Clustering Analysis of Music Tracks Dataset

Main Objective of the Analysis

The primary objective of this analysis is to apply unsupervised learning techniques, specifically clustering, to explore the inherent patterns within the music tracks dataset. The analysis focuses on clustering the data based on numeric features such as track attributes and length. Additionally, dimensionality reduction techniques, particularly Principal Component Analysis (PCA), were utilized to reduce the number of features while maintaining the important information, which made the clustering process more efficient. By grouping similar tracks together, we aim to identify patterns that could provide valuable insights for stakeholders such as music streaming platforms, artists, and record labels. Clustering can help uncover trends, optimize recommendations, and improve user engagement based on track characteristics.

Brief Description of the Dataset

The dataset used in this analysis contains several music tracks with various features. These features include numeric values such as track length, popularity, energy, danceability, and more. The dataset also includes non-numeric columns like `Genre`, `Title`, `Album_cover_link`, `Artist`, `explicit`, `release_date`, and `release_date_precision`. Since non-numeric columns were deemed either irrelevant to the clustering or would require unnecessary encoding steps, they were removed. The dataset was then pre-processed, and standard scaling was applied to the numeric features to ensure the clustering algorithms would perform optimally. After scaling, PCA was performed to reduce the dimensionality of the dataset to the most essential features for clustering.

Summary of Data Exploration and Actions Taken for Data Cleaning or Feature Engineering

- **Data Cleaning**:

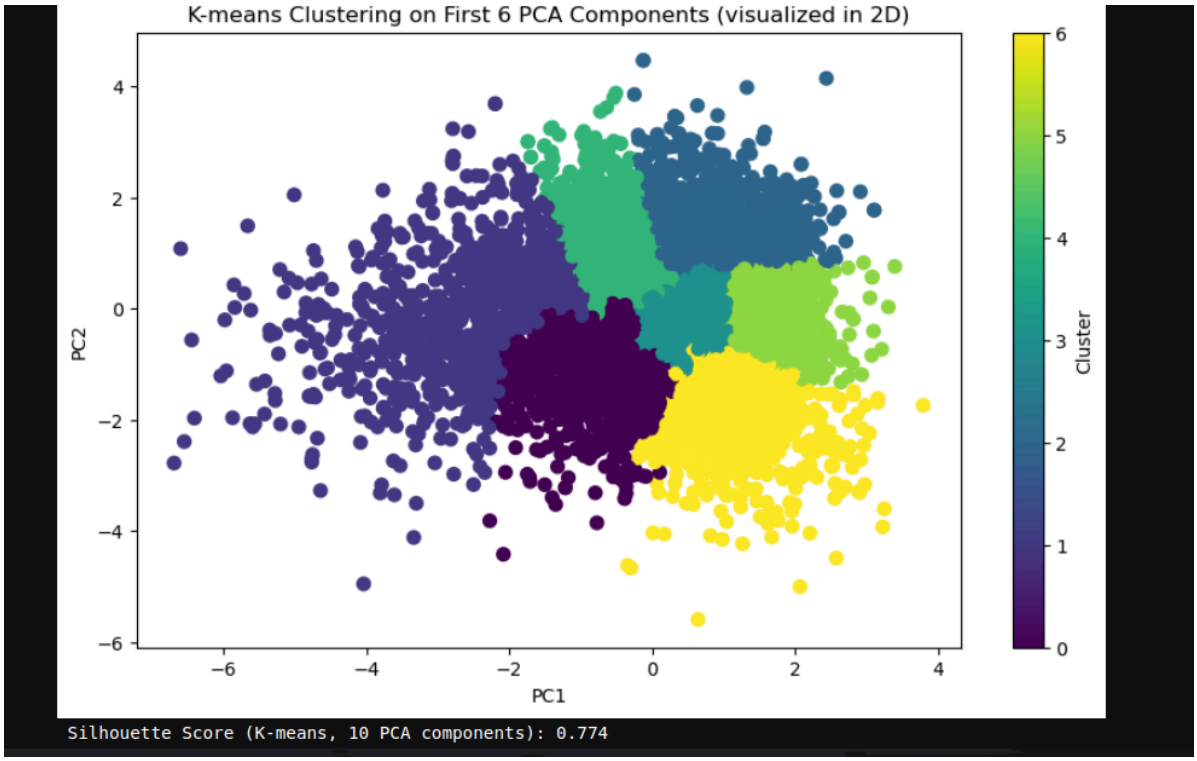
    - Non-numeric columns (`Genre`, `Title`, `Album_cover_link`, `Artist`, `explicit`, `release_date`, `release_date_precision`) were removed to simplify the dataset and avoid unnecessary encoding, which would not provide useful insights for clustering.

- **Feature Engineering**:
    - Standard scaling was applied to the data to ensure that all features had the same scale. This step was crucial for clustering algorithms like K-means, which are sensitive to the scale of the data.
    - PCA was performed to reduce the number of features from the original dataset, making the clustering process more efficient while retaining the most significant information.

