Brandon Thai Tran
535 - Assignment 1 writeup

Preprocessing
I use pandas to handle the CSV by simply reading it in. I then turn the resulting dataframe to a numpy array, then define some variables to be used.

Question 1
Here, we compute the Fleiss' Kappa inter-annotator agreement. I first use a for loop to populate the Fleiss matrix as shown here in the table under Worked Example. The process was fairly simple. First, I initialized a N x k matrix of zeros where N is the number of subjects (2199 subjects) and k is the number of categories (3 categories). I then looped through the Fleiss Matrix and filled it by adding a one to the categories by each rater. From there, I simply used statsmodel's fleiss_kappa function with my fleiss matrix as the argument. The result was a Fleiss' Kappa score of 0.650, meaning the ratings have "substantial agreement" based on Table 1 in the assignment

Question 2
Here, we compute Cohen's Kappa between each annotator and the majority vote. To calculate Cohen's Kappa, I simply looped through the ratings and ran sklearn's cohen_kappa_score(ratings1,rating2) function with ratings1 being the annotator's rating and rating2 being the majority votes. The scores are as follows, ranked in order of their agreement with their Cohen's Kappa value.

Annotator	Cohen's Kappa
1	0.895
3	0.798
2	0.784

Note, I also computed the Cohen's Kappa between all combinations of the annotators. I left it because I found the result to be fairly interesting. It was interesting to note that Annotator 1 had the highest two highest Cohen Kappa in this experiment, since it also had the highest Cohen's Kappa score when compared to the majority vote.

Question 3
Here, we recalculate the majority vote by giving Annotator 1's vote to be weighed twice as much as Annotators 2 and 3. The premise is simple, treat Annotator 1's vote as two votes, meaning that the majority vote is calculated from 4 votes rather than 3 votes. To calculate the weighted majority vote, I loop through the subjects, and calculate the new weighted majority votes for each subject. There are two conditions in the for loop: (1) when rating2 equals rating3, but rating1 does not equal rating2 (and rating 3 since they are equal) (2) all other cases. Condition (1) describes a scenario of a tie in the weighted majority vote. To deal with ties, I chose to

randomly choose between rating 1 and the other rating (i.e. rating 2 and rating 3). Thi was to balance the expertise of rater 1 and the actual majority of the annotators. The other condition is when there is no tie in the weighted majority vote. The majority vote won't actually change in this scenario, but I chose to do it to make it easier. From here, I write to a csv file using numpy savetxt, which saves things into a CSV file.

Question 4
Here are some correlations that I came up with. The descriptions for why I chose them are included. I believe that all are positively correlated. I ended up choosing 6 behaviors. V stands for verbal cue and NV stands for a nonverbal cue.

Positive label
- Positive words like really good. (V)
  - Saying a movie is "good" generally is a positive thing.
- Smiling before or after utterance. (NV)
  - Smiling before or after an utterance is generally a positive thing.

Negative label
- Negative words like bad. (V)
  - If the reviewer says that the movie is bad or something of the like, it is generally a negative sentiment.
- Speaking about the movie's lack of an aspect or element. (V)
  - When the reviewer says that the movie is lacking something, it generally seems to be a negative thing.

Neutral label
- Neutral words like average. (V)
  - Average or words of the like are neutral in my opinion.
- Person is describing the movie. (NV)
  - I thought that the clips where the reviewers were just describing the movie were neutral since it was just (usually) an impassioned description.

Question 5
Here, I label the clips for 0h-zjBukYpk * and 1DmNV9C1hbY * myself based on the behaviors outlined in the previous step. I used pearson(dataX,dataY) in Excel to calculate the Pearson Correlation Coefficient. I found the Pearson Correlation Coefficient to be 0.873, which is fairly high. Below are my ratings.

Video ID	annotator_1	annotator_2	annotator_3	mine	majority_vote	Pearson Coefficient
0h-zjBukYpk_0001	0	0	0	0	0	0.8726706447
0h-zjBukYpk_0002	-1	-1	-1	-1	-1	
0h-zjBukYpk_0003	0	0	0	0	0	
0h-zjBukYpk_0004	0	-1	-1	-1	-1	
0h-zjBukYpk_0005	0	0	0	0	0	
0h-zjBukYpk_0006	0	0	0	0	0	
0h-zjBukYpk_0007	1	1	1	1	1	
0h-zjBukYpk_0008	0	0	0	0	0	
0h-zjBukYpk_0009	0	-1	0	-1	0	
0h-zjBukYpk_0010	0	0	0	0	0	
0h-zjBukYpk_0011	0	0	0	0	0	
0h-zjBukYpk_0012	0	-1	0	0	0	
0h-zjBukYpk_0013	0	0	0	0	0	
0h-zjBukYpk_0014	0	0	0	0	0	
0h-zjBukYpk_0015	0	0	0	0	0	
0h-zjBukYpk_0016	0	0	0	0	0	
0h-zjBukYpk_0017	0	0	0	0	0	
0h-zjBukYpk_0018	-1	-1	-1	-1	-1	
0h-zjBukYpk_0019	0	0	0	0	0	
0h-zjBukYpk_0020	0	0	0	0	0	
0h-zjBukYpk_0021	1	0	1	1	1	
0h-zjBukYpk_0022	1	0	1	1	1	
0h-zjBukYpk_0023	1	1	1	1	1	
0h-zjBukYpk_0024	1	1	1	1	1	
0h-zjBukYpk_0025	1	1	1	1	1	
1DmNV9C1hbY_0001	-1	-1	-1	-1	-1	
1DmNV9C1hbY_0002	1	0	1	1	1	
1DmNV9C1hbY_0003	0	0	1	0	0	
1DmNV9C1hbY_0004	1	1	0	1	1	
1DmNV9C1hbY_0005	-1	0	0	-1	0	
1DmNV9C1hbY_0006	-1	0	-1	-1	-1	
1DmNV9C1hbY_0007	-1	-1	0	0	-1	

1DmNV9C1hbY_0008	-1	-1	0	-1	-1	
1DmNV9C1hbY_0009	1	1	1	1	1	
1DmNV9C1hbY_0010	1	1	1	1	1	
1DmNV9C1hbY_0011	-1	-1	0	-1	-1	
1DmNV9C1hbY_0012	0	0	0	0	0	
1DmNV9C1hbY_0013	0	0	-1	-1	0	