AI project - Audio Analysis

Overview :

Python Packages :

Audio Data Handling using Python

Library	Description	Source
Librosa	It is a Python module to analyze audio signals in general but geared more towards music. It includes the nuts and bolts to build a MIR(Music information retrieval) system. It has been very well documented along with a lot of examples and tutorials.	https://librosa.org/doc/latest/index.html
IPython.display.Audio	`IPython.display.Audio` lets you play audio directly in a jupyter notebook. I have uploaded a random audio file on the below page. Let us now load the file in your jupyter console.	https://ipython.org/ipython-doc/stable/api/generated/IPython.display.html#IPython.display.Audio
pyAudioAnalysis	pyAudioAnalysis is a Python library covering a wide range of audio analysis tasks. Through pyAudioAnalysis you can: • Extract audio *features* and representations (e.g. mfccs, spectrogram, chromagram) • *Train*, parameter tune and *evaluate* classifiers of audio segments • *Classify* unknown sounds • *Detect* audio events and exclude silence periods from long recordings • Perform *supervised segmentation* (joint segmentation - classification) • Perform *unsupervised segmentation* (e.g. speaker diarization) and extract audio *thumbnails* • Train and use *audio regression* models (example application: emotion recognition) • Apply dimensionality reduction to *visualize* audio data and content similarities	https://github.com/tyiannak/pyAudioAnalysis https://medium.com/behavioral-signals-ai/intro-to-audio-analysis-recognizing-sounds-using-machine-learning-20fd646a0ec5 http://rwx.io/posts/simple-audio-analysis-in-python/
`torchaudio`	The aim of torchaudio is to apply PyTorch to the audio domain. By supporting PyTorch, torchaudio follows the same philosophy of providing strong GPU acceleration, having a focus on trainable features through the autograd system, he benefits of PyTorch can be seen in torchaudio through having all the computations be through PyTorch operations which makes it easy to use and feel like a natural extension.	https://pytorch.org/audio/stable/index.html https://github.com/pytorch/audio
pydub		https://github.com/jiaaro/pydub

Feature extraction from Audio signal

Spectral Centroid	The **spectral centroid** indicates at which frequency the energy of a spectrum is centered upon or in other words It indicates where the ” center of mass” for a sound is located.	https://librosa.org/doc/latest/generated/librosa.feature.spectral_centroid.html?highlight=librosa%20feature%20spectral_centroid#librosa.feature.spectral_centroid
Spectral Rolloff	It is a measure of the shape of the signal. It represents the frequency at which high frequencies decline to 0. To obtain it, we have to calculate the fraction of bins in the power spectrum where 85% of its power is at lower frequencies.	https://librosa.org/doc/latest/generated/librosa.feature.spectral_rolloff.html?highlight=librosa%20feature%20spectral_rolloff#librosa.feature.spectral_rolloff
Spectral Bandwidth	The spectral bandwidth is defined as the width of the band of light at one-half the peak maximum (or full width at half maximum [FWHM]) and is represented by the two vertical red lines and SB on the wavelength axis.	https://librosa.org/doc/latest/generated/librosa.feature.spectral_bandwidth.html?highlight=librosa%20feature%20spectral_bandwidth#librosa.feature.spectral_bandwidth
Zero-Crossing Rate	A very simple way for measuring the smoothness of a signal is to calculate the number of zero-crossing within a segment of that signal. A voice signal oscillates slowly — for example, a 100 Hz signal will cross zero 100 per second — whereas an unvoiced fricative can have 3000 zero crossings per second.	https://librosa.org/doc/latest/generated/librosa.zero_crossings.html?highlight=librosa%20zero_crossings#librosa.zero_crossings
Mel-Frequency Cepstral Coefficients(MFCCs)	The Mel frequency cepstral coefficients (MFCCs) of a signal are a small set of features (usually about 10–20) which concisely describe the overall shape of a spectral envelope. It models the characteristics of the human voice.	

Chroma feature	A **chroma feature or vector** is typically a 12-element feature vector indicating how much energy of each pitch class, {C, C#, D, D#, E, …, B}, is present in the signal. In short, It provides a robust way to describe a similarity measure between music pieces.	https://librosa.org/doc/latest/generated/librosa.feature.chroma_stft.html?highlight=librosa%20feature%20chroma_stft#librosa.feature.chroma_stft
Spectrogram	A spectrogram is a visual way of representing the signal strength, or "**loudness**", of a signal over time at various frequencies present in a particular waveform. Not only can one see whether there is more or less energy at, for example, 2 Hz vs 10 Hz, but one can also see how energy levels vary over time. A spectrogram is usually depicted as a heat map, i.e., as an image with the intensity shown by varying the color or brightness. We can display a spectrogram using. `librosa.display.specshow.`	https://librosa.org/doc/latest/generated/librosa.display.specshow.html?highlight=librosa%20display%20specshow#librosa.display.specshow

Reference Links :

https://www.kdnuggets.com/2020/02/audio-data-analysis-deep-learning-python-part-1.html

https://pythonrepo.com/repo/tyiannak-pyAudioAnalysis-python-audio

Soundfile Installation

https://pysoundfile.readthedocs.io/en/latest/#installation

What we can do with audio files

- Extract audio *features* and representations (e.g. mfccs, spectrogram, chromagram)
- *Train*, parameter tune and *evaluate* classifiers of audio segments
- *Classify* unknown sounds
- *Detect* audio events and exclude silence periods from long recordings
- Perform *supervised segmentation* (joint segmentation - classification)

mfccs (Music Feature Extraction in Python) - > https://towardsdatascience.com/extract-features-of-music-75a3f9bc265d

https://musicinformationretrieval.com/chroma.html

https://towardsdatascience.com/audio-genre-classification-with-python-oop-66119e10cd05

https://github.com/makcedward/nlpaug

https://pypi.org/project/librosa/#history

https://medium.com/behavioral-signals-ai/intro-to-audio-analysis-recognizing-sounds-using-machine-learning-20fd646a0ec5

https://github.com/tyiannak/pyAudioAnalysis