International Journal of Innovative Research in Engineering & Management (IJIREM)
ISSN: 2350-0557, Volume-9, Issue-6, December 2022
https://doi.org/10.55524/ijirem.2022.9.6.8
Article ID IJIRP-1289, Pages 47-54
www.ijirem.org

Advanced Model Implementation to Recognize Emotion Based Speech with Machine Learning

Dr. Kanakam Siva Rama Prasad[1], N. Srinivasa Rao[2], and B. Sravani[3]

[1,] Professor & Head, Department of Artifical Intelligence & Data Science, Pace Institute of Technology and Sciences, Ongole, Andhra Pradesh, India
[2,] Associate Professor, Department of Artifical Intelligence & Data Science, Pace Institute of Technology and Sciences, Ongole, Andhra Pradesh, India
[3,] Assistant Professor, Department of Artifical Intelligence & Data Science, Pace Institute of Technology and Sciences, Ongole, Andhra Pradesh, India

ABSTRACT- Emotions are essential in developing interpersonal relationships. Emotions make emphasizing with others' problems easy and leads to better communication without misunderstandings. Humans possess the natural ability of understanding others' emotions from their speech, hand gestures, facial expressions etc and react accordingly but, it is impossible for machines to extract and understand emotions unless they are trained to do so. Speech Emotion Recognition is one step towards it, SER uses ML algorithms to forecast the emotion behind a speech. The features which include MEL, MFCC, and Chroma of a set of audio parts are extracted using python libraries and are used to build the ML model. An MLP (Multi-Layer Perceptron) is used which will be mapping the features along with the sound file and predicts the emotion. The project details more about the development and deployment of the model. A technique known as "Speech Emotion Recognition" could identify emotional characteristics in speech signals by computer and contrasts and analysis the characteristics parameters and the emotional change acquired. In current market, speech emotion recognition was emerging crossing field of artificial Intelligence.

KEYWORDS- Multi-Layer Perceptron, Speech Emotion Recognition, NLP, Mel-frequency cepstrum coefficients, modulation spectral features.

I. INTRODUCTION

A voice signal is the quickest and most organic ways that through which people are communicating. The quickest and most effective way to interact between humans and machines is through the use of speech signals. While emotional recognition is an extremely challenging assignment for machines, it comes naturally to people. Therefore, an emotion recognition system uses its understanding of emotion in a way which establishes communication between humans and machines. In speech emotion recognition, the feelings of the male or female speaker can be identified through speech. Due to the existence of various speaking rates, styles, sentences, and speakers, there is an introduction of accosting variability that influences the aspects of speech. Various emotions could be displayed by the It can be challenging to distinguish between these components of speech because they are spoken in different parts of the same utterance for each related mood. Another issue arises because the way in which emotions are expressed relies on the speaker's environment and culture, which also cause disparities in speaking style.

Speech signals can typically be obtained more easily and affordably than other biological signs (such an electrocardiogram). Because of this, most researchers are drawn to Speech Emotion Recognition which seeks to identify a speaker's emotional situation from her speech. The selection of an appropriate Emotional Speech data warehouse, the extraction of useful features as well as building of trustworthy classifiers utilizing machine learning techniques must be handled in order for the SER approach to be successful.

In actuality, the SER system's biggest problem is the extraction of emotional features. Energy, Pitch, Formant Frequency, Linear Prediction Cepstrum Coefficients-LPCC, Mel-Frequency Cepstrum Coefficients- MFCC, and Modulation Spectral Features are only some of the significant speech components that many researchers have considered as containing emotion information (MSFs).

Therefore, the bulk of researchers choose to integrate quite a few tendencies that carry more emotional information. However, using a mixed characteristic set may also end result in immoderate size and redundancy of speech features, complicating learning for maximum Machine Learning algorithms and elevating the threat of overfitting. Therefore, characteristic choice is important to lowering the tiers of characteristic redundancy.

There are different uses in contemporary settings for determining the emotion portrayed in a spoken percept. The study of interactive software between humans and computers is known as human-computer interaction (HCI) [1]. The computer system must be able to comprehend more than simply words in order for an HCI application to be successful. On the other hand, the Internet of Things industry is expanding quickly. Voice-based inputs are used by several real-world IoT apps that are used on a daily basis, including Amazon Alexa, Google Home, and Mycroft. Voice plays a crucial part in IoT applications. According to a recent estimate, by 2022, just voice commands will be used to fully operate around 12% of all IoT apps [6].

It is crucial to understand the speech signal in any situation of these voice exchanges, which may be mono-directional or bidirectional. Additionally, there are applications based on artificial intelligence (AI) and natural language processing (NLP) that build complex systems using IoT and HCI features. One such application that utilizes voice-based commands to operate a number of its features is self-driving cars. In this application, knowing the user's emotional state is quite advantageous. With regard to emergency circumstances in which the user might not be able to clearly give a spoken command, the user's tone of voice can be used to activate several emergency capabilities of the vehicle.

