Mood Prediction of a Spotify User based on The Music Streaming Sessions Dataset

Shanmuga Priya Ellappan*, Abhishek Bonageri† and Nikhil Sarika‡
Department of Computer Science, Illinois Institute of Technology
Chicago
Email: *sellappan@hawk.iit.edu, †abonageri@hawk.iit.edu, ‡nsarika@hawk.iit.edu

***Abstract*—Recommender systems play an essential role in music streaming services, prominently in the form of customized and personalized playlists. Exploring the user interactions within listening sessions can be beneficial to understanding the user preferences in the context of a single session. Also, the music providers are motivated to offer songs that users like in order to create a better user experience and increase the session duration. An important challenge to music streaming is determining what kind of music the user would like to hear. The skip button is a feature instance on these music services which empowers a user to skip a song that is out of his interest. Skipping behavior serves as a powerful signal about what the user does and does not like. For instance, in the afternoon, a user might be looking for classical study music, and thus skipping hip-hop. Later that evening, the user might skip classical for hip-hop. Being able to use skip behavior in the context of an entire listening session is key to recommending relevant content. Thus, this button plays a large role in the user experience and helps to identify what a user likes. In this project, we will build machine learning models that identifies if a user will skip a particular song or not based on the given dataset information about the user's previous actions during a listening session along with acoustic features of the previous songs; and further predict the user's current interest or mood based on the song choices.**

I. INTRODUCTION

We all listen to music in our daily lives. We select a preferred song and the song that we listen to in a session depends on our current mood. The challenge for content providers is to envision how a given user will react to some content, such that users are provided with content that brings out positive reaction. Recommender systems are used to analyze the patterns of customers to predict what else they might be interested in. This can be very important for internet-based stores to keep customers interested in their product by suggesting the ones that they might be interested in. Machine learning has been leveraged to build recommender systems which recommends music that pleases users. This can be made possible by incorporating various improvements in the music streaming platforms like customizing the playlists for each and every user based on his taste and preference and suggesting a user about a new release which he/she might be most interested of. However, there has not been much research done on how a user interacts sequentially with the music that he listens in a particular listening session.

In our project we focus on the famous music streaming application - Spotify. Spotify does not have dislike button, which would have provided a user's interest over a playlist. Rather, we have another feature known as the skipping behaviour of a user. This serves same purpose as a dislike button and sends out powerful signal about the user's like and dislike. Being able to utilize this skipping behaviour in the context of the entire listening session is the key to the recommender system for suggesting relevant content. The focus of our project is to build a sequential skip preiction machine learning model which will predict if a user will skip over a particular music track. This prediction is carried out based on the user's interactions with the previous songs and the song's musical qualities in an individual music listening session. The input data set is provided as a integrated file of audio features and user behaviours for various listening sessions and the output will be the prediction of a track being 'skipped' or 'not skipped'. And thereby we can narrow down the user's current mood. Knowing the mood of a user will help spotify in recommending songs for a listening session and thereby improving its user experience. We have implemented simpler sequential based models like LightGBM (Light Gradient Boosting Machine), RNN (Recurrent Neural Network) and LSTM (Long Short Term Memory) and predicted the mood of the user which can of any one of the following: Happy, Sad, Energetic, Calm and Anxious.

II. RELATED WORK

Spotify released the Music Streaming sessions data set (MSSD) as part of the Spotify Sequential Skip prediction Challenge. The task of the challenge is to predict whether individual tracks encountered in a listening session will be skipped by a particular user. The MSSD consists of 160 million streaming sessions with associated user interactions, audio features and metadata describing the tracks streamed during the sessions, and snapshots of the playlists listened to during the sessions. The data was split into various sessions. Each session is divided into two nearly equal halves, with the information about tracks and user interaction features in first half and track information alone in second half. However, the user interaction features are available only for the first half. The main task is to predict if the user skipped any of the tracks in the second half. Chang et. al worked on this dataset by merging the acoustic and behavioural features of the first ten tracks with the acoustic features of the second ten. It is then passed through several convolutional layers as well as a self attention layer to output the predictions for tracks 10 to 20.

Embedding layer was not used in this experiment, but directly sent in the concatenated feature vectors. They also explored both metrics and sequence learning methods by finding the sequence learning to be more successful. Beres et. al. took a different approach to the challenge by choosing to use Bi-LSTM to autoencoder as well as feature selection to create their model.

In this project, in order to capture the dynamics of a session, we have selected three different models: LightGBM binary classifier, Recurrent Neural network which can encode the temporal information well and LSTM with a memory cells. We plotted the loss and accuracy of each of the model and compared the performance results.

III. Dataset and features

For our project we used Spotify Sequnetial Skip Pridiction dataset which was owned and released by Spotify. The whole dataset consists of 130 million session details which is approximately about 50 gigabyte. Each session holds maximum of 20 music tracks. We received the the data as two files. One file consists of feature details and the other file consists of user session data. The user behaviour features and track ids are provided for the first half of the session. The second half of the session is provided with only the track ids. Theses track ids are linked with the features of the tracks which can be obtained from the Spotify API. The user behaviour features details on the user's activity in the streaming session. Some of the user features are: a boolean to specify is the user paused a song, hour of the day, etc. The music features details on the duration of the song, popularity estimate and audio characteristics breakdown. Some of the features of the music includes tempo, danceability, energy, loudness, liveness, etc. This dataset is then cleaned for any missing information or null value. Rows with null values are updated as -999. We then select 50,000 random rows of data for our project. The user behaviour features and the music features of each track in the first half of the session are merged using track id as the primary key. This merged dataset is converted to a csv file named as "spotify merged dataset" and it is used for training our models.

In the user behaviour features list, there are 3 boolean columns which represent the skipping behaviour of the user: 'skip1', 'skip2' and 'skip3'. Each of these skip columns represents the duration when a song a being skipped. Skip1 indicates that a song is skipped as soon as it is played; skip2 indicates that a song is skipped when it is a third of the way through a track; skip3 indicates that the song is skipped almost near the end of the track. We chose 'skip2' for project since it is more ideal and indicates the user's actions if they like or dislike the song.

IV. Methods

In our project, we have chosen 3 machine learning models:

- LightGBM
- RNN
- LSTM

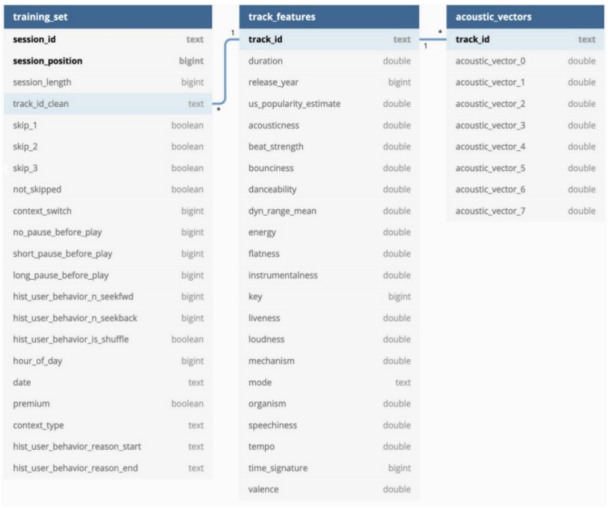

Fig. 1. Spotify Dataset

A. LightGBM

We first started with the Gradient Boosted Trees (GBT) model. LightGBM is a supervised learning model that extends the gradient boosting algorithm by adding a type of automatic feature selection as well as focusing on boosting with larger gradients. Gradient boosting refers to a class of ensemble machine learning models that can be used for classification predictive models. The boosting method in this model generates predictors sequentially and builds off of the previous data. This model improves upon the predictions of the first tree in order to build the second tree and continues the same way with other trees. This process of building upon the previous data aligns with the task that we are trying to solve which is the sequence-based classification. Basically, we will be building a model for each track that we are predicting based on the previous track. Light GBM shows high speed, low memory and capacity to handle large datasets. Thus this model when used for our model with huge data will show best prediction results.

We have split the data into training and testing sets of 67 percent and 33 percent respectively. The input dataset is then fed into the GBM model. The loss function that we used in our model was "Cross entropy". The purpose of cross-entropy in our case is to essentially measure the distance between the probability distribution that our model generates for a certain playlist and the true distribution. Minimizing this distance would result in a stronger model.

B. RNN

Next we chose the Recurrent Neural Network or the RNN model for our prediction. Deep learning models has been used in a variety of predictive tasks and have shown great results in many of them. A neural network is considered to be deep as long as it has more than one hidden layer. This unfolding reveals the reason behind neural networks ability to understand sequences and lists. RNN model which best suits

for sequential data adds additional weights to the network and maintains the internal state. Thus by adding the state to the neural network, this model can explicitly learn and exploit the context in predicting the sequential problems. These models can also remember its previous inputs and takes historical information for computation. In our project we need to keep track of the users interaction with the track. We will not be able to conclude if a song will be skipped or not based on a user's first skip instance which would be unfair.

We have spitted the data into training and testing sets of 67 percent and 33 percent respectively. The input dataset is fed into the RNN model with 'sigmoid' as activation function. We have one input layer and three dense layers of 28, 8 and 2 neurons respectively. The loss function that we used for our RNN model was "Categorical Cross entropy". The output result may fall under any of the possible categories. The purpose of cross-entropy in our case is to quantify the difference between two probability distributions. The more shorter this distance, the more efficient our model will be. We use Adamax optimizer to improve the accuracy of the model. Adamax is a variant of Adam optimization technique which has proved to be stable in the infinity order form. It is an adaptive form of Stochastic Gradient Descent (SGD) and it outperforms SGD with respect to insensitivity towards the parameter choices. Also, with sigmoid as activation function, it produces an activation based on the input and multiplies it by the weights of the succeeding layer to produce further activation.

C. LSTM

The final model that we chose for our project is the Long Short Term Memory (LSTM). Of the different deep learning models, the LSTM implementation have shown exceptional results in predictive tasks of time series data. RNNs are too good at using information from previous timesteps, however they can not go back very far in time and past a certain point it becomes very difficult to connect the information. To solve these problems the LSTM model uses memory cells. A memory cell is a structure made up of four main components: an input gate, a forget gate, an output gate and a neuron with a self-recurrent connection. The weight of the self-recurrent connection is set to one, which ensures that without any outside interference the memory cells state will remain constant between different timesteps. The input, output and the forget gates are used to modulate the interactions the memory cells has with its environment. The input gate can either allow or block incoming signals that would alter the state of memory cell. The output gate can decide if the state of the memory cell will be allowed to affect other memory cells and the forget gate can affect the memory cells self-recurrent connection allowing it to forget its previous state if needed. The hidden layer of a LSTM network can also be attached to any type of output layer like any other neural network depending on the task at hand, whether that task is classification or regression.

The input dataset is fed into the LSTM network. We represent every item in our playlist as a one-hot vector of shape 1*N. The reason for using one-hot vectors is first of all due to the fact that it makes our output very easy to calculate and because our keras framework uses one-hot vector internally. After the final hidden layer we reach the output layer. This is the layer where we generate our predictions for the mood of the listener. The loss function that we used for our LSTM model was "Categorical Cross entropy". The purpose of cross-entropy in our case is to essentially measure the distance between the probability distribution that our model generates for a certain playlist and the true distribution. Minimizing this distance would result in a stronger model. Adamax optimizer is used with Relu as activation function. The major advantage of relu function is that it just picks up the maximum value and it does not perform the expensive exponential operation like sigmoid function. Thus it shows better convergence performance than the others.

V. Experiments and Discussion

A. Test train split:

The main dataset is a combination of two different datsets called Track features dataset and User session dataset. The dataset is merged based on the track id from the track features dataset. Two thirds of the data is used to train the model and one third of the data is used as test data.

