Spotify Tracks Popularity Analysis

Jongwan Kim (jk2369), Richard Lin (rl556), Lingfei Zhao (lz382)

Course code: ORIE 5741
Github link: https://github.com/Richardlin112/Spotify-Tracks
Video Presentation link: https://drive.google.com/file/d/1XKcsBDIZMzbCfdQCVqahI3QHcwQZTlOL/view?usp=drive_link

Contents

1. Introduction

1.1 Background and Motivation

Music has always been a universal language that surpassed cultural boundaries, playing a vital role in human expression and connection. It has constantly evolved over time, encompassing genres from jazz and country music, to hip-hop and electronic dance music. The evolution of popular music genres over time underscores the dynamic nature of audience preferences and global trends. Identifying features that contribute to a song's popularity, on platforms like Spotify, will allow businesses involved in the creation, promotion, or distribution of music to strategically allocate their resources and effectively cater to their target audiences.

1.2 Problem Statement

Our paper answers the following 3 questions:

1. What features make a song more popular on Spotify
    (a) In general
    (b) Within its own genre?
2. How well can we predict a song's popularity based on various features of the song?
3. How can we build a system to recommend songs to Spotify users?

1.3 Dataset

We explored a dataset of almost 90,000 tracks on Spotify spanning across 125 genres, which was retrieved from Kaggle, from the user Maharshipandya. Each track is associated with a total of 19 features such as artist, tempo, danceability, and popularity score. These features consist of 8 nominal variables, 1 ordinal variable, 1 discrete variable, and 9 continuous variables.

For each track in the dataset, there is information on its various song attributes and the respective popularity score ranging from 0 to 100, which was calculated by an outside algorithm based on the total number of plays the track has had and how recent those plays were. This allowed us to fit several regression models to gauge the impact and significance of each feature on the popularity of a track. Due to the difficulty of converting nominal variables such as artist, album, and genre to numerical values for regression, we also used other feature selection methods for the categorical data.

To answer the second part of the first question, we conducted cluster analysis based on genre and determined if there are trends in attributes that would classify a song into a specific genre. We also segmented the data by genre and built a regression model on each to determine if the features that make a song popular are dependent on genre.

We then built an XGBoosted Decision Tree model to predict the popularity of new songs based on the features we determined to be most important. This allowed us to classify whether or not future songs are likely to be successful.

Finally, to answer the last question, we build a model using KNN to output the top 5 most similar tracks to a user inputted track.

The large amount of entries present in this dataset (just under 90,000 unique tracks), as well as the well-documented and diverse range of features, allow us to present robust predictive models with clear and easily comprehensible results.

Below we summarize several key attributes of the dataset as well as their initial types before any data processing.

Table 1: Description of several key features in the original dataset
***popularity** is our target prediction variable

Feature	**Type**	**Description**
track_id	String	The Spotify ID for the track (Unique identifier)
popularity*	Integer	Score (between 0 and 100, with 100 being the most popular) measuring the track's popularity
duration_ms	Integer	Track length (in milliseconds)
explicit	Boolean	Whether track contains explicit lyrics (If unknown, this is also set to 0)
danceability	Float	How suitable a track is (from 0.0 to 1.0) for dancing based on elements including tempo, rhythm stability, beat strength, and overall regularity.
key	Integer	The key the track is in, mapped to pitch using standard Pitch Class notation (0 = C, 1 = C$\sharp$/D$\flat$, 2 = D, etc. If no key was detected, the value is -1)
loudness	Float	Average loudness of a track in decibels (dB)
acousticness	Float	A confidence measure from 0.0 to 1.0 of whether the track is acoustic
speechiness	Float	Measure from 0.0 to 1.0 indicating the presence of spoken words. Values below 0.33 most likely represent music, values between 0.33 and 0.66 indicate both music and speech (e.g., rap), and values above 0.66 consist of almost entirely spoken words.
track_genre	String	Genre of the track

