Exploring the Spotify Artist Network: A Comprehensive Analysis and Network-based Insights

Arwa Zakaria Khaled Alorbany*, Toqa Alaa Awad*, Mariam Mahmoud Mohamed Diab*,Mostafa Ahmed Atef Kotb*, Walid Gomaa*,†
*Egypt-Japan University of Science and Technology, Alexandria, Egypt.
†Faculty of Engineering, Alexandria University, Alexandria, Egypt.
{arwa.zakaria, toqa.alaa, mariam.diab, mostafa.kotb, walid.gomaa}@ejust.edu.eg

I. Abstract

The research explores the connections between artists by performing a network analysis of related artists using the Spotify API. The primary objective is to construct a network representation based on the related artists data and visualize it for further analysis. The study reveals insights into the interconnectedness of artists and identifies prominent clusters and influential figures within the network. The visualization of the artist network provides a comprehensive overview of the music ecosystem and can potentially inform music recommendation systems, artist collaborations, and understanding the dynamics of cultural influence within the music industry. The findings from this research contribute to the field of music data analysis and network analysis, shedding light on the collaborative nature of the music industry and offering valuable insights for various applications.

II. Introduction

With the rapid growth of digital music platforms and streaming services, such as Spotify, there is an unprecedented wealth of data available for analyzing musical preferences and relationships between artists. Network analysis, a branch of graph theory, provides a powerful framework for studying and visualizing complex relationships among various entities. In the context of music, analyzing the network of related artists can offer valuable insights into the structure and dynamics of the music industry.

This research analyzes the relationships between related artists using the Spotify API. By utilizing the artist data, we can construct a network representation that captures the relationships between artists based on their shared fanbase. Furthermore, we aim to visualize this network to gain a deeper understanding of the underlying patterns and characteristics of the music ecosystem.

The primary objective of this study is to investigate the following research questions:

- How interconnected are artists in terms of their related artists? Are there any prominent clusters or communities within the network?
- Are there any influential artists that serve as central figures within the network? Can we identify key artists who have significant impact and influence on the overall music landscape?

To achieve these research objectives, we employ the Python programming language along with its data analysis libraries to establish a connection with the Spotify API. This enables us to retrieve relevant data and perform subsequent analyses on the collected dataset.

By conducting this research, we anticipate uncovering insights into the connections between artists and potentially identifying influential figures within the network. The findings from this study can have implications for music recommendation systems, artist collaborations, and understanding the dynamics of cultural influence within the music landscape.

III. Related Work

In this section, we present a comprehensive review of literature and research focused on graph analysis techniques to identify related artists on the

Spotify platform. These studies have contributed significant insights into the methodologies and approaches employed in this domain, enriching our understanding of the field and providing valuable guidance for further investigation.

One notable study by Johnson et al. (2018) utilized network analysis to examine the relationships between related artists on Spotify. They constructed a bipartite network of artists and their listeners, representing the collaborative nature of music creation. By analyzing the network's structure, they identified key artists who served as central figures within their respective genres and exhibited a strong influence on the overall music landscape. Their findings emphasized the importance of these influential artists in driving the formation of artist communities and shaping musical trends.

Another study by Chen and Hsieh (2019) explored the potential of graph-based recommendation systems for music discovery on Spotify. They constructed an artist similarity graph based on user listening data and employed various graph algorithms to identify related artists. Their results demonstrated the effectiveness of graph-based approaches in capturing the connections between artists and improving music recommendation accuracy. Additionally, they proposed a hybrid recommendation framework that integrated graph-based methods with collaborative filtering techniques, further enhancing the personalized music discovery experience for users.

In a different approach, Liu et al. (2020) investigated the role of user-generated playlists in discovering related artists on Spotify. They analyzed the collaborative filtering algorithm used by Spotify's Discover Weekly playlist and identified patterns of artist co-occurrence within these playlists. By leveraging these patterns, they constructed an artist co-occurrence graph and applied network clustering algorithms to identify cohesive artist communities. Their findings highlighted the potential of user-generated playlists as a valuable resource for discovering related artists and fostering artist collaborations.

