Mood2Music

Art and Machine Learning: Final Project

Group 19
Aarthi Ramsundar, Audrey Zhang, Nikitha Murikinati, Qixuan Teo

May 2022

PROJECT CONCEPT

Generating Mood-Based Personalized Music with AI

Pipeline

Spotify Lyric Data or User Spotify Playlist Data

Prompt-Based Lyric Generation with Fine Tuned GPT-3

Mood Based Artist and Genre Selection using Clustering

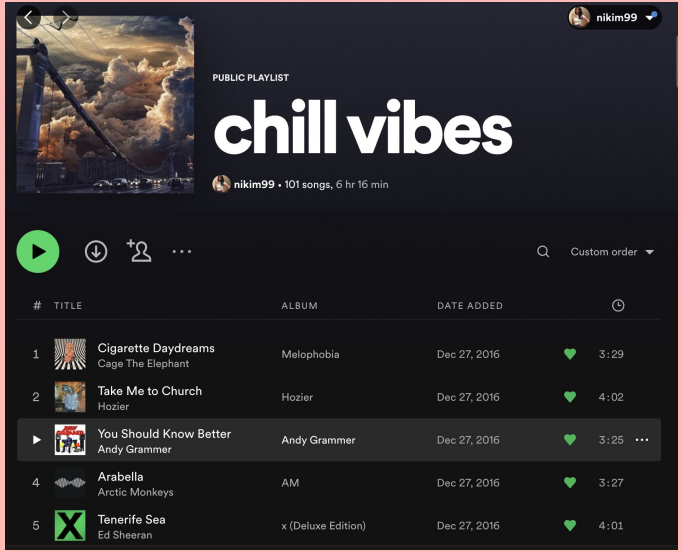
General Spotify Dataset or User Spotify Library

AI-generated Instrumental and Vocal Music with Jukebox

Human Reinterpretation of AI musical output

SAMPLE MUSICAL PROFILE: NIKI

- Generally Indie/Pop and Alternative Genre
- Artists include:
  - Cage the Elephant
  - Ed Sheeran
  - All Time Low
  - OneRepublic
  - Jonah Kagen
  - again&again



GENERATED LYRICS

- Fine-tuned GPT3 on one of Niki's Spotify playlists
- Passed in various prompts that Niki generated and evaluated which was best
  - Ex Prompts: "Far from home", "Fake love is", "Monsters come out", "Next time we", "A long time"
- **Evaluation Criteria**:
  - Syntax: Whether the lyrics had meaning (made sense as English phrases)
  - Structure: Whether they followed the structure of songs and where grammatically correct
  - Emotional Connection: Whether the user whose data was used to train the personalized model liked the lyrics and felt that they matched to the type of music that they typically listened to.

GENERATED LYRICS

Prompt - “Far From Home”

She said she broke it down
And then collected them like model ships
We danced through the rain
Her jaw was set and her eyes were blue

(Chorus)
And now she's leaving on an eastbound train
To find her youth in the west
But neither one of us will know where
She sleeps or who she kisses there
It's hard to leave an old love alone
Just hard to leave her

And now she's over the earth
On a westbound train
In a place where the years have no number
She sits alone, she sits alone (x2)

From birth until forever
No words are ever lost
The future frozen in the present
The pens that I hold
Take my future from me
But you can touch my heart
And support it alone

(Chorus Repeats)
And now she's over the waves
And sleeping in them too
She drives this shell
And goes on a westbound train

To be young and broke, in love but still alone
Must sound like heaven to you, babe

(Chorus Repeats)
And now she's over the waves
And sleeping in them too
She drives this shell
And goes on a westbound train

(Repeat Twice)
But she's young and broken too, whoa
She's young and broken too
Must sound like heaven to you

Mood Based Artist and Genre Selection

- “Akinator” method:
  - User inputs starting song that resembles their current mood
  - On a scale of 1 (prefer first song) to 10 (prefer second song), would you prefer song A or song B?
    - Start with original vs new, switch to last round winner vs now
    - Q4: The A Team - Ed Sheeran vs. Rocket Man - Elton John
    - Q5: The A Team - Ed Sheeran vs. Photograph - Ed Sheeran
    - Q6: Photograph - Ed Sheeran vs. This Town - Niall Horan
- Using top-k weighted scores:
  - Ed Sheeran | Pop
- Evaluation
  - Is this an artist/genre whose music the user likes/would listen to?
  - Does songs of this artist/in this genre tend to match the user's current mood?

AI GENERATED AUDIO OUTPUT

Human Interpreted AI Generated Output

Lyric Generation Methods

- Two fine-tuning sets used
  - Used Spotify's API to pull user's Spotify playlist songs and Genius Lyrics web scraping to get the lyrics
  - Used songs by artists in Jukebox's artist list and got lyrics from Genius lyrics
- Used these lyrics to finetune GPT3 Curie for 4 epochs
  - Training with more epochs didn't make much of a difference
- Passed in various 3 word prompts
  - Longer prompts tended to confuse it

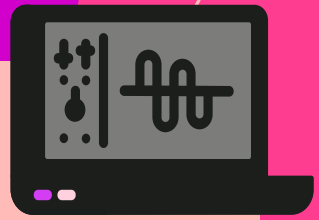
Artist and Genre Selection Methods

- Pulled audio features using Spotipy Python package (1.2M Kaggle dataset, or user's 'Liked Songs')
- Clustering using all combinations of 2/3/4/5 numeric features. Used scikit-learn's KMeans clustering method with num_clusters = N / 100
- Used momentum-based method to perform "Akinator" algorithm, finding out cluster weights (which of the above clusterings is the user really using implicitly?) and within-cluster truth values (in this clustering, which cluster does the user prefer?)

Clustering Details

- Coarse-to-fine grained search: Start by asking users to select between two very different songs, and slowly narrow down to two similar songs
  - Compare songs by percentile, with similarity measure as the weighted sum of ind(songs are in the same cluster)
- Given user answer from 1-10 of whether they preferred song A or B:
  - Clusterings that have song A and B in separate bins should be higher weighted when percentile is large, and lower weighted when percentile is small
  - Within a clustering, the cluster with the preferred song should be weighted higher.
  - Update percentile based on the strength of the response

Music Generation Methods



- Used OpenAI's Jukebox music generator
- Pass in
  - Artist from clustering results
  - Genre of above Artist
  - Lyrics generated from GPT3
- Ran with 1 billion parameter model
  - Not enough computer power for 5 billion model
- Ran 3 samples in parallel and evaluated which was the best
- Evaluation Method:
  - Lyrical Clarity
  - Artist Input Match
  - Smoothness of Instrumentals/Transitions
  - Emotional Connection with User (Niki)

Reflections

- Final Evaluator (Niki) opinion
  - Captures mood and music taste effectively
- Time/Performance Tradeoffs
  - 5B parameter model could give better results
- Extensively personalized, so not very generalizable
- Future work
  - Retrained BERT embeddings for lyric similarity to get artist/genre
  - Enriching dataset for clustering algorithm
  - Ideally want a faster pipeline, requires Jukebox optimization
  - Further parameter experimentation
  - Attempt different methods of combining AI and human outputs for best effect

Thank You!

References

OpenAI Jukebox: https://openai.com/blog/jukebox/

OpenAI GPT3 finetuning: https://beta.openai.com/docs/guides/fine-tuning

Spotify Audio Data: https://www.kaggle.com/datasets/rodolfofigueroa/spotify-12m-songs
Spotify library scraping code:
ttps://towardsdatascience.com/k-means-clustering-using-spotify-song-features-9eb7d53d105c

Spotify Audio Aura Info:
https://pr-newsroom-wp.appspot.com/2021-12-01/learn-more-about-the-audio-aura-in-your-spotify-2021-wrapped-with-aura-reader-mystic-michaela/

Spotify Wrapped
Info:https://towardsdatascience.com/k-means-clustering-using-spotify-song-features-9eb7d53d105c

Clustering math

$$song\_similarity = \sum_{clusters} weights * \delta(current\ song\ and\ selected\ song\ are\ in\ same\ cluster)$$

$$momentum = M * momentum + L * (5.5 - val)$$

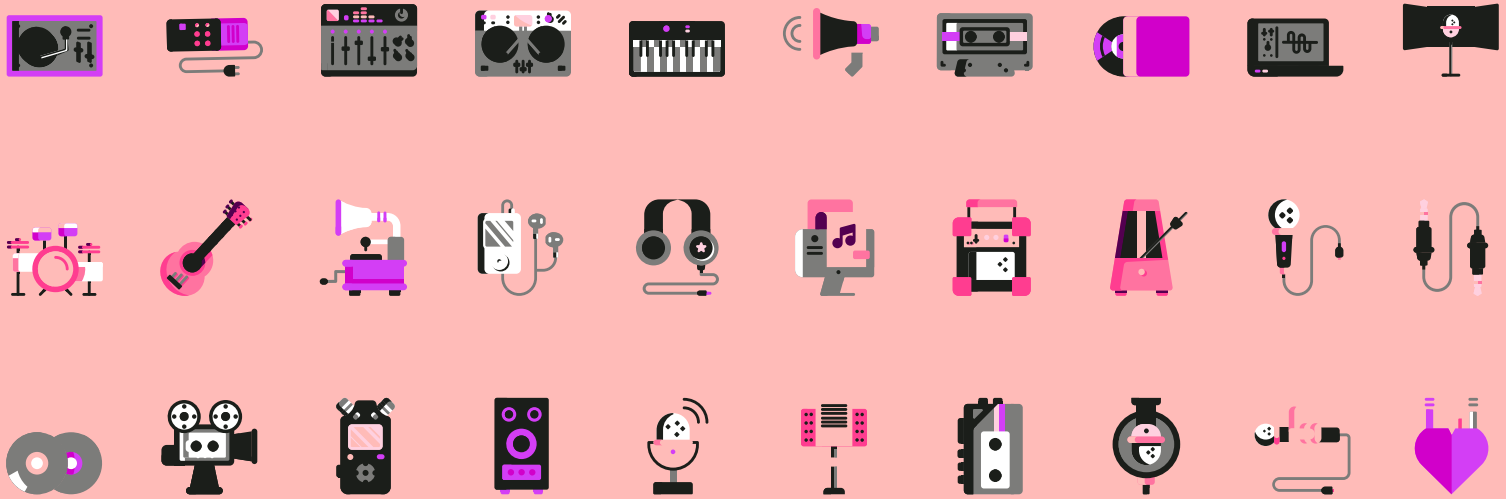
$$weights[\delta(songs\ are\ in\ different\ clusters)] *= (1 + momentum/100)$$

$$initial\_percentile *= (1 - momentum / 100)$$

$$current\_song = other\_song \text{ if } val > 5.5$$

$$song\_weights[song\ in\ same\ cluster\ as\ current\_song] *= (1 - momentum / 100)$$

RECORDING STUDIO ICONS



ALTERNATIVE RESOURCES