Technische Universität Braunschweig

Machine learning

Mini-project: Music/Audio Classification

Aman Jain

Master's Data Science

Introduction

- Audio classification is the task of automatically assigning labels to audio signals.
- Music genre classification is a common application of audio classification.
- Large volumes of digital music require automated analysis techniques.
- Machine learning enables learning patterns directly from audio features.
- Applications in **audio classification in music platforms:**

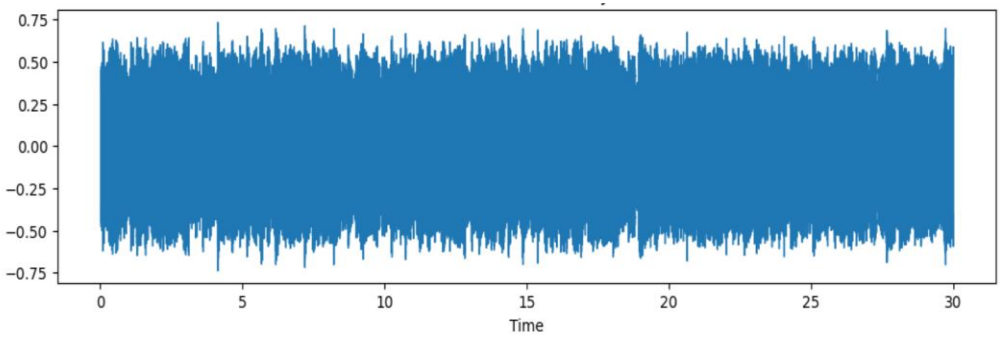
Music streaming apps | Automatic tagging | Playlist recommendation

- Applications in **Audio Content Moderation & Filtering**

Audio Visualization

How audio actually look like? Some common ways to visualize audio:

Waveform

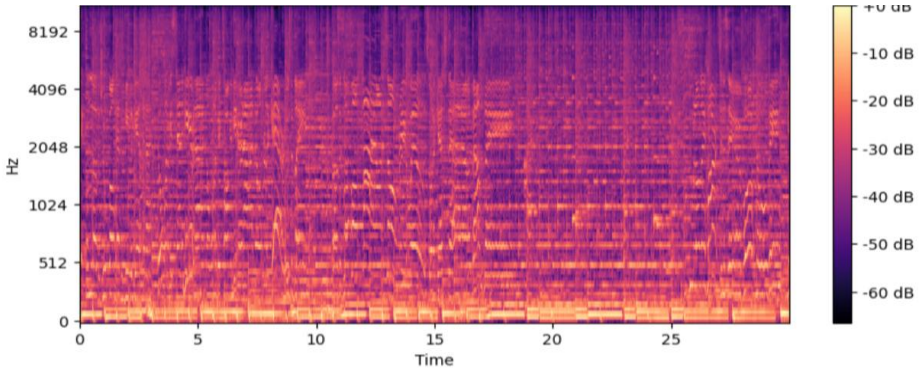


***Change in Amplitude** of audio signal over time*

- Loudness of the sound
- Silence vs active regions
- Peaks and valleys of the signal
- Duration of the audio

- Pitch and harmonic structure
- Timbre and texture
- Rhythmic patterns
- **Genre-specific frequency patterns**

Mel-Spectrogram

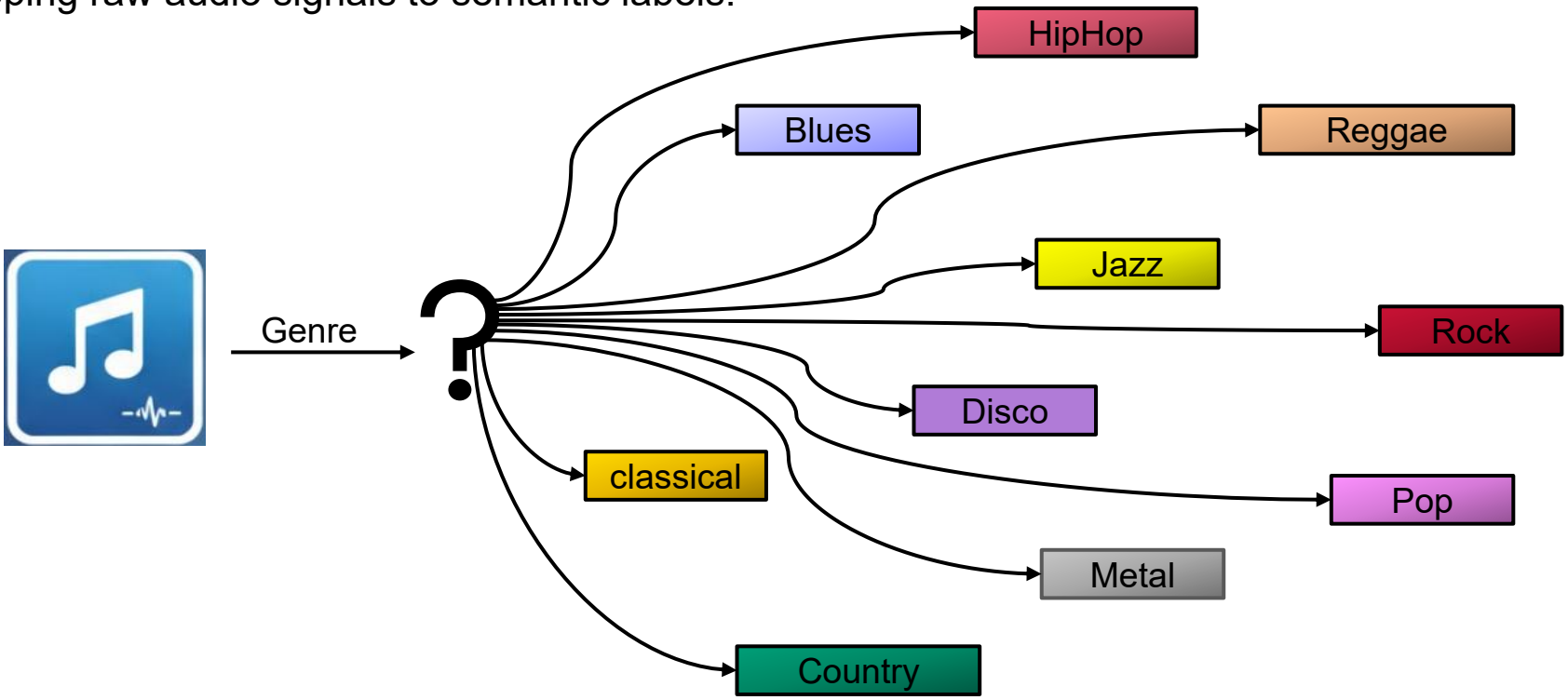


***Color Intensity:** Energy at that frequency*

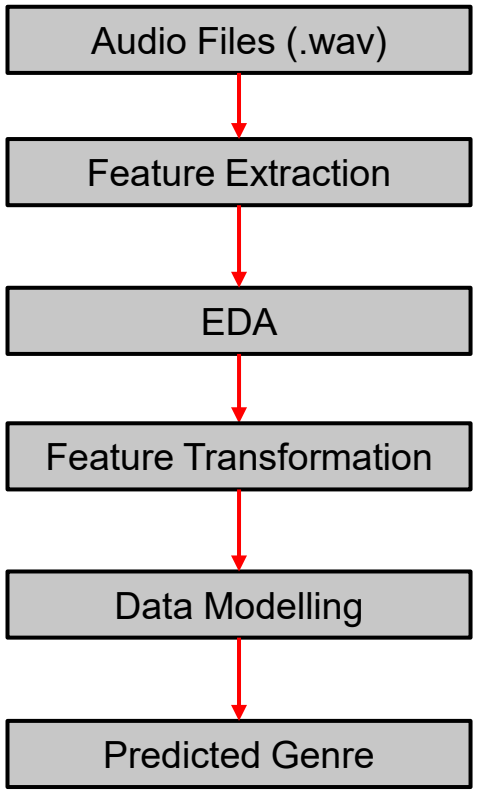
Motivation And Challenge

- Rapid growth of digital music and audio content.
- Manual music genre labeling is time-consuming and subjective.
- Need for automated and scalable audio analysis systems.
- Machine learning can learn discriminative patterns from audio signals.

Mapping raw audio signals to semantic labels!



Problem Statement / Workflow



Given an audio file in .wav format.

Extract meaningful acoustic features from the audio signal.

Exploratory Data Analysis on the features.

Feature Transformation: Standardization and PCA.

Train machine learning models using these features.

Predict the correct music genre among multiple classes.

Data – gtzan Data

- Music audio dataset organized by genre.
- Each genre stored in a separate folder and folder names used as class labels.

```
dataset/
├── blues/
├── classical/
├── country/
├── disco/
├── hiphop/
├── jazz/
├── metal/
├── pop/
├── reggae/
└── rock/
```

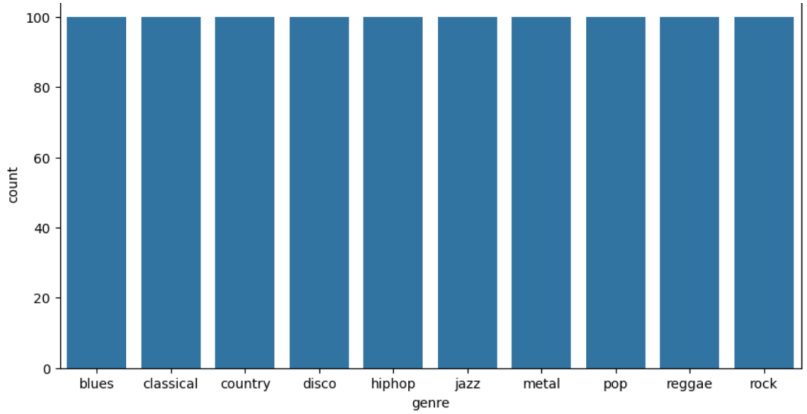
Genre	
blues	100
classical	100
country	100
disco	100
hiphop	100
jazz	100
metal	100
pop	100
reggae	100
rock	100

	Duration	SampleRate
count	999.000000	999.0
mean	30.024071	22050.0
std	0.080951	0.0
min	29.931973	22050.0
25%	30.000181	22050.0
50%	30.013333	22050.0
75%	30.013333	22050.0
max	30.648889	22050.0

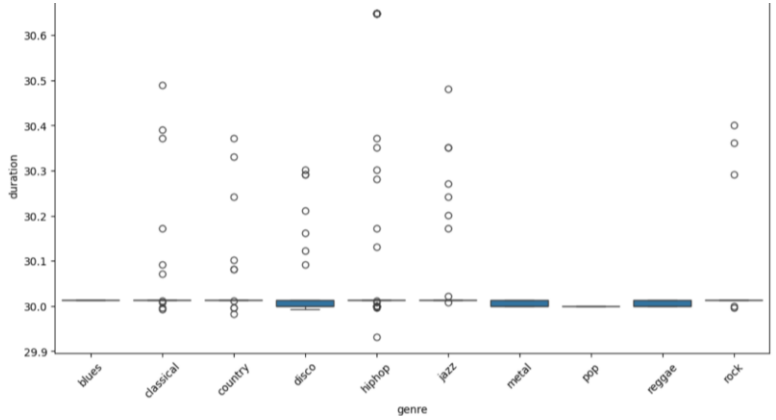
	Genre	Filename	Duration
0	blues	blues.00000.wav	30.013333
1	blues	blues.00001.wav	30.013333
2	blues	blues.00002.wav	30.013333
3	blues	blues.00003.wav	30.013333
4	blues	blues.00004.wav	30.013333

