ILLINOIS TECH

Illinois Institute of Technology

ONLINE SOCIAL NETWORK ANALYSIS

Project 1

OMDb API

The Open Movie Database

Lucia Colin Cosano A20552447

Ritu Tushar Bagul A20548051

CS 579

Contents

1 DEFINITION OF THE PROJECT

Initially, the project's concept centered around creating a social network with data sourced from Twitter. However, during the exploration phase of data collection methods, it became evident that accessing comprehensive user data through Twitter's API was restricted, limiting the scope of information available for analysis.

We also investigated the possibility of utilizing LinkedIn as a data source. LinkedIn's platform offers rich professional networking data, potentially providing insights into connections between individuals based on their educational backgrounds, professional affiliations, and industry involvement. The project envisioned leveraging LinkedIn's data to create a social network that could map out connections between individuals with education at IIT. This approach aimed to identify clusters or groups within the network and determine influential figures within these groups.

However, similar to the challenges encountered with Twitter data collection, accessing comprehensive user data through LinkedIn's API proved to be limited.

Subsequently, the project shifted its focus towards leveraging the Open Movie Database (OMDb) API, which offered more accessible and diverse datasets related to the film industry.

Open Movie Database (OMDb)is a readily movie database accessible via an Application Programming Interface (API). This API streamlines the retrieval of comprehensive movie information, including titles, synopses, ratings, cast, and crew. Additionally, the project aims to develop a co-authorship network wherein various actors collaborate, and an edge is established when they participate in the same movie.

In this co-authorship network, each actor represents a node, and connections (edges) between them signify instances where they share the screen in a film. By analyzing these connections, the project seeks to uncover patterns of collaboration and identify influential actors within the film industry. This network-based approach offers insights into the dynamics of movie production, casting choices, and the interplay between different talents within the filmmaking community.

Furthermore, the utilization of the OMDb API enhances the project's capabilities by providing a rich source of movie data, enabling detailed analysis and visualization of actor collaborations over time. Through this integrated approach, the project aims to offer valuable insights into the interconnected nature of the film industry and the relationships between its key players.

2 DATA COLLECTION

Acquiring an API key is imperative for data retrieval. This key can be obtained freely by registering on the OMDb website. Once secured, the API URL is constructed, and its parameters can be tailored to fetch data based on movie Id, Title, or via a Search query.

For streamlined automation, a Python script has been developed. This script orchestrates the extraction of pertinent information and orchestrates the creation of a CSV file housing movie titles alongside their corresponding cast members.

Explanation of the main features of the code:

a) **Importing libraries:**

- requests: it is used to send HTTP requests to the OMDb API.

- csv: used for reading and writing CSV files.

b) **API Key:**

- api_key: it is the provided OMDb API key.

c) **get_movies_info function**

This function is designed to retrieve detailed information about a movie using the OMDb API. The function takes a movie title as its input, constructs a URL for the OMDb API request with the provided API key and movie title, and then utilizes the requests library to send a GET request to the constructed URL. The JSON response received from the API is parsed into a Python dictionary for ease of manipulation.

Upon receiving the API response, the function checks for success by examining if the response is "True," indicating that the movie information was successfully found. In case of success, the function further processes the response by splitting the string of actors into a list, using commas as separators.

The final output is a dictionary that includes the movie title and a list of actors. Notably, the implementation includes error handling to ensure robustness in case of unexpected issues during the API request or response process.

d) **save_movies_to_csv function**

This function is designed to store movie information into a CSV file. It takes a list of movies as its input.

The code opens a CSV file in write mode, ensuring a new line is created for each entry. It then initializes a CSV DictWriter object with fieldnames set to "Title" and "Actors" and writes the header row to the CSV file.

The function iterates through the provided list of movies. If a movie is found, its details are written to the CSV file using the DictWriter. In cases where movie data is missing, a message is printed to the console, and the movie is skipped.

