Chromesthesia Chord Engine

Course: Data Science Final Project

Date: June 11, 2025

Team Members:

1. 나우팔 카마루딘 **(2024320084)**
2. 무함마드 이즈미르 빈 수하이미 **(2022320113)**
3. 아흐마드 나우팔 **(2023320085)**

Introduction / Problem Statement

Chromesthesia in a nutshell is a condition where some people could perceive colors when they hear music or sound. These sound-color associations are unique to each individual, for instance, one person might "see" the notes of C major as blue and A minor as red. Some of the famous musicians like Franz Liszt, Duke Ellington, and Billie Eilish have this condition, with Billie even mentioned that she could literally see colors when listening to music.

The Chromesthesia Chord Engine we have created aims to simulate the senses that these people experience. Using machine learning techniques, we could analyze audio input, identify the chords that are being played, and generate corresponding visual outputs, such as colors and shapes which translate to different chords. Our goal is to let users "see" the music in real time, even if they don't naturally have synesthesia.

This tool could help both creatively and educationally as it can help novice musicians grasp chord progressions and offer a new way to experience music visually. Major chords are associated with brightness and happiness, and they are mapped to warm colors and soft shapes, while minor chords which are perceived as sad are translated to cool colors and angular shapes. This project blends music, perception, and machine learning into a unique audiovisual experience for all of us to experience.

Dataset

We could not find any public "chords to colors" dataset, therefore we created our own by generating synthetic audio samples of musical chords. Each sample is a short clip of a few seconds containing a single chord, labeled manually such as C Major or A Minor. We focused on the common major and minor triads across the 12-note chromatic scale, resulting in dozens of chord classes.

Key Dataset Details:

- **Chords Covered:** Major and minor chords for all root notes (C to B), providing a wide range of targets for classification tasks.
- **Sample Generation:** We used pure sine waves to synthesize each chord's notes (e.g., C, E, G for C Major) and combined them into audio clips using the librosa library. Differences in octave and timbre also added more diversity to the dataset.
- **Feature Extraction:** Rather than using raw audio, we have extracted chroma features which are 12-dimensional vectors representing energy in each pitch class. For example, an F Major chord would show high values at F, A, and C. These chroma vectors became the input X, while the corresponding chord names were used as the labels y.
- **Train/Test Split:** The dataset was split (e.g., 80/20) into training and testing to evaluate the model's generalization capabilities. Because we've used multiple samples per chord, this ensures that the model learns patterns instead of memorizing individual recordings.

The color and shape mappings were not part of the dataset or training labels. They were applied later using a manual lookup table based on the predicted chord, as described later in the Methodology part of our report.

Methodology

Using this implementation we are able to identify chords from audio using machine learning, as well as mapping them to specific colors and shapes for visualization. The workflow comprises of five main steps:

1. Audio Synthesis & Data Collection

The way we generate our dataset is by synthesizing chord audio samples using sine waves created with the librosa library for each note. This gave us clean, noise-free and labeled data, as well as giving us full control over chord types and audio consistency.

2. Feature Extraction (Chroma Representation)

Each audio sample was converted into a chroma feature vector which is a 12-dimensional representation showing energy in each pitch class from C to B. This compact representation captures harmonic content and is not affected by the change in octave, making it ideal for chord recognition. We've also normalized these features to minimize the impact of low and high sound volumes. Chroma features are extracted using these three methods:

- Short-Time Fourier Transform (chroma_stft)
- Constant-Q Transform (chroma_cqt)
- Chroma Energy Normalized Statistics (chroma_cens)

3. Model Selection and Training

We have used **Random Forest Classifier** since it:

- Could achieve strong results on small datasets without experiencing overfitting
- Can be used for multi-class classification
- Is easy to interpret and have fast training time using scikit-learn

The final model used 100 trees with controlled depth to avoid overfitting.

4. Model Evaluation

We have evaluated the classifier on an unseen dataset to avoid any bias in the results, and were able to achieve an accuracy of around 97% with the model experiencing only a minor confusion between chords that share notes (e.g., C major and A minor). Accuracy, precision, recall, and a confusion matrix are also used to provide a stronger and a more consistent evaluation performance across all chord classes.

5. Chord-to-Color Mapping (Visualization)

After predicting a chord, the system maps it to a predefined color and shape based on emotional tone:

- **Major chords** → warm colors (e.g., yellow, orange) + circles
- **Minor chords** → cool colors (e.g., blue, purple) + triangles

These visual assignments were inspired by common synesthetic representations and are kept consistent through a lookup table. The model does not learn or modify the visuals, it only predicts the chord, and the mapping handles the display. Furthermore, we used matplotlib to draw visuals in sync with the music, creating a dynamic, real-time visualizer.

Results & Evaluation

The Random Forest model achieved approximately 97% accuracy on the test set, demonstrating strong performance in identifying chords from audio. This is expected, given the clean, synthetic dataset and the effectiveness of chroma features in capturing harmonic content.

A confusion matrix confirmed that most predictions were correct, with only minor confusion between harmonically similar chords such as A minor vs. C major, or E minor vs. G major due to overlapping notes. However, unrelated chords (e.g., D major vs. F# minor) could be classified without much of a problem. Precision and recall value exceeded 0.95 across all classes, indicating a reliable performance and little to no misclassifications.