In contact centers, speech emotion recognition is used in a much more straightforward way to forward automated voice calls to customer care representatives for more discussion. Other uses for voice emotion detection systems can be found in humanoids, lie detectors, and criminal department analysis.

A speech has three different types of features: lexical features (the vocabulary used), visual features (the speaker's facial expressions), and sonic features (sound properties like pitch, tone, jitter, etc).

Analyzing one or more of these features can help address the speech emotion recognition issue.

If one wanted to predict emotions from real-time audio, one would need to follow the lexical features, which would require a transcript of the speech and an additional phase of text extraction from voice. The analysis of acoustic features can be done in real-time while the conversation is happening because we only need the audio data. In a similar vein, moving forward with the analysis of visual features would require the excess to the video of the conversations, which might not be practical in every case. Therefore, in this work, we decide to investigate the acoustic properties. Additionally, there are two ways to represent emotions: Classifying feelings into specific categories such as wrath, happiness, boredom, etc.

There are 3 lessons of functions in a speech namely, the lexical functions (the vocabulary used), the visible functions (the expressions the speaker makes) and the acoustic functions (sound homes like pitch, tone, jitter, etc.).

The trouble of speech emotion reputation may be solved with the aid of using reading one or greater of those functions. Choosing to observe the lexical functions could require a transcript of the speech which could in addition require an extra step of text extraction from speech if one desires to are expecting feelings from real-time audio.Similarly,going ahead with reading visible functions could require the extra to the video of the conversations which won't be possible in every case case whilst the evaluation at the acoustic functions may be accomplished in real-time whilst the communique is taking area as We would simply want the audio statistics for carrying out our task. Hence, we pick out to investigate the acoustic functions at this work. Furthermore, the illustration of the feeling may be accomplished in ways: Discrete Classification: Classifying feelings in discrete labels like anger, happiness, boredom, etc. Using dimensions like Valence (on a terrible to nice scale), Activation or Energy (on a low to excessive scale), and Dominance to symbolize feelings (on an energetic to passive scale). Both of those strategies have advantages and drawbacks those strategies. The dimensional approach is greater complicated and gives a great context for prediction. However it's also greater hard to place into exercise due to the fact there is not always as lots annotated audio statistics to be had in dimensional formats. Althoughdiscrete class is easier to recognize and positioned into exercise, It lacks the dimensional illustration's context for the prediction. Due to a paucity of dimensionally annotated statistics with inside the public domain, we followed the discrete class method with inside the present day investigation.

On the subject of voice emotion recognition, various sorts of study are being done. It includes a theoretical definition, categorization of affective state, and a list of ways to express emotions. An SER system built on various classifiers and feature extraction techniques is created to carry out this study. The major objective of this research is to identify a person's emotion using feature extraction and machine learning methods. Audio speech is used to define the different emotional states, such as joyful, sad, furious, surprised, etc. In practise, there are numerous machine learning techniques that may identify emotional states, and each technique has a different significance depending on how the resulting dataset differs from one another.

1. Objective

The selection of an appropriate emotional speech database, the extraction of useful features, and the building of trustworthy classifiers utilizing machine learning techniques must be handled in order for the SER system to be successful. In actuality, the SER system's biggest problem is the extraction of emotional features. The energy, pitch, formant frequency, Linear Prediction Cepstrum Coefficients (LPCC), Mel-Frequency Cepstrum Coefficients (MFCC), and Modulation Spectral Features (MSFs) are only a few of the crucial speech components that many researchers [4] have proposed as containing emotion information.

As a result, the majority of studies favor the use of merging feature sets, which are made up of several types of characteristics and contain more emotional information [6]. Using a combined feature set, however, could result in high dimension and redundancy of speech features, which would be detrimental. For the majority of machine learning algorithms, it complicates the learning process and raises the risk of overfitting. Therefore, feature selection is essential to reducing the dimensions of feature redundancy. [7] Provides a review of feature selection models and methods.

Both feature extraction and feature selection can boost learning efficiency while reducing computational complexity, increasing the generalizability of models, and minimizing the amount of storage needed. Classification is the final stage in speech emotion recognition. On the basis of features extrapolated from the data, it entails categorizing the raw data in the form of an utterance or frame of an utterance into a specific class of emotion. Researchers have recently suggested a variety of classification methods for speech emotion identification, including the Gaussian Mixture Model (GMM), Hidden Markov Model (HMM), Support Vector Machine (SVM), Neural Networks (NN), and Recurrent Neural Networks (RNN). Other classifier types have also been put forth by some researchers, such as the modified brain emotional learning model (BEL) [19], which combines the Multilayer Perceptron (MLP) and the Adaptive Neuro-Fuzzy Inference System (ANFIS) to recognize speech emotions. The Radial Basis Function (RBF) kernel and the linear kernel are combined to present two comparable conceptions in the learning method in the multiple kernel GAUSSIAN PROCESS (GP) classification [17], which is another suggested approach. In contrast to the conventional approach, the voiced signal segment is dealt with in the Voiced Segment Selection (VSS) algorithm, which was previously proposed in [20].

