# Understanding Song Recommendations Through Musical Attribute Clustering and Analysis

Miguel Raphael Layos, Kris Brian Diaz
*College of Computing and Information Technologies*
*National University - Philippines*
Manila, Philippines
layosm@students.national-u.edu.ph
diazkv1@students.national-u.edu.ph

***Abstract*—This paper aims to explore song recommendations through musical attribute clustering and analysis. The dataset undergoes pre-processing and dimensionality reduction using Principal Component Analysis (PCA) before clustering. Clustering is performed using K-means, Agglomerative Clustering, and DBSCAN. After clustering, the models are evaluated using the Silhouette score, and the optimal model is selected for the development of a song recommendation system.**

***Index Terms*—clustering, PCA, K-means, Agglomerative, DBSCAN, song recommendation system**

## I. Introduction

Song recommendation systems have become an integral component of modern streaming platforms. To attract new users, streaming platforms implement increasingly advanced solutions in the field of music recommendation. There are several directions such as content-based recommendation, collaborative filtering-based recommendation, hybrid model-based recommendation, and label-based recommendation [1].

Among these methods, collaborative filtering is one of the most used approaches for music recommendation at present. It analyzes a user's past listening history and ratings to identify other users with similar preferences, then suggests songs that those users [2]. However, this method faces the cold-start problem, meaning it struggles to generate accurate recommendations for new users or newly added songs and popularity bias [3], which favors frequently played tracks over lesser-known but potentially relevant selections [4].

To address these limitations, this research explores an attribute-based clustering approach, wherein recommendations are derived exclusively from the intrinsic characteristics of musical compositions rather than user behavior. Unlike content-based filtering methods, attribute-based clustering does not rely on explicit user interactions, enabling a more unbiased and data-driven classification of music. The subjective nature of music taste makes recommendations highly challenging. Moreover, the human perception of sound must be considered during analysis. Personal preferences in music, which are inherently subjective, make defining effective recommendation systems complex. However, a system capable of precisely matching new music to individual preferences could significantly enhance user experience [5].

In this study, unsupervised clustering will be used to group songs based on musical attributes. By categorizing songs according to their inherent features using K-means, Agglomerative, and DBSCAN, this approach seeks to understand patterns that may naturally resonate with subconscious preferences. The study will analyze the distinctiveness of the resulting clusters to assess whether they form meaningful and well-separated groups. By examining the efficacy of different clustering techniques, this study aims to contribute to a deeper understanding of how objective musical properties can be leveraged for music recommendation. Additionally, we aim to develop a recommender system based on the clustering result. The findings will provide insights into whether clustering-based approaches can offer an alternative to traditional recommendation methods, potentially reducing biases and improving recommendations for users with diverse musical tastes.

## II. Review of Related Literature

The vast availability of music online has made it increasingly challenging for users to navigate and fully enjoy such an overwhelming amount of content. Song recommendation systems aim to address this issue by helping users discover songs that align with their preferences. Since music plays a significant role in human life, an effective recommendation system that accurately captures individual tastes could be highly beneficial. However, music preferences are inherently subjective and dynamic, what a person enjoys today may differ from what they prefer tomorrow. This unpredictability presents a major challenge for recommendation systems, as they struggle to account for real-time changes in user preferences. Despite these challenges, there is a continuous demand for more efficient and adaptable music recommendation systems [6].

Researchers, as well as independent analysts, have explored the use of unsupervised clustering techniques on musical attributes to uncover hidden patterns in songs. These studies aim to understand how different musical characteristics relate to one another and assess the effectiveness of clustering in enhancing song recommendation systems.

There exist a study that proves K-means clustering algorithm can be a powerful tool in developing music recommendation systems [7]. The algorithm demonstrated its potential.

However, the paper stated that there are still challenges to be addressed and ongoing research and development in the field to improve the accuracy and relevance of music recommendations

Chen, Dong and Liu's design of music recommendation system based on EDA and K-means cluster analysis has concluded that the results show that the construction and analysis of the music recommendation system can explore the relationship between music popularity and music acoustic characteristics through visual analysis and clustering analysis of music data, and establish an intelligent and personalized music recommendation system to improve users' listening experience and music sales performance [8].

