

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from pathlib import Path
from scipy.io import wavfile
import os.path
import IPython.display
import seaborn as sns
import librosa
import librosa.display
import soundfile

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import confusion_matrix

import tensorflow as tf
from tensorflow.keras import utils
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Conv1D, MaxPooling1D, Flatten, BatchNormalization
from keras import optimizers

import warnings

```

```
image_dir = Path('/content/Crema')
```

```
filepaths = list(image_dir.glob(r'**/*.wav'))
```

```
labels = list(map(lambda x: os.path.split(x)[1].split('_')[2], filepaths))
```

```
set(labels)
```

```
{'ANG', 'DIS', 'FEA', 'HAP', 'NEU', 'SAD'}
```

```
filepaths = pd.Series(filepaths, name='Filepath').astype(str)
labels = pd.Series(labels, name='Label')
```

```
audio_df = pd.concat([filepaths, labels], axis=1)
audio_df
```

```

Filepath  Label
0  /content/Crema/1080_IOM_ANG_XX.wav  ANG
1  /content/Crema/1082_TSI_SAD_XX.wav  SAD
2  /content/Crema/1070_ITH_HAP_XX.wav  HAP
3  /content/Crema/1075_IWW_FEA_XX.wav  FEA
4  /content/Crema/1077_DFA_ANG_XX.wav  ANG
...
2297 /content/Crema/1066_IOM_NEU_XX.wav  NEU
2298 /content/Crema/1075_MTI_ANG_XX.wav  ANG
2299 /content/Crema/1082_TSI_NEU_XX.wav  NEU
2300 /content/Crema/1090_DFA_ANG_XX.wav  ANG
2301 /content/Crema/1077_WSI_NEU_XX.wav  NEU

