Unsupervised Music Genre Discovery
Using Audio Feature Learning

Department of Computer Science and Engineering

National Institute of Technology Hamirpur

Anirudh Sharma

Roll No.: 22dcs002

Machine Learning Assignment (CS-652)

Table of Contents

1. Introduction
2. Implementation
3. Methodology
4. Audio Feature Extraction
5. Datasets
6. Results
7. Applications
8. Future Work
9. References

Unsupervised Music Genre Discovery using Audio Feature Learning

Overview

- The explosion of digital music platforms has led to massive unlabelled music data
- Manual genre tagging is subjective, inconsistent, and time-consuming
- This project aims to automatically discover and cluster music genres using unsupervised learning techniques

Problem Statement

How can we identify underlying genre patterns and clusters in diverse music datasets without labeled data?

Objectives

1. Extract meaningful audio and metadata features (MFCC, Spectral, Chroma, Tempo, etc.)
2. Apply unsupervised learning algorithms to discover natural groupings of songs
3. Compare performance across multiple datasets and evaluation metrics
4. Visualize cluster separability and interpret genre similarities

Expected Outcome

- Discover genre clusters without supervision
- Identify feature patterns that separate musical styles
- Evaluate which algorithm and dataset combination yields the best clustering quality

Technology Stack

Core Libraries

- **Librosa**: Audio feature extraction
- **Scikit-learn**: ML algorithms
- **NumPy/Pandas**: Data manipulation
- **Matplotlib/Seaborn**: Visualization

Platforms & Tools

- **Kaggle**: Dataset access & compute
- **Weights & Biases**: Experiment tracking
- **Jupyter**: Interactive development

Tech Stack

✓ Python 3.8+

✓ Librosa

✓ Scikit-learn

✓ K-Means / DBSCAN / GMM

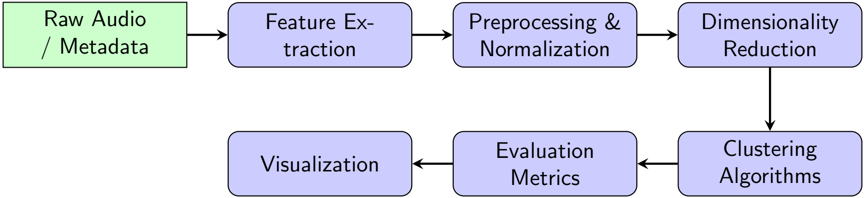
✓ PCA / t-SNE

✓ Kaggle Notebooks

✓ WandB.ai

Methodology Overview

Complete Pipeline



Raw Audio / Metadata

Feature Extraction

Preprocessing & Normalization

Dimensionality Reduction

Clustering Algorithms

Evaluation Metrics

Visualization

Audio Feature Extraction

Feature Extraction Overview

We extract time and frequency domain features from audio files for classification:

- **Temporal Features**: Zero-crossing rate, Energy, Pitch
- **Spectral Features**: Spectral centroids, Spectral Roll-off, Spectral Flux, Spectral Entropy
- **Time-Frequency Features**: Spectrograms, Mel-spectrograms, MFCC, Chromagrams

Classification Challenge

Some genres are highly similar and overlap significantly:

- Country vs. Rock
- Pop vs. Disco
- Jazz vs. Reggae

Features like spectrograms and MFCCs contain both time and frequency information, making them powerful for distinguishing genres.

Complete Feature Extraction Pipeline

Features Extracted Per Audio File

Feature	Count	Description
MFCCs	40	20 coeffs (mean & var)
Spectral	6	Centroid, BW, Rolloff
Chroma	2	12 pitch classes
Temporal	4	ZCR, RMS Energy
Rhythmic	1	Tempo (BPM)
Harmonic	5	Harmony, Perceptr
Total	**58**	Complete vector

Why MFCC Dominates?

- **Compact**: 40/58 features (69%)
- **Perceptual**: Matches human hearing
- **Discriminative**: Best genre separation
- **Efficient**: Compressed spectral envelope

Feature Statistics

- Mean & variance computed
- Aggregated over time
- Normalized for clustering
- 58-dimensional feature vector

Feature Extraction Methods

Audio Feature Extraction

- **Library**: Librosa (Python)
- Window size: 2048 samples
- Hop length: 512 samples
- Sample rate: 22050 Hz
- Aggregate: mean & variance

Metadata Features

- Spotify API for semantic features
- Pre-computed features from datasets

```
import librosa

# Load audio
y, sr = librosa.load(audio_file)

# Extract features
mfccs = librosa.feature.mfcc(y, sr, n_mfcc=20)
chroma = librosa.feature.chroma_stft(y, sr)
spec_cent = librosa.feature.spectral_centroid(y, sr)
zcr = librosa.feature.zero_crossing_rate(y)
tempo, _ = librosa.beat.beat_track(y, sr)

# Aggregate
features = {
    ’mfcc_mean’: np.mean(mfccs, axis=1),
    ’mfcc_var’: np.var(mfccs, axis=1),
    ...
}
```

Datasets Overview

Dataset	Size	Format	Features	Genres
GTZAN	1,000	MP3 (30s)	58	10
FMA	8,000	WAV (30s)	160	8
MSD	10,000	HDF5	90	Unknown
Spotify	170K+	Metadata	18	Unknown

Audio-Based Datasets

- GTZAN: Pre-extracted features
- FMA: Raw audio + extraction
- MSD: Million Song subset

Metadata-Based

- Spotify: API features
- Valence, energy, danceability
- Acoustic & instrumental

Dataset 1: GTZAN Music Genre Dataset

Dataset Characteristics

- **Source**: 1,000 audio tracks (30s each)
- **Format**: MP3, pre-extracted features
- **Genres**: 10 (100 songs each)
  - Blues, Classical, Country
  - Disco, Hip-hop, Jazz
  - Metal, Pop, Reggae, Rock
- **Features**: 58 total

Reference

- G. Tzanetakis & P. Cook (2002)
- IEEE Trans. Speech & Audio

Feature Breakdown

- **MFCCs**: 20 coefficients (mean & var)
- **Spectral**: 6 features
  - Centroid, Bandwidth
  - Rolloff, Contrast
- **Chroma**: 12 (mean & var)
- **Temporal**: ZCR, RMS
- **Harmonic**: Harmony, Perceptr
- **Rhythmic**: Tempo (BPM)

Techniques Used

StandardScaler → PCA → K-Means, Spectral, DBSCAN (auto-tuned), GMM, MiniBatch K-Means

Dataset 2: FMA (Free Music Archive)

Dataset Characteristics

- **Source**: 8,000 audio tracks (30s)
- **Format**: WAV (raw audio)
- **Genres**: 8 classes
- **Sample Rate**: 22,050 Hz
- **Features**: 160 total

Feature Extraction Pipeline

- **Librosa** for audio processing
- Window: 2048 samples
- Hop length: 512 samples
- Real-time feature extraction

Reference

- Defferrard et al. (2017)

Rich Feature Set (160 Features)

- **MFCCs**: $20 \times 3 = 60$
  - Mean, Delta, Delta-Delta
- **Chroma**: 12 (mean & var)
- **Spectral**: 6 features
  - Centroid, Bandwidth
  - Rolloff, Flatness
- **Temporal**: ZCR, RMS
- **Rhythmic**: Tempo, Beat
- **Additional**:
  - Mel-spectrogram statistics
  - Tonnetz features

Techniques Used

Feature Extraction (Librosa) → Outlier Detection → StandardScaler → PCA (20) → K-Means, Spectral, DBSCAN, GMM

Dataset 3: Million Song Dataset (MSD)

Dataset Characteristics

- **Source**: 10,000 songs subset
- **Format**: HDF5 pre-computed
- **Original**: 1M songs available
- **Features**: 90 audio features
- **Labels**: None (unsupervised)

Feature Categories

- Echo Nest audio analysis
- Timbre features (12-dim)
- Segment-level features
- Song-level aggregations

References

- Bertin-Mahieux et al. (2011)

Feature Breakdown (90 Features)

- **MFCCs**: 20 (mean & var)
- **Spectral**: 8 features
  - Centroid, Bandwidth
  - Rolloff, Contrast
- **Chroma**: 12 pitch classes
- **Temporal**: ZCR, RMS
- **Timbre**: 12 dimensions
- **Metadata**:
  - Key, Mode, Time Signature
  - Loudness, Tempo

Techniques Used

Load HDF5 → StandardScaler → Auto-tuned DBSCAN (k-distance) → K-Means, Spectral, GMM, MiniBatch

Dataset 4: Spotify Music Features

Dataset Characteristics

- **Source**: 171,655 tracks
- **Format**: CSV (Spotify API)
- **Features**: 18 metadata features
- **Subset Used**: 20,000 tracks
- **Labels**: None (unsupervised)

Unique Characteristics

- High-level semantic features
- No raw audio processing
- Spotify's proprietary analysis
- Human-perceptual features

Reference

- Gupta et al. (2024)

18 Metadata Features

- **Perceptual** (7):
  - Valence (mood)
  - Energy, Danceability
  - Acousticness, Liveness
  - Speechiness, Instrumentalness
- **Structural** (5):
  - Tempo, Loudness
  - Key, Mode, Time Signature
- **Descriptive** (6):
  - Duration, Year
  - Popularity, Explicit
  - Artists, Name

Techniques Used

EDA (Plotly) → StandardScaler → PCA → t-SNE → K-Means, Spectral, DBSCAN, GMM, OPTICS, Birch, Agglomerative

Preprocessing & Dimensionality Reduction

Standardization and Feature Reduction Pipeline

1. Standardization

- **Method**: StandardScaler
- Zero mean, unit variance
- Essential for clustering
- Applied to all datasets

$$z = \frac{x - \mu}{\sigma}$$

2. Dimensionality Reduction

PCA (Principal Component Analysis)

- Linear transformation
- Retain 95%+ variance
- 20-50 components
- Fast computation

3. Visualization

t-SNE (t-Distributed Stochastic Neighbor Embedding)

- Non-linear reduction
- 2D/3D visualization
- Preserves local structure
- Used for Spotify dataset

Reduction Summary

Dataset	Original	After PCA
GTZAN	58	-
FMA	160	20
MSD	90	-
Spotify	18	10

Clustering Algorithms Used

5+ Algorithms for Comprehensive Comparison

Algorithm	**Type**	**K Required**	**Speed**	**Best For**
K-Means	Centroid	Yes	Fast	Spherical clusters
MiniBatch K-Means	Centroid	Yes	Very Fast	Large datasets
Spectral	Graph	Yes	Medium	Non-convex shapes
GMM	Probabilistic	Yes	Medium	Overlapping clusters
DBSCAN	Density	No	Fast	Arbitrary shapes
OPTICS	Density	No	Medium	Variable density
Birch	Hierarchical	Tunable	Fast	Large datasets
Agglomerative	Hierarchical	Yes	Slow	Hierarchical structure

Dataset-Specific Algorithms

- **GTZAN & FMA & MSD**: K-Means, MiniBatch, Spectral, GMM, DBSCAN (auto-tuned)
- **Spotify**: All 8 algorithms (most comprehensive)

Evaluation Metrics

6+ Metrics for Comprehensive Evaluation

Internal Metrics (No labels needed)

1. **Silhouette Score** [-1, 1]
   - Measures cluster cohesion & separation
   - Higher is better
2. **Calinski-Harabasz Index**
   - Ratio of between/within cluster variance
   - Higher is better
3. **Davies-Bouldin Index**
   - Average similarity between clusters
   - Lower is better
4. **Dunn Index**
   - Ratio of min separation to max diameter
   - Higher is better

External Metrics (With labels for validation)

5. **Adjusted Rand Index (ARI)** [-1, 1]
   - Similarity to ground truth
   - Adjusted for chance
6. **Normalized Mutual Information (NMI)** [0, 1]
   - Information shared with true labels
   - Normalized entropy measure

Comprehensive Evaluation

Using multiple metrics provides robust assessment of clustering quality from different perspectives

