Data Mining and Warehousing

Prof. Zalak Kansagra, Assistant Professor
Computer Science & Engineering

CHAPTER - 7

Clustering

Unsupervised learning

- It can be considered as a self learning process.
- Discovering patterns from data without any labels.
- E.g. Students with all study material but no faculty to guide.

Image source : Google

Cluster Analysis

- Goal is to form groups with some similarity.
  - Most similar within group
  - Least similar among group
- E.g. Grouping of students studying similar subjects.
- E.g. Grocery grouped together based on its category.

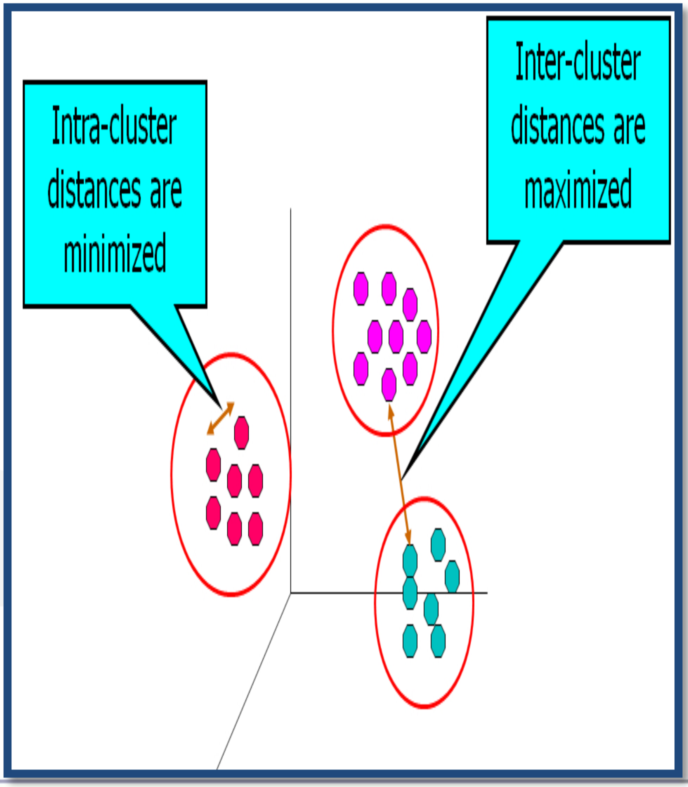


Image source : Google

The Clustering Example

Goal: To make 3 marketing strategies

Age (in years)

Engagement with the page (in days/week)

Age: 42 Eng. 7

Age: 18 Eng. 3

Age: 23 Eng. 2

Age: 49 Eng. 1

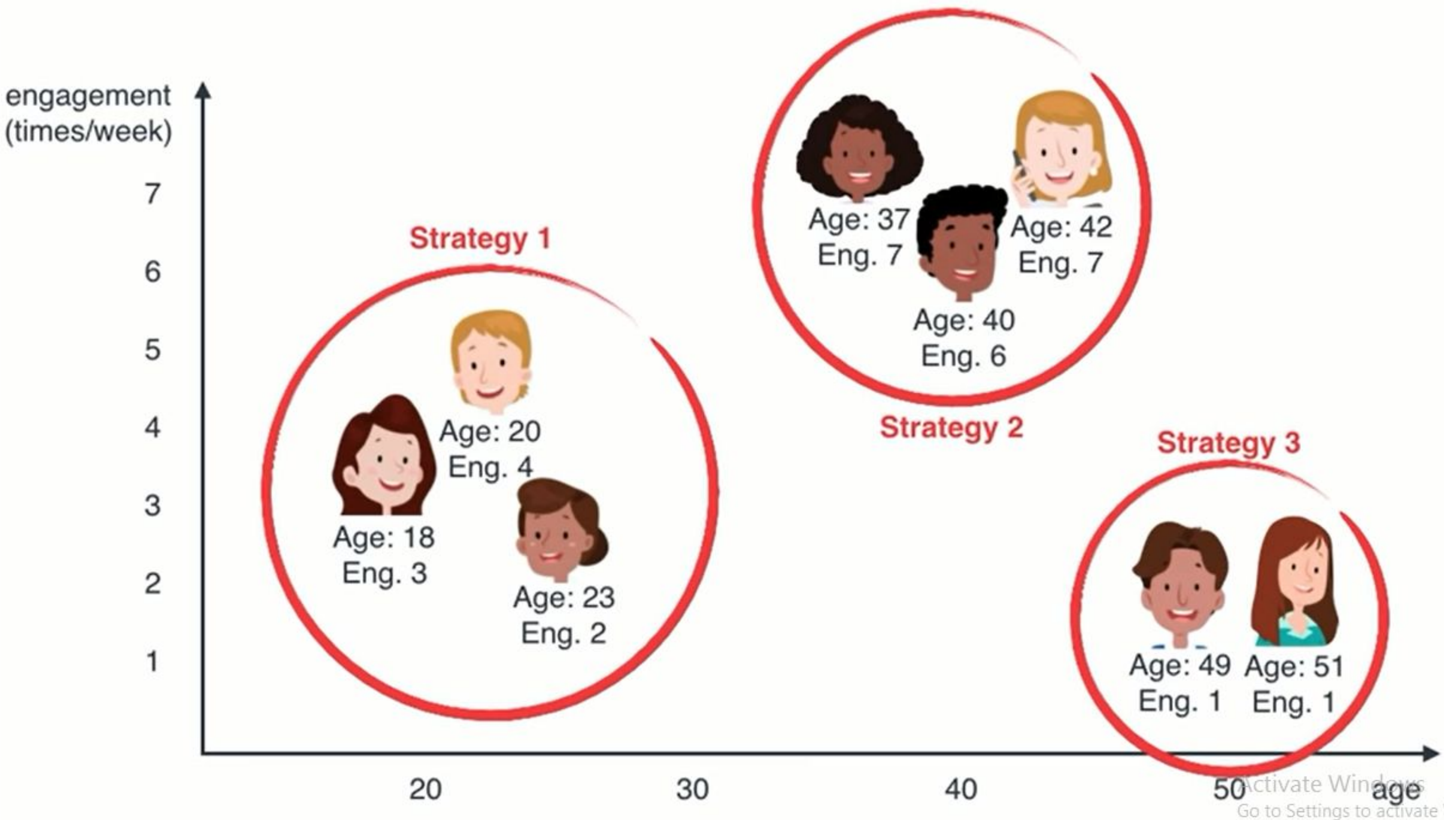
Age: 37 Eng. 7

Age: 51 Eng. 1

Age: 40 Eng. 6

Age: 20 Eng. 4

The Clustering Example

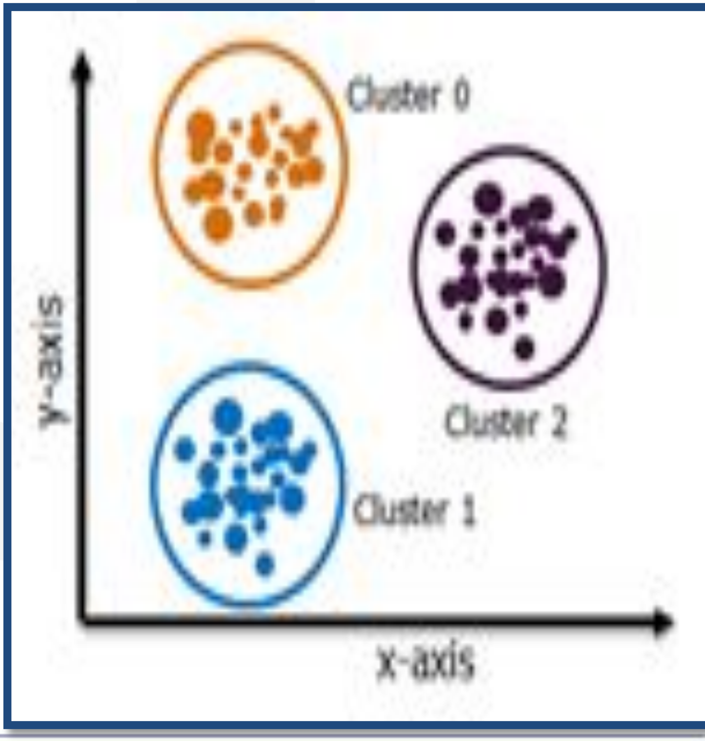


Partitioning Methods

- Partition a data of n items into set of K cluster such that sum of squared distance is minimized.

