Speech Understanding (CSL7770)
Mid-Sem Examination
Submission Report

Soham Niraj Deshmukh (B21EE067)

1 Question 1

In this question, the task defined is to collect the audio samples in our own voice, when speaking different types of texts at different pitch and volume. I collected 10 audio samples in my voice, trying to portray different scenarios and different emotions through modulations in the audio. The audio is collected for 10 different emotions - confident, curious, disappointed, doubtful, encouraging, frustrated, nervous, sarcasm and surprise.

1. Audio Feature Extraction:
   - Loads audio files using `librosa`.
   - Extracts the fundamental frequency (*F0*) using the `pyin` algorithm.
   - Computes the root mean square (RMS) energy for different frames.
   - Determines the zero-crossing rate (ZCR).
   - Calculates pitch statistics such as mean, median, min, and max frequencies.
2. Visualization:
   - Generates and saves three plots per audio file:
     (a) **Waveform:** Displays amplitude variations over time.
     (b) **Spectrogram:** Shows frequency distribution over time.
     (c) **Pitch Contour:** Plots fundamental frequency (*F0*) against time.
3. Data Processing and Storage:
   - Iterates over all audio files in the `data` directory.
   - Extracted features are stored in a dictionary.
   - Results are compiled into a Pandas DataFrame.
   - Saves the extracted features as an Excel file for reference.
4. Output Management:
   - Creates an output directory to store analysis results.
   - Saves plots and feature summaries in the `results` folder.
   - Prints progress updates during analysis.

Table 1: Audio Analysis Summary

File	Duration (s)	Sample Rate (Hz)	Max Amplitude	Min Amplitude	Overall RMS Energy	Peak RMS Energy	Mean ZCR	Mean Pitch (Hz)
Encouraging.wav	5.568	48000	0.8457	-0.8457	0.1212	0.3716	0.0506	131.16
Sarcasm.wav	5.312	48000	0.8344	-0.8344	0.1368	0.3639	0.0485	135.00
Doubtful.wav	5.120	48000	0.9485	-0.9485	0.1228	0.2912	0.0591	126.39
Curious.wav	6.315	48000	0.8476	-0.8476	0.1036	0.3237	0.0462	125.24
Disappointed.wav	6.080	48000	0.8986	-0.8880	0.1323	0.3465	0.0542	126.02

2 Question 2

In this question, we are given a dataset of speech recordings of about 20 personalities. Each of the speech has certain tone and frequency characteristics depending on the emotion behind the speech and the speaking style. The basic task was to find out the characteristics and features in the audio that help us distinguish between the speakers, between the emotions behind the speech, etc. These features include

- Zero Crossing Rate (ZCR)
- Short Time Energy (STE)
- 13 MFCC coefficients (means, std. dev.)

2.1 Dataset

The dataset contained 20 sound recordings from distinct people. The recordings were of varying lengths - ranging from 1.23 min to 36.06 min in duration.

2.2 Methodology

Initially, I defined some global parameters that would be useful throughout the feature analysis. They included - $FRAME\_SIZE = 1024$, $HOP\_SIZE = 512$, $N\_MFCC = 13$ (number of coefficients).

Next, I loaded the audio files (.wav format) into a vector representation using Librosa library. The sampling rate is not defined to preserve the quality of the audio samples using varying sampling rate for each audio file.

The first task is to calculate ZCR. For this, I

- padded the audio file by adding additional 0s to the audio vector to make sure that the audio is divisible into frames of size 1024.
- converted the padded vector into frames or windows of specified *FRAME_SIZE*.
- computed ZCR for a frame using formula

$$\text{ZCR} = \frac{1}{2 \times \text{Frame Length}} \sum |\text{sign}(x[n]) - \text{sign}(x[n-1])|$$

- computed mean of ZCR values of all windows and returned it as final ZCR.

The next task is to compute the Short Time Energy (STE). For this, I

- used the padded frames
- computed energy values by squaring the amplitude of each frame.
- computed mean of STE values of all windows and return it as final STE.

In the third task which is to extract the MFCC coefficients, I used the *librosa.feature.mfcc* function to extract MFCC coeffients (means and standard deviations). Overall we have 13 coefficients and corresponding means and coefficients for each of them.

2.3 Limitations

- **Noise Problems :** ZCR and STE features get confused when there's background noise in old recordings. The crackling, hissing, and room echoes in historical audio make these measurements unreliable. These noises can be mistaken for emotional signals, leading to incorrect analysis.

