Spotify Music Recommendations with Neo4j

Ujjawal Dwivedi, Sean Franco, Chaya Chandana Doddaiggaluru Appajigowda

Data Science Department, George Washington University
2121 I St NW, Washington, DC 20052, USA

***Abstract*— This study explores Spotify algorithms using machine learning and Neo4j. Spotify track attributes for artists, their albums, and tracks were graphed as nodes and a DFS was performed with a tree based machine learning model to explore alternative music recommendations algorithms in Neo4j.**

***Keywords*— Spotify, Algorithm, Neo4j, tree model, DFS**

I. Introduction

Spotify is a Swedish music and podcast streaming service with 602 monthly listeners (Wikipedia, 2024). It offers both free and paid subscription models for its services as the core of its business model. Currently Spotify uses a proprietary music and podcast recommendation algorithm that combines content based suggestions with historical user listening preferences.

Dmitry Pastukhov, a data scientist, wrote an article on the website Music-Tomorrow where he references that Spotify's recommendation algorithm has aims for "retention, time spent on the platform, and, ultimately, generated revenue" (Pastukhov, 2022). Machine learning techniques are used to disseminate selections based on metadata of each track which ranges from artist and label information, to geography of where the artist is based, to descriptors and classification categories for the music. The latter refers to typical categories like genre, but it also includes other descriptors related to mood. In tandem with track metadata, Spotifies algorithm also includes track content or sonic attributes of the track. Many of these groupings relate to the combination of dynamics, loudness, vocals (or not), and predictors for 'happy/upbeat' versus 'sad/angry' (Pastukhov, 2022).

Even though the Spotify algorithm is based upon statistical properties leveraging both the user history attributes and the other collection of track attributes with complex tree-based models, the daily perception of the algorithm takes different, cultural forms rooted in community. Siles et al. noted from research studies in Costa Rica on the public perception of Spotify's algorithm (Siles et al. 2020). They found that people generalized the algorithm in two main ways, with subset individualistic variations. One perspective views the algorithm as an omniscient, community member that has near perfect knowledge of one's musical tastes and preferences. Often akin to surveillance or the parent - child relationship. Not to dismiss this perception, but this view illustrates a lack of computer science literacy. The other perspective is viewing the algorithm as a curated library that leverages data choices for the user with supervised feedback. This view reflects a higher degree of computer literacy than the former perspective, but individuals may not specifically identify key mechanisms in the algorithm. In short, they are on the right track, and both theories offer community-oriented perspectives to utilize the algorithm without needing to correctly know the mechanisms. This paper is important because Siles et al. found that these perceptions are likely to be widespread in the global south. The computer literacy related perception also informs modifications for the algorithm to retain users.

Another issue is that users tend to outgrow the algorithm if they have a diversified listening history. In another sense, this diversified listening preferences and selections suggest that the user's brain is the better algorithm. However the opposite is true for users with a less diversified listening portfolio as found by (Anderson et al. 2020). These users benefit from suggestions from the algorithm, but over time they may find that the algorithm becomes stagnant after they interact and consume a wider portfolio of listening selections. So, our project attempts to create an algorithm based on track attributes to bridge the gaps for undiversified music listeners, who may also have low computer literacy.

To address these computer literacy differences, and to maximize retention on the app, we are interested in using machine learning to create an 'offbeat' music recommendation algorithm. This algorithm will utilize unusual suggestions for Spotify music listeners for both diversified listening tastes, and non-diversified listening tastes. The thought process is to expose the Spotify music enjoyer to music tastes that might be different than what is traditionally recommended. These unusual recommendations offer a different experience of recommendation which some users would enjoy as they explore different musical expressions.

II. Data and Methods

Data was collected from a Kaggle competition which used Spotify's API about track content ranging from genre, energy, valence etc. See Table 1 for the summary statistics of the track content variables. Data about users and customers will be generated using the Faker library in Python, but at a later step to be implemented in the algorithm. So the data will need cleaning like all datasets, but most of the key variables will be available which provides many variables to manipulate in our recommendation system. We identified one missing data point in our dataset of 113,999 observations and 20 parameters. See Table 1 for our summary statistics of each variable.

Neo4j is a graphical database, so we are grouping various relationships of our data into the database. One relationship pairing is artists and their albums. This relationship highlights retention of artists as a career choice in the database. The thought process being that artists who release more albums gain additional popularity and thus exposure in the algorithm based on user historical preferences. The longer their career, the likely more albums they will produce and gain more popularity in the algorithm. So we can expect these nodes to be easily identified by our machine learning tree characteristics. The second relationship is between artists, their albums, and tracks. This highlights royalties and sampling of tracks in other albums either in partial like sampling or in full such as in compilation albums. Both of these relationships increase exposure to longterm musical careers.

After performing Ordinary Least Squares regressions on the track attribute data, it was found that selected variables were significantly contributing to the dependent variable of popularity. Duration_ms, Loudness, Tempo, Danceability variables were all significant at the 0.1% confidence interval influential on track popularity. What this means is that louder, faster, more danceable tracks that are not too long in time lead to higher popularity. Time Signature and Mode of track attributes are significant at the 1% and 5% confidence intervals respectively. Time signature shows that certain time signatures like 4/4 and 6/8 are more popular than other time signatures such as ¾ which is common for waltzes or slow songs. Mode refers to the musical organization as major or minor, where major tracks are more popular than minor tracks.

We are utilizing python as the interpreter for Neo4j, where we will make graphs and machine learning models of our data. Before we model our algorithm, some summary graphs of our dataset are presented.

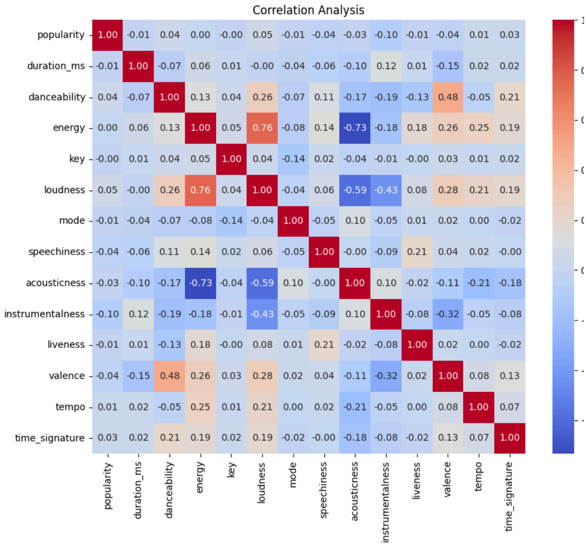

Fig. 1 Correlation Matrix of float variables.

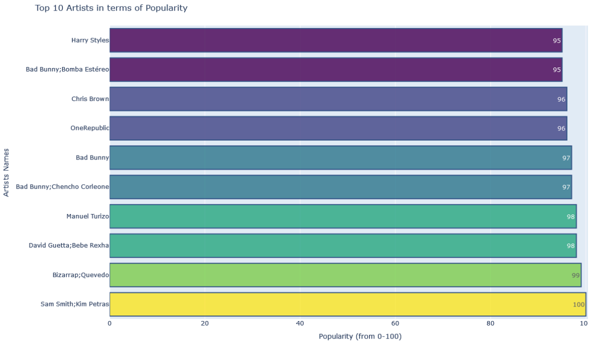

Fig. 2 Top ten artists by streaming popularity.

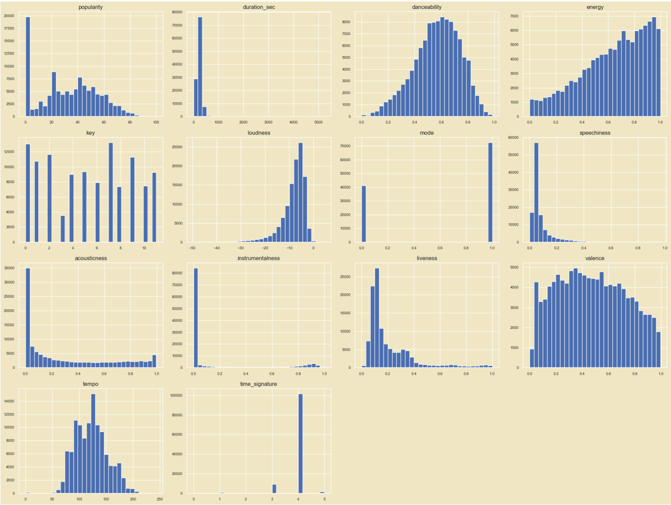

Fig. 3 Numeric and binary variable distributions.