$$E = \Sigma_{i=1}^{k} \Sigma_{p \in C_i} (p - c_i)^2$$

c_i – centroid of cluster



- Two methods
  - K means
  - k medoids

Image source : Google

The K-Means Clustering Method

- Given k number of clusters.

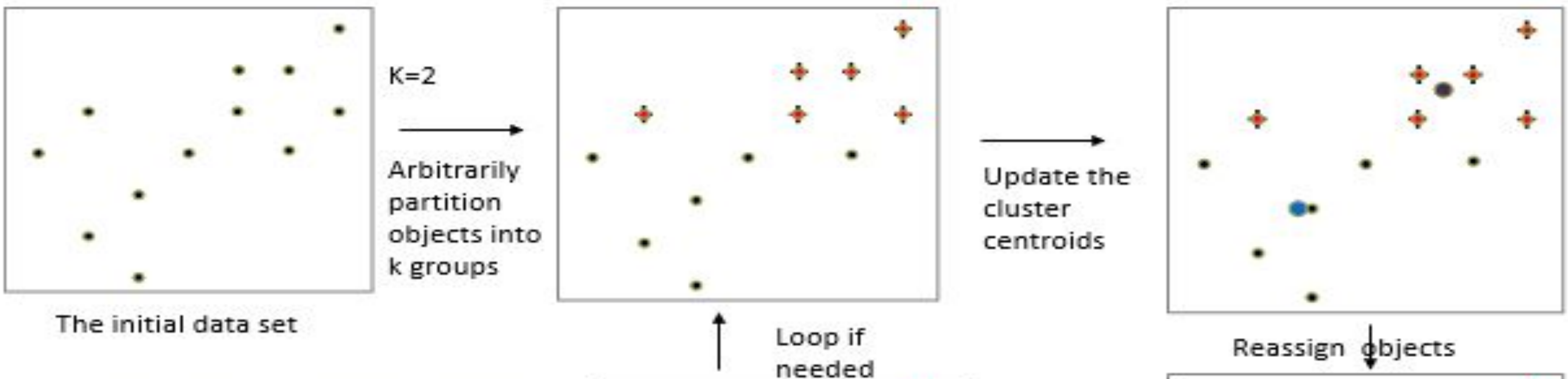
  - Partition given items in k nonempty subsets.

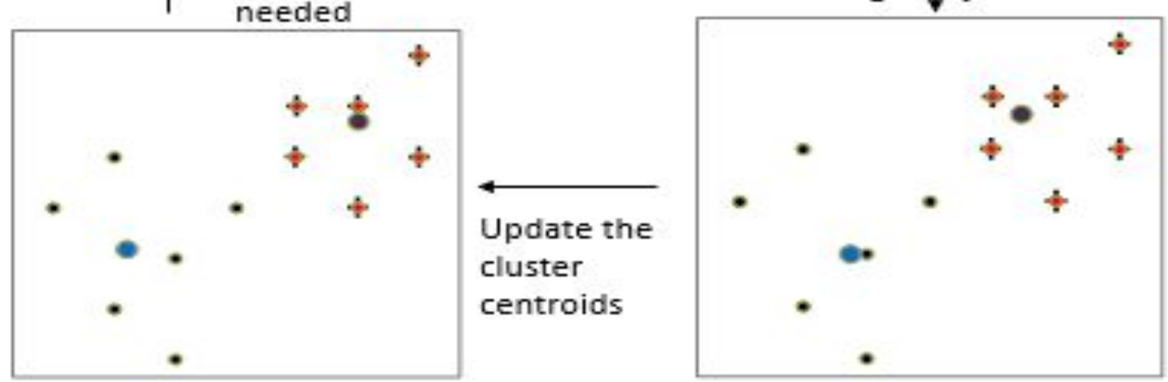
  - Compute the centre of current partition.

  - Assign each item to the cluster with nearest centre.

  - Perform step two again until the center doesn't change.

The K-Means Clustering Example





Partition objects into *k* nonempty subsets

Repeat

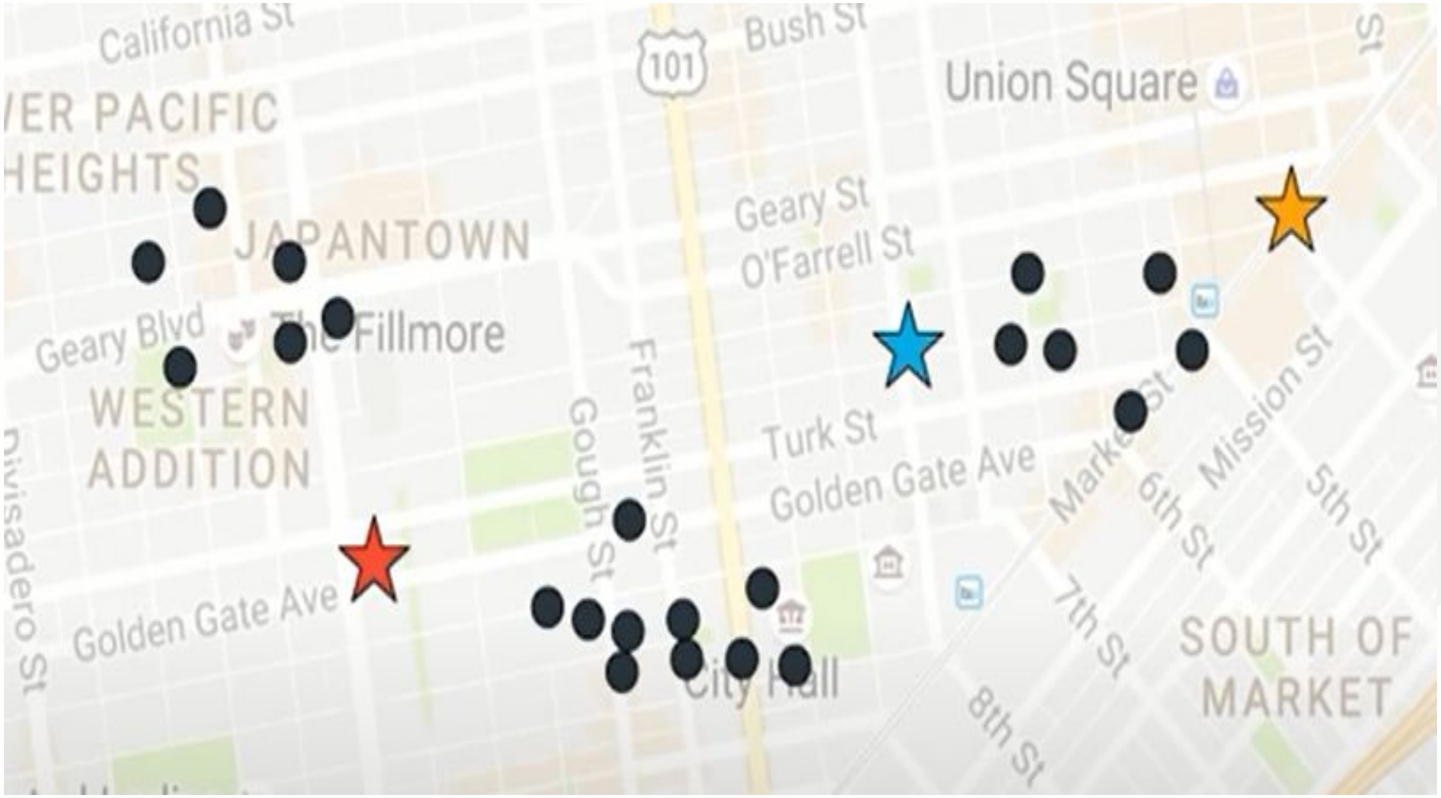
- Compute centroid (i.e., mean point) for each partition
- Assign each object to the cluster of its nearest centroid

