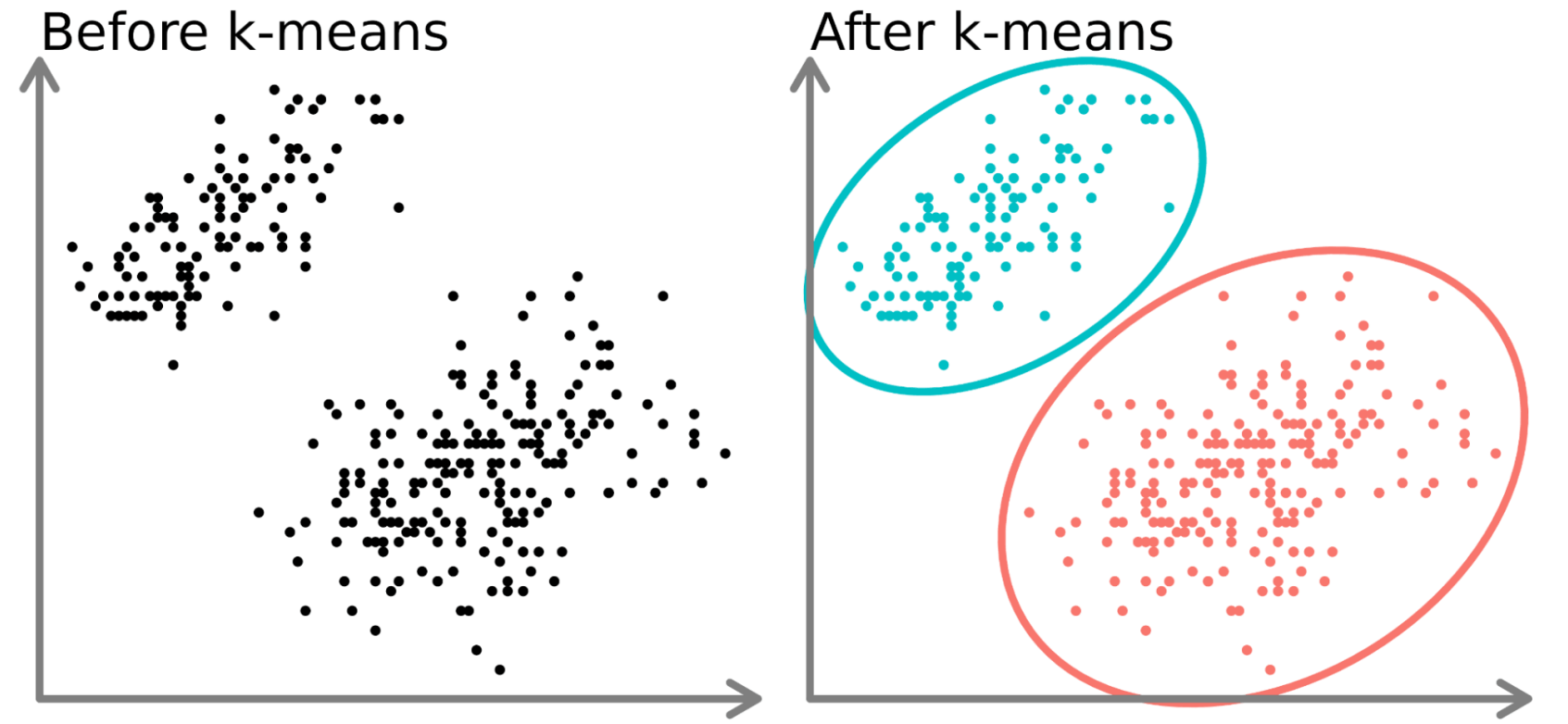
We are seeing 2 approaches:

- K-means
- Hierarchical Clustering



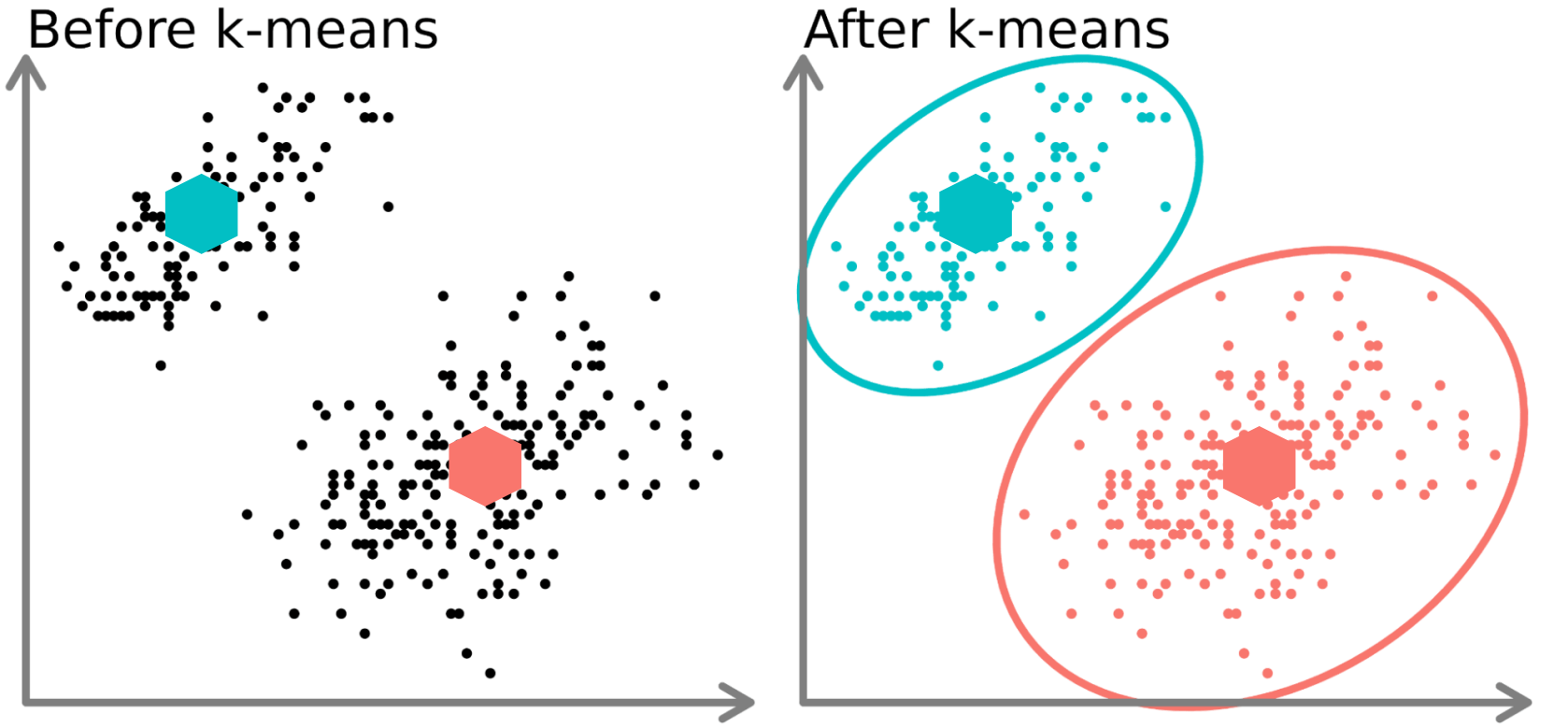
K-means

- Let’s suppose in this example, that we already know that we are looking for 2 clusters in the data
- How will you end up with a clustering as represented here? Ideas?



