STREAMFY

AUTOMATED PLAYLIST GENERATION USING CLUSTERING OF SPOTIFY SONGS

Benedict Ouma

28th July 2025

www.streamfy.com

INTRODUCTION

About Streamify:

Streamify is an innovative music-tech startup focused on revolutionizing the way users discover and enjoy music. At the heart of this mission is a personalized experience - delivering the right music at the right moment.

Why This Project?

- Today's music listeners have diverse, evolving tastes that cannot be fully captured through manual playlist curation.
- Users often rely on playlists for workouts, study sessions, relaxation, or parties - each requiring a different "vibe" or energy level.

Project Objective:

- Leverage unsupervised machine learning to group songs with similar audio features.
- Automatically generate playlists by clustering songs into meaningful categories.
- Enhance user satisfaction and music discovery through intelligent automation.

PROBLEM STATEMENT

The Challenge:

Manual playlist creation is time-consuming, subjective, and not scalable for millions of songs.

Why a New Approach?

- Music has complex dimensions - energy, tempo, mood, danceability - that go beyond ger artist.
- Users expect personalization, speed, and relevance.

Our Solution:

- Use K-Means and Hierarchical Clustering to group songs based on their musical feature
- Label each cluster with intuitive, mood-based names [e.g., Chill Vibes, High Energy Dance
- Recommend songs from each cluster for use in curated playlists.

Project Goals:

- Identify distinct clusters or "vibes" within the song dataset.
- Assign clear and meaningful labels to each cluster.
- Build a recommendation logic that assigns any new song to its appropriate cluster.
- Provide sample song recommendations from each cluster.

DATA DESCRIPTION

Source: Data was retrieved from Spotify's Web API, consisting of 140,000 songs and 21 columns.

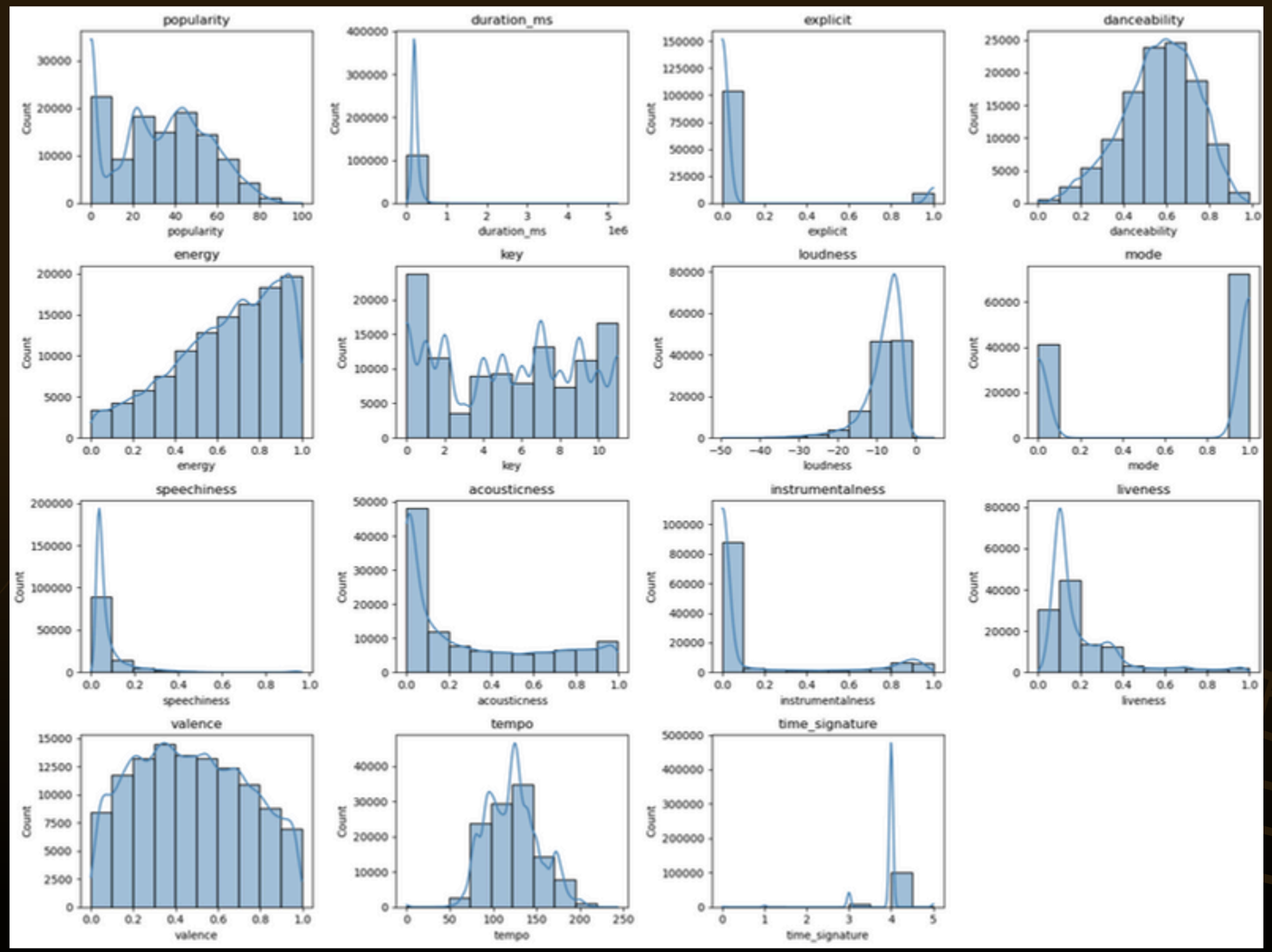
Key Features Used:
Selected audio-based numerical features included danceability, energy, tempo, valence, acousticness, along with loudness, speechiness, instrumentalness, liveness.

