Collaboration statement: I did not collaborate with anyone on this assignment, nor did I use any other outside sources besides the ones given to us in the HW 2 folder.

Report submission

1. Chosen minimum support threshold

For the minimum threshold, I decided to go with the minimum we were allowed to use for this assignment, which was 500. I decided on 500 because there are 26,000 tweets in the dataset, so I figured that a support of 500 would represent a frequently used keyword or keyword combination. Efficiently speaking, 500 is not too low where it may affect run time, and it is not too high where we may lose important information. So I decided that 500 was a good balance between efficient runtime, and ensuring that we do not lose valuable information.

2. Algorithm implementations & optimizations

To start, I uploaded the csv file using the Panda's library, and then I split the keywords to get each tweet's transaction. I did this by using the semi-colon to help identify the key words. Then for the Apriori algorithm, I used a lot of helper functions that I call in the main Apriori method to help me organize my thinking. My first helper function helped me set the frequacy of each key word from all of the tweets. So my output was all of the key words from the transactions, and all of their respective frequencies. Then, my next helper method helped me apply the minimum support threshold to all of the keywords. Like I said earlier, I decided on 500, so my output was a list of key words that met the threshold, along with their respective frequencies. My next helper method united frequent (k-1) itemsets to give me canidate k-itemsets. I used k=2 because at this point, we only had 1-itemset frequencies, so for this method to work, k needed to be 2. My output was all of the combinations of the 1-itemsets to 2-itemsets. The next helper method I tried to implement was a pruning method. This method stumped me, as I am not sure if it was implemented correctly, and I will talk more about that in the next section. The next helper method was count_candidates, which took the pruned candidates and counted the frequency of each. So the output is the candidates with their respective frequencies. Then, I created a main Apriori function where I call all of these methods, and the output is all of the k-itemset candidates with their respective frequencies.

3. Results Analysis

The most common keyword in this dataset was 'flu' which makes sense because this is a dataset mostly about the flu shot. The 8th and 9th most common combination of these keywords was "got shot" and "get shot." This leads us to believe that many of those people in the dataset did get the flu shot, and advised others to do so. We can say that many were in favor of getting the flu shot. Many of the top 20 combinations of the keywords in this dataset all fall along the lines of saying they got the flu shot, or advising to get the flu shot. This further backs up the claim that many were in favor of the flu shot during this time. Thus, we are able to get a sense of the public's health sentiment because words such as 'need' and 'go get' indicate a positive sentiment towards the flu shot.

4. Lessons learned.

Some challenges I've ran into were mostly from the pruning method. It seems as though it worked when I tested it on a test set, but the output was weird. Overall, next time I would try to improve on the pruning method and make sure that there is no confusion there. Besides that, everything else including the output txt file looks correct to me.