Northeastern University, Khoury College of Computer Science

CS 6220 Data Mining - Final Project

Due: December 12, 2023(100 points)

Xiaoshu Liu - liu.xiaosh@northeastern.edu

Xinyue Han - han.xinyue@northeastern.edu

Hao Li - li.hao8@northeastern.edu

Siying Lu - lu.siyin@northeastern.edu

PROJECT GIT REPOSITORY

1 Introduction

Introduce us to your problem, why it's important and interesting, and why it's worth solving. Incorporate elements of your proposal, adding both your objective and motivation. Make sure you mention what impact your work will have and what your solution enables others to do. Describe some of the challenges. Finally, outline the remainder of your document. (5pts)

The Spotify music recommendation system aims to meet the growing demand for personalized music recommendations in the modern digital age. Due to the sheer volume of music on platforms such as Spotify, users often face challenges in discovering new music that matches their preferences. This problem is important because effective music recommendations can increase user satisfaction and engagement with music streaming platforms, driving the success of the music industry.

Our project is particularly interesting because it integrates data mining techniques, machine learning models, and API integration to create a sophisticated music recommendation system.

By leveraging Spotify's massive dataset, combined with features such as audio features and user preferences, our solution aims to cluster songs and recommend new tracks that match user input. This approach provides listeners with a richer, personalized experience, changing the way users interact with their vast music library.

Challenges include handling missing data, interpreting correlations in audio features, and ensuring the system adapts to dynamic user input. In addition, API limitations, such as restrictions on free accounts, further complicate the development process. This report outlines the motivation, background, methods, results, and conclusions of our project, providing a comprehensive overview of our work.

2 Background

This section should describe how this problem arises, who is affected, and any literature that you plan on reviewing or survey. These references should relate to both the applications of the work as well the technical approaches to be taken. Emphasize why they are appropriate and relevant. (5pts)

The music recommendation problem stems from the difficulty of navigating large datasets to provide meaningful personalized recommendations. This problem affects millions of users who rely on streaming platforms to discover music. Spotify, one of the largest music platforms, provides an opportunity to leverage its rich dataset for clustering and recommendation.

This project draws inspiration from existing recommendation systems, such as those used by Netflix or YouTube, which employ collaborative filtering, content-based filtering, and hybrid models. For the technical implementation, we explored clustering techniques such as K-Means and dimensionality reduction methods such as PCA. The literature on music machine learning, including works on audio feature extraction and similarity measures, provided references for our approach. These references emphasize the importance of scalability and interpretability in music recommendation. By leveraging Spotify's data and API, we ensure that our system meets user needs and state-of-the-art practices.
Below is a list of literatures relevant to this project:

- **Current challenges and visions in music recommender systems research**: The paper reviews current challenges and advances in music recommender systems (MRSs), emphasizing the difficulty of integrating listener needs and preferences beyond basic user-item data. It surveys state-of-the-art methods, discusses their limitations, and highlights potential research directions for developing more user-centered systems.

- **Enhanced Music Recommendation Systems: A Comparative Study of Content-Based Filtering and K-Means Clustering Approaches** - This study explores a comparative approach using both K-means clustering and content-based filtering to develop effective recommendation systems. It identifies that K-means clustering, despite its value for grouping similar songs, may be less precise without extensive feature engineering. This insight reinforces the importance of PCA and data pre-processing in this project.
- **Analysis of machine learning-based music recommendation system using Spotify datasets**: This paper highlights PCA's effectiveness in transforming high-dimensional music features into orthogonal components, enhancing model stability and interpretability. Here, the combination of PCA and K-means clustering is particularly useful for creating clusters based on nuanced audio features.

3 Approach

3.1 Data and Data Analysis

3.1.1 Data

3.1.1.1 About the data

The Spotify dataset from Kaggle was used in our project. It contains 5 csv files - data.csv, data_by_artist.csv, data_by_genres.csv, data_by_year.csv, data_w_genres.csv. We mainly use data.csv and data_by_genres.csv for our model. It contains over 170,000 rows of data describing songs through a variety of audio features, metadata, and popularity metrics. Each song is characterized by features such as danceability, energy, acousticness, instrumentalness, and valence, which quantify musical qualities, along with attributes like tempo, key, and duration_ms. The dataset also includes metadata such as the song's name, artist, and release year, enabling analysis of trends over time.

