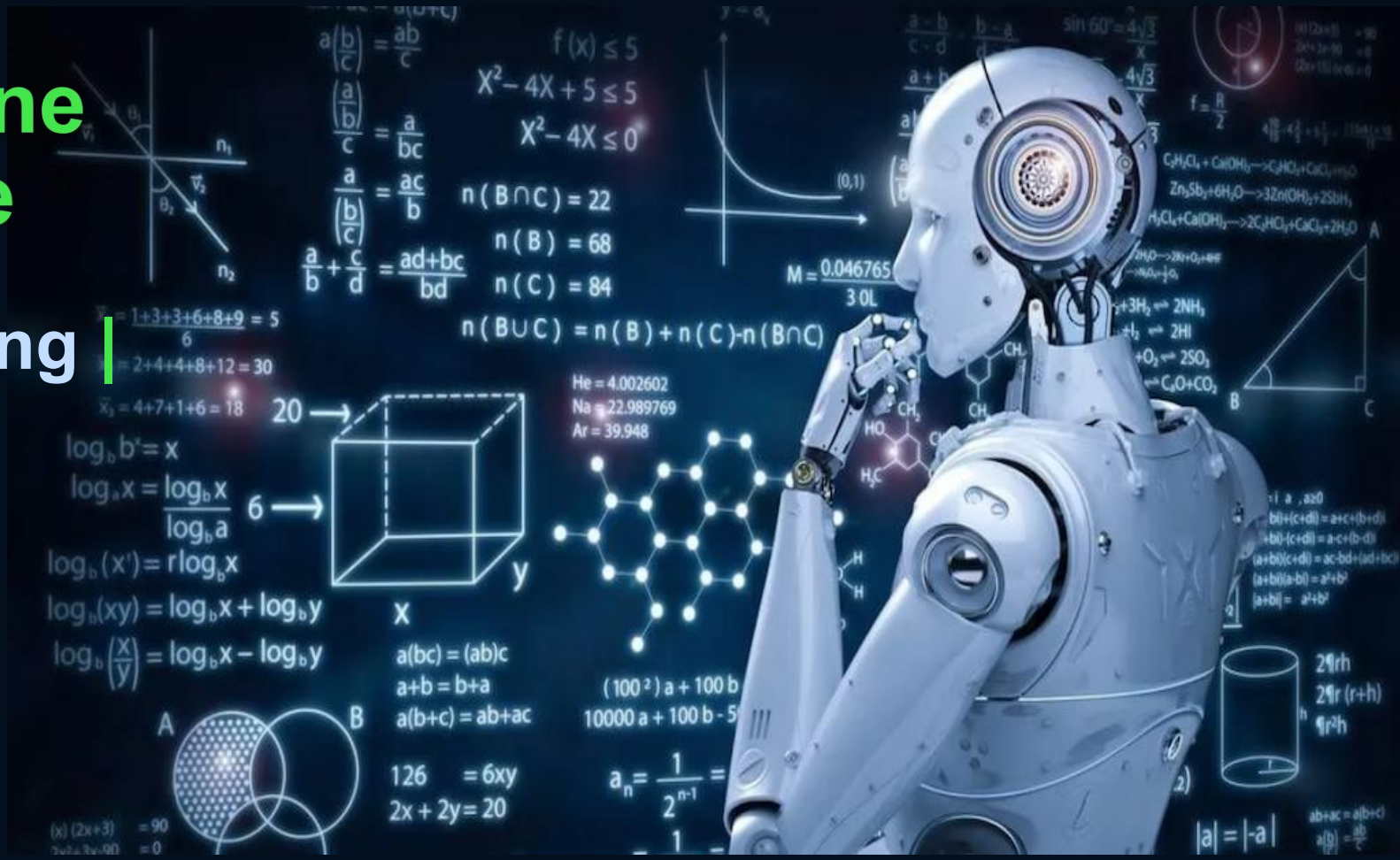
NBS Nottingham Business School

Principles of Machine Learning in Finance

5. Unsupervised Learning |

Clustering |

K-Means Model



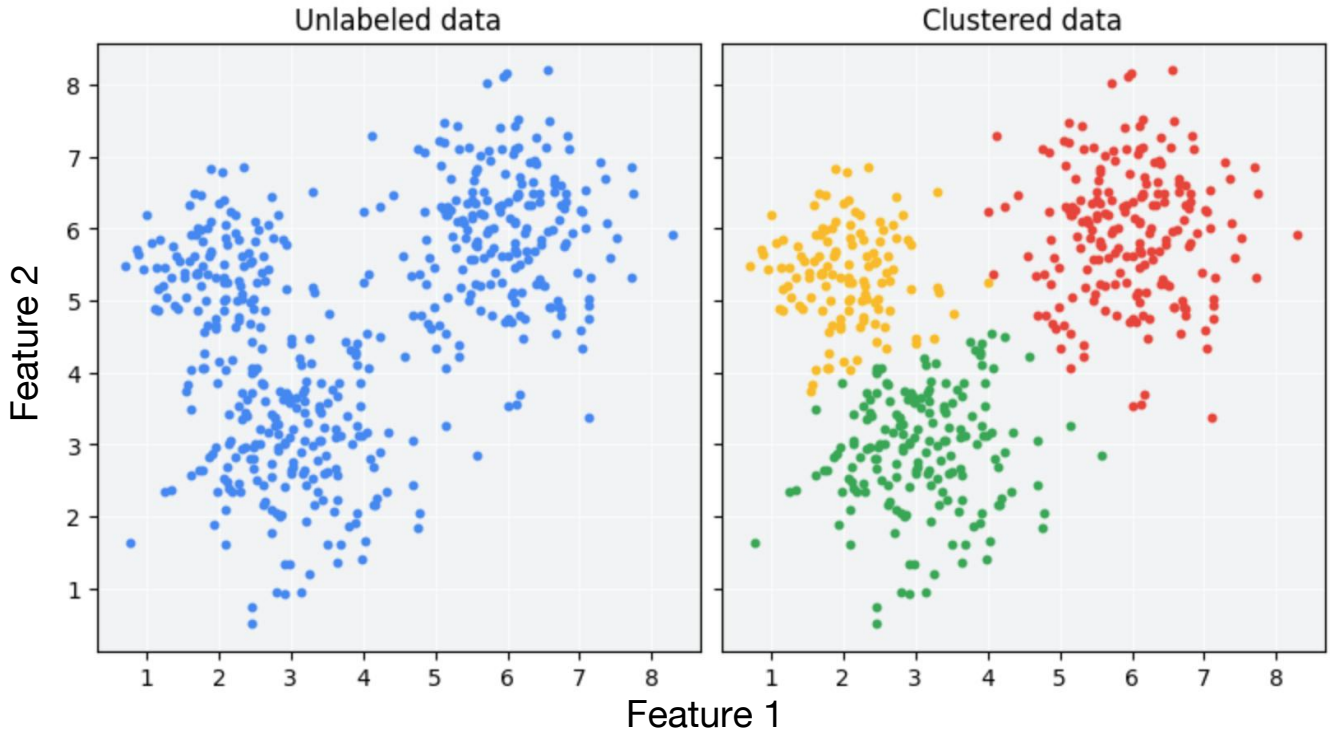
Learning Outcomes

- Clustering and K-Means Model
- Inertia and Silhouette Score
- Local Minima
- Elbow Curve
- **Coding Activity 5-1**: Unsupervised ML. Clustering. K-Means || [K-Means Model for Colour Compression with Python (Non-Synthetic Data)]
- **Coding Activity 5-2**: Unsupervised ML. Clustering. K-Means || [K-Means: Inertia and Silhouette Score with Python (Synthetic Data)]

Clustering

Clustering is an unsupervised machine learning technique designed to group unlabeled data (observations) based on their similarity to each other.

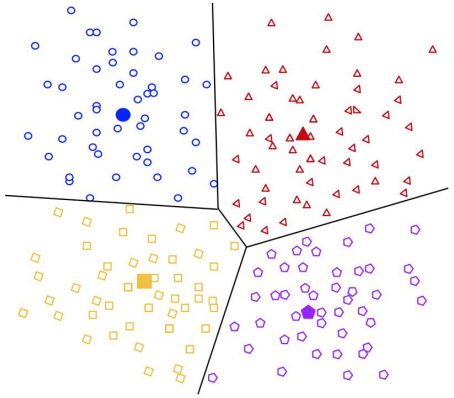
Note: If data is labeled, this kind of grouping is called classification.



Common Approaches to Clustering

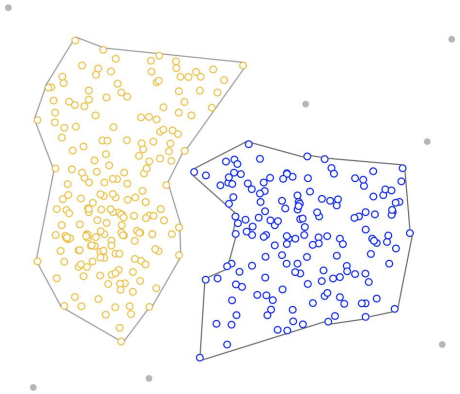
Centroid-based Clustering

Organizes the data into non-hierarchical clusters. Algorithms are efficient but sensitive to initial conditions and outliers.



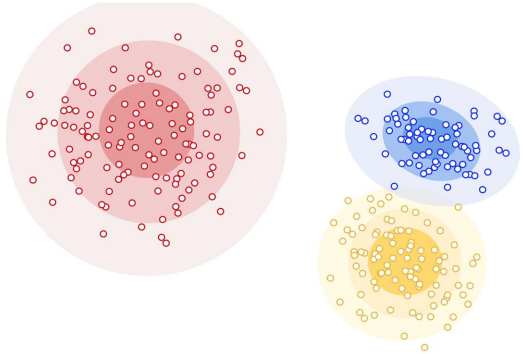
Density-based Clustering

Connects contiguous areas of high example density into clusters. This allows for the discovery of any number of clusters of any shape. Outliers are not assigned to clusters.



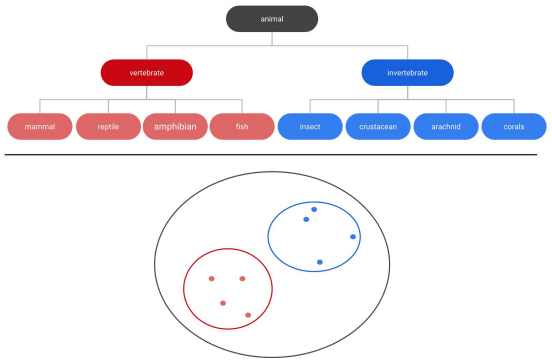
Distribution-based Clustering

Assumes data is composed of probabilistic distributions, such as Gaussian distribution



Hierarchical Clustering

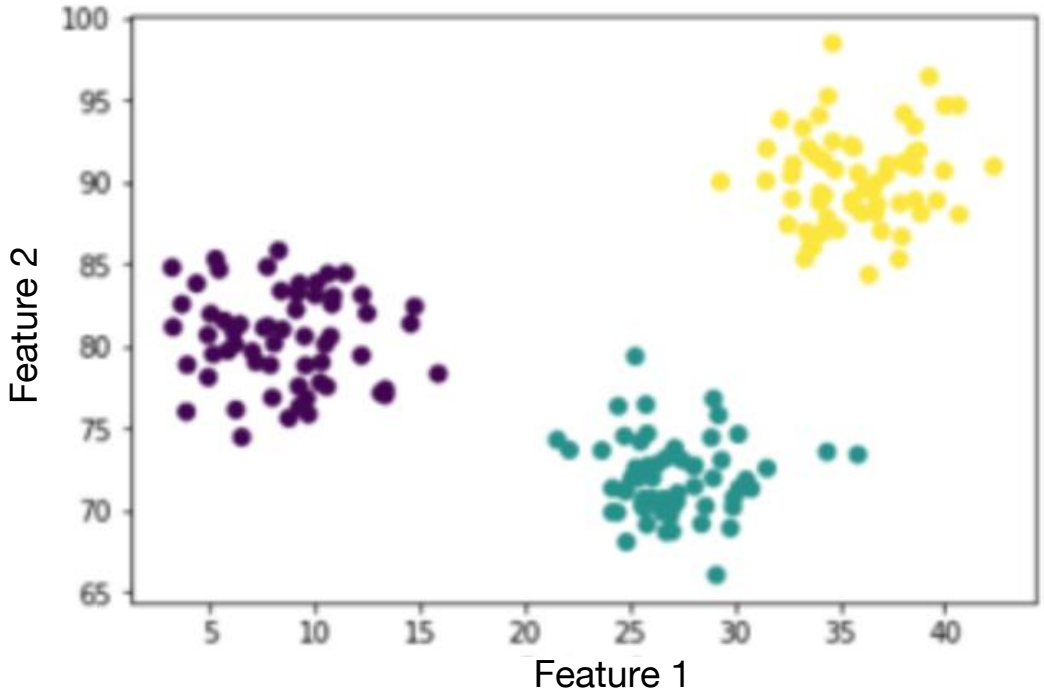
Creates a tree of clusters. Hierarchical clustering is well suited to hierarchical data, such as taxonomies.



K-Means

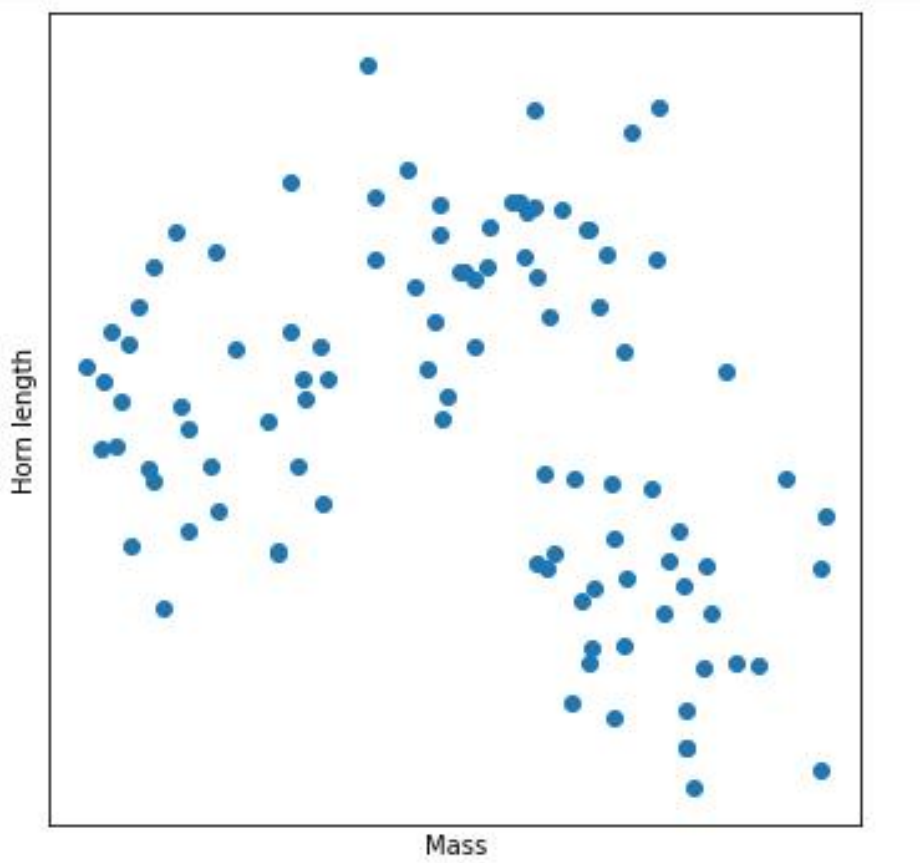
K-Means is an unsupervised machine learning technique used for data clustering, which groups unlabeled data into groups or clusters based on their similarity.

- Unsupervised learning
- Partitioning algorithm
- Clustering of unlabeled data

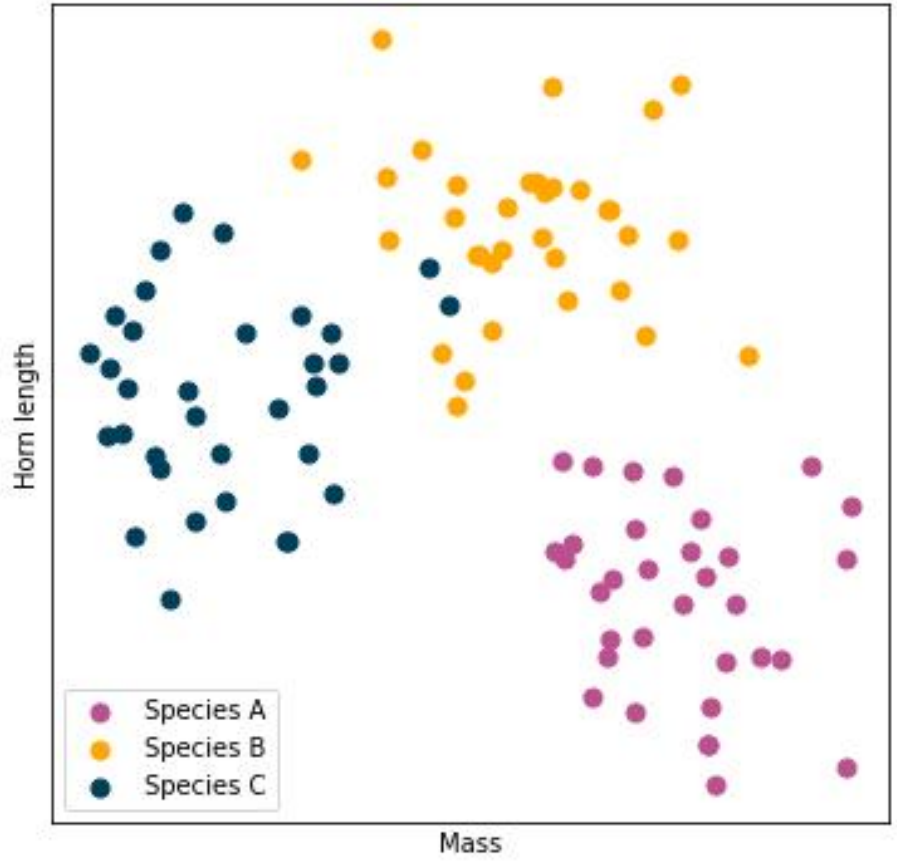


Example 1. K-Means Model

A. Before K-Means

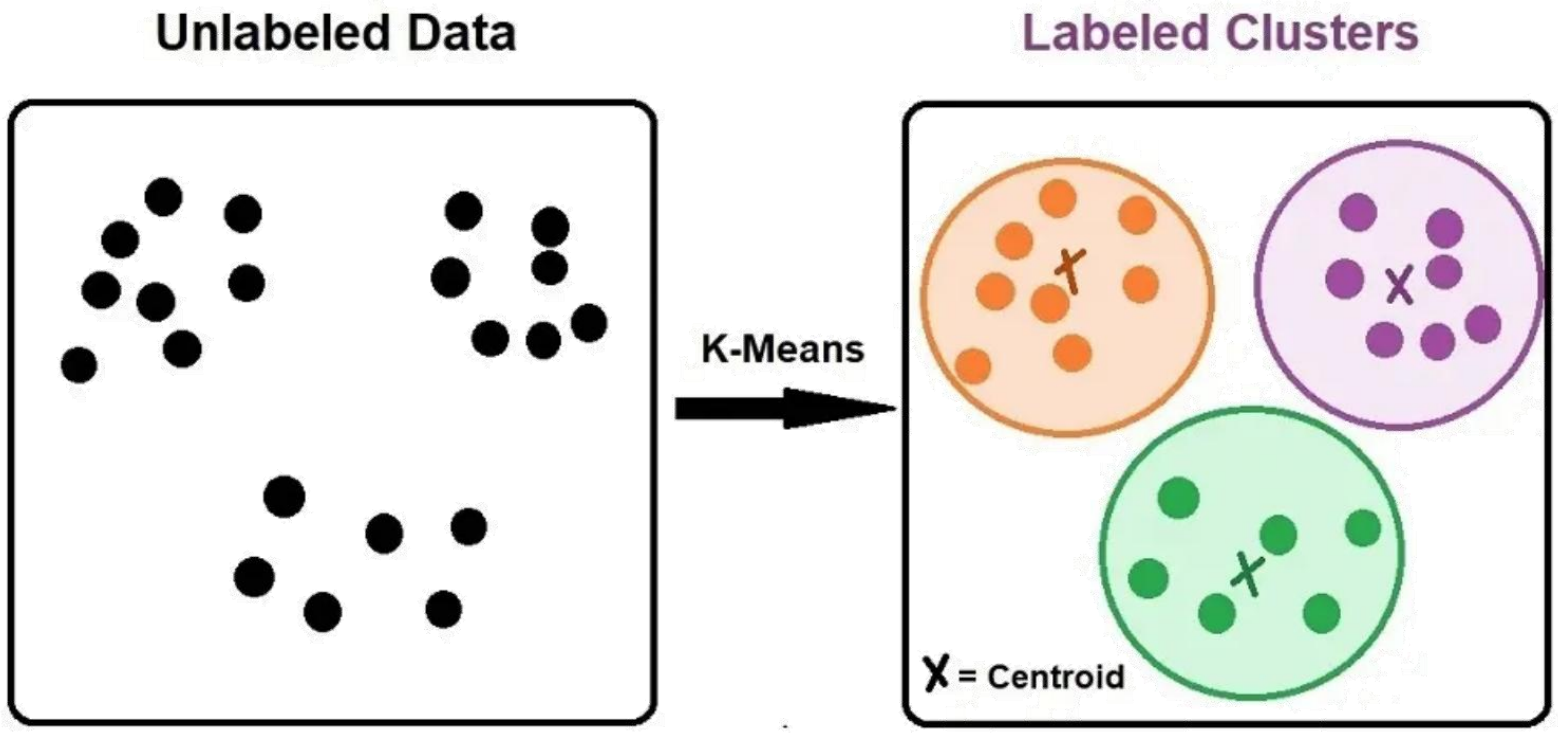


B. After K-Means



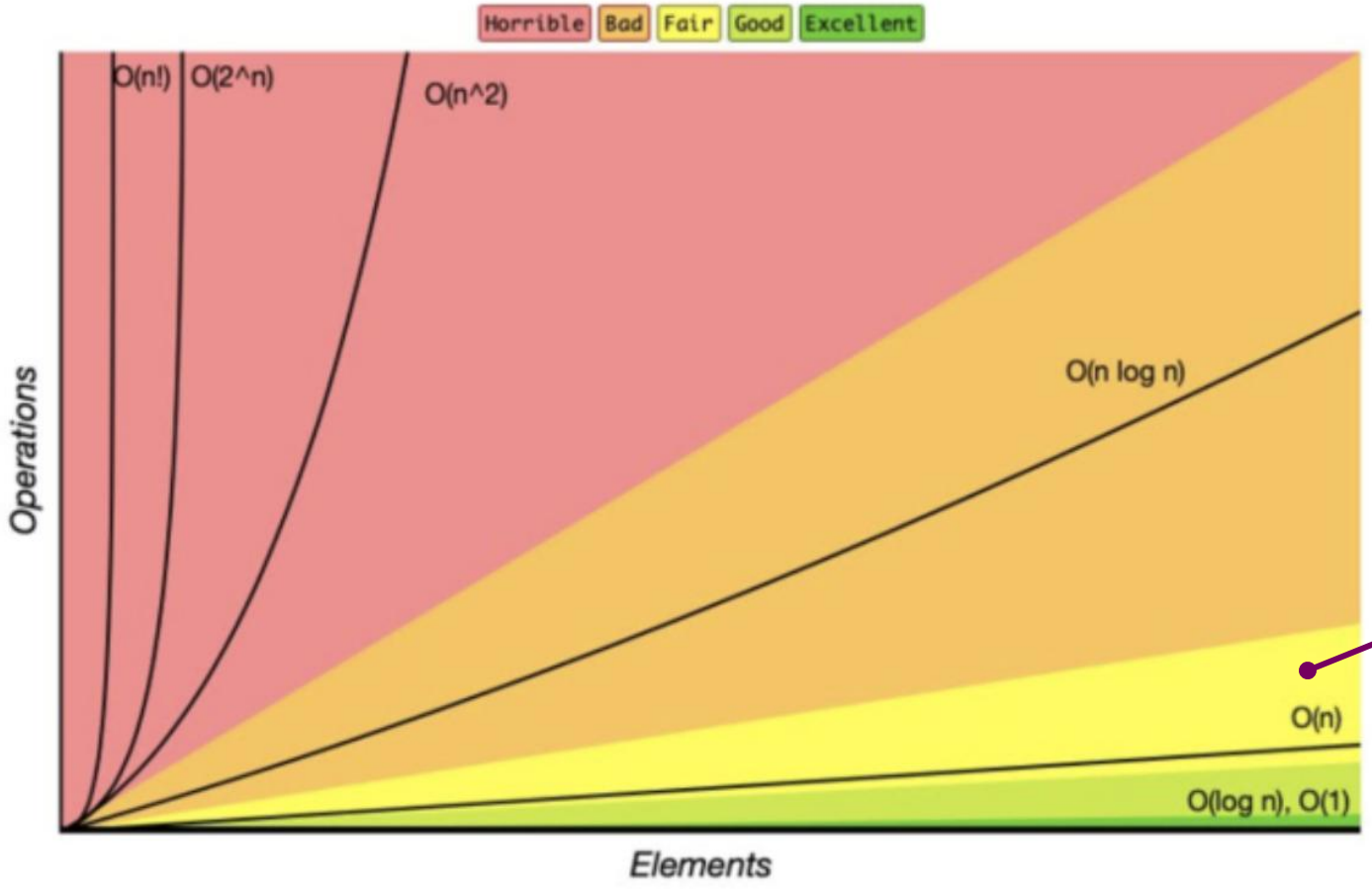
Centroid

Centroid is the center of a cluster determined by the mathematical mean of all the points in that cluster



Big-O Complexity Chart and K-Means

Complexity Chart for all Big O notations (Image: Wikimedia Commons)



Note: K-Means has a complexity of O(n)

K-Means: Four Steps

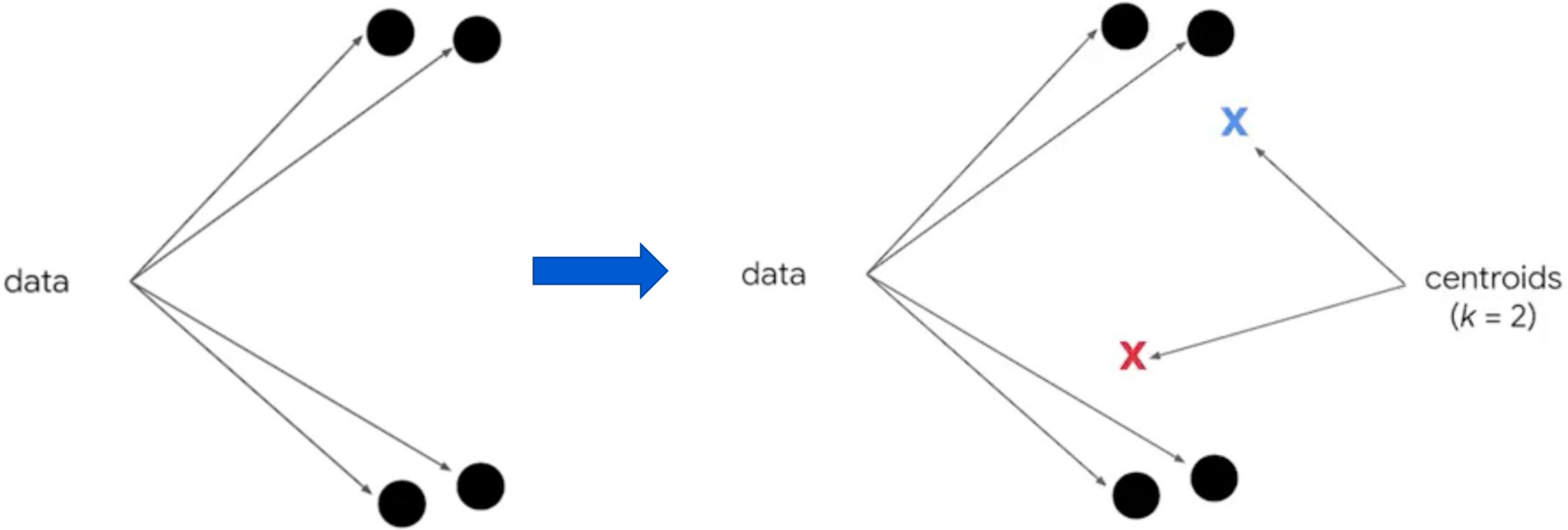
- **Step 1. Initialization**: Initiate k-centroid
- **Step 2. Assignment**: Assign all points to the nearest centroid
- **Step 3. Centroid Update**: Recalculate the centroid of each cluster based on the points assigned to it:

