University of Nottingham
UK | CHINA | MALAYSIA

An investigation of supervised classification of therapeutic process from text

Submitted September 2021, in partial fulfillment of the conditions for the award of the degree of MSc Computer Science

Matthew Laws
20317937

Supervised by Jeremie Clos

School of Computer Science
University of Nottingham

I hereby declare that this dissertation is all my own work, except as indicated in the text:

MATTHEW LAWS
09/09/2021

Abstract

Mental health therapy in the UK suffers from a lack of consistent quality assessment of therapist-patient interactions. This is due to the amount of manual effort required on the part of the therapist to go through and judge a session against a consistent metric. A machine learning program using Natural Language Processing (NLP) to automatically interpret and assess the quality of a session could go a long way to assist in this. To ensure the success of such a program it is necessary to develop effective classification methods, enhanced by constructive data augmentation techniques, to train the program on a small dataset. This paper investigates the effectiveness of seven classification algorithms and five data augmentation methods, alone and in conjunction with one another, in training the model to interpret therapy transcripts. This novel approach to automated analyzing of patient-therapist interactions found that the multi-layer perceptron neural network classifier (MLPC) in combination with random synonym replacement and random word deletion data augmentation techniques of 108% of the sanitized dataset (producing 62,000 sentences) produced the most successful models at a mean F-score of 0.825. This was significantly better (P=0.001) than the next most successful classification algorithm and data augmentation technique pair.

Key Words: Algorithm; Augmentation; Bayes (MNB); F-score; Gradient Boost Classifier (XGB); Neural Network (MLPC); Random Forest (RF); Stochastic Gradient Descent (SGD); TF-IDF;

Acknowledgements

I would like to offer an unlimited amount of thanks to my supervisor, Jeremie Clos, who is and always has been an absolute stand out credit to the University. He has been endlessly helpful, compassionate and an all-round hero.

I would also like to offer thanks to my sister-in-law, Caitlin O'Sullivan, who has gone to great lengths over the course of the year, up to and including this project, to try and teach me how to program and do it in a coherent/professional way. She has spent many hours helping me understand and get through all the challenges I've faced in this extremely difficult year as a computer science post-graduate qualification without a computer science undergraduate qualification and the difficulties of COVID learning.

Finally, I would like to thank my loving fiancée for supporting me through this project. Her endless encouragement and motivation was imperative to the success of this project. I would also like to thank her for helping me with the endless middle of the night alarms for the four weeks of data collection as the project could not have succeeded without this.

Table of Contents

1. Introduction

Concern over the importance of mental health in the UK population has been steadily rising over the past few years. In response to this, coupled with the increasing literature on the importance of mental health treatment, the NHS put forward the 2014 ‘Five Year Forward View’ in which it committed to an equal response to mental and physical health by 2020. This was followed by the ‘Five Year Forward View for Mental Health’ in 2016 to provide more detail and a wider scope to the plan to support the nation’s mental health (House of Commons Library, 2021a). To achieve these goals, additional funding was supplied and a framework of expectations for NHS trusts was created, but despite these improvements, mental health treatment still suffers from long wait times and mental distress in the UK is still on the rise (MHCYP, NHS Digital. 2018; Pierce *et al.*, 2020).

Due to the time and money pressures in the NHS at this time, an aspect of counselling that is not given enough focus is an assessment of the quality of the treatment given. The time cost to the healthcare provider is too high to manually reflect upon a session and to assure the quality of treatment sessions. Thus there is a risk that poor quality treatment goes unnoticed and that improvements go unmade (Rawsthorne *et al*., 2020). Technology could be extremely helpful in this context, automating the process of assessing a session, through flagging moments in a treatment session that indicate progression or regression of the patient, and by presenting ideas for improvement on the part of the therapist.

