**PROJECT REPORT**

**On**

**“SPEECH EMOTION RECOGNITION”**

**(CSE Semester VI Mini Project)**



**Submitted By-** Vanshika Chaudhary

**University Roll Number-** 1918093

CSE-D4-SEMESTER-VI

**Session: -** 2022-23

**Resource Person-** Mr. Yuvraj Joshi

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GRAPHIC ERA UNIVERSITY, DEHRADUN**

**CERTIFICATE**

Certified that Ms. Vanshika Chaudhary (roll number-1918093) has completed her Mini Project titled **“Speech Emotion Recognition”** for the CSE VI Semester Mini Project in Graphic Era University, Dehradun. The project carried out by the student is their own work as best of my knowledge.

**Date: - 30th June, 2022.**

Mr. Saurabh Mishra Mr. Yuvaj Joshi

**Project Co-Ordinator Project Guide**

Resource Person

(CSE Department) (CSE Department)

GEU Dehradun GEU Dehradun

**ACKNOWLEDGEMENT**

I would like to express my gratitude to The Almighty Shiva Baba, the most Beneficent and the most Merciful, for completion of project.

I wish to thank my parents for their continuing support and encouragement. I also wish to thank them for providing me with the opportunity to reach this far in my studies.

I would like to thank particularly my project Co-ordinator Mr. Saurabh Mishra and my Project Guide Mr. Yuvraj Joshi for their patience, support and encouragement throughout the completion of this project and having faith in me.

At last, but not the least I greatly indebted to all other persons who directly or indirectly helped me during this work.

**Vanshika Chaudhary**

**Roll Number- 1918093**

**CSE-D4-VI-SEM**

**Session: -2022-2023**

**GEU Dehradun**

**TABLE OF CONTENTS**

1. **Introduction 5-9**

1.1 About the Project

1.2 Requirements of the Project

1.3 References

1. **The Code 10-13**
2. **Output 14**
3. **Conclusion 15**

**1. INTRODUCTION**

**1.1 About the Project**

**Project Highlights:** This is the project that is designed to detect human emotions from their voice. Things I will learn by completing this project:

• How to use Librosa

• How to extract features from the voice

• How to split the data into training and testing sets

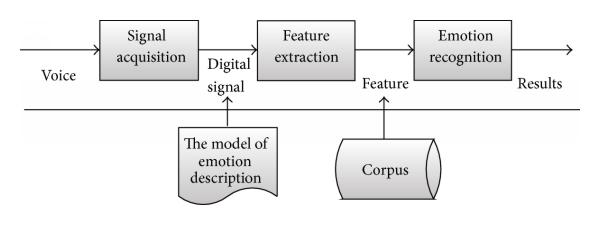
• How to use MlP classifier to train our model

• How the model able to predict the accuracy

**Description**

Speech Emotion Recognition, abbreviated as SER, is the act of attempting to recognize human emotion and affective states from speech. This is capitalizing on the fact that voice often reflects underlying emotion through tone and pitch. This is also the phenomenon that animals like dogs and horses employ to be able to understand human emotion.

SER is tough because emotions are subjective and annotating audio is challenging.



**Software and Libraries**

This project uses the following software and Python libraries: 1. Python 3.8

2. Pickle

3. Librosa

4. pyaudio

5. Sklearn

6. soundfile

You will also need to python installed in your system and pycharm and any other ide to code like VScode and other you also need to install all these packages in your system.

**What to Install?**

This project requires Python and the following Python libraries installed:

• Librosa

• sklearn

• Pyaudio

• soundfile

**Coding Part**

In this Python mini project, we will use the libraries librosa, soundfile, and sklearn (among others) to build a model using an MLPClassifier. This will be able to recognize emotion from sound files.

**LIBRARIES USED:**

1. Librosa: It is a python package for music and audio analysis. It is basically used when we work with audio data like in music generation, emotion extraction(using LSTM’s).It provides the building blocks necessary to create the music information retrieval systems.

