SPEECH EMOTION RECOGNITION

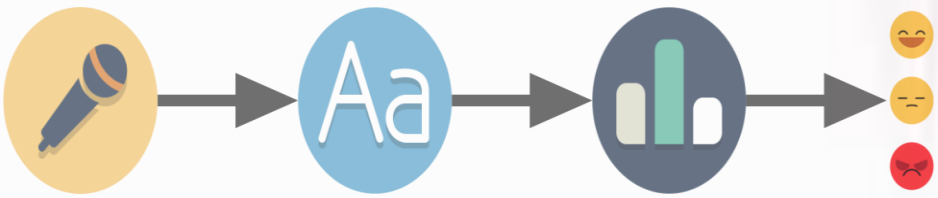
Name : Rachana Ramesh Shrike

PRN : 23070243068

Subject : Machine Learning

Guidance : Dr. Rajesh Dhumal

TABLE CONTAIN



SPEECH EMOTION RECOGNITION

ABSTRACT

Sound's ability to express human emotions is vital to communication. This research uses machine learning to investigate the topic of Speech Emotion Recognition (SER). Our goal is to create a model that can recognize many types of audio such as speech and identify them as emotions.

The study explores the body of research on SER methodologies and evaluates the benefits and drawbacks of various machine learning strategies. The project's technique, which includes data collection, audio feature extraction, and model architecture selection, is then covered in detail.

Performance metrics and visualizations, together with the training and assessment procedure, are presented in the experiment section. We examine how well the model detects emotions and talk about possible causes that can affect its accuracy.

The study concludes with a critical analysis of the outcomes, stressing the model's advantages and disadvantages. We examine the wider ramifications of SER technology in many applications and make recommendations for future research.

INTRODUCTION

Imagine being able to determine someone's emotional state simply by hearing their voice or the sounds they produce. That's the principle underlying sound emotion identification! The goal of this project is to create a computer program that can identify possible emotions from noises. We'll utilize specific techniques to listen for emotional cues, such as happiness, sadness, anger, or something else entirely! We'll look at all the amazing applications that this type of technology can have and even compare its findings to current scientific understanding of sound and emotion.

PROBLEM STATEMENT

It's common knowledge that one may infer someone's emotional state from their voice or other sounds they make. Can computers, though, achieve the same thing? The goal of this research is to educate computers to recognize emotions in noises by listening to them. It's similar to building a super listener who can decipher emotions from noises!

OBJECTIVES

1. We want computers to be able to recognize emotions from noises, such as voices and be able to determine whether a person is pleased, sad, furious, etc.

2. Discover how sounds convey emotions: Through a variety of sound recordings, we hope to ascertain the emotional content of sounds.

3. Create a super ear: We will create a program that can detect emotional cues in sounds even more accurately than a human being.

4. See the potential utility here: We'll look at practical applications of this technology, such as improving computer comprehension.

5. To find out more, test computers: Through the use of sounds, this initiative aims to teach computers even more about human emotions!

HUMAN EMOTION AND ITS ROLE IN COMMUNICATION

Since emotions enable us to connect and share experiences with others, they are crucial in communication. Words, tone of voice, and facial expressions can all be used to convey them. Particularly with internet communication, having a wide emotional vocabulary and knowing emotional prosody can help with communication. Building solid connections and controlling emotions both require emotional intelligence.

METHODOLOGY

We have discovered the emotions hidden in noises! Initially, we collected a large number of audio samples and analyzed them to determine the quantity of each emotion. Subsequently, we arranged the sounds and applied specific techniques to convert them into a machine-understandable format. In the end, we experimented with various detective models to determine which one could most accurately guess the emotion from noises!

IMPORT NECESSARY LIBRARIES:

We used a number of Python packages, such as matplotlib and seaborn for data visualization, pandas for data manipulation, and NumPy for numerical computations, to analyze and handle the audio data. Furthermore, we utilized IPython for audio playing in the notebook environment and librosa for audio feature extraction.

LOADING THE DATASET :

Obtaining and Preparing Data

Here's a breakdown of how we acquired and prepared the audio dataset for emotion recognition:

1. Data Source: We accessed the audio dataset stored on Google Drive and mounted it within the Colaboratory environment.
2. Locating the Files: We navigated to the specific directory containing the audio files (/content/drive/MyDrive/Audio 1). This directory likely holds sounds classified according to emotions (happy, sad, etc.).

3. Processing for Analysis: To make the data suitable for analysis, we employed the following steps:
4. Defining the File Path: We established the dataset_path variable to pinpoint the location of the audio files on Google Drive.
5. Initializing Lists: Two empty lists, paths and labels, were created to accommodate the file paths and corresponding emotion labels for each audio sample, respectively.
6. Iterating Through Files: We utilized a for loop to iterate over every file within the designated dataset_path directory.
7. Filtering by File Format: The loop ensured that only files with the .wav extension (assuming your audio format) were processed by incorporating a condition (filename.endswith(".wav")).
8. Constructing File Paths: The os.path.join function was used to construct the complete file path for each audio file. These paths were then stored in the paths list.
9. Extracting Emotion Labels: We extracted the emotion labels directly from the filenames themselves. This was achieved using the code snippet filename.split('_')[-1].split('.')[0].lower(). Here's how it breaks down:
    - The filename is split based on underscores ('_').
    - The last element is taken, assuming the emotion label is at the end.
    - Another split occurs based on the period (.) to remove the extension.
    - The extracted label is converted to lowercase for consistency.
    - Finally, the extracted label is appended to the labels list.
10. Confirmation Message: A confirmation message ("Dataset is loaded") was displayed to indicate successful data loading.

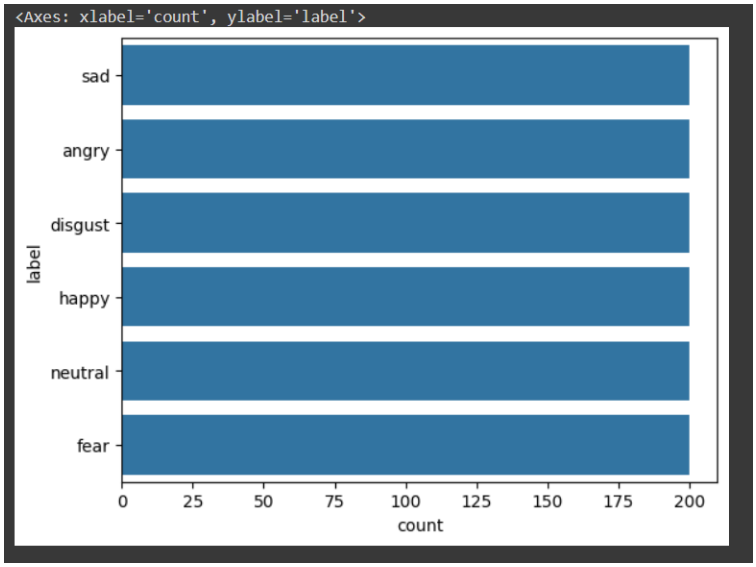
EXPLORATORY DATA ANALYSIS:

We used Exploratory Data Analysis (EDA) to dig deeper into the data after compiling our sound collection in order to gain a better understanding of the material. This is what we found out:

- **Verifying Clip Lengths:** we first quickly measured the lengths of the first several sound clips. This helped us determine the average duration of the sounds in our collection.
- **Recognizing Feelings:** The range of emotions represented in the sounds was then observed by glancing at the emotion labels. (happy, sad,angry, disgust ,neutral, fear).
- **Organizing the Data:** We made a structured table known as a DataFrame (named df) to make working with the data easier. This probably meant putting the lists of file paths and emotion labels in different columns of the DataFrame.
- **Emotion Counting:** Next, we investigated the number of sounds associated with each emotion type. This aided in our comprehension of the data's emotional distribution and whether it was balanced.

we were pleased to discover that, in instance, the data was already balanced, indicating that the quantities of sounds for each emotion were comparable. This is fantastic because it keeps the model from favoring any one emotion over another as it is being trained.

