Name - Sia Vashist

PRN.No - 20190802107

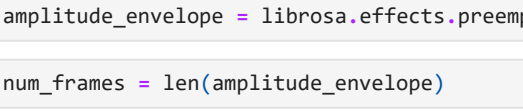
In [ ]:
```
import librosa
import librosa.display
import IPython.display as ipd
import matplotlib.pyplot as plt
import numpy as np
```

A. Amplitude envelope and loudness

In [ ]:
```
audio_file = '/content/Audio_file.wav'
y, sr = librosa.load(audio_file)
```

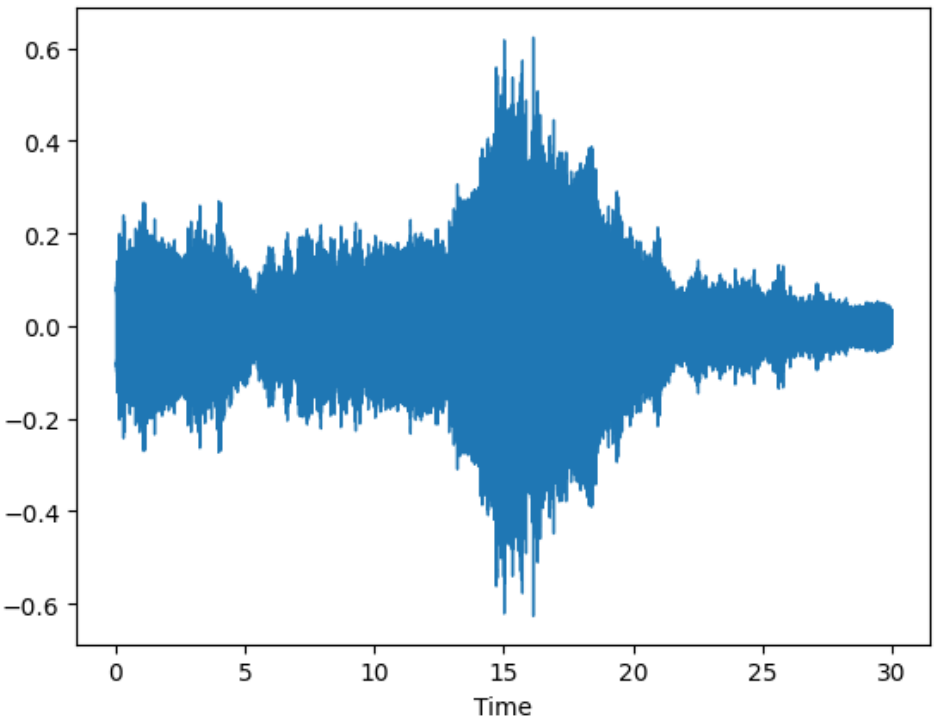
In [ ]:
```
ipd.Audio(audio_file)
```

Out[ ]:



In [ ]:
```
librosa.display.waveshow(y, sr=sr)
```

Out[ ]:
```
<librosa.display.AdaptiveWaveplot at 0x7af4c1396230>
```

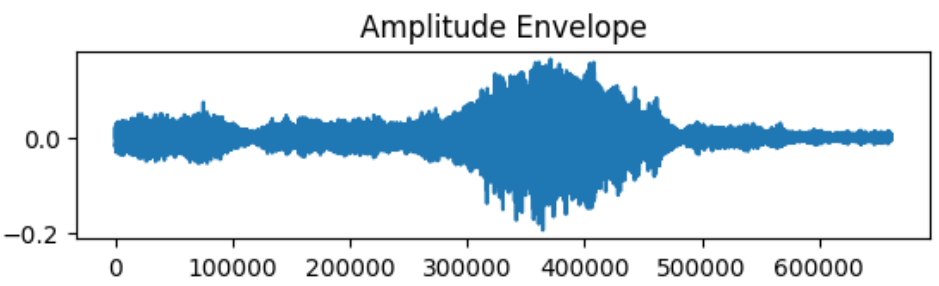


In [ ]:
```
amplitude_envelope = librosa.effects.preemphasis(y)
```

In [ ]:
```
num_frames = len(amplitude_envelope)
```

In [ ]:
```
plt.figure()
plt.subplot(3, 1, 3)
plt.plot(amplitude_envelope)
plt.title('Amplitude Envelope')
```

Out[ ]:
```
Text(0.5, 1.0, 'Amplitude Envelope')
```



Observation

1. Amplitude Envelope: The code extracts the amplitude envelope of the audio signal by dividing it into frames. The number of frames (i.e. 1290) in the amplitude envelope is determined by the frame length and hop length specified in the librosa.util.frame function. The amplitude envelope graph shows how the amplitude of the audio signal changes over time.

