MIM JOURNAL 2023 illuminate your work
Made in Millersville Journal | www.mimjournal.com

Machine Learning Techniques to Improve Users' Music Listening

Samantha Noggle

Abstract

Recommender systems play a crucial role in delivering a personalized user experience. These algorithms utilize the data they are given about a user to recommend content that is most relevant to them. This technique is especially prominent in music streaming services like Spotify, which aim to curate music based on listeners' preferences to maintain engagement and stay ahead in a competitive market. This article proposes a model designed to predict whether a user will skip a song based on information available within the user's current listening session. A dataset provided by Spotify for a machine learning challenge was utilized and augmented with features that we propose to help measure the "distance" between a song, as well as songs that have been either listened to or skipped within the same session. The final model that we developed with our engineered data can predict whether a user will skip the final song in their session with 78.23% accuracy. An analysis of feature importance is also presented to better understand which factors might be most influential in predicting this user behavior. The ability to predict whether a song will be skipped or played could be utilized in future work to dynamically impact song selection within a listening session to minimize the number of songs being skipped by the user.

Introduction

Music streaming platforms implement extensive and complex recommender systems to predict exactly what a user wants to listen to. Giving relevant recommendations is extremely important, as they improve a user's listening experience and overall satisfaction. In a competitive market with many services vying for each user's time and money, the ability to accurately predict what an individual user wants becomes a critical factor in the success of a music streaming service such as Spotify, Apple Music, or YouTube Music. Our research questions are as follows: (1) can we train a model with session-based data to predict if a user will skip a song, and (2), which features within the data play the biggest role in those predictions? We also hypothesize that a user's preferences will remain consistent within a session and that they will skip and play songs the same way they have done earlier in the session. For the purpose of this research, we focus on Spotify's dataset.

Approach

A random forest classifier is an ensemble learning method that operates by constructing a multitude of decision trees at training time. The output of the random forest classifier, the class selected by most trees, was trained to predict if a listener would skip the final song in a session using 200,000 listening sessions. Our focus on the final song is strictly to predict with as much of the session's history as possible. This data was a subset of a dataset of 130 million sessions released by Spotify in collaboration with The ACM International Conference on Web Search and Data Mining (Brost et al., 2019). The dataset was originally used in a challenge to predict song-skipping behavior in the first half of a session. After preprocessing, and feature engineering, this training data was used to explore our research questions.

Data

Each session in the dataset contains on average 15 tracks, and for each track, there are variables indicating how long a song was listened to before being skipped. However, we chose to predict on a Boolean (a binary variable, having two possible values called "true" and "false") "not_skipped" feature for the last track. Metadata features were stripped except for 4 generalized features, which are month, hour, day of the week, and if the user has Spotify premium. These data provide information about factors that have been shown to impact user behavior. For example, users have been shown to skip less often at times of the day when they are not paying attention to the music, such as when they are asleep or at work (Lamere, 2014). Users without Spotify Premium are also limited in skips, so this could cause them to skip much less. Additionally, each track in a session contained a track ID, which was used to obtain 21 track features and an 8-dimensional acoustic vector. These 29 numerical features represent the acoustic structure of the track and contain values such as tempo, energy, flatness, loudness, acousticness, and key.

Feature Engineering

The primary hypothesis in this project is that user behavior exhibits a degree of consistency within a session. Based on this premise, it is inferred that listeners are likely to skip songs that are similar to those they have previously skipped, while also being inclined to listen to songs that are similar to those they have already listened to within the same session. To test this hypothesis, two categories of additional features were created regarding the "closeness" of the last song to the previous tracks in a session and contextual features from the session.

Song Similarity Metrics

To evaluate the similarity of songs within a session, we represent the 21 track features in combination with the supplied 8-dimensional acoustic vector as a 29-dimensional vector, which we compare using linear algebra. Because we were unsure which distance measures would be most effective, we calculated several different distance metrics as shown in Table 1, augmented our dataset, and examined the resulting random forests to learn more about the behavior of these metrics.

The first distance calculation used was Euclidean distance, which is defined as the square root of the sum of the squared differences between the elements of two vectors (Weisstein, n.d.), a common metric that measures the straight-line distance between two points in N-dimensional space. Manhattan distance is another metric that measures the distance between two vectors. It is the sum of the absolute differences between the elements of both vectors (Barile & Weisstein, n.d.). Instead of a straight line, this calculation gives the distance by simulating movement along a grid, only being able to move horizontally or vertically. We also utilized the angle between two vectors, calculated using the cosine of the angle between the vectors, which is the dot product of the vectors divided by the product of their magnitudes (Helmenstine, 2020). The smaller the angle is, the more similar the vectors.

