Mini Project 1

Computational Social Science

Lukas Burtscher (11925939)
Dante Godolja (11929150)

Dataset

The dataset consists of a comprehensive social network of GitHub developers, gathered from the public API in June 2019. It comprises nodes representing developers who have demonstrated interest in at least 10 repositories, and edges that represent mutual follower relationships between them. Each node in the network is associated with various features, including information about the developer's location, repositories they have starred, their current employer, and their e-mail address.

The primary objective of this dataset is binary node classification, specifically to predict whether a GitHub user is a web developer or a machine learning developer. This classification task is based on the user's job title, which has been derived from the available information.

Methodology

In our network, consisting of 37,700 nodes (accounts) and 289,003 edges (connections), our computational resources limited us from conducting complex analysis algorithms.

To begin, we employed pandas and numpy for initial data exploration. Subsequently, we constructed the network using the NetworkX library. To analyze the network, we employed the following metrics:

Degree
In a NetworkX graph, the degree of a node refers to the number of edges incident to that node. It represents the measure of connectivity or centrality of a node within the graph.

Specifically, for an undirected graph, the degree of a node is the count of edges that are connected to it. In other words, it quantifies the number of neighbors a node has. For a directed graph, the degree is divided into two measures: the in-degree and the out-degree. The in-degree of a node indicates the number of incoming edges, while the out-degree refers to the number of outgoing edges.

Clustering
The clustering coefficient of a node quantifies the likelihood that its neighboring nodes are connected to each other. It reflects the tendency of nodes in a graph to form tightly interconnected groups or clusters. The clustering coefficient is defined for undirected graphs and can range from 0 to 1.

A high clustering coefficient suggests that the neighbors of a node are highly interconnected, forming a densely connected local neighborhood. This indicates the presence of cohesive subgroups or communities within the graph. In other words, nodes with a high clustering coefficient are more likely to have mutual connections among their neighbors.

Conversely, a low clustering coefficient indicates a more sparse or random pattern of connections among neighboring nodes. This suggests a lack of clustering or community structure in the graph, with nodes having fewer mutual connections.

Eigenvector Centrality
The eigenvector centrality takes into account both the number of connections a node has and the centrality of those connecting nodes. It assigns higher centrality scores to nodes that are connected to other highly central nodes.

The calculation of eigenvector centrality is based on the concept of eigenvectors, which are the special vectors associated with eigenvalues in linear algebra. In the context of graph theory, eigenvector centrality is computed by iteratively calculating the principal eigenvector of the adjacency matrix of the graph.

Nodes with higher eigenvector centrality scores are considered more influential or central within the network. They have a higher likelihood of being connected to other influential nodes. In other words, nodes with high eigenvector centrality are connected to other nodes that themselves have high centrality.

Eigenvector centrality is particularly useful in identifying important nodes in a network, often referred to as "hubs." These hub nodes have connections to other highly connected nodes, allowing them to control the flow of information or influence within the network.

Pagerank
The PageRank algorithm assigns a numerical score to each node in the graph based on the structure of incoming and outgoing links. The underlying idea is that a node is considered more important if it receives links from other important nodes.

When applying PageRank to an undirected graph, the algorithm calculates the scores based on the connectivity of nodes, considering both incoming and outgoing edges. The algorithm treats the undirected edges as bidirectional links, allowing the flow of importance to propagate through the graph.

Degree Centrality
Degree centrality quantifies the extent to which a node is connected to other nodes in the network. It is calculated by dividing the number of edges connected to a node (degree) by the maximum possible degree in the graph.

Nodes with higher degree centrality scores have a greater number of connections or edges, indicating their prominence or influence within the network. They serve as important communication hubs and are often referred to as "hubs" or "popular" nodes.

Degree centrality is straightforward to calculate and provides a basic measure of node importance. However, it does not take into account the centrality of the nodes to which a node is connected. In other words, it does not consider the influence or centrality of the neighboring nodes.

K-Core
To identify k-core communities, the k-core decomposition algorithm is used. The algorithm iteratively removes nodes with fewer than k connections from the network until no such nodes remain. The resulting subgraph, called the k-core community, contains nodes that are densely interconnected and have a minimum degree of k.

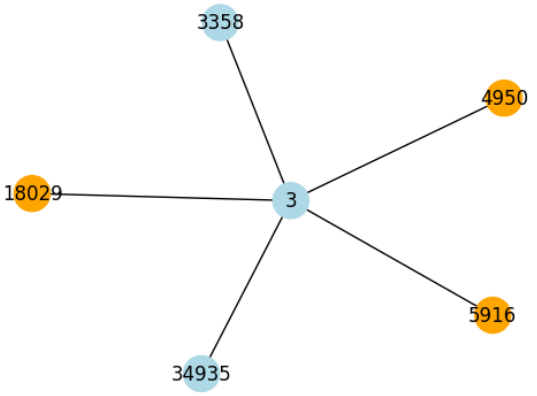
K-core communities are useful for understanding the hierarchical structure and connectivity patterns within a network. They provide insights into the cohesive groups or clusters of nodes that are highly interconnected and have stronger internal ties compared to the rest of the network.

