Daniel Ayabe, Remi Kalbe

Prof. Hood

CS 579

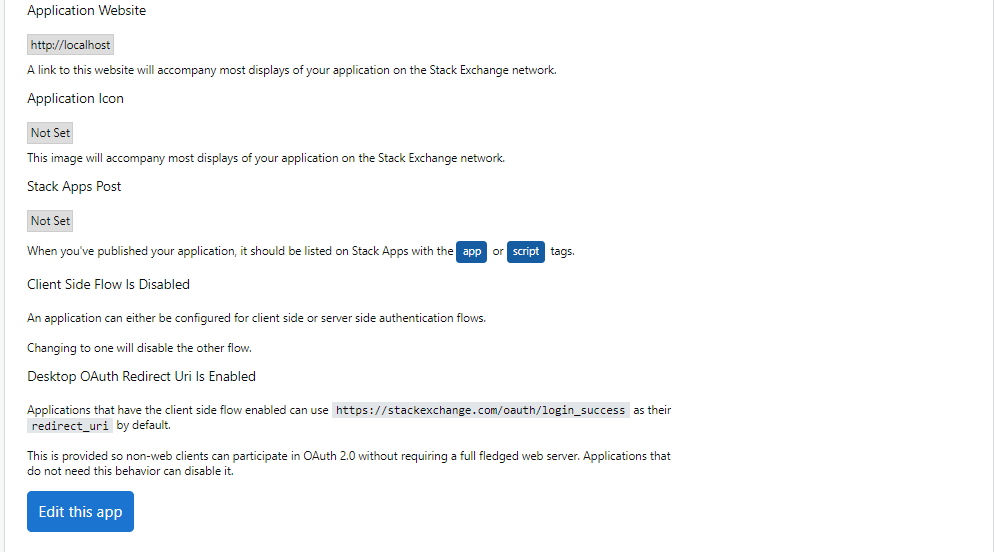
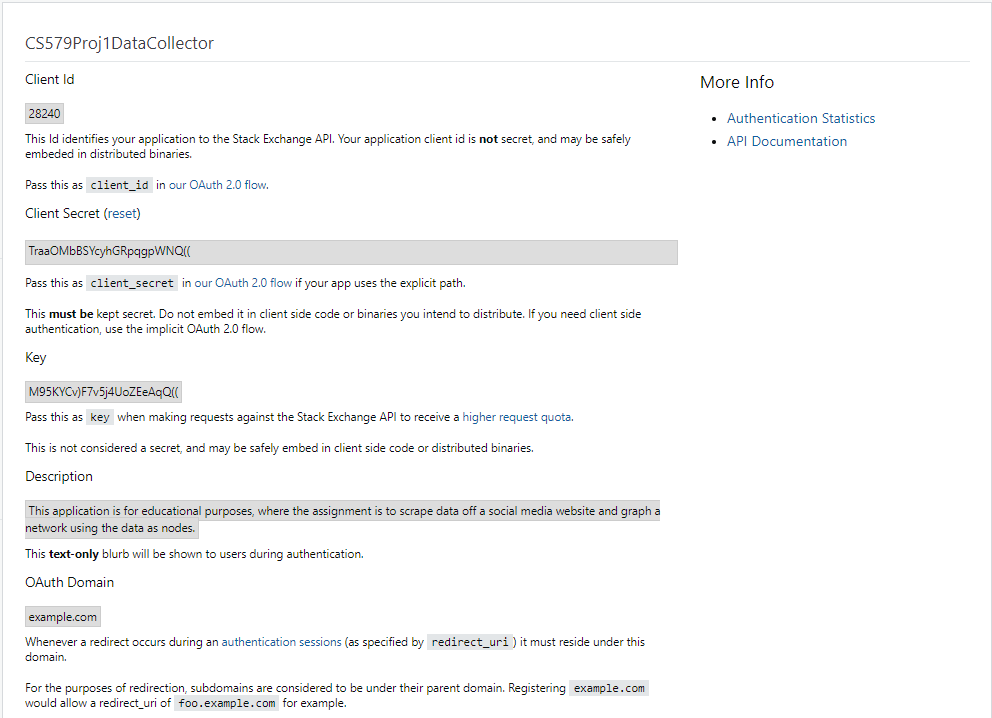
02/20/2024

**Project 1: Social Media Data Analysis**

Data Collection:

For data collection, originally, we had decided on selecting Reddit as our source of social media to analyze, with using API methods in order to gather data. However, when we looked further into checking the data licenses and the privacy licenses, it required too much effort just to use the Reddit data for educational purposes. We then decided to switch to Stack Overflow, where the policies were very simple to follow in order to legally use the data on the website for educational purposes.

