Introduction to Data Science

(Lecture 17)

Dr. Mohammad Pourhomayoun

Assistant Professor

Computer Science Department

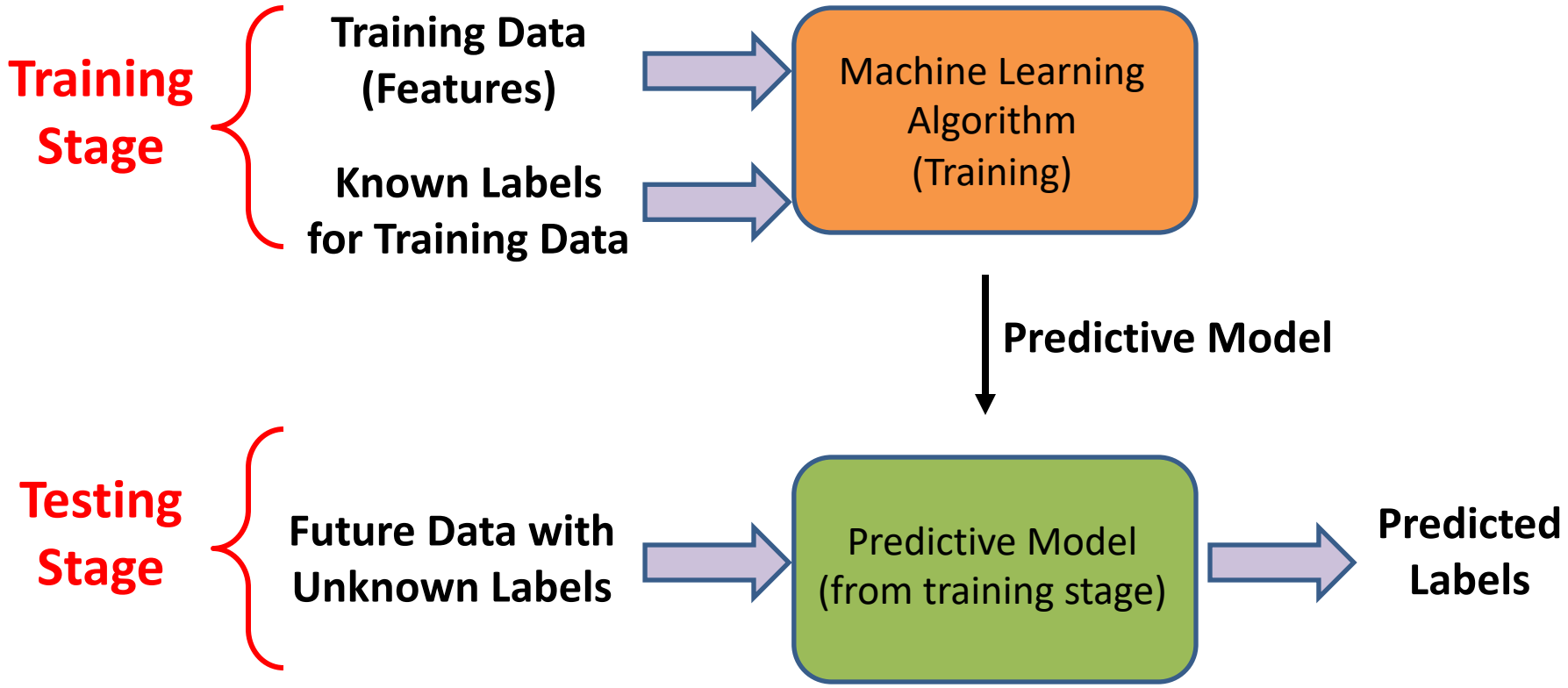
California State University, Los Angeles

Unsupervised Learning

Review

- **Supervised learning algorithms:** Learning from **labeled observations.**
- The Supervised Machine Learning Algorithms that we learned so far:
  - K-Nearest Neighbors classifier (KNN)
  - Decision Tree classifier
  - Linear Regression
  - Logistic Regression classifier
  - Polynomial Regression
  - Random Forest classifier

Supervised Learning: Learning from labeled Data



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.

- **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 cannot 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.

Some Examples and Applications of Unsupervised Learning

Astronomical Data Analytics

- The world's **largest radio telescope** (as big as 30 football fields!) being used to **explore alien life**.

Astronomical Data Analytics

- But, How do the aliens speak!!?
- We never saw this signal before! How to train your ML to recognize them?

Any hidden patterns or structure?

[Figure]: Ian H. Witten, etc., Data Mining: Practical Machine Learning Tools and Techniques, 2011.

Anomaly Detection

- In data science, **anomaly detection** is the identification of samples, items, observations, or events, which do not conform to an expected pattern or other items in a dataset[1].

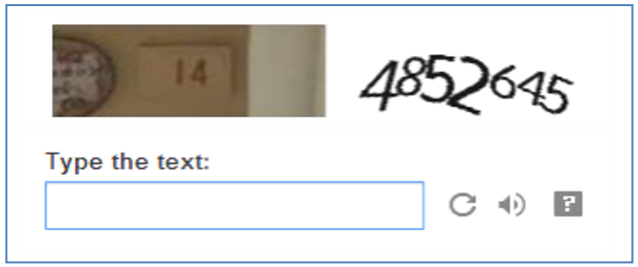
- In many cases, we don't know how an "**Anomaly**" looks like! We just know that it is something **different from a normal pattern!**
- Examples:
  - Bank **Fraud**
  - Structural, system, or signal **Defects**
  - Many medical **Problems**
  - Abnormal behaviors such as a "fall" in patient's activity monitoring

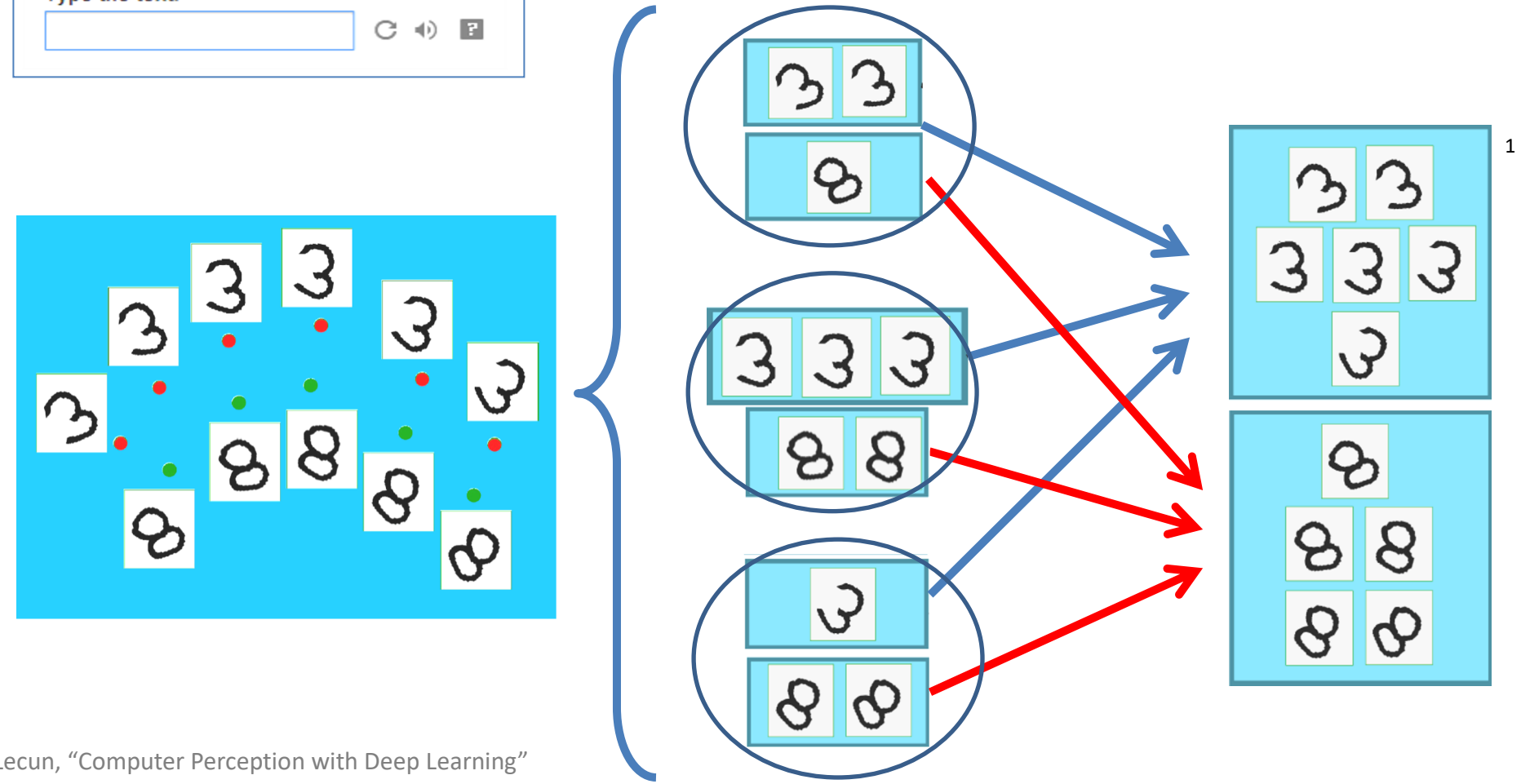
[1]: Wikipedia definition of anomaly

Big Unlabeled Data!

- Cat vs. Dog!
  - **Supervised** or **Unsupervised**, that is the **question**!

