CS60050: Machine Learning Mini Project 3

[CS4]Cricket Format Clustering using Single Linkage Hierarchical Clustering Technique

[By Ayan Chakraborty 18EC10075]

Formulation of Problem Statement

1) Implement K-Means Clustering of the unlabelled data, and choose the optimal value of K (number of clusters), depending on the value of the Silhouette Coefficient in each case.
2) Use the optimal value of K and implement a bottom-up hierarchical clustering algorithm using single linkage strategy.
3) Compare the similarity between clusters generated in each case, by using the Jaccard Similarity metric

Brief Theory and Methodology

Distance Metric Used: The distance metric is used as a measure of dissimilarity between 2 vectors. The lower the value of the distance metric for two vectors, the more similar they are. The distance metric used in our case is the Euclidean Distance, which is defined as:

$$d(A,B) = \sqrt{\sum_{i=1}^{N} (a_i - b_I)^2}, \text{ where } A = < a_1, a_2, a_3, ..., a_N > \text{ and } B = < b_1, b_2, b_3, ..., b_N >$$

Data Pre-Processing Method used: The data must be pre processed to bring the value of the features in the same range, otherwise features having a higher range of values will dominate the similarity calculation. We have used the Z-score normalization on each column of our dataset (i.e., on each feature for all examples). It is defined as follows:

$X' = \frac{X - \mu}{\sigma}$, where, $\mu = mean\ value\ and\ \sigma = standard\ deviation\ of\ the\ feature$

Silhouette Score: The silhouette coefficient is used as a metric to judge the result of the clustering algorithm. It is defined for a single sample as follows:

$s = \frac{b - a}{max(a, b)}$, where,

$a = mean\ distance\ between\ sample\ and\ all\ other\ points\ in\ the\ same\ cluster$

$b = mean\ distance\ between\ a\ samples\ and\ all\ other\ points\ in\ the\ next\ nearest\ cluster$

Calculation of 'a' is straightforward. It is just the average of distances to all points sharing the same cluster label. However for calculation of 'b', first, we need to find out the nearest cluster. There can be many different ways to do this, and 2 different metrics have been used individually on the K-Means and the Hierarchical Clustering Algorithm as follows:

K-Means: The distance between centroids of 2 clusters is used. The cluster having the nearest centroid is chosen. This is done because we have the value of clusters available to us, so it is computationally easier and can be done in a single step.

Hierarchical Clustering Algorithm: In this case, we have the proximity matrix is available to us, which tells us the minimum distance between any two clusters calculated using Single Linkage metric (explained later). Hence, we just directly use this table and choose the cluster having the smallest distance from the current cluster

The Silhouette Coefficient for a set of samples is given as the mean of the Silhouette Coefficient for each sample. The score is bounded between -1 for incorrect clustering and +1 for highly dense clustering.

Choosing the initial K random centroids for the K-means clustering Algorithm: The performance of the K-Means clustering algorithm has been found to be heavily dependent on the choice of the initial centroids of the K clusters. Selecting the initial K points randomly gives large variations on the Silhouette score across multiple runs. The results of this random initialization have been compared with another algorithm known as the K++ algorithm. The K++ algorithm seeks to maximize the distance between the initial centroids, and, hence leads to better results. [Link to original paper]. The centroid initialization of the K++ algorithm works as follows: [D(x) denote the shortest distance from a data point to the closest center we have already chosen]

