K-MEANS CLUSTERING

Thomas Tiam-Lee, PhD

Norshuhani Zamin, PhD

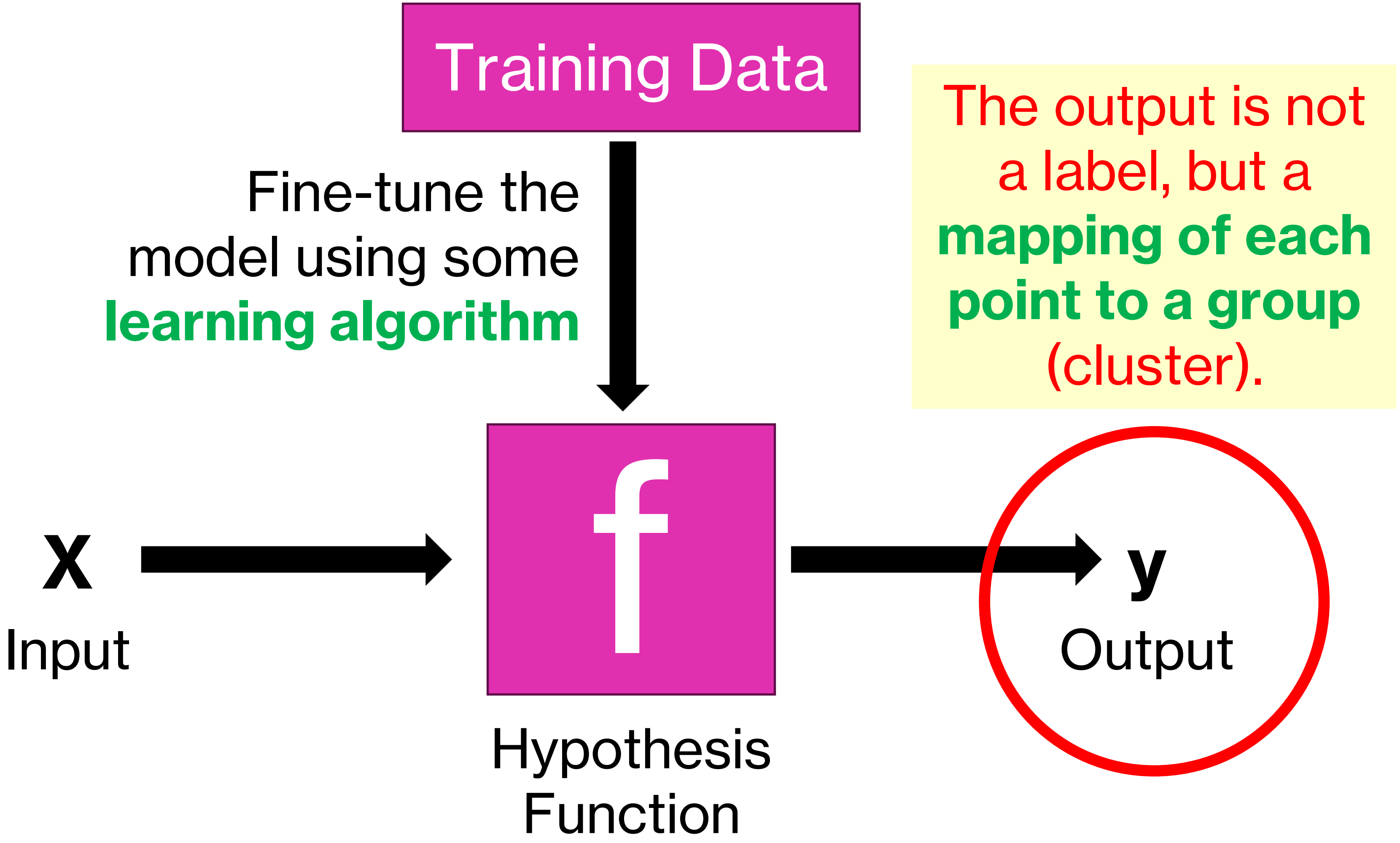
Unsupervised Learning Algorithm

- **Unsupervised learning algorithm**
  - No target variable (label)

- A model designed for **clustering**
  - Given a set of instances, group them according to similarity

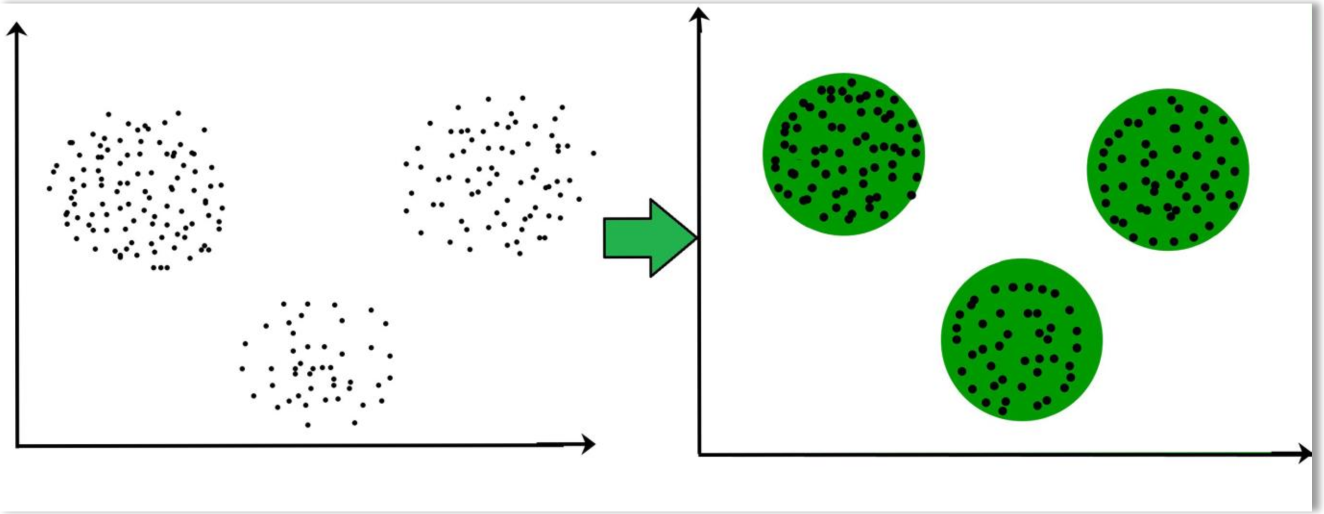
- **Example:** A company looks at their customer purchasing patterns and identifies a set of "customer profiles"

