Feature extraction

In [3]:

```
header = 'filename chroma_stft_mean chroma_stft_var rms_mean rms_var spectral_centroid_
for i in range(1, 21):
    header += f'mfcc_mean{i} '
    header+=  f'mfcc_var{i} '
for j in range(1,129):
    header += f'mel_specgram_mean{j} '
    header += f'mel_specgram_var{j} '
header +='  tempo'
header += ' label'
header = header.split()
```

In [4]:

```
len(header)
```

Out[4]:

343

In [7]:

```
file = open('music.csv', 'w', newline='')
with file:
    writer = csv.writer(file)
    writer.writerow(header)
genres = 'final_blues final_classical final_country final_disco final_hiphop final_jazz
for g in genres:
    for filename in os.listdir(f'./final_genres/{g}'):
        songname = f'./final_genres/{g}/{filename}'
        y, sr = librosa.load(songname, mono=True, duration=30)
        chroma_stft = librosa.feature.chroma_stft(y=y, sr=sr)
        rms = librosa.feature.rms(y=y)
        spec_cent = librosa.feature.spectral_centroid(y=y, sr=sr)
        spec_bw = librosa.feature.spectral_bandwidth(y=y, sr=sr)
        rolloff = librosa.feature.spectral_rolloff(y=y, sr=sr)
        zcr = librosa.feature.zero_crossing_rate(y)
        mfcc = librosa.feature.mfcc(y=y, sr=sr)
        harmonics, perceptual = librosa.effects.hpss(y)
        tempo=librosa.beat.tempo(y=y,sr=sr)
        spectral_contrast=librosa.feature.spectral_contrast(y,sr=sr)
        #Spectral flatness
        spectral_flatness=librosa.feature.spectral_flatness(y=y)
        mel_spectrogram=librosa.feature.melspectrogram(y=y,sr=sr)
        to_append = f'{filename} {np.mean(chroma_stft)} {np.var(chroma_stft)} {np.mean(
        for e in mfcc:
            to_append += f' {np.mean(e)}'
            to_append += f' {np.var(e)}'
        for j in range(0,7):
            to_append += f' {np.mean(spectral_contrast[j])}'
            to_append += f' {np.var(spectral_contrast[j])}'
            j=0

        to_append += f' {np.mean(spectral_flatness)}'
        to_append += f' {np.var(spectral_flatness)}'
        y_harmonic=librosa.effects.harmonic(y=y)
        tonnetz=librosa.feature.tonnetz(y=y_harmonic,sr=sr)
        for k in range(0,6):
            to_append += f' {np.mean(tonnetz[j])}'
            to_append += f' {np.var(tonnetz[j])}'
            k=0
        for l in range(0,128):
            to_append += f' {np.mean(mel_spectrogram[j])}'
            to_append += f' {np.var(mel_spectrogram[j])}'
        to_append += f' {float(tempo)}'
        to_append += f' {g}'
        file = open('music.csv', 'a', newline='')
        with file:
            writer = csv.writer(file)
            writer.writerow(to_append.split())
```

```
C:\Users\PROMIT\anaconda3\envs\tensorflow\lib\site-packages\librosa\core\spe
ctrum.py:222: UserWarning: n_fft=1024 is too small for input signal of lengt
h=1011
  warnings.warn(
```

In [8]:

```
import pandas as pd
data=pd.read_csv('music.csv')
data.head()
```

Out[8]:

	filename	chroma_stft_mean	chroma_stft_var	rms_mean	rms_var	spectral_centroid_
0	final_blues00.wav	0.335434	0.091088	0.130405	0.003521	1773.2
1	final_blues01.wav	0.343020	0.086142	0.112699	0.001450	1816.1
2	final_blues02.wav	0.346838	0.092210	0.132002	0.004620	1788.6
3	final_blues03.wav	0.363671	0.086856	0.132562	0.002447	1654.9
4	final_blues04.wav	0.335927	0.088291	0.143289	0.001701	1630.7

5 rows × 343 columns

In [9]:

```
data.shape
```

Out[9]:

```
(10000, 343)
```