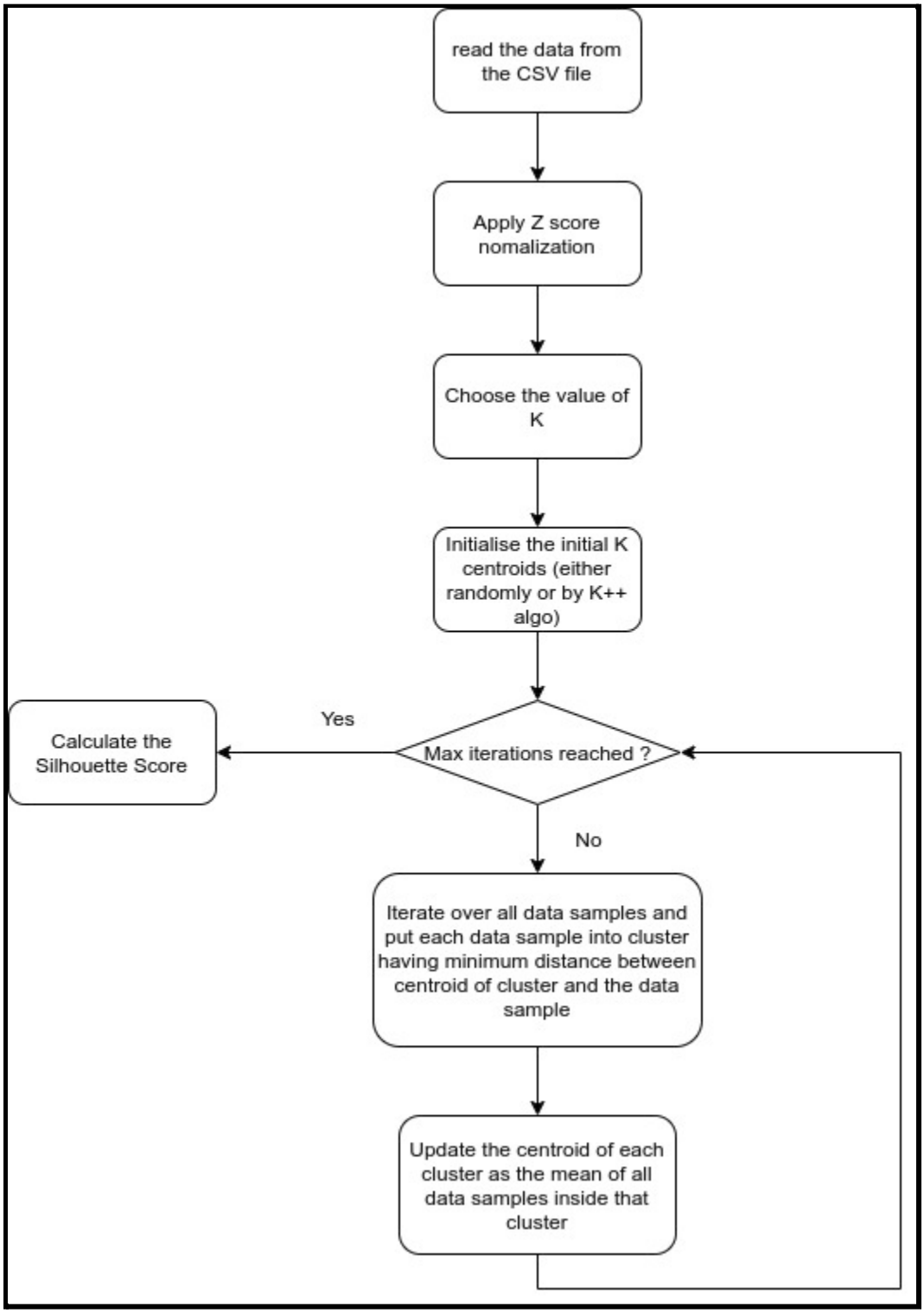
1) Take one centroid c_1, chosen uniformly at random from X .
2) Take a new centroid c_i, choosing x ∈ X with probability $\frac{D(x)^2}{\sum_{x \epsilon X} D(x)^2}$
3) Repeat Step 2 until we have taken K centroids altogether.

