# Predictive Modeling for Song Popularity on Spotify

## Project Final Report

Name: Ying Yang (leader), Yixin Qu, Zhiqian Li, Wenjing Huang

## Problem Definition

Our project is to build predictive models to predict the popularity of future songs on Spotify. More specifically, we have two objectives. The first objective involves applying prediction models to predict the popularity score of future songs based on selected music features. The second objective is to categorize songs as 'popular' and 'less popular' based on their track popularity scores with the threshold of 67 and develop classification models to predict if a given song will become a popular song or not based on its music features. By integrating these predictive models, our project aims to provide artists and producers with valuable predictive tools to make more informed decisions about music production and marketing strategies.

## Background Description

### *Why is This Meaningful?*

The program aims to empower industry stakeholders and artists with actionable insights. The global recorded music market is profitable, with revenues set to rise to $28.6 billion and the global recorded music increased by 10.2% in 2023 (International Federation of the Phonographic Industry, 2024). However, the music industry is also facing challenges such as market saturation and algorithm-driven content discovery, making it increasingly competitive for artists to achieve visibility and success (Kiberg & Spilker, 2023). Our forecasting model synchronizes creative efforts with anticipated 2024 trends, equipping artists with tools to navigate the dynamic music landscape for optimal music success and commercial returns.

### *Contribution*

Understanding the factors that influence the popularity of a song (on mainstream music platforms such as Spotify) is significant for both the music industry and data science research on music popularity. Our study is dedicated to designing music popularity prediction models, which can help music platforms improve their content curation strategies, provide users with a better experience, and assist artists in achieving music success.

### *Related Work*

Our study is an extension of research on music recommendation systems and predictive modeling conducted by McFee et al. (2012), who tested different machine learning methods such as decision trees and neural networks on the million song dataset challenge, and by Oord et al. (2013) on sophisticated music selection algorithms based on content, which provided the field of music feature retrieval with useful achievements about different algorithms. However, despite the advances in the subject, many of the above-mentioned studies focus on the music chosen by music listeners in general, rather than on how extremely successful songs are explicitly chosen. Our study thus addresses this issue, by exclusively looking at extremely popular songs, and modeling the music characteristics of these songs in order to provide new and useful insights for both the academic and industrial worlds.

## Description of Dataset

Due to the limited number of songs (only 1,000 top songs) in our last database reported in the progress report, it has been challenging to build a more specific analysis and modeling. So, we have switched to the "30,000 Spotify Songs" database, which offers a richer data set.

This dataset, titled "30,000 Spotify Songs", originates from the Spotify platform and covers the time range from 1960 to 2020. It was compiled through web scraping using the Spotify API. Additionally, the 'spotifyr package' package has been utilized as an efficient tool to easily pull the tracks' audio features and songs' information by automatically batching Spotify API requests.

This dataset exhibits both high quality and quantity. There are two reasons why this dataset is of high quality: firstly, because it is sourced directly from Spotify, a leading streaming platform, which makes it an official and reliable resource (Araujo et al., 2020). Secondly, it contains a large amount of information with 23 attributes across over 30,000 rows, enhancing the precision of our predictive models due to the breadth of our dataset.

Since our focus was on musical features, and although the dataset includes 23 attributes, we opted to exclude 10 non-musical features such as 'track_id,' 'track_name,' 'track_artist,' and 'track_album_id.' Ultimately, we used one main attribute, track popularity (scores range from 0-100), along with 12 musical attributes including danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, valence, tempo, duration_ms, key, and mode for our analysis. Additionally, we applied feature selection to further analyze which music features are significantly important; using forward selection and SelectKBest, we finally selected the following 7 music features based on their importance scores which are larger than 50. To provide a clearer understanding of these elements, we also provide a detailed explanation of the 8 main attributes in Table 1.

| **Final selected attributes** | **Definition** |
|---|---|
| Track_popularity (0-100) | Higher is better with more popularity. |
| Instrumentalness (0.0-1.0) | Predicts whether a track contains no vocals. |
| Energy (0.0-1.0) | A perceptual measure of intensity and activity. |
| Loudness (-60-0) | Overall loudness of a track in decibels (dB). |
| Acousticness (0.0-1.0): | A confidence measure, 1.0 represents high confidence the track is acoustic. |
| Liveness (0.0-1.0) | Higher liveness values indicate the track was performed with more live. |
| Danceability (0.0-1.0) | If the song was suitable for dancing. |
| Duration_ms | Duration of song in milliseconds. |

Table 1. The definition of partial attributes.

**Description of Methods Used**

Here are several procedures and methods we used in our experiment. Firstly, we began with data preprocessing using Pandas, which includes data cleaning and standardization to ensure accuracy and consistency. We used the standardization method because it normalizes the data and adjusts for differences in scale among variables, allowing for equitable comparisons and accurate modeling. Next, we conducted exploratory data analysis (EDA) utilizing visualization techniques such as Matplotlib to examine the dataset's distribution and provide an overall picture. Then, we performed feature selection using forward selection and SelectKBest to help eliminate redundant or irrelevant features and improve model performance.

Finally, we employed algorithms to contrast our predictive models. We used both prediction and classification methods with the objective of forecasting song popularity using two different approaches. Firstly, we employed prediction models such as Linear Regression, Random Forest Regression, and Decision Tree Regression to forecast the popularity score of a song by considering its musical features. Additionally, we used a range of classification algorithms, including Logistic Regression, Naive Bayes, Decision Trees, and Random Forest, to ascertain whether a song could be categorized as popular, with popularity defined as a score exceeding 67—a threshold representing the top 15% of songs based on popularity. Afterward, we evaluated the performance of these algorithms using accuracy, precision, recall, and F1 scores to determine which exhibited the highest level of performance

## Experiment: Experiment Setup and Analysis Results

### *Data Pre-processing*

For our data cleaning and preprocessing part, we first drop 10 attributes as mentioned in the Description of Dataset to focus on the primary features. After dropping the attributes, we utilized Pandas to check missing values and confirmed that the dataset contained none. We employed StandardScaler from sklearn.preprocessing with the standardization, which normalizes each feature to zero mean and unit variance, effectively minimizing the influence of outliers and adjusting for differences in scale

### *Exploratory Data Analysis (EDA)*

After the data cleaning, we move to the exploratory data analysis (EDA) part to better understand the whole database and the music patterns of Spotify's songs. We employed Pandas for data manipulation and Matplotlib for data visualizations. For the methodology, we began by conducting univariate analysis to explore the distribution of variables. To examine continuous variables like music features, we created histograms, and for categorical variables such as key and mode, we utilized bar charts.

Here are the results of the univariate analysis of music features, we discovered that Spotify songs generally have a moderately fast tempo, with an average beats per minute (BPM) of 120. These songs also exhibit relatively high danceability and energy levels, averaging 0.65 and 0.70 respectively, and a wide range of valence with an average of 0.51. In contrast, the percentages of acousticness (average of 0.18), instrumentalness (average of 0.08), liveness (average of 0.19), and speechiness (average of 0.107) are relatively lower in their musical characteristics. In addition, we found that key C♯/D♭ and mode major are the most frequently used among the 30000 Spotify songs (Figure 1).

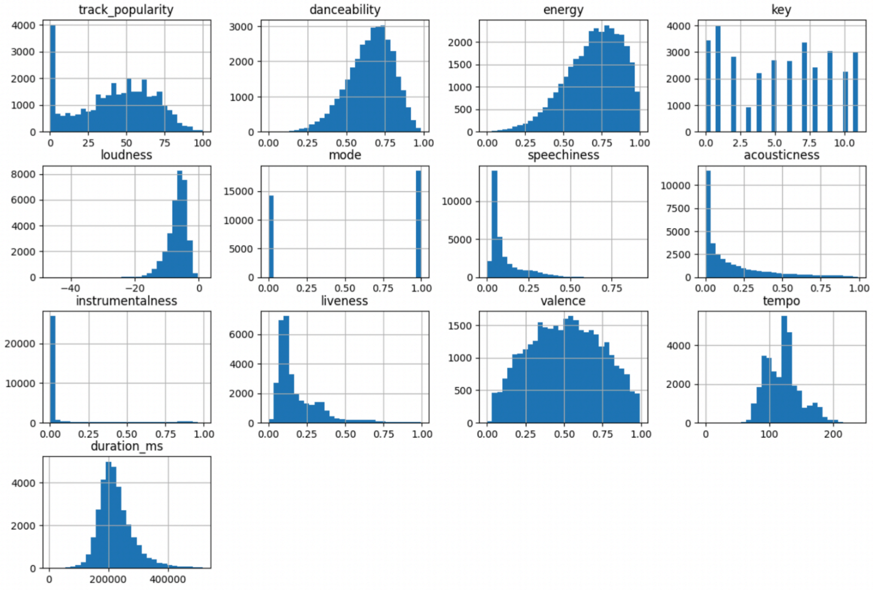

Figure 1. The univariate analysis of music features.

In addition, we moved into the bi-variate analysis and analyzed the correlation between variables. We found that energy and loudness have the highest correlation which is +0.68, while energy and acousticness have the highest negative correlation which is -0.54 (Figure 2).

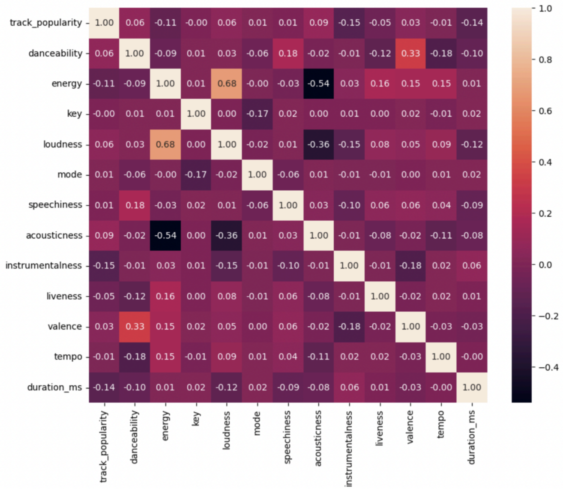

Figure 2. The correlation matrix heatmap.

