Universiteit Leiden

Master Computer Science

Multimodal Machine Learning for Language/Speech Markers Identification in Mental Health

Name:	Georgios Drougkas
Student ID:	s2722135
Date:	[25/06/2024]
Specialisation:	Computer Science: Data Science
1st supervisor:	Prof. Dr Marco R. Spruit
2nd supervisor:	Dr. Erwin M. Bakker

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Mental health disorders constitute a significant challenge in healthcare and a lot of studies and advancements have taken place over the past two decades in order to find a way to identify such illnesses early and accurately. Through research and experimentation, researchers have significantly bolstered their capabilities in detecting and understanding mental health disorder markers. A lot of related work has been conducted already, including various modalities and mental illnesses. Most studies, however, involve either unimodal approaches on various mental disorders or multimodal approaches on a single mental disorder (most commonly on depression). Our goal is to experiment with both approaches on a longer list of mental health illnesses and discover whether modality fusion can lead to a better and more reliable identification of such markers. This study emphasizes on the integration of textual and acoustic data to show the strength of multimodal machine learning in the case of marker identification. For our experiments we use a well known and robust dataset derived from clinical interviews, E-DAIC. First, we constructed two unimodal models to analyze text and audio data independently using feature extraction, based on the mental disorder markers that had been identified earlier through related studies. Then, we employed an early fusion strategy to combine our text and audio features before model processing. Our fused features set was then given as input to various machine and deep learning algorithms, including Support Vector Machines, Logistic Regression, Random Forests and a fully connected neural network classifier (Dense Layers). Overall, the unimodal text models achieved an accuracy in the range of 80 to 85% and an AUC-ROC score between 85 and 93%, while the unimodal audio models attained 67 to 72% accuracy and 55 to 75% AUC-ROC scores. The experimental results indicated that our multimodal models achieved comparable accuracy (ranging between 80 and 87%) and AUC-ROC scores (between 84 to 93%) with that of the unimodal text models, but managed to outperform them in F1 scores and specifically in the F1 of the positive class (F1 of 1s), which reflects how well the models perform in identifying the presence of a marker. This study underscores the importance of multimodal integration in the realm of mental health diagnostics and sets the stage for future research to explore more sophisticated fusion techniques and deeper learning models.

Contents

1 Introduction

A large portion of globally reported diseases is constituted by mental health disorders. In 2017, 10.7% of the global population was reported having (at that time or in the past) a mental illness [1], while, according to the *World Health Organization (WHO)*, in 2019, 1 in every 8 people (i.e. about 970 million people) had a mental health issue [2]. Mental health turns into mental disorder when the coping mechanisms of an individual crumble and there is a significant disturbance in their cognitive skills, behaviour or emotional balancing. There is a broad range of mental disorder types and each of them impairs a different functioning area [2]. Although depressive and anxiety disorders are the most common mental illnesses, there are others equally important and affecting—including *PTSD, Bipolar Disorder, Schizophrenia* and *Eating Disorders*. Diagnosing mental health disorders has been challenging and a lot of studies have been conducted over the past decades. Unlike other types of diseases, mental illnesses are complex and are affected by multiple genetic, environmental and psychological factors. Previous research has found that people who suffer from some kind of mental disease use distinctive language patterns that we call mental disorder markers [1]. Finding a way to identify said patterns accurately and promptly could lead to a faster diagnosis and consequently to an earlier treatment.

Such distinctive patterns have been identified through various modalities, including transcripts of speech, audio and even facial reactions/movements. Most studies have been conducted using the text modality but more and more researchers have started to involve various other modalities in their experiments. In this paper we will study the text and audio modalities. Each of these modalities presents its own characteristics when it comes to identifying mental disorder markers and we categorize these characteristics into language markers and speech markers accordingly. Language markers may include the involvement of specific themes or the use of particular pronouns repeatedly, while speech markers are characterized by speech pattern alterations (e.g. pauses and speech articulation/rate) and variations in the voice (e.g. pitch variations, intensity, monotonicity, etc). During the last decade studies have been focusing on combining different modalities, which leads to a multimodal approach, in order to get the best elements of each modality and use them for the diagnosing of mental disorders.

Research Question

In our literature research we noticed that most previous works either worked with a single modality and a variety of mental illnesses, or they worked with multimodal approaches and instead they focused on one or two mental disorders. Our objective is to experiment with multimodality, specifically with the combination of text and speech, and find out whether merging the elements of these different modalities can lead to a better identification of various mental disorder markers. The majority of previous works with multimodality and mental illnesses has been focused on depression and PTSD. Instead, in this paper we studied the patterns that might be presented through a wider range of mental illnesses and included them in our experiments. More specifically, our research revolves around *Major Depressive Disorder, PTSD, ASD, Bipolar Disorder, Schizophrenia, ADHD, OCD, Anxiety Disorders* and *Eating Disorders*.

Methodology concept

Our methodology for this project involved various steps, with the first one being an extensive literature review that would help us identify any useful information pertaining to any previously identified mental disorder markers (either language or speech), any features that are crucial and relevant to this research and any models that have been reported to achieve good results on similar projects. The next step was to find a suitable dataset that would allow for all the following processes and experiments of this project. Then, using the *E-DAIC / DAIC-WoZ* dataset, we created three different models; each employing a different approach. The first model was built using the text modality and it was based a lot on a recent, relevant paper performing the same task [1]. Textual features, like the LIWC categorical scores and GloVe embeddings, were extracted and combined into a final features set that could be used for modeling. Similarly, the second model was built using the audio modality and another set of extracted features was created, involving audio features like formants, pitch and MFCCs. For both of these two unimodal models the final features sets were created according to a 'wrapper' method that eliminated features based on scoring and in the end selected the top performing features. Finally, the third model entailed the integration of the previous two models, using early fusion. The top features from each modality were selected and concatenated into a new features set. Then, for all of the aforementioned models, we experimented with the same classifiers, namely SVM, Random Forest, Logistic Regression and a fully connected neural network. To evaluate the performance of each approach we used three metrics; accuracy, AUC-ROC score and F1 score, which are important metrics that were used in previous, relevant studies [7, 8, 14, 15, 22].

Contribution

Our contributions involve:

1. An extensive review of language and speech markers of various mental health disorders. This paper provides an analytical reference over various mental illnesses and possible identifiers and diagnosing indicators, which can be used in future studies to quickly and efficiently map markers with disorders.

2. An extensive feature extraction process for each modality, featuring more than 150 textual features and 160 acoustic ones. Of course, in these numbers we are also including the derived statistical features (post aggregation), like mean, median and standard deviation. The extraction of this high number of features can be seen both positively (as it provides a comprehensive representation of the data) and negatively (since it can also lead to overfitting and complexity). In order to prevent overfitting and to manage any computational complexity, we employed various techniques, like feature selection and dimensionality reduction.

3. A comparison between unimodal approaches and a multimodal approach, in the context of identifying mental disease markers, employing four different machine learning models and three different evaluation metrics. The evaluation of these models was performed on the DAIC-WoZ dataset and over various mental health disorders, including Depression, Anxiety Disorders, Bipolar Disorder, Schizophrenia, PTSD, ASD, Social Anxiety, OCD, Eating Disorders, ADHD and BPD.

The code for this project is available at: GitHub Repository.

2 Background

2.1 Related work

Dataset

Among the multimodal datasets mentioned throughout related work, the most popular ones were CMU-MOSEI, CMU-MOSI, IEMOCAP, ICT-MMMO and MOUD [6, 11]. CMU-MOSEI is the only one suitable for both sentiment and emotion classification tasks. The main disadvantage of all these datasets is the fact that they are not connected to the topic of mental health at all.

Aleem et al. in [8], discuss various databases related to depression, specifically AVEC2013, AVEC2014 and DAIC-WOZ. They mention how the existing databases consist of two or three modalities and that DAIC-WOZ comprises three modalities (audiovisual and text), although the original videos haven't been provided. DAIC-WOZ has been used a lot for the identification of depression and it is preferred for its relevance with mental health disorders, despite its limited target focus [24, 25, 26]. DAIC-WOZ supports the investigation of anxiety, depression and PTSD. The authors of paper [18] utilize the DAIC database for a depression detection task using a deep learning approach. Their study presents a robust method for depression detection using audio and text features, which shows promising results on the DAIC-WoZ dataset.

Features

Literature helped us a lot to narrow down which features to extract. In the case of the text model a lot of papers recommended Linguistic Inquiry and Word Count (LIWC) as a feature, especially in researches relevant with mental health or detection of various mental illnesses [3, 5, 7, 8, 23]. Calvo et al. (2017) also emphasized on the importance of feature selection when there is a large features set. Without reducing the input space (i.e. the number of features), classification algorithms are not as efficient and suffer from overgeneralization. On that note dimensionality reduction using Principal Component Analysis (PCA) was recommended [5, 22]. GloVe vectors constitute another popular text feature that appears in a number of papers in our literature [4, 21].

In the case of the audio models, there were a lot of studies revolving around Mel Spectograms, pitch and MFCCs [11, 16, 17, 20, 25]. In their paper [6], Zadeh et al. use GloVe embeddings to extract word vectors from transcripts, and 12 MFCC coefficients and pitch for the acoustic model. Yin et al. performed a unimodal audio analysis on the prediction of psychosis and their feature extraction was focused on pitch, MFCCs (13 coefficients), vowel space, voice quality and change in rhythm [16]. These acoustic features are popular for speech / emotion recognition tasks.

Models

Support Vector Machine (SVM), Logistic Regression and Random Forest are among the most commonly selected (supervised) machine learning models when it comes to classificationand are often used as baseline models in mental illness detection [3, 7, 8, 9, 15, 20, 23, 25]. The authors of paper [9] provide a comprehensive review of various machine learning algorithms for diagnosing mental health disorders. In their paper they make a reference to previous related work on the same field and they include in their own research a table that lists various modeling approaches in correlation with various mental illnesses. The table illustrates how previous papers experimented with SVM and Random Forest for the diagnosis of PTSD, Schizophrenia and ASD [8, 9, 15]. A similar study was performed by Assan et al. who focused on the detection of depression by exploring a big range of machine learning classifiers. These included the three main aforementioned classifiers, as well as KNNs, XGBoost, AdaBoost, Decision Trees and Artificial Neural Networks [23]. In *'Acoustic speech markers for schizophrenia-spectrum disorders: A diagnostic and symptom-recognition tool'*, Boer et al. perform a unimodal (speech) study with the goal of identifying markers that indicate schizophrenia and factors that can cause it. For the classifier they chose Random Forest and for evaluation they selected AUC-ROC [14]. According to Cho et al. (2019), SVM and RF clearly outperformed simpler models like Naïve Bayes and KNN during this diagnosis. The authors claim that the SVM model has been employed before for all domains in mental health and it has been revealed that it normally achieves more that 75% accuracy. They also concluded that in cases of uncertainty, multiple classifiers should be tested on the data, and the best-performing algorithm should be selected through cross-validation [9]. Yazdavar et al. provide a table that showcases the performances of various models towards the identification of depressed users. Among their selected models, SVM and Logistic Regression are used for both modalities and Random Forest for the video modality [3]. Chung J. and Teo J. provide a review on a 30 papers long literature related to mental illnesses and classifier approaches. They discuss and sum up the approaches of all those papers and they come to the conclusion that in the case of machine learning, the Random forest and SVM models were the most popular choices due to their high performance in accuracy. Moreover they mention that most common performance metric is accuracy, followed by AUC-ROC and F1 scores [15] Amanat et al. also explore the topic of depression detection, utilizing deep learning techniques. All evaluation metrics (accuracy, F1 score, precision and recall) achieved the highest accuracy through their proposed one-hot LSTM model, surpassing models like SVM and Decision Trees with TF-IDF.

Based on this literature, we chose to proceed with SVM, Logistic Regression and Random forest, as the main machine learning methods, for our study. The reason behind this choice is their consistent high performance in accuracy and reliability across various mental disorder diagnosing tasks, as well as their widespread approval as baseline models. Moreover, the aforementioned models provide a balance between simplicity and effectiveness and this balance renders them more suitable, for initial experiments, than other more complex methods like XGBoost and LSTMs.

Fusion

Papers [3] and [4] present the creation of models which exploit early fusion between the text and video modalities; the first one in the topic of multimodal mental health analysis in social media and the latter one for multimodal classification through social media analysis. According to the authors of [3], not only is early fusion less computationally expensive than late fusion, but also their model reduces the learning effort and has shown promising results. In paper [26], the authors investigate both early level and model level fusions in the context of using deep neural networks for diagnosing depression, adding value to our own approach in this research.

Google's Gemini

When discussing in the field of multimodality, one cannot avoid mentioning the foundation model that has gained extreme popularity recently. And this is none other than Google's *Gemini*. Gemini is Google's most capable AI built specifically for multimodality, involving speech/audio, text, images, videos and even scripts/code [45]. It is built upon the architecture and concept of a Generative Pre-trained Transformer (GPT), similarly to OpenAI's *ChatGPT*, but it also introduces great advancements, including its multimodality capabilities. Gemini can leverage its foundational LLM (Large Language Model) architecture to process any modality and also to understand and generate responses in any of the aforementioned modalities. This versatility allows it to be applied to a wide range of areas and with the next advancement it could potentially be applied on the medical fields. Gemini could possibly be pre-trained further in a way that it becomes able to diagnose mental health disorders or identify mental disorder markers by itself.

2.2 Identifying Language Markers

In the following table (Table 1), we are presenting our findings in regards to markers that can be identified from textual data. The table doesn't include all the mental disorders that have been identified in the world, as this would require a huge research and a lot of experts of the field. Instead, just like our research, it focuses on the most common and popular mental disorders and the markers that can possibly indicate towards them. These language markers were gathered with the help of related literature and previous works. Specifically for the case of textual data, there was a lot of influence from paper [1], which also identifies such markers from text and it can be considered a directly related previous work.

2.2.1 Detecting the most prominent examples per mental disorder

Language Markers		
Mental Disorder	**Marker / Identifier**	**Source**
Depression	Negativity / Frequent use of negative emotion words	[18, 22]
	Focus on sad themes	[15, 20, 22]
	Hopelessness / Suicidal thoughts	[15, 20, 22]
	Lack of interest	[15, 18, 20]
	Discussion of past depressive episodes	
	Self-focused (frequency of first-person singular (I, me, my))	[23]
Anxiety Disorders	Repetitive worry themes	[15, 20]
	Focus on physical symptoms of anxiety	[20]
	Word repetitions	
Bipolar Disorder	Alternating between themes of elation and depression	[15, 20]
	Rapid thematic shifts	[20]
	Grandiosity during manic phases	
	More words related to death	[1]
	Self-focused language	[1]

Continued on next page

Continued from previous page

Mental Disorder	**Marker / Identifier**	**Source**
Schizophrenia	Disorganized thinking	[14, 16, 20]
	Incoherence	[15, 16, 20]
	Neologisms/made-up words	[20]
	Tangential speech	[20]
	Usage of words related to religion and hearing voices and sounds	[1, 15, 20]
PTSD	Avoidance language	[28]
	Trauma-related keywords	[15, 27]
	Storytelling that may be disjointed	[27]
	Difficulty finding words or describing experiences	
	More singular pronouns	[1]
	Words related to death	[1]
ASD	Literal language use	[30]
	Difficulty understanding sarcasm of figurative language	[30]
	Repetitive language patterns	[30]
	Linked to motion, home, religion and death features	[1]
	Self-focused / More 1st person singular pronouns	[1]
Social Anxiety	Language centered on fears of social judgment	[15]
	Self-consciousness	[32]
	Avoidance of social interaction topics	[15]
OCD	Words related to anxiety and cognitive words	[1]
	Focus on specific themes (e.g. cleanliness, order, etc)	
	Self-focused (use of "I")	[31]
Eating Disorders	Words related to the body	[1]
	Negative emotive words	[1]
	Self-focused words and cognitive process words	[1]
ADHD	Difficulty staying on topic / frequent subject changes	
	More 3rd person plural pronouns	[1, 31]
	Fewer relevant words	[1]
BPD	More swear words and words related to death	[1]
	Fewer cognitive emotive words	[1]
	More 3rd person singular pronouns	[1]

Table 1: List of identified language markers.
PTSD stands for Post-Traumatic Stress Disorders, ASD for Autism Spectrum Disorders, OCD for Obsessive-Compulsive Disorders, ADHD for Attention Deficit Hyperactivity Disorders and BPD for Borderline Personality Disorders.

2.3 Identifying Speech Markers

Just like with the case of the language markers, Table 2 illustrates the most prominent examples of speech markers that we identified through literature review and previous related work. Once again, as can be observed by looking at the table, only the most popular mental disorders and their respective markers are presented.

2.3.1 Detecting the most prominent examples per mental disorder

Speech Markers		
Mental Disorder	**Marker / Identifier**	**Source**
Depression	Slowed speech / Slower speech rate	[19, 20, 23]
	Decreased fundamental frequency f0 and f0 range / Reduced pitch variability	[19, 20, 24]
	Flatter affect and monotone pitch	[19, 20, 24]
	Longer pauses	[20, 23]
	Hesitant, abrupt, low-volume speech	[13, 20, 24]
Anxiety Disorders	Speech modulation issues	[20]
	Increased pitch	[20, 24]
	Faster speech rate in anxious states	[20]
	Possible vocal/shaky tremors	[24]
	Higher jitter and shimmer	[24]
Bipolar Disorder	Increased speech rate and volume during manic episodes	[20]
	Lower pitch during depressive episodes	Depression marker
	Slower speech during depressive episodes	Depression marker
	Increased tonality, median f0, mean F1 and F2	[13, 24]
	Higher number of longer pauses in depressive states	[13]
	Changes in vocal patterns based on affective state change from euthymia to (hypo)mania	[20]
Schizophrenia	Slowed speech	[13, 14, 20, 24]
	Reduced pitch variability / Lower f0 mean*	[13, 14, 20, 24]
	Increased number of pauses (clause-initial)	[12, 13, 14, 16, 20, 24]
	Decreased syllable timing variability	[13, 20]
	Lack of semantic cohesion	[14, 20]
	Less variation in jitter (i.e. roughness of the voice)	[14]
	Specific qualities in vowel sounds (e.g., breathiness and rounding)**	[14, 16, 20]
	Lower MFCC coefficients' scores	[16]
PTSD	Speech hesitations	[33]
	Alterations in speech pace	
	Possible changes in voice quality during discussions of trauma	
	Monotonous, slower, flatter speech	[15, 24, 33]
	Decreased f0 variability	[24]
	Reduced tonality in the vowel space	[24]
ASD	Atypical prosody (Core marker)	[29]
	Unusual rhythm and pitch	[29, 30]

Continued on next page

Continued from previous page

Mental Disorder	Marker / Identifier	Source
	Inappropriate shifts between loud and quiet speech	[29]
	Might have a monotonous tone	[29, 30]
Social Anxiety	Increased mean f0	[20, 24]
	Increased pitch, especially in social situations	[20]
	Faster speech rate	Anxiety marker
	Possible stuttering or speech disturbances	[32]
OCD	Repetitive speech patterns	
	Higher jitter	[24]
	More hoarse and breathy voice	[24]
	Lower speech rate	[24]
Eating Disorders	Higher jitter, shimmer, frequency disturbance ratio, amplitude disturbance ratio (Bulimia Nervosa)	[24]
	Vocal fold edema and polypoid changes due to vomiting (Bulimia Nervosa)	[24]
	Higher mean f0 before menarche (Anorexia Nervosa)	[24]
	Weak, asthenic voice, and some hyperfunctional dysphonia in prolonged cases (Anorexia Nervosa)	[24]
ADHD	Rapid speech rate	
	More sentences	[34]
	Louder speech	[34]
	Fewer clauses per sentence	[1]

Table 2: List of identified speech markers.
**Pitch variation is directly related with the range of F0. Smaller F0 range means less pitch variation, which means monotonous of flat speech.
***"Indicated by reduced variation in F3 formant frequencies, a lower F1 formant frequency and a smaller F1 and F2 formant bandwidth (i.e. reach of the formants, affected by jaw/mouth opening as well as tongue and lip positioning", as mentioned in paper [14]

3 Dataset

For this project we searched for a dataset that is both mutlimodal (involving audio and transcripts, especially) and also related to the context of mental disorder markers and their identification. As part of our experiments we went through multiple datasets, most of which were mutlimodal but missing the right context and content.

