



# Machine Learning

Prof. Adil Khan

# Objectives

1. What is clustering? How is it different from classification? Why is it called unsupervised learning?
2. What is k-means? How does it work? What is its objective function? How is it motivated?
3. What is k-means++? Why is it motivated?
4. Limitations of k-means

# Recall: Supervised Learning

- Learning with a teacher
- Teacher provides the supervision
- In the context of machine learning, the labels  $Y$  provides the supervision for  $X$

$X$									$Y$
	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.5	4.5	9.65	1	0.92
1	2	324	107	4	4.0	4.5	8.87	1	0.76
2	3	316	104	3	3.0	3.5	8.00	1	0.72
3	4	322	110	3	3.5	2.5	8.67	1	0.80
4	5	314	103	2	2.0	3.0	8.21	0	0.65

# Recall: Supervised Learning (2)

- Easy and well-defined
- Given  $Y = f(X)$  and labeled dataset

$$D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N, \quad \mathbf{x} \in \mathbb{R}^d$$

- Estimate the function  $f$  as  $\hat{f}$ , such that when a new input data is given, we can predict its outcome.

# Unsupervised Learning

- We have only predictors (a.k.a inputs, or features) but no labels

$$D = \{\mathbf{x}_i\}_{i=1}^N, \quad \mathbf{x} \in \mathbb{R}^d$$

- Then what is the goal of learning?

# Scenario (1)

- Imagine you run an online store and would like to personalize your customers' shopping experience
- You think you can do this by providing each customer with personalized recommendations
- You do not know each user's personal preferences and tastes but you have lots of data on their purchases
- How can you use this data to create recommendations?

# Scenario (2)

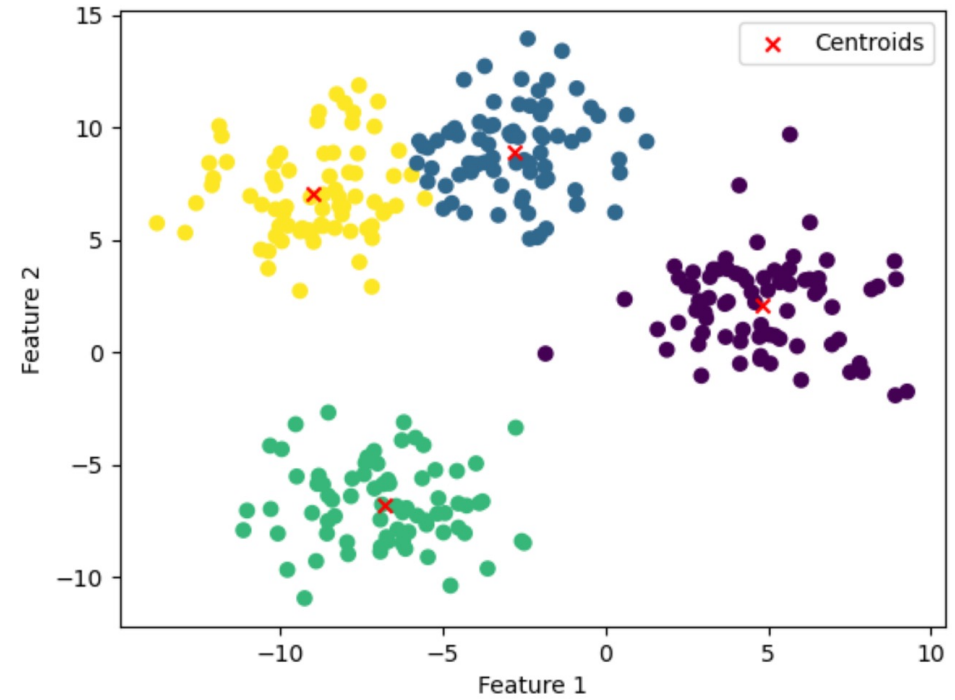
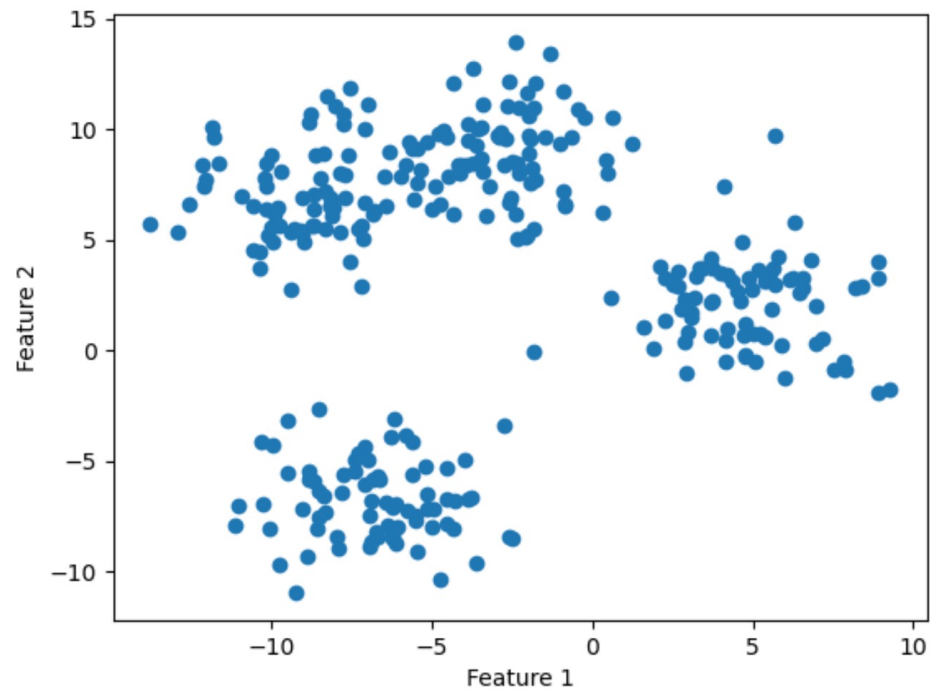
- Imagine you are a biologist interested in studying the behavior of bees
- You have collected a lot of videos (or other data on them)
- How can you use this data to study bee behavior?

# Unsupervised Learning

- In the discussed scenarios, we are not interested in prediction, because we do not have an associated response variable
- Instead, the unsupervised **learning goal** is to model the **hidden patterns** or **underlying structure** in the given input data in order to learn about the data
- This underlying structure is what we usually refer to as groups of data
- And these groups are what we refer to as **Clusters**



# Clustering



# Key Takeaway

## Unsupervised

- No labels provided
- Finds structure in unlabelled data
- Uses techniques such as clustering or dimensionality reduction.

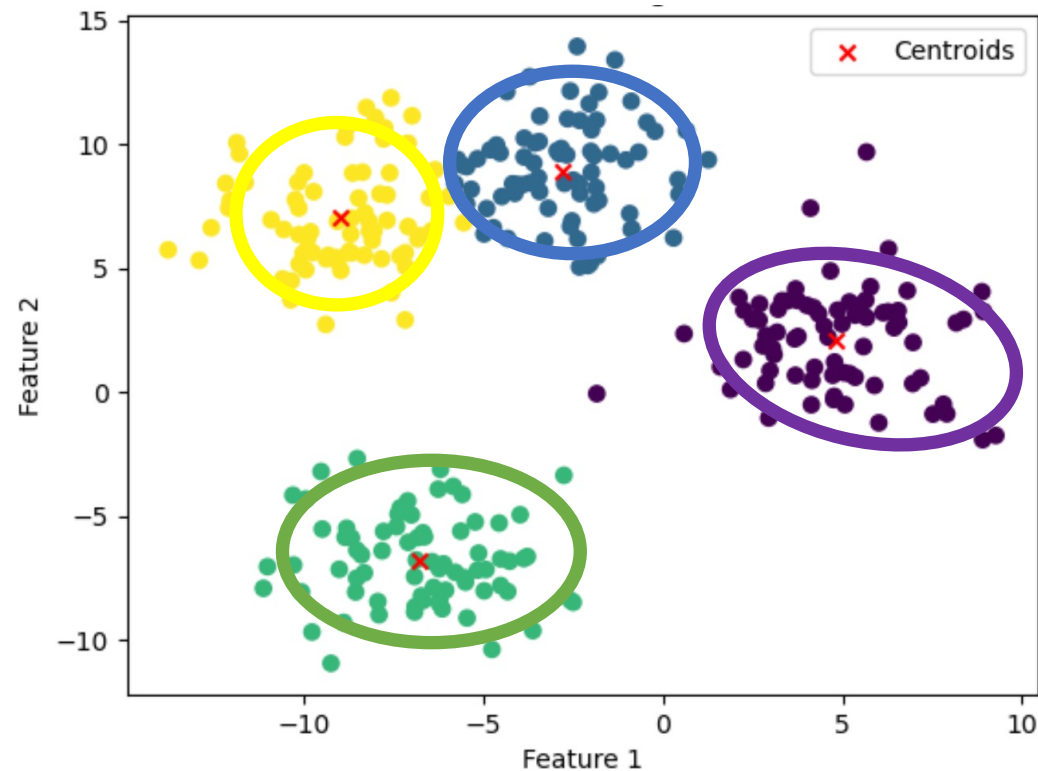
## Supervised

- Labels provided
- Finds patterns in existing structure
- Uses techniques such as regression or classification.

Let's first learn some basic  
things about clustering

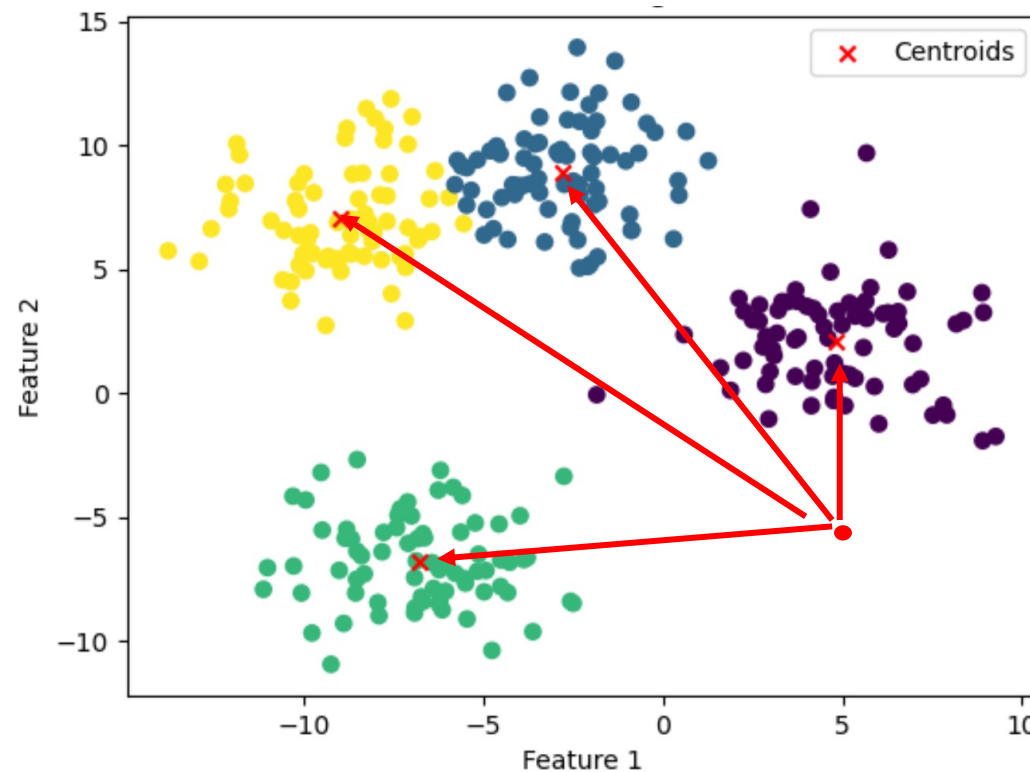
# What defines a cluster?

