CSE5DMI\_Assignment2 Report

~ Ananthakrishnan Nikarthil Baburaj

~ 21956917

Objective:

To analyse the dataframe “Wholesale Customers – A.csv” with two unsupervised machine learning clustering algorithms namely K-Means clustering & Hierarchical clustering & one supervised algorithm which is SVM or Support Vector machine & then answer the following queries.

Q1. Is it necessary to do normalization on the dataframe?

Q2. Justify the choice of optional parameters for both clustering algorithms?

Q3. For each clustering method, interpret the cluster centroids & describe the characteristics of each customer segment. Evaluate the clustering performance of K-Means and Hierarchical clustering using quantitative metric Silhouette Score. Finally, discuss strengths and weaknesses of K-Means and Hierarchical clustering with reference to your experimental results.

Q4. Compare the performance of the supervised machine learning method SVM with KMeans for predicting the target label “Region”. Suggest potential business strategies based on the discovered clusters.

Procedure:

Step 1: Data-preprocessing

a. Using pandas library read the dataframe. Then use info() function to look for any null values. Since there are no null values, we don’t have to skip imputation.

b. Proceed to use StandardScaler() to normalize the dataframe.

Step 2: Determining the optimal parameters

a. By plotting the mean silhouette values for a number of clusters ranging from 2 to 5, select the optimal number of cluster for the case of KMeans

b. For Hierarchical clustering, determine the optimal number of clusters & the linkage criterion by plotting dendrograms for all the linkage criterions.

Step3: Implementing the clustering algorithms

a. Use PCA to reduce dimensionality of the dataframe.

b. Implement the two clustering algorithms using the optimal parameters & visualize them.

c. Evaluate the mean silhouette score of the two clustering algorithms & also print the cluster centroids. Since there are no inbuilt methods for tabulating centroids for hierarchical clustering, we just have to take the average values of all the features of the dataframe grouped by cluster values.

Step 4: Compare the performance of SVM with KMeans for predicting “Region”

a. Separate the original dataframe into features “X” & target “y”(“Region”)

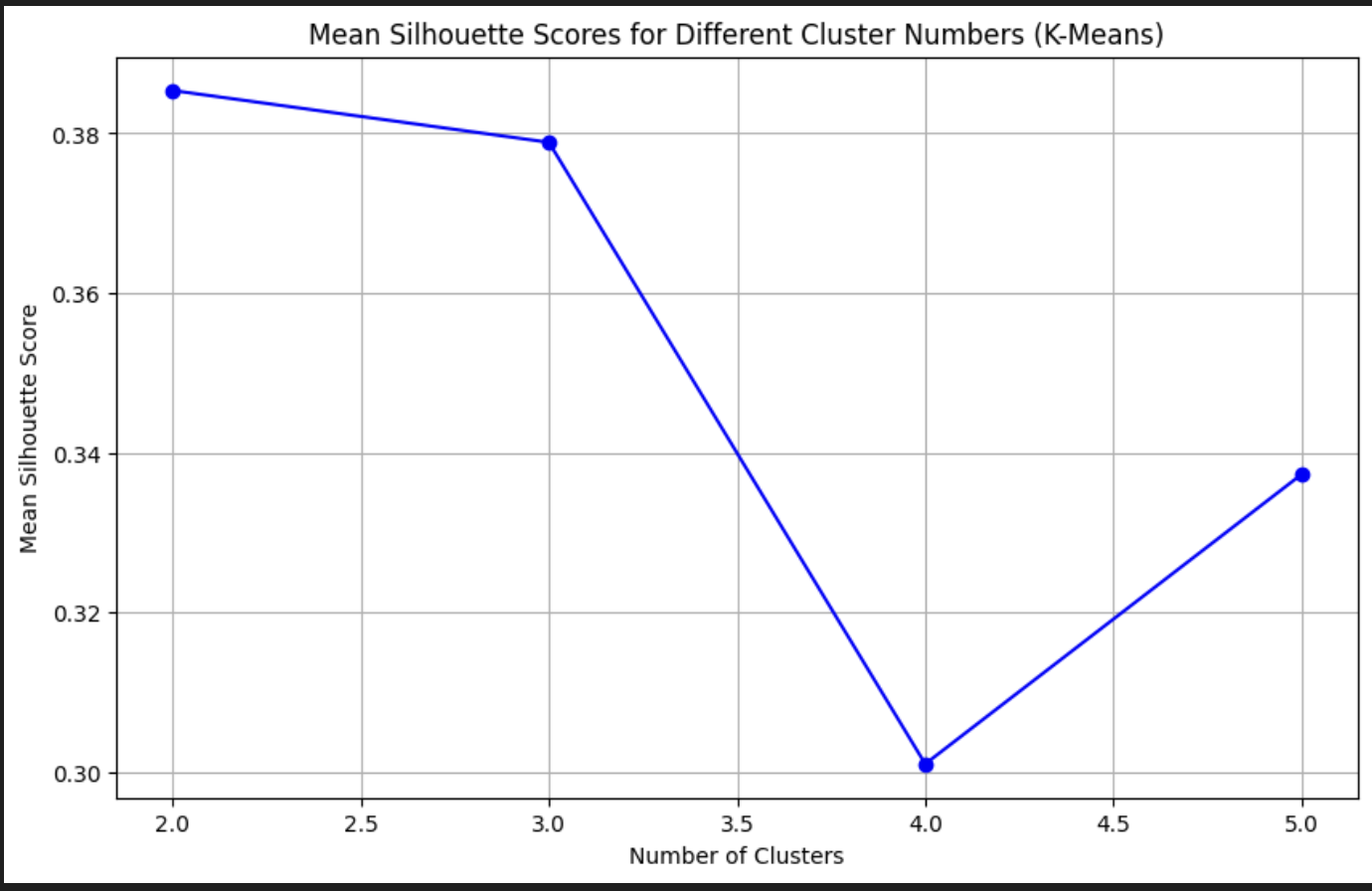
b. Scale X using StandardScaler()

