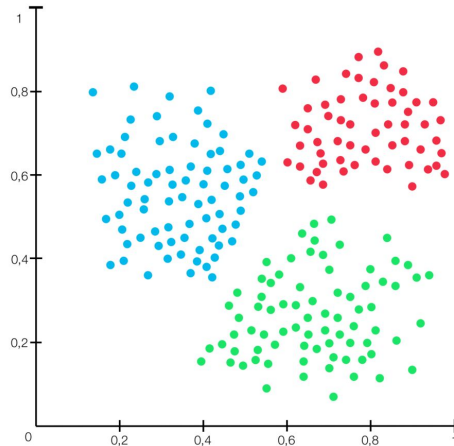
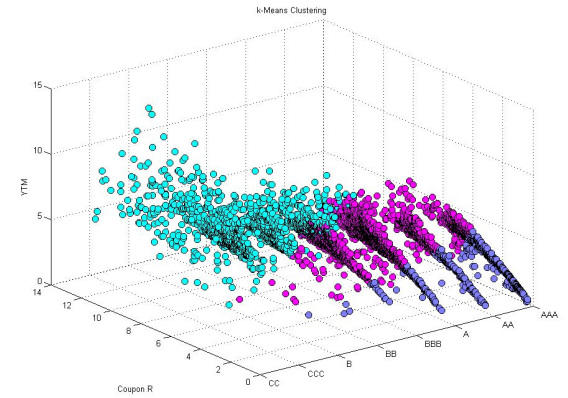


# Clustering with K-Means & DBSCAN



A thin, light brown L-shaped line that starts with a horizontal segment and then turns 90 degrees to become a vertical segment, framing the top-left corner of the title.

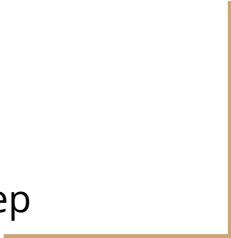
# What is Clustering?

Grouping **unlabeled examples** is called **clustering**.

A thin, light brown L-shaped line that starts with a horizontal segment and then turns 90 degrees to become a vertical segment, framing the bottom-right corner of the slide.



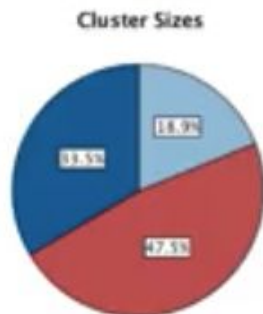
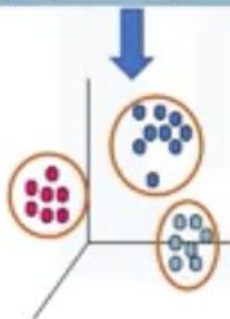
# Why do we need Clustering?

- Segmentation
  - Outlier Detection
  - Fraud Detection
  - Computer Vision
  - Pre-processing step
- 

# Clustering for segmentation

Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio	Defaulted
1	41	2	6	19	0.124	1.073	NBA001	6.3	0
2	47	1	26	100	4.582	8.218	NBA021	12.8	0
3	33	2	10	57	6.111	5.802	NBA013	20.9	1
4	29	2	4	19	0.681	0.516	NBA009	6.3	0
5	47	1	31	253	9.308	8.908	NBA008	7.2	0
6	40	1	23	81	0.998	7.831	NBA016	10.9	1
7	38	2	4	56	0.442	0.454	NBA013	1.6	0
8	42	3	0	64	0.279	3.945	NBA009	6.6	0
9	26	1	5	18	0.575	2.215	NBA006	15.5	1

Customer ID	Segment
1	YOUNG AND LOW INCOME
2	AFFLUENT AND MIDDLE AGED
3	AFFLUENT AND MIDDLE AGED
4	YOUNG AND LOW INCOME
5	AFFLUENT AND MIDDLE AGED
6	AFFLUENT AND MIDDLE AGED
7	YOUNG AND LOW INCOME
8	YOUNG AND LOW INCOME
9	AFFLUENT AND MIDDLE AGED



