UNIVERSITY of BRADFORD

Faculty of Engineering and Informatics
Department of Computer Science

Speech Emotion Recognition

Dissertation Project Report

Mikaeel Akhtar

18003714

A thesis submitted in part fulfilment of the degree of BSc (Hons) Computer Science

Supervisor: Dr Amr Rashad Ahmed Abdullatif

05 May 2022

Declaration

The candidate confirms that the work submitted is his/her own and that appropriate credit has been given where reference has been made to the work of others. The candidate agrees that this report can be electronically checked for plagiarism.

Mikaeel Akhtar

Table of Contents

Abstract

In this report, I will be exploring the field of speech emotion recognition (**SER**) in detail. I will be looking at machine learning (**ML**) algorithms and techniques to analyse audio files and detect the emotion being expressed within the audio clips.

Within my report, I will explore the work of others in the field of SER and other related fields and finally I shall produce a prototype application using the python programming language that can perform SER to an extent and I will then communicate my findings in this report.

1. Chapter One: Introduction

1.1 Relevance to degree

This project and report will be related to SER and will include concepts ranging from AI and machine learning to software engineering concepts that will be incorporated in the building of my GUI program.

The degree being pursued is a BSc in Computer Science and this project extends the modules taught at the University of Bradford for Computer Science.

1.2 What is Speech Emotion Recognition?

SER is a relatively new field of study with very few resources available detailing it. The perception of emotion within speech as a human is down to many factors such as; the amplitude of the voice (how loud or quiet), the frequency/pitch (how deep/high), the speed at which words are spoken, the emphasis put on a certain sound on a word or the quality of the sound made (is the voice shaky indicating nervousness or a coarse voice from shouting?). All these little indications can be picked up by a human to quickly and effortlessly perceive the emotion someone is conveying when they speak.

The same unfortunately does not translate over to a machine. Machines (so far) lack the general cognition of a human to understand speech but in the same way, we can simulate intelligence that we have amply named 'artificial intelligence', we can also simulate learning and understanding through machine learning.

1.3 Potential Uses of SER

With the current push towards AI assistants and them being built into our phones, cars and even homes, SER would be an obvious benefit for them to provide a more personal interaction with the user. If the user makes a request to their AI assistant in happiness and excitement, it would be much better for the assistant to recognise that emotion and then reciprocate it in its response. An assistant could also use emotion to more accurately deduce the intention of a question being asked by the user.

In regards to international security, SER could be used in airports or by police to notice more accurately if someone is not being truthful about their details if the system is capable of picking up emotions like nervousness.

Once optimised well enough to run efficiently and accurately in real-time, SER could then be used in the entertainment sector in places like video games to reflect the emotion the user has displayed on the character model (make the character smile if what is being said sounds happy) or for the AI to give accurate responses to your emotions.

1.4 What is Machine Learning

Machine learning is a part of artificial intelligence and uses computer algorithms to allow a model to 'learn' through experience. A dataset is required to train the machine learning model and once the model has been trained, it is then given test data to predict a variable. The predictions are then checked against the real values to see how accurate the model is.

Many things can impact the quality of the model. The training dataset being too small can be detrimental but the dataset being too large can also be bad for the model as it will then be too specialised for the data it has been trained on. This is called overfitting as it will have a very high accuracy with any data from the same dataset but when presented with real data, it will be mostly inaccurate. Any anomalies and errors in the data will also throw the model off with the model being fit to the errors.

1.5 Supervised Machine Learning

There are 3 types of machine learning; supervised, unsupervised and reinforcement learning. Supervised machine learning has its datasets labelled so that the algorithm can look at all the features corresponding to each labelled input and then accordingly adjust the weighting of said features. This is the type of ML that I will focus on as it's what will be used in my final project.

1.6 Report Overview

The following is how this report will be formatted:

Chapter 2: Literature Review – This section contains a literature review carried out prior to the implementation of my own SER project. The section will briefly touch upon the topic of the physiology of speech followed by computer recognition of speech as well as machine learning and AI.

Chapter 3: Requirements and Analysis – This section covers the requirements of the project and speaks more specifically on the actual techniques, libraries, datasets etc I planned on using for my actual project. This section will give a clear explanation as to what functionalities I planned on implementing in my project prototype. I will also justify why I decided to take the route that I did and communicate my thought process.

Chapter 4: Design, Implementation and Testing – This section explains how I designed my project. The section speaks specifically on how I met the functionalities I planned on including in Chapter 3. It also includes snippets of code where applicable to explain how the final product was achieved. This section also contains my chosen form of testing for this project and my justification for choosing said testing method.

Chapter 5: Results, discussion, accuracy and performance – This section will talk about the results I achieved as well as discuss the accuracy and performance of my ML models. I will also have some speculation as to why certain results came out the way they did.

Chapter 6: Conclusions and further work – This section will give a conclusion of the entire project and report as well as where this project could lead potentially in the future.

2. Chapter Two: Literature Review

2.1 Introduction

This chapter will be a literature review that has been conducted on the works of others available online. The literature reviewed will go from being broadly related to my works and become progressively more relevant and specific to my own project as this section goes on. Any literature will be referenced in the bibliography below.

2.2 The Science of Emotion in Speech

In 'The expression of emotions in man and animals' by Darwin (1872), he refers to the research on music by Herbert Spencer. Spencer suggested that music naturally developed from the natural musical rhythm, pitch, and contours of emotional speech but Darwin disagreed. Darwin countered that it was the opposite and that music and musical sounds came first, and they gave rise to speech. Darwin suggested that emotional expressions are evolved and adaptive. It is still inconclusive which idea is correct, but we can tell for sure that music and speech are both very closely linked.

However, sounds aren't objectively or directly linked to emotion. Words of anger can be spoken softly without the expected trait of the speaker shouting the words. This would probably throw off even a human and make it difficult to interpret the intention behind the words. Darwin suggesting that emotions are evolutionary would also mean that the hallmarks of an emotion such as anger in the future will not be the same as they are now. In the future anger may be expressed with what is now considered a 'soft' tone.

Regardless, currently, we learn how to perceive emotions at a very young age, through our environment such as parents shouting to imply their anger. Upbeat and soft songs on children's television create happy emotions, and loud sudden noises create fear. By being scared by a loud sound, a child can then imitate such a sound, such as shouting 'boo!' to try and create the same fear they experienced. Through that, they have learnt from feeling emotion how to instil that same emotion upon someone else through language and sound.

2.3 The Way Computers Interpret Sound

In our day-to-day lives, we communicate with computers through speech often. Whether it be a virtual agent on the phone directing you to more relevant information or a designated smart assistant from the many offerings of Google, Amazon and Apple. But how do all these computer systems take the vibrations from speech then turn that into something that it can process, and then process it to understand what words were actually spoken?

This is through the science of waves, more specifically sound waves. Figure 1 below is a graph of a part of a sound wave.

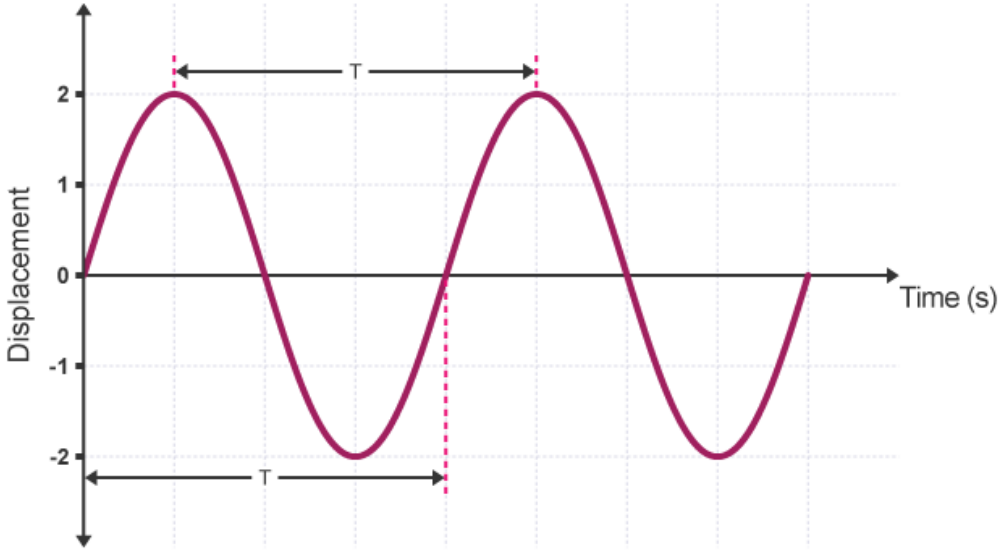


Figure 1 (BBC no date)

The x-axis of this wave is time and the y-axis is the displacement of the wave. The number of waves that occur within a second would tell you the frequency of the sound, the higher the frequency, the higher-pitched the sound will be. The lower the frequency will result in the opposite. The amplitude of the wave shows how loud the sound will be, a larger amplitude being louder.

Using a microphone, a computer system can then take the analogue wave of sound and then use something called a transducer which simply converts energy from one form to another. The sound wave is then translated into something the computer can interpret.

This raw data however isn't something the computer understands as words. It only means that the computer can now perform calculations and process the sound data to then understand it.

"Speech is actually a very complex mix of multiple waves coming at different frequencies" (Okrent, 2012) therefore there are many steps to turn this raw speech data into intelligible words. There are around 44 phenomes in the English language and a computer must be trained on each phenome to be able to split up the sound into chunks and then try and classify each chunk into one of the phenomes the system has been trained on. This creates a rough guess of what the words may be but it still won't be very accurate at predicting sentences and phrases. This is because many words sound similar and therefore further processing must be done to check how likely a word is to follow another word. This way the system will be more likely to guess the words correctly rather than creating sentences full of words that don't grammatically go together.

2.4 The Way Computers Interpret Emotion in Sound

Interpreting emotion in speech is much the same as how speech is understood by a computer system as detailed above. It is simply a step further in processing the data by training the system on the features that portray emotion and then classifying either chunks of the data into different emotions (happy, sad, angry, etc) or classifying an entire sound to get the general dominant emotion throughout the entire sound clip. The latter wouldn't be

particularly useful however as if there are multiple speakers, then the system may end up interpreting an emotion in the 'middle' of what is given off by each speaker.

Since human emotions aren't an exact science with so many different ways for people to show their emotions and so many different accents as well as the mannerisms and pitch of a female when angry will generally be vastly different from the voice of a man in anger. Because of this, the only real way currently to interpret emotion in speech is through machine learning.

An article by Popova et al (2017) shows researchers already managing to create a model that can correctly classify emotions with a 70% specificity. The researchers used a neural network on spectrograms (images of audio signals) to train their model which meant that they could use the same techniques used for image recognition. The model was trained to recognise 8 emotions as follows: neutral, calm, happy, sad, angry, scared, disgusted, and surprised. According to their report, their model had issues with recognising happiness and surprise, happiness was commonly confused for fear and sadness whilst surprise was confused for disgust.

2.5 Artificial Intelligence, Machine Learning and Deep Learning

Deep learning (**DL**) is a subset of Machine Learning (**ML**) and ML is a subset of Artificial Intelligence (**AI**). AI covers a very broad spectrum of fields and can be referring to one of many things in the world of computer systems and programming.

Traditional ML however has different use cases to that of DL even though 'deep learning *is* machine learning' (Grieve, 2020). Machine learning uses algorithms to learn from information given and rules set by an operator (a human). If something goes wrong and the predictions returned by the model are incorrect, then human intervention is required to tweak the way the algorithm works or the features used, or another factor to guide the machine on the right path.

DL on the other hand can determine on its own without any human interaction if a prediction is correct or not. It will then make its own decision on what to do next to further improve its accuracy.

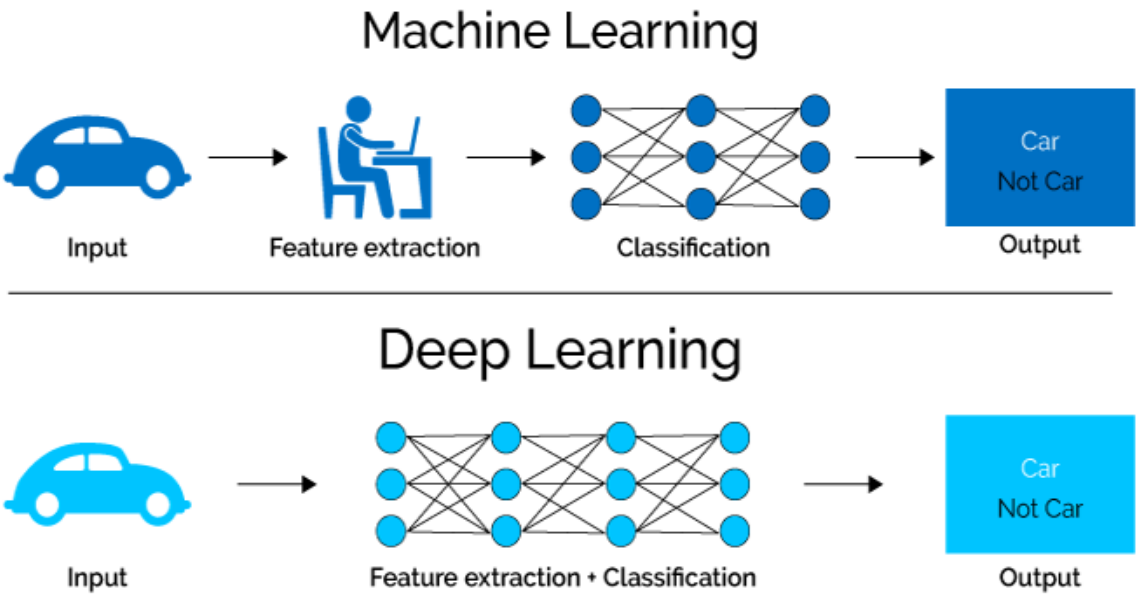


Figure 2 (Prakash 2020)

Figure 2 above shows a simplified illustration of the difference between traditional ML and DL. As it's easy to see, DL lacks the human component and carries out its feature extraction on its own to know what the best features are for what you're trying to find.

Below is a table of some more differences between traditional ML and DL:

Traditional Machine Learning	Deep Learning
Works better with a smaller amount of data	Requires thousands or even millions of data entries from a dataset to come to be accurate
Less computationally expensive	More computationally expensive
Takes less time to train and run the model	Takes much longer to run than traditional ML
Requires human intervention for feature extraction	Runs on its own, finds the best features on its own
The output from traditional ML will usually be less accurate than that of DL	More potential to be more accurate given the dataset is large enough
Traditional ML works well on both large data as well as on smaller datasets	DL is more exclusive to large data
Since the rules are clearly defined and the features are selected by the human operator, it's very clear how the algorithm gets to its conclusions	Even if the output has a high accuracy, it may be difficult to understand why or how the algorithm got to its conclusion (e.g., A cancer detection algorithm may be able to detect who may get cancer but as a human, it's difficult to understand what the causes may be to do something about it)

2.6 SER Using Convolutional Neural Networks

There have been multiple ventures into SER through traditional ML in the past but the use of DL for SER has been a relatively new experiment. According to an article by researchers at the University of Melbourne, "Supervised DL neural network models have been shown to outperform classical approaches in a wide range of classification problems" (Lech et al,

2020). In their report, they also mentioned that the classification of images has been particularly successful. By this, they are referring to the use of 2-dimensional spectrograms and using image processing algorithms instead of using the 1-dimensional speech waveforms on their own. Even though spectrograms add an extra dimension to the sound wave, it importantly doesn't degrade any information in the wave or modify it in any way.

With our knowledge of DL today and the increasing use of it over traditional ML, we know that it is likely that the output of DL will give more favourable results. This said, however, in the field of SER, datasets of speech are required. And the datasets have to have sufficient information and a mix of emotions.

From looking at some of the main datasets available as of now, there are inconsistencies in them all. Something such as the SAVEE dataset makes use of 7 different emotions and each clip is a couple of seconds at most.

The RAVDESS dataset has similar length audio clips but has audio clips for one more emotion than that of SAVEE. Combining the two datasets for processing isn't too difficult but it would be required to omit the extra emotion of 'calm'. Either that or combine them anyway but the 'calm' emotion will probably end up being overfitted for the dataset all of its samples came from compared to the rest of the emotions.

Even if you decide to proceed with only the 7 emotions, there would still be below 2000 audio files which likely wouldn't be ideal for deep learning. There are other datasets such as the IEMOCAP dataset which is an older dataset and it has short snippets cut from a dialogue. But the age of the dataset is shown with the lack of quality in the actual sound files as well as the way that the audio clips have been cut very roughly and abruptly which would make me hesitant to combine it with another dataset.

SER is very specific and the dataset it requires is very specific and with the datasets currently available, I'd opt for machine learning over deep learning for this reason.

2.7 Speech Emotion Analyser

A speech emotion analyser was created by Mitesh Puthran (2017) in python using both the RAVDESS and the SAVEE datasets. These are both datasets I was already looking into using.

On his GitHub repository of the project, it was mentioned that a convolutional neural network (CNN) was used as it was "the obvious choice" since it is a classification problem. He then goes on to say that 'multilayer perceptrons' were built as well as 'long short-term memory' models but they under-performed with low accuracies. This is useful information as I know what types of models to avoid and I can test my luck with other types of models instead.

In Puthran's project, he also separated emotions into male and female categories. E.g., male_angry, female_angry. This is something that I'm interested to test out both separating the files based on gender or combining them all.

From the conclusion of this project, the model created was able to distinguish between male and female voices ~100% of the time whereas it was able to detect emotions 70% of

the time. This is a respectable percentage accuracy and the conclusion mentioned that the accuracy can be further increased by including more files for the training process.

This project is very relevant to my own and therefore has a lot of weight on how I will conduct the creation of my project.

3. Chapter Three: Requirements and Analysis

3.1 Introduction

This chapter will provide information on what the goals are for the project. As explained in my abstract, this project will be related to speech emotion recognition (**SER**) and is what the final project will be centred around.

A detailed list of the features planned on being implemented will be given. Also, diagrams and illustrations will be used to aid in explanation.

The techniques that will be used to evaluate the results of the project will also be discussed here.

3.2 Requirements

The requirements for this project are quite simple. A machine learning model is required to be built that can detect emotions from speech. The knowledge gained from the literature review and the works of others will be used to understand the best way to go about this project and create the most accurate machine learning model possible. The different algorithms available will be evaluated different features will be tried and tested to see what will provide the highest accuracy when testing the data.

The model will also be tested on information that is at least moderately different from what it was trained on to see how it performs. A conclusion will be drawn from the results obtained from the project prototype.

3.3 Analysis

A traditional machine learning pipeline will follow the following steps:

- **Data extraction:** Getting the data that will be used. In the case of SER, this means either finding a dataset(s) that you will use to train the model or creating your own dataset. Because of the lack of resources, there won't be an attempt to create a custom dataset as it won't be large enough for there to be any obvious benefit over just using one of the publicly available datasets online.
- **Data preparation:** This is quite broad but includes any preprocessing of the data, such as getting rid of any null entries in a CSV file and replacing them with the zeroes instead. In this case, if it is desired to combine multiple datasets, it will be required to normalize the datasets, such as omitting the 'calm' emotion in the RAVDESS dataset so that it will be possible to combine it with the SAVEE dataset or combine them without omittance but this could potentially cause other problems later on.
  This step also includes feature extraction which is a very important step in machine learning. Finding the right feature is one of the biggest determining factors in the performance of your machine learning model.

- **Model Training:** This is the phase where you must adjust the 'weights and bias to minimize the loss function over the prediction range' (C3, no date). During this phase, you would want to find the best mathematical representation of the relationship between the features of speech waves and emotion.
- **Model evaluation and validation:** This is the phase in machine learning where data is gathered by testing the model against test data. Confusion matrices are created, graphs are made and precision metrics such as F1-score are calculated to find the percentage accuracy and loss. Once all this has been done, if you are happy with your model, you can then move on to the next stage of 'deployment'. Otherwise, if the evaluation returns poor results, a revision is required by revisiting the 'data preparation' stage or maybe even the 'data extraction' phase and looking for data that may yield better results.
- **Deployment:** The deployment stage is exactly as it sounds, if you are happy with your model, you can then go ahead and use it for whatever it is you'd like. For this project, this will be a program that allows you to input pre-recorded audio snippets and see if the software can accurately guess what emotion is being shown within the clip. The software will output the prediction of the emotion that has the highest percentage likelihood of being conveyed in the clip.

Figure 3 below illustrates the machine learning lifecycle in a similar manner to how it is explained above.

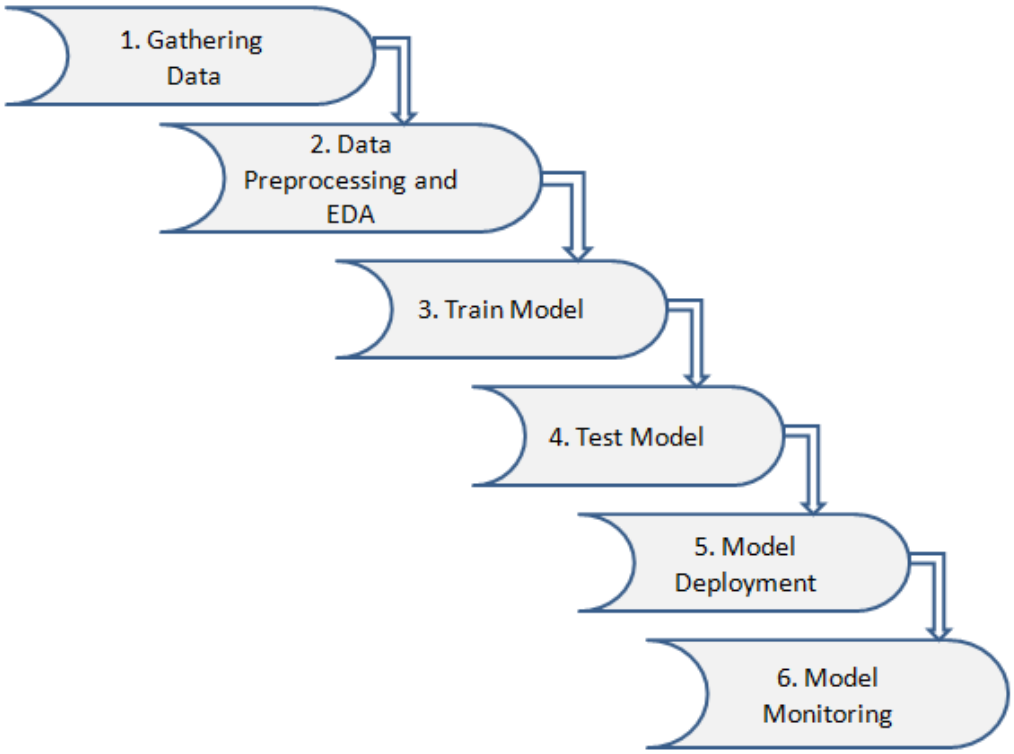


Figure 3 (Rajbanshi 2022)

There are three main types of machine learning, these are Supervised, Unsupervised and Reinforcement learning algorithms, this was spoken about lightly in chapter 1. Supervised machine learning will be used for this project as the datasets available are labelled with a corresponding emotion.

3.4 Datasets Used

Three datasets of audio files have been collated to create a single dataset with 4528 total data points. A fourth will be used for the text model. The following are the datasets used:

- The Surrey Audio-Visual Expressed Emotion (**SAVEE**) Database is a dataset made by the University of Surrey (2015). This dataset contains both videos and audio clips from 4 male actors. It contains 480 British English utterances in 7 different emotions: anger, disgust, fear, happiness, neutral, sadness, and surprise.

- The Ryerson Audio-Visual Database of Emotional Speech and Song (**RAVDESS**) (2018) database is a larger database with 24 actors (12 female, 12 male) giving a 50/50 split on gender. Just as the SAVEE database does, this database also has both videos, as well as audio clips. In addition to that, however, this dataset also contains audio clips of songs as suggested by the title.

- The Toronto Emotional Speech Set (**TESS**) is a dataset containing 2800 data points created by two female actresses. This dataset contains 62% of all data points from the three datasets combined (Dupuis and Pichora-Fuller, 2010).

- The dataset used for my text-based ML model is one by **dair-ai** on GitHub and contains over **400,000** labelled strings of pre-processed text in the form of a pandas dataframe object (Saravia et al, 2022).

None of the video clips will be used from any of the datasets nor will any of the audio clips of song be used from the RAVDESS dataset. Only use the audio snippets that are made specifically for emotion classification will be used.

The SAVEE dataset files are named with a shorthand for an emotion. An example of a .wav file that contains a sound that is ‘angry’ is a03.wav

Figure 4 shows six files. The first three contain ‘angry’ sounds and the three files proceeding them are files of ‘disgusted’ sounds.

Figure 5 shows the files in the RAVDESS dataset and it is formatted differently. The third number shows the emotion and it ranges from ‘01’ to ‘08’. In this picture, the first three files with the number ‘04’ are files containing ‘sad’ sounds and the last three with ‘05’ are files containing ‘angry’ sounds.

Figure 6 shows the files in the TESS dataset, this dataset is easier to tell the emotion for each file as it doesn’t use any shorthand or alternative but rather just uses the plain English word for each emotion such as fear, angry etc.

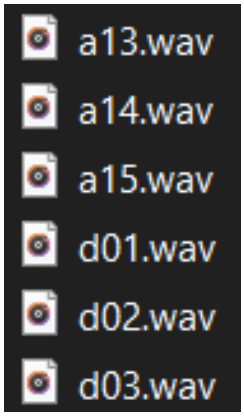


Figure 4

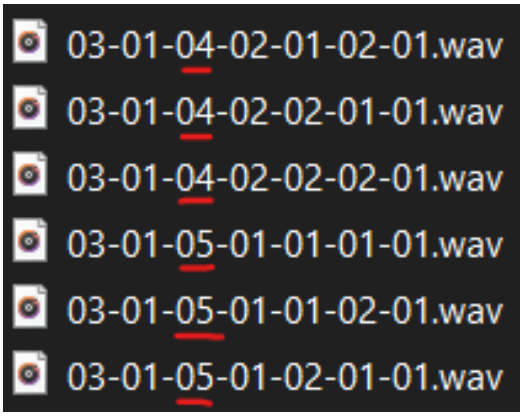


Figure 5

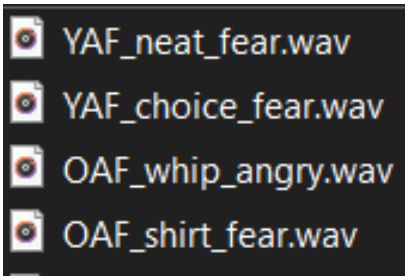


Figure 6

3.5 Libraries

Below, python libraries have been listed that are relevant to the project. All libraries below were considered for use but if they were not used in the final project, they will be marked as such.

Audio processing libraries:

- **Librosa:** A python package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems.
- **PyAudioAnalysis:** A Python library for audio feature extraction, classification, segmentation, and applications. NOT USED
- **Speech_Recognition:** Library for performing speech recognition, with support for several engines and APIs, online and offline.
- **Simpleaudio:** Library for audio playback functions.

Data analysis and ML libraries:

- **Pandas:** A fast, powerful, flexible and easy-to-use open-source data analysis and manipulation tool.
- **Numpy:** A library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
- **TensorFlow:** An end-to-end open-source platform for machine learning.
- **Keras:** An open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library.
- **Scikit-learn:** An open-source machine learning library that supports supervised and unsupervised learning. It also provides various tools for model fitting, data preprocessing, model selection, model evaluation, and many other utilities.

Graphing and visualisation libraries:

- **Matplotlib:** A comprehensive library for creating static, animated, and interactive visualisations.
- **Dash / Plotly:** This is ideal for building and deploying data apps with customized user interfaces. It's particularly suited for anyone who works with data. NOT USED
- **Seaborn:** A Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. NOT USED
- **PySimpleGUI:** A library used for the creation of GUI.

4. Chapter Four: Design, Implementation and Testing

4.1 Introduction

This section will explain how the system was designed and it will go into depth on the decisions that were made, the tools and software used and the libraries that were crucial for carrying out the task. It will include code extracts, and outputs from the software that was created including graphs and calculations of loss and accuracy.

The algorithms used will be detailed as well as which algorithms may have been tested before the final decision on which to use for the final project.

This section will also contain the testing that will be carried out on the results of the machine learning models and detail which form of testing will be used and why said testing method was chosen.

4.2 Technologies Used

For this project, Python was the language chosen. This was because of the abundance of libraries available both for AI and audio analysis. Jupyter Notebook was then used as it is easier to work with but also provides clarity for someone viewing the work externally. The code has been annotated extensively to make it clear what each code block's purpose is and to make the ML workflow clear to see.

Combine the path and label lists into a dataframe

In [3]:
```
df = pd.DataFrame()
df['path'] = paths
df['emotion'] = labels
df = df.sample(frac = 1)
df.head()
```

Out[3]:

	path	emotion
2977	Datasets/TESS\OAF_tough_happy.wav	happiness_female
1215	Datasets/RAVDESS\Actor_24\03-01-04-02-02-02-24...	sadness_female
4487	Datasets/TESS\YAF_wire_disgust.wav	disgust_female
139	Datasets/RAVDESS\Actor_03\03-01-06-02-02-02-03...	fear_male
1714	Datasets/SAVEE\KL\su02.wav	surprise_male

functions to create graphs easier

In [4]:
```
def waveplot(data, sr, emotion):
    plt.title(emotion)
    plt.ylabel('dB')
    librosa.display.waveplot(data, sr)
    plt.show()

def spectogram(data, sr, emotion):
    suppress_qt_warnings()
    x = librosa.stft(data)
    xdb = librosa.amplitude_to_db(np.abs(x))
    plt.title(emotion)
    librosa.display.specshow(xdb, sr=sr, x_axis='time', y_axis='hz')
    plt.colorbar()
    plt.show()
```

Figure 7

```
48  # Combine the path and label lists into a dataframe
49  df = pd.DataFrame()
50  df['path'] = paths
51  df['emotion'] = labels
52
53  # functions to create graphs easier
54  def waveplot(data, sr, emotion):
55      plt.figure(figsize=(10,4))
56      plt.title(emotion)
57      plt.xlabel()
58      librosa.display.waveplot(data, sr)
59      plt.show()
60
61  def spectogram(data, sr, emotion):
62      suppress_qt_warnings()
63      x = librosa.stft(data)
64      xdb = librosa.amplitude_to_db(np.abs(x))
65      plt.figure(figsize=(10,4))
66      plt.title(emotion)
67      librosa.display.specshow(xdb, sr=sr, x_axis='time', y_axis='hz')
68      plt.colorbar()
69      plt.show()
```

Figure 8

Above Figures 7 and 8 show the same snippet of code from Jupyter Notebook and VSCode respectively. From figure 7, you can see that Jupyter Notebook is easier to work with and can also output multiple graphs, etc at once whereas in VSCode the creation of something like a graph using Matplotlib would be a blocking call and therefore stop the execution of everything after the graph is displayed until the graph is closed.

4.3 Overview of Implementation

To start the project, a small subsample of one of the datasets was used to train an ML model. Once everything was verified to be working, work could start on making improvements. More data was gathered to train the model to improve its performance. This resulted in a Keras LSTM model that had been trained on three datasets. These were the SAVEE, RAVDESS, and TESS datasets.

A second model was then decided to be built using the same datasets, but the second model was gendered. For example, the first has only 7 labels such as ‘anger’, the second would have both ‘anger_male’ and ‘anger_female’. This means the second dataset has twice the number of labels at 14.

Then another model was created which was trained on text instead. The purpose of this model was to allow a second prediction based on different information that someone would give off when speaking. This ended up being a RandomForest model using scikit-learn, but a lot of the process was similar to creating the Keras model. The dataset used for this was a pre-processed dataset of tweets from Twitter with over 400,000 data points.

Now that the models had been created and trained, the next step that was carried out was to create a GUI to make it easier to try different audio files on the models, and it allows the project to be easily understood by someone with no prior knowledge in programming or ML This GUI was created in Python using PySimpleGUI. It allows the user to put whatever

files they want into a folder which they can then look through to see the model's prediction of what emotion is being conveyed in the audio clip. The GUI also shows a graph of the audio signal to provide visualisation and has built-in audio playback so that the user doesn't need to listen to the audio clip in another file. Figure 9 below shows the GUI when a file has been chosen for prediction.

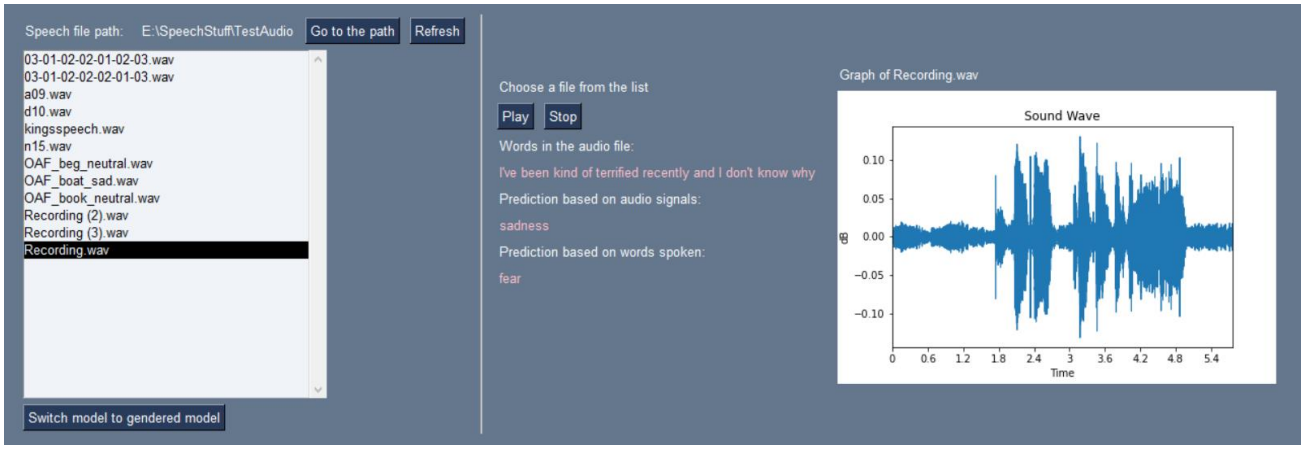
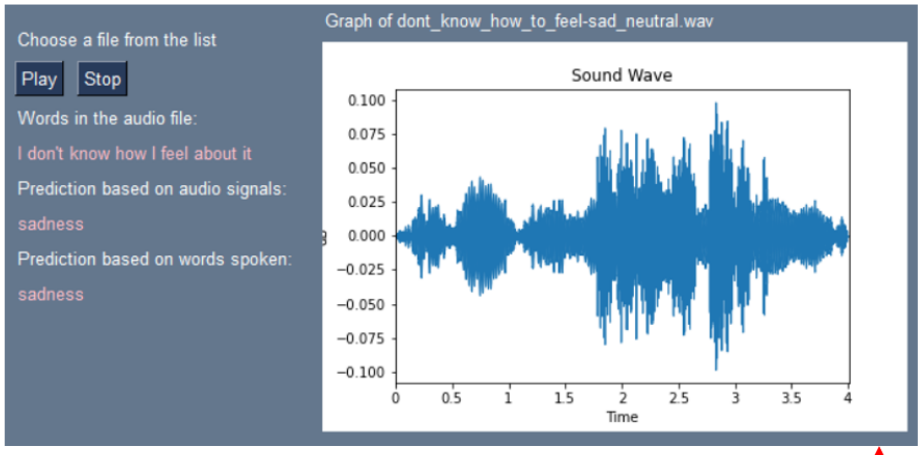


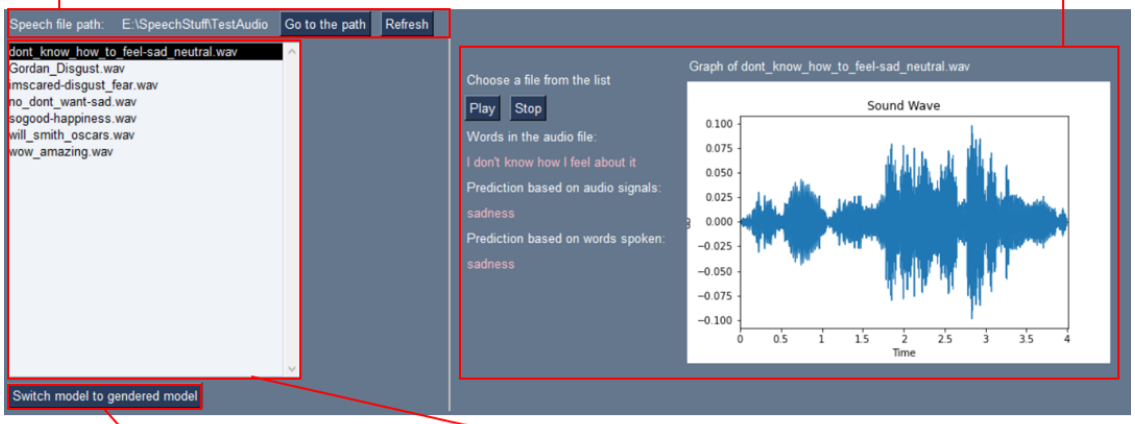
Figure 9

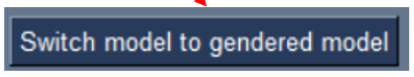
Figure 10 on the next page explains in more detail how to use the GUI software, what each button does and what information is being shown.

Path to TestAudio as well as a button that will take you to the path
Refresh button will convert compatible files to .WAV and refresh the list

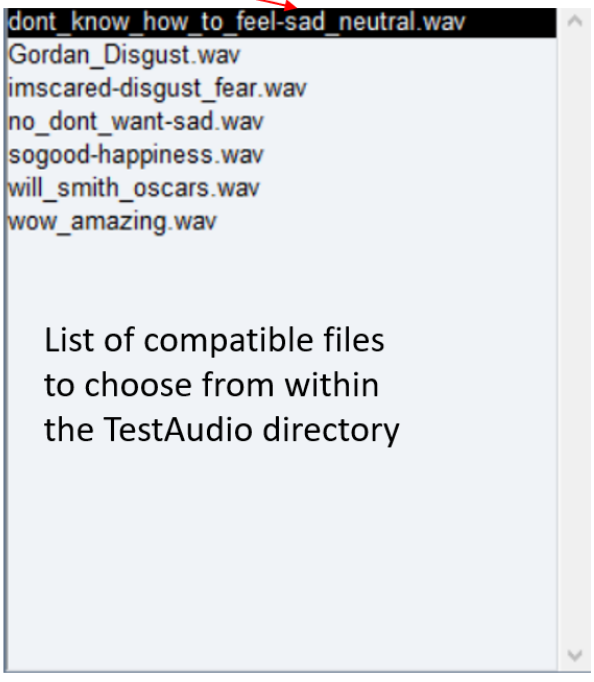


Predictions and information specific to the chosen file from the list





Button to switch the model being used between the default (ungendered) and gendered model
Change will take effect on the next file selection



List of compatible files to choose from within the TestAudio directory

Figure 10

4.4 Design Process

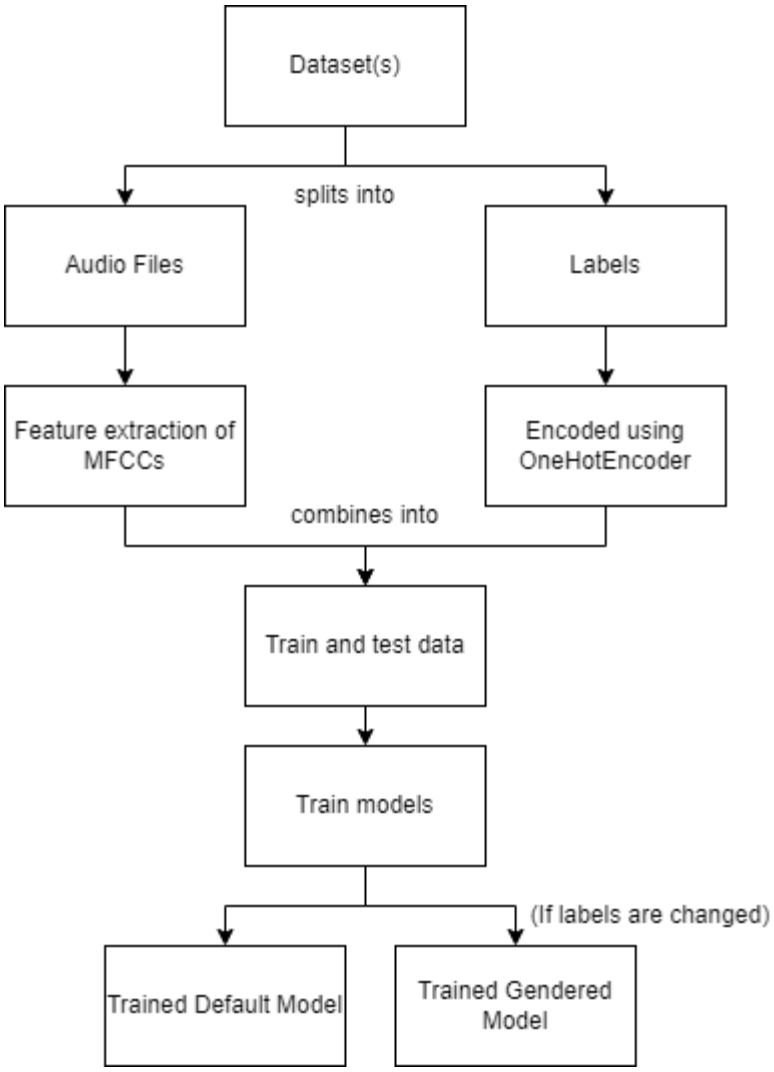


Figure 11

Figure 11 above shows the process of going from the initial dataset(s) to the creation and training of the audio signal models.

After creating the first audio model, it became apparent that there wasn't enough contextual information in the audio signals alone to provide an accurate prediction. Especially when examining the data points, at some points even I as a human couldn't tell what emotion was meant to be conveyed so it was unfair to expect the model to predict accurately based on that. Therefore, it was decided to add contextual information, speech recognition would be used to extract the spoken words as text from the audio file. Then those words would be put through a separate model that was trained on text instead to give a second prediction. The process for creating this model is shown in figure 12 below.

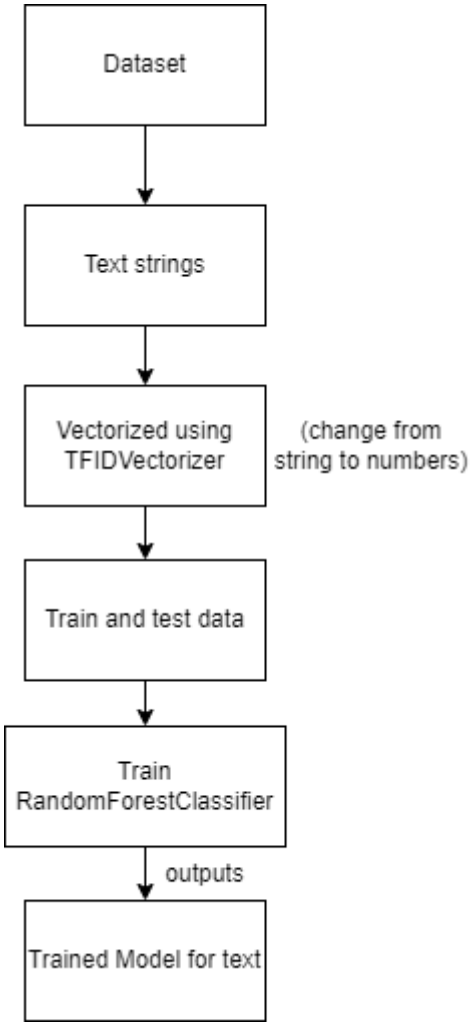


Figure 12

Unfortunately, the same data points couldn't be used as were employed for training the first model, as the sentences being spoken in the audio files were mostly gibberish and not representative of any emotion. Therefore, the twitter emotion dataset (Saravia et al, 2022) was found and the model was trained on that. This dataset unfortunately only has 6 labels as opposed to the 7 labels used for the audio model. From these, only 5 of said labels matched with the other model. For just this prototype, the software will just output both predictions to the GUI as separate predictions.

Finally, the GUI was created that connected all the models together and allowed for the models to be easily tested by passing any audio file the user would like to the software. The process for making the GUI is shown below in figure 13.

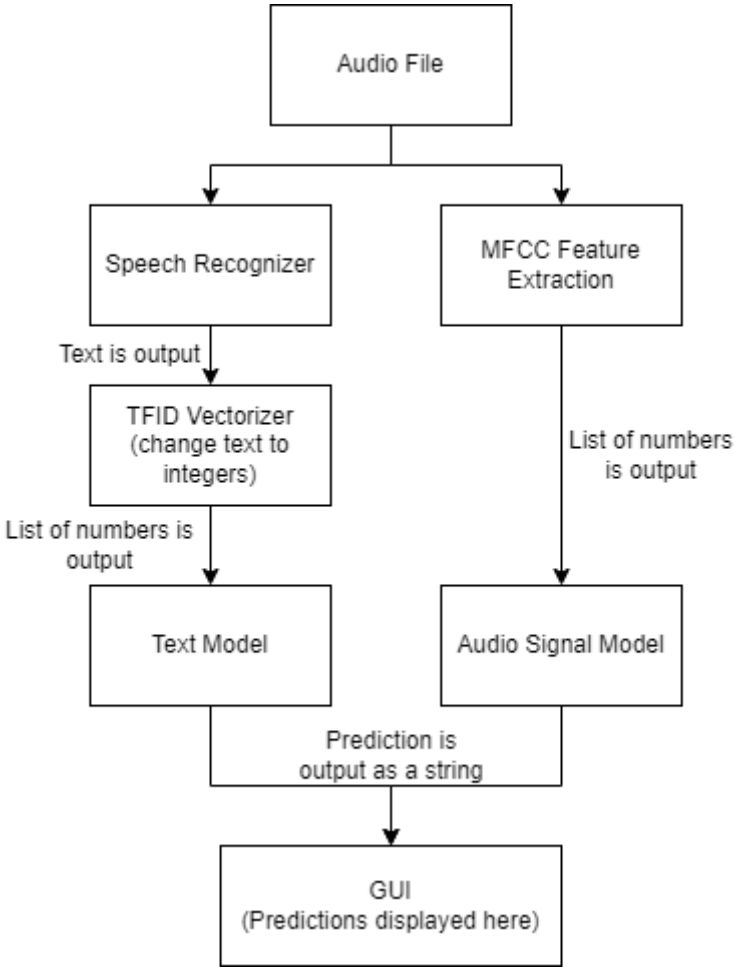


Figure 13

When an audio file is selected in the GUI, it will go through a speech recognizer and the words that are output will be converted to numbers, these numbers can then be run through the model for a prediction. At the same time, the MFCCs will be extracted from the audio signal data of the clip and these features will go through the audio model. The GUI will then present the predictions in a more presentable way than just printing to the console.

4.5 Testing: Functionality, Bugs, and Glitches

Test number	Test description	Expected outcome	Actual Outcome	Pass / Fail
1	Put an incompatible file in the path to test audio files (used a .PNG)	List in GUI should ignore it and the program won't crash as it won't try to convert it to .WAV	As expected	Pass
2	Place a .M4A or .MP3 into the file path	Files should be converted to .WAV	As expected	Pass
3	'Go to the path' button	Opens up file explorer at the specified path	As expected (On windows)	Pass
4	'Refresh' button	Any files in the path that can be converted to .WAV should be, and any that are incompatible are ignored	As expected	Pass
5	'Play' button	Stops any audio already playing and plays the audio for the file clicked on from the list	An error is thrown for some files Error: "Weird sample rates are not supported"	Fail
6	'Stop' button	Stops audio being played from the software	As expected	Pass
7	'Switch model' button	The next time an audio file is chosen, the prediction is made using the other model	As expected	Pass
8	On file click, are the words converted to string correctly	Words spoken should be accurately converted to a string	Most words are recognised correctly with some errors that generally don't detract from the meaning	Pass
9	Are both predictions always shown when a file is clicked	Two predictions are made in the software, one on audio signals and the other on the converted strings of the words spoken	As expected	Pass
10	Is a graph of the sound wave shown when a file is clicked	The graph is shown every time as no incompatible files should be displayed in the list	As expected	Pass

5. Chapter Five: Results, discussion, accuracy, and performance

5.1 Performance Metrics and Equations

Some important metrics to know when it comes to machine learning are:

- Precision
- Recall
- F1-Score
- Accuracy & Validation Accuracy

All of these metrics aim to do the same thing but in slightly different ways and have different ways of achieving that.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Figure 14

As you can see above from the equations for both precision and recall, they are both worked out in a very similar way. They use the number of true positives and negatives against the number of false positives and negatives to give a result that represents how many of the predictions were correct. Although very similar, they have a difference in purpose:

- Precision will evaluate to the proportion of positive predictions that were actually correct.
- Recall will evaluate to the proportion of predictions that were correctly identified.

This is where F1-Score comes in. Because precision and recall are subtly different, F1-Score aims to combine these two results to get something that is more representative of the actual performance of a model. Below is the equation for F1-Score.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Figure 15

Lastly is the accuracy and validation accuracy metric used by Keras. These are the values that will be shown with each epoch of a Keras model whilst training.

- Accuracy will show the accuracy of the model on the data it has already been trained on
- Validation accuracy (val_accuracy) will show the accuracy of the model on the training data (data that it has never been exposed to).

5.2 Audio model

The audio model was created using Keras in python and is an LSTM (Long-short term memory) model. The model was trained on the combination of the three datasets mentioned previously. Below shows the summary report of the model.

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 lstm_5 (LSTM)                (None, 123)               61500

 dense_15 (Dense)             (None, 64)                7936

 dropout_10 (Dropout)         (None, 64)                0

 dense_16 (Dense)             (None, 32)                2080

 dropout_11 (Dropout)         (None, 32)                0

 dense_17 (Dense)             (None, 7)                 231

=================================================================
Total params: 71,747
Trainable params: 71,747
Non-trainable params: 0
_________________________________________________________________
```

Figure 16

A 'Softmax' layer was used as the model had to learn and predict between 7 different labels as its preferred for multi-class classification.

The model was first trained using 80 generations just to see how it performed and grew. From the graph, it's possible to determine around where the model starts overfitting because the accuracy gain per generation started dropping off markedly. Below shows the accuracy graph created over the training epochs with markings.

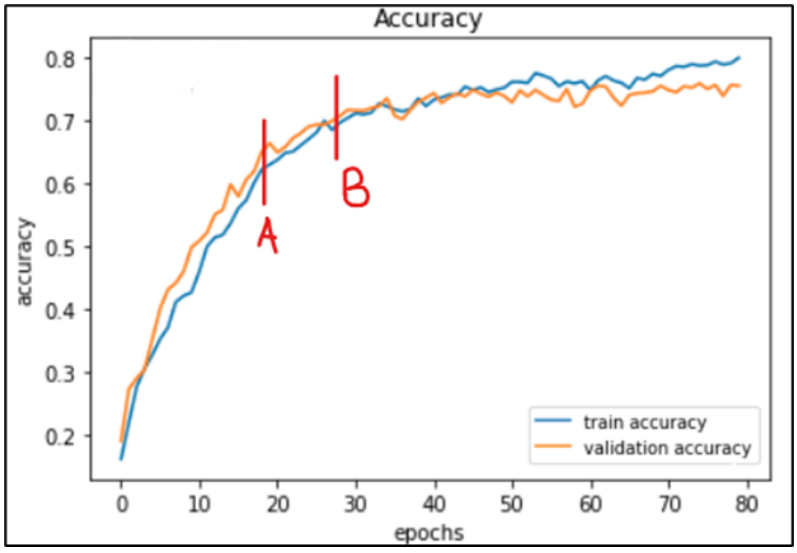


Figure 17

Between points, A and B in figure 17 is the ideal area to stop training the model. This is because anything under ~20 generations (Point A) would result in an underfitted model. Meaning that the model would perform badly for its purpose. And anything over ~30 generations (Point B) would result in an overfitted model, meaning the model would have a high accuracy whilst only being able to make predictions for the dataset it was trained on. This gives a false perception of it being good when it would perform poorly in real-world applications. Therefore, 30 generations opted for a good middle ground but this could be tinkered with for optimal results.

I have recreated this model multiple times already, with different amounts of MFCCs (features) as well as with different train/test splits, different units for the LSTM parameters, and even different optimizers. I'm quite consistently achieving ~70% accuracy and validation accuracy which is what I settled for, for the final model.

In the program that has been created, some audio files have been tested which will be available with the project download on GitHub. Some of these files were voiced personally as best as I can (bearing in mind I have no experience in voice acting so they may be poor in quality) and the Gordon Ramsey and Will Smith clips came from a soundboard site as I felt they voiced emotions very well although they contain vulgar language and the sound quality of the clips are not ideal. Below are the predictions from the model of my chosen sound files:

Filename	**File description**	**Intended/Expected Emotion**	**Predicted Emotion**
dont_know_how_to_feel-sad_neutral.wav	Made by me: "I don't know how I feel about it"	Neutral	Sadness
Gordan_Disgust.wav	From soundboard, not clean, has sound effects other than the speech (**includes vulgar language**)	Anger/Disgust	Disgust
imscared-disgust_fear.wav	Made by me: "Oh no I'm scared"	Fear	Disgust
no_dont_want-sad.wav	Made by me: "No I don't want that"	Sadness	Sadness
sogood-happiness.wav	Made by me: "That's so good"	Happiness	Happiness
will_smith_oscars.wav	From a soundboard and sound is slightly distant (**includes vulgar language**)	Anger	Anger
wow_amazing.wav	Made by me: "Wow that's amazing"	Happiness/Surprise	Surprise

5.3 Gendered Audio Model

This model was made after the first Audio model detailed above. I wanted to see if separating the clips by gender would yield a model with more accuracy. Luckily the datasets had enough information for me to programmatically separate them based on gender and this resulted in 14 labels. The rest of the process for creating the model was similar to what was mentioned above.

This resulted in a model with around 70% accuracy which seemed good until testing with real audio clips from outside of the dataset. Multiple audio snippets were then recorded of both myself as well as my colleagues and used the gendered model to label the audio clips. Every audio recording made by males was identified as female and the emotion was incorrect every time (from the emotion that was intended by non-professional voice actors).

It was noted from testing with these snippets that the model had a bias towards identifying things as female and it was concluded that this was due to the largest dataset (TESS) being composed of solely female actors. This meant that the model had a lesser understanding of the male voice. On top of this, 10 colleagues were asked to identify the gender of the speaker from one of the audio clips used in the dataset. The clip was by a female but 4/10 people incorrectly identified it as male. Albeit a small sample of 10 people, 40% of people getting it wrong shows that if 40% of humans are uncertain, then it can't be expected for an AI model that was trained on these voices to be consistent and accurate in its predictions.

Because of this, it was decided not to spend much further time trying to pursue this model as it would take a much larger dataset for the model to be remotely accurate meaning for diminishing returns in regards to the ultimate goal of speech emotion recognition. This isn't to say the model could not perform well, with the correct weighting of different datasets, and maybe more dropout layers being used to help the model fit more generally rather than overfit/underfit to the datasets given, then the model could be more accurate and precise resulting in a higher F1-score. Other changes like using a different optimizer instead of 'adam' may have had positive effects on the model.

Bearing its performance in mind, the model was still incorporated into the final GUI program to allow the user to get predictions from the gendered model too if they so wish, but it shouldn't be expected to yield desirable results.

5.4 Text Model

This is the final model created and was more experimental than the first two audio signal models. Since context is important in determining an emotion, it seemed appropriate to create a model that could complement the predictions of the others, and therefore created this text-based emotion model. This model was made using the sklearn machine learning tools rather than Keras and more specifically a RandomForestClassifier was used. The two figures below show a classification report created by the trained classifier as well as a confusion matrix.

	precision	recall	f1-score	support
anger	0.90	0.83	0.87	2293
fear	0.84	0.82	0.83	1863
happiness	0.84	0.95	0.89	5498
love	0.87	0.66	0.75	1420
sadness	0.92	0.92	0.92	4806
surprise	0.81	0.67	0.73	620
accuracy			0.87	16500
macro avg	0.86	0.81	0.83	16500
weighted avg	0.87	0.87	0.87	16500

Figure 18

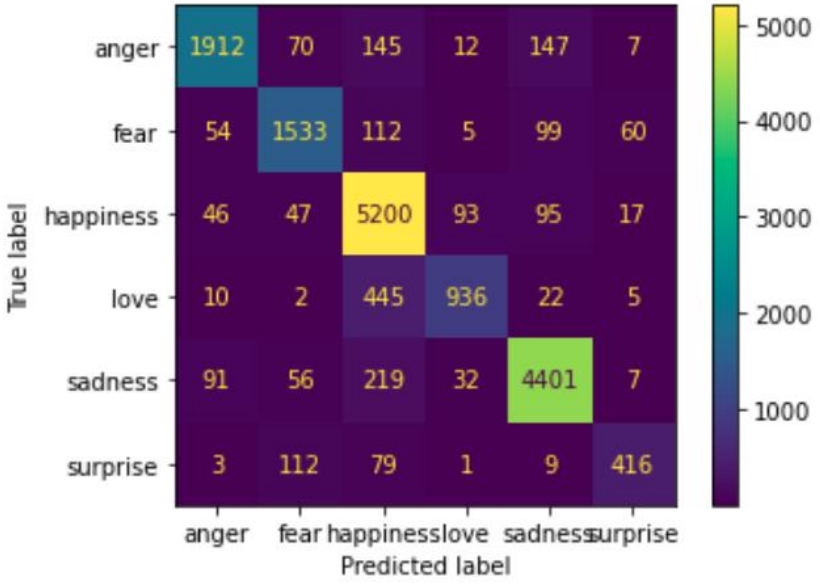


Figure 19

Due to the much larger dataset used for this model, the performance is immediately apparent through the metrics in the classification report, with the lowest F1-Score being 73% and the highest being 92%. The label that got 73% was the label with the least data points at just 620 so that would explain why the accuracy is lower. Even then, 73% is a respectable score.

The classification report in figure 18 gives enough information to make decisions such as removing the surprise label and all its data points from the training set as there isn't enough to train the model well. Or the weightings could be adjusted for this lower number of data points for the 'surprise' label.

The confusion matrix in figure 19 gives a good visual representation of how the model performs on the different emotion labels. From this, we can see that 'love' is commonly misclassified as 'happiness' which is understandable given the nature of the two emotions. What is strange is that 'sadness' is most often misclassified as 'happiness' which was unexpected given them being opposites but this could be because it's harder to tell in just text alone.

A prominent problem with this model comes from its reliance on the accuracy of speech recognition. Before a prediction is made on a file, the speech must first be converted to text using speech recognition. A python Speech Recognition library was used for this and although the exact percentage accuracy for the speech recognizer couldn't be found, it can be said with certainty that it's not 100% accurate. Even though the text model has an average of 87% accuracy, it will almost always perform worse than that because of the inaccuracy of the speech-to-text conversion. For example, if the speech recognition library converted to text with 80% accuracy and my model made its prediction with 87% accuracy, since an audio file would have to go through both, the accuracy would be 0.8*0.87 which would work out to 0.67 or 67% accuracy.

The table below will show the actual words spoken in 7 audio files and what the library recognises these words to be (the recognised words will be green if correct, orange if close to correct, and red if far off the actual words spoken):

Actual words spoken (as heard by me)	**Words recognised by the speech recognizer**
I don't know how I feel about it	I don't know how I feel about it
Oh no I'm scared	Oh no I'm scared
No, I don't want that	No, I don't want that
That's so good	So good
Wow, that's amazing	That's amazing
Keep my wife's name out your f****** mouth	Keep my name out of f******
inaudible it was your f***** kitchen then clean it you lazy c***. Now.	Cleaning your lady now

One thing to mention from this is that the two last sentences contain vulgar language and the string returned by the speech recognizer will be censored using asterisks (*). This can supposedly be changed with a parameter but I left it censored for it to be more appropriate given the academic context. The dataset also contained censored data points so it shouldn't detract too much from the model's accuracy.

6. Chapter Six: Conclusions and Further Work

6.1 Introduction

Thus far there has been a lot of research into machine learning, the different methods and algorithms available, and the way that machine learning works more specifically with audio files.

Using the research, a project has been made consisting of three models, two made with the use of Keras and the other using sklearn. These models have then been used to create a

working software with a GUI in python that allows the user to input any of their own files to be analysed by the models and be given a prediction on the emotion conveyed.

6.2 Accuracy

Both of the Keras models trained on audio signals had >70% accuracy, as well as a validation accuracy of closer to 80%, and the sklearn model has an overall F1-Score of 87% which is very good for a prototype.

However, this could be improved with the use of added datasets, more pre-processing to audio files, the use of different models/layers, etc. More specifically I'd like to:

- Use the k-fold algorithm to make better use of each datapoint since generally speaking, it is difficult to find suitable audio-emotion datasets.
- Create the audio signal models using image recognition techniques of the spectrograms of the files rather than using MFCCs as is currently in use.
- Look into ways of normalizing every audio file before using it for training.
- Adjust the weighting of certain emotions based on the number of data points I have for them.
- Create my own small dataset with the help of friends, family, and colleagues to add more of a real example of someone's emotion when training the models. This custom dataset would be added to the current dataset to expand it rather than using it exclusively.
- For the gendered dataset, adjust weighting based on gender since the dataset used had more female VA's (voice actors) than males creating a bias.

6.3 Root Cause Analysis

Root cause analysis (RCA) put simply is the process of finding the *root causes* of the issue being dealt with. RCA can be used in a variety of different contexts but in the case of this project, it could be used to find the reasons behind any issues such as underperformance of the model. For example:

- finding the root cause for why a model has low accuracy will allow a solution to be devised to increase the accuracy.
- Finding the root cause for misclassification can allow for steps to be taken to reduce the misclassification rate.

The below sections will point out and discuss some problems and the potential root causes.

6.4 Issues with Text Dataset

Whilst looking through the dataset being worked with; multiple issues were noticed, such as emotion labels that I would disagree with. The table below shows a couple of examples of text and their respective labels according to the dataset:

Text **(Written exactly the way each string is presented in the dataset)**	**Label**	**Comment**
i don t feel comfortable around you	joy	Clearly not joy

article published	joy	I'm not sure this text has enough information to be labelled as anything
im sick of not feeling safe in my own home	joy	Clearly not joy
i said when i got here i just feel like this is a special team echoed forward a href http www	joy	Not sure what the text is but I would not classify it as joy. Also the last words "a href http www" are probably an error with extracting text, these words could be further filtered out of the dataset
i feel accepted and be loved	joy	Since the dataset has a 'love' label, I feel that would be more appropriate but this isn't too bad of a mislabel
i do not feel jolly	joy	The negation doesn't seem to have been recognised

These are just a few examples found whilst scrolling through the entire dataset. It's to be noted that every example found happened to be labelled as 'joy'. This will be in part due to the 'joy' emotion making up the majority of the dataset but it can't be said for sure if the other emotions have the same mislabelling problem since they make up more of a minority of the dataset and also it would be very difficult to find every single mislabel partly due to the size of the dataset but also because the label that is given to a piece of text is subjective to the reader.

Another problem with this dataset is that there are a lot of abbreviations and slang being used in the dataset since the dataset was compiled using tweets from Twitter, although it can't be said for certain since it hasn't been found in writing that this is a Twitter dataset. The reason this is a problem is that; firstly, these abbreviations aren't being used in natural speech but only in text, and secondly because even if the speaker were to say the abbreviations, the speech recognizer will never pick up on the intended abbreviation and instead will look for real words through its connected dictionary. This means that the model is trained on a lot of abbreviations and slang that will never be used in the real world since my program extracts the text from speech. The figure below demonstrates an example of how this could happen.

Figure 20

6.5 Issues with Audio Dataset

The above-mentioned issues are solely related to the text dataset but there are also some issues with the audio datasets. Three datasets were used, and all of them used generic sentences for each emotion. An example from RAVDESS is "Kids are talking by the door" being a sentence spoken with multiple different emotions. It doesn't really make sense for this sentence to be used for an emotion like sadness but again this is subjective and doesn't impact the training of an audio model, it just meant that the text couldn't be extracted for the training of the text-based model but instead, a separate dataset was needed which wasn't too detrimental.

A more impactful issue is due to accents. The SAVEE dataset is from Surrey meaning the actors are British English whilst both the TESS and RAVDESS datasets are from Toronto in Canada. Whilst RAVDESS and SAVEE have quite neutral accents, I feel that the TESS dataset has a much stronger accent which makes the way different emotions sound very different. As mentioned prior in this report, a problem encountered because of this was the gross misclassification of genders using the gendered model. It seems because of this that it would be wise in the future to not only have separate models for different languages but also for accents that have very distinctive features to them. Something like a strong Scottish accent has already been shown to cause issues in speech recognition so for both speech recognition and speech emotion recognition separate models should be created that have been trained on people with different accents.

6.6 Image Recognition for Video Clips

For now, the audio model has been complemented with a text model which allows for contextual information. The natural progression from this would be to add functionality that would allow for extra contextual information in the form of image recognition of the speaker. This way, the image recognition could look for facial expressions that determine your emotion as well as the tone of your voice as well as the emotion relating to the words being spoken. Then ideally a mathematical function could be used to combine the predicted emotions giving each model a different weighting to find what emotion is most likely to be portrayed in the video clip.

Combining all of this contextual information with speech recognition could also help to improve the accuracy of current speech recognition. Current speech recognition already uses layers of context to come to its prediction, such as firstly attempting to identify each word, then checking if it's grammatically correct before looking for other potential sentences that may sound similar and maybe more likely to be correct. Adding this further emotional contextual data will therefore further refine the prediction.

6.7 Other Changes/Additions

One major feature that would improve upon my work is to allow the model to continue learning using input from the user. This would work by allowing the user to test their audio files using the software the same as it already does, but if the model guesses incorrectly, then the user can tell the program what the emotion in the file is actually portraying. The model can then use that experience to improve itself and increase its accuracy as well as its overall 'understanding' of emotions.

This can be further enhanced by creating a public website that is available to anyone with access to the internet. This will greatly increase the reach and with this addition, many more people will be contributing data to the program. As a result, the model will certainly improve in its predictions. An obvious problem with this is users entering audio clips that aren't of speech or intentionally mislabelling just to confuse and ruin the model so there would need to be a workaround for this such as having moderation for each response entered by a user or only allowing verified members of the community to correct and give feedback to the software regarding their prediction.

Another addition or change to my current project would be to allow longer audio clips, such as minute-long clips, and then let the user pick any section/chunk of the audio clip to run the model on. In the case of a dialogue audio clip, the file wouldn't need to be manually cut up into chunks and run separately, but instead, the software would allow the analysis of specific parts at a time to maybe differentiate between the emotions of both speakers.

6.8 Ethics

Like most things, ethics need to be considered. Especially since this project employs AI and machine learning, these are things people feel quite wary about already. Being able to identify the emotion someone is portraying is a very useful tool and will have so many different use cases but people may also feel that it is a violation of their personal space and that there is no need to collect or use data from their emotions for anything.

Especially in the case of marketing, if the marketing giants ever decided to use people's emotions like sadness to manipulate them into buying things, or even potentially marketing drugs or other fake remedies to make them happier, this would undoubtedly be a huge ethical and moral problem.

Another problem is in regards to bias. No matter what happens, a machine learning model will always have a bias based on the data points it has been trained on and this bias can be exploited to target certain groups of people.

These are all problems that need to be considered during the creation of a machine learning model, as well as during the deployment stage of a project such as this one.

Bibliography

Darwin, C. (1872) The Expression of Emotions in man and animal. http://darwin-online.org.uk/content/frameset?pageseq=1&itemID=F1142&viewtype=text
Accessed 3rd January 2022.

Okrent, A. (2012) How Do Computers Understand Speech.
https://www.mentalfloss.com/article/31609/how-do-computers-understand-speech
Accessed 8th January 2022

Popova, A., Rassadin, A., and Ponomarenko, A. (2017) Computer Can Recognize Emotions in Speech. https://www.technologynetworks.com/tn/news/computer-can-recognize-emotions-in-speech-294036
Accessed 8th January 2022

BBC. (no date) Features of waves.
https://www.bbc.co.uk/bitesize/guides/zc62tv4/revision/2
Accessed 11th January 2022

Grieve, P. (2020) Deep learning vs. machine learning: What's the difference?.
https://www.zendesk.co.uk/blog/machine-learning-and-deep-learning/
Accessed 11th January 2022

Prakash, S (2020) Traditional and Representational Machine Learning.
https://medium.com/@saiprakash513/traditional-and-representational-machine-learning-317495b74c1b
Accessed 11th January 2022

Lech, M., Stolar, M., Best, C., and Bolia, R. (2020) Real-Time Speech Emotion Recognition Using a Pre-trained Image Classification Network: Effects of Bandwidth Reduction and Companding. https://www.frontiersin.org/articles/10.3389/fcomp.2020.00014/full
Accessed 12th January 2022

C3. (no date) Model Training. https://c3.ai/glossary/data-science/model-training/
Accessed 12th January 2022

Rajbanshi, S. (2021) Everything you need to know about Machine Learning.
https://www.analyticsvidhya.com/blog/2021/03/everything-you-need-to-know-about-machine-learning/
Accessed 12th January 2022

The University of Surrey. (2015) Surrey Audio-Visual Expressed Emotion (SAVEE) Database. http://personal.ee.surrey.ac.uk/Personal/P.Jackson/SAVEE/
Accessed 12th January 2022

Livingstone, SR. and 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. https://zenodo.org/record/1188976#.Yd9nhf7P1jU
Accessed 12th January 2022

Giannakopoulos, T. (2015) pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis. https://github.com/tyiannak/pyAudioAnalysis
Accessed 13th January 2022

Harris, C.R., Millman, K.J., van der Walt, S.J. et al. (2020) Array programming with NumPy. https://numpy.org/
Accessed 13th January 2022

Chollet, F., and others. (2015) Keras. https://keras.io
Accessed 13th January 2022

Puthran, M. (2017) Speech Emotion Analyzer.
https://github.com/MITESHPUTHRANNEU/Speech-Emotion-Analyzer
Accessed 13th January 2022

Dupuis, K. and Pichora-Fuller. MK (2010) Toronto emotional speech set (TESS).
https://tspace.library.utoronto.ca/handle/1807/24487
Accessed 8th April 2022

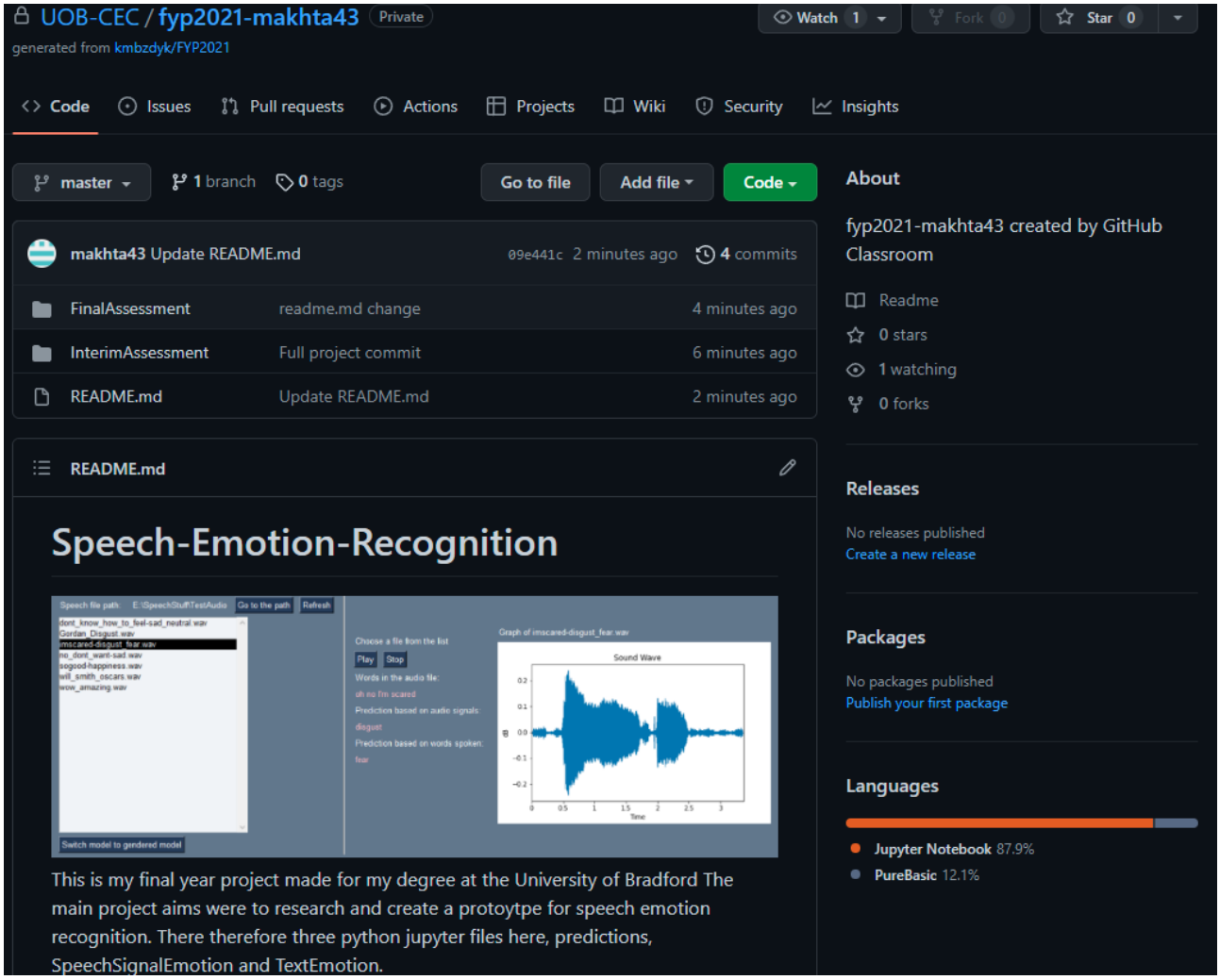
Saravia, E., Rodriguez-Cantelar, M. and Patil, S (2022) Dataset for Emotion Classification.
https://github.com/dair-ai/emotion_dataset
Accessed 22nd April 2022

7. Appendices

- Appendix A: GitHub Project Code

The code for this project as well as instructions on how to run it and links to the datasets required are all contained in the link below.

A1: https://github.com/UOB-CEC/fyp2021-makhta43



Screenshot **A2**: GitHub repository of project