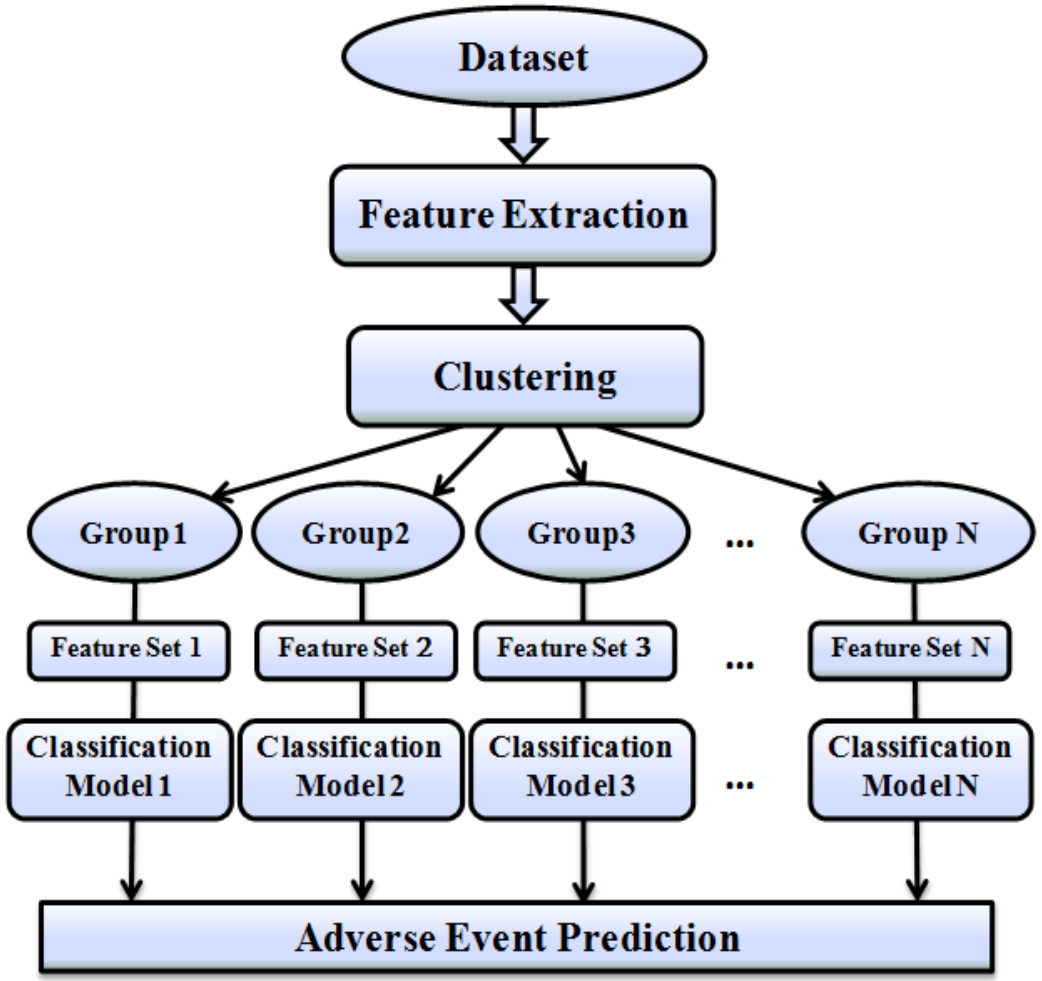
I am Not a Robot!





Ref: Yann Lecun, "Computer Perception with Deep Learning"

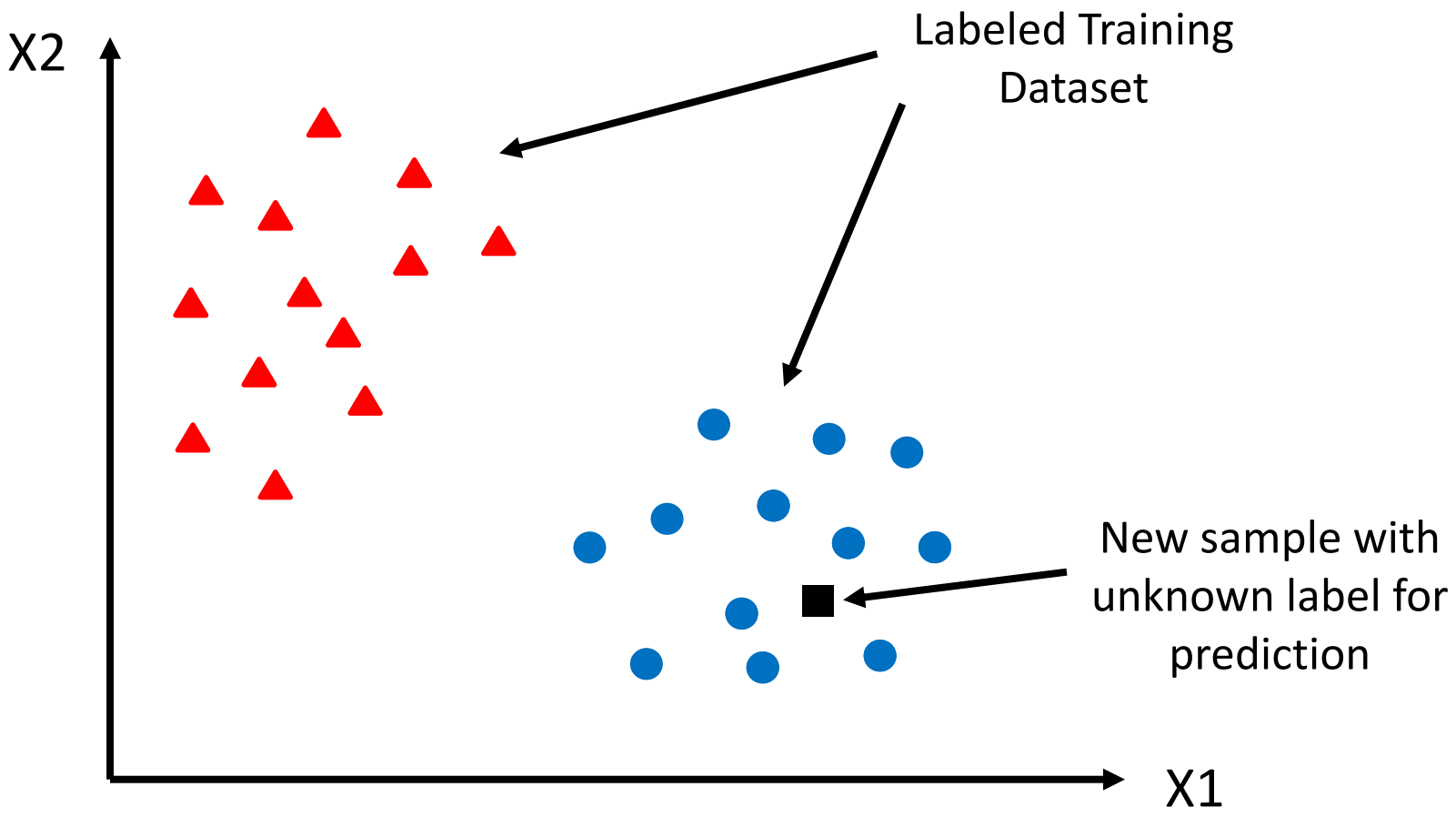
Multiple Prediction Modeling (1)



[1]: M. Pourhomayoun, etc., "Multiple Model Analytics for Adverse Event Prediction in Remote Health Monitoring Systems," IEEE Conf. on Healthcare Innovation & Point-of-Care Tech.

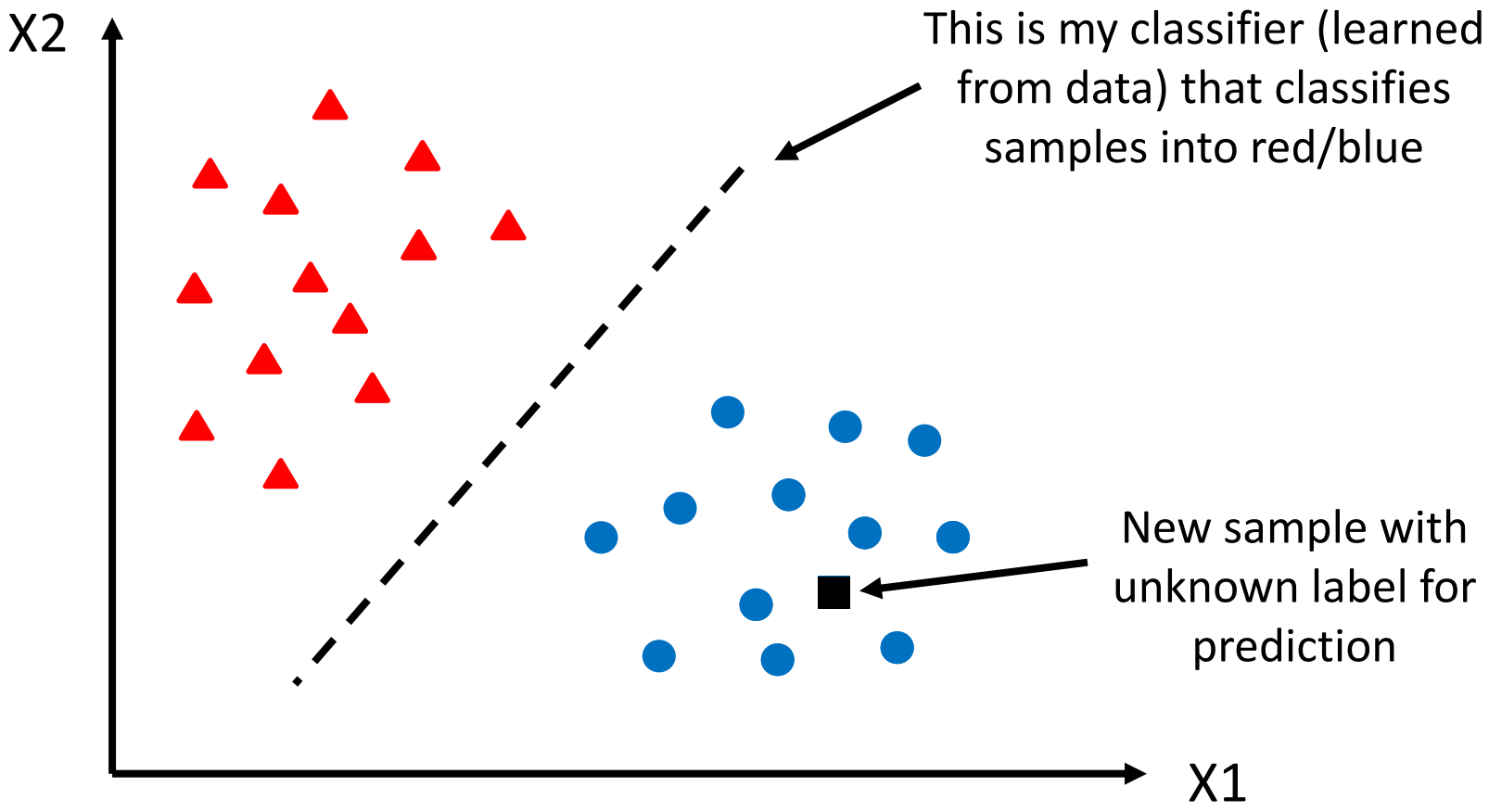
Unsupervised Learning
How does it work!!?