A study from Jin and Hang [9], where the data-driven technique of the K-means Clustering approach is discovered orchestrates a comprehensive symphony of song groupings based on shared audio attributes. extends an invitation to identify new, unexplored patterns and insights hidden inside the musical information and that the Adoption of either attribute based or content based filtering strategy is dependent on the end user's preference. However there is a limitation within Algorithmic Bias, The study admits the possibility of bias in the recommendation systems used. It is stated in that to guarantee fair and varied suggestions, future research should focus on addressing and minimizing these biases.

An independent project has provided valuable insights through the use of Kmeans clustering. After clustering from a large dataset of songs containing 300,000 data. An input of a song provided relevant recommendation. This however is a project based approach therefore a personal bias may occur [10].

A study implementing a music recommendation system using a K-means clustering algorithm demonstrated the potential of the algorithm in providing personalized music recommendations. The research stated that although it provided valuable results One potential drawback of this approach is that it may be too objective and not take into account other factors that can influence a user's music preferences, such as lyrics or artists [11].

Multiple studies provide valuable insights into structuring our methodology, promising results have been gathered on K-means clustering, however, There is a noticeable gap in studies exploring alternative approaches such as Agglomerative Clustering and DBSCAN, highlighting an opportunity for further research in these areas.

## III. Methodology

This section presents the methodology adopted in this study. The objective is to cluster songs based on their musical attributes, analyze patterns, and develop a recommendation system using the clustering results. To achieve this, a dataset of songs was collected and preprocessed before applying unsupervised clustering algorithms, including K-means, Agglomerative Clustering, and DBSCAN. To determine the optimal parameters, we examine validation metrics suitable for each algorithm, such as the Elbow Method, Silhouette Score, and manual evaluation. The most suitable clustering method for the song recommendation system will be determined based on the Silhouette Score.

For the recommendation system, achieving a balance between clustering structure and accuracy is essential. While clusters provide an organizational basis, overly small clusters may limit recommendation diversity, whereas overly broad clusters may reduce precision. An iterative approach will be applied, leveraging domain knowledge to refine clustering and ensure that recommendations maintain both relevance and variety. This ensures that the final system effectively groups similar songs while preserving flexibility for personalized recommendations. The following sections provide a detailed discussion of the methodology.

### A. Data Collection

The dataset used in this study is the Kaggle dataset titled *Spotify 30,000 Songs* [12]. It is a collection of songs obtained from the Spotify API, containing a total of 32,833 rows and 23 columns. The dataset includes various attributes that describe each track.

- *track_id* – A unique identifier for the song.
- *track_name* – The name of the song.
- *track_artist* – The artist of the song.
- *track_popularity* – A popularity score ranging from 0 to 100, where higher values indicate greater popularity.
- *track_album_id* – A unique identifier for the album.
- *track_album_name* – The name of the album containing the song.
- *playlist_name* – The name of the playlist in which the song appears.
- *playlist_id* – A unique identifier for the playlist.
- *playlist_genre* – The genre of the playlist.
- *playlist_subgenre* – The subgenre of the playlist.
- *danceability* – A measure (0.0 to 1.0) of how suitable a track is for dancing, based on tempo, rhythm stability, and beat strength.
- *energy* – A measure (0.0 to 1.0) of a track's intensity and activity, where higher values indicate more energetic songs.
- *key* – The estimated key of the track, represented as an integer using Pitch Class notation (e.g., 0 = C, 1 = C#).
- *loudness* – The average loudness of a track in decibels (dB), useful for comparing relative loudness.
- *mode* – The modality of the track, where 1 represents a major key and 0 represents a minor key.
- *speechiness* – A measure (0.0 to 1.0) indicating the presence of spoken words in a track.
- *acousticness* – A measure (0.0 to 1.0) estimating whether the track is acoustic.
- *instrumentalness* – A measure (0.0 to 1.0) predicting whether a track has vocals, where higher values indicate more instrumental tracks.
- *liveness* – A measure (0.0 to 1.0) detecting the presence of a live audience, where values above 0.8 indicate live performances.

