**SWE2009**

**DATA MINING DA-2**

**A.M.HARSHITHA**

**22MIS1051**

The dataset contains information about credit card customers and various attributes associated with them. Here's a brief summary of the columns:

1. ****Sl\_No****: Serial number or identifier for each customer.
2. ****Customer Key****: Unique identifier for each customer.
3. ****Avg\_Credit\_Limit****: Average credit limit of the customer.
4. ****Total\_Credit\_Cards****: Total number of credit cards held by the customer.
5. ****Total\_visits\_bank****: Total number of visits to the bank by the customer.
6. ****Total\_visits\_online****: Total number of visits made by the customer online.
7. ****Total\_calls\_made****: Total number of calls made by the customer to the bank's call center.

Some important points about the dataset:

* It includes both numerical and categorical data.
* The dataset seems to represent a sample of credit card customers.
* It can be used for various analyses such as customer segmentation, customer behavior analysis, and predictive modeling.

Before proceeding with any analysis or modeling, it's essential to explore the data further, check for missing values, outliers, and understand the distribution of each variable. This exploration will help in selecting appropriate methods for analysis and modeling.

窗体顶端

窗体底端

1.k mean:

K-means clustering is a popular unsupervised machine learning algorithm used for partitioning a dataset into a predetermined number of clusters. The algorithm iteratively assigns each data point to the nearest cluster center and then recalculates the cluster centers based on the mean of the data points assigned to each cluster. This process continues until the cluster assignments stabilize or until a specified number of iterations is reached. K-means clustering aims to minimize the within-cluster sum of squares, effectively grouping similar data points together while maximizing the dissimilarity between clusters. It is widely used for tasks such as customer segmentation, image compression, and anomaly detection. However, it is sensitive to the initial selection of cluster centers and may converge to local optima, requiring multiple runs with different initializations to find the optimal solution. Additionally, it assumes clusters are spherical and have similar sizes, which may not always be the case in real-world data.

2.kmedoid:

K-medoids clustering is a partitioning method used to divide a dataset into K clusters, where each cluster is represented by a single data point called a medoid. Unlike K-means clustering, which uses centroids to represent cluster centers, k-medoids selects actual data points as cluster representatives. The algorithm iteratively assigns each data point to the nearest medoid and updates the medoids to minimize a predefined dissimilarity measure, often using techniques like the PAM (Partitioning Around Medoids) algorithm. K-medoids clustering is advantageous in scenarios where using actual data points as cluster representatives is preferred over centroids, and when the dataset contains outliers or noise that can significantly affect centroid-based methods. Additionally, k-medoids can handle non-Euclidean distances and is robust to outliers, making it suitable for various clustering tasks.

3.clara:

CLARA (Clustering Large Applications) is a clustering algorithm designed for handling large datasets efficiently. It belongs to the category of partitioning clustering algorithms. Unlike some other methods like K-means, CLARA does not require the entire dataset to reside in memory at once, making it suitable for situations where memory resources are limited.

CLARA works by randomly sampling a subset of the data, performing clustering on this subset using another clustering algorithm (typically PAM - Partitioning Around Medoids), and then using the resulting clusters as representatives to partition the entire dataset. This process is repeated multiple times to ensure robustness and to account for the variability introduced by the random sampling.

CLARA is particularly useful for datasets that are too large to be handled by traditional clustering algorithms, as it allows for clustering to be performed on a smaller subset of the data while still providing reliable results. However, the trade-off is that CLARA may be computationally intensive, especially when repeated sampling and clustering are performed multiple times.

4.rock:

Rock clustering, also known as ROCK (RObust Clustering using linKs) clustering, is a robust hierarchical clustering algorithm designed to handle noisy datasets and detect clusters of varying shapes and sizes. Unlike traditional hierarchical clustering methods that rely solely on pairwise distances between data points, ROCK clustering considers the connectivity of data points through links or edges. The algorithm iteratively merges clusters based on the strength of links between them, allowing it to identify clusters even in the presence of noise and outliers. ROCK clustering is particularly suitable for datasets with complex structures and overlapping clusters, making it a valuable tool in exploratory data analysis and pattern recognition tasks. Additionally, ROCK clustering offers the advantage of being computationally efficient and scalable to large datasets, making it applicable in various domains such as bioinformatics, image analysis, and social network analysis.

5.agnes:

Agglomerative Nesting (AGNES) is a hierarchical clustering algorithm used to group similar objects into clusters. In AGNES, each data point initially belongs to its own cluster, and at each iteration, the algorithm merges the two most similar clusters based on a chosen distance metric until all data points belong to a single cluster. The resulting clustering structure can be visualized using a dendrogram, which illustrates the hierarchy of clusters and the order in which they were merged. AGNES is flexible and can accommodate various distance metrics and linkage methods to determine cluster similarity, making it widely applicable in data analysis and exploratory research. However, its computational complexity can be high for large datasets, and it may not always yield optimal results, especially when dealing with noisy or high-dimensional data.

6. PAM:

PAM (Partitioning Around Medoids) clustering is a partitioning algorithm that aims to partition a dataset into a specified number of clusters. It is similar to the more popular K-means clustering algorithm but uses medoids instead of centroids. Medoids are data points that represent the center of a cluster, minimizing the dissimilarity between themselves and other data points within the same cluster. PAM works iteratively to optimize the assignment of data points to clusters by swapping medoids and recalculating the total dissimilarity within clusters. Unlike K-means, PAM does not require the calculation of means, making it more robust to outliers and suitable for datasets with non-Euclidean distances. PAM is particularly useful for small to medium-sized datasets due to its computational efficiency compared to hierarchical clustering methods.

7. hierarchical:

Hierarchical clustering is a method of cluster analysis that builds a hierarchy of clusters. It starts with each data point as a single cluster and then iteratively merges or splits clusters based on their similarity until all data points belong to a single cluster. There are two main types of hierarchical clustering: agglomerative and divisive.

1. Agglomerative hierarchical clustering: It begins with each data point as a separate cluster and then iteratively merges the closest pairs of clusters until only one cluster remains. The merging process continues until a predefined criterion is met, such as a specified number of clusters or a certain level of similarity.

2. Divisive hierarchical clustering: It starts with all data points in a single cluster and then splits the cluster recursively into smaller clusters until each data point is in its own cluster. The splitting process continues until a stopping criterion is satisfied, such as a minimum cluster size or a maximum level of dissimilarity.

Hierarchical clustering produces a dendrogram, which is a tree-like diagram that illustrates the arrangement of the clusters. The vertical axis of the dendrogram represents the distance or dissimilarity between clusters, while the horizontal axis represents the individual data points or clusters being merged or split. Hierarchical clustering is useful for exploring the structure of the data and identifying nested clusters at different levels of granularity. It does not require specifying the number of clusters beforehand, making it suitable for exploratory analysis.

8.chamelion:

Chameleon clustering is a density-based clustering algorithm designed to handle datasets with varying densities and irregular shapes. It works by iteratively merging clusters based on their similarity and density. Unlike traditional density-based clustering algorithms like DBSCAN, Chameleon dynamically adjusts its parameters based on the local density of the data, making it more robust to varying density regions. It uses a hierarchical approach to build a dendrogram, which is then pruned to obtain the final clustering result. Chameleon clustering has been shown to perform well on datasets with complex structures and varying densities, making it suitable for applications such as image segmentation, spatial data analysis, and anomaly detection.

9.DBSCANS:

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a popular clustering algorithm used in data mining and machine learning. Unlike traditional clustering algorithms like k-means, DBSCAN doesn't require the user to specify the number of clusters beforehand. Instead, it defines clusters based on the density of data points in the feature space.

In DBSCAN, a core concept is the notion of "density-reachable" points. A point is considered core if within a specified radius, there are at least a minimum number of points. Points that are not core but fall within the neighborhood of a core point are considered part of the same cluster. This allows DBSCAN to identify clusters of arbitrary shapes and handle outliers effectively.

DBSCAN has two important parameters:

- Epsilon (eps): The maximum distance between two points to be considered neighbors.

- MinPts: The minimum number of points within eps distance for a point to be considered a core point.

DBSCAN is particularly useful for datasets with complex structures, where clusters may have varying densities or irregular shapes. It's robust to noise and capable of handling outliers effectively, making it suitable for a wide range of applications such as spatial data analysis, anomaly detection, and customer segmentation. However, choosing appropriate parameters can be challenging, and the algorithm may struggle with datasets of varying densities or high-dimensional data.

10.Dianna:

DIANA (Divisive Analysis Clustering) is a hierarchical clustering algorithm that works by recursively partitioning a dataset into subsets or clusters. Unlike agglomerative clustering methods, which start with individual data points and merge them into clusters, DIANA begins with the entire dataset and divides it into smaller subsets. This divisive approach continues recursively until each subset contains only one data point, resulting in a hierarchical tree structure known as a dendrogram. DIANA clustering is useful for exploring hierarchical relationships within datasets identifying natural and groupings or clusters based on the dissimilarity between data points. It is particularly suitable for datasets with complex structures or when the number of clusters is unknown or variable.