Complex Networks - DATA.ML.430 - Tampere University
Project Final Report - Part III

Team name: ComplexNetworksGroup
Team members:
Georgios Gerasimos Leventopoulos, Muhammad Zunair, Haris Khan
Submission Date: 25/04/2022
Final presentation slides: here
The dataset that we used: here

PART III 1.
Project Description:

This network is a Facebook social network. A social friendship network extracted from Facebook consists of people (nodes) with edges representing friendship ties.
The vertex type is “Person” and the “Edge Type” is Friendship/social relationship between two people. The format is “undirected” and the edge weights are “unweighted”.
The choice of this network is motivated by the fact that social media takes a big part of our lives. We believe as a team that social media usage is increasing day by day, and it is very important to understand people's relationships. Facebook is a very famous social media platform, so that is why we think analyzing this network and comparing it with another type of graph will help us to see the differences between a real word network and a randomly generated network**.** We calculated some extra statistics from the network that were also included in the dataset. We did not create our own network dataset, instead, we used an existing one from here. The file that we used is an edge list and it is called “socfb-nips-ego.edges” (we deleted the first two lines of the file) and we read the file using this command “G = nx.read_edgelist("socfb-nips-ego.edges")”. Our network contains 2.9K nodes and 3K edges.

Similar to our networks that have been used for similar research accomplishments:

(a)
Similar to my Networks have been used on covid 19 tweets. Specifically, the **purpose** of this dataset, dedicated data gathering started from March 11th yielding over 4 million tweets a day that was related to **covid 19 tweets.** A lot of collaborators seem to provide new tweets from a variety of languages. In this file, there were also included hashtags, mentions, emojis, and also their frequencies. Later they also added the location of each tweet. In addition to that, they created separate .csv files for the original tweets and for the retweets. Some general statistics per day are included for both datasets. The whole project on Github can be found here. Some general statistics per day are also included in some of the datasets. I think that there are a lot of **unexplored avenues** in this network. For example, things that can be added in the future, such as the gender and age of people that are making the tweets and also the duration of time that these people have their Twitter accounts. Also, we can also maybe check for these accounts how many followers they do have and how many people are following these accounts.

(b)

Furthermore, another similar network is this one from Stanford University, which is about **Social circles on Facebook**. This dataset contains 4039 nodes and 88234 edges and the data was collected from survey participants using facebook.com. This dataset contains profiles, circles, and ego networks. and the purpose of this dataset is dataset is to find similarities between two users on Facebook. This Facebook data has been anonymized by replacing the Facebook-internal ids for each user with a new value. Also, some data from the user similarities have been “encrypted” for privacy reasons. For example, where the original dataset may have contained a feature "political=Democratic Party", the new data would simply contain "political=anonymized feature 1". Thus, using the anonymized data it is possible to determine whether two users have the same political affiliations, but not what their individual political affiliations represent. There are also similar networks to that, for **Twitter** users and for **Google+** users. These datasets also contain statistics such as Nodes, Edges, Nodes in largest WWC, Edges in largest WWC, Nodes in largest SCC, edges in largest SCC, average clustering coefficient, number of triangles, diameter (longest and shortest path), a fraction of closed triangles, 90-percentile effective diameter. The unexplored avenues for this network are to find more statistics about the network and more similarities about its individual user of the network.

(c)
In addition, there is this dataset that is about the top 100 **most-followed accounts on Instagram**. The avenue that is unexplored for this dataset is categorizing the top 100 accounts by category/profession of each individual account. For instance, find the top 100 most-followed accounts of football players, top 100 accounts of actors, top 100 accounts of singers, and others.

(d)
Furthermore, another similar project is the **1.7 Billion Reddit Comments** that you can find here. This social media dataset features 1.7 billion JSON objects along with their corresponding comments, authors, scores, subreddits, and positions in the comment tree. Users can also find other fields to look into if they use Reddit’s API. This dataset is over 1 terabyte uncompressed, so this would be best for larger research projects. Total Comments: 53,851,542 were used for this project. This project has a lot of potential to grow and a lot of people helped in the contribution of this dataset.

(e)
We also find another similar dataset about **Twitter friends** here. This dataset is an extract of a wider database aimed at collecting Twitter users' *friends*. Basically to see what accounts a user follows. In this way, we can study users' interests through who they follow and their connection to the hashtag they've used. The content of this dataset is a list of users’ information, and every user is represented by a JSON object, which contains the user’s information. Specifically, every JSON object contains the following things:

1) An avatar, which holds the URL to the profile picture of the user
2) The followerCount, which shows the number of followers of the user
3) friendsCounter: that counts the number of people following this user.
4) friendName: stores the *name* of the user
5) id: user ID, every user maps to a user id that is unchanged
6) friends: the list of IDs the user follows

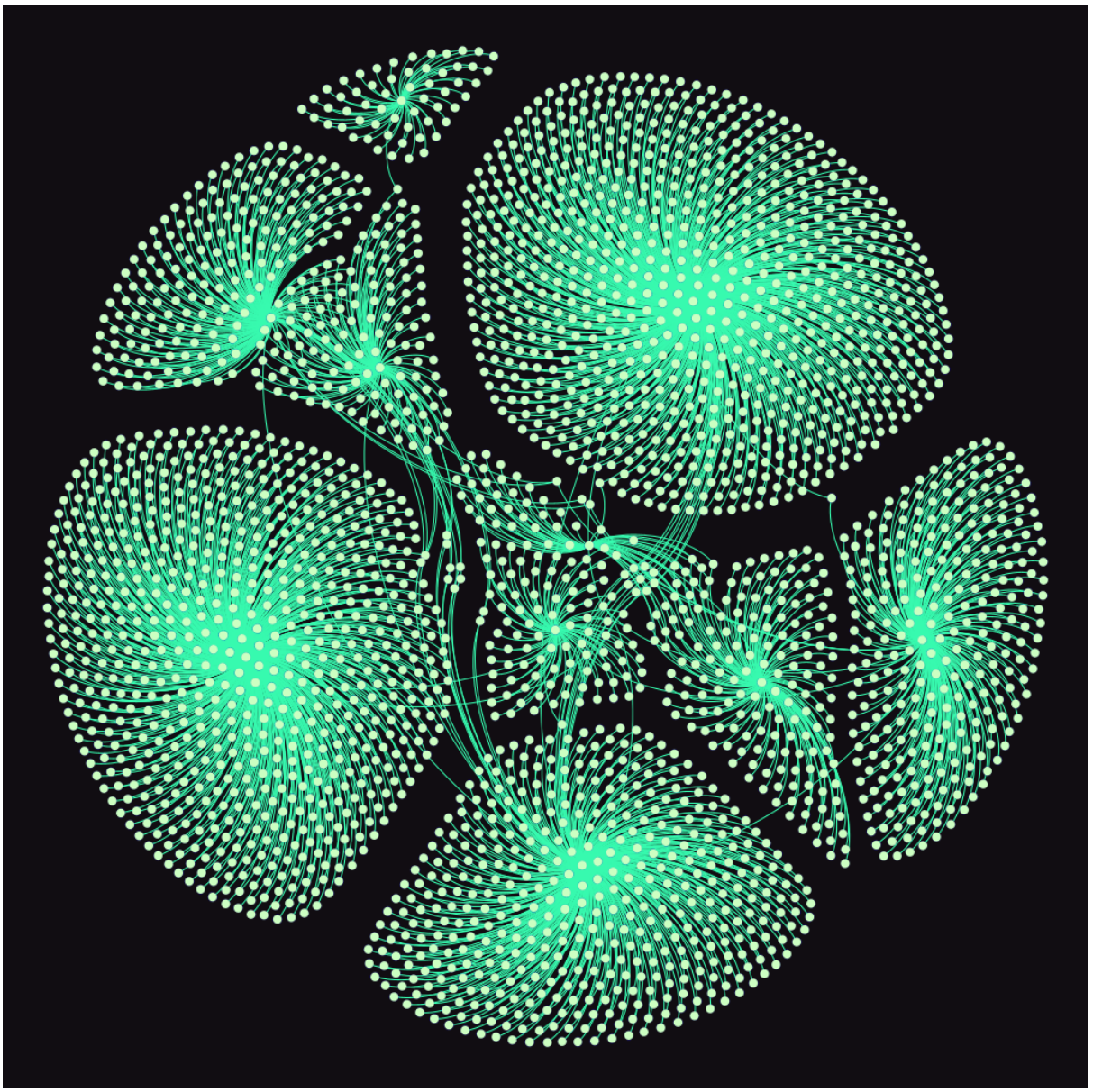
7) lang: the language declared by the user (only English in this dataset)
8) lastSeen: the timestamp of the date when this user has posted his last tweet.
9) tags: the hashtags used by the user
10) tweetID: Id of the last tweet posted by this user.

By using this dataset a researcher or research team can find stats about followers and followings, manyfold learning or unsupervised learning from friend lists,s and hashtag prediction from friend lists. These users are selected because they tweeted on Twitter *trending topics* and follow at least 100 people and are followed by at least 100 people. No public research has been done (until now) on this dataset, so the unexplored avenues of this project are more potential research on this and also checking other accounts such as accounts with at least a specific number of posts or research only on accounts that have been created after a specific date.

