K-MEANS vs DBSCAN

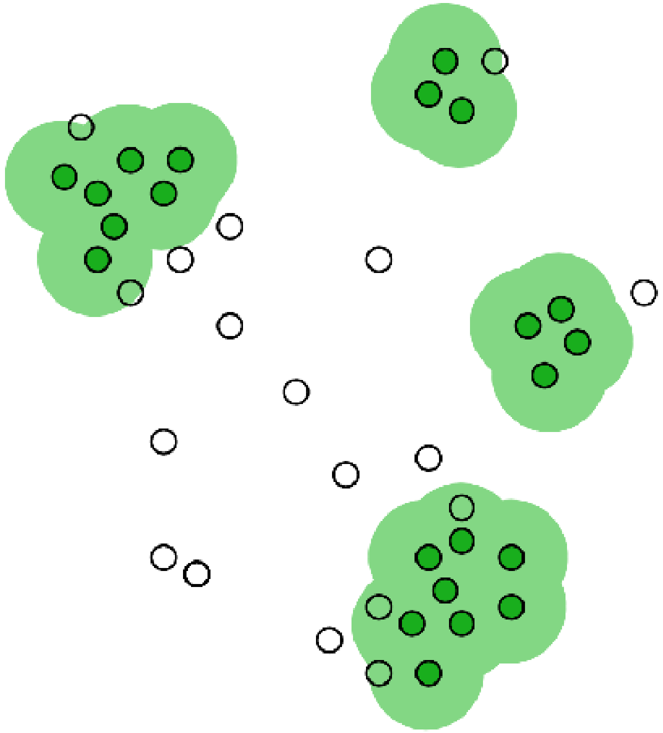
Algorithms for solving clustering problem

By:

- Ahmed Abd-elsattar Mahmoud
- Ahmed Abouabah Owais Tantawy
- Mohamed Salah-el-den Hussien
- Mohamed Ahmed Sayed
- Omar wael abdellatif
- Suhaib Ali Hassan Muhammed

Agenda

- Clustering problem
- Different approaches
- K-means algorithm
- K-means problems
- DBSCAN algorithm
- K-means vs DBSCAN
- Conclusion



Machine learning

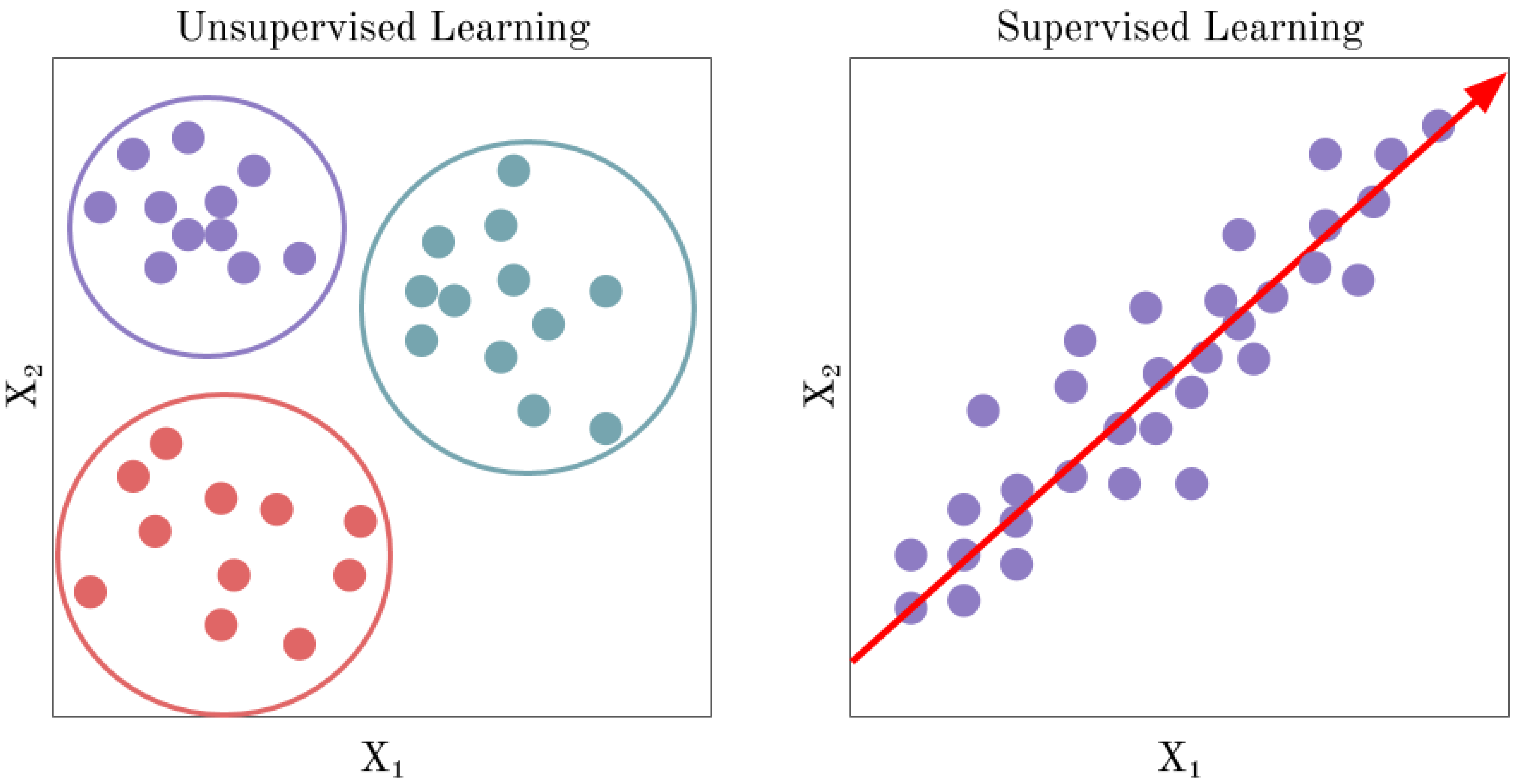
UN-Supervised

- **No labels** are given to the learning algorithm
- goal is to discovering hidden **patterns** in data
- **Ex : Clustering problem**

Supervised

- The system is presented with example **inputs** and their desired **outputs**
- the goal is to learn a **general rule** that maps inputs to outputs
- **EX : regression problem**

