Introduction to Data Science

(Lecture 18)

Dr. Mohammad Pourhomayoun

Assistant Professor

Computer Science Department

California State University, Los Angeles

Unsupervised Learning

Two Common Learning Settings

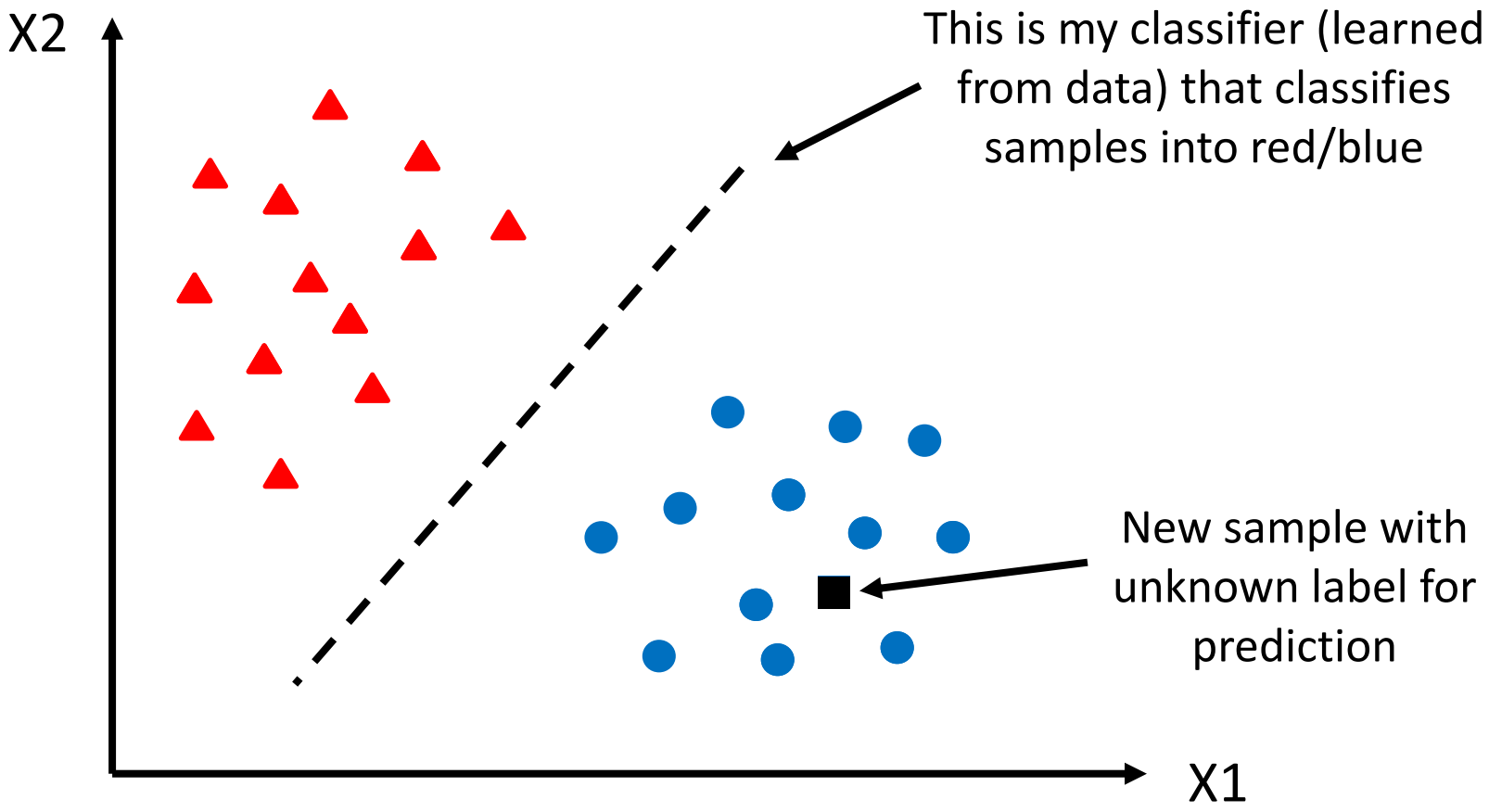
- **Supervised learning:** Learning from **labeled** observations.
  - In training stage, the algorithm is presented with **features and their known labels**, and the goal is to train a model that maps future inputs to new labels.
  - E.g. K-Nearest Neighbors classifier (KNN), Decision Tree classifier, Linear Regression, Logistic Regression, Polynomial Regression, Random Forest, …

- **Unsupervised learning:** Learning from **unlabeled** observations.
  - The algorithm is presented **Only** with **features!**
  - The goal is to **Discover hidden patterns and latent structure** from features alone.
  - It is like a Data Exploration to find hidden patterns.

Why/When to use Unsupervised Learning?

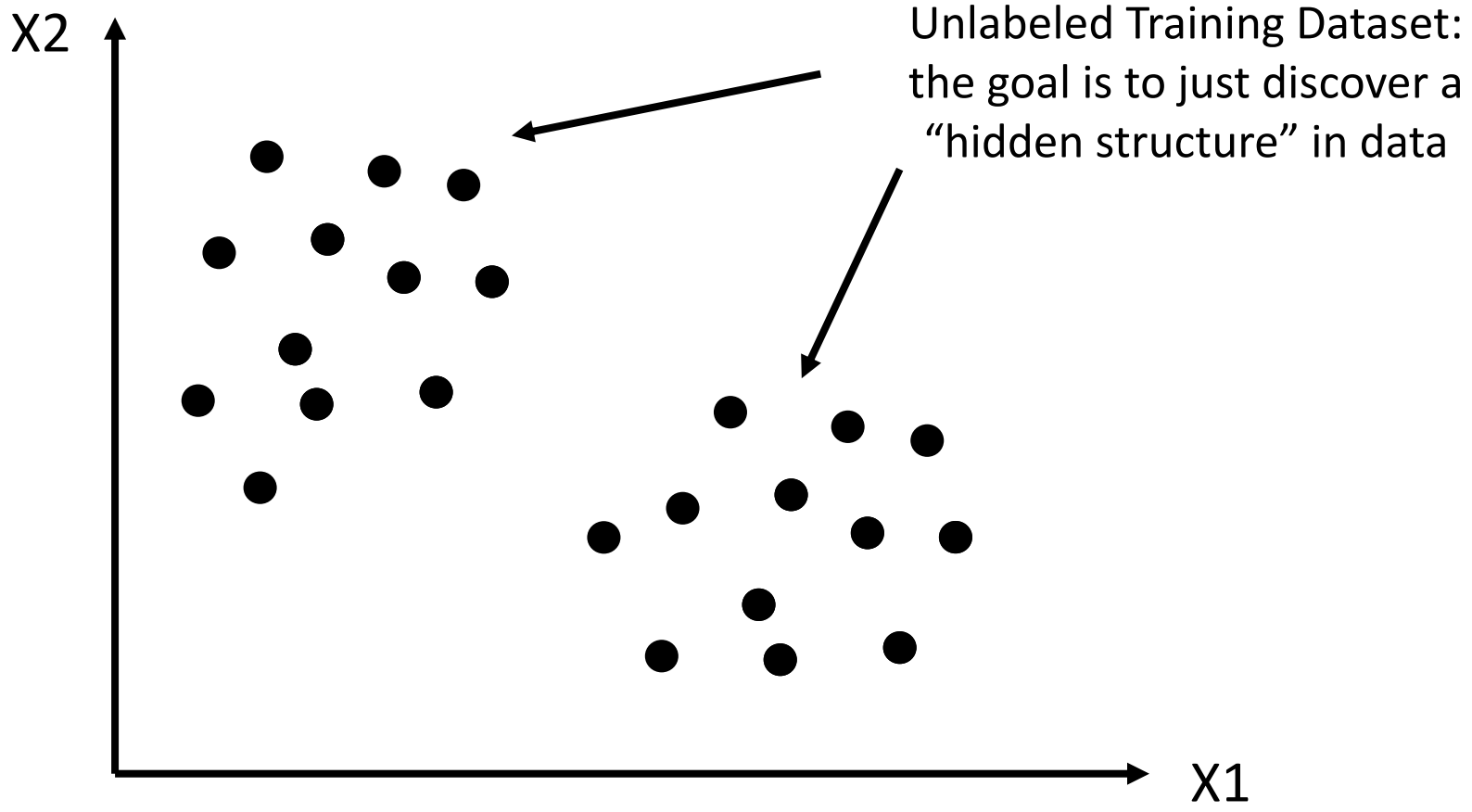
1. The Label is Unknown.
2. The data is Unlabeled because we can not afford labeling the data.
3. Sometimes, we don't care about the label, we just want to categorize the data.
4. Sometimes, applying an unsupervised algorithm prior to a supervised learning can improve the prediction results!
5. Sometimes, We want to manipulate the data w/o considering the label of that.

Review: Supervised Learning



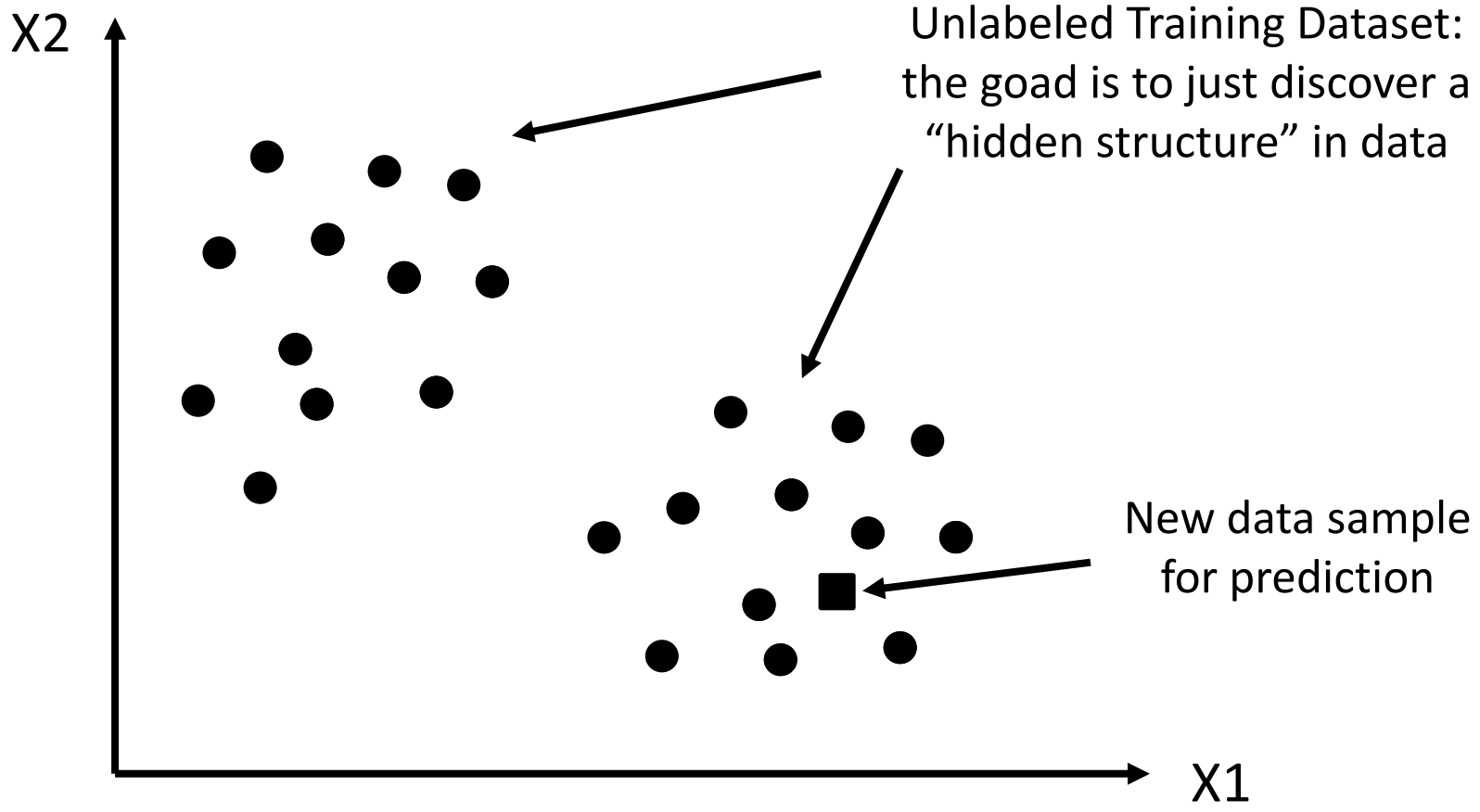
Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots, (x^{(m)}, y^{(m)})\}$

