Computer Perception Extended - Winter 2022 / 2023

AI creating a Spotify playlist

Roles:

Marina Willi: Frontend, Documentation
Calra Widmer: Backend, Documentation

Initial idea:

A website that receives prompts from the user describing their current or desired mood. The AI then creates a playlist containing songs that reflect the given prompt. The user should then be able to save the created playlist on their spotify profile, that was connected via an API provided by spotify.

Concept:

Miro Board:
https://miro.com/welcomeonboard/V1dyVGk2QXVuVWt1ZWRLNHBJNzZ2bWVtZkplV3FZaTg3a2d5amRBbTZaOFNubnZ0aVlaVkg5MnVrNDdIamhpMXwzNDU4NzY0NTM1MjkyMzU1OTI5fDI=?share_link_id=983047047405

Documents and Code:

Code Base of our project: https://github.com/Kritz3l/bingChilling
Website that actually does kinda what we wanted: https://deej-ai.online/
Code to his project: https://github.com/teticio/deej-ai.online-app
Other article that we read:
https://towardsdatascience.com/create-automatic-playlists-by-using-deep-learning-to-listen-to-the-music-b72836c24ce2
Train a Word2Vec model with Spotify lists ("sentences") of tracks ("words"):
https://github.com/teticio/Deej-AI/blob/master/notebooks/Track2Vec.ipynb
Spotify API: https://developer.spotify.com/documentation/web-api

Difficulties:

We already encountered our first difficulties with the AI early on in our project. How should we train our AI? On what Inputs and what would we like to have as an output?
First we searched through Hugging Face for a similar approach or project, sadly we couldn't find anything so we went and asked almighty Google for help. We found some websites and tutorials we read, but nothing matched our initial idea.
We had absolutely no Idea on how to train our AI, so we thought we would take a look at the notebooks that were provided to us on Colab. But that raised even more questions, like what should our table with the training data look like? In the end, we decided to "fake" the AI part and do a table where we would provide songs with keywords and write code that would search for matching keywords and post an API call to Spotify with the URI to the song.

We also had difficulties with the Spotify API. Mainly because two weeks into our project they changed their API Documentation. When searching for a tutorial on how to use the Spotify API we had a range of tutorials to choose from, the only thing was that they mainly used frameworks we were not familiar with. We couldn't find a Tutorial solely based on Node.js, plain HTML, CSS and JS. Carla had worked previously with APIs and Node Servers so she built a rudimentary server that would correspond with the API Authentication. But as one problem was solved, two new ones arose. First one: the server wouldn't serve a static folder on a url with get. Second one was that the API calls were declined with the error: "Access Token has expired". That raised some questions, mainly because the authentication with the personal account didn't cause any problems, so why would the access token be expired?

In the end, we agreed on doing all the parts of our projects separately, and later with more experience, build the app properly.

The changes we made to our idea:

We had to do a few changes to our initial approach due to the previously listed difficulties that we encountered along the way. Because of the continuous issues with the AI and its implementation we decided with Guillaume to "fake" the AI so that our end result at least works when we showcase it.
We decided to have a "working" front-end to showcase our idea properly and give the viewer a clear idea what our intentions with the project were.
Another important part for us was that the user could log in with their spotify account.