c. Train an SVM model, make predictions & gauge it’s accuracy.

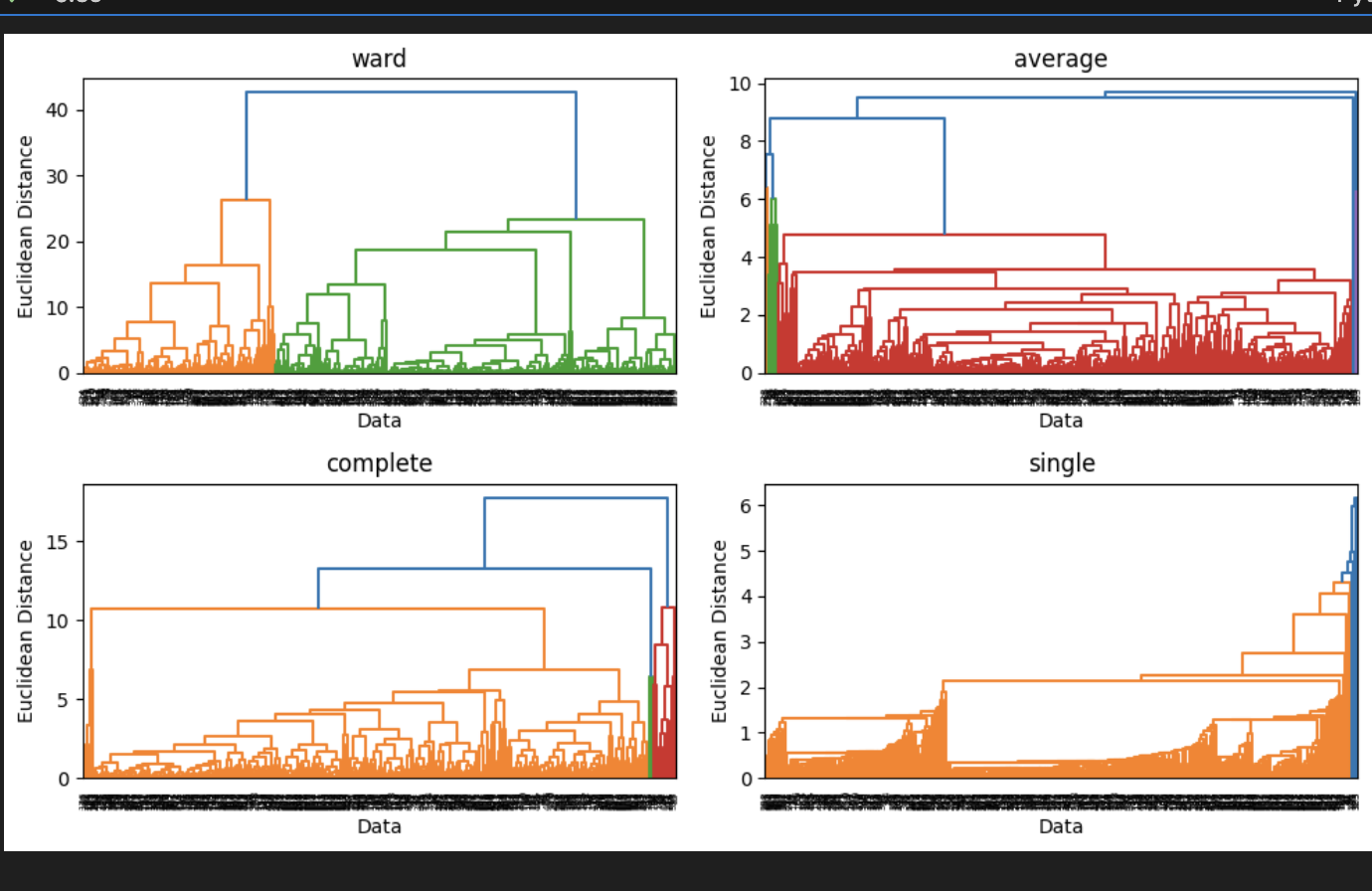
d. Perform the same operations for a new KMeans model with 3 clusters(for 3 regions) & compare the accuracy with that of the SVM model.

