Final Paper DS3001

Team: Joe Leonard (ymd3tv) and Alex Laplace (hqr7gc)

May 7, 2025

Abstract

This project explores the potential of predicting a song's popularity using audio and metadata features available through the Spotify API. Inspired by our experiences as musicians, we sought to understand and optimize music recommendation systems by leveraging machine learning techniques. Using a dataset from Kaggle with over 114,000 Spotify tracks, we conducted an extensive analysis of song-level variables including danceability, energy, loudness, tempo, and genre. After cleaning and preprocessing the data—including one-hot encoding categorical features and handling class imbalances—we applied and compared multiple regression models: Linear Regression, Random Forest, and XGBoost. Our main objective was to determine whether intrinsic audio and metadata characteristics can meaningfully explain and predict a track's popularity score. Among the models tested, XGBoost produced the best performance metrics with a Mean Absolute Error (MAE) of 10.75, Root Mean Squared Error (RMSE) of 15.55, and R^2 of 0.51 after hyperparameter optimization. These results indicate that over half of the variance in song popularity can be explained using song content alone—a strong foundation for further music analytics. In addition to evaluating model performance, we found that general music genres (e.g., pop, dance, indie) tended to dominate among highly popular tracks, suggesting that artist-defined genre labeling can impact discoverability. However, we also recognize limitations in our model, such as its inability to capture external factors such as marketing campaigns or viral trends. This underscores the value of combining content-based methods with collaborative filtering and behavioral data in future research. Ultimately, our findings demonstrate the viability of machine learning in predicting song popularity and provide actionable insights for artists, marketers, and streaming platforms. This work lays the groundwork for more comprehensive models that blend content, listener behavior, and cultural context to better understand what makes a song a hit.

Introduction

In this project, we started out by looking at Spotify track data in order to optimize the way new music is recommended to listeners. We wanted to find similar-sounding songs while incorporating different variables to keep recommendations from becoming repetitive. The reason we chose to go this route is because we are both in bands at UVA and thought it would be a cool way to incorporate that into our data science work. We thought it would be interesting to get an inside look at how the recommendation system works, especially in the context of us releasing new music to the public, and trying to reach the largest audience possible.