Removed Columns: "Unnamed: 0" [CSV row label], and non-numeric identifiers: track_id, track_name, artist_name.

Duplicates & Missing Values: 450 exact duplicate rows dropped - negligible impact on a dataset of this size. Single missing values in artist_name, album_name, and track_name verified and removed.

Preprocessing: StandardScaler was used to normalize all numeric features for fair clustering.

EXPLORATORY DATA ANALYSIS

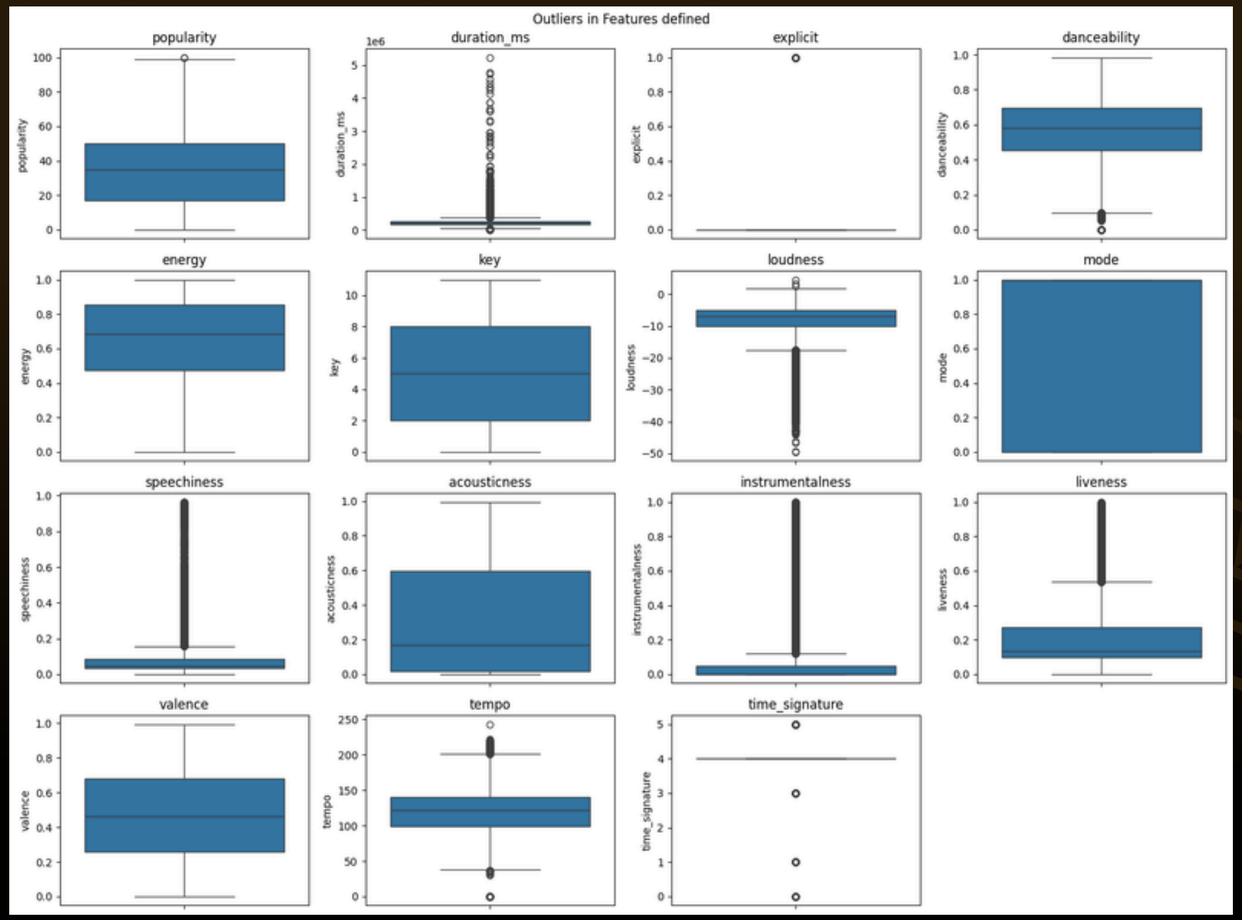


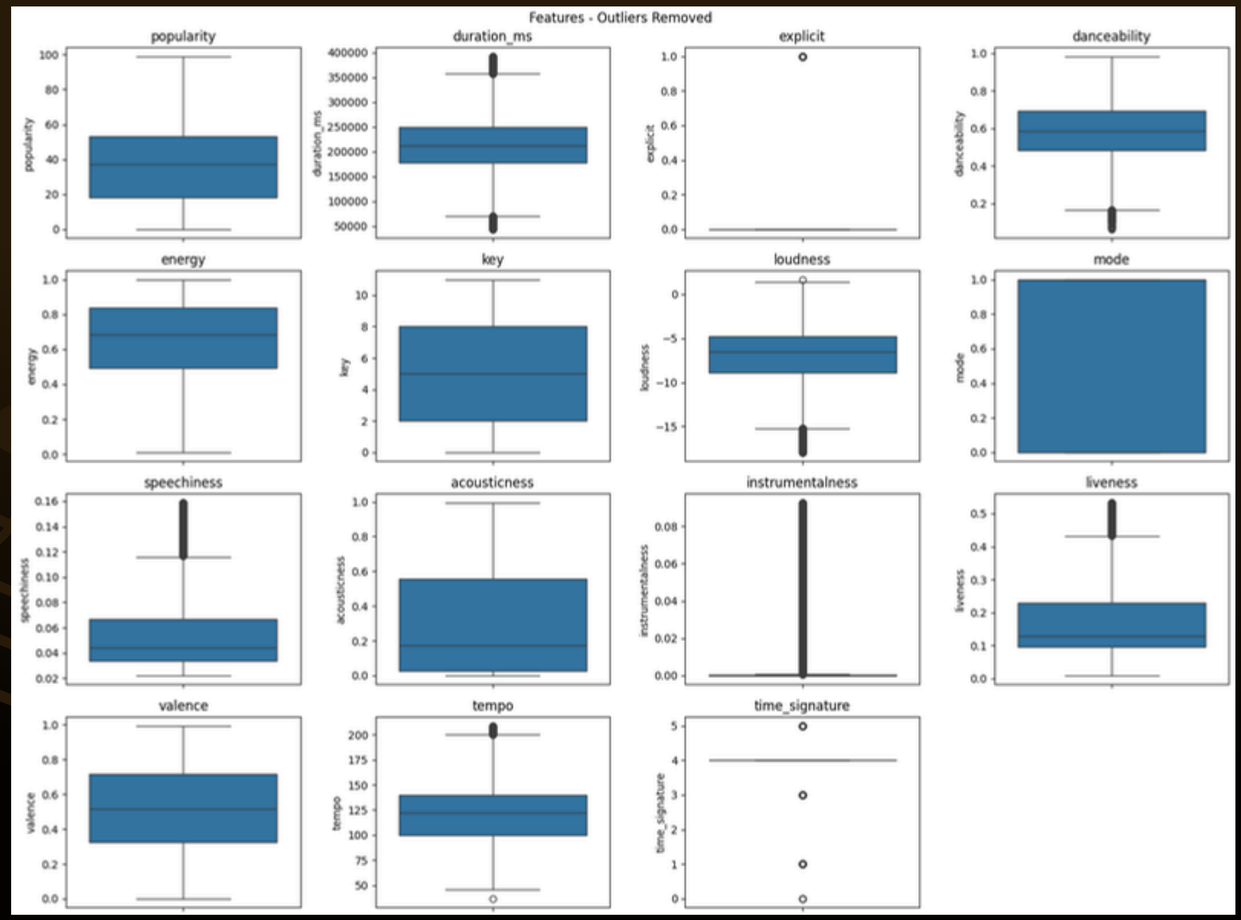
FEATURE DISTRIBUTION - HISTPLOT

The pair grid displays distributions of audio features from the music dataset revealing key patterns. Popularity is right-skewed, with most songs having low to moderate popularity. Duration_ms clusters between 3 to 5 minutes, though some outliers exist. Explicit and mode are binary, with most songs being non-explicit and in a major key. Danceability is normally distributed around 0.6, while energy is left-skewed, with many high-energy songs. Key is fairly uniform, and loudness centers around -10 dB. Speechiness, acousticness, and instrumentalness are all