Clustering Task



Clustering Task

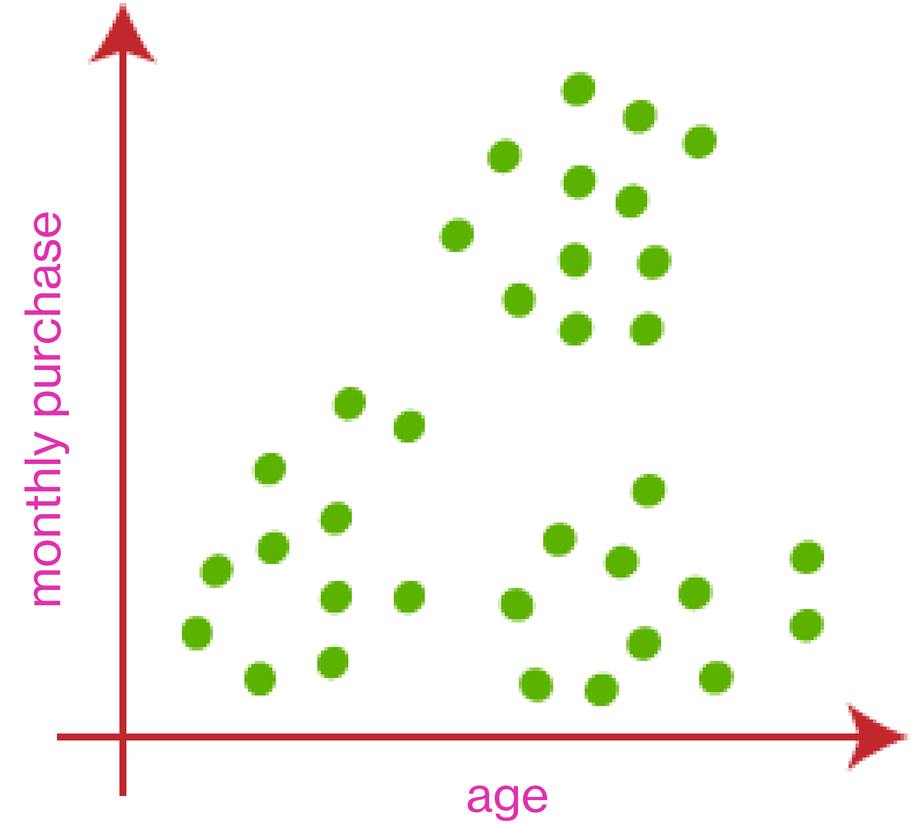
- Divide n data points such that the data points in the **same group are more similar** while data points in **different groups are more dissimilar**



Source: Geeksforgeeks.org

Example: 2 Features

- **Features**
  - x_1: the age of the customer
  - x_2: the average monthly purchase of the customer



Challenges

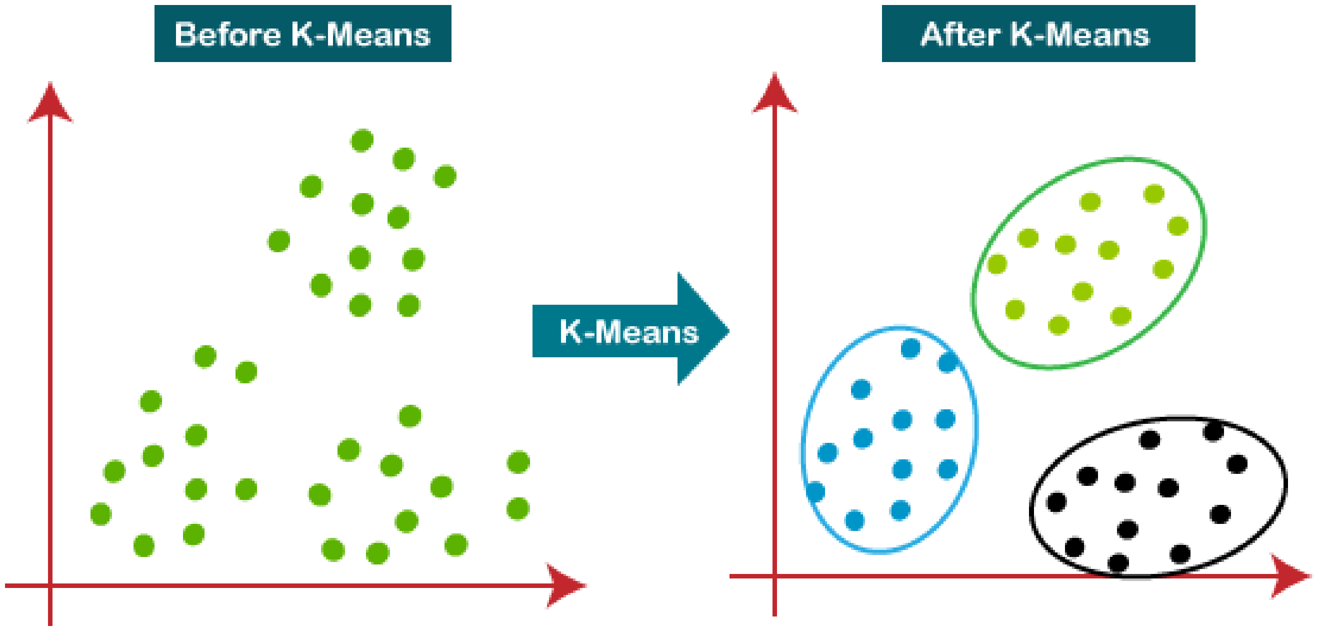
- How to mathematically define “similarity”?
- How do you know the number of groups?
- How to evaluate whether the grouping is good or not?

Clustering Algorithms

- **K-Means Clustering** is just one of many other clustering algorithms
  - DBSCAN
  - Hierarchical Clustering
  - Gaussian Mixture Models
  - BIRCH
  - and others...

K-Means Clustering

- An algorithm that aims to partition n observations into k clusters.
- Each cluster has a **centroid** (mean), serving as the "prototype" of the cluster.



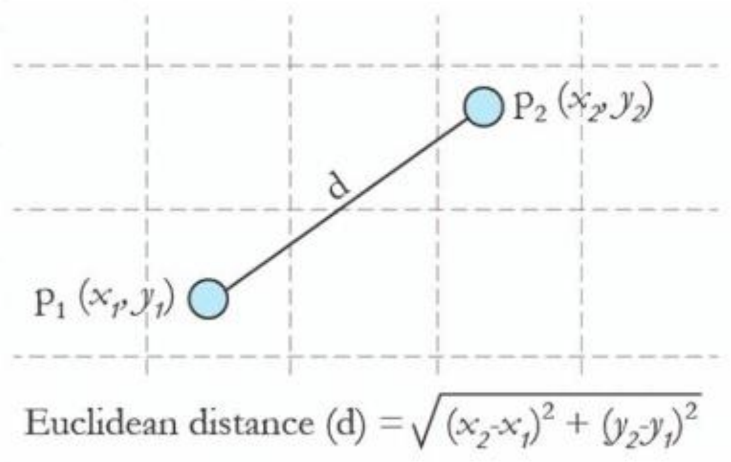
Source: Javatpoint.com

“Similarity” of 2 Data Points

- We can use a **distance metric**, such as Euclidean distance, to measure how similar two data points are.

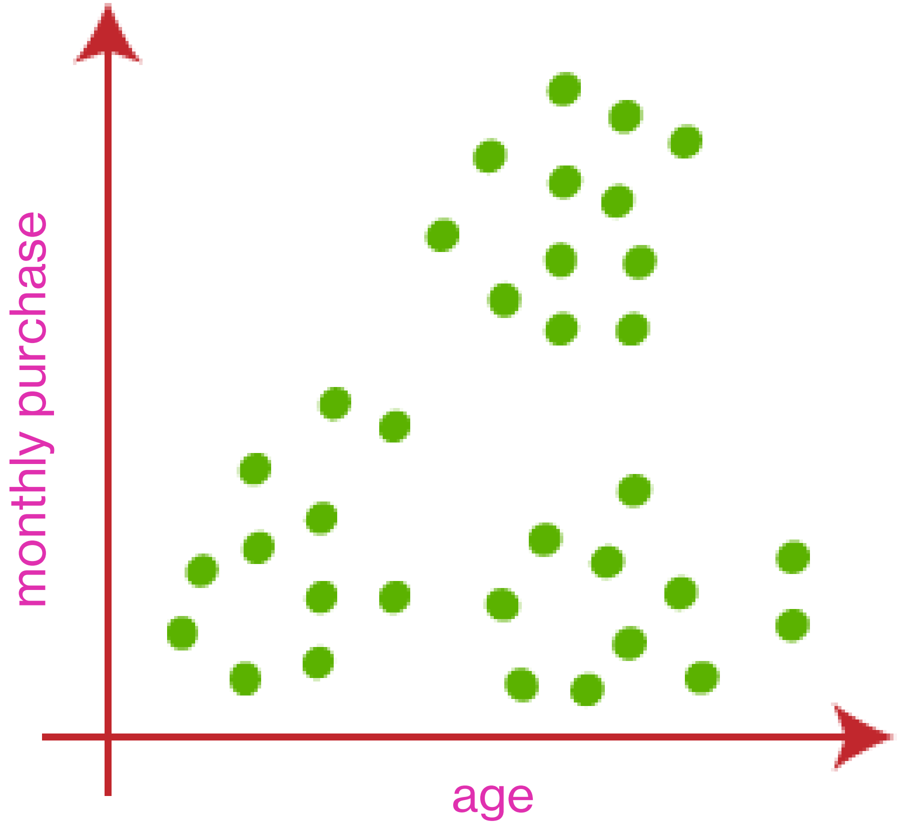
$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2}$$

- Lower distance = more similar!



