Clustering

K-Means Clustering

Unsupervised Learning

- We do not have target variable
- We do not have train test data
- We do not have accuracy, as we do not have any GROUND TRUTH to compare with
- We will try to find out patterns or similarities in our data
- The process of finding or grouping similar data points is known as Clustering

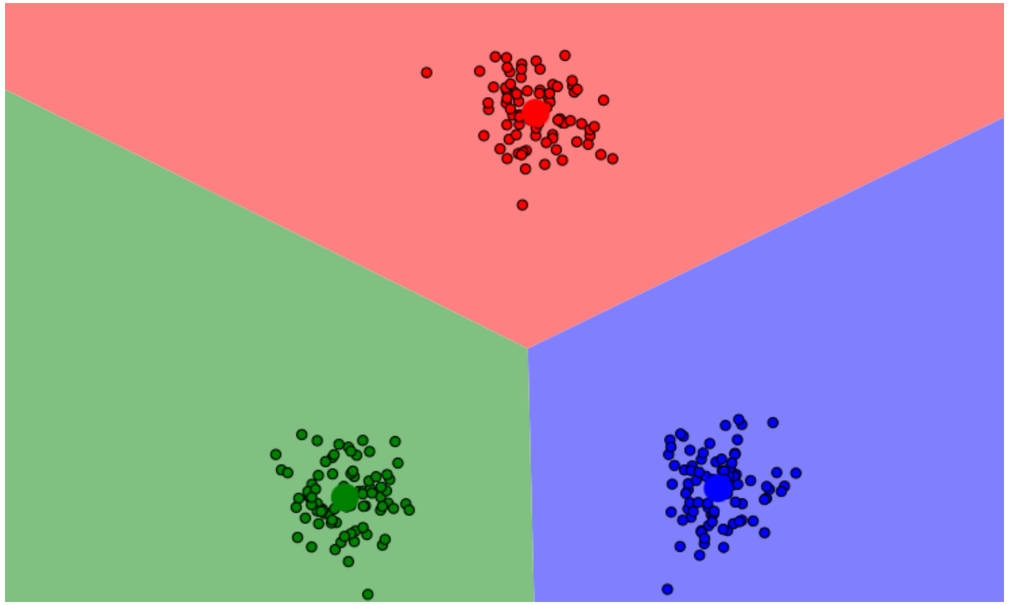
K Means Clustering

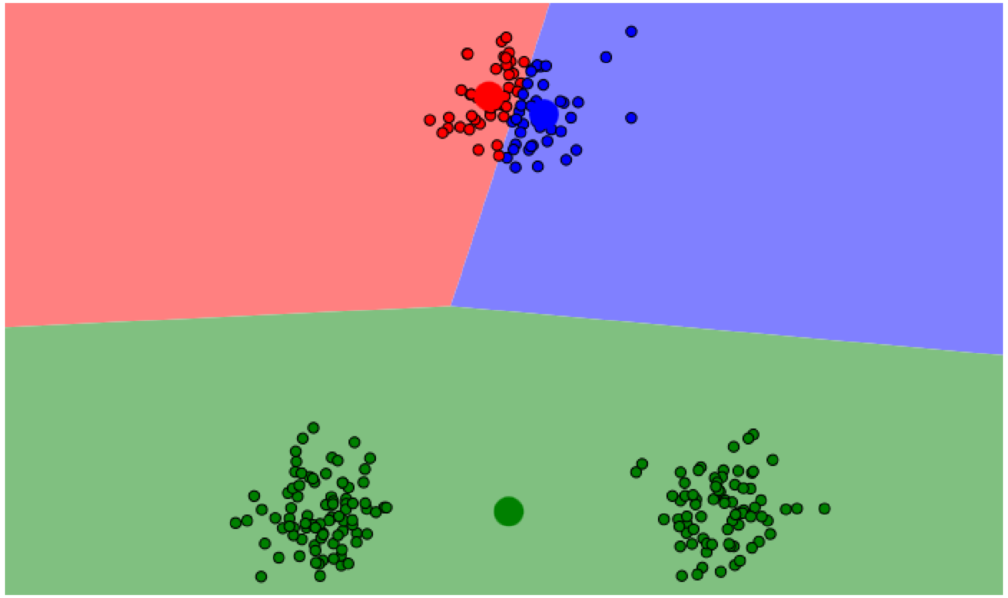
- **The process of organizing objects into groups whose members are similar in some way”**
- **A *cluster* is therefore a collection of objects which are coherent internally, but clearly dissimilar to the objects belonging to other clusters**
- Similarity = Euclidean Distance between the datapoints
- Euclidean distance can be calculated in more than 2 dimensions - the formula remains same.
- K-MEANS ALGORITHM
  - K = No. of clusters = You have to tell, the number of cluster you want to create
  - Step-1 – Algorithm generates K number of centres . To start with the centres are randomly located
  - Step-2 – Each data point in the data calculates its distance from the K -cluster centres. And assigns itself to the closest cluster center
  - Step -3 : Cluster Centers are supposed to be the center of the cluster. BUT REMEMBER in step-1 , we randomly located the centres. Now that we have datapoints in each of the clusters we can move the centre to the middle. MOVE THE CLUSTER CENTERS
  - Step-4: Because the centers have shifted, time to go back to Step-2.
  - Step-5 – Because reassignment of datapoints happened, we need to again move the center to the midpoint.
  - This step-4 and 5 carries on till the convergence happens.

