[Tai Tran, 5162-4453]

CIS4930 Individual Coding Assignment

Spring 2023

1. Problem Statement

In the history of speaking, any words that have been uttered always carry around some type of cadence, rhythm, and most importantly emotion. While most of the time, a person listening to speech can usually tell what type of emotion the speaker is feeling or what tone the speech is spoken with, sometimes, there are sometimes very subtle differences that the listener might miss and might misconstrue the original meaning of the statement. My goal is to build a model that very accurately picks up on these small minute details that a human might not even be aware of and be able to accurately figure out what emotion is tagged along with speech.

2. Data Preparation

As usual, there had to a separation of training and testing data. Like the assignments before, I stuck with 70% training and 30% testing out of a total of 100 audio files. This was done for each emotion and there was 4 of them: angry, fear, happy, sad. This meant there were a total of 280 audio files for training and 120 files for testing. I had also categorized the testing and dataset with each emotion as well. For data prep, the audio was not tampered with in any way to boost clarity or such since the one that was provided was very clean and had little background noise. As for seeing the different properties of each audio file, I tried out different graphing tools for each audio file. I used librosa to load in the audio file and was able to graph it that way. There were four audio files, each of which were chosen from each emotion at random. I graphed things such the time domain, frequency domain, time-frequency variation of the audio, and visualized the frequency spectrum of a signal. This was to better help understand the audio and what sort of data was later to be extracted from each feature and be fit into various models and classifiers. For this assignment, I was planning to extract features such as loudness (rms), MFCCS (Mel-Frequency Cepstral Coefficients), Zero Crossing Rate, Chroma, and the Mel-spectrogram. I used librosa since that library was well equipped to extract these features and I was already familiar with it through graphing.

3. Model Development

- Model Training
    - **I had a function that would extract all the features listed above and was able to organize it all into a data frame. When it was inserted into the matrix, I would also scale the data, as well as fit and transform it and fit into a data frame. I did this for both the x component training and testing. One thing I want to highlight is the**

process of using dictionary comprehension to make a dictionary with keys of various formats such as MFCC, Chroma, Mel_Spectrogram. I then unpacked it, so each key-value pair is passed as a separate argument to the pandas data frame. This transforms the dictionary into separate columns in the resulting data frame. I discovered this through some stack overflow forum that I cannot find again. As for the y component, I had a rating system for each emotion (1: angry, 2: happy, 3: sad, 4: fear) and set it up for both training (280 values) and testing (120 values). As for the models and classifiers, I didn't deviate much from the other assignments and stuck with what I knew which was SVC, Gaussian NB, and a random forest classifier. I had considered using LSTM and CNN. CNN sounded interesting to me since it's commonly used for image classification but could apply to audio signal as well. Due to the time crunch, I couldn't not experiment with it.

- Model Evaluation
  - *Refer to repository for full data.*

4. Discussion

- **The model performed very well overall; it had a very high overall accuracy of 0.97 to 0.98. Across the board, the precision, recall, and F1-scores for all models and classifiers were very high, with most of the scores being above 0.95. This is indicative of the model correctly and consistently identifying each class. It also reflected well in the confusion matrix for each model and classifier. There were never more than 4 times in which it incorrectly guess an emotion. My model fixed the problem very well and is almost as accurate as a person doing the same tasks.**

- **At first glance, I did not understand all the various features of an audio file that well, much less extracting the data for it. I felt that was the biggest hurdle I had to overcome. Once I solved it, I felt like it was just like all the other assignments, just fitting and testing the models with data. I went through a lot of lecture slides to better figure and had to refer to stack overflow and study the physics of soundwaves to understand all the features. Another problem I ran into was working with the librosa library and using it to extract the features of the audio file. I did not know the syntax or how to do it at all. I had to reference Professor Ma's 'extract_audio_features' python file frequently to properly learn how to extract it. After that, a minor obstacle was to figure out how to store the vast amount of data. That's when I found an overflow article talking about the process of creating a dictionary, unpacking it, and then putting it all into the matrix. This was useful and groundbreaking for MFCC, Chroma, and Mel_Spectrogram and helped ease the overall process.**

- **I like this assignment and thought this was very useful for the final project, "Rizzerator" since our group had to planned to use audio modalities as one of the features. Once again, I continued to learn more about libraries which I was introduced to like pandas, numpy, and librosa. A lot of the learning came from librosa which I am grateful for. Compare this to the last assignment, I did much better with the data preprocessing and organizing. There wasn't an oversampling issue like the last one, and all data points were represented equally and fairly. This is why I feel like the model this time around did so well.**

5. Appendix

- *https://github.com/taiphlosion/Individual-Coding-Assignment-03-Audio-Modality-/tree/main*
- *Yingbo's extract audio file python file.*
- *Stack overflow*