k-Nearest Neighbor and Prediction Code Description

The code under the name of KNN_prediction takes four arguments.

Arguments:

1. The first is the 3D array that has people as rows (on the vertical axis) and classes as columns (on the horizontal axis). For each person/class pairing in the 3D array, there is an array of answers for all the numerical questions (exclude any non numerical answer, such as the time the survey was taken, the email address, etc).

2. The second is the similarity matrix. This is a 2D array that stores all the Pearson correlation coefficients for a given pair of people.

3. The third is the absolute value version of the similarity matrix. This is a 2D array that stores all the Pearson correlation coefficients for a given pair of people.

4. The fourth is the value of k. This will be used when finding the k-nearest neighbors and the overall prediction.

General 6 Step Overview:

Steps 2-6 occur for each missing answer in the 3D matrix

1. Finding Missing Student/Class Answers

    a. The program begins by finding all the unanswered student/class ratings in the 3D matrix. Then, it stores the blank into a tuple, consisting of the person and class where the blank occurs.

2. Finding the k-Nearest Neighbors

    a. Using the tuples, the program obtains the person to find the k-nearest neighbors to. Using the absolute value similarity matrix, it then goes through all the Pearson correlation coefficients (PCC) between the person and the other students and stores the k largest PCCs.

    b. Comparing the the largest PCCs to the student's PCCs in the similarity matrix, the program then stores the person in a list if the PCCs are equal.

3. Collecting the PCCs for the k-Nearest Neighbors

    a. The program goes through all the PCCs between the person the k-nearest neighbors and stores them in a dictionary.

    b. The dictionary is 0 indexed and the keys are people and the values are their corresponding PCC.

4. Collecting the Student/Class pair Ratings for the k-Nearest Neighbors

    a. The program goes all the people in the 3D matrix and checks if they one of the k-nearest neighbors. If so, the specific class's ratings are stores in a dictionary.

    b. The dictionary is 0 indexed and the keys are people and the values are a list of their answers for the given class.

5. Flipping the Answers for Negative PCCs

    a. The program checks if the PCCs for the k-nearest neighbors is negative. If so, it changes all the scores by applying the equation $y = 6 - x$, where x is the class rating and y is the new score.

    b. The program also changes the negative PCC to be positive.

6. Predicting an Answer

    a. Using the PCCs stored in step 3, the program sums them all together to get a total. Then, the PCCs are divided by the total to get a new PCC.

    b. The program takes the new PCCs and multiplies them by the corresponding class ratings.

    c. Lastly, the program adds up all the scores to form a prediction for the student/class pairing