librosa is a powerful Python library designed for audio and music analysis. It provides a wide range of functionalities for loading audio files, manipulating audio signals, extracting features, and visualizing audio data. Here's an overview of how librosa works, focusing on its core functionalities:

**1. Loading Audio**

* **Function**: librosa.load()
* **Description**: Loads an audio file as a waveform and its corresponding sample rate. The audio is typically loaded as a time-series signal (1D array). The library supports various audio formats such as WAV, MP3, and FLAC.
* **Parameters**:
  + filename: Path to the audio file.
  + sr: Sample rate (if set to None, it uses the original sample rate of the file).
  + mono: If True, converts the audio to mono by averaging the channels.

**Example:**

python

Copy code

import librosa

y, sr = librosa.load('path/to/audio.wav', sr=22050, mono=True)

**2. Time-Domain Analysis**

* You can perform various analyses in the time domain, such as measuring the duration of the audio, slicing the signal, or visualizing waveforms.

**3. Frequency-Domain Analysis**

* **STFT**: Short-Time Fourier Transform can be computed using librosa.stft(), which represents the audio in the time-frequency domain.
* **Spectrogram**: A visual representation of the spectrum of frequencies in a sound signal as they vary with time. You can use librosa.feature.melspectrogram() or librosa.feature.chroma\_stft() to extract different types of spectrograms.

**Example:**

python

Copy code

import numpy as np

# Compute the Short-Time Fourier Transform (STFT)

D = librosa.stft(y)

# Convert the amplitude to decibels for visualization

DB = librosa.amplitude\_to\_db(np.abs(D), ref=np.max)

**4. Feature Extraction**

* librosa includes numerous built-in functions for extracting features commonly used in audio analysis, such as:
  + **Mel-frequency cepstral coefficients (MFCCs)**: Used in speech and music analysis.
  + **Chromagram**: Represents the energy of the 12 different pitch classes.
  + **Spectral contrast**: Represents the difference in amplitude between peaks and valleys in a sound spectrum.

**Example:**

python

Copy code

mfccs = librosa.feature.mfcc(y=y, sr=sr, n\_mfcc=13)

**5. Audio Effects and Transformations**

* You can apply various audio effects, such as time-stretching, pitch-shifting, and filtering using librosa functions. These transformations are essential in music processing and audio manipulation.

**Example:**

python

Copy code

# Time-stretching

y\_stretched = librosa.effects.time\_stretch(y, rate=1.5)

# Pitch-shifting

y\_shifted = librosa.effects.pitch\_shift(y, sr=sr, n\_steps=4)

**6. Visualization**

* librosa integrates well with libraries like matplotlib for visualizing audio data. You can plot waveforms, spectrograms, and features to better understand the audio content.

**Example:**

python

Copy code

import matplotlib.pyplot as plt

# Plot the waveform

plt.figure(figsize=(14, 5))

librosa.display.waveshow(y, sr=sr)

plt.title('Waveform')

plt.show()

**Summary**

* **Versatile**: librosa supports a wide range of audio formats and provides tools for loading, analyzing, and visualizing audio data.
* **Feature-Rich**: It includes numerous features for audio analysis, such as MFCCs, chroma features, and spectral analysis.
* **Easy to Use**: The library is user-friendly and well-documented, making it accessible for both beginners and experts in audio processing.

This combination of features makes librosa a go-to library for researchers, developers, and enthusiasts working in the field of audio and music analysis.

4o mini