When approaching this project, my initial thought was to use Librosa, a well-known library for audio analysis. However, I wanted to explore other options before making a final decision. During my research, I came across libfmp, which I found through this paper https://joss.theoj.org/papers/10.21105/joss.03326.pdf this paper.then.

After reviewing both libraries, I discovered that libfmp is designed primarily for academic music research, focusing on advanced beat tracking, music structure analysis, and harmonic analysis. On the other hand, Librosa is a general-purpose audio analysis library and comes with built-in functions for:

- Beat tracking
- Tempo estimation
- Onset detection
- Spectrogram processing

As that my previous project involved spectrograms and CNNs, I am already familiar with the key audio features like frequency, amplitude, and time. Since this project focuses on beat tracking, I decided to stick with using Librosa for its ease of use and robust functionality.

Differences From My Previous Work

1. Instead of classifying pitch, we're attempting to identify when beats happen.
2. Instead of spectrograms, onset detection and amplitude tracking are being used to find rhythmic patterns.

Implementation

1. Load the Audio File:

The waveform (y) and sampling rate (sr) are extracted from the given audio file using librosa.load().Detect Beats and Estimate Tempo:

The function librosa.beat.beat_track() processes the waveform to detect beats and estimate the tempo in beats per minute (BPM).

2. Convert Beat Frames to Time Values:

The detected beat positions, initially represented as frame indices, are converted into actual time values using librosa.frames_to_time().

3. Plot the Waveform and Mark Beats:

A waveform plot is created using librosa.display.waveshow(), and the detected beats are visually marked with vertical red dashed lines using plt.vlines().

4. Manually Verify Tempo Using Onset Detection:

The onset strength (which measures the energy of musical events) is computed using librosa.onset.onset_strength(). Then, librosa.beat.tempo() is used as a backup method to estimate tempo based on onset strength.

5. Display the Results:

The estimated tempo from beat_track() and the manually verified tempo are printed for comparison. The waveform visualization is also displayed with beat markers for easy analysis.

- Onset detection ensures accuracy by tracking the actual loudness peaks.If beat_track() provides an incorrect BPM, this acts as a backup verification method.Since tempo and beats are strongly tied to musical time signatures, this method helps analyze music structure rather than just frequencies.

Other Musical Aspects:

Now that we can track beats and estimate tempo, we can expand our analysis in several ways. We can identify the musical key (such as C Major or A Minor) to understand the song's tonality. By analyzing harmonic features, we can detect chord progressions and transitions. We can also classify the time signature (e.g., 4/4 or 3/4) to better understand the rhythm and groove of the music. Additionally, we can segment a track into instrumental sections like the intro, verse, and chorus to analyze its structure. Finally, by examining tempo and rhythmic patterns, we can explore emotion detection to infer the mood and energy of a song.