AASD 4004
Machine Learning - II

Applied AI Solutions Developer Program

Module 15
Feature Extraction in Audio

Vejey Gandyer

Agenda

- Feature Extraction
- Short-term feature extraction
- Mid-term feature extraction
- Spectral Centroids
- Spectral Rolloff
- Spectrogram
- MFCC

Feature Extraction

What is it?

Feature Extraction

Extracting a set of features that are informative with respect to the desired properties of the original data

Low-level features to construct a higher-level of understanding

Need to extract audio features capable of discriminating between different audio classes i.e. speakers, emotions, genres

Short-term windowing (framing)

Split audio signal into short-term overlapping or non-overlapping windows (frames)

Length of the frames 10ms to 100ms

For each frame, extract a set of short-term Time-domain features (directly from audio sample values) or Frequency-domain features (from FFT values of the signal)

Ex: If **energy** and **spectral centroid** are extracted, sequence of 2D short-term feature vectors are outputted

If the audio files have different duration, what will happen?

Representing arbitrary-sized audio segments

Zero pad the feature sequences to the maximum duration of the dataset, concatenate the different short-term feature sequences to a single feature vector

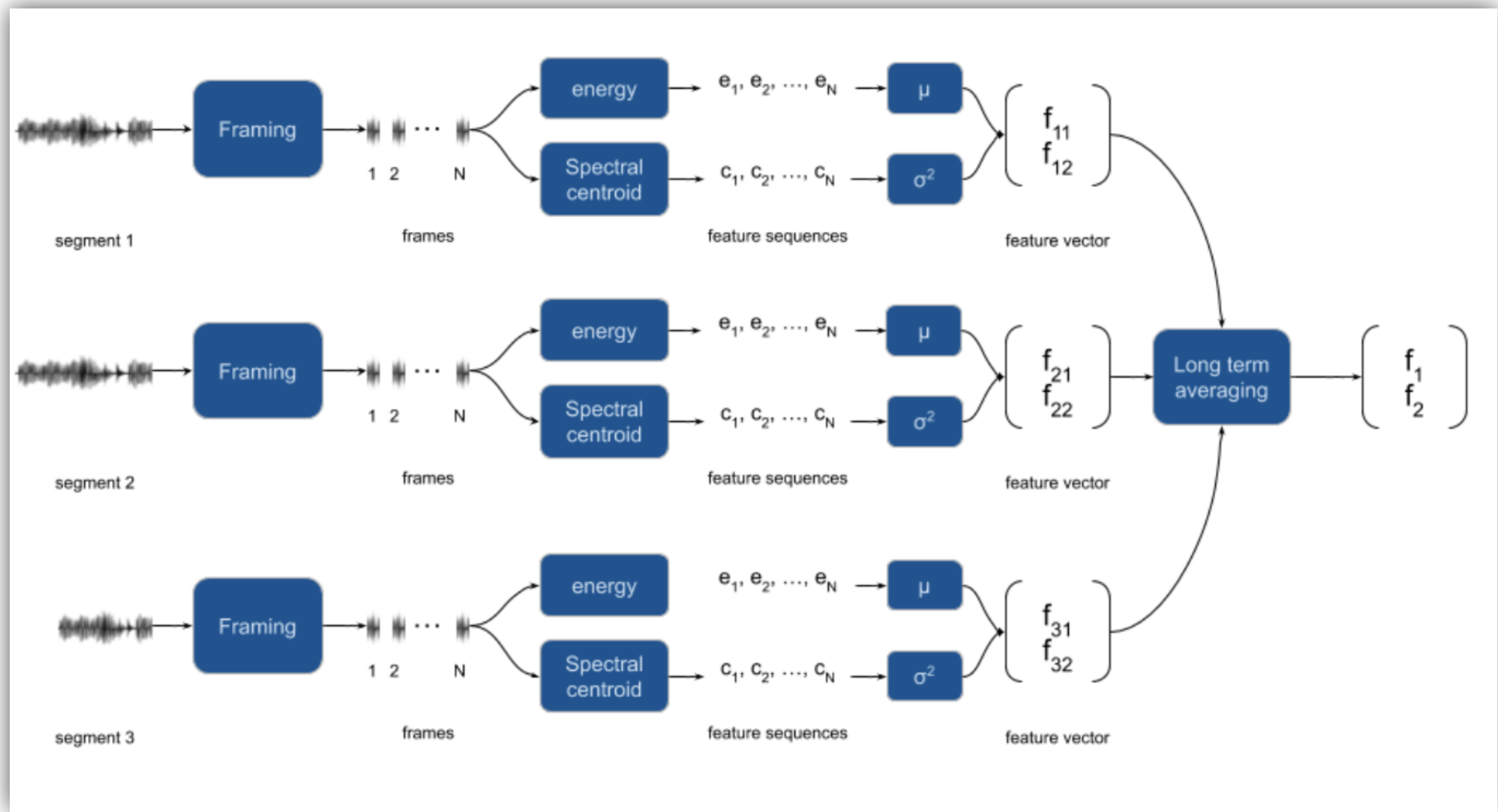
Issues

1. High dimensionality, so need for more data samples while training
2. Dependent on temporal positions of feature values