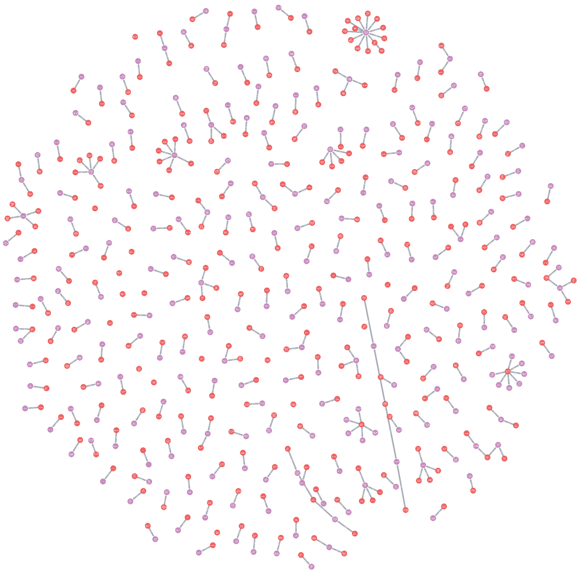

Fig. 4 300 nodes of artist (in purple) and album (in red) nodes.

In Figure 4, 300 artists nodes and their albums are graphed. The main take-away from this figure is that there is a mix of entry-level artists into Spotify, which is interpreted from artists having one node to an album. Artists with multiple nodes, or in this case albums showcase maturity in their career as a musical artist. Additionally, there are more entry-level artists than matured artists.

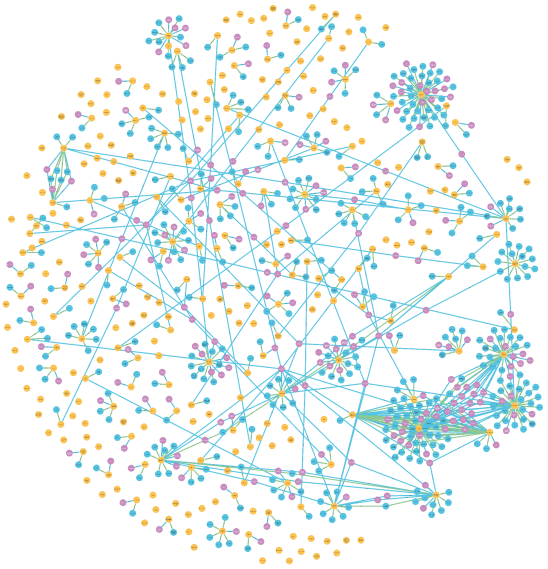

Fig. 5 300 nodes of artists (in yellow), albums (in blue), and tracks (in purple).

This graph showcases artists and their album tracks. The interconnected tracks to other albums showcase compilation albums and other featuring artists connected through the nodes. This showcases on a more drastic example the difference of artists involved in the music industry than entry level artists.

