Clustering Algorithms and their Application to Facial Image Analysis

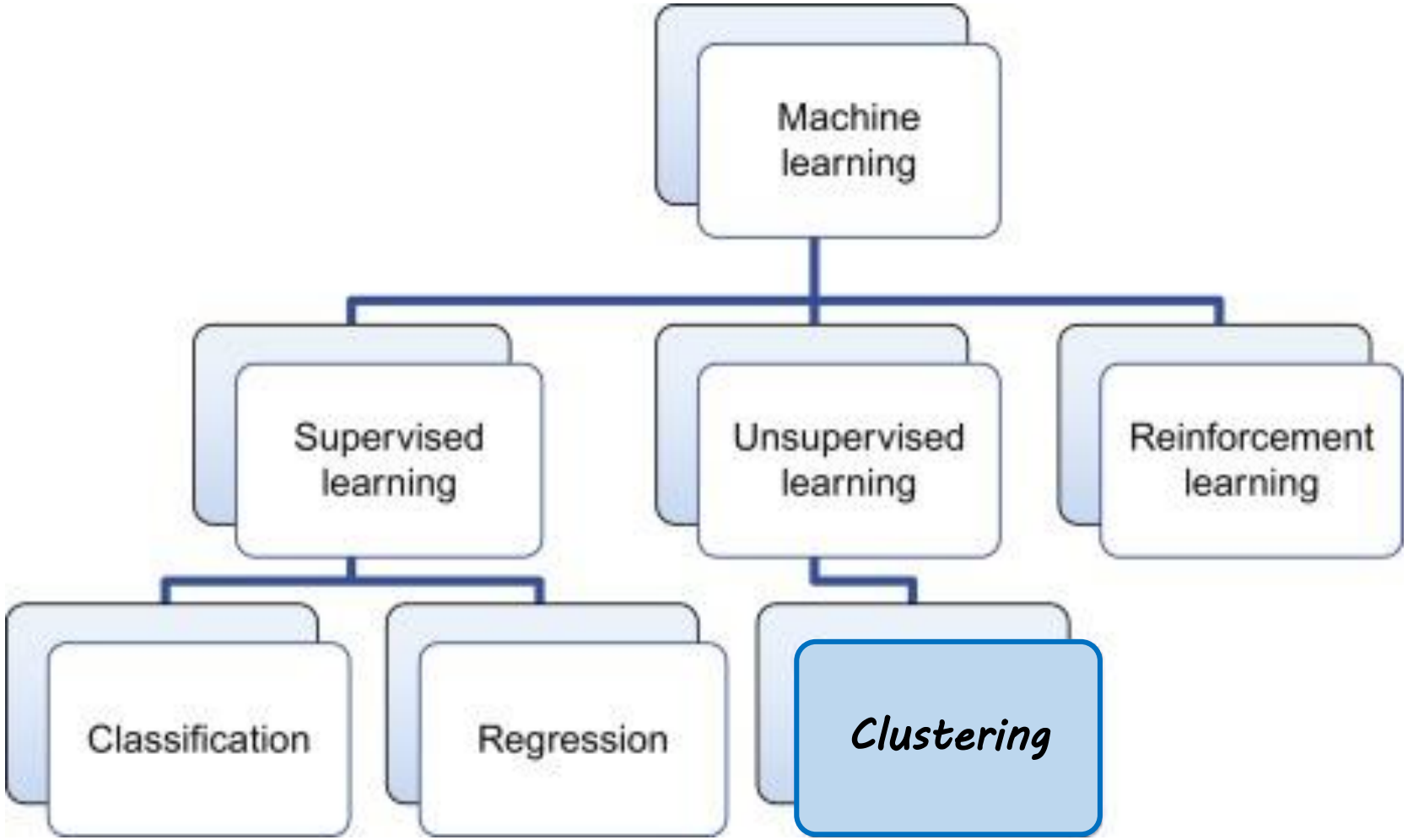
Hamid Sadeghi

nextera, FaceCup 1400 (2022)

Contents

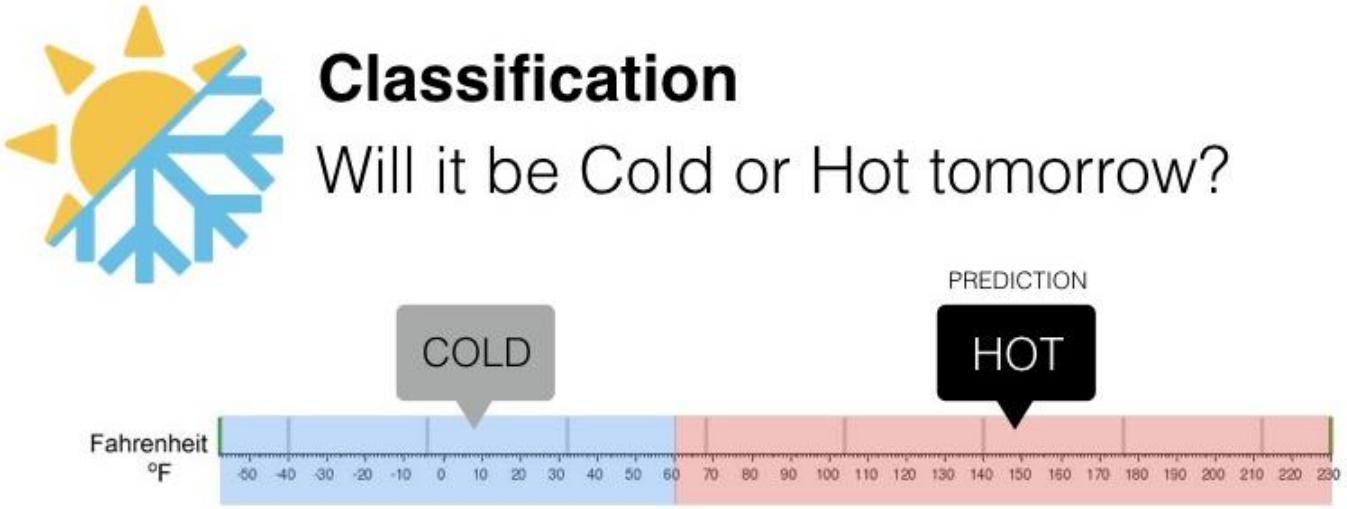
- Introduction
- Clustering Algorithms
- Evaluation
- Face Analysis

Introduction



Handbook of Statistics, Chapter 8: Machine Learning, Elsevier, 2018

Introduction (Classification vs Regression)

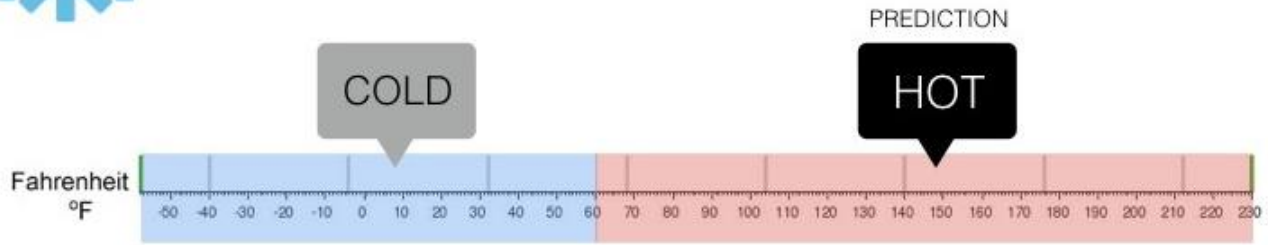


www.pinterest.com

Introduction (Classification vs Regression)

Classification

Will it be Cold or Hot tomorrow?

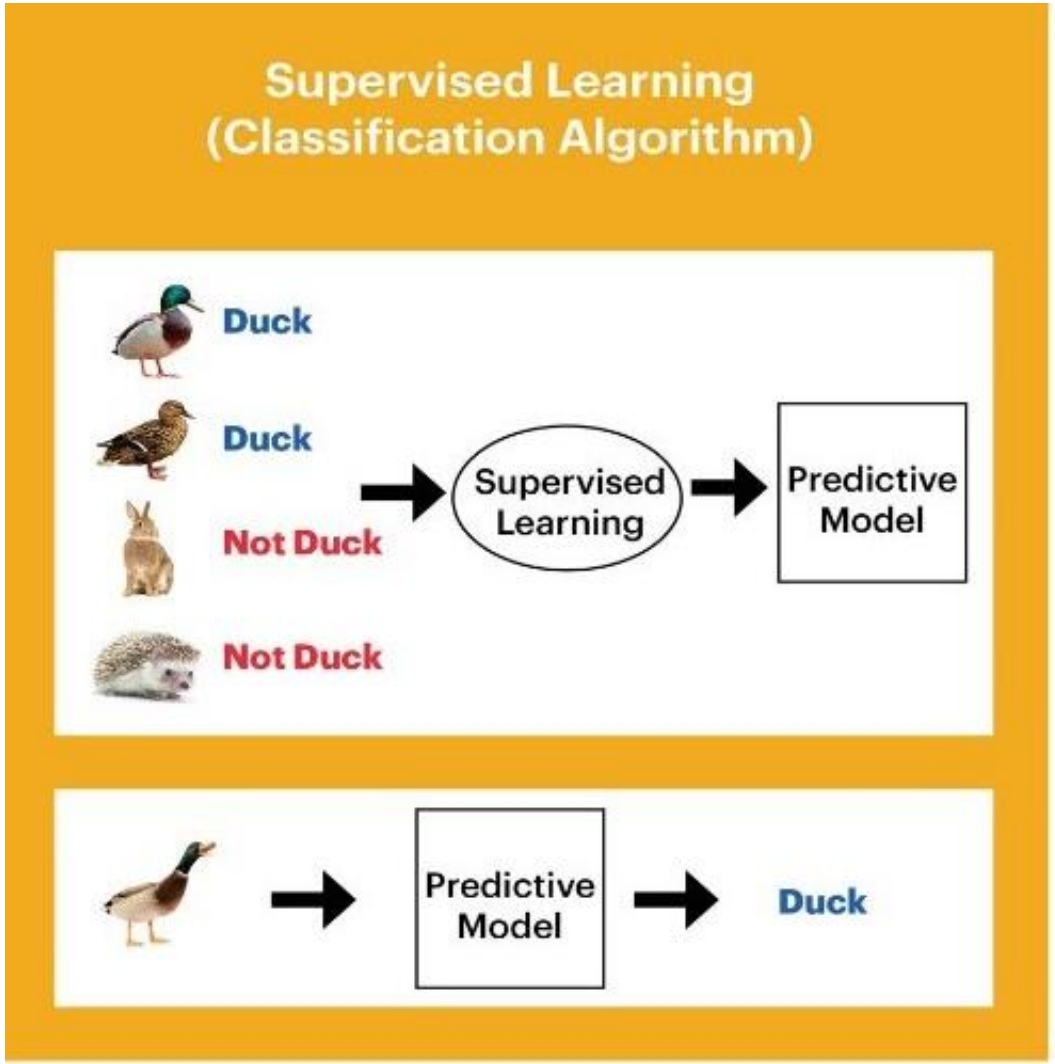


Regression

What is the temperature going to be tomorrow?