Cluster  
cluster-1  
cluster-2  
cluster-3



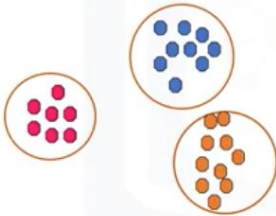
Cluster	Segment Name
cluster-1	AFFLUENT AND MIDDLE AGED
cluster-2	YOUNG EDUCATED AND MIDDLE INCOME
cluster-3	YOUNG AND LOW INCOME



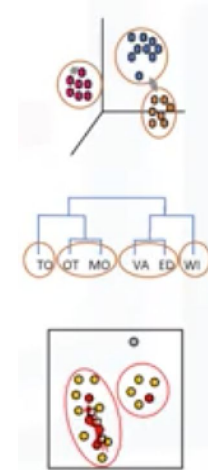
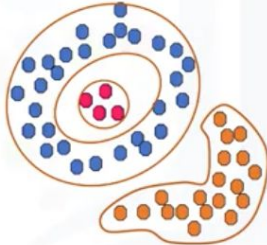
# Clustering Algorithms

1. **Partition-based Clustering** (k-Means, Fuzzy c-Means, etc.)
  - Efficient when used for Medium or Large sized datasets
2. **Hierarchical Clustering** (Agglomerative, Divisive)
  - Produces Trees of Clusters
  - Small sized datasets
3. **Density-based Clustering** (DBSCAN)
  - Produces **Arbitrary shaped** clusters
  - Efficient when there is **Noise** in dataset

