GENERAL ASSEMBLY

# DATA SCIENCE
## CLUSTERING

# AGENDA 

## I. SUPERVISED AND UNSUPERVISED LEARNING
## II. K-MEANS CLUSTERING
## III. DBSCAN CLUSTERING

# I. SUPERVISED AND UNSUPERVISED LEARNING

# MACHINE LEARNING



One definition: “Machine learning is the semi-automatic extraction of knowledge from data.”

- **Knowledge from data:** Starts with a question that might be answerable using data
- **Automatic extraction:** A computer provides the insight
- **Semi-automatic:** Requires many smart decisions by a human

# SUPERVISED LEARNING



There are two main categories of machine learning: **supervised learning** and **unsupervised learning**.

**Supervised learning** (aka “predictive modeling”):

- Predict an outcome based on input data
- Example: predict whether an email is spam or ham
- Goal is “generalization”

There are two categories of supervised learning:

**Regression**

- Outcome we are trying to predict is continuous
- Examples: price, blood pressure

**Classification**

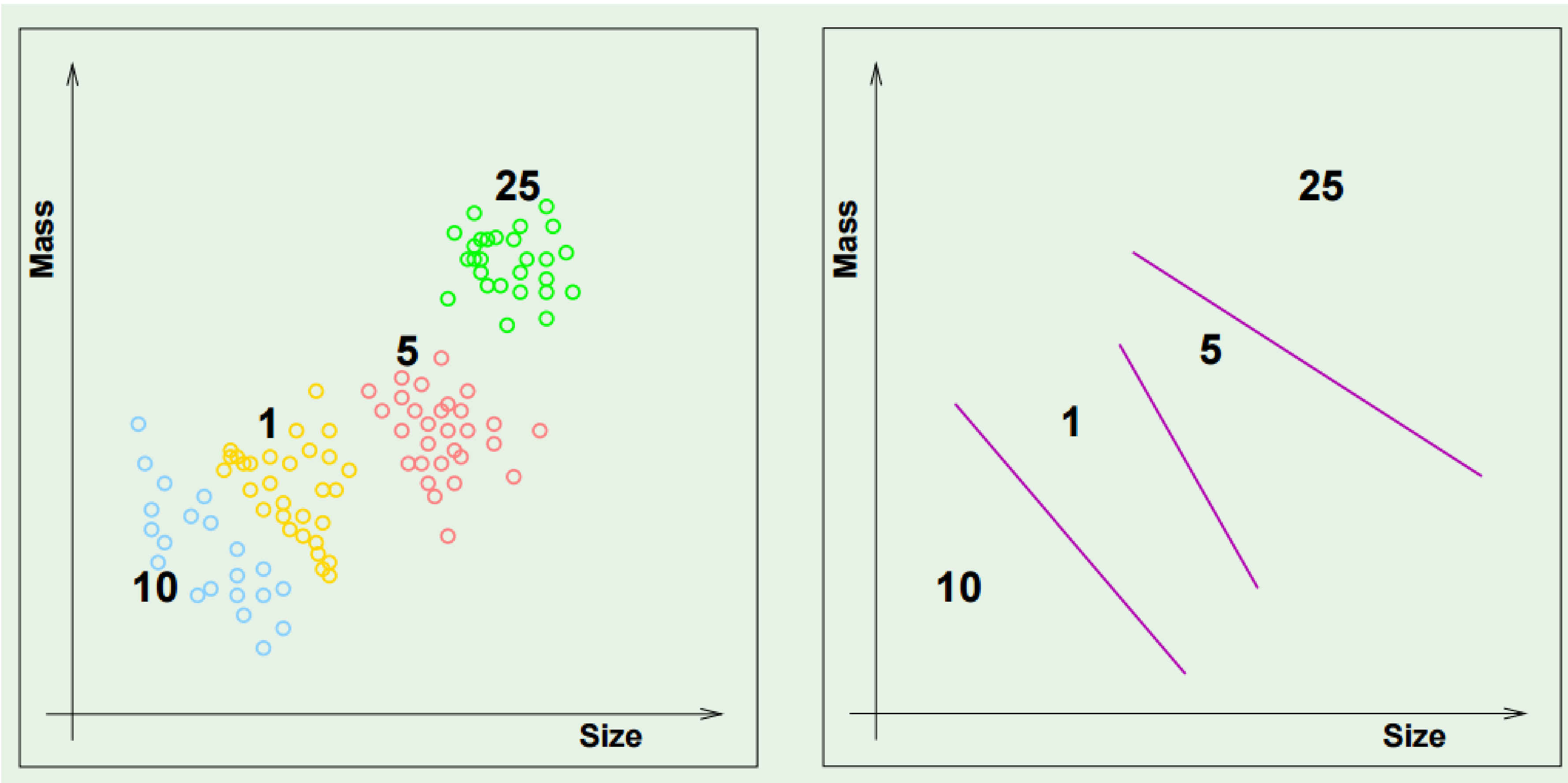
- Outcome we are trying to predict is categorical (values in a finite set)
- Examples: spam/ham, cancer class of tissue sample