- Clusters are defined by a **center** and a **spread** (which you can also call shape)



# Assigning a Point to a Cluster

- And we assign a given point to a cluster by computing its similarity ( using distance ) to the centers of the clusters, and choosing the one that is the most similar



New data point

# Similarity

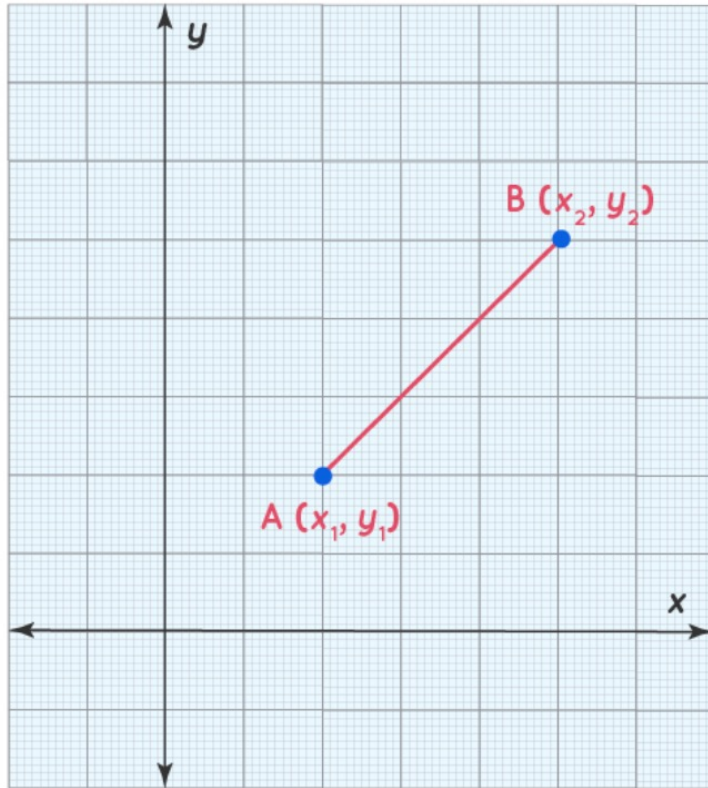
- Central to all of the goals of cluster analysis is the notion of the **degree of similarity (or dissimilarity)** between the individual objects being clustered.
- A clustering method attempts to group the objects based on the definition of similarity (which is **usually measured in the form of a distance from the center of a cluster**) supplied to it.

# Multiple Choices For Measuring Dissimilarity

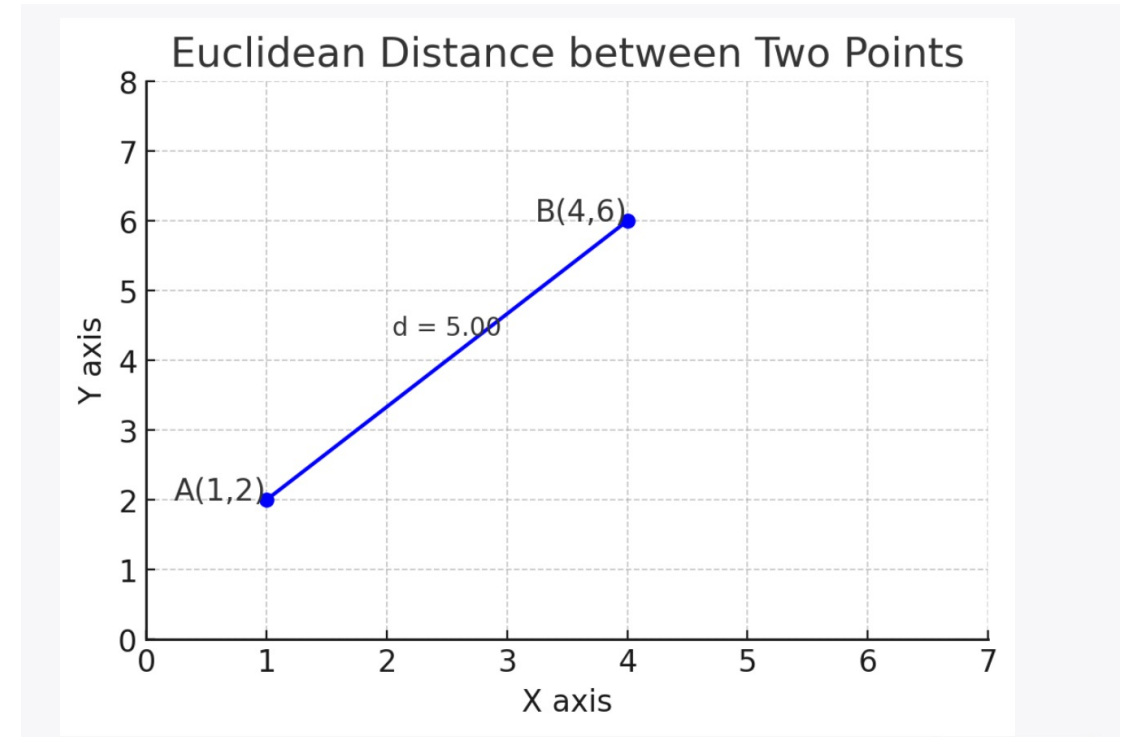
- *Euclidean distance (the most common)*
- Manhattan distance
- Correlation based distances
  - Pearson correlation distance
  - Eisen cosine correlation distance
  - Spearman correlation distance
  - Kendall correlation distance
- Etc.

# Euclidean Distance

## Euclidean Distance Formula



$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

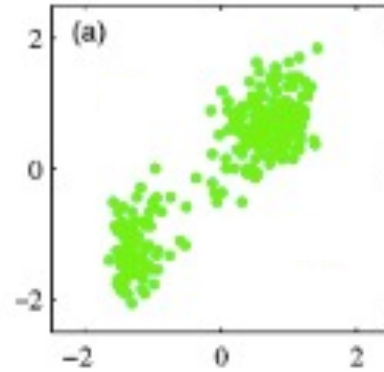




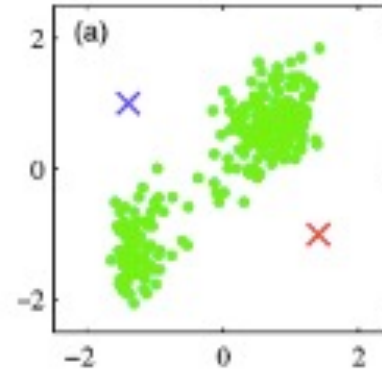
# k-means Clustering

# k-means Overview

- We are given unlabelled data points

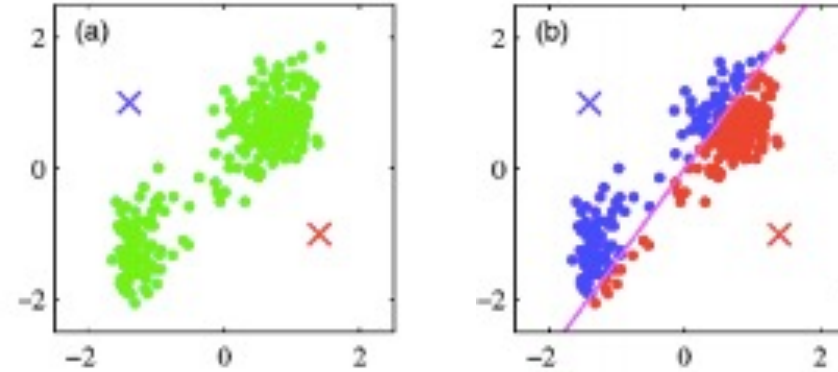


# k-means Overview



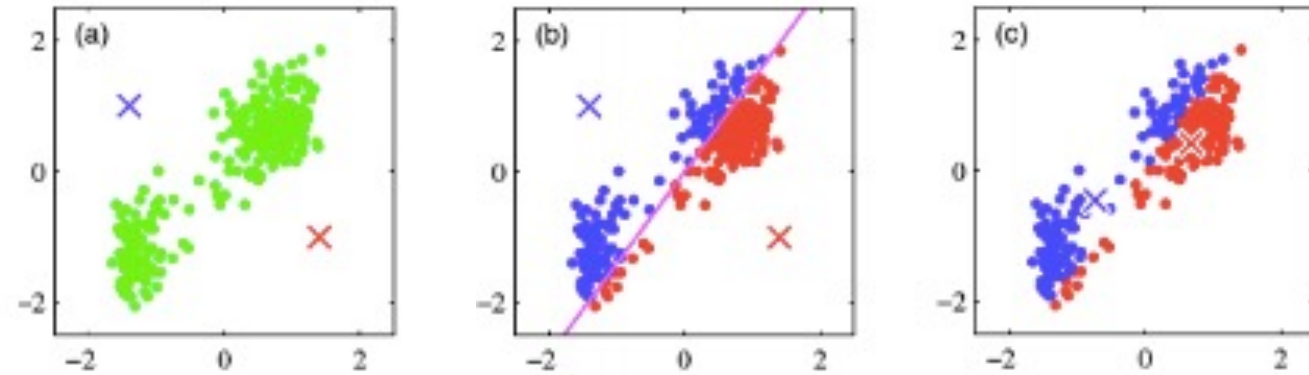
- We are given unlabelled data points
- 1. We choose  $K$  centroids ( $K$  represents the anticipated number of distinct clusters).
- 2. K-means randomly positions the  $K$  centroids within the data.

# k-means Overview



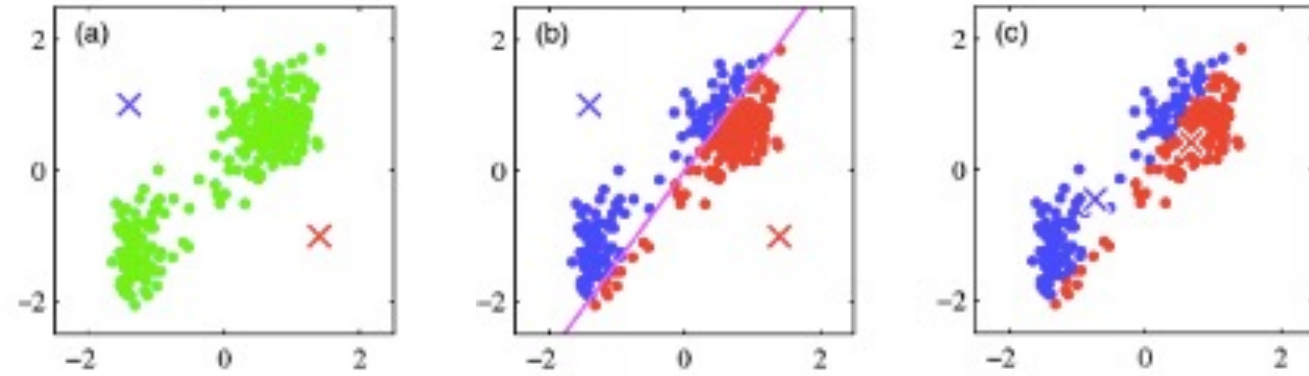
- We are given unlabelled data points
  1. We choose K centroids (K represents the anticipated number of distinct clusters).
  2. K-means randomly positions the K centroids within the data.
  3. It computes the Euclidean distance between each centroid and all the points in the data.
  4. And assigns each training data point to the closest centroid, forming groups.

