Music Recommendation System

Cyrus Sandhu, Ramtin Rezaei, Jaskamal Chouhan, Varsha Adusumalli
CMPT 310 Intro to Artificial Intelligence

Background and Motivation

Users on music streaming platforms often struggle with finding new music that truly align with their preferences. Current recommendation processes like user generated playlists by Spotify tend to recommend tracks based on if they are currently popular or trending. A lot of these recommendation systems don't feature music tracks outside of a specific range of popularity and fail to consider how niche or diverse someone's music taste can truly be which creates limitations with regards to the effectiveness of recommendations. We attempted to amend this problem by using a machine learning algorithm which recommends music to users based solely based off quantitative metrics, primarily the different audio features define a song.

Project Overview

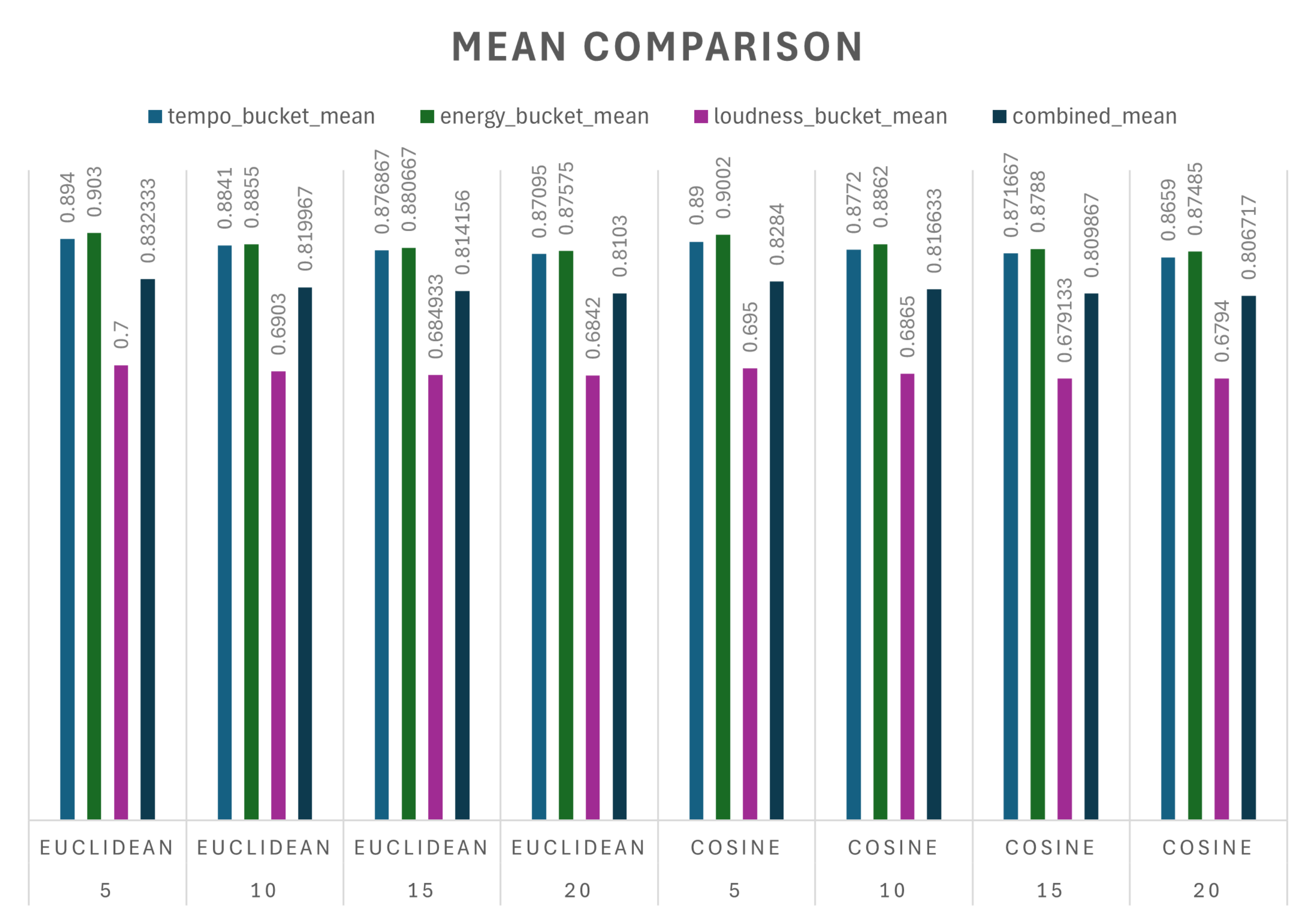
- The music recommendation system is built using unsupervised learning on a third-party data set featuring over 140,000 unique songs with their audio features.
- This was done using a KNN classifier, comparing the results of different hyperparameters to determine the best overall model
- Evaluated model performance through bucket mean comparisons and user feedback

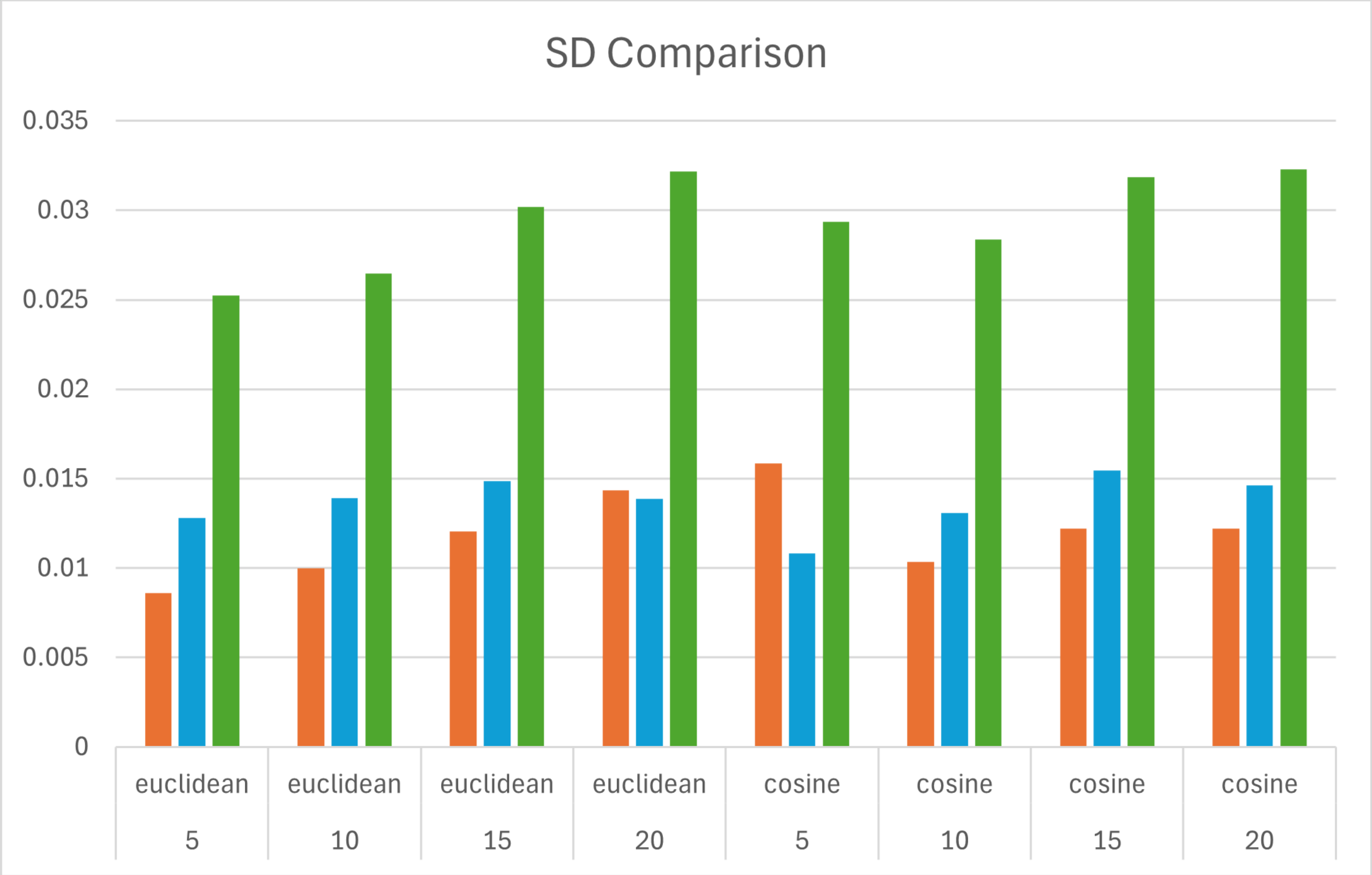
Key Methods

- K-Nearest Neighbors Algorithm
  - Make recommendations based on audio features
- Standard Scaling and other data cleaning techniques
  - Normalize features so values to dominate results
- Cross Validation and Smart Tuning
  - Rigorous testing on model to ensure accuracy
- PyQt6 App
  - Used in creation of User Interface

Training Results

- The model evaluation process featured 5-fold cross validation comparing the effect of different hyperparameters, measuring model accuracy by comparing bucket means for different features.

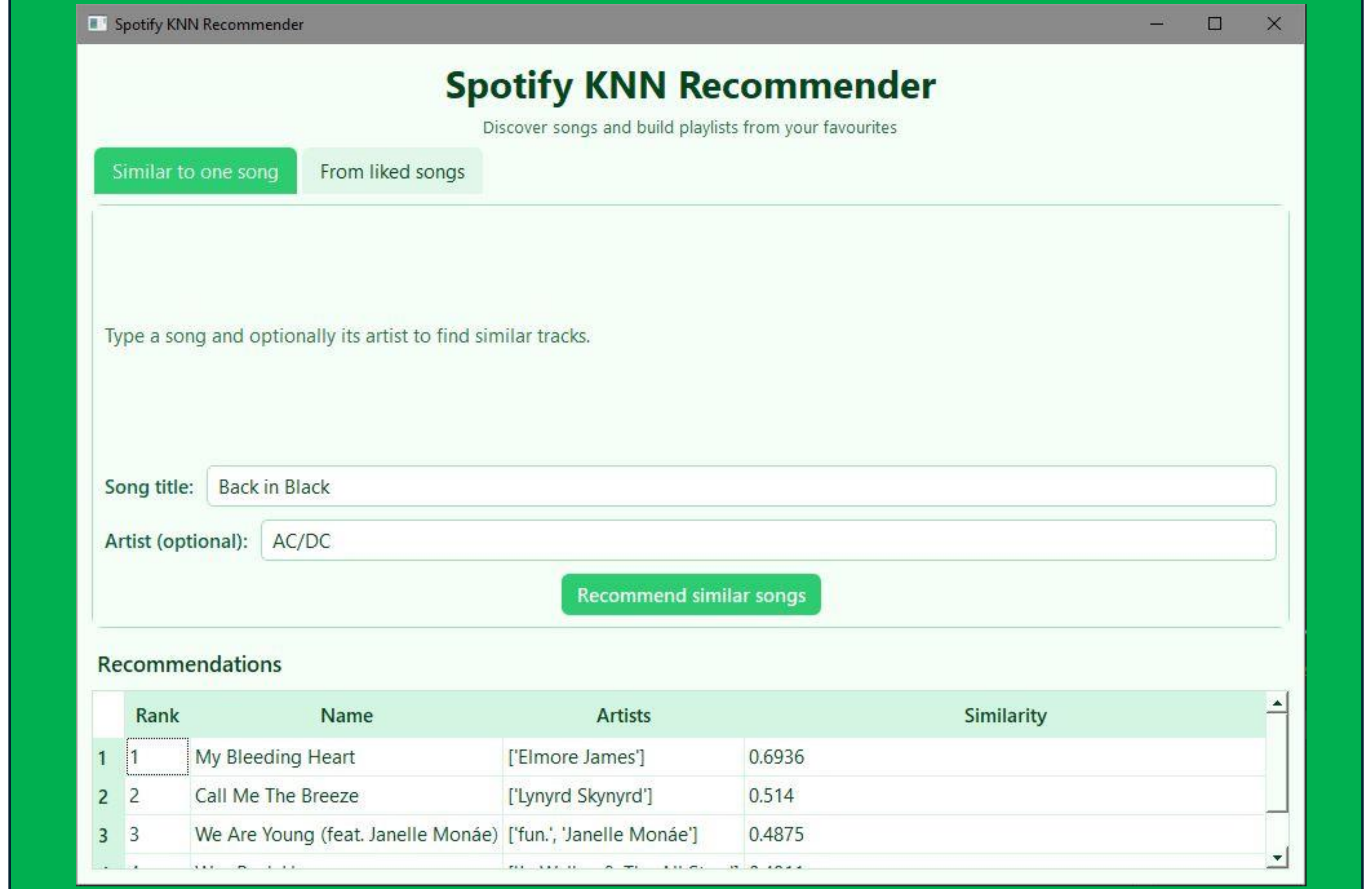




- From our evaluation, we were able to conclude that the optimal model is achieved when we set K=5 and use the Euclidean distance metric to compute nearest neighbors.

Visualizations

- Using the PyQt6 library we were able to create a user interface that was used to facilitate the user input and output process. Below are examples of two different inputs and outputs. The first is an input of just a singular song while the second features a list of songs from the user.



Spotify KNN Recommender

Discover songs and build playlists from your favourites

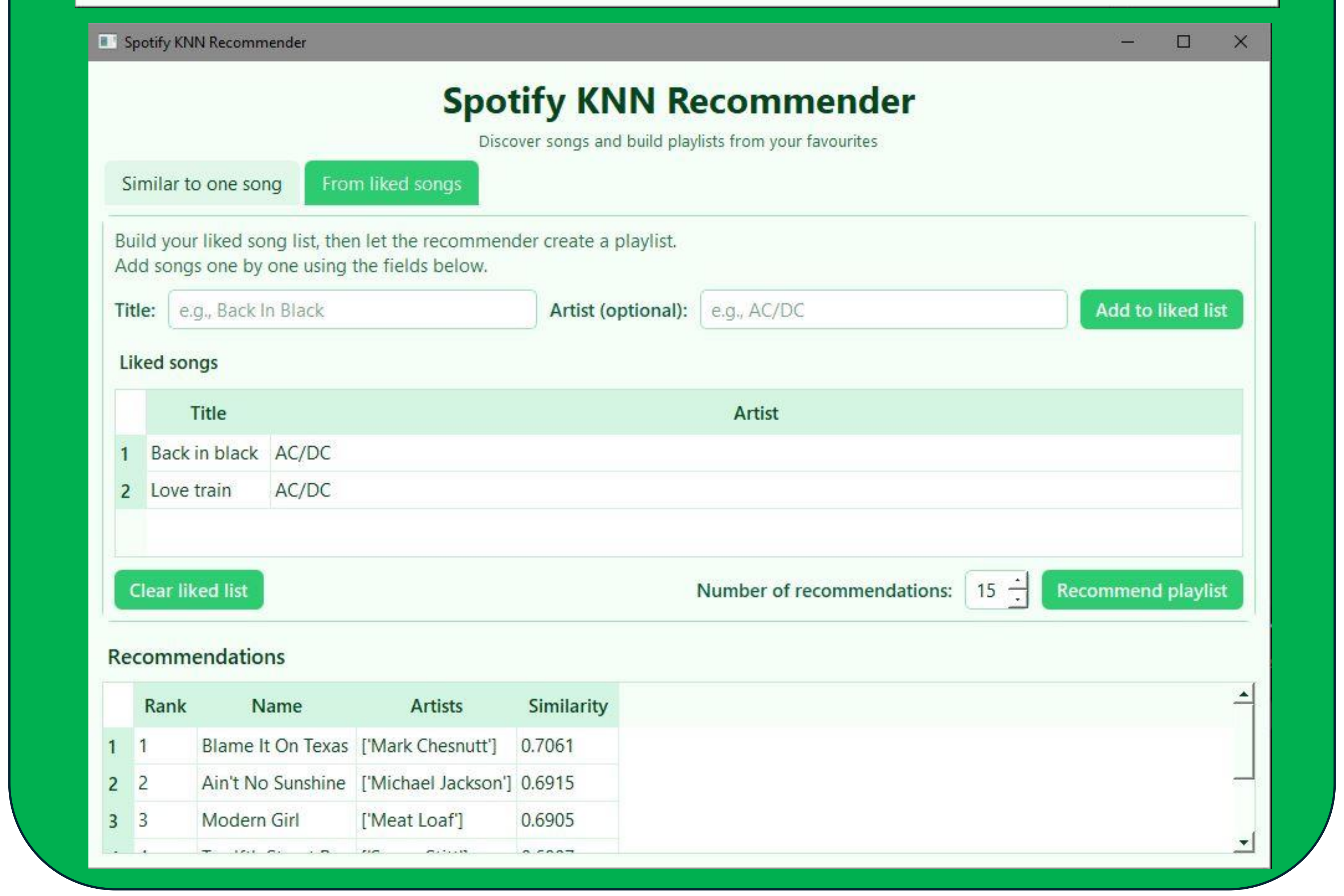
Type a song and optionally its artist to find similar tracks.

Song title: Back in Black

Artist (optional): AC/DC

Recommendations

	Rank	Name	Artists	Similarity
1	1	My Bleeding Heart	['Elmore James']	0.6936
2	2	Call Me The Breeze	['Lynyrd Skynyrd']	0.514
3	3	We Are Young (feat. Janelle Monáe)	['fun.', 'Janelle Monáe']	0.4875



Spotify KNN Recommender

Discover songs and build playlists from your favourites

Build your liked song list, then let the recommender create a playlist.
Add songs one by one using the fields below.

Title: e.g., Back In Black Artist (optional): e.g., AC/DC

Liked songs

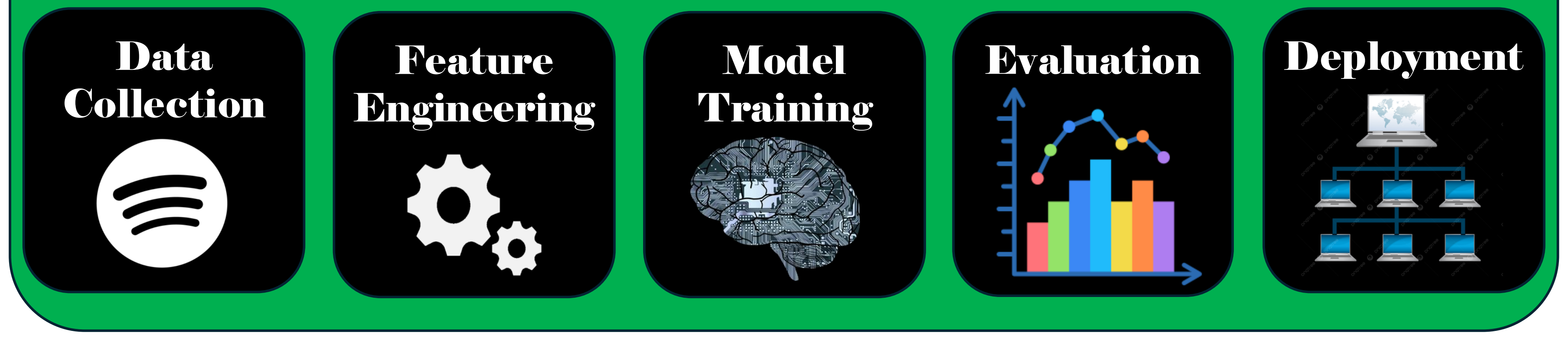
	Title	Artist
1	Back in black	AC/DC
2	Love train	AC/DC

Number of recommendations: 15

Recommendations

	Rank	Name	Artists	Similarity
1	1	Blame It On Texas	['Mark Chesnutt']	0.7061
2	2	Ain't No Sunshine	['Michael Jackson']	0.6915
3	3	Modern Girl	['Meat Loaf']	0.6905

Data Pipeline



References

Aggarwal, C. C. (2016). *Recommender systems: The textbook*. Springer.
Pohle, T., Pampalk, E., & Widmer, G. (2005). Evaluation of frequently used audio features for classification of music into perceptual categories. *International Conference on Music Information Retrieval* (ISMIR).
Schedl, M., Gómez, E., & Urbano, J. (2014). Music information retrieval: Recent developments and applications. *Foundations and Trends in Information Retrieval, 8*(2–3), 127–261.
Scikit-learn Developers. (2023). *Scikit-learn user guide*. https://scikit-learn.org
Spotify for Developers. (2024). *Spotify Web API documentation*. https://developer.spotify.com/documentation/web-api/
Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9, 2579–2605.