highly right-skewed, indicating most songs have low values for these features. Liveness suggests mostly studio tracks, though some are live. Valence is evenly spread, tempo is multimodal with peaks around 90, 120, and 150 BPM, and time_signature is mostly 4/4. Overall, the dataset reflects that most songs are upbeat, non-explicit, studio-recorded, and follow common musical patterns.

FEATURE COMPARISON - WITH AND WITHOUT OUTLIERS

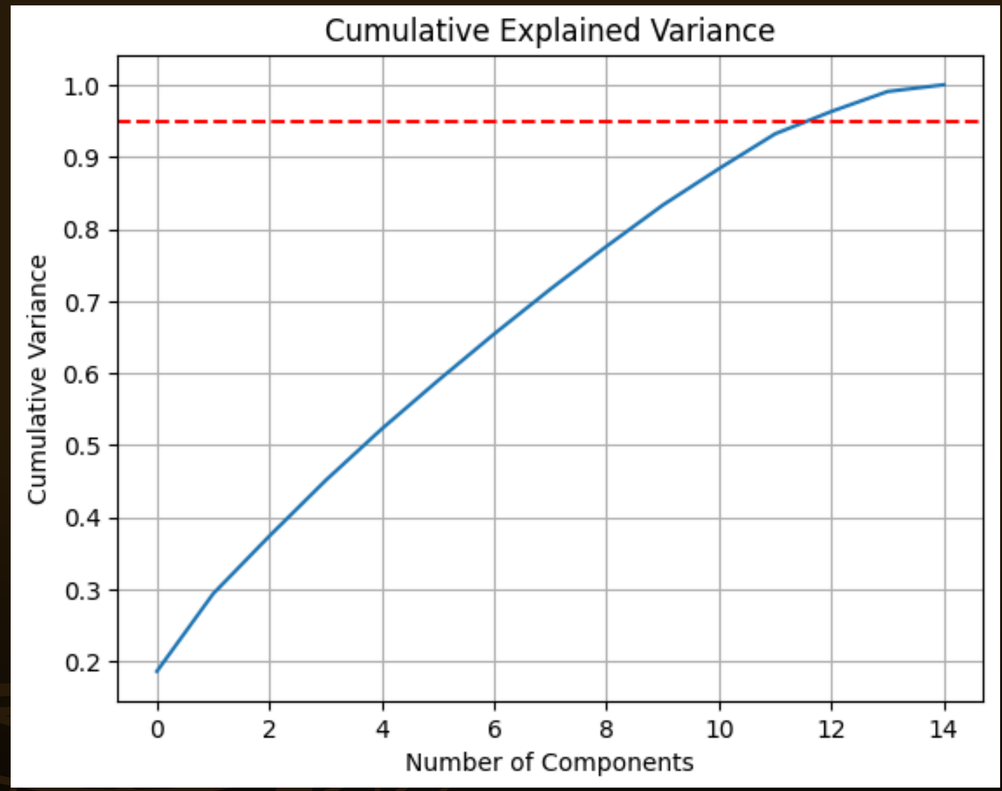
The left image displays box plots for 15 audio features, highlighting the distribution and variability across the dataset. Features like popularity, energy, and valence are tightly clustered around the median, while others such as duration_ms, speechiness, loudness, and time_signature show wider spreads with many outliers. The right image shows these distributions after outlier removal, revealing a noticeable reduction in extreme values - particularly for duration_ms, speechiness, and loudness. This process results in cleaner, more centered data with less skewness, improving overall quality and suitability for clustering.



