# A Neural Network Approach To Predicting Spotify 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. 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 accurately 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 reached an accuracy level of 65% and our Gradient Boosting Trees reach a performance of 66%. An LSTM will also be implemented and its performance levels 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 goal 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). Spotify's algorithms and song recommendations should be curated to users' personal tastes that they don't need to skip songs. For the current purposes, we intend to examine the specifics of the user experience on the mobile Spotify application from publicly available datasets 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 understand which dataset features are most influential in predicting skip behavior. Running statistical analytics assessing correlations between user skips and listening session data will assist in constructing a model incorporating these extracted dataset features. The 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

To contextualize the approach to predicting user skip behavior and understand which features to include in the model, we first looked at song skip behavior at large. Before examining past approaches to the Spotify Sequential Skip Challenge that our research directly responds to, will first examine these patterns in skip behavior.

## 2.1 Song Skip Behavior

Many streaming services and psychologists alike have dedicated research to exploring the question of how much time is necessary to make "accurate aesthetic judgements" (Montecchio et al., 2020). How users respond to songs they listen to and whether they choose to skip them falls within this accurate aesthetic judgement idea. After all, an estimated quarter of all streamed songs are skipped within the first five seconds, with only half of all streamed songs being played in full (Montecchio et al., 2020).

In a study conducted by Montecchio et al., 2020, they created a distribution of the likelihood of skipping relative to time using a dataset from Spotify composed of 100 popular songs released in April to May 2018 that were skipped by Spotify users over 3 billion times (Montecchio et al., 2020). As seen in Figure 1, the skipping likelihood rapidly declines the longer a user listens to a given song. In Figure 2 which narrows in on the skip likelihood within the first thirty seconds of a song, it can be seen that the peak skip likelihood or time of "aesthetic judgement" occurs at roughly 2 seconds. The skip likelihood is very low in the first 1-2 seconds, Montecchio et al., 2020 estimate, because the user has not yet identified what song they are listening to enough to make an accurate aesthetic judgement about it. While this study in no way attempts to predict skip behavior—they instead examine when it is most likely based on publicly available datasets—their work establishes that regardless of certain factors like the musical structure of the song or how long the user has been listening to music, user skips at large are more common in the first few seconds of a song.

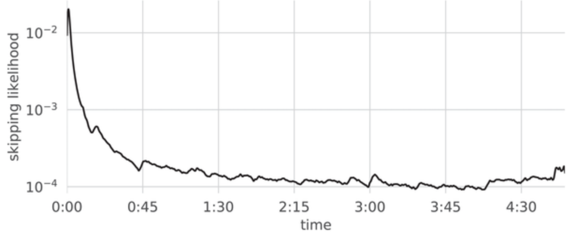


Figure 1: Skip profile of an individual song

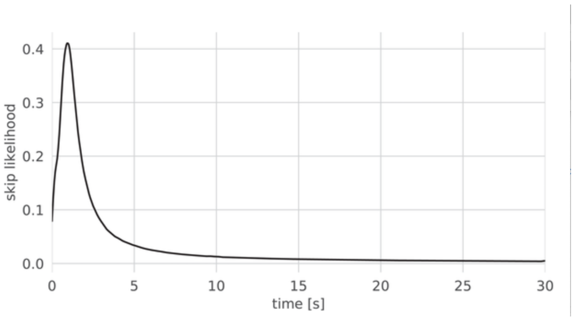


Figure 2: Skipping likelihood at the beginning of a song

## 2.2 Past Approaches to the Sequential Skip Challenge

The three approaches outlined in this section rely on the AICrowd publlic dataset collected by Brost. et al. (2019).

Using a multi-RNN approach, (Hansen et al., 2019) grouped the features they incorporated into their model into three categories: meta information associated with the user listening session, the sequence of playback tracks for the first half of that session, and the track-id and position in the session features of tracks in the second half of the listening session. As defined by the AICrowd challenge, a listening session is 20 songs long, meaning the first and second halves of a session are 10 songs each. 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. Unlike most approaches to the Sequential Skip Challenge, (Hansen et al., 2019) incorporated all dataset features into the model and involved very little data pre-processing. Their approach made use of two distinct stacked recurrent neural networks, one network directed to "encoding the first half of the session and the other network focus[ing] on utilizing the encoding to make sequential skip predictions" (Hansen et al., 2019). Their submission was the second highest ranked submission in the AI Crowd competition.

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 models outperformed their metric learning counterparts 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 (Chang et al., 2019). The amplified accuracy that came from incorporating user-log features into the model that before primarily relied on acoustic features led to the conclusion that user-logs are useful in predicting skips (Chang et al., 2019). This further corroborates that including more information and features can improve the prediction accuracy.

Ferraro, Bogdanov, and Serra (2019) 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. They 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 containing audio analyses of music tracks. 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. (2019) refer to as meta-information–produces the most accurate results.

## 3 Data

The dataset used in the current research 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

We chose to work with a split version of the training set representing 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. While we did have the intention of using the whole dataset due to its larger sampling size, we were limited by space constraints. We were able to process each data file inside the split training set–66 in total–and convert them to panda Dataframes for our analyses.

## 4 Methods

### 4.1 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. The significance of the skip_1, skip_2, and skip_3 variables are as follows:

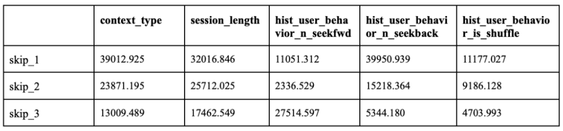
**skip_1**: the track was only played very briefly
**skip_2**: track was played briefly
**skip_3**: most of the track was played

Unfortunately the dataset description of features supplied by AICrowd did not specify the quantitative significance or number of seconds in a song played corresponding of ”very briefly,” ”briefly,” and ”most of track was played,” so the skip variables are inherently ambiguous. We chose to keep the skip_1, skip_2, skip_3 variable names the same for this very reason, as there is no way to specify the exact connotation of ”very briefly,” ”briefly” and ”most of track played.”

We chose to use the Chi-Square statistical test to determine correlation between categorical variables between skip_1, skip_2, skip_3 and the training set variables. For each chi-square test performed between two variables, the null hypothesis is that there is no relationship between the variables; they are independent. As an example, if correlating session_length and skip_1, one could assess if skips made in the very first part of a song dependent on the total length of a user’s listening session.

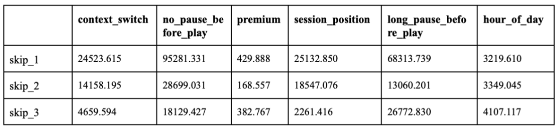
Given that within the training set folder, there are 66 different .csv files each containing their own data for all training set variables, we calculated the Chi-Square test statistic and p-value of the test for every set of variables (skip_1, skip_2, skip_3 and some variable x) for each .csv file and then averaged the test statistics across these 66 .csv files.

Results show every single p-value was less than 0.05, the highest of all p-values being 0.019, meaning all Chi-Square test statistics can be interpreted as signifying dependence between the variables.



| | context_type | session_length | hist_user_behavior_n_seekfwd | hist_user_behavior_n_seekback | hist_user_behavior_is_shuffle |
|---|---|---|---|---|---|
| skip_1 | 39012.925 | 32016.846 | 11051.312 | 39950.939 | 11177.027 |
| skip_2 | 23871.195 | 25712.025 | 2336.529 | 15218.364 | 9186.128 |
| skip_3 | 13009.489 | 17462.549 | 27514.597 | 5344.180 | 4703.993 |

Figure 3: Mean Chi-Square Statistical Values



| | context_switch | no_pause_before_play | premium | session_position | long_pause_before_play | hour_of_day |
|---|---|---|---|---|---|---|
| skip_1 | 24523.615 | 95281.331 | 429.888 | 25132.850 | 68313.739 | 3219.610 |
| skip_2 | 14158.195 | 28699.031 | 168.557 | 18547.076 | 13060.201 | 3349.045 |
| skip_3 | 4659.594 | 18129.427 | 382.767 | 2261.416 | 26772.830 | 4107.117 |

Figure 4: Mean Chi-Square Statistical Values

Since the data showed us that all variable pairs, regardless of the strength of their dependence, can be interpreted as statistically significant according to their p-values, we then had to turn to the magnitude of the test statistics to think about which variables we wanted to include in our model.

Figures 3 and 4 depict the mean Chi-Square test statistic values respective to skip_1, skip_2, skip_3 and the training set variables. As one can see in the figures, the magnitude of Chi-Square test statistic was highest respective to skip_1 with the hist_user_behavior_n_seekback, long_pause_before_play, and no_pause_before_play variables. The test statistic magnitude was highest respective to skip_2 with the no_pause_before_play, context_type, and session_length variables. Lastly, the magnitude of the test statistic was highest with respect to skip_3 with the hist_user_behavior_n_seekfwd, no_pause_before_play, and session_length variables.

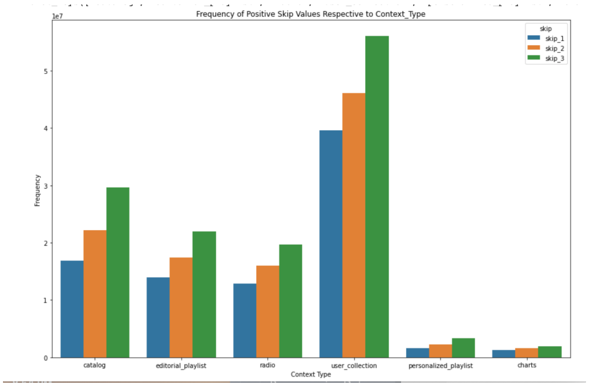


Figure 5: Frequency of context_type respective to skip_1

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’. As you can see in Figure 3, user skips are more frequent with the context type “user_collection” across skip_1, skip_2, and skip_3. While the user_collection is the most popular context_type regardless of whether songs were skipped or not, it is important nontheless to recognize that users more frequently skip songs in a user collection. The six types of music context types listed in Figure 3 signify:
**catalog**: songs pertaining to a a specific artist or album that is finite.
**editorial playlist**: playlist created by Spotify for its users, i.e. ”90’s Workout”
**radio**: Spotify Radio function specifies custom playlist for a given Artist or track, i.e. Lana Del Rey Radio.
**user collection**: playlist created by another Spotify user.
**personalized playlist**: playlist created by that user.
**charts**: playlist created by Spotify containing most listened songs from a given location, i.e. ”US Top 50 Songs.”

### 4.2 Random Forest Classifier

Next, we worked on creating our baseline model that would be evaluated against a test set. We obtained 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, it is important 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.

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 features being fed to the model are as follows:

**context_type:** What type of context the playback occurred within.

**context_switch:** Boolean indicating if the user changed context between the previous row and the current row. For example, this would happen if the user switched from one playlist to another.

**session_position:** Position of the listened track within a session, ranges from 1 to 20.

**session_length:** The number of tracks listened to in the session, ranges from 10 to 20.

**hist_user_behavior_n_seekback:** Number of times the user did a seek back within track.

**hist_user_behavior_n_seekfwd:** Number of times the user did a seek forward within track.

**no_pause_before_play:** Boolean indicating if there was no pause between playback of the previous track and this track.

**long_pause_before_play:** Boolean indicating if there was a long pause between playback of the previous track and this track.

**hist_user_behavior_is_shuffle:** Boolean indicating if the user encountered this track while shuffle mode was activated.

These are then evaluated in order to predict whether a song is skipped or not_skipped. A baseline model was created and accuracy was evaluated as a metric for its performance. One observation about this baseline is that training the model was very quick, because random forest's depth does not have to be too deep in order to make a correct prediction. Thus, it is a good strategy for large datasets if time is a constraint.

