SPANISH Sponge Set (not normalized, not shuffled)

Below you can see some of the tables with the corresponding output that I ran on the sponge set for k-means, complete Link and Single Link.

K-means

Cluster	Centroid Values	Data Per Cluster		SSE
0	1_CAPA, SIN_CAPA_INTERNA_DEL_CORTEX, SI, 2.444, SIN_TILOSTILOS_ADICIONALES, SIN_ESPICULA_PRINCIPAL_ESTILO, 2.611111111111111, 0.444, OTROS	18		61.167
1	SIN_CORTEX, SIN_CAPA_INTERNA_DEL_CORTEX, NO, 0.0, SIN_TILOSTILOS_ADICIONALES, SIN_ESPICULA_PRINCIPAL_ESTILO, 2.0, 0.0, OTROS	29		42
2	SIN_CORTEX, SIN_CAPA_INTERNA_DEL_CORTEX, NO, 0.0, SIN_TILOSTILOS_ADICIONALES, SIN_ESPICULA_PRINCIPAL_ESTILO, 0.5, 1.25, OTROS	4		5.75
3	2_CAPAS, TANGENCIAL, SI, 3.04, INTERMEDIARIOS, SIN_ESPICULA_PRINCIPAL_ESTILO, 2.52, 2.48, OTROS	25		95.44
			Grand Total	204.357

Complete Link

Cluster	Centroid Values	Data Per Cluster		SSE
0	2_CAPAS, TANGENCIAL, SI, 2.536, INTERMEDIARIOS, SIN_ESPICULA_PRINCIPAL_ESTILO, 2.5, 1.179, OTROS	28		119.071
1	SIN_CORTEX, SIN_CAPA_INTERNA_DEL_CORTEX, NO, 0.0, SIN_TILOSTILOS_ADICIONALES, SIN_ESPICULA_PRINCIPAL_ESTILO, 0.867, 0.333, OTROS	15		18.0667
2	2_CAPAS, TANGENCIAL, SI, 3.615, INTERMEDIARIOS_Y_ECTOSOMICOS, SIN_ESPICULA_PRINCIPAL_ESTILO, 2.615, 2.846, OTROS	13		51.846
3	SIN_CORTEX, SIN_CAPA_INTERNIA_DEL_CORTEX, NO, 0.1, SIN_TILOSTILOS_ADICIONALES, SIN_ESPICULA_PRINCIPAL_ESTILO, 2.65 ,0.0, AMARILLO_PALIDO	20		24.35
			Grand Total	213.223

