Social Network Analysis

Twitter

Abdelrahman Mohamed Ameen 120200213

Ahmed maher kamal 120200215

Mohamed Abdelsmaiea Baisam 120200191

Introduction:

This report focuses on analyzing Twitter data specifically from Egypt to uncover insights into the social network dynamics within the Egyptian online community. By studying the followers' network and interactions of Egyptian Twitter users, we aim to understand unique patterns, influential figures, and information flow within this context. The findings of this analysis contribute to our understanding of the Egyptian online community and have implications for various domains such as journalism, social activism, marketing, and public opinion research in Egypt.

Our Main Objective:

Our main objectives are to understand the network properties, centrality measures, and community structures within the collected social media data. We aim to identify key influencers, discover communities of users, and gain insights into the overall network dynamics.

Tools and Libraries Used:

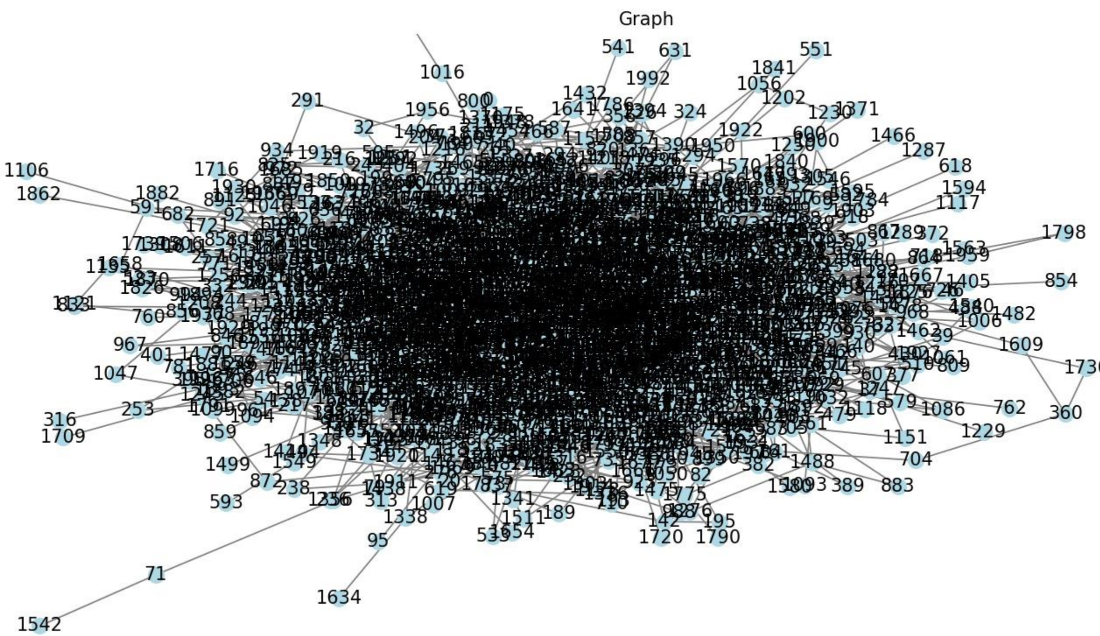
For this project, we utilized Python as our programming language. We made use of several libraries, including pandas, snscrape, networkx, and matplotlib, to perform data collection, analysis, and visualization. Let's delve into the details of our project and discover the fascinating insights we gained through network analysis of social media data.

Data Scraping:

The first step in our project is to gather relevant data from Twitter through the process of data scraping. To accomplish this, we utilize the snscrape library, a powerful tool specifically designed for scraping tweets. With snscrape, we can extract tweets from a specific Twitter user by providing their username as input. In addition to the username, we also specify the number of tweets we want to scrape. This allows us to control the size of our data set. By scraping tweets, we collect real-time information from the Twitter platform, enabling us to perform network analysis on the latest data available. Data scraping is a crucial step as it forms the foundation for our subsequent analysis. The quality and relevance of the scraped data directly impact the accuracy and insights derived from our network analysis. It's important to note that data scraping should be performed ethically and in accordance with the terms and conditions of the respective social media platform. Once we have successfully scraped the tweets, we move on to the next step: graph generation.

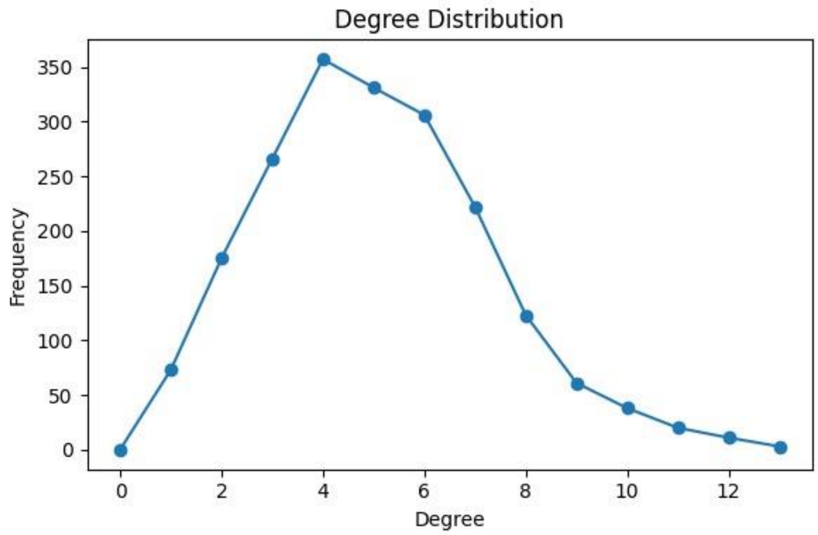
Graph Generation:

After gathering the relevant tweets, we move on to the graph generation step, where we create a representative graph using the NetworkX library. NetworkX is a powerful Python library for the study of complex networks, providing a wide range of graph generation and analysis capabilities. In our project, we generate a random graph using NetworkX. The random graph is created with a specified number of nodes and edges. The number of nodes represents the size of our graph, and the number of edges determines the connectivity between the nodes. Generating a random graph allows us to simulate a network structure for our Twitter data, providing a basis for further analysis. The graph generation process is essential as it lays the groundwork for network analysis techniques. It enables us to analyze the relationships, connectivity, and properties within our Twitter data. It's important to note that the generated graph is a simplified representation of the Twitter network, capturing the essential relationships between nodes without incorporating real-world complexities. With the graph successfully generated, we can now proceed to the next step: network analysis.



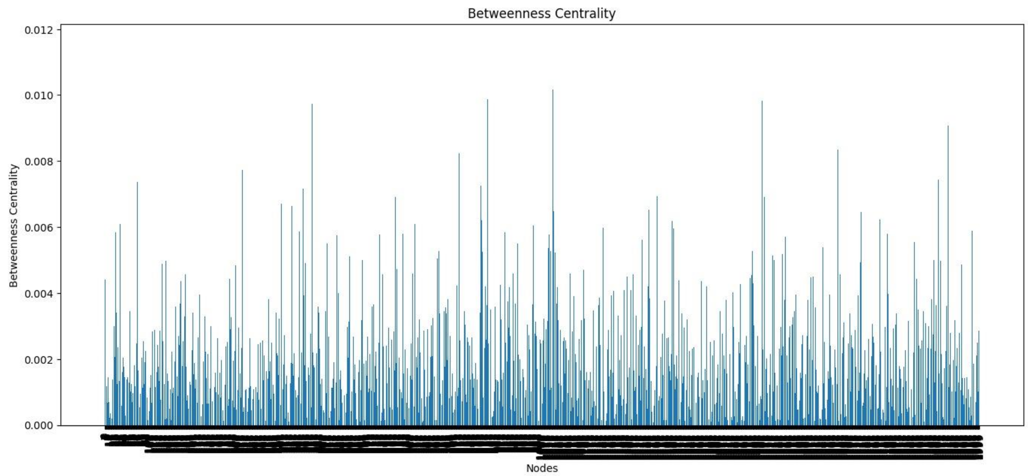
Network Analysis:

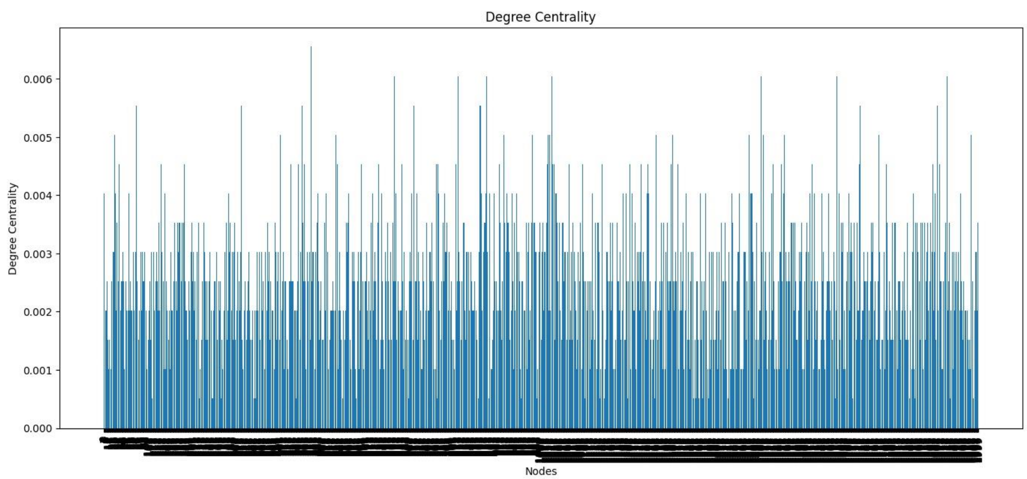
Degree Distribution Analysis: Degree distribution analysis examines the distribution of node degrees within a social network graph. It helps understand how many connections (edges) each node (user) has, revealing the network's overall structure. By analyzing the degree distribution, we can identify patterns such as power-law distributions, indicating the presence of influential nodes with a large number of connections. This analysis is implemented by calculating the degree of each node in the network and plotting a histogram or using statistical measures to understand the distribution pattern.