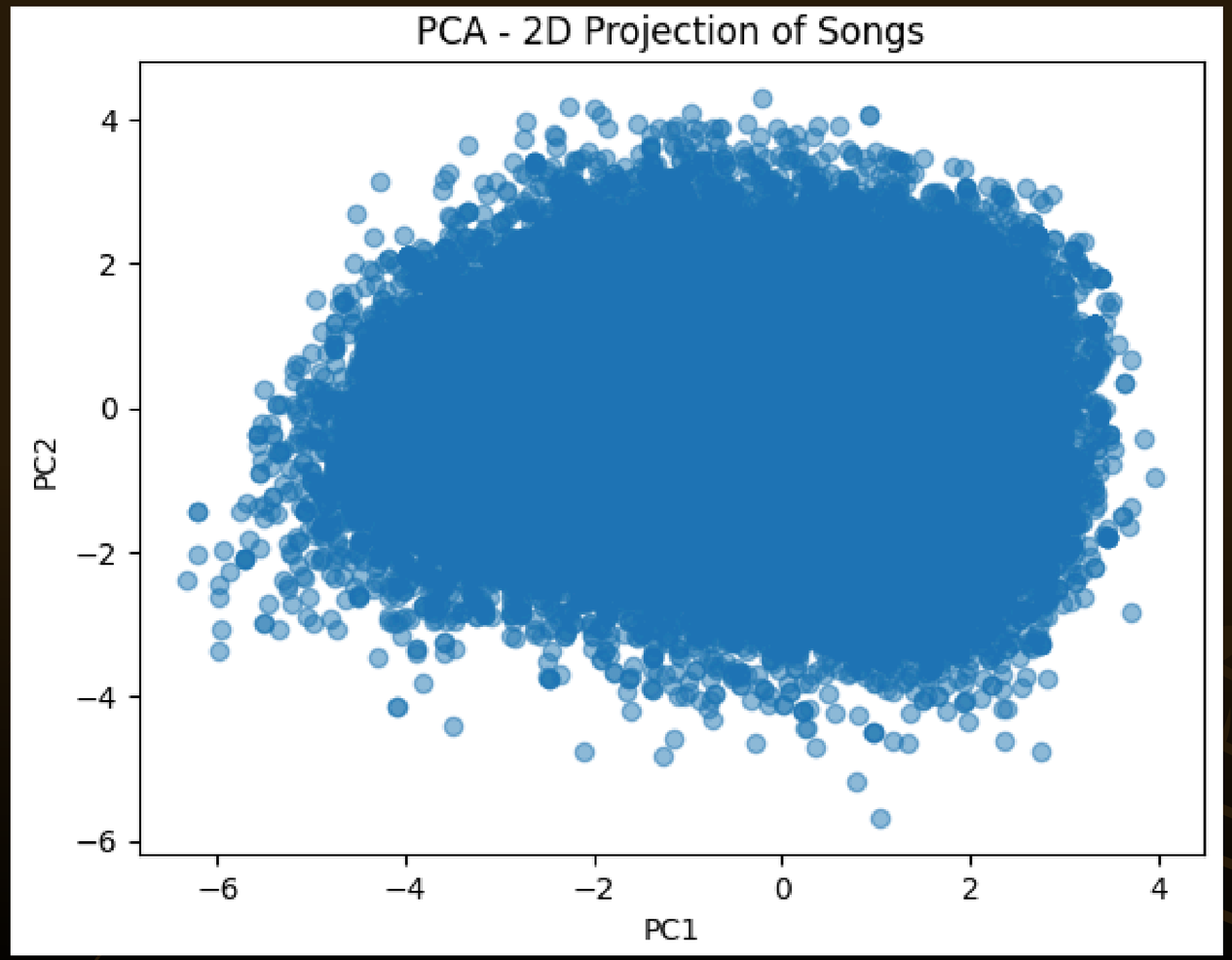


CUMULATIVE EXPLAINED VARIANCE

The image displays a line graph showing the cumulative explained variance against the number of components. The graph plots the increase in explained variance as the number of components used increases. A red dashed horizontal line represents a threshold, likely indicating a desired level of explained variance (around 0.95). The curve approaches this threshold, suggesting that a sufficient number of components are being used to capture most of the variance in the data. The graph is a visual representation of dimensionality reduction techniques, commonly used in data science and machine learning.



