Fundamentals of Artificial Intelligence

VICTORIA UNIVERSITY OF WELLINGTON
TE HERENGA WAKA

COMP307/AIML420

Clustering

Dr. Heitor Murilo Gomes
heitor.gomes@vuw.ac.nz
http://www.heitorgomes.com

Outline

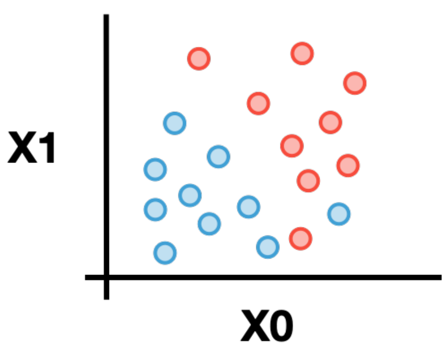
1. Unsupervised x Supervised learning
2. Clustering
3. K-means
4. DBSCAN
5. Elbow method for k-means

Unsupervised x Supervised

Supervised Learning

- Train on labeled data (classification: input **X** and class label **y**)
- Fit a model to make predictions for previously unseen data

- Example: Binary classification problem



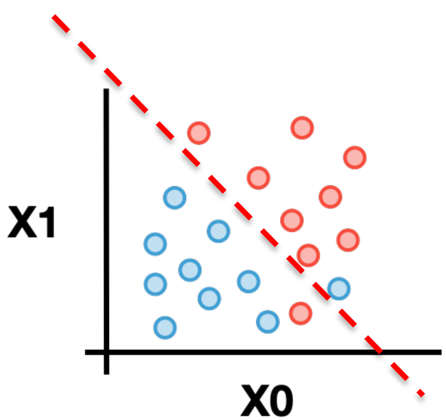
- Two classes
- Two features (X0 and X1)

Unsupervised x Supervised

Supervised Learning

- Train on labeled data (classification: input **X** and class label **y**)
- Fit a model to make predictions for previously unseen data

- Example: Binary classification problem



- Two classes
- Two features (X0 and X1)

Learn a decision boundary that adequately separates the training data

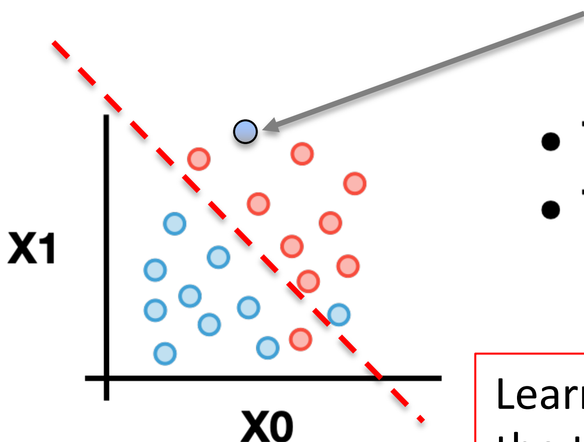
Unsupervised x Supervised

Supervised Learning

- Train on labeled data (classification: input **X** and class label **y**)
- Fit a model to make predictions for previously unseen data

- Example: Binary classification problem

Predict the class label of new instances



- Two classes
- Two features (X0 and X1)

Learn a decision boundary that adequately separates the training data

Unsupervised x Supervised

Unsupervised Learning*

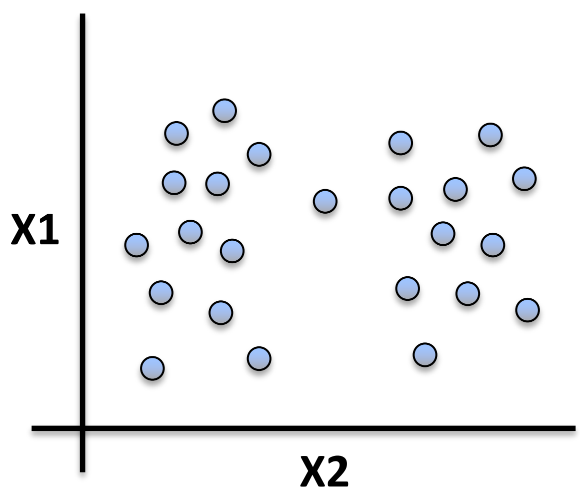
- There is no labeled data (only the input **X** is available)
- The goal is to explore the **structure of the data** and **discover patterns** or relationships that may exist among the features

** We are going to focus on clustering, there is more to unsupervised learning than clustering*

Unsupervised x Supervised

Unsupervised Learning*

- There is no labeled data (only the input **X** is available)
- The goal is to explore the **structure of the data** and **discover patterns** or relationships that may exist among the features



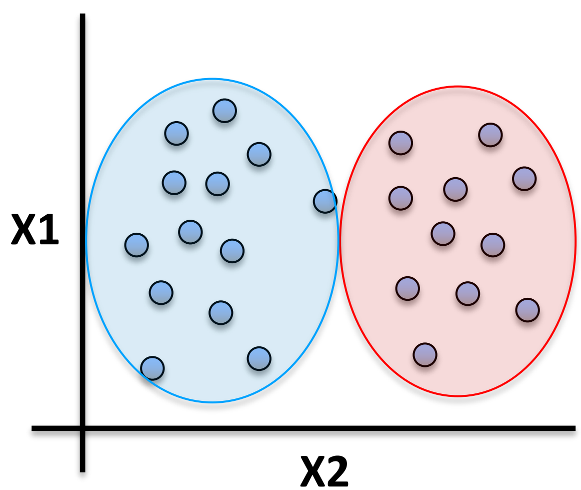
** We are going to focus on clustering, there is more to unsupervised learning than clustering*

Unsupervised x Supervised

Unsupervised Learning*

- There is <u>no labeled data</u> (only the input **X** is available)
- The goal is to explore the **structure of the data** and **discover patterns** or relationships that may exist among the features

We can split this data in two clusters like this…



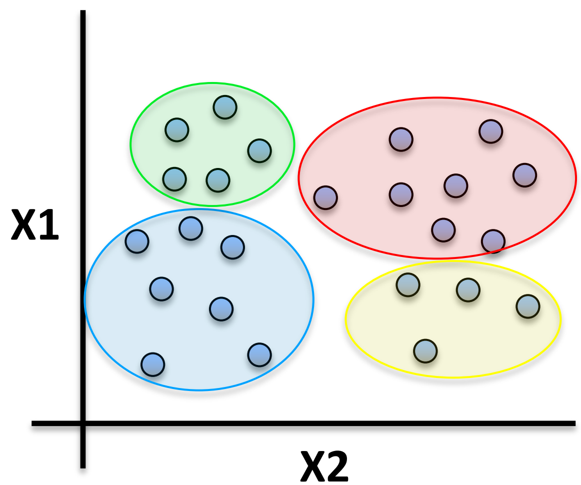
* *We are going to focus on clustering, there is more to unsupervised learning than clustering*

Unsupervised x Supervised

Unsupervised Learning*

- There is <u>no labeled data</u> (only the input **X** is available)
- The goal is to explore the **structure of the data** and **discover patterns** or relationships that may exist among the features

Or 4 like this…



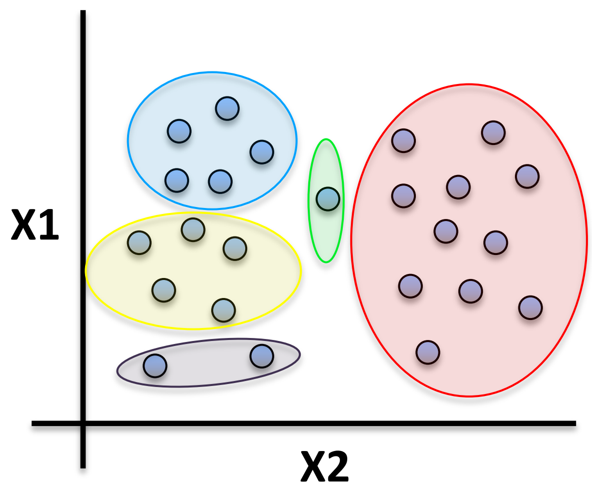
** We are going to focus on clustering, there is more to unsupervised learning than clustering*

Unsupervised x Supervised

Unsupervised Learning*

- There is no labeled data (only the input **X** is available)
- The goal is to explore the **structure of the data** and **discover patterns** or relationships that may exist among the features

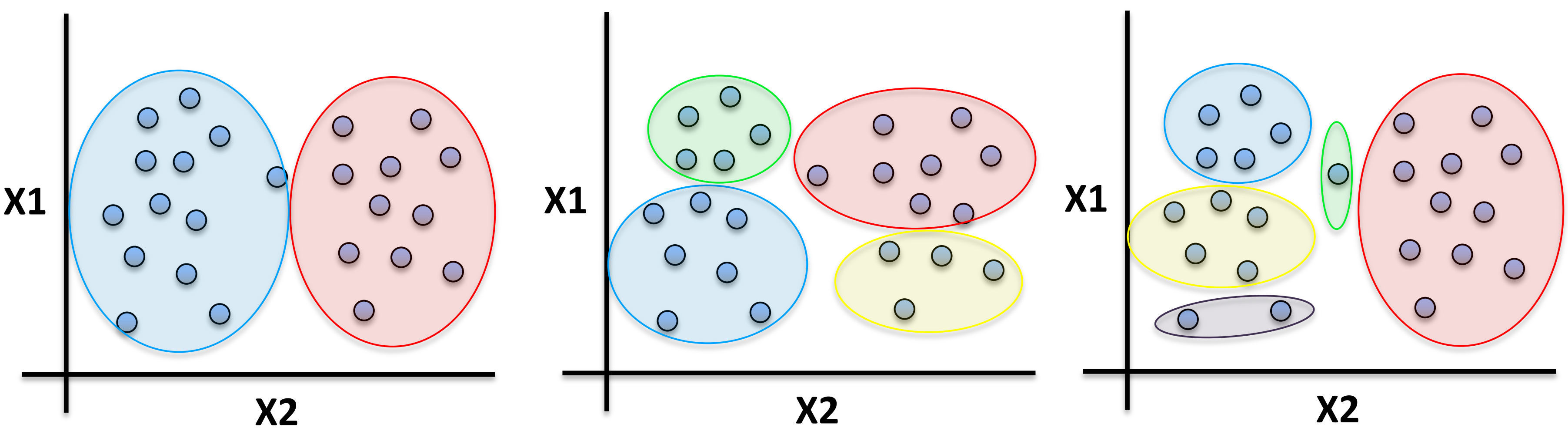
Why not 5 clusters?