Scaling / Normalization

- When working with features that are not on the same scale, need to perform **scaling** / **normalization** to make sure one feature does not overwhelm others in the distance metric.

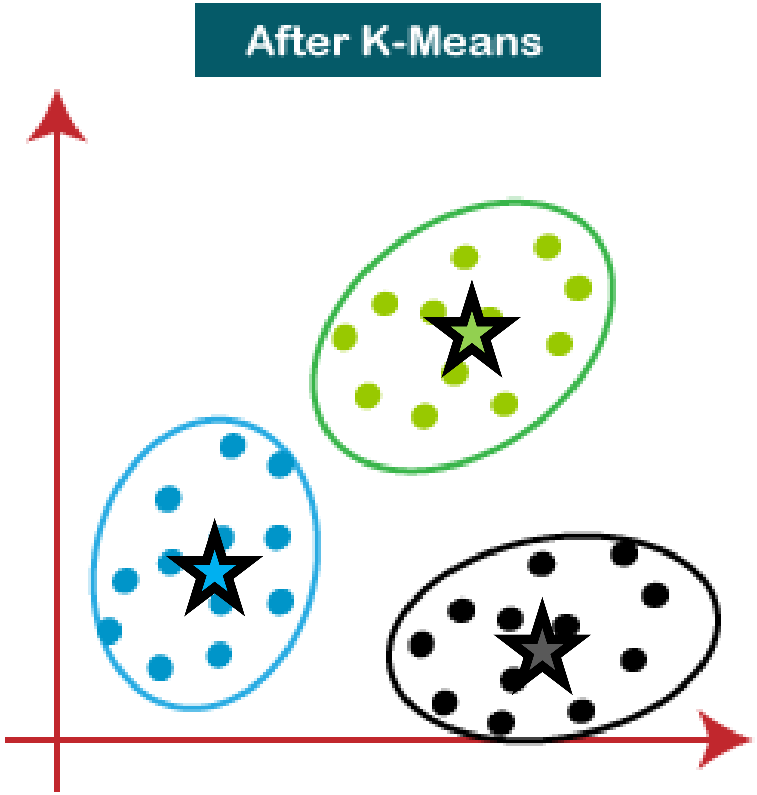


Other Distance Measures

- Manhattan distance
- Minkowski Distance
- Hamming Distance
- Cosine similarity

K-Means Clustering Model

- Each cluster has a "centroid" location.
- To determine which cluster an instance belongs to,
  - Compute distance of that point to each centroid.
  - The smallest distance will be the assigned cluster!



The Hyperparameter k

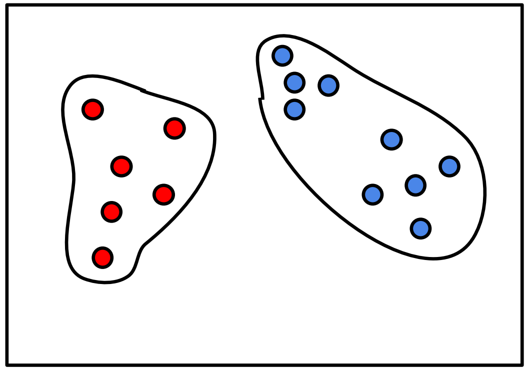
- The k-means clustering has a **hyperparameter**.

- **Hyperparameter:** a value that changes the behavior of a model. However, unlike a normal parameter, it is not fine-tuned by the learning algorithm. It has to be **set manually**.

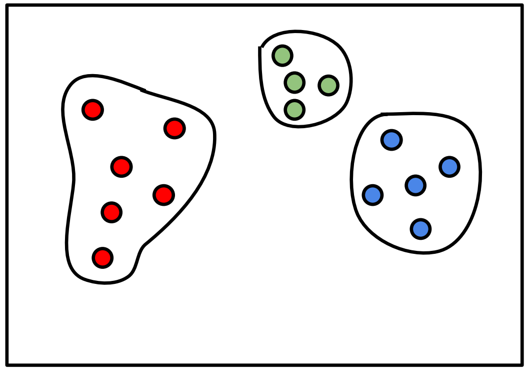
The Hyperparameter k

- k = **the number of clusters** / groupings
- The value of k tells how many clusters will result in the algorithm.
- Increasing k will result into less distortion.

$k = 2$



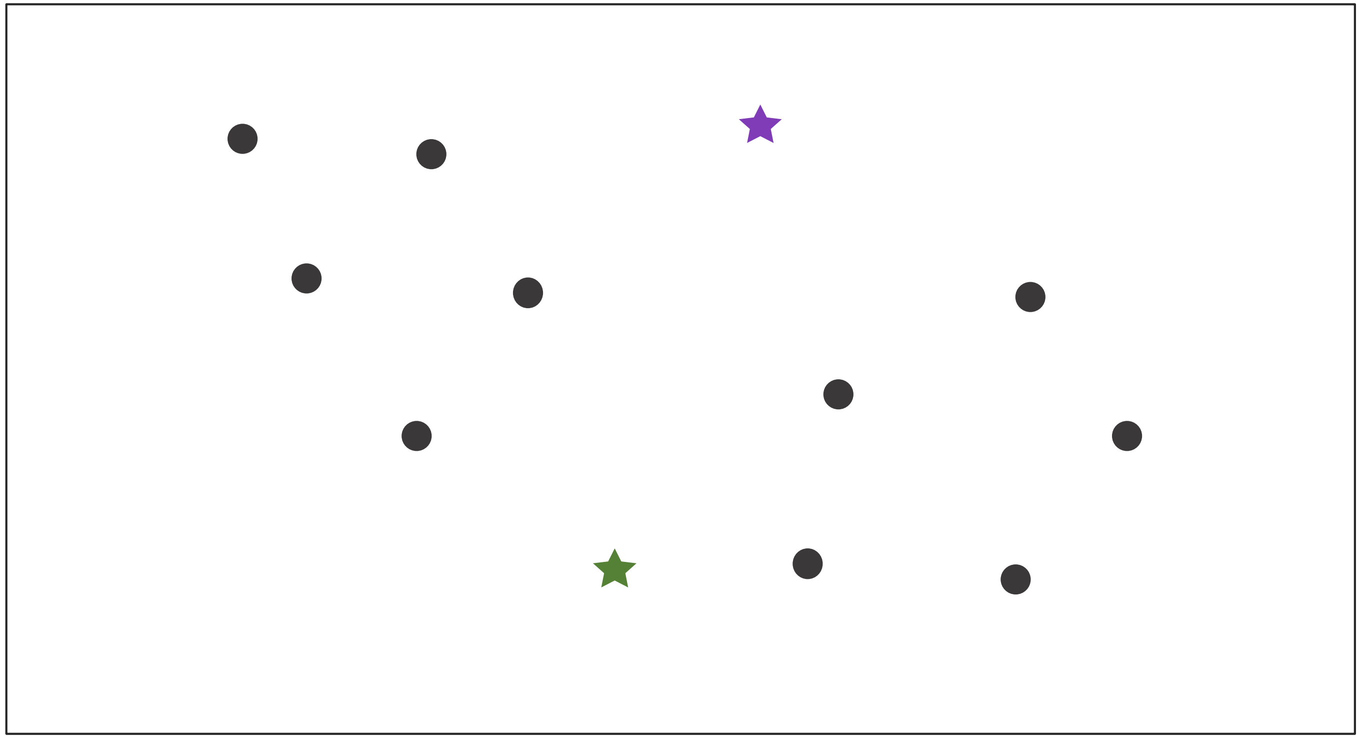
$k = 3$



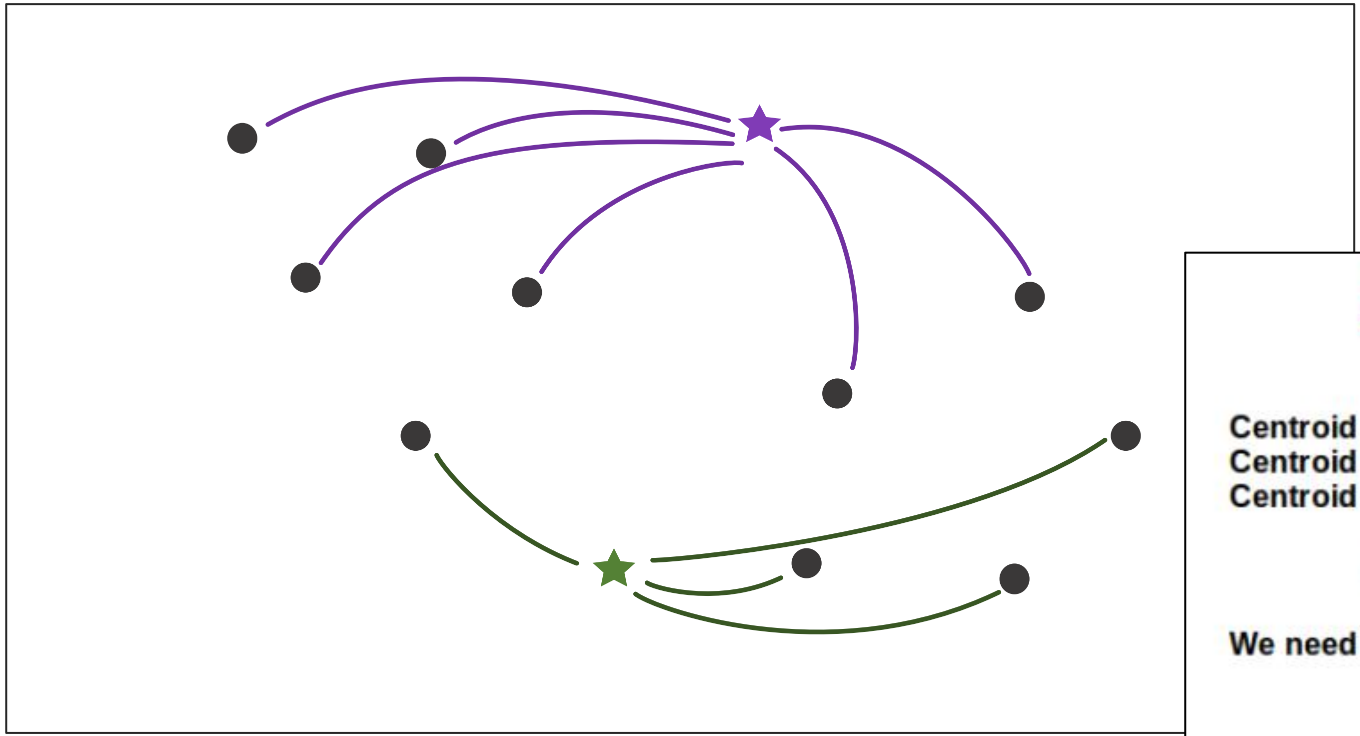
K-Means Clustering Learning Algorithm

1. Decide number of clusters (k)
2. Assign initial centroid values for each cluster (random)
3. Assign each instance to the closer cluster ***(expectation)***
4. Compute the new centroid of each cluster ***(maximization)***
5. Repeat 2 and 3 until convergence

1. Let’s assume number of cluster, k = 2.
2. Randomly assign initial centroid values for each cluster.



3. Assign each instance to the closer cluster.



For every i, set:

$$c_i := \min_j \left|\left|x^i - \mu_j\right|\right|^2$$

Example:

Assigning Clusters to data points

Centroid 1 = (4,3)
Centroid 2 = (11,6)
Centroid 3 = (8,10)

Datapoint = (7,4)

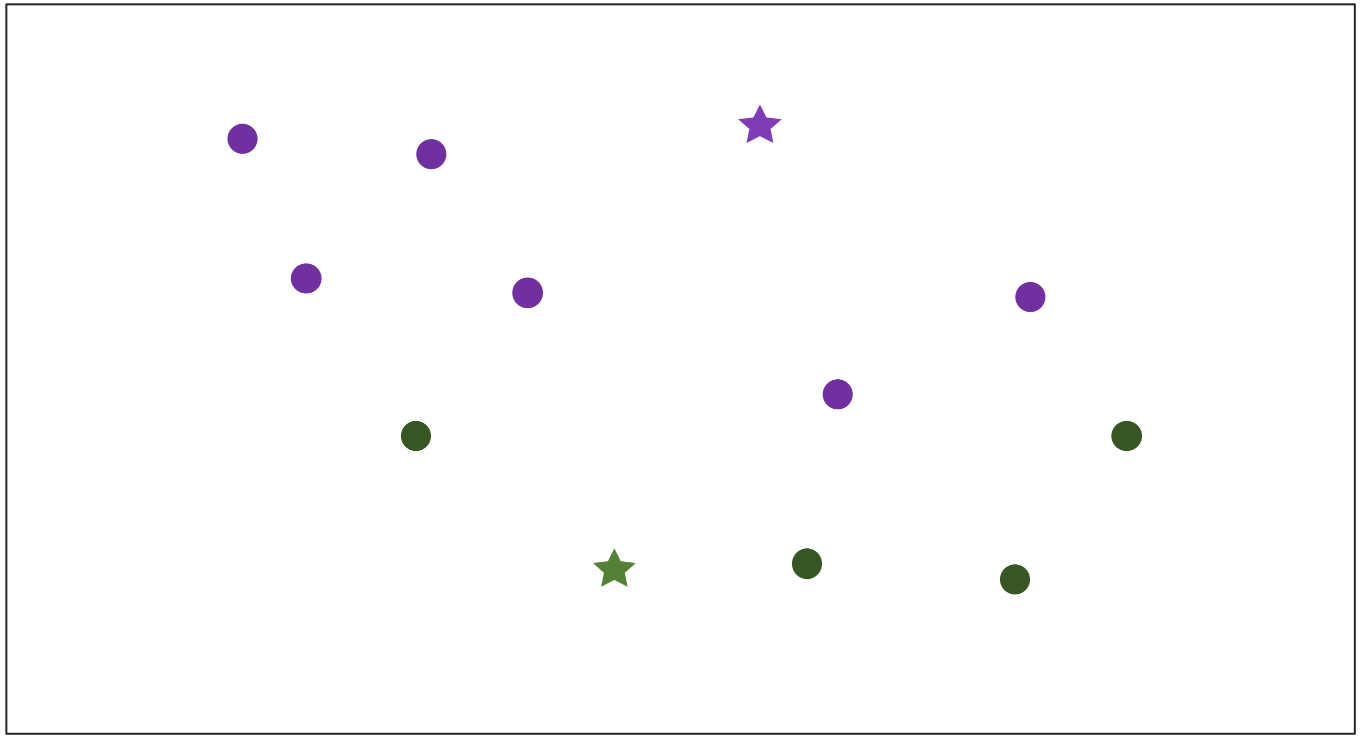
Apply Euclidean distance formula : $\sqrt{((x_2\text{-}x_1)^2 + (y_2\text{-}y_1)^2)}$

We need to find which centroid has minimum distance with the given datapoint.

