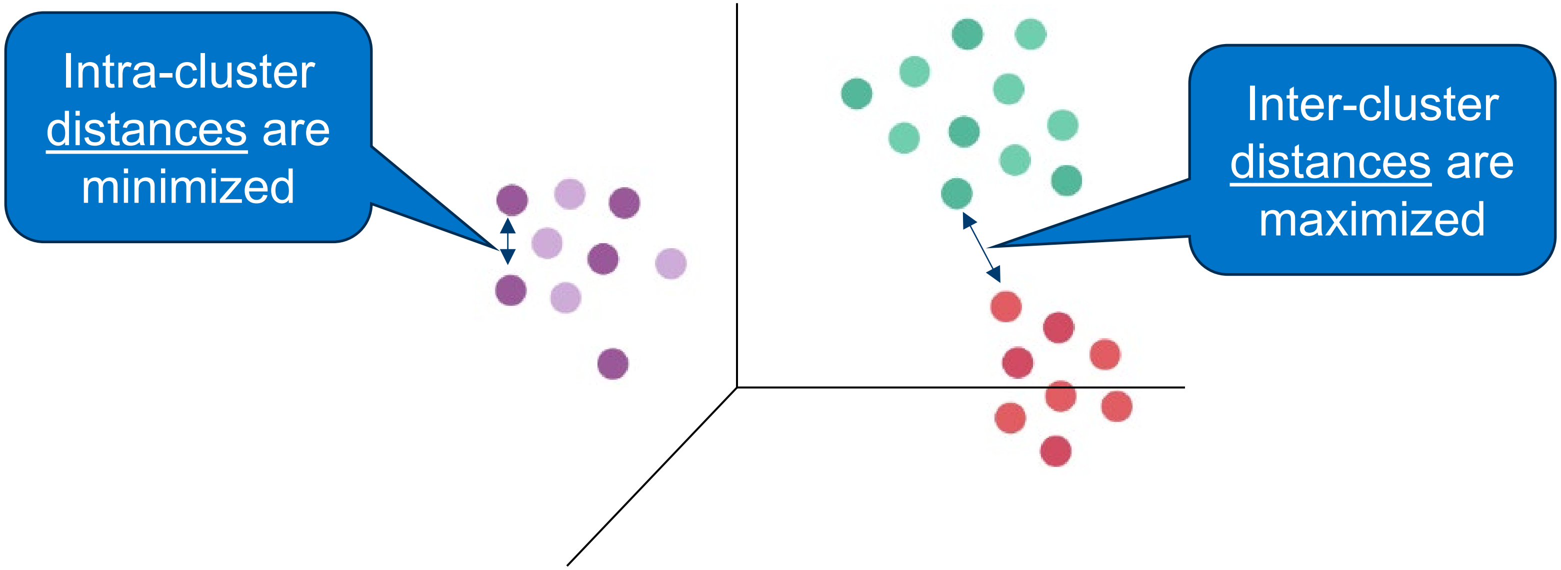
Data Mining for Business Analytics
MSBA 511

Cluster Analysis

What is Cluster Analysis?

Cluster analysis is an unsupervised learning (no labels) technique used to group data points into **clusters** based on certain characteristics.

✓ Maximize the similarity within a cluster and minimize similarity between clusters.

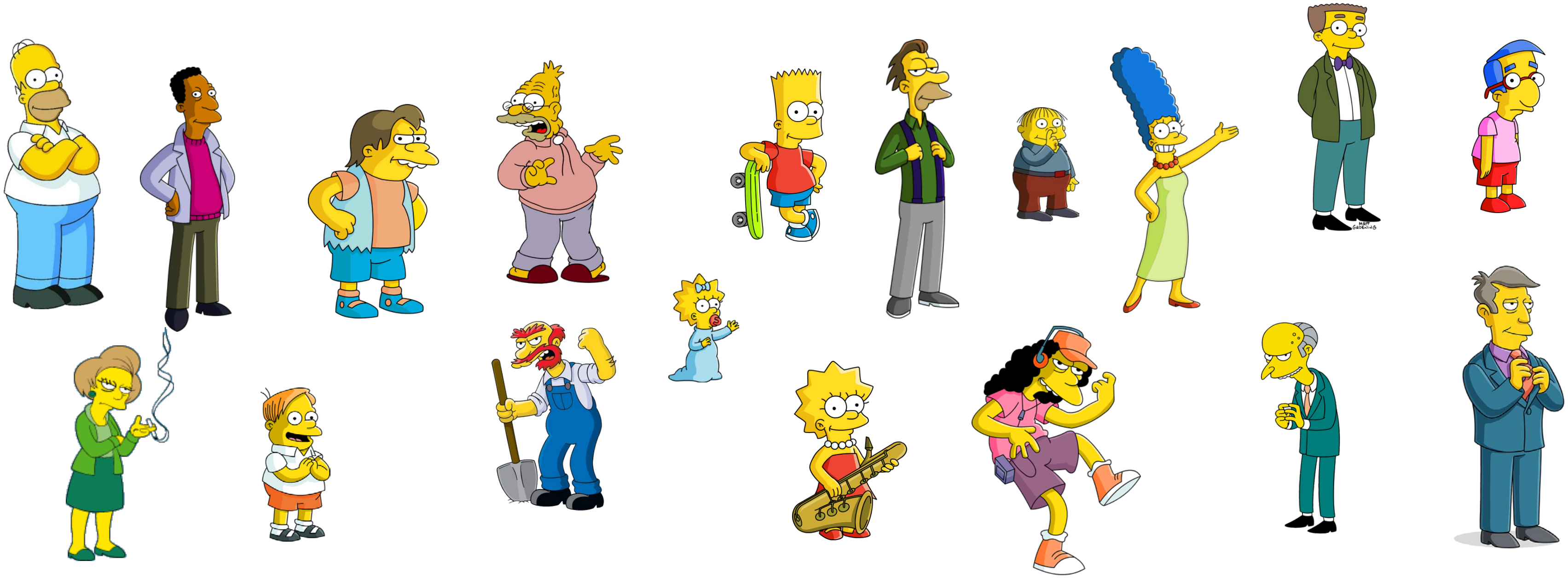


Intra-cluster distances are minimized

Inter-cluster distances are maximized

The Simpsons Example

If you were to apply cluster analysis to the characters from *The Simpsons*, what features would you use to group them into meaningful clusters?



How would you define similarity between the characters?

Applications of Cluster Analysis

There are two broad categories of cluster analysis applications:

1. Uncover hidden patterns, segment similar data points, and identify meaningful groups within a dataset.
2. As a preprocessing step for other algorithms.