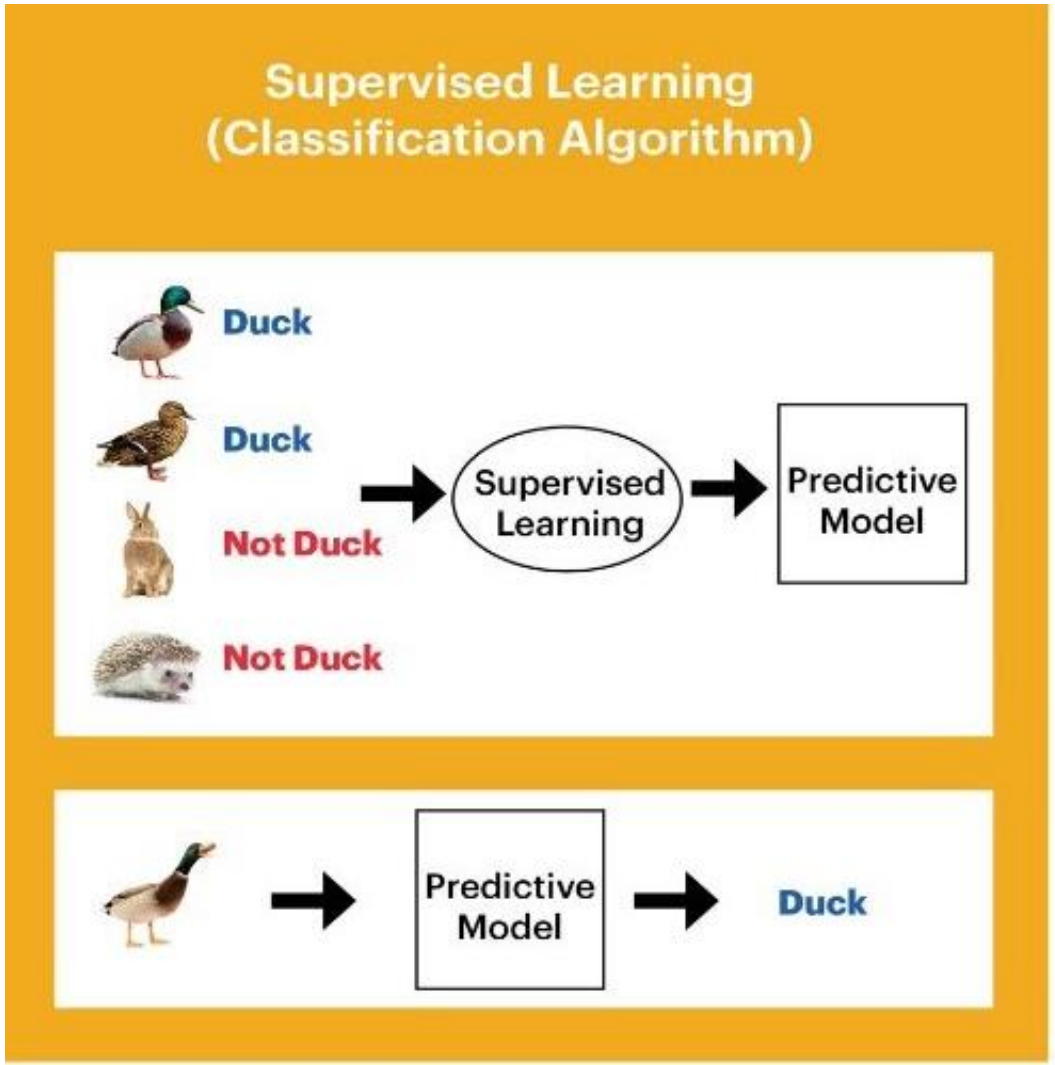
www.pinterest.com

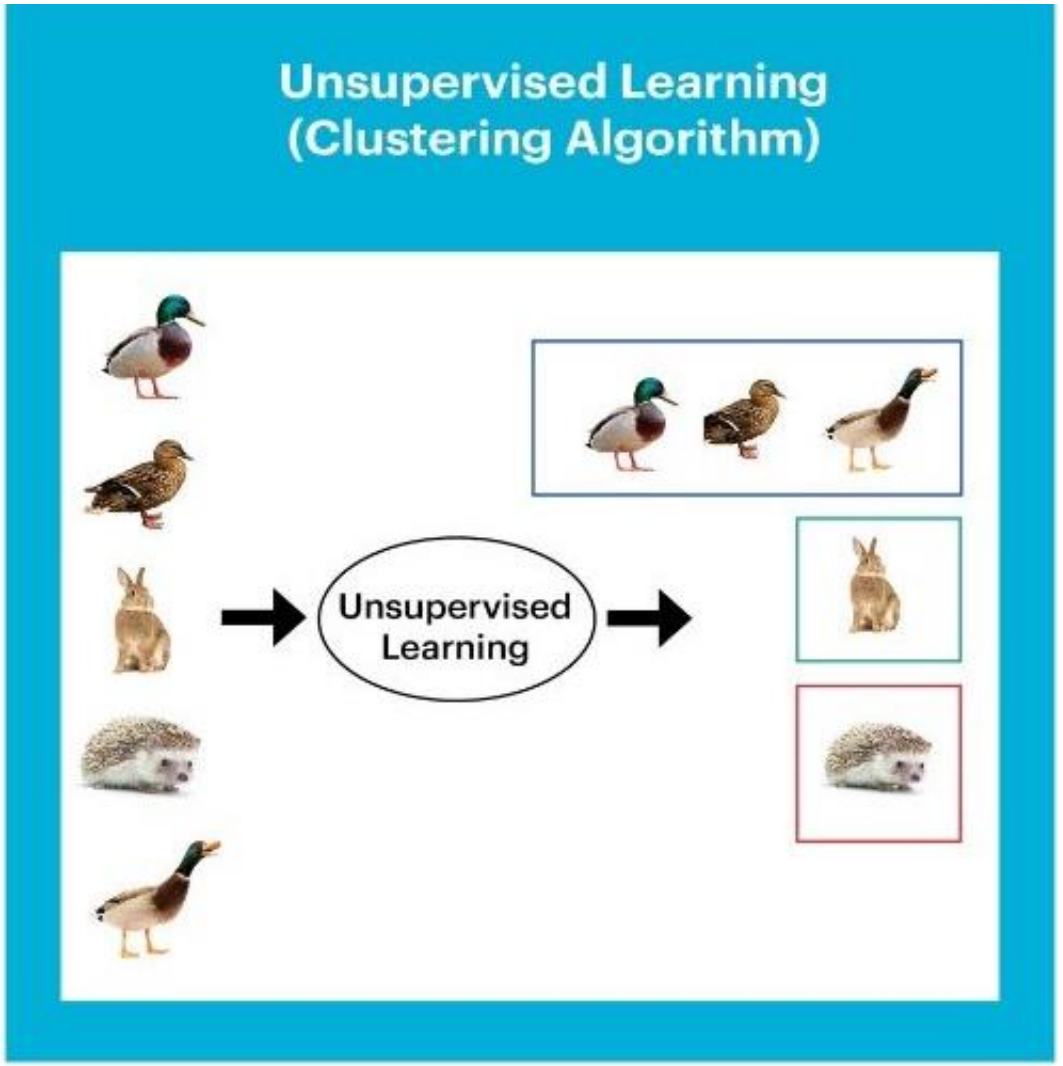
Introduction (Classification vs Clustering)



blog.westerndigital.com

Introduction (Classification vs Clustering)