2.1 Data privacy

The protection of user data and privacy is a fundamental concern for both individuals and organizations. The platform we have used is committed to upholding the highest standards of data privacy and security to ensure the trust and confidence of our users. This overview serves as a comprehensive guide to the principles, policies, and practices governing data privacy on our platform.

From the collection and handling of user information to the management of intellectual property rights, each aspect of data privacy is carefully addressed to safeguard the confidentiality and integrity of user data. By outlining clear guidelines for user conduct, contributions, and account security, we aim to create a safe and respectful online environment for all users.

Furthermore, this overview highlights our commitment to transparency and accountability by specifying copyright policy, procedures for handling disputes, and limitations of liability. Users are encouraged to familiarize themselves with these policies and actively engage with updates as needed to ensure compliance and understanding.

a) **User Information:**

- Users must provide truthful and accurate information during registration.
- Users commit to updating contact information as needed.

b) **User Conduct and Contributions:**

- Users are solely responsible for their contributions (posts, submissions) on the site.
- Prohibited content includes illegal, inappropriate, offensive, or misleading information.
- Users must comply with third-party licenses and not solicit personal information from minors.

c) **Grant of License for Contributions:**

- Users retain ownership rights to their contributions.
- Platform is granted a perpetual, non-exclusive license to use, modify, and distribute contributions worldwide.

d) **Use of Contributions by Users:**

- Users are granted a non-exclusive license to use other users' contributions for personal, non-commercial purposes.
- Specific restrictions on retaining IP notices and no unauthorized copying apply.

e) **Account Security:**

- Users must maintain the confidentiality of their account information.
- Responsible and reasonable use of the site is encouraged.

f) **Intellectual Property Rights:**

- Site content and trademarks are owned or licensed by the platform.
- Users are not allowed to interfere with security features or access the site by unauthorized means.

g) **Site Management and User Misconduct:**

- Platform reserves the right to monitor and take action against violations.
- Termination of users is at the platform's discretion.

h) **Copyright Policy:**

- Users are responsible for content submitted.
- Platform is authorized to make copies of user content for posting and storage.

i) **Modifications:**

- The platform may modify terms without notice, and users are encouraged to check for updates regularly.

j) **Non-commercial Use by Users:**

- Site is for personal use only, and commercial activities are prohibited.

k) **Third-Party Sites:**

- Users should review terms and conditions of third-party sites linked from the platform.

l) **Disputes and Choice of Law:**

- Governing laws for dispute resolution are specified.

m) **Disclaimers:**

- No warranties are provided for contributions, materials, or site availability.
- Users are advised of associated risks and encouraged to use the site responsibly.

n) **Limited Liability:**

- The platform's liability is limited, and users acknowledge the limitation.

o) **Indemnity:**

- Users agree to indemnify the platform against claims arising from their contributions or violations of the agreement.

p) **Miscellaneous:**

- The agreement constitutes the entire understanding between users and the platform.
- No third-party beneficiaries, independent contractor relationships, or waiver of rights are implied.

q) **Notices:**

- Users are responsible for keeping contact information up-to-date.
- Notices may be sent via email, post, or other reasonable means.

It was last updated on March 12, 2015.

Data privacy link: click here.

3 DATA PREPARATION AND VISUALIZATION

Following the storage of data in a CSV file, the visualization phase will be carried out using Gephi. Gephi was chosen due to prior familiarity with the platform and its compatibility with the collected data. Gephi is an open-source tool designed to unveil underlying connections within datasets, making it suitable for the current project's objectives.

This software is usually used in the following application:

- **Exploratory Data Analysis:** Gephi allows to manipulate networks in real-time, It helps you identify patterns, trends, and outliers with ease.
- **Link analysis:** uncover the complex structures underlying networks. Whether analyzing social media interactions, scientific collaborations, or financial transactions, Gephi reveals the hidden threads that bind elements together, offering a deeper understanding of their dynamics.

- **Social Networks:** by visualizing social data connectors, it helps the user gain insights into community structures, identify influential individuals, and even predict the flow of information within these networks.

