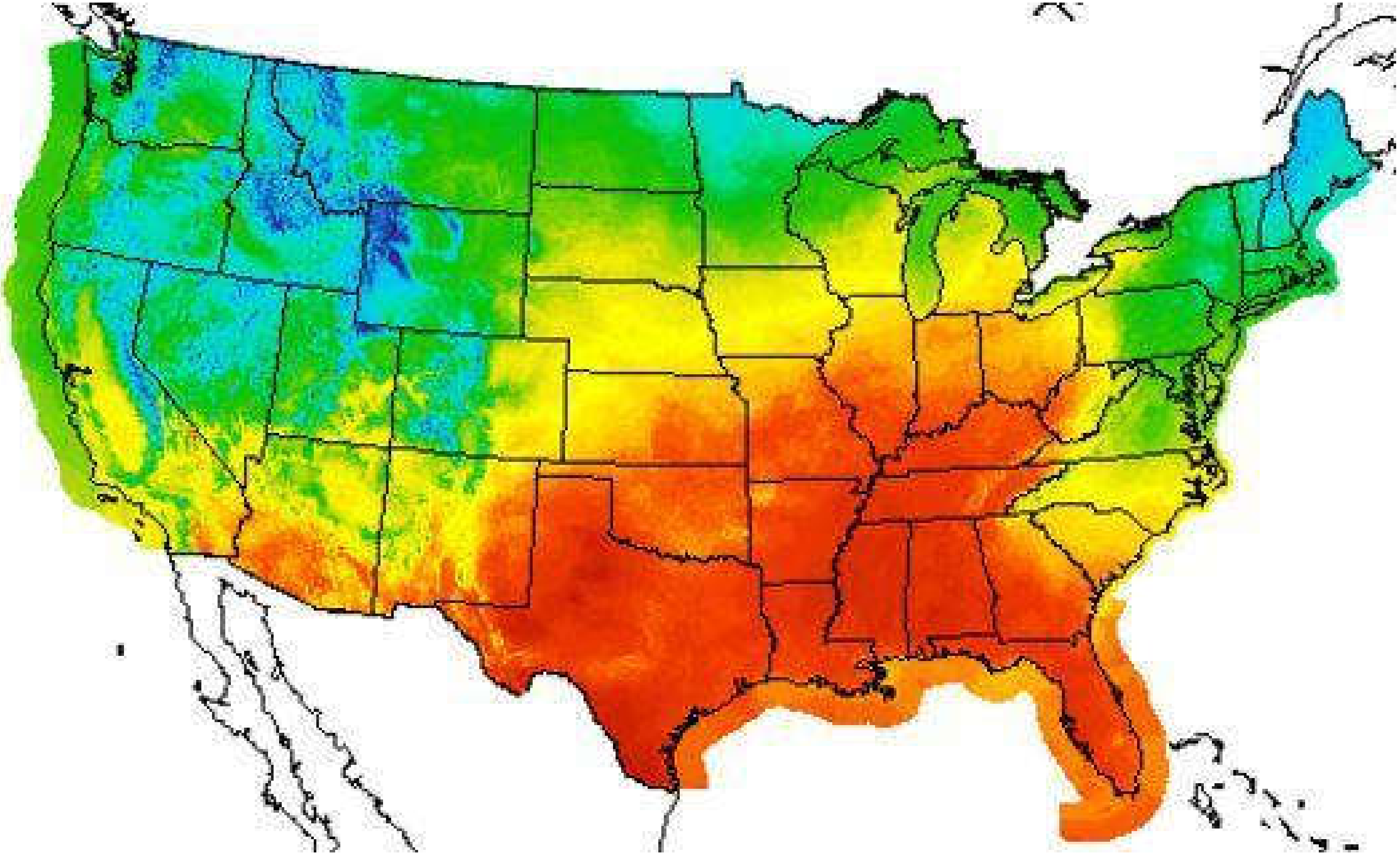
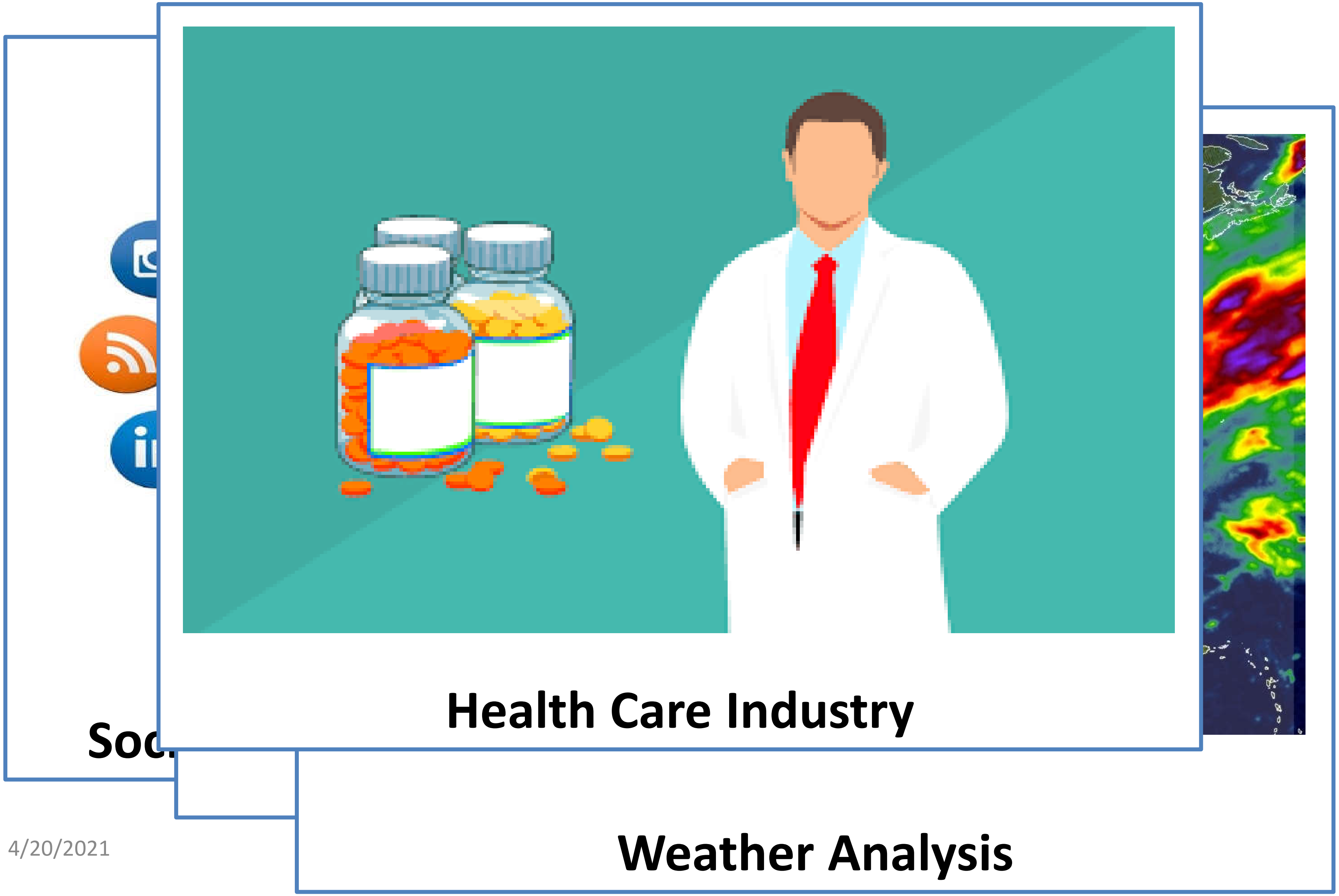


Clustering Algorithms

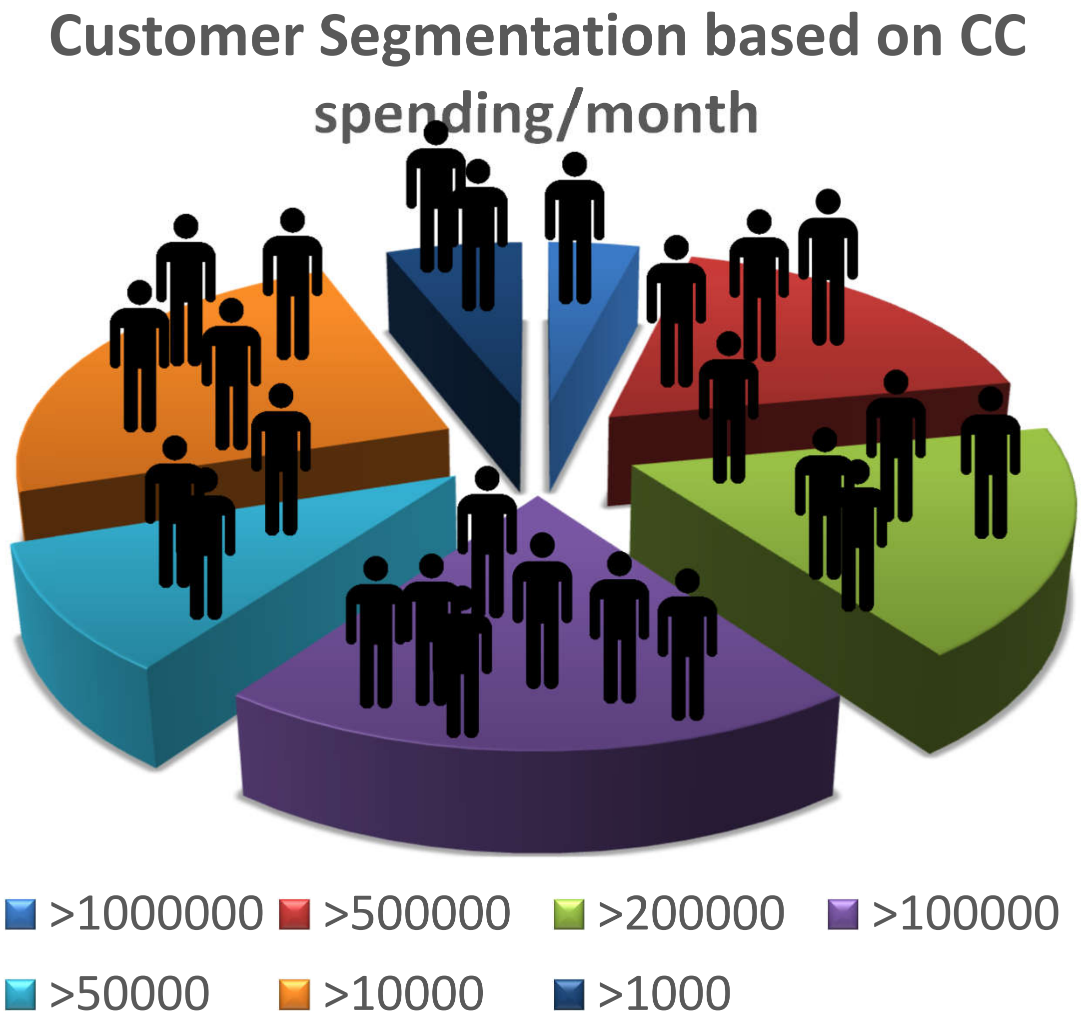
Machine Learning Algorithms

Supervised		Unsupervised		Other
Regression	Classification	Clustering	Association Rule Mining	Reinforcement Algorithms

