Music Genre Classification

Kyrus Wankadiya, Junichi Takano, Ritwik Awasthi

Overview and Purpose of Project

- Tasked with creating a model that classifies a musical audio file's genre as accurately as possible

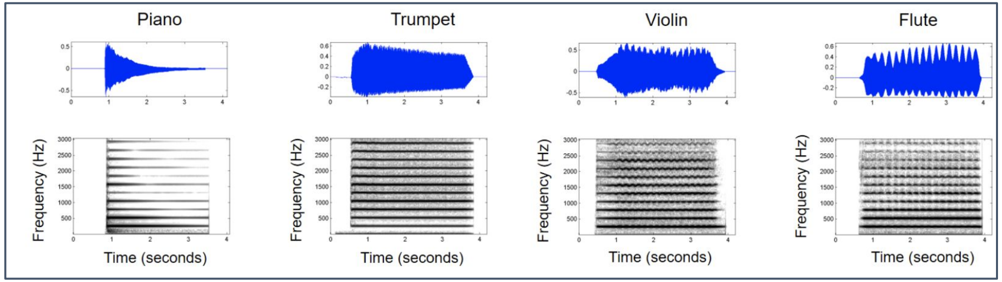
- Given 1,000 music audio files:
  - 100 30-second clips of music in each of 10 different genres
  - https://www.kaggle.com/datasets/carlthome/gtzan-genre-collection
  - blues, classical, country, disco, hiphop, jazz, metal, pop, reggae, rock

- Use cases of genre classification
  - Any music app such as Spotify, Apple Music, YouTube, or Shazam
  - Can be used to make recommendations and find songs that would be similar to a user's listening habits
  - Social media platforms with sophisticated techniques to gain insights from any audio and make recommendations or advising

Data Extraction

- Python Package: Librosa
- Read in .au or .wav files and return the following data of each audio clip:
  - MFCCs (Mel-Frequency Cepstral Coefficients)
  - Chroma features
  - Spectral Centroid
  - Spectral Bandwidth
  - ZCR (Zero Crossing Rate)
  - RMS Energy
  - Onset Envelope
  - Tempo
  - Spectral Contrast
  - Tonnetz
- Mean, standard deviation, minimum, and maximum values were used for all features that output multiple values across an audio segment.

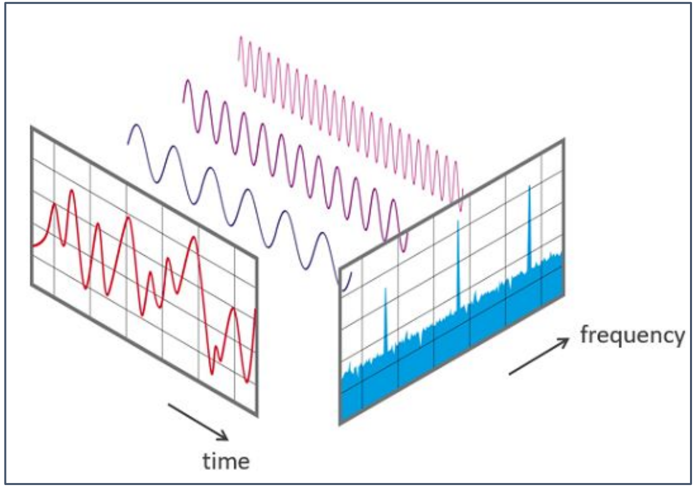
Spectrograms



[1] M. Muller, Fundamentals of Music Processing

Spectrograms are a representation of an audio signal’s power across time and frequencies.

Discrete Fourier Transform is applied to separate an audio signal into its constituent frequencies and visualized.

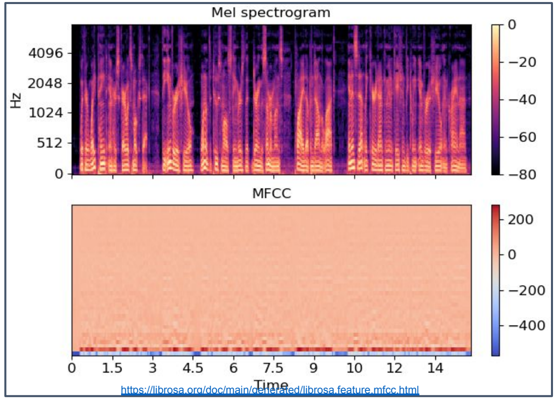


https://www.nti-audio.com/en/support/know-how/fast-fourier-transform-fft

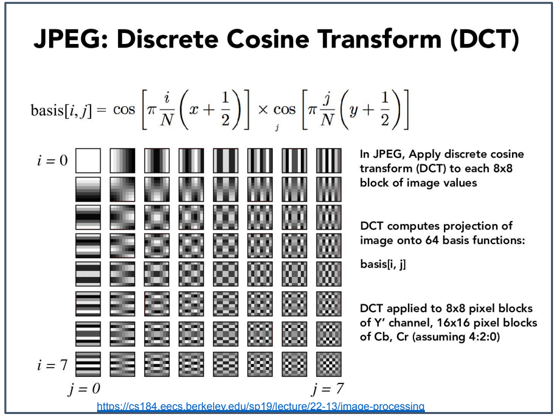
MFCCs (Mel-Frequency Cepstral Coefficients)

- A series of coefficients that summarize “textures” of the sound, derived from the spectrogram of the audio file

- Process
  - File split into short clips (ms)
  - Map spectrogram frequencies to Mel scale (human auditory perception scale) using Triangular bank filter
  - Log transform and apply discrete cosine transform (compression)



https://librosa.org/doc/main/generated/librosa.feature.mfcc.html



JPEG: Discrete Cosine Transform (DCT)

$$\text{basis}[i,j] = \cos\left[\pi \frac{i}{N}\left(x+\frac{1}{2}\right)\right] \times \underset{j}{\cos}\left[\pi \frac{j}{N}\left(y+\frac{1}{2}\right)\right]$$

In JPEG, Apply discrete cosine transform (DCT) to each 8x8 block of image values

DCT computes projection of image onto 64 basis functions:

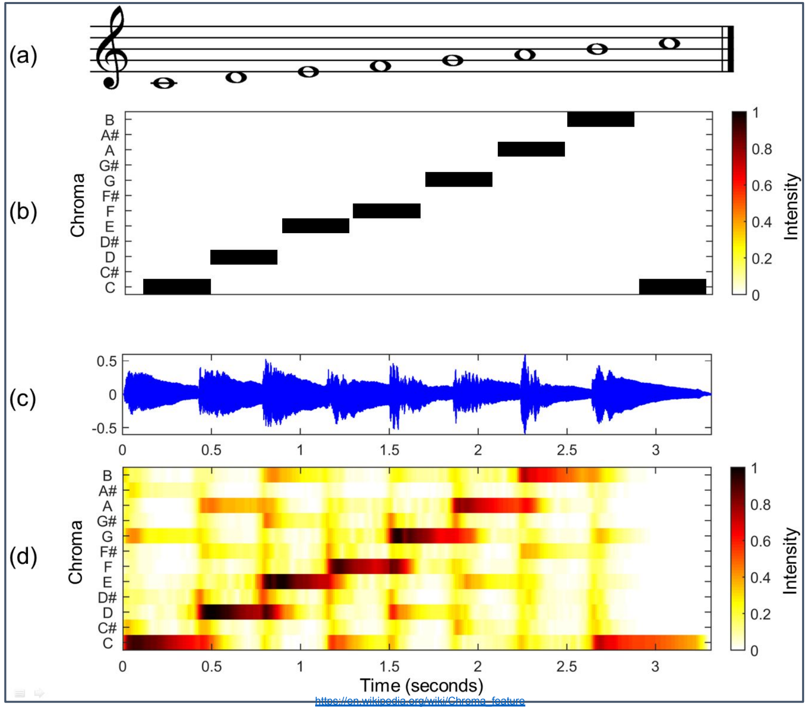
basis[i, j]

DCT applied to 8x8 pixel blocks of Y’ channel, 16x16 pixel blocks of Cb, Cr (assuming 4:2:0)

https://cs184.eecs.berkeley.edu/sp19/lecture/22-13/image-processing

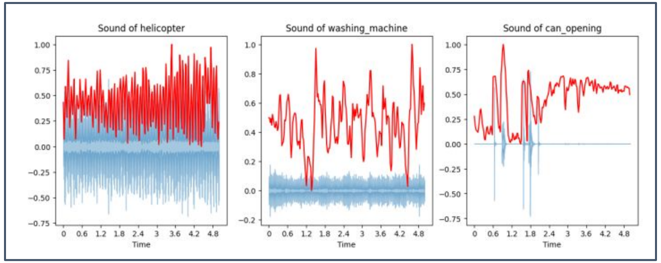
Chroma Features

- 12 coefficients capturing intensity of each note in an octave
  - C, C#, D, D#, E, F, F#, G, G#, A, A#, B
- Process
  - File split into short clips (ms) where one pitch is determined
  - Mapped to a spectrum
  - Mapped to the 12-variable chroma vector
  - Reveals common chords, patterns, tones



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

Spectral Centroid



[2] Gharbi, Rania. (2024)

- The brightness or darkness of the sound
  - The center of mass of the spectrum
  - Brighter audio = center of mass towards higher frequencies

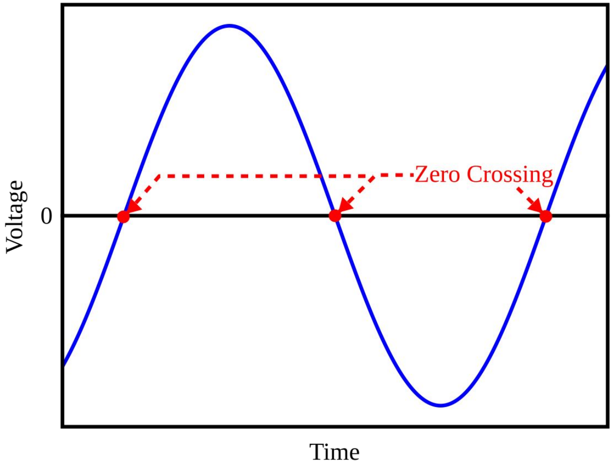
Spectral Bandwidth

- The variance of frequencies
  - Tightly packed frequencies = clean notes (such as a sine wave)
  - Wider bandwidth = distortion, noise, or more dynamic energy (metal music)

ZCR (Zero Crossing Rate)

- Rate of signal change
  - Indicates how frequently/quickly the signal crosses the 0 amplitude axis
  - Shows percussiveness / noisiness of the signal
  - Can help to detect rhythms

https://commons.wikimedia.org/w/index.php?curid=2994423



RMSE (Root Mean Square Energy)

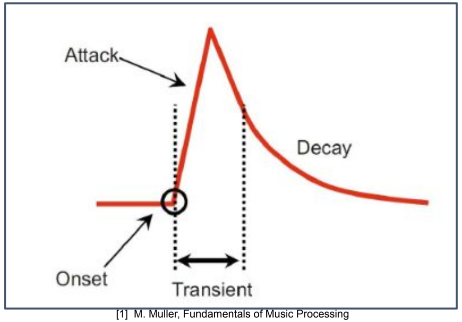
- The overall loudness of the audio
  - Measures the average power of the audio signal, or dynamic range

Spectral Rolloff

- The frequency below which 85% of the total energy exists
  - Estimates timbre, brightness
  - Turns spectral bandwidth to a discrete bins

Onset Envelope

- Time series used to detect the beat and rhythm
  - Analyzes raw waveform, broken into small frames
  - Tracks changes in energy



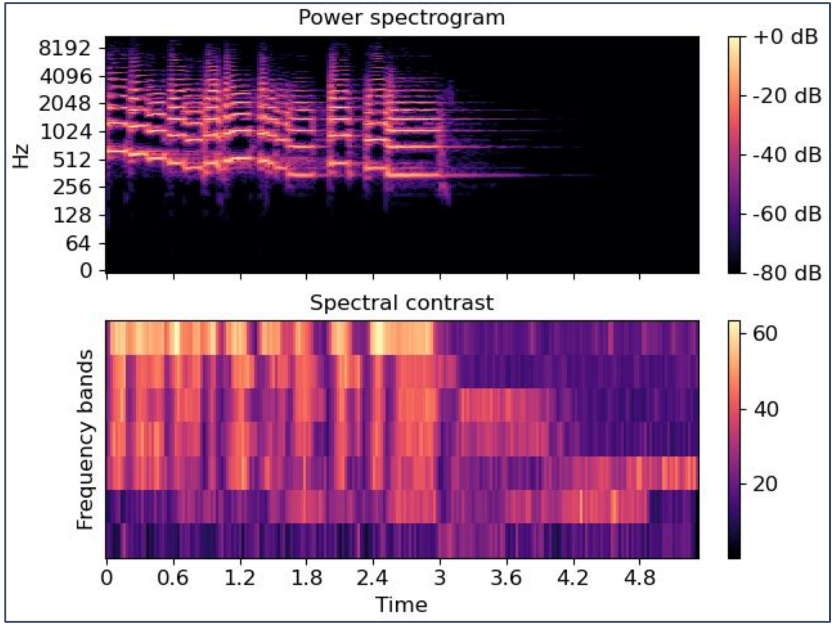
[1] M. Muller, Fundamentals of Music Processing