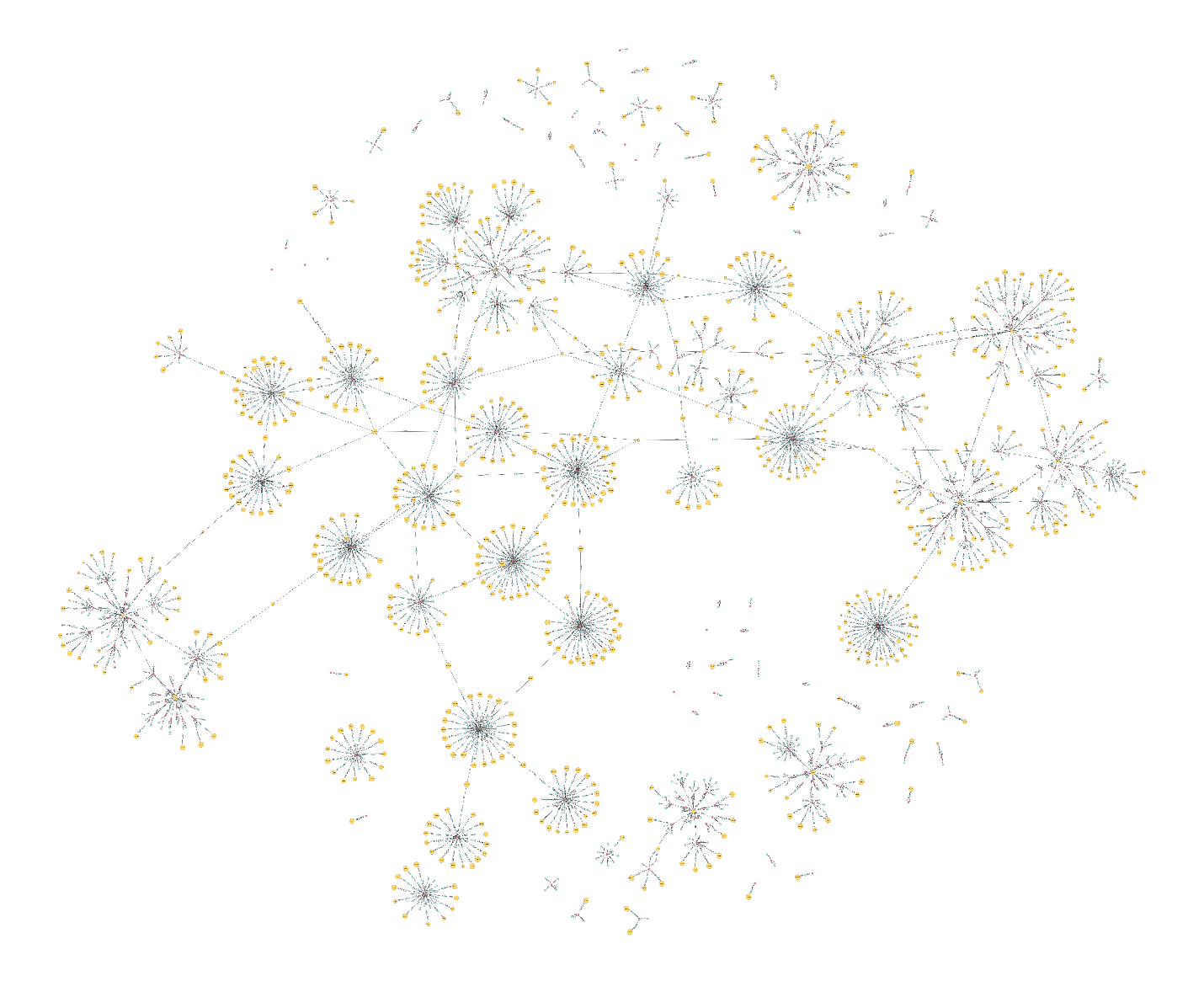
Next, we decided to create a Python script that would gather data of questions asked, filtering the questions, comments, authors, and highest upvotes specifically. The code can be found in the collection folder of our project repository. To make sure that we did not violate any of the policies mentioned beforehand, we had to create a stackapps project do receive a client ID in order to use the Stack Exchange API, which is shown below in the 2 screenshots provided.



Data Visualization:

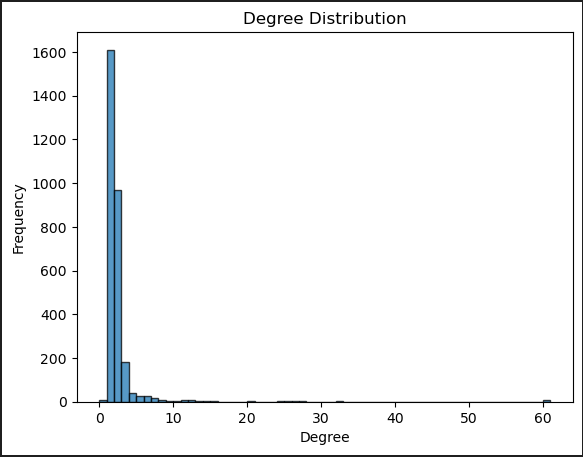
For the data visualization, we specified certain filters to gain specific aspects of information from each stack overflow post. The user’s account and user IDs, number of answers and question the user has asked and given, the number of upvotes and downvotes, the acceptance rate of the user’s answers, their location, and lastly, their reputation.

