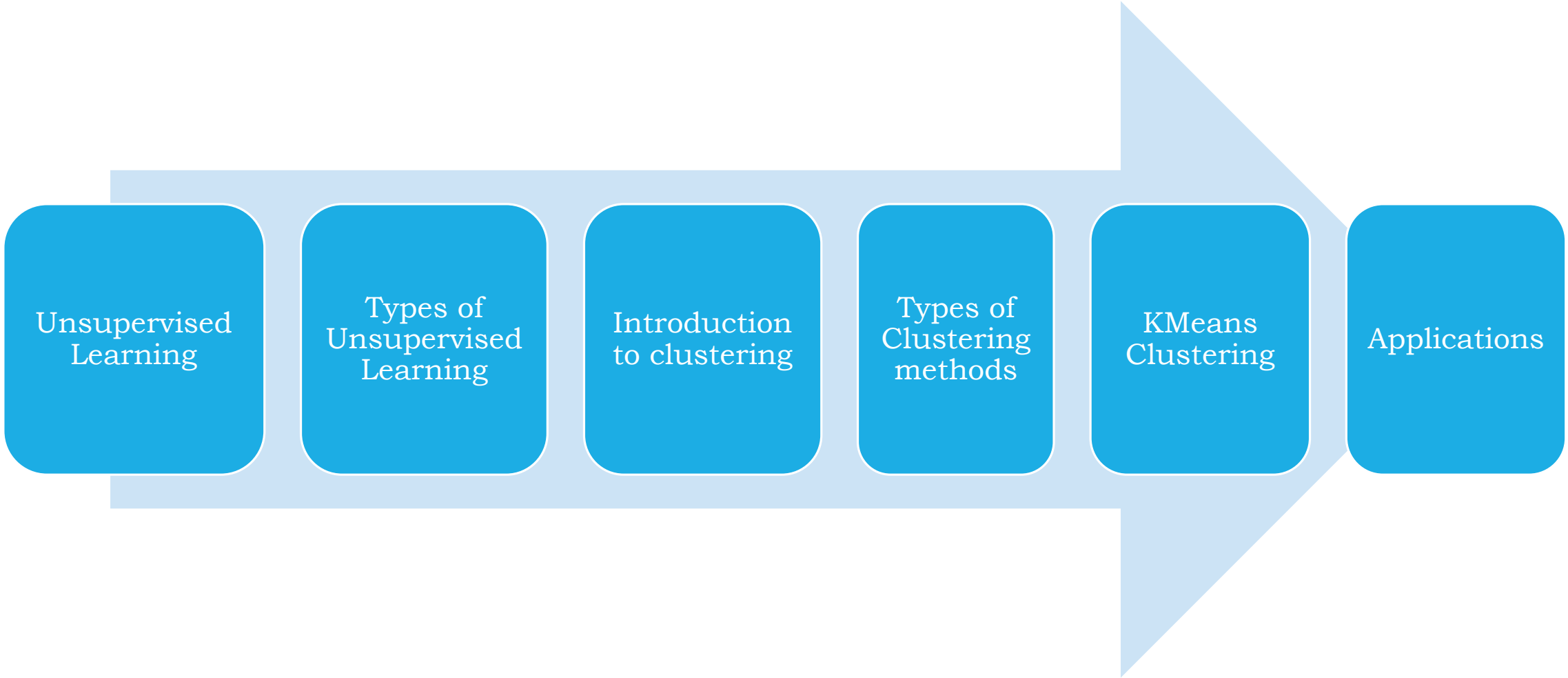
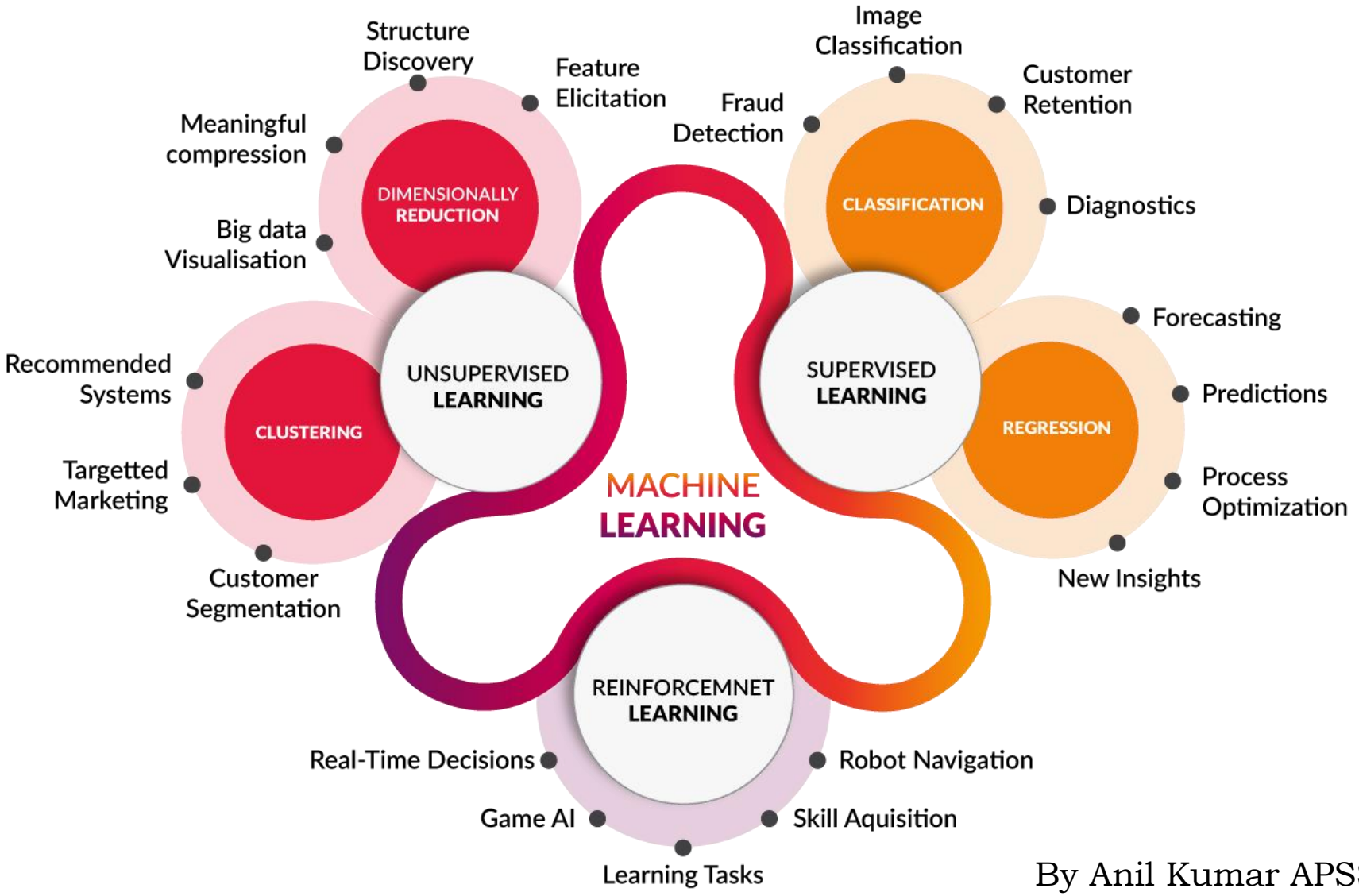
UNSUPERVISED LEARNING