Library used to split the data : "scikitlearn"

B. Feature tuning:

1) LightGBM:

- For this model, various features like tree length, number of epochs and the number of iterations are used to fine tune the model
- The graphs were analyzed and the features were finalized
- Loss function : cross entropy loss
- Learning rate : 0.1
- Epochs : 10

2) RNN:

- Simple RNN was used to build the model and the number of hidden layers are modified to fin tune the accuracy
- The graphs were analyzed and the features modified based on the graphs to improve the accuracy
- Activation function : relu (input layer) and sigmoid (Dense layer)
- Loss function : cross entropy loss
- Learning rate : 0.001
- Epochs : 5

3) LSTM:

- Number of units/cells where chosen and CdNNLSTM was also used to train the model quickly
- The graphs were analyzed and the features were finalized
- Activation function : Relu and sigmoid
- Loss function : cross entropy loss
- Learning rate : 0.001
- Epochs : 4

VI. Model Predictions

The user mood is predicted based on various song features like acousticness, loudness, dancibility , liveness and speechness The features were used to classify each listening session into one of energetic, chill , romantic and cheerful moods. Songs with high acousticness and low speechness are classified as chill mood Songs with high dancibility are classified as energetic Songs with high liveliness and speechness are classified as romantic Songs with high dancibility and speeciness are classified as cheerful.

VII. Results and Comparison

Model	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
LightGBM	58.59	54.75	68.41	68.93
RNN	50.16	50.08	69.32	69.31
LSTM	50.24	50.42	69.33	69.33

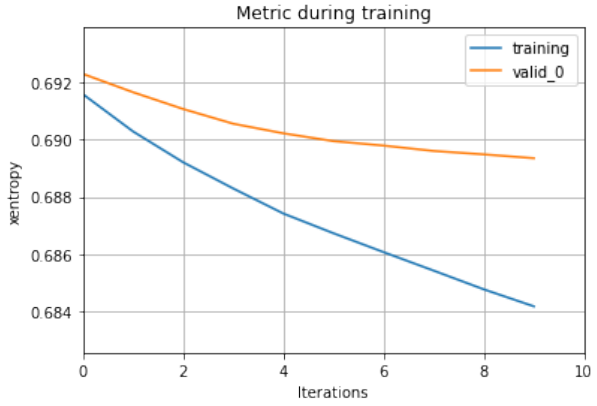

Fig. 2. Epoch Vs Loss - GBM

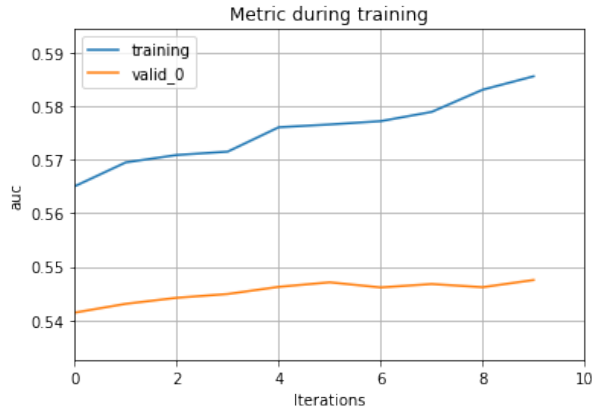

Fig. 3. Epoch Vs Accuracy - GBM

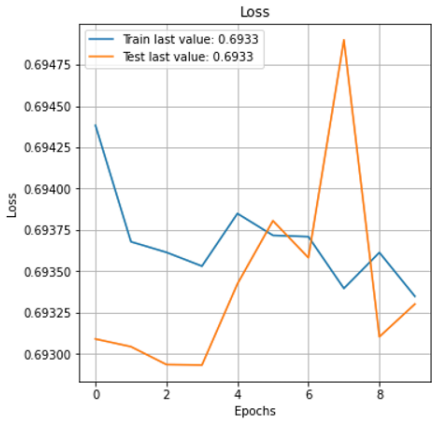

Fig. 4. Epoch Vs Loss - RNN

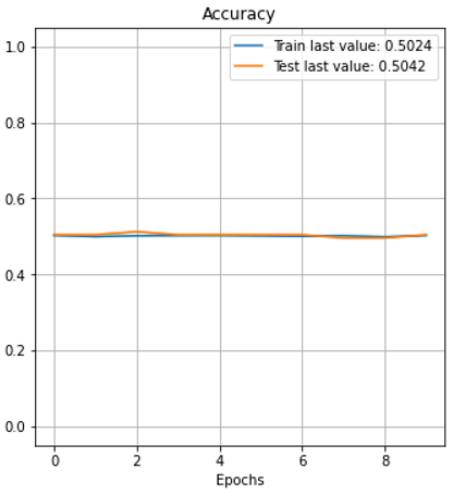

Fig. 5. Epoch Vs Accuracy - RNN

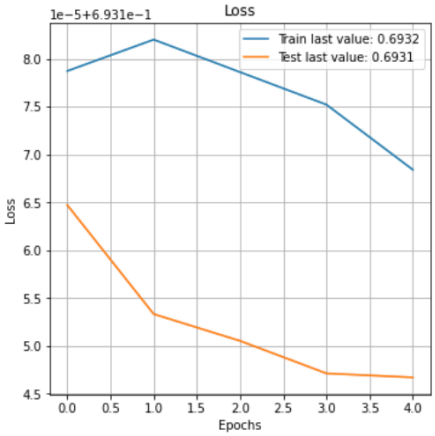

Fig. 6. Epoch Vs Loss - LSTM

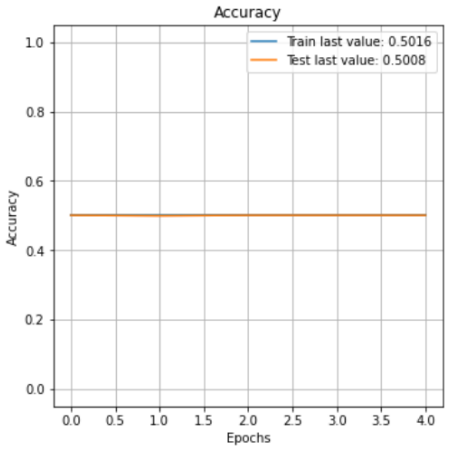

Fig. 7. Epoch Vs Accuracy - LSTM

VIII. Implementation challenges

The dataset provided by spotify has too many features that where not required for this project and the dataset had high noise which took a considerable amount of time to clean the dataset. Since the dataset was too big , we ended up considering a part of the dataset which had a direct impact on the model training and prediction.

The data is distributed in two different files which had to be merged based on a common feature which then had to be filtered and normalized.

IX. Future Work

This project can be a motivation for two further works:

Recommendation system

Now that this project can predict the mood of a user in a listening session, this prediction can be used to build a recommender system that can suggest songs based on the mood of the user hence improving the user experience and user interaction with the application.

Health care based alert system

Users health can be monitored and people suffering with mental health issues can be taken care of and given proper attention that they require based on there responses to various songs that they listen to.

X. Conclusion

Spotify is a widely used music application that directly has an impact over user's life, so it's important to increase the usability of the application with the help of machine learning. Spotify has lots of user data that can be analyzed and used to implement algorithms that can predict user's moods and then help them personalize songs and also assist them improve their mental health.

XI. Code Repository

The code to our implementation can be found in the below repository. https://github.com/AFA21SCM21BO/Project-Mood-Prediction

References

[1] Brian Brost, Rishabh Mehrotra, and Tristan Jehan. 2019. The Music Streaming Sessions Dataset. In Proc. the 2019 Web Conference ACM.
[2] Oscar Celma. 2010. Music recommendation. In Music recommendation and discovery. Springer, 43–85.
[3] Yoon Ho Cho, Jae Kyeong Kim, and Soung Hie Kim. 2002. A personalized recommender system based on web usage mining and decision tree induction. Expert systems with Applications 23, 3 (2002), 329–342.
[4] Chang, S., Lee, S., and Lee, K. Sequential skip prediction with few-shot in streamed music contents. CoRR, abs/1901.08203, 2019.
[5] Sutskever, I., Vinyals, O., and Le, Q. Sequence to sequence learning with neural networks. Advances in NIPS, 2014.
[6] Adapa, Sainath. (2019). Sequential modeling of Sessions using Recurrent Neural Networks for Skip Prediction. 10.13140/RG.2.2.31567.33440.
[7] Hurtado, A., Wagner, M.Mundada, S. (2019). Thank you, Next: Using NLP Techniques to Predict Song Skips on Spotify based on Sequential User and Acoustic Data.
[8] https://link.medium.com/M101B5PoJlb