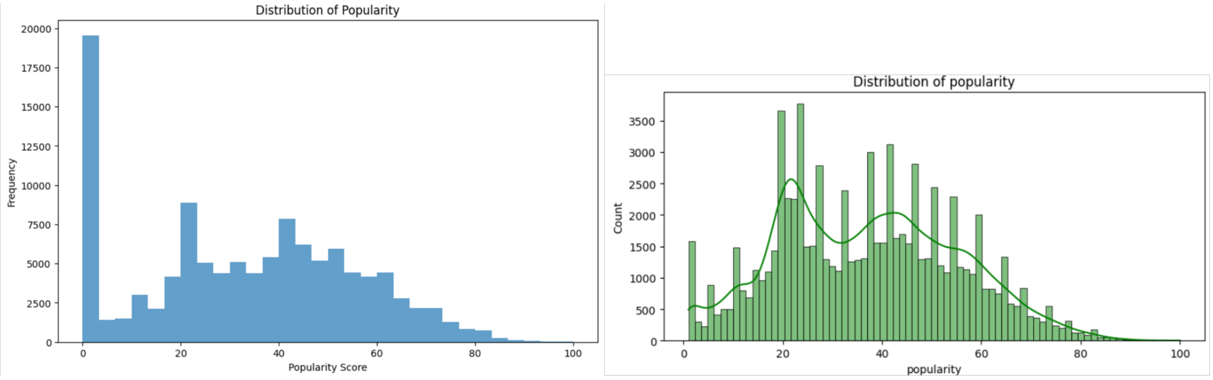
2. Data Preprocessing

2.1 Missing and Duplicate Values

The original dataset had 114,000 entries with 20 features (including popularity). We first removed all null values (only one entry), and then dropped duplicate rows. This resulted in a dataset with 89,740 entries. Imputation for missing values was not considered as there was only one row with missing values, and therefore deleting this had minimal effect on the quality of such a large and comprehensive dataset.

2.2 Feature Engineering

We also converted boolean values to binary 0/1, and transformed the duration of each track from milliseconds to minutes for ease of use in our models. We then changed specific track genres into broader music categories. For example, 'house', 'techno', and 'dubstep' would all be grouped under the 'Electronic Dance Music' category. This process allowed us to transform 125 smaller genres into 8 major categories (Electronic Dance Music (EDM), Rock, Hip-Hop and R&B, Pop, Latin and Dance, Funk and Disco, Jazz and Blues, and Other) which would make the data less noisy and susceptible to overfitting. Thus we can still add the genre feature to our regression without adding too many feature dimensions. See Figure 3 for a distribution of these 8 music categories. For all categorical features, we used a label encoder to encode feature values to be used as input in our models.



(a) Before Data Preprocessing

(b) After Data Preprocessing

Figure 1: Popularity Distribution Before and After Data Preprocessing

2.3 Further Pruning

Because we wanted to focus on the musical features of each song rather than their fixed categorical attributes, the following non-numerical columns were dropped: 'track_id', 'artists', 'album_name', 'track_name'. We also decided to remove all entries with a popularity of 0, as a surprisingly large percentage (10.5%) of our data had a score of 0, which heavily skewed the entire dataset. Because the popularity score is both a metric of the track's popularity (in terms of plays) and the track's age (based on how recent the plays were), a popularity score of 0 likely indicates that the song is much older, which we deemed to be less relevant to our question of what makes a song popular *now*. Removing these entries left the remaining data in a more normal distribution representative of the real world (See Figure 1). Our final dataset thus had 80,293 entries and 16 features.