DAY08 AGENDA

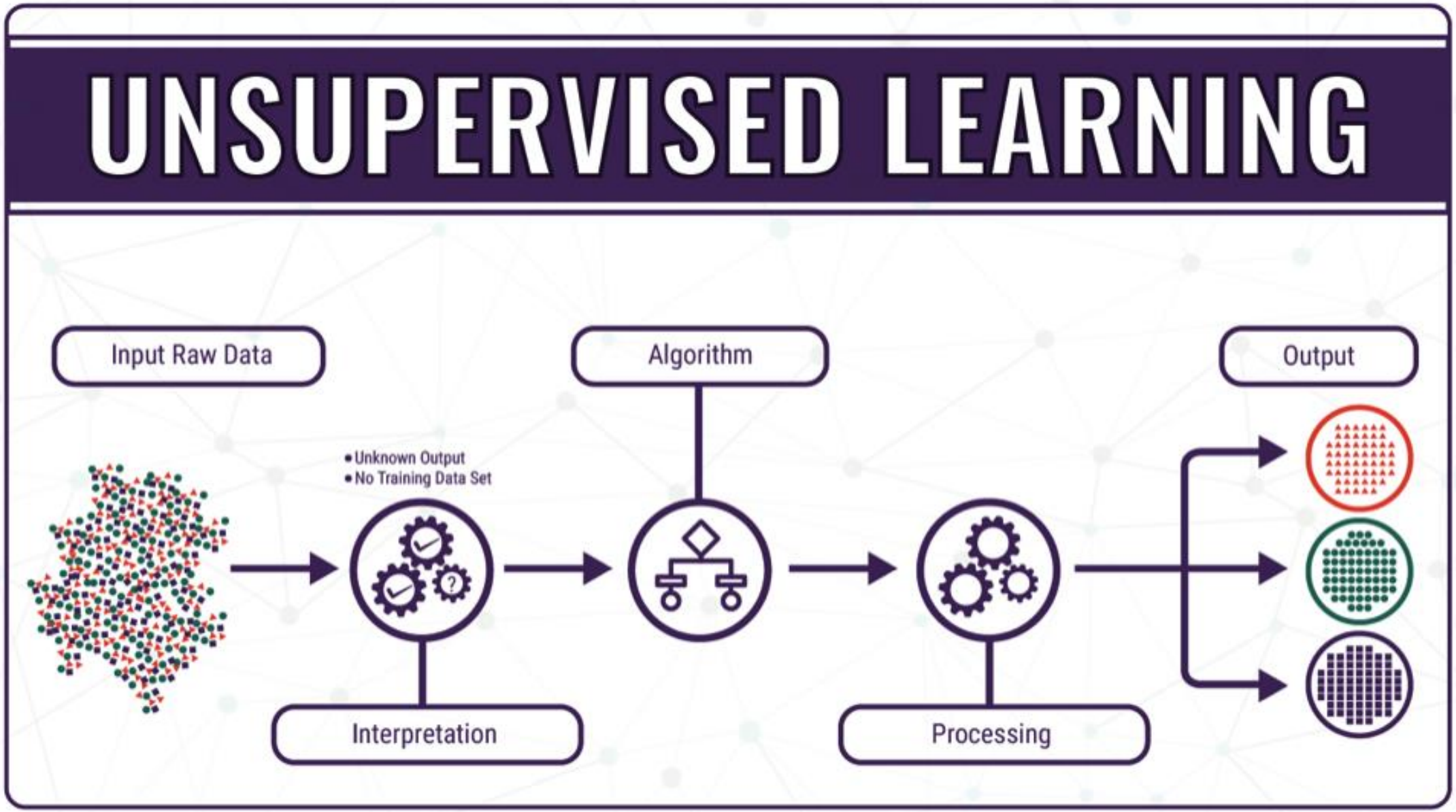


By Anil Kumar APSSDC

MACHINE LEARNING CATEGORIES



By Anil Kumar APSSDC

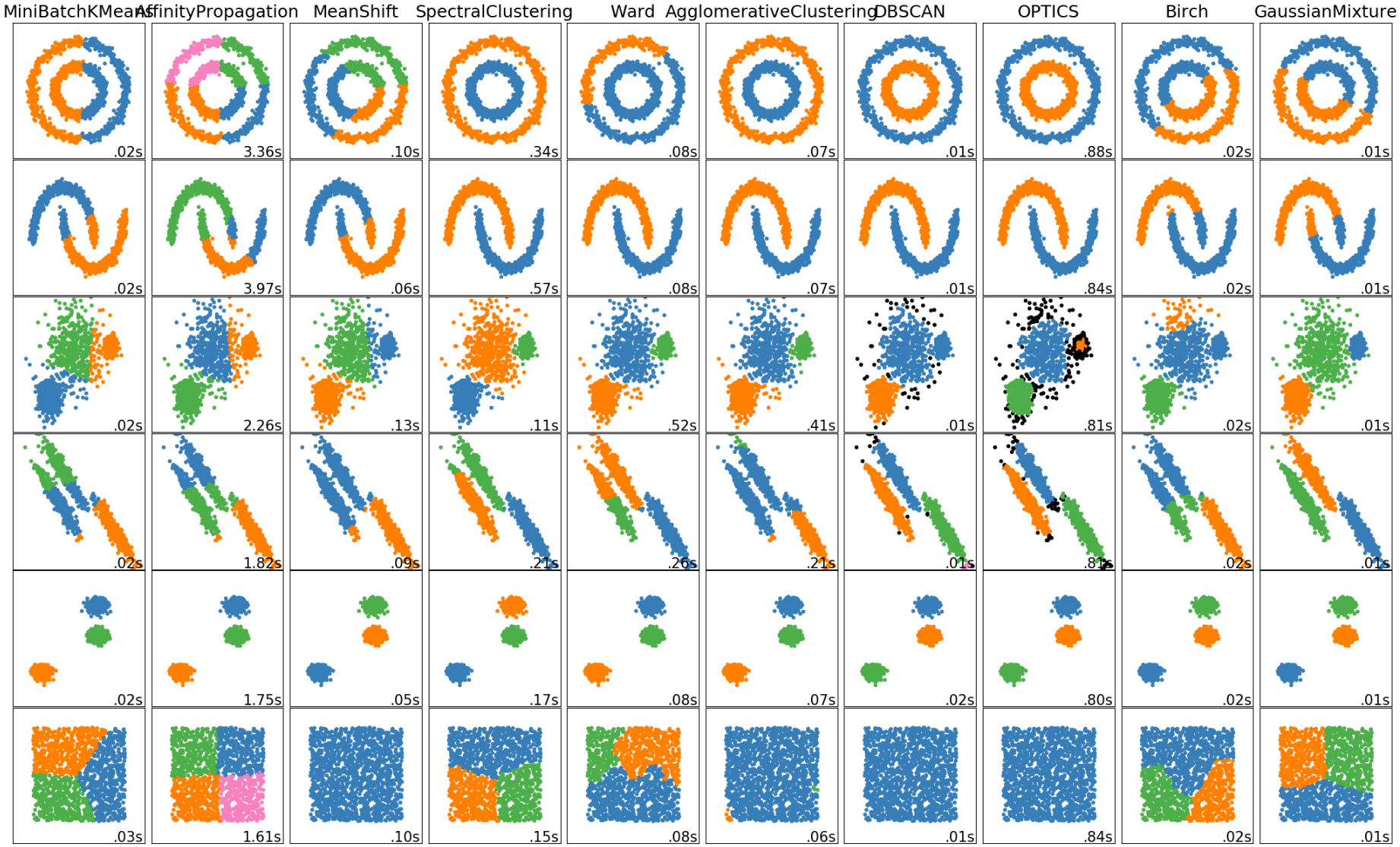


By Anil Kumar APSSDC

CLUSTERING

A cluster is a group of data points or objects in a dataset that are similar to other objects in the group, and dissimilar to datapoints in other clusters

TYPES OF CLUSTERING



By Anil Kumar APSSDC

CLUSTERING APPLICATIONS

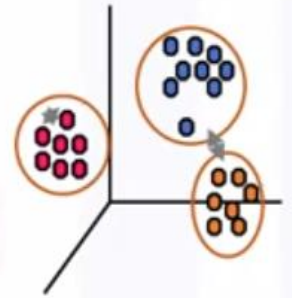
- RETAIL/MARKETING:
  - Identifying buying patterns of customers
  - Recommending new books or movies to new customers
- BANKING:
  - Fraud detection in credit card use
  - Identifying clusters of customers (e.g., loyal)
- INSURANCE:
  - Fraud detection in claims analysis
  - Insurance risk of customers

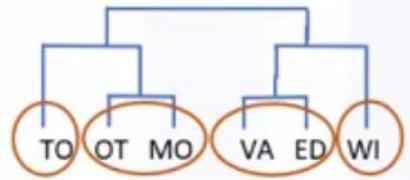
CONTD..

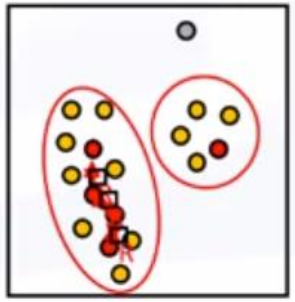
- PUBLICATION:
  - Auto-categorizing news based on their content
  - Recommending similar news articles
- MEDICINE:
  - Characterizing patient behavior
- BIOLOGY:
  - Clustering genetic markers to identify family ties

CLUSTERING ALGORITHMS

- Partitioned-based Clustering
  - Relatively efficient
  - E.g. k-Means, k-Median, Fuzzy c-Means
- Hierarchical Clustering
  - Produces trees of clusters
  - E.g. Agglomerative, Divisive
- Density-based Clustering
  - Produces arbitrary shaped clusters
  - E.g. DBSCAN



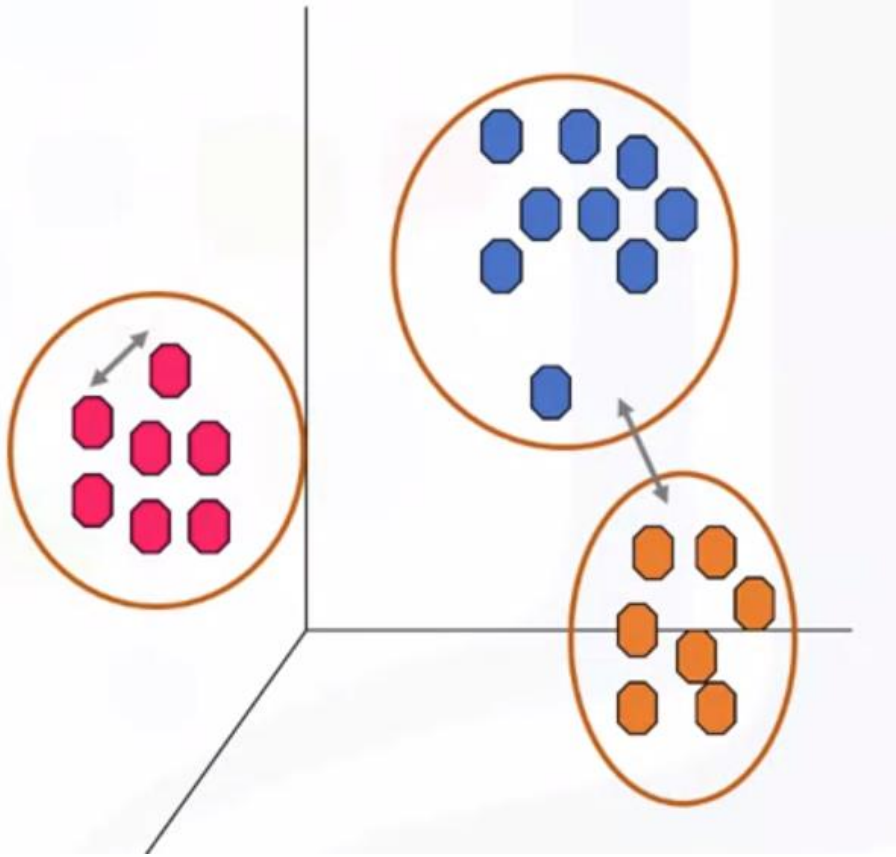




By Anil Kumar APSSDC

K-MEANS ALGORITHM

- Partitioning Clustering
- K-means divides the data into non-overlapping subsets (clusters) without any cluster-internal structure
- Examples within a cluster are very similar
- Examples across different clusters are very different



By Anil Kumar APSSDC

WHAT IS THE OBJECTIVE OF K-MEANS?

1. To form clusters in such a way that similar samples go into a cluster, and dissimilar samples fall into different clusters.

