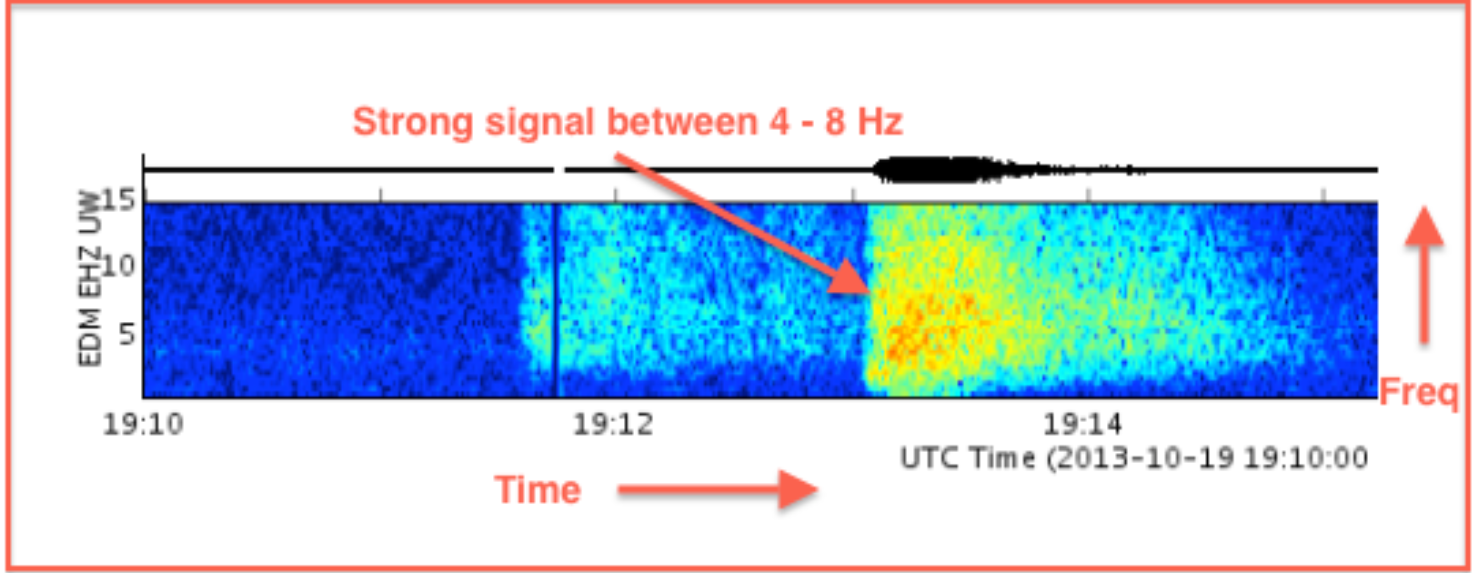
**Audio Analysis**

**1) What is Librosa?**

LibROSA is a python package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems.

**2) What is a 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. In other sciences spectrograms are commonly used to display frequencies of sound waves produced by humans, machinery, animals, whales, jets, etc., as recorded by microphones.

****

**3) Define the following spectrogram characteristics:**

**- MFCC (Mel frequency cepstral coefficients)**

The first step in any automatic speech recognition system is to extract features i.e. identify the components of the audio signal that are good for identifying the linguistic content and discarding all the other stuff which carries information like background noise, emotion etc.

The main point to understand about speech is that the sounds generated by a human are filtered by the shape of the vocal tract including tongue, teeth etc. This shape determines what sound comes out. If we can determine the shape accurately, this should give us an accurate representation of the phoneme being produced. The shape of the vocal tract manifests itself in the envelope of the short time power spectrum, and the job of MFCCs is to accurately represent this envelope.

**Mel Frequency Cepstral Coefficents** (MFCCs) are a feature widely used in automatic speech and speaker recognition. They were introduced by Davis and Mermelstein in the 1980's, and have been state-of-the-art ever since. Prior to the introduction of MFCCs, Linear Prediction Coefficients (LPCs) and Linear Prediction Cepstral Coefficients (LPCCs) and were the main feature type for automatic speech recognition (ASR).