Machine learning

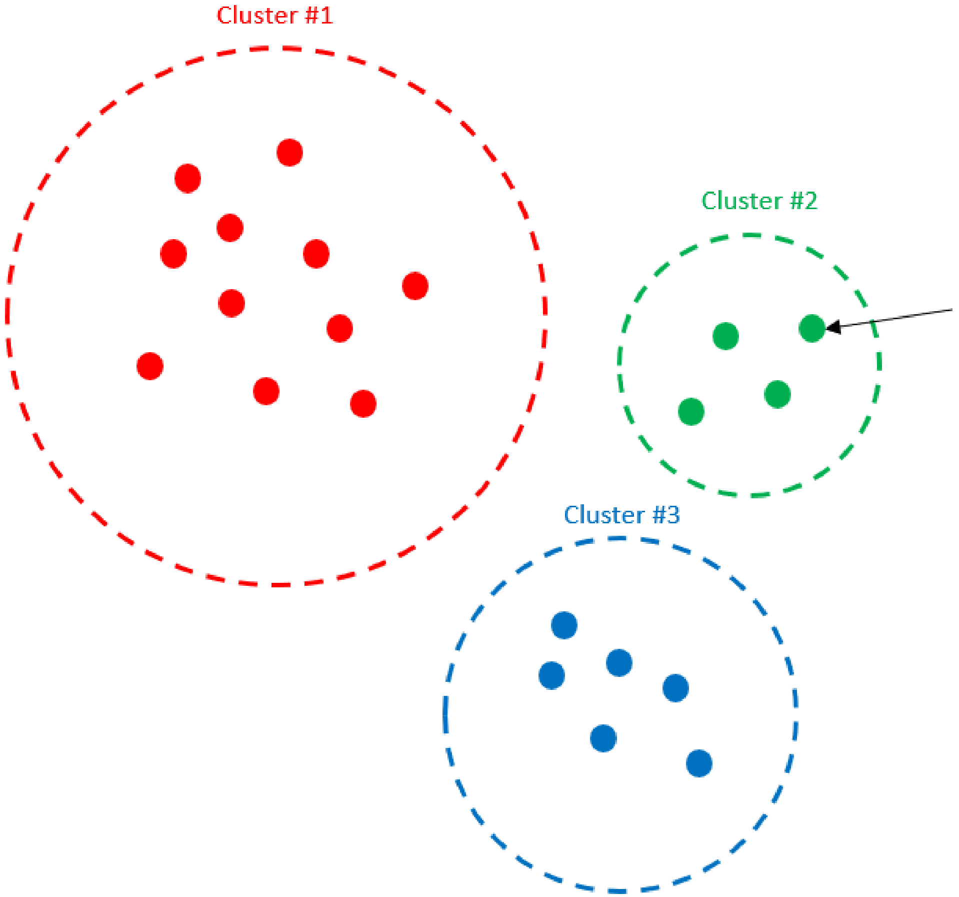


Unsupervised Learning

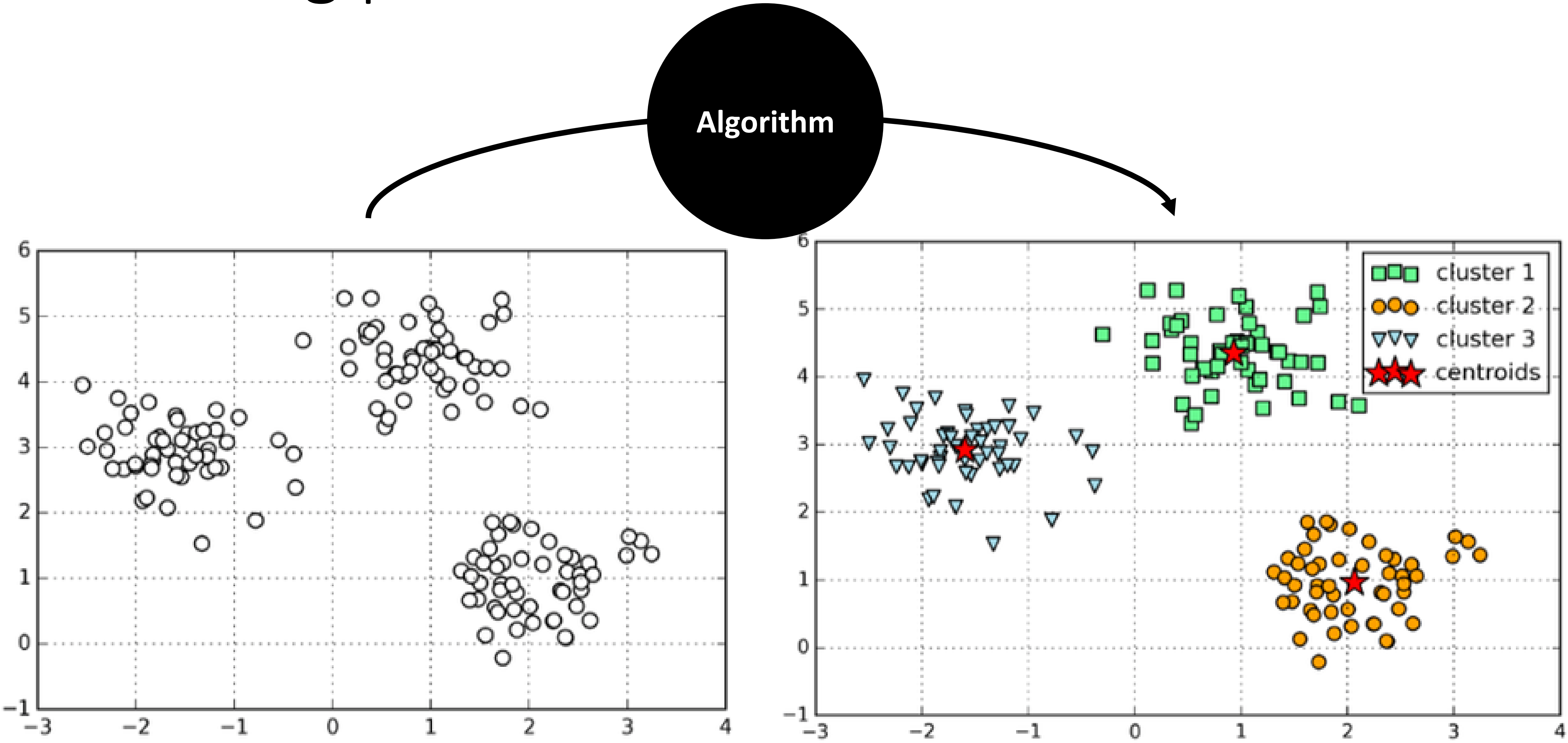
Supervised Learning

Clustering definition

“the process of organizing objects into groups whose members are similar in some way”



Clustering problem



Algorithm

Clustering applications

Identifying Fake News

- by taking in the content of the fake news article, the corpus, examining the words used and then clustering them

Spam filter

- looking at the different sections of the email
(header, sender, and content).
- The data is then grouped together.
- These groups can then be classified to identify which are spam

Marketing and Sales

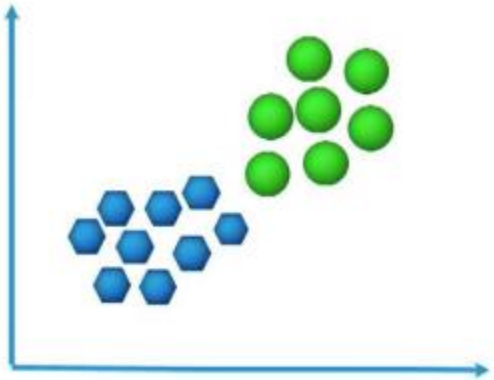
- group together people with similar traits and likelihood to purchase

Different approaches to solve clustering

Clustering types

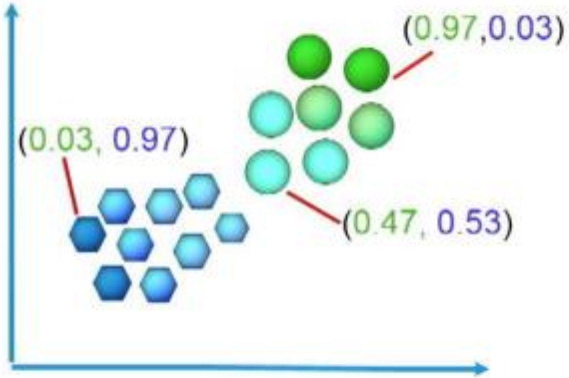
Hard Clustering:

each data point either belongs to a cluster completely or not.



Soft Clustering:

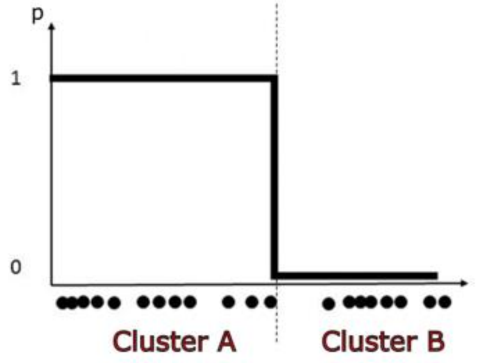
instead of putting each data point into a separate cluster, a probability or of that data point to be in those clusters is assigned.