Some examples:

- Customer segmentation to tailor marketing strategies to specific demographics and customer traits.
- Anomaly detection (fraud detection, outlier identification)
- Grouping securities in financial portfolios.
- Document categorization (e.g., group similar news articles)

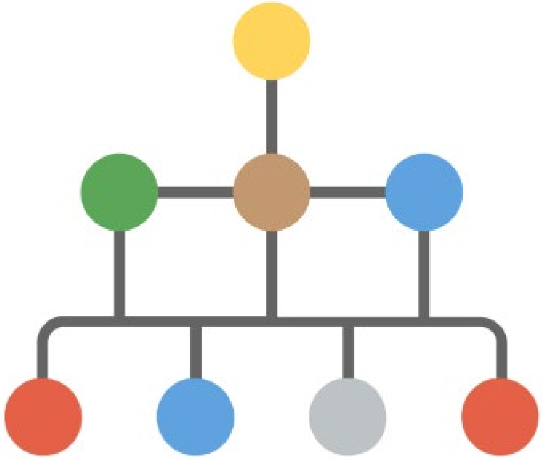
Types of Clustering Methods

Partitional Clustering

Method of dividing a dataset into **non-overlapping groups (clusters)** such that each data point belongs to exactly one cluster.

1. **k-Means**
2. **k-Medoids**

Hierarchical Clustering



Method of organizing data into a **hierarchy of nested clusters**, forming a tree-like structure called a dendrogram.

1. **Agglomerative (Bottom-Up)**
2. **Divisive (Top-Down)**

Types of Clusters

Clusters can be categorized into four main types:

1. **Center-based** – Clusters are defined by **centroids** either the **average** of all the points in the cluster, or a **medoid**, the most "representative" point in the cluster.

2. **Contiguous** – Hierarchical clustering, based on the proximity of data points.

3. **Density-based** – Such as DBSCAN, which identifies clusters as dense regions of points.

4. **Conceptual** – Such as latent class analysis, where clusters are formed based on shared underlying concepts or models).

Clustering Methods Depend on Similarity Measures

The choice of similarity measure directly impacts the cluster shapes, sizes, and overall quality, making it crucial to select a measure that aligns with the data type, scale, and the clustering algorithm used.

Similarity can be expressed in terms of a distance function ***d(x, y)***

Euclidean distance is the most popular and works well for numeric, continuous data in spherical clusters.

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_{i)}{}^{2}}$$

Other **distance measures** include:

- **Correlation** based similarity
- **Cosine similarity** for high-dimensional text data or sparse vectors
- **Manhattan distance** (only horizontal/vertical steps)

Measuring Similarity using Euclidian Distance

Homer

Age: 68

Income: $50,000

# of credit cards: 5

Mr. Burns

Age: 97

Income: $1,000,000

# of credit cards: 0

$$d(Homer, Mr.Burns) = \sqrt{\sum_{i=1}^{n}(x_i - y_{i)}{}^{2}}$$

$$= \sqrt{(68-97)^2+(50{,}000-1{,}000{,}000)^2+(5-0)^2}$$

$$= \mathbf{950,000}$$

Do you see any issues measuring similarity in this way?

Let's Practice: Euclidian Distance

Homer

Age: 68
Years at Power Plant: 30
Actual Work Hours: 15

Mr. Burns

Age: 97
Years at Power Plant: 36
Actual Work Hours: 0

$$d(Homer, Mr. Burns) = \sqrt{\sum_{i=1}^{n} (x_i - y_{i)}{}^2}$$

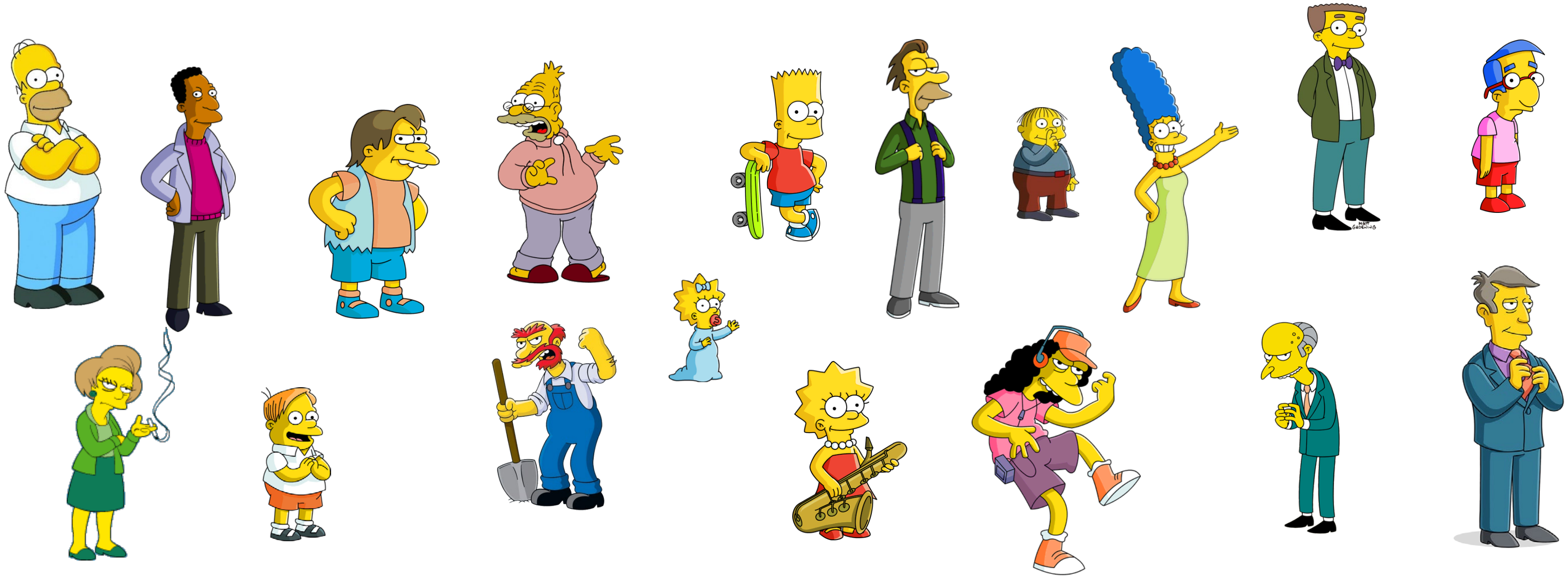
$$= \sqrt{(68-97)^2+(30-36)^2+(15-0)^2}$$

$$= \mathbf{33}$$

From one perspective, Homer and Mr. Burns are very similar!