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

- **Step 4. Repetition:** Repeat Step 2 and 3 until the algorithm converges:

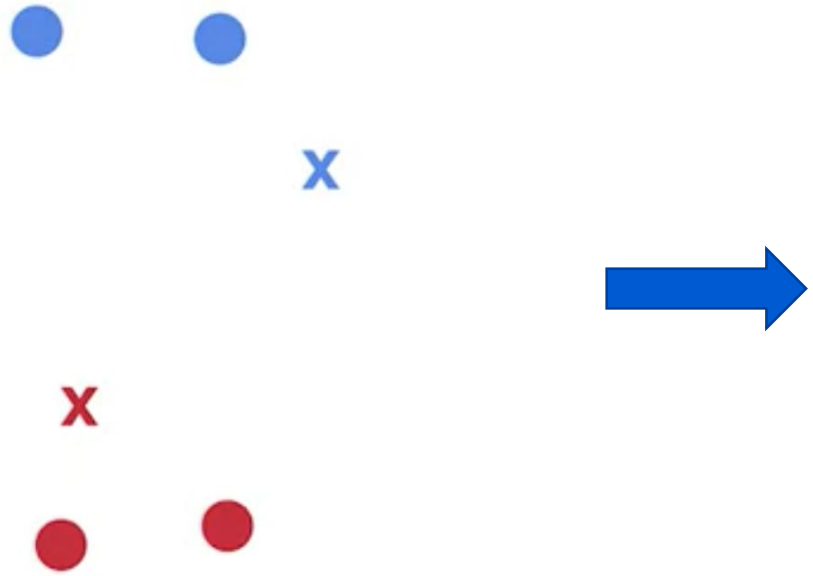
$$C_i = \frac{1}{|N_i|} \bullet \sum_{i=1}^{n} x_i$$

Example 2. K-Means: Step 1



Example 2. K-Means: Step 2, 3 and 4

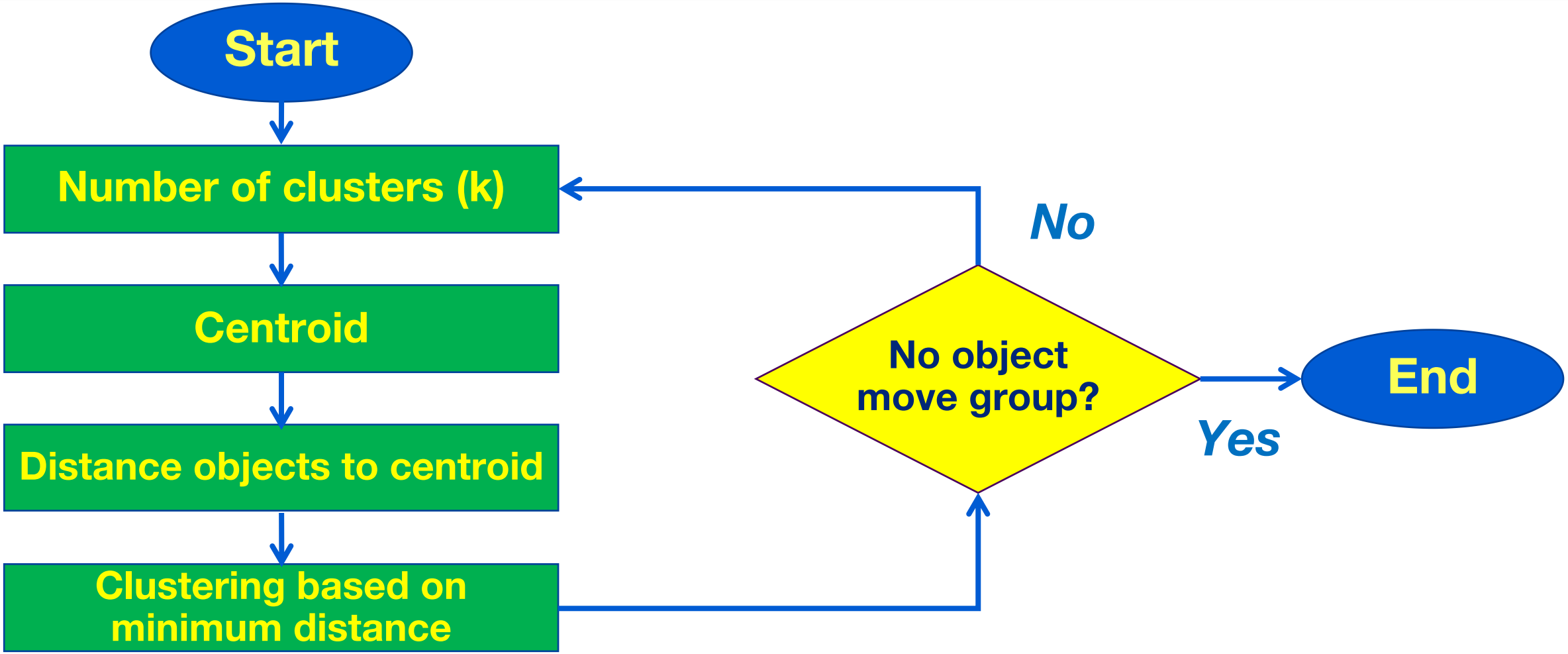
Step 2:



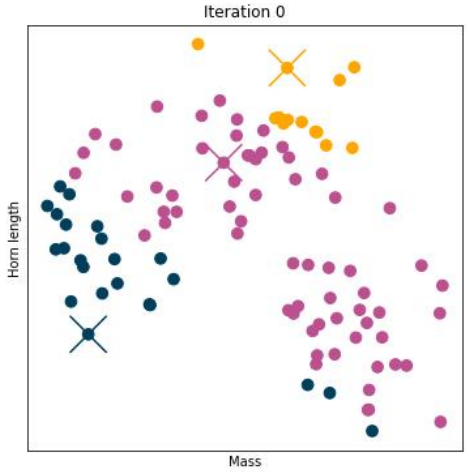
Step 3:

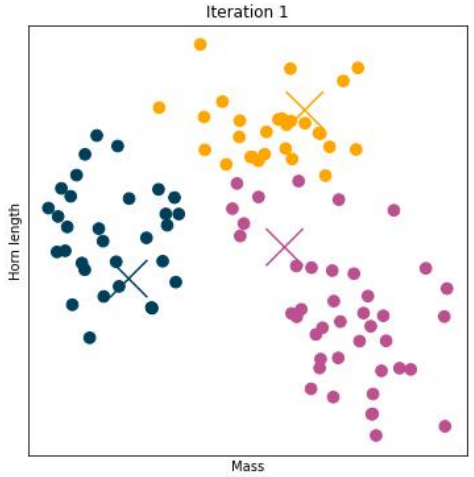
Step 4: To repeat Step 2 and 3 until the algorithm converges