**librosa.feature.mfcc(*y=None*, *sr=22050*, *S=None*, *n\_mfcc=20*, *dct\_type=2*, *norm='ortho'*, *lifter=0*, *\*\*kwargs*)[[source]](https://librosa.github.io/librosa/_modules/librosa/feature/spectral.html" \l "mfcc)**

**- Spectral Centroid**

Spectral Centroid The spectral centroid is commonly associated with the measure of the brightness of a sound. This measure is obtained by evaluating the “center of gravity” using the Fourier transform’s frequency and magnitude information. The individual centroid of a spectral frame is defined as the average frequency weighted by amplitudes, divided by the sum of the amplitudes, or: ∑ ∑ = = = N k N k F k kF k Spectral Centroid 1 1 [ ] [ ] Here, F [k] is the amplitude corresponding to bin k in DFT spectrum. In practice, centroid finds this frequency for a given frame, and then finds the nearest spectral bin for that frequency. The centroid is usually a lot higher than one might intuitively expect, because there is so much more energy above (than below) the fundamental which contributes to the average. It is not sure if the spectral centroid would be useful for classifying different genres of musics. At least, it will show some spectral components of the music, which are mixed sounds.

**librosa.feature.spectral\_centroid(*y=None*, *sr=22050*, *S=None*, *n\_fft=2048*, *hop\_length=512*, *freq=None*, *win\_length=None*, *window='hann'*, *center=True*, *pad\_mode='reflect'*)[[source]](https://librosa.github.io/librosa/_modules/librosa/feature/spectral.html" \l "spectral_centroid)**

**- Zero Crossing Rate**

By looking at different speech and audio waveforms, we can see that depending on the content, they vary a lot in their smoothness. For example, voiced speech sounds are more smooth than unvoiced ones. Smoothness is thus a informative characteristic of the signal.

A very simple way for measuring 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 crossing per second.

To calculate of the zero-crossing rate of a signal you need to compare the sign of each pair of consecutive samples. In other words, for a length N signal you need O(N) operations. Such calculations are also extremely simple to implement, which makes the zero-crossing rate an attractive measure for low-complexity applications. However, there are also many drawbacks with the zero-crossing rate:

The number of zero-crossings in a segment is an integer number. A continuous-valued measure would allow more detailed analysis.

Measure is applicable only on longer segments of the signal, since short segments might not have any or just a few zero crossings.

To make the measure consistent, we must assume that the signal is zero-mean. You should therefore subtract the mean of each segment before calculating the zero-crossings rate.

**librosa.feature.zero\_crossing\_rate(*y*, *frame\_length=2048*, *hop\_length=512*, *center=True*, *\*\*kwargs*)[[source]](https://librosa.github.io/librosa/_modules/librosa/feature/spectral.html" \l "zero_crossing_rate)**

**- Chroma Frequencies**

In music, the term chroma feature or chromagram closely relates to the twelve different pitch classes. Chroma-based features, which are also referred to as "pitch class profiles", are a powerful tool for analyzing music whose pitches can be meaningfully categorized (often into twelve categories) and whose tuning approximates to the equal-tempered scale.

**librosa.feature.chroma\_stft(*y=None*, *sr=22050*, *S=None*, *norm=inf*, *n\_fft=2048*, *hop\_length=512*, *win\_length=None*, *window='hann'*, *center=True*, *pad\_mode='reflect'*, *tuning=None*, *n\_chroma=12*, *\*\*kwargs*)[[source]](https://librosa.github.io/librosa/_modules/librosa/feature/spectral.html" \l "chroma_stft)**

**- Spectral Roll-off**

**librosa.feature.spectral\_rolloff(*y=None*, *sr=22050*, *S=None*, *n\_fft=2048*, *hop\_length=512*, *win\_length=None*, *window='hann'*, *center=True*, *pad\_mode='reflect'*, *freq=None*, *roll\_percent=0.85*)[[source]](https://librosa.github.io/librosa/_modules/librosa/feature/spectral.html" \l "spectral_rolloff)**

The roll-off frequency is defined for each frame as the center frequency for a spectrogram bin such that at least roll\_percent (0.85 by default) of the energy of the spectrum in this frame is contained in this bin and the bins below. This can be used to, e.g., approximate the maximum (or minimum) frequency by setting roll\_percent to a value close to 1 (or 0).

|  |
| --- |
|  |