[The probability measure is done by first calculating the D(x), then the CDF and then picking the centroid as the data point having its CDF value just lesser than a value generated by a uniform random number generator between 0 to 1]

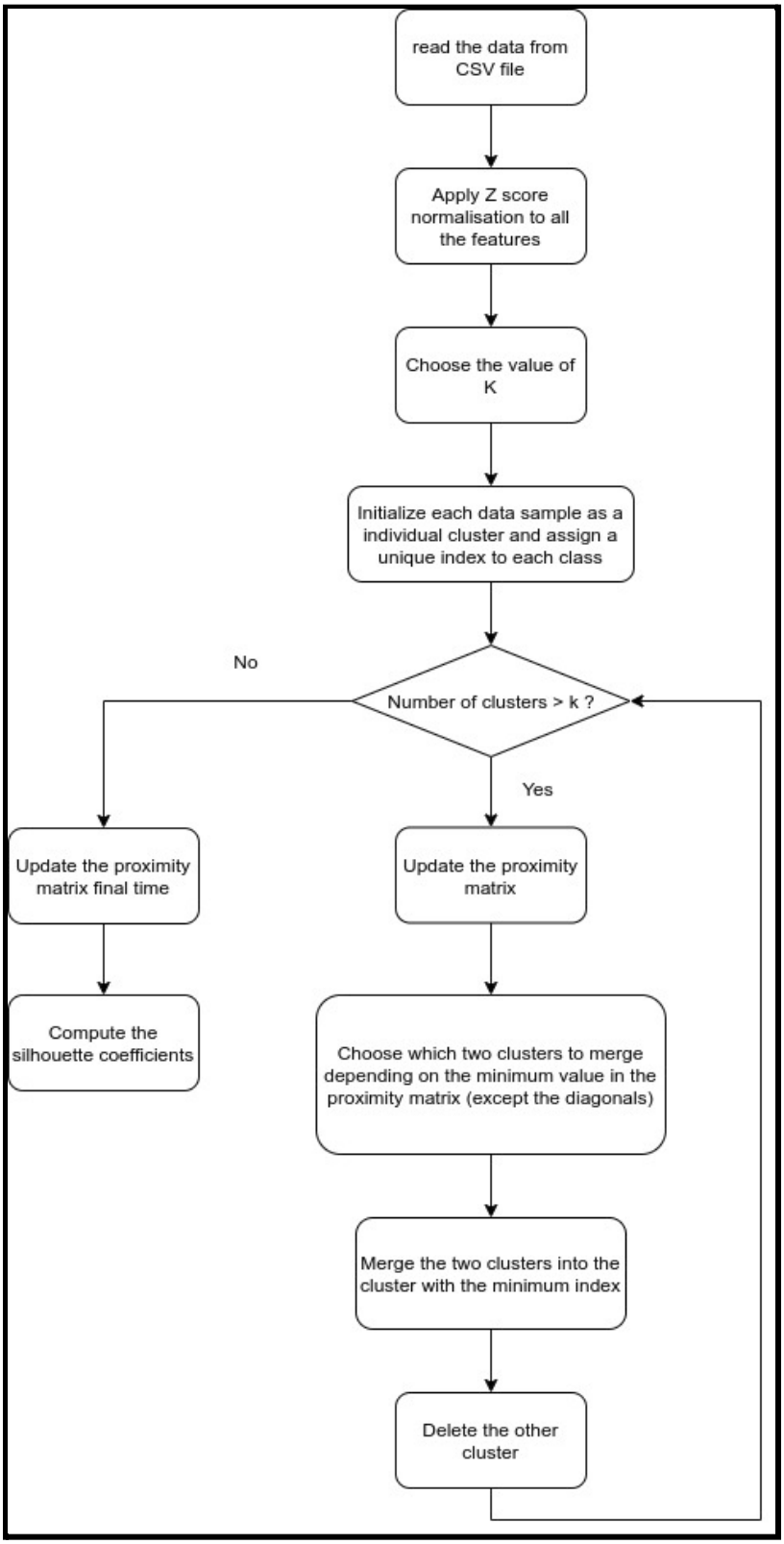
Bottom up Hierarchical clustering using Single Linkage Strategy: Bottom up basically means initially, each of the data points behaves as an individual cluster, and then at each step, we recursively merge two of the closest clusters. Hence, at each step, the number of clusters reduces by 1. We stop when the number of clusters = K. Single Linkage refers to the distance metric between 2 clusters and is computed as the minimum distance between any 2 points, each of which is present in 1 of the 2 clusters.