- *valence* – A measure (0.0 to 1.0) describing the musical positiveness of a track, where higher values indicate happier-sounding music.
- *tempo* – The estimated tempo of a track in beats per minute (BPM).
- *duration_ms* – The duration of the song in milliseconds.

### B. Data Pre - Processing

Data pre-processing is an essential step in ensuring accurate, consistent, and unbiased analysis. During this stage, all null values were removed, reducing the number of records to 32,817. Features relevant to clustering songs based on musical attributes were then selected. The chosen features for this research are danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, valence, and tempo. Additionally, a separate dataset was maintained to preserve information such as track name, track artist, playlist genre, and playlist subgenre, as these features are essential for analysis and the development of the recommendation system.

### C. Principal Component Analysis

Principal Component Analysis (PCA) is a dimensionality reduction technique that transforms a high-dimensional dataset into a lower-dimensional space while preserving as much variance as possible. It achieves this by identifying the principal components, which are new orthogonal axes that capture the highest variance in the data [13].

Given a dataset $\mathbf{X} \in \mathbb{R}^{n \times d}$ with $n$ observations and $d$ features, PCA first standardizes the data by subtracting the mean and scaling by the standard deviation. The covariance matrix $\mathbf{\Sigma}$ of the dataset is then computed as

$$\mathbf{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \tag{1}$$

where $\bar{\mathbf{x}}$ is the mean of the dataset. The principal components are obtained by computing the eigenvalues and eigenvectors of $\mathbf{\Sigma}$. Let $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k]$ be the top $k$ eigenvectors corresponding to the largest eigenvalues, then the transformed data $\mathbf{X}'$ is given by

$$\mathbf{X}' = \mathbf{X}\mathbf{V} \tag{2}$$

where $\mathbf{X}'$ represents the lower-dimensional representation of the dataset. PCA is particularly useful for reducing noise and improving computational efficiency while maintaining the most important patterns in the data.

### D. K-means Clustering Algorithm

K-means is an unsupervised learning algorithm used to group data into $k$ clusters. It works by assigning each data point to the closest cluster and then updating the cluster centers based on the assigned points. This process repeats until the clusters no longer change significantly.

First, $k$ cluster centers, $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k$, are chosen randomly. Each data point $\boldsymbol{x}_i$ is then assigned to the nearest cluster using the Euclidean distance:

$$C_j = \left\{ \boldsymbol{x}_i \mid \|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^2 \leq \|\boldsymbol{x}_i - \boldsymbol{\mu}_m\|^2, \forall m \in \{1, \ldots, k\} \right\} \tag{3}$$

Once all points are assigned, the cluster centers are updated by calculating the average of all points in the cluster:

$$\boldsymbol{\mu}_j = \frac{1}{|C_j|} \sum_{\boldsymbol{x}_i \in C_j} \boldsymbol{x}_i \tag{4}$$

This process continues until the cluster centers stop changing significantly. The goal is to minimize the total variation within clusters:

$$\sum_{j=1}^{k} \sum_{\boldsymbol{x}_i \in C_j} \|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^2 \tag{5}$$

K-means is fast and efficient but depends on the initial choice of cluster centers and may not always find the best solution [14].

### E. Agglomerative Hierarchical Clustering

Agglomerative Hierarchical Clustering (AHC) is a bottom-up clustering technique where each data point initially forms its own cluster. The algorithm iteratively merges the two most similar clusters until a single cluster remains or a predefined number of clusters is reached. Unlike partition-based methods such as K-means, AHC produces a dendrogram, which provides a hierarchical representation of the data and allows for flexible selection of clusters [15].

The similarity between clusters is determined using a linkage criterion, which defines how the distance between two clusters is measured. Various linkage methods exist, including single linkage (minimum distance), complete linkage (maximum distance), and average linkage (mean distance). In this study, Ward's method is used, which minimizes the increase in intra-cluster variance when merging clusters.

In Ward's method, the distance between two clusters $C_i$ and $C_j$ is computed as:

$$d(C_i, C_j) = \frac{|C_i||C_j|}{|C_i| + |C_j|} \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|^2 \tag{6}$$

where $|C_i|$ and $|C_j|$ represent the number of points in clusters $C_i$ and $C_j$, and $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$ denote their respective centroids. This ensures that merging clusters results in minimal variance increase, leading to compact and balanced groupings.

The clustering process begins with each data point as an individual cluster. At each step, the two clusters that result in the smallest increase in total variance are merged. The distance matrix is updated accordingly, and the process continues until the desired number of clusters is reached. The hierarchical nature of AHC allows for a flexible selection of clusters by cutting the dendrogram at different levels.

A key advantage of AHC is that it does not require a predefined number of clusters, unlike K-means. Instead, the dendrogram provides a structured way to determine an appropriate number of clusters. Ward's method, in particular, helps maintain compact and well-balanced clusters, making it suitable for structured datasets. However, AHC has a computational complexity of $O(n^2 \log n)$, which can be computationally expensive for large datasets compared to K-means, which scales more efficiently [16].

### F. DBSCAN Algorithm

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a density-based clustering algorithm that groups closely packed points together while marking points in low-density regions as noise. Unlike centroid-based or hierarchical clustering methods, DBSCAN does not require specifying the number of clusters in advance and can discover clusters of arbitrary shapes.

The algorithm relies on two parameters: $\epsilon$, which defines the neighborhood radius, and $MinPts$, the minimum number of points required to form a dense region. A point $p$ is classified based on these parameters. If at least $MinPts$ points, including itself, exist within an $\epsilon$-radius neighborhood, then $p$ is considered a core point:

$$N(p) = \{q \in D \mid d(p,q) \leq \epsilon\} \tag{7}$$

where $d(p,q)$ represents the distance between points $p$ and $q$, and $D$ is the dataset. A point $p$ that lies within the $\epsilon$-neighborhood of a core point but does not satisfy the $MinPts$ requirement itself is classified as a border point. If a point does not belong to any cluster and is not density-reachable from any core point, it is labeled as noise.

The clustering process begins by selecting an unvisited point from the dataset. If the point is a core point, a new cluster is created, and all density-reachable points are recursively added. Two points $p$ and $q$ are density-reachable if there exists a sequence of core points linking them. If a point is not density-reachable from any existing cluster, it is marked as noise.

DBSCAN is effective at identifying clusters of varying shapes and sizes while handling noise efficiently. However, its performance depends on the choice of $\epsilon$ and $MinPts$, and it may struggle when clusters have different densities. The algorithm has a time complexity of $O(n \log n)$ when using an efficient spatial index such as an R-tree or k-d tree, making it suitable for large datasets [17].

### G. Elbow Method

We used the elbow method to find the ideal k for K-means and Agglomerative algorithms. The Elbow Method is a technique used to determine the optimal number of clusters ($k$) in clustering algorithms such as K-means. It is based on analyzing the *Within-Cluster Sum of Squares* (WCSS), which measures the compactness of clusters by calculating the squared distances between data points and their assigned cluster centroids. The WCSS is given by:

$$WCSS = \sum_{j=1}^{k} \sum_{i \in C_j} ||x_i - \mu_j||^2 \tag{8}$$

where $k$ represents the number of clusters, $C_j$ is cluster $j$, $x_i$ is a data point in cluster $C_j$, and $\mu_j$ is the centroid of the cluster.

The method involves plotting WCSS against different values of $k$. As $k$ increases, WCSS decreases since clusters become smaller and more compact. However, beyond a certain point, the decrease in WCSS becomes marginal. The optimal number of clusters is identified at the "elbow point," where the rate of decrease in WCSS slows significantly. This point represents the best trade-off between minimizing intra-cluster variance and avoiding excessive fragmentation of data [18].

By applying the Elbow Method, we ensure that the chosen $k$ value is neither too high, leading to overfitting, nor too low, causing under-segmentation of the dataset.

### H. Silhouette Score

The Silhouette Score is a metric used to evaluate the quality of clustering by measuring how well data points are assigned to their clusters. It quantifies how similar a point is to its own cluster compared to other clusters. The Silhouette Score for a given data point $i$ is defined as:

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{9}$$

where $a(i)$ represents the average distance between point $i$ and all other points in the same cluster, and $b(i)$ is the lowest average distance between point $i$ and points in the nearest neighboring cluster.