** We are going to focus on clustering, there is more to unsupervised learning than clustering*

Clustering

- There are **multiple ways** of clustering the data
- There is not necessarily a **"correct" clustering**, it depends on our goals and the evaluation metric we are using
- For example, we may want a **specific number of clusters** or we may be interested in grouping clusters of ***non-spherical shape***



K-means

- **Centroid-based*** clustering algorithm
- Sensitive to the **centroids initialization**
- The hyperparameter **K determines the number of clusters**
- Can generate **spherical clusters**

K-means pseudo-code

1. **Initialize** K cluster centroids randomly

2. Repeat until **convergence**:

  a. **Assign** each data point to the closest centroid

  b. **Update** the centroid of each cluster as the mean of the data points assigned to it

3. Return the K cluster centroids and the cluster assignments of each data point

** A centroid is a point that represents the arithmetic mean of all the points in a cluster of points*

K-means example

K-means

1. Initialize K cluster centroids randomly

2. Repeat until convergence:

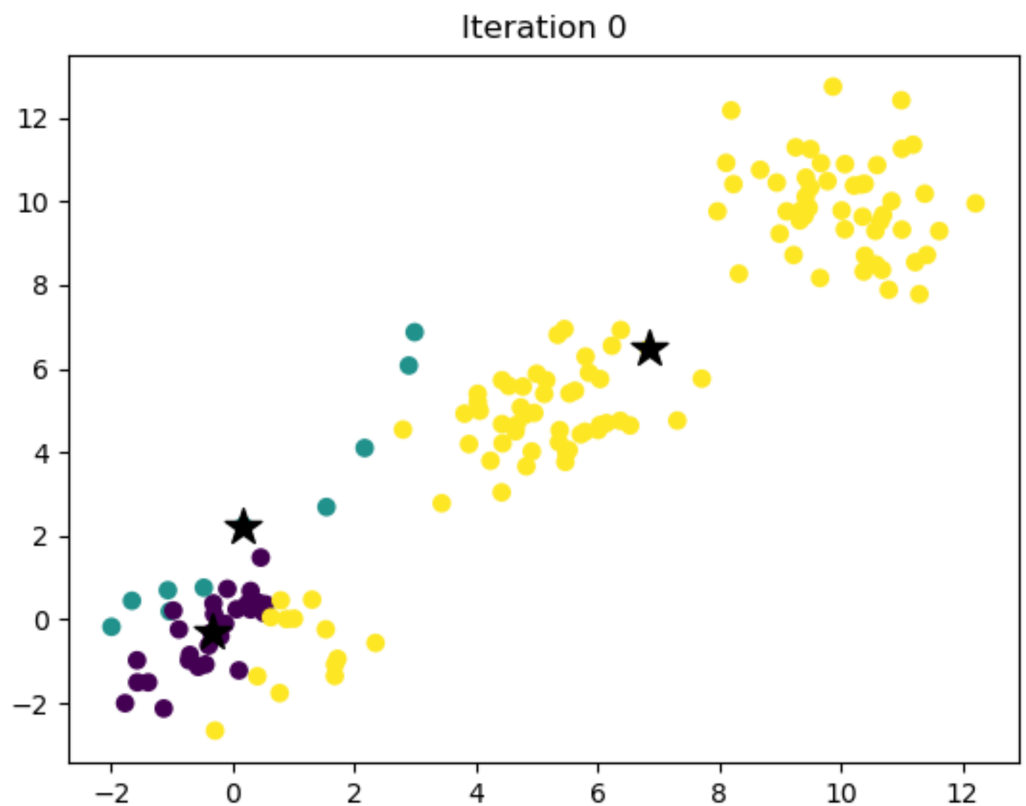
a. Assign each data point to the closest centroid

b. Update the centroid of each cluster

as the mean of the data points assigned to it

3. Return the K cluster […]

K=3



K-means example

K-means

1. Initialize K cluster centroids randomly

2. Repeat until convergence:

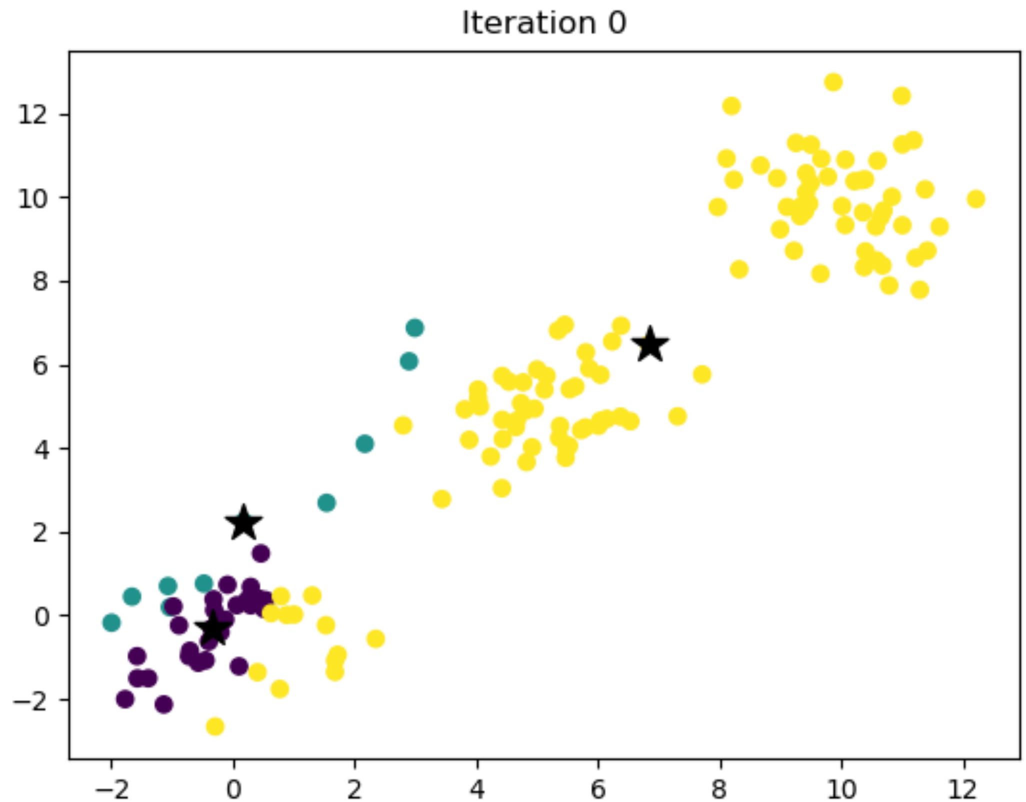
a. Assign each data point to the closest centroid

b. Update the centroid of each cluster

as the mean of the data points assigned to it

3. Return the K cluster […]

K=3



K-means example

K-means

1. Initialize K cluster centroids randomly

2. Repeat until convergence:

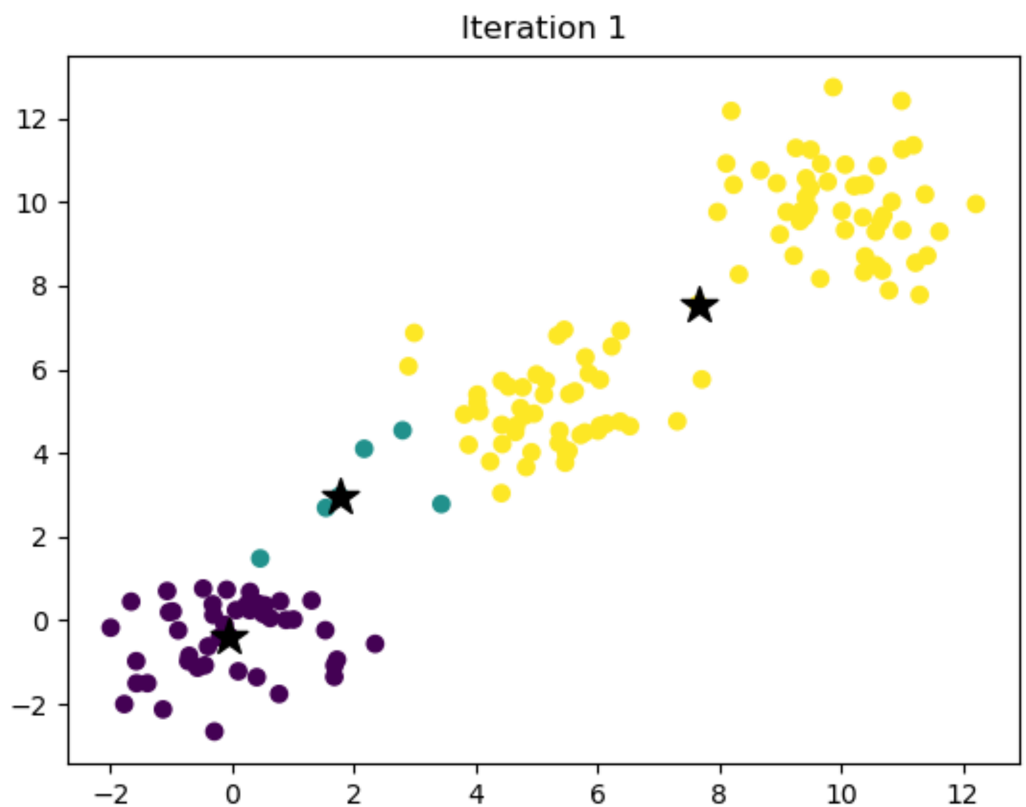
a. Assign each data point to the closest centroid

b. Update the centroid of each cluster

as the mean of the data points assigned to it

3. Return the K cluster […]

K=3



K-means example

K-means

1. Initialize K cluster centroids randomly

2. Repeat until convergence:

a. Assign each data point to the closest centroid

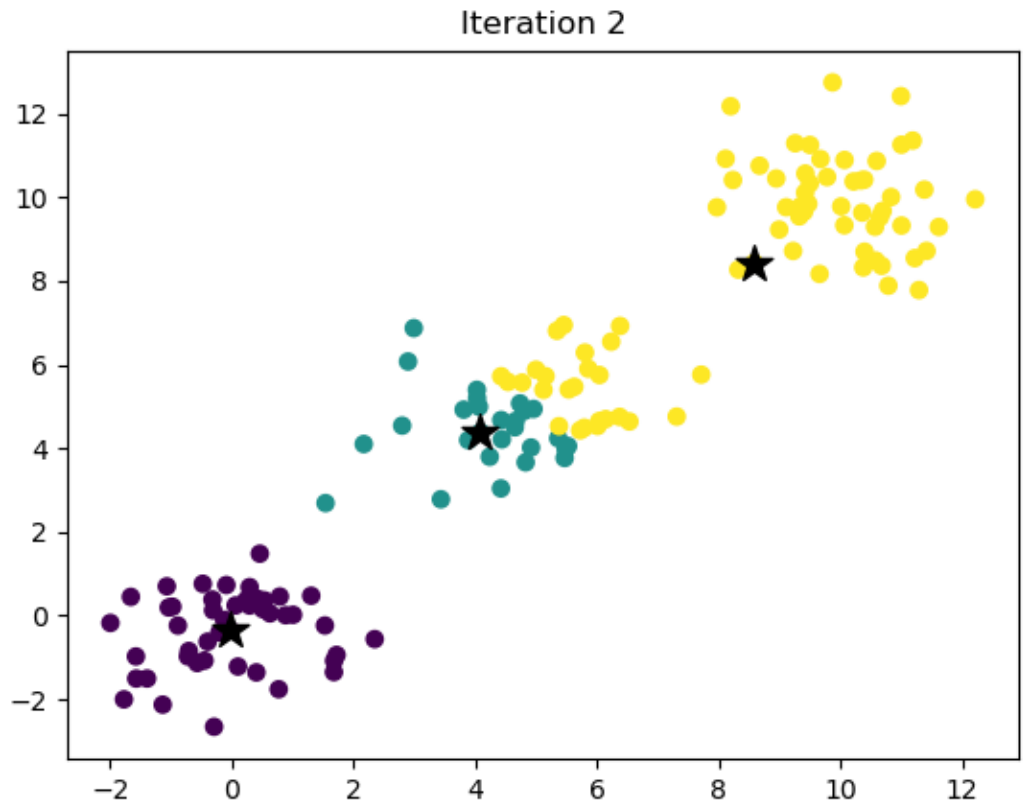
b. Update the centroid of each cluster

as the mean of the data points assigned to it

3. Return the K cluster […]

K=3

Convergence criteria?



K-means example

K-means

1. Initialize K cluster centroids randomly

2. Repeat until convergence:

a. Assign each data point to the closest centroid

b. Update the centroid of each cluster

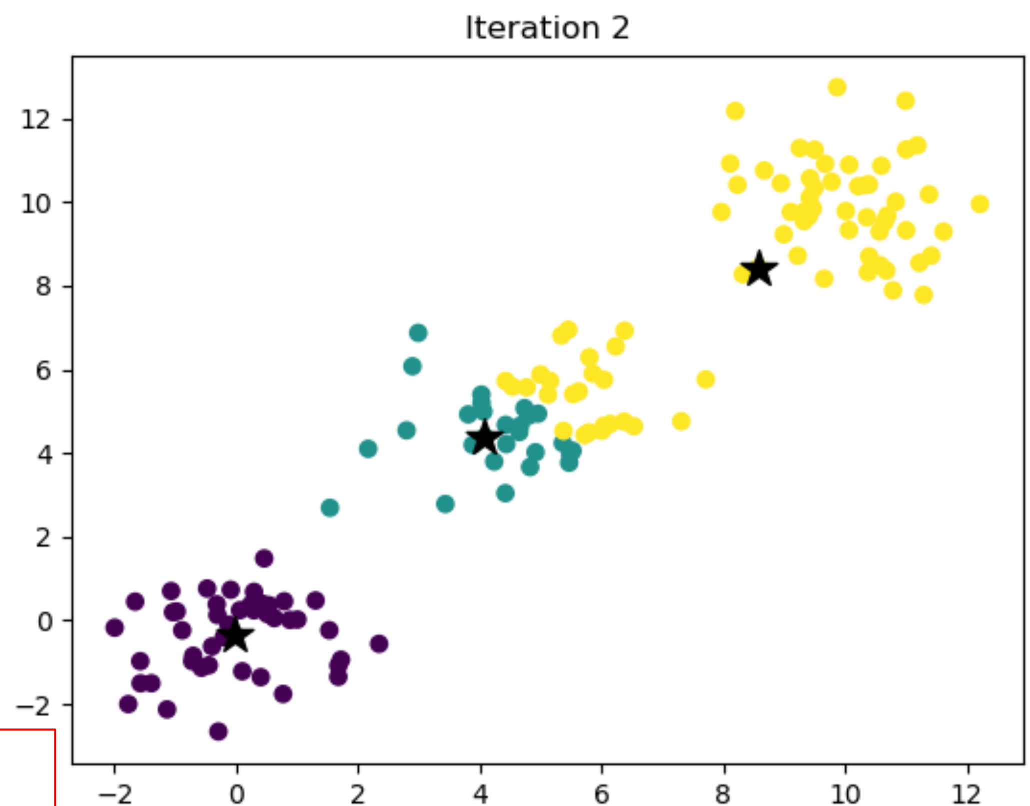
as the mean of the data points assigned to it

3. Return the K cluster […]

K=3

Convergence criteria?

