Data Mining Assignment 2 – Clustering Algorithms

Giridhar Dhanapal – 201682772

1. K-means clustering algorithm: -

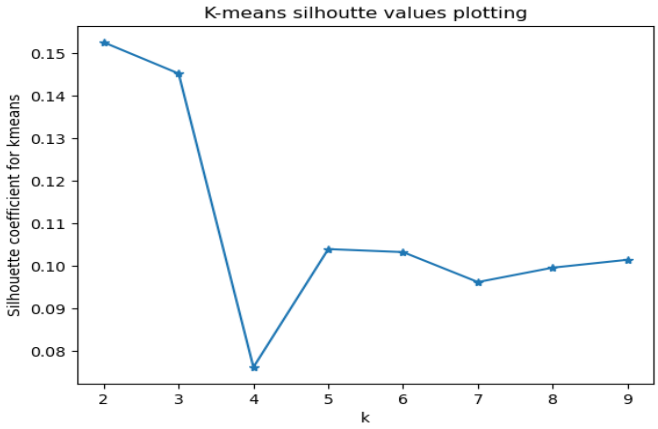
K-means clustering is an unsupervised machine learning approach. The main objective of K-means clustering approach is to combine comparable objects or data points into the same cluster. The number of groups or cluster is represented by k. The algorithm runs repeatedly and assigns each data points to one of the cluster or k groups based on the characteristic that are previously given. The approach functions by determining the centroid of every cluster. The centroid is the mean or average of all data points inside the cluster.

Pseudo code for the k-means clustering algorithm: -

```
X is the dataset
function k_means_clustering(X, k, iters=100, seed=25):
    setting random seed value
    centroids = randomly initialize k centroids from X(dataset)
    for i in range(iters):
        Minimum_distance, centroids = update_centroids(X, centroids, k)
    return Minimum_distance, centroids

function to update_centroids(X, centroids, k):

    Euclidean_distance = []
    for i in range(X.shape[0]):
        distance = []
        for j in range(k):
            calculate Euclidean distance between X[i] and centroids[j]
            append the distance to the list
        append the list of distances to Euclidean_distance
    Minimum_distance = get the index of minimum distance in each row of Euclidean_distance
    for j in range(k):
        centroids[j] = calculate the mean of all X[i] where Minimum_distance == j
    return Minimum_distance, centroids
```



Advantages of K-means clustering algorithm: -

- ✓ K-means clustering is very straightforward to apply to even big data sets and guarantees convergence.
- ✓ Generalizes to clusters of varied forms and sizes, such as elliptical clusters. The method delivers clear and readable outcomes, with each data point allocated to a distinct cluster, making it easier to analyse and examine the structure of the data.
- ✓ K-means clustering may be used for a broad variety of applications, like customer segmentation, picture classification, and anomaly detection.

Disadvantages of K-means clustering algorithm: -

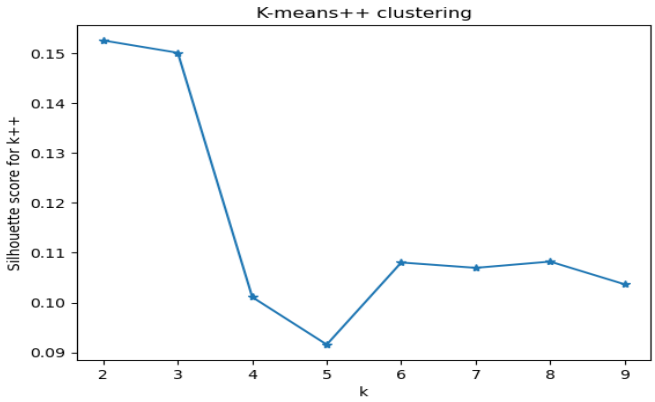
- ✓ The outputs of k-means clustering could vary greatly depending on the starting position of the cluster centroids.
- ✓ Choosing the ideal number of clusters for a specific dataset may be hard.
- ✓ K-means clustering is subject to outliers, which may alter the position of the cluster centroids and lead to less accurate cluster choices.
- ✓ K-means approach is prone to converging to a local minimum.

2. K-means++ clustering algorithm: -

The k-means++ method is an upgrade on the standard k-means algorithm. The k-means++ technique tries to pick better starting cluster centroids than those generated randomly in the original k-means algorithm, resulting in superior clustering outcomes. K-means++ differs from the conventional K-means solely in the initialization stage. K-Means++ picks one initial centroid evenly at random from the data points. Next centroid is selected from the data points with probability proportional to the square of the distance to the next centroid that has been previously picked.