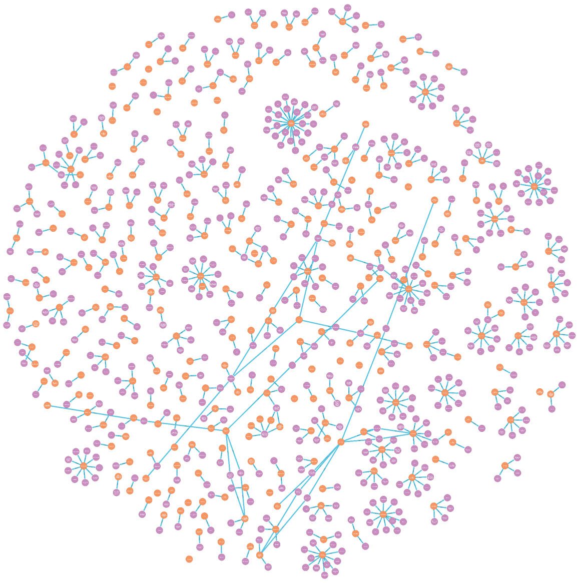

Fig. 6 300 nodes of artist to artist (in yellow)track (in purple) collaborations.

This graph showcases artists and track collaborations where multiple artists maye contribute to a track as a featured artist, or duo collaboration. Often these collaborations signified matured artists in their career.

III. Results

NGS, or the default machine learning library in Neo4j had some trouble, so the SciKit-Learn library was employed to troubleshoot the tree model. A tree model was performed to predict artist names based on album and track attributes. The model had an accuracy of 72%, a precision of 73%, a recall of 72% and an F1-Score of 72%. These scoring metrics incur a loss around 25% so true positives are balanced compared to false positives and false negatives. So the model is performing robustly across this dataset. Implementing a graphical structure as specified in Neo4j improves how the tree model can penetrate the nodes. Thus all four classification metrics incur reasonable scores for evaluating artists and recommending other artists.

IV. Conclusions

Spotify's recommendation algorithm plays a crucial role in user experience and platform retention. It uses data from user listening histories to suggest content as well as analysis of track attributes to suggest content. Generally, users with a diversified portfolio of listening content do not rely on the algorithm for recommendations. This is due to users already having a method or way to find diversified tracks (Anderson et al. 2020). However, users that have less diversified listening portfolios directly benefit from algorithms to suggest new content. Both outcomes result in short-term increased engagement on the platform, which increases the chances a user would upgrade their subscription to the premium service. Algorithms are especially important due to the integration of their perception into society regardless of an individual's computer literacy (Siles, 2020). This means that companies need different algorithms to constantly gain new audiences.

Our team leveraged a graphical database to construct a machine learning recommendation algorithm based on track attributes. In addition to OLS exploration of variables to create weights for our machine learning model, exploratory graphs were created in Neo4j to showcase artists that have established a long term career in the music industry. This relationship is visually showcased through the relationship between artists and their album releases, artists and their album tracks in other albums and collaborations. The more nodes for each artist to their albums are a proxy for long term careers in the music industry. Artists with short term musical careers are more likely to be represented with lesser nodes for albums, tracks, and collaborations.

Our machine learning algorithm uses the regression trees in a depth first search based on artists-album-artist popularity for recommendations. Overall, our tree based model is robust to false negatives and positives, while correctly identifying artists based on other track information. This is to lump similar artists together based on their style of tracks content based on sonic characteristics. created a machine learning algorithm based on track attribute data. The model examines half of the total algorithm by examining track attributes, but the user perspective is also important. We generated fake user data with weights based on track popularity and its associated artists. This assumption essentially distinguishes users with low to middle musical portfolio diversity. In theory, these users are the targets that would most benefit from an enhanced algorithm. However, non fabricated data is needed to properly graph and implement our algorithm.

References

[1] Anderson, Ashton, et al. "Algorithmic Effects on the Diversity of Consumption on Spotify." *Proceedings of The Web Conference 2020*, Association for Computing Machinery, 2020, pp. 2155–2165.

[2] Pastukhov, Dmitry. “How Spotify’s Algorithm Works? A Complete Guide to Spotify Recommendation System [2022]: Music Tomorrow Blog.” *How Spotify’s Algorithm Works? A Complete Guide to Spotify Recommendation System [2022] | Music Tomorrow Blog*, Music Tomorrow, 9 Feb. 2022, www.music-tomorrow.com/blog/how-spotify-recommendation-system-works-a-complete-guide-2022

[3] Siles, I., Segura-Castillo, A., Solís, R., & Sancho, M. (2020). Folk theories of algorithmic recommendations on Spotify: Enacting data assemblages in the global South. Big Data & Society, 7(1). https://doi.org/10.1177/2053951720923377

[4] “Spotify.” *Wikipedia*, Wikimedia Foundation, 5 Apr. 2024, en.wikipedia.org/wiki/spotify.