blog.westerndigital.com

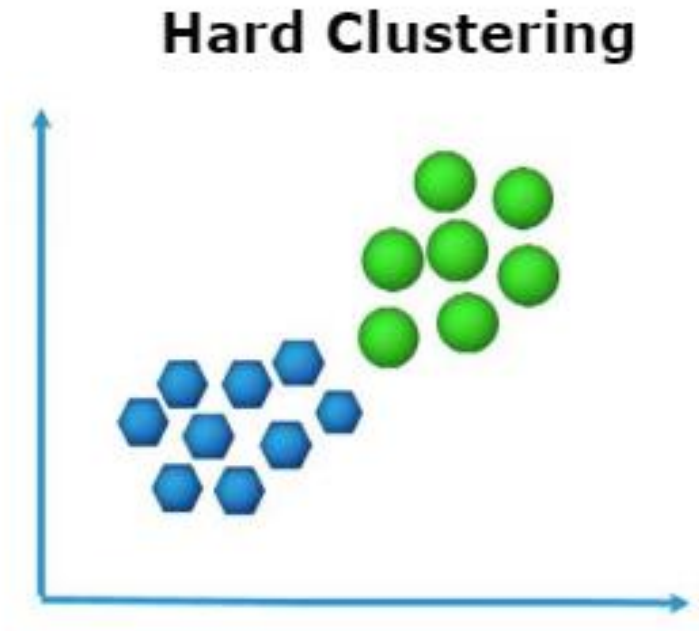
Introduction

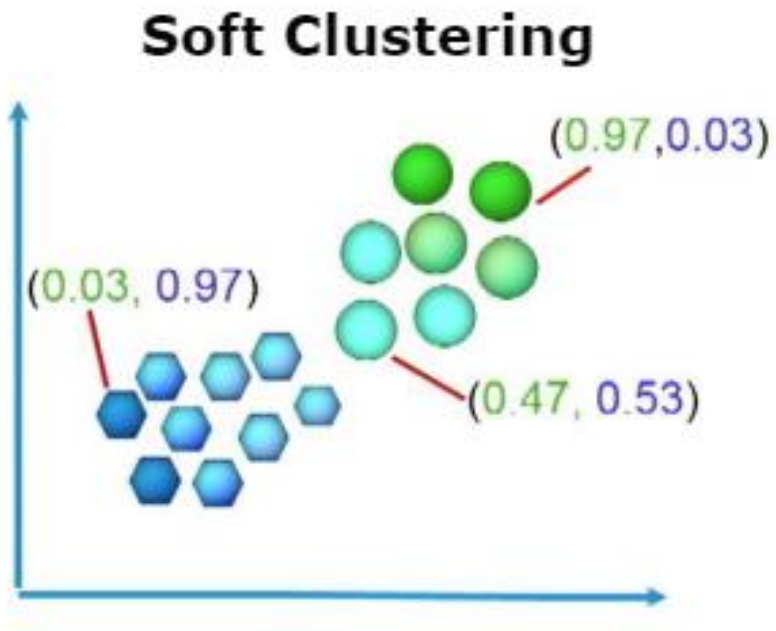
Clustering

Clustering Algorithms

- Introduction
- K-means Clustering
- DBSCAN Clustering
- Agglomerative Clustering

Clustering Algorithms: Introduction

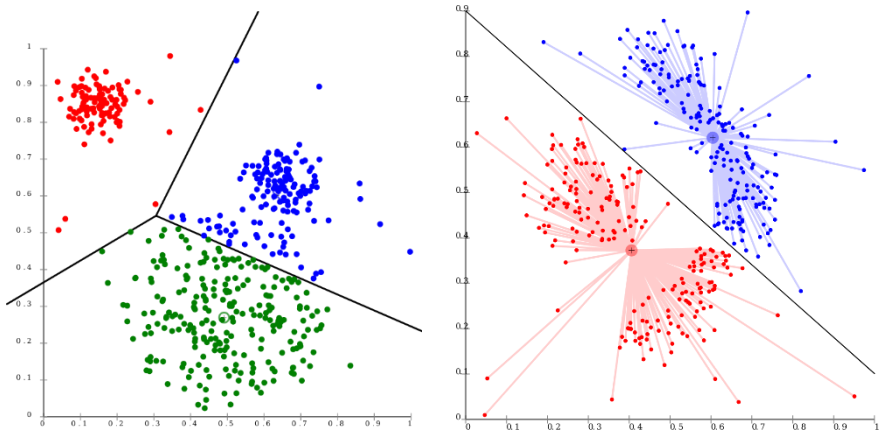




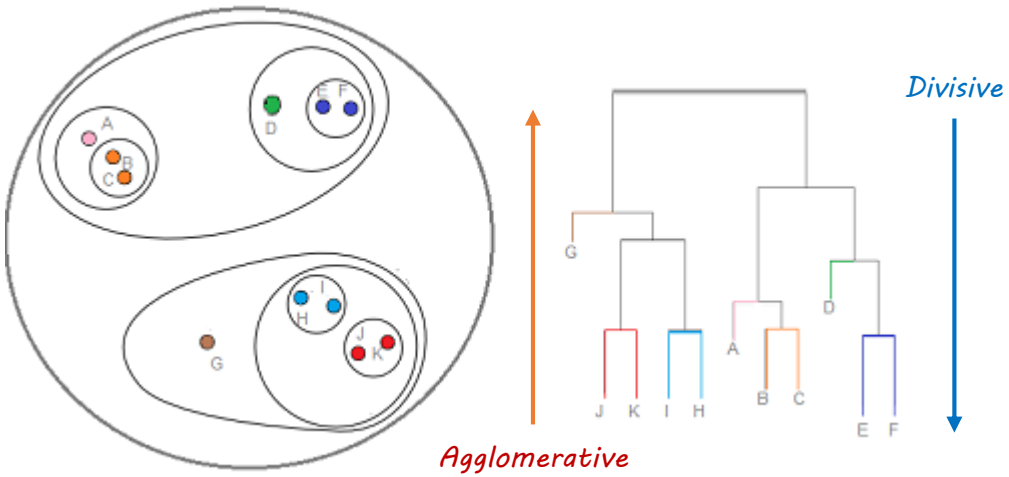
S. Kumar, towardsdatascience.com

Clustering Algorithms: Introduction

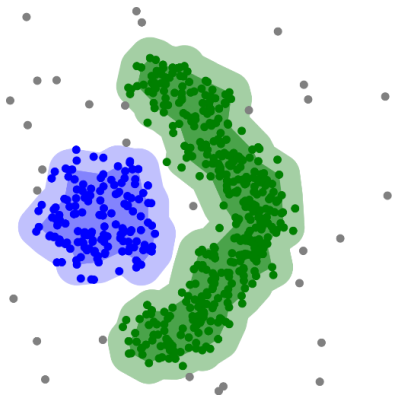
Centroid-based (Partitional) clustering



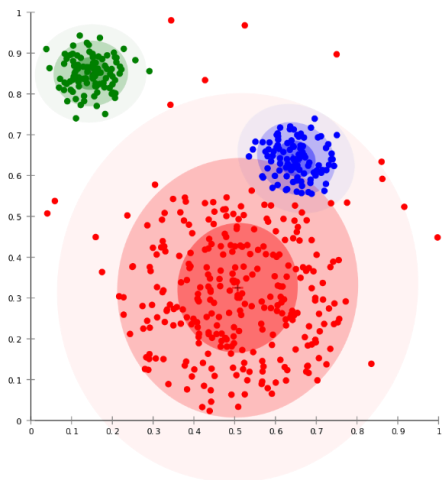
Connectivity-based (hierarchical) clustering



