**MSAI-349**

**Homework # 2**

1. Please see the starter.py file for our Euclidean and cosine similarity distance calculation functions.
2. Please see the starter.py file for our k-nearest neighbor classifier functions, based on Euclidean and cosine similarity distance calculations. For data transformation to achieve computational efficiency, greyscale intensity values > 1 were amended to 1.
3. Please see the starter.py file for our k-means classifier functions, based on Euclidean and Cosine Similarity distance calculations.
4. User-based collaborative filtering involves comparing user preferences and/or behavior, to make recommendations and rating predictions. A user/movie interaction matrix can be constructed, in which rows correspond to users, columns to movies and entries to user ratings for each movie (ranging from 1-5), respectively. Very small non-zero default values can be set for missing entries, to avoid a sparse matrix. The mean of a user’s ratings is computed and subtracted from his or her rating for each individual movie. Based on these entries, cosine similarities between each user can be calculated and entered into a user/user similarity matrix. Values range from -1 (opposite) to 1 (most similar). Once sorted in descending order, those K users with the highest cosine similarity compared with another query user A would be considered nearest neighbors. As ratings are continuous, query user A’s rating of a given unseen movie can then be predicted, based on their nearest neighbors’ mean rating for that movie. Optimal K would be determined by that generating the highest accuracy on the test set.