• Spherical-shape clusters



• Arbitrary-shape clusters





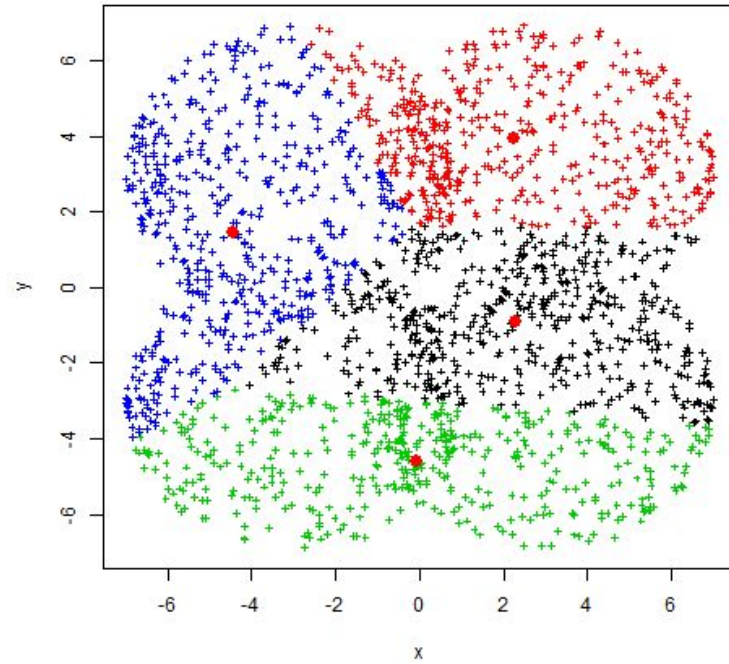
k-Means



# k-Means algorithm steps

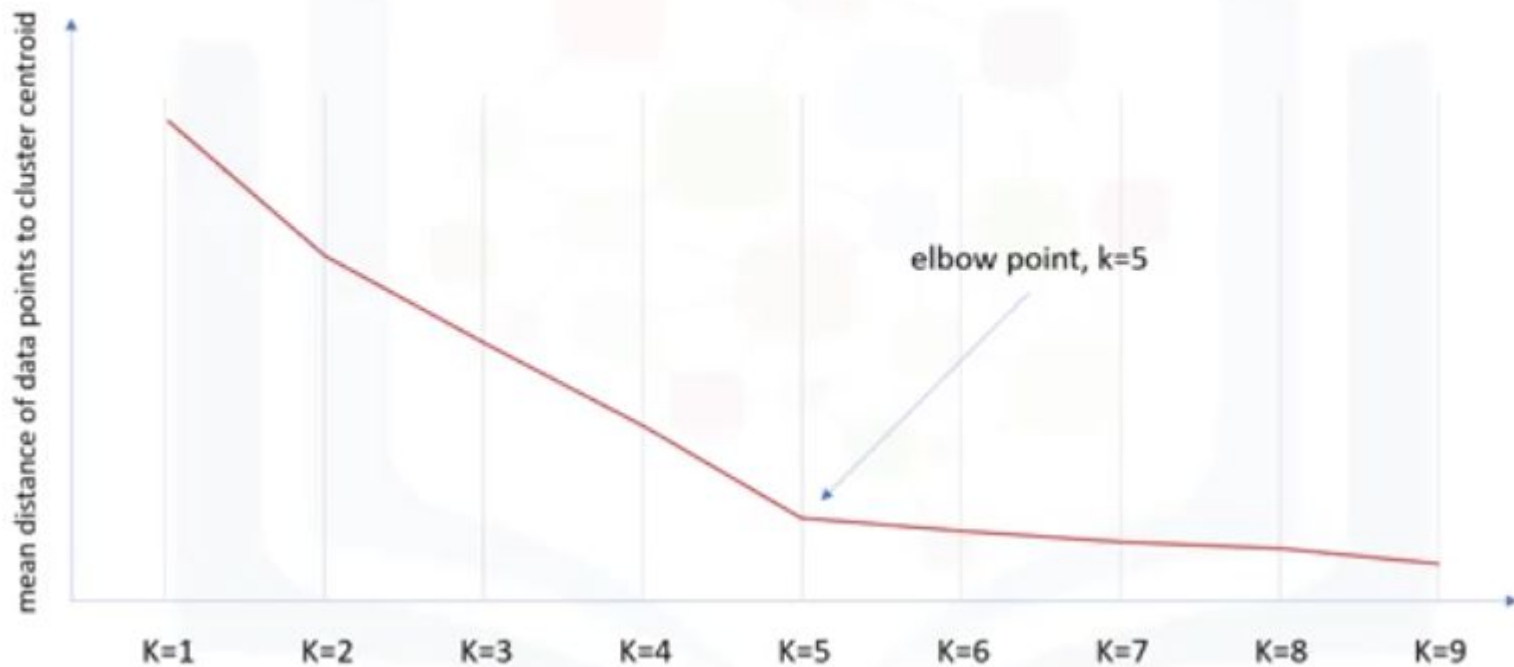
1. Define num of clusters
  - a. Initialize k value
2. Calculate distance of each data point from the centroids
3. Assign each point to closest centroid
  - b. Clusters produced
4. Move the existing centroids to its clusters's data points means.
5. Repeat steps 2-3-4 until no centroids will move