- Max number of iterations
- Min change in centroids
- Min change in cluster assignments



K-means example

K-means

1. Initialize K cluster centroids randomly

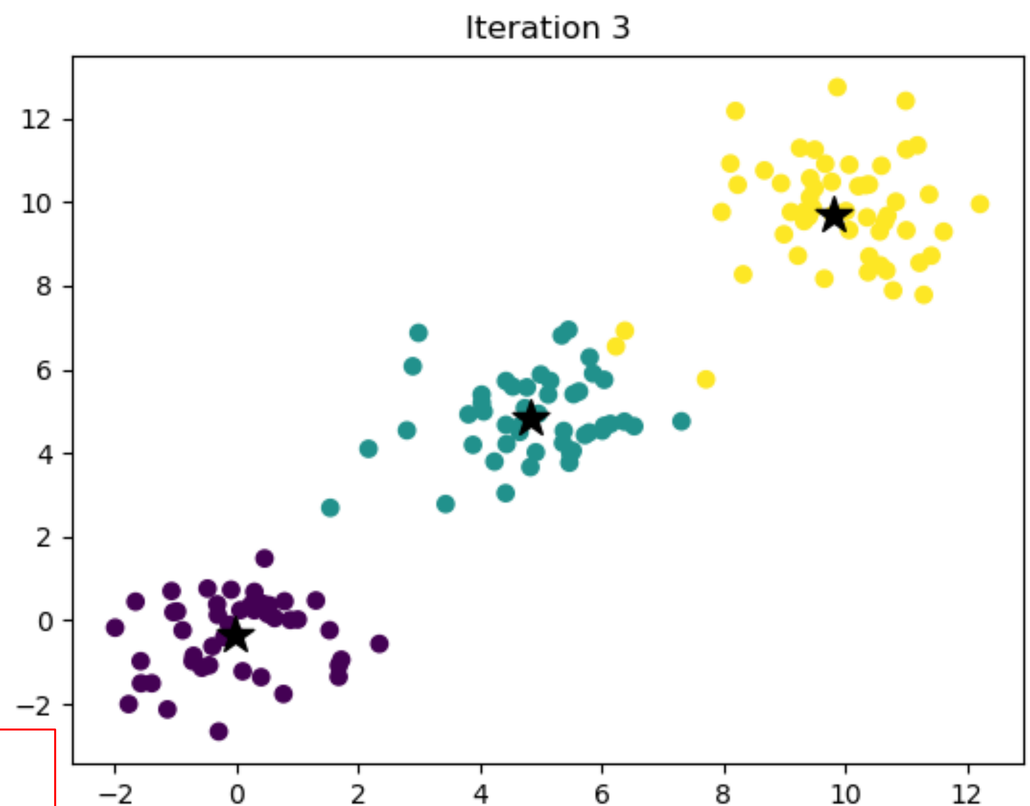
2. Repeat until convergence:

  a. Assign each data point to the closest centroid

  b. Update the centroid of each cluster

as the mean of the data points assigned to it

3. Return the K cluster […]



K=3

Convergence criteria?

- **Max number of iterations**
- Min change in centroids
- Min change in cluster assignments

K-means example

K-means

1. Initialize K cluster centroids randomly

2. Repeat until convergence:

a. Assign each data point to the closest centroid

b. Update the centroid of each cluster

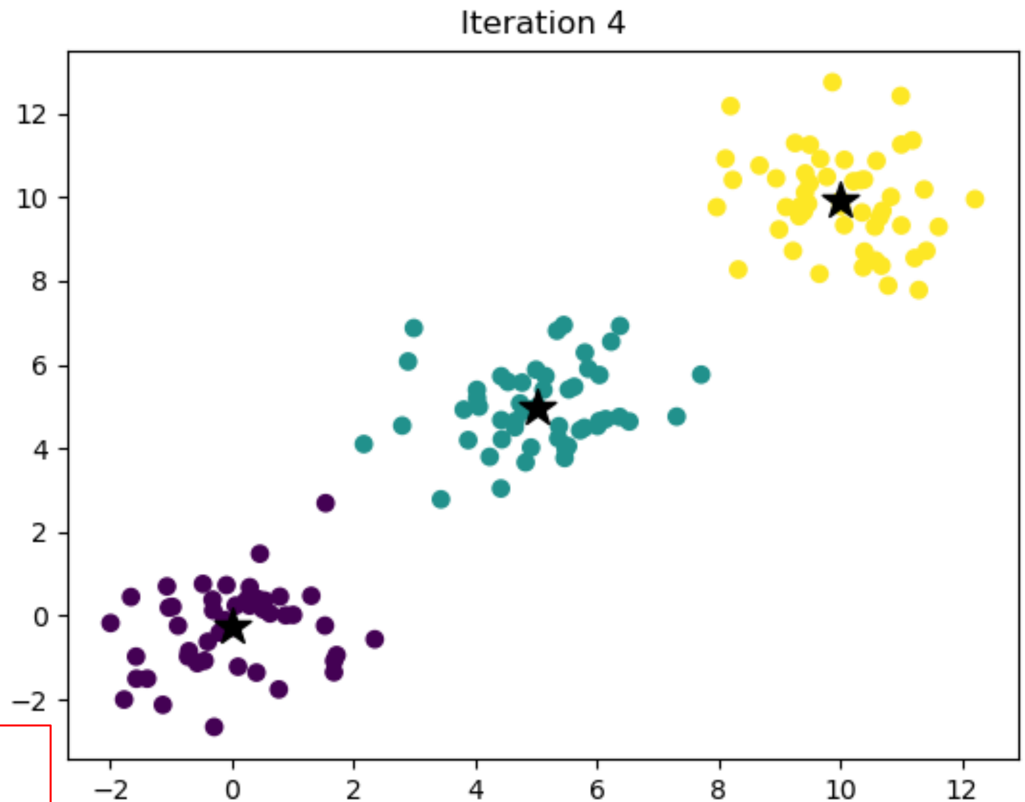
as the mean of the data points assigned to it

3. Return the K cluster […]

K=3

Convergence criteria?

- **Max number of iterations**
- Min change in centroids
- Min change in cluster assignments



K-means example

K-means

1. Initialize K cluster centroids randomly

2. Repeat until convergence:

a. Assign each data point to the closest centroid

b. Update the centroid of each cluster

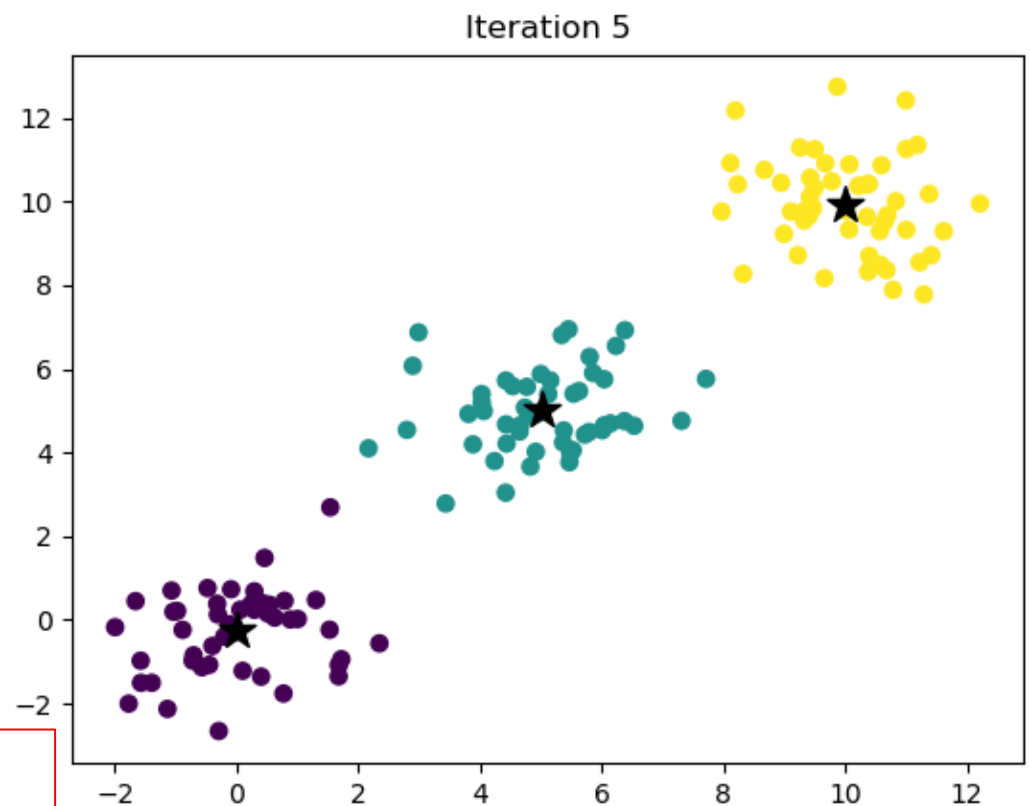
as the mean of the data points assigned to it

3. Return the K cluster […]

K=3

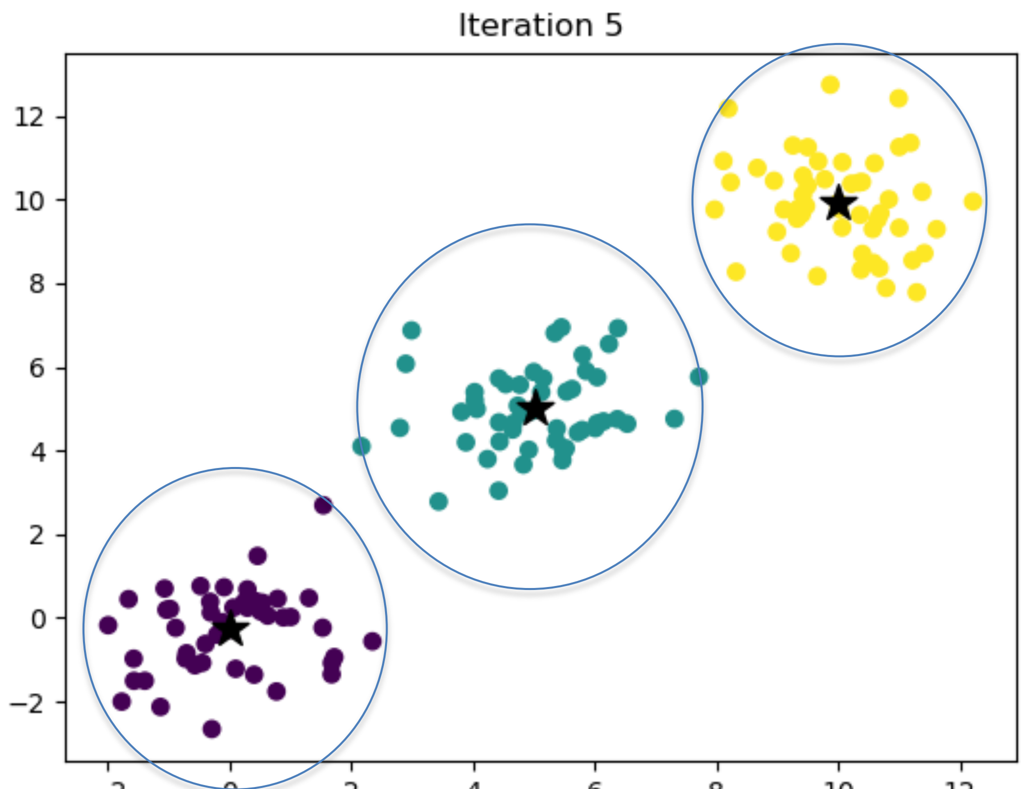
Convergence criteria?