B. Spectral centroid

In [ ]:
```
# Compute the spectral centroid
spectral_centroids = librosa.feature.spectral_centroid(y=y, sr=sr)[0]
spectral_centroids[:30]
```

Out[ ]:
```
array([953.00240752, 986.00813886, 974.08758414, 863.35212412,
       779.21423659, 768.75590376, 783.18691236, 818.26104735,
       799.46717821, 805.08159365, 832.46909096, 817.45305478,
       790.01675607, 767.10431754, 775.57920934, 806.41221567,
       810.0239551 , 798.91724688, 813.70593599, 829.9125703 ,
       833.64296138, 906.13288255, 882.11627873, 878.7223385 ,
       902.84719925, 882.00253163, 928.67691976, 943.78866101,
       961.57645623, 985.08957792])
```

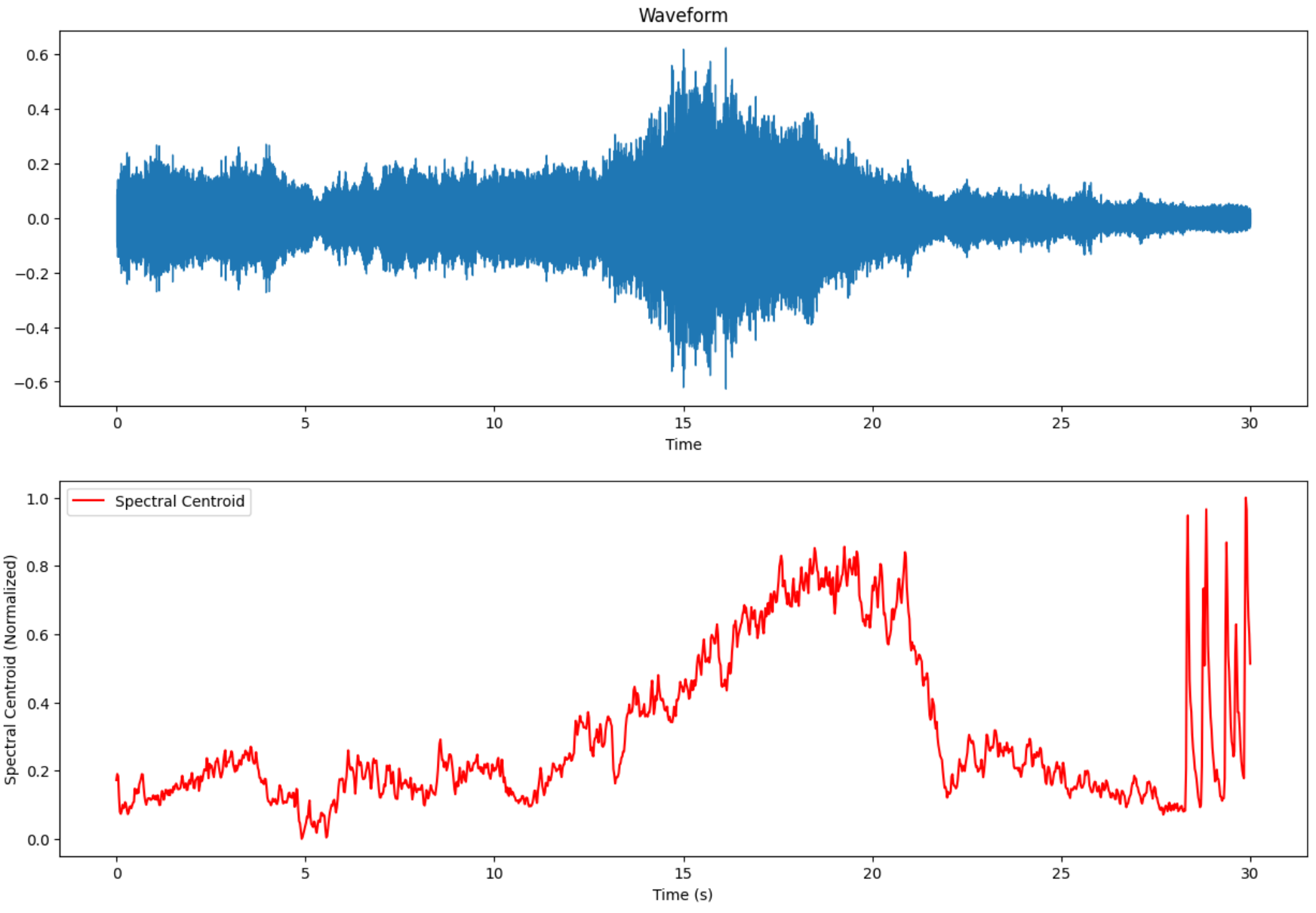
In [ ]:
```
# Normalize the spectral centroid values between 0 and 1
spectral_centroids_normalized = (spectral_centroids - np.min(spectral_centroids)) / (np.max(spectral_centroids) - np.min(spectral_centroids))
spectral_centroids_normalized[:30]
```