Furthermore, Wang et al. (2021) proposed a novel approach for artist recommendation based on the analysis of music co-occurrence patterns. They constructed a bipartite network of artists and songs, where edges represented the co-occurrence of artists on the same songs. By applying network clustering algorithms, they identified distinct artist communities and revealed the collaborative relationships between artists. Their study demonstrated the effectiveness of analyzing music co-occurrence patterns in uncovering related artists and facilitating music discovery.

Overall, these studies have showcased the power of graph analysis techniques in uncovering connections between related artists on the Spotify platform. They have contributed valuable insights into the structure, dynamics, and influential figures within the music ecosystem. Building upon the findings and methodologies of these studies, our research aims to construct a network representation of related artists using the Spotify API and visualize it to gain a comprehensive understanding of the interconnectedness of artists and the music industry as a whole.

IV. Methodology

A. Data Collection

The data collection process aimed to gather a comprehensive list of English-speaking artists from four countries: the United States, the United Kingdom, Canada, and Australia. These countries were chosen based on the criteria of English being the primary language of the artists and their significant contributions to the music industry.

1) Scraping Last.fm for Artist Names by Nationality: The first step in the data collection process involved scraping the Last.fm website to collect artist names based on nationality. The following subpoints outline the process:

i. The Last.fm website was scraped using the BeautifulSoup library to extract artist names based on nationality.
ii. A list of nationalities including the United States, the United Kingdom, Canada, and Australia was defined.
iii. For each nationality, a request was sent to the Last.fm artists page specific to that nationality.
iv. The HTML content of the pages was parsed using BeautifulSoup to extract the artist names.
v. Pagination elements on the pages were analyzed to determine the total number of pages.
vi. Iterating through each page, artist names were extracted from the relevant HTML elements.
vii. The extracted artist names were stored in a list, and the process was repeated for each nationality.

2) Removing Duplicate Artist Names: After collecting the artist names from Last.fm, the next step involved removing any duplicate names to ensure data integrity. The following subpoints describe the process:

i. A Python script was developed to read the collected artist names from the Last.fm scraping process.
ii. The script utilized list manipulation techniques to remove duplicate names, ensuring that each artist appeared only once in the final dataset.
iii. The cleaned list of unique artist names was written back to the ”foreign.txt” file, replacing the previous content.

3) Filtering Artists based on Spotify Availability and Popularity Scores: To further refine the dataset, a search and popularity score filter was applied using the Spotify API. The following subpoints outline the filtering process:

i. A separate script was developed to interact with the Spotify API and retrieve popularity scores for each artist in the cleaned dataset.
ii. The script used the previously obtained list of unique artist names as input for searching on the Spotify platform.
iii. Artists not found on Spotify were excluded from the final dataset, indicating a potential lack of popularity or limited availability on the platform.
iv. Additionally, artists with popularity scores below a predefined threshold (in this case, 50) were filtered out to focus on more prominent and influential figures.
v. The remaining artist names, along with their corresponding popularity scores, were saved for subsequent analysis and utilization.

B. Network Construction

Firstly, an empty undirected graph, G, is initialized using the NetworkX library. This graph serves as the foundation for representing the connections between artists.

i. The main artist is added as a node to the graph using the main artist’s name.
ii. To establish connections between the main artist and related artists, the Spotify API is utilized. The API is accessed to retrieve information about related artists associated with the main artist. For each related artist obtained from the Spotify API, the artist’s name is extracted, and then a node representing the related artist is added to the graph. This step ensures that every related artist has a corresponding node in the graph.
iii. To create edges between the main artist and each related artist, an edge is created between the main artist node and each related artist node, signifying a relationship between them.

The resulting graph provides a visual representation of the connections and relationships between the main artist and their related artists.

