Politecnico di Milano
School of Industrial and Information Engineering
M.Sc. in High Performance Computing

Final project:
Classify musical genre using audio files

Prof. Edie Miglio
A.Y. 2024-2025

Authors:
Peng Rao (ID 270661)

Contents

1 Introduction

In today's fast-paced world, with millions of new songs released daily, organizing music into genres is essential for improving user experiences on streaming platforms and managing music libraries efficiently. However, manually classifying songs by genre is a time-consuming task. This is where **machine learning** offers a powerful solution.

2 Data Exploration

2.1 Data Description

The **GTZAN dataset** is the most-used public dataset for evaluation in machine listening research for music genre recognition (MGR). The files were collected in 2000-2001 from a variety of sources including personal CDs, radio, microphone recordings, in order to represent a variety of recording conditions.[1]

The dataset consists of 1000 audio tracks each 30 seconds long. It contains 10 genres, each represented by 100 tracks. The tracks are all 22050Hz Mono 16-bit audio files in .wav format. The features of the dataset can be summarized as follows table 1.

Category	Feature	Description
General	filename, length, label	File metadata and classification
Spectral	chroma_stft, rms, spectral_centroid, rolloff, etc.	Frequency-based characteristics
Harmonic	harmony, perceptr, tempo	Harmonic and perceptual content
MFCCs	mfcc1-20 (mean, var)	Captures timbral properties of audio

Table 1: GTZAN Dataset Features

2.2 Waveform Analysis

The waveform of an audio signal is a representation of the amplitude of the sound wave as a function of time. I visualized the waveform of a sample track in figure 1. The waveform figures show that some genres, like pop, metal, and hip-hop, exhibit high amplitude variations throughout the track. Others, like classical and jazz, show more varied amplitude patterns, reflecting their dynamic nature. Blues and reggae have distinct waveform structures that differentiate them from other genres.

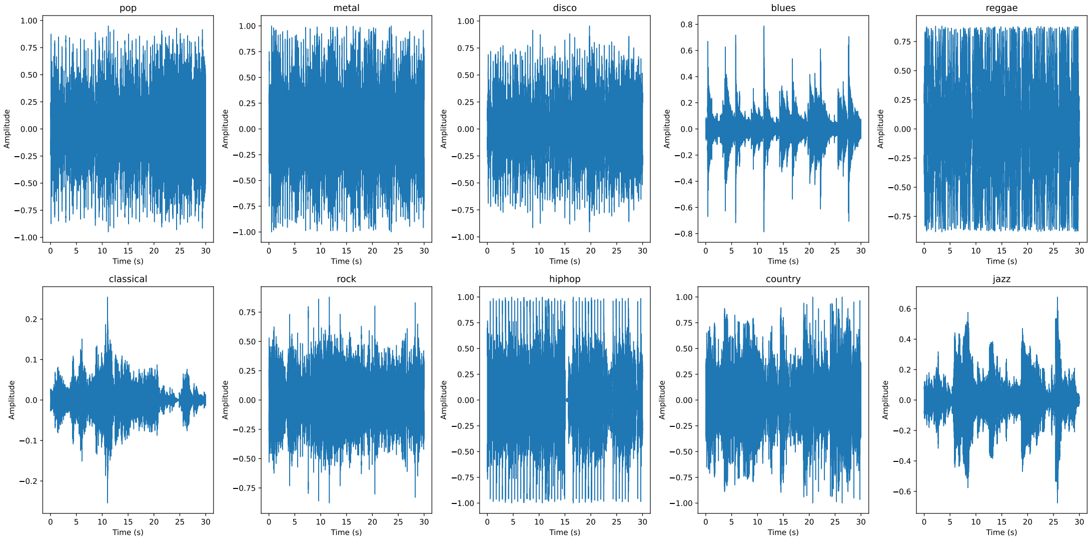


Figure 1: Waveform of various sample audios track over time

2.3 Chroma Analysis

Chroma features (or chroma vectors) are audio features used in music information retrieval to represent the harmonic and tonal content of an audio signal. They capture the distribution of energy across the 12 different pitch classes (e.g., C, C#, D, D#, ..., B) regardless of octave. I visualized the changes in chroma features over time for a sample track in figure 2. Each row corresponds to one pitch class, showing how much energy is present in that pitch class over time. Genres like rock, blues, pop, and hip-hop show dense chroma activity, indicating frequent chord changes. Classical and jazz have more structured patterns (jazz is showing more harmonic complexity).

