**Portfolio Milestone: Dynamic Time Warping App**

Chioma Chance

Colorado State University Global

CSC506: Design and Analysis of Algorithms

Dr. Banerjee

03/09/2025

**Algorithm Implementation and Testing for Dynamic Time Warping (DTW) Music Analysis**

**Introduction**

The purpose of this project is to implement and test algorithms for extracting and analyzing audio features using Dynamic Time Warping (DTW) techniques. DTW is commonly used for time series alignment, particularly in music and speech processing. The goal was to develop a program that processes audio files, extracts Mel-Frequency Cepstral Coefficients (MFCCs), and visualizes the extracted features. This project involved implementing the necessary algorithms, handling audio file formats, and ensuring proper execution of the program.

**Program Implementation**

The program was developed in Python using the **Librosa** library for audio processing, **Matplotlib** for visualization, and **Scipy** for distance calculations. The implementation consisted of three main functions:

1. **load\_audio(file\_path)**: Loads an audio file and extracts the waveform and sample rate.
2. **extract\_features(signal, sr)**: Extracts MFCC features from the audio signal to capture relevant frequency-based information.
3. **plot\_mfcc(mfcc, title)**: Generates a heatmap visualization of the extracted MFCC features to analyze the frequency components over time.

The program was executed in a virtual environment within Visual Studio Code. A test audio file (test\_audio.wav) was used to verify the program's output, and the resulting MFCC visualization successfully displayed the extracted features.

**Challenges Faced**

Implementing the program required careful handling of dependencies and library functions. The main challenges encountered were:

* **Ensuring compatibility with required libraries**: Installing and managing dependencies such as librosa, matplotlib, and numpy required configuring the virtual environment correctly.
* **Handling file formats**: Initially, the test audio file was in .mp3 format, which led to compatibility issues with librosa.load(). Converting the file to .wav resolved the issue.
* **Understanding MFCC extraction**: Extracting meaningful features required determining appropriate parameter values, such as the number of MFCC coefficients to retain (13 in this case).
* **Debugging file path errors**: The program initially failed to locate the test file due to incorrect naming (.wav.wav). Fixing the filename resolved the issue.

Through trial and error, watching YouTube videos, along with referencing pseudocode from various sources, I successfully structured the program to function correctly.

**Analysis of Algorithm Efficiency**

The implemented approach efficiently extracts and visualizes MFCC features, which is crucial for DTW-based music analysis. MFCC extraction operates in **O(n log n)** time complexity due to Fourier transforms, making it computationally feasible for real-time applications. The overall performance is optimized for handling multiple audio inputs, ensuring efficient processing within reasonable execution times.

**Conclusion**

This project enhanced my understanding of audio processing techniques and the practical implementation of DTW-related algorithms. While Bubble Sort and Merge Sort were conceptually easier to grasp in previous projects, implementing audio feature extraction required deeper exploration of signal processing concepts. Successfully visualizing the MFCC features validated the correctness of the implementation. Moving forward, additional enhancements such as real-time DTW comparison and extended feature extraction can further improve the application’s capabilities.

**Screenshots**

**A screen shot of a graph

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**References**

* Lysecky, R., & Vahid, F. (2019, August). *Design and analysis of algorithms.* In R. Lysecky, & F. Vahid, *Data structures essential: Pseudocode with Python examples.* Zybooks. ISBN: 9781394012268
* Müller, M. (2015). *Fundamentals of music processing: Audio, analysis, algorithms, applications.* Springer.
* Ellis, D. P. W. (2005). *PLP and RASTA (and MFCC, and inversion) in Matlab.* Columbia University. Retrieved from <https://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/>