Audio files in .wav format

https://www.tensorflow.org/datasets/catalog/gtzan



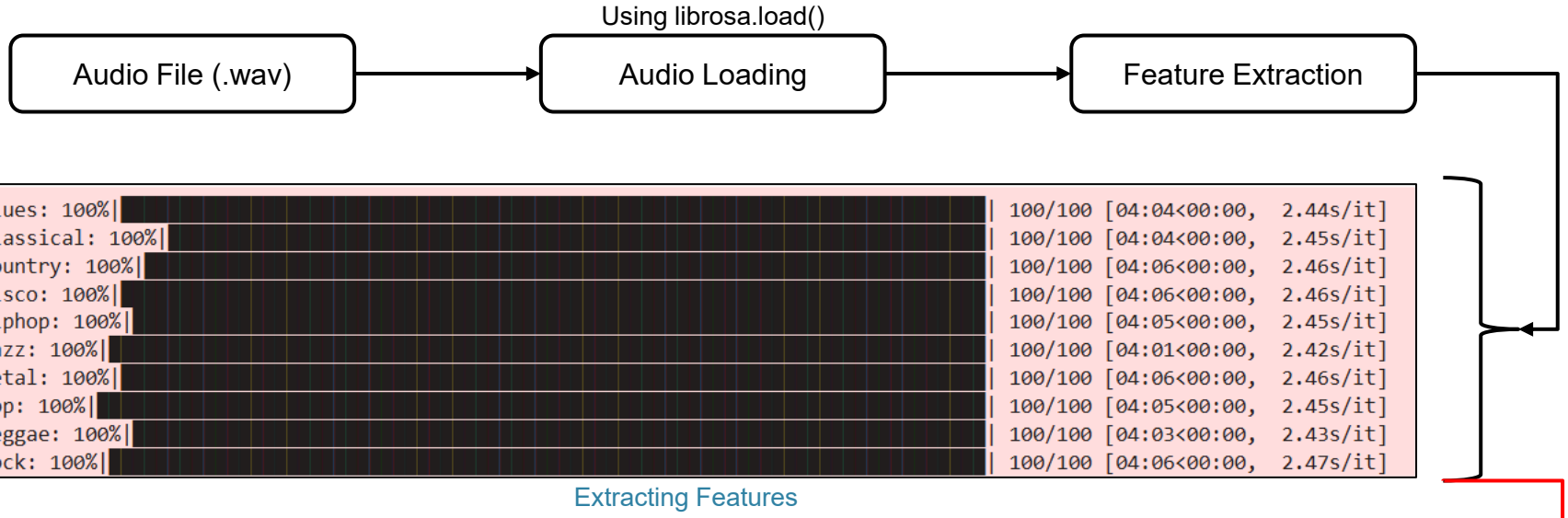
Number of files per genre



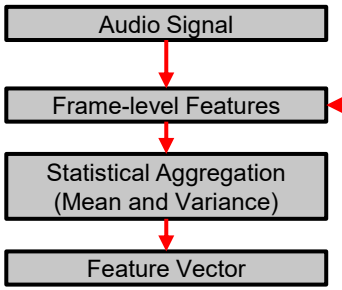
Duration Distribution per Genre

Feature Extraction

- Audio files loaded using Librosa Package.
- Audio passed directly to feature extraction stage.



- Raw audio signals are high-dimensional and not directly suitable for ML.
- Extracted features capture timbre, pitch, rhythm, and spectral properties.
- Feature extraction converts audio into compact numerical representations
- Each audio file is converted into a fixed-length feature vector.



https://librosa.org/doc/latest/index.html

Harmonic Feature - Chroma Features

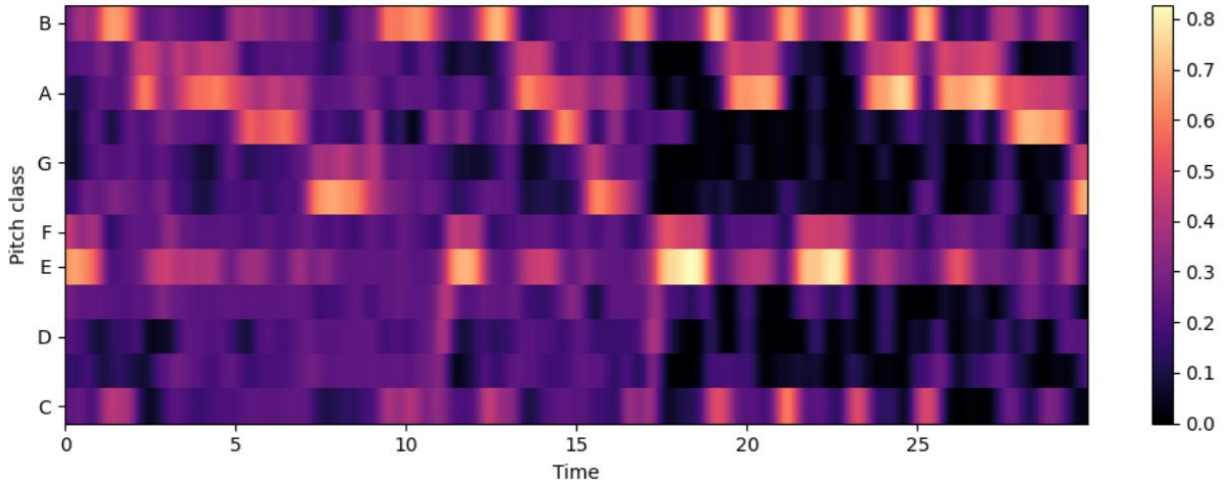
Chroma features represent the 12 pitch classes (C, C#, D, …, B), capturing harmonic content, making them invariant to octave and ideal for harmonic analysis.

$$\text{Chroma}_p = \sum_k |X(k)|^2 \; W_p(k)$$

(https://en.wikipedia.org/wiki/Chroma_feature)

Where:
- Chroma_p is the energy for pitch class p.
- $W_p(k)$ is a weighting function, often distributing energy across octaves

Chroma features are derived from the Short-Time Fourier Transform (STFT) magnitude, |X(k)|, by mapping the frequencies to one of the 12 pitch classes.



Chroma Feature Representation: A heatmap showing the energy distribution across the 12 pitch classes over time (harmonic content).

Spectral Shape Features

Spectral features describe the frequency distribution, measuring brightness, spread, and the shape of the spectrum. Basically, helps to differentiates High frequency from Low frequency sound.

Spectral centroid: The weighted average of the frequency bins

$$SC = \frac{\sum_{k=0}^{N_{FFT}-1} f_k \, S(k)}{\sum_{k=0}^{N_{FFT}-1} S(k)}$$

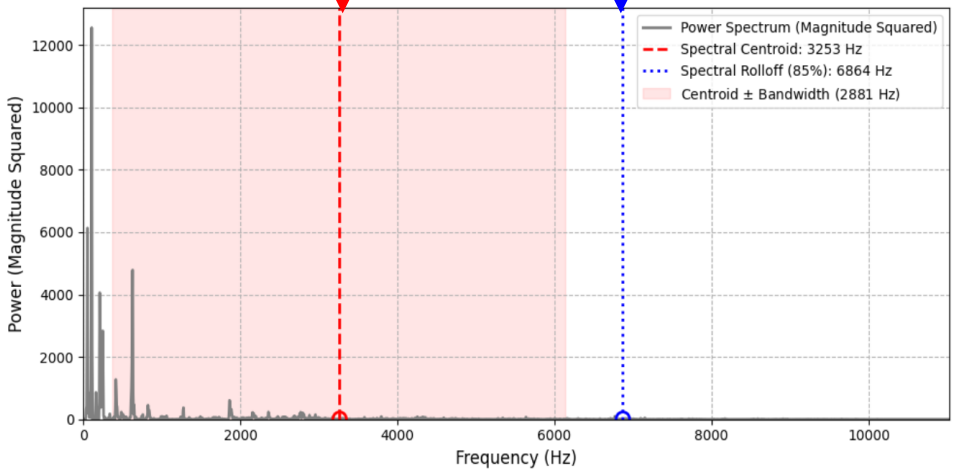
Frequency at bin k (in Hz)

Magnitude or power of the Fourier Transform at frequency bin k.

Spectral Rolloff: The frequency R_α (below which α (e.g., 85%) of the total spectral power is concentrated), high-frequency power boundary.

$$\sum_{k=0}^{R_\alpha} S(k) = \alpha \sum_{k=0}^{N_{FFT}-1} S(k)$$

(https://en.wikipedia.org/wiki/Spectral_centroid)



Annotated Frequency Spectrum: Centroid (brightness) is the center of mass, and Rolloff defines the high-frequency power boundary.

Time Domain - Zero Crossing Rate (ZCR)

ZCR measures how often the signal changes sign, indicating the noisiness or percussive nature of the sound.

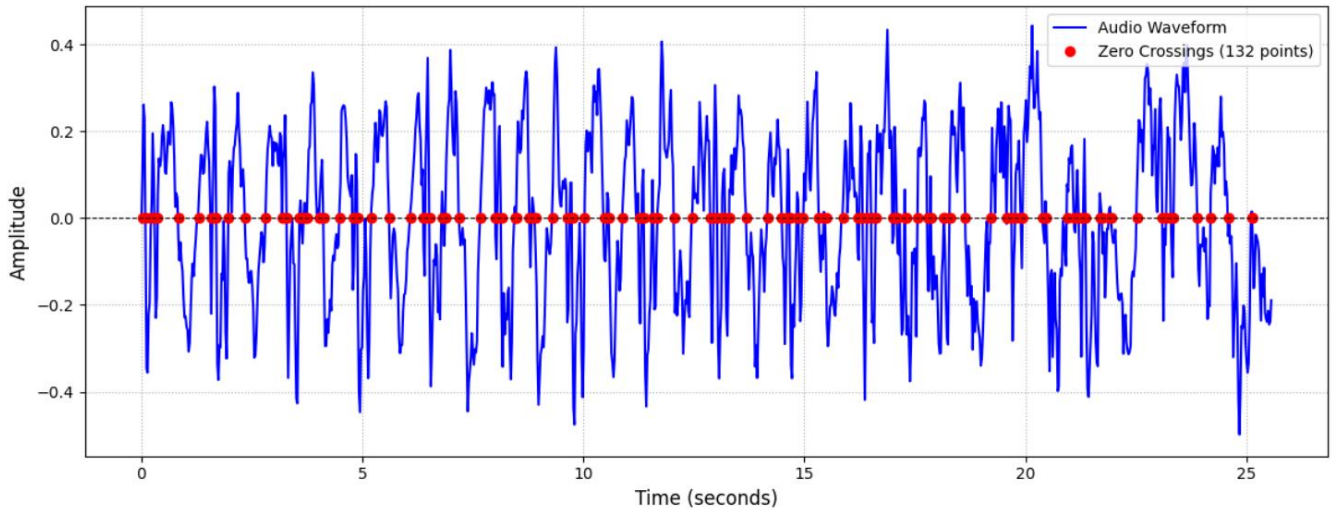
Audio sample at index n

Sign function: returns:
- 1; if the input is positive,
- -1; if negative, and
- 0; if zero.

$$\text{ZCR} = \frac{1}{2(N-1)} \sum_{n=1}^{N-1} \left| Sgn\left[\left(x(n)\right)\right] - Sgn\left[\left(x(n-1)\right)\right] \right|$$

(https://en.wikipedia.org/wiki/Zero-crossing_rate)

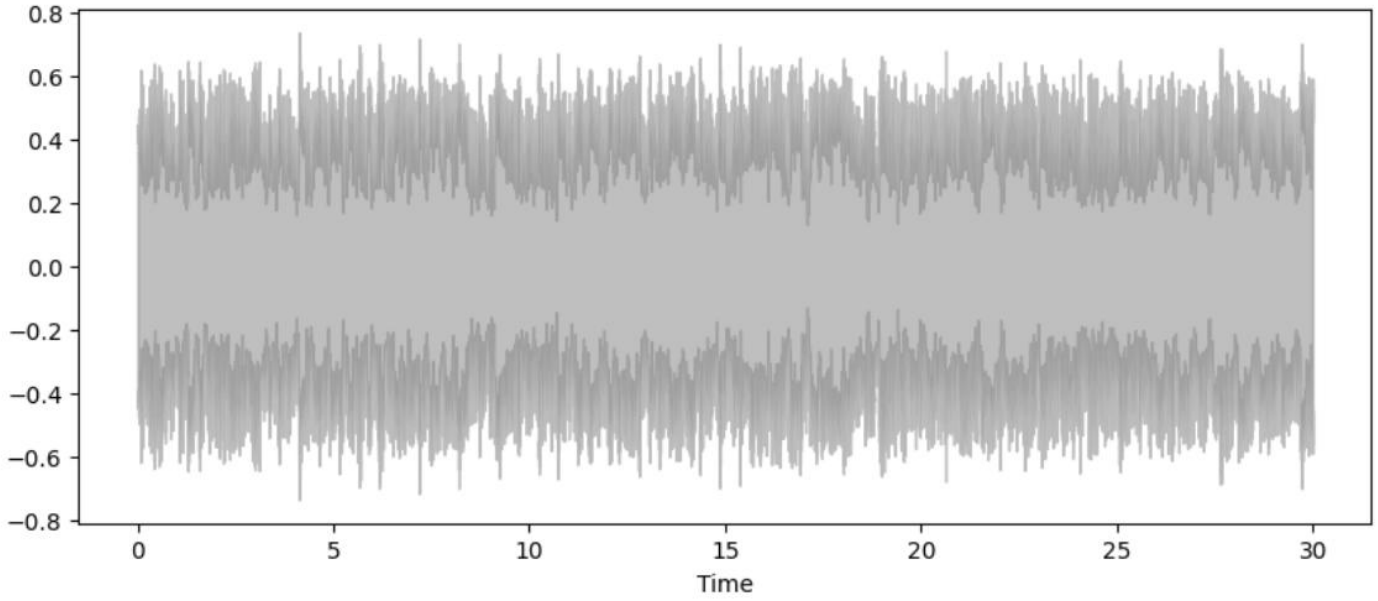
Equal to: 2 if a sign change occurs and 0 otherwise. We divide by 2 to count the crossing event.



Zero Crossings: Marks every point where the waveform crosses the zero-amplitude line; a high rate indicates noise.

Rhythmic Structure - Tempo Feature

Tempo captures the rhythmic speed (BPM), while beat tracking captures the exact timing of musical pulses (Ex: slow Classical vs. fast Disco/EDM).



Waveform with Detected Beats: Beat markers overlaid on the waveform to illustrate the rhythmic pulse and tempo.

Timbre (MFCC (Mel-Frequency Cepstral Coefficients))

- MFCCs capture the essential spectral envelope (timbre (tone quality)) of the sound, independent of pitch.
- Based on human auditory perception (Mel scale)

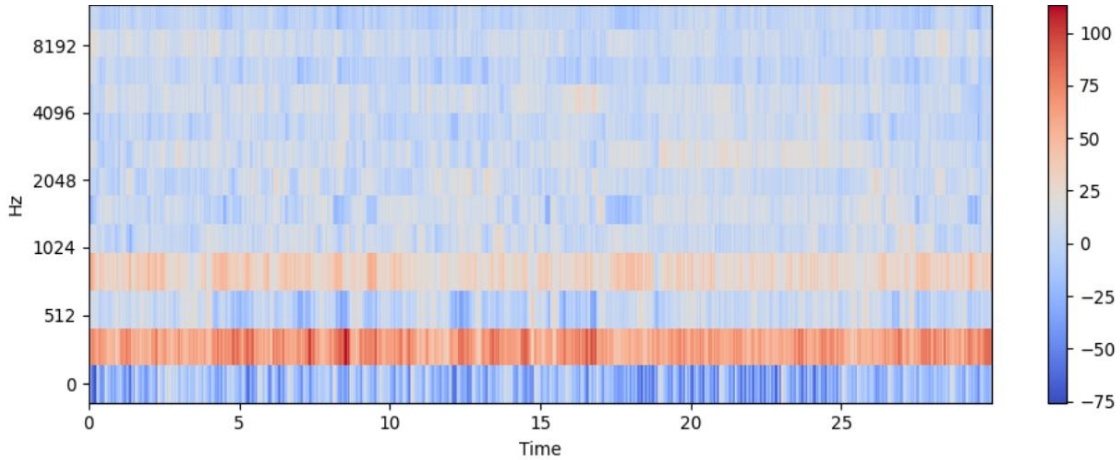
Step 1: Mel-Filter Bank Energies (E_m)
The power spectrum is passed through M triangular Mel filters $H_m(k)$

$$E_m = \sum_{k=0}^{N_{\mathrm{FFT}}-1} |X(k)|^2 \, H_m(k)$$

Step 2: Discrete Cosine Transform (DCT)
The DCT is applied to the logarithm of the Mel-filter bank energies to produce the MFCCs (c_j).

$$c_j = \sum_{m=1}^{M} \log(E_m) \cos\left[j\left(m - \frac{1}{2}\right)\frac{\pi}{M}\right], \qquad j = 1, \ldots, C$$

https://en.wikipedia.org/wiki/Mel-frequency_cepstrum



Mel Spectrogram: The spectral energy plotted on a Mel scale that matches human auditory perception.

Feature Dataset: Feature Aggregation & Final Vector

- Frame-level features are computed and aggregated using mean and variance, producing a fixed-length feature vector per audio file.
- Each file is converted into a single vector, with features stored in a tabular dataset format.
- The target label corresponds to the music genre. This format is suitable for classical machine

Note:

These Features are computed for many short frames of an audio signal, producing multiple values over time. Taking the mean and variance summarizes these into a single fixed-length feature vector.

```
df.columns

Index(['filename', 'label', 'length', 'chroma_stft_mean', 'chroma_stft_var',
       'rms_mean', 'rms_var', 'spectral_centroid_mean',
       'spectral_centroid_var', 'spectral_bandwidth_mean',
       'spectral_bandwidth_var', 'rolloff_mean', 'rolloff_var',
       'zero_crossing_rate_mean', 'zero_crossing_rate_var', 'harmony_mean',
       'harmony_var', 'perceptr_mean', 'perceptr_var', 'tempo', 'mfcc1_mean',
       'mfcc1_var', 'mfcc2_mean', 'mfcc2_var', 'mfcc3_mean', 'mfcc3_var',
       'mfcc4_mean', 'mfcc4_var', 'mfcc5_mean', 'mfcc5_var', 'mfcc6_mean',
       'mfcc6_var', 'mfcc7_mean', 'mfcc7_var', 'mfcc8_mean', 'mfcc8_var',
       'mfcc9_mean', 'mfcc9_var', 'mfcc10_mean', 'mfcc10_var', 'mfcc11_mean',
       'mfcc11_var', 'mfcc12_mean', 'mfcc12_var', 'mfcc13_mean', 'mfcc13_var',
       'mfcc14_mean', 'mfcc14_var', 'mfcc15_mean', 'mfcc15_var', 'mfcc16_mean',
       'mfcc16_var', 'mfcc17_mean', 'mfcc17_var', 'mfcc18_mean', 'mfcc18_var',
       'mfcc19_mean', 'mfcc19_var', 'mfcc20_mean', 'mfcc20_var'],
      dtype='object')
```

Dataset Columns/Features

Feature Vector

	filename	label	length	mfcc20_mean	mfcc20_var
0	blues.00000.wav	blues	30.013333	1.222467	46.941350
1	blues.00001.wav	blues	30.013333	0.530644	45.788700
2	blues.00002.wav	blues	30.013333	-2.238128	30.653150
3	blues.00003.wav	blues	30.013333	-3.405046	31.965258
4	blues.00004.wav	blues	30.013333	-11.704385	55.190254
⋮	⋮	⋮	⋮	⋮	⋮
997	rock.00098.wav	rock	30.013333	-3.587599	41.29636
998	rock.00099.wav	rock	30.013333	1.150108	49.73514

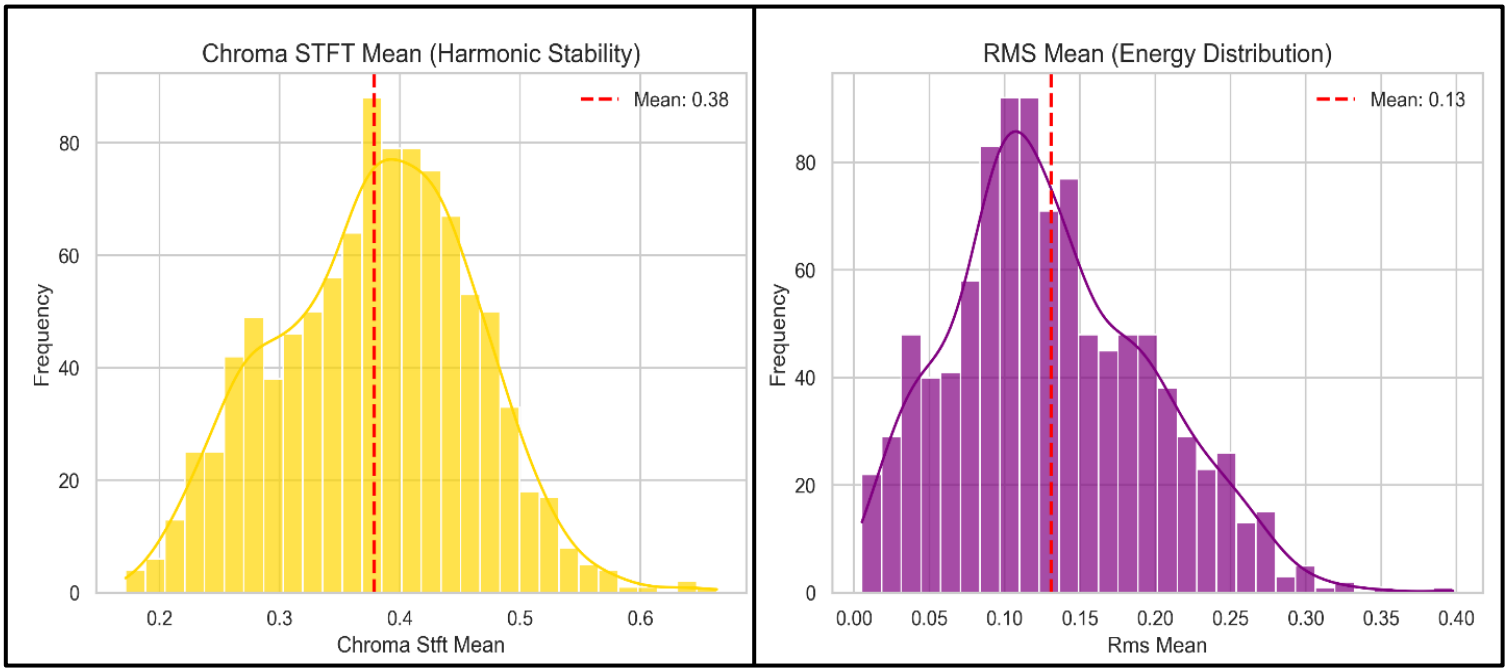
[999 x 60]

target label

- *Number of Unique Audio Files (Feature Vectors)*: **999**
- *Number of Features:* **58** [~ 'filename', 'label']
- *Target Label*: Music Genre [10 classes: blues, classical, country, disco, hiphop, metal, pop, reggae, rock, jazz]

Analysis on Features: Harmonic & Energy Analysis

How pitch and loudness differentiate genres.

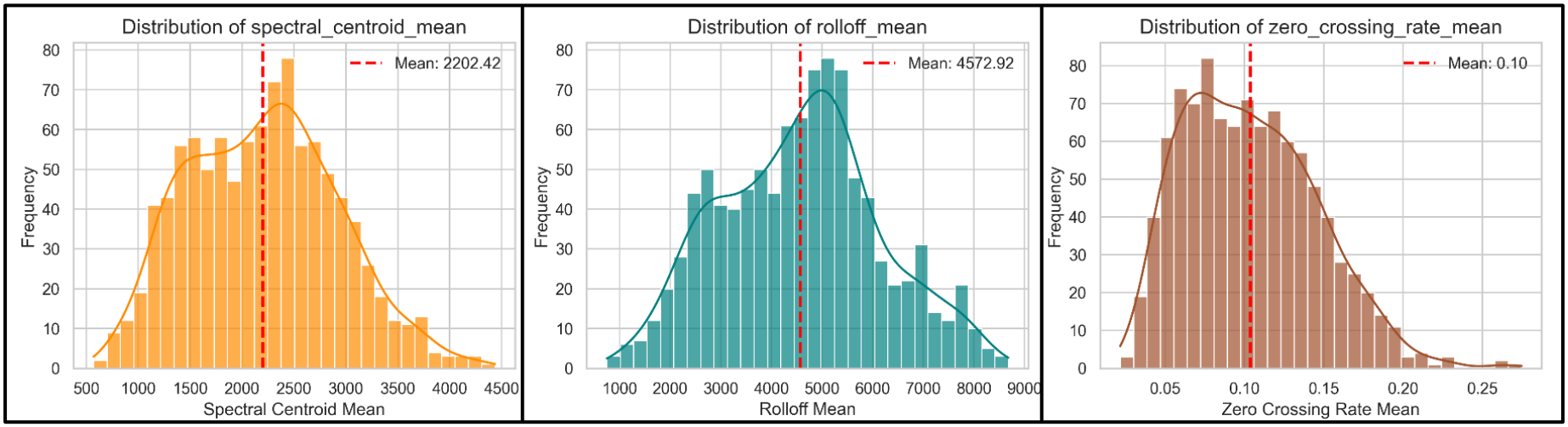


Harmonic Content & Energy

- *Harmonic Richness:* Chroma mean is roughly normal, while variance is long-tailed, highlighting complex harmonic genres like Jazz or Classical.
- *Energy Skewness:* RMS is strongly right-skewed, separating high-energy genres (Hiphop, Metal) from others.
- *Discriminative Feature:* Energy variance across genres makes it (RMS Mean) a primary feature for distinguishing "aggressive" vs. "mellow" styles.

Spectral Shape: Centroid, Rolloff, and ZCR

Differentiating "brightness" and "noisiness" between genres.

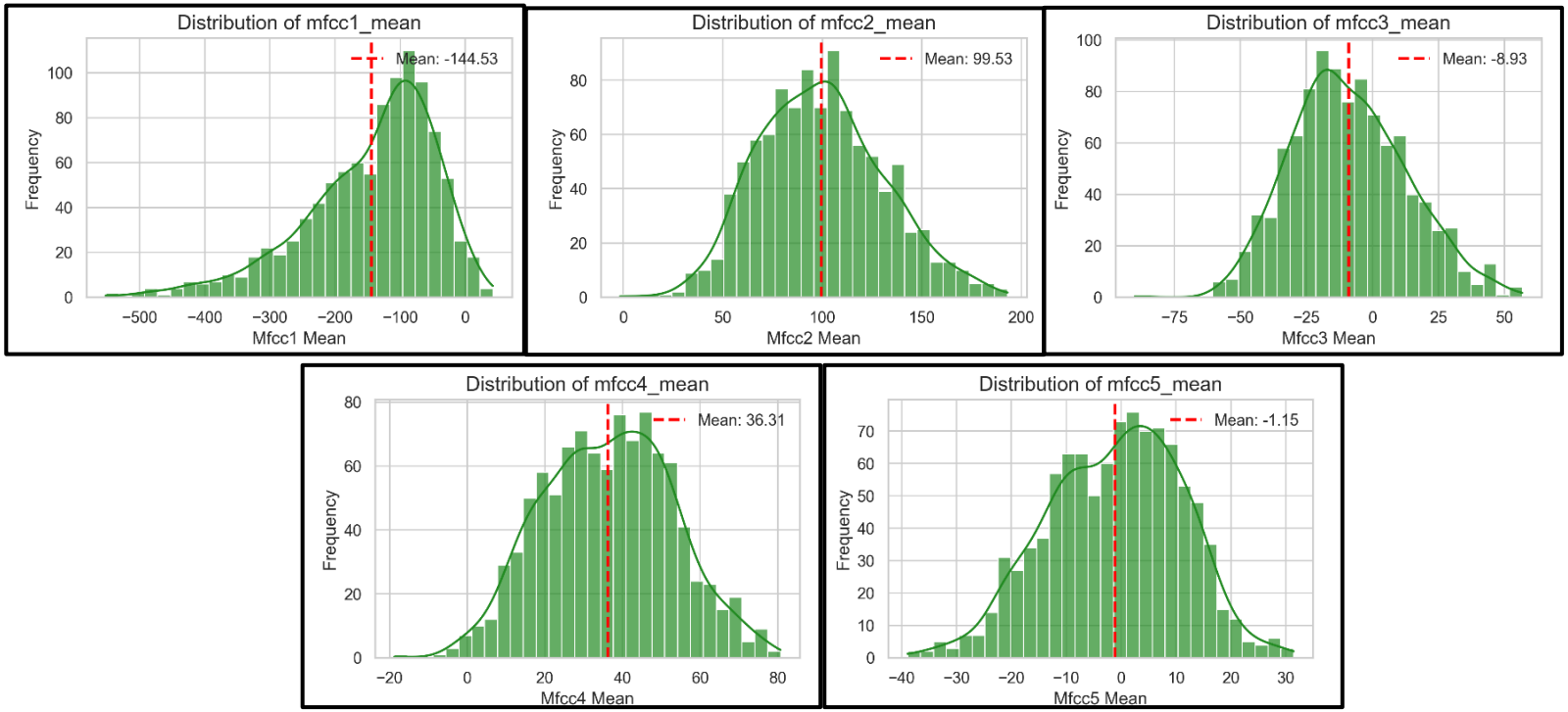


Histograms of Spectral Shape: Brightness and Noisiness Indicators

- *Brightness (Centroid):* Bell-shaped mean represents general brightness, metal/rock typically sit at the (right) higher end.
- *Texture Variability:* Rolloff and ZCR show variability in high-frequency content and noisiness.
- Therefore, these are crucial features because due to their long-tailed distributions they can effectively separate genre extremes.

Timbral Stability (MFCC Means)

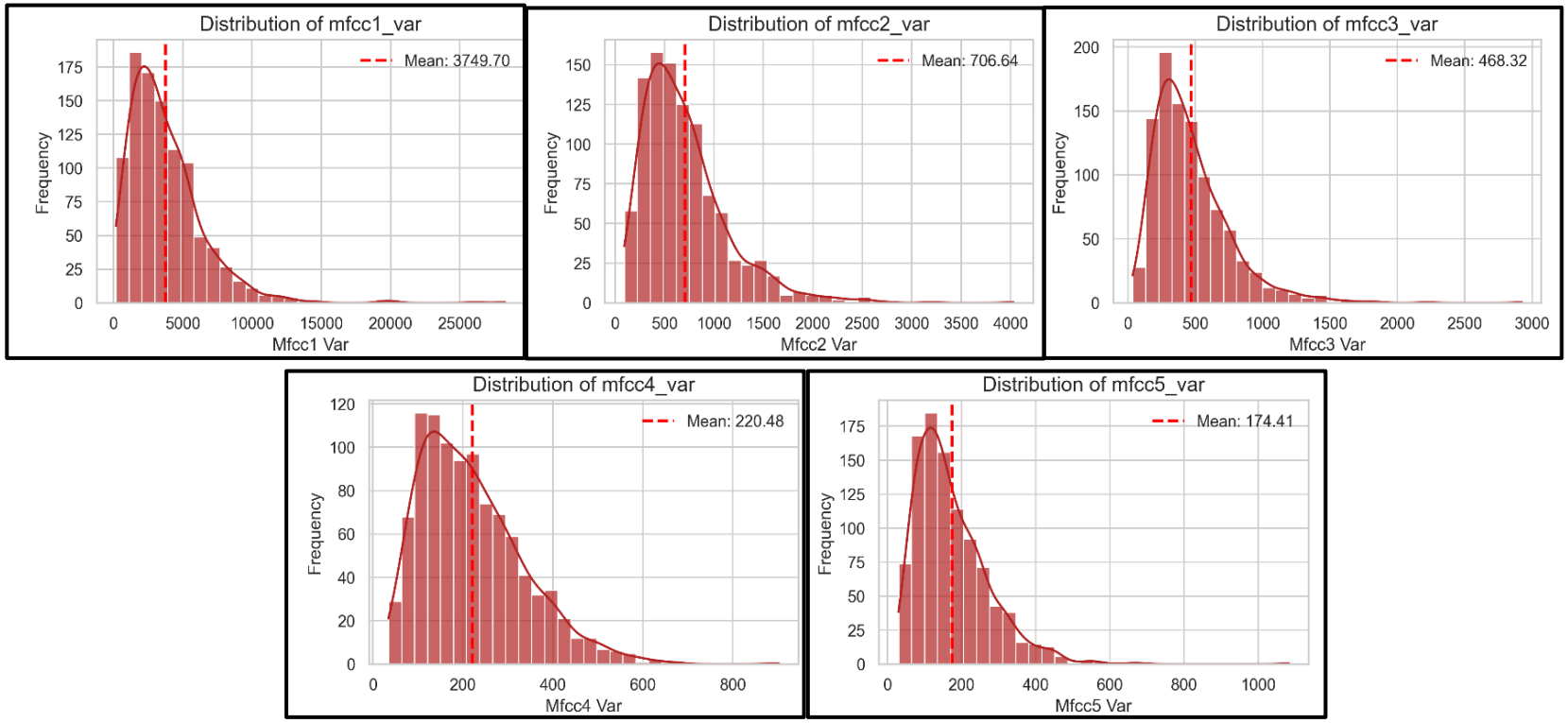
MFCCs are the backbone of this classification!



Stable Timbral Features: MFCC Means (Near – Normal Distribution)

- *Stable Distribution*: MFCC means follow roughly normal distributions, indicating stable timbral patterns.
- *Discriminative Potential*: Near-normal shapes suggest these features are highly effective for general genre classification.
- *Multimodality*: Bi-modal peaks in specific MFCCs suggest the feature space naturally contains two major, separate clusters, aiding the classifier in making clean distinctions between genres.

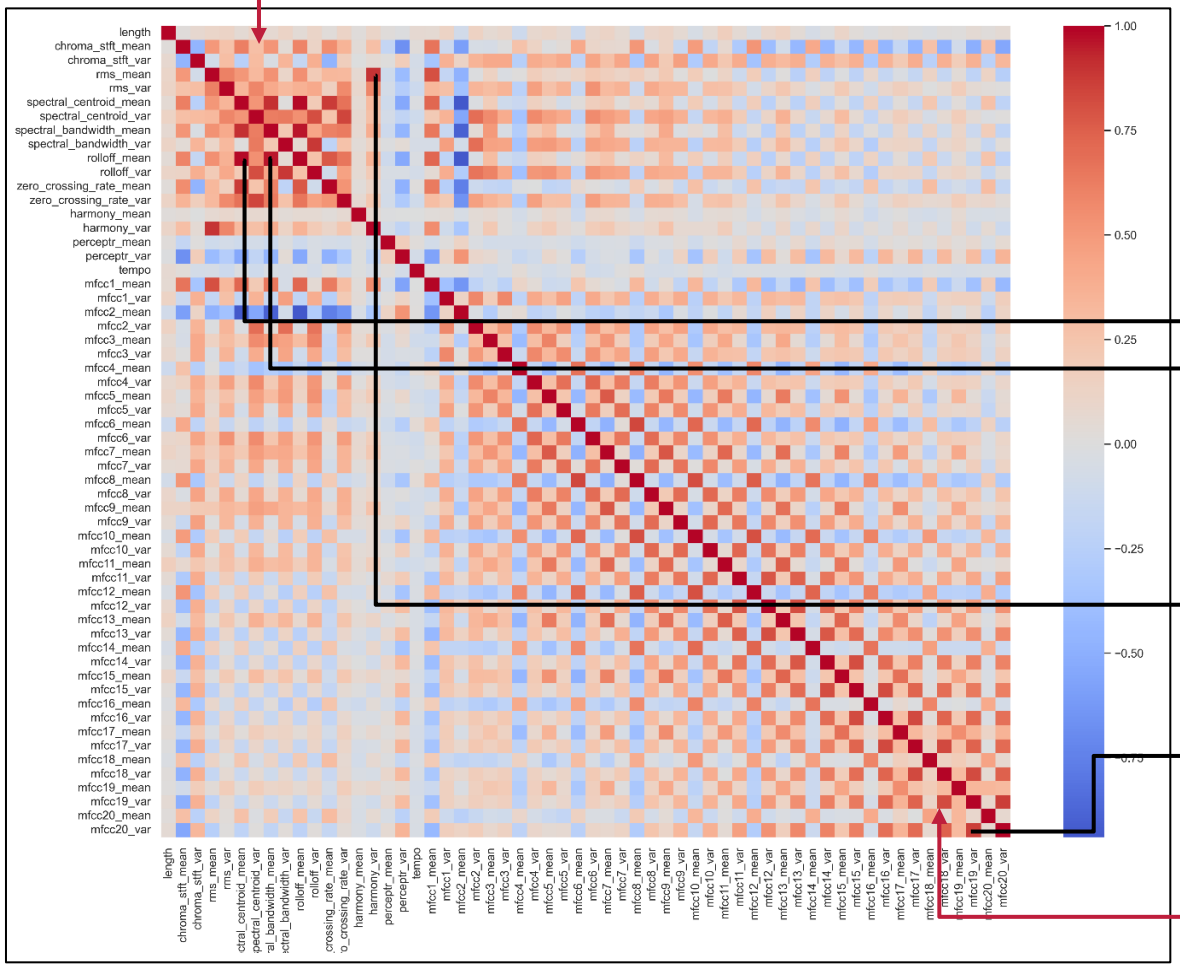
Dynamic Variability (MFCC Variances)



Timbral Dynamical Variability and Skewness: MFCC Variances (Right- Skewed)

- *High Skewness:* Variance features are heavily right-skewed with long tails.
- *Capturing Fluctuation:* These capture the dynamics and variation in timbre across different musical genres.
- Large variances indicate certain genres possess extreme timbre fluctuations, requiring careful scaling later.

Correlation In Features



Correlation Heatmap

A large block of high positive correlation (top-left quadrant) exists between *Mean spectral features* (Centroid, Rolloff, Bandwidth).

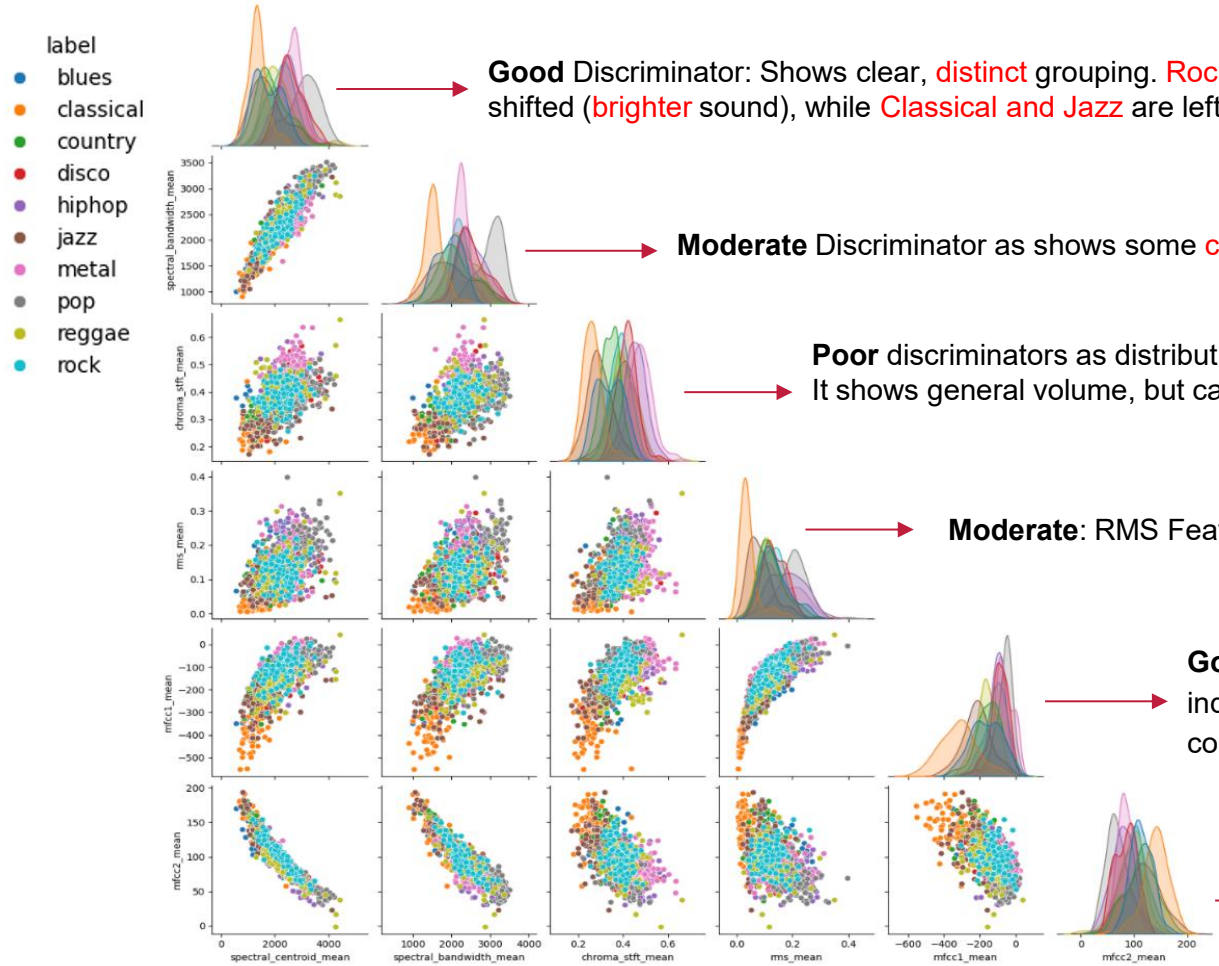
```
#Highly Correlated Pairs:
high_corr = corr.abs().unstack().sort_values(ascending=False)
high_corr = high_corr[high_corr < 1]   #Removing the diagonal
high_corr[high_corr > 0.85].head(20)
```

rolloff_mean	spectral_centroid_mean	0.979621
spectral_centroid_mean	rolloff_mean	0.979621
rolloff_mean	spectral_bandwidth_mean	0.956278
spectral_bandwidth_mean	rolloff_mean	0.956278
mfcc2_mean	spectral_centroid_mean	0.940176
spectral_centroid_mean	mfcc2_mean	0.940176
mfcc2_mean	rolloff_mean	0.934289
rolloff_mean	mfcc2_mean	0.934289
spectral_centroid_mean	spectral_bandwidth_mean	0.904512
spectral_bandwidth_mean	spectral_centroid_mean	0.904512
mfcc2_mean	spectral_bandwidth_mean	0.896706
spectral_bandwidth_mean	mfcc2_mean	0.896706
rms_mean	harmony_var	0.893937
harmony_var	rms_mean	0.893937
rolloff_var	spectral_bandwidth_var	0.884917
spectral_bandwidth_var	rolloff_var	0.884917
spectral_centroid_mean	zero_crossing_rate_mean	0.874667
zero_crossing_rate_mean	spectral_centroid_mean	0.874667
mfcc19_var	mfcc20_var	0.869332
mfcc20_var	mfcc19_var	0.869332

dtype: float64

High positive correlation is also visible among many *Variance features*, especially the high-order MFCC variances (bottom-right block).

Feature Overlap: PairPlots



Feature PairPlots

Spectral Feature

Good Discriminator: Shows clear, distinct grouping. Rock and Metal are heavily right-shifted (brighter sound), while Classical and Jazz are left-shifted (darker sound).

Moderate Discriminator as shows some clustering.

Chroma Feature

Poor discriminators as distributions for most genres heavily overlap. It shows general volume, but cannot cleanly separate groups.

RMS Feature

Moderate: RMS Features still show some separation in classes

MFCC Feature

Good: Classical and Blues on the lower end, indicating lower overall energy/loudness compared to genres like Rock and Metal.

Moderate: Shows separation between low-mean genres (Metal, Rock, Hip-Hop) and high-mean genres (Classical, Jazz, Blues).

Initial Genre Discriminability

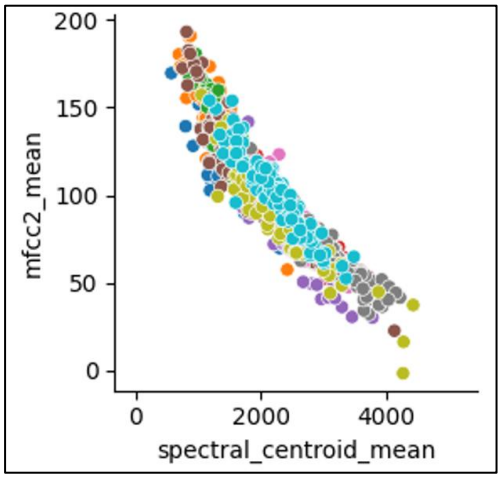


Fig 1

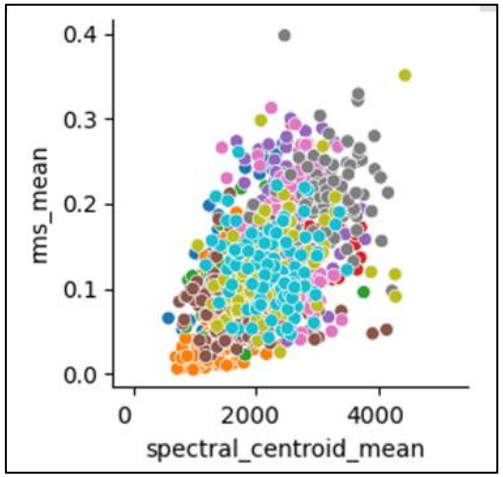


Fig 2

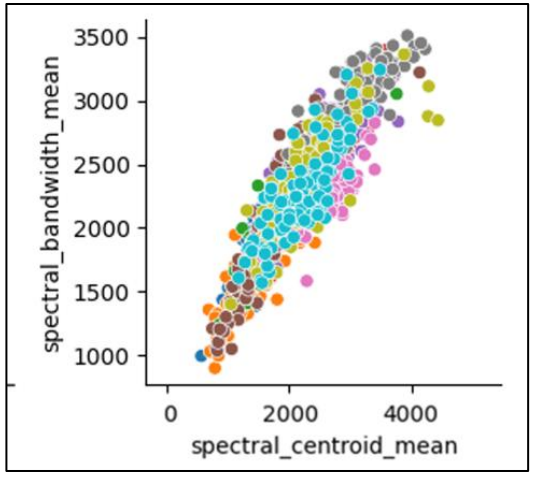


Fig 3

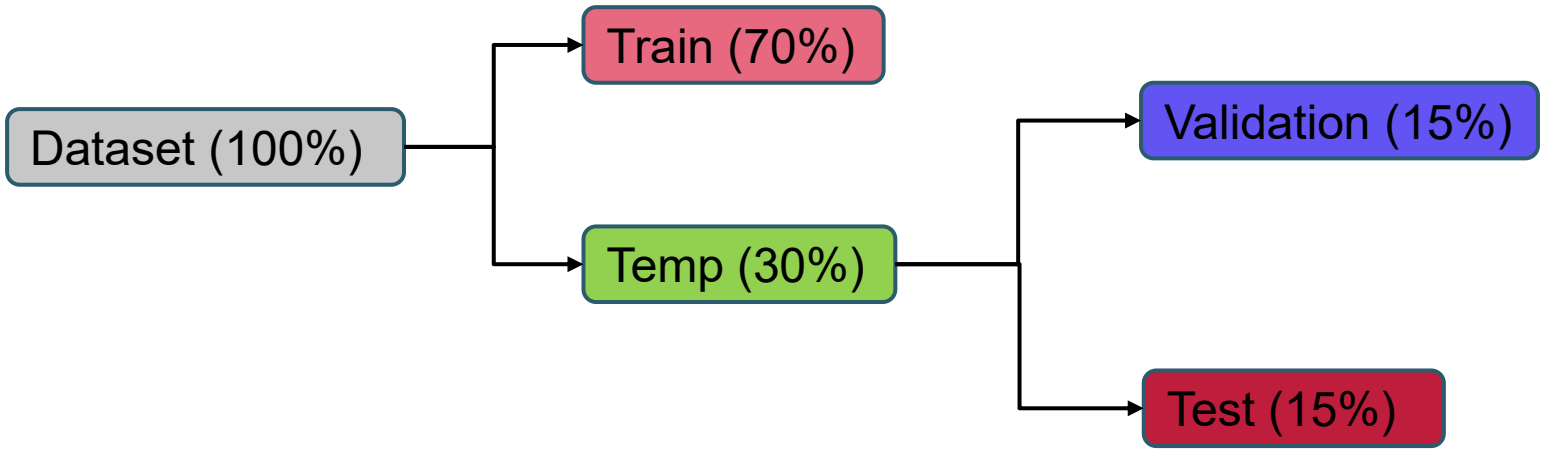
Feature Combination	Discriminatory Power	Cluster Separation Example
Spectral Centroid vs. MFCC2 Mean (Fig 1)	Highest	Separates Classical/Blues from Metal/Rock
Spectral Centroid vs. RMS Mean (Fig 2)	High	Isolates Classical from Metal
Spectral Centroid vs. Spectral Bandwidth (Fig 3)	Moderate	Separates extremes (Ex: Classical vs. Metal), but their strong correlation suggests redundant information.

Note

The **Pop, Disco, and Reggae** genres consistently cluster together in the center of most plots, exhibiting significant overlap and posing the greatest difficulty for separation using these core features.

Dataset Splitting Strategy

Train – Validation - Test Split:



- Stratified sampling is used to preserve the genre distribution across all splits, ensuring fair model evaluation.
- Stratification ensures that the proportion of each genre c is approximately the same in the training, validation, and test sets.
- Prevents data leakage during preprocessing.

Feature Transformation

- Extracted features have different ranges and scales
- Feature transformation is required before model training
- *Standardization* ensures fair contribution of all features
- *PCA* reduces redundancy and dimensionality

Note

The scaler is fitted only on the training data and then applied to validation and test sets to avoid data leakage.

Feature Standardization:

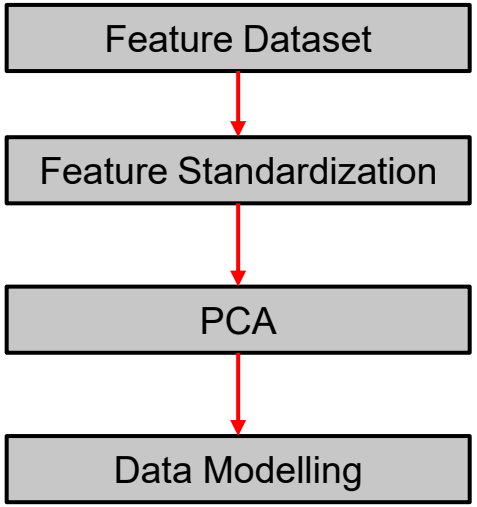
- StandardScaler() used to normalize Mean-centered and variance-scaled features.

- Prevents dominance of large-scale features

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

μ → Mean of the training samples (μ)

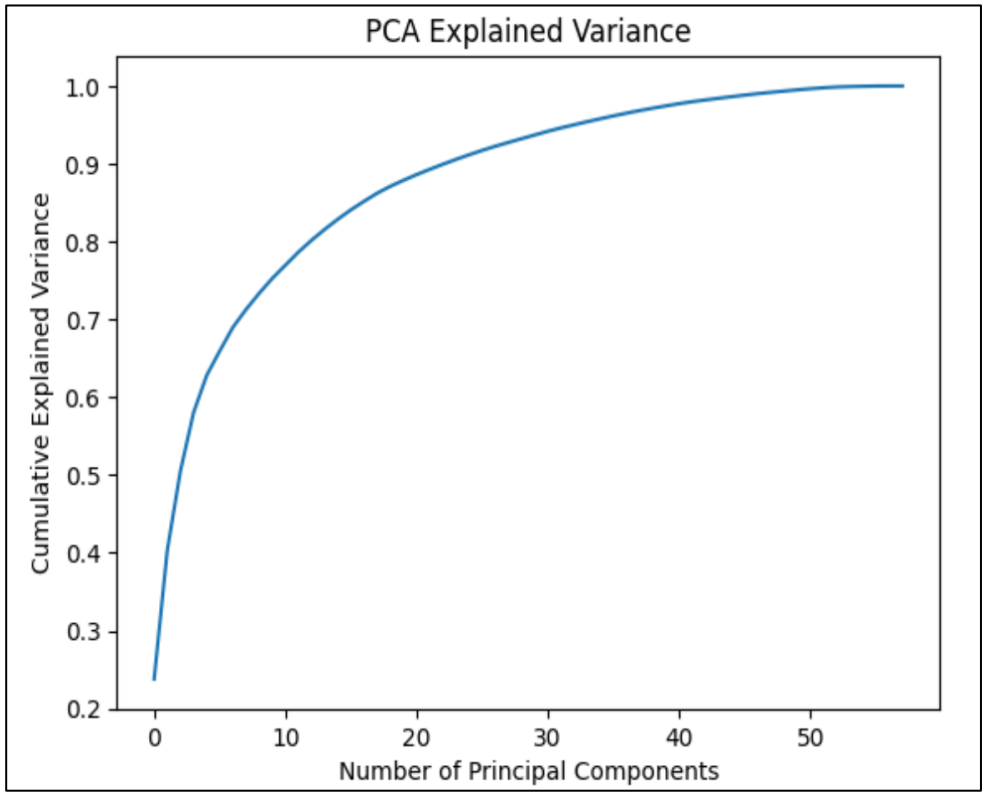
σ → standard deviation of the training samples (σ)



- Essential for PCA and scale-sensitive models (For ex: kNN)

- Improves convergence, stability, and fair feature contribution

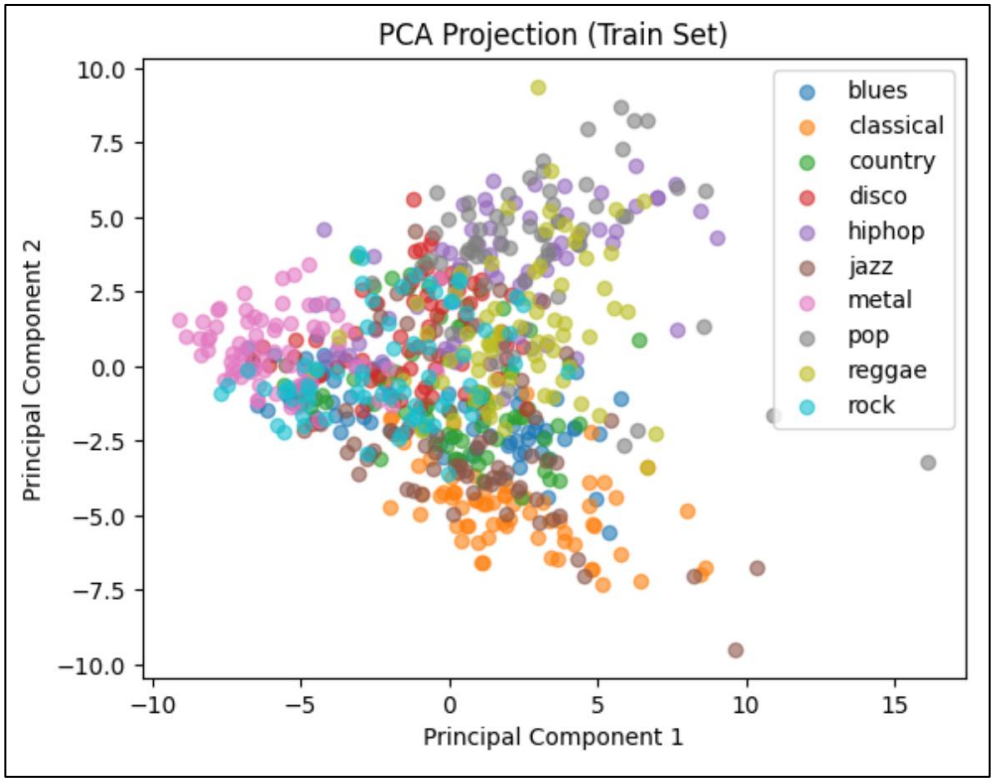
PCA Explained Variance: Dimensionality Reduction



PCA Cumulative Explained Variance

- High redundancy detected (r > 0.85).
- Reduced features from 58 to 33 PCs (42% reduction).
- Retained 95% cumulative variance.
- Removed multicollinearity and simplified training.

Genre Separation in PC Space: PCA Visualization



PCA 2D Projection scatter plot

- High Dimensional Data projected onto PC1 & PC2 for low-dimensional view.
- Distinct Clusters for some genres (For ex: Classical, Metal).
- Overlap among Pop, Disco, Rock shows shared features.
- Confirms PCA preserves key variance for effective classification.

Data Modelling:
K-Nearest Neighbors (KNN)

Baseline

Validation Accuracy: 0.6333333333333333

	precision	recall	f1-score	support
blues	0.89	0.53	0.67	15
classical	0.75	1.00	0.86	15
country	0.55	0.80	0.65	15
disco	0.47	0.47	0.47	15
hiphop	0.62	0.67	0.65	15
jazz	0.70	0.47	0.56	15
metal	0.85	0.73	0.79	15
pop	0.75	0.80	0.77	15
reggae	0.42	0.33	0.37	15
rock	0.47	0.53	0.50	15
accuracy			0.63	150
macro avg	0.65	0.63	0.63	150
weighted avg	0.65	0.63	0.63	150

(n_neighbors = 5, uniform weights):
achieved a Validation Accuracy: 63.3%.

Hyperparameter Tuning

Validation Accuracy: 0.64

	precision	recall	f1-score	support
blues	1.00	0.53	0.70	15
classical	0.83	1.00	0.91	15
country	0.71	0.80	0.75	15
disco	0.38	0.40	0.39	15
hiphop	0.69	0.60	0.64	15
jazz	0.75	0.40	0.52	15
metal	0.79	0.73	0.76	15
pop	0.57	0.80	0.67	15
reggae	0.53	0.53	0.53	15
rock	0.45	0.60	0.51	15
accuracy			0.64	150
macro avg	0.67	0.64	0.64	150
weighted avg	0.67	0.64	0.64	150

- GridSearchCV with 5-fold cross-validation was used on the training set to optimize parameters.
- Best Parameters: {'n_neighbors': 7, 'p': 1:(Manhattan Distance), 'weights': 'distance'}.
- **Tuning Gain:** The tuned model slightly improved the Validation Accuracy: 64.0%.

KNN Final Test Performance

Note

The test set is used only once for final unbiased evaluation on Best Model after Hyper Parameter Tuning.

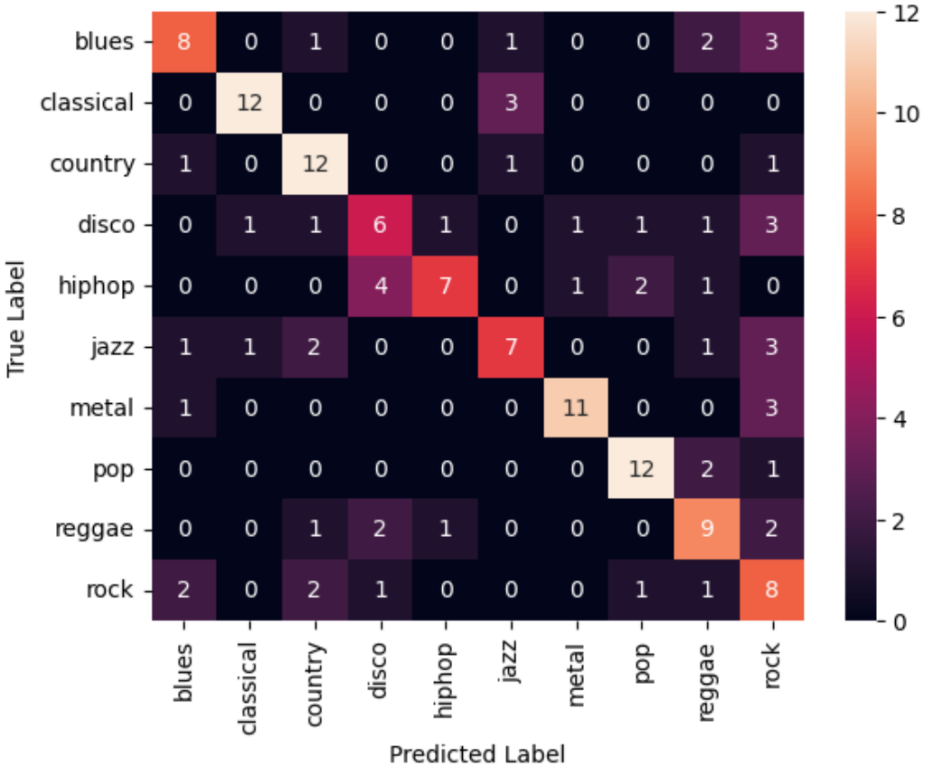
Best Model_KNN

Test Accuracy: 0.6133333333333333

	precision	recall	f1-score	support
blues	0.62	0.53	0.57	15
classical	0.86	0.80	0.83	15
country	0.63	0.80	0.71	15
disco	0.46	0.40	0.43	15
hiphop	0.78	0.47	0.58	15
jazz	0.58	0.47	0.52	15
metal	0.85	0.73	0.79	15
pop	0.75	0.80	0.77	15
reggae	0.53	0.60	0.56	15
rock	0.33	0.53	0.41	15
accuracy			0.61	150
macro avg	0.64	0.61	0.62	150
weighted avg	0.64	0.61	0.62	150

- Test Accuracy: 61.3%
- Macro F1-Score: 0.62 (Average F1 across all genres)
- Top Performing Genres (High F1/Recall):
  - Classical: 0.83 - F1-score (80% Recall)
  - Metal: 0.79 - F1-score (73% Recall)
  - Pop: 0.77 - F1-score (80% Recall)
- Challenging Genres (Low F1/Recall):
  - Rock: 0.41 - F1-score (33% Precision)
  - Disco: 0.43 - F1-score (40% Recall)

KNN Performance: Confusion Matrix



Confusion Matrix: KNN (Test Set)

- High True Positives (Good Performance):
  - Classical, Country and Pop: 12/15 correctly classified (Distinct spectral/timbral features aid separation)
  - Metal: 11/15 correctly classified (good separation for high-energy genres)
- Systematic Misclassifications (Model Weakness):
  - Rock Misclassified: Only 53% recall, confused with Reggae and Disco
  - Reggae Confusion: Confused with Disco and Rock (9/15 correct)
  - Hip-hop: High confusion, often misclassified as Jazz

Overall: The diagonal indicates acceptable performance for genres with distinct timbral features, but Confusion among rhythmically/spectrally similar genres (Pop, Disco, Rock, Reggae).

Logistic Regression

Baseline Model

Validation Accuracy: 0.7733333333333333

	precision	recall	f1-score	support
blues	0.90	0.60	0.72	15
classical	1.00	0.93	0.97	15
country	0.86	0.80	0.83	15
disco	0.50	0.60	0.55	15
hiphop	0.85	0.73	0.79	15
jazz	0.83	1.00	0.91	15
metal	0.76	0.87	0.81	15
pop	0.87	0.87	0.87	15
reggae	0.71	0.80	0.75	15
rock	0.57	0.53	0.55	15
accuracy			0.77	150
macro avg	0.78	0.77	0.77	150
weighted avg	0.78	0.77	0.77	150

Hyperparameter Tuning

Validation Accuracy: 0.7866666666666666

	precision	recall	f1-score	support
blues	0.90	0.60	0.72	15
classical	1.00	0.93	0.97	15
country	0.86	0.80	0.83	15
disco	0.57	0.53	0.55	15
hiphop	0.69	0.73	0.71	15
jazz	0.83	1.00	0.91	15
metal	0.87	0.87	0.87	15
pop	0.78	0.93	0.85	15
reggae	0.71	0.80	0.75	15
rock	0.71	0.67	0.69	15
accuracy			0.79	150
macro avg	0.79	0.79	0.78	150
weighted avg	0.79	0.79	0.78	150

The Initial model achieved a strong Validation Accuracy: 77.3%.

- GridSearchCV: used to tune Inverse regularization Stregnth (C) to prevent overfitting.
- Best params: { C = 0.1, L2: Ridge penalty, lbfgs solver }.
- Validation accuracy improved: 78.7%.

Logistic Regression Final Test Performance

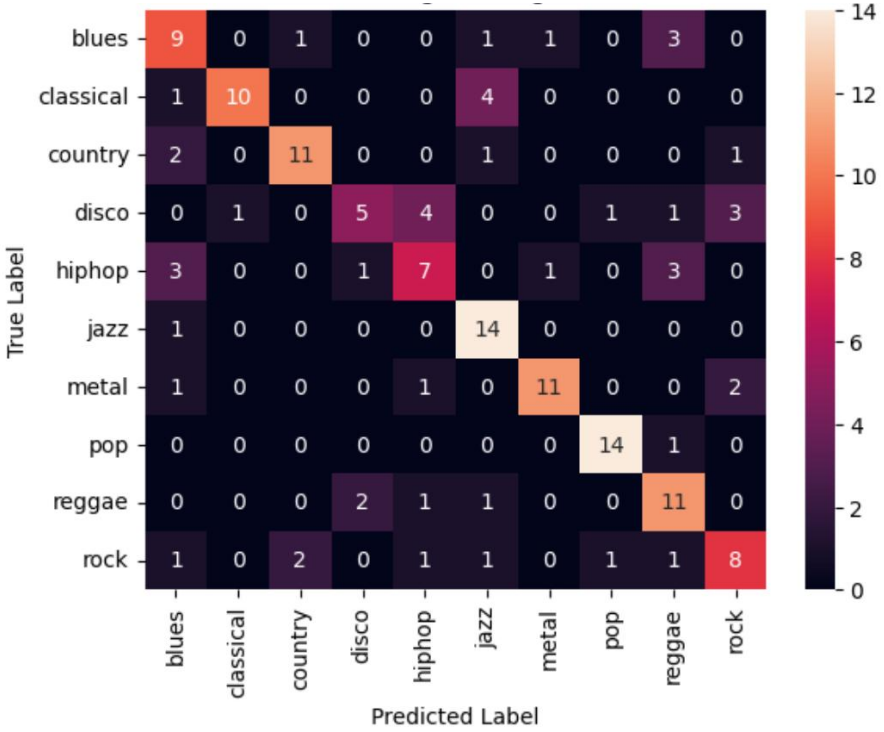
Best Model_LogReg

Test Accuracy: 0.6666666666666666

	precision	recall	f1-score	support
blues	0.50	0.60	0.55	15
classical	0.91	0.67	0.77	15
country	0.79	0.73	0.76	15
disco	0.62	0.33	0.43	15
hiphop	0.50	0.47	0.48	15
jazz	0.64	0.93	0.76	15
metal	0.85	0.73	0.79	15
pop	0.88	0.93	0.90	15
reggae	0.55	0.73	0.63	15
rock	0.57	0.53	0.55	15
accuracy			0.67	150
macro avg	0.68	0.67	0.66	150
weighted avg	0.68	0.67	0.66	150

- Test Accuracy: 66.7%
  (A significant improvement over KNN's 61.3%).
- Macro F1-Score: 0.66
- Top Performing Genres (High F1/Recall):
  - Pop: 0.90 - F1-score (93% Recall)
  - Classical: 0.77 - F1-score
  - Jazz: 0.76 - F1-score (93% Recall)

Logistic Performance: Confusion Matrix



Confusion Matrix: Logistic Regression (Test Set)

<u>Overall Gain</u>
Accuracy improved by >5% over kNN

<u>Strong Performance</u>
- Pop: 14/15 correctly classified
- Jazz: 14/15 correct (93% recall), highly linearly separable

<u>Ongoing Challenges</u>
- Disco: Low recall (33%), confused with Hip-hop and Rock
- Hip-hop: Low recall (47%), mixed with Blues, Metal, Reggae

Decision Trees

Baseline

Validation Accuracy: 0.49333333333333335

	precision	recall	f1-score	support
blues	0.67	0.40	0.50	15
classical	0.86	0.80	0.83	15
country	0.40	0.53	0.46	15
disco	0.38	0.40	0.39	15
hiphop	0.39	0.47	0.42	15
jazz	0.69	0.73	0.71	15
metal	0.42	0.53	0.47	15
pop	0.54	0.47	0.50	15
reggae	0.50	0.47	0.48	15
rock	0.18	0.13	0.15	15
accuracy			0.49	150
macro avg	0.50	0.49	0.49	150
weighted avg	0.50	0.49	0.49	150

The initial, unconstrained DT:
(which tends to overfit) achieved validation Accuracy of 49.3%.