3. Exploratory Data Analysis

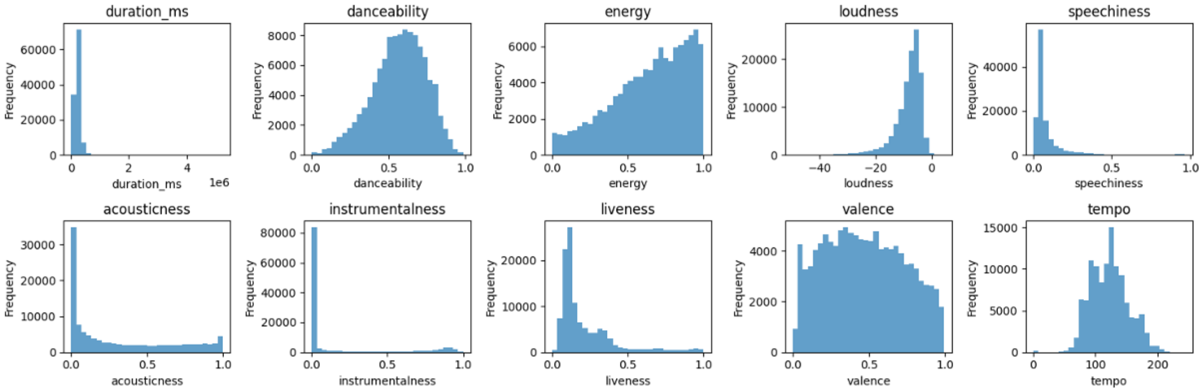


Figure 2: Distributions of Numerical Features

We conducted several exploratory data analyses before the modeling process in order to have a better idea of the feature distributions and take note of any patterns that may be important.

Firstly, Figure 3 shows a distribution of the 8 music category buckets. With the exception of the 'Other' category, Latin & Dance and Rock encompass the highest percentage of datapoints (around 12.6% and 12.3%, respectively), while the lowest categories Funk & Disco and Blues & Jazz make up barely over 2% of the data. This disproportionate representation suggests that in addition to using the full dataset, we can also analyze each music category separately.

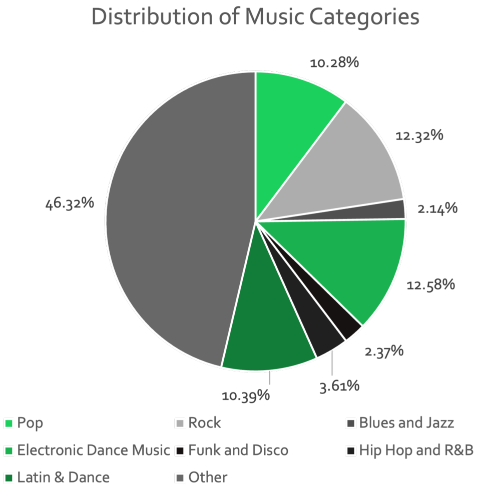


Figure 3: Distribution of Music Categories

Figure 2 depicts the distributions of all numerical features. We can see that some features (valence, danceability) are more evenly spread while many others (duration, acousticness, speechiness, etc) are heavily skewed.

4. Methods and Results

4.1 Regression

We began our analysis by fitting seven regression models to our data and comparing their results to determine which would best predict the impact of song features on popularity score. Table 2 below lists the models and their performance.

Table 2: Regression Model Performance Metric Comparison

Regression Model	Mean Squared Error	R-Squared
Linear	306.2943	0.0818
Ridge	306.2943	0.0818
Lasso	309.4885	0.0723
Decision Tree Regressor	485.6066	-0.4557
XGBoost Regressor	257.0330	0.2295
Polynomial (2 degrees)	284.7026	0.1466
Polynomial (3 degrees)	277.2290	0.1690

Overall, the Extreme Gradient Boosted (XGB, or XGBoost) Decision Tree Regression model resulted in the lowest mean squared error and the highest R-Squared value, indicating a better fit compared to the other models. From Figures 4 and 5, we can see that the model tends to underpredict songs with high actual popularity scores and overpredict songs with low actual popularity scores; nonetheless, due to its higher performance than the other models we focused on this XGBoost regressor to calculate the importance of each feature (1) across all songs in the dataset and (2) by music category.

