FUNDAMENTALS OF DATABASE MANAGEMENT SYSTEM

END TERM PROJECT

TERM - III

SOCIAL NETWORK ANALYSIS

2018_Contributions_Reddit_Amber Heard Data

Submitted to:

Prof. Ashok Harnal

Submitted by:

Group-11

Yashasvi Goyal – 025038

Malvika Saxena – 025020

Savitri Pathak – 025030

(PGDM- Big Data Analytics, FORE School Of Management, New Delhi

This dataset is of the **year 2018** which gives us an insight about Reddit threads of the actress Amber Heard. After getting our nodes and edges file of our Network Analysis on **Amber Heard's Reddit** thread to further analyse it we imported all the 3 files to Gephi. We have in total **2826 Nodes and 2693 Edges.**

For **Graph Processing** we choose the option **Fruchterman Reingold** with parameters(Area = 30000 and Speed = 30) and the output can be clearly visible in Fig 1.

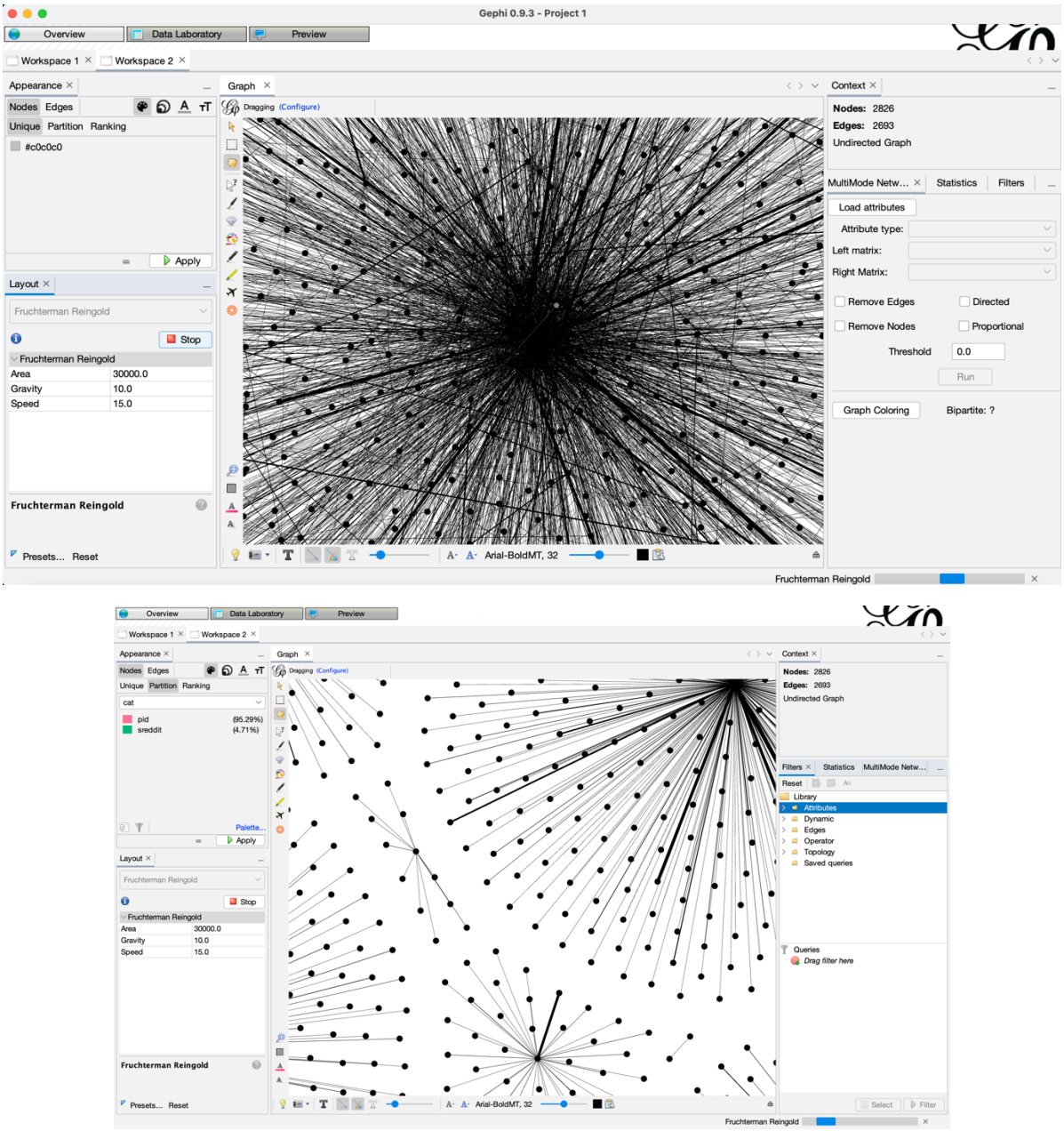


Figure 1

In Figure 2 we can see the zoomed out picture of the same

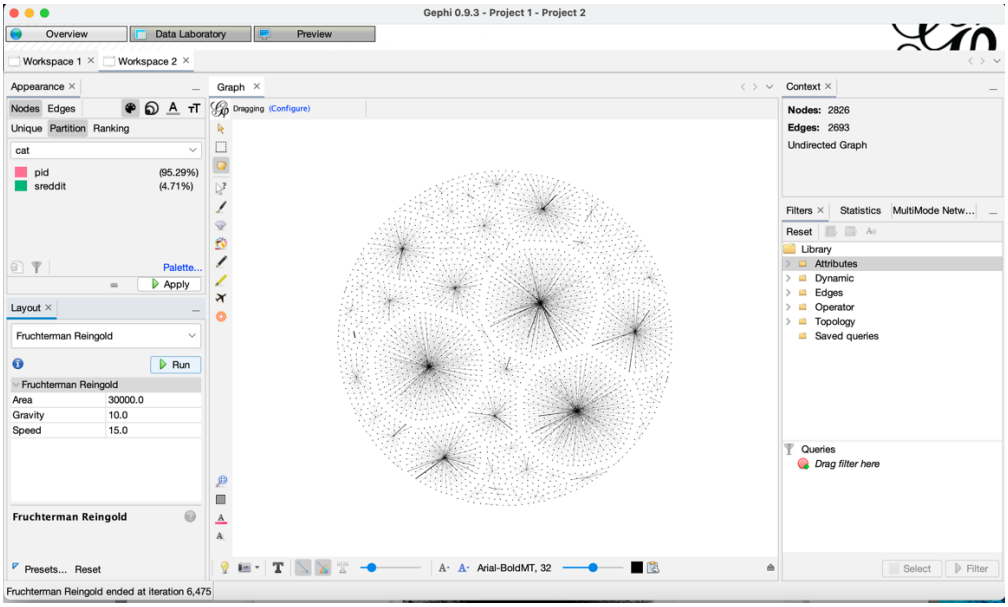


Figure 2

After getting the layout of our Network Analysis to further analyse it we portioned them as per category and coloured them.

In the Partition Dropdown we selected cat**. The Pink one denotes Sreddit id (Subreddit id) and Green ones denotes pid (Parent id)** which is clearly visible in Figure 3 and 4.

