Submitted by: Shalini Uppala

Student ID:23032074

Subject COde:7PAM2021-0901-2024

Report on K –means Clustering

Overview of Clustering: Clustering is an important task in unsupervised learning, a type of machine learning in which the algorithm discovers patterns and structures in data without the use of predetermined labels. The purpose of clustering is to split a dataset into groups, or clusters, so that data points within the same cluster are more comparable to one another than to those in other clusters.

Importance of clustering: It is especially beneficial in exploratory data analysis, where understanding the underlying patterns may lead to better decisions. It is used in various fields like market segmentation, image recognition, document categorization, and customer behavior analysis and so on.

K- means Clustering: K-Means is one of the most basic and widely used clustering algorithms due to its simplicity of implementation and effectiveness in dealing with a variety of datasets. The goal is to partition a dataset into k clusters by minimizing the sum of squared distances between data points and their corresponding cluster centroids [ref 1]. The strategy has been shown to be a very effective way to achieve good clustering results. However, it is ideal for creating globular clusters-means clustering is a partition-based algorithm. The main aim is to ensure that:

- **Intra-cluster Similarity:** The data points in a cluster are as comparable as feasible.
- **Inter-cluster Difference:** Clusters are as separate as they can be.

K-means Clustering Algorithm: In 1967, MacQueen invented the k-means method, one of the most simple, non-supervised learning algorithms, which was employed to tackle the problem of the well-known cluster. It is a partitioning clustering technique that classifies supplied date objects into k separate clusters iteratively, eventually converges to a local minimum. As a result, the clusters that are created are compact and independent.

Steps involve in k-means algorithm is discussed below

Step 1. Initialization:

Choose k initial centroids $\{c1, c2, c3 \ldots . ck\}$ These can be selected randomly from the dataset.

Step 2. Assignment:

For every data point x_i identify the closet centroid. The Euclidean distance is used for this is

$$c_j^{(i)} = \arg \quad \min_{1 \le j \le k} \left|\left| x_i - c_j \right|\right| 2$$

Where

- x_i is i^{th} data point
- c_j is j^{th} centroid
- $\left|\left|x_i - c_j\right|\right|^2$ is Euclidean distance between x_i,c_j.

Each data point is then linked to a cluster, represented by C_j, where

$$C_j = \left\{x_i \middle| c_j^{(i)} = j \middle|\right\}$$

Step 3: Update:

Recalculate the centroids c_j

$$c_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$$

Where,

- $|c_j|$ is number of data points in cluster C_j
- The sum is taken over all data points in cluster C_j.

Step 4: Repeat:

Repeat steps 2, 3 until convergence occurs.

The stopping criteria occurs in 2 conditions.

1. The change in centroids is smaller than a threshold $\in$ i.e.

$$\left|\left|c_j^{(new)} - c_j^{(old)}\right|\right| < \in \text{ for all j.}$$

2.When the algorithm reaches to maximum number of iterations.

Flow chart:

K-means algorithm can be represented Graphically for easy understanding.