# k-means Overview



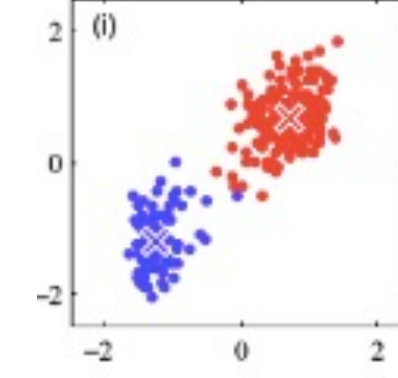
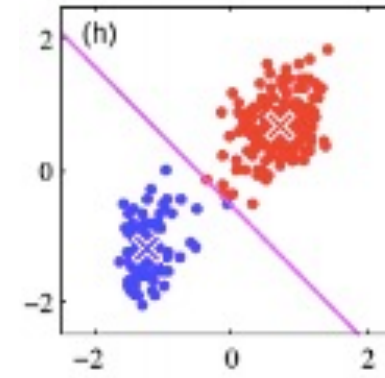
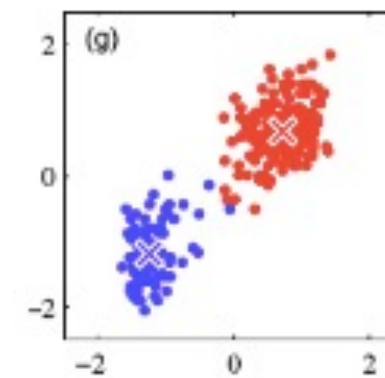
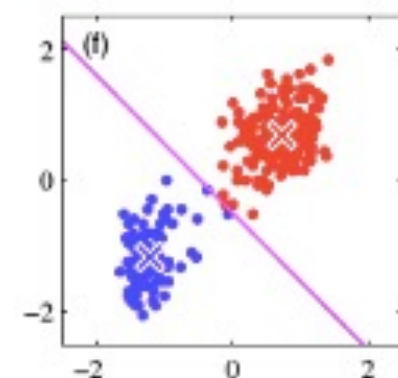
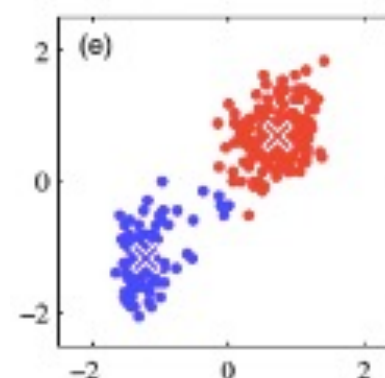
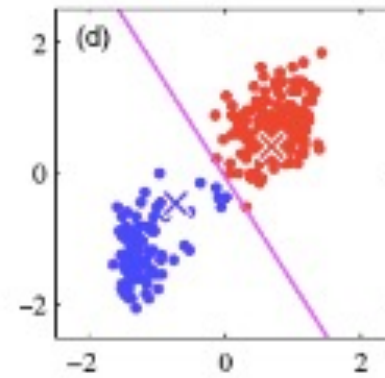
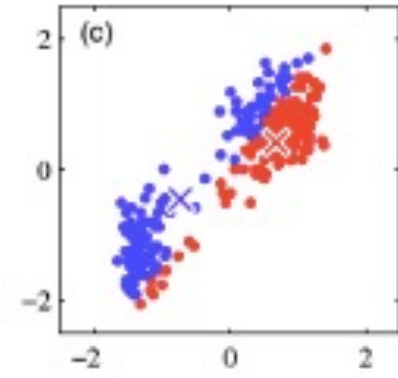
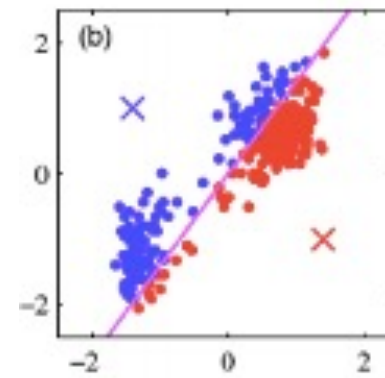
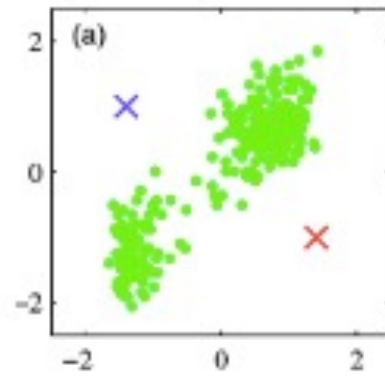
- We are given unlabelled data points
1. We choose K centroids (K represents the anticipated number of distinct clusters).
  2. K-means randomly positions the K centroids within the data.
  3. It computes the Euclidean distance between each centroid and all the points in the data.
  4. And assigns each training data point to the closest centroid, forming groups.
  5. Within each group, it calculates the average data point and moves the corresponding centroid to that position.

# k-means Overview



- We are given unlabelled data points
  1. We choose K centroids (K represents the anticipated number of distinct clusters).
  2. K-means randomly positions the K centroids within the data.
  3. It computes the Euclidean distance between each centroid and all the points in the data.
  4. And assigns each training data point to the closest centroid, forming groups.
  5. Within each group, it calculates the average data point and moves the corresponding centroid to that position.
  6. The algorithm continues this process (steps 3 - 5) until convergence is reached, which occurs when there are no more changes in group membership.

# k-means Overview



# k-means Clustering Pseudocode

- k-means simply works as follows
  - **First Step:** Initialization - randomly initialize cluster centers
  - **Second Step:** The algorithm then iteratively alternates between the following two steps
    - **Assignment Step:** assign each data point to the closest center
    - **Refitting Step:** move each cluster center to the mean of the data points assigned to it



Let's Formulate Clustering as an  
Optimization Problem

# First Thing to Understand

1. Clustering takes unsupervised data (only  $X$ ) as input – Let  $N$  be the total number of data points
2. And returns two things:
  - $K$  cluster centers
  - $N$  cluster assignments

# Deriving Clustering Objective (1)

- Given  $D = \{\mathbf{x}_i\}_{i=1}^N$  Find cluster centers  $\{\mathbf{c}_k\}_{k=1}^K$  and assignments  $\{\mathbf{a}_i\}_{i=1}^N$  to minimize the sum of squared distances of data points to their assigned centers

$$\mathbf{x}_i \in \mathbb{R}^d$$

$$\mathbf{c}_k \in \mathbb{R}^d$$

$$\mathbf{a}_i \in \mathbb{R}^k$$

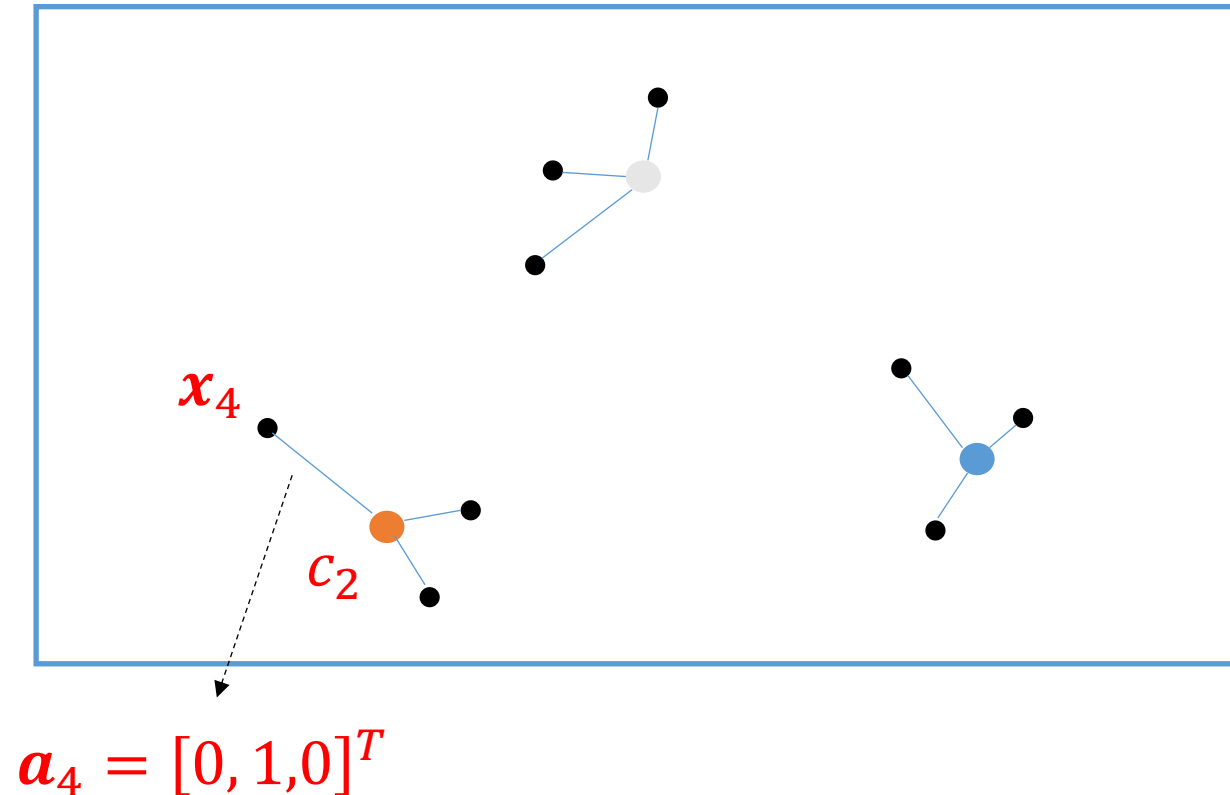
# Deriving Clustering Objective (2)

- Given  $D = \{\mathbf{x}_i\}_{i=1}^9$  Find cluster centers  $\{\mathbf{c}_k\}_{k=1}^3$  and assignments  $\{\mathbf{a}_i\}_{i=1}^9$  to minimize the sum of squared distances of data points to their assigned centers

$$\mathbf{x}_i \in \mathbb{R}^2$$

$$\mathbf{c}_k \in \mathbb{R}^2$$

$$\mathbf{a}_i \in \mathbb{R}^3$$



# Deriving Clustering Objective (3)

- Optimization problem:

$$\arg \min_{c_k, a_i} \sum_{i=1}^N \sum_{k=1}^K a_i^k d(c_k, x_i)$$

- Problem is hard when minimizing jointly
- But becomes easy when we fix one and minimize over the other

# How to Solve the Optimization Problem? (2)

- That is:
  - First we fix the centers, and find the optimal cluster assignments by assigning each point to the cluster with the nearest neighbor

