

Product Requirements Document

- Music Analyzer App

1. Introduction

- **1.1 Purpose:** This document outlines the vision, features, and requirements for the Music Analyzer App. It serves as a guide for development, design, and testing, ensuring the product meets user needs and project goals.
- **1.2 Goals:**
 - Provide users with an easy-to-use tool for identifying the genre of music from live audio or uploaded files.
 - Offer reliable genre classification based on established audio analysis techniques and pre-trained models.
 - Present analysis results in a clear, understandable web-based interface.
 - Enable users to listen back to the audio segment that was analyzed.
 - Create an engaging and informative experience for users interested in music characteristics.
- **1.3 Scope:** The current version of the app focuses on analyzing up to 60 seconds of audio input (microphone or file upload) to determine its music genre using a Convolutional Neural Network (CNN) pre-trained on the GTZAN dataset. It displays the predicted genre and allows audio playback. Future scope could include confidence scores, more detailed analysis visualizations (rhythm, melody, instrumentation), and integration of alternative classification models.

2. User Personas

- **2.1 Music Enthusiast (Alex):** Alex loves listening to various genres and is curious about identifying songs playing on the radio or in cafes. Alex wants a quick way to get a genre label for unfamiliar music using their phone or laptop microphone.

- **2.2 Music Student (Sam):** Sam is studying music theory and production. Sam uses the app to analyze reference tracks or their own compositions to understand genre characteristics. Uploading files is important for Sam's workflow.
- **2.3 Casual User (Jamie):** Jamie occasionally comes across interesting sounds or music and wants a simple tool to satisfy their curiosity about the genre without needing deep technical knowledge.

3. Use Cases / User Stories

- **UC-01: Analyze Live Audio:** As a user, I want to analyze audio directly from my microphone so that I can quickly get the genre of music playing around me.
- **UC-02: Analyze Uploaded Audio File:** As a user, I want to upload an audio file for analysis so that I can determine the genre of a music track I have saved.
- **UC-03: View Genre Result:** As a user, after an analysis is complete, I want to clearly see the predicted genre so that I understand the application's findings.
- **UC-04: Play Analyzed Audio:** As a user, after an analysis is complete, I want to play back the audio that was analyzed so that I can hear the segment the results correspond to.

4. Features

- **4.1 Real-time Audio Recording:**
 - Allow users to start/stop recording audio via microphone input through the browser.
 - Handle browser permissions for microphone access.
 - Provide visual feedback during recording.
- **4.2 Audio File Upload:**
 - Allow users to upload audio files via a file selection dialog or drag-and-drop.
 - Support common audio formats (WAV, MP3, OGG, FLAC, M4A).
 - Provide feedback on upload status and supported formats.
- **4.3 Audio Analysis Core:**
 - Process up to 60 seconds of the provided audio (recorded or uploaded).

- Utilize the GenreClassifier loaded with gtzan_model.pt pre-trained weights.
- Extract necessary features (e.g., mel-spectrograms) for the model.
- Predict the music genre from the 10 supported GTZAN categories (Blues, Classical, Country, Disco, Hip Hop, Jazz, Metal, Pop, Reggae, Rock).
- **4.4 Results Display:**
 - Clearly display the single predicted genre name in the UI upon analysis completion.
 - (Future Enhancement: Display confidence score).
- **4.5 Audio Playback:**
 - Provide a "Play Audio" button enabled after successful analysis.
 - Use an embedded HTML5 audio player to play back the analyzed 60-second (or shorter, if recording stopped early) segment.
- **4.6 User Interface:**
 - Web-based interface built with Dash/Flask and Bootstrap.
 - Logical layout separating input controls, results display, and playback.
 - Visual indicators for different states (recording, uploading, analyzing).

5. Design & UX Considerations

- **Simplicity:** The primary user flow (record/upload -> analyze -> view result) should be straightforward.
- **Feedback:** Provide clear visual cues for ongoing processes (recording, uploading, analyzing) and errors (mic denied, wrong file type).
- **Responsiveness:** The interface should adapt reasonably well to different screen sizes (desktop focus primarily).
- **Clarity:** Analysis results should be presented unambiguously.

6. Release Criteria / Success Metrics

- **Functionality:** All core features (recording, upload, GTZAN-based analysis, result display, playback) operate correctly.

- **Stability:** The application runs without frequent crashes or critical errors during typical use.
- **Basic Usability:** Users can successfully complete the primary use cases without significant confusion.
- **(Post-Release) Success Metrics:**
 - *Accuracy:* Evaluate classification accuracy on a test set of audio files (if available).
 - *User Engagement:* Track the number of analyses performed (uploads vs. recordings).
 - *Task Completion Rate:* Observe if users successfully obtain genre results.
 - *Qualitative Feedback:* Gather user opinions on ease of use and result usefulness.

7. Future Considerations

- **Confidence Scores:** Re-implement and display confidence scores alongside genre predictions.
- **Alternative Models:** Integrate and allow switching to other models (like VGGish with proper weights, or the feature-based HighConfidenceClassifier) for comparison or improved performance.
- **Enhanced Visualizations:** Re-enable and refine visualizations for rhythm, melody, and instrumentation analysis.
- **Error Handling:** Improve granularity of error messages.
- **Performance Optimization:** Further optimize analysis time, especially for longer files if the 60s limit is removed.
- **Expanded Genre Set:** Train or find models supporting a wider range of genres or sub-genres.