Predicting Track Popularity Using Spotify Metrics

Final Project for DSC 190: Introduction to Data Mining

Vince Wong
University of California, San Diego
vlw003@ucsd.edu

Abstract

As of 2020, Spotify is one of the largest audio streaming platforms around the globe. The company has generated its own metrics to determine different aspects of tracks available on their platform. Using these metrics, I sought out to find what qualities of music made the pieces popular. Plotting the metrics against their popularity ratings showed that many of the metrics had a linear correlation between the two variables. Features with strong correlations include acousticness, danceability, energy, explicit, instrumentalness, loudness, tempo, and year. More information about these features can be found in a data dictionary in the Appendix section. Because of the observed linear trend, I decided to try out multiple regression models to predict popularity of songs available on Spotify. I chose a simple linear regression model for my baseline model and also experimented with decision tree regressors, random forest regressors, and gradient boosting regressors. The baseline linear regression model performed reasonably well with a root mean squared error of 10.23, and R-squared score of 0.77, and a mean absolute error of 7.74. Implementing the random forest regressor resulted in the best results with a root mean squared error of 9.02, and R-squared score of 0.82, and a mean absolute error of 6.54. More details regarding model optimization and performance metric selection will be discussed later in the report.

CCS Concepts

- **Computing** → Machine learning
- **Mathematics of Computing** → Probability and statistics; Mathematical analysis

Keywords

Spotify, linear regression, decision tree regression, gradient boosting regression, random forest regression

ACM Reference format

Vince Wong. 2020. Predicting Track Popularity Using Spotify Metrics: Final Project for DSC 190: Introduction to Data Mining. *La Jolla , CA, USA, 6 pages.*

1 Introduction

With over 130 million paid (premium) subscribers and 286 million monthly active users as of May 2020 , Spotify has become one of the most popular audio streaming platforms. Despite the uncertainty that COVID-19 imposes on the economy, the company has been able to exceed its Quarter I success metrics. Subscribers and monthly active users have continued to rise with a 31% year-over-year growth from Quarter I 2017 to Quarter I 2020. With that being said, Spotify's cultural prevalence and earnings prove that it is an exceptional platform to publicise and promote music for upcoming and popular artists alike.

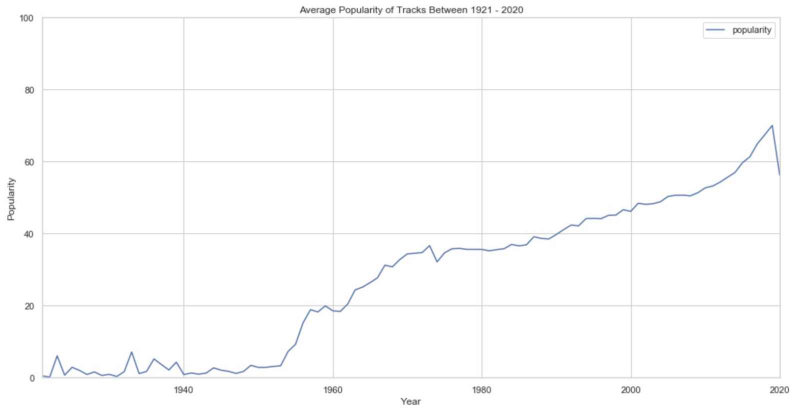
For those looking to increase the number of Spotify streams, I thought it would be fun and possibly helpful to see what makes a song popular on the platform. Using Spotify's Web API, audio features from over 160000 songs were compiled into a dataset. The dataset was compiled by Yamaç Eren Ay who posted his work publicly on Kaggle. There are 19 columns in the dataset: id, acousticness, danceability, energy, energy, instrumentalness, valence, popularity, tempo, liveness, loudness, speechiness, year, mode, explicit, key, artists, release_date, and name. Again, a data dictionary with more information on these features can be found in the Appendix section of the report.

2 Exploratory Data Analysis

The dataset contains 168,592 rows/tracks that are uniquely identified by their Spotify ID. There are 19 rows as mentioned above. It contains tracks that were released between 1921 to 2020, inclusively. While exploring the dataset, I first began by using Python's .describe() method to get quick calculations of central tendency. For the target variable, popularity, the average score was 31.63 out of 100.0. The standard deviation for popularity was roughly 21.39. Additionally, the minimum and maximum popularity scores were 0.0 and 100.0, respectively.

Next, I performed time-series analysis on the target variable to get a better understanding of how popularity of tracks change over time. Popularity of songs began to rise in the 1950s and continues to rise throughout time. There is a drop in average popularity in 2020. This is most likely attributed to the fact the music that has just been released may not have enough time to gain streams and recognition (see Figure 1).

Figure 1: Average popularity of tracks



released between 1921 and 2020

Knowing that popularity of music is rising with recency, I believed that it would be useful to plot the changes in audio characteristics throughout time to see if there was any correlation between the characteristics and popularity. I plotted the changes in the following features: acousticness, danceability, energy, instrumentalness, loudness, spechiness, liveness, and valence (see Figure 2).