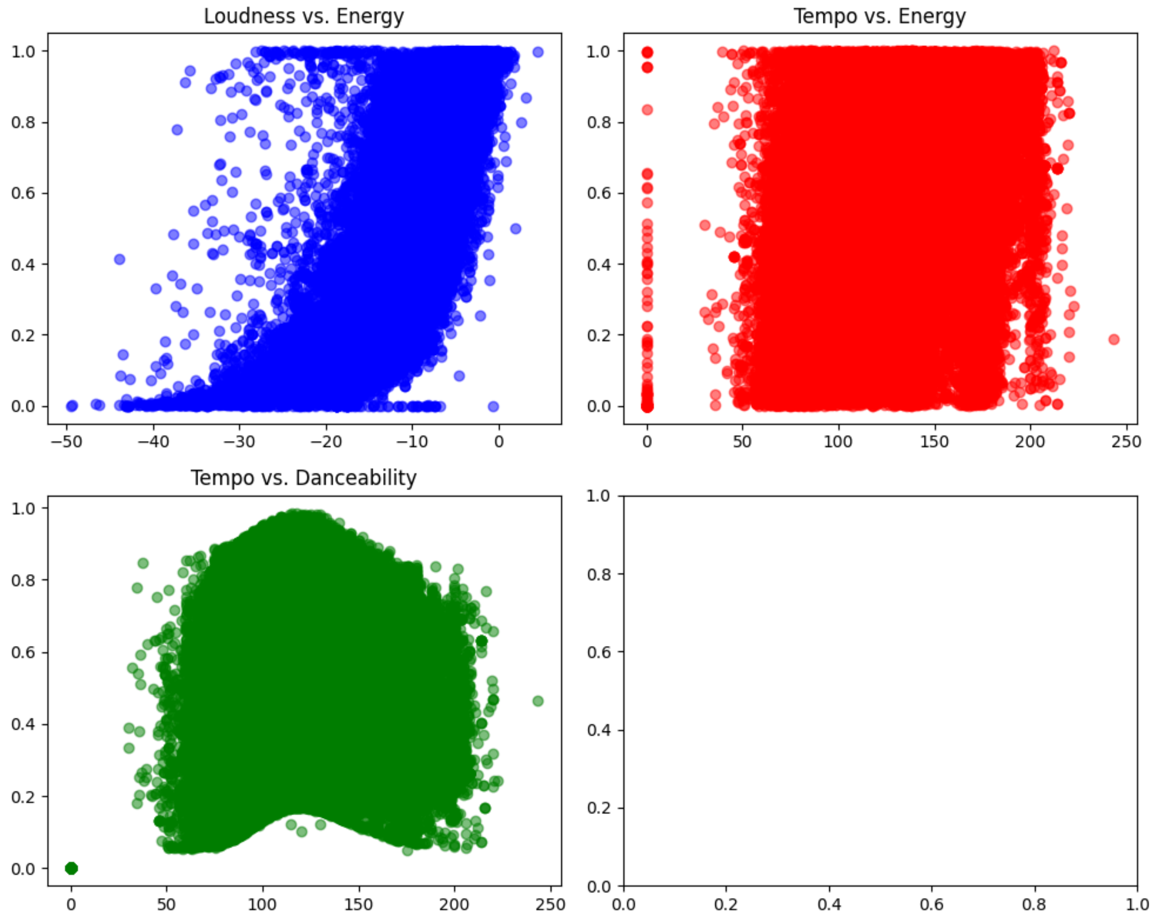
For help on this, we've referred to previous work on spotify recommendation systems. Spotify uses two main methods to provide users with new songs: Collaborative and Content-based filtering (Chang, 2023; Dieleman, 2014; Mavani, n.d.). Content-based filtering uses quantitative metrics to suggest song recommendations tailored to your preferences based on the audio characteristics of songs that you've previously enjoyed. Collaborative filtering uses listening history to identify users who share similar listening habits and suggests songs that another user with similar preferences has enjoyed that you have not explored yet. One method we came across used content-based filtering for their recommendations and our analysis focuses on using song content data to predict popularity (Chang, 2023). This paper reviews how we began with input data from Kaggle in the form of songs and descriptive variables for each song record. After cleaning the data, we assessed it for potential trends between categories that stood out to us such as tempo and danceability or loudness and energy. After finding that there was very little correlation, we filtered out songs with popularity scores over 75 out of 100 to figure out if track_genre and popularity had any kind of relationship and found that more general genres have higher counts of popularity over 75. Our research question evolved into testing whether we could accurately predict a song's popularity score using audio and metadata features available through the Spotify API? This question is relevant to record labels, streaming platforms, and music marketers who want to identify potential hit songs early using metadata. The models we used were Linear Regression, Random Forest, and XGBoost. Because XGBoost's Mean Absolute Error (MAE), Root Mean Squared Error, and R-Squared values were cumulatively the best performing, we chose to optimize the model fitting 4 folds for each of 25 candidates for a total of 100 fits. Our optimized XGBoost statistics were: MAE was 10.754, RMSE was 15.546, and R2 was 0.510. After applying random search to tune hyperparameters for XGBoost, our model's ability to predict popularity improved significantly and demonstrates meaningful explanatory power. However, just under half of the popularity is therefore unexplainable by our model, resulting from other non-metadata related factors. While our model makes a step toward explainable music analytics, it also invites future research that blends content-based models with behavioral, cultural, and temporal data to capture the full complexity of music popularity.

