MUSIC RECCOMENDATION SYSTEM

Abstract

This project presents a personalized music recommendation system designed using a comprehensive Spotify dataset comprising over 170,000 tracks spanning nearly a century (1921–2020). The system aims to provide tailored song recommendations that cater to individual user preferences by leveraging advanced data mining techniques and machine learning algorithms. The dataset includes a diverse range of features, such as acoustic attributes (e.g., valence, danceability, energy, tempo, and loudness), alongside metadata like artist names, song popularity, and release dates. These features are meticulously standardized and normalized to ensure consistency and enhance the model's performance.

The system employs clustering techniques such as K-Means and Mini-Batch K-Means to group songs with similar characteristics, enabling a robust categorization of tracks. Dimensionality reduction techniques, including PCA and t-SNE, are utilized to visualize these clusters effectively and provide insights into the relationships between different songs. The effectiveness of clustering methods is evaluated using metrics like the Silhouette Score and Davies-Bouldin Index, which demonstrate the quality and coherence of the formed clusters.

To enhance user engagement, the system incorporates two recommendation functionalities. The first suggests songs based on the artist and cluster of a selected track, while the second identifies tracks within the same genre by analyzing acoustic and tempo similarities. Together, these features bridge the gap between extensive music libraries and individual user preferences, offering a dynamic and personalized listening experience. This project illustrates the potential of data-driven approaches to revolutionize music recommendation systems, making them more intuitive and adaptable to user tastes and trends.

Background and Introduction

The digital age has revolutionized how we consume music, granting us access to a vast and diverse collection of tracks from around the world. However, this overwhelming abundance comes with its own set of challenges. Users often find it difficult to navigate through millions of tracks to discover music that aligns with their unique tastes. This project addresses this issue by developing a personalized music recommendation system that leverages advanced data mining techniques to analyze and predict user preferences.

A personalized music recommendation system is essential for enhancing user engagement and satisfaction. It enables users to move beyond the limitations of generic playlists and explore music that resonates with their preferences, thereby creating a more meaningful and immersive listening experience. By analyzing features like valence, energy, tempo, and popularity, such systems bridge the gap between the user's musical preferences and the vast array of options available, ensuring that each recommendation feels tailored and relevant.

This project employs cutting-edge algorithms and tools to analyze musical attributes and cluster tracks into meaningful groups. Using Python as the primary programming language and Jupyter Notebook as the development environment, the system implements techniques such as K-Means clustering, Mini-Batch K-Means, Principal Component Analysis (PCA), and t-SNE (t-distributed Stochastic Neighbor Embedding). These methods enable the system to normalize data, perform dimensionality reduction, and visualize musical clusters effectively.

By leveraging these advanced techniques, the project aims to transform how users interact with music platforms. Personalized recommendations not only save time but also enhance the joy of music discovery, making it easier for users to find songs and artists that match their mood, style, or preferences. In doing so, this work contributes to the broader evolution of personalized services in the music industry, paving the way for a richer and more data-driven user experience.

Related Work

The development of music recommendation systems has been an area of active research, driven by the need to improve user satisfaction and engagement in the face of an evergrowing music catalog. Various methods and models have been explored to address this challenge, ranging from collaborative filtering to deep learning-based approaches. Collaborative filtering, one of the earliest techniques, analyzes user interactions, such as song ratings or listening history, to recommend tracks based on similarities between users or items. Notable platforms like Spotify and Pandora have leveraged collaborative filtering to enhance their recommendation capabilities.

Spotify's Hybrid Recommendation Model: Spotify employs a hybrid recommendation system that combines collaborative filtering, natural language processing (NLP), and audio analysis. Collaborative filtering is used to track user behavior and identify preferences based on interactions with songs. NLP analyzes metadata and user-generated content, such as playlists and tags, while audio analysis extracts acoustic features directly from the song's waveform. This multifaceted approach enables Spotify to provide robust and personalized recommendations that reflect both user behavior and the intrinsic features of songs.