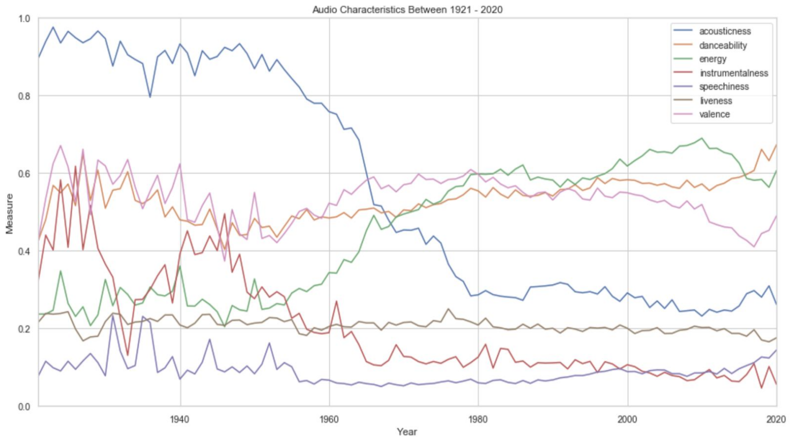


Figure 2: Average values of audio characteristics of tracks released between 1921 and 2020

Throughout time, acousticness of music has seen a drastic fall; however, characteristics such as danceability, energy, and valence (positiveness conveyed in a track) have grown.

I thought it may be helpful to see which artists were the most popular on the platform. I plotted the average popularity ratings for all artists with at least ten or more songs on Spotify, keeping only the top ten artists (see Figure 3). Billie Eilish, Dua Lipa, and Harry Styles were the top three most popular artists as of June 2020. The top 10 artists will be later used in our machine learning model as one-hot encoded features.

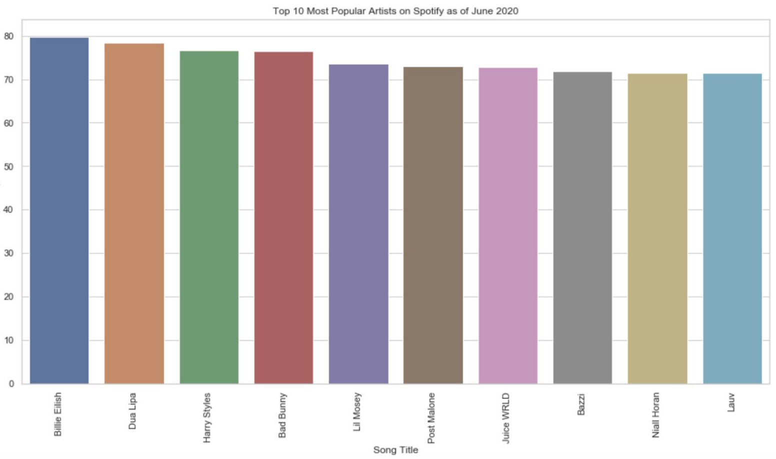


Figure 3. Top ten most popular artists as of June 2020

3 Model Selection

To begin model selection, I used Python's .corr() to examine the linear correlation between the numeric features and the target variable. The .corr() method calculates the Pearson correlation coefficient between two numeric features which is calculated as followed:

$$r = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2\sum(y-\bar{y})^2}}$$

Top features include acousticness, danceability, energy, explicit, instrumentalness, loudness, tempo, and year. The Pearson correlation coefficients for these features are -0.61, 0.22, 0.48, 0.27, -0.31, 0.45, 0.13, and 0.87, respectively. The relatively high values of the Pearson correlation coefficients indicate that the features and the target variable have some linear correlation. Consequently, I used sklearn's linear regression algorithm to create my baseline model.

To measure the performance of my models, I used three metrics: root mean squared error, coefficient of determination (R-squared), and mean absolute error. The range of the target variable, popularity, only ranges between 0 and 100. Because of this, the root mean squared error and mean absolute error will be relatively low. To verify the performance of my models, I used R-squared which tells us how well the model explains the variability of the response data with respect to its mean. The equations for these metrics will be displayed below.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

$$R^2 = 1 - \frac{\overset{\text{Sum Squared Regression Error}}{SS_{Regression}}}{\underset{\text{Sum Squared Total Error}}{SS_{Total}}}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n} |y_j - y_j|$$

3.1 Baseline Model: Linear Regression

I began by training my baseline model using only the numeric features. Using sklearn's linear regression model, I fit my data to an ordinary least squares linear regression model. After predicting on the test set, I measured the model's performance using the metrics stated above. The root mean squared error was 10.23, the mean absolute error was 7.79, and the R-squared value was 0.77. The model performed fairly well with the R-squared value indicating that most of the variability of the model was accounted for.

3.2 Final Predictive Model

To improve my baseline model, I played around with some of the features. One attempt to improve my model consisted of performing log-transformations for some of the features that had high bias or skewed distributions. After adding these features to my training data and making predictions, the model performed slightly worse. For that reason, I decided to leave those features out of the training data.