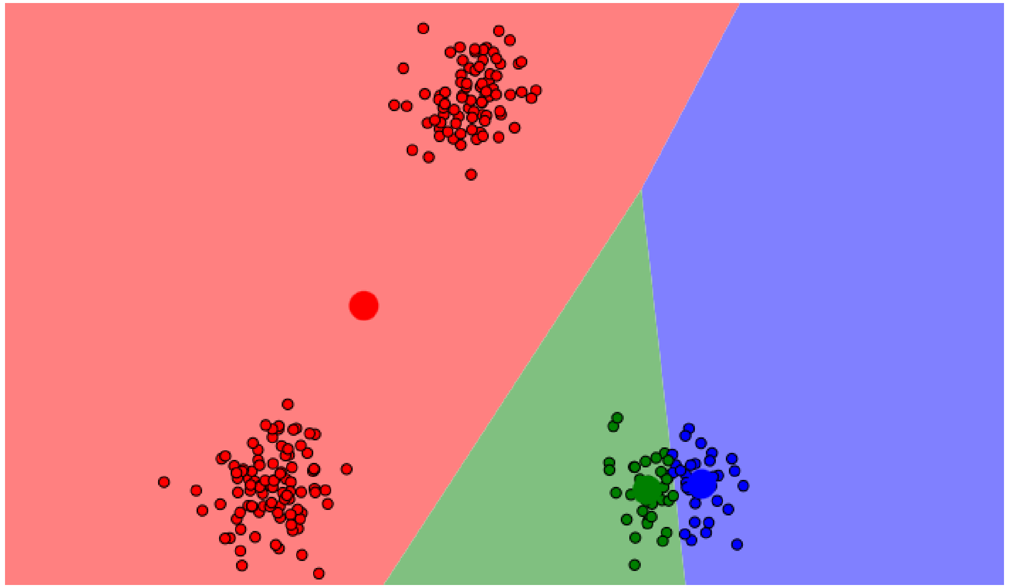
K Means Clustering

- At the end of the clustering exercise, in K-means – U do not care anymore about individual datapoints
- What u only care is about the centers. Because centers being the midpoint they are the flagbearer of the cluster
- So centers characteristics can be assumed the characteristics of the cluster
- ***Lower the "within cluster" distance, better the clustering is***
- ***Greater the "between cluster" distance better the clustering is***

K-Means Algorithm is NOT deterministic







- Based on the initial random centroid positions chosen, the algorithm converges
- Kmeans in NOT a deterministic algorithm as can be seen in the 3 runs of Kmeans for the same data. Based on the starting random centroids, the clusters identified are different
- Hence you need to run Kmeans multiple times with different random centroids. And calculate WCSS (inertia) for each result, and choose the result with least WCSS

In computer science, a deterministic algorithm is an algorithm that, given a particular input, will always produce the same output, with the underlying machine always passing through the same sequence of states

