AMJES | American Journal of Emerging Scholars

Demystifying the Popularity of Songs Using Machine Learning Algorithms

Albert Wang
Miramonte High School, California, USA

Abstract
Song popularity is an influential subject within the modern music streaming industry. It determines which artists can gain media attraction, gather loyal fans, and ultimately succeed. Analyzing song popularity with ML algorithms contributes to demystifying success within the music industry. Two datasets, datasets 1 and 2, collected from the Spotify Web API contain audio information on respectively 2000 songs and 240,057 songs. Ordinary Least Squares Linear Regression (OLS LR) and Neural Network (NN) algorithms were used on each dataset to predict song popularity. The most complex NN structure used in this study contains three hidden layers, achieving the best regression performances on both datasets; however, it was superior to other models by a small margin. Overall, models trained with dataset 2 achieved superior results, particularly in the R^2 metrics, but were unimpressive due to low regression metrics.

Introduction
Music entertainment has had a substantial influence on various aspects of society over time. The creation of music dates to tens of thousands of years ago; flutes made of bone and ivory were found in a cave in Germany by archaeologists, which were believed to demonstrate signs of the "well-established musical tradition" present within human society at the time (Wilford, 2009). Methods of musical expressions have evolved drastically since. Technological inventions continuously push forward the limits and bounds of musical creation and enjoyment, and with the internet boom in the 21st century, every aspect of music entertainment has been thoroughly revolutionized.

Within recent years, the music industry has become dominated by subscription-based music streaming services. "Music streaming in the U.S. contributed $14.32 billion to the U.S. gross domestic product (GDP) in 2021", stated Digital Media Association (Stoner & Dutra, 2023). The industry is currently led by companies like Spotify and Apple Music, who stand as the pioneers of this new modern streaming model for music enjoyment.

The success of streaming services simultaneously fostered the rise and evolution of digital music. Streaming services allowed for the transition from music downloads and enabled users to access high-quality music with minimal effort and waiting time, drastically increasing overall accessibility to music. The new era of digital music additionally enables the usage of data-driven approaches for enhancing customer experiences. Through collecting and analyzing vast amounts of customer data, companies like Spotify have developed recommendation algorithms capable of understanding users' music consumption behaviors. This enables Spotify to provide features such as "Spotify Blend" which generates personalized playlists for one or more users that fit their taste, "Enhance Playlist" which adds new songs to a playlist, and numerous other algorithm-driven features (*Inside Spotify's Data Mission*, 2017).

By focusing the research on the analysis of audio data collected by Spotify, this study can imitate the research that has already been done by music streaming companies like Spotify but remained confidential; however, given the substantially higher volumes of data accessible by corporate entities, the volume of data is likely a limited factor for this study.

If Machine Learning or other regression algorithms can reasonably predict song popularity using audio features, there are substantial implications for not only corporate entities but also individual artists. Music streaming platforms capable of predicting hits can selectively promote songs to garner more users and media attention. On the other hand, artists who hope to increase their popularity can use ML algorithms as either a preemptive popularity analysis or as a guiding tool during the song creation process.

If the prediction of song popularity is inaccurate or unreliable, it can be concluded that audio features alone cannot reasonably explain the complexity of music popularity. This may be due to the taste of the population being too overly diverse, which could result in difficulties grasping correlations. Or the relative objectiveness of audio features is unable to explain popularity given its subjective nature.

Using audio feature data collected by Spotify, this study analyzes the modern trend in music with the utilization of Machine Learning algorithms and linear regression models. The overarching research topic is whether the popularity of a song can be predicted by using its musical metrics. Specifically, the objective of this study is to 1) understand how attributes contribute to the popularity of songs; 2) understand the complexity of the problem by applying the same algorithms to different numbers of samples.

Methods

Data Information

Two datasets of varying magnitudes were selected to understand the complexity of the subject in relation to the data volume. The datasets are labeled as dataset 1 and dataset 2.

Dataset 1

Dataset 1 is Top Hits Spotify from 2000-2019 from the open Kaggle database (Koverha, 2022). This dataset is sourced from the official Spotify Web API. It contains 18 song attributes and 2000 samples. Each sample corresponds to one of the top 2000 tracks from 2000-2019. The summary of the 18 song attributes contained in Dataset 1 is shown in Table 1.