Gephi has numerous implemented metris, of which the following stand out: centrality, ensity or path length.

3.1 Data input format

To facilitate the upload of data into Gephi, several transformations are necessary. These transformations have been executed through Python code to achieve the desired structure: two distinct sheets, one dedicated to nodes and the other to edges. This structuring ensures compatibility with Gephi's data import functionality, streamlining the visualization process.

Nodes Sheet:

First column: this column has to be labeled with "Id" and provide unique identifiers for each entity in the network. These IDs must be text strings or numbers and cannot contain spaces or special characters.

Additional column: it represent attributes describing each node. In this case a "Name" column will be created to insert the Name of the Actor. It will be renamed as Label in order to be able to show labels once data has been uploaded to Gephi.

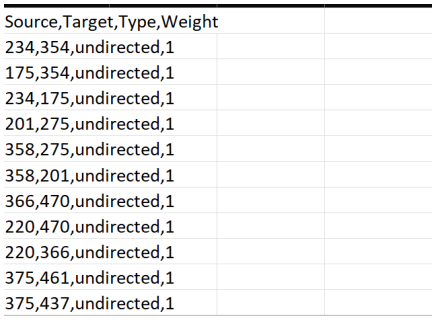
Id,Label
1,Megan Connors
2,Jeong Jin-woon
3,Kristina Cole Geddes
4,Woo-min Byeon

Figure 3.1: Nodes sheet.

Edges Sheet:

First two columns: they need to be labeled "Source" and "Target," respectively. Each row represents a connection between two nodes. In these columns, specify the corresponding IDs of the connected nodes from the "nodes" sheet.

Optional columns: This column can be used for edge attributes, describing properties of the connection itself.



Source,Target,Type,Weight
234,354,undirected,1
175,354,undirected,1
234,175,undirected,1
201,275,undirected,1
358,275,undirected,1
358,201,undirected,1
366,470,undirected,1
220,470,undirected,1
220,366,undirected,1
375,461,undirected,1
375,437,undirected,1

Figure 3.2: Edges sheet.

3.2 Instructions for Gephi

After obtaining data in the appropriate format, the next phase involves initiating work in Gephi. Initially, a new project is created. Subsequently, the Data Laboratory is accessed, providing the capability to import CSV files. During this process, files containing data pertaining to nodes and edges are uploaded, facilitating subsequent visualization and analysis within Gephi.

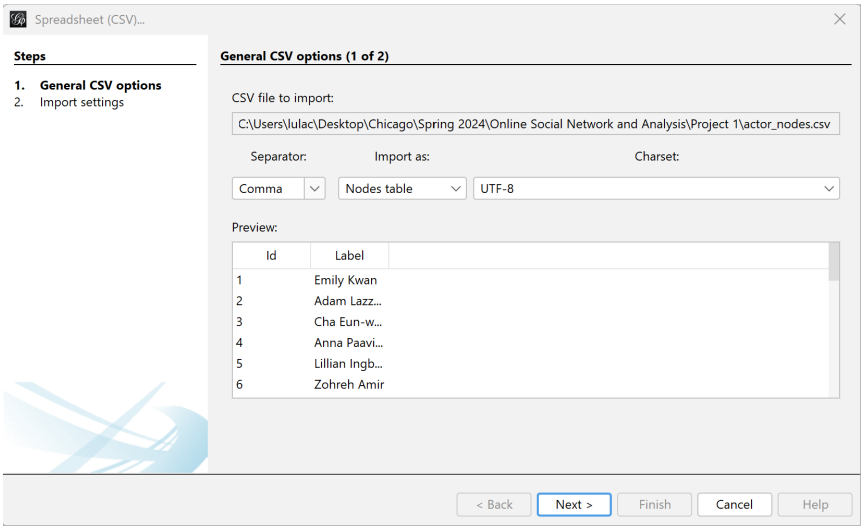


Figure 3.3: Importing nodes file.

