

Data Source:

The data I used is from [Kaggle](#). It is a Twitter Communication Network dataset with the names of 1000 users and dummy variables to show if they follow each of the other users. The graph is directed and labeled and has 1000 nodes, and 50153 edges. Huawei uses this Twitter data as part of its social network analysis for customer analysis in its marketing strategy. The company employs social network analysis tools and techniques in its research and development efforts. The data collected from Twitter, along with Facebook and Instagram, is used to enhance Huawei's business positions and promote its products through social media.

Data Cleaning:

The first step was data cleaning and preparation. I converted the Excel spreadsheet into a txt file. Then I systematically assigned each person a unique number ID from 0 to 999. This created nodes to represent each person. Next I went through the list of connections between people. Whenever two people knew each other, I created a connection or "edge" between their number IDs in the system. For example, if the original data showed that "Mary" knew "John", I would create an edge between the node representing Mary and the node representing John. By going through all the connections, I effectively converted the readable name data into an abstract representation using numbered nodes and edges in a network structure. This prepared graph could then be analyzed using graph analysis learned in class to understand patterns and insights about the relationships between people. The text-based edge list format made it possible to run calculations and metrics on the connections. By doing the work of data cleaning and conversion up front, I enabled deeper analysis on this people network down the line.

Next, I had to read the text-file (named `t_twitter.txt`) that I had created, and convert it into a Vector of tuples in Rust for further processing (using Rust's file I/O capabilities). A Vector in Rust allows storing data in a resizable array format. I needed to get the edge data into a structure I could manipulate. I defined a tuple struct that held a start node ID and end node ID for each edge. As I parsed the text file, I created many tuple instances capturing each edge and inserted them into the Vector. This gave me a clean representation of all connections from the original Excel sheet. Finally I defined my own custom Graph struct in Rust to hold nodes and edges. I implemented various functions for this Graph to add nodes, add edges between nodes, fetch neighbors of a node etc.

Graph Analysis

After I had built my Graph data structure of all the people and connections, I wanted to filter down to the most influential people. To do this, I implemented the PageRank algorithm which assigns a score to each node based on the number and weight of incoming edges. By running PageRank, I got a ranking of nodes - essentially people - by importance. I took the top 50 highest ranked nodes to make a filtered graph.

I made a new Graph keeping only the top 50 nodes and any edges between them from the original graph. The graph containing the Top 50 edges, could vary each time we run our code, as it contains a finite amount of randomness.) This gave me a subset graph focused on the well-connected people in the network. I could analyze just this filtered down graph to understand patterns between the influencers. On this graph, I ran a Breadth First Search algorithm starting

from specified root nodes. This allowed me to traverse all nodes reachable from, for example, the most influential people. By recording the nodes visited in order, I could see how connections flowed outward through this elite sub-graph. Ultimately I calculated distances between all pairs of the top 50 nodes with shortest paths. This final metric exposed insights into how closely connected various influencers were based on traversing friendship links. The exact top 50 varied due to randomness in PageRank, but the process revealed core insights each time into this key network.

Conclusion

After applying my graph analysis process, I could see that the distances between the top 50 nodes were very small - mostly between 1 and 2 steps, with a few being 3. This is a surprisingly tiny distance compared to other typical social networks. The top nodes tend to be very tightly interconnected - they either connect directly or have lots of mutual connections. This contrasts with two random people in a network, who would likely be many more hops apart. The short path lengths arise because prominent people in this network cluster together. My algorithmic analysis quantitatively demonstrated that the top ranked people form a dense mesh of connectivity between themselves.