Table 1. Track Attributes Information for *Top Hits Spotify from 2000-2019.* Documentation is collected from official Spotify Web API documentation (https://developer.spotify.com/documentation/web-api) and Kaggle dataset documentation.

Attribute	Mean	Standard Deviation	Description	Data Type
Artist	N/A	N/A	Name of artist	String
Song	N/A	N/A	Name of track	String
Duration_ms	228748.125	39136.569	Duration of the track in milliseconds	Integer
Explicit	N/A	N/A	Whether or not the track has explicit lyrics. Explicit lyrics are denoted with True. Non-explicit lyrics are False.	Boolean
Year	2009.494	5.860	The release year of the track	Integer
Popularity	59.873	21.336	The popularity of the album. The value will be between 0 and 100, with 100 being the most popular.	Integer
Danceability	0.667	0.140	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.	Float
Energy	0.720	0.152	Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity.	Float
Key	5.378	3.615	The key the track is in. Integers map to pitches using standard Pitch Class notation.	Integer
Loudness	-5.512	1.933	The overall loudness of a track in decibels (dB).	Float

Mode	0.554	0.497	Mode indicates the modality (major or minor) of a track. Major is represented by 1 and minor is 0.	Binary
Speechiness	0.104	0.0962	Speechiness detects the presence of spoken words in a track.	Float
Acousticness	0.0962	0.173	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.	Float
Instrumentalness	0.0152	0.0878	Predicts whether a track contains no vocals.	Float
Liveness	0.181	0.141	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live.	Float
Valence	0.552	0.221	A measure from 0.0 to 1.0 that describes the musical positiveness conveyed by a track. Tracks with high valence sound more positive, while tracks with low valence sound more negative.	Float
Tempo	120.122	26.967	The overall estimated tempo of a track in beats per minute (BPM).	Float
Genre	N/A	N/A	Genre of the track.	String

Dataset 2

Dataset 2 is *Spotify Audio Features* from the Kaggle database (Tomigelo, 2019). This dataset contains two sets of data retrieved from the Spotify API. The first set contains 130,326 unique songs retrieved in April 2019. The second set contains 116,191 unique songs retrieved in November 2018. The two sets were combined into a single dataset of of 247,035 samples. The 17 attributes contained in this dataset are described in Table 2.

Table 2. Track Attributes Information for *Spotify Audio Features.*

Attribute	**Mean**	**Standard Deviation**	**Description**	**Data Type**
Artist_name	N/A	N/A	Name of artist	String
Track_id	N/A	N/A	Name of track	String
Track_name	N/A	N/A	Name of track	String
Duration_ms	212592.159	123705.356	Duration of the track in milliseconds	Integer
Popularity	24.221	18.895	The popularity of the album. The value will be between 0 and 100, with 100 being the most popular.	Integer
Danceability	0.582	0.190	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.	Float
Energy	0.570	0.259	Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity.	Float
Key	5.236	0.603	The key the track is in. Integers map to pitches using standard Pitch Class notation.	Integer

Loudness	-9.960	6.525	The overall loudness of a track in decibels (dB).	Float
Mode	0.608	0.488	Mode indicates the modality (major or minor) of a track. Major is represented by 1 and minor is 0.	Binary
Speechiness	0.112	0.124	Speechiness detects the presence of spoken words in a track.	Float
Acousticness	0.339	0.344	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.	Float
Instrumentalness	0.227	0.362	Predicts whether a track contains no vocals.	Float
Liveness	0.195	0.168	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live.	Float
Valence	0.439	0.259	A measure from 0.0 to 1.0 that describes the musical positiveness conveyed by a track. Tracks with high valence sound more positive, while tracks with low valence sound more negative.	Float
Tempo	119.535	30.156	The overall estimated tempo of a track in beats per minute (BPM).	Float
Time_signature	119.535	0.512	Genre of the track.	String

Kaggle Notebook, a cloud computational environment, is used for data preprocessing and the implementation of regression models such as linear regression and neural networks. All processing was done with Python 3.7. The performance of models trained separately on each dataset will be compared to understand how the quantity of data affects regression performance,

Data Preprocessing

Data Quality

Both datasets were examined using the built-in Kaggle data viewer and the Pandas library. Results show that there are no missing data points.

Feature Selection

Both datasets contain several attributes that only contributed to song identification (e.g., song names, artists names, and track_id). These attributes do not contribute to the analysis of song popularity and are excluded from both datasets.