The Silhouette Score ranges from $-1$ to 1. A higher score indicates that clusters are well-separated and compact, while a lower or negative score suggests that points are assigned to the wrong clusters. The overall clustering quality is assessed by computing the mean Silhouette Score across all data points:

$$S = \frac{1}{n} \sum_{i=1}^{n} S(i) \tag{10}$$

where $n$ is the total number of data points. This metric is particularly useful in determining the optimal number of clusters in unsupervised learning [19].

## IV. Results and Discussion

### A. Principal Component Analysis Result

in Figure 1. The first principal component captures approximately 25% of the total variance, indicating that it holds the most significant patterns in the data. The second component retains 15%, while the third accounts for 10% of the variance. Together, these components preserve 50% of the total variance. There is a significant loss in variance, however, as the dataset contains multiple dimensions, some which have high correlation, PCA is still viable to reduce these dimensions into 3 while keeping valuable information.


Fig. 1. PCA Explained Variance Result

### B. K-means Result

After using the PCA result. We started clustering and evaluating. We run the Elbow Method to find the ideal K for K-means As shown in figure 2.

The results indicate that 3 would be the optimal number of clusters for clustering. However, based on domain knowledge, using 3 clusters may not be ideal for our goal of developing a song recommendation system. Instead, we opted for 5 clusters to introduce more variation in recommendations while maintaining a tolerable drop in evaluation metrics. The K-means clustering result is shown in figure 3.

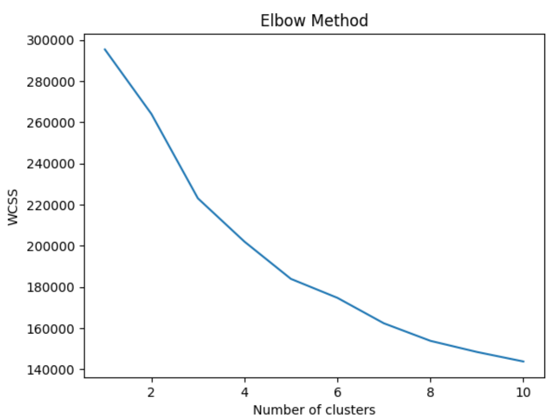


Fig. 2. K-means Elbow Method Result

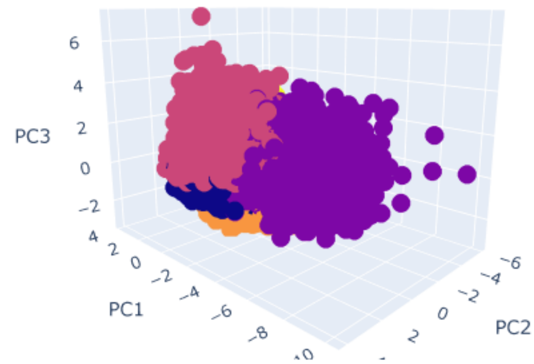


Fig. 3. K-means Clustering Result

### C. Agglomerative Clustering Result

To find the best K for Agglomerative Algorithm, we evaluated different silhouette score for k values. Figure 4 shows the Silhouette scores for Agglomerative Algorithm.

The results indicate that 2 would give the highest Silhouette score, however 2 would be very general and will not give relevance to our purpose. Thus we opted for 6 as shown in figure 5.