Limitations

Regarding the size of the network we did not have the computational power to compute betweenness and closeness centrality. Moreover we were not able to draw the network in the notebook due to computational limitations.

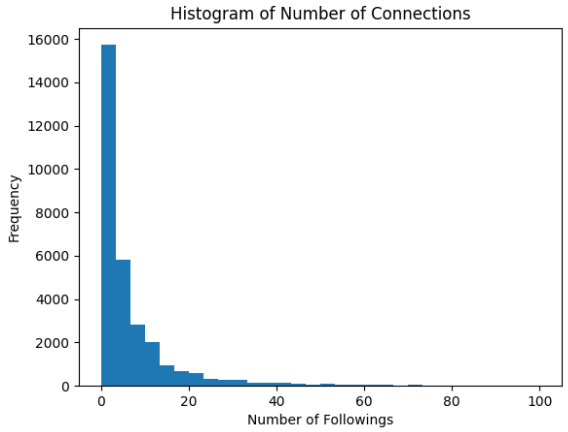
Insights and Results

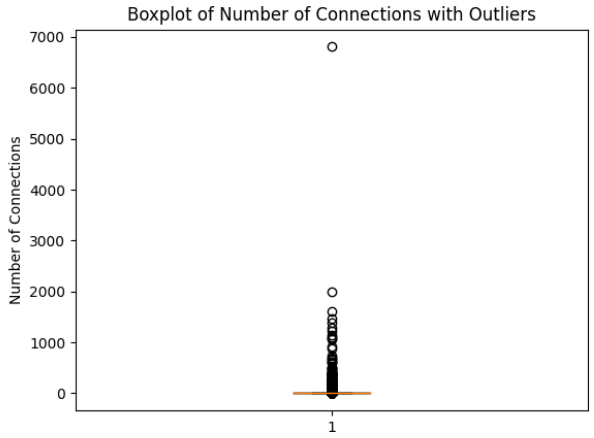
The dataset comprises 37,700 GitHub developers, with 9,739 identified as machine learning developers and 27,961 as web developers. This indicates a ratio of approximately 25% machine learning developers and 75% web developers within the dataset.



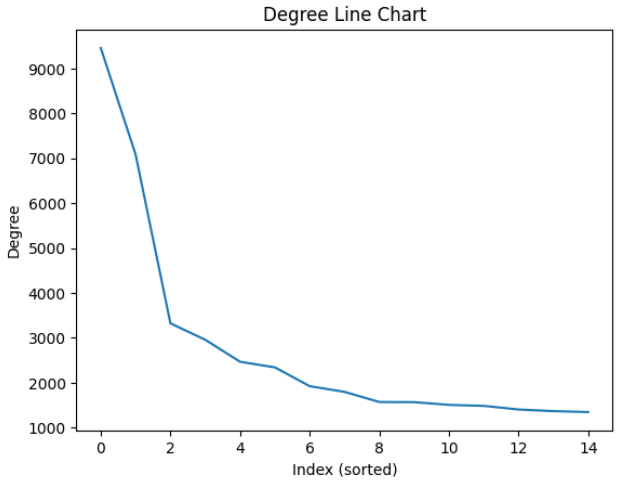
Here we can see an example subgraph consisting of node 3 with all its neighbours. The blue colour indicates that this account belongs to a web developer.

Upon examining the friend count (number of following connections) among the GitHub users, we observe that the median friend count is 3. However, there are a few outliers with significantly higher friend counts, which would impact the average value. To visualize this, a histogram of the friend count is presented alongside a box plot. The x-axis of the histogram is limited due to the presence of these outliers. Notably, one account has around 7,000 friends, while the majority of accounts have between 1 and 20 friends.



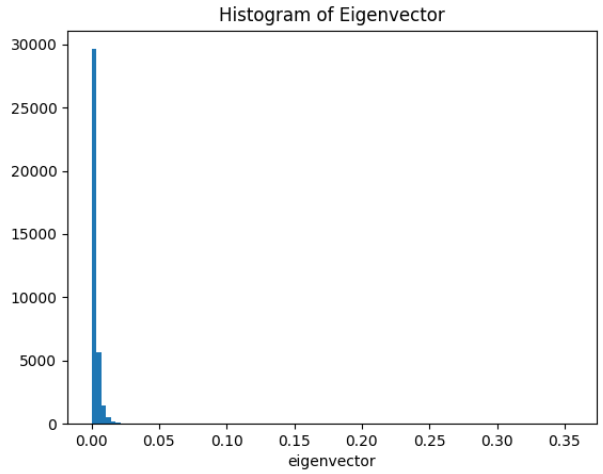


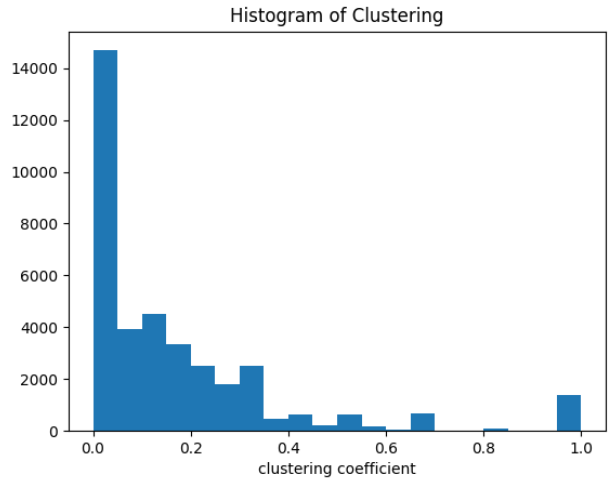
When analyzing the connectivity of the developers, the average number of connections (degree) is found to be 15.33. This indicates the average number of friends or connections each account has. Specifically, machine learning developers have an average of 29 connections, while web developers have an average of 11 connections.



