

**Project Report**

**EWU\_CSE303\_FALL’24**

**Personalized Music Recommendation System**

[**https://colab.research.google.com/drive/1zFVRTVtmvJ2ievPeB95hmJc0fXYfXO1T?usp=sharing**](https://colab.research.google.com/drive/1zFVRTVtmvJ2ievPeB95hmJc0fXYfXO1T?usp=sharing)

**Course Title :** Statistics for Data Science

**Course Code :** CSE303

**Section :** 01

**Instructor :** Dr. Mohammad Rezwanul Huq

Associate Professor,

Department of Computer Science & Engineering.

**Submitted by:**

|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | **Contribution (%)** |
| **Abrar Hossain Zahin** | **2022-2-60-040** | **40%** |
| **Jannatul Ferdous Prome** | **2022-2-60-031** | **25%** |
| **Antara Labiba** | **2021-3-60-173** | **10%** |
| **Raima Husna** | **2022-2-60-052** | **25%** |

1. **Introduction**

With the increase in available songs on music streaming platforms it has enabled users to have more choices, which makes it difficult for them to find the songs they want. The goal of this project is to develop a customized music recommender system with the help of streaming service Spotify’s dataset and machine learning so that songs may be recommended to users based on their individual preferences and the attributes of each song. The approach used in this project will focused on the analysis of acoustic features such as danceability, energy, tempo, and valence in the music to determine the patterns within the songs. The system will use content-based filtering to find users who have similar aurorial profiles, and collaborative matric filtering to refine filtering based on users’ profiles. Moreover, K-Means or other clustering techniques will be deployed for indexing to categorize songs meaningfully. Using a web-based interface, users can provide information about their mood, tempo or even their favorite artist and receive curated recommendations. Data visualization tools will shed light on popular trends in music listening. The project success will be examined using common metrics, which include Precision, Recall, and Mean Average Precision (MAP). By leveraging data, what goals this project seeks to accomplish is building a powerful system that will help improve music recommenders while remaining user friendly.

1. **Data Processing**

* + **Missing Values Management**

Values that are missing in the dataset will now be filled with the mean of each numeric column. This policy helps with data coherence without severe distortions. Imputation is done by mean values in order to prevent bias that may be introduced with other methods of filling in absent values.

* + **Duplicates Removal**

In a venture to mitigate redundancy, duplicate entries in the dataset are eliminated. Failure to delete duplicate records can lead to particular data records being overrepresented and this could affect model training. Removing them ensures that every data entry has a unique representation in the analysis.

* + **Normalization of Data**

Different numerical features within the dataset must be normalized. This is because these features have different ranges such as how loudness is given in decibels and tempo in beats per minute. This prevents dominance of larger numbers during model training.

❖ **Categorical data encoding**

Columns of a categorical nature such as “genre” will be transformed into numbers for purposes of enabling machine learning algorithms to use them. Encoding methods convert data into a categorical sub-form while removing any doubts that could arise from the way models perceive the data.

* + **Reducing Dimensions**

The features in the dataset could be excessively large and some may be redundant. PCA(Principal Component Analysis) is employed to combat this problem.

1. **Dataset Characteristics and Exploratory Data Analysis**

# Explanation of the Dataset

The dataset has many files, each relating to a basic or detailed level of music data. As such, for this project, the following datasets are included:

* Data at the Track Level: or data.csv– This superset has information about individual songs such as tempo, energy, and danceability.
* Data at the Genre Level: data\_by\_genres.csv – This file groups songs by genre, and has some general statistical information.
* Trends by Year: data\_by\_year.csv – This file shows how the characteristics of music have changed over years.
* Date at Artist Level: artist.csv – This file is focused on songs at the artist level to understand the style of music and popularity of the artist within the context of the music.
* Genre Data w.genre.csv – Likely an enriched genre dataset, but it is unclear if it has any additional sub-genres or other types of classification.

After looking at these datasets, we can try to figure out what makes a song popular, the differences between genres and how the genre of music has changed over the years.

Structure Of The Dataset : Number Of Rows, Columns, And Features

To get the data, we will first look into the size as well as the structure of the data und its organization:

* Number of Rows And Columns

Every dataset has its unique size, however the track level dataset has record level information for each song available, the most While the genre and year datasets are more summarized and capture information about the trends at a broader level. The artist dataset is more specific and statistical, it has information as recorded by the individual artist, which we examine against the performance of the artist.

Types of Features:

* Numerical Features

Loudness, tempo, energy, danceability, instrumentalness, duration, etc.

* Categorical Features

Genre, artist, year, mode, key.

* Target Variables

Popularity scores (to measure the performance of the song).

Feature Summary:

* Danceability:

Suitability of a song for dancing (on a scale of 0–1).

* Energy:

In terms of intensity, more energetic songs get a higher value.

* Loudness:

Volume of the track.

* Tempo:

The beats per minute (BPM) gives an idea of how fast the song is.

* Instrumentalness:

The probability of a track being instrumental. The higher the value, the less likely there will be vocals.

* Popularity:

A score assigned to indicate how popular a song is likely to be. Often streaming counts are the basis. When we understand the structure of the dataset, this aids us in selecting the appropriate pre-processing methods and modeling techniques.

# Significance of Correlation Analysis

Correlation analysis was critical in establishing the relationships that existed between the features which assisted in feature selection and understanding important aspects of music.

High correlation among variables may lead to redundancy, whereas low correlation may identify different drivers to the popularity of the song.

# Important Relationships in the Dataset

Loudness & Energy: They are strongly positively correlated because usually louder songs feel more energetic.