Unsupervised Learning



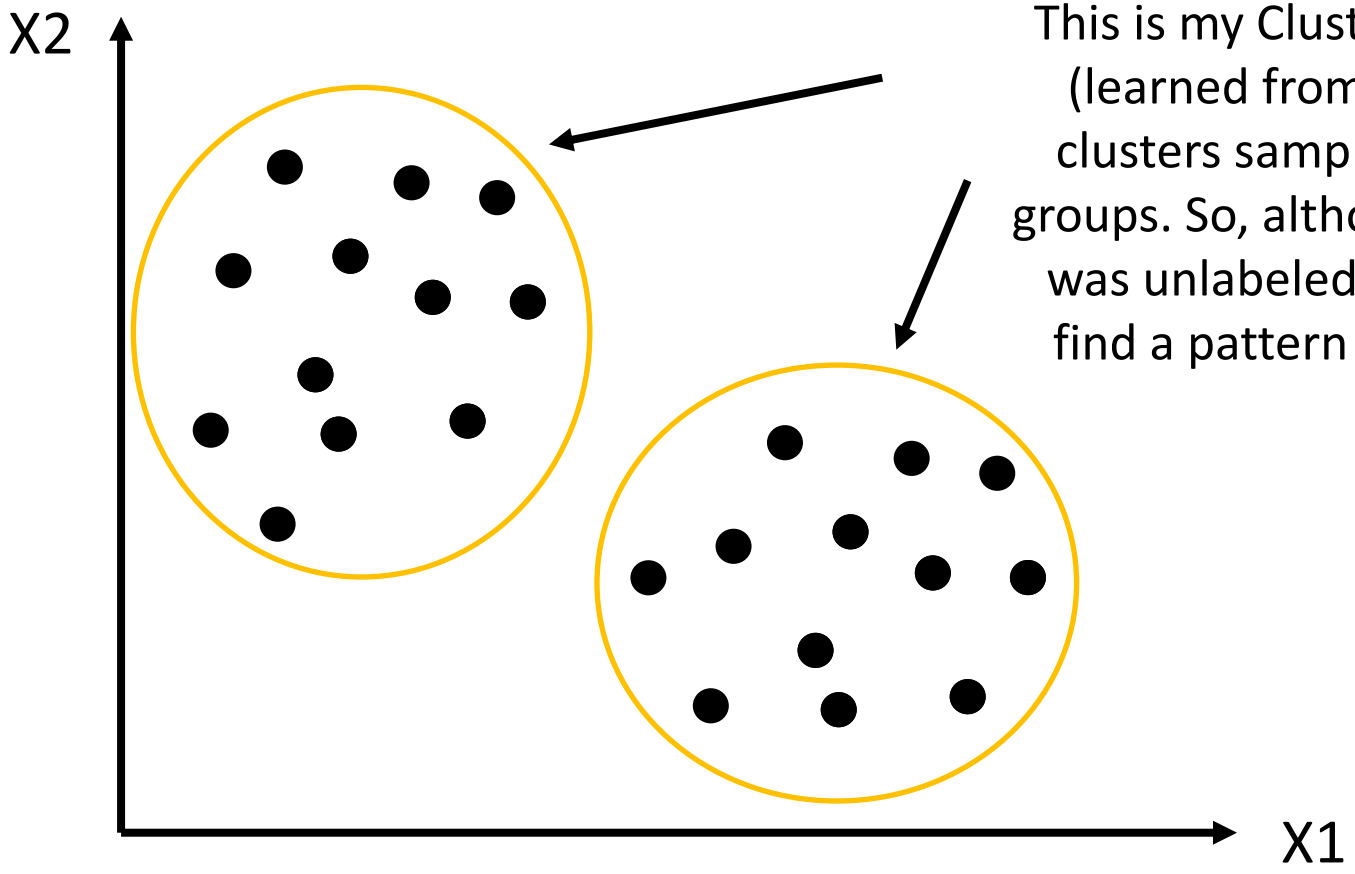
Training set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$

Unsupervised Learning



Training set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$

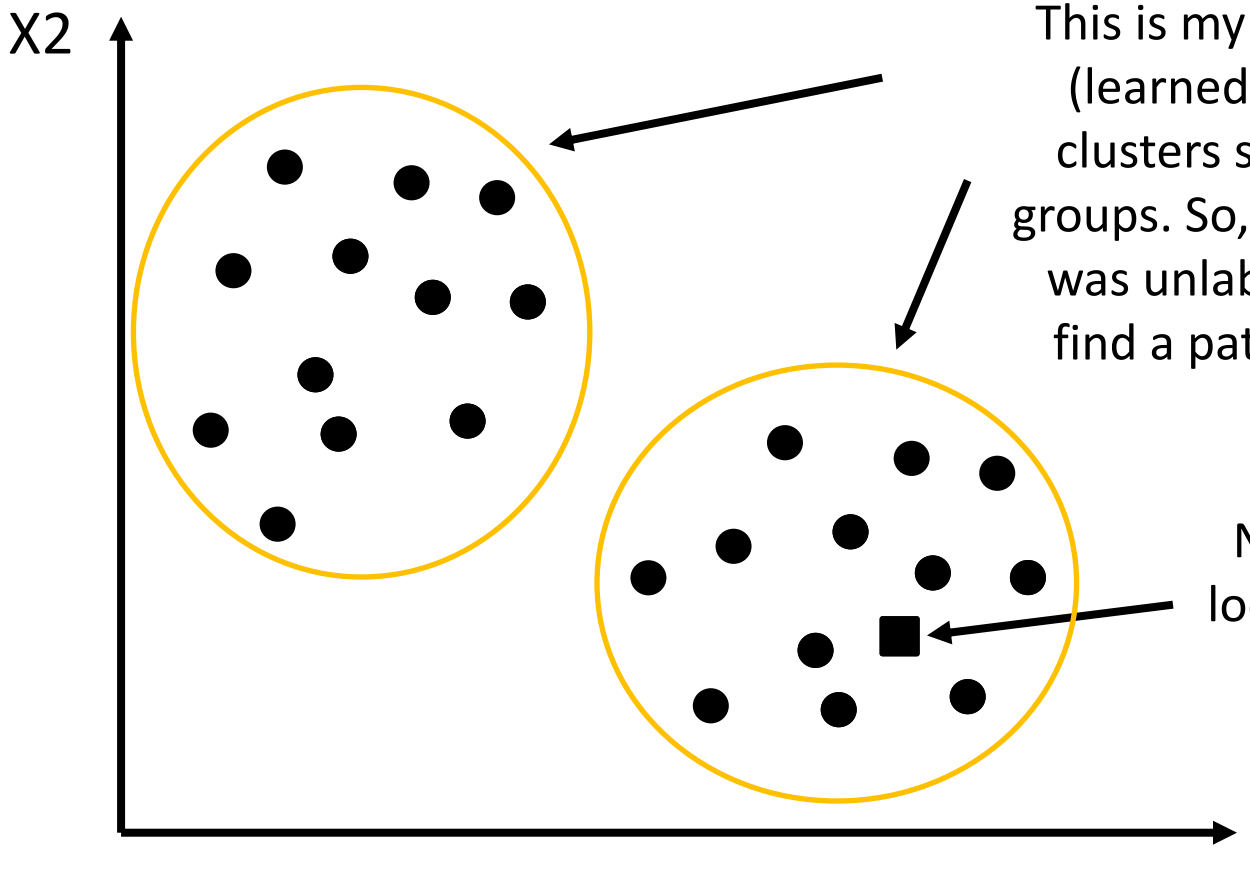
Unsupervised Learning



This is my Clustering Model (learned from data) that clusters samples into two groups. So, although the data was unlabeled, but I could find a pattern in the data.

Training set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$

Unsupervised Learning

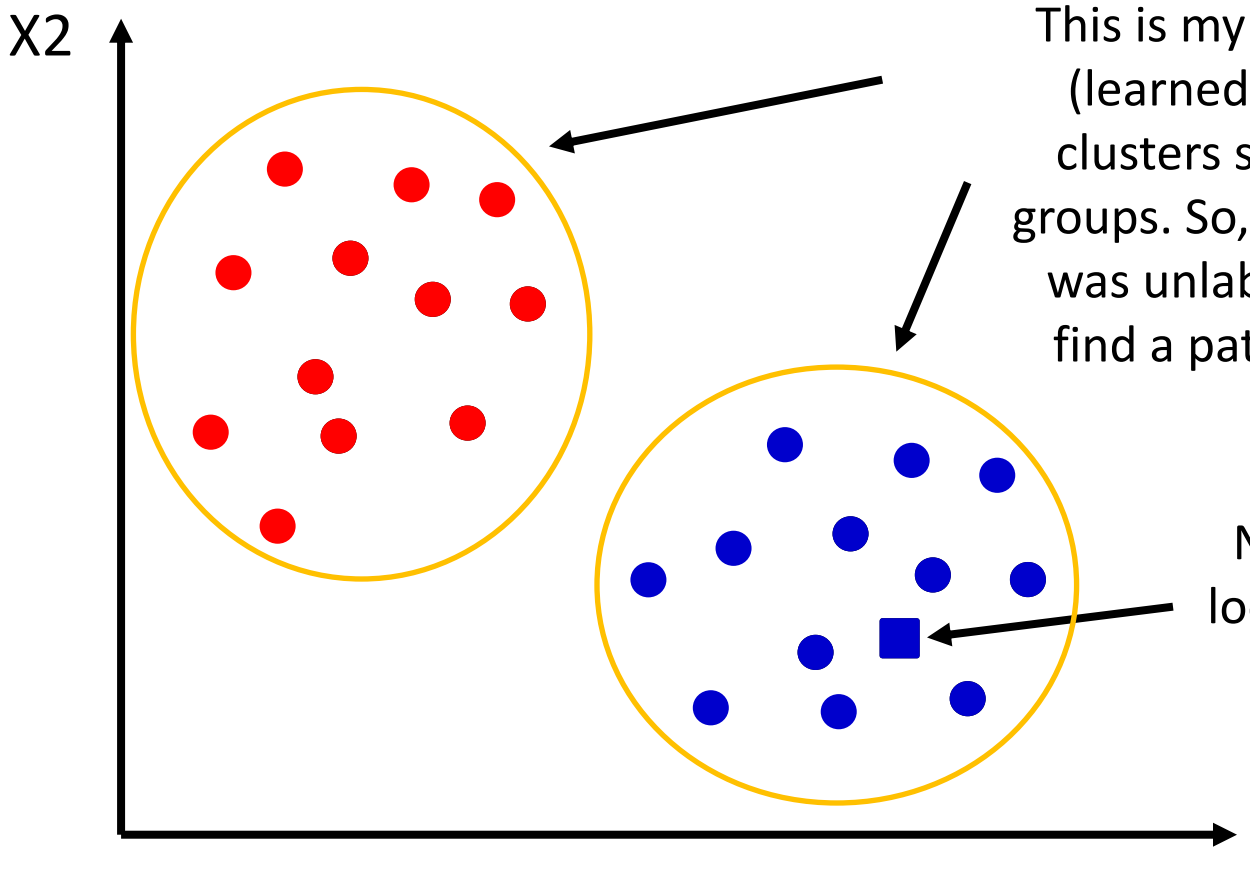


This is my Clustering model (learned from data) that clusters samples into two groups. So, although the data was unlabeled, but I could find a pattern in the data.

New sample to be located in this group

Training set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$

Unsupervised Learning



This is my Clustering model (learned from data) that clusters samples into two groups. So, although the data was unlabeled, but I could find a pattern in the data.

New sample to be located in this group

Training set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$

K-Means Clustering Algorithm

K-Means Clustering Algorithm

- **k-means** is a simple and very popular clustering algorithm that tries to **partition n data samples into k clusters** so that each sample belongs to the cluster with the **nearest mean** (centroid).

