Spotify: Sequential Track Skip Prediction

A Mini Project Report

Submitted by

Karthik Rayan V
CB.SC.I5DAS18017

In partial fulfillment of the requirements for the award of the degree of

Integrated Master of Science
in
Data Science

DEPARTMENT OF MATHEMATICS
AMRITA SCHOOL OF ENGINEERING
AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112 (INDIA)

JUNE - 2022

Spotify: Sequential Track Skip Prediction

A Mini Project Report

Submitted by

Karthik Rayan V
CB.SC.I5DAS18017

In partial fulfillment of the requirements for the award of the degree of

Integrated Master of Science
in
Data Science

DEPARTMENT OF MATHEMATICS
AMRITA SCHOOL OF ENGINEERING
AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112 (INDIA)

JUNE - 2022

AMRITA SCHOOL OF ENGINEERING
AMRITA VISHWA VIDYAPEETHAM
COIMBATORE - 641 112

BONAFIDE CERTIFICATE

This is to certify that the mini project report entitled **"Spotify: Sequential Track Skip Prediction"** submitted by **Karthik Rayan V (Reg.no CB.SC.I5DAS18017)**
In partial fulfillment of the requirements for the degree of **Integrated Master of Science** in **Data Science** is a bonafide record of the work carried out at the Department of Mathematics, Amrita School of Engineering, Coimbatore.

Signature of Class Advisor

Signature of Chairperson

Signature of the Internal Examiner

AMRITA SCHOOL OF ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112

DEPARTMENT OF MATHEMATICS

DECLARATION

I, Mr. Karthik Rayan V (Register number: CB.SC.I5DAS18017), hereby declare that this mini project entitled **"Spotify: Sequential Track Skip Prediction"** is the record of original work done by me under the Department of Mathematics, Amrita School of Engineering, Coimbatore. To the best of my knowledge, this work has not formed the basis for awarding any degree/diploma/associateship/fellowship/or a similar award to any candidate in any University.

Place: Coimbatore

Signature of the Student

Date: 06-06-2022

COUNTERSIGNED

Dr. Prakash P

Class Advisor

Department of Mathematics

Acknowledgement

I am deeply grateful to my Class Advisor, Dr. Prakash P, Department of Mathematics, Amrita School of Engineering, for their encouragement, criticism, and faith. Their dedication and support have inspired and have left an indelible mark in my memory.

I express my sincere thanks to Dr. Ravichandran J, Chairperson, Department of Mathematics, Amrita School of Engineering, for his encouragement and providing necessary facilities in the Department.

I would like to thank the faculty members of the department of Mathematics for allowing me to work on this project.

I would like to thank Spotify for bringing out this problem and making this data available through AI Crowd Challenge.

I acknowledge my deep sense of gratitude and dedicate this work to my family members and friends who have always been a source of inspiration throughout my study. I am so grateful for the patience, love, and care they have shown during the period of my project work.

Karthik Rayan V

Contents

Abstract

In music streaming services, features like recommendation and shuffle play a vital role in increasing user experience on aspects like personalized playlists. Understanding a user's interaction within a single session will be beneficial in such cases to ensure better engagement. This project gives a study about the interaction of people with music, specifically with the skip button, and tries various statistical methods and machine learning models to predict whether a particular user will skip the next song given the interaction with the previous songs.

1. Introduction

Personalized music recommendation systems play a critical role in improving user experience in streaming services such as Spotify. While there is a large body of work on recommender systems, there is very little work or data, describing how users sequentially interact with the streamed content they are presented with.

Particularly when a user skips a track is an important implicit feedback signal. A user’s skip behavior is defined as the time interval between the start of the song and the time at which the user skips the song.
A user’s skip interaction forms a core aspect of the user experience as it gives more control to the user and helps engineers and researchers to better understand the user’s taste. So, the question that comes is whether we can predict whether a user will skip a song shortly after playtime which is an indication of non-interest when a playlist of songs is played sequentially based on historical play data of the user.

This will give us intuition about the user's music preference and help to design a “Shuffle Song” feature which is more focused on user experience

The data provided by Spotify through the AI Crowd organization challenge contains several music listening sessions and the objective is to infer which songs will be skipped in the second part of each session.
The number of dimensions in the dataset and the layout of sequential data makes this challenge both strenuous and novel.

2. Data

The challenge data consists of 160 million sessions with rich session and track metadata. There's also a sample of the same provided by Spotify reflecting the same characteristics consisting of around 10000 such sessions which are being used in this project.

The length of each session is up to 20 songs. For most sessions, a track start and end reasons, a premium user flag, and timestamps are also available. The dataset also includes track metadata like instrumentalness, release year, and track acoustic features vectors.

Skip events are also given as three types,

skip_1: “very briefly played”

skip_2: “briefly played”

skip_3: “most of the track was played”

These were binary encoded, if none of these fields is set it is assumed that the user listened to the whole track.

skip_2 serves as the ground truth label, which means that the song was only played very briefly or briefly.

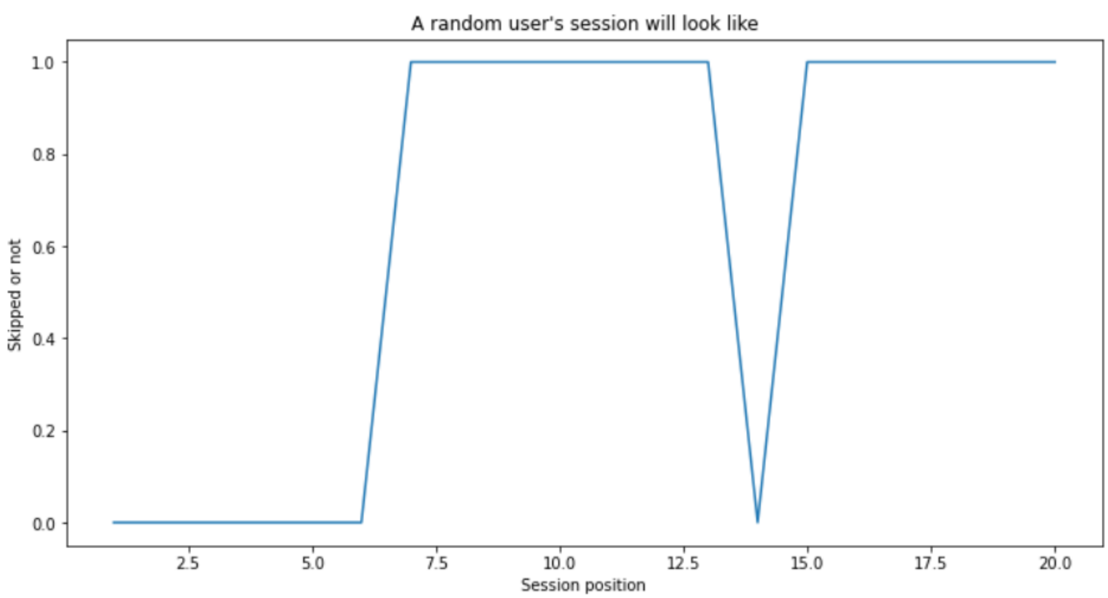


Fig: Highs in the figure represent the user skipping a particular track.

3. Feature Engineering

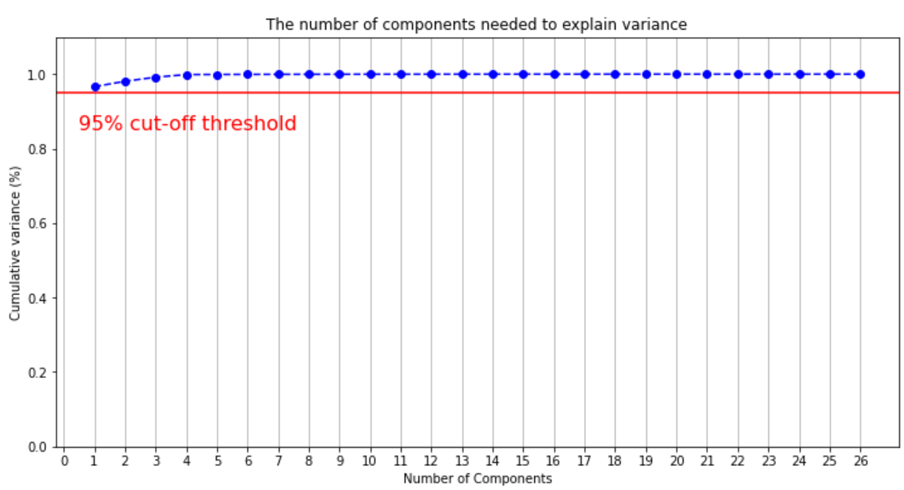
There are 20 features in the session dataset and 30 features in track metadata. Both the datasets were merged and categorical columns were encoded using label and one-hot encoder techniques.

3.1. Handling curse of dimensionality

Curse of dimensionality refers to various phenomena that arise when analyzing and organizing data in high-dimensional spaces. When we do categorical encoding and include the sequential data (3.2), the feature space would expand according to the tuning. Hence there is a need of shortening the dimensionality in this stage. For achieving this the 26 features relating to the musical and acoustic measures of the track are selected and PCA is performed.

Principal Component Analysis or PCA is a dimensionality reduction method that is often used to reduce the dimensionality of large datasets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

In this project, the 26 features were reduced to 2 sets of principal components capturing 98-99% variance. This can be seen in the scree plot given below.



3.2. Imposing sequential data

A very important aspect in data preprocessing that has been done to this data is imposing features of previous historical sessional song data as individual features of the row. That is the last n songs played before the concerned session position will be added to the current song as an individual row. Where this n can be considered as a hyper parameter.
For example : when n = 3. The last 3 songs of the user in the current session will be added to the current song.

4. Model training

The model training has been in mainly two forms

1. Without sequential information: Same as predicting the skip with just the information and interaction with the current song
2. With sequential information: Including the last ‘n’ songs interaction along with the current song.

4.1. The evaluation metric

Mean Average Accuracy(MAA) is used as the primary metric for the project. The average accuracy is defined as

$$AA = \sum_{i=1}^{T} \frac{A(i)L(i)}{T}$$

Where

- *T* is the number of tracks to be predicted for the given session
- *A(i)* is the accuracy at the position i of the sequence
- *L(i)* is the Boolean indicator for if the *i*’th prediction was correct

This metric gives more importance to the prediction of the immediate next track to be played.

4.2. Train-Test split

The train and test data cannot be random here as the sessions for each user will be varying from 16-20. Therefore a training size of 75% is used.

The 75% will contain the first part of the user's session and the remaining would be allotted as the test set. This split dynamically changes according to the varying session lengths.

4.3. Machine Learning model

As a primary step, the data was fit on a basic Logistic Regression model to form a baseline performance for the comparison. This results in an MAA of 71.2% and an individual skip accuracy of 84.54%.
Next, the modeling is done using Gradient Boosted Trees (XGB) with and without sequential info.

Gradient Boosted Trees - XGBoost Classifier

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. When a decision tree is a weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest. A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.
XGBoost is an implementation of gradient boosted decision trees designed for speed and performance. XGBoost is an extreme gradient boost algorithm. In this algorithm, decision trees are created in sequential form. Weights play an important role in XGBoost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results. The weight of variables predicted wrong by the tree is increased and these variables are then fed to the second decision tree. These individual classifiers/predictors then ensemble to give a strong and more precise model. It can work on regression, classification, ranking, and user-defined prediction problems.
XGBoost Classifier was trained with and without sequential information and the former gave an MAA of 77.12% and an individual skips accuracy of 87.8%

With sequential information includes a hyperparameter tuning for selecting the number of previous song interactions to be included.
N values of 5,10,15 were iteratively chosen and the model was fit.
Where N = 5 resulted in an optimal MAA value of 78.94% and an individual skips accuracy of 89.8%.

4.4. Deep Learning model

Because of the involvement of the sequential data, the performance of models like Recurrent Neural Networks needs to be tested on the data.
A **Recurrent Neural Network(RNN)** is a type of Neural Network where the output from the previous step is fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is the Hidden state, which remembers some information about a sequence.
Thus here when we use RNN on this data the hidden state would remember the pattern of the data and would be able to identify the sequences which would lead to skipping the current track.
The model follows a simple architecture to reduce the complexity of the given task. The inputs are passed to a simple RNN layer of 64 units, where sequential learning happens then it is passed on to a Dense layer of 128 neurons with a ReLU(Rectified Linear Units) activation, and finally the output layer with a single neuron with sigmoid activation function to provide the probability of this partial binary classification problem.
ReLU will give the input directly if it is positive and otherwise zero.
Sigmoid exists between 0-1, therefore it is used to map the probability of the target labels from the output of the neural networks.