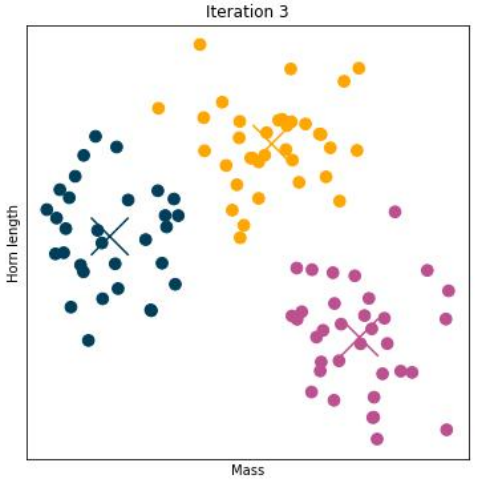
K-Means Model: Flowchart



Example 3. K-Means Modelling with k =2



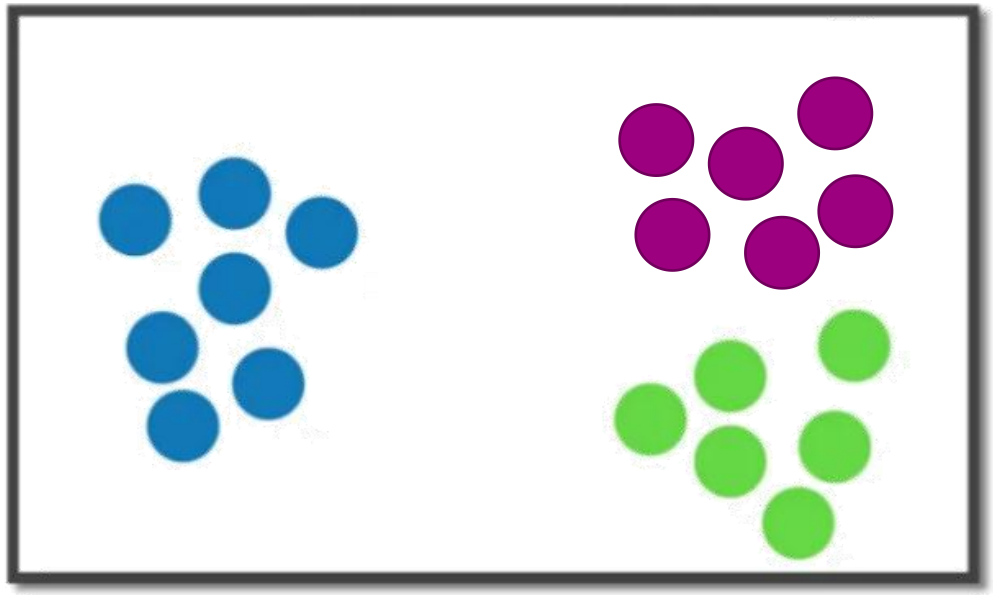




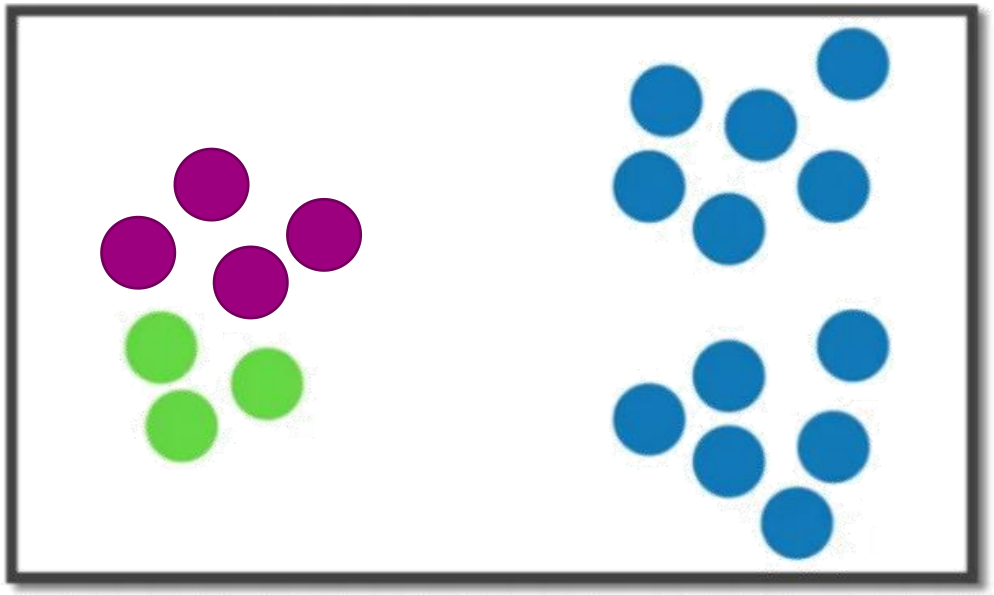
K-Means: Local Minima

Local Minima are suboptimal solutions that occur when the algorithm converges to a configuration that minimizes the objective function within a limited region, but not globally (**not global minima**).

A. Global Minimum



B. Local Minimum

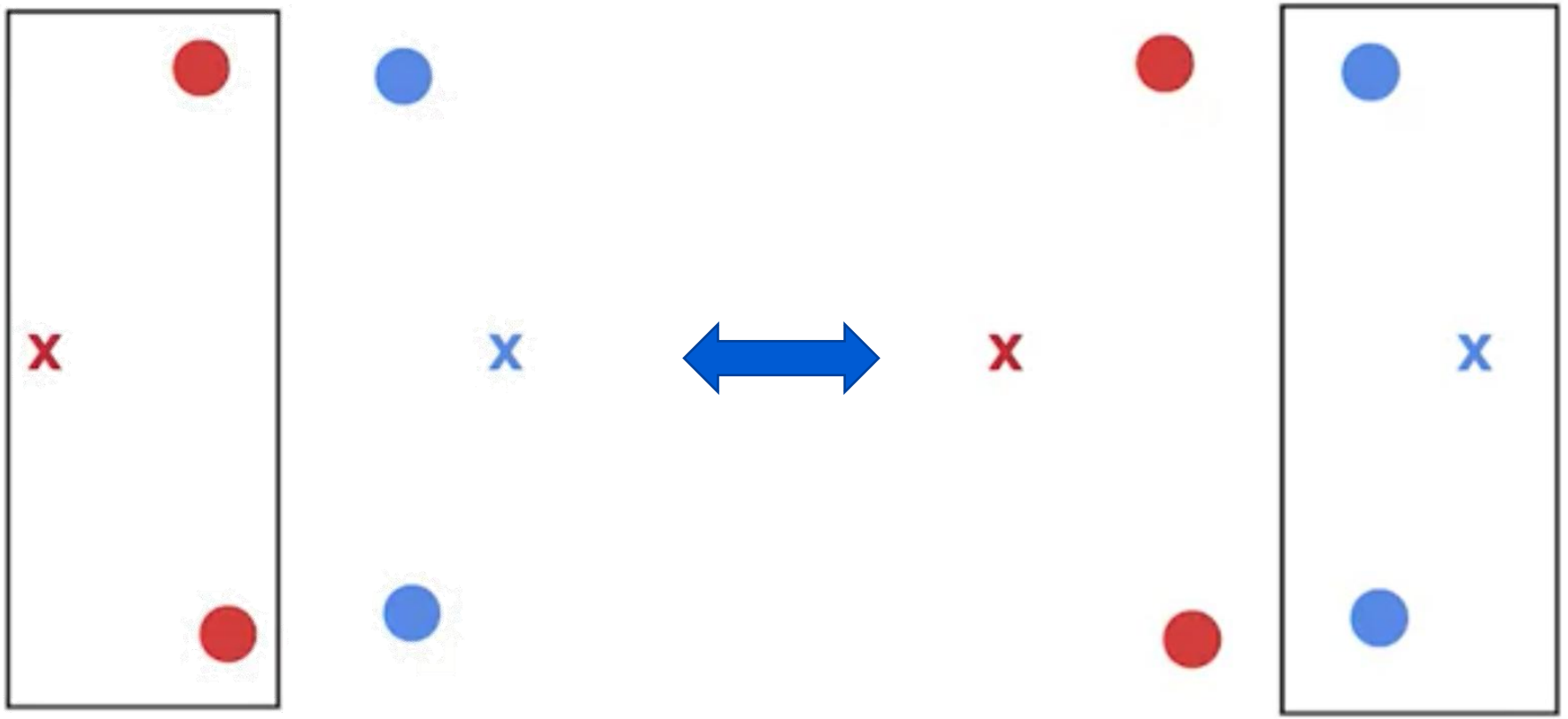


Example 4. K-Means Model: Local Minima

Step 1:

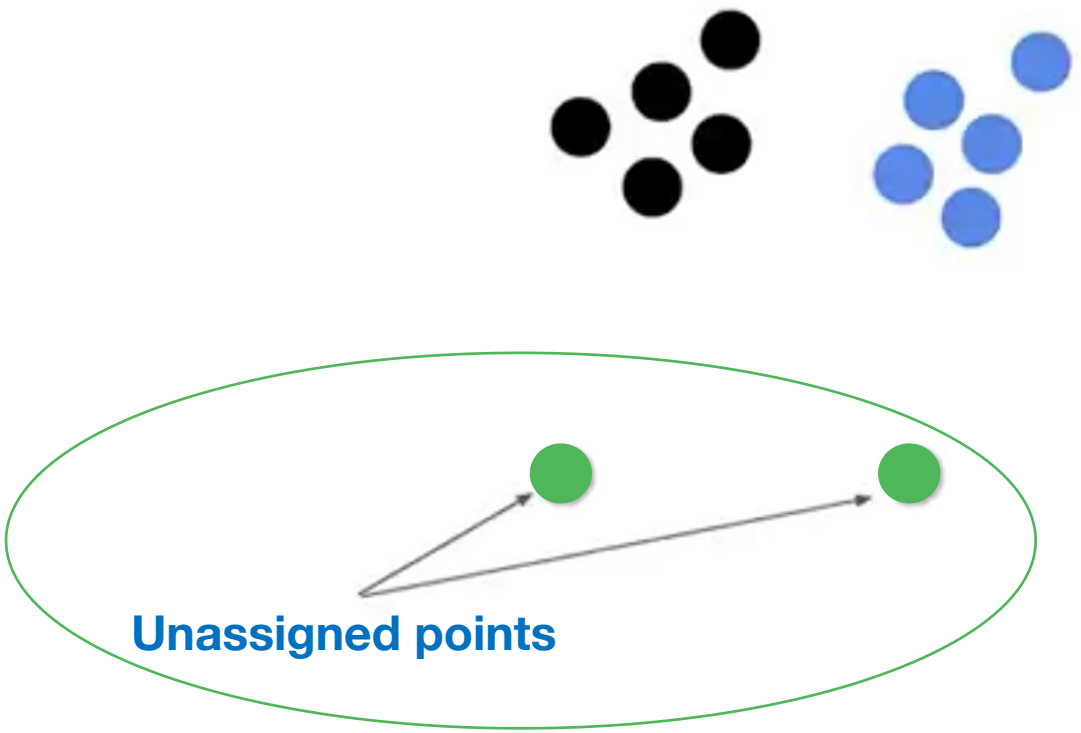
Example 4. K-Means Model: Local Minima (2)