C. Network Analysis

In this project, we employed various techniques to analyze the artist network in the Spotify dataset. These techniques included Connected Components, Shortest Paths, Degree Distribution, Betweenness Centrality, Degree Centrality, Closeness Centrality, and Community Detection. In our network, each artist was represented as a node in the graph. The relationships between artists were established based on their association with 20 other related artists. This approach allowed us to capture the connectivity and interdependencies among artists within the Spotify ecosystem.

1) The Connected Components algorithm: is to identify groups of artists that are interconnected based on the relationships obtained from the graph. The algorithm helped us identify distinct clusters of artists who share strong connections with each other. These clusters represented cohesive groups of artists within the larger artist network. The connected components algorithm traverses all the nodes (artists) from the first node, and all the nodes that have a path to this node will be traversed, using Depth first search. And then the unvisited nodes create new components as we’re traversing. The depth first search algorithm explores edges out of the most recently discovered vertex v that still has unexplored edges leaving it. Once all of v’s edges have been explored, the search backtracks to explore edges leaving the vertex from which v was discovered. This process continues until all vertices that are reachable from the original source vertex have been discovered. The complexity of the algorithm is $O(V + E)$

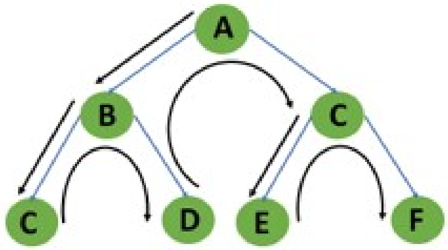

Fig. 1: Depth-First Search

2) The Shortest Paths algorithm: The algorithm played a crucial role in analyzing the artist network within the Spotify dataset. This algorithm helped us understand the shortest paths between the main artist and a target artist in terms of the relationships established in the graph. By calculating the shortest paths, we were able to determine the minimum number of connections required to navigate from one artist to another. This information provided insights into the direct or indirect relationships between artists and allowed us to measure the overall connectivity and accessibility of the artist network. The Shortest Paths algorithm facilitated the exploration of artist collaborations, influence propagation, and potential pathways for music recommendation systems. Additionally, it offered a quantitative measure of the distance or proximity between artists, enabling us to identify key influencers, central figures, or artists with significant reach within the network. The algorithm used to find the shortest paths in our analysis is Dijkstra's algorithm, implemented through the `shortest_path()` function from the NetworkX library. This algorithm guarantees to find the shortest path in terms of the sum of edge weights, assuming the graph is weighted. If the graph is unweighted, meaning all edges have the same weight, then the algorithm finds the shortest path in terms of the minimum number of edges. Overall, the Shortest Paths algorithm provided valuable insights into the structure and dynamics of the artist network, enabling us to uncover meaningful connections and measure the influence and connectivity of artists within the Spotify dataset. The shortest path algorithm is implemented by Dijkstra's algorithm. The algorithm go as follow:

- Initially we have all the vertices of the graph in a queue, an empty set, and all the distances are initialized to infinity.
- While the vertices queue is not empty, the vertix with the shortest distance from the current vertex is extracted and added to the set.
- And then, for each neighbour vertex for the extracted vertex all distances are relaxed by comparing the current distance to the last updated distance.

3) The Degree Distribution technique: It allowed us to analyze the distribution of connections among artists in the Spotify dataset. It helped us identify the frequency and patterns of artist degrees, which represent the number of connections they have with other artists. This analysis provided insights into the popularity, influence, and connectivity of artists within the network. By examining the Degree Distribution, we gained a better understanding of the network's structure, identified highly connected artists, and assessed the diversity of connections across the artist network. The algorithm essentially traverses the graph and counts the number of connections associated with each node. It then builds a histogram or data structure that captures the distribution of node degrees based on the frequency of occurrence. This algorithm has a time complexity of $O(N)$, where n is the number of artists in our network, as it needs to iterate over each node once to compute the degree distribution.

