A Neural Network Approach To Predicting Song Skip Behavior

Aditi Patil
Columbia University
aap2205@columbia.edu

Sophie Johnson
Columbia University
smj2173@barnard.edu

Abstract

The Spotify streaming service hosts upwards of 400 million unique users each fiscal quarter, providing a platform for individuals to engage with millions of songs and curated content seamlessly. The 2021 Spotify Sequential Skip Prediction Challenge arose out of a need to understand the quality of Spotify's personalized song recommendation system, and challenges participants to predict if a track in a user listening session will be skipped or not. Based on prior work, we hypothesize that metadata regarding user behavior and acoustic features are integral in understanding song skip behavior, and aim to use Long-Short Term Memory (LSTM) and Gradient Boosting Tree models in order to properly predict whether a user will skip a given song during their listening session. In this paper, our approach to predicting user song skip behavior using a Random Forest Classifier (RFC) baseline reaches an accuracy level of 65% and our Gradient Boosting Trees reach a performance of 66%. By the end of our project, we will have also implemented an LSTM and reported its performance levels, which hopefully will be better than the baseline performance.

1 Introduction

The streaming service Spotify, offering users access to a wide selection of music in digital form, has grown immensely in popularity ever since its inception in April 2006. The company's mission statement writes that "by giving a million creative artists the opportunity to live off their art and billions of fans the opportunity to enjoy and become inspired by it," their service can "unlock the potential of human creativity" (Spotify). Part of this lofty goal of unlocking the potential of human creativity is seen through how central the user experience is to the streaming platform's mobile application. The streaming service provides a host of personalized algorithmically-generated playlists refreshed with new song selections every week, and opportunities for users to 'blend' their music preferences with other users and share playlists. These are but a few of the features Spotify promotes as part of this ethos of creating an experience for the user that inspires creativity and engagement.

Despite the many features and offerings of Spotify's streaming service, instances of users skipping songs raise questions about how this "implicit feedback signal" reflects user satisfaction and future engagement with the streaming service (Meggetto et al., 2021). In theory, Spotify's algorithms and song recommendations should be so curated to users' personal tastes that they don't need to skip songs. For the purposes of our project, we intend to examine the specifics of the user experience on the mobile Spotify application from publicly available datasets as related to instances of users skipping songs. Our central research question is how can one computationally predict user song skip behavior? We will first examine past approaches to the Spotify Sequential Skip Prediction Challenge to get an idea of what dataset features are most influential in predicting skip behavior. We will then perform statistical analytics assessing correlation between user skips and listening session data and construct a model incorporating these extracted dataset features. Our research question will thus enable us to understand how well our model can predict user skip behavior as well as how Spotify can use skip prediction to improve their song recommendation systems and the overall user experience. We hypothesize that metadata regarding user behavior and acoustic features will help us better understand song skip behavior, and aim to use Long-Short Term Memory (LSTM) and Gradient Boosting Tree models for our predictions.

2 Related Work

In their multi-RNN approach, Hansen et. al. grouped the features they incorporated into their model into three categories: "meta information associated with the whole session," "the sequence of playback tracks for the first half of the listening session," and the track-id and position in the session features of tracks in the second half of the listening session (Hansen et al., 2019). Features falling under the first category included whether the user is a premium user, length of session (number of tracks), and day of the week. The second category–sequence of playback tracks in the first half of the session–used all features from the data set relating to the first-half of the session (Hansen et al., 2019).

Another approach out of Seoul National University constructed metric learning and sequence learning models that structured comparison based on input of acoustic features of the song tracks (Chang et al., 2019). The Spotify dataset includes 16 metrics relating to acoustics: acousticness, beat strength, bounciness, danceability, energy, flatness, instrumentalness, liveness, loudness, mean dynamic range, mechanism, organism, popularity, tempo, speechiness, and valence (Brost et al., 2019). Their findings revealed that the "sequence learning-based approaches outperformed the metric learning-based approaches by at least 5.9 percent," and a subsequent model trained using both acoustic features and user-log features outperformed this model by 21.1 percent. The amplified accuracy that came from incorporating user-log features into the model that before primarily relied on acoustic features led Chang, Lee, and Lee to the conclusion that user-logs "contain useful information for sequential skip predictions " (Chang et al., 2019). This further corroborates that including more information and features can improve the prediction accuracy.

Ferraro, Bogdanov, and Serra reached a similar conclusion in their approach that earned them a 14th most accurate model in the open Spotify challenge and a 4th most creative approach award (Ferraro et al., 2019). Ferraro, Bogdanov, and Serra trained boosting trees using the 16 acoustic features from the dataset, like Chang, Lee and Lee, but combined them with acoustic features extracted from Essentia–an "open-source library for audio analysis for music information between tracks and playlists" (Ferraro et al., 2019). They also incorporated variables such as whether a user is a premium (non-premium users have a limited number of skips per hour), the time of day the user was listening to a given track, and the ratio of skipped track with respect to the skip-1, skip-2 variables (Ferraro et al., 2019). The study boasted higher accuracy when the model included the external acoustical analyses from Essentia. Therefore like the findings of (Chang et al., 2019), incorporating more features into the model is expected to improve the accuracy of the system.

It is important to note as well that the winning model of the Spotify Sequential Skip Challenge "used a model based on boosting trees to improve the recommendation based on some acoustic features and other features extracted from the playlist and the tracks" (Ferraro et al., 2019). Though the three cited approaches to the challenge vary in terms of implementation and features extracted, each study concludes that a combination of acoustic features and user-log data (what Hansen et. al. refer to as "meta-information") produces the most accurate results (Hansen et al., 2019).

3 Data

The dataset that we collected information from regarding Spotify skip behavior is from the open-sourced AICrowd's Sequential Spotify Skip Prediction Challenge (Brost et al., 2019). The goal of this challenge was to predict whether a given song would be skipped by a particular user during the second half of their listening session. The dataset contains 130 million unique Spotify listening sessions, with each listening session defined by 21 characteristics, including a session ID, the sequence of tracks played during the session, the time of day of the session, etc. The dataset contains 50704 distinct tracks that users heard during their listening sessions. Each of these tracks is defined by 29 features, including a track ID, track duration, track popularity estimate, track beat strength, etc. With over 2500 participants competing in the challenge.

3.1 Preprocessing Data

When first setting up our coding environment, we initially attempted to download the entire training set locally and then deploy it onto a personal GitHub repository. We quickly realized this was not feasible due to the 56GB size of the training set and hence chose to work with a split version of the

set totalling around 5.63GB, or roughly 10% or 13 million of the 130 million listening sessions associated with user interactions in the dataset. This split training set was supplied by the AIcrowd challenge, so we did not manually have to split it from the original 56GB training set. While we did have the intention of using the whole dataset due to its larger sampling size, we were materially constrained by the amount of storage and space available on our laptops, our primary material resource for this project. Had we had a computer or device that could accommodate the large size of our dataset we would have moved forward with analyzing the entire dataset.

To preprocess the data, we downloaded the training set and untarred the .tar.gz file in Google Colab. We then were able to process each .csv file inside the training set–66 in total–and convert them to panda Dataframes for our analyses. Given that we would be correlating the data using the Pearson correlation coefficient which requires numeric variables, we converted variables in the training set that were of type Boolean hist_user_behavior_is_shuffle, skip_1, skip_2, skip_3 to 0s and 1s, 0 corresponding to False and 1 corresponding to True.

3.2 Correlations

To assess which variables we wanted to include in our model, we performed correlations between the variables in the training dataset and the Boolean skip_1, skip_2, and skip_3 variables. As a reminder, the skip_1 variable captures if the track was only played very briefly, the skip_2 variable captures if the track was played briefly, and skip_3 indicates most of the track was played.

We chose to use Pearson's Correlation Coefficient to assess the statistical linear correlation between skip_1, skip_2, skip_3 and the training set variables. This type of correlation is defined as long as there is some value, either binary or non-binary in nature, for each of the two variables to be correlated. As an example, if correlating session_length and skip_1, one can assess the strength of the linear relationship between skips very quickly made at the beginning of a track and how long the user's complete listening session was. The binary nature of all skip values hence limits the resulting skip_1, skip_2, and skip_3 results to the binary connotation of if it was or was not skipped.

Given that within the training set we had 66 different .csv files each containing their own data for all training set variables, we calculated the Pearson correlation coefficient between each set of variables (skip_1, skip_2, skip_3 and some variable x) for each .csv file and then averaged the Pearson coefficient across these .csv files.

A Pearson correlation coefficient as well can only be deemed statistically significant, regardless of its magnitude between -1 and 1, if its corresponding p-value is less than 0.05. In turn, we calculated the average p-value corresponding to the average Pearson coefficient for each variable pair. We found that every single p-value hovered around zero per the scipy.pearsonr method, meaning all Pearson coefficients can be interpreted as statistically significant.

Since the data showed us that all variable pairs, regardless of the strength of their linear relationship, can be interpreted as statistically significant according to their p-values, we then had to turn to the magnitude of the Pearson correlation coefficients to think about which variables we wanted to include in our model. We decided on an

	context_type	session_length	hist_user_behavior_n_seekfwd	hist_user_behavior_n_seekback	hist_user_behavior_is_shuffle
skip_1	0.019	0.097	-0.048	-0.054	0.059
skip_2	0.002	0.086	0.013	-0.037	0.054
skip_3	0.001	0.070	0.065	-0.017	0.039

Figure 1

	context_switch	no_pause_before_play	premium	session_position	long_pause_before_play	hour_of_day
skip_1	-0.088	0.171	-0.008	0.060	-0.145	0.023
skip_2	-0.067	0.092	-0.001	0.044	-0.060	0.022
skip_3	-0.039	-0.075	0.009	0.007	-0.060	0.025

Figure 2

arbitrary threshold of 0.05, since most correlations hover below or above that. Figures 1 and 2 depict the mean Pearson Correlation Coefficient values respective to skip_1, skip_2, skip_3 and the training set variables. Table items highlighted in blue represent values where the Pearson Correlation coefficent is above 0.05 with corresponding p-values less than 0.05.

We were also interested in looking at how different context types of tracks affected skip behavior. To explore this a bit further, we calculated the frequency of each context_type respective to when skip_1, skip_2 and skip_3 were 'True' or equal to 1. As you can see in Figures 3, 4, and 5, user skips are more frequent with the context

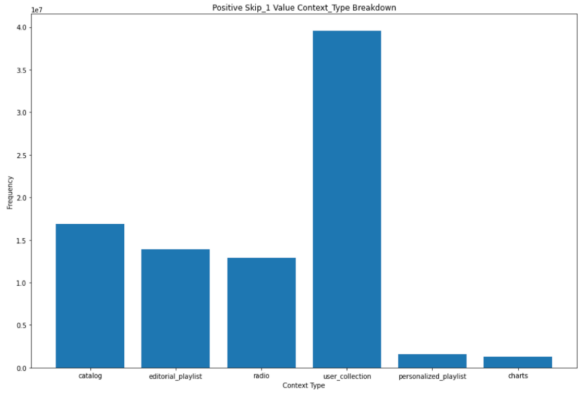

Figure 3: Frequency of context_type respective to skip_1

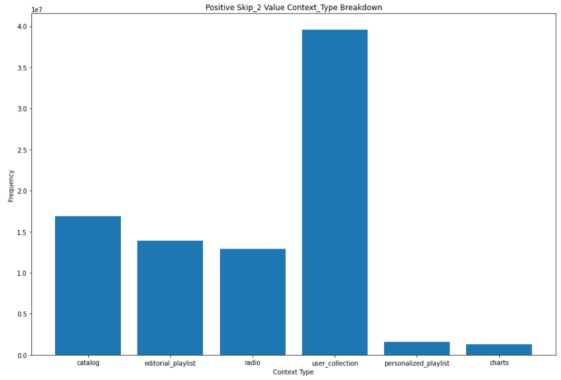

Figure 4: Frequency of context_type respective to skip_2

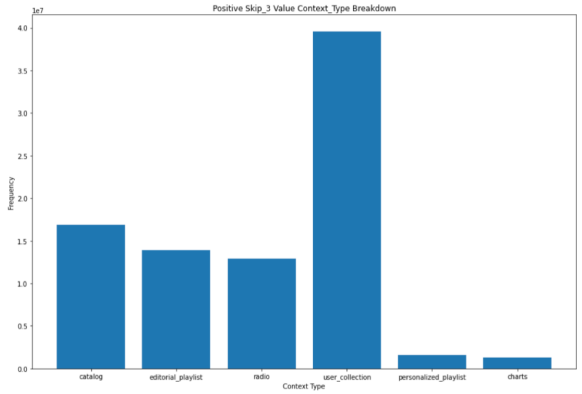

Figure 5: Frequency of context_type respective to skip_3

type "user_collection" across skip_1, skip_2, and skip_3. The wide variance in the frequency of skips respective to context_type suggests to us that even though the context_type Pearson correlation coefficient is relatively weak, context_type should be included in the model.

4 Methods

Next, we worked on creating our baseline model that would be evaluated against a test set. We used a shortened version of the training data set, and conducted a training/test split of 70%/30% on this sample. That way, we were able to still use a large portion of the data without running into errors regarding training and test size. To implement the Random Forest classifier, first, we had to understand how the Random Forest algorithm works. It can be described in four steps: 1) From a dataset, select random samples of data, 2) Construct a decision tree for every sample and receive a resulting prediction from each tree. 3) Conduct a vote for each predicted result, and 4) For the final prediction, select the prediction result with the most votes. Using the Pearson coefficients and p-values of significance, the statistically significant features for the model were extracted for $skip\_1$, $skip\_2$ and $skip\_3$ respectively. All of these variables were turned into float values, and normalized so that they could be fed to the Random Forest Classifier. However, one drawback is that we had to remove context_type because it had a string value, which was incompatible for the classifier and in the time we have had until now, we have not figured out a mapping for the string values to numerical values. We hope to remedy this before the final deadline. In order to predict whether a song was skipped or not, the three categories of skipped were aggregated into one. The model was then tasked with predicting whether a given session was skipped or $not\_skipped$. Our justification of this is that a later skip is dependent on the earlier skip's Boolean value. For example, if $skip\_1$ is true, it is guaranteed that $skip\_2$ and $skip\_3$ are true as well because the user never reached that part of the song. The same is true for $skip\_3$ depending on $skip\_2$'s Boolean value. Thus, it makes sense to aggregate the columns if they are dependent on one another in order to get overall skip behavior. The variables being fed to the model are as follows, where X represents the independent variables and Y represents the dependent variables.

X: context_type, session_length, hist_user_behavior_n_seekback, session_position, context_switch, hist_user_behavior_n_seekfwd, no_pause_before_play, long_pause_before_play, hist_user_behavior_is_shuffle.

Y: skip, not_skipped

We were able to create a baseline model and evaluate accuracy as a metric for its performance. One thing we observed about this baseline is that training was very slow, because random forest has to calculate every decision tree before outputting a result. Thus, it's not the most feasible model on large datasets.