2. To minimize the “intra cluster” distances and maximize the “inter-cluster” distances.

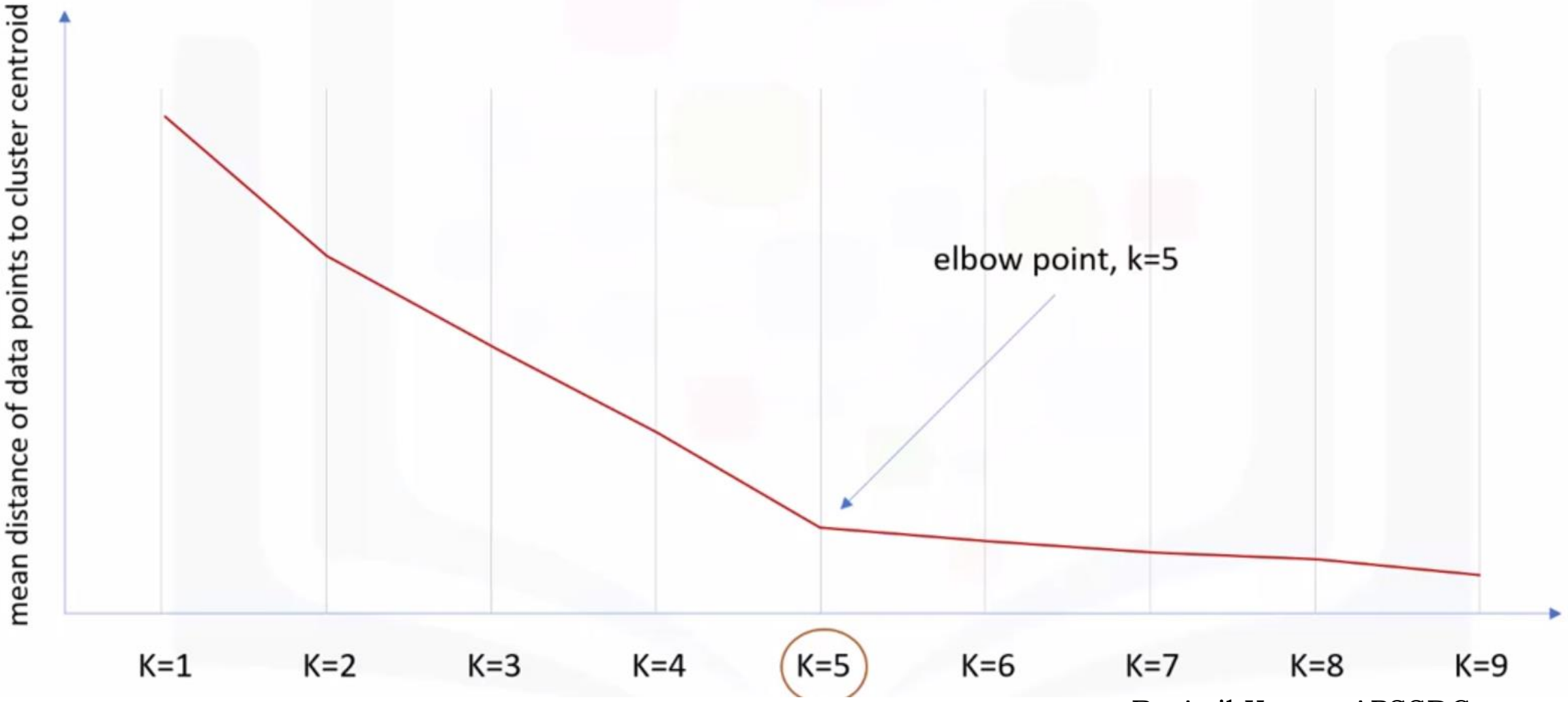
3. To divide the data into non-overlapping clusters without any cluster-internal structure

By Anil Kumar APSSDC

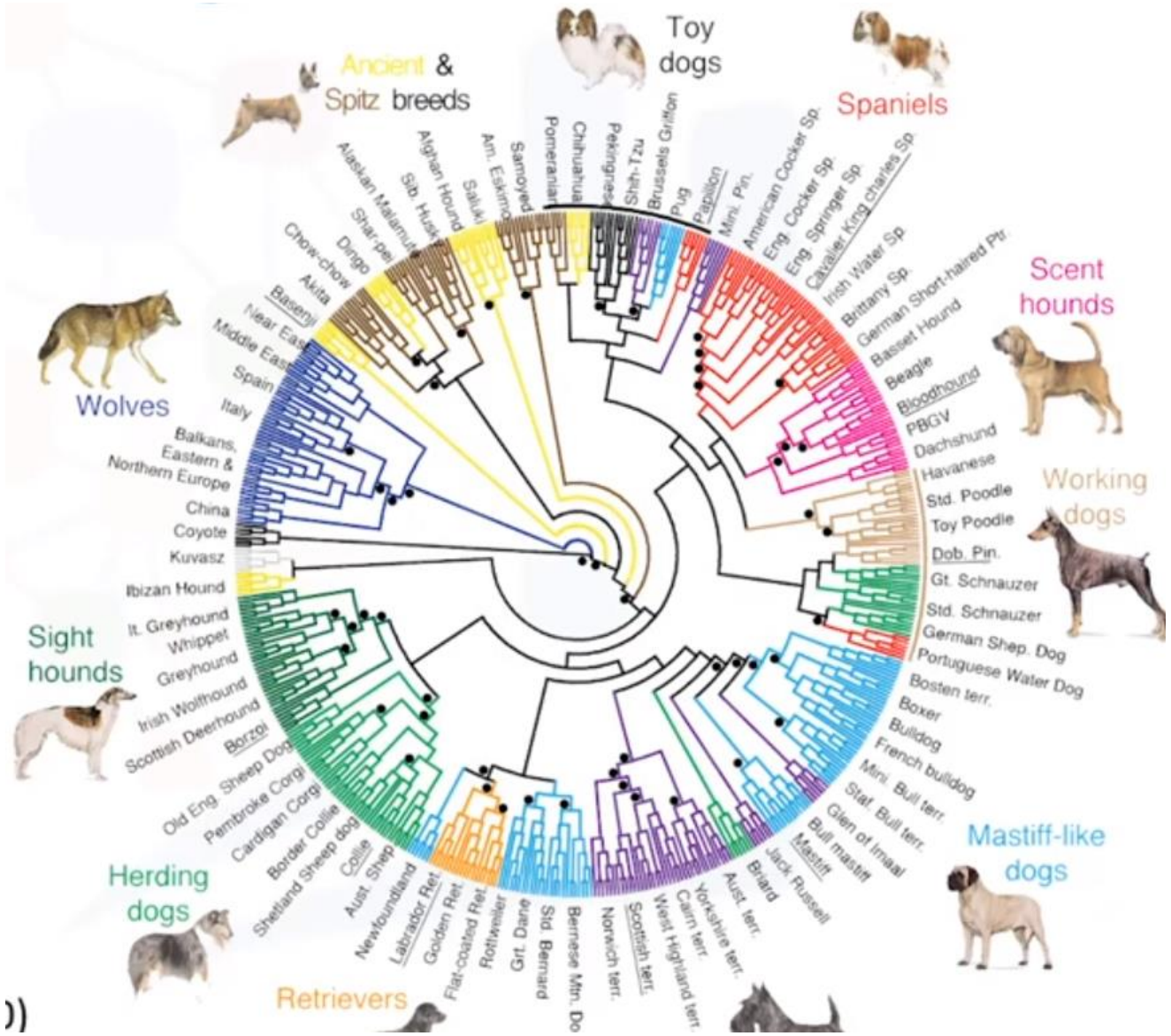
HOW K-MEANS ALGORITHM WORKS

1. Randomly placing *k* centroids, one for each cluster.
2. Calculate the distance of each point from each centroid.
3. Assign each data point (object) to its closest centroid, creating a cluster.
4. Recalculate the position of the *k* centroids.
5. Repeat the steps 2-4, until the centroids no longer move.

CHOOSING BEST ‘K’

