Song Recommendation

Data Mining Final Project Presentation
Group 11
Apitsada (Pearl) Ruknapapong, Daeun Ji, Kavya Bhat,
Chi Nguyen, Shubham Kumar

Northwestern | McCORMICK SCHOOL OF ENGINEERING

Agenda

- Project Objective
- Data Overview
- Methodology
- Findings
- Conclusions

Project Objective

- Develop a song recommendation system that suggests songs based on
  - audio features
  - lyrical content
  - emotional responses from listeners

Data Overview

- We combine 2 datasets using Spotify track IDs. The datasets include
  - Spotify song URL & YouTube URL
  - Spotify song attributes & lyrics
- Then, we scrape top 10 YouTube comments per songs based on number of likes from each comment.
- We categorize the comments based on listeners’ emotional responses using **LLM**.
- Lastly, we use **LDA** and **LLM** for topic modeling, retrieving song topic from the lyrics before assigning a topic to each song

Data Overview

Column	Description
Spotify ID	Track ID on Spotify
Artist	Artist name
Spotify URL	Track URL on Spotify
Track	Song name
Album	Album name
Album_type	Album Type (e.g., album, single)
Duration_ms	Duration of song in milliseconds
Song Attributes	Score for song attributes (Danceability, Energy, Key, Loudness, Speechiness, Acousticness, Instrumentalness, Liveness, Valence, Tempo)

Data Overview

Column	Description
YouTube URL	YouTube URL for the song
YouTube Comment Categories	Number of times that listeners’ emotional responses appear in the category (Humour & Memes, Appreciation & Praise, Words of Encouragement, Words of Empathy, Personal Stories & Experiences, Nostalgia & Memories)
Title	Youtube video title
YouTube Video ID	Video ID on YouTube
Lyrics	Song lyrics
Tokens	Word tokens from the lyrics
Dominant Topic Labels	Topic from lyrics (Love & Relationships, Self-Reflection and Personal Struggles, Social and Political Themes, Celebration and Fun, Philosophical and Existential, Storytelling and Narrative, Escape and Fantasy, Spiritual and Religious, Cultural and Lifestyle, Fun and Humor)

Data Overview

YouTube Comment Categories

Lyrics Topics

Raw Data

- Song attribute & lyrics: 955,320 rows x17 cols
- Spotify & YouTube URL: 18,862 x 28 cols

Match by Spotify Track ID

Cleaned Data

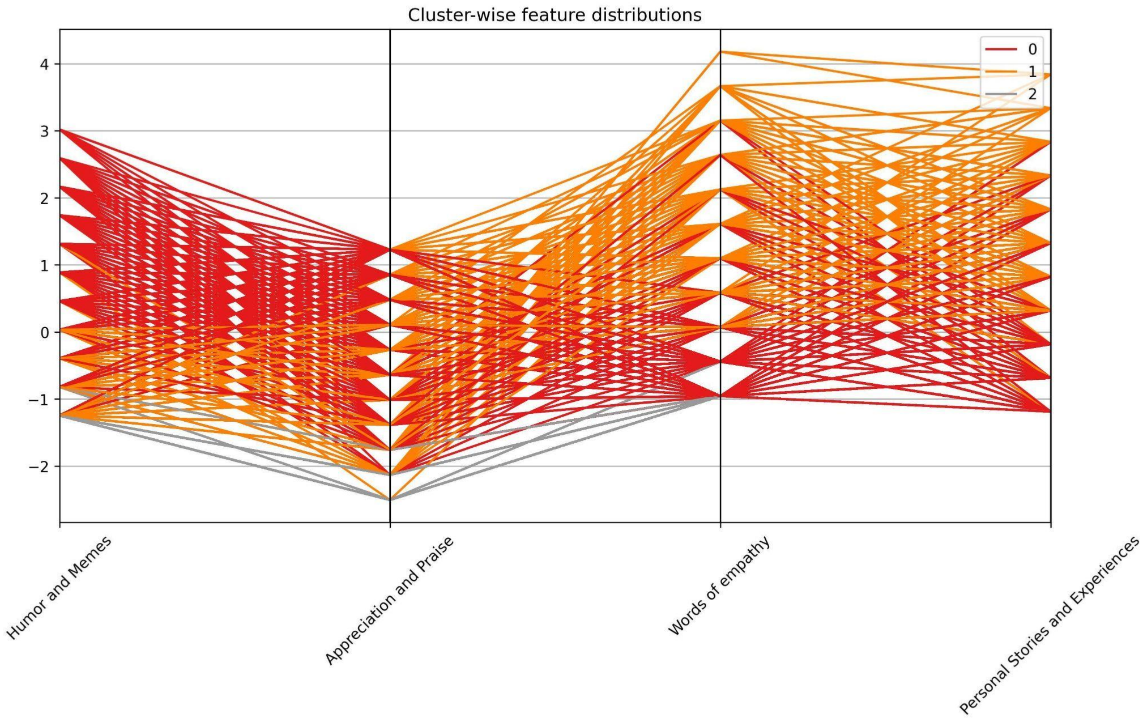
- Filtered English songs
- 3,234 rows x 38 cols

Methodology

- After we have the complete dataset, we use **clustering** to visualize the relationship of the data.
  - Group1: Spotify data (song topics & attributes)
  - Group2: YouTube comment categories
- Next, taking the clustering results for Spotify data and YouTube comments, we use **recommender** to match what audience prefer. We use euclidean and cosine distance to compare existing songs with the songs that user listen to in history.
- Audiences are asked to rank between Spotify data (song topics & attributes) and YouTube comments (audience’s sentiments)
- We also use **multi-label/ multi-class classification** to predict audience response using final dataset.

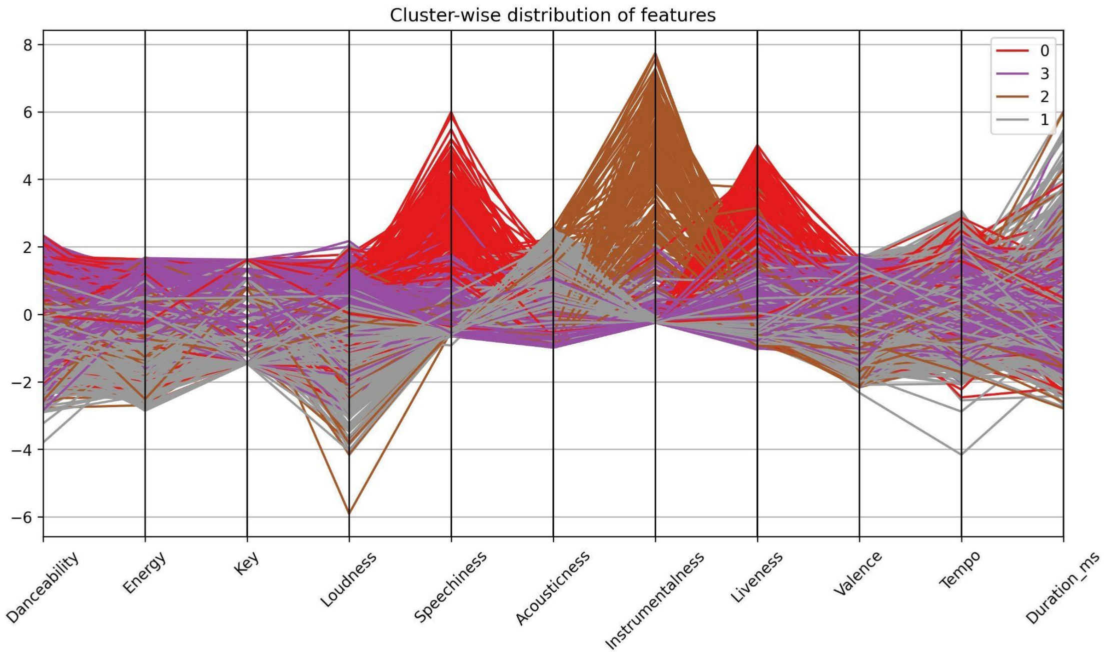
Findings - Clustering

- Cluster 0 represents higher concentration of **Humor and Memes** related comments
- Cluster 1 represents higher concentration of **Words of empathy** (deeply emotional) related comments as well **Personal Stories and Experiences**
- Cluster 2 represents all records which have relatively lower concentrations of all types of comments



Findings - Clustering

- Cluster 0 represents high concentrations of songs with **Speechiness** and **Liveness**
- Cluster 1 represents highest concentration of **Acousticness**
- Cluster 2 represents high concentration of **Instrumentalness**
- Cluster 3 represents **balanced distribution** across all clusters.



Findings - Recommender

Recommendation Basis:

- Uses **audio features** (e.g., danceability, tempo, energy) and **YouTube comments** (e.g., emotional response) for recommendations.

Clustering:

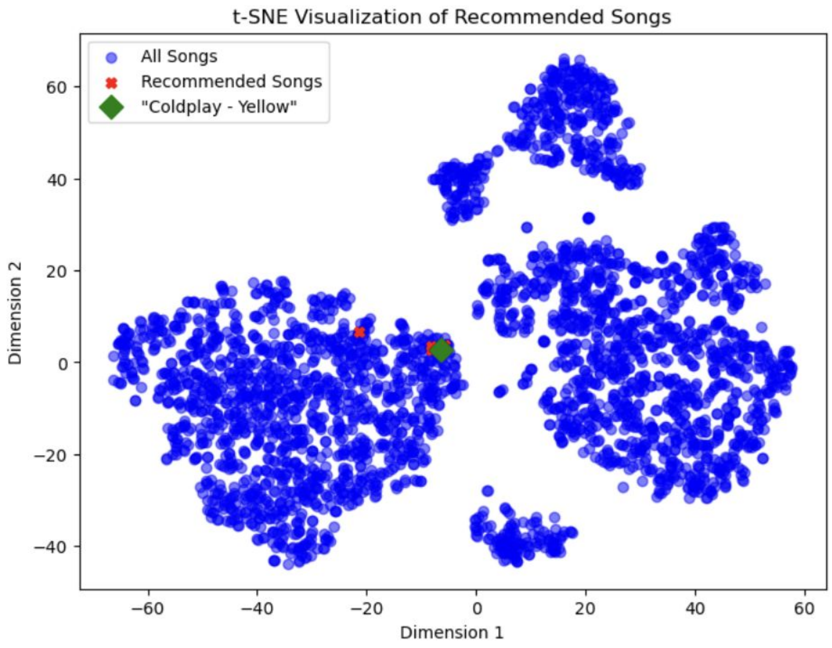
- Songs are clustered based on audio features and comments.
- 5 clusters are formed based on similarity in audio characteristics or emotional responses.

Similarity Calculation:

- **Euclidean Distance** and **Cosine Similarity** are used to calculate song similarity.
- Top 5 similar songs are recommended.

Findings - Recommender Example

Based on **"Coldplay - Yellow"**, songs with similar **audio features** or **emotional responses** are recommended.



Graph:

Blue dots: All songs.

Red Xs: Top 5 recommended songs based on similarity to "Coldplay - Yellow".

Green diamond: "Coldplay - Yellow".

Top 5 Recommended Songs:

Sia - Unstoppable, Luis Miguel - La Incondicional, Bryan Adams - Heaven, Toby Keith - Made in America, Creed - My Own Prison

	Album	Title	spotify_id
200	This Is Acting (Deluxe Version)	Sia - Unstoppable (Official Video - Live from ...	1yvMUkIOTeUNtNWIWRgANS
275	Busca Una Mujer	Luis Miguel - "La Incondicional" (Video Oficial)	6F9yAYUaNbUhdlQyt5uZ3b
958	Reckless (30th Anniversary / Deluxe Edition)	Bryan Adams - Heaven	7Ewz6bJ97vUqk5HdkvguFQ
2736	Clancy's Tavern	Toby Keith - Made In America (Official Music V...	7Lmwj2fe8MpGXypOuLGO2C
2840	My Own Prison	Creed - My Own Prison	5vRPXm59z8ewWO6WiJHg3m

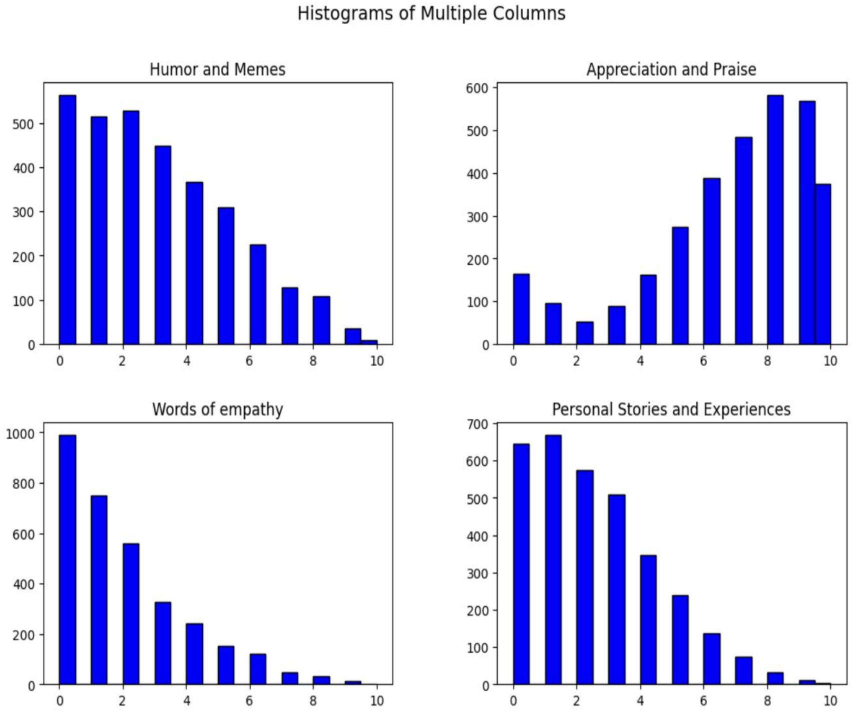
Findings - Multi-label/ Multi-class Classification

Multi-label Classification:

- **Definition**: Predicting **multiple labels** for each sample. Each sample can have more than one label assigned simultaneously.
- **Example**: A song can have multiple labels, such as **"Humor"** (comment category) and **"Energy"** (Spotify attribute), assigned at the same time.

Multi-class Classification:

- **Definition**: Predicting **one label** for each sample, where each sample belongs to only one class or label.
- **Example**: A song can belong to only one main label, such as **"Humor"** (comment category) or **"Energy"** (Spotify attribute), but not both at the same time.



Conclusion

Project Summary:

- Developed a recommendation system using **audio features** (e.g., danceability, tempo, energy), **user-generated comments**, and **lyric data**.
- Applied **multi-label** and **multi-class classification** techniques to assign multiple emotions (multi-label) and primary emotional tones (multi-class) to each song.

Key Takeaways:

- Combining **audio features**, **user-generated comments**, and **lyric data** provides a powerful framework for music recommendation.
- Using both **multi-label** and **multi-class classification** enables the system to offer diverse and personalized suggestions.

Appendix A: References

- Spotify URL & YouTube URL: https://www.kaggle.com/datasets/salvatorerastelli/spotify-and-youtube/data
- Spotify song attributes & lyrics: https://www.kaggle.com/datasets/bwandowando/spotify-songs-with-attributes-and-lyrics

Appendix B: YouTube Comments Category

- YouTube Comment Code Snippet

Appendix C: Lyrics Topic Modeling

- Lyrics Topic Modeling Code Snippet

Appendix D: Spotify Data Clustering

- Clustering Code Snippet

Appendix E: YouTube Data Clustering

- Clustering Code Snippet

Appendix F: Recommender

- Recommender Code Snippet

Appendix G: Multi-label/ Multi-class Classification

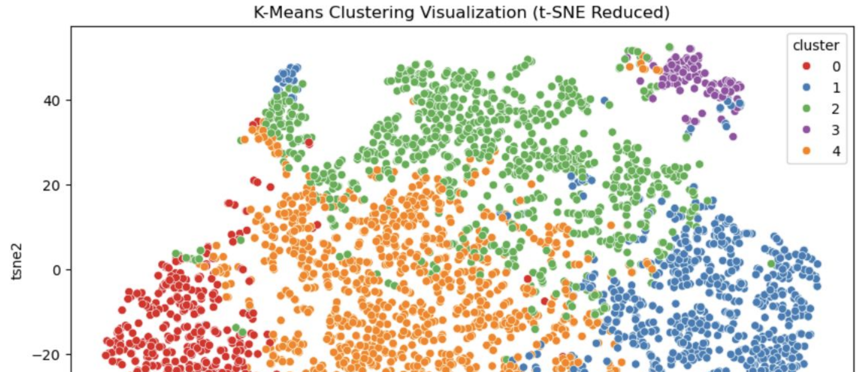
- Code Snippet

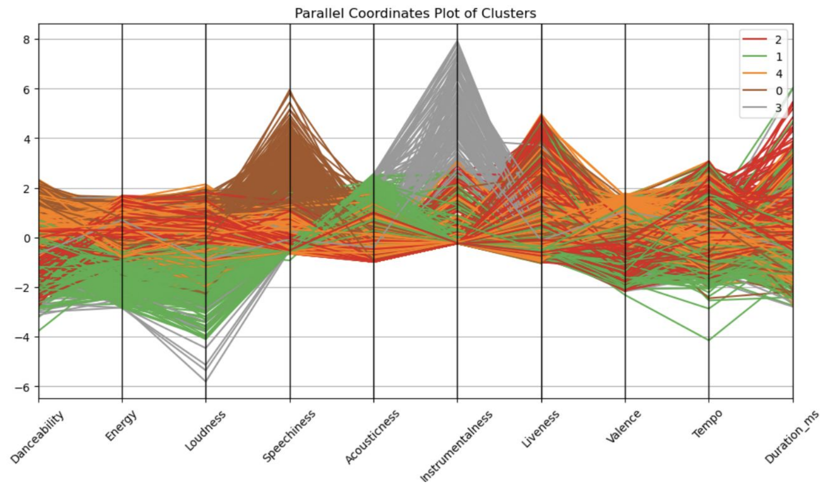
Appendix D: Library/ Frameworks used

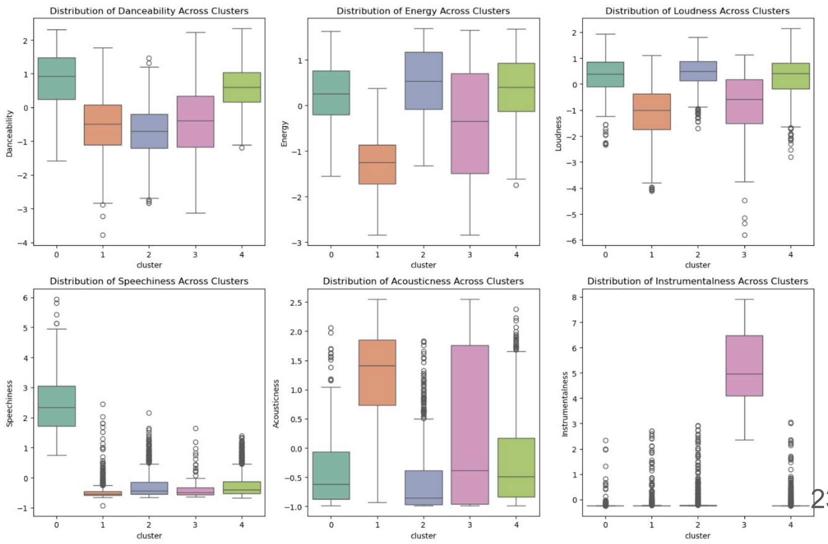
Findings - Audio Features

t-SNE visualization shows distinct clusters, indicating meaningful separ

- **Parallel Coordinates Plot highlights differences in feature distrib**
- **Boxplots reveal key feature variations, such as:**
  - 🎵 **Cluster 0** has higher **danceability**
  - 🎧 **Cluster 3** is characterized by **higher speechiness**
  - 🎼 **Cluster 2** has more **instrumentalness**
- **Findings can be applied to music recommendation systems**







Data

- Data from spotify > matrix 0 to 1 (audio feature) > have to normalize everything, url youtube
- Do EDA show duplicate data/ null value
- Disable video & … in list
- Order = relevance = top 10 comment > look for next page, keep track of video not getting comment > take spotify is, youtube id, comment, likes to new dataframe
- Same song but artists name different

Create Category for comment

- Use LLM > define category > ask open ai to

Challenge combining datasets

Category enhance by topic per song

Multilabel multiclass, clustering

Clustering > one group always dominate > exploring 1 comment group 2 audio feature (EDA)