Thank you, Next: Using NLP Techniques to Predict Song Skips on Spotify based on Sequential User and Acoustic Data

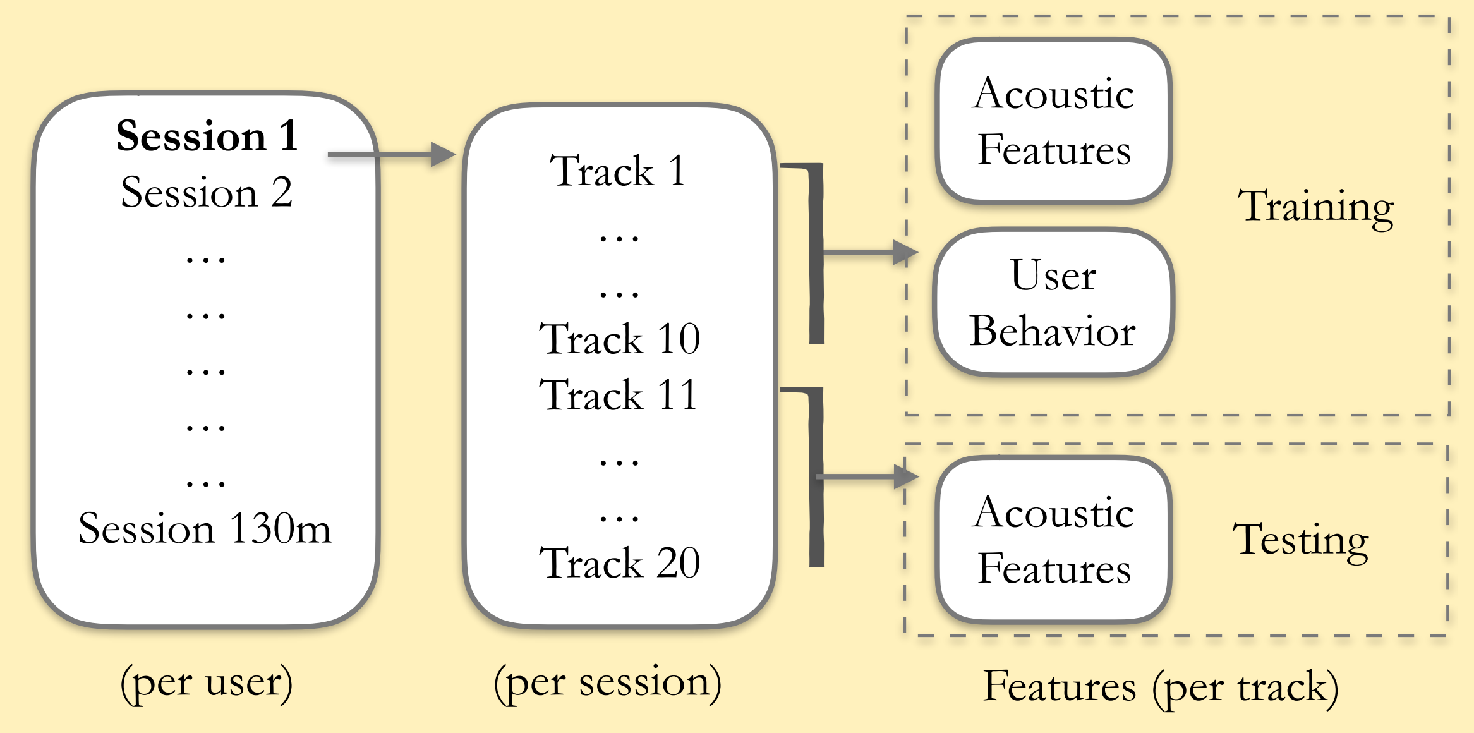
Introduction

Music consumption habits have changed dramatically with the rise of streaming services like Spotify, Apple Music, and Tidal. The skip button plays a large role in the user's experience, as they are free to abandon songs as they choose. Music providers are also incentivized to recommend songs that their users like in order to increase user experience and time spent on the platform.

Machine learning in the context of music often uses recommender system. There hasn't been much research on how a user's interaction with music over time can help recommend music to the user.