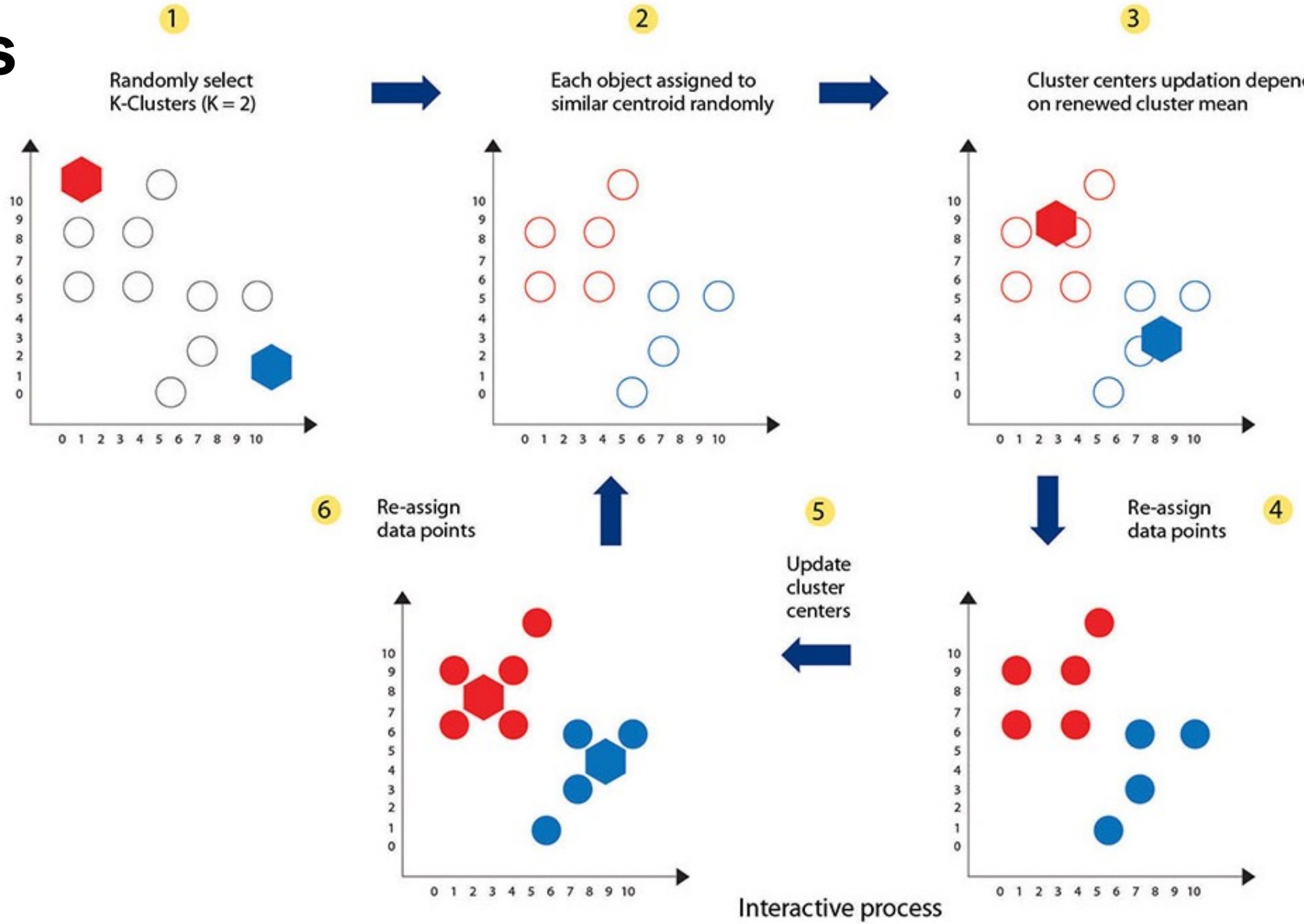
K-means

- Let's suppose in this example, that we already know that we are looking for 2 clusters in the data
- How will you end up with a clustering as represented here? Ideas?



- If we know the 'centroids' of the 2 clusters, we could assign each data point to the closest centroids!

K-means


1
Randomly select
K-Clusters (K = 2)
2
Each object assigned to
similar centroid randomly
3
Cluster centers updation depending
on renewed cluster mean
4
Re-assign
data points
5
Update
cluster
centers
6
Re-assign
data points
Interactive process

Steps of K-Means Clustering

1. Choose K (hyperparameter): decide the number of clusters K you want to find
2. Initialize Centroids: randomly select K data points as the initial centroids (cluster centers)
3. Assign Points to Nearest Centroid

- For each data point, compute the distance to all centroids.
- Assign the point to the closest centroid (based on Euclidean or another distance)

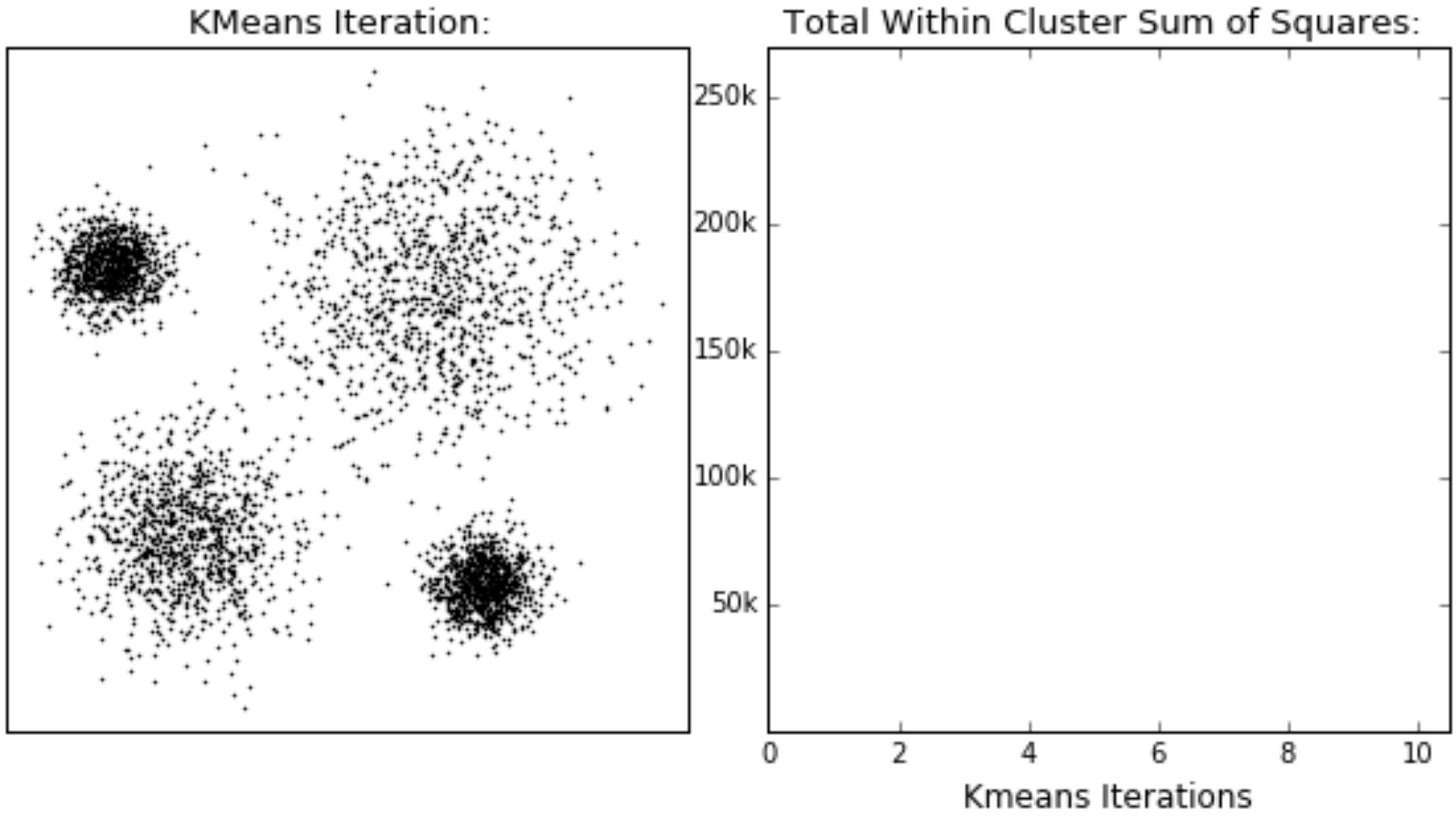
4. Update Centroids: recalculate each centroid as the mean of all points assigned to that cluster
5. Repeat 3-4 until: ‘convergence’ / max number of iterations

