Documentation for Audio Analysis with Librosa

Overview:

This Python script leverages the **Librosa** library to process and analyze an audio file (in this case, a music file). It performs various operations like:

- Loading the audio file.
- Generating a Mel Spectrogram.
- Calculating the Chromagram.
- Detecting Onset Strength and Tempo.
- Separating harmonic and percussive components of the audio.

The output includes visualizations such as the Mel spectrogram and Chromagram, along with the tempo and beat information extracted from the audio.

Libraries Used:

1. **Google Colab** (`google.colab`):
    - Used to mount Google Drive to access datasets stored on Google Drive.
2. **Librosa**:
    - A Python package for music and audio analysis. It provides functions for loading audio files, generating various audio features, and performing time-domain and frequency-domain analysis.
3. **Matplotlib**:
    - A Python library used for plotting and visualizing data, including waveforms, spectrograms, and other audio features.
4. **IPython.display**:
    - Used to display audio and other multimedia in Jupyter notebooks and Google Colab.

Code Explanation:

Mounting Google Drive:

```
from google.colab import drive
drive.mount('/content/drive')
```

1.
    - This code mounts Google Drive to the Colab environment so that files stored in Google Drive can be accessed. The user is prompted to authenticate.

Installing Librosa:

```
!pip install librosa==0.6.0
```

2.
    - This installs a specific version (0.6.0) of the **Librosa** library required for audio processing.

Importing Libraries:

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.style as ms
import librosa.display
ms.use('seaborn-muted')
%matplotlib inline
import IPython.display as ipd
import librosa
```

3.
    - Various libraries are imported:
        - **NumPy** for numerical computations.
        - **Matplotlib** for plotting.
        - **Librosa** for audio processing.
        - **IPython.display** for audio display.

Loading the Audio File:

```
data, sr = librosa.load('/content/drive/MyDrive/Datasets/Music.mp3')
print(data.shape)
```

4.

- The audio file `Music.mp3` from Google Drive is loaded using `librosa.load()`, which returns the audio data (`data`) and the sample rate (`sr`).

- `print(data.shape)` outputs the dimensions of the audio data.

Mel Spectrogram:

```
 S = librosa.feature.melspectrogram(data,sr)
log_S = librosa.power_to_db(S,ref = np.max)
plt.plot(log_S)
plt.show()
```

5.
   - A Mel spectrogram is computed using `librosa.feature.melspectrogram()`.

   - The spectrogram is then converted to decibels using `librosa.power_to_db()`.

   - The resulting log Mel spectrogram is plotted using `matplotlib`.

Chromagram:

```
 chromagram = librosa.feature.chroma_stft(y = data,sr = sr)
plt.figure(figsize = (15,5))
librosa.display.specshow(chromagram,x_axis = 'time',y_axis = 'chroma')
```

6.
   - A Chromagram is computed using `librosa.feature.chroma_stft()`, which represents the 12 different pitch classes in the audio signal.

   - The Chromagram is displayed as a heatmap using `librosa.display.specshow()`.

Onset Strength and Tempo Detection:

```
 onset_env = librosa.onset.onset_strength(data,sr=sr)
tempo = librosa.beat.tempo(onset_env,sr=sr)
```

7.
   - The **Onset Strength** is calculated with `librosa.onset.onset_strength()`, which detects changes in the audio

signal that may indicate a musical onset (like a beat).

- **Tempo** is estimated using `librosa.beat.tempo()`, which computes the beats per minute (BPM) from the onset envelope.

Harmonic and Percussive Separation:

```
y_harmonic, y_percussive = librosa.effects.hpss(data)
```

8.
   - The **Harmonic/Percussive Source Separation** (HPSS) method is applied to the audio signal, which separates the harmonic (melodic) and percussive (rhythmic) components of the audio.

Beat Tracking:

```
tempo, beats = librosa.beat.beat_track(y = y_percussive,sr=sr)
print(tempo)
print(beats)
```

9.
   - The `librosa.beat.beat_track()` function is used to track the beats in the percussive component of the audio. It returns:
     - **Tempo** (beats per minute).
     - **Beats** (indices of the detected beats).

Outputs:

- **Mel Spectrogram Plot**: Visual representation of the Mel frequency spectrogram in decibels.
- **Chromagram Plot**: Visual representation of the chromagram over time.
- **Tempo**: The estimated beats per minute (BPM) of the music.
- **Beats**: A list of the indices where the beats occur in the audio.

Example Output:

- For the `Music.mp3` file, the tempo might be calculated as `120.0` beats per minute, with corresponding beat positions listed.

Notes:

- The audio file used should be available on Google Drive and accessible to Colab.
- The version of Librosa is fixed to `0.6.0` due to compatibility reasons.