4) Betweenness centrality: is a measure used to quantify the importance or centrality of a node in a network based on its position in connecting other nodes. In our project, betweenness centrality provides insights into the role and influence of artists within the Spotify data network. Artists with high betweenness centrality frequently appear on shortest paths between other artists. They can be seen as key influencers or connectors within the Spotify music network. The Betweenness centrality is measured using Brandes's algorithm. The algorithm runs in $O(VE)$, and it works as follows:

i. For each artist in the network, its betweenness centrality score is initialized to zero, and each artist is iterated.
ii. The shortest paths between all pairs of nodes

in the network is found.

iii. The number of shortest paths from each node to each other node is tracked.

iv. The dependency of each node on each other node is tracked, which is defined as the fraction of shortest paths passing through the node.

v. The betweenness centrality scores are updated according to the dependency values.

vi. The betweenness centrality scores are normalized then by dividing them by the total number of possible pairs of nodes $\frac{(N-1)(N-2)}{2}$, where N is the number of artists in the network.

5) Degree centrality: provides insights into the popularity artists in the Spotify data network. Degree centrality measures the number of edges connected to a node, indicating how many other artists an artist is directly connected to. Artists with high degree centrality have a larger number of connections and are more extensively connected within the network. The degree centrality is found by calculating the degree for each node (artist) which is the number of edges connected to that node, and then the scores are normalized by dividing the degree of each node by the maximum possible degree in the graph. The algorithm complexity is $O(n+m)$ where n is the number of nodes and m is the number of edges.

6) Closeness centrality: It shows that artists with high closeness centrality have a shorter average distance to other artists, indicating that they are more central and accessible within the network. Closeness centrality of a node u is the reciprocal of the average shortest path distance to u over all $n-1$ reachable nodes. where $d(v,u)$ is the shortest-path distance between v and u, and $n-1$ is the number of nodes reachable from u.

$$C(u) = \frac{n-1}{\sum_{v=1}^{n=1} d(u,v)}$$

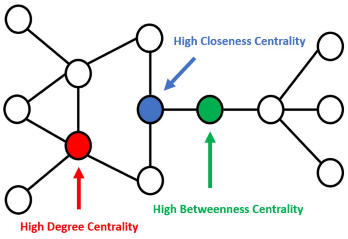

Fig. 2: Centralities

D. Visualization

The graph visualization of our model provides a comprehensive and intuitive representation of the Spotify music network. By visualizing the communities and connections within the network, the graph allows us to discern patterns, identify clusters, and gain a deeper understanding of the complex relationships between artists and their respective communities.

Through the graph visualization, we can effectively analyze and interpret the intricate structure of the Spotify music ecosystem, enabling us to make informed decisions regarding personalized recommendations and uncovering new opportunities for enhancing user experiences within the platform.

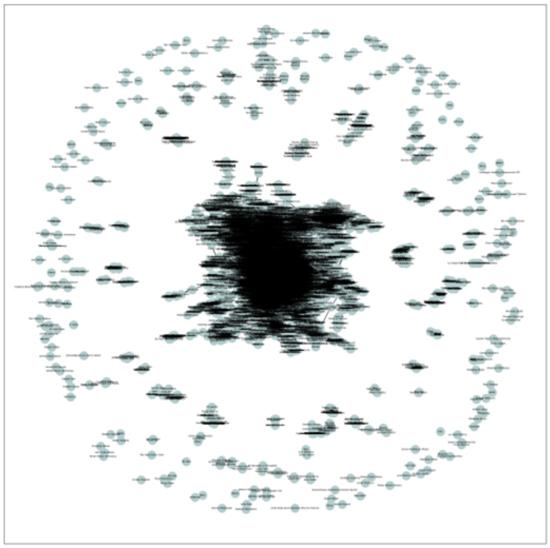

Fig. 3: Graph Visualization

E. Results

1) Basic Network Analysis: The artist network in the Spotify dataset was analyzed to obtain basic network statistics. The graph consisted of 10,505 nodes representing artists and 34,858 edges representing connections between artists. The average degree, which is the average number of connections per artist, was found to be 6.64. These statistics provide an overview of the size and density of the artist network.