Hyperparameter Tuning

Validation Accuracy: 0.4666666666666667

	precision	recall	f1-score	support
blues	0.75	0.40	0.52	15
classical	0.86	0.80	0.83	15
country	0.33	0.60	0.43	15
disco	0.39	0.47	0.42	15
hiphop	0.26	0.33	0.29	15
jazz	0.53	0.53	0.53	15
metal	0.62	0.53	0.57	15
pop	0.70	0.47	0.56	15
reggae	0.38	0.20	0.26	15
rock	0.28	0.33	0.30	15
accuracy			0.47	150
macro avg	0.51	0.47	0.47	150
weighted avg	0.51	0.47	0.47	150

- GridSearchCV: used to optimize constraints (max_depth, min_samples_leaf, etc.) to control complexity and reduce variance.
- Best Parameters {'criterion': 'gini', 'max_depth': 10, 'min_samples_leaf': 2, 'min_samples_split': 2}.
- Tuning Outcome:
  - Validation accuracy dropped to 46.7%
  - Single tree struggles to model complex patterns without overfitting

Decision Trees Final Test Performance

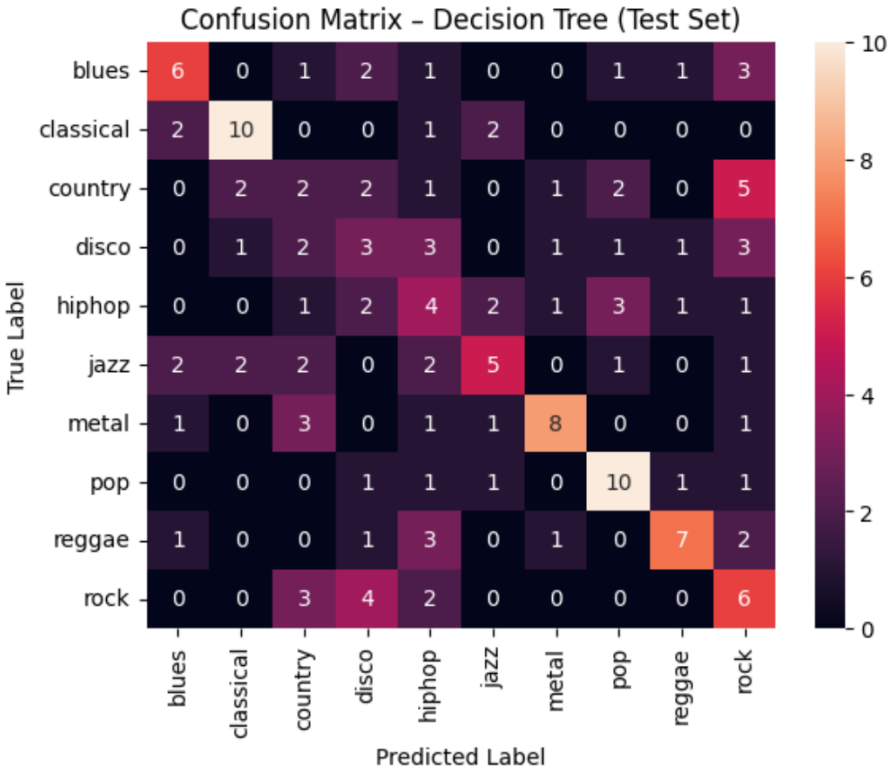
Best Model_DT

Test Accuracy: 0.4066666666666667

	precision	recall	f1-score	support
blues	0.50	0.40	0.44	15
classical	0.67	0.67	0.67	15
country	0.14	0.13	0.14	15
disco	0.20	0.20	0.20	15
hiphop	0.21	0.27	0.24	15
jazz	0.45	0.33	0.38	15
metal	0.67	0.53	0.59	15
pop	0.56	0.67	0.61	15
reggae	0.64	0.47	0.54	15
rock	0.26	0.40	0.32	15
accuracy			0.41	150
macro avg	0.43	0.41	0.41	150
weighted avg	0.43	0.41	0.41	150

- Test Accuracy: 40.7%
  (The lowest performing model)
- Macro F1-Score: 0.41
- Top Performing Genres (High F1/Recall):
  - Classical: 0.67 - F1-score (67% Recall)
  - Pop: 0.61 - F1-score (67% Recall)

Decision Trees: Confusion Matrix

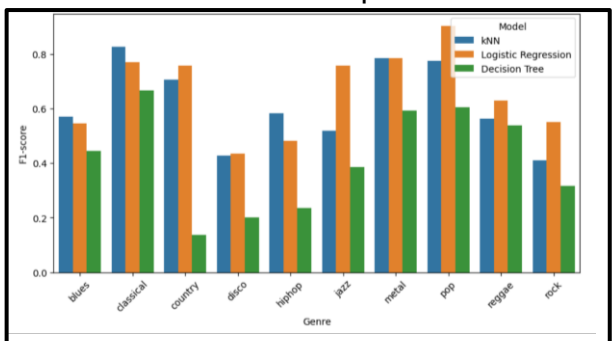


- Poor Generalization
  - Widespread off-diagonal errors
  - Low test accuracy indicates over-generalization
- Worst Recall
  - Country: 13% - recall (often misclassified as Rock)
  - Disco: 20% - recall
- High Confusion:
  - Jazz: 5/15 correct; confused with Hip-hop and Classical
  - Rock: 6/15 correct; mixed with Country, Disco, Reggae

Conclusion: The decision boundaries created by a single Decision Tree are too rigid and failed to capture the smooth, complex relationships present in the PCA-transformed audio feature space.

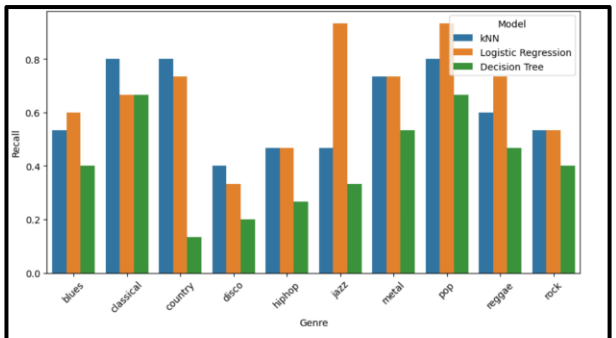
Model Comparison: Genre Wise Analysis

F1 Score Comparison



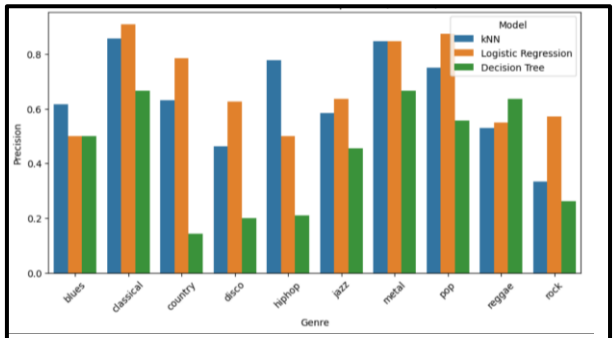
	Genre	kNN	Logistic Regression	Decision Tree
0	blues	0.571429	0.545455	0.444444
1	classical	0.827586	0.769231	0.666667
2	country	0.705882	0.758621	0.137931
3	disco	0.428571	0.434783	0.200000
4	hiphop	0.583333	0.482759	0.235294
5	jazz	0.518519	0.756757	0.384615
6	metal	0.785714	0.785714	0.592593
7	pop	0.774194	0.903226	0.606061
8	reggae	0.562500	0.628571	0.538462
9	rock	0.410256	0.551724	0.315789

Recall Comparison



	Genre	kNN	Logistic Regression	Decision Tree
0	blues	0.533333	0.600000	0.400000
1	classical	0.800000	0.666667	0.666667
2	country	0.800000	0.733333	0.133333
3	disco	0.400000	0.333333	0.200000
4	hiphop	0.466667	0.466667	0.266667
5	jazz	0.466667	0.933333	0.333333
6	metal	0.733333	0.733333	0.533333
7	pop	0.800000	0.933333	0.666667
8	reggae	0.600000	0.733333	0.466667
9	rock	0.533333	0.533333	0.400000

Precision Comparison



	Genre	kNN	Logistic Regression	Decision Tree
0	blues	0.615385	0.500000	0.500000
1	classical	0.857143	0.909091	0.666667
2	country	0.631579	0.785714	0.142857
3	disco	0.461538	0.625000	0.200000
4	hiphop	0.777778	0.500000	0.210526
5	jazz	0.583333	0.636364	0.454545
6	metal	0.846154	0.846154	0.666667
7	pop	0.750000	0.875000	0.555556
8	reggae	0.529412	0.550000	0.636364
9	rock	0.333333	0.571429	0.260870

- **Logistic Regression (LR) Dominance:** Highest F1 in 6/10 genres, including Pop (0.90), Jazz (0.76), Country (0.76).
- **KNN's Strength:** KNN remained competitive in highly separable genres.
- **Shared Strength:** Both LR & KNN excel on Metal (0.79), showing its distinct timbral features regardless of the classification algorithm used.
- **Decision Tree's Weakness:** Worst overall, particularly Country (0.14) and Disco (0.20), highlighting its lack of generalization.
- **Persistent Challenge:** All three models struggled most with **Disco** (max F1: 0.43), **Rock** (max F1: 0.55), **HipHop** (max F1: 0.58) and **Reggae** (max F1: 0.62) reinforcing the finding from the Confusion Matrices that these genres have the highest acoustic overlap.

Advantages and Disadvantages

Approach	Traditional Models	Similar Methods (Deep Learning (CNN/RNN))
Core Method	Linear/Non-linear model on **Hand-Engineered** Features (MFCCs, Spectral Centroid etc.) compressed by PCA.	**Learns Features** automatically from raw audio or spectrograms using deep neural networks.
Performance	Strong Baseline (Test Acc. 67%): Good **for linearly separable** features.	State-of-the-Art (Acc. approx. 80-95%): Achieves **high acc.** by identifying complex, hierarchical **patterns**.
Advantage	High Interpretability & Resource Efficient (Fast, **low** hardware **needs**).	Highest Accuracy as it **overcomes** the **limitations** of fixed, human-engineered features.
Disadvantage	Accuracy Ceiling is limited by fixed features. **Not scalable** to massive datasets.	Resource **Intensive** as requires **significant GPU power** and massive datasets and Black Box: Learned features are **uninterpretable**.
When to USE It	**Small**, balanced **datasets** or projects prioritizing **interpretability** and low resource use.	Projects demanding the **highest** possible **accuracy** and when **vast** quantities of training **data** and hardware are available.
When to AVOID It	When the goal is **80%+** accuracy or handling large-scale, **unstructured** commercial **data**.	When interpretability or **limited** computational **resources** are key constraints.

References

- Faruqui, Z., McIntire, M. S., Dubey, R., & McEntee, J. (2025). *Explainability of CNN based classification models for acoustic signal* (arXiv:2509.08717v1).

(https://arxiv.org/abs/2509.08717v1)

- Sturm, B. L. (2013). *The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use*. arXiv.

(https://arxiv.org/abs/1306.1461)

- Peeters, G. (2003). *Cuidado: A real-time audio feature extraction system*.

(http://recherche.ircam.fr/equipes/analyse-synthese/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf)

- Gouyon, F., Pachet, F., & Delerue, O. (2000). On the use of zero-crossing rate for classification of percussive sounds. *DAFX-00*, Verona, Italy. Accessed April 26, 2011.

(https://www.dafx.de/paper-archive/2000/pdf/GouyonPachetDelerue.pdf?utm_source=chatgpt.com)

- Librosa Development Team. (2025). *Librosa documentation*.

(https://librosa.org/doc/latest/index.html)

- scikit-learn Developers. (2025). *sklearn.preprocessing.StandardScaler*.

(https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html)

- scikit-learn Developers. (2025). *sklearn.model_selection.GridSearchCV*.

(https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html)

- Gtzan Data

(https://www.tensorflow.org/datasets/catalog/gtzan)

Thank You ☺