Kmeans clustering

Download dataset from here.

Task: **TODO**

1. You have to implement K-means clustering algorithm with the seed data with 3 dimensions.
2. A sample implementation file is provided. Please download it. You can use notebook to complete the **TODOs**. The implementation file is actually a skeleton or pseudocodes. You need to fill up your code blocks where asked. Numpy array has been used to store the datapoints, centroids, indices in this skeleton.
3. At first, data are loaded. You can use the load function or panda function to load numerical data in numpy matrix. Use K = 3
4. Use initialize_centroids_simple() to initialize your centroids. This is the simple assignment function you need to implement. Randomly select **K points** from the sampled data and assign them as initialized centroids.
5. Then, in the kmeans function,
    a. You have to calculate the cluster_affiliation array based on the distance of each data points from all centroids.
    b. you have to write your own code to count the number of points assigned for each cluster based on the cluster_affiliation and store in the defined structure `clutser_point_count`
    c. You have to re-compute the centroids based on the cluster_affiliation and `clutser_point_count`
    d. Then write your own code to terminate the process based on the termination criteria discussed in the class. We evaluate the quality of the clustering using the clustering objective

$$J = \frac{1}{N}\sum_{i=1}^{N} \min_{j=1,\ldots,k} \|x_i - z_j\|^2$$

Where N is the total number of sampled points. $\boldsymbol{x_i}$ is the i^{th} data point. z_k is the centroid for k^{th} cluster. The algorithm is terminated when J is nearly equal in two successive iterations (e.g., we terminate when $|J - J_{prev}| \leq 10^{-5}J$, where J_{prev} is the value of J after the previous iteration, flag = False).

    e. In the main function, draw a 3D plot where:
        i. Using the given ClassLabel data, determine the shape of the each data point (triangle, square or circle).
        ii. plot the data points. Points in different clusters will have different colors. Such as for cluster 0: use blue, for cluster 1: red, cluster 2: yellow
        iii. Visually calculate how many data points are wrongly clustered where most of the data points are same in a cluster.