Pseudo code for the k-means++ clustering algorithm: -

```
function k_plusplus(X, k, seed=25): # for initializing the centroids
    set random seed to seed # setting seed value so that the output will be constant
    centroids = [randomly select a single data point from X]
    while the number of centroids is less than k:
        distances = []
        for i in range(len(X)):
            dist = []
            for j in range(len(centroids)):
                Euclidean distance = Distance between X[i] and centroids[j]
                append the distance to the list dist
            append the minimum value in dist to the list distances
        distances = converting distance list to array
        probabilities = distances / np.sum(distances)
        select a new centroid from X based on probabilities using np.random.choice
        appending new centroid we got to list centroids
    return the array of centroids
```



Advantages of K-means++ clustering algorithm: -

- ✓ The k-means++ algorithm picks initial centroids that are well isolated from each other. This decreases the probability of becoming caught in a local minimum and gives more precise and consistent results.
- ✓ The k-means++ method often generates higher quality clustering results than the traditional k-means algorithm. Because the algorithm is less sensitive to the initial choice of centroids and is more likely to identify the global optimum solution.

Disadvantages of K-means++ clustering algorithm: -

- ✓ The k-means++ technique needs the user to define the number of clusters to be in the data, choosing the improper number of clusters could result to poor clustering results.
- ✓ The k-means++ approach may be computationally costly, particularly when working with huge datasets or high-dimensional data.

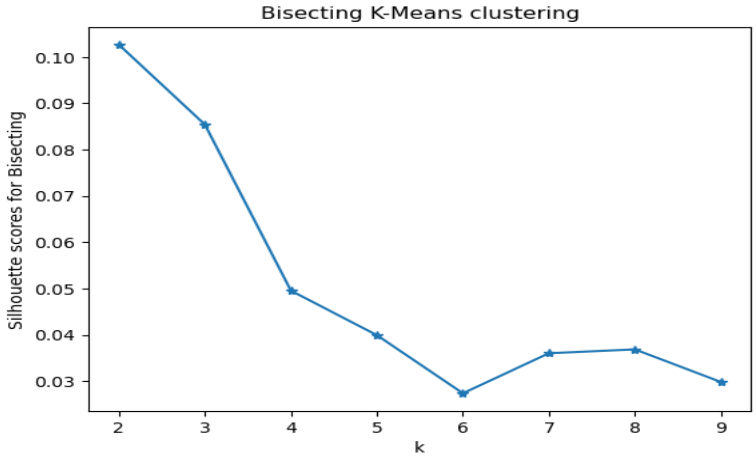
- ✓ The k-means++ technique is also susceptible to outliers, which may dramatically impact the clustering results. Outliers may cause the algorithm to construct centroids that are not typical of the data, resulting to poor results.
- ✓ The k-means++ method performs good when the clusters are well-separated and have a spherical form. It may struggle to detect non-linear clusters, such as clusters with irregular forms or clusters that overlap with one other.

3. Bisecting k-Means hierarchical clustering algorithm: -

The Bisecting k-Means hierarchical clustering algorithm is a sort of hierarchical clustering technique that relies on the k-means algorithm to build a hierarchical grouping. This approach is based on splitting a cluster into two halves until a desired number of clusters is attained. Choose the number of clusters you intend to produce (k)and then randomly initialise k centroids. Assigning each data point to the nearest centroid. Each clusters new centroid is calculated by taking mean of all data points that are allotted to that cluster.

Pseudo code for Bisecting k-Means hierarchical clustering algorithm: -

clusters = [X] # Start with the entire dataset X as a single cluster
While length of cluster is < k:
  Largest_cluster = Select the largest cluster from the list of clusters
  Applying k-means algorithm to the largest cluster with k=2 to obtain two new clusters
  Removing the largest cluster from the list of clusters
  Adding the two new clusters to the list of clusters
return clusters
Repeat steps until the desired number of clusters has been reached.



Advantages of Bisecting k-Means hierarchical clustering algorithm: -

- ✓ Bisecting k-Means can handle large datasets efficiently since it only performs the k-means algorithm on a subset of the data at each step.

- ✓ Bisecting k-Means has been shown to produce high-quality clusters, particularly when the clusters are of different sizes.

Disadvantages of Bisecting k-Means hierarchical clustering algorithm: -

- ✓ Bisecting k-Means can be computationally expensive, especially for large datasets or when the number of clusters is large.
- ✓ It can be difficult to determine the optimal number of clusters.
- ✓ The algorithm tries to create equally sized clusters at each level of the hierarchy, which can bias the results towards clusters of similar size, it is problematic if the data contains clusters of very different sizes.

7. Comparing the different clustering algorithms

By comparing the three different clustering algorithms, I came up with the conclusion that the K-means++ clustering algorithm gives better clustering results compared to the other two algorithms. The silhouette score values range between -1 to 1 in which -1 represents that the data point is wrongly classified, if the silhouette score is zero it means that the clusters are overlapping and +1 indicates that the cluster is most likely correctly classified. In my observation, I got a better silhouette coefficient for K-means ++ clustering.