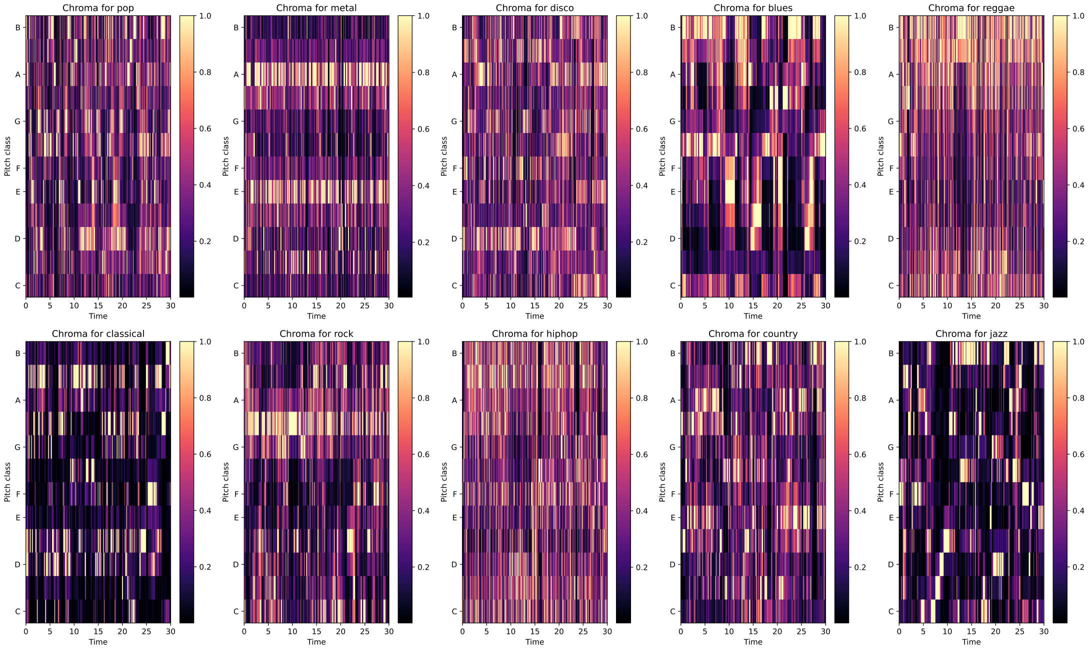


Figure 2: Chroma Features of various pitch classes over time

2.4 MFCC Analysis

MFCC is a feature extraction technique commonly used in speech and audio processing. It transforms a raw audio signal into a compact, perceptually relevant representation by mimicking the human auditory system. Typically, the first 13-20 coefficients are used as features. I visualized the MFCCs of a sample track in figure 3. The MFCCs capture the timbral properties of the audio signal, showing how the spectral content changes over time. Genres like rock, blues, and pop exhibit high variability in MFCCs, reflecting their diverse timbral characteristics. Classical and jazz have more structured patterns, indicating more consistent timbral properties.