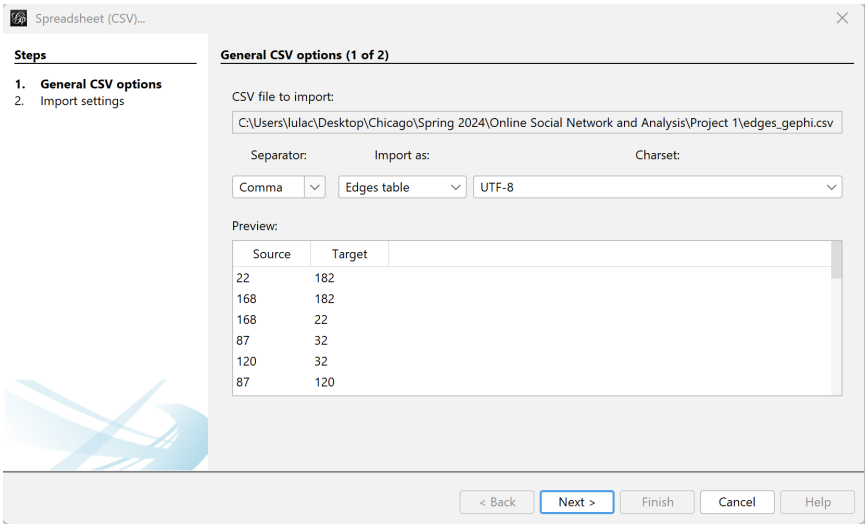


Figure 3.4: Importing edges file.

When importing data, Gephi offers users the flexibility to define settings such as the type of graph structure (directed, undirected, or mixed). This feature enables users to tailor the graph representation according to the specific characteristics of their dataset, ensuring an accurate and meaningful visualization within Gephi.

Once data is uploaded the visualization is done.