Before discussing on our main dataset, let us make a brief illustration on the most noteworthy datasets we experimented with during our research. Namely, they are the *CMU Multimodal Opinion Sentiment and Emotion Intensity (CMU-MOSEI)*, the *CMU Multimodal Corpus of Sentiment Intensity (CMU-MOSI)* and the *Persuasive Opinion Multimedia (POM)* datasets and they are available here: (`http://multicomp.cs.cmu.edu/datasets/`).

In the *'Modalities'* column, t stands for text, v for video and a for audio.

Dataset List

Dataset	Size	Modalities	Tasks
CMU-MOSI	2.199 videos, 98 speakers	t, v, a	Sentiment Analysis
CMU-MOSEI	23.500 videos, 1.000 speakers	t, v, a	Sentiment Analysis, Emotion Recognition
POM	1000 videos	t, v, a	Sentiment Analysis, Emotion Analysis

3.1 DAIC-WoZ Dataset

After experimenting with various datasets and searching through related work for a proper one, we selected the DAIC-WOZ dataset. This dataset appeared to be the most suitable one for our project and it was also the one that helped us proceed with the particular research question the most inclusively. The selection of this dataset allowed us to compare our methods to the current State Of The Art (SOTA) techniques. DAIC-WOZ stands for *Distress Analysis Interview Corpus - Wizard of Oz* and it is a set of clinical interviews designed to help diagnose mental health disorders like depression, post-traumatic stress disorder (PTSD) and anxiety.

As a matter of fact the DAIC-WOZ dataset was specifically designed to help with projects very similar to this one. The interviews included in this database were meant for a larger project, which aimed to identify verbal and nonverbal markers of mental disorders using a virtual agent as the interviewer.

The dataset includes interviews taking place between a human-controlled, virtual interviewer called 'Ellie' and a number of real participants. Three modalities are present in this database, namely transcript, audio and video. Facial features, voice and text information of the participants are available through this dataset, which has also been labelled using a 'Patient Health Questionnaire (PHQ-8)' survey (https://pubmed.ncbi.nlm.nih.gov/18752852/). The particular questionnaire consists of 8 questions (as implied by its name) and it aims to identify the presence of depression. As such there is a binary labelling system that classifies between the existence or absence of depression. More precisely, a PHQ-8 score of less than 10 indicates a mentally healthy patient, while a score greater than 10 indicates a patient with depressive symptoms (worsening the higher the score is).

DAIC-WOZ contains 189 interviews, one interview per participant, and for each participant there is a recorded voice interview with an average length of 16 minutes. To be more detailed, the minimum and maximum voice recording lengths are 7 minutes long and 33 minutes long, respectively. All audio files are sampled at a rate of 16kHz, which proved to be really advantageous for this research. The 16kHz sampling rate is ideal not only because it suits our specific needs but also because it eliminates the necessity to standardize the sampling rates of the audio files ourselves. Standardization in this context refers to ensuring consistency in the sampling rates across all audio files. This uniformity is crucial as it allows both the pre-processing steps and the feature extraction processes to be conducted on data with a consistent structure.

Regarding the DAIC-WOZ dataset, one can acquire it by applying for it through the official website, which falls under the ownership of the University of Southern California (https://dcapswoz.ict.usc.edu/). Once given approval for download, one has access to a variety of files, including the transcripts, voice recordings, facial features, pre-extracted formant and covarep features, as well as the original *train-test-split* files. The original approach for this data split was having 107 out of the 189 interviews used as training data, 47 as test data and 35 as verification data.

Due to technical reasons, we had to remove one of the interviews (along with its text and audio files). To be precise, the audio of the particular interview was full of noise, to the point that we couldn't improve it even with extreme noise reduction. The noise seemed like a microphone sensitivity issue and there were barely any clean speech parts overall. As such, although this project followed the same *train-test-split* approach, it actually focused on 188 out of the 189 interviews.

4 Feature Engineering Methodology

4.1 Text Features

When it comes to textual features, we experimented with a variety of them in our attempt to find the most suitable ones for the particular research. Among these features, the ones that made the most effective and interesting impact in regards to our objective were extracted using:

- *Linguistic Inquiry and Word Count (LIWC) analysis*
- *K-means clustering* (applied on LIWC results)
- *GloVe Embeddings*
- *POS-Tag counts*

These are described in more detail in the following subsections.

4.1.1 Feature Selection

4.1.1.1 Analysis of selected text features

For the textual model, directions were given based on paper [1], which also aimed to identify language markers from text data. As such, LIWC was chosen as an important feature of this process. More specifically, *Linguistic Inquiry and Word Count* is an easy to use software tool created with the overall purpose of analyzing word use, on semantic, emotional and syntactic levels. In the case of the unimodal text model approach, LIWC, by itself, could be enough to perform an analysis for the identification of language markers. LIWC provides categorical scores based on predefined dictionaries and it offers linguistic and psychological context of words.

K-means clustering is an unsupervised machine learning approach and its objective is to distinguish groups of text segments or participants who show similar linguistic characteristics, so that the components of each group are more comparable to one another and as different as possible from the components of other groups. In short, it's about distinct grouping based on similarities and differences. In this study, the clustering is applied to the numerical data of the

LIWC results, in order to help us identify, and consequently illustrate, language patterns. The reason that clustering was chosen to represent a more 'theoretical' feature, along with the 'practical' LIWC, is the fact that it can allow us to discover the most related LIWC categories per cluster. This can help distinguish the linguistic and psychological characteristics of each cluster and provide us with a direction as to which cluster, for instance, tends more towards negative / positive emotions or includes more specific themes like 'religion' or 'socialization'.

GloVe (Global Vectors for Word Representation) embeddings, which are pre-trained in large corpora, are a great means to discover and understand the semantic meaning behind word use. To achieve that, these embeddings map the words or even sentences into high-dimensional vector spaces. The GloVe embeddings encode words with similar context in a way that their vectors are located close to each other in the vector space [35]. For instance, words like 'sadness' and 'unhappy'. Furthermore, they are especially useful when working with machine learning models, since they can improve the model's performance or perform dimensionality reduction. So, in the context of identifying language markers for mental disorders, GloVe embeddings can be used to perform a more nuanced and morphological analysis of the used words.

The last selected features for our text feature set is the POS-Tag counts and more specifically the pronoun counts. POS-Tag features provide insights into the roles and grammatical patterns of words in a given text. The inclusion of particular POS Tags, which are mentioned in the *Feature Extraction* subsection and counts of first person singular, third person singular and third person plural pronouns can offer interesting insights into the focus of the speaker/participant. Moreover, the latter has already been proven to be an indicator towards various mental health disorders (e.g. frequent use of first person singular is a language marker indicating depression or bipolar disorder). LIWC includes the "I", "they" categories as well, but unlike LIWC, POS tagging offers a syntactic analysis, categorizing words based on their parts of speech and can even extend our analysis to verb tenses or other parts of speech.

4.1.2 Data pre-processing

4.1.2.1 Linguistic Inquiry and Word Count (LIWC)

Given the structure of the transcripts available by the DAIC-WOZ dataset, where the speaker is identified in the 'speaker' column (can be either "Participant" or "Ellie", the virtual agent), we modified the script so that it selectively kept only the responses given by the participant. All the questions asked by the interviewer and any other exclamations were disregarded from the newly created text files. This was considered an important pre-processing step for the LIWC analysis, as our overall goal is exclusively focused on the mental health and language of the participants. Another pre-processing step was the concatenation of all the responses of a single participant in a single string; this way we ended up with 189 strings with an average length of 7306 words. At this point, the processed and cleaned text files were ready to become input for the LIWC software.

4.1.2.2 K-means clustering

Before performing k-means clustering, we performed the elbow method to determine the optimal number of clusters for our data [36]. So, given a range of cluster numbers, this method plots the sum of squared distances of samples to their closest cluster center. Then, one has to look in the plot and identify that point in which the decrease acutely changes; pretty much like looking at an elbow at a ninety degree hold. The particular point is the one that most often represents the optimal number of clusters. After performing this method, the result indicated that the most suitable number of clusters is between 4 and 5, where the inertia showed the smallest decrease. Experimenting with less than 3 and more than 5 clusters would definitely prove less insightful.

4.1.2.3 GloVe embeddings

In the case of text pre-processing for the GloVe embeddings there were three main steps. First, english stopwords were removed. Then, the text was tokenized and finally it was transformed to lower cases, since GloVe embeddings are actually case sensitive. All of these steps aimed at achieving text uniformity before proceeding with the extraction of these features.

4.1.3 Feature Extraction

4.1.3.1 Linguistic Inquiry and Word Count (LIWC)

The result of the analysis performed by the LIWC software was a text file with scores over all of the available categories provided by LIWC. We inputted all of our cleaned texts to the software simultaneously and as a result we got a single file with distinct interviews as the rows and categorical scores as the columns. However, as part of the research we were mostly interested in specific categories, which are more connected with our objective of identifying language markers. To manage this, we tried to map various mental health disorders' language markers, that we identified, with relevant LIWC categories. We also kept in mind that there may be language markers that align with multiple categories or even some without a perfect fit, since LIWC offers categories with a broad nature. In the following table you can see some interesting examples of this mapping process.

Note that Table 3 is not a complete table of all the mental disorders and all of their respective language markers that we identified. Our research included some extra mental health disorders, like OCD and social anxiety, and there are many other markers that are not (easily) mappable to LIWC categories, which is the reason why they were excluded from the table. We still included a couple of examples corresponding to '<Not mappable>', but they are there for illustration purposes. Regardless, the above table presents most of the categories that we focused on extracting before moving to modeling. The rest of the LIWC categories, which are not shown in the table, but were still present in our research are *anx, we, you, they, cogproc, bio* and *relativ*.

4.1.3.2 K-means clustering

The k-means clustering was implemented using the *KMeans* method of the *sklearn* library (*from sklearn.cluster import KMeans*). Although clustering was explained earlier as well, we need to refer to the process again for comprehension purposes. So, each of the created clusters in our case represents a group of texts that have similar LIWC metrics with each other. Setting forth the average values of these metrics, we could deduce the dominant characteristics of each cluster's text segments. At this point, the extracted information can be used for both comparative analysis (for instance, noticing that a cluster has significantly higher scores in

LIWC Categories Mapping		
Mental Disorder	Language Marker	LIWC categories
Depression	Negativity	sad, anger, negemo
	Frequent use of negative emotion words	negemo
	Lack of interest	leisure, work, and social
	Expressions of guilt	guilt
	Discussions of past depressive episodes	focuspast
Anxiety Disorder	Repetitive worry themes, excessive concern about future events	focusfuture
	Word repetitions	<Not mappable>
Bipolar Disorder	Alternating themes of elation and depression	posemo, negemo
	Jumping from topic to topic	<Not mappable>
	Self-focused language	I, me, my
Schizophrenia	Disorganized thinking, tangential speech	cogproc
	Using words related to religion and hearing voices	relig
PTSD	Difficulty finding words or describing experiences	Word Count
	Using more singular pronouns and words related to death	I, me, my, death
ASD	Linked to motion, home, religion, death features	physical, home, relig, death
	Self-focused (more 1st person singular pronouns)	I, me, my

Table 3: Mapping of LIWC categories

posemo (positive emotions) can mean that it has a more positive tone compared to others) and for contextual understanding (i.e. drawing hypothesis or identifying patterns based on what the clusters represent). It should be obvious at this point that clustering, as a feature, for this particular research is really beneficial and can help a lot in identifying patterns and consequently language markers.

4.1.3.3 GloVe embeddings

The whole procedure of extracting GloVe embeddings, as a feature, included four steps. First, we had to download a pre-trained GloVe model. Visiting the official website [35] one can find many versions, based on the dataset in which the GloVe embeddings were trained on and the size of the word vectors. Since most of the coding took place from a laptop with an *i7 8th gen processor and a RAM of 16 GB*, it was deduced that the 100-dimensional (100d) GloVe model would potentially be the most optimal choice. Not only does the 100d model offer a good balance between providing sufficient semantic detail for nuanced text analysis and being computationally efficient (not intensive like the higher-dimensional models), but it also provides enough depth in the word embeddings without overfitting. Just for reference purposes, one can also work with the 200d model if they can support the extra computational load and if they are not satisfied with the capabilities of the 100d model.

The *Wikipedia 2014 + Gigaword 5* GloVe model was selected as the pre-trained model. A wide range of language use, along with various styles (formal and informal), are included in this combined corpus. More particularly, Wikipedia articles cover a huge number of topics and Gigaword 5 helps to add value to the diversity in topics. All this diversity in style and topics is crucial in 'catching' embeddings with terms relevant to mental health contexts.

The next step was to load the GloVe embeddings into python by creating a dictionary, where the keys are words and the values are the corresponding vector representations. The third step of this feature extraction was to pre-process the textual data, which was explained in earlier sections and finally the last step was to convert our text to GloVe embeddings. Having performed tokenization, for each token in our text data, we found the corresponding GloVe vector (through an automated process of course) and used them to represent our text.

4.1.3.4 POS-Tag counts

For this research we decided to work with a feature-rich approach. So, to complement our feature set we also extracted some information on relevant POS-tags. More specifically, we applied tokenization and part-of-speech (POS) tagging in the cleaned text files, using the *nltk* package. Through the *nltk* package, we downloaded the necessary components for tokenization (i.e. *punkt*) and POS tagging (i.e. *averaged_perceptron_tagger*), which constitute fundamental NLP tasks. The following table presents the relevant POS-tag counts that we selected.

POS-Tag	**Representation**
PRP$	Possessive pronoun (e.g. my, your, his, her, its, our, their)
NN	Noun, singular or mass (e.g. dog, bike, hope)
NNS	Noun, plural (e.g. dogs, bikes, hopes)
NNP	Proper noun, singular (e.g. George, Amsterdam, IBM)
NNPS	Proper noun, plural (e.g. Americans)
VB	Verb, base form (e.g. write, live, run)
VBD	Verb, past tense (e.g. wrote, lived, ran)
VBG	Verb, gerund or present participle (e.g. writing, living, running)
VBN	Verb, past participle (e.g. written, lived, run)
VBP	Verb, non-3rd person singular present (e.g. write, live, run in "I write", "we live", "they run")
VBZ	Verb, 3rd person singular present (e.g. writes, lives, runs in "he writes", "she lives", "it runs")
JJ	Adjective (e.g. happy, sad, large)
JJR	Adjective, comparative (e.g. happier, sadder, larger)
JJS	Adjective, superlative (e.g. happiest, saddest, largest)
RB	Adverb (e.g. quickly, not, very, there)
RBR	Adverb, comparative (e.g. faster, better, more)
RBS	Adverb, superlative (e.g. fastest, best, most)
UH	Interjection (e.g. uh, ah, oh, ouch, hi, hello)

Table 4: Relevant POS-Tags

Furthermore, apart from the relevant tag (Table 4) counts, we also extracted counts for first-person singular, third-person singular and third-person plural pronouns. See Table 5 to understand what is included in these counts.

Pronoun	**Includes**
1st SG	'i', 'me', 'my'
3rd SG	'he', 'she', 'it', 'his', 'her','hers'
3rd PL	'they', 'them', 'their', 'theirs'

Table 5: Additional counts

4.2 Audio Features

For the unimodal audio model there was a larger pool of features, related to our objective, to choose from. After consulting with our literature and gathering the most prominent features used in mental health disorder related projects, we concluded in the following speech/audio features:

- *Pitch*
- *Jitter (local, ppq5 and abs)*
- *Shimmer (local and apq5)*
- *Harmonic-to-Noise-ratio (HNR)*
- *Mel-frequency cepstral coefficients (MFCCs) (13 coefficients)*
- *Energy*
- *Formants*

4.2.1 Feature Selection

4.2.1.1 Analysis of selected speech features

As it was just discussed, the feature selection for the audio model was based on the proposed features throughout the literature and the related work. Among all those features, pitch was one that was present in about 90% of the cases. It has been described as being a good indicator in mental health disorder identification. The fundamental frequency f0 mean, which indicates the lowest frequency of the speech signal is perceived as pitch (mean, median). Pitch is closely related to the frequency of sound waves and is measured in Hertz (Hz). In the specific research, pitch was also selected because it is helpful when it comes to identifying emotional states or stress.

Jitter and shimmer usually go as a package and they were selected through various studies (e.g. paper [24]) as suitable features for the classification of various mental diseases. There are more than one options for jitter and shimmer values. However, for this project we selected the following ones.

- *Local Jitter* represents the short-term variations in fundamental frequency from one cycle to the next.
- *ppq5 Jitter* is the jitter value calculated over five cycles. This way it provides a frequency variation measure that is relatively smoother.
- *Absolute (abs) Jitter* represents the absolute differences of the voice signal in consecutive cycle lengths.
- *Local Shimmer* is a value that represents the variation in the sound wave cycles' amplitude.
- *apq5 Shimmer* is the shimmer value calculated over five cycles. This provides an assessment of amplitude variations that is relatively more robust.

Overall, jitter calculates the voice frequency's stability and is often chosen as a measure for the detection of voice disorders or vocal pathologies. In short, jitter shows the variations in pitch. In the case of shimmer, it is actually an evaluation measure for the stability of amplitude in the voice. Like jitter, shimmer is critical in diagnosing and researching voice quality.

The Harmonic-to-Noise Ratio (HNR), as the name implies, is a measure of the harmonic sound to noise ratio in a voice signal. It measures the ratio between the fundamental frequency f0 and the noise components [24]. The higher the HNR values, the less noisy the sound is. The particular feature has decibel (dB) as its measurement value.

Mel-frequency Cepstral Coefficients (MFCCs) are coefficients that altogether formulate a MEL-frequency cepstrum, which represents a sound's short term power spectrum. These coefficients are calculated on the audio segment level and they are derived by computing a spectrum of the log-magnitude MEL-spectrum [24]. The particular speech feature is popular across a huge variety of audio processing and speech recognition projects and it's always present in the feature extraction process.

What we refer to as energy is the sum of squares of the signal values, normalized by the length of the signal. And this is especially the case in the context of audio signals. Energy is particularly useful in speech/non-speech detection and speaker diarization as it roughly corresponds to how loud the signal is.

Finally, the last feature, selected for our unimodal audio model, was formants. They are the resonant frequencies of the vocal tract and they are critical in determining the phonetic quality of a vowel. Formants are a characteristic component of the quality of a speech sound and analyzing them can prove vital in distinguishing between different vowels and speech sounds.

4.2.2 Data pre-processing

In the case of the audio features the data pre-processing part was more straightforward. The first pre-processing step involved applying noise reduction on our audio files to guarantee a certain level of noise avoidance. While listening to the audios we also noticed that the participant's voice was always louder than that of the virtual interviewer. The goal was to reduce background noise while preserving the natural quality of both the interviewer's and the participant's voices; but with bigger emphasis given on the participant, who would be the center of our research. The experiments involved testing a couple of different noise reduction methods, as well as downloading the *Audacity* software in order to verify the noise quality and experiment further with noise profiles. The noise reduction methods that we tried involved *spectral subtraction* and *conservative noise reduction* and the one that performed considerably better was the first one. As such, our audio files were noise reduced using the spectral subtraction method. This method works well for constant noise like hums or hisses, while still preserving the quality of speech. The second part of the experiments was based on the strengths of the *Audacity* software. The particular software is an open-source digital audio editor that allowed us to not only verify that our noise reduction step was a success; by listening to the previously noise reduced audio files and comparing with the original files, but also to create noise profiles. Normally, we wouldn't need to go for the noise profiles, but we noticed that in a couple of audio files there was a distinct noise (probably caused by their microphone), which wouldn't be eliminated even with the spectral subtraction method. As such, we attempted to create noise profiles, specifically tailored to identify this noise and then erase it from any point it appeared at. Since even these specified noise profiles failed to eliminate the particular noise; possibly because it was part of the audio files provided, we proceeded with excluding the particular audio file (audio file 300) from our research. The overall outcome of the noise reduction preprocessing was the generation of cleaner audio files, which excluded most surrounding sounds and focused mainly on the participant's voice, while only including a quieter, softer version of the interviewer's voice.