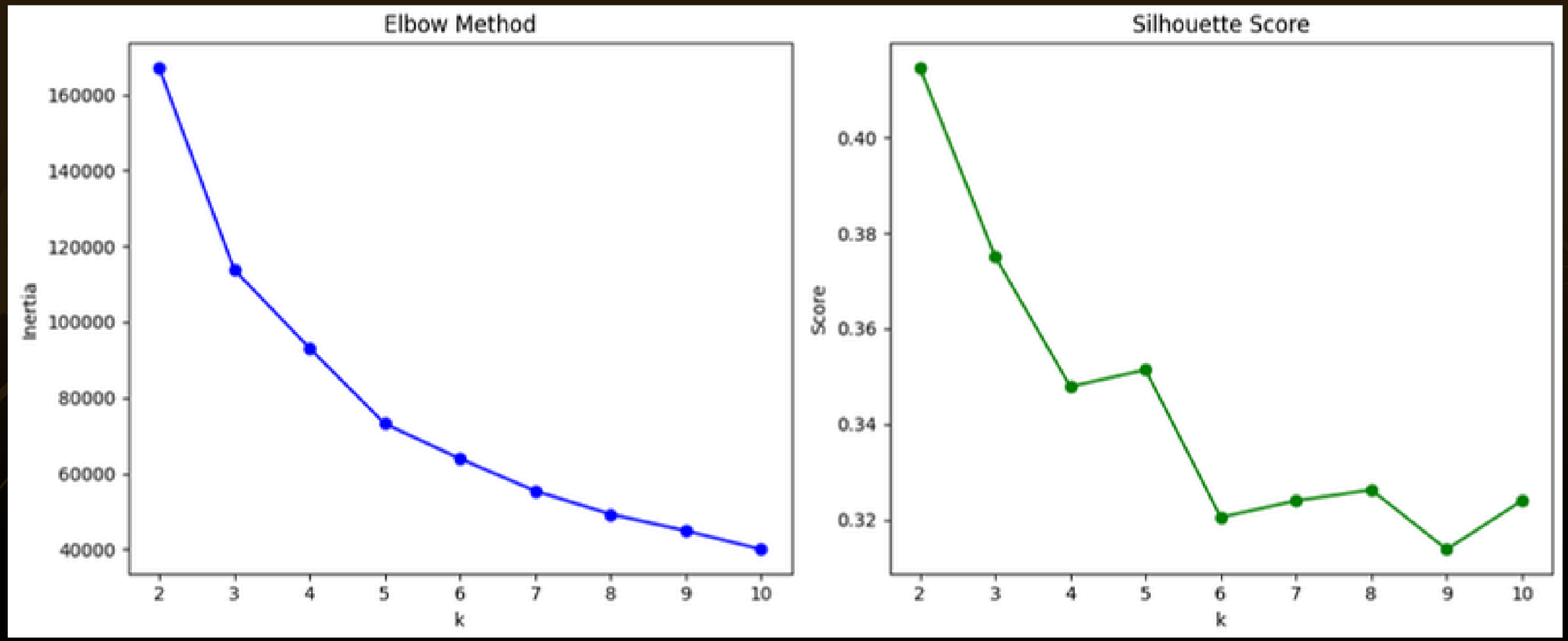
2D PCA PROJECTION



The PCA plot maps all Spotify tracks onto the first two principal components, capturing most of the variability in key musical features like tempo, energy, valence, and acousticness. The wide distribution of data points reflects the variety of musical styles present, while denser regions point to inherent groupings of tracks with similar audio characteristics. This demonstrates that PCA is a powerful tool for visualizing complex datasets and provides a solid foundation for subsequent clustering techniques such as K-Means.

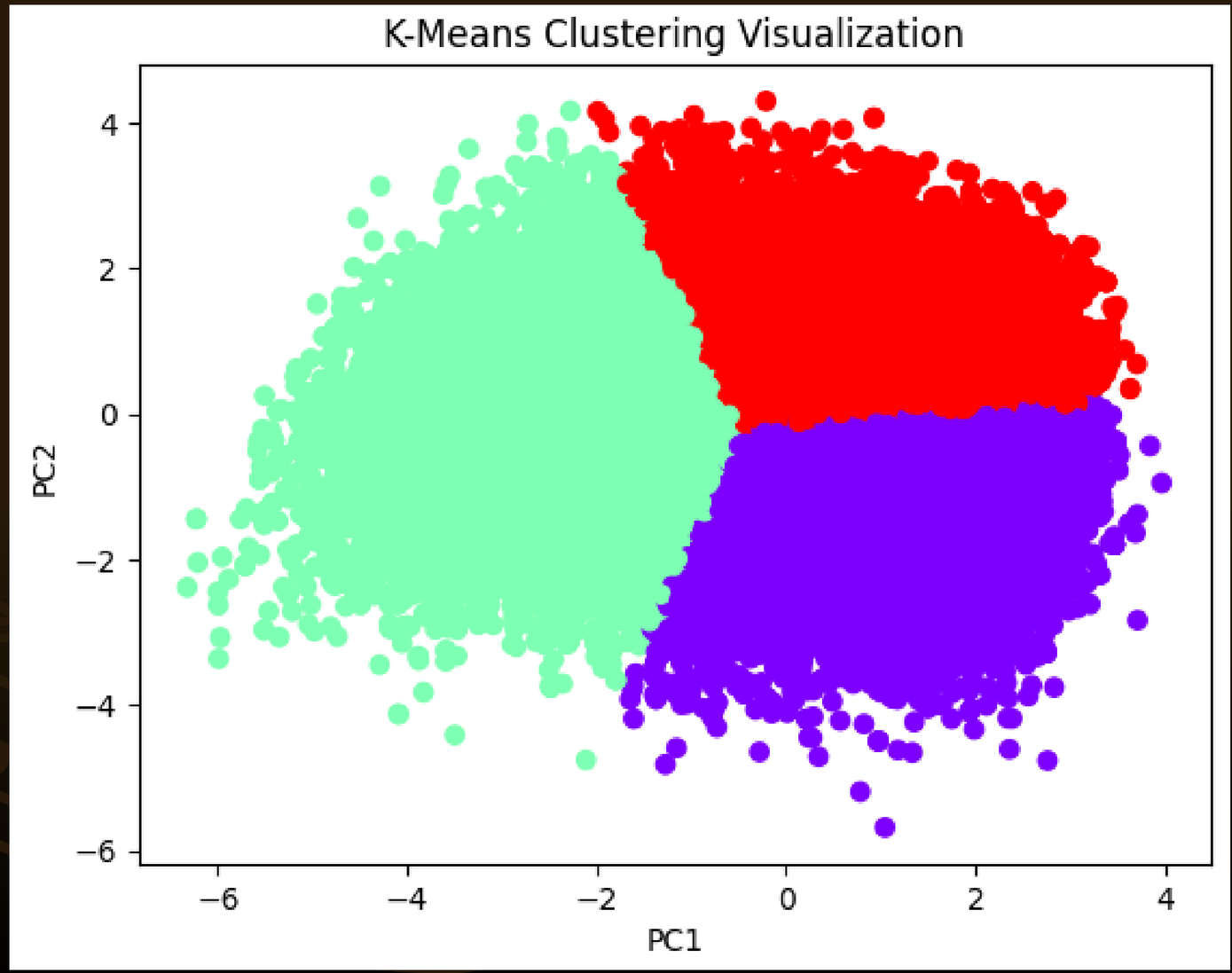
OPTIMAL K USING ELBOW METHOD AND SILHOUETTE SCORE