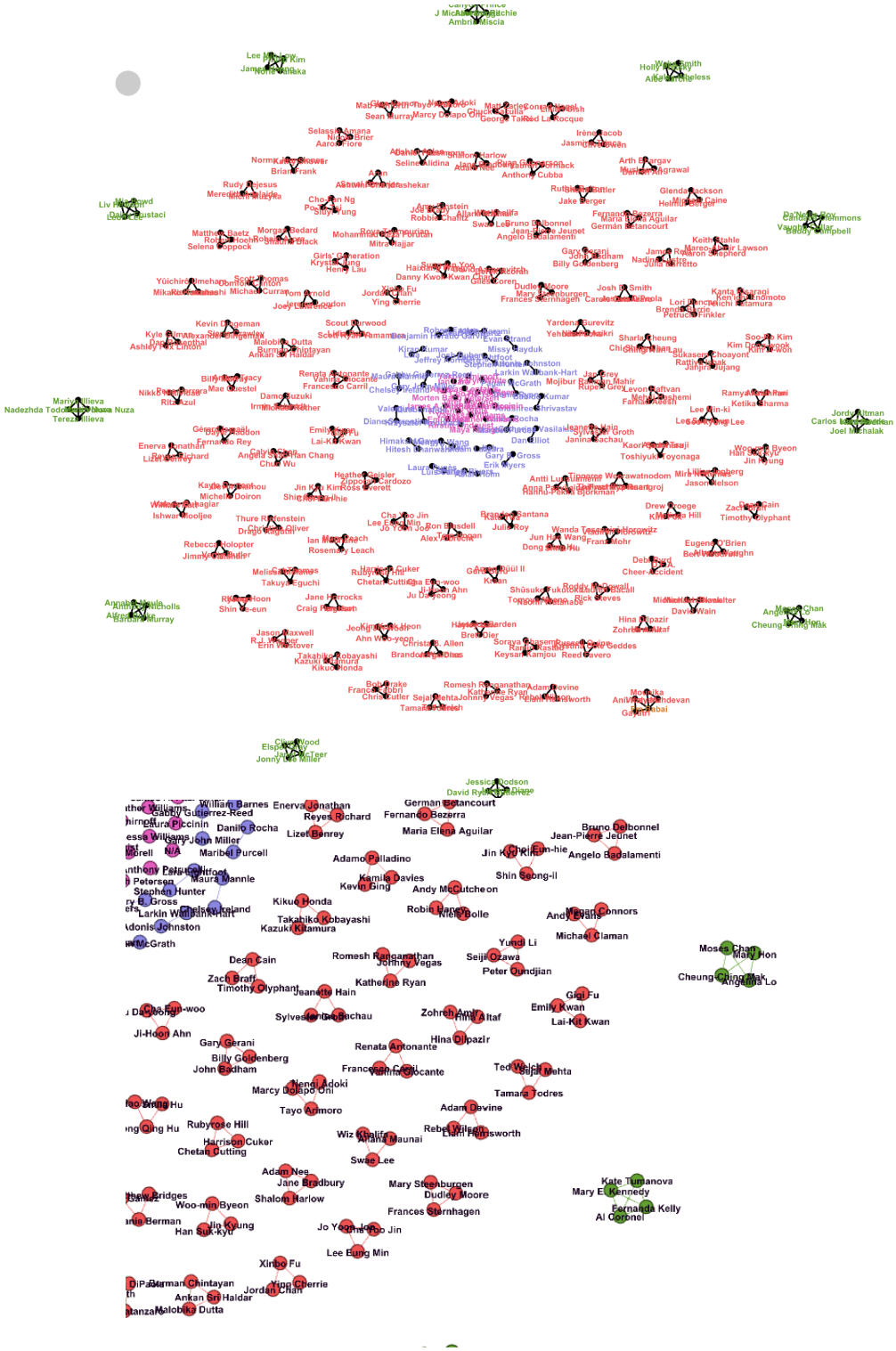


Figure 3.5: Graph visualization in Gephi.

Gephi provides a range of options for visualizing graphs, and in this instance, the Force Atlas layout algorithm has been selected. This algorithm mimics the behavior of physical systems to spatialize a network. In essence, nodes within the graph repel each other akin to charged particles, while edges exert an attractive force on their connected nodes, similar to springs. This interplay of forces results in a dynamic movement that eventually settles into a balanced configuration. The aim of this process is to generate a visually comprehensible representation

of the data, facilitating easier interpretation and analysis.

In this particular setup, the outer periphery of the graph reveals nodes with higher degrees, indicative of movies where data for four distinct actors has been collected. Conversely, nodes situated at the center represent individual actors whose names are the sole data obtained for the respective movies. This spatial arrangement visually emphasizes the distinction between movies featuring multiple actors versus those highlighting a single actor, aiding in the interpretation of the network's structure.

Furthermore, this visualization approach has been enhanced by incorporating a colored scale, where nodes sharing the same degree are assigned identical colors. This color coding provides additional clarity by visually grouping nodes based on their connectivity level. As a result, the graph becomes more informative, enabling easier identification of clusters and patterns within the network.

It's worth noting, and it's curious that there isn't any actor that appears in two different movies. This is mainly due to the volume of movies used and to having chosen a specific genre, because of the limitations of the OMDb API. This causes the graph to lack interaction between groups of nodes.

To obtain the desired measure, Gephi provides a column with various statistical settings.

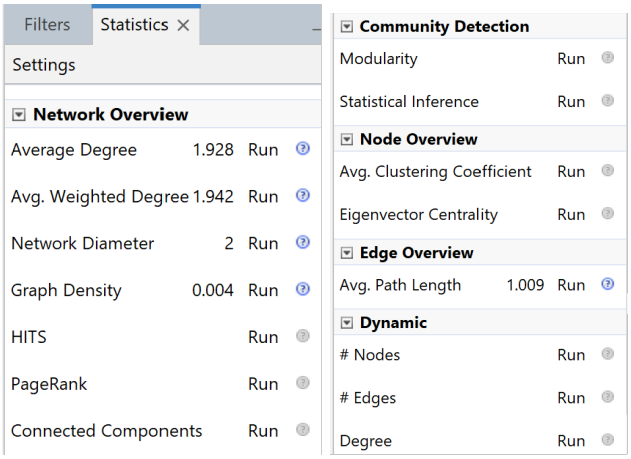


Figure 3.6: Statistical settings.