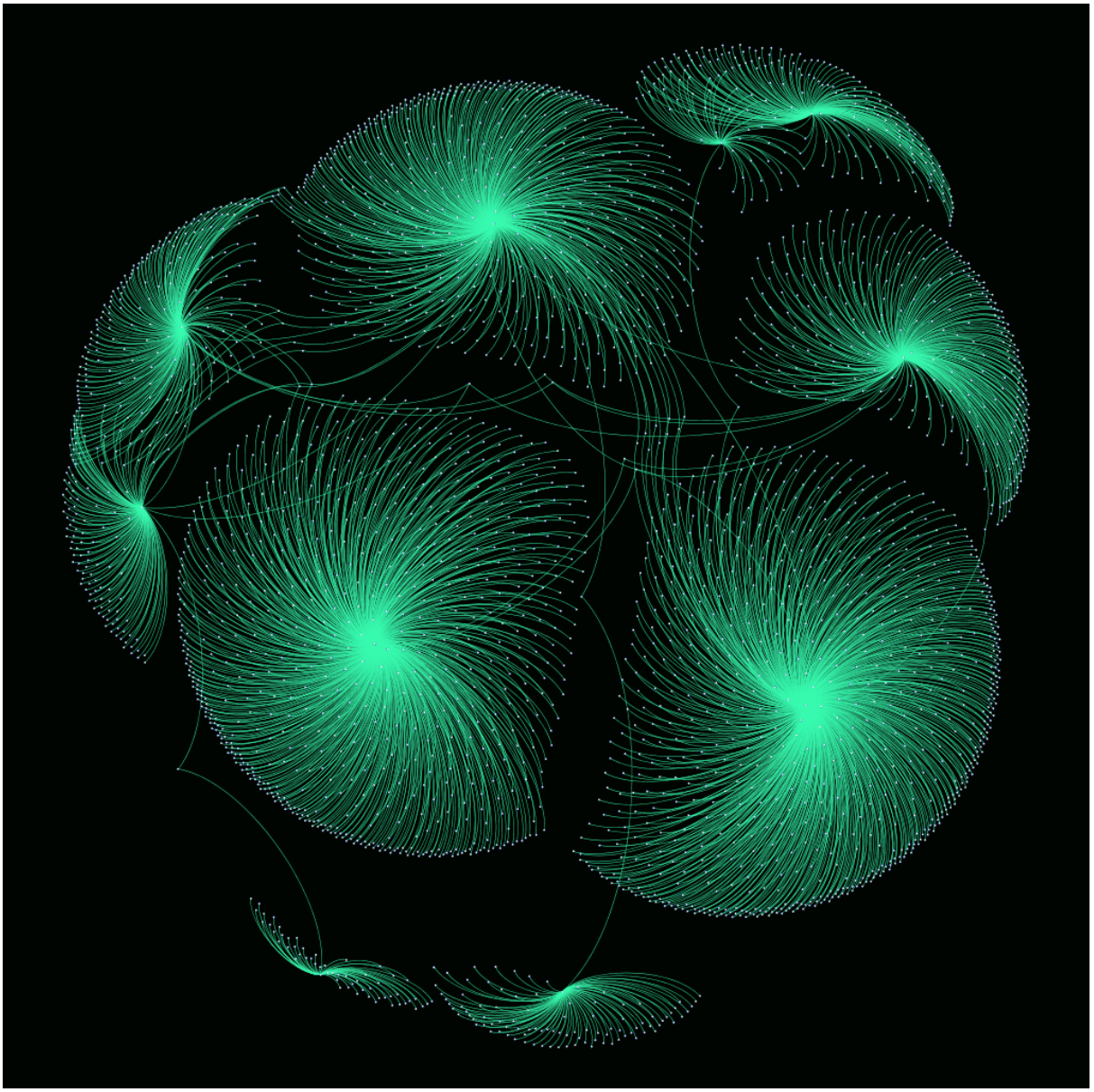
PART III - 2.

Descriptive network statistics. Degree, clustering, and shortest path distributions are presented and discussed via figures and statistics.
Network components are identified and discussed. Density.
We used “Gephi” for the visualization of the networks.

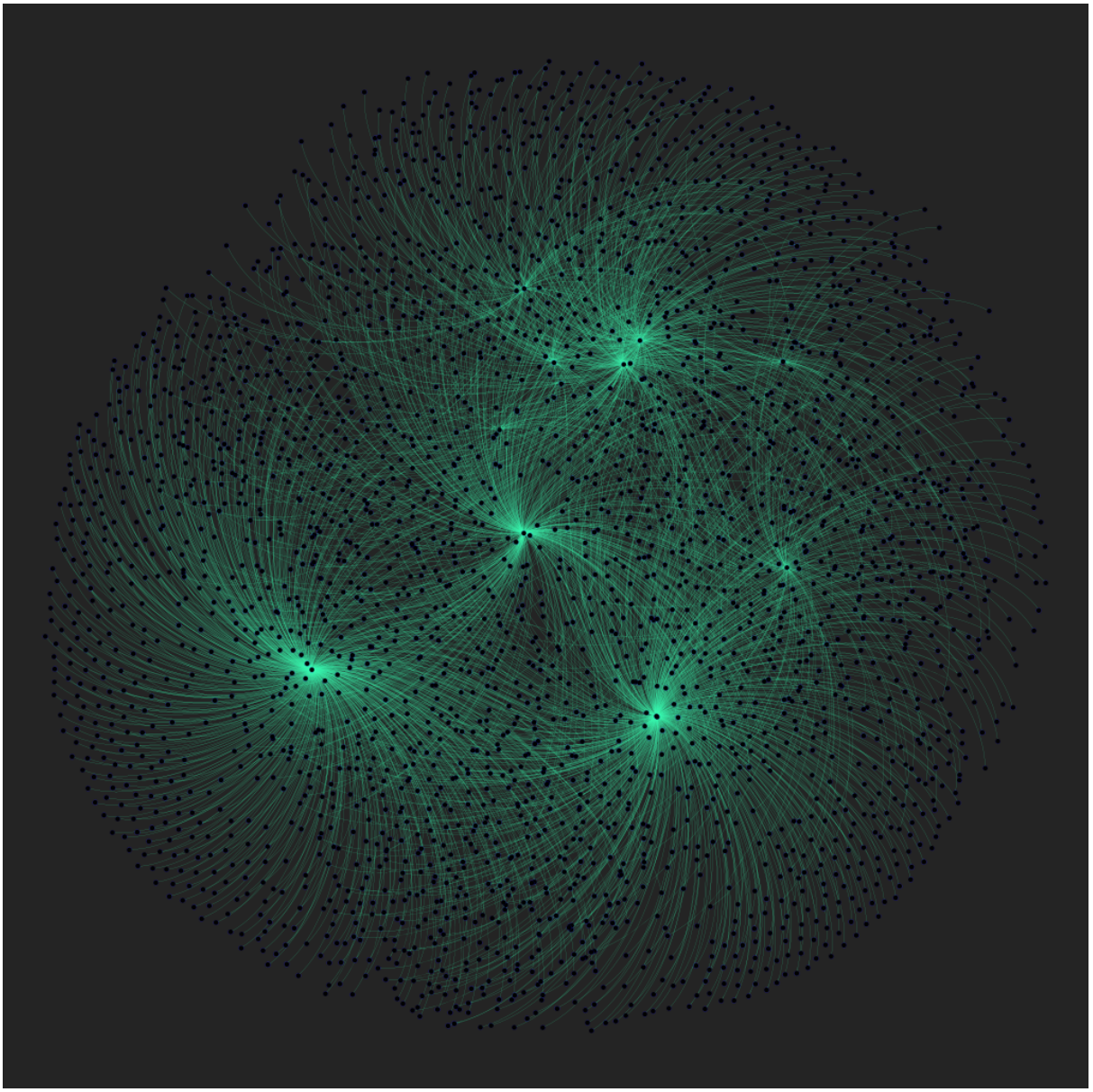
Graph VIsualization Real Network



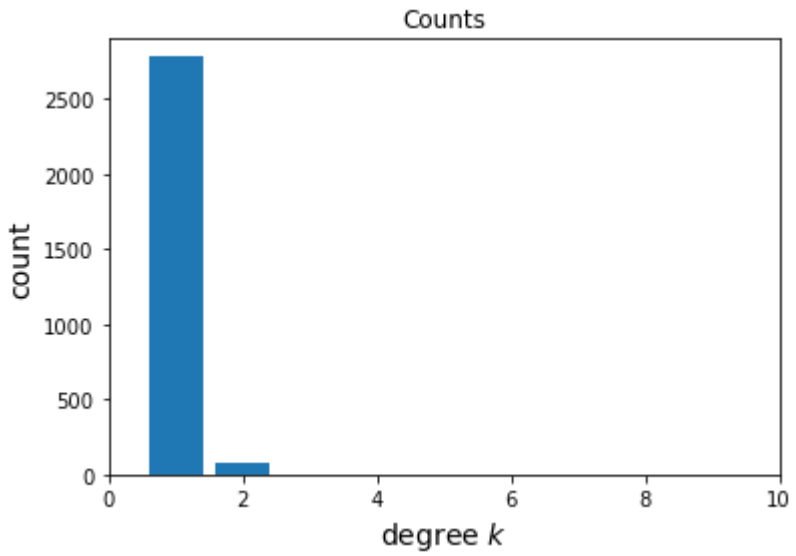
Graph Visualization for ER Model Network

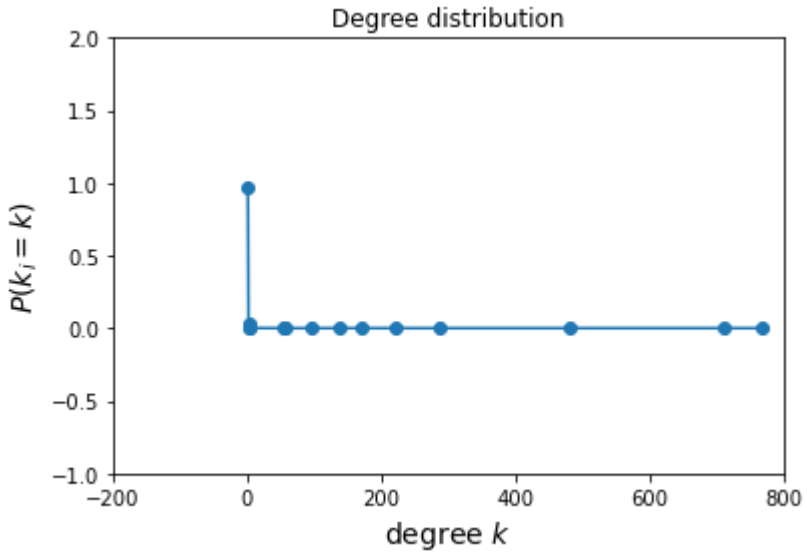


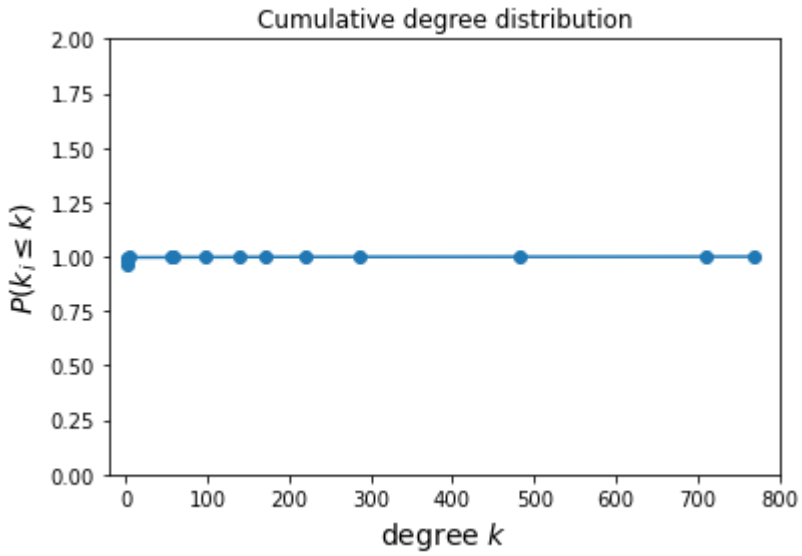
Graph Visualization BA Model Network

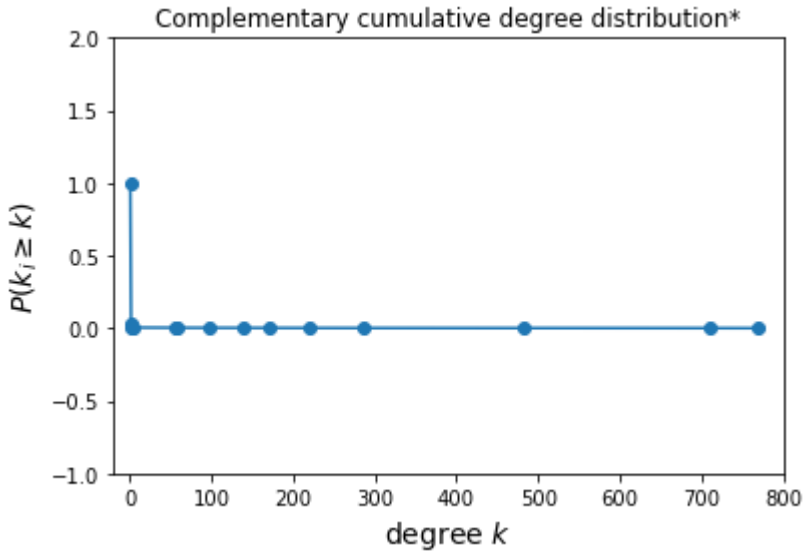


Degree Distributions and Figures for Real Facebook Network:

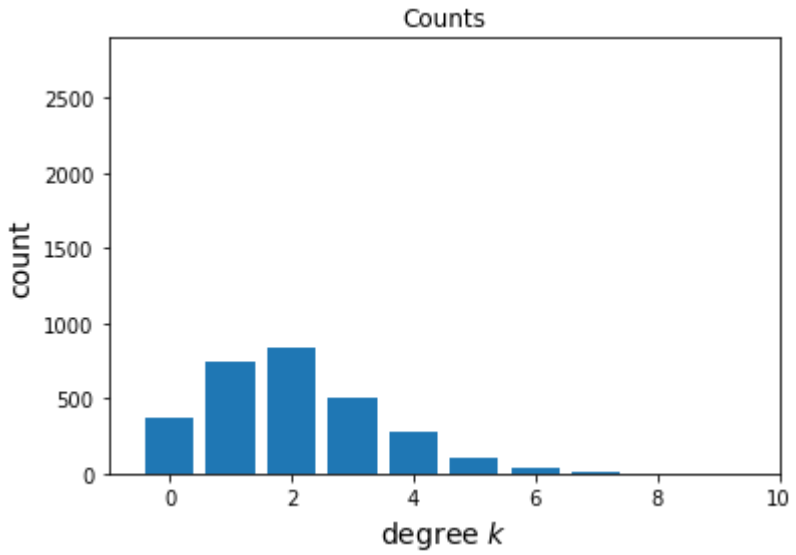


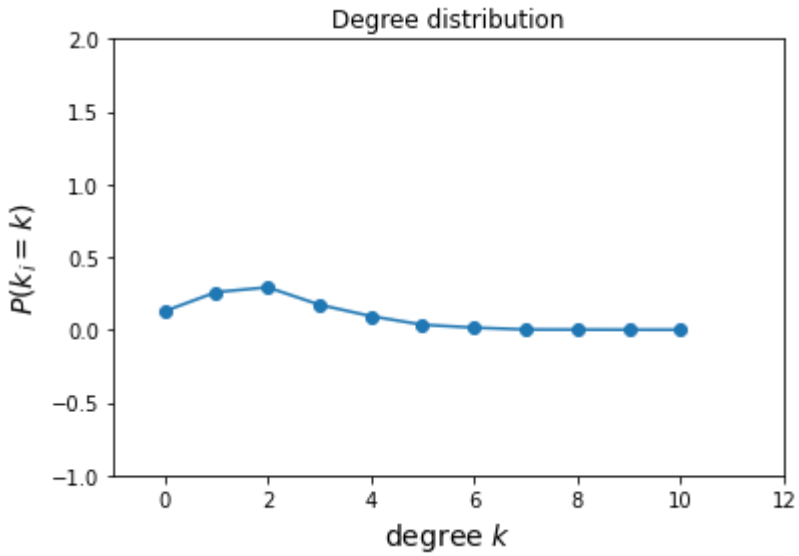


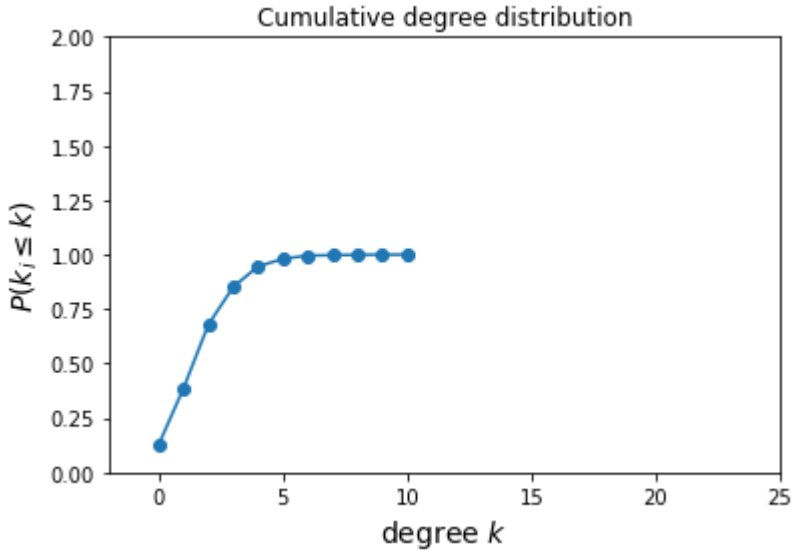

Cumulative degree distribution
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
$P(k_i \leq k)$
0
100
200
300
400
500
600
700
800
degree k

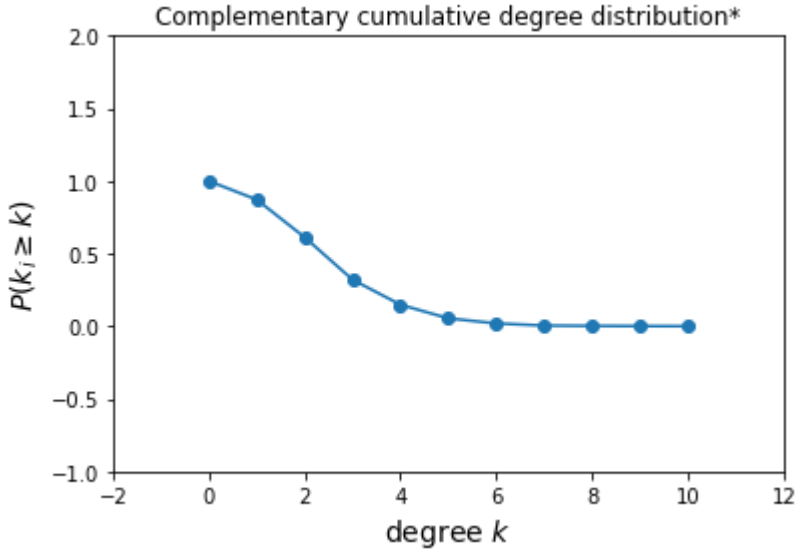

Complementary cumulative degree distribution*
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
$P(k_i \geq k)$
0
100
200
300
400
500
600
700
800
degree k

Degree Distributions and Figures for Random ER Network:

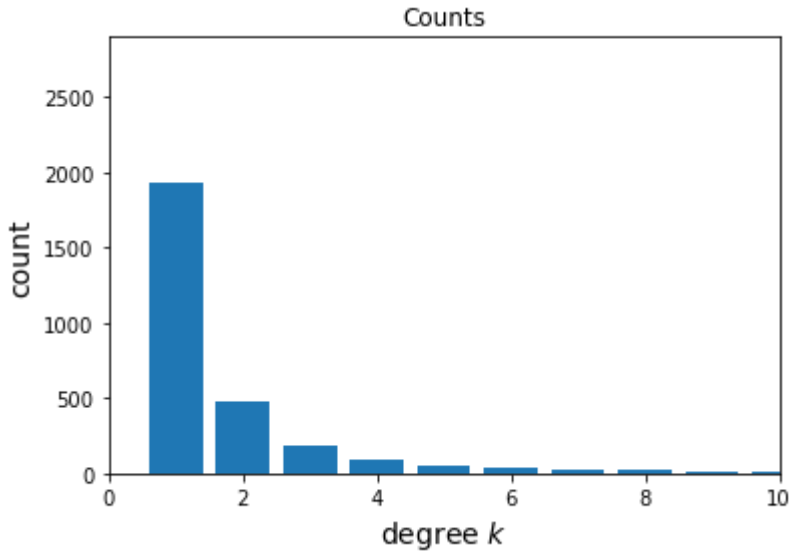


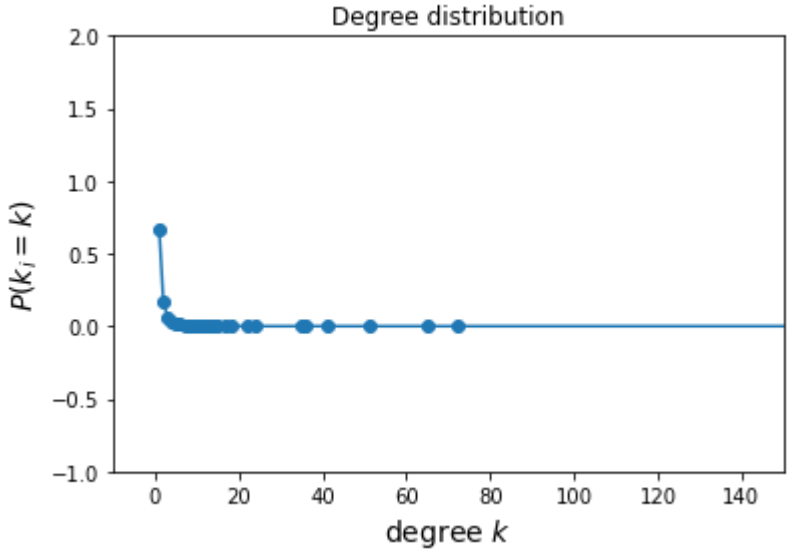


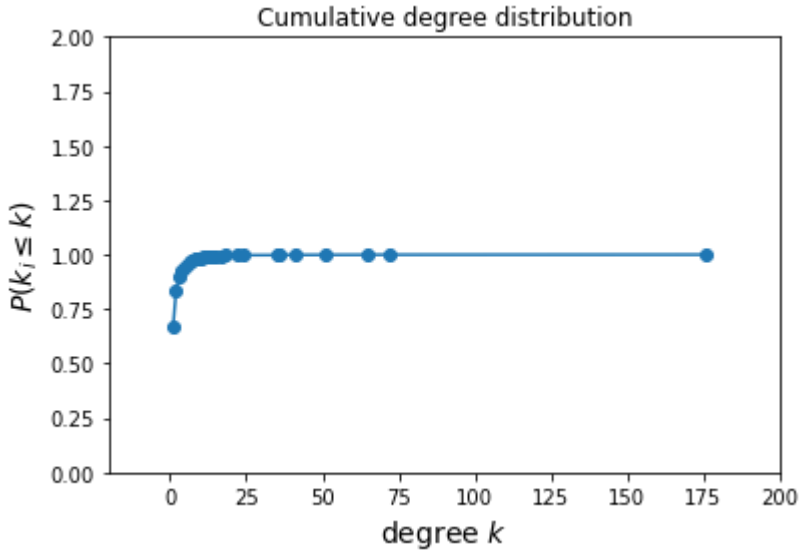

Cumulative degree distribution
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
$P(k_i \leq k)$
0
5
10
15
20
25
degree k

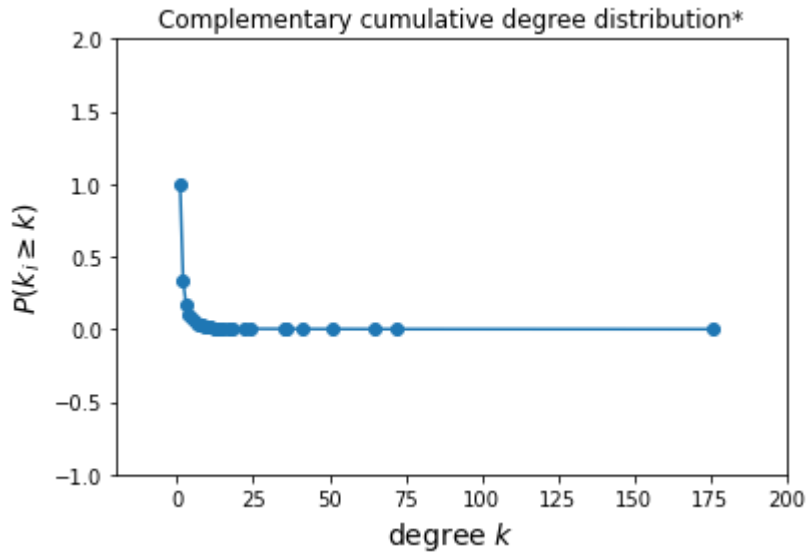

Complementary cumulative degree distribution*
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
$P(k_i \geq k)$
−2
0
2
4
6
8
10
12
degree k

Degree Distributions and Figures for Random BA Network:





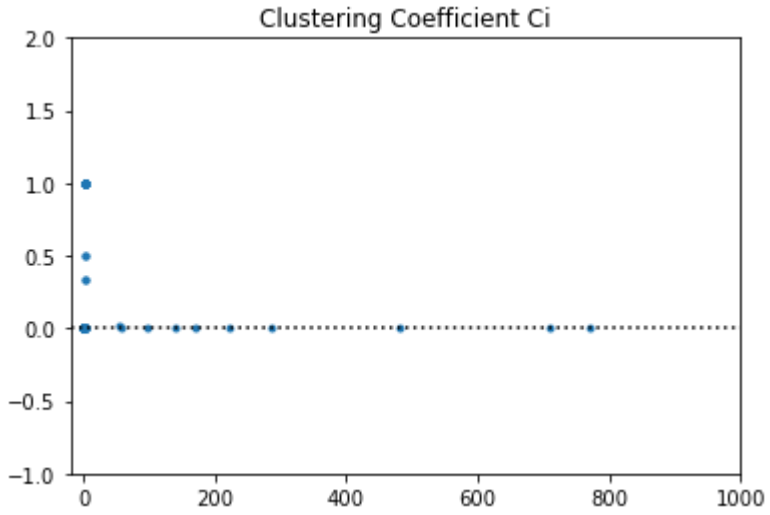




The first figure shows the degree ki of each node i ∈ V, which means how many times we see a node with degree k. The second figure shows the degree distribution which is symbolized using P(k). The degree of a node in a network is the number of connections it has to other nodes and the degree distribution is the probability distribution of these degrees over the whole network. So by counting how many nodes have each degree, we form the degree distribution. The Cumulative Degree Distribution (CDF), and the 4th diagram shows the Complementary Cumulative Degree Distribution 1-CDF(k) of the network.

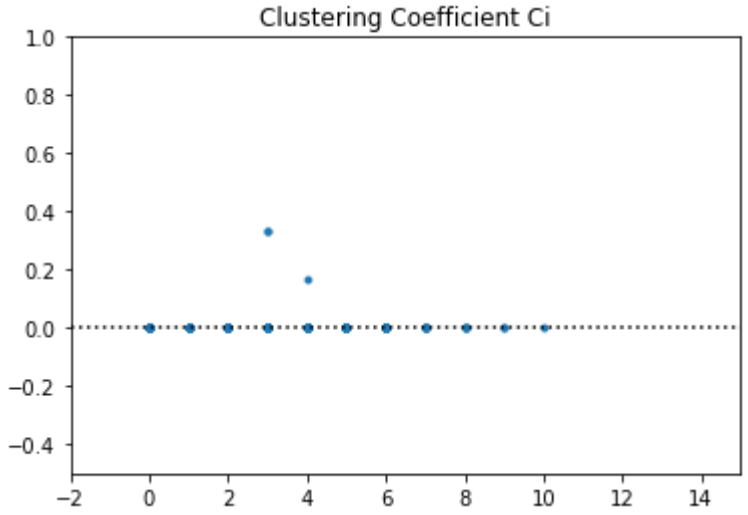
In the Facebook Real Network, the Degree Distribution figure shows that most of the nodes have degree 1 (almost all of them). So, the Facebook Real Network has more nodes with a very low degree. This might mean that in this network not many people share the same friends. On the other hand, the ER random network has a range of degrees from 0 to 7 with the majority of degrees being 2, something that is not happening in the Facebook real network. So, here there are a lot of nodes that have moderate degrees and most of them have a low degree. Additionally, for the BA Random Network, we can see that the Degree Distribution is similar to the Facebook network. the network has more nodes with a very low degree and very few with a high degree.

Clustering Distributions and Figures for Real Facebook Network:



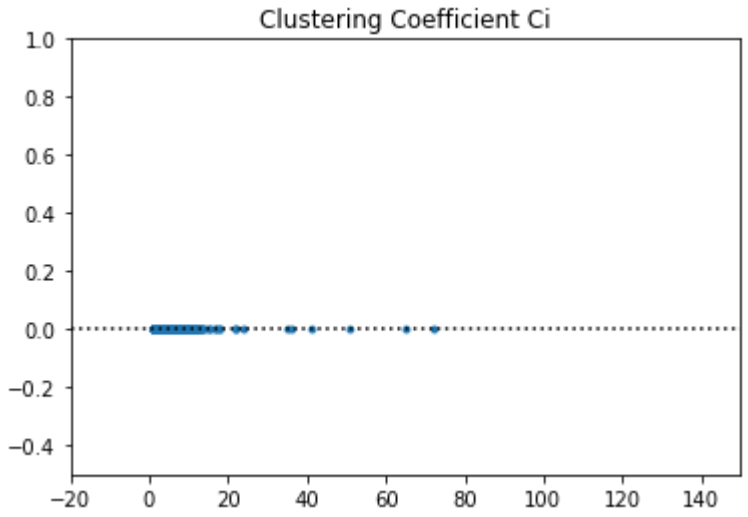
Average Clustering Coefficient <C>: 0.027247421431211827

Clustering Distributions and Figures for ER Network:



Average Clustering Coefficient <C>: 0.0002885503231763619

Clustering Distributions and Figures for BA Network:

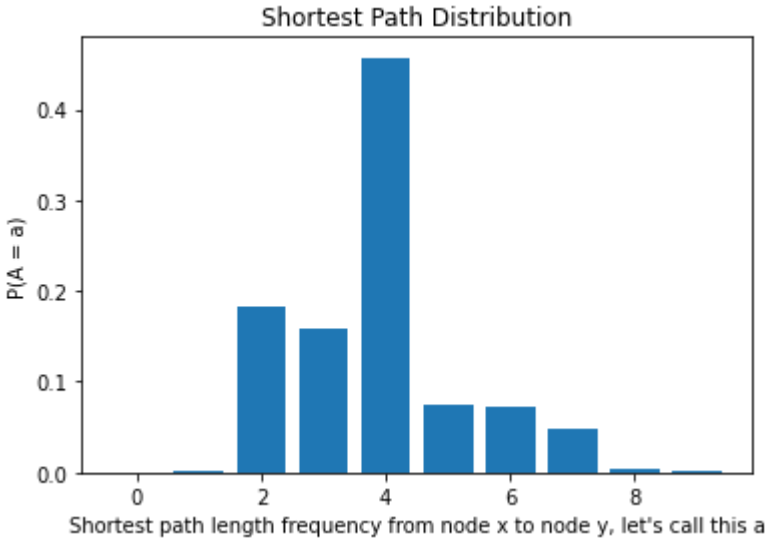


Average Clustering Coefficient <C>: 0.0

In graph theory, a clustering coefficient is a measure of the degree to which nodes in a graph tend to cluster together. The clustering coefficient of a network plays a vital role to influence the behavior of the link prediction technique. We see a relationship between the degree and clustering in real-world networks.

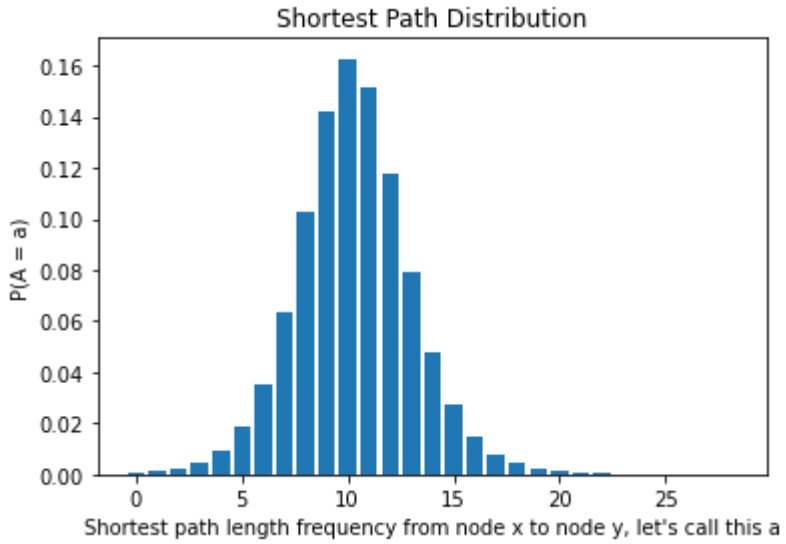
According to G (N, p) - Random Network Model, the clustering coefficient for a node has no relationship between the actual node degree and its clustering coefficient. In the BA Clustering Coefficient, we have the average clustering coefficient is zero. On the other hand, on the real-world Facebook Network, we see a relationship between the degree and clustering, so the clustering coefficient is not really well explained by the Random Network Model G (N , p).

Shortest Path Distributions and Figures for Facebook Network:



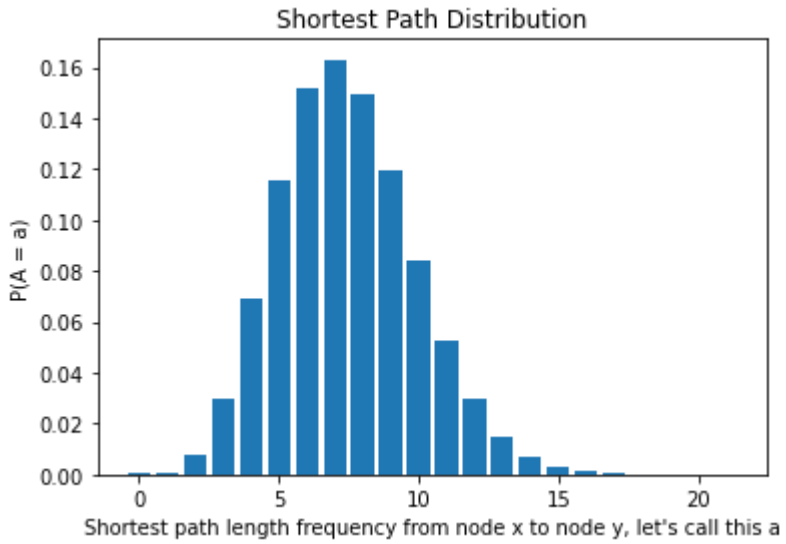
Using this method “nx.average_shortest_path_length()” we take the average shortest path for a real network <d> real 3.8674212512485524

Shortest Path Distributions and Figures for ER Network:



The expected average distance in a G(N, p) network is <d> = log(N)/log(k)
<d> of G(N, p) 8.748454382874677

Shortest Path Distributions and Figures for BA Network:



The expected average distance in a G(N, p) network is <d> = log(N)/log(k)
<d> of G(N, p) 11.501601625603493

For the Facebook real network, the probability of taking the shortest path of length 4 is the highest, which is about 0.45 the other 0.55 are other probabilities. The second frequent

shortest path length is 2 which gives the probability of 0.18. Additionally, the third is 0.15.

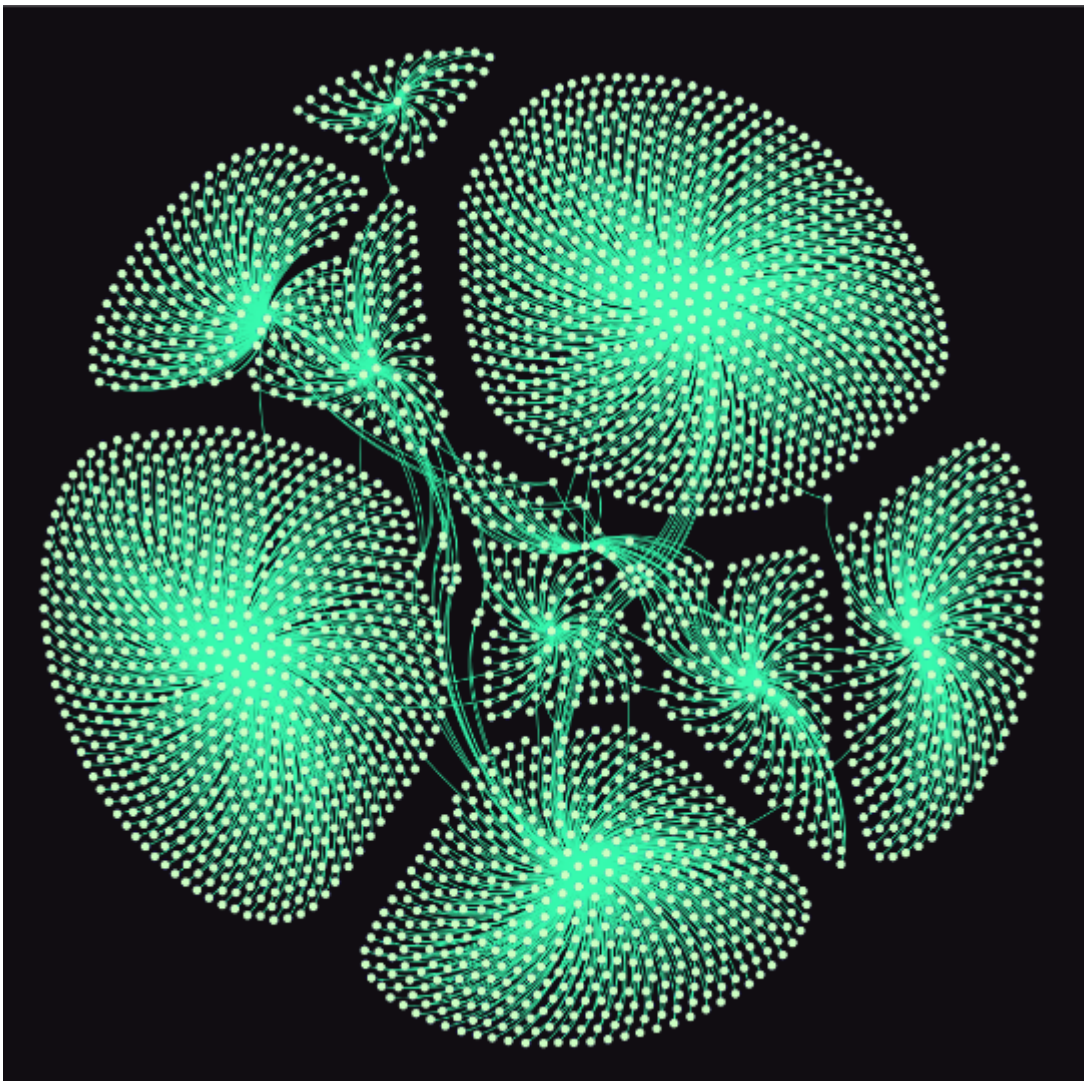
For the Random networks ER and BA, the shortest path distribution is similar to the “normal distribution” and it is totally different than the real Facebook network distribution.

The distribution is similar to the normal distribution and that means that is symmetric about the mean (half the values fall below the mean and half above the mean).

The probability for the shortest path length to be 10 is the highest probability in ER random network and for the BA random network, it is 7.

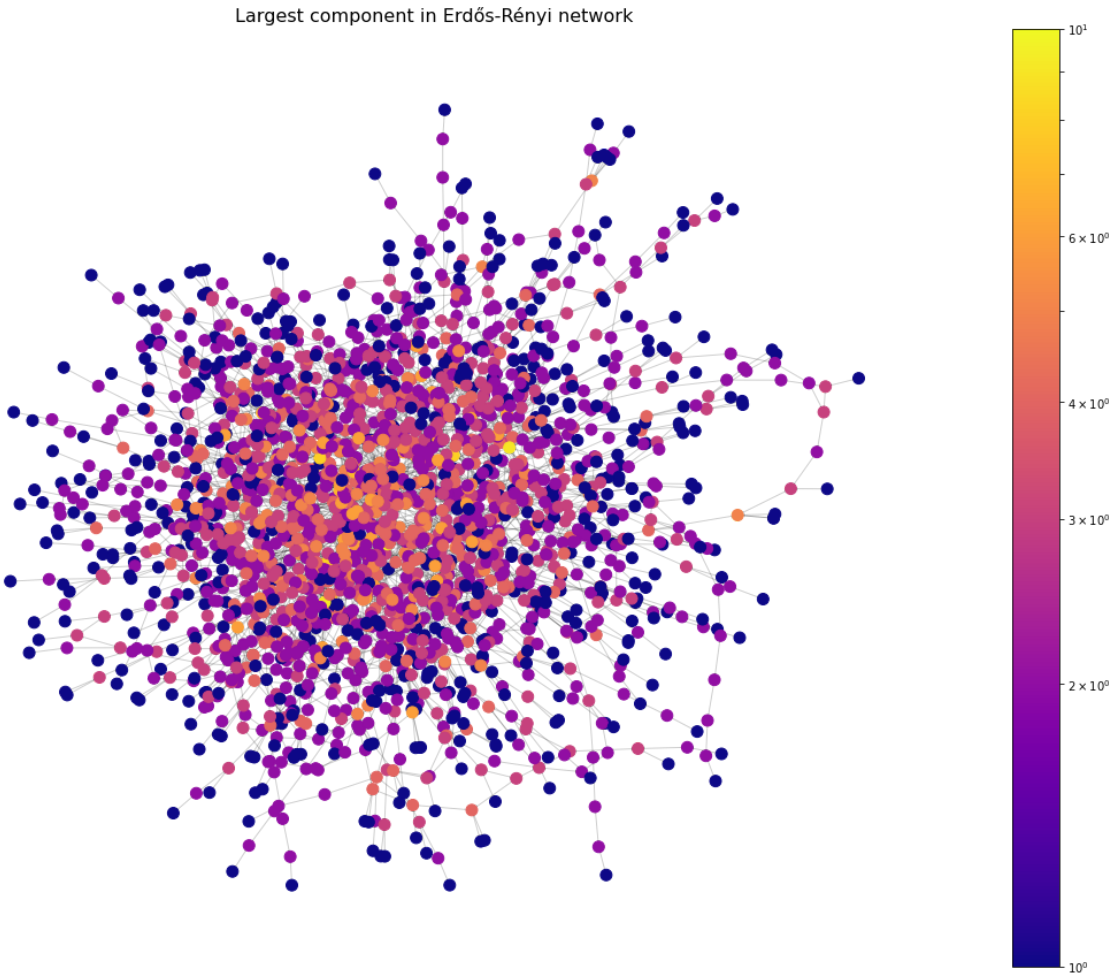
For the implementation, we run Dijkstra Algorithm starting from every node to every other node to find the shortest path from every node to every other node. Then we counted the frequencies of every shortest path length and then we calculated the probability of the shortest path for every shortest path length.

Network components are identified and discussed for Facebook Network:



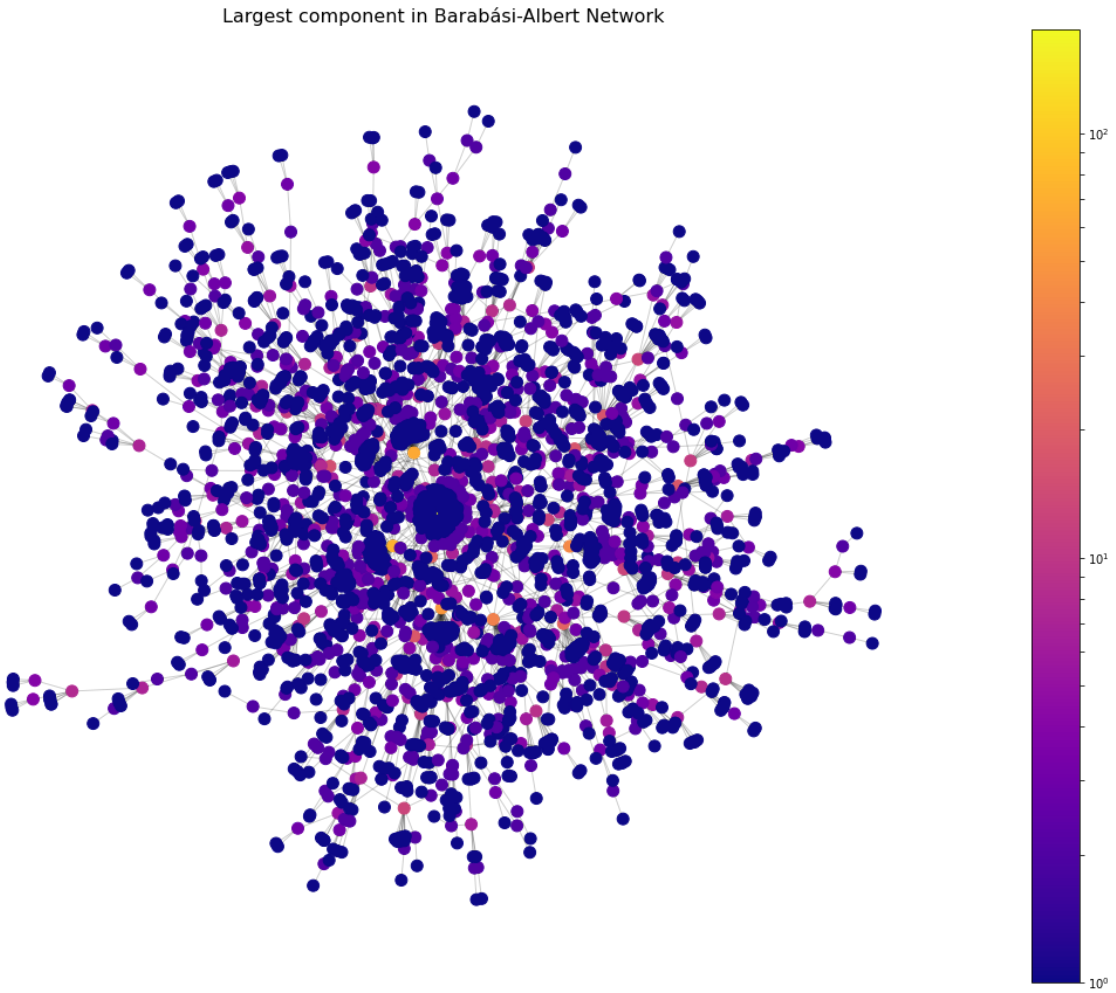
Graph Components: The graph is connected and there is only one connected component

Network components are identified and discussed for ER Network:



The graph is not connected. The number of connected components is 445
The difference between the Facebook Real Network and this one is that the graph is not connected, but on the Facebook network, it is. Here, the graph is composed of many different components and the number of connected components is 445. This is a subgraph of the largest component in the graph with 2326 nodes and 2821 edges

Network components are identified and discussed for BA Network:



The graph is connected and the number of connected components is 1

In BA Network the graph is connected and the number of connected components is 1 (the whole graph) The largest component has 2888 nodes and 2887 edges. The same thing is happening on Facebook's real network and not happening on ER Random Network.

Density

The **Density** for the real Facebook Network is 0.0007150690793671507
The **Density** for the random ER Network is 0.0007049943053539268
The **Density** for the random BA Network is 0.0006925207756232687

Network density “ρ” is the total number of links over the maximum number of links. As we can see, the Density of Facebook's real network is bigger than the Density of the random ER network. The density of the ER random network is bigger than the Density of the BA random network.

We also added some extra statistics. The Facebook real network statistics were included on the website where we also collected the dataset.

Extra statistics for Facebook Network:

Nodes: 2.9K
Edges: 3K
Density: 0.0007150690793671507
Maximum degree: 769
Minimum degree: 1
Average degree: 2
Assortativity: -0.6682140067239859
Number of triangles: 273
Average number of triangles: 0
Maximum number of triangles: 52
Average clustering coefficient: 0.0272474
Fraction of closed triangles: 0.00035938

Extra statistics for ER Network:

Nodes: 2.9K
Edges: 3K
Density: 0.0007049943053539268
Maximum degree: 10
Minimum degree: 10
Average degree: 2
Assortativity: 0.030511546098633806
Number of triangles: 3
Average number of triangles: 0.001038781163434903
Maximum number of triangles: 1
Average clustering coefficient: 0.0002885503231763619
Fraction of closed triangles: 0.0005022601707684581

Extra statistics for BA Network:

Nodes: 2.9K
Edges: 3K
Density: 0.0006925207756232687
Maximum degree: 176
Minimum degree: 1
Average degree: 1.9993074792243768
Assortativity: -0.07223392425340028
Number of triangles: 0
Average number of triangles: 0
Maximum number of triangles: 52
Average clustering coefficient: 0.0
Fraction of closed triangles: 0

PART III - 3

3a. Community Discovery

For the 3a. Community Discovery part, we used the following community detection methods: K-clique, DEMON, Louvain, Infomap, Fast Greedy, Girvan, and Newman

K-clique

A k-clique community is the union of all cliques of size k that can be reached through adjacent (sharing k-1 nodes) k-cliques. We choose k = 3

Results: The total number of 3-clique communities is 3
The sizes of the communities are: 13, 52, 24
The size of the largest community is 52
The number of nodes that do not belong to any 3-clique community is 2800
The maximum number of communities a single node belongs to is 2
The number of nodes that belong to at least two communities 1
The Modularity is -0.037554911397365304

Kclique and Demon have negative modularity and the return communities and the sizes of these communities are identical. It is not a strong community structure, because we have negative modularity.

DEMON

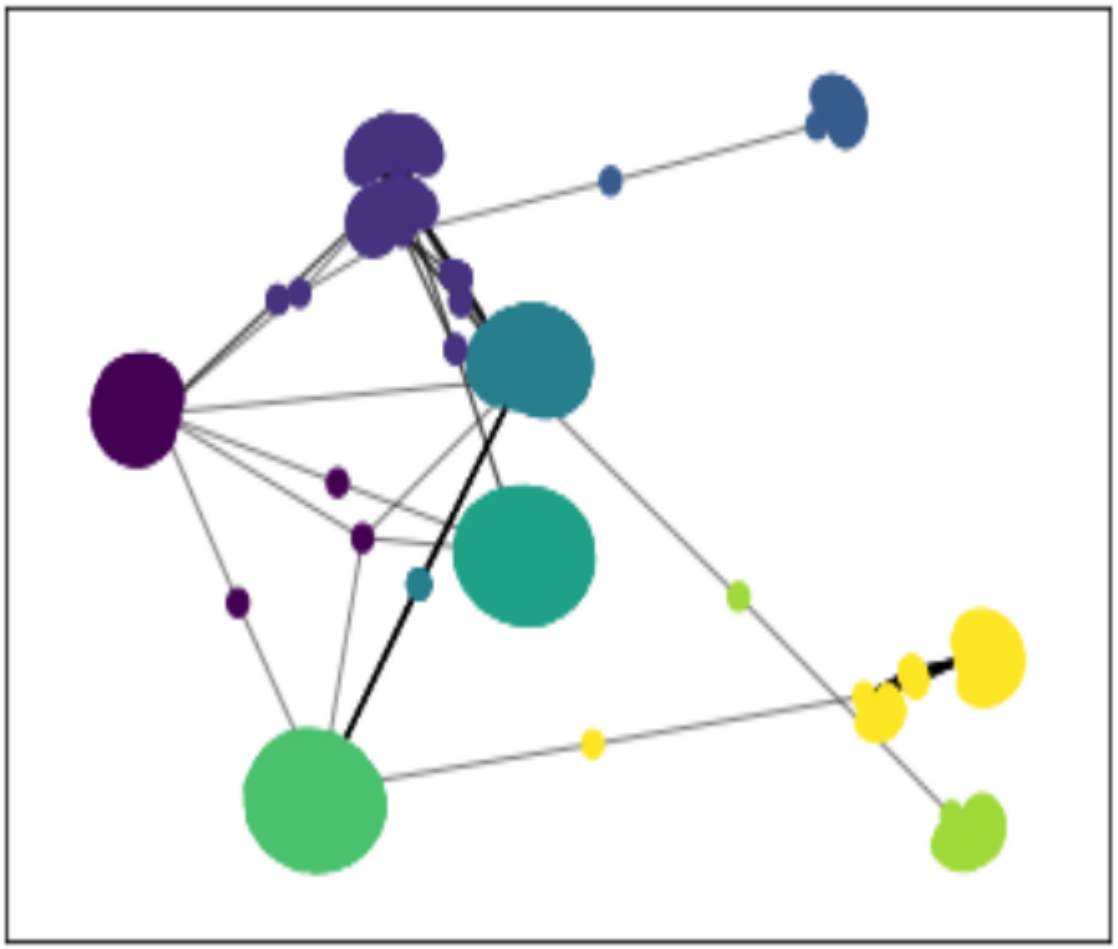
The Demon approaches the community discovery problem through the analysis of simpler structures (ego-networks).

Results:
-The total communities are 3
-The sizes of communities are: 13, 52, 24
-The length of the biggest community is 52
- The Modularity is -0.03755491139736614
-As we can see the Demon and K-clique have the same partition and modularity and the same number of communities.

Demon in this case is not a strong community structure, because the modularity is not good.

Louvain

Results:
-The number of communities is 8
-The sizes of the communities are: 284, 315, 465, 707, 757, 98, 203, 59
-The length of the biggest community is 757
- Modularity: 0.808687549359893, we made a color for every different community



The Louvain is almost identical to the fast greedy, also the length of the biggest community is the same as Fast Greedy. So far, this community discovery method has the best partition.

Infomap (https://www.mapequation.org/)

Results:
The number of communities is 3
The sizes of the communities are: 1523, 993, 372
The length of the biggest community is 1523
Modularity: 0.46727152992647863, it is not the best but it is not the worst

The size of this community is the same as K-clique and Demon, but Infomap has better modularity than these two.

Fast Greedy

Results:
The best partition found consists of the following 8 communities with sizes:
59, 98, 203, 707, 313, 757, 465, 286

The length of the biggest community is 1523
The modularity of this partition is: 0.8087217591092699
Fast Greedy is very similar to Louvain and is the best modularity so far

Girvan and Newman

Results:

The best partition found consists of the following 2 communities with sizes: 2180, 708

The length of the biggest community is 2180. The modularity of this partition is 0.3612020713540672

3e. The curiosity-driven task can be anything else you come up with. For example, you can investigate different centrality measures, and report and discuss the findings.

In this curiosity-driven task, we decided to investigate different centrality measures and report and discuss the findings. Specifically, we found about Degree Centrality,
Eigenvector Centrality, Katz Centrality, Closeness Centrality, and Betweenness Centrality. We used some of the methods that are included here .

Centrality:

Centrality is a general term for how close a node is relative to the network as a whole.

Degree Centrality:

The Degree Centrality for a node v is the fraction of nodes it is connected to. Degree centrality is a simple centrality measure that counts how many neighbors a node has.
We used the degree_centrality() method from here.
Maximum Degree Centrality in our graph is 0.26608996539792384
Minimum Degree Centrality in our graph is 0.0

Eigenvector Centrality:

Eigenvector centrality computes the centrality for a node based on the centrality of its neighbors. It is a measure of the influence of a node in a network. We used eigenvector_centrality_numpy() from here .
Maximum Eigenvector Centrality: 0.7024818331273988
Minimum Eigenvector Centrality: 2.881674340430497e-07

Katz Centrality:

We calculated Katz's Centrality using this method. Katz Centrality sais that a node is important if it is linked from other important nodes or if it is highly linked. Katz Centrality computes the centrality for a node based on the centrality of its neighbors. It is a generalization of the eigenvector centrality. If a directed network is not strongly connected, only nodes that are in strongly connected components or in the out-component of such components can have non-zero eigenvector centrality.

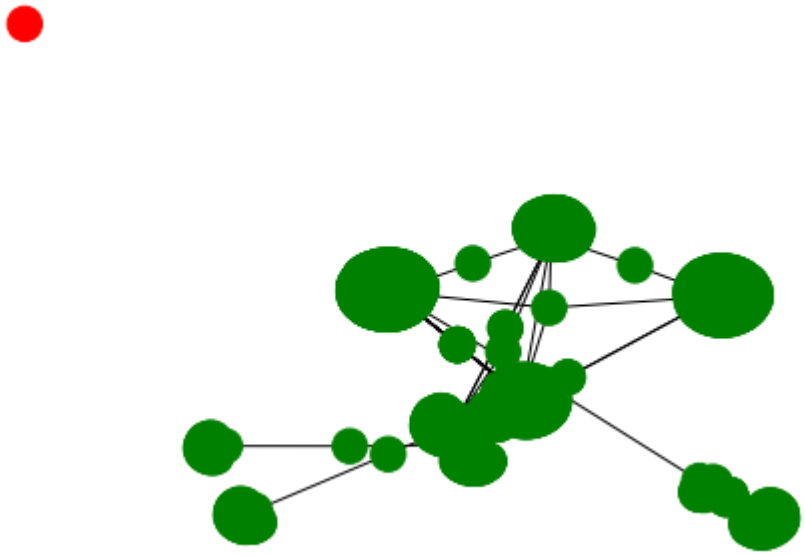
Closeness Centrality:

We used the closeness_centrality() method from here. Closeness centrality is the average length of the shortest path between the node and all other nodes in the graph. In other words, is based on the mean distance from one node to other nodes. Higher values of closeness indicate higher centrality. Nodes from small components receive a smaller closeness value.

There is no node that has closeness centrality equal to zero so there is no node that is isolated. On the other hand, there is no node with closeness centrality equal to one, so there is no node directly connected to all the other nodes

Maximum Closeness Centrality: 0.4266276796134395
Minimum Closeness Centrality: 0.0

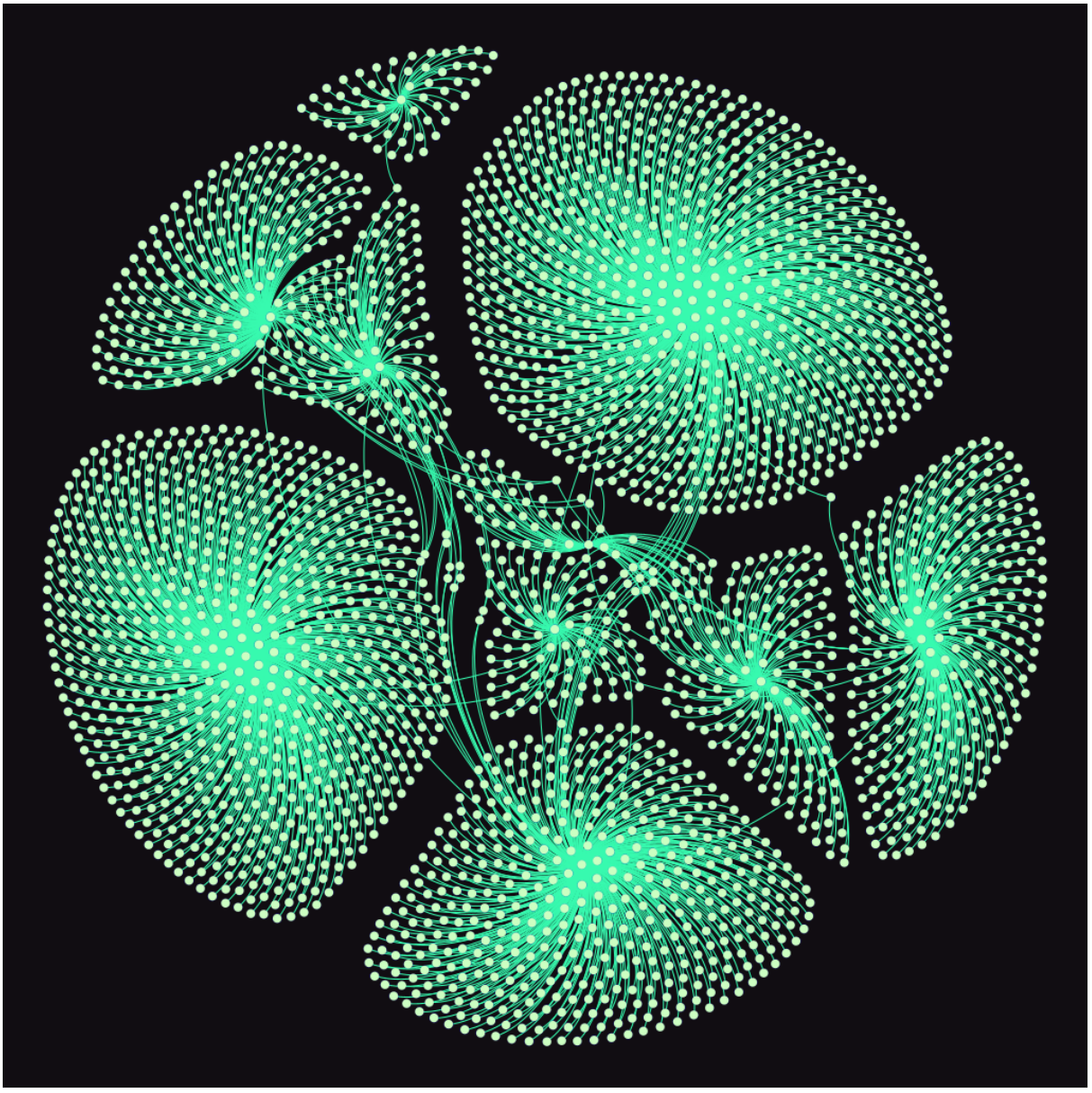
In order to test this, we added a single node with the number “9000” and we realized that his closeness centrality is 0.0, so it is an isolated node. We can see that node 9000 is isolated in the graph below:



Betweenness Centrality:

Betweenness centrality measures the extent to which a node lies on paths between other nodes. Nodes with high betweenness may have considerable influence within a network by virtue of their control over information passing between others. In our graph, we found out that the number of nodes that have high betweenness is 21. That means that if we remove these nodes from the network will most disrupt communications between other nodes because they lie on the largest number of paths traversed by messages.
The maximum betweenness centrality in our graph is 0.5485656806652164
and the minimum betweenness centrality in our graph is 0.0.

Graph before removing nodes with high betweenness



Graph after removing nodes with high betweenness