Gowtham Vuppala
gvuppala@albany.edu
ICSI 432 Network Science
STUDENT ID: 001270914
Final Project Report

SETUP

NOTE: For the purpose of labelling using python, I changed database_5.xlsx to database_5.csv(UTF-8 encoded), because when I'm trying to load database_5.xlsx using numpy or pandas, I kept on getting errors where it keeps saying that the database_5.xlsx has characters that UTF-8 can't encode. So, I changed the extension to .csv without any compromise in the data.

PYTHON VERSION: 3.6 (**Program uses some functions which need 3.5+ to compile**)

LIBRARIES USED: **networkx, pandas, sklearn, numpy, statistics, matplotlib, collections, warnings.**

TASK 1:

1) For task 1, I completely re-labelled all the labels and roles manually which is completely different from the one I submitted for task 1.

OBSERVATIONS:

- There seems to be a lot of sessions with no harmful comments. Among the toxic sessions, there seems to be mostly single victims but multiple bullies and assistants with some exceptions.
- Also, the commenters and owners columns are swapped in the given dataset, so I fixed that one in my dataset.

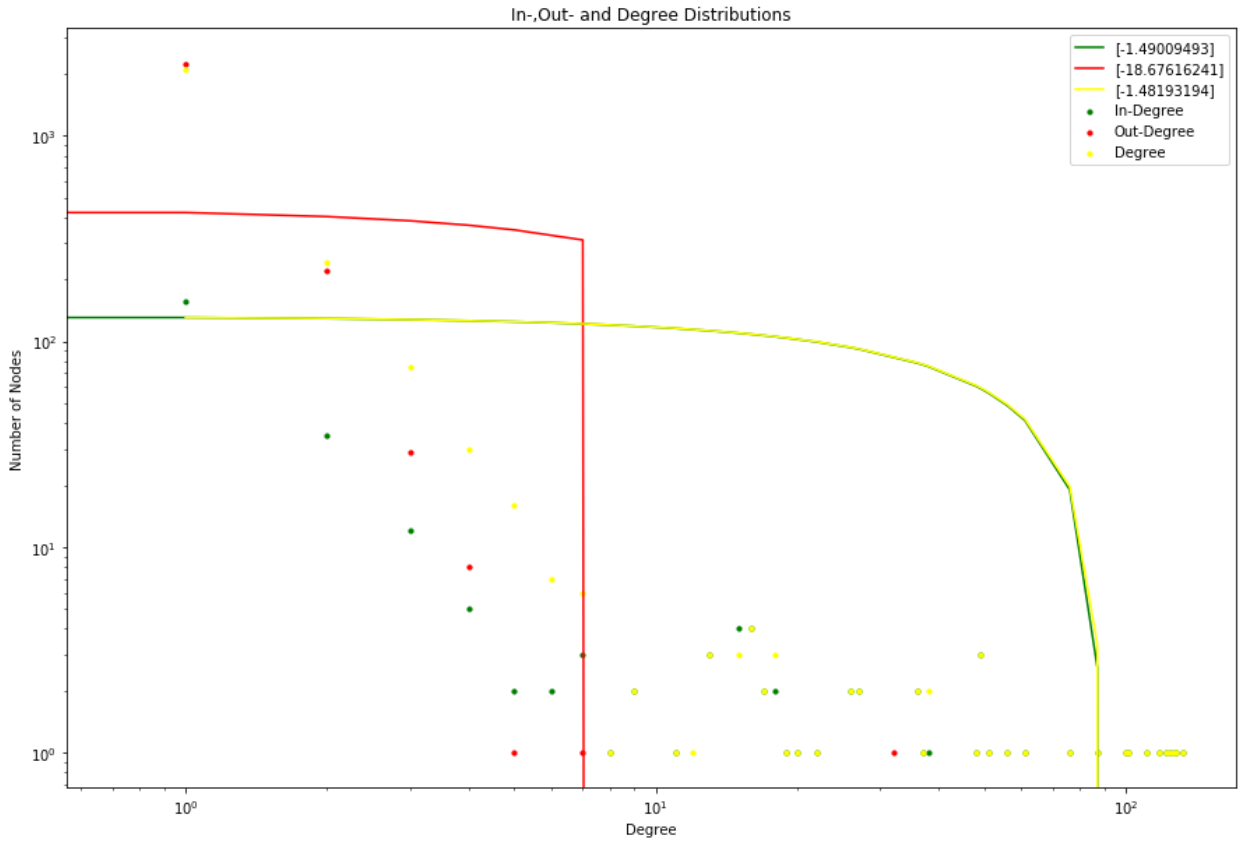
TASK 2:

GRAPH CONSTRUCTION: For constructing the graph, I first loaded owners, commenters, comments, roles into seperate lists and looped through each owner, commenter and comments and added an edge if the node u has mentioned or directly replied to v in the context of media session with the specified role.
Time Complexity: O(n^3) for creating the graph

a) STATISTICS:
   Total number of nodes: 2551 (mean = 42.52, Standard Deviation = 39.13)
   Total number of edges: 2836(Mean = 47.27, Standard Deviation = 41.31)
   Total number of bidirectional edges: 97(Mean = 1.62, Standard Deviation = 2.64)
   Min Degree: 1 (Mean = 1.05, Standard Deviation = 0.29)
   Max Degree: 132 (Mean = 41.55, Standard Deviation = 39.10)
   Total Degree: 141.9730993857119 (Mean = 2.37, Standard Deviation = 0.60)
   In-Degree: 1.183109161547599 (Mean = 1.18, Standard Deviation = 0.30)

Out-Degree: 1.183109161547599(Mean = 1.18, Standard Deviation = 0.298)
OBSERVATIONS: In the data, there seems to be a lot of sessions with no bully-victim pairs, this is the reason there are very few bidirectional edges. Also, in a lot of media sessions, lot of nodes only interact with the owner resulting in a lower min degree.

b)



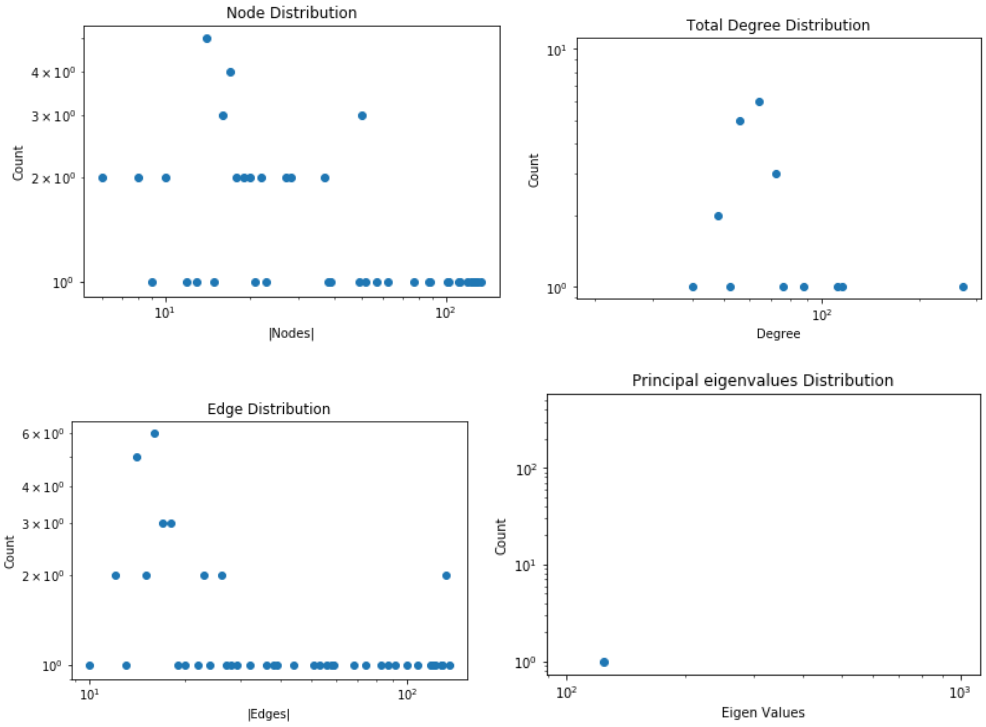
Coefficient for InDegree: [-1.49]
Coefficient for Out-degree: [-18.68]
Coefficient for Total Degree:[-1.48]

For calculating the coefficients and drawing the best fitting regression line(s), I used sklearn libraries LinearRegression() function.
REPORT:

1) The LinearRegression() couldn't find the best fitting curve for both In_degree and degree because, the function LinearRegression() uses MSE to minimize the error between the two features, but in our data we don't have well labelled data. Due to this, the regression line **underfits** the data.

c) BRIDGES: There are a total of 253 bridges and 253 local bridges in the graph Gs. This means that there are a decent number of nodes (mostly the owner of the sessions) if removed increases the components of Gs. Since, most of the commentators reply the owner of the session, removing the owner will disconnect the undirected graph.

e)



a) Node Distribution, b) Total Degree Distribution c) Edge Distribution d) Principal eigenvalues distribution

OBSERVATIONS:

1) Both the node and edge distribution have a lot of similar data points.

2) The plot for principal eigenvalues is incorrect due to the fact that the function that calculates the eigenvalues(numpy.linalg.eigs) returns output which contains types complex and imaginary which matplotlib couldn't plot. But accurate principal eigenvalues for each graph in Gs can be viewed in the code.

TASK 3

a) For creating node x feature matrix for a single graph, I used networkx functions to create two seperate lists: one list with number of edges in node i's egonet and the other one with number of neighbors of ego(i). Later, I merged the two lists along with the nodes of the graph to create the feature matrix(n x 3). Repeated the process for all the graphs in Gs.

b) For creating single feature vector for each graph in Gs, I took the mean, median and SD of each feature individually and then stored them together as a single vector.

CONCLUSION:

Overall, few of the tasks couldn't come out as they intended due to bad data. For example, the LinearRegression() function couldn't find the best fitting line for the in, out and total degrees because of bad features. Also, a lot of sessions have no bullies, marking them as useless for bullying detection. In conclusion, more accurate analysis could have been made if the dataset had more examples of cyberbullying and less of normal sessions. Other than that the remaining tasks were accurate.