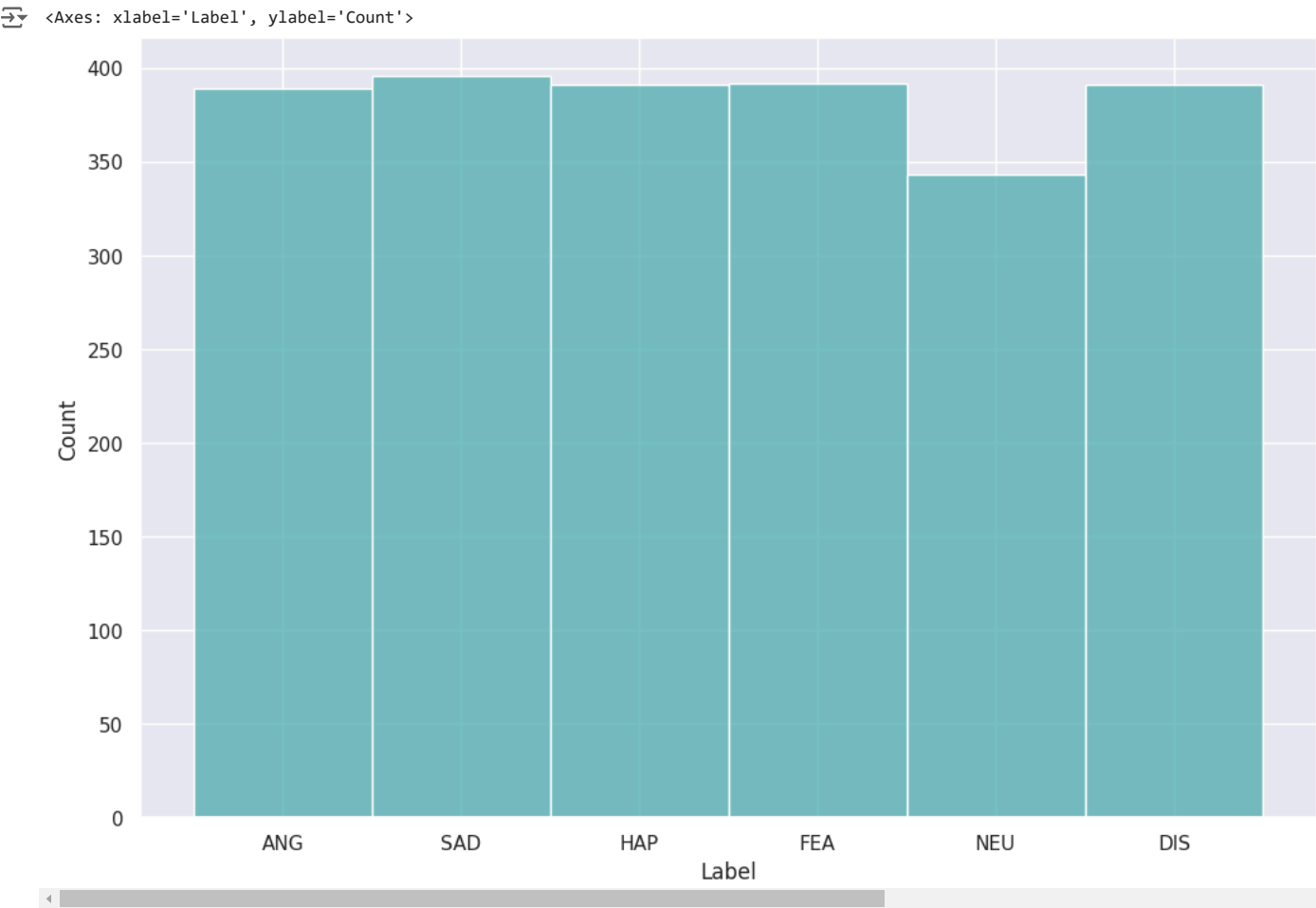
```

2302 rows × 2 columns

```

sns.set(rc={'figure.figsize':(12,8)})
sns.set_style('darkgrid')
sns.histplot(labels, color='#4FAEB0')

```



```
audio_arrays = []

for i in audio_df['Filepath']:
    x, sr = librosa.load(i, sr=44100)
    audio_arrays.append(x)

audio_df['Arrays'] = audio_arrays
```

audio_df

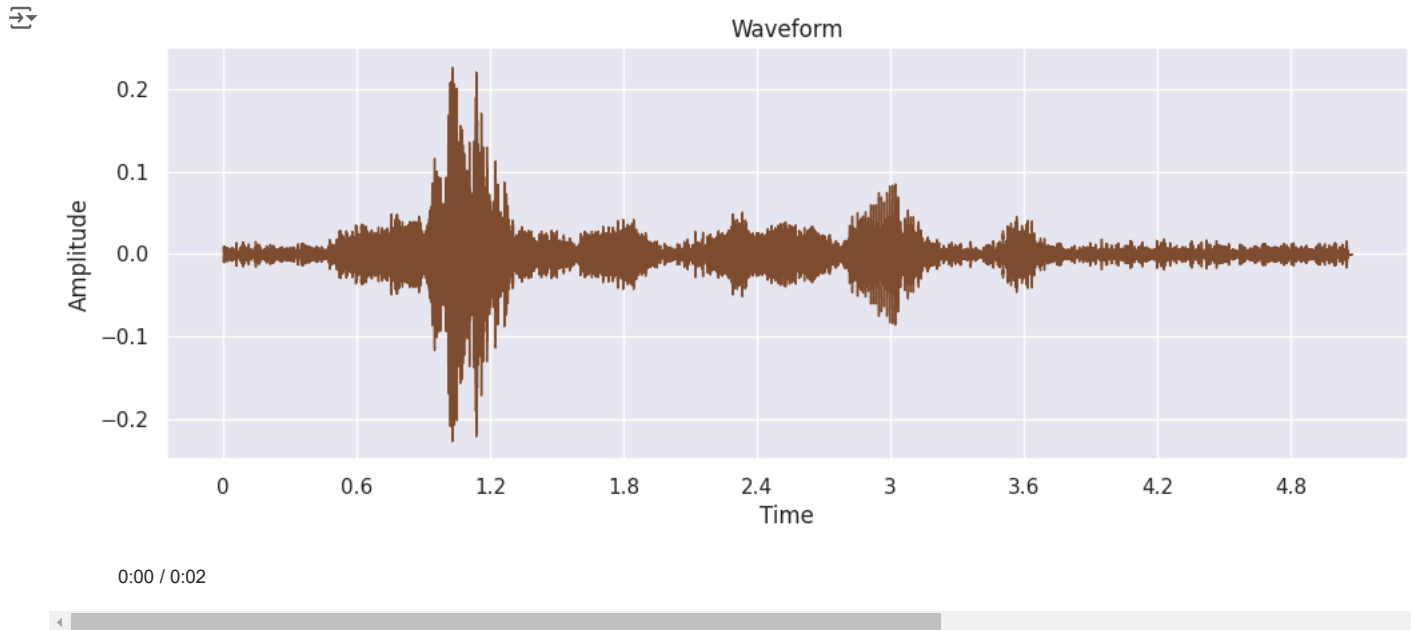
	Filepath	Label	Arrays
0	/content/Crema/1080_IOM_ANG_XX.wav	ANG	[0.010350021, 0.013110722, 0.0121039525, 0.011...
1	/content/Crema/1082_TSI_SAD_XX.wav	SAD	[-0.00081090245, -0.001021223, -0.000854598, -...
2	/content/Crema/1070_ITH_HAP_XX.wav	HAP	[-0.005582463, -0.0068380367, -0.006075979, -0...
3	/content/Crema/1075_IWW_FEA_XX.wav	FEA	[-1.5905773e-10, 3.2262687e-10, 1.6328676e-10,...
4	/content/Crema/1077_DFA_ANG_XX.wav	ANG	[-0.0016305591, -0.0021876423, -0.002074254, -...
...
2297	/content/Crema/1066_IOM_NEU_XX.wav	NEU	[0.012238709, 0.015352141, 0.013884371, 0.0126...
2298	/content/Crema/1075_MTI_ANG_XX.wav	ANG	[0.0023061726, 0.0028557023, 0.0024697906, 0.0...
2299	/content/Crema/1082_TSI_NEU_XX.wav	NEU	[0.0024152591, 0.0031653764, 0.003077399, 0.00...
2300	/content/Crema/1090_DFA_ANG_XX.wav	ANG	[0.0008279504, 0.0010322211, 0.00091686397, 0...
2301	/content/Crema/1077_WSI_NEU_XX.wav	NEU	[-0.009082937, -0.011823807, -0.011242002, -0...

2302 rows × 3 columns