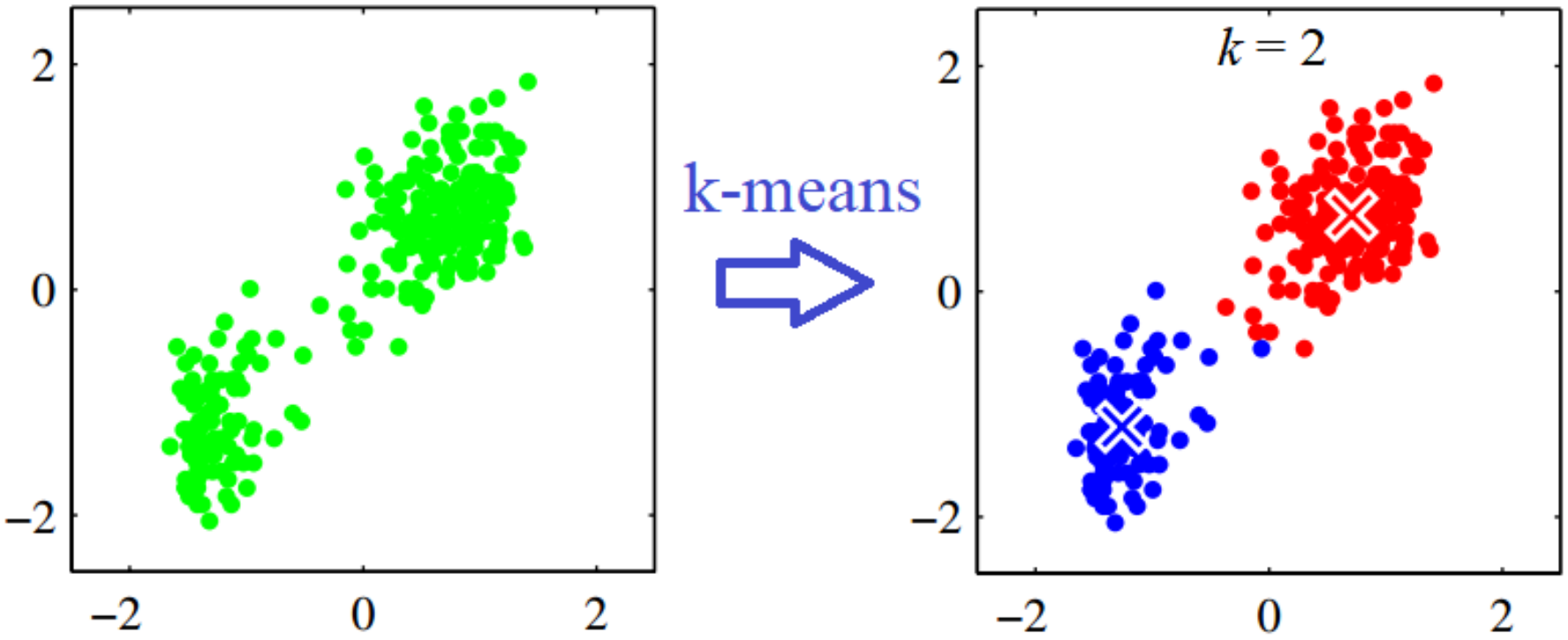
Density-based clustering



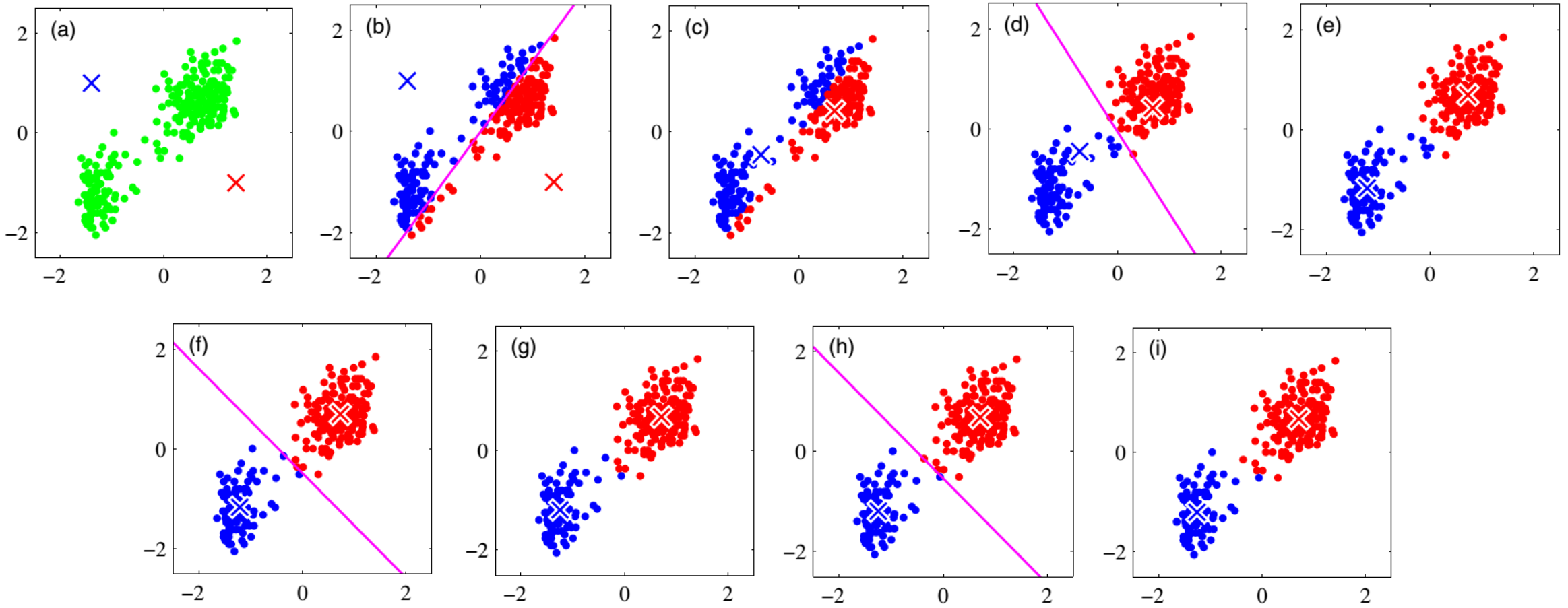
Distribution-based clustering



Clustering Algorithms: K-means

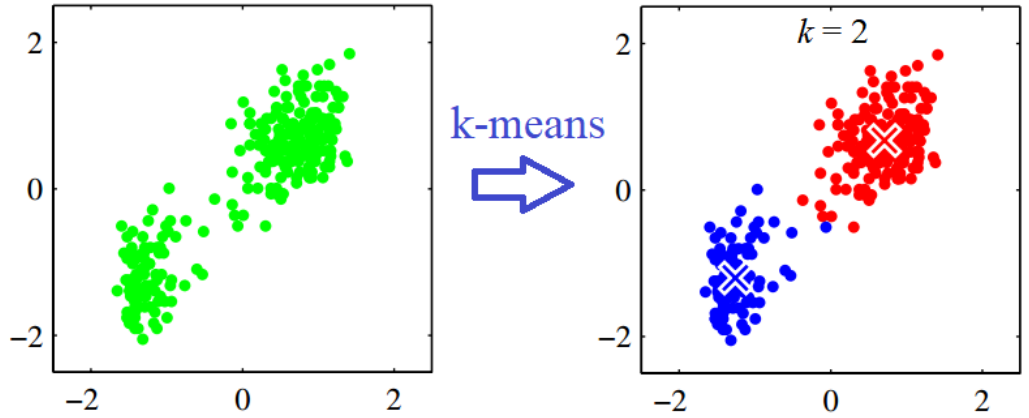


Clustering Algorithms: K-means



C. M. Bishop, Pattern Recognition and Machine Learning, Ch. 9: Mixture Models and EM

Clustering Algorithms: K-means



Algorithm 1 k-means algorithm

1: Specify the number k of clusters to assign.
2: Randomly initialize k centroids.
3: **repeat**
4: **expectation:** Assign each point to its closest centroid.
5: **maximization:** Compute the new centroid (mean) of each cluster.
6: **until** The centroid positions do not change.

