In this assignment, you should solve the problem of clustering using the TinyMNIST dataset.

PROBLEM 1

Implement agglomerative hierarchical clustering using predefined python libraries (scikit-learn). The function you should use is called *AgglomerativeClustering*. Compare mean of distances in each cluster. Calculate the within and between scatter matrices. Also calculate the $S_w^{-1}S_b$ and compare the results.

PROBLEM 2

Implement sequential clustering (also known as iterative optimization) using predefined python libraries (scikit-learn). The function you should use is called *AffinityPropagation*. Report measures mentioned in problem 1.

PROBLEM 3

K-means is a type of optimization-based clustering in which an objective function should be minimized. It aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a *prototype* of the cluster. Given a set of observations $(x_1, \dots, x_n)$ where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into $k \le n$ sets $S = \{S_1, \dots, S_n\}$ so as to minimize the within-cluster sum of squares (WCSS) (i.e. variance). Formally, the objective is to find:

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} \left|\left|x - \mu_i\right|\right|^2$$

Where μ_i is the mean of points in S_i.

K-Means Algorithm:

The *K*-means clustering algorithm uses iterative refinement to produce a final result. The algorithms starts with initial estimates for the *K* centroids, which can either be randomly generated or randomly selected from the data set. The algorithm then iterates between two steps:

1. Data assignment step
2. Centroid update step

The algorithm iterates between steps one and two until a stopping criteria is met (i.e., no data points change clusters, the sum of the distances is minimized, or some maximum number of iterations is reached).

Detailed algorithm is as below:

1- Begin: initialize $n, k, \mu_1, \dots, \mu_k$
2- Do classify n samples according to nearest μ_i
3- Recomputed μ_i
4- Until no change in μ_i

5- Return $\mu_1, \ldots, \mu_k$
6- End

Also these links can be useful to learn this type of clustering:
https://www.datascience.com/blog/k-means-clustering
https://en.wikipedia.org/wiki/K-means_clustering

Implement K-means clustering (also known as iterative optimization) using predefined python libraries (which is scikit-learn and the function is called kmeans) and report measures mentioned in problem 1.

PROBLEM 4

Cluster validity measures describe the quality of a complete clustering. *Separation Index* is one of such measures that is proportional to the ratio of between to within distance in clusters:

$$SI = \min_{j}\left\{\min_{i(i\neq j)}\left\{\frac{d(S_i,S_j)}{\max_{l} d(S_l,S_l)}\right\}\right\}$$

Where:

$$d(S_i,S_j) = \min\{d(x_i,x_j) | x_i \in S_i, x_j \in S_j\}$$

$$d(S_l,S_l) = \min\{d(x_i,x_j) | x_i, x_j \in S_l\}$$

Calculate this measure for problems 1, 2 and 3. Report and compare the results. Which method has better performance?