```
disfile = audio_df[audio_df['Label'] == 'DIS']['Filepath']
disarray = audio_df[audio_df['Label'] == 'DIS']['Arrays']

plt.figure(figsize=(12, 4))
librosa.display.waveshow(disarray.iloc[0], color='#804E2D')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.title('Waveform')
plt.show()
```

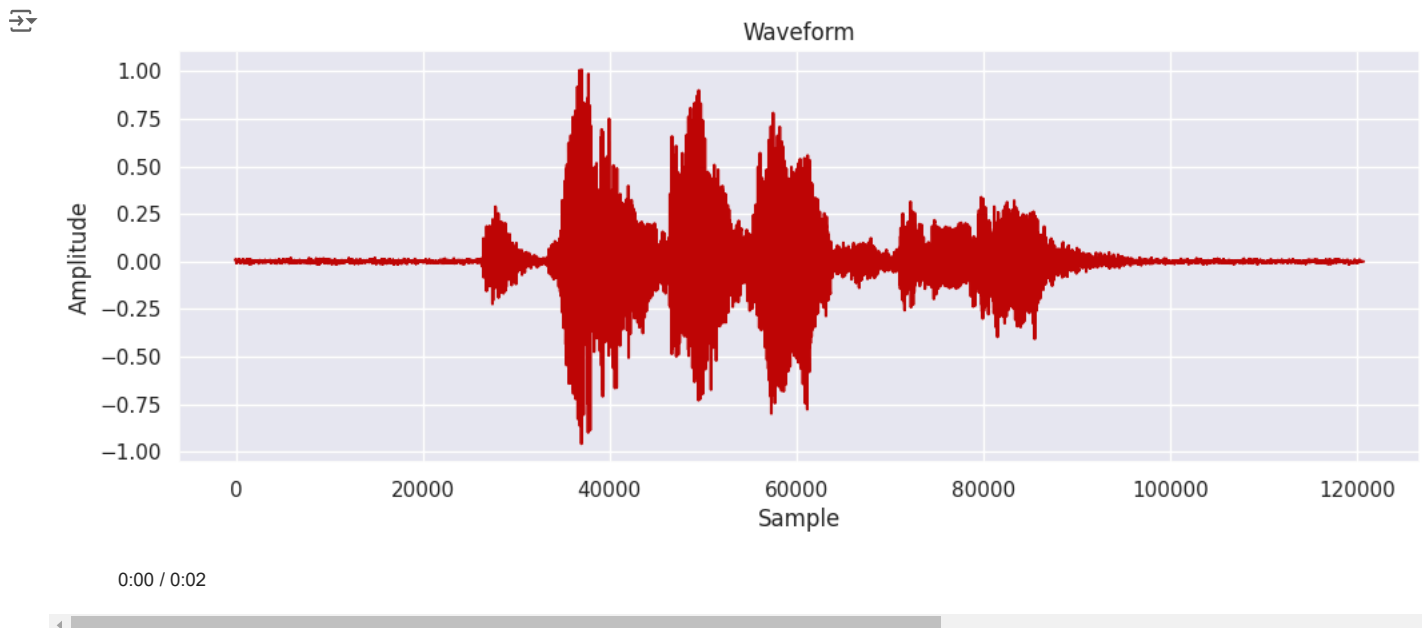
```
IPython.display.Audio(disfile.iloc[0])
```