### K Means Clustering





# Choosing k



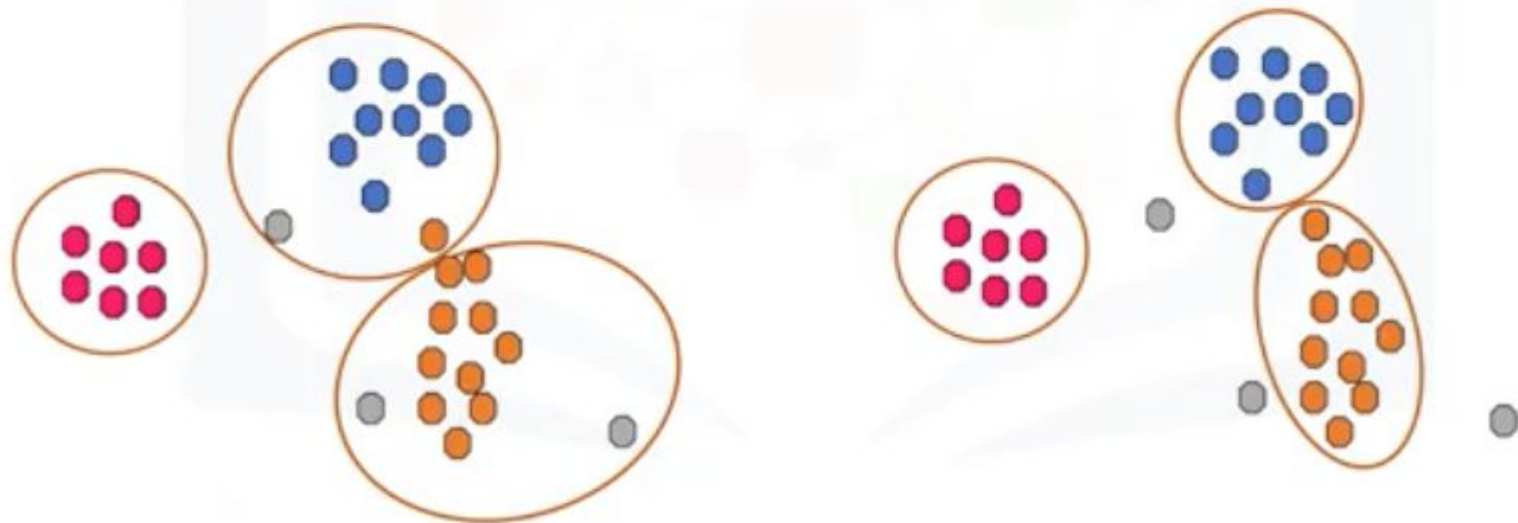


DBSCAN



## k-Means Vs. density-based clustering

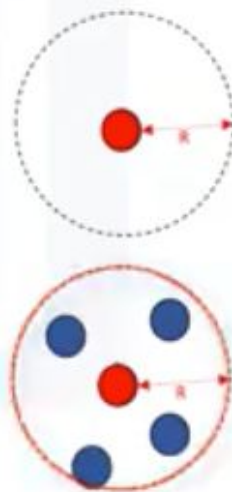
- k-Means assigns all points to a cluster even if they do not belong in any
- Density-based Clustering locates regions of **high density**, and separates outliers