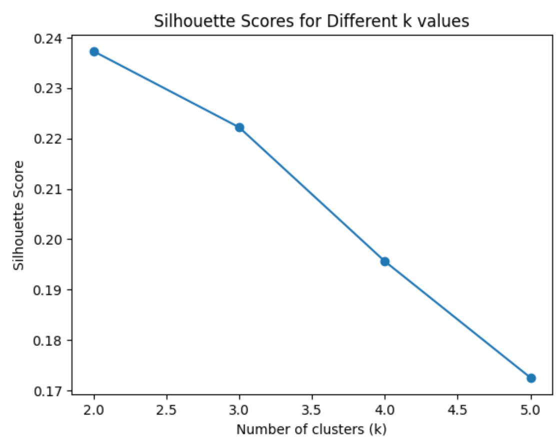


Fig. 4. Agglomerative Silhouette Scores

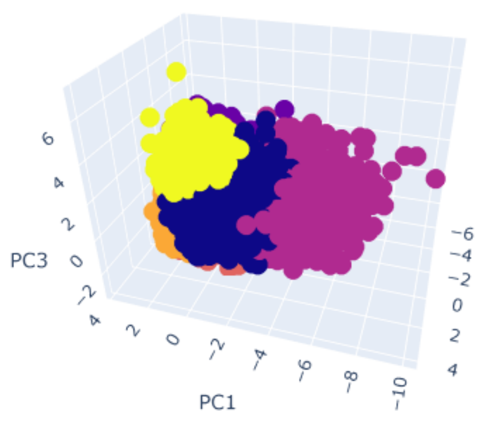


Fig. 5. Agglomerative Clustering Result

### D. DBSCAN Clustering Result

We Find the ideal epsilon and K values in DBSCAN Algorithm by manual searching. After manually searching for the best parameters. We got a lot of clusters. This is due to outliers being its own cluster. To fix this, all outliers was placed into a single cluster, which brings the number of clusters to 3 as shown in figure 6.

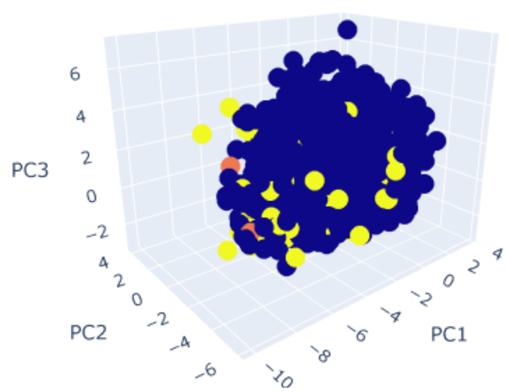


Fig. 6. DBSCAN Clustering Result

### E. Model Performance Evaluation

We used Silhouette score to evaluate our chosen number of cluster for each algorithm. Our K-means model has achieved a silhouette score of 0.24, Agglomerative 0.18, and DBSCAN of 0.36. We compared this with models of existing studies. We could only find a study utilizing K-means, but none in Agglomerative and DBSCAN as shown in Table I.

TABLE I
SILHOUETTE SCORES

| Research Studies | Algorithms | Silhouette score |
|---|---|---|
| Existing Studies | K-means | 0.39 |
| | Agglomerative | NA |
| | DBSCAN | NA |
| Our Study | K-means | 0.24 |
| | Agglomerative | 0.18 |
| | DBSCAN | 0.36 |

### F. Recommendation System Result

DBSCAN has gained the highest silhouette score, however the clustering is generalized into a single cluster and there is only 3 clusters, Agglomerative has 6 clusters and can be a choice, however K-means offers flexibility while having a high silhouette score, thus we used the k-means. we analyzed patterns that formed in the model built and used it to build the recommendation system.

figure 7 shows that Each cluster observed from the K-means model contains all genres. There is a noticeable gap in distributions of each clusters. The biggest cluster is cluter 1 and offers a diverse range of genres. Cluster 2 contains a lot of r & b songs, cluster 3 contains a lot of rap songs, cluster 4 is the smallest cluster and primarly contains edm, and cluster 5 is the second biggest cluster that has a lot of rock and edm songs.

figure 8 shows the mean of musical attributes for each k-means cluster. We could observe that some attribute dominates specific clusters. Cluster 1 is dominated with danceability and valeance attributes, with a certain number of energy and loudness. Cluster 2 is primarily dominated with acousticness. Cluster 3 is dominated with spechiness, with danceability coming in second. Cluster 4 is dominated by instrumentalness and energy being second. Finally, cluster 5 presents an equal amount of energy, loudness, liveness, and tempo.

