Homework Assignment 3 : K-Means

Weerdhawal Chowgule

February 25, 2018

1 Introduction

K-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as vary centers of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more. Finally, this algorithm aims at minimizing an objective function, which here refers to the different distance metrics used
Although it can be proved that the procedure will always terminate, the k-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centers. The k-means algorithm can be run multiple times to reduce this effect.

2 Preprocessing Dataset

In the first assignment we created a Document-Term matrix (DTM), it was used. Also, the sparsity was considered as 0.99, but as the matrix was to huge. Hence, I changed the sparsity to 0.85. The DTM consists of 18828 documents and 270 features (i.e. the words).

Feature selection was performed, and top 100 words were selected. A list of top features were found in the last assignment, I took the column(feature) names and extracted only those columns from the main DTM and a file named ‘checkfinal.csv’ is obtained. Also due to the huge size of the data set I applied random sampling while executing the code.

3 Packages and Libraries

The program has been scripted in Python 3.6 and a varied list of modules were used to generate the results

- **random** : Used for random sampling
- **timeit** : Used for calculate execution time
- **math** : Used for mathematical calculation
- **Tkinter** : Used for visualization of clusters

4 Distance Metrics and Similarities

A *distance metric or function* is a function that defines a distance between each pair of element of a set.Next, the purpose of a measure of *similarity* is to compare two lists of numbers (i.e. vectors), and compute a single number which evaluates their similarity. The most prominent distance metrics/Similarity metrics are Euclidean, Cosine and Jaccard which are explained further.
Euclidean Distance: Euclidean distance is a standard metric for geometrical problems. It is the ordinary distance between two points and can be easily measured with a ruler in two- or three-dimensional space.

Cosine Distance: The documents which are represented as term vectors, the similarity of two documents corresponds to the correlation between the vectors. This is quantified as the cosine of the angle between vectors, that is, the so-called cosine similarity.

Jaccard Distance: The Jaccard coefficient, which is sometimes referred to as the Tanimoto coefficient, measures similarity as the intersection divided by the union of the objects. For text document, the Jaccard coefficient compares the sum weight of shared terms to the sum weight of terms that are present in either of the two document but are not the shared terms.

To calculate the Cosine and Jaccard distance we need to subtract the respective similarity score from 1 every time we calculate the distance function.

