**CSC 535/635 – HW 5**

This homework assignment is optional. It will replace the lowest grade homework assignment for students who choose to do it. If you choose not to do this assignment, then please do not worry about Blackboard showing no grade for this assignment.

**Assignment description:** For this assignment, you will **implement the k-means clustering algorithm** as given in Module 13 PowerPoint slides, and apply the algorithm to the given dataset. In your implementation of k-means, you must **choose an appropriate stopping criterion**. Please seed your random number generator with a constant value so that different runs of your code will provide the same results. This is done in Python by calling random.seed(n), where n is a constant integer.

Use the given two-dimensional synthetic dataset to test your code. The data contains a total of 500 objects from 3 clusters. There are 200 objects in the first cluster, 150 in the second cluster, and 150 objects in the third cluster. The third attribute in the dataset is for the class/cluster label, which is provided so that you can test the accuracy of your implementation. Use the Euclidean distance measure to calculate distances.

The output from your program **must** display the following:

* The initial k-means picked by the algorithm.
* The number of objects in each cluster.
* The number of objects misclassified in each cluster.
* The objects classified in each cluster, along with the original class label for each object. You may show the original or normalized objects here.
* The accuracy rate.

# **Remarks:**

* Use the Euclidean distance measure to compute distances. You must use your own implementation of Euclidean distance. You cannot use any built-in implementation of Euclidean distance.
* You must **implement the k-means algorithm from scratch**:
  + You are not allowed to use any library implementation of the algorithm, which is provided in many languages.
  + You are not allowed to use any online sources. If you have questions about the algorithm, please feel free to ask either the instructor or the GA.
* You may use the statistics module in Python or any built-in function to compute statistics such as means, modes, standard deviation …etc. or to help with your code.
* If you like, you may visualize the input data using matplotlib. You do **not** need to turn in this as part of your solution for this homework assignment.

Sample output is as follows:

Initial k means are  
mean[0] is ((0.05841377449418182, 1.0301888224696443), 0)  
mean[1] is ((-1.810171639070208, -0.44470900915877676), 1)  
mean[2] is ((-0.9090375460986547, 1.050673514575594), 2)  
=====================  
Cluster 0  
Size of cluster 0 is 139  
Cluster label: 1  
Number of objects misclustered in this cluster is 8  
((0.2690684975264931, 0.05375183208601334), 2)  
((1.1272914432136873, -0.8680593126817495), 1)  
((2.859341388146024, 0.08106475489394709), 1)  
((0.8932306398444521, -1.305066077608689), 1)  
…  
((1.5056897419939497, 0.03326713998006349), 1)  
((0.5031293008957276, 0.4122339439401437), 2)  
=====================  
Cluster 1  
Size of cluster 1 is 200  
Cluster label: 0  
Number of objects misclustered in this cluster is 17  
((-0.6164615418871113, -0.10671158941059687), 0)  
((-1.3966642197845602, -0.2432762034502656), 0)  
((-0.5228372205394174, -0.6188288920593541), 0)  
…  
((-1.2913368582684048, -1.2675108087477802), 0)  
=====================  
Cluster 2  
Size of cluster 2 is 161  
Cluster label: 2  
Number of objects misclustered in this cluster is 24  
((0.12863201550495246, 2.1431904268929434), 2)  
((-0.47992607325505793, 0.914108900535926), 2)  
((0.05841377449418182, 1.0301888224696443), 2)  
((-0.04301257363248646, 0.28591167595345046), 1)  
((-0.323885537675568, 0.6887772873704727), 2)  
…  
((-0.0391115602429993, 2.6553077295417005), 2)

Accuracy rate is 90.20%

Class label is the majority class label for the objects in the current cluster.

Each normalized objects is formatted as ((normalized\_x\_value, normalized\_y\_value),   
 original\_class\_label)

# **Submission instructions:**

* For this assignment, you **must work alone**. This is **not** a group assignment.
* Use Python or IPython in your implementation.
* In a **docstring at the beginning of your code**, describe the stopping criterion you selected for the algorithm, the accuracy of the algorithm, how you normalize the data (if you normalize), and any useful information about the assignment you like to include.
* You need to achieve accuracy of at least 85%
* Make sure that your code is well organized and properly documented and commented.
* Name your program hw5\_YourFirstName\_YourLastName.py or hw5\_YourFirstName\_YourLastName.ipynb, as appropriate. For example, if a student’s name is John Doe, the file will be named hw5\_John\_Doe.py or hw5\_John\_Doe.ipynp.