## Supervised learning example: Coin classifier

- Observations: Coins
- Features: Size and mass
- Response: Coin type, hand-labeled

1. Train a **machine learning model** using **labeled data**
   - Model learns the relationship between the features and the coin type

2. Make predictions on **new data** for which the response is unknown
   - Give it a new coin, predicts the coin type automatically


Mass
Size
25
5
1
10
Mass
Size
25
5
1
10

# UNSUPERVISED LEARNING

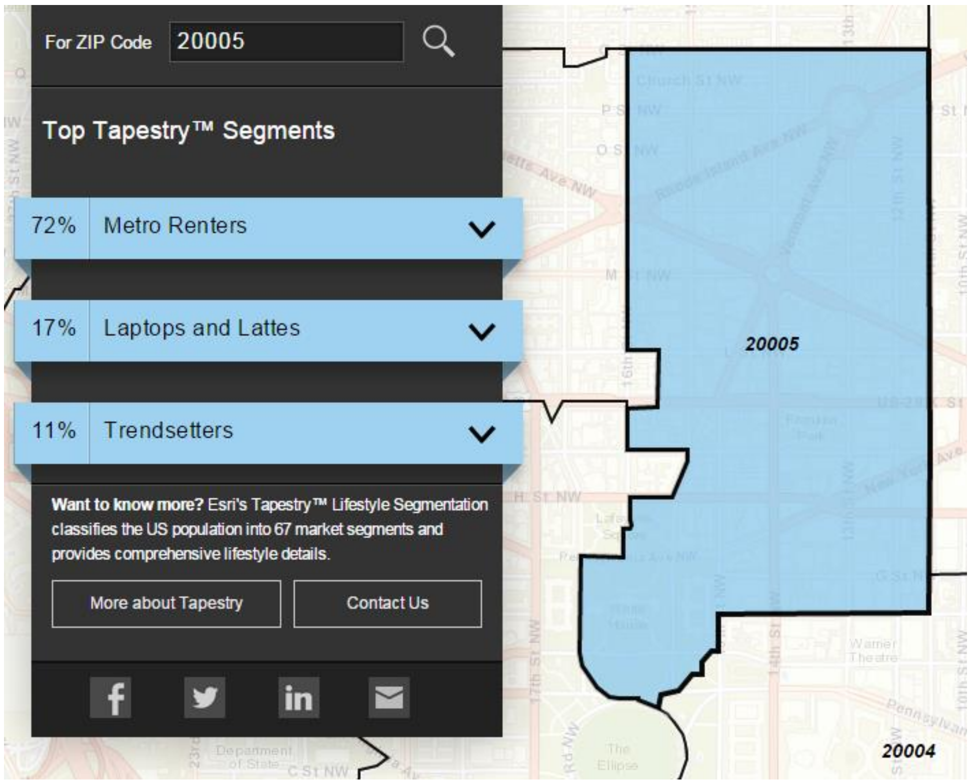


There are two main categories of machine learning: **supervised learning** and **unsupervised learning**.