‘Convergence’ in clustering

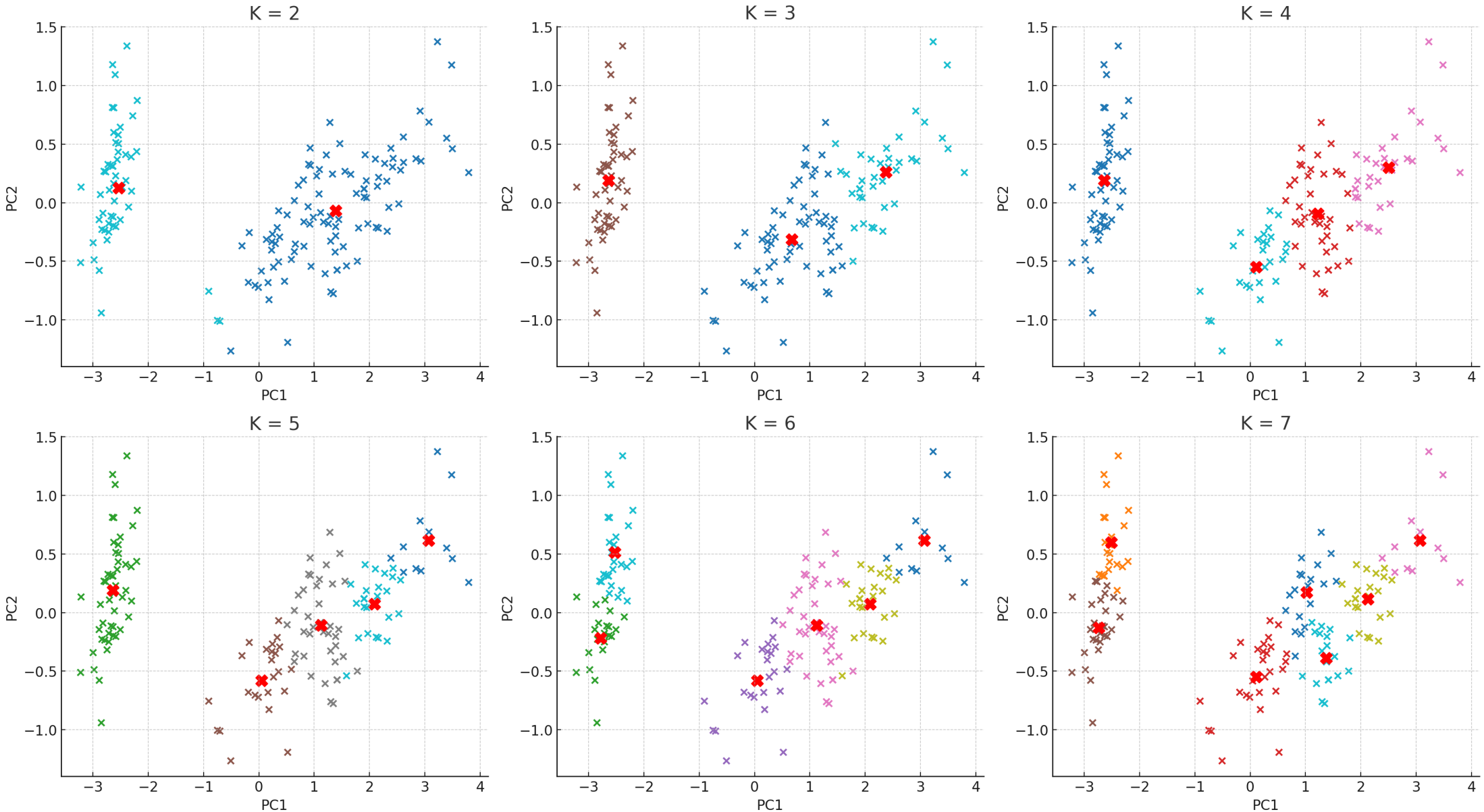
We need to define a metric: Mean Within-Cluster Mean Squared Error

$$\mathrm{MSE}_{\mathrm{within}} = \frac{1}{N} \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2$$

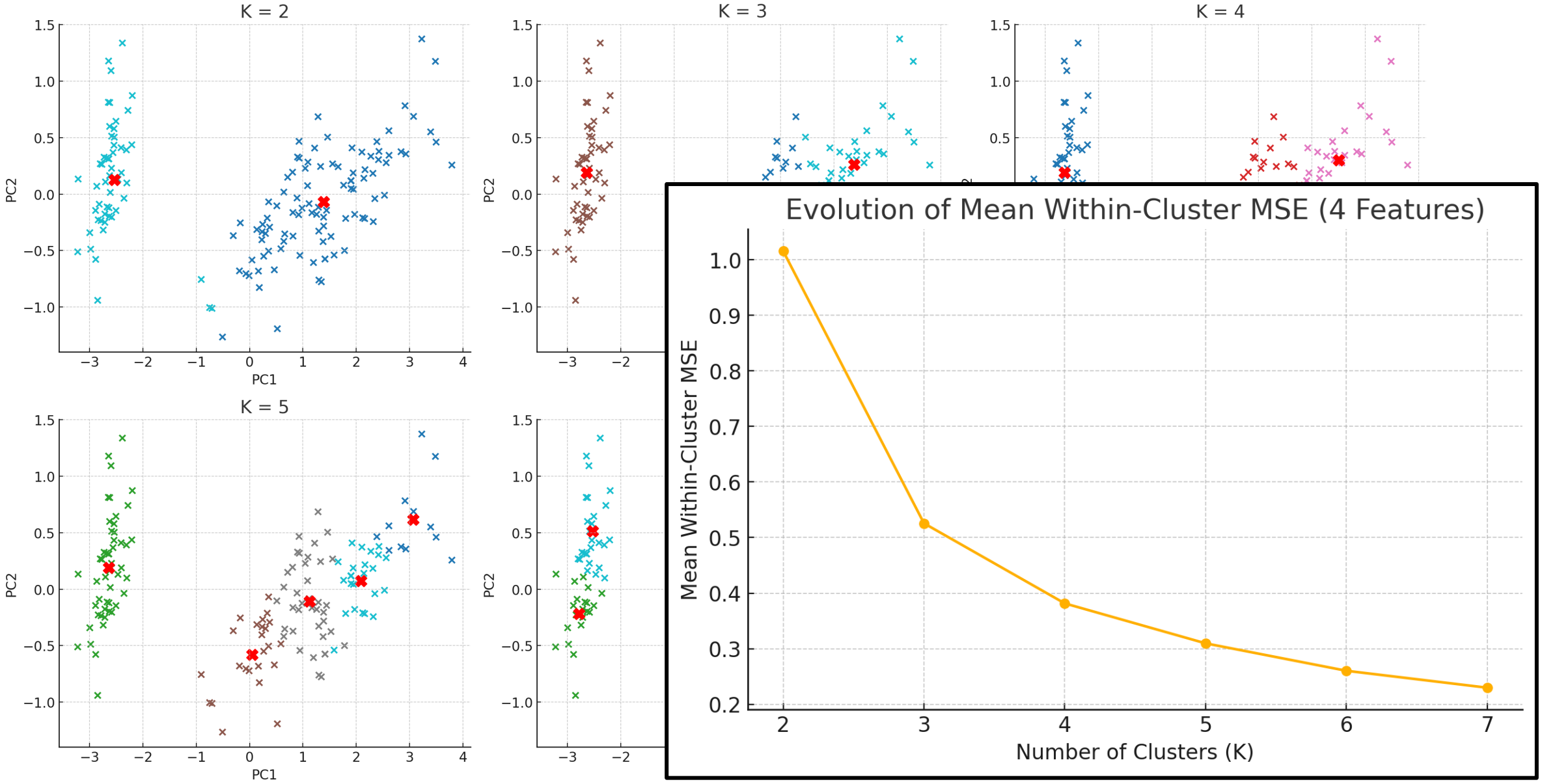
- N = total number of data points
- K = number of clusters
- C_k = set of points in cluster k
- x_i = a data point in cluster C_k
- μ_k = centroid (mean) of cluster C_k