Step 2:

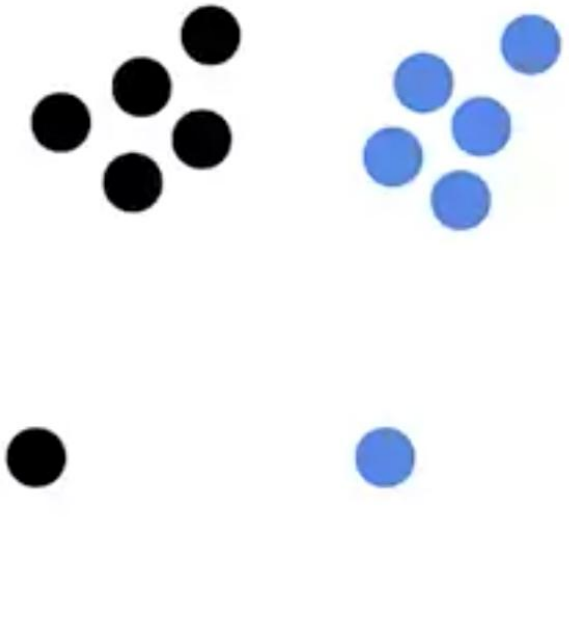


Example 5. Clustering vs Partitioning

A. Clustering



B. Partitioning



K-Means Model: Summary

- **K-Means** is an unsupervised learning technique that groups unlabeled data into K clusters based on similarity;
- **Unlabeled data** is a raw data that lacks explicit tags or categories;
- **The clustering process** has **four steps** that repeat until the **model converges**;
- **The value for K** is a decision that the modeler makes;
- It's important to build **multiple models** to avoid poor clustering.

Coding Activity 5-1. Unsupervised Machine Learning. Clusterization

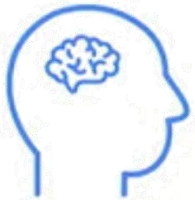
Lab 5-1. Unsupervised Machine Learning. Clustering || K-Means for color compression with Python (Non- Synthetic Data)

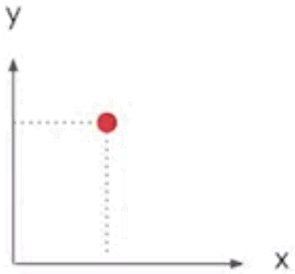
Steps to follow:

1. Upload the following files from the module learning room:
   - Jupiter notebook "Lab_5-1_K_Means_for_color_compression_with_Python.ipynb"
   - Data jpg-file "kmeans_for_color_compression_photo.jpg"
2. Follow along in the Jupiter notebook

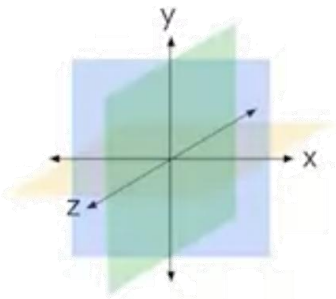
K-Means Models: Evaluation

2-D





3-D



4-D+

?

Example 6. Evaluation Metrics

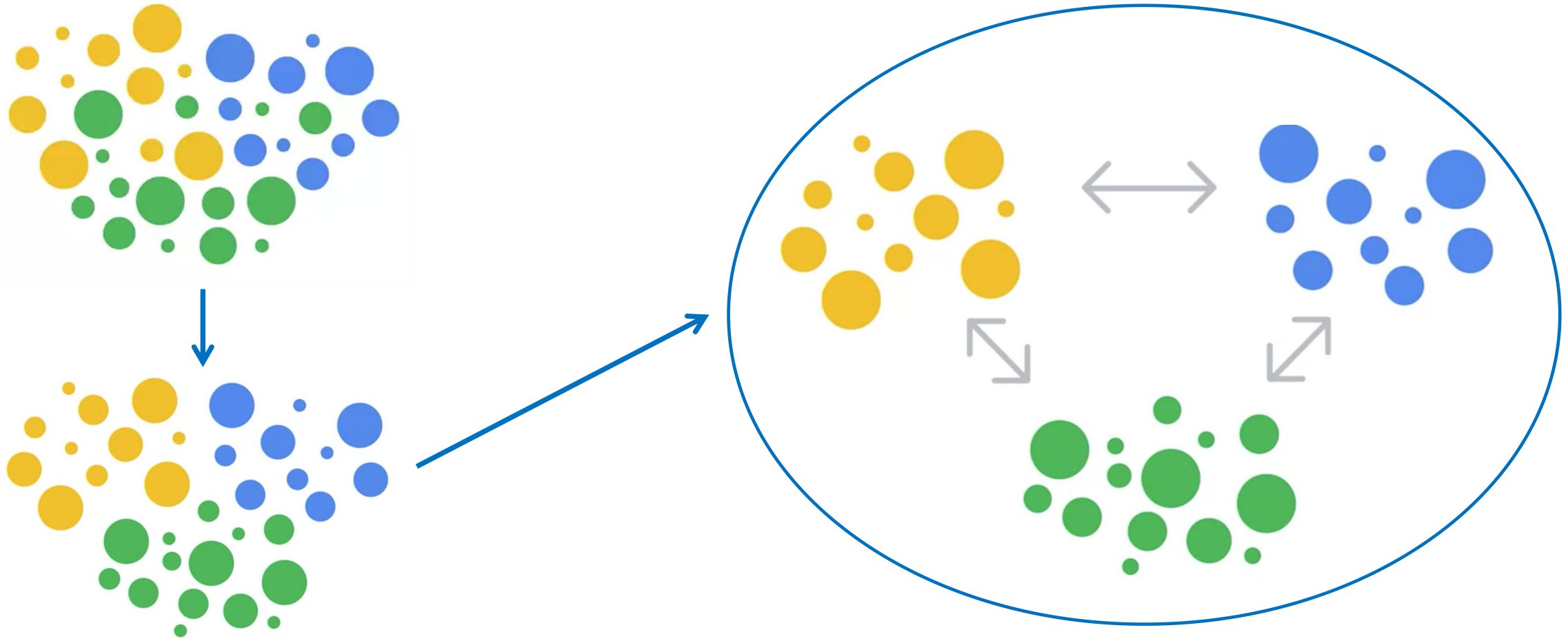
K-Means Model

- R^2
- MSE
- AUC
- Precision
- Recall

Linear and Logistic Regression

- R^2
- MSE
- AUC
- Precision
- Recall

Example 7. Clustering (k = 3)



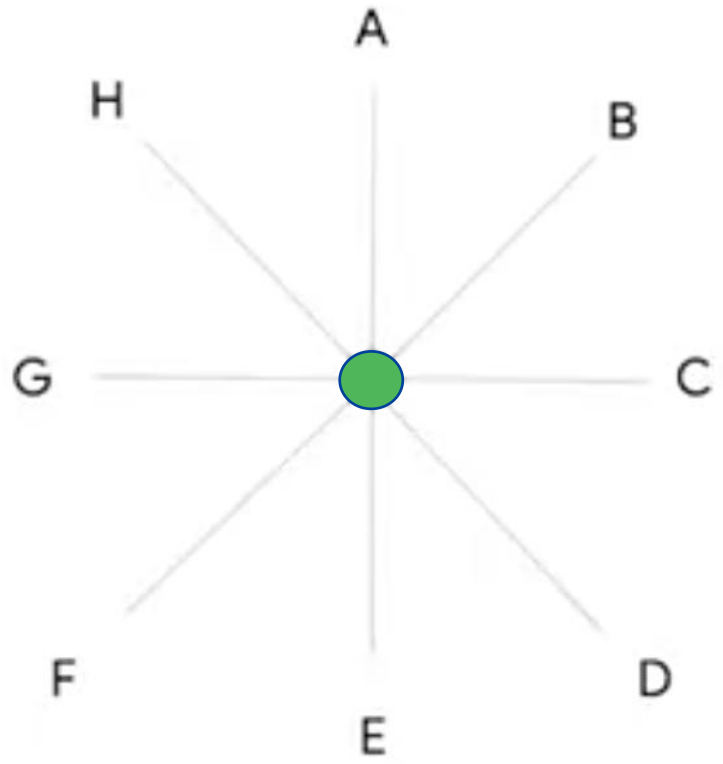
K-Means Evaluation Metrics: Inertia

Inertia is a sum of squared distances between each observation and its nearest centroid