Review: Supervised Learning



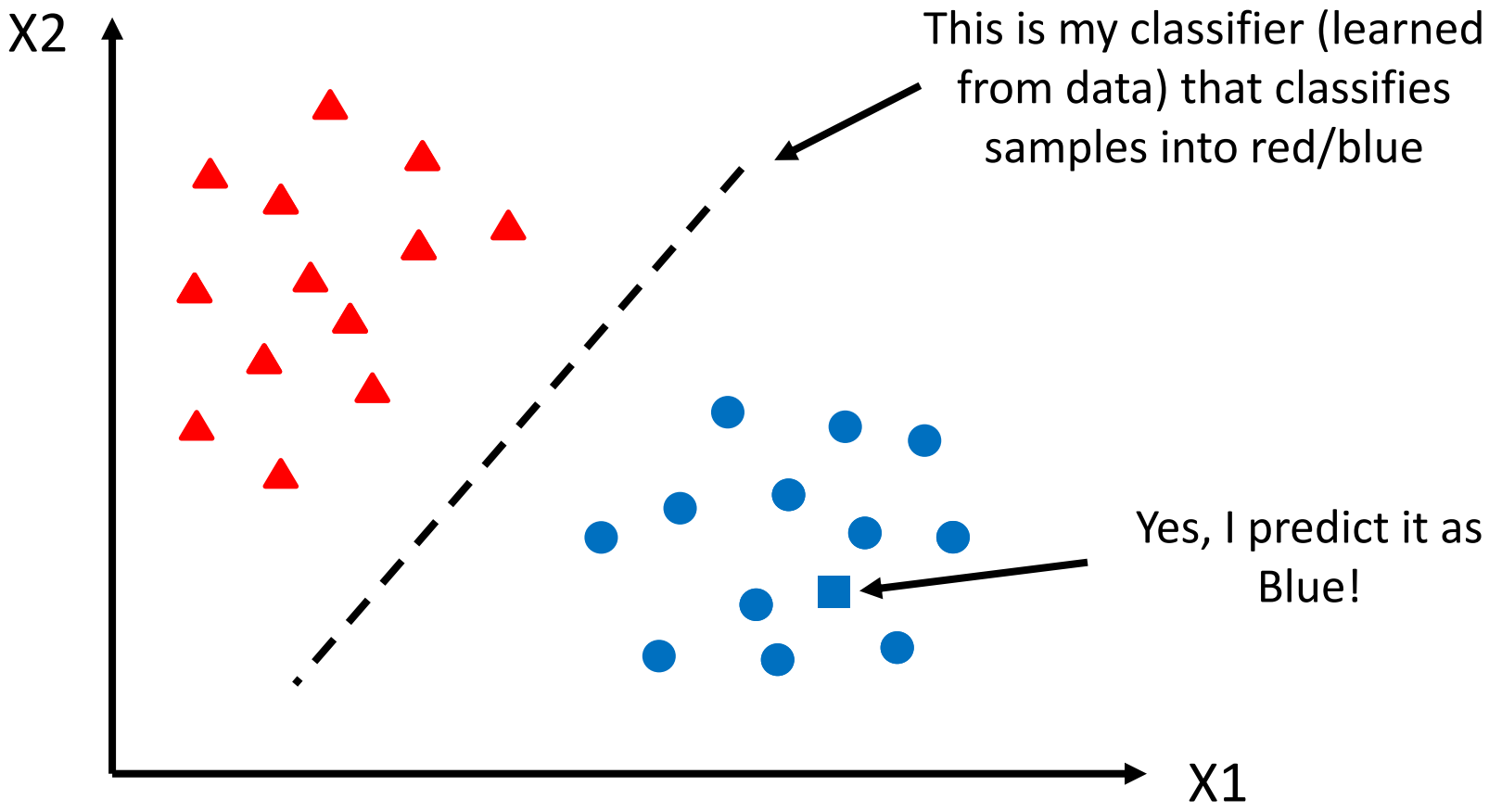
Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots, (x^{(m)}, y^{(m)})\}$

Review: Supervised Learning



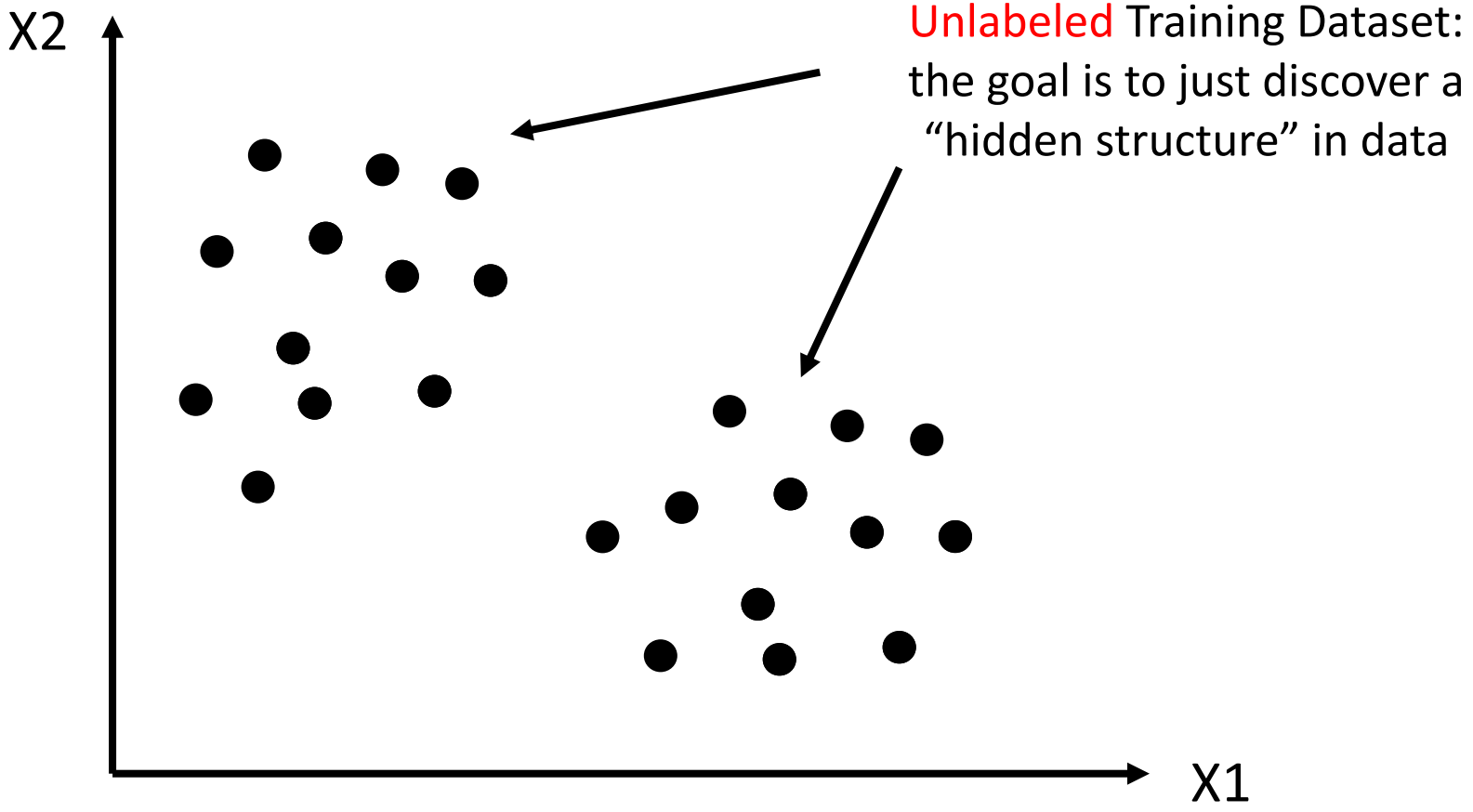
Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots, (x^{(m)}, y^{(m)})\}$