Jaccard similarity: We have got K clusters each following the K-means and the Hierarchical Clustering methods. This is used as a metric to compare the similarity of 2 clusters, and is defined as follows: $J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$, for two sets A and B. In our case, there can be K x K combinations, and the jaccard index is computed for each combination. Ideally, there should be a 1-to-1 mapping between the clusters computed in each case. The jaccard index should be 1 for these combinations and 0 everywhere else.

Flowchart of the K-Means Algorithm

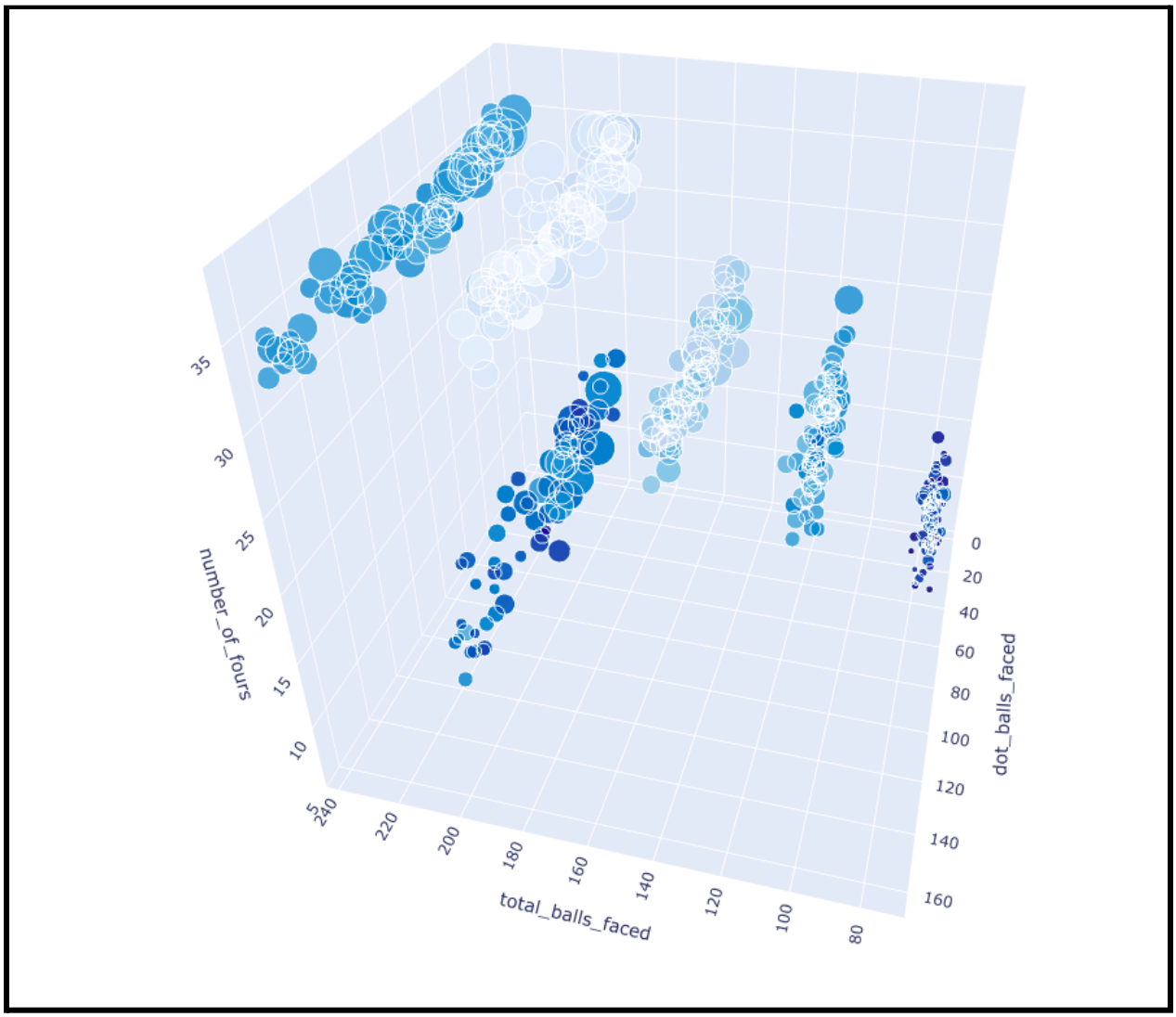


Flowchart of the Hierarchical Clustering Algorithm



Plots of Results and Hyper-Parameter Tuning

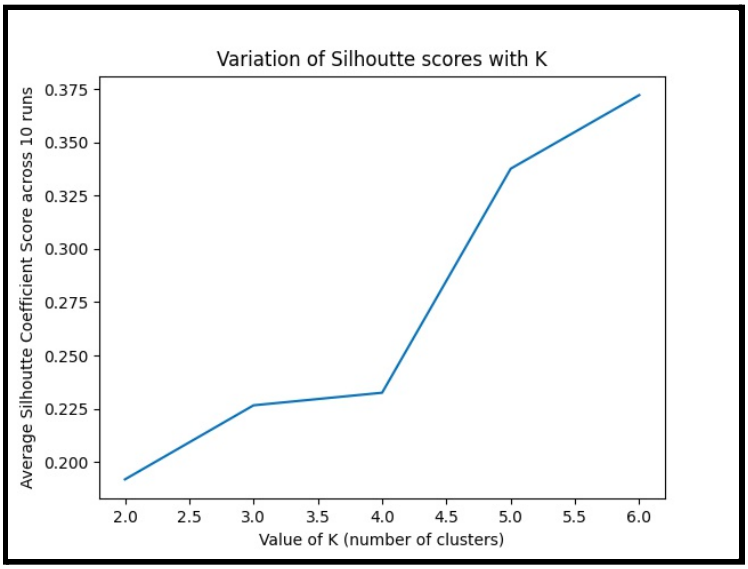
Visualization of the data:



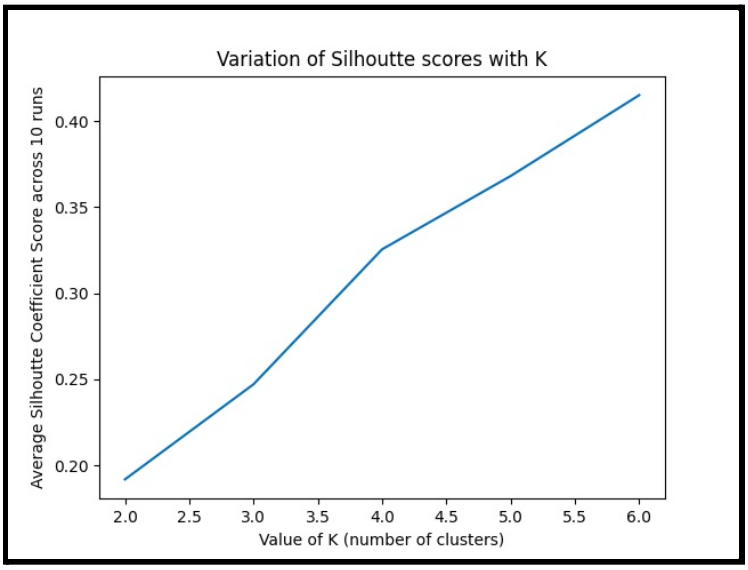
The data is 5 dimensional, so the data cannot be plotted directly. The data is plotted in 3 dimensions. The size and the color of the markers are the 2 extra dimensions to the data. [Note: One set of data points are very faintly coloured, so they might miss the eye but they are actually present if we look closely at the top left corner of the image]