Data Cleaning

In dataset 1, the genre column contained default values that were excluded from the dataset. In addition, 63.95% of songs in dataset 1 were categorized into more than one genre (e.g.“pop, hip hop, and R&B). New identical samples except with singular genres were added to reduce the number of unique categorical values. For example, a single song can be categorized into pop, hip hop, and R&B. The song would then be replaced by three samples containing identical audio attributes but with their genres being respectively pop, hip hop, and R&B.

One-Hot-Encoding

In dataset 1, the one-hot-encoding technique was used on the genre column to obtain usable data. Each unique genre string value was assigned a new column with binary values such that the data could contribute to Machine Learning algorithm analysis.

Outlier Treatment

For song attributes in numerical format (integer or float) in both datasets, outlier treatment was independently applied on each of the columns. The mean and standard deviation of each numerical column were computed to detect outliers. The standard deviation defined as:

$$SD = \sqrt{\frac{\sum (x_i - \overline{x})^2}{N}}$$

where N is the total number of observations and $\overline{x}$ is the mean.

For each column, values 3 standard deviations distant from the mean were detected as outliers. The outlier criteria were chosen as such to account for the high variability within the dataset. An outlier boundary of 3 standard deviations identifies a reasonable proportion of 2% ~ 6% of the two datasets as outliers. After applying the method, 217/3,682 outliers were excluded from dataset 1, and 5,022/247,035 outliers were excluded from dataset 2.

Linear Regression

Linear regression is a model that uses a linear relationship to predict the relationship between variables.

Train Test Split

The train-test ratio used for linear regression is 75:25.

Metric

The Scikit-learn library is used to execute a simple ordinary least squares linear regression (OLS). In OLS, the coefficient of determination (R^2) serves as the evaluation metric. R^2 is defined as:

$$R^2 = \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \overline{y})^2}$$

where f is the predicted value by the model and $\overline{y}$ is the mean of the observed data.

Neural Networks

In this study, NN structures are built using the Sequential class from the Keras Library. Models are trained and compiled with a mean squared error (MSE) loss function, the conventional choice for regression. In addition to the loss function, the scikit-learn library is used to calculate R^2 (coefficient of determination) and mean absolute error (MAE), both as metrics to assess the performance of a model.

K-Fold Cross-validation

Cross-validation is used as a data resampling technique to iteratively partition the data into train and test portions. 4-fold cross-validation was used for both dataset 1 and dataset 2 to train and evaluate NNs.

Data Scaling

Using the StandardScalar module in the Scikit-learn library, features were scaled before being used for the cross-validation process.

Optimizer Algorithms

Adaptive moment estimation (Adam), stochastic gradient descent (SGD), and root mean square propagation (RMSprop) optimizers were compared during NN development. SGD optimizer was chosen as the optimal optimizer due to its superior efficiency during training.

Neural Network Structures

Model structures with variable parameters were tested at three separate levels of complexity: one, two, and three hidden layers. Within each level, the model that produced the highest metrics (R^2) was chosen to be shown below.

Model A: 1 hidden layer
The development of the NN model began with a model containing three dense layers, an input layer, a hidden layer, and an output layer. The input layer and output layer contain nodes adjusted to the same number of nodes as respectively the number of features and labels. One hidden layer of 16 nodes using the RELU activation function was placed between the input and output layers. Batch normalization layers were placed after each hidden layer. The optimizer used is SGD with a learning rate of 0.001 and momentum of 0.4. 40 epochs were used, and an early stopping regularization with a patience of 5 epochs was implemented.

Model B: 2 Hidden Layers
Model B was built with an input layer, two hidden layers using the RELU activation function, and an output layer. Input and output layers are adjusted to the same number of nodes as respectively the number of feature and label columns. The first hidden layer contains 16 nodes while the second contains 32 nodes. Batch normalization layers are placed after each hidden layer. The optimizer used is SGD with a learning rate of 0.0001 and momentum of 0.4. 50 epochs were used, and an early stopping regularization with a patience of 5 epochs.

Model C: 3 Hidden Layers
Model C was developed with an input layer, three hidden layers using the RELU activation function, and an output layer. Input and output layers are adjusted to the same number of nodes as respectively the number of feature and label columns. The first hidden layer contains 16 nodes; the second and third hidden layers contain 32 nodes. Batch normalization layers are placed after each hidden layer. The optimizer used is SGD with a learning rate of 0.0001 and momentum of 0.4. 70 epochs were used, and an early stopping regularization with a patience of 5 epochs.

Results

Linear Regression

Dataset 1 and dataset 2 were independently split into train and test data. The linear regression model is fitted on the train data and evaluated on the test data. The results are shown in Table 1.

Table 1. Linear Regression performances in cross-validation.

Regression Metrics	**OLS[2] Linear Regression**	
	Dataset 1	**Dataset 2**
R^2	0.0166	0.0843
MAE[1]	14.593	14.669

[1]Mean Absolute Error
[2]Orindary Least Square

Neural Network

Dataset 1 and dataset 2 were separately used to train and evaluate each NN model (model A, model B, and model C). The cross-validation results are shown in Table 2.

Table 2. Neural Network performances in cross-validation.

	Neural Network Models					
Regression Metrics	**Model A**		**Model B**		**Model C**	
	Dataset 1	**Dataset 2**	**Dataset 1**	**Dataset 2**	**Dataset 1**	**Dataset 2**
R^2	0.00750 ± 0.00421	0.133 ± 0.00433	0.00750 ± 0.0109	0.133 ± 0.00433	0.0250 ± 0.0364	0.135 ±0.00412
MAE^2	14.325 ± 0.313	14.293 ± 0.0245	14.377 ± 0.332	14.293 ± 0.0245	14.306 ± 0.435	14.176 ± 0.0446

Discussion

Evaluating regression models (LR, NN) trained on dataset 1, NN model C achieved the highest R^2 of 0.0250 ± 0.0364 while all LR and NN models achieved comparable MAE, averaging 14.400. This suggests that the complexity of model C enabled it to further grasp the relationship between features and labels; however, the resulting improvement in popularity prediction was non-substantial.

Examining regression models (LR, NN) trained on dataset 2, NN model C achieved the highest R^2 of 0.135 ± 0.00412 and MAE of 14.176 ± 0.0446, while LR achieved R^2 of 0.0843 and MAE of 14.669.

Comparing the performance of NN separately on each dataset, models trained on dataset 2, which contain a greater number of samples, have shown significantly higher R^2 regression metrics than models trained on dataset 1. The model with the highest R^2 values for both datasets, NN model C, achieved an R^2 value 0.11 greater for dataset 2 than dataset 1. However, the relative difference in MAE between the two datasets is only 0.917% for the same model.

The best-performing model by a small margin, model C trained with dataset 2, achieved results that were not exceptionally high. An MAE of 14.176 and an R^2 of 0.135 is low under the consideration that the output value, popularity, is a value between 0 and 100, and R^2 of 0.135 indicates only 13.5% of the variability of the output variable around its mean is explained by the model.

Numerous methods were applied to improve the likelihood of regression models grasping the relationships between the audio features and the popularity. During data preprocessing, outliers for columns in both datasets were detected and excluded to increase the signal-to-raise ratio and emphasize the central contributing factor to popularity in both datasets. Features were scaled before cross-validation to normalize the range of values and improve the stableness and efficiency of NNs. During NN training, batch normalization layers re-center and re-scale the input for the subsequent hidden layer to further stabilize training, which can improve NN performances.

Interpretations

The results confirm several hypothesized relationships. In this study, the audio features of a song could not reasonably predict its popularity. This aligns with the hypothesis that the pattern in popularity may be overly complex or diverse as the Machine Learning models in this study were unable to extract clear variable correlations in the datasets. Reliable predictions of popularity will require additional data beyond audio features, which is addressed in Related & Future Studies.

In addition, increases in dataset sizes showed substantial improvement in the R^2 metric and could suggest that the trend within music popularity will become apparent in larger sample sizes. The structural complexity in Neural Networks were not strongly associated with performance gain. Addressing limitations in future studies may result in more conclusive insights on the importance of model complexity in the context of predicting music popularity.

Limitations

The low regression performance of this study can be reasonably attributed to three limiting factors: 1) the inherent complexity of music popularity, 2) the selection of data, and 3) the selection of ML algorithms.
Because popularity is a product of countless factors, "External" factors such as a song suddenly going viral on social media due to a particular influencer, or a particular genre gaining traction in music communities can substantially impact the popularity of a song while being nearly independent of the audio features. Furthermore, audio features do not include information on traits such as sentiments and lyrics. These traits can provide additional details and contribute to more comprehensive representations of songs.

