Spotify Artist Collaboration Network Analysis

Nichole Constanzo

1. Introduction

This goal of this project is to analyze a network of artists collaborations on Spotify via machine learning techniques to make artist popularity predictions and to find groups of similar artists. This will be achieved through the analysis of centrality measures, regression modeling and clustering. The way music has been distributed to users has changed from physical to digital with music streaming platforms now at the highest in usage. Recent studies such as (South, Roughan, & Mitchell, 2020) examine the effect of popularity bias between music genres on Spotify through the monitoring of the eigenvector centrality scores as low popularity artists are removed from the network. Another study (Donker, 2019) focuses on a more niche section of the Spotify network (Dutch drum and bass artist Noisia and their related network) and measures influence and connections via other centrality measures such as closeness, betweenness, and degree centrality. The (Donker, 2019) study seeks to use the results of the study to answer what opportunities, limitations, collaborations, and obstacle workarounds are available for artists in the networks. My Spotify network analysis will address some of the limitations gaps in previous works by incorporating the use of more centrality measures (degree, betweenness, closeness, eigenvector, and PageRank). This project will be using them as features in machine learning models to predict artist popularity and identify artist communities through clustering. This project combines network analysis with predictive modeling to add onto the understanding of artist influence in the Spotify network. The dataset in this project consists of 156,422 artists as nodes and 300,386 collaborations as edges. The features available are Spotify id, name, followers, popularity, genres, and chart hits but the last two categories aren't relevant in the context of this project. The Spotify id is the unique artist identifier assigned by Spotify and this will help us link the artist metadata stored in the nodes to the edges for our centrality measures' top 5 lists. Followers (number of users following the artist) and popularity (artist popularity score) will be our focus in our logistic regression and cluster analysis.

2. Dataset Overview

The dataset consisted of two files: Nodes and Edges. There were 156,422 nodes representing artists, and 300,386 edges representing artist collaborations. The edge file is two columns of Spotify artist unique ids and represents the connections between different artists in the form of music collaborations. The node file includes artist metadata such as Spotify id, name, followers, popularity, genres, and chart hits. The Spotify id is the unique artist identifier assigned by Spotify, followers are the number of users following the artist, popularity is the artist's popularity score according to Spotify, genre is a list of the categories of music the artists' songs fall into, and chart hits is a list showing the number of Spotify chart hits in different countries. The amount of followers each artist has ranges from 0 to 102,156,853 and the average number of followers each artist had was 86,222 users. The popularity score ranges from 0 to 100 and the average popularity score was 21.15. When previewing the data there were 4 missing values in the followers column and 136,724 missing values in the chart hits column. I decided to fill these missing values in with zeros rather than deleting the rows because I did not want to disturb the structural integrity of the artist network by removing nodes.

3. Data Preprocessing

Due to issues with loading the PySpark library into Jupyter Notebook and unfamiliarity with Google Collab, I decided to load the dataset into Databricks, which has PySpark built into it. For Databricks, I loaded both the node and edge files into the volumes section of data ingestion as a non-tabular dataset. Upon inspection of the dataset properties the first preprocessing step that I took was to fill in the missing values with zero, as to not disturb structural integrity of the graph via row deletion. The second preprocessing step was to fill a string "vocal'" that presented itself in the descriptive stats of the followers column with a zero and change the column to a numerical attribute. For this analysis, I used a subset of 20,000 artists because a smaller subset was necessary to avoid extraneously slow computation times. To determine the subset, I began at 10,000 nodes and ran the degree distribution for each sampling method (Random Walk, BFS, and Forest Fire), increasing by 10K in each run to capture as much of the original network as possible. At 80,000 nodes (a little more than half of the original dataset) I decided to stop because I was wary of later computationally expensive operations. Once I started to run the network properties such as diameter and the centrality measures, the processing time for the code was much slower. I then began testing reduced sample sizes from the 80,000 by quantities of 10k at each run. Ultimately, I ended up with 20,000 nodes (13% of the original) where run times began to be reasonable. I tested amounts between 20,000 and 30,000 but there was still a significant difference in computation speeds between these two sample sizes. Although a 20,000-node dataset doesn't run the quickest, its computation speeds are still reasonable and preserves my intention of retaining as much of the original dataset as possible. To preserve the graph's topology, a non-random sampling approach needed to be used for the subset. My sampling options were reduced to random walk, breadth first search (BFS) method, and forest fire sampling. The choice of subset method relied on a comparison of degree distribution graphs where the sampling graphs that generated the closest degree distribution to that of the original network's would be selected. Ultimately after comparing the degree distribution graphs, BFS was the closest in similarity to the original meaning it can be concluded that the BFS sampling is representative of the larger graph. Along with the graph structure I also printed the average degree for each sampling method. The original graph's average node degree is about 3.92, whereas the random walk had an average of around 5.42, the forest fire had an average of 5.39 and BFS had an average of 5.20 (closest to the original). Since each time the code is run the subset of nodes created via the sampling method changes, I immediately created a data frame to keep our results constant for the rest of project analysis.