Traditional Content-Based Filtering: Content-based filtering recommends songs based on comparing track features like tempo, energy, and danceability with user preferences. This method allows for highly personalized recommendations, as it directly correlates with the user's past listening behavior. However, it has limitations in terms of diversity, as it typically suggests tracks that are very similar to the ones a user has already listened to, without incorporating collaborative insights or temporal trends in user preferences.

Hybrid models, which combine collaborative and content-based filtering, have also gained traction for their ability to overcome the limitations of each method. These models improve recommendation accuracy by utilizing both user-item interactions and song features. Moreover, advancements in machine learning and artificial intelligence have introduced

neural networks and deep learning models into music recommendation systems. These models can capture complex relationships between users, tracks, and their features, providing more nuanced and dynamic recommendations.

Problem Definition

The problem addressed in this project is the development of an effective and personalized music recommendation system that can handle the vast and diverse music catalog available on streaming platforms. With millions of tracks spanning multiple genres, moods, and eras, users often struggle to discover music that resonates with their unique preferences. The core challenge lies in accurately predicting and recommending songs that align with individual user preferences, which are influenced by a wide range of factors such as genre, tempo, energy, mood, and past listening behavior. This calls for sophisticated algorithms that can bridge the gap between user expectations and the overwhelming volume of musical choices.

As music catalogs continue to grow, traditional recommendation methods like collaborative filtering and content-based filtering face limitations in terms of scalability, diversity, and the cold-start problem, where insufficient data on new users or songs reduces the effectiveness of the recommendations. Moreover, these traditional methods often fail to capture the dynamic nature of user preferences and temporal trends in music. To address these challenges, this research leverages advanced data mining techniques, such as clustering and dimensionality reduction, to identify hidden patterns in user preferences and musical features. These techniques not only enhance the precision of recommendations but also improve the system's ability to adapt to varying user preferences and emerging trends.

The goal is to create a recommendation system that provides personalized song suggestions, enhances diversity, improves scalability, and addresses challenges like the cold-start problem. By integrating innovative techniques, this project aims to redefine how users interact with music recommendations, making the system more effective and user centric.