The Elbow plot shows a sharp drop in inertia up to k = 4, after which the curve flattens, suggesting 4 as an optimal point where adding more clusters yields minimal improvement. Meanwhile, the Silhouette Score plot peaks at k = 2, indicating the best cluster cohesion and separation at that point. While k = 2 offers the most distinct clustering, k = 4 provides a good balance between compactness and model complexity. Depending on the analytical goal, either can be justified as optimal.

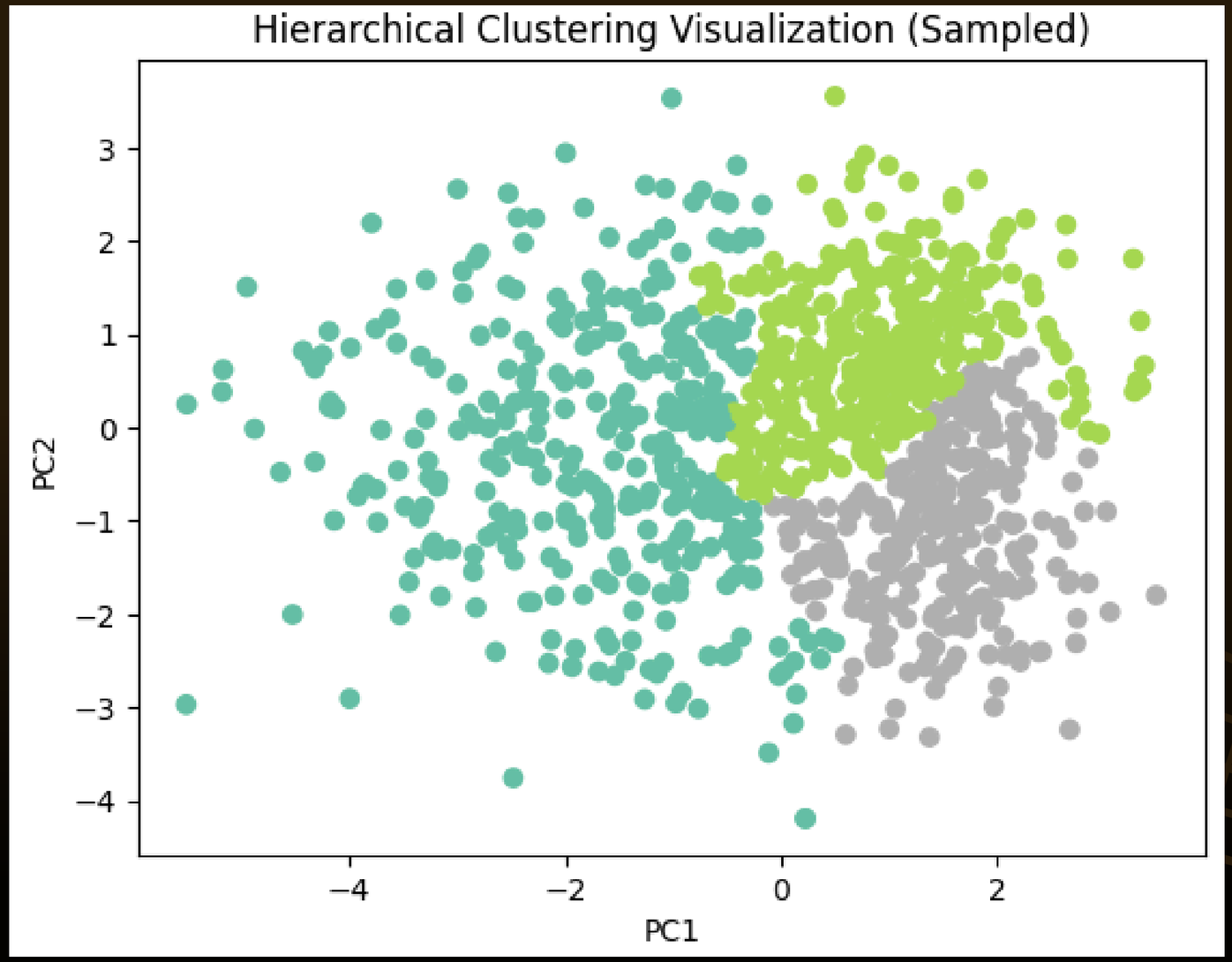


KMEANS CLUSTERING VISUALIZED

This chart compares clustering performance across k=2 to k = 5 using two metrics: Silhouette Score [blue, higher is better] and Davies-Bouldin Index [red, lower is better]. At k = 2, the Silhouette Score is highest, indicating well-separated clusters, but the Davies-Bouldin Index is also highest, suggesting poor inter-cluster distinction. At k = 3, the Silhouette Score slightly drops but remains acceptable, while the Davies-Bouldin Index improves significantly - indicating a better balance. At k = 4, both metrics worsen, and at k = 5, the Davies-Bouldin Index reaches its lowest [best], though the Silhouette Score is at its lowest too. Overall, k = 3 appears to offer the most balanced clustering quality based on both metrics.

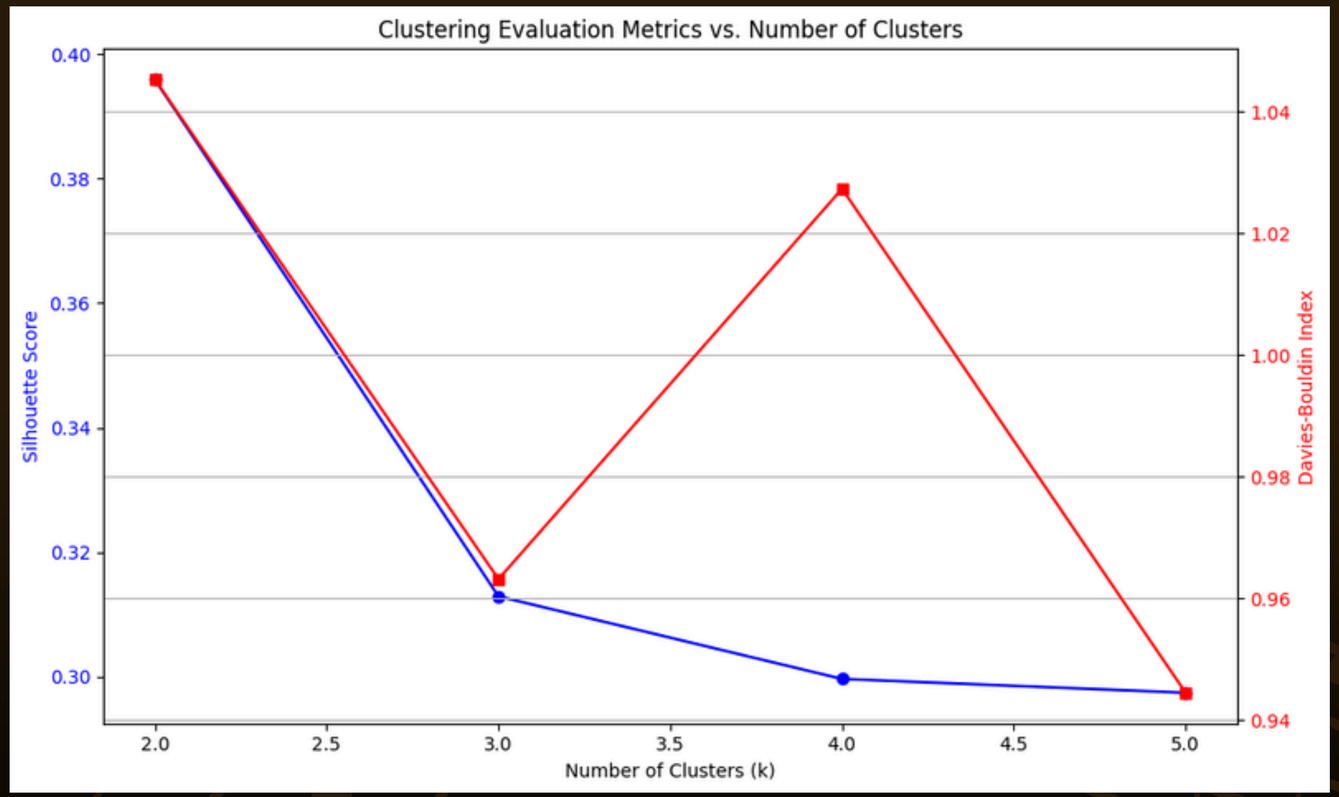


HIERACHIAL CLUSTERING - SAMPLED



This scatter plot shows the outcome of hierarchical clustering projected onto two principal components [PC1 and PC2], where each point represents a data entry and colors indicate cluster membership. The x and y axes correspond to PC1 and PC2, which capture the most variance in the dataset. The plot reveals three distinct clusters: one clearly separated on the right [gray], another on the left [teal], and a third [green] positioned between them with slight overlap - indicating shared features with the neighboring clusters. Although there is some internal variation within each group, the clusters are generally well-formed, suggesting that hierarchical clustering effectively uncovered three meaningful patterns in the data.

