Edge Intelligence

Lab 1

M Rahul – 25MML0011

Audio Dataset

Error :-

```
[2]
0s
# --- STEP 8: PLOT SPECTROGRAM ---
fig, ax = plt.subplots(figsize=(10, 5))
img = librosa.display.specshow(S_db, x_axis='time', y_axis='log', ax=ax)
ax.set_title('Spectrogram Example', fontsize=20)
fig.colorbar(img, ax=ax, format=f'%0.2f')
plt.show()
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
/tmp/ipython-input-3254073883.py in <cell line: 0>()
      1 import kagglehub
----> 2 import pandas as pd
      3 import numpy as np
      4 import matplotlib.pylab as plt
      5 import seaborn as sns

                                4 frames
/usr/local/lib/python3.12/dist-packages/numpy/__init__.py in __getattr__(attr)

ModuleNotFoundError: No module named 'numpy.random'

---------------------------------------------------------------------------
NOTE: If your import is failing due to a missing package, you can
manually install dependencies using either !pip or !apt.

To view examples of installing some common dependencies, click the
"Open Examples" button below.
---------------------------------------------------------------------------
```

OPEN EXAMPLES

Next steps: Explain error

Corrected Code :-

```
[3]
6s
pip install librosa kaggle
```

```
Requirement already satisfied: librosa in /usr/local/lib/python3.12/dist-packages (0.11.0)
Requirement already satisfied: kaggle in /usr/local/lib/python3.12/dist-packages (1.7.4.5)
Requirement already satisfied: audioread>=2.1.9 in /usr/local/lib/python3.12/dist-packages (from librosa) (3.1.0)
Requirement already satisfied: numba>=0.51.0 in /usr/local/lib/python3.12/dist-packages (from librosa) (0.60.0)
Requirement already satisfied: numpy>=1.22.3 in /usr/local/lib/python3.12/dist-packages (from librosa) (2.0.2)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.12/dist-packages (from librosa) (1.16.3)
Requirement already satisfied: scikit-learn>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from librosa) (1.6.1)
Requirement already satisfied: joblib>=1.0 in /usr/local/lib/python3.12/dist-packages (from librosa) (1.5.2)
Requirement already satisfied: decorator>=4.3.0 in /usr/local/lib/python3.12/dist-packages (from librosa) (4.4.2)
Requirement already satisfied: soundfile>=0.12.1 in /usr/local/lib/python3.12/dist-packages (from librosa) (0.13.1)
Requirement already satisfied: pooch>=1.1 in /usr/local/lib/python3.12/dist-packages (from librosa) (1.8.2)
Requirement already satisfied: soxr>=0.3.2 in /usr/local/lib/python3.12/dist-packages (from librosa) (1.0.0)
Requirement already satisfied: typing_extensions>=4.1.1 in /usr/local/lib/python3.12/dist-packages (from librosa) (4.15.0)
Requirement already satisfied: lazy_loader>=0.1 in /usr/local/lib/python3.12/dist-packages (from librosa) (0.4)
Requirement already satisfied: msgpack>=1.0 in /usr/local/lib/python3.12/dist-packages (from librosa) (1.1.2)
Requirement already satisfied: bleach in /usr/local/lib/python3.12/dist-packages (from kaggle) (6.3.0)
Requirement already satisfied: certifi>=14.05.14 in /usr/local/lib/python3.12/dist-packages (from kaggle) (2025.11.12)
Requirement already satisfied: charset-normalizer in /usr/local/lib/python3.12/dist-packages (from kaggle) (3.4.4)
Requirement already satisfied: idna in /usr/local/lib/python3.12/dist-packages (from kaggle) (3.11)
Requirement already satisfied: protobuf in /usr/local/lib/python3.12/dist-packages (from kaggle) (5.29.5)
Requirement already satisfied: python-dateutil>=2.5.3 in /usr/local/lib/python3.12/dist-packages (from kaggle) (2.9.0.post0)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.12/dist-packages (from kaggle) (8.0.4)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from kaggle) (2.32.4)
Requirement already satisfied: setuptools>=21.0.0 in /usr/local/lib/python3.12/dist-packages (from kaggle) (75.2.0)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.12/dist-packages (from kaggle) (1.17.0)
Requirement already satisfied: text-unidecode in /usr/local/lib/python3.12/dist-packages (from kaggle) (1.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from kaggle) (4.67.1)
Requirement already satisfied: urllib3>=1.15.1 in /usr/local/lib/python3.12/dist-packages (from kaggle) (2.5.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.12/dist-packages (from kaggle) (0.5.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.12/dist-packages (from lazy_loader>=0.1->librosa) (25.0)
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in /usr/local/lib/python3.12/dist-packages (from numba>=0.51.0->librosa) (0.43.0)
Requirement already satisfied: platformdirs>=2.5.0 in /usr/local/lib/python3.12/dist-packages (from pooch>=1.1->librosa) (4.5.1)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn>=1.1.0->librosa) (3.6.0)
Requirement already satisfied: cffi>=1.0 in /usr/local/lib/python3.12/dist-packages (from soundfile>=0.12.1->librosa) (2.0.0)
Requirement already satisfied: pycparser in /usr/local/lib/python3.12/dist-packages (from cffi>=1.0->soundfile>=0.12.1->librosa) (2.23)
```

```
# kagglehub.dataset_download() → Downloads dataset from Kaggle automatically.

# glob() → Searches and returns all file paths matching a pattern (e.g., all .wav files).

# librosa.load() → Loads an audio file and returns:
#    y  = waveform (audio samples)
#    sr = sample rate (samples per second)

# pd.Series().plot() → Plots the audio waveform as a graph.

# librosa.effects.trim() → Removes silence from start and end of the audio clip.

# librosa.stft() → Short-Time Fourier Transform:
#     Converts audio from time domain to frequency domain (for spectrogram).

# librosa.amplitude_to_db() → Converts amplitude values into decibels (makes spectrogram readable).

# librosa.display.specshow() → Displays the spectrogram as an image.

# plt.show() → Shows the plot/graph.

# MFCCs → Capture voice tone and speech shape (most important for speech/emotion recognition).

# Chroma → Represents harmonic and pitch information of the voice.

# Mel Spectrogram → Shows energy in different frequency bands (used in deep learning).

# Zero Crossing Rate (ZCR) → Measures noisiness or sharpness of the voice.

# Spectral Centroid → Indicates brightness of the sound (higher = sharper tone).
```

[4] ✓ 0s

```
import kagglehub
import pandas as pd
import numpy as np
import matplotlib.pylab as plt
import seaborn as sns
from glob import glob
import librosa
import librosa.display
from itertools import cycle


# --- STEP 1: SETUP AND STYLE ---
# Set the aesthetic style for the plots
sns.set_theme(style="white", palette=None)
color_pal = plt.rcParams["axes.prop_cycle"].by_key()["color"]
color_cycle = cycle(plt.rcParams["axes.prop_cycle"].by_key()["color"])
```

[5] ✓ 11s

```
# --- STEP 2: DOWNLOAD AND LOCATE DATA ---
# Download the dataset
path = kagglehub.dataset_download("uwrfkaggler/ravdess-emotional-speech-audio")
print("Path to dataset files:", path)

# Find all .wav files in the downloaded directory (Recursive search)
audio_files = glob(f"{path}/**/*.wav", recursive=True)
```

```
Using Colab cache for faster access to the 'ravdess-emotional-speech-audio' dataset.
Path to dataset files: /kaggle/input/ravdess-emotional-speech-audio
```

[7]

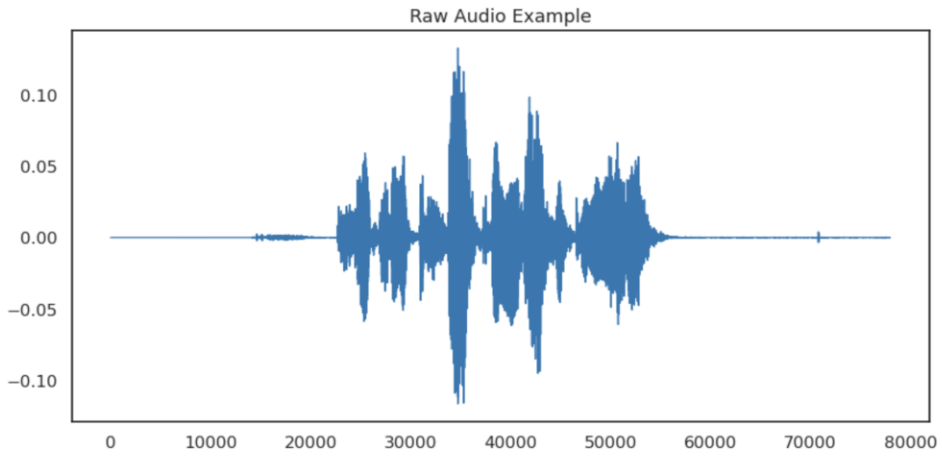
```
# --- STEP 3: LOAD AUDIO ---
# Load the first audio file found
# y = raw audio data (waveform), sr = sample rate
y, sr = librosa.load(audio_files[0])

print(f'y: {y[:10]}')      # Print first 10 numbers of the audio data
print(f'shape y: {y.shape}') # Print the total number of data points
print(f'sr: {sr}')         # Print the sample rate (Hz)
```

```
y: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
shape y: (77988,)
sr: 22050
```

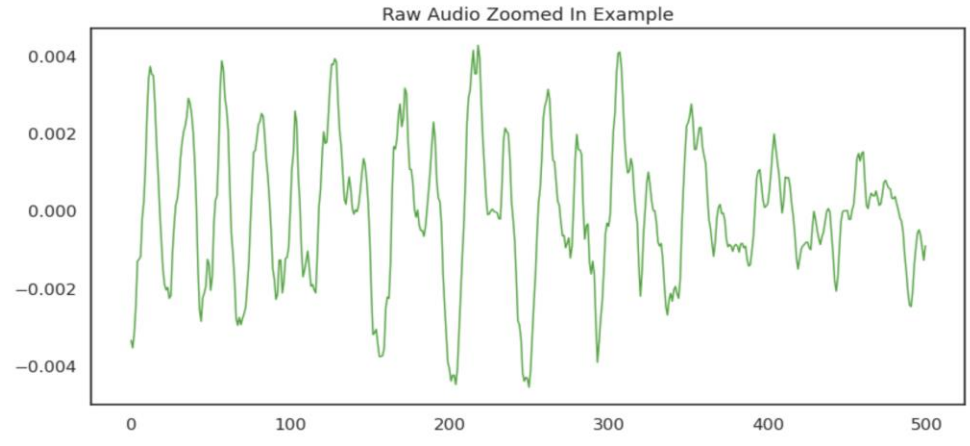
[8]
Os

```
# --- STEP 4: PLOT RAW AUDIO ---
pd.Series(y).plot(figsize=(10, 5), lw=1, title='Raw Audio Example', color=color_pal[0])
plt.show()
```



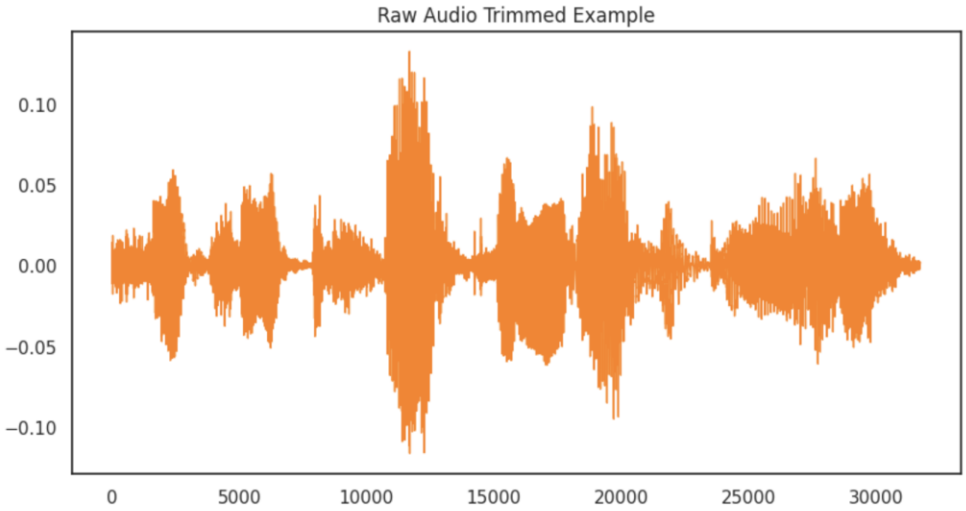
[9]
Os

```
# --- STEP 5: PLOT ZOOMED AUDIO ---
# Looking at a specific small slice of time
pd.Series(y[30000:30500]).plot(figsize=(10, 5),lw=1, title='Raw Audio Zoomed In Example', color=color_pal[2])
plt.show()
```



[10] ✓ 3s

```
# --- STEP 6: TRIM SILENCE ---
# Remove silence from the beginning and end of the clip
y_trimmed, _ = librosa.effects.trim(y, top_db=20)
pd.Series(y_trimmed).plot(figsize=(10, 5),lw=1, title='Raw Audio Trimmed Example',color=color_pal[1])
plt.show()
```



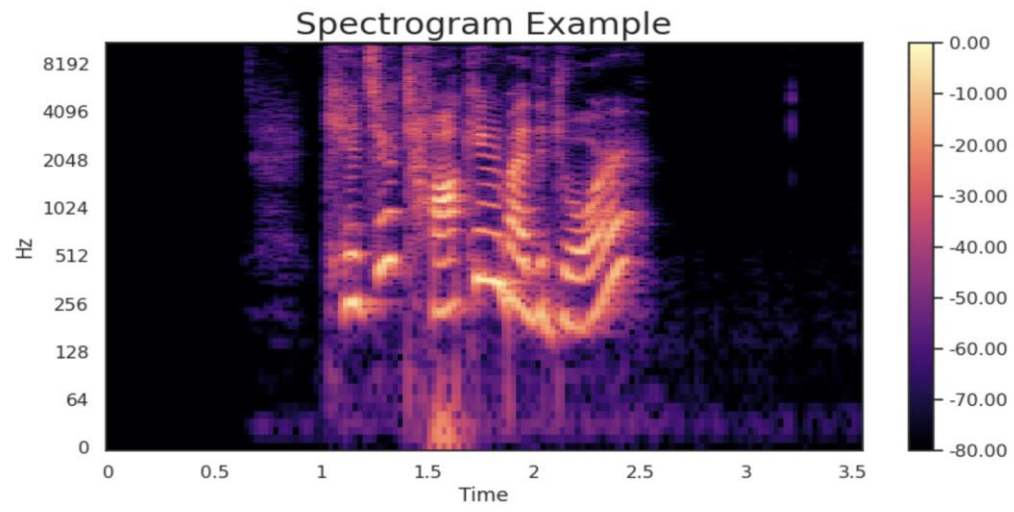
[11] ✓ 0s

```
# --- STEP 7: CREATE SPECTROGRAM ---
# Transform audio from Time Domain -> Frequency Domain
D = librosa.stft(y)  # Short-time Fourier transform
S_db = librosa.amplitude_to_db(np.abs(D), ref=np.max) # Convert to Decibels
print(f'Spectrogram shape: {S_db.shape}')
```

```
Spectrogram shape: (1025, 153)
```

[12] ✓ 1s

```
# --- STEP 8: PLOT SPECTROGRAM ---
fig, ax = plt.subplots(figsize=(10, 5))
img = librosa.display.specshow(S_db, x_axis='time', y_axis='log', ax=ax)
ax.set_title('Spectrogram Example', fontsize=20)
fig.colorbar(img, ax=ax, format=f'%0.2f')
plt.show()
```



[13]
0s

```python
# --- EXTRA STEP: Extract useful audio features ---

# MFCCs
mfccs = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=13)
print("MFCCs shape:", mfccs.shape)

# Chroma
chroma = librosa.feature.chroma_stft(y=y, sr=sr)
print("Chroma shape:", chroma.shape)

# Mel Spectrogram
mel = librosa.feature.melspectrogram(y=y, sr=sr)
print("Mel Spectrogram shape:", mel.shape)

# Zero Crossing Rate
zcr = librosa.feature.zero_crossing_rate(y)
print("ZCR shape:", zcr.shape)

# Spectral Centroid
centroid = librosa.feature.spectral_centroid(y=y, sr=sr)
print("Spectral Centroid shape:", centroid.shape)
```

```
MFCCs shape: (13, 153)
Chroma shape: (12, 153)
Mel Spectrogram shape: (128, 153)
ZCR shape: (1, 153)
Spectral Centroid shape: (1, 153)
```