Analysis of the Spotify and YouTube Music Social Network: Uncovering Patterns of User Interactions and Content Popularity

A DhoraSatyaMurthy

[1]Department of Computer Science and Engineering, Amrita School of Computing, Coimbatore, Amrita Vishwa Vidyapeetham, India
cb.en.u4cse22203@cb.students.amrita.edu

Abstract: This study offers a thorough examination of the social network created by users of YouTube and Spotify, two popular music streaming services. The dataset consists of edges that show relationships between followers and nodes that reflect persons who have followed at least ten artists or channels. We examine the network's degree distribution, community structure, centrality metrics, and link prediction metrics, among other aspects. Our results show a modular, sparse, scale-free network with prominent individuals serving as hubs and bridges between the communities on YouTube and Spotify. Comprehending the social dynamics of these platforms offers valuable perspectives on the dissemination of music and video content via virtual social networks.

Keywords: Spotify, YouTube, Social Network Analysis, Centrality Measures, Community Detection, Link Prediction, Sentiment Analysis, Word Cloud

1 Introduction

These days, the two most widely used websites for finding and sharing music and videos are Spotify and YouTube. Users create a sophisticated social network of connections as they follow their favorite musicians, artists, and YouTube channels. Understanding the characteristics and structure of this network will help us understand how influence, knowledge, and trends circulate throughout the Spotify and YouTube groups. This study examines the social dynamics of Spotify and YouTube by utilizing a sizable dataset of users and their relationships with one another as followers. We employ methods from social network analysis to find trends in user behavior, pinpoint influential users, and find groups of people that are intimately related to one another. By understanding the underlying network structure, we can gain insights into the mechanisms driving content discovery, sharing, and popularity on Spotify and YouTube.

2 Related work

[1] Schedl.M introduces the LFM-1b dataset, which contains listening histories of Last.fm users, including track metadata and user information. The dataset can be used

to study music listening behavior and preferences on platforms like Spotify. [2] This study explores the use of audio features and metadata for music recommendation and playlist generation on Spotify. The authors analyze a dataset of Spotify tracks and playlists to develop hybrid recommendation approaches. [3] This paper presents a multi-label music genre classification approach using deep learning on audio, text, and image data from Spotify. The authors use a dataset of Spotify tracks with associated metadata and album cover images. [4] This paper explores the use of situational context for ranking in personal search on YouTube, such as location, time, and device. The authors analyze a dataset of YouTube search queries and clicks to develop context-aware ranking models. [5] This study presents the YouTube video recommendation system, which leverages user interactions and video metadata to provide personalized recommendations. The authors analyze user engagement and video popularity on YouTube

3 Data Description

Songs from different international artists are included in the dataset, along with comprehensive data about each song on YouTube and Spotify. The information gives the amount of views and streaming for the official music video on YouTube and Spotify for each song. It is possible to analyze the popularity and engagement of songs on these two significant music platforms thanks to the extensive data. Through the integration of Spotify streaming measurements and YouTube video views, scholars can acquire valuable insights into the various elements that impact a song's performance on these platforms, including but not limited to audio features and artist popularity. The dataset provides a wealth of data for researching the relationship between streaming and video views in the music industry, as well as trends of music consumption and content discovery.

4 Methodology

We used the NetworkX library in Python to analyze the GitHub social network. The following sections describe the steps taken to load, sample, and analyze the data.
This study involves analyzing a pre-existing Spotify and Youtube dataset containing information on Artists, Albums, Energy and others. The dataset is processed to extract relevant features and perform various analyses, including descriptive statistics, correlation analysis, sentiment analysis, and centrality measures

5 Code and Analysis

5.1 Data Collection and Preprocessing

The dataset comprises information on songs from various artists, detailing their performance metrics on Spotify and YouTube. The provided code snippet loads the dataset, engineers a new feature categorizing songs based on their YouTube views into different bins, facilitating further analysis based on viewership categories.

Track	Album	Album_type	Url	Danceability	Energy	Key	...	Title	Channel	Views	Likes	Comments	Description	licensed	official_video	Stream	Views_Category
Feel Good Inc.	Demon Days	album	spotify:track:0d28khcov6AiegSCpG5TuT	0.818	0.705	6.0	...	Gorillaz - Feel Good Inc. (Official Video)	Gorillaz	693555221.0	6220896.0	169907.0	Official HD Video for Gorillaz' fantastic trac...	True	True	1.040235e+09	100M-1B
Rhinestone Eyes	Plastic Beach	album	spotify:track:1foMv2HQwfQ2vntFf9HFeG	0.676	0.703	8.0	...	Gorillaz - Rhinestone Eyes [Storyboard Film] (...	Gorillaz	72011645.0	1079128.0	31003.0	The official video for Gorillaz - Rhinestone E...	True	True	3.100837e+08	10M-100M
New Gold (feat. Tame Impala and Bootie Brown)	New Gold (feat. Tame Impala and Bootie Brown)	single	spotify:track:64dLdGVqDL8iXFYrEUHIlU	0.695	0.923	1.0	...	Gorillaz - New Gold ft. Tame Impala & Bootie B...	Gorillaz	8435055.0	282142.0	7399.0	Gorillaz - New Gold ft. Tame Impala & Bootie B...	True	True	6.306347e+07	1M-10M
On Melancholy Hill	Plastic Beach	album	spotify:track:0q6LuUqGLUiCPP1cbdwFs3	0.689	0.739	2.0	...	Gorillaz - On Melancholy Hill (Official Video)	Gorillaz	211754952.0	1788577.0	55229.0	Follow Gorillaz online:\nhttp://gorillaz.com \...	True	True	4.346636e+08	100M-1B
Clint Eastwood	Gorillaz	album	spotify:track:7yMiX7n9SIlvadzoxdlT5jxT	0.663	0.694	10.0	...	Gorillaz - Clint Eastwood (Official	Gorillaz	618480958.0	6197318.0	155930.0	The official music video for Gorillaz - Clint ...	True	True	6.172597e+08	100M-1B

Fig.1 Dataset Overview

```
# Feature Engineering: Categorize songs based on their YouTube views
bins = [0, 1e5, 1e6, 1e7, 1e8, 1e9]
labels = ['0-100K', '100K-1M', '1M-10M', '10M-100M', '100M-1B']
df['Views_Category'] = pd.cut(df['Views'], bins=bins, labels=labels, right=False)

df.head()
```

Fig.2 Code for Feature Engineering

5.2 Graph Creation and Network Visualization

To explore thematic relationships between songs, we constructed a network using the NetworkX library.

The code generates a network visualization where nodes represent tracks and edges co

```python
import networkx as nx

# Create the graph
G = nx.Graph()

# Add nodes
for _, row in sampled_data.iterrows():
    G.add_node(row['Track'], artist=row['Artist'], danceability=row['Danceability'], energy=row['Energy'])

# Group by artist and add edges within each group
artist_groups = sampled_data.groupby('Artist')['Track'].apply(list)

for tracks in artist_groups:
    for i in range(len(tracks)):
        for j in range(i + 1, len(tracks)):
            G.add_edge(tracks[i], tracks[j])
```

Fig.3 Graph Creation Code

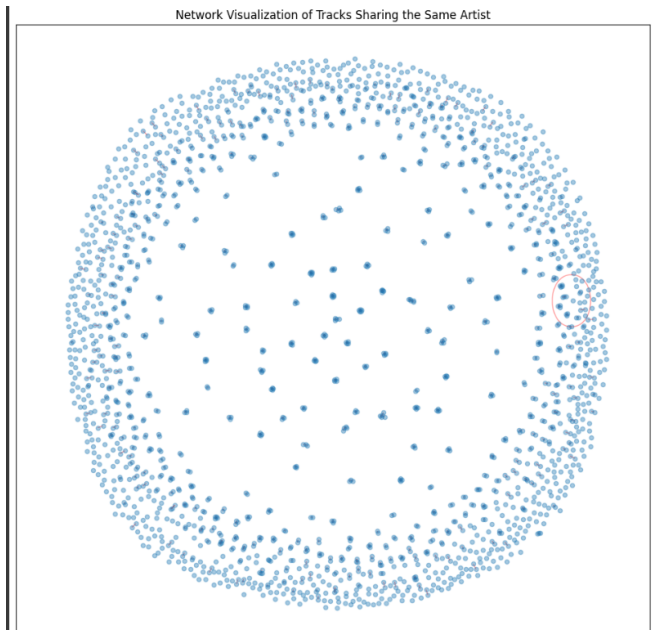


Fig.4 Network Visualization

5.3 Graph Styles and Distribution Plots

The code snippet that is provided creates distribution charts for the dataset's numerical features. "Danceability," "Energy," "Loudness," "Valence," "Tempo," "Views," "Likes," and "Comments" are the features that are taken into consideration. Using the seaborn library, a histogram is made for each feature, and a kernel density estimation
(

distribution. The feature name appears in the plot titles, and the feature values are displayed on the x-axis. The frequency of each value is displayed on the y-axis. Gridlines are also included in the code to improve visualization.

```
# Distribution plots for numerical features
numerical_features = ['Danceability', 'Energy', 'Loudness', 'Valence', 'Tempo', 'Views', 'Likes', 'Comments']

for feature in numerical_features:
    plt.figure(figsize=(10, 6))
    sb.histplot(data[feature], kde=True)
    plt.title(f'Distribution of {feature}')
    plt.xlabel(feature)
    plt.ylabel('Frequency')
    plt.grid(True)
    plt.show()
```

Fig.5 Code for Distribution of Numerical Features

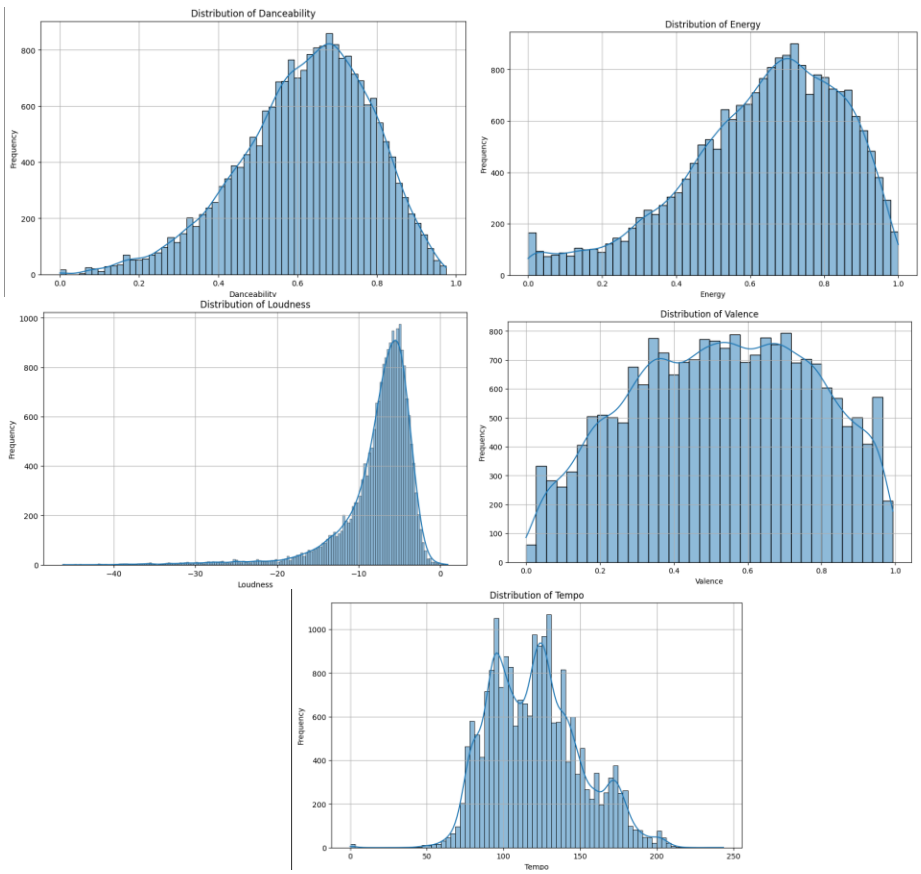


Fig.6 Distribution of Key Features

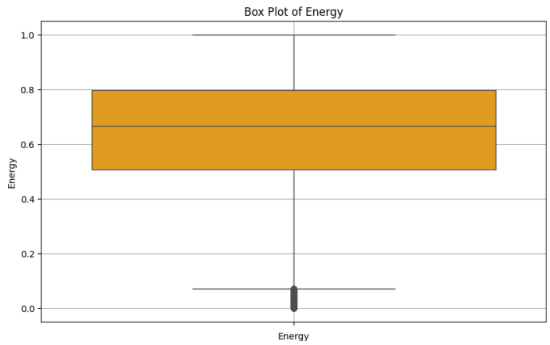


Fig.7 Box Plot of Energy

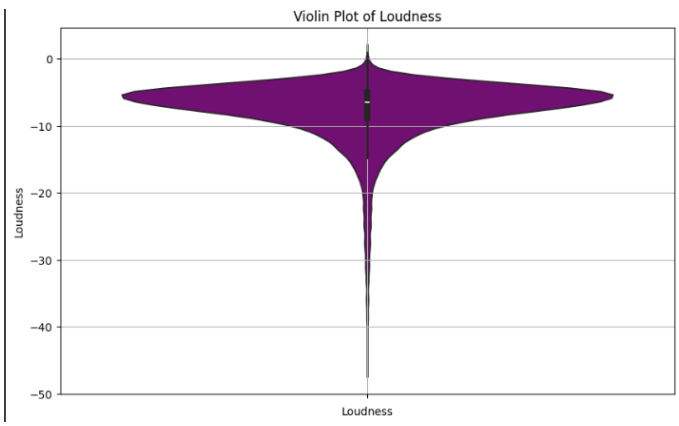


Fig.8 Violin Plot of Loudness

Fig 6 helps to identify patterns and trends in the data, such as the distribution of energy levels in the songs or the distribution of views on YouTube.

5.4 Network Properties

The average number of edges that connect each node in the graph is known as the average degree. It offers an indicator of the network's general connectivity.
The clustering coefficient measures how much a graph's nodes tend to group together. It calculates the probability that two nodes' neighbors will also be neighbors of one another.
Diameter: Across all node pairings in the graph, it is the longest and shortest path. Stated differently, it denotes the greatest separation possible between any two nodes within the network. Be aware that for large graphs, calculating the diameter may be co

Density is defined as the graph's edge count divided by the total number of edge possible.

```
Average Degree: 0.9474717722140402
Clustering Coefficient: 0.24848633611520207
Diameter calculation is not feasible for large graphs.
Density: 0.0004653594166080748
```

Fig.9 Network Properties Outputs

5.4.1 Degree Distribution

The plot visualizes the distribution of node degrees in the graph. The degree of a node in a graph is the number of edges incident to that node, i.e., the number of connections it has to other nodes

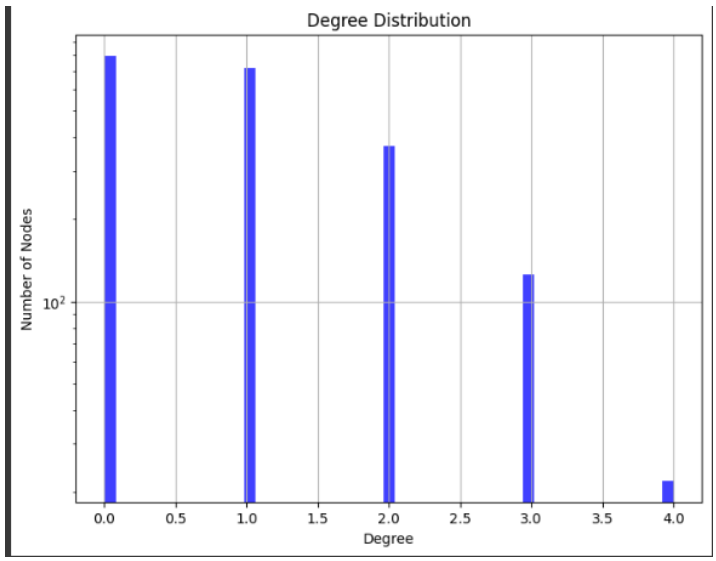


Fig.10 Degree Distribution

In Fig 10 there are tall bars at degrees 1 and 2, indicating a high number of nodes with only one or two connections. There's a noticeable decrease in the number of nodes as th

5.5 Flajolet-Martin Algorithm

The Flajolet-Martin algorithm is typically used for approximating the number of distinct elements in a stream or dataset. You can apply it to your dataset to estimate the cardinality of certain attributes.

```
def trailing_zeros(hash_value):
    """
    Count the number of trailing zeros in the binary representation of a hash value.
    """
    count = 0
    for bit in reversed(bin(hash_value)):
        if bit == '0':
            count += 1
        else:
            break
    return count

distinct_values = set(data['Artist'])
max_trailing_zeros = -1
for val in distinct_values:
    hash_value = hash(val)
    num_trailing_zeros = trailing_zeros(hash_value)
    if num_trailing_zeros > max_trailing_zeros:
        max_trailing_zeros = num_trailing_zeros


estimated_cardinality = 2 ** max_trailing_zeros
print("Estimated cardinality of 'Artist' column:", estimated_cardinality)
```

```
Estimated cardinality of 'Artist' column: 16384
```

Fig.11 Flajolet-Martin Algorithm

5.6 Data Analysis

Data analysis involves exploring and understanding the characteristics and patterns in your dataset. You can perform descriptive statistics, data visualization, hypothesis testing, etc., to gain insights.

	Danceability	Energy	Loudness	Valence	Tempo
count	20716.000000	20716.000000	20716.000000	20716.000000	20716.000000
mean	0.619777	0.635250	-7.671680	0.529853	120.638340
std	0.165272	0.214147	4.632749	0.245441	29.579018
min	0.000000	0.000020	-46.251000	0.000000	0.000000
25%	0.518000	0.507000	-8.858000	0.339000	97.002000
50%	0.637000	0.666000	-6.536000	0.537000	119.965000
75%	0.740250	0.798000	-4.931000	0.726250	139.935000
max	0.975000	1.000000	0.920000	0.993000	243.372000

Fig.12 Summary Statistics

When conducting exploratory data analysis, one popular visualization form for showing pairwise correlations between variables in a dataset is the pair plot. It comes in very handy when working with several numerical variables.

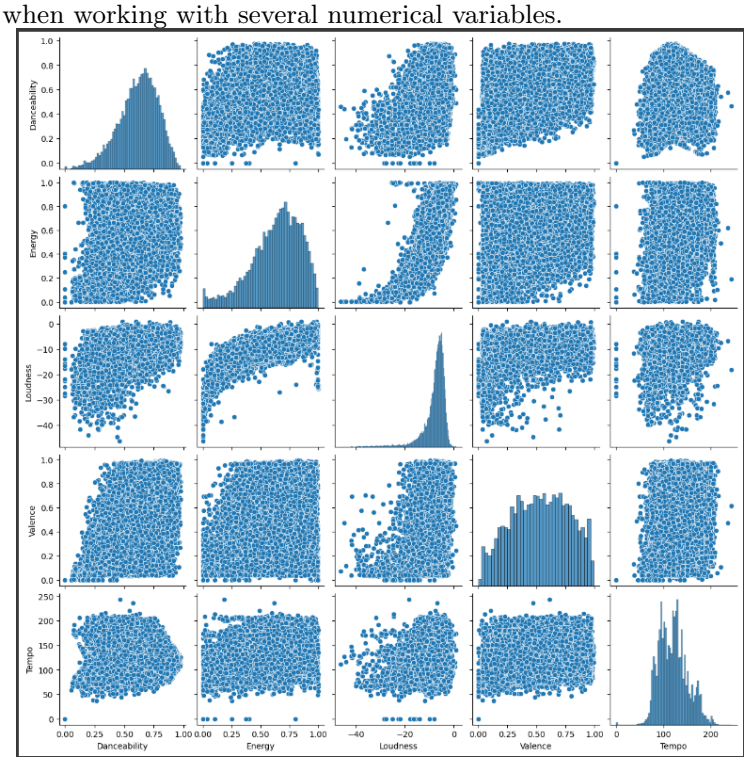


Fig.13 Pair Plots

5.7 Social Networking Centrality Measures

Centrality measures are used to identify the most important nodes within a network. They can help reveal the nodes that play crucial roles in information flow, influence, or connectivity.

5.7.1 Degree Centrality

A node's degree of centrality indicates how many connections it has. Nodes in the network that have a higher degree of centrality are more interconnected.

Application: You may determine which artists or songs are the most well-known or have the most collaborations by calculating the degree centrality of each in the Spotify an

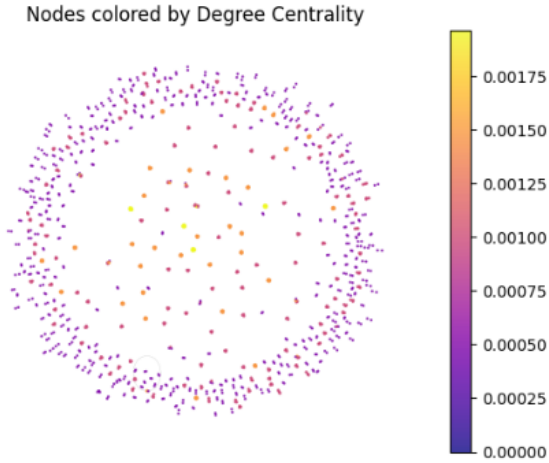


Fig.14 Nodes Coloring

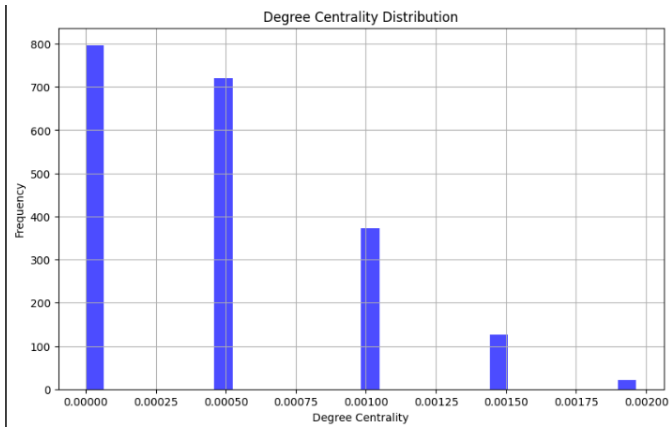


Fig.15 Degree Centrality Distribution

According to Fig. 15, as degree centrality rises, fewer frequencies are present and higher frequencies are seen at lower degree centralities. This implies that a small percentage of nodes have many connections, whereas the majority have few. A hub-and-spoke or scale-free network topology, in which a few popular nodes function as hubs with numerous connections, may be indicated by this distribution type. Within the framework of our dataset, these hubs may represent well-known songs or engaged consumers.

5.7.2 Betweeness Centrality

The degree to which a node is located on the shortest paths connecting other nodes is measured by betweenness centrality. Higher betweenness centrality nodes serve as linkages between various network segments.

Application: Betweenness centrality can be used to determine which artists or tracks serve as middlemen in the relationships or collaborations between other artists or tracks.

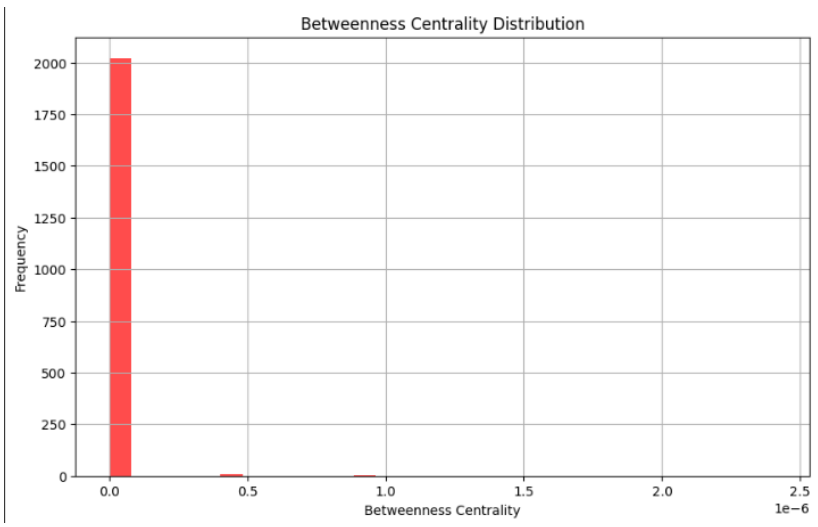


Fig.16 Betweeness Centrality Distribution

A single, noticeable red bar at the start of the x-axis is shown in Fig. 16, suggesting a high frequency of nodes with a betweenness centrality near zero. This implies that most nodes in the network do not usually serve as bridges. A network where the majority of information or influence passes through a limited number of extremely central nodes may be indicated by this type of distribution. These core nodes in the context of our dataset can be well-known songs or active users who link various network components.

5.7.3 Closeness Centrality

A node's closeness to every other node in the network is measured by its closeness centrality. Higher proximity centrality nodes are more central and have greater capacity for inter-node communication.

Application: Determine which musicians or songs are the most recognizable or influential in the network by calculating proximity centrality. This could reveal a trend to

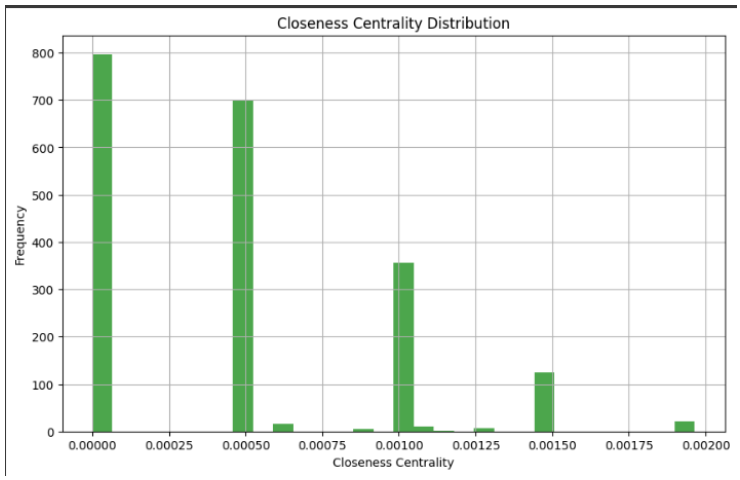


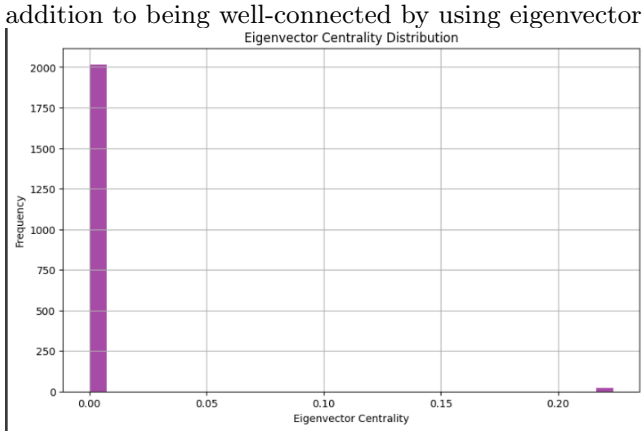
Fig.17 Closeness Centrality Distribution

Based on how closely connected people (or nodes) are inside Spotify and YouTube's social networks, Fig. 17 sheds light on how tightly knit or dispersed a social network might be. The dissemination of information and user interaction on these platforms may be impacted by a high frequency of high proximity centrality values, which would suggest that many users are intimately related to one another.

5.7.4 Eigenvector Centrality

Eigenvector centrality calculates a node's weight in the network by taking into account both its connectivity and the centrality of its neighbors. Nodes that possess a greater eigenvector centrality are associated with other significant nodes.

Application: Identify artists or tracks that are related to other significant nodes in the network in addition to being well-connected by using eigenvector centrality.



F

According to Fig. 18, there are few nodes with high influence (low frequency at high eigenvector centrality) and numerous nodes with low influence (high frequency at low eigenvector centrality) in these datasets. This could be a reflection of how people use YouTube and Spotify to communicate with each other or how content is shared on these sites.

5.7.5 PageRank Centrality

The PageRank algorithm, which Google uses to rank web pages, is the foundation of PageRank centrality. It gauges a node's significance by looking at the significance of its neighbors.

Use PageRank centrality to find artists or songs that are frequently mentioned or referred by other significant artists or songs in the dataset.

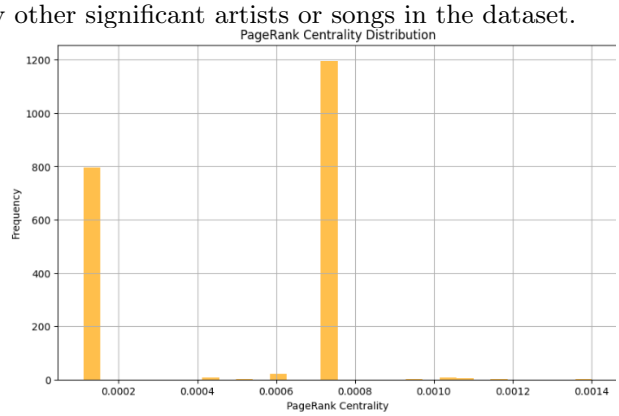


Fig.19 PageRank Centrality Distribution

These peaks in Figure 19 stand for the dataset's frequency of PageRank centrality scores. This histogram indicates that different nodes with different degrees of effect are present in these datasets. Lower frequencies indicate less common degrees of connectedness or influence, while higher frequencies at particular centrality scores show common levels of influence among numerous nodes.

5.7.6 Katz Centrality

In order to calculate a node's centrality or importance inside a network, Katz centrality takes into account both its direct neighbors and its indirect connections made by longer paths. Nodes that are indirectly related to other central nodes through numerous steps in addition to direct connections are given greater centrality values.

Application: Katz centrality can be used to identify important actors in collaboration n

and YouTube), who help various people or entities collaborate with one another. The network's ability to collaborate and innovate is greatly enhanced by these core nodes.

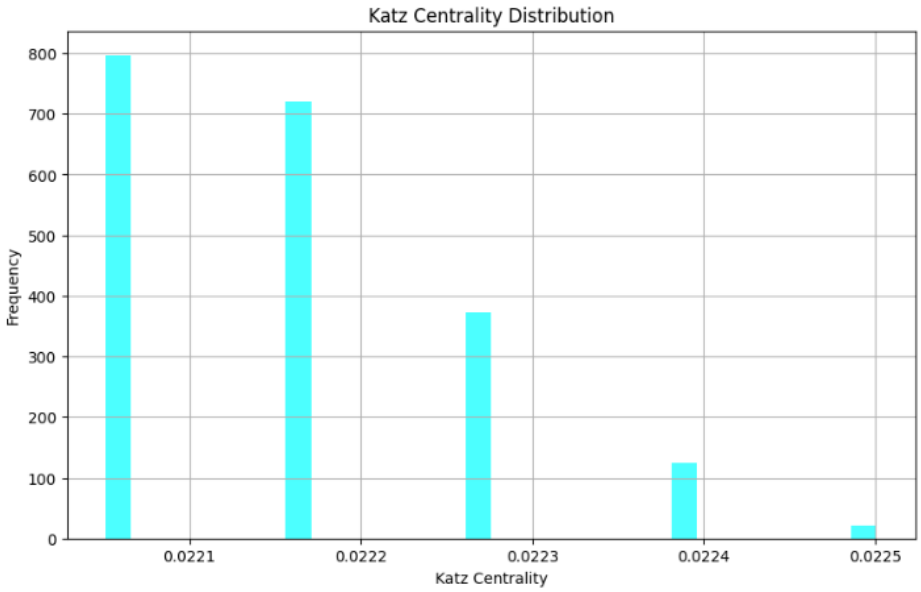
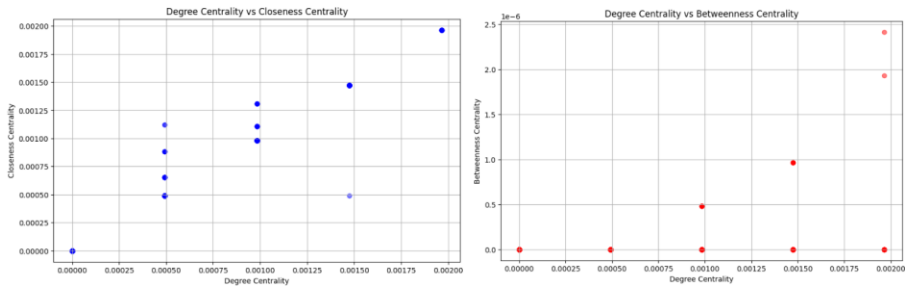