### *Feature Selection*

To improve model accuracy, reduce complexity, and increase efficiency by removing extraneous data features, we employed feature selection before creating the predictive model. We began with forward selection, a method that sequentially adds features to the model based on their statistical significance. Using the SequentialFeatureSelector with a linear regression model, we identified 'energy', 'loudness', 'instrumentalness', 'valence', and 'duration_ms' as features that would have the greatest impact on the dataset. However, considering its potential limitations as a greedy algorithm—where previously selected features are permanently retained—this method raises concerns about redundancy and missing potentially better combinations from subsequently added features. Therefore, we shifted to a univariate feature selection method, SelectKBest, which selects features based solely on their overall statistical significance, avoiding the consequences of ordering. Furthermore, the resulting scores provide a better understanding of each variable's significance. As a result, we selected features such as 'instrumentalness', 'duration_ms', 'energy', 'acousticness', 'danceability', 'loudness', and 'liveness', each with a significance score above 50. This transition allowed us to create a more accurate and efficient predictive model.

### *Modeling*

### *Prediction Models*

We began with prediction models to forecast the popularity score of future songs based on selected musical features. We used three common algorithms: Linear Regression, Random Forest Regression, and Decision Tree Regression. To evaluate model performance, we utilized mean square error, root mean squared error, and the coefficient of determination ($R^2$). The results show that Random Forest Regression is the top performer among the three models. However, the overall performance of the prediction model is not ideal. More specifically, the top-performing Random Forest Regression has an R^2 of only 0.27, indicating that it explains less than 27% of the variance in song popularity, and a high mean square error of 72.12%, which indicates a significant prediction error. These results suggest that the musical features are not promising variables for precisely predicting song popularity. Therefore, we shifted our focus to our second objective, treating it simply as a categorization problem.

### *Classification Models*

Lastly, we moved to our second objective, which is to develop classification models to predict whether a given song will become popular based on its selected music features. We first created a new variable called 'popularity category' as the Y value to categorize a song's popularity based on whether standardized track popularity is greater than or equal to 1 (or track popularity score above or equal to 67), which is the threshold representing the top 15% of songs based on popularity score. We used the selected features as the X values. Then, we applied Random Forest, Logistic Regression, Naive Bayes, and Decision Tree algorithms to create the classification models and evaluated performance using accuracy, precision, recall, and F1 scores.

The results showed that Random Forest performed the best with an accuracy score of 92.5%, meaning this model correctly predicts the outcome 92.5% of the time, and an F1 score of 73.2%, suggesting it has a good balance between precision and recall. We also recorded a precision score of 91.4% and a recall score of 61.05%. These results indicate that this classification model is effective for categorizing songs as popular or less popular, thereby achieving our second objective.

**Observation and Conclusion**

To conclude, after completing our analysis and modeling, we evaluated our two objectives. We observed relatively low-quality results for our first objective, which was to predict the popularity scores of future songs based on selected music features, because the top-performing model, Random Forest Regression, only achieved an $R^2$ score of 27.4% and had a high mean square error of 72.12%. Upon reflection, we determined that the poor performance was not due to the modeling approach itself but rather because the music features data are not well-suited to predicting popularity scores. For our second objective, which was to develop classification models to determine if a given song would become popular, we achieved good results: the top-performing Random Forest classifier achieved a high accuracy score of 92.5% and an F1 score of 73.2%. Combining these results, it becomes clear that although the selected music features did not accurately predict the exact popularity values, they were effective in classifying songs into popular and less popular categories.

For our understanding of the overall project, our main goal is to provide artists with useful predictive tools to help them create popular music content and make better marketing decisions. To achieve this goal, we divided our objectives into two main areas: one is to build a prediction model to forecast the track popularity score, and the other is to build classification models to categorize given songs into popular and less popular groups based on selected features. We then selected a database of 30,000 Spotify songs and conducted data preprocessing, exploratory data analysis, feature selection, as well as prediction and classification modeling to achieve our goal. In the end, we found that the Random Forest model can be the best classification model with high accuracy and an F-score, making it suitable for predicting the popularity of future songs on Spotify.

## References

Araujo, C. V. S., Cristo, M. A. P., & Giusti, R. (2020). A model for predicting music popularity on streaming platforms. *Revista de Informatica Teorica e Aplicada*, *27*(4), 108–117. https://doi.org/10.22456/2175-2745.107021

International Federation of the Phonographic Industry. (2024). *IFPI Global Music Report 2024*. IFPI.https://globalmusicreport.ifpi.org/

Kiberg, H., & Spilker, H. (2023). One More Turn after the Algorithmic Turn? Spotify's Colonization of the Online Audio Space. *Popular Music and Society*, *46*(2), 151–171. https://doi.org/10.1080/03007766.2023.2184160

McFee, B., Bertin-Mahieux, T., Ellis, D. P. W., & Lanckriet, G. R. G. (2012). The million song dataset challenge. *Proceedings of the 21st International Conference on World Wide Web*. https://doi.org/10.1145/2187980.2188222

Oord, A.V., Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. *Neural Information Processing Systems*.https://dl.acm.org/doi/10.5555/2999792.2999907