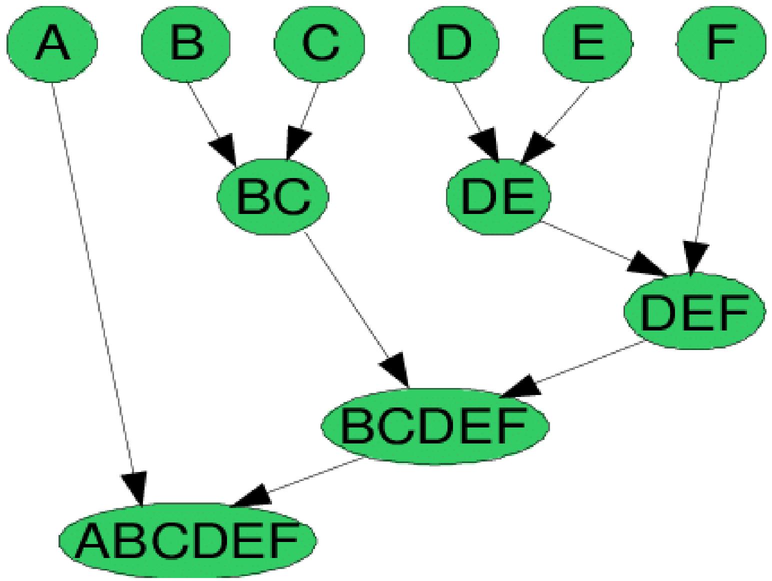
Hierarchical Clustering

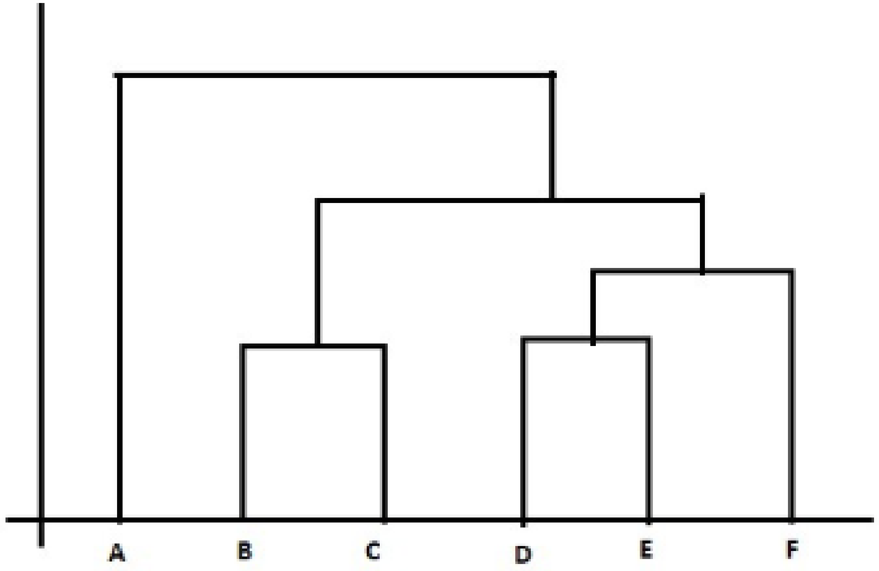
Hierarchical Clustering

- Hierarchical clustering technique is of two types:
    - Agglomerative
    - Divisive

Agglomerative Hierarchical Clustering

- In this technique, initially each data point is considered as an individual cluster. At each iteration, the similar clusters merge with other clusters until one cluster or K clusters are formed
- Steps
  - Compute the proximity matrix
  - Let each data point be a cluster
  - Repeat: Merge the two closest clusters and update the proximity matrix
  - Until only a single cluster remains
- Lets say we have six data points {A,B,C,D,E,F}
- The picture demonstrates the steps, starting with 6 clusters and finally merging them in a single cluster, hierarchically
- It can also be viewed as a Dendogram





Divisive Hierarchical clustering Technique

- We consider all the data points as a single cluster
- In each iteration, we separate the data points from the cluster which are not similar
- Each data point which is separated is considered as an individual cluster
- In the end, we'll be left with n clusters
- We can say that the Divisive Hierarchical clustering is exactly the opposite of the Agglomerative Hierarchical clustering

