Speech Emotion Recognition for Enhanced User Experience: A Comparative Analysis of Classification Methods

[1]Samjhana Pokharel (Author)
Department of Computer Science and Engineering
Kathmandu University
Dhulikhel, Nepal

[2]Ujwal Basnet (Author)
Department of Computer Science and Engineering
Kathmandu University
Dhulikhel, Nepal

Abstract:- Speech recognition has gained significant importance in facilitating user interactions with various technologies. Recognizing human emotions and affective states from speech, known as Speech Emotion Recognition (SER), has emerged as a rapidly growing research subject. Unlike humans, machines lack the innate ability to perceive and express emotions. Therefore, leveraging speech signals for emotion detection has become an adaptable and accessible approach. This paper presents a project aimed at classifying emotional states in speech for applications such as call centers, measuring emotional attachment in phone calls, and real-time emotion recognition in online learning. The classification methods employed in this study include Support Vector Machines (SVM), Logistic Regression (LR), and Multi-Layer Perceptron (MLP). The project utilizes features such as Mel-frequency cepstrum coefficients (MFCC), chroma, and mel to extract relevant information from speech signals and train the classifiers. Through a comparative analysis of these classification methods, this research aims to enhance the understanding of speech emotion recognition and contribute to the development of more effective and accurate emotion recognition systems.

Keywords:- *Speech Emotion Recognition, Speech Recognition (SER), Emotion Classification, Support Vector Machines (SVM), Logistic Regression (LR), Multi-Layer Perceptron (MLP), Mel-frequency Cepstrum Coefficients (MFCC), Chroma, Mel Features.*

I. INTRODUCTION

Speech recognition has become increasingly important in recent years as a means of assisting others with ease of use. Several well-known technology companies, including Google, Samsung, and Apple, have used speech recognition to convert human speech into sentences so that their customers may quickly navigate around their products.

Speech emotion recognition, SER, is the act of attempting to recognize human emotion and the associated affective states from speech. This uses the fact that tone and pitch in the voice often indicate underlying emotion. In recent years, emotion recognition has become a rapidly increasing research subject. Machines, unlike humans, lack the ability to perceive and express emotions. Speech, psychological signals, facial expressions, and other modalities can all be used to detect emotions. Speech signals are far more adaptable and simple to acquire than other modalities. Mel-frequency cepstrum coefficients (MFCC), chroma, and mel features are extracted from the speech signals and used to train the classifiers.

Our project aims to classify the emotional state of the speech which can be used in a number of applications like call centers, measuring the degree of emotional attachment from phone calls, real-time emotion recognition in online learning, etc. There are three classifying methods that are used in this project for analyzing emotions (calm, happy, fearful, angry, disgust, surprised) using SVM, Logistic Regression (LR), and Multi-Layer Perceptron (MLP).

➢ *Motivations for Doing the Project*

In today's world, identifying the emotion exhibited in a spoken percept has various applications. Human-Computer Interaction (HCI) is a branch of study that looks into how humans and computers interact with each other. A computer system that understands more than simply words is required for an efficient HCI application. Voice-based inputs are used by several real-world IoT applications, including Amazon Alexa, Google Home, and Mycroft. In IoT applications, voice plays a critical role. According to a recent survey, about 12% of all IoT applications will be completely functional by 2022. Self-driving automobiles are one example of the emerging field that uses voice commands to operate several of its tasks. In emergency scenarios where the user may be unable to offer a clear spoken command, the emotion communicated through the user's tone of voice can be used to activate specific car emergency functions.

➢ *Objectives*

The primary objective of speech emotion recognition is to improve human-machine interaction interface by detecting the emotional state of a person using speech.

II. RELATED WORKS

There are a number of studies done on speech emotion recognition and different companies are doing research and work related to speech emotion recognition directly or as an application for different parts of the work.

audEERING, an audio analysis company based in Germany, that specialises in emotional artificial intelligence. Their team are experts in voice emotion analytics, machine learning and signal processing.

Alexa, is a virtual assistant AI technology developed by Amazon, first used in the Amazon Echo smart speaker and the Echo Dot, Echo Studio and Amazon Tap speakers developed by Amazon Lab, working on detecting emotions like sadness, happiness, anger, etc, for understanding the mental state of a speaker from the sound of your voice.

III. DATASETS

In this project we have used the RAVDESS(Ryerson Audio-Visual Database of Emotional Speech and Song) dataset. It contains 7356 files rated by 246 persons 10 times on emotional validity. The dataset is 24.8 GB from 24 different actors. The dataset is huge so we used the sample rate lowered versions which is around 171 MB. The dataset includes the following emotions: Neutral, Calm, Happy, Sad, Angry, Fearful, Disgust, Surprised.

➢ *File Naming Convention*

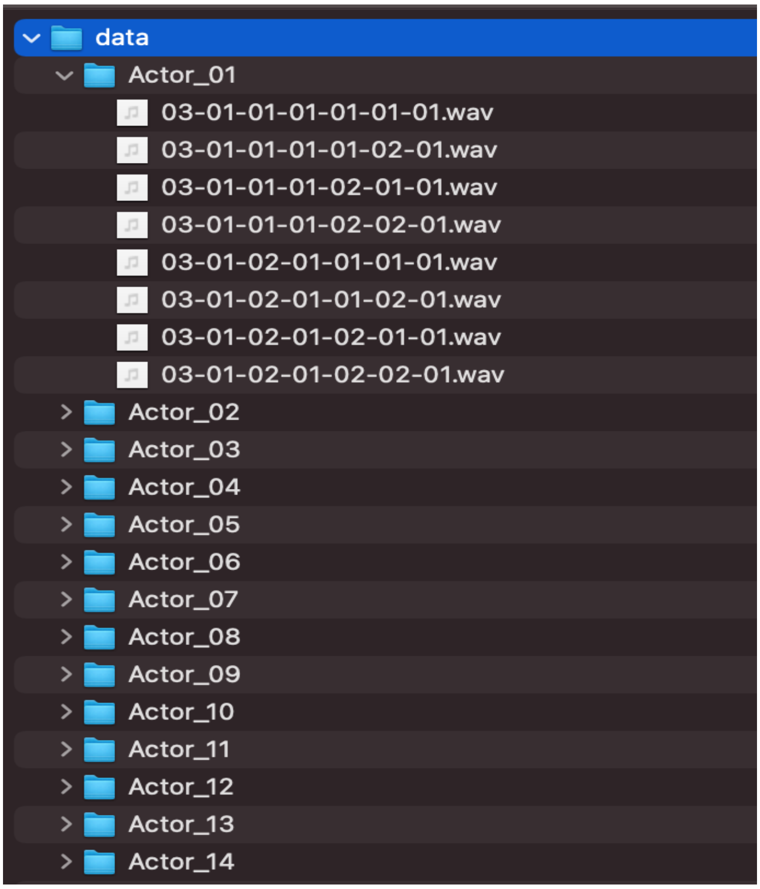
Each of the 1440 files has a unique filename. The filename consists of a 7-part numerical identifier (e.g., 03-01-06-01-02-01-12.wav). These identifiers define the stimulus characteristics:

➢ *Filename Identifiers*

- Modality (01 = full-AV, 02 = video-only, 03 = audio-only).
- Vocal channel (01 = speech, 02 = song).
- Emotion (01 = neutral, 02 = calm, 03 = happy, 04 = sad, 05 = angry, 06 = fearful, 07 = disgust, 08 = surprised).
- Emotional intensity (01 = normal, 02 = strong). NOTE: There is no strong intensity for the 'neutral' emotion.
- Statement (01 = "Kids are talking by the door", 02 = "Dogs are sitting by the door").
- Repetition (01 = 1st repetition, 02 = 2nd repetition).
- Actor (01 to 24. Odd numbered actors are male, even numbered actors are female).

➢ *Filename Example 03-01-06-01-02-01-12.wav*

- Audio-only (03)
- Speech (01)
- Fearful (06)
- Normal intensity (01)
- Statement "dogs" (02)
- 1st Repetition (01)
- 12th Actor (12) Female, as the actor ID number is even.



data
Actor_01
03-01-01-01-01-01-01.wav
03-01-01-01-01-02-01.wav
03-01-01-01-02-01-01.wav
03-01-01-01-02-02-01.wav
03-01-02-01-01-01-01.wav
03-01-02-01-01-02-01.wav
03-01-02-01-02-01-01.wav
03-01-02-01-02-02-01.wav
Actor_02
Actor_03
Actor_04
Actor_05
Actor_06
Actor_07
Actor_08
Actor_09
Actor_10
Actor_11
Actor_12
Actor_13
Actor_14

IV. METHODS AND ALGORITHMS USED

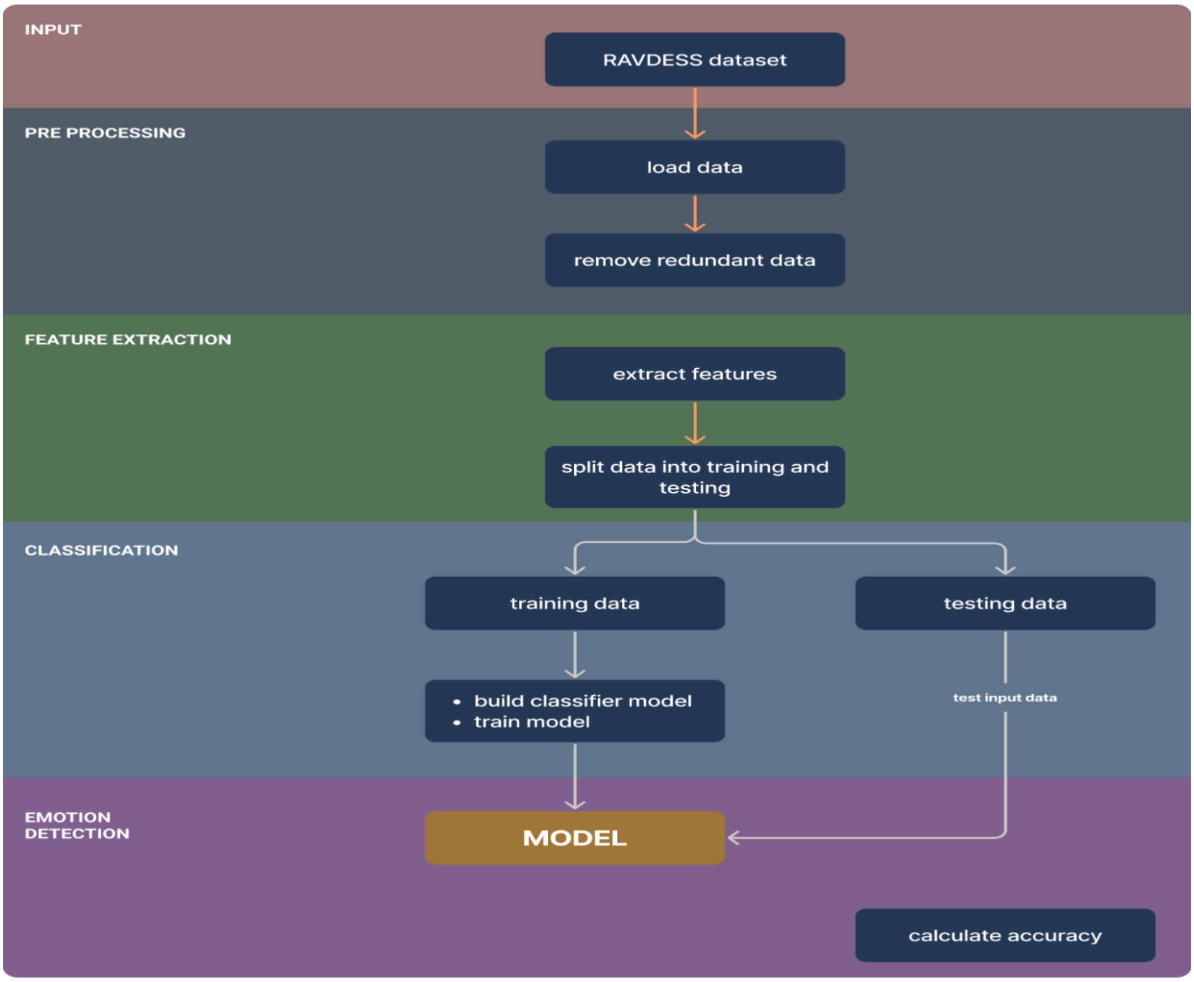


Fig 1 Methodology

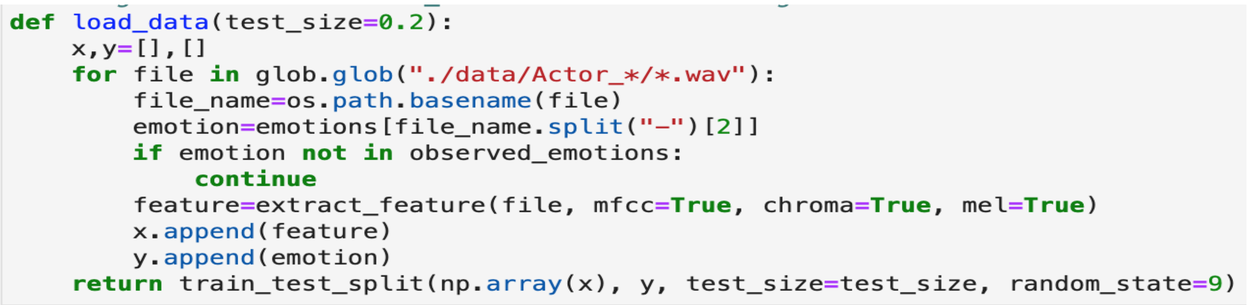
The above figures show the general flow-chart of our project. It consists of 5 different phases. The data is stored in files in the project directory. The files are loaded using different python libraries then unnecessary files are removed. And, we extract different features of sound files like mfcc, mel, chroma, which will be used as features for mapping classifier function. The dataset is then divided into two different sets: testing and training sets. We then build different classifier models. And using the training set, we train the model. After that we use a testing set for different evaluations and accuracy calculations of the model. The whole process is generalized by the above diagram.

- *Phase 1: Data Collection*

The RAVDESS dataset is used in the project. The dataset is downloaded into our system.

Audio files in the directory are loaded using libraries like: os, glob, and soundfile.

We use glob module which finds all the path names matching a specified pattern as the dataset consists of audio files named in some specific pattern, which also consists of the emotion decoded value in file name only. os module is used to get the base name of the file. Then, using soundfile library we read sound files along with the sample rate of the audio.



```
def load_data(test_size=0.2):
    x,y=[],[]
    for file in glob.glob("./data/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)
```

➢ *Phase 2: Extracting Features*

Different features of sound are extracted, Mel Frequency Cepstral Coefficients(MFCCs), Chroma, Mel-Spectrogram.

- *MFCCs*

It is a frequency domain feature of the sound, which captures timbral or textural and the phonetical crucial characteristics of the speech. It is widely used in speech, music genre, and musical instrument classifications.

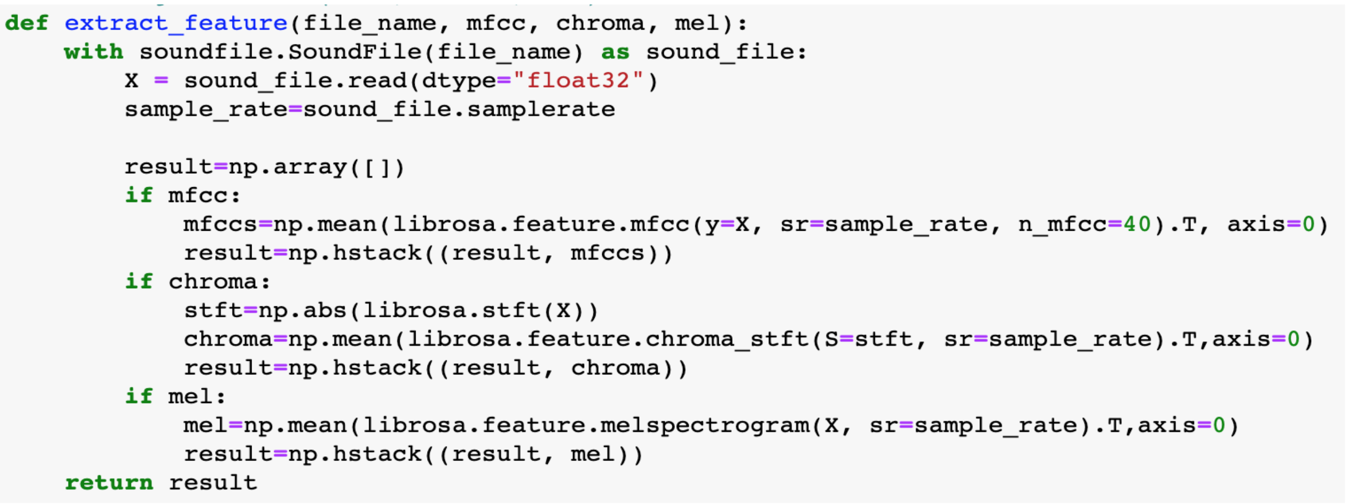
- *Chroma*

Chroma captures harmonic and melodic characteristics of music, while being robust to changes in timbre and instrumentation. It is also referred to as pitch class profiles.

- *Mel-Spectrogram*

A spectrogram where the frequencies are converted to the mel scale. It takes samples of sound files over time to represent audio signals. Then, the audio signal is mapped from time domain into frequency domain using fast Fourier transform then shifted frequency and amplitude to form a spectrogram.

Librosa library is used to extract features from the audio file. Librosa is a python library for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems.



```
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

        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:
            stft=np.abs(librosa.stft(X))
            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
```

➢ *Phase 3: Classification*

In this phase different algorithm models are used for classifying the emotions:

- MLP Classifier
- Logistic Regression
- SVM

The dataset was divided into two sets:

- Training set (80%)
- Testing set (20%)

- *MLP Classifier*

Multi-Layer Perceptron (MLP) is a part of feedforward artificial neural network which consists of an input layer, multiple hidden layers, and an output layer which are connected. Based on adjustments of parameters, biases, weights of the model, the model represents the target function. Activation function that is used during the experiment was relu which makes the model easier to train and often achieves better performance.

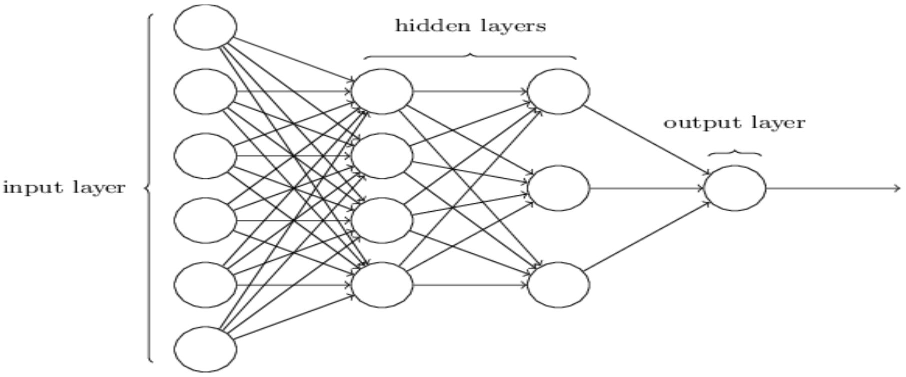


Fig 2 MLP Classifier

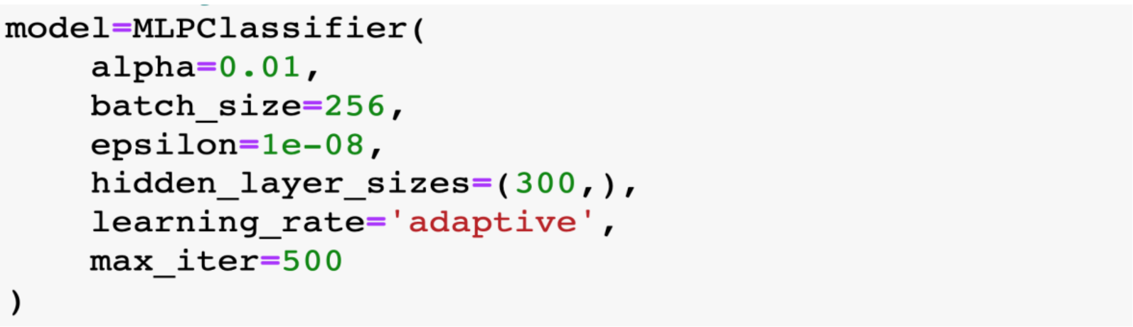
MLP (Multi-Layer Perceptron) Classifier is used to categorize the given data into respective groups. It is capable of approximating boolean and nonlinear functions. It is frequently used in supervised learning problems. The network works on real-values, so the categorical values must be converted into real-value representation.

Following values of parameters were used in our model:

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

- ✓ Alpha: a parameter for regularization term, aka penalty term, that combats overfitting by constraining the size of the weights.
- ✓ Batch Size: the number of samples that will be propagated through the network.
- ✓ Epsilon: value for numerical stability.
- ✓ Hidden Layer Sizes: 1 hidden layers with 300 hidden units,
- ✓ Learning Rate: learning rate for weight updates

'adaptive' keeps the learning rate constant to 'learning_rate_init' as long as training loss keeps decreasing. Each time two consecutive epochs fail to decrease training loss by at least tol, or fail to increase validation score by at least tol if 'early_stopping' is on, the current learning rate is divided by 5.



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

- *Logistic Regression*

The logistic model is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick. This can be extended to model several classes of events such as determining whether an image contains a cat, dog, lion, etc. Each object being detected in the image would be assigned a probability between 0 and 1, with a sum of one. This model is preferable for dependent variable (categorical) data since the data used have small size of output (happy and sad).

This linear relationship can be written in the following mathematical form (where ℓ is the log-odds, b is the base of the logarithm, and β_i are parameters of the model):

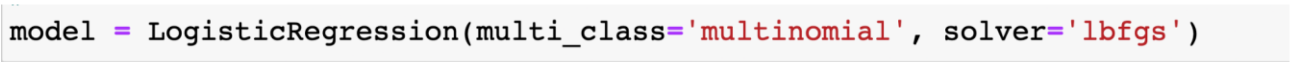
$$\ell = \log_b \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

$$\frac{p}{1-p} = b^{\beta_0 + \beta_1 x_1 + \beta_2 x_2}$$

Following are the parameters used in building the model:

- ✓ multi_class='multinomial',
- ✓ solver='lbfgs'

- ✓ Multi_class: multinomial, extension of logistic regression that adds support for multi-class classification problems.
- ✓ Solver: lbfgs, solver is an algorithm used for optimization problems. In our case, lbfgs is used, it approximates the second derivative matrix updates with gradient evaluations, and stores only the last few updates, so it saves memory, also it isn't super fast with large data sets.



```
model = LogisticRegression(multi_class='multinomial', solver='lbfgs')
```

- *SVM*

SVM (Support Vector Machine) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. It is one of supervised machine learning models that linearly separable binary sets. The goal of this model is to calculate and create a hyperplane that classifies all training vectors. After creating a hyperplane, the next step is to determine the maximum margin between data point and hyperplane which can be called as support vectors.

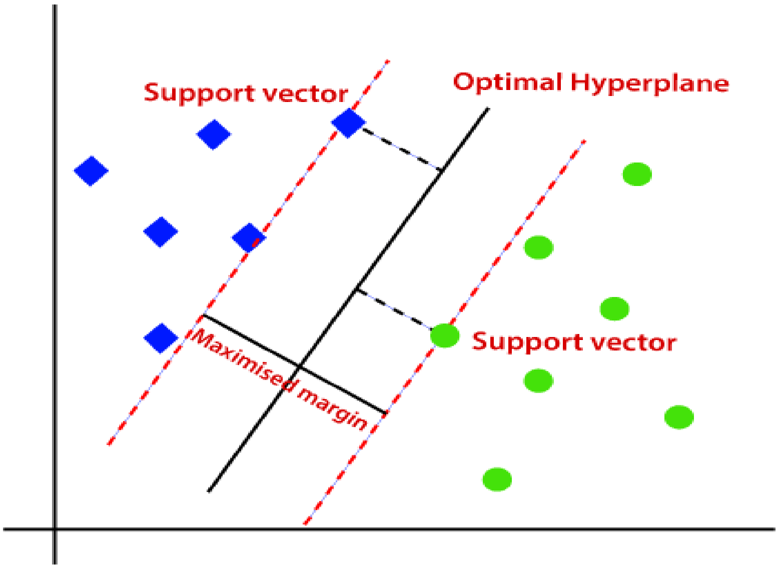
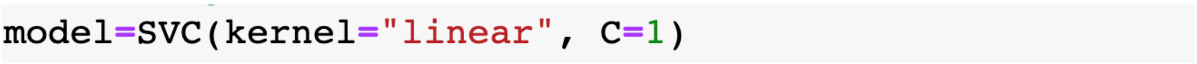


Fig 3 SVM

Following are the parameters used in building model:

- ✓ kernel="linear",
- ✓ C=1

- ✓ kernel="linear", specify kernel type of the algorithm
- ✓ C=1, Regularization parameter. The strength of the regularization is inversely proportional to C and it must be strictly positive.



```
model=SVC(kernel="linear", C=1)
```

- ➢ *Phase 4: Evaluation*

Evaluation of the experiment involves comparison between each model classification report and accuracy. Evaluation of the experiments includes comparison of accuracy between the multiple experiment's of each algorithm and between different algorithms.

V. EXPERIMENTS AND EVALUATIONS

In this phase different algorithm models are used for classifying the emotions: MLP Classifier, Logistic Regression, SVM.

Datasets consisted of following data:

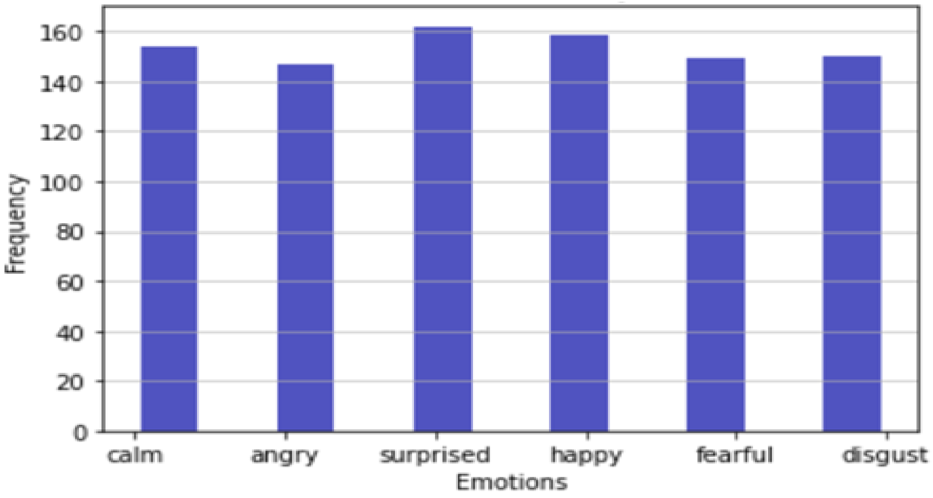


Fig 4 Datasets consisted

- *MLP Classifier*

Table 1 Experiments Conducted with Respective Evaluations:

SN	Experiments	Accuracy	F1 Score	Precision	Recall
I	alpha=0.01, batch_size=256, epsilon=1e-08, hidden_layer_sizes=(300,), learning_rate='adaptive', max_iter=500	0.63	0.63	0.67	0.64
II	hidden_layer_sizes=(300,150,),	0.61	0.59	0.63	0.6
III	hidden_layer_sizes=(600,),	0.7	0.7	0.71	0.71

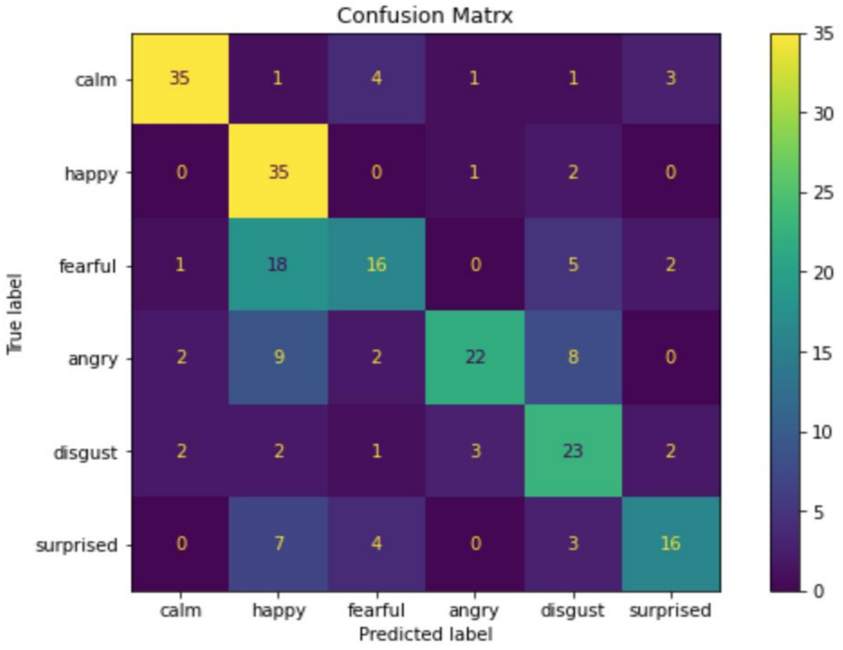


Fig 5 MLP, Experiment I

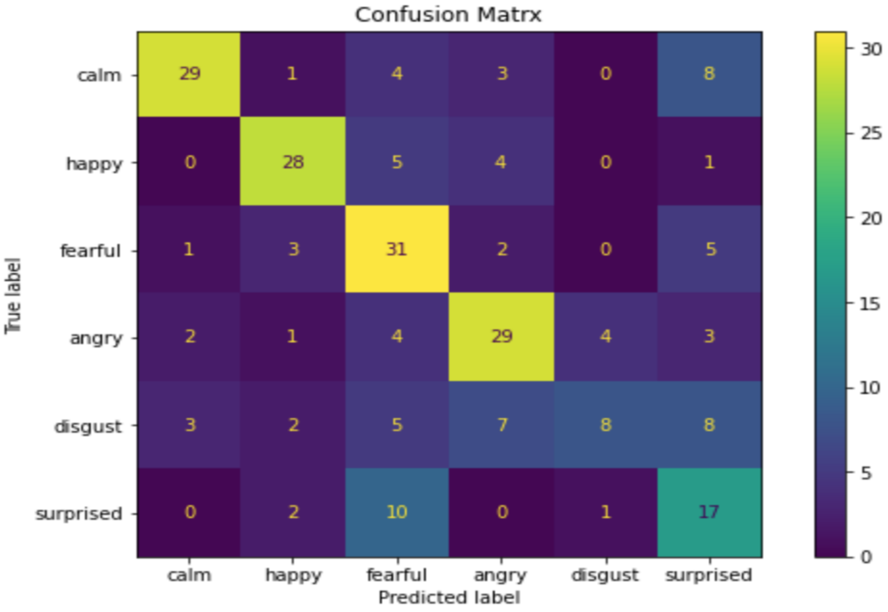


Fig 6 MLP, Experiment II

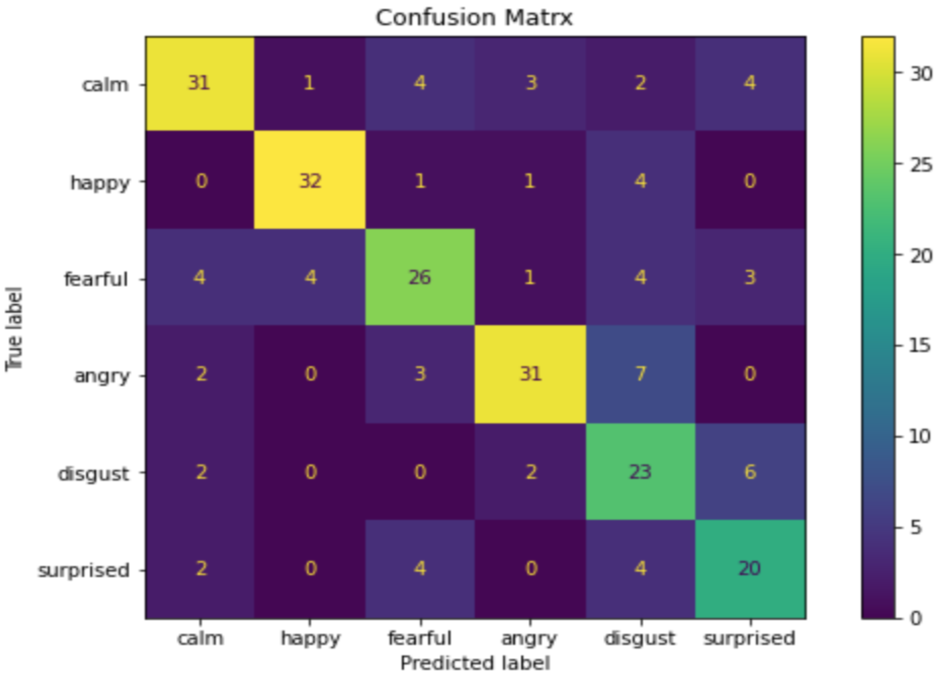


Fig 7 MLP, Experiment III

- *Logistic Regression*

Table 2 Experiments Conducted with Respective Evaluations:

SN	Experiments	Accuracy	F1 Score	Precision	Recall
I	solver='lbfgs'	0.53	0.52	0.53	0.53
II	solver='saga'	0.50	0.50	0.53	0.50
III	solver='newton-cg'	0.58	0.58	0.58	0.58

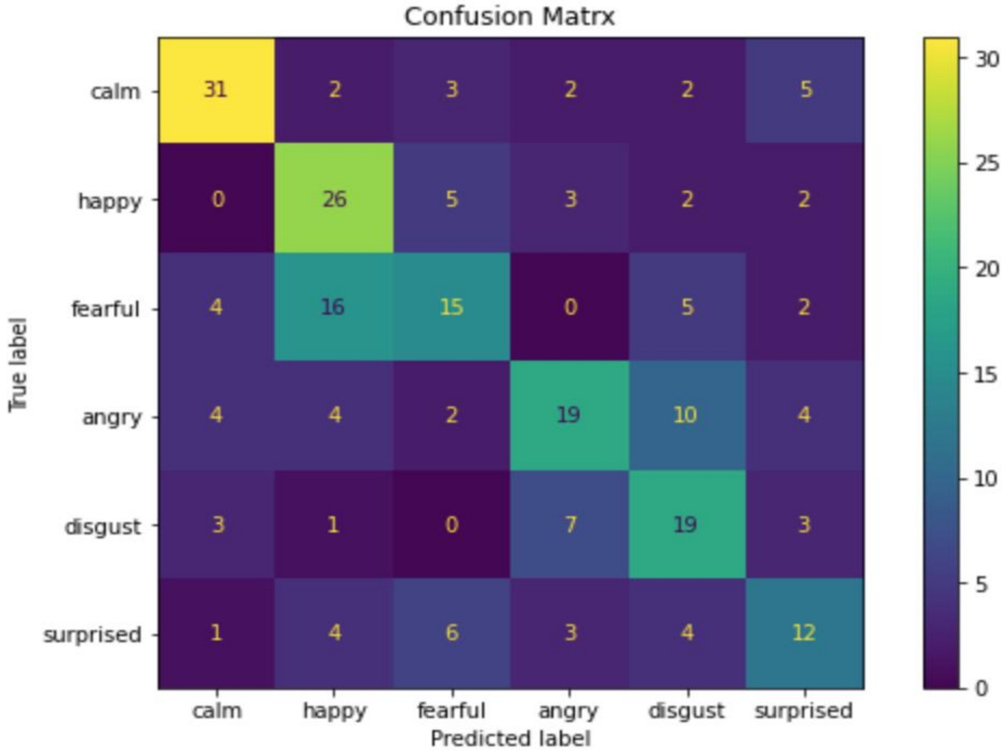


Fig 8 LR, Experiment I

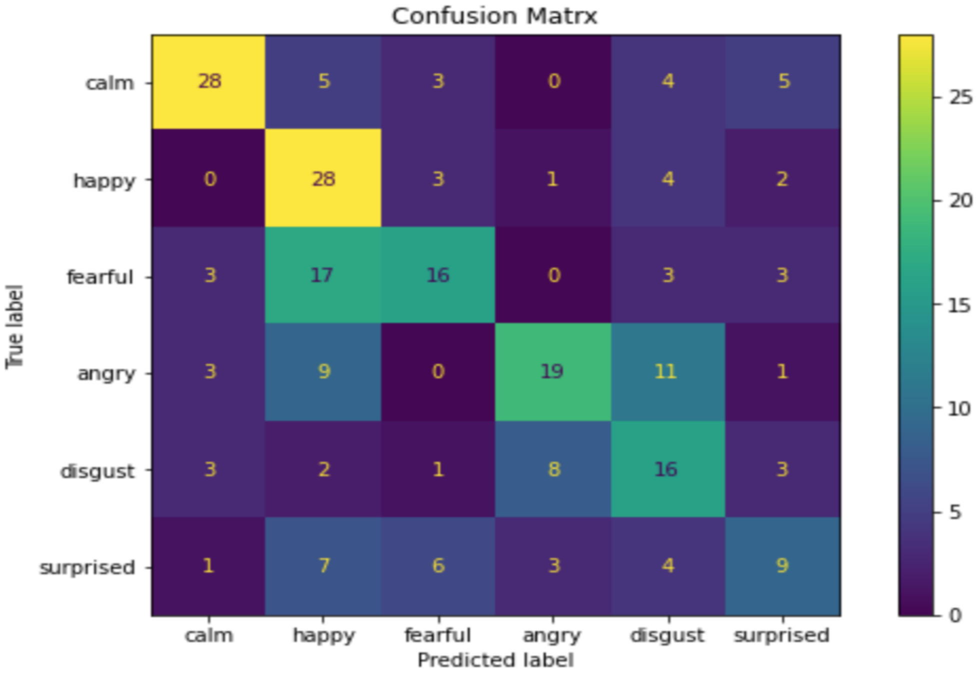


Fig 9 LR, Experiment II

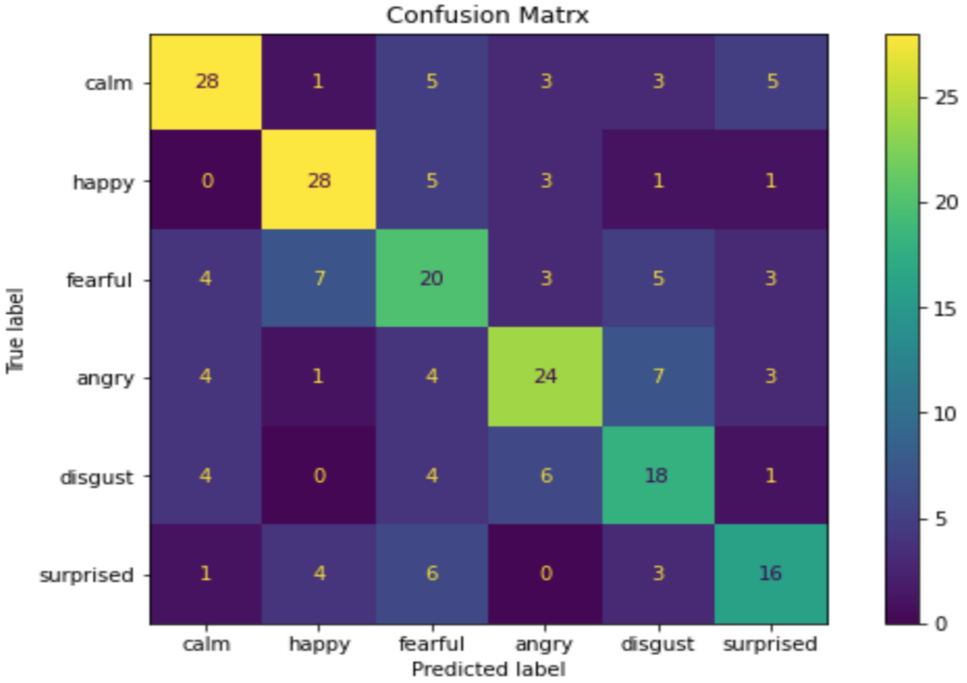


Fig 10 LR, Experiment III

- *SVM*

Table 3 Experiments Conducted with Respective Evaluations:

SN	Experiments	Accuracy	F1 Score	Precision	Recall
I	kernel="linear", C=1	0.43	0.56	0.56	0.56
II	kernel="poly", C=1	0.33	0.26	0.26	0.32
III	kernel="linear", C=2	0.42	0.57	0.57	0.57

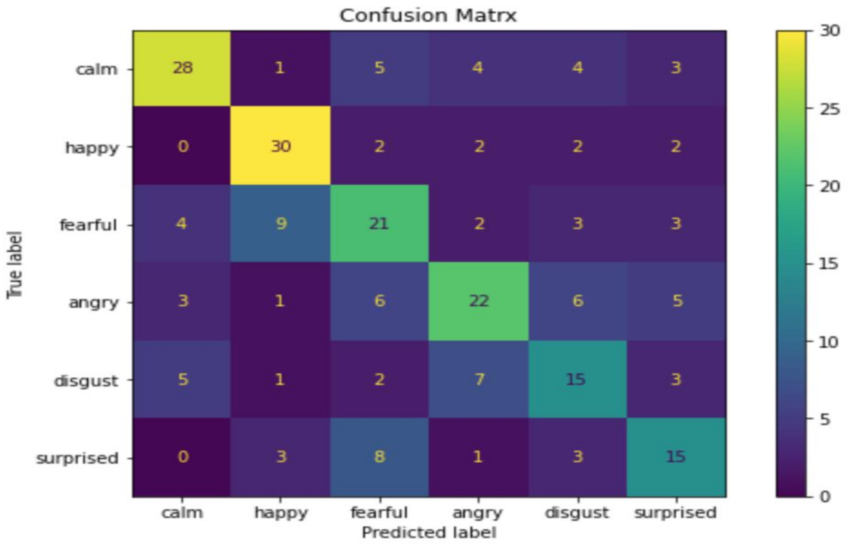


Fig 11 SVM, Experiment I

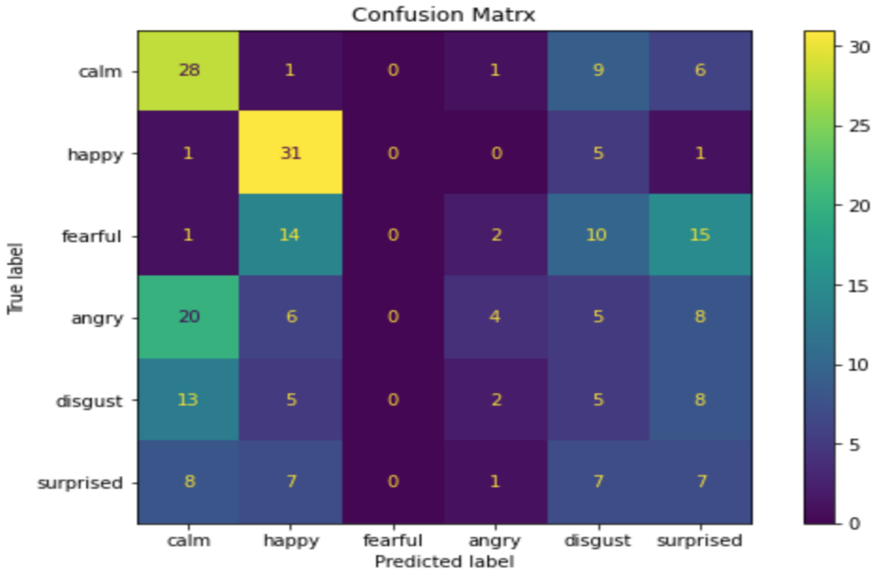


Fig 12 SVM, Experiment II

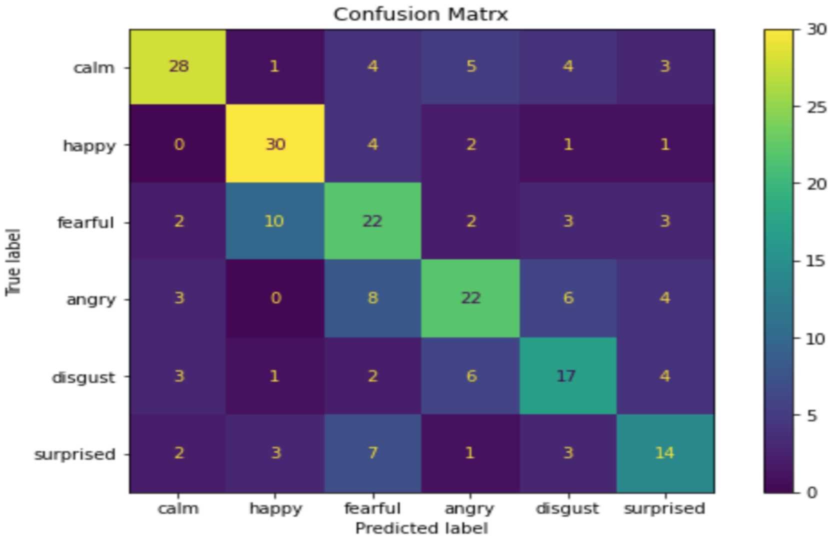


Fig 13 SVM, Experiment III

VI. DISCUSSION ON RESULTS

Evaluation of the experiments includes comparison of accuracy

- between the multiple experiment's of each algorithm and
- between different algorithms

The table below describes the comparison between different algorithms and the table consists the best result of the algorithm after performing multiple experiments:

Model	Best experiment condition	Accuracy	F1 Score	Precision	Recall
MLP	alpha=0.01, batch_size=256, epsilon=1e-08, hidden_layer_sizes=(600,), learning_rate='adaptive', max_iter=500	0.7	0.7	0.71	0.71
SVM	kernel="linear", C=1	0.43	0.56	0.56	0.56
LR	solver='newton-cg'	0.58	0.58	0.58	0.58

MLP has best accuracy, f1-score, precision and recall when it has a hidden layer with 600 hidden units. SVM has best results when it uses linear kernel and value 1 for regularization parameter, and LR has best results when it uses newton-cg as solver for optimization.

Compared to different models, we get best results with MLP Classifier and other two SVM and LR have similar results.

VII. CONTRIBUTIONS OF EACH GROUP MEMBER

SN	Name	Contribution
1	Samjhana Pokharel	Research on Speech Emotion Recognition Study of sound file and its features for speech emotion recognition Visualization of speech features Visualization of datasets Study of ML models for speech emotion recognition Multiple experimentation with Support Vector Machine and Logistic Regression Performance and accuracy measurement of SVM and LR Comparison of each experiments and models Drawing Conclusions
2	Ujwal Basnet	Research on Speech Emotion Recognition Study of sound file and its features for speech emotion recognition Visualization of speech features Visualization of datasets Study of ML models for speech emotion recognition Multiple experimentation with Multi-Layer Perceptron Classifiers Performance and accuracy measurement of MLP Comparison of each experiments and models Drawing Conclusions

VIII. CODE

Snippet codes for the implementation of different methods have already been specified and discussed above.

The complete code can be accessed via public github repository:

https://github.com/SamjhanaP/speechemotionrecognition

IX. CONCLUSION AND FUTURE EXTENSIONS TO THE PROJECT

The new era of automation has begun as a result of the increasing growth and development in the fields of AI and machine learning. The majority of these automated gadgets are controlled by the user's vocal commands. Many advantages can be created over present systems if, in addition to identifying words, the machines can interpret the speaker's emotion.

The processes for creating a voice emotion recognition system were covered in detail in this project, and several experiments were conducted to determine the influence of each step. Three different learning models were used: MLP, LR and SVM. Firstly, speech features like mfcc, chroma, and mel were extracted from the audio files. Then each model is trained in multiple experiments with variation in the parameters. And using the test dataset, accuracy of each model and each experiment were studied. And, at the end we conclude that MLP Classifier performs better when hidden units in a hidden layer are increased.

So, we conclude the following are the advantages of using MLP classifier in Speech Emotion Recognition:

- Allows you to work with nonlinear values with ease.
- Higher performance compared to other models
- Missing values can be handled,
- Complicated relationships can be modelled, and
- Many inputs can be supported.

For future enhancements, the proposed project can be further modeled in terms of efficiency, accuracy, and usability. The model may be extended to recognize more emotional states and sensations like sarcasm. And, a number of interactive systems can be developed using trained models in the underlying system to provide a system where users can interact with the machine or more like command using voice. Also, the communication can be made bi-directional instead of directional.

REFERENCES

[1]. Brownlee, J. (2016, April 1). *Logistic Regression for Machine Learning*. Machine Learning Mastery. https://machinelearningmastery.com/logistic-regression-for-machine-learning/

[2]. B.V., E. (2020). Speech Emotion Recognition with deep learning. *Procedia Computer Science*, *176*(2020), 251-260. https://www.sciencedirect.com/science/article/pii/S1877050920318512

[3]. Gandhi, R. (2018, June 7). *Support Vector Machine — Introduction to Machine Learning Algorithms*. Towards Data Science. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[4]. https://www.audeering.com/. (2020). *Audeering*. Audeering. https://www.audeering.com/

[5]. Nair, A. (2019, June 20). *A Beginner's Guide To Scikit-Learn's MLPClassifier*. Analytics India Magazine. https://analyticsindiamag.com/a-beginners-guide-to-scikit-learns-mlpclassifier/

[6]. Peerzade, G. N., & Deshmukh, R. R. (2018, March). A Review: Speech Emotion Recognition. *International Journal of Computer Sciences and Engineering*, *6*(3), 2347-2693. https://www.researchgate.net/publication/325774548_A_Review_Speech_Emotion_Recognition

[7]. scikit-learn developers. (n.d.). *Support Vector Machines*. Scikit Learn. https://scikit-learn.org/stable/modules/svm.html

[8]. SMART Lab. (n.d.). *RAVDESS*. Smart Laboratory. https://smartlaboratory.org/ravdess/

[9]. Stojiljković, M. (n.d.). *Logistic Regression in Python*. Real Python. https://realpython.com/logistic-regression-python/

[10]. Wikipedia. (2021). *Multilayer perceptron*. Wikipedia. https://en.wikipedia.org/wiki/Multilayer_perceptron

[11]. Wikipedia. (2021, March 9). *Chroma feature*. Wikipedia. https://en.wikipedia.org/wiki/Chroma_feature

[12]. Wikipedia. (2021, May 7). *Mel-frequency cepstrum*. Wikipedia. https://en.wikipedia.org/wiki/Mel-frequency_cepstrum#:~:text=From%20Wikipedia%2C%20the%20free%20encyclopedia,nonlinear%20mel%20scale%20of%20frequency.

[13]. Wikipedia. (2021, May 19). *Support-vector machine*. Wikipedia. https://en.wikipedia.org/wiki/Support-vector_machine

[14]. Wikipedia. (2021, May 22). *Logistic regression*. Wikipedia. https://en.wikipedia.org/wiki/Logistic_regression