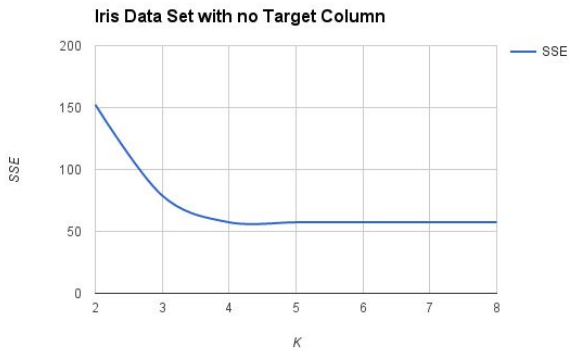
Isai Mercado Oliveros
April 1, 2016
Clustering Lab

Report your exact results on sponge data set with k=4 clusters. For k-means use the first 4 elements of the data set as initial centroids.

Cluster 1	Cluster 2	Cluster 3	Cluster 4
point 0	point 1	point 2	point 6
point 13	point 7	point 3	point 15
point 14	point 8	point 4	point 17
point 16	point 9	point 5	point 18
point 20	point 10		point 19
point 22	point 11		point 21
point 25	point 12		point 23
point 26	point 33		point 24
point 27	point 34		point 28
point 29	point 35		point 32
point 30	point 37		point 36
point 31	point 38		point 47
point 39	point 42		point 71
point 40	point 43		
point 41	point 44		
point 46	point 45		
point 48	point 50		
point 49	point 51		
point 53	point 52		
point 60	point 54		
point 61	point 55		
point 63	point 56		
point 64	point 57		
point 70	point 58		
point 72	point 59		
point 73	point 62		
point 74	point 65		
point 75	point 66		
	point 67		
	point 68		
	point 69		

Run your Clustering Algorithm on Iris data with no target column. Run it for k = 2-7. State whether you normalize or not (your choice). Graph the total SSE for each k and discuss your results

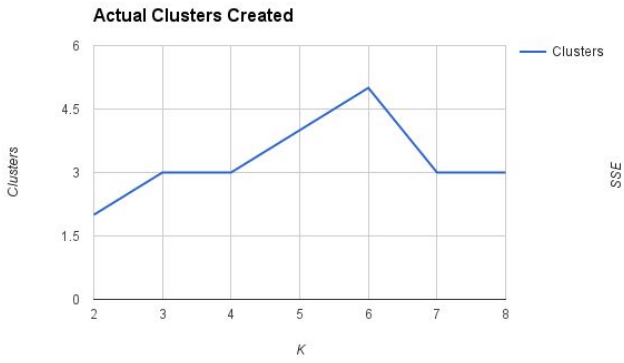
K	SSE
2	152.3687
3	78.9450
4	57.3454
5	57.3550
6	57.4732
7	57.4732
8	57.4732

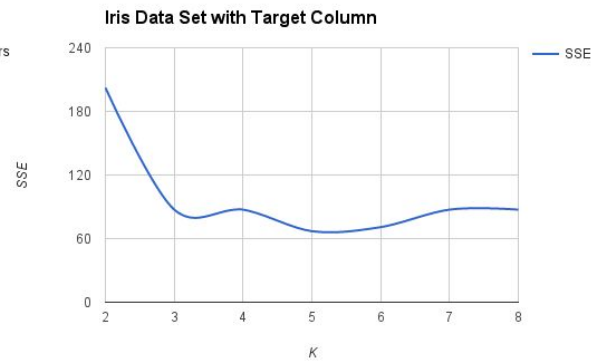


I did not normalize my data. From this graph I can see that as K increases, the SSE value levels out. I think this happens because even if K is increasing, the number of clusters found by K Means stops at certain point, and some clusters end up being empty, so the SSE value is calculated from non empty clusters whose quantity, as mentioned before, gets stabilized at certain K.

Now do it again where you include the output class as one of the input features and discuss your results and any differences.

K	SSE
2	202.368
3	87.3296
4	87.3296
5	66.9785
6	70.7993
7	87.3296
8	87.3296

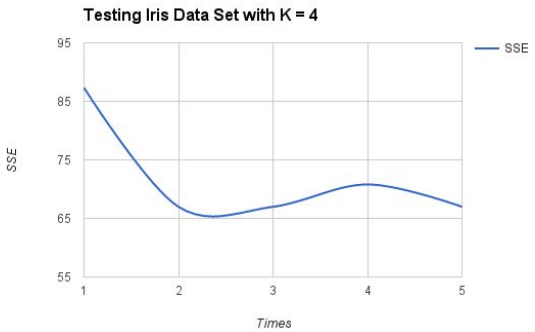




Now that the output class is included, it is easier for KMeans to find the actual 3 clusters. An interesting fact that I observed while increasing K is that even though K got larger and larger sometimes KMeans only found 3 clusters and the rest of the clusters where empty. So in this experiment, the SSE leveled out at 3 clusters.

Run k-means 5 times with k=4, each time with different initial random centroids and discuss any variations in the results.

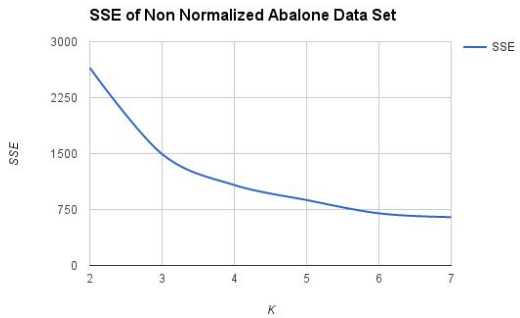
Times	SSE	Clusters
1	87.3296	3
2	66.9785	4
3	66.9785	4
4	70.7993	4
5	66.9785	4



As observed before, the SSE leveled out between 3 and 4 clusters. Sometimes, it returned 3 clusters because the Iris data set has 3 real clusters, but at the same time, since KMeans was given 4 as the K, it was trying to force points to cluster into 4 clusters

Run your variation abalone data set. Treat "rings" as a continuous variable. Why would I suggest that? Run it for k = 2-7. Graph your results without normalization.

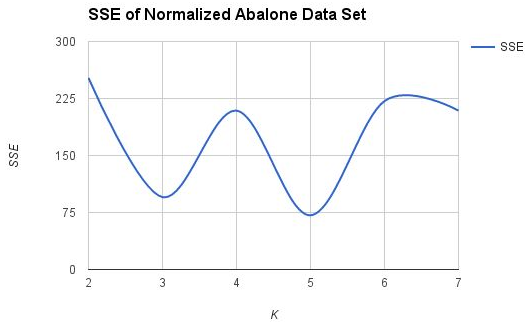
K	SSE
2	2650.3459
3	1493.5063
4	1077.8740
5	876.8568
6	698.7114
7	647.920



It is better to use rings as a continuous value because since it represents years, it is better to calculate their distance by euclidean distance rather than 0 or 1. In addition from this graph, I can see that without normalization, the increment in K makes the SSE go down. I think that this happens because more clusters cause the inner cluster-points distance to decrease, so the overall SSE is smaller.

Run it again with normalization.

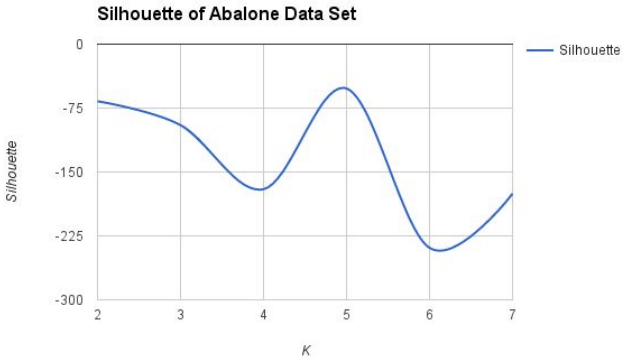
K	SSE
2	252.118
3	95.2109
4	208.9906
5	71.3128
6	221.6001
7	208.9906



This graph is completely different from the graph without normalization. I think that SSE is more stable with normalization since all dimensions pull with equal magnitudes. Thus, the SSE is more perceptible to changes in distances. In this case, I think that the data has 5 clusters since K = 5 has the smallest SSE which means that the clusters are more compact.

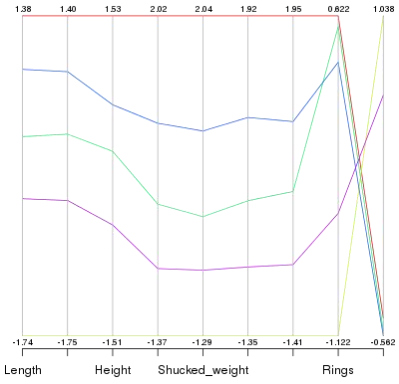
For your normalized abalone experiments, calculate and graph a performance metric for the clusterings for k = 2-7. Use the Silhouette. Discuss how effective the metric you used might be in these cases for selecting which number of clusters is best.

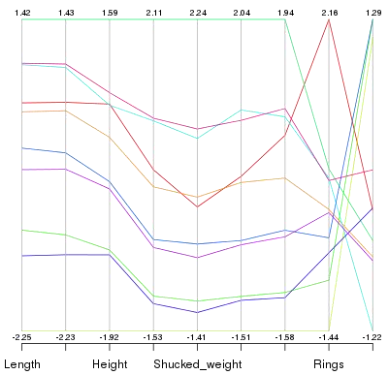
K	SSE	Silhouette
2	282.0261	-67.0026
3	221.6001	-94.9887
4	240.6865	-170.2089
5	71.3094	-52.0247
6	58.6403	-239.0877
7	49.6857	-175.4321



The results of the silhouette function make me think that the abalone data set must have 5 clusters because K = 5 has the greatest silhouette of all Ks. This means that the overall cluster-inner-distance is smaller than the overall cluster-outer-distance which means that points are better pertaining to their current cluster rather than a neighboring cluster. Therefore, K = 5 has better quality clusters which makes me think that the abalone data set might have 5 groupings.

Do an experiment of your own regarding clustering.





For my experiment I wanted to visualize the means in order to see if having 5 clusters for the abalone data set made sense. I used R to make two parallel coodinate plots for kmeans with K = 5, and K = 10. My goal was to see if 10 means would kind of cluster into 5 means... And it looks like the 10 clusters could actually be merged into 5 clusters. Thus, my hypothesis of the abalone data set having 5 clusters could be correct.