**Unsupervised learning:**

- Extracting structure from data
- Example: segment grocery store shoppers into “clusters” that exhibit similar behaviors
- Goal is “representation”

Group US residential neighborhoods into 67 unique segments based on demographic and socioeconomic characteristics




Metro Renters:

Young, mobile, educated, or still in school, we live alone or with a roommate in rented apartments or condos in the center of the city. Long hours and hard work don't deter us; we're willing to take risks to get to the top of our professions… We buy groceries at Whole Foods and Trader Joe's and shop for clothes at Banana Republic, Nordstrom, and Gap. We practice yoga, go skiing, and attend Pilates sessions.

Source: http://www.esri.com/landing-pages/tapestry/

# UNSUPERVISED LEARNING



Common types of unsupervised learning:

- **Clustering:** group “similar” data points together
- **Dimensionality Reduction:** reduce the dimensionality of a dataset by extracting features that capture most of the variance in the data

# UNSUPERVISED LEARNING



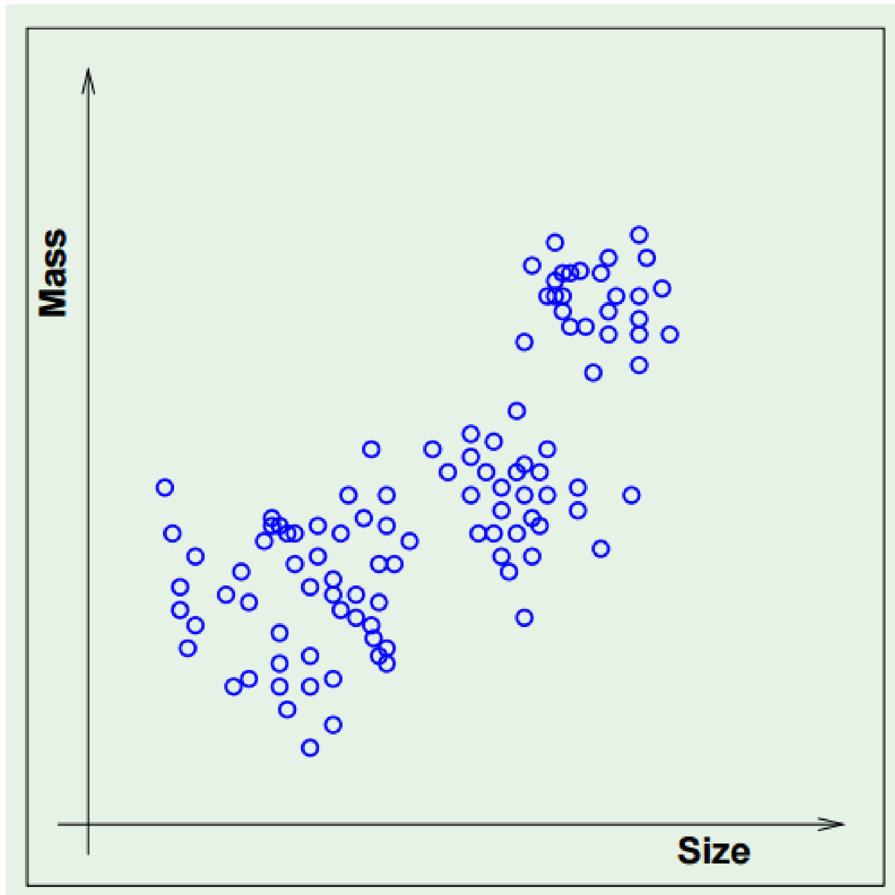
Unsupervised learning has some clear differences from supervised learning. With **unsupervised learning:**

- There is no clear objective
- There is no “right answer” (hard to tell how well you are doing)
- There is no response variable, just observations with features
- Labeled data is not required

## Unsupervised learning example: Coin clustering

- Observations: Coins
- Features: Size and mass
- Response: There isn't one (no hand-labeling required!)

1. Perform **unsupervised learning**
    - Cluster the coins based on "similarity"
    - You're done!

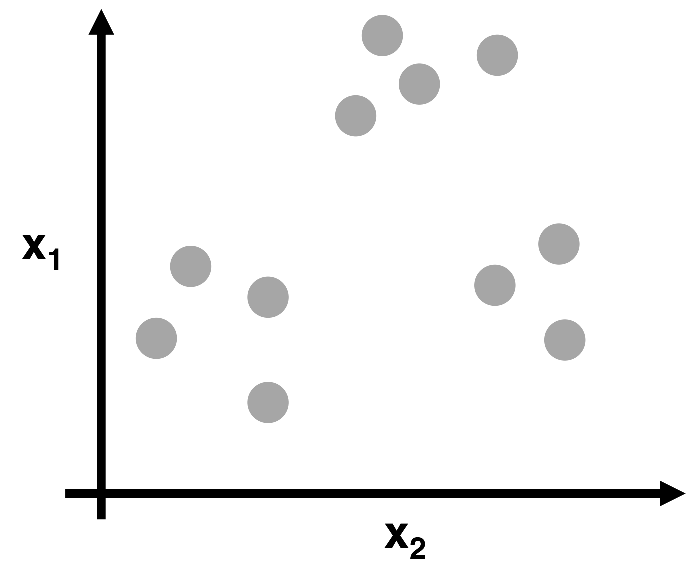

Mass
Size

# II. K-MEANS CLUSTERING

1) *choose k initial centroids (note that k is an input)*

2) *for each point:*
   - *find distance to each centroid*
   - *assign point to nearest centroid*

3) *recalculate centroid positions*
4) *repeat steps 2–3 until stopping criteria met*

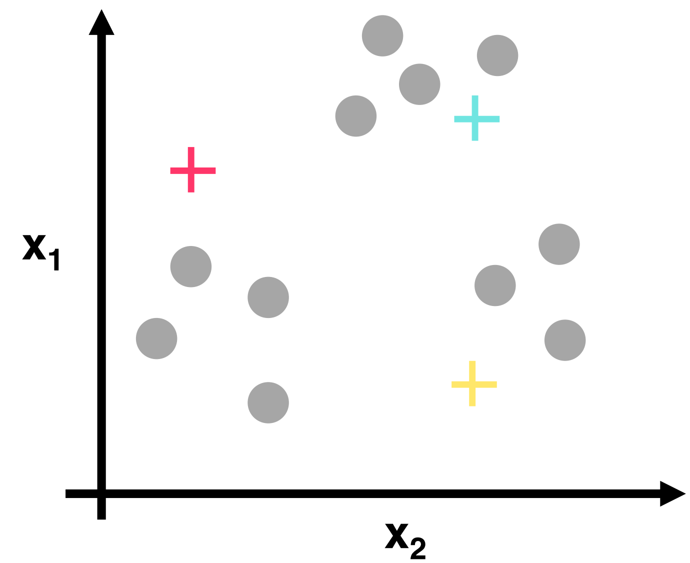


1) ***choose k initial centroids (note that k is an input)***

2) *for each point:*
   - *find distance to each centroid*
   - *assign point to nearest centroid*