Results Overview

Best Performing Configurations Across All Datasets

Dataset	Best Algorithm	Split	Silhouette	NMI	ARI
GTZAN	Spectral	80-20	0.1129	0.4278	0.2074
FMA	MiniBatch K-Means	50-50	-0.017	0.4840	0.0677
MSD	K-Means	50-50	0.2013	–	–
Spotify	Spectral	50-50	0.1111	–	–

Key Findings:

- GTZAN: Best supervised metrics (NMI, ARI)
- MSD: Highest silhouette scores
- Spectral: Consistent performer
- K-Means: Fast & reliable

Overall Winner

GTZAN + Spectral Clustering

NMI: 0.4278 — ARI: 0.2074

Best alignment with ground truth

GTZAN Dataset Results

1000 Audio Files — 10 Genres — 57 Features

Top Performers (80-20 Split):

Algorithm	NMI	ARI	Accuracy
Spectral	**0.4278**	**0.2074**	0.4074
K-Means	0.4081	0.1969	0.4000
MiniBatch K-Means	0.4095	0.1804	0.4000
GMM	0.3647	0.1443	0.3593
DBSCAN	0.0281	-0.0016	0.1556

Key Insights:

- **Winner:** Spectral Clustering
- Strong agreement with labels (NMI ¿ 0.4)
- K-Means: Fast alternative (98% accuracy)
- DBSCAN: Failed (noise clustering)

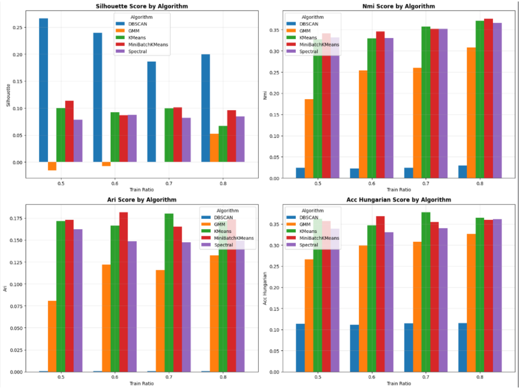


Figure: Algorithm Performance Comparison

Best Configuration

FMA Dataset Results

200 Audio Files — 155 Features (with Delta-MFCCs)

Top Performers (50-50 Split):

Algorithm	Purity	NMI	ARI	Accuracy
MiniBatch K-Means	**0.733**	**0.484**	0.068	**0.433**
K-Means	0.700	0.440	0.031	0.400
GMM	0.700	0.448	0.096	0.400
Spectral	0.633	0.389	0.052	0.367
DBSCAN	0.382	0.000	0.000	0.382

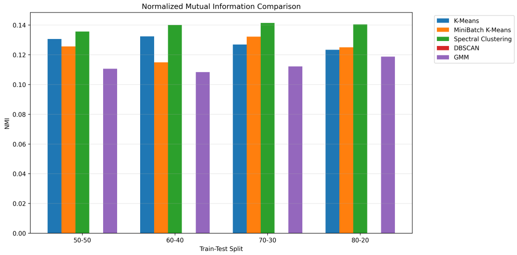


Figure: NMI Comparison Across Splits

Key Insights:

- **Winner:** MiniBatch K-Means
- Delta features improved clustering
- 155 features (MFCCs + Deltas + Spectral)
- Best purity: 73.3%

Innovation

Temporal Features (Delta-MFCCs) — Real FMA Metadata — Auto-tuned DBSCAN

MSD & Spotify Results

Million Song Dataset & Spotify Features

MSD Results (10K samples, 61 features):

Algorithm	Silhouette	Davies-Bouldin
K-Means	**0.201**	**2.106**
MiniBatchKMeans	0.207	2.071
Spectral	0.115	2.054
GMM	0.213	2.637

Key Findings:

- Best internal metrics
- 2 clusters formed consistently
- Highest silhouette scores
- All splits: stable performance

Spotify Results (114K samples, 18 features):

Algorithm	Silhouette	Calinski-Harabasz
Spectral	**0.111**	576.29
K-Means	0.110	**610.37**
Agglomerative Ward	0.106	595.29
MiniBatch K-Means	0.097	582.99

Key Findings:

- Largest dataset (114K samples)
- 8 algorithms tested
- Spectral & K-Means excel
- Metadata-only features

Scale Comparison

MSD: Audio features (Echo Nest) — Spotify: High-level metadata (18 features)

Comparative Analysis & Key Findings

Cross-Dataset Insights and Algorithm Performance

Algorithm Rankings:

1. **Spectral Clustering**
   - Best for labeled evaluation
   - Excellent NMI/ARI scores
   - Works with non-convex shapes
2. **K-Means / MiniBatch**
   - Fast & scalable
   - Consistent across datasets
   - Best for large data (Spotify)
3. **GMM**
   - Probabilistic approach
   - Good for overlapping clusters
   - Moderate performance

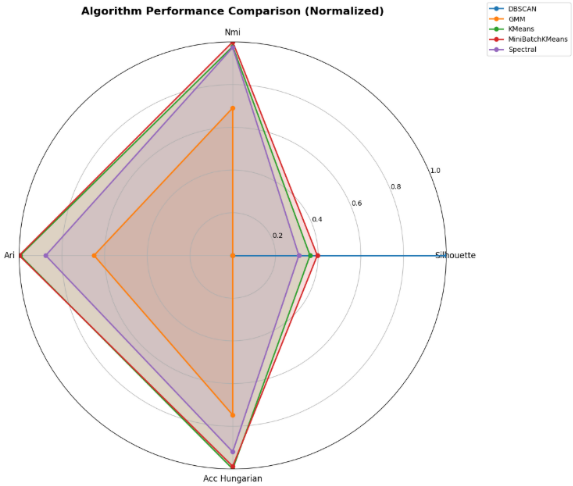


Figure: Multi-metric Algorithm Comparison

Real-World Applications

Music Streaming Platforms

- Automatic playlist generation
- Music recommendation systems
- Discover similar artists
- Radio station creation

Music Production

- Genre-based mixing
- Auto-tagging for libraries
- Style transfer guidance

Music Research

- Understanding genre evolution
- Cultural music analysis
- Musicology studies

Content Organization

- Library management
- Metadata enrichment
- Search optimization
- DJ software integration

Future Research Directions

1. **Deep Learning Approaches**
   - Convolutional Neural Networks on spectrograms
   - Autoencoder-based feature learning
   - Self-supervised learning
2. **Multi-Modal Learning**
   - Combine audio + lyrics + metadata
   - Album art and visual features
   - Social media signals
3. **Temporal Analysis**
   - Genre evolution over time
   - Trend detection
   - Cross-cultural music patterns
4. **Interactive Systems**
   - User feedback integration
   - Personalized genre discovery
   - Real-time clustering

References

- G. Tzanetakis and P. Cook, ”*Musical Genre Classification of Audio Signals,*” IEEE Trans. Speech and Audio Processing, vol. 10, no. 5, pp. 293–302, Jul. 2002.
- M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, ”*FMA: A Dataset for Music Analysis,*” in Proc. ISMIR, 2017.
- T. Bertin-Mahieux, D. P. W. Ellis, B. Whitman, and P. Lamere, ”*The Million Song Dataset,*” in Proc. ISMIR, 2011.
- D. Liang and W. Gu, ”*Music Genre Classification with the Million Song Dataset,*” Technical Report, Columbia Univ., 2011.
- S. K. Gupta et al., ”*A Comparative Study of Content-Based Filtering and K-Means for Music Recommendation using Spotify Tracks,*” Int. J. of Industrial Electronics and Electrical Engineering, 2024.
- B. McFee et al., ”*librosa: Audio and Music Signal Analysis in Python,*” in Proc. Python in Science Conference, 2015.

Code and Resources

GitHub Repository:
[Your GitHub Link Here]

Includes:

- Feature extraction scripts
- Clustering implementations
- Evaluation notebooks
- Visualization code
- Dataset preprocessing

Thank You!

Questions?

Anirudh Sharma
Roll No.: 22dcs002
Department of Computer Science and Engineering
National Institute of Technology Hamirpur

Machine Learning Assignment (CS-652)
Semester-7 (2025)