PUNE INSTITUTE OF COMPUTER TECHNOLOGY

Subject: Machine Learning (LP-1 LAB)

Name: Aditya Kangune **Roll No. : 33323**

Batch: K11 **Academic Year: 2021-22**

Assignment 4
K Means Clustering

Problem Statement:

Perform the following operations on the given dataset:

A. Apply Data pre-processing (Label Encoding, Data Transformation….) techniques if necessary.
B. Perform data-preparation (Train-Test Split)
C. Apply Machine Learning Algorithm
D. Evaluate Model.
E. Apply Cross-Validation and Evaluate Model

Objective:

This assignment will help the students to realize how to do Clustering using the K Means Clustering algorithm and Hierarchical Clustering Algorithm.

Theory:

KMeans Clustering:

1. K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups).
2. The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K.
3. The algorithm works iteratively to assign each data point to one of the K groups based on the features that are provided. Data points are clustered based on feature similarity.

The **results** of the K-means clustering algorithm are:

1. The centroids of the K clusters, which can be used to label new data.

2. Labels for the training data (each data point is assigned to a single cluster) rather than defining groups before looking at the data, clustering allows you to find and analyze the groups that have formed organically.

The "**Choosing K**" section below describes how the number of groups can be determined. Each centroid of a cluster is a collection of feature values that define the resulting groups.

Examining the centroid feature weights can be used to qualitatively interpret what kind of group each cluster represents.

This introduction to the K-means clustering algorithm covers:

- Common business cases where K-means is used
- The steps involved in running the

  algorithm.

Some examples of use cases are:

- **Behavioral segmentation:**
  - o Segment by purchase history
  - o Segment by activities on application, website, or platform.
  - o Define personas based on interests
  - o Create profiles based on activity monitoring
- **Inventory categorization:**
  - o Group inventory by sales activity
  - o Group inventory by manufacturing metrics
- **Sorting sensor measurements:**
  - o Detect activity types in motion sensors o Group images
  - o Separate audio
  - o Identify groups in health monitoring
- **Detecting bots or anomalies:**
  - o Separate valid activity groups from bots
  - o Group valid activity to clean up outlier detection In addition, monitoring if a tracked data point switches between groups over time can be used to detect meaningful changes in the data.

Algorithm:

The K-means clustering algorithm uses iterative refinement to produce a final result. The algorithm inputs are the number of clusters K and the data set. The data set is a collection of features for each data point.

The algorithms start with initial estimates for the K centroids, which can either be randomly generated or randomly selected from the data set.

The algorithm then iterates between two steps:

1. Data assignment step:

$$\underset{c_i \in C}{\operatorname{argmin}}\ dist(c_i, x)^2$$

1. Each centroid defines one of the clusters.
2. In this step, each data point is assigned to its nearest centroid, based on the squared Euclidean distance.
3. More formally, if ci is the collection of centroids in set C, then each data point x is assigned to a cluster based on where dist(·) is the standard (L2) Euclidean distance
4. . Let the set of data point assignments for each i th cluster centroid be Si.
5.

2. Centroid update step:

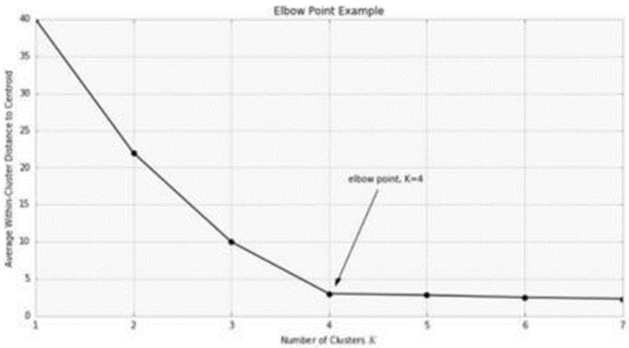
In this step, the centroids are recomputed. This is done by taking the mean of all data points assigned to that centroid's cluster.

$$c_i = \frac{1}{|S_i|}\sum_{x_i \in S_i} x_i$$

1. The algorithm iterates between steps one and two until a stopping criteria is met (i.e., no data points change clusters, the sum of the distances is minimized, or some maximum number of iterations is reached).
2. This algorithm is guaranteed to converge to a result.
3. The result may be a local optimum (i.e. not necessarily the best possible outcome), meaning that assessing more than one run of the algorithm with randomized starting centroids may give a better outcome.

