Speech Emotion Recognition in Customer Feedback

Anantha Krishnan Hari

Department of Physics and Computer Science

Wilfrid Laurier University

04/17/2020

CP680 Capstone Project

Supervisor: Dr. Abdul-Rahman Mawlood-Yunis

Abstract

Understanding customer's feedback has always a challenge to all types of business. Often customer-facing companies don't get enough information from the customer about their services and product. Also, in most cases, even if the companies get the information, it often doesn't happen to be the right information that they want from the customer. Unfortunately, this information is the most essential one for companies to help in making business decisions and improve their product and its service. To tackle this problem, this project uses Speech Emotion Recognition (SER) which helps companies to automatically detect customer satisfaction through their emotions without the need for customer feedback. This is achieved by implementing a neural network technique called Multi-layer Perceptron (MLP) which is based on a supervised learning algorithm. By using librosa [31] [32], soundfile [36] and sklearn libraries [30] [35] and MLPClassifier [34] in python, we build a model to predict the emotion of a speech. First, we load the data from the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) dataset [33] and then start to extract features such as Mel Frequency Cepstral Coefficient (MFCC), Chroma and Mel Spectrogram Frequency (MEL). The

next step is to split the dataset into training and testing set. Then, we will initialize an MLPClassifier, train the model and predict human emotions.

Table of Contents

List of Figures

List of Equations

List of Tables

1. Introduction

Customer contention is the key to success for any business. To retain customers, companies often go way beyond the customer's expectations. To survive in a competitive environment, successful companies realize that listening to customer's feedback is the most valuable information which they can ever get. So, it no secret that the customer's opinion shapes the business of the company. This information is collected in various ways such as customer reviews, opinion polls, customer feedback through online and calls. Over the years, customer service is the medium to connect your brand with the customers. They provide service for their product and by making the customer happy, they make business happy. Often during customer service or after helping the customer, they heavily rely on the feedback to improve the product and service. Unfortunately, companies are finding it hard to get customer feedback for every interaction with the customer service [40] [41]. This is where technology comes into play where we can detect the human emotions in a customer service call recording and predict the satisfaction of the customers without requiring the customer to take any additional action. By this approach, we will make every customer interaction count.

2. Background research

Over the years, various researches have been made for detecting human emotions from speech. The question is “why emotional awareness by machines is desirable?”. One of the reasons why a lot of efforts and research are made on this topic is that emotional awareness improves customer experience. Further, these emotions can be represented into different categories such as anger, disgust, fear, happiness, sadness, surprise and neutral. Most of the research on speech emotional recognition is implemented mainly through machine learning and deep learning techniques. These researches were mainly based on to determine which features in the audio influences the emotional recognition of the speech and which algorithms can be used effectively to classify human emotions.

In this project, an extensive study on the various researches is made on Speech Emotion Recognition (SER) and use this knowledge to implement this idea of predicting human emotion to solve a real-world problem. Here we concentrate on implementing this concept on the customer support field where we implement using Multi-layer Perceptron (MLP) to detect customer satisfaction through their emotions without the need for customer feedback.

In this section, we will look in detail on Artificial Neural Network (ANN) techniques and its implementation on speech emotion recognition (SER) and about multi-layer perceptron (MLP) which is used in this project to predict the human emotions.

2.1 Artificial Neutral Networks (ANN)

Artificial Neural Networks (ANN) is a computing model that can perform several tasks including clustering, classification and pattern recognition. Inspired from biological neural in the human mind which contains input, output, and hidden layers.

The input layer is the first layer which contains many nodes and these nodes represent individual features from each of the datasets. These input layers are connected to the next layer which is called a hidden layer. There is a connection that transfers the output from the previous unit as an input to the receiving unit. Each connection from one unit to another will contain its own assigned weight and this represents the strength of the connections between the units. This input will be multiplied by the weight assigned to the particular connection. Then the weighted sum is then computed with each of the connections that are pointing to this neuron. Later the sum is passed to the activation function which transforms

the result to a number between 0 and 1. The result obtained from the activation function is what will be passed to the next neuron in the next layer. This will occur repeatedly until it reaches the output layer. Also, the weight for each connection will change continuously to reach optimized weight for each connection as the model continues to learn from the data. Finally, the output layer consist of different units and each unit will represent different categories.

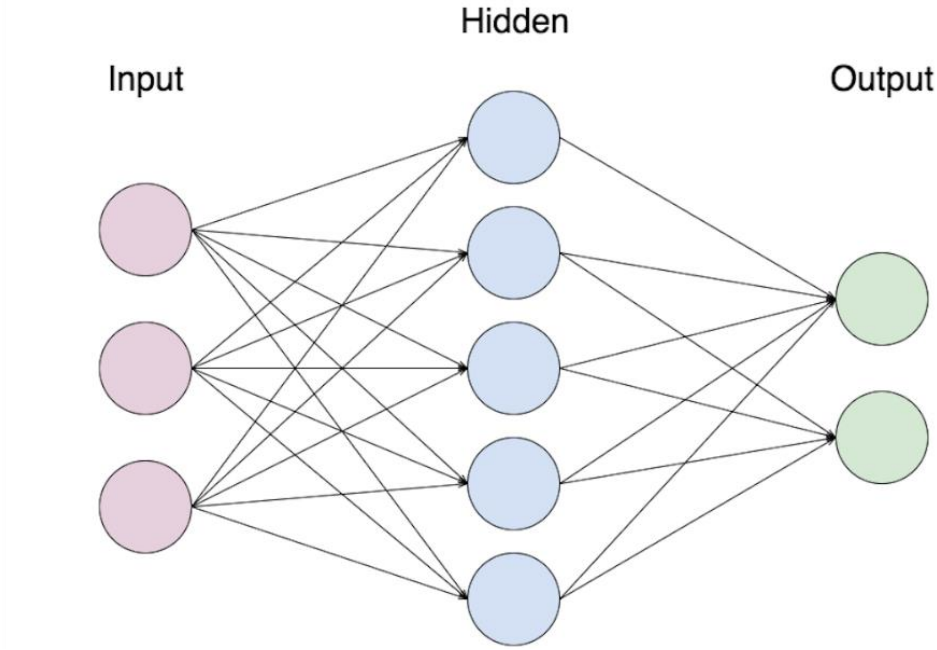

Figure 1. Artificial Neural Network (ANN)

Figure 2 describes the structure of the neuron where the weight of the connection is being multiplied by the previous layer's output values and then it is added at the next layer of the neuron(Σ). The argument of an activation function

($\varphi(.)$) is this result (w_0.bias + $w_1.x_1$ +…+ $w_n.x_n$) which represents the continuous values of the neuron output. A sigmoid function such as the logistic (Equation 1) can be used in the implementation of a sigmoid. The image set is [-1,1] which is a hyperbolic tangent function (Equation 2) or it can be also an image set which is [0,1] [6].

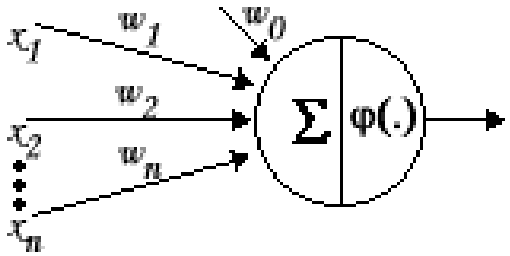

Figure 2. Neuron Structure

$$\varphi(z) = \frac{1}{1+e^{-z}} \quad (1)$$

$$\varphi(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}} \quad (2)$$

Equation 1, 2

Further, Artificial Neural Networks (ANN) can be classified into different types and few some them have been studied and researched over the years for the field of speech recognition. The most popular types of neural networks are Convolutional Neural Network (CNN), Multilayer Perceptron (MLP) and Recurrent Neural Network (RNN). Now, we will go through each one of these and the background research on these neural networks.