# How to Solve the Optimization Problem? (3)

- Next:
  - For the found (and fixed) assignments, we can find the optimal cluster centers by setting each cluster's centers to the average of its assigned data points

# How to Solve the Optimization Problem? (4)

- Thus

$$\arg \min_{c_k, a_i} \sum_{i=1}^N \sum_{k=1}^K a_i^k d(c_k, x_i)$$

- We solve this optimization problem by alternating between minimizing our objective with respect to  $\{c_k\}$  and  $\{a_i\}$
- This is called [alternating minimization](#)



# k-means Clustering Pseudocode

➤ Given:

- $K$ , data  $X = \{X_1, X_2, \dots, X_N\}$

➤ Goal:

- Cluster  $X$  into  $K$  clusters

- $D(X_i, c_j)$ : the between instance  $X_i$  and cluster centre  $c_j$ ;

- Euclidian distance:

$$D(X_i, c_j)^2 = \sum_{k=1}^p (X_{ik} - c_{jk})^2$$

❑ Input:  $K$ , data points  $X_1, X_2, \dots, X_N$

❑ Place centroids  $c_1, c_2, \dots, c_K$  at random locations

❑ Repeat until convergence:

- For each point  $X_i$ :
  - Find nearest centroid  $c_j$ :  
 $\operatorname{argmin}_j D(X_i, c_j)$
  - Assign the point  $X_i$  to cluster  $j$
- For each cluster  $j = 1, 2, \dots, K$ :
  - New centroid  $c_j$  is set to the mean of all points  $X_i$  assigned to cluster  $j$  in previous step

❑ Stop when none of the cluster assignments change

# Convergence of k-means

Converges to:

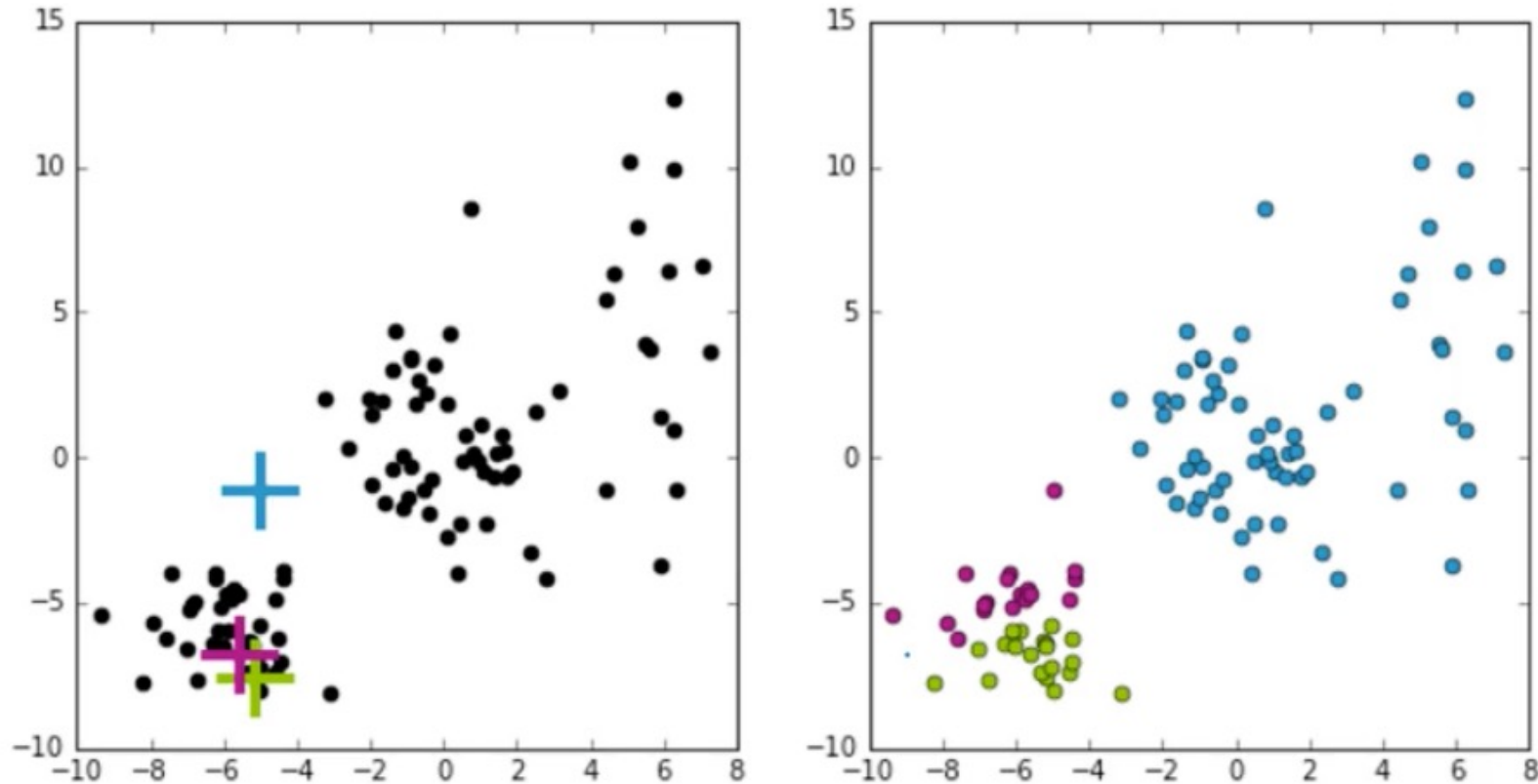
~~- Global optimum~~

- Local optimum

~~- neither~~

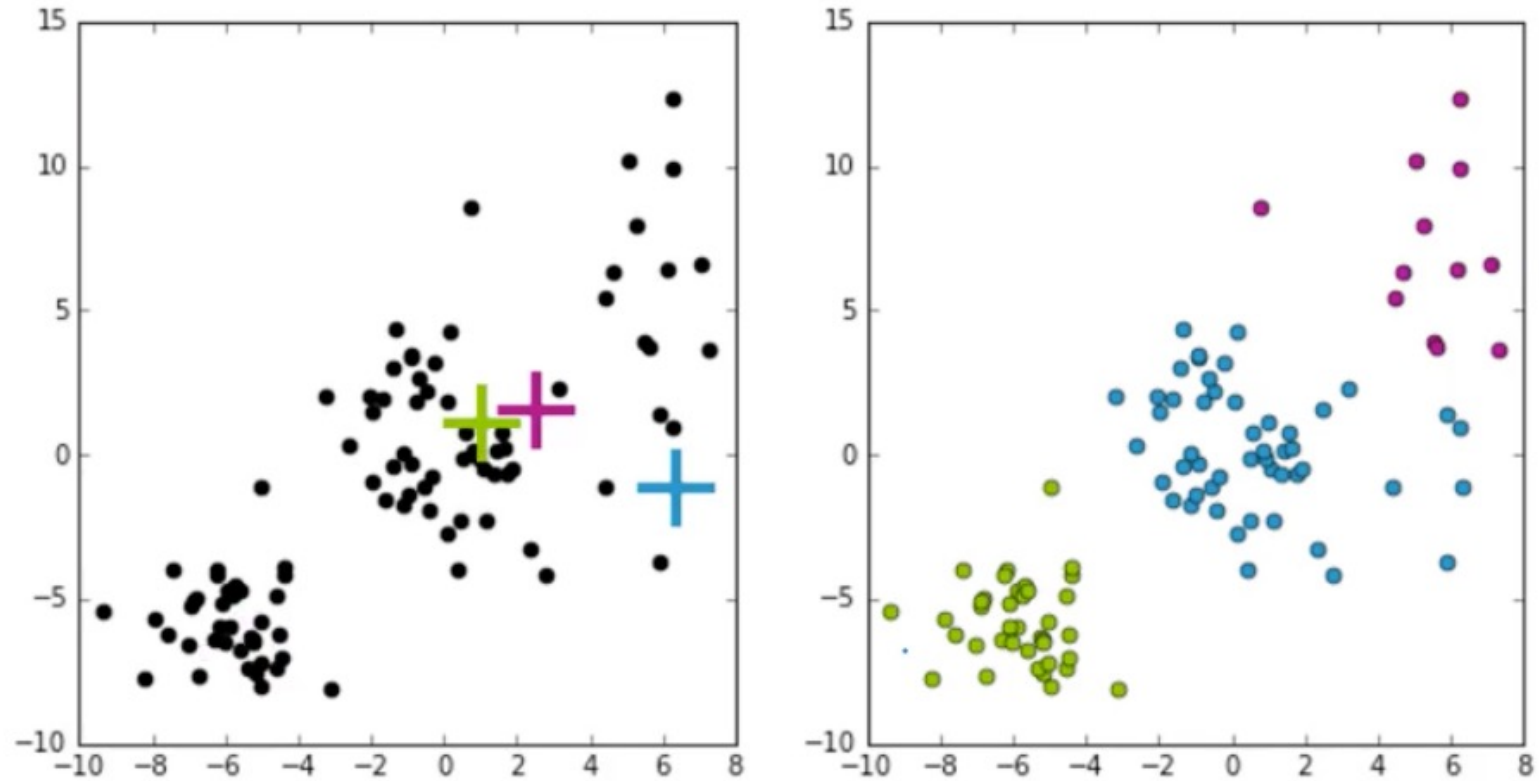
The first one does not mean that it can never converge to global optimum, it is just that there is no guarantee that it will.

# Initialization Matters



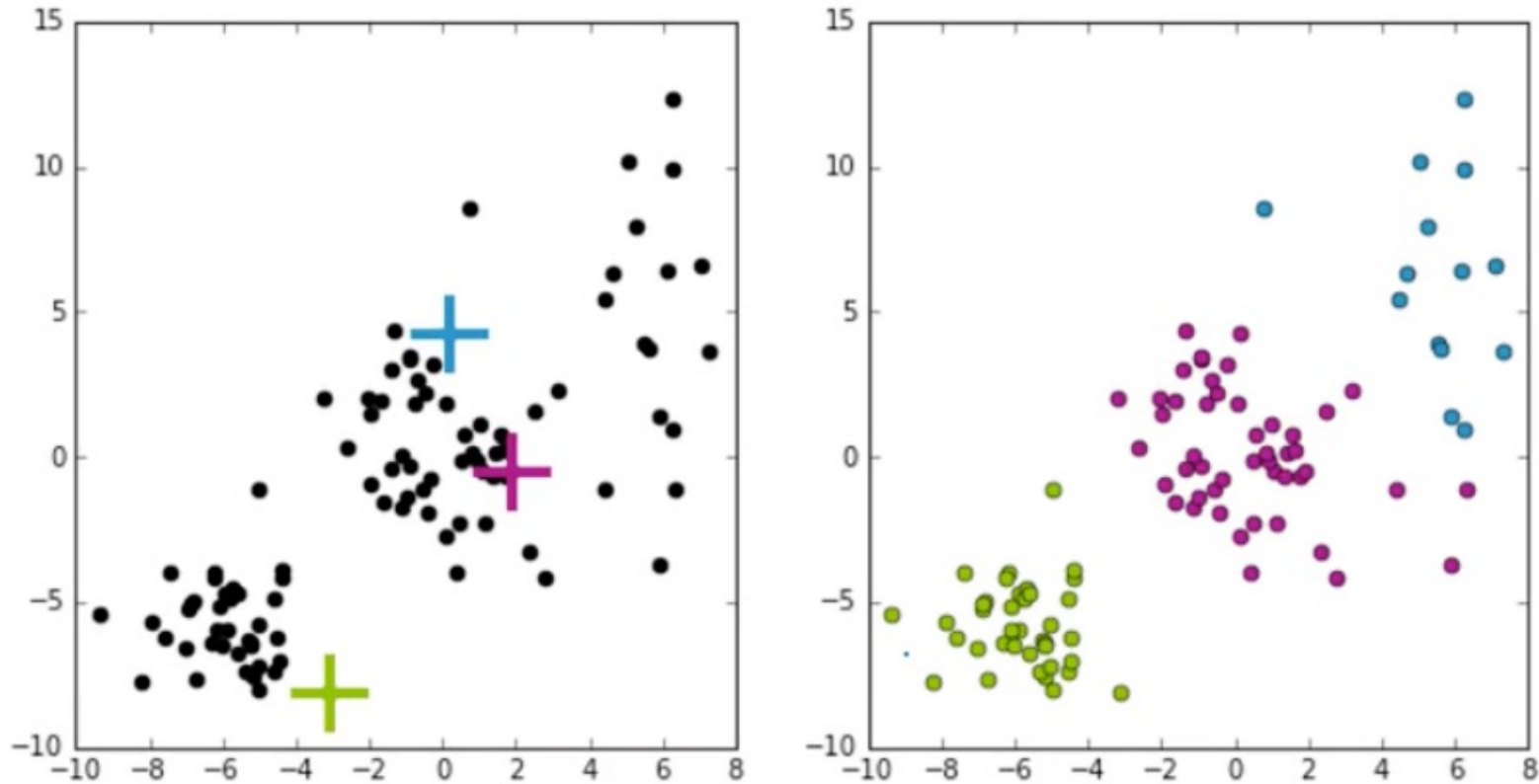
Centroids are chosen as shown in the first figure

# Initialization Matters (2)



Different Initialization

# Initialization Matters (3)



An even different initialization

# Bottom-line

**k-means is sensitive to center initialization** and we can get completely different solutions, by converging to a local mode

It **can be improved** by using a specific way of choosing centers

k-means++

# k-means++

- The intuition behind this approach is that **spreading out the  $k$  initial cluster centers** is a good thing
- The **first cluster center is chosen uniformly at random** from the data points that are being clustered
- After which **each subsequent cluster center is chosen from the remaining data points with probability proportional to its squared distance** from the existing cluster center.



# How does k-means++ work?

- Let  $d(x)$  denote the shortest distance from a data point to the closest center we have already chosen
- Then k-means ++ works as follows
  1. Take the first center  $m_1$ , chosen uniformly at random from  $D$
  2. Take the next center, choosing  $x_i$  with probability  $\frac{d(x_i)^2}{\sum_{x_j \in X} d(x_j)^2}$
  3. Repeat step 2 until we have chosen K cluster centers
  4. Proceed with the standard k-means algorithm

# Example

- Let's say we have four data points **A**, **B**, **C** and **D**
- We have already chosen the first centroid as point **A**, and need to choose the second centroid from the remaining point
- First, we compute the squared distance from each of these points to point **A** (since it is the nearest centroid) – Let's say this turns out to be:

Point	Distance
B	1
C	4
D	9

# Example

Point	Distance
B	1
C	4
D	9

- Now, to choose the next centroid, we assign probabilities to each of these points proportional to their distances
- The sum of the squared distance is  $1+4+9 = 14$ , thus the probabilities will turn out to be

Point	Probability
B	$1/14$
C	$4/14$
D	$9/14$

# Example

Point	Probability
B	1/14
C	4/14
D	9/14

- We then choose the next centroid based on these probabilities
- Point **D** has the highest probability in this case
- This process ensures that data points that are farther away from the existing centroid are most likely to be chosen

# k-means++ Summary

- Smart initialization is computationally costly relative to random initialization
- However, due to smart initialization, the subsequent k-means often converges more rapidly
- Tends to improve the quality of the local optimum

How to choose  $k$  for k-means?

# Hyperparameters

1. Hyperparameters are the choice that we make about the model
2. Examples:
  - Polynomial Regression: Degree of the polynomial
  - k-means: Value of k (number of clusters)
3. It is important to choose their optimum values

# k-means Objective

$$\min_{\{\mathbf{m}_k\}, \{\mathbf{r}^{(n)}\}} \sum_{n=1}^N \sum_{k=1}^K r_k^{(n)} \|\mathbf{m}_k - \mathbf{x}^{(n)}\|^2$$



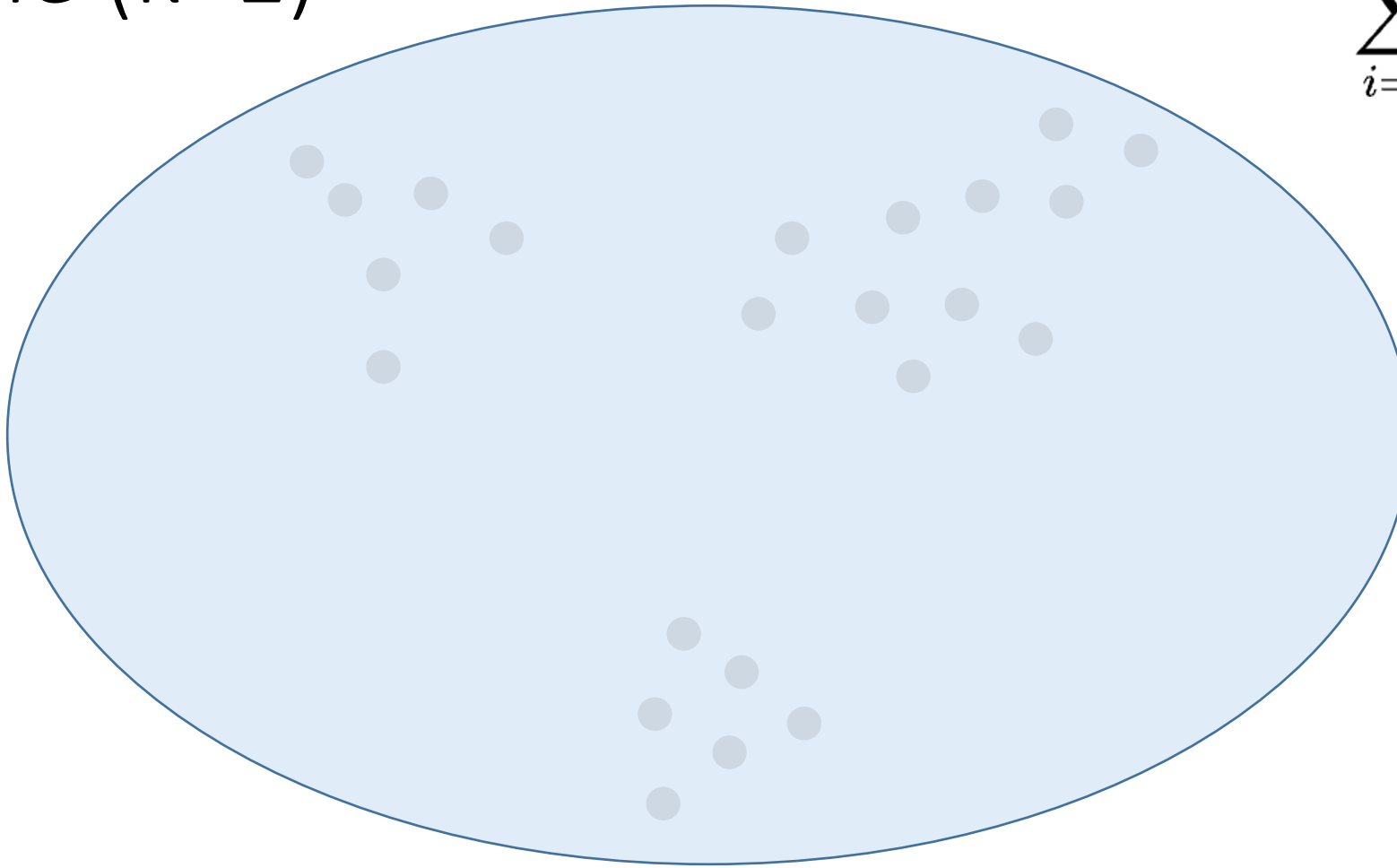
# Example

$$\sum_{i=1}^N \sum_{k=1}^K a_i^k d(c_k, x_i)$$



# Example (k=1)

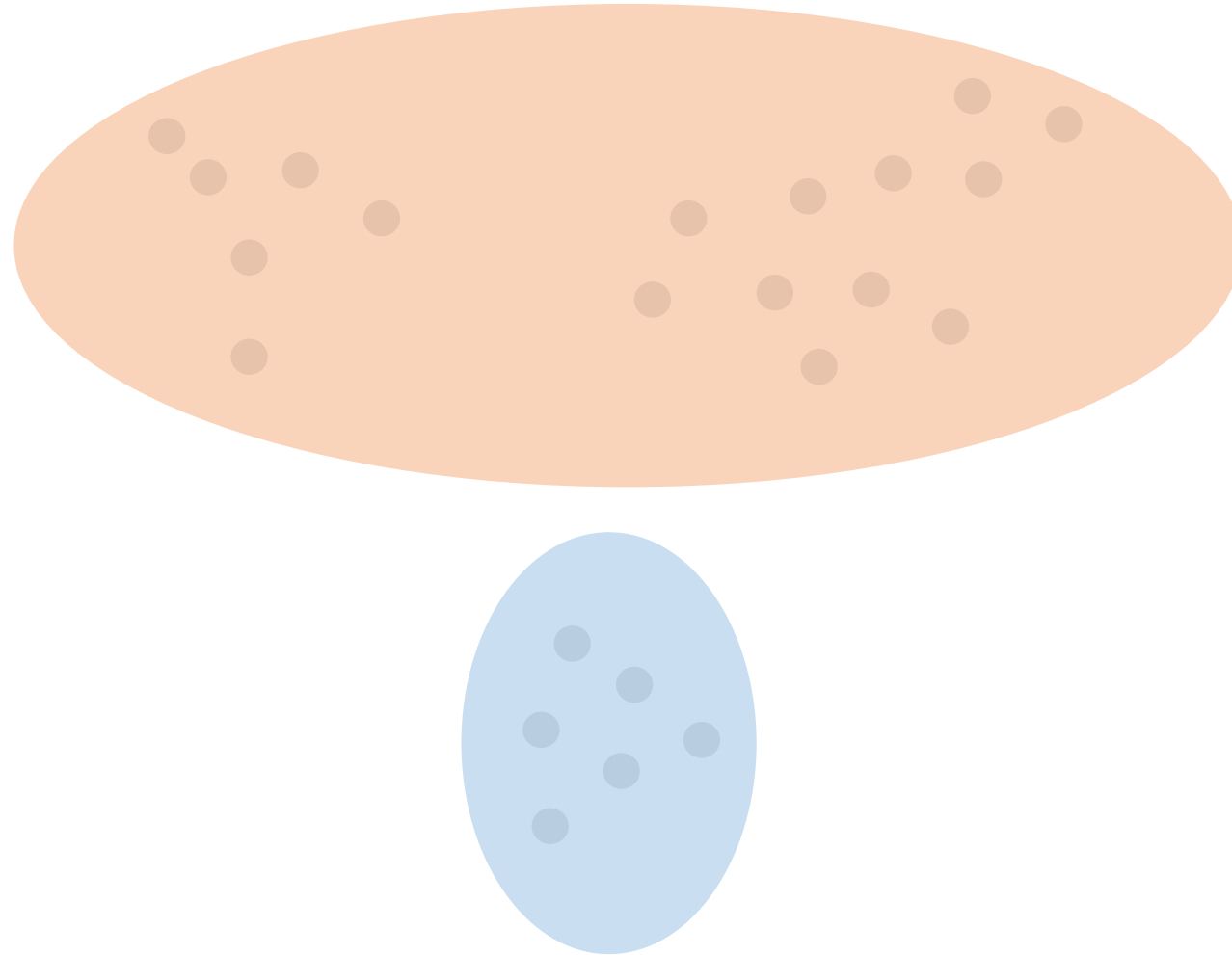
$$\sum_{i=1}^N \sum_{k=1}^K a_i^k d(c_k, x_i)$$