Applications of Clustering…



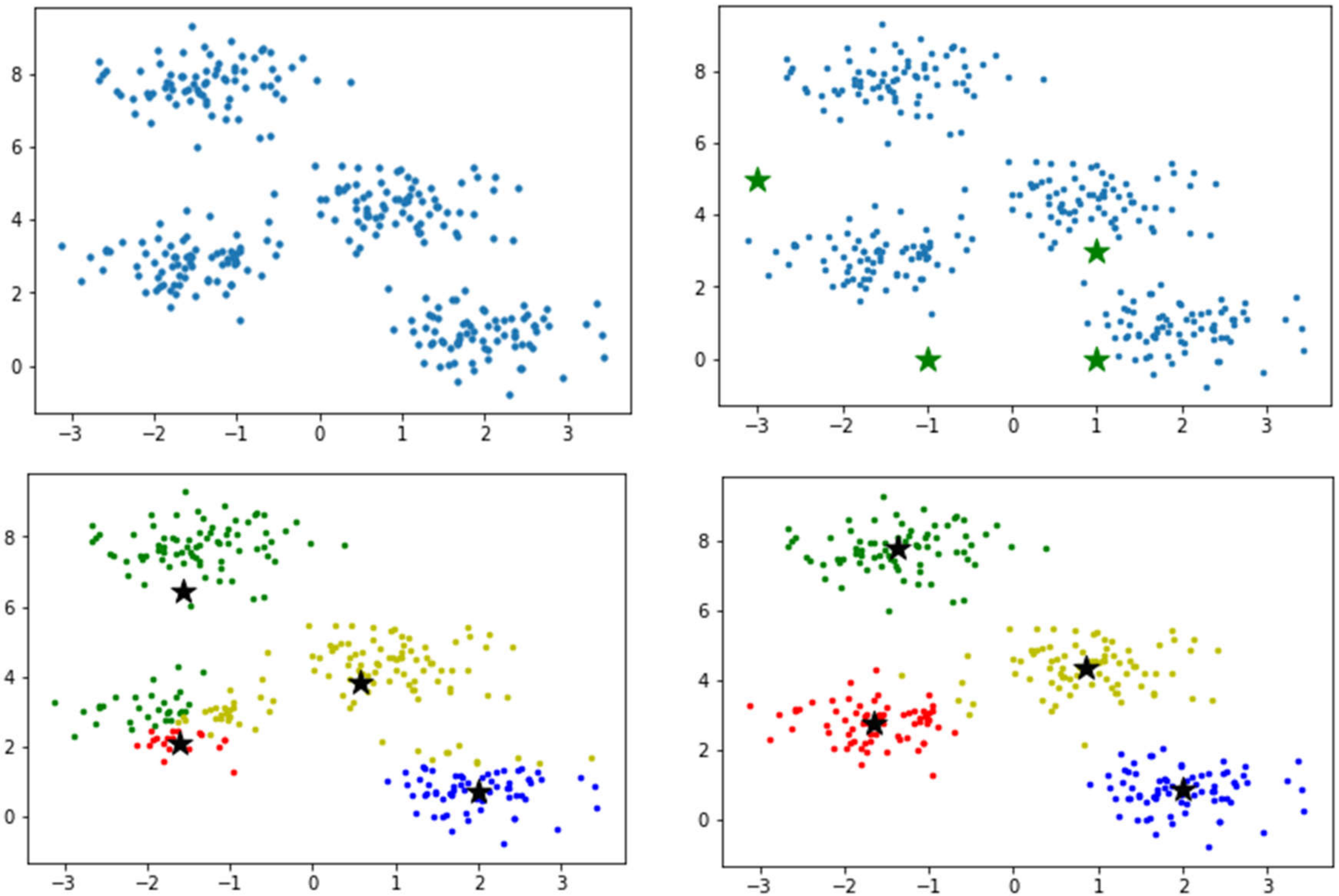
Applications of Clustering

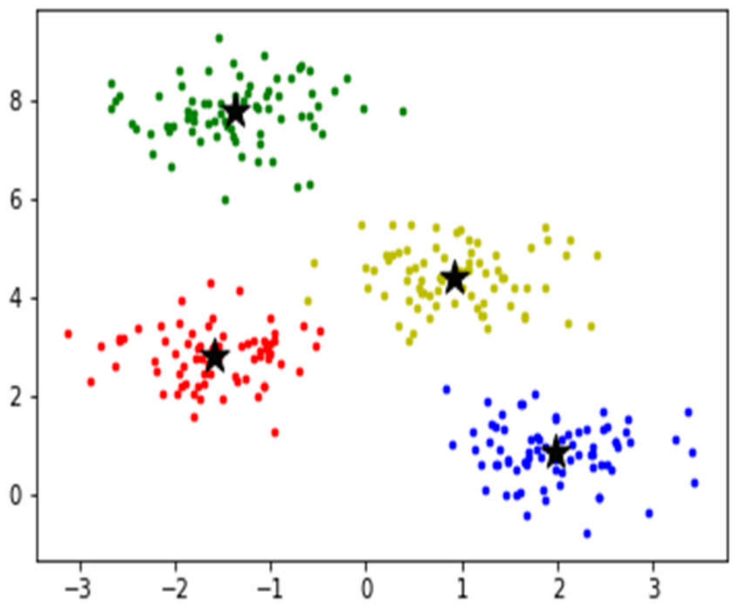


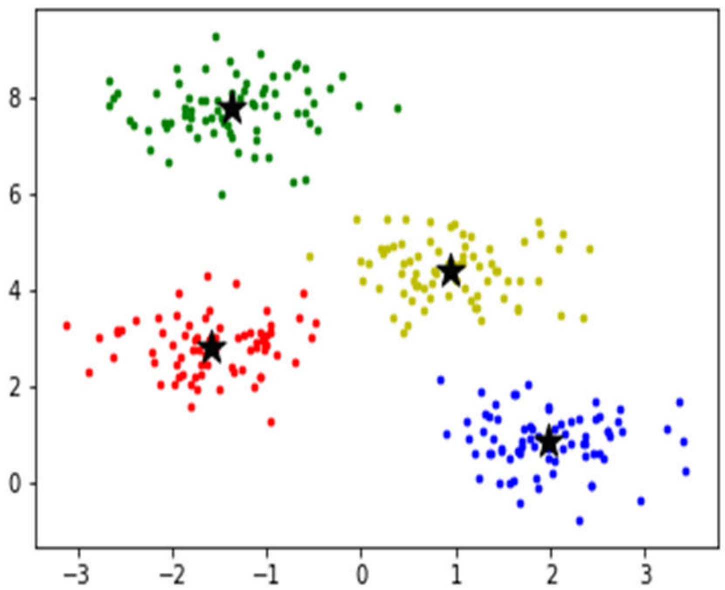
Clustering Algorithms

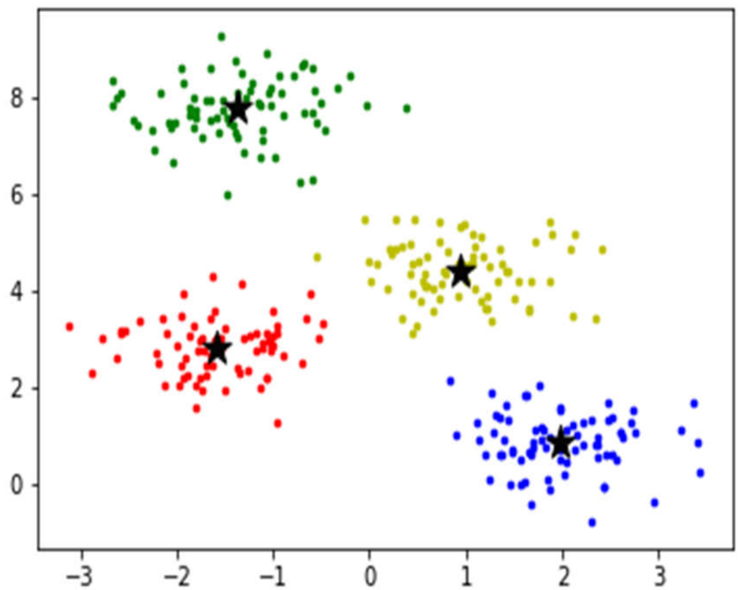
- **Unsupervised Learning:** Data Labels are not known
- Properties of dataset decide the clusters
- Popular Clustering Algorithms
  - Centroid Based
    - K-Means
    - Mean-shift
    - EM
  - Density Based Algorithms
    - DBSCAN
  - Hierarchical
    - Agglomerative

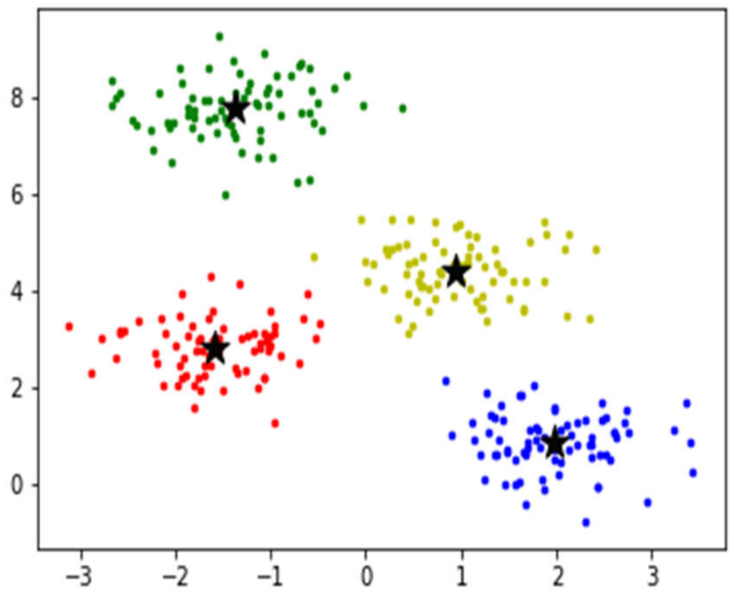
K-Means Clustering – How it works…




8
6
4
2
0
−3 −2 −1 0 1 2 3


8
6
4
2
0
−3 −2 −1 0 1 2 3


8
6
4
2
0
−3 −2 −1 0 1 2 3


8
6
4
2
0
−3 −2 −1 0 1 2 3

K-Means Algorithm

- Input:
  - Number of Clusters $K \in \{+ve\ odd\ Integer\}$
  - Dataset: $\{x^1, x^2, x^3, \dots \dots \dots \dots, x^m\}$
  - $Each\ xi\ \in\ \mathbb{R}^d$

Randomly Initialize K cluster centers $\mu_1, \mu_2, \dots \mu_k$
Repeat {
$for\ i = 1\ to\ m$:
$calculate\ distance\ of\ xi\ from\ \ k\ \ cluster\ centers$
$c^i = index\ of\ closest\ cluster$

$for\ j = 1\ to\ k$:
$calculate\ mean\ of\ all\ the\ data\ points\ with\ ci = j$
$and\ assigned\ to$ μ_j
}

