Spotify-Skip-Action-Prediction

Nourhan Sowar

Introduction

As streaming services like Spotify become the go-to providers for music, it is increasingly important that these platforms are able to recommend the right music to their users.

Machine learning has long been leveraged to build recommender systems in order to recommend music.

The focus of our project is to build a sequential skip prediction model to predict if a user will skip over a track based on the user's interactions with previous songs and the song's musical qualities in an individual music listening session.

Music Streaming Sessions Dataset (MSSD)

- ➔ It consists of over 160 million listening sessions with associated user interaction information.
- ➔ It contains 2 CSV files
  - ◆ The first is log which contains the streaming sessions as shown in (Figure2).
  - ◆ The second includes features for the tracks like acousticness, a measure of confidence that the track is acoustic.

The Music Streaming Sessions Dataset (arxiv.org)

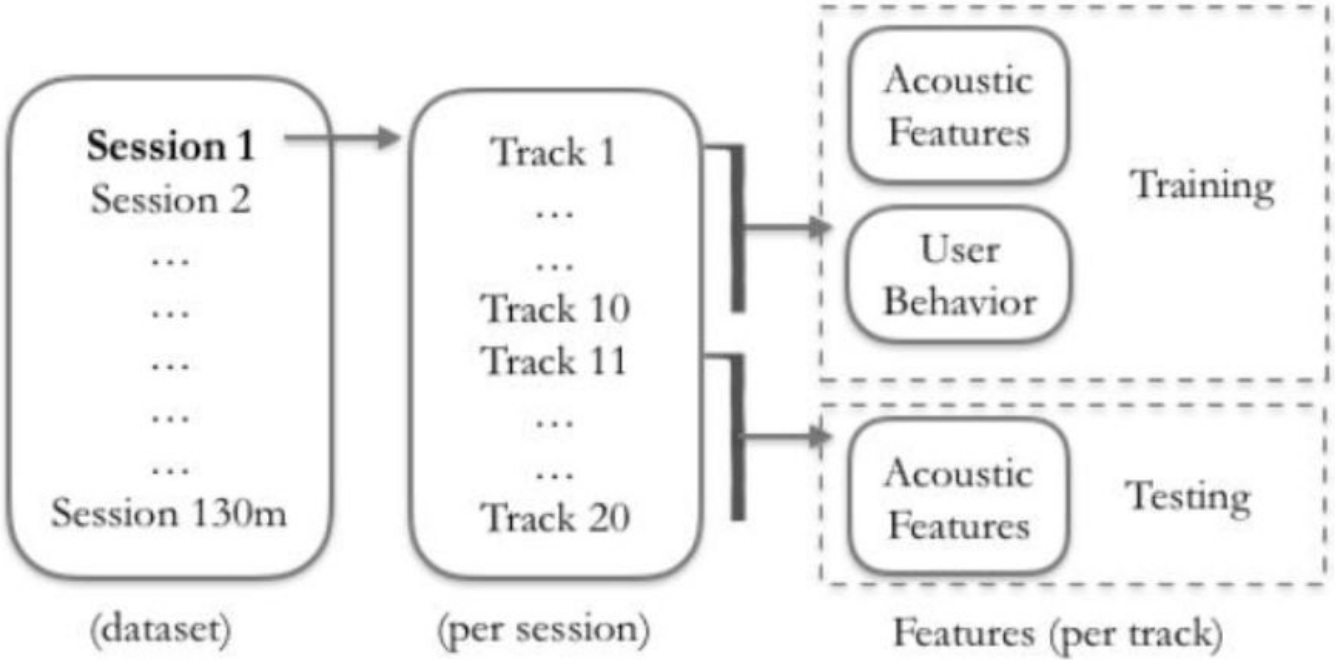

Figure 1: Dataset Structure

Column name	Column description
session id	unique session identifier
session position	position of track within session
session length	length of session
track id	unique track identifier
skip 1	whether the track was only played very briefly
skip 2	whether the track was only played briefly
skip 3	whether most of the track was played
not skipped	whether the track was played in its entirety
context switch	whether the user changed context between the previous row and the current row
no pause	whether there was no pause between playback of the previous track and current track
short pause	whether there was a short pause between playback of the previous track and current track
long pause	whether there was a long pause between playback of the previous track and current track
num seekfwd	the number of times the user scrubbed forward during playback
num seekbk	the number of times the user scrubbed backward during playback
shuffle	whether the track was played with shuffle mode activated
hour of day	hour of day (integers between 0 and 23)
date	date in YYYY-MM-DD format
premium	whether the user was on premium or not
context type	what type of context the playback occurred within
reason start	cause of this track play starting
reason end	cause of this track play ending
uniform random	whether shuffle would be uniformly random for this session

Figure 2 : log_mini.csv

track id	duration	release year
popularity	**acousticness**	**beat strength**
bounciness	danceability	**dyn range mean**
energy	**flatness**	**instrumentalness**
key	**liveness**	loudness
mechanism	mode	**organism**
speechiness	tempo	time signature
valence	**acoustic vector**	

Figure 3 : features.csv

Preprocessing

- ➔ Exploring the Spotify API in Python
  - ◆ USing Spotify API for developer to enhance and update the features in the csv file by connecting the Spotify with python script using spotipy module and get the data we want.
- ➔ May we use spark and pyspark.

Feature Engineering

- ➔ Track skip information
  - ◆ As each track is identified by a tracking ID, we can calculate the total number of listenings, as well as the skip ratio for every song and for each skip type.
- ➔ Session heterogeneity
  - ◆ The information about track repetitions. Thus we considered i.)repeat count, the number of times a given track was repeated in the current session, and ii.)unique track ratio, which is the number of unique tracks in the session divided by its length
- ➔ Skip-pattern features
  - ◆ For the 1024 possible skip patterns, the least frequent one appeared in 3278 sessions of the public data. The binary model itself reached MAA 0.546 and AUC 0.695

Model

- ➔ Gradient Boosted Trees
  - 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.
  - high speed, low memory, and capacity to handle large datasets.
- ➔ LSTM and Bi-LSTM
  - 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.
- ➔ Transformer: Feature-Forcing Method
  - The encoder outputs a set of attention vectors K and V, which the decoder utilizes in the encoder-decoder attention layer.

Results

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