Data

For this project, we'll be using the Spotify Tracks Dataset from Kaggle (Pandya, 2022). This dataset has numerically evaluated variables that are connected to various parts of a song. Taking a look at our data, there are 114,000 records and 21 variables. Each record is a song and the variables are as follows: 'Unnamed: 0', 'track_id', 'artists', 'album_name', 'track_name', 'popularity', 'duration_ms', 'explicit', 'danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo', 'time_signature', and 'track_genre'. Immediately we saw 'Unnamed: 0'. This column likely comes from downloading the data from Kaggle, so let's get rid of it to avoid later problems. It may act as a general identifier, but we already have 'track_id' as a variable, so we don't need to double down. Looking at the number of NAs in each column, we only found 1 record that had NA values, likely because of the data source being Kaggle and thus we could go into our analysis. The first thing we wanted to take a look at is how some of these variables are connected. Some of the variable descriptions led me to believe they would be pretty directly correlated. If so, we can find a way to use this to our advantage when modeling. The following variable descriptions lead me to believe this: **danceability:** Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable, **energy**: Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale, **loudness**: The overall loudness of a track in decibels (dB), **tempo**: The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.

Our assumption is that high energy songs will also be associated with high loudness or tempo. High danceability takes into account temp meaning it'll also have a high tempo. To test this, we created scatter plots for Loudness vs. Energy, Tempo vs. Energy, and Tempo vs. Danceability as seen in Figure 1.

Figure 1. Scatterplots of Variables of Interest



Our assumption was not correct and thus we were able to use the data independently. We hypothesized that we might get into a linear relationship when looking at 'loudness' and 'energy' or a negative parabolic relationship with 'tempo' and 'danceability'. Next we filtered out songs with a popularity score over 75 to see if there is a connection between 'track_genre' and 'popularity'. We thought we might find an issue where Spotify is leaning towards recommending a song more purely based on its genre.

Table 1. Top and Bottom 10 Genres with Popularity Scores over 75

Top 10 Genres		Bottom 10 Genres	
pop	181	industrial	2
dance	178	pagode	2
rock	158	metalcore	2
latino	120	acoustic	2
indie	116	j-dance	1
reggaeton	105	club	1
electro	105	classical	1
house	97	children	1
indie-pop	89	brazil	1
alternative	87	turkish	1

We can see that more general genres (indie, dance, alternative) are favored and more likely to be popular than specific genres (industrial, metalcore, turkish). One could argue that acoustic is a pretty general genre, but in reality acoustic songs are tied to another genre. It could be indie, alternative, or country, but artists likely don't put their genre down as 'acoustic'. We can already see the value in this as we could recommend to artists to generalize their songs when it comes to genre selection. These genres were not auto populated by any data, but instead submitted by the artist.

Methods

Here we will detail our plan of analysis. Each observation in our study corresponds to a song listed in Spotify's dataset, including attributes such as its popularity, duration, tempo, danceability, and other relevant audio features. The type of machine learning we are conducting is supervised learning, specifically focusing on regression to predict a song's popularity based on its features. We plan to experiment with multiple models to take an approach that touches on various areas of modeling and machine learning. Those models include: Linear Regression, Random Forest Regression, Gradient Boosting (XGBoost), and Neural Networks. Success will be evaluated using: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) for regression performance, cross-validation scores to ensure model generalization, and feature importance analysis to interpret key predictors of song popularity. Our anticipated weaknesses & mitigation includes: Feature Correlations - some features may be highly correlated, requiring dimensionality reduction or feature selection techniques, Data Imbalance - popularity scores may be skewed, requiring potential transformations or resampling, Overfitting - regularization techniques will be used to prevent overfitting, and Unpredictable External Factors - song popularity can be influenced by external trends, which our model cannot capture and which we will acknowledge in our analysis. Our feature engineering will include: one-hot encoding categorical features such as key and mode, normalization of numerical features like tempo and loudness, and creating interaction terms for potential synergies between features. To share our results we will use: Visualizations, Performance Metrics Tables, and Model Interpretability Discussions. Ultimately, our pre-analysis plan ensures a structured approach to predicting song popularity using machine learning techniques while addressing potential limitations and emphasizing interpretability.