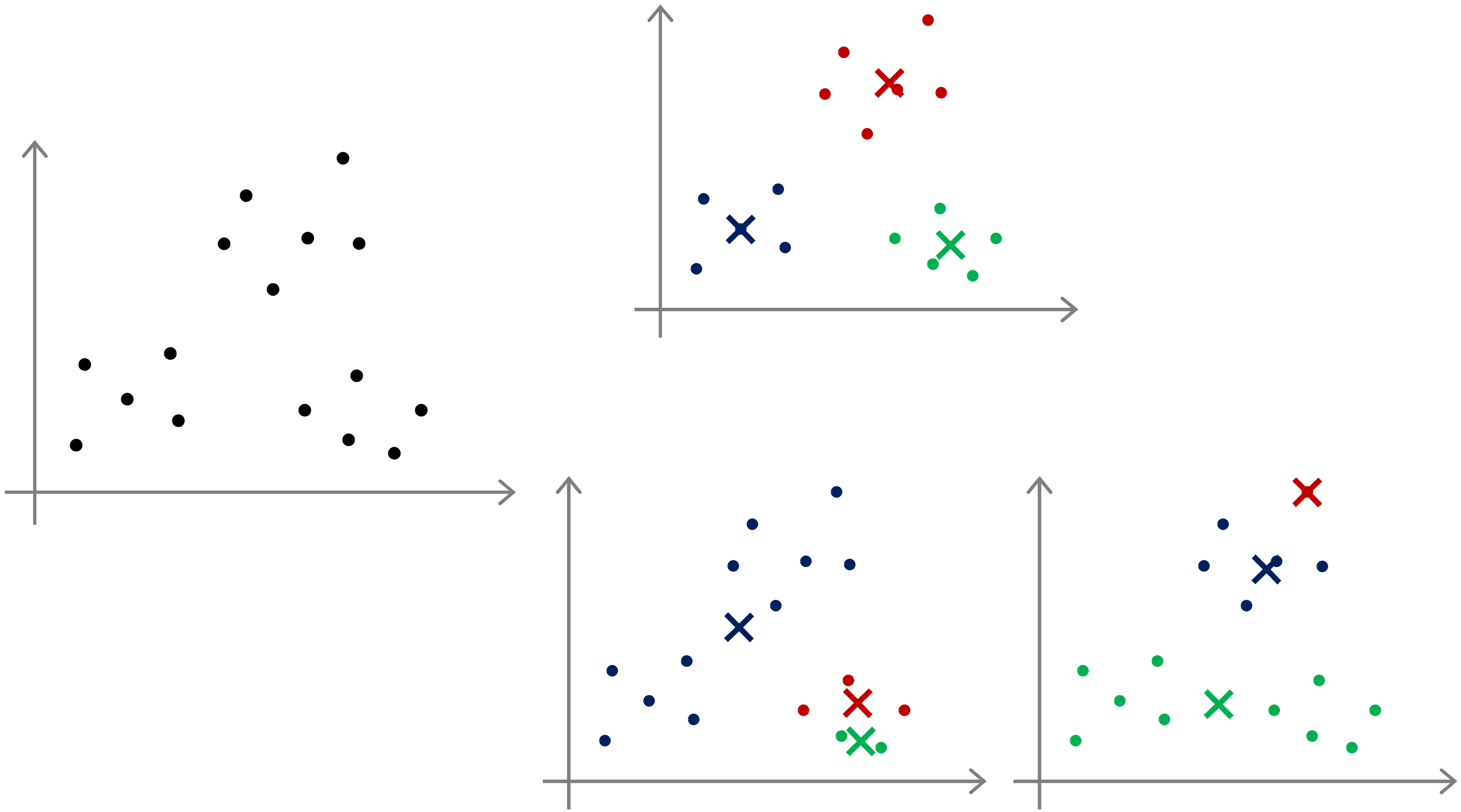
Cost Function

- $J(c^1, c^2, \ldots \ldots, cm, \mu^1, \ldots \ldots, \mu^k)$

$$= \frac{1}{m}\sum_{i=1}^{m}\left\|x^i - \mu_c^i\right\|^2$$

- μ_c^i = Cluster center currently assigned to x^i
- OF: Minimize the Cost function J

Local optima



Dealing with Local Optima:
Random initialization

For i = 1 to 100 {

Randomly initialize K-means.
Run K-means. Get $(c^1, c^2, \dots \dots, cm, \mu^1, \dots \dots, \mu^k)$
Compute cost function

$$J(c^1, c^2, \dots \dots, cm, \mu^1, \dots \dots, \mu^k)$$

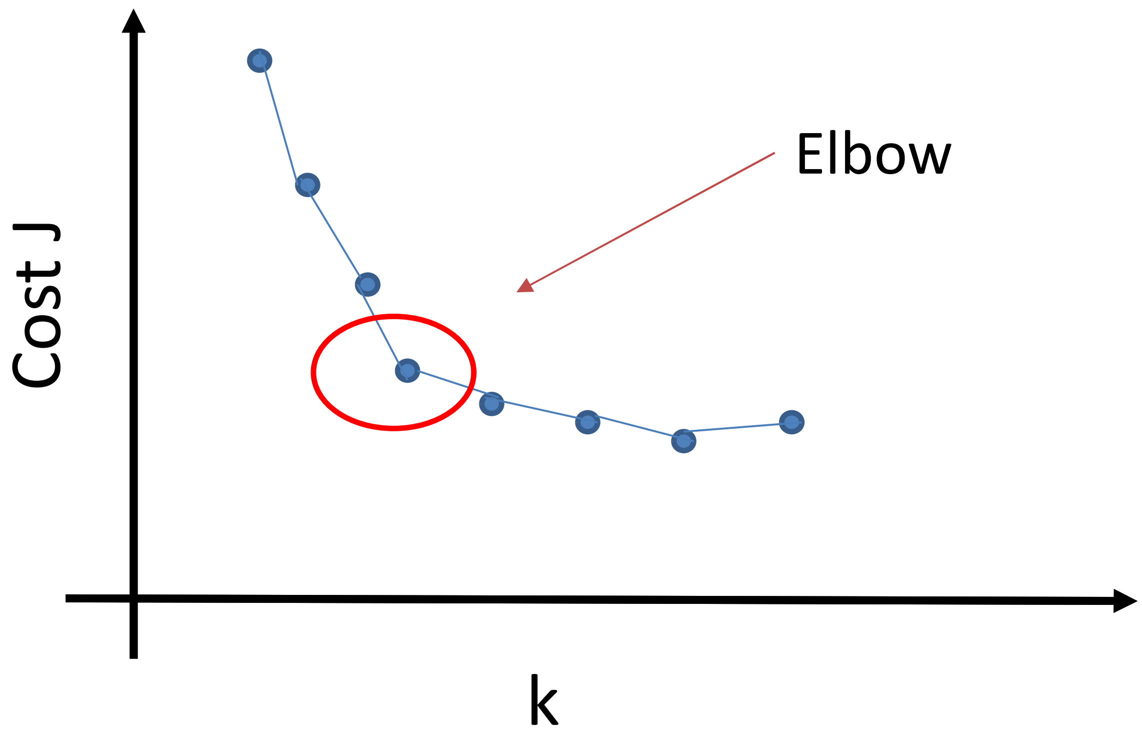
}

Pick clustering that gave lowest cost

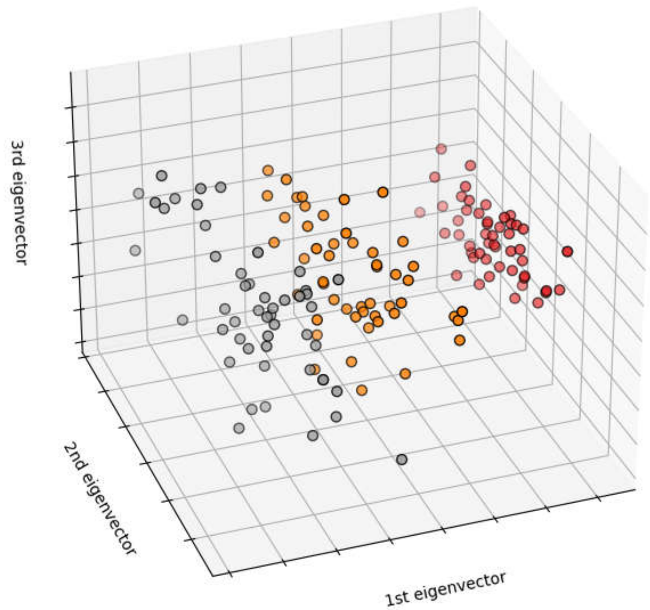
$$J(c^1, c^2, \dots \dots, cm, \mu^1, \dots \dots, \mu^k)$$

What should be the ‘K’

- Elbow Method



Clustering Iris data



K-Means Clustering Example

- Hand written Digit Recognition
- Dataset: 8X8 size Images of 0…9 digits [grayscale]
- Size: 1797 images