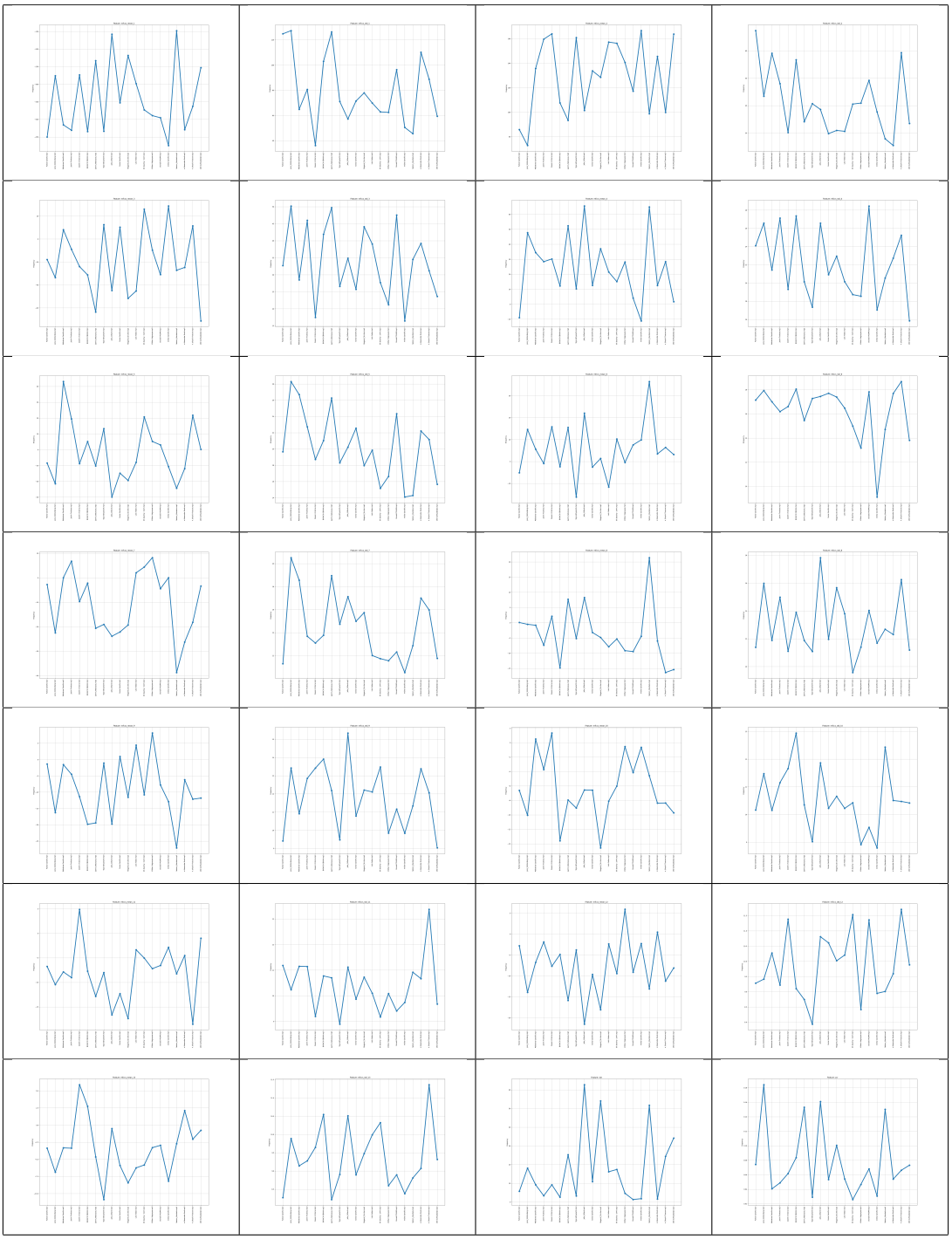

Fig. 1: (1.1-1.24) : Mean and Std. Deviation (alternate) of MFCC coefficients (1-13)
(1.25) : Short Time Energy (STE)
(1.26) : Zero Crossing Rate (ZCR)

- **Equipment Issues :** Old microphones and recording devices captured sound differently than modern equipment. They often missed certain frequencies, especially higher ones that can be important for detecting emotions. This different sound profile makes modern analysis tools less accurate when working with historical recordings.
- **No Global Knowledge :** These traditional features only analyze small chunks of speech at a time, usually 20-30 milliseconds. Emotions often develop and change over several seconds or even minutes of speech. Without looking at these longer patterns, important emotional context can be missed.
- **Different Speaking Styles :** People from different time periods had distinct ways of speaking, using different pacing and emphasis. Formal speaking styles from the past might sound emotionally intense to modern ears when they were actually neutral. These historical speaking patterns can confuse emotion detection systems trained on contemporary speech.
- **Sound Quality Problems :** Historical recordings suffer from specific issues like wow and flutter (speed variations), vinyl scratches, or tape degradation. These unique problems create sound patterns that modern analysis tools weren't designed to handle, that can interfere with emotion detection.

2.4 Improvements

- **Cleaning of noisy data :** Cleaning of sound samples and removal of background noise using audio restoration softwares, suppression of selective frequencies can help increase results. Historical recordings can be pre-processed using spectral subtraction to remove background noise. Detection and interpolation algorithms can also identify and repair these disturbances.
- **Equipment Issues :** Deep learning models like convolutional neural networks (CNNs) can learn to recognize emotional patterns despite noise, by adversarial training. Transformer-based architectures can capture long-range emotional context that traditional features miss, allowing the model to consider broader speech patterns when classifying emotions.

3 Question 3

In this question, the task is to develop a classifier to classify 5 vowel sounds (in both male and female voices), using frequency based features extracted by Linear Predictive Coding (LPC).

3.1 Dataset

The dataset contained 2 sections - male and female. In each section, we had multiple sound recordings for each of the 5 vowels.

3.2 Methodology

1. **Feature Extraction -**
   I extracted formants from the audio files (F0, F1, F2, F3). For this I first loaded each audio file into a vector representation using librosa. Then, I computed the fundamental frequency (f0) of the signal and pre-emphasized the signal to remove any low frequency noise. I then computed the autocorrelation coefficients using the Levinson-Durbin algorithm. We get the coeffiecients for which we can find out the complex roots, find the angle for each of these roots, and finally convert the angle into frequency to get our 3 formant frequencies (f1, f2, f3).
   This entire procedure is also called the Linear Predictive Coding (LRC).

2. **Feature Analysis -**
   To perform the feature analysis and find trends in the frequency data, I plotted the following graphs -
   a) F1 v/s F2 plot,
   b) F0 v/s Vowels,
   c) F1 v/s Vowels,
   d) F2 v/s Vowels,
   e) F3 v/s Vowels.

3. **Classification -**
   In order to classify the audio signals into classes, I decided to use 3 common classifiers - Support Vector Machines (SVMs), Decision Tree Classifier (DTC), Gaussian Mixture Models (GMM) and KNearestNeighbours (KNN).
   For preprocessing, I label encoded all the vowels into integers. I scaled all the features using Standard Scaler.
   The classification results in the form of the confusion matrices are provided below.

Table 2: Training and Testing Accuracy of Different Models

Model	**Training Accuracy (%)**	**Testing Accuracy (%)**
KNN	54.16	25
Decision Tree (DT)	100.00	83.33
Support Vector Machine (SVM)	56.25	25.00
Gaussian Mixture Model (GMM)	22.92	0.08

3.3 Analysis and Reflection

– **Potential Sources of Error or Confusion:**
Variations in speaking style and pronunciation among participants most likely caused variations in formant values. Speaker-dependent variables other than gender (such as age, accent, and vocal tract architecture) may have resulted in overlapping formant regions that caused errors by the classifier. When the order parameter is not optimal, the LPC analysis may miss or misidentify formants. Furthermore, applying fixed-order LPC to all speakers misses individual variances in vocal tract complexity.

– **Relation to Historical Speech Recognition Systems:**
This approach mirrors early speech recognition systems in the manner that it depends on identifying distinct acoustic patterns (particularly formants) for classification. Our methodology uses spectral features extracted though Linear Predictive Coding (LPC). The more modern approaches such as those involving deep learning, transformer-based approaches, have shifted towards more datapdriven, less feature-engineered approaches. Our formant based method provides interpretability making it suitable for research work.

– **Suggested Improvements**

1) Higher-Order Formants and Additional Acoustic Features:
Extract higher-order formants (F4, F5) as well as other properties such as Mel-Frequency Cepstral Coefficients (MFCCs), spectral centroid, and bandwidth. Increasing complexity of MFCC equation can help resolve complex classification between similar sounds. These features provide more spectrum information than F1-F3.

2) Pitch Normalization
Normalize the fundamental frequency (F0) across speakers to eliminate variability due to external factors. Z-score normalization and min-max scaling can help.

3) Dynamic feature extraction
Consider employing dynamic characteristics such as delta () and delta-delta (^{2}) to capture transitions across time, rather than static formants. Formants change throughout a vowel, so noting their movements can also help in classification.

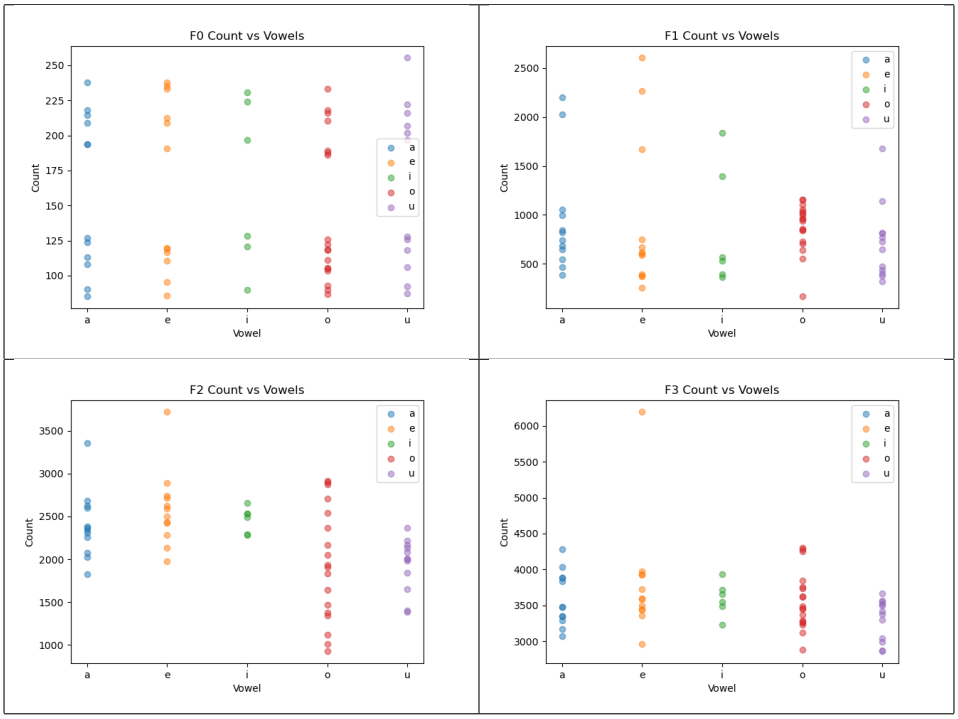

Fig. 2: Features v/s Vowels

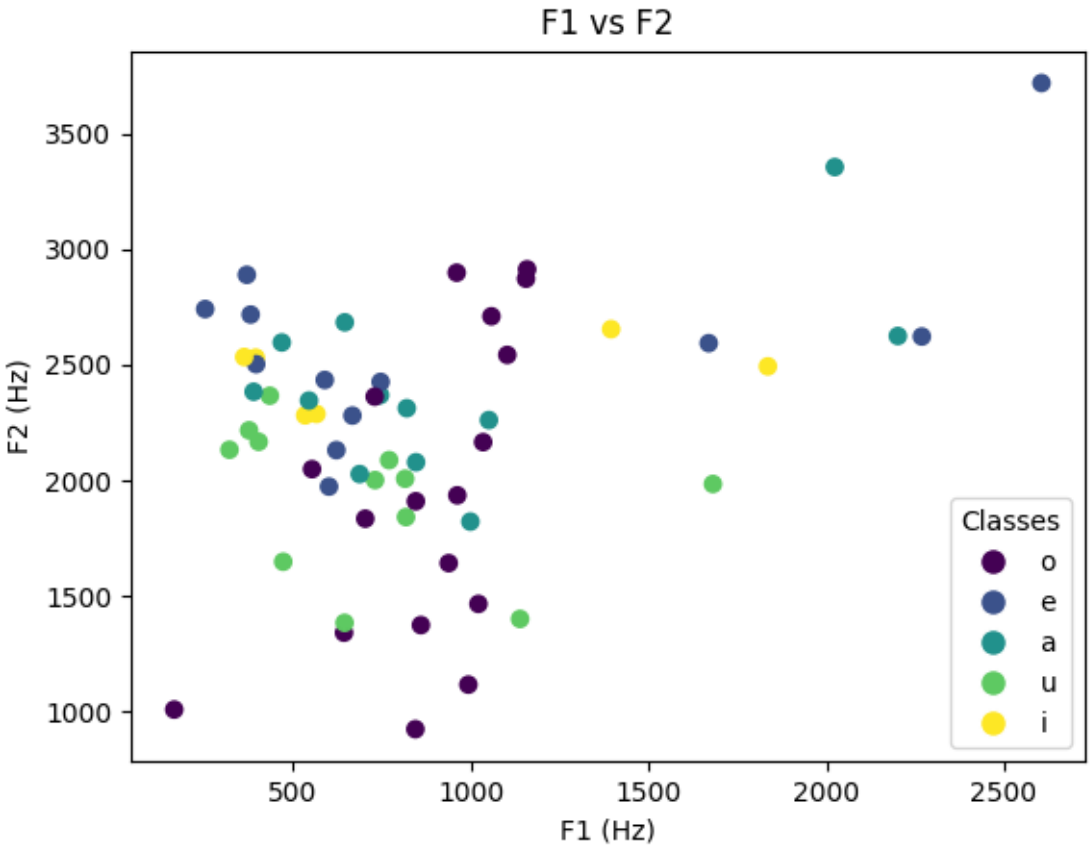

Fig. 3: Distribution of F1 (Hz) v/s F2 (Hz)

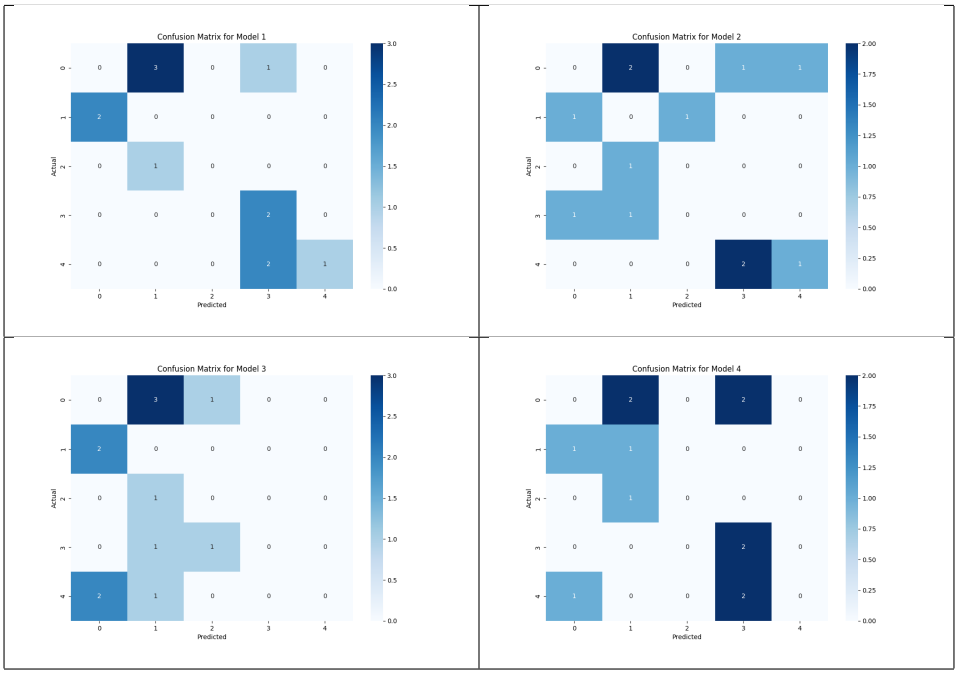

Fig. 4: Confusion Matrices - (a) SVM, (b) Decision Tree, (c) Gaussian Mixture, (d) K-Nearest Neighbors

4 References

– Librosa Official Documentation -
https://librosa.org/doc/main/_modules/librosa/core/audio.html#lpc
https://librosa.org/doc/main/util.html
https://librosa.org/doc/main/generated/librosa.feature.zero_crossing_rate.html

– Kuniga - Linear Predictive Coding (LPC) -
https://www.kuniga.me/blog/2021/05/13/lpc-in-python.html