Next, we decided to use a graph analysis software to visualize our network as a graph. We chose graph-tool because it is a Python module, and we wrote our code in Python. Its memory usage and computational time are also efficient when graphing too. Along with that, we also needed to install pixi for managing the dependencies when using graph-tool. Our format for our data input files were in the .parquet file format because this file format enables less data scanned at ingestion, meaning that faster scans are performed with lower query and memory costs because the file format enables you to only read the columns of data that are needed. Below is the screenshot of the graph.

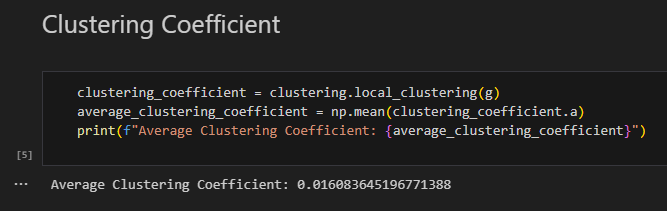


Network Measures:

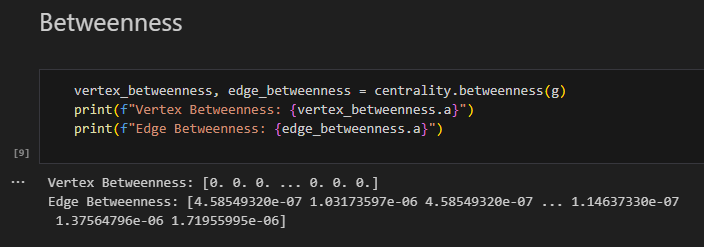
Degree Distribution Graph Histogram:



Clustering Coefficient:



Betweenness:



Discussion of Results:

*Data Visualization*:

For the data visualization network graph, the users are represented in yellow; the questions are represented in red; and the answers are represented in blue. There are a significant number of instances where a user only replied to one question (when the users are in the outer edge of the circle) as supposed to some instances where the users reply to more than one question (when the users are in the center of the circle). In addition, there are also questions and answers with no user, most likely due to deletion of accounts.

Some further questions for these results could be how introducing another factor (such as upvotes and downvotes) affect the clustering and circles of the network graph? Could they affect the number of deleted users as well as the number of answers? A possible next step could be to re-graph the network graph with these added factors, or graph a different network graph using these factors only.

*Degree Distribution Graph Histogram:*

For the histogram of the degree distribution, we plotted frequency over degree to see where the most common degrees were for the data. Degrees of 0-5 were most frequent, with the frequency values being 200, 950, and 1600. With such low degrees, this means that there were very few questions between the stack overflow data gathered given the questions, answers, and authors.

Some further questions could be why the degrees were so low? A possible next step to this question could be to filter in the most common topics of stack overflow questions, as a possible hypothesis to this question could be that most questions asked have to do with programming as opposed to traditional social media trends and topics.

*Clustering Coefficient:*

As seen in the screenshot above, the average clustering coefficient is 0.016083645196771388. With a coefficient value of 1 meaning that a node and its neighbors are close, and a value of 0 meaning that a node and its neighbors are loosely connected, this coefficient value means that our data recorded does not hold a close connection from one node to each other.

One question that may be asked is why the coefficient is very low, given that stack overflow is tailored to programming questions? Further steps to investigate this could be to filter for a specific topic of questions asked (such as python only or java only), and perform the clustering coefficient calculation again.

*Betweenness:*

The betweenness of the of the vertexes of the network graph is 0, while the edge betweenness averages from 4.585E-7 to 1.71E-6. This low betweenness value means that the nodes are not very well connected to the rest of the nodes that belong to the other clusters within the network. This means that the posts on stack overflow do not play a role in facilitating flow of information across the stack overflow network, which makes sense due to the fact that most users will post a question specifically for the answer, not for spreading information across the stack overflow network.

A follow-up question could be how stack overflow would be impacted if its structure changed so that the betweenness would be a high value? In this case, the posts would have to in some way spread information across the network for a steady flow of information, which is a possible future topic to investigate.

References

* <https://stackoverflow.com/legal/terms-of-service/public>
* <https://stackapps.com>
* <https://api.stackexchange.com/docs>
* <https://stackapps.com/apps/oauth/register>
* <https://graph-tool.skewed.de/>
* <https://pixi.sh/>
* <https://docs.pola.rs/py-polars/html/reference/>