Until no change

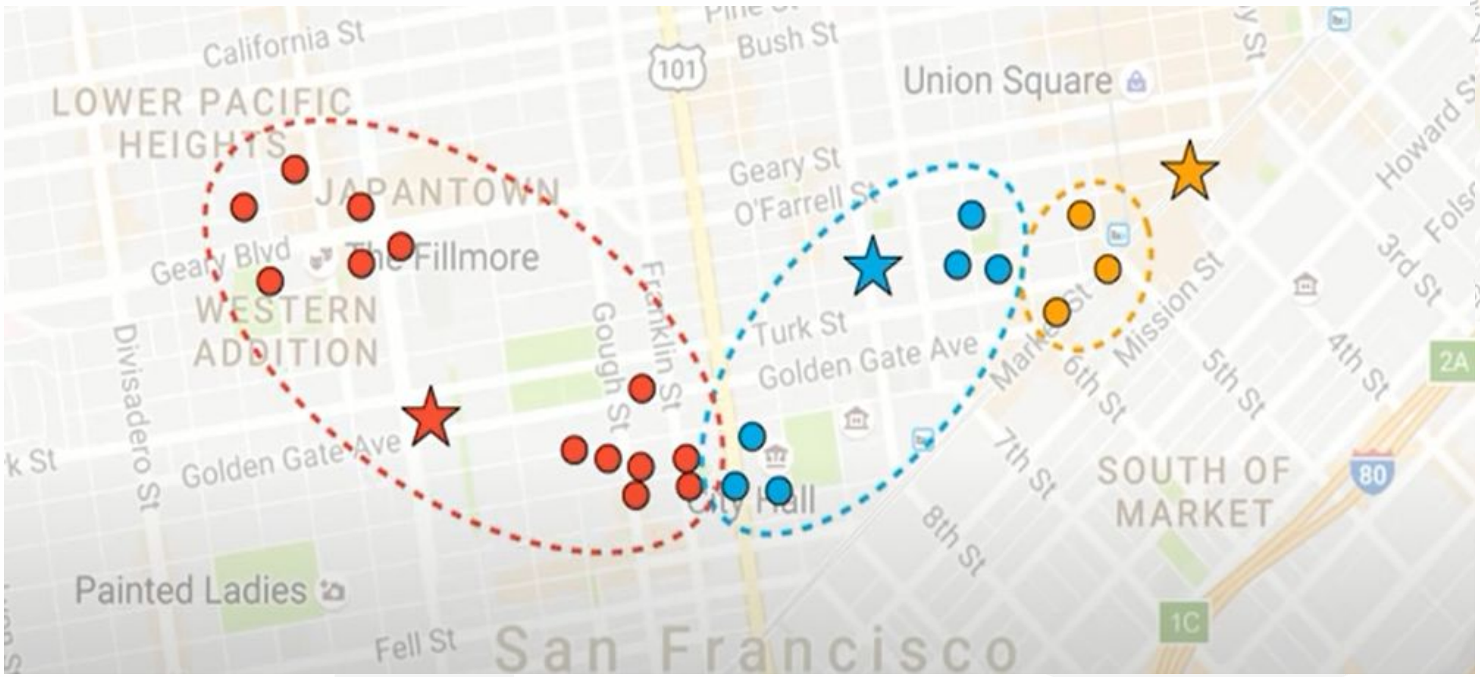
Parul® University

DIGITAL LEARNING CONTENT

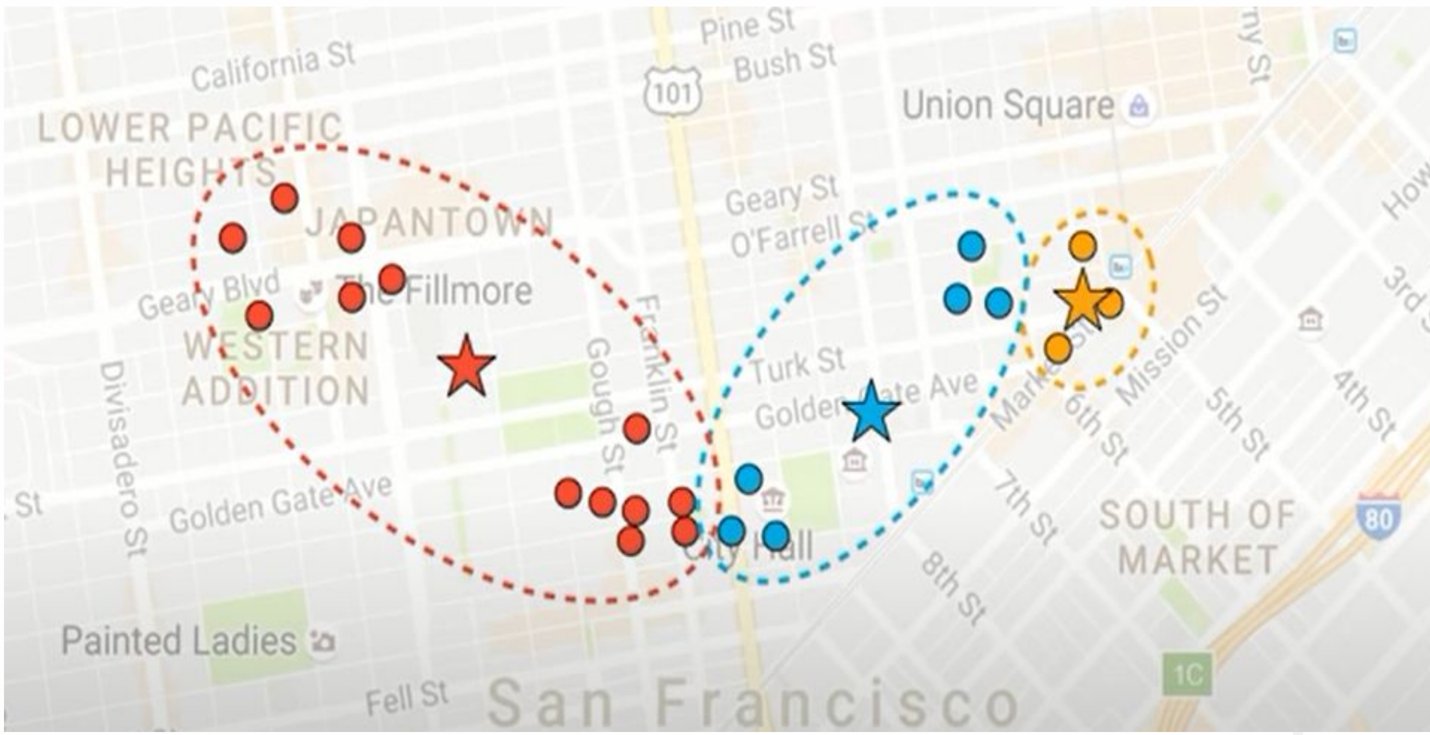
The K-Means Clustering Example



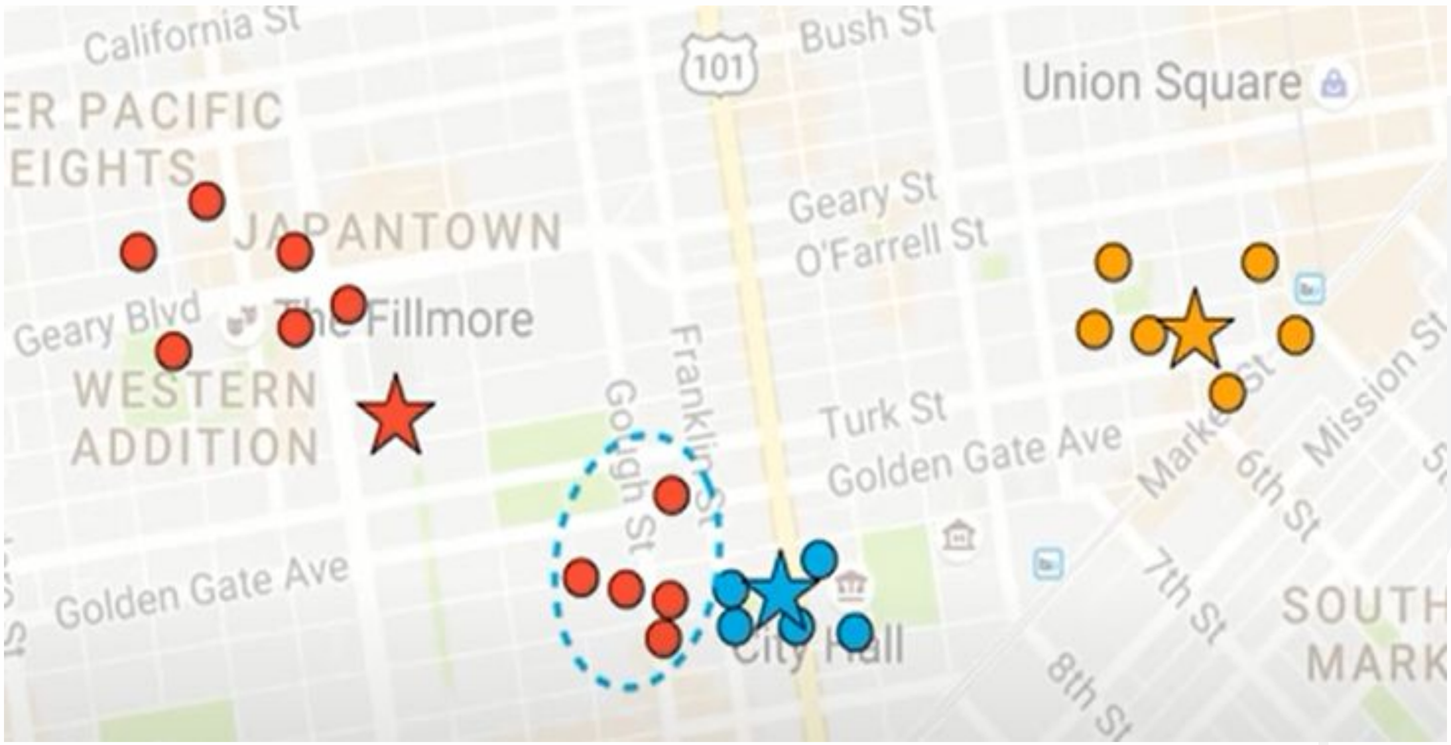
The K-Means Clustering Example



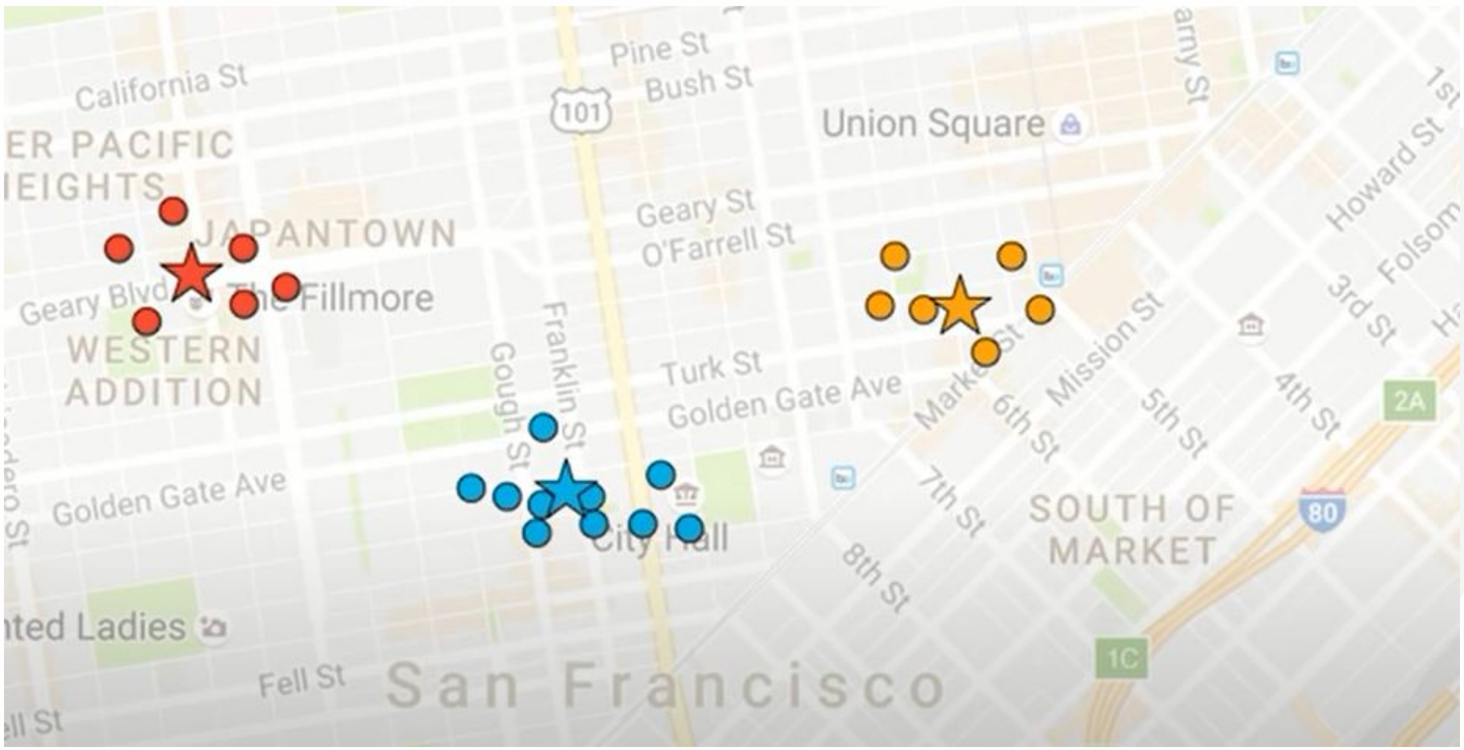
The K-Means Clustering Example



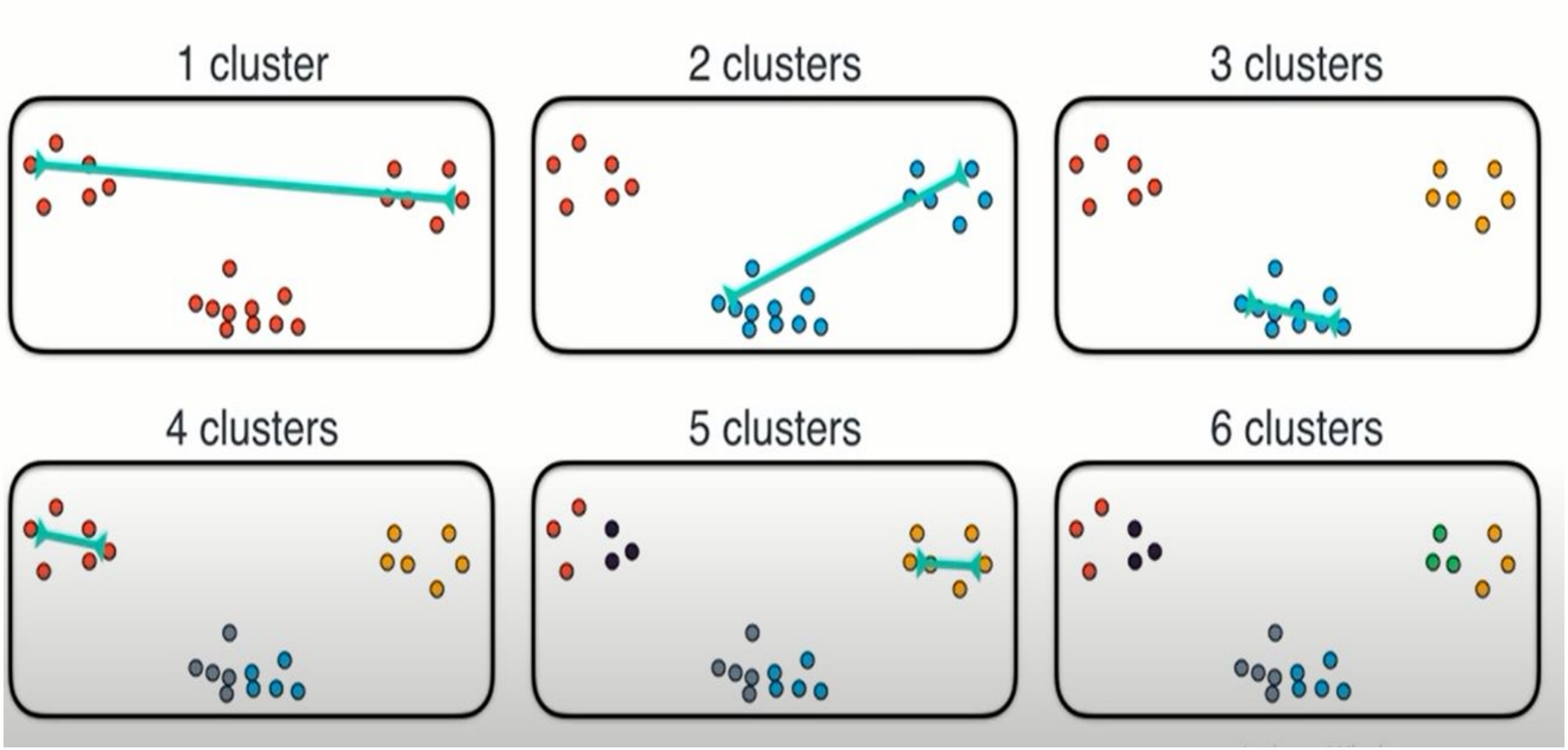
The K-Means Clustering Example