2. SKLeran: Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

3. PYAUDIO: Py Audio provides Python bindings for PortAudio, the cross-platform audio I/O library. With PyAudio, you can easily use Python to play and record audio on a variety of platforms. PyAudio is inspired by: .... tkSnack: cross-platform sound toolkit for Tcl/Tk and Python.

4. Soundfile: PySoundFile is a Python module used for reading and writing audio files, see an audio file as NumPy array including of pitches and all. This module can read the audio file i.e. it extracts the NumPy array from audio(.wav files) and able to write it too.

**HOW TO RUN:**

To run the project give the run command once our model.

**1.2 Requirements of the Project**

1.2.1 Hardware Requirements

1. Memory and disk space required per user: 512MB RAM + 1GB of disk + .5 CPU core.

2. Server overhead: 2-4GB or 10% system overhead (whatever is larger), .5 CPU cores, 10GB disk space.

1.2.2 Software Requirements

Latest version of python and pycharm all the packages must be installed in the machine.

**1.3 REFERENCES**

1. Youtube

2. Udemy

3. Google

**2. THE CODE**

import librosa

import soundfile

import os, glob, pickle

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.neural\_network import MLPClassifier

from sklearn.metrics import accuracy\_score

#Extract features (mfcc, chroma, mel) from a sound file

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

Firstly, Install the required libraries used in the project. Define a function **extract\_feature** to extract the mfcc, chroma, and mel features from a sound file. This function takes 4 parameters- the file name and three Boolean parameters for the three features:

• mfcc: Mel Frequency Cepstral Coefficient, represents the short term power spectrum of a sound

• chroma: Pertains to the 12 different pitch classes

• mel: Mel Spectrogram Frequency.

# Emotions in the dataset being used

emotions={

'01':'neutral',

'02':'calm',

'03':'happy',

'04':'sad',

'05':'angry',

'06':'fearful',

'07':'disgust',

'08':'surprised'

}

#Emotions we observe

observed\_emotions=['calm', 'happy', 'fearful', 'disgust']

Now, let’s define a dictionary to hold numbers and the emotions available in the RAVDESS dataset, and a list to hold those we want- calm, happy, fearful, disgust.

#Load the data and extract features for each sound file

def load\_data(test\_size=0.2):

x,y=[],[]

#glob used here to return the file paths matching to given pattern

for file in glob.glob("/content/drive/MyDrive/Speech\_detection\_dataset/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)

#Split the dataset

x\_train,x\_test,y\_train,y\_test=load\_data(test\_size=0.25)

#Initialize the Multi Layer Perceptron Classifier

model=MLPClassifier(alpha=0.01, batch\_size=256, epsilon=1e-08, hidden\_layer\_sizes=(300,), learning\_rate='adaptive', max\_iter=500)

Now, let’s load the data with a function load\_data() – this takes in the relative size of the test set as parameter. x and y are empty lists; we’ll use the glob() function from the glob module to get all the pathnames for the sound files in our dataset. The pattern we use for this is: “D:\\DataFlair\\ravdess data\\Actor\_\*\\\*.wav”.

#Train the model

model.fit(x\_train,y\_train)

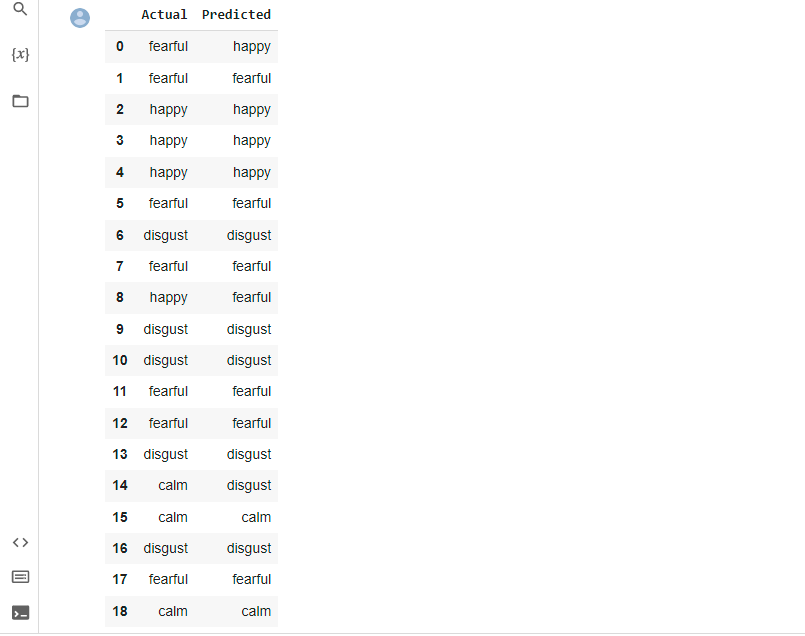
#Predict for the test set

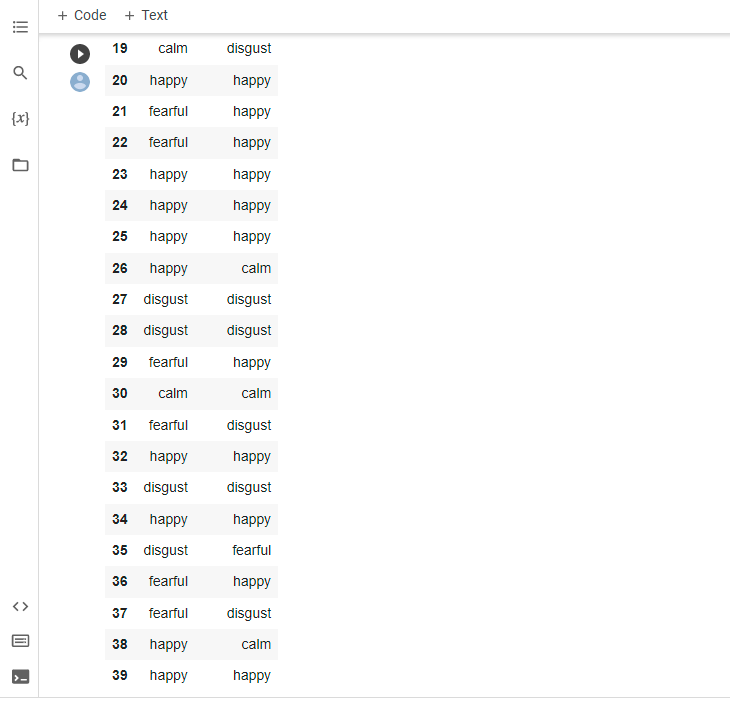
y\_pred=model.predict(x\_test)

#Calculate the accuracy of our model

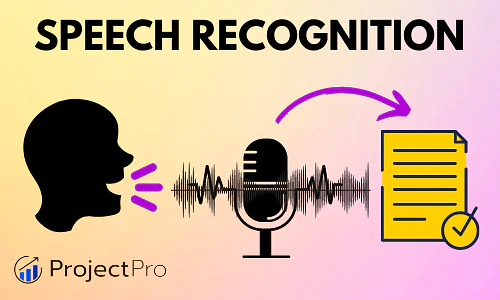
#accuracy=accuracy\_score(y\_true=y\_test, y\_pred=y\_pred)

1. **OUTPUT**





**CONCLUSION**



Applications of simple speech recognition are widespread – YouTube auto-generated subtitles, live speech transcripts, transcripts for online courses, and intelligent voice-assisted chatbots like Alexa and Siri. Because of this, heavily dedicated research has yielded lucrative and fruitful results – YouTube auto-generated subtitles improve each year. However, applications of speech emotion recognition are more nuanced and add a newer dimension to the use of AI and how it can make our lives easier to improve them.

A very recent application of SER has emerged from the sudden rise in online learning where educators can observe a student’s response in class and highlight pointers that might help them assist the student’s education. Another upcoming use is to evaluate candidates applying for managerial positions by analysing their responses during audio or video interviews. Their confidence or trepidations can be quantitatively measured for the first-time using SER, and thus hiring managers can select the candidate with the ideal fit.