Clustering types

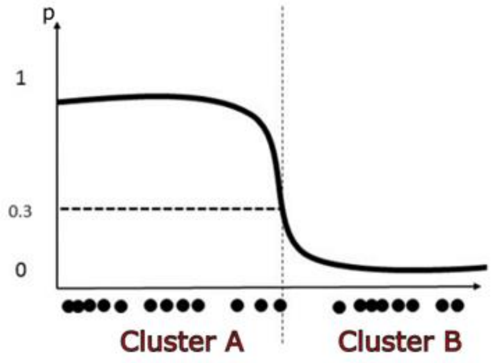
Hard Clustering

- **DBSCAN**
- **K-Means**
- Hierarchical clustering



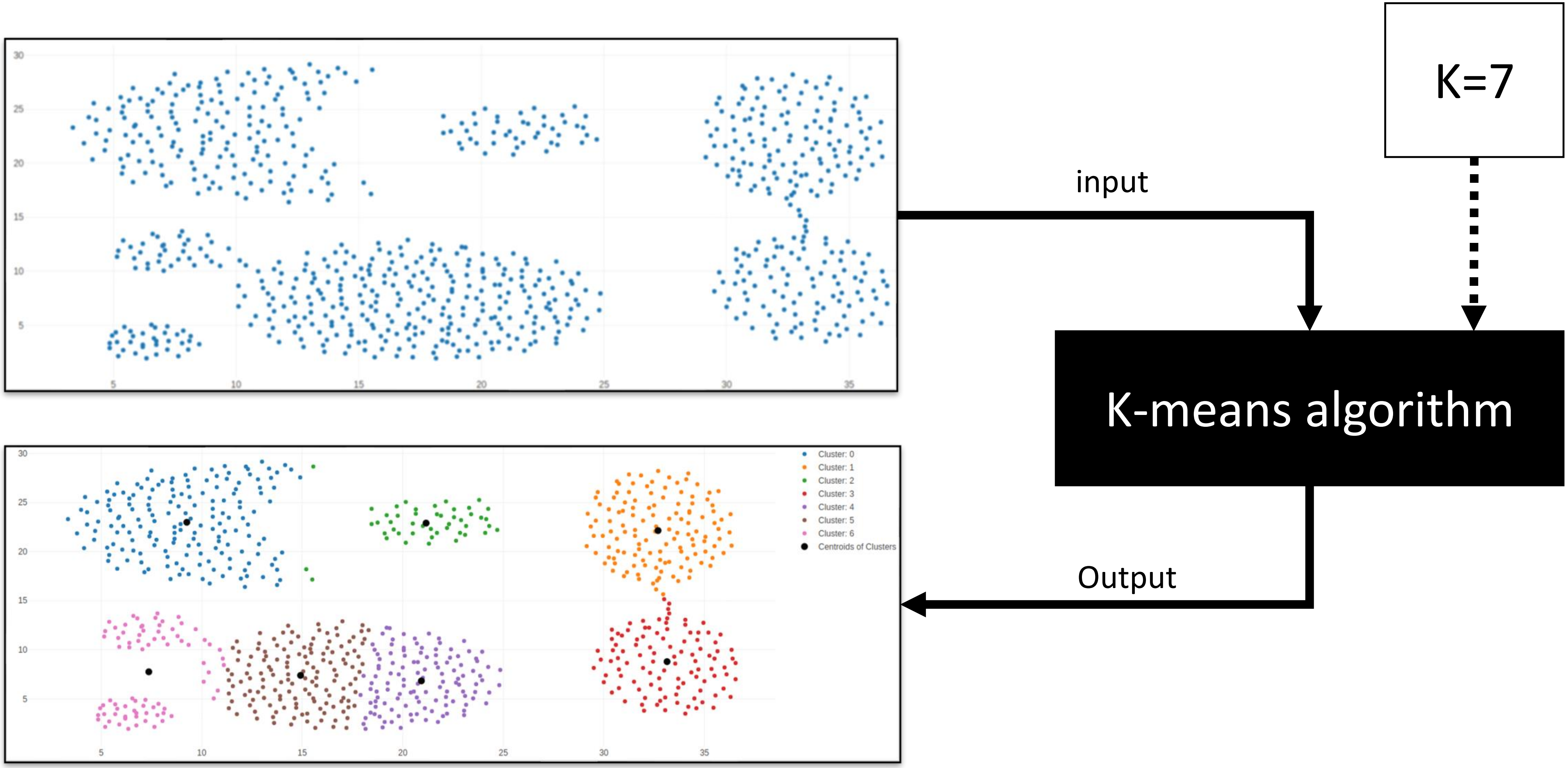
Soft Clustering

- Gaussian Mixture Model
- Fuzzy K-means



K-means algorithm

K-means algorithm



K=7

input

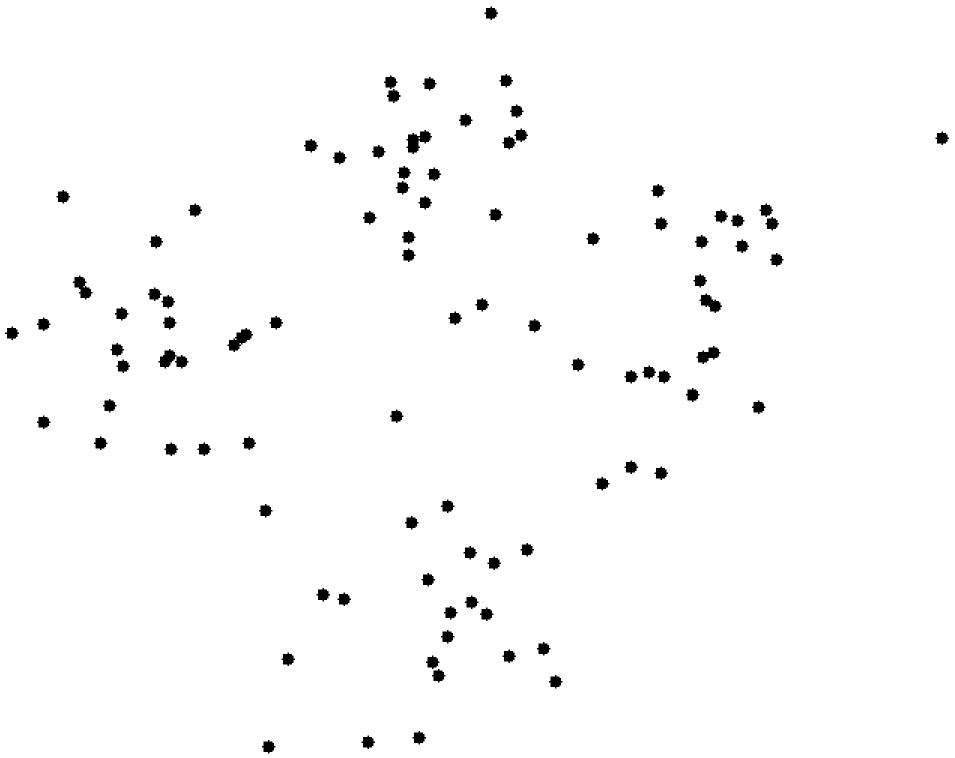
K-means algorithm

Output

K-means algorithm

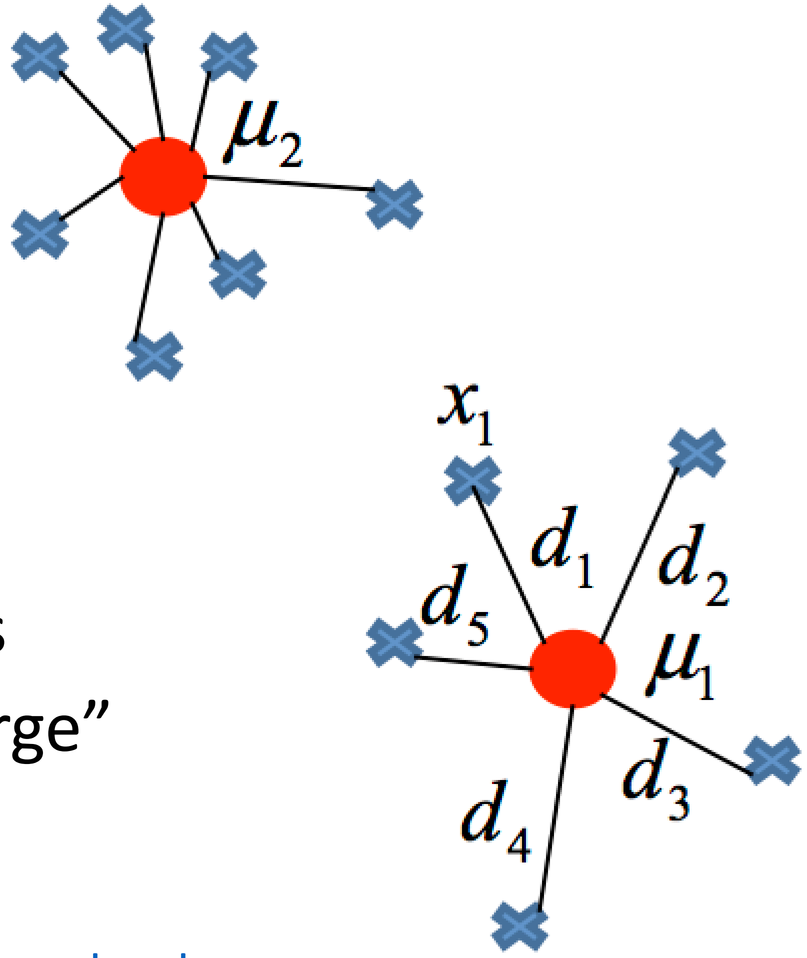
- The approach behind this simple algorithm is just about some iterations and updating clusters as per distance measures that are computed repeatedly.
- *“k”* is the number of clusters that are to be formed
- The *“means”* part because we update the center of the cluster by the mean of the near points

K-means algorithm steps



K-means problems steps

1. Specify the **K** “number of clusters”
2. Randomly initialize the centroids
3. Assign each data point to a cluster
4. Calculate the **means** of the clusters
5. Update the centroids with the new means
6. Iterate until the update is so small “converge”

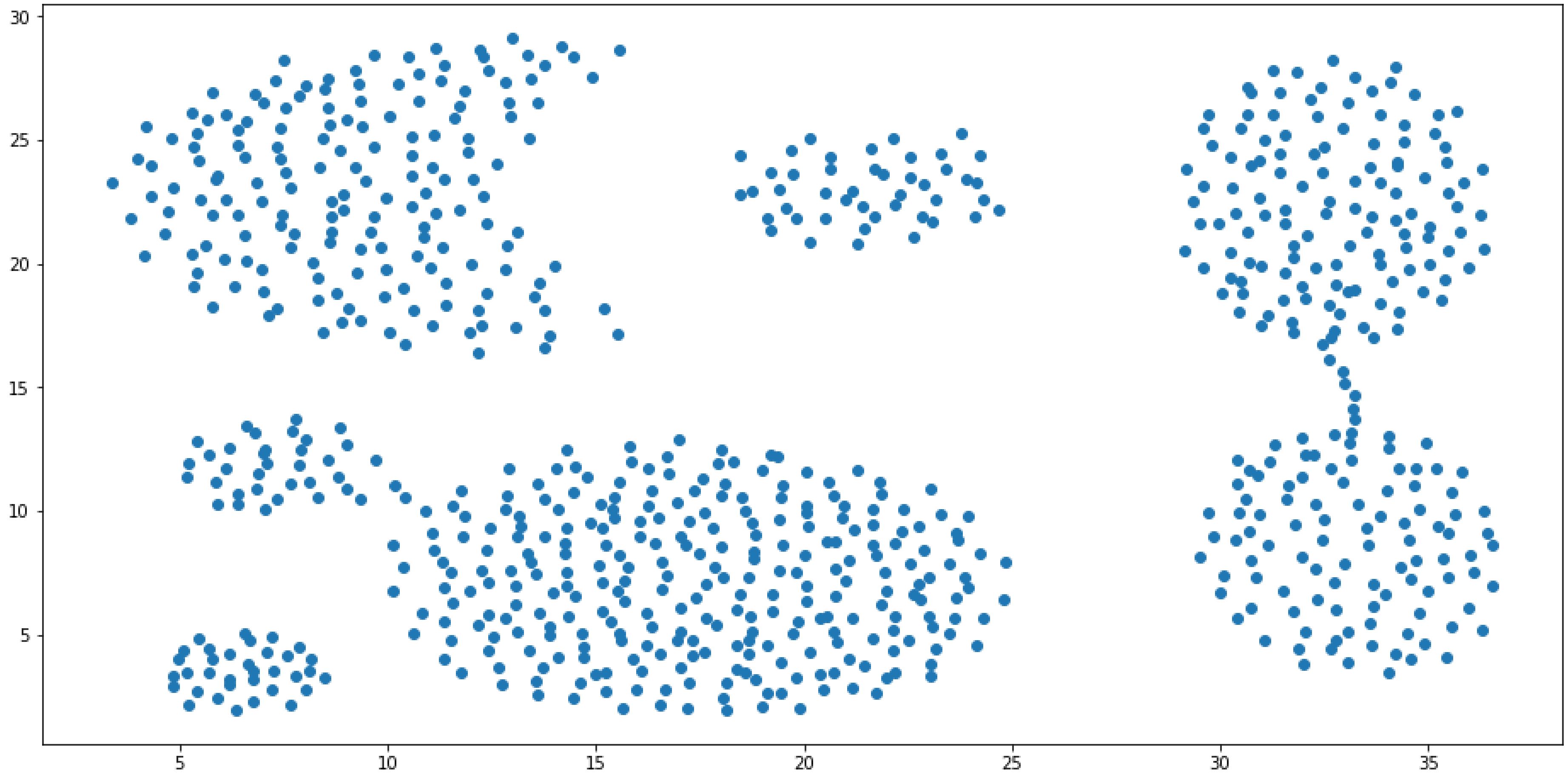


Try it yourself

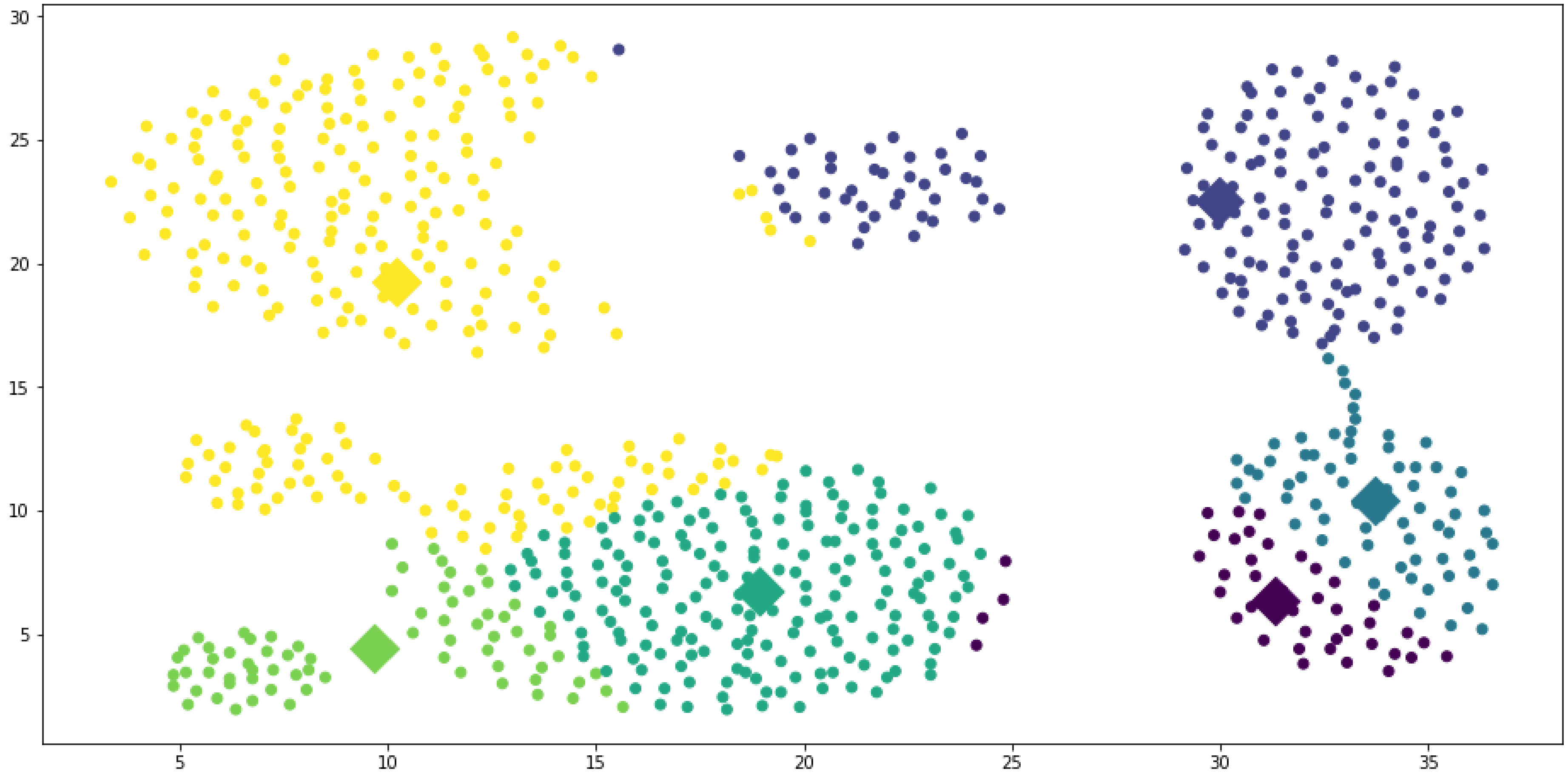
https://stanford.edu/class/engr108/visualizations/kmeans/kmeans.html

