Sarah Batta, Arushi Srivastava, Stephanie Cao, Madelyn Dempsey

This project uses data from a kaggle set called “All COVID-19 Vaccines Tweets” by Gabriel Preda - https://www.kaggle.com/gpreda/all-covid19-vaccines-Tweets.

We parsed the data from the csv file above in the coding portion of our project. In our proposal we wrote that we intended to use the Instagram API to investigate the use of hashtags among public users in order to track which hashtags were growing in popularity, and use these conclusions to determine which hashtags and topics popular creators were using on Instagram. However, we ran into some issues in getting the permissions for the Instagram API, so we researched the Twitter API instead, but again we ran into trouble with figuring out how to actually use the API. As a result, we met with our project TA Julie who recommended finding a Twitter dataset on Kaggle and using that instead. On Kaggle we found a dataset for coronavirus Tweets, so we decided to use this set to form a recommendation system that recommends hashtags based on the hashtag that a user searches for. In the implementation of this system, we parse a csv file of the Tweets and relevant information for each Tweet, and we made a Tweet object for each line. Then, we created a linked list of each of these Tweets so that we could use this set to make a corpus of documents where the document content was the Tweet content. Using these documents, we checked their cosine similarity to each of the three covid vaccine brands in order to see which was the most popular on Twitter, and we predicted that the most effective vaccine would have the greatest popularity in recent coronavirus-related Tweets. Then we performed an empirical analysis on this hypothesis. However, we ran into several issues with this implementation with heap space and other technical issues so we changed our empirical analysis to an experiment on the graph we created based on hashtags to compare the relevance of different vaccine brands in terms of their hashtag use.

Work Breakdown:

Arushi: I worked on the recommendations feature, which used the data from kaggle about COVID-19 tweets. Using the HashMap mapping hashtags to a Linked List of Tweets that contain that hashtag that Steph made, I was able to make a weighted graph in the WeightedGraph.java file. In this graph, the nodes were all the hashtags present in all the tweets and there was an edge between two hashtags if they were mentioned in the same tweet. Furthermore, the weight of each edge was the number of tweets that had both hashtags. Creating this graph was not that difficult once the HashMap was made, but because of the size of the dataset, it took too long of a time to make the graph. Thus, instead of checking every hashtag with every other hashtag to see if they are contained within the same tweet, we made an adjacency matrix to reduce the number of

comparisons made. I also helped debug some of the issues we were having with the GUI layout and Java Swing.

Sarah: I worked on the empirical analysis part of this project. When considering a hypothesis for this project, I thought that drawing a correlation between the different brands of the covid-19 vaccines and their popularity in tweets would be quite relevant. In order to analyze this, the hypothesis looks into the correlation between vaccine effectiveness (given a specific brand) and the popularity of the vaccine brand in tweets. At first glance, it seems as if the most effective brand would have the highest popularity, but after considering this further, I realized that non-effective vaccine brands could also potentially gain popularity for the wrong reasons.

Madelyn: I researched external datasets for twitter data after we ran into problems with the APIs and I found the covid-19 dataset which looked like it would be useful and interesting for our project ideas. I was in charge of writing and modifying code for the empirical analysis portion of the project. I copied the document, corpus, vector space model, and tester files from homework 4, and then rewrote them to accommodate tweets instead of text files, and I wrote the methods to go through the tweets we parsed and check for cosine similarity to 5 different covid vaccines and then made a popularity calculation to see which vaccine brand was the most relevant on twitter recently. Then I worked with Sarah to complete the code for getting the most relevant tweet data for our hypothesis. Then I wrote the writeup and analysis of our data collection process and findings for the empirical analysis experiment. I also wrote the project summary and a few of the explanations for the key features of our project. Once I was almost finished with the empirical analysis code and analysis I ran into issues getting the final results so after trying to debug and alter my plan it still wasn't working so I chose another area for experimentation and designed and implemented a new empirical analysis, then wrote the writeup and analysis for it in a separate document. I also worked with Steph to add and change some of the methods in the WeightedGraph class in order to print out a graph visualization that I could use to see exactly which hashtags were connected and by what weight. Then, once I finished the empirical analysis portion I helped Steph and Arushi create the final UI.

Stephanie: I initially worked to get access to the Instagram and Twitter APIs, but then once we found a better dataset that was easier to use, I wrote a program to parse the CSV file. In order to parse the file, I used the Scanner class and separated the different columns by the delimiter comma, or by certain characters in each column. I then created a Tweet class so that we could initialize Tweet objects with fields corresponding to different columns in the row for that Tweet. Fields that I kept track of include the date, the Tweet ID, the username, the text, the hashtags, number of reTweets, and number of favorites. I created a HashMap mapping hashtags to a Linked List of Tweets that contain that hashtag so that Arushi could create a weighted graph connecting hashtags. We were running to a runtime issue with the implementation of our hashtag graph (it was taking multiple hours), so I changed most of it to run faster. Instead of iterating

through every combination of two tweets and calculating the weight of each hashtag in each tweet every time, I wrote a method to update the weight of an edge when two hashtags were found in a tweet together. Additionally, I added in an adjacency matrix representation of the graph so that I could decrease the time it took to check for the existence of a particular edge. I also edited the getRecommendations method so that it was able to find the edges of a particular node in decreasing order so that it could be returned to the user.

I also helped Maddie with her code for looking at cosine similarity between Tweets, however we were not very successful. We ran into a few issues since we were running out of heap space very quickly, and when I suggested modifying the code so we would only make vectors for the query terms and not every term in the corpus, the program broke.

Additionally, I worked on creating the graphic user interface for a user to use our getRecommendations feature. I used Java Swing and created input dialog boxes so that the user could enter the hashtag they wanted recommendations for and the number of recommendations they wanted. However, I had a lot of trouble making the layout of the GUI the way I wanted it to; labels weren't centering the way I wanted them to and my text boxes weren't going in the right spot. In the end, I think it worked out but it didn't really turn out the way I imagined.

Finally, I also was responsible for putting everyone's code together, from setting up our Github repository to sending files back and forth.

Features:

1. Graph: We created a graph where the nodes are the hashtags and there is an edge between two nodes if a Tweet contains the same hashtag. The edges are also weighted and the weight of an edge is the number of Tweets that contain those two hashtags. Then, for a given hashtag, we find the top three edges with the highest weights and recommend those hashtags to the user.
2. Recommendations: Building on the Graph aspect, we prompt the user to choose a hashtag to look at, and then based on that hashtag we find the top three edges with the highest weights and recommend those hashtags to the user since in this implementation the nodes are hashtags. This allows us to give the user a recommendation for other hashtags that might be related to or relevant to what the user is searching for or interested in.
3. Social Networks: Our dataset uses data from Twitter representing tweets related to covid-19 vaccines worldwide. In our project, we use social network principles to analyze connections between common hashtags used in coronavirus tweets as well as the growth in conversations about certain vaccine brands compared to others in our empirical analysis.