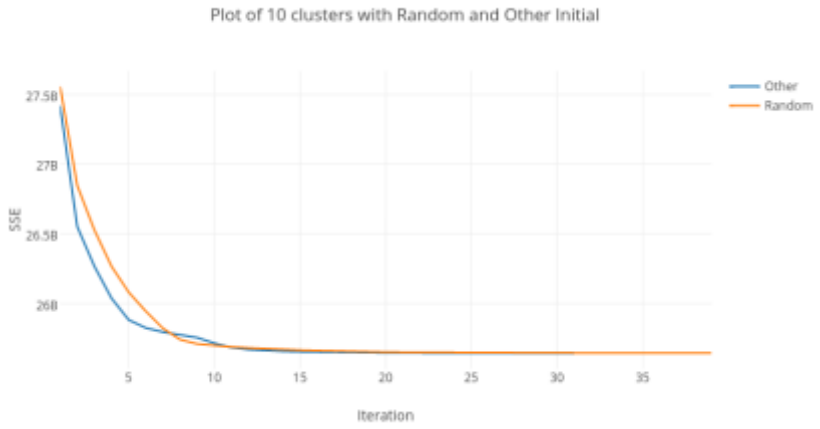
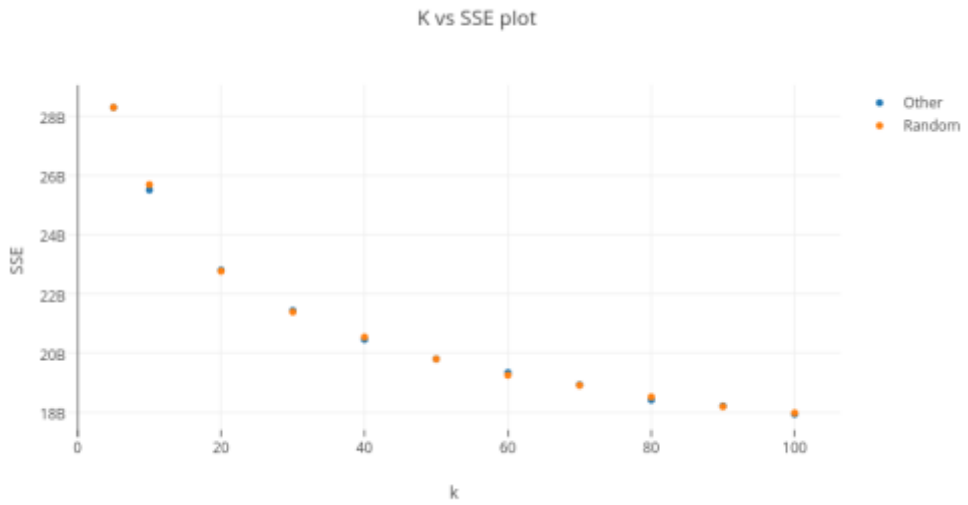
Sabastian Mugazambi
HMW 6 : Data Mining Spring 2017

1. Change in SSE



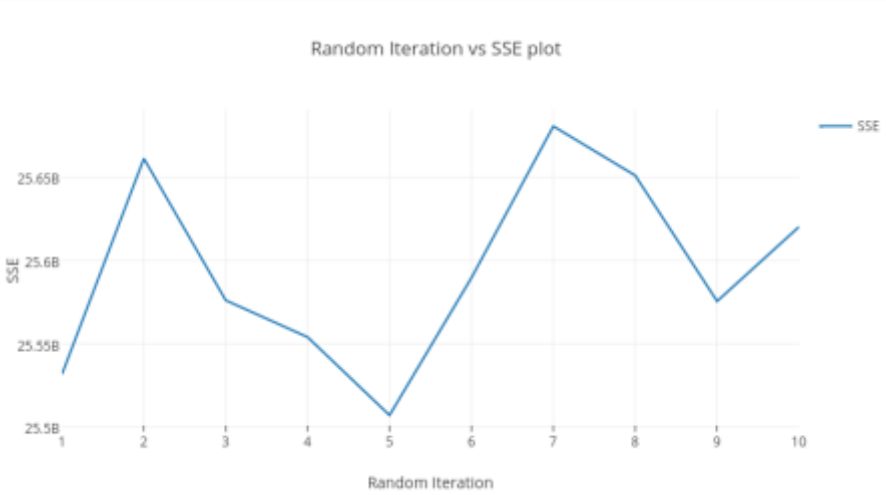
As seen from the graph the SSE is monotonically decreasing with each iteration with newly assigned centroids. Using the kmeans++ way of initializing centroid gives us a overall lower SSE for each iteration and it decreases faster than the random initialization.

2. Increasing K



K= 100 has the lowest SSE but the right K to use for this data set will be K=10 since we know there are 10 digits we want to cluster. So k=10 fits this dataset due to the 10 digits we have.

3. Random Initialized Iterations for K 10

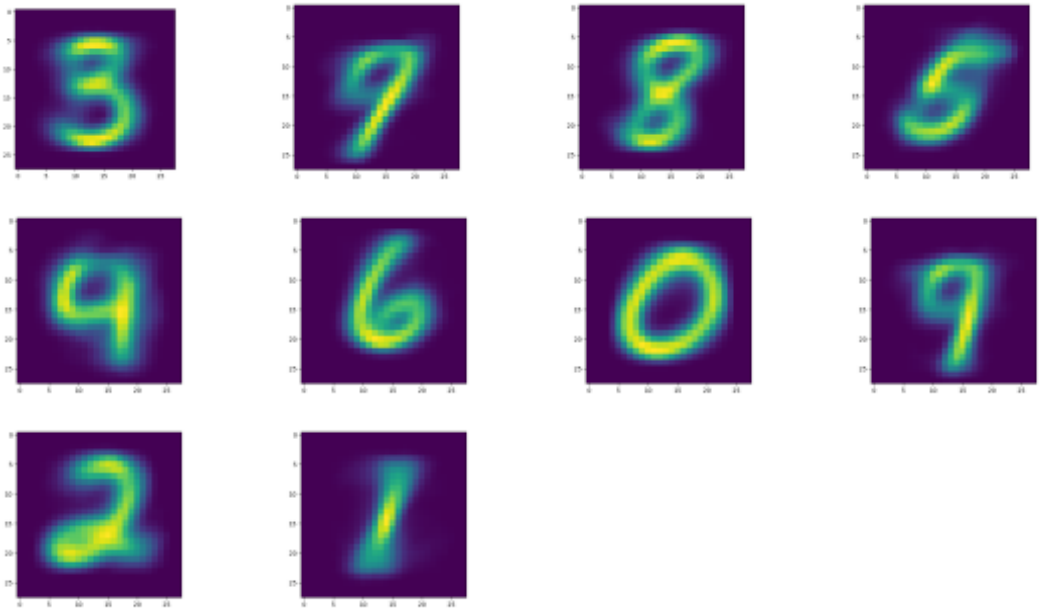


Random Iteration	SSE
1	25531702212
2	25661287757
3	25576084383
4	25553971107
5	25507040902
6	25589615487
7	25680687456
8	25651093297
9	25575570935
10	25620227692

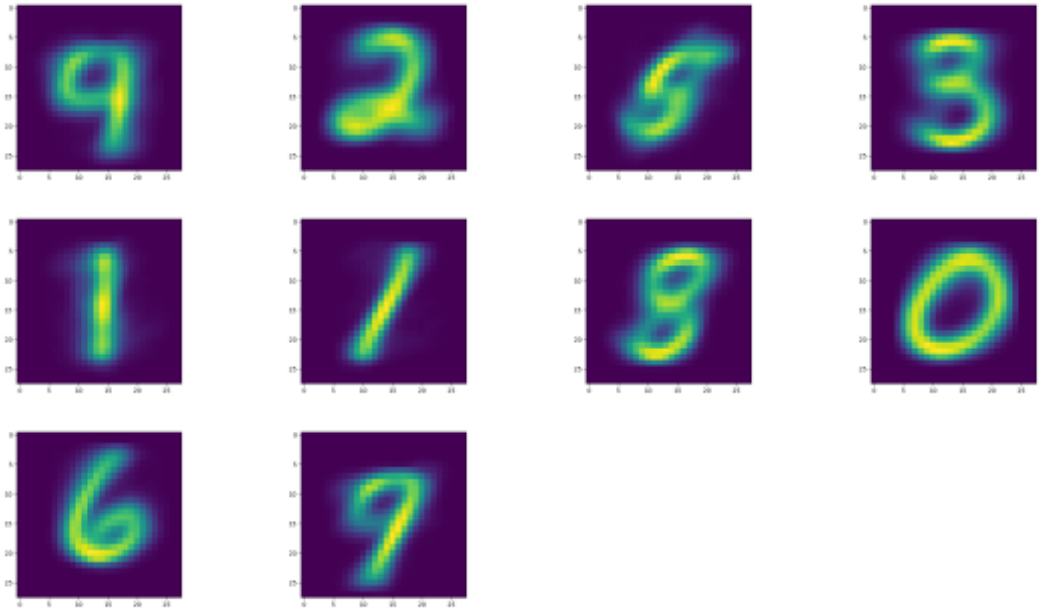
As seen in the graph above, there is no trend to the final SSE when we run 10 iterations randomly initializing centroids every time. To evaluate the degree of change one could possibly look at the SSE and see when it is lower and check the distances between the randomly chosen centroids to see how spread out these random centroids are.

4. Images of Centroids

K=10 Other 1



K=10 Random



In general, the digits look like what I would expect except that 1 or 2 digits seem to be miss represented. The ideal situation would be to see centroids from each digit. The two centroids are different as we can see from 1 iteration above of each initialization method. For “Other” /Kmeans++ we have every digit represented by a centroid although one of the digits (7) is confusing and kind of looks like a 9 but in general kmeans++ performs better than Random which has 2 sets of repeated digits in the centroids.

5. Running With Cheat

When run with “Cheat” it is clear that the iterations converge faster as the centroids are chosen with the help of the trues data.

When the images of the final centroids at convergence are saved,(below), we can see that each digit is represented.