Review: Supervised Learning



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

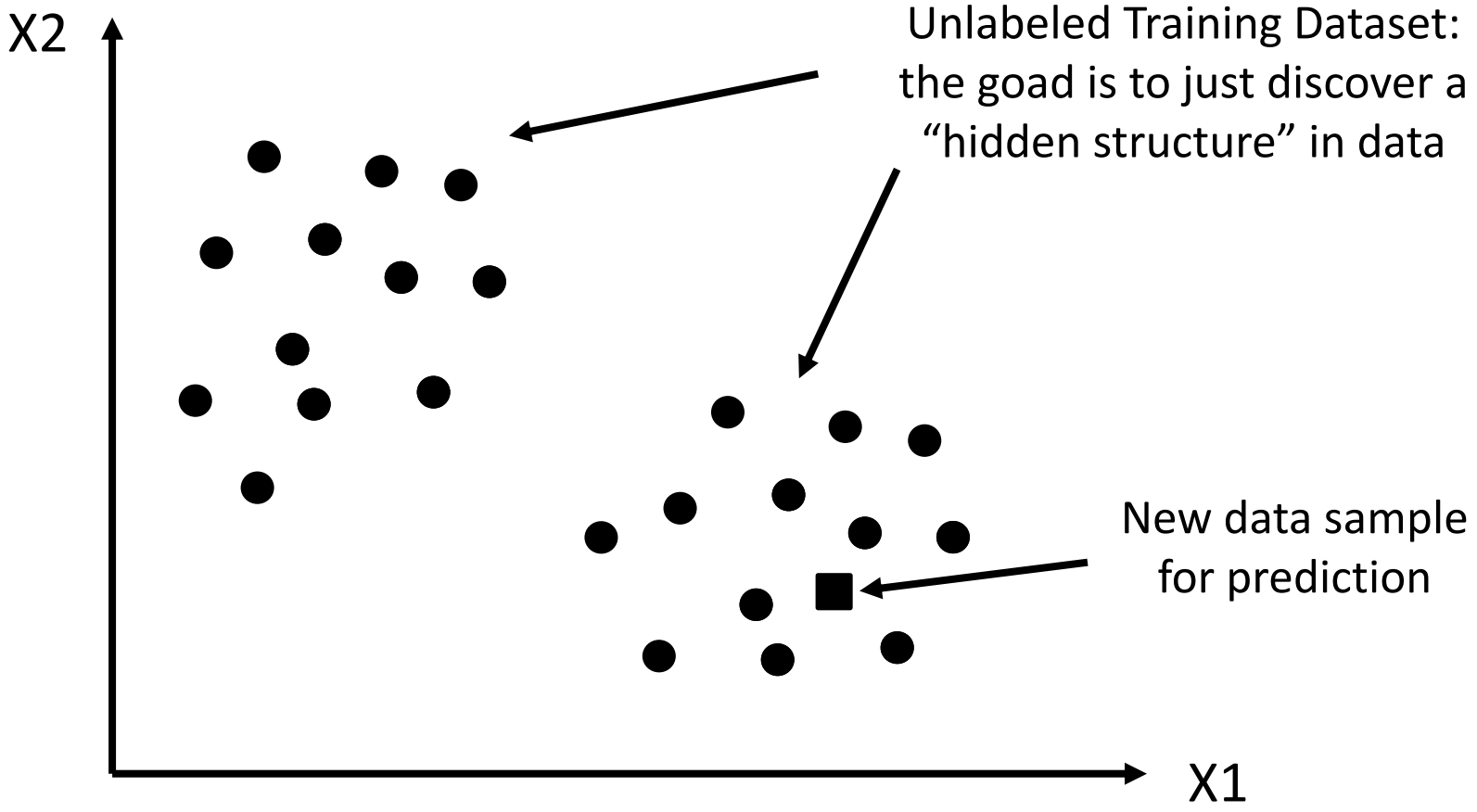
Unsupervised Learning



Unlabeled Training Dataset:
the goal is to just discover a "hidden structure" in data

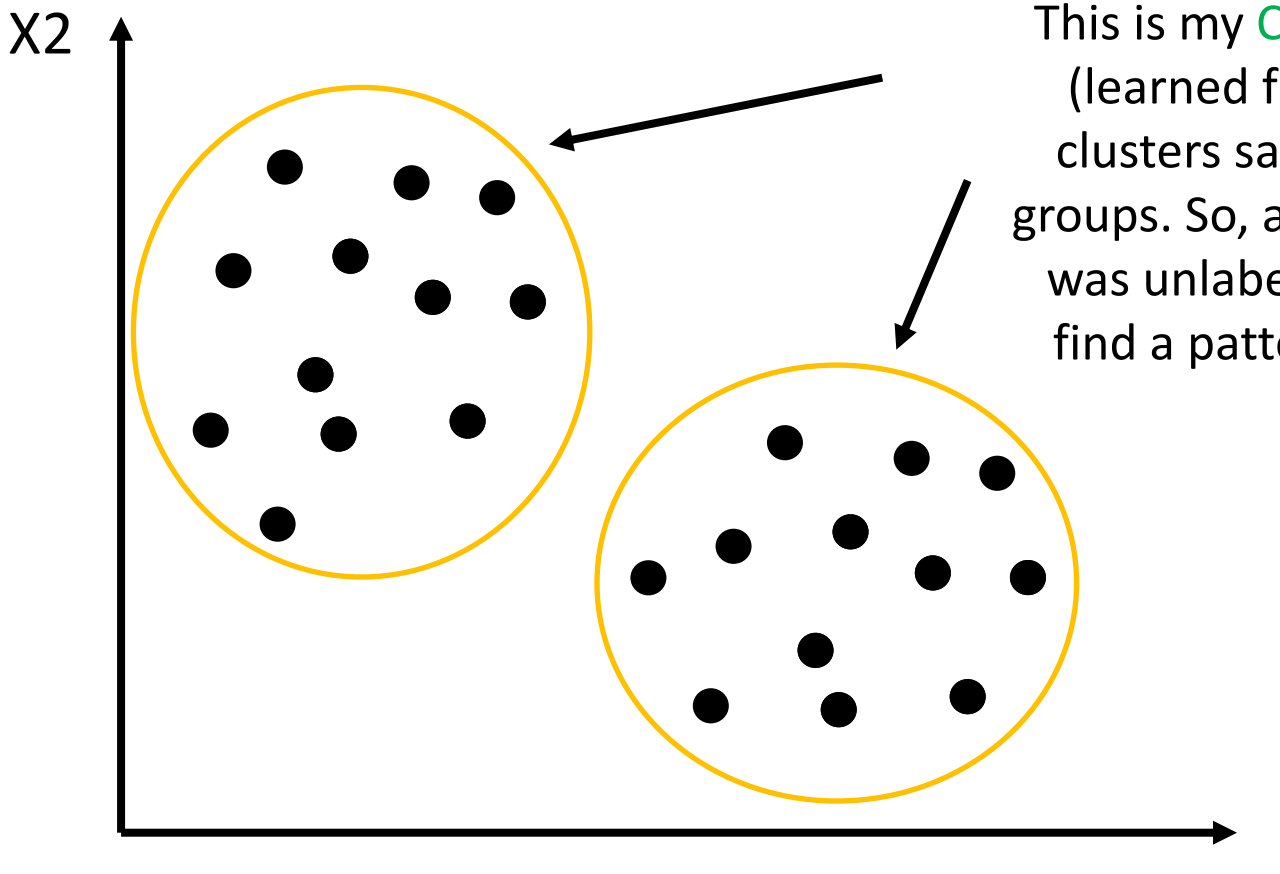
Training set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$

Unsupervised Learning



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

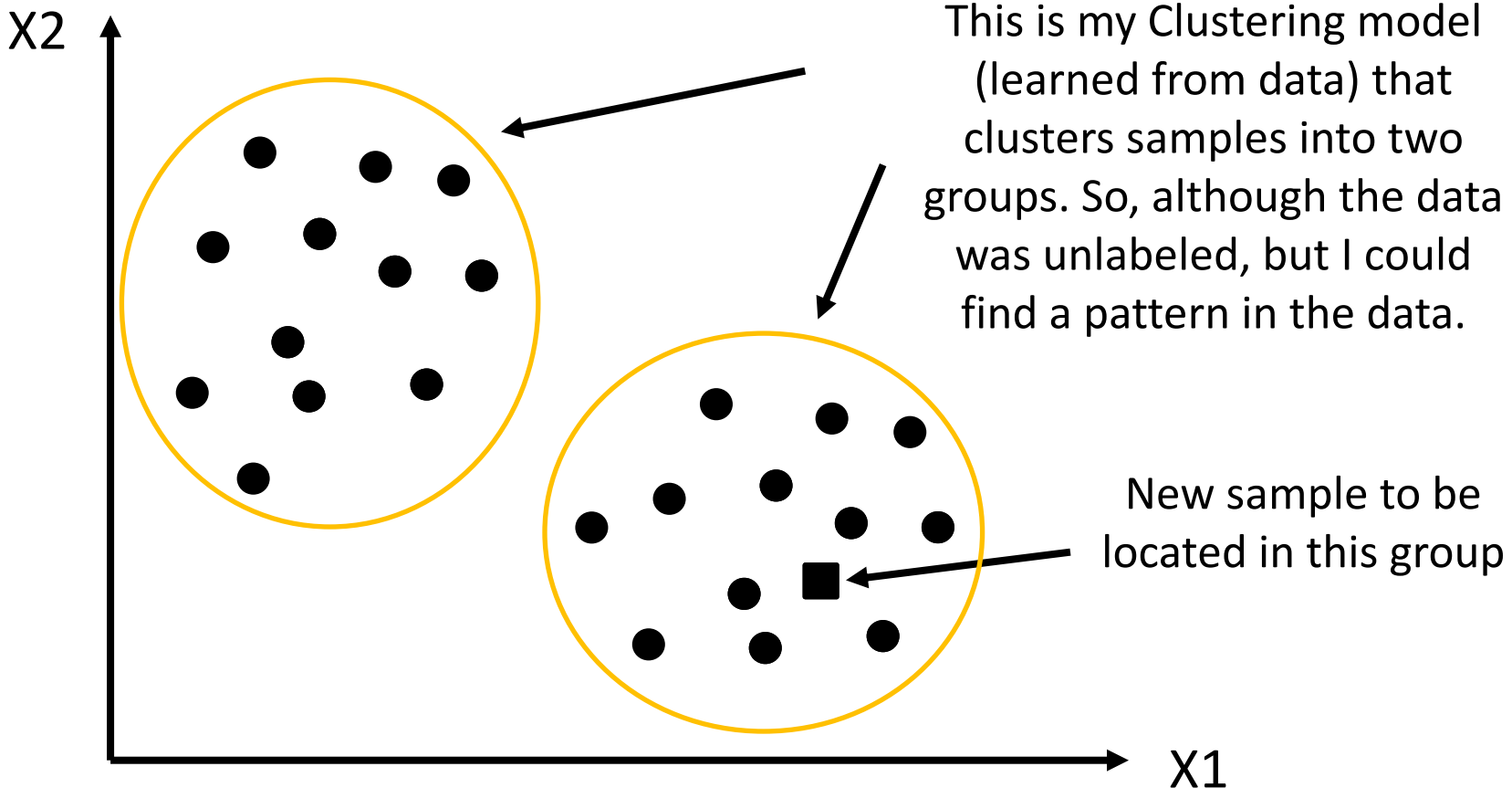
Unsupervised Learning



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.