Out[ ]:
```
array([0.17238812, 0.19014614, 0.18373255, 0.12415372, 0.07888515,
       0.07325827, 0.08102256, 0.09989344, 0.08978181, 0.09280252,
       0.10753778, 0.09945872, 0.08469721, 0.07236967, 0.0769294 ,
       0.09351844, 0.09546165, 0.08948593, 0.09744267, 0.10616229,
       0.10816935, 0.14717099, 0.13424937, 0.13242334, 0.1454032 ,
       0.13418817, 0.15930032, 0.16743087, 0.17700121, 0.18965193])
```

In [ ]:
```
# Create a time array for visualization
time = np.linspace(0, len(y) / sr, len(spectral_centroids_normalized))
time[:30]
```

Out[ ]:
```
array([0.       , 0.0232378, 0.0464756, 0.0697134, 0.0929512, 0.116189 ,
       0.1394268, 0.1626646, 0.1859024, 0.2091402, 0.232378 , 0.2556158,
       0.2788536, 0.3020914, 0.3253292, 0.348567 , 0.3718048, 0.3950426,
       0.4182804, 0.4415182, 0.464756 , 0.4879938, 0.5112316, 0.5344694,
       0.5577072, 0.580945 , 0.6041828, 0.6274206, 0.6506584, 0.6738962])
```

In [ ]:
```
plt.figure(figsize=(15, 10))
plt.subplot(2, 1, 1)
librosa.display.waveshow(y, sr=sr)
plt.title('Waveform')
plt.subplot(2, 1, 2)
plt.plot(time, spectral_centroids_normalized, label='Spectral Centroid', color='r')
plt.ylabel('Spectral Centroid (Normalized)')
plt.xlabel('Time (s)')
plt.legend(loc='upper left')
plt.show()
```



Observation:

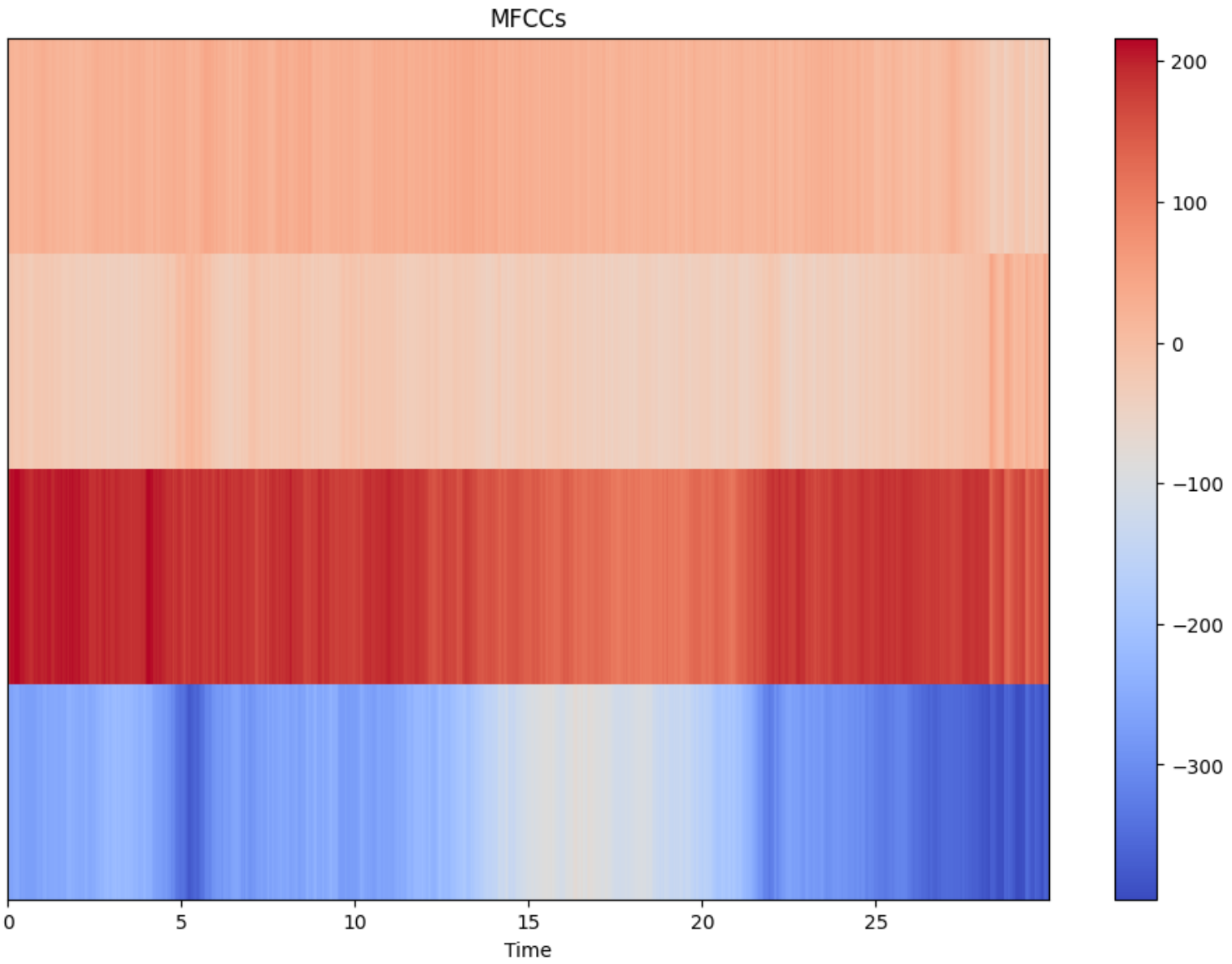
1. Spectral Centroids: The code computes the spectral centroids of the audio signal, which represent the center of mass of the spectrum. Spectral centroids provide information about the perceived brightness of sound over time. The graph shows the normalized spectral centroid values over time, helping to identify changes in the audio's spectral Characteristics.

C. MFCC

In [ ]:
```
mfccs = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=4)
print(mfccs)
```

```
[[-274.7724   -247.03934  -247.98555  ... -365.7285   -372.42188
  -372.88034 ]
 [ 199.77222   190.55592   192.33011  ...  135.26819   133.76526
   136.13632 ]
 [ -30.521141  -34.24643   -30.447289 ...   30.379436   24.217224
    20.17454 ]
 [  11.602886   14.324823   15.433358 ...  -31.331648  -29.230034
   -24.165249]]
```

In [ ]:
```
plt.figure(figsize=(12, 8))
librosa.display.specshow(mfccs, x_axis='time')
plt.colorbar()
plt.title('MFCCs')
plt.show()
```



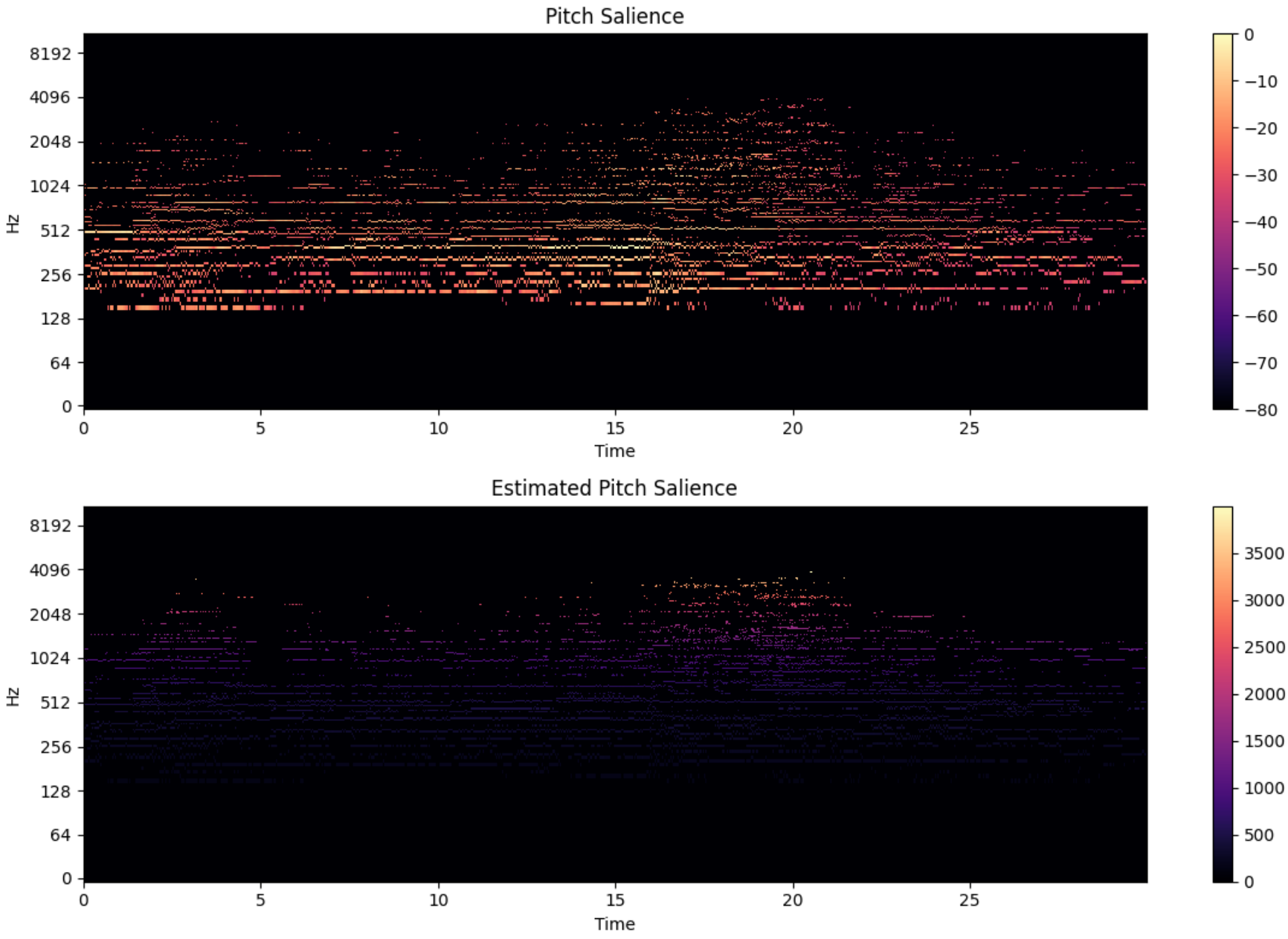
Observation

1. MFCC (Mel-Frequency Cepstral Coefficients): MFCCs are extracted from the audio signal, which are coefficients that capture the spectral characteristics of the audio. These coefficients are commonly used in audio and speech processing tasks. The visualization of MFCCs as a spectrogram-like graph helps in understanding the audio's spectral content, including important features for audio analysis.

D. Pitch salience

In [ ]:
```
# Perform Pitch Scaling
pitches, magnitudes = librosa.piptrack(y=y, sr=sr)
```

In [ ]:
```
# Display Output graph
plt.figure(figsize=(12, 8))
plt.subplot(2, 1, 1)
librosa.display.specshow(librosa.amplitude_to_db(magnitudes, ref=np.max), y_axis='log', x_axis='time')
plt.title('Pitch Salience')
plt.colorbar()
plt.subplot(2, 1, 2)
librosa.display.specshow(pitches, y_axis='log', x_axis='time')
plt.title('Estimated Pitch Salience')
plt.colorbar()
plt.tight_layout()
plt.show()
```



Observation:

Pitch Scaling: The code performs pitch scaling on the audio, allowing you to increase or decrease the perceived pitch. The pitch factor is adjusted to control the degree of pitch scaling. The graphs display the original and pitch-scaled audio, showing the effects of pitch scaling on the waveform

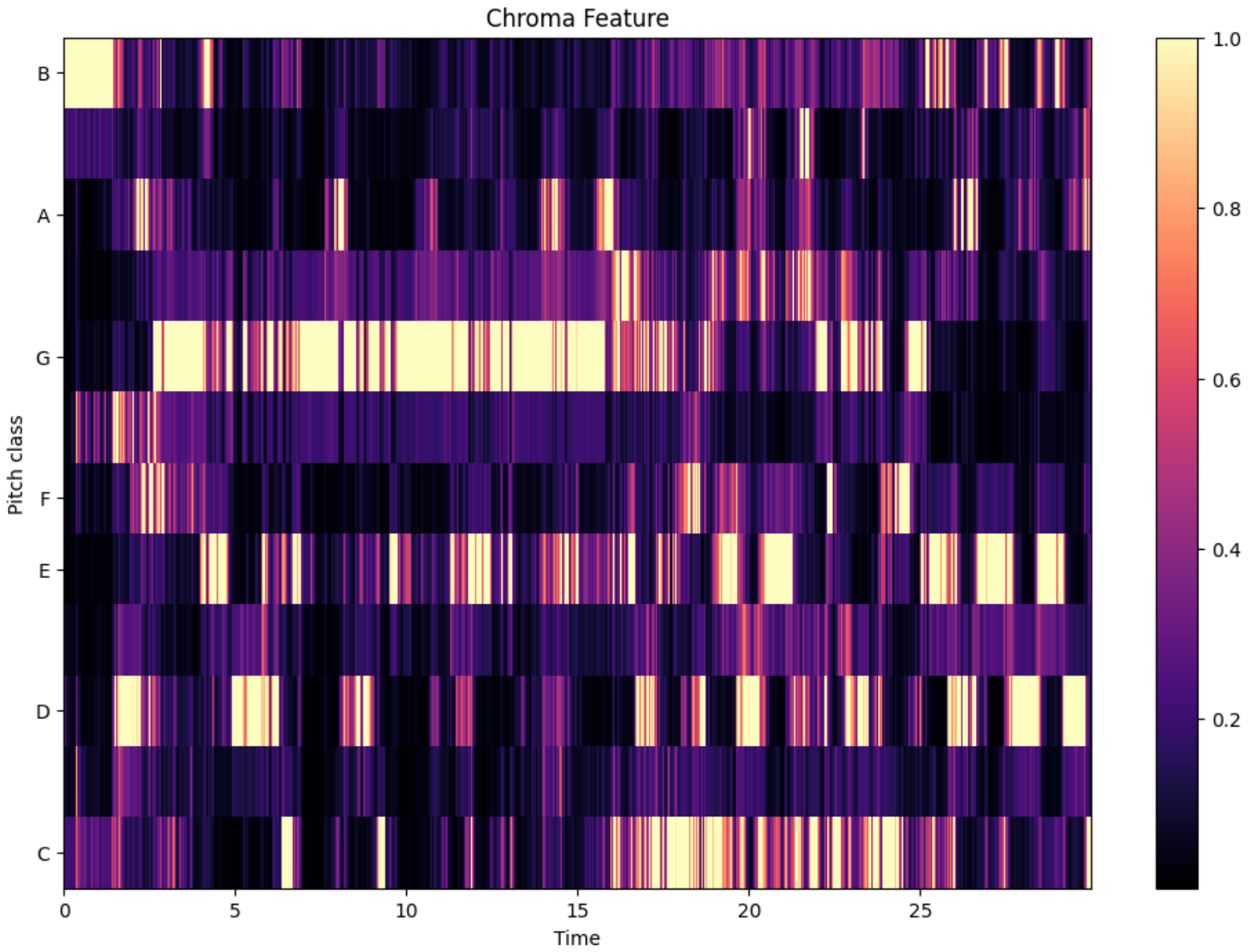
---

E. Chroma

In [ ]:
```
chroma = librosa.feature.chroma_stft(y=y, sr=sr)
print(chroma[:30])
```

```
[[0.3748383  0.2850431  0.23942405 ... 1.         1.         1.        ]
 [0.22583853 0.16383454 0.1030711  ... 0.18677554 0.13008022 0.11525967]
 [0.6439057  0.41414827 0.25331077 ... 0.264069   0.19289418 0.10033013]
 ...
 [0.11273479 0.04331427 0.03686681 ... 0.88813144 0.8336089  0.3830379 ]
 [0.3084843  0.23125449 0.23313238 ... 0.55701536 0.58833814 0.34212232]
 [1.         1.         1.         ... 0.43472394 0.5627108  0.5757672 ]]
```

In [ ]:
```
plt.figure(figsize=(12, 8))
librosa.display.specshow(chroma, y_axis='chroma', x_axis='time')
plt.colorbar()
plt.title('Chroma Feature')
plt.show()
```



Observation:

1. Chroma Feature: The code extracts the chroma feature from the audio signal, which represents the pitch class content. Chroma features provide information about the distribution of musical notes in the audio. The graph visualizes the chroma feature over time, allowing for insights into the tonal content and harmonic structure of the audio.