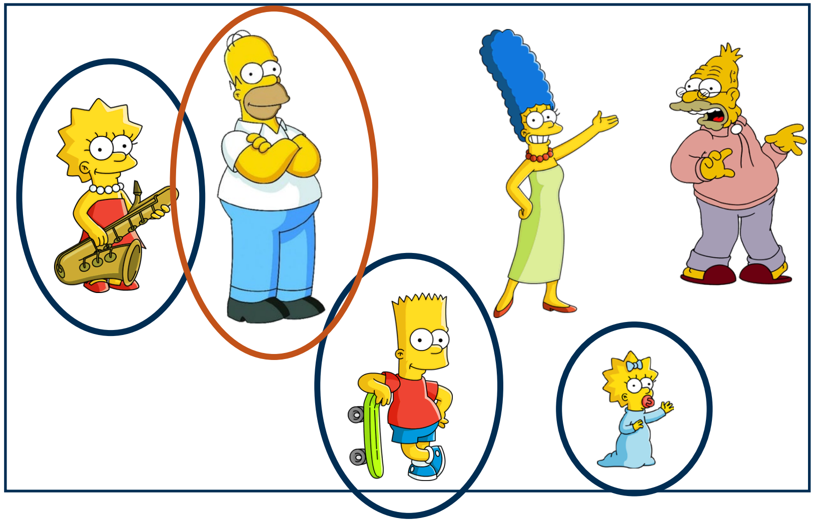
Clustering is Subjective

Cluster analysis is subjective because the type of clustering method used, similarity measure, parameters such as number of clusters, and the nature of the data used can lead to different conclusions.

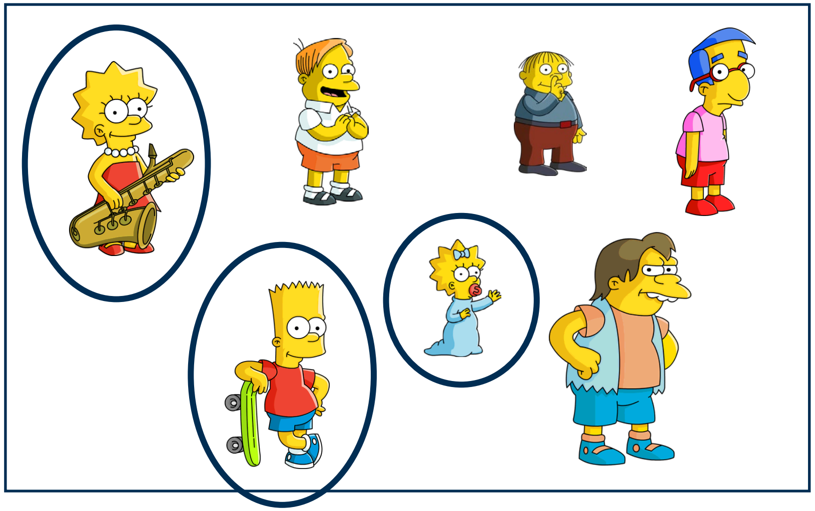


Clustering is Subjective

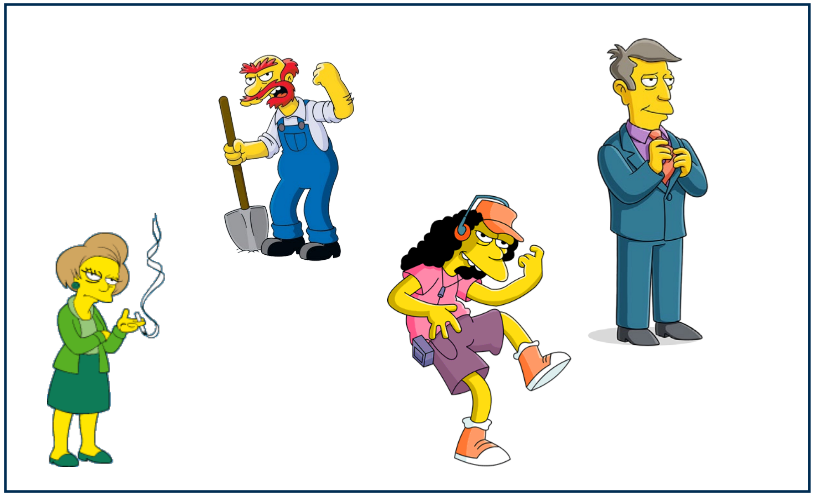
Family-oriented



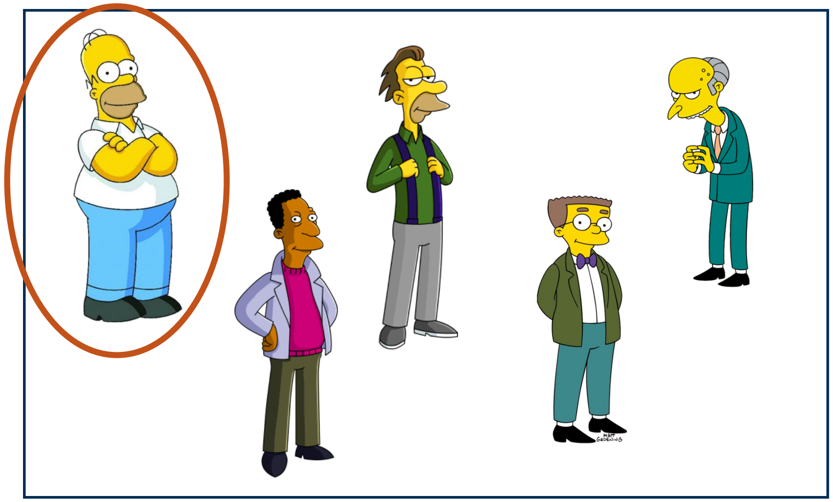
Kids



Springfield Elementary Employees

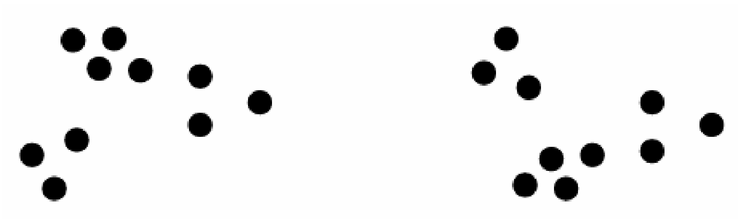


Power Plant Employees

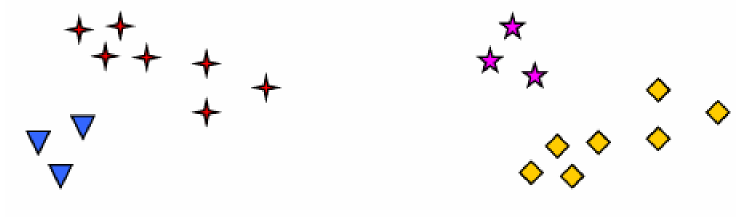


Clustering is also Ambiguous

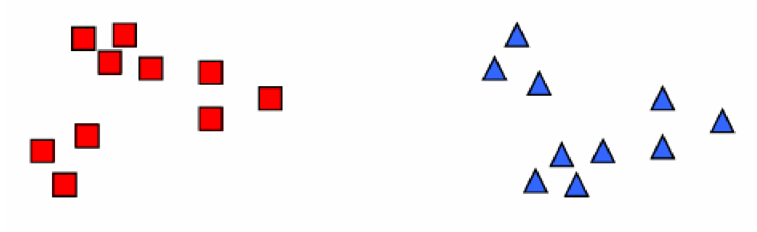
Cluster analysis is also ambiguous because the same data can yield different results depending on the algorithm, initialization, and settings. There is no definitive "correct" clustering solution to determine which result best represents the underlying structure of the data.