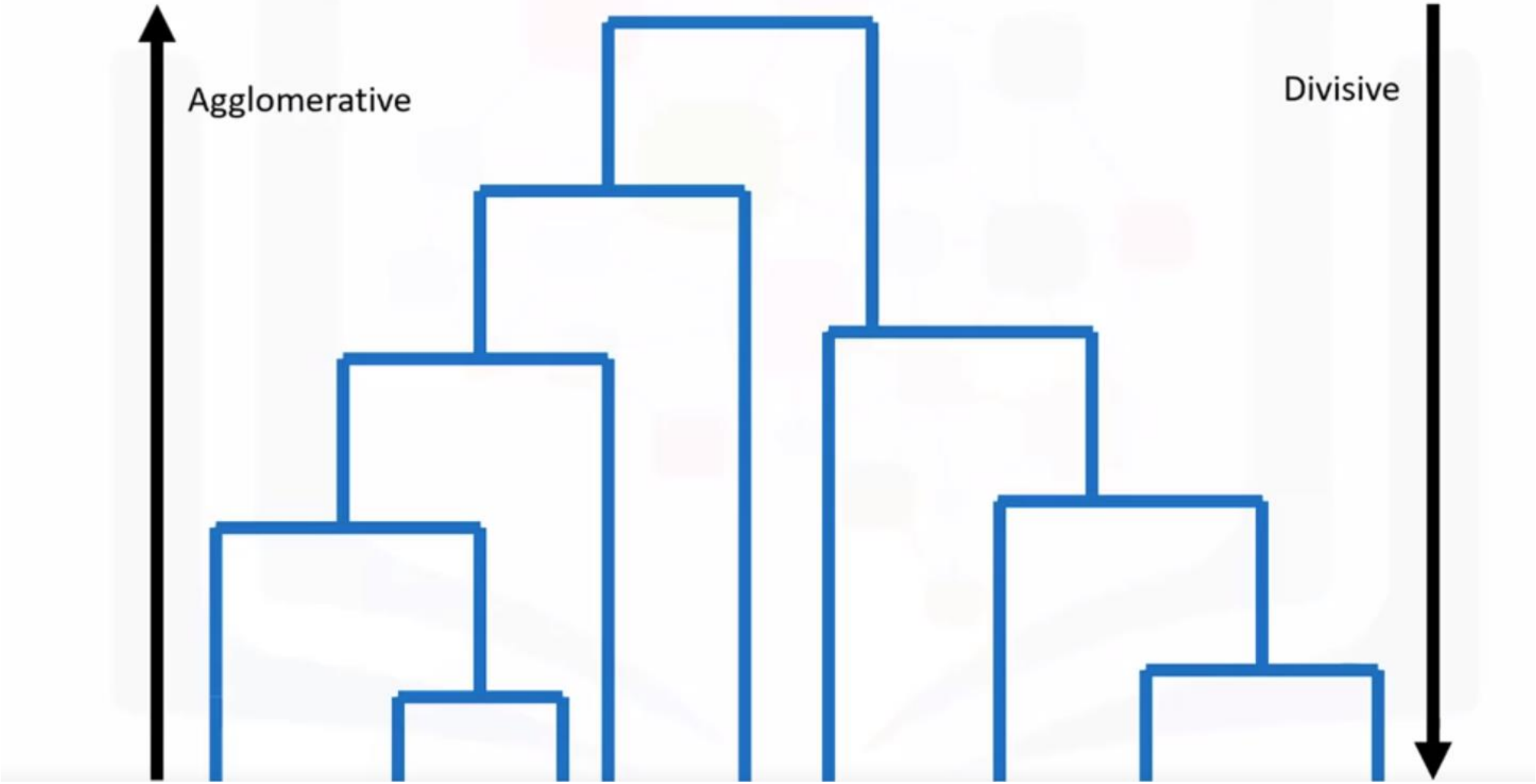


HIERARCHAL CLUSTERING



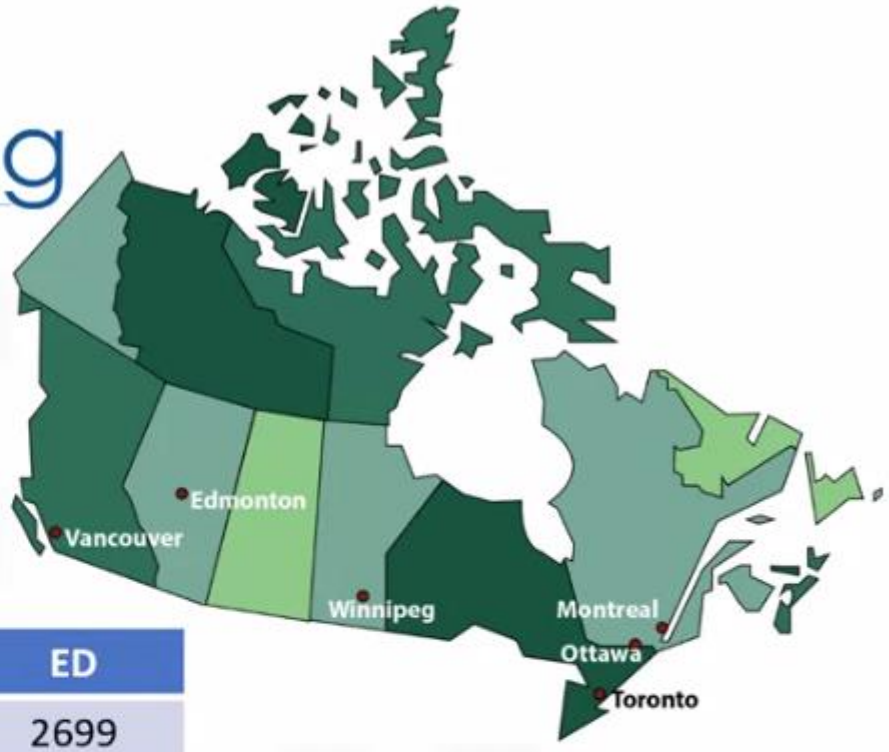
By Anil Kumar APSSDC

HIERARCHAL CLUSTERING DENDROGRAM



By Anil Kumar APSSDC

Agglomerative clustering



j

	TO	OT	VA	MO	WI	ED
TO		351	3363	505	1510	2699
OT			3543	167	1676	2840
VA				3690	1867	819
MO					1824	2976
WI						1195
ED						

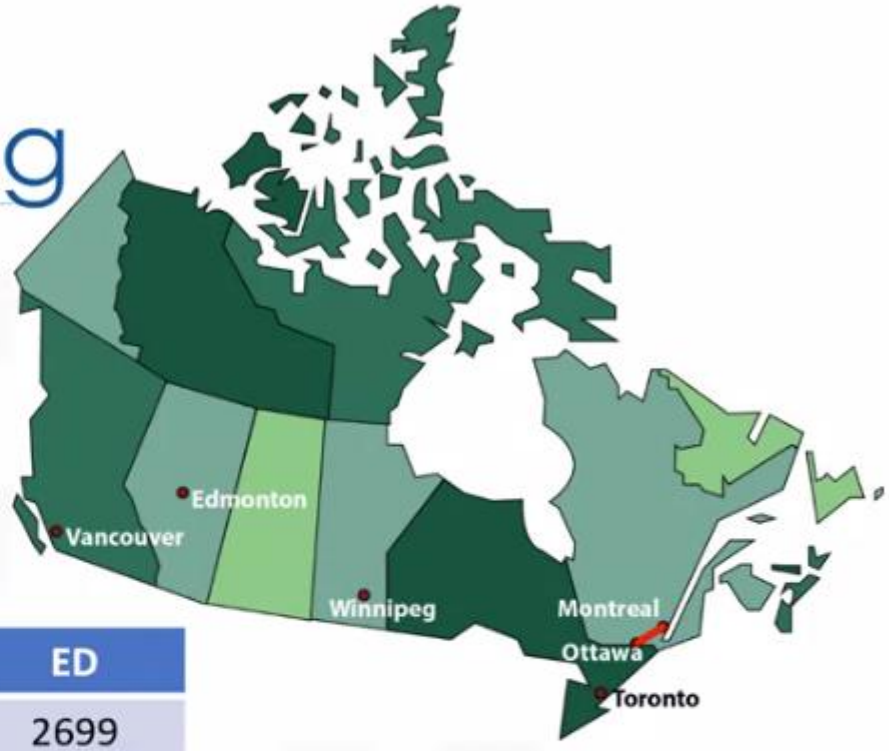
i

dis(i,j)

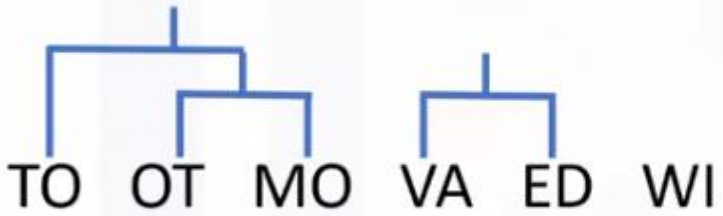
Agglomerative clustering

TO OT MO VA ED WI

	TO	OT	VA	MO	WI	ED
TO		351	3363	505	1510	2699
OT			3543	167	1676	2840
VA				3690	1867	819
MO					1824	2976
WI						1195
ED						



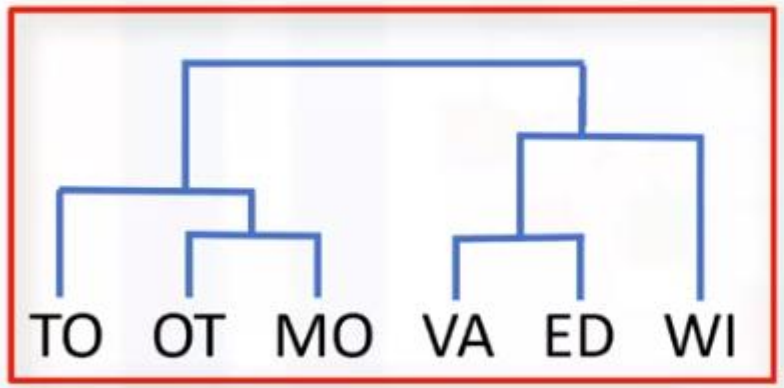
Agglomerative clustering



	TO/OT/MO	VA	WI	ED
TO/OT/MO		3543	1676	2840
VA			1867	819
WI				1195
ED				

By Anil Kumar APSSDC

Hierarchical clustering

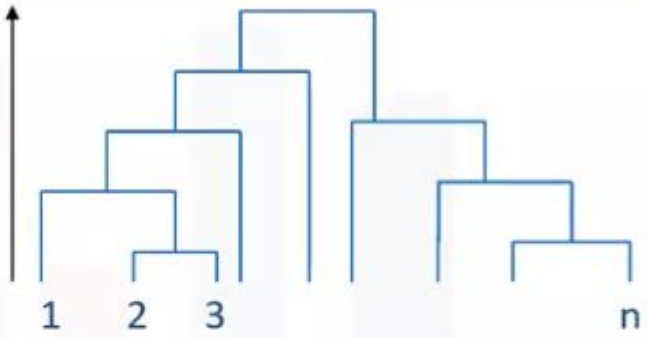


	TO/OT/MO	VA/ED/WI
TO/OT/MO		1676
VA/ED/WI		

By Anil Kumar APSSDC

Agglomerative algorithm

1. Create *n* clusters, one for each data point
2. Compute the Proximity Matrix
3. **Repeat**
   i. Merge the two closest clusters
   ii. Update the proximity matrix
4. **Until** only a single cluster remains