Appendix

Table 1. Summary Statistics

	popularity	duration_ms	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	time_signature
mean	33	228,031	1	1	5	-8	1	0	0	0	0	0	122	4
std	22	107,296	0	0	4	5	0	0	0	0	0	0	30	0
min	0	8,586	0	0	0	-50	0	0	0	0	0	0	0	0
25%	17	174,066	0	0	2	-10	0	0	0	0	0	0	99	4
50%	35	212,906	1	1	5	-7	1	0	0	0	0	0	122	4
75%	50	261,506	1	1	8	-5	1	0	1	0	0	1	140	4
max	100	5,237,295	1	1	11	5	1	1	1	1	1	1	243	5

```
OLS Regression Results
==============================================================================
Dep. Variable:           popularity   R-squared:                       0.017
Model:                          OLS   Adj. R-squared:                  0.017
Method:               Least Squares   F-statistic:                     74.01
Date:              Sun, 14 Apr 2024   Prob (F-statistic):           6.16e-92
Time:                      12:38:51   Log-Likelihood:            -1.1058e+05
No. Observations:             25150   AIC:                         2.212e+05
Df Residuals:                 25143   BIC:                         2.212e+05
Df Model:                         6
Covariance Type:          nonrobust
==============================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
const            50.0192      0.662     75.519      0.000      48.721      51.317
duration_ms    -1.951e-05  1.22e-06    -15.939      0.000   -2.19e-05   -1.71e-05
danceability    5.757e-13  2.05e-14     28.124      0.000    5.36e-13    6.16e-13
energy           -6.5498      5.451     -1.202      0.230     -17.234       4.134
loudness          0.3157      0.024     13.320      0.000       0.269       0.362
speechiness    -1.282e-15  1.98e-15     -0.647      0.517   -5.17e-15     2.6e-15
acousticness   -1.187e-15  4.36e-15     -0.272      0.786   -9.74e-15    7.36e-15
instrumentalness  17.3178     8.829      1.961      0.050       0.012      34.623
liveness          5.3456     19.649      0.272      0.786     -33.169      43.860
tempo            -0.0275      0.004     -6.435      0.000      -0.036      -0.019
==============================================================================
```

```
OLS Regression Results
==============================================================================
Dep. Variable:           popularity   R-squared:                       0.337
Model:                          OLS   Adj. R-squared:                  0.334
Method:               Least Squares   F-statistic:                     99.58
Date:              Sun, 14 Apr 2024   Prob (F-statistic):               0.00
Time:                      12:38:55   Log-Likelihood:            -1.0562e+05
No. Observations:             25150   AIC:                         2.115e+05
Df Residuals:                 25021   BIC:                         2.126e+05
Df Model:                       128
Covariance Type:          nonrobust
=========================================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
```

```
-----------------------------------------------------------------------------------------
const                        44.2445      1.484     29.818      0.000      41.336      47.153
danceability                  3.65e-13   3.58e-14     10.205      0.000    2.95e-13    4.35e-13
energy                       -2.9338      4.521     -0.649      0.516     -11.796       5.928
mode                         -0.3369      0.223     -1.514      0.130      -0.773       0.099
time_signature                0.7148      0.242      2.954      0.003       0.241       1.189
track_genre_afrobeat        -19.6808      1.465    -13.434      0.000     -22.552     -16.809
track_genre_alt-rock         -0.6055      1.802     -0.336      0.737      -4.138       2.927
track_genre_alternative       5.3435      2.073      2.578      0.010       1.281       9.406
track_genre_ambient           0.6578      1.544      0.426      0.670      -2.369       3.685
track_genre_anime             1.1554      1.573      0.734      0.463      -1.929       4.239
track_genre_black-metal     -22.0627      1.571    -14.042      0.000     -25.142     -18.983
```