I then added top artists on Spotify as categorical features for my training data. The list of top artists consisted of the top ten artists referred to in Figure 1. Additionally, artists from "A Decade Wrapped" Spotify's Top Lists 2010 - 2019 who were not featured in Figure 1 were added to as categorical features. These artists include Drake, Ed Sheeran, Ariana Grande, Eminem, Sia, Beyoncé, and Taylor Swift. After one-hot encoding these categorical variables, the linear model saw very slight improvements to the metrics. Even though the improvements were minimal, I decided to keep them in the training data.

Next, I tried training and predicting on different types of regressors. The regressors include sklearn's decision tree regressors, random forest regressors, and gradient boosting regressors. Out of the three new models, the random forest regressor performed the best with a root mean squared error of 9.24, a mean absolute error of 6.65, and a R-squared score of 0.78.

To make final improvements to my model, I used sklearn's RandomizedSearchCV. Only a fixed amount of parameters are evaluated using RandomizedSearchCV in contrast to GridSearchCV; however, GridSearchCV took too much time to execute on my computer. Because there were many instances in the Spotify dataset, I randomly selected 10000 instances to do the cross-validation search. Without this step, the run time would be quite large. After applying the parameters from RandomizedSearchCV, the model saw improvements with a root mean squared error of 9.07, a mean absolute error of 6.54, and a R-squared score of 0.82.

4 Literature

There are a few people who have also used this dataset who have publicly posted their kernels to Kaggle. Many of these users used the dataset for projects such as artist recommendations, genre clustering, or data exploration/visualization.

While looking at reports similar to my studies, I found one project particularly interesting. *Show Me What You Got* by Yiyi Chen, Avi Dixit, Sayan Sanyal, and Ed Yip from UC Berkeley. In their project, they predicted the popularity of songs using custom and pre-existing features such as key, dissonance, and dynamic variation. They explored what features made a track popular based on different genres. For jazz and hip-hop, they found that speechiness, valence and instrumentalness were the most effective in predicting popularity; however, for pop music, they could not find features that were particularly helpful in predicting popularity.

Additionally, they found that speechiness, acousticness, and dissonance were the top three features in predicting popularity — regardless of genre. Regarding musical features (not including time/year) my model's top three features were acousticness, energy, and loudness. The dataset from *Show Me What You Got* used a different dataset and contained more specific features about each track such as chords, pitch, key changes, etc., which may account for the differences in top predictor features.

5 Results

Strong linear correlations between many of the numeric features and popularity led me to use a linear regression model as the baseline. The baseline model performed fairly well with a root mean square error of 10.23, the mean absolute of 7.79, and a R-squared value of 0.77. After testing multiple regression models, I found that sklearn's random forest regressor performed the best. Using RandomizedSearchCV, I was able to obtain some optimized parameters for the random forest regressor (n_estimators=200, min_samples_split=5, min_samples_leaf=4, max_features='auto',max_depth=10, bootstrap=True). For better optimization, GridSearchCV may be a better option; however, due to time and processing limitations, I did not implement this method. The addition of one-hot encoded categorical features made a slight, but insignificant improvement to the model.

Features with strong correlations include acousticness, danceability, energy, explicit, instrumentalness, loudness, tempo, and year. From the correlation coefficients generated during exploratory data analysis, we can observe that acousticness and instrumentalness play a negative role for a track's popularity. Energy, loudness, and the recency of a track (release year) are some of the more important features that play a positive role in the prediction of a track's popularity.

Acknowledgements

I would like to thank Professor Jingbo Shang, Ria Aggarwal, and Dheeraj Mekala for teaching and assisting me throughout DSC 190. I would also like to make a special thank you for the support Shang, Aggarwal, and Mekala have made for students throughout Spring 2020.

References

[1] Yiyi Chen, Avi Dixit, Sayan Sanyal, and Ed Yip, 2017. *Show Me What You Got - Song Popularity Prediction Using FMA Dataset*. University of California, Berkeley, Berkeley, CA.

[2] Spotify. *Shareholder Letter Q1 2020*. (April 2020).

[3] Spotify. *Spotify Wrapped 2019 Reveals Your Streaming Trends, from 2010 to Now*. (December 2019).

Appendix (Data Dictionary)

Primary:

- id (Id of track generated by Spotify)

Numerical:

- acousticness (Ranges from 0 to 1)
- danceability (Ranges from 0 to 1)
- energy (Ranges from 0 to 1)
- duration_ms (Integer typically ranging from 200k to 300k)
- instrumentalness (Ranges from 0 to 1)
- valence (Ranges from 0 to 1)
- popularity (Ranges from 0 to 100)
- tempo (Float typically ranging from 50 to 150)
- liveness (Ranges from 0 to 1)
- loudness (Float typically ranging from -60 to 0)
- speechiness (Ranges from 0 to 1)
- year (Ranges from 1921 to 2020)

Dummy:

- mode (0 = Minor, 1 = Major)
- explicit (0 = No explicit content, 1 = Explicit content)

Categorical:

- key (All keys on octave encoded as values ranging from 0 to 11, starting on C as 0, C# as 1 and so on…)
- artists (List of artists mentioned)
- release_date (Date of release mostly in yyyy-mm-dd format, however precision of date may vary)
- name (Name of the song)