https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

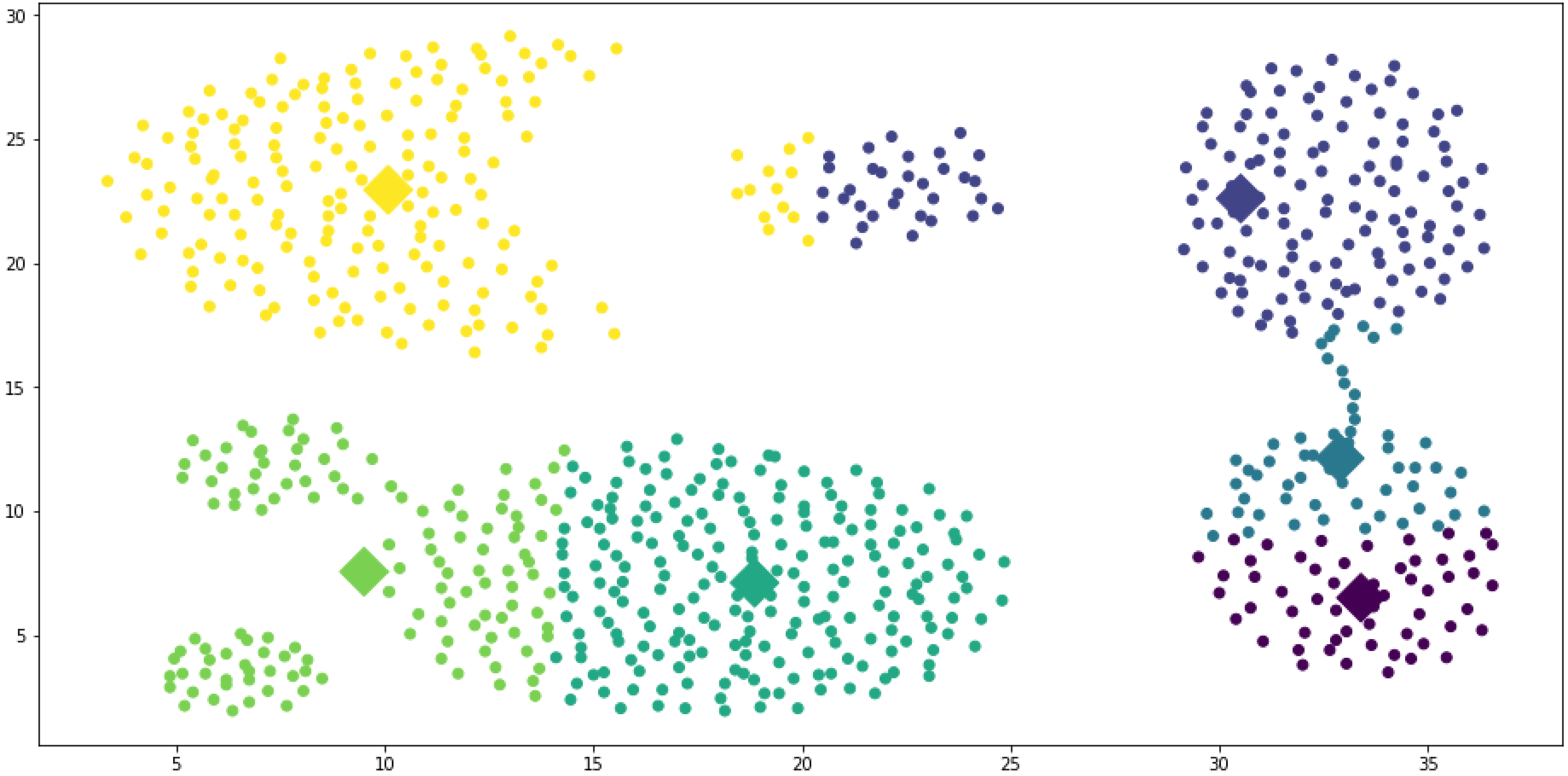
Input



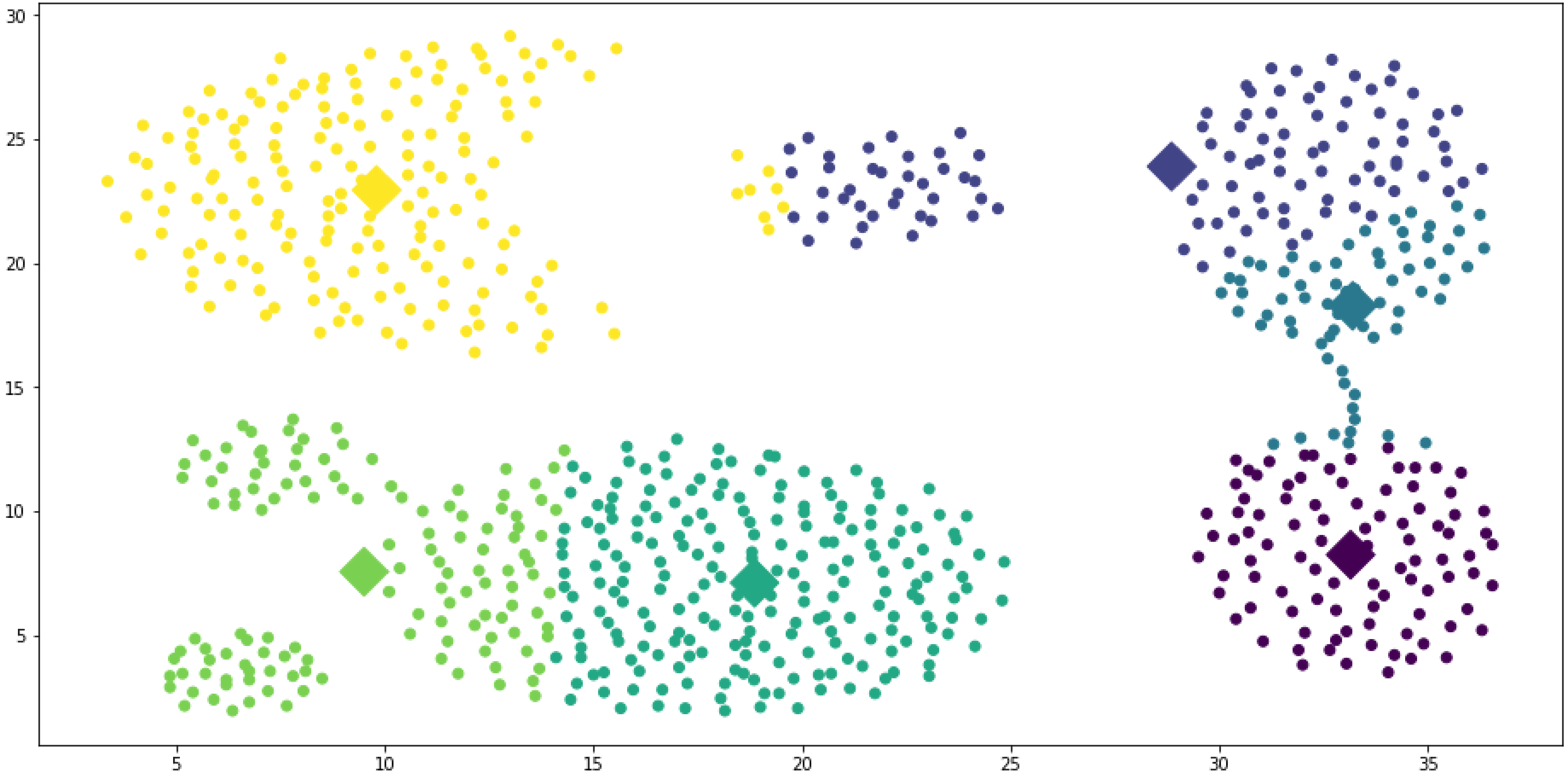
Randomly initialize centroids



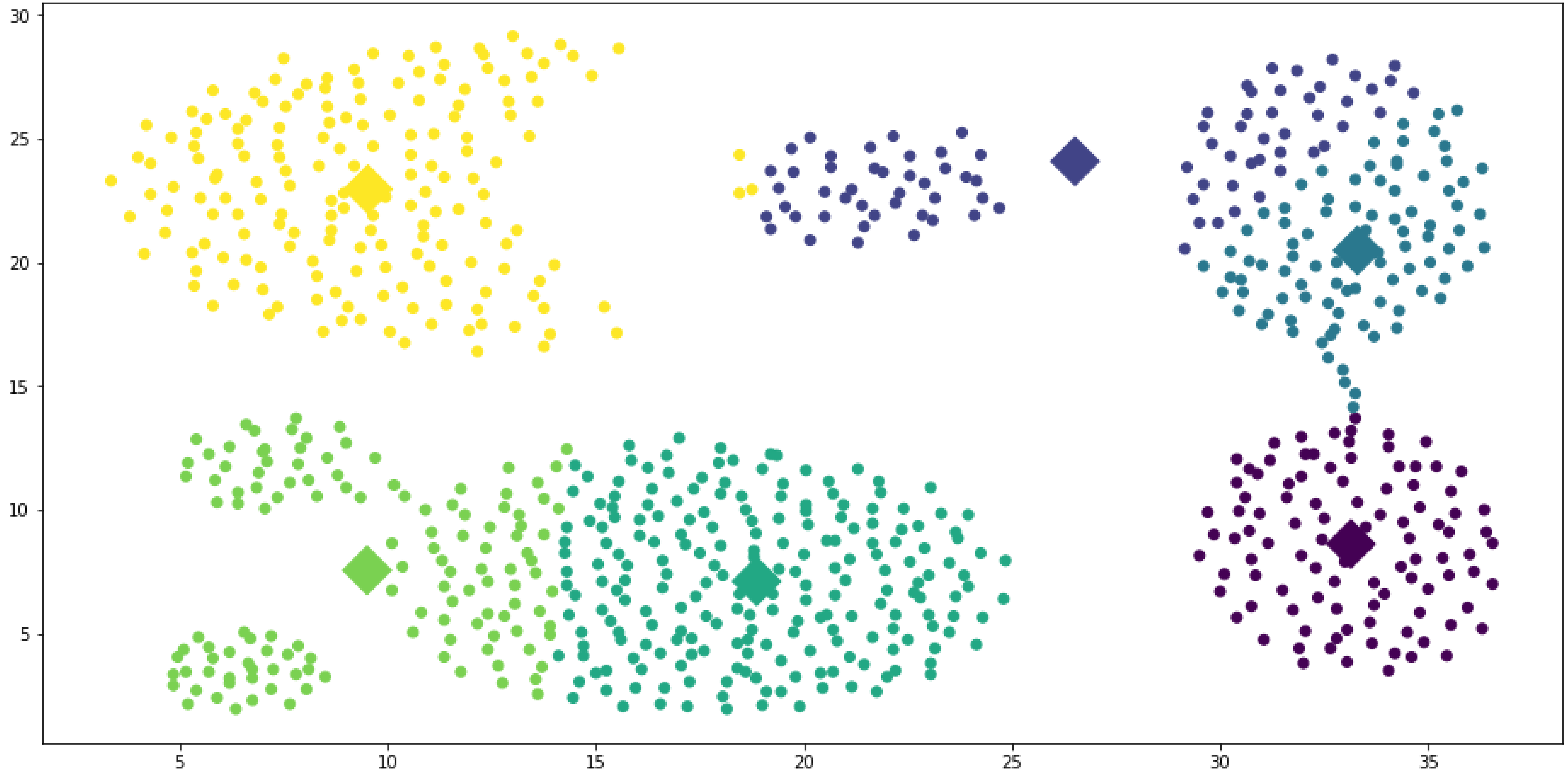
Update centroids with means #1



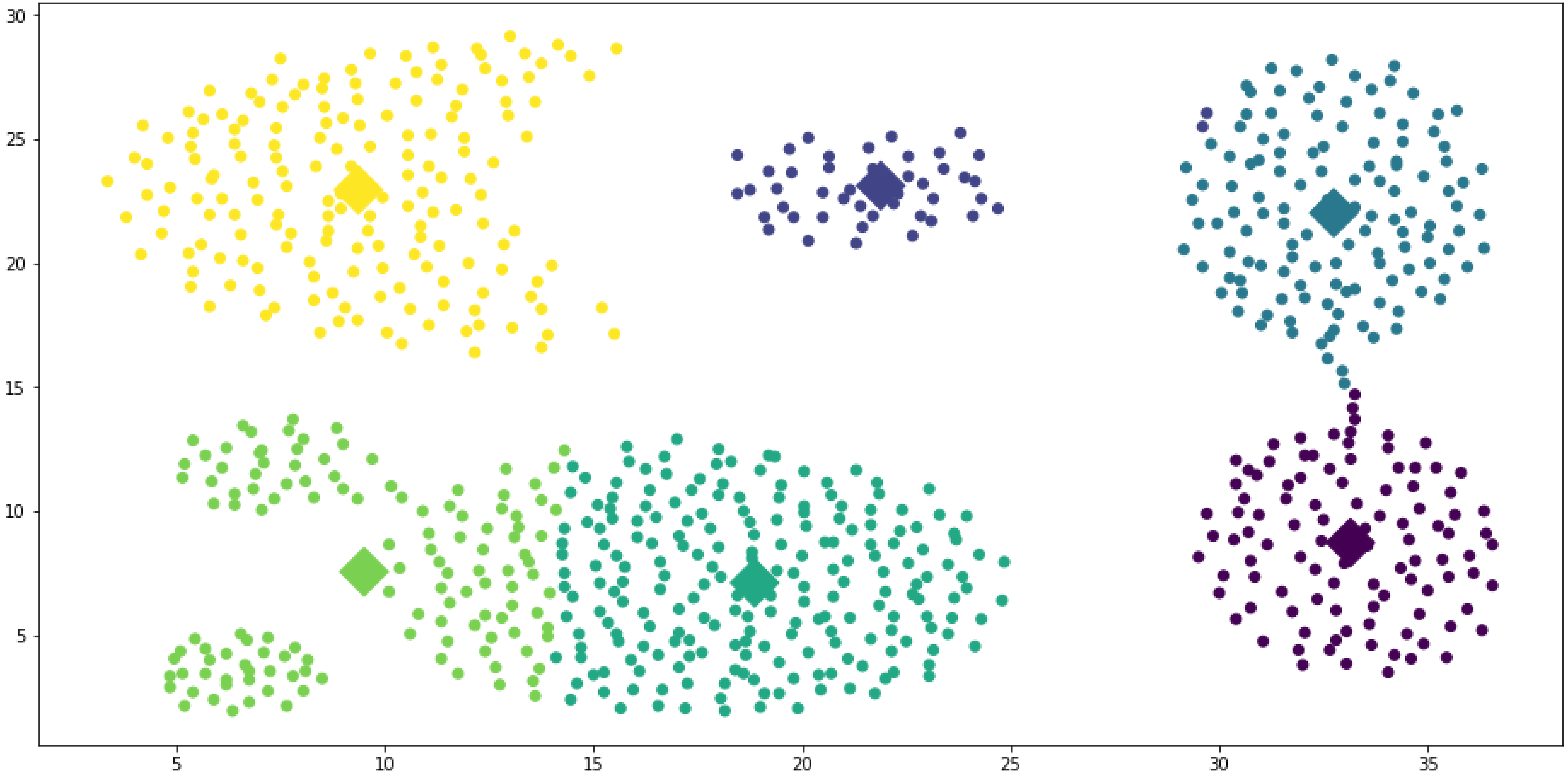
Update centroids with means #2



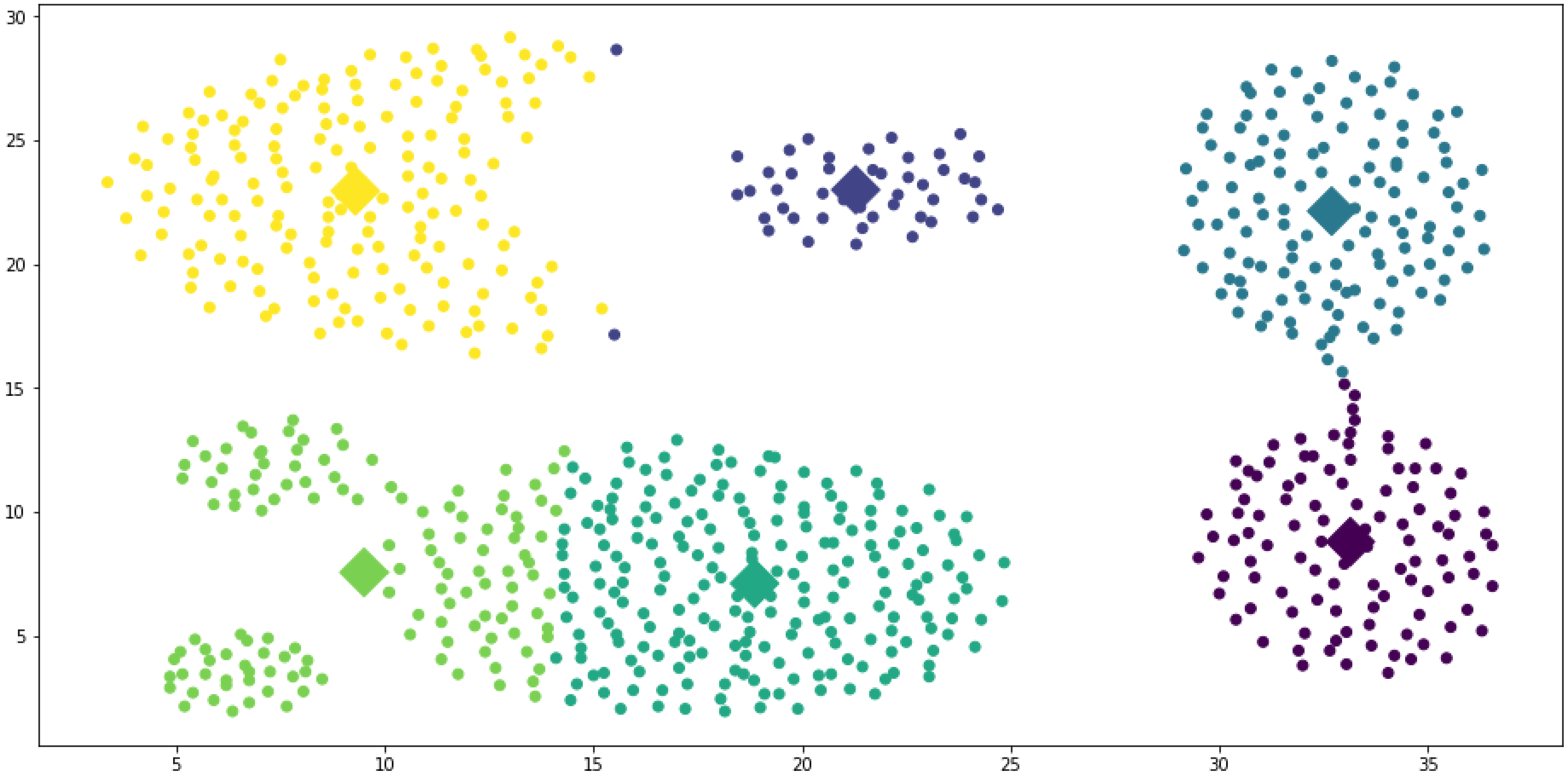
Update centroids with means #3



Update centroids with means #4



Update centroids with means #5



Algorithm implementation

1. Let $x = \{x_1, x_2, x_3, \dots, x_n\}$ be the data points and c $= \{c_1, c_2, \dots, c_k\}$ be the centroids with randomly initialized values
2. Calculate the distance between each data point and each centroid
3. Assign the data point to the closest centroid
4. Recalculate the new cluster center using

$$c_i = \frac{\sum_{i=0}^{l_i} x_i}{l_i} \quad \rightarrow \quad l_i\ represnets\ the\ number\ of\ data\ point\ in\ the\ c_i\ cluster$$