The performance of models is highly dependent on the data available for analysis. Both the quantity and the selection of data are significant factors in contributing to the extent to which regression models can grasp relationships between features and labels, as shown by the result comparison between dataset 1 and 2. Both datasets used for analysis are both collected from the Official Spotify Web API; however, the context of the data collected is not perfectly consistent. Dataset 1 contains the top 2000 songs from 2000~2019 on Spotify; dataset 2 contains data on 240,000+ songs of varying popularity, collected in 2018 and 2019. The difference in context resulted in a different sample of song chosen; as a result, the mean popularity in dataset 1 is 35.652 higher than that of dataset 2, which could have influenced the results.

In this study, OLS LR and NN are the only ML algorithms utilized for analysis. As a result, there may be a possibility that other ML regression algorithms are more suitable for the analysis of song popularity due to their unique characteristics. The diverse tastes of music enjoyers may have resulted in more segmented patterns between features and labels, resulting in the underperformance of LR and NN.

Related & Future Studies

Recent studies have investigated the effect of feature selection on the accuracy of ML algorithms when applied to classifying popularity, using logistic regression algorithms, random forest algorithms, and k-nearest neighbor algorithms. (Khan et al., 2022). The selection of features with only high correlation has been shown to produce comparable performance to algorithms without feature selection, with the benefits of reduced computation time.

A future study would be to utilize a large master dataset for analysis. This dataset would contain a larger number of songs collected from the Spotify Web API under the same context, e.g. randomly collecting a data sample of 500,000 songs from the Spotify Web API. The study can apply sentimental lyrical analysis and other similar techniques to expand the scope of features. Metrics that measure current trends, public opinions, artist popularities, or other external factors should be considered in the analysis as well to improve regression performance.

The master dataset can be used to create subsets of varying magnitudes of volume such that a selection of regression methods, with additional ML algorithms that may be more suitable, can be applied to each subset separately.

Conclusion

In this study, the relative performance of regression algorithms (LR, NN) on two datasets with different magnitudes of data volume, dataset 1 with 2000 samples and dataset 2 with 247,035 samples, were compared. Results show that NN model C, the most complex regression model used, performed the best on dataset 2 with a coefficient of determination of 0.1325 and MAE of 14.176. However, the performance of NN model C on dataset 2 is only superior to the performance of other models, trained on either dataset, by a small margin. Overall, the regression models (LR, NN) used in this study underperformed when trained independently on both dataset 1 and 2, reflecting the complex relationship between music popularity and song attributes. The component of human behavior is hypothesized to be a key factor contributing to the low predictability of music popularity achieved with LR and NN; the subjectiveness of song popularity cannot be fully represented by the audio features alone. Future studies can utilize more refined data collection techniques and a greater variety of ML algorithms to analyze the influence of data volume on popularity prediction performance and further dissect the popularity of music.

References

Get tracks' audio features. Web API Reference | Spotify for Developers. (n.d.). https://developer.spotify.com/documentation/web-api/reference/get-several-audio-features

Khan, F., Tarimer, I., Alwageed, H. S., Karadağ, B. C., Fayaz, M., Abdusalomov, A. B., & Cho, Y.-I. (2022). Effect of feature selection on the accuracy of music popularity classification using machine learning algorithms. *Electronics, 11(21)*, 3518. https://doi.org/10.3390/electronics11213518

Koverha, M. (2022, May 31). Top hits Spotify from 2000-2019. Kaggle. https://www.kaggle.com/datasets/paradisejoy/top-hits-spotify-from-20002019

Stoner, R., & Dutra, J. (2023, April). An Economic Analysis of the Impact of Digital Music Streaming. Washington; Digital Media Association (DiMA). https://digmedia.wpenginepowered.com/wp-content/uploads/2023/04/An-Economic-Analysis-of-the-Impact-of-Digital-Music-Streaming_April-2023.pdf

Spotify Editorial Team. (2017, October 17). Inside Spotify's Data Mission. Spotify Advertising. https://rb.gy/sn6i3

Tomigelo. (2019, April 14). Spotify audio features. Kaggle. https://www.kaggle.com/datasets/tomigelo/spotify-audio-features

Wilford, J. N. (2009, June 24). Flutes offer clues to stone-age music. The New York Times. https://www.nytimes.com/2009/06/25/science/25flute.html?scp=1&sq=nicholas+j+conard&st=cse.