- **Max number of iterations: 6**
- Min change in centroids
- Min change in cluster assignments



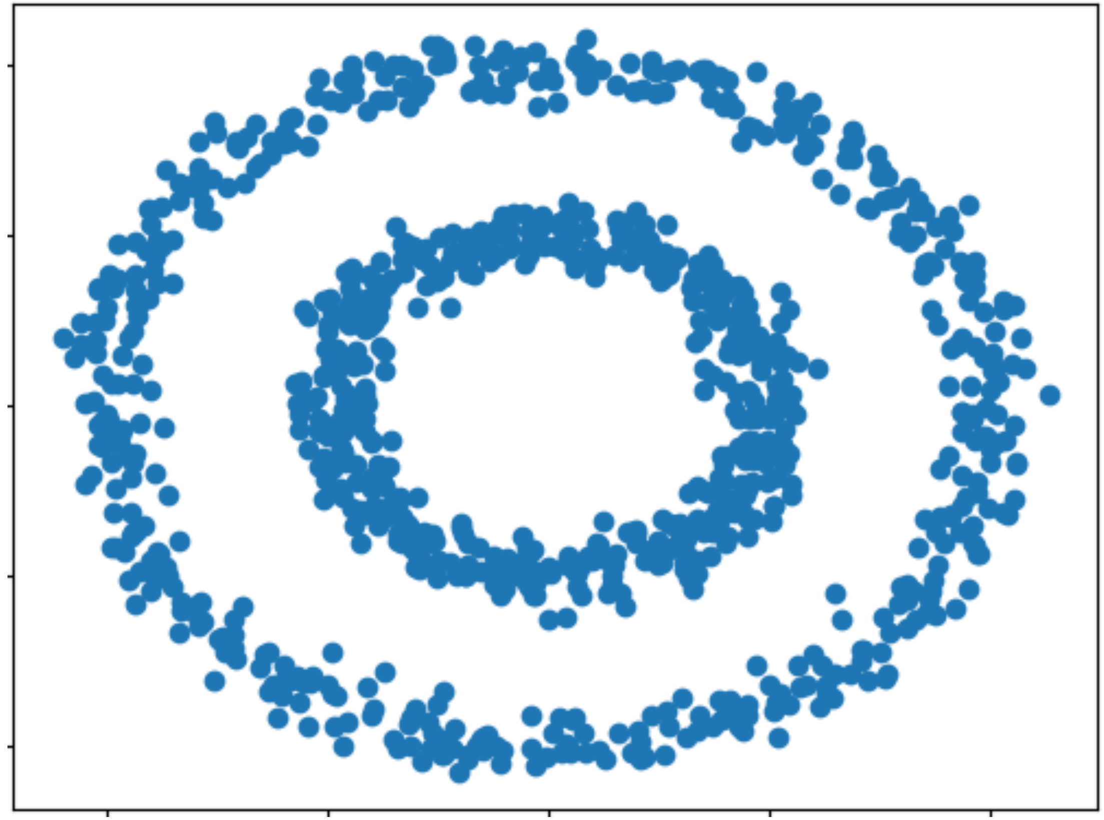
K-means example

- **Spherical clusters**



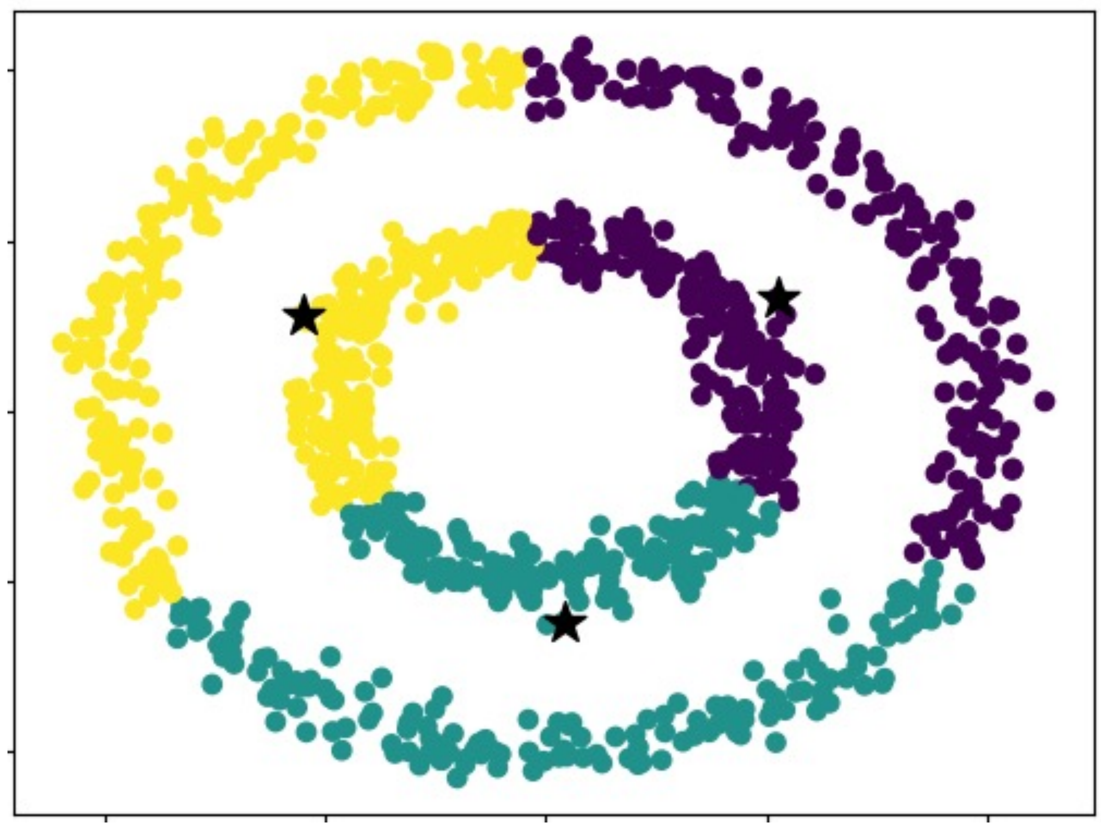
Another example

- **What about now?**



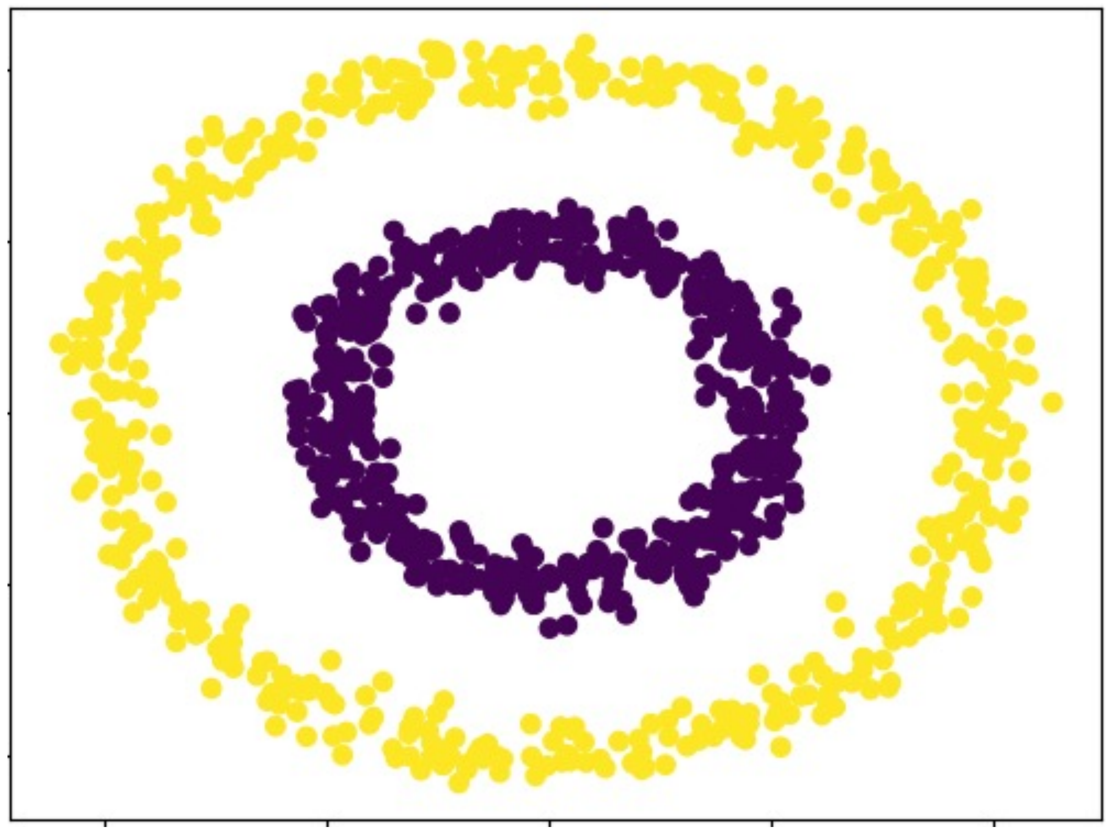
Another example

- **What about now?**

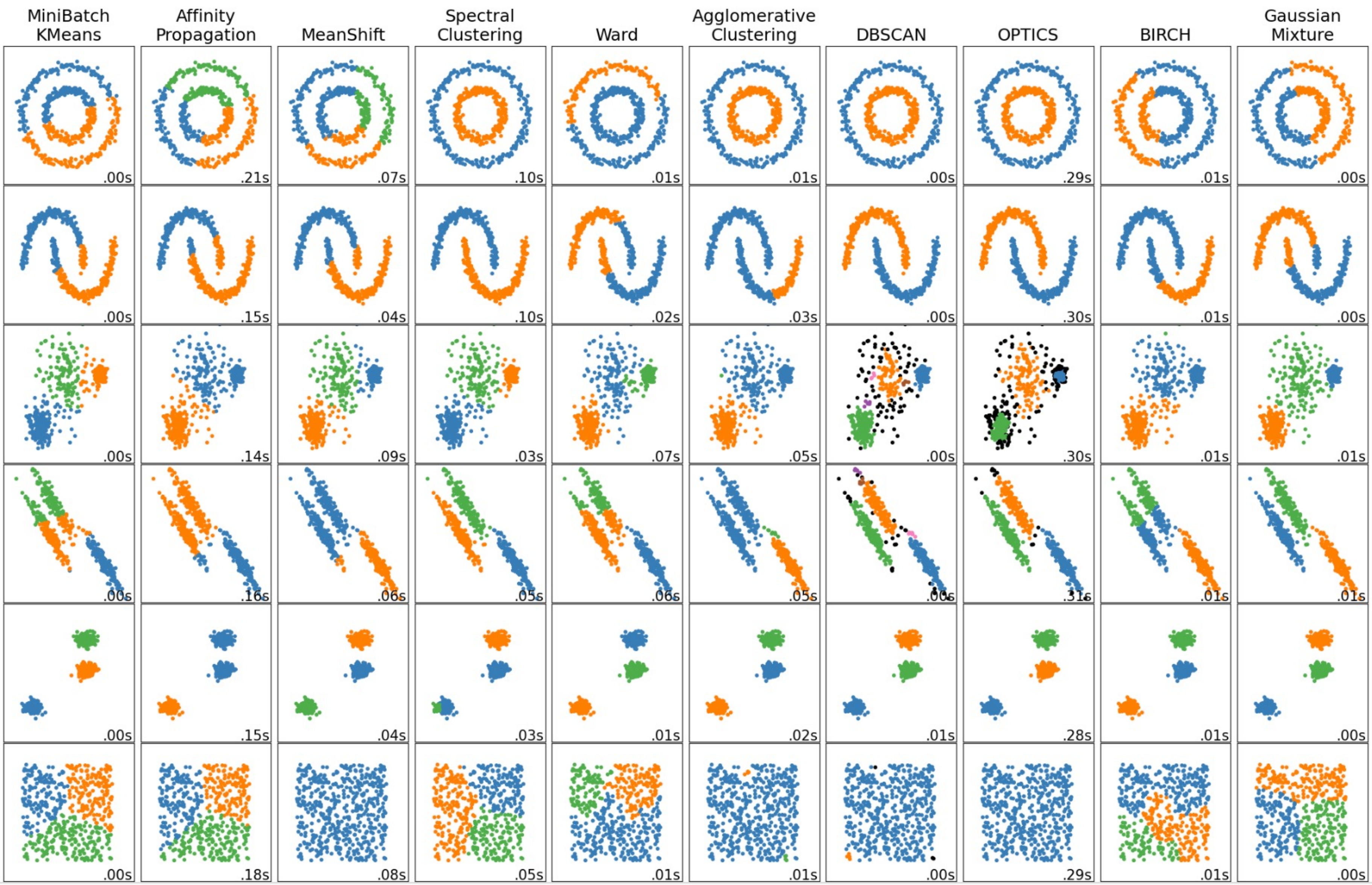


Another example

- **Intuitively, we would like something like this…**



Then we need some other algorithm…



Source: https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

DBSCAN*

- **Density-based** clustering algorithm
- Can discover clusters of varying shapes, sizes and densities
- Hyperparameters:
  - **min_points**.** The minimum number of points required to form a dense region
  - **ε (eps).** The radius of the neighborhood around each point

DBSCAN pseudo-code

1. Find the points in the **eps** neighborhood of every point, and identify the **core points** with more than **min_points** neighbors.

2. Find the **connected components** of core points on the neighbor graph, ignoring all non-core points.

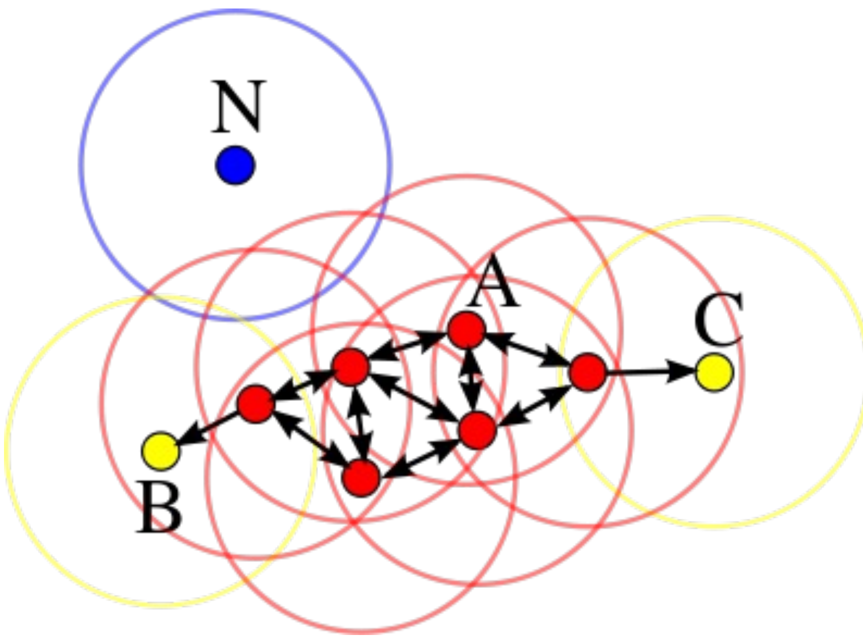
3. Assign each non-core point to a nearby cluster if the cluster is an **eps** neighbor, otherwise assign it to **noise**.