5 Visualization

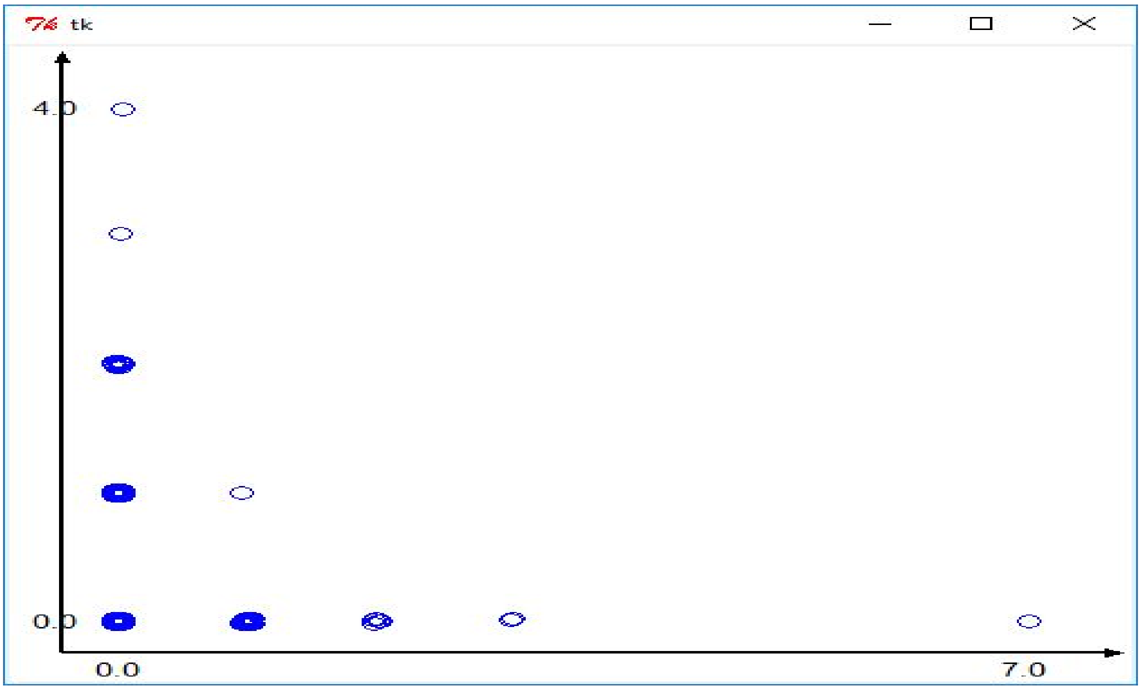

Figure 1: Plot of data points from the dataset)