Data and Preprocessing

Dataset: Spotify Sequential Skip Prediction dataset [2], consisting of roughly 130 million listening sessions with associated user behaviors.
Each session consists of multiple music tracks (songs, podcasts, etc.). User interaction features are provided for the first half of the session, but only track ids are provided for the second half.



Pre-Processing: We merged user behavior and acoustic features for each track using the track ids. We also preprocessed categorical features into one-hot representations and normalized them (z-score and min/max). This pre-processing created an input track embedding for our model.

Features Details: We chose to use 'skip_2' as our output label for whether the song was skipped/not skipped since it better represents whether a user likes the track they are on.

Accuracy Metric

Our evaluation metric on the task of sequential skip prediction is defined as mean average accuracy (MAA). For a single example session, we can calculate average accuracy as $AA = \frac{1}{T}\sum_{i=1}^{T} A(i) \cdot 1\{y^{(i)} = \hat{y}^{(i)}\}$, where T is the number of tracks in the given session, $y^{(i)}$ is the ground truth label for track i, $\hat{y}^{(i)}$ is the predicted label for track i, and $A(i)$ is the accuracy in the session up to track i.

We leveraged two loss functions for our task. The first is a simple binary cross-entropy loss function, and the second is calculated as the binary cross-entropy loss minus average accuracy per batch. For a single example session, we define

$$\text{Loss} = -\frac{1}{T}\sum_{i=1}^{T}\left(y^{(i)}\log \hat{y}^{(i)} + (1-y^{(i)})\log(1-\hat{y}^{(i)}) + A(i)1\{y^{(i)} = \hat{y}^{(i)}\}\right)$$

Baseline 1: Gradient Boosted Trees

We used the Gradient Boosted Trees (GBT) algorithm as our baseline because the boosting method generates predictors from the algorithm sequentially [1], aligning with our task of sequence-based classification.

We decided to implement Light GBT, due to its high speed, low memory, and capacity to handle large datasets.

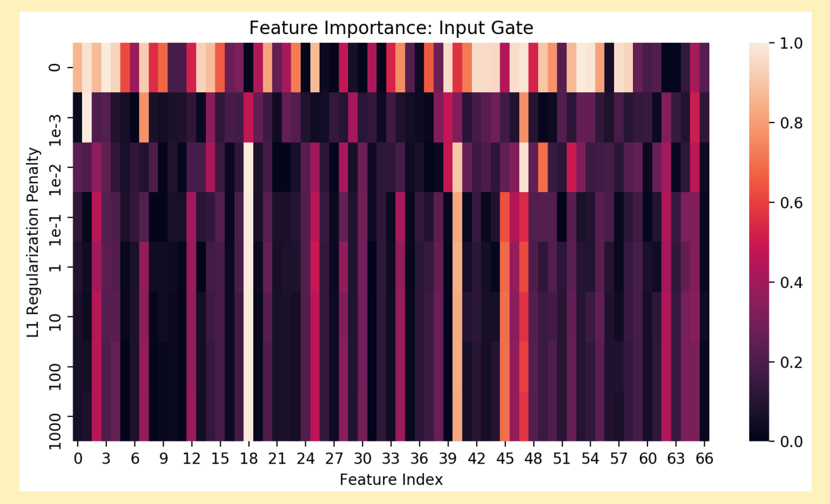
Method	Subset Accuracy
Gradient Boosted Tree	0.516202049

LSTM and BiLSTM

As our recurrent neural models, we opted to use LSTMs and BiLSTMs. Specifically, we leveraged LSTMs and BiLSTMs in an encoder-decoder architecture to train and evaluate our sequence-to-sequence skip prediction task. In the world of sequential deep learning models, LSTMs have experienced broader use over GRUs due to its gate architecture and it ability to forget, maintain, and regulate cell memory between states. Compared to LSTMs, bidirectional LSTMs are able to perform better on classification tasks since they are able to capture information from both present and future states.
The results from our experiments are given below:

Method	Validation MAA	Test MAA
LSTM - CrossEntropyLoss	0.5699	0.3549
LSTM - CrossEntropyLoss + MAA	0.5836	0.3982
BiLSTM - CrossEntropyLoss	0.5789	0.4041
BiLSTM - CrossEntropyLoss + MAA	0.5759	0.4234

Feature Learning



We can learn about the relative importance of each input feature by examining a heatmap of their activations on the encoder of our LSTM model across increasing L1 regularization penalties. On the input gate (visualized above), we see increased importance on features representing the user skipping the previous track, the user playing a personalized playlist, and the tracks' US popularity estimate, dynamic range mean, and flatness.