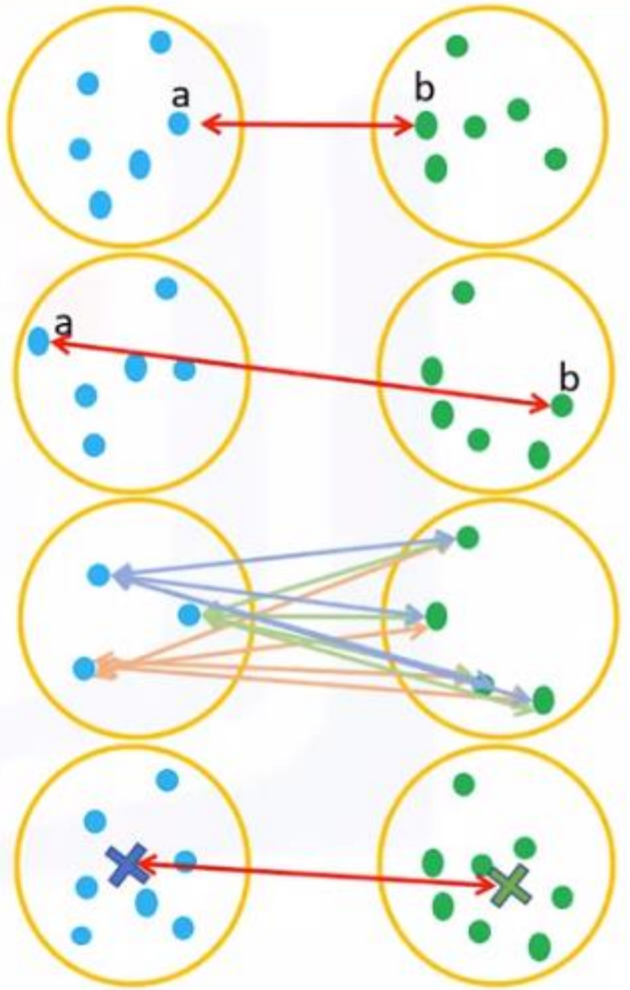
$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

Distance between clusters

- Single-Linkage Clustering
  - Minimum distance between clusters
- Complete-Linkage Clustering
  - Maximum distance between clusters
- Average Linkage Clustering
  - Average distance between clusters
- Centroid Linkage Clustering
  - Distance between cluster centroids

Distance between clusters

- Single-Linkage Clustering
  - Minimum distance between clusters
- Complete-Linkage Clustering
  - Maximum distance between clusters
- Average Linkage Clustering
  - Average distance between clusters
- Centroid Linkage Clustering
  - Distance between cluster centroids



By Anil Kumar APSSDC

ADVANTAGES AND DISADVANTAGES

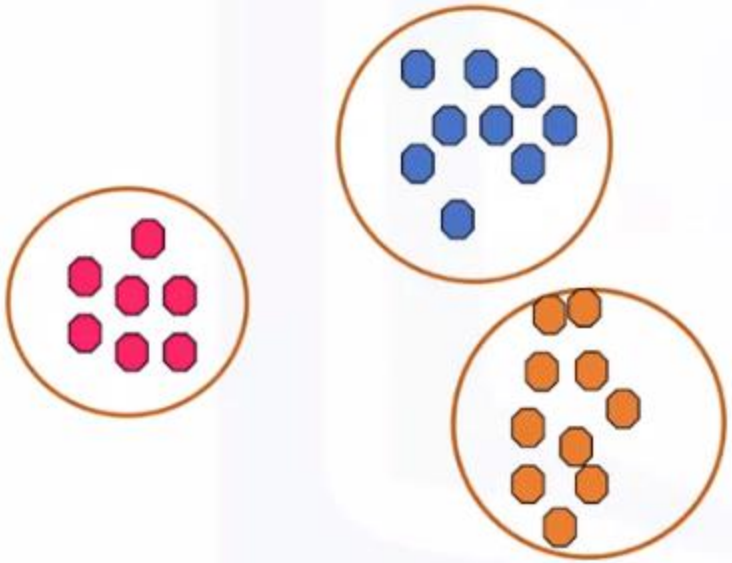
Advantages	Disadvantages
Doesn't required number of clusters to be specified.	Can never undo any previous steps throughout the algorithm.
Easy to implement.	Generally has long runtimes.
Produces a dendrogram, which helps with understanding the data.	Sometimes difficult to identify the number of clusters by the dendrogram.

Hierarchical clustering Vs. *K*-means

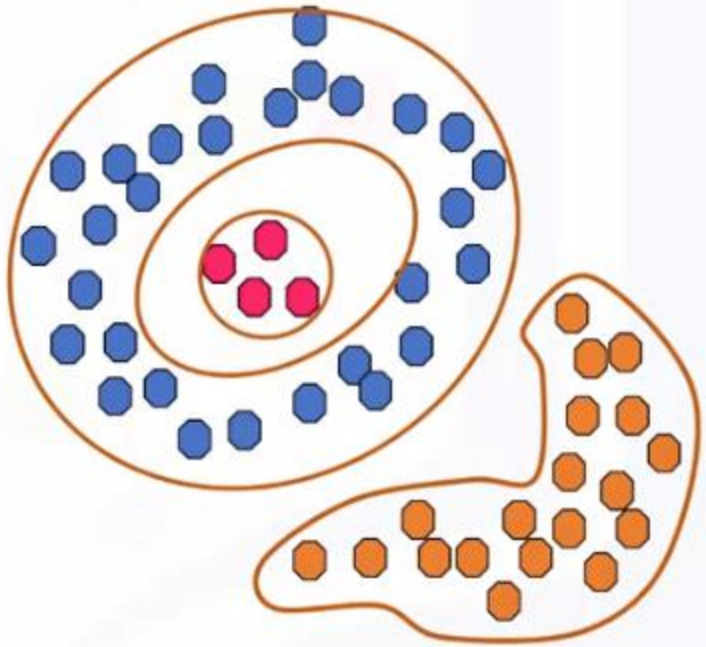
K-means	Hierarchical Clustering
1. Much more efficient	1. Can be slow for large datasets
2. Requires the number of clusters to be specified	2. Does not require the number of clusters to run
3. Gives only one partitioning of the data based on the predefined number of clusters	3. Gives more than one partitioning depending on the resolution
4. Potentially returns different clusters each time it is run due to random initialization of centroids	4. Always generates the same clusters