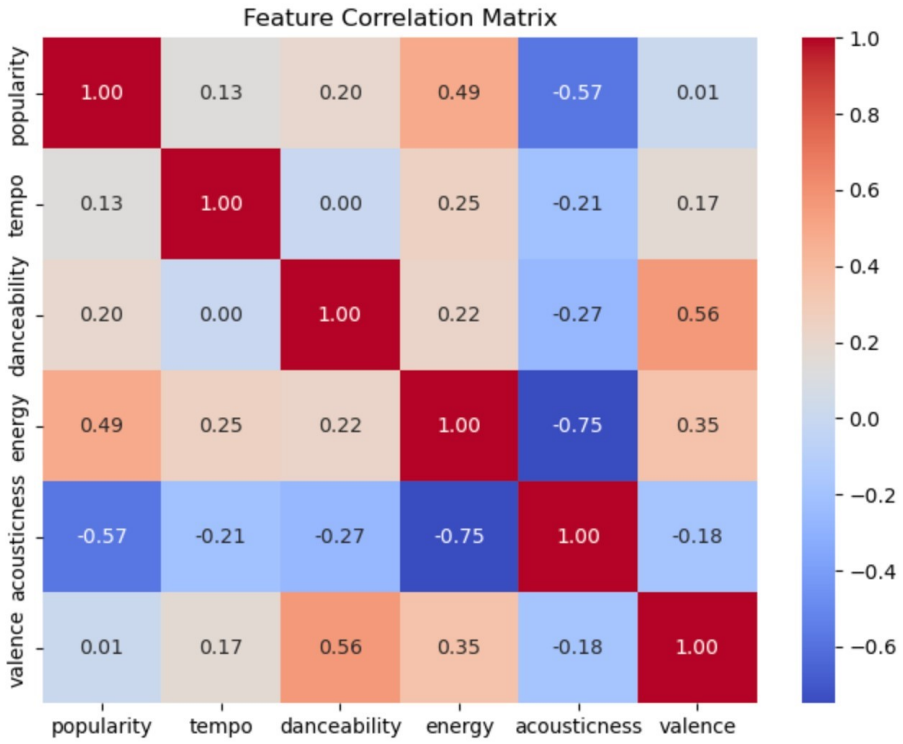
Methodology

The development of the personalized music recommendation system involves a series of well-defined steps, integrating data preprocessing, machine learning techniques, and recommendation algorithms. The methodology reflects a balance between existing frameworks and customized approaches tailored to the Spotify dataset and project objectives.

Data Preprocessing

We began by importing necessary libraries, such as pandas, numpy, sklearn, and matplotlib, and then loaded our datasets. We checked for missing values in the data and ensured data cleanliness, which is crucial for accurate analysis and modeling. Preprocessing ensures data quality and suitability for machine learning tasks.

- **Normalization:** We normalized numerical features using the StandardScaler to standardize the data and ensure uniformity across features. This normalization step helped avoid biases during clustering, as algorithms like K-Means are sensitive to feature scale.
- **Feature Selection**: Key features like valence, popularity, and acoustic characteristics are retained for analysis. Unnecessary or redundant attributes are excluded. We also computed the correlation matrix. The correlation matrix helps identify relationships between features.



Clustering Algorithms

Once the data was preprocessed, clustering algorithms were applied to group similar songs based on their acoustic features, energy, popularity, and tempo. The goal was to identify meaningful clusters that could inform the recommendation system.

- **K-Means clustering**

The K-Means algorithm, an unsupervised machine learning technique, was used to divide the songs into distinct clusters. The number of clusters (k) was determined using the Elbow Method, which helped identify the optimal value by observing the point at which the within-cluster sum of squares (inertia) starts to level off. This ensured that the clustering process would balance computational efficiency and model accuracy.

- **Mini-Batch K-Means Clustering**

To handle large datasets more efficiently, we used Mini-Batch K-Means, which processes the data in smaller batches, significantly reducing computation time. While this technique uses a subset of the data at each step, its performance was evaluated to ensure that it still maintained a comparable level of cluster quality as the standard KMeans algorithm.

Dimensionality Reduction

We used Principal Component Analysis (PCA) to reduce the high-dimensional feature space into two components for better visualization. PCA helps in retaining the variance of the data while reducing the number of dimensions, making it easier to interpret clustering results. We also employed t-SNE for further dimensionality reduction, particularly to visualize non-linear relationships between features in a 2D plane. t-SNE captures local patterns better than PCA, providing a more intuitive visualization of clusters.

Recommendation System

Based on the clusters formed by K-Means and Mini-Batch K-Means, we developed recommendation systems that suggest songs like a user's input based on either the artist or genre.

Recommendation Based on Artist and Cluster

The recommendation system based on artist and cluster suggests songs that belong to the same artist and are within the same cluster as the input song. After identifying the artist and cluster of the selected song, the system filters for songs that match both. This ensures that the recommendations are similar not only in terms of the artist but also in terms of acoustic features, as songs in the same cluster share similar characteristics. The system then excludes the input song from the recommendations and returns the top N songs based on their similarity to the input song. The implementation uses clustering information to group and filter songs, ensuring the recommendations are both artistspecific and acoustically relevant.

Recommendation Based on Genre

For genre-based recommendations, the system identifies the genre of the selected song and then suggests songs from the same genre with similar acoustic features (like tempo and acousticness). It compares the features of the selected song with other songs in the same genre to find the best matches. The recommended songs are ranked by popularity and presented to the user as a list of tracks that align with their musical preferences, ensuring a personalized experience. This approach leverages the genre's acoustic features for accurate song suggestions. The implementation first identifies the genre by comparing acousticness and tempo, then filters for songs that share similar traits within the same genre.

Experimental Setting

Datasets Used : We used two types of datasets in our project

- **Main Dataset**: This dataset includes granular information for each song, such as valence (musical positivity), energy, tempo, danceability, and popularity. These

features are essential for analysing the music's characteristics and are used as input for the clustering model.

- **Artist-Level Dataset**: This dataset provides aggregated features for each artist, such as acousticness and instrumentalness. These features help provide context for song recommendations by grouping tracks from the same artist or similar artists.

Parameters and Metrics

1. Clustering Parameters

- **K-Means Parameters**: The optimal number of clusters was determined using the Elbow Method, which helped identify the point where adding more clusters no longer significantly reduces the within-cluster sum of squares. K-Means was run with its default settings after determining the number of clusters.

- **Mini-Batch K-Means Parameters**: To handle larger datasets efficiently, we optimized the batch size parameter. Mini-Batch K-Means was used as an alternative to K-Means to improve computational efficiency without sacrificing too much accuracy.

2. Clustering Evaluation Metrics

Since clustering is an unsupervised learning task, traditional training and testing splits were not applied. Instead, the clustering quality was evaluated using the following metrics:

- **Silhouette Score**: This metric measures how similar an object is to its own cluster compared to other clusters. Scores closer to +1 indicate well-separated clusters. For both K-Means and Mini-Batch K-Means, we obtained similar Silhouette Scores around 0.21, indicating moderate cluster separation.

- **Davies-Bouldin Score**: This score evaluates the compactness and separation of clusters, with lower values indicating better clustering. The Mini-Batch K-Means algorithm achieved a slightly better Davies-Bouldin Score (1.5157) compared to K-Means (1.5293), suggesting that Mini-Batch K-Means provided slightly better separation and compactness of the clusters.

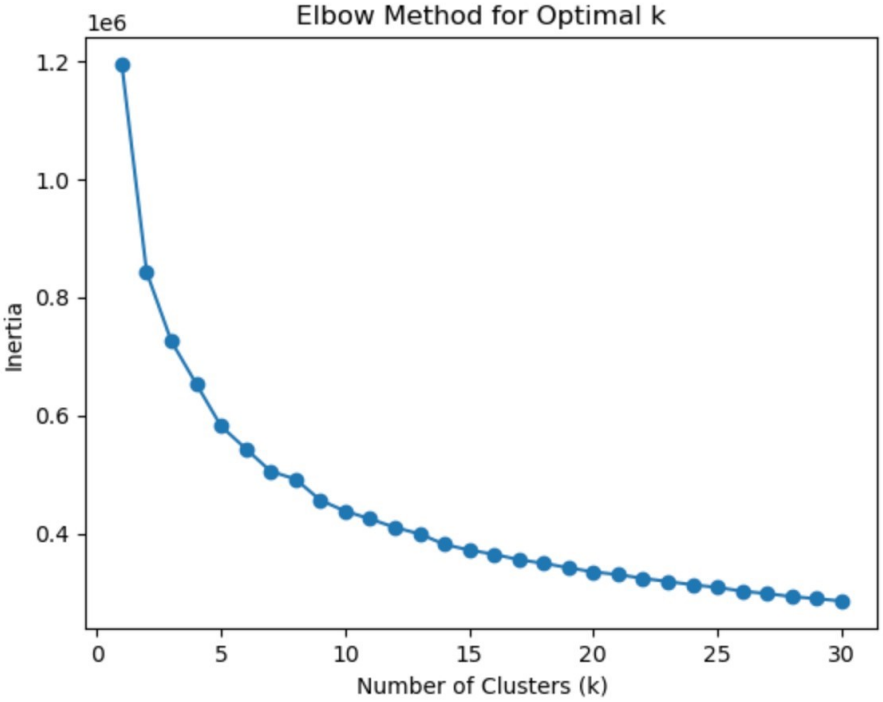
Experimental Results and Analysis

In this project, clustering algorithms (K-Means and Mini-Batch K-Means) were evaluated using Silhouette and Davies-Bouldin Scores. These metrics helped assess cluster quality, visualized with PCA and t-SNE. The project also evaluated personalized recommendation systems based on these clusters, with results presented through the Elbow Method, PCA/tSNE plots, and recommendation outputs.

Elbow Method for Determining Optimal Clusters

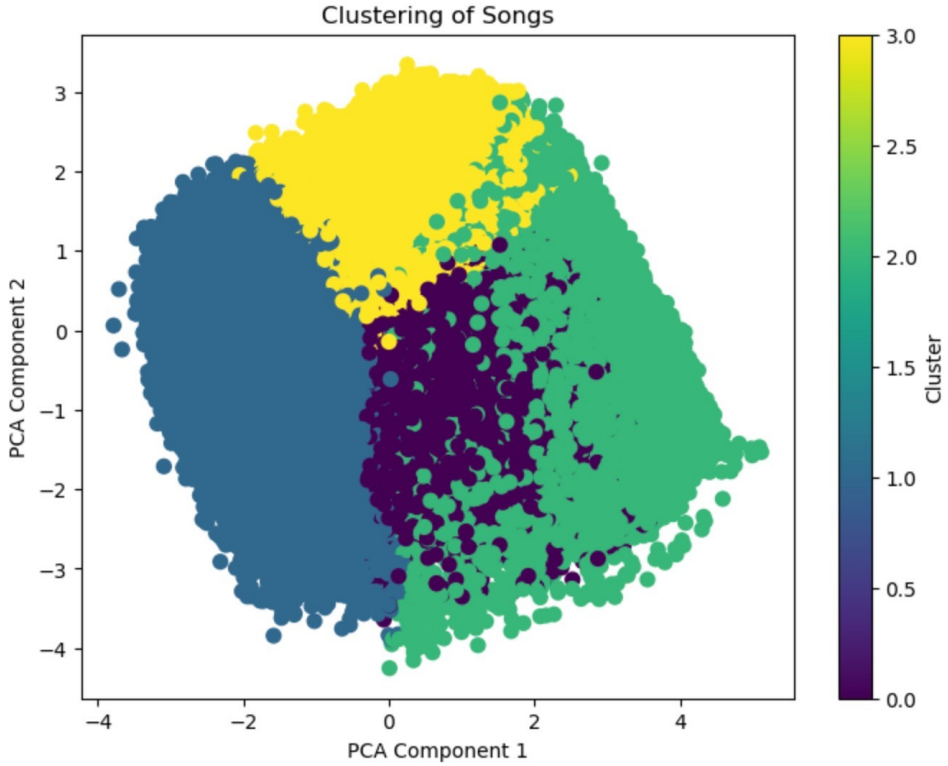
The Elbow Method was used to determine the optimal number of clusters (k) for the KMeans and Mini-Batch K-Means algorithms. The graph below illustrates the relationship between the number of clusters and the Within-Cluster Sum of Squares. The optimal value of k was found to be 4, as indicated by the sharp drop followed by a flattening curve. The

inertia decreased significantly up to k=4, after which the rate of decrease slowed, indicating that adding more clusters did not result in better grouping. This suggests that four clusters best represented the data, balancing model accuracy with computational efficiency. The clustering process resulted in four groups of songs that varied significantly in terms of attributes such as energy, tempo, and popularity. For example, one cluster contained songs with high energy and tempo, suitable for workout playlists, while another group included low-energy, mellow tracks ideal for relaxation.

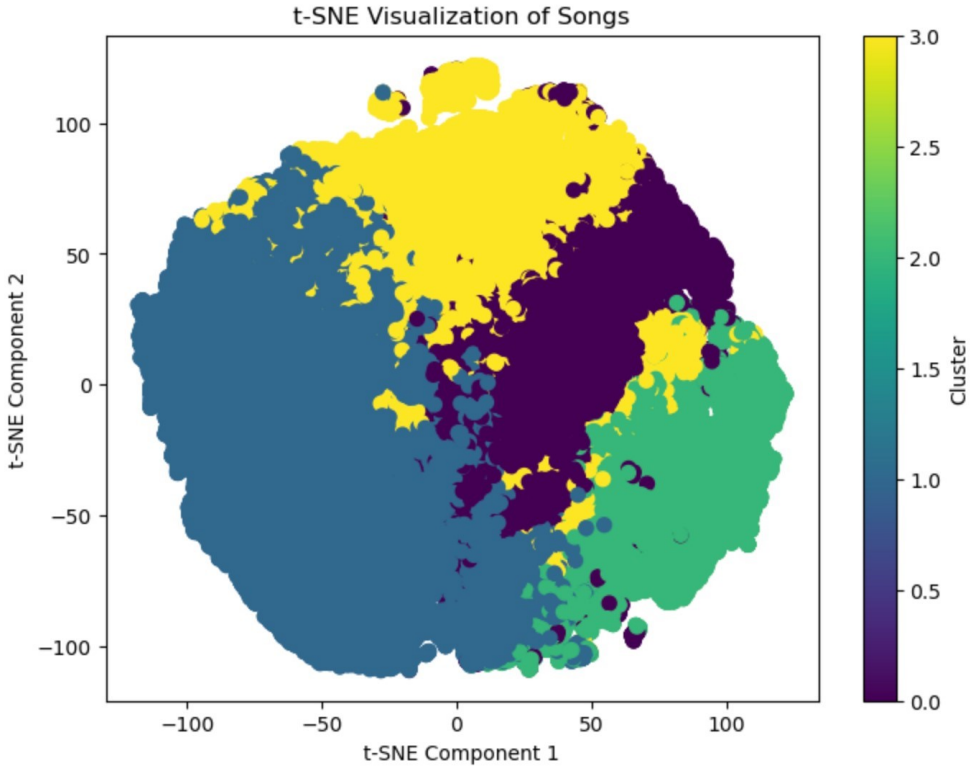


K-Means Clustering

- **PCA Visualization:** The dimensionality of the dataset was reduced to two components (PCA1 and PCA2), and a scatter plot was used to visualize the clusters. The four clusters were clearly distinguishable, with distinct groups forming based on the features like energy, tempo, and popularity.



- **t-SNE Visualization:** Unlike PCA, t-SNE preserves the local structure of the data, revealing more granular patterns within each cluster. The color-coded t-SNE plot showed well-defined boundaries between clusters, confirming that songs with similar features are grouped together.



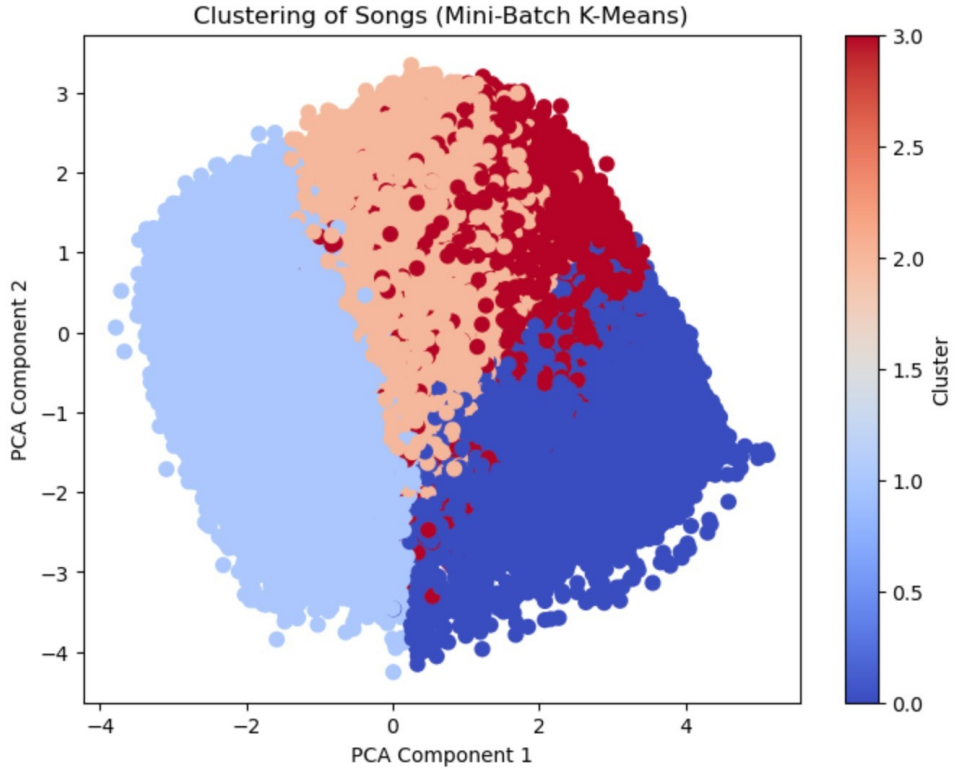
- **Metrics:** The quality of the K-Means clustering was evaluated using two metrics: the Silhouette Score (0.21) and the Davies-Bouldin Score (1.5293). A moderate Silhouette Score suggests that the clusters are not perfectly well-defined but are still meaningful. The Davies-Bouldin Score, while relatively high, indicates that further refinement could be beneficial to improve cluster separation.

```
Davies-Bouldin Score: 1.5293491261170322
Silhouette Score for KMeans: 0.20999886269550724
```

Mini-Batch K-Means

The Mini-Batch K-Means algorithm, optimized for large datasets, demonstrated superior performance in terms of computational efficiency. By processing data in smaller batches, it significantly reduced the time required for clustering compared to standard KMeans. This makes Mini-Batch K-Means especially useful for real-world applications involving large music datasets.

• The visualization of clusters using PCA revealed that songs within the same cluster shared similar acoustic features, energy levels, and danceability, enabling more targeted recommendations.



Metrics: The quality of the Mini-Batch K-Means clustering was evaluated using two metrics: the Silhouette Score (0.21) and the Davies-Bouldin Score (1.5157). Like KMeans, the moderate Silhouette Score suggests that the clusters are not perfectly welldefined but

still meaningful. The lower Davies-Bouldin Score indicates slightly better cluster separation and compactness compared to K-Means.

```
Davies-Bouldin Score (Mini-Batch K-Means): 1.5156664241951083
Silhouette Score for Mini-Batch KMeans: 0.2104579548607184
```

Artist-Based Recommendation System

The artist-based recommendation system is designed to recommend songs by the same artist that belong to the same cluster as the input track. This ensures that the recommendations not only align with the artist's musical style but also share similar characteristics such as energy, tempo, and danceability. The clusters are derived from the K-Means algorithm, which groups songs based on their acoustic features, making the recommendations even more personalized.

For the input song *"Diamonds"* by *Rihanna*, which belongs to Cluster 1, the system identified other songs by *Rihanna* within the same cluster. This approach combines artistspecific filtering with clustering insights. The recommended songs reflect tracks from Cluster 1, highlighting *Rihanna's* work that resonates with similar energy and acoustic traits as *"Diamonds."* Below is the output screenshot showcasing the suggested tracks.The system generated up to 5 recommendations by *Rihanna* within the same cluster. If fewer than 5 matches are available, the system outputs all matching songs.

```
Enter the name of the song to get recommendations:  Diamonds

Recommended Songs based on Artist and Cluster:
                       name  artists  cluster
16623         Pon de Replay  Rihanna        1
16891                   SOS  Rihanna        1
16907            Unfaithful  Rihanna        1
17052              Umbrella  Rihanna        1
17067  Don't Stop The Music  Rihanna        1
```

Recommendation System Based on Genre

The genre-based recommendation system identifies the genre of the input song by analyzing its acoustic properties, such as *acousticness* and *tempo*. Once the genre is determined, the system recommends songs within the same genre that have similar musical features and high popularity.