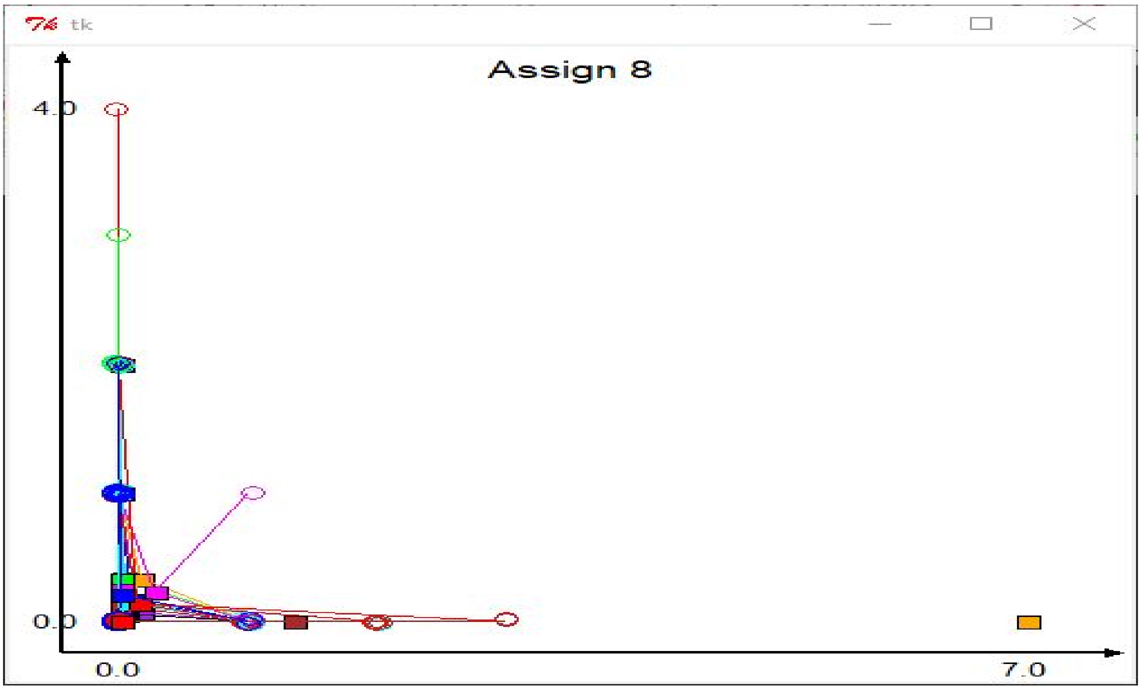

Figure 2: Snap of centroid assignment of Euclidean distance

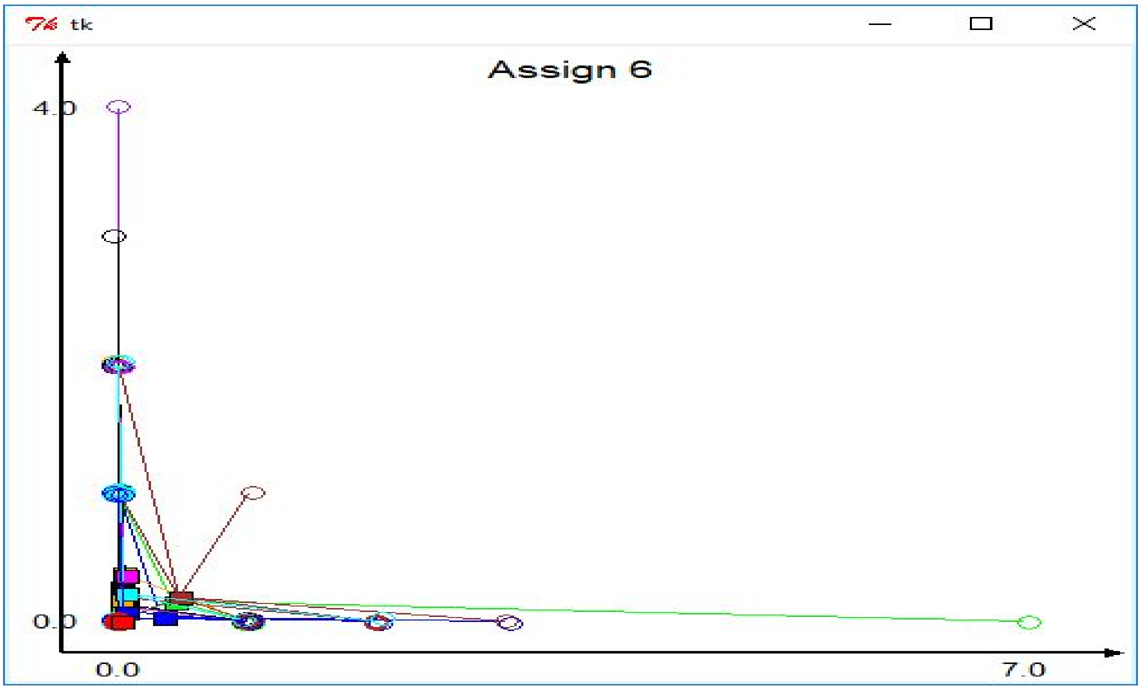

Figure 3: Cosine Distance Assignment Step No.6

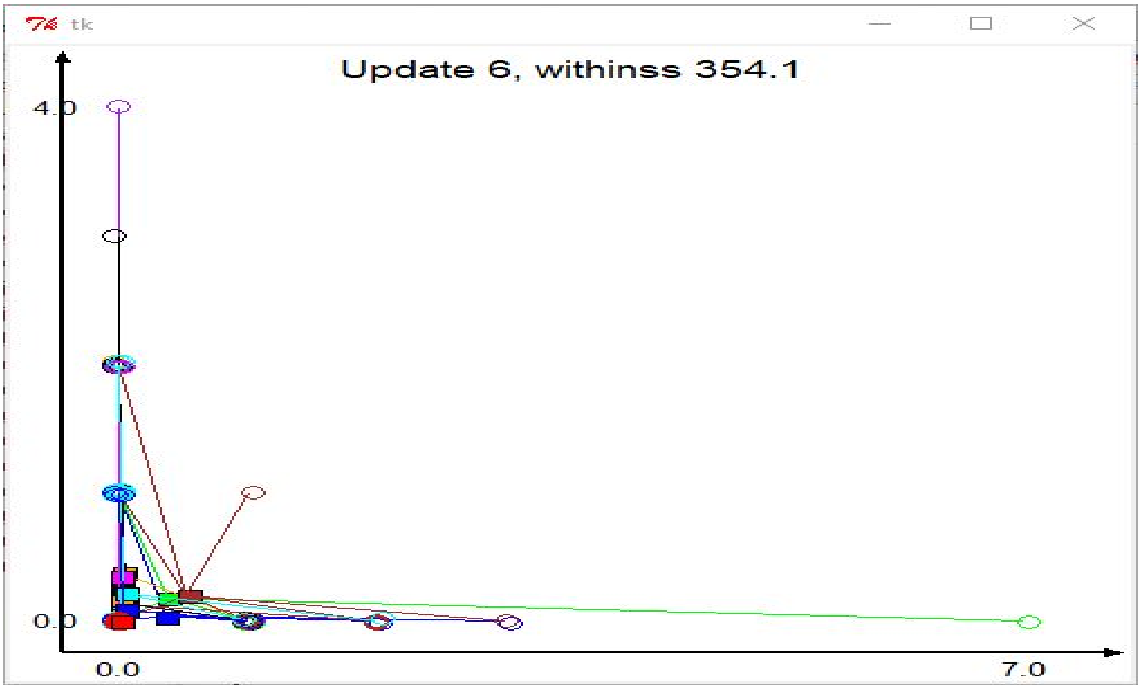

Figure 4: Cosine distance after updating centroid Step No.6

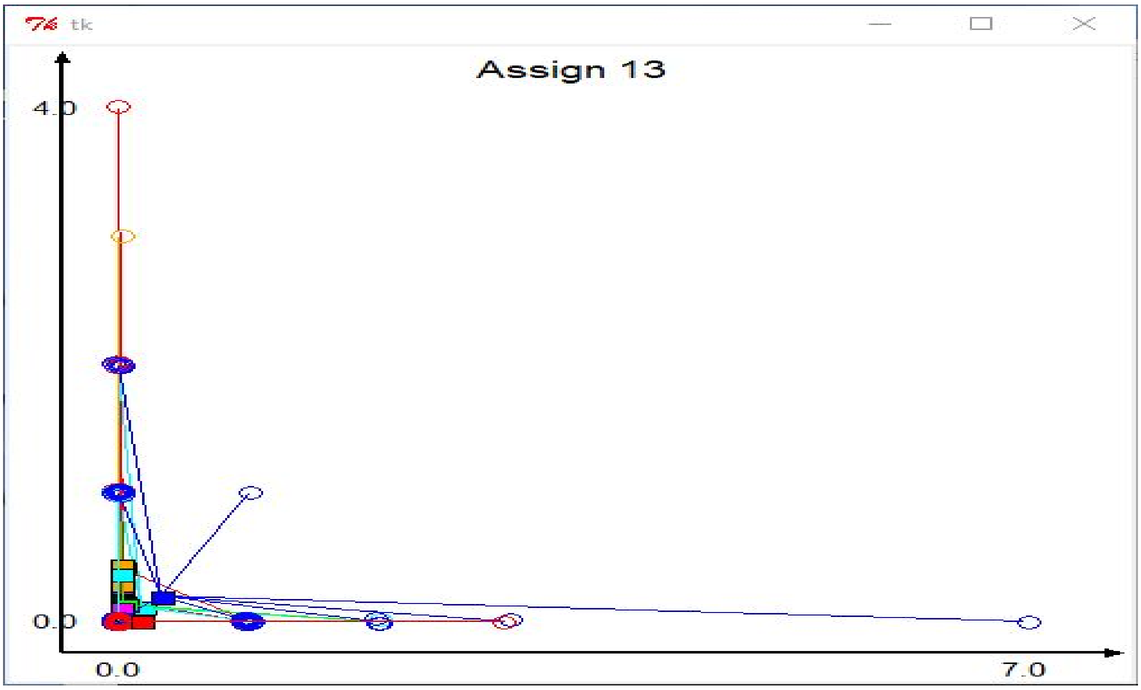

Figure 5: Jaccard Distance assignment

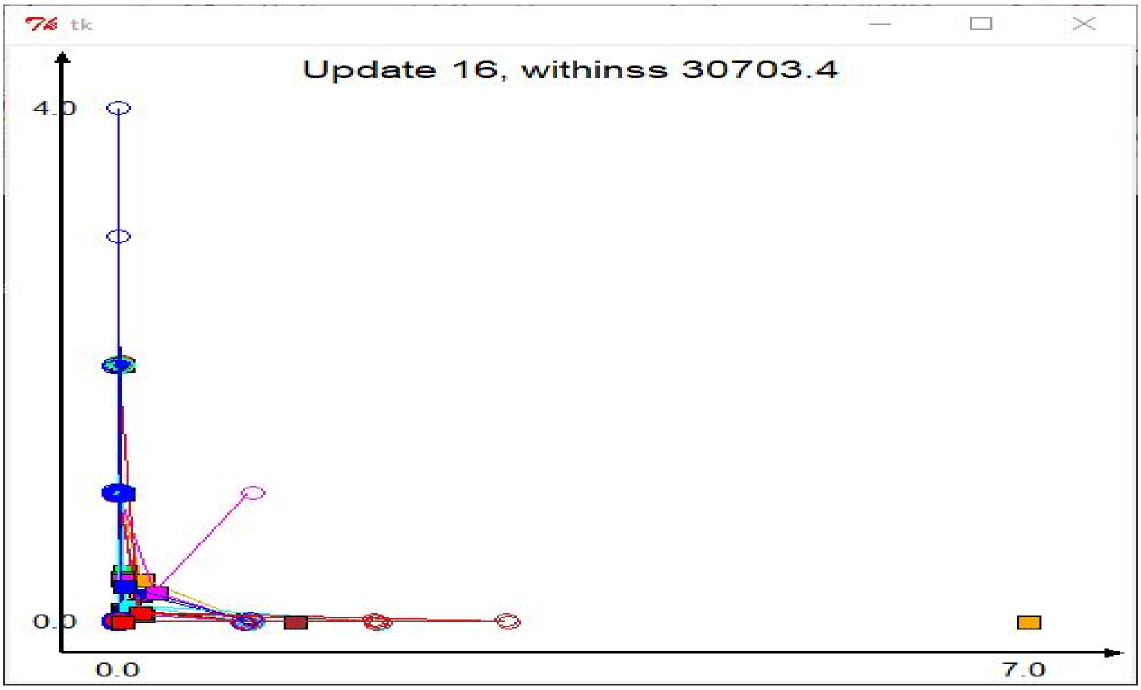

Figure 6: Final iteration of Euclidean Distance

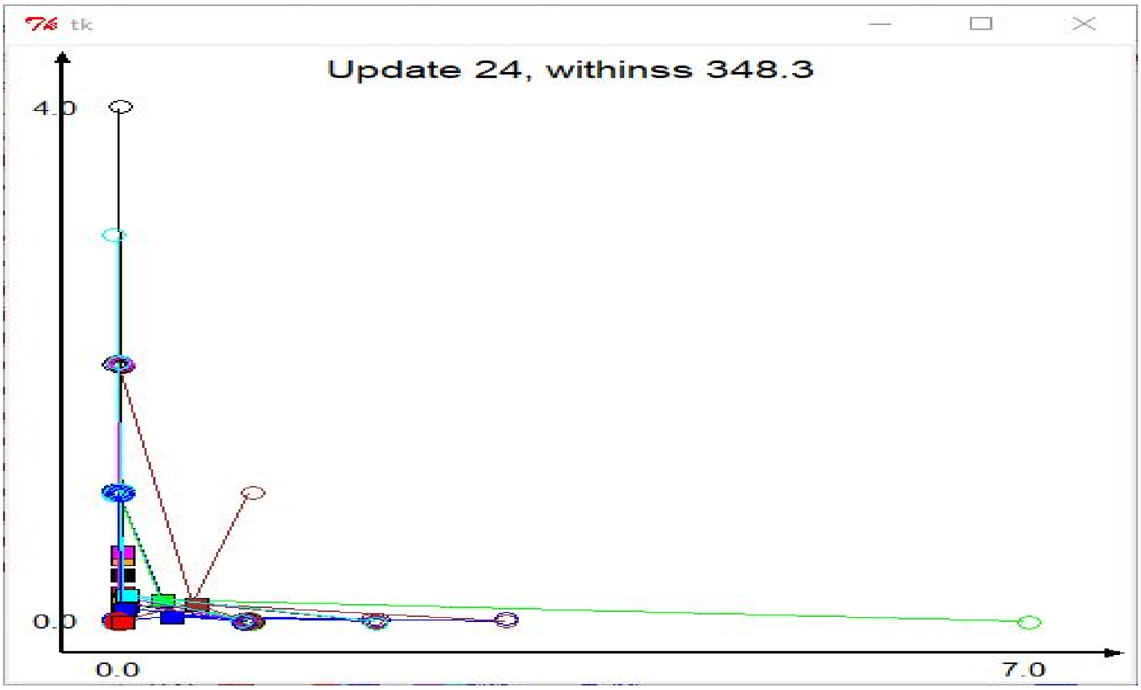

Figure 7: Final iteration of Cosine Distance

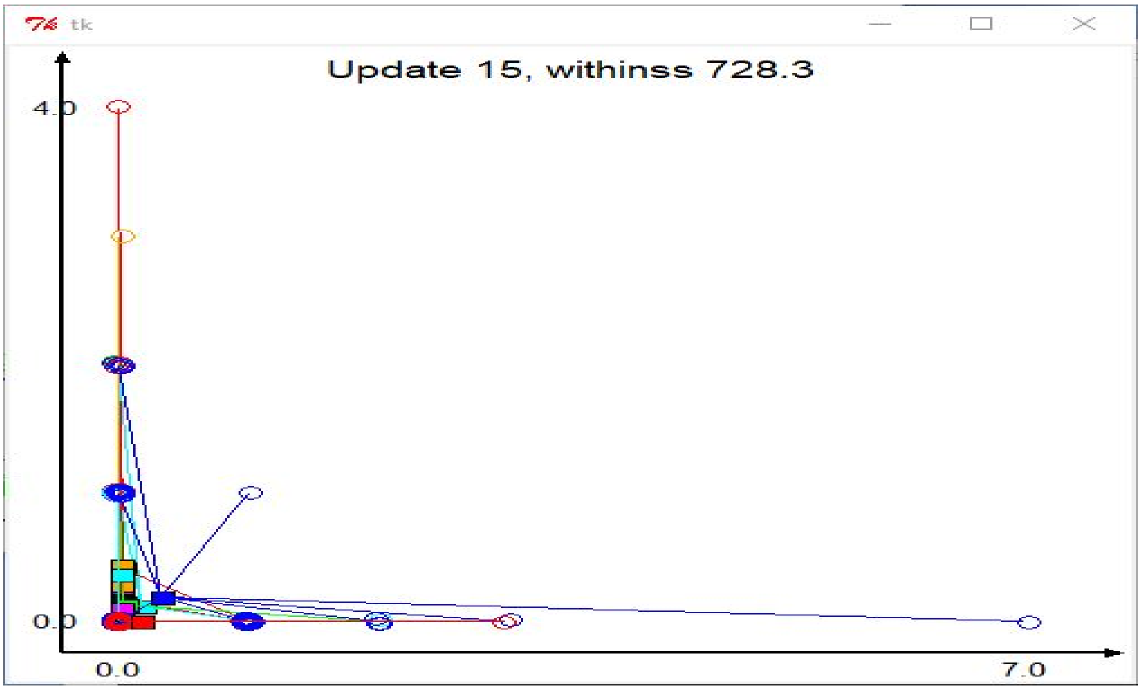

Figure 8: Final iteration of Jaccard Distance

The Figure 1 represents the point from the dataset which were plotted suing the "showDataset2D()" function. The blue circles are the data-points and we can see that they are cluttered mostly towards the left-bottom corner in the plot. From Figure 2 to Figure 5 are the plots of Euclidean, Cosine and Jaccard respectively. In the plots we can see that there are multiple square boxes, they represent the centroids of each cluster. From these figures of the past steps and the final step we can see how centroids have changed. The "withinss" in the plots is nothing but the Sum of Square Errors(SSE) produced by the respective metrics.

6 Experimental Review

6.1 Comparison of Metrics

The program was run multiple times for different conditions as specified in the Assignment. Firstly, we had to perform K-means for using all the metrics/similarities using the feature selected data. Secondly, we we had to do it for the complete data set. So we could see that in most of the cases SSE of the Cosine was the least and Jaccard following it and Euclidean had the highest in terms of distance function. But where as when it comes to similarity metric Euclidean had the least SSE and Cosine the Highest. As it was taking to long for these similarity metrics a terminating condition of "iteration <=500" was used.

6.2 Cluster Distribution

Another consideration I made was that I used random 1000 documents and performed all the matrix on the same dataset at one so that comparison would make some sense. Output of the cluster distribution was taken and we could see make two conclusions that distance metric use lesser iterations and lesser computational time and give proper clusters of dataset, where as, the similarity metrics take comparatively more time and more number of iterations to complete.

6.3 Accuracy

The data has been cleansed and proper features were extracted and their corresponding values were normalized to give accurate results.Even though this is unsupervised learning algorithm we can find accuracy by labeling the categories, which was done by considering the corresponding document number. To compute accuracy the labels were cross validated. Cosine was the best due to low SSE value, formed clusters which predicted accurate results. Euclidean took many assignments to settle down to the final clusters. Jaccard had most different results mainly due to the input data which was

