MSAI-349, Fall 2021
Homework #2: K-Nearest Neighbors and K-Means
Due Date: Tuesday, October 26th @ 11:59PM
Total Points: 10.0 (plus optional 1.0 bonus points)

In this assignment, you will work with your project group to implement distance metrics, a KNN classifier and a K-Means classifier. You may discuss the homework with other groups, but do not take any written record from the discussions. Also, do not copy any source code from the Web.

You will be working with the MNIST handwritten digits dataset. In this dataset the digits zero through nine are represented as a series of integers on a 28x28 grid, where the integers provide the grayscale intensity. From this data we constructed `train.csv`, `valid.csv` and `test.csv` splits with 2000, 200 and 200 observations, respectively. Each of the files contain one observation per row with a 'label' followed by 784 'grayscale' intensity values. A short program `starter.py` illustrates how to read these files and how to display the digit as 'bits' or 'intensity'.

Steps to complete the homework

1. (1,0 points) Implement functions to compute Euclidean distance and Cosine Similarity between two input vectors ***a*** and ***b***. These functions should return a scalar float value. To ensure that your functions are implemented correctly, you may want to construct test cases and compare against results packages like `numpy` or `sklearn`.
2. (4.0 points) Implement a k-nearest neighbors classifier for both Euclidean distance and Cosine Similarity using the signature provided in `starter.py`. This algorithm may be computationally intensive. To address this, you must use transform your data in some manner (*e.g.,* dimensionality reduction, mapping grayscale to binary, dimension scaling, etc.) -- the exact method is up to you. This is an opportunity to be creative with feature construction. Similarly, you are free to select your own hyper-parameters (e.g., K, the number of observations to use, default labels, etc.). Please describe all of your design choices and hyper-parameter selections in a paragraph. Once you are satisfied with performance on the validation set, run your classifier on the test set and summarize results in a 10x10 confusion matrix. Analyze your results in another paragraph.
3. (4.0 points) Implement a k-means classifier in the same manner as described above for the k-nearest neighbors classifier. The labels should be ignored when training your k-means classifier. Describe your design choices and analyze your results in about one paragraph each.
4. (1.0 points) Collaborative filters are essentially how recommendation algorithms work on sites like Amazon ("people who bought blank also bought blank") and Netflix ("you watched blank, so you might also like blank"). They work by comparing distances between users. If two users are similar, then items that one user has seen and liked but the other hasn't seen are recommended to the other user. What distance metric should you use to compare user to each other? Given the k-nearest neighbors of a user, how can these k neighbors be used to estimate the rating for a movie that the user has not seen? In about one paragraph describe how you would implement a collaborative filter, or provide pseudo-code.

5. (1.0 bonus points) Prepare a soft k-means classifier using the guideline provided for Question #3 above.

Please note there several aspects of this assignment are deliberately vague. This is to give you some experience with the issues you will likely encounter when working on your Final Projects. You may need to make simplifying assumptions about your data, devised workarounds for computation bottlenecks or balance time spent on hyper-parameter tuning versus results. Additional information on the MNIST dataset can be found here http://yann.lecun.com/exdb/mnist/. This site includes several academic papers on performing classification with the full version of these datasets.

Submission Instructions

Turn in your homework as a single zip file, in Canvas. Specifically:

1. Create a single pdf file hw2.pdf with the answers to the questions above and summaries of your results.
2. Create a single ZIP file containing:
   - `hw2.pdf`
   - All of your `.py` code files
3. Turn the zip file in under Homework #2 in Canvas.

Good luck, and have fun!