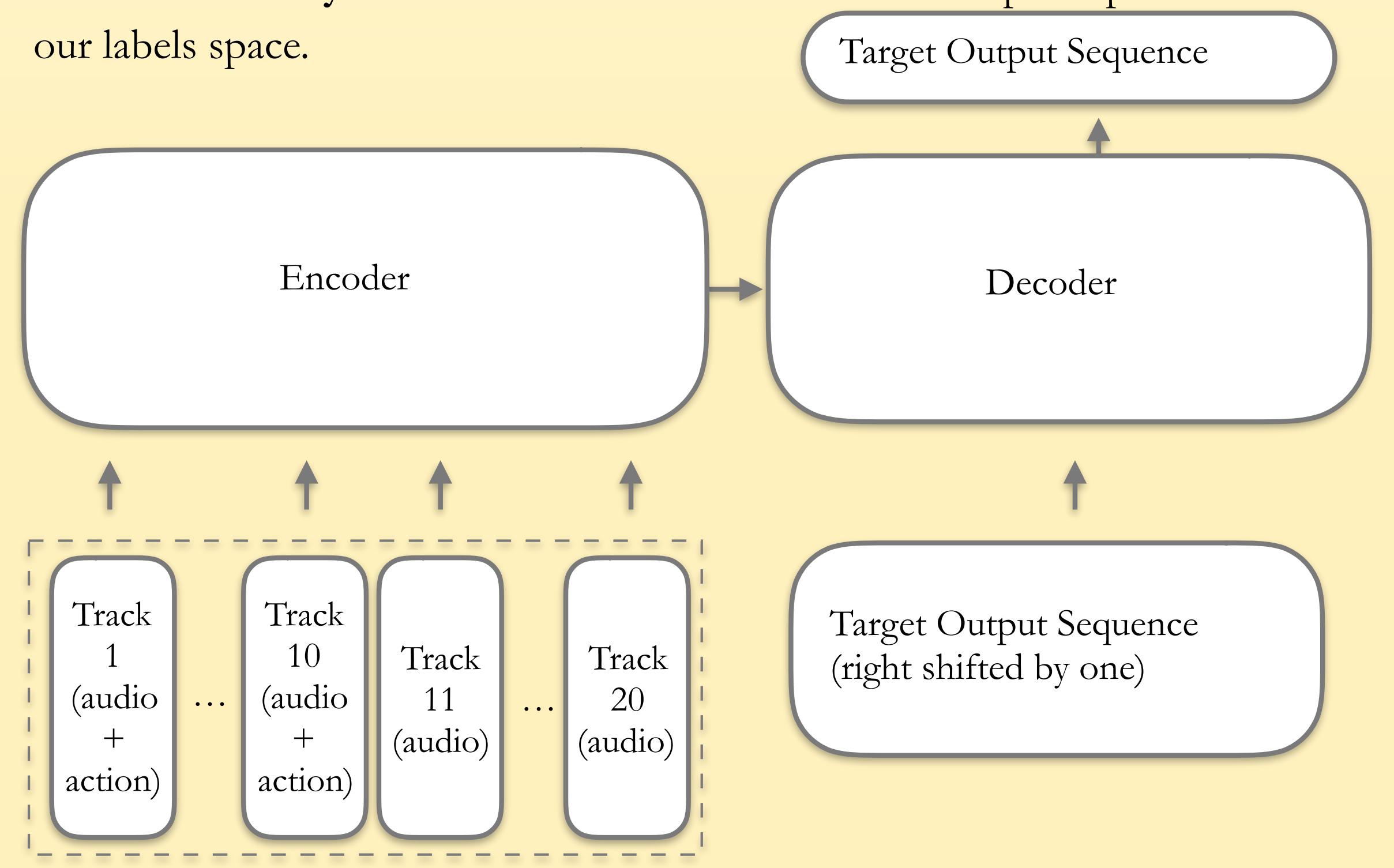
Doing the same procedure for our cell gates, we see that features like session position and session length are highly activated, implying that our LSTM learns to place importance on maintaining memory of a track's relative position in session! Additionally, an analysis on our output gates reveal that our model learns to pass features representing the user skipping the track and playing from a personalized playlist onto the next cell state, as one would expect.