3) *recalculate centroid positions*

4) *repeat steps 2-3 until stopping criteria met*

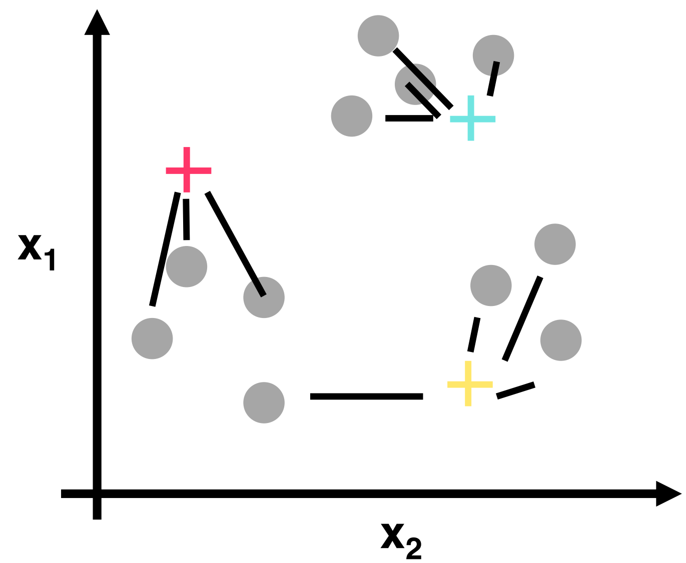


*1) choose k initial centroids (note that k is an input)*

***2) for each point:***

***- find distance to each centroid***

*- assign point to nearest centroid*




*3) recalculate centroid positions*

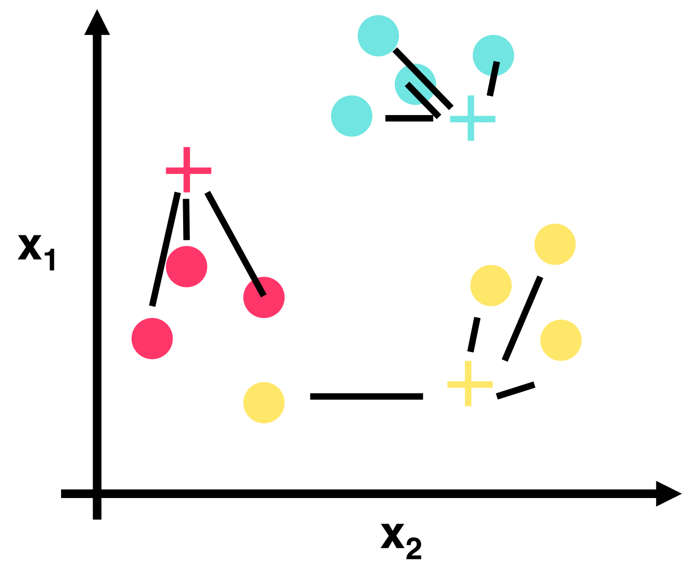
*4) repeat steps 2–3 until stopping criteria met*

1) *choose k initial centroids (note that k is an input)*

2) ***for each point:***
   - *find distance to each centroid*
   - ***assign point to nearest centroid***

3) *recalculate centroid positions*

4) *repeat steps 2–3 until stopping criteria met*

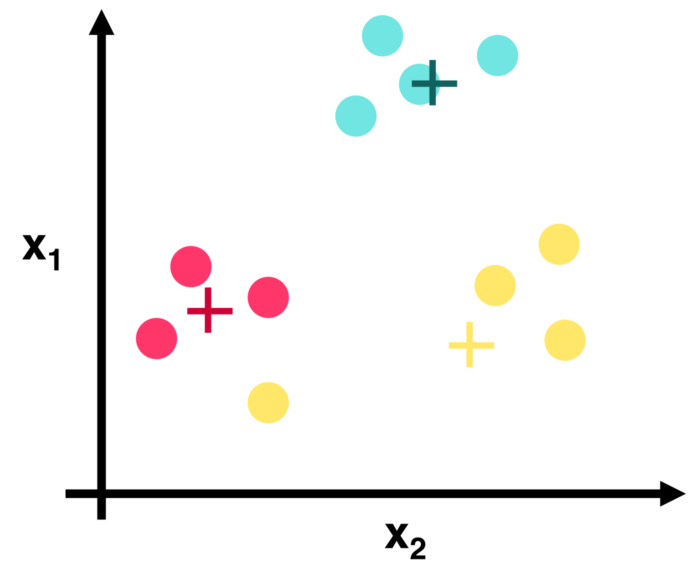


1) *choose k initial centroids (note that k is an input)*

2) *for each point:*
   - *find distance to each centroid*
   - *assign point to nearest centroid*