Results

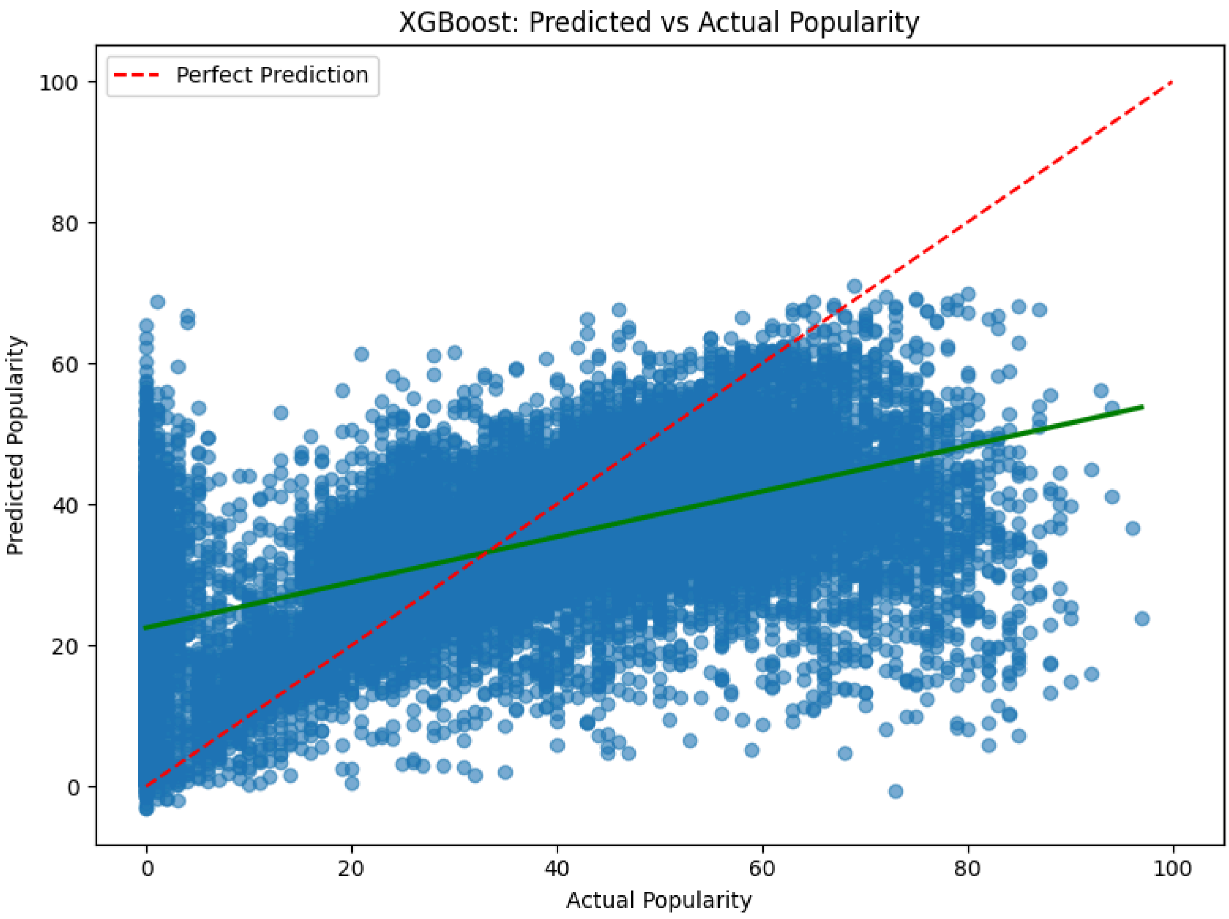
Our research question is as follows: Can we accurately predict a song's popularity score using audio and metadata features available through the Spotify API? This question is relevant to record labels, streaming platforms, and music marketers who want to identify hit songs early. In order to test this, we first had to prep our data for use in our models. We used the columns: 'popularity', 'duration_ms', 'explicit', 'danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo', 'time_signature', 'track_genre'. We then dropped our NA values, cased the 'explicit' variable as integer, and one-hot encoded the variables 'key', 'mode', 'time_signature', and 'track_genre'. We used a Linear Regression, Random Forest, and XGBoost model on our cleaned spotify data.