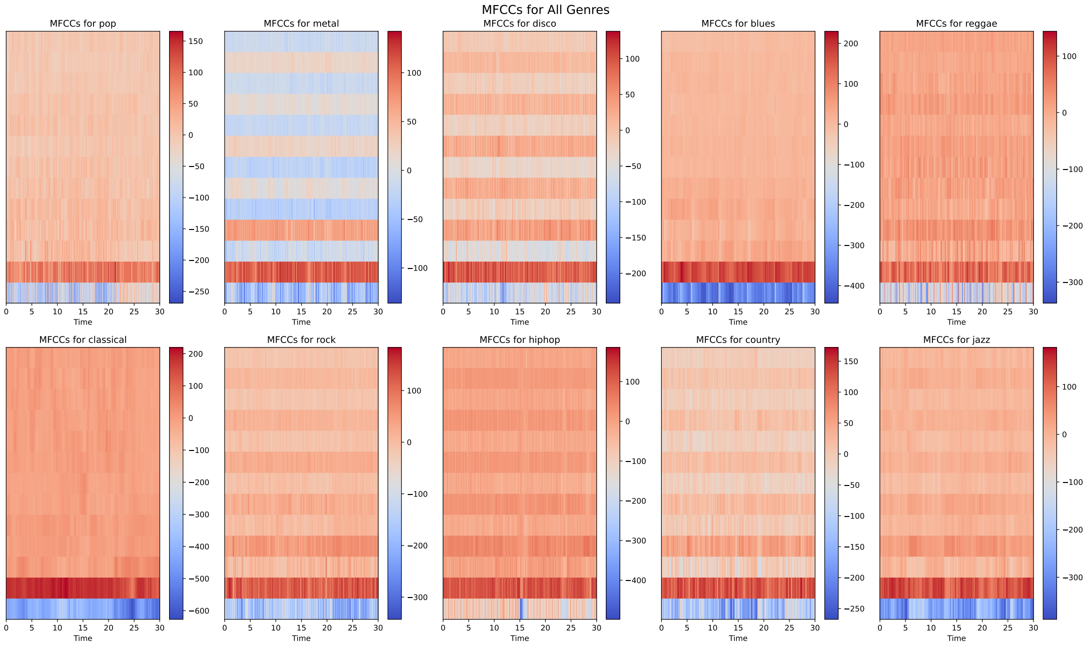


Figure 3: MFCC13 of various sample audios track over time

2.5 Principal Component Analysis

Principal Component Analysis (PCA) is a dimensionality reduction technique that transforms data into a lower-dimensional space while preserving as much variance as possible. I applied PCA to the dataset and visualized the first two principal components in figure 4.

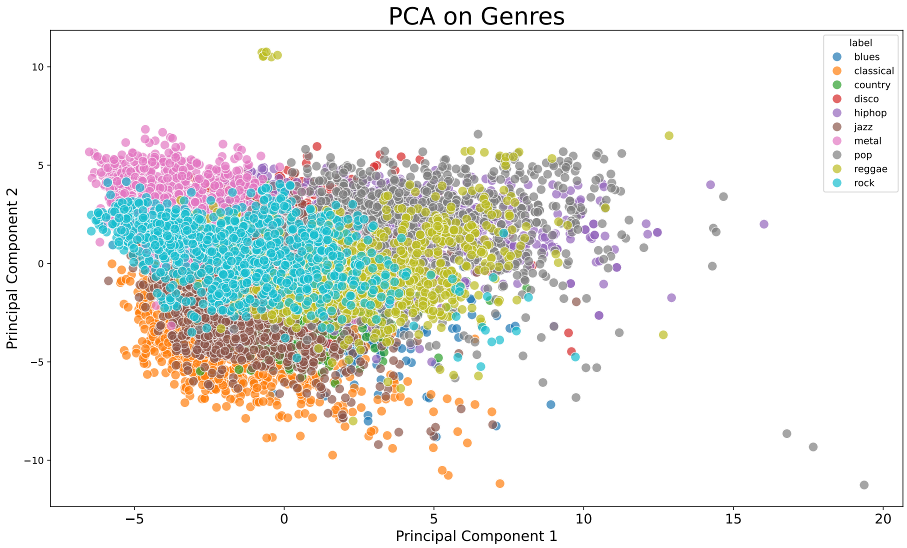


Figure 4: PCA of the GTZAN dataset

b

3 Data Preprocessing

3.1 Normalization

I normalized the dataset to ensure that all features have a similar scale. I used the Min-Max scaling technique to scale the features:

$$X_{\text{scaled}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{1}$$

3.2 Train/Test Split

I split the dataset into training and testing sets using an 80/20 split. I used the training set to train the model and the testing set to evaluate its performance. The validation has been used during the training phase of the classifers to find good values of the hyperparameters.

3.3 Cross-Validation Split

I used a 5-fold cross-validation split to evaluate the model's performance. Cross-validation is a technique used to assess the model's generalization performance by training and testing the model on different subsets of the data, as shown in figure 5.