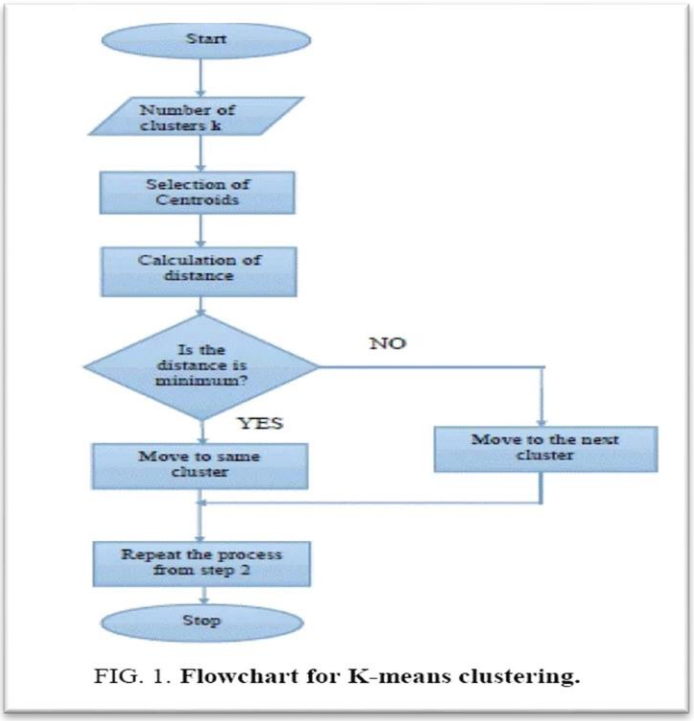

FIG. 1. **Flowchart for K-means clustering.**

Figure 1: Flow Chart of K-means algorithm

Dataset selection:

Mall Customer Segmentation Dataset: A real-world dataset suitable for customer segmentation using K-Means clustering. You may use this dataset to study customer behavior based on their annual income and expenditure score patterns. The dataset having the attributes like Customer ID, Gender, Age, Annual Income, Spending Score (1-100).

Data preprocessing:

Load the data set: import the necessary libraries.

Data Normalization: Data normalization is a critical step in preparing datasets for machine learning models, particularly when the features have varying ranges or units. In this example, we normalized the dataset's numerical columns using Scikit-Learn's MinMaxScaler.

Drop irrelevant columns: Here Customer ID is eliminated since it is not useful for clustering.

Handling Missing Values: The numerical characteristics in the dataset may have missing values. To address this, we fill up the missing values with the mean of their respective columns:

Scaling the Data: K-Means clustering is scale-sensitive. Therefore, we scale the numerical characteristics using Standard Scaler to standardize the data to have a mean of 0 and a standard deviation of 1.

K-Means Clustering in Code:

Step 1: Choosing the Right Number of Clusters (Elbow Method):

Elbow method: It is one of the common methods to choose the optimal number of clusters, which involves by plotting the distortion (inertia) for various K values and looking for the "elbow" point on the plot. Here the distortion (inertia) is calculated as the sum of squared distances between each point and its allocated cluster center. Mathematically the inertia J(k) for k clusters is given by:

$$J(k) = \sum_{i=1}^{n} \sum_{j=1}^{k} \left|\left|x_i - \mu_j\right|\right| 2$$

A large distortion suggests poorly defined clusters, whereas a minor distortion shows well-formed clusters. By visualizing this for various K values, we can visually locate the ideal K at which the rate of loss in inertia slows. The appropriate number of clusters in the elbow technique is often decided at the point where the inertia curve begins to level off (i.e. "elbow").

The "elbow" on the figure occurs at K=3, showing that 3 is the optimal number of clusters. And further increasing *k* won't significantly enhance clustering quality.

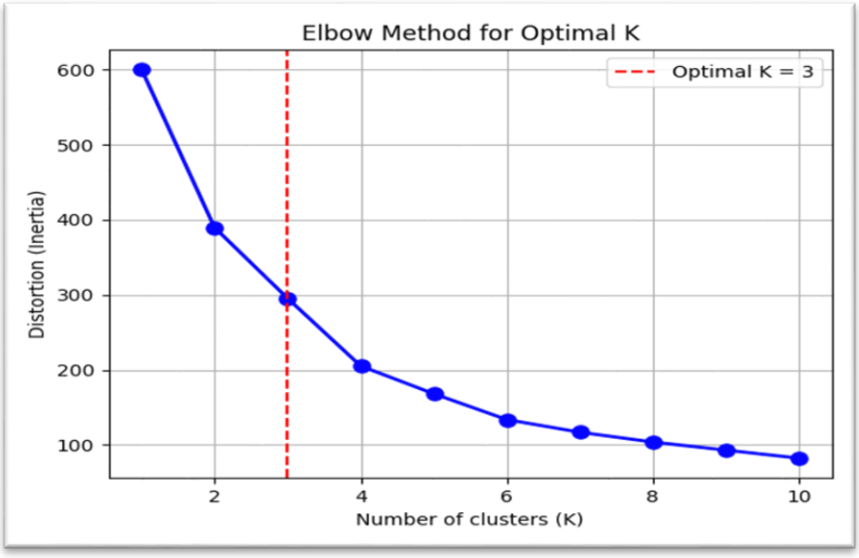

Figure 2: Elbow method

Why Does the Elbow Work?