Representing arbitrary-sized audio segments

Solution: Extract a set of **feature statistics** (usually Mean and SD) per fix-sized segment

Segment-level statistics extracted over the short-term feature sequences are the representations for each fix-sized segment



Short-term Feature Extraction

Short-term Feature Extraction

Splits the input signal into short-term windows (frames) and computes several features for each frame

Total number of short-term features is 34

Short-term Feature Extraction

Feature ID	Feature Name	Description
1	Zero Crossing Rate	The rate of sign-changes of the signal during the duration of a particular frame.
2	Energy	The sum of squares of the signal values, normalized by the respective frame length.
3	Entropy of Energy	The entropy of sub-frames' normalized energies. It can be interpreted as a measure of abrupt changes.
4	Spectral Centroid	The center of gravity of the spectrum.
5	Spectral Spread	The second central moment of the spectrum.
6	Spectral Entropy	Entropy of the normalized spectral energies for a set of sub-frames.
7	Spectral Flux	The squared difference between the normalized magnitudes of the spectra of the two successive frames.
8	Spectral Rolloff	The frequency below which 90% of the magnitude distribution of the spectrum is concentrated.
9-21	MFCCs	Mel Frequency Cepstral Coefficients form a cepstral representation where the frequency bands are not linear but distributed according to the mel-scale.
22-33	Chroma Vector	A 12-element representation of the spectral energy where the bins represent the 12 equal-tempered pitch classes of western-type music (semitone spacing).
34	Chroma Deviation	The standard deviation of the 12 chroma coefficients.

Audio Analysis Tasks

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) & extract audio thumbnails

Train and use audio regression models (example application: emotion recognition)

Apply dimensionality reduction to visualize audio data and content similarities

Short-term feature sequences

```
from pyAudioAnalysis import ShortTermFeatures as aF
from pyAudioAnalysis import audioBasicIO as aIO
fs, s = aIO.read_audio_file("data/object.wav")
win, step = 0.050, 0.050
[f, fn] = aF.feature_extraction(s, fs, int(fs * win), int(fs * step))
```

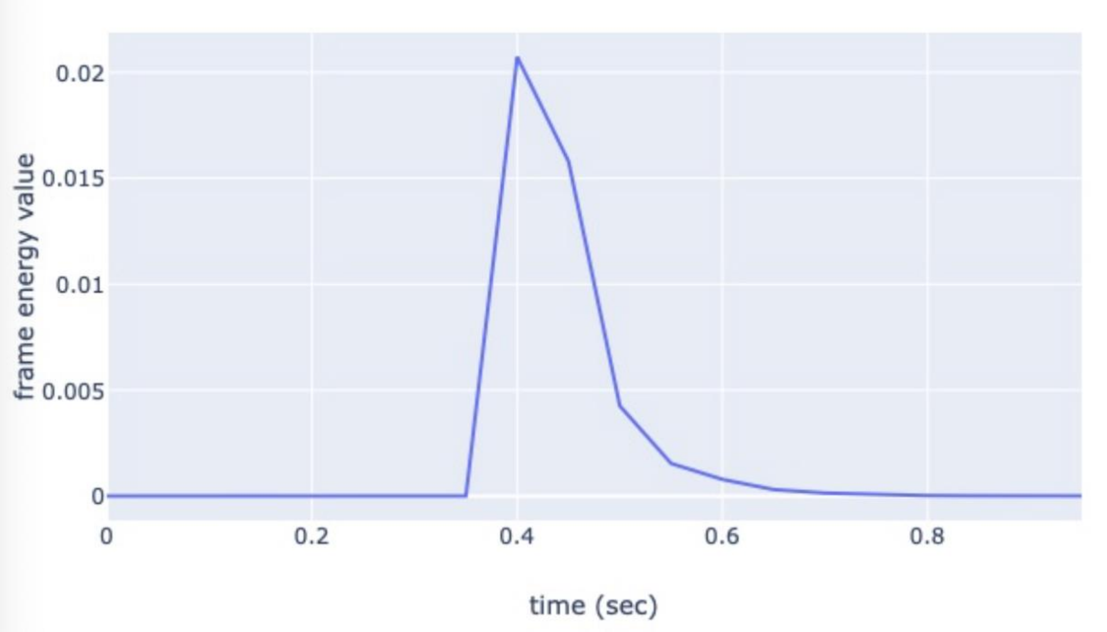
read_audio_file()

Sampling rate $\mathbf{F_s}$ & Array of raw values **s**

Feature_extraction()

68 x 20 Short-term feature matrix

68-length list of strings



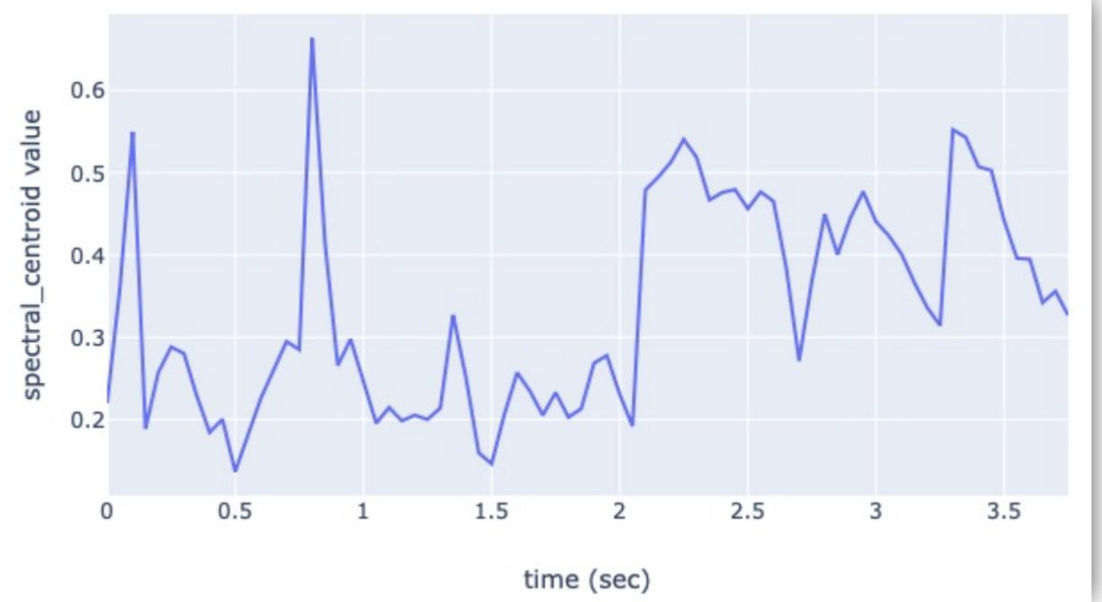
Spectral Centroid Short-term feature

```
from pyAudioAnalysis import ShortTermFeatures as aF
from pyAudioAnalysis import audioBasicIO as aIO
fs, s = aIO.read_audio_file("data/trump_bugs.wav")
win, step = 0.050, 0.050
[f, fn] = aF.feature_extraction(s, fs, int(fs * win), int(fs * step))
energy = f[fn.index('spectral_centroid'), :]
```

Spectral centroid

Centroid of the FFT magnitude normalized in the frequency range [0, Fs / 2]

Spectral centroid = 0.5 = Fs / 4 in Hz



Mid-term Feature Extraction

Segment-level Statistics Mid-term feature

```
from pyAudioAnalysis import MidTermFeatures as aF
mt, st, mt_n = aF.mid_feature_extraction(s, fs, 1 * fs, 1 * fs, 0.05 * fs, 0.05 * fs)
```

mid_feature_extraction() Mean and SD

Example:

Duration of audio clip = 3.8 seconds

Mid-term window step & size = 1 second

No. Of Mid-term segments = 4

No. Of Short-term frames = 3.8 / 0.05 = 76

```
signal duration 3.812625 seconds
76 68-D short-term feature vectors extracted
4 136-D segment feature statistic vectors extracted
mid-term feature names
0:zcr_mean
1:energy_mean
2:energy_entropy_mean
3:spectral_centroid_mean
4:spectral_spread_mean
5:spectral_entropy_mean
6:spectral_flux_mean
7:spectral_rolloff_mean
8:mfcc_1_mean
...
131:delta chroma_9_std
132:delta chroma_10_std
133:delta chroma_11_std
134:delta chroma_12_std
135:delta chroma_std_std
```

Segment-level Statistics Mid-term feature

Example: Analyze a song of 120 seconds with a short-term window (and step) of 50 ms and a mid-term (segment) window and step of 1 second. What are the steps?