Calculating similarity(proximity) between clusters

- Calculating the similarity between two clusters is important to merge or divide the clusters
  - MIN
    - Pick the two closest points such that one point lies in cluster one and the other point lies in cluster 2 and take their similarity and declare it as the similarity between two clusters
  - MAX
    - Pick the two farthest points such that one point lies in cluster one and the other point lies in cluster 2 and take their similarity and declare it as the similarity between two clusters
  - Group Average
    - Take all the pairs of points and compute their similarities and calculate the average of the similarities
  - Distance Between Centroids
    - Compute the centroids of two clusters C1 & C2 and take the similarity between the two centroids as the similarity between two clusters
  - Ward's Method
    - Exactly the same as Group Average except that Ward's method calculates the sum of the square of the distances

DBScan Clustering

DBSCAN

- Clusters are dense regions in the data space, separated by regions of the lower density of points
- The DBSCAN algorithm is based on this intuitive notion of “clusters” and “noise”
- The key idea is that for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points.

DBSCAN

- The DBSCAN algorithm uses two parameters:
  - **minPts**: The minimum number of points (a threshold) clustered together for a region to be considered dense.
  - **eps (ε):** A distance measure that will be used to locate the points in the neighborhood of any point.
- Algorithmic steps for DBSCAN clustering
  - The algorithm proceeds by arbitrarily picking up a point in the dataset (until all points have been visited).
  - If there are at least ‘minPoint’ points within a radius of ‘ε’ to the point then we consider all these points to be part of the same cluster.
  - The clusters are then expanded by recursively repeating the neighborhood calculation for each neighboring point

DBSCAN – Reachability and Connectivity

- **Reachability** in terms of density establishes a point to be reachable from another if it lies within a particular distance (eps) from it.
- **Connectivity**, on the other hand, involves a transitivity based chaining-approach to determine whether points are located in a particular cluster. For example, p and q points could be connected if p->r->s->t->q, where a->b means b is in the neighborhood of a.

DBSCAN Algorithm

- Algorithmic steps for DBSCAN clustering
    - The algorithm proceeds by arbitrarily picking up a point in the dataset (until all points have been visited).
    - If there are at least ‘minPoint’ points within a radius of ‘ε’ to the point then we consider all these points to be part of the same cluster.
    - The clusters are then expanded by recursively repeating the neighborhood calculation for each neighboring point