### 4.3 Gradient Boosting Tree

Next, the Gradient Boosting Tree (GBT) model was implemented. 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 the implementation the model is tuned in order to get better performance. The learning rate was set to 0.1, the maximum depth of the individual regression estimators to 5, and used a logarithmic loss function. (Xuan et al., 2019)

### 4.4 Long Term Short Memory (LSTM)

Finally, we created a model that uses deep learning recurrent neural networks in order to predict song skip behavior. LSTMs are excellent in sequence prediction, which is why their architecture can be leveraged for song skip prediction. In our architecture, we constructed a deep learning model that inputs the vectorized forms of the nine features we highlighted, and output whether a song will be skipped or not. This consists of an embedding layer for the inputs, a LSTM of 100 units, and a dense layer to output the skip prediction.

The LSTM was trained using the hyperparameters of a learning rate of 0.1, an Adam optimizer, and a batch size of 300 on 100 epochs.

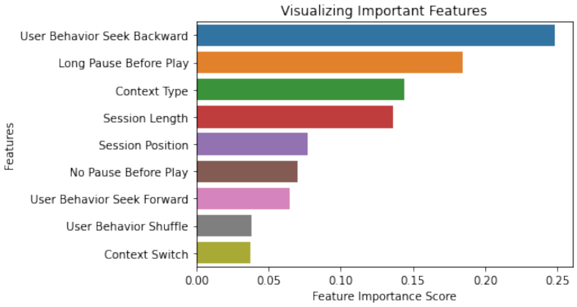


Figure 6: The top features ranked by importance of all selected features according to the Random Forest Classifier.

## 5 Results

The Random Forest Classifier performed at an accuracy level of 65%. This is a good baseline performance, and expect our model will improve upon this accuracy level. Experimenting with the SciKit's built in feature importance tool on the Random Forest Classifier 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, one 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 for the Random Forest Classifier. While this is not ground truth, this is still helpful when creating the LSTM and Gradient Boosting Tree models because that helps us tune the model's weights while training.

The Gradient Boosting model performed at an accuracy level of 66%, however its further breakdown of precision, recall, and F1-score can be seen in Table 1. Class 0 represents "Not Skipped" and Class 1 represents "Skipped." From these results, GBT is much better at predicting if a song has not been skipped than predicting whether a song was skipped. This could be due to the class imbalance in our dataset, where there were many more "not skipped" sessions compared to "skipped." Regardless, this method provides good results when compared to other scores from this competition, as the winners were able to achieve accuracy levels of 81% on first skip prediction and a mean average accuracy of 60.4%.

| | 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.

The LSTM further performed at an accuracy level of 64%. This was the same as Random Forest, and slightly worse than Gradient Boosting Trees. Figure 7 shows the validation and train loss graph for this model, depicting how loss decreases with each epoch. An explanation for this performance can be multiple reasons. Firstly, since we used a smaller dataset, this could have caused the LSTM to perform worse, while unaffecting the decision tree models.

Another consideration is that LSTM training time is long as it requires much time for the memory gates to input or forget information. Thus, the LSTM approach may not be feasible for large datasets with complex features.

## 6 Conclusion

## 7 Future Work

In this work, we looked at metadata and user behavior in order to predict song skips. However, the Spotify Challenge dataset also contains information regarding each track and its own properties, such as energy, acousticness, date it was released, etc. that has been shown to impact skip behavior and be a useful feature set for skip prediction. However, incorporating too many features could potentially lead to the model overfitting and therefore, the model should be carefully trained and finetuned. Additionally, given this challenge and the results from hundreds of submissions, it would be important to synthesize these findings into a business strategy by Spotify in order to better suggest songs to users.

## 8 Acknowledgements

We would like to thank our Professor and Research mentor Michelle Levine as well as the Columbia University Computer Science Department for enabling us to conduct this research.

## References

Brian Brost, Rishabh Mehrotra, and Tristan Jehan. 2019. Spotify sequential skip prediction challenge. In *Pro-*

*ceedings 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)*.

Nicola Montecchio, Pierre Roy, and Francois Pachet. 2020. The skipping behavior of users of music streaming services and its relation to musical structure. *PLoS ONE*, 15(9):e0239418.

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.