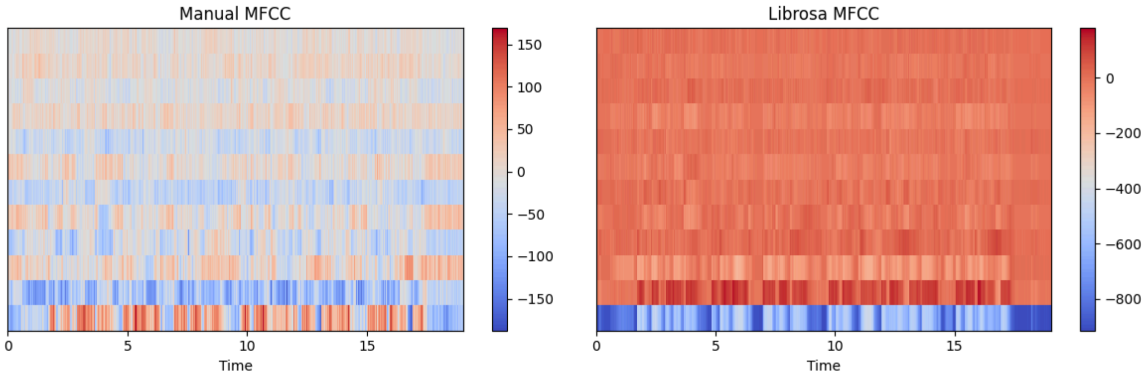
```
pip install librosa numpy matplotlib
```

```
Requirement already satisfied: librosa in /usr/local/lib/python3.11/dist-packages (0.11.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (2.0.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (3.10.0)
Requirement already satisfied: audioread>=2.1.9 in /usr/local/lib/python3.11/dist-packages (from librosa) (3.0.1)
Requirement already satisfied: numba>=0.51.0 in /usr/local/lib/python3.11/dist-packages (from librosa) (0.60.0)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.11/dist-packages (from librosa) (1.14.1)
Requirement already satisfied: scikit-learn>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from librosa) (1.6
Requirement already satisfied: joblib>=1.0 in /usr/local/lib/python3.11/dist-packages (from librosa) (1.4.2)
Requirement already satisfied: decorator>=4.3.0 in /usr/local/lib/python3.11/dist-packages (from librosa) (4.4.2)
Requirement already satisfied: soundfile>=0.12.1 in /usr/local/lib/python3.11/dist-packages (from librosa) (0.13.
Requirement already satisfied: pooch>=1.1 in /usr/local/lib/python3.11/dist-packages (from librosa) (1.8.2)
Requirement already satisfied: soxr>=0.3.2 in /usr/local/lib/python3.11/dist-packages (from librosa) (0.5.0.post1
Requirement already satisfied: typing_extensions>=4.1.1 in /usr/local/lib/python3.11/dist-packages (from librosa)
Requirement already satisfied: lazy_loader>=0.1 in /usr/local/lib/python3.11/dist-packages (from librosa) (0.4)
Requirement already satisfied: msgpack>=1.0 in /usr/local/lib/python3.11/dist-packages (from librosa) (1.1.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.3
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (4.
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (24.2
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (3.2
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.11/dist-packages (from matplotlib)
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in /usr/local/lib/python3.11/dist-packages (from numba>
Requirement already satisfied: platformdirs>=2.5.0 in /usr/local/lib/python3.11/dist-packages (from pooch>=1.1->l
Requirement already satisfied: requests>=2.19.0 in /usr/local/lib/python3.11/dist-packages (from pooch>=1.1->libr
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.7->ma
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn
Requirement already satisfied: cffi>=1.0 in /usr/local/lib/python3.11/dist-packages (from soundfile>=0.12.1->libr
Requirement already satisfied: pycparser in /usr/local/lib/python3.11/dist-packages (from cffi>=1.0->soundfile>=0
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests>=2.19.0->po
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests>=2.19
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests>=2.19
```

```
import librosa
import numpy as np
import matplotlib.pyplot as plt

# Load an audio file from LibriSpeech (replace with your file path)
audio_path = "/content/common_voice_en_41227113.mp3"
y, sr = librosa.load(audio_path, sr=None)  # sr = sample rate (16 kHz for LibriSpeech)

# Pre-emphasis (optional, but improves performance)
pre_emphasis = 0.97
y = np.append(y[0], y[1:] - pre_emphasis * y[:-1])


frame_size = 0.025  # 25 ms frames
frame_stride = 0.01  # 10 ms stride (overlap of 15 ms)

# Convert to samples
frame_length = int(round(frame_size * sr))
frame_step = int(round(frame_stride * sr))

# Pad the signal to ensure all frames have equal length
signal_length = len(y)
num_frames = int(np.ceil(float(np.abs(signal_length - frame_length)) / frame_step))
pad_signal_length = num_frames * frame_step + frame_length
pad_signal = np.append(y, np.zeros((pad_signal_length - signal_length)))

# Split into frames
indices = np.tile(np.arange(0, frame_length), (num_frames, 1)) + \
          np.tile(np.arange(0, num_frames * frame_step, frame_step), (frame_length, 1)).T
frames = pad_signal[indices.astype(np.int32, copy=False)]

# Apply Hamming window
# Apply hamming window
frames *= np.hamming(frame_length)


NFFT = 512  # Number of FFT points
mag_frames = np.absolute(np.fft.rfft(frames, NFFT))  # Magnitude of FFT
pow_frames = ((1.0 / NFFT) * (mag_frames ** 2))  # Power spectrum


nfilt = 40  # Number of Mel filters
low_freq_mel = 0
high_freq_mel = 2595 * np.log10(1 + (sr / 2) / 700)  # Convert Hz to Mel

# Create Mel points
mel_points = np.linspace(low_freq_mel, high_freq_mel, nfilt + 2)
hz_points = 700 * (10 ** (mel_points / 2595) - 1)  # Convert Mel back to Hz

# Create filter banks
bin = np.floor((NFFT + 1) * hz_points / sr).astype(int)
fbank = np.zeros((nfilt, int(np.floor(NFFT / 2 + 1))))

for m in range(1, nfilt + 1):
    f_m_minus = bin[m - 1]  # left
    f_m = bin[m]            # center
    f_m_plus = bin[m + 1]   # right

    for k in range(f_m_minus, f_m):
        fbank[m - 1, k] = (k - bin[m - 1]) / (bin[m] - bin[m - 1])
    for k in range(f_m, f_m_plus):
        fbank[m - 1, k] = (bin[m + 1] - k) / (bin[m + 1] - bin[m])

# Apply filter bank to power spectrum
filter_banks = np.dot(pow_frames, fbank.T)
filter_banks = np.where(filter_banks == 0, np.finfo(float).eps, filter_banks)  # Avoid log(0)
filter_banks = 20 * np.log10(filter_banks)  # dB


from scipy.fftpack import dct
num_ceps = 12
mfcc = np.dot(dct(filter_banks, type=2, axis=1, norm='ortho')[:, 1:num_ceps + 1], np.diag(np.ones(num_ceps)))


# Librosa's built-in MFCC function (simpler and optimized)
mfcc_librosa = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=12, n_fft=NFFT, hop_length=frame_step)
```

```
/usr/local/lib/python3.11/dist-packages/librosa/feature/spectral.py:2148: UserWarning: Empty filters detected in
  mel_basis = filters.mel(sr=sr, n_fft=n_fft, **kwargs)
```

```
plt.figure(figsize=(12, 4))
plt.subplot(1, 2, 1)
librosa.display.specshow(mfcc.T, sr=sr, x_axis='time')
plt.title('Manual MFCC')
plt.colorbar()

plt.subplot(1, 2, 2)
librosa.display.specshow(mfcc_librosa, sr=sr, x_axis='time')
plt.title('Librosa MFCC')
plt.colorbar()

plt.tight_layout()
plt.show()
```



```
# Save MFCCs to a file (e.g., for machine learning)
np.save("mfcc_features.npy", mfcc_librosa)
```