The dataset contains music from the year 1921 to 2020. For more recent songs released after 2020, we implemented Spotify API to retrieve the music.

3.1.1.2 Exploratory Data Analysis

The song data spans from 1920s - 2020s, with the most updated music in 2020, as seen in figure 1. Most songs from the dataset are in 1950s to 2010s.

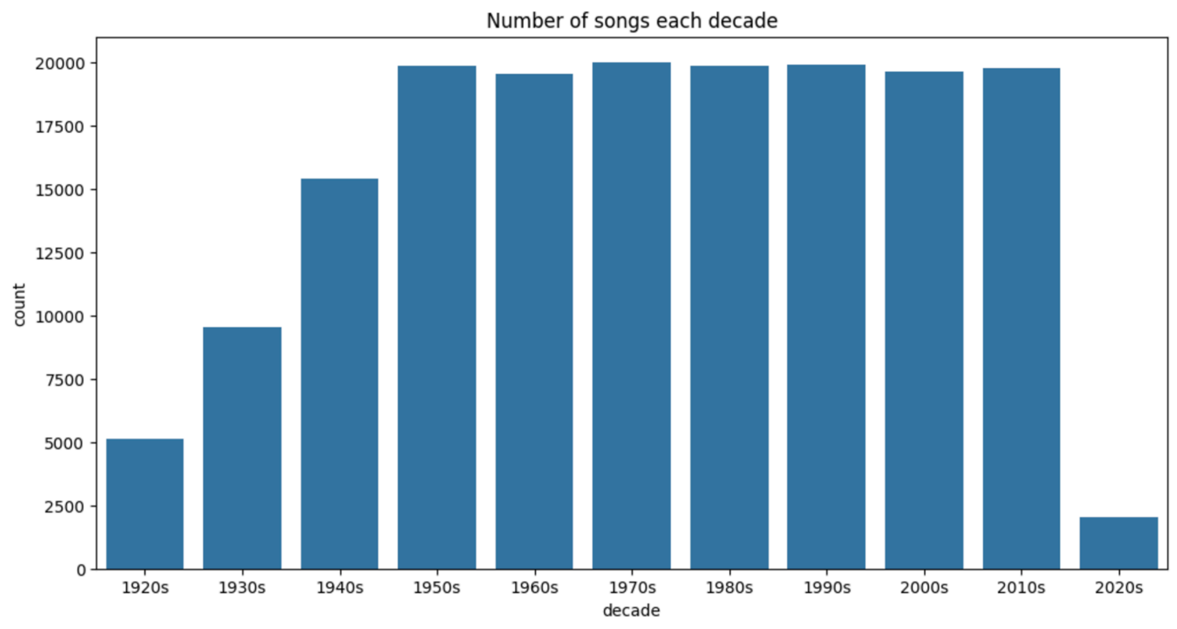

Figure 1. Number of songs each decade in data.csv

We can also see the song features, such as acousticness, danceability and energy, evolved over time (Figure 2). The graph highlights a clear transition in musical trends, with an increasing focus on electronic production (lower acousticness, higher energy) and vocal-centric music (lower instrumentalness). The increasing danceability aligns with the dominance of dance-friendly genres in contemporary music.

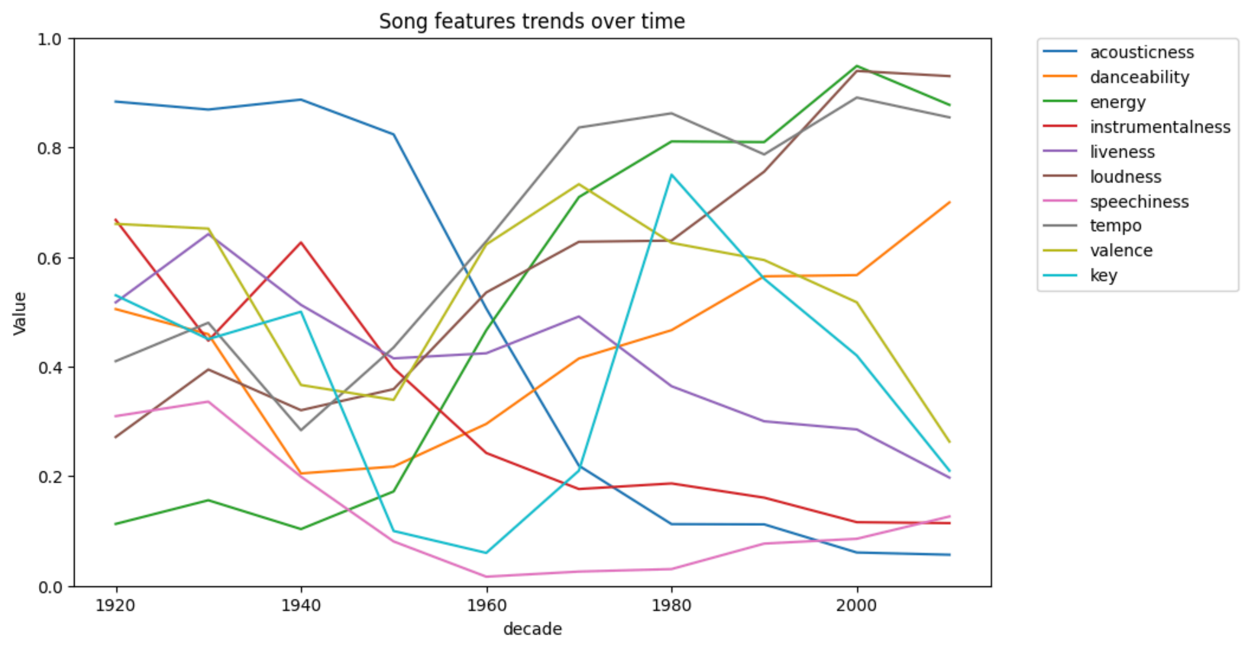

Figure 2. Song features over time (normalized)

Next, the feature correlation was studied. Several strong positive correlations were observed. For example, energy and loudness has a correlation of 0.78, which indicates loud tracks tend to be more energetic, reflecting the relationship between audio intensity and perceived energy. Also, valence and danceability has a correlation of 0.56, indicating songs with a positive mood (higher valence) are generally more danceable. On the other hand, acousticness has a strong negative relation with energy (-0.75) and loudness (-0.71), which means acoustic songs tend to have lower energy levels and are generally quieter. These correlations will be helpful for dimensionality reduction in the downstream.

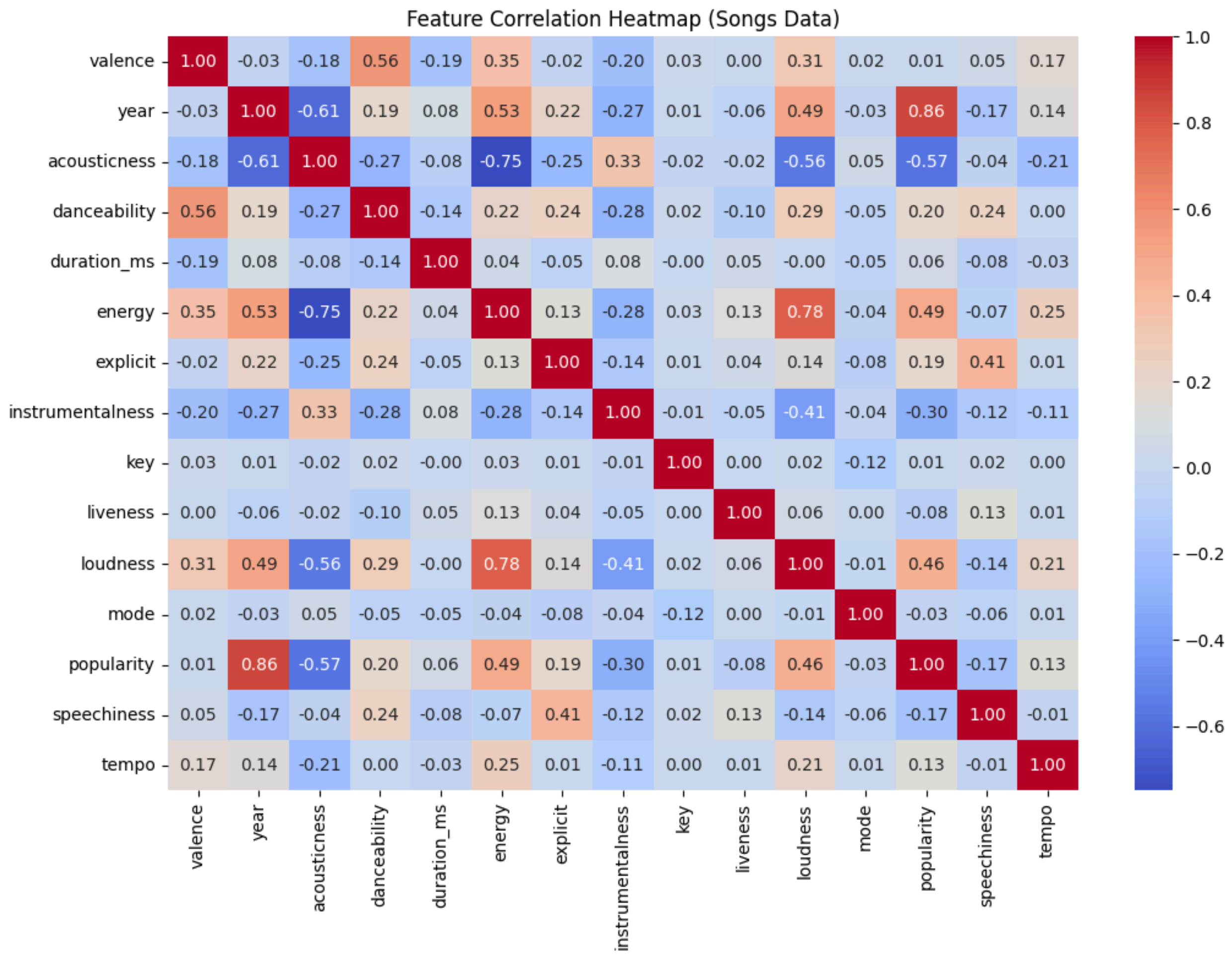

Figure 3. Feature Correlation in data.csv

Below in figure 4 is another visualization of correlations between song features and popularity. Popular songs are typically newer, high-energy, loud, and danceable. Conversely, acoustic, instrumental, or speech-heavy tracks tend to be less popular.

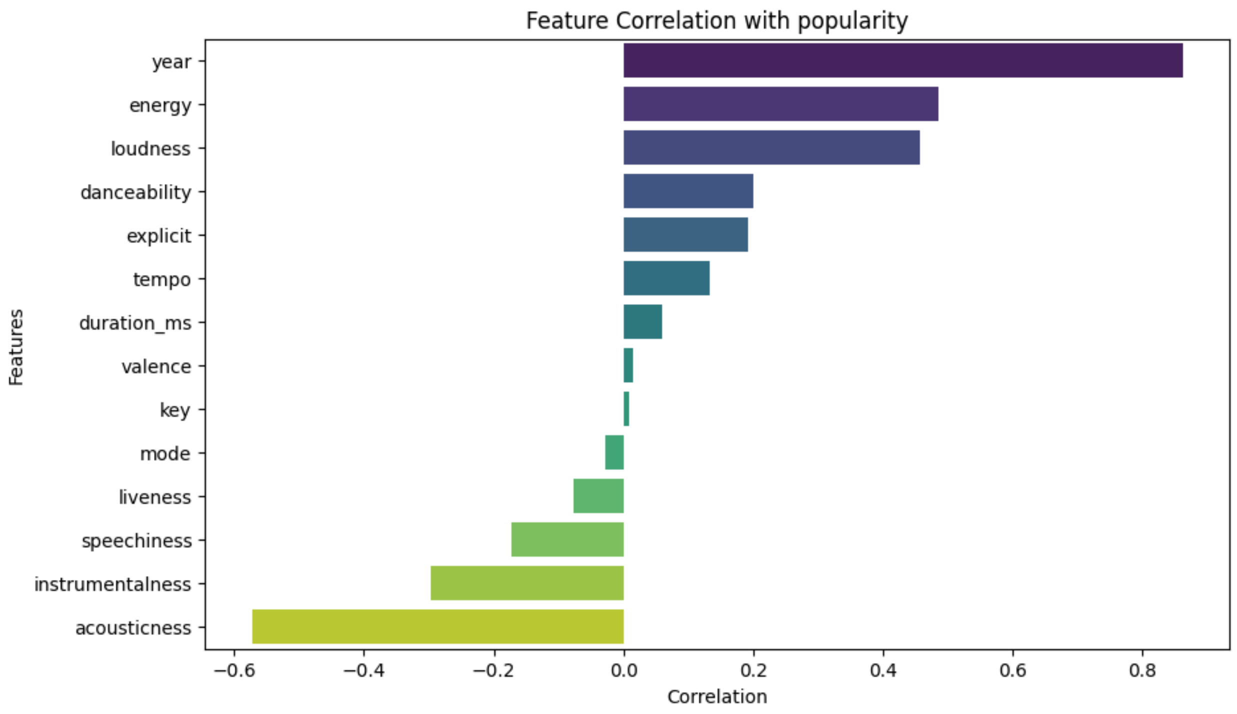

Figure 4. Feature correlation with popularity

3.1.2 Feature Selection

Since there are 19 features in the original data.csv, which would be too many for training the model and might cause overfitting issues. We selected 12 out of 19 most relevant features to a song and performed Principal Component Analysis (PCA) prior to performing the k-means clustering.

For the 12 selected features, a standardization using the StandardScaler was used prior to the PCA. The number of components was chosen to be 10 to retain the 95% variance.

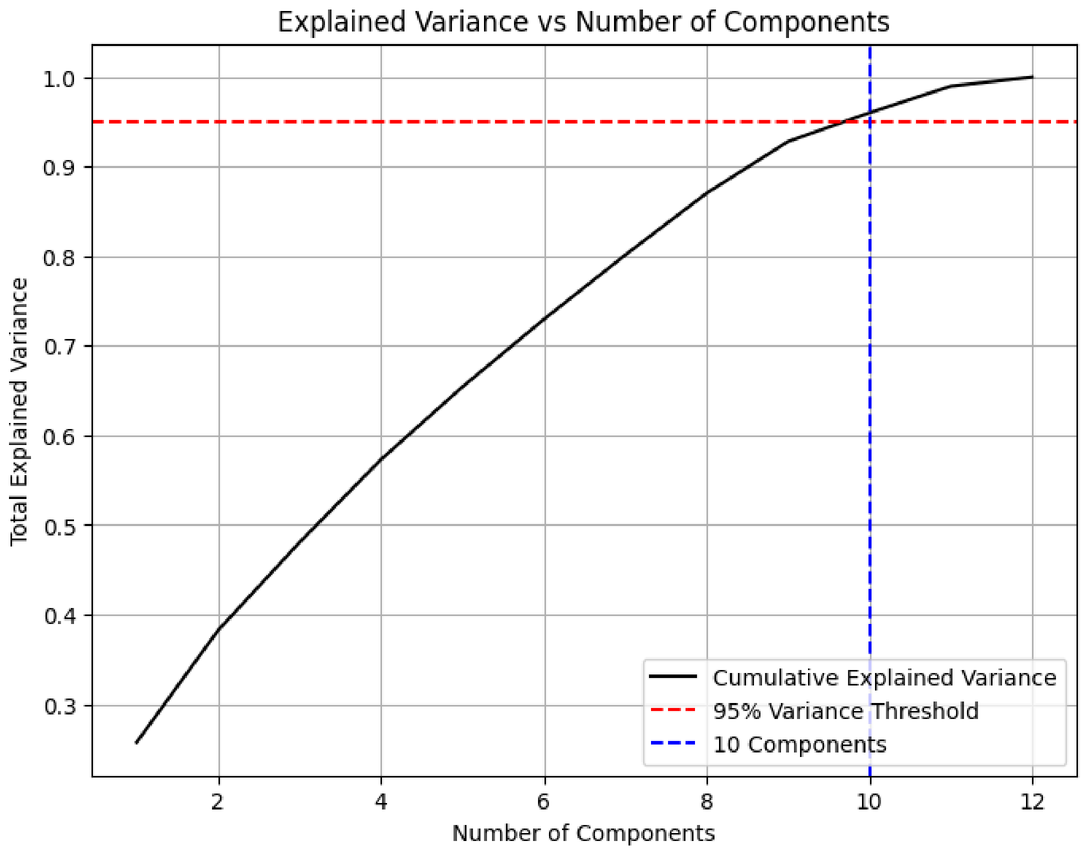

Figure 5. Explained Variance vs Number of Components

3.2 Implementation

3.2.1 Data Preprocessing

Objective: Prepare the dataset for analysis and modeling by handling missing values and standardizing features.

Steps:

1. Data Loading: Import datasets to perform analysis at song, genre, and year levels.
2. Feature Engineering: Add derived features such as decade for historical trends and normalize key audio features (acousticness, danceability, etc.) using MinMaxScaler for consistent scaling.

3. Cleaning and Handling Missing Data: Fill missing values in numeric columns with the mean, ensuring clean data for clustering and dimensionality reduction.

3.2.2 EDA

Objective: Analyze feature distributions and relationships to understand the data better.

Steps:

1. Visualize the number of songs by decade to observe historical distribution.
2. Plot song features over time to identify trends in audio characteristics (e.g., increasing energy, decreasing acousticness).
3. Generate a correlation heatmap to explore relationships among features, focusing on the target (popularity).
4. Use a barplot to highlight feature correlations with popularity for feature prioritization.

3.2.3 Dimensionality Reduction Using PCA

Objective: Reduce the dimensionality of audio features while retaining 95% of the variance.

Steps:

1. Standardize numerical features using StandardScaler.
2. Apply PCA to reduce the dataset to principal components that explain 95% of the variance.
3. Visualize the Explained Variance vs. Number of Components to determine the optimal number of components.
4. Analyze Principal Component Loadings to understand feature contributions to each component.

3.2.4 Clustering with K-Means

Objective: Group similar songs into clusters based on PCA-reduced data.

Steps:

1. Use the Elbow Method to identify the optimal number of clusters by analyzing inertia values. See figure 6.

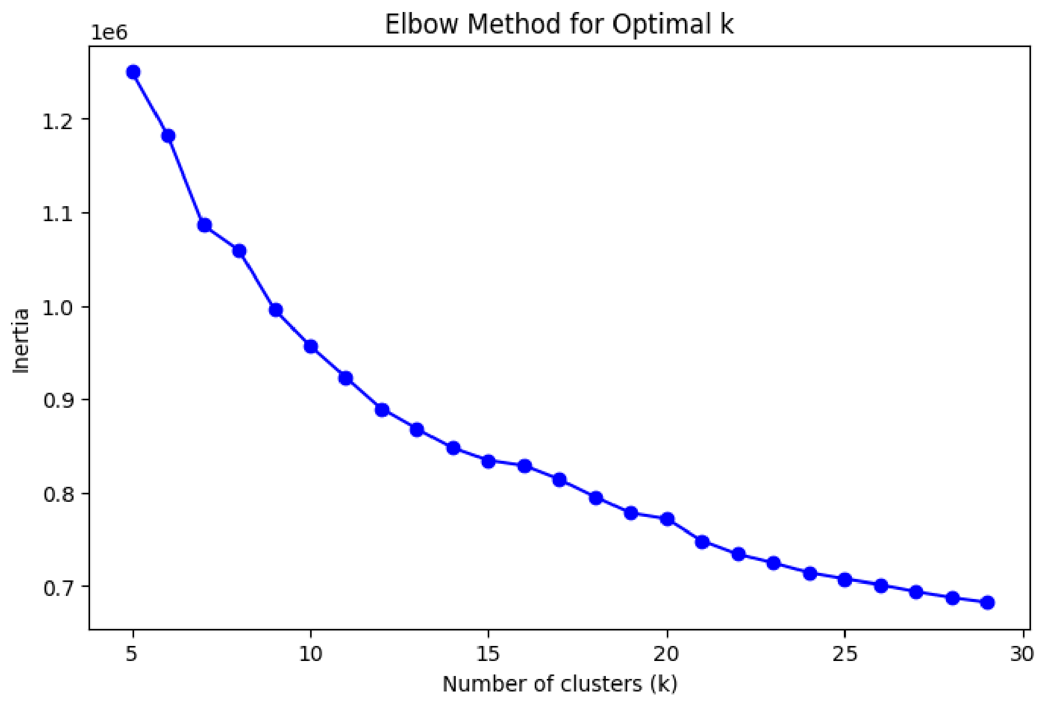

Figure 6. Optimal k

2. Perform K-Means Clustering with the selected number of clusters (k=8). See cluster visualization in figure 7.

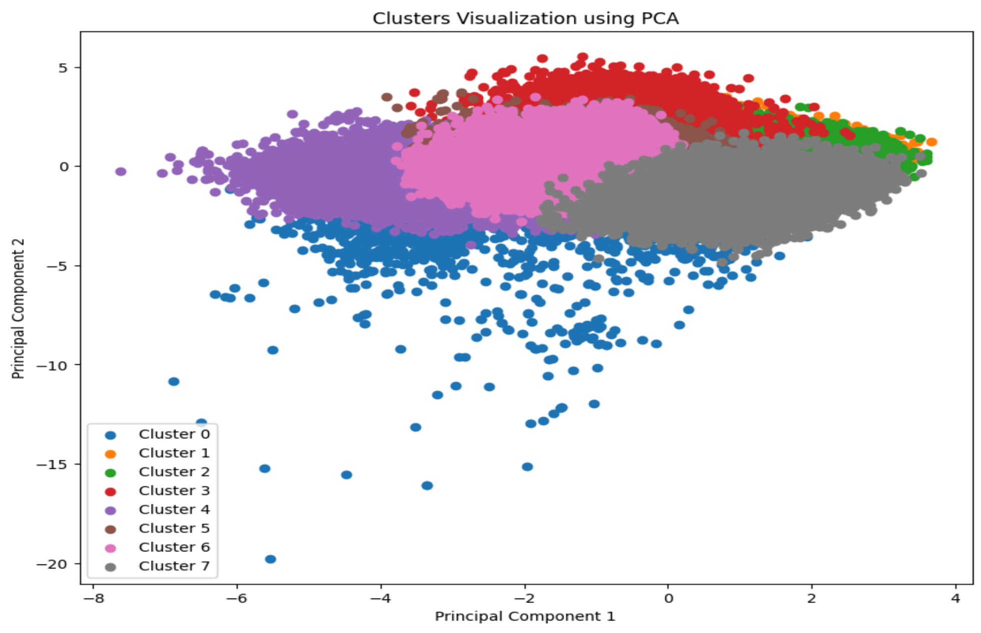

Figure 7. Clusters visualization with PCA (k=8)

3. Assign cluster labels to each song, enabling cluster-based recommendations.

3.2.5 Recommendation System Implementation

Objective: Recommend songs based on user input using the trained K-Means model.

Steps:

1. Accept user-provided songs with names and years as input.
2. Retrieve audio features for input songs, dynamically fetching missing data from Spotify's API if user input could not be found in the dataset.
3. Use PCA-transformed data and the K-Means model to identify clusters corresponding to input songs.
4. Randomly choose songs from the same clusters for recommendation, ensuring diversity and relevance.
5. Test recommendations and evaluate their average popularity using Spotify's API.

3.2.6 Enhanced K-means with Genre Integration

Objective: Improve clustering and recommendations by incorporating genre information.

Steps:

1. Classify each song into a genre based on its proximity to genre-level data.
2. Encode the genre labels and include them as an additional feature for clustering.
3. Re-run PCA and K-Means clustering with the enhanced feature set.
4. Test recommendations and evaluate their average popularity using Spotify's API.

To ensure model correctness, PCA was used to retain 95% of variance, reducing noise and redundancy while preventing overfitting. The Elbow Method determined the optimal number of clusters for K-Means, avoiding under- or overfitting. Features were normalized to ensure equal contribution, and PCA-transformed components emphasized the most impactful data. The model's performance was validated through cluster visualizations and tests on various user inputs, ensuring accurate and generalized recommendations.

4 Results and Evaluation

Well-Reasoned Evaluation Clearly define the evaluation metrics you're using and why. Properly evaluate and draw conclusions. Determine the confidence you have in the outcomes of your processing and modeling. (10pts)

To evaluate the performance of our system, we used several key metrics:

- Cluster interpretability: Using K-Means clustering on principal components derived from audio features, we retained 12 components after applying PCA. The Elbow Method identified 8 optimal clusters, with distinct groupings visible in a 2D PCA plot, confirming that the clustering captured meaningful distinctions in musical characteristics.
- Recommendation quality: Input songs such as *Cruel Summer* (2019) and *Love Story* (2008) yielded diverse recommendations spanning different artists and genres. The enhanced model, which incorporated genre information, improved personalization and diversity, addressing both mainstream and niche user preferences.
- Popularity metric: The average popularity score for the basic model was 44.3, reflecting moderate recognition among Spotify users. The enhanced model, incorporating genre-based clustering, yielded a slightly lower score of 37.2, suggesting a shift toward recommending less mainstream but potentially more personalized tracks. This trade-off highlights the model's adaptability to diverse user preferences.

The results show that the eight groups of clusters can capture meaningful patterns in song features such as energy, tempo, and price. The enhanced system includes genre information to further improve recommendations by placing songs in similar styles. Cluster visualizations in the PCA reduced space highlight clear distinctions between groups. The average popularity of recommendations indicates a balance between mainstream and niche tracks, demonstrating the versatility of the system.

Our confidence in the results stems from the strong correlation between song features and popularity, as shown in the heatmap and feature importance plots. However, limitations such as incomplete API responses and dataset bias affect generalizability.

5 Conclusions

Discuss what you've learned. How do you think it can be applied? If you had more time, what are the future directions? If there's literature on future directions, include them. (5pts)

Through this project, we learned the power of combining clustering, dimensionality reduction, and API-driven real-time data to build recommender systems. Our solution can effectively group songs by similarity and dynamically adjust based on user input to provide relevant and personalized recommendations.

Future applications include extending the system to support collaborative filtering, integrating user listening history, and incorporating feedback loops to refine recommendations over time. Incorporating sentiment analysis into user reviews or lyrics can further enhance the model over time. The literature on hybrid recommender systems shows that content-based approaches can be combined with collaborative filtering to achieve greater personalization.

This project demonstrated the potential of data mining techniques to transform the user experience on music streaming platforms, paving the way for smarter and more intuitive recommender systems.
Below is a list of literature on future directions:

User-Centric and Collaborative Filtering:

Koren, Y., Bell, R., & Volinsky, C. (2009). *Matrix factorization techniques for recommender systems*. IEEE Computer, 42(8), 30–37.

- Discusses collaborative filtering and matrix factorization for scaling recommendations while addressing the cold-start problem.

Genre-Specific Recommendations:

Knees, P., & Schedl, M. (2016). *Music similarity and retrieval: An introduction to audio- and web-based strategies*. Springer.

- A detailed discussion on leveraging genre-specific clustering and user interaction for improving music recommendation accuracy.

Incorporating Sentiment Analysis:

Goh, K.-Y., Heng, C.-S., & Lin, Z. (2013). *Social media brand community and consumer behavior: Quantifying the relative impact of user- and marketer-generated content*. Information Systems Research, 24(1), 88–107.

- It highlights how sentiment analysis of user-generated content, like reviews, can guide personalization in recommender systems.

6 Submission Guidelines

Upload your PDF file and Github URL to Gradescope as a *single* submission with all members of the team. On the top of the document, include the teammates and project repository. *Try to keep your proposal below three pages.*