KMeans Iteration:
Total Within Cluster Sum of Squares:
250k
200k
150k
100k
50k
0
2
4
6
8
10
Kmeans Iterations

K-means of Iris dataset



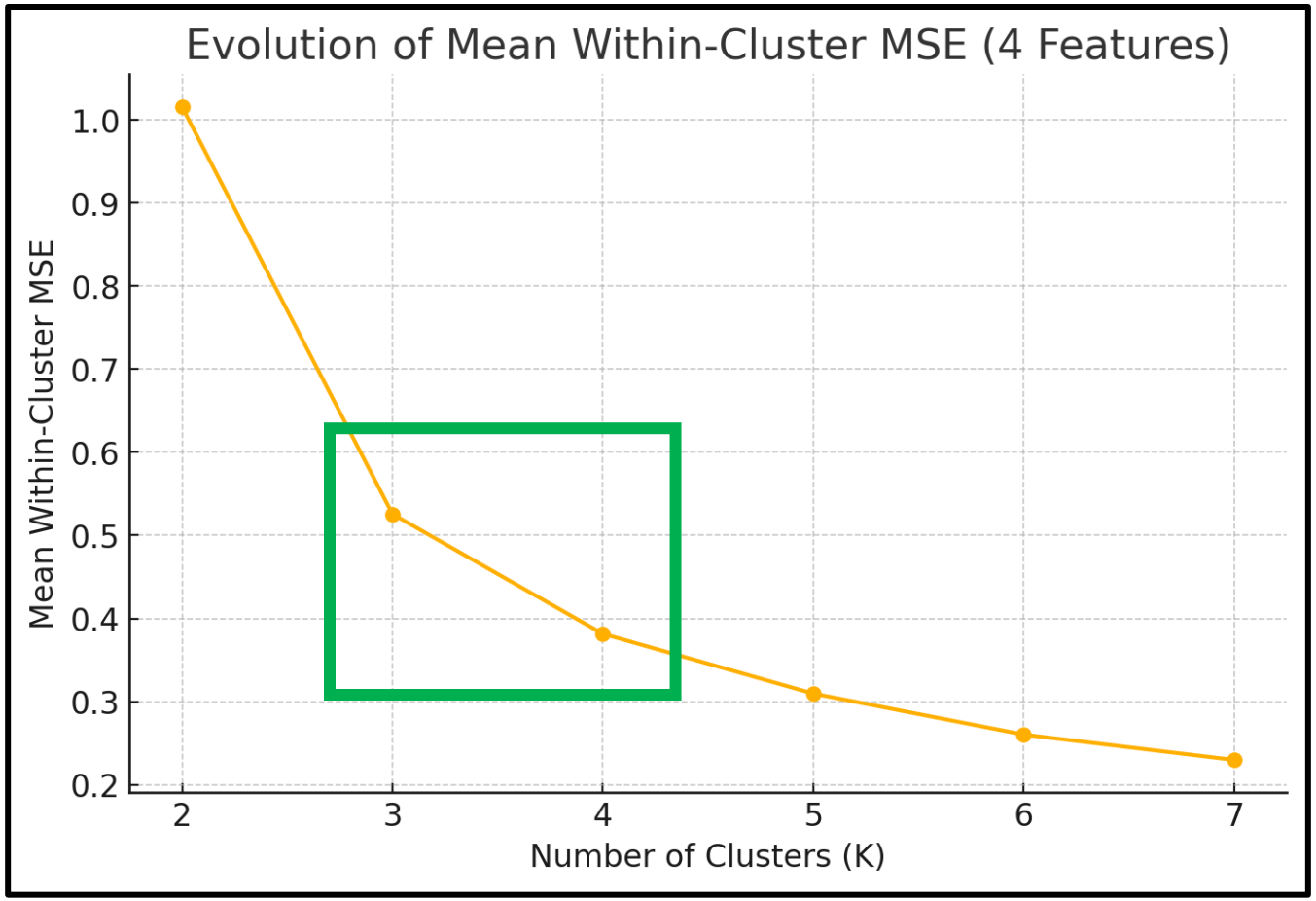
K-means of Iris dataset



How many clusters? #01 Elbow method

As you increase K:

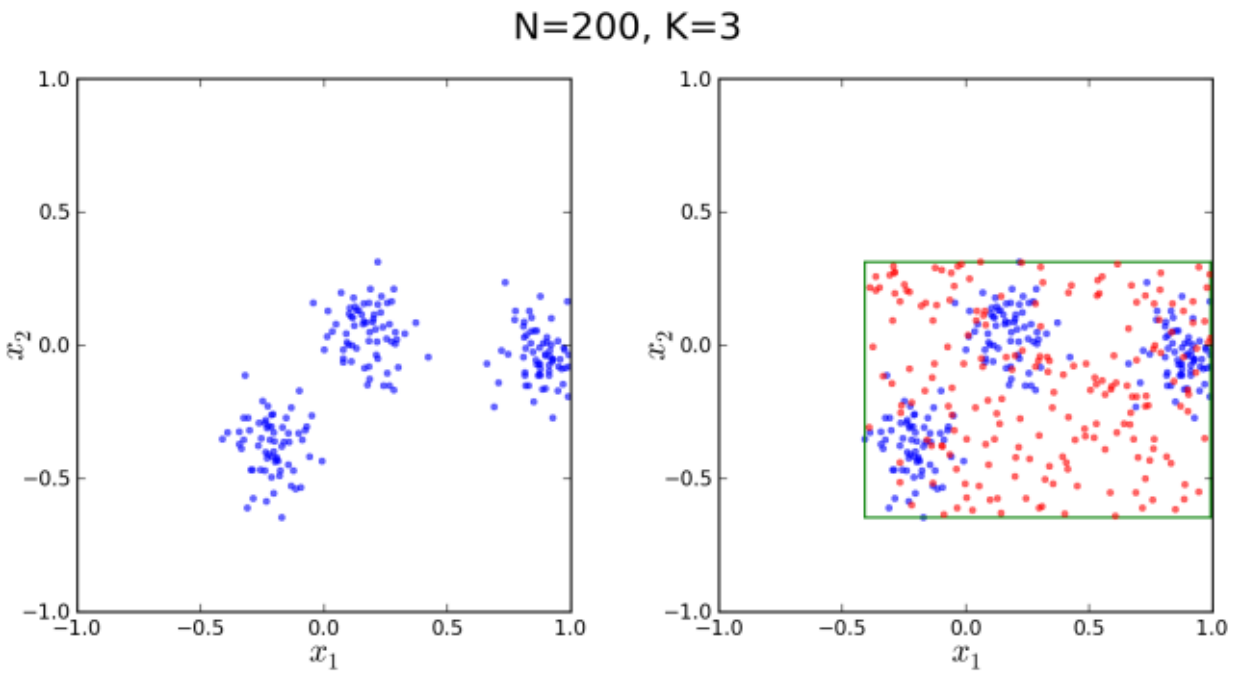
1. The Mean Within-Cluster MSE (or inertia) decreases — because adding more clusters always makes the fit tighter.
2. However, after a certain point, the gain becomes marginal: the clusters stop adding real value and just overfit the data.
3. The "elbow" point in the MSE vs. K plot is where the improvement starts to level off — like an arm bending at the elbow.



