I actually implemented the Kmeans algorithm to help me with question 1 on the homework because no sane person would carry out 4 iterations of this algorithm by hand when they have access to programming. I created both a binary Kmeans and 3 class Kmeans algorithm, and animated them so I could see the centroids of each cluster changing as I iterate through.. As a result, it was relatively straightforward to convert my custom 3 class Kmeans code to the format required in this assignment. The only thing I had to change was creating a new list called “classes”, and appending a 0,1,or 2 to it depending on which class the data point was in.

Unfortunately, I would have liked to have created a more general algorithm to accept different sized classes but I didn't feel like messing around with the shapes and sizes of all the arrays.

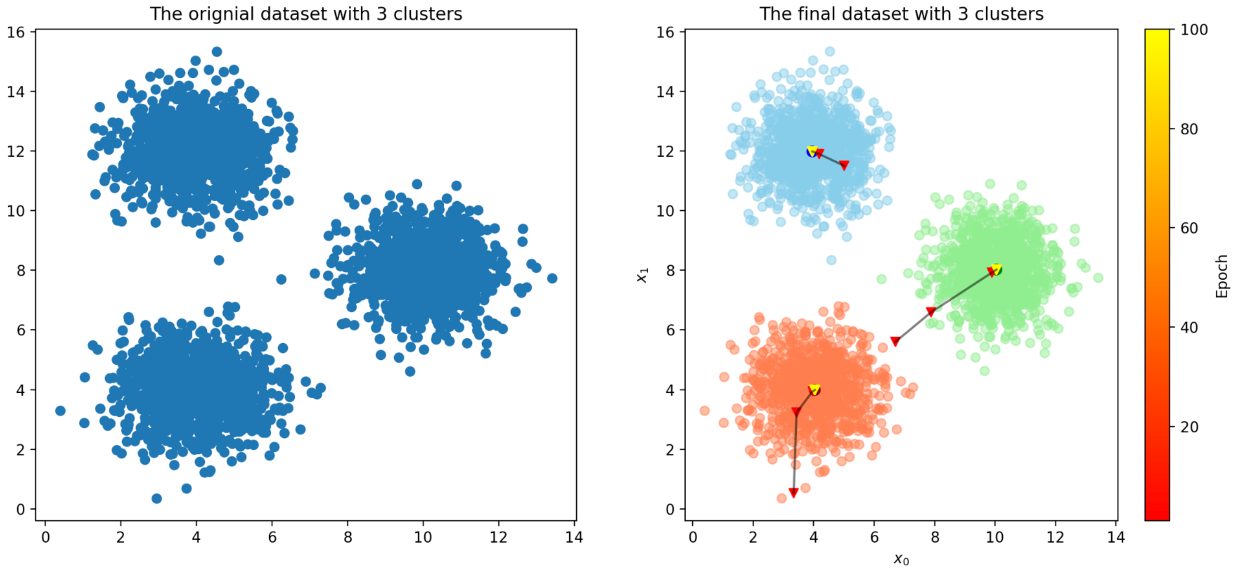

Figure 3: Kmeans_example.py updated to include the Kmeans class algorithm.

See below for my versions….

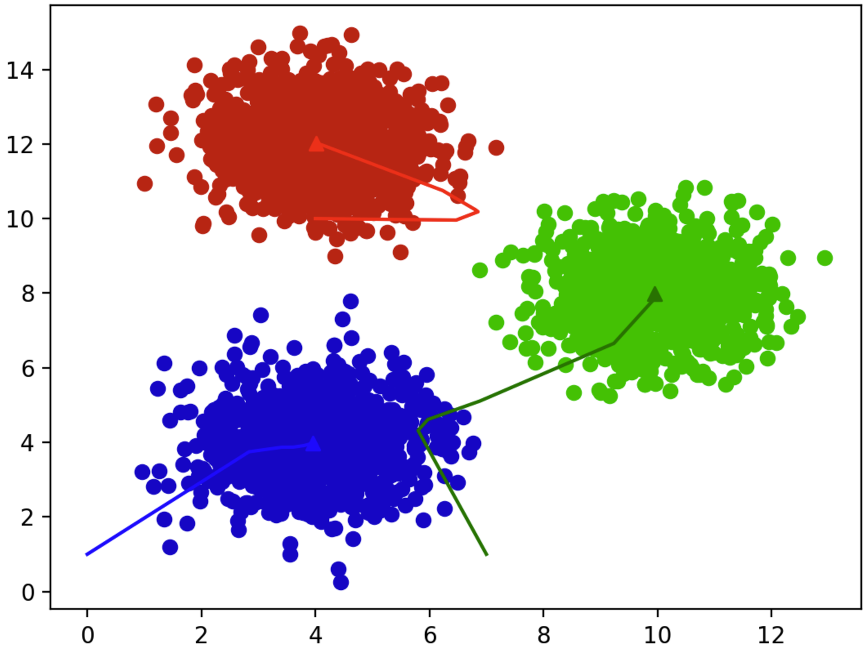

Figure 2: My Tertiary Kmeans classifier with dataset 1 having mean centered at 4,4 ,dataset 2 having a mean centered at 4,12, and dataset 3 having a mean centered at 10, 8. Initial centroids were [0,1], [4,10], and [7,1] respectively. It uses the same dataset as from the example

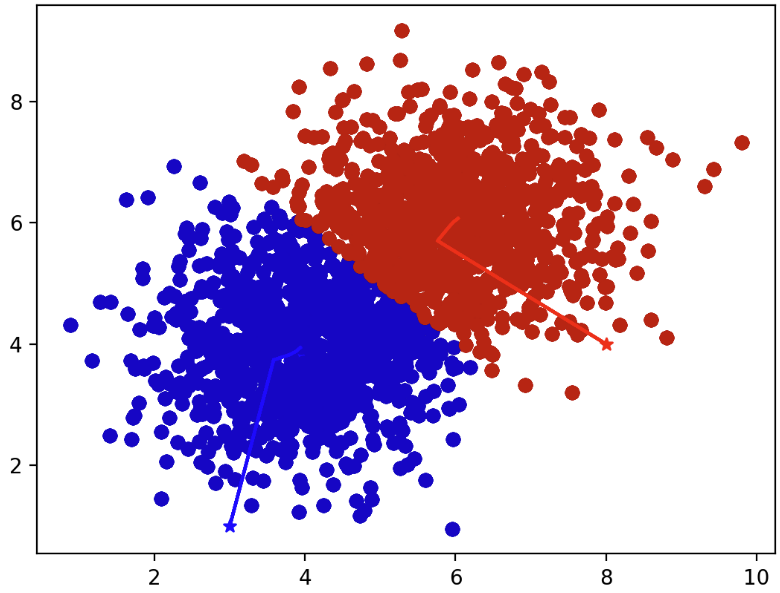

Figure 1: My Binary Kmeans classifier with dataset1 having mean centered at 4,4 and dataset2 having a mean centered at 6,6. Initial centroids were [3,1], [8,4] respectively.