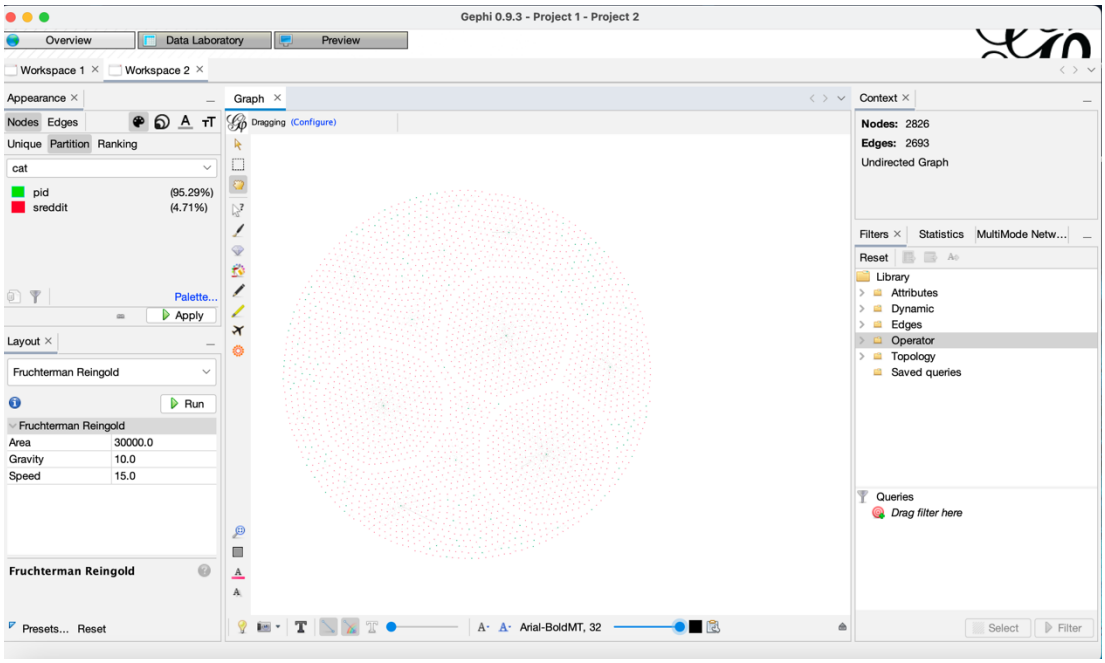


Figure 3

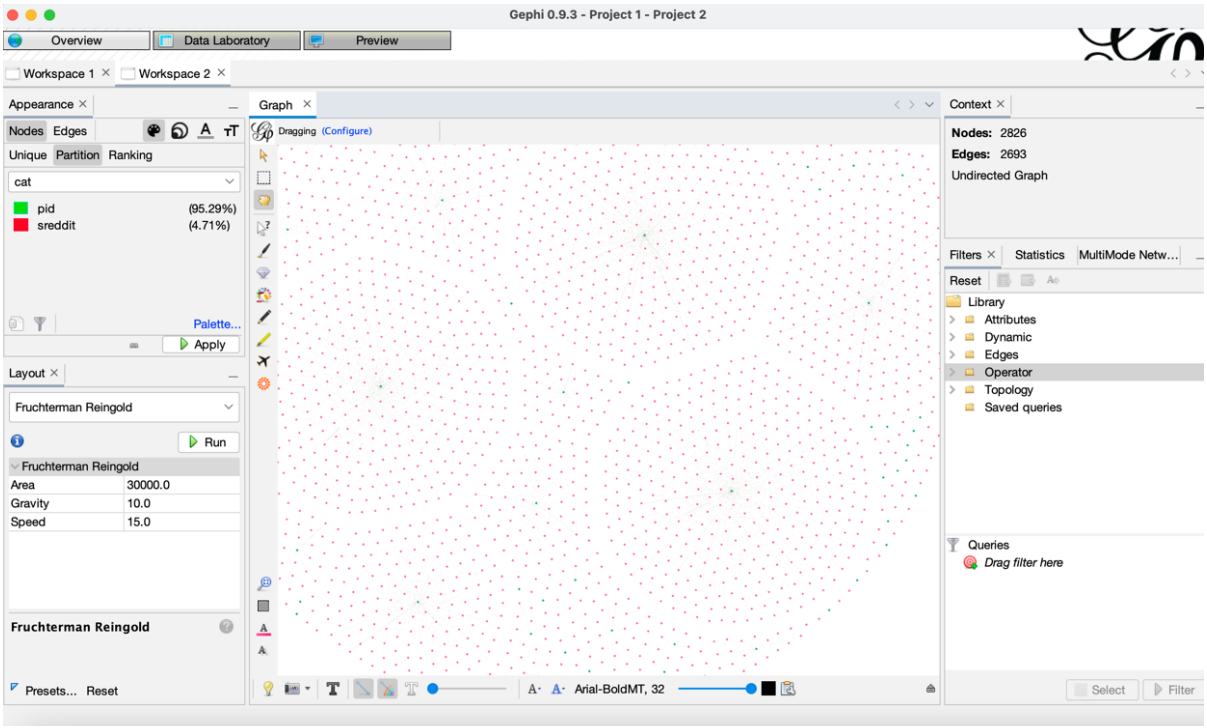


Figure 4 (Zoomed in)

To extract the more valuable insights we used sentiments and **increased the size of the nods** to make them clearly visible on the rim which is visible in Figure 5.

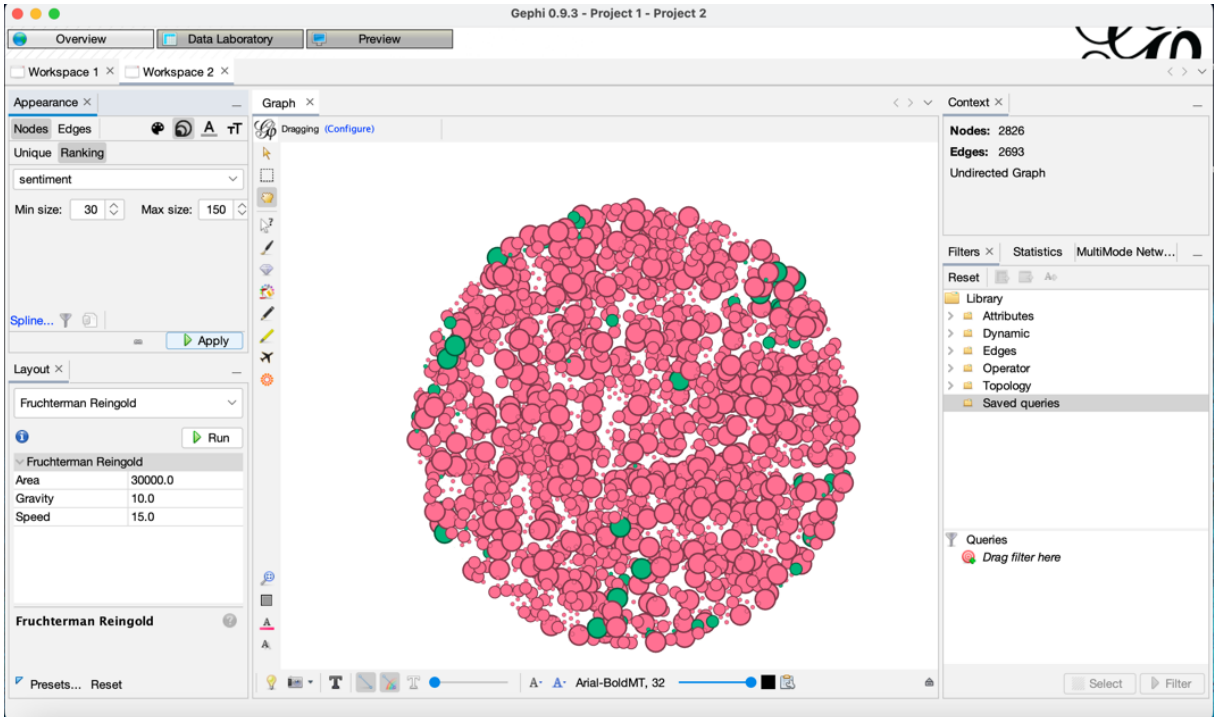


Figure 5

Then we **Projected the two mode graph to one mode**. In **Multimode Network for Projection** .We selected our attribute as cat. Then we selected our left and Right matrix and removed extra nodes and edges.

In figure 6 we can see the output as just sreddit nodes when we selected our Left Matrix as sreddit–pid and Right Matric – pid–sreddit.

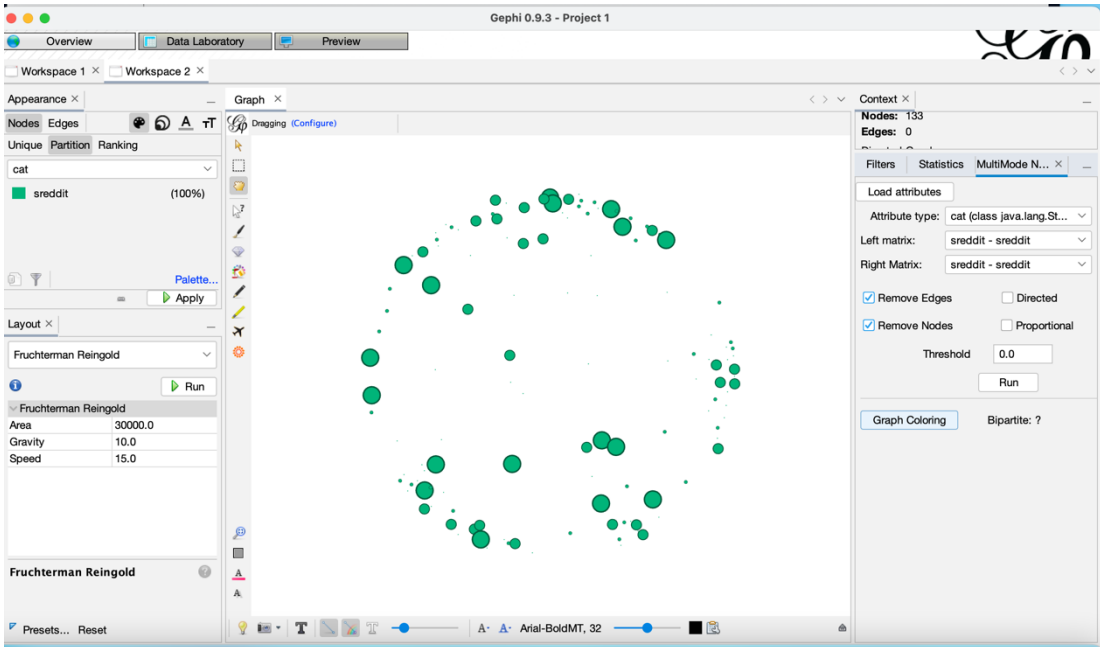


Figure 6

In figure 7 we can see the output when we selected our left matrix as pid-sreddit and right matrix as sreddit-pid which is only pid nodes.

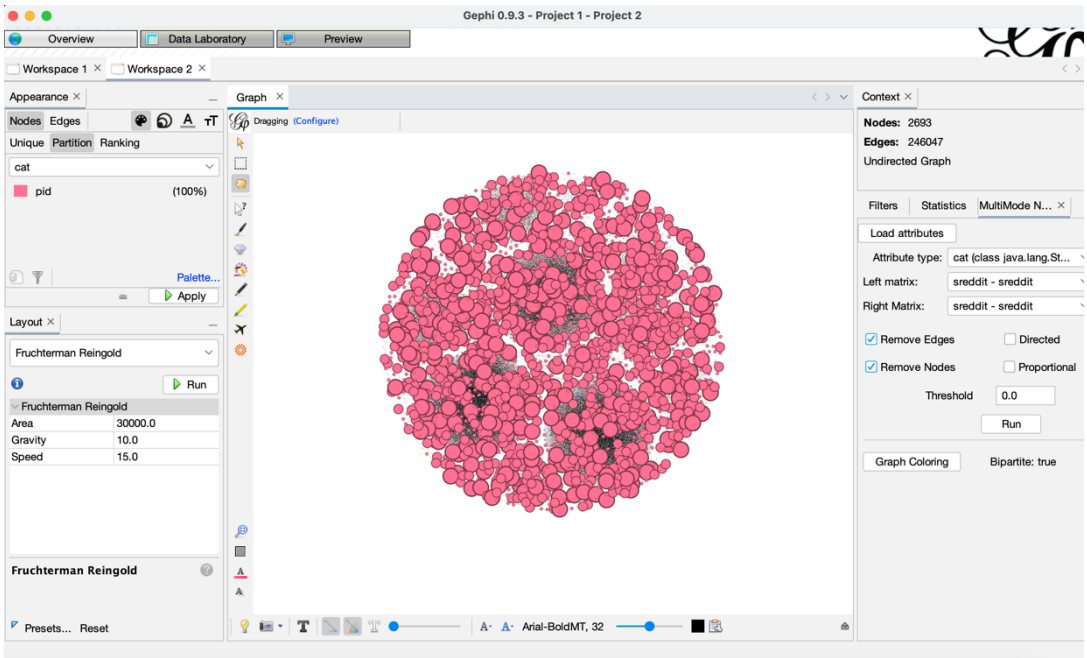


Figure 7

Then we checked the **modularity report** in the Statistics tab. In which the modularity came out as 0.801with 133 as number of communities as denoted in Figure 8 .

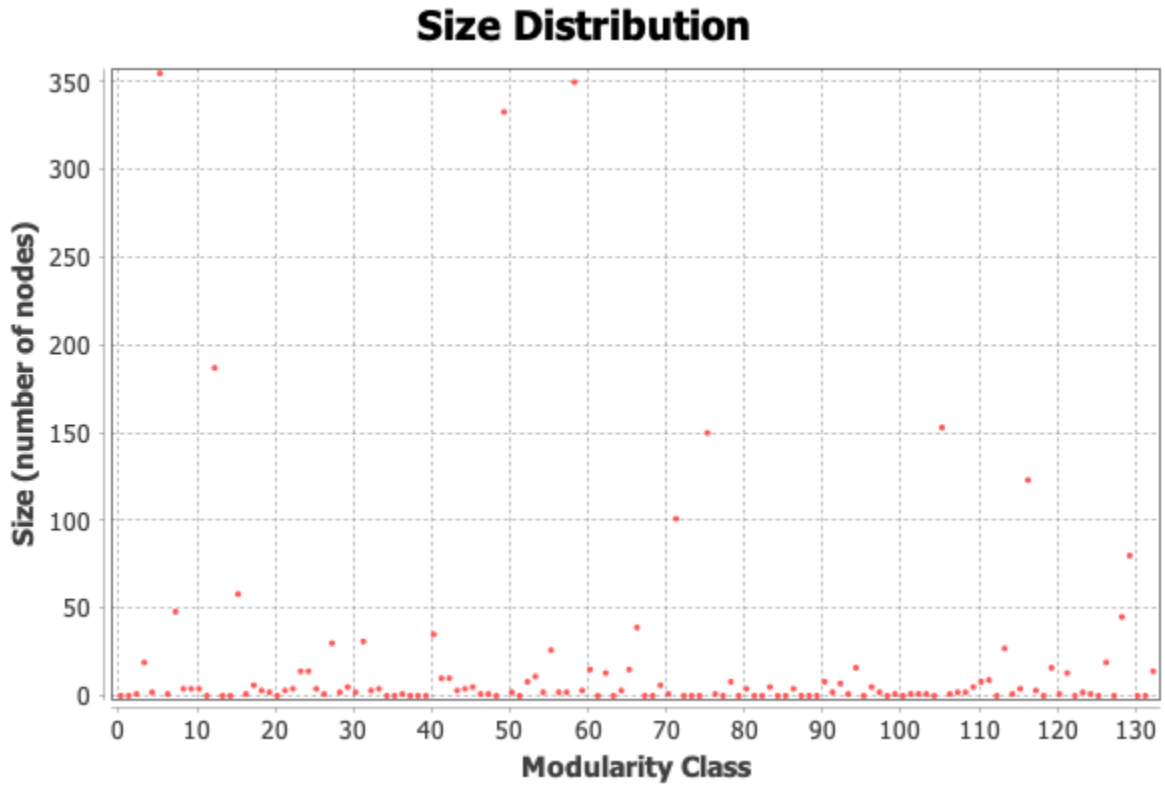
Modularity Report

Parameters:

Randomize: On
Use edge weights: On
Resolution: 1.0

Results:

Modularity: 0.801
Modularity with resolution: 0.801
Number of Communities: 133



Algorithm:

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre, *Fast unfolding of communities in large networks*, in Journal of Statistical Mechanics: Theory and Experiment 2008 (10), P1000

Resolution:

R. Lambiotte, J.-C. Delvenne, M. Barahona *Laplacian Dynamics and Multiscale Modular Structure in Networks 2009*

Figure 8

Then we **Coloured nodes as per community class**. We Clicked on Nodes and Partition, colour icon and in the drop-down we selected Modularity class. We can see the output in Figure 9

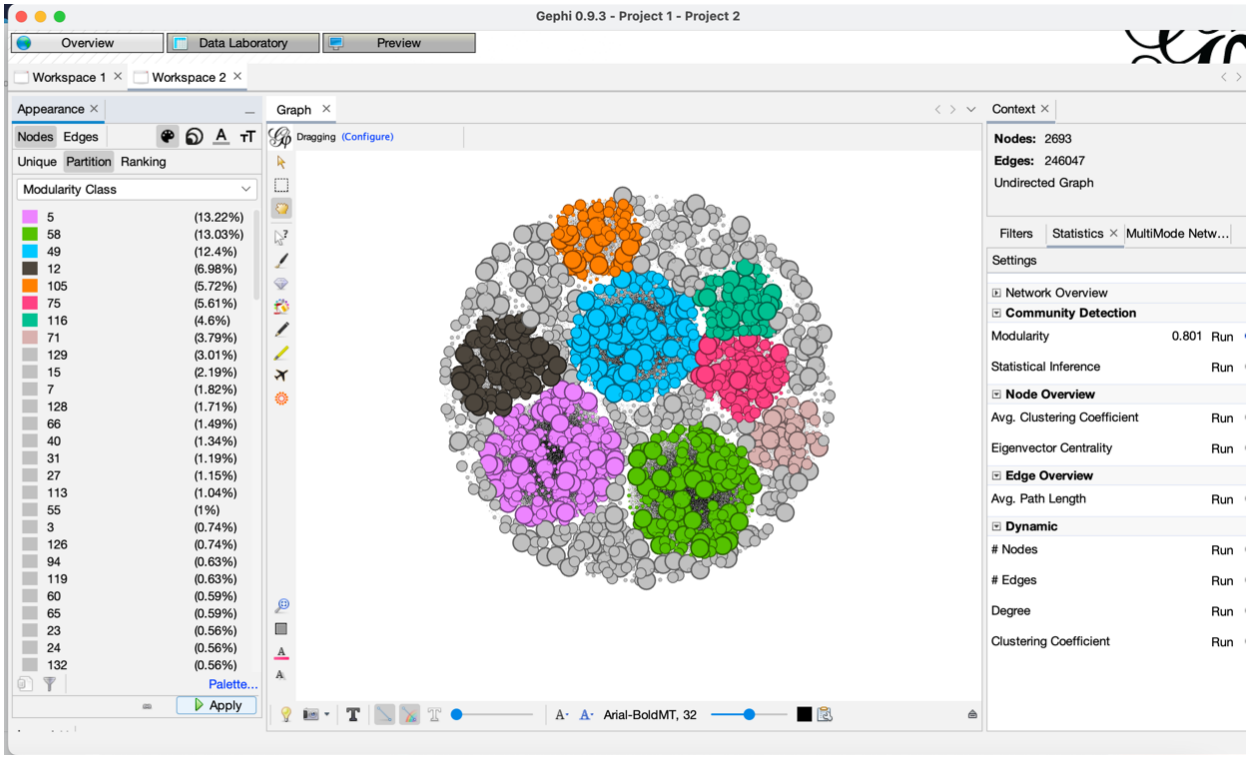


Figure 9

Then we started with **Filtration** where we created **nested filters**. In the filter tab in Range we selected (U._ mean). Then we added a filter which we selected from Partition i.e., Sentiment.

In figure 10 we can see all the messages with sentiment 0 (30.61%)

In figure 11 we can see all the messages with sentiment 1(24.77%)

In Figure 12 we can see all the messages with sentiment 2(13.8%)

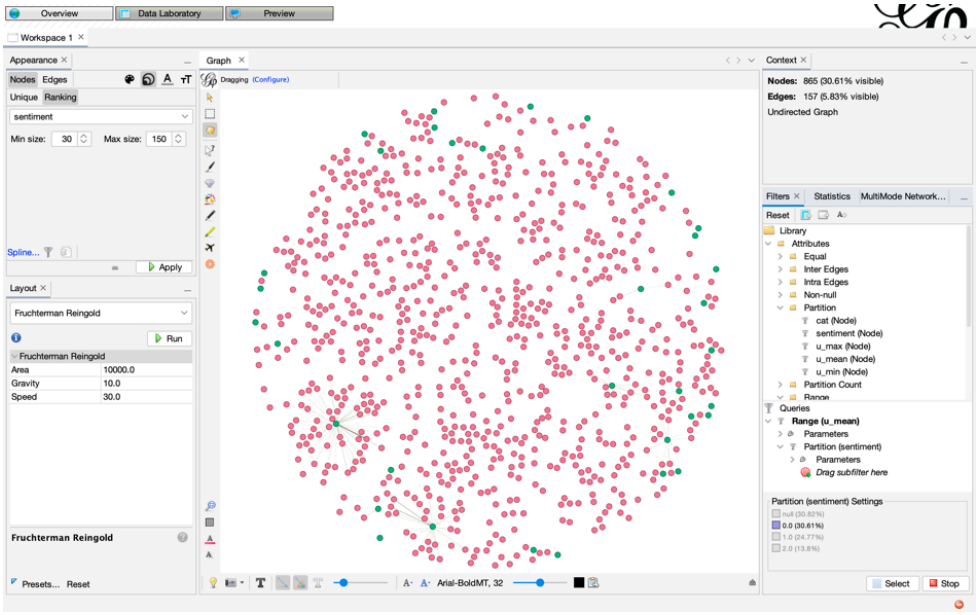


Figure 10

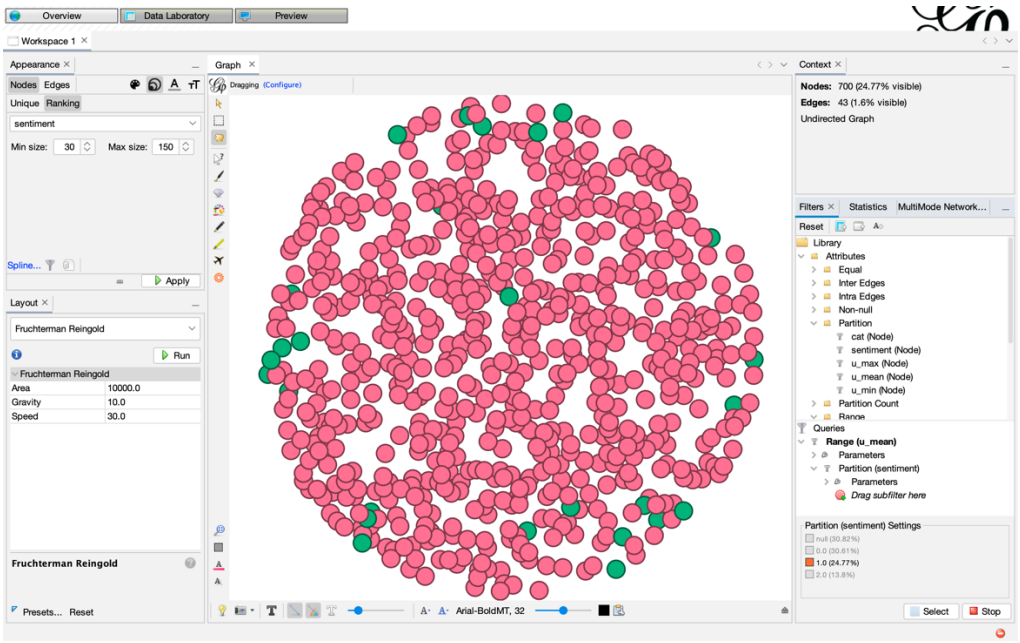


Figure 11

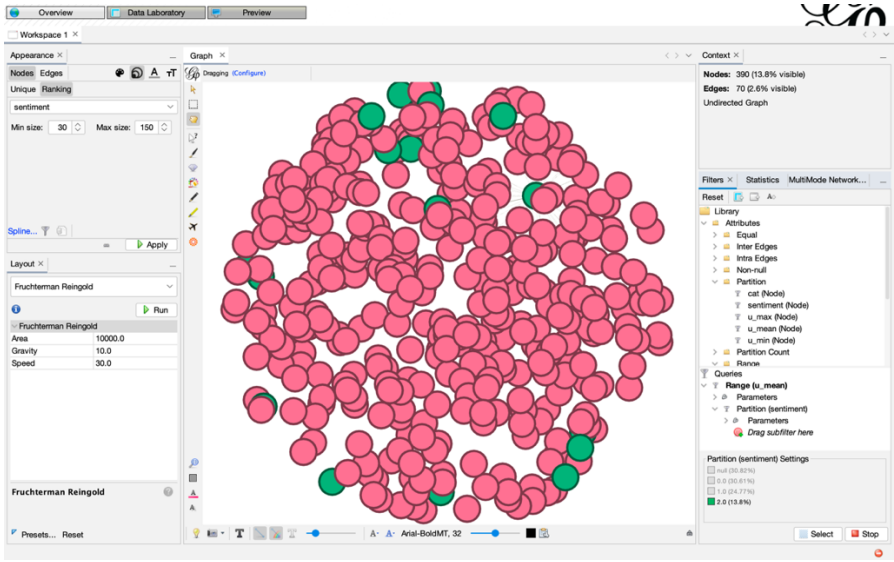


Figure 12

Masking Operator

Masking operator can mask all edges but the ones you want. We have 3 filters here one after another as you can see in figure 13