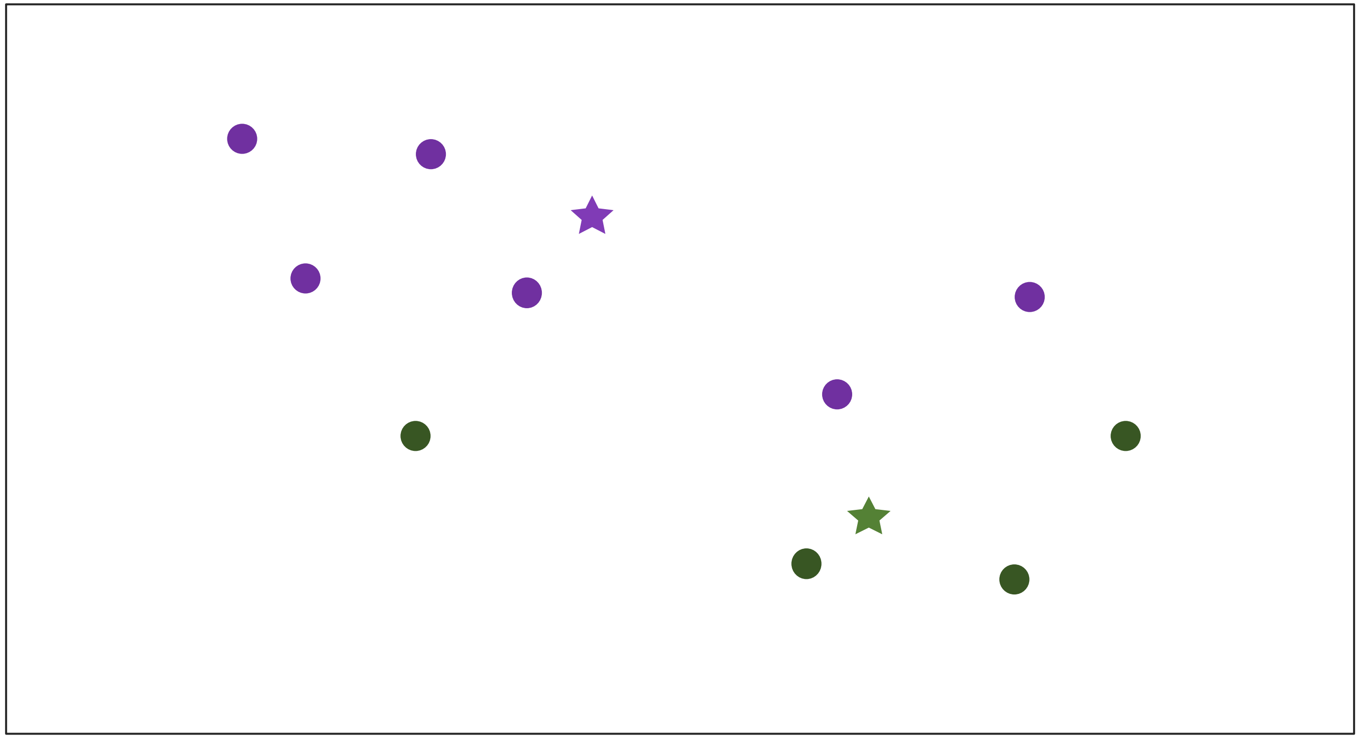
With centroid 1 : $((7 - 4)^2 + (4 - 3)^2)$ = 10
With centroid 2 : $((7 - 11)^2 + (4 - 6)^2)$ = 20
With centroid 1 : $((7 - 8)^2 + (4 - 10)^2)$ = 37

As we can compare, datapoint (4,7) is closest to centroid 1, therefore it will be assigned in cluster 1

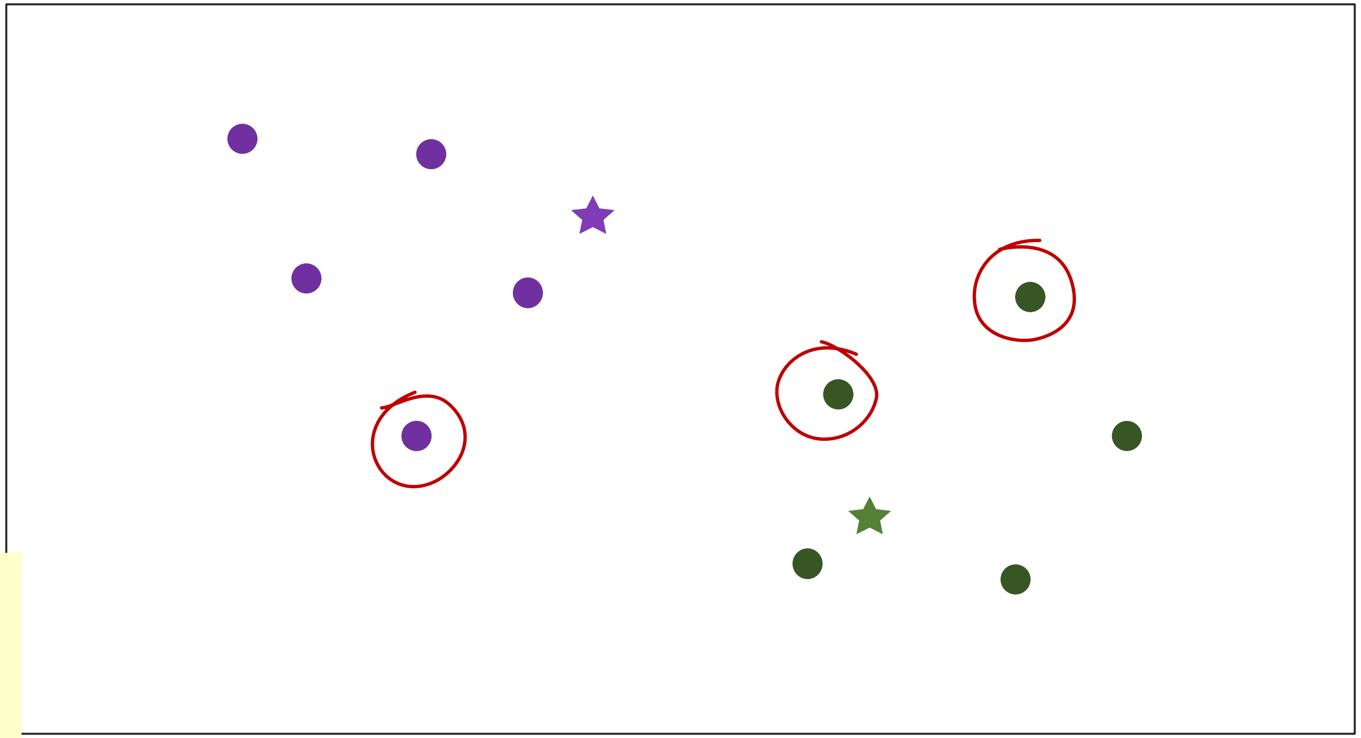
Note: Assigned instances to the random clusters in the first iteration.