The K-Means Clustering Example



Choosing K- Elbow method



The K-Medoid Clustering Method

- Given k number of clusters.

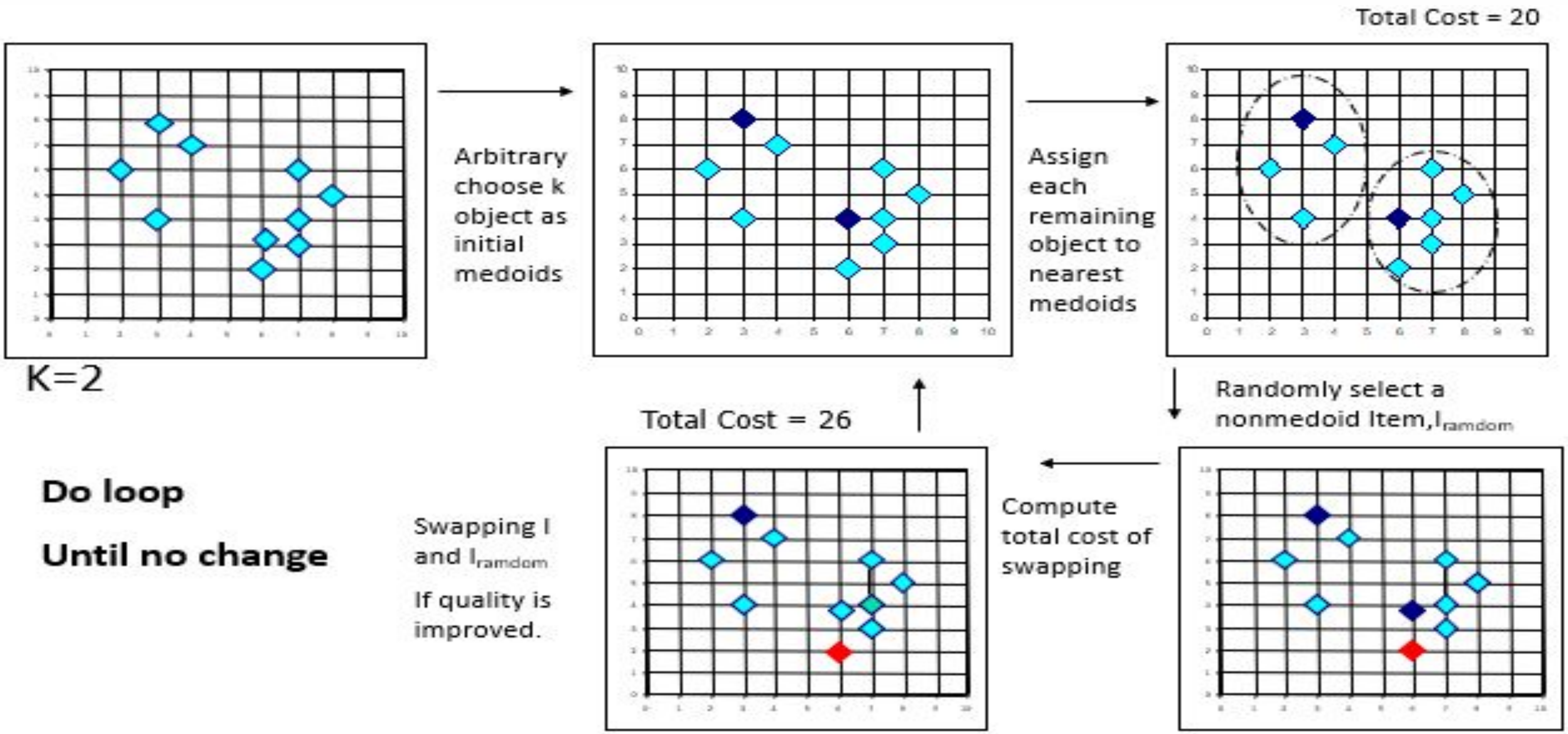
  - Select k random items out of n as medoid.