Output:

```
Length of our entire list of edges (connections between nodes):1245
Length of our Graph: 50

Biggest Vertex plus 1 (our N value): 927

These are the top 50 nodes and their connections between them:
(0) Node: 19 - Edges: [19, 47, 73, 84, 153, 196, 210, 211, 217, 250, 252, 255, 274, 283, 307, 349, 371, 390, 403, 412, 414, 445, 449, 450, 472, 473, 479, 506]
(1) Node: 34 - Edges: [34, 39, 47, 59, 66, 73, 84, 105, 133, 151, 153, 196, 210, 217, 222, 238, 250, 252, 307, 320, 340, 343, 349, 367, 400, 412, 414, 445, 449, 450, 472, 473, 506]
(2) Node: 39 - Edges: [19, 34, 39, 54, 73, 84, 105, 151, 153, 218, 238, 250, 252, 255, 274, 340, 343, 349, 376, 400, 403, 449, 472, 479]
(3) Node: 47 - Edges: [19, 34, 47, 54, 66, 105, 133, 151, 153, 210, 217, 218, 238, 256, 309, 320, 340, 349, 367, 369, 371, 376, 390, 412, 445, 450, 473]
(4) Node: 54 - Edges: [34, 47, 54, 73, 105, 133, 151, 211, 218, 238, 250, 252, 283, 307, 309, 334, 340, 343, 349, 367, 371, 376, 403, 412, 414, 450, 506, 926]
(5) Node: 59 - Edges: [47, 54, 66, 84, 105, 133, 151, 153, 210, 238, 255, 256, 274, 309, 334, 340, 367, 369, 371, 376, 400, 403, 414, 445, 449, 479, 506, 926]
(6) Node: 66 - Edges: [34, 59, 73, 84, 133, 153, 196, 218, 250, 274, 307, 320, 340, 343, 349, 369, 376, 390, 412, 473, 479, 506, 926]
(7) Node: 73 - Edges: [19, 39, 47, 54, 66, 73, 105, 133, 153, 196, 210, 211, 218, 222, 238, 256, 283, 309, 340, 343, 349, 369, 371, 376, 390, 403, 414, 449, 450, 472, 473, 506]
(8) Node: 84 - Edges: [19, 39, 59, 84, 105, 133, 151, 217, 218, 238, 250, 256, 274, 320, 334, 340, 343, 349, 371, 390, 445, 449, 450, 472, 473, 479, 926]
(9) Node: 105 - Edges: [19, 39, 64, 66, 84, 151, 153, 211, 250, 252, 255, 274, 283, 309, 343, 367, 403, 926]
(10) Node: 133 - Edges: [19, 54, 59, 66, 105, 153, 210, 211, 217, 238, 250, 252, 274, 309, 334, 340, 349, 367, 371, 376, 390, 400, 403, 479]
(11) Node: 151 - Edges: [34, 47, 54, 59, 73, 84, 105, 133, 210, 211, 218, 250, 256, 274, 307, 320, 334, 340, 343, 403, 414, 449, 450, 473, 479]
(12) Node: 153 - Edges: [39, 54, 73, 84, 105, 153, 196, 210, 217, 238, 250, 255, 274, 320, 340, 369, 371, 403, 414, 479, 506]
(13) Node: 196 - Edges: [19, 34, 47, 105, 151, 196, 210, 211, 217, 218, 238, 252, 256, 309, 320, 340, 343, 367, 369, 371, 414, 449, 450, 479, 926]
(14) Node: 210 - Edges: [19, 34, 47, 54, 66, 73, 196, 210, 218, 238, 252, 255, 274, 320, 340, 343, 349, 367, 414, 445, 472, 473, 479, 506]
(15) Node: 211 - Edges: [34, 39, 54, 66, 84, 105, 133, 153, 211, 217, 218, 222, 238, 252, 256, 274, 283, 309, 320, 414, 479]
(16) Node: 217 - Edges: [19, 34, 39, 47, 66, 84, 105, 133, 151, 210, 218, 222, 238, 250, 252, 255, 256, 274, 307, 334, 340, 343, 371, 400, 412, 414, 445, 472, 926]
(17) Node: 218 - Edges: [34, 39, 66, 105, 133, 153, 210, 211, 218, 238, 250, 252, 256, 283, 320, 349, 367, 390, 506]
(18) Node: 222 - Edges: [34, 39, 47, 59, 133, 151, 210, 218, 256, 274, 334, 343, 349, 371, 400, 403, 414, 450, 473, 479, 506]
(19) Node: 238 - Edges: [34, 59, 151, 153, 196, 218, 256, 307, 334, 340, 343, 376, 390, 403, 445, 449, 472, 473, 479, 506]
(20) Node: 250 - Edges: [19, 34, 39, 66, 73, 84, 133, 196, 218, 211, 218, 222, 238, 255, 274, 343, 349, 367, 371, 403, 412, 414, 449, 473, 506]
(21) Node: 252 - Edges: [19, 34, 39, 64, 66, 84, 105, 133, 151, 196, 218, 238, 274, 307, 309, 334, 367, 371, 403, 412, 414, 445, 450, 473, 479]
(22) Node: 255 - Edges: [34, 39, 54, 59, 73, 151, 153, 210, 217, 222, 250, 255, 256, 307, 320, 340, 369, 390, 400, 412, 445, 450, 472, 473, 506, 926]
(23) Node: 256 - Edges: [19, 34, 39, 47, 54, 105, 133, 196, 210, 211, 217, 222, 250, 252, 255, 309, 320, 340, 343, 349, 367, 371, 414, 449, 450, 472]
(24) Node: 274 - Edges: [34, 39, 84, 105, 151, 217, 218, 255, 256, 274, 283, 320, 367, 369, 390, 400, 403, 414, 472, 506]
(25) Node: 283 - Edges: [19, 34, 39, 66, 73, 84, 105, 133, 151, 153, 196, 210, 211, 217, 218, 250, 252, 307, 334, 343, 369, 400, 412, 414, 449, 473, 479]
(26) Node: 307 - Edges: [19, 34, 39, 59, 66, 84, 133, 210, 211, 218, 238, 250, 255, 256, 283, 334, 340, 343, 367, 371, 414, 445, 449, 450, 473, 479]
(27) Node: 309 - Edges: [34, 39, 47, 54, 59, 73, 151, 153, 222, 255, 256, 307, 334, 343, 367, 369, 390, 400, 412, 449, 450, 473, 479, 506]
(28) Node: 320 - Edges: [19, 59, 84, 133, 153, 196, 210, 218, 222, 238, 283, 309, 390, 412, 414, 449, 450, 473, 926]
(29) Node: 334 - Edges: [19, 34, 47, 59, 73, 105, 153, 196, 218, 218, 250, 255, 256, 274, 307, 309, 320, 334, 340, 369, 414, 449, 450, 472, 473, 479, 506]
(30) Node: 340 - Edges: [34, 84, 133, 151, 218, 250, 252, 255, 274, 283, 309, 334, 343, 367, 371, 376, 412, 414, 445, 473, 479, 506]
(31) Node: 343 - Edges: [19, 47, 54, 59, 73, 84, 105, 133, 153, 210, 211, 217, 222, 255, 256, 274, 343, 349, 367, 371, 390, 412, 414, 445, 449, 473, 506]
(32) Node: 349 - Edges: [19, 34, 54, 59, 73, 84, 105, 133, 196, 210, 211, 218, 238, 250, 307, 309, 349, 371, 376, 390, 400, 403, 412, 449, 450, 472, 926]
(33) Node: 367 - Edges: [19, 39, 47, 54, 59, 153, 196, 211, 218, 222, 250, 307, 309, 320, 334, 343, 349, 376, 412, 414, 449, 450, 472, 506, 926]
(34) Node: 369 - Edges: [19, 54, 59, 66, 84, 133, 196, 250, 255, 283, 307, 334, 340, 349, 367, 371, 390, 403, 412, 414, 449, 473, 479]
(35) Node: 371 - Edges: [34, 59, 73, 84, 133, 153, 211, 218, 222, 238, 252, 274, 283, 307, 320, 340, 349, 369, 371, 445, 472, 473]
(36) Node: 376 - Edges: [19, 34, 54, 59, 84, 151, 153, 196, 218, 238, 250, 252, 255, 256, 343, 349, 367, 369, 371, 400, 403, 412, 450, 479, 506, 926]
(37) Node: 390 - Edges: [39, 54, 59, 84, 151, 153, 196, 211, 218, 222, 250, 252, 334, 343, 349, 367, 369, 371, 376, 390, 403, 412, 445, 479]
(38) Node: 400 - Edges: [19, 34, 59, 66, 73, 105, 153, 210, 211, 250, 252, 256, 283, 320, 343, 349, 367, 371, 403, 412, 449, 472, 473, 479, 506, 926]
(39) Node: 403 - Edges: [19, 34, 39, 54, 59, 73, 105, 133, 196, 210, 218, 222, 238, 250, 252, 307, 320, 340, 367, 376, 390, 412, 414, 445, 450, 473, 479, 926]
(40) Node: 412 - Edges: [19, 47, 54, 59, 66, 73, 151, 153, 196, 210, 211, 222, 238, 283, 309, 320, 334, 367, 369, 371, 376, 400, 414, 449, 472, 479]
(41) Node: 414 - Edges: [19, 54, 59, 73, 133, 153, 196, 211, 222, 250, 256, 274, 343, 367, 369, 371, 376, 390, 400, 403, 450, 472, 473]
(42) Node: 445 - Edges: [39, 54, 59, 73, 84, 151, 153, 196, 211, 217, 218, 252, 256, 283, 307, 309, 320, 334, 343, 349, 371, 376, 390, 400, 412, 414, 445, 449, 450, 472, 479, 926]
(43) Node: 449 - Edges: [19, 39, 47, 54, 59, 66, 73, 105, 133, 153, 196, 222, 256, 274, 307, 309, 320, 334, 367, 369, 371, 376, 414, 445, 473, 479]
(44) Node: 450 - Edges: [34, 54, 66, 151, 153, 211, 218, 222, 250, 255, 256, 274, 307, 320, 334, 343, 367, 400, 403, 412, 414, 445, 472, 473, 926]
(45) Node: 472 - Edges: [19, 34, 39, 66, 73, 84, 105, 196, 210, 238, 256, 309, 320, 343, 349, 367, 400, 403, 412, 445, 450, 473]
(46) Node: 473 - Edges: [19, 39, 54, 66, 84, 133, 153, 217, 218, 222, 250, 256, 274, 283, 349, 369, 371, 400, 449, 450, 473, 479, 926]
(47) Node: 479 - Edges: [34, 54, 66, 151, 153, 211, 218, 222, 250, 255, 283, 307, 334, 340, 343, 349, 369, 390, 412, 445, 449, 450, 473, 926]
(48) Node: 506 - Edges: [19, 34, 47, 66, 84, 151, 153, 217, 222, 238, 252, 274, 283, 320, 343, 367, 376, 390, 449, 450, 472, 473, 479, 506]
(49) Node: 926 - Edges: [19, 34, 59, 73, 105, 151, 196, 210, 222, 255, 256, 274, 283, 307, 309, 334, 343, 367, 369, 376, 390, 400, 403, 412, 450, 472, 479, 506, 926]

Number of Edges Visited in BFS: 33
```

The distance between the first node = (19) and the rest of nodes inside the top 50 nodes is:

```
->The distance between node 19 and node 19 is 0.
->The distance between node 19 and node 34 is 2.
->The distance between node 19 and node 39 is 2.
->The distance between node 19 and node 47 is 1.
->The distance between node 19 and node 54 is 2.
->The distance between node 19 and node 59 is 2.
->The distance between node 19 and node 66 is 2.
->The distance between node 19 and node 73 is 1.
->The distance between node 19 and node 84 is 1.
->The distance between node 19 and node 105 is 2.
->The distance between node 19 and node 133 is 2.
->The distance between node 19 and node 151 is 2.
->The distance between node 19 and node 153 is 1.
->The distance between node 19 and node 196 is 1.
->The distance between node 19 and node 210 is 1.
->The distance between node 19 and node 211 is 1.
->The distance between node 19 and node 217 is 1.
->The distance between node 19 and node 218 is 2.
->The distance between node 19 and node 222 is 2.
->The distance between node 19 and node 238 is 2.
->The distance between node 19 and node 250 is 1.
->The distance between node 19 and node 252 is 1.
->The distance between node 19 and node 255 is 1.
->The distance between node 19 and node 256 is 2.
->The distance between node 19 and node 274 is 1.
->The distance between node 19 and node 283 is 1.
->The distance between node 19 and node 307 is 1.
->The distance between node 19 and node 309 is 2.
->The distance between node 19 and node 320 is 2.
->The distance between node 19 and node 334 is 2.
->The distance between node 19 and node 340 is 2.
->The distance between node 19 and node 343 is 2.
->The distance between node 19 and node 349 is 1.
->The distance between node 19 and node 367 is 2.
->The distance between node 19 and node 369 is 2.
->The distance between node 19 and node 371 is 1.
->The distance between node 19 and node 376 is 2.
->The distance between node 19 and node 390 is 1.
->The distance between node 19 and node 400 is 2.
->The distance between node 19 and node 403 is 1.
->The distance between node 19 and node 412 is 1.
->The distance between node 19 and node 414 is 1.
->The distance between node 19 and node 445 is 1.
->The distance between node 19 and node 449 is 1.
->The distance between node 19 and node 450 is 1.
->The distance between node 19 and node 472 is 1.
->The distance between node 19 and node 473 is 1.
->The distance between node 19 and node 479 is 1.
->The distance between node 19 and node 506 is 1.
->The distance between node 19 and node 926 is 2.
```