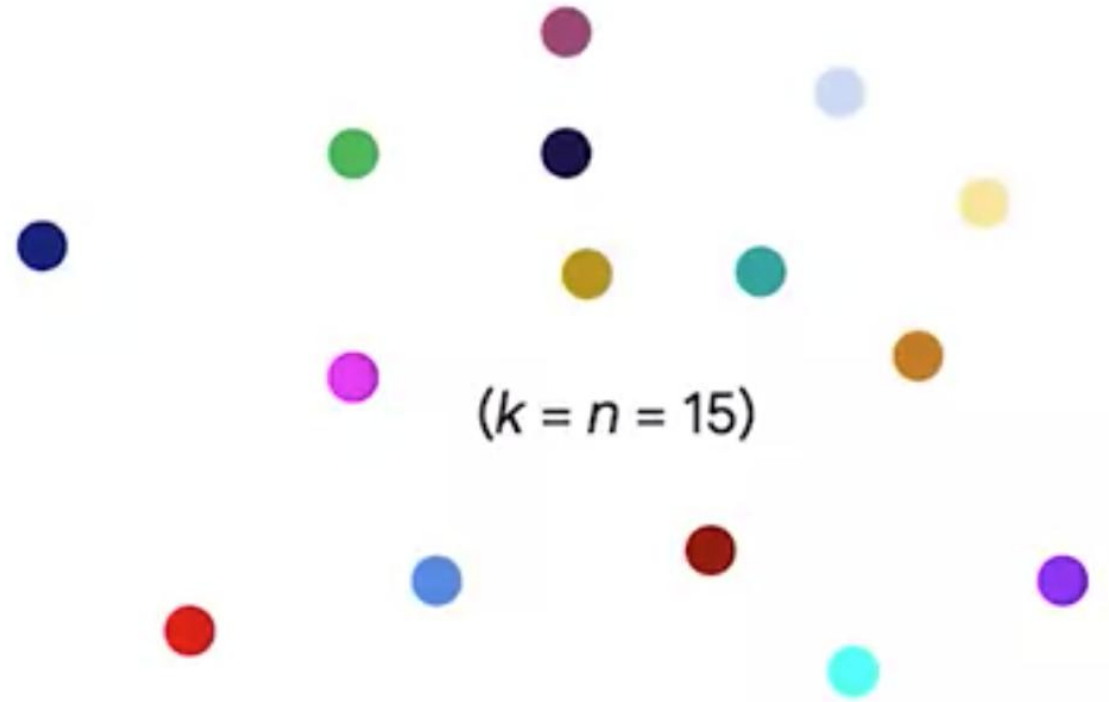
$$Inertia = \sum_{i=1}^{n} (x_i - c_k)^2$$

Example 8. When Inertia Equals Zero

A. All observations are identical:



B. N of clusters = N of observations

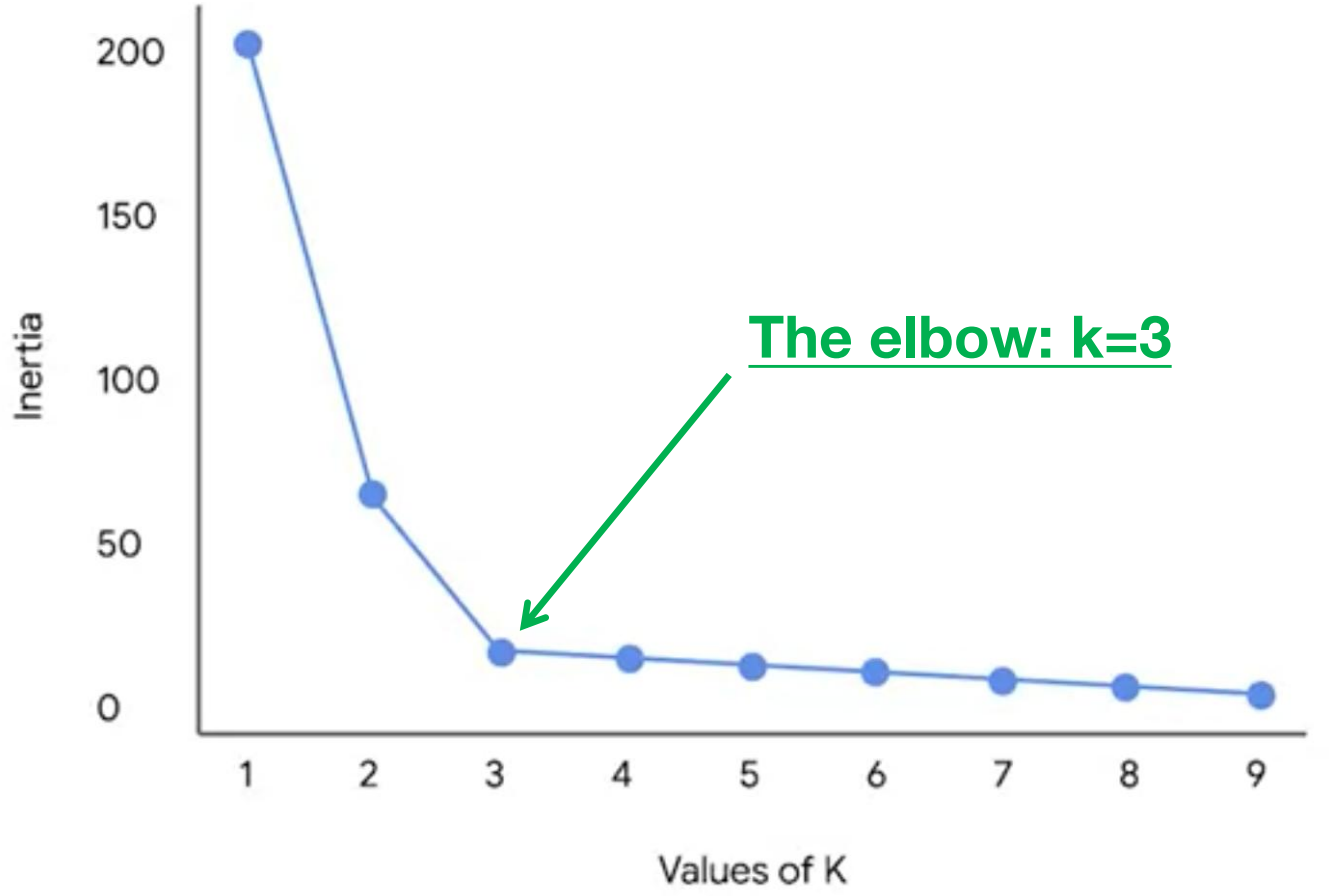


The Elbow Method

The Elbow Method:

1. Build models with different values of k;
2. Plot the inertia for each k-value.
3. Identify the elbow of the curve

The Elbow Method using Inetria



K-Means Evaluation Metrics: Silhouette Score

Silhouette score is the mean of the silhouette coefficients of all observations in the model:

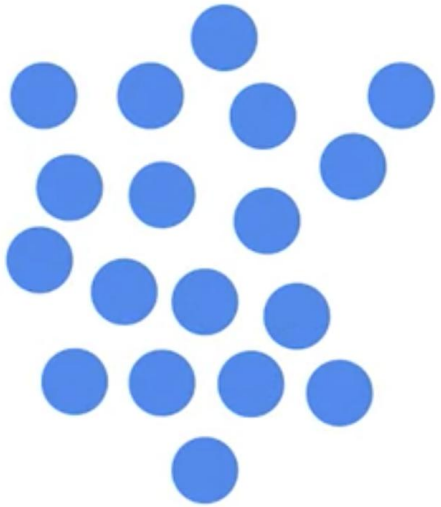
$$S = \frac{b - a}{max(a, b)} \; ; \; S \in [-1; \; 1]$$

where:

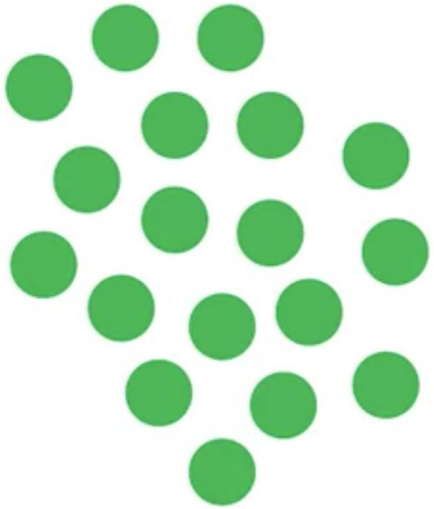
- **a** is the mean distance from that observation to all other observations in the same cluster;
- **b** is the mean distance from that observation to each observation in the next closest cluster.

