BTCOL606: Machine Learning Lab **(Date: 05/05/2023)**

Experiment NO : 07

Instructor: Mr.Katkar Atish R. *Name:* Suyog Rawas, Roll No: CS3257

AIM

Implement and demonstrate the K-means clustering algorithm.

INTRODUCTION TO K-MEANS CLUSTERING ALGORITHM

The K-means clustering algorithm is a popular unsupervised machine learning technique used for grouping data into clusters based on their similarity. It is an iterative algorithm that aims to partition a dataset into K distinct clusters, where K is a predefined number.

The algorithm works by assigning data points to the nearest centroid, which represents the center of each cluster. Initially, K centroids are randomly selected from the dataset. Then, the algorithm iteratively updates the centroids and reassigns data points to the nearest centroid until convergence is achieved.

The steps involved in the K-means clustering algorithm :

1. Initialization: Randomly select K data points as initial centroids.
2. Assignment: Assign each data point to the nearest centroid based on a distance metric, commonly the Euclidean distance.
3. Update: Recalculate the centroids by taking the mean of all data points assigned to each centroid.
4. Iteration: Repeat the assignment and update steps until convergence or a maximum number of iterations is reached.
5. Convergence: Stop the algorithm when the centroids no longer change significantly or when the maximum number of iterations is reached.

The algorithm aims to minimize the within-cluster sum of squares, also known as the inertia or distortion. It seeks to create compact and well-separated clusters by minimizing the total distance between data points and their respective centroids.

One challenge in using K-means clustering is determining the optimal value of K, the number of clusters. Various methods, such as the elbow method or silhouette analysis, can be employed to find an appropriate K value.

K-means clustering is widely used in various applications, including customer segmentation, image segmentation, anomaly detection, and recommender systems. Its simplicity and efficiency make it a popular choice for exploratory data analysis and clustering tasks.

DATASET DESCRIPTION :

In this code, the data list represents your dataset, where each element is a data point with its corresponding features. The K variable determines the the number of clusters you want to create.

The max-iterations sets the maximum number of iterations for convergence.The code initializes centroids randomly, assigns data points to the closest centroids, updates the centroids based on the mean of the assigned points, and repeats this process until convergence or reaching the maximum number of

IMPLEMENTATION CODE FOR K-MEANS CLUSTERING ALGORITHM

```
import random
import math
# Euclidean distance between two points
def euclidean_distance ( point1 , point2 ) :
    squared_distance = 0
    for i in range (len ( point1 ) ) :
        squared_distance += ( point1 [ i ] - point2 [ i ]) ** 2
    return math . sqrt ( squared_distance )

# Initialize centroids randomly
def initialize_centroids ( data , k ) :
    centroids = random . sample ( data , k )
    return centroids

# Assign each data point to the closest centroid
def assign_clusters ( data , centroids ) :
    clusters = [[] for _ in centroids ]
    for point in data :
        distances = [ euclidean_distance ( point , centroid ) for centroid in
            centroids ]
        closest_centroid_index = distances . index (min( distances ) )
        clusters [ closest_centroid_index ]. append ( point )
    return clusters

# Update centroids based on the mean of the assigned points
def update_centroids ( clusters ) :
    centroids = []
    for cluster in clusters :
        centroid = [ sum( dim ) / len( cluster ) for dim in zip (* cluster ) ]
        centroids . append ( centroid )
    return centroids

# K- means clustering algorithm
def k_means ( data , k , max_iterations ) :
    centroids = initialize_centroids ( data , k )
    for _ in range ( max_iterations ) :
        clusters = assign_clusters ( data , centroids )
        new_centroids = update_centroids ( clusters )

        # Check for convergence
        if new_centroids == centroids :
        break

        centroids = new_centroids

        return clusters , centroids

        # Example usage
data = [
    [2 , 10] ,
    [2 , 5] ,
    [8 , 4] ,
    [5 , 8] ,
    [7 , 5] ,
    [6 , 4] ,
    [1 , 2] ,
    [4 , 9]
]
k = 2
max_iterations = 100
```

```
62
63 clusters , centroids = k_means ( data , k , max_iterations )
64
65 # Print the results
66 print (" Clusters :")
67 for i , cluster in enumerate ( clusters ) :
68      print ( f" Cluster {i +1}: { cluster }")
69 print (" Centroids :")
70 for i , centroid in enumerate ( centroids ):
71     print ( f" Centroid {i +1}: { centroid }")
```

OUTPUT

1 Output Of K-means Clustering Algorithm

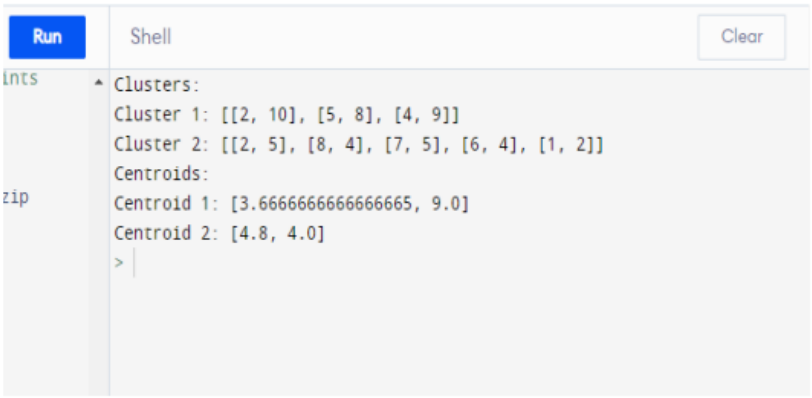

Figure 1: K-means Clustering Output

2 Work done by the Algorithm

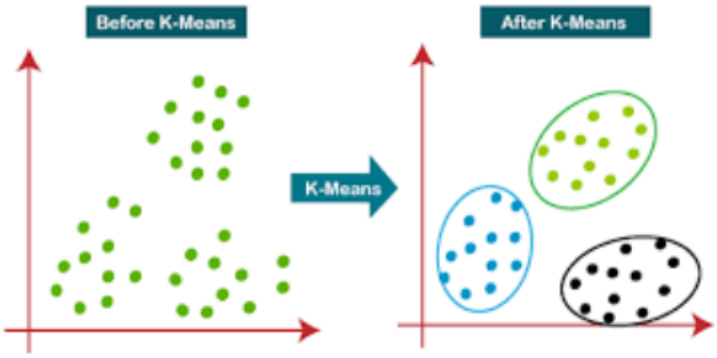

Figure 2: Work done by the Algorithm

CONCLUSION

In conclusion, the K-means clustering algorithm is a popular unsupervised machine learning technique used for grouping similar data points into clusters. It is a simple and efficient algorithm that can handle large datasets and is widely used in various applications such as customer segmentation, image compression, and anomaly detection