Duration of audio clip = 120 seconds

No. Of Short-term frames = 120 / 0.05 = 2400; Short-term feature vectors = 2400 x 68

Mid-term window step & size = 1 second; No. Of Mid-term segments = 120 / 1 = 120

Segment-level feature statistics = 120 x 136 (Mean & SD of 68-D vector sequences)

120 x 136 matrix is long-term averaged for the whole song

2 beat-related features are appended as well; Total 120 x 138 values

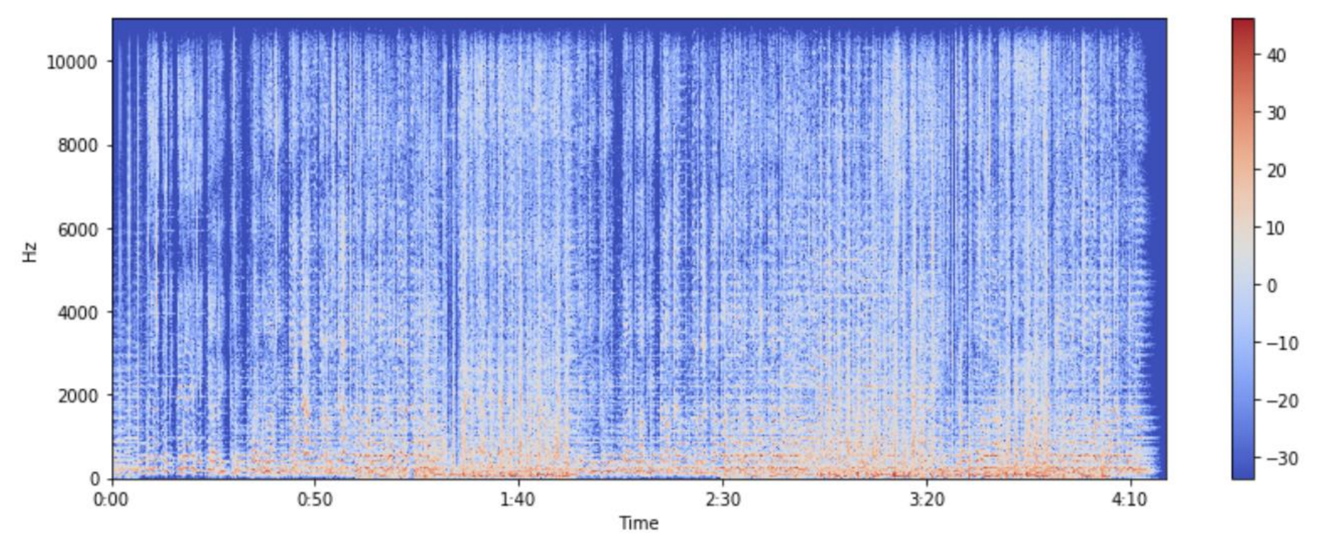
Spectrogram

Spectrogram

Visual Representation of frequencies changing with respect to time for given music

```
X = librosa.stft(x)
Xdb = librosa.amplitude_to_db(abs(X))
plt.figure(figsize=(14, 5))
librosa.display.specshow(Xdb, sr=sr, x_axis='time', y_axis='hz')
librosa.display.specshow(Xdb, sr=sr, x_axis='time', y_axis='log')
plt.colorbar()
```

```
X = librosa.stft(x)
```