realpython.com/k-means-clustering-python/

Clustering Algorithms: K-means

image segmentation and image compression

$K = 2$

each pixel: {R, G, B}

K = 2

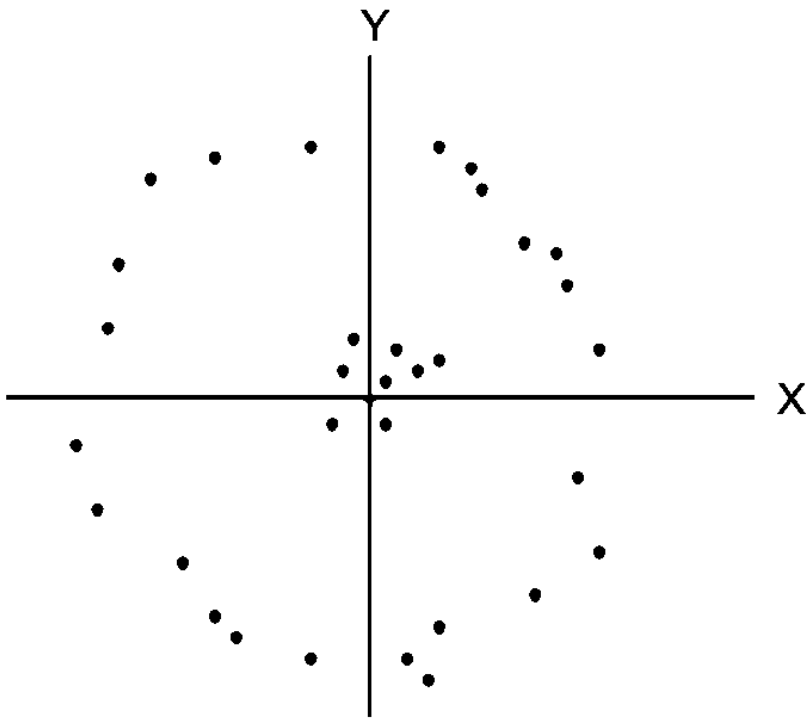
Original image

Clustering Algorithms: K-means

image segmentation and image compression

C. M. Bishop, Pattern Recognition and Machine Learning, Ch. 9: Mixture Models and EM

Clustering Algorithms: K-means

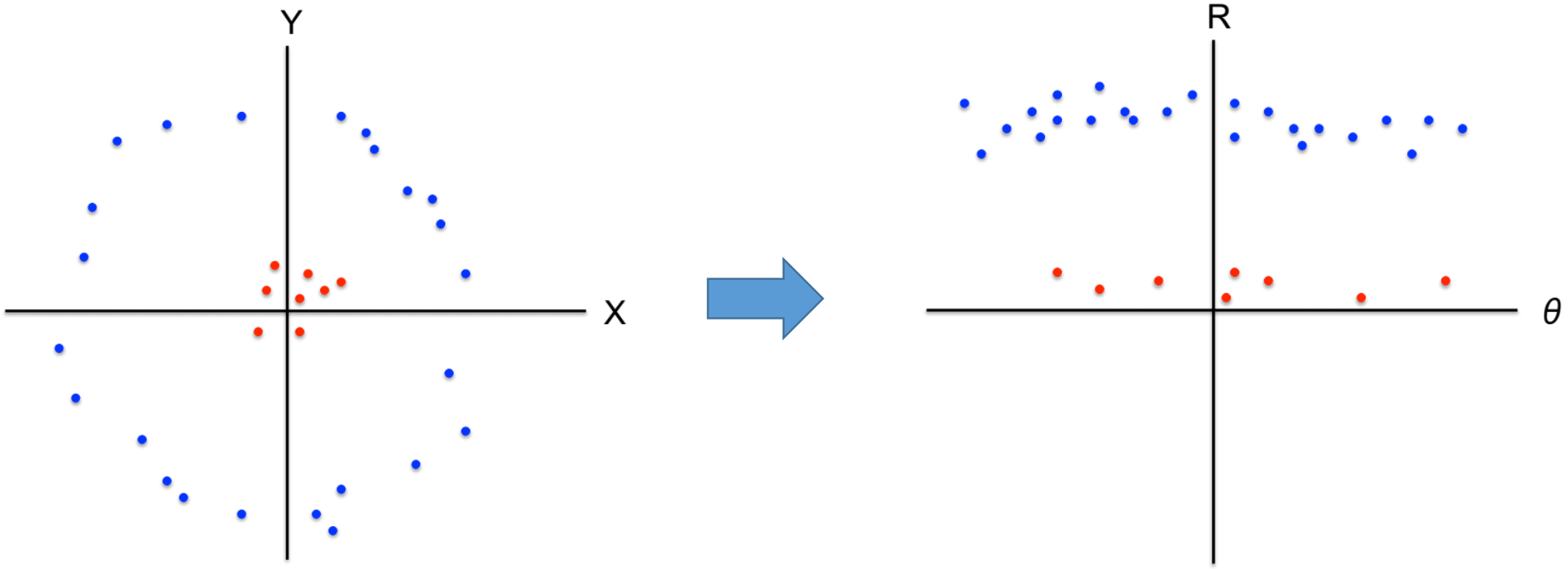


K-means not able to properly cluster

D. Sontag, http://people.csail.mit.edu/dsontag/courses/ml13/

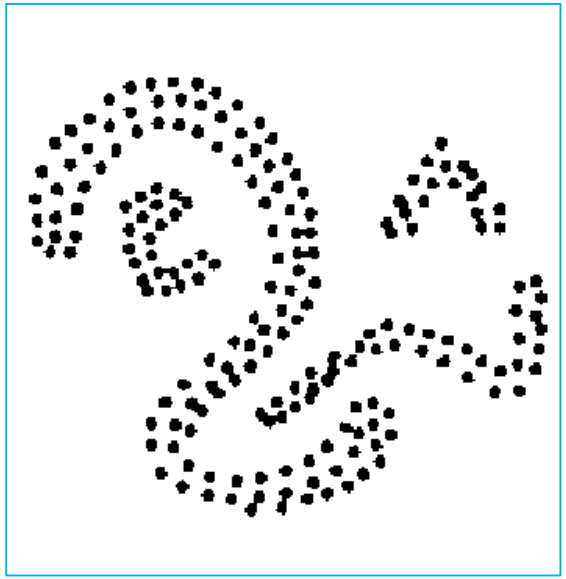
Clustering Algorithms: K-means

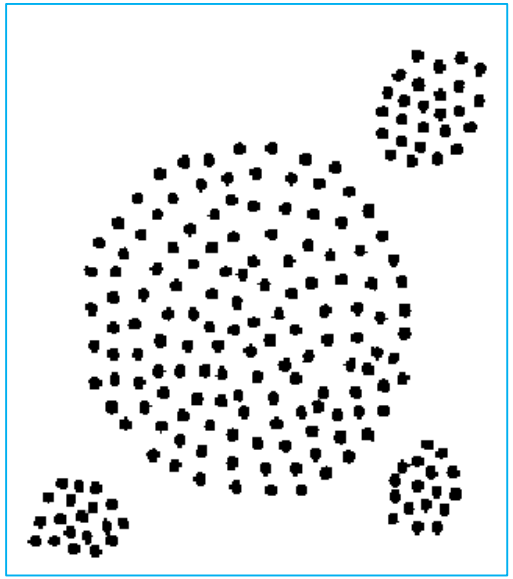
Changing the features (distance function)

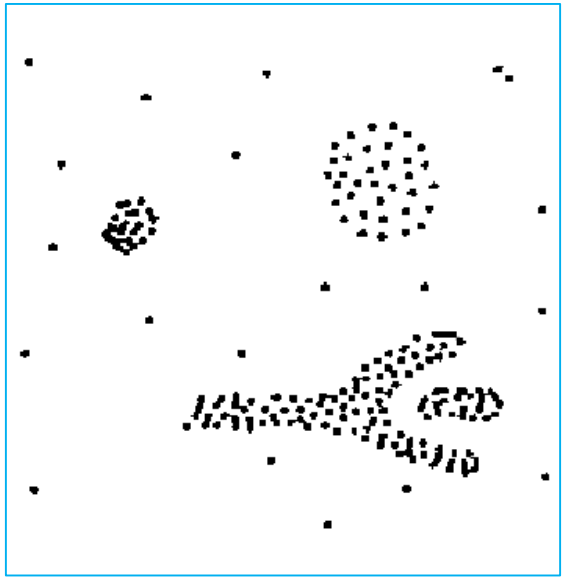


D. Sontag: http://people.csail.mit.edu/dsontag/courses/ml13/

Clustering Algorithms: K-means



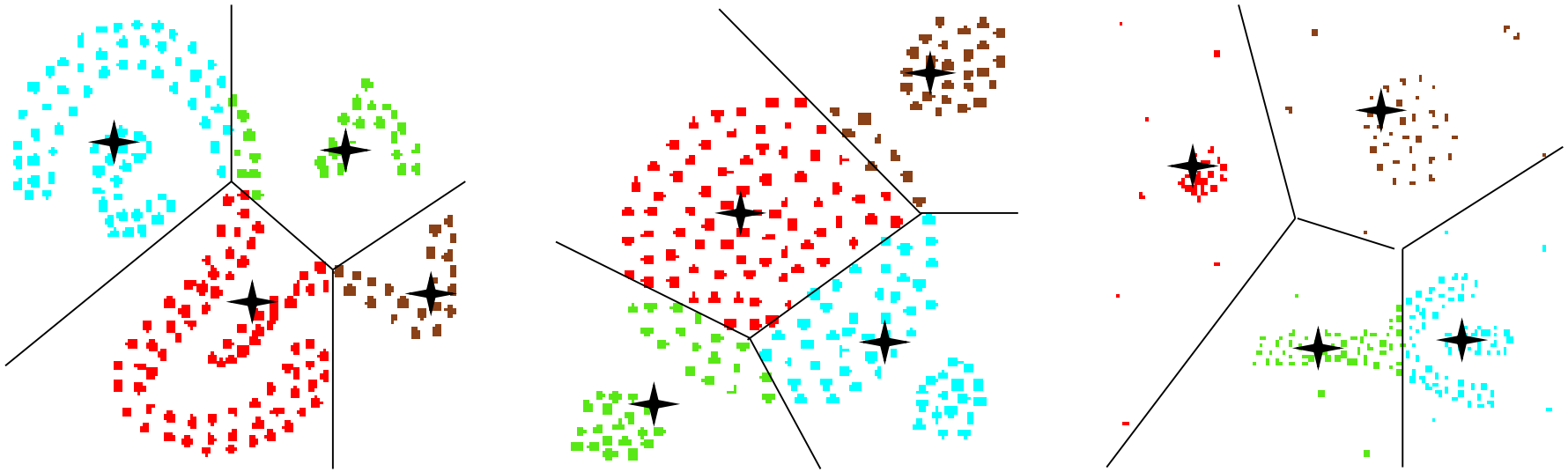




M. Ester, et al., A density-based algorithm for discovering clusters in large spatial databases with noise, 1996.

Clustering Algorithms: K-means

Results of k-medoids algorithm for k=4



(In contrast to the k-means algorithm, k-medoids chooses actual data points as centers·)

www.cse.buffalo.edu/faculty/azhang/cse601/density-based.ppt

Clustering Algorithms: K-means

- Implementations:

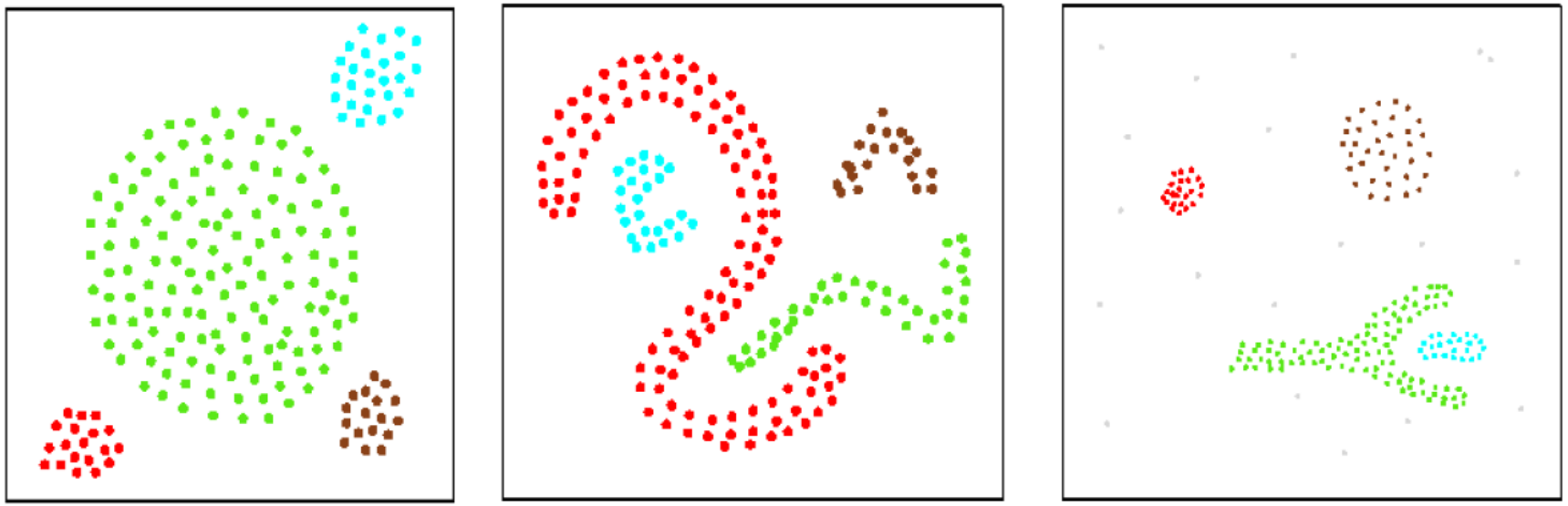
  GitHub https://github.com/hamidsadeghi68/face-clustering

- K-means:

  https://colab.research.google.com/github/hamidsadeghi68/face-clustering/blob/main/clustering_kmeans.ipynb