Based on the data presented, we can infer the characteristics of each cluster. Cluster 1 exhibits dominance in danceability and valence, indicating that the songs in this cluster are highly danceable and have a positive mood. This suggests that Cluster 1 primarily consists of happy songs that are suitable for dancing. Cluster 2 has a high score for acousticness, which means that the tracks in this group are mainly played using acoustic instruments. As a result, the songs in this cluster are likely more traditional and less reliant on modern, electronically produced sounds. Cluster 3 has the highest speechiness, with danceability coming in second. This suggests that songs in this cluster contain a significant amount of vocals and are also suitable for dancing. However, unlike Cluster 1, the valence in this group is moderate, meaning that the songs range from happy to sad. Cluster 4 is characterized by the highest levels of instrumentalness and energy, indicating that this cluster consists mainly of intense, instrument-heavy songs. Cluster 5 presents a mix of energy, loudness, liveness, and tempo, suggesting that the songs in this cluster are loud, energetic, and often performed live.

After analyzing the patterns in genres and attributes of each cluster, we implemented a recommendation system to provide users with relevant song suggestions based on our model. To test its effectiveness, we searched for the track ”thank u, next” by Ariana Grande. The system generated ten song recommendations, including popular tracks such as ”Act Up,” ”What a Heavenly Way to Die,” and ”No Games.”. The result is show in figure 9.

Overall, the recommendations appeared to be relevant, as they shared characteristics with the searched track. However, the final assessment of the system’s effectiveness ultimately depends on user experience and whether the suggested songs align with their expectations.

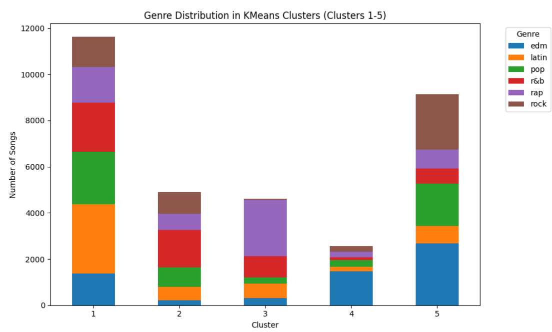


Fig. 7. Genre distribution in K-means

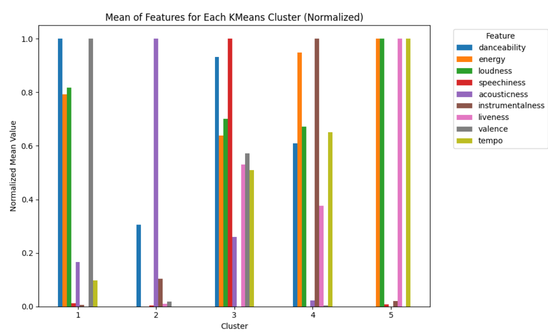


Fig. 8. Mean of each Features for each K-means cluster

| | track_name | track_artist |
|---|---|---|
| 23036 | Act Up | City Girls |
| 30040 | Stray Lines | Mishaal |
| 11539 | In My House | Mary Jane Girls |
| 20303 | UN PESO | J Balvin |
| 14427 | What A Heavenly Way To Die | Troye Sivan |
| 12963 | Rock And Roll All Nite | KISS |
| 30215 | RITMO (Bad Boys For Life) | The Black Eyed Peas |
| 15824 | Chance | Davide Cristofori |
| 7569 | No Games | Soulja Boyz |
| 14104 | Saigon - 2002 Digital Remaster | Martha and the Muffins |

Fig. 9. Recommendations Based on Searched Song

## V. Conclusion

This research explores an attribute-based clustering approach to understanding song recommendation systems, where recommendations are derived solely from the intrinsic characteristics of musical compositions rather than user behavior. We used a dataset from Kaggle, comprising songs obtained from the Spotify API. Initially, we explored the data and identified significant correlations among musical attributes. To enhance efficiency, Principal Component Analysis (PCA) was applied to reduce dimensionality while retaining variance. The results showed that the features were highly compact.

Using the principal components from PCA, we implemented three clustering algorithms: K-means, Agglomerative Clustering, and DBSCAN. After clustering, we assessed the quality of the clusters using the Silhouette score and our domain knowledge. We selected the final clustering parameters by balancing the requirements of our recommendation system with clustering performance. The best algorithm was determined based on the Silhouette score, followed by an analysis of similarities and differences among the clusters. We observed that certain clusters had dominant genres and attributes, though the distribution of clusters was not entirely equal.