4. Compute the new centroid of each cluster



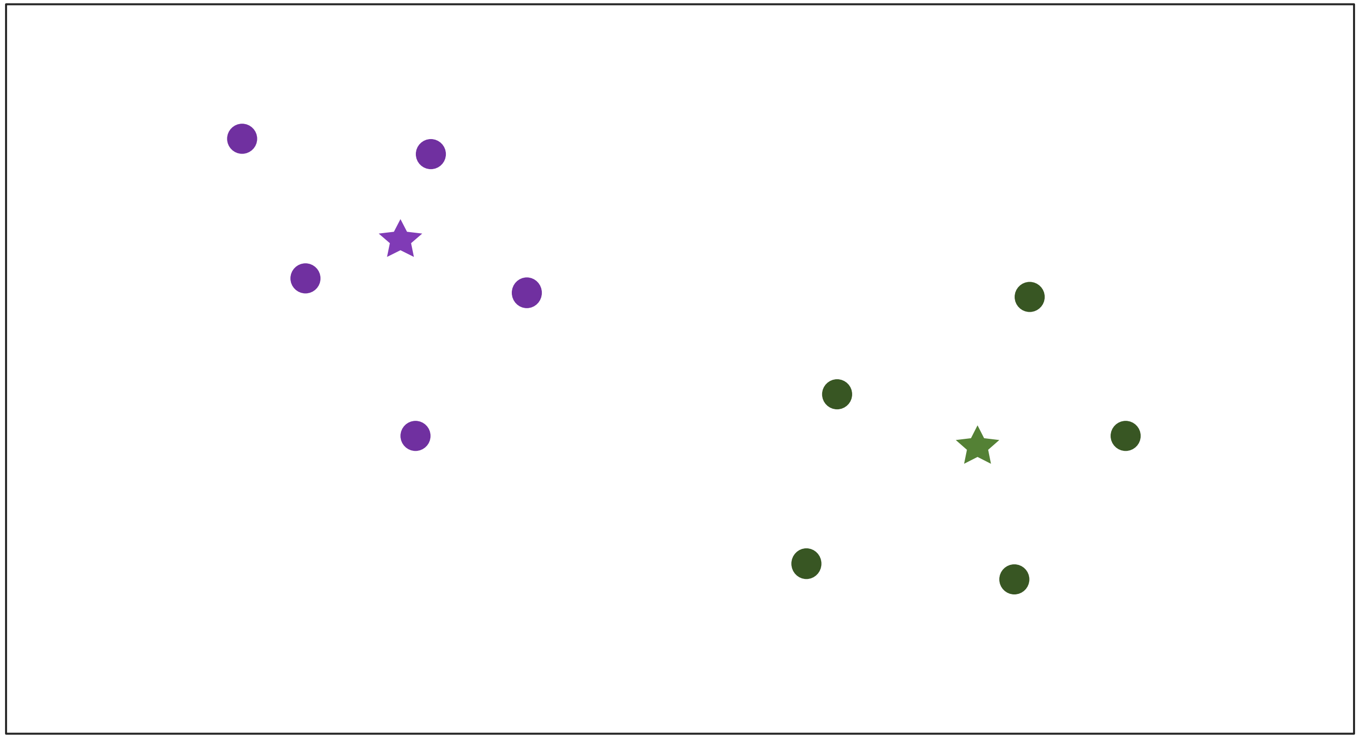
2 (repeat). Assign each instance to the closer cluster



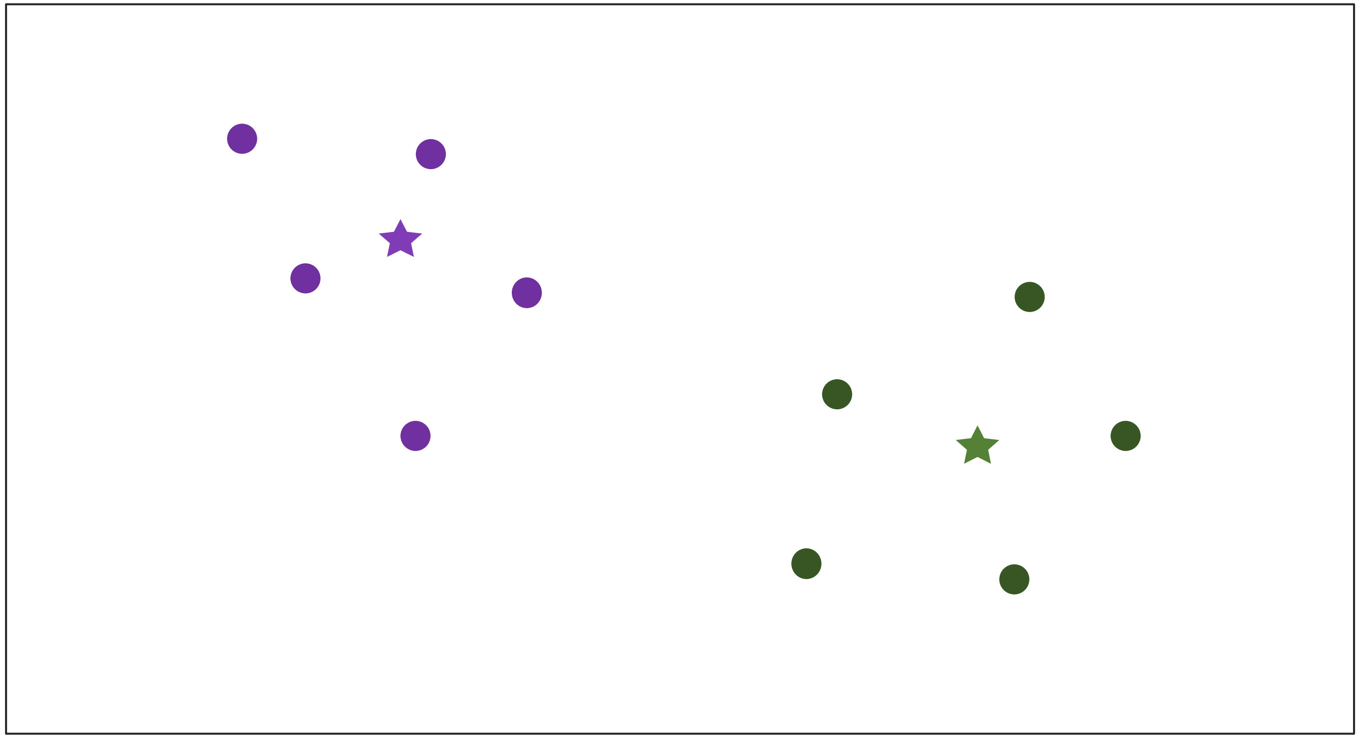
For every i, set:

$$c_i := \min_j \left|\left|x^i - \mu_j\right|\right|^2$$

3 (repeat). Compute the new centroid of each cluster



2 (repeat). Assign each instance to the closer cluster



For every i, set:

$$c_i \coloneqq \min_j \left|\left|x^i - \mu_j\right|\right|^2$$

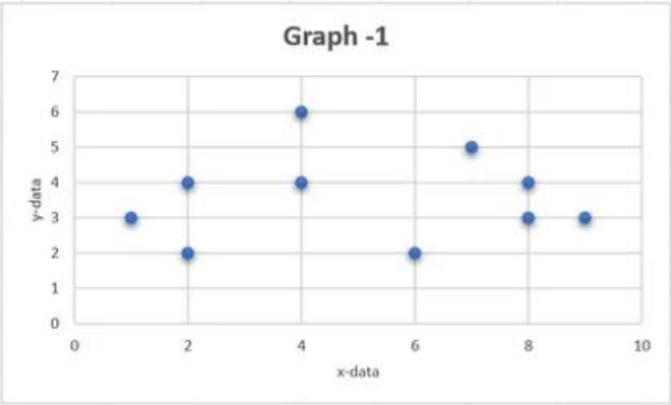
No change in cluster assignment! Converged.