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

*** min_points or min_samples, both denominations are common*

DBSCAN intuition

- **Core points**, **eps** and **min_points**



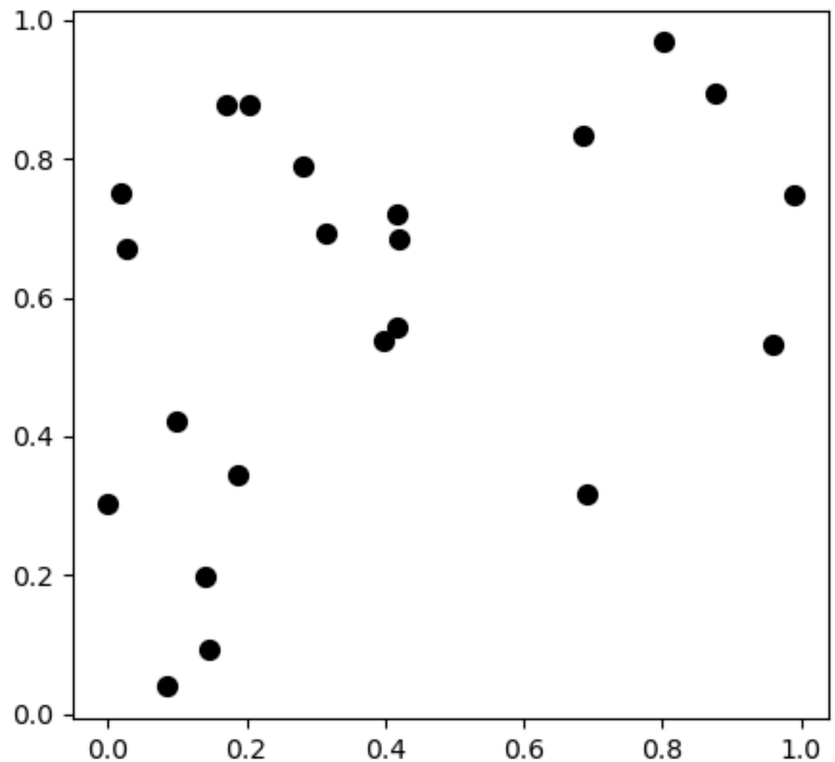
"In this diagram, minPts = 4. Point A and the other red points are core points, because the area surrounding these points in an ε radius contain at least 4 points (including the point itself). Because they are all reachable from one another, they form a single cluster. Points B and C are not core points, but are reachable from A (via other core points) and thus belong to the cluster as well. Point N is a noise point that is neither a core point nor directly-reachable." [1]

[1] Source of the image and caption: https://en.wikipedia.org/wiki/DBSCAN

DBSCAN intuition

eps = 0.1 and **min_points** = 3

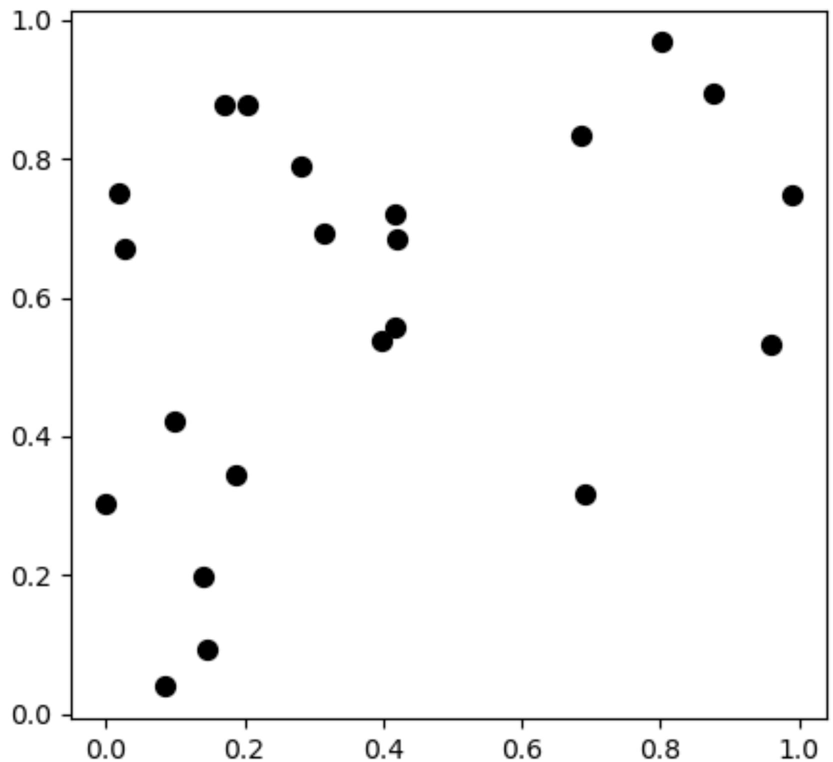
Let's identify the core points…



DBSCAN intuition

eps = 0.1 and **min_points** = 3

Let's identify the core points…



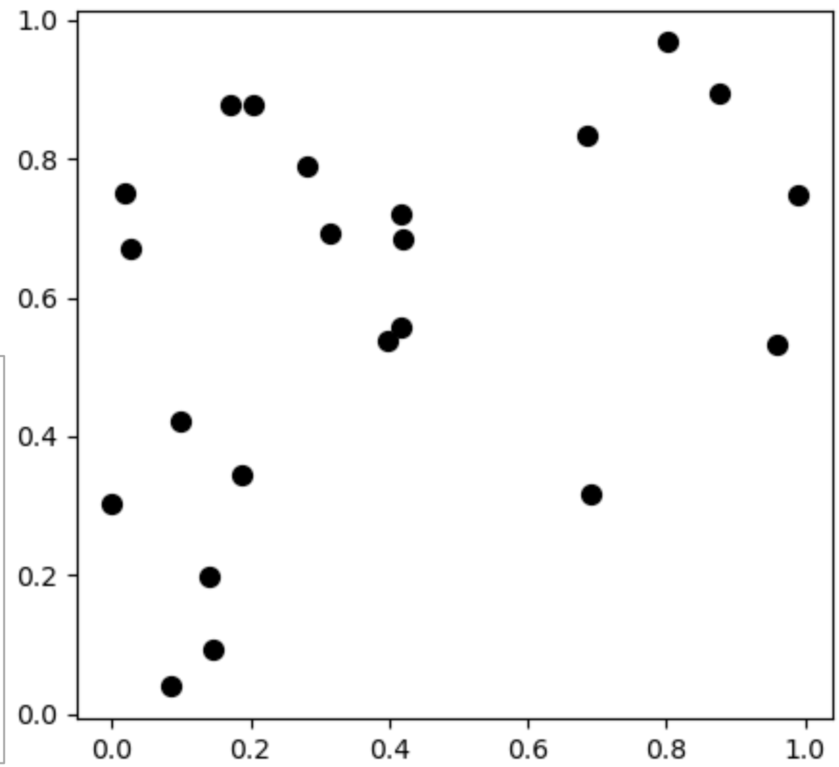
For each point we check if there are min_points in its neighborhood

DBSCAN intuition

eps = 0.1 and **min_points** = 3

Let's identify the core points…

For each point we check if there are min_points in its neighborhood



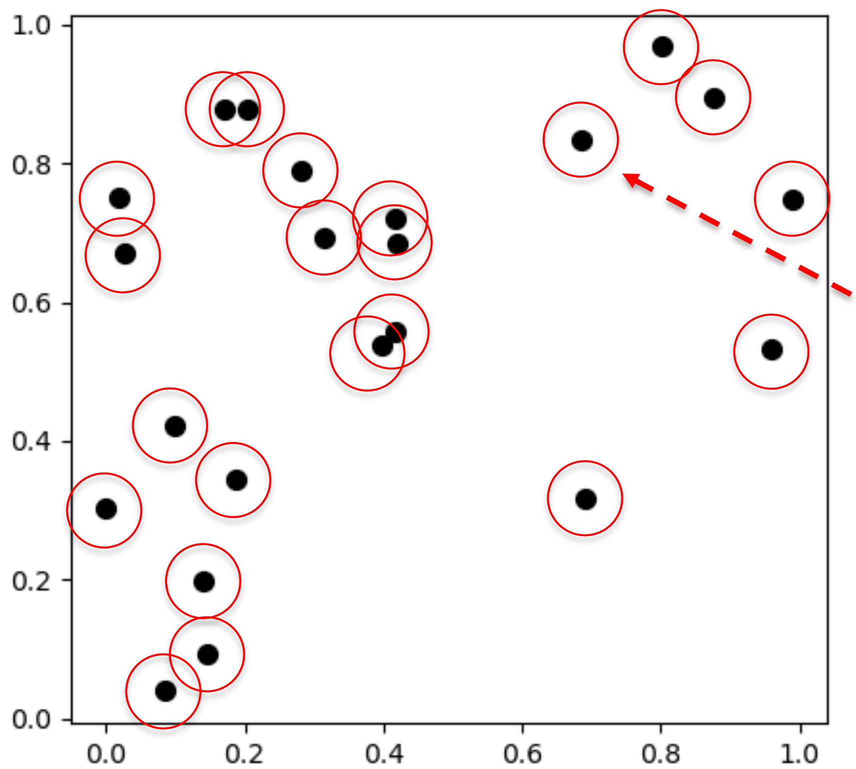
What **eps=0.1** mean?