Next, we worked on implementing the Gradient Boosting Tree (GBT) model. Gradient Boosting Trees is similar to Random Forest, except that each successive predictor tries to improve on its predecessor by reducing the errors. This is conceptualized by the idea that a weak hypothesis can be tweaked in order to bring stronger hypotheses/learning algorithms. The goal of GBT is to minimize the loss, which is the difference between the real class of the training example and the predicted class. As a result, Gradient Boosting classifiers highly rely on loss functions, which is why in our implementation we chose to tune the model in order to get better performance. We set learning rate to 0.1, the maximum depth of the individual regression estimators to 5, and used a logarithmic loss function. (Xuan et al., 2019)

5 Results

The Random Forest Classifier performed at an accuracy level of 65%. This is a good baseline performance, and we hope that our model will improve upon this accuracy level. We also experimented with the SciKit's built in feature importance tool on the Random Forest Classifier. This allowed us to confirm how important each feature selected is in predicting the skip or not skipped outcome. The graph is plotted in Figure 6. From this plot, we can see that the features related to the user's behavior with seeking forward and a long pause before playing were the top two most important in predicting skip or not skip. This is very helpful moving forward when we create our LSTM and Gradient Boosting Tree models because that can help us with the model's weights while training. The Gradient Boosting model we implemented performed at an accuracy level of 66%, however its breakdown among precision, recall, and F1-score can be seen in Table 1. Class 0 represents "Not Skipped" and Class 1 represents "Skipped." From these results, we can see that the GBT is much better at predicting if a song has not been skipped than predicting whether a song was skipped. We hope to improve the performance on this GBT model before the final draft is due as this is one of the models in which we want to improve baseline performance considerably.

	precision	recall	f1-score
0 (skipped)	0.54	0.11	0.18
1 (not skipped)	0.66	0.95	0.78

Table 1: Precision/Recall/F1 score of class prediction for the Gradient Boosting Tree classifier.

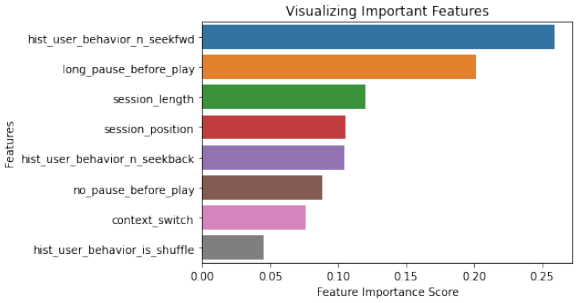

Figure 6: The top features used of all selected features in the Random Forest Classifier.

6 Next Steps

We are in the process of implementing our LSTM model and fine-tuning our Gradient Boosting algorithm. By our final project deadline, we will have completed both and done some error analysis to see how they perform in comparison to one another and our baseline, Random Forest Classifier. From that work, we hope to achieve high accuracy levels on song skip prediction as well as be able to formalize a pattern as to which features are most correlated with song skip behavior.

References

Brian Brost, Rishabh Mehrotra, and Tristan Jehan. 2019. Spotify sequential skip prediction challenge. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM '19)*. Association for Computing Machinery.

Sungkyun Chang, Seungjin Lee, and Kyogu Lee. 2019. Sequential skip prediction with few-shot in streamed music contents. arXiv:1901.08203.

Andres Ferraro, Dmitry Bogdanov, and Xavier Serra. 2019. Skip prediction using boosting trees based on acoustic features of tracks in session. arXiv:1903.11833.

Christian Hansen, Casper Hansen, Stephen Alstrup, Jakob Grue Simonsen, and Christina Lioma. 2019.

Modeling sequential music track skips using a multi-rnn approach. arXiv:1903.08408.

Francesco Meggetto, Crawford Revie, John Levine, and Yashar Moshfeghi. 2021. On skipping behaviour types in music streaming sessions. In *Proceedings of the 30th ACM International Conference on Information Knowledge Management (CIKM '21)*.

Spotify. Listening is everything. *Spotify*.

Ping Xuan, Chang Sun, Tiangang Zhang, Yilin Ye, Tonghui Shen, and Yihua Dong. 2019. Gradient boosting decision tree-based method for predicting interactions between target genes and drugs. *Frontiers in Genetics*, 10.