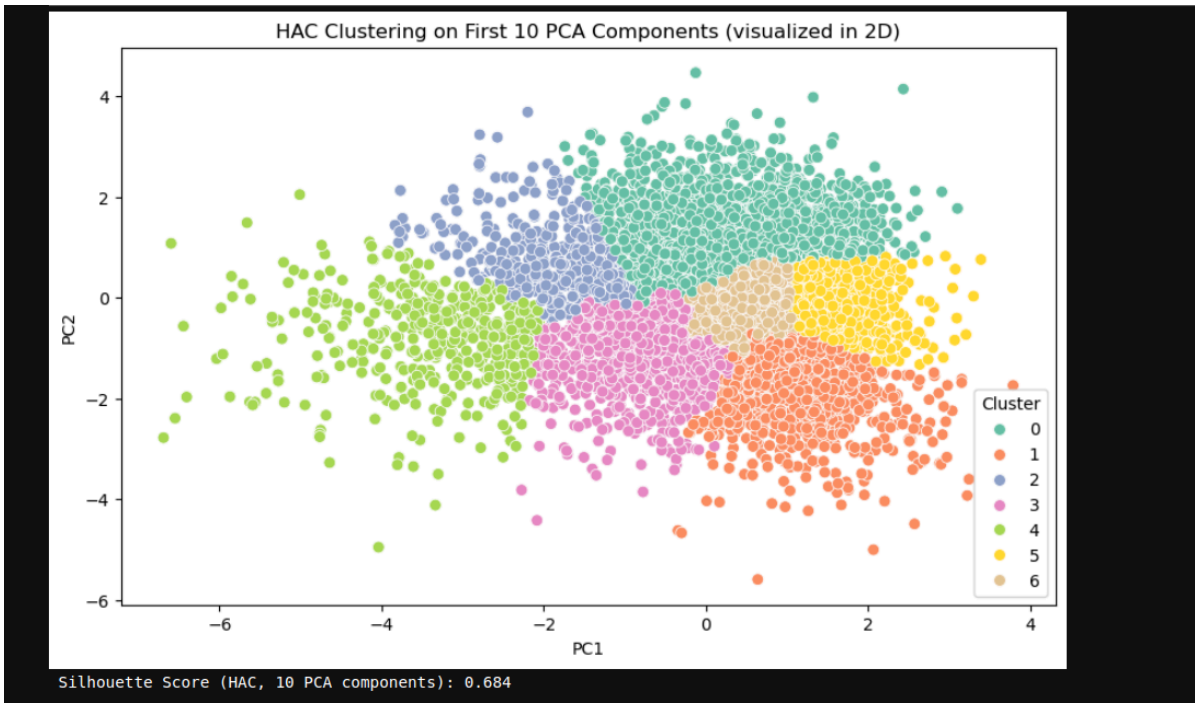
Summary of Training Clustering Models

Three different clustering algorithms were applied and evaluated on the dataset using silhouette score as the metric:

1. **K-means Clustering**:
    - K-means was applied with the number of clusters set to 8, which was determined through the WCSS (Within-Cluster Sum of Squares) method.
    - The silhouette score was used to evaluate the performance of K-means, helping us assess the cohesion and separation of the clusters.
    - Score : 0.774
    -



2. **Hierarchical Agglomerative Clustering (HAC)**:
    - The `ward` linkage method was chosen for hierarchical clustering, which aims to minimize the variance within clusters. This method was applied, and the silhouette score was calculated to determine the clustering quality.
    - score : 0.684
    -



3. **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**:
    - DBSCAN was particularly effective in this analysis due to its ability to find clusters of arbitrary shape and handle noise in the data. The model performed well, with minimal noise in the results, making it the most reliable algorithm for this dataset.
    - score :0.798

Recommended Model

After testing the models, **DBSCAN** is the recommended clustering algorithm for this analysis. Despite the other models (K-means and HAC) showing reasonable performance, DBSCAN outperformed them with a significantly higher silhouette score. Additionally, DBSCAN's ability to detect clusters of varying shapes and handle noise effectively makes it the best choice for this dataset.

Key Findings and Insights

- **Dimensionality Reduction**: PCA successfully reduced the dimensionality of the dataset from its original state to a more manageable size while retaining

key information. This simplification allowed clustering algorithms to perform more efficiently.

- **DBSCAN's Performance**: DBSCAN provided the best clustering results, with a high silhouette score and minimal noise in the clusters. The algorithm's strength lies in its ability to form clusters of varying shapes, making it highly suitable for this dataset.
- **Comparison with K-means and HAC**: While both K-means and HAC were able to generate clusters, their silhouette scores were lower than that of DBSCAN, indicating that DBSCAN created more cohesive and well-separated clusters.
- **Cluster Interpretation**: The final clusters produced by DBSCAN revealed distinct groups of music tracks with similar attributes, such as popularity, energy, and danceability. These groupings could be useful for recommending music to users with similar preferences.

Suggestions for Next Steps

- **Further Data Enhancement**: To improve clustering results, additional features such as user interaction data, track lyrics, or genre metadata could be included. This could provide deeper insights into the clusters.
- **Testing Advanced Models**: Although DBSCAN worked well, testing more advanced clustering models like Gaussian Mixture Models (GMM) or Spectral Clustering could offer a comparison in terms of performance.
- **Larger Dataset**: Re-training the model on a larger, more diverse dataset could help in obtaining better and more generalized clustering results.
- **Model Refinement**: Fine-tuning the hyperparameters for DBSCAN (such as `eps` and `min_samples`) could further improve clustering quality.

Conclusion

This clustering analysis applied unsupervised learning techniques to the music tracks dataset, successfully reducing the dimensionality using PCA and clustering the tracks into distinct groups. DBSCAN emerged as the best-performing model, providing a higher silhouette score and minimal noise compared to K-means and HAC. These findings can be used to enhance recommendation systems and

improve music discovery platforms by grouping similar tracks together. Further steps could involve adding more features and experimenting with different clustering algorithms for improved accuracy.