  - Assign each item to the nearest medoid using any distance matric method.

  - If the cost decline.

    - for all medoid m with items I which are not medoid.
    - Swap m and I and assign each item to the nearest medoid.
    - Recompute the cost.
    - If cost more than previous undo previous step

The K-Medoid Clustering Example



Hierarchical Clustering

- Groups data into tree like clusters.
- Treats every data as a different cluster.
- Perform following steps
  - Finding two clusters that can be nearest to each other.
  - Merge maximum two approximately similar clusters.
  - Continue above step till all the clusters are merged.
- It aims at producing hierarchy like nested clusters.

Agglomerative Methods

- Calculating similarity among clusters.
- Every data point is taken as an individual cluster.
- Merging clusters with higher proximity among each other.
- Recompute the similarity matrix for each cluster
- Repeat above two steps till only a single cluster is left.
- Follow Bottom up Approch

Agglomerative Methods Example

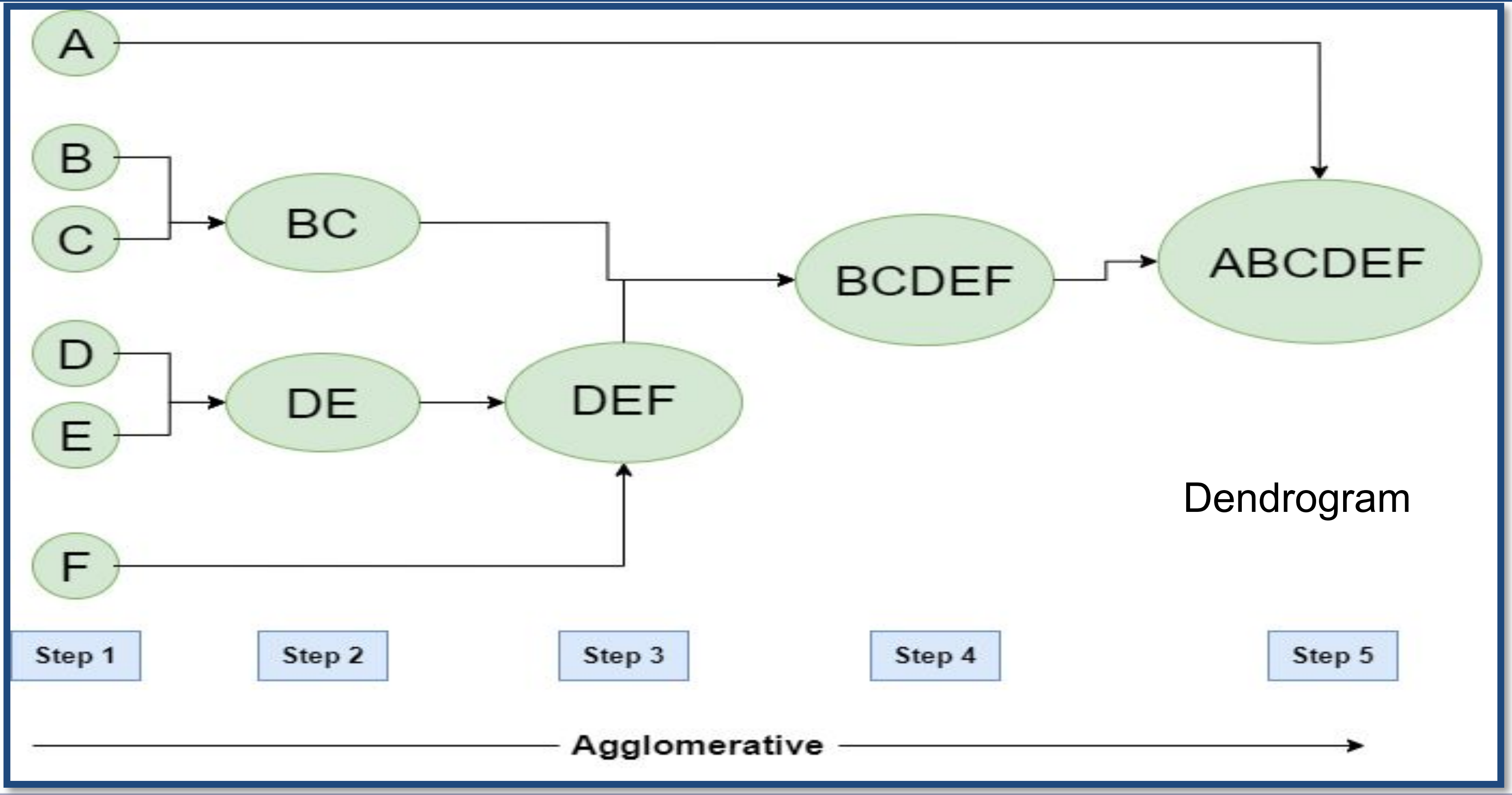


Image source : Google

Divisive Methods

- Opposite of Agglomerative method.
- All data points are initially considered as a single cluster.
- After every iteration the data points are separated from the cluster that doesn't show any similarity.
- It results into n clusters at the end.

