School of Information Technology and
Engineering at the
ADA University

School of Engineering and
Applied Science at the
George Washington University

A NON-LINGUISTIC SPEECH EMOTION RECOGNITION

A Thesis
Presented to the Graduate Program of Computer Science and Data Analytics
of the School of Information Technology and Engineering
ADA University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Computer Science and Data Analytics
ADA University

By
Mustafa Aslanov

April, 2022

THESIS ACCEPTANCE

This Thesis by: Mustafa Aslanov

Entitled: *A Non-Linguistic Speech Emotion Recognition*

has been approved as meeting the requirement for the Degree of Master of Science in Computer Science and Data Analytics of the School of Information Technology and Engineering, ADA University.

Approved:

Abzatdin Adamov — 28.04.2022
(Adviser) — (Date)

Abzatdin Adamov — 28.04.2022
(Program Director) — (Date)

Sencer Yeralan — 28.04.2022
(Dean) — (Date)

ABSTRACT

In computer-human interaction, understanding human behavior is one of the main tasks for a machine to make the whole process as natural as possible. Conversation is one of the quickest and most natural methods of communication between humans. By understanding the emotion behind the speech, machines can better assist humans. The purpose of this project is to build a machine learning model that will recognize emotion in the audio-containing speech. The modeling of the SER can be done respective and irrespective of the semantic contents of the speech itself. This research will aim to focus on tone and pitch recognition of the voice. Which will help in recognizing the emotional state of the human using the non-linguistic aspect of the speech. As human emotions can get complicated to the point that even humans cannot always precisely understand the emotion of the speaker, we can summarize them into four main categories neutral, positive, angry, and sad emotions. These will be the focus of the predictions for the developed model. Comparison will be done between various implementations to find out the best fitting model for the non-linguistic approach.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Explanation
SER	Speech Emotion Recognition
MFCC	Mel-Frequency Cepstral Coefficients
LSP	Linear Spectral Pair
BoW	Bag Of Words
BoVW	Bag Of Visual Words
STFT	Short-Time Fourier Transform
SAVEE	Surrey Audio-Visual Expressed Emotion
GEW	Geneva Emotion Wheel
EMO-DB	Emotional Database
eGeMaps	Geneva Minimalistic Acoustic Parameter Set
HMM	Hidden Markov Model
SVM	Support Vector Machine
DBN	Deep Belief Networks
DNN	Deep Neural Network
IEMOCAP	Interactive Emotional Dyadic Motion Capture
BLSTM	Bidirectional Long Short-Term Memory Network
BLSTMATT	Bidirectional Long Short-Term Memory Network With Attention
AM	Attention Mechanism
NLP	Natural Language Processing
MLP	Multi-Layer Perceptron
SFFS	Sequential Floating Forward Selection
ERB	Equivalent Rectangular Bandwidth
LP filter	Low-Pass Filter
STDP	Spike Time-Dependent Plasticity
PCA	Principal Component Analysis
LDA	Linear Discriminant Analysis
FLAC	Free Lossless Audio Codec
CD	Compact Disc
DVD	Digital Video Disc
DAT	Digital Audio Tape
Hz	Hertz
RAVDESS	Ryerson Audio-Visual Database Of Emotional Speech And Song
AAC	Advanced Audio Coding
TESS	Toronto Emotional Speech Set
API	Application Programming Interface

FFmpeg	Fast Forward Moving Picture Experts Group
SUSAS	Speech Under Simulated And Actual Stress
LPC	Linear Predictor Coefficients
OSALPC	One-Sided Autocorrelation Linear Predictor Coefficients
STCM	Short Time Coherence Method
LSMYWE	Least-Squares Modified Yule-Walker Equations
CART	Classification And Regression Tree
RF	Random Forest
GBDT	Gradient Boosted Decision Tree
NB	Naïve Bayes
RFE	Recursive Feature Elimination
ANOVA	Analysis Of Variance
SFM	Select From Model
SFS	Sequential Feature Selector
SMOTE	Synthetic Minority Oversampling Technique
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
HDF	Hierarchical Data Format
ReLU	Rectified Linear Unit
CNN	Convolutional Neural Network
SVC	Support Vector Classifier

1 INTRODUCTION

There has been significant progress in the speech recognition field such as speaker verification, text-to-speech synthesis and vice-versa, etc. However, there is one thing that still differs machines from humans in the field of speech, which is human emotions. It is yet still not a completely developed area of research and technologies still cannot accurately detect human emotions from speech, or generate an emotional (emotionally expressive) speech themselves.

There are various ways of measuring a human's emotional state. This can be done through visual cues like facial expressions, repetitive movements of hands, bodily fluids like sweat (fear), or tears (sadness). Such cues are recorded by cameras or body sensors. At the same time, emotion can be evaluated by audial cues. By tone and pitch, and shakiness of the voice one can assume the emotional state of a human. This is usually supported by the context of the speech.

There are many spoken languages in the world, but often they all have similar intonation of emotions, i.e., when angry, a person's voice is usually high and shakier, when sad it is quieter. That is why, to create a universal system that can understand the emotional state of speech regardless of language or context, the tone and pitchiness of the voice should be prioritized. This way it is likely that machines will understand human emotion despite not understanding what they say.

Humans often can easily understand the emotions of the speaker without needing to concentrate much. The voice of a human often reflects underlying emotion through tone and pitch. Sometimes animals like dogs can understand human emotion cause of the tone [1]. However, this is incredibly hard for a machine. Multiple audio recordings are needed for a computer to differentiate various tones and associate them with basic emotions. Understanding the emotion of the speaker helps in exploiting several applications in human-computer interactions, like computer tutorial applications, where the emotional state of the human plays its role [2]. And even assist human-human interactions by automating the process. Such fields include therapeutical fields, where the emotional state of a patient is diagnosed by a tool and can be of help to a therapist [3].

While it is fairly easier to find linguistic patterns in a language using speech recognition systems and automatic the process, it is still a challenging problem for non-linguistic content. There are several factors such as cultural differences that can affect the style of emotional expression. Not only outside factors like geographical location can affect it, but the various speaker and speaker styles are existing, which may also affect the recognition [4].

Another important detail for machines to recognize emotion is unobtrusive data collection. According to [5], microphones, namely recording the audio data is the least obtrusive method to interfere with humans' daily lifestyle, while cameras point out to humans or body sensors are being constantly in contact with skin. Such methods can make humans feel annoyed or maybe too focused on the process of being watched which makes them lose the point of natural behavior.

One of the issues for such training of the models is data. While there are publicly available datasets, they are usually a collection of recorded actors' "performing" an emotion. That is, those emotions are rather fabricated than naturally expressed. The actor's performance of a certain emotion can be more dramatic compared to the real-life version of it [6].

The data in such recognition tasks have more value compared to the input features and parametrization [7]. That is, the more 'natural' the database is used in the training process the higher the degree of the performance of the system in practical conditions [8]. In real life, the speeches are often a conversational matter, while in the pre-recorded audio files it is often one-way conversing. Therefore, this may affect recognition in real-world applications.

But by using the realistic 'natural data' there come its issues. One of the common ones is the presence of noise in the recorded files. This severely influences and reduces the performance of emotion detection [9].

Another commonly observed issue with the speech in real life is that it is often neutral. This means there is a great data imbalance being observed. The researchers in [8] have manually analyzed 10 hours of data where they have found out that ~62% of it contains neutral emotion.

SER is not a concept in Data Science projects, though it has never been used with the Azerbaijani language. An entirely new dataset needs to be obtained and worked on. A great deal of feature engineering is expected to be done considering the novelty of this project. While the purpose of this project is not to focus on the language and context of the speech, a collection of the Azerbaijani data will be attempted to test the theory and differences between the multinational tonality uniqueness.

Features used for the training of the model could be classified into two groups prosodic and acoustic. Prosodic features show more of the perceptual side of the speech like rhythm, intonation, and loudness. Examples would be phone duration, fundamental frequency, and energy [10]. While the acoustic features are the ones that this paper will be focusing on. They show the physical sides of the speech like MFCC (Mel-Frequency Cepstral Coefficients), LSP (Linear Spectral Pair), etc. Though, for both cases, global statistics are used to minimize the linguistic dependency of the speech. They include meaning, variance, range, min/max, etc. This shows us that there various speech databases that are collected using different labeling methodologies.

There are various emotions that humans can express throughout a speech or physical mannerisms. In research provided by Alan S. Cowen and Dacher Keltner from the Department of Psychology of the University of California, there are 27 distinct categories of emotions present in the human brain [11]. But for the recognition of emotions, more simplistic models are better to use. There is a model of basic emotions provided by Ekman, who states that there is a set of 6 basic emotions [12]. Those are neutral expression, anger, fear, surprise, joy, and sadness emotions. At the same time, there are more detailed models like the Russel model [13]. Russel's model relies on the dimensionality of the emotions, namely that emotions can be shown in two-dimensional space. In such space, the x-axis shows valence (the capacity of one person to react to another [14]) and the y-axis shows arousal. Another multidimensional model is the Plutchik model [15]. In his model, emotions can be expressed in three-dimensional space, combining the fundamental emotions with the emotions of the bi-dimensional model. Essentially, the outer emotions are a combination of the inner emotions.

But even for humans, so many emotions aren't always easy to distinguish without external clues like mimics, environment, etc. That is why in this research I will be focusing on four main emotion categories neutral, positive, angry, and sad. The neutral emotions will include calm, boredom, and neutral speeches, the positive ones will include happy, pleasantly surprised, and hopeful speeches, angry ones will be comprised of angry and disgusted speeches, and lastly, sad ones will be comprised of sad and frustrated speeches.

2 LITERATURE REVIEW

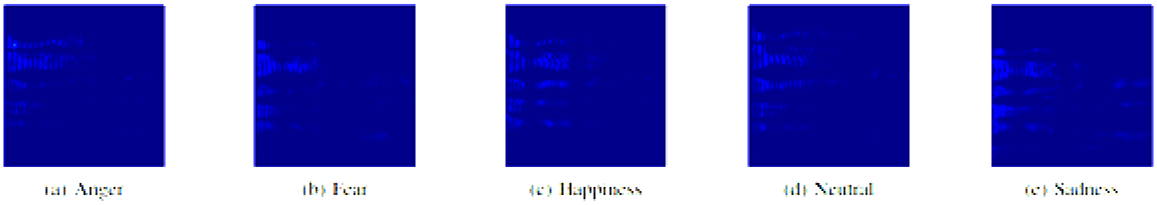
This research will be focusing on extracting the spectral features of the audio. This is a common approach for the non-linguistic analysis of the speech [16]. There are various reasons why speech emotion recognition as a task could be challenging. To begin with, there is no universally accepted theoretical definition of emotion [52].

Certain emotions can be expressed differently from person to person, and such factors as culture and environment can be affecting them. Often in the practice, the research done on monolingual speech was making assumptions that such differences as a culture do not affect the classification result. Though, the multi-lingual speech emotion classification has been done on a context-independent basis in [53].

One of the factors that affect the emotional speech variety is the mental health of the speaker, i.e. the speaker is going through a sad phase in life which erases the line between certain emotional expressions making some certain emotions more transient. This will make a machine confused, which may focus on the long-term emotional stage or the transient one [4].

Emotions do not have a certain structure, meaning in a single speech piece, there could be more than one emotion present. Finding the boundaries between them is not a difficult task for humans, but for machines, it is a tough assignment. And because humans can do it, researchers keep analyzing and finding ways to help machines learn how to do it as well.

The researchers in [5] decided that they will try a different approach where they will be transforming an audio segment into a spectrogram. Spectrograms are visual representations of the before-mentioned spectral features (frequencies) of the audio. Essentially, they have tried to approach the problem from a computer vision perspective. The algorithm they have used is called Bag of Words or in this specific case Bag of Visual Words [17]. Bag of Words is a process of describing a text document as a histogram based on the word frequencies. BoVW on the other hand is a model built on visual vocabulary that helps with computer vision concepts like image classification or object recognition [18]. The dataset they used included speeches in various languages that are open datasets, as well as manually collected recordings of the middle-school students in a class environment. They generated the spectrograms by segmenting the audio and then applying a short-time Fourier transform (STFT) on the original signal. And at the end of the evaluation, they have got spectrograms of five emotions (Fig. 1).



(a) Anger (b) Fear (c) Happiness (d) Neutral (e) Sadness

Figure 1: Example of spectrogram images per emotion

To put shortly the process, they have gone through is following: speech signals were transformed into the spectrograms, then from spectrograms descriptor was extracted and compared with the visual vocabulary. Using the feature vectors, they have finalized the process of emotion recognition. The accuracies varied depending on the dataset the model worked on; the SAVEE dataset (an emotional English dataset with four actors performing 15 sentences) showed an accuracy of 54%, while their own manually collected data showed an accuracy of 83%.

According to [8], the realism of the data used for the training has more importance on the predictions compared to the feature engineering of the models. Therefore, they have decided to use the data they have collected that is used in practical settings. The data they have annotated using WaveSurfer has been based on the Geneva Emotion Wheel (GEW) where the selected speech is labeled by 3 top emotions and then marked with a confidence level between 1-5 (Fig. 2) [19]. The data they focused on is the continuous conversational data, which they believe is the most natural setting, which has been segmented and trained on. By the end of their analysis, they found out that the top 3 emotions in the data they have analyzed are neutral, pride and elation. If categorized into three main emotions, the most common emotion is neutral, then positive (pleasant), and lastly negative (unpleasant) emotion.

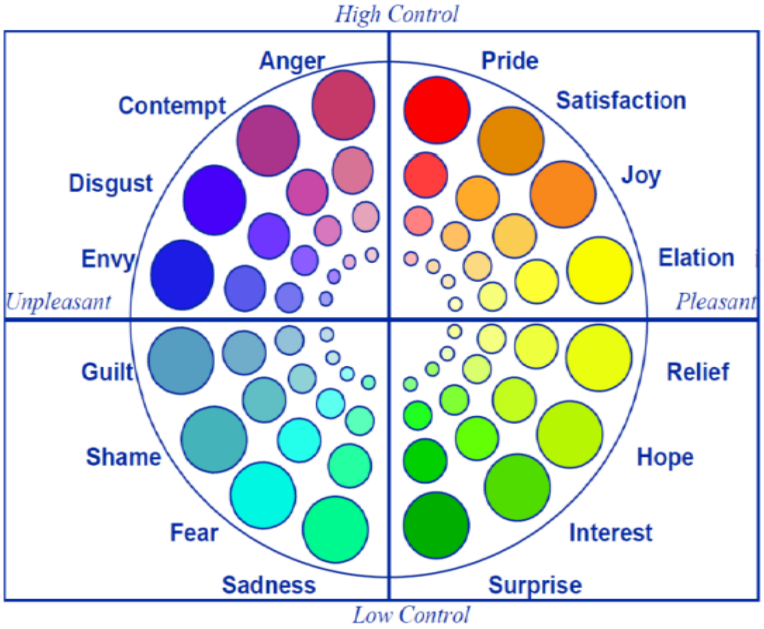


Figure 2: Geneva Emotion Wheel

It is often that recognition of emotions is used as a classification problem. Researchers at [10] decided to approach the problem from a clustering side. Specifically, they applied fuzzy clustering analysis to the speech signals. Fuzzy clustering (also known as soft clustering) is a special type of K-means clustering where a point can belong to more than one cluster. They believe that using the fuzzy clustering method will allow them to detect emotions from a speech and analyze the purity of that emotion at the same time. The so-called sum of the "memberships" of each point in the cluster shows the purity or the confidence level of the recognized emotion.

The scheme of emotion recognition they have used is mentioned in [20] research where they have used a deep belief network (Fig. 3). In this scheme the voice is transformed into a digital signal using signal acquisition, then the feature is extracted, and then emotion is recognized.

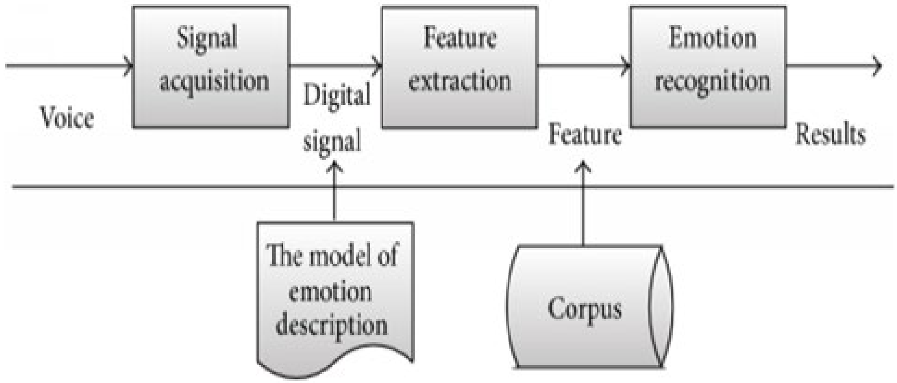


Figure 3: Deep belief network

They grouped the emotions in the data they have used, EMO-DB, based on the arousal and valence level. Essentially, those emotions that were symmetrical within the axis lines were grouped together (Fig. 4). With that, they had four clusters in total:

1. the anger and happiness/joy cluster.
2. fear and disgust cluster,
3. sadness cluster,
4. and boredom and neutral cluster.

And thus, in the end, the result they obtained contained mostly neutral emotions like in the previous research.

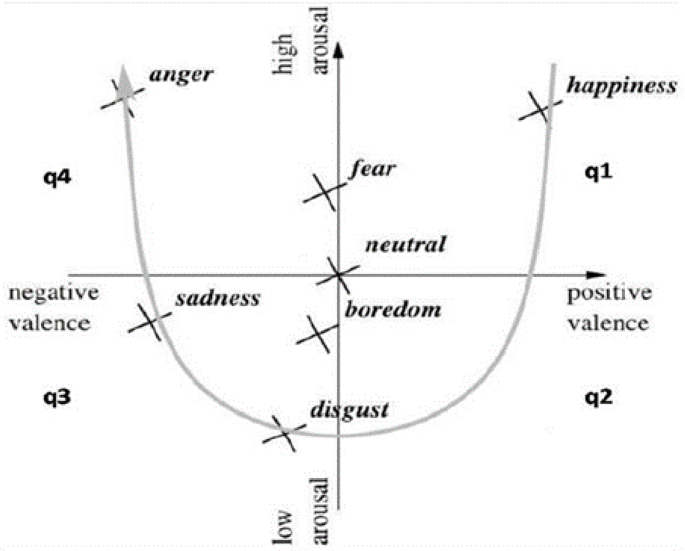


Figure 4: Emotions based on arousal and valence level

According to the research done in [21] and [22], the contrast between vowels and consonants is the most essential in speech sounds. These sounds also contain sociolinguistic information, and past phonetics and psychology research has debated whether vowel or consonant sounds are more important in determining the underlying mood. However, the majority of current studies indicate that vowels are crucial [23, 24]. Vowels have a wider range of formant patterns, making them richer in an auditory setting [21]. Two computational SER models were analyzed in the [25] study in light of hypothesized phonetic, linguistic, and psychological assertions concerning acoustics cues for speech emotion recognition in humans. It is demonstrated that the suggested networks' attention weights are heavily skewed toward vowel sounds, and that word significance is imposed depending on preceding/following acoustic context and prosody. It is also demonstrated that, as previously hypothesized, smaller auditory settings are critical in conveying emotions. Yet, according to [26] Opensmile, eGeMaps, MFCCs, and filterbanks are the most common features in practice. Hidden Markov models (HMMs), support vector machines (SVMs), deep belief networks (DBNs), and deep neural networks (DNNs) all employ these properties (DNNs). The dataset they have used is IEMOCAP [6]. The corpus contains nearly 12 hours of speech from ten individuals (five men and five women) with five dyadic (two-person) sessions, which are either scripted or improvised to elicit emotions. And a total of four categories can be classified: joyful, angry, neutral and sad. BLSTM (a bidirectional long short-term memory network [25]) with attention (BLSTMATT) system is used for the training of the model and the resulting accuracy was up to 80.1%.