How many clusters? #02 Gap Statistics

Unlike the elbow method (which relies on visual heuristics), the gap statistic compares your clustering result to what you'd expect from random data.

"Is the clustering we observe better than what we would expect if the data had no real structure?"



1. For each K (e.g., from 1 to 10):
   - Run K-Means on your real data and compute the **within-cluster dispersion** W_k:

$$W_k = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - \mu_i\|^2$$

2. Generate **reference datasets**:
   - Create synthetic datasets with the **same shape and bounds** as your original data, but **randomly distributed**.
3. Run K-Means on the **random data** and compute their W_k^{ref}.
4. Compute the **Gap Statistic** for each K:

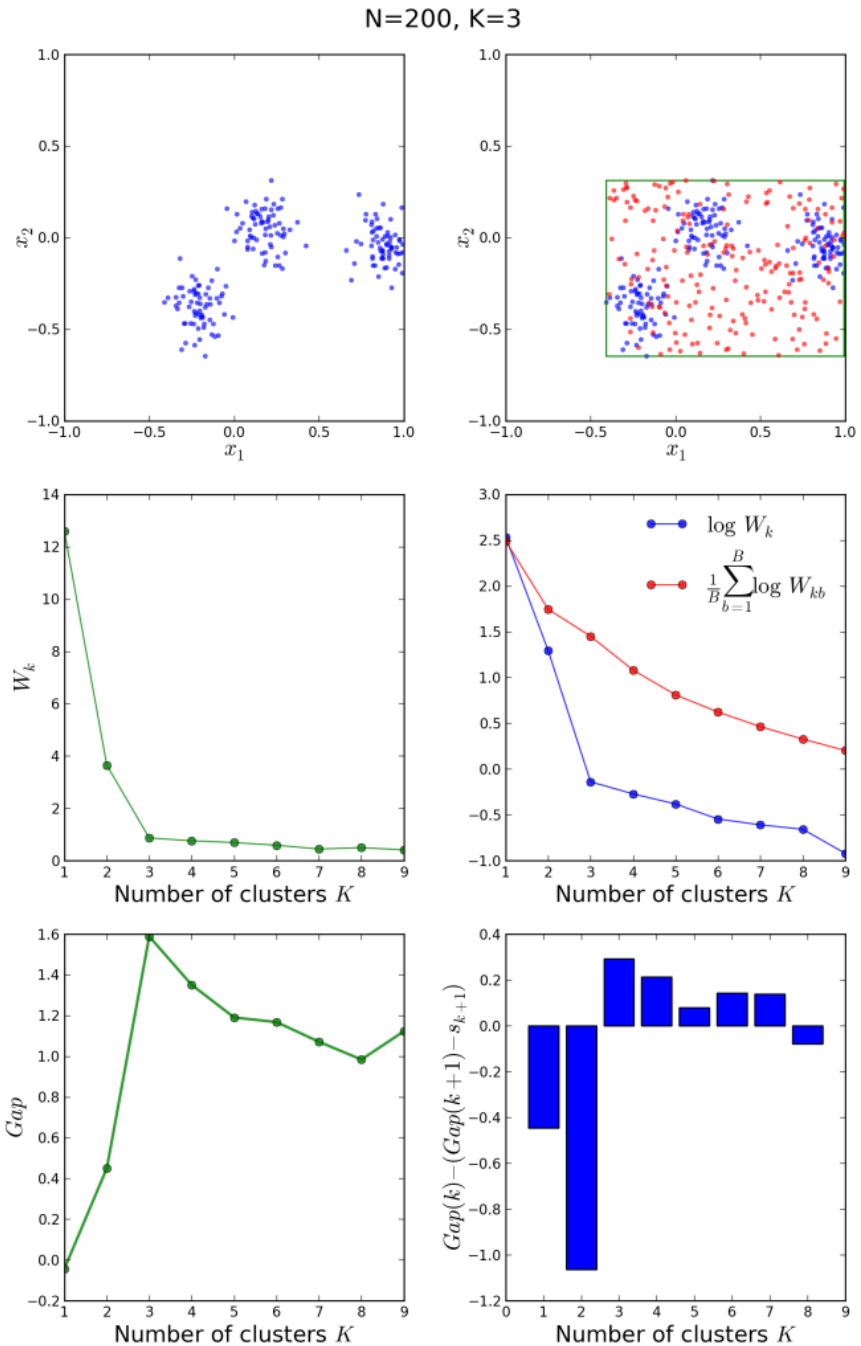
$$\text{Gap}(K) = \mathbb{E}[\log W_k^{\text{ref}}] - \log W_k$$

The higher the gap, the better your clustering is compared to random data.

5. Choose the **smallest** K such that:

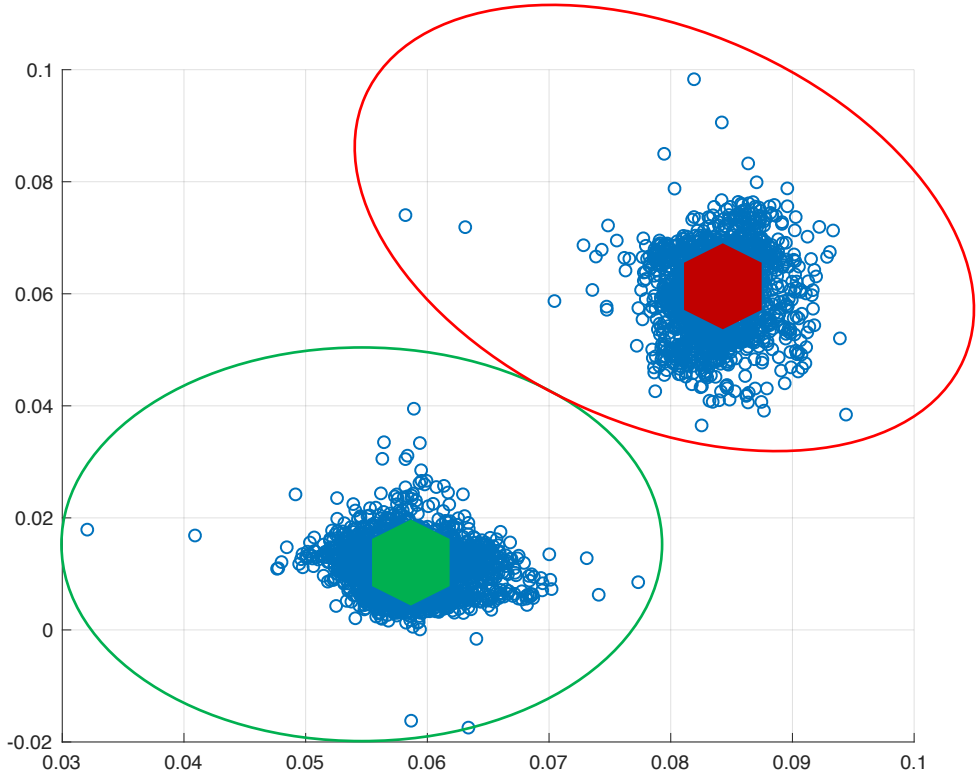
$$\text{Gap}(K) \geq \text{Gap}(K+1) - s_{K+1}$$

where s_{K+1} is the standard deviation of the reference dispersions.



