Report (Assignment 2) – CMPT 459

Navjot Kaur (301404765)

Task 1:

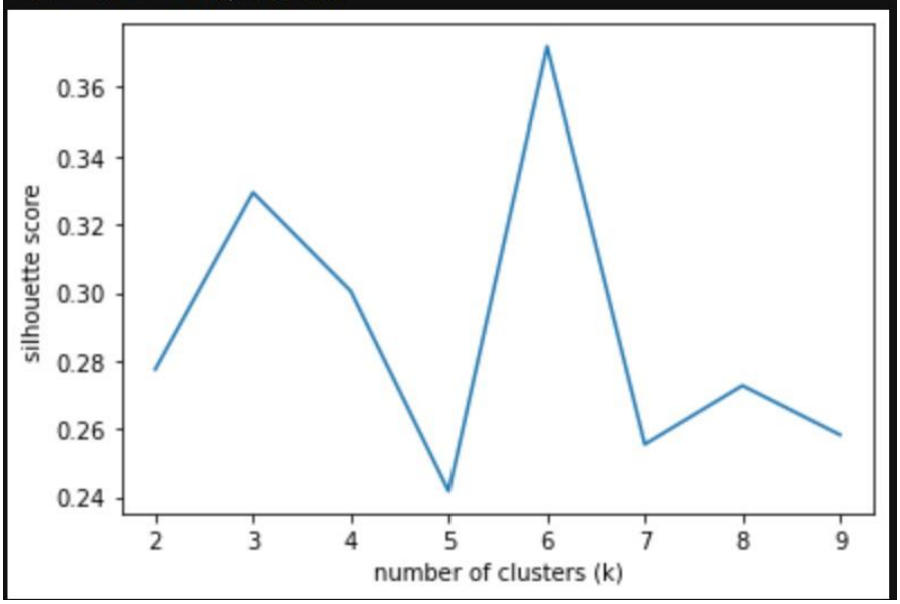
Task 1 is completed in the kmeans.py file where fit, initialize_centroids, update_centroids, euclidean_distance, and silhouette functions are implemented.

An additional method make_cluster is also implemented which calculates the distance for every point from their cluster and returns the appropriate clustering 1d array.

Task 2:

After the dimensionality is reduced to 100 by PCA, main.py is executed using random initialization, and clusters are produced for k = 2 to 9. The silhouette scores for k=2 to 9 and the plot for silhouette coefficient vs k is given below:

```
Silhouette coefficient:
k =  2 :  0.27745193
k =  3 :  0.32940596
k =  4 :  0.30050108
k =  5 :  0.2417185
k =  6 :  0.3720329
k =  7 :  0.25541973
k =  8 :  0.27266827
k =  9 :  0.25820738
```



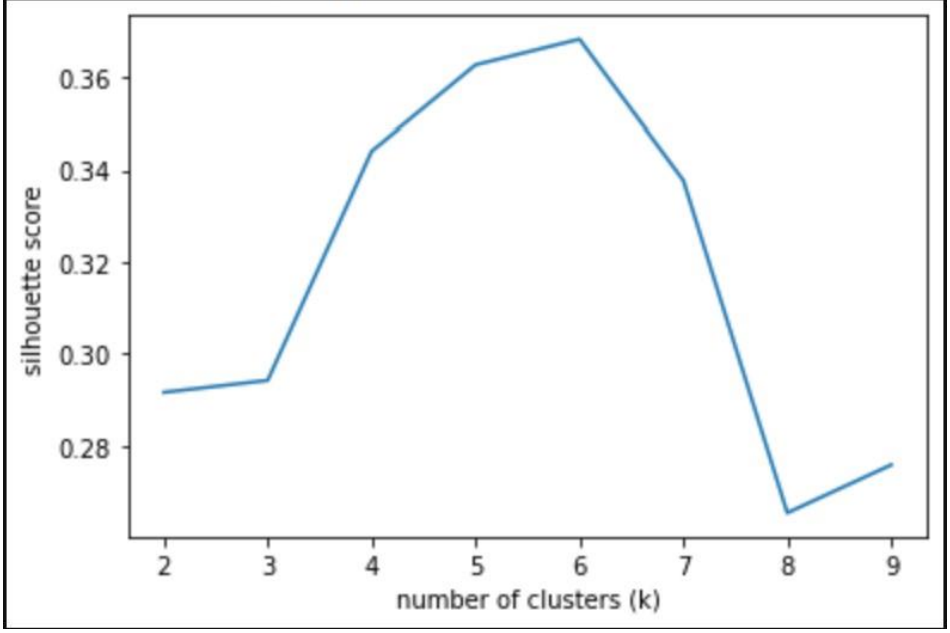
Best k is determined by the highest silhouette score i.e., the score that is closer to 1 for some k. Here the silhouette score for k = 6 is 0.3720329 which is the highest, it is chosen as the best k.

Best k = 6

Task 3:

Now, main.py is executed using kmeans++ initialization and clusters are produced for k = 2 to 9. The silhouette scores for k=2 to 9 and the plot for silhouette coefficient vs k is given below:

```
Silhouette coefficient:
k =  2 :  0.2916432
k =  3 :  0.29425588
k =  4 :  0.3441319
k =  5 :  0.3627392
k =  6 :  0.3683269
k =  7 :  0.33780012
k =  8 :  0.2653924
k =  9 :  0.27584928
```



Best k = 6

The best k for both initializations come out to 6, however, it can be concluded that with random initialization the silhouette score fluctuates a lot whereas, with the kmeans++ initialization, the silhouette score remains quite constant between the values of 4 to 7, and then dropped significantly to 8. Every time the code is run different the value of k changes as well. Since random initialization randomly assigns the centroids and sometimes the centroids are too close to each other that clustering is not that good. But with kmeans++ the first centroid is chosen randomly, and others are chosen such that the probability of choosing the data point as a centroid that is far away from the previously founded centroids is more than other data points and this way the clustering is done which is reasonable and clusters are formed in nicely and organized manner. Hence, it can be seen that kmeans++ initialization is better than random initialization because of consistent silhouette scores for kmeans++ when compared to fluctuating silhouette scores for random initialization.

Task 4:

Scatter plot for best k in task 2 which is k = 6 with random initialization: