AMAZON MUSIC CLUSTERING

Presented by: yogapriya

Amazon Music Clustering

Leveraging unsupervised machinelearning to automatically group millions of songs based on audio features like tempo, energy, danceability, and mood transforming how we discover and experience music.

The Challenge: Making Sense of Millions

Problem Statement

Withmillions of songs available on streaming platforms, manual genre classification becomes impractical and subjective. Traditional categorization fails to capture the nuanced audio characteristics that truly define a song's essence.

Our Goal: Automatically group songs based on quantifiable audio features creating data-driven clusters that reflect actual musical similarities.

Business Impact

- **Personalized Playlists:** Curate tailored listening experiences based on audio DNA
- **Smart Discovery:** Help users find hidden gems in similar clusters
- **Artist Insights:** Understand an artist's sonic evolution across albums
- **Market Analysis:** Identify trending musical patterns and audience segments

Skills & Technical Arsenal

Data Engineering

- Dataexploration & cleaning
- Feature selection & normalization
- Preprocessing pipelines

Unsupervised ML

- K-Means clustering
- DBSCAN algorithms
- Hierarchical clustering
- PCA dimensionality reduction

Visualization

- Cluster interpretation
- Feature analysis
- Insight communication

Technical Stack: Python, pandas, NumPy, scikit-learn (KMeans, DBSCAN, Agglomerative), Matplotlib & Seaborn

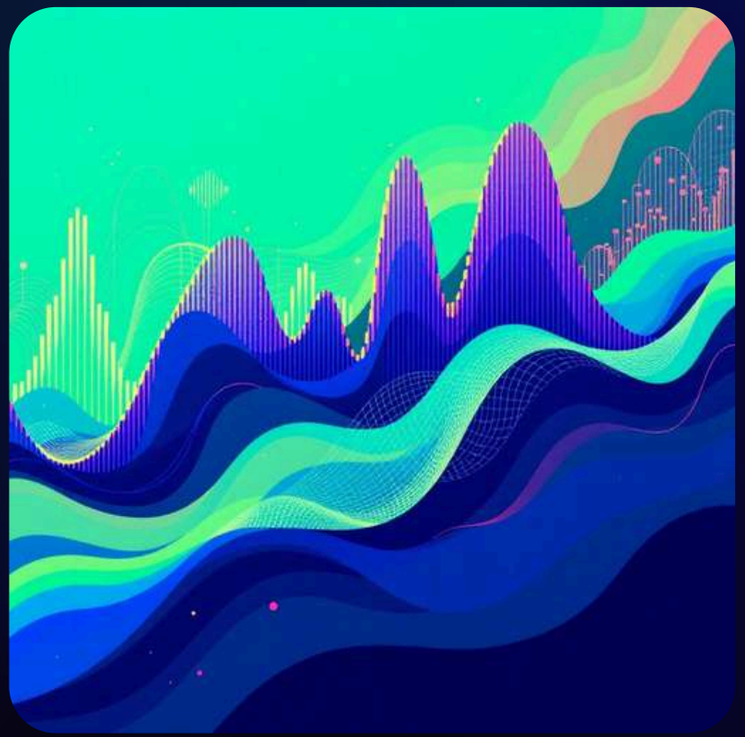
Dataset: 95,837 Songs Analyzed

Source:single_genre_artists.csv

Our analysis leveraged acomprehensivedatasetcontaining **95,837 songs** with 23 feature columns, capturing the essence of each track through quantifiable audio characteristics.

Key Audio Features

- **Danceability** :rhythmic suitability for movement
- **Energy** : intensity and activity level
- **Valence** : musical positivity (happy vs. sad)
- **Acousticness** : presence of acoustic instruments
- **Tempo** : overall pace (BPM)
- **Speechiness, Instrumentalness, Liveness**



Preprocessing Pipeline

1. Removed non-numeric identifiers (track_name, artist_name, id_songs)
2. Validated data integrity , no nulls or duplicates
3. Standardized features using StandardScaler for uniform scale

K-Means Clustering: Finding Natural Groups

01

Elbow Method Analysis

Plotted inertiaacross different k values to identify the optimal number of clusters where adding more clusters yields diminishing returns.

02

Silhouette Validation

Measured cluster separation quality to ensure meaningful groupings with minimal overlap between clusters.

03

Model Training & Assignment

Fitted KMeans algoritham assigned cluster labels to all 95,837 songs based on feature similarity.

Results: 3 Distinct Song Personalities

Silhouette Score: 0.242

Cluster 0
Chill / Acoustic
Low energy, high acousticness

Cluster 1
Mainstream / Party
High energy & danceability

Cluster 2
Happy / Dance
High valence, upbeat tempo

DBSCAN: Uncovering Hidden Patterns

Density-Based Clustering

UnlikeK-Means, DBSCAN identifies clusters of arbitrary shape and automatically detects outliers and songs that don't fit conventional patterns.

Methodology

1. Applied **PCA dimensionality reduction** to 2 components for computational efficiency
2. Tuned hyperparameters (eps, min_samples) using k-distance plot analysis
3. Fitted DBSCAN to identify core points, border points, and noise

4

Clusters Identified

Naturally formed groups based on feature density

72

Noise Points

Outlier songs with unique characteristics

0.394

SilhouetteScore

63% higher than K-Means superior cluster separation

Hierarchical Clustering: Building the Music Tree

Tree-Based Approach

Hierarchicalclustering builds a dendrogram 4 a tree structure showing how songs merge into larger groups at different similarity thresholds. This reveals the nested relationships between musical styles.

Implementation Details

- Sampled **3,000 songs** for computational efficiency
- Used **Ward linkage** to minimize variance within clusters
- Applied Agglomerative Clustering bottom-up approach

Results Overview

Final clusters identified

Silhouette score

The dendrogram visualization reveals the hierarchical merging process, showing which song groups are most similar and at what similarity threshold they combine into larger clusters.

Insights & Future Directions

Visualization Techniques

- **PCA2D ScatterPlots:**Songs colored by cluster membership
- **Feature Heatmaps:** Compare average characteristics per cluster
- **Bar Charts:**Analyze tempo, danceability distributions

Key Insights

Clusters successfully capture musical characteristics and moods, enabling intelligent recommendation systems and automated playlist generation based on audio DNA rather than subjective genres.

Real-Time Integration

Connect to Spotify / Amazon Music APIs for live data

Interactive Search

Let users explore song cluster placement

Personal Analysis

Upload listening history for customized insights

Global Trends

Visualize worldwide musical preferences

Thank You

"Data tells stories this helps decode music."