Tempo

- Beats per minute

Spectral Contrast

- 7 variables representing bands of frequency
  - In each band, differences between peaks and valleys of frequency measured
  - Texture analysis determining how dynamic or monotone the sound is

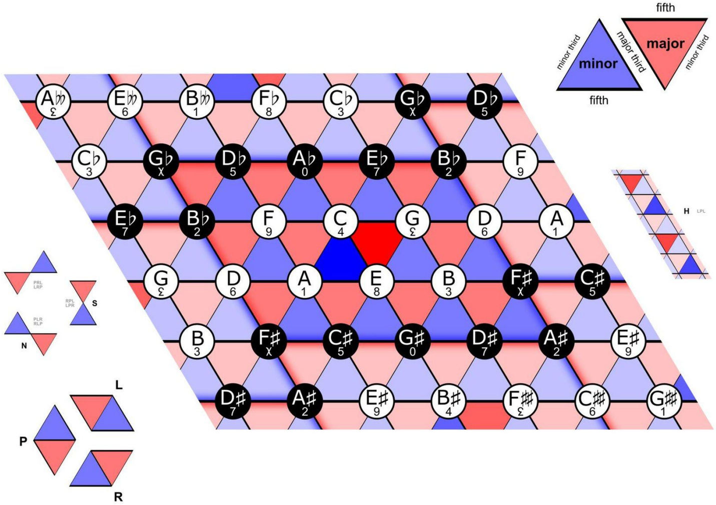


https://librosa.org/doc/latest/generated/librosa.feature.spectral_contrast.html

Tonnetz

- Harmonic relationships related to chord progressions, tonality, and other defining features like mood, song structure, dissonance

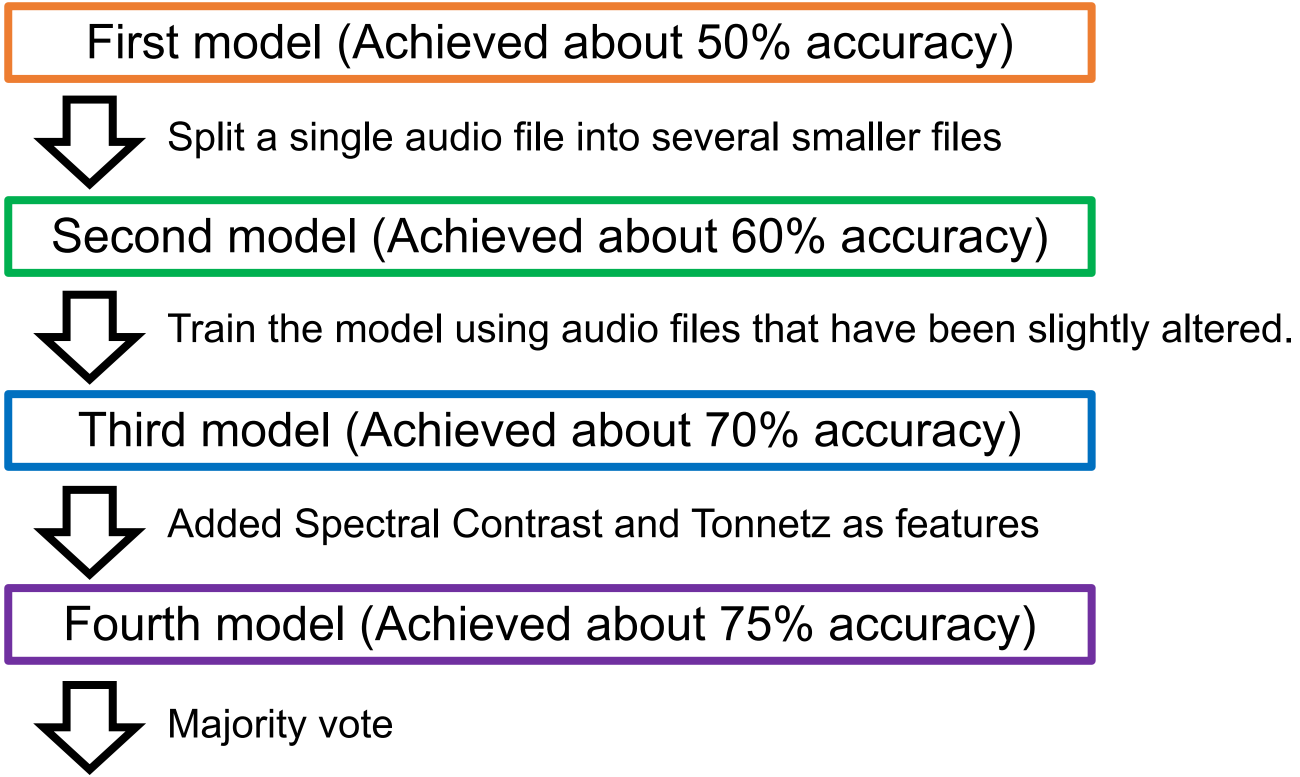
  - Given time-domain signal
  - Uses chroma vectors to map a 6-dimensional space
    - Movement by fifths (chords that are five notes apart)
    - Minor Thirds (darker steps)
    - Major Thirds (brighter steps)
    - Relative Major/Minor (between major and minor)
    - Parallel Major/Minor (between major and minor, but on the same note- i.e. C major to C minor)
    - Chromatic Motion (more complex and unpredictable movements)
- Used mean and standard deviation of each of the 6 dimensions in model



https://en.wikipedia.org/wiki/Tonnetz

Approach

Models tuned using GridSearchCV



First model (Achieved about 50% accuracy)

Split a single audio file into several smaller files

Second model (Achieved about 60% accuracy)

Train the model using audio files that have been slightly altered.

Third model (Achieved about 70% accuracy)

Added Spectral Contrast and Tonnetz as features

Fourth model (Achieved about 75% accuracy)

Majority vote

The accuracy exceeded 80%

Models

We mainly considered the following two models.