DBSCAN intuition

eps = 0.1 and **min_points** = 3

Let's identify the core points...

For each point we check if there are min_points in its neighborhood



What **eps=0.1** mean?

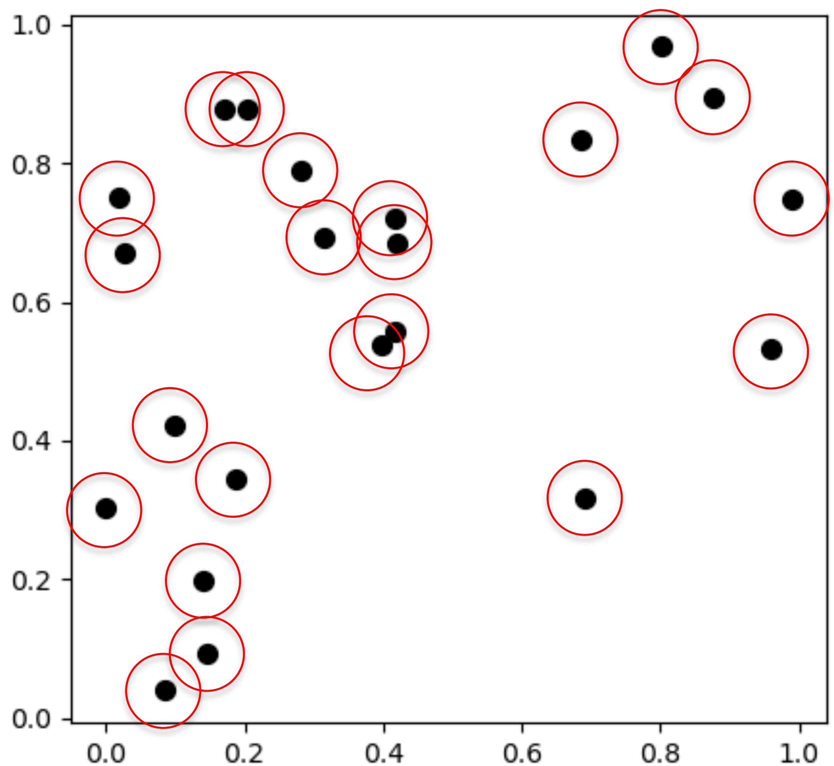
The radius of the neighborhood

DBSCAN intuition

eps = 0.1 and **min_points** = 3

Let's identify the core points…

For each point we check if there are min_points in its neighborhood



What **eps=0.1** mean?

The radius of the neighborhood

Turns out eps=0.1 and min_points=3 does not allow us to find any core points…

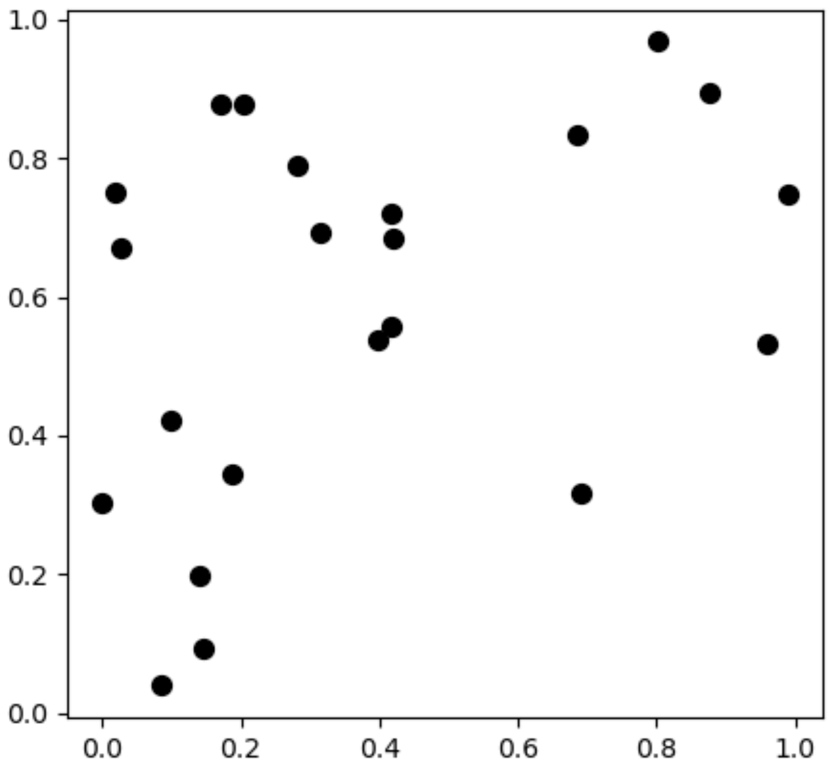
DBSCAN intuition

eps = 0.1 and **min_points** = **4**

What if we increase min_points to **4**!?

DBSCAN intuition

eps = 0.1 and **min_points** = **4**



Nothing happens ☹

We are being more **restrictive** by increasing min_points

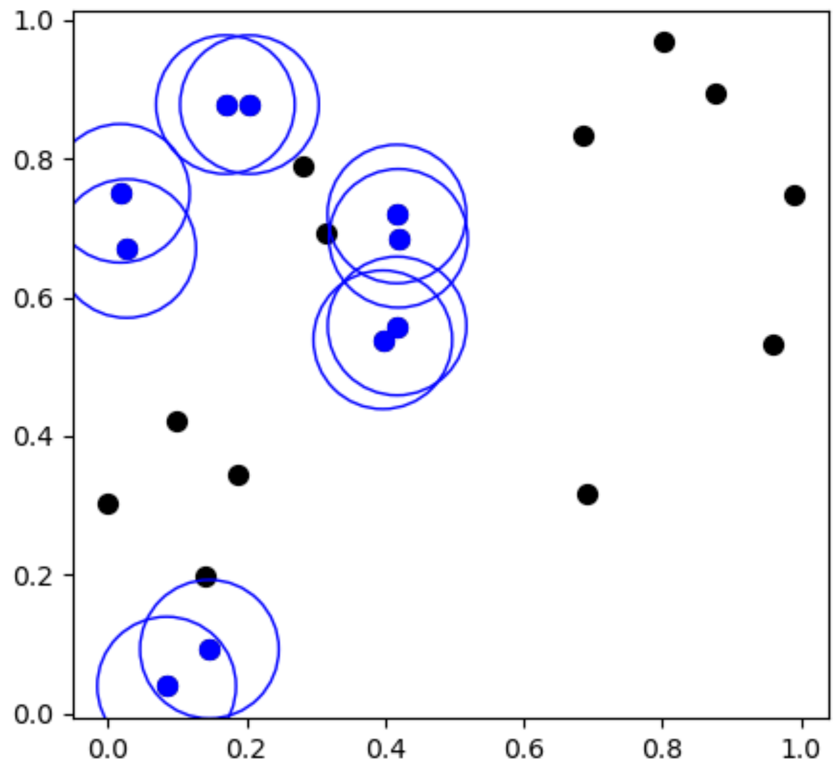
DBSCAN intuition

eps = 0.1 and **min_points** = **2**

What if we decrease min_points to **2**!?

DBSCAN intuition

eps = 0.1 and **min_points** = **2**



Success!!!

Success!!!

Success!!!

Success!!!

DBSCAN intuition

All good with **min_points** (hopefully),
but what about **eps**?

DBSCAN intuition

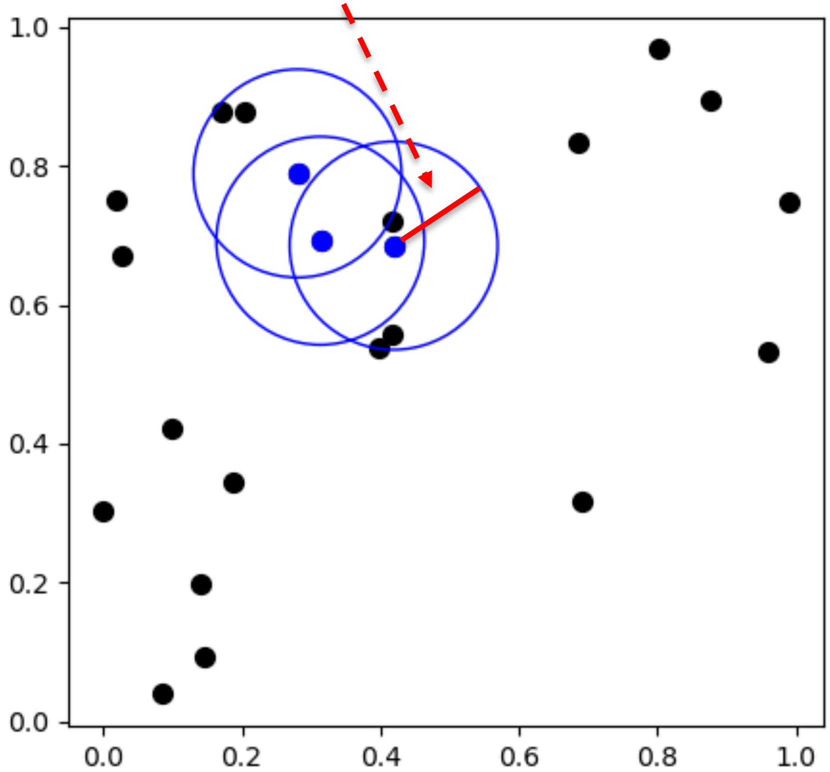
eps = **0.15** and **min_points** = **4**

All good with **min_points** (hopefully), but what about **eps**?

What if we keep min_points=4 and just increase eps to 0.15?

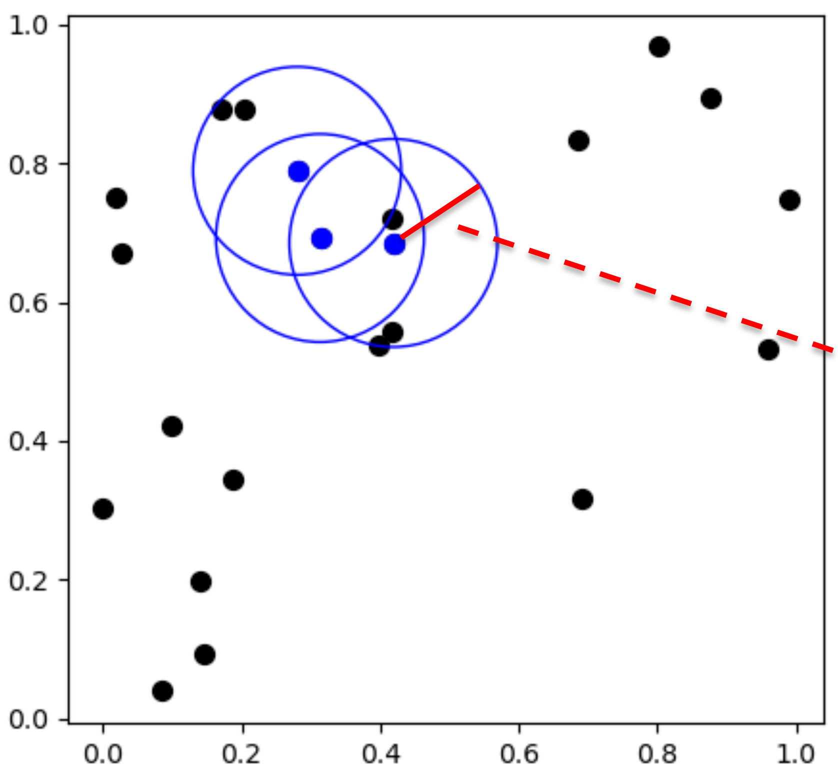
DBSCAN intuition

eps = **0.15** and **min_points** = **4**



DBSCAN intuition

eps = **0.15** and **min_points** = **4**



eps = **0.15** means that the radius of the neighborhood around each data point is **0.15** units of distance.

