Homework 1: Unsupervised Learning

CS 447: *Machine Learning*

Instructor: Collin Engstrom

Due: Friday, Sept 29th, by **11:59 p.m.**

Directions: Complete the following questions using the slides and textbook to help you. You may choose to handwrite your answers and scan them in or type them up. If you handwrite them, please make sure your handwriting is clear and legible, or I will deduct points.

For both the written and coding portions of this homework, ***DO NOT Google or use ChatGPT for material!!!*** *Use your own resources (textbook, slides, notes, etc.) or talk to me if you need help. If I catch plagiarized work, it will be handled accordingly.*

1. Hierarchical Agglomerative Clustering

Consider the following set of singleton clusters:
{{6}, {8}, {10}, {20}, {26}, {30}, {32}, {33}}
Recall that in performing Hierarchical Agglomerative Clustering (HAC), there are three linkages that are typically used in calculating inter-cluster distance: single linkage, complete linkage, and average linkage. Assume that in merging clusters ties are broken by merging the two clusters that result in the smallest sum for the points in the cluster. So, for instance, if we have the three clusters {1}, {2}, and {3}, the first two would be merged first since they result in a sum of 3, whereas merging the last two would result in a sum of 5.

a) Starting with the eight singleton clusters in the dataset above, perform HAC until you are left with one cluster. For full credit, show each step in the clustering process. Do this with:

   (i) Single linkage
   (ii) Complete linkage
   (iii) Average linkage

b) For your clustering using single linkage, also draw the resulting binary tree.

c) Suppose we wanted to know what the best number of final clusters would be. Briefly describe one way you might determine this. *Hint:* Consider how you would first determine how "good" each clustering is.

2. KMeans

Using Jupyter notebooks and the Python programming techniques we've been covering in class, complete the KMeans programming assignment. The skeleton code is available on D2L. Please note the following in your implementation:

1. You will fill in all the necessary code to make the KMeans code run. Note that in certain places I've left hints in the code. All input and return values are also stated in the function headers. Example outputs are currently in the Jupyter notebook to help check your work.
2. The code will expect both the `data_points.csv` and `centroids.csv` files in the same directory as your notebook. I have included sample files for you to work with in the zip archive. You should be able to run your KMeans implementation on these files and get the same output as what's currently in the Jupyter notebook.
3. You may **NOT** import any libraries (e.g., Pandas, Sci-Kit Learn, etc.) beyond what is already included in the skeleton code.
4. I will be calling all of your functions with my own test code and input files, so you must **NOT** change the function headers from what is in the skeleton code. Doing so will break the execution of your code, and I will deduct points.
5. You must also take care to return the expected value(s) from all of your functions.
6. You will likely need to refer to various APIs and tutorials online related to using Python. Please be certain that you are **NOT** copying code, though. Doing so constitutes cheating, as does copying code from others in class. If I catch instances of either of these, I will handle them in accordance with university policy.
7. You should also **NOT** consult other implementations of KMeans, as the final product you submit must be your own. You may, however, use notes, labs, and any other material we've used for class.
8. Since there are likely some details I've forgotten to include in this document, I've created a D2L discussion called "HW1." To access it in D2L, use the "Communications" menu and select "Discussions." You are free to post questions and discuss your ideas with each other and myself. Please be aware, though, that you must **NOT** discuss things at the implementation level. So, for instance, a discussion at the algorithmic level is permissible and encouraged, while a discussion of specific loops, if/else-statements, and so on should not be included. (You may privately talk to me about these matters if you're stuck, however.)

When you are finished, please submit your Jupyter notebook (.ipynb) file (from question 2) and your written file (from question 1) all in one .zip archive on D2L.

Please make sure you are submitting the correct version of your code!!!!