Jinghan Sun

jsun288@emory.edu

Student ID: 2520665

Collaboration statement: Resources used: course material;towardsdatascience.com;geeksforgeeks.org;stackoverflow.com; http://fimi.uantwerpen.be/src/apriori.c;My own code from CS334 course and my own previous project codes.

2.3 Report (15 points): Write a report in a .pdf file presenting your results in your .txt file. Report the threshold of frequent count you chose to generate your output file and explain why. Explain and discuss, if any, the algorithmic optimizations you have used in your implementation. Discuss the experiences and lessons you have learned from this assignment. Report the minimum support value you use and analyze the results and discuss about what knowledge you can learn from the patterns your implemented method has found.

The chosen threshold for frequent itemsets was a minimum support count of 500. This threshold was selected based on the following considerations:

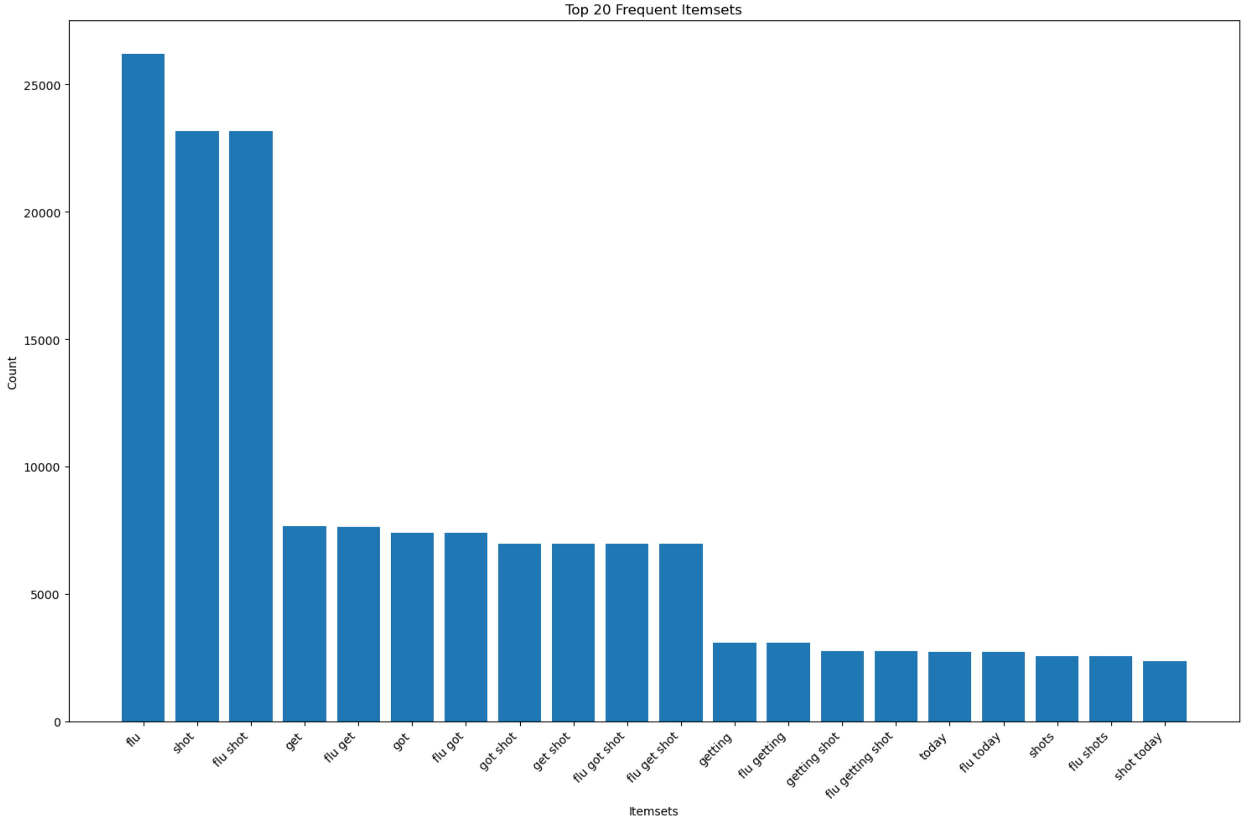
- Total keywords: 141,525
- Average keywords per transaction: 5.4
- Desired balance between significance and computational efficiency

Explanation: The threshold of 500 represents approximately 0.35% of the total keyword count. This relatively low percentage allows for the discovery of meaningful patterns while still filtering out infrequent itemsets that may not provide valuable insights. It takes about 35 seconds to run on my computer with CPU 13th Gen Intel(R) Core(TM) i7-13620H 2.40 GHz.

The most important optimization is the use of the Fk-1 xFk-1 method. In generate_candidates method, I first sorted the frequent itemsets and then merge two frequent (k-1)-itemsets if their first (k-2) items are identical.

From this assignment I understand the apriori algorithm more thoroughly by hands on implementation of it. I also learned that optimizations in algorithms can reduce runtime drastically, so I should not only make sure the program are compatible without errors or bugs, but also try to reduce its runtime through optimization. I also became more familiar with some python syntax and functions, such as extend(), as in the past I only familiar with append().

Minimum support value: 500.



Above is the plot for top 20 frequent itemsets with most numbers of support counts. “flu” “shot” and “flu shot” are the 3 most frequently occurring words, showing that the dataset is in the context of flu shot, with not much outliers or messy information unrelated to the topic. Also, there’s a very strong association between "flu" and "shot". Almost every mention of "shot" is in the context of "flu shot". Words like "get" (7,647), "got" (7,396), and "getting" (3,087) are among the top frequent items, indicating actions such as getting flu shots, which is the most concerning actions for people. "today" and “flu today” appears frequently (2,729 times and 2726 times), suggesting many people discuss getting flu shots on the day they receive them. "arm" appears 2,023 times, likely referring to the injection site. There are lots of words such as “sore” “hurts” and “like” occurs, suggesting people like to discuss their feelings and post-vaccination conditions, or this is the part most people concerned about. "free" appears 1,013 times, suggesting discussions about vaccine availability or cost. "mom" (552 occurrences) indicates discussions in the context of workplace vaccinations and family health decisions. Overall, people likes to share their thoughts on the day they get flu shot, and their discussion and concerns mainly centered around feeling or physical reactions after vaccination, cost, and family.