Density-based clustering

- Spherical-shape clusters

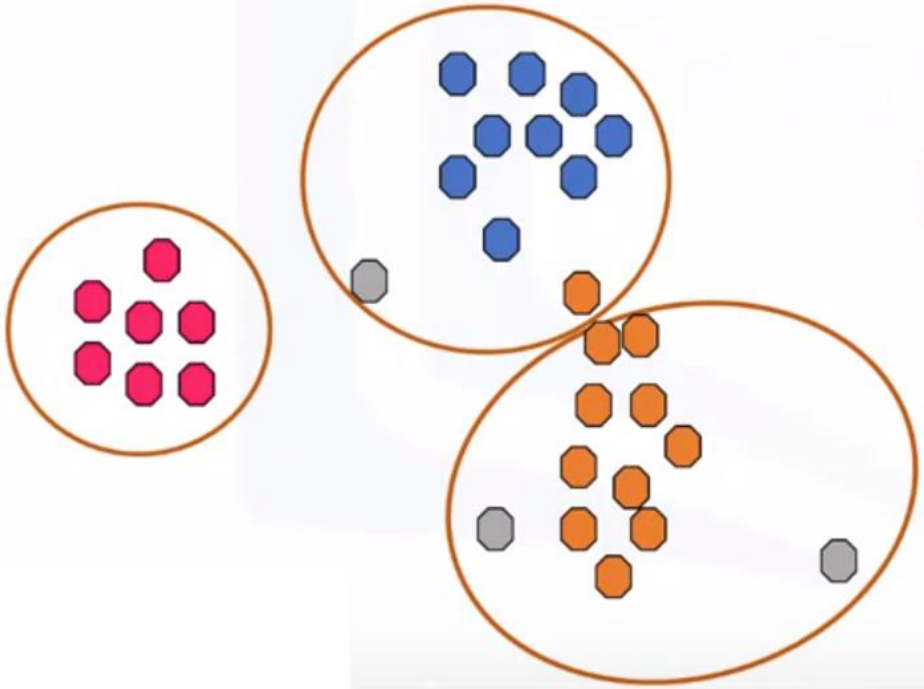


- Arbitrary-shape clusters

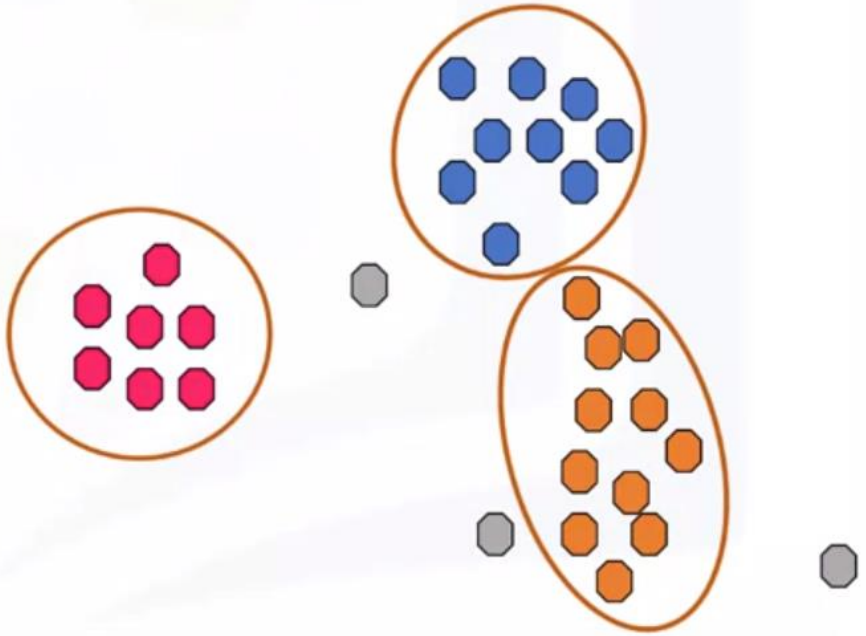


K-MEANS Vs. DENSITY BASED CLUSTERING

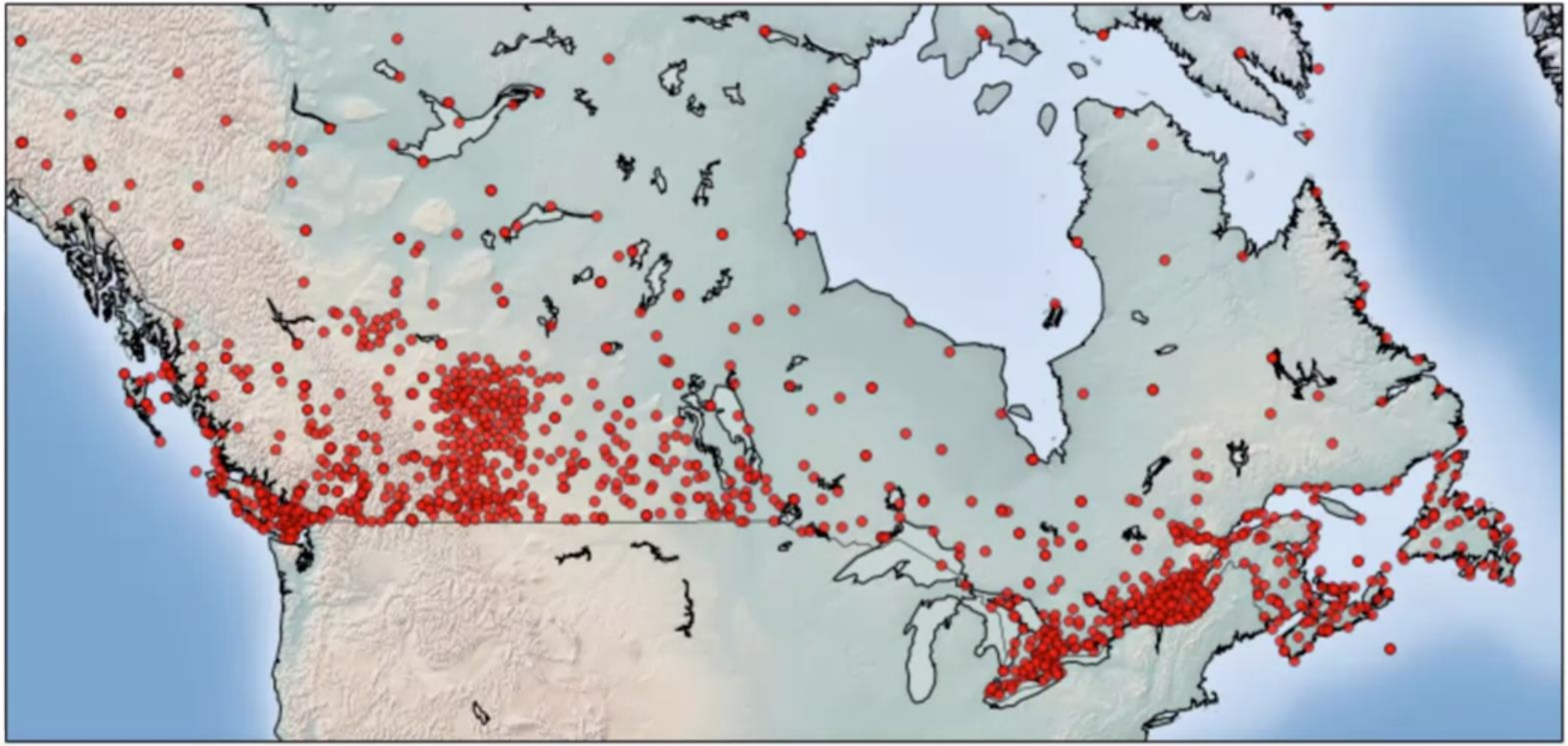
- k-Means assigns all points to a cluster even if they do not belong in any



- Density-based Clustering locates regions of high density, and separates outliers

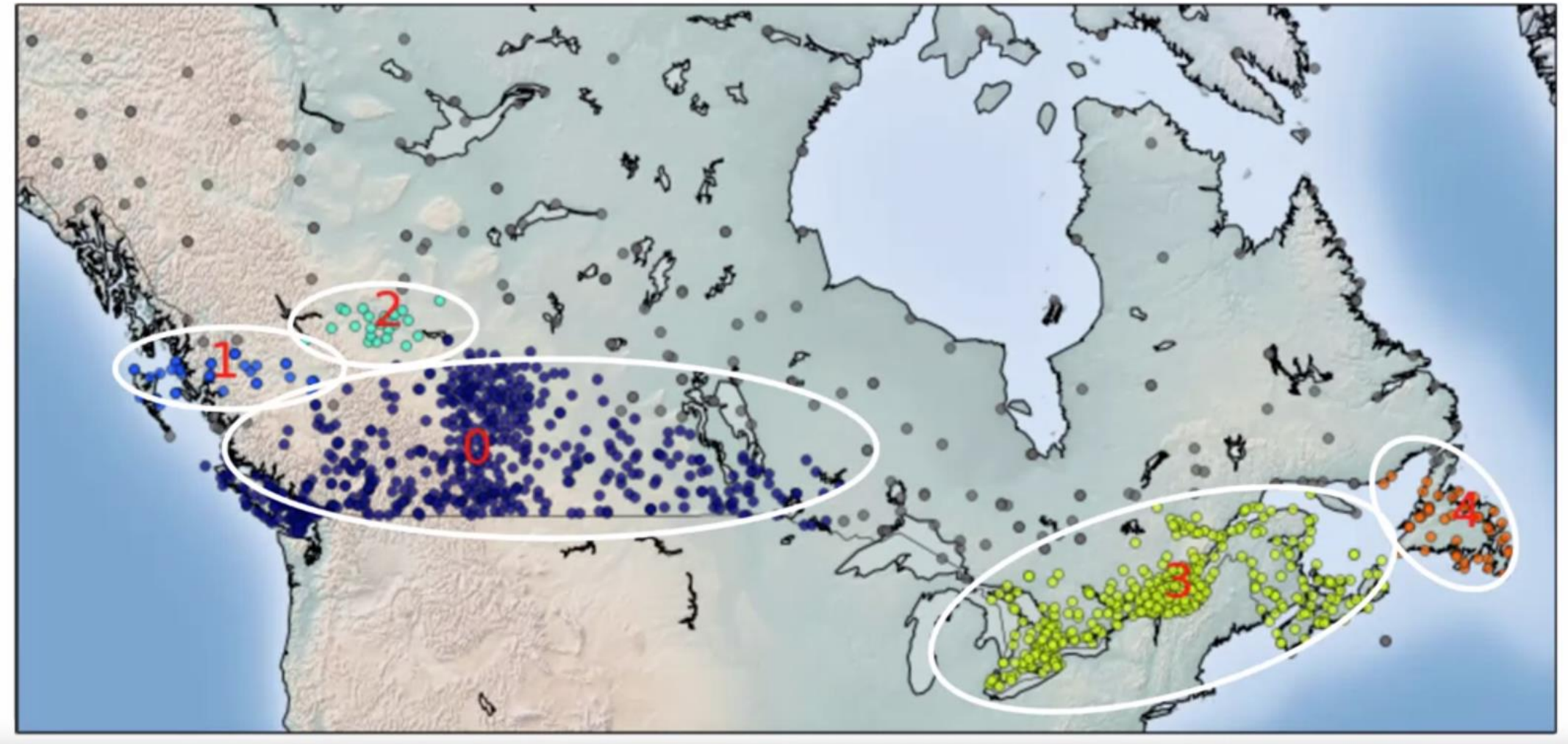


DBSCAN FOR CLASS IDENTIFICATION



By Anil Kumar APSSDC

DBSCAN FOR CLASSICAL IDENTIFICATION

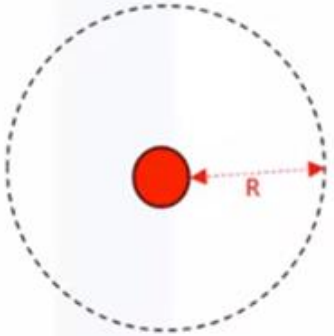


By Anil Kumar APSSDC

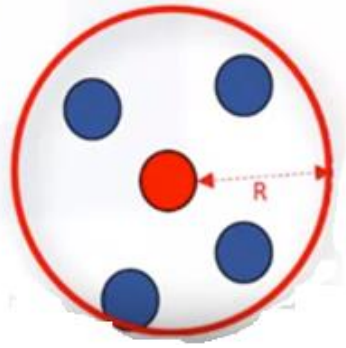
WHAT IS DBSCAN?