From the visualization, we can observe that it is nicely separated into 6 clusters on the basis of 3 dimensions: <number_of_fours>, <total_balls_faced>, <dot_balls_faced>. It is not easy to visualize the clusters that might be present due to the other 2 dimensions. Still, it provides a nice visualization of the data

Results of the K-Means Algorithm (Using random initialization) averaged across 10 runs for different values of 'k'



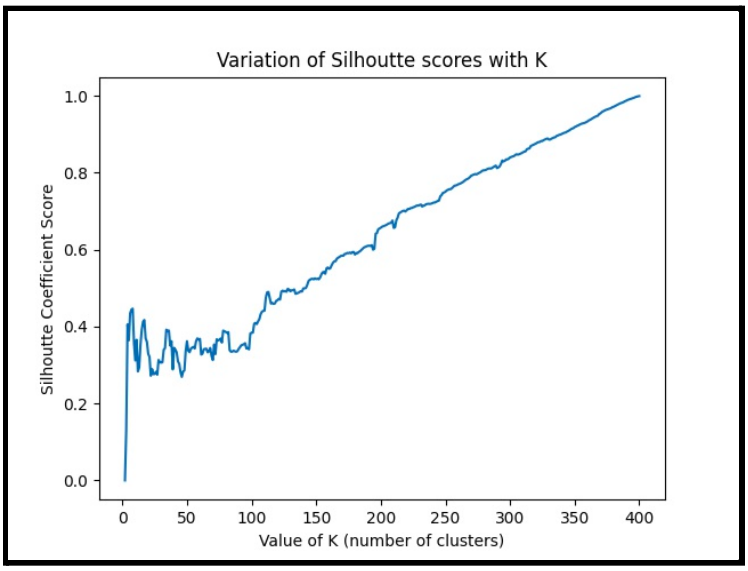
Results of the K-Means Algorithm (Using K++ initialization) averaged across 10 runs for different values of 'k'



Observation Table for Silhouette Scores

Initialization	K = 2	K = 3	K = 4	K = 5	K = 6 (optimal)
random	0.192	0.226	0.232	0.337	0.372
K++	0.192	0.247	0.325	0.368	0.415

Results of the Bottom Up Hierarchical Clustering using Single Linkage method



We observe that the maxima of silhouette scores appears at K = 8, (when considered among a reasonable number of clusters < 100). Silhouette scores from K = 2 to 8:

K = 2	K = 3	K = 4	K = 5	K = 6	K = 7	K = 8
0	0.132	0.406	0.364	0.434	0.444	0.446

Jaccard Similarity mapping using optimal number of clusters = 6 (generated by K-Means Algorithm)

		Hierarchical Clustering Algorithm					
	Class ID	0	1	2	3	4	5
K-Means Cluster Algo	0	0.866	0	0	0	0	0
	1	0	0.708	0	0	0	0
	2	0.005	0	0.985	0	0	0
	3	0.060	0.18	0	0.28	0.008	0.017
	4	0.034	0	0	0.534	0	0
	5	0	0	0	0	0.985	0

Discussions and Conclusions

1) The silhouette score is positive in all cases, which indicates that clustering is being done properly. For the K-Means Algorithm it is increasing as K is increased, which is as expected from the visualization of the data. Since, we are only considering in the range of 2- 6, as mentioned in the problem statement, the optimal number of clusters is chosen as 6 since it has the highest silhouette score (≈ 0.415). If the range of K is increased, then the optimal value of K might change. On printing the number of data samples in each cluster for K = 6, we get 66, 68, 74, 62, 69, 61, which actually shows that the data points have been uniformly distributed in the 6 clusters. However, in some iterations of the code, K = 5 has also been reported as the optimal value of K. But the number of times it has happened is low, and most of the times, it gives K = 6, hence, K = 6 has been chosen as the optimal number of clusters.