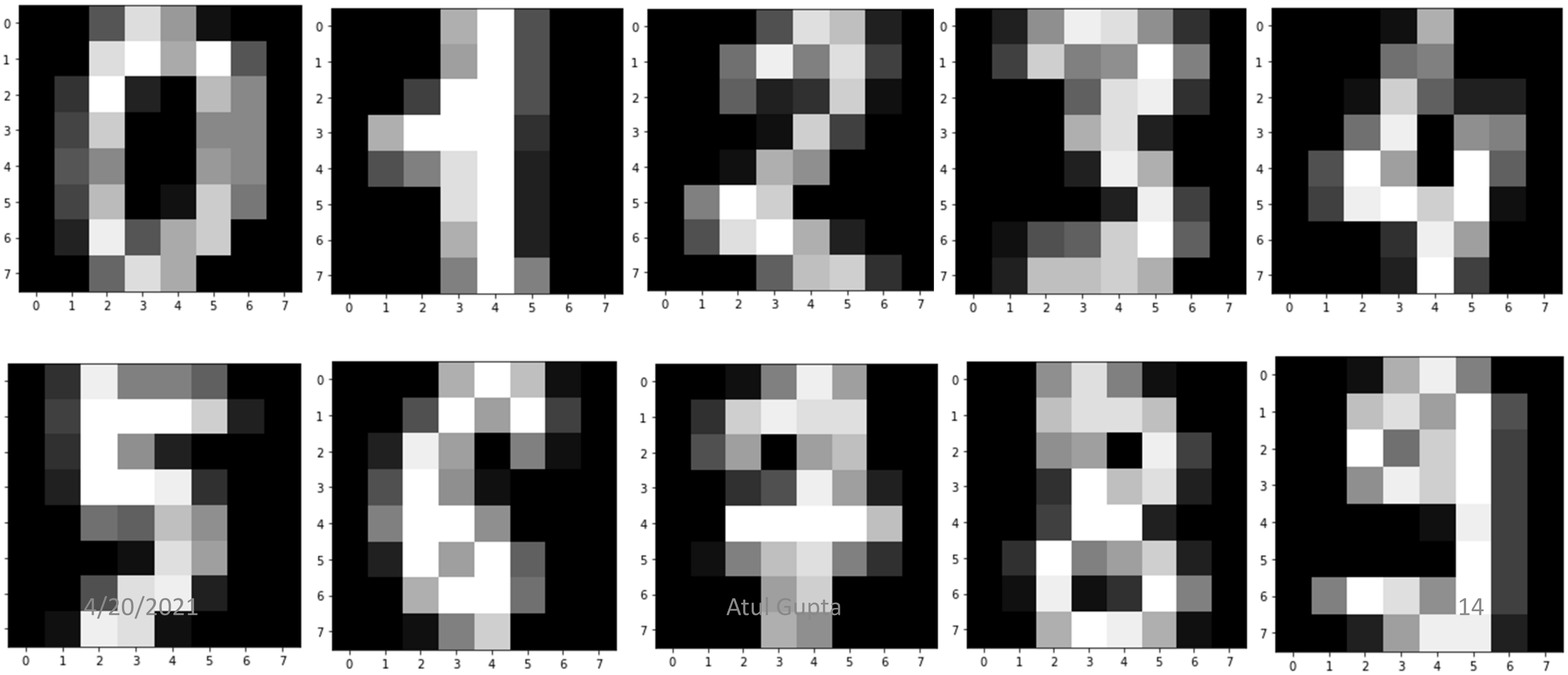
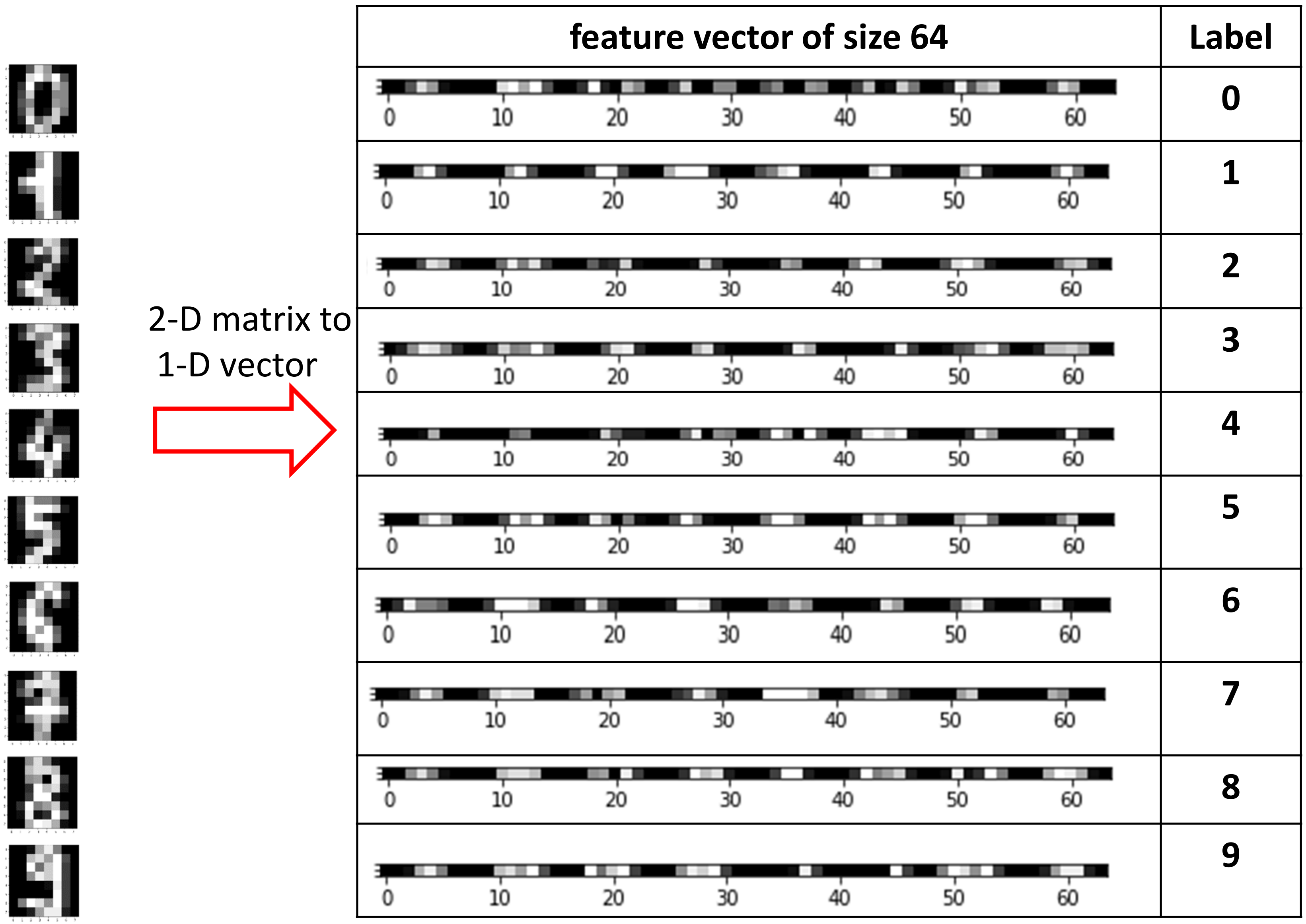
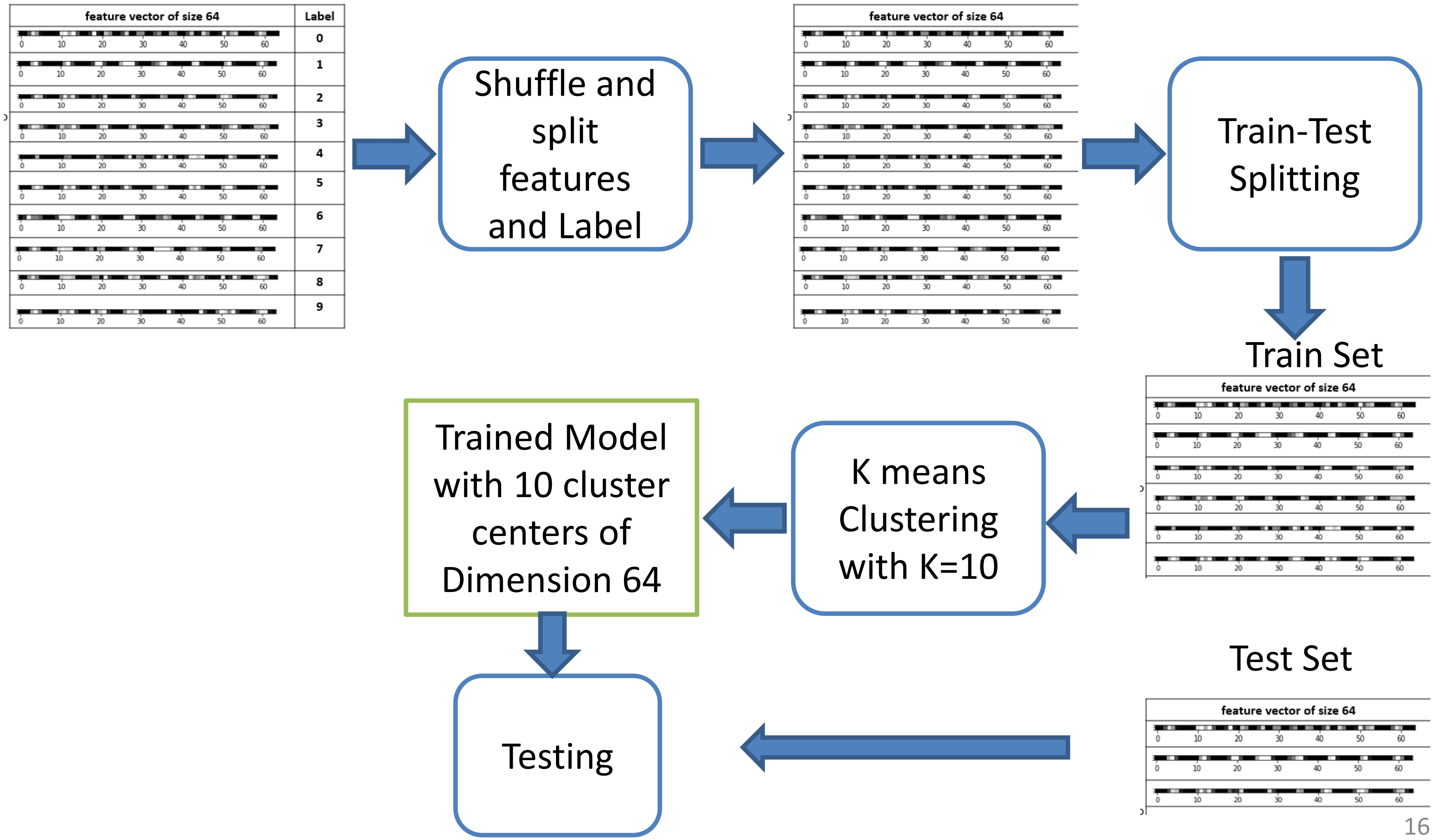