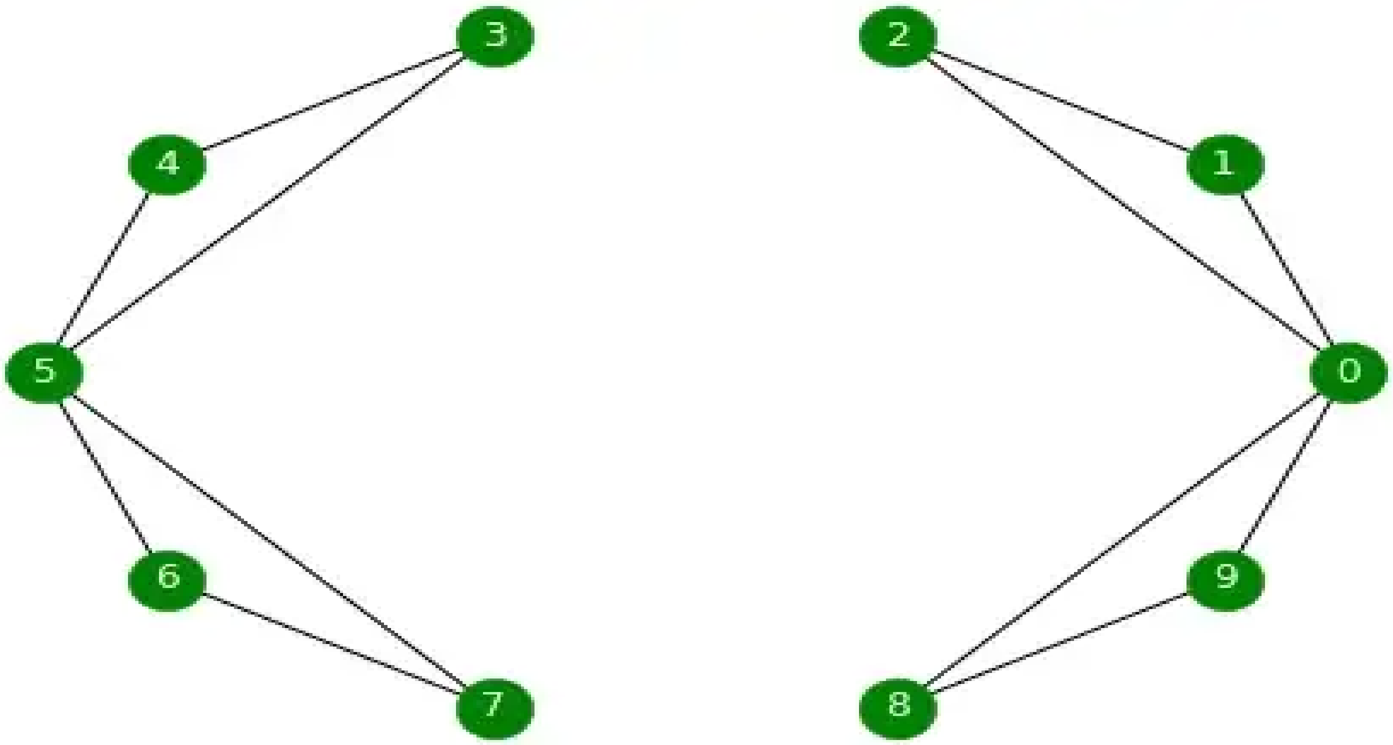
Spectral Clustering

Spectral Clustering

- Spectral clustering reduces complex multidimensional datasets into clusters of similar data
- The data points are treated as nodes of a graph and similar data points (immediately next to each other) are connected in a graph
- The nodes are then mapped to a low-dimensional space that can be easily segregated to form clusters.
- Spectral Clustering uses information from the eigenvalues of special matrices derived from the graph or the data set.

Spectral Clustering

A Clustering technique that treats each data point as a graph-node and performs graph-partitioning

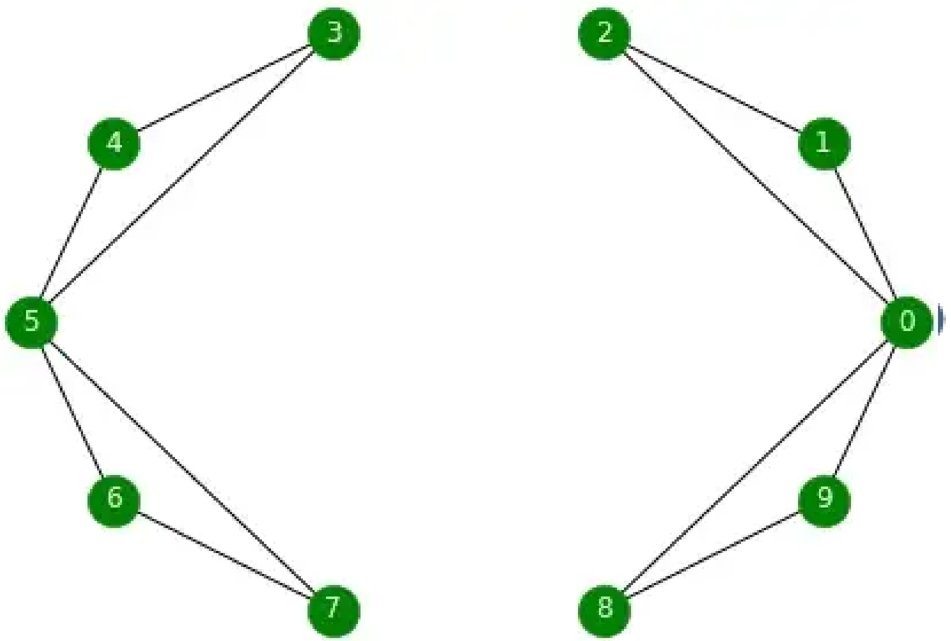


Graph with two disconnected components.

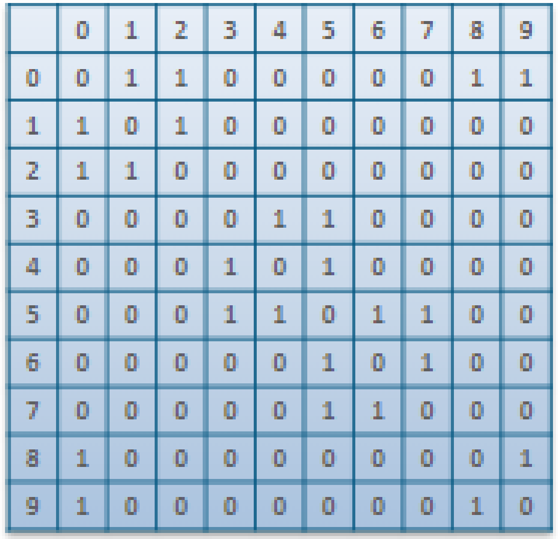
Spectral Clustering: Steps

Step 1-

- **Building Similarity Graph-** Builds adjacency matrix and degree matrix based on Epsilon-Neighborhood Graph, KNN or Fully Connected Graph

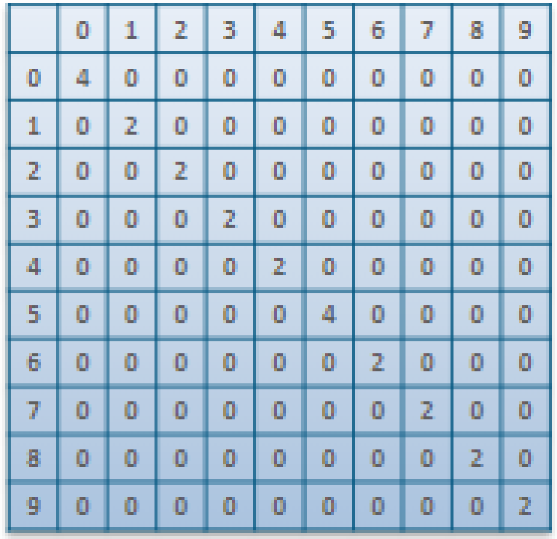


Adjacency Matrix(A)



	0	1	2	3	4	5	6	7	8	9
0	0	1	1	0	0	0	0	0	1	1
1	1	0	1	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0	0
3	0	0	0	0	1	1	0	0	0	0
4	0	0	0	1	0	1	0	0	0	0
5	0	0	0	1	1	0	1	1	0	0
6	0	0	0	0	0	1	0	1	0	0
7	0	0	0	0	0	1	1	0	0	0
8	1	0	0	0	0	0	0	0	0	1
9	1	0	0	0	0	0	0	0	1	0

Degree Matrix(D)



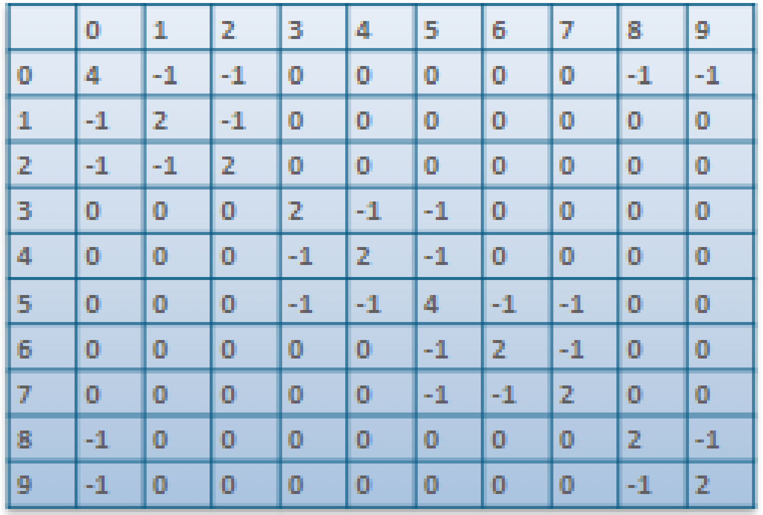
	0	1	2	3	4	5	6	7	8	9
0	4	0	0	0	0	0	0	0	0	0
1	0	2	0	0	0	0	0	0	0	0
2	0	0	2	0	0	0	0	0	0	0
3	0	0	0	2	0	0	0	0	0	0
4	0	0	0	0	2	0	0	0	0	0
5	0	0	0	0	0	4	0	0	0	0
6	0	0	0	0	0	0	2	0	0	0
7	0	0	0	0	0	0	0	2	0	0
8	0	0	0	0	0	0	0	0	2	0
9	0	0	0	0	0	0	0	0	0	2