Table 1: Distance Metrics

Name	Description
AvPlay	Distance between the final song and the non-skipped average vector, calculated with Euclidean, Manhattan, or Angle
AvSkip	Distance between the final song and the skipped average vector, calculated with Euclidean, Manhattan, or Angle
LastPlay	Distance between the final song and the last non-skipped song, calculated with Euclidean, Manhattan, or Angle
LastSkip	Distance between the final song and the last skipped song, calculated with Euclidean, Manhattan, or Angle
neighborSkipped	A boolean indicating whether the final song's nearest neighbor was skipped

When examining the importance of these features in isolation, the Manhattan metrics ranked the highest relative to the Euclidean and Angle metrics. This aligns with the findings of Aggarwal et al. (2001) who demonstrate that in high dimensions, Manhattan is preferable compared to Euclidean. Given this analysis, we utilized the Manhattan metrics manAvPlay, manAvSkip, manLastPlay, manLastSkip, and neighborSkipped, in our final dataset as Table 1 indicates.

Contextual Features

We also calculated two contextual features to include in the dataset. These features were intended to provide additional information beyond the distance metrics used to compare the songs within the session. These features are shown in Table 2.

Table 2: The two contextual features that were added and their descriptions

Name	Description
percent_skipped	The percentage of songs in the session that the user has skipped
prevSongPlayed	If the user skipped or played the previous song

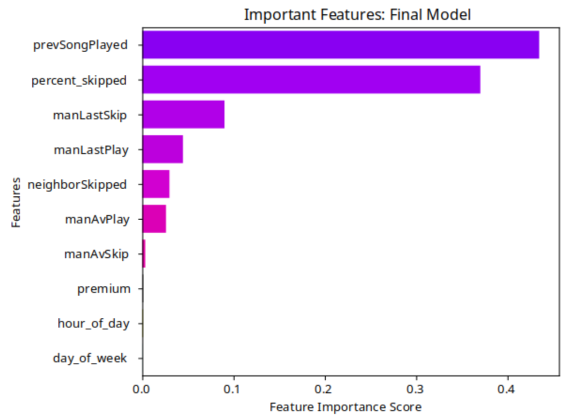
Results

After preprocessing and feature engineering was complete, each row of our dataset (representing a session) contained 5 metric features, 2 contextual features, 4 metadata features, 21 original track features, and an 8-dimensional acoustic vector totaling 40 features and the target value of whether the final song was skipped. We then split the 200,000 sessions into a training set containing 70% of the sessions and a test set containing 30% of the sessions.

The random forest models were implemented in Python and utilized the random forest classifier from the sklearn library. The final ensemble was made up of 100 trees, each with a maximum depth of 6. To understand the role of different features and feature categories in the model's success, several random forests were created, each using different sets of features. The results of these models are summarized for comparison in Table 3, along with the results of our final model. It appears that our newly engineered features perform much better than the original Spotify data. Our final model was 78.23% accurate and was trained on all the features except the original track features of the final song from Spotify, which surprisingly lowered the performance when included.

Table 3: Random forest results using different groups of features.

Source	Categories	Accuracy
Spotify	Track Features	53.67%
	Metadata	59.55%
	Track Features & Metadata	59.81%
Engineered	Metrics	67.57%
	Contextual Data	76.01%
	Metrics & Contextual Data	78.26%
All Data		77.69%
Final Model		78.23%



Feature Importance

One of the goals of this research was to discover which features in the data were the most predictive. When using a random forest, it is possible to calculate which features contributed most to the final predictions. For this, Mean Decrease Impurity (MDI) was used to calculate feature importance. MDI measures the average decrease in the impurity of nodes when splitting the data on a particular feature. Nodes are impure when they contain data from different classes. If a feature has a high MDI score, it effectively splits data into more homogeneous groups, making the most progress toward a prediction. Using the Python libraries matplotlib and seaborn, visualizations of these scores were generated.

As shown in Figure 1, our engineered features greatly outperform the original metadata given by Spotify. Of our engineered features, the contextual data is extremely predictive.

Focusing only on the metrics, manLastSkip and manLastPlay both rank higher than the distances between the average vectors. From this result, we can assume that the type of song a user skipped or played immediately before is more relevant when predicting the type of songs they have skipped or played on average throughout their session. However, it is possible that because the average vectors represent every song that has been skipped or played in a session, as a user's session grows, these vectors will become more obscure and less relevant.

References

Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In J. Van den Bussche & V. Vianu (Eds.), *Database theory --- ICDT 2001* (pp. 420–434). Springer Berlin Heidelberg.

Barile, M., & Weisstein, E. W. (n.d.). *Taxicab metric*. Wolfram MathWorld. https://mathworld.wolfram.com/TaxicabMetric.html.

Brost, B., Mehrotra, R., & Jehan, T. (2019). The music streaming sessions dataset. *Proceedings of the 2019 Web Conference*. https://arxiv.org/pdf/1901.09851.pdf

Helmenstine, A. M. (2020, August). *Angle between two vectors and vector scalar product*. ThoughtCo. https://www.thoughtco.com/angle-between-to-vectors-problem-609594

Lamere, P. (2014, May). *The skip*. Music Machinery. https://musicmachinery.com/2014/05/02/the-skip/

Weisstein, E. W. (n.d.). *Euclidean metric*. Wolfram MathWorld. https://mathworld.wolfram.com/EuclideanMetric.html

Recommended Citation
Noggle, S. (2023). Machine learning techniques to improve users' music listening. *Made in Millersville Journal*, 2023. Retrieved from https://www.mimjournal.com/