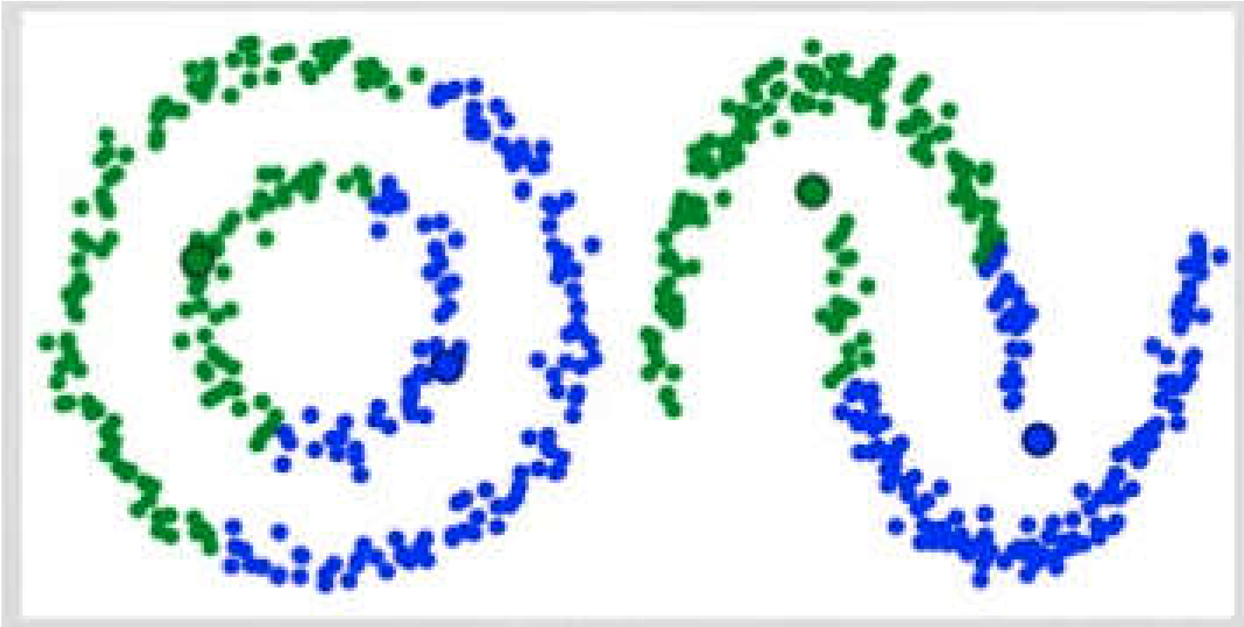
Image to Feature Vector Conversion



K-Means Model Training



We will see the Python code for this Problem …



Bad Clustering with k-Means

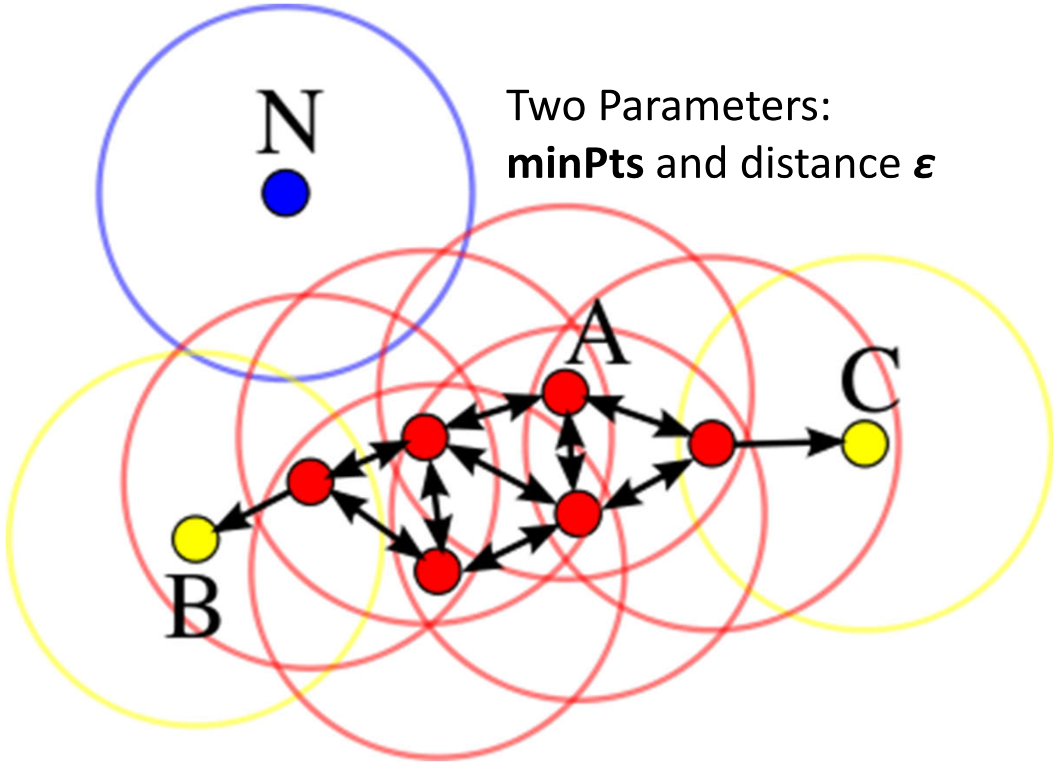
DBSCAN

Density-Based Spatial Clustering of Applications with Noise

Data Points

- Core point (A)
- (Density) – Reachable point (A, B, and C)
- Outlier (N)