Novel NLP Techniques for Sequential Classification

Transformer - Traditional NLP Model

Inspired by the success of transformers in tackling a variety of different NLP tasks, we were interested in leveraging a **transformer, attention,** and **positional encodings** for a sequential classification task, such as the one we are solving.

Technique: We developed a meaningful **embedding** of the user behavior and acoustic features for each track in the entire listening session. The embedding gets passed into our **encoder**, creating a **hidden state**, which gets passed into the decoder along with the **output target sequence.** The output target sequence that gets passed into the decoder is right-shifted by one so that the decoder uses the output prediction of the previous track and properly predicts in-sequence. We added a **linear layer** at the end of our transformer to map our predictions into our labels space.



Method	Validation MAA	Test MAA
Transformer (Traditional NLP) + MAA	0.308747	0.31359

Transformer - Feature-Forcing

Here, the encoder serves to create a compact representation of the first half of the listening session. The encoder leverages the sequential nature of the data, with the audio and behavior features for tracks 1-10 being passed in as inputs. The output from this encoder is a hidden state that is passed into the decoder as an input. We notice that Transformers are well suited for question response tasks as well as language modeling tasks. Generally, ability comes in part from the decoders' structure, which takes in both the encoder's representation of the inputs as well as the previous target output's label. For our decoder, we leveraged this structure in a unconventional way. For a target track i, instead of sending in the previous prediction of skip_2 (i -1) for a target track, we sent in the audio features of the the track i.
In the end, this experiment proved to be a moderate success. Our feature-forced Transformer enjoyed a MAA of 0.447, which places us at 29th place in the online submissions for the Spotify Sequential Skip Prediction Challenge.

Method	Test MAA	Validation MAA
Transformer (Teacher-forcing)	0.469507	0.470839242
Transformer (Teacher-forcing) + MAA	0.477419	0.409576

Discussion/Error Analysis

Gradient Boosted Trees achieves around 50% accuracy, more than the sequential based models, perhaps because of random guessing and a large difference in this model's architecture and input processing as compared to our other models

LSTM/Bi-LSTM: We can see that both of these models are overfitting to the data, which could be fixed by better hyperparameter searching and tuning. The more detailed internal representation of the sequence could account for why Bi-LSTM performs slightly better than LSTM.

Transformers (Traditional NLP): Based on the output, we can see that the transformer which we developed did not necessarily help improve accuracy. Mapping from an input size of 20 to and output size of 10 (tracks) may not be the best representation of the problem. While the sequential nature of data is modeled, the separation between the first half of the session and the second half of the session is less pronounced in this model, as all tracks are sent together.

Transformer (Feature-Forcing): Having the encoder represent the meaningful information about the first half of the session and having the decoder pay attention to both the representation of the first half and the individual audio features for each given track it was predicted proved to be a better model architecture for the problem at hand. One source of information loss is that the output predictions from the decoder are not incorporated in as inputs for the next sequential prediction, as we are substituting with the track's audio features instead.

Future

Given more time, we would love to explore some of the following topics:

- Improve Transformer model: Dynamically append the output prediction for each track from the decoder with the track's audio features and inject an embedding of this into our decoder
- Have our models work with variable length tracks
- Feature analysis and explainability for the Transformer: This will help us better understand how the attention layers work in the model and which features are more significant for prediction

References

[1] Y. Zhang and A. Haghani, "A gradient boosting method to improve travel time prediction," Transportation Research Part C: Emerging Technologies. 2015.
[2] Brost, B., Mehrotra, R., and Jehan, T. The music streaming sessions dataset. In Proceedings of the 2019 Web Conference. ACM, 2019.
[3] Schedl, Zamani, C. D. Y. and Elahi. Current challenges and visions in music recommender systems research. International Journal of Multimedia Information Retrieval, 7, Jun 2018.
[4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need