Clustering Algorithms: DBSCAN

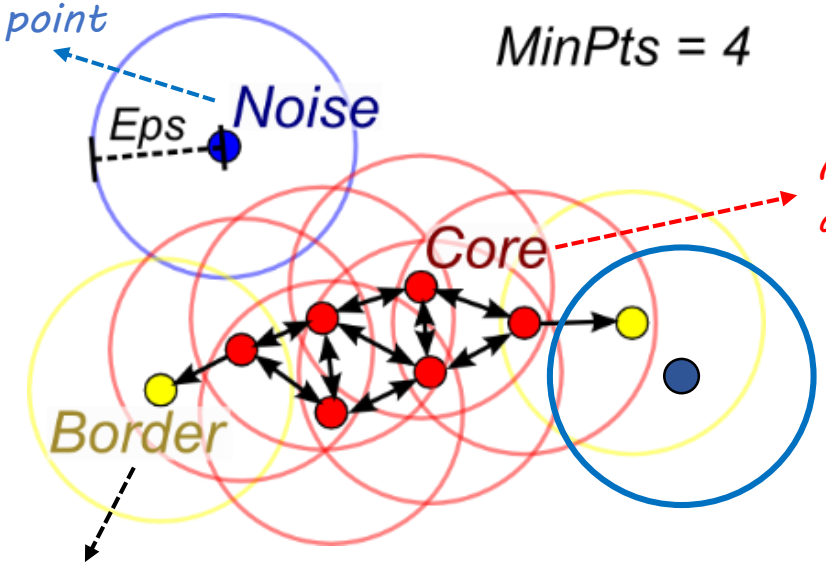
Results of a DBSCAN clustering



www.kdnuggets.com/2021/10/clustering-what-is-how-works.html

Clustering Algorithms: DBSCAN

not reachable from any core point

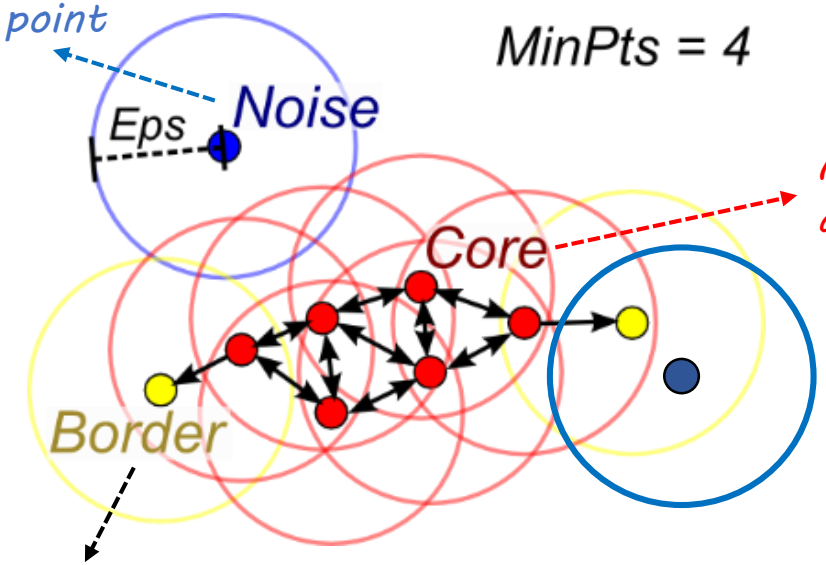


At least minPts points (including p) are within distance ε of it

The point is within distance ε from core point p

Clustering Algorithms: DBSCAN

not reachable from any core point



At least minPts points (including p) are within distance ε of it

The point is within distance ε from core point p

ALGORITHM 2: Abstract DBSCAN Algorithm

```
1 Compute neighbors of each point and identify core points      // Identify core points
2 Join neighboring core points into clusters                    // Assign core points
3 foreach non-core point do
4     Add to a neighboring core point if possible               // Assign border points
5     Otherwise, add to noise                                   // Assign noise points
```

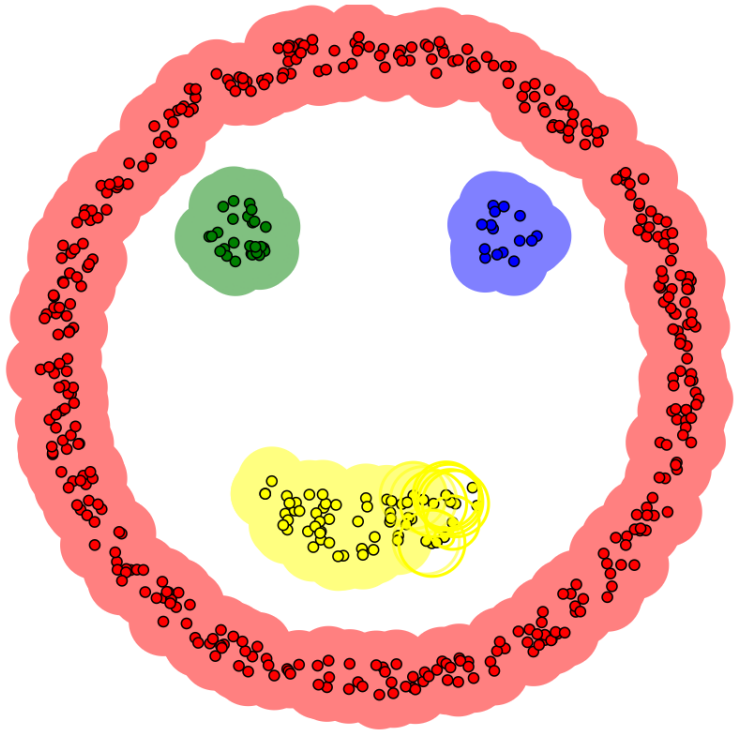
Clustering Algorithms: DBSCAN

```
DBSCAN (database: DB, distFunc, eps, minPts) {
    C ← 0                                                   /* Cluster counter */
    for each point P in DB {
        if label(P) ≠ undefined then continue               /* Previously processed in inner loop */
        Neighbors N ← RangeQuery(DB, distFunc, P, eps)      /* Find neighbors */
        if |N| < minPts then {                              /* Density check */
            label(P) ← Noise                                /* Label as Noise */
            continue
        }
        C ← C + 1                                           /* next cluster label */
        label(P) ← C                                        /* Label initial point */
        SeedSet S ← N \ {P}                                 /* Neighbors to expand */
        for each point Q in S {                             /* Process every seed point Q */
            if label(Q) = Noise then label(Q) ← C           /* Change Noise to border point */
            if label(Q) ≠ undefined then continue           /* Previously processed (e.g., border point) */
            label(Q) ← C                                    /* Label neighbor */
            Neighbors N ← RangeQuery(DB, distFunc, Q, eps)  /* Find neighbors */
            if |N| ≥ minPts then {                          /* Density check (if Q is a core point) */
                S ← S ∪ N                                   /* Add new neighbors to seed set */
            }
        }
    }
}
```

Clustering Algorithms: DBSCAN

- Online visualization:

  https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/



epsilon = 1.00
minPoints = 3

Clustering Algorithms: DBSCAN

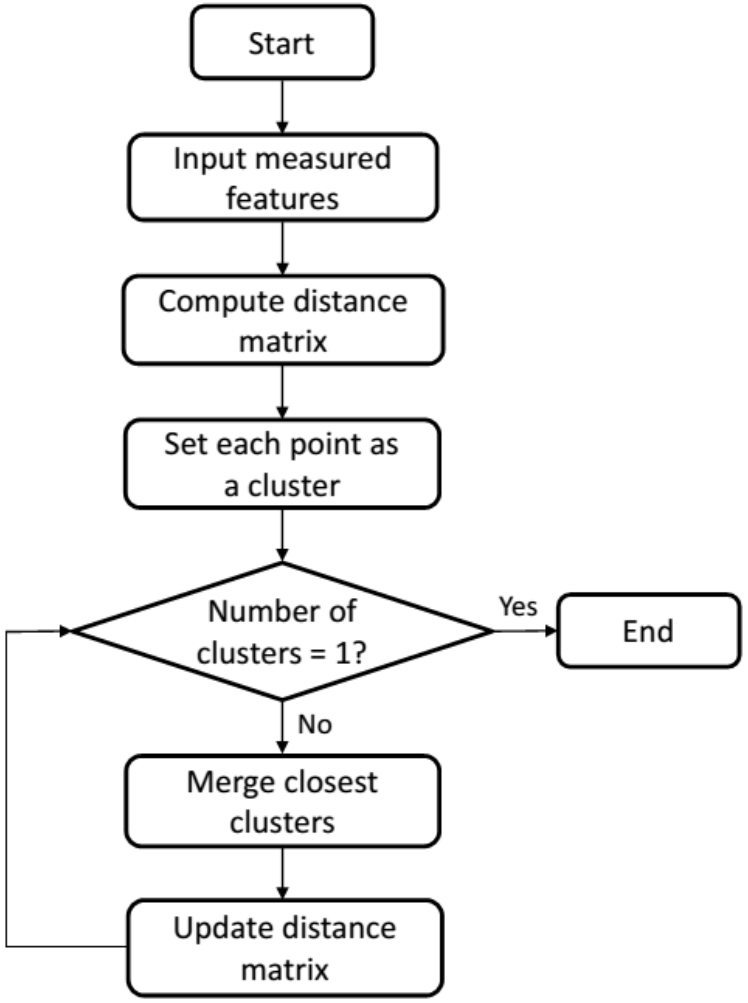
- *implementation:*

https://colab.research.google.com/github/hamidsadeghi68/face-clustering/blob/main/clustering_dbscan.ipynb