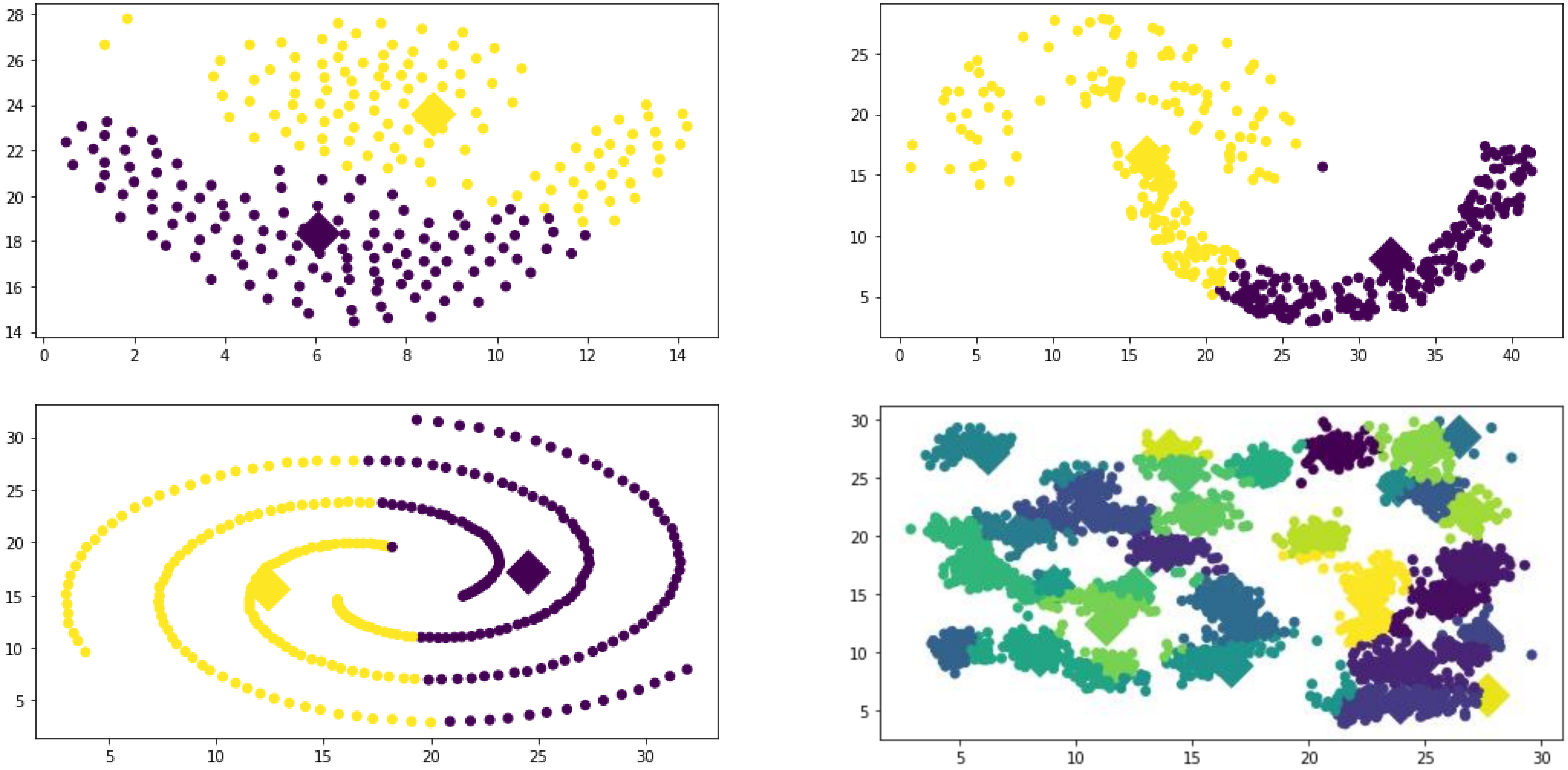
Algorithm implementation (continued)

5. Recalculate the distance between each data point and the new cluster centers
6. If centroids are no more updated … stop

Problems with K-means

- How **many clusters $k = ?$**
- the **results depends on value of K**
- The way we **initialize the centroids** is not specified one way is to initialize it randomly
- **Different run** could lead to different results
- It can happen to have **empty cluster** *"the centroid is never updated"*
- Strongly affected with **outliers**
- Cannot handle **arbitrary shape**

Arbitrarily shaped clusters with K-means



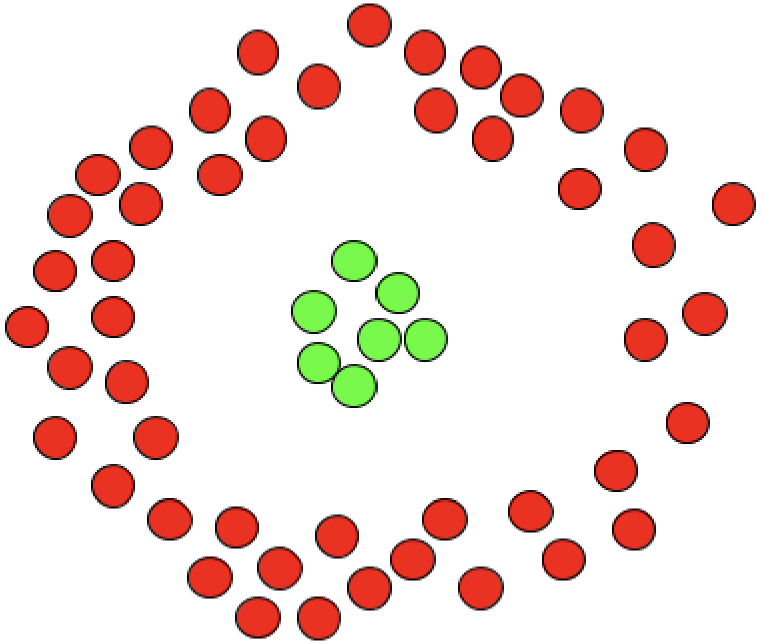
To view and run the code
https://github.com/MohamedSalahElden/clustering_algorithms/blob/main/K_means_algo_7.3.ipynb

DBSCAN

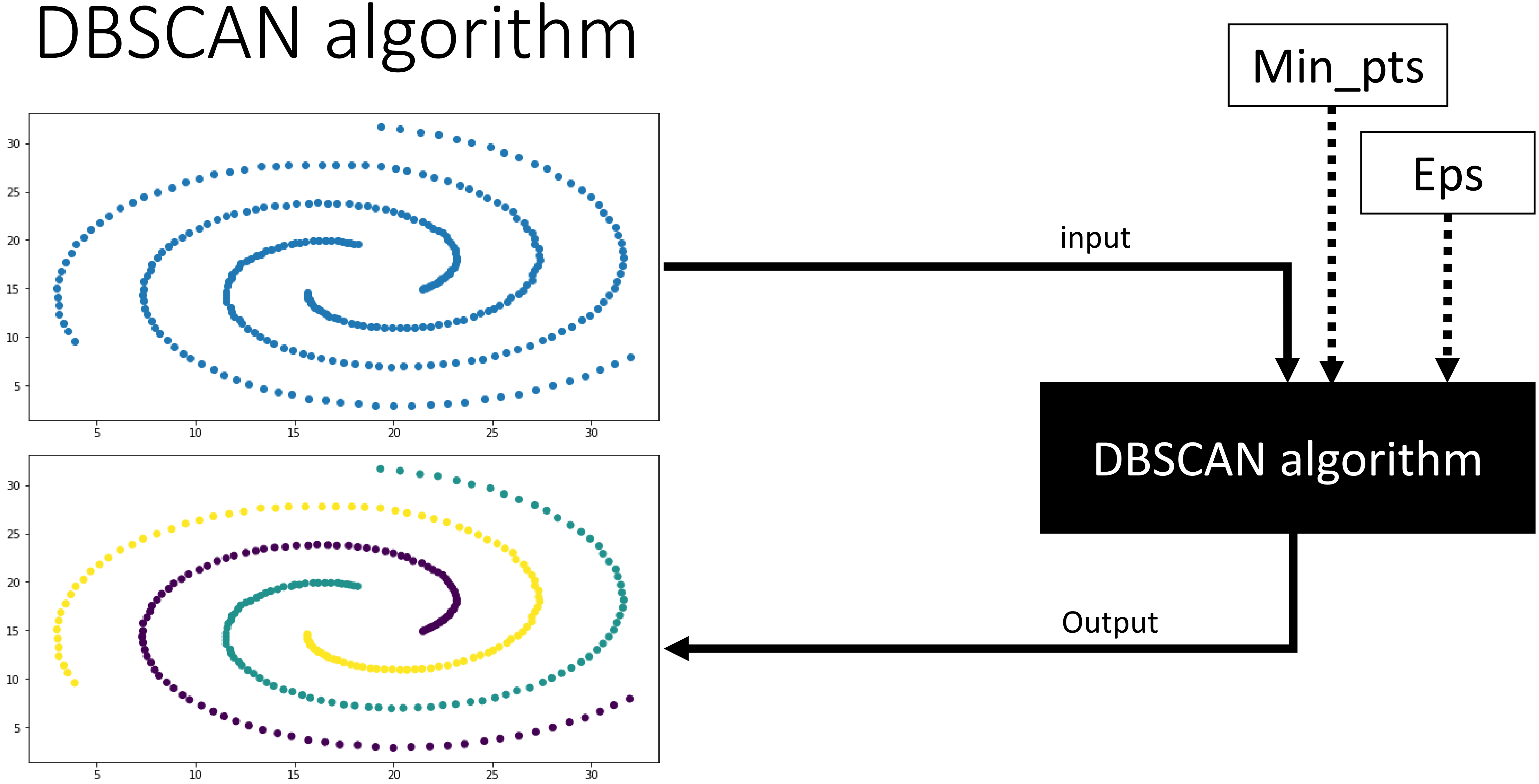
a new approach to solve K-means problems

DBSCAN algorithm

- Density Based Spatial Clustering of Application with Noise
- **Discover clusters with arbitrary shape**
- A Cluster is defined as a set of density connected points