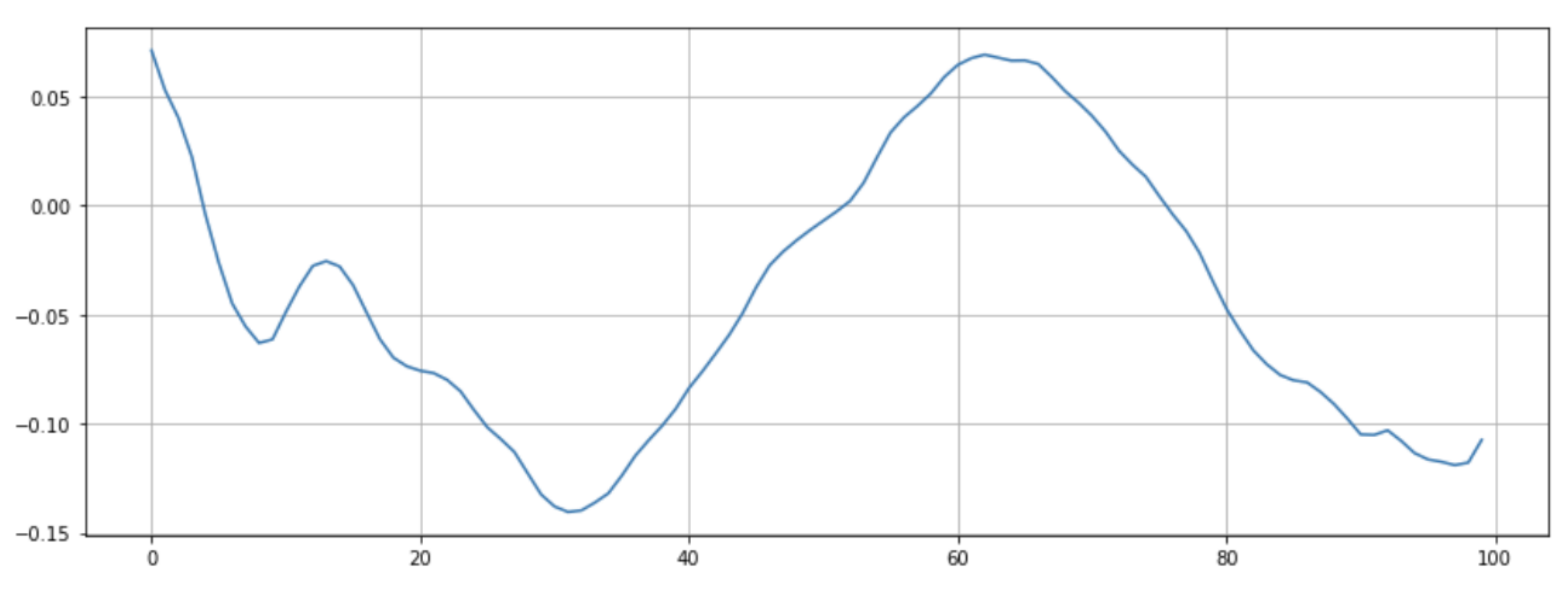


Zero Crossings

Zero Crossings

Rate of sign-changes along a signal

```
zero_crossings = librosa.zero_crossings(x[n0:n1], pad=False)
```

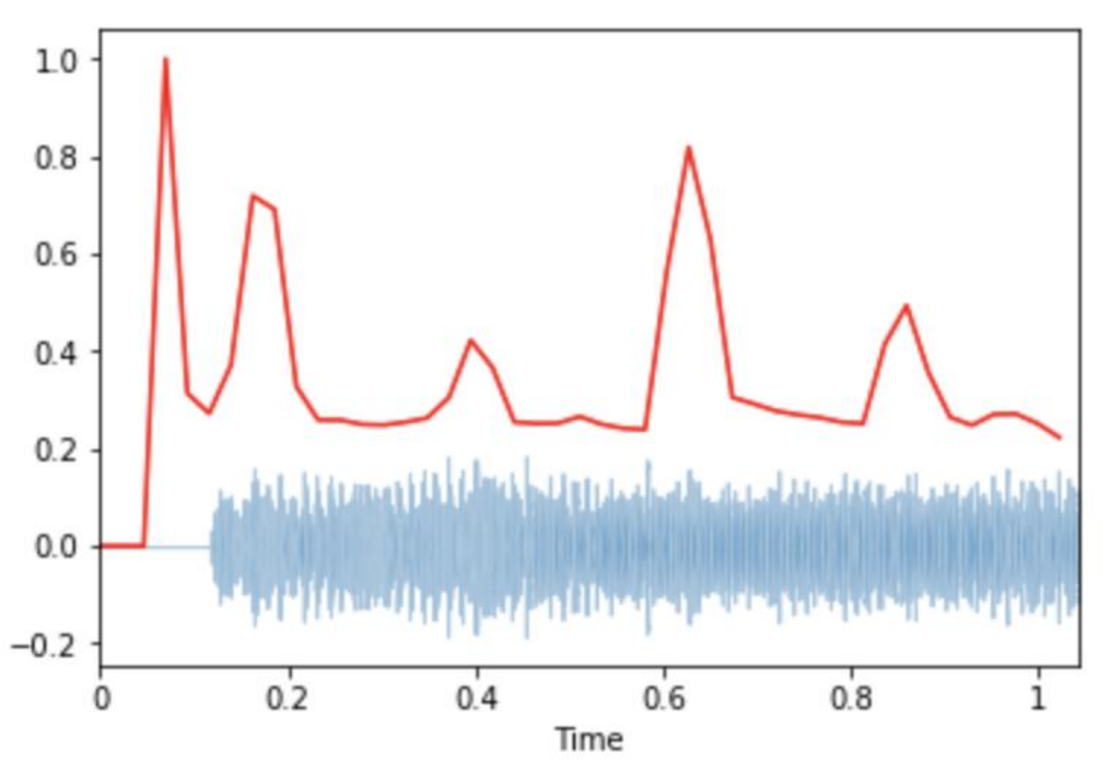


Spectral Centroid

Spectral Centroid

Centre of mass for a sound signal - weighted mean of the frequen

```
spectral_centroids = librosa.feature.spectral_centroid(x, sr=sr)[0]
```

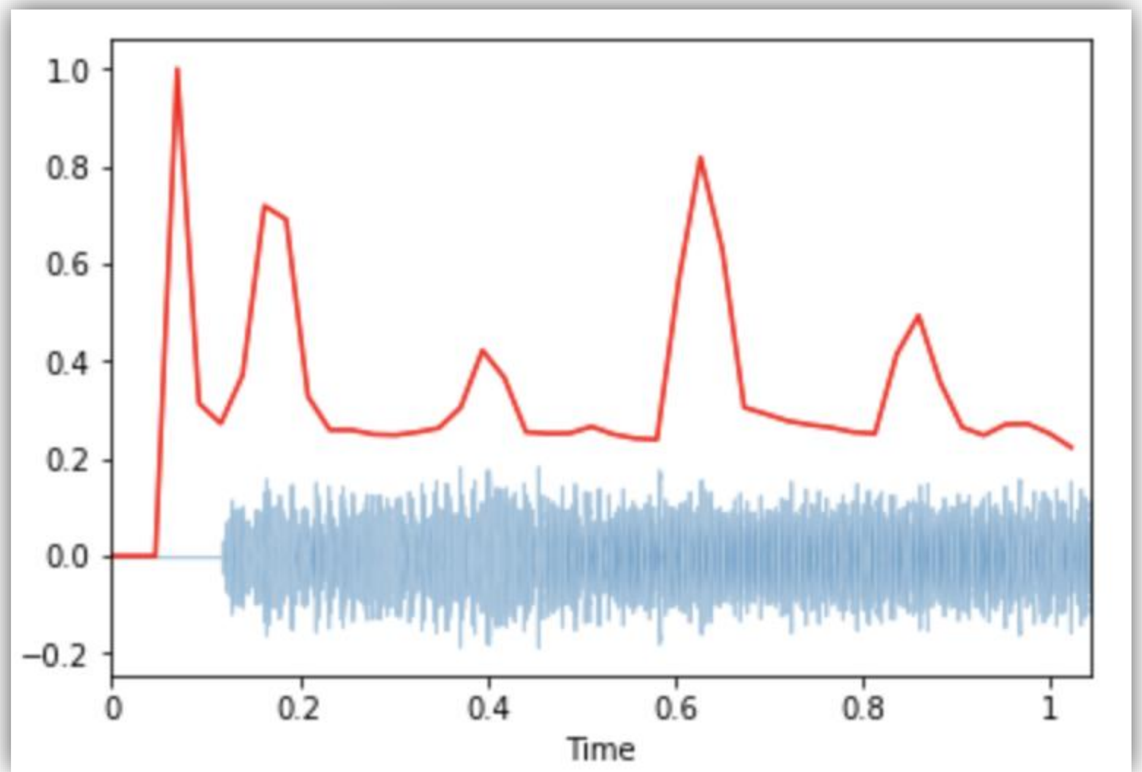


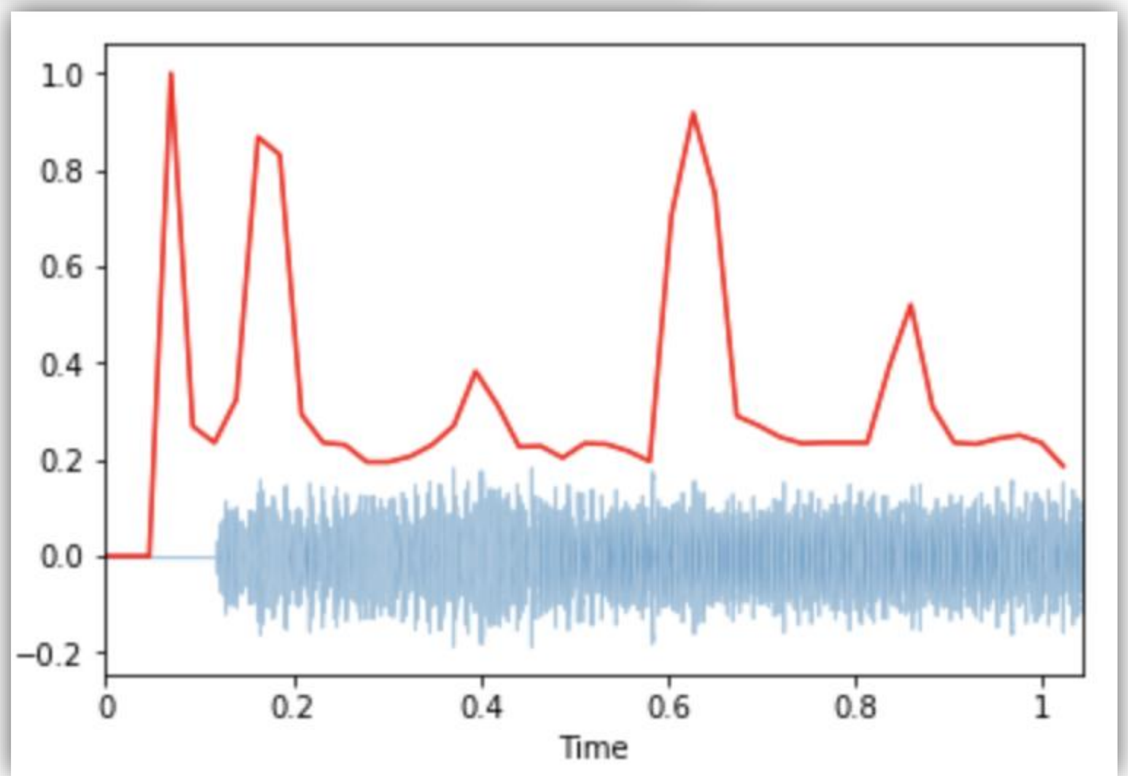
Spectral Rolloff

Spectral Rolloff

Frequency below which a specified percentage of the total spectral energy lies

```
spectral_rolloff = librosa.feature.spectral_rolloff(x, sr=sr)[0]
```