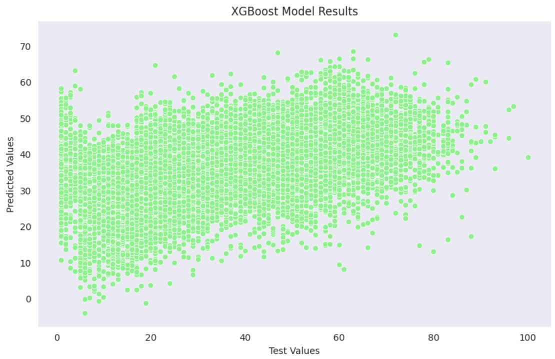


Figure 4: XGB Actual vs. Predicted Popularity

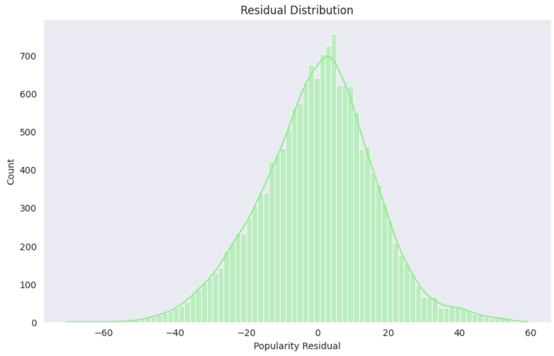


Figure 5: XGB Residual Distribution

4.2 Decision Trees

Focusing on the XGBoost model, we examined which features contribute the most to song popularities by incorporating decision trees to detrmine the relative imporance of the features. More specifically, a decision tree regressor was employed on the entire dataset and we plotted the feature weights to visually depict not only what features contribute the most to popularity but also how important those features are.

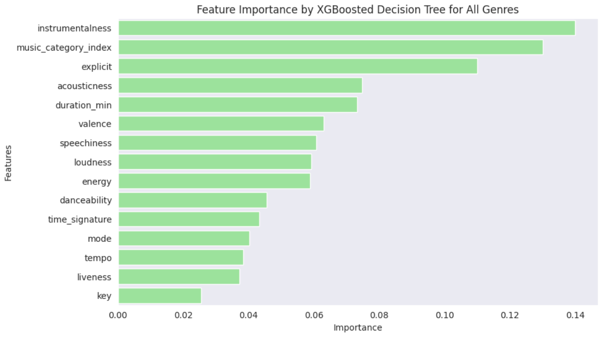


Figure 6: Feature Importance by XGBoosted Decision Tree for All Genres

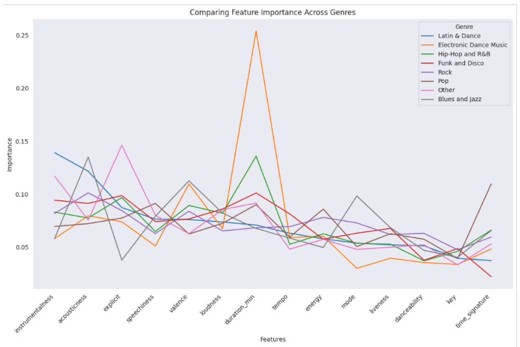


Figure 7: Comparing Feature Importance Across All Music Categories

From the results above, we can see that while the overall most important feature was 'instrumentalness' (Figure 6), we can see that 'music_category_index' (the music category we grouped each genre into) had the second most importance. Thus, it is clear that the song genre matters significantly in determining popularity. We therefore dove deeper to examine the feature importance for each music category by repeating this process of constructing a decision tree regressor and graphing feature weights.

Figure 7 displays a consolidated view of feature importance variation across all 8 music categories. We can see that different categories have a preference for different features. For example, track duration has the largest effect on popularity for EDM, while Pop was most impacted by time signature. Therefore, when choosing attributes to focus on during the music production process and predicting the success of a track afterwards, it is necessary to keep the genre in mind.

4.3 KNN Clustering & Principal Component Analysis

The previous sections provide insight to be used before a track's release; this is useful for Spotify artists in the production process, as well as stakeholders when determining resource allocation between artists/songs. However, in this section we will now shift our focus to developing a model targeted towards Spotify users.