After verifying the consistency of the sampling rate across all audio files, which was indeed the case (everything sampled at 16 kHz), the next and final step of the audio data preprocessing was to perform segmentation. The audio files were too long to be managed and analyzed properly and this is why we performed segmentation; to transform them into smaller chunks of audio. When trained on more focused and analyzable chunks, models are able to learn more effectively than when trained on longer audio files. That is so, because they are able to focus on specific, relevant portions of the audio, which consequently leads to better feature extraction and improved model performance [38]. Once again we experimented with two different approaches. The first one was *silence-based segmentation* and the second segmentation approach, which proceeded with was *volume-based segmentation with buffering*. The volume-based method, with its direct focus on volume levels and additional buffering for context, appears to be more aligned with my audio data. So, although simpler, this approach actually appears to be a better fit in our case. In this particular approach there are three main parameters that can be fine-tuned.

- *Minimum Silence Length*: the algorithm will only consider any silence longer than the value assigned to this parameter (value in milliseconds) as a potential split point. The segment length is affected on the silences' frequency. The more often the participant pauses, the shorter the segments.

- *Silence Threshold*: this threshold is set to capture quieter parts of the speech, for instance the interviewer's voice which is considerably lower than that of the participant's. It represents the upper bound for what is considered silence (in dB).

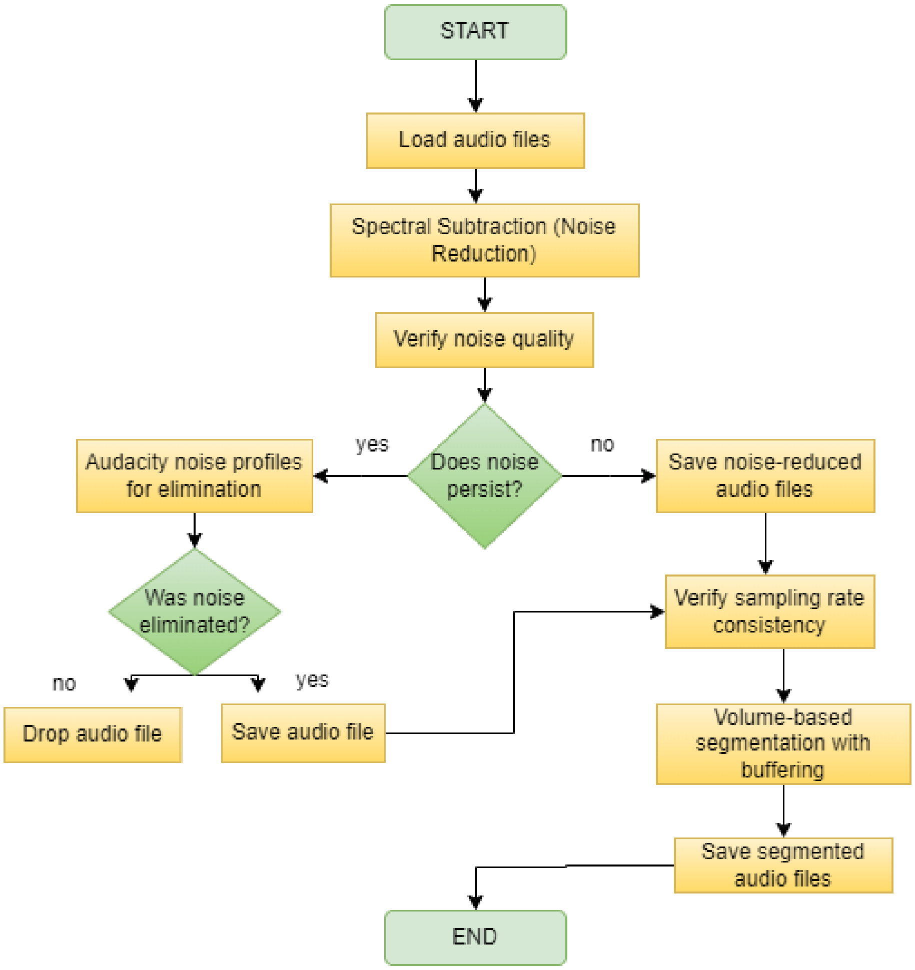
- *Buffer Length*: adding buffer at the start and end of each segments to ensure that the speech isn't cut off abruptly (i.e. having more complete sentences or thoughts).

After a lot of fine-tuning and experiments, we found the most optimal values for the three aforementioned parameters. The most accurate segmentation, in terms of preciseness and inclusiveness, was achieved with:

Parameter	Value	Description
min_silence_len	1500 ms	Minimum silence length of 1.5 seconds
silence_thresh	-65 dB	Silence threshold of -65 decibel
buffer	1000 ms	Buffer length of 1 second (start and end)

Table 6: Fine-Tuned segmentation parameters

The length of the segments that derived following this segmentation approach ranged between 1 and 22 seconds long, with a higher frequency of shorter segments (3 to 5 seconds). This means that the participants often paused for more than 1.5 seconds (1500 milliseconds). Also, based on the length of the audio and the participants' pauses, we ended up with a number of segments ranging between 50 and 150, varying per interview.


START
Load audio files
Spectral Subtraction (Noise Reduction)
Verify noise quality
Does noise persist?
yes
no
Audacity noise profiles for elimination
Save noise-reduced audio files
Was noise eliminated?
no
yes
Drop audio file
Save audio file
Verify sampling rate consistency
Volume-based segmentation with buffering
Save segmented audio files
END

4.2.3 Feature Extraction

For the extraction of the audio features we mainly used the *librosa* and *praat-parselmouth* libraries, which offer a variety of methods to work with and each of them includes most of the features that interest us for this project. The particular libraries also constitute great replacements for the source-available software for automatic extraction of features from audio signals (in short *OpenSmile*) and the collaborative voice analysis repository for speech technologies (in short *COVAREP*), both of which are really popular when it comes to the extraction of audio/speech features (including almost every feature and a big range of functionalities).

In the following subsections, the features extracted with librosa and praat-parselmouth are actually extracted on the segment level; meaning that the feature extraction functions were applied on the audio segments created during the data pre-processing stage. On the other hand, the formant features were extracted on the interview level, which means that they were extracted one step earlier; i.e. after the noise reduction process and before segmentation. The common point, however, among all those features is that they are focused and extracted on the participants' speech. Since we were mostly interested in identifying the existence of mental disorder markers and not really interested in understanding the semantic meaning between the interviewer's questions and the participants' responses, there wasn't really a reason to include the interviewer's speech.

4.2.3.1 Features extracted using Librosa

As we discussed earlier in the above subsection, some of our audio features were extracted through the librosa module and some others through the praat-parselmouth module. In the case of librosa, the extracted features included pitch, energy and MFCCs.

Pitch

In order to keep the most reliable pitches, we implemented a script so that it keeps only the magnitudes greater than the median of all magnitudes. Moreover, we filtered out zero pitches and kept only the non-zero ones. Then we modified the pitch extraction function to return the mean, median and standard deviation (std) values of the pitches.

Energy

In the case of energy the function was simpler. All we had to do was to calculate the root-mean-square (RMS) value for each frame (which is of course done automatically using *librosa.feature.rms* and then just like with pitch, we made the function return mean, median and std.

MFCCs

To extract the MFCCs we used the *librosa.feature.mfcc* function. We selected 13 coefficients, which is enough to capture important information in the speech spectrum. Finally, similarly with the previous two features, we calculated the mean, median and std of the MFCCs over each segment. This offers a compact representation of the segment's spectral characteristics. However, since there were 39 values (13 coefficients and 3 statistics for each) for each segment of every interview, the value assigned as a result of this function was an array of values. Since the values of all the other features in our dataframe were 1-dimensional numbers, we decided to flatten our MFCC arrays into single values as well. As such, we created 39 unique columns, each corresponding to a MFCC coefficient and its respective statistics (e.g. naming sense:

MFCC_1_mean, *MFCC_1_median*, *MFCC_1_std*, *MFCC_2_mean* and so on). At the same time, the original MFCC arrays (pre-flattening) were saved at a separate dataframe, so that we could later on use them for visualizations and overall feature analysis.

4.2.3.2 Features extracted using Praat-Parselmouth

Prosodic analysis was performed using the praat-parselmouth library. More specifically, we extracted jitter, shimmer (features which are related with harmonicity) and HNR using Praat scripts.

Jitter

Extracting jitter for our audio data proved to be a bit tricky, since we needed to identify first the correct parameters for each jitter type. Following the advice from [39], we created a jitter function by using and combining the praat-parselmouth call() and point process functions. Then using the above functions we extracted three types of jitter, namely *jitter local, jitter ppq5 and jitter absolute*. Each of these types of jitter offer a single value per audio segment of each interview and as such it's not possible to extract statistical measures of these features on the segment level. For each type of jitter we had to set the time range (s), period floor (s), period ceiling (s) and maximum period factor [39]. Further information available at section *Voice_2__Jitter* of manual [40].

Shimmer

The extraction process for shimmer followed the exact same path, with the only difference that the extracted features in this case were *shimmer local* and *shimmer apq5* [39]. Once again the function returned a single value (per type) for each audio segment, preventing us from extracting additional statistical measures. Further details can be found under section *Voice_3__Shimmer* of the manual [40].

When someone analyzes speech signals, they can utilize a point process to model the moments when vocal folds vibrate. In cases like jitter and shimmer extraction, the point process can help identify these exact moments of vocal fold vibration. The analysis of the timing and amplitude variations between these points (moments) allows for the computation of frequency and amplitude stability, which are represented by jitter (variation of frequency) and shimmer (variation of amplitude) respectively. In Praat and Praat-Parselmouth, the *To PointProcess (periodic, cc)* function is used to convert a speech waveform into such a point process, specifically focusing on points of vocal fold closure, which are essential for jitter and shimmer analysis [40].

Harmonics-to-Noise Ratio

Just like with jitter and shimmer, HNR was extracted by creating a function that combines the call and point process functions of the Praat software and only one value is returned in this case, too [39]. This process was guided by the Praat manual [40] and more specifically section *Voice_4__Additive_noise* of the manual.

4.2.3.3 Formants

The formant features were actually already extracted by the authors of the particular dataset and as it was mentioned earlier they were extracted on the interview level. For this part of the modeling we decided not to devote additional time to extract formants all over again on the segment level. Instead, we moved on with two simultaneous processes; one handling the 'segmented' features and one the formant features. Extracting formant features from the entire audio gives us an overall picture of the formant characteristics throughout the interview, which still offers a comprehensive analysis.

The formant files provided with the dataset were in the format of XXX-FORMANT.csv, where 'XXX' corresponds to the interview number (301 to 492, respective of each file). These csv files included the first 5 formants (F1 to F5) of each interview. Then, just like we did with the other audio features, we aggregated the formants and calculated the mean, median and std values; resulting with 15 columns (5 formants and 3 statistics for each).

5 Methodology

5.1 Pipeline

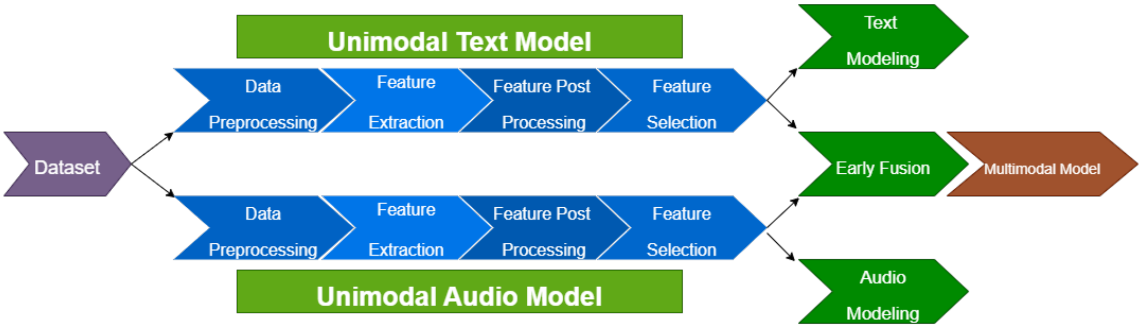


Figure 1: Workflow Process

5.2 Unimodal model (Text)

5.2.1 Post-extraction processing of textual features

To begin with, as it was also explained in the feature extraction subsection, we narrowed down and extracted only particular LIWC categories, which we deemed important for the particular research.

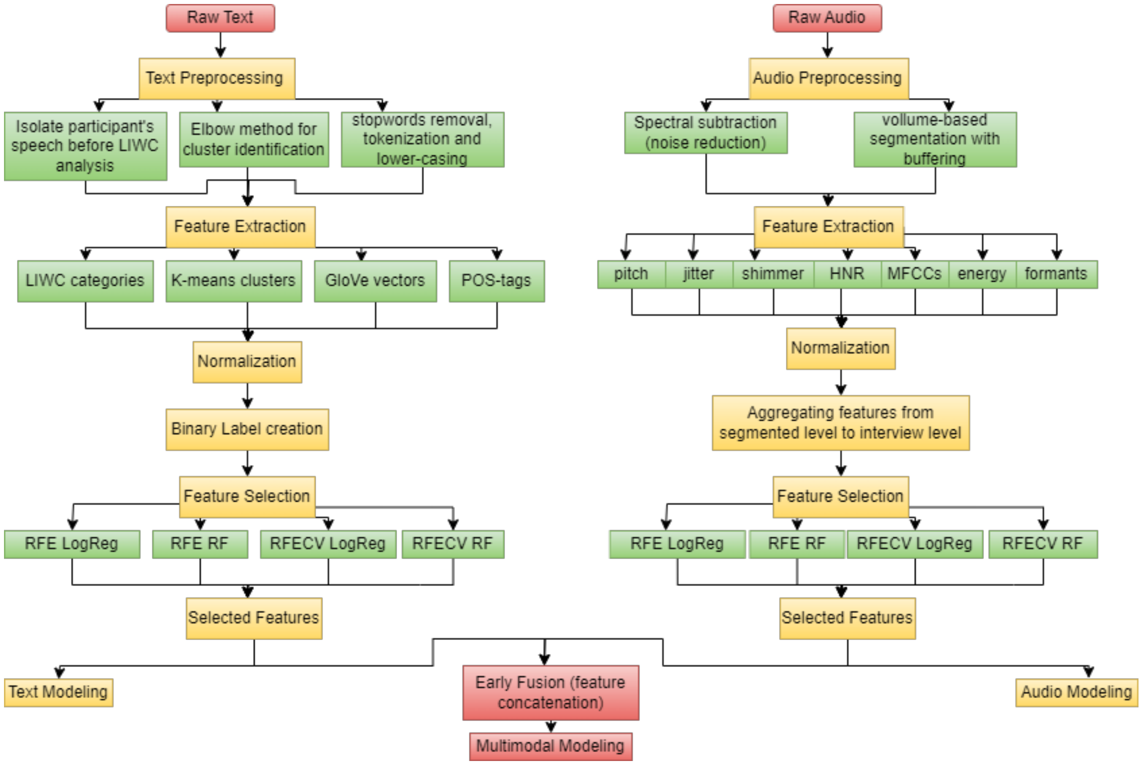


Figure 2: Dataflow Diagram

Z-Score Normalization - LIWC

The very next step was to normalize the LIWC categorical scores, in order to ensure a consistent scale and consequently improve the comparability of the data. This entails making them more the data more homogeneous and possibly revealing new patterns in the data that weren't evident before. For the normalization task, Z-score normalization, also known as standard scaler or standard normal distribution, was selected. This method is a specific type of probability distribution that standardizes the data to have a mean of 0 and a standard deviation of 1 for each feature.

Standard Scaler Formula:

$$x_{standard} = \frac{x - mean(x)}{std(x)}$$

k-means clustering and silhouette score

Then, the next step was to create some clusters on the normalized results and assign them to our data. Since the elbow method [36] indicated that 4 or 5 clusters were the most optimal approach, we decided to utilize the *silhouette scores* method [37], so that we could conclude on the clustering number. This method performs an iteration over 4 and 5 clusters (in our case) and for each number of clusters it performs k-means clustering. Then, based on the object's similarity to its own cluster and compared to other clusters, the silhouette score is calculated. Finally, the calculation and comparison of those scores identified 4 clusters as the best clustering number for our case.

As such we proceeded our experiments with 4 clusters as the parameter. For relevance and effectiveness purposes, k-means clustering was applied to the dataframe that included only the selected LIWC categories (i.e. the specific normalized categorical scores). While on that, we also decided to perform a Principal Component Analysis (in short PCA) for both visualization and dimensionality reduction purposes. Specifically, we reduced the dimensionality of our data to two principal components, aiming to create a more distinct visualization and inspection of our clusters. This PCA transformation aimed to capture the most important variance within our LIWC features in two dimensions. This makes visualization easier and enhances the potential of revealing underlying patterns. The plots that were created clearly show how the text segments were grouped based on their LIWC features.

Below, we present a visualization of the clustering results across some of the selected LIWC categories, applied on their normalized scores.

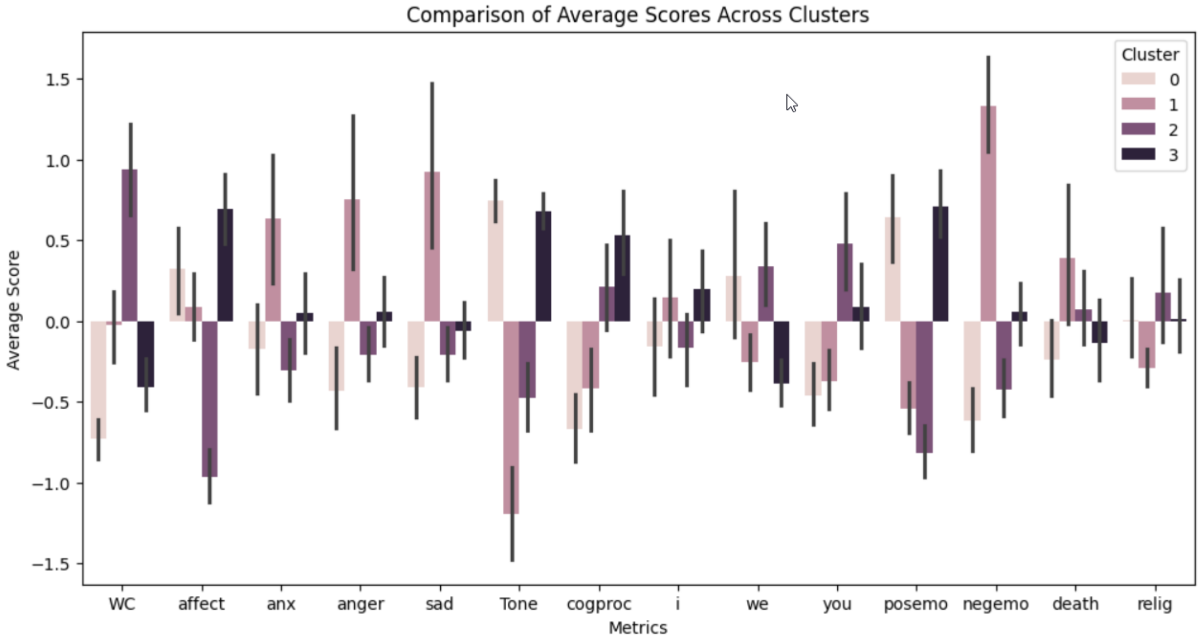


Figure 3: Visualizing clustering results across various LIWC categories.

As part of our analytical approach, we also performed a qualitative and a statistical analysis on the LIWC categorical results. The qualitative analysis was more about identifying the extent to which clustering was useful. To find out, we merged our previously cleaned transcripts with the clustered data and we extracted 3 random samples of text from each cluster to see the context's similarity (mostly on the semantic level). On the other hand, statistical analysis was performed for more than one reasons. The main reason behind the importance of the particular analysis has to do with the creation of our own binary labels, something that will be properly discussed later on in this section. Another reason was because this allowed us to create a correlation matrix and observe how the extracted categories interact with each other.

Z-score Normalization - GloVe Embeddings

The same standardization method was used in the case of the GloVe embeddings as well. The only thing that changed in this case is that while recreating the dataframe that would hold these features, we named the columns in a conventional way. Since we had utilized the *100d* model, column 1 was named '*glove_dim_1*' and so on till '*glove_dim_99*'. At this point, for convenience purposes again, we also created an incomplete features set, by merging the normalized glove dimensions with the normalized LIWC categorical scores and their clusters, serving as a checkpoint.

Z-score Normalization - POS-Tag Counts

While researching on textual features it was indicated that part of speech tags may offer some enriched information when it comes to understanding the grammatical structure of sentences and the role of each word withing those sentences. The script we advised in this case iterates over all our cleaned text files, reads its content, applies the tokenization and pos-tagging function and creates a new column in the dataframe that contains a list of (token, tag) tuples for each row. Then using *json.dumps()* we convert the lists of tuples into a JSON string. This was an important step of the process since JSON can encode complex data structures (the lists of tuples in our case) into a string format that makes it easier to store in a text file. This guaranteed that the structured information was preserved in a widely recognized format, allowing for easier processing during next steps. Overall, the choice of using JSON was motivated by the need to serialize these lists of tuples into a format that could be machine-parseable, while maintaining compatibility with various data processing tools. So, up to this point we had created a dataframe that contained two columns; i.e. *filename* (for identification) and *tokenized_and_tagged*.

The above dataframe is then merged with the previously extracted *POS-tag counts* (see in 'Feature Extraction' section), including the various pronoun counts that we devised. This complete dataframe is then normalized, using the standard scaler, to match our LIWC and GloVe embedding features.

Complete Text Features set

The previously created 'incomplete features set', containing the normalized and clustered LIWC categorical scores and the normalized glove vectors, was merged with the dataframe holding the normalized POS-tag counts. This constituted the creation of our final features set. The reason why we decided to combine all these unique types of features into a single bigger set was the variety of benefits each had to offer in this research. The structured, psychologically informed LIWC features can combine smoothly with the denser, contextual representations of text offered by the GloVe embeddings. This combination allows capturing both specific linguistic markers and nuanced semantic and syntactic relationships between words. Along with the enriched information obtained by the POS-tag counts, the final features set is more than comprehensive enough for the modeling tasks that follow.

The following list sums up the complete textual features set:

- Normalized LIWC categorical scores (narrowed down)

- Normalized PCA-transformed LIWC features (2 components)
- Cluster labels (4)
- Normalized GloVe embeddings (100 dimensions)
- Normalized POS-Tag counts (including various pronoun counts)

Algorithm 1 Post-extraction processing of text features

```
liwc_df = z_score_normalization(liwc_df)
optimal_clusters = silhouette_scores(liwc_df, [4, 5])
kmeans_df = kmeans_clustering(liwc_df, optimal_clusters)
pca_df = pca_transformation(kmeans_df)
glove_df = z_score_normalization(glove_df)
Merge glove_df with normalized LIWC dimensions

for each text file in cleaned_text_files do
  Read content and apply tokenization and POS tagging
  Create a column with (token, tag) tuples for each row
  Convert the lists of tuples into a JSON string using json.dumps()
end for
pos_tag_df = create_dataframe(filename, tokenized_and_tagged)
pos_tag_df = z_score_normalization(pos_tag_df)
Merge pos_tag_df with previously extracted POS-tag counts
Merge glove_df, liwc_df, pos_tag_df into final_df
```

Binary Label creation

As part of our research for identifying potential mental disorder markers, we also had to create a binary label that would serve as an indicator of our models' performance. This was actually a delicate process, as the label accuracy would have a direct impact in the overall results. Since, we didn't have direct access or contact with clinical specialists to guide us, we took the longer route, which involved more work and lots of experiments. For this process we used our main dataset, E-DAIC, and the procedure was focused on most of the mental disorders studied in this paper (i.e. Depression, Bipolar Disorder, Schizophrenia, PTSD and even ASD).

More specifically, the first step was to revisit the LIWC category mapping with various mental health markers that had been identified through literature (Table 1). Most of these markers are available in the *Feature Extraction* section, but for reference purposes here are the categories that were deemed most relevant: *anx, bio, cogproc, death, i, negemo, posemo, relig, sad, social* and *they*. Using these LIWC categories we decided to create some threshold values in order to classify between the presence or absence of a marker across our data. In order to discern the optimal thresholds, we moved on with extracting the statistical summary of the aforementioned normalized categorical scores. Our purpose was to observe and compare the *mean, min, max* and the *top 75 percentile* statistics. Along with the statistical summary we also extracted information on *Skewness* and *Kurtosis* for each category. Finally, to complement the labeling process we created distribution plots for each chosen category, with *Frequency* on the y-axis and *Score* on the x-axis. These distribution plots proved to be essential for setting the thresholds. By taking into account the actual statistical distribution of each category, we set the thresholds to capture truly significant deviations (both positive and negative) that can potentially indicate underlying patterns related to mental health markers. Knowing the skewness (asymmetry of the distribution) and kurtosis (mean outliers) also played a big part in setting thresholds. It helped ensure that the marker identification criteria are not only based on central tendency but also on the distributions' tails. Particularly for categories with high values on these two measures, setting thresholds to capture extreme values ensures that the labeling process is more likely to identify instances that stand out significantly from the norm. And discovering these instances can be crucial for the detection of mental health disorder markers.

Utilizing all the information obtained from the aforementioned steps, we created 10 initial threshold sets, varying from very sensitive ones to very strict ones. In the following table we present the results of the these initial sets.

Set	Characteristic	Marker Presence
1	Starting setup	45%
2	Sensitive	75%
3	Balanced Sensitivity	86%
4	Moderately Strict	39%
5	Increased Sensitivity	96%
6	Increased Specificity	22%
7	Refined based on statistics	43%
8	**Refined based on statistics and distribution plots**	**29.6%**
9	Re-refined based on statistics	34%
10	Moderately Inclusive	49%

Before proceeding with additional refinement of our final threshold set, we brought up the binary labels that were previously set (provided along with the data) on our dataset by the Patient Health Questionnaire –8 (PHQ-8). We noticed that this questionnaire had flagged 30.22% of the participants as depressed (given 1 as a presence label). Although those binary labels only represent the depression disorder and are not taking into account all the other mental health issues that we are interested in, we still decided to go for a threshold set that achieves a similar marker presence. This is so, that there is a considerate amount of present markers, while still being moderate. More simply, we believed that a marker presence of approximately 30% achieves a good trade off between sensitivity (over-inclusiveness would identify too many instances) and specificity (strictness may lead to missing potential markers).

As such, we selected our most accurate threshold set (based on the statistical summary and distribution plots) that returns a 29.6% marker presence and created 5 more modified sets around this.

Set	Characteristic	Marker Presence
11	Balancing Towards Sensitivity	32%
12	Increasing Specificity	22%
13	Increased, yet balanced, Sensitivity	41%
14	Composite set based on the top 3 sets till this point	33%
15	**Refined composite set**	**31.7%**

In the table above, the top three sets that made up the composite set were sets 8, 9 and 11. These three sets seemed to achieve the best balance between sensitivity and specificity and this was also shown in the result of the composite set. Moreover, with the slight modifications made on set 14 (specifically, a slight increase on the threshold of category 'i'), we got a set that achieved the most optimal balance as it was neither over-inclusive nor too strict, respecting the complexity of identifying linguistic markers in mental health research. With a marker presence of 31.7%, we selected set 15 for our labeling. The refined composite set's alignment with the known prevalence suggests that it effectively captures a realistic proportion of instances with potential markers, indicating its thresholds are well-calibrated. The following table shows the exact values of the final threshold set:

Category	Threshold
negemo	1.9
sad	2.25
anx	2.25
i	1.5
they	1.75
social	lower: -2, upper: 2
cogproc	lower: -2, upper: 2
bio	1.75
relig	2.75
death	2.25

The thresholds for categories 'negemo', 'bio' and 'death' are a middle ground, or compromise if you want, between the strictness of set 8 and the insights of sets 9 and 11. Category 'i' has the lowest threshold because of its lower distribution. Finally, categories 'social' and 'cogproc' have a dual threshold. Lower values on 'social' aimed to capture potential social withdrawal and upper values aimed to capture potential excessive social referencing. Similarly, lower thresholds in 'cogproc' aimed to capture disorganized thinking (for instance, for schizophrenia) and upper thresholds aimed to capture higher cognitive processing (for instance, overthinking in the case of OCD).

This binary label creation process can be effectively repeated on any dataset that involves textual data. This methodology is independent of the used dataset (E-DAIC), but dependent on the LIWC features. In short, to replicate this process, one needs to identify which LIWC categories match with the aim of their research (in our case, the categories that are linked with mental disorder markers). Then, by extracting the statistical summary of these categorical scores (post normalization, to ensure a common scale) and plotting their distribution graphs, one can start experimenting with thresholds. These experiments involve increasing/decreasing the threshold for each category, based on their statistics and distribution, until a satisfying setup is realized. In a sensitive topic, like the one studied in this paper, the final goal is to find the balance between specificity (strictness, less inclusive) and sensitivity (lower thresholds, over-inclusiveness). To evaluate this process there are two options; either compare with other similar binary labels assignment or mark the exceeded threshold for each sample of the dataset and use the observations for additional modifications.

Feature Selection

Feature selection is an important process, especially in cases like ours where the complete features set is comprised of more than 150 features. A good feature selection can offer reduced computational complexity, as well as increased model accuracy and interpretability. For these reasons we decided to approach feature selection using wrapper methods. Wrapper methods prepare different combinations of sets of features, they evaluate them and eventually they compare each combination with the rest of them. These methods utilize a predictive model in order to perform the evaluation of each combination and then assign a score based on the chosen model's accuracy.

In our case, we proceeded with the *Recursive Feature Elimination* (RFE) approach. As the name implies RFE works by recursively removing the least important features based on the weights of the model and it re-builds the model until the specified number of features is reached. This is an effective feature selection method that can be used along with various machine learning algorithms [10]. In order to identify the optimal number of features and the most suitable predictive model for the selection of these features we performed exhaustive experiments. In the following table (Table 7) we present the parameters of these experiments and later on, the most optimal choice is discussed.

Wrapper Method Experimentation			
Method	Model	Re-scaling	Number of Features
RFE	Logistic Regression	yes	10
RFE	Logistic Regression	yes	15
RFE	Logistic Regression	yes	20
RFE	Logistic Regression	no	10
RFE	Logistic Regression	no	15
RFE	Logistic Regression	no	20
RFE	Random Forest	yes	10
RFE	Random Forest	yes	15
RFE	Random Forest	yes	20
RFE	Random Forest	no	10
RFE	Random Forest	no	15
RFE	Random Forest	no	20
RFECV	Logistic Regression	yes	25
RFECV	Logistic Regression	no	19
RFECV	Random Forest	yes	89
RFECV	Random Forest	no	89

Table 7: Refining Text Feature Selection

Concerning our experimentation on feature selection (see Table 7) we devoted some additional time because a good features set would eventually bring better results and could potentially uncover some underlying patterns and relationships between features. This is why we iterated over this process while tuning each of the parameters, with respect to the recommended features subset and the modeling results achieved over each iteration. The first parameter was using *RFE* or *RFECV*. The first one performs the feature selection given a fixed number of features to select (set by us), while the latter performs the selection using cross validation and finally recommends the best number of features (along with the corresponding top features names) as judged by the assigned scores. The selection process of these methods is performed on the training data, in order to avoid any information from the test set being leaked. This ensures an unbiased evaluation of the model's performance. In both approaches we experimented both with *Logistic Regression* and with *Random Forest*. RFE (as well as RFECV) is a model-specific feature selection method. This means that the final set of features, proposed by the process, is influenced by the characteristics and requirements of the respective model and emphasizes on maximizing its performance. Then, for each predictive model we iterated three times, setting the number of features to 10, 15 and finally 20 (for a broader view). At this point, we noticed that we had been performing the experiments including an additional scaling (standardizing the normalized scores) in our method. Although research showed that this wouldn't necessarily be something negative, we decided to run all the previous iterations (over Logistic Regression and Random Forest and for all 3 sets of numbers of features) while tuning a new parameter, that being the addition or exclusion of re-scaling. Similarly, in the case of RFECV, we performed two more iterations (one for each model) by removing the additional scaling of our features.

As shown in Table 7, *RFECV with Random Forest* (RFECV-RF) suggests the same number of features (and the same features as well) regardless of rescaling or not, which means that Random Forest as a model is not affected by it. This was also proven while observing the selected features from the RFE-RF, which proposed the same top 10, top 15 and top 20 features whether we added the scaler or not.

Python Libraries

Each unimodal model required a particular set of python modules to run properly. Specifically, for the text model we had to install *pandas, numpy, nltk, spacy, gensim, tf-keras tensorflow, sklearn* and *matplotlib.pyplot*.

Overfitting

First of all, let's start by introducing the concept of cross-validation (CV) as a good method to mitigate overfitting, especially in cases where the dataset is limited (like ours). Generally, overfitting occurs when the model performs well on the training data but poorly on the test data (which are the unseen data). This can happen when the model learns the training data too well, along with its noise and outliers, which consequently harms the model's performance on new, unseen data. CV allows us to rotate the test set through the entire dataset, providing us this way with a more comprehensive view of the model's performance on the unseen data.

In order to test our models for overfitting, while using cross-validation, we created a script that shows the comparison between train and test set performances across all folds. Specifically, we compared two metrics, the accuracy and the AUC-ROC score. Consistently high performance across all folds indicates that the model generalizes well. On the other hand if there is an intense variance in the performance across the different folds then we have the opposite conclusion. This overfitting test was performed on the top 10, 15 and 20 features of the most prominent feature selector. We will present and elaborate on our findings in the *'Experimental Results'* section of this report.

5.3 Unimodal model (Speech)

5.3.1 Post-extraction processing of speech features

Having completed the extraction of all the selected audio features, our next step was to create a loop that iterates over every single segment of each interview and performs all the previously mentioned feature extraction functions. This consequently provides a feature dataframe populated by values on the segment level (188 x number_of_segments rows ≈ 20560 rows). Then a similar loop was created for the formant features, creating this way a second dataframe populated by singular values on the interview level (188 rows).

Z-score Normalization - segmented features

Following similar processes, as with the textual features, we proceeded with normalizing the audio features that we had extracted on the segment level. In order to achieve a solid level of consistency throughout the whole model, normalization was once again performed using the standard scaler formula. Then we saved the normalized features and their values into a new dataframe for easier access.

Z-score Normalization - Formants

Since we were planning on including the formant features in our final audio features set as well, the next logical step was to normalize the formant features. Z-score normalization was perfomed once again, leading to a uniform scale across our various features and enabling their smoother feeding in machine learning and deep learning models.

Binary Labels

Given that both unimodal models were created around the same dataset, it was only natural that we assigned the same binary labels. As such, we created a *'numerical_id'* column, which included only the integer representing the respective interview, in both the text features dataframe and the segmented features dataframe. Then, we loaded both dataframes and we merged the *'Marker'* column of the text features dataframe with the segmented features dataframe, matching them based on the *numerical_id*. As a result all the segments of an interview were assigned the exact same *'Marker'*. Of course, having each different segment of an interview hold the value of the *'Marker'* is kind of redundant, but it did offer some insights while creating visualizations.

Our formant features dataframe already had a column holding the integer that represented the interview id. So, we just merged the *'Marker'* column of the text features dataframe with the formants dataframe matching the pre-existing column with the *'numerical_id'* column of the latter dataframe.

Aggregating to the interview level

While revisiting our dataframes and the values of our extracted features we noticed that some segments held blank values in the *jitter_local, jitter_ppq5, jitter_abs, shimmer_local*, and *shimmer_apq5* columns. This indicated moments in the audio where these measurements were not applicable or could not be calculated; something that could be attributed to reasons like silence or non-vocal sounds within those segments. The only way to go from this point would be to either exclude these features or exclude the relevant segments, since not all algorithms can handle missing values intrinsically. Considering this issue, along with our interest in merging the segmented features with the formants into a complete audio features set, lead us to the aggregation concept.

To aggregate our segmented features to the interview level properly, we dropped the *'filename'*, *'Marker'* and *'numerical_id'* columns, we grouped everything by *'interview_id'* and aggregated for three statistics by using the *.agg(['mean', 'median', 'std'])* method. For each interview id, this method calculated the mean, median and std values of each feature across all segments attributed to that interview.

Example: To clarify the process, let's take pitch as an example. So, for pitch we had already extracted mean, median and std on the segment level. Following the above aggregation method we got the mean of the *pitch_mean* values across all segments (which is actually the average of the averages), the median of those *pitch_mean* values (central tendency of pitch variation across segments) and also the standard deviation (pitch variability) among those values within an interview. Similarly, we got the mean, median and std of the *pitch_median* and *pitch_std* values of all segments within each interview.

This aggregation provided us with a second-level summary of each feature, encapsulating the overall distribution of that feature (like pitch_mean) across an interview.

Complete Audio Features set

Finally, all audio features were on the same level and scale. The final step of our feature engineering for the unimodal audio model included merging all our feature types into a single complete set. Before merging, we dropped all the duplicate columns and converted the identifiers of both dataframes into integers. Then we merged the previously segmented-level features with the interview-level features and ended up with our final audio features set, including:

- Normalized and aggregated librosa features
- Normalized and aggegated praat-parselmouth features
- Normalized formant features

Feature Selection

Our complete audio features set is comprised of 168 features (including the aggregated statistics on each). This amount of features, although comprehensive, is still hard to handle and would most probably lead to overfitting. So, before moving on to modeling feature selection was deemed as a necessary process. Just like with the unimodal text model, we went with the

Algorithm 2 Post-extraction processing of speech features

```
for each interview in interviews do
    segments = split_interview_into_segments(interview)
    for each segment in segments do
        features = extract_features(segment)
        append features to segmented_features_df
    end for
    formant_features = extract_formant_features(interview)
    append formant_features to formant_features_df
end for
segmented_features_df = z_score_normalization(segmented_features_df)
formant_features_df = z_score_normalization(formant_features_df)
save_dataframe(segmented_features_df, “segmented_features_normalized.csv”)
save_dataframe(formant_features_df, “formant_features_normalized.csv”)
Create a numerical_id column with interview IDs.
for each dataframe in [text_features_df, segmented_features_df] do
    Load dataframe
    Merge Marker column from text_features_df with dataframe based on numerical_id.
end for
Merge Marker column with formant_features_df based on numerical_id.
for each dataframe in [segmented_features_df, formant_features_df] do
    Drop filename, Marker, and numerical_id columns.
    Group by interview_id and aggregate using .agg(['mean', 'median', 'std']).
    Append aggregated features to final dataframe.
end for
```

wrapper methods and more specifically with the *Recursive Feature Elimination (RFE)* technique. We performed the exact same experiments with the text model in order to be consistent and objective. Hence, in the following table we are presenting only the RFECV experiments, which performed completely differently and are consequently worthy of mention.

Wrapper Method Experimentation			
Method	Model	Re-scaling	Number of Features
RFECV	Logistic Regression	yes	1
RFECV	Logistic Regression	no	1
RFECV	Random Forest	yes	37
RFECV	Random Forest	no	37

Table 8: Refining Audio Feature Selection

By observing Table 8, one can notice that both of our chosen classifiers (i.e. Random Forest and Logistic Regression) weren't affected at all by the presence or absence of additional rescaling; meaning that the number of proposed features didnt' change. Moreover, even though the number of audio features was larger than that of the text features, the RF classifier with the

RFECV approach suggested a much smaller number of features as optimal. Also, interestingly enough, the Logistic Regression classifier with the RFECV approach kept on recommending only a single feature. For reference, that feature was the *MFCC_median_7_median*, which represents the median of all medians for the seventh MFCC coefficient. This result can be due to various reasons. Among them, the most possible scenarios are *Logistic Regression's simplicity*; in which case if the addition of more features does not significantly improve the cross-validation score RFECV will opt for the simpler model with fewer features, and *feature redundancy*; i.e. highly correlated features might not provide additional value when used all together and as such RFE with cross-validation might consider the extra features unnecessary.

Python Libraries

In the case of the audio/speech model, the necessary modules included *praat-parselmouth, librosa, numpy, eyed3, pydub, pyAudioAnalysis, hmmlearn, noisereduce, soundfile, tf-keras tensorflow, sklearn* and *matplotlib.pyplot*.

Overfitting

Due to the performance of the audio model being significantly lower than that of the text model (see Tables 15 and 16), which will be analyzed further during the *'Experimental Results'* section, the process of testing for overfitting wasn't as exhaustive here. We still applied the same script as with the text models, just so that we could have an idea of how intense the presence of overfitting actually was.

5.4 Multimodal model (Text + Speech)

We aimed for the multimodal model to be a pure combination of the text and speech modalities, which we had experimented with up to this moment. Hence, our methodology for the multimodal model entailed an early fusion of the selected text and acoustic features. The resulting merged features set was used as input in our machine learning models, with the overall purpose of showing the strength of such a combination.

Fusion Approach

Among the various options, the one that seemed as the cleanest and "most" solid for our current objective was early fusion, also known as feature level fusion. This specific fusion involves the combination of features from different modalities (text and audio in this case) into a single feature vector before feeding them into machine/deep learning models. This approach not only allowed the model to learn from the integrated information presented by these two modalities (from the very first stage of the training process), but it also provided us with a means of completing the multimodal model without having to revert back to text-speech alignment. Overall, it leads to a straightforward, computationally less intensive model that doesn't require sophisticated synchronization between modalities.

Some of the benefits of early level fusion are that it allows for early correlation of distinct multimodal features and that it simplifies the architecture of the model. Furthermore, early fusion offers consistency in data handling. Since we had processed our data the exact same way in both modalities (i.e. performing normalization immediately after extraction, using the same scaler), we didn't have to perform any further processing post-fusion. This means that as soon as the fusion was complete, the combined features set could be directly inputted in the classification models.

The concept was to enable our models to learn from the interactions between text and audio/speech characteristics directly by merging the features of the two modalities into a single dataframe and consequently using that combined set for training. We hoped that this approach could potentially lead to a richer representation of the data and ideally improve the accuracy and predictive power of our models.

Feature Selection

As we elaborated earlier on the *Methodology* section, for each modality we had extracted a big number of features, totaling in about 300 features combined. Of course creating a combined features set of this size wouldn't be either practical or efficient for modeling. Hence, for each modality, we decided to select the top 20 features of the selector (*see 'Feature Selection' experiments*) that performed best. The selection was based on the overall performance of the various feature selectors across different metrics; specifically accuracy, AUC-ROC score and f1 scores (*these will be presented in the 'Results' section*). Having identified the top feature selector from each modality, we finally had a sum of 40 multimodal features. Moreover, having noticed the performance gap between the unimodal text and the unimodal audio models, we decided to experiment by adding some weight on the text features. Since one unimodal approach did better than the other one, giving some additional emphasis on its features could potentially lead to better results overall. On the other hand, we also wanted to be able to compare with some unbiased and more balanced combinations and this is why we actually created three different dataframes (sets of features). Further details available at Table 9.

Multimodal feature combinations		
Set	**Features**	**Description**
1	20 textual, 10 acoustic	Significant weight given to the text features
2	20 textual, 15 acoustic	Slightly more weight on the text features - more balanced
3	15 textual, 15 acoustic	Totally balanced combination

Table 9: Refining Multimodal Feature Selection

Python Libraries

The multimodal model was purely the combination of our previous two unimodal approaches. As such, we mainly needed to install the libraries that would be necessary for the machine/deep learning models and for visualization. These included *sklearn, pandas, scikeras, tensorflow==2.15.0* (the specific version was necessary to support scikeras), *matplotlib.pyplot* and *seaborn*.

Additionally, for reference purposes, before rejecting the text-speech alignment approach we also used the *pratio* and *chardet* modules.

5.5 Experimental Design

Given that our research question focuses on comparing unimodal and multimodal approaches to identify mental health disorder markers, we structured our approach around building distinct models: one purely based on textual data, one solely on audio data, and one multimodal model that integrates both. For our experimental design, we opted to explore four different models, selecting three supervised machine learning models— *SVM (with a linear kernel), Random Forest* and *Logistic Regression*—and one deep learning model; i.e. a fully connected neural network (*Dense Layers*). To evaluate and validate the comparison between the modality approaches, we decided to use the same machine/deep learning models across all cases. Before getting to the modeling implementation and based on the contacted feature selection process (using the RFE wrapper methods as explained in earlier sections), we selected the top performing features of each modality, anticipating that their integration in the multimodal model would yield complementary benefits.

Moreover, regarding the train-test split process before modeling, our initial approach was to use the typical train-test split with 80% for the train set and 20% for the test set. However, during our experiments we noticed that with this approach SVM and Logistic Regression achieved the exact same scores (between them) across all metrics and numbers of features and Dense Layers and RF had the same scores (individually) regardless of the feature set size. We believe this score consistency occurred because of the relatively small dataset, which led to a correspondingly small test set (about 38 samples). That's what pushed us to try a train-test-val split, with 80% on the train set, 10% on the test and 10% for the validation set. The results achieved with this approach were overall worse and we quickly rejected it because it led to an even smaller test set of 19 samples. To solve the issue that arose, we implemented k-fold cross validation. Cross validation (CV) not only fixed the issue with the duplicate results but also proved to be the most reliable and effective approach. The concept behind CV is to divide our dataset into 'k' distinct folds (5 in our case). Then the model is trained on 'k-1' folds and 1 fold is used for the test set. This process is then repeated 'k' times, having each of the 'k' folds used exactly once as the test set. k-fold CV helps to efficiently assess the models' effectiveness and also to prevent the models from overfitting.

5.5.1 Machine Learning Models

In selecting the machine learning models, our criteria were based on two key considerations. Firstly, all three models—SVM, Random Forest, and Logistic Regression—have been frequently employed in relevant literature, including studies on multimodality and mental health disorder analysis. Secondly, our choice was influenced by model complexity. Given that this research involves an initial exploration into new territories, opting for models with more manageable complexity that can still efficiently handle high-dimensional data and features was essential. Going forward, based on the outcomes of this preliminary phase, adjustments to increase model complexity or incorporate additional features might be considered to enhance predictive performance.

5.5.1.1 Support Vector Machine (SVM)

SVM is one of the most popular supervised machine learning algorithms both within our gathered literature and overall. The core principle of SVM revolves around binary classification. In such cases, the objective is to categorize data points into two distinct classes. SVM operates by identifying and establishing a boundary in the feature space that optimally divides these classes. The simplest form of spatial division is linear separation [9]. The choice of Support Vector Machine, with a linear kernel, as one of our models was an easy one, since its fundamental concept matched the aim of our own research; i.e. binary classification between two classes (presence or absence of a mental illness marker). Moreover, as it was also discussed earlier, SVM has been showing a great performance through relevant works and this was another reason why we wanted to experiment with this classifier.

5.5.1.2 Random Forest (RF)

Similarly with SVM, Random Forest was selected as a classifier because of its popularity across previous related works and because of its built-in skill to avoid overfitting to the training data (originating from its randomness). Random Forest is a prominent ensemble learning technique that employs decision trees. RF applies a bagging approach to manage the weaker classifiers and it works similarly to boosting. However, in bagging, each new weak learner is added by searching for the optimal feature within a randomly selected subset of the data, rather than focusing on data points that challenge the existing classifiers. Random Forest essentially merges numerous decision trees, each constructed from different random data samples, into a single model—hence its name [9].

5.5.1.3 Logistic Regression (LogReg)

Although Logistic Regression was not selected as commonly as the previous two ML models, paper [15] discuses a lot of related works that utilized the particular classifier and presented a variety of results, all ranging between medium to high performance. LogReg is similar with SVM in the sense that this is also widely used as a statistical method for binary classification. The concept of LogReg is that it estimates the probabilities by using a logistic function, which is an S-shaped curve that can take any real number and map it between 0 and 1. This approach enables the classifier to handle binary cases where the dependent variable is categorical. The particular method is specifically efficient for problems with linear relationships between the dependent and independent variables.

Particularly in the case of Logistic Regression, we also experimented further with a regularization technique in order to prevent overfitting. We tried the L2 regularization method, which helps penalize large coefficients. This approach effectively decreases overfitting by encouraging the classifier to keep the weights small.

5.5.2 Deep Learning Models

For the Deep Learning models approach, our initial choice was between Gated Recurrent Unit (GRU) and Long Short Term Memory (LSTM). These two approaches were the most popular across papers that aimed to diagnose mental illnesses using deep learning methods. Moreover, both of these methods demonstrated great results in those works and their performances were always ranked among the top ones. However, both GRU and LSTM are designed to handle sequential data and as such we decided to reject both, since the LIWC categorical scores (which are the primary features of our unimodal text model) are not sequential but instead aggregate summaries. Instead, a model architecture that starts with Dense layers would be more appropriate for the type of data representation we have.

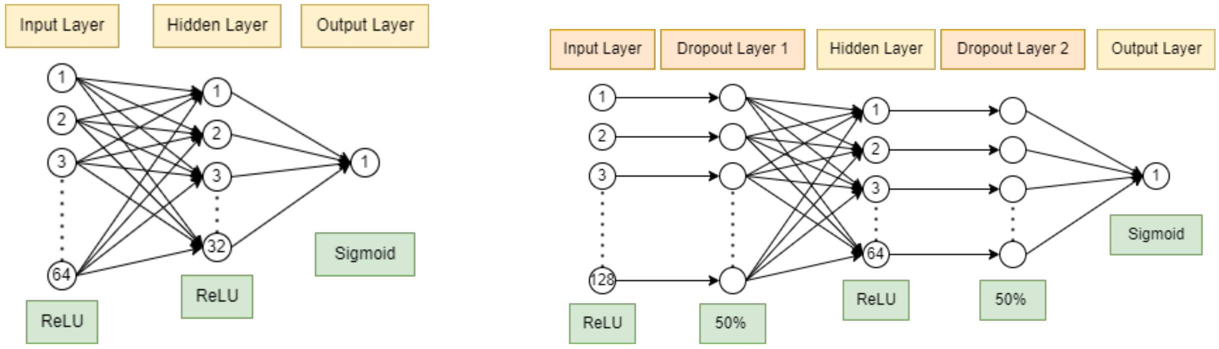
5.5.2.1 Dense Layers

Dense layers can work effectively with the non-sequential, aggregated GloVe vectors we've created, as well as with the other non-sequential features—like the LIWC categorical scores. We selected Dense Layers because of their versatility. They are fundamental to neural networks and can be used to solve a wide range of tasks, including classification. Models that include Dense layers, can learn to identify patterns in the input data through training.

In our project, we implemented two distinct neural network configurations to address the complexity difference between multimodal and unimodal data. First, for both of our unimodal approaches, we implemented a network optimized for simpler datasets. The unimodal network offers the sufficient complexity needed for the identification of effective patterns, while excluding any overfitting prevention steps; associated with more extensive models. On the other hand, the multimodal network is designed with higher capacity (larger input and hidden layer) and it also employs regularization techniques (dropout) to tackle the higher dimensionality that arises from the merging of multiple data types. The dropout step helps with the prevention of overfitting by randomly setting a proportion of the input to 0. This step is activated over each update while the model is training. In cases with diverse data types, this approach is recommended to effectively capture any nuanced patterns vital for the accuracy of the predictions. Both networks, however, include ReLU (Rectified Linear Unit) activation and Adam optimization functions to ensure robust generalization and learning across varied scenarios. Moreover, in both networks the output layer uses the sigmoid activation function to address the binary classification task. All the details on the configuration of the Dense Layers for each approach are demonstrated at Table 10.

Dense Layers configuration		
Model	**Parameter**	**Configuration**
Unimodal	Input Layer	64 neurons with ReLU activation function
	Hidden Layer	32 neurons with ReLU activation function
	Output Layer	1 neuron with sigmoid activation function
	Optimizer	Adam optimizer
	Loss Function	Binary crossentropy
Multimodal	Input Layer	128 neurons with ReLU activation function
	Dropout	50% rate after 1st and 2nd dense layers
	Hidden Layer	64 neurons with ReLU activation function
	Output Layer	1 neuron with sigmoid activation function
	Optimizer	Adam optimizer
	Loss Function	Binary crossentropy

Table 10: Configuration and compilation of the two different model approaches



(a) Unimodal FCNN config

(b) Multimodal FCNN config

Figure 4: Fully Connected Neural Network configurations per modality approach

5.6 Evaluation Metrics

The primary evaluation metric that all models are typically equipped with is, of course, *accuracy*. Accuracy is one of the most intuitive performance metrics that calculates the proportion of correctly predicted observations to the total number of observations. Although popular, accuracy proves most useful in cases where the target classes in the data are nearly balanced. Thus, in scenarios where the target/data classes are imbalanced, accuracy may not provide a reliable evaluation. Because, for instance, in a dataset where 95% of the elements belong to one of two classes, then the model can achieve a 95% accuracy just by predicting the majority class for all observations. In the particular study, approximately 30% of the data belongs to one class and approximately 70% belongs to another, indicating an evident class imbalance.

Accuracy Formula:

$$Accuracy = \frac{Number\,of\,correct\,predictions}{Total\,number\,of\,predictions}$$

Alternative metrics like the F1 score or AUC-ROC might provide a more accurate reflection of model performance in such cases, which is why we decided to utilize those as well. The *Area Under the Curve - Receiver Operating Characteristics (AUC-ROC)* score constitutes another popular evaluation metric, especially for binary classification tasks. It is calculated by measuring the area under the ROC curve, which indicates the performance of the classifier at various threshold settings. The ROC curve plots the *True Positive rate* (TPR, also known as recall or sensitivity) against the *False Positive rate* (FPR, or 1-specificity). AUC provides an aggregate performance measure across all possible classification thresholds. An AUC of 1 indicates a perfect model, while an AUC of 0.5 suggests that the model has no discriminative power; akin to random guessing.

Finally, the last evaluation metric we used was the *F1 score*, through the classification report. In short, the F1 score is the harmonic mean of precision and recall and it offers a way to combine those two into a single measure. F1 Score is particularly recommended in cases like ours where there is a clear class imbalance (large number of actual negatives).

F1 score Formula:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Along with the classification report, and for similar validation purposes, we also decided to plot the confusion matrices for each model. A *confusion matrix* is a table that describes a classifier's performance on a known test set and it also popular as a method of visualizing the quality of a classifier.

5.7 IDE and Hardware

This whole project was handled under the same circumstances from start to finish. The chosen IDE for the programming part was *Google Colab* and the reason behind this was the fact that it provides some packages for the utilization of additional RAM. Other than that it also offers a relatively clean interface and allows for the organization of the scripts into various sections. Furthermore, we bought a google drive package for the expansion of the storage from 15GB to 200GB. This allowed us to store our complete dataset, along with any modifications, dataframes, experimentations and final stage files (like the complete text features set, complete audio features set, etc.). After that it was really easy to mount google drive into google colab and directly manage and work with our dataset and extracted/saved files.

Concerning the hardware, all the work took place from a DELL laptop with the following specs:

- RAM: 16 GB
- Processor: Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz 1.99 GHz
- OS: 64-bit
- Windows: 11 Home
- Storage: 256 GB SSD

Since our dataset, dataframes and other extracted files amounted to about 140 GB, it was only natural to find an efficient solution for storing them. And google drive cleaned our hands of this issue.

6 Experimental Results

6.1 Text Features Discussion

When the feature elimination and selection process were complete, we observed that certain features stood out across all methods and settings. First of all, we noticed a consistency in the appearance of the *'anx', 'sad', 'they'* and *'death'* LIWC categories. This indicates that these psychological and thematic aspects of the text are highly relevant to the identification of mental health disorder markers. Their consistent presence underscores the significance of emotional and thematic content in the analysis. Apart from LIWC categories, there were also some GloVe dimensions that were repeatedly selected. This consistency of certain GloVe dimensions suggests that they capture key semantic features relevant to the identification of language markers associated with mental health diseases. Another feature that was prevalent, across the various feature selectors, was *PCA2*. In the context of Principal Component Analysis (PCA), the second principal component (PCA2) accounts for the next highest variance after the first principal component (PCA1). The fact that PCA2 appears more than PCA1 implies that PCA2 captures significant aspects of the data that are not captured by PCA1. Finally, concerning our last type of textual features, POS Tag counts, we noticed that *'VBG_count'* (verb, gerund or present participle) and *'JJR_count'* (comparative adjective) both belonged to the top 15 features selected across all feature selectors. The frequency of the first one points to the syntactic structures of sentences as informative features, while the frequency of the latter suggests that certain grammatical constructs may play a role in distinguishing texts related to mental health. Overall, features that appear consistently tend to be less sensitive to variations in the modeling process or data sampling, making them reliable choices for critical analyses.

Verifying the importance of GloVe Dimensions

Due to the high-dimensionality and abstractness of the GloVe embeddings, it is considerably complex and hard to map the glove dimensions with a specific semantic theme or words. Each dimension in these embeddings doesn't necessarily correspond to a human-interpretable feature or concept. Instead, each dimension contributes to capturing semantic and syntactic patterns based on the model's training on large text corpora. The dimensions together create a space, where the relationship between words can be mathematically examined through operations like vector addition and cosine similarity and potentially lead to the uncovering of hidden patterns. During the particular research, to verify the importance of gloves, we performed a series of experimentation involving the addition and subtraction of the top 15 (most commonly selected) glove dimensions from the final text features set. The first experimenting round involved starting with no GloVe features and adding each time only a single glove dimension (out of 15) and observing the variance on the performance of our models (Appendix Table 20). Doing that, we also managed to identify the top 4 glove dimensions that had the largest impact on the performance; i.e. glove dimensions 9, 36, 64 and 71. Then, we proceeded with the second round of experimentation, which included a performance comparison with the original features set and a features set that excluded all glove dimensions (Appendix tables 21 and 22). Once again, it was clearly shown that the presence of GloVe embeddings was crucial for our models as their presence was bringing upon an incline in performance ranging between 5 and 12%. The models' performance was not only enhanced in terms of accuracy but also in terms of AUC-ROC score, especially in the case of Logistic Regression. Furthermore, unlike accuracy the impact of GloVe embeddings for AUC-ROC was even greater, since we observed a gap between 15 and 20% between the models that included gloves and those that didn't. Our conclusion from these experiments was that the semantic information carried by these features are more than beneficial in our case. More specifically, the huge variance in AUC-ROC scores, which is crucial for classification tasks that focus on the distinction between two classes, proved the importance of the GloVe dimensions.

This is a good moment to present the top 20 text features of our top performing feature selector, the *RFE with Logistic Regression with re-scaling*. The following table (Table 11) shows the features from most to least important; meaning that if someone wanted to extracted the top 10 features they could simply selected the first 10 inputs.

Feature	**Feature Type**	**Description**
posemo	LIWC category	Indicates words that express positive emotions
anx	LIWC category	Captures words related to anxiety
sad	LIWC category	Captures words related to sadness
death	LIWC category	Includes words related to death and dying
glove_dim_9	GloVe embeddings	Specific dimension of GloVe embeddings
glove_dim_63	GloVe embeddings	Specific dimension of GloVe embeddings
glove_dim_64	GloVe embeddings	Specific dimension of GloVe embeddings
glove_dim_71	GloVe embeddings	Specific dimension of GloVe embeddings
VBG_Count	POS-Tag counts	Count of gerunds (verb forms ending in -ing)
RB_Count	POS-Tag counts	Count of adverbs
glove_dim_33	GloVe embeddings	Specific dimension of GloVe embeddings
glove_dim_46	GloVe embeddings	Specific dimension of GloVe embeddings
glove_dim_87	GloVe embeddings	Specific dimension of GloVe embeddings
JJR_Count	POS-Tag counts	Count of comparative adjectives
WP_Count	POS-Tag counts	Count of WH-pronouns (e.g., who, what)
glove_dim_10	GloVe embeddings	Specific dimension of GloVe embeddings
glove_dim_36	GloVe embeddings	Specific dimension of GloVe embeddings
glove_dim_83	GloVe embeddings	Specific dimension of GloVe embeddings
glove_dim_89	GloVe embeddings	Specific dimension of GloVe embeddings
RBS_Count	POS-Tag counts	Count of superlative adverbs

Table 11: Top 20 text features recommended by RFE w/ LogReg w/ re-scaling.

Similarly, Table 12 illustrates the top 15 text features across all selector approaches. In this table we also include a column called 'Count', which indicates the frequency of the particular feature. The max count one feature could have is 11. For further understanding visit Table 7. From that table, we excluded the features selected by 'RFECV with RF' and we only kept the features selected without re-scaling from the 'RFE with RF' selector, since they were exactly the same with and without re-scaling.

Comparing the features of our top selector (i.e. the selector with the best model performance) with the most popular features across all selectors we observed that 10 out of the 15 most popular features were also recommended by the top selector approach. On the other hand, we can also see that there are two interesting cases of features that weren't included in the best performing features set. We are referring to *'they'* and *'PCA2'*, both of which appear in 8 out of the 11 features sets. Another fact that must be noted is the frequency of the *'death'* LIWC category as a selected feature. The specific feature appears 10 out of 11 times and obviously indicates its importance when it comes to the identification of mental health disorder markers.

6.2 Audio Features Discussion

After the completion of the feature selection experiments, we noticed that across all different methods and settings, certain features (particularly from the MFCCs and formant features) show a consistency in appearing as significant. This suggests that there may be an impor-

Feature	Presence count (out of 11)
death	10
glove_dim_9	9
they	8
PCA2	8
glove_dim_63	7
JJR_Count	7
anx	6
glove_dim_47	5
glove_dim_87	5
sad	5
glove_dim_64	5
glove_dim_38	5
glove_dim_30	4
VBG_Count	4
glove_dim_46	4

Table 12: Top 15 text features across all RFE settings.

tant relationship between these features and the target variable across both linear (LogReg) and non-linear (RF) model perspectives. Another notable fact that we noticed is that pitch-related features were more prominently selected by the RF model and specifically with the RFE method. This indicates that the relationship between pitch features and the target variable can probably be captured more effectively compared to linear models (at least in some contexts). Further weight was given in the research of pitch-related features, as they were discussed a lot during related projects. This, in association with the lack of pitch-related features selected by the Logistic Regression model, lead us to conclude that the linear nature of the particular model may not always capture the complex ways in which the specific features contribute to the specific classification task. It's probably why, Random Forest, being a non-linear model, might be better in capturing such complexities and interactions; for instance if pitch interacts with other features in a way that doesn't lend itself to linear separation.

The following table (13) showcases the top 20 audio features recommended by our best performing selector approach, the *RFE with RF*. This table follows the same concept as Table 11.
Finally, Table 14 shows the top 15 audio features across all feature selectors. In this case the maximum count is 10. Along with the exclusions mentioned in the respective text-features table, we also excluded the features selected by 'RFECV with LogReg', which recommended only a single feature.

Placing the audio features of the above two tables (tables 13 and 14) next to each other, we can see that 10 out of the top 15 (overall) features match with the features of our best performing features set. What's noteworthy, is the fact that *'stf_F1'* appears in every single feature selector (with no exception), which implies that the particular feature is potentially crucial for the identification of mental health speech markers. Similarly important seems to be *'MFCC_median_3_std'*, which appears 8 out of 10 times. Furthermore, unlike the case of the

Feature	Feature Type	Description
MFCC_median_4_mean	MFCCs	Mean of all median values of 4th coef.
MFCC_std_11_std	MFCCs	Std of all std values of 11th coef.
MFCC_mean_13_median	MFCCs	Median of all mean values of 13th coef.
pitch_std_mean	Segmented	Mean of all std values of pitch
pitch_std_median	Segmented	Median of all std values of pitch
shimmer_local_std	Segmented	Std of all shimmer local values
hnr_std	Segmented	Std of all HNR values
median_F1	Formant	Median of the 1st formant F1
std_F1	Formant	Std of the 1st formant F1
std_F5	Formant	Std of the 5th formant F5
MFCC_median_4_median	MFCCs	Median of all median values of 4th coef.
MFCC_mean_11_median	MFCCs	Median of all mean values of 11th coef.
MFCC_mean_11_std	MFCCs	Std of all mean values of 11th coef.
MFCC_mean_13_mean	MFCCs	Mean of all mean values of 13th coef.
std_F2	Formant	Stf of 2nd formant F2
MFCC_mean_3_std	MFCCs	Std of all mean values of 3rd coef.
MFCC_median_3_std	MFCCs	Std of all median values of 3rd coef.
pitch_mean_median	Segmented	Median of all mean values of pitch
pitch_median_mean	Segmented	Mean of all median values of pitch
jitter_local_std	Segmented	Std of all jitter local values

Table 13: Top 20 audio features recommended by RFE w/ Random Forest.

Feature	Presence count (out of 10)
std_F1	10
MFCC_median_3_std	8
MFCC_median_6_std	6
median_F1	6
MFCC_median_9_median	6
shimmer_apq5_std	6
energy_median_mean	5
MFCC_median_8_std	4
MFCC_mean_11_median	4
MFCC_std_11_std	4
MFCC_mean_13_median	4
pitch_std_mean	4
pitch_std_median	4
shimmer_local_std	4
hnr_std	4

Table 14: Top 15 audio features across all RFE settings.

text features, it is obvious here that there isn't a steady decline in frequency; meaning that the feature selectors provide relatively different sets of features, with large deviations.

6.3 Unimodal model (Text) results

In Table 15 we are showcasing the results achieved by our models, using the features set recommended by the our best performing selector approach; i.e. RFE with Logistic Regression and with additional scaling. Like we discussed in the *Experimental Design* section, we implemented a cross-validation approach for the modeling part. Since every fold of the cross validation produced a different accuracy and AUC-ROC score, we modified our script so that it returned the mean cv values of all folds. As such, in the following table (15), 'Accuracy' represents the *mean CV accuracy* and 'AUC-ROC' represents the *mean CV AUC-ROC score* across all folds. Moreover, 'F1 - 0s' and 'F1 - 1s' represent the F1 scores achieved when predicting the absence and the presence of mental disorder markers, respectively. For the three machine learning models we used a random state seed, so regardless of how many times we reran the models the results didn't deviate much from the ones presented in the table. On the other hand, in the case of the deep learning model, every run produced different results (with observable variance) and as such we are providing the best scores out of four runs for each set of features.

Model	**Features**	**Accuracy**	**AUC-ROC**	**F1 - 0s**	**F1 - 1s**
SVM	10	80.40%	87.09%	0.86	0.67
SVM	15	81.97%	89.38%	0.87	0.70
SVM	20	**86.77%**	**93.33%**	**0.91**	**0.78**
SVM	25	84.68%	92.85%	0.89	0.76
RF	10	78.27%	85.57%	0.85	0.60
RF	15	83.60%	85.16%	0.89	0.69
RF	20	78.28%	82.09%	0.86	0.57
RF	25	**85.72%**	**91.75%**	**0.90**	**0.73**
LogReg	10	80.92%	87.94%	0.86	0.68
LogReg	15	83.57%	89.52%	0.88	0.73
LogReg	20	**87.82%**	92.44%	**0.91**	**0.79**
LogReg	25	85.72%	**93.95%**	0.90	0.76
Dense Layers	10	**84.65%**	88.94%	0.88	0.73
Dense Layers	15	82.05%	87.81%	0.88	0.72
Dense Layers	20	83.57%	88.01%	0.87	0.67
Dense Layers	25	84.11%	**91.79%**	**0.89**	**0.74**

Table 15: Unimodal Text results based on RFE w/ LogReg w/ rescaling.
F1-0s refers to the F1 score related to the absence of mental disorder markers and F1-1s to the presence of such markers.

Based on the presented results, we concluded that each model works best with a different number of features. SVM with a linear kernel and Logistic Regression both perform best with 20 features and it is also clear that they achieve similar scores in all metrics. On the other hand, when tested with 25 features the performance drops and when tested with 30 features the performance dropped even further. This means that the presence of overfitting became more intense and that the addition of more features was redundant. In the case of Random Forest and Dense Layers, the models performed their best across all metrics (with a small exception at the accuracy of the Dense Layers) when given 25 features. Both Random Forest and Dense Layers benefit from having more features. As an ensemble learning method, Random Forest, can use the larger number of features to create more informative splits across its decision trees, which also explains the considerably better score at the AUC-ROC score. Similarly, neural networks can use larger sets of features to learn more complex patterns because of their capacity to handle and learn from additional features. However, when tested with a set of 30 features (using the same feature selector), every model under-performed in every single metric. This implies that after a specific point the models cannot generalize as well.

Usually, as long as normalization has already been performed, additional scaling is a useless process. However, it appears that in the case of Logistic Regression, this re-scaling proves to be a great addition, as it performs much better than the other selector approaches (see Appendix tables 23, 24, 25 and 26). Overall, given the balance between complexity, interpretability, and performance, the 'RFE with Logistic Regression with additional scaling' approach is recommended. Not only does it provide the highest Mean CV AUC-ROC scores but also maintains a high level of accuracy. It signifies a strong model that can generalize well while retaining interpretability of features.

6.3.1 Overfitting results

Following the overfitting methodology discussed in earlier sections, we attempted to identify which models involve overfitting; even if only a low level. Our results showed that Logistic Regression and SVM only show minimal signs of overfitting, while Random Forest came along with a more intense level of overfitting. In the case of the Dense Layers, the model showed clear overfitting signs with the 20 and 25 features set, but only a minimal amount with the 10 features set (the gap between the train and test sets scores was really small). With the 25 features, Dense Layers achieved a score of 100% at accuracy and AUC-ROC on the train set, while on the test set the model performed about 10% worse. Overall, with the exception of the Random Forest model, while the training scores are slightly higher than the testing scores (which is common in machine learning models), the differences are not large enough to indicate a serious overfitting problem.

6.4 Unimodal model (Speech) results

In the case of the unimodal audio model, the various features sets recommenced by the feature selectors achieved very similar scores with each other and it was actually harder to distinguish one over the others. This is why we decided to go with the selector approach that achieved the overall highest results across all three metrics. This was the 'RFE with RF' selector, who recommended the same features regardless of any additional scaling. The columns of Table 16 represent the same values as with the corresponding table (15) of the unimodal text model.

By observing the results it's noticeable that the models performed best with the same feature set sizes, as in text. SVM had the best results with the 20 features set, but it did surprisingly bad in the case of predicting marker presence (F1 score of 1s). With 10 and 15 features, SVM actually had a score of 0 on this metric. This can be attributed to various reasons, like overfitting or feature selection impact. It's possible that SVM might be overfitting to the majority class (ignoring the minority class entirely), or that the feature selection might not be sufficiently informative for the particular model to distinguish between classes. Logistic Regression,

Model	Features	Accuracy	AUC-ROC	F1 - 0s	F1 - 1s
SVM	10	68.08%	57.68%	0.81	0.0
SVM	15	68.08%	53.86%	0.81	0.0
SVM	20	**68.61%**	**61.58%**	**0.81**	**0.12**
RF	10	70.77%	74.56%	0.80	**0.46**
RF	15	**71.83%**	**75.20%**	**0.82**	0.4
RF	20	68.62%	70.95%	0.80	0.31
LogReg	10	**68.62%**	**59.12%**	**0.80**	0.21
LogReg	15	67.55%	56.85%	0.80	0.19
LogReg	20	67.03%	58.20%	0.79	**0.23**
Dense Layers	10	**69.70%**	**68.23%**	**0.80**	**0.46**
Dense Layers	15	66.56%	60.63%	0.77	0.33
Dense Layers	20	64.40%	65.99%	0.77	0.35

Table 16: Unimodal Audio results based on RFE w/ RF

although comparable with SVM at the other metrics, managed to achieve approximately double the score of SVM's 'F1 score - 1s'. Dense Layers did better than SVM and LogReg, especially with 10 features and it actually achieved the best score on 'F1 - 1s', along with RF. Both LogReg and Dense Layers performed the best with the 10 features set. Finally, the RF model got the best scores with the 15 features, with the single exception of 'F1 - 1s', which appeared to be higher with less features (10). It was also interesting how RF performed the best across all models and metrics. We hypothesize that this is due to the fact that the feature sets used were picked by the selector that used RF as its classifier. It is noteworthy to mention that the particular results of RF (on 15 features) were the highest results (for every single evaluation metric) achieved by the unimodal audio model across all feature selectors. The results of all the selectors are available in the Appendix (tables 27, 28 and 29) For reference, unlike the unimodal text model, the audio model under-performed with any additional features (tested on 25 features set as well). This indicates that our models cannot handle and learn the same with the acoustic features, as they did with the textual ones.

6.4.1 Overfitting results

The overfitting experiments indicated that Logistic Regression had an increasing level of overfitting, the more features we added to the features set. Yet, with the 10-features set the overfitting level was really low. Random Forest, in this case as well, showed clear signs of overfitting, while the mild overfitting appearing in SVM can be probably attributed to the model's simplicity. Finally, Dense Layers showed high levels of overfitting with the 15-features set (90% accuracy achieved with train set and barely 70% with test set) and a bit lower level with the 10 features.

6.5 Multimodal model (Text + Speech) results

Table 17 showcases the results achieved by the combination of the text and speech modalities. For the multimodal model, we selected the top 20 features of the top performing feature selectors; i.e. *RFE with Logistic Regression with re-scaling* for textual features and *RFE with Random Forest* for acoustic features. In the following table, *'t'* stands for textual features and *'a'* for acoustic. The number in front of the letter reflects the number of respective features.

Model	**Features**	**Accuracy**	**AUC-ROC**	**F1 - 0s**	**F1 - 1s**
SVM	20t, 15a	86.17%	**92.80%**	0.91	0.79
SVM	20t, 10a	**86.71%**	92.74%	**0.92**	**0.80**
SVM	15t, 15a	82.97%	90.61%	0.89	0.72
RF	20t, 15a	79.80%	84.45%	0.87	0.60
RF	20t, 10a	**80.87%**	**86.39%**	**0.87**	**0.63**
RF	15t, 15a	79.82%	84.38%	0.87	0.60
LogReg	20t, 15a	85.14%	91.05%	0.89	0.77
LogReg	20t, 10a	**86.73%**	**92.36%**	**0.90**	**0.80**
LogReg	15t, 15a	84.57%	91.01%	0.89	0.74
Dense Layers	20t, 15a	**84.59%**	89.55%	**0.90**	**0.76**
Dense Layers	20t, 10a	84.04%	**90.22%**	0.89	0.74
Dense Layers	15t, 15a	84.07%	89.91%	0.88	0.70

Table 17: Multimodal results based on combined text-audio feature selectors

The presented results indicate that the '20t, 10a' features set works great with our models. In all cases, with a couple of exceptions, the particular set pulls off the highest results. The exceptions include the AUC-ROC score achieved by the '20t, 15a' features set on the SVM model, which is the highest one observed across all cases, and the case of the Dense Layers. Dense Layers appear to work best (overall) with the '20t, 15a' set, although the score gap with the other two sets isn't that noteworthy. The '20t, 15a' features set is the largest one out of the 3, consisting of 35 features, instead of 30. This could possibly mean that Dense Layers can be effectively trained on larger feature sets and even improve their attained results further.

In the case of the multimodal model, we decided to experiment further and as such we applied *GridSearch* on our SVM and RF models in order to identify the best parameters (see Appendix Table 30). Also, since we had observed some signs of overfitting in our models, both in this case and in the unimodal approaches, we decided to experiment with a regularization technique on our Logistic Regression model. Concerning the Dense Layers, the model that achieved the results on Table 17 was already equipped with *Dropout* layers to prevent/reduce overfitting. Table 18 presents the attained results after applying the aforementioned techniques. For SVM and RF, we only present the scores of the top performing features set.

Model	**Features**	**Accuracy**	**AUC-ROC**	**F1 - 0s**	**F1 - 1s**
SVM	20t, 10a	86.71%	92.74%	0.91	0.80
RF	20t, 10a	81.39%	85.39%	0.87	0.64
LogReg	20t, 15a	71.27%	85.59%	0.83	0.18
LogReg	20t, 10a	71.27%	86.24%	0.83	0.18
LogReg	15t, 15a	70.20%	87.24%	0.82	0.12

Table 18: Additional experimentation with GridSearch and Regularization

In the case of SVM, the post-GridSearch results showed no improvement. Additionally, we can even notice that the 'F1 - 0s' score is barely lower than before. On the other hand, for the RF model, it is interesting how two out of the four metrics seem to improve, even if only in a small degree. The mean fold accuracy is increased by approximately 0.5% and the F1 score predictor for the marker presence is increased by 0.01. At the same time, the results of the Logistic Regression model after applying regularization appear to have dropped greatly across all metrics and feature set sizes. What's worse is that the F1 scores calculating the marker's presence have decreased by 60%. This huge downfall can be probably attributed to the regularization method. By studying the results though, we noticed that the reason behind the low F1 scores is the low recall rate. For each features set the model has a precision rate of 100%, while the recall rate ranges between 7% and 10%.

Modality - Model	**Features #**	**Accuracy**	**AUC-ROC**	**F1 - 0s**	**F1 - 1s**
Text - SVM	20	86.77%	93.33%	0.91	0.78
Audio - SVM	20	68.61%	61.58%	0.81	0.12
Multimodal - SVM	20t, 10a	86.71%	92.74%	0.92	0.80
Text - RF	25	85.72%	91.75%	0.90	0.73
Audio - RF	15	71.83%	75.20%	0.82	0.40
Multimodal - RF	20t, 10a	80.87%	86.39%	0.87	0.63
Text - LogReg	20	87.82%	92.44%	0.91	0.79
Audio - LogReg	10	68.62%	59.12%	0.80	0.21
Multimodal - LogReg	20t, 10a	86.73%	89.55%	0.90	0.80
Text - FCNN	25	84.11%	91.79%	0.89	0.74
Audio - FCNN	10	69.70%	68.23%	0.80	0.46
Multimodal - FCNN	20t, 15a	84.59%	89.55%	0.90	0.76

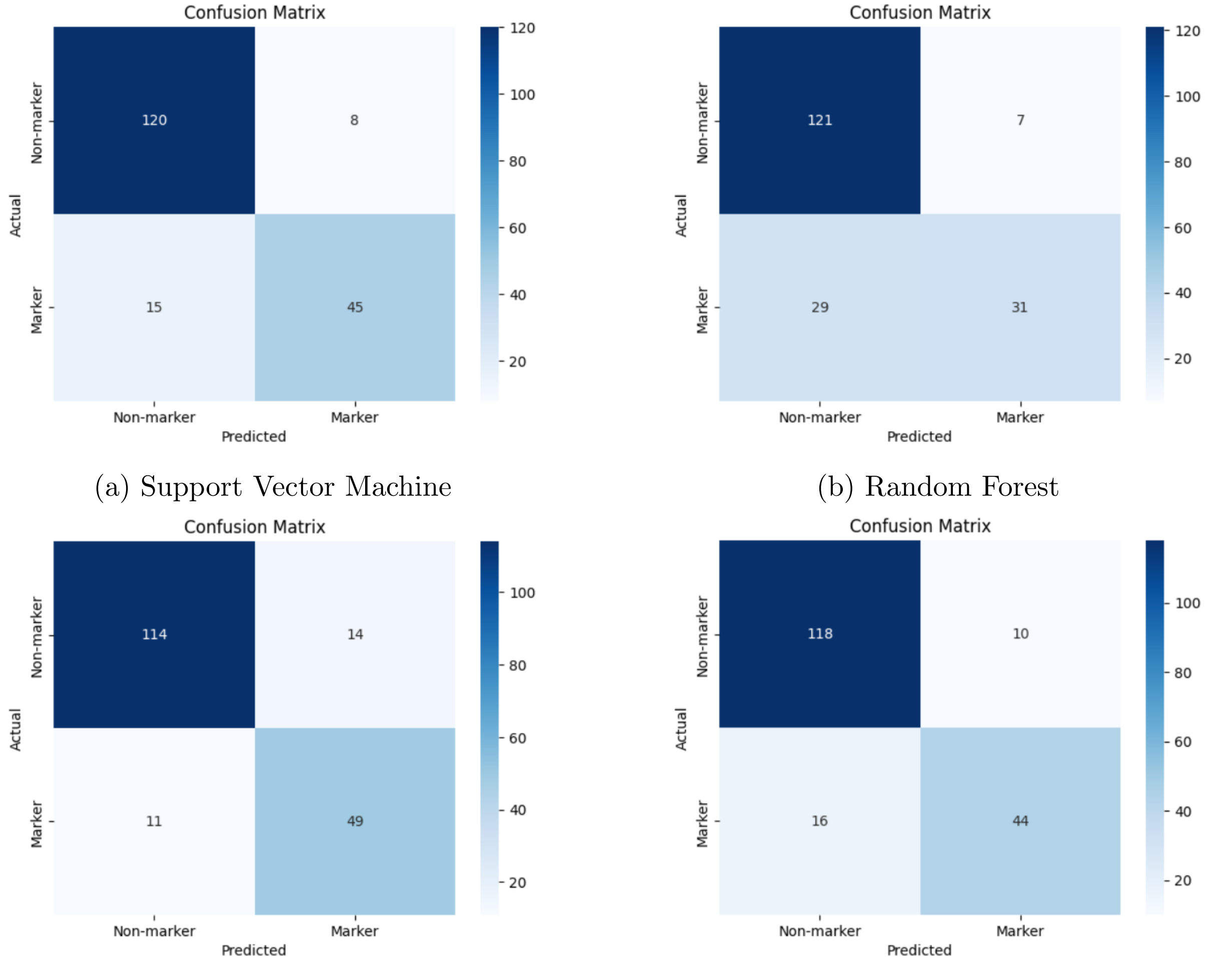
Table 19: Results comparison between Unimodal and Multimodal approaches, per ML model.
FCNN stands for Fully Connected Neural Network (Dense Layers in this case)

6.5.1 Overfitting

Just like with the unimodal approaches, the RF model shows high levels of overfitting once again. We discovered that the RF model achieves a 100% accuracy and AUC-ROC scores on the train sets, while the results presented on Table 17 are the ones of the test set. This gap is a clear indicator of the presence of overfitting. Unlike RF, SVM only indicates minimal overfitting on the '20t, 15a' and '20t, 10a' features sets and mild overfitting on the '15t, 15a' set. Finally, in the case of logistic regression, regularization leads to mitigated levels of overfitting by simplifying the model and thereby improving its generalization ability, as evidenced by closer training and testing scores. On the other hand, without regularization, the overfitting levels are slightly higher (yet still manageable).

6.5.2 Confusion Matrix

In Figure 5, we present the confusion matrices that are plotted on the *20t, 10a* features set, for all studied ML methods.



(a) Support Vector Machine

(b) Random Forest

(c) Logistic Regression

(d) Dense Layers

Figure 5: Confusion Matrices of all models

The results presented in this section were generated using the scripts available in our GitHub repository: GitHub Repository.

7 Discussion

7.1 Discussion on Unimodal Models' Results

Observing the performances of the two unimodal approaches, it is more than obvious that the unimodal text model is by far outperforming the unimodal audio model. More importantly, this gap is even more intense in some of the classifiers. In the case of SVM and Logistic Regression, the text model achieves an $\approx 18\%$ higher accuracy and $\approx 32\%$ higher AUC-ROC score than the audio model, while the F1 score of 1s is higher by more than 60%. Similarly, for Dense Layers the audio model attains an accuracy $\approx 15\%$ lower and an AUC-ROC $\approx 20\%$ lower than the text model. On the other hand, in the case of Random Forest, which shows the best performance for the audio model, the gap in accuracy is only $\approx 12\%$ and the gap in AUC-ROC score is barely 10%. Concerning the F1 score of 1s metric however, we can still observe a performance difference of more than 20%.

Our hypothesis, which is based on our observations and results, is that the huge performance gap is owed to the process of creating our binary labels. Since that process was based on the LIWC categories, which are a text feature, it makes sense for the text features to be more accurate during predictions and for the audio features to encounter some difficulties. Still, we consider it weird that the audio models achieved such low scores. However, we strongly believe that the text model would perform better regardless of this factor. On that note, assigning binary labels differently, for instance with the help of clinical experts, could potentially lead to better predictions by the audio features as well.

It's not rare for text features to outperform audio features in tasks related to the particular topic. When it comes to identifying mental disorder markers, the text modality has proven to be extremely capable of leading to better predictions, even more so when there is a relevant textual content that offers clear linguistic markers. Although the text model significantly outperforms the audio model, we should still not diminish the value provided by the audio features. Analyzing these features can help discover distinct and complementary insights and this is where multimodal models, that utilize both of these modalities, can shine. By combining the strengths offered by each modality and implementing fusion techniques that bring forth those strengths, it is possible to capture more comprehensive information and details of mental health states.

7.1.1 Challenges

It's also noteworthy to mention two more audio features that have been marked through related work as very useful indicators when it comes to the identification of speech markers for mental health. *Speech rate* (how fast or how slow someone speaks) and *pause duration* (how much one pauses during their speech) are features that should have been part of the unimodal audio model. The calculation of these two features requires either a *text-speech alignment* process, which we decided to avoided in order to keep our model as pure unimodal as possible, or experimenting with *zero-crossing rate*, which can be used to distinguish voiced and unvoiced segments (specifically ZCR measures the rate at which a signal changes from a positive to a negative sign and vice versa). Although the extraction of the two aforementioned acoustic features could be possible with either of these two approaches, we decided to avoid any additional complexity at this stage of our research. Instead we moved on with the already extracted features of each modality.

Once our binary labels creation process was complete we decided to compare our approach's marker presence (assigned 1) with the marker presence (1 again) that was previously assigned to the DAIC-WoZ dataset by the PHQ-8 questionnaire. The PHQ-8 questionnaire was created by expert clinicians and is particularly focused in diagnosing depression [44]. This questionnaire can be accessed through `https://www.childrenshospital.org/sites/default/files/2022-03/PHQ-8.pdf`. The outcome of the comparison showed that our marker-presence labels matched the depression-presence labels of the PHQ-8 test at 45% of the cases for the train set and 50% of the cases for the test set.

Although our comparison with the assigned values of the PHQ-8 questionnaire indicated that some of our labels may be off target, we still had to take into account some factors that could have led to this variance. One of those factors was the fact that this questionnaire is focused solely on depression while our own research focuses on a much bigger list of mental health disorders. This means that in our case there were far more '1s' assigned overall, since we were also attempting to identify markers of different mental illnesses. Another factor was the deep level of expertise behind the creation of the PHQ-8. The questionnaire may have been created with a broader criterion when identifying the possibility of depression. The 50% overlap indicates that some of the mental health disorders, captured by our labeling system, may not be severe enough for the PHQ-8 to classify them as clinical depression. This discrepancy between the two labeling approaches could be further attributed to the complexity that mental health holds overall. Not all people with depression effectively show identifiable language or speech markers. Moreover, a few people might even exhibit distress signs or different mental disorder markers without reaching the clinical threshold for depression, more so in our case where we probably set different thresholds.

7.2 Discussion on Multimodal Model

As indicated by the results on Table 17, the best scores are attained by the *20t, 10a* set of features, with a few exceptions appearing with the *20t, 15a* set. However, there is not a single instance of the *15t, 15a* features set pulling off the best results over both of the other two sets. It is evident that assigning additional weight over the textual features brings forwards a boost in the models' performance. By emphasising more on the text features our multimodal model learns to generalize better. This was already indicated earlier, through the big gap between the text model's and the audio model's results, but the latest comparison even proved this observation. Regardless, experimenting with a modality balanced features set proved helpful in realizing the flaws and the correct steps in the whole modeling approach.

7.2.1 Challenges, Limitations

One of the challenges that we overcame was avoiding the loss of modality-specific insights. When merging features at an early stage, it sometimes becomes challenging to discern which modality is contributing to predictions and this can lead to reduced model interpretability. However, in our case, by creating three different sets of combined features (one of which represented the absolute balance between the two modalities), we showed that for the majority of the models the best performing features set was the one with the 66% weight on the textual features. Then, for some of the models, the challenger was the features set with the slightly extra weight on textual features. Yet, in none of the models did the balanced features set perform better than the other two. This clearly indicates that the textual features have more insight to offer in the identification of mental disorder markers, at least on the individual level.

Risk of feature dominance posed another challenge during the early fusion. Although, giving additional weight to the better performing text features (compared to the acoustic features) proved that it increased the performance of our models, there could have appeared an imbalance in feature contribution. It's possible for the text modality to have dominated the feature importance, overshadowing the audio modality. If this was indeed the case then we would have ended up with partly biased models. However, we believe that we have countered this challenge by performing modality-specific pre-processing. We followed the exact same scale-balancing (normalization) techniques in every single feature during the development of each unimodal model. By doing that we guaranteed that all features (of both modalities) were on the same scale and their impact should be balanced such that any possible feature dominance would be prevented.

Another potential challenge was the choice of early level fusion. Although this approach untied our hands when facing problems with text-speech alignment, it also came with a set of possible dangers. Feature level fusion increases dimensionality, and when increasing the parameter 'space' the possibility of overfitting actually increases. Especially in cases like ours, where the amount of training data was limited. We are referring to this challenge as 'potential' because in our case the results showed that the number of features selected from each modality didn't have a direct impact on the level of overfitting. Instead, although overfitting was indeed there, its intensity varied from one model to another and in some models the level was minimal. However, this fusion approach also involves inherent limitations that might impact the model's effectiveness. One primary drawback is that early fusion does not account for the temporal dynamics between modalities, which can be crucial for tasks like the identification of mental disorder markers, where the timing of spoken words relative to vocal attributes might carry significant diagnostic information.

7.3 Closing Discussion

Although E-DAIC was a great asset for our project, it did limit our research because of its size and its specificity towards a couple of particular mental disorders. Moreover, because of the dataset's interval overlaps in the transcript we had to avoid text-speech alignment; a process that has a fundamental role in multimodality, especially between the particular two modalities. Another factor that could potentially alter the course of this project would be implementing a labeling system, verified by clinical experts or even better applied directly by such individuals. The presence of accurate binary labels would most probably influence the results of the unimodal audio model. This is also indicated by the fact that our binary labels matched the previously assigned PHQ-8 binary labels to a rate of approximately 50%. Finally, we believe that implementing models with increased complexity could help to capture more nuanced details that might have been missed by simpler models.

8 Conclusion

In this project multimodality is brought into comparison with two unimodal approaches in an inquiry to find out its capacity in identifying mental health disorder markers. Our objective was to pit these approaches against each other and see if the multimodal approach can outperform the other two. To do that effectively we create two different models, one corresponding to each modality, we extracted the corresponding features for each and then we tested their prediction capabilities on four distinct machine/deep learning models; Support Vector Machine, Random Forest, Logistic Regression and Dense Layers, and on three different evaluation metrics; accuracy, AUC-ROC and F1 scores. Between the two unimodal approaches, the text models outperformed all of the audio models in every single metric by a significant margin (Tables 15 and 16. Then, based on feature selection, we combined one set of features from each modality into a common features set that was consequently used to test the prediction accuracy of the multimodal model. As our fusion approach, we employed early fusion and as for the classifiers and evaluation metrics we used the exact same ones as for the unimodal models.

The first results from this study indicated that, even if slightly, the multimodal approach does indeed outperform the two unimodal approaches (results at Table 17). To be more direct, we are mainly comparing with the unimodal text approach, since the audio models did significantly worse. Although, the accuracy and AUC-ROC scores between the text models (Table 15) and the multimodal models (Table 17) were extremely comparable, the multimodal approach showed its strength in the 'F1 scores of 1s' metric, which calculates the ability of the models to predict the presence of a marker. In most cases the multimodal model outperformed the text model by 2-3%.

Led by our mentions at the 'Future Work' section and based on our project's outcomes we have concluded that by changing a few variables (like including the extraction of useful acoustic features-'speech rate' and 'pause duration'- and a different binary label creation process) we could achieve better results from our unimodal audio model and consequently improve the performance of the multimodal model. One of those influencing factors would be a more specified and relevant dataset.

Our research question was: 'Does a multimodal approach perform better than unimodal approaches in the identification of language/speech markers for mental health disorders?' Based on our findings, the answer is yes, multimodality can actually perform better than unimodal approaches when it comes to the identification of language/speech markers in a topic as sensitive as mental health disorders. Since this conclusion was drawn under our own circumstances, then we are confident that these results would be consistent (if not more obvious) if the same methodology was applied to a more suitable dataset that can handle text-speech alignment effectively.

9 Future Work

Although we selected feature-level fusion (early fusion) for this project, we recognize that it was primarily because it simplified avoiding the text-speech alignment process. Under different circumstances, we would prefer to implement text-speech alignment, which is a significant aspect of multimodality that facilitates a more comprehensive integration of the two modalities. Implementing this process would open more opportunities for experimenting with other fusion types. As explained more analytically through the 'Challenges' section, early fusion may oversimplify the interaction between the two modalities and it is less flexible in handling their distinct characteristics. Thus, if any follow up work is to be done on this research, we strongly advise the implementation of text-speech alignment and the extraction of the *speech ratio* and *pause duration* features. As related work has shown, these two features can provide further evidence on the presence of speech markers. Moreover, we recommend exploring hybrid and end-to-end fusion techniques, as they are both capable of countering some of the limitations present in early fusion.

Text-speech alignment would not only enable the extraction of additional important features (*speech ratio* and *pause duration*), but also the option of experimenting with deeper deep learning models, like GRUs and LSTMs. These models excel in processing sequences where the timing and order of inputs are crucial to understanding the data's context and dynamics. The alignment is particularly important in tasks where the synchronization of spoken words and vocal characteristics adds value to the interpretation. GRUs, for instance, are tailored to handle sequences by capturing dependencies at different time steps. In order for the GRU to effectively analyze the responses of the textual information to the speech variations (e.g. speech rate, pauses, intonation, etc), it is crucial for the two modalities to be properly aligned.

Other than the implementation of text-speech alignment we would also recommend experimenting with the $F_{0.25}$ metric, which is a variance of the F_β. F_β is a generalization of the F1 score that allows the alteration of the original weights between precision and recall. The equation for F_β looks like this:

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$$

In the $F_{0.25}$, β is set to 0.25, indicating that precision is considered more important than recall. This alteration of the original F1 metric is recommended in cases where the consequences of false positives are crucial. For instance, in our project, we might prefer falsely marking a few samples as marker-including (higher precision) over missing too many actual markers (lower recall).The specific equation for the $F_{0.25}$ score, emphasizing precision, is given by:

$$F_{0.25} = (1 + 0.25^2) \cdot \frac{\text{precision} \cdot \text{recall}}{0.25^2 \cdot \text{precision} + \text{recall}}$$

Another recommendation for future work is further implementation of dimensionality reduction techniques, like PCA and t-SNE, on the combined features set. This would decrease the dimensionality, while still maintaining variance, and would consequently lead to the mitigation of overfitting.

Furthermore, hybrid fusion is recommended as an alternative fusion approach because it combines aspects of both early and late fusion. For instance, some features may be combined at an early stage, while others may be merged after some additional individual processing. This way modality-specific processing is preserved (at least at some level) and the model can learn the distinct properties of each modality better.

On this note, end-to-end fusion involves the integration of modalities at a deep level, with a preference for using deep neural networks, which learn to extract and combine relevant features autonomously. Following this approach, we could for example feed all 300 features into the a deep neural network and have it recommend and extract the most optimal combination of features. This is even supported by the fact that this method can dynamically adjust to the modality and feature importance through training. It's particularly advantageous when there is a more complex and highly non-linear interaction between the modalities.

It's possible that exploring these two fusion approaches could allow for a more sophisticated handling of modalities, as well as lead to a potential improvement of the accuracy and robustness of identifying mental health disorder markers.

References

[1] Spruit, M., Verkleij, S., de Schepper, K., & Scheepers, F. (2022). Exploring language markers of mental health in psychiatric stories. *Applied Sciences (Switzerland)*, *12*(4), Article 2179. https://doi.org/10.3390/app12042179

[2] World Health Organization. (June 2022). Mental disorders. From https://www.who.int/news-room/fact-sheets/detail/mental-disorders

[3] Yazdavar, A. H., Mahdavinejad, M. S., Bajaj, G., Romine, W., Sheth, A., Monadjemi, A. H., Thirunarayan, K., Meddar, J. M., Myers, A., Pathak, J., & Hitzler, P. (2020). Multimodal mental health analysis in social media. PloS one, 15(4), e0226248. https://doi.org/10.1371/journal.pone.0226248

[4] Duong, C. T., Lebret, R., & Aberer, K. (2017, August 7). Multimodal Classification for Analysing Social Media. Retrieved from https://arxiv.org/abs/1708.02099

[5] Calvo, R., Milne, D., Hussain, M., & Christensen, H. (2017). Natural language processing in mental health applications using non-clinical texts. *Natural Language Engineering*, *23*(5), 649-685. https://doi.org/10.1017/S1351324916000383

[6] Zadeh, A., Liang, P., Vanbriesen, J., Poria, S., Tong, E., Cambria, E., Chen, M., & Morency, L. (2018). Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion. Retrieved from https://aclanthology.org/P18-1208/

[7] Zhang, T., Schoene, A. M., Ji, S., & Ananiadou, S. (2022, April 8). Natural language processing applied to mental illness detection: A narrative review. *npj Digital Medicine*, *5*, Article 46. https://doi.org/10.1038/s41746-022-00589-7

[8] Aleem, S., Huda, N. U., Amin, R., Khalid, S., Alshamrani, S. S., & Alshehri, A. (2022). Machine learning algorithms for depression: Diagnosis, insights, and research directions. *Electronics*, *11*(7), Article 1111. https://doi.org/10.3390/electronics11071111

[9] Cho, G., Yim, J., Choi, Y., Ko, J., & Lee, S. (2019). Review of Machine Learning Algorithms for Diagnosing Mental Illness. *Psychiatry Investigation*, *16*(4), 262-269. https://doi.org/10.30773/pi.2018.12.21.2

[10] Niquini, F. G. F., Brito Branches, A. M., Costa, J. F. C. L., Moreira, G. de C., Schneider, C. L., Araújo, F. C. de, & Capponi, L. N. (2023). Recursive Feature Elimination and Neural Networks Applied to the Forecast of Mass and Metallurgical Recoveries in a Brazilian Phosphate Mine. Minerals, 13(6), 748. https://doi.org/10.3390/min13060748

[11] Gu, Y., Yang, K., Fu, S., Chen, S., Li, X., & Marsic, I. (2018). Hybrid attention based multimodal network for spoken language classification. *Journal Name*, *Volume*(Issue), 2379-2390. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6217979/

[12] Bianciardi, B., Gajwani, R., Gross, J., Gumley, A. I., Lawrie, S. M., Moelling, M., Schwannauer, M., Schultze-Lutter, F., Fracasso, A., & Uhlhaas, P. J. (2023). Investigating temporal and prosodic markers in clinical high-risk for psychosis participants using automated acoustic analysis. *Early Intervention in Psychiatry*, *17*(3), 327–330. https://doi.org/10.1111/eip.13357

[13] Garoufis, C., Zlatintsi, A., Filntisis, P., Efthymiou, N., Kalisperakis, E., Garyfalli, V., Karantinos, T., Mantonakis, L., Smyrnis, N., & Maragos, P. (2021, July). An unsupervised learning approach for detecting relapses from spontaneous speech in patients with psychosis. In *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*. Athens, Greece. https://eprevention.gr/an-unsupervised-learning-approach-for-detecting-relapses-from-spontaneous-speech-in-patients-with-psychosis/

[14] De Boer, J., Voppel, A., Brederoo, S. G., Schnack, H., Truong, K. P., Wijnen, F., & Sommer, I. E. C. (2023). Acoustic speech markers for schizophrenia-spectrum disorders: A diagnostic and symptom-recognition tool. *Psychological Medicine*, *53*(4), 1302-1312. https://doi.org/10.1017/S0033291721002804

[15] Chung, J., & Teo, J. (2022). Mental health prediction using machine learning: Taxonomy, applications, and challenges. *Applied Computational Intelligence and Soft Computing*, *2022*, Article 9970363. https://doi.org/10.1155/2022/9970363

[16] Yin, P. L., Zhang, L., Wu, X. Y., Hou, W. S., Chen, L., Tian, X. L., & Wen, H. Z. (2020). Analyzing acoustic and prosodic fluctuations in free speech to predict psychosis onset in high-risk youths. In *Proceedings of the 42nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Enabling Innovative Technologies for Global Healthcare, 20-24 July 2020, Montreal, Canada (pp. 5575-5579). IEEE.

[17] Vergyri, D., Knoth, B., Shriberg, E., Mitra, V., McLaren, M., Ferrer, L., Garcia, P., & Marmar, C. (2015). Speech-based assessment of PTSD in a military population using diverse feature classes. In *Proceedings of Interspeech 2015* (pp. 3729-3733).

[18] Shen, Y., Yang, H., & Lin, L. (2022). Automatic depression detection: An emotional audio-textual corpus and a GRU/BiLSTM-based model. *arXiv*. https://arxiv.org/abs/2202.08210

[19] Yang, Y., Fairbairn, C., & Cohn, J. (2013). Detecting depression severity from vocal prosody. *IEEE Transactions on Affective Computing*, *4*(2), 142-150. https://doi.org/10.1109/T-AFFC.2012.38

[20] Espinola, C. (2022). Detection of major depressive disorder, bipolar disorder, schizophrenia, and generalized anxiety disorder using vocal acoustic analysis and machine learning. *https://doi.org/10.21203/rs.3.rs-648044/v1*

[21] Dey, J., & Desai, D. (2022). NLP based approach for classification of mental health issues using LSTM and GloVe embeddings. *International Journal of Advanced Research in Science, Communication and Technology*, *2022*, 347-354. https://doi.org/10.48175/ijarsct-2296

[22] Amanat, A., Rizwan, M., Javed, A., Abdelhaq, M., Alsaqour, R., Pandya, S., & Uddin, M. (2022). Deep learning for depression detection from textual data. *Electronics (Switzerland)*, *11*(5). https://doi.org/10.3390/electronics11050676

[23] Assan, J., Flannery, M., Gao, Y., Resom, A., & Wu, Y. (2019, March 21). Machine learning for mental health detection. Retrieved from https://digital.wpi.edu/pdfviewer/b8515p953

[24] Low, D., Bentley, K., & Ghosh, S. (2020). Automated assessment of psychiatric disorders using speech: A systematic review. *Laryngoscope Investigative Otolaryngology*, *5*(1), 96-116. https://doi.org/10.1002/lio2.354

[25] Yoo, H., & Oh, H. (2023). Depression detection model using multimodal deep learning. Retrieved from https://www.preprints.org/manuscript/202305.0663/v1

[26] Muzammel, M., Salam, H., & Othmani, A. (2021). End-to-end multimodal clinical depression recognition using deep neural networks: A comparative analysis. *Computer Methods and Programs in Biomedicine*, *211*. https://doi.org/10.1016/j.cmpb.2021.106433

[27] Broek, E., Sluis, F., & Dijkstra, T. (2010). Telling the story and re-living the past: How speech analysis can reveal emotions in post-traumatic stress disorder (PTSD) patients. In *Title of the Book or Conference Proceedings* (pp. 153-180). https://doi.org/10.1007/978-90-481-3258-4_10

[28] Burback, L., Brémault-Phillips, S., Nijdam, M., McFarlane, A., & Vermetten, E. (2023). Treatment of posttraumatic stress disorder: A state-of-the-art review. *Current Neuropharmacology*, *22*(4), 557-635. https://doi.org/10.2174/1570159X21666230428091433

[29] Fusaroli, R., Lambrechts, A., Bang, D., Bowler, D., & Gaigg, S. (2016). Is voice a marker for autism spectrum disorder? A systematic review and meta-analysis. https://doi.org/10.1101/046565

[30] Demouy, J., Plaza, M., Xavier, J., Ringeval, F., Chetouani, M., Périsse, D., Chauvin, D., Viaux, S., Golse, B., Cohen, D., & Robel, L. (2011). Differential language markers of pathology in autism, pervasive developmental disorder not otherwise specified and specific language impairment. *Research in Autism Spectrum Disorders*, *5*(4), 1402-1412. https://doi.org/10.1016/j.rasd.2011.01.026

[31] Coppersmith, G., Dredze, M., Harman, C., & Hollingshead, K. (2015). From ADHD to SAD: Analyzing the language of mental health on Twitter through self-reported diagnoses. In Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality (pp. 1-10).

[32] Iverach, L., & Rapee, R. (2014). Social anxiety disorder and stuttering: Current status and future directions. *Journal of Fluency Disorders*, *40*, 69-82. https://doi.org/10.1016/j.jfludis.2013.08.003

[33] Marmar, C., Brown, A. D., Qian, M., Laska, E., Siegel, C., Li, M., Abu-Amara, D., Tsiartas, A., Richey, C., Smith, J., Knoth, B., & Vergyri, D. (2019). Speech-based markers for posttraumatic stress disorder in US veterans. *Depression and Anxiety*, *36*(7), 607-616. https://doi.org/10.1002/da.22890

[34] Von Polier, G., Ahlers, E., Amunts, J., Langner, J., Patil, K., Eickhoff, S., Helmhold, F., & Langner, D. (2021). Predicting adult attention deficit hyperactivity disorder (ADHD) using vocal acoustic features. https://doi.org/10.1101/2021.03.18.21253108

[35] Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532-1543). Association for Computational Linguistics. https://aclanthology.org/D14-1162.pdf

[36] "Elbow Method for Optimal Value of K in KMeans." (n.d.). GeeksforGeeks. Retrieved from https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/

[37] "Silhouette Analysis." (n.d.). Scikit-Learn. Retrieved from https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

[38] Bai, Z., Lu, X., & Wei, J. (2021). Artificially Synthesising Data for Audio Classification and Segmentation to improve speech and music detection in Radio Broadcast. arXiv preprint arXiv:2102.09959. https://arxiv.org/abs/2102.09959

[39] Feinberg, D. R. (2022, January 1). Parselmouth Praat Scripts in Python. https://doi.org/10.17605/OSF.IO/6DWR3 and https://osf.io/qe9k4

[40] Boersma, P., & Weenink, D. (2023). PointProcess. In *Praat: doing phonetics by computer* [Manual]. Retrieved from https://www.fon.hum.uva.nl/praat/manual/PointProcess.html

[41] Chodroff, E. (n.d.). Montreal Forced Aligner. Retrieved from `https://eleanorchodroff.com/tutorial/montreal-forced-aligner.html`

[42] McAuliffe, M., & Sonderegger, M. (2024, February). English (US) ARPA acoustic model v3.0.0. Retrieved from `https://mfa-models.readthedocs.io/acoustic/English/English%20(US)%20ARPA%20acoustic%20model%20v3_0_0.html`

[43] Gorman, K., Howell, J., & Wagner, M. (2011). Prosodylab-aligner: A tool for forced alignment of laboratory speech. *Canadian Acoustics*, *39*(3), 192-193.

[44] Kroenke, K., Strine, T. W., Spitzer, R. L., Williams, J. B., Berry, J. T., & Mokdad, A. H. (2009). The PHQ-8 as a measure of current depression in the general population. *Journal of Affective Disorders*, *114*(1-3), 163–173. https://doi.org/10.1016/j.jad.2008.06.026

[45] DeepMind. (2024). Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. Retrieved from https://storage.googleapis.com/deepmind-media/gemini/gemini_v1_5_report.pdf

Appendices

APPENDIX A

Precision and Recall

In binary classification tasks with uneven class distribution, like in our case, precision and recall are two critical metrics. *Precision* measures the model's accuracy of positive class predictions. It is calculated as the number of correct positive predictions (true positives) against the total number of instances predicted as positive (both true and false positives).

Precision Formula:

$$Precision = \frac{True\,Positives\,(TP))}{True\,Positives\,(TP) + False\,Positives\,(FP)}$$

On the other hand, *recall* is a metric that calculates the classifier's ability to identify all the relevant cases within the dataset. It aims to find all the positive instances (true positives) from all the actual positives existing in the dataset. Recall is simply the ratio of the number of true positives against the number of positive instances (even those identified as negative).

Recall Formula:

$$Recall = \frac{True\,Positives\,(TP))}{True\,Positives\,(TP) + False\,Negatives\,(FN)}$$

Increasing precision decreases recall and vice versa. In some cases, one of the two metrics could possibly be more important than the other and consequently one could focus on giving more weight to that metric. In our research, for example, it could potentially be beneficial to prioritize precision, which could lead in minimizing the number of false positives (i.e. having less instances of negatives predicted as positives, *Type I error*).

APPENDIX B

GloVe Features - Experiment 1

As discussed in the *Experimental Results* section, the first experiment had to do with the inclusion of only a single glove dimension per run of the model. Since Logistic Regression showed better overall results in accordance with the specific features, we performed this experiment around this model. Moreover, we went with cross validated test sets, instead of the typical train-test split.

GloVe dimension	**CV Accuracy**	**CV AUC-ROC**
glove_dim_9	**77.23%**	**86.97%**
glove_dim_10	72.46%	75.85%
glove_dim_33	75.11%	71.61%
glove_dim_36	**78.82%**	**77.38%**
glove_dim_46	74.05%	73.16%
glove_dim_63	75.09%	74.13%
glove_dim_64	**77.18%**	**81.14%**
glove_dim_71	75.62%	**78.33%**
glove_dim_83	74.58%	72.97%
glove_dim_87	**76.16%**	71.75%
glove_dim_89	74.05%	73.99%

Table 20: GloVe impact - Experiment 1

APPENDIX C

GloVe Features - Experiment 2

The second experiment entailed a performance comparison between feature sets that included GloVe embeddings and features sets that excluded them. For this round of experiments we used Logistic Regression and Random Forest as our classifiers.

LogReg w/ GloVe dimensions		
Features	**CV Accuracy**	**CV AUC-ROC**
10	80.92%	87.94%
15	83.57%	89.52%
30	84.64%	93.72%
LogReg w/o GloVe dimensions		
10	75.09%	72.11%
15	72.48%	73.79%
30	79.35%	76.42%

Table 21: Logistic Regression with and without GloVes.

RF w/ GloVe dimensions		
Features	**CV Accuracy**	**CV AUC-ROC**
10	78.28%	85.57%
15	83.60%	85.16%
30	79.87%	89.02%
RF w/o GloVe dimensions		
10	71.92%	65.15%
15	74.55%	70.27%
30	84.10%	78.29%

Table 22: Random Forest with and without GloVes.

APPENDIX D

Unimodal Text model - Feature Selectors experiments

The following tables illustrate the results achieved from the other feature selector approaches. These results can be used for comparison or for further reference.

Model	Features	Accuracy	AUC-ROC	F1 - 0s	F1 - 1s
SVM	10	79.90%	85.75%	0.86	0.66
SVM	15	82.02%	85.77%	0.87	0.70
SVM	20	**83.06%**	**92.09%**	**0.88**	**0.73**
RF	10	81.47%	90.44%	0.87	**0.67**
RF	15	**82.52%**	89.96%	**0.88**	0.65
RF	20	81.48%	**90.97%**	**0.88**	0.64
LogReg	10	75.85%	86.54%	0.85	0.64
LogReg	15	**82.01%**	87.42%	**0.87**	**0.70**
LogReg	20	80.95%	**91.59%**	0.86	0.68
Dense Layers	10	80.44%	88.98%	**0.87**	**0.70**
Dense Layers	15	80.44%	88.56%	0.86	0.69
Dense Layers	20	**84.11%**	**89.33%**	**0.87**	**0.70**

Table 23: **Unimodal Text results based on RFE w/ LogReg w/o rescaling**

Model	Features	Accuracy	AUC-ROC	F1 - 0s	F1 - 1s
SVM	10	**77.24%**	**78.09%**	**0.84**	**0.60**
SVM	15	75.68%	77.23%	0.83	0.56
SVM	20	75.09%	75.36%	0.83	0.55
RF	10	**86.76%**	91.93%	**0.91**	**0.77**
RF	15	84.64%	92.51%	0.89	0.73
RF	20	84.10%	**92.40%**	0.89	0.72
LogReg	10	**77.77%**	**79.58%**	**0.84**	**0.61**
LogReg	15	76.20%	78.38%	0.83	0.59
LogReg	20	77.23%	78.68%	0.84	0.58
Dense Layers	10	**80.44%**	81.75%	**0.86**	**0.64**
Dense Layers	15	80.41%	**82.28%**	0.83	0.58
Dense Layers	20	78.31%	81.61%	0.84	0.63

Table 24: **Unimodal Text results based on RFE w/ RF w/o rescaling**

Model	Features	Accuracy	AUC-ROC	F1 - 0s	F1 - 1s
SVM	19	79.39%	84.24%	0.85	0.65
RF	19	80.43%	83.07%	0.86	0.65
LogReg	19	80.46%	84.63%	0.86	0.67
Dense Layers	19	75.66%	84.24%	0.87	0.67

Table 25: **Unimodal Text results based on RFECV w/ LogReg w/o rescaling**

Model	**Features**	**Accuracy**	**AUC-ROC**	**F1 - 0s**	**F1 - 1s**
SVM	25	79.87%	84.84%	0.85	0.68
RF	25	81.98%	88.27%	0.88	0.66
LogReg	25	79.87%	86.22%	0.85	0.67
Dense Layers	25	80.40%	87.67%	0.86	0.68

Table 26: **Unimodal Text results based on RFECV w/ LogReg w/ rescaling**

In the case of Random Forest as a feature selector, the RF model achieves the best AUC-ROC score, meaning that the RF model learns and generalizes very well with the feature set selected through this selector.

APPENDIX E

Unimodal Audio model - Feature Selectors experiments

Model	**Features**	**Accuracy**	**AUC-ROC**	**F1 - 0s**	**F1 - 1s**
SVM	10	67.57%	65.58%	**0.81**	0.0
SVM	15	**68.09%**	65.61%	0.80	0.23
SVM	20	66.50%	**65.94%**	0.78	**0.28**
RF	10	**64.91%**	52.72%	0.78	0.08
RF	15	64.34%	**54.33%**	0.78	0.08
RF	20	64.86%	50.84%	0.78	0.08
LogReg	10	67.04%	**67.38%**	0.80	0.14
LogReg	15	**69.17%**	65.88%	**0.80**	**0.31**
LogReg	20	67.60%	66.33%	0.79	0.30
Dense Layers	10	**67.03%**	**63.83%**	**0.80**	0.14
Dense Layers	15	67.01%	63.26%	0.80	**0.40**
Dense Layers	20	65.95%	59.16%	0.79	0.33

Table 27: **Unimodal Audio results based on RFE w/ LogReg w/ rescaling**

Model	**Features**	**Accuracy**	**AUC-ROC**	**F1 - 0s**	**F1 - 1s**
SVM	10	67.55%	**69.18%**	0.80	0.06
SVM	15	69.15%	66.34%	**0.81**	**0.17**
SVM	20	**69.66%**	65.99%	**0.81**	0.20
RF	10	**66.43%**	**55.43%**	0.78	**0.24**
RF	15	65.42%	54.97%	0.78	0.13
RF	20	65.95%	50.21%	**0.79**	0.14
LogReg	10	**69.17%**	**69.73%**	**0.81**	**0.26**
LogReg	15	68.09%	68.45%	0.80	0.25
LogReg	20	68.63%	68.95%	0.80	0.23
Dense Layers	10	67.03%	66.65%	0.80	0.31
Dense Layers	15	**71.85%**	**67.26%**	**0.80**	**0.38**
Dense Layers	20	67.03%	67.10%	0.78	0.24

Table 28: **Unimodal Audio results based on RFE w/ LogReg w/o rescaling**

Model	**Features**	**Accuracy**	**AUC-ROC**	**F1 - 0s**	**F1 - 1s**
SVM	37	65.43%	54.55%	0.78	0.22
RF	37	70.73%	69.49%	0.81	0.35
LogReg	37	62.77%	54.66%	0.76	0.20
Dense Layers	37	64.38%	59.28%	0.75	0.35

Table 29: **Unimodal Audio results based on RFECV w/ RF**

APPENDIX F

GridSearch implementation

In the forthcoming table we are presenting the best parameters for the SVM and RF models, as recommended by the applied GridSearch method.

Model	**Parameter 1**	**Parameter 2**	**Parameter 3**
SVM	'max_depth': 10	'min_samples_split': 10	n_estimators': 300
RF	'C': 1	'gamma': 'scale'	'kernel': 'linear'

Table 30: **GridSearch parameters for SVM and RF**

APPENDIX G

Text-speech alignment methodology

Our initial plan for the multimodal model involved a text-speech alignment step, which would also allow us to extract two more significant features; namely speech ratio and pause duration. The chosen aligner would be the *Montreal Forced Aligner (MFA)*, which shows one of the best performances as suggested by our online research. As a forced alignment system, MFA time-aligns a transcript to a corresponding audio file. The particular aligner only requires a set of pretrained acoustic models and a pronunciation dictionary (a.k.a. lexicon) of the words in the transcript with their canonical phonetic pronunciation(s) [41]. At this point we had already identified the pre-trained acoustic model, which was the *English (US) ARPA acoustic model v3.0.0* [42] along with the *English (US) ARPA dictionary v3.0.0* lexicon [43]. The transcript required by the MFA aligner should be in *TextGrid* or *txt* format, while the transcripts provided by the *E-DAIC* dataset were in *csv* format. Hence, the next step was to transform all of our transcipt.csv files into transcript.TextGrid. For that, we installed *praatio*, which has a preinstalled *textgrid* function. However, unfortunately for this plan, the transformation from csv to TextGrid wasn't possible because there were a lot of cases of time overlaps in the transcripts. Running another script for the identification of the overlaps showed us that about 90% of the transcripts included one or two cases of time overlaps between the speaker and the interviewer. If we wanted to move on with text-speech alignment there were only two possible paths to take. Either fix the overlaps one by one manually, which would be extremely time consuming, or delete the corresponding speech segment of the participant, which would be inefficient and could possibly affect the integrity of the two features that we wanted to extract.

Since there are a lot of ways to go for the creation of a multimodal model, we concluded that the most suitable approach in our case (for this particular project and and at this particular time/date) would be to work our way outside the scope of text-speech alignment. Considering that a lot of issues kept on appearing during this process and a lot of manual work and time would need to be devoted to overcome them, finding a way to skip alignment seemed like the most optimal solution. On one hand this meant that we could not properly extract the speech ratio and pause duration features, but on the other hand we knew that we had already extracted a more than extensive features set from each (uni-)modality. In the next subsections we are discussing our final methodology for the completion of the multimodal model.