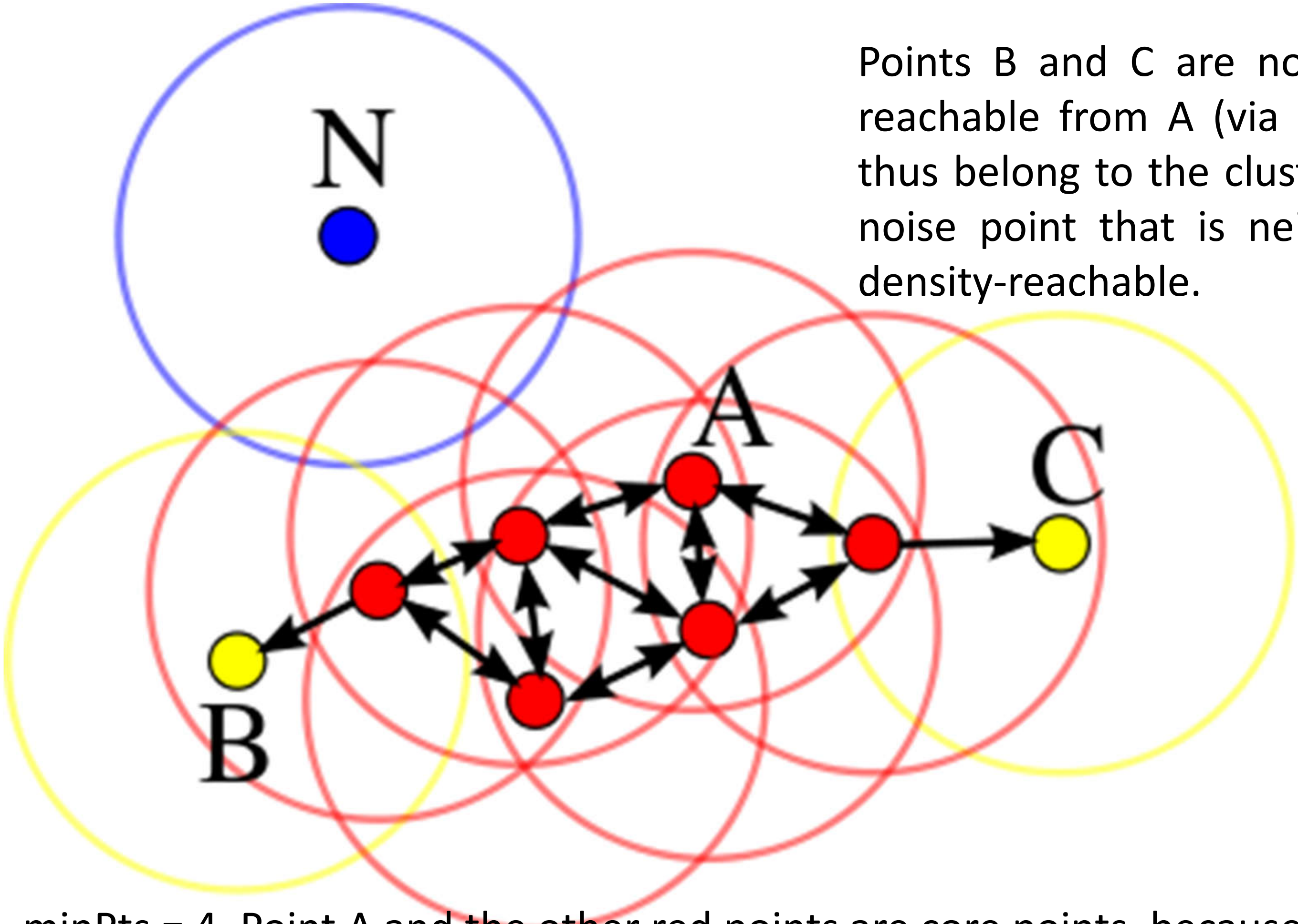
Two Parameters:
minPts and distance ε



Data Point

- **Core point -** A point p is a core point if at least **minPts** points are within distance ε(ε is the maximum radius of the neighborhood from p) of it (including p).
- **(Density) – Reachable point -** A point q is reachable from p if there is a path $p_1, ..., p_n$ with $p_1 = p$ and $p_n = q$, where each p_{i+1} is directly reachable from p_i and all the points on the path must be core points, with the possible exception of q).
- **Outlier -** All points not reachable from any core point are outliers.

The type of points



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 density-reachable.

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).

How clusters are formed

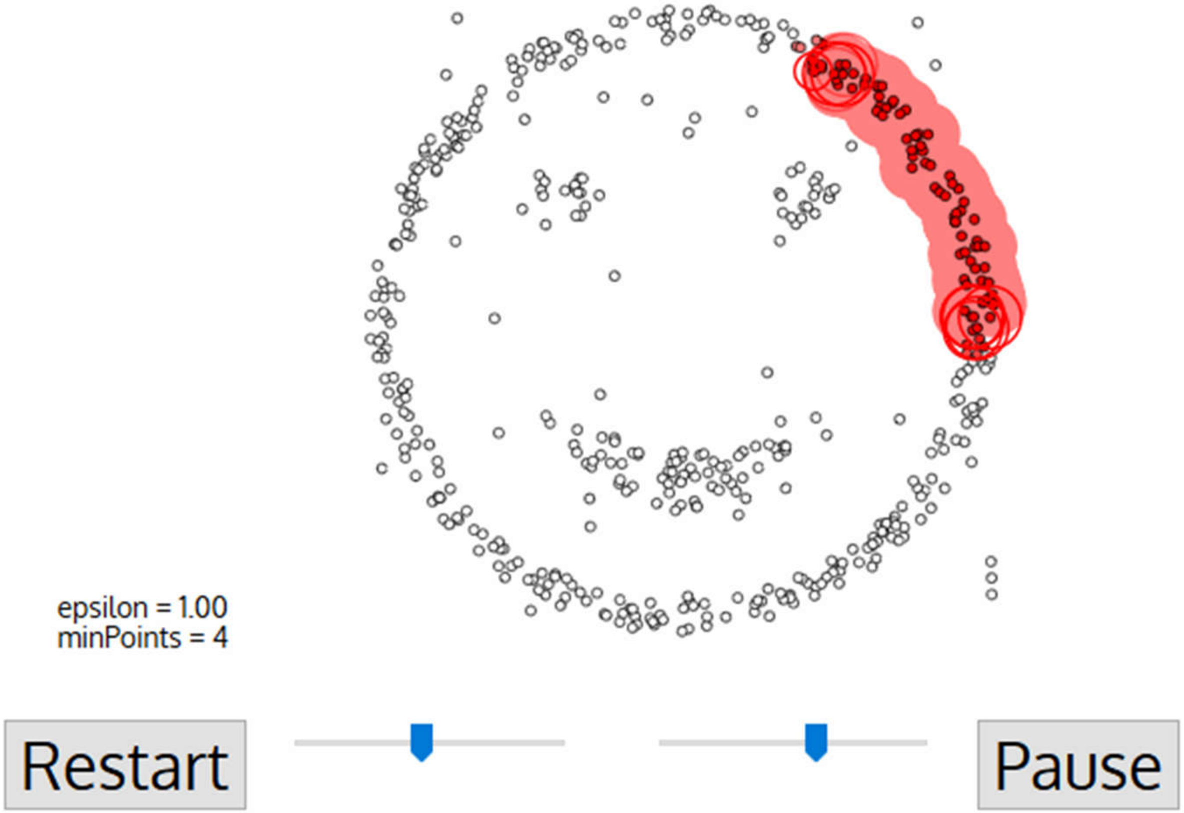
- A cluster then satisfies two properties:
  - All points within the cluster are mutually density-connected.
  - If a point is density-reachable from any core point of the cluster, it is part of the cluster as well.

DBSCAN Algorithm

```
DBSCAN(D, eps, MinPts) {
  C = 0
  for each point P in dataset D {
        if P is visited
              continue next point
        mark P as visited
        NeighborPts = regionQuery(P, eps)
        if sizeof(NeighborPts) < MinPts
              mark P as NOISE
        else {
        C = next cluster
        expandCluster(P, NeighborPts, C, eps, MinPts)
        }
}}
```

DBSCAN Algorithm...

```
expandCluster(P, NeighborPts, C, eps, MinPts) {
  add P to cluster C
  for each point P' in NeighborPts {
    if P' is not visited {
      mark P' as visited
      NeighborPts' = regionQuery(P', eps)
      if sizeof(NeighborPts') >= MinPts
        NeighborPts = NeighborPts joined with NeighborPts'
    }
    if P' is not yet member of any cluster
      add P' to cluster C
  }
}
regionQuery(P, eps)
  return all points within P's eps-neighborhood (including P)
```


epsilon = 1.00
minPoints = 4
Restart
Pause

DBSCAN

Advantages

- Does not require to specify the number of clusters priory
- Identifies outliers
- Able to find arbitrarily sized and arbitrarily shaped clusters

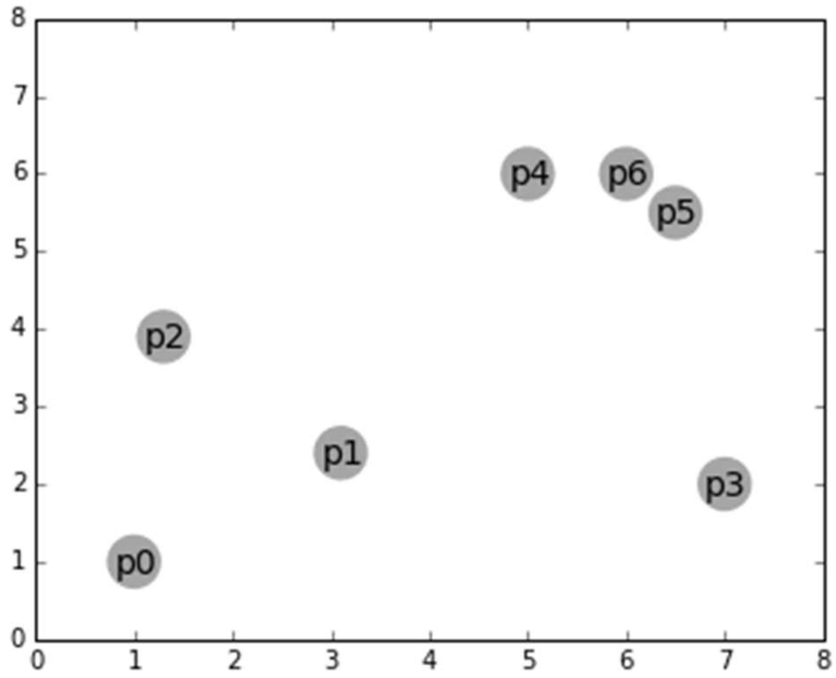
Drawbacks

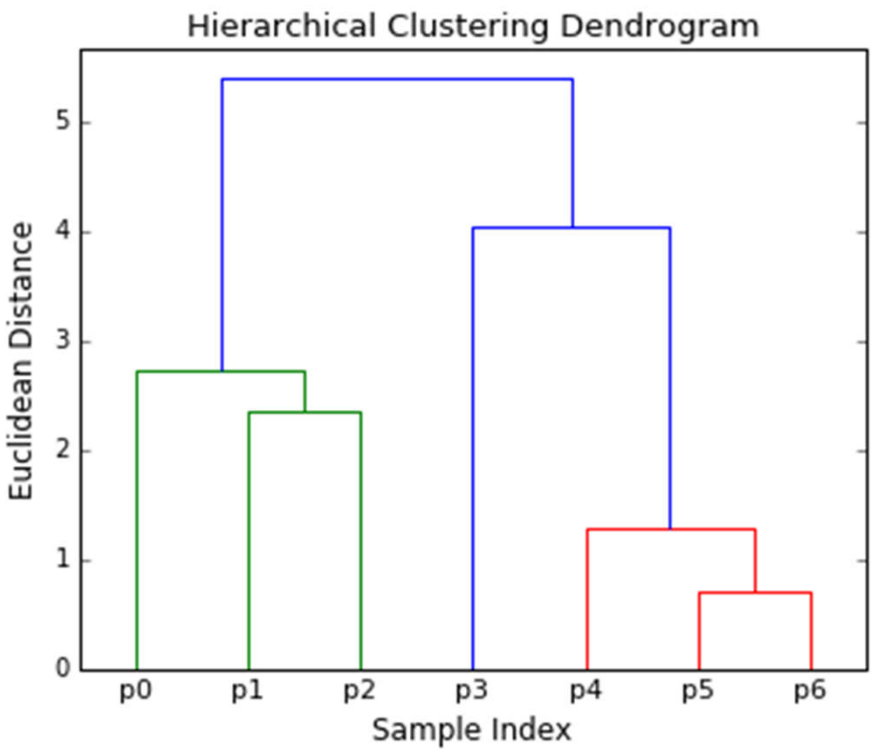
- Doesn't perform as good as others when the clusters are of varying density
- This drawback also occurs with very high-dimensional data since again the distance threshold ε becomes challenging to estimate

Hierarchical Agglomerative Clustering

- We start bottom-up, i.e. each data point as a single cluster
- Repeat until we reach the root of the tree (or any other stopping criteria)
  - On each iteration we combine two clusters into one.
  - The two clusters to be combined are selected as having minimum average inter-cluster distance.

Hierarchical Agglomerative Clustering





Hierarchical Agglomerative Clustering

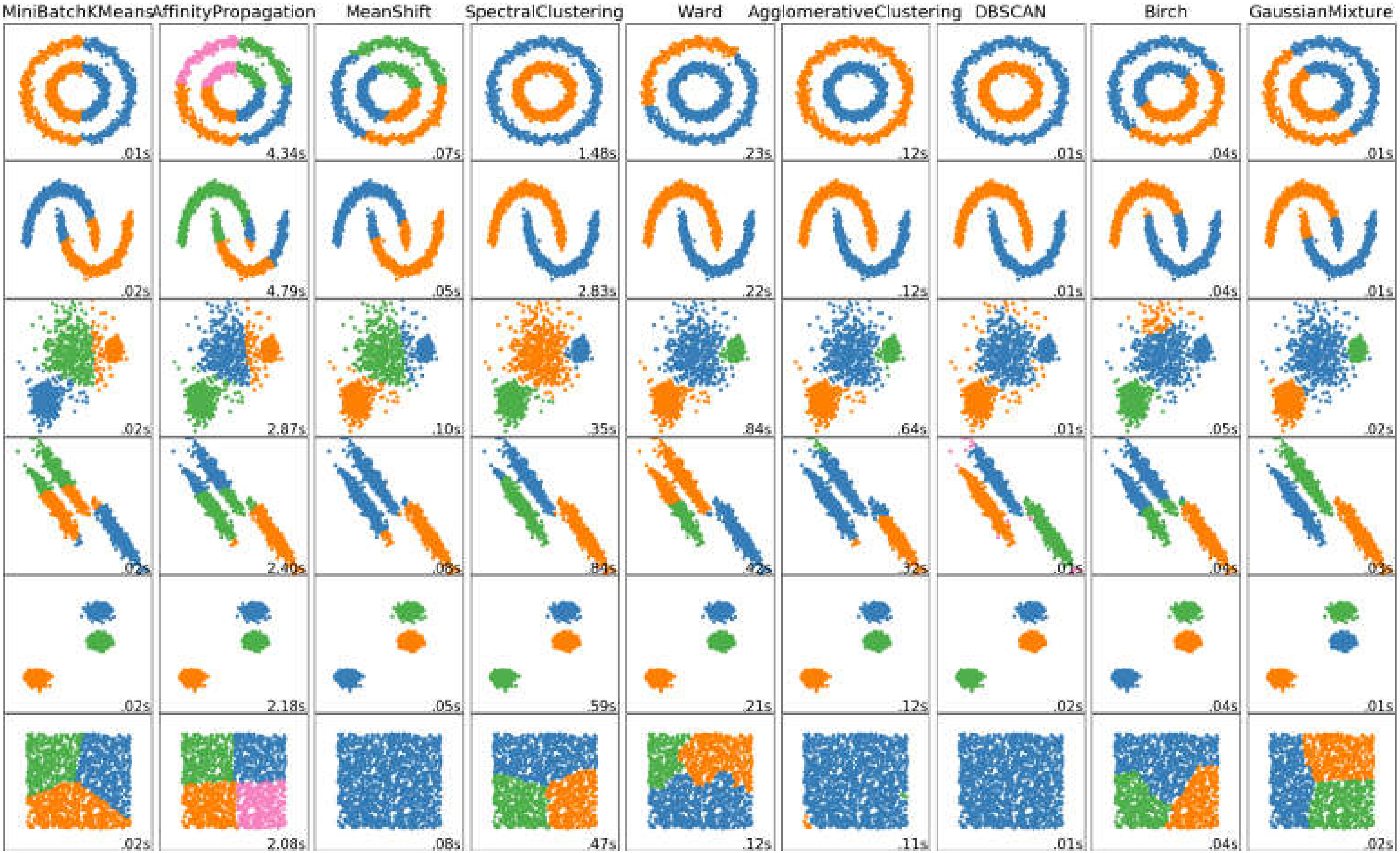
- The Distance Measure
  - We will use *average linkage* which defines the distance between two clusters to be the average distance between data points in the first cluster and data points in the second cluster.

How good is a Clustering?

- McClain–Rao Ratio
  - the ratio of the average Intra-cluster distance (A) to the average inter-cluster distance (B)
- Silhouette coefficient

$$= (B-A) \,/\, \max(A, B)$$

  - where A = average intra-cluster distance and B is the average inter-cluster distance
  - ranges between −1 to +1
- Other measures: Rand index, Jaccard coefficient, Fowlkes and Mallows index and Dunn index



https://scikit-learn.org/stable/modules/clustering.html courtesy of Scikit Learn

Challenges in Clustering

- Clustering problems with non-numeric attributes
- Identify number of clusters
- Quality of Clusters

Clustering: Summary

- Clustering is one of the most important **Unsupervised Learning** problem
- **DBSCAN and its variates** are perhaps the most useful clustering algorithms
- Measuring the **goodness of clustering** is a challenge

Thank You

atul@iiitdmj.ac.in