Table 2. Mean Absolute Error, Root Mean Squared Error, and R Squared Values by Model

	MAE	**RMSE**	**R2**
Linear Regression	14.086871	19.119256	0.259250
Random Forest	10.297055	15.270064	0.527490
XGBoost	13.093322	17.919202	0.349321

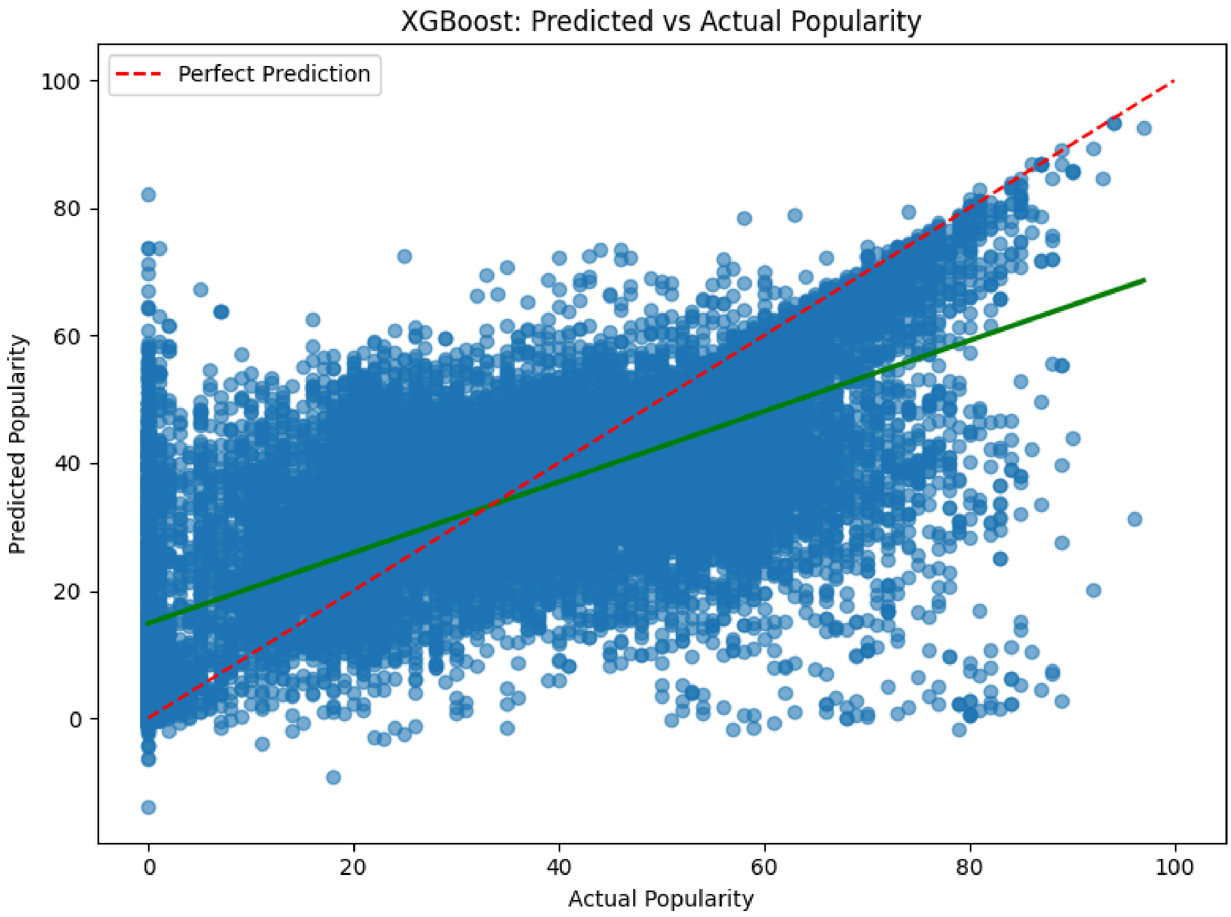
We then plotted the actual popularity against the predicted popularity for the XGBoost model seen in Figure 2. Figures 2 and 3 each include A red dashed line indicating perfect predictions (where predicted = actual) and a green line showing the best fit line for the model’s actual predictions.