Spectral Clustering: Steps

Step 2-

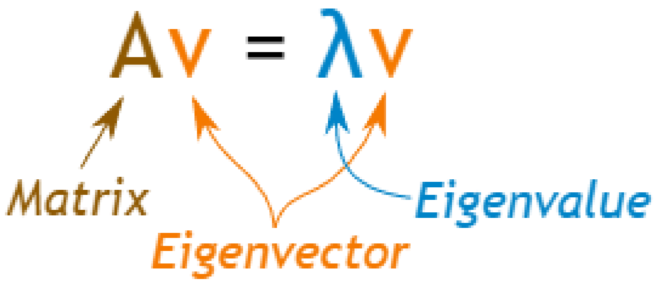
- **Projecting the data onto a lower Dimensional Space-** Compute Graph Laplacian matrix by L=D-A and Calculate First Eigen Vectors with nonzero values

Laplacian matrix



	0	1	2	3	4	5	6	7	8	9
0	4	-1	-1	0	0	0	0	0	-1	-1
1	-1	2	-1	0	0	0	0	0	0	0
2	-1	-1	2	0	0	0	0	0	0	0
3	0	0	0	2	-1	-1	0	0	0	0
4	0	0	0	-1	2	-1	0	0	0	0
5	0	0	0	-1	-1	4	-1	-1	0	0
6	0	0	0	0	0	-1	2	-1	0	0
7	0	0	0	0	0	-1	-1	2	0	0
8	-1	0	0	0	0	0	0	0	2	-1
9	-1	0	0	0	0	0	0	0	-1	2

For a square matrix A, an Eigenvector and Eigenvalue make this equation true:



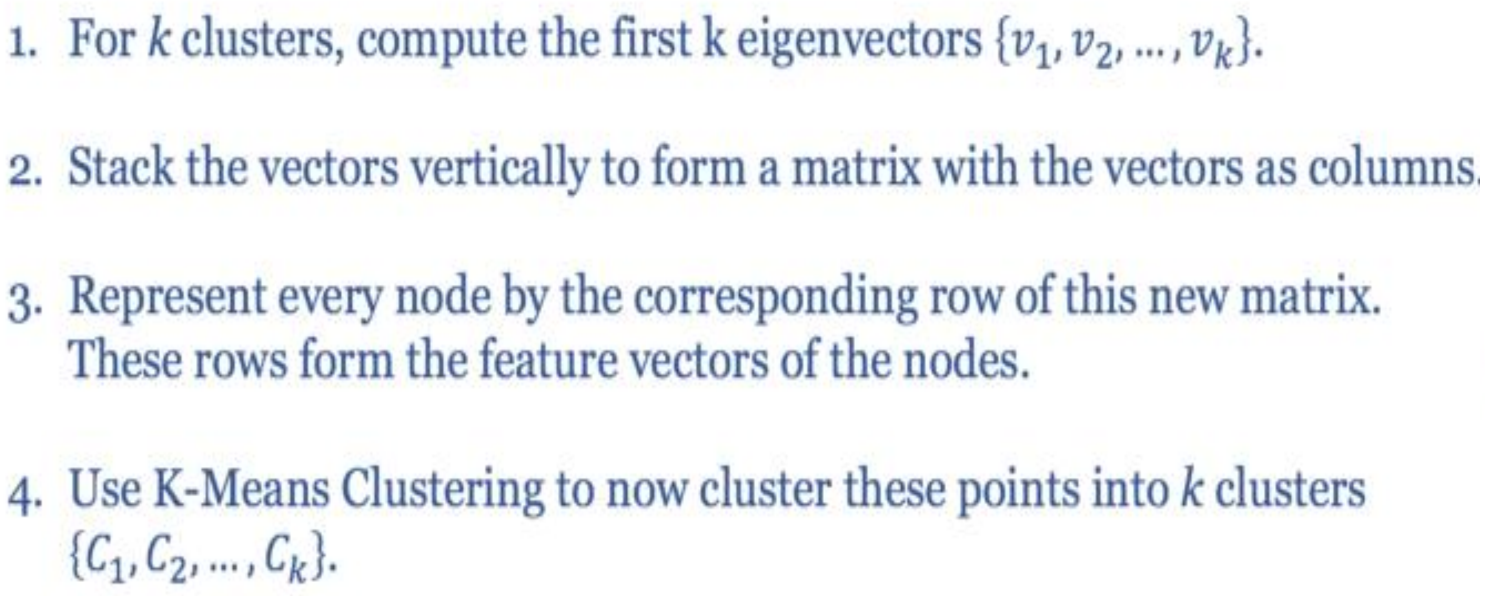
$$Av = \lambda v$$

https://www.mathsisfun.com/algebra/eigenvalue.html

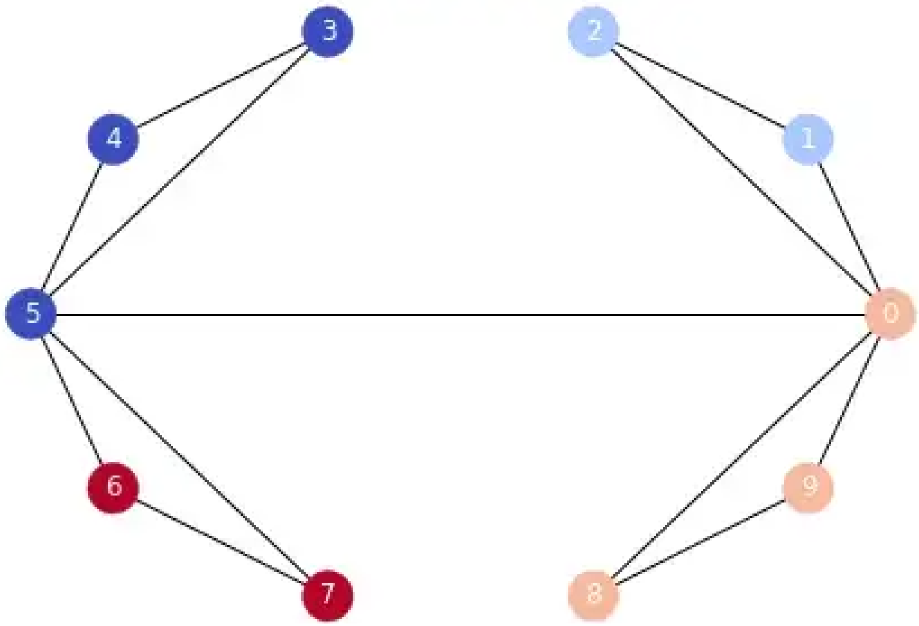
Spectral Clustering: Steps

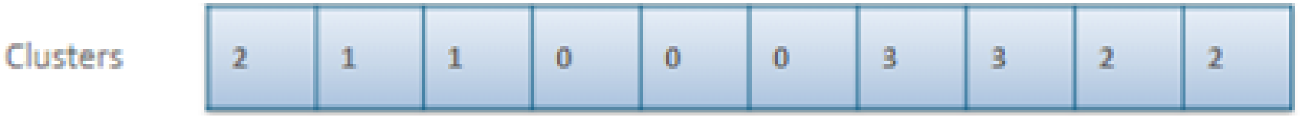
Step 3-

- Perform any clustering method(K-Means) on the eigen vectors



1. For k clusters, compute the first k eigenvectors $\{v_1, v_2, \dots, v_k\}$.
2. Stack the vectors vertically to form a matrix with the vectors as columns.
3. Represent every node by the corresponding row of this new matrix. These rows form the feature vectors of the nodes.
4. Use K-Means Clustering to now cluster these points into k clusters $\{C_1, C_2, \dots, C_k\}$.





Clusters	2	1	1	0	0	0	3	3	2	2

https://towardsdatascience.com/spectral-clustering-82d3cff3d3b7
https://towardsdatascience.com/spectral-clustering-aba2640c0d5b#:~:text=Spectral%20clustering%20is%20a%20technique,non%20graph%20data%20as%20well