2) On running Sci-kit library, we get the value of silhouette score for K = 6 as 0.3583. Our average silhouette score for K = 6 is 0.415, hence, this implementation is performing at par with the standard library. The distance metric used is very important. Earlier, when cosine similarity was being used, the silhouette score was coming in the range of 0.1 to 0.2. This was because the data was originally partitioned using Euclidean distance based measure, hence, Euclidean distance works best as both similarity measure and Euclidean Measure.

3) Initialization of the initial K centroids is very important, because if it is not chosen correctly, then it can get stuck in a local maxima or converge very slowly, and hence, not provide correct results. When the centroids are initialized, there are a lot of variations across different runs, with the silhouette scores ranging from 0.154 to 0.51 for K = 6. Hence, 10 runs are done and the average silhouette score is taken. Another approach to solve this problem has been taken by using the K++ algorithm as explained in the theory to initialize the centroids. The variations have been reduced by a large amount. Now, the silhouette scores are in the range of 0.324 to 0.51 for K = 6. Also, when the average scores for random initialization and K++ initialization are compared for different K, as shown earlier in the observations, then the K++ initialization gives higher mean silhouette coefficient values for each K. Thus as predicted, the K++ initialization is an improvement over the random initialization.

4) Then, we perform the Hierarchical Clustering using Single Linkage and the variation of Silhouette scores is observed. From K in 0 - 100, it varies nicely showing global maxima at K = 8 and several local maximas. This reinforces the result that increasing the number of clusters arbitrarily does not increase the goodness of our clustering. However, after a certain point, the Silhouette score starts increasing monotonically. This is because the number of points per cluster is reducing, and finally only a single point will be left per cluster and hence, the intra cluster distance tends to 0. Hence as a = 0, then max(a, b) = b and hence (b-a)/max(b, a) = b/b = 1.

5) Next, we calculate the Jaccard similarity for each combination of clusters produced by the K-Means and the Hierarchical clustering respectively and try to make a 1-to-1 mapping. We expect 6 very high scores and the rest 0, since, each class is expected to map to a single other class only. We observe that, 4 of the combinations give very high scores (≈ 0.9), 1 of the combinations give moderately high scores (≈ 0.5), 1 of the combinations gives a low score (≈ 0.3), and the rest 0. This again aligns with our expectations, but we observe that 1 to 1 mapping is not possible because 1 cluster produced by the hierarchical clustering does not give a high Jaccard score with any other cluster. When the number of data samples in that cluster was checked, it was found to contain only a single data sample. Hence, it is expected that it is an outlier point and hence, negatively affecting the clustering. Hence, the 1 to 1 mapping is not possible because all the clusters produced by the K-Means have the number of samples uniformly distributed around 60. The K-means algorithm does not seem to be affected by this problem, since the updation of the centroid requires taking the average hence, the effect of outlier is reduced. On the other hand, since the hierarchical clustering uses single linkage, only a single distance (i.e., the minimum distance) is important so it becomes susceptible to outliers.

6) However, in some of the cases, when the optimal number of clusters was reported as 5, then the corresponding Jaccard Matrix is:

		Hierarchical Clustering Algorithm				
	Class ID	0	1	2	3	4
K-Means Cluster Algo	0	0.918	0	0	0	0.005
	1	0.019	0.428	0	0.248	0
	2	0.005	0	0.985	0	0
	3	0.025	0.218	0	0.406	0
	4	0	0	0	0	0.985

In this case, we get a very nice 1 to 1 mapping between each class generated by the KMeans Algorithm and the Hierarchical Algorithm. This is because that the outlier that was present in the case of K = 6, has now been merged with another cluster and now, the sizes of the clusters are in the same range.