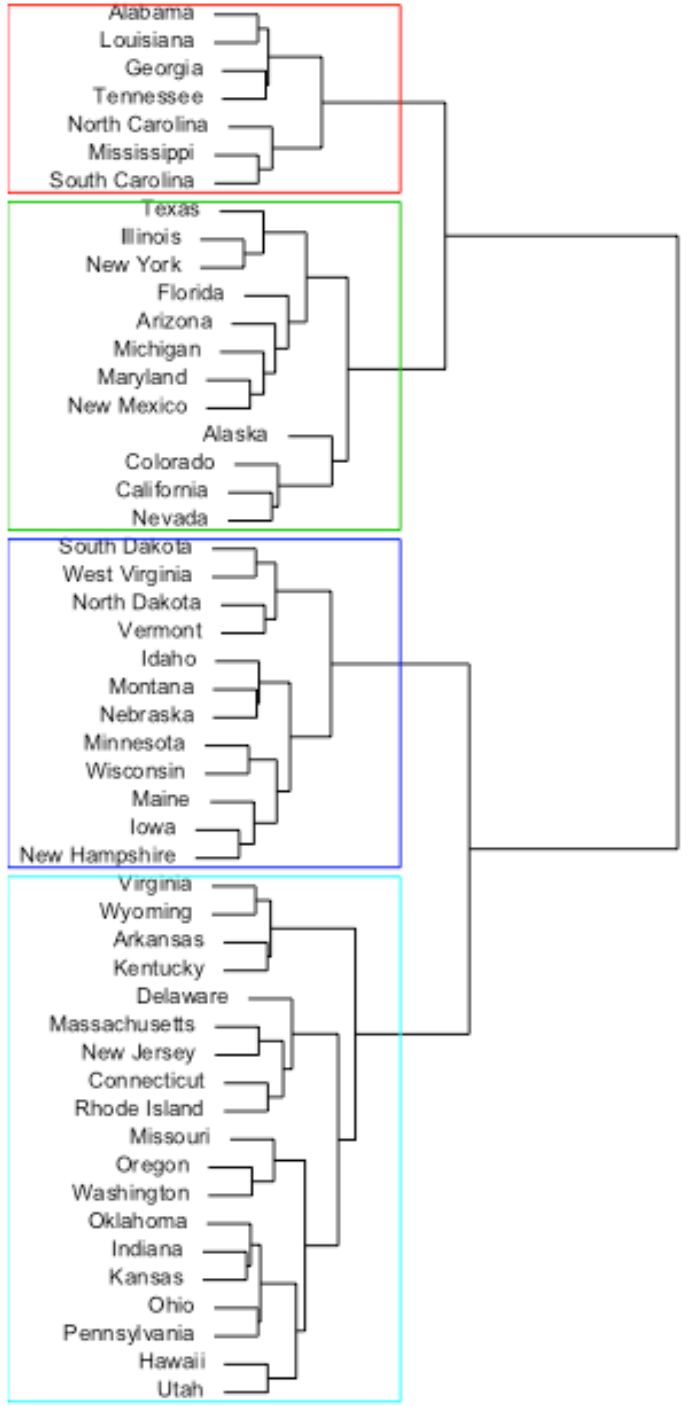
A real-world applications of k-means

- Manufacturing case study seen on Monday
- Once derived the clusters, a classification tool could be put in place



Hierarchical Clustering

- Hierarchical clustering builds a tree of clusters, called a dendrogram, by either:
  - Agglomerative (bottom-up): start with each data point as its own cluster and merge them step by step.
  - Divisive (top-down): start with one big cluster and split it step by step.



Hierarchical Clustering

- Hierarchical clustering builds a tree of clusters, called a dendrogram, by either:
  - Agglomerative (bottom-up): start with each data point as its own cluster and merge them step by step.
  - Divisive (top-down): start with one big cluster and split it step by step.

Agglomerative (bottom-up

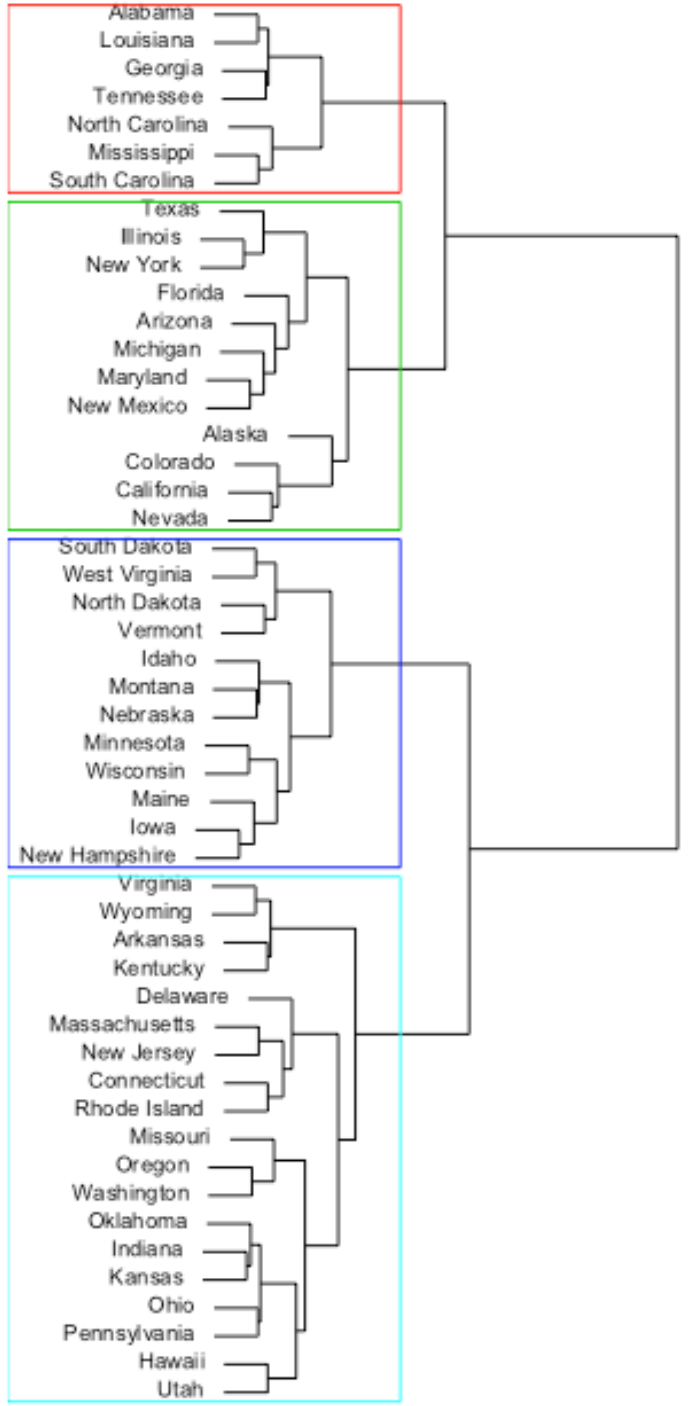
1) Start: Each point is its own cluster.

2) Compute distances between all clusters (initially between all points)

3) Merge the two closest clusters.