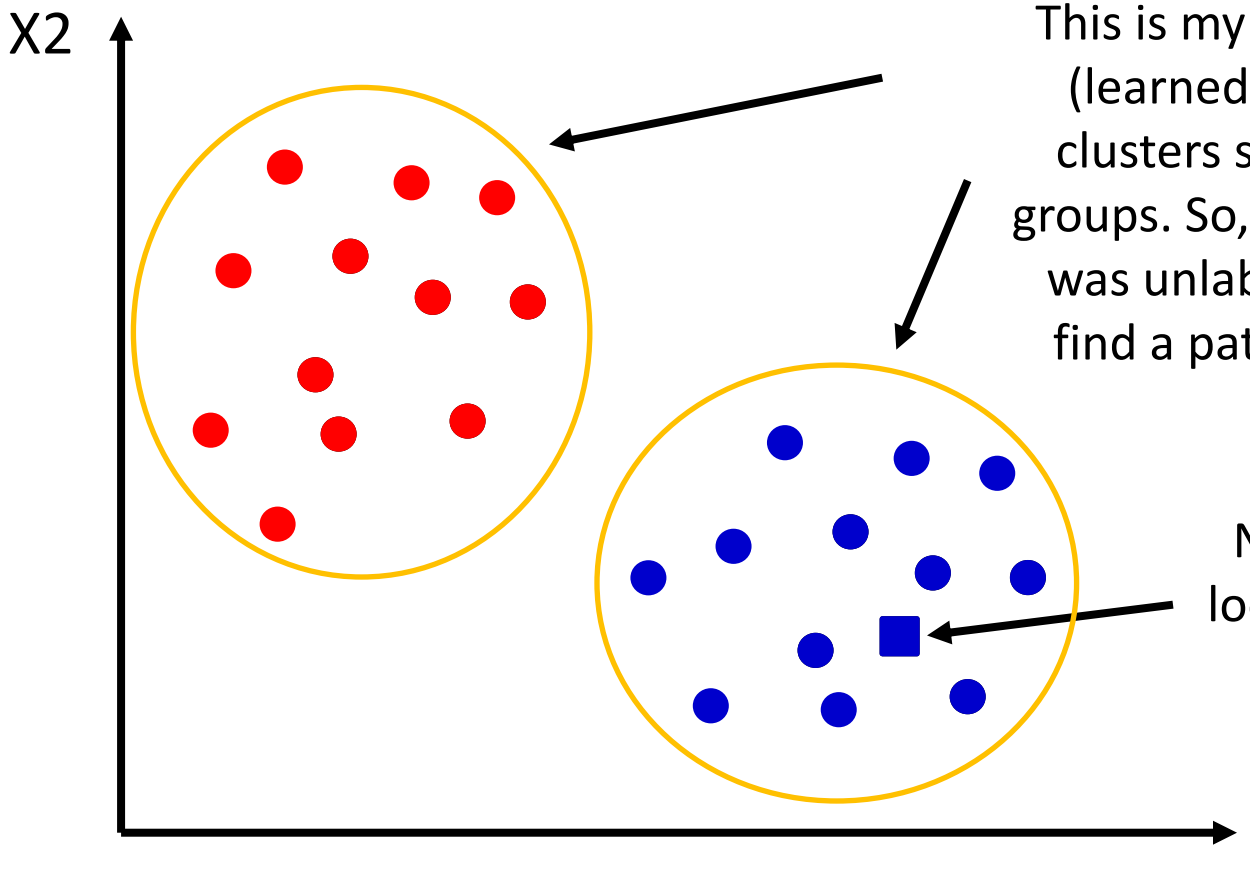
Training set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$

Unsupervised Learning



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

Review: 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.

- **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.

Two Commonly-Used Approaches of Unsupervised Learning

- **Clustering:** To partition data samples into homogeneous groups (clusters) based on similarity or common properties.
  - e.g., to identify "communities" within large groups of people in social networks.

- **Dimensionality Reduction:** Reducing the dimensionality of the data in an unsupervised fashion!

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** in a feature space so that each sample belongs to the cluster with the **nearest mean** (centroid).

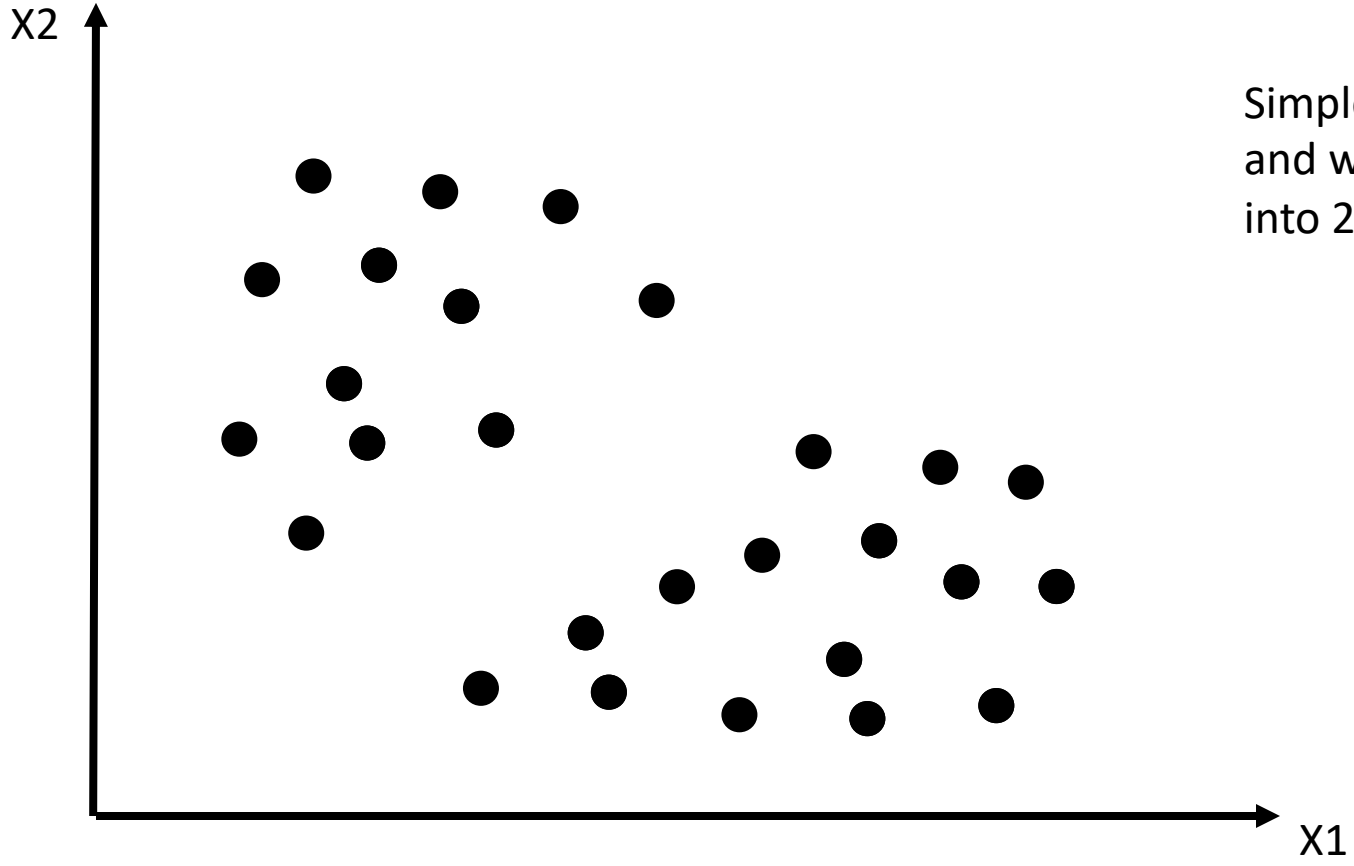
- It is an ***unsupervised learning*** algorithm: we don't have a label demonstrating how the data samples should be grouped!