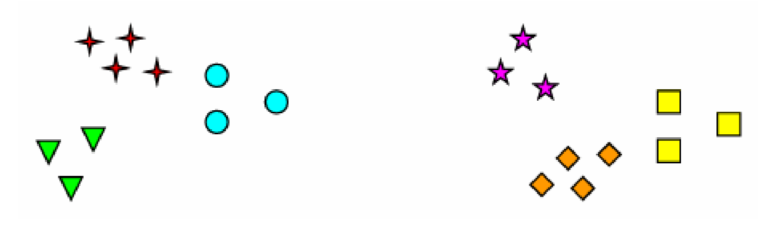
How many clusters?



Four clusters?



Two clusters?



Six clusters?

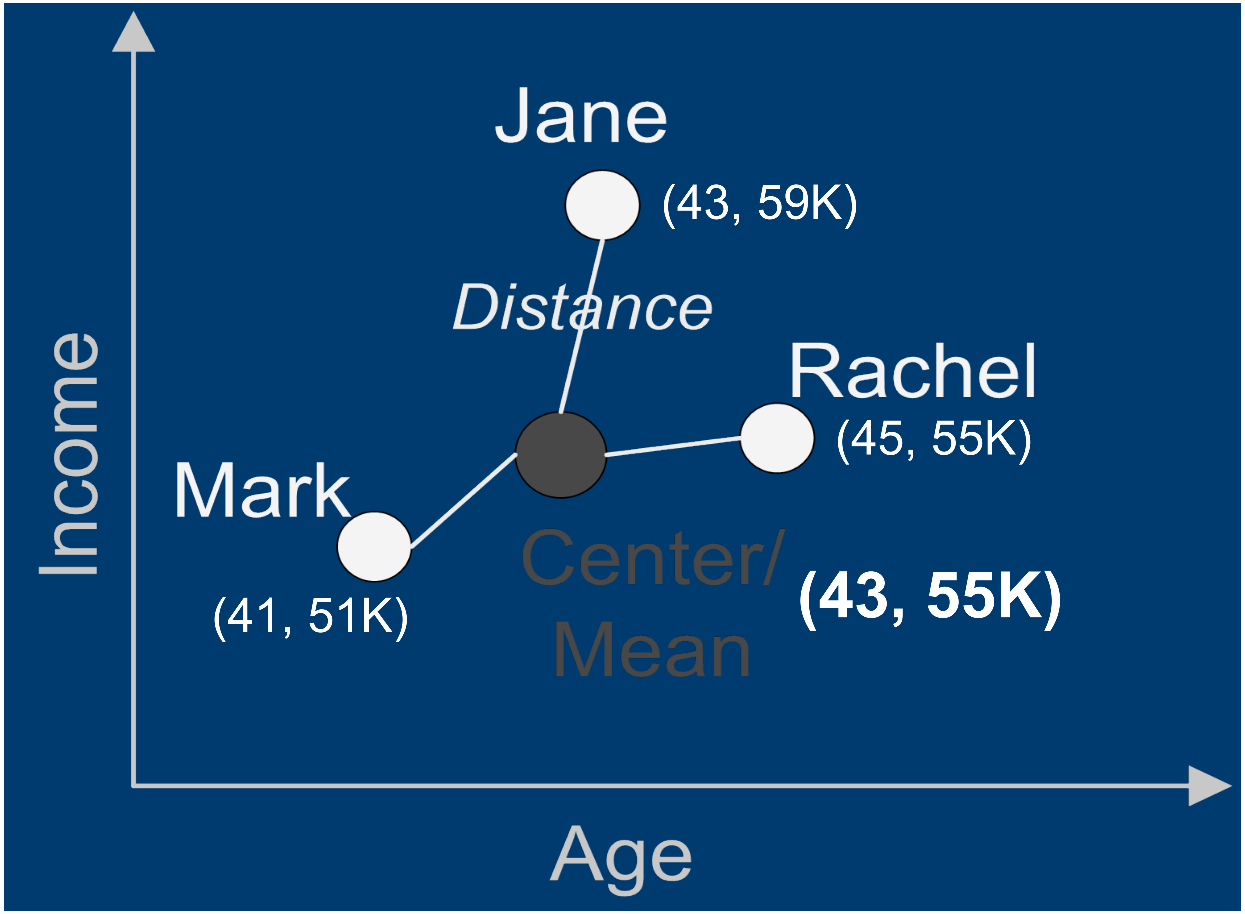
Partitioning Approach

In our class, we are going to focus on partitioning clustering methods. The idea is to **partition** the data points *D* of *n* objects into *k* clusters.

Given k, find a partition of k clusters that optimizes the chosen partitioning criterion:

- **Global optimal**: Exhaustively enumerator all partitions (computationally challenging)
- **Heuristic methods**: k-Means and k-Medoids algorithms
  - k-Means – Each cluster is represented by the cluster center (mean)
  - k-Medoids – Each cluster is represented by one of the objects in the cluster

Computing the Centroid (Mean)



Cluster center:

- Age: (41+43+45)/3=43
- Income: (51K+59K+55K)/3=55K

k-Means Clustering Steps

Given k, the k-Means algorithm is implemented in the following steps:

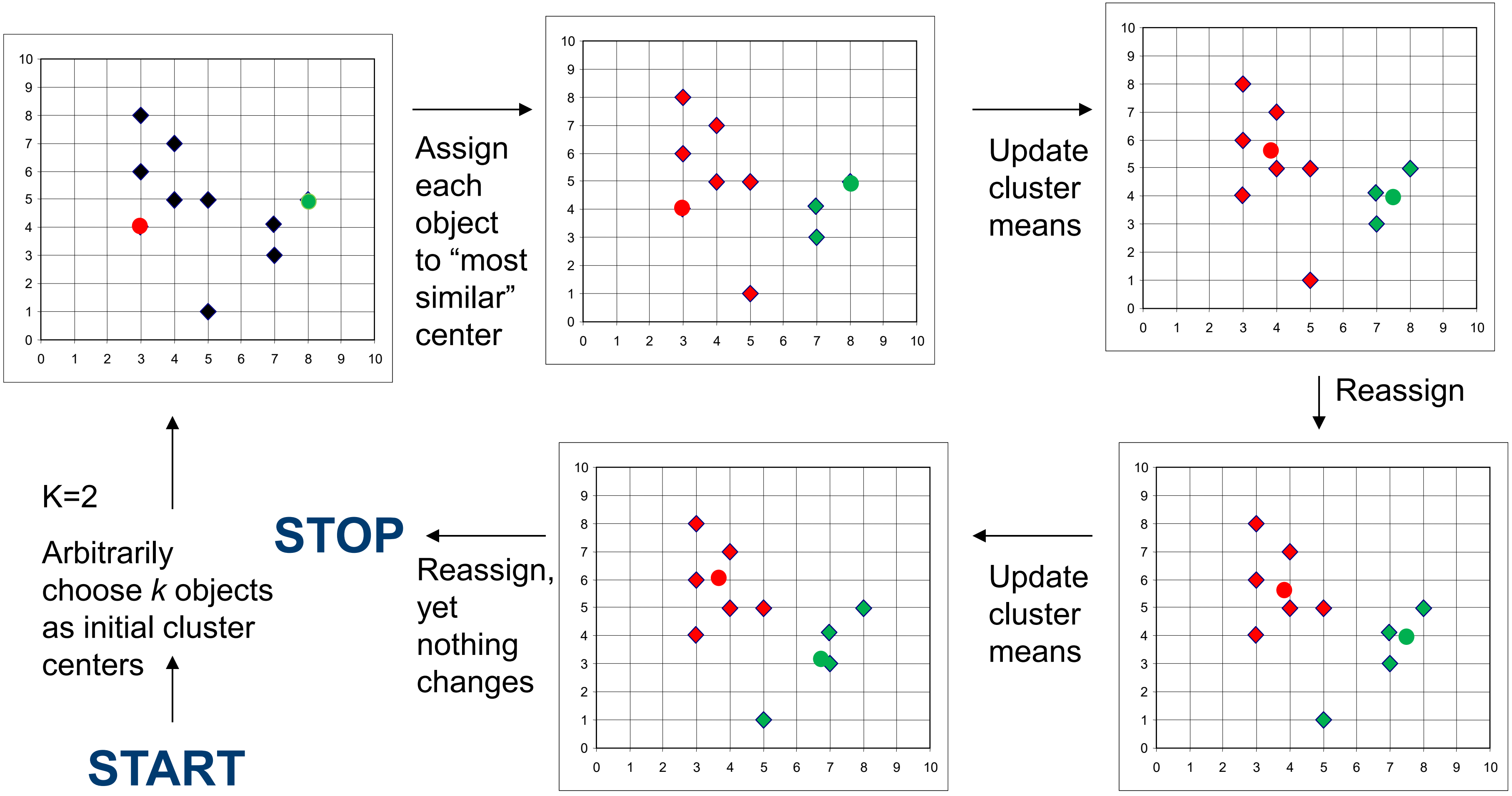
1. Start with a partition into k clusters often based on **random** selection of k initial centroids

2. Assign each record to cluster with closest centroid measured by **Euclidean distance**

3. Recompute centroids (mean of the points in the cluster), repeat step 2

4. Stop when cluster assignment is unchanged

Visualizing the k-Means Steps



K=2

Arbitrarily choose *k* objects as initial cluster centers

START

Assign each object to “most similar” center

Update cluster means

Reassign

Update cluster means

Reassign, yet nothing changes

STOP

Normalizing

As we observed earlier, one **problem** with raw distance measures is that they are highly influenced by the scale of the measurements. The **solution** is to normalize (standardize) the data first:

There are two common options for normalization:

1. Convert to z-scores by subtracting the mean and dividing by the standard deviation.

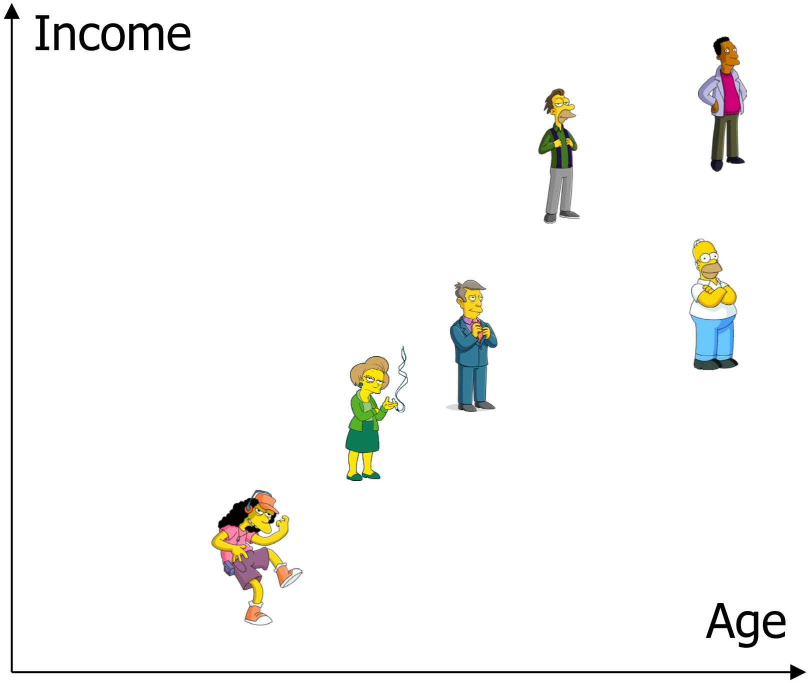
$$z = \frac{x - \mu}{\sigma}$$

2. Rescale to 0-1 range by subtracting the minimum value and then dividing by the range (maximum – minimum)

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

K-Means Clustering Example

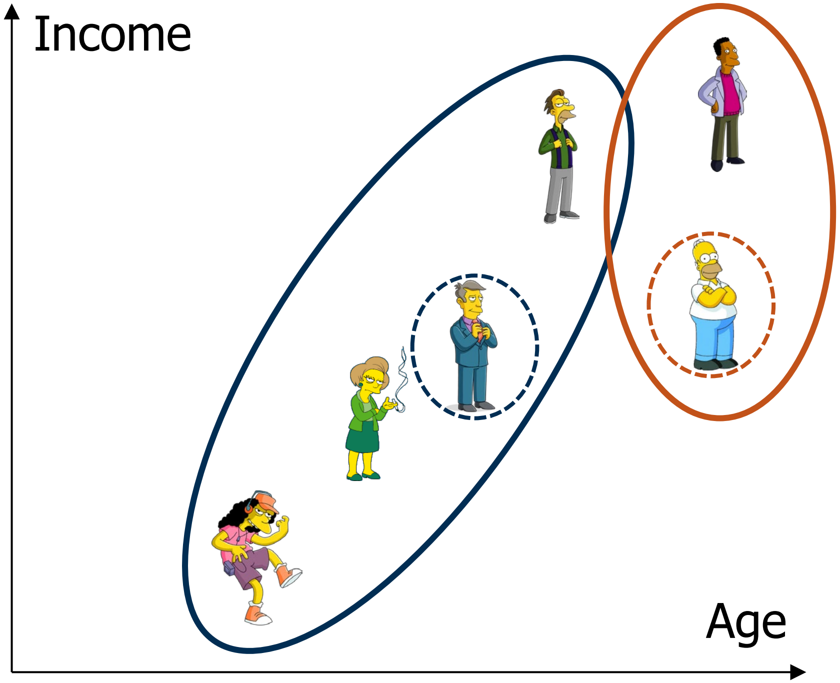
Character	Age	Income
Homer	0.694	0.05
Seymour	0.50	0.04
Lenny	0.582	0.054
Carl	0.714	0.062
Edna	0.418	0.035
Otto	0.296	0.026