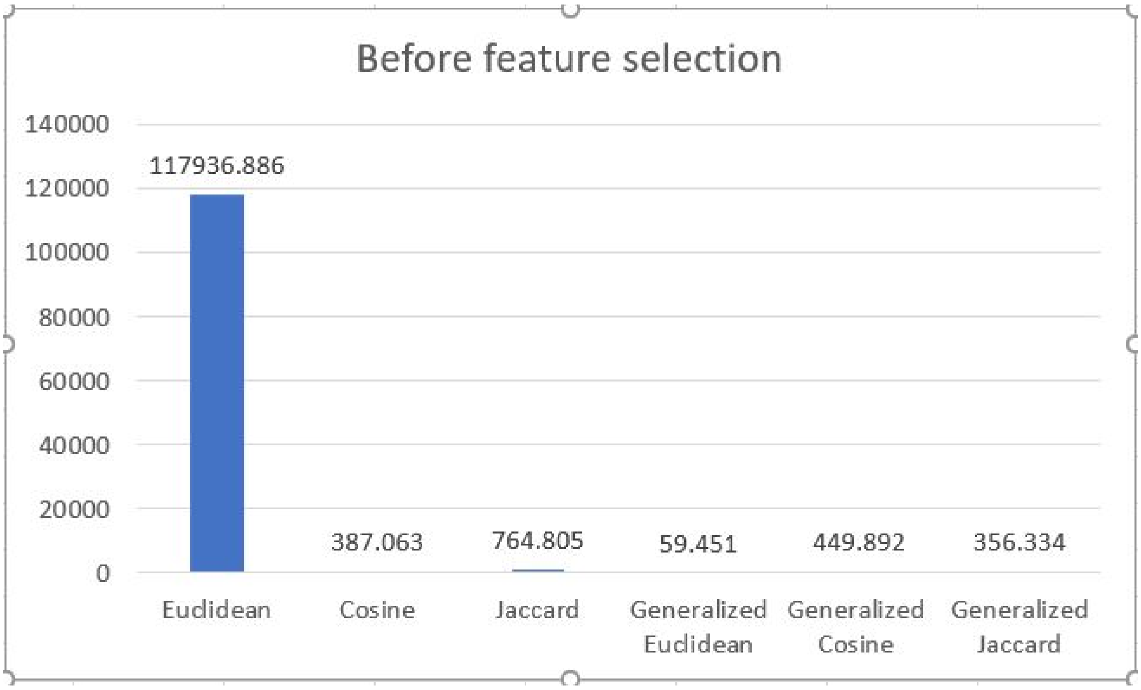

Figure 9: SSE before Feature selection

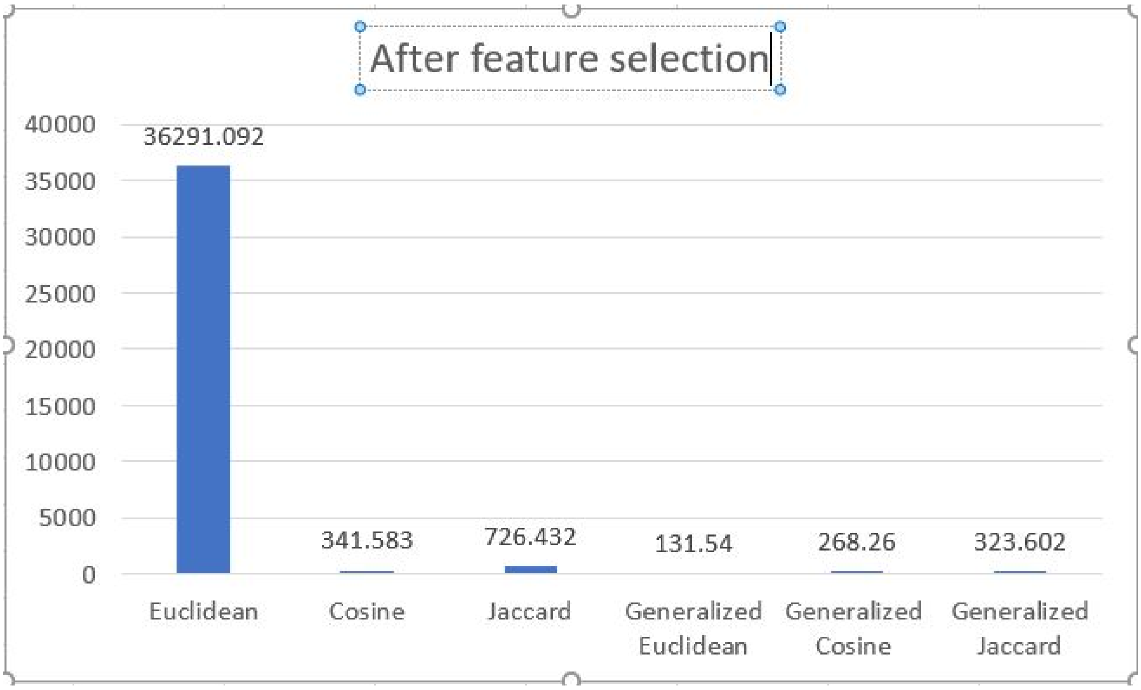

Figure 10: SSE after Feature selection

an existence matrix rather than a frequency matrix like the earlier versions. All in all, K-means is one of the best clustering algorithm mainly because it's very easy to understand with just a short algorithm.

6.4 Conditional K-means(Step 9)

The last step of assignment stated three conditions which were carried out. For this experiment on of the condition was "iteration<=100" the complete dataset(i.e 18828 documents) were considered and we can see that cosine distance was the best with the least SSE, even though it took more number of iterations it shows great results. We can also see that the condition "when SSE value increases in next iteration" was fulfilled as it stopped at early conditions for the similarity Metric even if the cluster was oddly distributed.

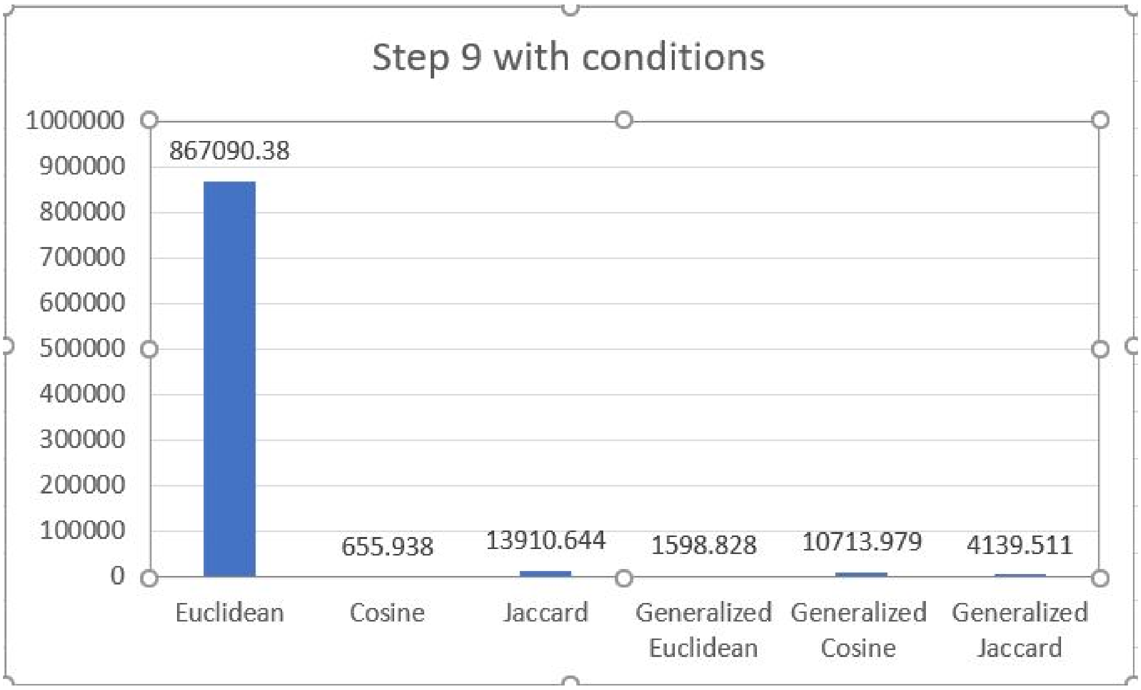

Figure 11: Conditional K-means

NOTE:

- *Please note the Generalized euclidean, cosine and Jaccard in the document mean Similarity metrics.*
- *In the .zip there is a output folder that contains the output of time of execution,iteration, and cluster data accross 3 files before_feature.txt, after_feature.txt, and step_9_hw.txt. You could have a look at it for more results.*

7 Conclusion

To conclude I would say that cosine distance was the best metric that I found for the current dataset purely considering the SSE values and the cluster distribution. It would be more easier to plot and produce better visualization if the dataset was small and randomization would not be needed.