Results:

* Mean-Silhouette Graph:



* Dendrograms for hierarchical clustering:



* KMeans cluster

A graph of a number of dots

Description automatically generated

* Hierarchical cluster

A diagram of clusters of dots

Description automatically generated

* K-Means Cluster Centroids:

Channel Region Fresh Milk Grocery \

Cluster

0 1.054487 2.509615 13767.737179 3383.195513 3973.743590

1 1.976562 2.625000 7692.164062 11678.125000 17646.515625

Frozen Detergents\_Paper

Cluster

0 3725.368590 814.000000

1 1479.179688 7921.007812

* Hierarchical Centroids:

Channel Region Fresh Milk \

Cluster

1 1.039474 2.506579 13363.723684 3157.164474

0 1.955882 2.625000 8952.639706 11695.433824

Grocery Frozen Detergents\_Paper

Cluster

1 3814.674342 3439.970395 750.759868

0 17197.801471 2249.257353 7644.308824

* Mean Silhouette score for Kmeans method: 0.5341852355100489
* Mean Silhouette score for Hierarchical method: 0.5174408533430241
* SVM Accuracy: 0.8409090909090909

SVM Classification Report:

precision recall f1-score support

1 0.00 0.00 0.00 12

2 0.00 0.00 0.00 9

3 0.84 1.00 0.91 111

accuracy 0.84 132

macro avg 0.28 0.33 0.30 132

weighted avg 0.71 0.84 0.77 132

* Accuracy of KMeans for region prediction: 0.29318181818181815

Conclusion:

Q1.

It is indeed necessary to implement normaliztion. The magnitude of certain features like “Region” & “Channel” are dwarfed by the others by a large margin. This tends to lead our algorithms to be disproportionally affected by different features & to not pay enough attention to features that could provide more insight into the data. Thus we use “StandardScalar()” to scale the features to a form where their mean is 0 with a standard deviation of 1, preventing any feature from being overlooked.

Q2.

For KMeans clustering, the most important parameter is to find the optimal number of clusters. To achieve this, we compare the mean silhouette values for a variety of cluster values. Mean silhouette is a metric that interprets intra cluster distance & nearest cluster distance to calculate a value that indicates how well each data point is clustered & how well they are separated from each other. Using the Mean-Silhouette graph shown in Results, we can conclude that the number of optimal clusters for our use case is 2.

For Hierarchical clustering, we must choose the optimal number of clusters based on dendrograms created using Euclidean distance between the data points. Based on the subplots in Results, it is safe to decide on the linkage criterion as ‘ward’ & the optimal number of clusters as 2.

Q3.