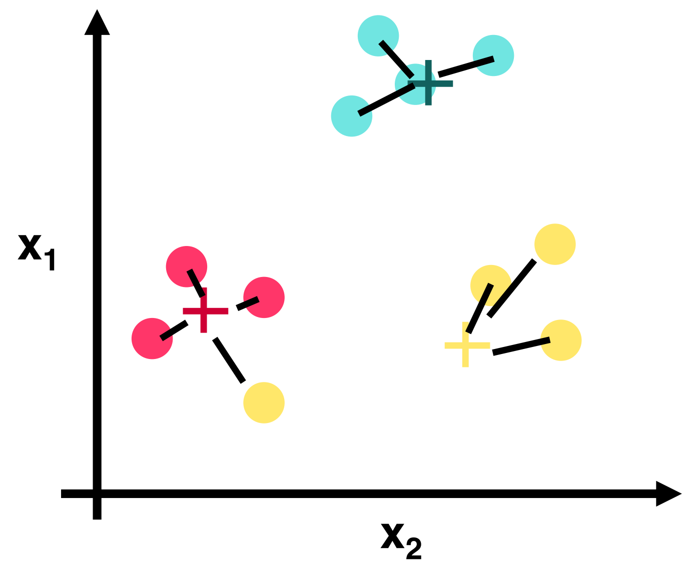
3) ***recalculate centroid positions***

4) *repeat steps 2–3 until stopping criteria met*



1) *choose k initial centroids (note that k is an input)*

2) *for each point:*
   - *find distance to each centroid*
   - *assign point to nearest centroid*

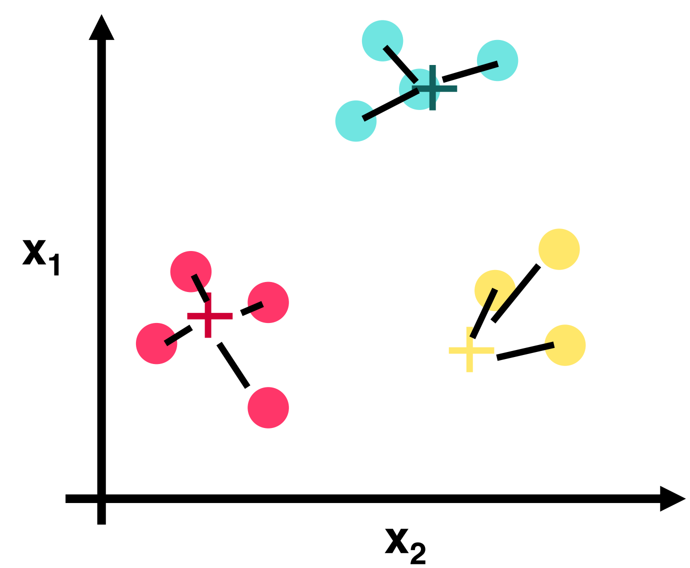



3) *recalculate centroid positions*

4) ***repeat steps 2-3 until stopping criteria met***

1) *choose k initial centroids (note that k is an input)*

2) *for each point:*
   - *find distance to each centroid*
   - *assign point to nearest centroid*




3) *recalculate centroid positions*

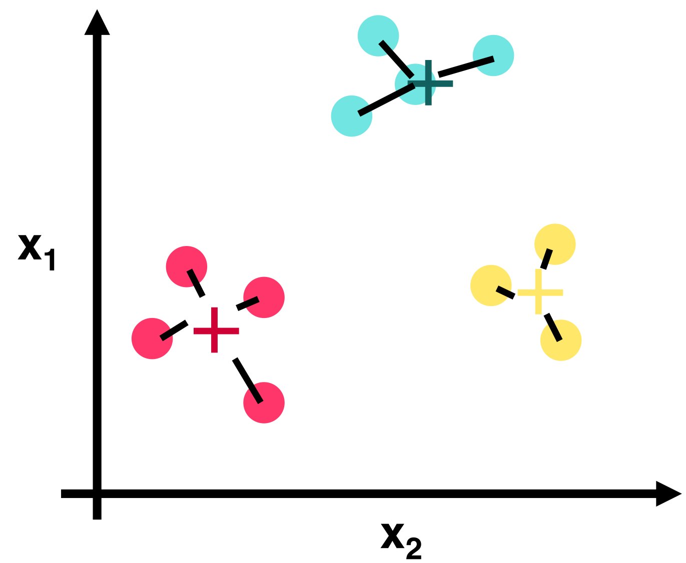
4) ***repeat steps 2-3 until stopping criteria met***

