Miniproject

1. Brief description of your ML-based application, including the purpose, what the system takes as an input and what it produces as an output.

The application would be a session-based music recommendation algorithm, which takes one song as an input, and it produces a playlist (a set of songs) as an output. It was identified in recent studies that deep learning-based “neural” approaches tend to perform worse in classical session-based recommendation tasks than simpler algorithms, such as kNNs. The purpose is to evaluate a state of the art kNN session-based recommendation to a state of the art DL-based recommendation system, on Spotify's Million Playlist Dataset.

2. Which ML technique are you planning to use at the core of your system and why do you think it is suitable.

I'm planning to use session based kNN (SKNN), which instead of recommending based on the last event in the current listening session, it compares the entirety of the current listening session to the previous sessions of the dataset while training.

Given a session ***s***, determine the ***k*** most similar past sessions (neighbours) ***Ns*** by applying a suitable session similarity measure, like the Jaccard index or cosine similarity on binary vectors. Next, given the current session ***s***, its neighbours ***Ns***, and the chosen similarity function ***sim(s1,s2)*** for two sessions ***s1*** and ***s2***, the recommendation score for each item ***i***:

$$scoreknn(i,s) = \Sigma n \in Nssim(s,n) \cdot 1n(i)$$

Using V-SKNN, which is a sequence-aware form of SKNNs with vector multiplication, obtaining better results are plausible. This technique weighs the most recent event in the session as "1", whereas the previous elements are considered gradually less important from the weigh of recommendation, via linear decay function.

3. Which feature extraction techniques are you planning to use and why do you think they are suitable, or explain why you plan to use directly raw data.

This is not thought through in detail at this point, but using the features of individual tracks on the playlist, such as: artist name, album year, genre mixed with audio features of tracks coming from Spotify API, such as liveness, loudness, tempo, etc.

4. What is the data set that you are going to use to train and test your system, explaining whether you are going to build it by yourself or source it elsewhere.

I plan to use Spotify's Million Playlist Dataset, which gives access to the following details to playlists from Spotify's library:

- 'collaborative': boolean (describes whether or not it is a collaborative playlist)
- 'duration_ms': int (the duration of the entire playlist in milliseconds)
- 'modified_at': int (the Unix Epoch Time value of when the playlist was last modified)
- 'name': str (name of the playlist)
- 'num_albums': int (number of unique albums in the playlist)
- 'num_artists': int (number of unique artists in the playlist)
- 'num_edits': int (number of times the playlist has been edited)
- 'num_followers': int (number of users that follow the playlist)
- 'num_tracks': int (number of tracks on the playlist)
- 'pid': int (the playlist ID number, ranging from 0 - 999,999,999)
- 'tracks': list of track objects (contains a list of tracks, where each track is an object containing the following attributes:
    - 'album_name': str (the name of the track’s album)
    - 'album_uri': str (the unique album ID -- uniform resource identifier)
    - 'artist_name': str (the name of the artist)
    - 'artist_uri': str (the unique artist ID -- uniform resource identifier)
    - 'duration_ms': int (the duration of the track in milliseconds)
    - 'pos': int (the track’s position in the playlist)
    - 'track_name' : str (the name of the track))

Moreover, I aim to connect additional data to individual tracks within these playlists via Spotify API and Audio Features. I believe that they may reveal interesting patterns or shared features of music similarity and could affect whether specific songs are included within a playlist. Similarly to this approach.

5. How are you planning to evaluate your system or compare it with others.

Technically evaluating against a state of the art RNN-based recommendation algorithm, gru4rec. Additionally, if time allows a very small qualitative user study (1-2 people) on which recommendation algorithm performs better (maybe benchmark with Spotify's original recommendation).

6. What is going to be your original contribution in this project and what will you source elsewhere.

The individual contribution will be to base the sequence-based recommendation on more meaningful track features form the Audio Features API, as explained in the dataset section. This aligns with the suggestions from a recent publication's future work proposal: *Looking at future directions, in particular methods that leverage **side information** about users and **items** seem to represent a promising way forward.* To use a V-SKNN based on these features for recommendation is considered a unique contribution, however the conceptual background of this algorithm will be sourced from fellow researchers. Also the implementation of the evaluation algorithm will be sourced from somewhere as that is not the main contribution of this project.