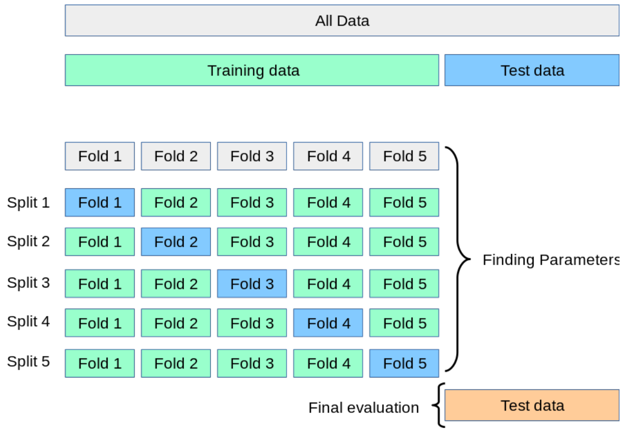


Figure 5: 5-fold Cross-Validation Split[2]

4 Traditional Machine Learning Models

4.1 Baseline Models

I trained several traditional machine learning models on the dataset to establish a baseline performance. The models I used are:

- **Logistic Regression**
- **Support Vector Machine**
- **Decision Tree**
- **Random Forest**
- **XGBoost**

The models were trained using the default hyperparameters and evaluated using cross-validation. The results of training set are summarized in figure 6.

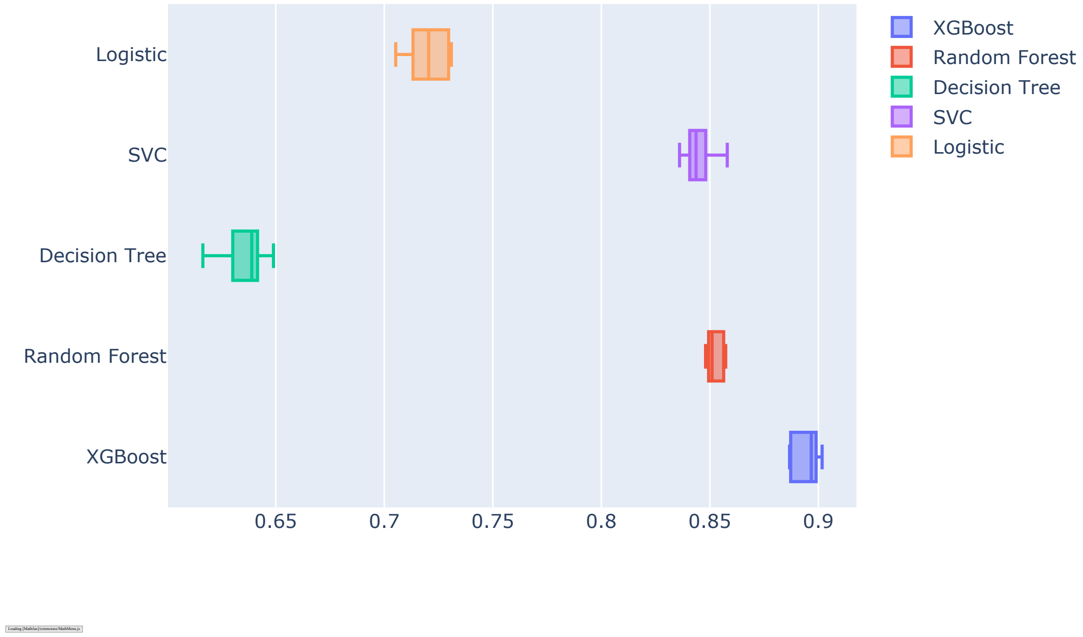
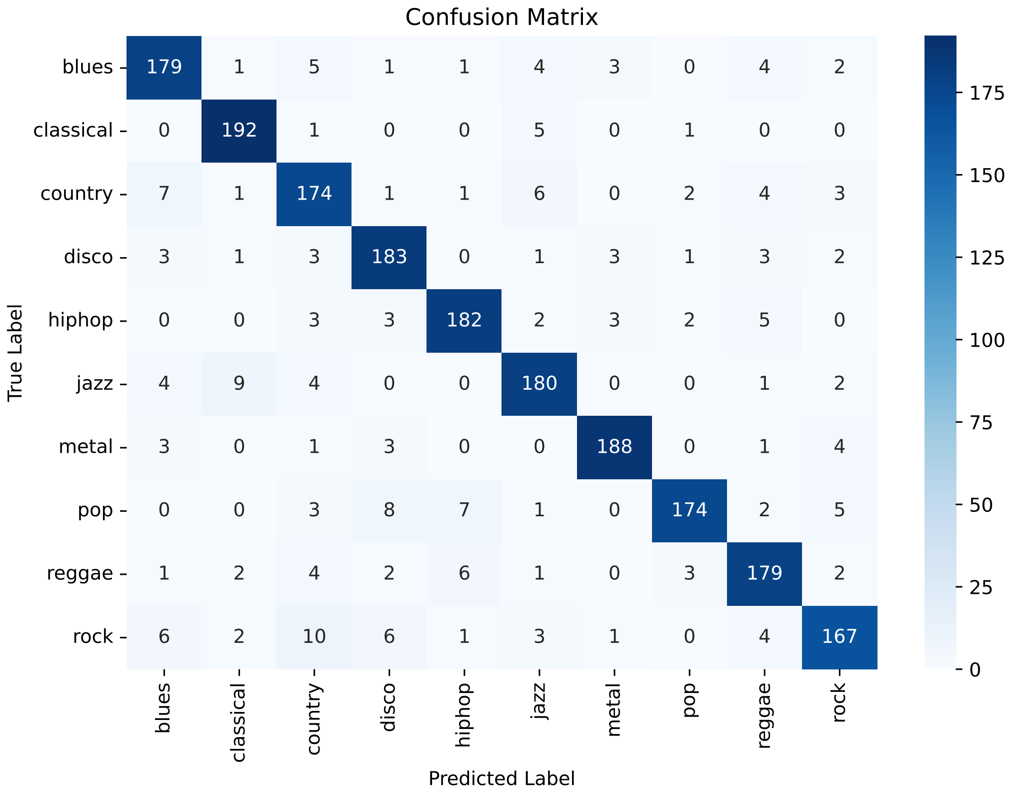