# What is DBSCAN?

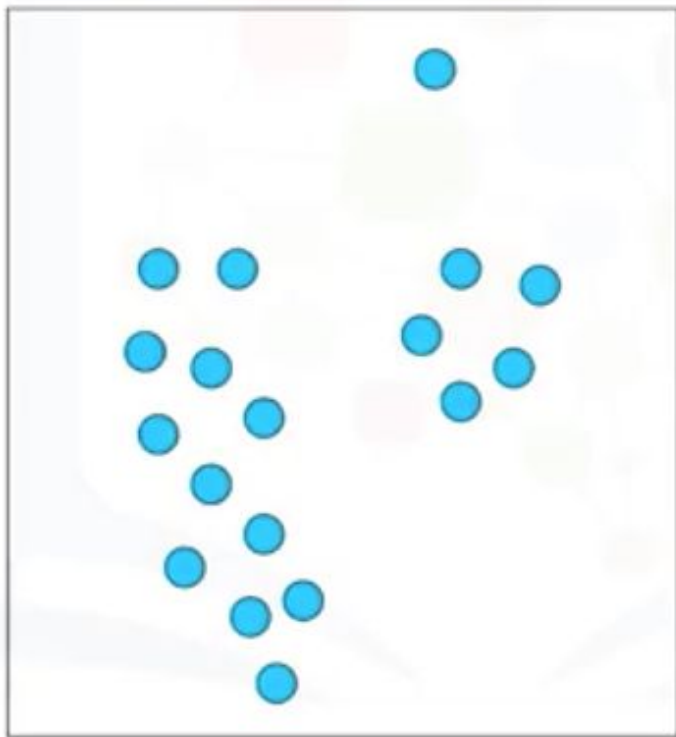
---

- 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



# How DBSCAN works

---

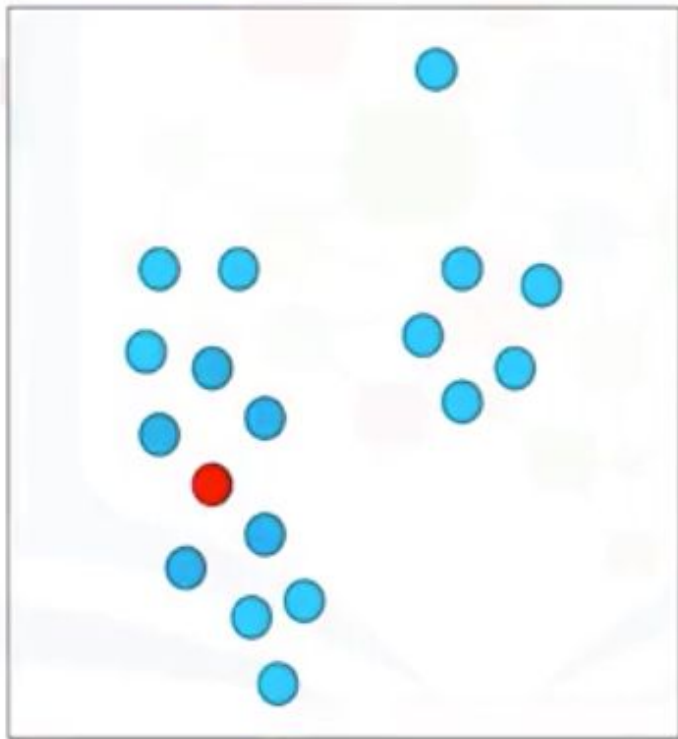


Each point is either:

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

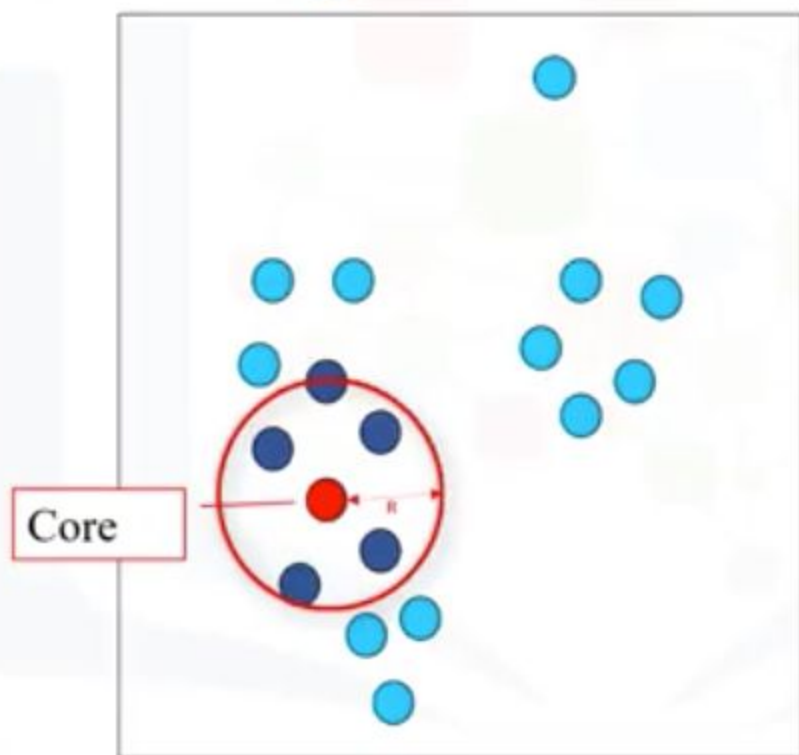
$R = 2\text{unit}$  ,  $M = 6$

## DBSCAN algorithm – core point?



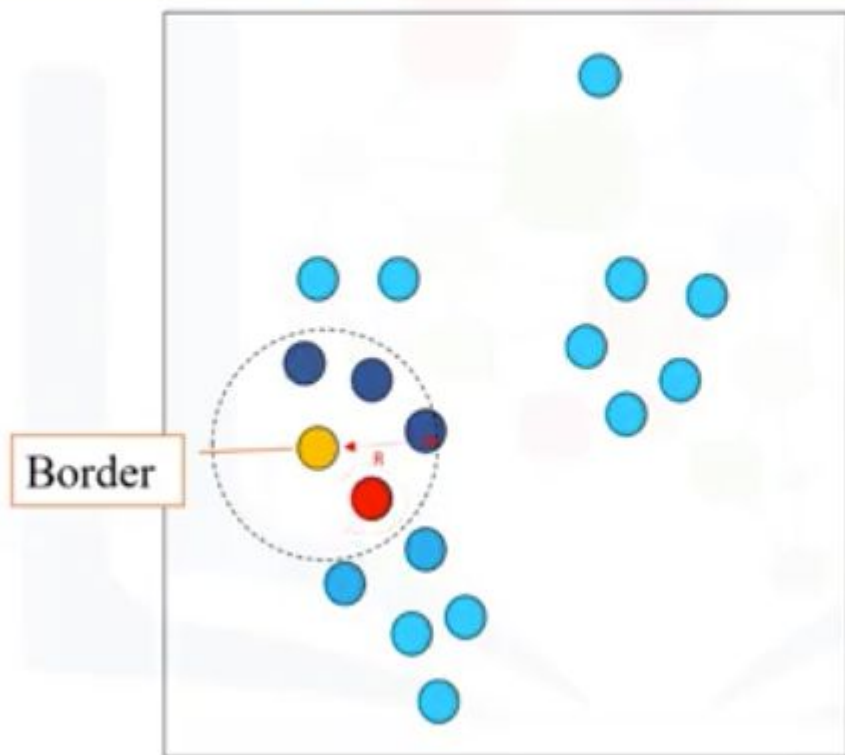
$R = 2\text{unit}$  ,  $M = 6$

## DBSCAN algorithm – core point



$R = 2\text{unit}$  ,  $M = 6$

## DBSCAN algorithm – border points?



$R = 2\text{unit}$  ,  $M = 6$

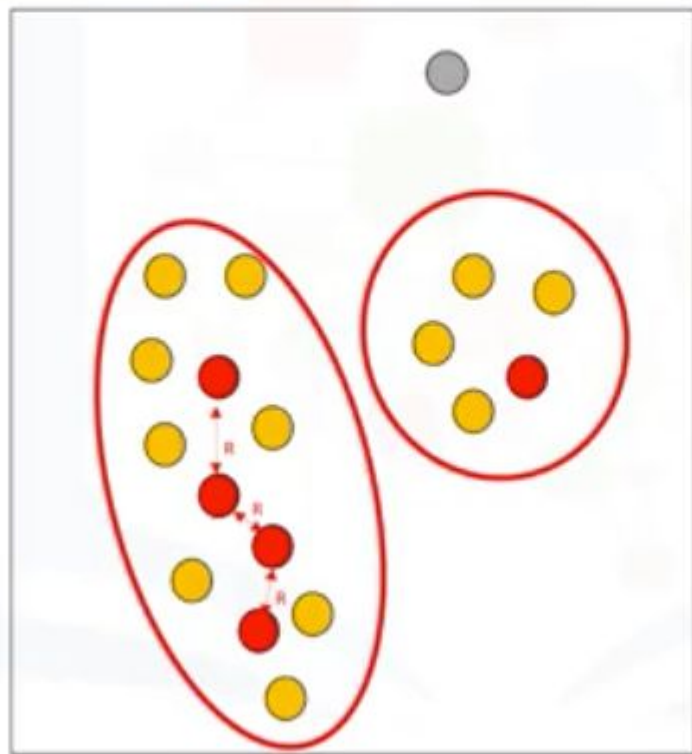


# DBSCAN algorithm – outliers



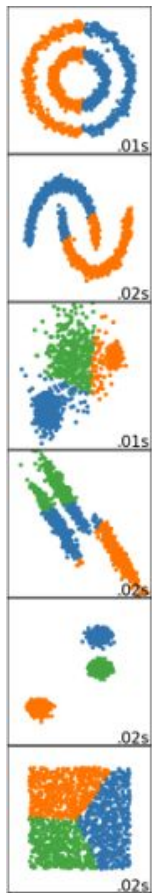
$R = 2\text{unit}$  ,  $M = 6$

# DBSCAN algorithm – clusters?



$R = 2\text{unit}$  ,  $M = 6$

# k-Means



# DBSCAN