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

K-Means Clustering Algorithm

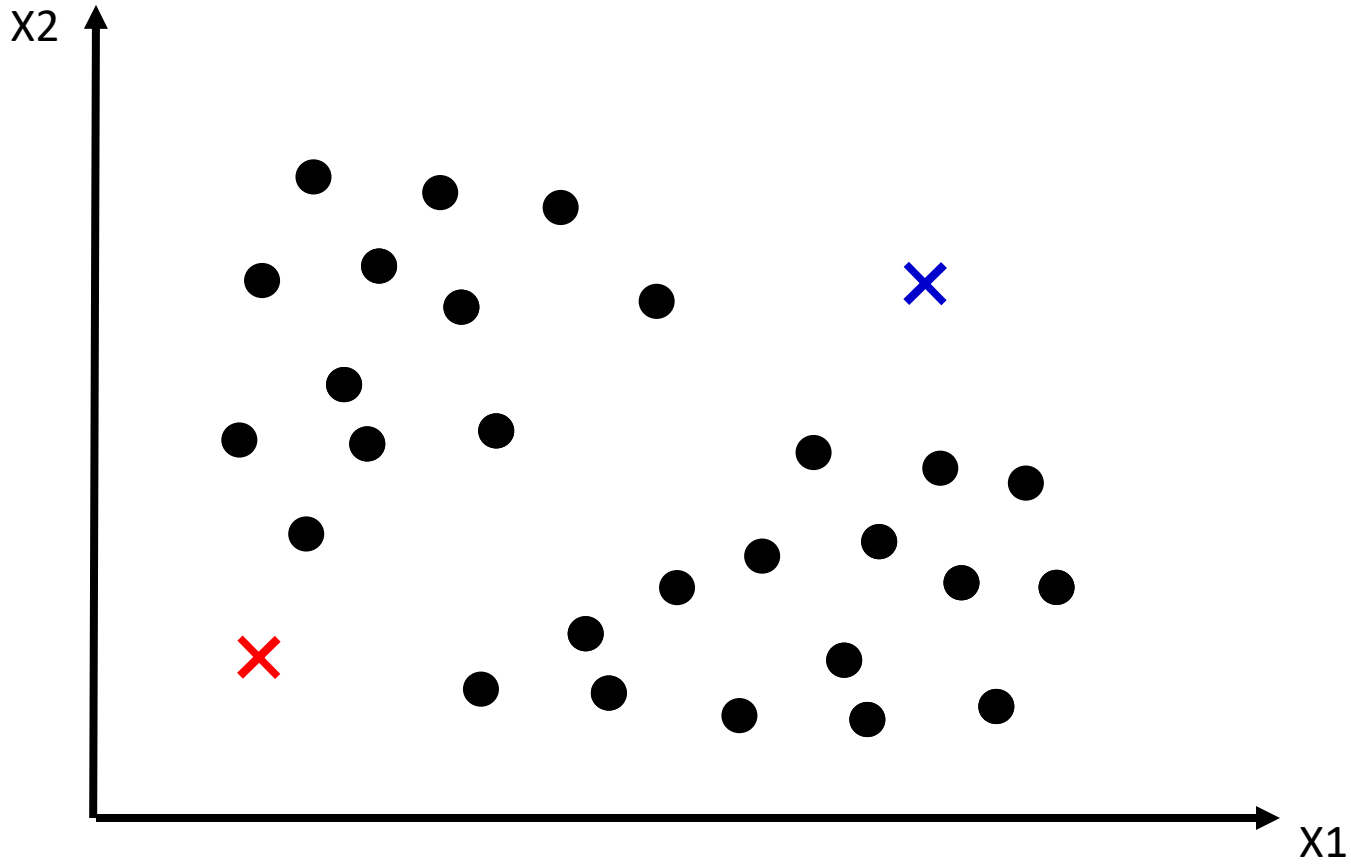
- **Step 0: Initialization:** set K random points in feature space as initial centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K \in \mathbb{R}^n$
- **Step 1: Cluster Assignment:** Assign each data sample to the cluster of the nearest centroid point (for each point, we need 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



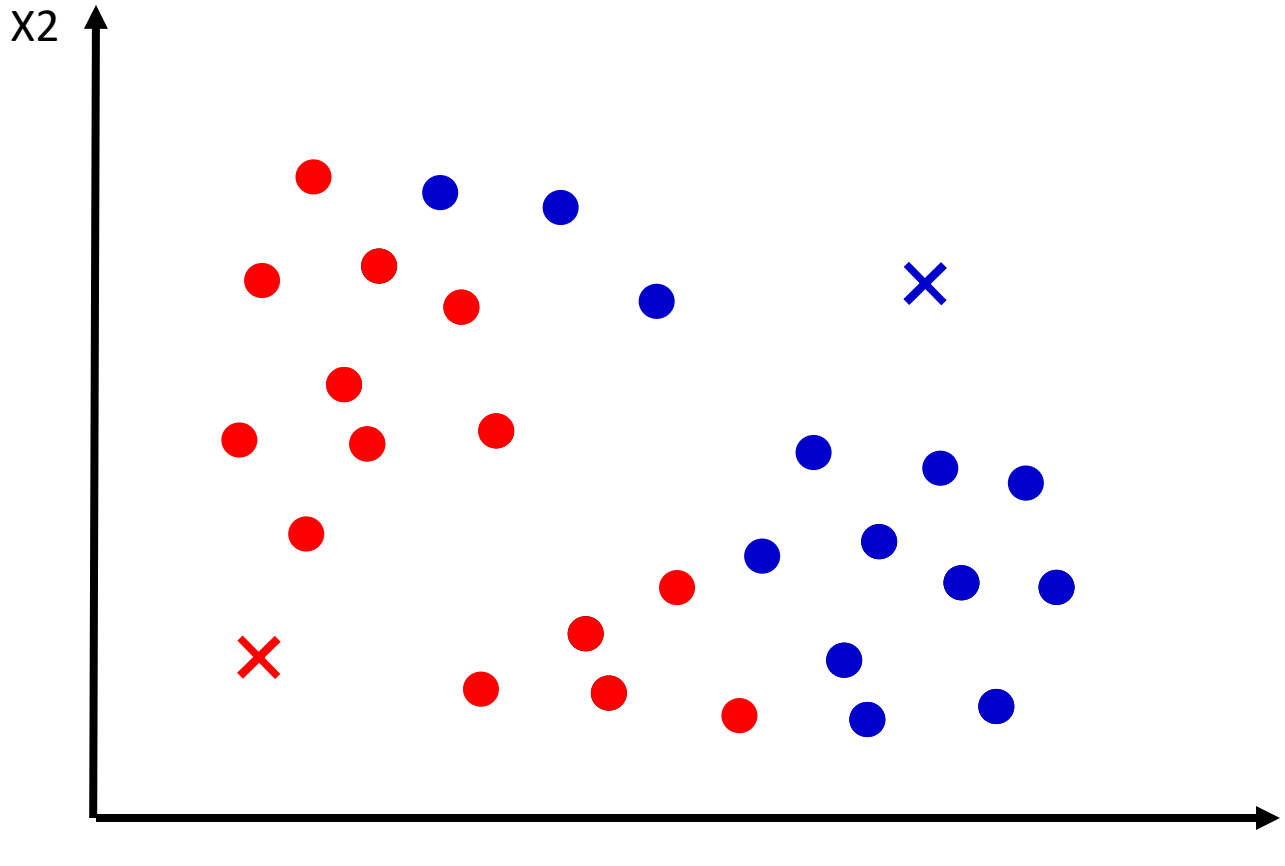
Simplest Case: 2 features X1 & X2, and we want to cluster the data into 2 clusters (K=2).

Step 0: Random Initialization



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

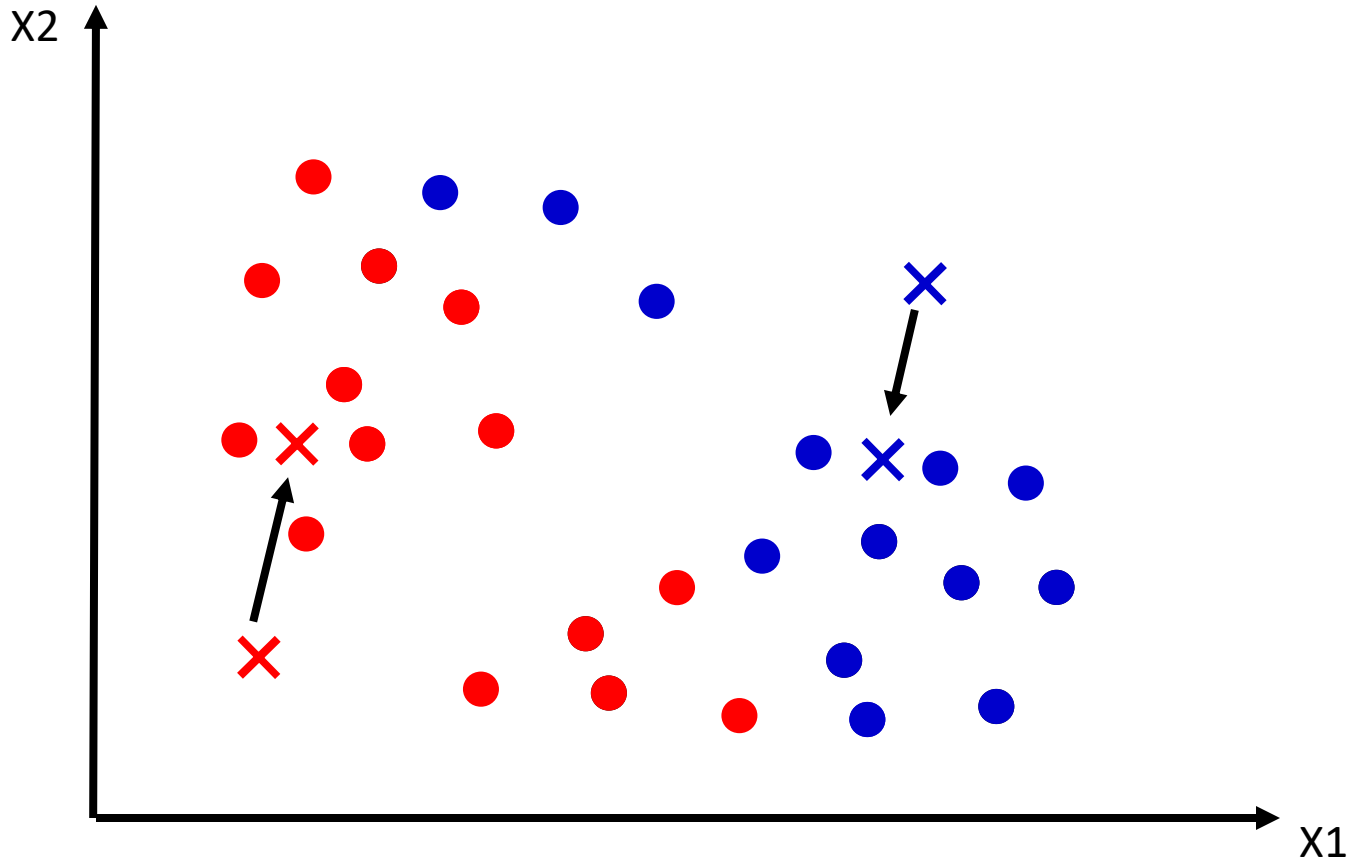
Step 1: Assign the Points



NOTICE: Blue and Red are not labels here! They just show cluster1 and cluster2.