To test the real-world usability, we have analyzed a piano sound clip featuring the common C → G → Am → F chord progression. The model could accurately predict most chords in real-time, and the visualizer displayed synchronized shapes and colors. Some minor errors occurred during arpeggios or dense harmonics, but the overall output effectively reflected the musical mood.

Performance-wise, the system is lightweight and could easily be used in real-time. Chroma extraction and chord prediction could run instantly, enabling smooth live visualization.

Discussion

Our results confirm that the model could effectively learn recognizing chords from chroma features. It identifies key note patterns, such as detecting F, A, and C as "F Major" similar to how a musician would. Feature importance analysis showed that certain notes like B were helpful in distinguishing between similar chords, aligning with music theory.

The visual mapping created an intuitive representation of emotions in the music. So, even without knowing the song, viewers could grasp emotional shifts just by observing the visuals.

Limitations

- **Real-world generalization**: The model was trained on clean, isolated chords. In real music, overlapping notes, inversions, or instrument noise can confuse the classifier. However, additional training on diverse and noisy data would help a long way in improving the model's generalization performance.
- **Subjective mapping**: Color-shape assignments are not universal. Synesthetic experiences vary between the individuals with synesthesia, and the choices that we made for the visual representations are based on general emotion-tone logic. In the future, allowing user customization could definitely improve personalization.
- **Chord variety**: We have only modeled major and minor triads. Real music includes many other types (7ths, diminished, etc) that weren't accounted for. A system that could classify more complex chords will be an interesting approach in the future.
- **Simple visuals**: The current visual output is static. While clear, it's limited. Adding animations like pulsating effects, spinning triangles, or smoother transitions could better reflect musical movement and enhance the chromesthetic experience.

Despite these limitations, this project showcases how a simple ML pipeline with musically informed features can produce meaningful, engaging results.

Conclusion

We have successfully developed the Chromesthesia Chord Engine, a system that analyzes musical chords and visualizes them as color-shape combinations, simulating a chromesthetic experience through machine learning. By combining chroma-based feature extraction, Random Forest classifier, and a visualizer, our model accurately identifies chords from synthetic audio and provides an interesting visual experience that shows how people with synesthesia ‘sees’ music.

The project successfully integrated several core data science concepts such as feature engineering, classification, evaluation, and creative output. It demonstrated how a subjective idea like “hearing colors” can be captured with structured design and ML tools.

Future Improvements

- **Expanded Chord Coverage**: Include 7ths, diminished, and more complex chords to improve recognition accuracy in music.
- **Enhanced Visuals**: Add animations or effects and allow user-customized mappings or feedback-based learning.
- **Live Instrument Integration**: Build an interactive version for real-time use with MIDI instruments or microphones.
- **Multisensory Extension**: Explore other types of chromesthesia conditions beyond visuals, such as vibration or scent, to further emulate synesthetic perception.

Ultimately, this project captures the essence of creative machine learning which uses simple tools to produce expressive, intuitive results. It’s a foundation for future exploration into how data science can bridge sensory boundaries and how it could be somewhat useful for future learning purposes.

Appendix

Project Files

All code, model files, and datasets are included in the submitted GitHub/Google Drive folder, along with setup instructions and dependencies (librosa, scikit-learn, etc.).

Chord-to-Color Mapping

Chord	Assigned Color	Assigned Shape	Rationale (Emotion)
C Major	Bright Yellow	Circle	Happy, “sunny” resolution
A Minor	Deep Blue	Triangle	Sad or mellow
G Major	Orange-Red	Circle	Lively, warm
E Minor	Purple (violet hue)	Triangle	Somber but rich
D Major	Vibrant Green	Circle	Triumphant, fresh

Table 1: *Sample of the chord-to-visual mapping used in the Chromesthesia Chord Engine. Major chords were mapped to brighter, warmer colors and smooth circular shapes, while minor chords were mapped to cooler colors and angular shapes (triangles). This design was chosen to reflect the general emotional quality of the chords.*

Code Snippet – Chroma Feature Extraction: The following snippet (from our Jupyter Notebook) demonstrates how we convert an audio signal into chroma features using librosa:

```
import librosa

import numpy as np


# Load an audio waveform (y) for a chord, with a given sampling rate sr

y, sr = librosa.load("C_major_chord.wav", sr=22050)

# Compute the chromagram (12 x N matrix of 12 pitches over N time frames)

chroma = librosa.feature.chroma_stft(y=y, sr=sr)

# Take the mean chroma vector across time frames to represent the chord

chroma_vector = np.mean(chroma, axis=1)

print("Chroma feature for C major:", chroma_vector)
```

This produces a 12-dimensional vector like [0.10, 0.01, 0.00, 0.85, ...] (example values), where indices 0-11 correspond to pitch classes C, C#, D, …, B. For a C major chord, the chroma vector has high values at C, E, and G positions, verifying that the feature captures the chord's identity. These vectors are then fed into the Random Forest for training and prediction.

Reference

The Synesthesia Tree. (2021, February). *Chord-colour*. Retrieved June 9, 2025, from https://www.thesynesthesiatree.com/2021/02/chord-colour.html

The London Piano Institute. (n.d.). *Synesthesia and piano: How some musicians see colours when they play*. Retrieved June 9, 2025, from https://www.londonpianoinstitute.co.uk/synesthesia-and-piano/