K-Means Clustering Example

Character	**Distance from Seymour**	**Distance from Homer**
Homer		
Seymour		
Lenny	**0.08**	0.11
Carl	0.22	**0.05**
Edna	**0.08**	0.28
Otto	**0.20**	0.40

Seymour and **Homer** selected as cluster centers **A** and **B** respectively



K-Means Clustering Example

Calculate cluster centers

Cluster A

Age:

$$\frac{\mathbf{0.50 + 0.582 + 0.418 + 0.296}}{\mathbf{4}} = \mathbf{0.449}$$

Income:

$$\frac{\mathbf{0.040 + 0.054 + 0.035 + 0.026}}{\mathbf{4}} = \mathbf{0.03875}$$

Cluster B

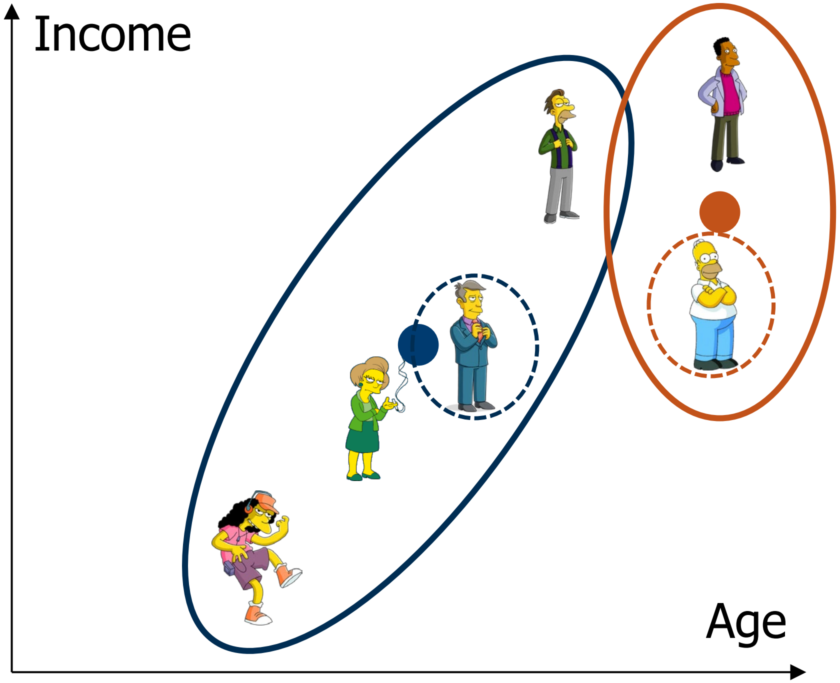
Age:

$$\frac{\mathbf{0.694 + 0.714}}{\mathbf{2}} = \mathbf{0.704}$$

Income:

$$\frac{\mathbf{0.050 + 0.062}}{\mathbf{2}} = \mathbf{0.056}$$

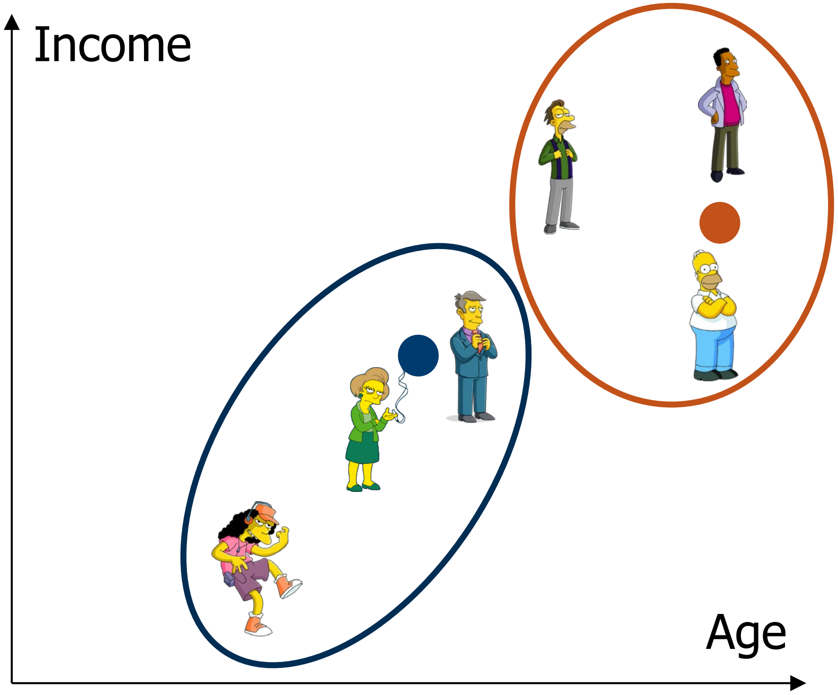
Seymour and **Homer** selected as cluster centers **A** and **B** respectively



K-Means Clustering Example

Character	Distance from A	Distance from B
Homer	0.25	**0.01**
Seymour	**0.05**	0.20
Lenny	0.13	**0.12**
Carl	0.27	**0.01**
Edna	**0.03**	0.29
Otto	**0.15**	0.41

Assign characters to cluster **A** and **B** based on new cluster centers



K-Means Clustering Example

Recalculate cluster centers

Cluster A

Age:

$$\frac{\mathbf{0.50 + 0.418 + 0.296}}{\mathbf{3}} = \mathbf{0.40467}$$

Income:

$$\frac{\mathbf{0.040 + 0.035 + 0.026}}{\mathbf{3}} = \mathbf{0.03367}$$

Cluster B

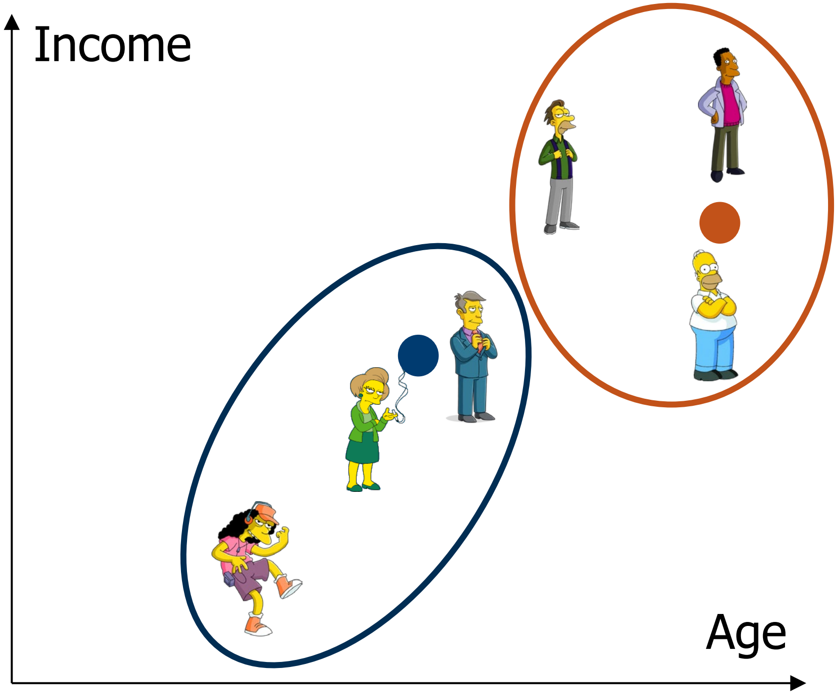
Age:

$$\frac{\mathbf{0.694 + 0.582 + 0.714}}{\mathbf{3}} = \mathbf{0.663}$$

Income:

$$\frac{\mathbf{0.050 + 0.54 + 0.062}}{\mathbf{3}} = \mathbf{0.0553}$$

Assign characters to cluster **A** and **B** based on new cluster centers



Another round would be required to calculate distances for each character to the new cluster centers.

No change -> STOP

k-Means: Strengths

Simplicity and efficiency: Straightforward to implement and computationally efficient, ideal for large datasets with numerical features.

Scalable: Scales well to larger datasets and is faster compared to many other clustering methods.

Flexible: Works well when clusters are roughly spherical and evenly sized.

Interpretability: Cluster centroids are easy to understand and explain, which is beneficial in a business context.

k-Means: Weaknesses

- **Predefined Number of Clusters**: Need to specify *k* (number of clusters) in advance.
- **Initialization Sensitivity**: Different initial placement of centroids can lead to different results.
- **Outlier Sensitivity**: Very sensitive to outliers, which can skew the centroids and degrade cluster quality.
- **Magnitude Sensitivity**: Requires data to be scaled properly (standardization or normalization).
- **Irregular Clusters**: Not suitable for non-spherical, overlapping, varying density clusters.
- **Distance Metric Dependency**: Relies on Euclidean distance which may not always be suitable, especially for high-dimensional/categorical data.

Complex datasets with varying cluster shapes, densities, or mixed data types, alternatives like DBSCAN, hierarchical clustering, or Gaussian Mixture Models may be more appropriate.

Evaluating k-Means Clusters

Most common measure is Sum of Squared Error (**SSE**)

- For each data point, the error is the distance to the nearest cluster center.

Number of clusters

Centroid of cluster C_i

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} \underline{\|x - \mu_i\|}^2$$

Squared Euclidean distance

Data point in cluster C_i

Choosing k and Initial Partitioning

Choosing k

1. Based on external considerations such as previous knowledge of the data or business objective.
    - ➢ How many market segments do we want?

2. Try different values of k and then evaluate the cluster results for each k value.

Initial Partitioning

1. If domain knowledge suggests a certain initial partitioning, this information should be used.

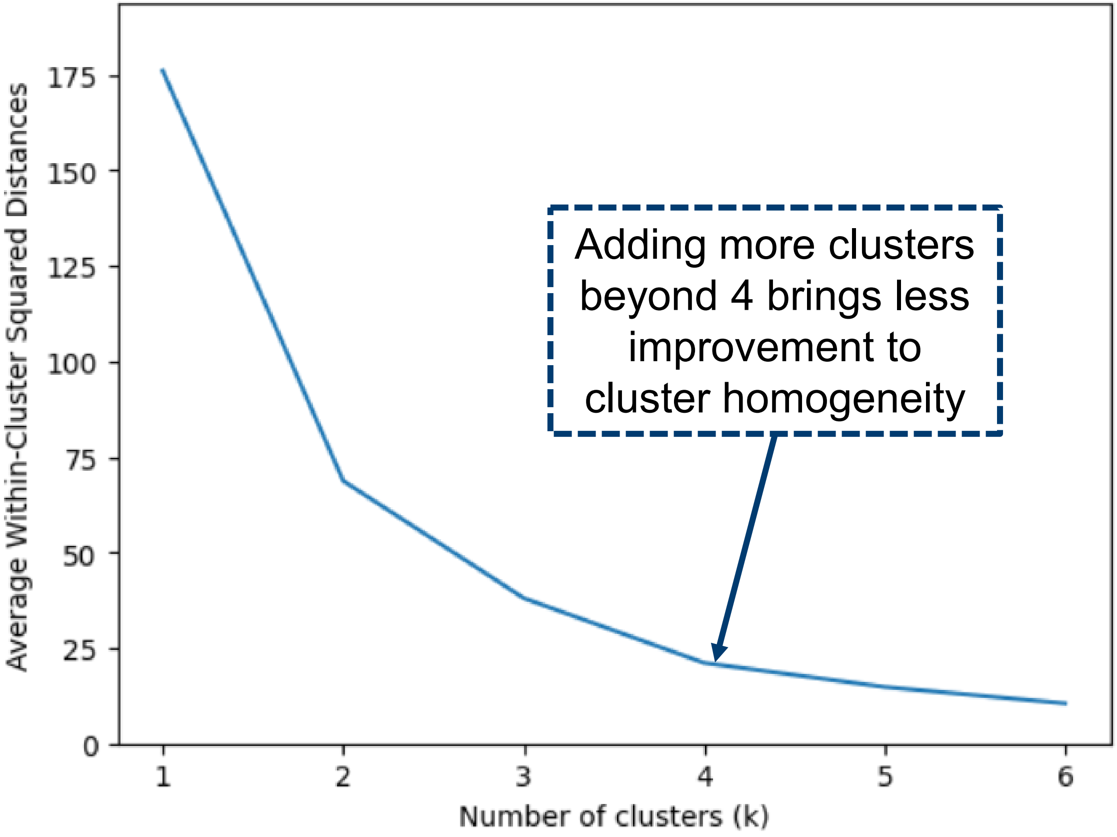
2. Otherwise, a **random** approach is often used.
    - Repeat the process with different random partitions

The k-means++ algorithm selects initial cluster centers more strategically starting with a random point and then choosing subsequent centers with a probability proportional to their squared distance from the nearest existing center.

Evaluating Different k Values

A common approach is referred to the Elbow method that compares the SSE for each cluster against each k value.

An "elbow chart" is a line chart that depicts the decline in cluster heterogeneity as we add more clusters.



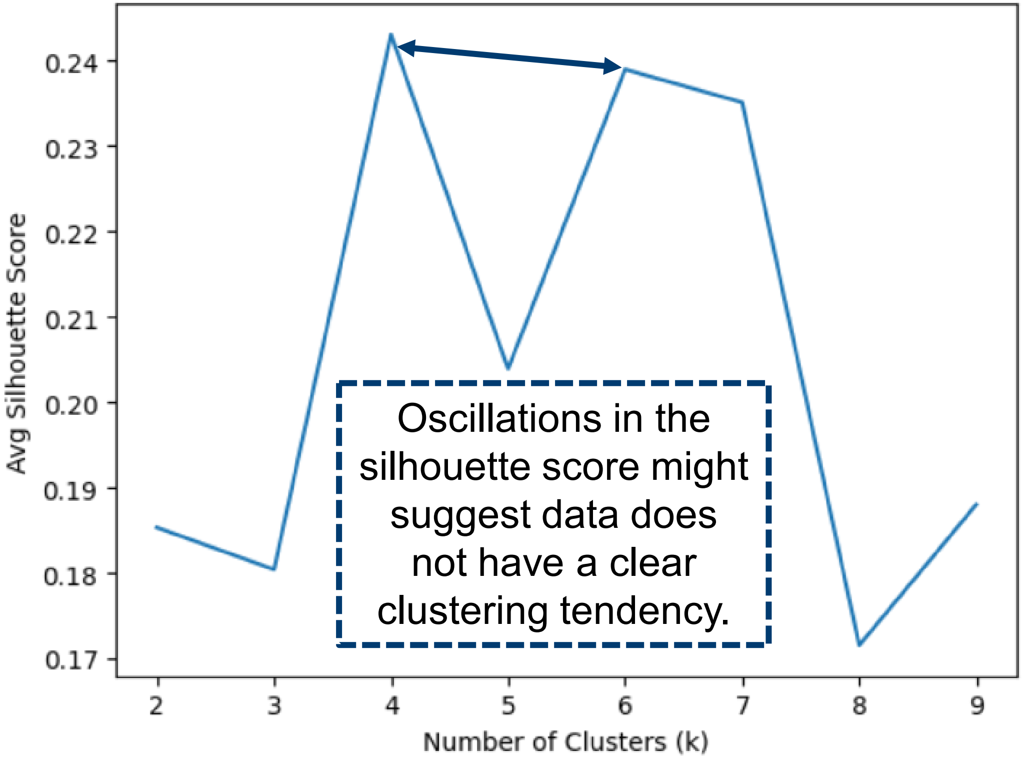
Adding more clusters beyond 4 brings less improvement to cluster homogeneity

Identifying the "elbow" point can be subjective, as the transition from steep decline to a more gradual slope is not always clear, and different people may choose a different k value.

Another Method: Silhouette Score

The Silhouette Method measures how well each data point x_i "fits" its assigned cluster and how poorly it fits into other clusters.

The silhouette score is defined as: $s(x_i) = \dfrac{b_{x_i} - a_{x_i}}{\max(a_{x_i}, b_{x_i})}$



Oscillations in the silhouette score might suggest data does not have a clear clustering tendency.

a_{x_i} - Average distance from x_i to all other points within its own cluster k.

b_{x_i} - Minimum average distance from x_i to points in a different cluster, minimized over clusters.

Range	Interpretation
0.71 – 1.0	A strong structure has been found.
0.51 – 1.0	A reasonable structure has been found.
0.26 – 0.5	The structure is weak and could be artificial.
< 0.25	No substantial structure has been found.

Silhouette Score Wiki

Profile Plot

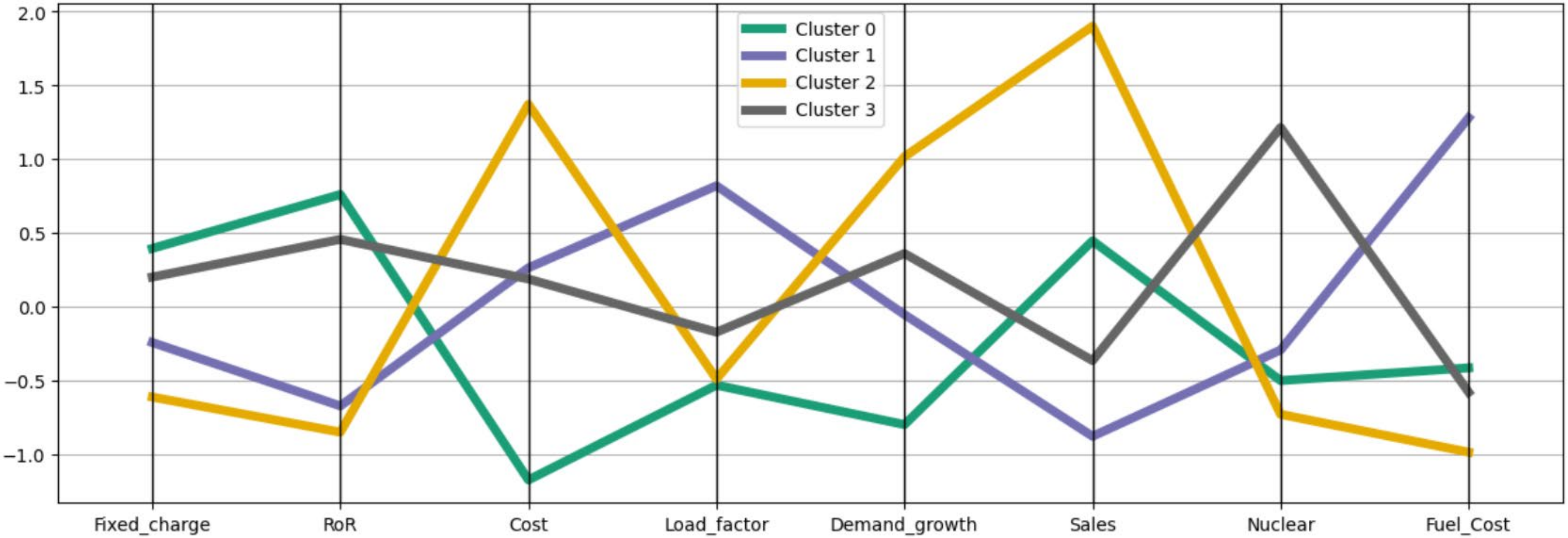
A **profile plot** (or **parallel coordinate plot**) is used to visually compare the characteristics of clusters by plotting their mean values across different variables. It helps:

Cluster Interpretation – Visualize how clusters differ in terms of key variables.

Cluster Validation – Verify if clusters make sense in the context of the data and domain knowledge.

Feature Importance – Highlights the features that are driving the cluster formation.

Anomaly Detection – Identify Unusual cluster behaviors or outliers if a cluster shows an unexpected profile.



Meaningful Clusters

A very important goal of cluster analysis is to make sure the results yield **meaningful clusters** that can be used to make actionable insights.

Cluster Interpretability

- Clusters should have distinct and logical characteristics.
  - ➢ Attempt to label the clusters by assigning a name or implied meaning

Cluster Stability

- Results should be consistent when re-running the analysis.

Cluster Separation

- Examine the inter and intra-cluster distances to see if separation is reasonable.

Number of Clusters

- The number of resulting clusters must be useful, given the purpose of the analysis.