The elbow technique works because, as the number of clusters increases, the inertia (the sum of squared distances) automatically reduces since more centroids allow for a finer partitioning of data. However, at a certain point, adding more clusters does not result in a meaningful reduction in inertia, implying that the data is overfitted. The elbow is a compromise between too few clusters (resulting in poor grouping) and too many clusters (resulting in unimportant complexity).

Step 2: Fitting K-Means:

Once K is determined we apply k means clustering with the value of k=3 and obtain the cluster labels

Step 3: Assigning Cluster Labels:

The generated cluster labels are applied to the original dataset for further analysis:

Step 4: Visualizing the Clusters:

After applying the K-Means method to a dataset, it is critical to observe how the data points are distributed among the clusters in order to assess the segmentation quality. The visualization helps in determining if the algorithm has effectively recognized unique groups and gives an understandable manner to analyze the findings. It can illustrate the separation between clusters, the compactness of data points within each cluster, and potential overlaps that indicate the need for additional modifications in the number of clusters or characteristics. Clusters are frequently color-coded to make it easier to identify between them.

A scatter plot can indicate how customers are categorized based on their spending habits and income levels when clustering customer data by parameters such as Annual Income (k$) and Spending Score (1-100).

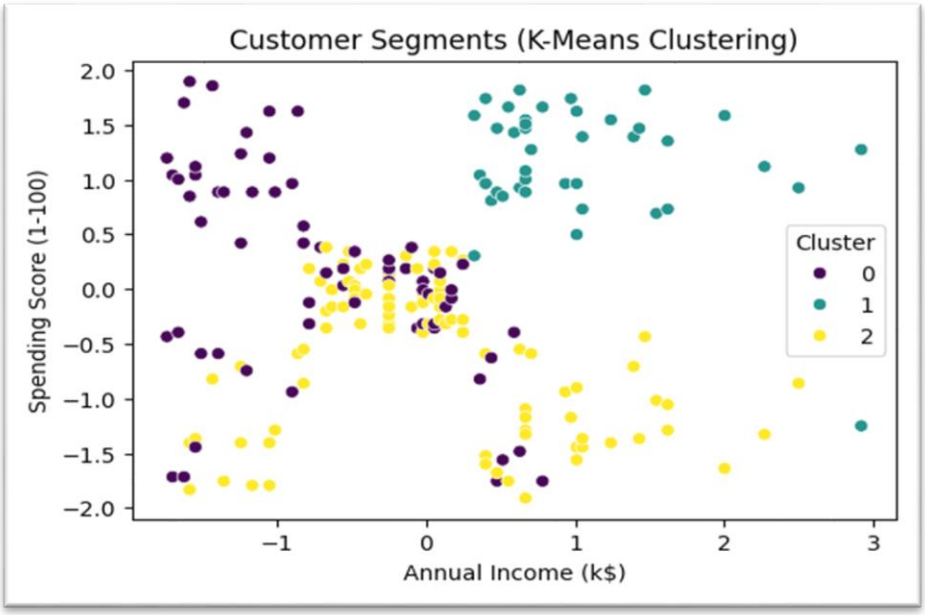

Figure 3: Visualizing the Clusters

From the graph we observe that 3 clusters are formed,

- Cluster 0 represents low income, low spending customers.

- Cluster 1 represents moderate income, moderate spending habits.
- Cluster 2 represents high income, high spending customers.

5. Cluster Analysis and Profiling:

We want to generate a profile for each cluster by grouping the data points according to their cluster assignments and computing the mean values of numerical attributes. This profile aids in recognizing the similarities and differences amongst clusters, resulting in more accurate insights and decisions.

The clusters are well-separated, demonstrating that the K-Means clustering algorithm correctly recognized different client groups. There is very little overlap between the groups, indicating strong clustering efficiency.