Center will be somewhere in the middle of this sphere, and the sum of distances will be some large positive value

# Example (k=2)

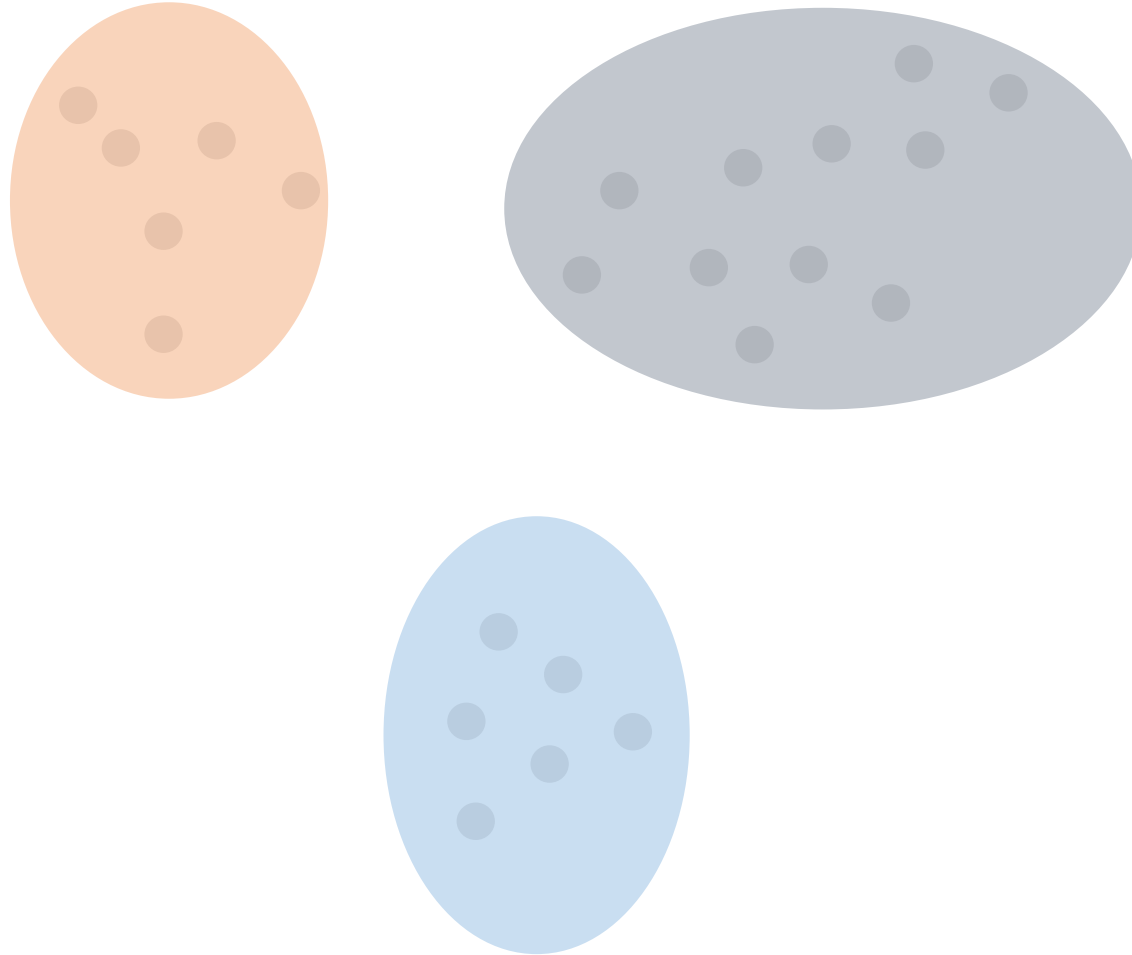
$$\sum_{i=1}^N \sum_{k=1}^K a_i^k d(c_k, x_i)$$



Each point is now more close its center as compared to the previous case, so the sum of distances would decrease

# Example (k=3)

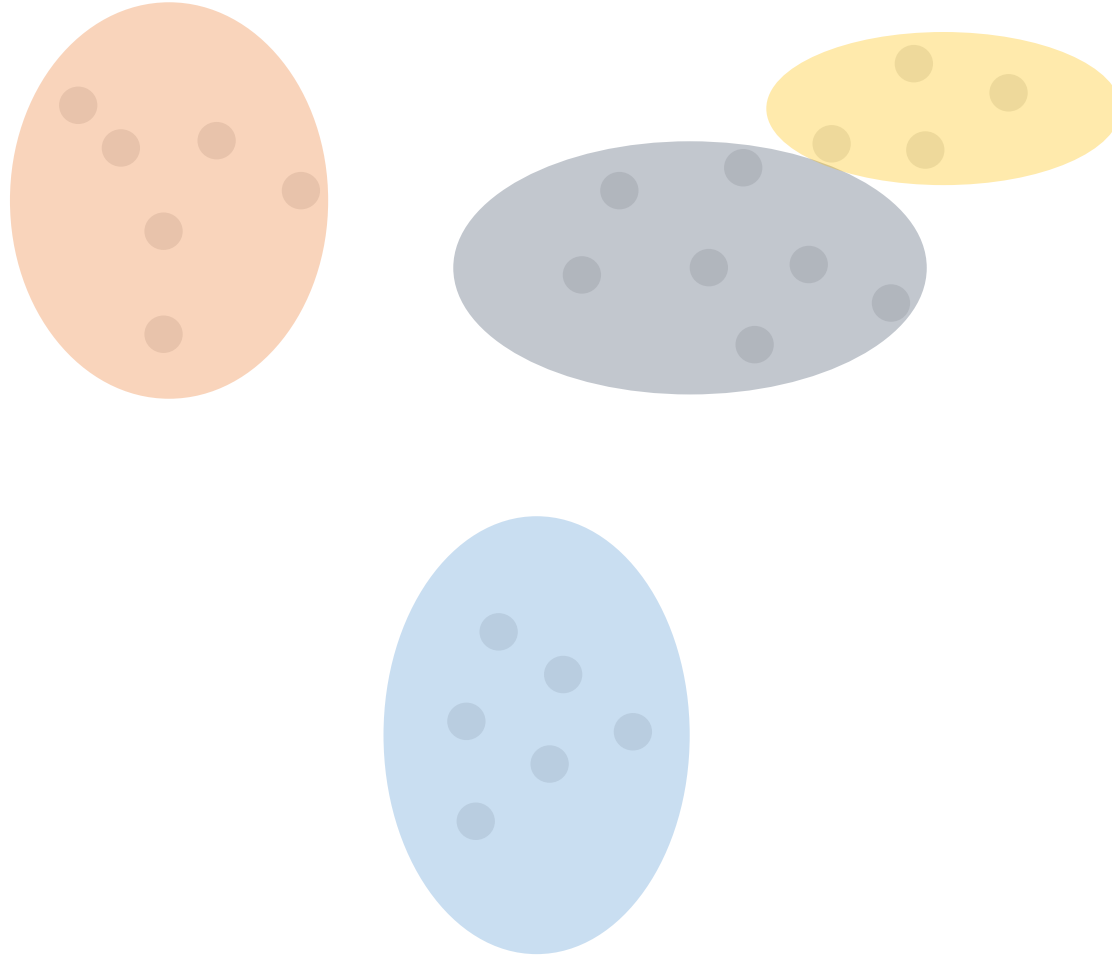
$$\sum_{i=1}^N \sum_{k=1}^K a_i^k d(c_k, x_i)$$



The sum would decrease even further

# Example (k=4)

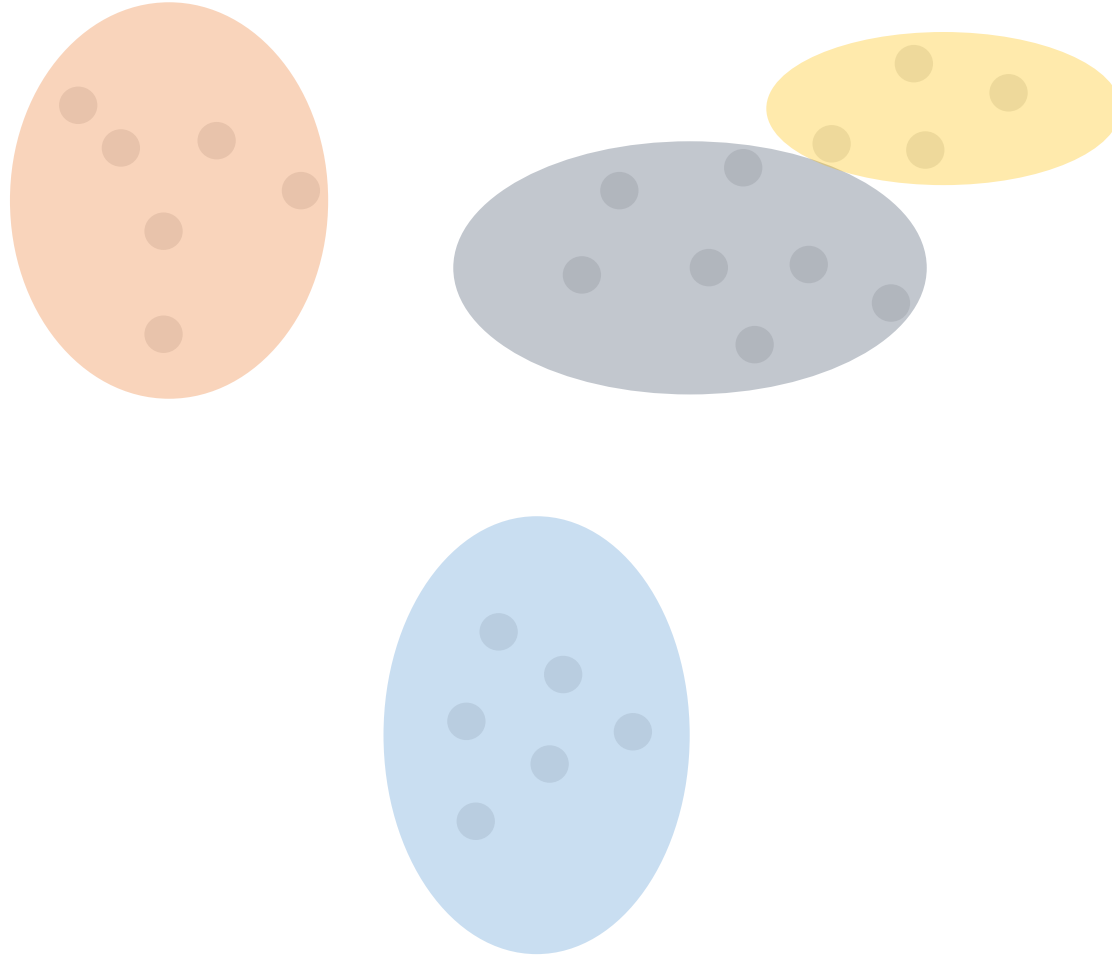
$$\sum_{i=1}^N \sum_{k=1}^K a_i^k d(c_k, x_i)$$



Even further decreases!!!