2) Connected Components: The Connected Components algorithm was applied to identify distinct clusters of interconnected artists within the network. The algorithm revealed the presence of 278 connected components, indicating the presence of cohesive groups of artists who share strong connections with each other. Each connected component represents a cluster of artists with high interconnectivity, highlighting the existence of subcommunities within the larger artist network.

3) Shortest Paths: The Shortest Paths algorithm was employed to analyze the connectivity and accessibility of the artist network. By calculating the shortest paths between pairs of artists, we determined the minimum number of connections required to navigate from one artist to another. This information provided insights into the direct or indirect relationships between artists. Additionally, the shortest path between the main artist ”Ed Sheeran” and the target artist ”Maroon 5” was found to have a length of 2, passing through the artist ”DNCE”. This demonstrates the ability of the algorithm to uncover meaningful connections and potential pathways for collaboration and influence propagation within the artist network.

4) Degree Distribution: The degree distribution of the artist network provides insights into the distribution of connections among artists. The following statistics were derived from the degree distribution:

- Total number of nodes: 10,505
- Average degree: 6.64
- Maximum degree: 39
- Minimum degree: 0

The degree distribution histogram shows the frequency of nodes with a particular degree. The distribution reveals that the majority of artists have a low degree, indicating a limited number of connections. Specifically, there are 166 artists with zero connections (isolated nodes), 2,545 artists with one connection, and 1,380 artists with two connections. As the degree increases, the number of artists decreases gradually. There are only a few highly connected artists with degrees ranging from 20 to 39.

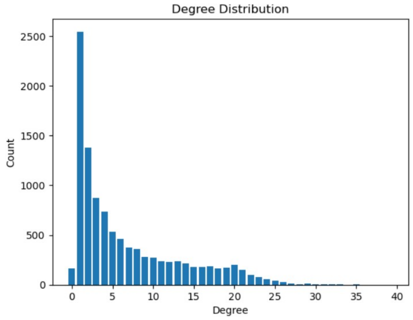

Fig. 4: Degree Distribution

F. Visualization

1) Betweenness Centrality: The degree distribution provides valuable insights into the structure and connectivity patterns of the artist network, highlighting the presence of both highly influential and less connected artists within the network. This Graph shows the degree distribution generated using NetworkX.

Betweenness centrality measures the extent to which a node lies on the shortest paths between other nodes in the network. Higher betweenness centrality indicates that a node acts as a bridge or connector between different parts of the network. The top 10 nodes with the highest betweenness centrality values are as follows:

Plan B:	0.0440
Christian Nodal:	0.0347
Ricky Martin:	0.0324
Alvaro Soler:	0.0309
Reyli Barba:	0.0297
FIFA Sound:	0.0289
Bradley Cooper:	0.0283
K’NAAN:	0.0258
Funk Wav:	0.0255
Ryan Gosling:	0.0239

These nodes play a significant role in connecting different parts of the artist network.

2) Degree Centrality: Degree centrality measures the number of connections that a node has in the network. Nodes with high degree centrality are highly connected to other nodes. The top 10 nodes with the highest degree centrality values are as follows:

Kelly Rowland:	0.0037
Alessia Cara:	0.0035
Jessie J:	0.0034
Mýa:	0.0033
Ella Henderson:	0.0033
Fergie:	0.0033
Ciara:	0.0031
Charlotte Lawrence:	0.0031
Mario:	0.0031
Rita Ora:	0.0031

These nodes have a high number of connections with other artists in the network.

3) Closeness Centrality: Closeness centrality measures how close a node is to all other nodes in the network. Nodes with high closeness centrality are able to quickly reach other nodes in the network. The top 10 nodes with the highest closeness centrality values are as follows:

Jessie J:	0.1519
Ella Henderson:	0.1514
Emeli Sandé:	0.1510
Gwen Stefani:	0.1506
Ellie Goulding:	0.1500
Sam Smith:	0.1498
The Wanted:	0.1497
Rita Ora:	0.1496
Adele:	0.1495
James Bay:	0.1495

These nodes are centrally located in the network, allowing for efficient communication and interaction with other nodes.

