SPOTIFY SKIP ACTION PREDICTION

Name: Krishna More

POINTS OF DISCUSSION

INTRODUCTION

Abstract

DATASET

Featuress

MODEL IMPLEMENTATION

Accuracy Metric

Future Work

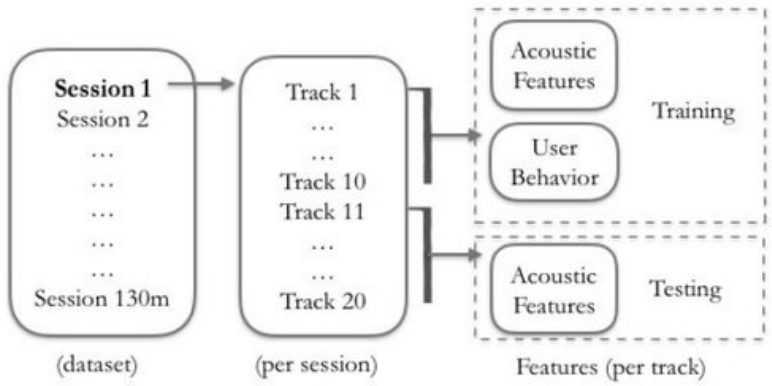
INTRODUCTION

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 a 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.

ABSTRACT

Music consumption habits have changed dramatically in the past decade with the rise of streaming services such as Spotify, Apple Music, and Tidal. Today, these services are ubiquitous. These music providers are also incentivized to provide songs that users like in order to create a better user experience and increase time spent on their platform. Thus, an important challenge to music streaming is determining what kind of music the user would like to hear. The skip button is one feature instance on these music services that play a large role in the user's experience and helps identify what a user likes, as with this button, users are free to abandon songs as they choose. Thus, in this project, we will build a machine learning model that will predict if a user will skip a song or not give information about the user's previous actions during a listening session along with acoustic features of the previous songs.

DATASET



Dataset Structure

We are using the Spotify Sequential Skip Prediction dataset for our project (Brost et al., 2019), which consists of roughly 130 million listening sessions. Each session has at most 20 music tracks. All the user behavior features and track ids are provided for the first half of the session, but only the track ids are provided for the second half. The track ids are associated with the track's acoustic features that can be extracted via the Spotify API. The user behavior features consist of actions like whether the user paused, hour of day, [illegible] acoustic features consist of [illegible] like danceability, tempo, etc.

FEATURES

IN ORDER TO PRE-PROCESS THE DATA FOR TRAINING, WE MERGED USER BEHAVIOR AND ACOUSTIC FEATURES USING TRACK IDS FOR EACH TRACK IN THE FIRST HALF OF A LISTENING SESSION. WE THEN PRE PROCESSED CATEGORICAL FEATURES INTO ONE-HOT REPRESENTATIONS AND NORMALIZED THEM (USING EITHER Z-SCORE AND MIN/MAX). THIS PROCESS CREATED AN INPUT TRACK EMBEDDING FOR OUR MODEL. FOR THE TRACKS USED IN THE TESTING PHASE (TRACKS FROM THE SECOND HALF OF THE SESSION), WE ONLY USED THE PRE-PROCESSED ACOUSTIC FEATURES EMBEDDINGS. WE CHOSE TO USE THE 'SKIP 2' FEATURE AS OUR OUTPUT LABEL FOR WHETHER A TRACK WAS SKIPPED/NOT SKIPPED. THIS IS BECAUSE 'SKIP 2' INDICATES WHETHER A USER SKIPPED A TRACK EARLY ON (I.E. A THIRD OF THE WAY THROUGH A TRACK), WHEREAS OTHER THE SKIP FEATURES 'SKIP 1' AND 'SKIP 4' RESPECTIVELY INDICATE WHETHER A USER SKIPPED THE TRACK ALMOST AS SOON AS IT PLAYED OR NEAR THE END OF THE TRACK. 'SKIP 2' IS MORE IDEAL IN INDICATING THE USER'S ACTIONS OF LISTENING TO A SONG FOR A LITTLE BIT AND FIGURING OUT IF THEY LIKE/DISLIKE THAT SONG (SINCE MANY PEOPLE TEND TO SKIP SONGS THEY LIKE EVEN HALFWAY THROUGH)

MODEL IMPLEMENTATION

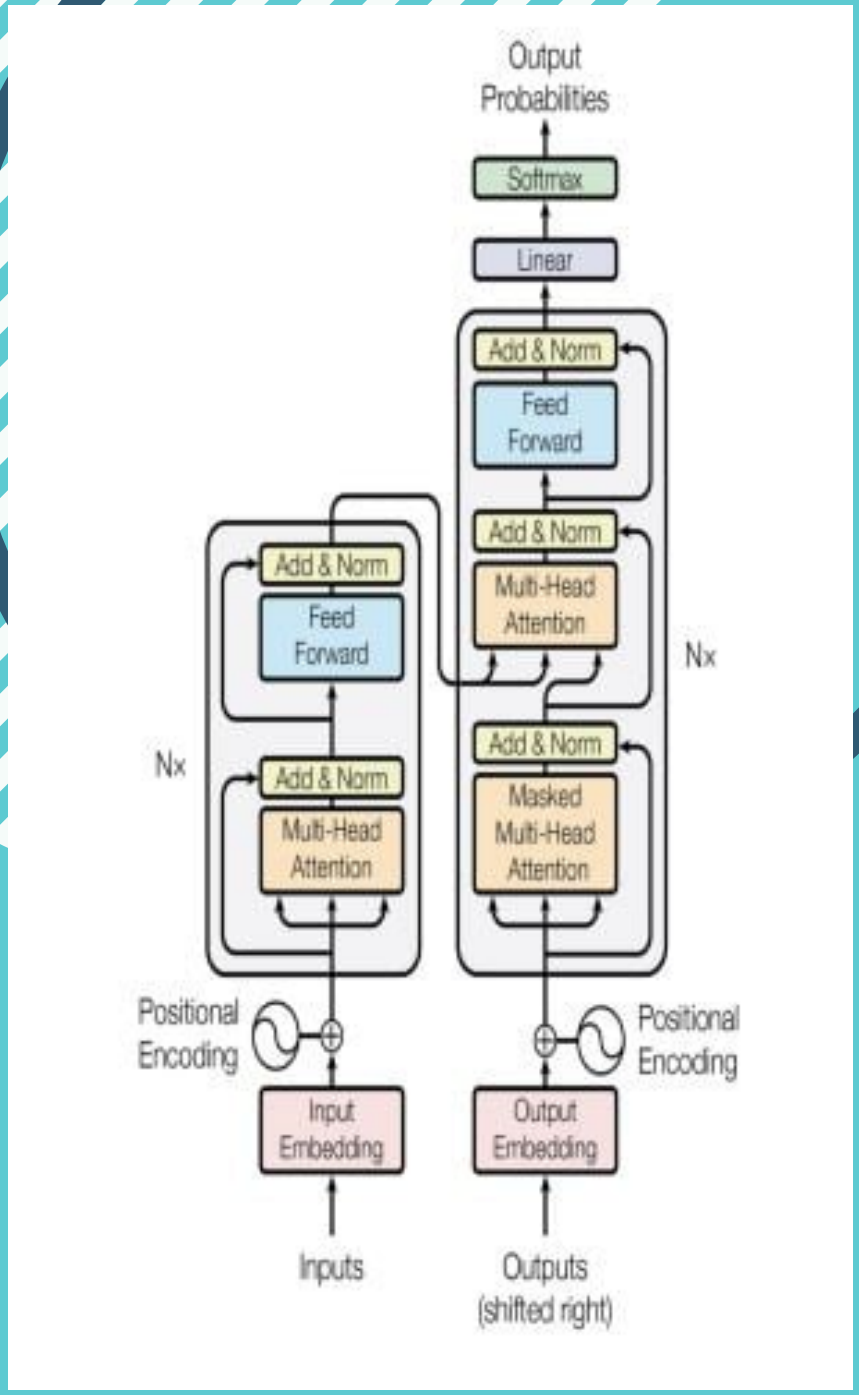
1.Preliminary Baseline:

GRADIENT BOOSTED TREES WE STARTED WITH USING THE GRADIENT BOOSTED TREES (GBT) ALGORITHM AS OUR BASELINE BECAUSE THE BOOSTING METHOD IN GBT GENERATES PREDICTORS SEQUENTIALLY AND BUILDS OFF OF PREVIOUS DATA (ZHANG & HAGHANI, 2015); I.E. THE IDEA OF GRADIENT BOOSTED TREES IS TO IMPROVE UPON THE PREDICTIONS OF THE FIRST TREE IN ORDER TO BUILD THE SECOND TREE, AND SO ON. THIS METHOD OF BUILDING ONTO PREVIOUS DATA ALIGNS WITH THE TASK WE WERE SOLVING: SEQUENCE-BASED CLASSIFICATION WHERE WE WANT TO BUILD A MODEL FOR EACH TRACK WE ARE PREDICTING ON BASED ON THE PREVIOUS TRACK. MORE SPECIFICALLY, FOR THE ALGORITHM ITSELF, WE DECIDED TO IMPLEMENT LIGHT GBT, DUE TO ITS HIGH SPEED, LOW MEMORY, AND CAPACITY TO HANDLE LARGE DATASETS.H)

MODEL IMPLEMENTATION

2.Recurrent Neural Baselines: LSTM and Bi-LSTM:

THE NATURE OF THE TASK - BUILDING A SEQUENTIAL SKIP PREDICTION MODEL - LENT ITSELF TOWARDS THE USE OF MODELS THAT COULD UNDERSTAND SEQUENTIAL BEHAVIOR. IN THIS SENSE, RECURRENT NEURAL MODELS LIKE LSTMS AND BI-LSTMS ARE SHOWN TO HAVE GREAT CAPACITY TO COMPREHEND SEQUENCE-BASED, VARIABLE LENGTH DATA WHEN COMPARED TO OTHER STATIC DEEP LEARNING MODELS. THIS ARCHITECTURE USES TWO SEPARATE RECURRENT NEURAL MODELS, THE FIRST TO ENCODE THE INPUTS INTO A SINGLE HIGH DIMENSIONAL STATE, AND THE SECOND TO DECODE THE STATE PASSED FROM THE ENCODER TOWARDS THE DEFINED PREDICTION TASK.H)



3.TRANSFORMER: TRADITIONAL NLP METHOD

Due to the sequential nature of our data, when investigat ing what advances in other domains may be applicable to our problem and while experimenting with the LSTM/Bi LSTMs, we found many parallels to NLP. Like many nat ural language problems, our challenge required a model that could leverage the sequential structure of the data and understand both long and short-range dependencies. Upon investigating the current advances in NLP, we were drawn to the Transformer and the introduction of attention mecha nisms and positional encodings in the encoders and decoders.

4. Transformer: Feature-Forcing Method

We also decided to take another novel approach to solving this challenge: enhancing the architecture of the traditional NLP transformer model for our specific task and data. Thus, the question became ”how might we leverage the structure of the transformers to predict skips?”. Transformers do well with taking an input sequence and converting it into an output sequence (i.e. language translation). However, our problem is unique in that only the second half of the input sequence (session) needs predictions, while the first half has ground truth labels we want to leverage in the second half’s predictions. With this problem structure in mind, we came to create our own model