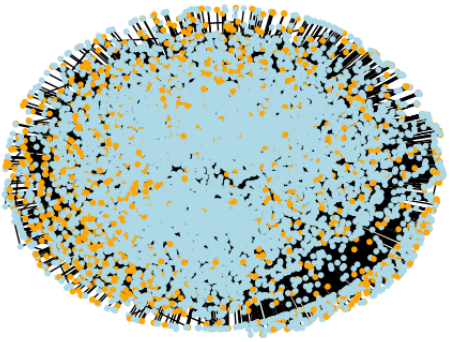
Very interesting to see is that, there are really only a few accounts with more than 1000 connections. The line charts show the top 15 accounts sorted by degree in descending order and there is a significant gap between the highest degree node (>9000) and the third highest node (<3500). After 15 out of over 30 000 measured accounts, we are under 1500 connections.

The clustering coefficient, which measures the tendency of nodes to form clusters or tightly interconnected groups, has an average value of 0.1675. This suggests a relatively low level of clustering within the network. Additionally, the eigenvector distribution among the accounts shows that nearly all accounts have eigenvector centrality values smaller than 0.02. On the other hand, the clustering coefficients exhibit a more even distribution, with the majority of values falling between 0 and 0.3.

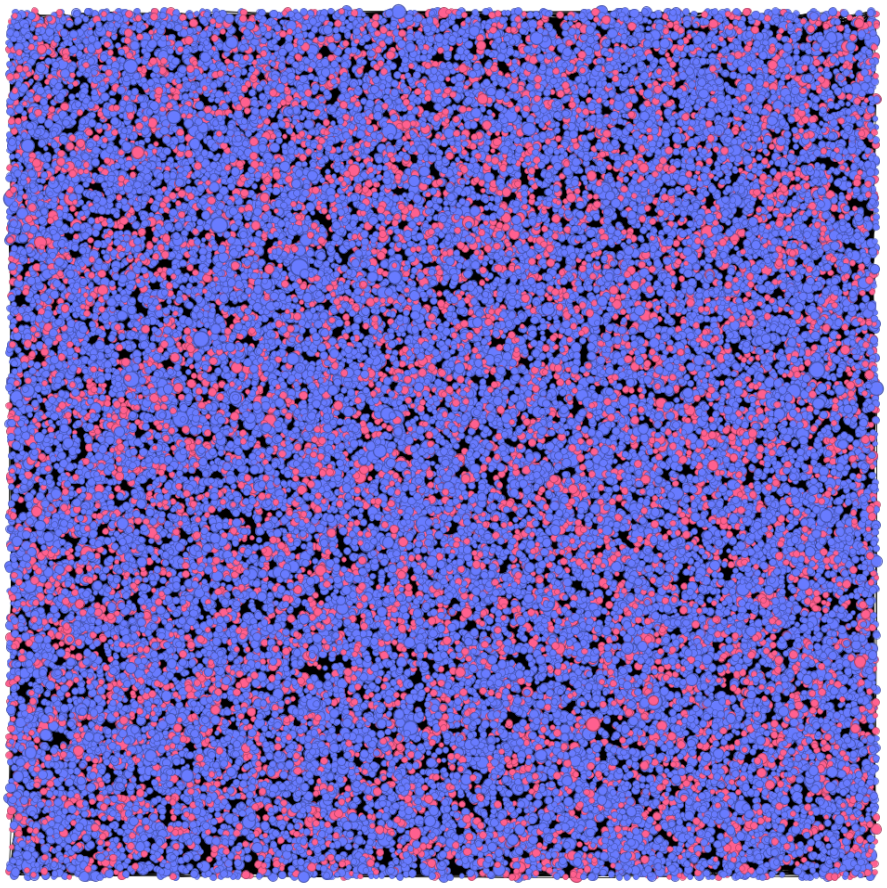




The node with the most connections also achieved the highest eigenvector coefficient, PageRank and degree centrality score. When plotting all of the neighbours the subgraph gets quite messy.



Plotting the whole network was only possible in Gephi and can be seen in the following picture. Blue indicates the web developers and pink the machine learning developers. The node sites depend on the PageRank score.



Unfortunately, the structure of the graph points to a more random pattern with nodes having low importance scores. On the other hand, the data sets provide a new data set containing numerical feature vectors extracted from the meta information of the individual accounts like work data or profile description. When training a machine learning model we certainly will also have a look into that feature vector and see if we can use this data to improve the model.