For example, when the song *"Valentines Day"* was used as input, the system identified its genre as *"anarcho-punk"*. Based on this genre, the system recommended 10 tracks, including *"Blinding Lights"* by *The Weeknd* and *"Relación - Remix"* by various artists. These songs share similar acoustic characteristics and are among the most popular in their genre, providing listeners with diverse yet genre-coherent options

```
Enter a song name: Valentines Day

Top 10 song recommendations for the genre of 'Valentines Day':
                          Song                                                Artist        Genre  Popularity
              Blinding Lights                                          ['The Weeknd'] anarcho-punk          96
             Relación - Remix ['Sech', 'Daddy Yankee', 'J Balvin', 'ROSALÍA', 'Farruko'] anarcho-punk          94
UN DIA (ONE DAY) (Feat. Tainy)            ['J Balvin', 'Tainy', 'Dua Lipa', 'Bad Bunny'] anarcho-punk          92
                  Una Locura                 ['Ozuna', 'J Balvin', 'Chencho Corleone'] anarcho-punk          91
               Hawái - Remix                                ['Maluma', 'The Weeknd'] anarcho-punk          90
                    Caramelo                                            ['Ozuna'] anarcho-punk          87
                    Relación                                             ['Sech'] anarcho-punk          85
                        Nada                    ['Cali Y El Dandee', 'Danna Paola'] anarcho-punk          85
                          21                                           ['Polo G'] anarcho-punk          84
                      Yellow                                        ['Coldplay'] anarcho-punk          84
```

Conclusion

In conclusion, the integration of K-Means clustering with PCA and t-SNE visualization techniques successfully enabled the grouping of songs based on their acoustic features. The clusters formed were meaningful and distinct, allowing for personalized music recommendations. The Mini-Batch K-Means algorithm demonstrated efficiency in clustering large datasets while maintaining a similar quality of results compared to standard K-Means.

The Silhouette and Davies-Bouldin Scores indicated that the clusters were moderately well-separated, though there was room for improvement. The combination of artistbased and genre-based recommendation strategies ensured that users received tailored song suggestions based on both their artist preferences and the musical characteristics of the tracks.

Future Work

Future work can focus on enhancing the clustering model by exploring other clustering algorithms, such as DBSCAN and hierarchical clustering, which can potentially handle non-linear patterns in the data more effectively. DBSCAN is particularly useful for discovering clusters of varying shapes and densities, which could help in identifying more subtle groupings in the dataset. Moreover, incorporating additional features such as song lyrics, mood, or sentiment analysis could further refine the clustering process and improve the accuracy of music recommendations.

Additionally, the current recommendation system could be enhanced by integrating collaborative filtering techniques, which consider users' preferences and historical interactions with songs. A hybrid recommendation system that combines content-based filtering (using clustering) and collaborative filtering would offer even more personalized music recommendations. Furthermore, more advanced deep learning techniques, such as autoencoders, could be explored for feature extraction and clustering, offering improved performance in handling complex and high-dimensional data.

References

1. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.
2. Leland, M., Hinton, G., & Roweis, S. (2008). *Visualizing Data Using t-SNE*. Journal of Machine Learning Research, 9, 2579–2605.
3. P. Knees and M. Schedl, Music similarity and retrieval: an introduction to audio- and web-based strategies. Berlin: Springer, 2016.
4. A. van den Oord, S. Dieleman and B. Schrauwen, “Deep content based music recommendation”. 2013.
5. J. Cleveland, D. Cheng, M. Zhou, T. Joachims and D. Turnbull, “Content-based music similarity with triplet networks”. 2020.
6. G. Shani and A. Gunawardana, “Evaluating recommendation systems”. 2011.
7. M. Defferrard, K. Benzi, P. Vandergheynst and X. Bresson, “FMA: a dataset for music analysis”. 2017.
8. K. Gurjar and Y. Moon, “A comparative analysis of music similarity measures in MIR systems”. 2018.
9. J. Kaitila, “A content-based music recommender system”. 2017.