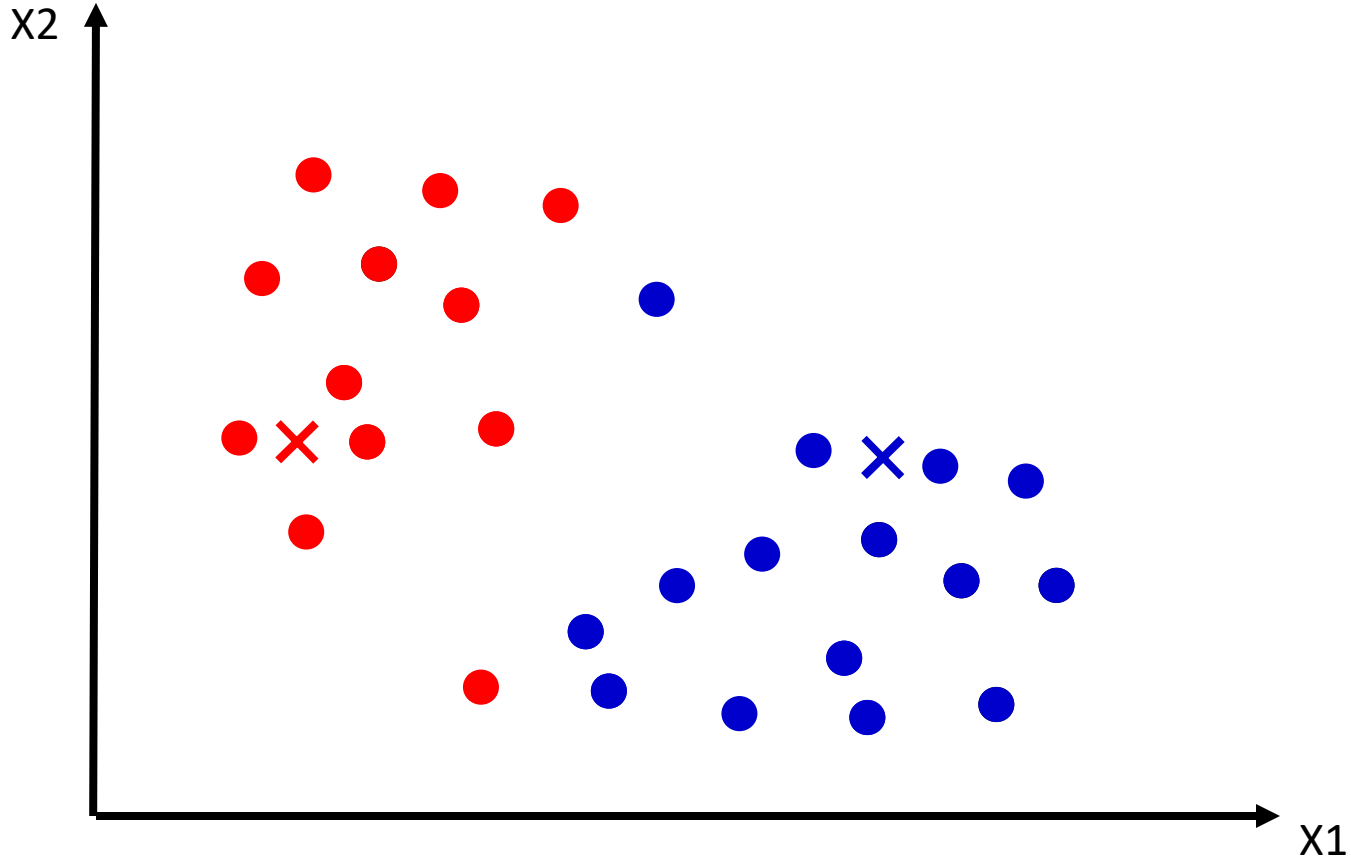
Cluster Assignment: Assign each data sample to the cluster of the nearest centroid point.

Step 2: Find new Centroid



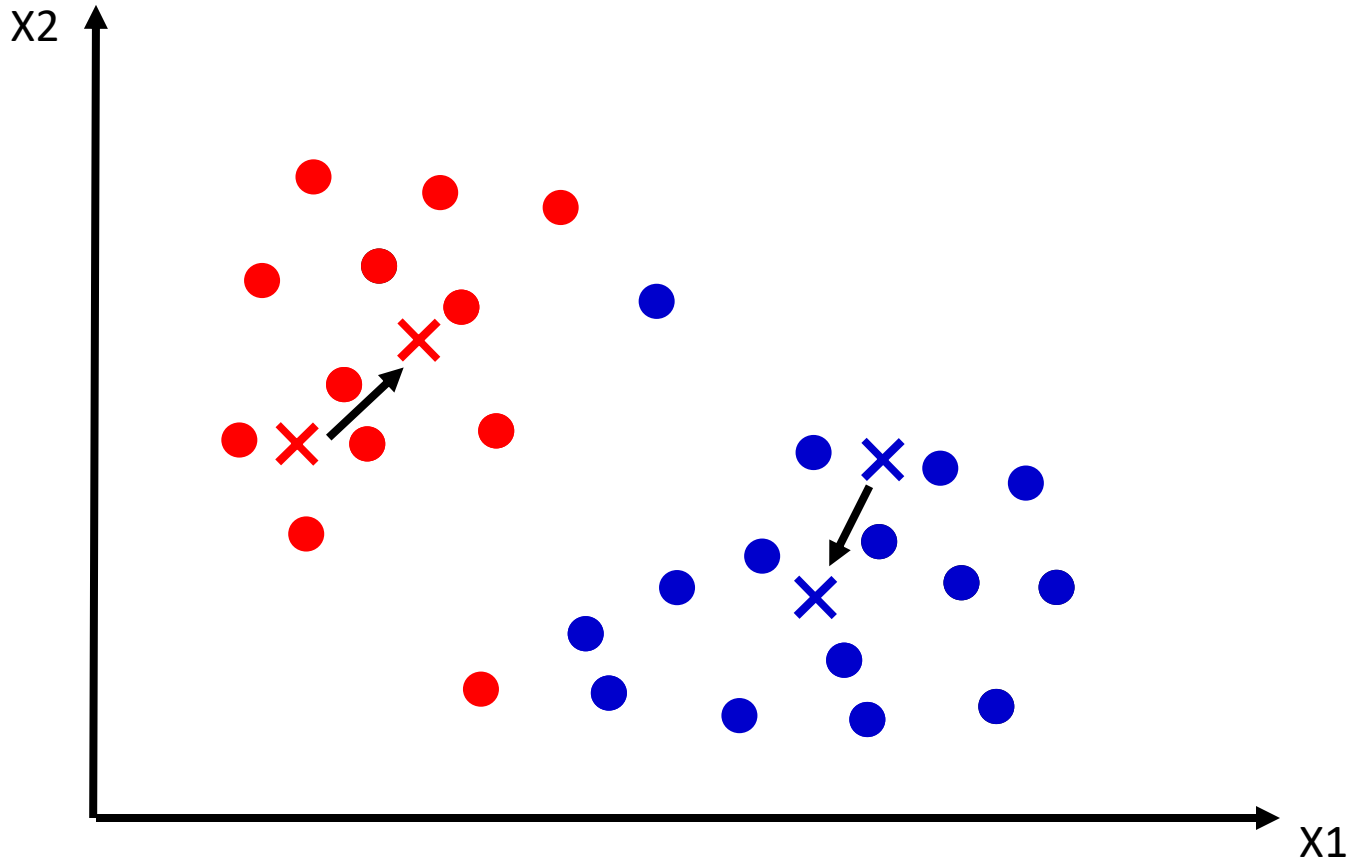
Centroid Move: Update centroid locations to the mean location of the members of the current cluster.