Figure 2. Initial XGBoost Predictions vs. Actual Popularity



Then, we optimized XGBoost fitting 4 folds for each of 25 candidates for a total of 100 fits. Our MAE was 10.754, RMSE was 15.546, and R2 was 0.510. The Predicteed vs Actual Popularity is shown below in this scatter plot in Figure 3.

Figure 3. Optimized XGBoost Predictions vs. Actual Popularity



In the initial model (Figure 2), predictions tend to cluster too tightly around a narrower popularity range, especially underpredicting higher values. After optimization (Figure 3), the slope of the best-fit line becomes steeper and better aligns with the identity line, suggesting improved tracking of actual popularity across its range. The scatter also tightens slightly, indicating reduced prediction error. Quantitatively, the model performance after optimization improved to: Mean Absolute Error (MAE): 10.75, Root Mean Squared Error (RMSE): 15.55, and R^2 Score: 0.51. This R^2 value means that over half the variance in popularity scores is now explained by the model, a marked improvement over the unoptimized version (whose R^2 was visibly lower, as suggested by the looser scatter and flatter fit).

After applying random search to tune hyperparameters for XGBoost, our model's ability to predict popularity improved significantly. While still imperfect, this version of the model is now better aligned with actual values and demonstrates meaningful explanatory power.

Conclusion

In this project, we set out to determine whether it is possible to accurately predict a song's popularity using quantitative audio and metadata features from the Spotify API. Through a methodical process involving data cleaning, feature engineering, and model experimentation, we were able to construct and optimize a predictive model using XGBoost that explains over half the variance in song popularity ($R^2 = 0.51$). Our final model demonstrated strong performance metrics (MAE = 10.75, RMSE = 15.55) compared to the baseline models we tested, including Linear Regression and Random Forest.

One of our key insights came not only from the final performance metrics, but from how the model's visual fit improved after optimization. The refined XGBoost predictions aligned with the identity line and reduced clustering bias around the middle range of popularity scores, especially for higher popularity values. These findings underscore the potential for content-based models to support decision-making in music marketing and recommendation systems, particularly in contexts where collaborative filtering data is sparse or unavailable.

However, the model is not without limitations. While 51% of variance explained is meaningful, nearly half remains unexplained likely due to external, non-quantifiable influences on popularity such as marketing campaigns, social media virality, or artist reputation. This highlights a major limitation of content-based approaches: they can only predict popularity based on intrinsic song features, not the cultural or contextual forces that often drive listener behavior.

Additionally, genre labeling posed some challenges, as artist-submitted genre tags varied in granularity and consistency, potentially reducing predictive clarity.

Looking forward, several promising extensions could be explored. First, integrating collaborative filtering signals could enhance predictive power by capturing social and behavioral dynamics. Second, natural language processing techniques could be applied to song lyrics, if available, to capture emotional or thematic content. Third, time-series analysis of release timing and seasonal listening trends could add another layer of predictive context, helping account for external patterns that influence popularity. Finally, a live feedback system incorporating listener reactions could turn this static model into a dynamic one that evolves as listener preferences shift.

In conclusion, this project validates the feasibility of predicting Spotify track popularity from audio and metadata features and provides a solid foundation for further investigation. While our model marks a meaningful step toward explainable music analytics, it also invites future research that blends content-based models with behavioral, cultural, and temporal data to capture the full complexity of music popularity.

References

Chang, J. (2023, December 9). Data-Driven Music Exploration: Building a Spotify Song Recommender. *Medium*. https://medium.com/@joshjc038/data-driven-music-exploration-building-a-spotify-song-recommender-5780cabfe194

Dieleman, S. (2014, August 5). *Recommending music on Spotify with deep learning*. Sander Dieleman. https://sander.ai/2014/08/05/spotify-cnns.html

Mavani, V. (n.d.). *Music Recommendation System using Spotify Dataset*. Retrieved May 7, 2025, from https://kaggle.com/code/vatsalmavani/music-recommendation-system-using-spotify-dataset

Pandya, M. (2022). *🎹 Spotify Tracks Dataset* [Dataset]. https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset