K-Means Clustering Algorithm

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. In this topic, we will learn what is K-means clustering algorithm, how the algorithm works, along with the Python implementation of k-means clustering.

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

The working of the K-Means algorithm is explained in the below steps:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate new centroid of each cluster.

Step-5: Repeat the third steps, which means reassign each data point to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

Step-7: The model is ready.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.
The k-means clustering algorithm mainly performs two tasks:

•Determines the best value for K center points or centroids by an iterative process.
•Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

Hence each cluster has data points with some commonalities, and it is away from other clusters.

Assume, you want to cluster 7 observations into 3 clusters using K-Means clustering algorithm. After first iteration the clusters: C1, C2, C3 has the following observations:

C1: {(1,1), (4,4), (7,7)}

C2: {(0,4), (4,0)}

C3: {(5,5), (9,9)}

What will be the cluster centroids if you want to proceed for second iteration?

a. C1: (4,4), C2: (2,2), C3: (7,7)
b. C1: (2,2), C2: (0,0), C3: (5,5)
c. C1: (6,6), C2: (4,4), C3: (9,9)
d. None of these

Following Question what will be the Manhattan distance for observation (9, 9) from cluster centroid C1 in the second iteration?

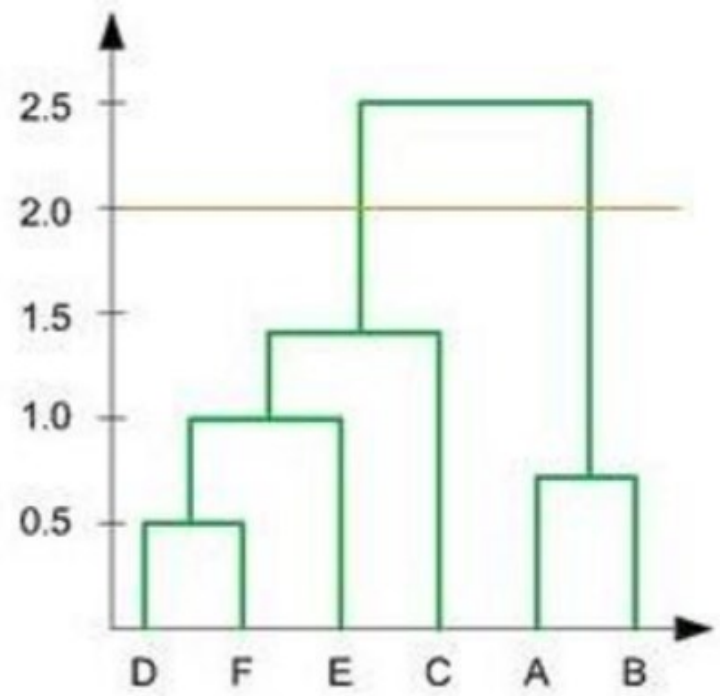
a. 10
b. 5
c. 6
d. 7

Suppose we have four different objects O_1, O_2, O_3, and O_4 with two features f_1 and f_2 as shown in the table below. We would like to cluster these four objects into two classes C_1 and C_2. We would like to use K-means clustering approach to achieve this. To start with consider O_1 and O_2 as initial cluster centroids of class C_1 and C_2 respectively. After one iteration, if the centroids of class C_1 is (a, b) and class C_2 is (c, d), then $a + b + c + d$ is

Object	f_1	f_2
O_1	1	1
O_2	2	1
O_3	4	3
O_4	5	4

- 2
- 6.4
- 7.0
- 8.3

In the figure below, if you draw a horizontal line on y-axis for y=2. What will be the number of clusters formed?



a. 1
b. 2
c. 3
d. 4

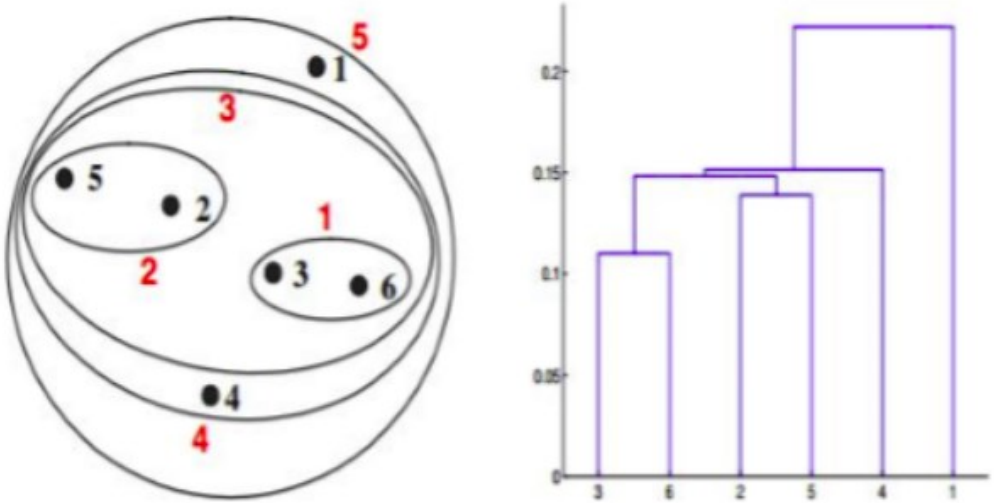
point	x coordinate	y coordinate
p1	0.4005	0.5306
p2	0.2148	0.3854
p3	0.3457	0.3156
p4	0.2652	0.1875
p5	0.0789	0.4139
p6	0.4548	0.3022

Table : X-Y coordinates of six points.

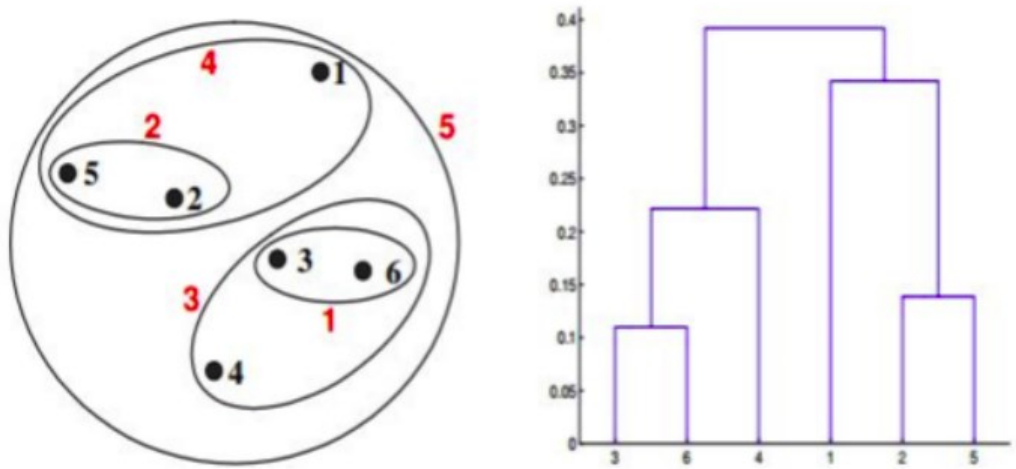
	p1	p2	p3	p4	p5	p6
p1	0.0000	0.2357	0.2218	0.3688	0.3421	0.2347
p2	0.2357	0.0000	0.1483	0.2042	0.1388	0.2540
p3	0.2218	0.1483	0.0000	0.1513	0.2843	0.1100
p4	0.3688	0.2042	0.1513	0.0000	0.2932	0.2216
p5	0.3421	0.1388	0.2843	0.2932	0.0000	0.3921
p6	0.2347	0.2540	0.1100	0.2216	0.3921	0.0000

Table : Distance Matrix for Six Points

A.



B.



Which of the following clustering representations and dendrogram depicts the use of MIN or Single link proximity function in hierarchical clustering:

) Selects some facts about hierarchical clustering –

- A hierarchical method comes under either agglomerative or divisive algorithms
- Hierarchical methods suffer from the fact that once a step (merge or split) is done, it can never be undone
- A tree structure called a dendrogram is commonly used to represent the process of hierarchical clustering
- All of the above

Which one of the following is correct?

a. Complete linkage clustering is computationally cheaper compared to single linkage.
b. Single linkage clustering is computationally cheaper compared to K-means clustering.
c. K-Means clustering is computationally cheaper compared to single linkage clustering.
d. None of the above.

What is true about K-Mean Clustering?

1. K-means is extremely sensitive to cluster center initializations
2. Bad initialization can lead to Poor convergence speed
3. Bad initialization can lead to bad overall clustering

a. 1 and 2
b. 1 and 3
c. All of the above
d. 2 and 3

Comparison between PCA and LDA:

1.Objective:

1. PCA: Aims to maximize variance and is an unsupervised technique.
2. LDA: Aims to maximize the separation between classes and is a supervised technique.

2.Supervision:

1. PCA: Unsupervised, meaning it does not consider class labels.
2. LDA: Supervised, as it takes class labels into account during the analysis.

3.Use Case:

1. PCA: Often used for dimensionality reduction, noise reduction, and visualization.
2. LDA: Primarily used for classification and feature extraction when there are distinct classes in the data.

Both PCA and LDA are techniques for linear dimensionality reduction, LDA is specifically designed for classification tasks and considers class information during the transformation, making it more suitable for scenarios where the goal is to discriminate between different classes.

Linear Discriminant Analysis (LDA):

•**Objective:** LDA is a supervised dimensionality reduction technique used for classification and feature extraction. The primary goal of LDA is to find the linear combinations of features that best separate two or more classes in the data.

•**Method:** It maximizes the distance between the means of different classes while minimizing the spread (variance) within each class. The result is a set of linear combinations, known as discriminant functions, which can be used for classification.

•**Use Cases:** LDA is commonly used in pattern recognition, machine learning, and statistics for tasks such as face recognition, handwriting recognition, and medical diagnosis.

Principal Component Analysis (PCA):

1.Standardization:

If the features have different scales, it's common to standardize them (subtract mean and divide by standard deviation) to ensure that each feature contributes equally to the analysis.

2.Compute Covariance Matrix:

Calculate the covariance matrix of the standardized data. The covariance matrix provides information about the relationships between different features.

3.Calculate Eigenvectors and Eigenvalues:

Compute the eigenvectors and eigenvalues of the covariance matrix. Eigenvectors represent the principal components, and eigenvalues indicate the amount of variance explained by each principal component.

4.Sort Eigenvalues:

Arrange the eigenvectors in descending order based on their corresponding eigenvalues. This helps identify the principal components in order of significance.

5.Select Principal Components:

Choose the top k eigenvectors to form the principal components, where k is the desired dimensionality of the reduced data.

6.Projection:

Project the original data onto the selected principal components to obtain the reduced-dimensional representation.

Linear Discriminant Analysis (LDA):

1. **Compute Class Means:**
   Calculate the mean vector for each class in the dataset.
2. **Compute Within-Class Scatter Matrix (SW) and Between-Class Scatter Matrix (SB):**
   Compute the within-class scatter matrix (SW) and the between-class scatter matrix (SB). SW represents the spread of data within each class, and SB represents the spread between class means.
3. **Compute Eigenvalues and Eigenvectors:**
   Solve the generalized eigenvalue problem to find the eigenvalues and corresponding eigenvectors of the matrix (SW^(-1) * SB).
4. **Sort Eigenvalues:**
   Sort the eigenvalues in descending order and select the top k eigenvectors. These eigenvectors become the discriminant functions.
5. **Projection:**
   Project the original data onto the selected discriminant functions to obtain the reduced-dimensional representation. The number of discriminant functions used corresponds to the number of classes minus one (C-1), where C is the number of classes.