```
angfile = audio_df[audio_df['Label'] == 'ANG']['Filepath']
angarray = audio_df[audio_df['Label'] == 'ANG']['Arrays']

plt.figure(figsize=(12, 4))
plt.plot(angarray.iloc[0], color='#C00808')
plt.xlabel('Sample')
plt.ylabel('Amplitude')
plt.title('Waveform')
plt.show()
```

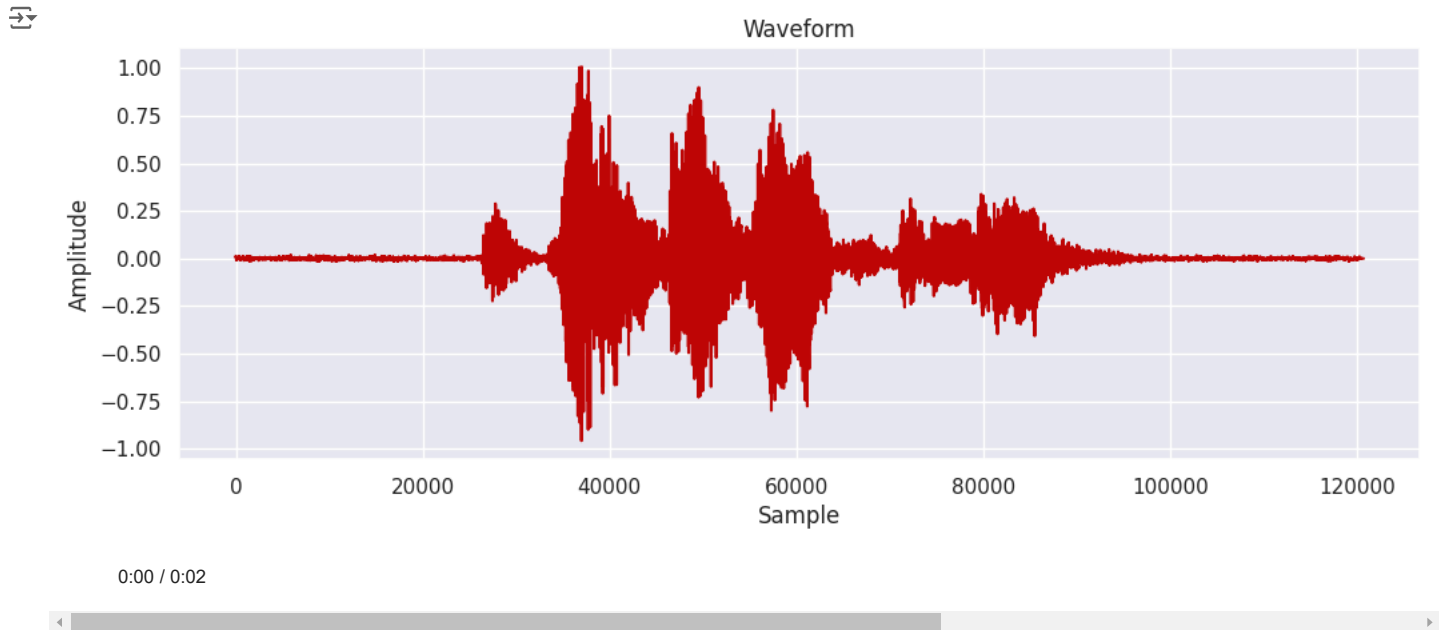
```
IPython.display.Audio(angfile.iloc[0])
```