An attention mechanism (AM) in deep learning, as described in the preceding paragraph, might be regarded as another milestone in sequential data processing. The goal of AM is to pick important information while filtering out extraneous data, similar to how human visual attention works. The attention mechanism, which was initially proposed for a machine translation job [28], has since evolved into a crucial component of neural networks. Even for extended sequences, including AM into encoder–decoder-based neural networks greatly improved machine translation performance [28, 29]. Many researchers have recognized attention as a crucial component of neural networks for a surprisingly wide variety of applications, including natural language processing (NLP) and voice processing, motivated by the effectiveness of attention on machine translation. The SER research community is likewise interested in integrating AM into NN architecture since emotional salient information is unevenly distributed across speech utterances.

References [30-32] compare the accuracy of speaker-based speech emotion categorization and the time it takes to build a model using Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP) classifiers. The WEKA unit was used to classify the data, while PRAAT was used to extract the features. To compare the mentioned classifiers, a simple SER module structure was adopted. The effectiveness of supervised learning methods is evaluated using the confusion matrix, classification precision, and build time. SVM's preparation was faster than MLP's, although MLP outperformed SVM in overall emotion classification. Acceptance rates for MLP and SVM were 78.69 and 76.82, respectively. In MLP, the highest emotion recognition was for anger (87.4%), with disgust and fear being the most confusing feelings, whereas, in SVM, the highest emotion recognition was for despair (89%), with pleasure and anxiety being the most baffling emotions.

Wavelet packet approaches were utilized to identify voice emotion in the studies [33-36]. The wavelet packet approach is an extension of wavelet decomposition that allows for a more detailed examination of signals. Wavelet packet atoms are waveforms that are indexed by three attributes that are naturally interpreted: location, scale (as in wavelet decomposition), and frequency. The wavelet packet coefficients were evaluated, processed, and used as inputs to Support Vector Machine (SVM) classifiers at five decomposition phases. The results revealed that applying these features to seven emotional states in two languages, German and Chinese, boosted efficiency by 4.5 percent and 16.9 percent, respectively, when compared to a single wavelet packet method without these features. The ultimate success rates for these two datasets are 61.9 percent and 62.2 percent, respectively. As a consequence, wavelet packet coefficient features outperform Mel-frequency Cepstral Coefficient (MFCC) features, according to the findings.

A speaker-independent technique for identifying emotional vocal sounds is proposed in research [37-39]. The procedure split the process of emotion recognition into two halves. The first phase comprises a coarse encoding and grouping of six emotional states to decide which pair of feelings is most likely. At the same time, low-level encoding approaches were recommended, and the retrieved characteristics were merged to provide the best emotional state descriptive acoustic vectors. Second, contemporary encoding methodologies were employed to produce a unique set of acoustic properties for each pair of emotions that may be used to distinguish them using the Sequential Floating Forward Selection (SFFS) algorithm. In the end, 72 high-level acoustic features in total were found.

In [40] they have approached the problem from a biological perspective. The input voice signal was split into two orthogonal and complementary components in the preprocessing stage, which were then altered and perceptually molded according to the features of the human cochlea. An LP analysis was done on a frame-by-frame basis. A 77-channel gammatone filterbank with ERB scaling was used to calculate and deconstruct the prediction residual. This served as the initial reservoir's input. The frequency response of each all-pole LP filter was also calculated in parallel to reveal the speech signal's formant structure. This frequency response was similarly formed using the same ERB scaling and served as the second reservoir's input. When it came to the lower auditory nuclei, according to [41] the cortex has a tonotopic structure. Closer frequency channels are processed by closer clusters of neurons, according to this arrangement. The neurons are placed in 3D configurations in the reservoirs, with each reservoir holding 77 layers of 3×3 neurons. Only one of the 77 input channels, in a sequence of increasing frequency, activated each layer of neurons. Closer connections were preferred, with the chance of a connection between neuron n1 and neuron n2 varying depending on the distance D(n1, n2). The integrate-and-fire neuron was implemented using Troyer's standard approach [42]. The Asymmetric Spike Time-Dependent Plasticity (STDP) learning rule was then utilized to modify the conductance of the synapses over the speech sample. This learning method has been shown to produce stable networks that are highly good at extracting correlations from the input [43]. The average activity of the neurons from each reservoir was subjected to Principal Component Analysis (PCA) to minimize dimensionality. PCA has the benefit of being able to reduce the output of the two reservoirs individually, as opposed to the frequently utilized ridge regression. The two PCAs' results were simply mixed. Linear Discriminant Analysis (LDA) was employed for final recognition. For the vocal tract and the source reservoirs, the greatest recognition rate of 82.35 percent was attained for 29 and 44 principal components, respectively.

3 DATA

In this section, I will be justifying the audio data format I will be using, what datasets I will include in the training process and some information about them.

3.1 WAV audio file format

The WAV format is the highest quality audio format available. The WAV format preserves the original recording's data. FLAC and AIFF are two other file formats with equal quality, however WAV files are far more common. Wave files are greater in size than newer audio file formats like MP3, which utilize lossy compression to shrink files while keeping audio quality. Larger files, on the other hand, are high-resolution audio files with a greater sampling frequency/sample rate.

The WAV, which is Waveform Audio File Format, uses RIFF specification to store audio files. RIFF is Microsoft's Resource Interchange File Format (RIFF). The bitstream is not compressed, and audio recordings with varying sampling rates and bitrates are stored in this format. It has been and continues to be one of the most widely used audio CD formats. A WAV file's header is 44 bytes long, with the following bytes being the most relevant for this project:

Table 1. Bytes in the WAV file’s header

Position of bytes	Example	Explanation
23-24	2	Number of Channels - 2-byte integer
25-28	44 100	Sample Rate (Number of Samples per second, or Hertz) - 32-byte integer. Common values are 44 100 (CD), 48 000 (DAT).

Any number of channels can be present in a WAV file. It is either one (for mono) or two (for stereo). Mono or monaural sound only requires one channel for converting a signal to sound. In contrast to mono sound, stereo sound uses several channels to turn a signal into sound. This implies that each signal put out is distinct from the others. A WAV file's data is presented as a series of frames. Each frame is made up of samples. A sample width, or the amount of bytes per sample, is present in every WAV file. The number of audio samples conveyed each second is referred to as the sampling rate (measured in Hz). For example, 48 000 Hz means there are 48 000 samples per second. The greatest audio frequency that may be reproduced is determined by the sampling rate. The maximum frequency that may be expressed theoretically is half the sampling rate (Nyquist frequency). Usually, the top frequency limit is a bit lower, therefore for a sample rate of 44 100 Hz, the realistic upper-frequency limit is a little over 20 000 Hz but less than 22 050 Hz [44].

There are two common sampling rates 44.1 kHz and 48 kHz. The sampling rate of audio CDs is 44.1 kHz (with a maximum frequency of 20 kHz), while a 48 kHz sample rate is used for the DVDs. Because 20 kHz is the greatest frequency that humans can hear, 44.1 kHz is the natural option for most audio content, but 48 kHz files contain more information that is used for the projects.

3.1.1 Python library that reads WAV files

There are multiple Python libraries that read the WAV files. One of the popular ones is probably the io.wavfile.wav package of scipy library. However, at the moment of doing this project, because there are some bugs with this package (that returns errors while reading any WAV file) I have decided to switch to the wavio library at first. It does the same work that the others do and there is no specific reason why I have decided to use it over the others.

When reading any WAV file, the output consists of data (which is an array of values), sample rate, and sample widths. An example of the output for the Fig. 5 is below:

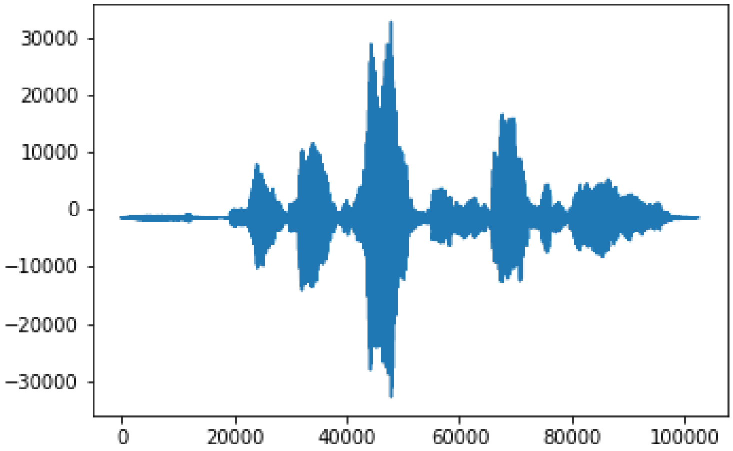


Figure 5. An example of the audio file plot from the RAVDESS dataset using matplotlib.

```
Wav(data.shape=(57311, 1), data.dtype=int16, rate=16000, sampwidth=2)

Data: [-1121. -1117. -1114. ...  -973.  -973.  -974.]

Rate: 16000
```

The reason for my further abandonment of the wavio library and switching to librosa for feature extraction and training methods was that I needed to ensure that all data points are scaled with the same logic. Librosa’s load method allows specifying what sample rate should be targeted (and not only). With the help of that librosa returns values as a floating-point time series by resampling all the values based on the sample rate. Considering that the datasets I have used included all types of sample rates (16 kHz, 22.5 kHz, 44 kHz, and 48 kHz) I have selected the highest one (48 kHz). With that, a simulation of scaling has been created. This is why I neglected the usage of StandardScaler (as explained in 4.2).

```
librosa.load(“filename.wav”, sr=48000, mono=False)

Data: [-0.00077622 -0.00093784 -0.00098862 ...  0.00449469 0.00311664 0.00146859]

Rate: 48000
```

3.2 RAVDESS Dataset

The “Ryerson Audio-Visual Database of Emotional Speech and Song”, also known as RAVDESS [45], is an open-source public dataset consisting of speech and song files.

3.2.1 Composition of the database

There are 7356 files in the RAVDESS dataset, totaling 24.8 GB in size. 24 professional actors (12 female, 12 male) took part in the data collection. With a neutral North American dialect they were requested to read phrases. There are 7 emotions present in the speech dataset: happy, angry, calm, sad, fearful, disgusted, and surprised expressions. Whereas the song contains 5 emotions: calm, happy, sad, angry, and fear. Two emotional intensity levels, normal and strong, are present in the expressions. There is also a neutral expression besides the emotionally expressed recordings, i.e. 7 emotions + 1 neutral expression.

3.2.2 Ratings of the database

On emotional validity, intensity, and authenticity, each file was assessed ten times. 247 persons who were typical of untrained adult study volunteers from North America supplied ratings. A total of 72 people participated in the test-retest study. Emotional validity, interrater reliability, and test-retest interrater reliability were all found to be quite high.

3.2.3 Modalities

All three modalities are available: audio-only, audio-video, and video-only [45]. As this research is focusing only on the speech files, audio-visual and video-only files were eliminated from the dataset. Audio-only files are composed of speech and song files. The speech file itself contains in total of 1440 recordings. Example of a plot of one of the utterances can be seen in Fig. 6.

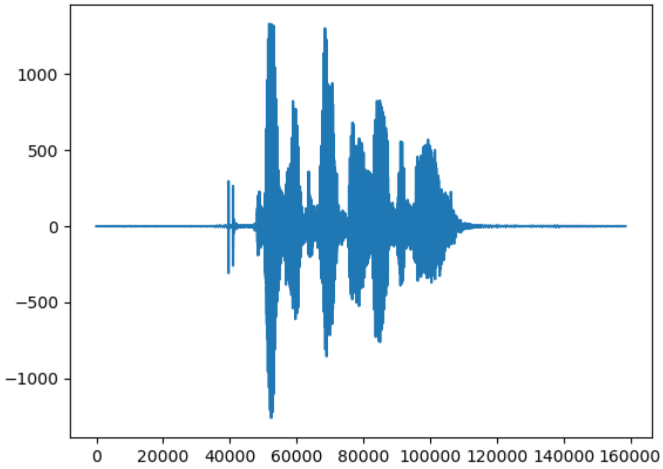


Figure 6. Wave plot of the first audio file from the RAVDESS dataset using matplotlib library of Python (03-01-01-01-01-01-01.wav)

3.2.4 File naming conventions

Each file has a unique name consisting of a 7-part numerical identifier with a dash ("-") separator (e.g., 03-01-07-02-01-02-03.wav). These numbers represent specific characteristics of each file [45]:

- Modality: 01 means the file is both audio and visual, 02 – visual only, and 03 – audio-only. Only 03's are going to be used, others are eliminated.

- Vocal channel: 01 means it is a speech file, and 02 means it is a song recording. This research only focuses on speech, therefore, 01's are the only ones being used.

- Emotion: 01 is neutral, 02 is calm, 03 is happy, 04 is sad, 05 is angry, 06 is fearful, 07 is disgust, and 08 is surprised. All emotions will stay; however, they will be grouped in the previously mentioned way: emotions will be grouped under 3 main categories: negative (sad, angry, fearful, disgust), neutral (calm and neutral), and positive (happy and surprise).

- Emotional intensity: normal intensity is represented by 01, while 02 shows strong intensity. According to [45], the neutral emotion lacks intensity level.

- Statement: There are only two (lexically related) statements used in the speech files, "Kids are talking by the door" and "Dogs are sitting by the door". Each of them is represented as 01 and 02 respectively.

- Repetition: Each vocalization is repeated twice. With this, 01 represents 1st repetition, whereas 02 - 2nd repetition.

- Actor: The last number represents the actor's number as if 01 means 1st actor, 02 – 2nd, etc. The odd numbers represent male vocalists, and the even numbers – females.

Considering the above-mentioned restrictions and focuses, the full design of the speech dataset will be in the following way:

[12(V) × 7(E) × 2(G) × 2(S) × 2(R) × 2(I) × (M)] × 2 (emotional & neutral) = 1440 recordings. Where V is Vocalist, G is Gender, S is Statement, E is Emotion, I is Intensity, R is Repetition and M is Modality. Note that, even though the emotions will be grouped in 4 categories, this doesn't reduce the variations of the files.

Therefore, an example of the naming convention 03-01-07-02-01-02-03.wav means that this file is only audio; from a speech dataset; that has disgust emotion present; with strong intensity; contains "Dogs are sitting by the door" statement; the statement is repeated 2nd time; and the 3rd actor, who is male, vocalizes it.

As I am going to be focusing on neutral (neutral/bored/calm), sad (sad/frustrated), angry (angry/disgusted), and positive (happy/pleasant surprise) emotions I am only going to use some of the RAVDESS audio files. These are going to be all files that are inside the dataset with a slight limitation on the "surprised" emotions – 08. I have decided to only use some of them as I need pleasant surprise emotions and in my personal opinion, after listening to files, only strong intensity and 2nd repetition audio files resemble them. Therefore, these naming conventions are the ones I will be extracting from the dataset:

ALGORITHM 1: RAVDESS files selection

```
SELECT    03-01-0Y-0X-0X-0X-0X.wav and 03-01-08-02-0X-02-0X.wav
WHERE     'X' stands for any digit
AND       'Y' stands for digits in range [1, 2, 3, 4, 5, 6, 7]
```

3.2.5 Statistics

After the removal of the extra files out of the 1440 samples, only 1296 samples were left, meaning 144 of them were deleted.

In a total amount of 1296 speech files, there are 288 files representing neutral emotions, 240 files representing positive emotions, 384 files representing sad emotions, and 384 files representing angry emotions. Clearly, there is some data imbalance being seen. Which needs to be fixed according to techniques. This will be dealt with in the end when all the data will be already combined.

The speech files have a sample rate of 48 kHz, a data shape minimum of 140 941 and a maximum of 253 053 (these are arrays of 1×n where n is the numbers mentioned), and a sample width of 2 (meaning they are stereo audio files). The shortest audio file is 2.94 seconds, while the longest is 5.27 seconds. Variance is approximately 0.11 seconds. Together files are 509 MB in size. It is noteworthy that these speech files are unedited. This means, that some files have pauses from each end that may affect the training. This will also be fixed later.

3.3 SAVEE Dataset

The “Surrey Audio-Visual Expressed Emotion” database was created as a prerequisite for developing an autonomous emotion identification system [46].

3.3.1 Composition of the database

The database contains recordings from four native English male speakers, who are postgraduate students and researchers at the University of Surrey (between the ages of 27 and 31). There are 480 utterances overall, with 7 different emotions (anger, contempt, fear, happiness, sorrow, surprise, and neutral). They sought to create an audio-visual database in British English that might be used to construct a multimodal emotion identification system. For each emotion, sentences were taken from the normal TIMIT corpus and phonetically balanced. Each emotion has 15 TIMIT sentences: three common, two emotion-specific, and ten generic statements that are unique to each emotion.

A Gaussian classifier was used to classify audio, visual, and combined audio-visual data. The database's relevance for study in the field of emotion recognition is demonstrated by human evaluation and machine learning trial findings. Speaker-independent identification had an accuracy of 61%, visual - 65%, and audio-visual modality of 84%. To summarize, visual characteristics outperformed auditory, and combination modalities were the most effective.

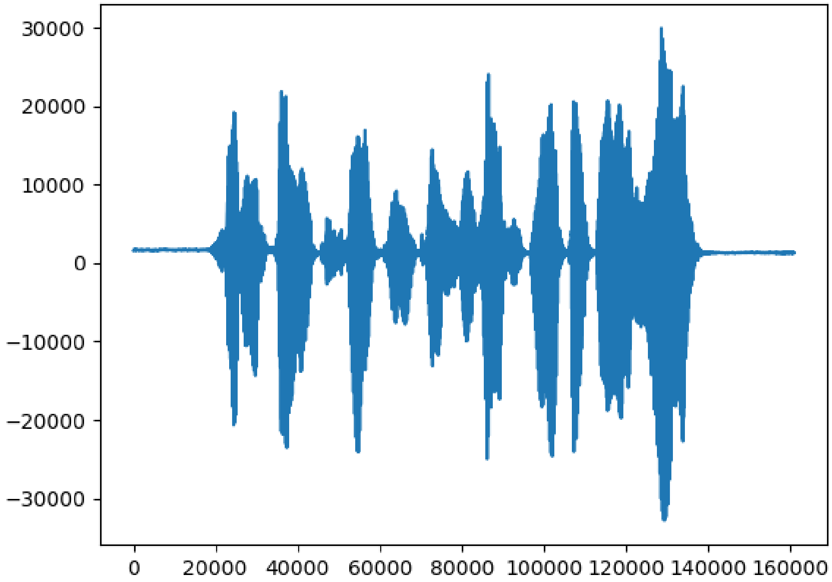


Figure 7. Wave plot of the first audio file of the SAVEE dataset using matplotlib library of Python (DC_a01.wav)

3.3.2 File naming convention

Each file’s naming starts with the speaker’s identifier proceeded with an underscore and then the shortened version of the emotion and the number of the utterance. For example, “JK_sa08.wav”. Where “JK” is the speaker, “sa” is an emotion “sadness” and “08” is “8th recording”.

Table 2. EMO-DB’s structure of the audio files

4 Speakers:	JK, DC, JE, KL
7 Emotions:	sa – sadness: 15 utterances, su – surprise: 15 utterances, d – disgust: 15 utterances, f – fear: 15 utterances, h – happiness: 15 utterances, n – neutral: 30 utterances, a – anger: 15 utterances.
An example of the emotion-sentence combination provided by the SAVEE team [46]:	Common: “She had your dark suit in greasy wash water all year.” Anger: “Who authorized the unlimited expense account?” Disgust: “Please take this dirty table cloth to the cleaners for me.” Fear: “Call an ambulance for medical assistance.” Happiness: “Those musicians harmonize marvelously.” Sadness: “The prospect of cutting back spending is an unpleasant one for any governor.” Surprise: “The carpet cleaners shampooed our oriental rug.” Neutral: “The best way to learn is to solve extra problems.”

As for data manipulation of this dataset, I will be excluding the fear utterances, as they do not belong to any category I am focusing on. Upon manual examination of the surprise speech pieces, I concluded that most of them could not be considered “pleasant surprise” emotions and therefore, decided to also neglect them from the training dataset. The rest will be used for training purposes. Thus, the naming conventions that will be used are:

ALGORITHM 2: SAVEE files selection

```
SELECT      XX_YYZZ.wav
WHERE       ‘XX’ can be any combination (speakers)
AND         ‘YY’ is in range [sa, d, h, n, a] (emotions)
AND         ‘ZZ’ can be any digit             (numbers)
```

3.3.3 Statistics

Out of the 480 audio files, only 360 were left after removing the unnecessary ones. The shortest file is 1.63 seconds, while the longest one is 7.14 seconds. On average, recordings are about 4 seconds long, with skewness being equal to ~0.46 and variance to ~1.16 seconds. This shows that such drastically long and short files are not too common.

In a total of 360 files, there are 120 files both neutral and angry emotions, and 60 of both positive and sad. These speech files have a sample rate of 44.1 kHz, a data shape average of 43500, and a sample width of 2 (stereo files).

3.4 EMO DB Dataset

EMODB is a database of emotional speech gathered in Berlin between 1997 and 1999. The anechoic chamber of the Technical University Berlin's department of Technical Acoustics was used to record the performers' utterances [93].

3.4.1 Composition of the database

Ten actors (5 females and 5 males) acted out the emotions, resulting in ten German utterances (5 short and 5 longer words) that might be utilized in normal conversation and can be interpreted in any emotion. The following 7 emotional labels (German terminology in brackets) were used to compare the results with prior investigations of their research group [47, 48, 49]: neutral (neutral), anger (Ärger), fear (Angst), joy (Freude), sadness (Trauer), disgust (Ekel) and boredom (Langeweile).

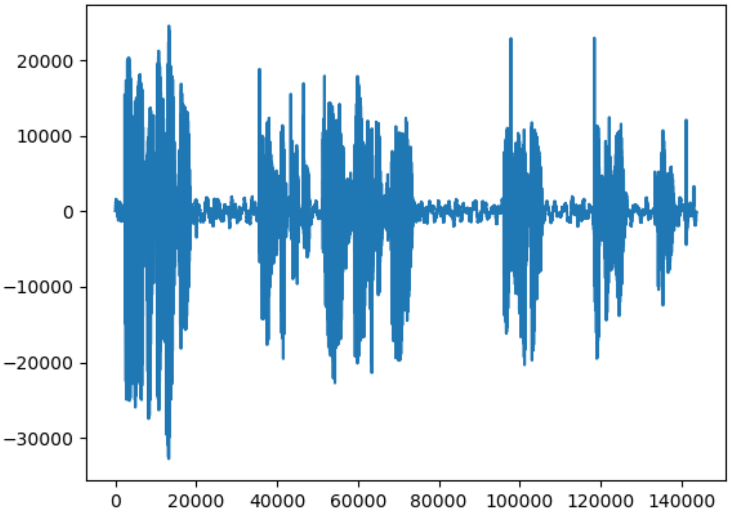


Figure 8. Wave plot of the first audio file of the EMO-DB dataset using matplotlib library of Python (08b02Tc.wav)

Two primary text categories were chosen to eliminate any bias in the emotional perception of the sentences: 1. Nonsensical writing, such as a chaotic succession of numbers or letters, or fantasy phrases, and 2. Ordinary sentences that may be employed in regular life. Although nonsensical information is guaranteed to be emotionally neutral. And according to [50], it tends to lead to stereotypical overacting. One of the goals of creating the database was to make articulatory reduction analysis easier. As a result, the test sentences' phonotactic design had to allow for many reduction forms. According to [51], the test phrases were written in such a way that all conceivable segment deletions and assimilations could be dispersed among the 10 sentences, shown in Table 3. The test sentences were to contain as many vowels as feasible to do formant analysis.

Table 3. Sentences, their codes, and English translations that are used in EMO-DB [93]

Code	Text	English translation
a01	Der Lappen liegt auf dem Eisschrank.	The tablecloth is lying on the fridge.
a02	Das will sie am Mittwoch abgeben.	She will hand it in on Wednesday.
a04	Heute abend könnte ich es ihm sagen.	Tonight I could tell him.
a05	Das schwarze Stück Papier befindet sich da oben neben dem Holzstück.	The black sheet of paper is located up there besides the piece of timber.
a07	In sieben Stunden wird es soweit sein.	In seven hours it will be.
b01	Was sind denn das für Tüten, die da unter dem Tisch stehen?	What about the bags standing there under the table?
b02	Sie haben es gerade hochgetragen und jetzt gehen sie wieder runter.	They just carried it upstairs and now they are going down again.
b03	An den Wochenenden bin ich jetzt immer nach Hause gefahren und habe Agnes besucht.	Currently at the weekends I always went home and saw Agnes.
b09	Ich will das eben wegbringen und dann mit Karl was trinken gehen.	I will just discard this and then go for a drink with Karl.
b10	Die wird auf dem Platz sein, wo wir sie immer hinlegen.	It will be in the place where we always store it.

3.4.2 File naming convention

Every speech is given a name that follows the same pattern: Positions 1-2: speaker number; positions 3-5: text code; position 6: emotion (letter stands for German emotion term); position 7: if there are more than two versions, they are numbered a, b, c two versions, they are numbered a, b, c...

For example, “08b02Tc.wav” means that this utterance was recorded by the 8th actor, saying the phrase “Sie haben es gerade hochgetragen und jetzt gehen sie wieder runter” with the emotion of sadness and this is the 3rd version of this combination.

I'm going to delete the songs that include fear feelings since they're not useful to me. Boredom and neutral-representing tracks will be grouped in the neutral category, disgust and anger will be grouped together in the anger category, while joy and sadness will be separated into their own categories:

ALGORITHM 3: EMO-DB files selection

```
SELECT      XXYYYZW.wav
WHERE       ‘X’ or ‘Y’ or ‘W’ stands for any letter/digit
AND         ‘Z’ is not  ‘A’
```

3.4.3 Statistics

After the reduction of the unneeded emotion category, out of 535 files, 69 got removed, and 466 stayed. On average, the length of the utterance is approximately 2.78 seconds with 1.23 being the shortest and 8.98 being the longest. Of 466 audios, 160 of them are neutral, 71 are positive, 62 are sad and 173 are ‘angry’ labeled. The sample rate is 16 kHz with 45000 data shapes on average. The sample width is stereo.

3.5 TESS Dataset

Toronto Emotional Speech Set was modeled on the Northwestern University Auditory Test No. 6 (NU-6; Tillman & Carhart, 1966) and put together by the University of Toronto, Psychology Department, 2010 [94].

3.5.1 Composition of the database

Two actresses (ages 26 and 64) recited a set of 200 target phrases in the carrier phrase "Say the word _____," and recordings were taken of the set depicting each of seven moods (anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral). Both actresses are native English speakers with a university education and musical background

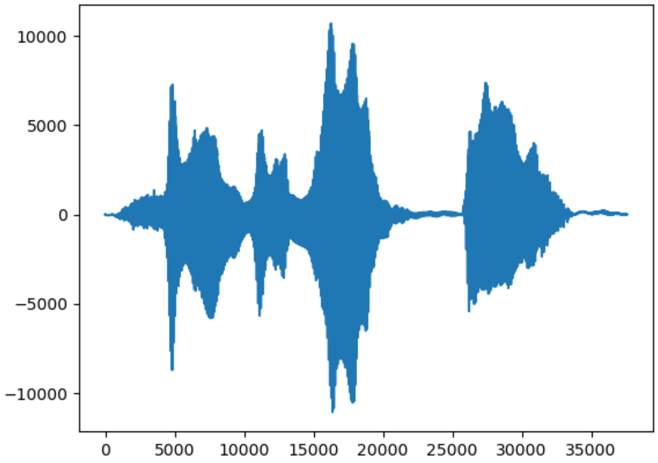


Figure 9. Wave plot of the first audio file of the TESS dataset using matplotlib library of Python (OAF_back_angry.wav)

3.5.2 File naming convention

Each file's naming convention is very straightforward and follows the structure: which actress speaks, what targeted phrase is being said, and with which emotion. For example, "OAF_back_angry.wav" represents an older actress saying the phrase "back" angrily.

Similarly, with the other datasets, I will remove the fear-representing recordings. A pleasant surprise and happy-representing tracks will be grouped in the positive category, disgust and anger will be grouped in the anger category, while neutral and sad will be separated into their categories. At the same time as this project does not focus on the age groups, both speakers' age will be neglected:

ALGORITHM 4: TESS files selection

```
SELECT      XAF_Z_W.wav
WHERE       'X' stands for any letter in ['Y', 'O']
AND         'Z' stands for any phrase in a list of 200 phrases
AND         'W' is not 'fear'
```

Out of 2800 tracks, 399 were removed as part of the fear representing recordings, and 2 files were corrupted which were also removed, leaving a total of 2399 recordings on which the models will be trained. On average the lengths of tracks are 2 seconds, the shortest being 1.25 and the longest being 2.98. The sample rate of the files is 24.4 kHz with an average data shape of 51 000 and stereo formatting. There are 400 files representing neutral emotions, 800 positive, 400 sad, and 799 anger. In total the dataset has a size of 270 MB.

3.6 All datasets combined

After all the data manipulations and statistical data collection previously mentioned, I have collected all files into one to train models on the whole data I have collected. The naming convention I chose is in the following format: "{Dataset name}_{emotion in numerical representation}_{number of the utterance in the dataset}.wav", where the numerical representation of the emotion is in the following way: 0 – neutral, 1 – positive, 2 – anger, 3 – sadness.

For example, "SAVEE_1_233.wav" means that the utterance representing positive speech is 233rd from the SAVEE dataset. While the naming convention doesn't follow the static positional regulation, Python's ability to split strings based on key character ("_" in this case) solves the issue of reading the filenames.

In total, there are 4666 utterances collected. Of them 969 are neutral, 1316 are positive, 1477 are angry and 907 are sad utterances. The shortest utterance is 1.1 seconds while the longest is 8.9 seconds. The sample width for all is the same, being stereo. However, the sample rate differs lowest being at 16 kHz and the highest at 48 kHz.

3.7 Azerbaijani Dataset

To make this project more unique compared to others I needed to add the Azerbaijani dataset to it. This was not an easy task, especially after reading how other datasets were collected.

3.7.1 Emotions

As I previously mentioned, I want to group emotions to limit them but for the sake of simplicity and easiness for the person who will record speech (I will refer to them as a subject from now on), these are the emotions I have decided to collect.

Neutral – it will contain phrases that are said in a neutral (calm or bored) fashion.

Sad – will contain phrases that are said in a sad (or frustrated) manner.

Anger – phrases that are said expressing anger (or disgust).

Happy – phrases that are said in a happy (or pleasantly surprised) manner.

3.7.2 Phrases

I thought of making phrases that are similar to RAVDESS fashion because this technique will be easier for the person compared to having natural conversations or recording just 1 word in multiple attempts. The phrases I chose are:

Phrase 1 - "Onlar qapının ağzında dayanırlar" (in English translation: They are standing at the door).

Phrase 2 - "Qonşu evin yanında danışır" (in English translation: The neighbor talks near the house).

These phrases seem to be similar-sounding but have different structures. At the same time, they seem to be phrases that could be naturally expressed in each mentioned emotion.

3.7.3 Telegram bot

In comparison with them, I lack the professional tools they have used and did not know any actors to ask them to help me. That is why I have decided to use an alternative method to collect some data. With that, I have created a telegram bot using Python's Telegram Bot API [116]. And the "actors" in my case are just any person who wished to help with collecting the dataset.

Telegram allows creating bots pretty easily, for that you need to use their main bot-creator bot @BotFather after which it will give you a token key with which you can now access your bot. The bot (@azeserbot Fig. 10) is written in the Azerbaijani language as it is targeted at the Azerbaijani audience. It starts by welcoming a user and explaining what will be expected from him, which is recording the voice of the mentioned sentence with the mentioned emotion. It also provides useful commands that can help the user to navigate through the bot with more ease. The commands are the following:

/start – generic command for each bot with which it starts the interaction with the user.

/help – this command shows a list of commands that a user may need.

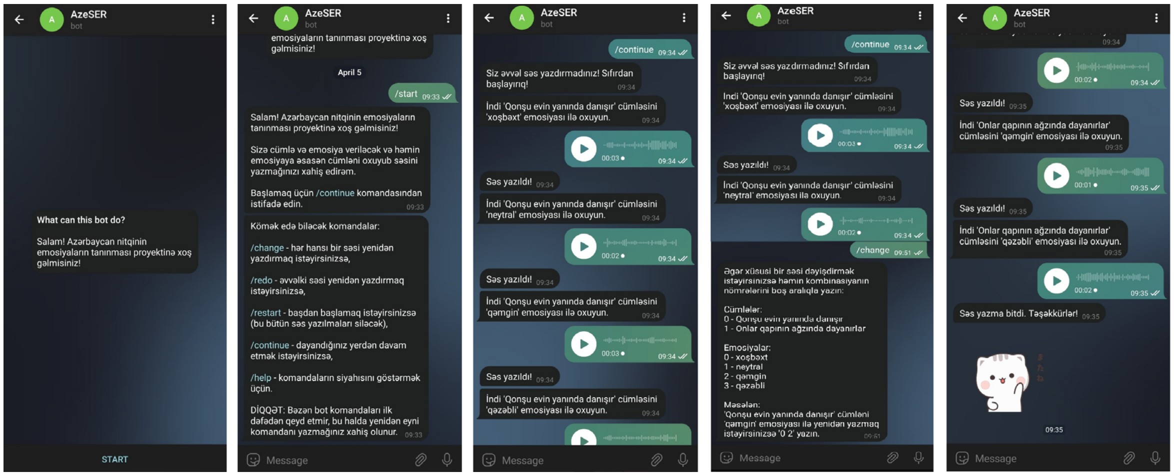
/continue – this command proceeds to the voice recording process. In case a user previously recorded some of the voices, they do not have to start from the beginning. This command will make sure they will continue from the spot they have stopped.

/change – this command allows a user to change any recording he previously recorded and felt like it was not good. After prompting the command the bot will request to write a combination of which recording the user wants to change. It will provide numberings for each sentence and emotion and ask which numbers the users want to change (i.e., 0 1 -> 0th sentence 1st emotion).

/redo – this command is very similar to the previous one with the only difference being that it will change the previously recorded voice only. So, there is not much flexibility with this command.

/restart – is a command that will delete all previous recordings and start the whole process again. To avoid accidental pressing on / typing this command, I have made a secondary confirmation message that prompts the user to confirm if they wanted to restart the process by typing 'y', otherwise they can proceed with the process again with /continue command.

In total (1+1+1+1)*2 = 8 in total recordings were expected from each subject.



What can this bot do?

Salam! Azərbaycan nitqinin emosiyaların tanınması proyektinə xoş gəlmisiniz!

emosiyaların tanınması proyektinə xoş gəlmisiniz!

April 5

/start 09:33

Salam! Azərbaycan nitqinin emosiyaların tanınması proyektinə xoş gəlmisiniz!

Sizə cümlə və emosiya veriləcək və həmin emosiyaya əsasən cümləni oxuyub səsini yazmağınızı xahiş edirəm.

Başlamaq üçün /continue komandasından istifadə edin. 09:33

Kömək edə biləcək komandalar:

/change - hər hansı bir səsi yenidən yazdırmaq istəyirsinizsə,

/redo - əvvəlki səsi yenidən yazdırmaq istəyirsinizsə,

/restart - başdan başlamaq istəyirsinizsə (bu bütün səs yazılmaları siləcək),

/continue - dayandığınız yerdən davam etmək istəyirsinizsə,

/help - komandaların siyahısını göstərmək üçün.

DİQQƏT: Bəzən bot komandaları ilk dəfədən qeyd etmir, bu halda yenidən eyni komandanı yazmağınız xahiş olunur. 09:33

/continue 09:34

Siz əvvəl səs yazdırmadınız! Sıfırdan başlayırıq! 09:34

İndi 'Qonşu evin yanında danışır' cümləsini 'xoşbəxt' emosiyası ilə oxuyun. 09:34

00:03 09:34

Səs yazıldı! 09:34

İndi 'Qonşu evin yanında danışır' cümləsini 'neytral' emosiyası ilə oxuyun. 09:34

00:02 09:34

Səs yazıldı! 09:34

İndi 'Qonşu evin yanında danışır' cümləsini 'qəmgin' emosiyası ilə oxuyun. 09:34

00:03 09:34

Səs yazıldı! 09:34

İndi 'Qonşu evin yanında danışır' cümləsini 'qəzəbli' emosiyası ilə oxuyun. 09:34

/continue 09:34

Siz əvvəl səs yazdırmadınız! Sıfırdan başlayırıq! 09:34

İndi 'Qonşu evin yanında danışır' cümləsini 'xoşbəxt' emosiyası ilə oxuyun. 09:34

00:03 09:34

Səs yazıldı! 09:34

İndi 'Qonşu evin yanında danışır' cümləsini 'neytral' emosiyası ilə oxuyun. 09:34

00:02 09:34

/change 09:51

Əgər xüsusi bir səsi dəyişdirmək istəyirsinizsə həmin kombinasiyanın nömrələrini boş aralıqla yazın:

Cümlələr:
0 - Qonşu evin yanında danışır
1 - Onlar qapının ağzında dayanırlar

Emosiyalar:
0 - xoşbəxt
1 - neytral
2 - qəmgin
3 - qəzəbli

Məsələn:
'Qonşu evin yanında danışır' cümləni 'qəmgin' emosiyası ilə yenidən yazmaq istəyirsinizsə '0 2' yazın. 09:51

09:34

00:02 09:34

Səs yazıldı! 09:35

İndi 'Onlar qapının ağzında dayanırlar' cümləsini 'qəmgin' emosiyası ilə oxuyun. 09:35

00:01 09:35

Səs yazıldı! 09:35

İndi 'Onlar qapının ağzında dayanırlar' cümləsini 'qəzəbli' emosiyası ilə oxuyun. 09:35

00:02 09:35

Səs yazma bitdi. Təşəkkürlər! 09:35

またね

09:35

Figure 10. Telegram bot with its functionalities and general look.

3.7.4 *FFmpeg*

After collecting the files, I needed to encode them into the proper .wav format. With that FFmpeg, which is a multimedia framework that supports these tasks [117], helped me. A simple code

```
subprocess.call(['ffmpeg', '-i', file, f'./AZE_reload/{file_name}'])
```

converted each of the audio files into a proper .wav format.

3.7.5 *Post-collection*

By the time of writing this paper, I only managed to collect 104 utterances, each 26 of them representing a certain emotion. Compared to the previous datasets, I assumed that trimming the audio files would not be necessary, as naturally when a person records a voice they often do not add pauses (silenced audio pieces) at the beginning or end of the recording. Nevertheless, I have done it just in case. The shortest length before the trimming was 1.67 seconds and 1.66 after trimming. And the longest length is 4.79 seconds both before and after the trimming. The sample rate for each file is 48 kHz, with stereo sample width. The average data value is 123 554. An example of an audio wave plot can be seen in Fig. 11.

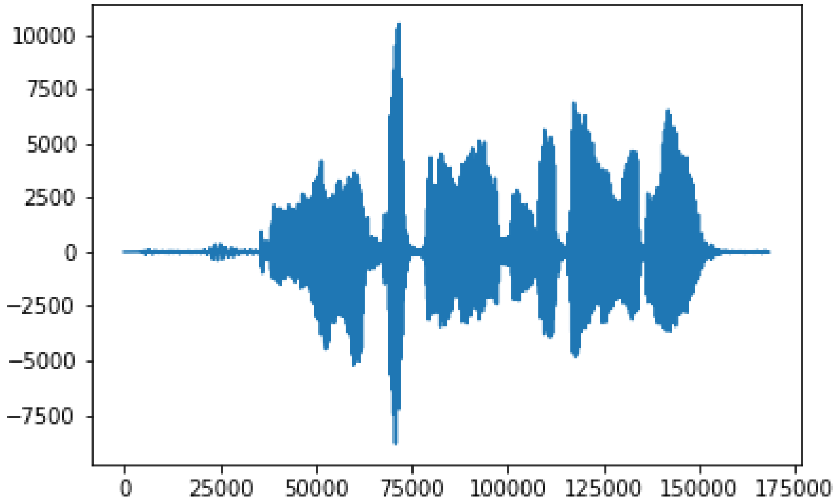


Figure 11. Wave plot of the audio file of the Azerbaijani dataset using matplotlib library of Python.

4 FEATURE MANIPULATION

In this section, I will describe the feature manipulation methods I have used throughout the project, aside from the ones already mentioned when conversing about the datasets.

4.1 Feature extraction

In pattern recognition, the used techniques are often dependent on the problem domain which provides a belief that a proper feature selection process aids the classification performance. Which is another reason why speech emotion recognition is hard. It is not very clear which speech features are the most responsible for the expression of emotion. The commonly retrieved speech parameters like pitch and energy contours are often impacted by the acoustic variability that is generated by the presence of diverse sentences, speakers, speaking styles, and speaking speeds [54]. Which itself is yet another obstacle to finding accurate features. To help with this issue researchers analyzed speech in different aspects, and one of the conclusions that came from the analysis is that an emotion can be described in two dimensions: activation and valence [55].

4.1.1 Activation

To express an emotion certain amount of energy is required. This energy is called an activation. According to [56] to express emotions like Joy, Anger, and Fear certain “sympathetic nervous system” is aroused which leads to higher blood pressure and heart rate, changes in respiratory movements, dryness of the mouth, etc. This makes the sentence sound louder, faster, and with a broader variety of pitches. In contrast, lower heart rate and blood pressure, and increased salivation produce quieter, slower, and low-pitched speeches that often represent sad emotions. With these findings, it is clear that acoustic features of the speech (namely pitch, timing, voice quality, and the articulation of the speech) have a correlation with the expressed emotion [57].

4.1.2 Valence

Though activation helps to distinguish among certain emotions, it doesn't completely help identify the concrete ones. From [56] it's clear that anger and happiness emotions have high activation energy, yet they both convey completely different effects. With that, the valence dimension comes in handy. Yet, in [58] researchers argue that there is no clear correlation of this dimension with the speech's acoustic features. This shows that while using the activation dimension it is easier to classify certain high- and low-activated emotions, it is still challenging to differentiate among those emotions.

According to [4], when it comes to feature extraction 4 issues need attention.

1. What region of the speech should be used for the extraction? Should they be divided into frames, or the whole piece instead?
2. What are the best features (pitch, energy, zero-crossing) that need to be focused on?
3. How the background noise should be dealt with? Will the additional filters for the noise reduction affect the classifier's accuracy for the better or for worse?
4. Are the acoustic features enough for this task? Or are the facial and linguistic features irreplaceable pieces for the whole process?

4.1.3 Global features vs Local features.

To answer the first question, for this task the speech signals that will be processed are required to be stationary. And since they are not in a wide sense, fragmenting speech pieces into smaller segments (frames) is a common method to approximately make them stationary [59]. The prosodic features are extracted from the segments and they are called "local features" [4]. Prosody is a branch of linguistics that studies components of speech that are not individual phonetic segments (vowels and consonants), but rather features of syllables and larger units of speech, such as intonation, stress, and rhythm. Those features include pitch and energy. But there are also global features that deal with the whole speech piece and derive information from it. According to [60, 61, 62, 63], global features are more helpful to the process of speech emotion recognition. Not only do they improve the classification accuracy and time, but there are fewer global features compared to local ones. This aids the speed of cross-validation and feature selection algorithms.

But this does not necessarily mean that global features are always efficient. According to [64], global features are only helpful in differentiating between emotions that require high activation energy (like anger, joy) and low activation energy (like sadness). The emotions within the same activation level are not distinguishable using global features. At the same time, global features lead to information loss in speech signals [4]. Which by itself makes them unsustainable for the more complex model training (like HMM, SVM). For those models, a large number of local features are more beneficial to get more reliable predictions.

In [65, 66] researchers discuss using phonemes for the feature extraction in the speech segments by focusing on different spectral shapes of one of the phonemes within various emotions. Phonemes are perceptually unique units of sound that distinguish one word from another in a given language. But this attempt only works for the vowel sounds, and the general phoneme segmentation process by itself is poor if the phonetic transcriptions are not provided. This forces the usage of linguistic features to help with the task. Researchers at [4, 59] suggest that extracting voiced segments (generated by vocal cords and oscillatory continuous parts of the speech) instead of phonemes is a much easier task to achieve.

4.1.4 Categorization of the speech features

The speech features that will be discussed in this paper are [4]:

1. continuous features
2. qualitative feat
3. spectral

4.1.4.1 Continuous features

Researchers in [67, 68, 69] claim that prosodic continuous features (pitch and energy) are the ones that show the emotional side of the fragment. The total energy, energy distribution across the frequency spectrum, and the frequency and duration of speech signal pauses are all affected by the activation energy used in the speech [56, 70, 72]. Those features are the ones that are most commonly used in emotion recognition tasks.

In the [54] research, they have used several speech features that are part of the fundamental frequency, the energy, the articulation rate, and the spectral information in voiced and unvoiced portions. These are the acoustic features that can be grouped into 5 categories [67, 73, 74]:

(1) pitch-related features;
(2) formants feature;
(3) energy-related features;
(4) timing features;
(5) articulation features.

The following are the most commonly used global features in speech emotion recognition:

- Fundamental frequency (F_0): mean, median, standard deviation, maximum, minimum, range (max-min), linear regression coefficients, 4th order Legendre parameters, vibrations, mean of the first difference, mean of the absolute of the first difference, jitter, and ratio of the sample number of the up-slope to that of the downslope of the pitch contour.
- Energy: mean, median, standard deviation, maximum, minimum, range (max-min), linear regression coefficients, shimmer, and 4th order Legendre parameters.
- Duration: speech rate, a ratio of the duration of voiced and unvoiced regions, and duration of the longest voiced speech. Formants: first and second formants, and their bandwidths.

There are also seen more complex statistics usages like parameters of the F_0-pattern generation model [77].

These features seem to be showing connection with basic emotions [54, 67, 71, 73, 75, 76, 78, 79]. And therefore, it's clear that prosodic features are correlated with the emotional representation of the speech segment.

Some emotions have similar characteristics in the fundamental F_0 frequency mentioned in [57, 59]

4.1.4.2 Voice quality features

Researchers at [67, 80, 81] believe that voice quality represents the emotional content of the given speech piece. Similar studies in [82] show the same results.

The quality of the voice is often correlated to the emotions that influence human behavior, compared to the "underlying emotions" that are influencing humans but are not controlling them [67]. According to the same research, the quality of the voice can be categorized into 4 groups:

(1) Voice level: signal amplitude, energy, and duration be reliable measures of voice level;

(2) Voice pitch;

(3) Phrase, phoneme, word, and feature boundaries;

(4) Temporal structures.

However, [4] argues that voice quality is not researched enough to correlate its role in the emotion recognition field. The arguments they provide are that the terms that are used to describe voice (i.e. tense, harsh, breathy) can have different interpretations [4, 82]. While, for example, Sherer [81] suggests that a tense voice implies anger, joy, and fear, Murray and Arnott [73] suggest that it's the breathy voice that implies those emotions.

Another issue with the quality of the voice is that differentiating the mentioned terms in the speech signal is a difficult task, especially when done automatically. The problems associated with this issue can be described in two methods.

The first method is based on the notion that the speech signal may be represented by the output of a vocal tract filter activated by a glottal source signal [59]. As a result, by eliminating the vocal tract's filtering impact and evaluating glottal signal characteristics, voice quality may be better assessed. However, because neither the glottal source signal nor the vocal tract filter is known, the glottal signal is calculated using knowledge of the source signal and the vocal tract filter's properties. As the inverse-filtering approaches are not often employed in speech emotion identification because of their intrinsic complexity [83].

The voice quality is mathematically represented in the second method by parameters determined directly from the speech signal (essentially, with no calculation of the glottal source signal). In [84] jitter and shimmer [85] were used to depict speech quality in with a continuous HMM classifier to classify utterances from the SUSAS database [86] that had angry, fast, Lombard, question, slow, and soft speaking styles. The categorization job was not reliant on the speaker, rather it was depending on the dialect. The accuracy of employing simply MFCC as features as a baseline was 65.5 percent. When the MFCC was coupled with the jitter, 68.5 percent when the MFCC was mixed with the shimmer, and 69.1 percent when the MFCC was combined with both, the classification accuracy was 68.1 percent.

4.1.4.3 Spectral based features

Spectral features are also used in speech signal representation alongside time-dependent acoustic features. In [64] it is mentioned that the emotional content of an utterance has been shown to affect the distribution of spectral energy across the speech frequency range. As mentioned previously (in the activation energy part), it has been observed that utterances expressing happiness have high energy at high-frequency ranges, but utterances expressing sadness have low energy in the same range [54, 87].

The ordinary linear predictor coefficients (LPC) [59], one-sided autocorrelation linear predictor coefficients (OSALPC) [88], short time coherence method (STCM) [89], and least-squares modified Yule-Walker equations (LSMYWE) [90] are all methods for extracting spectral properties. But because the human perception of pitch does not follow a linear scale [91], i.e., people are better at detecting subtle changes in a speech at lower frequencies, the estimated spectrum needs to be routed through a bank of band-pass filters. This will match what the human ear can hear more closely and better utilize the spectral distribution throughout the audible frequency range. The outputs of these filters are then analyzed for spectral characteristics. The bandwidths of the filters are usually evenly distributed concerning a suitable nonlinear frequency scale such as the Bark scale [103], the Mel-frequency scale [91, 92], the modified Mel-frequency scale, and the ExpoLog scale [90]. In [90], it was demonstrated that cepstral analysis-based features like LPCC, OSALPCC, and MFCC beat linear-based features like LPC and OSALPC in identifying stress in speech signals.

According to [4], Continuous parameters such as the basic frequency and pitch should be utilized to identify high-arousal vs low-arousal emotions. Spectral features such as the MFCC are the most promising features for speech representation in k-way classification. Integration of continuous and spectral characteristics is expected to improve classification performance even more. They also believe that there are some connections between the above-mentioned feature kinds, e.g., spectral variables, which are related to voice quality, while pitch contours are related to tonal patterns.

4.1.5 Conclusion of feature selections

Coming from all the above-mentioned features that can be extracted from a speech piece, at the same time realizing the limitations of python's libraries in manipulating utterances, the following features will be extracted from the dataset: MFCC, Mel, Chroma, Zero-Crossing, Spectral Roll-off, Spectral flux, and Pitch. These features will be tested to see which combination performs the best.

4.1.5.1 Mel Spectrogram

A Mel spectrogram is a spectrogram that converts frequencies to the Mel scale (Fig. 12). The Mel scale is a scale that compares the perceived frequency of a tone to the frequency that can be measured. It adjusts the frequency to fit what the human ear can hear more closely. Humans do not perceive frequencies on a linear scale, according to research. People are better at spotting differences in lower frequencies than in higher frequencies. For example, a person can easily distinguish between 300 and 800 Hz but will struggle to distinguish between 12 000 and 12 500 Hz, even though the distance between the two pairs is the same. That, in 1937, Stevens, Volkmann, and Newmann devised the Mel scale, a unit of the pitch in which equal distances in pitch sounded equally far to the listener. The following formula can be used to convert a frequency measured in Hz (f) to the Mel scale:

$$Mel(f) = 2595\, log\left(1 + {}^{f}/_{700}\right) \quad (1)$$

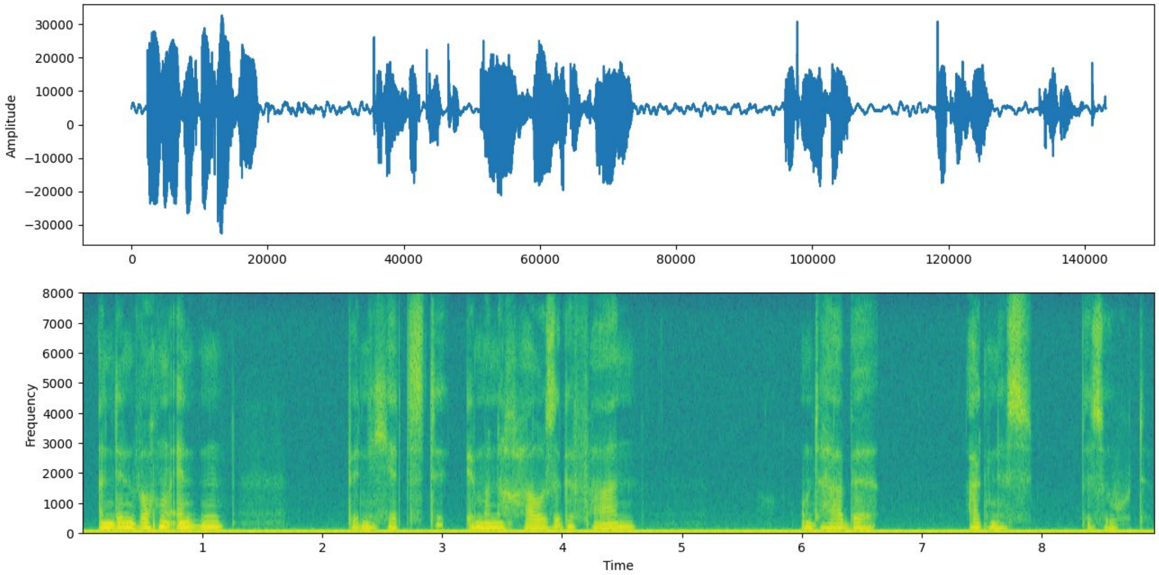


Figure 12. An example of how spectrograms look like

4.1.5.2 Pitch

Pitch can be defined as a sound's relative highness or lowness. A high-frequency sound wave correlates to a high pitch sound, whereas a low-frequency sound wave corresponds to a low pitch sound. To completely comprehend pitch, we must first understand music's pitch classes and octaves, but this side of the explanation is irrelevant for the project, therefore it will be skipped. In a speech, the pitch is one of the characteristics of a signal and is quantified as the frequency of the signal.

4.1.5.3 MFCC

Or Mel Frequency Cepstral Coefficients, indicate phonemes (different units of sound) due to the form of the vocal tract (which is responsible for sound creation). The form of a person's vocal tract determines the sound they make (including the tongue, teeth, etc). Any sound generated can be precisely described if its form can be identified appropriately. The envelope of the voice signal's temporal power spectrum depicts the vocal tract, and MFCC properly captures this envelope. The fundamental goal of MFCC is to transform time-domain signals into frequency-domain signals by utilizing Mel filters to simulate the cochlea function.

Researchers at [4] for classifying high-arousal versus low-arousal emotions, continuous features such as the fundamental frequency and the pitch should be used. For n-way classification, spectral features such as the MFCC are the most promising features for speech representation. They also believe that combining continuous and spectral features will provide even a better classification performance for the same task. It seems like there are some relationships among the feature types described above (i.e., spectral variables relate to voice quality, and the pitch contours relate to the patterns arising from different tones) but links are rarely made in the literature.

4.1.5.4 Chroma

The chroma feature is a condensed description that reflects the tonal component of (often, a musical) audio source. As a result, chroma characteristics may be thought of as a necessary precondition for high-level semantic analysis such as chord identification or harmonic similarity estimates. For chroma feature extraction, Short Time Fourier Transforms and Constant Q Transforms are utilized. Summing the log-frequency magnitude spectrum across octaves yields the chroma. The following formula shows the calculation method:

$$C_f(b) = \sum_{z=0}^{Z-1} |x(b + z\beta)| \quad (2)$$

Here X is the log-frequency spectrum, z is an octave index, Z is the number of octaves, b is the integer pitch class index and β is bins per octave.

4.1.5.5 Zero-crossing

It is represented by the number of times the amplitude of speech signals crosses through a value of zero in a particular time interval. This characteristic is a significant feature for classifying percussive sounds and has been utilized extensively in both speech recognition and music information retrieval. It's also widely utilized in a variety of other audio applications, including musical genre categorization, highlight detection, speech analysis, singing voice detection in music, and ambient sound recognition, to name a few. Analyzing the zero-crossing rate is the easiest way to discern between voiced and unvoiced speech. There is no dominant low-frequency oscillation if there are a lot of zero crossings.

4.1.5.6 Spectral-roll off

This is a metric for how skewed the power spectrum is to the right. The spectral roll-off point in the power spectrum is the percentage of bins in which 85 percent of the power is at lower frequencies. The roll-off, in other words, is the frequency at which 85 percent of the total spectral magnitude is concentrated. For right-skewed spectra, it takes on greater values, just like the centroid (Fig. 13).

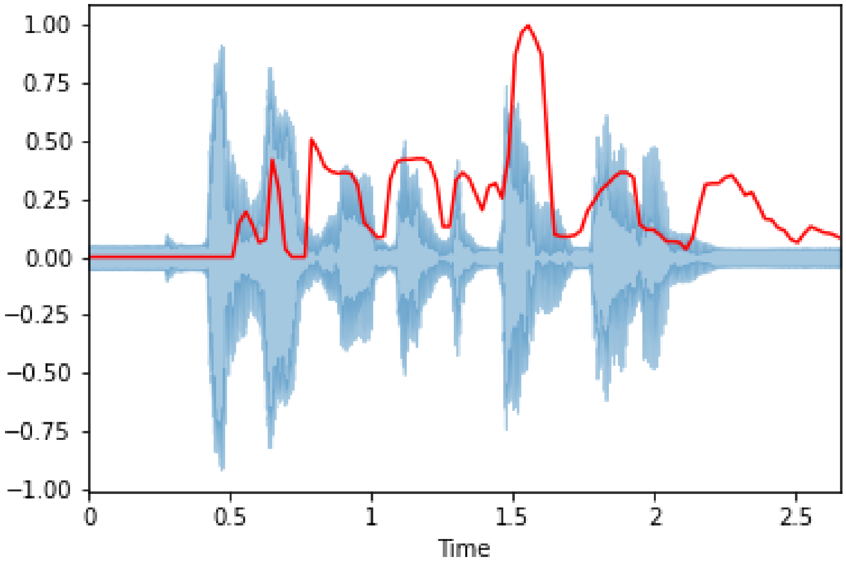


Figure 13. Spectral flux visualized with librosa library of Python

4.1.5.7 Spectral flux

Spectral flux is a measurement of how rapidly a signal's power spectrum changes. It is obtained by comparing the power spectrum of two consecutive frames and is calculated as the squared difference between the normalized magnitudes of the spectra of the two short-term windows.

$$F_r = \sum_{k=1}^{\frac{N}{2}} \left(|x_r[k]| - |x_{r_{-1}}[k]| \right)^2 \qquad (3)$$

It gives you a sense of how fast the spectral rate of change in your area is fluctuating. A large spectral flux value implies a rapid shift in spectral magnitudes and, as a result, a likely segment boundary at the r frame.

4.2 Feature scaling

When feeding data into a machine learning system, the algorithm works with numbers and has no idea what those numbers mean. This means, that if there is a significant variation in range, such as a few thousand against a few tens, the underlying assumption is that higher ranging numbers have some form of superiority. So, if the data in any circumstance comprises data points that are far apart, scaling is a strategy for bringing them closer together, or, to put it another way, scaling is used to make data points more generalized so that the distance between them is reduced. So, in machine learning algorithms, if the values of the features are closer together, the algorithm has a better chance of being trained correctly and quicker, however, if the data points or feature values are far apart, it will take longer to grasp the data and the accuracy will be lower.

Most classifiers use the distance to compute the range between two endpoints. If one of the characteristics has a large range of values, it is governed by the distance. As a result, all features' ranges should be normalized such that each contributes about equally to the final distance. This results with the following important statement: feature scaling is one of the most important phases in the pre-processing of data. A proper scaling helps building a good machine learning model.

Another rationale for using feature scaling is that some algorithms, such as neural network gradient descent, converge more quickly with it than without it. Due to the broad range of values in raw data, objective functions in some machine learning algorithms do not perform correctly without normalization. I've personally observed this happen while putting an unscaled dataset into a NN model that couldn't converge or even improve.

Normalization and Standardization are the most frequent feature scaling approaches. When one wants to limit data to a range between two integers, such as [0, 1] or [-1, 1], they utilize normalization. Standardization makes data unitless by transforming it to have a zero mean and variance of 1.

However, some algorithms do not require normalization/scaling. Those are the ones that rely on rules. They would not be affected by any monotonic transformations of the variables (scaling being a monotonic transformation). Those algorithms are tree-based (CART, Random Forests, Gradient Boosted Decision Trees, etc.) algorithms. They are relatively unaffected by the size of the characteristics. A decision tree splits a node exclusively on the basis of a single characteristic. Other features have no bearing on this feature separation. As a result, the remaining characteristics have very minimal influence on the split. This is why they are unaffected by the size of the features. Though, in my particular case in practice scaling the data slightly improved the performances of RF and GBDT.

There are certain algorithms, however, that do not require normalization or scaling. Those that rely on rules are the ones. Any monotonic transformations of the variables would have no effect on them (scaling being a monotonic transformation). These are tree-based algorithms (CART, Random Forests, Gradient Boosted Decision Trees, etc). They are relatively unaffected by the size of the characteristics. A decision tree splits a node exclusively based on a single characteristic. The decision tree divides a node based on a property that enhances the node's homogeneity. Other features have no bearing on this feature separation. As a result, the remaining characteristics have very minimal influence on the split. This is why they are unaffected by the size of the features. In my situation, though, scaling the data marginally enhanced the RF and GBDT results in practice.

The options for feature scaling are shown below [97].

1) Min Max Scaler
2) Standard Scaler
3) Max Abs Scaler
4) Robust Scaler
5) Quantile Transformer Scaler

6) Power Transformer Scaler

7) Unit Vector Scaler

Standard Scaler and MinMax are the ones I've used in practice. StandardScaler usually yielded superior results, while MinMax normalizer was essential for some models, such as Multinomial NB, that do not tolerate negative values.

4.2.1 Standard Scaler

The Standard Scaler assumes that data inside each feature is normally distributed and scales it so that the distribution is centered around 0 with a standard deviation of 1.

In Python, within the library of sci-kit learn and its preprocessing package there is a StandartScaler (`StandardScaler()`) [95] scaling tool that I have also used for the project.

4.2.2 MinMax Scaler

Minmax scales each feature to a certain range to transform it. This estimator scales and translates each feature independently such that it falls inside the training set's predefined range, such as 0 to 1. In the case that there are negative values present, the Scaler will compress the data to a range of -1 to 1.

$$\mathrm{x_{new}} = \frac{\mathrm{x} - \mathrm{x_{min}}}{\mathrm{x_{max}} - \mathrm{x_{min}}} \qquad (4)$$

The formula is the following: x_{new} is the new value, x is the original value, x_{min} is the minimum value of the column and x_{max} is the maximum value of the column. In Python, within the library of scikit learn and its preprocessing package there is a MinMaxScaler (`MinMaxScaler()`) [96] method that came in handy for scaling features.

MinMax normalizer is the one that should be used for this project, but after trying both MinMax normalizer and StandardScaler, the latter performed somewhat better. I was receiving an accuracy of 50-55 % after training RAVDESS dataset without scaling on the MLP model. The accuracy improved once the features were scaled. The accuracy of the Standard Scaler was about 67 %, whereas the accuracy of the MinMax normalizer was around 65 %. However, MinMax can normalize values between the 0 and 1 range, which is helpful with the Multinomial Naïve Bayes model as it does not accept negative values. As a result, I've kept both of them for the training.

These Scalers helped only with specific dataset training. When applying to different datasets scaler was not performing how it is supposed to because the features changed. That is why the usage of StandardScaler was abandoned for experiments by using librosa's data reading method's simulation of the scaling.

4.3 Feature selection

Sometimes features can be in an overwhelming amount which may lead to either overfitting or underfitting of the model on data. The feature selection techniques come in helpful at this point. Feature selection processes minimizes the number of input variables. With their assistance, the number of features may be restricted. Of course, the essential goal is to avoid sacrificing any predictive value. The model's performance may suffer as a result of redundant features. Even if the memory use does not impair forecast accuracy, it may be too much for a system to handle. In some models, such as self-driving automobiles, the time factor plays a crucial part in making an appropriate decision. While its sensors should be focused on the environment, little items such as a plastic bag blowing close due to severe wind gusts, or even a bird flying over the car, should not cause it to make a mistake that puts that inside at risk.

The feature selection process eliminates extraneous variables from the model, allowing it to focus on the ones that matter. This frequently leads to an improvement in the accuracy of the resultant model while reducing the number of processing resources used. The procedure determines which variables are connected and/or the goal variables, as well as which variables have low variance.

There are different approaches for selecting features. The ones I utilized for this project may be divided into two categories: brute force and statistical feature selection.

4.3.1 Brute force

The fundamental idea behind the brute force approach, also known as the exhaustive feature selection method, is to utilize every conceivable combination of features to evaluate which ones perform the best. The model's performance is tested by generating numerous combinations.

While a brute-force search is straightforward to execute and will always locate a solution if one exists, it has one major drawback. Its algorithm takes a long time to run, and the cost is proportional to the number of potential solutions (combinations). As the size of the problem grows larger, this tends to expand quite fast in many actual cases.

When the issue size is constrained or there are problem-specific heuristics that may be utilized to minimize the collection of possible solutions to a manageable size, brute-force search is often used. When ease of implementation is more essential than performance, the approach is also utilized.

When comparing various algorithms or metaheuristics, brute-force search can be used as a baseline.

As mentioned before, for this project I have selected 7 distinct audio features to use for the training purpose of the model. With them, I have decided to see if each feature helps identify the emotion of the speech by using the brute force search method. 7 features result in 126 possible combinations using the formula of combinatorics:

$$\sum_{r=1}^{n} C(n,r) = \sum_{r=1}^{n} \frac{n!}{(r!\,(n-r)!)} \qquad (5)$$

Here n is 7 (features) and r is all combinations from 1 to 7.

The dataset I have decided to test on the brute force feature selection algorithm is the RAVDESS dataset. As mentioned, the brute force increases the computational time and memory usage the bigger the data it has. Therefore, by limiting the dataset to a single one I have observed the general behavior and its effects on the performance of the training. The model for this case I have selected is

Multilayer Perceptron, which before was giving me promising results. It is also the model I have tuned the most and therefore, had high hopes for the performance results.

Generally, the algorithm I wrote showing the brute force search is:

ALGORITHM 5: Combinatorics

```
TRAIN      model with each of 126 combinations
SAVE       the top 3 highest accuracy combinations
REPEAT     the process 20 times
```

The whole training took up to an hour of training. As a result of testing an exhaustive search of features on RAVDESS with MLP, I have observed that out of 7 features 4 combinations perform the best. The combination of 6 features, namely MFCC, Mel, pitch, zero-crossing, chroma, and spectral roll-off was among the top performers. On average, it showed an accuracy of 48%. Then came combinations of chroma and zero-crossing (~46%); MFCC, Mel, and chroma (~50%); mfcc, roll-off, and pitch (~45%). As seen, between them they vary slightly not showing drastic differences in the performance. In some loops, one outperformed the other. This of course was also affected by the machine's random state-changing after each train. The mentioned combinations will be used to test all datasets together to see how they perform.

Interestingly enough, spectral flux has performed poorly in each attempt, and when even being observed, it has not shown any improvement in emotion detection. Therefore, from this point on the spectral flux feature will not be used for the training of models.

4.3.2 Statistical-based feature selection

The relationship between each input variable and the goal variable is evaluated using statistical-based feature selection, and the input variables having the strongest link with the target variable are selected. The kind of data in both the input and output variables influences the statistical measures chosen. These methods, however, can be swift and effective.

Unsupervised feature selection strategies, such as those that use correlation to eliminate redundant variables, disregard the target variable. Techniques that employ the target variable, such as approaches that eliminate unnecessary variables, are supervised feature selection techniques.

Wrapper approaches analyze many models by adding and/or removing variables to identify the best combination of predictors that optimizes model performance [98].

Although they can be computationally costly, these approaches are unconcerned with variable types. A wrapper feature selection approach like RFE is a nice example [98].

Filter feature selection approaches employ statistical techniques to assess the connection between each input variable and the target variable, with the results serving as the foundation for selecting (filtering) the input variables that will be employed in the model [98].

Finally, as part of the learning process, certain machine learning algorithms do feature selection automatically. These strategies are known as intrinsic feature selection methods. This includes techniques like Lasso's penalized regression model and decision trees and random forest.

The selection of filter features is frequently based on correlation type statistical measurements between input and output variables. As a result, the statistical measures used are heavily influenced by the data types that are changeable. The inputs and outputs can be either categorical or numerical. For my case, I have numerical input, whilst my output is categorical. Which is a common thing for a classification problem.

For such cases techniques, ANOVA correlation coefficient (linear) [99] and Kendall's rank coefficient (nonlinear) are commonly used. In the latter case, the categorical variable is assumed to be ordinal.

Now there is also an important note that some techniques do not work with Neural Network models. From the ones I will mention below, Recursive Feature Elimination (RFE), Select From Model (SFM), Sequential Feature Selector (SFS) methods are for supervised classifiers that are not neural network models like Logistic Regression, k-NN, Perceptron (one node neural network), SVM. For the case of neural network models like MLP or CNN, f_classif which performs ANOVA F-value computation in combination with the SelectKBest method performs the feature selection.

4.3.3 RFE

The process is made of recursive feature removal using feature ranking. The purpose of recursive feature elimination (RFE) is to pick features by iteratively examining smaller and smaller sets of features, given an external estimator that gives weights to features (e.g., the coefficients of a linear model). To begin, the estimator is trained on a small collection of features, and the relevance of each feature is determined using any attribute or callable. After that, the least significant characteristics are removed from the original set. This procedure is repeated recursively on the reduced set until the needed number of features to choose from is reached.

```
RFE(estimator, *, n_features_to_select=None, step=1, verbose=0,
importance_getter='auto')
```

Recursive Feature Elimination is provided by sci-kit learn library in Python [100] where parameters are the next: an estimator is a fitted model, n_features_to_select if a specific number of features is needed to be used, importance_getter with providing weights of features either depending on coef_ or feature_importances_.

4.3.4 SFS

This Sequential Feature Selector uses a greedy technique to build a feature subset by adding (forward selection) or removing (backward selection) features. Based on the estimator's cross-validation score, this estimator finds the best feature to add or remove at each stage.

```
SequentialFeatureSelector(estimator,n_features_to_select=None,
direction='forward', cv=5, n_jobs=None)
```

Sequential Feature Selector provided by sci-kit learn library in Python [101] where parameters are the next: an estimator is an unfitted model, n_features_to_select if a specific number of features needed to be used, the direction is either forward or backward, cv is cross-validation and n_jobs is a number of jobs to work in parallel.

4.3.5 SFM

SFM is another transformer for selecting features. It is based on importance weights and is similar to the SFS; however, SFM has different parameters that can suit special cases.

```
SelectFromModel(estimator, threshold=None, prefit=False, norm_order=1, max_features=None, importance_getter='auto')
```

Select From Model provided by sci-kit learn library in Python [102] where parameters are the next: estimator can be either unfitted or fitted model in which case prefit should be selected accordingly, a threshold is used to specify if specific digit or string of importance is a minimum for the modeling, and the rest will be discarded. max_features if a specific number of maximum features needed to be used, the direction is either forward or backward, cv is cross-validation, and n_jobs is a number of jobs to work in parallel.

By testing each of the methods mentioned before with all combined datasets and Logistic Regression Classifier I have come to the decision to use SFM for non-neural network classifiers. Its performance metrics showed higher results (accuracy up to 80%) and in total it limited the number of features to 508 out of 1207.

4.3.6 SelectKBest with f_classif.

f_classif is needed to compute the ANOVA F-value for the supplied sample, and SelectKBest picks features based on the k highest scores. They are also part of the sci-kit's library [99, 103].

```
SelectKBest(score_func=<function f_classif>, *, k=10)
```

Testing this method on all datasets yielded a result of top 3 feature combinations based on k-value where k was 84, 50, and 40 giving accuracies of 86%, 85.6%, and 85.4% respectively. By comparing other performance metrics, namely precision, recall, F-1 score, and confusion matrix, k=50 turned out to be the best choice.

5 DATA MANIPULATION

5.1 Trimming of the audio files

Upon investigation of the audio files, I saw that some utterances have silenced pieces at the beginning and end of the track. This essentially means that not all files consist of pure speech. This, in turn, would give 0 values on each end of the feature data. And even though people tend to have pauses in between talking when they feel certain emotions (e.g., when expressing sadness people tend to be slower and have more pauses [119]) the one in the beginning and end did not contribute to the expression of any emotion. As while parsing and extracting features I take mean values for each data piece, I realized that those pauses may confuse the models and decided to remove them.

However, I made sure that utterances that contain pauses in between words are kept intact. For example, in Fig. 14 you can observe a slight difference in the plots and the length of the file in total was reduced by 0.6 seconds.

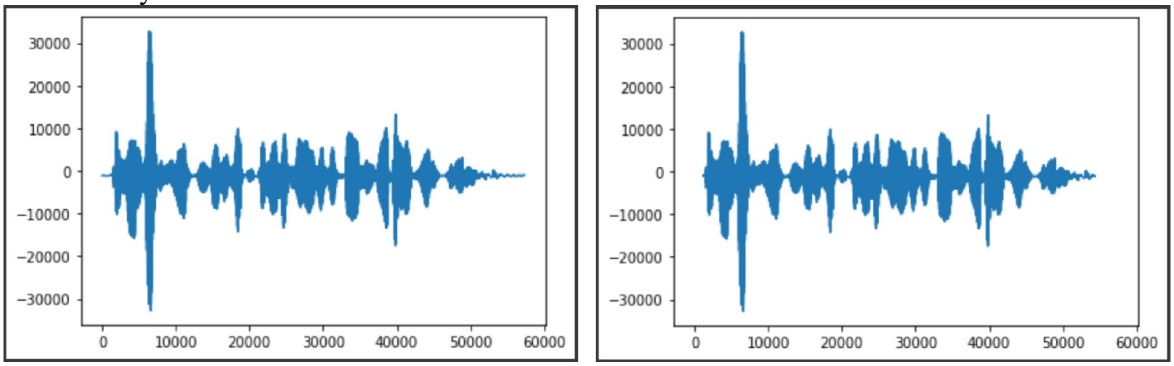


Figure 14. The before and after trimming were plotted using the matplotlib library of Python.

5.2 Data imbalance

After all manipulation with the datasets that were mentioned before, in the end, I got 969 neutral, 1316 positive, 1477 angry, and 907 sad utterances. There can be observed some data imbalances between sad and neutral utterances.

Imbalance classification has the drawback of having too few samples of the minority class for a model to learn the decision boundary adequately.

There are various methods in the machine learning field that deals with data imbalance. For example by increasing dataset size. But when it is impossible to do so, the resampling, which is undersampling and oversampling, is being used to fill in the gaps.

Undersampling is lowering the data size of the larger class by removing instances (utterances in this case). Oversampling on the other hand is increasing the data size of the smaller, under-represented class.

Considering that my dataset is not particularly big, I've decided to use the oversampling technique to fix the imbalanced data issue. The simplest approach involves duplicating examples in the minority class, before fitting a model. This can help to balance the class distribution, but it doesn't provide the model with any extra information. Synthesizing new instances from the minority class is an improvement over replicating examples from the minority class. However, this is a sort of data augmentation that works well with tabular data.

As stated, there is the main disadvantage of this, that the oversampled data is synthetic. It is referred to as the Synthetic Minority Oversampling Technique or SMOTE for short. SMOTE discovers the k closest minority class neighbors after selecting a minority class instance alpha at random. The synthetic instance is then produced by picking one of the k closest neighbors betta at random and drawing a line in the feature space between the alpha and betta. The synthetic instances are created by combining the two chosen examples alpha and betta into a convex combination [105].

The technique has the drawback of creating synthetic instances without taking into account the majority class, which might result in unclear examples if there is a lot of overlap between the classes.

This approach may be used to generate as many synthetic instances as necessary for the minority class. According to the study [105], random undersampling is used to reduce the number of instances in the majority class, and subsequently SMOTE is used to oversample the minority class to balance the class distribution.

The random undersampling is available in the imbalanced-learn library with the RandomUnderSampler class. The general algorithm is to oversample the minority class and then use random undersampling to reduce the number of examples in the majority class. With the help of Pipeline [106], it is possible to chain these transformations.

```
over = SMOTE(sampling_strategy=0.1)

under = RandomUnderSampler(sampling_strategy=0.5)

Pipeline(steps = [over, under])
```

The Pipeline may then be applied to a dataset, applying each transformation in turn and providing a final dataset that contains the combination of the transforms done to it, in this instance oversampling and undersampling.

At the end of the imbalance dealing procedure, there were 1476 samples representing each emotion.

6 MODELS AND THEIR OUTPUTS

6.1 Common processes

Before moving on to the models, I will include the processes I have added that all models share in common.

6.1.1 Performance metrics

Most of the models all include the performance metrics of the accuracy of prediction, precision, recall, and F-1 score.

6.1.1.1 Accuracy of prediction

The difference between observed and expected values is used to calculate accuracy.
Anticipated values are frequently used to refer to values that have been predicted or modeled using training data. This accuracy, on the other hand, is simply a measure of how well the model matches the training data, not of predicting accuracy. The disparities between the expected and observed

values of new samples may also be used to determine predictive accuracy (e.g., validation samples).

6.1.1.2 Precision

A good classifier should have a precision of 1 which happens only when the numerator and denominator are equal, i.e. TP = TP +FP. This also implies that FP is zero. The value of the denominator grows bigger than the numerator as FP rises, and the precision value lowers.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (6)$$

6.1.1.3 Recall

A good classifier should have a recall of 1 which happens only when the numerator and denominator are equal, i.e. TP = TP +FN, does recall become 1. This also implies that FN is zero. The value of the denominator grows bigger than the numerator as FN rises, and the recall value lowers.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (7)$$

6.1.1.4 F-1 Score

Only when precision and recall are both 1 does the F1 Score become 1. It is a better metric than accuracy since it is the harmonic mean of precision and recall.

$$\mathrm{F1} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (8)$$

6.1.2 Saving models

To avoid re-training the models that have already been trained previously, saving them into the proper formats was required. There are two file formats I used: .pickle and .h5.

An H5 file is a hierarchical data formatted file system (HDF). It contains scientific data in multidimensional arrays. Training the Convolutional Neural Network model requires saving the model in this specific format.

Pickle may be used to serialize Python object structures, which is the conversion of a memory object to a byte stream that can be saved as a binary file on a disk. All other models can be saved in this format.

6.1.3 Splitting datasets into train test split.

Within the sci-kit learn library’s model_selection package, there is a convenient tool that will split any data into training and testing segments [118].

```
x_train, x_test, y_train, y_test = train_test_split(data, labels, test_size = 0.15,
random_state = 42)
```

In the method’s parameters, one can specify what portion of the dataset will be tested and what portion will be the training subset. In my case by standard, I have decided to use 15% of the dataset as testing and 85% as training. The random state parameter helps us to always use the same subsets. This helps to monitor changes in the model's performance more accurately as now we are certain that the same data segments were used in each training and testing.

6.2 Multilayer Perceptron (MLP)

To understand MLP, one must first understand Perceptron. Frank Rosenblatt began by enhancing McCulloch and Pitt's neuron model and developed the Perceptron algorithm (which was originally designed as an image recognition system) that could learn the weights to provide an output. Rosenblatt's perceptron machine was based on the neuron, which is a fundamental unit of computing, that receives a series of pairs of inputs and weights.

The main distinction in Rosenblatt's model is that the inputs are merged in a weighted total, and the neuron fires and provides an output if the weighted sum exceeds a predetermined threshold. The activation function is represented by threshold T (image). The neuron produces the value 1 if the weighted total of the inputs is greater than zero; else, the output value is zero.

$$y = \begin{cases} 1, \text{if } \Sigma_i w_i x_i - T > 0 \\ 0, \text{otherwise} \end{cases} \quad (9)$$

Minsky and Papert demonstrated in 1969 [107] that a Perceptron with only one neuron cannot be used for non-linear data. The Multilayer Perceptron is used to overcome this constraint. Input, one (or more) hidden layers, and output with numerous neurons grouped together are what make up a Multilayer Perceptron. In a Perceptron, neurons must have an activation function that imposes a threshold (i.e., ReLU or sigmoid), however in a Multilayer Perceptron, any activation function may be used. Because inputs are integrated with starting weights in a weighted sum and applied to the activation function, the Multilayer Perceptron falls under the category of feedforward algorithms. Each linear combination, on the other hand, gets transmitted to the next layer.

MLP classifier is provided in Python within the sci-kit learn library's neural network package [108]. It has many adjustable parameters, but for the sake of confusion avoidance, I will keep only the ones I've changed from default values in my training process.

```
MLPClassifier(hidden_layer_sizes=(350,), alpha=0.001, batch_size=256, learning_rate=adaptive, max_iter=550, verbose=True, early_stopping=True , n_iter_no_change=20)
```

Setting max iterations to 550 I limited the number of epochs with which the model will train. Besides that, by specifying that the model should stop if the validation score is not improving more than the tol value (which by default is 0.000100) after consecutive 20 epochs I made sure that it does not overfit.

With these settings, the model stopped at epoch 103 (meaning it converges at epoch 83) with a validation score being 87.2% and loss being 0.064. The accuracy turned out to be 88.94% with the average precision value being 0.8530, average recall being 0.8542 and average F-1 score being 0.8681. Refer to Appendix1 for more detailed information with all results.

6.3 Convolutional Neural Networks

A Convolutional Neural Network (CNN) is a Deep Learning algorithm that can take in data, assign learnable weights and biases, and distinguish between them. The amount of pre-processing required by a ConvNet is much less than that required by other classification techniques. One of the advantages of ConvNets is that they can learn filters/characteristics with enough training.

Keras library of Python provides 1D convolutional layers for tasks that are not image related (compared to 2D and 3D) [111]. This layer creates a convolution kernel that is convolved with the layer input over a single spatial (or temporal) dimension to produce a tensor of outputs.

```
model = Sequential()

model.add(Conv1D(128, 5,padding='same', input_shape=(Xtrn.shape[1],1)))

model.add(Activation('relu'))

model.add(Dropout(0.1))

model.add(MaxPooling1D(pool_size=(8)))

model.add(Conv1D(128, 5,padding='same',))

model.add(Activation('relu'))

model.add(Dropout(0.1))

model.add(Flatten())

model.add(Dense(4))

model.add(Activation('softmax'))

lr_schedule = tf.keras.optimizers.schedules.ExponentialDecay(

    initial_learning_rate=1e-2,

    decay_steps=1000,

    decay_rate=0.9)

es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=10)

opt = tf.keras.optimizers.Adam(learning_rate=lr_schedule)

model.compile(loss='categorical_crossentropy',                      optimizer=opt,
metrics=['accuracy'])

history=model.fit(Xtrn,   Y,   batch_size=16,   epochs=100,   validation_data=(Xtst,
Ytst),callbacks=[es])
```

The model uses 2 Conv1D layers and one MaxPooling1D layer. The activation function is softmax. The learning rate is used with the decay method. And an early stopping mechanism is set to make sure that the model does not overfit. If there are no improvements in validation loss score within 10 epochs the model will stop. The batch size is set to 16 to make sure that utterances from different datasets learn together. The epochs were set to 100 but the model converged after the 39th epoch resulting in 88.49% accuracy with 0.2961 loss and validation accuracy of 84.3% with validation loss of 0.4555. The average precision is 0.89, the average recall is 0.88 and the average F-1 score is 0.88. The plot of the model's convergence can be seen in Fig. 15. And refer to Appendix1 for more detailed information with all results.

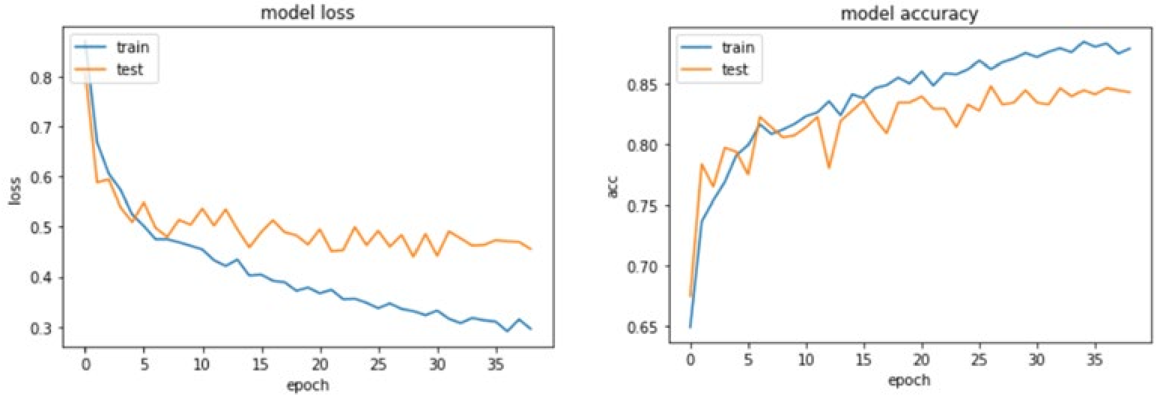


Figure 15. Model Loss and Model Accuracy plots using matplotlib

6.4 Decision Tree

For classification and regression, the Decision Tree is a non-parametric supervised learning approach. The objective is to learn basic decision rules from data attributes to develop a model that predicts the value of a target variable. A tree is an approximation of a piecewise constant. Internal nodes represent dataset features, branches represent decision rules, and each leaf node represents the result (Fig. 16). It is, in essence, a graphical representation for obtaining all feasible answers to a problem/decision based on specified circumstances.

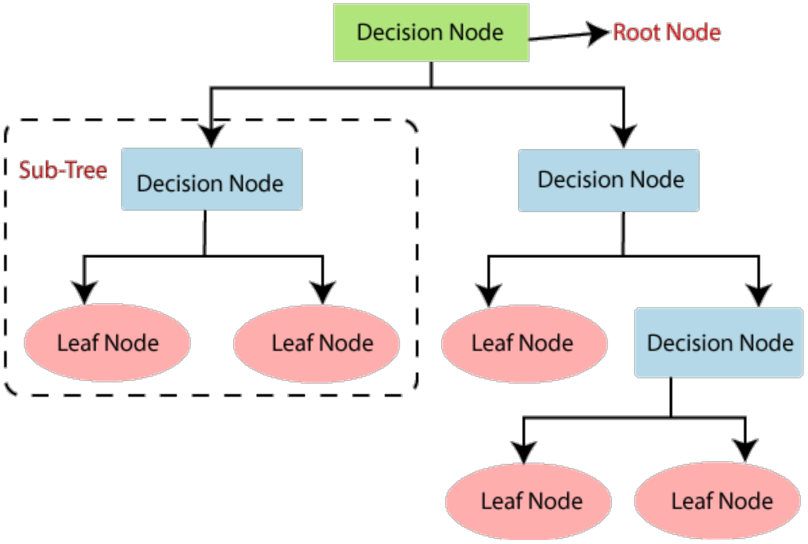


Figure 16. The depiction of the Decision Tree's structure [109]

```
DecisionTreeClassifier(criterion='gini',  splitter='best',  min_samples_split=2,
min_samples_leaf=1,  max_features=None)
```

The Decision Tree classifier can be found within the sci-kit learn library [110]. Setting splitter to best helps the accuracy by 3%. Changing other values generally did not improve the performance of the model, so it was best to keep them default. The highest accuracy was 74.27% with average recall, average precision, and average F-1 score all being 0.75. Refer to Appendix1 for more detailed information with all results.

6.5 Random Forest

As the name indicates, a random forest is made up of a huge number of individual decision trees that work together as an ensemble. Each tree in the random forest produces a class prediction, and the class with the most votes becomes the prediction of the model (Fig. 17).

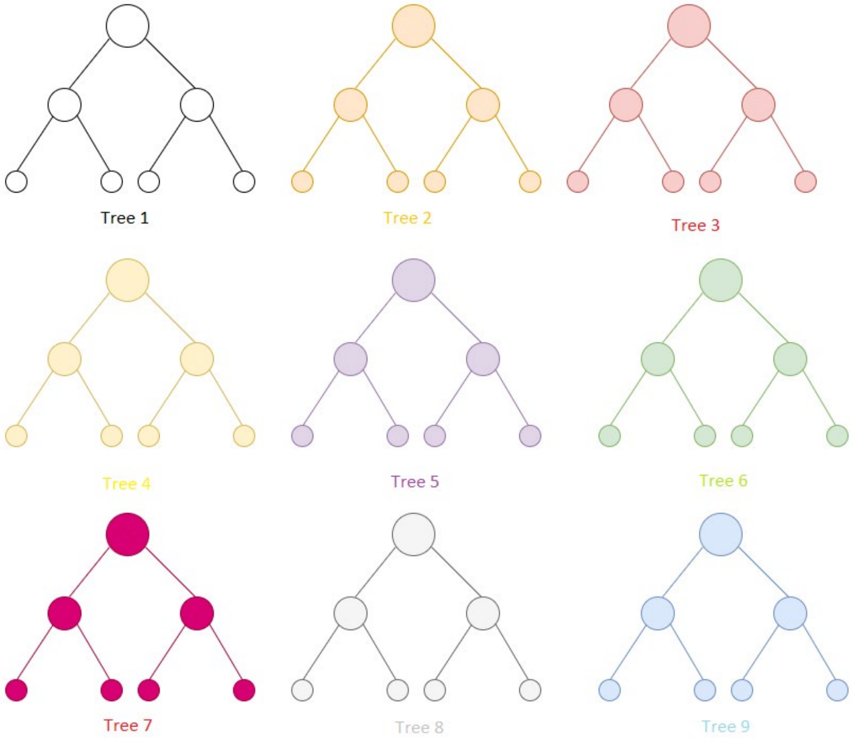


Figure 17. The depiction of the Random Forest that is made up of Decision Trees

The wisdom of crowds is the basic principle underlying random forest, and it's a simple yet effective one. The reason the random forest model works so well is that it consists of a huge number of largely uncorrelated models (trees) that work together to outperform any of the individual constituent models.

The sci-kit's ensemble package provides the RandomForestClassifier model with lots of adjustable parameters [112]. I will keep them short to the ones I have changed from default values to not confuse the reader.

```
RandomForestClassifier(n_estimators=100, max_features=None, bootstrap=True,
oob_score=True, n_jobs=-1, verbose=1)
```

Setting bootstrap to true and o0b_score to true (which can be true only if bootstrap is true) performed by 12% better from an accuracy perspective. While keeping most of the parameters to default had a better output. The highest accuracy was 88.26% with average recall being 0.8773, average precision 0.8773 and average F-1 score being 0.8757. Refer to Appendix1 for more detailed information with all results.

6.6 Gradient Boost Decision Tree

The name "Gradient Boosting" comes from Friedman's article Greedy Function Approximation: A Gradient Boosting Machine. XGBoost stands for "Extreme Gradient Boosting." XGBoost is used to solve supervised learning issues, such as predicting a target variable using training data (with many features). Gradient boosting machines merge decision trees as well, but they do it from the beginning rather than the conclusion in comparison to Random Forests. Gradient boosting combines outcomes along the way, whereas Random Forests combine findings after the process.

```
xgb.XGBClassifier(max_depth=7, eta=0.008, objective='multi:softprob',
sub_sample=0.8, booster='gbtree')
```

This method is part of the xgboost library [27]. The learning rate is specified by the eta parameter. The learner type is determined by the booster parameter. This is often a tree or a linear function. The model for trees will be made out of an ensemble of trees. The objective parameter is set to multi:softprob. Which by itself is the same as softmax, but outputs a vector of `ndata * nclass`, and then into `ndata * nclass` shaped matrix. Each data point corresponding to each class has a projected probability of the outcome. The ratio of the training instances is determined by subsample and it occurs in every boosting cycle. Setting it to 0.5 implies that before building trees, XGBoost will randomly pick half of the training data. Overfitting will be avoided as a result of this. The maximum depth of a tree is set to 7 because any higher number will make the model more complex and prone to overfitting. The highest accuracy was 84.42% with average recall being 0.8531, average precision 0.8527 and average F-1 score being 0.8516. Refer to Appendix1 for more detailed information with all results.

6.7 Linear Support Vector Classifier

The support vector machine algorithm's goal is to find a hyperplane in an N-dimensional space that categorizes data points. There are several hyperplanes with help of splitting the two kinds of data points. SVC's goal is to locate a plane with the greatest margin, or the greatest distance between data points from both classes. Maximizing the margin distance gives some reinforcement, making it easier to classify subsequent data points. SVC with kernel = 'linear' is equivalent to Linear SVC. The distinction between the two is that LinearSVC is written in liblinear, whereas SVC is written in libsvm. That's why LinearSVC gives you additional options when it comes to penalties and loss functions. It also handles a larger number of samples better. At the same time accepts both dense and sparse input, and multiclass support is implemented via a one-vs-rest (ovr) approach. The classifier method can be found in the sci-kit library [114].

```
LinearSVC(penalty='l2',  loss='squared_hinge',  tol=0.0001,  multi_class='ovr', 
verbose=1,  max_iter=10000)
```

The penalty is chosen to be l2 and l1 leads to coef_ vectors that are sparse. But considering that we will be using coef_for the feature selection this must be avoided. Linear SVC supports multiclass classification when y has more than 2 classes. According to the documentation setting the dual parameter to false is a preferred setting when the number of features is more than the number of samples. At the same time, SVC tends to require more iterations which is why the max_iter parameter was set to 10000. After the training, the model converged on the 5015th iteration with an accuracy of 77.65%. The average precision is 0.7817, the average recall is 0.7825 and the average F-1 score is 0.7810. Refer to Appendix1 for more detailed information with all results.

6.8 Multinomial Naïve Bayes

The Multinomial Naive Bayes algorithm is a common Bayesian learning method in Natural Language Processing. The Naive Bayes classifier is made up of several algorithms, all of which have one thing in common: each feature being classified is unrelated to any other feature. The presence or absence of one trait does not influence the inclusion or exclusion of another.

It is part of the sci-kit learn library's naïve Bayes package model [115]. Compared to the other models previously mentioned, it does not have as many parameters (just 3).

```
MultinomialNB(alpha=1.0, fit_prior=True, class_prior=None)
```

Specifying any of the default values ruins the performance of the model. This model is the worst performing out of all, which makes sense because it is not suited for this type of classification. The highest accuracy was 49.98% with average recall being 0.47, average precision 0.539 and average F-1 score being 0.458. Refer to the Appendix1 for more detailed information with all results

6.9 Logistic Regression

The process of finding a discrete result given input is logistic regression modeling. The outcome of the most frequent logistic regression models is binary. However, depending on many independent factors, Multinomial logistic regression may be used to predict categorical placement in or the probability of classification on a dependent variable. Multinomial logistic regression requires careful consideration of sample size and outlying case examination. The method of modeling the probability of a discrete result given an input variable is known as logistic regression. The outcome of the most frequent logistic regression models is binary. However, depending on many independent factors, Multinomial logistic regression may be used to predict categorical placement in or the probability of classification on a dependent variable. Multinomial logistic regression requires careful consideration of sample size and outlying case examination.

Within the sci-kit library's linear model package Logistic Regression exists [104]. In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme.

```
LogisticRegression(dual=False,     tol=0.005,     C=1.0,          solver='saga',
multi_class='multinomial', max_iter=100, verbose=1, n_jobs=-1)
```

The 'multi_class' option is set to 'multinomial' cause it is not a binary classification. When there are more samples than features, the dual parameter should be set to false. Setting solver to saga improved the performance a bit compared to lbfgs. The tolerance is set to 0.005 (meaning it should stop iterations after it hits this point) and max_iter to 100. With this, the model converged after the 88th epoch. The highest accuracy was 79.12% with the average recall being 0.8079, average precision of 0. 8074 and the average F-1 score being 0. 8079. Refer to the Appendix1 for more detailed information with all results

7 EXPERIMENTS WITH AZERBAIJANI DATASET

After manually analyzing the Azerbaijani dataset that I have collected through the Telegram app, I have realized that these speech pieces are not as emotionally expressive as the English or German datasets. But the main concern of mine was that this dataset being small is something that will likely affect the results as well (i.e., overfitting model). Therefore, I was expecting to have some poor classification results. With that in mind, I have decided that I should try multiple experiments on how to include this dataset in others. Some experiments showed really poor accuracies of 17-35% while one of them showed a very decent performance. The results of the experiments are below.

7.1 Experiment 1

The first experiment I have done is by training all models with the previously mentioned datasets (SAVEE, TESS, RAVDESS, and EMODB) and saving the models into a pickle file. Then, using these models predict Azerbaijani datasets. It is also worth mentioning that the validation accuracy was around 80%, which shows that there are clear issues with the Azerbaijani dataset. This resulted in poor accuracy predictions (Table 4).

Table 4. Outputs of the first experiment

Model	Accuracy	Validation Accuracy	Average Precision	Average Recall	Average F-1 Score	Confusion Matrix
Multilayer Perceptron	19.23%	86.5%	0.172	0.192	0.17	[10 4 10 2] [18 0 4 4] [15 0 5 6] [13 4 4 5]
Convolutional Neural Network	15.38%	78.8%	0.168	0.153	0.121	[2 11 0 13] [1 12 6 7] [1 19 0 6] [1 17 6 2]
Decision Tree	16.35%	80.1%	0.162	0.163	0.16	[6 7 9 4] [7 2 9 8]

						[8 0 5 13] [6 5 11 4]
Random Forest	14.42%	79%	0.095	0.144	0.11	[10 6 5 5] [22 0 1 3] [15 0 0 11] [11 2 8 5]
Gradient Boost Decision Tree	17.31%	80.3%	0.11	0.173	0.129	[12 5 5 4] [18 0 2 6] [17 0 0 9] [12 2 6 6]
Linear SVC	21.15%	78.3%	0.251	0.211	0.166	[16 6 4 0] [20 0 5 1] [23 0 2 1] [15 2 5 4]
Multinomial Naïve Bayes	22.12%	68%	0.221	0.221	0.219	[7 6 7 6] [9 4 4 9] [8 1 5 12] [4 11 4 7]
Logistic Regression	18.27%	78.9%	0.131	0.182	0.126	[15 7 2 2] [20 1 1 4] [22 0 0 4] [17 3 3 3]

7.2 Experiment 2

The next experiment was to train the models with the Azerbaijani dataset included in the whole input dataset and then test only on the Azerbaijani dataset. This resulted still in the best performance out of all experiments. Because in this case, models were more "aware" of the less expressive emotional speeches and at the same time Azerbaijani utterances were trained with the rest datasets. The results of each model are shown below (Table 5).

Table 5. Outputs of the second experiment

Model	Accuracy	Validation Accuracy	Average Precision	Average Recall	Average F-1 Score	Confusion Matrix
Multilayer Perceptron	79.81%	83.36%	0.82	0.798	0.798	[20 1 4 1] [1 17 1 7] [0 0 23 3] [2 0 1 23]
Convolutional Neural Network	73.08%	92.01%	0.735	0.73	0.731	[20 2 1 3] [2 20 2 2] [1 5 18 2] [1 3 4 18]

Decision Tree	100%	89.3%	1	1	1	[26 0 0 0] [0 26 0 0] [0 0 26 0] [0 0 0 26]
Random Forest	100%	91%	1	1	1	[26 0 0 0] [0 26 0 0] [0 0 26 0] [0 0 0 26]
Gradient Boost Decision Tree	74.04%	77.8%	0.76	0.74	0.744	[19 1 4 2] [4 19 0 3] [7 0 18 1] [2 0 3 21]
Linear SVC	29.81%	73.1%	0.311	0.298	0.264	[6 12 8 0] [6 6 12 2] [4 4 17 1] [4 8 12 2]
Multinomial Naïve Bayes	23.08%	56%	0.182	0.23	0.174	[3 7 0 16] [2 5 0 19] [3 5 0 18] [2 8 0 16]
Logistic Regression	24.04%	65.3%	0.233	0.24	0.229	[3 8 11 4] [3 8 9 6] [1 5 10 10] [6 9 7 4]

7.3 Experiment 3

The last experiment I tested was by training and testing on just the Azerbaijani dataset. But the dataset is small and not as structural as the other datasets. This opened a gate to an underfit results. See the results in Table 6.

Table 6. Outputs of the third experiment

Model	**Accuracy**	**Validation Accuracy**	**Average Precision**	**Average Recall**	**Average F-1 Score**	**Confusion Matrix**
Multilayer Perceptron	37.50%	22%	0.395	0.312	0.312	[1 1 0 2] [1 0 0 1] [0 0 2 3] [2 0 0 3]
Convolutional Neural Network	25%	79.5%	0.204	0.287	0.225	[2 1 2 0] [2 1 1 0] [0 0 1 1] [1 2 2 0]

Decision Tree	25%	29%	0.287	0.225	0.238	[2 1 0 1] [1 0 1 0] [1 3 1 0] [1 1 2 1]
Random Forest	18.75%	35%	0.375	0.162	0.201	[1 2 1 0] [1 0 1 0] [0 4 1 0] [2 1 1 1]
Gradient Boost Decision Tree	31.25%	40%	0.416	0.262	0.261	[1 1 2 0] [0 0 2 0] [0 2 3 0] [2 0 2 1]
Linear SVC	18.75%	30%	0.312	0.162	0.205	[1 2 0 1] [1 0 1 0] [1 3 1 0] [1 3 0 1]
Multinomial Naïve Bayes	18.75%	19.6%	0.166	0.312	0.154	[1 1 2 0] [0 2 0 0] [0 5 0 0] [1 4 0 0]
Logistic Regression	25%	39.1%	0.375	0.362	0.226	[1 3 0 0] [0 2 0 0] [0 4 1 0] [2 3 0 0]

8 PROTOTYPE IMPLEMENTATION USING TELEGRAM

After all these experiments I have decided to create a simple interfaced prototype using Telegram Bots once again. With this bot (@SERuiBot) user can record a voice and the bot will immediately start processing it with the following steps:

1. Before the processing anything it loads all models saved into a pickle and h5 files;
2. It converts the recorded audio track into the .wav file;
3. Extracts features from that wav file;
4. Predicts the emotion using all the models;
5. Sends back a message with predicted emotion.

The models used for this bot to function are the ones from experiment 2, as they have included all possible datasets I had. At the same time, the predicted emotions have different weights depending on the model. The models that had higher overall performance should have a higher importance level. Neural networks have 2x of the importance, tree-based models have 1.5x importance and the rest have 1x importance weights. The manual check of performance showed average results, and that is mainly due to the limitation of available data.

9 DISCUSSION AND FUTURE WORK

In this project, the non-linguistic approach to emotion detection is observed as a difficult task. As discussed in the literature review of this paper, multiple cues besides the words spoken like facial expression and environment could potentially help better understand the underlying emotion than just a voice sound. This task by itself can be hard for humans, not to mention machines.

But even if the project is focused on the non-linguistic approach, it is better if the targeted audio focuses on one specific language. With that machine can learn about specific tonalities that may be affected by cultural differences in pronunciations. And for that, in the case of the Azerbaijani dataset, I would need to collect more samples. And of course, this will also require specific tuning that will help in recognizing an emotion for a specific dataset easier.

To improve the project's outcome, it would be even better to add aiding features of sentiment analysis. This is a combination of different tasks to create a bigger one. By knowing what a person says (for that speech-to-text and then sentiment analysis of the text is required) the machine is likely to understand what their underlying emotional state is.

At the same time, there could be more potential improvements in segmenting the audio files and finding which emotion represents which piece. As it is not uncommon that people tend to have different emotions within little time period, machine should be able differ them to identify what may have cause the change of the expressions. With that, being able to show emotion at the time of communication (i.e., not analyzing a recorded file but the real-time speech) would be another improvement in the field.

Clearly, the techniques I have used in this project are not the only available ones and there potentially could be more experiments with various approaches. The topic is broad and requires more time and detailed analysis of all perspectives.

REFERENCES

[1] Tiffany, L. A. (2016, September 2). Dogs distinguish words and tone much like humans. PBS NewsHour. https://www.pbs.org/newshour/science/yes-dog-understand-youre-saying

[2] B. Schuller, G. Rigoll and M. Lang, Speech emotion recognition combining acoustic features and linguistic information in a hybrid support vector machine-belief network architecture, 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004, pp. 577–580, doi: 10.1109/ICASSP.2004.1326051.

[3] D. J. France, R. G. Shiavi, S. Silverman, M. Silverman and M. Wilkes, Acoustical properties of speech as indicators of depression and suicidal risk, in IEEE Transactions on Biomedical Engineering, vol. 47, no. 7, pp. 829-837, July 2000, doi: 10.1109/10.846676.

[4] el Ayadi, M., Kamel, M. S., & Karray, F. (2011). Survey on speech emotion recognition: Features, classification schemes, and databases. Pattern Recognition, 44(3), 572–587. https://doi.org/10.1016/j.patcog.2010.09.020.

[5] E. Spyrou, I. Vernikos, R. Nikopoulou and P. Mylonas, A Non-Linguistic Approach for Human Emotion Recognition from Speech, 2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA), 2018, pp. 1-5, doi: 10.1109/IISA.2018.8633644.

[6] Busso, C., Bulut, M., Lee, C. C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J. N., Lee, S., & Narayanan, S. S. (2008). IEMOCAP: interactive emotional dyadic motion capture database. Language Resources and Evaluation, 42(4), 335–359. https://doi.org/10.1007/s10579-008-9076-6

[7] Neumann, M., & Vu, N.T. (2017). Attentive Convolutional Neural Network Based Speech Emotion Recognition: A Study on the Impact of Input Features, Signal Length, and Acted Speech. INTERSPEECH.

[8] Sarma, B.D., Das, R., Dey, A., & Haukioja, R. (2018). Analysis of Speech Emotions in Realistic Environments.

Workshop on Speech, Music and Mind (SMM 2018).

[9] F. Chenchah and Z. Lachiri, Speech emotion recognition in noisy environment, 2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), 2016, pp. 788-792, doi: 10.1109/ATSIP.2016.7523189.

[10] Rovetta, Stefano & Mnasri, Zied & Masulli, Francesco & Cabri, Alberto. (2019). Emotion recognition from speech signal using fuzzy clustering. 10.2991/eusflat-19.2019.19.

[11] Cowen, A. S., & Keltner, D. (2017). Self-report captures 27 distinct categories of emotion bridged by continuous gradients. Proceedings of the National Academy of Sciences, 114(38). https://doi.org/10.1073/pnas.1702247114

[12] Ekman, P. (1992). An argument for basic emotions. Cognition and Emotion, 6(3–4), 169–200. https://doi.org/10.1080/02699939208411068.

[13] Russell, J. A. (1980). A circumplex model of affect. Journal of Personality and Social Psychology, 39(6), 1161–1178. https://doi.org/10.1037/h0077714.

[14] Definition of valence. (2022). Www.Dictionary.Com. https://www.dictionary.com/browse/valence

[15] Plutchik, R. (2001). The Nature of Emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice. American Scientist, 89(4), 344–350. http://www.jstor.org/stable/27857503.

[16] Anagnostopoulos, C. N., Iliou, T., & Giannoukos, I. (2012). Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011. Artificial Intelligence Review, 43(2), 155–177. https://doi.org/10.1007/s10462-012-9368-5.

[17] L. Fei-Fei and P. Perona, A Bayesian hierarchical model for learning natural scene categories, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005, pp. 524-531 vol. 2, doi: 10.1109/CVPR.2005.16.

[18] J. Sivic and A. Zisserman, Efficient Visual Search of Videos Cast as Text Retrieval, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 4, pp. 591-606, April 2009, doi: 10.1109/TPAMI.2008.111.

[19] Shuman, Vera & Scherer, Klaus & Fontaine, Johnny & Soriano, Cristina. (2015). The GRID meets the Wheel: Assessing emotional feeling via self-report. 10.13140/RG.2.1.2694.6406.

[20] Huang, C., Gong, W., Fu, W., & Feng, D. (2014). A Research of Speech Emotion Recognition Based on Deep Belief Network and SVM. Mathematical Problems in Engineering, 2014, 1–7. https://doi.org/10.1155/2014/749604.

[21] Ladefoged, P., & Broadbent, D. E. (1957). Information Conveyed by Vowels. The Journal of the Acoustical Society of America, 29(1), 98–104. https://doi.org/10.1121/1.1908694.

[22] Ladefoged, P., & Ladefoged, P. (2005). Vowels and consonants: An introduction to the sounds of languages. Malden, MA: Blackwell Pub.

[23] R. A. Cole, Yonghong Yan, B. Mak, M. Fanty and T. Bailey, The contribution of consonants versus vowels to word recognition in fluent speech, 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings, 1996, pp. 853-856 vol. 2, doi: 10.1109/ICASSP.1996.543255.

[24] Fogerty, D., & Kewley-Port, D. (2009). Perceptual contributions of the consonant-vowel boundary to sentence intelligibility. The Journal of the Acoustical Society of America, 126(2), 847–857. https://doi.org/10.1121/1.3159302.

[25] R. Milner, M. A. Jalal, R. W. M. Ng and T. Hain, A Cross-Corpus Study on Speech Emotion Recognition, 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), 2019, pp. 304-311, doi: 10.1109/ASRU46091.2019.9003838.

[26] Jalal, Asif & Milner, Rosanna & Hain, Thomas. (2020). Empirical Interpretation of Speech Emotion Perception with Attention Based Model for Speech Emotion Recognition. 10.21437/Interspeech.2020-3007.

[27] XGBoost Tree Methods — xgboost 1.5.2 documentation. (n.d.). Xgboost.readthedocs.io. Retrieved April 4, 2022, from https://xgboost.readthedocs.io/en/stable/treemethod.html

[28] Bahdanau, Dzmitry & Cho, Kyunghyun & Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. ArXiv. 1409.

[29] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing,

pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

[30] I. Idris and M. S. H. Salam, Emotion detection with hybrid voice quality and prosodic features using Neural Network, 2014 4th World Congress on Information and Communication Technologies (WICT 2014), 2014, pp. 205-210, doi: 10.1109/WICT.2014.7076906.

[31] K. Wang, N. An and L. Li, Speech emotion recognition based on wavelet packet coefficient model, The 9th International Symposium on Chinese Spoken Language Processing, 2014, pp. 478-482, doi: 10.1109/ISCSLP.2014.6936710.

[32] Xu, Lu & Xu, Mingxing & Yang, Dali. (2009). ANN based decision fusion for speech emotion recognition. 2035-2038. 10.21437/Interspeech.2009-585.

[33] H. Atassi and A. Esposito, A Speaker Independent Approach to the Classification of Emotional Vocal Expressions, 2008 20th IEEE International Conference on Tools with Artificial Intelligence, 2008, pp. 147-152, doi: 10.1109/ICTAI.2008.158.

[34] C. Huang, Y. Jin, Y. Zhao, Y. Yu and L. Zhao, Speech emotion recognition based on re-composition of two-class classifiers, 2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops, 2009, pp. 1-3, doi: 10.1109/ACII.2009.5349420.

[35] Zhang, J., Yin, Z., Chen, P., & Nichele, S. (2020). Emotion recognition using multi-modal data and machine learning techniques: A tutorial and review. Information Fusion, 59, 103–126. https://doi.org/10.1016/j.inffus.2020.01.011.

[36] Kerkeni, Leila & Serrestou, Youssef & Mbarki, Mohamed & Raoof, Kosai & Mahjoub, Mohamed. (2018). Speech Emotion Recognition: Methods and Cases Study. 175-182. 10.5220/0006611601750182.

[37] Kerkeni, Leila & Serrestou, Youssef & Raoof, Kosai & Cléder, Catherine & Mahjoub, Mohamed & Mbarki, Mohamed. (2019). Automatic Speech Emotion Recognition Using Machine Learning. 10.5772/intechopen.84856.

[38] Amiripalli, S. S., Bobba, V., & Potharaju, S. P. (2018). A Novel Trimet Graph Optimization (TGO) Topology for Wireless Networks. Cognitive Informatics and Soft Computing, 75–82. https://doi.org/10.1007/978-981-13-0617-4_8.

[39] Potharaju, S , Sreedevı, M . (2018). A Novel Cluster of Quarter Feature Selection Based on Symmetrical Uncertainty. Gazi University Journal of Science, 31 (2) , 456-470 .

[40] R. Lotfidereshgi and P. Gournay, Biologically inspired speech emotion recognition, 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 5135-5139, doi: 10.1109/ICASSP.2017.7953135.

[41] R. Munkong and B. Juang, Auditory perception and cognition, in IEEE Signal Processing Magazine, vol. 25, no. 3, pp. 98-117, May 2008, doi: 10.1109/MSP.2008.918418.

[42] T. W. Troyer and K. D. Miller, Integrate-and-fire neurons matched to physiological fi curves yield high input sensitivity and wide dynamic range, in Computational Neuroscience, pp. 197–201. Springer, 1997.

[43] Song, S., & Abbott, L. (2001). Cortical Development and Remapping through Spike Timing-Dependent Plasticity. Neuron, 32(2), 339–350. https://doi.org/10.1016/s0896-6273(01)00451-2.

[44] Sample Rates - Audacity Manual. (n.d.). Audacity Manual. https://manual.audacityteam.org/man/sample_rates.html

[45] 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. https://doi.org/10.1371/journal.pone.0196391

[46] Surrey Audio-Visual Expressed Emotion (SAVEE) Database. (n.d.). SAVEE. http://kahlan.eps.surrey.ac.uk/savee/

[47] G. Klasmeyer, Akustische Korrelate des stimmlichemotionalen Ausdrucks in der Lautsprache, Forum Phoneticum 67, Hector-Verlag, Frankfurt, 1999

[48] W. Sendlmeier, G. Klasmeyer, Voice and Emotional States, in Voice Quality Measurement, p. 339-357, Singular, San Diego, CA, 2000

[49] W. Sendlmeier, Phonetische Reduktion und Elaboration bei emotionaler Sprechweise, in Von Sprechkunst und Normphonetik, p. 169-177. Verlag Werner Dausien, Hanau, Halle, 1997

[50] K. R. Scherer, “Speech and Emotional States”, in: Darby, J. K. (ed.), The Evaluation of Speech in Psychiatry, New York: Grune & Stratton, p. 189-220, 1981

[51] K. J. Kohler, Articulatory Reduction in Different Speaking Styles, Proceedings ICPhS ‘95, Stockholm, Vol. 2, p. 12-19, 1995

[52] Kleinginna, P. R., & Kleinginna, A. M. (1981). A categorized list of emotion definitions, with suggestions for a consensual definition. Motivation and Emotion, 5(4), 345–379. https://doi.org/10.1007/bf00992553

[53] Hozjan, V., Kačič, Z. Context-Independent Multilingual Emotion Recognition from Speech Signals. International Journal of Speech Technology 6, 311–320 (2003). https://doi.org/10.1023/A:1023426522496

[54] Banse, Rainer & Scherer, Klaus. (1996). Acoustic Profiles in Vocal Emotion Expression. Journal of personality and social psychology. 70. 614-36. 10.1037/0022-3514.70.3.614.

[55] Fernandez, R., & Picard, R.W. (2004). A computational model for the automatic recognition of affect in speech.

[56] C. Williams, K. Stevens, Vocal correlates of emotional states, In JK Darby (Ed.), The evaluation of speech in psychiatry. New York: Grune & Stratton, 1981, pp. 189–220.

[57] Janet E. Cahn (1990). Generation of Affect in Synthesized Speech. Journal of the American Voice I/O Society, 8, 1–19.

[58] Hirschberg, J., & Liscombe, J. (2007). Prosody and speaker state: paralinguistics, pragmatics, and proficiency.

[59] Rabiner, L., Schafer, R. (1978). Digital Processing of Speech Signals. Englewood Cliffs: Prentice Hall.

[60] Hu, Hao & Xu, Mingxing & Wu, Wei. (2007). Fusion of global statistical and segmental spectral features for speech emotion recognition. 2269-2272. 10.21437/Interspeech.2007-616.

[61] D. Ververidis and C. Kotropoulos, Emotional Speech Classification Using Gaussian Mixture Models and the Sequential Floating Forward Selection Algorithm, 2005 IEEE International Conference on Multimedia and Expo, 2005, pp. 1500-1503, doi: 10.1109/ICME.2005.1521717.

[62] M. T. Shami and M. S. Kamel, Segment-based approach to the recognition of emotions in speech, 2005 IEEE International Conference on Multimedia and Expo, 2005, pp. 4 pp.-, doi: 10.1109/ICME.2005.1521436.

[63] R. W. Picard, E. Vyzas and J. Healey, Toward machine emotional intelligence: analysis of affective physiological state, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 10, pp. 1175-1191, Oct. 2001, doi: 10.1109/34.954607.

[64] Nwe, Tin & Foo, S.W. & De Silva, Liyanage. (2003). Speech Emotion Recognition Using Hidden Markov Models. Speech Communication. 41. 603-623. 10.1016/S0167-6393(03)00099-2.

[65] Lee, Chul & Yildirim, Serdar & Bulut, Murtaza & Kazemzadeh, Abe & Busso, Carlos & Lee, Sungbok & Narayanan, Shrikanth. (2004). Emotion Recognition based on Phoneme Classes. Proc. ICSLP. 10.21437/Interspeech.2004-322.

[66] Leinonen, L., Hiltunen, T., Linnankoski, I., & Laakso, M. J. (1997). Expression or emotional-motivational connotations with a one-word utterance. The Journal of the Acoustical Society of America, 102(3), 1853–1863. https://doi.org/10.1121/1.420109

[67] R. Cowie et al., Emotion recognition in human-computer interaction, in IEEE Signal Processing Magazine, vol. 18, no. 1, pp. 32-80, Jan 2001, doi: 10.1109/79.911197.

[68] C. Busso, S. Lee and S. Narayanan, Analysis of Emotionally Salient Aspects of Fundamental Frequency for Emotion Detection, in IEEE Transactions on Audio, Speech, and Language Processing, vol. 17, no. 4, pp. 582-596, May 2009, doi: 10.1109/TASL.2008.2009578.

[69] Louis ten Bosch. 2003. Emotions, speech and the ASR framework. Speech Commun. 40, 1–2 (April 2003), 213–225. doi: https://doi.org/10.1016/S0167-6393(02)00083-3

[70] Cowie, Roddy & Cornelius, Randolph. (2003). Describing the emotional states that are expressed in speech. Speech Communication. 40. 5-32. 10.1016/S0167-6393(02)00071-7.

[71] R. Cowie and E. Douglas-Cowie, Automatic statistical analysis of the signal and prosodic signs of emotion in speech, Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96, 1996, pp. 1989-1992 vol.3, doi: 10.1109/ICSLP.1996.608027.

[72] T. Johnstone, K.R. Scherer, Vocal Communication of Emotion, second ed., Guilford, New York, 2000, pp. 226–235.

[73] Murray, I. R., & Arnott, J. L. (1993). Toward the simulation of emotion in synthetic speech: a review of the literature on human vocal emotion. The Journal of the Acoustical Society of America, 93(2), 1097–1108.

https://doi.org/10.1121/1.405558.

[74] Chul Min Lee and S. S. Narayanan, Toward detecting emotions in spoken dialogs, in IEEE Transactions on Speech and Audio Processing, vol. 13, no. 2, pp. 293-303, March 2005, doi: 10.1109/TSA.2004.838534.

[75] A. Oster, A. Risberg, The identification of the mood of a speaker by hearing impaired listeners, Speech Transmission Lab. Quarterly Progress Status Report 4, Stockholm, 1986, pp. 79–90.

[76] Beeke, Suzanne & Wilkinson, Ray & Maxim, Jane. (2009). Prosody as a compensatory strategy in the conversations of people with agrammatism. Clinical linguistics & phonetics. 23. 133-55. 10.1080/02699200802602985.

[77] K. Hirose, H. Fujisaki and M. Yamaguchi, Synthesis by rule of voice fundamental frequency contours of spoken Japanese from linguistic information, ICASSP '84. IEEE International Conference on Acoustics, Speech, and Signal Processing, 1984, pp. 597-600, doi: 10.1109/ICASSP.1984.1172814.

[78] M. Borchert and A. Dusterhoft, Emotions in speech - experiments with prosody and quality features in speech for use in categorical and dimensional emotion recognition environments, 2005 International Conference on Natural Language Processing and Knowledge Engineering, 2005, pp. 147-151, doi: 10.1109/NLPKE.2005.1598724.

[79] Jianhua Tao, Yongguo Kang and Aijun Li, Prosody conversion from neutral speech to emotional speech, in IEEE Transactions on Audio, Speech, and Language Processing, vol. 14, no. 4, pp. 1145-1154, July 2006, doi: 10.1109/TASL.2006.876113.

[80] Davitz, J. R., & Beldoch, M. (1964). The communication of emotional meaning: [by] Joel R. Davitz, with Michael Beldoch [et al.]. New York: McGraw-Hill.

[81] Scherer, K.R. (1986). Vocal affect expression: a review and a model for future research. Psychological bulletin, 99 2, 143-65.

[82] Gobl, C., & Chasaide, A.N. (2003). The role of voice quality in communicating emotion, mood and attitude. Speech Commun., 40, 189-212.

[83] R. Sun, E. Moore and J. F. Torres, Investigating glottal parameters for differentiating emotional categories with similar prosodics, 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, 2009, pp. 4509-4512, doi: 10.1109/ICASSP.2009.4960632.

[84] X. Li et al., Stress and Emotion Classification using Jitter and Shimmer Features, 2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07, 2007, pp. IV-1081-IV-1084, doi: 10.1109/ICASSP.2007.367261.

[85] M. P. Gelfer, D. M. Fendel, Comparisons of jitter, shimmer, and signal-to-noise ratio from directly digitized versus taped voice samples, Journal of Voice, Volume 9, Issue 4, 1995, pp 378-382, ISSN 0892-1997, doi: 10.1016/S0892-1997(05)80199-7.

[86] Hansen, J.H., & Bou-Ghazale, S.E. (1997). Getting started with SUSAS: a speech under simulated and actual stress database. EUROSPEECH, vol. 4, 1997, pp. 1743–1746.

[87] Kaiser, L. Communication of affects by single vowels. Synthese 14, 300–319 (1962). https://doi.org/10.1007/BF00869311.

[88] J. Hernando and C. Nadeu, Linear prediction of the one-sided autocorrelation sequence for noisy speech recognition, in IEEE Transactions on Speech and Audio Processing, vol. 5, no. 1, pp. 80-84, Jan. 1997, doi: 10.1109/89.554273.

[89] R. Le Bouquin. 1996. Enhancement of noisy speech signals: application to mobile radio communications. Speech Commun., 18, 1 (Jan. 1996), 3–19. doi: 10.1016/0167-6393(95)00021-6.

[90] S. E. Bou-Ghazale and J. H. L. Hansen, A comparative study of traditional and newly proposed features for recognition of speech under stress, in IEEE Transactions on Speech and Audio Processing, vol. 8, no. 4, pp. 429-442, July 2000, doi: 10.1109/89.848224.

[91] Rabiner, L. R., & Juang, B. H. (1993). Fundamentals of speech recognition. Englewood Cliffs, N.J: PTR Prentice Hall.

[92] Deller, J. R., Proakis, J. G., & Hansen, J. H. L. (1993). Discrete-time processing of speech signals. New York: Macmillan Pub. Co.

[93] Emo-DB. (n.d.). Emodb.bilderbar.info. http://emodb.bilderbar.info/index-1280.html

[94] Toronto emotional speech set (TESS) | TSpace Repository. (n.d.). Tspace.library.utoronto.ca.

https://tspace.library.utoronto.ca/handle/1807/24487

[95] sklearn.preprocessing.StandardScaler — scikit-learn 0.21.2 documentation. (2019). Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[96] sklearn.preprocessing.MinMaxScaler — scikit-learn 0.22.1 documentation. (2019). Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

[97] Roy, B. (2020, April 7). All about Feature Scaling. Medium. https://towardsdatascience.com/all-about-feature-scaling-bcc0ad75cb35

[98] Kuhn, M., & Johnson, K. (2013). Applied Predictive Modeling (1st ed. 2013, Corr. 2nd printing 2018 ed.). Springer.

[99] sklearn.feature_selection.f_classif. (n.d.). Scikit-Learn. Retrieved April 4, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_classif.html

[100] sklearn.feature_selection.RFE — scikit-learn 0.23.1 documentation. (n.d.). Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html

[101] sklearn.feature_selection.SequentialFeatureSelector. (n.d.). Scikit-Learn. Retrieved April 4, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SequentialFeatureSelector.html

[102] sklearn.feature_selection.SelectFromModel — scikit-learn 0.23.1 documentation. (n.d.). Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectFromModel.html

[103] sklearn.feature_selection.SelectKBest — scikit-learn 0.23.0 documentation. (n.d.). Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html

[104] sklearn.linear_model.LogisticRegression — scikit-learn 0.21.2 documentation. (2014). Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[105] Haibo He and Yunqian Ma. 2013. Imbalanced Learning: Foundations, Algorithms, and Applications (1st. ed.). Wiley-IEEE Press.

[106] Pipeline — Version 0.9.0. (n.d.). Imbalanced-Learn.org. Retrieved April 4, 2022, from https://imbalanced-learn.org/stable/references/generated/imblearn.pipeline.Pipeline.html

[107] Minsky, M., & Papert, S. (1969). Perceptrons: An introduction to computational geometry. Cambridge, Mass: MIT Press.

[108] sklearn.neural_network.MLPClassifier — scikit-learn 0.20.3 documentation. (2010). Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

[109] Machine Learning Decision Tree Classification Algorithm - Javatpoint. (n.d.). www.javatpoint.com. https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm

[110] sklearn.tree.DecisionTreeClassifier — scikit-learn 0.22.1 documentation. (2019). Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[111] Team, K. (n.d.). Keras documentation: Conv1D layer. Keras.io. https://keras.io/api/layers/convolution_layers/convolution1d/

[112] Scikit-learn. (2018). 3.2.4.3.1. sklearn.ensemble.RandomForestClassifier — scikit-learn 0.20.3 documentation. Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[113] XGBoost Tree Methods — xgboost 1.5.2 documentation. (n.d.). Xgboost.readthedocs.io. Retrieved April 4, 2022, from https://xgboost.readthedocs.io/en/stable/treemethod.html

[114] sklearn.svm.LinearSVC — scikit-learn 0.24.1 documentation. (n.d.). Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

[115] sklearn.naive_bayes.MultinomialNB — scikit-learn 0.22 documentation. (2019). Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html

[116] Telegram Bot API. (n.d.). Telegrambot. https://core.telegram.org/bots/api

[117] About FFmpeg. (n.d.). Ffmpeg. https://ffmpeg.org/about.html

[118] sklearn.model_selection.train_test_split. (n.d.). Scikit-Learn. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

[119] Sauter, Disa & Eisner, Frank & Calder, Andrew & Scott, Sophie. (2010). Perceptual Cues in Nonverbal Vocal

Expressions of Emotion. Quarterly journal of experimental psychology (2006). 63. 2251-72. 10.1080/17470211003721642.

APPENDICES

A.1 Complete table with models and their outputs.

Note that these values are affected by usage of StandardScaler, static-based feature selection methods and slightly different tuning methods for the classifiers.

Dataset	**Model**	**Epochs**	**Accuracy**	**Precision**				**Recall**				**F-1 Score**				**Confusion Matrix**
				Neutral	Positive	Angry	Sad	Neutral	Positive	Angry	Sad	Neutral	Positive	Angry	Sad	
RAVDESS + SAVEE + TESS + EMODB	Multilayer Perceptron	103	88.94%	0.87	0.878	0.886	0.925	0.924	0.899	0.850	0.88	0.893	0.888	0.868	0.903	[196 9 5 8] [6 171 18 6] [4 8 197 4] [17 5 8 224]
	Convolutional Neural Network	28	88.49%	0.827	0.863	0.935	0.929	0.913	0.92	0.898	0.803	0.868	0.89	0.916	0.862	[201 11 6 2] [12 208 2 4] [12 4 203 7] [18 18 6 172]
	Decision Tree	-	74.27%	0.669	0.825	0.675	0.8	0.704	0.761	0.726	0.771	0.686	0.792	0.7	0.785	[166 12 17 23] [8 146 34 13] [10 40 150 13] [17 18 23 196]
	Random Forest	34	88.26%	0.865	0.871	0.873	0.915	0.873	0.931	0.82	0.897	0.869	0.9	0.846	0.906	[203 4 7 4] [7 165 18 11] [3 18 186 6] [20 2 4 228]
	Gradient Boost Decision Tree	-	84.42%	0.831	0.805	0.867	0.88	0.882	0.931	0.751	0.811	0.856	0.863	0.805	0.844	[203 4 7 4] [9 151 23 18] [5 14 188 6] [35 5 8 206]
	Linear Support Vector Classifier	5015	77.65%	0.76	0.76	0.753	0.824	0.788	0.784	0.731	0.795	0.774	0.772	0.742	0.809	[171 19 14 14] [10 147 26 18] [14 20 168 11] [30 9 13 202]
	Multinomial Naïve Bayes	-	48.98	0.574	0.725	0.401	0.456	0.342	0.302	0.353	0.881	0.429	0.427	0.375	0.601	[66 26 21 105] [6 71 25 99] [10 67 73 63] [9 13 8 224]
	Logistic Regression	94	79.12	0.803	0.762	0.84	0.771	0.826	0.839	0.761	0.744	0.814	0.799	0.798	0.757	[183 4 9 22] [8 153 19 21] [6 18 176 13] [43 7 15 189]