II. LITERATURE SURVEY

Numerous factors in the voice signal that reflect emotional traits are present. What features should be employed is one of the challenging issues in emotion recognition. Many common parameters, including energy, pitch, formant, and some spectrum information, including Linear Prediction Coefficients (LPC), Mel-Frequency Spectral Coefficients (MFCC), and modulation spectral features, are retrieved in current research. To extract the emotional aspects for this work, we have chosen MFCC and Modulation Spectral Features.

After feature extraction, classifiers are used to define the emotion of the speech from the saved audios. Many machine learning classifiers are used to find the emotions of a speech where different algorithms vary from another. The classifiers which are used in speech emotion recognition are Multivariate Linear Regression (MLR), Support Vector Machines (SVM), and Recurrent Neural Networks (RNN).

In numerous branches of psychology, affective science, and emotion research, the classification of emotions has long been a contentious issue. It is mostly based on the categorical (also known as discrete) and dimensional techniques (termed continuous). The first method uses a discrete set of classes to define emotions. To ascertain which emotions are fundamental, numerous theories have undertaken studies. One of the most well-known examples is Ekman, who listed six fundamental emotions: anger, disgust, fear, happiness, sadness, and surprise. In contrast to being a distinct emotional state, he explains that each emotion functions as a distinctive category. Emotions are categorized by axes and comprise a variety of psychological characteristics. For the classification of discrete emotions, there are numerous machine learning techniques that have been used. There are numerous machine learning methods that have been tried for discrete emotion classification, although there is no clear cut best approach to utilize. Every method has benefits and drawbacks of its own.

Emotion plays a significant role in daily human interactions. It helps to match and understand the feelings of others. Speech Emotion recognition aims to recognize the emotional state of the speaker from his/her voice. The speech emotion recognition system uses audio data. It takes a part of speech as input and then determines in what emotions the speaker is speaking. We can identify the emotions like sad, happy, surprised, angry, etc. The best example of it can be seen at call centers. If you ever noticed, call center employees never talk in the same manner, their way of pitching /talking to the customers. Three key issues need to be addressed for a successful SER system:

- Choice of good emotional database, Extracting effective features, designing reliable classifiers
- To achieve this study, an SER system, based on different classifiers and methods for feature extraction
- Mel-frequency Cepstral coefficients (MFCC) modulation spectral(MS) are extracted from speech signals and used to train different classifiers
- Multivariate linear regression classification (MLR)
- Support vector machines (SVM)
- Recurrent Neural Networks (RNN)

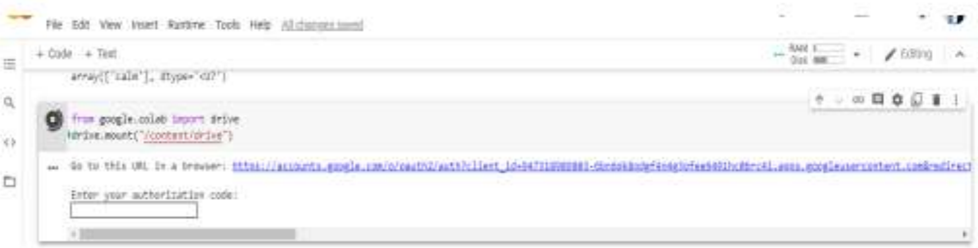
Training the system for correct recognition of speech and testing it to determine the accuracy and accuracy and consistency of recognized character can make speech recognition a computationally heavy task. Here we choose to compare the performance of three different classifiers. The Market size of both hardware and software for speech recognition has reached 55 billion dollars in 2016 and it continues to grow approximately 11% year. The obtained result shows that the proposed system can reliably identify the single emotion from the speech samples. The performance highly depends on the emotional speech samples. So, it is another future enhancement is that it can be applied to the biggest set of modules.