```
angfile = audio_df[audio_df['Label'] == 'ANG']['Filepath']  
angarray = audio_df[audio_df['Label'] == 'ANG']['Arrays']
```

```
plt.figure(figsize=(12, 4))  
plt.plot(angarray.iloc[0], color='#C00808')  
plt.xlabel('Sample')  
plt.ylabel('Amplitude')  
plt.title('Waveform')  
plt.show()
```

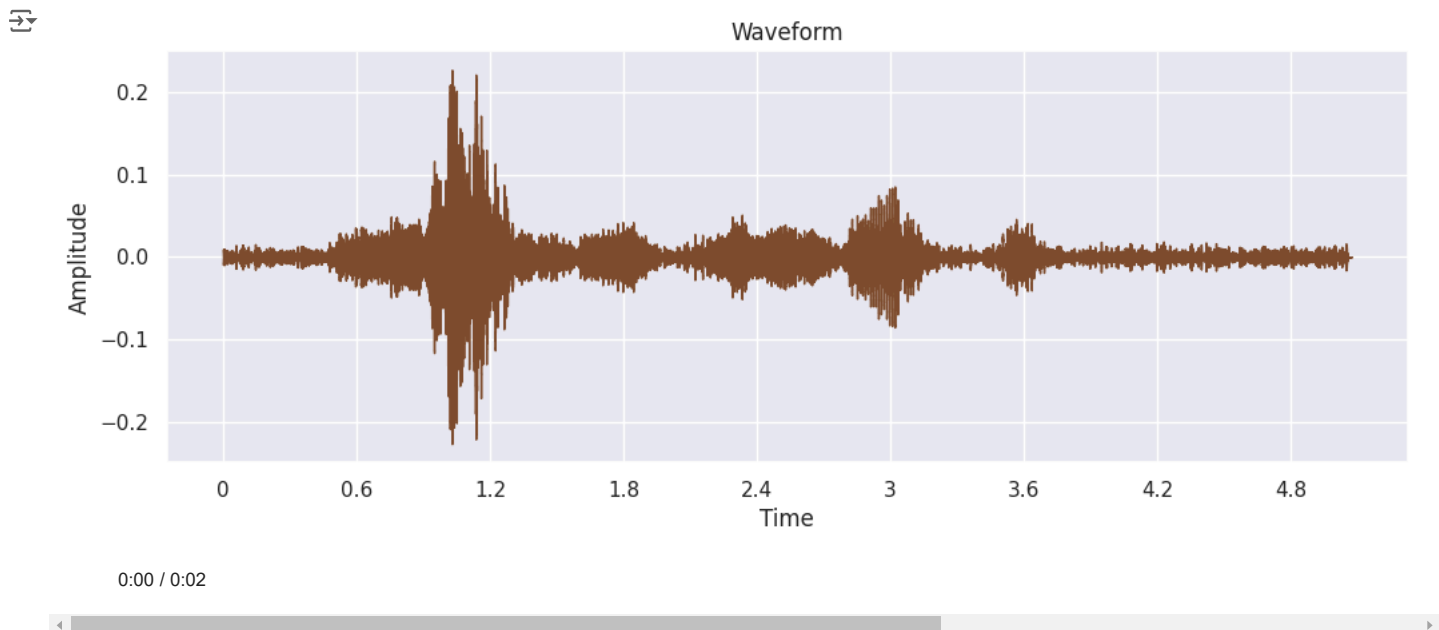
```
IPython.display.Audio(angfile.iloc[0])
```



```
disfile = audio_df[audio_df['Label'] == 'DIS']['Filepath']  
disarray = audio_df[audio_df['Label'] == 'DIS']['Arrays']
```

```
plt.figure(figsize=(12, 4))  
librosa.display.waveshow(disarray.iloc[0], color='#804E2D')  
plt.xlabel('Time')  
plt.ylabel('Amplitude')  
plt.title('Waveform')  
plt.show()
```

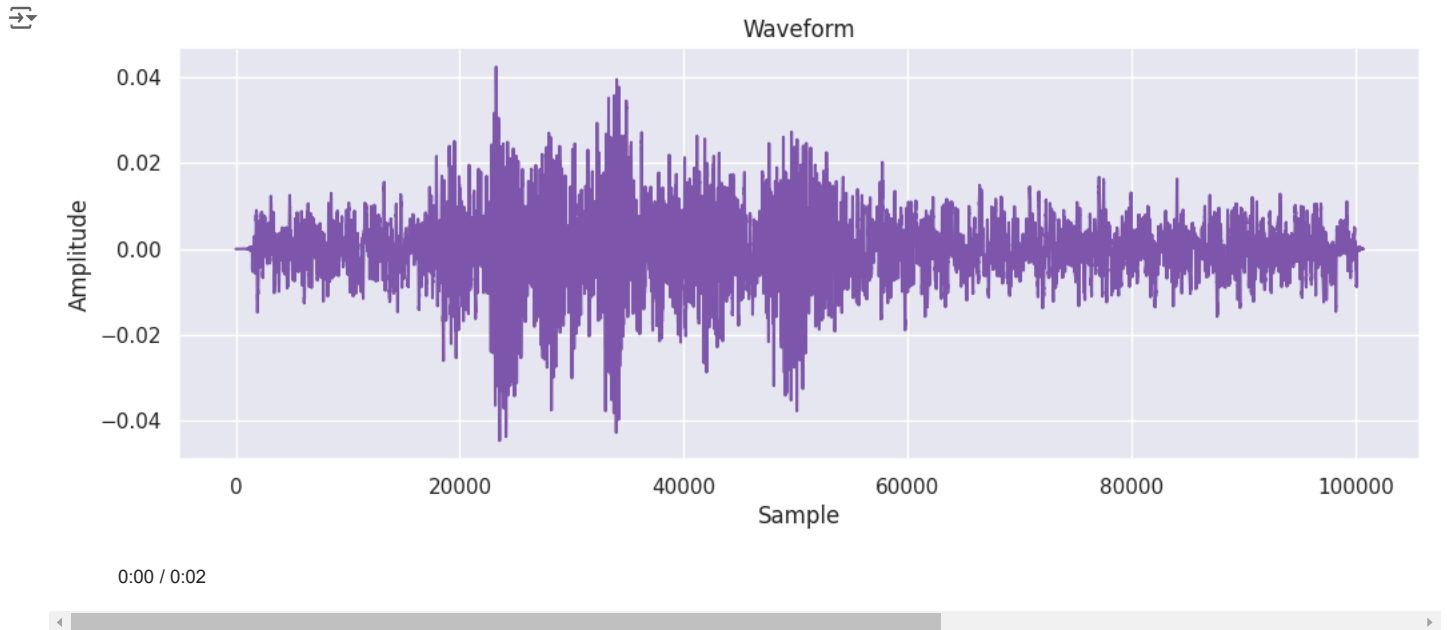
```
IPython.display.Audio(disfile.iloc[0])
```