DBSCAN algorithm



Min_pts

Eps

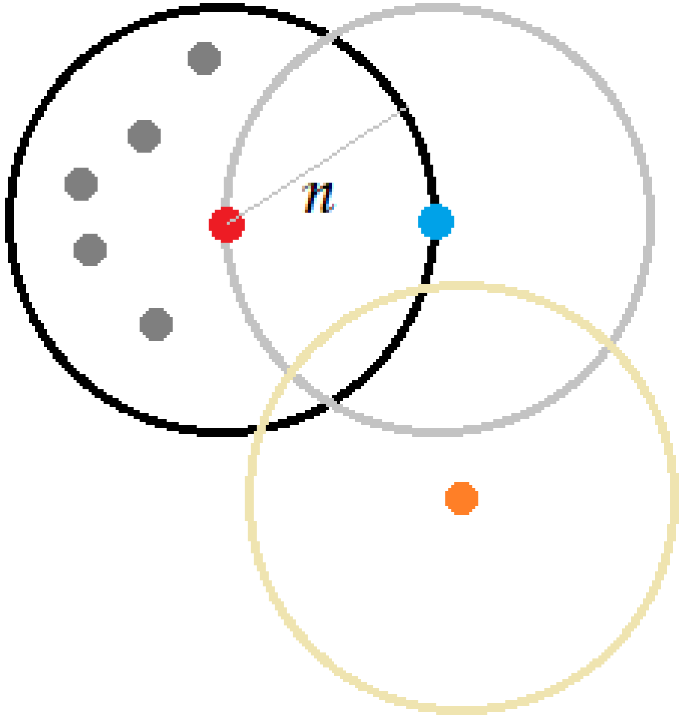
input

DBSCAN algorithm

Output

DBSCAN parameters

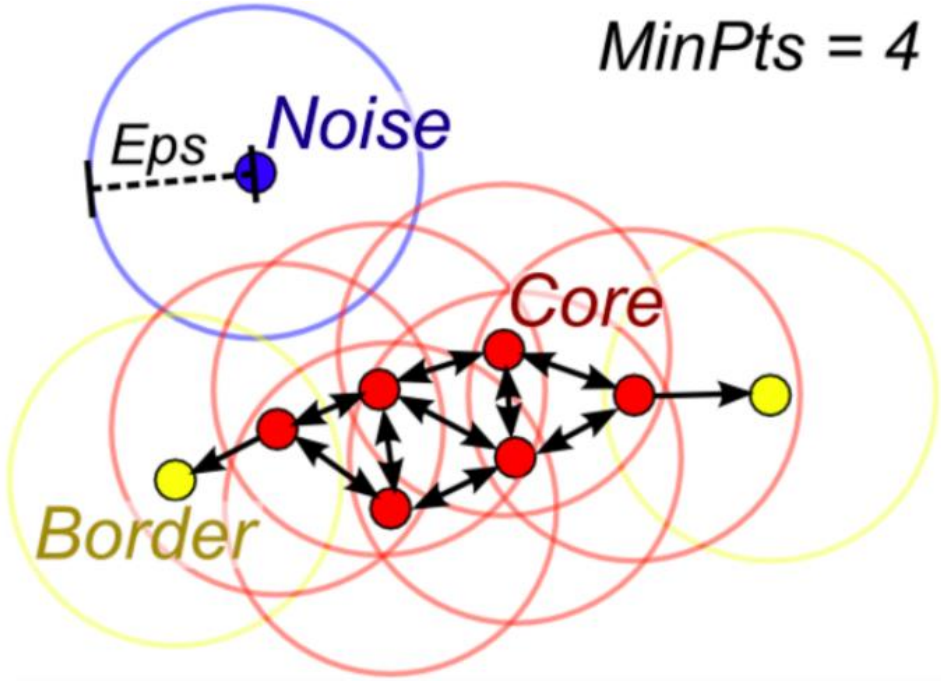
- Eps “n” : max radius of the neighborhood
- Min_pts : minimum number of points in Eps neighborhood of points
  “how dense is this region should be”



Core , border , Noise

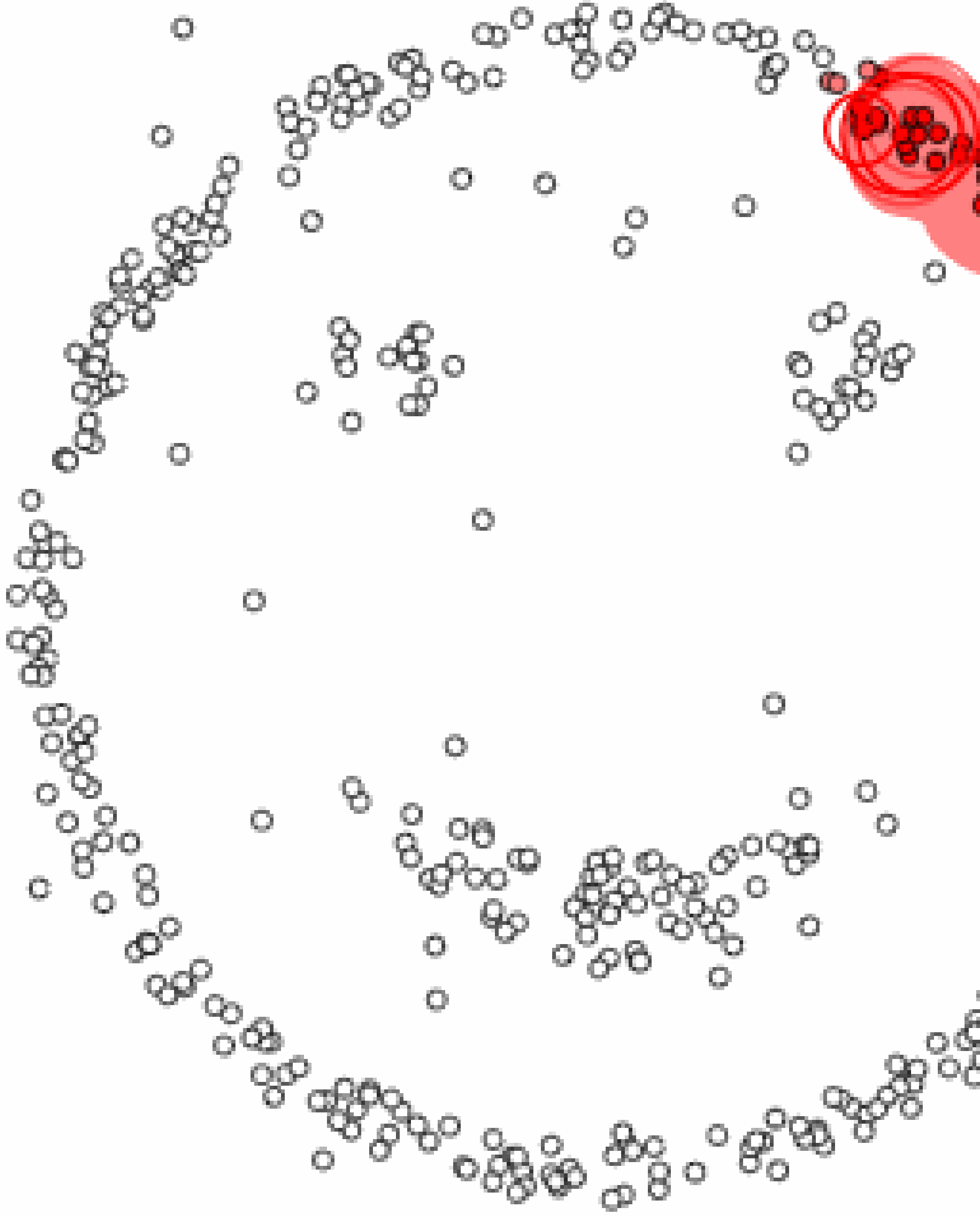
- Each point in data set could be either

1. Core
   number of points within the Eps region greater than or equal MinPts
2. Noise
   number of points within the Eps region equals to 0
3. Border
   number of points within the Eps region less than MinPts and it has a core point in it's region

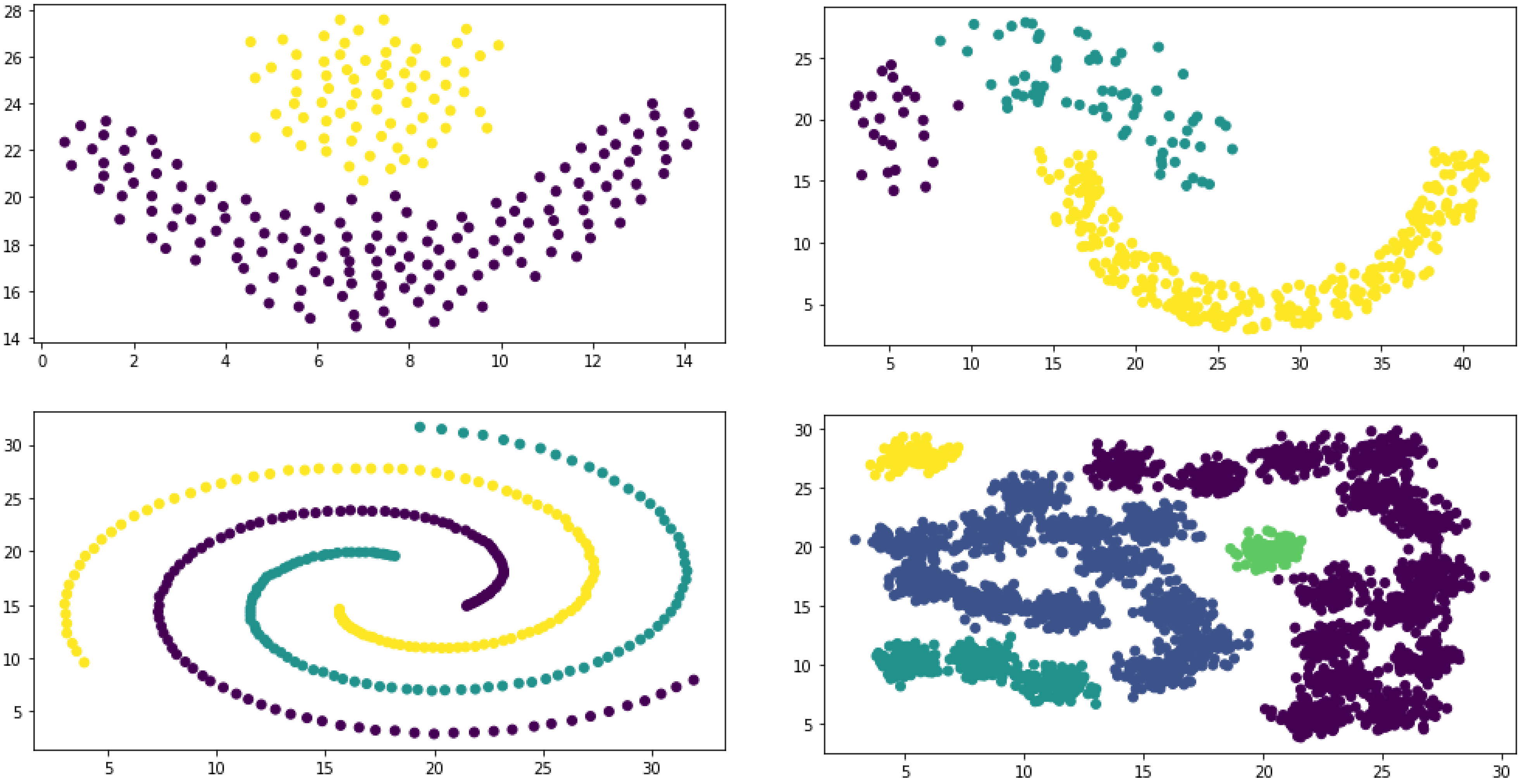


DBSCAN algorithm

- Randomly select a point p
- Get all points that are density-reachable with respect to Eps , $MinPts$
- If p is core point , a cluster is formed
- If p is boarder , visit the next point
- Continue the process until all points have been processed