The defined measures are shown below:

This plot illustrates the degree distribution. It's noticeable that the majority of nodes have a degree of 2, indicating that for most movies, data for three different actors has been collected.

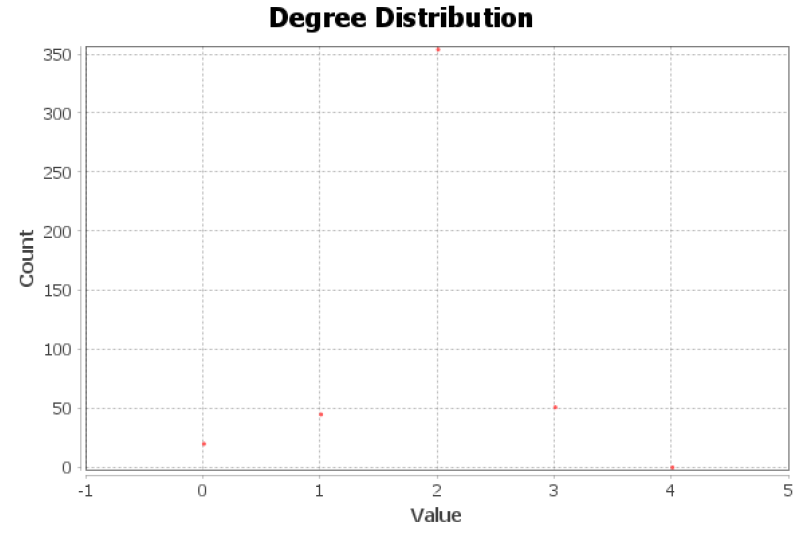


Figure 3.7: Degree distribution.

Density represents the ratio of actual connections (edges) in a graph to the total possible connections. A density of 0.004 in a graph means that only a very small fraction of all possible connections are present in the graph. This suggests a sparse network where nodes are relatively disconnected from each other, with few edges linking them. Sparse networks often indicate a lower level of interaction or connectivity among nodes compared to denser networks.

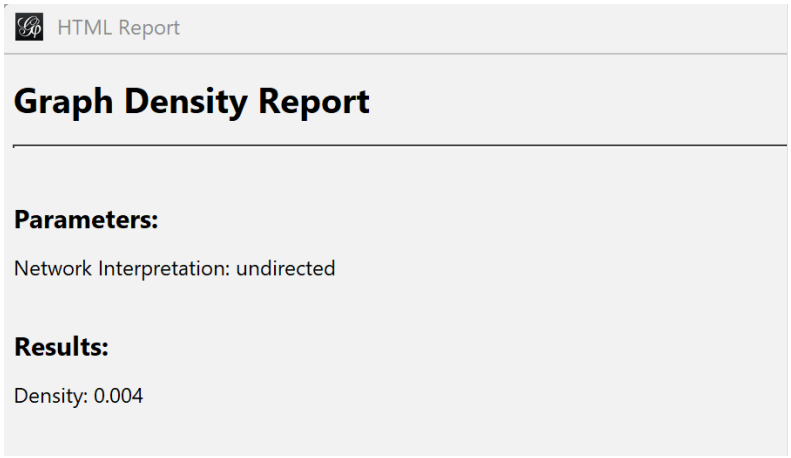


Figure 3.8: Density of the graph.

The clustering coefficient is a measure of the degree to which nodes in a graph tend to cluster together. It quantifies the likelihood that two nodes that are connected to a common neighbor are also connected to each other.