III. IMPLEMENTATION

A. Mounting Google Drive

If we want to use any kind of storage device with our computer or desktop firstly our operating system must be made accessible through the computer file system. This process is called mounting. One can be able to access media only on mounted media. Steps required to mount the google drive with that of the google colab are

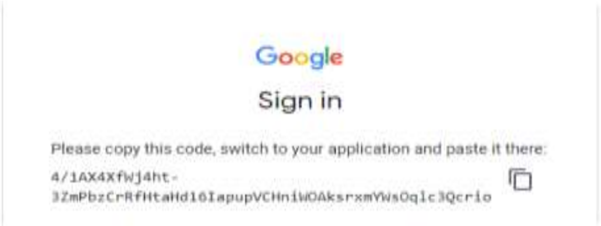
- Upload the data to Google Drive.
- Run the following script in the colab shell



```
array(['calm'], dtype='<U7')

from google.colab import drive
!drive.mount("/content/drive")

Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdf4n4g3qfee5491hc0brc4l.apps.googleusercontent.com&redirect

Enter your authorization code:
```

Figure 1: Running a Script in the Colab shell

3. Copy the authorization code of your account.



Google

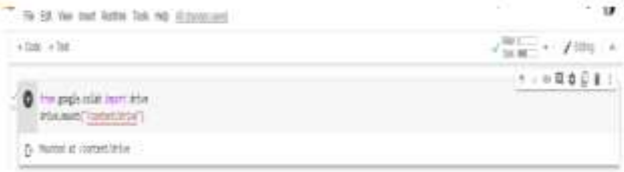
Sign in

Please copy this code, switch to your application and paste it there:

```
4/1AX4XfWj4ht-3ZmPbzCrRfHtaHd16IapupVCHniWOAksrxmYWsOqlc3Qcrio
```

Figure 2: Copying the authorization code

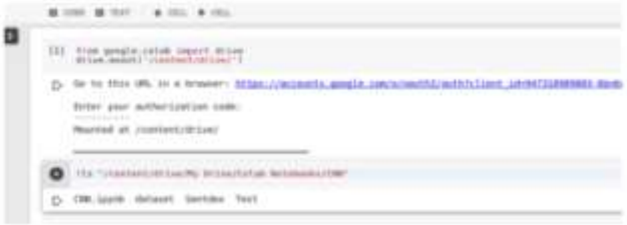
4. Paste the authorization code into the output shell.



```
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive
```

Figure 3: Pasting the authorization code

5. Now the Google Drive is mounted to the mentioned location



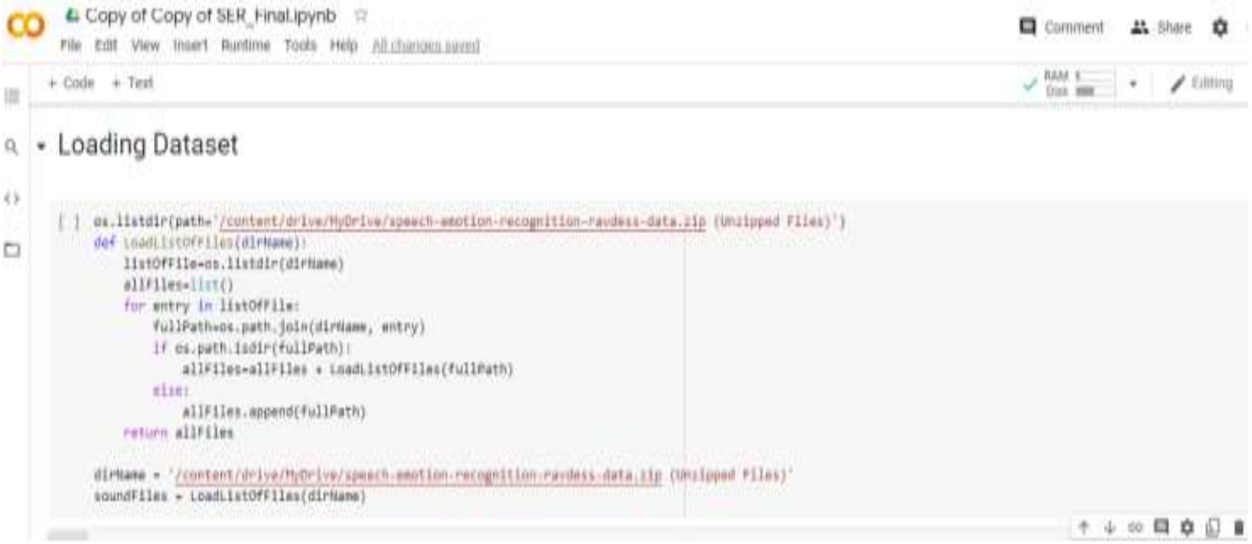
```
[1]  from google.colab import drive
     drive.mount('/content/drive/')

Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6

Enter your authorization code:
··········
Mounted at /content/drive/

!ls "/content/drive/My Drive/Colab Notebooks/CNN"

CNN.ipynb  dataset  Sentdex  Test
```

Figure 4: Mounted the Google Drive

B. Loading The Data

1.Install the librosa library.