Arbitrarily shaped clusters with DBSCAN

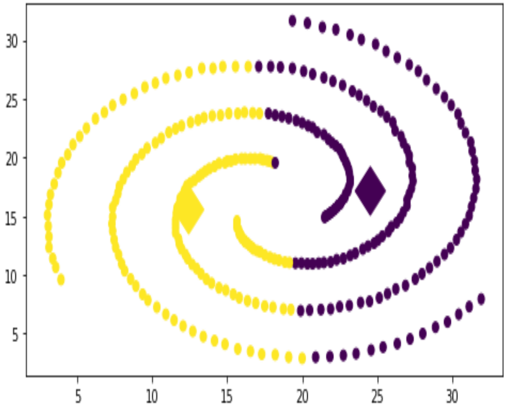


K-mean **VS** DBSCAN

Cluster shape

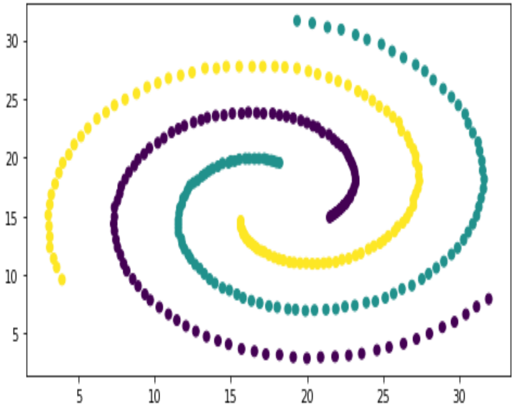
- K-means Clustering

Clusters formed are more or less **spherical** shape and must have **same feature size**.



- DBSCAN Clustering

Clusters formed are **arbitrary** in shape and **may not have same feature size.**

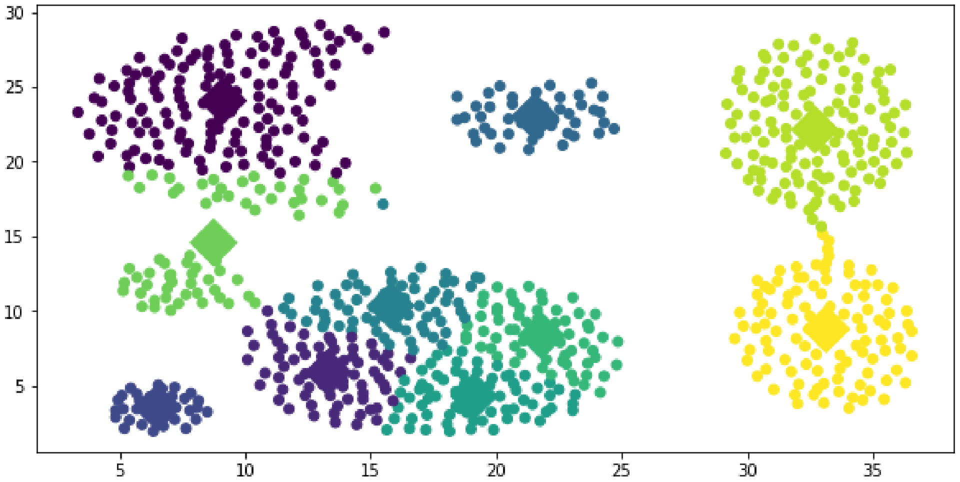


Number of clusters

- K-means Clustering

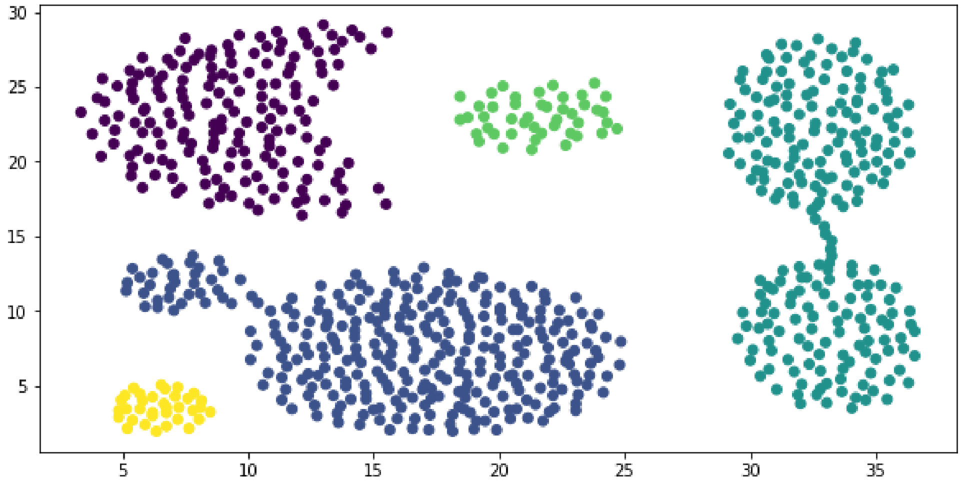
K-means clustering **is sensitive to the number of clusters** specified.

For K=10



- DBSCAN Clustering

Number of clusters need not be specified.



Number of parameters

- K-means Clustering

It requires one parameter Number of clusters **(K)**

- DBSCAN Clustering

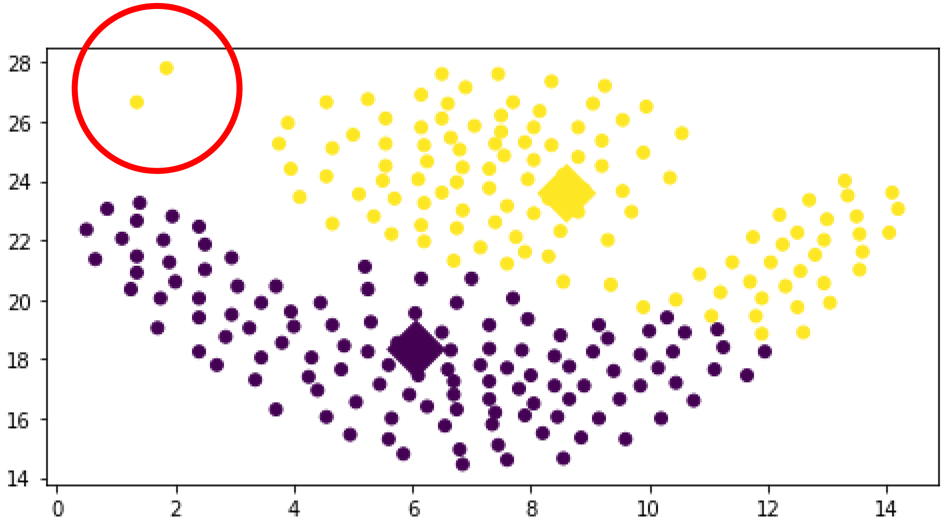
It requires two parameters : Radius(**R**) and Minimum Points(**M**)

- **(R)** determines a chosen radius such that **if it includes enough points** within it, **it is a dense area**.
- **(M)** determines the minimum number of data points required in a neighborhood to be defined as a cluster.

Outliers

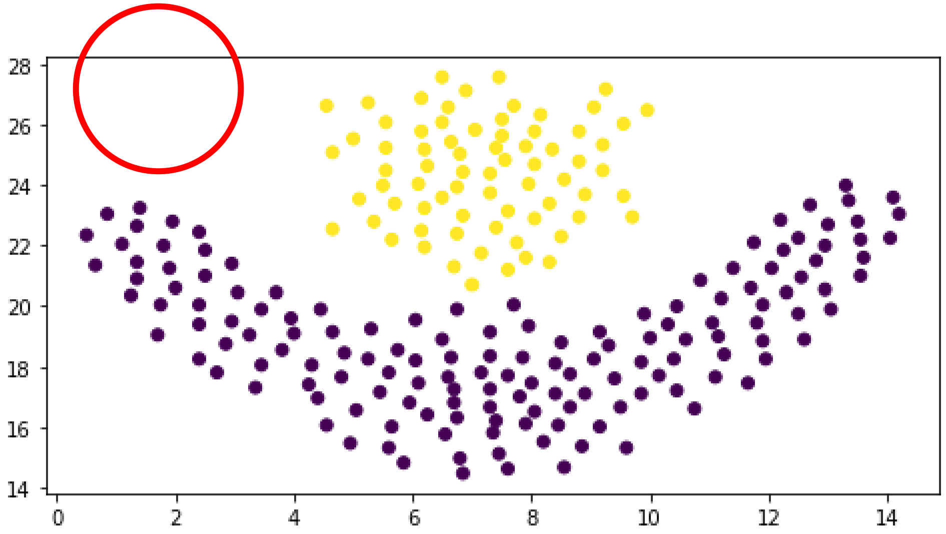
- K-means Clustering

K-means Clustering does not work well with outliers and noisy datasets.



- DBSCAN Clustering

DBSCAN clustering efficiently handles outliers and noisy datasets.



Conclusion

References

- https://towardsdatascience.com/unsupervised-learning-and-data-clustering-eeecb78b422a
- https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/
- https://towardsdatascience.com/k-means-clustering-algorithm-implementation-da0f735ab0f9
- http://cs.joensuu.fi/sipu/datasets/
- https://www.geeksforgeeks.org/difference-between-k-means-and-dbscan-clustering/

Thank you