Figure 6: Baseline Models Performance on training set

4.2 Model Evaluation

I evaluated the Xgboost model on the testing set and obtained an accuracy of 0.75. The confusion matrix shows that the model performs well for some genres (e.g., classical, jazz) but struggles with others (e.g., metal, hip-hop). The model's precision, recall, and F1-score are summarized in figure 7.



True Label \ Predicted Label	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock
blues	179	1	5	1	1	4	3	0	4	2
classical	0	192	1	0	0	5	0	1	0	0
country	7	1	174	1	1	6	0	2	4	3
disco	3	1	3	183	0	1	3	1	3	2
hiphop	0	0	3	3	182	2	3	2	5	0
jazz	4	9	4	0	0	180	0	0	1	2
metal	3	0	1	3	0	0	188	0	1	4
pop	0	0	3	8	7	1	0	174	2	5
reggae	1	2	4	2	6	1	0	3	179	2
rock	6	2	10	6	1	3	1	0	4	167

Figure 7: XGBoost Model Evaluation on testing set

4.3 Hyperparameter Tuning

I used RandomizedSearchCV to tune the hyperparameters of the XGBoost model. I performed a randomized search over a predefined hyperparameter grid and selected the best hyperparameters based on the model's performance. The hyperparameters I tuned were:

- **max_depth**: Maximum depth of the tree
- **learning_rate**: Step size shrinkage used in update to prevent overfitting
- **n_estimators**: Number of boosting rounds
- **subsample**: Subsample ratio of the training instances
- **colsample_bytree**: Subsample ratio of columns when constructing each tree

The best hyperparameters found by RandomizedSearchCV were:

- **max_depth**: 6
- **learning_rate**: 0.1
- **n_estimators**: 100
- **subsample**: 0.8
- **colsample_bytree**: 0.8
- **objective**: 'multi:softmax'

5 Convolutional Neural Network (CNN)

6 Conclusion

References

[1] *GTZAN Dataset.* URL: https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification.

[2] *Cross-validation: Evaluating Estimator Performance.* scikit-learn. URL: https://scikit-learn.org/stable/modules/cross_validation.html.