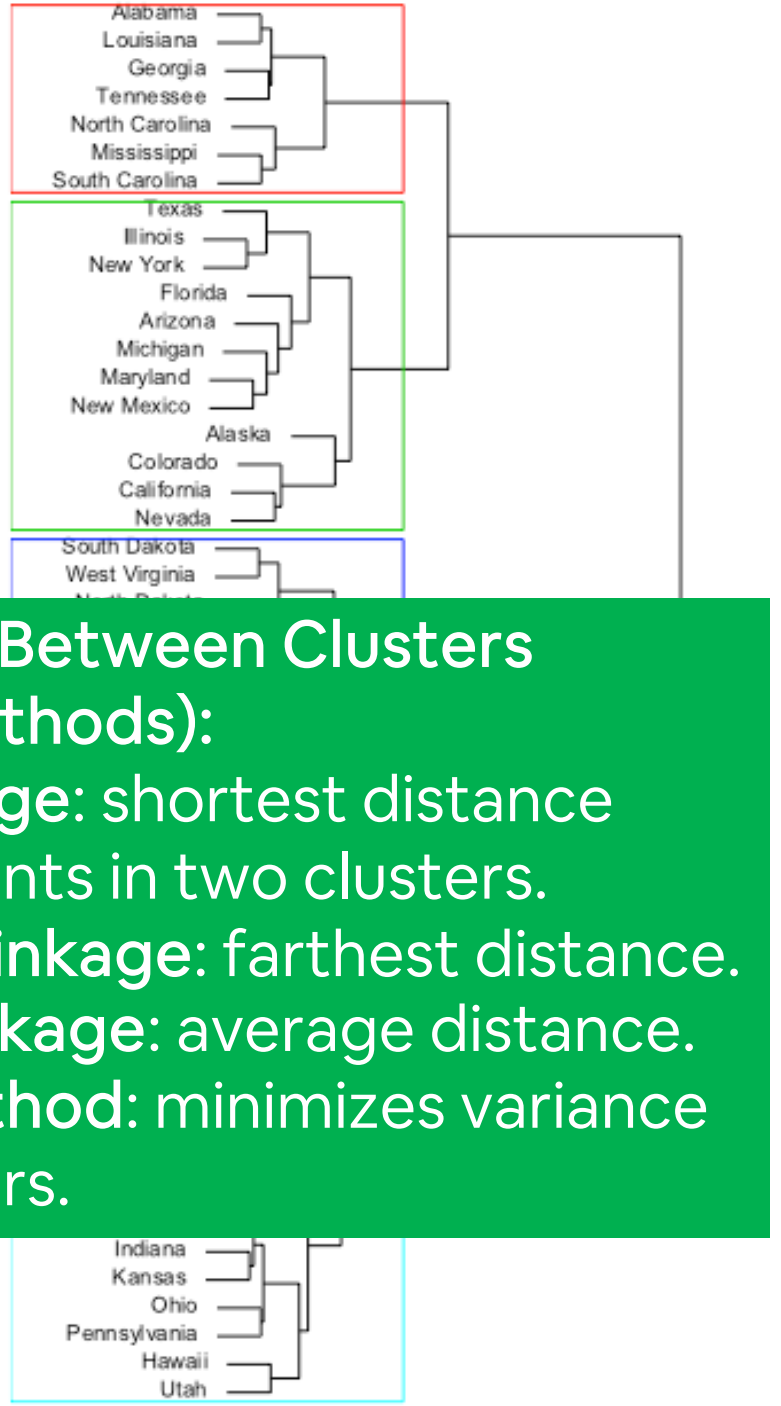
4) Update distances between clusters.

5) Repeat until all points are merged into one big cluster.



Hierarchical Clustering

- Hierarchical clustering builds a tree of clusters, called a dendrogram, by either:
  - Agglomerative (bottom-up): start with each data point as its own cluster and merge them step by step.
  - Divisive (top-down): start with one big cluster and split it step by step.



Agglomerative (bottom-up

1) Start: Each point is its own cluster.

2) Compute distances between all clusters (init between all points)

3) Merge the two closest clusters.

4) Update distances between clusters.

5) Repeat until all points are merged into one big cluster.

📏 Distance Between Clusters (Linkage Methods):

- **Single linkage**: shortest distance between points in two clusters.
- **Complete linkage**: farthest distance.
- **Average linkage**: average distance.
- **Ward's method**: minimizes variance within clusters.

Hierarchical Clustering

- Hierarchical clustering builds a tree of clusters, called a dendrogram, by either:
  - Agglomerative (bottom-up): start with each data point as its own cluster and merge them step by step.
  - Divisive (t
    split it ste

A dendrogram is a tree-like diagram that shows the **merging process**. You can "cut" the tree at a certain level to decide **how many clusters** you want.

Agglomerative

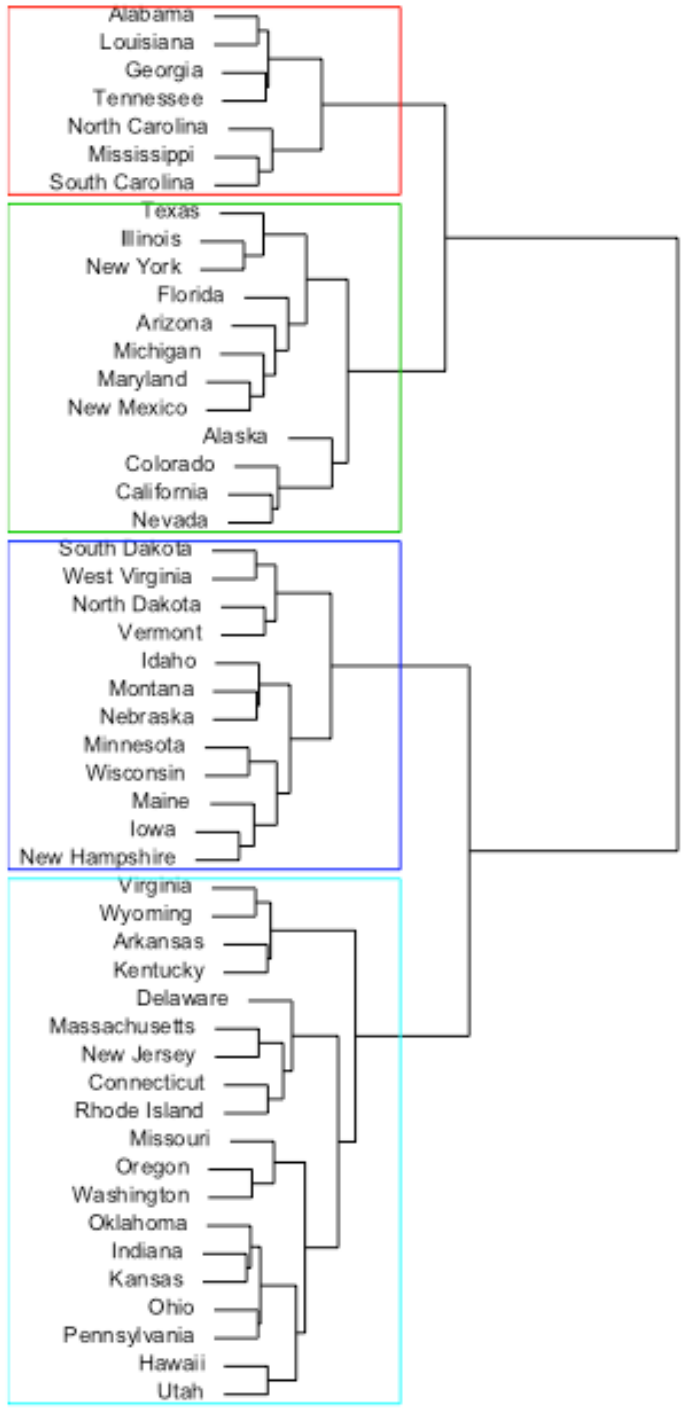
1) Start: Each p

2) Compute distances between all clusters (initially between all points)

3) Merge the two closest clusters.

4) Update distances between clusters.