Cluster	**Color**	**Annual Income (Normalized)**	**Spending Score (Normalized)**	**Interpretation**
0	Purple	Low (Negative values)	Wide range (Low to High)	Low-income, varied spending behavior
1	Yellow	Average (Near-zero values)	Average (Near-zero values)	Moderate income and moderate spending
2	Teal	High (Positive values)	High (Positive values)	High-income, high-spending customers

6. Silhouette Score:

The Silhouette Score is a statistic used to assess clustering quality by calculating how similar each data point is to its own cluster in comparison to other clusters. It gives a quantitative measurement of cohesiveness (the distance between points within a cluster) and separation. The score ranges between -1 and 1.

- **1** denotes that the clusters are well-separated and the data points are appropriately organized.
- **0** indicates overlapping clusters, which means that the data points are evenly spaced between Clusters.
- **-1** indicates that numerous data points are probably assigned to wrong grouping indicating poor clustering.
- From the graph we observe that Silhouette is **0.36.**

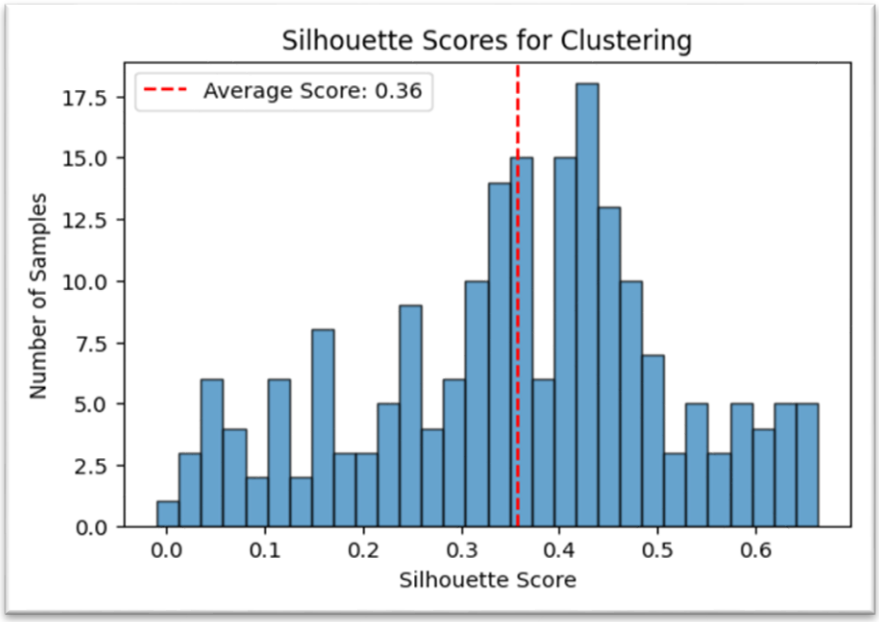

Figure 4: Silhouette Score

Visualizing the Cluster Centroids: The centroids, which appear as red "X" marks on the scatter plot, reflect the average location of the customers within each cluster.