To implement such a system the program would need to be able to understand written text. Text classification has been increasing in importance in the field of machine learning. Natural Language Processing (NLP) is a field of research that has been created in an attempt to understand ‘natural’ use of language and the nuances therein. It follows then, that these techniques could be used to classify the sentiments and opinions of patients in a therapeutic setting based on the words and phrases used. The ability of NLP to interpret and group disorganised text has allowed it to work with great success in business, health, and marketing. Additionally, it has excelled at interpreting sentiments, opinions, and even the political leanings of social media outputs, campaigns, and news media (Ayo *et al*., 2020; Singh *et al*., 2020).

A machine learning NLP framework could therefore be the ideal candidate for implementing the analysis of these therapeutic texts. The system would train on existing data, and then use that training to interpret novel scenarios and conversations. Machine learning algorithms have, over the past decades, been improved, developed and expanded significantly. The methods by which these algorithms make their classifications of the text, the requirements for the functionality of the algorithms, and the success of these algorithms vary widely depending on the algorithm itself, the implementation of the algorithm and the data the algorithm is working with. Due to these factors, there is rarely a method of machine learning that is clearly the primary candidate for any given project and it is important to explore multiple options before selecting the most appropriate algorithm. In this project we have explored the value of some of the most popular algorithms currently used in the field of text classification: a random forest classifier, a neural network, a gradient boost classifier, a support vector machine with stochastic gradient descent, and three of the most common naive Bayes classifiers: the Bernoulli, multinomial, and Gaussian naive Bayes classifiers. For all classification algorithms the data was split into two thirds used for training the model which was then tested on the remaining third of data through k-fold cross validation.

A major limitation to the power of machine learning are the inaccuracies that can arise when the model is trained on a small dataset. If bias exists in the small dataset, or if the dataset is not varied, or if the dataset is not representative of the true data, the predictions made by the model may be inaccurate when used in real-world scenarios as the model can over or under-predict the likeliness of a scenario based on its chance of occurring in the training dataset. Additionally, small datasets with high complexity can lead to overfitting of the model to the data, so that when the model is presented with a novel scenario it lacks the knowledge to come up with a novel solution. Therefore, to ensure the success of the model on this moderately sized dataset, consisting of 30,000 rows of therapist-patient interactions post-sanitisation, expanding the size of the dataset could go a long way to increasing the value of the model produced.

Data augmentation is the most common way to increase the size of a dataset when the option to collect more data is unavailable or impractical. It is a process of increasing the number of datapoints by manipulating the existing data, essentially creating many subsamples of variations off of one datapoint. This can prevent overfitting by presenting the model with a range of ways in which the same piece of information may appear and ensuring the model is accustomed to the potential variation in real-world scenarios. In this study we have looked at a range of data augmentation techniques on the training dataset to evaluate their success. We have then looked at the most promising of these augmentations in conjunction with the most successful classification algorithms to produce the most effective method for creating a training model for this type of dataset.

The use of machine learning to assist healthcare providers in a medical setting is not a new idea (Mykowiecka, Marciniak and Kupść, 2009; Kluegl et al., 2014; Bing, Chaudhari, Wang and Cohen, 2015), however the process is not used extensively, nor has it been developed to the point of wide usage across many healthcare departments. This paper is an investigation into the best machine learning tools to apply to this dataset to interpret the therapist-patient interactions most accurately. Specifically, we are analytically comparing classification algorithms and data augmentation techniques and then looking at the most successful combination of the two to reliably produce a successful machine learning model. The final classifier needs to display a good representation of the text that aligns with the interpretations you would expect to see made but, also, display good preprocessing of the data despite a small dataset. This would be an entirely novel system for use in a medical setting which, with successful implementation, could be an inexpensive way of seeing significant improvements on the quality of mental health treatment.