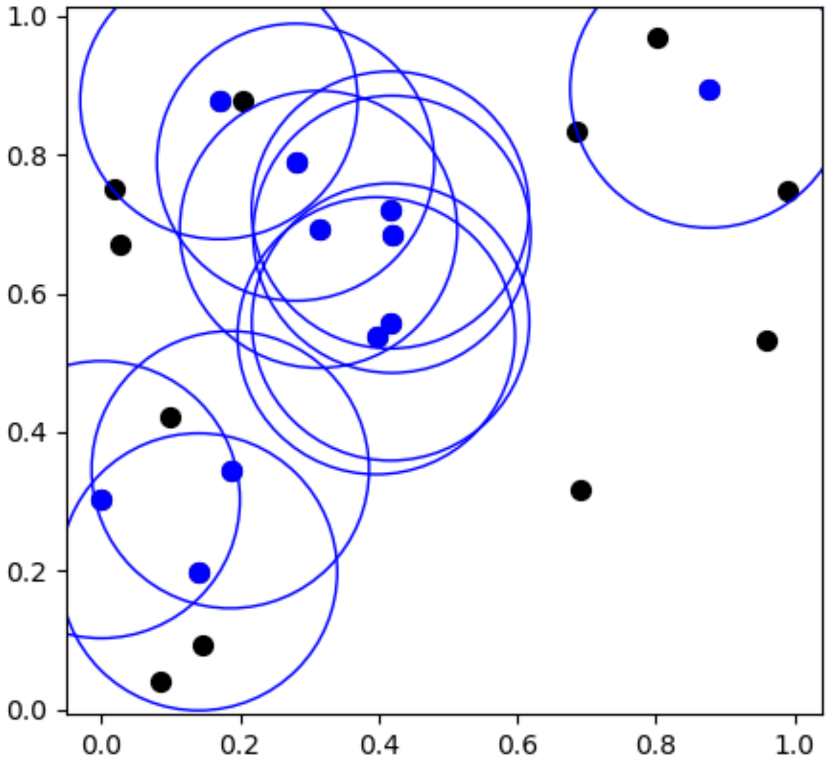
DBSCAN intuition

eps = 0.2 and **min_points** = **4**

So what happens if we increase **eps** from **0.15** to **0.2**?

DBSCAN intuition

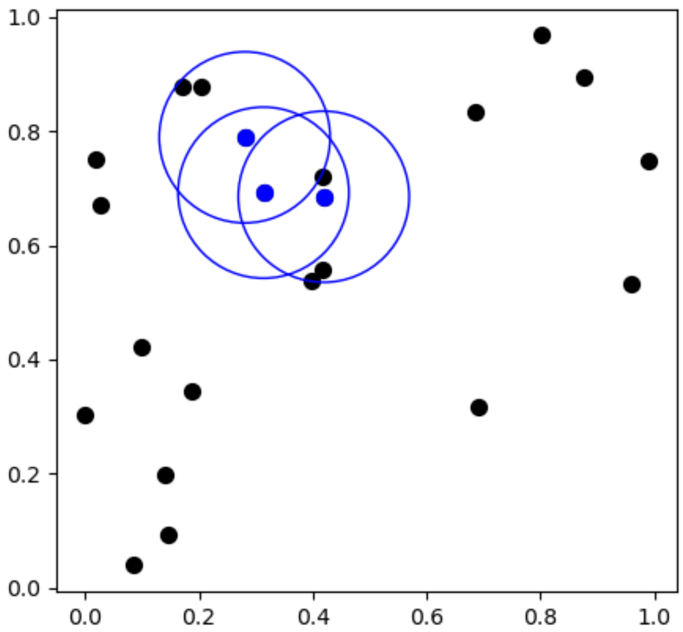
eps = 0.2 and **min_points** = **4**



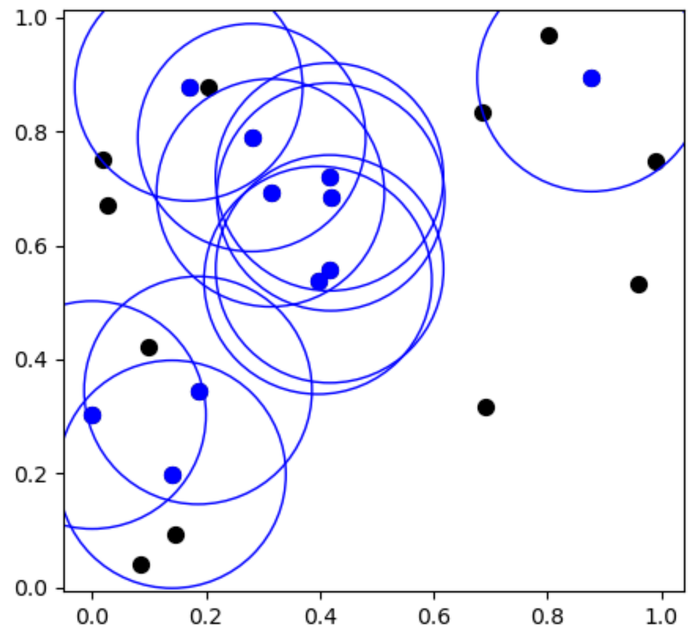
DBSCAN intuition

Side by side comparison varying **eps**

eps = 0.15 and **min_points** = **4**



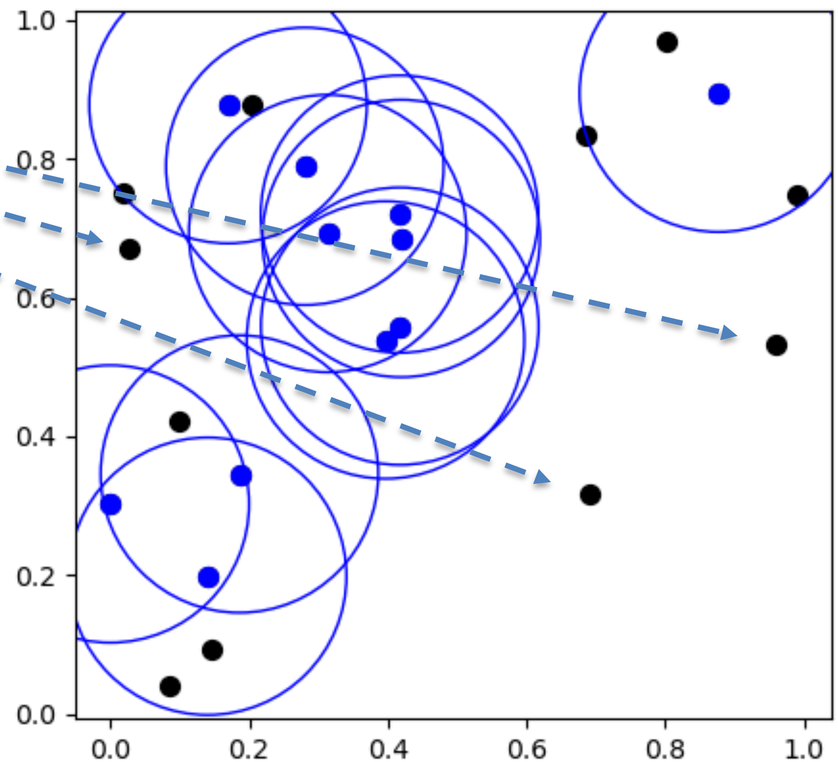
eps = 0.2 and **min_points** = **4**



DBSCAN intuition

eps = 0.2 and **min_points** = 4

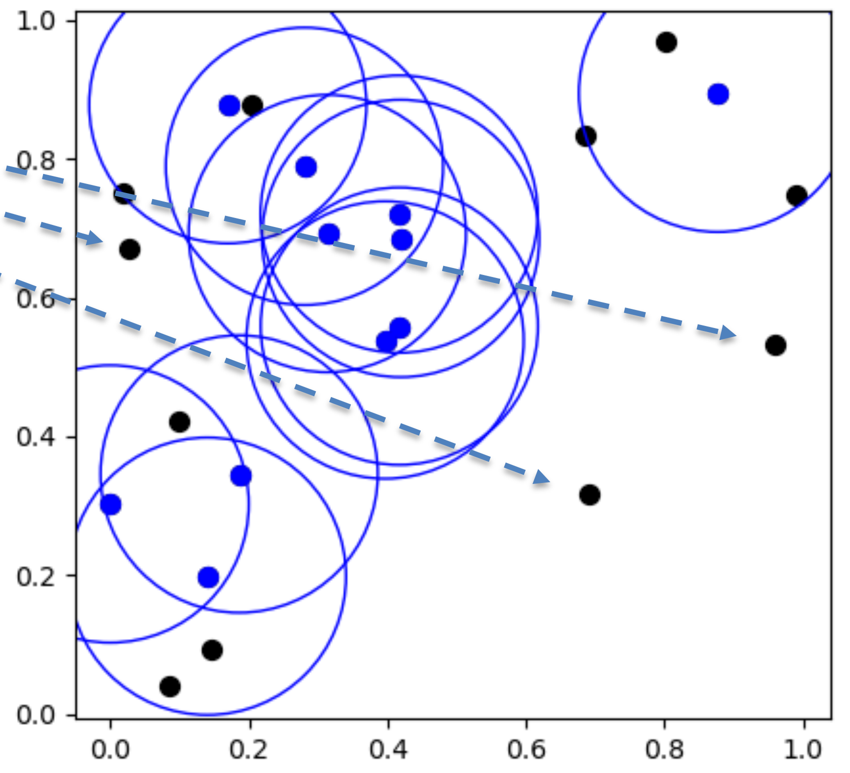
What are **these points** that fall outside the core points radius?



DBSCAN intuition

eps = 0.2 and **min_points** = 4

What are **these points** that fall outside the core points radius?

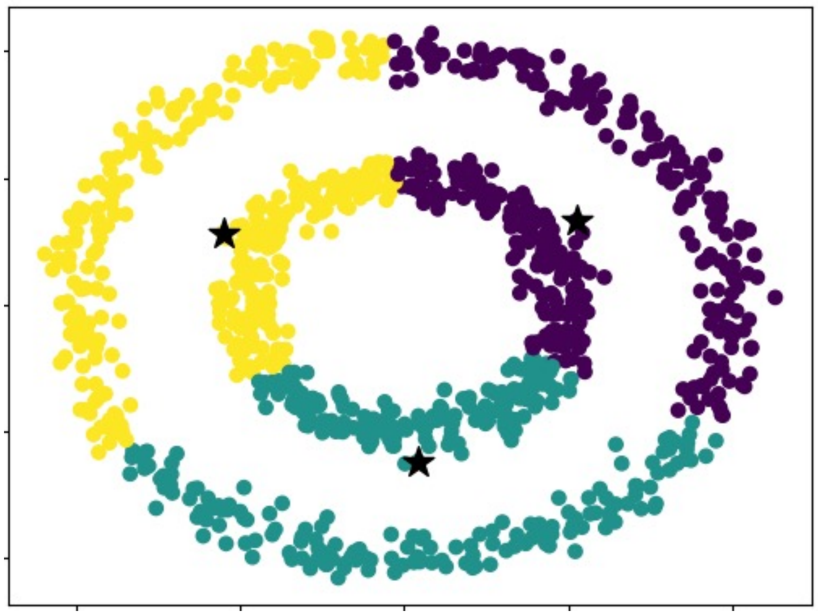


Noise!*

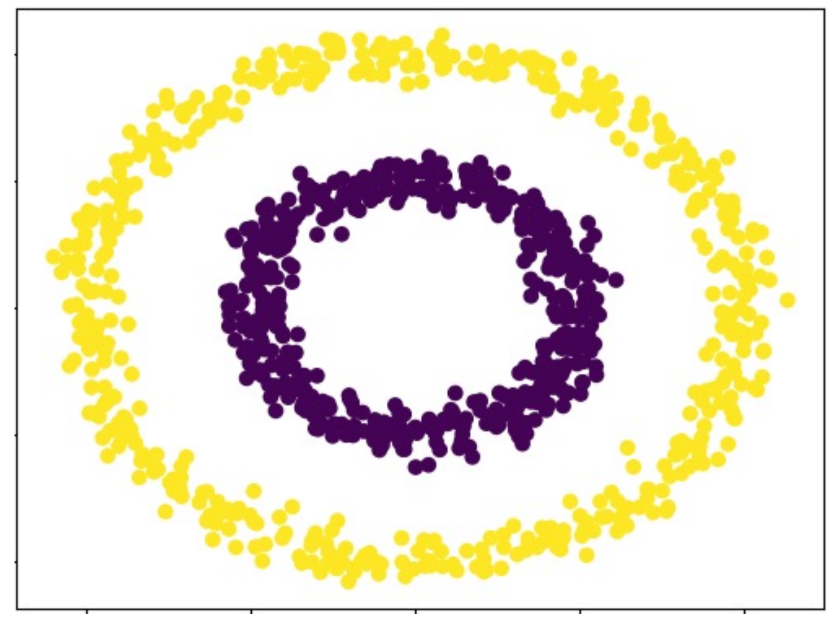
* In most implementations, these points are assigned to the -1 cluster