2.1.1 Multilayer Perceptron (MLP)

A particular configuration of neural networks called feed-forward neural network (FFNN) and more specifically the architecture is called as multilayer perceptron (MLP). This is often suited for different applications including pattern discrimination and classification, pattern recognition and also for speech recognition [7] [8]. Multi-layer Perceptron (MLP) mostly has three or more layers to classify data which cannot be linearly separated. In this network, all outputs of one layer are connected to each input of the succeeding layer. Hence, this architecture can be called as fully connected [9]. In a Multi-layer Perceptron (MLP), hyperbolic tangent or a non-linear activation function such as a logistic function is often used. This architecture is specifically used for machine translation and speech emotional recognition technologies [9]. As seen in Figure 3, the activation function g_l has one index: l. Each unit has its parameter $W_{l,u}$ and $B_{l,u}$, where u is the index of the unit, and l is the index of the layer. The vector y_{l-1} in each unit is defined as [$y_{l-1}^{(1)}$, $y_{l-1}^{(2)}$, $y_{l-1}^{(3)}$, $y_{l-1}^{(4)}$] [10]. This is used on supervised learning where input and output pairs are trained to create a model the dependencies between input and output.

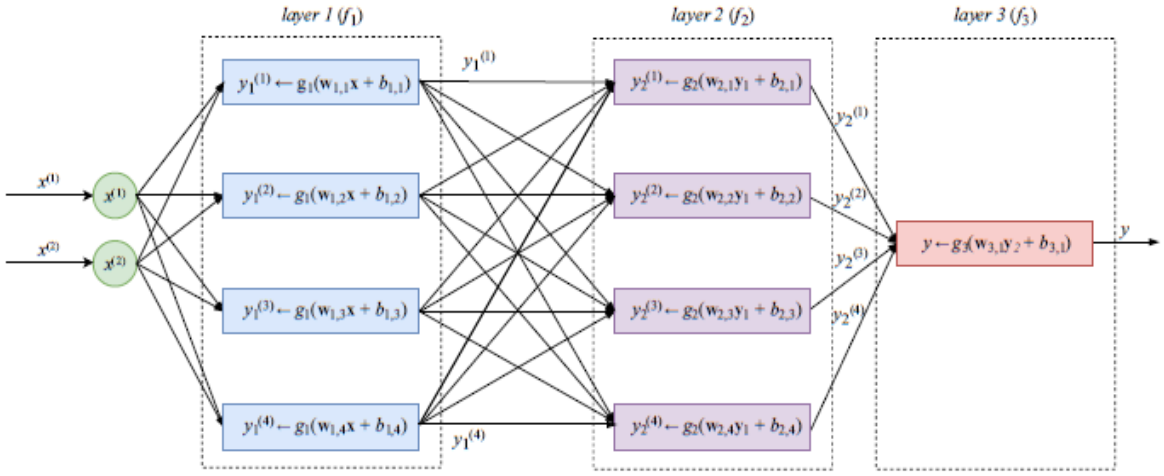

Figure 3. Multi-layer Perceptron (MLP) with two-dimensional input; two layers with four units and one output layer with one unit [10]

Suppose a Multi-layer Perceptron (MLP) network with one hidden layer and p x q x r structure, where q, p, and r are the number of neurons of the hidden layer, input layer, and output layer respectively. The output vector Z is related to an input vector X with below formulas:

$$Y_j = f_1(A_j + \sum_{i=1}^{p} W_{ij} X_i) \quad j = 1, 2, ..., q$$

$$Z_k = f_2(B_k + \sum_{j=1}^{q} V_{jk} Y_j) \quad k = 1, 2, ..., r$$

Equation 3

where A and B are bias vectors, W and V are network weights, f_2 and f_1 are the activation functions of neurons of the output layer and hidden respectively. The

activation function of the output layer (f_2) is linear, but the activation function of the hidden layer (f_1) is non-linear with the following equations [8]

$$f_1(x) = \frac{2}{1 + \exp(-2x)} - 1$$

$$f_2(x) = x$$

Equation 4

Since, this network has often preferred for speech emotional recognition [11] [12] [4], in this project we have implemented a speech emotion recognition (SER) system using Multi-layer Perceptron (MLP).

2.1.1.1 Training on Multilayer Perceptron (MLP)

After each processing, the weight of the connections is continuously adjusted and, in this way, the Multi-layer Perceptron (MLP) is being trained. The output might be wrong, so to train the network we have to infinite the correct output and it adjusts to get close to these outputs. So, it continuously checks the error between the output it produces and the actual output and continuously makes the necessary adjustment until it gets a minimum error. This process is called backpropagation which is also a supervised learning process. Its efficient algorithm for computing

gradients on neural networks using the chain rule. Backpropagation has two types: forward pass and backward pass [10].

2.1.2 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is the most popular technique to perform high-quality complex image recognition. Unlike Multi-layer Perceptron (MLP), where it uses fully connected hidden layers, this network uses a special network structure that contains convolution and pooling layers [13]. This convolution ply is different from normal traditional full connected layers in 2 aspects; one way is by receiving input only from the local area of input where each unit consist of some local region features, another way is that these convolution ply units are organized into the number of features maps [13]. Even some research has been made where merged deep Convolutional Neural Network (CNN) is implemented to classify emotion which contains 1D and 2D CNN branch to learn high-level features from raw audio audios [14]. There was also another experiment where a very deep convolution neural network (CNN) was proposed which consist of 10 convolution layers. This model was mainly tested against noisy speech audios [15].

Figure 4 explains the two ways in which input features of speech can be organized. It takes 40 Mel-frequency spectral coefficients (MFSC) features and 1st and 2nd derivatives with the context of 15 frames for each speech frame [13].

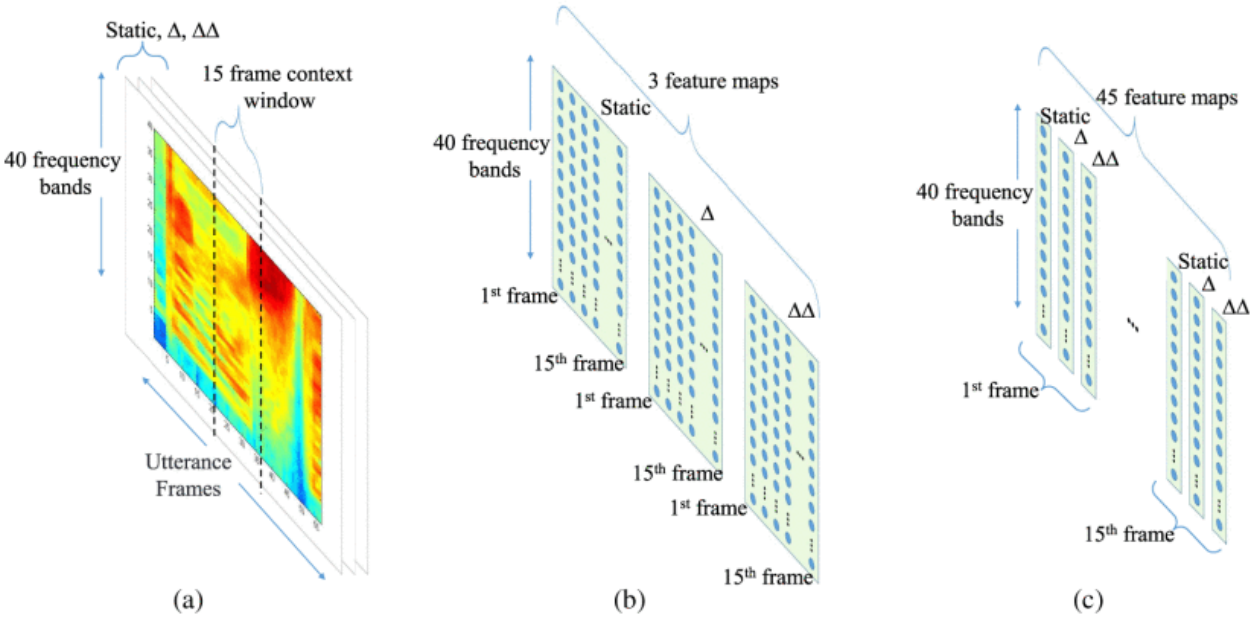

Figure 4. Convolution Neural Network (CNN); 40 MFSC features with 15 frames of the context window.

The main difference between Convolutional Neural Network (CNN) and other neural network techniques is that before transferring the output to the next layer, this network uses convolution operation on the input which makes a deeper network with fewer parameters. This is one of the reasons why it outperforms other techniques in image and video recognition. A simple image of a convolutional neural network (CNN) looks like figure 5.

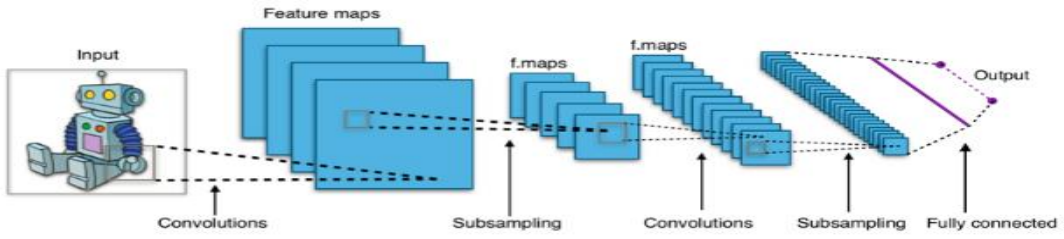

Figure 5. A simple CNN

2.1.3 Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) is another popular deep learning model that makes use of sequential information, unlike other neural networks that all input and output are independent of each other. In other words, Recurrent Neural Network (RNN) has a memory that learns about the information which has been calculated so for. It comes into play when other neural networks can't perform well when input data is interdependent in a sequence pattern. Figure 6 is an example of how a typical Recurrent Neural Network (RNN) looks like.

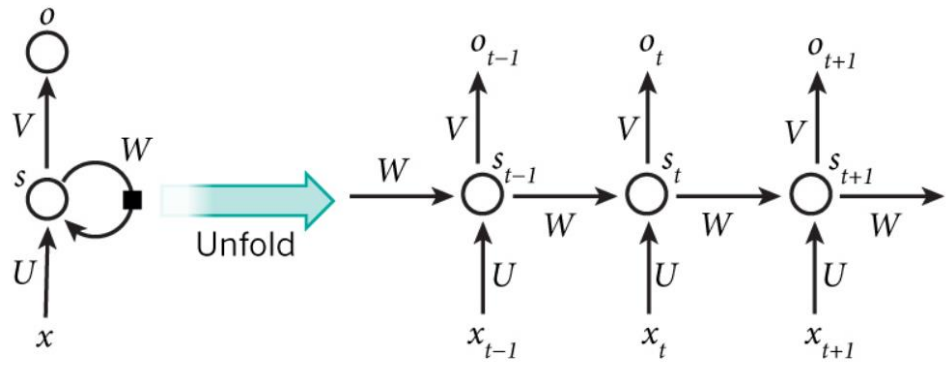

Figure 6. Recurrent Neural Network (RNN) [26]

There are many studies and researches which show the Recurrent Neural Network (RNN) is very difficult to train as it has to overcome the vanishing gradient problem [23] [24] [25]. Even many comparative studies on different types of Recurrent Neural Network (RNN) were conducted, among them the most popular and effective modified version is long short-term memory based recurrent neural network (LSTM-RNN) [27] [28]. The gradient problem which is talked earlier is resolved in this modified version. In Long Short Term Memory (LSTM), the given time lags of duration are predicted, processed and classified where backpropagation is used to train the model.

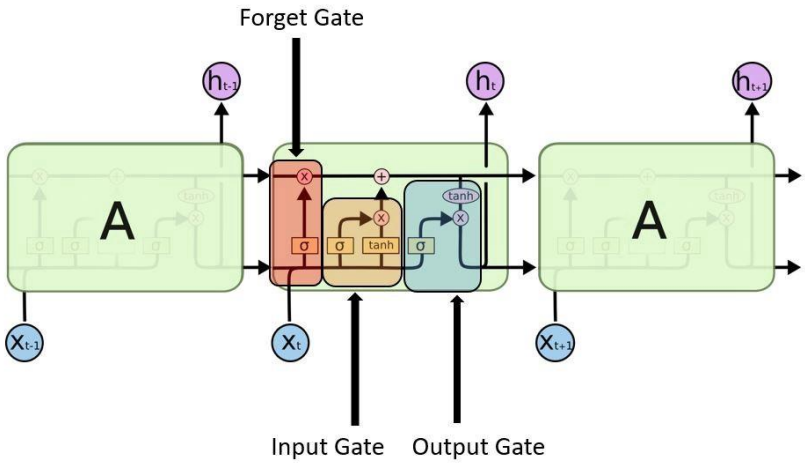

Figure 7. Long Short Term Memory (LSTM) network: three gates, input, forget and output gate [29].

The input gate determines the input value that has to be used to modify the memory. Whereas, the forget gate decides the details that will be removed from the block and finally in the output gate, the output is decided by input and memory of the block [29].

2.2 Which technique to select?

Even though the Convolutional Neural Network (CNN) appears to be an effective technique to be used in image recognition and Recurrent Neural Network (RNN) is effective for predictive analytics and statistical analysis. Whereas Multi-layer Perceptron (MLP) outperforms other techniques in data visualization, data compression, and encryption, classification and regression prediction problems [19] [20]. In the speech emotional recognition field, neural networks are rapidly replacing the existing techniques. There is always a chance of improvement in the existing technologies. For instance, even Google has changed its technology on speech recognition over the years. First 2009, Google Voice used the Gaussian Mixture Model (GMM) model, then later they implemented deep neural network techniques. In 2012, an Android speech recognizer was launched using long short-term memory based recurrent neural network (LSTM-RNN) [22].

In reality, a lot of effort has been made in speech recognition over many years to adopt the best algorithm for optimal performance. But in fact, it's not easy to identify a single solution as the best among all other techniques. Many research and study have made a comparative analysis on various techniques and few types of research were also made by implemented hybrid techniques by combining more than one algorithm/technique [16] [17] [18] [21]. Therefore, in this project, we are going to concentrate on various ways to effectively implement one single technique called Multi-layer Perceptron (MLP) to achieve the best possible performance results.

2.3 Other related works

Over the years, various researches have been made for detecting human emotions from speech. Most of the research on speech emotion recognition (SER) is implemented mainly through machine learning techniques such as a deep neural network (DNN) [1]. Whereas, R. Ram, H. K. Palo and M. N. Mohanty [4] have attempted to enhance the emotional speech signal adaptively before classification. Here, the author has concentrated on differentiating fear and neutral emotion alone in a speech. And, few types of research were focused on determining an ideal combination of extracted features along with the implementation of several

different classifiers for efficient speech emotion recognition (SER) system [2]. Even several studies were concentrated on gender identification in the speech using Multi-layer Perceptron (MLP), Gaussian Mixture Model (GMM), Learning Vector Quantization (LVQ) [3]. There are also attempts to analyze and establish even for human emotion from a whispered speech. Generally, the features which are related to pitch are often used as extraction features to determine the emotions. But for whispered speech, these parameters should be modified or substituted. Considering other parameters such as sentence duration and energy is the deciding factor in the prediction of emotion in whispered speech [5].

There is another similar research work that was based on Speech Emotion Recognition (SER) using the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) dataset [42]. That research work was mainly concentrating on the comparison of performance between Convolutional Neural Network (CNN), Hidden Markov Models (HMM) and Recurrent Neural Network (RNN). The difference between our research work and this comparative performance research is that we concentrate on a particular neural network technique called Multi-layer Perceptron and try to modify the model training and its selection of features to give

the best possible accuracy. Whereas comparative research work focuses on the other three techniques and their performance against each other.

3. Methodology

This section outlines the process that went into the implementation of the project.

3.1 Project Structure

This project has a typical architecture of every deep learning project. Figure 8 outlines the process involved in the speech emotion recognition (SER) system.

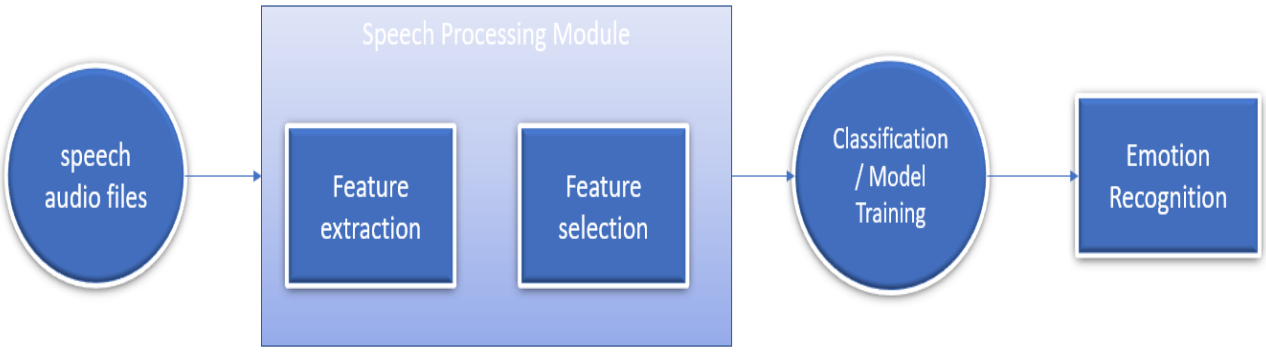

Figure 8. Project Flow

This Emotional recognition system is based on the following fundamental components, input dataset (speech audio files), feature extraction, feature selection, classification/model training, and emotion recognition. First, the data is taken from the Ryerson Audio-Visual Database of Emotional Speech and Song

(RAVDESS) dataset which is an audio input file. Then this input data is subjected to data cleaning if needed based on the noisy component in the data. After this process, features from the data are extracted using various methods and then relevant features are selected from these extracted features. After that, the selected feature is used to train the model. Model training is implemented by the classification process using a neural network technique called Multi-Layer Perceptron (MLP). This feedforward Artificial Neural Network (ANN) is used for training and testing the model. Finally, the trained model is used for predicting emotion for any input data. Once this implementation is completed, the same model is used to predict the emotion of live recording to test the performance of the model. In the following section, each of these steps is discussed in a detailed manner.

3.2 Dataset

In this project, the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) dataset is used as input data for emotion recognition. It's a free dataset that has 7356 recordings and each rated 10 times on emotional intensity, genuineness, and validity. Untrained research participants of 247 individuals from North America provided the ratings for this voice dataset [37]. This dataset contains

both audio and video files but only the audio files are considered as this project is dealing with finding emotions from speech and also the sample rate is lowered on all the files to effective implementation.

3.2.1 Data Cleaning

For general deep learning implementation, most often the input data has to be cleaned before undergoing the further process. For the case of audio input data, the dataset can contain noisy data which may disrupt the efficiency of the model. Therefore, these noisy components are removed by various methods such as Beat synchronous features aggregation [1] and adaptive methods such as Normalized Least Mean Square (NLMS) algorithm, Least Mean Square (LMS) algorithm, and Recursive Least Square (RLS) algorithm [4]. We also have to convert the raw audio data to .wav format. This is to make sure it is compatible with python for reading the audio files. There are various open-source modules such as SoX [39] which can be used for converting the audio file format.

These methods to filter and clean the data are needed only if the input data depends on it. But for this project, the RAVDESS dataset is used which is already a cleaned dataset that is ready to use without any further filtration. Even the audio

files are in .wav format which is compatible with python. Hence, throwing the dataset for the cleaning process is considered to be overkill in this case.

3.2.2 Voice Content

Emotion in a speech can be often categorized into basic emotion types which include happiness, sadness, anger, disgust, fear, surprise, and neutral emotions. To identify such emotions, parameters like intensity, pitch, and rate of spoken words are taken into consideration. These emotions can also have a clear relationship with the acoustic parameters which is shown in table1.

Emotions	Pitch	Intensity	Speaking rate	Voice quality
Anger	abrupt on stress	much higher	marginally faster	breathy, chest
Disgust	wide, downward inflections	lower	very much faster	grumble chest tone
Fear	wide, normal	lower	much faster	irregular voicing
Happiness	much wider, upward inflections	higher	faster/slower	breathy, blaring tone
Joy	high mean, wide range	higher	faster	breathy; blaring timbre
Sadness	slightly narrower	downward inflections	lower	resonant

Table 1. Observed emotional variation in speech.

The parameters of voice are always considered for analysis and prediction of emotion in the speech. Also, there can be various factors that might affect the

mapping from acoustic parameters to the emotion itself. The possible factors include high voice variations if the speaker is acting and also the mood of the speaker. The fundamental of the emotions can be illustrated with the help of the graph shown in figure X. The x-axis represents the valence which is a positive and negative effect in emotion. Whereas, the y-axis represents the arousal which describes the intensity of excitement or calmness [38].

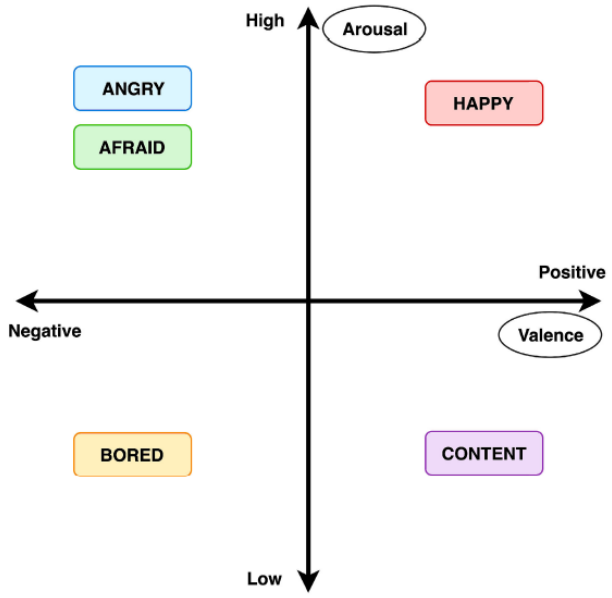

figure 9. graph representation of emotions.

3.3 Feature extraction and selection

Now meaningful and useful features are needed to perform a classification task for the prediction of the emotions. To extract the features from the audio samples, we need a python library called Librosa [31] [32]. Librosa is a python library for music and audio analysis. An audio file may contain the various number of features that can be extracted for further processing. But these features do not need to be useful for speech emotion recognition. Taking all the features might end up with increased computation power and time. Even it may result in poor classification performance. As these problems might lead to the curse of dimensionality problem, it is preferred that we invest enough effort in selecting the relevant features needed for Speech Emotion Recognition (SER). Therefore, the elimination of irrelevant features will help in classifying the emotion in a better and efficient way.

In this project, three features are considered, and they are extracted from the audio samples using Librosa package in python. The three features are Mel Frequency Cepstral Coefficient (mfcc), chroma and Mel Spectrogram Frequency (mel). The below section explains in detail about the three features that are used for the classification process.

3.3.1 Mel Spectrogram Frequency (MEL)

Mel Spectrogram (Mel) mainly relies on the time-frequency domain which is used to the purpose of extracting time-frequency domain information. Emotional informative features are captured by converting the input audio samples. Figure X illustrates the implementation of extracting the Mel spectrogram frequency feature from the raw audio sample.

```
[6]:
import numpy as np
import librosa.display
import matplotlib.pyplot as plt

filename = 'aud.wav'
y, sr = librosa.load(filename)
test_sound, _ = librosa.effects.trim(y)

S = librosa.feature.melspectrogram(test_sound, sr=sr, n_fft=2048,
                                   hop_length=512,
                                   n_mels=128)
S_DB = librosa.power_to_db(S, ref=np.max)
librosa.display.specshow(S_DB, sr=sr, hop_length=512,
                         x_axis='time', y_axis='mel');
plt.colorbar(format='%+2.0f dB');
```

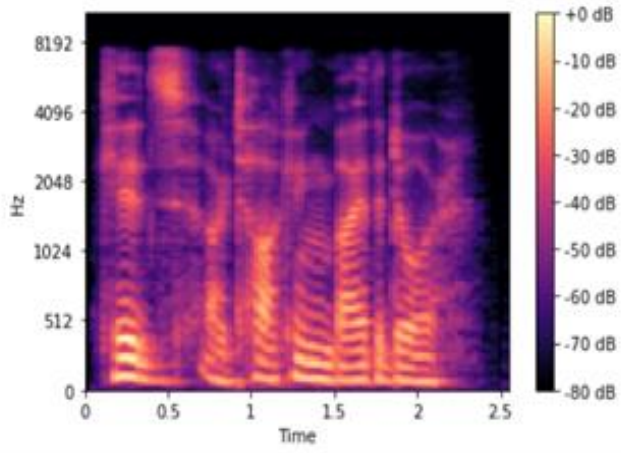

figure 10. Mel Spectrogram Frequency

3.3.2 Mel Frequency Cepstral Coefficient (MFCC)

Similar to Mel Spectrogram Frequency, Mel Frequency Cepstral Coefficient (MFCC) also relies on the time-frequency domain intending to convert raw audio input into Emotional informative features. It also represents the short-term power spectrum of a sound. Librosa.feature.mfcc function is used to compute Mel Frequency Cepstral Coefficient (MFCC) from an audio input signal. Figure 11 explains how to compute Mel Frequency Cepstral Coefficient (MFCC) from the raw audio signal.

```
import numpy as np
import librosa.display
import matplotlib.pyplot as plt

filename = 'aud.wav'
x, sr = librosa.load(filename)

mfccs = librosa.feature.mfcc(x, sr=sr)
print(mfccs.shape)
#Displaying  the MFCCs:
librosa.display.specshow(mfccs, sr=sr, x_axis='time')
```

```
(20, 173)
<matplotlib.axes._subplots.AxesSubplot at 0x2aefcb99548>
```

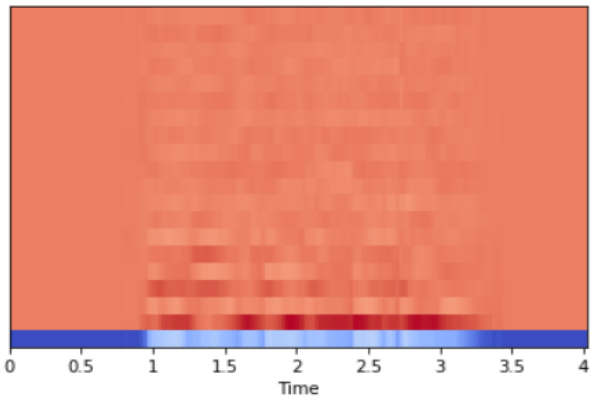

figure 11. Mel Frequency Cepstral Coefficient (MFCC)

Figure 11, Mel Frequency Cepstral Coefficient (MFCC) computed 20 MFCCs over 173 frames. This model the characteristics of the human voice. In this feature, the input audio signal passes through a pre-emphasis filter and transforms into frames and window function is applied to each frame. Figure 12 illustrates the implementation of feature scaling with each coefficient dimension having zero-unit variance and zero mean.

```
import sklearn
mfccs = sklearn.preprocessing.scale(mfccs, axis=1)
print(mfccs.mean(axis=1))
print(mfccs.var(axis=1))
librosa.display.specshow(mfccs, sr=sr, x_axis='time')
```

```
[ 4.37671736e-15  5.51902775e-16 -7.05922154e-17  4.01733881e-16
  9.14490064e-17  4.01092133e-16  1.91240729e-16 -4.46014452e-17
 -1.15514534e-17  1.54019379e-17  1.28349483e-17  7.25174577e-17
 -1.08776187e-16 -1.63003843e-16 -1.25782493e-16  2.00225193e-16
 -1.09097060e-17  4.23553293e-17  6.37736492e-18  1.06530071e-16]
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
<matplotlib.axes._subplots.AxesSubplot at 0x2aefc57c988>
```

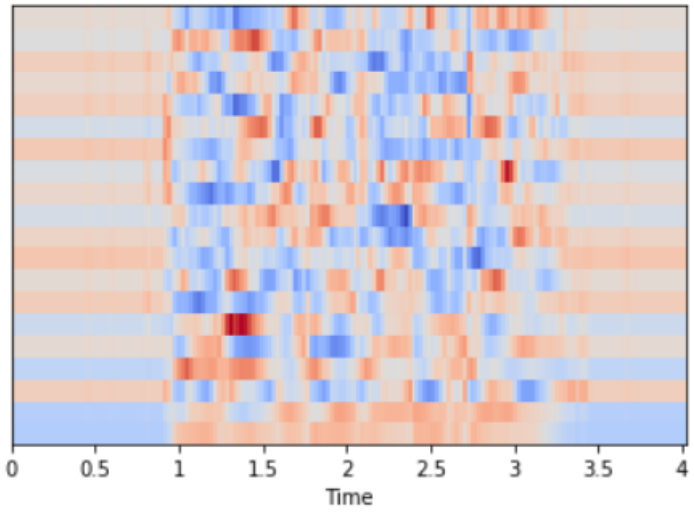

Figure 12. Feature scaling – Mel Frequency Cepstral Coefficient (MFCC)

3.3.3 Chroma

The pitch class information is obtained by extracting chroma features from the raw audio signal. This is a representation of music audio in which the entire spectrum is projected onto 12bins representing the 12 distinct chromas of the musical octave. By normalization, this feature can be transformed into invariant to dynamic variations. There are two main chroma features are Chroma deviation (standard deviation of 12 chroma coefficients) and Chroma vector (12 element representation of the spectral energy). It can be implemented as shown in figure x.

```
import numpy as np
import librosa.display
import matplotlib.pyplot as plt

filename = 'aud.wav'
x, sr = librosa.load(filename)

hop_length = 512
chromagram = librosa.feature.chroma_stft(x, sr=sr, hop_length=hop_length)
plt.figure(figsize=(15, 5))
librosa.display.specshow(chromagram, x_axis='time', y_axis='chroma', hop_length=hop_length, cmap='coolwarm')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2aefc425dc8>
```

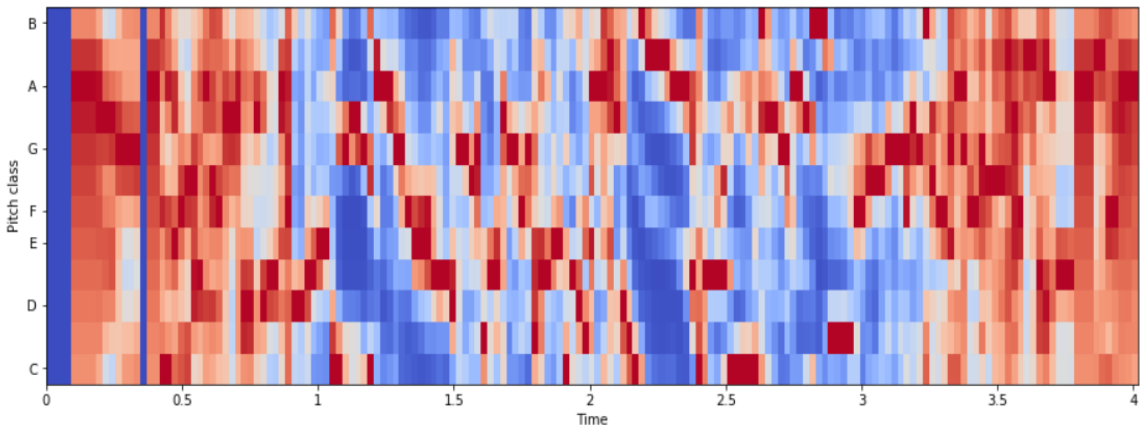

figure 13. Chroma

3.4 Classification or Model training

Once the necessary features are extracted from the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) dataset, the dataset is then split into training and testing sets. In this project, we take 25% of the dataset as the testing dataset, the remaining 75% is used for training the model. Before training the model, the emotions are to be defined against this dataset for prediction. The dataset contains eight types of emotional mood but here the model is going to be trained to predict four basic types of emotions such as calm, happy, fearful and disgust. Figure 14 explains the dictionary definition to hold various emotions.

[3]:
```
# feeling that we want to predict
mood_state=['calm mood', 'happy mood', 'fearful mood', 'disgust mood']

# feeling defined in RAVDESS dataset
human_emotions={
  '01':'neutral mood',
  '02':'calm mood',
  '03':'happy mood',
  '04':'sad mood',
  '05':'angry mood',
  '06':'fearful mood',
  '07':'disgust mood',
  '08':'surprised mood'
}
```

figure 14. Emotion definition

After defining emotion, features are extracted as explained in the previous section and then MLPClassifier from the sklearn library is used to train the model using the dataset. To achieve better accuracy and better results of prediction for the Multi-layer Perceptron (MLP) model, the proper combination of parameters in MLP Classifier has to be defined. The parameter set for MLP Classifier depends on the kind of the dataset and its extracted features. To find the best parameters, GridSearchCV from sklearn library is implemented. GridSearchCV helps to tune the hyperparameter which directly how well a model trains. All possible parameters are defined in GridSearchCV and then it evaluates the performance of a given parameter combination and helps to pick the best parameters for that specific dataset. Figure 15 describes how to implement GridSearchCV.

```
[8]: # Use GridSearchCV to find the best parameter combination for MLP classfier
     # reference for GridSearchCV- https://datascience.stackexchange.com/questions/36049/how-to-adjust-the-hyperparameters-of-mlp-classifier-to-ge

     combination_parameter = {'hidden_layer_sizes': [(150,150,150), (200, 200, 200), (300,300,300), (300,), (200,), (400,)],
                              'activation': ['identity', 'logistic', 'tanh', 'relu'],'solver': ['lbfgs', 'sgd', 'adam'],
                              'alpha': [0.0001, 0.05, 0.01], 'max_iter': [800,400,200], 'learning_rate': ['constant','adaptive','invscaling'],}
     model = GridSearchCV(MLPClassifier(), combination_parameter, n_jobs=-1, cv=3)
     model.fit(train_feature,train_emotion)
```

figure 15. GridSearchCV

In figure 15, the model is trained with all parameter combinations to find the optimal combination which achieves the best results. Once the model is trained with GridSearchCV, the best parameters are identified as shown in figure 16.

```
# To display best parameter combination

print('Best parameters found:\n', model.best_params_)
```

```
Best parameters found:
 {'activation': 'tanh', 'alpha': 0.05, 'hidden_layer_sizes': (400,), 'learning_rate': 'adaptive', 'max_iter': 800, 'solver': 'adam'}
```

figure 16. Best parameter using GridSearchCV

Using the best parameters, we train our model to achieve better performance in the prediction of emotions. Note that the maximum iterations defined here is 800 which doesn't mean the iteration has to be performed for 800 times, it all depends on the training loss of the model. For example, in this case, we had till 476 iterations after which the model training stopped because of unimproved training loss which stopped improving at tol=0.000100 for 10 consecutive epochs after 476th iterations. Figure 17 shows the details of epochs and iterations when the model is trained. After every iteration of training the model, the training loss is expected to be lowered and if not, the training is stopped, and the model is not further trained again. The lower training loss indicates a better

model being trained and it is being calculated on validation, training and its interpretation is how well the model is doing for these two sets.

```
# After finding the best parametercombination using GridSearchCV, comment the GridSearchCV section and use those parameter
# This option enables you to plot the loss curve in graph!!!
trained_model=MLPClassifier(hidden_layer_sizes=(400), activation = 'tanh', solver = 'adam', alpha = 0.05,  max_iter=800,
                            learning_rate='adaptive', verbose='true')

model= trained_model.fit(train_feature,train_emotion)
```

```
Iteration 1, loss = 2.60182830
Iteration 2, loss = 1.48221069
Iteration 3, loss = 1.55620743
Iteration 4, loss = 1.53497943
Iteration 5, loss = 1.36899939
Iteration 6, loss = 1.28344770
Iteration 7, loss = 1.27582713
Iteration 8, loss = 1.26830696
Iteration 9, loss = 1.20593196
Iteration 10, loss = 1.17446503
.
.
.
.
Iteration 465, loss = 0.02465280
Iteration 466, loss = 0.02458867
Iteration 467, loss = 0.02456577
Iteration 468, loss = 0.02450426
Iteration 469, loss = 0.02444631
Iteration 470, loss = 0.02439532
Iteration 471, loss = 0.02434274
Iteration 472, loss = 0.02434589
Iteration 473, loss = 0.02425452
Iteration 474, loss = 0.02430728
Iteration 475, loss = 0.02416514
Iteration 476, loss = 0.02428933
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
```

figure 17. Model iteration

4. Experimental Setup

This section provides details on how to reproduce the same project. This project can be set up on a Windows or Mac system, and it is implemented in python language. The sklearn libraries of python will make use of CPU cores to train the model. The model training can be completed at maximum speed if all CPU cores are allowed to be used. For this project, GridSearchCV function to find the best possible parameter combination will take around 6 to 10 hours to compute and find the best parameters for MLP Classifier. But if the model training needs to be completed at a faster pace, then the TensorFlow library can be used. This library has the option to use GPU instead of CPU cores which will reduce 50% of computation time. This can be implemented if the system has a gaming graphics card that has a powerful GPU. The features extracted in this project are Mel Frequency Cepstral Coefficient (MFCC), Chroma and Mel Spectrogram Frequency (MEL). The dataset used is Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) which is the free dataset. This RAVDESS dataset contains both the video and audio data is around 25 Gb which consists of voice data of 24 different actors. But in this project, we concentrate only on the audio dataset and can be downloaded from this Kaggle link. For this project, the sample rate is

lowered on all the audio files and this lowered sample rate audio files can be downloaded from this link. It contains 1440 different audio files. The python code to reproduce the project can be also downloaded from this link. All the necessary libraries mentioned in the code have to installed using any type of python tool such as Pip.

5. Results and discussion

This section provides the results of the project using the training and testing dataset. This section also provides the results of live recorded voice prediction using the trained model of Multi-layer Perceptron (MLP) classifier.

5.1 Results of RAVDESS dataset

After the model is trained with Multi-layer Perceptron (MLP) classifier for the training dataset. It is then used to predict the emotions of the testing dataset. Once the prediction is completed the accuracy of the prediction is computed which is shown in figure 18.

```
print("Train set accurary: %f" % model.score(train_feature, train_emotion))
print("Test set accurary: %f" % model.score(test_feature, test_emotion))


def calculate_accuracy(confusion_matrix):
   X = confusion_matrix.trace()
   Y = confusion_matrix.sum()
   return X / Y


cm = (100*(confusion_matrix(predicted_emotion, test_emotion)))
accuracy_value=((calculate_accuracy(cm))*100)
print("Accuracy of MLPClassifier :{:.2f}% ".format(accuracy_value))
```

```
Train set accurary: 1.000000
Test set accurary: 0.822917
Accuracy of MLPClassifier :82.29%
```

figure 18. Model accuracy

The model came up with an accuracy of 82% which is good enough when comparing to the amount of data used for training the model. Our model would perform much better if more data is used for the training phase. The following figure 19 shows our prediction with the actual emotion of test data.

```
#to display in table for predicted vs expected emotions of test data set

table = PrettyTable()

table.field_names = ["Predicted Emotion", "Expected Emotion"]

for pred_emotion, expected_emotion in zip(predicted_emotion, test_emotion):
    table.add_row([pred_emotion,expected_emotion])

print(table)
```

```
+-------------------+------------------+
| Predicted Emotion | Expected Emotion |
+-------------------+------------------+
|     happy mood    |    happy mood    |
|    fearful mood   |    calm mood     |
|     happy mood    |    happy mood    |
|     happy mood    |    happy mood    |
|    disgust mood   |   disgust mood   |
|     calm mood     |    calm mood     |
|    disgust mood   |    happy mood    |
|     happy mood    |    happy mood    |
|    disgust mood   |   disgust mood   |
|     happy mood    |    happy mood    |
|     happy mood    |    happy mood    |
|    disgust mood   |   disgust mood   |
|    fearful mood   |    happy mood    |
|     happy mood    |    happy mood    |
|    disgust mood   |   disgust mood   |
|     happy mood    |   fearful mood   |
|     calm mood     |    calm mood     |
|    disgust mood   |    happy mood    |
|    disgust mood   |   disgust mood   |
|    disgust mood   |   disgust mood   |
|     calm mood     |    calm mood     |
|    disgust mood   |   disgust mood   |
|    disgust mood   |   disgust mood   |
|     calm mood     |    calm mood     |
|     happy mood    |    happy mood    |
|     happy mood    |    happy mood    |
|    disgust mood   |   disgust mood   |
|     happy mood    |   disgust mood   |
|    fearful mood   |    calm mood     |
|     happy mood    |    calm mood     |
|     happy mood    |    happy mood    |
|    disgust mood   |   disgust mood   |
|     happy mood    |    happy mood    |
|    fearful mood   |   fearful mood   |
```

figure 19. Predicted Emotion Vs Expected Emotion

To evaluate the quality of predictions from a classification algorithm, this project uses a classification report to show metrics such as precision, recall, and f1-score on a per-class basis. Figure 20 represents the classification report for the prediction results.

```
print('Results on the test set:')
print(classification_report(defined_emotion, predicted_emotion))
```

```
Results on the test set:
              precision    recall  f1-score   support

   calm mood       0.94      0.88      0.91        57
disgust mood       0.83      0.81      0.82        48
fearful mood       0.69      0.84      0.76        37
  happy mood       0.79      0.74      0.76        50

    accuracy                           0.82       192
   macro avg       0.81      0.82      0.81       192
weighted avg       0.83      0.82      0.82       192
```

figure 20. Classification report

To debug any neural network, the loss curve is used. The loss curve shows a snapshot of the direction in which the network learns and its training process. The loss curve is calculated and plotted in the form of a graph in figure 22. And figure 21 shows the effects of the learning rate on Loss.

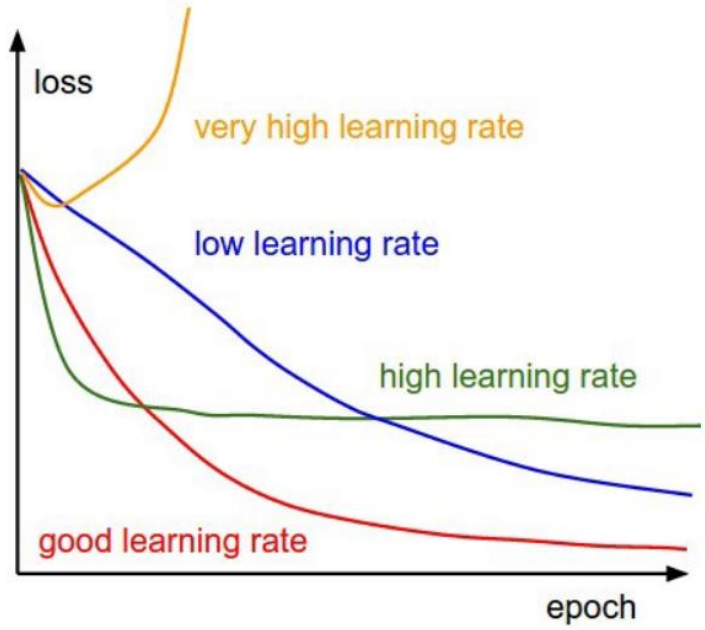

figure 21. Types of learning rate

```
#plotting loss curve
graph.plot(model.loss_curve_)
graph.title('model loss')
graph.ylabel('loss')
graph.xlabel('epoch/iteration')
graph.legend(['Loss Curve'], loc='upper right')
graph.show()
```

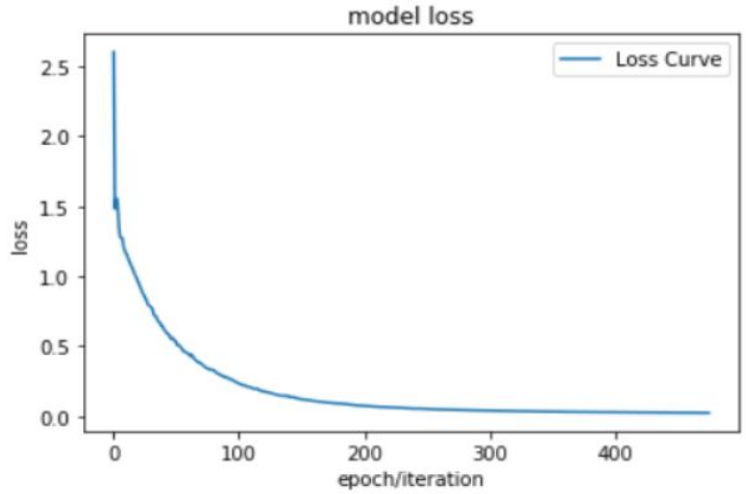

figure 22. Training loss curve

5.2 Results of live audio recording

The project attempts to record live audio speech for 4 seconds and uses this audio data as input to the trained Multi-layer Perceptron (MLP) model to predict the emotions. As this model is trained to predict only 4 basic types of emotion such as calm, happy, fearful and disgust mood, all the live recording attempts were aimed to mimic only these 4 types of emotion. The live recording is made by the author and so, this audio data is completely different in terms of voice accent and pitch modulation when compared with the dataset which is used to train and test the model.

First, the python code is used to record the live voice which then converts this audio to a mono audio channel with a sample rate of 16 Khz and bits per sample value as 16. After the recording is completed successfully, the audio is given as input to extract the features. Once the features are extracted, the next step is to feed these features to the Multi-layer Perceptron (MLP) model to predict its emotion. The live recording was attempted various times to mimic different emotions. The prediction accuracy for live audio recording is not accurate as compared to testing dataset prediction accuracy. The model predicted 6 times correctly out of 10 audio recording attempts. This difference is mainly due to the

lack of large variations in the dataset which is essential for predicting emotions for any given kind of audio data. Also, the dataset used for training the model is a collection of voiced actors used for researches of speech analysis. We also need to train our model with a non-acted voice dataset which is more related to real-world data as these two types of data vary drastically in terms of how the emotions are expressed. Through this experiment, it is clear that with more variety of data for training, the model will have the ability to predict emotion as accurately as possible. Figure 23 shows one of the results of live record prediction.

```
def live_load_data():
    x=[]
    file_name=os.path.basename("D:\\studies\\Term3\\Capstone Project\\liveVoice.wav")
    feature=audioFeatureExtract(file_name, set_mfcc=True, set_chroma=True, set_mel=True)
    x.append(feature)
    print("feature extracted successfully")
    return x
```

```
live_pred_feature = live_load_data()
```

```
feature extracted successfully
```

```
live_predicted_emotion=model.predict(live_pred_feature)
```

```
print("The emotion for live recording is",live_predicted_emotion[0])
```

```
The emotion for live recording is disgust mood
```

figure 23. Live audio record prediction

6. Conclusion

From this experiment on Speech Emotion Recognition (SER) using Multi-layer Perceptron (MLP), it is seen that predicting emotion from the voice data involves various challenges. One of the most important factors that we found in this project is the choice of a good emotional speech database. The more quantity and variety of data to train, the better the performance of the model. The next challenge is the importance of extracting effective features from the dataset which directs impacts on the performance of the trained model. The last but not the least is the choice of selecting and designing efficient classifiers using machine learning and neural network algorithms.

The Multi-layer Perceptron (MLP) model achieves a prediction accuracy of 82% which is a very good result when compared to the amount of data used for training the model. Although the prediction accuracy for live recording didn't show the same level of accuracy when compared with the testing set of Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) dataset. It has to be noted that these live voice data are completely different from the training dataset in terms of accent, pitch modulation, and other aspects. Still, this model can be further improvised to give better results by training large databases of voice data.

The large database should ideally also contain real-time voice data in addition to the voice of actors used for researches of speech analysis. This consideration is extremely important when the model is aimed to be used in real-world problems because different individuals reveal their emotions in a diverse degree and manner. Therefore, just the data of actors from research labs is not enough for training an efficient model.

Once the model is improvised considering all these factors, it can be easily used in the customer support field to help the companies to understand the customer's satisfaction through their emotions without the need for customer feedback. The futuristic work of this project will involve training the model with a large real-time dataset along with the typical actor-based voice dataset and also implementing a more sophisticated hybrid classification model by combining various classifiers to accurately predict the non-acted speech data.

7. References

[1]. K. Tarunika, R. B. Pradeeba and P. Aruna, "Applying Machine Learning Techniques for Speech Emotion Recognition," *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Bangalore, 2018, pp. 1-5.

[2] N. Kamaruddin and A. Wahab, "Emulating human cognitive approach for speech emotion using MLP and GenSofNN," *2013 5th International Conference on Information and Communication Technology for the Muslim World (ICT4M)*, Rabat, 2013, pp. 1-5.

[3] R. Djemili, H. Bourouba and M. C. A. Korba, "A speech signal based gender identification system using four classifiers," *2012 International Conference on Multimedia Computing and Systems*, Tangier, 2012, pp. 184-187.

[4] R. Ram, H. K. Palo and M. N. Mohanty, "Recognition of fear from speech using adaptive algorithm with MLP classifier," *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, Nagercoil, 2016, pp. 1-5.

[5] Y. Jin, Y. Zhao, C. Huang and L. Zhao, "Study on the Emotion Recognition of Whispered Speech," *2009 WRI Global Congress on Intelligent Systems*, Xiamen, 2009, pp. 242-246.

[6] M. A. Sovierzoski, F. I. M. Argoud and F. M. d. Azevedo, "Evaluation of ANN Classifiers During Supervised Training with ROC Analysis and Cross Validation," 2008 International Conference on BioMedical Engineering and Informatics, Sanya, 2008, pp. 274-278. URL:

[7] A. Ahad, A. Fayyaz and T. Mehmood, "Speech recognition using multilayer perceptron," IEEE Students Conference, ISCON '02. Proceedings., Lahore, Pakistan, 2002, pp. 103-109 vol.1.

[8] M. P. Ghaemmaghami, H. Sameti, Farbod Razzazi, B. BabaAli and Saeed Dabbaghchian, "Robust speech recognition using MLP neural network in log-spectral domain," 2009 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Ajman, 2009, pp. 467-472.

[9] A Comprehensive Guide to Types of Neural Networks. Retrieved from URL.

[10] Andriy, B. (2019). The Hundred-Page Machine Learning Book, page 60-65.

[11] F. Valente, M. M. Doss, C. Plahl, S. Ravuri and W. Wang, "Transcribing Mandarin Broadcast Speech Using Multi-Layer Perceptron Acoustic Features," in IEEE Transactions on Audio, Speech, and Language Processing, vol. 19, no. 8, pp. 2439-2450, Nov. 2011.

[12] H.K. Palo, M.N. Mohanty, M. Chandra, "Emotion recognition using MLP and GMM for Oriya language" in International Journal of Computational Vision and Robotics, Inderscience, 2015.

[13] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn and D. Yu, "Convolutional Neural Networks for Speech Recognition," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 22, no. 10, pp. 1533-1545, Oct. 2014.

[14] J. Zhao, X. Mao and L. Chen, "Learning deep features to recognise speech emotion using merged deep CNN," in IET Signal Processing, vol. 12, no. 6, pp. 713-721, 8 2018.

[15] Y. Qian, M. Bi, T. Tan and K. Yu, "Very Deep Convolutional Neural Networks for Noise Robust Speech Recognition," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 24, no. 12, pp. 2263-2276, Dec. 2016.

[16] P. Pujol, S. Pol, C. Nadeu, A. Hagen and H. Bourlard, "Comparison and combination of features in a hybrid HMM/MLP and a HMM/GMM speech recognition system," in IEEE Transactions on Speech and Audio Processing, vol. 13, no. 1, pp. 14-22, Jan. 2005.

[17] M. Sundermeyer, I. Oparin, J. -. Gauvain, B. Freiberg, R. Schlüter and H. Ney, "Comparison of feedforward and recurrent neural network language models," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, 2013, pp. 8430-8434.

[18] T. Iliou and C. Anagnostopoulos, "SVM-MLP-PNN Classifiers on Speech Emotion Recognition Field - A Comparative Study," 2010 Fifth International Conference on Digital Telecommunications, Athens, 2010, pp. 1-6.

[19] When to Use M++LP, CNN, and RNN Neural Networks. Retrieved from machine learning mastery.

[20] A Comprehensive Guide to Types of Neural Networks. Retrieved from URL.

[21] M. Sundermeyer, I. Oparin, J. -. Gauvain, B. Freiberg, R. Schlüter and H. Ney, "Comparison of feedforward and recurrent neural network language models," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, 2013, pp. 8430-8434.

[22] The neural networks behind Google Voice transcription. Retrieved from Google AI Blog

[23] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, “Advances in optimizing recurrent networks,” in Proc. 2013 IEEE Int. Conf. Acoust., Speech, Signal Process., 2013, pp. 8624–8628.

[24] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in Proc. Int. Conf. Mach. Learn., vol. 28, 2013, pp. 1310–1318.

[25] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, “Learning precise timing with LSTM recurrent networks,” J. Mach. Learn. Res., vol. 3, no. Aug, pp. 115–143, 2002

[26] Recurrent Neural Networks Tutorial. Retrieved from URL.

[27] E. Song, F. K. Soong and H. Kang, "Effective Spectral and Excitation Modeling Techniques for LSTM-RNN-Based Speech Synthesis Systems," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 25, no. 11, pp. 2152-2161, Nov. 2017.

[28] G. Gelly and J. Gauvain, "Optimization of RNN-Based Speech Activity Detection," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 26, no. 3, pp. 646-656, March 2018.

[29] Understanding RNN and LSTM. Retrieved from URL.

[30] F. Pedregosa et al., "Scikit-learn: Machine learning in Python", J. Mach. Learn. Res., vol. 12, pp. 2825-2830, Oct. 2011.

[31] P. Raguraman, M. R. and M. Vijayan, "LibROSA Based Assessment Tool for Music Information Retrieval Systems," 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), San Jose, CA, USA, 2019, pp. 109-114.

[32] Documentation of LibROSA python package. Retrieved from URL.

[33] RAVDESS Dataset. Retrieved from URL.

[34] Documentation of MLPClassfier. Retrieved from URL.

[35] Documentation of sklearn. Retrieved from scikit-learn.

[36] Documentation of Soundfile in python. Retrieved from pypi.org

[37] Livingstone SR, Russo FA (2018) The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. PLoS ONE 13(5): e0196391.

[38] 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," in IEEE Access, vol. 7, pp. 117327-117345, 2019.

[39] SoX – Sound eXchange. Retrieved from URL.

[40] Losing Customer. Retrieved from URL.

[41] 6 reasons why the customer does not complain. Retrieved from URL.

[42] Kannan Venkataramanan and Haresh Rengaraj Rajamohan (2019): Emotion Recognition from Speech, arXiv:1912.10458v1 [cs.SD] 22 Dec 2019. Retrieved from URL.

Appendix A: Tools and Technologies

This section lists the tools, technologies, and libraries that were used in the project. Individual algorithms are not listed but the overall libraries are listed.

Language	Python 3.7.6
Libraries	Librosa, Scikit-Learn, SoundFile, Matplotlib, NumPy, PrettyTable, Anaconda
Environment	Jupyter Lab

Appendix B: Abbreviations

This section lists all the abbreviations used in this report work.

SER	Speech Emotion Recognition
MLP	Multi-layer Perceptron
MFCC	Mel Frequency Cepstral Coefficient

MEL	Mel Spectrogram Frequency
RAVDESS	Ryerson Audio-Visual Database of Emotional Speech and Song
LSTM	Long Short Term Memory
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
FFNN	Feed-Forward Neural Network
ANN	Artificial Neural Network
LSTM-RNN	Long Short-Term Memory Based Recurrent Neural Network
DNN	Deep Neural Network
LVQ	Learning Vector Quantization
GMM	Gaussian Mixture Model
NLMS	Normalized Least Mean Square
LMS	Least Mean Square
RLS	Recursive Least Square