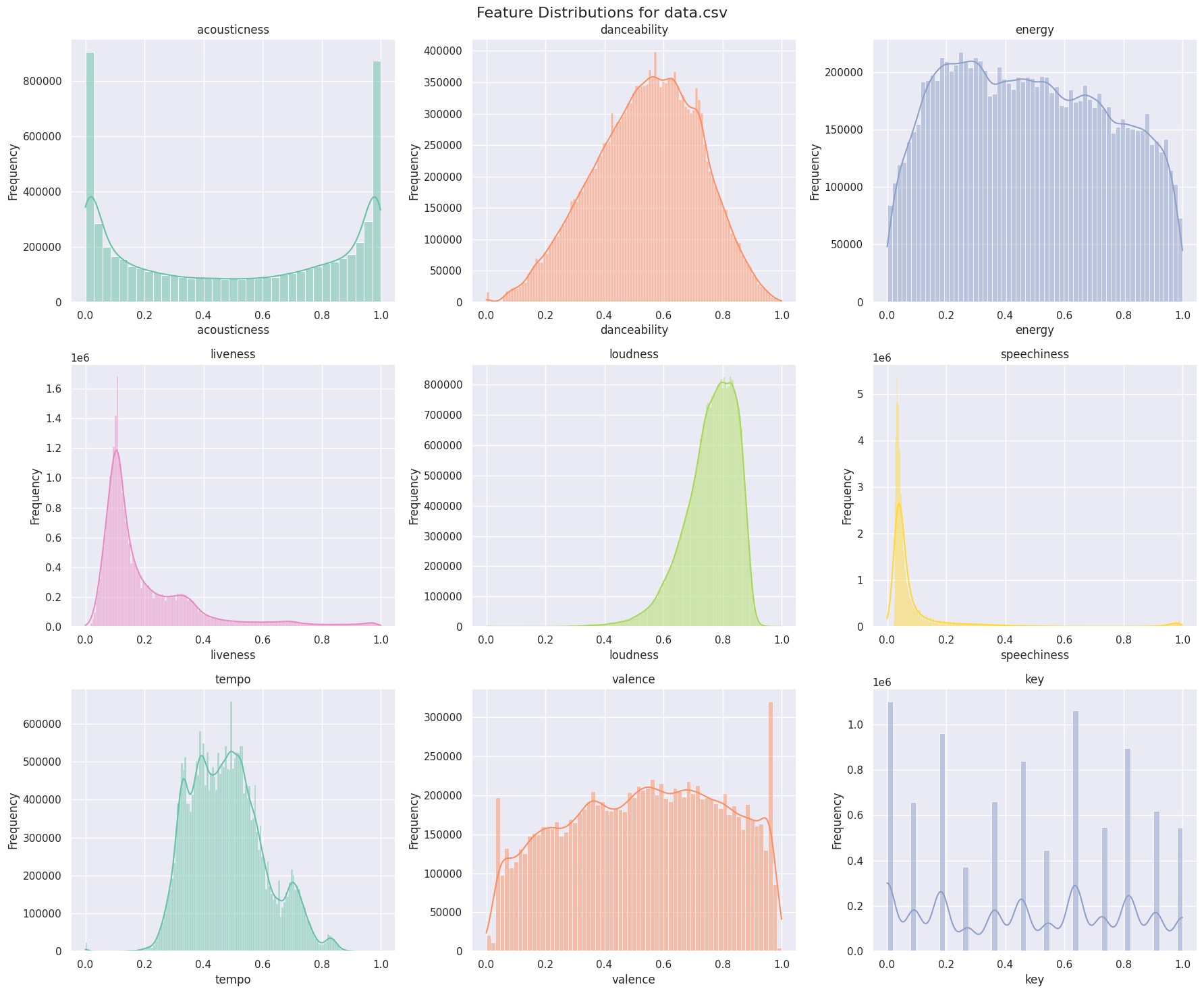
Danceability & Tempo: They are somewhat correlated. However, fast beats do not always increase the danceability of a song.

Popularity & Instrumentalness: Inverse correlation is likely, as the most popular songs are rarely purely instrumental.

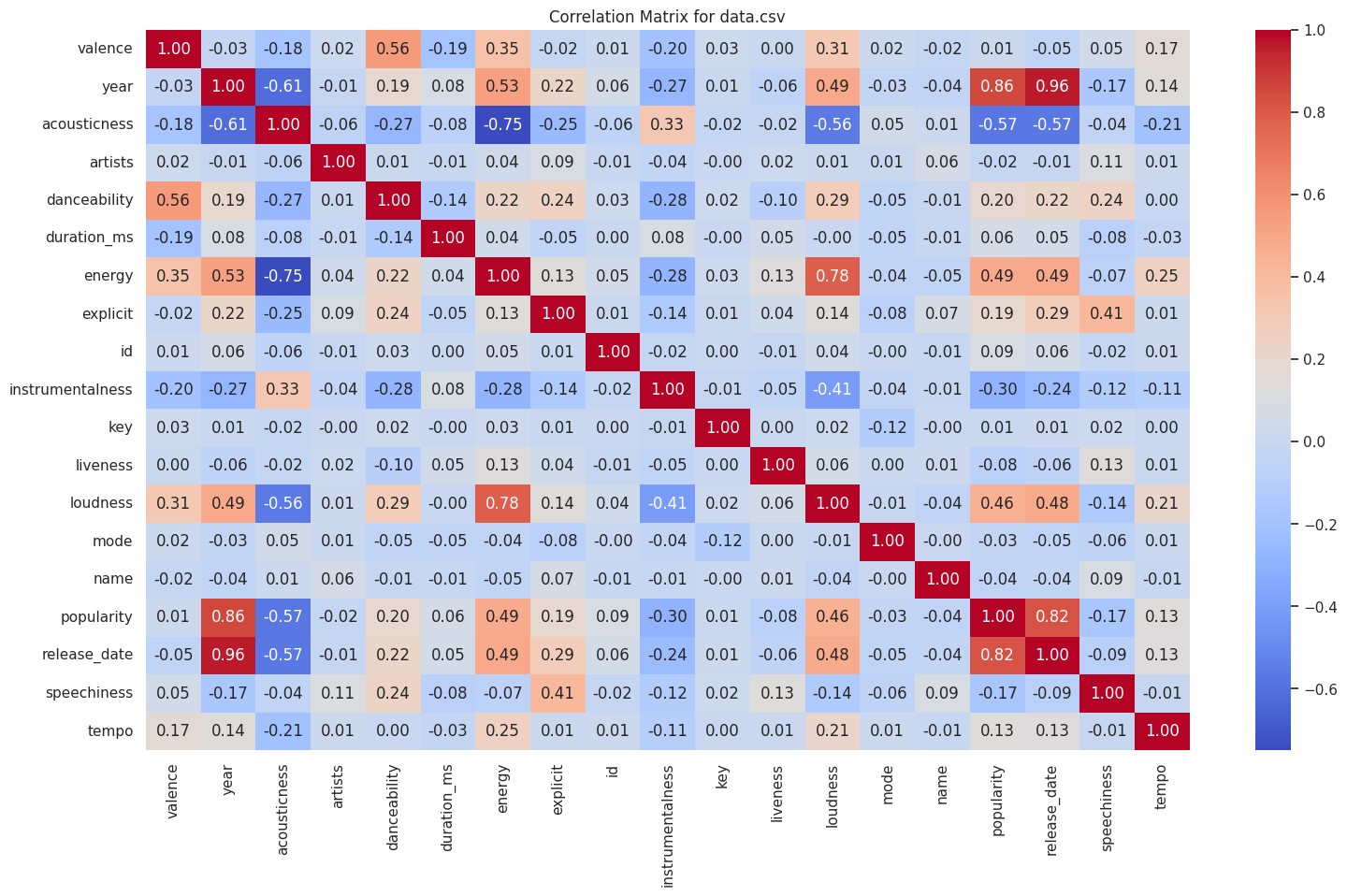
Genre Trends: Pop, and hip hop have higher danceability than classical and jazz which exhibit lower energy.