- DBSCAN (**D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise)
  - Is one of the most common clustering algorithms
  - Works based on density of objects

- R (**R**adius of neighborhood)
  - Radius (R) that if includes enough number of points within, we call it a dense area

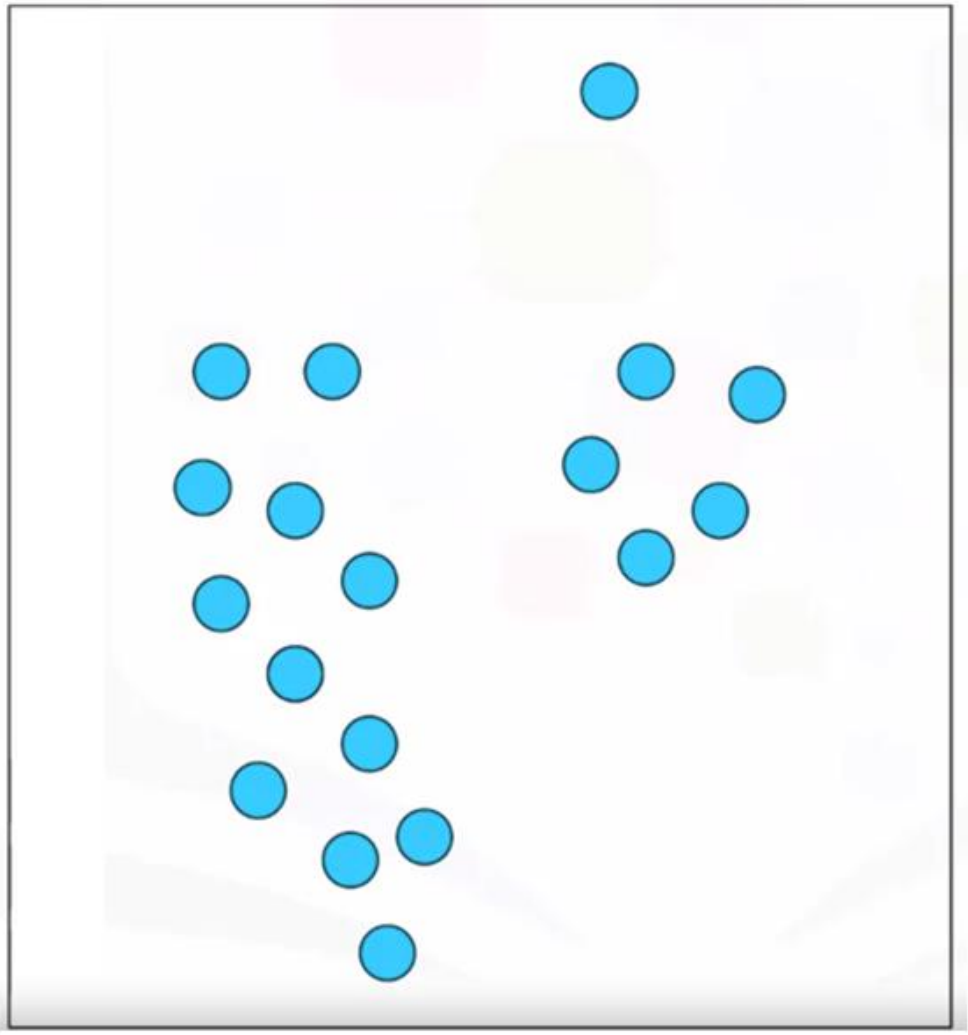


- M (**M**in number of neighbors)
  - The minimum number of data points we want in a neighborhood to define a cluster



By Anil Kumar APSSDC

HOW DBSCAN WORKS



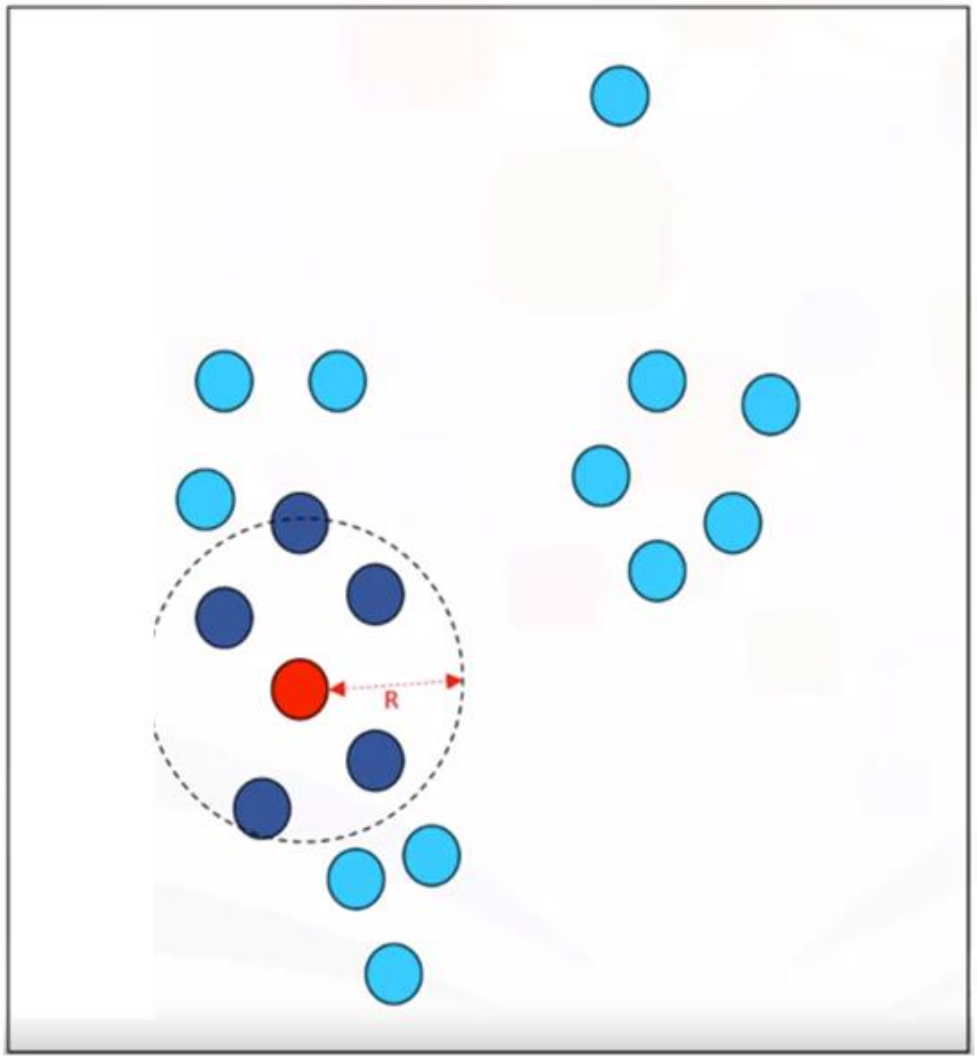
Each point is either:

- *core point*
- *border point*
- *outlier point*

R = 2unit , M = 6

By Anil Kumar APSSDC

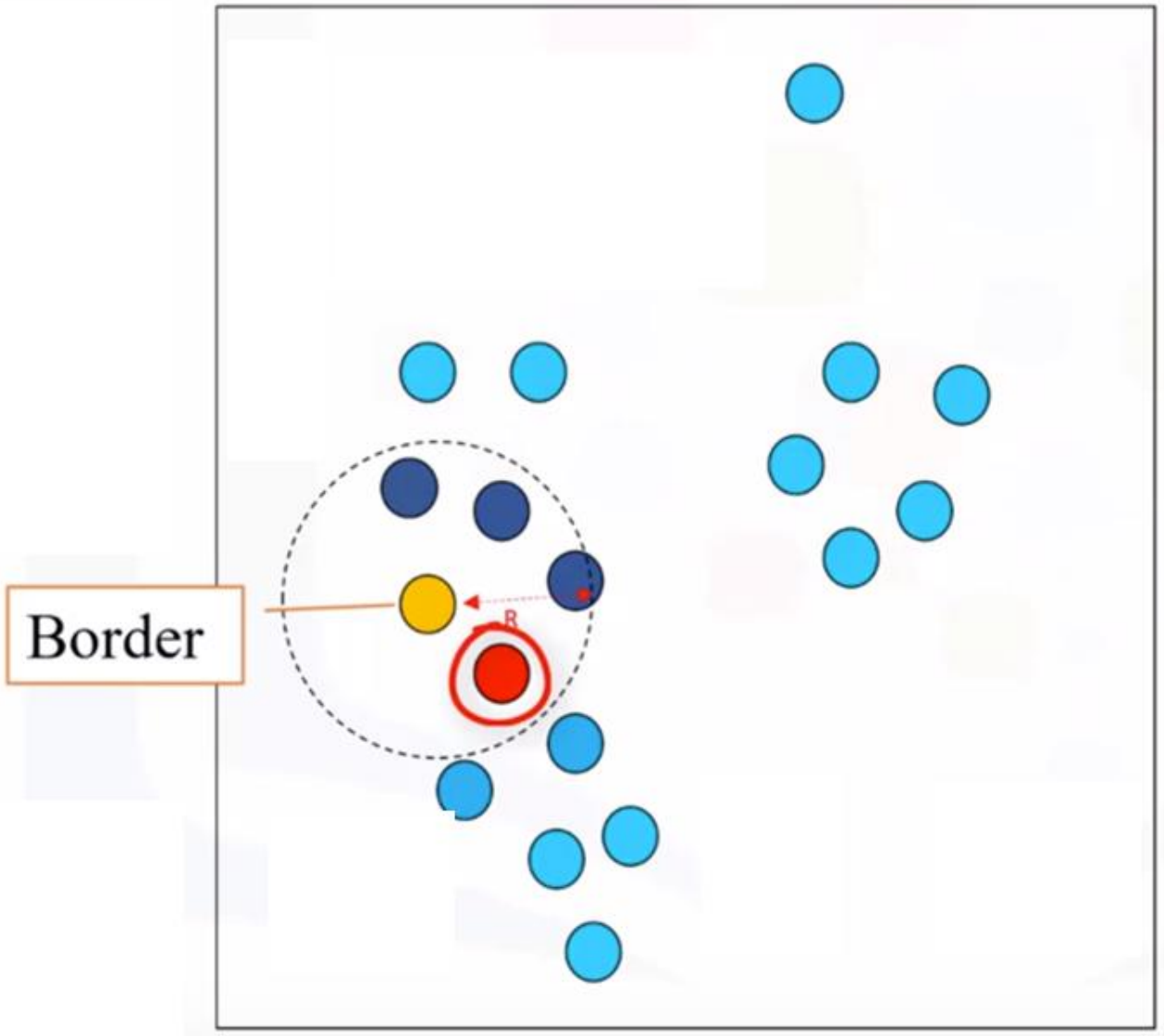
DBSCAN ALGORITHM - CORE POINT



R = 2unit , M = 6

By Anil Kumar APSSDC

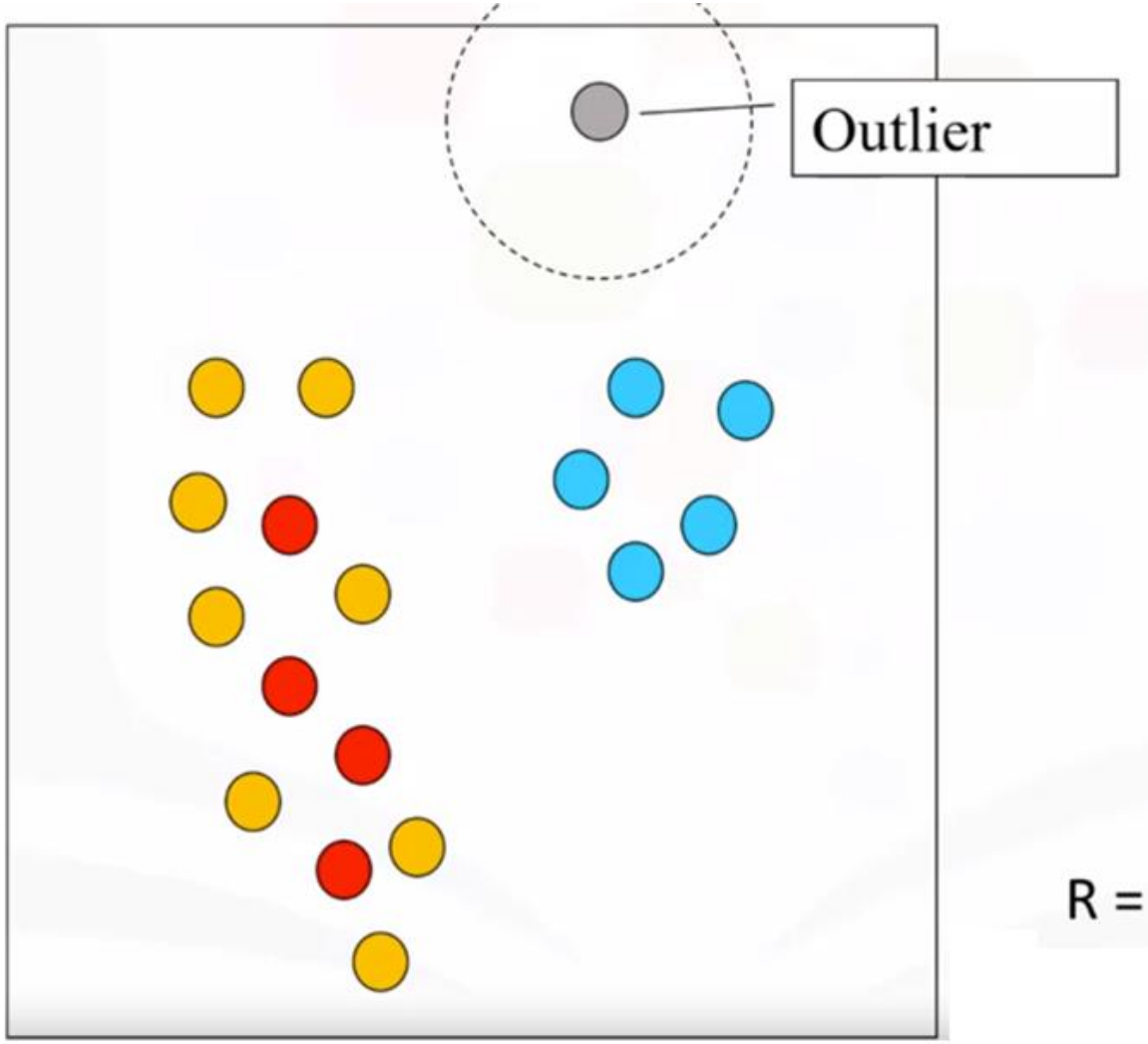
DBSCAN ALGORITHM - BOARDER POINTS?



R = 2unit , M = 6

By Anil Kumar APSSDC

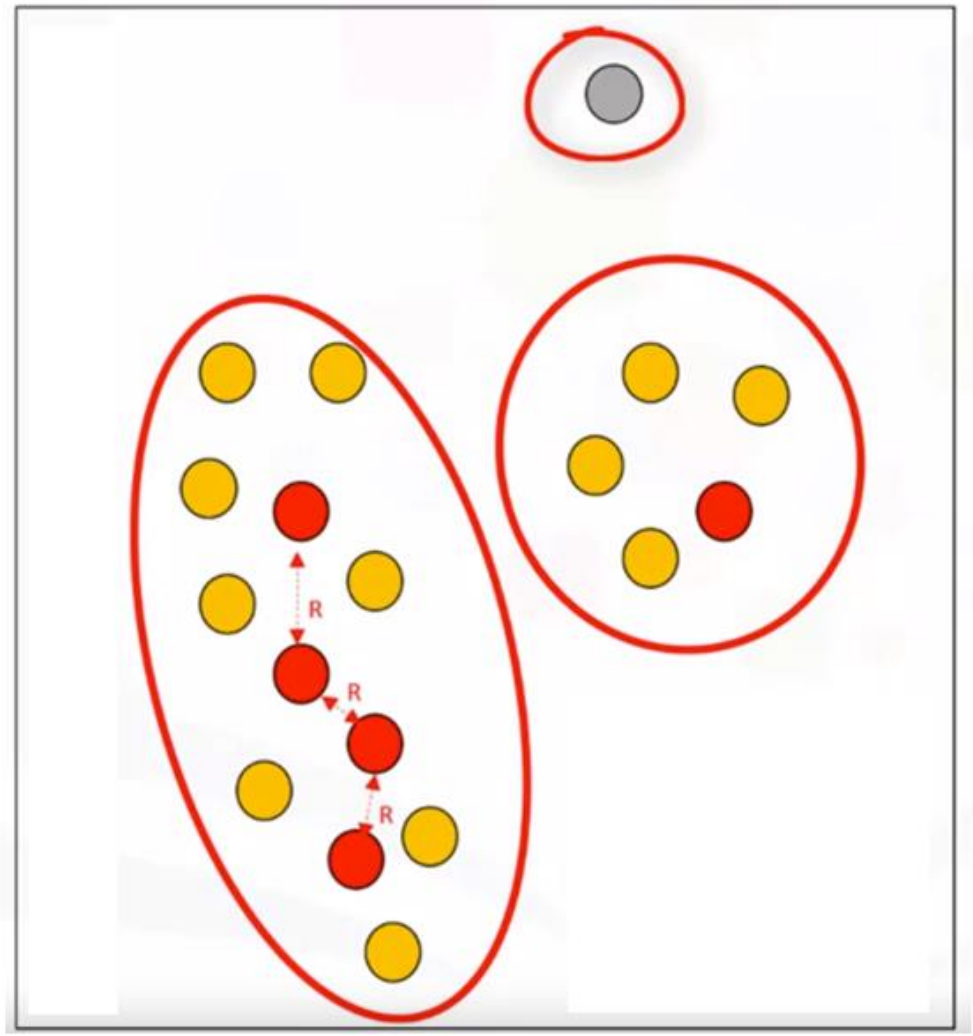
DBSCAN ALGORITHM - OUTLIERS



R = 2unit , M = 6

By Anil Kumar APSSDC

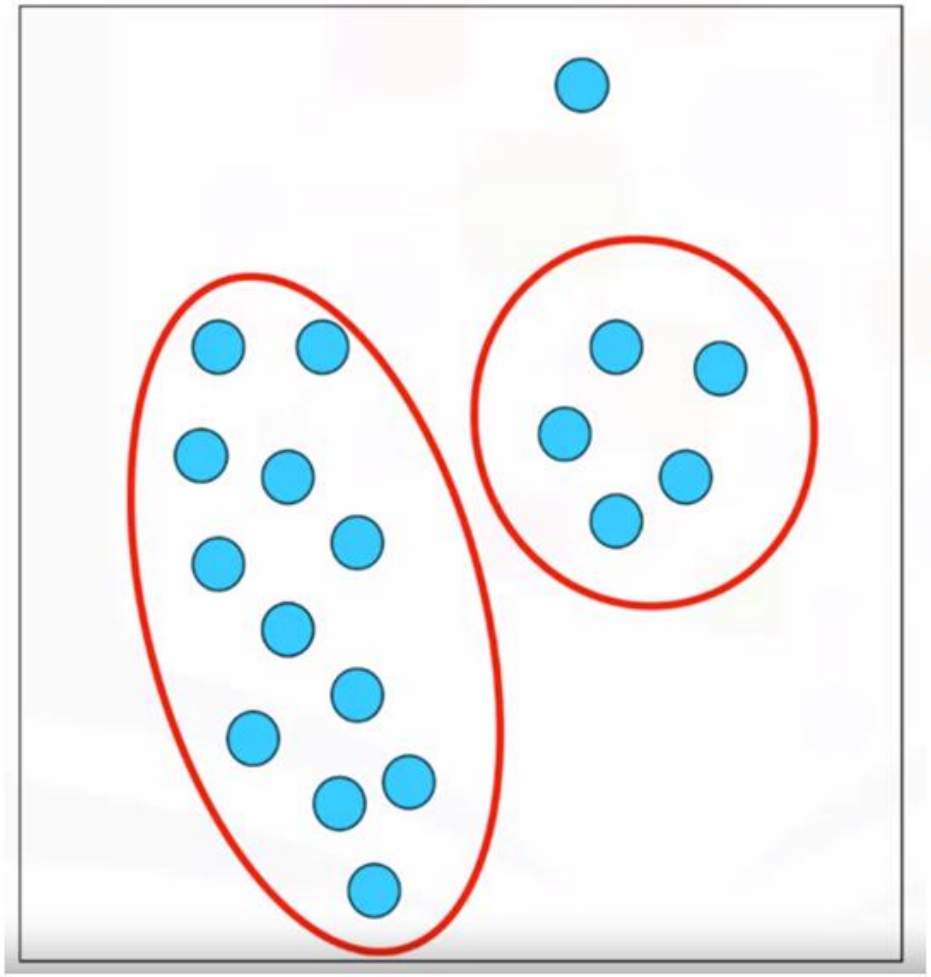
DBSCAN ALGORITHM – CLUSTERS?



R = 2unit , M = 6

By Anil Kumar APSSDC

ADVANTAGES OF DBSCAN



1. Arbitrarily shaped clusters
2. Robust to outliers
3. Does not require specification of the number of clusters