Accuracy Metric

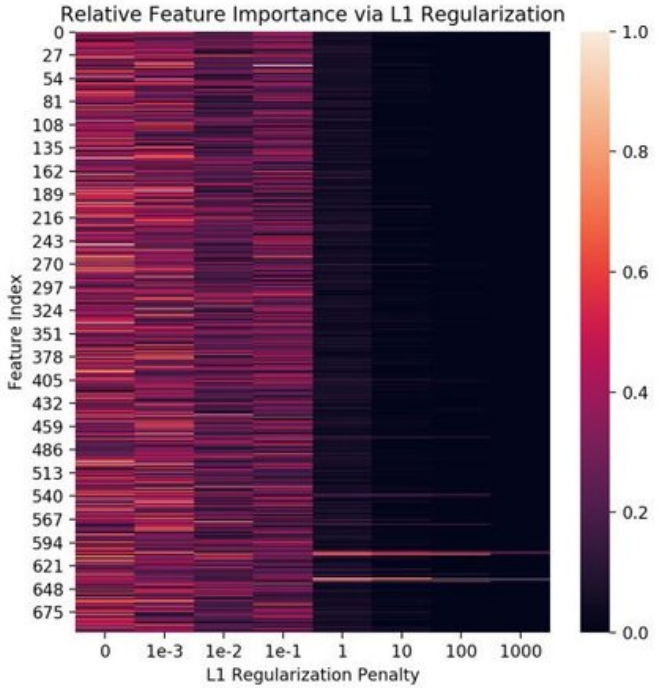
The evaluation metric used in the original Spotify Sequential Skip Prediction Challenge is mean Average Accuracy (m AA or MAA). For a single example session i with T tracks, we can define Average Accuracy as

$$AA = \frac{1}{T}\sum_{i=1}^{T} A(i) \cdot 1\{y^{(i)} = \hat{y}^{(i)}\}$$

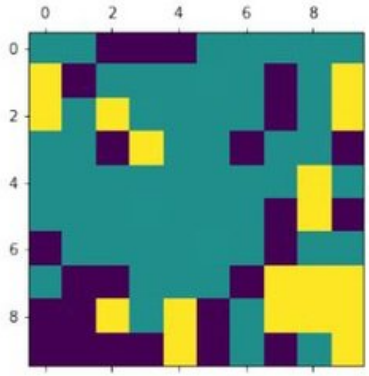
Ground truth sequence	Predicted sequence	AA
1, 1, 1, 1, 1	0, 1, 1, 1, 1	0.543
1, 1, 1, 1, 1	1, 0, 1, 1, 1	0.643
1, 1, 1, 1, 1	1, 1, 0, 1, 1	0.710
1, 1, 1, 1, 1	1, 1, 1, 0, 1	0.760
1, 1, 1, 1, 1	1, 1, 1, 1, 0	0.800

Below is a datatable with the rest of our models' accuracies.

Method	Validation MAA	Test MAA
LSTM - CrossEntropyLoss	0.5699	0.3549
LSTM - CrossEntropyLoss + MAA	0.5836	0.3982
Bi-LSTM - CrossEntropyLoss	0.5789	0.4041
Bi-LSTM - CrossEntropyLoss + MAA	0.5759	0.4234
Transformer (Traditional NLP) - CrossEntropyLoss + MAA	0.3088	0.3136
Transformer (Teacher-forcing) - CrossEntropyLoss	0.4708	0.4695
Transformer (Teacher-forcing) - CrossEntropyLoss + MAA	0.4736	0.4580



Heatmap of relative weight importance of each feature across increasing L1 regularization penalty



The Encoder-Decoder Attention Weights Visualized

Future Work

Given more time, we want to explore the transformer architecture more. Based on the transformers implemented successfully in NLP applications and the performance of our feature-forcing transformer, we could combine aspects of the traditional NLP transformer with the feature-forcing transformer in order to create a better model. We could dynamically append the output prediction for each track from the decoder with the track's audio features and inject these concatenated embeddings into our decoder for prediction. This architecture would perhaps better follow the transformers that are known to work well in NLP, as well as would better represent our model by considering the audio features that are inputted into the decoder. Finally, exploring variable-length sessions and improved feature analysis would help us better understand and improve our model.

THANK YOU