Module 1: Summative Assesment

Emotion Recognition thru voice.

We will be using the package Librosa for the audio analysis

Importing packages

```
# importing packages
import librosa
import IPython.display as ipd
import csv
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
import os
#pip install --ignore-installed --upgrade
https://storage.googleapis.com/tensorflow/mac/cpu/tensorflow-1.6.0-
py3-none-any.whl
```

Preparing the Dataset

```
# Headers for the dataset

header = 'filename pitch magnitude harmony Perceptrual chroma_stft
spectral_centroid spectral_bandwidth rolloff zero_crossing_rate LPC'
for i in range(1, 21):
    header += f' mfcc{i}'
header += ' Emotion '
header = header.split()
```

Feature Extraction

In this part audio files of different from different folders will be passed into the loop. In the loop it will perform the feature extraction. After getting the values it will be stored in the dataset 'dataset_combined'.

The features to be extracted are the following:

1. Pitch

   - position of a single sound in the complete range of sound. Sounds are higher or lower in pitch according to the frequency of vibration of the sound waves producing them

2. Harmonics and Perceptrual

   - Harmonics are characteristichs that represent the sound color. Perceptrual shock wave represents the sound rhythm and emotion

3. Chroma Frequencies

- tells us the intensity of each of the 12 notes at a specific point in time. for music audio in which the entire spectrum is projected onto 12 bins representing the 12 distinct semitones (or chromas) of the musical octave.

4. Spectral Centroid

    - It indicates where the ”centre of mass” for a sound is located and is calculated as the weighted mean of the frequencies present in the sound.

5. Spectral Bandwidth

6. Spectral Rolloff

    - Spectral rolloff is the frequency below which a specified percentage of the total spectral energy, e.g. 85%, lies.

7. Zero crossing rate

    - the rate of sign-changes along a signal, i.e., the rate at which the signal changes from positive to negative or back.

8. Mel-Frequency Cepstral Coefficients

    - The mel frequency cepstral coefficients (MFCCs) of a signal are a small set of features (usually about 10–20) which concisely describe the overall shape of a spectral envelope.

9. Linear Prediction Coefficients (LPC)

    - determines the coefficients of a forward linear predictor by minimizing the prediction error in the least squares sense

```
  # adding the header to the dataset
  file = open('dataset_combined.csv', 'w', newline='')
  with file:
      writer = csv.writer(file)
      writer.writerow(header)

  # names for the categories of emotions used. Also the names of
each folders
  genres = 'Angry Disgusted Fearful Happy Neutral Sad
Surprised'.split()

  # loop for getting .wav files from different folders
  for g in genres:
      for filename in os.listdir(f'./Emotions/{g}'):

          # getting name for audio categories. Ex Angry
          emotion = f'./Emotions/{g}'

          # getting each .wav file
          audio_path = f'./Emotions/{g}/{filename}'
          y , sr = librosa.load(audio_path, mono=True, duration=3)

          ## Feature Extraction
```

```
            #pitch
            pitches, magnitudes = librosa.piptrack(y=y, sr=sr)
            #Harmonics and Perceptrual
            y_harm, y_perc = librosa.effects.hpss(y)
            #Chroma Frequencies
            chroma_stft = librosa.feature.chroma_stft(y=y, sr=sr)
            # Spectral Centroid
            spec_cent = librosa.feature.spectral_centroid(y=y, sr=sr)
            #Spectral Bandwidth
            spec_bw = librosa.feature.spectral_bandwidth(y=y, sr=sr)
            #Spectral Rolloff
            rolloff = librosa.feature.spectral_rolloff(y=y, sr=sr)
            #Zero crossing rate
            zcr = librosa.feature.zero_crossing_rate(y)
            #Mel-Frequency Cepstral Coefficients
            mfcc = librosa.feature.mfcc(y=y, sr=sr)
            #LPC
            lpc = librosa.lpc(y, order=16)

            # appending values per row in the dataset
            to_append = f'{filename}  {np.mean(pitches)}
{np.mean(magnitudes)} {np.mean(y_harm)} {np.mean(y_perc)}
{np.mean(chroma_stft)} {np.mean(spec_cent)} {np.mean(spec_bw)}
{np.mean(rolloff)} {np.mean(zcr)} {np.mean(lpc)}'
            #appending all MFCC values
            for e in mfcc:
                to_append += f' {np.mean(e)}'

            # appending audio category
            to_append += f' {g}'

            # writing to dataset
            file = open('dataset_combined.csv', 'a', newline='')
            with file:
                writer = csv.writer(file)
                writer.writerow(to_append.split())
```