Consider the centroids shown in Results for both clustering algorithms:

Using the centroids from KMeans algorithm, we can summarize that customers belonging to cluster 0, will have a tendency to belong to “Channel” 1 & “Region” of either 2 or 3. They will generally spend around 13768 monetary units on “Fresh” foods, 3383 units on “Milk”, 3974 units on” Grocery”, 3725 units on “Frozen” foods & 814 units on “Detergent\_paper” annually. Meanwhile, customers belonging to cluster 1 will have a tendency to belong to “Channel” 2 & “Region” of either 2 or 3(leaning slightly more towards 3). They will generally spend around 7692 monetary units on “Fresh” foods, 11678 units on “Milk”, 17646 units on” Grocery”, 1479 units on “Frozen” foods & 7921 units on “Detergent\_paper” annually.

Using the centroids from Hierarchical algorithm, belonging to cluster 1, will have a tendency to belong to “Channel” 1 & “Region” of either 2 or 3. They will generally spend around 13363 monetary units on “Fresh” foods, 3157units on “Milk”, 3815 units on ”Grocery”, 3740 units on “Frozen” foods & 751 units on “Detergent\_paper” annually. Meanwhile, customers belonging to cluster 0 will have a tendency to belong to “Channel” 2 & “Region” of either 2 or 3(leaning more towards 3). They will generally spend around 8953 monetary units on “Fresh” foods, 11695 units on “Milk”, 17198 units on” Grocery”, 2249 units on “Frozen” foods & 7644 units on “Detergent\_paper” annually.

Based on the two cluster graphs & the two centroid values from Results , we can summarize that the cluster 0 in KMeans is very similar to cluster 1 from Hierarchical & the cluster 0 from Hierarchical is similar to cluster 1 from KMeans. From such similar observations on the 2 models, we can note that they both will cluster the dataframe to a similar degree with minor differences.

From results, the mean silhouette for KMeans clustering 0.5342 & the mean silhouette value for Hierarchical clustering is 0.5174. Such similar values also support the above conjecture with KMeans clustering faring slightly better. Since the mean silhouette scores for both methods are around 0.5, we can conclude that they both produce a moderate level of clustering quality. Data points are closer to their own cluster than the neighbouring ones. However, there might still be some ambiguity or overlapping.

Both methods have their strengths and weaknesses.

KMeans method is faster to compute & highly scalable making it suitable for larger dataframes. However, they are sensitive to the initial centroids which may cause some variance in the clustering results. Moreover, it likes to assume rigid spherical boundaries which may also cause some inconsistencies. Based on the visual diagram of KMeans clustering as seen in Results, we can see how the boundaries are not that rigid which could explain why the mean silhouette is only decent.

Hierarchical clustering is more computationally expensive compared to KMeans, making it slower & more difficult to scale. However, it allows for a clear visual interpretation of the optimal number of clusters through the use of dendrograms & unlike KMeans, it doesn’t assume that clusters are spherical which allows for more flexibility. The mean silhouette score is close to KMeans & still considered only decent.

Although KMeans is slightly more suited to cluster this dataframe, both of them still succeed in clustering the data reasonably well.

Q4.

SVM or Support Vector Machine is a supervised model designed for predicting target labels while KMeans is a clustering algorithm designed to make sense of datasets by organising them into clusters. This means that SVM’s are very suitable to this scenario of predicting “Region” values from the analytics of the other features while KMeans is not. This is reflected in the accuracy scores shared in Results where SVM scores a respectable 84% while KMeans makes predictions with an unacceptable score of around 30%.

Viable business strategies can be implemented by observing the centroid values of both clustering methods. Businesses should shift the allocation of resources from for regions or channels connected to the different clusters.

Since both clusters have a centroid region value of around 2.5 for both methods, this means that all resources should be more shifted towards regions 2 & 3 from 1 by some degree to increase profit margins.

Unlike regions, both clusters correspond to a specific channel. This means that we can shift resources originally allocated to one channel towards the one if it’s centroid value is lower than it’s counterpart as these centroid values corresponds to the popularity of those resources in that channel. Since fresh & frozen foods are more popular in channel 1, businesses should divert some of the fresh & frozen foods that was to be supplied to channel 2 towards channel 1. Using the same logic, shipments of milk, groceries & detergent paper should be shifted towards channel 2 from channel 1.