The K-means model achieved a Silhouette score of 0.24, Agglomerative Clustering scored 0.18, and DBSCAN scored 0.36. K-means was chosen for the development of the recommendation system as it provided the best balance between structure and accuracy. Analyzing the clusters of the K-means model, we found that the genres in each cluster were diverse, although there was a significant dominance of certain musical attributes.

Finally, we built the recommendation system to simulate the user experience. The system generated a diverse selection of music genres, reflecting the attribute-based clustering approach. Since the model relied solely on music attributes, the recommendations were shaped by how the songs were grouped based on their intrinsic features. While music preference is highly subjective, creating a system that aligns with users’ subconscious preferences can be highly beneficial.

Through this study, we gained valuable insights into attribute-based clustering for music recommendations. For future work, researchers could explore a larger dataset with more songs to introduce greater variance. Additionally, other model evaluation techniques could be implemented to ensure robustness. Since music preference varies among individuals, domain knowledge could be further leveraged to extract the most relevant features. Future studies may also incorporate user feedback and perception analysis through subjective evaluation, allowing for a more user-centered recommendation system.

## REFERENCES


[1] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*. Springer, 2015.

[2] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, Hong Kong, China, 2001, pp. 285–295.

[3] J. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, "Facing the cold start problem in recommender systems," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 2065–2073, 2014.

[4] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[5] X. Hu, J. S. Downie, K. West, and A. F. Ehmann, "Mining music reviews to understand music perception," in *Proc. 10th Int. Soc. Music Inf. Retrieval Conf. (ISMIR)*, Kobe, Japan, 2009, pp. 201–206.

[6] D. Paul and S. Kundu, "A survey of music recommendation systems with a proposed music recommendation system," in *Proc. Int. Conf. Comput. Intell.*, 2020, pp. 345–356. doi: 10.1007/978-981-13-7403-6_26.

[7] International Journal of Computer Trends and Technology. (2023). Music Recommendation System. Retrieved from IJCTT

[8] Chen,P.;Dong,L.;Liu,Y. (2024). Design of music recommendation system based on EDA and K-means cluster analysis. Applied and Computational Engineering,50,317-323

[9] Jin, L., & Han, P. (2020) – Clustering-Based Latent Factor Model for Enhancing Collaborative Filtering in Music Recommendation. Retrieved from: iieta.org

[10] R. O'Brien, "Song discovery with K-means clustering: A music recommendation system," *Medium*, Apr. 10, 2023. [Online]. Available: https://medium.com/@roryobrien03/song-discovery-with-k-means-clustering-a-music-recommendation-system-2481b2fa425c.

[11] Brijmohan Daga, Harsh Kadam, Sarthak Shrungare, Scott Fernandes, Ashwin Johsnon, "Music Recommendation System," International Journal of Computer Trends and Technology, vol. 71, no. 5, pp. 26-36, 2023. Crossref, https://doi.org/10.14445/22312803/IJCTT-V71I5P105

[12] J. Arvidsson, "30000 Spotify Songs," Kaggle, 2022. [Online]. Available: https://www.kaggle.com/datasets/joebeachcapital/30000-spotify-songs. [Accessed: Mar. 3, 2025]

[13] Jolliffe, I. T., and Cadima, J., "Principal component analysis: a review and recent developments," Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 374, no. 2065, pp. 20150202, 2016.

[14] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), New Orleans, LA, USA, 2007, pp. 1027–1035.

[15] J. H. Ward Jr., "Hierarchical grouping to optimize an objective function," Journal of the American Statistical Association, vol. 58, no. 301, pp. 236–244, 1963.

[16] F. Murtagh and P. Legendre, "Ward's hierarchical agglomerative clustering method: Which algorithms implement Ward's criterion?," Journal of Classification, vol. 31, no. 3, pp. 274–295, 2014.

[17] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining (KDD), Portland, OR, USA, 1996, pp. 226–231.

[18] T. Kodinariya and P. Makwana, "Review on determining the number of clusters in K-Means clustering," Int. J. Adv. Res. Comput. Sci. Manage. Stud., vol. 1, no. 6, pp. 90–95, 2013.

[19] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," J. Comput. Appl. Math., vol. 20, pp. 53–65, 1987.