Back to Step 1: Assign the Points



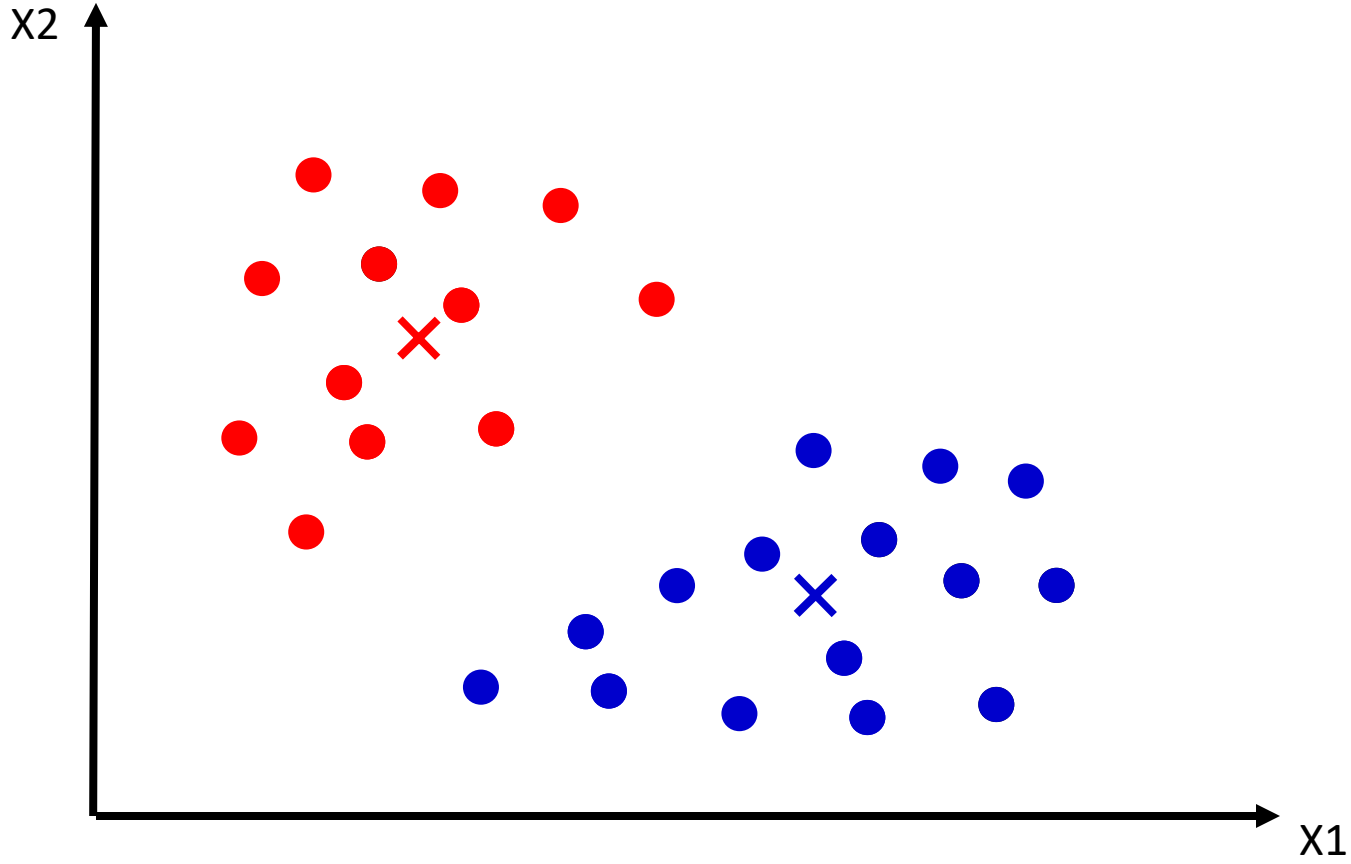
Cluster Assignment: Assign each data sample to the cluster of the nearest centroid point.

Step 2: Find new Centroid



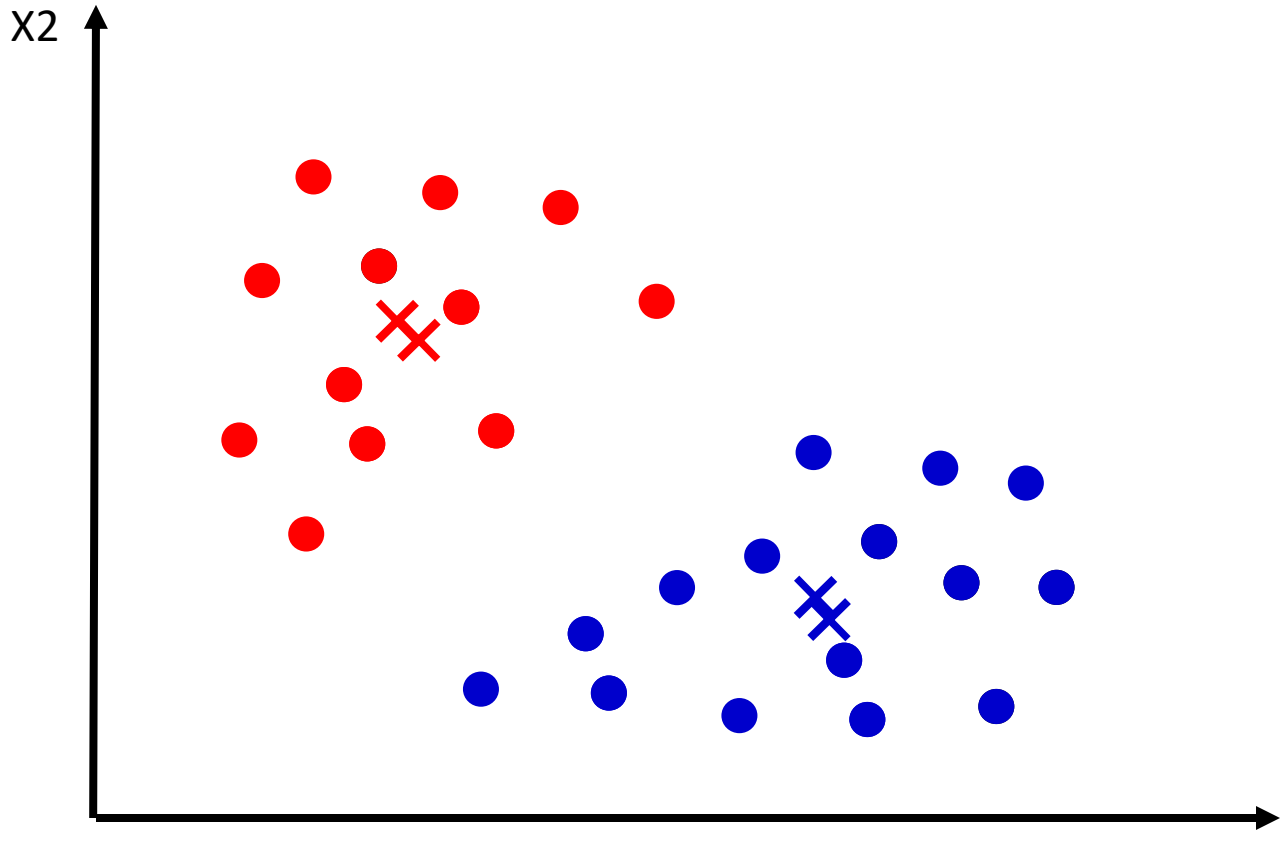
Centroid Move: Update centroid locations to the mean location of the members of the current cluster.

Back to Step 1: Assign the Points



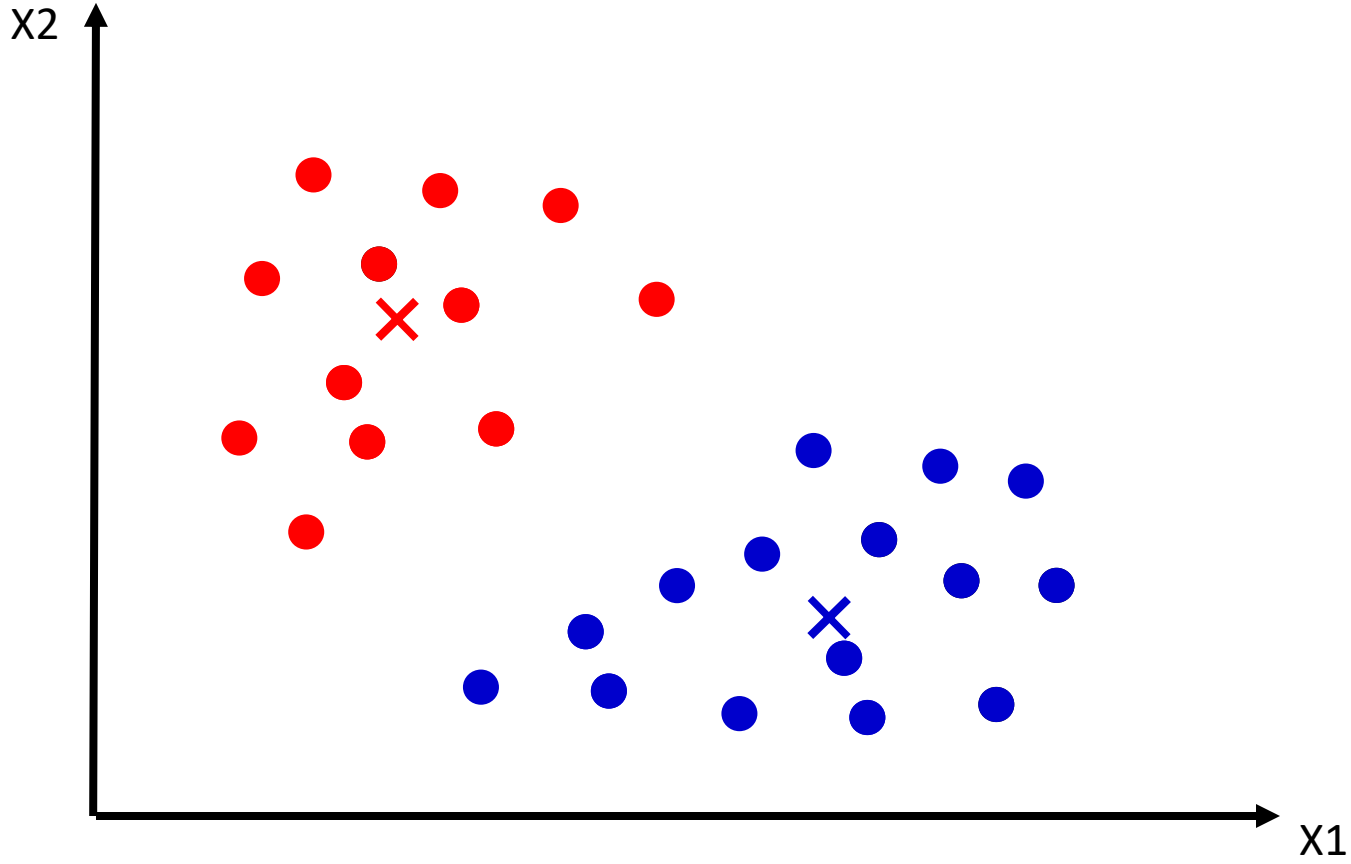
Cluster Assignment: Assign each data sample to the cluster of the nearest centroid point.

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).

Thank You!

Questions?