K-means vs DBSCAN

- You specify the number of clusters in k-means via **k**
- In DBSCAN you have to specify **eps** and **min_points**, you can't determine beforehand the number of clusters
- Side by side example in the 2 concentric circles data



K-means

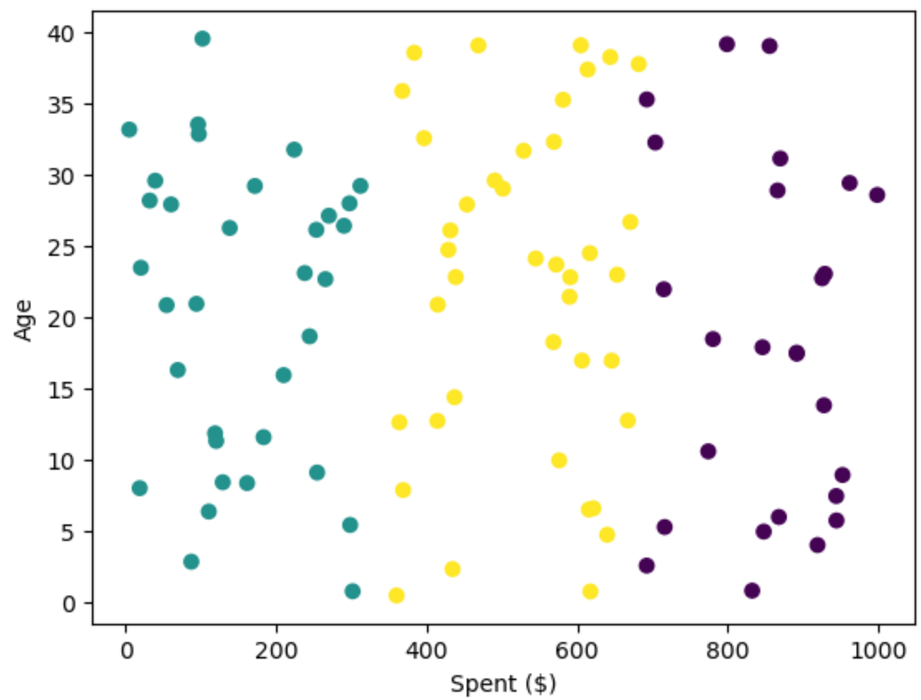


DBSCAN

How to define k in k-means?

- May be determined by the problem

*A company wants to **target three customer segments**. We don't know an appropriate way to group such customers beforehand, then we can use k-means with k = 3*



How to define k in k-means?

- Maybe we have some domain knowledge about the problem

We may want to segment an image, and we know there are 4 distinct regions in the image (e.g. ocean, beach, mountains-sky and city)

Image generated with DALL-E

How to define k in k-means?

- Maybe we have some domain knowledge about the problem

We may want to segment an image, and we know there are 4 distinct regions in the image (e.g. ocean, beach, mountains-sky and city)

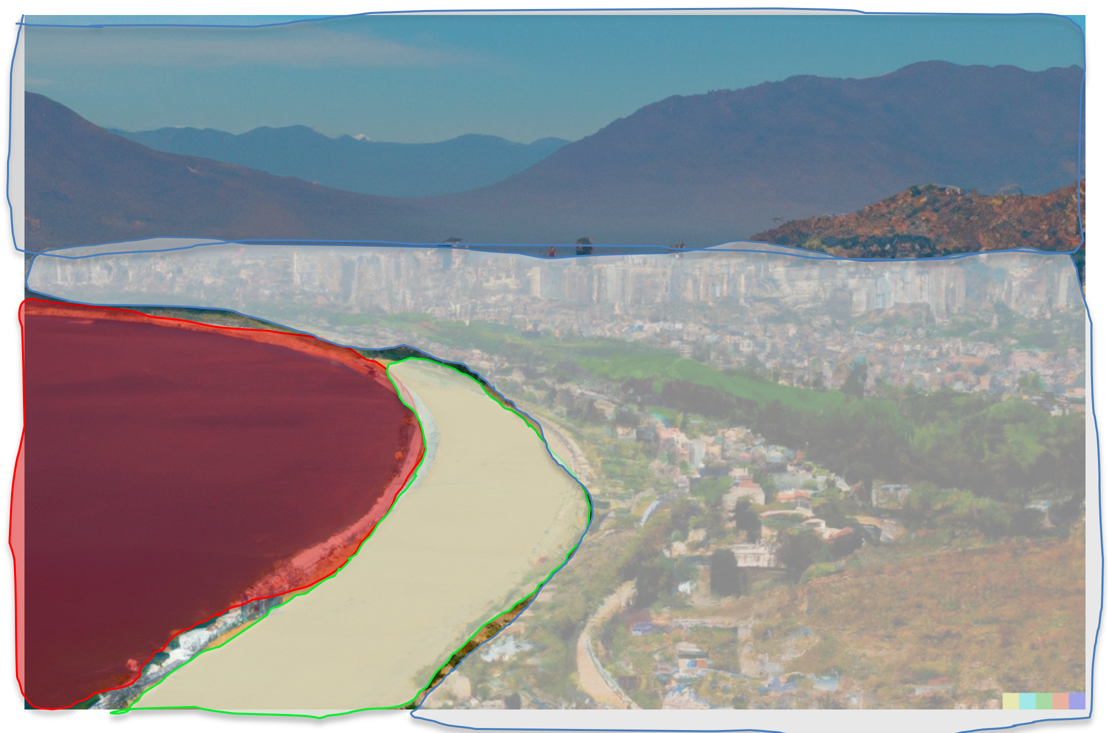
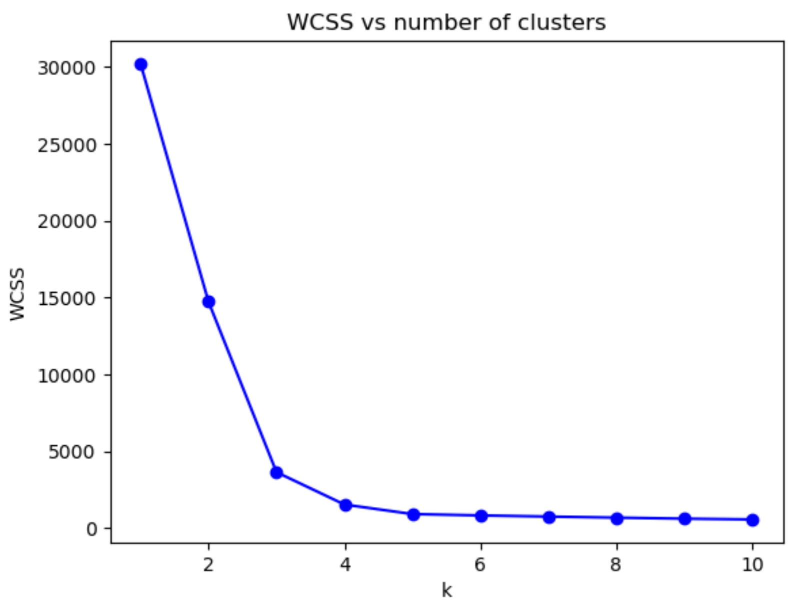


Image generated with DALL-E

How to define k in k-means?

- An “uninformed” approach is to use the **elbow method**

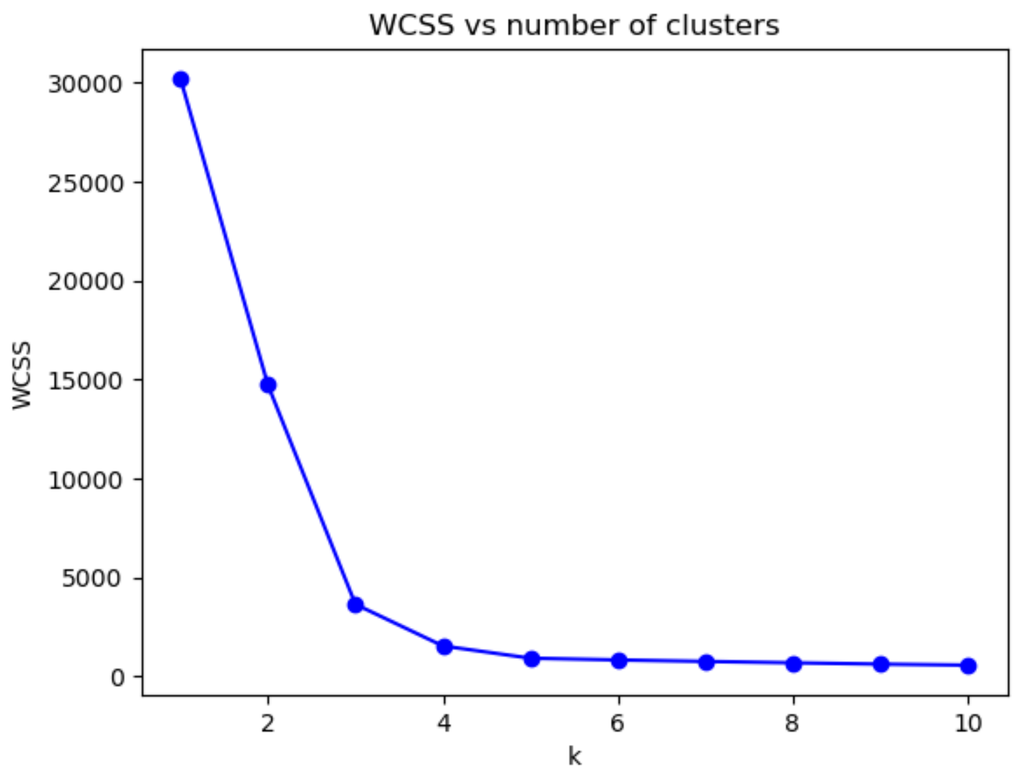
1. Use a **clustering quality measure** to assess the quality of different clustering executions
2. Plot such measure **varying k**
3. Where we find the "**elbow**" is the number of appropriate clusters



** WCSS = within-cluster sum of squares*

Elbow method (WCSS)

- WCSS is the sum of the squared distances between each point in the cluster and the centroid of that cluster
- **WCSS is a measure of how well the data points in a cluster can be represented by the centroid of that cluster**
- WCSS measure of how spread out the points in a cluster are around the centroid of that cluster



Summary

- Clustering is an important ML task (very useful in practice!)
- **k-means** (centroid-based) and **DBSCAN** (density-based)
- **Hyperparameters** can be difficult to set, if you are not familiar with what they represent to the algorithm
- **Clustering evaluation.** we haven't discussed them in detail, but most metrics focus on how "close" points in the same cluster are and how far they are from points in another cluster (See **silhouette score**)

Coming up next…

- Clustering and ensemble examples (Tutorial this week)
- Search (Next lectures)