```
feafile = audio_df[audio_df['Label'] == 'FEA']['Filepath']
fearray = audio_df[audio_df['Label'] == 'FEA']['Arrays']
```

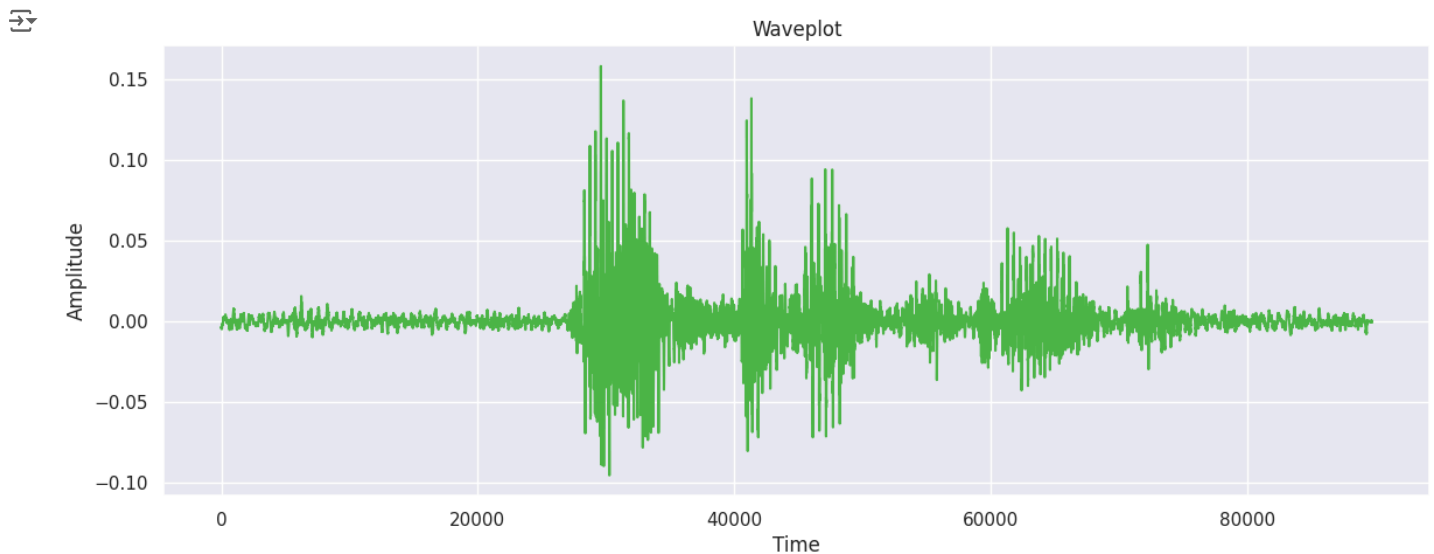
```
plt.figure(figsize=(12, 4))
plt.plot(fearray.iloc[0], color='#7D55AA')
plt.xlabel('Sample')
plt.ylabel('Amplitude')
plt.title('Waveform')
plt.show()
```

```
IPython.display.Audio(feafile.iloc[0])
```



```
neuarray = audio_df[audio_df['Label'] == 'NEU']['Arrays']
```

```
plt.figure(figsize=(14, 5))
plt.plot(neuarray.iloc[0], color='#4CB847')
plt.title('Waveplot')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```



data

```
array([ 1.03500206e-02,  1.31107224e-02,  1.21039525e-02, ...,
        -2.86973773e-05, -1.88939866e-05,  0.00000000e+00], dtype=float32)
```

#new

```
!pip install --upgrade transformers
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-packages (4.42.4)
Collecting transformers
  Downloading transformers-4.43.4-py3-none-any.whl.metadata (43 kB)
    43.7/43.7 kB 1.6 MB/s eta 0:00:00
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from transformers) (3.15.4)
Requirement already satisfied: huggingface-hub<1.0,>=0.23.2 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.23.5)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (1.26.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers) (24.1)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (6.0.1)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (2024.5.15)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from transformers) (2.31.0)
Requirement already satisfied: safetensors>=0.4.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.4.3)
Requirement already satisfied: tokenizers<0.20,>=0.19 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.19.1)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transformers) (4.66.4)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.23.2->transformers) (2024.5.1)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.23.2->transformers) (4.11.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (2024.7.4)
Downloading transformers-4.43.4-py3-none-any.whl (9.4 MB)
    9.4/9.4 MB 59.3 MB/s eta 0:00:00
Installing collected packages: transformers
  Attempting uninstall: transformers
    Found existing installation: transformers 4.42.4
    Uninstalling transformers-4.42.4:
      Successfully uninstalled transformers-4.42.4
  Successfully installed transformers-4.43.4
```

```
import transformers
print(transformers.__version__)
```

```
4.43.4
```

```
import librosa
import torch
import IPython.display as display
from transformers import Wav2Vec2ForCTC, Wav2Vec2Tokenizer
import numpy as np
```

```
#load pre-trained model and tokenizer
tokenizer = Wav2Vec2Tokenizer.from_pretrained("facebook/wav2vec2-base-960h")
model = Wav2Vec2ForCTC.from_pretrained("facebook/wav2vec2-base-960h")
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:89: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
tokenizer_config.json: 100% 163/163 [00:00<00:00, 818B/s]
vocab.json: 100% 291/291 [00:00<00:00, 2.89kB/s]
special_tokens_map.json: 100% 85.0/85.0 [00:00<00:00, 623B/s]
config.json: 100% 1.60k/1.60k [00:00<00:00, 14.4kB/s]
The tokenizer class you load from this checkpoint is not the same type as the class this function is called from. It may result in unex
The tokenizer class you load from this checkpoint is 'Wav2Vec2CTCTokenizer'.
The class this function is called from is 'Wav2Vec2Tokenizer'.
/usr/local/lib/python3.10/dist-packages/transformers/models/wav2vec2/tokenization_wav2vec2.py:720: FutureWarning: The class `Wav2Vec2To
  warnings.warn(
model.safetensors: 100% 378M/378M [00:09<00:00, 34.7MB/s]
Some weights of the model checkpoint at facebook/wav2vec2-base-960h were not used when initializing Wav2Vec2ForCTC: ['wav2vec2.encoder.
- This IS expected if you are initializing Wav2Vec2ForCTC from the checkpoint of a model trained on another task or with another archit
- This IS NOT expected if you are initializing Wav2Vec2ForCTC from the checkpoint of a model that you expect to be exactly identical (i
Some weights of Wav2Vec2ForCTC were not initialized from the model checkpoint at facebook/wav2vec2-base-960h and are newly initialized:
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```

```
#load audio file
#audio, sampling_rate = librosa.load("../input/audio-dataset/Voice 002.m4a",sr=16000)
audio, sampling_rate = librosa.load("/content/Ashwin sir.wav",sr=16000)
#audio1, sampling_rate = librosa.load("/content/Crema/1003_TSI_FEA_XX.wav",sr=16000)
```

```
audio,sampling_rate
#audio1,sampling_rate
```

```
(array([ 2.4446618e-09, -2.7890532e-09,  3.4814298e-09, ...,
         2.3986280e-03,  2.3839644e-03,  1.7409117e-03], dtype=float32),
 16000)
```

```
display.Audio("/content/Ashwin sir.wav", autoplay=True)
#display.Audio1("/content/Crema/1012_IE0_FEA_HI.wav", autoplay=True)
```

```
0:00 / 0:02
```

```
input_values = tokenizer(audio, return_tensors = 'pt').input_values
input_values
```

```
tensor([[-0.0001, -0.0001, -0.0001, ...,  0.0227,  0.0226,  0.0165]])
```

```
logits = model(input_values).logits
logits
```

```
tensor([[[ 14.4578, -29.6694, -29.3485, ..., -8.0865, -7.4964, -8.2638],
          [ 14.8893, -29.6845, -29.3455, ..., -8.0365, -7.4710, -8.2575],
          [ 15.1475, -29.8394, -29.4861, ..., -8.0169, -7.7054, -8.2599],
          ...,
          [ 14.1810, -29.8533, -29.5769, ..., -8.5200, -8.5832, -7.9956],
          [ 14.3211, -30.1341, -29.8398, ..., -8.5849, -8.5931, -8.1111],
          [ 14.2160, -29.9607, -29.6680, ..., -8.5319, -8.6026, -8.1177]]],
      grad_fn=<ViewBackward0>)
```

```
# store predicted id's
# pass the logit values to softmax to get the predicted values
predicted_ids = torch.argmax(logits, dim=-1)
```

```
# pass the prediction to the tokenizer decode to get the transcription
transcriptions = tokenizer.decode(predicted_ids[0])
```

```
transcriptions
```

```
'GOOD MORNING'
```

```
# Save the object to a pickle file
with open('df.pkl', 'wb') as file:
    pickle.dump(data, file)
```

Start coding or [generate](#) with AI.