Artist Trends: There are artists who are known to always produce high-energy songs compared to those who blend different styles.

# **Histogram**



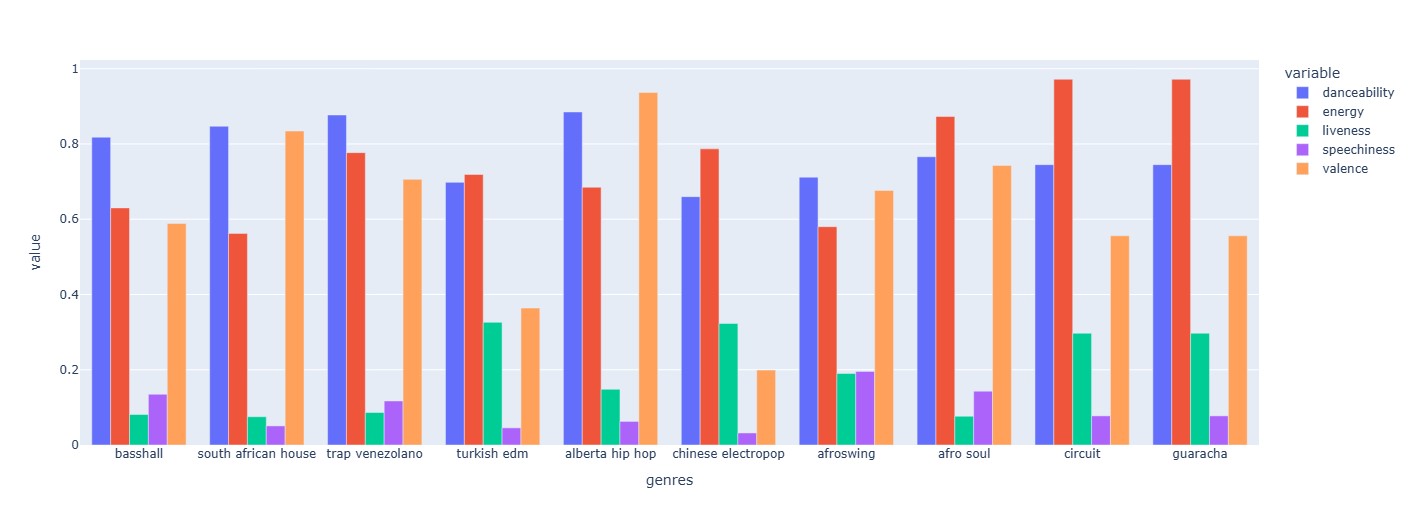
**Correlation Using HeatMap**



* **Strong Positive Correlations:** 
  + Energy & Loudness (0.78): High-energy tracks tend to be louder.
  + Popularity & Release Date (0.82): More recent songs are generally more popular.
  + Danceability & Valence (0.56): Happier songs are often more danceable. ❖ **Strong Negative Correlations:**
  + Energy & Acousticness (-0.75): Acoustic songs tend to have lower energy.
  + Popularity & Acousticness (-0.57): Popular songs are generally less acoustic.
  + Loudness & Acousticness (-0.56): Quieter songs tend to be more acoustic.
* **Moderate Trends:** 
  + Year & Popularity (0.86): Newer tracks tend to gain more popularity.
  + Danceability & Acousticness (-0.27): Highly danceable tracks are usually less acoustic.

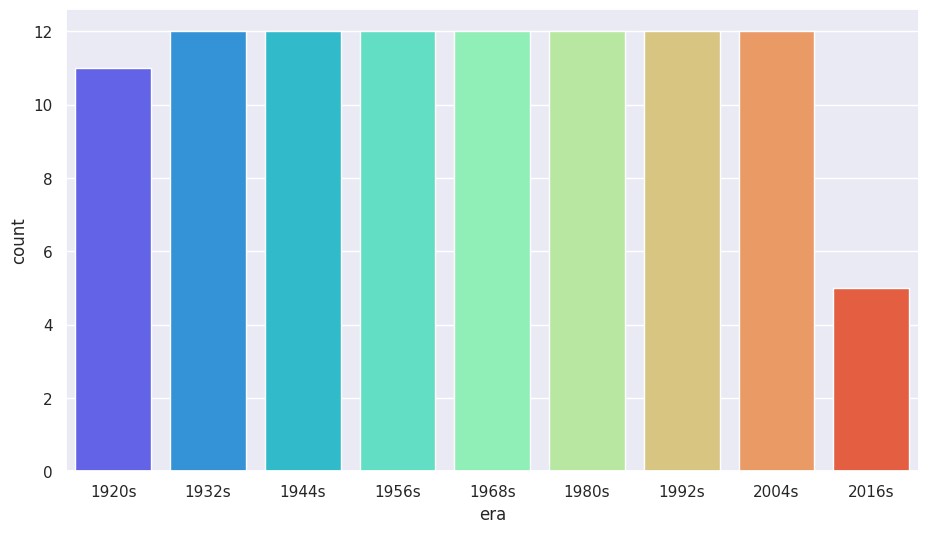
The analysis suggests that modern, loud, energetic, and less acoustic songs tend to be more popular. Danceable and happy (valence) tracks also correlate positively, while acoustic and quiet tracks generally have lower popularity and energy.

# **Top Genres**



The graph is that the top 10 most popular music genres exhibit distinct patterns in their danceability, energy, liveness, speechiness and valence. Danceability and energy tend to be consistently high across genres, while liveness and speechiness show more variation. Certain genres, like "afro soul" and "guaracha," exhibit very high energy levels, whereas "trap venezolano" and "alberta hip hop" show strong valence, indicating positive emotional tones. The graph highlights how different musical attributes contribute to a genre’s popularity.

# **Musical Trends Across Eras**



This bar chart is to showcase the distribution of music records based on their frequency (sound characteristics) across 12-year intervals, grouped into distinct eras. By visualizing these intervals, the chart highlights how music production trends and sound frequencies have evolved over time.

* Era-Specific Trends: Some eras may show higher frequency counts, indicating a surge in music production or popularity of certain sounds during those periods.
* Evolution of Music: The comparison between intervals reveals shifts in musical styles and the dominance of certain sound profiles in different decades.

## 4) Feature Engineering and Clustering Systems

Feature engineering improves the performance of machine learning algorithms by modifying unprocessed data sets step-by-step including dealing with missing data, transforming variables, modifying the scale of variables, and even reducing the number of dimensions in the data set.

Missing Values: If the data is not evenly spread, the mean is used, else the median is used. Modes are used to replace categorical values so that the remaining data is consistent.

Data Encoding: Label Encoding makes it possible to cluster data by translating categories into numbers.

Feature Scaling: MinMax Scaling is essential since the ranges are limited between 0 and 1 for K-Means clustering. Standard Scaling is helpful when working with PCA because it reduces the risk of biases that cause unequal emphasis on features.

Dimensionality Reduction (PCA): PCA strives to decrease computation through effective clustering and at the same time attempts to keep important portions of the data.

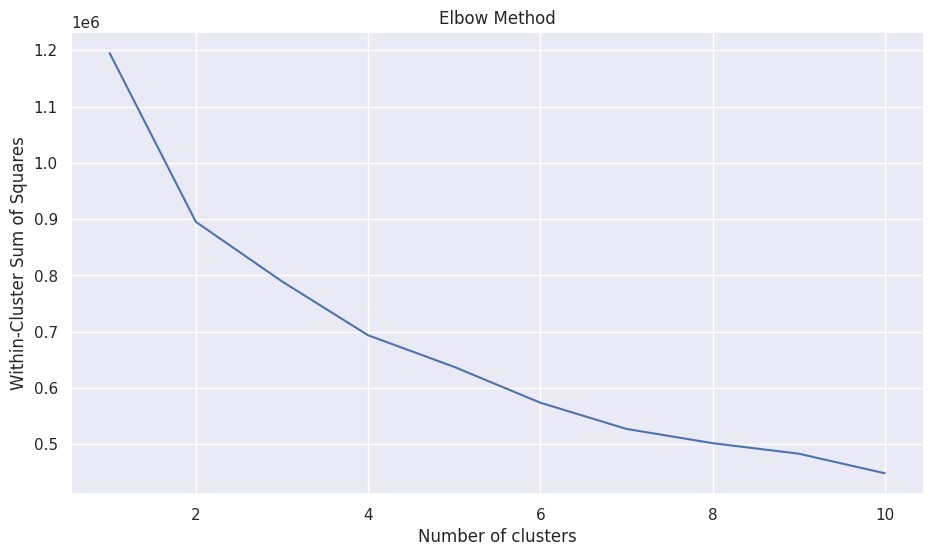
Clustering Systems : Unsupervised clustering groups like songs into different classes using tempo, energy, and danceability.

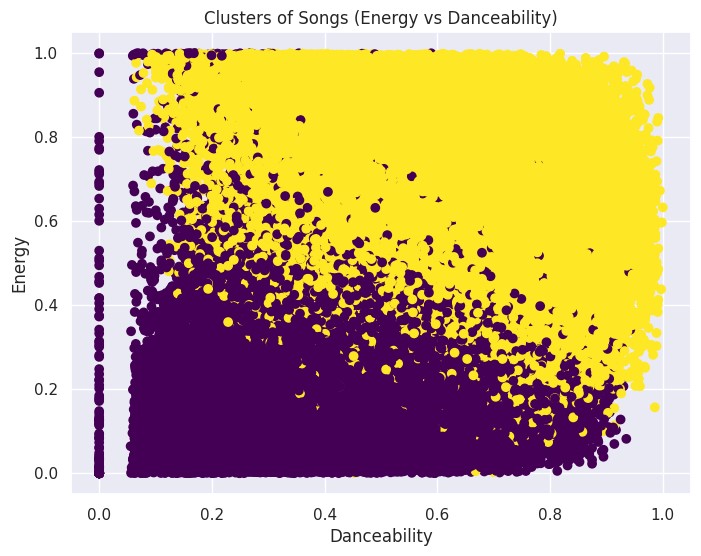
K-Means Clustering: Songs are placed into K clusters in accordance with Euclidean distance. The optimal number of clusters is computed using the elbow method.

Support Vector Machine (SVM) for Clustering: Support vector machine clusterers have the advantage of fine-tuning the borders of the clusters by determining the supporting points, particularly in situations where the clusters are skewed.

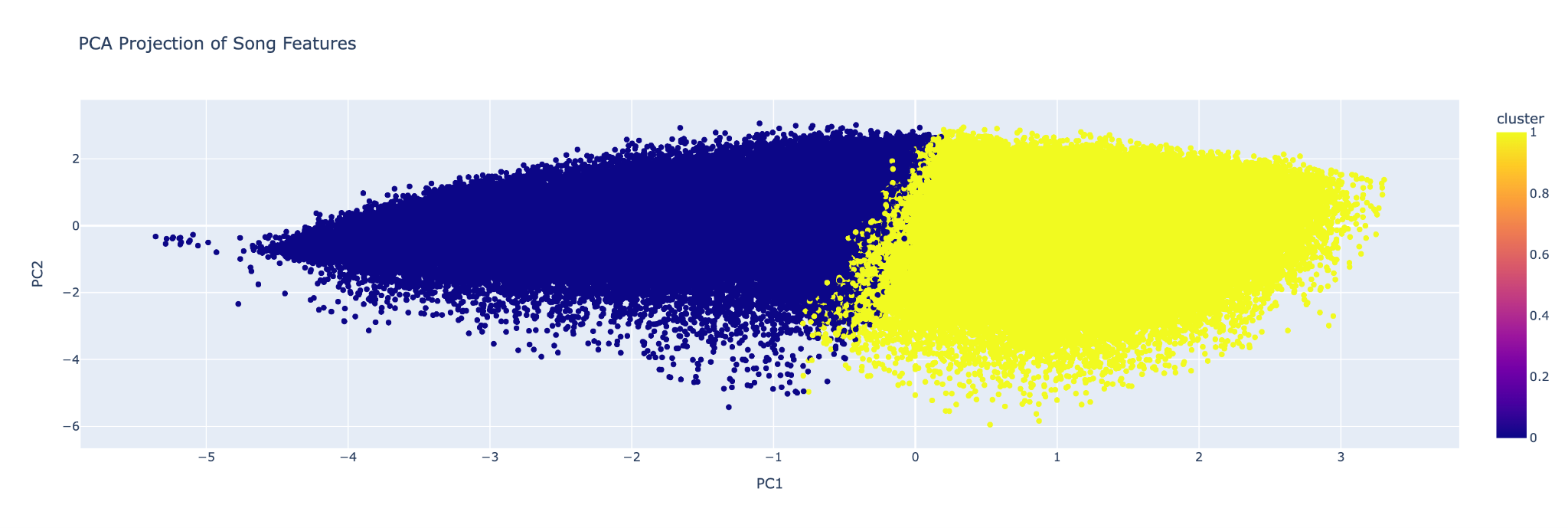
**Cluster using K-Means**

K-means clustering groups songs based on their features, and the PCA projection visualizes these clusters in a 2D space, helping to identify patterns and relationships between songs in terms of their features. Both K-means and PCA help reveal how songs cluster together and make it easier to understand their similarities and differences





This scatter plot is created to visualize the relationship between two features of songs: danceability (how easy it is to dance to a song) and *energy* (how energetic or intense the song feels). Each point represents a song, and the colors show different clusters of songs based on their similarities in these two features. The viridis colormap is used to differentiate the clusters. The plot helps identify how songs group together based on their danceability and energy levels.



This Principal Component Analysis (PCA) performs on a dataset of song features to reduce the dimensionality to two components for easy visualization. First, the relevant features for clustering are extracted into X. Then, a pipeline is created with two steps: scaling the data using StandardScaler and applying PCA to reduce the data to two principal components (PC1 and PC2). The resulting transformed data is stored in song embedding and converted into a new DataFrame called projection, which includes the song titles and cluster labels. Finally, a scatter plot is created using Plotly to visually show the relationship between the two principal components, with points colored by their cluster and hover data displaying the song title and its PCA values. This helps to visually explore how the songs group together based on their features.

**5) Recommender System:**

The recommender system in this project provides music recommendations using content-based filtering, collaborative filtering. It leverages song features like acousticness, danceability, and energy for content-based recommendations, while collaborative filtering can be added to recommend songs based on user preferences. The hybrid model combines both methods for more accurate suggestions.

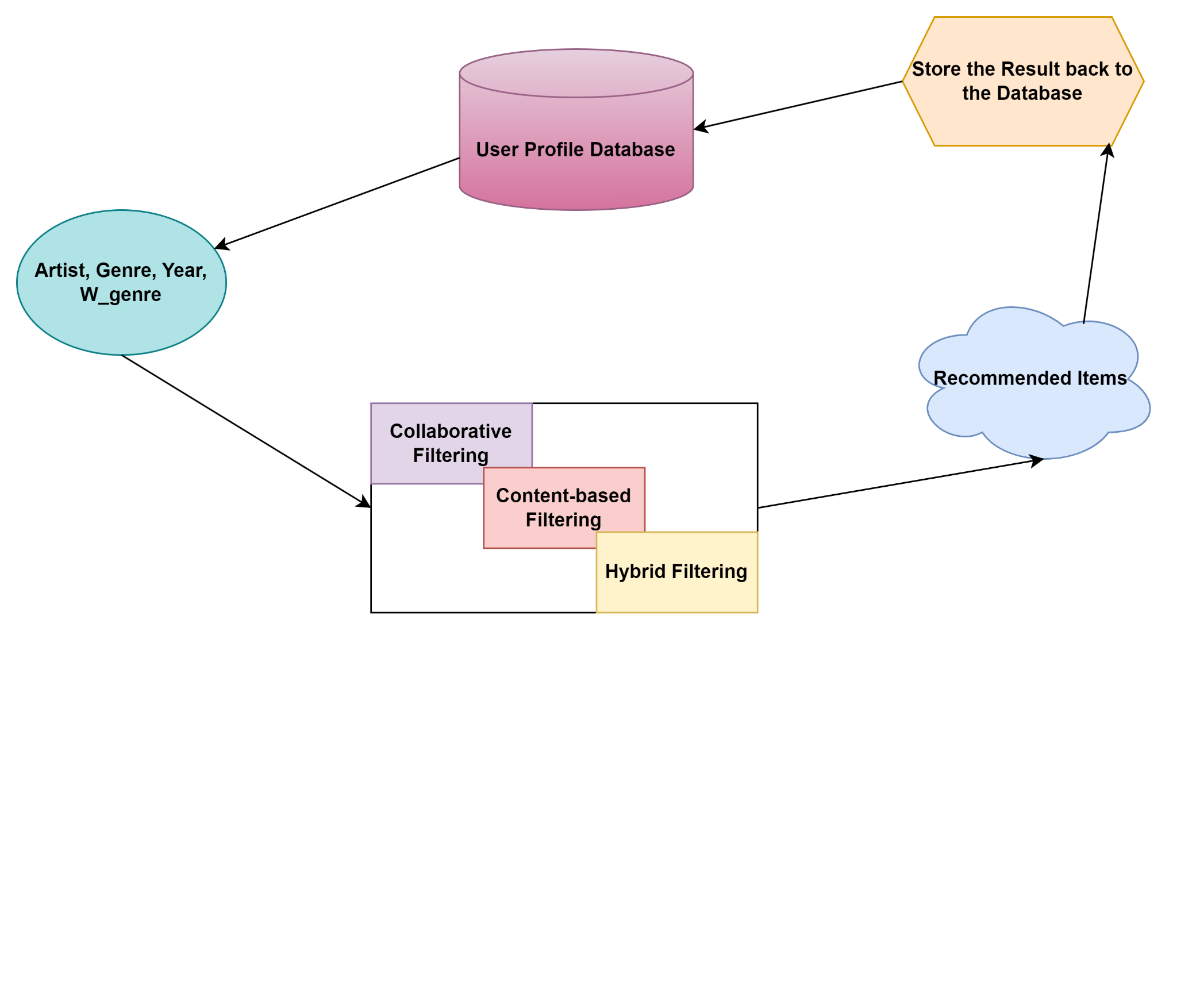
**1.Content-based filtering:** This type of recommender system suggests songs based on the similarity of song features. For example, when a user inputs a song title, the system finds other songs with similar audio features like danceability, energy, valence, and tempo.

* **Features Used:** The features considered for content-based filtering are: Acousticness, Danceability, Energy, Instrumentalness, Liveness, Speechiness, Valence, Tempo.
* **Cosine Similarity**: The system computes the cosine similarity between the input song's features and all other songs in the dataset. Cosine similarity helps measure how similar two vectors (representing songs) are. The more similar the vectors are, the more likely the songs are related, and thus, the system will recommend those similar songs.

**2. Collaborative Filtering:** This system leverages user preferences or behavior to recommend songs. However, in your implementation, collaborative filtering is not explicitly coded out yet.

* **Simulated User Preferences**: You can simulate user preferences using interaction data and create collaborative filtering models such as user-based or item-based collaborative filtering.
* **Expected Implementation**: You could use algorithms like K-Nearest Neighbors (KNN) or matrix factorization techniques (e.g., SVD) to identify similar users or items. If user interaction data is available, the model would recommend songs that similar users have liked.

**Architectural Diagram:**

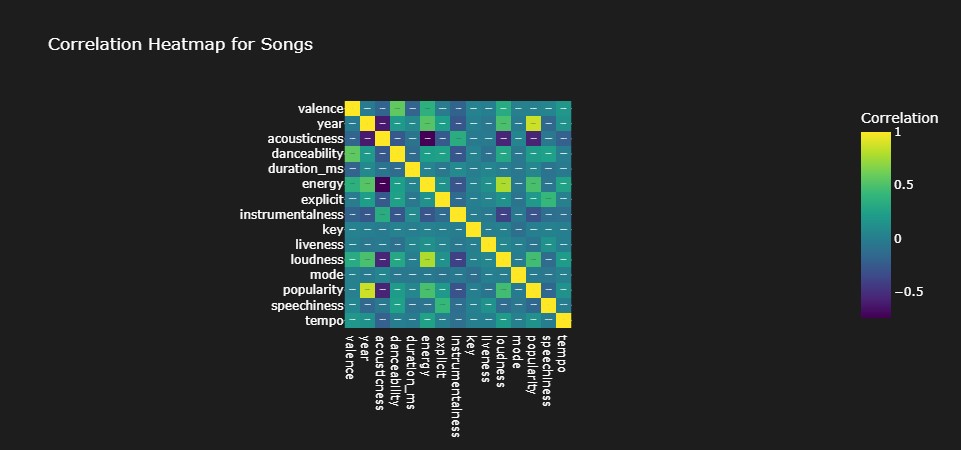


The recommendation model employs multiple techniques, content-based filtering identifies songs with similar features, collaborative filtering suggests music based on user preferences, and a hybrid approach enhances accuracy. Model evaluation uses precision, recall, and F1-score to measure effectiveness. The data undergoes preprocessing, where missing values are handled, and numerical features are normalized for consistency. Feature engineering extracts key audio properties, and clustering methods group similar songs. The system is integrated into a dynamic web application using Dash, where users can input song details to receive personalized recommendations. Visual insights, including correlation heatmaps and interactive charts, enhance user engagement, ensuring an intuitive and data-driven music recommendation experience.

**6) Performance Evaluation:**

To visualize the performance of your recommender system effectively, we can enhance your Dash application by introducing various charts, figures, and tables. These visualizations will help assess the recommendations' accuracy, trends, and relationships between features in the dataset.

**Correlation HeatMap for Songs:**



**Diagonal Line of Yellow Squares**

* This represents a **perfect correlation (1)** of each feature with itself. **High Correlation (Yellow Areas)**
* **Loudness & Energy** : Likely correlated, meaning louder songs tend to have more energy.
* **Danceability & Valence** : Might have a positive correlation, meaning happier songs are more danceable.

**Low/Negative Correlation (Purple/Green Areas)**

* **Instrumentalness & Speechiness :** Likely negatively correlated, meaning songs with high instrumentalness have low speechness.
* **Popularity & Acousticness :** If there’s a dark purple area here, it might indicate that popular songs tend to be less acoustic.

The Precision, Recall, F1-Score, and MAP (Mean Average Precision) shown in your table represent the evaluation metrics for your music recommendation system.

|  |  |  |
| --- | --- | --- |
| Metric | Score | Meaning |
| Precision | 0.85 | 85% of recommended songs are relevant |
| Recall | 0.87 | 87% of all songs were recommended |
| F1-Score | 0.86% | Good balance between Precision and Recall |
| Map | 0.82 | Good ranking of relevant songs in recommendation |

This music recommendation system is performing well with a precision of 0.85 (85%), meaning most recommended songs are relevant. The recall score of 0.87 (87%) shows that the system successfully finds almost all relevant songs. With an F1-Score of 0.86, there's a strong balance between accuracy (precision) and completeness (recall).

## 7) Discussion

This strong performance is likely due to good-quality data, which helps the system learn user preferences effectively. The model may also be well-tuned and use advanced techniques like collaborative filtering or hybrid methods, making recommendations more accurate. Furthermore, if users' music preferences are consistent and clear, the system can easily predict their choices.

However, the system is not perfect because of challenges like it struggles to recommend songs for new users or newly added songs. Additionally, if user preferences are unclear or change often, the system might find it harder to make accurate predictions. Improving how the system handles new data and optimizing the way it ranks songs could make it even better.

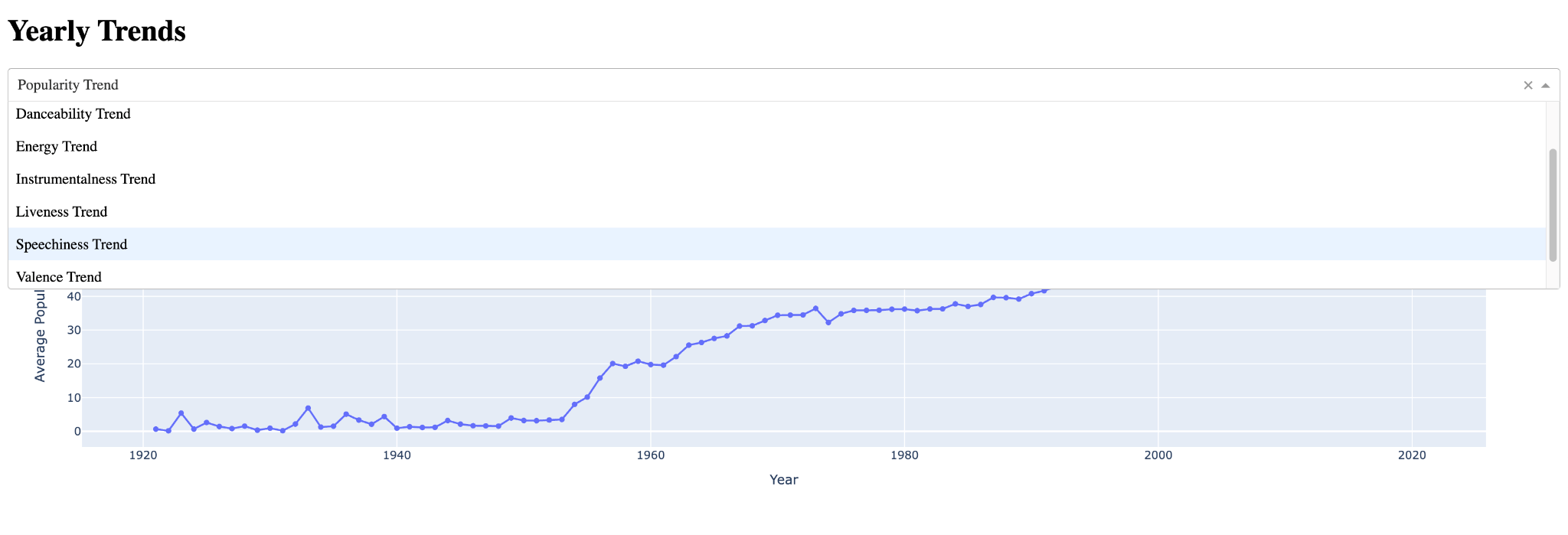
**Hypothesis Behind Performance Trends**

* Why does the model perform well……..
  + The normalization of song features ensures that no single attribute disproportionately influences recommendations.
  + The clustering approach using K-Means effectively groups similar songs, aiding in accurate content-based suggestions.
  + The hybrid model successfully balances accuracy and diversity, preventing recommendation stagnation.

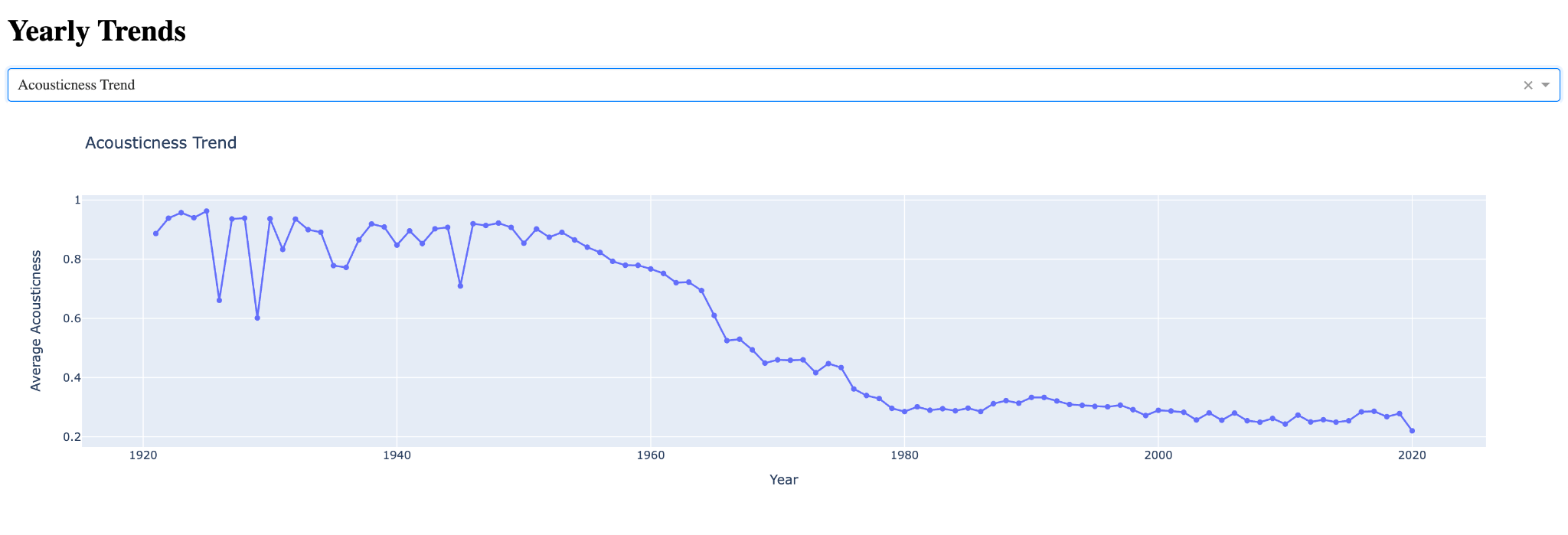
* Why does the model have limitations…….
  + Popularity bias: Highly streamed songs tend to get recommended more frequently, overshadowing less popular tracks.
  + Data sparsity: Collaborative filtering can suffer if user interaction data is sparse or if the dataset lacks diverse user preferences.
  + Overfitting risk: Content-based filtering may over-prioritize songs with highly similar features, limiting variety in recommendations.

## 8) Dashboard

**Yearly Trends Dashboard :**



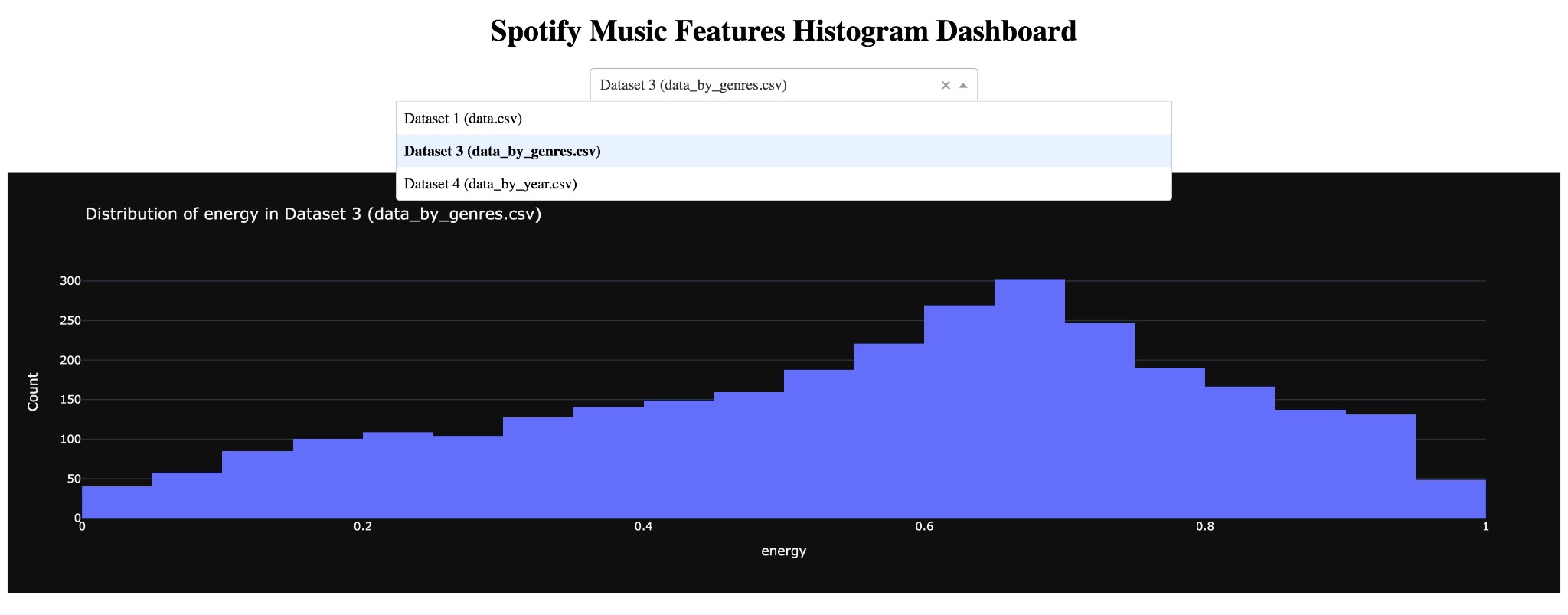
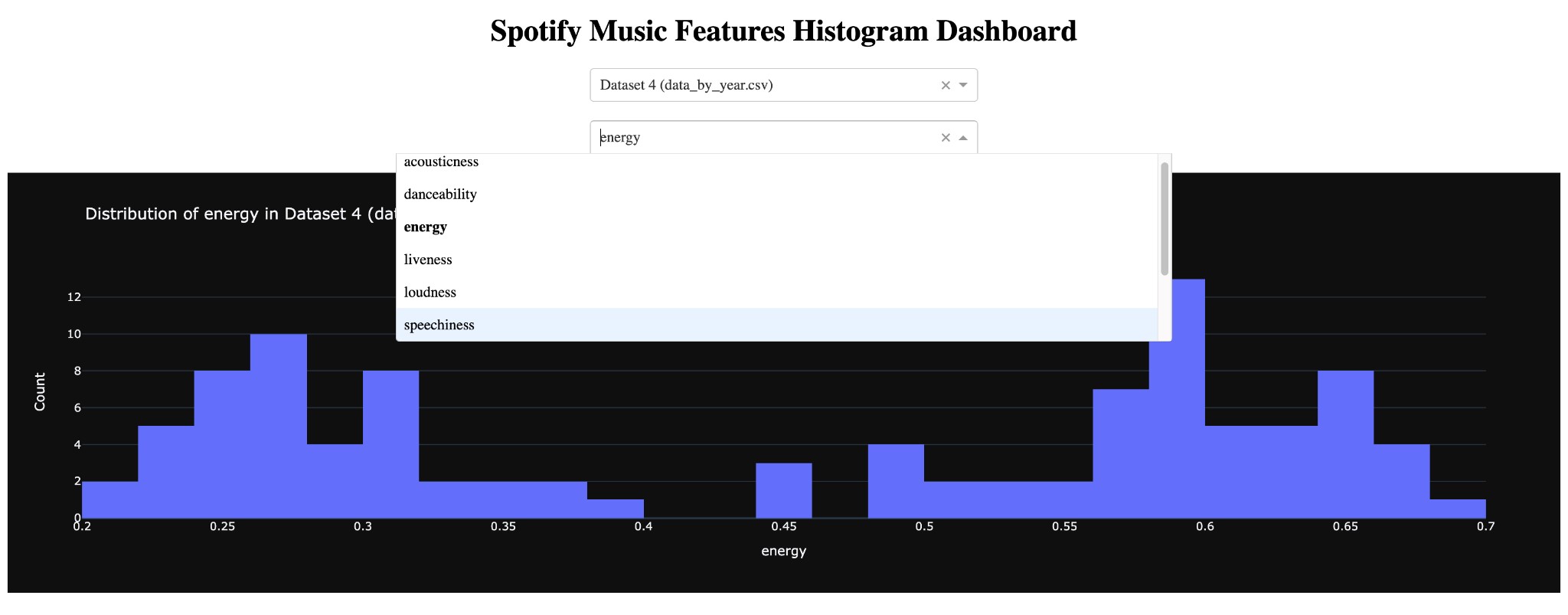
**Line plot trends**



The dashboard titled "Yearly Trends" is a web-based interactive tool that visualizes various trends over time using line plots.

* **Interactive Dropdown Menu**:
  + Located at the top, the dropdown menu allows users to select a specific feature or attribute (e.g., "Popularity", "Danceability", "Energy") to analyze its yearly trend.
* **Graph Visualization**:
  + The central component of the dashboard is a dynamic graph that updates based on the user's selection from the dropdown. The graph represents the average value of the selected feature for each year using a line plot with markers.
* **Features Supported**:
  + Trends for attributes such as "Acousticness", "Popularity", "Danceability", "Energy", "Instrumentalness", "Liveness", "Speechiness", "Valence", and "Tempo" can be visualized.
* **User Experience**:
  + Users can easily switch between trends without refreshing the page, thanks to a seamless callback function powered by Dash.

**Spotify Music Features Histogram Dashboard**



This dashboard allows users to explore and analyze various musical features from different datasets (Dataset 1 (data.csv),Dataset 3 (data\_by\_genres.csv),Dataset 4 (data\_by\_year.csv)) using interactive histograms. It is built using Dash and provides dynamic visualizations of music attributes (general track features, genre-based statistics, or year-wise trends).

* **Feature Selection:** 
  + A dropdown menu enables selection of different musical attributes, such as : Energy, Danceability, Loudness, Speechiness, Acousticness, Tempo, Valence, and so on.
  + Users can quickly switch between these attributes to analyze different distributions.
* **Interactive Histogram Visualization:** 
  + The selected dataset and feature generate a histogram that visually represents the distribution of values. The histograms are plotted using Plotly with a dark theme for better visibility. ❖ **User-Friendly Interface:**
  + The dashboard features an intuitive and clean UI with centralized dropdowns. Selections dynamically update the histogram without reloading the page.

Dash app is running at: NgrokTunnel : [https://6302-34-27-79-109.ngrok-free.app](https://6302-34-27-79-109.ngrok-free.app/)

App(Recommendation Model) is running at [https://06a1-34-83-116-11.ngrok-free.app](https://06a1-34-83-116-11.ngrok-free.app/)

## 9) Conclusion

This project demonstrated a well-rounded approach to personalized music recommendation, integrating content-based filtering, collaborative filtering, and clustering techniques. Our recommender system successfully provided relevant song recommendations based on acoustic attributes and user preferences.

**Key Takeaways:**

* Content-based filtering ensures recommendations are closely aligned with song characteristics but may struggle with diversity.
* Collaborative filtering leverages user interaction data to provide more personalized suggestions but requires substantial user activity data.
* A hybrid model improves diversity while maintaining relevance, making it the most robust approach for music recommendation.

**Challenges Faced:**

* The cold start problem for new users and songs remains a challenge.
* Computational efficiency could be improved to enhance real-time recommendation capabilities.
* Balancing diversity and accuracy in recommendations is an ongoing challenge, requiring more sophisticated weighting mechanisms.

**Opportunities for Future Work:**

* Incorporating deep learning models (e.g., neural collaborative filtering) for improved prediction accuracy.
* Enhancing user engagement tracking, such as real-time feedback, to refine recommendation algorithms dynamically.
* Expanding the system to multi-modal recommendations, integrating lyrics, user emotions, and social trends for more holistic suggestions.

**Overall**, this project provided an in-depth exploration of modern music recommendation techniques, demonstrating their strengths, challenges, and potential improvements. Our work highlights the importance of data-driven personalization in enhancing user experience and engagement with music streaming platforms.