CLUSTERING EVALUATION



This chart illustrates how clustering performance varies with different numbers of clusters [k] using two evaluation metrics: Silhouette Score [blue] and Davies-Bouldin Index [red]. The Silhouette Score, which favors higher values, peaks at k = 2 and steadily declines as k increases, indicating tighter clusters at lower k values. Conversely, the Davies-Bouldin Index, where lower values indicate better clustering, reaches its lowest point at k = 5, with a noticeable dip also at k = 3. Together, these trends help identify an optimal k by balancing high intra-cluster similarity and low inter-cluster overlap - suggesting that k = 3 might offer a good trade-off between the two metrics.

METHODOLOGY

CLUSTER LABELS

Cluster	Label	Characteristics
0	Chill Vibes	Low energy, acoustic, mellow
1	High Energy Dance	High tempo, danceable, electronic
2	Acoustic Mellow	Instrumental, slow, calming

PREDICTION FUNCTION

I built a function as shown below that predicts the cluster of new songs by taking in audio features such as tempo, valence, and energy as input, and outputting a corresponding cluster label - for example, “High Energy Dance” - based on the characteristics and patterns learned from the original clustering model.

```
def predict_song_cluster(song_features, model = kmeans, scaler = scaler, pca_model = pca_2d):
  features_scaled = scaler.transform([song_features])
  features_pca = pca_model.transform(features_scaled)
  cluster = model.predict(features_pca)[0]
  return cluster_labels[cluster]
```

SAMPLE PLAYLIST RECOMMENDATIONS FROM EACH OF THE 3 CLUSTERS

Acoustic Mellow Playlist:

- Track 92061 | Popularity: -1.4770141428200718 | Energy: -1.1111486112194775
- Track 1258 | Popularity: -0.412191836452575O7 | Energy: 1.041245000712867
- Track 78839 | Popularity: -0.454784728707274 93 | Energy: 0.8251050564184055

High Energy Dance Playlist:

- Track 34631 | Popularity: 1.6748598840277187 | Energy: -1.0391019631213239
- Track 96255 | Popularity: 0.3544802241320226 | Energy: -1.475884767216381
- Track 56828 | Popularity: 0.2692944396226229 | Energy: -2.1558250086427075

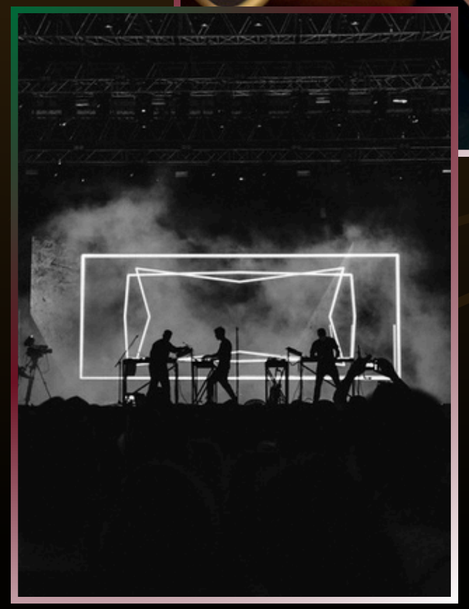
Chill Vibes Playlist:

- Track 34156 | Popularity: 0.3970731163867225 | Energy: 0.2082056320779635 8
- Track 72907 | Popularity: 0.5248517931508221 | Energy: 1.2258645364643863
- Track 2911 | Popularity: 1.121152284716620 4 | Energy: 0.280252280176117 4

MODELING & EVALUATION

We evaluated clustering performance using internal validation metrics. K-Means achieved a Silhouette Score of 0.35 and a Davies-Bouldin Index of 0.94, while Hierarchical clustering (using the same optimal number of clusters) yielded a slightly lower Silhouette Score of 0.31 and a higher Davies-Bouldin Index of 0.96. This indicates that K-Means formed more defined and compact clusters compared to Hierarchical clustering. Although both methods showed fairly similar trends, K-Means offered marginally better separation. Additionally, results remained stable across different random seeds and PCA initializations, affirming the consistency of the clustering process.

RESULTS & INSIGHTS



- **Optimal Clusters Identified:** Using evaluation metrics, K = 3 was selected as the optimal number of clusters, balancing both Silhouette Score and Davies-Bouldin Index.

- **Cluster Characteristics:** Each cluster represents a unique combination of musical attributes:
  - Cluster 0: High energy, danceable tracks (e.g., "High Energy Dance").
  - Cluster 1: Mellow, acoustic songs with low tempo.
  - Cluster 2: Balanced tracks with moderate features.

- **Model Stability:** Results remained stable across PCA transformations and multiple random seeds, confirming reliable clustering structure.

- **KMeans vs Hierarchical:**
  - KMeans outperformed with better internal metrics (Silhouette: 0.35, DBI: 0.94).
  - Hierarchical clustering had similar trends but weaker performance (Silhouette: 0.31, DBI: 0.96).

- **Practical Application:** A function was built to predict the cluster label for any new song using features like energy, tempo, and valence - enabling practical genre/style tagging.

RECOMMENDATIONS

- **Adopt KMeans Clustering:** Given its superior performance and interpretability, KMeans should be the preferred method for segmenting songs based on audio features.
- **Use Cluster Labels for Personalization:** Integrate predicted cluster labels into music platforms to recommend songs by mood, energy, or style.
- **Feature Enhancement:** Consider incorporating more features (e.g., lyrics sentiment, artist popularity) to improve cluster richness.
- **Monitor Cluster Evolution:** Re-run clustering periodically to accommodate changing music trends and styles.
- **Scalability:** Deploy the prediction function in production pipelines for real-time tagging of new songs uploaded to streaming platforms.


THE END
THANK YOU