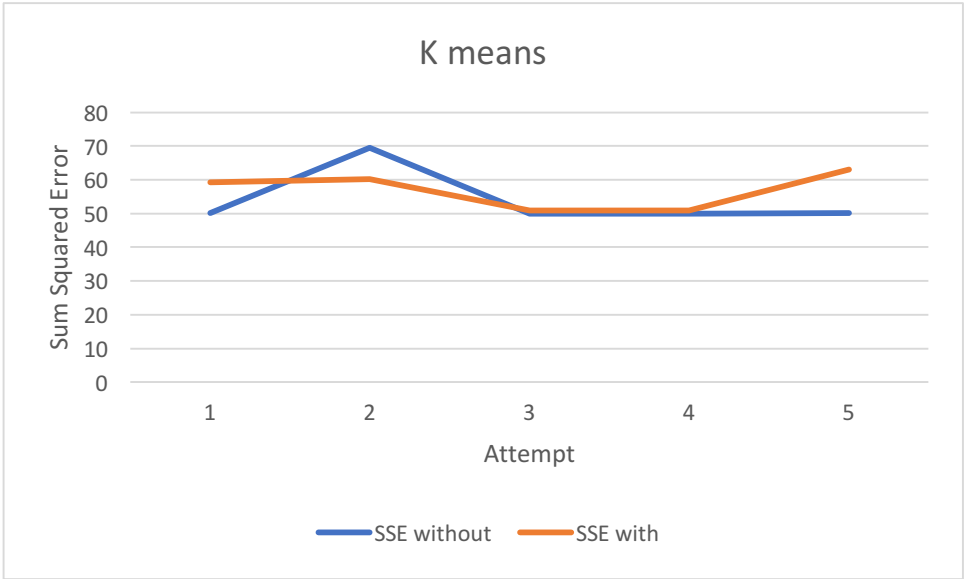
Single Link

Cluster	Centroid Values	Data Per Cluster	SSE	
0	SIN_CORTEX, SIN_CAPA_INTERNA_DEL_CORTEX, SI , 1.521, SIN_TILOSTILOS_ADICIONALES, SIN_ESPICULA_PRINCIPAL_ESTILO, 2.26027397260274, 0.9315068493150684, OTROS	73		511.932
1	1_CAPA, SIN_CAPA_INTERNA_DEL_CORTEX, SI, 4.0, SIN_TILOSTILOS_ADICIONALES, SIN_ESPICULA_PRINCIPAL_ESTILO ,1.0, 3.0, OTROS	1		0
2	3_CAPAS, TANGENCIAL, SI, 3.0, ECTOSOMICOS_DISPERSOS, FUSIFORME, 3.0, 4.0, ?	1		1
3	3_CAPAS, TANGENCIAL, SI , 2.0, SIN_TILOSTILOS_ADICIONALES, SIN_ESPICULA_PRINCIPAL_ESTILO, 1.0, 0.0, ?	1		1
			Grand Total	513.932

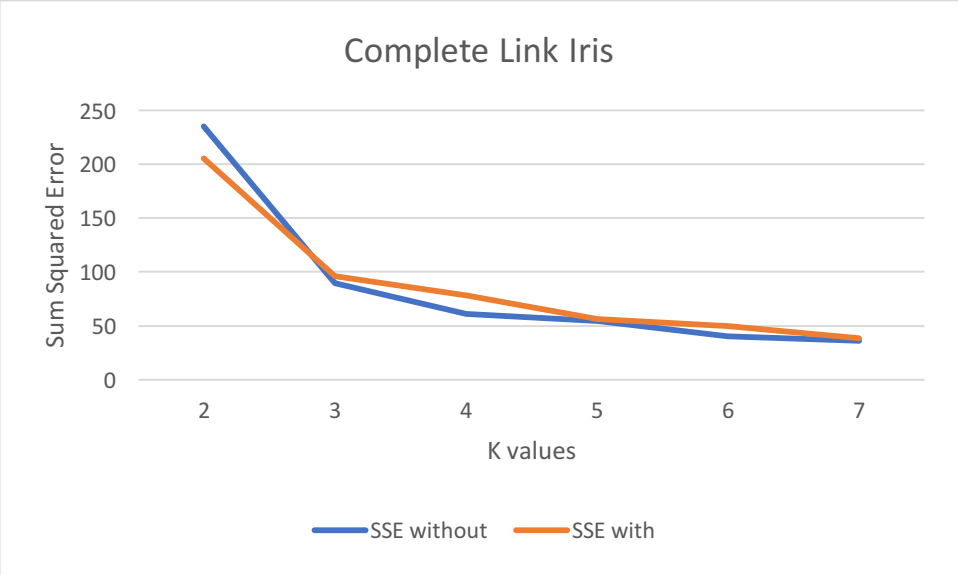
Iris Data

So as you can see in the results below, it seems that the complete link method seemed to work just about the same whether we include the output or not, while the single link seemed to work better with the last output included. I assume then that the last data set has the effect of making the distances more accurate and precise for complete single link distances. For both of them it also appears that making more clusters lowers the SSE because the clusters are able to be better divided. As for the K means it appears that the values fluctuate depending on which initial centroid is chosen. This makes sense because if poorly chosen centroids are selected, then it can be harder to split the data into the corresponding centroids correctly. I didn't normalize the data. For kmeans it didn't seem to matter if the output was included or not. This would imply that the distance for kmeans isn't affected with the output value.