Divisive Methods Example

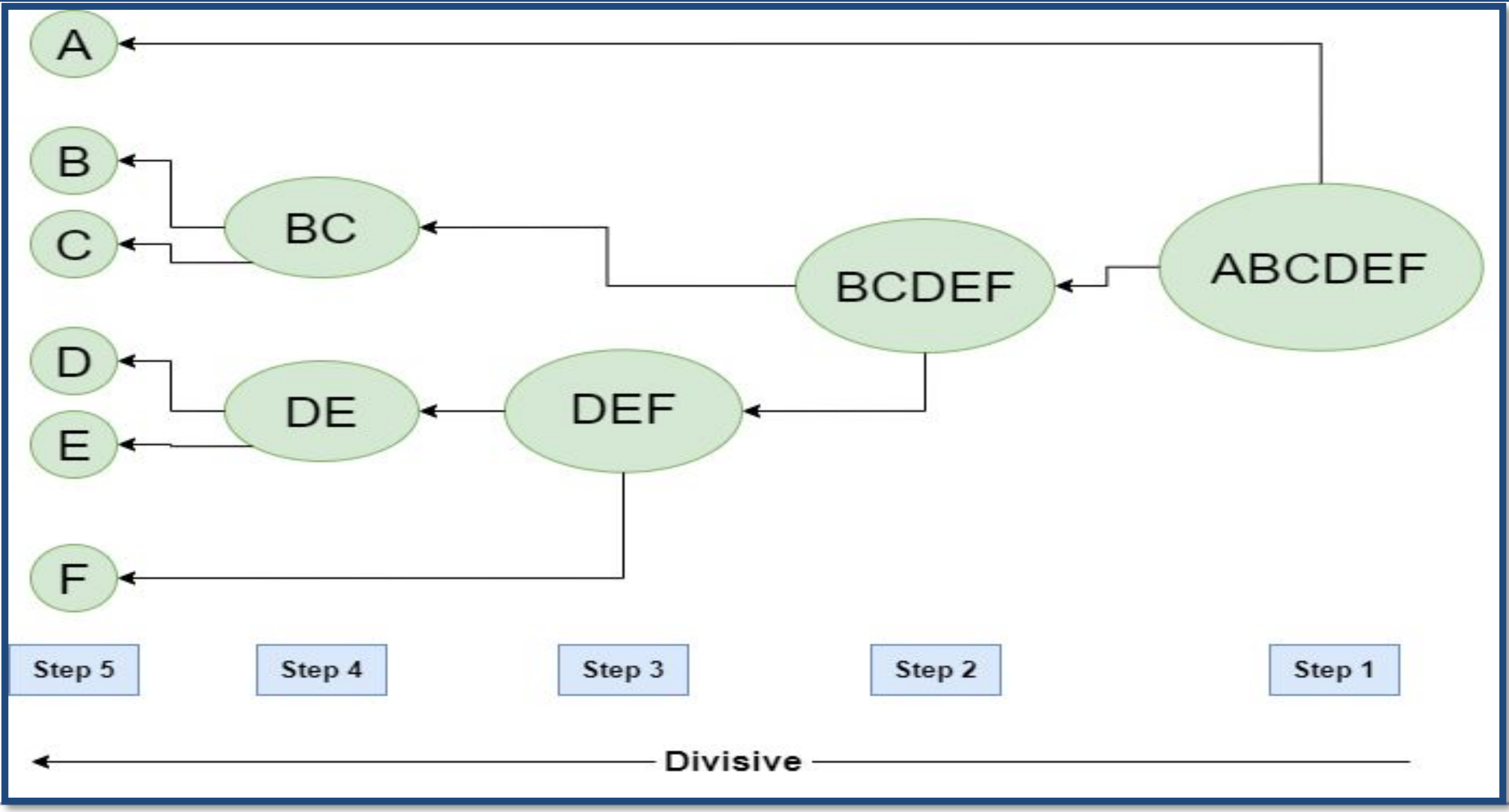


Image source : Google

Density-Based Clustering DBSCAN

Density Based Spatial Clustering of Applications with noise

- Used to identify clusters of arbitrary shape.
- Requires two parameter.
  - **Eps (**epsilon**) -** Identify neighbor of a data point.
  - defines the radius of neighborhood around a point x. It's called called the ϵ-neighborhood of x.
  - For distance smaller or equal to eps, the data points become neighbour.
  - **K-distance graph** is used to find the value of eps.

Density-Based Clustering DBSCAN

- **MinPts (**minimum points**)–** minimum numbers of points within eps radius.
- Any point x in data set, with a neighbour count greater than or equal to *MinPts*, is marked as a ***core point***.
- x is ***border point***, if the number of its neighbors is less than MinPts.
- Its value can found from dimension of dataset.
- MinPts = D+1
- The larger the data set, the larger the value of minPts should be chosen. minPts must be chosen at least 3.

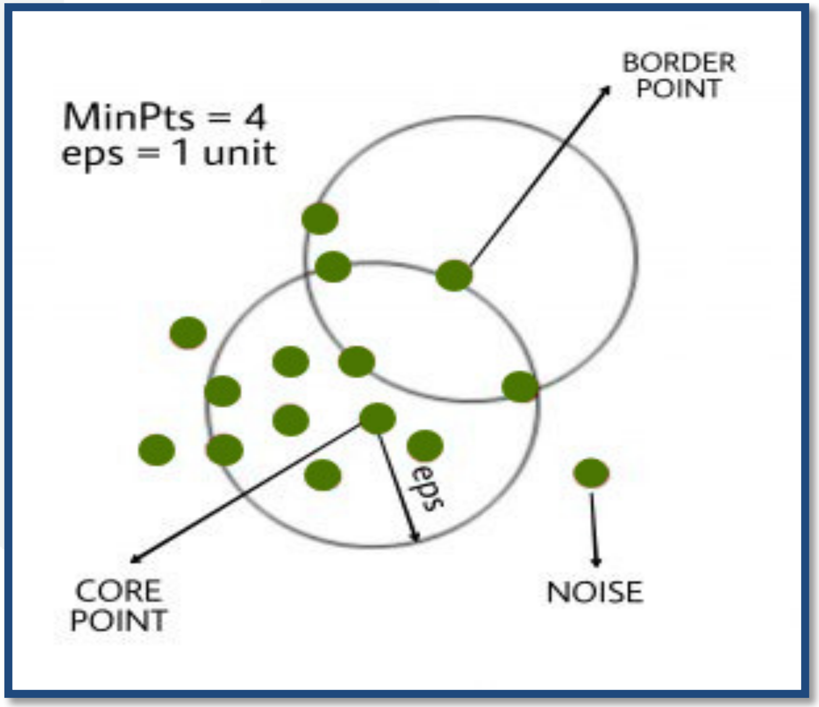


Image source : Google

DBSCAN Reachability

Direct density reachable: A point "A" is directly density reachable from another point "B" if: **i)** "A" is in the ϵ-neighborhood of "B"

And **ii)** "B" is a core point.

Density reachable: A point "A" is density reachable from "B" if there are a set of core points leading from "B" to "A". ie. there is a chain of objects b_1, b_2..., b_n, with b_1=a, b_n=b such that b_{i+1} is directly density-reachable from b_i w.r.t ε and *MinPts* for all 1 <= i <= n

Density connected: Two points "A" and "B" are density connected if there are a core point "C", such that both "A" and "B" are density reachable from "C".

Density-Based Clustering DBSCAN

- For each point xi, compute the distance between xi and the other points.
- Finds all neighbor points within distance eps of the starting point (xi). Each point, with a neighbor count greater than or equal to MinPts, is marked as core point or visited.
- For each core point, if it's not already assigned to a cluster, create a new cluster.
- Find recursively all its density connected points and assign them to the same cluster as the core point.
- Iterate through the remaining unvisited points in the data set.
- Those points that do not belong to any cluster are treated as outliers or noise.

DBSCAN Characterstics

- Unlike K-means, DBSCAN does not require the user to specify the number of clusters to be generated.
- DBSCAN can find any shape of clusters. The cluster doesn't have to be circular.
- DBSCAN can identify outliers.

Evaluation of Clustering

- Three evaluation factors using which clustering is evaluated.
  - Clustering Tendency
  - Number of Clusters
  - Clustering Quality

Clustering Tendency

- Non uniformity among data points is vital for clustering.
- Measuring the probability of data points generated by uniform data distribution.
- Null Hypothesis :- Non random uniform data distribution
- Alternate Hypothesis :- Random data generation.
- For H>0.5 reject null hypothesis as data contains cluster.
- For H closer to 0, no clustering tendency.

Number of Clusters

- Correct number of clusters depends on
  - Distribution shape
  - Scale in data set.
  - Clustering resolution
- Two approach for finding optimal number of clusters.
  - Domain Knowledge
  - Data driven approach

Number of Clusters

- Domain Knowledge
  - Gives initial knowledge on forming number of clusters.
- Data driven approach

•Data Driven Approach
  •Empirical Method

  •Elbow Method

Quality of Clustering

- Characteristic of cluster :- minimum intra cluster distance

  maximum inter cluster distance

- Two types of measures
  - Extrinsic Measures :- True labels required.
  - Intrinsic Measures :- True labels not required.

Outlier Detection

- Values that deviate from other values resulting into some suspicion.
- Two Types
  - Univariate :- can be identified looking at one dimensional space.
  - Multivariate :- identified in n dimensional space.
- Other characteristics
  - Point outlier
  - Contextual outliers
  - Collective outliers.