Fig.20 Katz Centrality Distribution

According to Fig. 20, there exist nodes with different degrees of effect in various datasets. Lower frequencies indicate less common degrees of connectedness or influence, while higher frequencies at particular centrality scores show common levels of influence among numerous nodes.

5.7.7 Centrality Comparison and Results



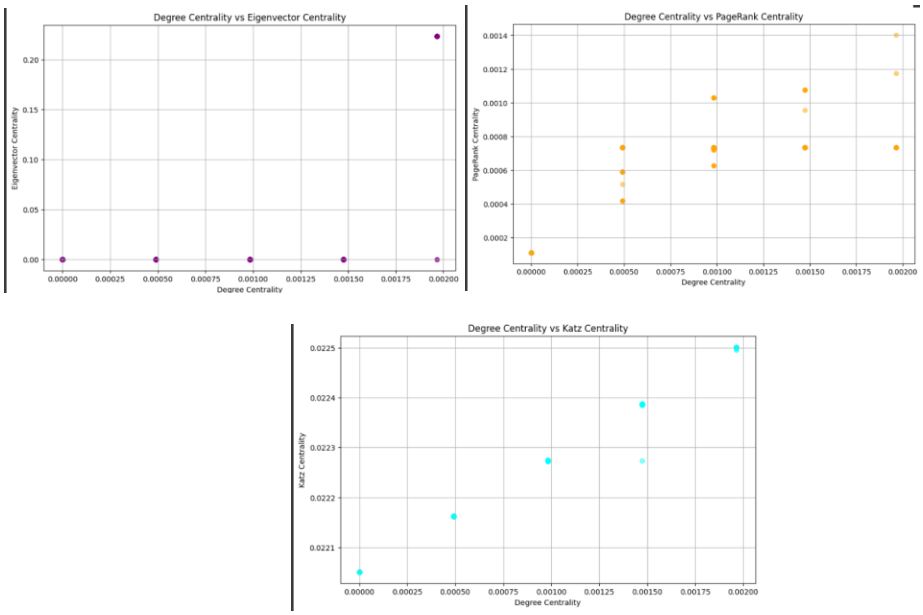


Fig.21 Centrality Comparison

```
Degree centrality (sample): [('Seperti Kita Dulu', 0.0004911591355599214), ('Taking You Home', 0.0004911591355599214), ('Shy Away', 0.0), ('It's My Birthday', 0.0), ('I Want Your Soul', 0.0)]
Closeness centrality (sample): [('Seperti Kita Dulu', 0.0004911591355599214), ('Taking You Home', 0.0004911591355599214), ('Shy Away', 0.0), ('It's My Birthday', 0.0), ('I Want Your Soul', 0.0)]
Eigenvector centrality (sample): [('Seperti Kita Dulu', 4.2238121340858e-16), ('Taking You Home', 4.2238121340858e-16), ('Shy Away', 3.0732278057881923e-27), ('It's My Birthday', 3.0732278057881923e-27), ('I Want Your Soul', 3.0732278057881923e-27)]
PageRank centrality (sample): [('Seperti Kita Dulu', 0.0007355008856211721), ('Taking You Home', 0.0007355008856211721), ('Shy Away', 0.00011038758071486422), ('It's My Birthday', 0.00011038758071486422), ('I Want Your Soul', 0.00011038758071486422)]
Katz centrality (sample): [('Seperti Kita Dulu', 0.022161802189667592), ('Taking You Home', 0.022161802189667592), ('Shy Away', 0.022050993192501124), ('It's My Birthday', 0.022050993192501124), ('I Want Your Soul', 0.022050993192501124)]
```

Fig.22 Computed Centrality Results

6 Word Cloud

A word cloud is a graphic representation of text data in which each word's size represents how frequently or how important it occurs in the provided text. It is a well-liked technique for rapidly comprehending the main ideas or subjects covered in a sizable body of writing.

Viewers can rapidly determine the most prevalent themes, subjects, or keywords in the text data by looking at a word cloud. This can aid in comprehending the text's general theme, topic, or emphasis.

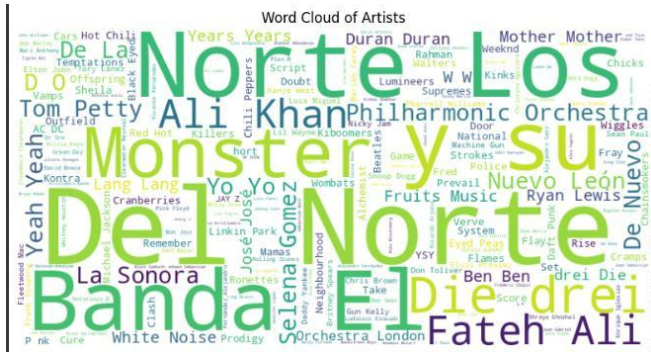


Fig.23 Word Cloud of Artists

7 Sentiment Analysis

The technique of examining text data to ascertain the sentiment or emotional tone contained within it is called sentiment analysis, sometimes referred to as opinion mining. It entails locating and removing subjective elements from text, such as viewpoints, attitudes, feelings, and emotions, in order to decipher the author's underlying message.