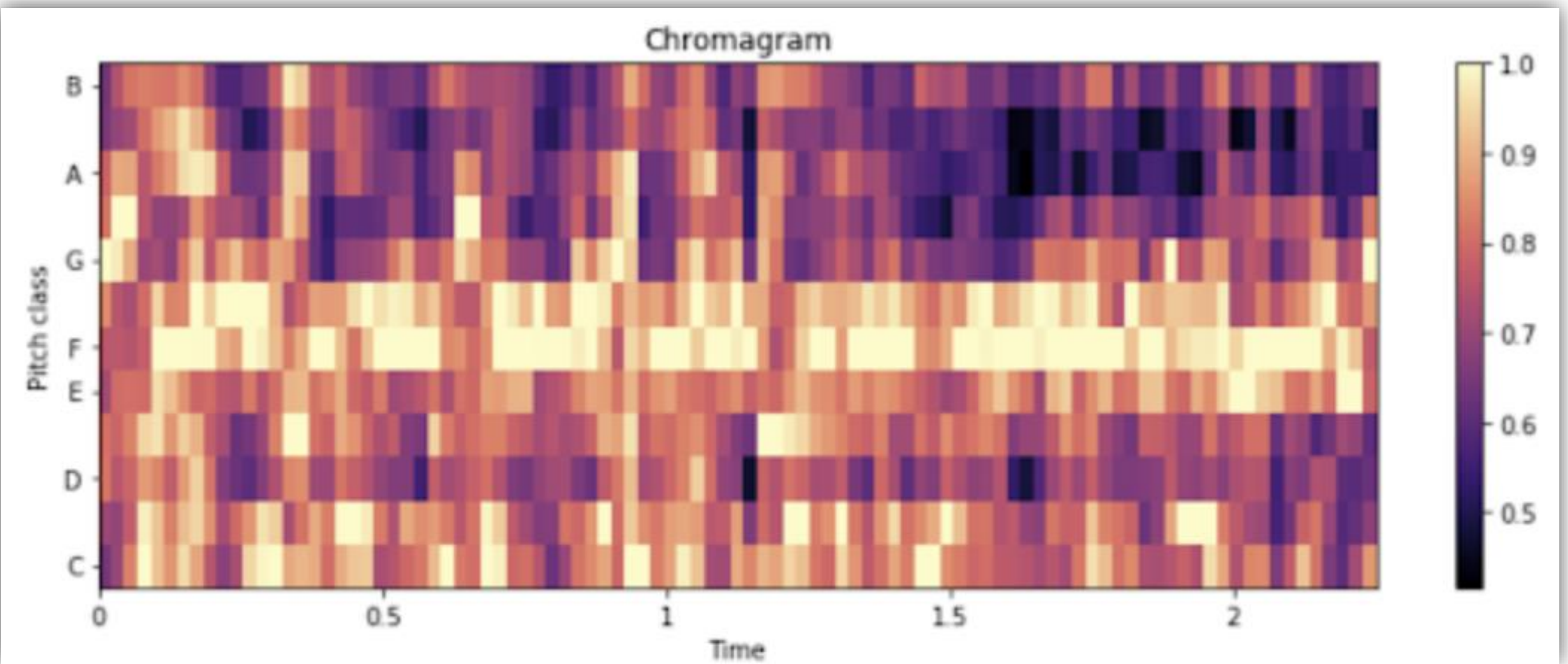


Chromogram

Chromagram

Relates to twelve different pitch classes

Powerful tool for analyzing music with respect to pitches



MFCC

Mel-Frequency Cepstral Coefficients

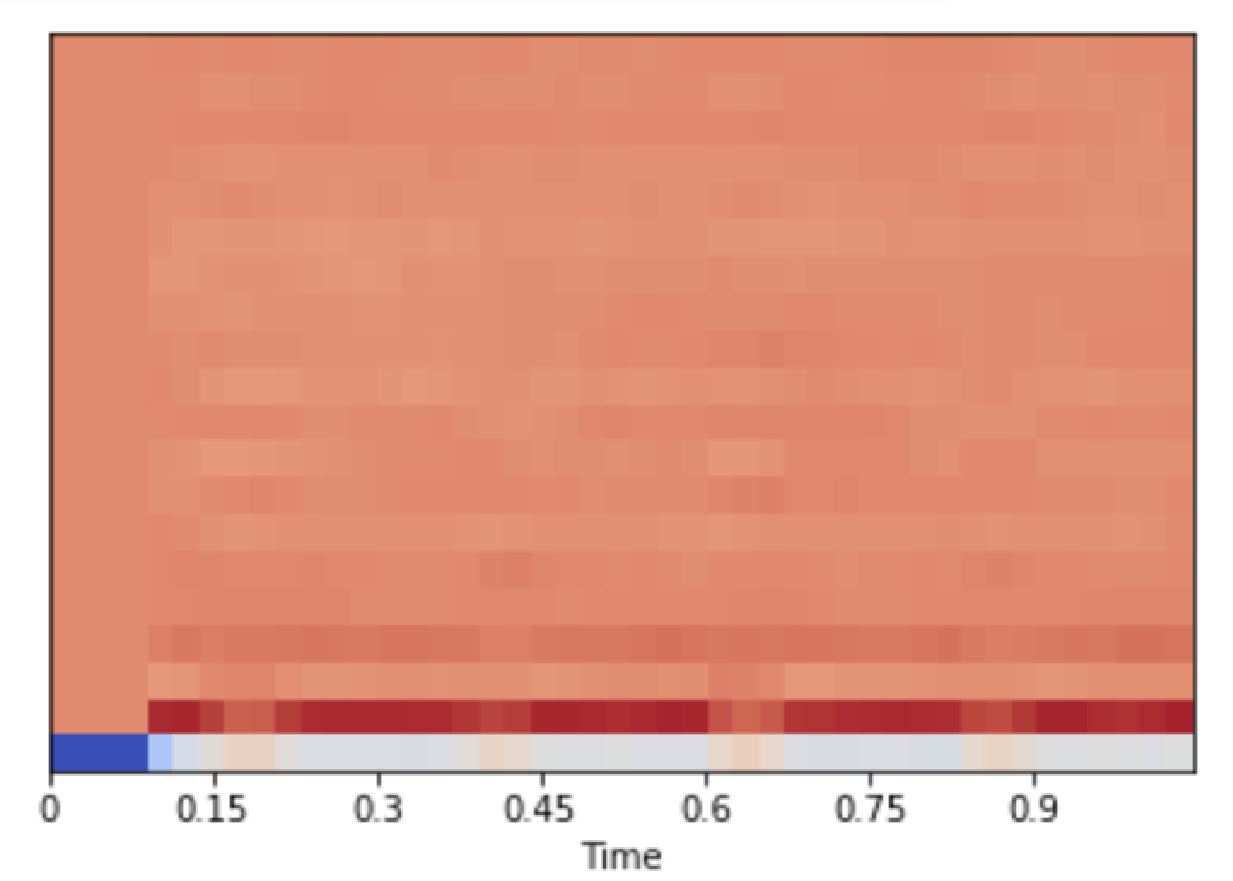
```
mfccs = librosa.feature.mfcc(x, sr=sr)
librosa.display.specshow(mfccs, sr=sr, x_axis='time')
```

Small set of features

(usually about 10-20)

Describes shape of a

spectral envelope



Further Reading

PyAudioAnalysis

https://github.com/tyiannak/pyAudioAnalysis

Feature Extraction

https://github.com/tyiannak/pyAudioAnalysis/wiki/3.-Feature-Extraction

https://github.com/tyiannak/basic_audio_analysis/blob/master/notebobok.ipynb

https://hackernoon.com/intro-to-audio-analysis-recognizing-sounds-using-machine-learning-qy2r3ufl