SPEECH EMOTION RECOGNITION

A project report submitted in partial fulfillment of the requirements for the award of the degree of

Master of Computer Applications
in
Computer Applications

By
ANKIT SINGH
(205120018)

DEPARTMENT OF COMPUTER APPLICATIONS

NATIONAL INSTITUTE OF TECHNOLOGY,
TIRUCHIRAPPALLI 620015

JUNE 2023

BONAFIDE CERTIFICATE

This is to certify that the project **“SPEECH EMOTION RECOGNITION”** is a project work successfully done by

ANKIT SINGH (205120018)

in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications from the National Institute of Technology, Tiruchirappalli, during the academic year 2022-2023 (6th Semester – CA750 Project Work).

Dr. G.R. Gangadharan	Dr. Michael Arock
Project Guide	Head of the Department

Project viva-voce held on …………………………….

Internal Examiner	**External Examiner**

ABSTRACT

Speech emotion recognition (SER) has been actively studied in recent decade. Mainly SER are based on classification approach that classifies an emotion among different classes. speech emotion is a challenging problem because it is unclear that what features are effective for classification. In this report, I have taken MESD (Mexican Emotional Speech database) as a dataset, of year 2022. It has total of 864 audio files of single-word emotional utterances with Mexican cultural shaping. It has total six emotions anger, disgust, fear, happiness, neutral, and sadness. First I tried classification with model support vector machine, which gives giving an accuracy of 57.39%. But after adding some extra augmented voices, like after noise injection, time shifting etc, accuracy is increased by around 10%. Later I used ensemble learning with 3 algorithms logistic regression, random forest, and support vector machine, and I got an accuracy of 0.66%. By using only random forest algorithm gave maximum around 72% accuracy. I have used a python package called librosa for audio analysis. Librosa helps to visualize the audio signals and also we can do feature extraction using different processing techniques.

Keywords: Emotion recognition, Machine Learning, Ensemble learning, Feature extraction.

ACKNOWLEDGMENTS

We express our deep sense of gratitude to Dr. G. Aghila, Director, National Institute of Technology, Tiruchirappalli for giving me an opportunity to do this project.

I wish to extend my sincere gratitude to Dr. Michael Arock, Professor and Head, Department of Computer Applications, National Institute of Technology, Tiruchirappalli for supporting and encouraging me to carry out this project work and duly evaluating my progress.

I express my heartfelt gratitude to my internal guide Dr. G. R. Gangadharan, Department of Computer Applications, National Institute of Technology, Tiruchirappalli for her immense support and guidance during the study.

I also thank the review panel for helping me identify the shortages of the project and helping me to overcome those shortages by guiding me.

Finally, I would like to express my regards to all the faculty members of the Department of Computer Applications and others who have helped me develop this project directly or indirectly.

Ankit Singh

205120018

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Emotion Plays a vital role in our daily life. This helps us to make an intelligent decision. It helps us to understand others feeling and make others comfortable by giving feedback or suggestion based on their emotion. Research has revealed that emotion plays powerful role in human social interaction. Emotions tell a considerable amount of information about others mental state. This has opened a new research field called speech emotion recognition. This main goal is to understand about others mental state and emotion with their speech. Now a days human and computer interaction are become very important. Despite the great progress made in artificial intelligence, we are still far from being able to naturally interact with machines, partly because machines do not understand our emotion states. A good HCI (human - computer interaction) system required to know the user emotion to for enriching their experience. In human–machine interaction, the machine can be made to produce more appropriate responses if the state of emotion of the person can be accurately identified. Most state-of-the-art automatic speech recognition systems resort to natural language understanding to improve the accuracy of recognition of the spoken words. Such language understanding can be further improved if an emotional state of the speaker can be extracted, and this in turn will enhance the accuracy of the system. Several ways have been explored to recognize emotion such as facial expression, physiological signals etc. Compared to many biological signals (e.g., electrogram), speech signals can be acquired more easily and economically. There are many applications of detecting the emotion of the persons like in audio surveillance, interface with robots, commercial applications, clinical studies, video games, automatic call centers, web based E-learning, voice assistance like google assistance or Siri.

The output of an automatic emotion recognizer will naturally consist of labels of emotion.

The choice of a suitable set of labels is important. The Mexican Emotional Speech Database (MESD) provides single-word utterances for anger, disgust, fear, happiness, neutral, and sadness affective prosodies with Mexican cultural shaping. The MESD has been uttered by both adult and child non-professional actors: 3 female, 2 male, and 6 child voices are available. The audio recordings took place in a professional studio with the following materials: (1) a Sennheiser e835 microphone with a flat frequency response (100 Hz to 10 kHz), (2) a Focusrite Scarlett 2i4 audio interface connected to the microphone with an XLR cable and to the computer, and (3) the digital audio workstation REAPER (Rapid Environment for Audio Production, Engineering, and Recording). Audio files were stored as a sequence of 24-bit with a sample rate of 48000Hz. The amplitude of acoustic waveforms was rescaled between -1 and 1. 24 utterances per emotion are available for each type of voice, corpus, and level of naturalness.

Feature Extraction is an important part of SER sysytem. Choice of feature extraction will have great impact on accuracy of model. Feature analysis in emotion recognition is much less studied than that in speech recognition. In this project I have worked on MFCC, Chromagram, and Mel spectrogram features. I have tried all single, all together and other many combinations with these features and recorded the accuracy. In the experiment section I have explained all these.

CHAPTER 2

Literature Review

A paper Automatic speech emotion recognition using modulation spectral features was published[1]. Author of this paper is Siqing Wu, Department of Electrical and Computer Engineering, Queen's University, Kingston, ON, Canada K7L 3N6. In this study, modulation spectral features (MSFs) are proposed for the automatic recognition of human affective information from speech. The features are extracted from an auditory-inspired long-term spectro-temporal representation. Obtained using an auditory filterbank and a modulation filterbank for speech analysis, the representation captures both acoustic frequency and temporal modulation frequency components, thereby conveying information that is important for human speech perception but missing from conventional short-term spectral features. On an experiment assessing classification of discrete emotion categories, the MSFs show promising performance in comparison with features that are based on mel-frequency cepstral coefficients and perceptual linear prediction coefficients, two commonly used short-term spectral representations. The MSFs further render a substantial improvement in recognition performance when used to augment prosodic features, which have been extensively used for emotion recognition. Using both types of features, an overall recognition rate of 91.6% is obtained for classifying seven emotion categories. Moreover, in an experiment assessing recognition of continuous emotions, the proposed features in combination with prosodic features attain estimation performance comparable to human evaluation.

In the second paper which is written by Leila Kerkeni, Youssef Serrestou, Mohamed Mbarki, Kosai Raoof, Mohamed Ali Mahjoub and Catherine Cleder. Their SER system based on different classifiers and different methods for features extraction, is developed. Mel-frequency cepstrum coefficients (MFCC) and modulation spectral (MS) features are extracted from the speech signals and used to train different classifiers. Feature selection (FS) was applied in order to seek for the most relevant feature subset. Several machine learning paradigms were used for the emotion classification task. A recurrent neural network (RNN) classifier is used first to classify seven emotions. Their performances are compared later to multivariate linear regression (MLR) and sup- port vector machines (SVM) techniques, which are widely used in the field of emotion recognition for spoken audio signals. Berlin and Spanish databases are used as the experimental data set. This study shows that for Berlin database all classifiers achieve an accuracy of 83% when a speaker normalization (SN) and a feature selection are applied to the features. For Spanish database, the best accuracy (94 %) is achieved by RNN classifier without SN and with FS.

Another paper which is written by Kun Han1, DongYu2, Ivan Tashev2 in 2014[2]. In this paper they propose to utilize deep neural networks (DNNs) to extract high level features from raw data and show that they are effective forspeech emotion recognition. We first produce an emotion state probability distribution for each speech segment using DNNs. We then construct utterance-level features from segment-level probability distributions. These utterance level features are then fed into an extreme learning machine (ELM), a special simple and efficient single-hidden-layer neural network, to identify utterance-level emotions. The experimental results demon- strate that the proposed approach effectively learns emotional information from low-level features and leads to 20% relative accuracy improvement compared to the state of-the-art approaches[3]. Fourth paper published in 2014.

The fourth paper has provided a detailed review of the deep learning techniques for SER. Deep learning techniques such as DBM, RNN, DBN, CNN, and AE have been the subject of much research in recent years. These deep learning methods and their layer-wise architectures are briefly elaborated based on the classification of various natural emotion such as happiness, joy, sadness, neutral, sur- prise, boredom, disgust, fear, and anger. These methods offer easy model training as well as the efficiency of shared weights. Limitations of deep learning techniques include their large layer-wise internal architecture, less efficiency for temporally varying input data and over-learning during memorization of layer-wise information. This research work formsa base to evaluate the performance and limitations of current deep learning techniques. Further, it highlights some promising directions for better SER systems.

Fifth paper is doing prediction with hidden Markov model[4]. The proposed method makes use of shorttime log frequency power coefficients (LFPC) to represent the speech signals and a discrete hidden Markov model (HMM) as the classifier. Performance of the LFPC feature parameters is compared with that of the linear prediction Cepstral coefficients (LPCC) and mel-frequency Cepstral coefficients (MFCC) feature parameters commonly used in speech recognition systems. Results show that the proposed system yields an average accuracy of 78% and the best accuracy of 96% in the classification of six emotions.

Table 2.1: Literature summary

SR No	Reference	Technique	Preprocessing	Dataset	Accuracy	Remarks
1	S. Wu, T. H. Falk, and W.-Y[1]	Support Vector Machine using modulation spectral features	✓	Berlin emotional speech database, Vera am Mittag (VAM) database	91.6%	The MSFs give superior performance except in the case when speaker normalization is applied.

2	L. Kerkeni, Y. Serrestou, M. Mbarki, K. Raoof, M. A. Mahjoub, and C. Cleder[5]	Recurrent neural network(RNN), Multivariate linear regression (MLR), Support vector machines (SVM)	✓	Berlin and Spanish databases	94%	Highest accuracy is achieved by RNN classifier
3	K. Han, D. Yu, and I. Tashev[2]	Deep Neural Networks	✓	Interactive Emotional Dyadic Motion Capture(IEMOCAP) database	54.3%	outperform SVM by around 5%

4	R. A. Khalil, E. Jones, M. I. Babar, T. Jan, M. H. Zafar, and T. Alhussain[3]	Deep Learin-ing such as DBM, RNN, DBN, CNN, and AE	✓	IEMOCAP, Emo-DB and SAVEE datasets	92.3%	combining MFCCs, PLPs, FBANKs these three features showed better performance
5	T. L. Nwe, S. W. Foo, and L. C. De Silva [4]	Hidden Markov Model	✓	specially designed database consisting of 6 emotions	96%	Average accuracy is 78%

CHAPTER 3

Methodology

3.1 Work Flow

In my SER system first step is voice sample collection. Then feature vector creation by extracting features. Then we give the feature vector to the machine learning classifier for learning. After that, we can classify voices among six different emotions. This process is described in the figure.

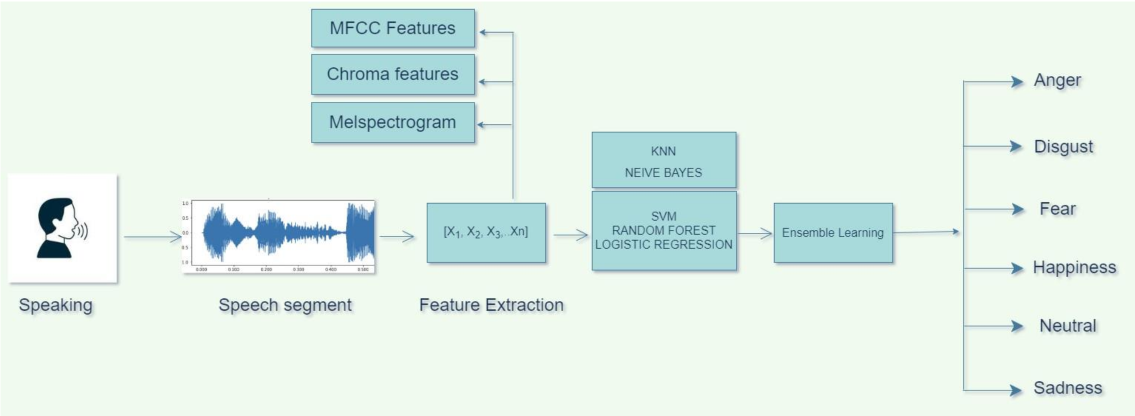


Figure 3.1: Work flow diagram

3.2 Data Collection

I have taken MESD (Mexican emotional speech dataset) to evaluate the performance of different methods[6]. The MESD provides single-word utterances for anger, disgust, fear, happiness, neutral, and sadness affective prosodies with Mexican cultural shaping. The MESD has been uttered by both adult and child non-professional actors: 3 female, 2 male, and 6 child voices are available. In the figure below shows the count of all emotions in the dataset.

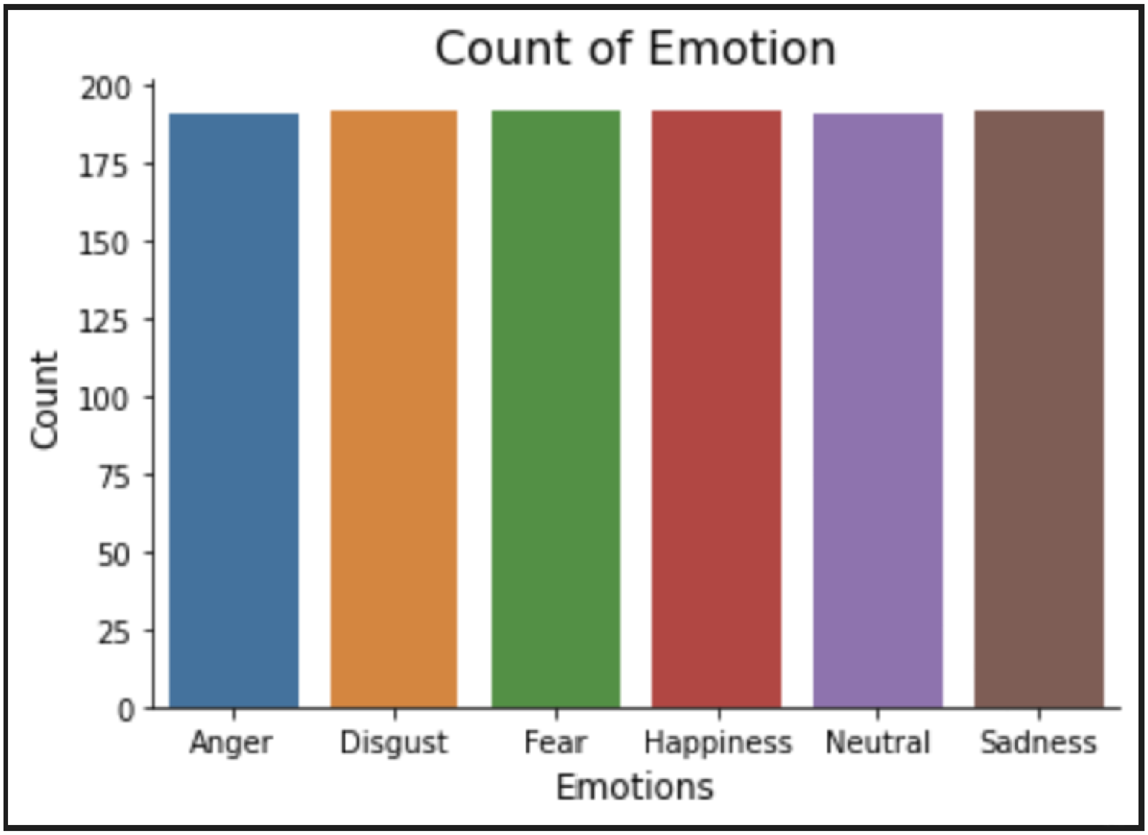


Figure 3.2: count of emotions

3.3 Data Visualisation and Exploration

I have taken one audio file of each type of emotion and plotted wave-plot and spectrogram for them. Below I have shown wave-plot and spectrogram of 3 emotions.

Fear Emotion:

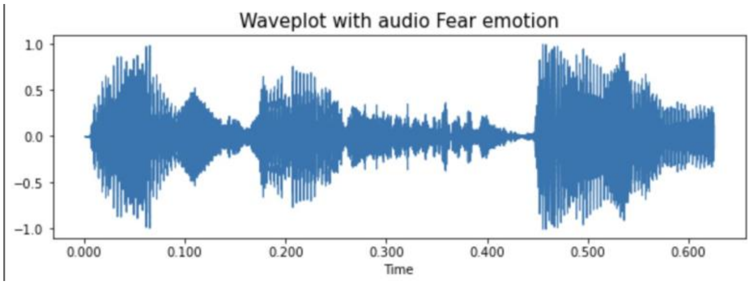


Figure 3.3: Wave plot for fear emotion

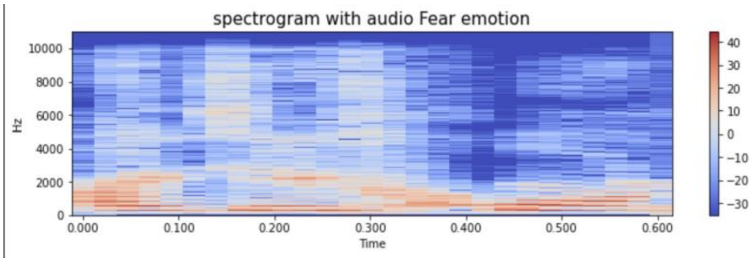


Figure 3.4: Spectrogram of fear emotion

Anger Emotion:

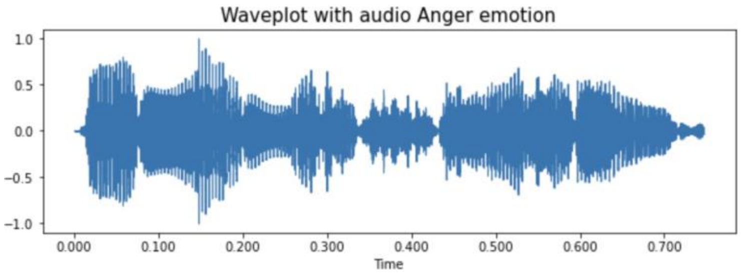


Figure 3.5: Wave plot for anger emotion

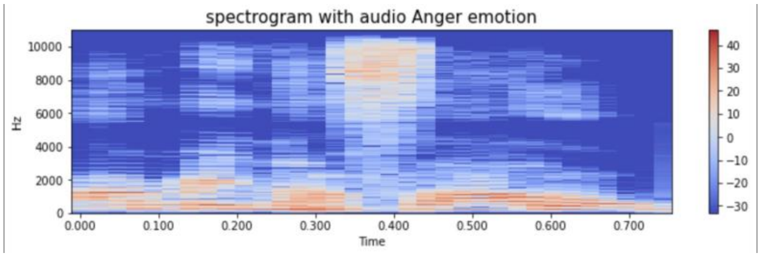


Figure 3.6: Spectrogram of anger emotion

Happy Emotion:

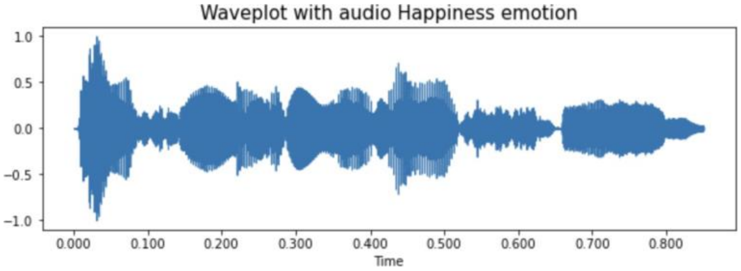


Figure 3.7: Wave plot for happy emotion

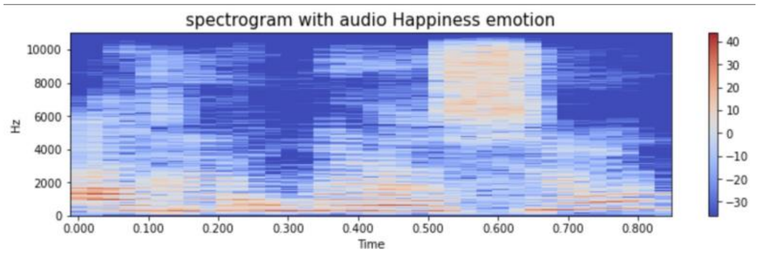


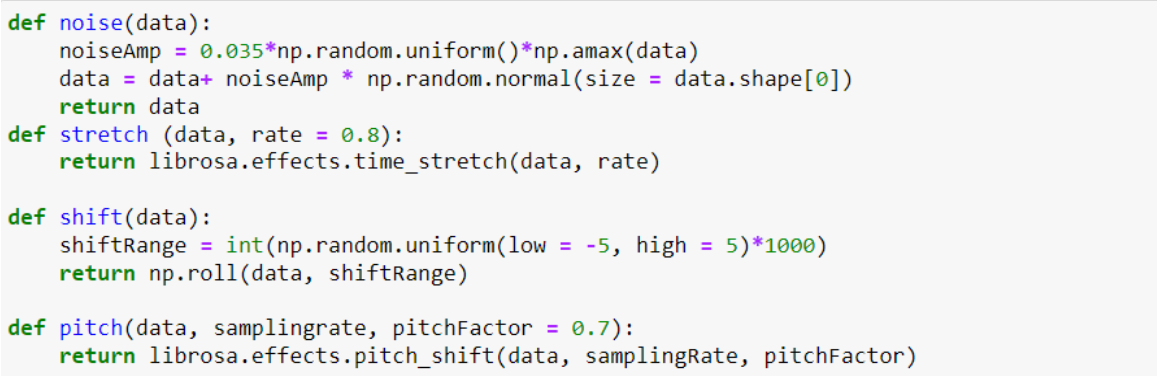
Figure 3.8: Spectrogram of happy emotion

3.4 Data Augmentation

- Data augmentation is a process in which we generate new data by doing some alteration on existing dataset.
- To generate new data for audio, we add small perturbations for that we can apply noise injection, shifting time, changing pitch and speed of audio.

- The aim is to make our model invariant to those perturbations and enhance its ability to generalize.

Here is the image shows the functions to apply data augmentation to audio file.



```
def noise(data):
    noiseAmp = 0.035*np.random.uniform()*np.amax(data)
    data = data+ noiseAmp * np.random.normal(size = data.shape[0])
    return data
def stretch (data, rate = 0.8):
    return librosa.effects.time_stretch(data, rate)

def shift(data):
    shiftRange = int(np.random.uniform(low = -5, high = 5)*1000)
    return np.roll(data, shiftRange)

def pitch(data, samplingrate, pitchFactor = 0.7):
    return librosa.effects.pitch_shift(data, samplingRate, pitchFactor)
```

Figure 3.9: Functions for data augmentation

Here I am taking an audio file from dataset and plotted waveplot of that audio. After applying data augmentation, we again plot waveplot. So that we can clearly see the changes.

Waveplot for Sample Audio

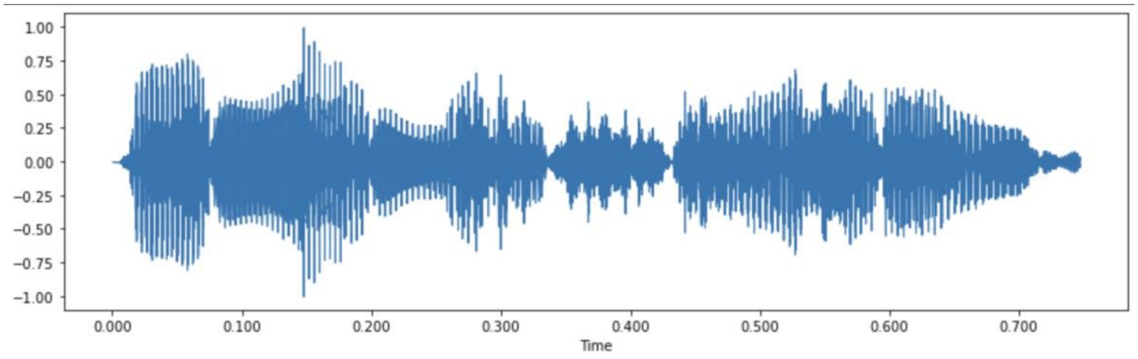


Figure 3.10: Sample audio waveplot

After Noise injection

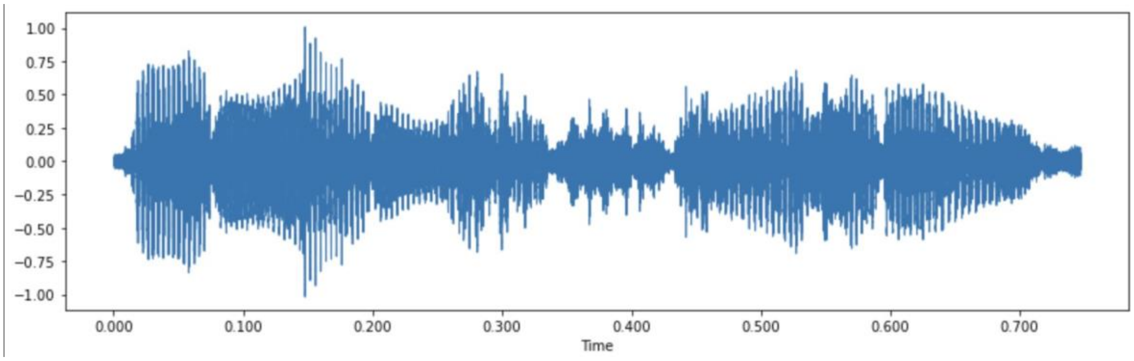


Figure 3.11: Audio after noise injection

Stretching

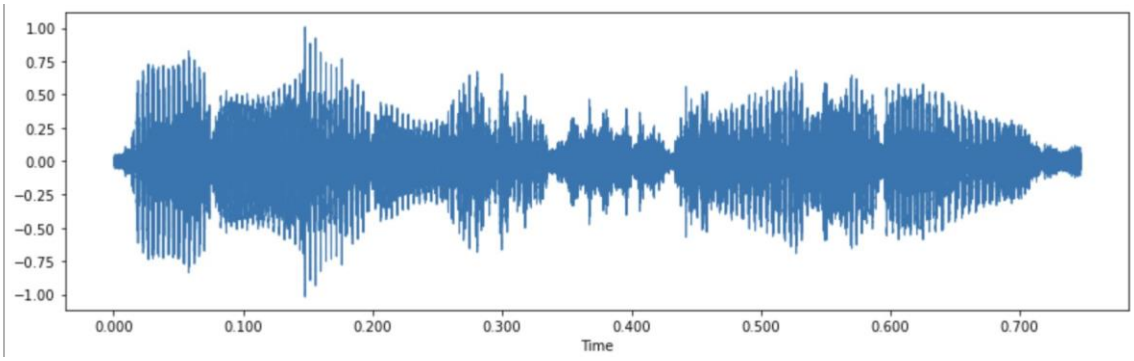


Figure 3.12: Audio after Stretching

Shifting

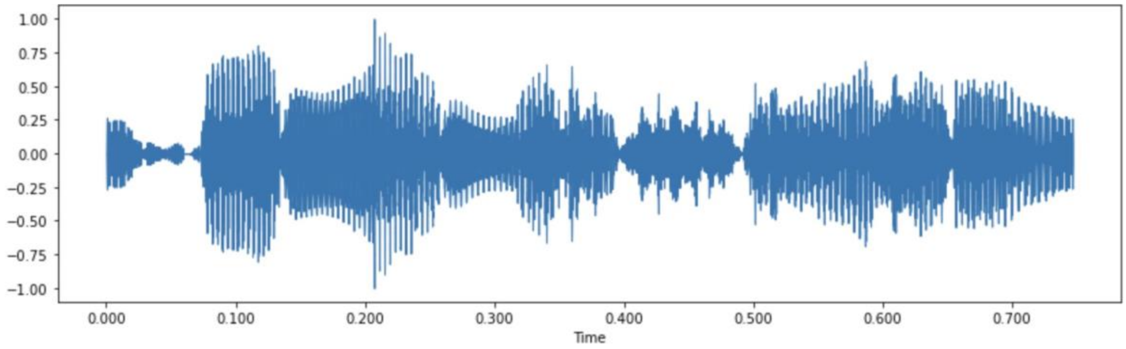


Figure 3.13: Audio after Shifting

Pitching

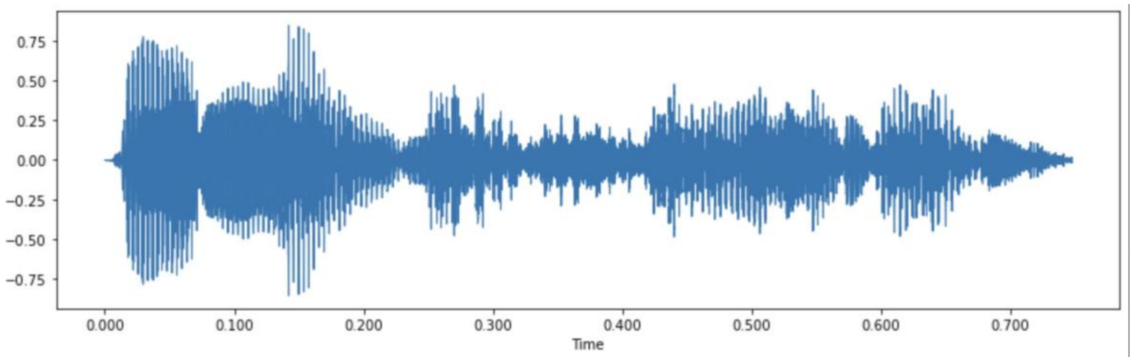


Figure 3.14: Audio after Pitching

3.5 Feature Extraction

It is the most important part of SER system. Because the accuracy of the model totally depends on selection of features. If we able to understand which features of audio is more relevant to know emotion then our model will perform well. As we know that data provided by audio cannot be directly understood by our model, so we need to convert them into understandable format for which feature extraction is used. After applying feature extraction, we get an array as result which is used as feature variable for model. HereI am extracting 3 features MFCC, Chroma Features, and Melspectrogram and appended all three features in one vector. My final feature variable size is 180. I have also tried different combinations of all features, but we get highest accuracy when all three are used together.

3.5.1 Mel frequency cepstrum coefficient (MFCC)

- The word Mel represents the scale used in Frequency vs Pitch measurement[7]. The value measured in frequency scale can be converted into Mel scale using the formula

$$m = 1125 ln(1 + (f/700))$$

- The word Cepstrum represents the Fourier Transform of the log spectrum of the speech signal.

Steps to calculate MFCC :

- Frame the signal into short frames.
- For each frame calculate the power spectrum.
- Apply the mel filterbank to the power spectra, sum the energy in each filter.
- Take the logarithm of all filterbank energies.
- Take the logarithm of all filterbank energies.
- Take the Discrete Cosine Transform (DCT) of the log filterbank energies.
- The lower-level coefficients of each frame represent steady changes in the pitch and energy values, and therefore they are better for analysis. These lower-level coefficients are called the Mel Frequency cepstral coefficients.

I have extracted 40 mfcc features from one audio file.

3.5.2 Chroma Features

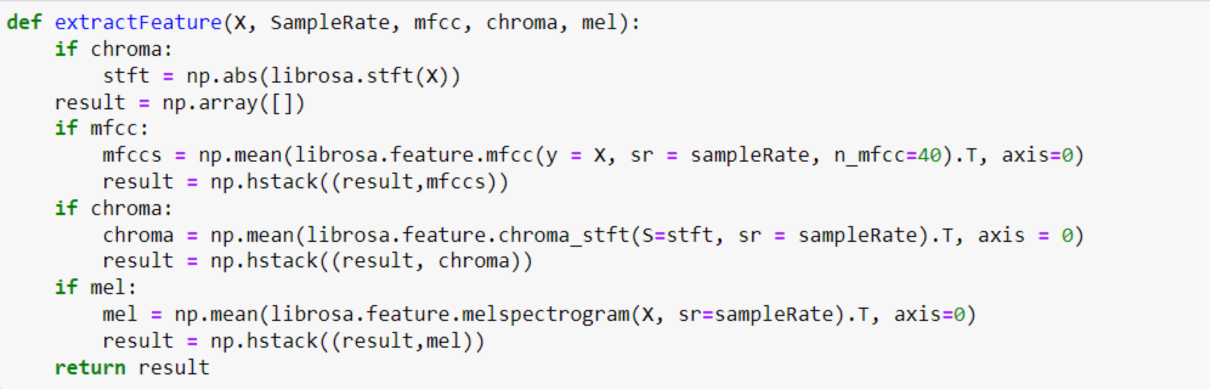
It is also known as chromagram[8]. An audio file contains 12 different pitch classes. These pitch class profiles are helpful for analyzing audio files. The term chromagram represents all pitches of an audio file. Pitches are the property of any sound or signal based onfrequency related scale. So, I have extracted 12 features of chroma.

3.5.3 Melspectrogram

Fourier Transform – It converts the signal from the time domain into the frequency domain. The fourier transform is a mathematical formula that allows us to decompose a signal into its individual frequency's amplitude.

The Spectrogram – When signal's frequency content varies over time these signals are known as non-periodic signals. We need a way to represent the spectrum of these signals because they vary over time. We calculate fast fourier transform on several windowed segment of signal. This is called short time fourier transform. The result we get is called spectrogram.

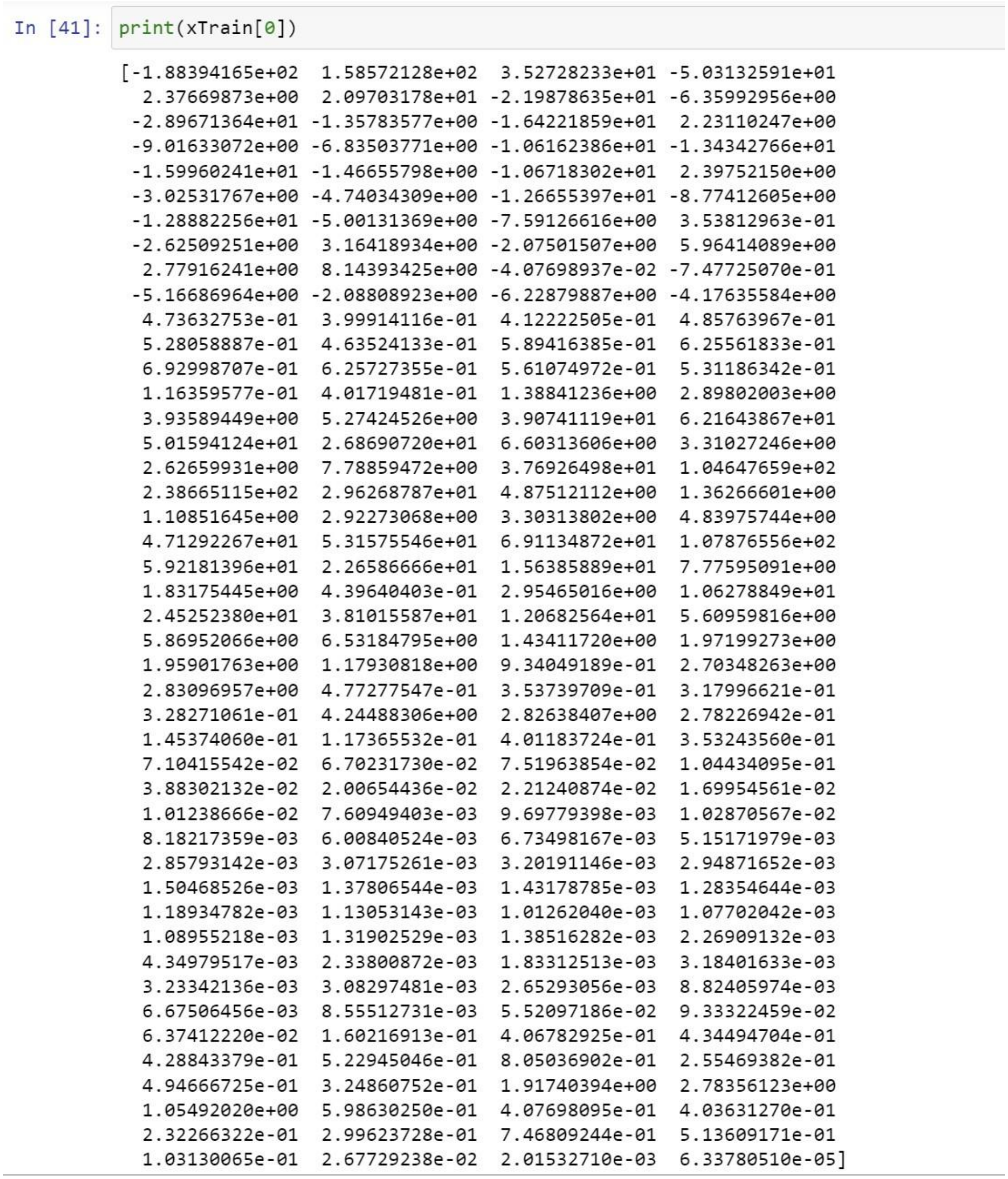
So, a Mel spectrogram is a spectrogram where the frequencies are converted to the mel scale using the fourier transform[9]. From here we got 128 features.



```
def extractFeature(X, SampleRate, mfcc, chroma, mel):
    if chroma:
        stft = np.abs(librosa.stft(X))
    result = np.array([])
    if mfcc:
        mfccs = np.mean(librosa.feature.mfcc(y = X, sr = sampleRate, n_mfcc=40).T, axis=0)
        result = np.hstack((result,mfccs))
    if chroma:
        chroma = np.mean(librosa.feature.chroma_stft(S=stft, sr = sampleRate).T, axis = 0)
        result = np.hstack((result, chroma))
    if mel:
        mel = np.mean(librosa.feature.melspectrogram(X, sr=sampleRate).T, axis=0)
        result = np.hstack((result,mel))
    return result
```

Figure 3.15: Function for feature Extraction

After extracting all three features we get a vector of 180 length as a result. Which will further use for model training.



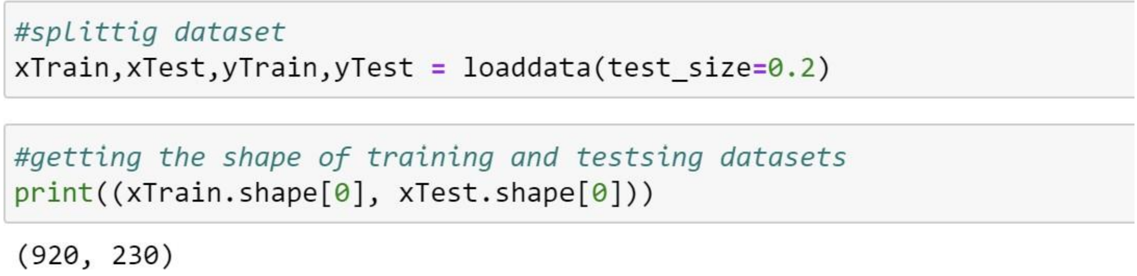
```
In [41]: print(xTrain[0])

         [-1.88394165e+02  1.58572128e+02  3.52728233e+01 -5.03132591e+01
           2.37669873e+00  2.09703178e+01 -2.19878635e+01 -6.35992956e+00
          -2.89671364e+01 -1.35783577e+00 -1.64221859e+01  2.23110247e+00
          -9.01633072e+00 -6.83503771e+00 -1.06162386e+01 -1.34342766e+01
          -1.59960241e+01 -1.46655798e+00 -1.06718302e+01  2.39752150e+00
          -3.02531767e+00 -4.74034309e+00 -1.26655397e+01 -8.77412605e+00
          -1.28882256e+01 -5.00131369e+00 -7.59126616e+00  3.53812963e-01
          -2.62509251e+00  3.16418934e+00 -2.07501507e+00  5.96414089e+00
           2.77916241e+00  8.14393425e+00 -4.07698937e-02 -7.47725070e-01
          -5.16686964e+00 -2.08808923e+00 -6.22879887e+00 -4.17635584e+00
           4.73632753e-01  3.99914116e-01  4.12222505e-01  4.85763967e-01
           5.28058887e-01  4.63524133e-01  5.89416385e-01  6.25561833e-01
           6.92998707e-01  6.25727355e-01  5.61074972e-01  5.31186342e-01
           1.16359577e-01  4.01719481e-01  1.38841236e+00  2.89802003e+00
           3.93589449e+00  5.27424526e+00  3.90741119e+01  6.21643867e+01
           5.01594124e+01  2.68690720e+01  6.60313606e+00  3.31027246e+00
           2.62659931e+00  7.78859472e+00  3.76926498e+01  1.04647659e+02
           2.38665115e+02  2.96268787e+01  4.87512112e+00  1.36266601e+00
           1.10851645e+00  2.92273068e+00  3.30313802e+00  4.83975744e+00
           4.71292267e+01  5.31575546e+01  6.91134872e+01  1.07876556e+02
           5.92181396e+01  2.26586666e+01  1.56385889e+01  7.77595091e+00
           1.83175445e+00  4.39640403e-01  2.95465016e+00  1.06278849e+01
           2.45252380e+01  3.81015587e+01  1.20682564e+01  5.60959816e+00
           5.86952066e+00  6.53184795e+00  1.43411720e+00  1.97199273e+00
           1.95901763e+00  1.17930818e+00  9.34049189e-01  2.70348263e+00
           2.83096957e+00  4.77277547e-01  3.53739709e-01  3.17996621e-01
           3.28271061e-01  4.24488306e+00  2.82638407e+00  2.78226942e-01
           1.45374060e-01  1.17365532e-01  4.01183724e-01  3.53243560e-01
           7.10415542e-02  6.70231730e-02  7.51963854e-02  1.04434095e-01
           3.88302132e-02  2.00654436e-02  2.21240874e-02  1.69954561e-02
           1.01238666e-02  7.60949403e-03  9.69779398e-03  1.02870567e-02
           8.18217359e-03  6.00840524e-03  6.73498167e-03  5.15171979e-03
           2.85793142e-03  3.07175261e-03  3.20191146e-03  2.94871652e-03
           1.50468526e-03  1.37806544e-03  1.43178785e-03  1.28354644e-03
           1.18934782e-03  1.13053143e-03  1.01262040e-03  1.07702042e-03
           1.08955218e-03  1.31902529e-03  1.38516282e-03  2.26909132e-03
           4.34979517e-03  2.33800872e-03  1.83312513e-03  3.18401633e-03
           3.23342136e-03  3.08297481e-03  2.65293056e-03  8.82405974e-03
           6.67506456e-03  8.55512731e-03  5.52097186e-02  9.33322459e-02
           6.37412220e-02  1.60216913e-01  4.06782925e-01  4.34494704e-01
           4.28843379e-01  5.22945046e-01  8.05036902e-01  2.55469382e-01
           4.94666725e-01  3.24860752e-01  1.91740394e+00  2.78356123e+00
           1.05492020e+00  5.98630250e-01  4.07698095e-01  4.03631270e-01
           2.32266322e-01  2.99623728e-01  7.46809244e-01  5.13609171e-01
           1.03130065e-01  2.67729238e-02  2.01532710e-03  6.33780510e-05]
```

Figure 3.16: Function for feature Extraction

3.6 Data Preparation

After getting the feature vector, the next step is to split data into training and testing. I have split 80% of the data for the training dataset and 20% for testing the model. Finally, we got 920 audios for training and 230 for testing.



```
#splittig dataset
xTrain,xTest,yTrain,yTest = loaddata(test_size=0.2)
```

```
#getting the shape of training and testsing datasets
print((xTrain.shape[0], xTest.shape[0]))
```

```
(920, 230)
```

Figure 3.17: Test Train splitting of dataset

3.7 Algorithms

I have used 5 machine-learning classification algorithms. Which is Logistic Regression, Naive Bayes, K - Nearest Neighbor, Random Forest, and Support Vector Machine. I have compared the accuracy of all algorithms. Then I have voting classifier in ensemble learning technique using the best three algorithms.

3.7.1 Logistic Regression

Logistic regression is one of the most popular Machine Learning algorithms[10], which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. There are three types of Logistic Regression algorithms. These are Binary class, Multi-class and Ordinal class logistic algorithms depending on the type of target class. The Wikipedia definition states that "Logistic regression computes the relationship between the target (dependent) variable and one or more independent variables using the estimated probability values through a logistic function". The logistic function, also known as a sigmoid function, maps predicted values to probability values.

The procedure of a multiclass logistic regression algorithm is as follows:

1. For an N class problem, divide into N pairs of binary class problems.
2. For each binary class problem compute probability values of the observation belonging to a class.
3. Make the final prediction by computing the maximum probability value amongst all classes.

Logistic regression equation :

Equation of straight line

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + ... + b_nx_n$$

3.7.2 Naive Bayes

Naive Bayes classifier is based on Bayes theorem[11], which determines the probability of an event based on a prior probability of events. Bayes theorem is used to compute prior probability values. This classifier algorithm assumes feature independence. No correlation between the features is considered. The algorithm is said to be Naive because it treats all the features to independently contribute to deciding the target class.

The steps of a simple Naïve Bayes algorithm is as follows:

1. Create a frequency table for all features individually. Tag the frequency of each entry against the target class.
2. Create a likelihood table by computing probability values for each entry in the frequency table.

3. Calculate posterior probability for each target class using the Bayes theorem.

4. Declare the target class with the highest posterior probability value as the predicted outcome.

There are three types of Naive Bayes algorithm, namely: the Gaussian Naive Bayes (GNB) which is applicable with features following a normal distribution, the Multinomial Naive Bayes (MNB) which is most suited to use when the number of times the outcome occurs is to be computed, and the Bernoulli Naive Bayes (BNB) for a dataset with binary features.

Bayes theorem for calculating posterior probability: -

$$P\left(\frac{c}{x}\right) = P\,\frac{\left(\frac{x}{c}\right)P(c)}{P(x)}$$

3.7.3 K - Nearest Neighbor

K-Nearest Neighbor (KNN) is the simplest classification algorithm[12]. Here we plot all data points on space, and with any new sample, take k nearest points on space observe it and make a decision based on majority voting. Thus, KNN algorithm involves no training, and it takes the least calculation time when implemented with an optimal value of k.

The steps of KNN algorithm is as follows :

1. For a given instance, find its distance from all other data points. Use an appropriate distance metric based on the problem instance.

2. Sort the computed distances in increasing order. Depending on the value of k, observe the nearest k points.

3. Identify the majority class amongst the k points and declare it as the predicted class.

Choosing an optimal value of k is a challenge in this approach. Most often, the process is repeated for a number of different trials of k. The evaluation scores are then observed using a graph to find the optimal value of k.

3.7.4 Random Forest

Random forest is a classification algorithm[13]. It is expansion of decision trees. While the root node and splitting features in the decision tree are based on the Gini and Information gain values, the random forest algorithm does it in a random fashion. The random forest is a collection of decision trees. Therefore, a large number of trees gives better results. Overfitting is a potential drawback of random forests, but increasing the number of trees can reduce overfitting. Random forest also has several advantages like its capability to handle missing values and classify multi-class categorical variables.

The steps of building a random forest classifier are as follows :

1. Select a subset of features from the dataset.
2. From the selected subset of features, using the best split method, pick a node.
3. Continue the best split method to form child nodes from the subset of features.
4. Repeat the steps until all nodes are used as split.
5. Iteratively create n number of trees using steps 1 -4 to form a forest.

3.7.5 Support Vector Machine

Support Vector Machines (SVM) is a supervised algorithm[14]. It can be used for both classification and regression problems. Support vectors are coordinate points in space, formed using the attributes of a data point. Briefly, we plot each data item as a point in n-dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate. Then classification between the classes is performed by finding a hyperplane in space that clearly separates the distinct classes. SVM works best for high dimensional data. The important aspect of implementing SVM algorithm is finding the hyperplane. Two conditions are to be met in the order given while choosing the right hyperplane.

1. The hyperplane should classify the classes most accurately.
2. The margin distance from the hyperplane to the nearest data point must be maximized.

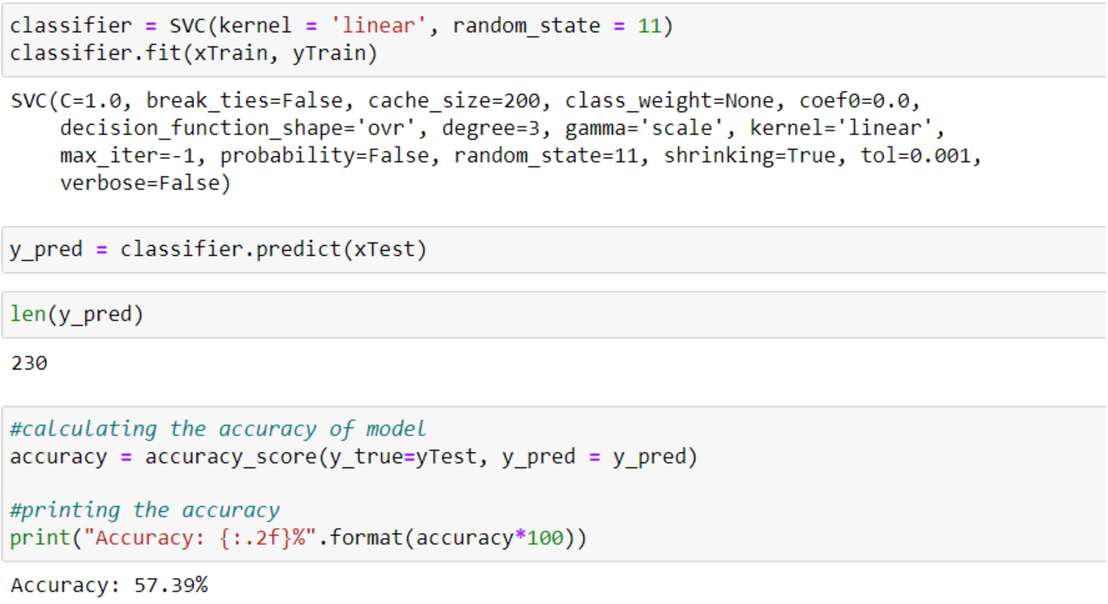
For a low dimensional dataset, the method of kernel trick in SVM introduces additional features to transform the dataset to a high dimensional space and thereby make identifying the hyperplane achievable. The linear solver based SVM is giving higher accuracy than rbf, But linear solver is taking more time.

CHAPTER 4

Experimental Results

4.1 Approach 1 - with support vector machine using Train Test split

I have trained support vector machine with 80% of the data. Support vector machine is giving more accuracy with linear type kernel. Currently it is giving 57.39% accuracy.



```
classifier = SVC(kernel = 'linear', random_state = 11)
classifier.fit(xTrain, yTrain)
```

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear',
    max_iter=-1, probability=False, random_state=11, shrinking=True, tol=0.001,
    verbose=False)
```

```
y_pred = classifier.predict(xTest)
```

```
len(y_pred)
```

```
230
```

```
#calculating the accuracy of model
accuracy = accuracy_score(y_true=yTest, y_pred = y_pred)

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

```
Accuracy: 57.39%
```

Figure 4.1: Using Support Vector Machine

4.2 Approach 2 - Using K-fold method

Using K-fold cross-validation method, the dataset is split into training and validation sets. The model is trained using the training data. The trained model is evaluated using

Table 4.1: Accuracy with different features

Features	Accuracy
MFCC	53%
Croma	35%
MEL	38%
MFCC, Chroma	57.39%
MFCC,MEL	57.39%
MFCC, Chroma, MEL	57.39%

the validation set and accuracy score is computed. I have tested with 5 algorithms below is the accuracy of each algorithms.

4.3 With data augmentation

I have applied data augmentation like noise injection, pitching, time shift, and stretch. So in dataset one is original audio and others 4 copies with applying different alterations is added. After doing this I have checked accuracy of all algorithms.

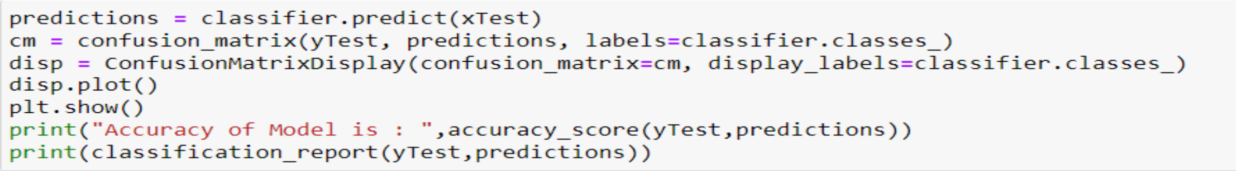
Here I am comparing accuracy of all models with and without data augmentation.

Table 4.2: Accuracy Of Different Algorithms

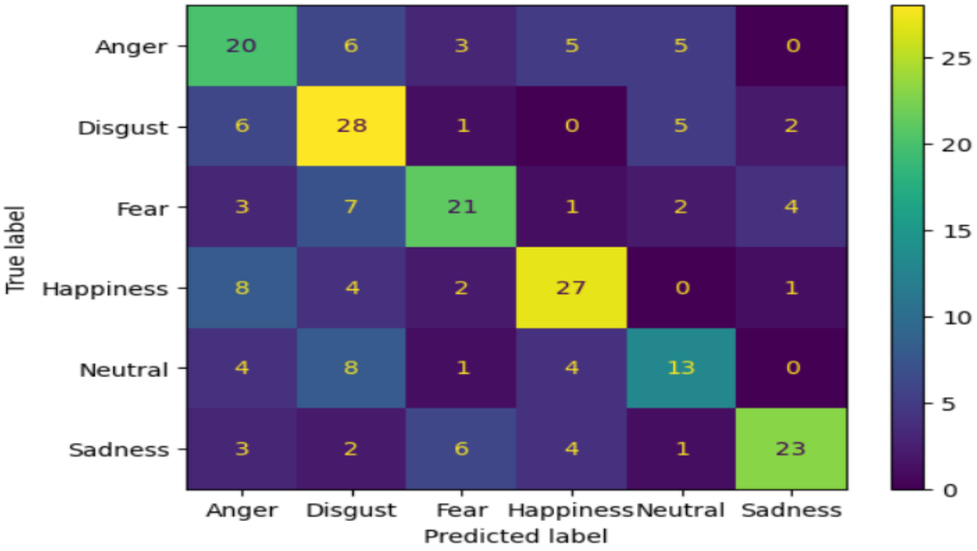
Algorithms	Accuracy without data augmentation	Accuracy with data augmentation
Logistic Regression	57%	55%
Random Forest	70%	70%
K - Nearest Neighbors	49%	51%
Support Vector Machine	50%	58%
Naive Bayes	44%	43%

So here KNN and Naive Bayes has minimum accuracy among all five. So I have applied ensemble voting classifier on three algorithms which are logistic regression, Random Forest and support Vector Machine.

Confusion matrix:



```
predictions = classifier.predict(xTest)
cm = confusion_matrix(yTest, predictions, labels=classifier.classes_)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=classifier.classes_)
disp.plot()
plt.show()
print("Accuracy of Model is : ",accuracy_score(yTest,predictions))
print(classification_report(yTest,predictions))
```



	Anger	Disgust	Fear	Happiness	Neutral	Sadness
Anger	20	6	3	5	5	0
Disgust	6	28	1	0	5	2
Fear	3	7	21	1	2	4
Happiness	8	4	2	27	0	1
Neutral	4	8	1	4	13	0
Sadness	3	2	6	4	1	23

Ensemble Learning with voting classifier - It is a type of machine learning model in which we take multiple machine learning model train them and predicts an output based on their highest probability of chosen class as a output. It simply takes the output of each classifier and predicts output based on highest majority of voting.Voting classifier supports two types of voting.

Hard Voting - In hard voting, the final output is the class with majority votes.

Soft Voting - In this type of voting, the output class is predicted based on the average probability given to the class.

In my ensemble learning with hard voting, I got an accuracy of 63% with all algorithms and after removing 2 less accurate algorithms now the accuracy is increased to 66%.

In soft Voting method the I got accuracy of 65% in both ways with all algorithms as well as with three algorithms.

CHAPTER 5

Conclusions

Several observations and conclusions can be derived from the experimental results. There is an overall improvement in performance with different approaches. Support Vector Machine normally giving accuracy of 57.39%. But after data augmentation, dataset size is increased from 1150 to 5750. This data is split to 80% - 20% for training and testing. Then accuracy of support vector machine is increased. Finally, I tested this dataset with 5 classifier in which random forest has highest accuracy of 70%. I did ensemble leaning with top three performer among these five classifiers then the accuracy is 66% with hard voting type and 65% with soft voting type. So overall random forest gives the highest accuracy among all experiments.

Future Work -

In future work, this project can be further modeled in terms of accuracy. We can study more about types of features and combinations of features that will help increase accuracy. Additional to the emotions, the model can be extended to recognize feelings such as depression and mood changes. There is another domain of depression detection using speech emotion. This project can be further Such systems can be used by therapists to monitor the mood swings of the patients.

Bibliography

[1] S. Wu, T. H. Falk, and W.-Y. Chan, “Automatic speech emotion recognition using modulation spectral features,” *Speech Communication*, vol. 53, no. 5, pp. 768–785, 2011, perceptual and Statistical Audition. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167639310001470

[2] K. Han, D. Yu, and I. Tashev, “Speech emotion recognition using deep neural network and extreme learning machine,” in *Interspeech 2014*, September 2014. [Online]. Available: https://www.microsoft.com/en-us/research/publication/speech-emotion-recognition-using-deep-neural-network-and-extreme-learning-machine/

[3] R. A. Khalil, E. Jones, M. I. Babar, T. Jan, M. H. Zafar, and T. Alhussain, “Speech emotion recognition using deep learning techniques: A review,” *IEEE Access*, vol. 7, pp. 117 327–117 345, 2019.

[4] T. L. Nwe, S. W. Foo, and L. C. De Silva, “Speech emotion recognition using hidden markov models,” *Speech Communication*, vol. 41, no. 4, pp. 603–623, 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167639303000992

[5] L. Kerkeni, Y. Serrestou, M. Mbarki, K. Raoof, M. A. Mahjoub, and C. Cleder, “Automatic speech emotion recognition using machine learning,” in *Social Media and Machine Learning*, A. Cano, Ed. Rijeka: IntechOpen, 2019, ch. 2. [Online]. Available: https://doi.org/10.5772/intechopen.84856

[6] D. I. I.-Z. Mathilde Marie Duville, Luz María Alonso-Valerdi, “Mexican emotional speech database (mesd),” 2022. [Online]. Available: https://data.mendeley.com/datasets/cy34mh68j9/5

[7] P. Cryptography, “Practical cryptography,” 2022. [Online].

Available: http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/

[8] Analyticsindimag, “A tutorial on spectral feature extraction for audio,” 2022. [Online]. Available: https://analyticsindiamag.com/a-tutorial-on-spectral-feature-extraction-for-audio-analytics/

[9] L. Roberts, “Understanding the mel spectrogram,” 2022. [Online]. Available: https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53

[10] wikipedia, “Logistic regression,” 2022. [Online]. Available: https://en.wikipedia.org/wiki/Logistic_regression

[11] AnalyticsVidya, “Learn naive algorithm | naive bayes classifier examples,” 2022. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

[12] ——, “K nearest neighbor | knn algoritm,” 2022. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/

[13] Medium, “How random forest algorithm works in machine learning,” 2022. [Online]. Available: https://synced.medium.com/how-random-forest-algorithm-works-in-machine-learning-3c0fe15b6674

[14] AnalyticsVidya, “Support vector machine algorithm in machine algoritm,” 2022. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/