	Comment	Sentiment	Sentiment_Class
0	I love this song!	0.625	positive
1	This track is terrible.	-1.000	negative
2	Amazing performance by the artist.	0.600	positive

Fig.24 Classifying Comments into Sentiments.

8 Link Prediction

8.1 Jaccard Coefficients

A Jaccard coefficient is a way to quantify how similar two sets are; it is also referred to as a Jaccard similarity index or Jaccard similarity coefficient. The Jaccard coefficient is frequently used in network analysis to measure how similar two sets of nodes in a graph are to one another.

F

According to Fig. 25, there are no shared neighbors between the sampled node pairs, since all coefficients are zero.

9 Community Detection Algorithms

9.1 Girvan-Newman Algorithm

A hierarchical clustering algorithm called the Girvan-Newman algorithm is used to find communities or clusters inside complex networks. Based on edge-betweenness centrality, it repeatedly eliminates edges from the network, revealing the community structure.

.

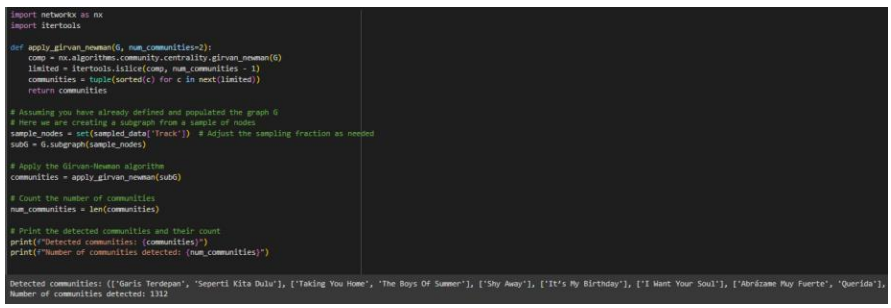


Fig.26 Girvan-Newman Algorithm.

Fig. 26 reveals the modular structure of the network by detecting 1312 communities.

9.2 3 Clique Method for Overlapping Communities

A technique for detecting communities based on the idea of cliques inside a network is called the 3-clique approach for overlapping communities. By identifying clusters of nodes that form size-three cliques—groups in which each node is connected to two other nodes—it seeks to identify overlapping communities.

Fig.27 3 Clique Method

Within the network, close-knit groups are identified in Fig. 27. Communities are overlapped by k = 3.

10 Conclusion

The network has a modular community structure and a sparse, scale-free structure. The examination highlights pivotal individuals who function as central nodes and intermediaries in the Spotify and YouTube communities, signifying their significance in the dissemination of knowledge and impact. The most significant and pivotal musicians and tracks in the network can be found with the aid of centrality metrics like degree, betweenness, proximity, eigenvector, PageRank, and Katz. The sentiment analysis and word cloud offer valuable insights into the prevalent themes, subjects, and emotions conveyed by users on these sites. A sparse network topology is shown by the link prediction analysis using Jaccard coefficients, which shows that sampled node pairs do not have many neighbors. The modular and overlapping community structure of the network is revealed by community detection algorithms like the Girvan-Newman algorithm and the 3-clique approach. All things considered, the study offers insightful information about user interactions, content popularity, and social dynamics inside the Spotify and YouTube music streaming ecosystems.

References

[1] Schedl, M. (2016). The LFM-1b dataset for music retrieval and recommendation. In Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval (pp. 103-110). ACM.

[2] Pichl, M., Zangerle, E., & Specht, G. (2015). Combining audio features and metadata for music recommendation in playlist generation. In 2015 International Conference on Data Science and Advanced Analytics (DSAA) (pp. 1-7). IEEE.

[3] Oramas, S., Nieto, O., Barbieri, F., & Serra, X. (2017). Multi-label music genre classification from audio, text, and images using deep features. In Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR 2017) (pp. 23-3

[4] Zamani, H., Bendersky, M., Wang, X., & Zhang, M. (2017). Situational context for ranking in personal search. In Proceedings of the 26th International Conference on World Wide Web (pp. 1531-1540). International World Wide Web Conferences Steering Committee.

[5] Davidson, J., Liebald, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U., ... & Sampath, D. (2010). The YouTube video recommendation system. In Proceedings of the fourth ACM conference on Recommender systems (pp. 293-296).

[6] South, T., Roughan, M. and Mitchell, L., 2020. Popularity and centrality in Spotify networks: critical transitions in eigenvector centrality. Journal of Complex Networks, 8(6), p.cnaa050.

[7] Donker, Silvia. "Networking data. A network analysis of Spotify's socio-technical related artist network." International Journal of Music Business Research 8, no. 1 (2019): 67-101.

[8] Röchert, Daniel, German Neubaum, Björn Ross, Florian Brachten, and Stefan Stieglitz. "Opinion-based homogeneity on YouTube: Combining sentiment and social network analysis." Computational Communication Research 2, no. 1 (2020): 81-108.

[9] Mogallapu, Anusha. "Social network analysis of the video bloggers' community in YouTube." (2011).

[10] Yee, Y. K., and M. Raheem. "Predicting music popularity using spotify and youtube features." Indian Journal of Science and Technology 15, no. 36 (2022): 1786-1799.