An average clustering coefficient of 0.998 suggests that the graph has a very high level of clustering, meaning that nodes are highly likely to form clusters or triangles.

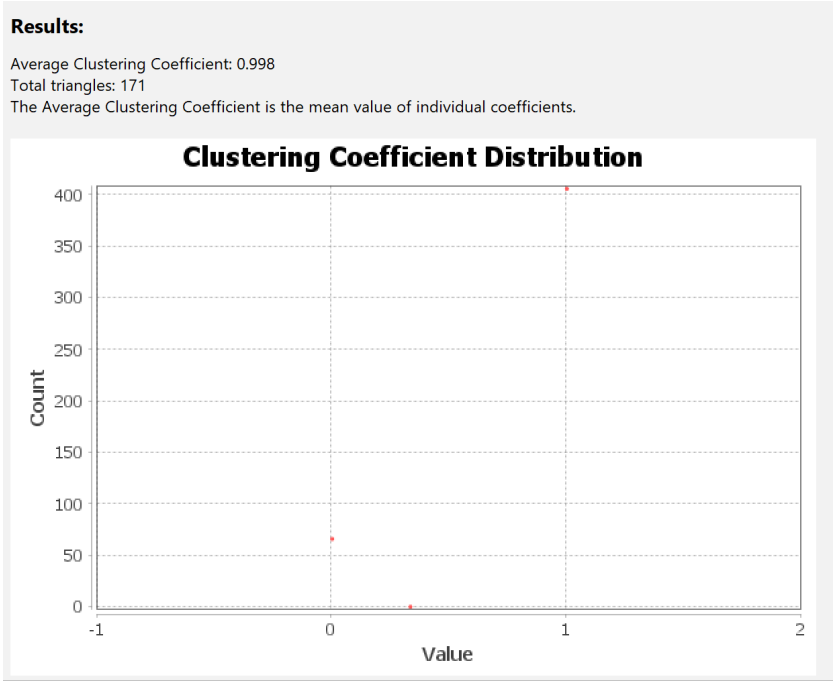


Figure 3.9: Clustering coefficient.

4 CONCLUSIONS

Throughout the project journey, several significant insights emerged from the utilization of the OMDb API for network creation.

Initially, challenges in data access arose due to restrictions on Twitter and LinkedIn APIs, prompting a pivot towards the more accessible and diverse datasets offered by the OMDb API.

Once data acquisition was achieved, careful consideration was given to the selection of a data visualization tool, a pivotal step in the data preparation process.

Subsequently, the analysis of the network within Gephi provided profound insights into its structure and characteristics. Notably, the visualization highlighted various aspects including the distribution of connections, network density, and clustering coefficient. However, the project also acknowledged limitations such as data sparsity and genre-specific biases.

Looking forward, future directions were identified, including the expansion of the dataset, incorporation of additional data sources, and refinement of analysis methods to address these acknowledged limitations. Such steps are essential for further enriching our understanding of actor collaborations within the film industry.

References

[1] X Privacy Policy. "X Privacy Policy," n.d. https://twitter.com/en/privacy.

[2] "LinkedIn privacy policy," n.d. https://es.linkedin.com/legal/privacy-policy?

[3] "OMDB API - the Open Movie Database," n.d. https://www.omdbapi.com/.

[4] Fitzgerald, Anna. "API Calls: What They Are & How to Make Them in 5 Easy Steps." Blog, April 28, 2022. https://blog.hubspot.com/website/api-calls.

[5] "Quick Start," n.d. https://gephi.org/users/quick-start/.

[6] "Supported Graph Formats," n.d. https://gephi.org/users/supported-graph-formats/.

[7] "Updated Gephi Quick Start Tutorial for v 0.9," June 9, 2018. https://www.youtube.com/watch?v=371n3Ye9vVo.

[8] Jacomy, Mathieu, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. "FORCEATLAS2, a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software." Accessed February 7, 2024. https://journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0098679.