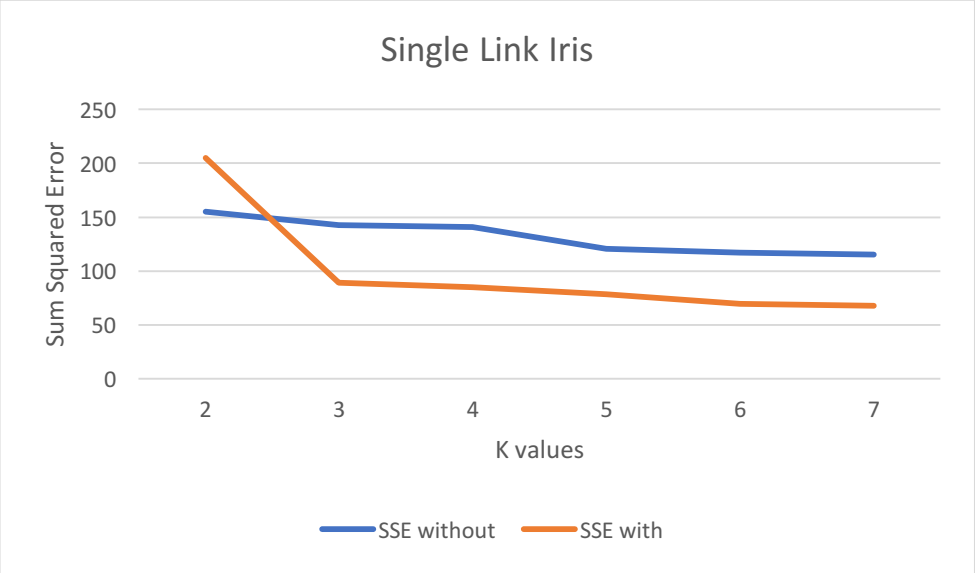
K-means



Complete Link



Single Link

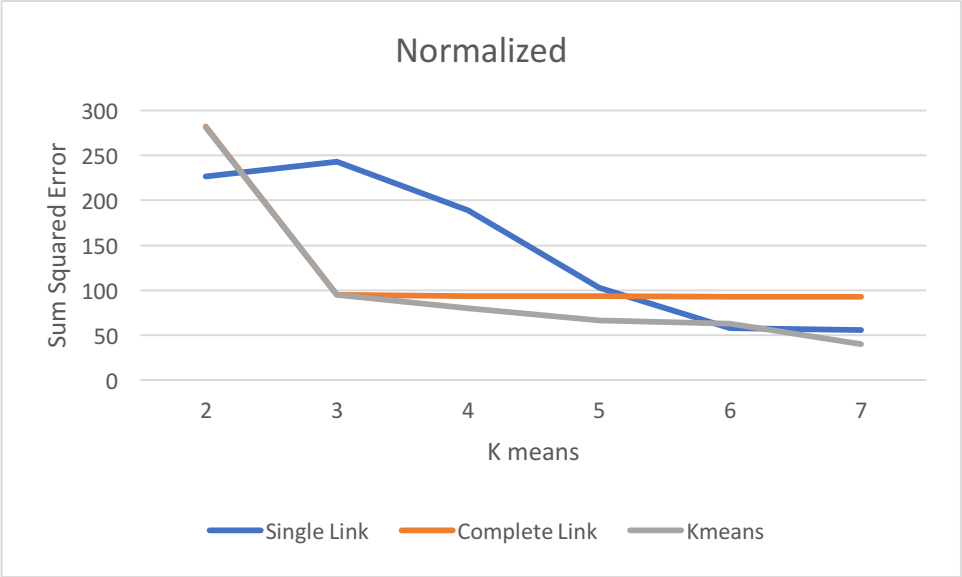


Abalone

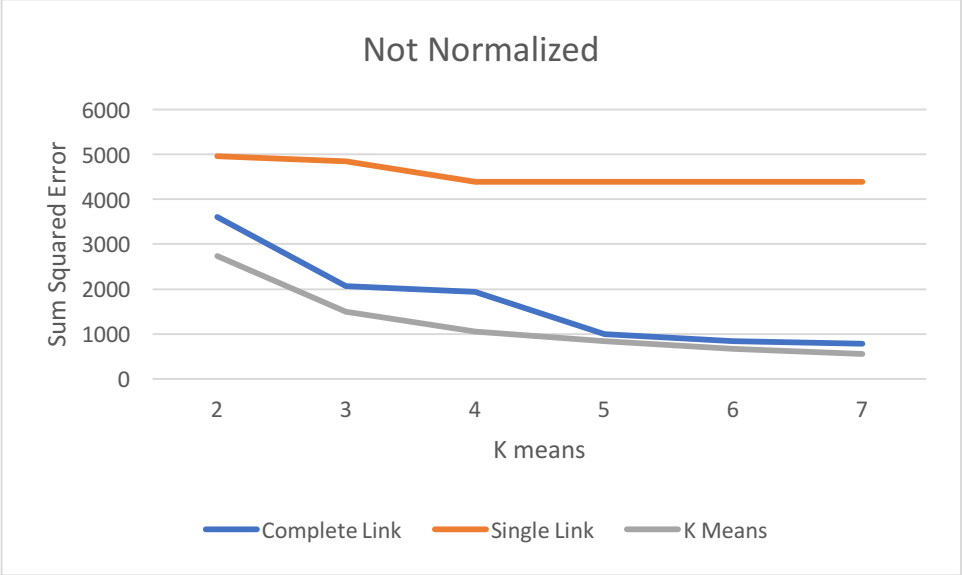
For the abalone data set we do indeed want to Normalize the data as that significantly lowers the SSE for every method that I ran. Also we want to use the rings as continuous variables (which is what I did), because that variable definitely has a numeric value and thus distance between the rings can be actually measured with a Euclidean distance and would have meaning

while using it categorically would detract some meaning and purpose from those values and would make it a lot more likely to not contribute to the distance score metric.

Normalized



Not Normalized



Abalone Silhouette Score

The abalone when run with the silhouette score definitely seemed to have some potential significance. It seems that the higher the k-values for all of my methods that I ran the higher the Silhouette score. It still though seemed to have quite some variance among the different clusters. This would make sense because some clusters are going to be closer to other clusters than others and the higher the score the more apart they are and the lower the score the closer

they are(or that data is incorrectly assigned). If split too much though, the values will start to go towards zero but as for this data set it seems that 7 is a good split.

K-means

K-Values	0	1	2	3	4	5	6
2	-0.3	-0.2					
3	-0.1	-0.3	-0.2				
4	0	-0.1	-0.3	0			
5	0.1	0.2	0.3	0.1	0.2		
6	0.1	0.2	0.5	0.3	0.4	0.9	
7	0.7	0.4	0.8	0.6	0.5	0.6	1

Complete Link

K-Values	0	1	2	3	4	5	6
2	-0.3	-0.6					
3	-0.4	-0.4	-0.2				
4	0	-0.1	-0.2	0.1			
5	-0.1	0	0.1	0.3	0.2		
6	0	0.3	0.2	0.3	0.4	0.6	
7	0.4	0.5	0.8	0.6	0.7	0.5	0.6

Single Link

K-Values	0	1	2	3	4	5	6
2	-0.7	-0.5					
3	-0.3	-0.1	-0.4				
4	0	-0.1	-0.2	0			
5	0	0.2	0.3	0.1	0.2		
6	0.2	0.1	0.5	0	0.4	0.9	
7	0.7	0.2	0.7	0.3	0.5	0.7	1

Experiment

For my experiment, I decided to look into various means of finding initial starting clusters for K means. I ran my experiment on the normalized abalone data set file. I ran on five various methods to approach this including running on randomized initial centroids (I took the average of running it 5 times), first k instances in the data set, last k instances in the data set, the k MOST similar instances, the k LEAST similar instances. I ran with k ranging from 2-7 and you can see the results in the chart below. My hypothesis was that Least Similar would be the most accurate because that way the data could be divided into clusters from the get go and would thus be more accurate and wouldn't have to keep gradually shifting clusters around to be made more accurate. Interestingly for this data set it appears that all of them run about the same accuracy with little variation between all the various methods that I tried and they all finished about the same number of cycles as well. Perhaps with k means in this data set, the clusters are able to be found regardless of the initial centroids because of the shear amount of information that eventually dilutes the initial impact of the initial centroids through averaging the data together.