Path Analysis: Path analysis focuses on understanding the paths or routes between nodes in a social network graph. It helps identify the shortest paths, connectivity, and navigation patterns within the network. This analysis is implemented by finding the shortest path between pairs of nodes using algorithms like Dijkstra's algorithm or breadth-first search. It can reveal important connections and potential information flow paths within the network.

Centrality Analysis: Centrality analysis determines the importance or influence of nodes within a social network. It helps identify key nodes that play a crucial role in information flow, communication, or control within the network. Different centrality measures like degree centrality, betweenness centrality, and closeness centrality can be used to assess the prominence of nodes. This analysis is implemented by calculating the centrality measures for each node in the network and ranking them accordingly.





Connected Components Analysis: Connected components analysis identifies distinct groups or clusters of nodes within a social network graph. It helps understand the level of connectivity and the presence of isolated groups in the network. By detecting connected components, we can identify communities or sub networks with high internal connectivity. This analysis is implemented by using algorithms such as depth-first search or breadth-first search to identify groups of connected nodes.

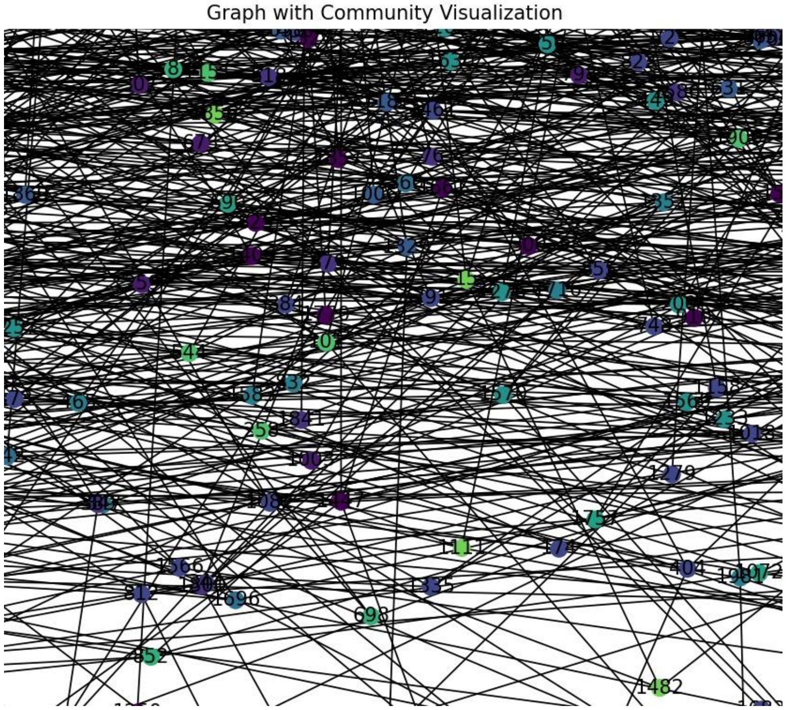
Clustering Coefficients: Clustering coefficients measure the level of clustering or interconnectedness among a node's neighbors within a social network. It helps identify the presence of tightly knit communities or cliques. Higher clustering coefficients indicate dense local connections within a network. This analysis is implemented by calculating the clustering coefficient for each node or aggregating it to assess the overall clustering tendency of the network.

Density Analysis: Density analysis determines the density of connections within a social network graph. It quantifies the proportion of actual connections compared to the total number of possible connections in the network. Higher density indicates a tightly interconnected network with extensive communication and information flow. This analysis is implemented by calculating the density using the number of edges and nodes in the network.

Network Type: Network type analysis categorizes the social network graph based on its characteristics and structure. It helps understand whether the network is scale-free, small-world, hierarchical, or random, among other types. This analysis is implemented by examining the distribution of node degrees, path lengths, and clustering coefficients to identify the network's underlying type.

Community Discovery: Community discovery identifies groups of nodes that exhibit strong internal connections and weak connections with nodes outside the group. It helps identify communities or clusters within the social network that share common interests, affiliations, or interactions. Various algorithms such as modularity optimization or label propagation can be employed for community discovery. This analysis is implemented by applying community detection algorithms to partition the network into cohesive groups.

Graph with Community Visualization



Dynamic Community Discovery: Dynamic community discovery focuses on identifying and analyzing communities in social networks that evolve over time. It helps understand how communities form, dissolve, merge, or change their structure over different time intervals. This analysis is implemented by considering time stamped data and applying algorithms that capture the temporal evolution of communities. It can provide insights into the dynamics of relationships and interests within the network.

Our code:

https://github.com/ahmed5036/Algorithms-project