1. Mask all sources
2. Where sentiment is 2
3. Where u._mean is 1

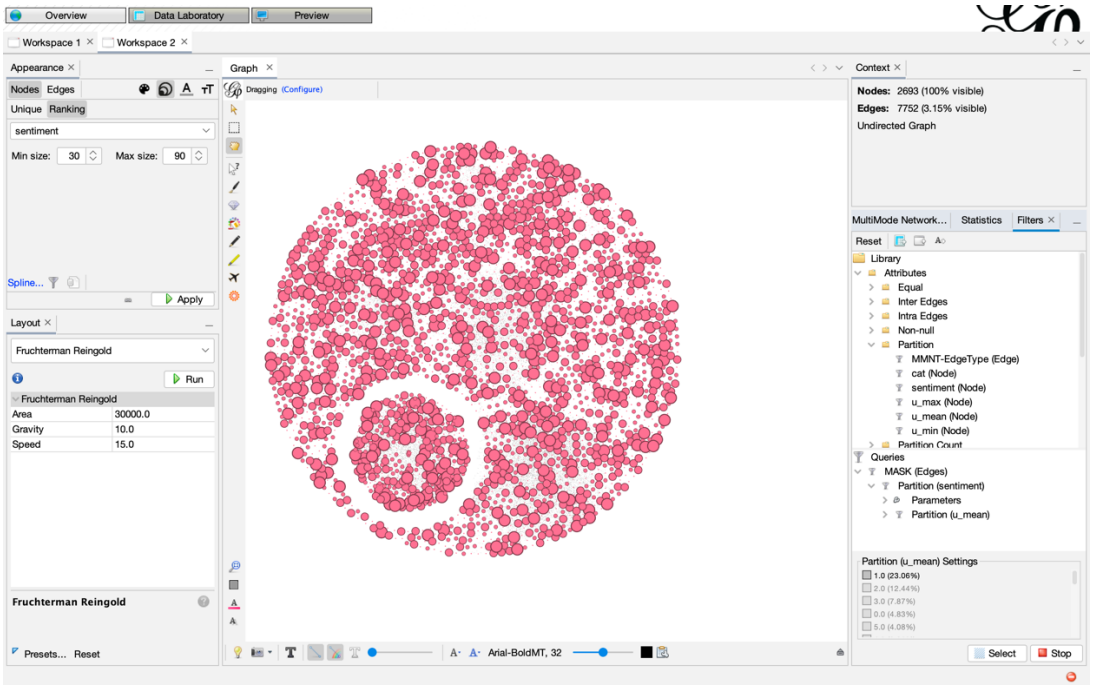


Figure 13

Towards the end we applied **Edge weight filter** which displays strong bindings. In Figure 14 and 15 we can see some strong connections.

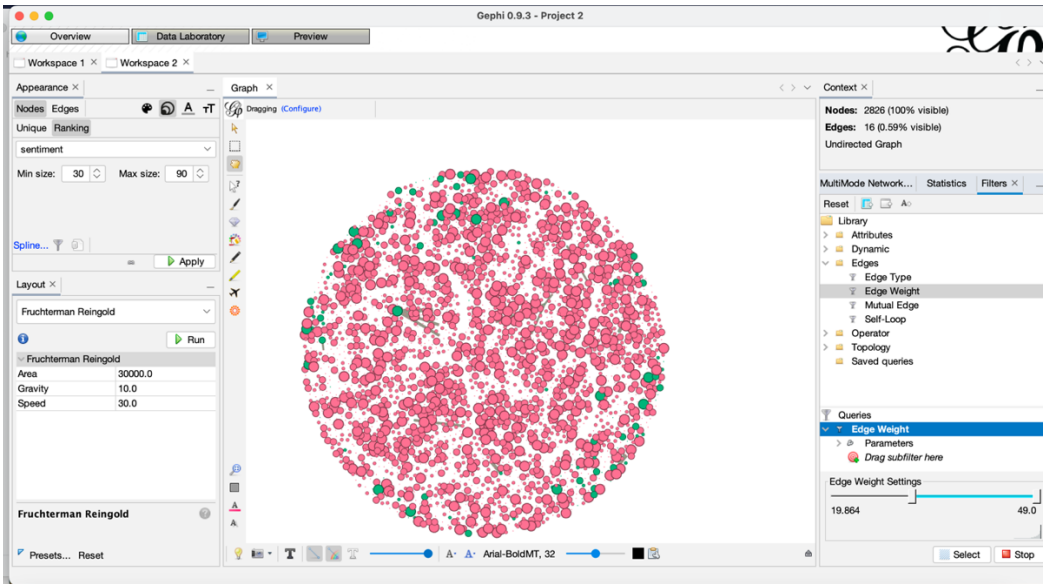


Figure 14

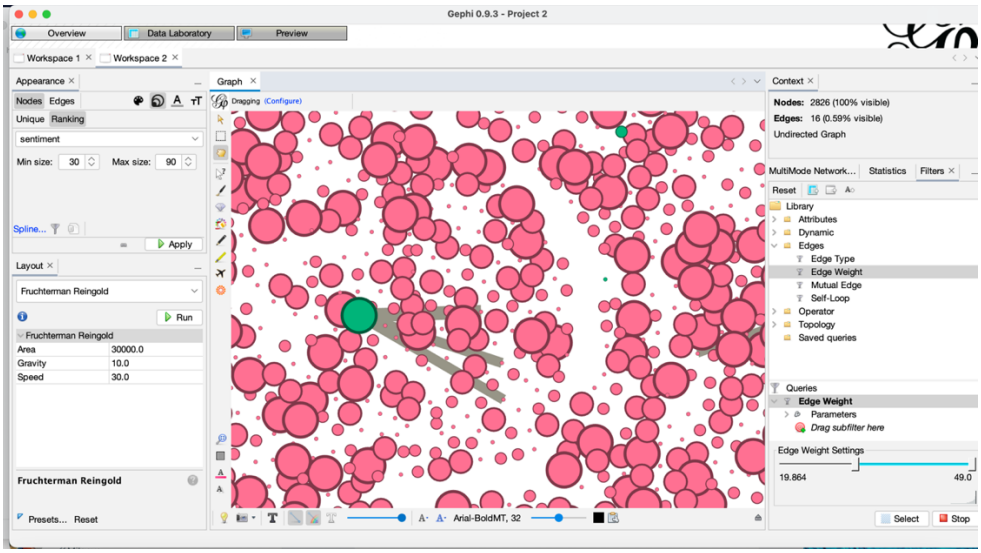


Figure 15 (zoomed in)

x----x