# Example

$$\sum_{i=1}^N \sum_{k=1}^K a_i^k d(c_k, x_i)$$



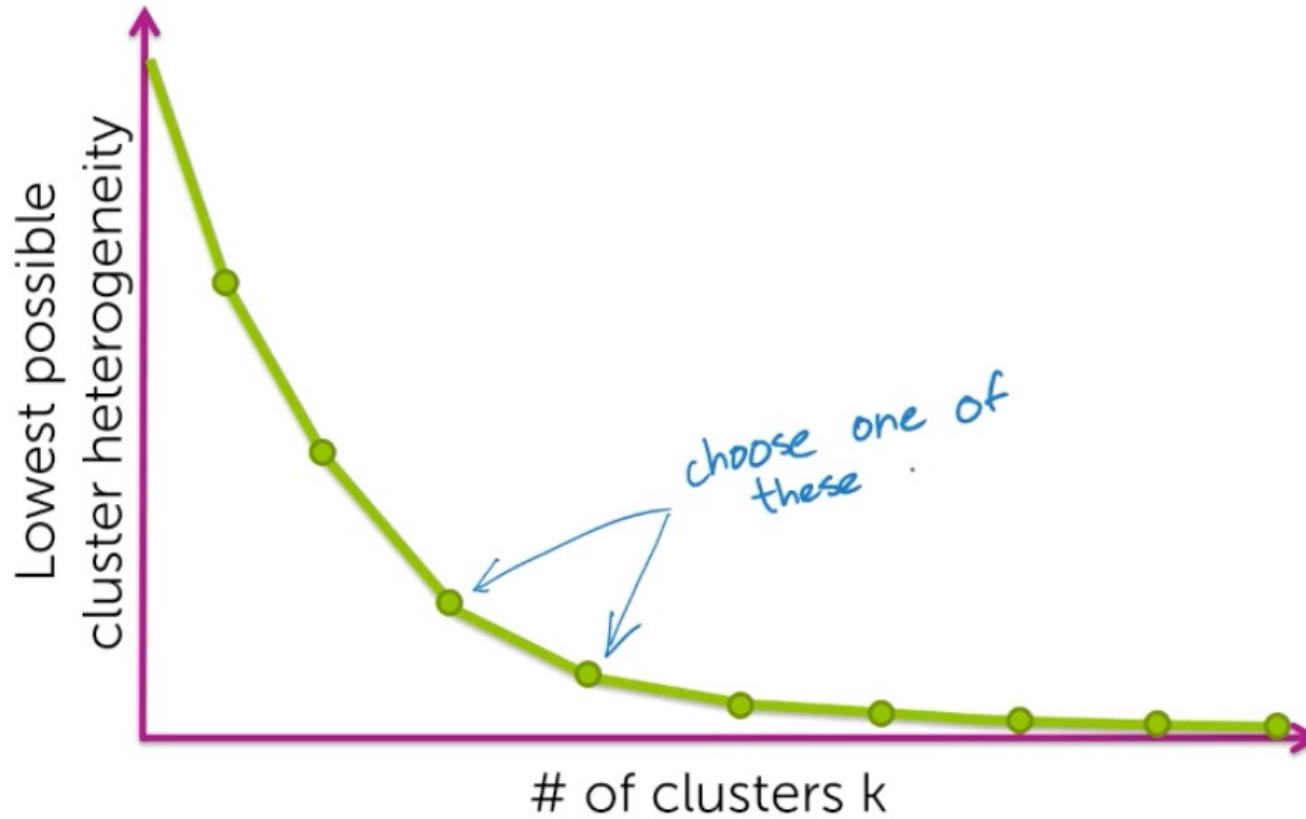
Now, take a moment and think about the case where set  $k = N$

The sum will become 0!!

# Overfitting in Clustering

- Remember, we are in unsupervised learning.
  - In this case, overfitting would mean that we did not learn the true structures in the data, instead we ended up memorizing the position of the sample in the space.