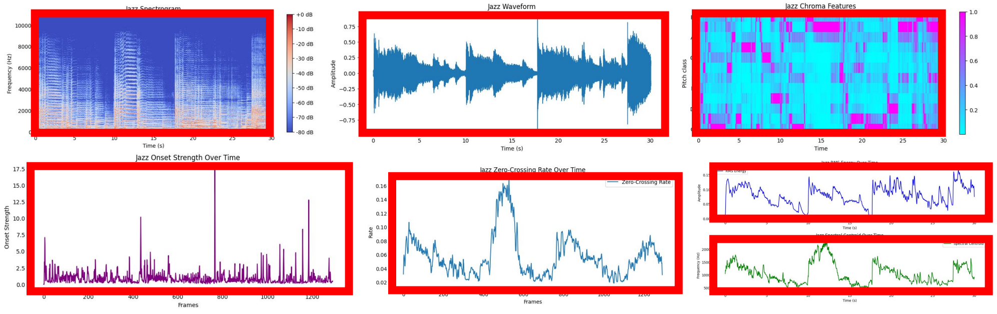
 SVM (Support Vector Machines)

- margin-based model
- Effective in high dimensions

 XGB (Extreme Gradient Boosting)

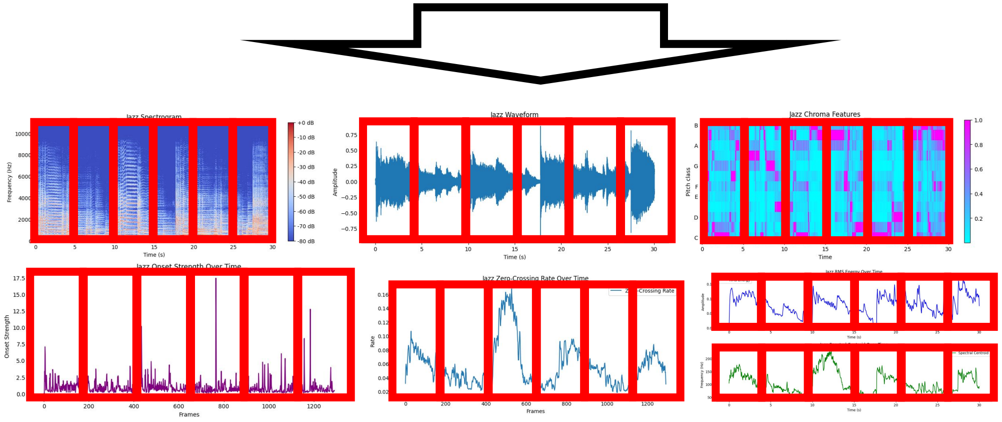
- Boosted ensemble model
- Multiple weak learners combine to form a strong learner

Audio Segmentation



Extract one set of features

Number of train samples: 800



Extract multiple feature sets

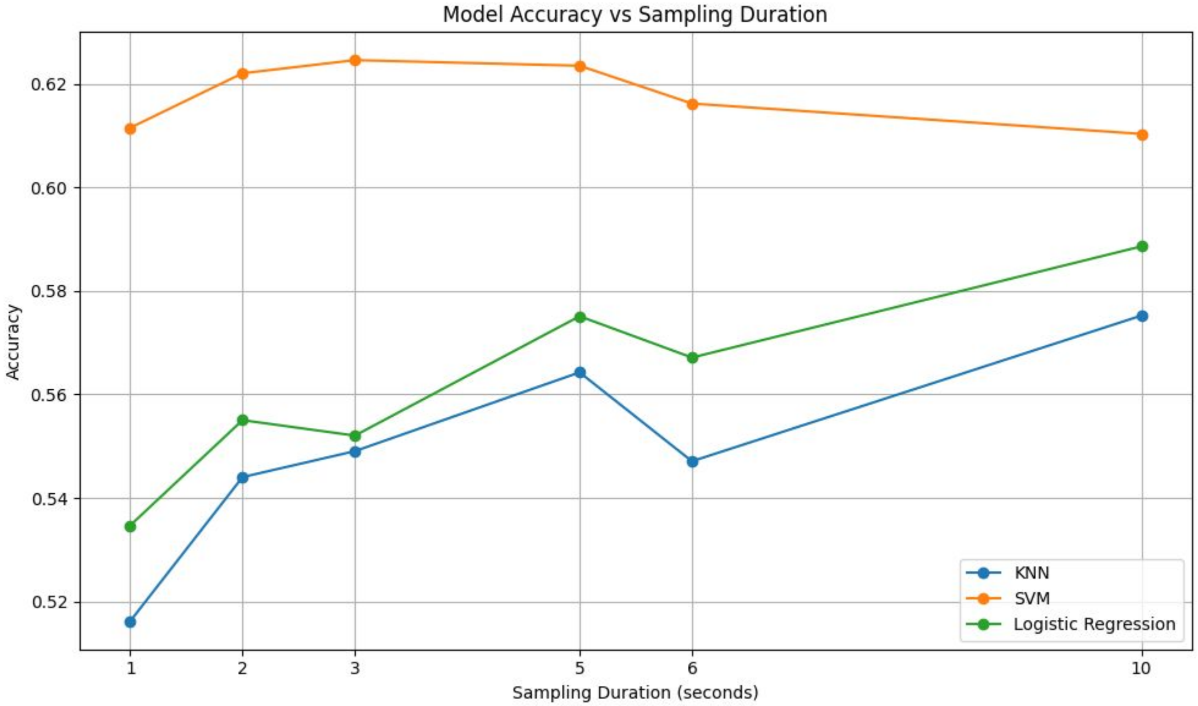
Number of train samples: Around 4,800
(If divided by 5 seconds)

Accuracy by Sampling Time

- Model performance varied with with smaller/larger sample sizes.

- SVM performed best with 3 second samples

- XGB (not pictured) performed best with 3 second samples



Data Augmentation

- Pitch shifting
  Changing the pitch of a sound. With librosa, you can shift the pitch without changing the speed. e.g.,

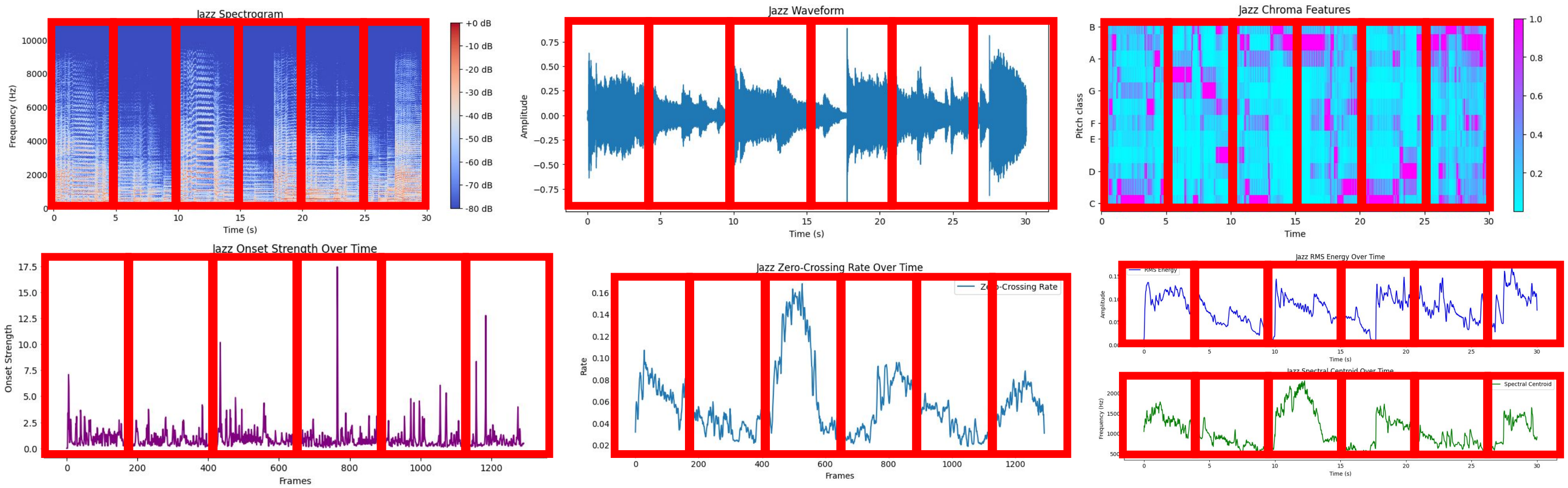
- Speed shifting
  Changing the playback speed of audio. With librosa, you can shift the speed without changing the pitch. e.g., speed +10%, -10%

Obtaining more training data without new audio files.

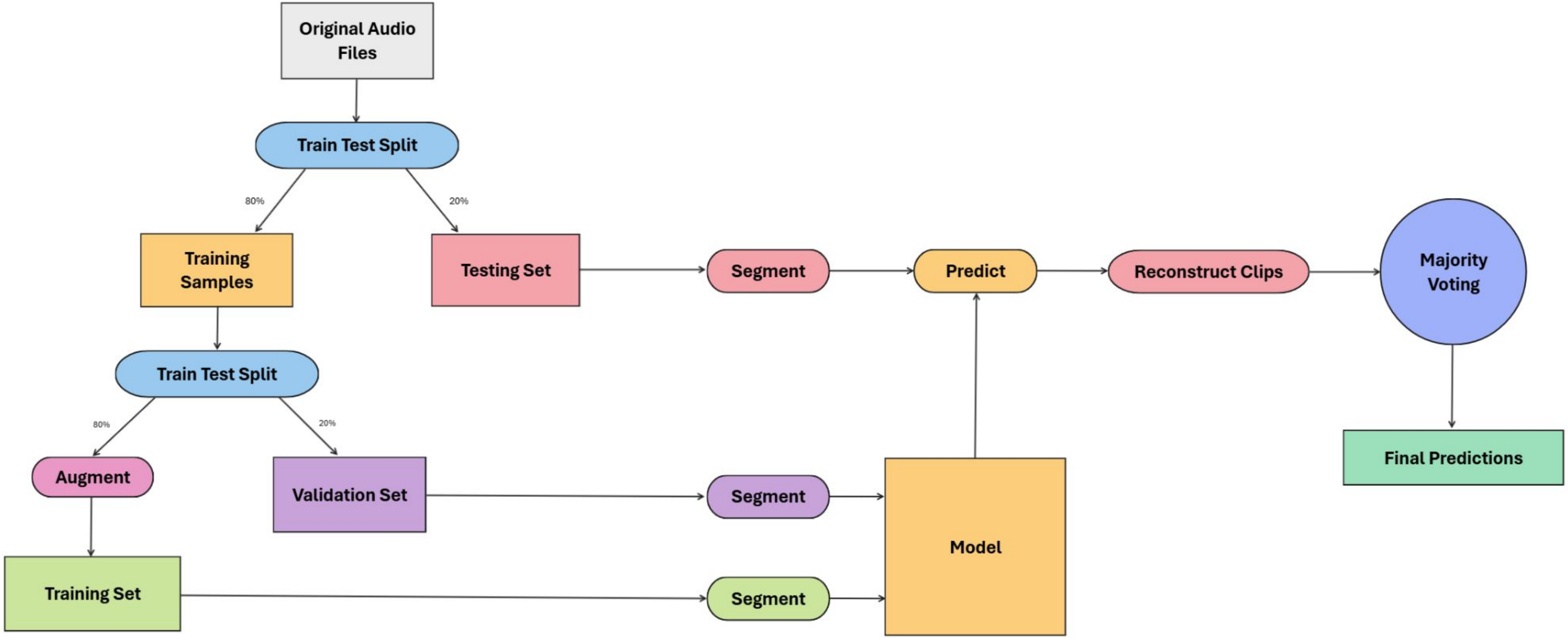
e.g., 4,800 samples → approx. 24,000 samples
(pitch +2, -2 and speed +10%, -10%)

Enabling the model to recognize the same genre when played back at different keys or speeds.

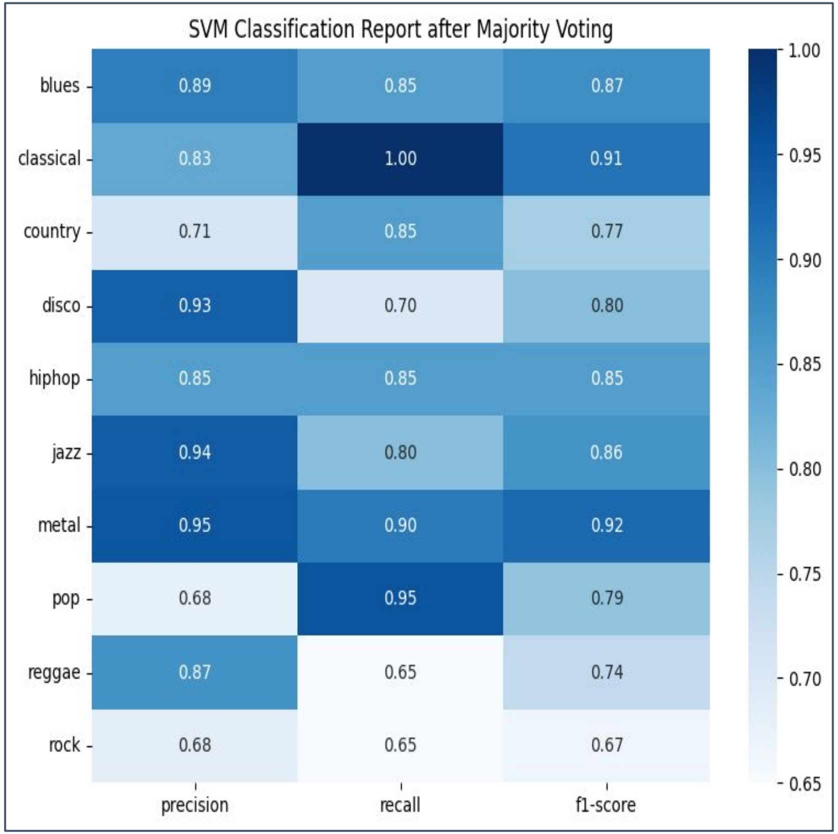
Stacked Majority Voting



Predict the genre of the original music file based on the majority vote of the prediction results for the divided samples.

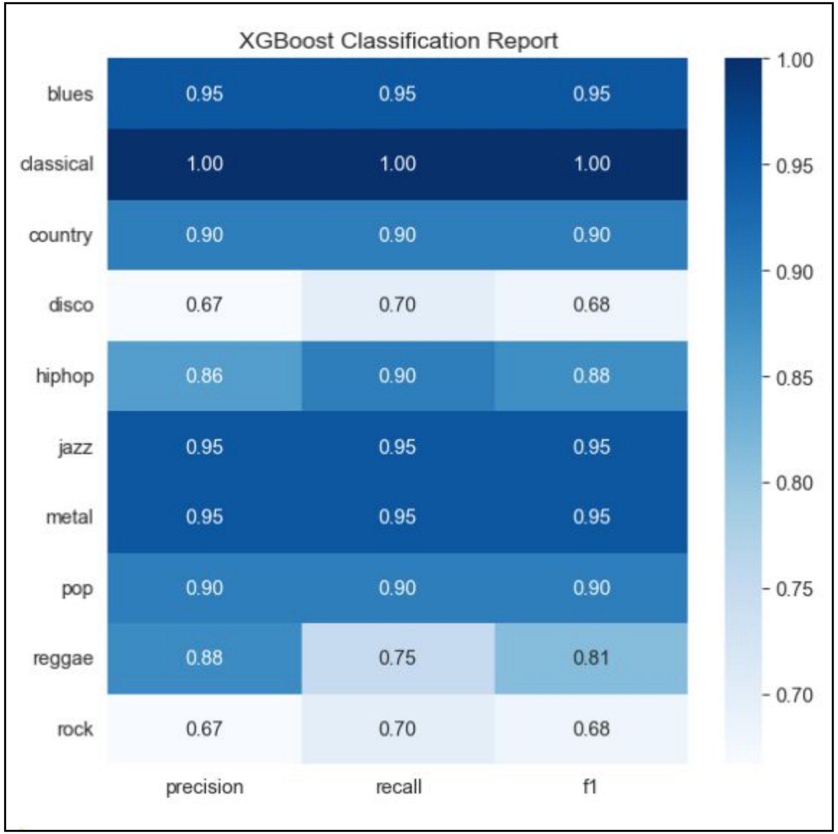

Original Audio Files
Train Test Split
80%
20%
Training Samples
Testing Set
Segment
Predict
Reconstruct Clips
Majority Voting
Train Test Split
80%
20%
Augment
Validation Set
Segment
Model
Final Predictions
Training Set
Segment

Final Results


SVM Classification Report after Majority Voting

	precision	recall	f1-score
blues	0.89	0.85	0.87
classical	0.83	1.00	0.91
country	0.71	0.85	0.77
disco	0.93	0.70	0.80
hiphop	0.85	0.85	0.85
jazz	0.94	0.80	0.86
metal	0.95	0.90	0.92
pop	0.68	0.95	0.79
reggae	0.87	0.65	0.74
rock	0.68	0.65	0.67

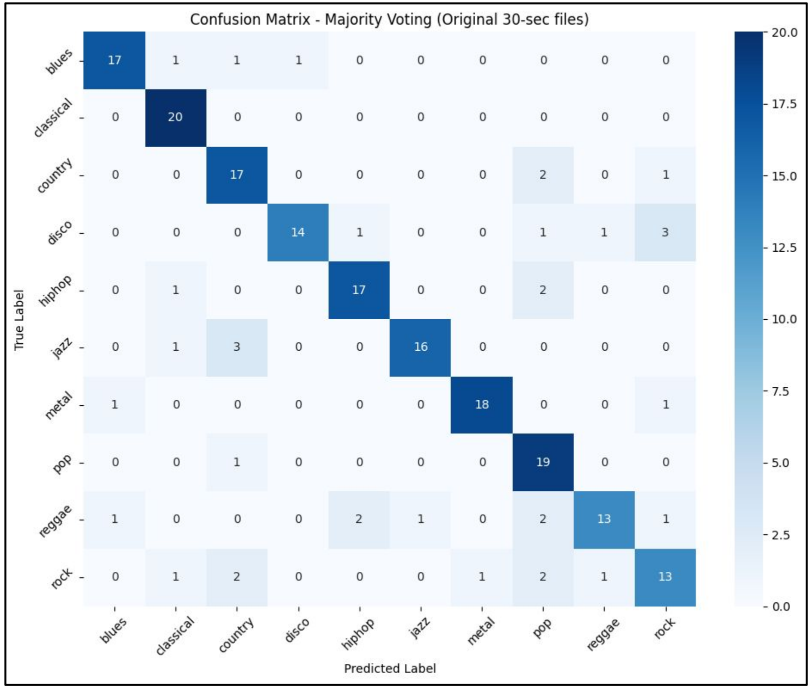
Accuracy: 0.82


XGBoost Classification Report

	precision	recall	f1
blues	0.95	0.95	0.95
classical	1.00	1.00	1.00
country	0.90	0.90	0.90
disco	0.67	0.70	0.68
hiphop	0.86	0.90	0.88
jazz	0.95	0.95	0.95
metal	0.95	0.95	0.95
pop	0.90	0.90	0.90
reggae	0.88	0.75	0.81
rock	0.67	0.70	0.68

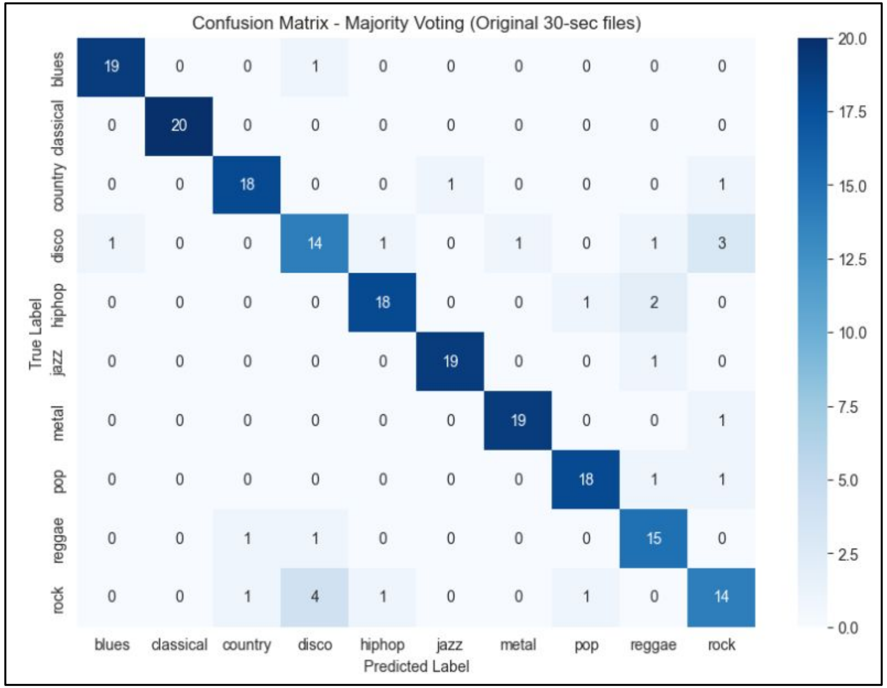
Accuracy: 0.87

Confusion Matrix


Confusion Matrix - Majority Voting (Original 30-sec files)

	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock
blues	17	1	1	1	0	0	0	0	0	0
classical	0	20	0	0	0	0	0	0	0	0
country	0	0	17	0	0	0	0	2	0	1
disco	0	0	0	14	1	0	0	1	1	3
hiphop	0	1	0	0	17	0	0	2	0	0
jazz	0	1	3	0	0	16	0	0	0	0
metal	1	0	0	0	0	0	18	0	0	1
pop	0	0	1	0	0	0	0	19	0	0
reggae	1	0	0	0	2	1	0	2	13	1
rock	0	1	2	0	0	0	1	2	1	13

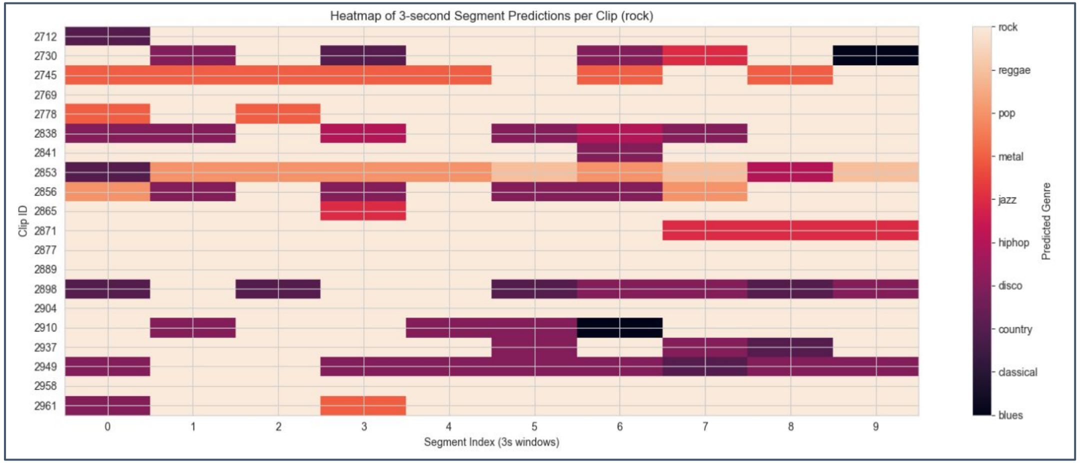
SVM

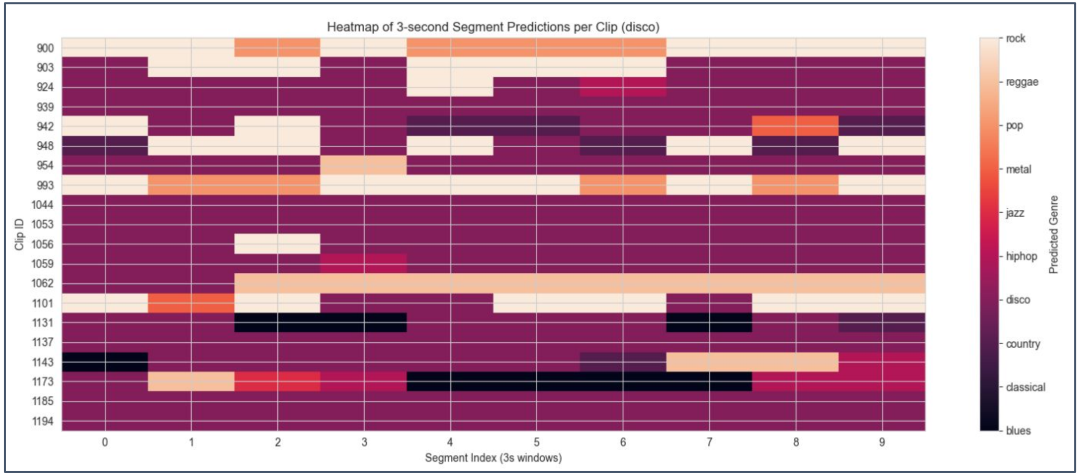

Confusion Matrix - Majority Voting (Original 30-sec files)

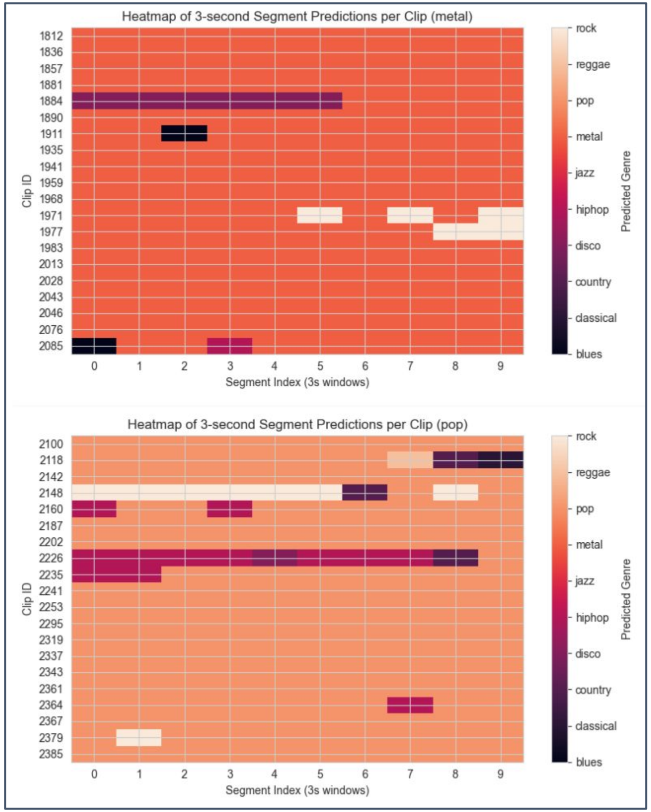
	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock
blues	19	0	0	1	0	0	0	0	0	0
classical	0	20	0	0	0	0	0	0	0	0
country	0	0	18	0	0	1	0	0	0	1
disco	1	0	0	14	1	0	1	0	1	3
hiphop	0	0	0	0	18	0	0	1	2	0
jazz	0	0	0	0	0	19	0	0	1	0
metal	0	0	0	0	0	0	19	0	0	1
pop	0	0	0	0	0	0	0	18	1	1
reggae	0	0	1	1	0	0	0	0	15	0
rock	0	0	1	4	1	0	0	1	0	14

XGBoost

Misclassification by Genre (XGBoost)







Summary

Results

- XGBoost model achieves 87% accuracy, beating all models on the PapersWithCode leaderboard.

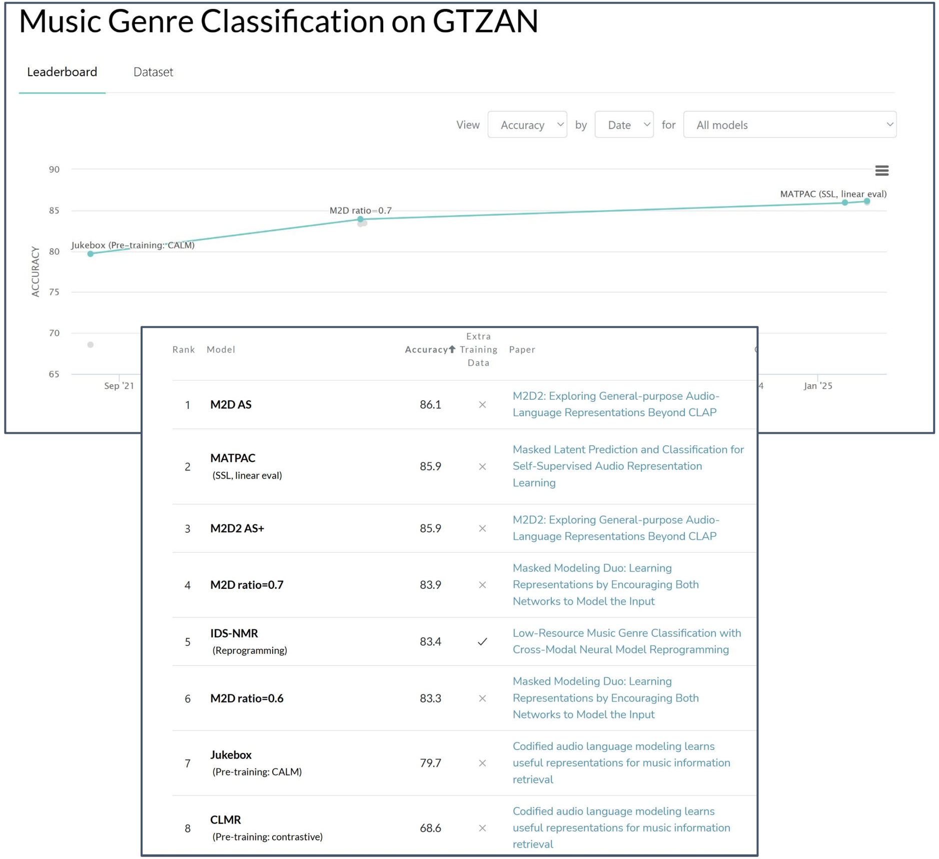
- SVM achieves 82% accuracy securing 7th position on the PapersWithCode leaderboard.

Future work

- Comparing performance on a better curated dataset.

- Introducing more features such as first order/second order derivatives of MFCCs.

- Feature engineering to better classify genres that are similar.



Music Genre Classification on GTZAN

View: Accuracy by: Date for: All models

Rank	Model	Accuracy	Extra Training Data	Paper
1	M2D AS	86.1	×	M2D2: Exploring General-purpose Audio-Language Representations Beyond CLAP
2	MATPAC (SSL, linear eval)	85.9	×	Masked Latent Prediction and Classification for Self-Supervised Audio Representation Learning
3	M2D2 AS+	85.9	×	M2D2: Exploring General-purpose Audio-Language Representations Beyond CLAP
4	M2D ratio=0.7	83.9	×	Masked Modeling Duo: Learning Representations by Encouraging Both Networks to Model the Input
5	IDS-NMR (Reprogramming)	83.4	✓	Low-Resource Music Genre Classification with Cross-Modal Neural Model Reprogramming
6	M2D ratio=0.6	83.3	×	Masked Modeling Duo: Learning Representations by Encouraging Both Networks to Model the Input
7	Jukebox (Pre-training: CALM)	79.7	×	Codified audio language modeling learns useful representations for music information retrieval
8	CLMR (Pre-training: contrastive)	68.6	×	Codified audio language modeling learns useful representations for music information retrieval

Sources

[1] M. Muller, Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications. Springer, 2015
[2] Gharbi, Rania. (2024). "A Study on Environmental Sound with Machine Learning and CNNs".