Case 1. Silhouette Score: S ≈ 1

Cluster A

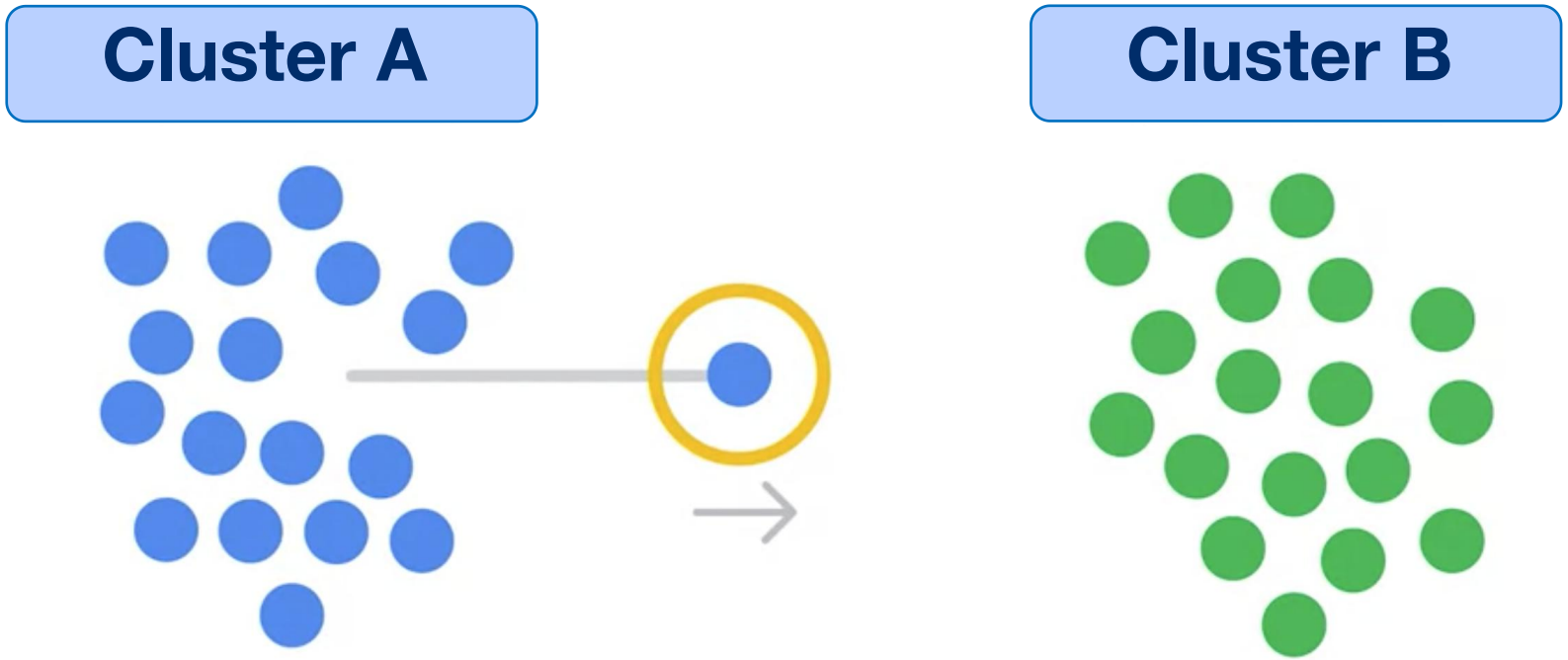


Cluster B



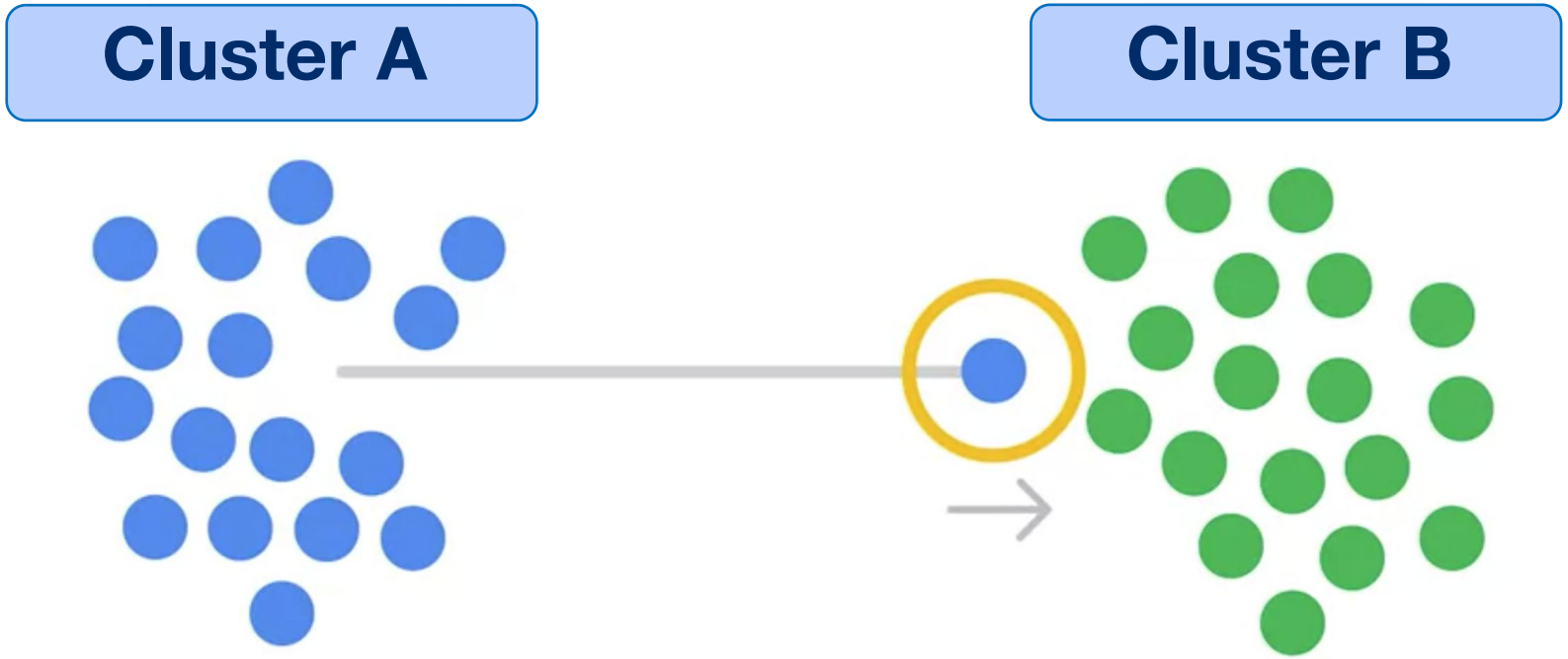
$$\frac{(b-a)}{\max(a,b)} = \frac{8.65}{9.50} = 0.91$$

Case 2. Silhouette Score: S ≈ 0



$$\frac{(b-a)}{\max(a,b)} = \frac{0.65}{5.22} = 0.12$$

Case 3. Silhouette Score: S ≈ -1



$$\frac{(b-a)}{\max(a,b)} = \frac{-7.07}{8.12} = -0.87$$

Coding Activity 5-2. Unsupervised Machine Learning. Clustering

Lab 5-2. Unsupervised Machine Learning. Clustering. K-Means || K-Means: Inertia and Silhouette score with Python (Synthetic Data)

Steps to follow:

1. Upload the following files from the module learning room:
   - Jupiter notebook
     "Lab5-2_K_Means_Inertia_Silhouette_with_Python.ipynb"
2. Follow along in the Jupiter notebook

Coding Activity 5-2. Unsupervised Machine Learning. Clustering

Lab 5-2. Unsupervised Machine Learning. Clustering. K-Means || K-Means: Inertia and Silhouette score with Python (Synthetic Data)

Overview:

- Packages to import
- Data scaling
- Instantiating and fitting a K-Means model
- Usins the *labels_* and *inretia_attributes*
- Using the silhouette_score() function
- Determening a final value for k

Coding Activity 5-2: Scikit-Learn Scalers

1. StandardScaler:

$$X_{Scaled} = \frac{X_i - \mu_X}{\sigma_X}$$

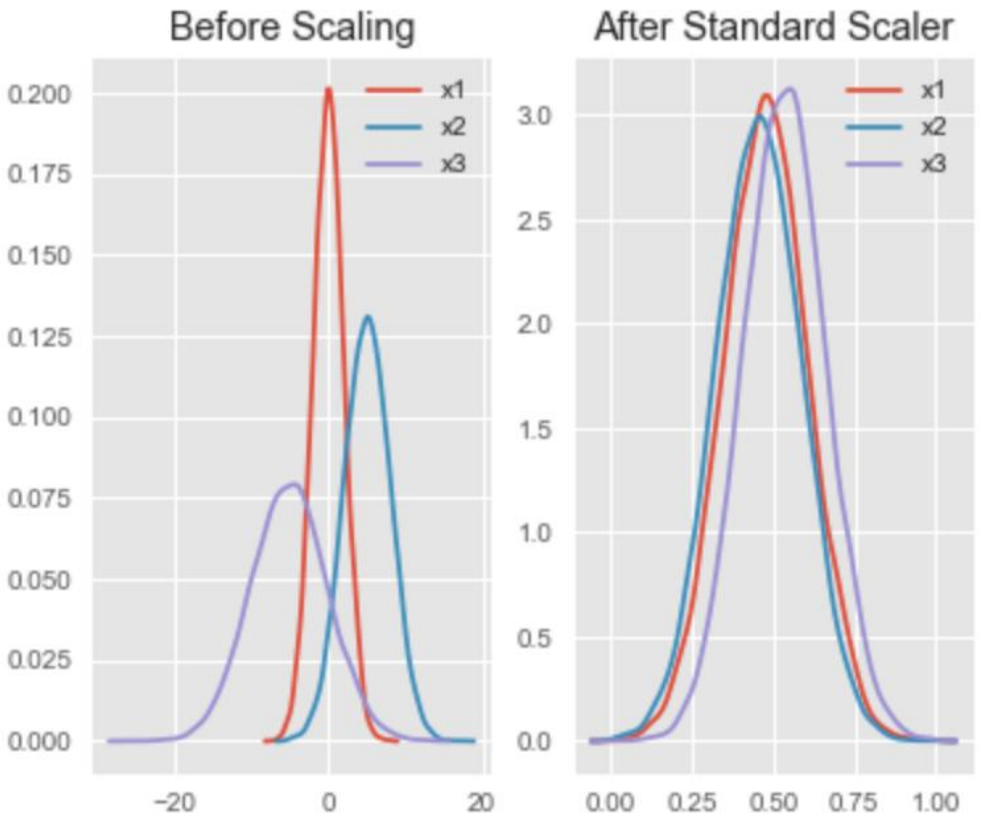
2. MinMaxScaler:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}};$$

$$X_{scaled} \in [0;\ 1]$$

3. Normalizer

Case 1. StandardScaler



Q&A

Thank you!