We employed K-Nearest Neighbors (KNN) clustering (based on Euclidean distance and using 6 neighbors) to devise a song recommendation system based on user input (a single track that they enjoyed). This recommendation system disregards song popularities and only considers actual musical features such as music category, liveness, loudness, etc. There is also a filter it passes in order to isolate the songs by the same artist or genre as the selected track. By doing so, we are able to make song recommendations that most closely align with the user's input regardless of how popular these songs are. The KNN is configured to output the 5 most similar songs to any song input, and is visualized via principal component analysis (PCA). Since KNN clustering works in multi-dimensional space in this case, PCA with 2 components was employed to reduce the dimensionality in order to produce a user-friendly visualization (See Figure 8 to the right).

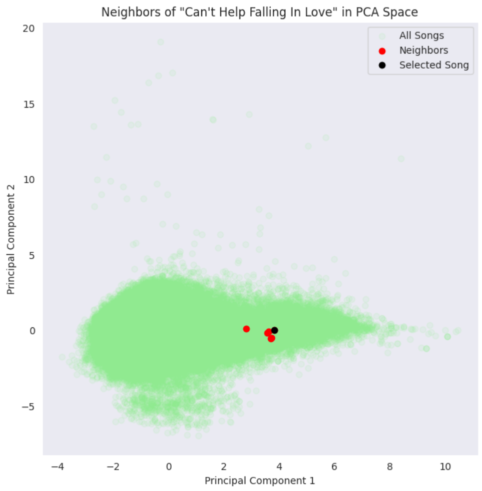


Figure 8: Visualization of the user's input and its nearest neighbors

Finally, we generated a heatmap of the pairwise Euclidean distances between the first 50 songs in the dataset (Figure 10), which helps to visualize which songs are the most similar and the most different. For example, songs #17 and #25 have the least similarities. This framework can be modified to provide users a short list of new songs and the similarities to their current preferred songs.

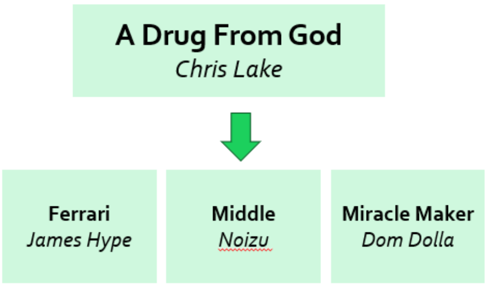


Figure 9: Reccomended Songs by KNN for the track "A Drug From God"

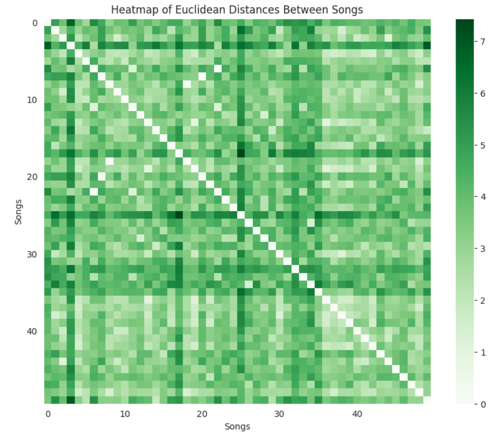


Figure 10: Heatmap of Pairwise Euclidean Distances Between the First 50 Songs in the Dataset

5. Conclusion

This project has provided multiple insights on which musical features have the largest impact on Spotify track popularity, both with and without respect to genre. Our results show that feature importance by music categories do correlate to the characteristics of that music. Moreover, we implemented an XGBoosted Decision Tree Regression model to gauge the relative popularity of a track based on the features concluded to be important. Finally, we designed a song recommendation system using KNN clustering that works as follows: The user will input a song that they enjoy, and the algorithm will output 5 of the most similar songs to their input, regardless of popularity. This process may introduce new songs that are less popular and thus provide smaller artists a chance to gain exposure.