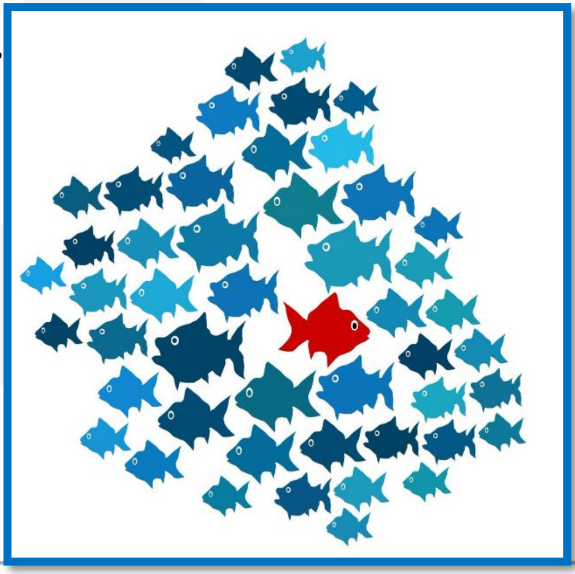


Image source : Google

Numerical

- Cluster the following eight points (with (x, y) representing locations) into three clusters:

A1(2, 10), A2(2, 5), A3(8, 4), A4(5, 8), A5(7, 5), A6(6, 4), A7(1, 2), A8(4, 9)

Initial cluster centers are: A1(2, 10), A4(5, 8) and A7(1, 2).

The distance function between two points a = (x1, y1) and b = (x2, y2) is defined as- P(a, b) = |x2 – x1| + |y2 – y1|

euclidean distance = sqrt [$(x2 – x1)^2 + (y2 – y1)^2$]

Use K-Means Algorithm to find the three cluster.

Numerical

Iteration-01:

Calculate distance of each point from each of center of three clusters.

- The distance is calculated by using the given distance function.

Calculating Distance Between A1(2, 10) and C1(2, 10)-

P(A1, C1) = |x2 – x1| + |y2 – y1| = |2 – 2| + |10 – 10| = 0

Calculating Distance Between A1(2, 10) and C2(5, 8)-

P(A1, C2) = |x2 – x1| + |y2 – y1| = |5 – 2| + |8 – 10| = 3 + 2 = 5

Calculating Distance Between A1(2, 10) and C3(1, 2)-

P(A1, C3) = |x2 – x1| + |y2 – y1| = |1 – 2| + |2 – 10| = 1 + 8= 9

Numerical

Given Points	Distance from center (2, 10) of Cluster-01	Distance from center (5, 8) of Cluster-02	Distance from center (1, 2) of Cluster-03	Point belongs to Cluster
A1(2, 10)	0	5	9	C1
A2(2, 5)	5	6	4	C3
A3(8, 4)	12	7	9	C2
A4(5, 8)	5	0	10	C2
A5(7, 5)	10	5	9	C2
A6(6, 4)	10	5	7	C2
A7(1, 2)	9	10	0	C3
A8(4, 9)	3	2	10	C2

Numerical

Cluster-01: A1(2, 10)

Cluster-02:

- A3(8, 4)
- A4(5, 8)
- A5(7, 5)
- A6(6, 4)
- A8(4, 9)

Cluster-03:

- A2(2, 5)
- A7(1, 2)

For Cluster-01: only one point A1(2, 10) in Cluster-01. So, cluster center remains the same.

For Cluster-02:

Center of Cluster-02

= ((8 + 5 + 7 + 6 + 4)/5, (4 + 8 + 5 + 4 + 9)/5)

= (6, 6)

For Cluster-03:

Center of Cluster-03

= ((2 + 1)/2, (5 + 2)/2) = (1.5, 3.5)

This is completion of Iteration-01.

Now, re-compute the new cluster clusters.

Numerical

Calculating Distance Between A1(2, 10) and C1(2, 10)-

P(A1, C1) = |x2 – x1| + |y2 – y1| = |2 – 2| + |10 – 10| = 0

Calculating Distance Between A1(2, 10) and C2(6, 6)-

P(A1, C2) = |x2 – x1| + |y2 – y1| = |6 – 2| + |6 – 10| = 4 + 4 = 8

Calculating Distance Between A1(2, 10) and C3(1.5, 3.5)-

P(A1, C3) = |x2 – x1| + |y2 – y1| = |1.5 – 2| + |3.5 – 10| = 0.5 + 6.5 = 7

Numerical

Given Points	Distance from center (2, 10) of Cluster-01	Distance from center (6, 6) of Cluster-02	Distance from center (1.5, 3.5) of Cluster-03	Point belongs to Cluster
A1(2, 10)	0	8	7	C1
A2(2, 5)	5	5	2	C3
A3(8, 4)	12	4	7	C2
A4(5, 8)	5	3	8	C2
A5(7, 5)	10	2	7	C2
A6(6, 4)	10	2	5	C2
A7(1, 2)	9	9	2	C3
A8(4, 9)	3	5	8	C1

Numerical

Cluster-01:

- A1(2, 10)
- A8(4, 9)

Cluster-02:

- A3(8, 4)
- A4(5, 8)
- A5(7, 5)
- A6(6, 4)

Cluster-03:

- A2(2, 5)
- A7(1, 2)

Re-compute the new cluster clusters.

For Cluster-01:

Center of Cluster-01

= ((2 + 4)/2, (10 + 9)/2) = (3, 9.5)

For Cluster-02:

Center of Cluster-02

= ((8 + 5 + 7 + 6)/4, (4 + 8 + 5 + 4)/4)

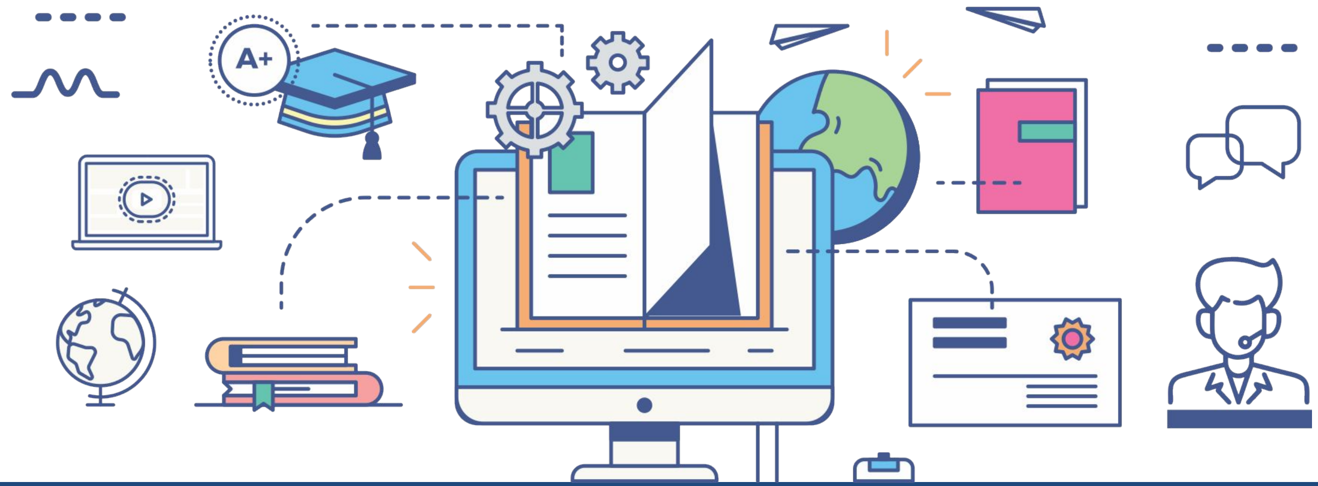
= (6.5, 5.25)

For Cluster-03:

Center of Cluster-03 = ((2 + 1)/2, (5 + 2)/2)

= (1.5, 3.5)

DIGITAL LEARNING CONTENT



Parul® University

www.paruluniversity.ac.in