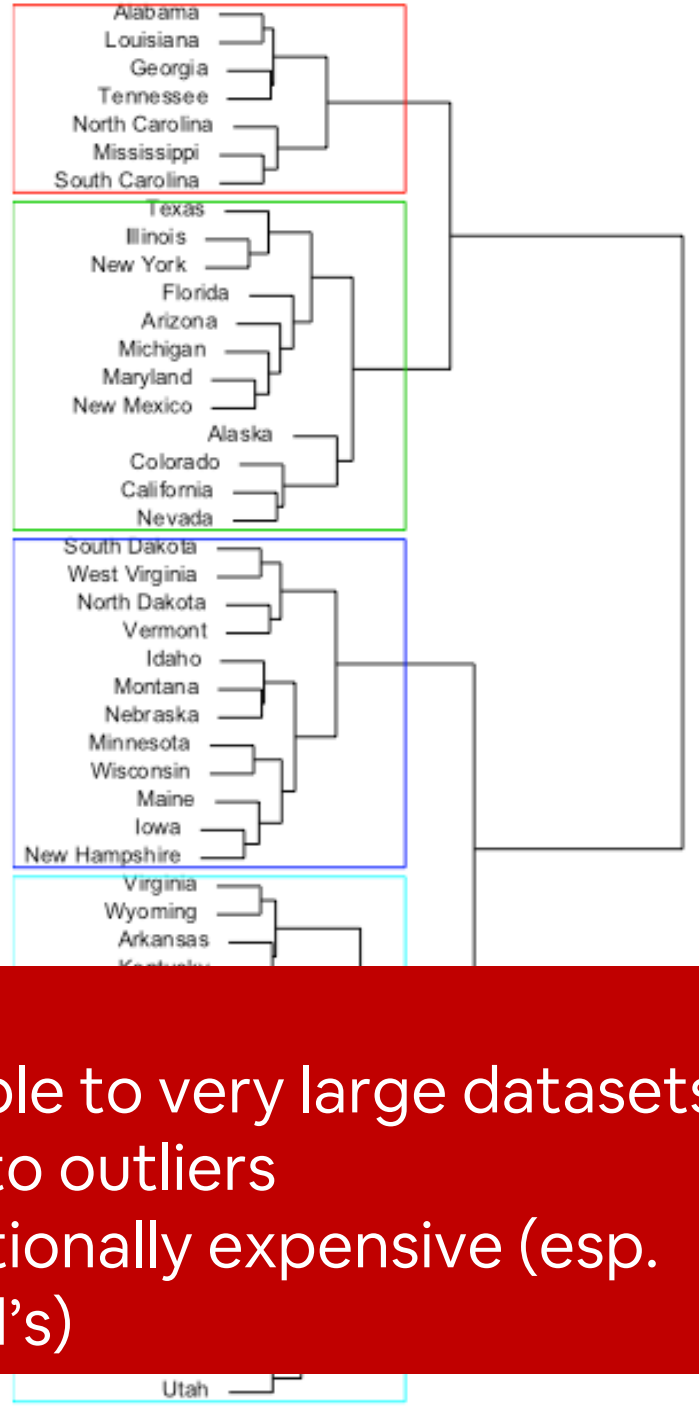
5) Repeat until all points are merged into one big cluster.



Hierarchical Clustering

- Hierarchical clustering builds a tree of clusters, called a dendrogram, by either:
  - Agglomerative (bottom-up): start with each data point as its own cluster and merge them step by step.
  - Divisive (t
    split it ste

A dendrogram is a tree-like diagram that shows the **merging process**. You can "cut" the tree at a certain level to decide **how many clusters** you want.



Agglomerative

1) Start: Each p

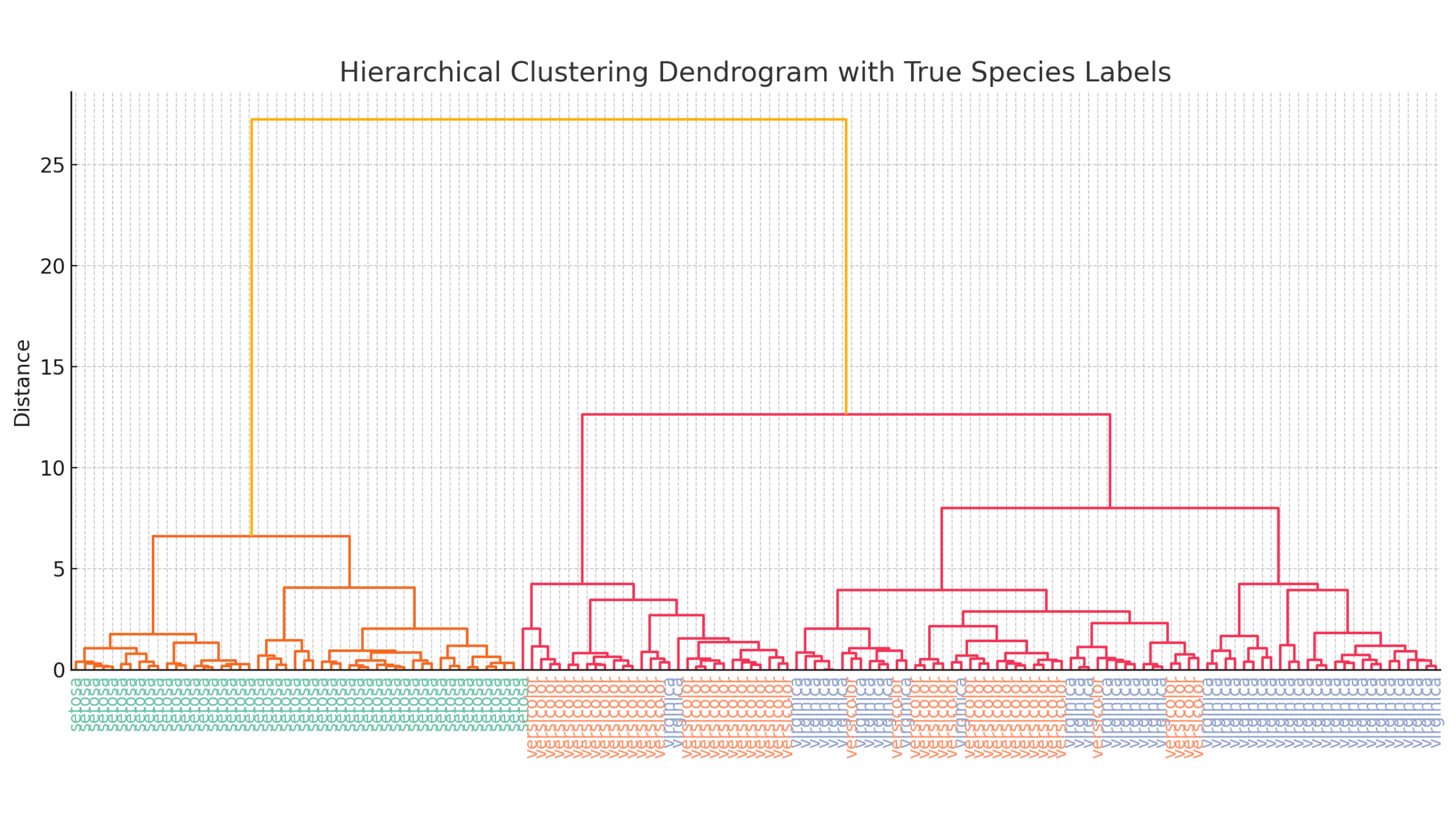
2) Compute distances between all clusters (initi
between all points)

3) Merge the two closest clusters.

4) Update distances between clusters.

5) Repeat until all points are merged into one b

Cons:
- Not scalable to very large datasets
- Sensitive to outliers
- Computationally expensive (esp. with Ward's)


Hierarchical Clustering Dendrogram with True Species Labels
Distance
0
5
10
15
20
25

Thank you!

Gian Antonio Susto