1.1 Motivation

The mental health of the citizens of the UK is an important factor to consider when looking at the overall health and productivity of the country. In a review requested by the Prime Minister Theresa May in 2017, Stevenson and Farmer found that there is an annual cost to employers of between £33 billion and £42 billion, largely due to productivity reductions in employees whose mental health is poor in the workplace. The government alone shoulders a burden of between £24 billion and £27 billion annually due to the costs of providing benefits, tax revenue losses, and NHS costs, and the loss to the economy as a whole is estimated at between £74 billion and £99 billion annually. Since this study was undertaken, the levels of mental distress seen in the UK have been steadily increasing, with the rate of increase also increasing as a

result of the COVID-19 pandemic and the public health measures that were implemented to tackle it (Pierce *et al.*, 2020). Prior to the pandemic, more people were seeking mental health treatment (NHS Digital, 2016) requiring increased funding and staffing to handle this while attempting to prevent extensive wait-times (House of Commons Library, 2021a). However, during the pandemic, both the supply and demand for mental health treatment reduced due to limited access to mental health services; the concern is that when the restrictions of the pandemic begin to fall away, mental health services may be subject to more patients and more severe issues due to the delay of treatment (Chen *et al*., 2020).

It is important that not only is the NHS prepared to deal with this influx, but that it deals with it as successfully as possible. The quality of care provided to patients is a topic of great importance and interest to public health as the quality of mental health treatment has not increased at the same rate as treatment for physical conditions (Kilbourne *et al*., 2018). Key barriers identified to be holding back progress in this area are a lack of standardised data sources, cultural barriers to the implementation of these mental health treatments in general health treatment, and limited scientific evidence for measures of mental health treatment quality (Kilbourne *et al*., 2018). This project presents a potential solution to the issue of measures of mental health quality in the creation of a novel system of assessment and reflection on the effectiveness of individual treatment sessions. It also does so without significantly increasing the workload of the therapist as the process would be automatic and so should not add to the stress of the NHS wait times. Following the success of this project, if this research is applied to the grander scheme of the project this falls under, it would bring about the first real advancements in psychoanalytic assessment ever and dawn improvements to the NHS in its efficacy and ability to treat mental health in the UK.

1.2 Aims and Objectives

The overall aim of the project is to develop a system that allows effective quality assessments of patient-psychiatrist interactions through the use of machine learning that also informs the physician of the reasoning behind the assessment made. To contribute to this overall aim, this paper aims to assess key classification algorithms and data augmentation techniques to determine the best combination, i.e. the pairing with the highest F-score, for interpreting therapist-patient treatment transcripts. Broken down, this aim then becomes four research questions (RQ) which we will answer within this report:

> ***RQ1****: Is there a classification algorithm that outperforms the rest without the use of any data augmentation?*
> ***RQ2****: Is there an optimal data augmentation technique and classification algorithm pair that outperforms the rest?*
> ***RQ3****: Is there an optimal percentage of data augmentation for this data augmentation technique and classification algorithm pair?*
> ***RQ4****: Is there a combination of data augmentation techniques that improves the best model found in Research Question 3?*

1.3 Description of the work

The project begins with a review of the existing literature surrounding the subject. Firstly, a description of the state of the country in terms of mental health treatment and its effectiveness, including the current evaluation procedures in place in therapeutic settings. Then, we will move onto the existing literature on the technical side. As the project is a novel exploration of the concept within this setting, there is little existing literature that covers the subject area. Therefore, we will also be looking into similar projects that use NLP and machine learning in a medical setting and similar settings where sentiment analysis using NLP and machine learning has proven effective. This section will conclude with an overview of where the research has gotten so far and where the gaps in the literature are, such that this project should close those holes.

We will then explain the design of the project and the justifications for this design. We will begin with an overview of how the data was prepared and sanitized for use in the project and the importance of this section for the success of the following processes. We will also provide an introduction to the classification algorithms being used, their success in previous projects, and their advantages and disadvantages. This is to be followed by descriptions of the augmentation techniques being used alongside their advantages and disadvantages, and the rationale for choosing these augmentation methods in the context of this study. The design section ends with a description of the evaluation process used for obtaining results and then how those results will be reviewed and analysed.

This is followed up with an explanation of the Methods, beginning with the implementation of this design and the experimental loops. We will then describe the method used to train and test the machine learning technique. We will describe the parameters used for the overall testing, as well as specific parameters used for the different classification algorithms and the rationale behind doing so. This section will end with a description of some additional implementation information, specifically how and why parameters changed throughout the project based on increased understanding and how time constraints affected the parameters used.

We will then select the best design and implementation of this project through statistical analysis. As there are 100 data points for each subset of F-scores produced by the model assessments, the success of these classification algorithms, and data augmentation methods, will be determined through ANOVAs and independent-samples Kruskal-Wallis tests, following tests for the normality of residuals and commonality of variance. This will be followed up by post-hoc tests to determine the details of any significance found and the selection of the most successful classification algorithms, data augmentation techniques, and data augmentation percentages. The section will work through each research question sequentially and will end with a section of additional research where the results of the research questions were used to further build on the most successful classification algorithm and data augmentation technique pairing.

Looking at the results achieved, we will then be comparing them with existing data and existing literature, particularly that mentioned in the ‘Background and Related Work’ section. We will be comparing these F-scores with others’ results to look at the success of the classification algorithms and data augmentation techniques in a wider context. The reasons behind differences between this work and the expected results will also be discussed, as will how our results fit into the wider context of the therapeutic assessment

process. We will then use our results as a basis for recommending further research to improve the F-score of the final model through a potential method to improve pre-processing and a different data augmentation technique that could be used in that case.

Finally, we will be reflecting back on the project as a whole. We will be looking at the final results, the chosen design and implementation, and our assessment of its value and success. Then, we will be looking back at our own approach to the topic and assessing whether we were logical and effective in our methodology, and the ways we would approach it differently now with the value of hindsight. We will end with our recommendations for effective implementation of this research in the therapeutic assessment program.

Post-conclusion, a glossary of all abbreviations is available (Section 8) and following this the bibliography (Section 9).

---

2. Background and Related Work

In this section we will be exploring the state of the UK's current systems of mental health care and the ways that this is currently evaluated for quality. Proposing a machine learning-based solution for this problem, we will then be looking through the existing research surrounding automatic language processing and machine learning in medical contexts.

2.1 Introduction

We will begin with a look into recent statistics regarding the medical approach to mental healthcare in the NHS and the aims and objectives the service has for the future. We will then talk about the current evaluation process for mental healthcare and where we believe there is scope for the creation of a program to help therapeutic treatment providers that will create the basis of this thesis.

2.1.1 Background statistics

The undeniable rise in mental health disorders has increased the need for psychotherapeutic resources in England and the UK. For context, the NHS conducted a series of mental health surveys on children and adolescents in 1999, 2004 and 2017, and a followup to this survey in 2020 (MHCYP, 2018; 2020). The 2017 report found that one in eight (12.8%) 5 to 19 year olds suffer from at least one mental health disorder. These disorders were grouped into four categories: behavioural, emotional, hyperactivity and those that are less common. Of those four categories, emotional disorders were most prevalent within this age range. It was also found that mental health disorders were more common as age advanced with 5.5% of 2 to 4 year olds and 16.9% of those aged 17-19 falling into these categories. Importantly, the survey identifies an increase of mental health disorders identified at this age range progressively in each of the surveys, 9.7%, 10.1% and 11.2% respectively. In the follow-up report in 2020, the NHS reported there has been a rise of 3.2% in mental health disorders in the same age range in just 3 years. To apply some further final context to the rise in mental health disorders and the need for improved care, only 62.6% of children with mental health disorders have regular support outside of the home. Similar statistics have also been seen in adults (House of Commons Library, 2021) and, as the effects of the COVID-19 pandemic are beginning to be seen (O'Connor et al., 2020; Jacob et al., 2021), there is a great need to investigate the current mental health treatments in the UK and to look into improving upon them.

2.1.2 Current evaluation process

The process of documenting therapy sessions is currently conducted manually by care providers. It is transcribed and labelled with a therapeutic process by hand and takes a substantial amount of time, both to input and to qualitatively corroborate (Tseng, Baucom and Georgiou, 2017). In an attempt to provide a new process in which therapeutic evaluations, and their evolution over time, can be assessed, an investigation of how supervised classification can be utilised, and their efficacy, is salient. The automated tool will be essential in allowing a wider adoption of reflective practices in mental health care which would likely lead to an improvement in therapy on a large scale.

2.2 Literature

In this section we will be looking at the existing literature surrounding machine learning in understanding the written language, followed by the use of machine learning in a therapeutic setting, finishing with an assessment of where current research has gotten to and how this project would be building upon that base.

2.2.1 Natural Language Processing and Supervised Machine Learning

Natural language processing (NLP) is a subfield of artificial intelligence, computer science and information engineering that occupies the field of interactions between computer systems and natural human languages as an application of supervised machine learning. Specifically, the process in which a system takes large quantities of this natural language data to process and analyze it (Liddy, 2001; Chowdhury, 2003). Machine learning, in the context of supervised learning, can be explained as both the classification algorithms and statistical applications that are carried out by computers using training data without the need for user input. This training data is created by the system building a model using sample data, that it will use to make projections and decisions, without being specifically asked to, utilizing patterns and inference (Shalev-Shwartz and Ben-David, 2014; Razno, 2019).

Text classification has been seen to be efficient and effective as one of supervised machine learning's primary applications (Razno, 2019). The input, text-based documents such as a web pages, books or clinical evaluations, will be assigned classes to their data objects using pre-defined class labels based on their relationships (Hussain et al., 2019), to the outputs for the use of a range of applications such as trust networks in social media (Zolfaghar and Aghaie, 2011), large data analytics (Lux, Pittman, Shende and Shende, 2016) and sentiment analysis (Melville, Gryc and Lawrence, 2009; Rawsthorne et al., 2020).

2.2.2. Supervised Machine Learning in a Medical and Psychotherapeutic Setting

Machine learning has been used in the medical field on multiple occasions for the role of clinical text classification tasks (Mykowiecka, Marciniak and Kupść, 2009; Kluegl et al., 2014; Bing, Chaudhari, Wang and Cohen, 2015). Despite the success seen herein, some problems are still seen. For example, successful machine learning software requires large amounts of human input to train the model on the dataset and its nuances. In a clinical setting, the development of such a system can also struggle due to a lack of relevant data to train the system on due to confidentiality, and the barrier to understanding medical documentation for the system trainer (Wang et al., 2019).

Shining a light on the application of supervised learning within a psychotherapeutic setting, a position paper (Rawsthorne et al., 2020) went a long way to exploring the gaps in the systems available for psychotherapists, describing the lack of quality assessments of patient-therapist interaction due to the manual effort needed by the therapists and additionally, its necessity. It further points out that providing explanations to those acting on the outputs of the paradigm, both the reasoning behind the decisions made and the paradigm itself, is good practice in data driven health care and technology. They attain their results by independently surveying the team for identifiable key language markers, which were then coded into Python scripts. These scripts were then run through 120 health anxiety sessions through which they were then able to build a network of interactions of interest as a predictor of clinical outcomes. Their experiment into text classification of psychotherapy transcripts, which only used the single algorithm

Chi-square Automatic Interaction Detector (CHAID), registered success but a relatively low accuracy reporting a maximum of 0.74 accuracy with a low of 0.60. The position paper constructed a strong basis for improving on their paradigm and identified what further work would be necessary to do so.

One way to improve upon this work is through the use of Neural Networks. Wang et al. (2019) carried out an experiment on two institutional case studies and one public case study. They used the machine learning algorithms Support Vector Machine (SVM), Random Forest (RF), Multilayer Perceptron Neural Networks (MLPC) and Convolutional Neural Networks (CNN). They found that CNN outperformed all the other machine learning models, with accuracies of 0.92 and 0.97. In addition, an area that Wang et al.'s study in 2019 found of import and expanded on previously by Wang et al. (2016), was the use of pre-trained word embeddings, used in conjunction with their rule-based NLP algorithm for generating labels which were found to significantly outperform all the other modelling features. A limitation however is that CNN is sensitive to data size. This report found that until the training data size reached 5,000 for the tasks the deep learning method underperformed against the three other techniques.