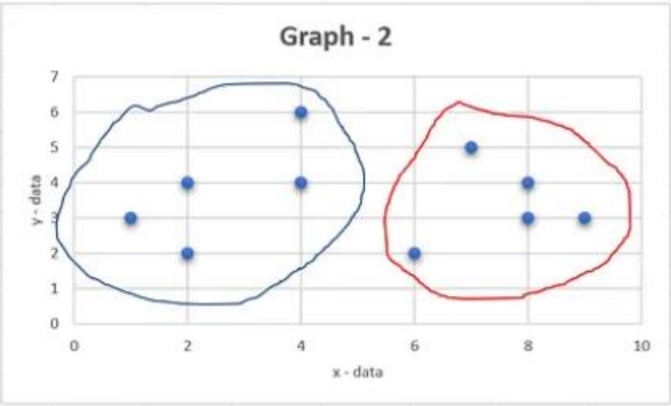
K-Means Clustering

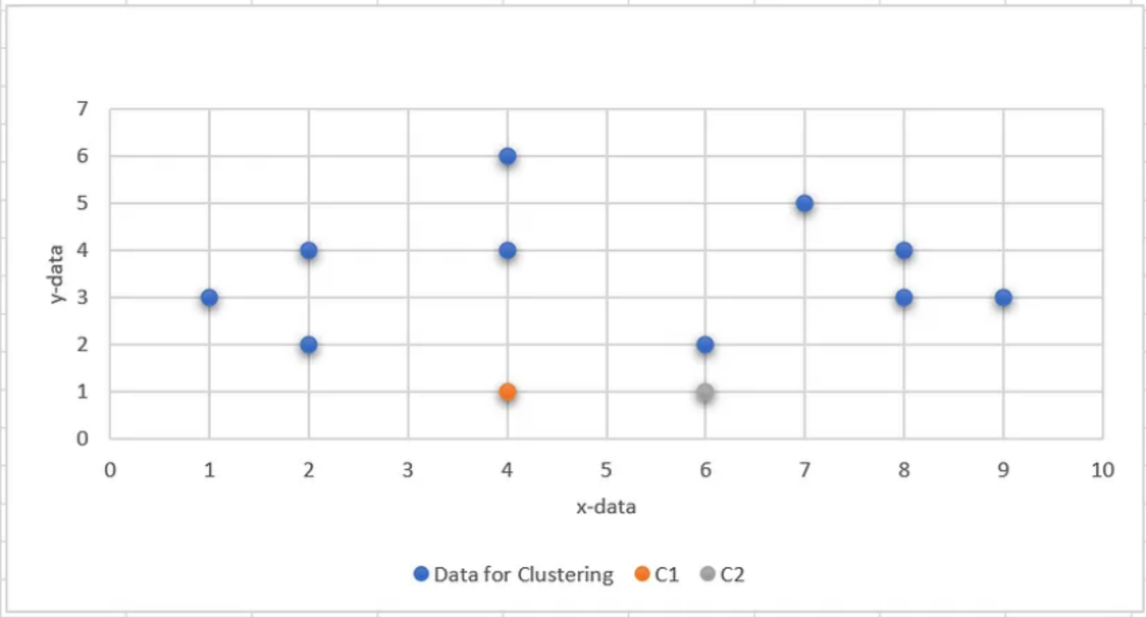
Example

Point	x	y
1	1	3
2	2	2
3	2	4
4	4	6
5	4	4
6	6	2
7	7	5
8	8	4
9	9	3
10	8	3

$$\sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$







Randomly assigned C1, C2

K-Means Clustering

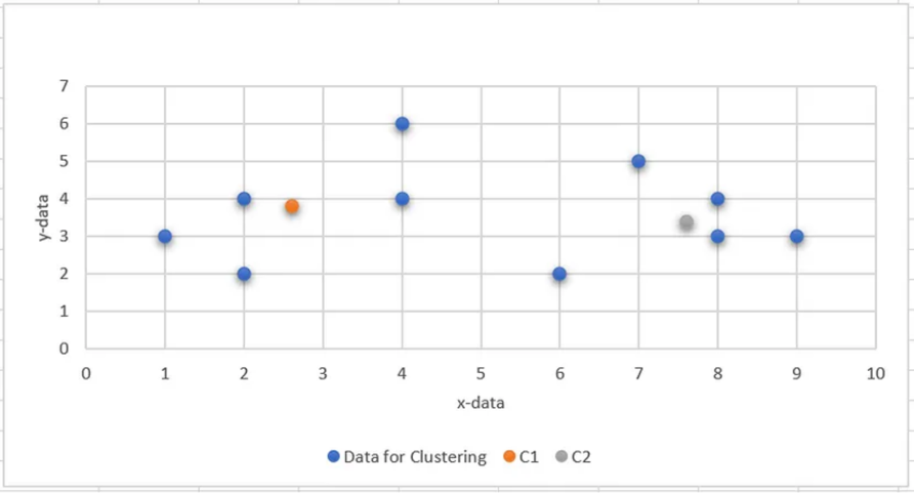
Example

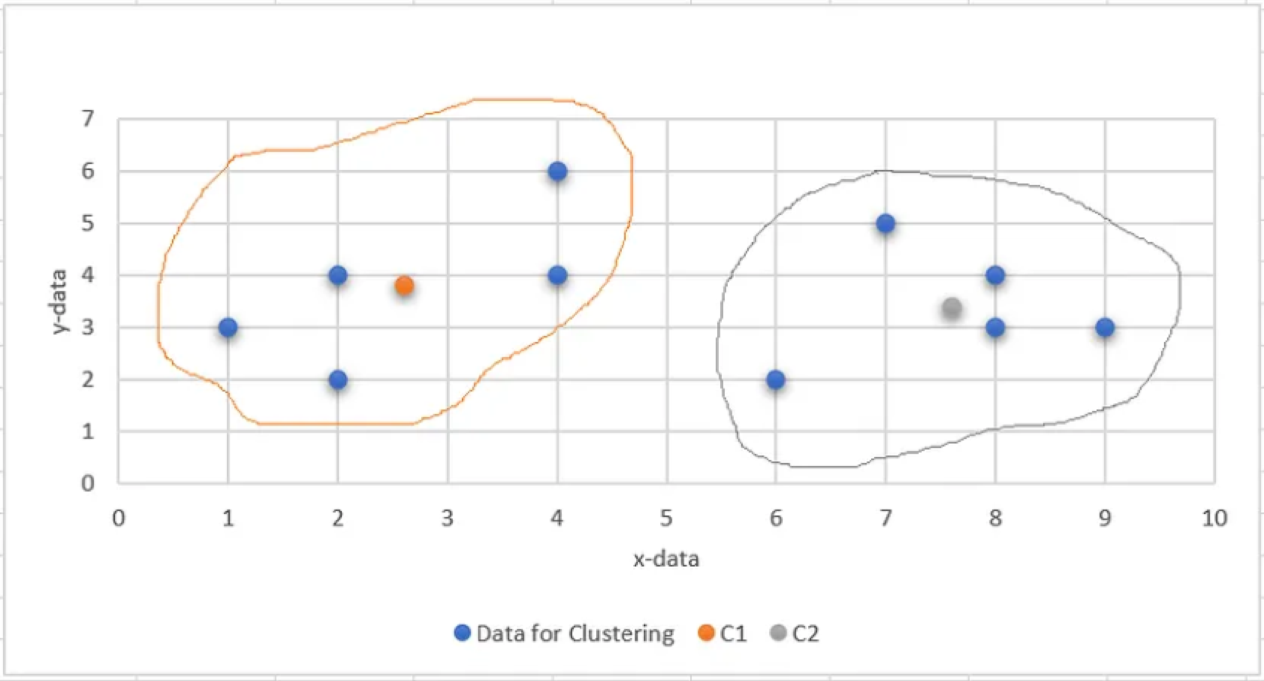
Euclidean distance of each point between the 2 centroids:

x	y	Distance from C1	Distance from C2
1	3	3.6	5.4
2	2	2.2	4.1
2	4	3.6	5
4	6	5	5.3
4	4	3	3.6
6	2	2.2	1
7	5	5	4.1
8	4	5	3.6
9	3	5.4	3.6
8	3	4.4	2.8

	Points assigned to C1		Points assigned to C2	
	x	y	x	y
	1	3	6	2
	2	2	7	5
	2	4	8	4
	4	6	9	3
	4	4	8	3
Mean =	2.6	3.8	7.6	3.4

The new position of centroids will be C1 = (2.6, 3.8) and C2 = (7.6, 3.4)





Adjusted clusters based on the new Centroid coordinates

REPEAT this Process until no change in cluster formed

K-Means Clustering

What happen if the distance between a data point is similar to all the centroids?
Which centroid should we assign the data point?

Equal Euclidean distance of a single data point to all the Cluster Centers

Asked 10 years, 11 months ago Modified 5 years, 9 months ago Viewed 13k times

5

In K means Clustering, suppose, if there exists equal euclidean distance of a data point to all of its k cluster centers, which cluster the data point will choose to become its member? Is there any literature proof supporting it?

clustering k-means

You can just assign your data-point to one cluster centers (random choice), and continue to use K-means until convergence. However, if this is a new data-point that you want to consider after the convergence of your algorithm, and this data-point have "almost" the same distance to all cluster centers, then you may consider creating a new cluster having this data-point as a center. – shn Dec 18, 2012 at 23:18

Evaluating Cluster Quality

- **Within Cluster Sum of Squared error (WCSS)**

$$WCSS = \sum_{C_k}^{C_n}\left(\sum_{d_i \in C_i}^{d_m} distance(d_i, C_k)^2\right)$$

Where *C* is cluster centroid and *d* is data point in each cluster.

Note:

Best result is when WCSS value is lower. It shows a high density cluster.

Evaluating Cluster Quality

- **Silhouette** (how similar is a data point within its own cluster vs. other clusters)

$$s(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}}, if\ |C_I| > 1, \text{ where}$$

a(i) is average distance between i and all other objects in the cluster
b(i) is average distance from i to all clusters

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C, i \neq j} distance(i,j)$$

$$b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} distance(i,j)$$

Note:

S(i) value is always between -1 and +1.

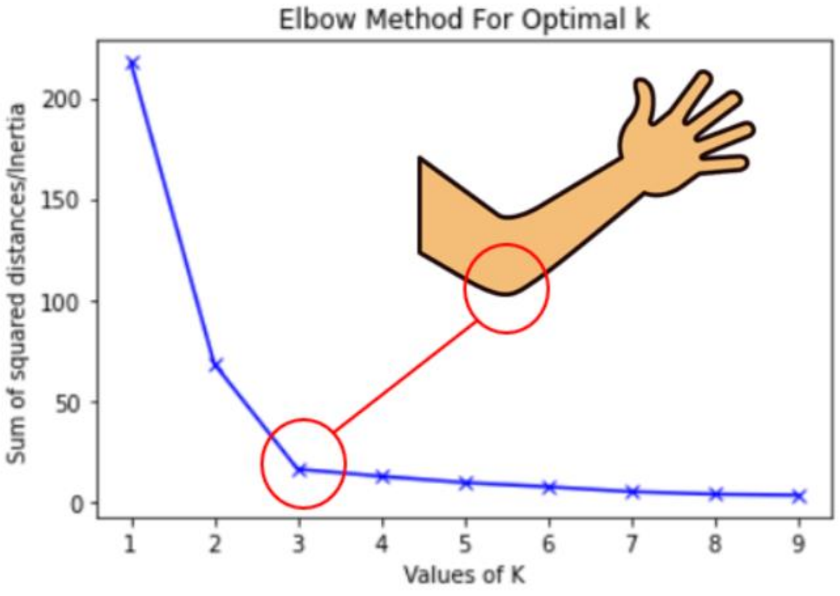
If s(i) close to 0 means that the point is between two clusters.

If *s(i)* is closer to -1, then it would be better off assigning it to the other clusters.

If *s(i)* is close to 1, then the point belongs to the 'correct' cluster.

How to select optimal k?

- Challenge to choose k by ourselves.
- Methods to choose k:
  - Elbow method
  - Silhouette method

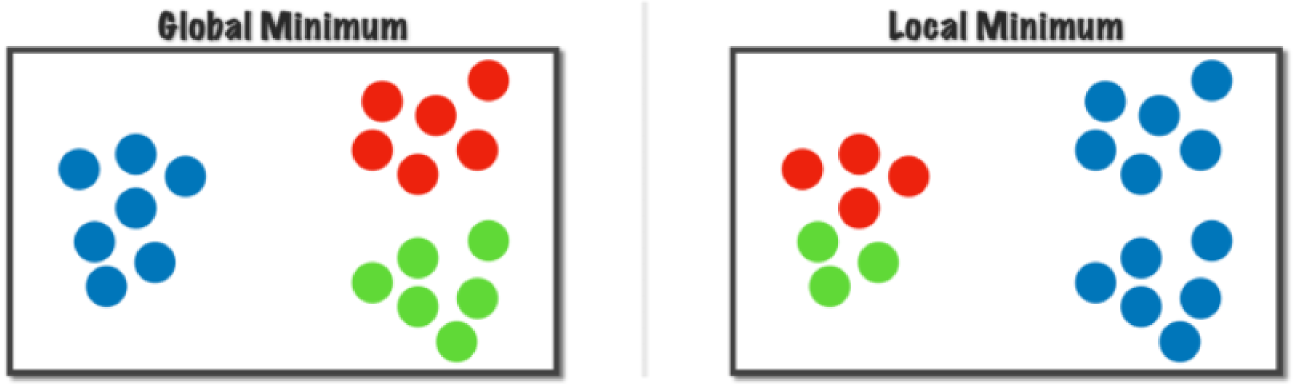


Line plot between K and inertia

Sum of Squares Error (SSE) is the difference between the observed value and the predicted value. Plot the SSE against the number of K. WCSS and Silhouette values can also be used in Elbow Method.

Characteristics of K-Means Clustering

- Guaranteed to converge on local minima (not the best solution).
- Not guaranteed to converge to a global minimum.
- Depending on which values we choose for our initial centroids we may obtain differing results.



Acknowledgments

- Previous STINTSY slides by the following instructors:
  - Courtney Ngo
  - Arren Antioquia
- Reading:

https://alanjeffares.wordpress.com/tutorials/k-means/
https://pub.towardsai.net/one-stop-for-k-means-clustering-b58fa59334e5
https://towardsdatascience.com/k-means-clustering-for-beginners-2dc7b2994a4