2.Then the audio file is loaded into a NumPy array after being sampled at a particular sample rate



CO
Copy of Copy of SER_Final.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing

Loading Dataset

```
[ ] os.listdir(path='/content/drive/MyDrive/speech-emotion-recognition-ravdess-data.zip (Unzipped Files)')
    def loadListOfFiles(dirName):
        listOfFile=os.listdir(dirName)
        allFiles=list()
        for entry in listOfFile:
            fullPath=os.path.join(dirName, entry)
            if os.path.isdir(fullPath):
                allFiles=allFiles + loadListOfFiles(fullPath)
            else:
                allFiles.append(fullPath)
        return allFiles

    dirName = '/content/drive/MyDrive/speech-emotion-recognition-ravdess-data.zip (Unzipped Files)'
    soundFiles = LoadListOfFiles(dirName)
```

Figure 5: Loading the dataset

Emotion reg part 2.pdf

Fig 5.1.2: (1) Loading the dataset

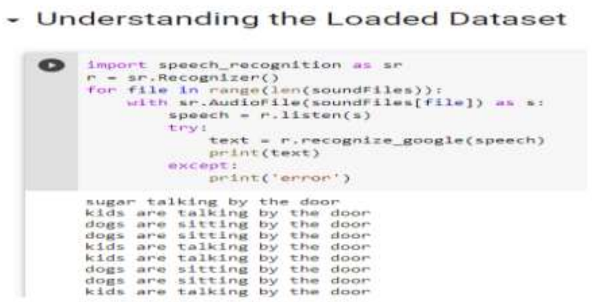
Fig 5.1.2: (2) Loading the dataset from the drive

Figure 6: Loading the dataset from the drive

Figure 7: Audio Files of Ravdass Dataset

C. *Understanding The Loaded Dataset By Converting Speech Into Text*

In this the loaded dataset in the form of audio clip files are converted into text for understanding purpose.


Understanding the Loaded Dataset

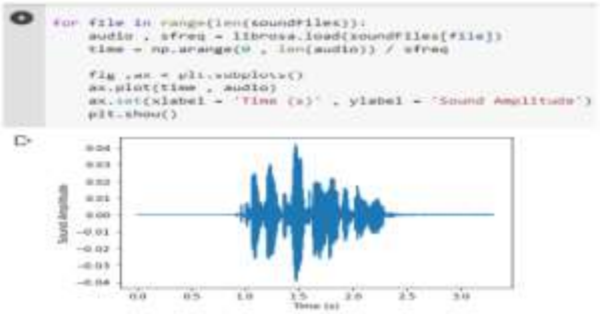
```
import speech_recognition as sr
r = sr.Recognizer()
for file in range(len(soundFiles)):
    with sr.AudioFile(soundFiles[file]) as s:
        speech = r.listen(s)
        try:
            text = r.recognize_google(speech)
            print(text)
        except:
            print('error')
```

```
sugar talking by the door
kids are talking by the door
dogs are sitting by the door
dogs are sitting by the door
kids are talking by the door
kids are talking by the door
dogs are sitting by the door
dogs are sitting by the door
kids are talking by the door
```

Figure 8: Understanding the dataset by converting into speech

D. *Plotting The Audio Part Files*

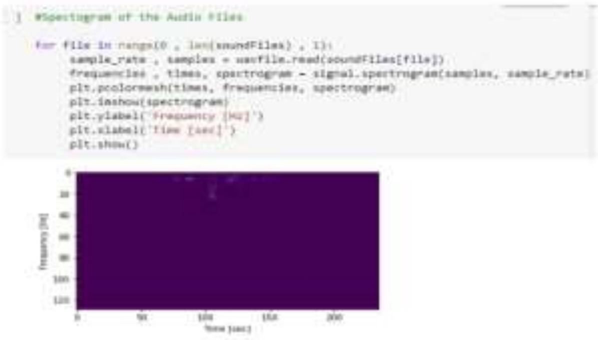
After loading the dataset, the audio part files present in the dataset are plotted in the form of spectrum. The form of spectrum.



```
for file in range(len(soundFiles)):
    audio , sfreq = librosa.load(soundFiles[file])
    time = np.arange(0 , len(audio)) / sfreq

    fig ,ax = plt.subplots()
    ax.plot(time , audio)
    ax.set(xlabel = 'Time (s)' , ylabel = 'Sound Amplitude')
    plt.show()
```

Figure 9: Plotting the audio files

Another is plotted against the frequency and time

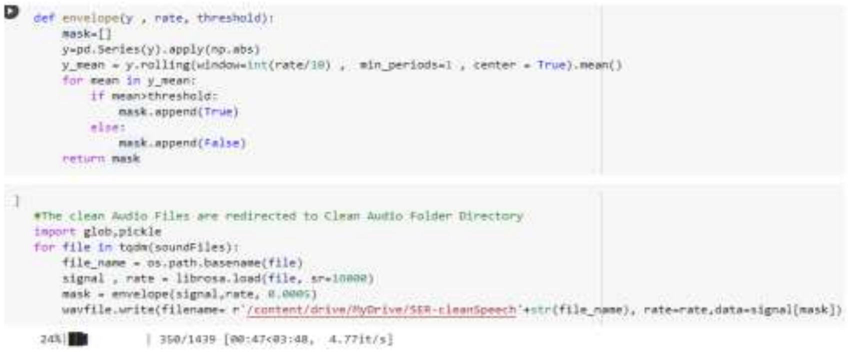


```
#Spectogram of the Audio Files

for file in range(0 , len(soundFiles) , 1):
    sample_rate , samples = wavfile.read(soundFiles[file])
    frequencies , times, spectrogram = signal.spectrogram(samples, sample_rate)
    plt.pcolormesh(times, frequencies, spectrogram)
    plt.imshow(spectrogram)
    plt.ylabel('Frequency [Hz]')
    plt.xlabel('Time [sec]')
    plt.show()
```

Figure 10: Plotting the files against frequency and time.

E. *Cleaning The Dataset*

Before extracting the features from the loaded dataset it has to be free from noise, disturbances, etc. The one which is below the threshold value will be eliminated and the one which is above the threshold will be included.



```
def envelope(y , rate, threshold):
    mask=[]
    y=pd.Series(y).apply(np.abs)
    y_mean = y.rolling(window=int(rate/10) ,  min_periods=1 , center = True).mean()
    for mean in y_mean:
        if mean>threshold:
            mask.append(True)
        else:
            mask.append(False)
    return mask
```

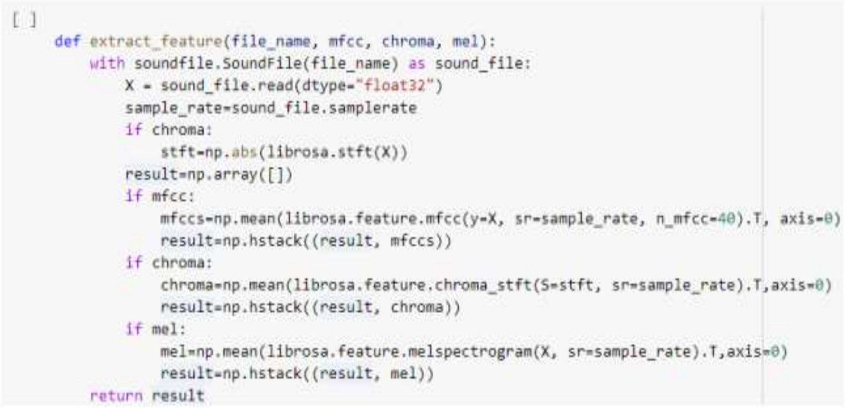
```
#The clean Audio Files are redirected to Clean Audio Folder Directory
import glob,pickle
for file in tqdm(soundFiles):
    file_name = os.path.basename(file)
    signal , rate = librosa.load(file, sr=16000)
    mask = envelope(signal,rate, 0.0005)
    wavfile.write(filename= r'/content/drive/MyDrive/SER-cleanSpeech'+str(file_name), rate=rate,data=signal[mask])
```

```
 24%|██        | 350/1439 [00:47<03:48,  4.77it/s]
```

Figure 11: Cleaning the dataset

F. Extracting Features From Audio File

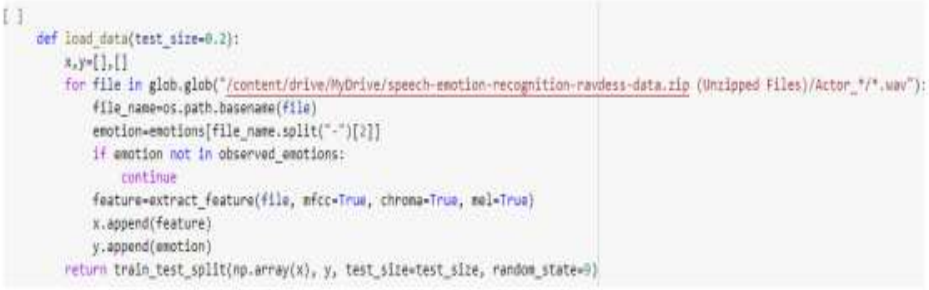
These features have been extracted using mfcc, mel, chroma from the audio files that are loaded. MFCC (Mel-frequency cepstral coefficient) is used mainly when we are working on the audio files of different voice inputs. After this the audio files are processed in order to calculate the coefficients of different frequencies. In the time analysis of signals it will be represented as peaks. Whereas Mel is an logarithmic scale which is used to get the spectrums. It acts as an fourier transform which is mainly computed on overlapping windowed segments. The other feature is chroma, this is a 12 element feature vector stating the pitch present in the signal of audio clips



```
def extract_feature(file_name, mfcc, chroma, mel):
    with soundfile.SoundFile(file_name) as sound_file:
        X = sound_file.read(dtype="float32")
        sample_rate=sound_file.samplerate
        if chroma:
            stft=np.abs(librosa.stft(X))
        result=np.array([])
        if mfcc:
            mfccs=np.mean(librosa.feature.mfcc(y=X, sr=sample_rate, n_mfcc=40).T, axis=0)
            result=np.hstack((result, mfccs))
        if chroma:
            chroma=np.mean(librosa.feature.chroma_stft(S=stft, sr=sample_rate).T,axis=0)
            result=np.hstack((result, chroma))
        if mel:
            mel=np.mean(librosa.feature.melspectrogram(X, sr=sample_rate).T,axis=0)
            result=np.hstack((result, mel))
    return result
```

Figure 12: Extracting features

G. Loading The Dataafter Extraction From Audio Files

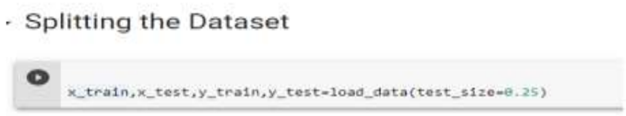


```
def load_data(test_size=0.2):
    x,y=[],[]
    for file in glob.glob("/content/drive/MyDrive/speech-emotion-recognition-ravdess-data.zip (Unzipped Files)/Actor_*/*.wav"):
        file_name=os.path.basename(file)
        emotion=emotions[file_name.split("-")[2]]
        if emotion not in observed_emotions:
            continue
        feature=extract_feature(file, mfcc=True, chroma=True, mel=True)
        x.append(feature)
        y.append(emotion)
    return train_test_split(np.array(x), y, test_size=test_size, random_state=9)
```

Figure 13: Loading the data after extraction

IV. SPLITTING THE DATA

In this the data set is classified for training and testing purposes. This mainly helps us in evaluating the model. In this classification the majority of the dataset is given for training and a smaller portion the data will be given for testing. Using the same data for training and testing will help us to reduce the errors. With this it gets easy for us to understand the model.

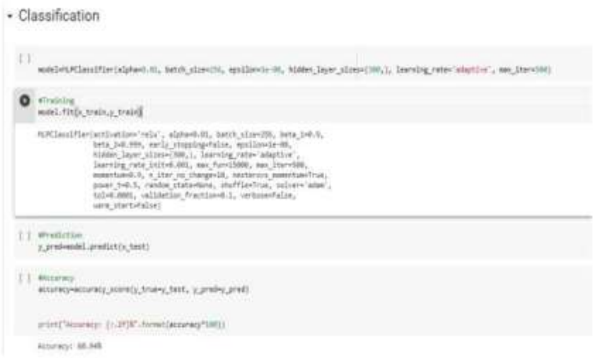


Splitting the Dataset

```
x_train,x_test,y_train,y_test=load_data(test_size=0.25)
```

Figure 14: Splitting the data

In classification we use MLP classifiers for processing purposes. MLP consists of input layers, hidden layers, output layers. In which inputs will be given in the form of batches and are continuously processed to hidden layers. Hidden layers after getting the inputs from the input layer they perform continuous iterations and match them to the respective output.



Classification

```
model=MLPClassifier(alpha=0.01, batch_size=256, epsilon=1e-08, hidden_layer_sizes=(300,), learning_rate='adaptive', max_iter=500)
```

```
#Training
model.fit(x_train,y_train)
```

```
MLPClassifier(activation='relu', alpha=0.01, batch_size=256, beta_1=0.9,
              beta_2=0.999, early_stopping=False, epsilon=1e-08,
              hidden_layer_sizes=(300,), learning_rate='adaptive',
              learning_rate_init=0.001, max_fun=15000, max_iter=500,
              momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
              power_t=0.5, random_state=None, shuffle=True, solver='adam',
              tol=0.0001, validation_fraction=0.1, verbose=False,
              warm_start=False)
```

```
#Prediction
y_pred=model.predict(x_test)
```

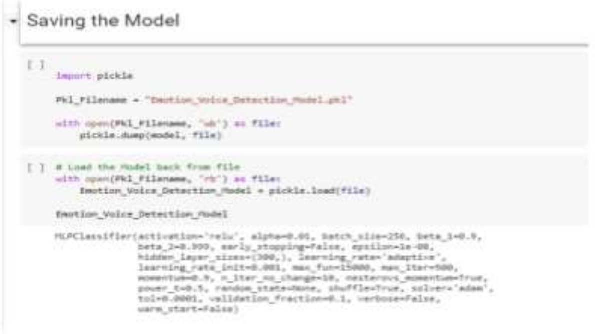
```
#Accuracy
accuracy=accuracy_score(y_true=y_test, y_pred=y_pred)


print("Accuracy: {:.2f}%".format(accuracy*100))
```

```
Accuracy: 68.84%
```

Figure 15: Classifying the splitted data

H. Save The Model

After classification of data now save the model which is obtained.



Saving the Model

```
import pickle

Pkl_Filename = "Emotion_Voice_Detection_Model.pkl"

with open(Pkl_Filename, 'wb') as file:
    pickle.dump(model, file)
```

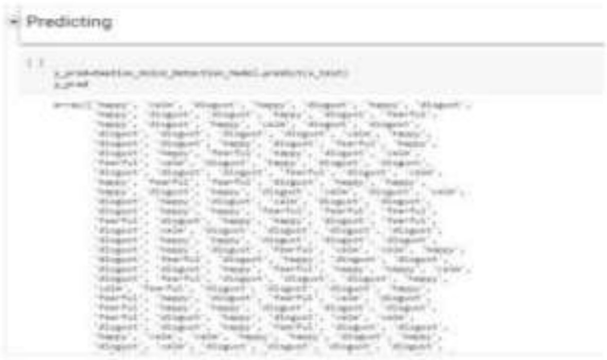
```
# Load the Model back from file
with open(Pkl_Filename, 'rb') as file:
    Emotion_Voice_Detection_Model = pickle.load(file)

Emotion_Voice_Detection_Model
```

```
MLPClassifier(activation='relu', alpha=0.01, batch_size=256, beta_1=0.9,
              beta_2=0.999, early_stopping=False, epsilon=1e-08,
              hidden_layer_sizes=(300,), learning_rate='adaptive',
              learning_rate_init=0.001, max_fun=15000, max_iter=500,
              momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
              power_t=0.5, random_state=None, shuffle=True, solver='adam',
              tol=0.0001, validation_fraction=0.1, verbose=False,
              warm_start=False)
```

Figure 16: Saving the model

Prediction

Now after saving the model let us run the model and check the prediction of the output obtained.



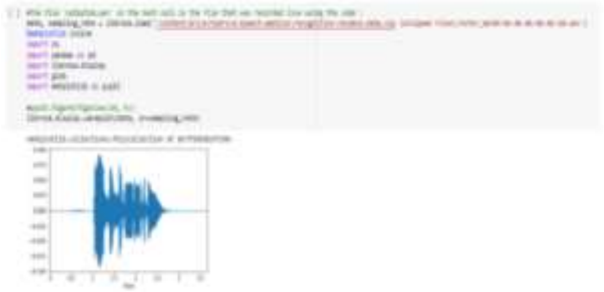
Predicting

```
y_pred=Emotion_Voice_Detection_Model.predict(x_test)
y_pred
```

```
array(['happy', 'calm', [illegible]
```

Figure 17: Checking the prediction of output

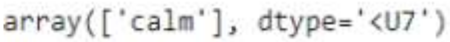
Live Prediction:



```
[illegible]
```

Figure 18: Live prediction of the recorded voice

Output:



```
array(['calm'], dtype='<U7')
```

V. CONCLUSION

The work of building the model was difficult because it required numerous trial-and-error techniques, tuning, etc. The model has received extensive training to differentiate between male and female sounds, and it does so with average accuracy. With more than 70% accuracy, the model was trained to recognize emotions. Including more audio training files helps improve accuracy.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] Geoffrey Z, Picheny M (2004) Advances in large vocabulary continuous speech recognition. Adv Comput 60:249–291CrossRefGoogle Scholar

[2] Campbell N (2007) On the use of nonverbal speech sounds in human communication. In: Campbell N (ed) Verbal and nonverbal communication behaviours LNAI, vol 4775. Springer, New York, pp 117–128CrossRefGoogle Scholar

[3] Laver J (1980) The phonetic description of voice quality. Cambridge University Press, CambridgeGoogle Scholar

[4] Roach P, Stibbard R, Osborne J, Arnfield S, Setter J (1998) Transcription of prosodic and paralinguistic features of emotional speech. J Int Phonetic Assoc 28(1–2):83–94CrossRefGoogle Scholar

[5] Crystal D (1969) Prosodic systems and intonation in English: David Crystal. Cambridge University Press, CambridgeGoogle Scholar

[6] Carlson R (2002) Dialogue system. Slide presentation, speech technology, GSLT, Göteborg, Oct 2002

[7] Rolf C, Granström B (1997) Speech synthesis. In: Hardcastle WJ, Laver J (eds) The handbook of phonetic sciences. Blackwell Publishers Ltd, Oxford, pp 768–788Google Scholar