One way to combat this limitation if there is insufficient data is the use of data augmentation (Wei and Zou, 2019). Methods such as back translation and synonym replacement allow for the near-replication of data without affecting the underlying class assigned to the data objects through the text classification. This has been seen to improve the results and accuracy of CNNs; and other data sensitive deep learning models. The rule-based machine learning methods, those which are more resistant to data size, prove to be more effective at smaller data sets but show little to no improvement as data size increases whereas the improvement curve for CNN is significant and its efficacy becomes dominant (Wang et al., 2019).

Further limitations identified with CNNs being used for text based classification, as a form of supervised text classification, were also outlined in this report as multiclass classification and the interpretability of the results. It was supposed that the reason behind the less than optimal results in the third case study for CNNs, one requiring multiclass classification, was caused by a small training data size. Therefore it would be necessary to use and test this paradigm with an improved training data set. The final limitation, interpretability, was identified in the report's error analysis. They went on to state that the rule-based paradigms, including the NLP MLPC paradigm, were much easier to interpret and that rules can be added or modified much more readily. It is also assessed that a need for extensive human input to utilise and an expansive knowledge and experience to interpret the results is necessary. Therefore, a rule-based deep learning approach may prove to be superior to a CNN in an automatic tool for evaluating patient-therapist interactions. This is validated by accuracies of 0.92, on par with a CNN in the first test, and 0.93 in the second; 0.3 points below the performance of the CNN. The positive ramification of using the MLPC approach is its data resistance, where accuracy was observed from a data size of just 1,000 and remained consistent as the data size increased.

2.3 Conclusion

> *"I believe that some aspects of psychoanalytic theory are not presently researchable because the intermediate technology required – which really means instruments-cum-theory – does not exist. I mean auxiliaries and methods such as a souped-up, highly developed science of psycholinguistics, and the kind of mathematics that is needed to conduct a rigorous but clinically sensitive and psychoanalytically realistic job of theme tracing in the analytic protocol"* (Meehl, 1978).

It was 43 years ago that Meehl posited that the lack of progress into psychoanalytic theory is hindered by the lack of intermediate technology and this still stands true. There have been no fundamental changes to the principle science behind patient-therapist interactions and in fact the field hasn't seen any real advancements in 70 years (Imel, Steyvers and Atkins, 2015). The research conducted by Imel, Steyvers and Atkins went a long way to providing evidence that a computational approach to advancing the field is entirely possible.

One path to improve the assessment of patient-therapist interactions and the evolution of a patient's psychotherapy in the mental health field is creating a system that can interpret counselling transcripts, give information on the patient's response, and allow the therapists to make more informed decisions on the progression of treatment. To that end, a system that uses NLP algorithms through machine learning would be a promising path as a basic version of this process (Rawsthorne et al., 2020) and has found accuracies of up to 0.74 in its labelling. Refinement of the classification tool, and updating of the classification algorithm would need to occur. Additionally, as therapy to induce behaviour change often involves persuasion by the therapist, inclusion of aspects of the Argumentation Theory and empathy would be significant.

Success in this project would result in a system that would allow therapists a broader view of the value of the treatment they were providing, alongside assisting in making decisions on where to target therapy in the future using an automated tool developed using neural networks, pretrained data and data augmentation.

---

3. Design and Justification

This section will detail the pre-processing of the text required to clean and prepare it for machine learning algorithms to train on. We will then explain the purpose, advantages, and disadvantages of the different algorithms to be used throughout the project. This is followed by a brief explanation of each augmentation technique and how these are carried out. The section ends with a description of the processes of assessment of the classification algorithms and data augmentation methods followed by the analytical techniques to be used to assess statistically-significant differences between these methods.

3.1 Data and Sanitization

The data for this project was provided by the School of Medicine at the University of Nottingham. It was professionally anonymized within the turns, the individual speak and respond components of a conversation, to replace the name of the speaker to generally “T” for therapist and “P” for patient.