3. Network Analysis

For the network analysis, I calculated and compared the artists' ranking results for each of the following centrality measures degree centrality, betweenness centrality, closeness centrality, eigenvector centrality, and PageRank. Degree centrality measures how many collaboration connections an artist has with other artists. An artist with a high degree centrality is one that frequently features with different artists. Betweenness centrality measures how often an artist lies on the shortest path between two other artists. An artist with high betweenness probably does collaborations across different music communities and genres. Closeness centrality measures how close (via collaboration distances) artists are from each other. An artist who has high closeness centrality is a central to the network and can quickly reach other artists via the collaboration network. Eigenvector Centrality measures the influence of artists based on the number of and influence of the collaborators they are connected to. An artist with eigenvector centrality is likely themselves to be very famous and/or famous by proxy via connections to other popular/famous artists. PageRank is originally Google's webpage ranking system, but in this

context, it measures artist's importance based on the number and the quality (high-status) of the collaboration artist. An artist with high page rank is important due to their connected peers (social groups). Among the top 5 influential artists lists, there are a few names that appear in multiple centrality lists. Johann Sebastian Bach and Traditional are in a top 3 position for degree centrality, betweenness centrality, closeness centrality, and page rank, suggesting that they are the *most influential* in this sampled network. John Williams and Andrea Bocelli appear in between centrality, closeness centrality and page rank lists suggesting that they are very central to the network but not as influential as Bach and Traditional. The Eigenvector centrality has a completely different set of artists and genre from all the other lists that were dominated by classical music artists. The Eigenvector list consists of Ty Dolla $ign, Lil Wayne, French Montana, Chris Brown, and Gucci Mane. These results suggest that eigenvector artists are influential due to their high-level connections through collaborations with high status artists. These artists are part of a prestigious influence network rather than being highly central or highly connected in the network like the classical music artists.

The average degree of a node in the network is 5.46 which indicates that on average artists have about 5 to 6 collaborations with other artists. This is a very low average degree which indicates that the network is sparse. The network's spareness is also proved through the graph's edge density result of 0.0002729836, meaning that only 0.027% of all possible connections/edges exist. Other interesting graph properties include that the graph contains 1 connected component, therefore the Spotify network does not have any disconnected subgraphs and so it is a unified network. The diameter (longest shortest path between nodes) of the BFS subgraph of 20,000 nodes was 8. This indicates that although this is a sparse network, the nodes are still fairly well connected (maximum 8 artists collaborations away from every other artist).

5. Predicting Popularity with Linear Regression

Through the use of PySpark's Linear Regression I was able to create a predictive model that estimated Spotify artist popularity. The features used in this model included the number of followers and the centrality measures: degree centrality, betweenness centrality, closeness centrality, eigenvector centrality, and PageRank. The use of all these features will give a well-rounded insight into different types of influence levels and connections that the artists possess. For the ML training and testing, the BFS subset of 20,000 nodes was split 80% for training (roughly 16,000 nodes) and 20% for testing (roughly 4,000 nodes).

In the evaluation of the testing dataset's linear regression model results, it received an RMSE of 18.99 and an R^2 of 0.25. The RMSE indicates that the model isn't super precise, missing the true popularity score by 19 points (positively or negatively). The R^2 indicates that the model only captures 25% of the variance that contributes to the artists' popularity scores. It would seem that there are more factors outside of the number of followers and network centrality that contribute to the popularity scores. When analyzing the coefficients, the feature that had largest positive influence on the popularity score was degree centrality. This indicates that the more collaborations an artist has with other artists, the higher their popularity score. Closeness centrality and the number of followers were the following two top positive features (respectively) that helped improve popularity scores in lesser amounts.

6. Clustering Users with K-Means

To analyze patterns in the artist network I used K-means clustering through the PySpark ML library. This K-means model utilizes the standardized version of the previous features used in the logistic regression: number of followers degree centrality, betweenness centrality, closeness centrality, eigenvector centrality, and PageRank. The optimal number of clusters was determined through the Elbow Method, by both analyzing numerically how the WSSSE (within-cluster sum of squared errors) changed across different k clusters and the printed graph of the WSSSE. I chose 6 clusters for k, because that is the last point where the WSSSEs stopped dropping drastically and started to level out. Analysis of the clusters' average and min/max values of the input features showed that: Cluster 5 included the high-profile artists. Cluster 5 consisted of only 46 artists, but it had the highest popularity scores and highest number of followers per artist. Cluster 3 is the largest cluster with 11,347 artists but had very low centrality scores across the board. Cluster 3 seems to be where small, lesser-known artists are grouped. Cluster 1 is very interesting because it only contains 2 artists: Johann Sebastian Bachand Traditional, who scored in the top 3 of every centrality measure list we created earlier. Cluster 1 must be our top influencer artists of the network, with an average popularity of 67 and average following of 1.87 million. Clusters 4 and 2 lie in the middle of range of all the clusters.

7. Conclusion

Although the network was sparse, the network analysis proved that it was still quite connected with a network diameter of 8. After calculating top artists across centrality measures we found that Johann Sebastian Bach and Traditional were the top influential and central nodes in the network by ranking in the top 3 list of each centrality. The classical music artists dominated the degree, betweenness, closeness, and PageRank rankings, whereas mainstream rap artists appeared in only the eigenvector centrality top 5 list. This highlighted a distinction between artists that are highly connected vs artists that have high-status connections. The linear regression model with evaluation scores of $R^2 = 0.25$ and RMSE = 18.99, revealed that popularity cannot be entirely predicted by just followers and network centrality alone and that there are additional factors that need to be taken into consideration. The linear regression model also revealed that degree centrality (the number of collaborations per artist) was the strongest positive indicator from our feature set for popularity. The K-means clustering identified 6 groups of artists based on influence and popularity. Cluster 1 was the top network influencers, Cluster 5 was the high-profile artists with high followers and popularity, Clusters 2 and 4 were the mid-level artists. Cluster 3 was a large group of smaller, lesser-known artists. Overall, this study has shown that artist collaborations can be used to understand popularity and influence in the Spotify artist network. Some applications of this study could be used to increase artist popularity and visibility through strategic collaborations or optimize music recommendations to users based on their favorite artists' connections.

Related Work References

South, T., Roughan, M., & Mitchell, L. (2020). Popularity and Centrality in Spotify Networks: Critical Transitions in Eigenvector Centrality. *Journal of Complex Networks*. https://www.researchgate.net/publication/350004417_Popularity_and_centrality_in_Spotify_networks_critical_transitions_in_eigenvector_centrality

Donker, S. (2019). Networking data. A network analysis of Spotify's socio-technical related artist network. In University of Groningen, ERC, & Vienna Music Business Research Days, *International Journal of Music Business Research* (Vol. 8, Issue 1, pp. 68–69). https://musicbusinessresearch.wordpress.com/wp-content/uploads/2019/04/volume-8-no-1-april-2019-donker_end.pdf