Part 2:

The libraries using K-means and Expectations Maximization algorithm we found are “sklearn.cluster.KMeans” and “sklearn.mixture.GaussianMixture”.

1. K-means

The part1 k-means algorithm generates the similar outcomes from the sklearn library when k equal to 3. [check if different k has different results]

There are many parameters used to create clusters through KMeans library. Among them, we found that we could work on ‘init’ and ‘n\_init’ to optimize our codes. The method for initialization are chose through the parameter ‘init’. There are two methods, ‘k-means++’, ‘random’ and the default is set to be ‘k-means++’ which selects initial centroids in a smarter way to speed up convergence. ‘random’ means choose initial centroids randomly from data, which is also the way our kmeans algorithm uses. We might be able to decrease the number of iterations if we can utilize ‘k-means++’.

As the K-means algorithm guarantees only the local minimum solution, it is better that we run the code several times to get chance to find global solution. Therefore, the parameter ‘n\_init’ is introduced! By setting the parameter, the KMeans library algorithm will run with different initial centroids and returned the best output. We could improve our algorithm’s accuracy by adding the parameter.

1. Expectation Maximization algorithm (GMM)