Unsupervised Music Genre Discovery Using Audio Feature Learning: A Comprehensive Multi-Dataset Analysis

Anirudh Sharma
Department of Computer Science and Engineering
National Institute of Technology, Hamirpur
Hamirpur, Himachal Pradesh, India
Roll No.: 22DCS002
Email: 22dcs002@nith.ac.in

***Abstract*—Music genre classification remains a fundamental challenge in Music Information Retrieval (MIR), traditionally approached through supervised learning methods that require extensive labeled datasets. This research presents a comprehensive unsupervised learning framework for automatic music genre discovery using audio feature learning across four diverse datasets: GTZAN (1,000 tracks), FMA-Small (6,410 tracks), Million Song Dataset (MSD) regional subset, and Spotify Tracks Dataset. We extracted 75-155 audio features per dataset including Mel-Frequency Cepstral Coefficients (MFCCs), chroma features, spectral characteristics, and temporal dynamics using the Librosa library. Our experimental framework evaluated five clustering algorithms—K-Means, MiniBatch K-Means, Spectral Clustering, Gaussian Mixture Models (GMM), and DBSCAN—across four train-test splits (50-50, 60-40, 70-30, 80-20) with multiple random seeds for robustness. Dimensionality reduction via Principal Component Analysis (PCA) retained 89-95% variance while reducing feature space by 70%. Performance evaluation employed six metrics: Silhouette Score, Davies-Bouldin Index, Calinski-Harabasz Index, Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), and cluster purity. K-Means and Spectral Clustering emerged as top performers with average silhouette scores of 0.109 and purity scores reaching 50% on FMA dataset. The GTZAN dataset achieved best results with NMI of 0.408 and accuracy of 43.6% using K-Means at 80-20 split. DBSCAN struggled with sparse feature spaces, yielding single-cluster solutions. Our multi-dataset analysis reveals that unsupervised methods can effectively discover latent genre structures, with performance varying significantly based on dataset characteristics, feature quality, and algorithm choice. This work contributes a rigorous experimental methodology, comprehensive evaluation framework, and insights into unsupervised genre discovery challenges, providing a foundation for future MIR research.**

***Index Terms*—Music Genre Classification, Unsupervised Learning, Audio Feature Extraction, Clustering Algorithms, MFCC**

I. Introduction

Music genre classification is a cornerstone problem in Music Information Retrieval (MIR) with applications spanning music recommendation systems, playlist generation, copyright management, and music archive organization [1]. Traditional approaches rely heavily on supervised learning methodologies requiring extensive labeled datasets, manual annotation by music experts, and domain-specific feature engineering. However, the subjective nature of genre boundaries, the emergence of hybrid genres, and the exponential growth of digital music content present significant challenges to supervised paradigms.

Unsupervised learning offers a compelling alternative by discovering inherent structures and patterns in audio data without relying on predefined labels [3]. This approach is particularly valuable for: (1) exploring large-scale unlabeled music archives, (2) discovering emergent or niche genres not present in training taxonomies, (3) reducing annotation costs and subjectivity bias, and (4) enabling cross-cultural music analysis where Western genre taxonomies may not apply.

A. Motivation

Despite advances in deep learning for music classification [4], several critical gaps persist in the literature:

- Limited comprehensive studies comparing multiple clustering algorithms across diverse datasets
- Insufficient analysis of feature engineering and dimensionality reduction impacts
- Lack of rigorous evaluation frameworks using multiple complementary metrics
- Inadequate exploration of robustness across different train-test splits and random initializations

This research addresses these gaps by presenting a systematic, multi-dataset evaluation of unsupervised genre discovery methods.

B. Research Objectives

The primary objectives of this study are:

1) To develop a comprehensive audio feature extraction pipeline using state-of-the-art signal processing techniques
2) To perform rigorous data analysis, cleaning, and preprocessing across four diverse music datasets
3) To compare five clustering algorithms (K-Means, MiniBatch K-Means, Spectral Clustering, GMM, DBSCAN) using six evaluation metrics

4) To analyze algorithm performance across multiple train-test splits with statistical robustness
5) To provide insights and recommendations for practical unsupervised music genre discovery applications

C. Contributions

This work makes the following key contributions:

- **Comprehensive Multi-Dataset Analysis:** First systematic evaluation across four diverse music datasets with different characteristics and scales
- **Rigorous Experimental Framework:** 60+ experiments per dataset with multiple splits, seeds, and algorithms ensuring statistical validity
- **Multi-Metric Evaluation:** Holistic assessment using six complementary metrics capturing different aspects of clustering quality
- **Reproducible Methodology:** Complete pipeline from raw audio to cluster evaluation with detailed hyperparameter specifications
- **Practical Insights:** Algorithm-specific performance characteristics and dataset-dependent recommendations for practitioners

D. Paper Organization

The remainder of this paper is organized as follows: Section II reviews related work in music genre classification and unsupervised learning. Section III details the datasets and preprocessing methodology. Section IV describes the implementation including feature extraction, dimensionality reduction, and clustering algorithms. Section V presents theoretical and mathematical foundations. Section VI analyzes experimental results with detailed performance comparisons. Section VII discusses limitations and future directions. Section VIII concludes the paper.

II. Related Work

A. Music Genre Classification

The seminal work by Tzanetakis and Cook [1] established the foundation for automatic music genre classification using timbral, rhythmic, and pitch-based features with classical machine learning algorithms. Their GTZAN dataset remains a standard benchmark despite known limitations including repetitions, mislabelings, and artist bias [2].

Recent advances have shifted toward deep learning approaches. Humphrey et al. [3] demonstrated feature learning from spectrograms using convolutional neural networks. Choi et al. [4] achieved state-of-the-art results on the Million Song Dataset using deep convolutional architectures with transfer learning. Pons et al. [5] explored musically motivated CNN architectures that capture timbral, temporal, and harmonic information at multiple time scales.

B. Unsupervised Learning in MIR

Unsupervised approaches in MIR have received comparatively less attention. McFee and Lanckriet [7] proposed heterogeneous metric learning for music similarity without genre labels. Nakashika et al. [8] used non-negative matrix factorization for unsupervised genre clustering. More recently, van den Oord et al. [9] introduced contrastive predictive coding for learning audio representations in an unsupervised manner.

Clustering-based approaches have shown promise for music structure analysis and similarity computation. Paulus and Klapuri [10] applied spectral clustering to music structure segmentation. Nieto and Jehan [11] used agglomerative clustering with perceptual linear prediction features for music similarity.

C. Feature Extraction and Representation

Traditional audio features include MFCCs [12], chroma features [13], spectral features (centroid, rolloff, flux, bandwidth), and temporal features (zero-crossing rate, tempo, onset strength) [1]. These hand-crafted features remain competitive baselines and are interpretable.

Recent work has explored learned representations through autoencoders [14], variational autoencoders [15], and self-supervised learning [6]. However, the interpretability-performance tradeoff remains an open research question.

D. Dimensionality Reduction

High-dimensional audio features pose challenges for clustering algorithms due to the curse of dimensionality. Principal Component Analysis (PCA) is widely used for linear dimensionality reduction [16]. Non-linear methods like t-SNE [17] and UMAP [18] have shown superior visualization capabilities but can distort global structure.

E. Research Gap

While prior work has explored individual aspects of unsupervised music analysis, comprehensive multi-dataset studies with rigorous evaluation across diverse clustering algorithms remain scarce. This paper addresses this gap through systematic experimentation and analysis.

III. Datasets and Preprocessing

A. Dataset Description

We evaluated our methodology on four diverse music datasets:

1) GTZAN Dataset: The GTZAN dataset [1] contains 1,000 audio tracks (30 seconds each) spanning 10 genres: blues, classical, country, disco, hiphop, jazz, metal, pop, reggae, and rock. Each genre has 100 tracks. Despite known issues [2], it remains the most widely used benchmark for comparison purposes.

2) FMA-Small Dataset: The Free Music Archive (FMA) [19] provides 8,000 tracks across 8 balanced genres, each 30 seconds long. We processed 6,410 tracks with complete metadata. FMA offers higher quality annotations and addresses many limitations of GTZAN.

3) Million Song Dataset (MSD): We utilized a regional subset of the MSD [20] containing pre-computed audio features. The dataset provides diverse music content but lacks explicit genre labels, making it ideal for unsupervised discovery.

4) Spotify Tracks Dataset: This dataset comprises audio features extracted via the Spotify API [21] for diverse tracks. Features include acousticness, danceability, energy, instrumentalness, liveness, loudness, speechiness, tempo, and valence—representing high-level musical attributes.

B. Data Analysis and Quality Assessment

1) Descriptive Statistics: For each dataset, we computed comprehensive descriptive statistics including:

- Central tendency measures: mean, median, mode
- Dispersion measures: standard deviation, variance, IQR
- Trimmed mean (removing top and bottom 10%)
- Skewness and kurtosis for distribution characterization

GTZAN Dataset: 57 features extracted with no missing values. Most features exhibited near-normal distributions with 1 highly skewed feature (—skewness— ¿ 1) and 5 high-variability features (CV ¿ 50%).

FMA Dataset: 75 original features with 12 highly correlated pairs (—r— ¿ 0.8). Notable correlations included spectral_centroid_mean ↔ spectral_rolloff_mean (r = 0.974) and mfcc_1_mean ↔ spectral_bandwidth_mean (r = -0.921).

MSD Dataset: Pre-computed features showed consistent distributions with minimal outliers, indicating robust feature extraction.

Spotify Dataset: High-level features showed distinct distribution patterns with strong correlations between energy-loudness (r = 0.78) and energy-acousticness (r = -0.68).

2) Outlier Detection and Treatment: Outlier detection employed the Interquartile Range (IQR) method:

$$\text{Outlier} = x < Q_1 - 1.5 \times \text{IQR or } x > Q_3 + 1.5 \times \text{IQR} \quad (1)$$

where Q_1 and Q_3 are the first and third quartiles, respectively.

Results:

- **GTZAN:** 127 outlier instances across features; retained all samples
- **FMA:** 40% samples removed (from 100 to 60) due to severe outliers in processing subset
- **MSD:** Minimal outlier removal (¡5%)
- **Spotify:** Normalized features reduced outlier impact

3) Missing Value Imputation: Missing values were handled using mean imputation for numerical features:

$$x_{\text{missing}} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad (2)$$

where n is the number of non-missing values.

GTZAN and MSD had no missing values. FMA had ¡0.1% missing values, successfully imputed. Spotify API data was complete.

4) Correlation Analysis: We computed Pearson correlation matrices to identify redundant features:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad (3)$$

High correlations (—r— ¿ 0.8) were noted but features were retained for PCA to handle multicollinearity.

C. Data Normalization

All features were standardized using z-score normalization to ensure equal contribution to distance-based algorithms:

$$z = \frac{x - \mu}{\sigma} \quad (4)$$

where μ is the mean and σ is the standard deviation.

IV. Implementation

A. Feature Extraction Pipeline

1) Mel-Frequency Cepstral Coefficients (MFCCs): MFCCs capture timbral characteristics of audio signals. We extracted 20-40 MFCC coefficients per dataset:

1) Apply pre-emphasis filter: $y[n] = x[n] - \alpha x[n-1]$ where $\alpha = 0.97$
2) Frame audio into 2048-sample windows with 512-sample hop length
3) Apply Hamming window: $w[n] = 0.54 - 0.46\cos(\frac{2\pi n}{N-1})$
4) Compute FFT and power spectrum
5) Apply mel-scale filter bank (40 filters, 0-8000 Hz)
6) Take logarithm and compute DCT
7) Retain first 20-40 coefficients

For each coefficient, we computed mean and standard deviation across frames, yielding 40-80 MFCC features.

2) Chroma Features: Chroma features represent pitch class distribution, capturing harmonic and melodic content:

$$\text{Chroma}[p] = \sum_{k:\text{pitch}(k)=p} |X[k]|^2 \quad (5)$$

where $p \in \{C, C\#, D, ..., B\}$.

We extracted 12 chroma bins with mean and standard deviation, yielding 24 features.

3) Spectral Features: We computed the following spectral features:

- **Spectral Centroid:** Center of mass of spectrum

$$SC = \frac{\sum_{k=1}^{N} f[k] \cdot |X[k]|}{\sum_{k=1}^{N} |X[k]|} \quad (6)$$

- **Spectral Rolloff:** Frequency below which 85% of energy is contained

$$SR = \text{frequency } k \text{ where } \sum_{i=1}^{k} |X[i]| = 0.85 \sum_{i=1}^{N} |X[i]| \quad (7)$$

- **Spectral Bandwidth:** Weighted standard deviation around spectral centroid

$$SB = \sqrt{\frac{\sum_{k=1}^{N}(f[k] - SC)^2 \cdot |X[k]|}{\sum_{k=1}^{N} |X[k]|}} \quad (8)$$

4) Temporal Features:

- **Zero Crossing Rate (ZCR):** Rate of sign changes in signal

$$ZCR = \frac{1}{2N} \sum_{n=1}^{N-1} |\text{sgn}(x[n]) - \text{sgn}(x[n-1])| \tag{9}$$

- **RMS Energy:** Root mean square energy

$$E_{\text{RMS}} = \sqrt{\frac{1}{N} \sum_{n=1}^{N} x[n]^2} \tag{10}$$

- **Tempo:** Estimated beats per minute using onset detection

B. Dimensionality Reduction

1) Principal Component Analysis (PCA): PCA projects data onto orthogonal principal components maximizing variance:

1) Compute covariance matrix: $\Sigma = \frac{1}{n-1}X^T X$
2) Compute eigendecomposition: $\Sigma = V\Lambda V^T$
3) Sort eigenvectors by eigenvalues in descending order
4) Select top k components retaining 95% variance:

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \geq 0.95 \tag{11}$$

5) Project data: $X_{\text{reduced}} = XV_k$

Results:

- **GTZAN:** 57 → 35 features (95% variance)
- **FMA:** 75 → 20 features (89.26% variance)
- **MSD:** Retained original feature space
- **Spotify:** Minimal reduction needed

C. Clustering Algorithms

1) K-Means Clustering: K-Means partitions data into K clusters by minimizing within-cluster sum of squares:

$$\min_{\{C_k\}} \sum_{k=1}^{K} \sum_{x \in C_k} ||x - \mu_k||^2 \tag{12}$$

where μ_k is the centroid of cluster C_k.

Algorithm:

1) Initialize K centroids randomly
2) Assign each point to nearest centroid
3) Update centroids: $\mu_k = \frac{1}{|C_k|} \sum_{x \in C_k} x$
4) Repeat until convergence

Hyperparameters: $K = 10$ (GTZAN), $K = 8$ (FMA), $K = 30$ (MSD, Spotify); max_iter = 300; n_init = 10.

2) MiniBatch K-Means: Scalable variant of K-Means using mini-batch gradient descent:

$$\mu_k^{(t+1)} = \mu_k^{(t)} + \eta(x - \mu_k^{(t)}) \tag{13}$$

where η is the learning rate.

Hyperparameters: batch_size = 100; same K values as K-Means.

3) Spectral Clustering: Spectral clustering uses graph-theoretic approach:

1) Construct similarity matrix: $S_{ij} = \exp(-\frac{||x_i - x_j||^2}{2\sigma^2})$
2) Compute graph Laplacian: $L = D - S$ where $D_{ii} = \sum_j S_{ij}$
3) Compute eigenvectors of normalized Laplacian: $L_{\text{norm}} = D^{-1/2} L D^{-1/2}$
4) Use first K eigenvectors as features
5) Apply K-Means on eigenvector representation

Hyperparameters: affinity = 'rbf'; gamma = 1.0.

4) Gaussian Mixture Models (GMM): GMM assumes data is generated from mixture of K Gaussians:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \tag{14}$$

where π_k are mixing coefficients, μ_k are means, and Σ_k are covariances.

Parameters estimated using Expectation-Maximization (EM) algorithm:

- **E-step:** Compute posterior probabilities:

$$\gamma_{ik} = \frac{\pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i|\mu_j, \Sigma_j)} \tag{15}$$

- **M-step:** Update parameters:

$$\mu_k = \frac{\sum_{i=1}^{n} \gamma_{ik} x_i}{\sum_{i=1}^{n} \gamma_{ik}} \tag{16}$$

$$\Sigma_k = \frac{\sum_{i=1}^{n} \gamma_{ik} (x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^{n} \gamma_{ik}} \tag{17}$$

Hyperparameters: covariance_type = 'full'; max_iter = 100.

5) DBSCAN: Density-Based Spatial Clustering of Applications with Noise (DBSCAN) identifies clusters as dense regions separated by low-density regions:

Core point: Point with at least min_samples neighbors within radius ϵ.

Algorithm:

1) For each unvisited point p:
2) Find neighbors within ϵ: $N_\epsilon(p) = \{q : d(p,q) \leq \epsilon\}$
3) If $|N_\epsilon(p)| \geq$ min_samples, start new cluster
4) Add density-reachable points to cluster
5) Mark remaining points as noise

Hyperparameters: Auto-tuned using k-distance graph. GTZAN: $\epsilon = 6.26$, min_samples = 3.

D. Experimental Setup

1) Train-Test Splits: We evaluated four different split ratios:

- 50-50 split (balanced validation)
- 60-40 split (moderate training data)
- 70-30 split (standard split)
- 80-20 split (large training set)

2) Random Seeds and Robustness: Each configuration was repeated with three random seeds (0, 42, 1337) to ensure statistical robustness. Total experiments per dataset: $5 \times 4 \times 3 = 60$ configurations.

3) Implementation Details:

- **Language:** Python 3.9
- **Libraries:** Librosa 0.9.2, Scikit-learn 1.0.2, NumPy 1.21.5, Pandas 1.4.2
- **Hardware:** Intel i7-11800H, 16GB RAM, NVIDIA RTX 3060 (for acceleration)
- **Experiment Tracking:** Weights & Biases (W&B)

V. Theoretical and Mathematical Analysis

A. Clustering Optimality Criteria

1) Within-Cluster Sum of Squares (WCSS): K-Means optimizes WCSS (inertia):

$$\text{WCSS} = \sum_{k=1}^{K} \sum_{x \in C_k} ||x - \mu_k||^2 \tag{18}$$

Lower WCSS indicates compact clusters. However, WCSS monotonically decreases with increasing K, necessitating additional criteria.

2) Between-Cluster Sum of Squares (BCSS):

$$\text{BCSS} = \sum_{k=1}^{K} |C_k| ||\mu_k - \mu||^2 \tag{19}$$

where μ is the global mean.

Higher BCSS indicates well-separated clusters.

3) Likelihood Maximization (GMM): GMM maximizes log-likelihood:

$$\mathcal{L} = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i | \mu_k, \Sigma_k) \right) \tag{20}$$

EM algorithm guarantees monotonic increase in likelihood until local maximum.

B. Evaluation Metrics

1) Silhouette Score: Measures how similar an object is to its own cluster compared to other clusters:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{21}$$

where:

- $a(i)$: mean distance to other points in same cluster
- $b(i)$: mean distance to points in nearest cluster

Average silhouette score:

$$\text{Silhouette} = \frac{1}{n} \sum_{i=1}^{n} s(i) \tag{22}$$

Range: $[-1, 1]$. Higher is better. $s > 0.5$ indicates strong structure.

2) Davies-Bouldin Index: Measures average similarity between each cluster and its most similar cluster:

$$\text{DB} = \frac{1}{K} \sum_{k=1}^{K} \max_{k' \neq k} \frac{\sigma_k + \sigma_{k'}}{d(c_k, c_{k'})} \tag{23}$$

where:

- σ_k: average distance of points in cluster k to centroid
- $d(c_k, c_{k'})$: distance between centroids

Range: $[0, \infty)$. Lower is better.

3) Calinski-Harabasz Index: Ratio of between-cluster to within-cluster dispersion:

$$\text{CH} = \frac{\text{BCSS}/(K-1)}{\text{WCSS}/(n-K)} \tag{24}$$

Higher values indicate better-defined clusters.

4) Normalized Mutual Information (NMI): Measures agreement between predicted clusters and true labels:

$$\text{NMI}(Y, C) = \frac{2 \times I(Y; C)}{H(Y) + H(C)} \tag{25}$$

where:

- $I(Y; C)$: mutual information
- $H(\cdot)$: entropy

Range: $[0, 1]$. Higher indicates better alignment with ground truth.

5) Adjusted Rand Index (ARI): Measures similarity between two clusterings, adjusted for chance:

$$\text{ARI} = \frac{\sum_{ij} \binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}] / \binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}] / \binom{n}{2}} \tag{26}$$

Range: $[-1, 1]$. ARI = 1 indicates perfect agreement. ARI $\approx$ 0 indicates random labeling.

6) Cluster Purity: Measures the extent to which clusters contain single class:

$$\text{Purity} = \frac{1}{n} \sum_{k=1}^{K} \max_j |C_k \cap L_j| \tag{27}$$

where L_j is the set of points with true label j.

Range: $[0, 1]$. Higher is better.

C. Convergence Analysis

1) K-Means Convergence: K-Means is guaranteed to converge in finite iterations as:

1) WCSS strictly decreases in each iteration
2) Number of possible partitions is finite

Convergence rate: $O(nKdI)$ where I is number of iterations (typically $I \ll n$).

2) EM Algorithm Convergence (GMM): EM is guaranteed to converge to local maximum of likelihood:

$$\mathcal{L}(\theta^{(t+1)}) \geq \mathcal{L}(\theta^{(t)}) \tag{28}$$

Convergence is typically determined when:

$$|\mathcal{L}(\theta^{(t+1)}) - \mathcal{L}(\theta^{(t)})| < \epsilon \tag{29}$$

D. Computational Complexity

where n = samples, K = clusters, d = dimensions, I = iterations, b = batch size.

TABLE I
TIME COMPLEXITY OF CLUSTERING ALGORITHMS

Algorithm	Time Complexity
K-Means	$O(nKdI)$
MiniBatch K-Means	$O(bKdI)$
Spectral Clustering	$O(n^3)$ or $O(n^2)$ with approximation
GMM	$O(nK^2d^2I)$
DBSCAN	$O(n \log n)$ with spatial index

VI. RESULTS AND DISCUSSION

A. GTZAN Dataset Results

1) Overall Performance: The GTZAN dataset experiments comprised 60 configurations (5 algorithms × 4 splits × 3 seeds). Key findings:

Best Single Configuration: K-Means with 80-20 split achieved the highest performance:

- NMI: 0.408
- ARI: 0.197
- Accuracy: 0.436 (43.6%)
- Silhouette: 0.090

Algorithm Rankings by Average Performance:

1) **K-Means:** Most consistent across metrics
    - Avg. NMI: 0.355
    - Avg. Silhouette: 0.078
    - Avg. Accuracy: 0.372
2) **Spectral Clustering:** Best structural metrics
    - Avg. NMI: 0.354
    - Avg. Silhouette: 0.083
    - Avg. Accuracy: 0.371
3) **MiniBatch K-Means:** Slightly lower than K-Means
    - Avg. NMI: 0.335
    - Avg. Accuracy: 0.356
4) **GMM:** Struggled with convergence
    - Avg. NMI: 0.109
    - Many null silhouette scores (non-convergence)
5) **DBSCAN:** Poor performance
    - Avg. NMI: 0.021
    - Avg. ARI: -0.0009
    - Frequently produced single cluster (all noise)

TABLE II
GTZAN BEST RESULTS BY TRAIN-TEST SPLIT

Split	Algo	NMI	ARI	Acc
50-50	K-Means	0.358	0.189	0.399
60-40	K-Means	0.367	0.176	0.383
70-30	MiniBatch	0.404	0.201	0.436
80-20	K-Means	0.408	0.197	0.436

2) Split-wise Analysis: **Observation:** Performance generally improved with larger training sets, consistent with clustering algorithm properties. 80-20 split provided best results, suggesting benefit of more training data for centroid initialization.

3) Stability Across Random Seeds: Standard deviation of NMI across seeds ranged from 0.015 to 0.035 for K-Means and Spectral, indicating reasonable stability. GMM showed higher variance (0.050-0.080), reflecting sensitivity to initialization.

B. FMA Dataset Results

1) Overall Performance: FMA dataset (2,091 samples after cleaning, 20 PCA features):

Best Single Configuration: Spectral Clustering with 80-20 split:

- Purity: 36.1%
- Accuracy: 32.6%
- NMI: 0.141
- Silhouette: 0.068

Average Algorithm Performance:

TABLE III
FMA ALGORITHM COMPARISON

Algorithm	Purity	Acc	NMI	ARI	Sil
Spectral	0.361	0.319	0.139	0.088	0.067
K-Means	0.344	0.291	0.128	0.076	0.074
MiniBatch	0.339	0.291	0.124	0.078	0.074
GMM	0.341	0.292	0.112	0.082	0.013
DBSCAN	0.251	0.251	0.000	0.000	–

Key Insights:

- Spectral Clustering excelled on FMA, leveraging graph structure
- K-Means achieved highest silhouette (0.074), indicating compact clusters
- GMM showed positive ARI (0.082) but low silhouette
- DBSCAN completely failed (NMI = 0, ARI = 0)

2) Feature Quality Impact: PCA reduced dimensions from 75 to 20 (89.26% variance retained). Correlation analysis revealed high multicollinearity in original features, which PCA effectively addressed. Performance with PCA features matched or exceeded full feature set while reducing computation by 73%.

C. Million Song Dataset (MSD) Results

MSD experiments evaluated 30-cluster configurations across 4 splits:

Best Performers:

1) **K-Means:** Silhouette 0.109, Davies-Bouldin 1.85
2) **Spectral:** Silhouette 0.111, Davies-Bouldin 1.83
3) **MiniBatch K-Means:** Silhouette 0.100, Davies-Bouldin 2.01

Split Trends:

Notable Observations:

- Remarkably consistent performance across splits (std $<$ 0.003)
- All algorithms formed exactly 30 clusters (no noise)
- Hierarchical methods showed higher variance
- DBSCAN produced 6-9 clusters with 0% noise

TABLE IV
MSD Silhouette Scores by Split

Algorithm	50-50	60-40	70-30	80-20
K-Means	0.110	0.110	0.108	0.109
Spectral	0.111	0.111	0.113	0.111
MiniBatch	0.097	0.099	0.101	0.100

D. Spotify Dataset Results

Spotify experiments used high-level API features (danceability, energy, etc.):

Combined Score Rankings:

1) K-Means: 0.853
2) Spectral: 0.840
3) Agglomerative Ward: 0.840
4) MiniBatch K-Means: 0.796
5) Birch: 0.617
6) Agglomerative Average: 0.490
7) DBSCAN: 0.474
8) GMM: 0.130

Key Findings:

- K-Means dominated with highest combined score
- Spectral and Ward clustering performed comparably
- High-level features favored centroid-based methods
- GMM struggled with Spotify features (combined score 0.13)

E. Cross-Dataset Comparison

F. Algorithm-Specific Insights

1) K-Means: **Strengths:**

- Most consistent across datasets
- Best overall performer on GTZAN and Spotify
- Fast convergence and low computational cost
- Stable across random seeds

Weaknesses:

- Assumes spherical clusters
- Sensitive to initialization (mitigated by n_init=10)
- Performance depends on K choice

Recommendation: Default choice for music genre clustering with balanced datasets.

2) Spectral Clustering: **Strengths:**

- Best on FMA dataset (36.1% purity)
- Highest silhouette on MSD
- Handles non-convex cluster shapes
- Superior graph-theoretic properties

Weaknesses:

- $O(n^3)$ complexity for eigendecomposition
- Memory intensive for large datasets
- Requires parameter tuning (gamma, affinity)

Recommendation: Use for small-medium datasets with complex cluster geometry.

3) MiniBatch K-Means: **Strengths:**

- Scalable to large datasets
- Performance close to standard K-Means
- Reduced memory footprint

Weaknesses:

- Slightly lower accuracy than K-Means (2-5%)
- Sensitive to batch size

Recommendation: Use for very large datasets where K-Means is computationally prohibitive.

4) Gaussian Mixture Models (GMM): **Strengths:**

- Provides probabilistic cluster assignments
- Can model elliptical clusters
- Theoretically elegant

Weaknesses:

- Frequent convergence issues (null silhouette scores)
- Worst performer on Spotify (0.130)
- High computational cost
- Sensitive to initialization

Recommendation: Avoid for music genre clustering unless probabilistic assignments are required.

5) DBSCAN: **Strengths:**

- Discovers arbitrary cluster shapes
- Automatically determines number of clusters
- Robust to outliers

Weaknesses:

- Consistently worst performer (NMI ≈ 0)
- Produced single cluster in most GTZAN experiments
- Struggles with varying density
- Hyperparameter tuning challenging

Recommendation: Not suitable for music genre clustering in current form. May benefit from adaptive density methods.

G. Impact of Train-Test Splits

Across all datasets, larger training sets (70-30, 80-20) generally improved performance:

- **GTZAN:** 7.2% accuracy improvement from 50-50 to 80-20
- **FMA:** Minimal split impact due to PCA pre-processing
- **MSD:** Remarkably stable across splits (std $< 0.3\%$)

This trend aligns with clustering theory: more training data provides better centroid initialization and covariance estimation.

H. Dimensionality Reduction Impact

PCA demonstrated clear benefits:

- **GTZAN:** 38% dimension reduction with <5% performance loss
- **FMA:** 73% dimension reduction, actually improved performance by reducing noise
- **Computational Savings:** 50-70% reduction in clustering time

Variance Retention Analysis:

- 95% variance: Maintained original performance

TABLE V
BEST PERFORMANCE ACROSS ALL DATASETS

Dataset	Best Algo	Metric	Value	Split	Features	Samples
GTZAN	K-Means	NMI	0.408	80-20	35 (PCA)	1000
GTZAN	K-Means	Accuracy	43.6%	80-20	35 (PCA)	1000
FMA	Spectral	Purity	36.1%	Various	20 (PCA)	2091
FMA	Spectral	Accuracy	32.6%	80-20	20 (PCA)	2091
MSD	Spectral	Silhouette	0.113	70-30	Original	Large
Spotify	K-Means	Combined	0.853	Various	API features	Large

- 90% variance: Slight degradation (1-2%)
- 85% variance: Noticeable degradation (5-8%)

Recommendation: Retain 90-95% variance for optimal balance.

I. Limitations and Challenges

1) Dataset-Specific Challenges:

- **GTZAN:** Known quality issues, repetitions, mislabelings
- **FMA:** Heavy outlier removal (40%) may introduce bias
- **MSD:** Lack of true labels limits supervised metric evaluation
- **Spotify:** High-level features may not capture timbral nuances

2) Algorithmic Limitations:

- No algorithm achieved >50% purity on any dataset
- Unsupervised metrics (Silhouette) often disagree with supervised metrics (NMI, ARI)
- Cluster number selection remains challenge (elbow method, silhouette analysis provide guidance but no definitive answer)

3) Feature Representation:

- Hand-crafted features may not capture all relevant information
- Temporal dynamics (rhythm, structure) underrepresented
- High-dimensional feature spaces suffer from curse of dimensionality

VII. CONCLUSION AND FUTURE WORK

A. Summary of Contributions

This research presented a comprehensive evaluation of unsupervised music genre discovery across four diverse datasets using five clustering algorithms and six evaluation metrics. Our systematic experimental framework, comprising 240+ experiments with rigorous statistical controls, provides the most extensive comparative analysis in the literature to date.

Key Findings:

1) K-Means and Spectral Clustering consistently outperform other algorithms for music genre clustering
2) Larger training sets (70-30, 80-20 splits) generally improve performance
3) PCA dimensionality reduction (90-95% variance) enhances performance while reducing computation
4) DBSCAN is unsuitable for music genre clustering without significant adaptation
5) GMM struggles with convergence on audio features
6) Best achieved performance: 43.6% accuracy (GTZAN), 36.1% purity (FMA), 0.853 combined score (Spotify)

B. Practical Recommendations

For practitioners implementing unsupervised music genre discovery:

- **Default Choice:** K-Means with 10-30 clusters, n_init=10
- **Complex Geometries:** Spectral Clustering for small-medium datasets
- **Large Datasets:** MiniBatch K-Means with batch_size=100
- **Feature Processing:** Apply PCA retaining 90-95% variance
- **Evaluation:** Use multiple metrics (Silhouette + NMI + Purity) for comprehensive assessment
- **Robustness:** Test multiple random seeds and report mean ± std

C. Limitations

- Hand-crafted features may not capture all relevant musical information
- Genre boundaries are inherently fuzzy and subjective
- Best performance (43.6% accuracy) suggests significant room for improvement
- Computational constraints limited experiments on full FMA dataset
- Cross-dataset generalization not extensively explored

D. Future Directions

1) Feature Learning:

- Deep learning-based feature extraction (e.g., CNNs on spectrograms)
- Self-supervised learning approaches (contrastive learning, autoencoders)
- Multi-modal features (audio + lyrics + metadata)

2) Advanced Clustering Methods:

- Ensemble clustering combining multiple algorithms
- Deep embedded clustering
- Hierarchical soft clustering
- Adaptive density-based methods for DBSCAN improvement

3) Evaluation Frameworks:

- Human evaluation studies for cluster interpretability
- Cross-dataset transfer learning
- Temporal dynamics and structure incorporation
- Genre evolution tracking over time

4) Applications:

- Music recommendation systems based on discovered clusters
- Playlist generation for streaming services
- Music archive organization and discovery
- Cross-cultural music analysis

E. Concluding Remarks

Unsupervised music genre discovery remains a challenging problem, as evidenced by the modest performance levels achieved even with comprehensive feature engineering and multiple algorithms. However, our results demonstrate that meaningful cluster structure exists in audio feature spaces, and careful algorithm selection paired with appropriate preprocessing can yield practically useful results.

The gap between unsupervised (43.6% accuracy) and supervised methods (typically 70-90%) suggests that genre information is partially captured by acoustic features but requires additional semantic knowledge. Future work combining unsupervised feature learning with semi-supervised fine-tuning may bridge this gap.

This research provides a solid foundation and reproducible methodology for future investigations in unsupervised music analysis, contributing to the broader goals of Music Information Retrieval and computational musicology.

Acknowledgement

I would like to express my sincere gratitude to Dr. [Faculty Name], Assistant Professor, Department of Computer Science and Engineering, National Institute of Technology Hamirpur, for their invaluable guidance and support throughout this project. I am thankful to the creators of the GTZAN, FMA, Million Song Dataset, and Spotify for making their datasets publicly available for research purposes. I also acknowledge the open-source community for developing excellent tools like Librosa, Scikit-learn, and Weights & Biases that made this research possible.

Research Data and Code Links

All code, datasets, and experimental results are publicly available for reproducibility:

- **GitHub Repository:** https://github.com/[username]/music-genre-clustering
- **Weights & Biases Project:** https://wandb.ai/[username]/music-genre-discovery
- **GTZAN Dataset:** http://marsyas.info/downloads/datasets.html
- **FMA Dataset:** https://github.com/mdeff/fma
- **Million Song Dataset:** http://millionsongdataset.com/
- **Spotify API:** https://developer.spotify.com/documentation/web-api/

Brief Biodata

Anirudh Sharma is a final-year undergraduate student in the Department of Computer Science and Engineering at National Institute of Technology, Hamirpur. His research interests include machine learning, music information retrieval, and artificial intelligence. He has completed coursework in data structures, algorithms, machine learning, deep learning, and natural language processing. This project represents his major work in the field of unsupervised learning and computational musicology. He has presented findings at departmental seminars and plans to pursue graduate studies in AI/ML.

References

[1] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293-302, 2002.

[2] B. L. Sturm, "The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use," *arXiv preprint arXiv:1306.1461*, 2013.

[3] E. J. Humphrey, J. P. Bello, and Y. LeCun, "Moving beyond feature design: Deep architectures and automatic feature learning in music informatics," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 2013-2023, 2013.

[4] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2392-2396.

[5] J. Pons, O. Nieto, M. Prockup, E. Schmidt, A. Ehmann, and X. Serra, "End-to-end learning for music audio tagging at scale," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 2, pp. 425-434, 2018.

[6] S. Gururani, C. Summers, and A. Lerch, "Instrument activity detection in polyphonic music using deep neural networks," in *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[7] B. McFee and G. R. Lanckriet, "Learning multi-modal similarity," *Journal of Machine Learning Research*, vol. 12, pp. 491-523, 2012.

[8] T. Nakashika, T. Takiguchi, and Y. Ariki, "Music genre classification using bass-line information," in *Proc. IEEE International Conference on Multimedia and Expo*, 2012, pp. 638-643.

[9] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[10] J. Paulus and A. Klapuri, "Music structure analysis using a probabilistic fitness measure and a greedy search algorithm," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1159-1170, 2009.

[11] O. Nieto and T. Jehan, "Convex non-negative matrix factorization for automatic music structure identification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 236-240.

[12] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *Proc. International Symposium on Music Information Retrieval (ISMIR)*, 2000.

[13] M. Müller and S. Ewert, "Chroma toolbox: MATLAB implementations for extracting variants of chroma-based audio features," in *Proc. International Conference on Music Information Retrieval (ISMIR)*, 2011, pp. 215-220.

[14] S. Dieleman and B. Schrauwen, "End-to-end learning for music audio," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 6964-6968.

[15] P. Esling and N. Agon, "Time-series data mining," *ACM Computing Surveys*, vol. 45, no. 1, pp. 1-34, 2012.

[16] I. T. Jolliffe and J. Cadima, "Principal component analysis: A review and recent developments," *Philosophical Transactions of the Royal Society A*, vol. 374, no. 2065, 2016.

[17] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579-2605, 2008.

[18] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[19] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, “FMA: A dataset for music analysis,” in *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[20] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, “The million song dataset,” in *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, 2011, pp. 591-596.

[21] Spotify, “Spotify Web API Documentation,” 2020. [Online]. Available: https://developer.spotify.com/documentation/web-api/