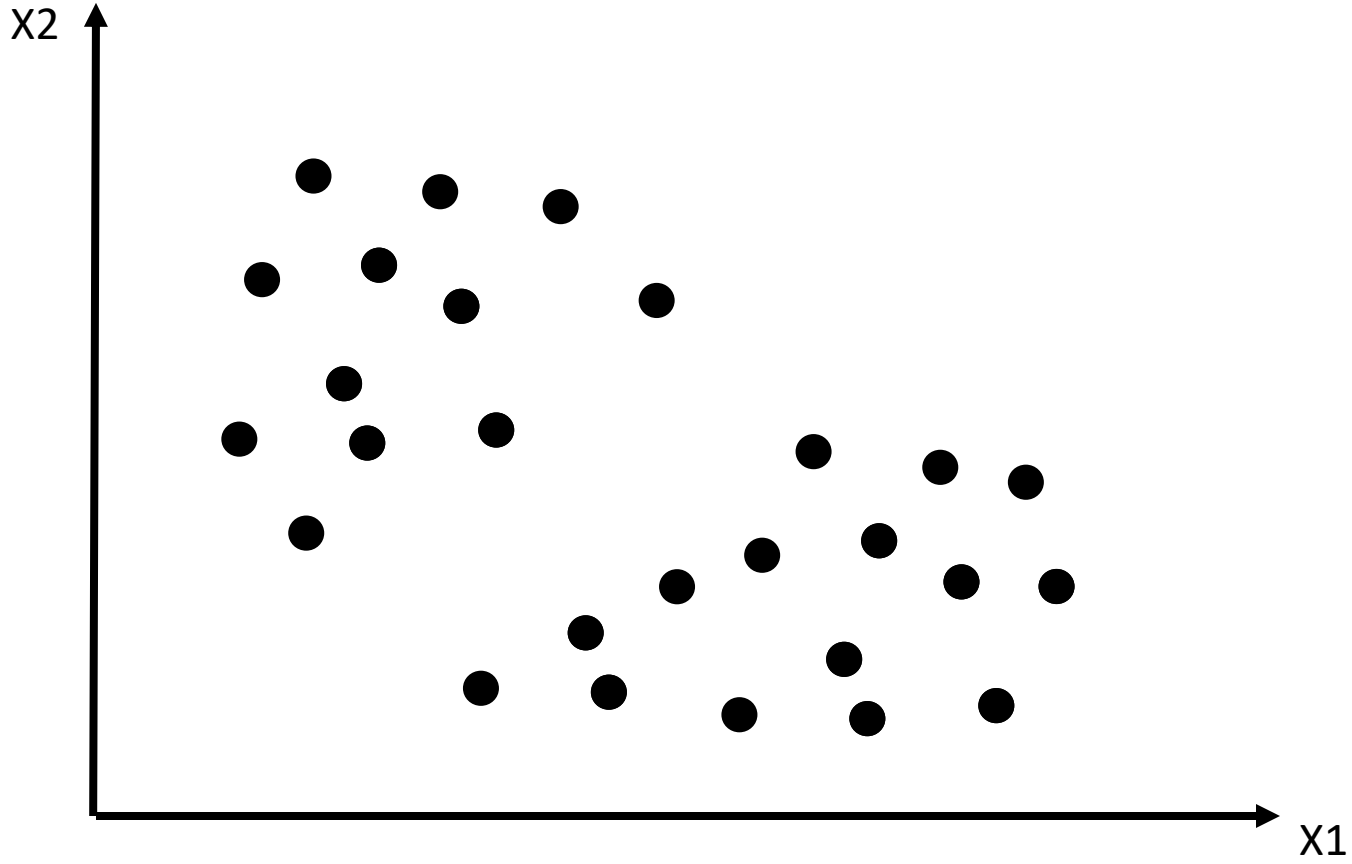
- It is ***unsupervised learning***: we don't have labeled samples demonstrating how the data should be grouped!

- "k" (the number of desired clusters) should be predefined.

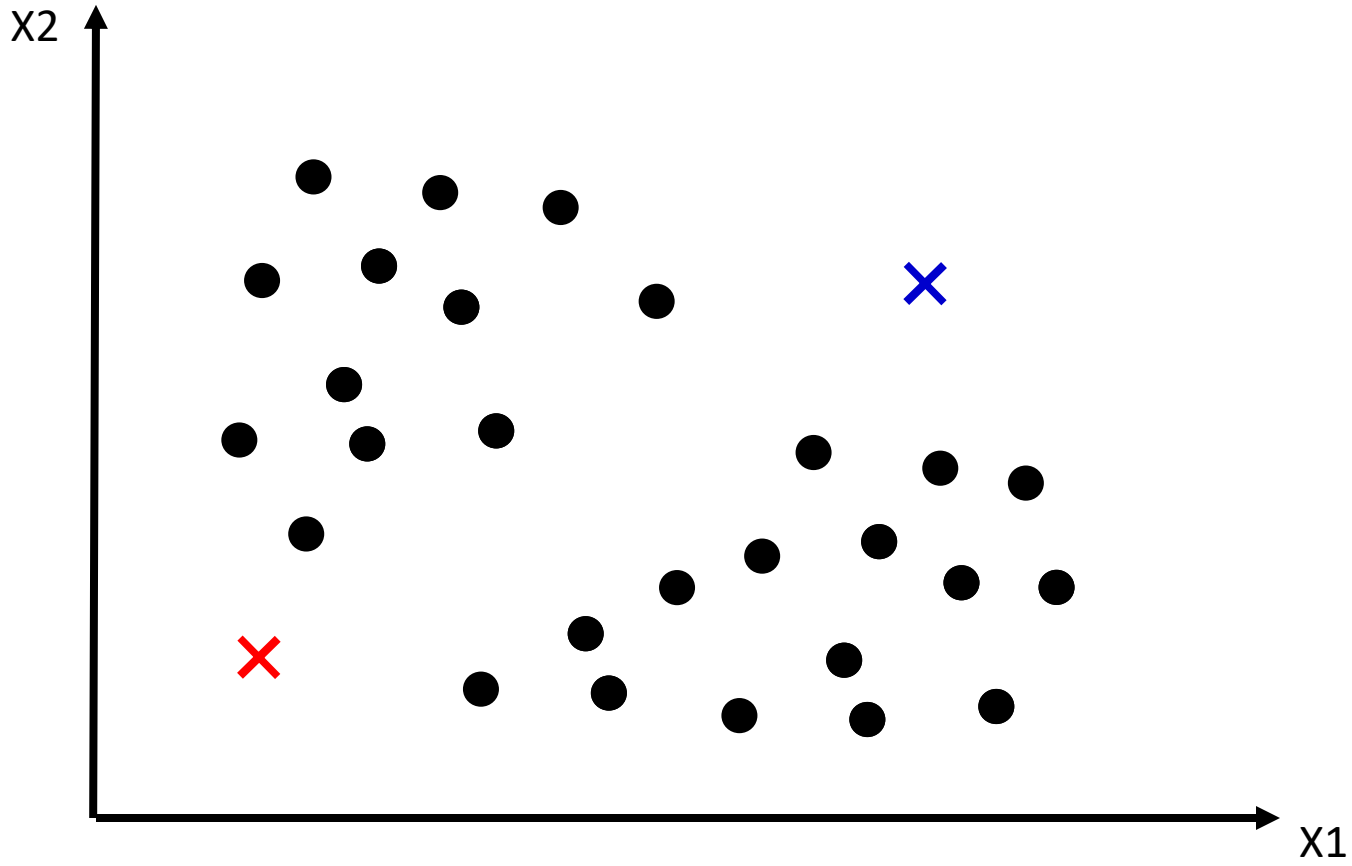
K-Means Clustering Algorithm

- **Step 0: Initialization:** set K random points in feature space as initial centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K \in \mathbb{R}^n$
- **Step 1: Cluster Assignment:** Assign each data sample to the cluster of the nearest centroid point (for each point, we need to calculate the distance from that point to all centroids and select the nearest one).
- **Step 2: Centroid Move:** Update centroid locations to the mean location of the members of the current cluster.
- **Step 3:** Go back to Step 1. Repeat the procedure until the samples and centroids get stable positions (i.e., the samples in each cluster no longer changes).

Example: K-Means with K=2

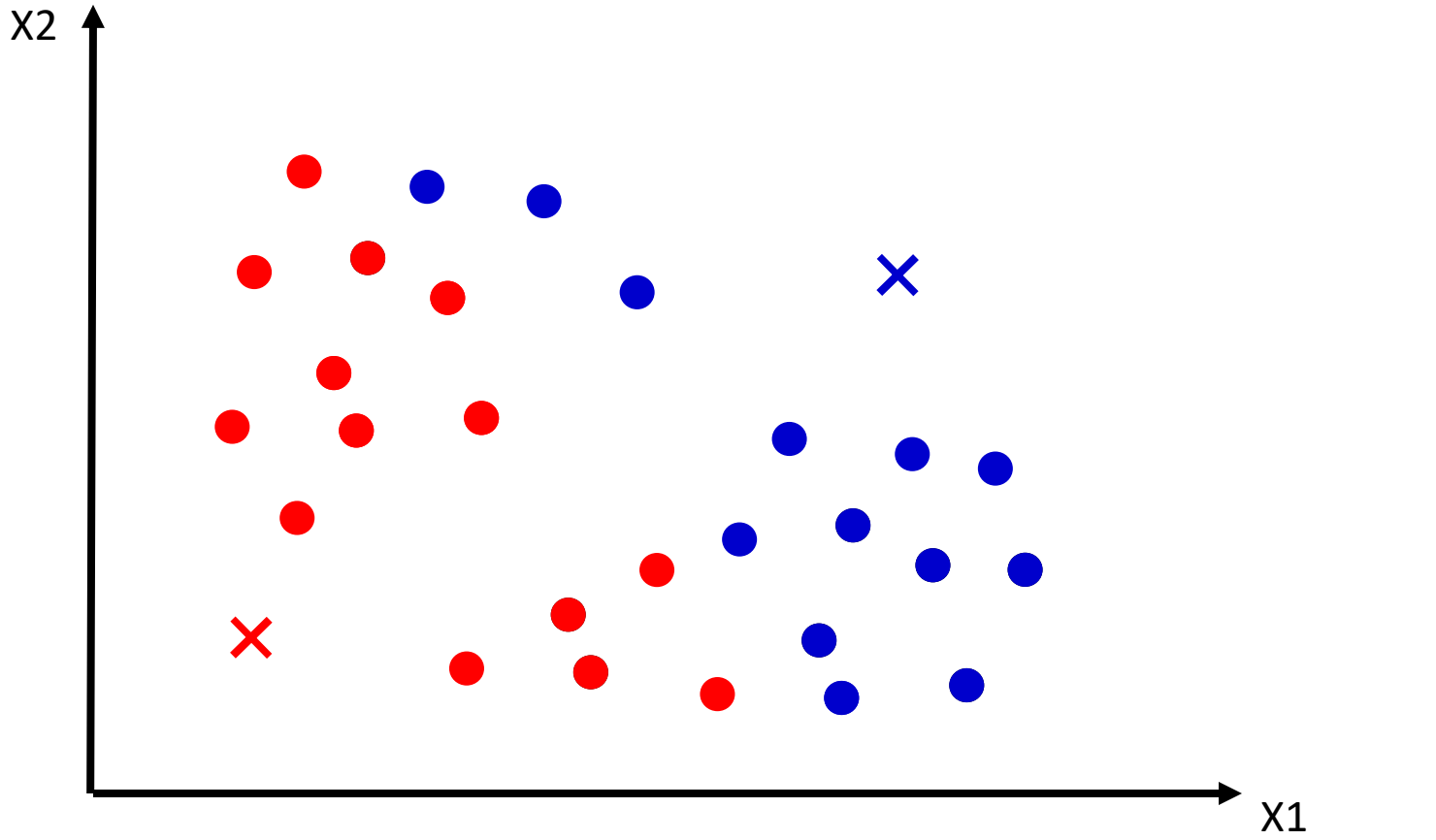


Step 0: Random Initialization



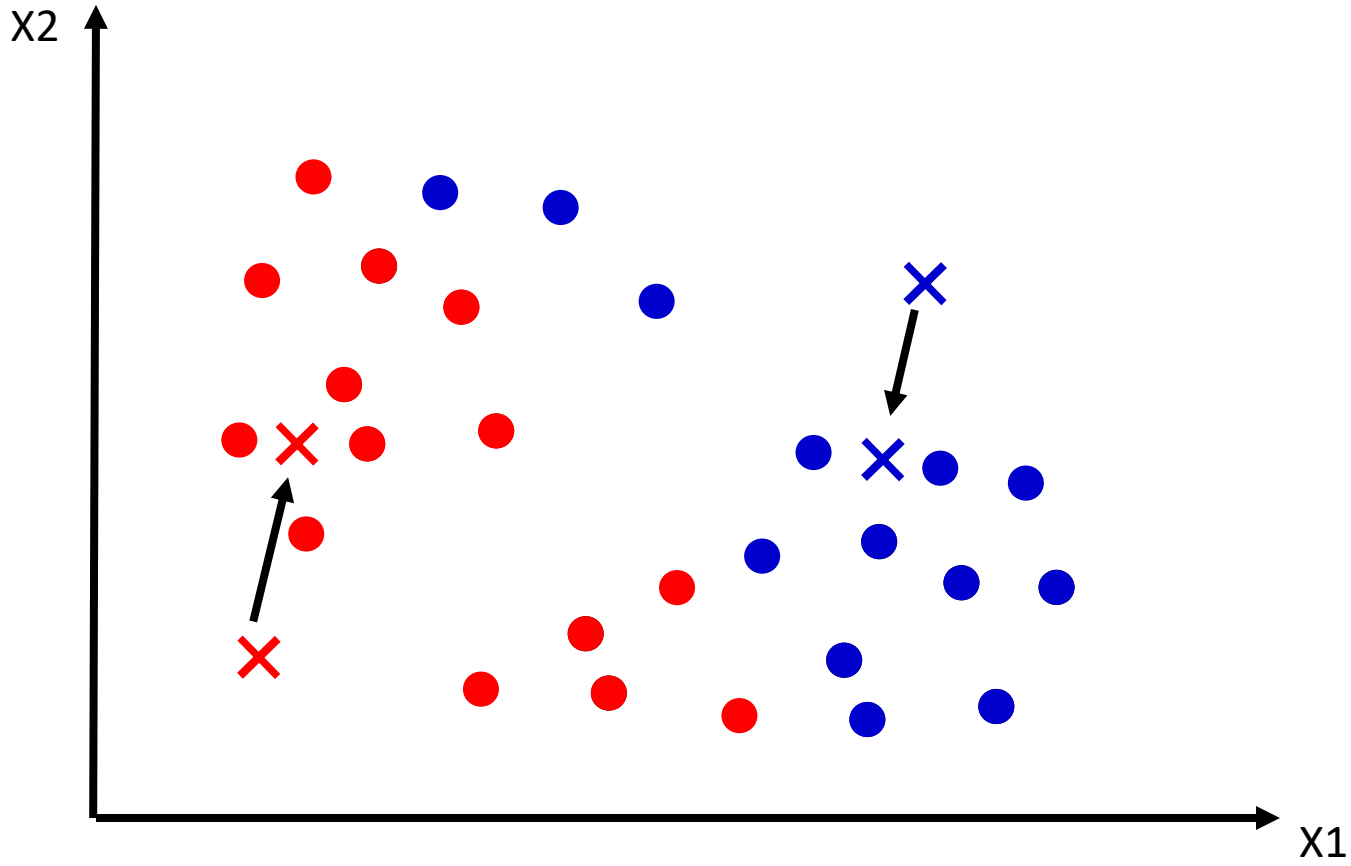
Initialization: set K random points in feature space as initial centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K \in \mathbb{R}^n$

Step 1: Assign the Points



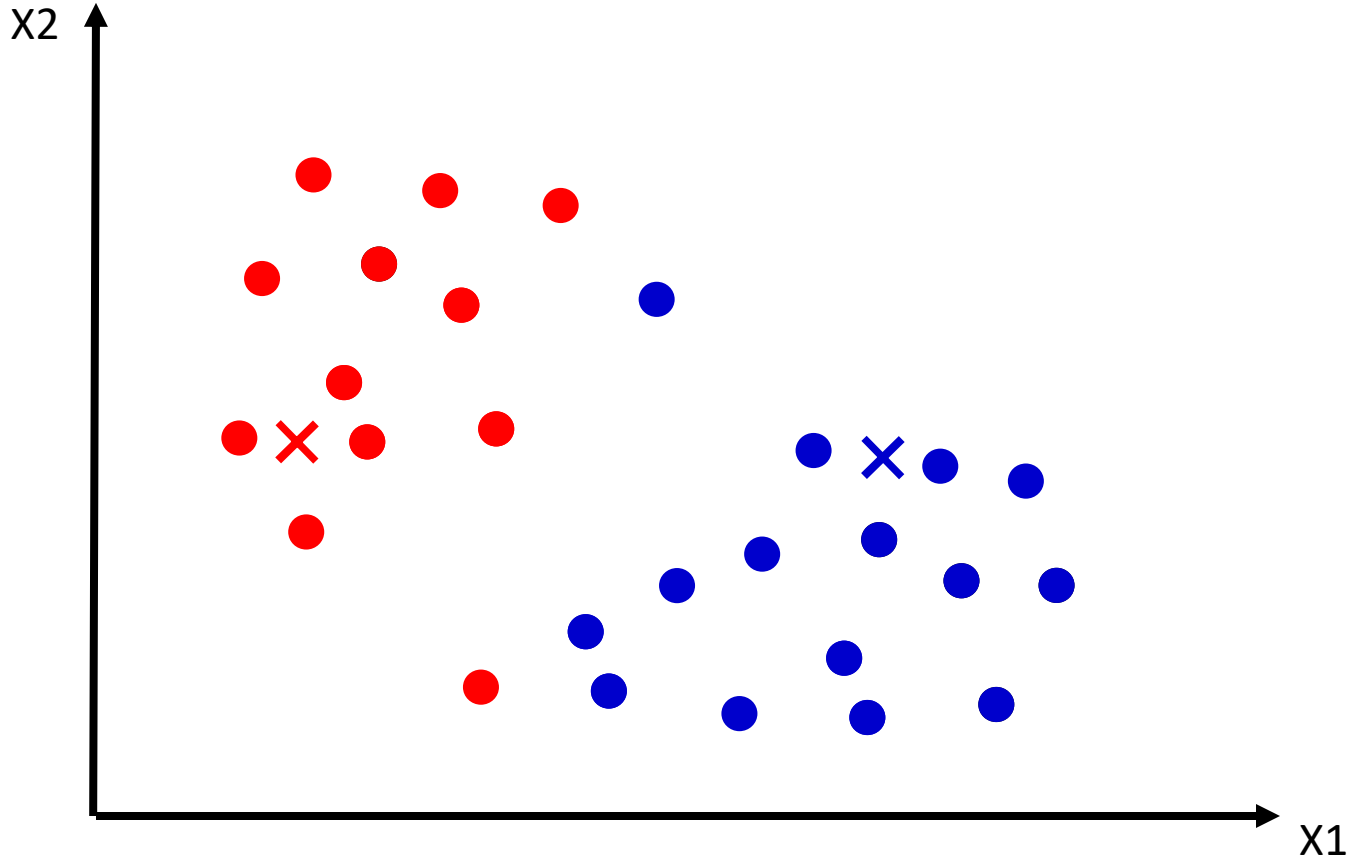
Cluster Assignment: Assign each data sample to the cluster of the nearest centroid point.

Step 2: Find new Centroid



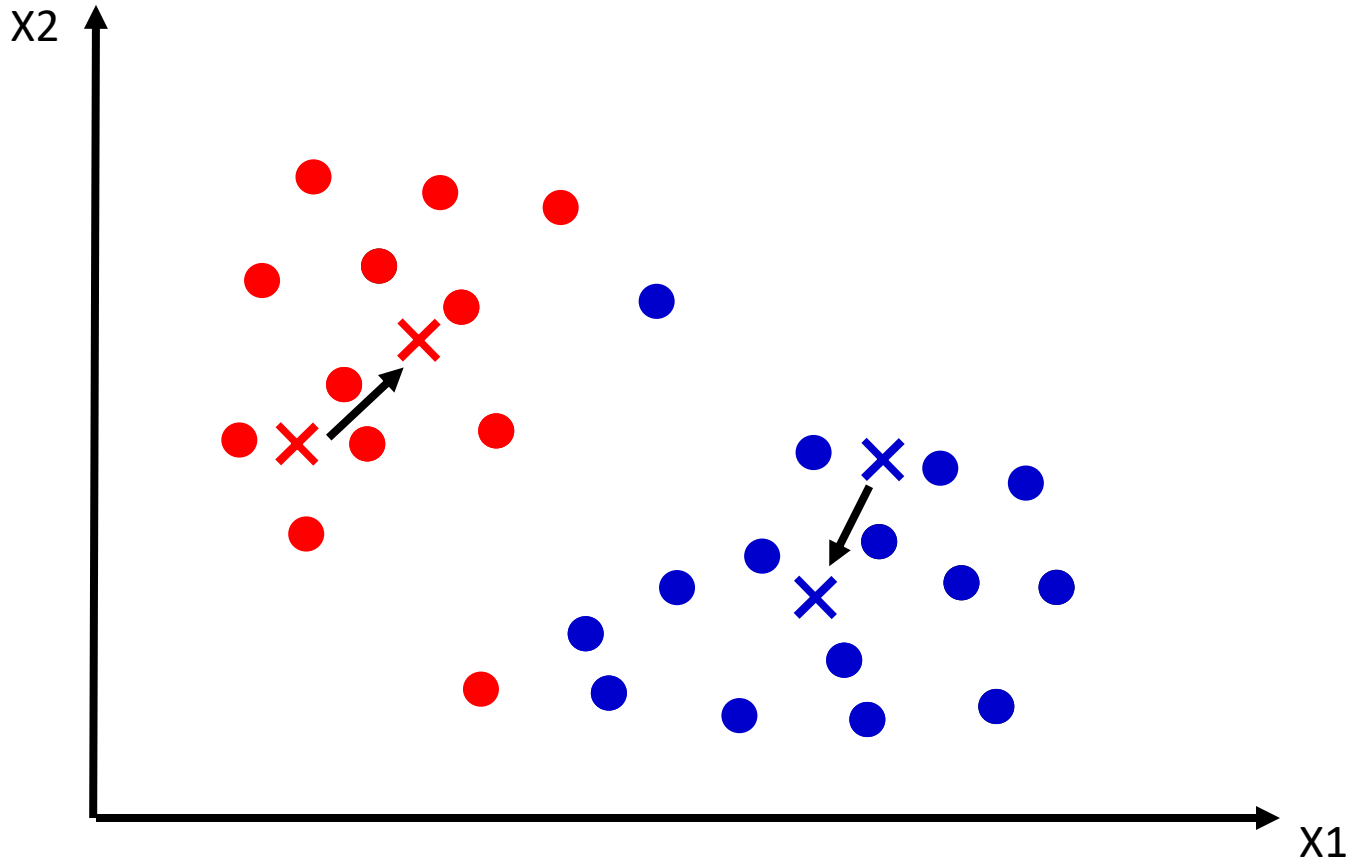
Centroid Move: Update centroid locations to the mean location of the members of the current cluster.

Back to Step 1: Assign the Points



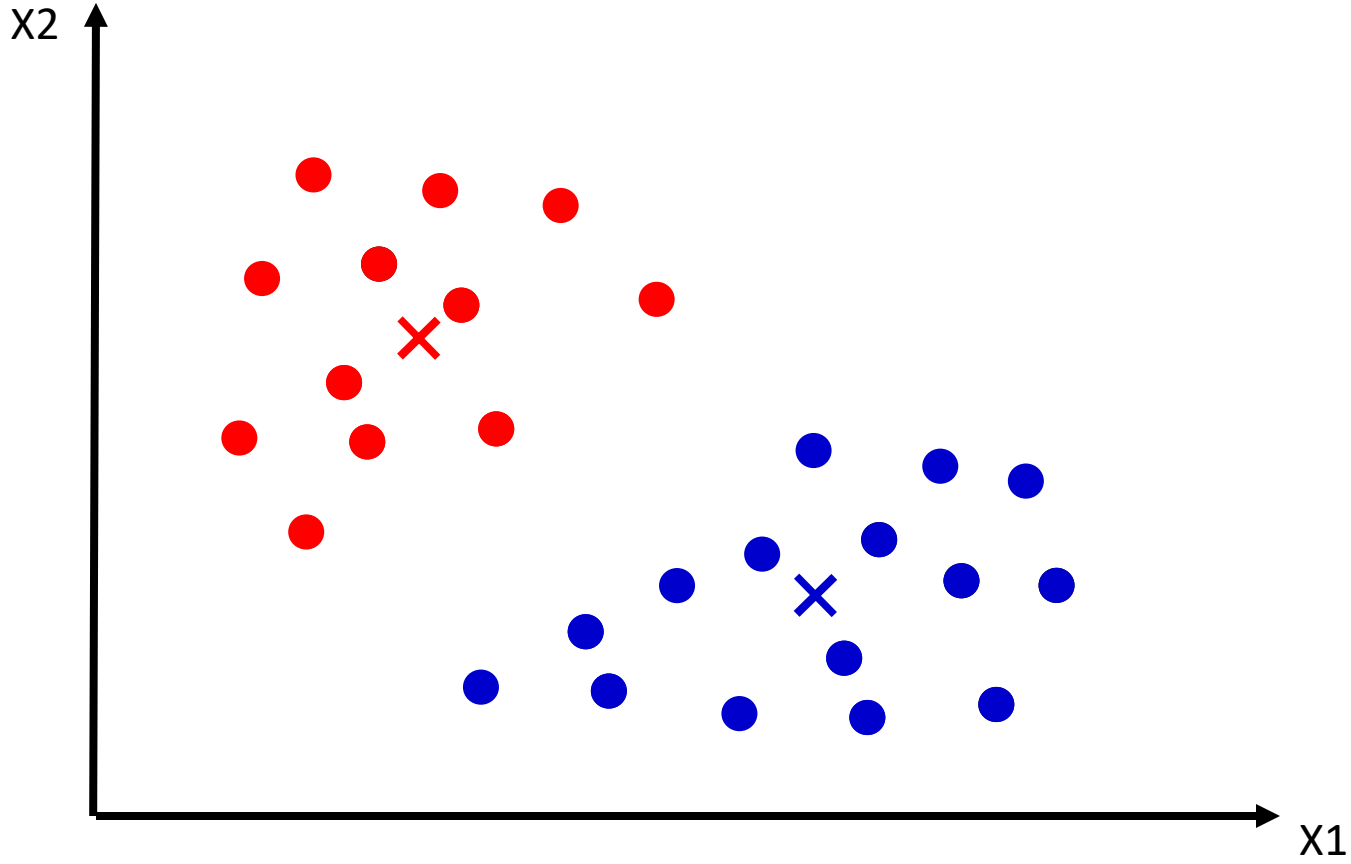
Cluster Assignment: Assign each data sample to the cluster of the nearest centroid point.

Step 2: Find new Centroid



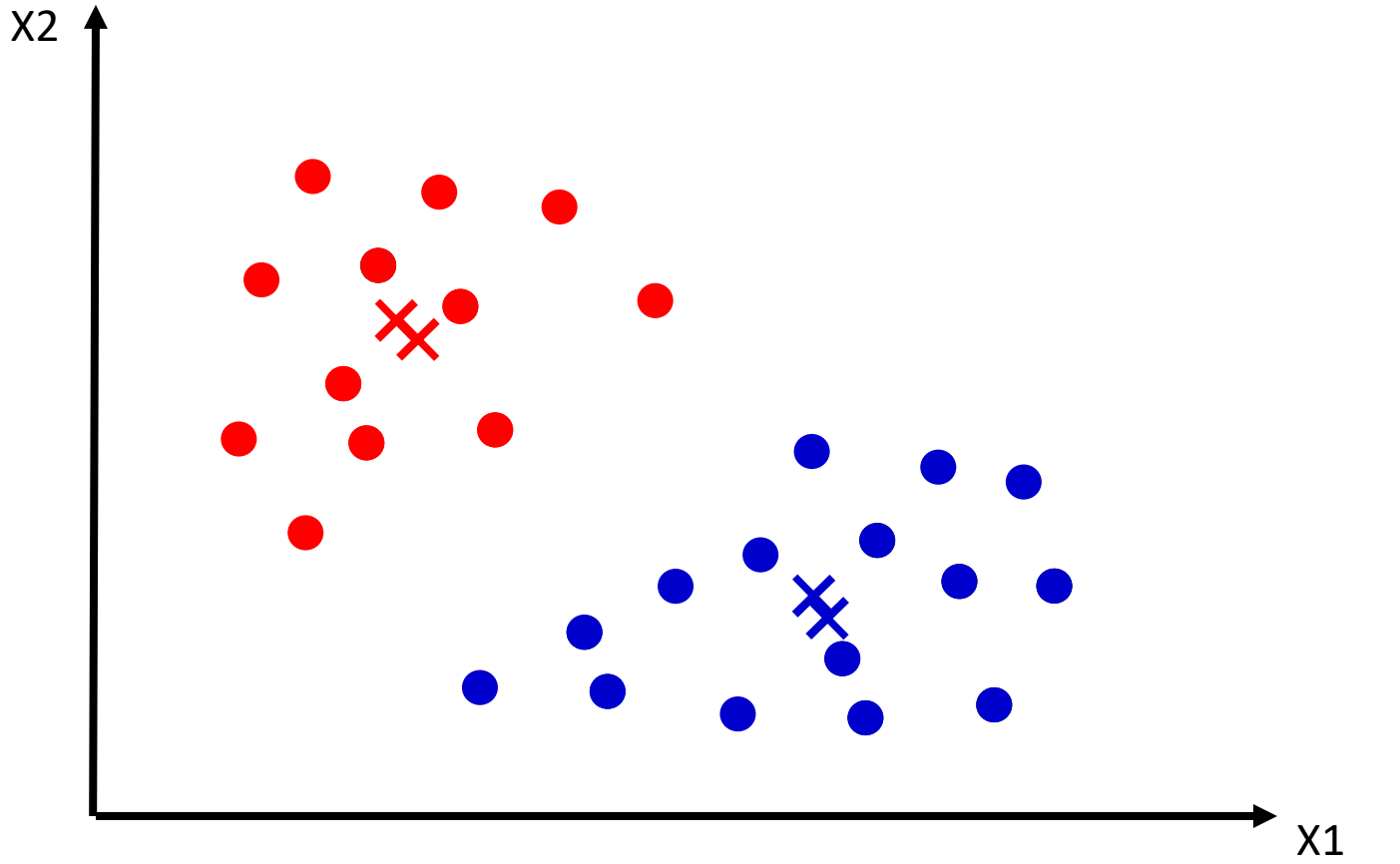
Centroid Move: Update centroid locations to the mean location of the members of the current cluster.

Back to Step 1: Assign the Points



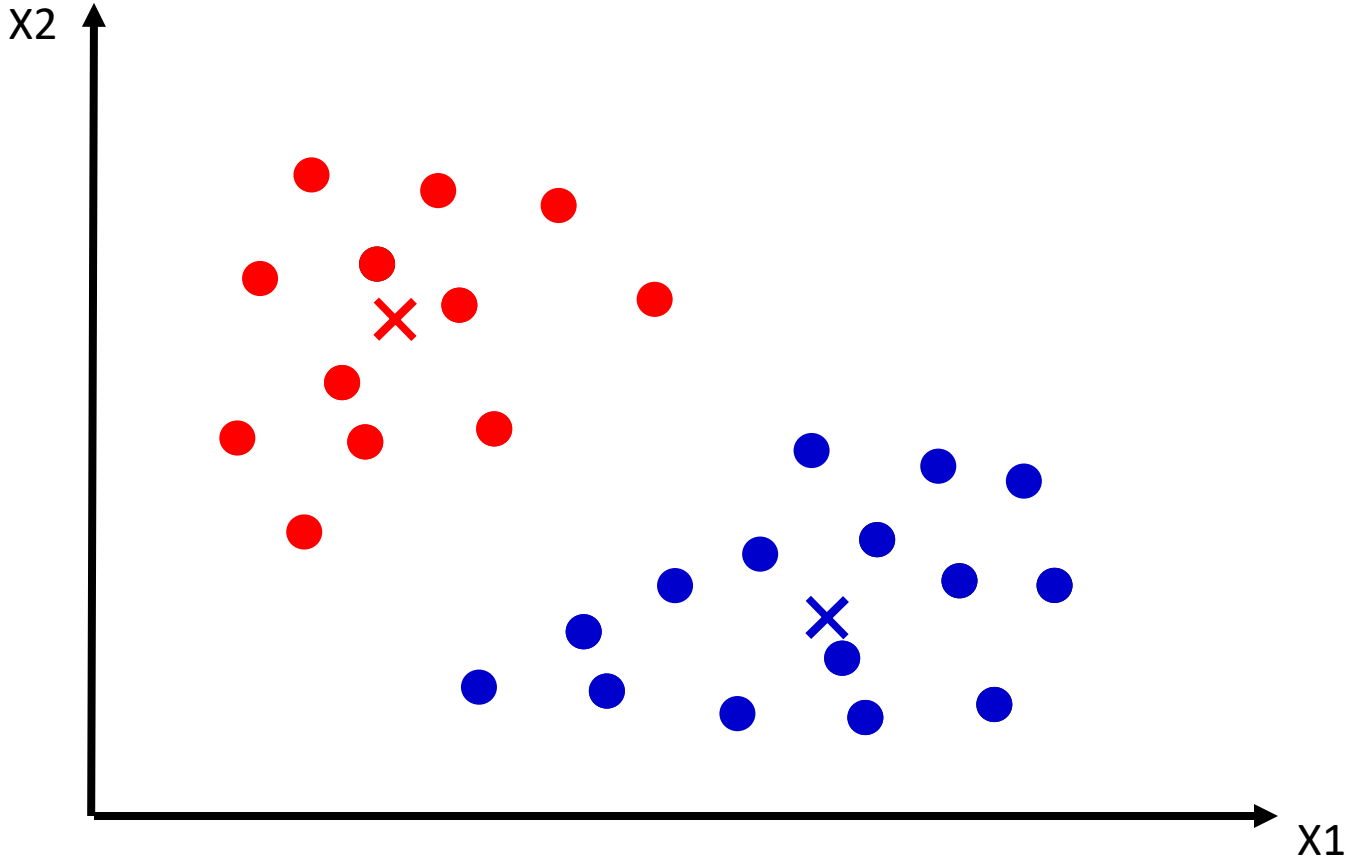
Cluster Assignment: Assign each data sample to the cluster of the nearest centroid point.

Step 2: Find new Centroid



Centroid Move: Update centroid locations to the mean location of the members of the current cluster.

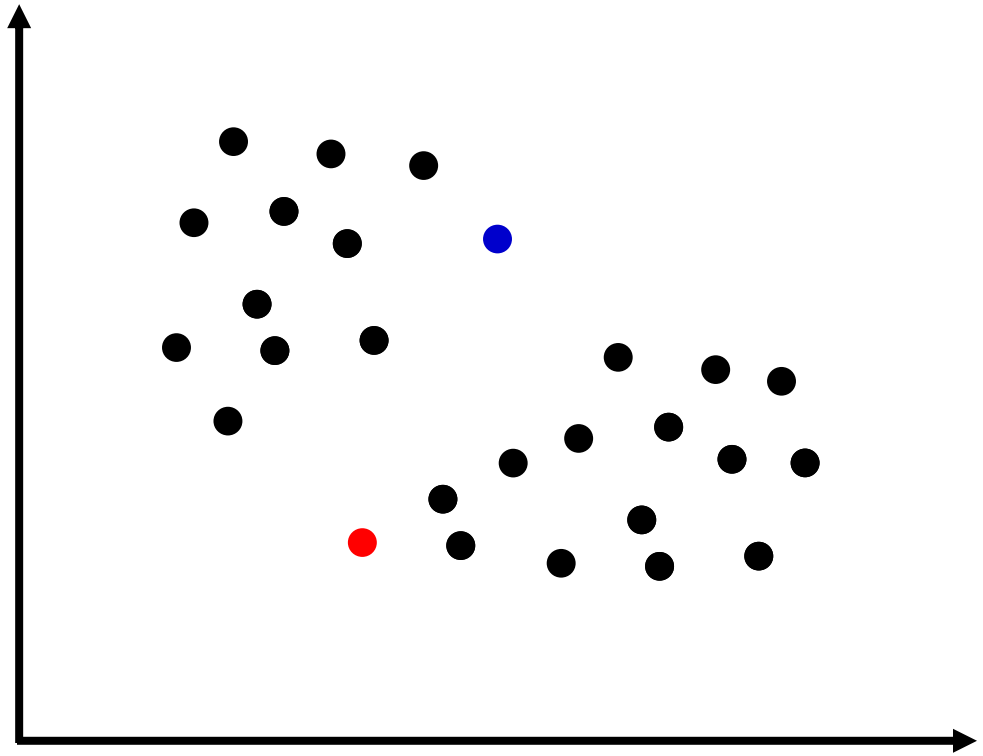
Step 3: Done!



Stop when the samples and centroids gets stable enough position (i.e., the samples in each cluster no longer changes).

Random Initialization

- To make sure that the centroid points are around, we usually randomly pick "K" of our data samples as the **initial** centroid points.



Pseudo Code for K-Means

Notations:

i = the index of the data sample (1,2,…, m) .

$c^{(i)}$ = index of the cluster (1,2,…,K) to which example $\boldsymbol{x}^{(i)}$ is currently assigned.

$\boldsymbol{\mu}_k$ = cluster centroid k ($\boldsymbol{\mu}_k \in \mathbb{R}^n$).

E.g.: if $\boldsymbol{x}^{(1)}$ is in cluster 5, and $\boldsymbol{x}^{(2)}$ is in cluster 7, then

$c^{(1)} = 5$ and $c^{(2)} = 7$

Pseudo Code for K-Means

Randomly initialize K cluster centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$.

Repeat {

Data samples re-assign

for i = 1 to m :

$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $\boldsymbol{x}^{(i)}$: k for

$$\min_k \left|\left| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_k \right|\right|^2$$

Centroid re-assign

for k = 1 to K:

$\boldsymbol{\mu}_K$:= average (mean) of points assigned to cluster k

}

Random Initialization

- Notice that the final clustering results may depend on the choice of initial points!

Random Initialization

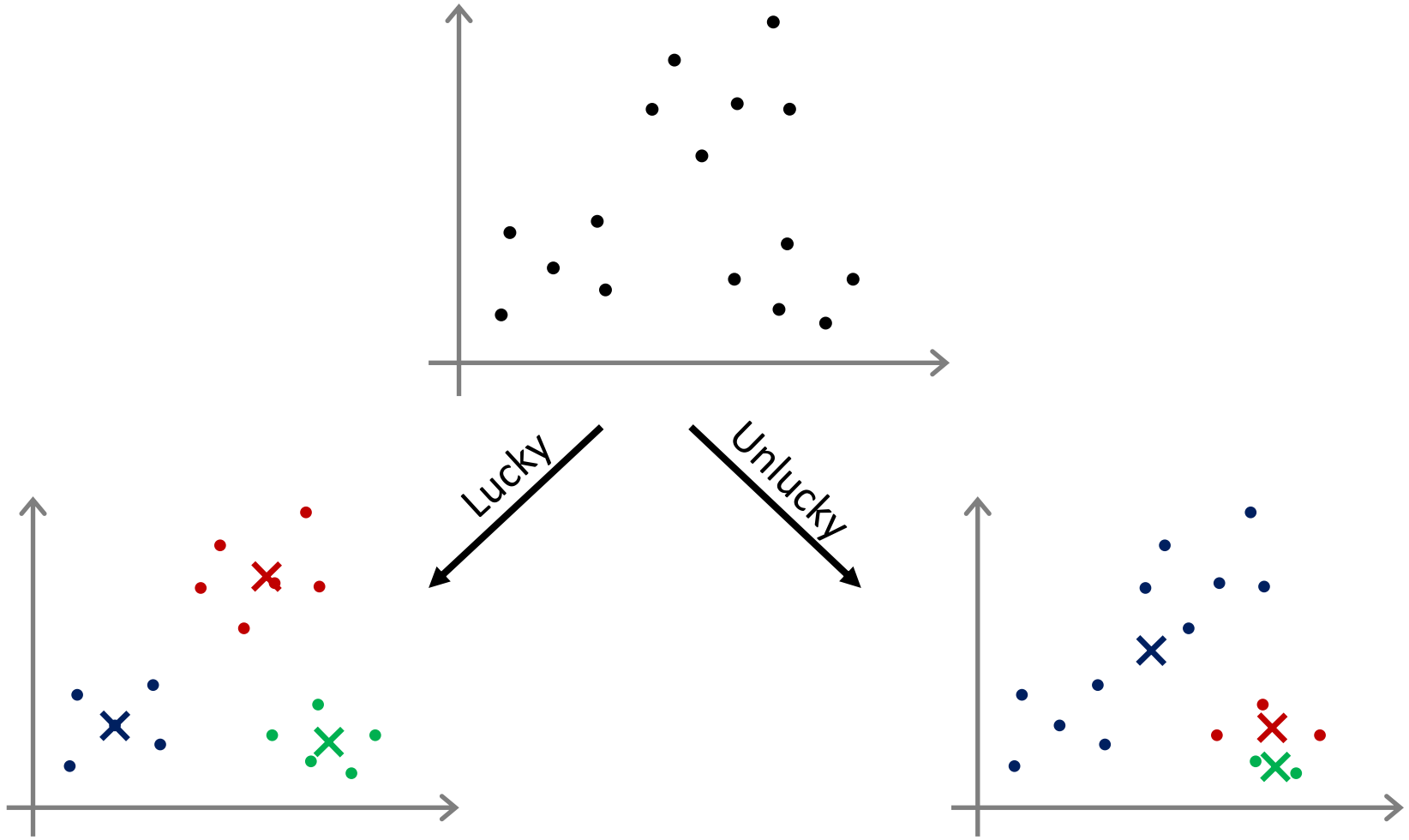


Figure Ref: Andrew Ng

Random Initialization

- Notice that the final clustering results may depend on the choice of initial points!

- Thus, The best approach is to **repeat random initialization multiple times** (rather than trusting on one single initialization), perform clustering several times, and finally select the the best clustering results.

Random Initialization

Notation:

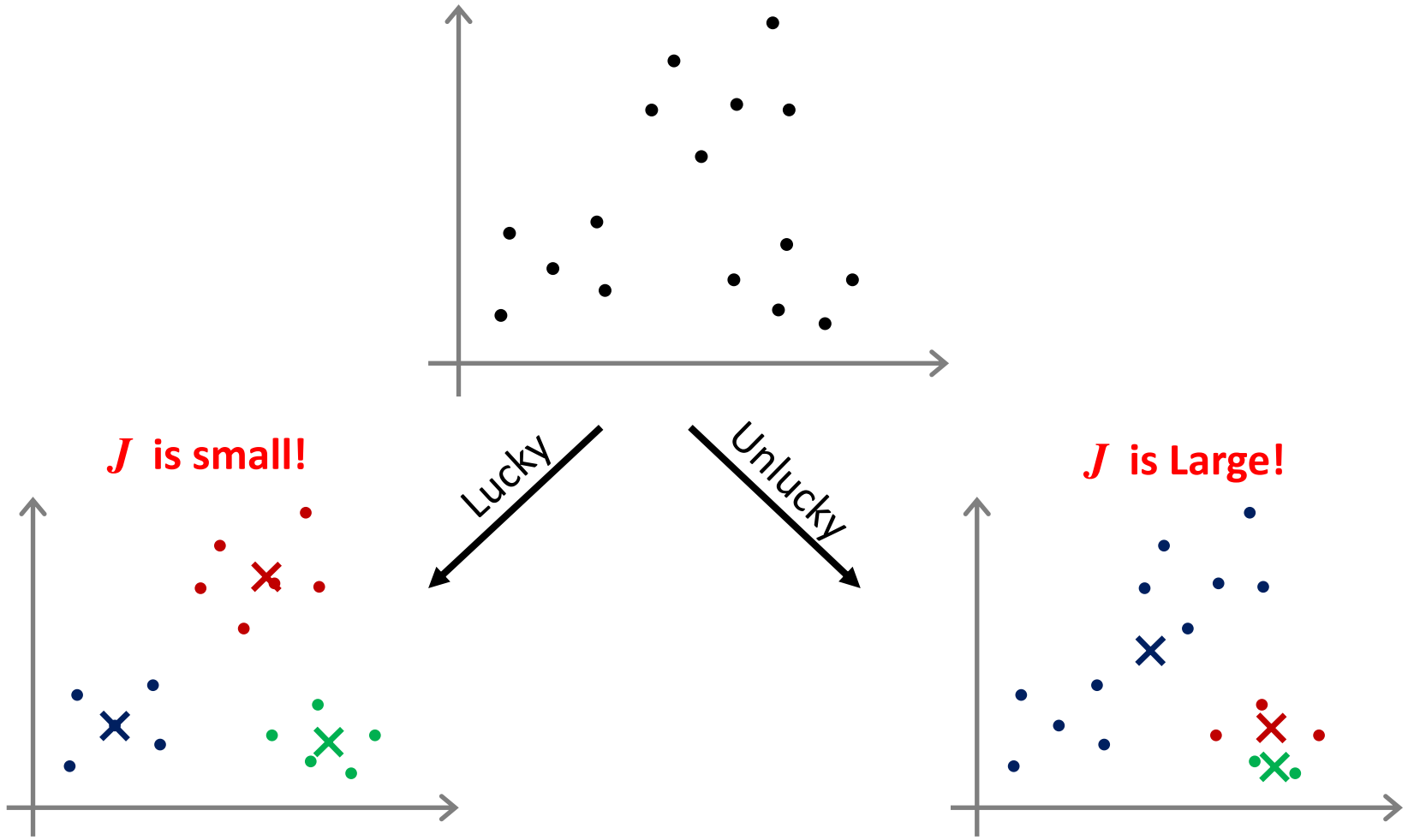
$c^{(i)}$ = index of cluster (1,2,…, K) to which example $\boldsymbol{x}^{(i)}$ is currently assigned.

$\mu_{c^{(i)}}$ = cluster centroid of cluster to which example $\boldsymbol{x}^{(i)}$ has been assigned.

J = "**clustering cost function**" defined as the **average distance of each sample to its cluster centroid**:

$$J = \frac{1}{m}\sum_{i=1}^{m} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_{c^{(i)}} \right\|^2$$

Random Initialization



Random Initialization

Multiple Random Initialization:

For i = 1 to 50 {

Randomly initialize K-means.

Run K-means. Get $c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K$.

Compute cost function as following:

$$J = \frac{1}{m} \sum_{i=1}^{m} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_{c^{(i)}} \right\|^2$$

}

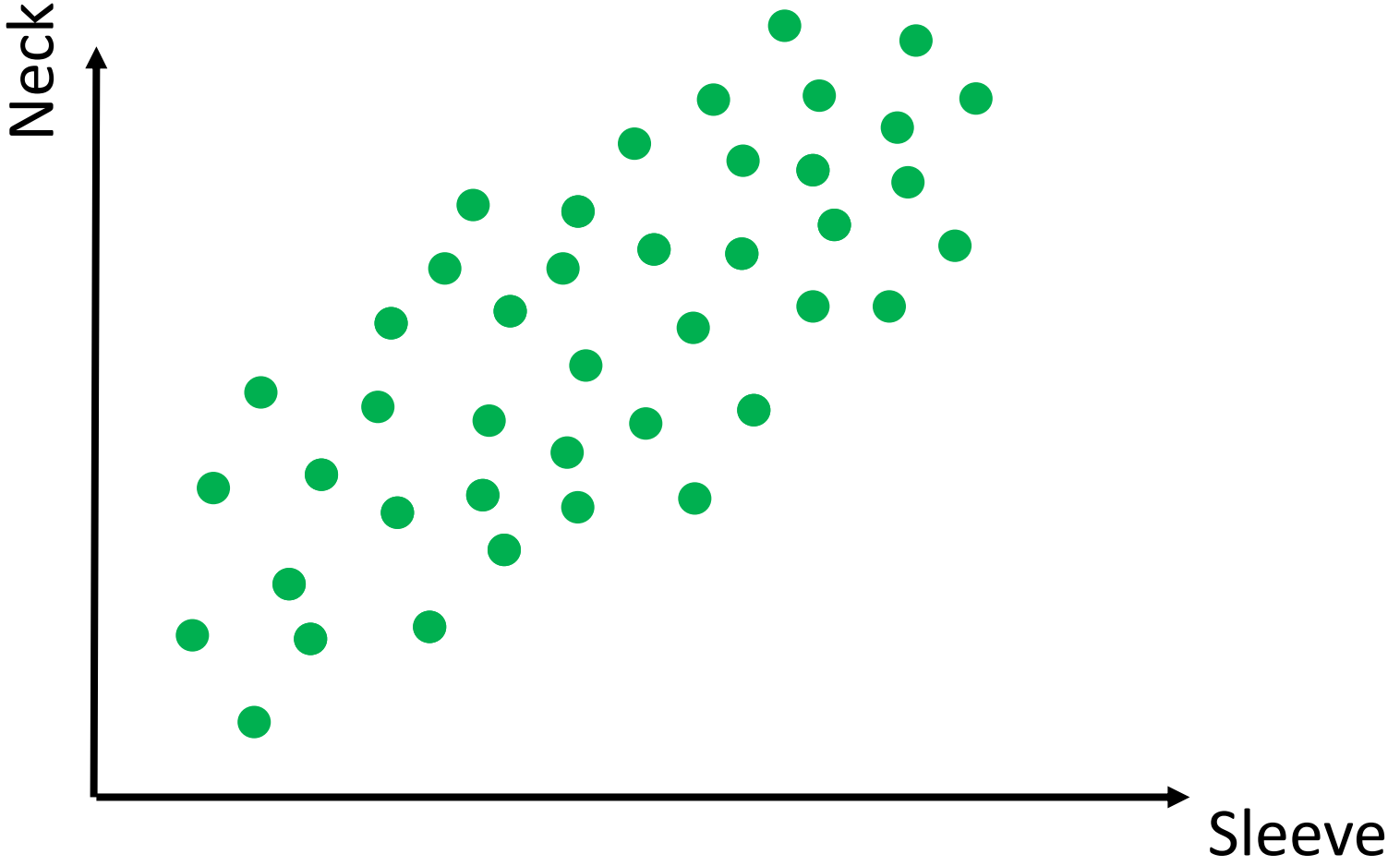
In this approach, we try kmeans clustering for 50 times. Then, we pick the one that gave the lowest cost J.

K-Means for Non-Separated Groups

- Sometimes, K-Means can be very helpful to cluster non-separated data.
- It is particularly very useful for "**product segmentation**" in marketing.
- **Example:** defining clothing size based on sleeve length, neck, chest, …
  - XS, S, M, L, XL, XXL, …

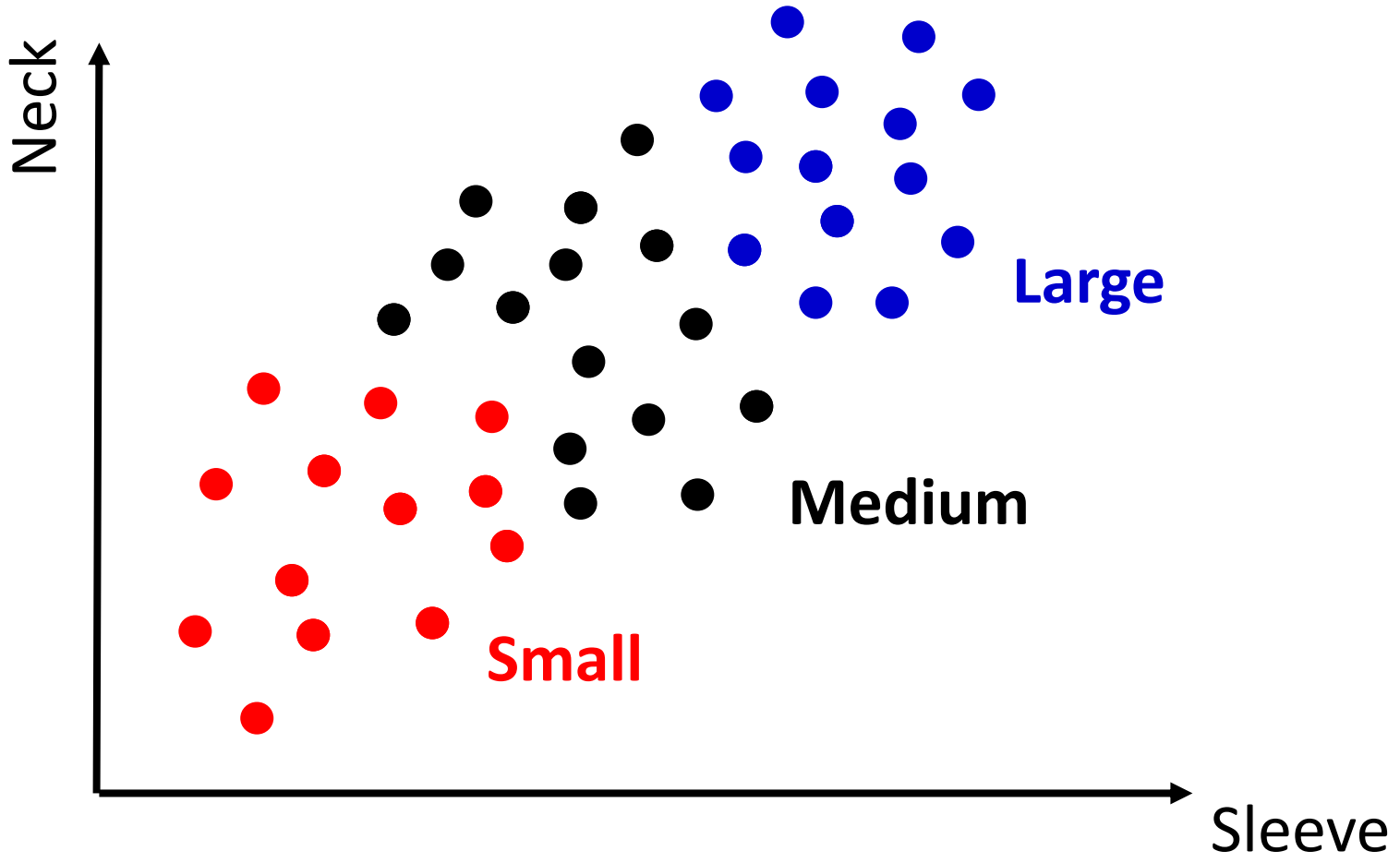
K-Means for Non-Separated Groups

- **Shirt Size:**

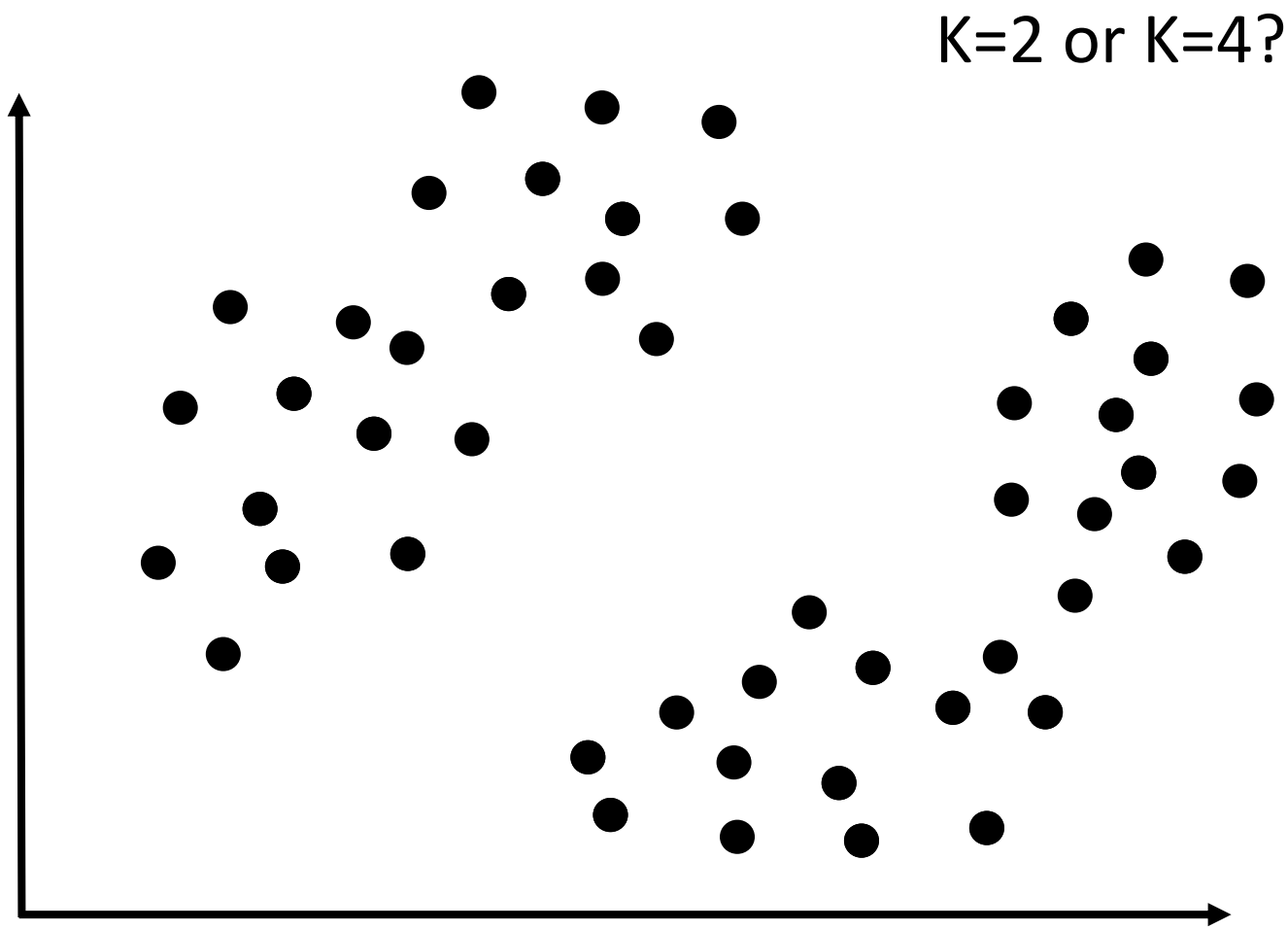


K-Means for Non-Separated Groups

- **Shirt Size:**

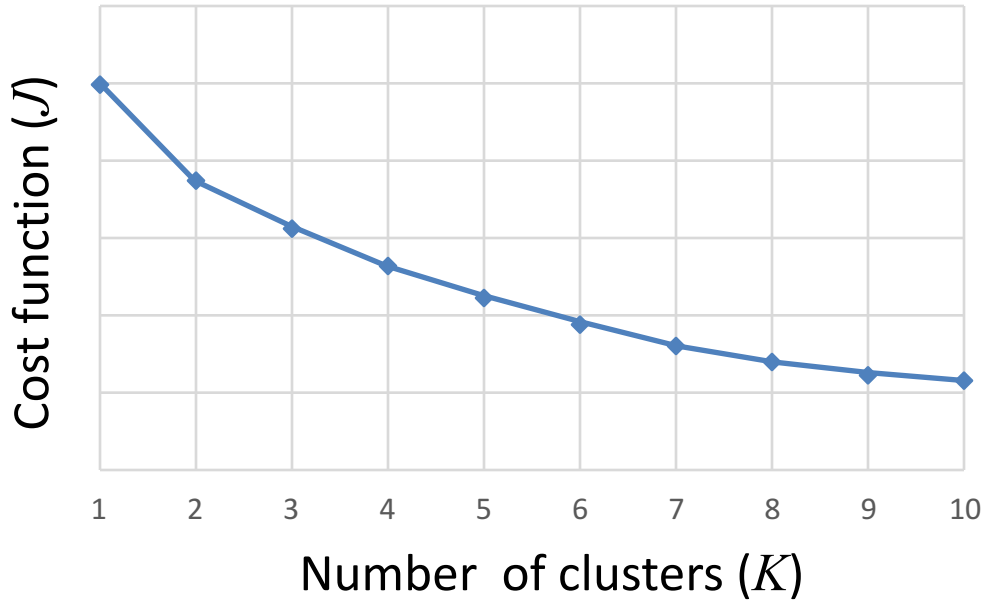


How to choose the Number of Clusters?



How to choose the Number of Clusters?

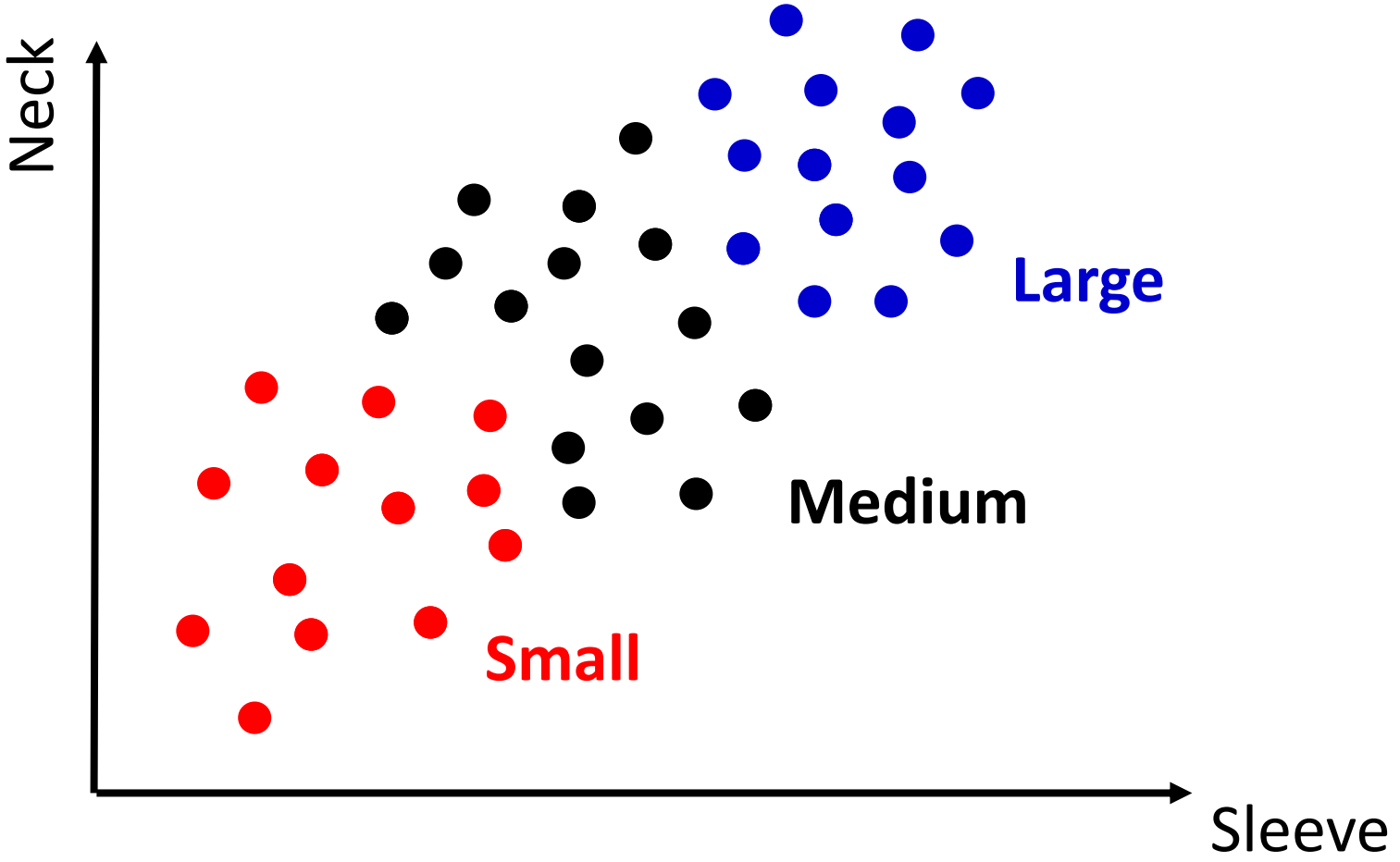
- Trade-off between "Cost Function (J)" and "Number of Clusters(K)":



- Sometimes, it really depends on the application (next example).

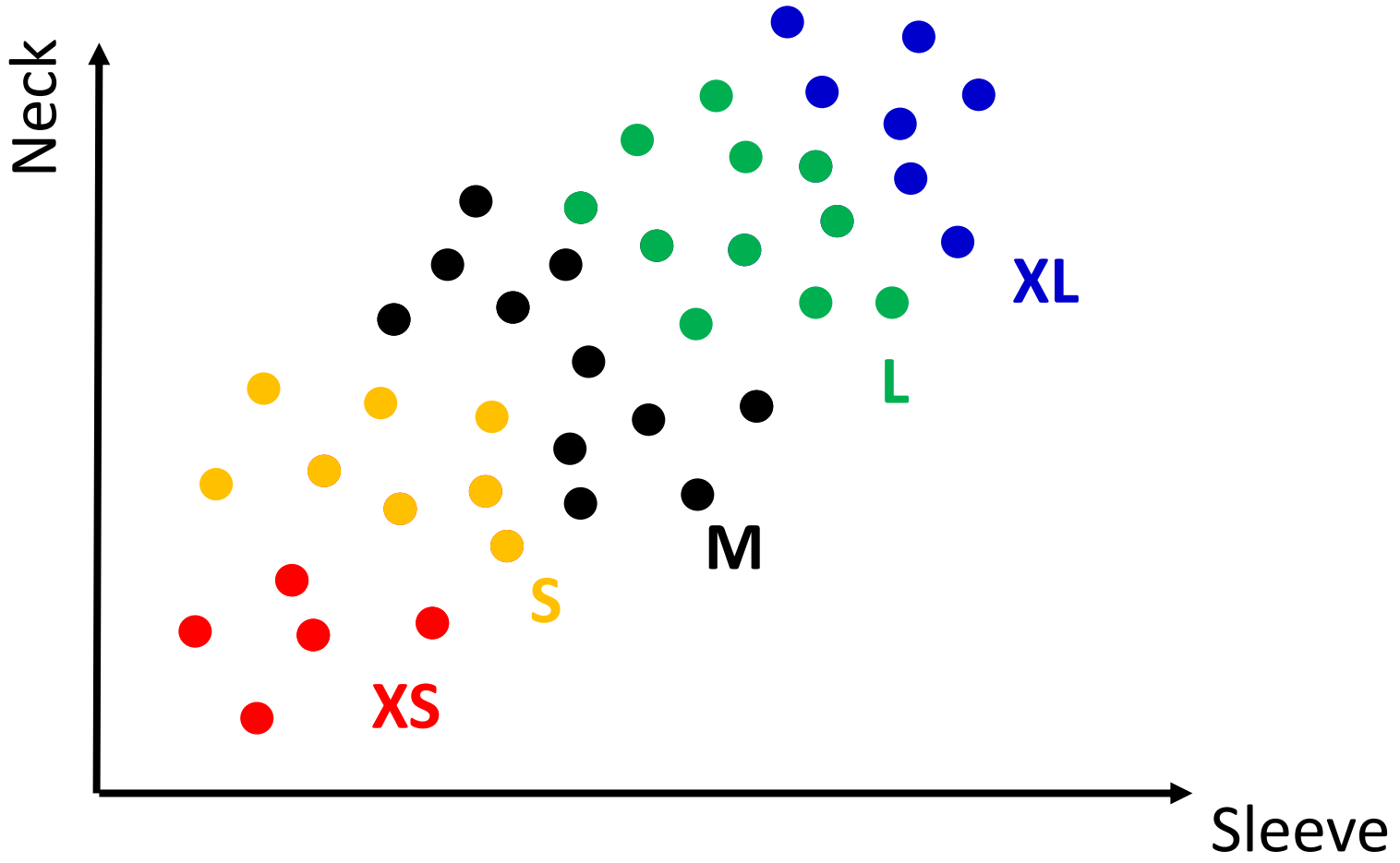
How to choose the Number of Clusters?

- **Shirt Size:** S, M, L



How to choose the Number of Clusters?

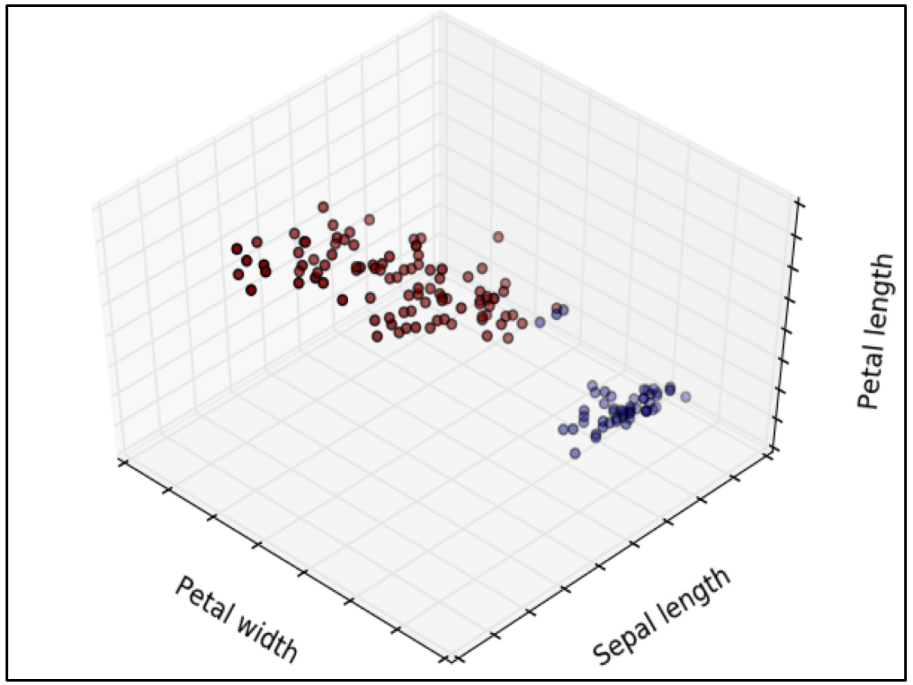
- **Shirt Size:** XS, S, M, L, XL



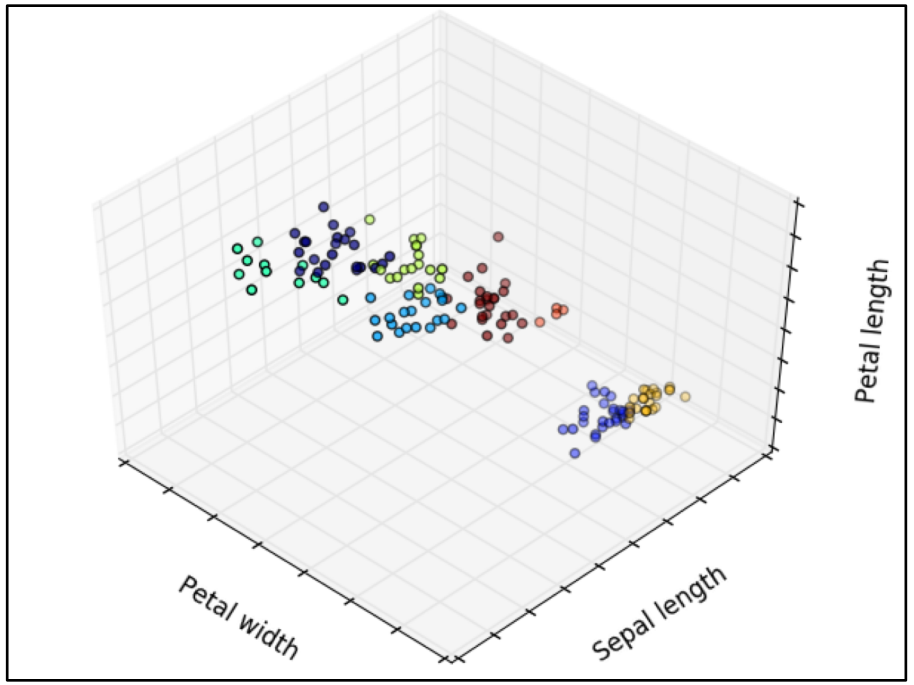
K-Means in Python

```
from sklearn.cluster import KMeans

my_Kmeans = KMeans(n_clusters=3)

my_Kmeans.fit(iris_data)
label_clustered = my_Kmeans.labels_
print(label_clustered)

my_Kmeans.predict(new_iris_data)
```

K-Means for iris dataset



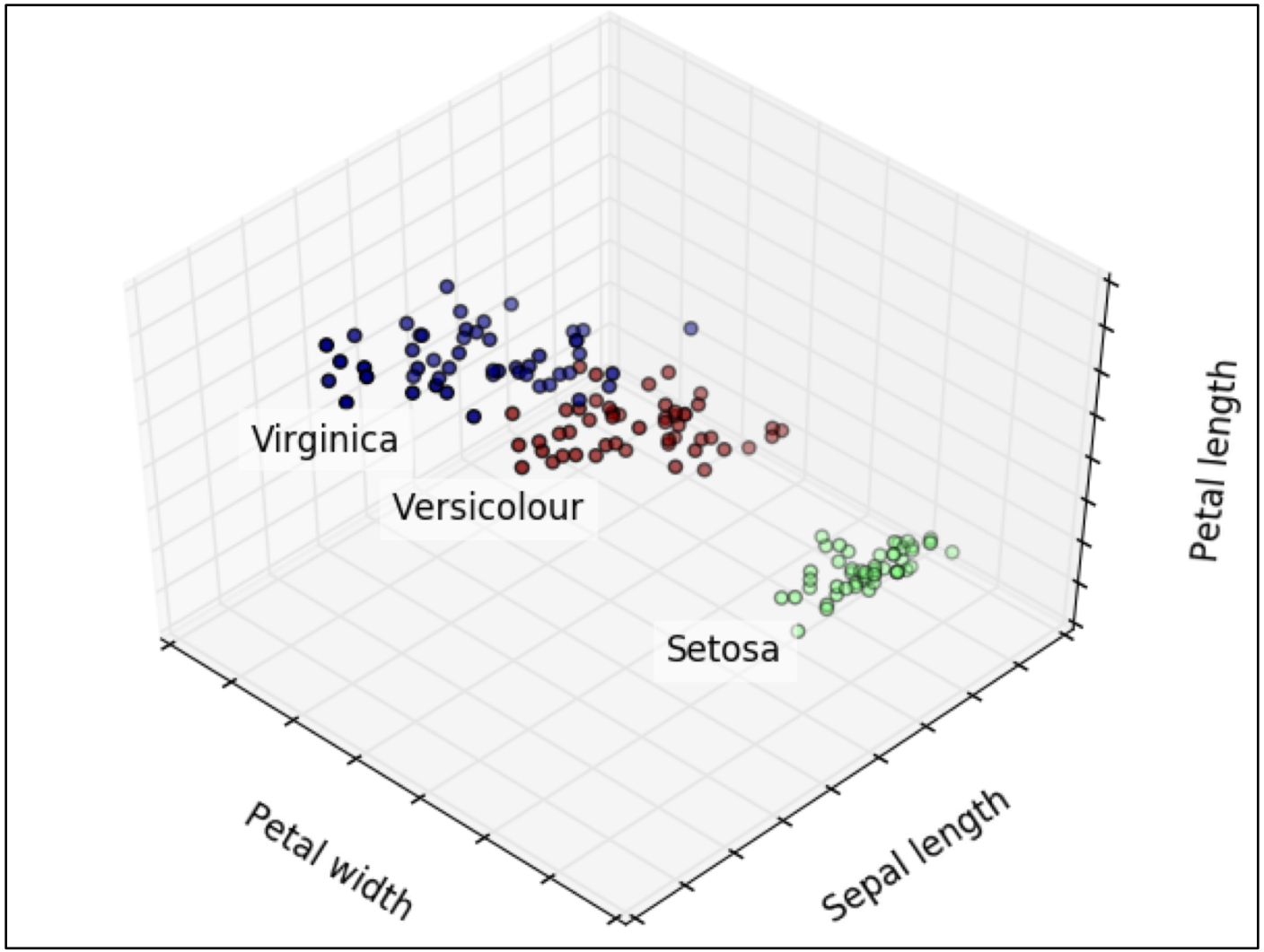
K = 2



K = 8

K-Means for iris dataset

K = 3



Thank You!

Questions?