*1) choose k initial centroids (note that k is an input)*

*2) for each point:*
- *find distance to each centroid*
- *assign point to nearest centroid*

*3) recalculate centroid positions*

***4) repeat steps 2–3 until stopping criteria met***

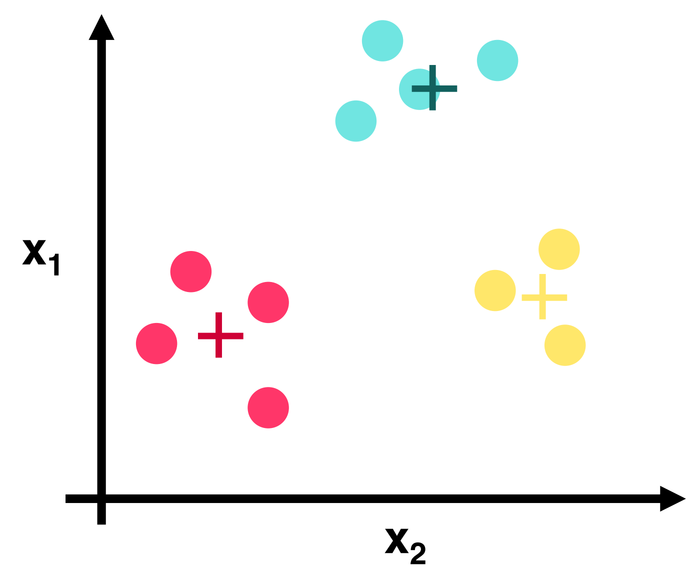


1) *choose k initial centroids (note that k is an input)*

2) *for each point:*
   - *find distance to each centroid*
   - *assign point to nearest centroid*

3) *recalculate centroid positions*

4) ***repeat steps 2-3 until stopping criteria met***



## Questions:

- What will happen if no actual clusters exist?
- How should you choose K?
- How should you choose the initial centroid positions?
- When should the algorithm stop?
- When might it produce poor results?

**What will happen if no actual clusters exist?**

- It will still find clusters!
  - Visualization: I’ll Choose, Uniform Points

## How should you choose K?

- It will find the number of clusters specified
  - Visualization: I’ll Choose, Gaussian Mixture, K=2/3/4
- Try different values for K and evaluate the "performance"

## How should you choose the initial centroid positions?

- Randomly
  - Doesn’t tend to work well
  - Visualization: Randomly, Gaussian Mixture, K=3
- Farthest point
  - Visualization: Farthest Point, Packed Circles, K=7
- K-means++
  - Similar to farthest point, but adds some randomness
  - Used by default in scikit-learn
- In all cases: Run it several times and pick the best result to avoid local minima

## When should the algorithm stop?

- Tends to converge quickly
- Set stopping criteria:
  - Maximum number of iterations
  - Once centroids move less than a threshold amount
  - Once fewer points than a threshold amount change clusters
  - Visualization: Randomly, Pimpled Smiley, K=5

**When might it produce poor results?**

- Data with varying shapes
  - Visualization: I’ll Choose, Smiley Face, K=4
- Data with different scales
- Still the most popular clustering algorithm, used for a wide range of applications

# III. DBSCAN CLUSTERING

# DBSCAN CLUSTERING 

**DBSCAN:** Density-Based Spatial Clustering of Applications with Noise

For DBSCAN, clusters are areas of **high density** separated by areas of **low density**.

## DBSCAN Algorithm:

1. Choose “epsilon” and “min_samples”
2. Pick an arbitrary point, and check if there are at least “min_samples” points within the distance “epsilon”
   - If yes, add those points to the cluster and check each of the new points
   - If no, choose another arbitrary point to start a new cluster
3. Stop once all points have been checked

Visualization: Uniform Points

**DBSCAN Advantages:**

- Clusters can be any shape or size
- No need to choose the number of clusters

**DBSCAN Disadvantages:**

- More parameters to tune
- Doesn’t work with clusters of varying density

**Note:** Not every point is necessarily assigned to a cluster!