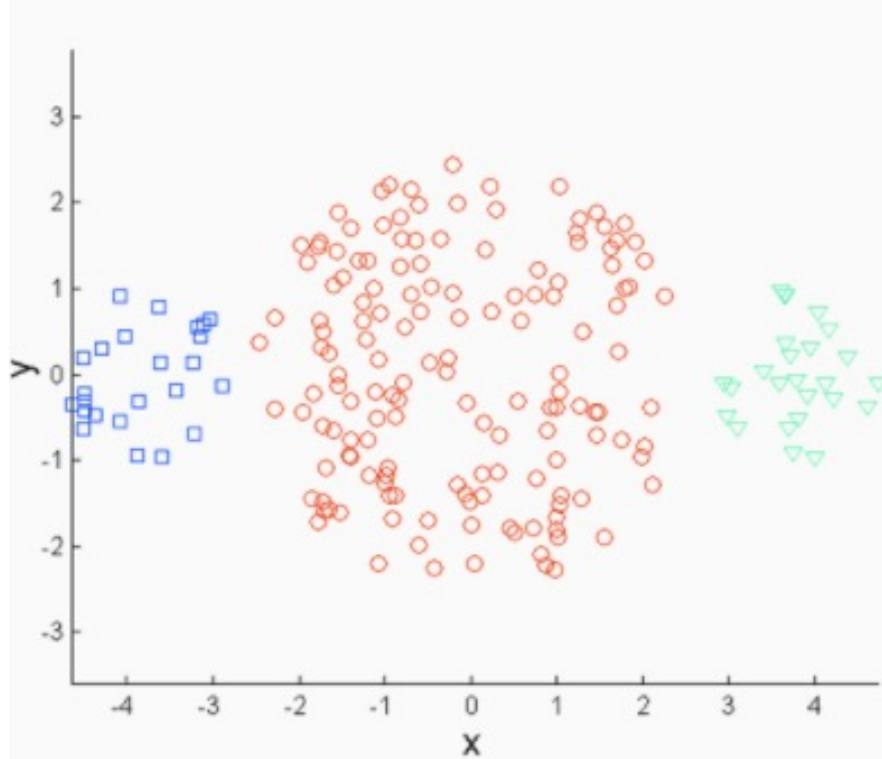
# Choosing k – The Elbow Method



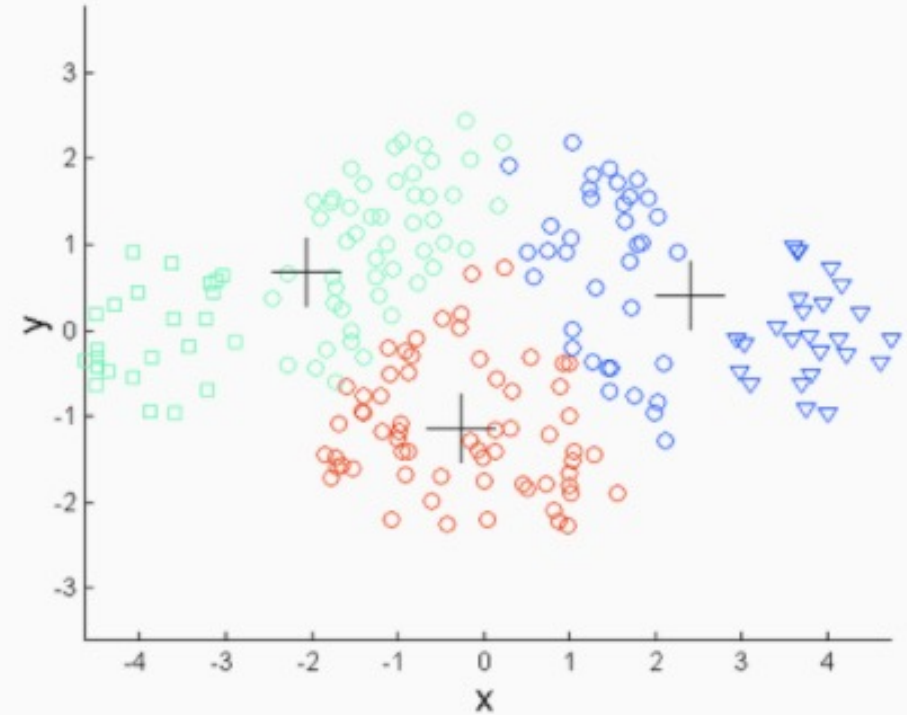
Elbow of the curve



# Issues with k-means (1)

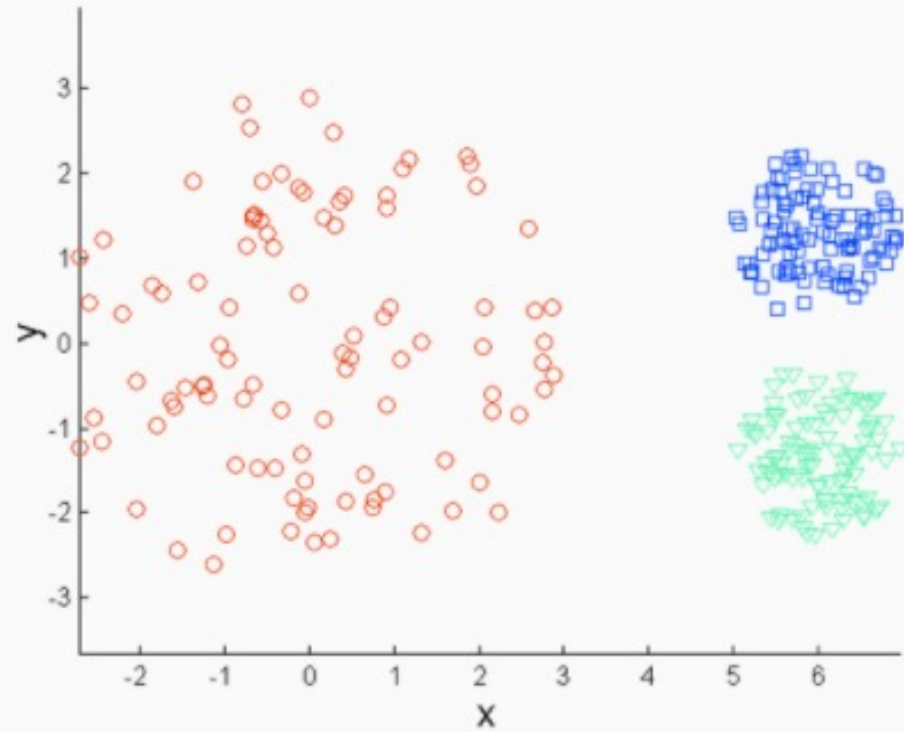


**Original Points**

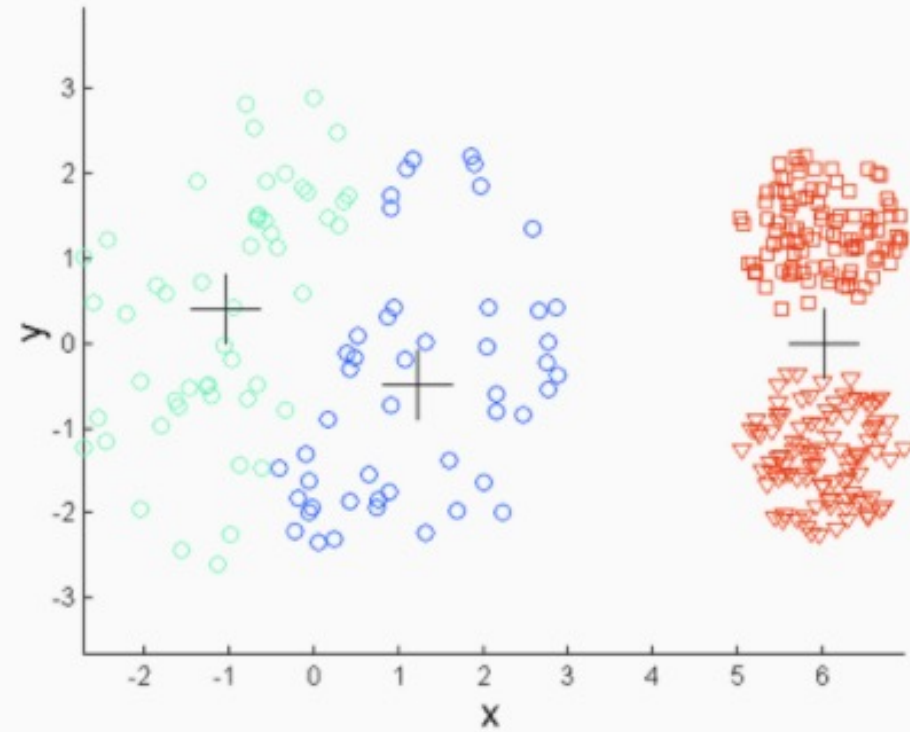


**K-means (3 Clusters)**

# Issues with k-means (2)

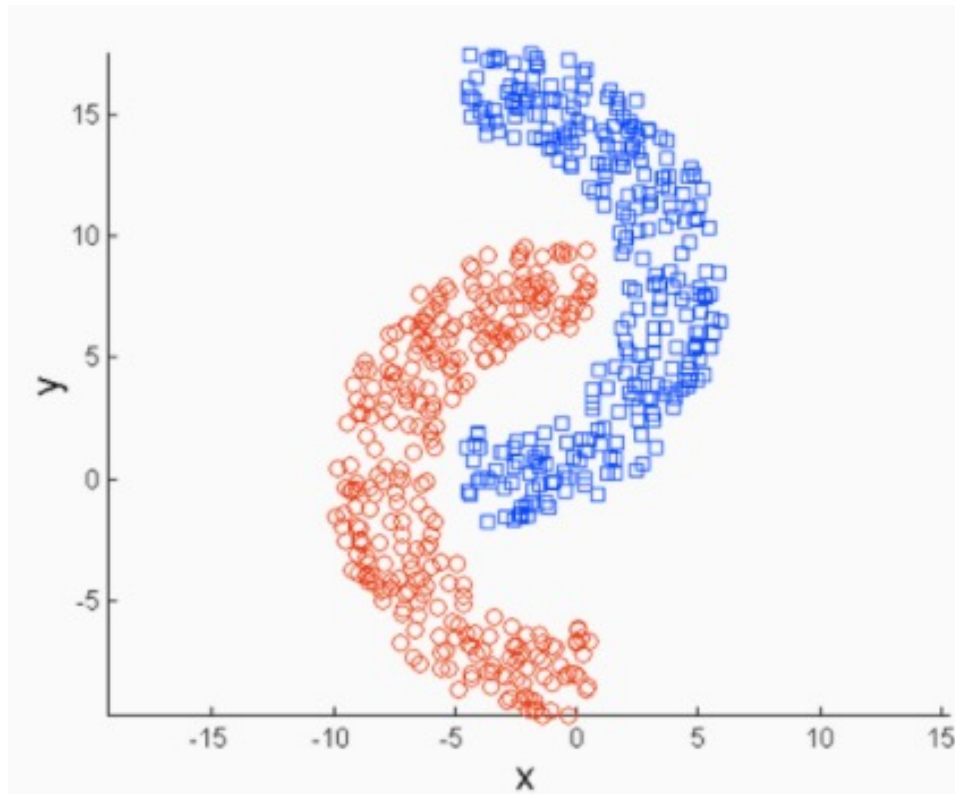


**Original Points**

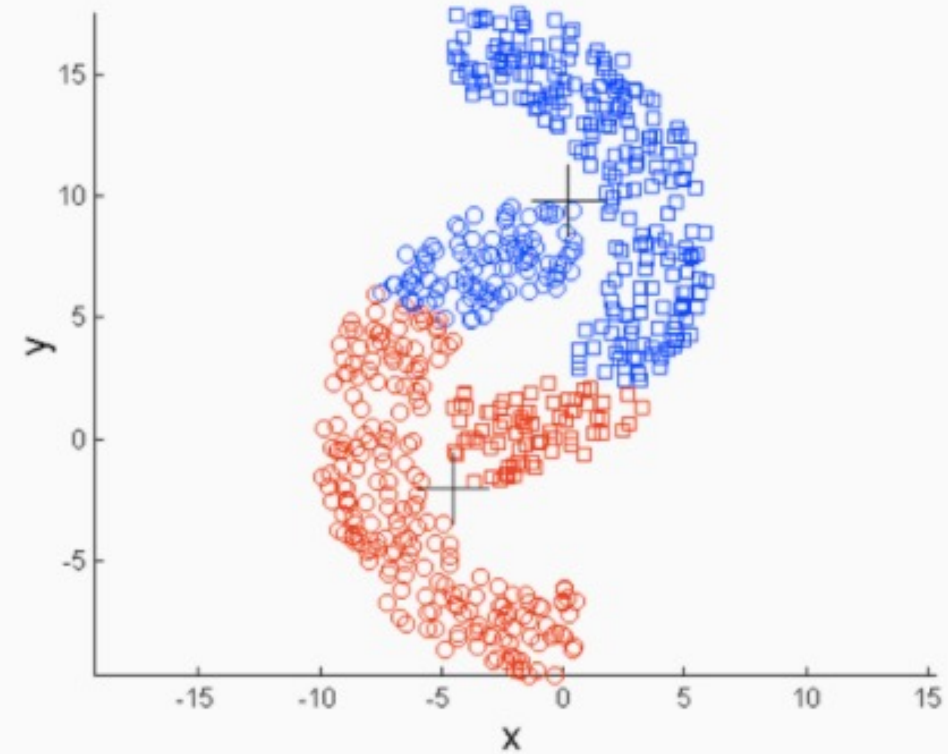


**K-means (3 Clusters)**

# Issues with k-means (3)

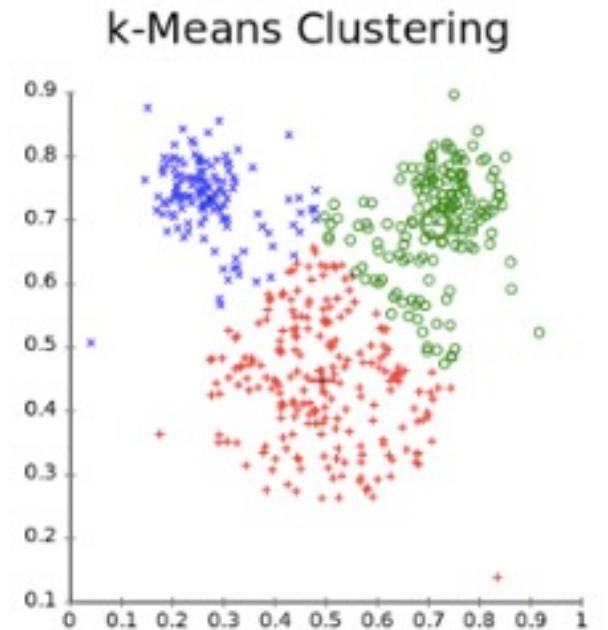
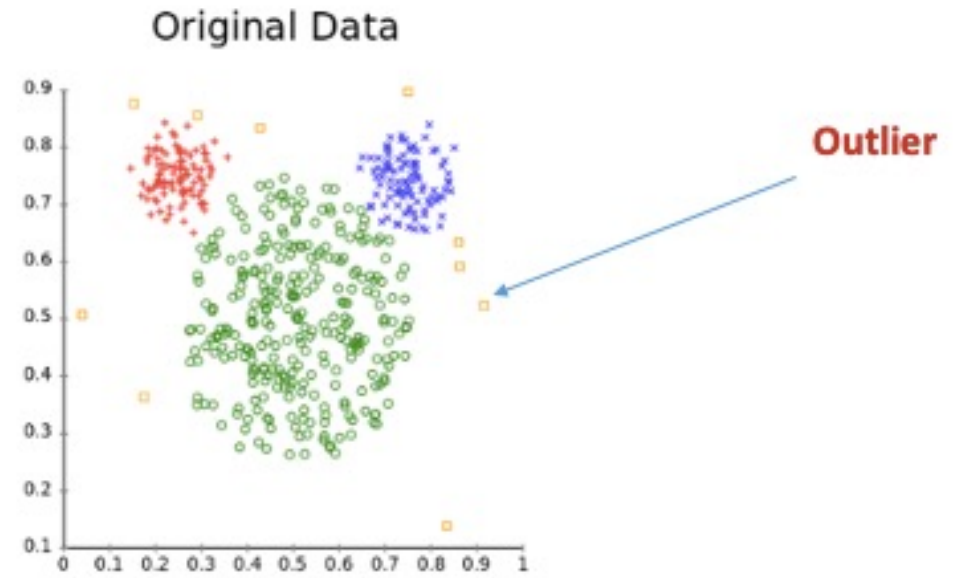


**Original Points**



**K-means (2 Clusters)**

# Issues with k-means (4)



# Issues with k-means (Summary)

1. Not knowing the optimum value of  $K$
2. Also, k-means clustering does not work well when
  - We want to discover clusters of varying sizes, densities and shapes
  - We do not want to include noisy points (outliers) into any clusters

# Are there any Alternatives?

- Many
- But the two that you will be introduced later (level 6) include
  - Hierarchical Clustering
  - Density Based Clustering (DBSCAN)

# Reflection

1. What are the key differences between supervised and unsupervised machine learning?
2. How might a business use clustering to make more informed decisions or improve their services?
3. What are some of the metrics used to measure similarity between data points when clustering?
4. How does the K-means clustering algorithm work? Could you explain its main steps and how it minimises within-cluster variation?
5. Why is it important to consider the initial positioning of centroids in K-means clustering, and how might suboptimal initial positions impact the results?
6. How does K-means++ select initial centroids, and why might this method lead to more optimal clustering solutions than the original K-means approach?