5.1 Applications

This project provides great utility and valuable insights not only to the music producers but also to Spotify. For the music producers, this project will allow them to get an insight into the components of a song that are the most critical in gaining popularity. With this insight, the producers will have a better idea of what aspects of a song to focus on for an increased chance of "success". For Spotify, this project can add more value to the existing recommendation system. The current Spotify recommendation system shows any and all songs that roughly align with the user's preferences, demographics, and current trends (Understanding Recommendations on Spotify). With this project, however, this recommendation system can be refined to have an extra layer of filtering which prioritizes songs that both align with the users' preferences and have the most likelihood of being popular to the users. This improvement will lead to a higher quality recommendation system, which will help enhance the users' trust in Spotify's recommendations. Due to the great benefits and insights that this project brings as mentioned above, we are confident that this development will prove to be meaningful and worthwhile.

5.2 Next Steps

Several steps can be taken to further improve our predictions. Regarding the data preparation stage, various data pre-processing/feature selection methods will produce different results; thus it is important to understand underlying patterns in the data beforehand. For example, there is valuable information in the 'artist', 'track_name', and 'album_name' features that we dropped, as an artist's reputation has a large effect on the popularity of their songs. In other words, the same song published by different artists will not perform similarly. Therefore, rather than dropping these features, we can use natural language processing methods to extract important information from them and incorporate this into our model. Secondly, future iterations can explore the reason behind such a high proportion of tracks having a popularity score of 0, and whether it would be more accurate to keep these in our dataset. Lastly, because the dataset we used for this project disregards time, it would be helpful to also incorporate data specifying the release date of each track. this would allow us to analyze trends over time and note patterns that do or do not carry over across the years. Moreover, removing songs that we deemed too old would also allow us to keep the tracks with a 0 popularity score without risk of irrelevancy in terms of age.

5.3 Weapon of Math Destruction

Our project answers three major questions: (1) Which features make a song more popular on Spotify in general and within its own genre, (2) How can we predict a song's popularity before its release, and (3) How should we build a recommendation system based on song similarity? The first question can be used to aid the music production process; artists can use this insight to focus on specific characteristics when producing music. In this aspect, fairness is still maintained. The third question only focuses on objective musical attributes of songs and thus fairness is not an issue.

However, the second question may be used by record labels and stakeholders who decide which artists and which songs to release and/or promote. Thus, songs with low predicted popularity are less likely to be released, giving our model the potential to become a *weapon of math destruction (WMD)*. WMDs are predictive models with the following characteristics (Udell):

1. The outcome(s) is not easily measurable
2. Predictions can have negative consequences
3. It may create self-fulfilling (or defeating) feedback loops

While the first attribute does not apply to our model (The outcome is not hard to measure, although can be subjective based on how popularity is calculated), the second and third attributes are relevant here. As described above, songs with low predicted popularity may be disregarded by record labels and thus will result in low actual popularity, while songs with high predicted popularity may be more actively promoted and thus result in high actual popularity. As a result, the high popularity characteristics will be further reinforced and the low popularity characteristics will be further disregarded, essentially creating a feedback loop. Furthermore, smaller or newer demographics may have features that lean towards "less popular" and therefore receive a low predicted popularity score simply because these songs are not as well represented in the dataset. This may cause artists from these demographics to be overlooked as record labels choose not to release their songs. While our song recommendation system ignores popularity when recommending songs, which can help provide a small counter to this situation, it is crucial to ensure in future iterations that all demographics are well represented in the dataset. The feature selection process must also acknowledge the existence of demographic biases in order to maintain fairness.

6. Team Contributions

All members contributed to every aspect of this project, with the primary objective of each memeber listed below.

- Jongwan Kim: KNN and PCA Analysis to build a recommendation algorithm.
- Richard Lin: Random Forest Regression to determine feature importance.
- Lingfei Zhao: Data pre-processing and regression analysis predicting popularity of songs.

7. References

- Dataset: https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset
- Udell - Weapon of Math Destruction:
  https://people.orie.cornell.edu/mru8/orie4741/lectures/limits.pdf
- Understanding Recommendations on Spotify:
  https://www.spotify.com/us/safetyandprivacy/understanding-recommendations