5. Results

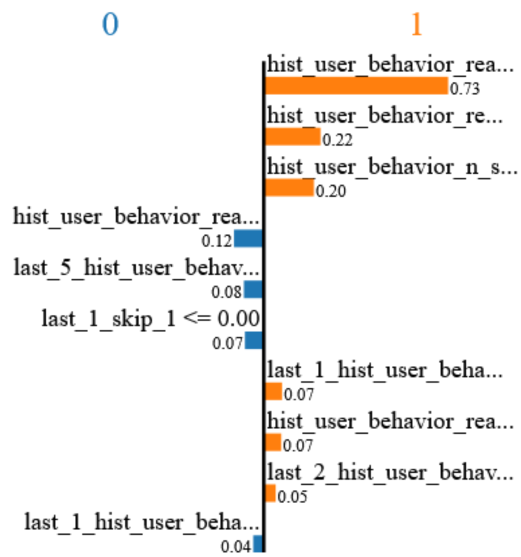
All the models used in the task have resulted in decent scores in the testing data. Even among them, we can see that the ones which use sequential features have a clear upper hand over the ones which do not consider previous track information. These results are summarised in the following table and both the metrics MAA and individual skip accuracy are compared.

Metrics	Logistic regression	XGBoost Classifier	XGB with sequential			RNN
			n = 5	n=10	n=15	
MAA	71.2%	77.12%	78.94%	77.8%	78.59%	76.32%
Ind. Acc.	84.54%	87.8%	89.08%	88.46%	88.78%	87.04%

6. Model interpretations

A brief analysis was done on how the features were used by the model using feature importance scores and LIME (Local Interpretable Model-agnostic Explanations). When looking at the interpretations,
The dataset features related to the user behavior such as hist_user_behavior_reason_end, hist_user_behavior_reason_start for the current song as well as the previous songs play a vital role in the prediction process.

In this test case of the user skipping the song, we can see how the model predicted this:



For values of these historical features, we can see the weightage given by the model, and the side/color signifies which class does the feature provide importance.

Feature	Value
hist_user_behavior_reason_end_trackdone	0.00
hist_user_behavior_reason_end_logout	0.00
hist_user_behavior_n_seekback	0.00
hist_user_behavior_reason_end_backbtn	0.00
last_5_hist_user_behavior_reason_start_trackerror	0.00
last_1_skip_1	0.00
last_1_hist_user_behavior_reason_end_trackdone	0.00
hist_user_behavior_reason_end_endplay	0.00
last_2_hist_user_behavior_reason_start_remote	0.00
last_1_hist_user_behavior_reason_start_playbtn	0.00

i.e, if the value of reason_start_end_trackdone is one, in this case, the model would likely classify this as not skipped because the user probably listened to the song full till the track end.
When can also see the reflection of these features when we look at some initial analysis of the data, and we could see it has an influence on the frequency distribution.

	skip_2
hist_user_behavior_reason_start	
appload	892
backbtn	8935
clickrow	6630
endplay	8
fwdbtn	61120
playbtn	75
remote	60
trackdone	9059
trackerror	45

Some specific reasons like the buttons, and track done has more skip rates. This is only for the current song, When features like these combine for some of the previously played songs when we include the historical data. The model could derive decisions mainly based on user behavior.

7. Conclusion

In this project the main focus was on understanding a user's interaction within a single session. Hence the imposing of the sequential interactions was critical to compare and analyze. The results from the models do show that the sequential information does result in an improvement in the performance of the models. The Gradient Boosted Trees outperform the basic RNN with the help of the imposed sequential information. And from the inference from the model interpretations, we can conclude that user behavior pattern feedback signals like a reason for ending the previous song, which button leads to the current song, the click row, etc play a vital role in predicting the skip interactions.

8. References

- *Brian Brost, Rishabh Mehrotra, Tristan Jehan,* The Music Streaming Sessions Dataset.

  https://www.researchgate.net/publication/330700807_The_Music_Streaming_Sessions_Dataset

- AI crowd: Spotify sequential skip challenge: https://www.aicrowd.com/challenges/spotify-sequential-skip-prediction-challenge-old

- Sequence learning: https://paperswithcode.com/paper/sequential-skip-prediction-with-few-shot-in

- The behavioral science behind Spotify wrapped's viral success: https://thedecisionlab.com/insights/consumer-insights/the-behavioral-science-behind-spotify-wrappeds-viral-success