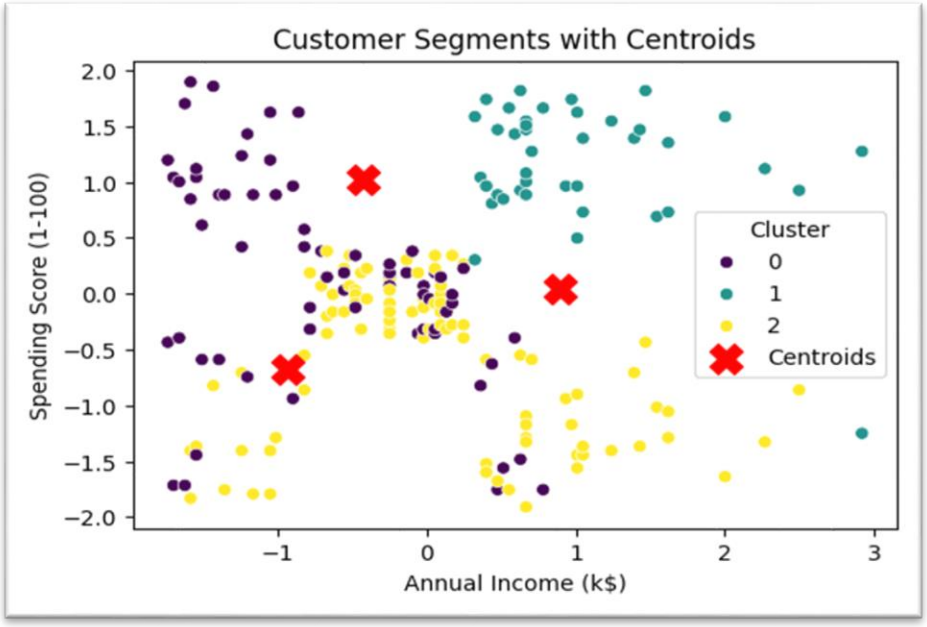

Figure 5: Visualizing the Cluster Centroids

Centroids can be used as summary points for decision-making and customer segment analysis. Their placements are approximate and based on the graph's normalized values. Here is a tabular representation of the observations from the graph:

Cluster	Color	Centroid Mark	Centroid position(approx.)	Annual Income (Normalized)	Spending Score (Normalized)	Interpretation
0	Purple	Red (Centroid)	(-1, 0.5)	Low (Negative values)	Wide range (Low to High)	Low-income, varied spending behavior
1	Yellow	Red (Centroid)	(0, -0.5)	Average (Near-zero values)	Average (Near-zero values)	Moderate income and moderate spending
2	Teal	Red (Centroid)	(2, 1)	High (Positive values)	High (Positive values)	High-income, high-spending customers

Advantages:

1.Simplicity and Ease of Implementation:

K-Means is easy to learn and use. The approach uses minimum processing resources and may be implemented in a few lines of code, making it suitable for beginners as well as professionals.

2. Scalability:

K-Means is scalable and capable of handling big datasets. The temporal complexity is O(n·k·i) where n is the number of data points, k is the number of clusters, and i is the number of iterations. This makes it ideal for clustering datasets with millions of points.

3. Efficient for Exploratory Data Analysis (EDA):

K-Means is effective in exploratory data analysis because it reveals natural groups in the data.

Disadvantages:

1. Not Suitable for High-Dimensional Data:

In high dimensions, distance calculations lose relevance.

2. Limited to numerical data:

It does not function properly with categorical or mixed data types without modification.

3. Converges to local minima:

K-Means may reach a local minimum rather than the global optimum. This implies that the final cluster assignment may not represent the best feasible data segmentation, particularly if the starting centroids are poorly chosen.

Areas of Applications:

1. Customer Segmentation (Marketing and Business):

Customers might be grouped according to their purchase habits, income, or demographics etc.

2. **Image Compression**:

Reducing image size by grouping comparable pixel colors together and replacing them with the group's centroid.

3. Healthcare and Medical Diagnostics:

Patients are grouped according to comparable medical problems or treatment responses.

4.Gene Expression Data (Bioinformatics):

Genes or biological samples are grouped based on expression level similarity.

Conclusion:

K-means clustering is a fundamental technique in unsupervised machine learning that allows data to be divided into discrete clusters based on their inherent characteristics. This study discussed the K-means algorithm, its methodology, and practical applications, using the Mall Customer Segmentation dataset as an example. The method accomplishes meaningful data point grouping through a systematic process of initialization, repeated assignment, and centroid updating. K-means can remain an important tool in data-driven decision-making across a wide range of industries with suitable changes and cautious use.

References:

- Git hub link:**https://github.com/Uppala19/K-means-clutsering**
- Kaggle, "Mall Customer Segmentation Dataset". https://www.kaggle.com/datasets/vjchoudhary7/customer-segmentation-tutorial-in-python

- Scikit-learn: Machine Learning in Python. https://scikit-learn.org/1.5/modules/generated/sklearn.cluster.KMeans.html
- Matplotlib: Visualization library for Python. https://matplotlib.org/
- J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pp. 281-297, 1967.

https://projecteuclid.org/euclid.bsmsp/1200512992.

- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- R. Xu and D. Wunsch, "Clustering," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, May 2005.

  https://ieeexplore.ieee.org/abstract/document/5453745