Clustering Algorithms: Agglomerative

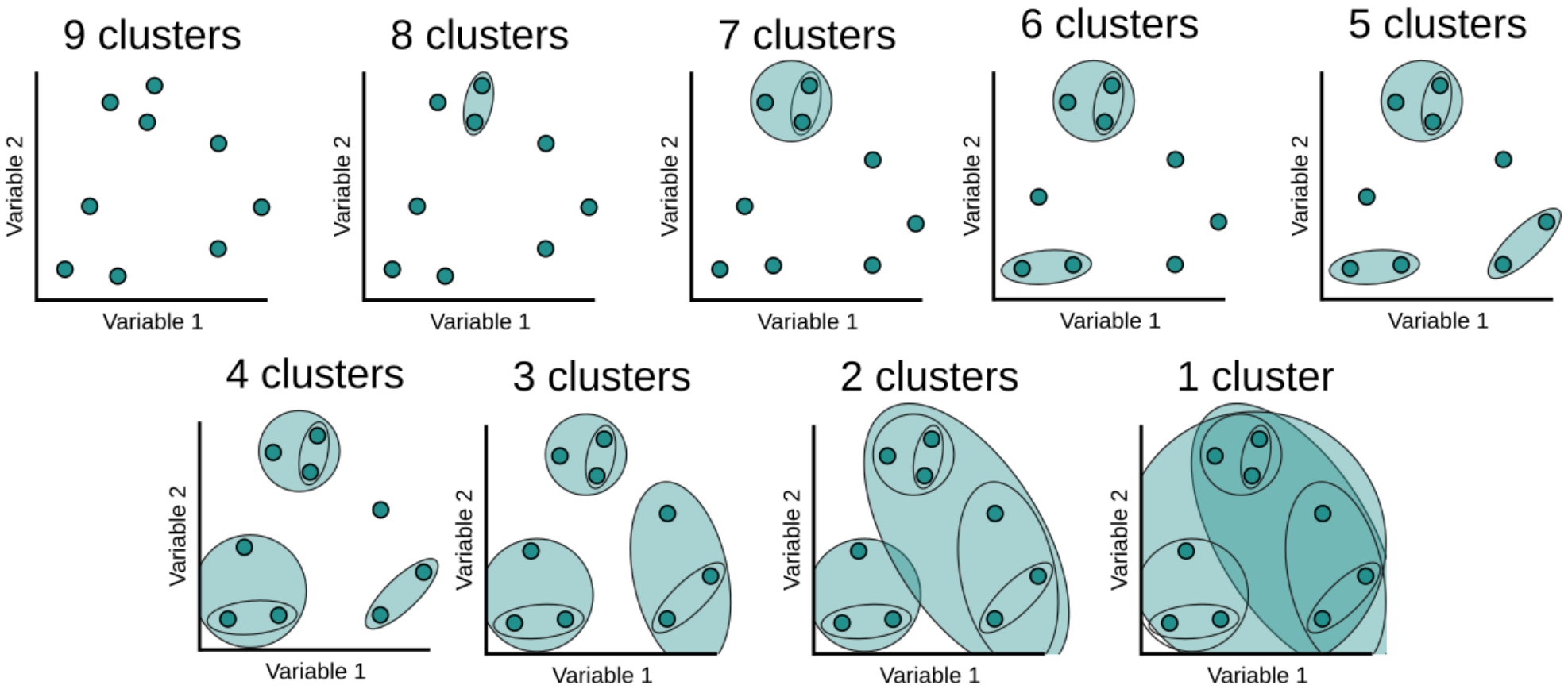
- each observation starts in its own cluster
- pairs of clusters are merged

Clustering Algorithms: Agglomerative



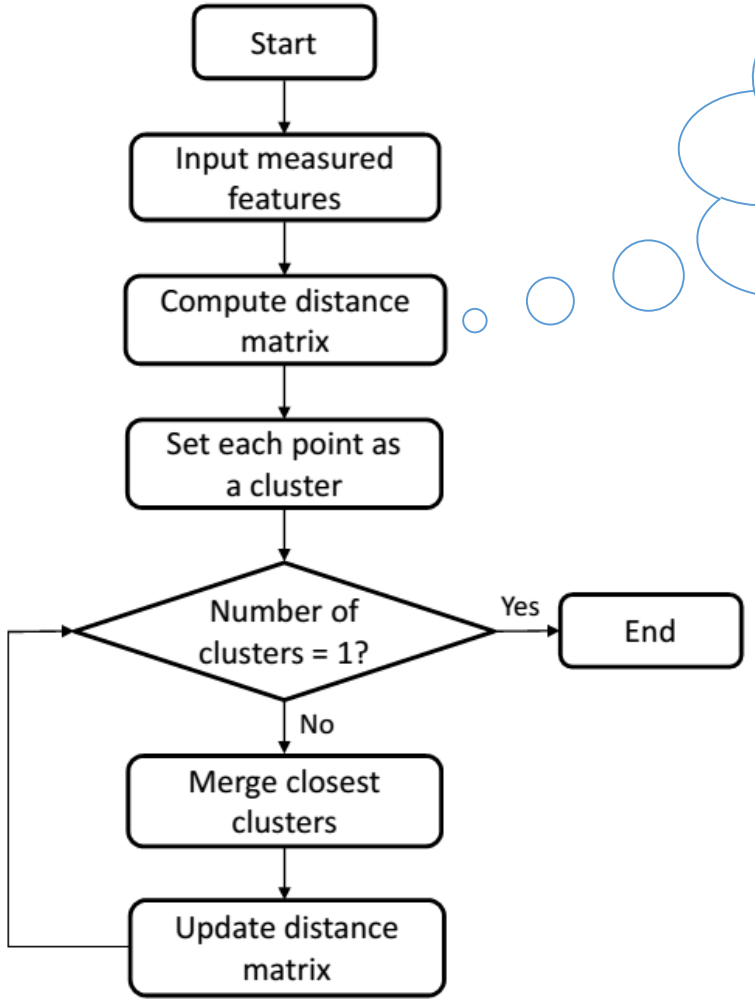
M. Silva, et al. Mechanical Systems and Signal Processing, 2017

Clustering Algorithms: Agglomerative



Hefin I. Rhys, Machine Learning with R, the tidyverse, and mlr, Ch. 17, 2020

Clustering Algorithms: Agglomerative



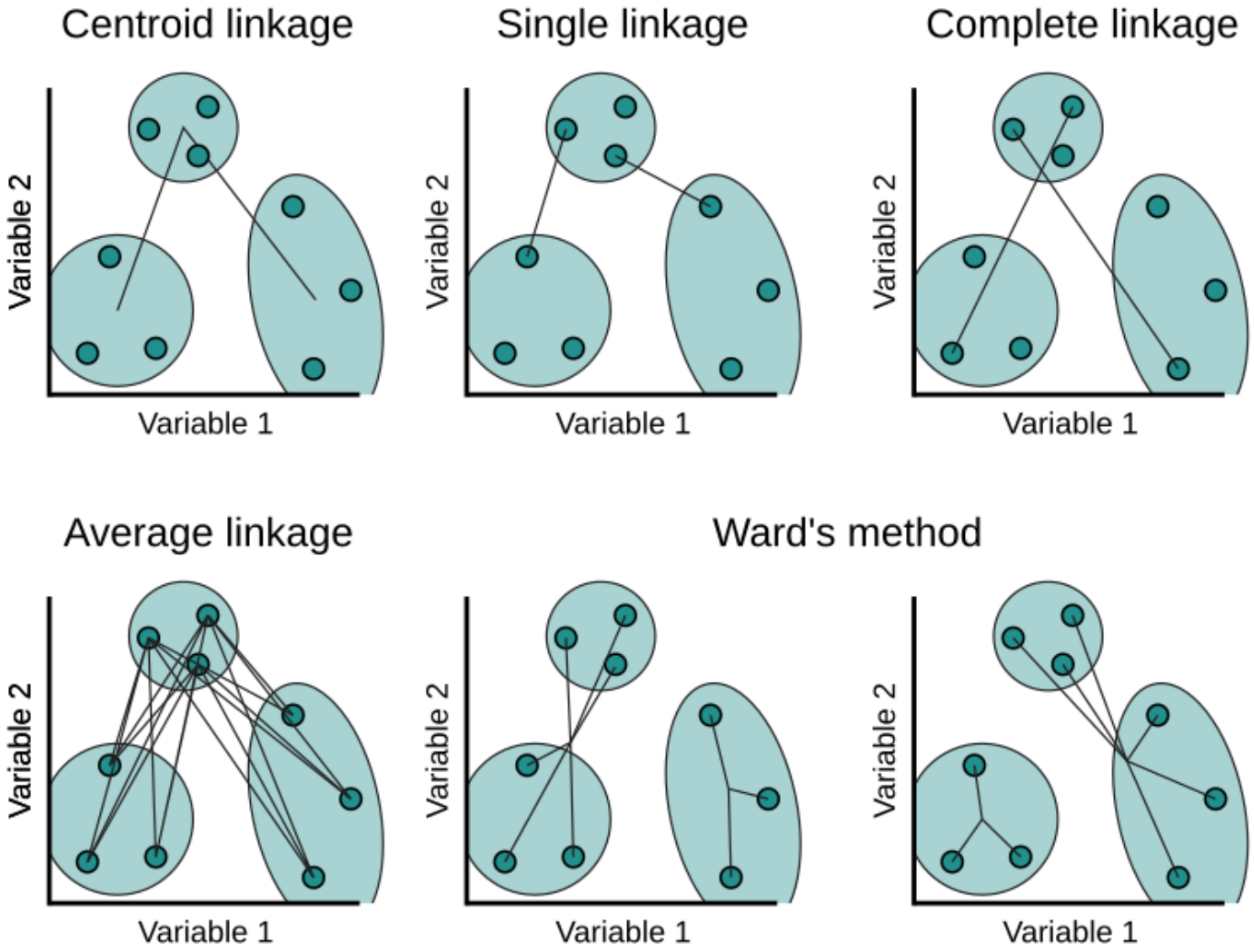
- ward
- average
- complete or maximum
- single or minimum
- centroid

