In this code, we perform two main tasks: emotion prediction and keyword detection from audio files. Here's a step-by-step explanation of the approach:

1. **Load and Prepare the Data:**
   * We start by loading a dataset from a CSV file ('cleaned\_emotion.csv') containing audio features as columns (e.g., meanfreq, sd, median, etc.) and their corresponding emotion labels.
   * We encode the emotion labels into numerical values using the LabelEncoder from scikit-learn.
   * We split the dataset into training and testing sets to train our SVM classifier.
2. **Train an SVM Classifier:**
   * I trained a Support Vector Machine (SVM) classifier on the audio features to predict emotions. The SVM is chosen as it works well for classification tasks.
3. **Predict Emotions:**
   * I defined a function **predict\_emotion** that takes an audio file as input.
   * Inside this function, we can load the audio file, extract relevant audio features (e.g., MFCCs), and use the trained SVM model to predict the emotion.
4. **Keyword Detection:**
   * I define another function **detect\_keyword** for keyword detection using Speech Recognition Library.
   * This function loads an audio file, performs speech recognition to convert audio to text, and checks if a specified keyword is present in the recognized text.
5. **User Interface:**
   * I create tables and line charts to display the predicted emotions for a list of input audio files, however I didn’t created line charts for keyboard detection since keyword detection results are typically binary (either detected or not detected), creating a smooth line chart is not appropriate for this type of data.
6. **User Manual:**
   * I provided a user manual to guide users on how to input their audio files and interpret the results.

**Features and Preprocessing:**

* The features used for emotion prediction are not directly related to the provided features like meanfreq, sd, etc. Instead, we use Mel-frequency cepstral coefficients (MFCCs), which are commonly used for audio-based emotion recognition. We compute the mean of MFCCs as features.

**Libraries and Tools:**

* Libraries used include pandas for data handling, scikit-learn for SVM classification, librosa for audio feature extraction, matplotlib for data visualization, and speech\_recognition for keyword detection.

**Challenges:**

* Finding an appropriate dataset for this task was the biggest challenge I faced.
* Keyword detection accuracy can be affected by the quality of audio recordings and background noise.

**User Manual:**

1. **Input Your Audio Files:**
   * Place your audio files in the same directory as the script or provide the correct file paths.
   * Ensure that your audio files are in WAV format.
2. **Run the Script:**
   * Run the script in your Python environment.
   * The code will predict emotions and perform keyword detection on the provided audio files.
3. **View Results:**
   * The results will be displayed in tables and line charts in the console.
   * The tables show predicted emotions for each input audio file.
   * The line chart visualizes the predicted emotions over time.
4. **Test Your Own Voice:**
   * If you want to test your own voice, replace **'000(1).wav'** with the path to your audio file in the code.
   * Run the script again to predict the emotion for your audio.
5. **Keyboard Detection:**

* The script also contains functions for keyboard detection, you can use this by fully executing the code till the end, and you will get the results.