Choosing K:

1. The algorithm described above finds the clusters and data set labels for a particular pre-chosen K.
2. To find the number of clusters in the data, the user needs to run the K-means clustering algorithm for a range of K values and compare the results.
3. In general, there is no method for determining the exact value of K, but an accurate estimate can be obtained using the following techniques.
4. One of the metrics that is commonly used to compare results across different values of K is the mean distance between data points and their cluster centroid.
5. Since increasing the number of clusters will always reduce the distance to data points, increasing K will always decrease this metric, to the extreme of reaching zero when K is the same as the number of data points.
6. Thus, this metric cannot be used as the sole target. Instead, mean distance to the centroid as a function of K is plotted and the "elbow point," where the rate of decrease sharply shifts, can be used to roughly determine K.
7. A number of other techniques exist for validating K, including cross-validation, information criteria, the information-theoretic jump method, the silhouette method, and the G-means algorithm.
8. In addition, monitoring the distribution of data points across groups provides insight into how the algorithm is splitting the data for each K.



Hierarchical Clustering:

Hierarchical clustering analysis is a method of cluster analysis that seeks to build a hierarchy of clusters i.e., tree-type structure based on the hierarchy.

Basically, there are two types of hierarchical cluster analysis strategies –

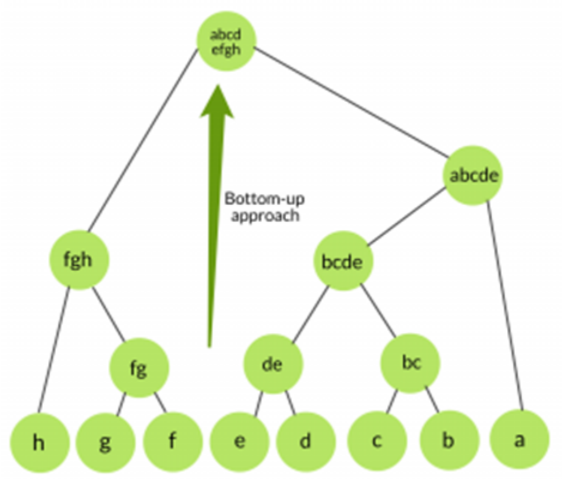
1. Agglomerative Clustering:

1. Also known as the bottom-up approach or hierarchical agglomerative clustering (HAC).
2. A structure that is more informative than the unstructured set of clusters returned by flat clustering.
3. This clustering algorithm does not require us to prespecify the number of clusters.
4. Bottom-up algorithms treat each data as a singleton cluster at the outset and then successively agglomerates pairs of clusters until all clusters have been merged into a single cluster that contains all data.

Algorithm :

```
given a dataset ($d_1$, $d_2$, $d_3$, ....$d_N$) of size N

# compute the distance matrix

for i=1 to N:

  # as the distance matrix is symmetric about

  # the primary diagonal so we compute only lower

  # part of the primary diagonal

  for j=1 to i:

    dis_mat[i][j] = distance[$d_i$, $d_j$]

each data point is a singleton cluster

repeat
```

```
    merge the two clusters having a minimum distance
    update the distance matrix
until only a single cluster remains
```



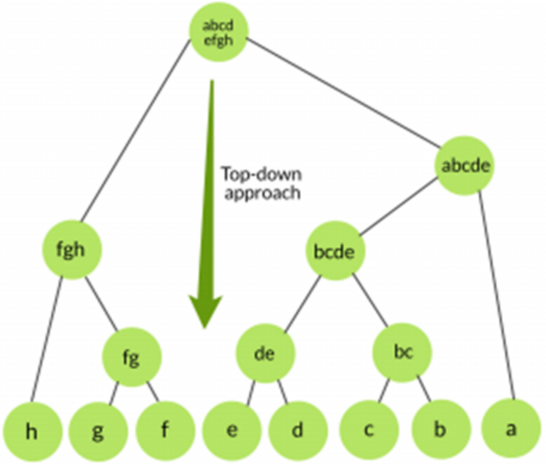
Hierarchical agglomerative clustering

2. Divisive clustering:

1. Also known as a top-down approach.
2. This algorithm also does not require to prespecify the number of clusters.
3. Top-down clustering requires a method for splitting a cluster that contains the whole data and proceeds by splitting clusters recursively until individual data have been split into singleton clusters.

Algorithm:

```
given a dataset ($d_1$, $d_2$, $d_3$, ....$d_N$) of size N
at the top, we have all data in one cluster
the cluster is split using a flat clustering method eg. K-Means etc
repeat
choose the best cluster among all the clusters to split
split that cluster by the flat clustering algorithm
until each data is in its own singleton cluster
```



Hierarchical divisive clustering

Conclusion:

Implemented K means and hierarchical clustering algorithm on the given dataset.

Implementation: