OVERVIEW

Some important info:

- In my graph representation:
    - Each node is a user
    - An edge between two users is undirected and unweighted
    - An edge only exists if the similarity index for that user-to-user pair is above a certain threshold. The threshold can be changed up to see how that affects the graph.
    - The similarity index between user 1 and user 2 is calculated by taking all the words that user 1 has ever used in their comments, all the words user 2 has ever used in their comments, and putting all that into a hashset of total words. Then, the words that belong to BOTH separate users will be a hashset of shared words. The similarity index is the length of the shared set divided by the length of the total set.

Project dataset: Youtube Comments Spam Dataset; Size: 1953

Project goals/questions to answer:

- Compare spam content similarity across users
- How many disconnected subgraphs are within the entire graph? How does this number change as the threshold for similarity changes?
- How many disconnected subgraphs are there in the *spam network only*? (including only nodes that represent spammers)
- Who are the spammers with the most similarity to each other? What words did they use?

DATA PROCESSING

- It was loaded in through a CSV. The CSV itself was added to my cloud environment path and I had a function read the data in and turn it into an array.
- Not a lot of cleaning was required. The data was separated by commas, but elements that contained commas were in double quotes, so I just needed to adjust my reader parameters accordingly.
- The columns I extracted were those of index 1, 3, 4, and 5. This is the user, comment content, video name, and the classification as spam or not.
- Each element was put inside a ColumnVal enum as either a String or boolean.

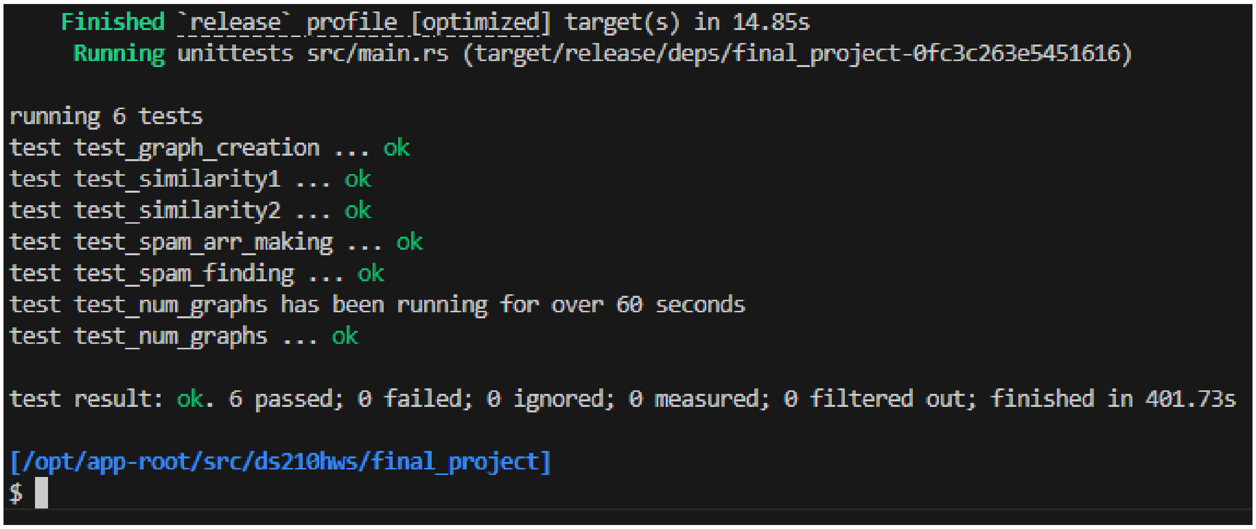
CODE STRUCTURE

- spam_functions.rs is a module that contains spam-specific functions--the functions that will be used to identify only spammers' nodes and to find the best spammers.
- csv_functions.rs is a module that contains the two functions that read in the CSV and return arrays. They deal directly with how the data is read in and are thus in the same module.
- The key functions that aren't so obviously explained are as follows:
    - find_spam() is a function to find information on the spam-only network—the nodes in the graph of all nodes that correspond to spammers. It takes in an array

of data and a vector of unique users. It returns the number of spam users and a vector of spam users. It iterates over the column of the array that corresponds to the classification as spam or not. When it comes across a “true” (classification for spam) it will take note of that user and increase the spammer counter

    - find_best_spammer() finds a list of the best spammers—the nodes in the graph that have the most neighbors. It takes in a graph and returns a hashset of names. It iterates through the graph (hashmap) to find the key(s) with the longest vector of neighbors.
    - map_users_to_words() essentially maps users to the words they used across all their comments. It takes in an array and returns a hashmap of users mapped to a hashset of words they used, as well as a vector of unique users. The logic of this function is to iterate over the array, extract the comment content column and split it up into words, filtering out non-alphanumeric characters.
    - find_similarities() finds the similarity index between any two users. It takes in two people’s names, the hashmap mapping names to words, and returns an Option. The logic of this function is that it iterates through the hashmap values for the people of interest, finds the number of unique common words they used, and divides that by the number of total unique words they used.
    - find_num_disconnected_graphs() finds the number of disconnected subgraphs in the graph. This number obviously changes as the threshold for similarity changes. It takes in a graph and returns a u32. The way it counts the number of graphs is by looking at each node, trying to traverse all possible neighbors that the node is connected to as well as the neighbors of neighbors, until all possible neighbors are explored. Then it jumps to the next node that hasn’t been seen before (a counter increments each time a new subgraph is found) and continues until all nodes have been seen.
- Main Workflow
    - Data is loaded in → data of interest is put into an array → that array is used to map users to words they used → the words used help calculate similarity indices for different users → build a graph where edges exist when similarity reaches a certain threshold

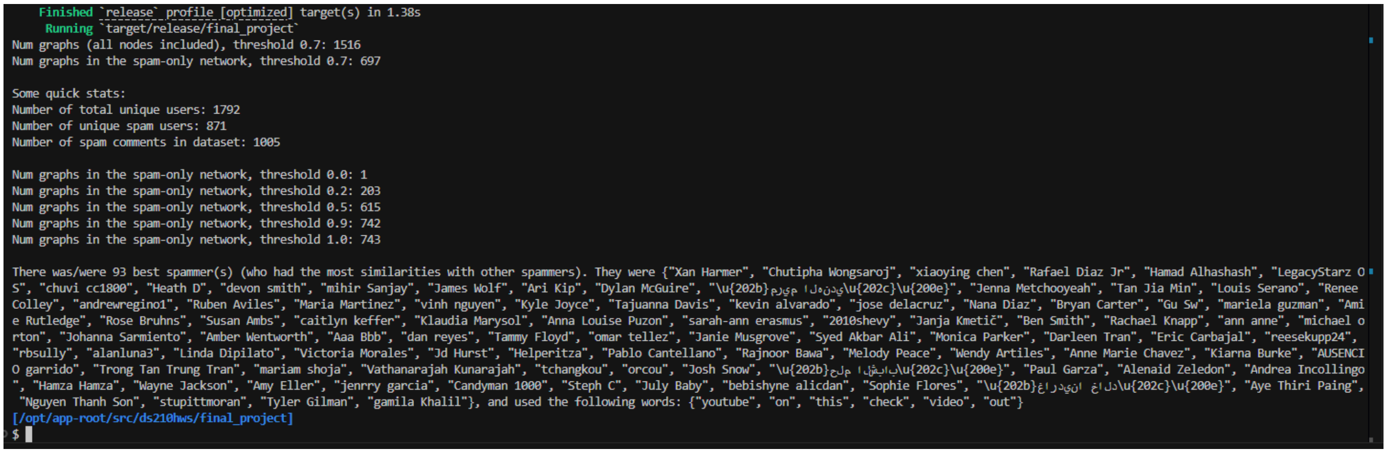
TESTS



```
    Finished `release` profile [optimized] target(s) in 14.85s
     Running unittests src/main.rs (target/release/deps/final_project-0fc3c263e5451616)

running 6 tests
test test_graph_creation ... ok
test test_similarity1 ... ok
test test_similarity2 ... ok
test test_spam_arr_making ... ok
test test_spam_finding ... ok
test test_num_graphs has been running for over 60 seconds
test test_num_graphs ... ok

test result: ok. 6 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 401.73s

[/opt/app-root/src/ds210hws/final_project]
$ 
```

- The first two test functions test the function find_similarities(). I calculated the similarity index by hand for two different pairs and used the assert_eq!() macro to see if the similarity index was being calculated correctly.
- The third function tests the graph creation function by creating a very small graph of three users and some words they used. With a threshold of 1.0, edges will only be drawn if the two users share the exact same set of used words. Three mutable booleans initialized to false will only become true if the requirements I coded are met, such as the graph having all three users as keys, Sara and John each having each other as their neighbors, and Teah having an empty vector of neighbors. If all three tester bools are true, the final tester bool is made true, and this is used in assert_eq!() against a true.
- test_num_graphs() is a simple test to make sure that when the threshold is 0.0, there should only be 1 disconnected subgraph, AKA the entire graph itself since every node should be connected at this threshold.
- test_spam_finding() creates a mini dataset/array and ensures that users are accurately being determined as spammers. I wanted to make sure that users that had multiple comments, at least one of which was marked spam, be marked as spammers.
- The last test function fn test_spam_arr_making() tests the function that makes an array specifically for the spam network (we exclude nodes that are non-spammers). It just makes sure that after we create the spam-only array, the number of unique spammers there is equal to 871, which we have determined previously.

RESULTS



```
    Finished `release` profile [optimized] target(s) in 1.38s
     Running `target/release/final_project`
Num graphs (all nodes included), threshold 0.7: 1516
Num graphs in the spam-only network, threshold 0.7: 697

Some quick stats:
Number of total unique users: 1792
Number of unique spam users: 871
Number of spam comments in dataset: 1005

Num graphs in the spam-only network, threshold 0.0: 1
Num graphs in the spam-only network, threshold 0.2: 203
Num graphs in the spam-only network, threshold 0.5: 615
Num graphs in the spam-only network, threshold 0.9: 742
Num graphs in the spam-only network, threshold 1.0: 743

There was/were 93 best spammer(s) (who had the most similarities with other spammers). They were {"Xan Harmer", "Chutipha Wongsaroj", "xiaoying chen", "Rafael Diaz Jr", "Hamad Alhashash", "LegacyStarz O
S", "chuvi cc1800", "Heath D", "devon smith", "mihir Sanjay", "James Wolf", "Ari Kip", "Dylan McGuire", "\u{202b}مريم الهندي\u{202c}\u{200e}", "Jenna Metchooyeah", "Tan Jia Min", "Louis Serano", "Renee
Colley", "andrewregino1", "Ruben Aviles", "Maria Martinez", "vinh nguyen", "Kyle Joyce", "Tajuanna Davis", "kevin alvarado", "jose delacruz", "Nana Diaz", "Bryan Carter", "Gu Sw", "mariela guzman", "Ami
e Rutledge", "Rose Bruhns", "Susan Ambs", "caitlyn keffer", "Klaudia Marysol", "Anna Louise Puzon", "sarah-ann erasmus", "2010shevy", "Janja Kmetič", "Ben Smith", "Rachael Knapp", "ann anne", "michael o
rton", "Johanna Sarmiento", "Amber Wentworth", "Aaa Bbb", "dan reyes", "Tammy Floyd", "omar tellez", "Janie Musgrove", "Syed Akbar Ali", "Monica Parker", "Darleen Tran", "Eric Carbajal", "reesekupp24",
"rbsully", "alanluna3", "Linda Dipilato", "Victoria Morales", "Jd Hurst", "Helperitza", "Pablo Cantellano", "Rajnoor Bawa", "Melody Peace", "Wendy Artiles", "Anne Marie Chavez", "Kiarna Burke", "AUSENCI
O garrido", "Trong Tan Trung Tran", "mariam shoja", "Vathanarajah Kunarajah", "tchangkou", "orcou", "Josh Snow", "\u{202b}حلم الشباب\u{202c}\u{200e}", "Paul Garza", "Alenaid Zeledon", "Andrea Incollingo
", "Hamza Hamza", "Wayne Jackson", "Amy Eller", "jenrry garcia", "Candyman 1000", "Steph C", "July Baby", "bebishyne alicdan", "Sophie Flores", "\u{202b}غاردينا غادل\u{202c}\u{200e}", "Aye Thiri Paing",
 "Nguyen Thanh Son", "stupittmoran", "Tyler Gilman", "gamila Khalil"}, and used the following words: {"youtube", "on", "this", "check", "video", "out"}
[/opt/app-root/src/ds210hws/final_project]
$ 
```

Pasted:

Num graphs (all nodes included), threshold 0.7: 1516
Num graphs in the spam-only network, threshold 0.7: 697

Some quick stats:
Number of total unique users: 1792
Number of unique spam users: 871
Number of spam comments in dataset: 1005

Num graphs in the spam-only network, threshold 0.0: 1
Num graphs in the spam-only network, threshold 0.2: 203
Num graphs in the spam-only network, threshold 0.5: 615
Num graphs in the spam-only network, threshold 0.9: 742
Num graphs in the spam-only network, threshold 1.0: 743

There was/were 93 best spammer(s) (who had the most similarities with other spammers). They were {"Xan Harmer", "Chutipha Wongsaroj", "xiaoying chen", "Rafael Diaz Jr", "Hamad Alhashash", "LegacyStarz OS", "chuvi cc1800", "Heath D", "devon smith", "mihir Sanjay", "James Wolf", "Ari Kip", "Dylan McGuire", "\u{202b}مريم الهندي\u{202c}\u{200e}", "Jenna Metchooyeah", "Tan Jia Min", "Louis Serano", "Renee Colley", "andrewregino1", "Ruben Aviles", "Maria Martinez", "vinh nguyen", "Kyle Joyce", "Tajuanna Davis", "kevin alvarado", "jose delacruz", "Nana Diaz", "Bryan Carter", "Gu Sw", "mariela guzman", "Amie Rutledge", "Rose Bruhns", "Susan Ambs", "caitlyn keffer", "Klaudia Marysol", "Anna Louise Puzon", "sarah-ann erasmus", "2010shevy", "Janja Kmetič", "Ben Smith", "Rachael Knapp", "ann anne", "michael orton", "Johanna Sarmiento", "Amber Wentworth", "Aaa Bbb", "dan reyes", "Tammy Floyd", "omar tellez", "Janie Musgrove", "Syed Akbar Ali", "Monica Parker", "Darleen Tran", "Eric Carbajal", "reesekupp24", "rbsully", "alanluna3", "Linda Dipilato", "Victoria Morales", "Jd Hurst", "Helperitza", "Pablo Cantellano", "Rajnoor Bawa", "Melody Peace", "Wendy Artiles", "Anne Marie Chavez", "Kiarna Burke", "AUSENCIO garrido", "Trong Tan Trung Tran", "mariam shoja", "Vathanarajah Kunarajah", "tchangkou", "orcou", "Josh Snow", "\u{202b}حلم

الشباب\u{202c}\u{200e}", "Paul Garza", "Alenaid Zeledon", "Andrea Incollingo", "Hamza Hamza", "Wayne Jackson", "Amy Eller", "jenrry garcia", "Candyman 1000", "Steph C", "July Baby", "bebishyne alicdan", "Sophie Flores", "\u{202b}غاردينا خالد\u{202c}\u{200e}", "Aye Thiri Paing", "Nguyen Thanh Son", "stupittmoran", "Tyler Gilman", "gamila Khalil"}, and used the following words: {"youtube", "on", "this", "check", "video", "out"}

Interpretation:

- Multiple users had a similarity index of 1.0, because multiple users had the exact same comment text in the form “Check out this video on YouTube:”.
- There were 93 spammers with identical words used and these were the ones classified as the “best spammers.” They had the highest similarity indices among themselves.
- As expected, the number of disconnected graphs goes up as the threshold increases. A threshold of zero yields a single interconnected graph.
- Spammers comprised of 871 / 1792 or about 48.6% of the total users.
- At a baseline threshold of 0.7, the number of disconnected subgraphs using nodes from the ENTIRE dataset was 1516. The number of disconnected subgraphs using ONLY spammers as nodes was 697, so about a 54% decrease.
- Relevance: by analyzing the similarity among comment content of spammers and finding the common words they used, we can potentially use this data to train algorithms to classify spam.

USAGE INSTRUCTIONS

- I would highly suggest using cargo run --release.
- If you run into a permission denied error while running on the cloud environment, paste the following into the terminal: export CARGO_HOME=/opt/app-root/src/.cargo.
- Run-time should be under a minute using cargo run --release.

CITATIONS

- No collaborators