M. Silva, et al. Mechanical Systems and Signal Processing, 2017

Clustering Algorithms: Agglomerative

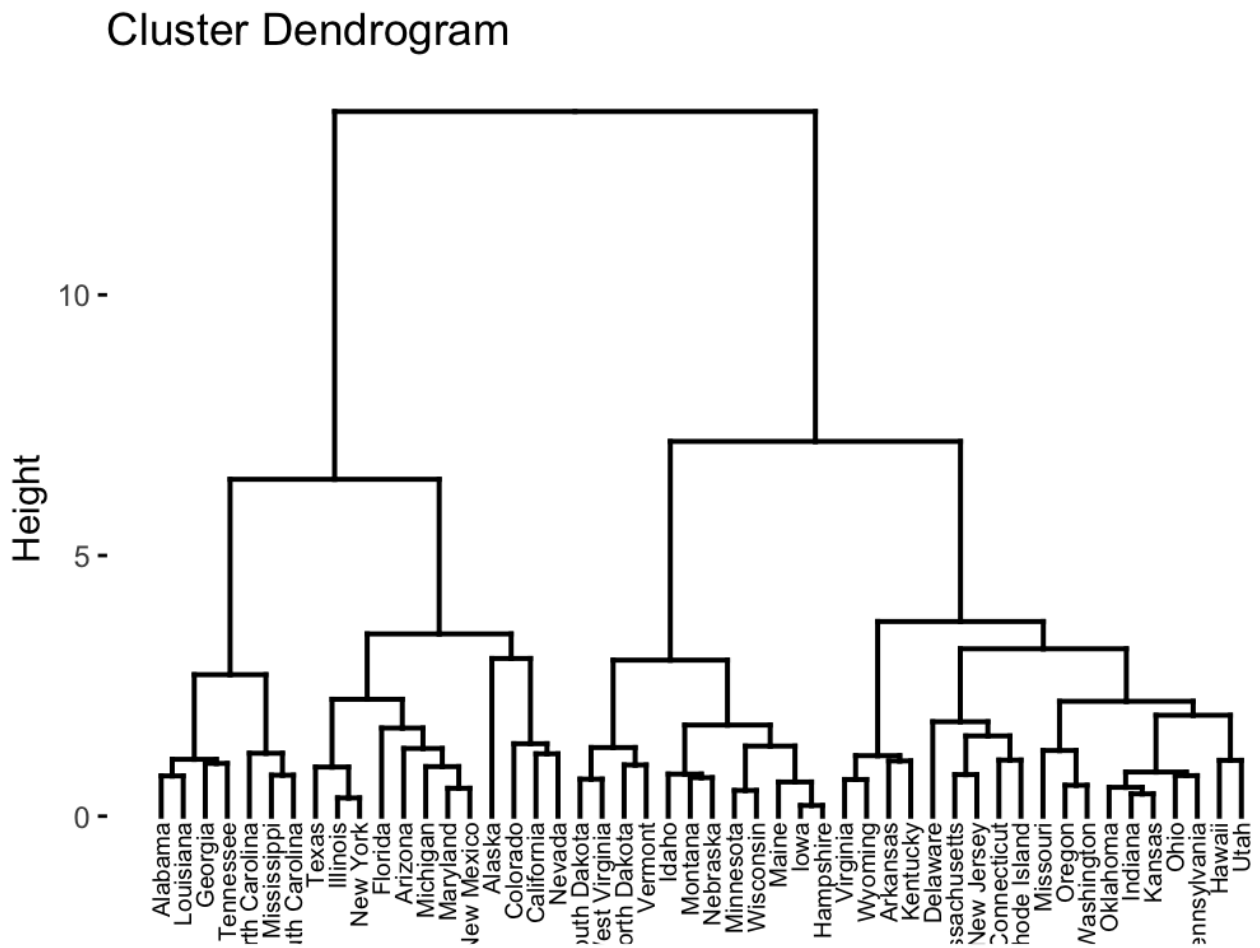
- '**ward**' minimizes the variance of the clusters being merged.
- '**average**' uses the average of the distances of each observation of the two sets.
- '**complete**' or '**maximum**' linkage uses the maximum distances between all observations of the two sets.
- '**single**' or '**minimum**' uses the minimum of the distances between all observations of the two sets.
- '**centroid**': uses the distance between the centroid for cluster 1 (a mean vector of length p variables) and the centroid for cluster 2

Clustering Algorithms: Agglomerative



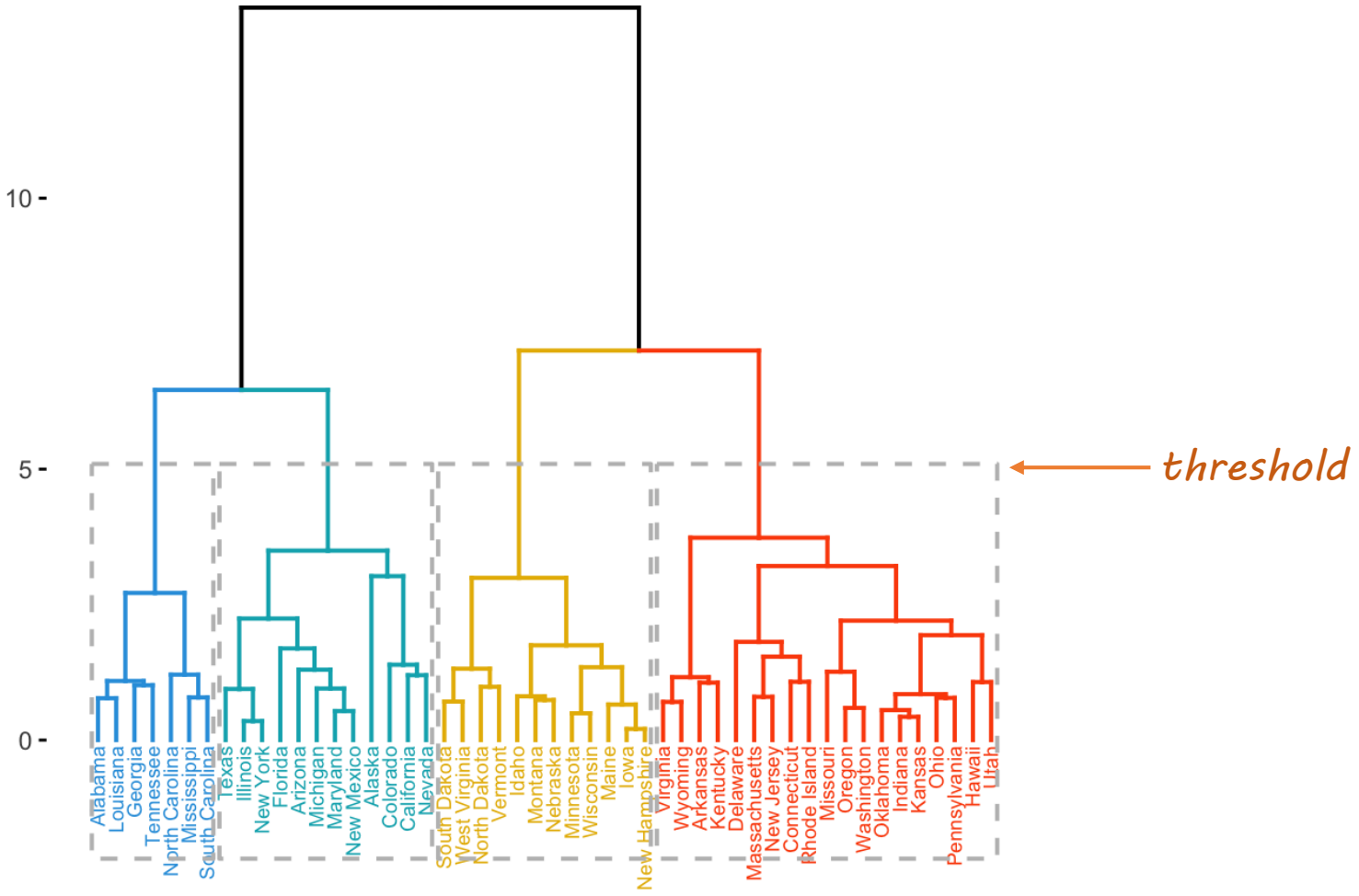
Hefin I. Rhys, Machine Learning with R, the tidyverse, and mlr, Ch. 17, 2020

Clustering Algorithms: Agglomerative



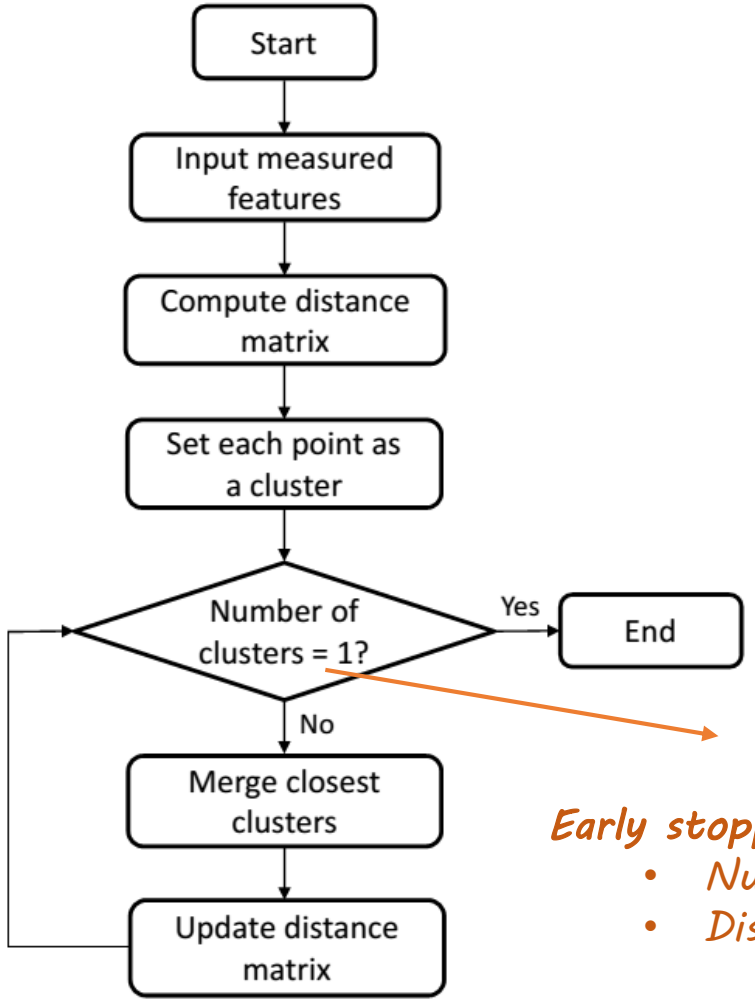
https://www.datanovia.com/en/lessons/agglomerative-hierarchical-clustering/

Clustering Algorithms: Agglomerative



https://www.datanovia.com/en/lessons/agglomerative-hierarchical-clustering/

Clustering Algorithms: Agglomerative



Full tree

Early stopping can be performed based on:

- Num. of clusters
- Distance/similarity criteria

M. Silva, et al. Mechanical Systems and Signal Processing, 2017

Clustering Algorithms: Agglomerative

- *implementation:*

https://colab.research.google.com/github/hamidsadeghi68/face-clustering/blob/main/clustering_agglomerative.ipynb