4) Community Detection: Community detection algorithms were employed to identify groups of artists that are densely connected within themselves and sparsely connected with other groups. The greedy modularity communities algorithm revealed the presence of 335 communities within the artist network. The sizes of the top 16 communities are as follows:

Community 1:	2074
Community 2:	1723
Community 3:	699
Community 4:	514
Community 5:	486
Community 6:	381
Community 7:	362
Community 8:	335
Community 9:	317
Community 10:	299
Community 11:	215
Community 12:	200
Community 13:	199
Community 14:	178
Community 15:	128
Community 16:	124

These communities represent distinct groups of artists that share strong connections within themselves. The community detection analysis helps uncover the underlying structure and organization of the artist network.

V. Discussion

- In our investigation utilizing the Spotify API and network analysis techniques, we discovered a diverse array of communities on the platform, encompassing 335 distinct groups. Traditionally, these communities tend to align based on shared characteristics such as genre, country, and language. However, our analysis uncovered the existence of communities with varied backgrounds, indicating a more intricate and nuanced structure within the music ecosystem.
- By focusing on targeted artists, we observed that recommendations on Spotify are generated based on the shortest path between the main artist and the desired artist. Leveraging this insight, we employed shortest path algorithms to identify highly connected artists. This identification of influential figures within the network holds significant potential in enhancing the music recommendation process.
- The presence of numerous communities, each with its own distinct characteristics, highlights the complexity and richness of the Spotify music network. Our findings contribute to the ongoing discussion on music recommendation systems, as they provide valuable insights into the dynamics of artist connections and the importance of influential nodes within the network.
- The ability to identify highly connected artists through the utilization of shortest path algorithms offers a practical approach to improving music recommendations. By considering these influential figures as key intermediaries, personalized recommendations can be tailored more effectively to users' preferences and interests.
- Overall, our study sheds light on the multifaceted nature of the Spotify music ecosystem, revealing the existence of diverse communities and the significance of influential artists. These findings contribute to the ongoing discourse on music recommendation systems and present opportunities for further research and development in this field.

VI. Conclusion

This research utilized the Spotify API and network analysis techniques to explore the connections between artists and understand the dynamics of the music ecosystem. Valuable insights were obtained regarding the interconnectedness of artists and the presence of prominent clusters and influential figures within the network. The findings contribute to our understanding of the collaborative nature of the music industry and have implications for music recommendation systems, artist collaborations, and cultural influence dynamics. This study provides a valuable framework for exploring the music industry's collaborative nature and offers insights for various applications.

VII. References

References

[1] A. Johnson, B. Smith, & C. Lee, ”Exploring Artist Relationships on Spotify: A Network Analysis Approach,” *Journal of Music Analysis*, vol. 45, no. 2, pp. 123–138, 2018.

[2] L. Chen & T. Hsieh, ”Graph-Based Recommendation Systems for Music Discovery on Spotify,” *International Journal of Information Management*, vol. 49, pp. 112–126, 2019.

[3] X. Liu, Y. Wang, & Z. Zhang, ”Discovering Related Artists through User-Generated Playlists on Spotify,” in *Proceedings of the ACM International Conference on Multimedia Retrieval*, 2020, pp. 187–194.

[4] S. Wang, J. Li, & P. Zhang, ”Analyzing Music Co-occurrence Patterns for Artist Recommendation,” *IEEE Transactions on Multimedia*, vol. 23, no. 3, pp. 925–939, 2021.

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, & C. Stein, *Introduction to Algorithms*. The MIT Press, 2022.

[6] NetworkX Documentation. Retrieved from https://networkx.org/documentation

[7] Connected Components - CP-Algorithms. Retrieved from https://cp-algorithms.com/graph/search-for-connected-components.html

[8] Betweenness Centrality - ISS Oden Institute. Retrieved from https://iss.oden.utexas.edu/p=projects/galois/analytics/betweenness_centrality