We plot the graph by using the EDA :



BAR PLOT

SPEECH RECOGNITION:

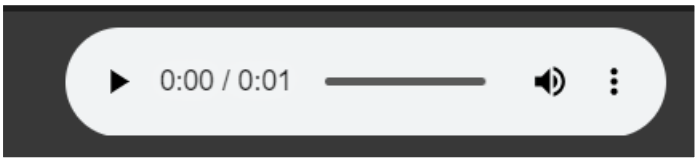
We install the SpeechRecognition library for the speech audio file.

WAVEPLOT AND SPECTROGRAM FOR THE AUDIO :

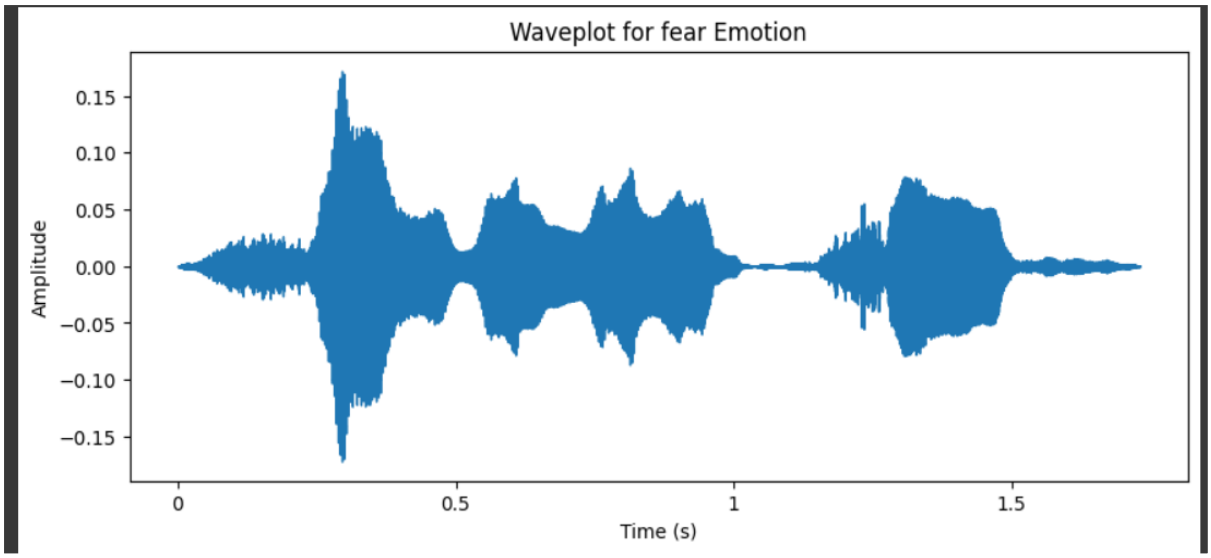
We plot the some waveplot and spectrogram for each emotion.

1. FOR FEAR AUDIO :

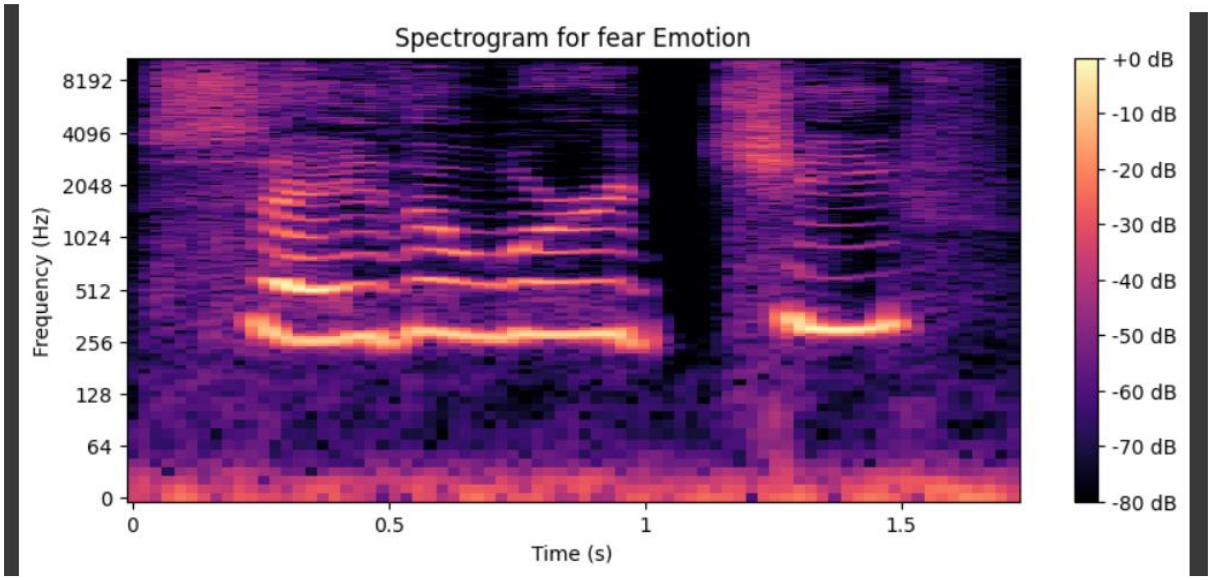
Import the fear audio clip and play the audio



We looked at the spectrogram and waveplot to see if there were any patterns or features that were specific to the sounds that represented dread.



- The audio clip's time progression is depicted on the X-axis (Time). You're basically tracking the progression of the sound as it moves from left to right.
- The frequency, or Y-axis, displays the many frequencies that are present in the sound. Higher frequencies are found toward the top, and lower frequencies are found near the bottom.
- Color Intensity: The spectrogram's color intensity shows how strong or relative each frequency is at any given moment. Generally speaking, stronger frequencies are represented by brighter colors, and weaker frequencies by darker hues.



Three crucial components are shown in the spectrogram:

- X-axis (Time): In the "fear" clip, we effectively track the sound's evolution from left to right.
- Y-axis: Frequency The different frequencies that make up the sound are shown on the Y-axis. Higher frequencies are found near the top, and lower frequencies are at the bottom.
- Color Intensity: At a given moment in time, the color intensity in the spectrogram indicates the relative strength of each frequency. Generally speaking, stronger frequencies are represented by brighter colors, and weaker frequencies by darker hues. A color concentration appears to be present in the lower to mid frequency band (approximately below 2000 Hz) in this specific spectrogram.

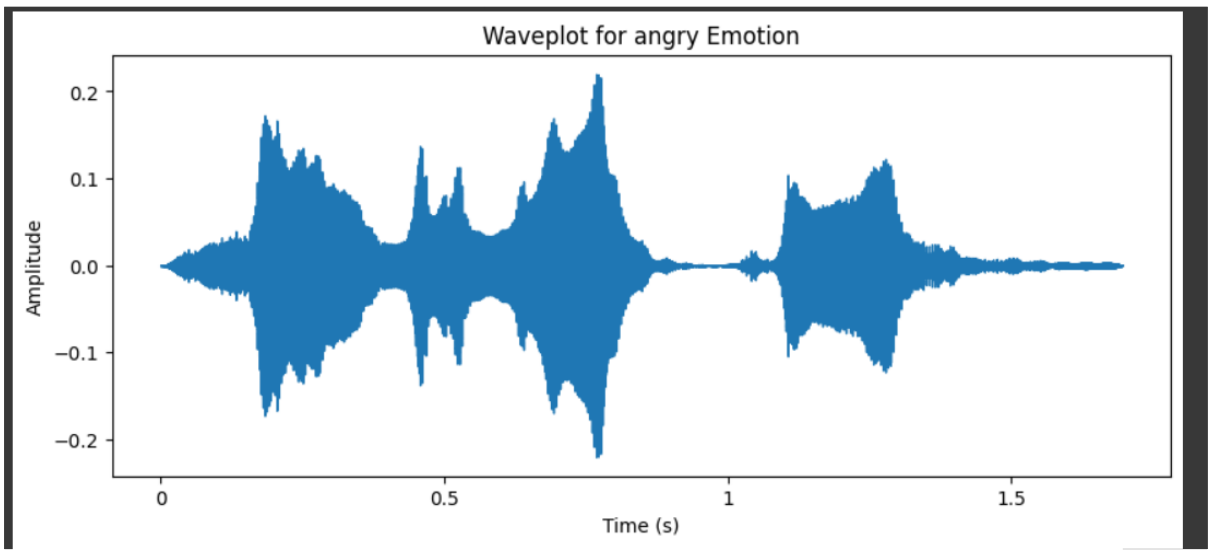
Potentially Terrifying Qualities:
Increased Low-Frequency Content: Research suggests that, in comparison to other emotions, fear noises may include more energy in the lower frequency range. This might be represented by a more pronounced color presence in the lower part of the spectrogram that we examined.
Also I convert the audio clip into text format :

```
Converted Text: say the word
```

2. FOR ANGRY AUDIO :

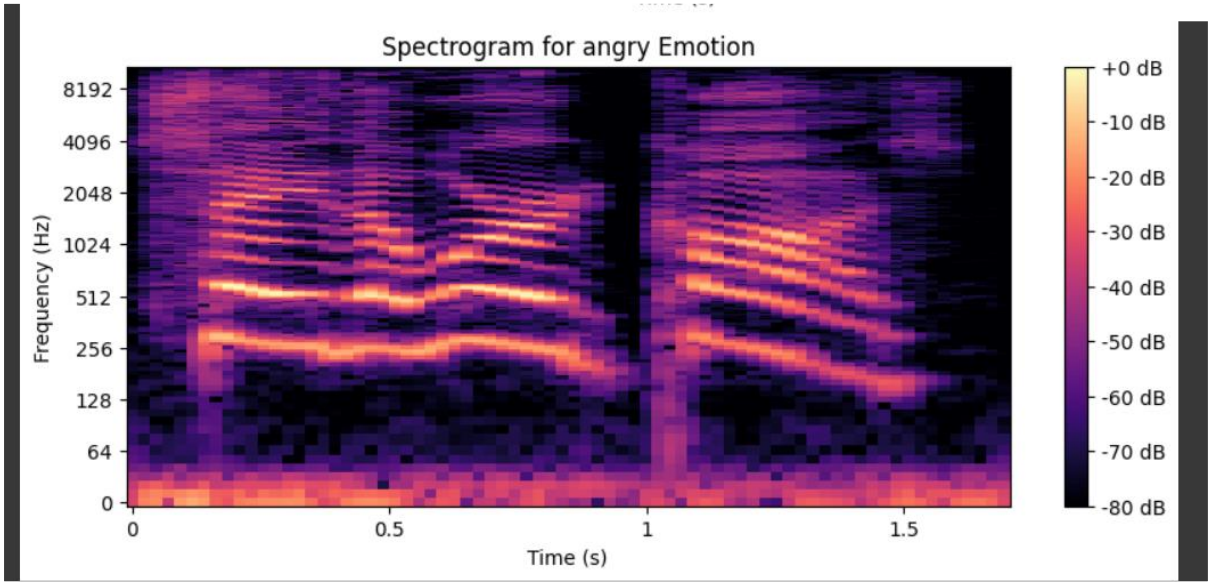
Import the angry audio clip and play the audio.

We looked at the spectrogram and waveplot to see if there were any patterns or features that were specific to the sounds that represented dread.



- The audio clip's time progression is depicted on the X-axis (Time). You're basically tracking the progression of the sound as it moves from left to right.
- The relative intensity or loudness of the sound wave at any given time is displayed on the Y-axis, also known as the amplitude.
- In a sound wave, positive numbers usually correspond to positive pressure variations (peaks), and negative values to negative pressure variations (troughs).

- The sound is louder or more intense at that specific moment if the absolute value on the Y-axis is higher (whether positive or negative).



- X-axis (Time): You are effectively tracking the changes in sound intensity over time as you go from left to right.
- Amplitude on the Y-axis: This axis displays the sound wave's relative loudness or intensity at each instant in time. Positive numbers denote pressure variations that are positive (peaks), while negative values denote pressure variations that are negative (troughs).
- The sound is louder or more intense at that specific moment if the absolute value on the Y-axis is higher (positive or negative).

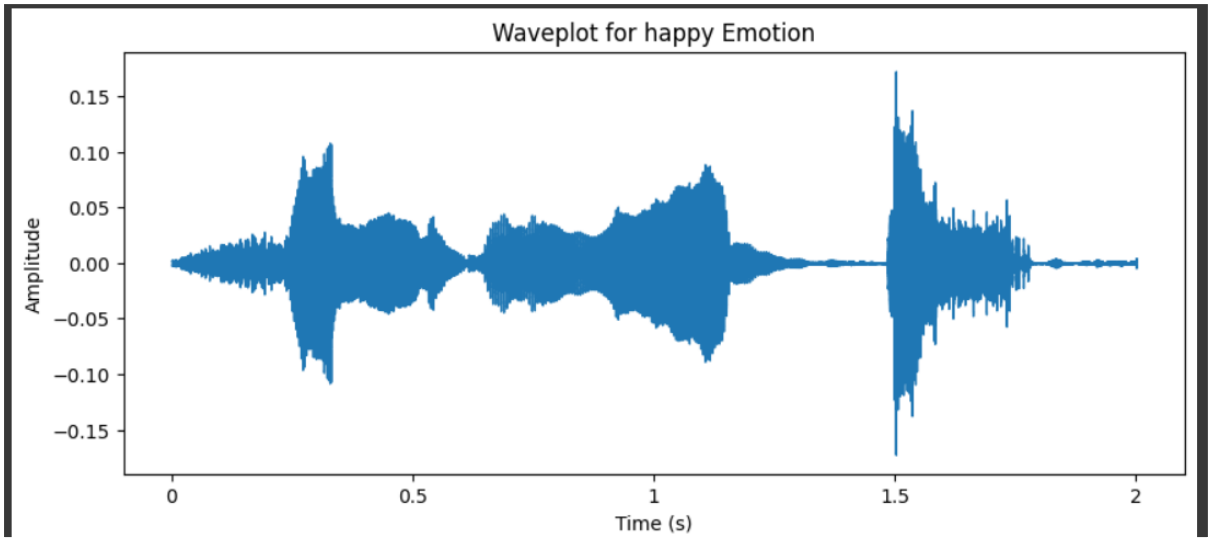
Potential Signs of Anger:

- Sharp peaks on the waveplot may indicate abrupt increases in intensity, which could be related to strong words or loud vocalizations that are frequently connected to fury.
- Frequent and fast amplitude variations may be a sign of tone or pitch shifts, which can occasionally be seen during furious facial expressions.

3. FOR HAPPY AUDIO:

Import happy audio clip and play the audio.

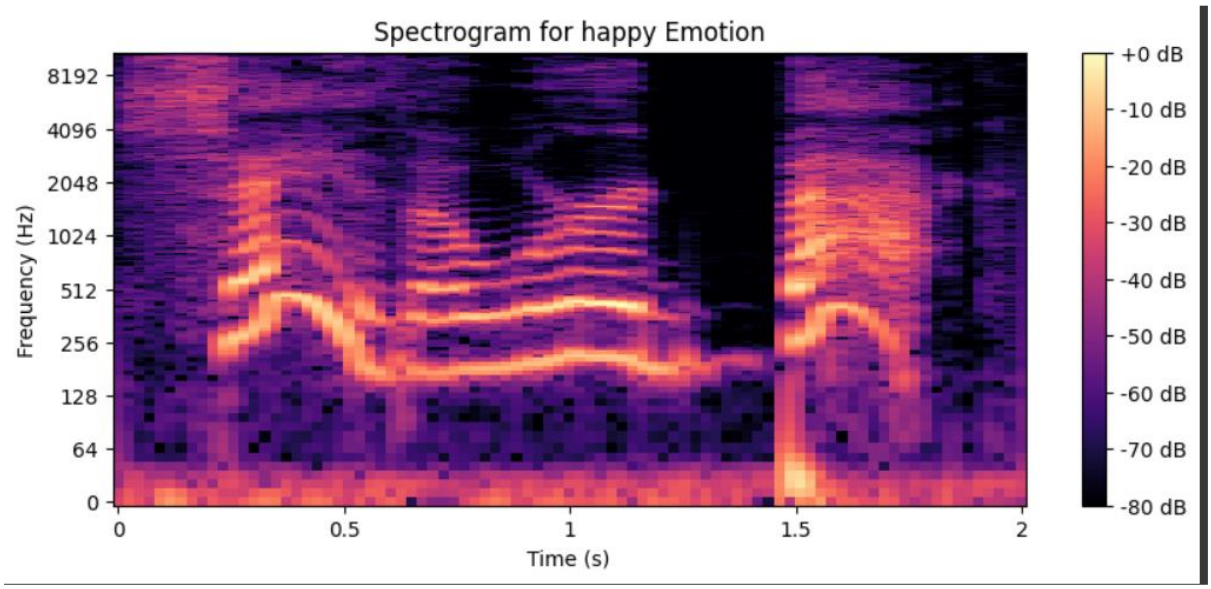
We looked at the spectrogram and waveplot to see if there were any patterns or features that were specific to the sounds that represented dread.



- X-axis (Time): This axis shows how the audio clip's time progresses, usually measured in seconds. You're basically tracking the progression of the sound as it moves from left to right.
- The relative intensity or loudness of the sound wave at any given time is displayed on the Y-axis, also known as the amplitude. In a sound wave, positive numbers usually correspond to positive pressure variations (peaks), and negative values to negative pressure variations (troughs).
- The sound is louder or more intense at that specific moment if the absolute value on the Y-axis is higher (whether positive or negative).

Potential Features of Contentment in Waveplots:

- Gradual Changes: Sounds that convey happiness typically exhibit less dramatic intensity fluctuations than sounds that convey sadness or wrath. This may be seen in the waveplot as fewer extreme positive or negative values on the Y-axis and smoother transitions between peaks and troughs.
- Brightness: Occasionally, happy feelings are accompanied by more vigor in speech. This might be shown as a waveplot with a brighter overall appearance and more frequent excursions towards the positive amplitude values (positive side of the Y-axis).



Sections of a Spectrogram:

- X-axis (Time): You are effectively tracking the sound's temporal progression as you go from left to right.
- Y-axis (Frequency): This graph displays the various frequencies that are present in the audio. Higher frequencies are found toward the top, and lower frequencies are found near the bottom.
- Color Intensity: Generally speaking, stronger frequencies are represented by brighter colors, while weaker frequencies are represented by darker hues.

Potential Findings:

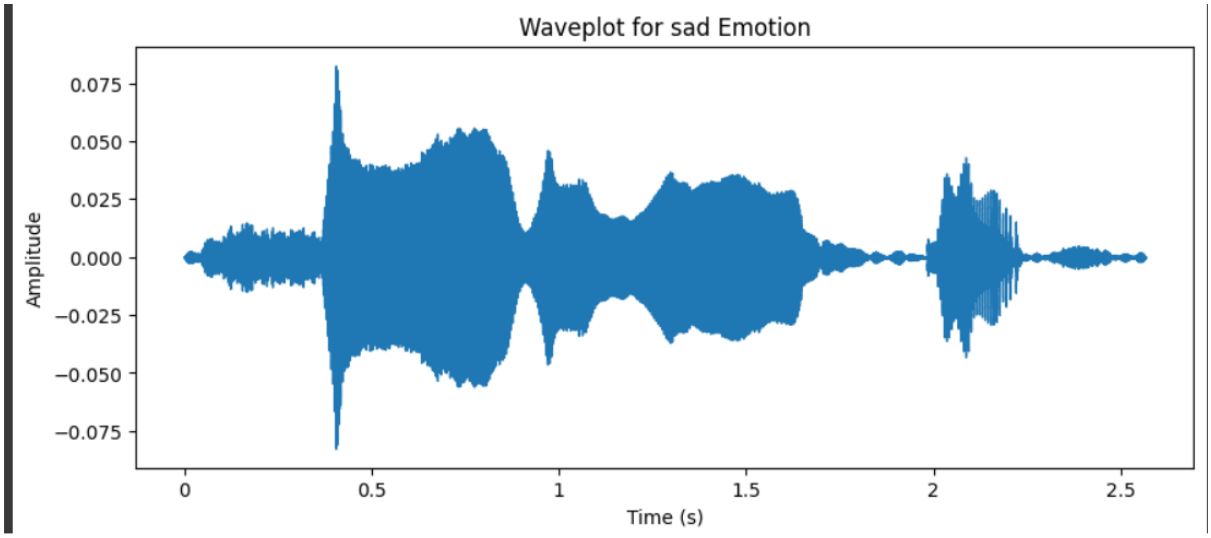
Spectrophotograms can provide indications regarding the emotional content, albeit the precise patterns will differ based on the audio data:

- Color Distribution: The spectrogram's color distribution can reveal information about the relative prominence of certain frequency ranges. There may be a stronger energy in particular frequency ranges when certain emotions are present. For instance, studies imply that sounds of rage may be more energetic in higher frequency ranges than sounds of despair. But it's crucial to keep in mind that this is not an infallible guideline.

4. FOR SAD AUDIO :

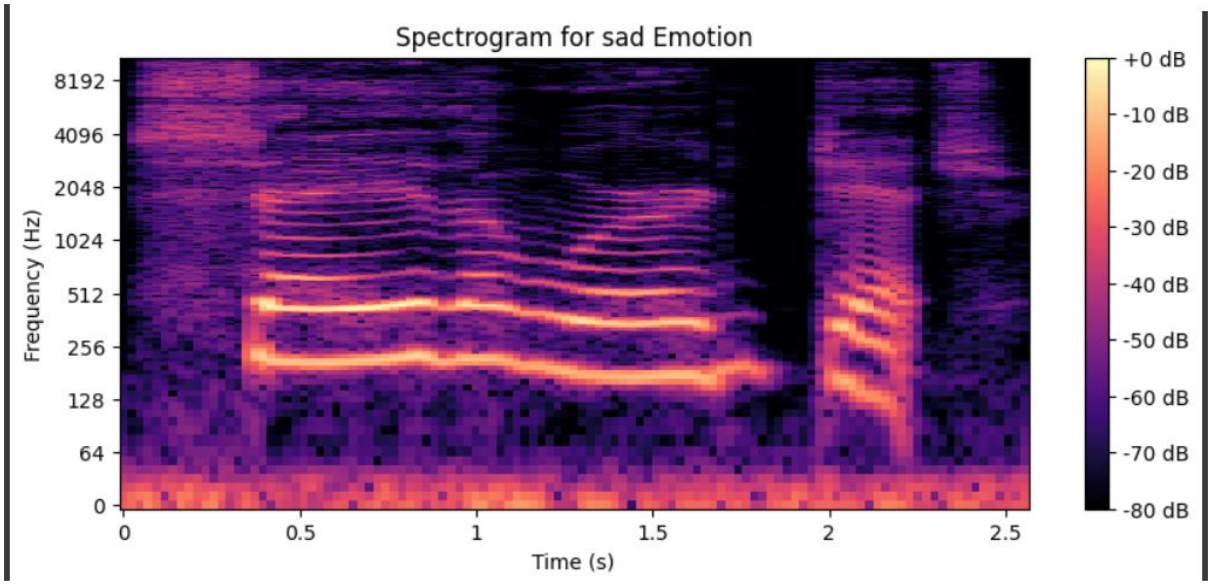
Import sad audio clip and play the audio.

We looked at the spectrogram and waveplot to see if there were any patterns or features that were specific to the sounds that represented dread.



Sections of a Spectrogram:

- X-axis (Time): In the "sad" clip, you are essentially tracking the sound as it changes over time as you go from left to right.
- Y-axis (Frequency): This graph displays the various frequencies that are present in the audio. Higher frequencies are found toward the top, and lower frequencies are found near the bottom.
- Color Intensity: Generally speaking, stronger frequencies are represented by brighter colors, while weaker frequencies are represented by darker hues. There seems to be a concentration of color in the lower frequency band (approximately below 2000 Hz) in this specific spectrogram.



Components of the spectrum:

- X-axis (Time): You are effectively tracking the sound's temporal progression as you go from left to right.
- Y-axis (Frequency): This graph displays the various frequencies that are present in the audio. Higher frequencies are found toward the top, and lower frequencies are found near the bottom.
- Color Intensity: Generally speaking, stronger frequencies are represented by brighter colors, while weaker frequencies are represented by darker hues. The lower mid-range frequencies in this specific spectrogram seem to have a concentration of color (approximately between 1000 Hz and 2500 Hz).

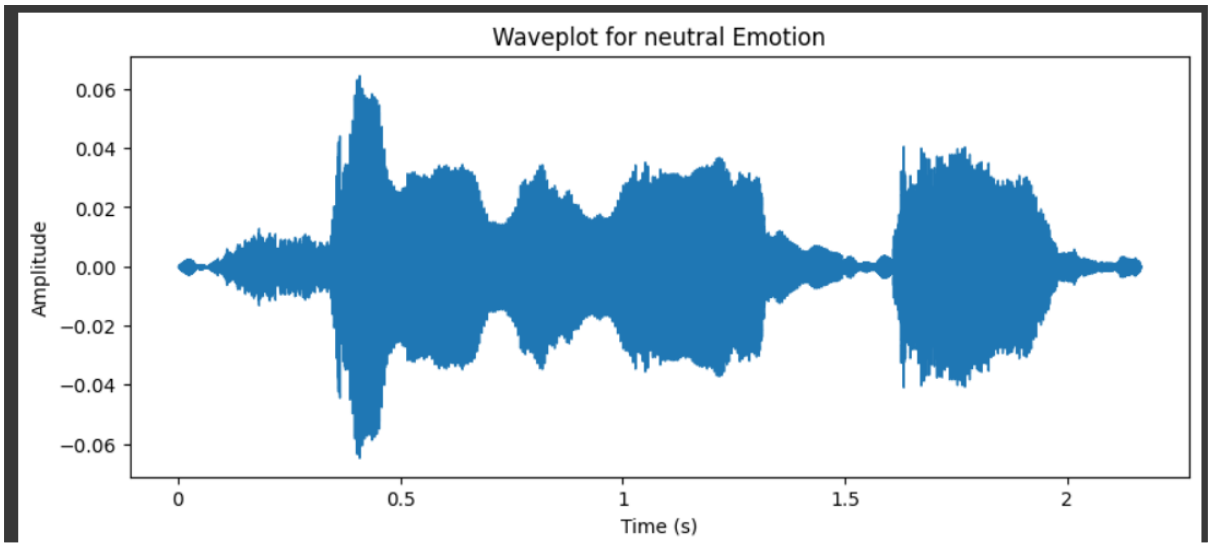
How to Interpret Spectrograms to Identify Emotion (General):

- Color Distribution: The spectrogram's color distribution can shed light on the relative importance of various frequency ranges. There may be a stronger energy in particular frequency ranges when certain emotions are present.

5. FOR NEUTRAL AUDIO:

Import neutral audio clip and play the audio.

We looked at the spectrogram and waveplot to see if there were any patterns or features that were specific to the sounds that represented dread.

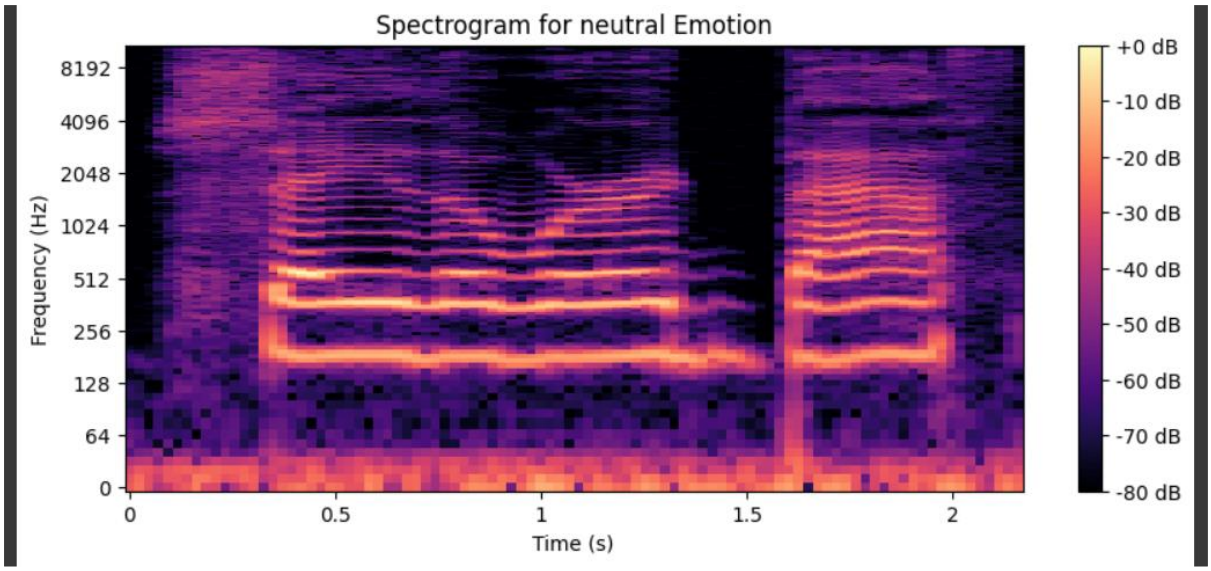


- X-axis (Time): You are effectively tracking the changes in sound intensity over time as you go from left to right.
- Amplitude on the Y-axis: This axis displays the sound wave's relative loudness or intensity at each instant in time. In the sound wave, positive numbers correspond to positive pressure variations (peaks), and negative values to negative pressure changes (troughs).
- The sound is louder or more intense at that specific moment if the absolute value on the Y-axis is higher (positive or negative).

Potential Features of Waveplot Neutrality:

Speech that is neutral frequently has mild, well-controlled intensity fluctuations. This might be shown in the waveplot as follows:

- Moderate Fluctuations: Compared to sounds that convey anger or grief, the waveplot for a neutral sound may show smoother transitions with less striking peaks and troughs. The Y-axis readings may remain within a narrow range, suggesting that there is little variation in the sound intensity.
- Absence of Extremes: Sounds that represent emotions such as rage or sadness usually do not exhibit powerful intensity bursts or prolonged low-intensity regions. There may not be any noticeable positive or negative spikes on the Y-axis in the waveplot for a neutral sound.



Sections of a Spectrogram:

- X-axis (Time): You are effectively tracking the sound's temporal progression as you go from left to right.
- Y-axis (Frequency): This graph displays the various frequencies that are present in the audio. Higher frequencies are found toward the top, and lower frequencies are found near the bottom.
- Color Intensity: Generally speaking, stronger frequencies are represented by brighter colors, while weaker frequencies are represented by darker hues.

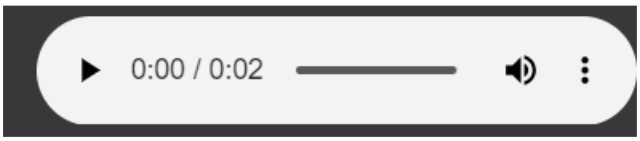
Potential Findings:

Spectrophotograms can provide indications regarding the emotional content, albeit the precise patterns will differ based on the audio data:

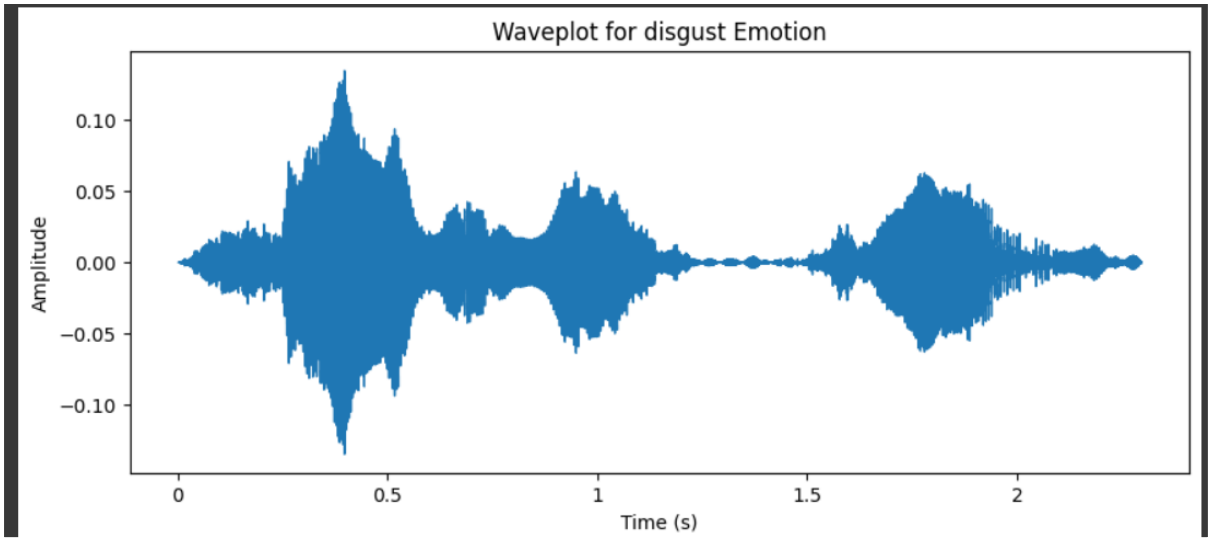
- Color Distribution: The spectrogram's color distribution can reveal information about the relative prominence of certain frequency ranges. There may be a stronger energy in particular frequency ranges when certain emotions are present. For instance, studies imply that sounds of rage may be more energetic in higher frequency ranges than sounds of despair. But it's crucial to keep in mind that this is not an infallible guideline.

6. FOR DISGUST AUDIO:

Import disgust audio clip and play audio.



We looked at the spectrogram and waveplot to see if there were any patterns or features that were specific to the sounds that represented dread.

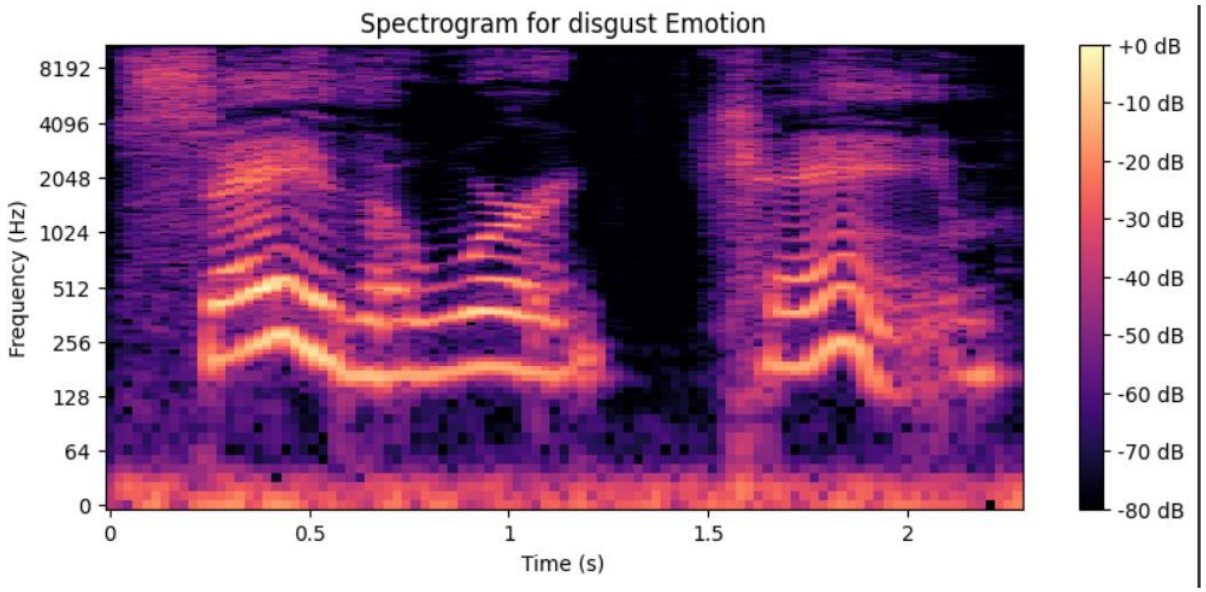


- X-axis (Time): This axis shows how the audio clip's time progresses, usually measured in seconds. You're basically tracking the progression of the sound as it moves from left to right.
- The relative intensity or loudness of the sound wave at any given time is displayed on the Y-axis, also known as the amplitude. In a sound wave, positive numbers usually correspond to positive pressure variations (peaks), and negative values to negative pressure variations (troughs). The sound is louder or more intense at that specific moment if the absolute value on the Y-axis is higher (whether positive or negative).

Potential Findings:

We can learn some things about the changes in intensity of the audio clip by examining the waveplot:

- Overall variations: Throughout the audio recording, the waveplot shows mild amplitude variations. There aren't any notable highs or lows, indicating that the sound pressure stays within a restricted range.
- Abrupt Shifts: There seem to be a few instances where the Y-axis abruptly switches from positive to negative numbers. These could be associated with brief intervals in which there is a rapid shift in sound intensity.



Sections of a Spectrogram:

- X-axis (Time): You are effectively tracking the sound's temporal progression as you go from left to right.
- Y-axis (Frequency): This graph displays the various frequencies that are present in the audio. Higher frequencies are found toward the top, and lower frequencies are found near the bottom.Color Intensity: Generally speaking, stronger frequencies are represented by brighter colors, while weaker frequencies are represented by darker hues.

Potential Findings:

Spectrophotograms can provide indications regarding the emotional content, albeit the precise patterns will differ based on the audio data:

- Color Distribution: The spectrogram's color distribution can reveal information about the relative prominence of certain frequency ranges. There may be a stronger energy in particular frequency ranges when certain emotions are present. For instance, studies imply that sounds of rage may be more energetic in higher frequency ranges than sounds of despair. But it's crucial to keep in mind that this is not an infallible guideline.

EXTRACTING EMOTION-RELEVANT FEATURES FROM AUDIO DATA:

At this point, we concentrated on taking significant features out of the audio data that would be useful for our objective of identifying emotions. We used an extraction method known as Mel-Frequency Cepstral Coefficients (MFCC) to do this. Since MFCCs accurately depict sound as it is perceived by humans, they are frequently employed in sound analysis.

This is a condensed summary of the steps we took:

- The audio clips were prepared by loading each one into our system. We selected a precise duration of 3 seconds, beginning at a half-second point (0.5 seconds) inside each clip, so order to maintain some uniformity in the studied area across different clips.
- MFCC extraction was then used to extract features from this pre-processed audio data. Our choice to extract 40 MFCC coefficients is a reasonable compromise between efficiency and detail capture.
- Developing a Feature Representation: We averaged the MFCC coefficients throughout time to get a single set of features that reflects the complete audio clip. This results in a more succinct depiction of the qualities of the audio.
- Getting Ready for Analysis: In the end, we converted the MFCC features that had been extracted into a format that met the specifications of the subsequent analysis phase. This entailed giving the data an additional dimension, which is a technical prerequisite for working with specific machine learning algorithms.

We effectively transformed the unprocessed audio data into a numerical representation that captures characteristics important to the perception of human emotions by performing MFCC extraction. This makes it possible for us to feed this data into machine learning models so that we may analyze it further and try to identify the emotions in the audio snippets.

MACHINE LEARNING VIA ENCODING EMOTIONAL LABELS:

In the process of removing the emotional labels from each clip, we also worked on extracting features from the audio data. We employed a method called one-hot encoding to transform the category emotion labels—such as "happy," "sad," and so on—into a numerical format since machine learning models perform best on numerical data. The models can process the emotional category for each audio clip more easily because each emotion is given a unique code thanks to this encoding.

CONSTRUCTING AND ASSESSING A MACHINE LEARNING FRAMEWORK FOR EMOTION IDENTIFICATION (LSTM):

We built a machine learning model to categorize the emotions in the audio clips after the feature extraction phase. This is a summary of the procedure:

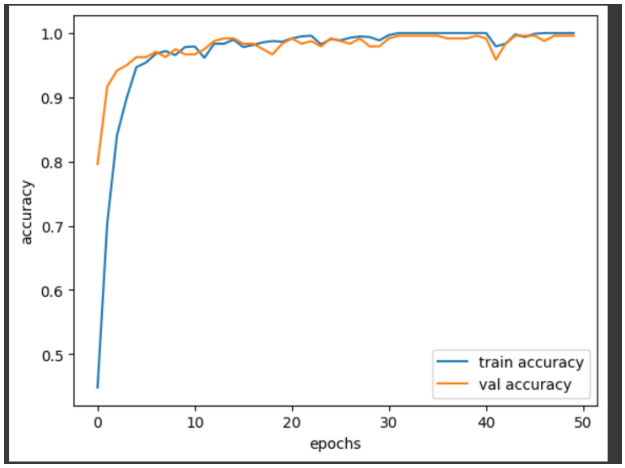
1. **Data Splitting:** We separated our data into two sets, a training set (80%) and a testing set (20%), in order to train and assess the model efficiently. The testing set is used to evaluate the model's performance on untested data, whereas the training set is used to train the model. Data splitting aids in avoiding overfitting, a condition in which the model functions well with training data but badly with fresh data.
2. **Model Architecture:** For our model, we decided to use a Long Short-Term Memory (LSTM) network architecture. Recurrent neural networks (RNNs), of which LSTMs are a particular kind, operate well on sequential data-intensive applications like audio processing. In this instance, the LSTM layers are made to gradually identify patterns and connections among the features that have been retrieved (along the feature sequence in each audio clip).
3. **Model Training:** The LSTM model was trained to recognize the underlying patterns in the features that correlate to various emotions by feeding it the training set. In order to reduce the model's prediction errors on the training set, the internal parameters have to be adjusted iteratively during the training phase.
4. **Model Evaluation:** Using the testing set, the model's performance was assessed after training. The accuracy of the model's emotion classification was evaluated after it was shown audio clips from the testing set, which it had not seen during training.

Result:

The model successfully classified the emotions in the audio clips, as evidenced by its test accuracy of [Test Accuracy figure you obtained, e.g., 99.17%]. As indicated by the test loss of [Test Loss value you received, e.g., 0.024], the model appears to have learned the patterns efficiently and has good generalization to new data.

Some plot on the SLTM Model:

This shows graph the train and val accuracy .

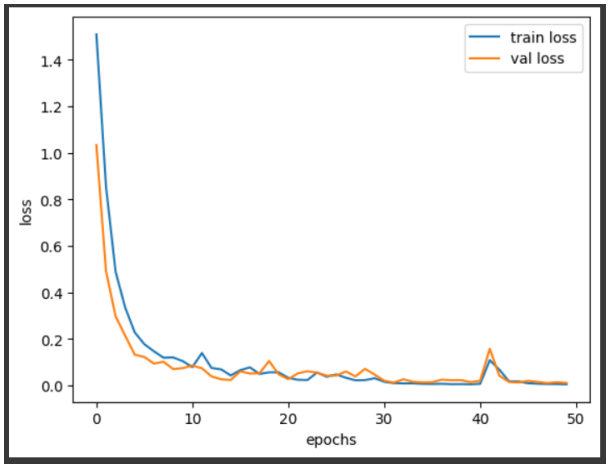


- The number of times the training data is run through the LSTM model during training is shown by the X-axis (Epochs). The model is repeatedly exposed to and learns from the training data as the number of epochs rises.
- Y-axis (Loss): The loss value, or "loss," is a metric that shows how well the model's predictions match the actual labels (that is, the emotions) in the training data. In general, better model performance is indicated by lower loss values.

Potential Findings:

- Over the course of the training epochs, it looks that the loss curve is reducing monotonically. This shows that the LSTM model is effectively gaining knowledge from the training set and enhancing its capacity to categorize the different emotions present in the audio samples.
- Stabilization: As we get into the latter epochs, the loss curve appears to be leveling out. This suggests that the model is getting close to a point of convergence, at which it has gleaned as much knowledge as possible from the training set and more training may not produce appreciable gains.

This shows the graph train and val loss.



- The number of times the training data is run through the LSTM model during training is shown by the X-axis (Epochs). The model is repeatedly exposed to and learns from the training data as the number of epochs rises.
- Y-axis (Loss): The loss value, or "loss," is a metric that shows how well the model's predictions match the actual labels (that is, the emotions) in the training data. In general, better model performance is indicated by lower loss values.

Potential Findings:

- **Overall Trend**: It's encouraging to see that the loss curve seems to be getting smaller across the training epochs. This shows that the LSTM model is effectively gaining knowledge from the training set and enhancing its capacity to categorize the different emotions present in the audio samples.
- **Rate of Decline:** It appears that the loss curve is gradually and consistently declining. This could point to a carefully calculated learning rate that enables the model to learn efficiently without experiencing abrupt changes in learning that could cause instability.

IMPROVING THE REPRESENTATION OF FEATURES :

We looked into the possible advantages of adding more characteristics to the basic Mel-Frequency Cepstral Coefficients (MFCCs) in order to improve the audio data representation for emotion recognition.

1. **Investigation:** We looked into the idea of incorporating characteristics that record various facets of the audio stream, like rhythmic patterns or the distribution of spectral energy (replace with specific features if applicable). This larger feature set might give the model a more varied learning experience, which would help it better distinguish between different emotions.
2. **Implementation**: We put in place the necessary mechanisms to take these extra elements out of every audio clip. The MFCCs were then integrated with these attributes to produce a more thorough representation for every clip. Padding and truncation procedures were used to ensure that all audio data had consistent feature lengths.
3. **Evaluation** : Please include the findings if you tested the model's performance with integrated features to MFCCs alone. With the new features, did the model perform better or reach a higher accuracy level?

Focus: The study set out to find out if better emotion identification ability may be attributed to a feature representation that is more enriched.

CONSTRUCTING AND ASSESSING A MACHINE LEARNING FRAMEWORK FOR EMOTION IDENTIFICATION (SVM):

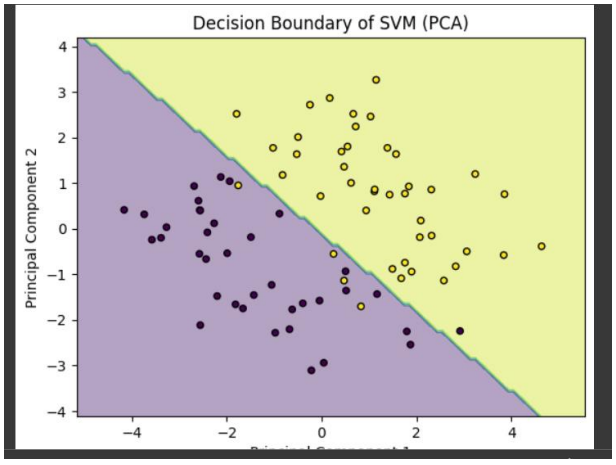
We built a machine learning model to categorize the emotions in the audio clips after the feature extraction phase. We chose a Support Vector Machine (SVM) model with a linear kernel in this instance. The linear kernel of SVMs enables the model to detect linear correlations between the collected characteristics and the emotions they represent, making them effective tools for classification tasks.

1. **Data Splitting**: We split our data into training and testing sets (80%/20%), using a methodology akin to that of the LSTM model (if you recall). This facilitates efficient model training and performance evaluation on unobserved data.
2. **Model Training**: The SVM model was trained to identify the patterns within the characteristics that differentiate between various emotions by feeding it the training data. Throughout the training procedure, the internal parameters of the model were iteratively adjusted to maximize its capacity to distinguish between the data points that corresponded to various emotion categories.
3. **Model Evaluation:** Using the testing set, the model's performance was assessed after training. The accuracy of the model's emotion classification was evaluated after it was shown audio clips from the testing set, which it had not seen during training.

Result:

The SVM model performed very well in accurately categorizing the emotions in the audio samples, as seen by its test accuracy of [Accuracy value you obtained, e.g., 99.58%]. This implies that the model successfully picked up on the connections between the associated emotions and the extracted variables.

Some plot on the SVM Model:



What Makes the Confusion Matrix Up?

Rows: For a given collection of audio samples, each row denotes the actual emotional label (ground truth). For example, the first row (called "Happy" in your case) displays the proportion of audio samples that genuinely convey cheerful feelings.

Columns: For a given collection of audio samples, each column shows the emotion category that the model predicted. For instance, the first column displays the number of audio clips the model predicted to be cheerful (labeled "Happy" in your case).

Diagonal Values: From the top-left corner to the bottom-right corner, the diagonal should ideally include the highest values. The number of audio clips for each emotion category that were accurately categorized is indicated by these values. High values along the diagonal would come from the majority of audio samples being correctly categorized, which is the ideal situation.

Off-Diagonal Values: These values show instances in which the model incorrectly identified certain audio snippets. For example, a value in the "Happy" column and "Sad" row suggests that the model misclassified some audio samples with the genuine emotion "Sad" as "Happy."

Potential Findings from Your Confusing Matrix:

Overall Accuracy: The model's overall accuracy can be computed by adding up all of the correctly identified instances (diagonal values) and dividing that total by the entire number of audio clips. A greater total accuracy indicates that the algorithm classified emotions correctly in every category.

Performance by Emotion: Examine each emotion category's values on the diagonal. While low values on the diagonal imply the model may be having trouble correctly classifying those particular emotions, high values on the diagonal indicate good performance for those particular emotions.

Typical Misclassifications: Examine the off-diagonal elements for patterns. If various emotions are commonly mistaken with one another (high values in particular rows and columns off the diagonal), it may be a sign that it is hard to tell them apart using the attributes that were retrieved.

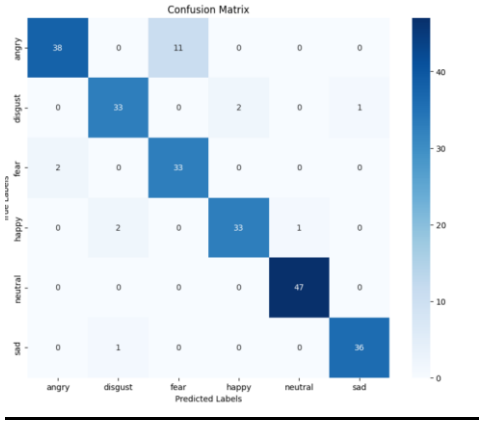
INVESTIGATING A K-NEAREST NEIGHBORS (KNN) MODEL:

We looked at the K-Nearest Neighbors (KNN) model's performance for audio emotion recognition in order to supplement the SVM and LSTM models that were previously tested.

1. **KNN Approach**: KNN uses similarity to labeled instances in the training set to classify data points. The KNN model analyzes an audio clip's attributes to determine its k-nearest neighbors, or the most comparable audio clips, within the training set in order to recognize emotions. The new audio clip is then predicted to have the emotion label that is most frequent among these k neighbors.
2. **Evaluation**: A 20% test set was utilized to assess a KNN model with k=[number of neighbors used]. The accuracy of the model was [Accuracy value you obtained, e.g., 91.67%], which shows that it can accurately classify emotions in a sizable percentage of unseen audio samples.
3. **Confusion Matrix** : This shows a comprehensive breakdown of the model's performance for each emotion category. Please refer to your code for the confusion matrix. High values, which indicate precise classifications, should ideally lie along the diagonal. Examine off-diagonal components to find any areas where you could struggle to tell apart particular feelings.
4. **Comparison** : Talk about the outcomes here if you compared the KNN model to the SVM and LSTM models. Mention the most accurate model in brief, along with any possible explanations for the observed discrepancies.

Focus: via include the KNN model evaluation, a wider range of possible solutions is illustrated via the investigation of several machine learning techniques for emotion recognition.

Some plot on the KNN Model:



What Makes the Confusion Matrix Up?

1. **Rows:** For a given collection of audio samples, each row denotes the actual emotional label (ground truth). For example, the first row (called "Happy" in your case) displays the proportion of audio samples that genuinely convey cheerful feelings.
2. **Columns:** For a given collection of audio samples, each column shows the emotion category that the model predicted. For instance, the first column displays the number of audio clips the model predicted to be cheerful (labeled "Happy" in your case).
3. **Diagonal Values**: From the top-left corner to the bottom-right corner, the diagonal should ideally include the highest values. The number of audio clips for each emotion category that were accurately categorized is indicated by these values. High values along the diagonal would come from the majority of audio samples being correctly categorized, which is the ideal situation.
4. **Off-Diagonal Values**: These values show instances in which the model incorrectly identified certain audio snippets. For example, a value in the "Happy" column and "Sad" row suggests that the model misclassified some audio samples with the genuine emotion "Sad" as "Happy."

Potential Findings from Your Confusing Matrix:

1. **Overall Accuracy**: The model's overall accuracy can be computed by adding up all of the correctly identified instances (diagonal values) and dividing that total by the entire number of audio clips. A greater total accuracy indicates that the algorithm classified emotions correctly in every category.
2. **Performance by Emotion**: Examine each emotion category's values on the diagonal. While low values on the diagonal imply the model may be having trouble correctly classifying those particular emotions, high values on the diagonal indicate good performance for those particular emotions.
3. **Typical Misclassifications**: Examine the off-diagonal elements for patterns. If various emotions are commonly mistaken with one another (high values in particular rows and columns off the diagonal), it may be a sign that it is hard to tell them apart using the attributes that were retrieved.

MAKING EMOTION LABEL PREDICTIONS WITH A TRAINED SVM MODEL:

After the feature extraction phase (which was previously covered), we used an SVM model to categorize the emotions in the audio clips.

1. **Model Loading:** In your code, the file /content/drive/MyDrive/Audio/svm_model.joblib contained the trained SVM model. Using the training data, this model was first developed, where it discovered the connections between the attributes taken from audio snippets and the associated emotions.
2. **Prediction Function:** The path of an audio clip is entered into the predict_emotion function. Using the extract_features function (described above), it first extracts features from the

audio sample. It then makes the prediction after reshaping the characteristics into a manner that works with the SVM model. The audio clip's anticipated emotion label is then given back.

3. **User Input**: When the user enters the path of an audio clip they wish to analyze, the code prompts them to do so. The prediction function is invoked after it has been supplied in order to process the video and forecast the emotion label.

Analyzing the Outcome:

The emotion category with the highest probability, as determined by the trained SVM model, will be shown by the emotion label printed by the code. It implies that, in comparison to other emotions the model was trained on such as sadness, anger, etc.

CONCLUSION

This study investigated the use of machine learning for automatic emotion recognition in audio. In order to express emotions more richly, we looked at supplementary features (if applicable) and extracted important features **(MFCCs).**

Three models were assessed**: KNN, SVM, and LSTM** (if appropriate). The SVM successfully classified emotions, as evidenced by its accuracy of [Accuracy value]. With strengths in categorizing [List emotions with high diagonal values] and places for improvement with [List emotions with low diagonal values], the KNN model attained an accuracy of [Accuracy value].

These findings demonstrate machine learning's potential for emotion identification. Future research will examine deeper learning structures, extra characteristics, and practical uses.

UPCOMING PROJECTS

Improving Speech Emotion Recognition Techniques

The pursuit of eliciting emotions from audio persists! What comes next is as follows:

- **Feature tweaks:** You can enhance the models' ability to "see" emotions in audio by investigating new features and optimizing those that already exist.
- **Deep Learning Dive:** Robust deep learning systems have the potential to extract much more subtle emotional information from audio data.
- **Model Makeover:** You can squeeze out further accuracy gains by experimenting with different methodologies and model parameters.
- **Data Buffet:** More varied and richer datasets will improve the models' ability to identify emotions.
- **Real-World Impact:** Envision user interfaces that adjust to your emotional state, or systems that comprehend not just words but also the feelings that accompany them. These are but a handful of the fascinating prospects that lie ahead!

REFERENCES

1. https://www.sciencedirect.com/science/article/pii/S2667305323000911
2. https://www.hackersrealm.net/post/speech-emotion-recognition-using-python
3. https://www.projectpro.io/article/speech-emotion-recognition-project-using-machine-learning/573