In order to sanitize the data for use in the project, firstly the Pandas (Reback *et al.,* 2020) library for Python was used for the purposes of handling the data as a dataframe and additionally for concatenating variables to series and arrays as necessary. Firstly, unnecessary columns were removed from the CSV and then the remaining columns were labelled more appropriately - for the purpose of this experiment the following columns were kept: label, voice, score and sentence; these columns were also renamed to the aforementioned titles for ease of comprehension when designing the code. Score is redundant in this experiment due to the time constraints but it remained as a stretch goal to see if the score could also be predicted for.

Next, a short python script was used to drop the various rows within voice labelled as NA and any preexisting or subsequent NA entries in sentence. Following on from there, all remaining rows were made uniform containing T or P within the voice column denoting whether it is the therapist (T) or the patient (P) speaking, and the turn (the order of the conversation) removed. Without this step, the turn and the voice were labelled together, e.g. T108. The sentence column required two types of sanitization. Firstly, some of the turn_ids, or voice entries, had spilled their formatting over into the sentence column, for example some speech entries began with T108 or P64, and these needed to be removed from the sentence column and, in some cases, added back to the voice column from which they were missing. Secondly, due to a formatting entry error, most, if not all, sentences began with tab spacing coded as \t. In order to remedy both of these issues, a split script was used which would either split the sentence at the colon of the T108: or the t in \t - creating two strings in a list for that sentence. Subsequently, the first item would then be removed from the list leaving us with just the sentence. A condition was set within this script to ensure this wouldn’t break a sentence where a colon cropped up later in the sentence as a part of the natural grammar. Finally, during the text vectorization method (see Section 4.1) Natural Language Toolkit (NLTK) is used to remove all “stop-words”. Stop words are defined generally as those that carry little-to-no weight for the algorithm, for example: “the”, “a”, “is” or “are”, and are therefore removed from the sentences. By removing this low-level information, we allow the algorithms to focus more directly on the high-level information carried by the rest of the sentence.

3.2 Classification Algorithms

The classification algorithms for this project were implemented using SciKit Learn (sklearn) (Pedregosa *et al.,* 2011). Unless specifically stated otherwise, any algorithm described will be using the sklearn implementation method. Additionally, in order to gain a baseline score for evaluating the results of the classification algorithms, a *DummyClassifier* paradigm, obtained from the sklearn package, was used (Pedregosa *et al.,* 2011). Using the "most_frequent" parameter for this method, it gives us an f-score of 0.597; before any data augmentation. Therefore, if when implementing the algorithms to be assessed in this project, the resulting f-score is consistently equal to, or less than, the dummy classifier at 0.597, then we know that the model is predicting the most frequent label and therefore is not performing. Thus, models providing f-scores above the dummy classifier are deemed as performing and are kept in the project to be considered to be improved and used with data augmentation techniques. Similarly, models below this dummy classifier's result are considered underperforming and are removed from consideration to answer the research questions.

3.2.1 Random Forest (RF)

Random forests (random decision forests) are an 'ensemble method' where multiple decision trees (a process of making a classification through a series of forked decision points) are built in conjunction with one another to improve the performance and accuracy of the classification (Kam Ho, 1995). Each tree is constructed from different, random sections of the data totalling up to the percentage of total data chosen to be used for that process. The tree repeatedly 'forks' through the data in the subset, dividing the data into the groups it determines are most similar, a factor determined in this case using 'Gini importance' (Pedregosa *et al.,* 2011), until it reaches a 'leaf': the final group after repeated splitting, i.e., the classification (Kensert *et al*., 2018). Essentially, the final classification for the whole forest is decided by whichever classification is made by the most trees. The whole system acts on the 'law of large numbers', that building many trees results in a more appropriate generalisation of the data than a single tree could achieve (Kensert *et al*., 2018).

A benefit of a single decision tree is that the decision process can be easily interpreted by the researcher, which is rare in machine learning. The researcher can see every decision made by the tree and why it was made. When the RF is introduced that interpretability goes away (Densiko & Hoffman, 2018). For this project, the ability for the therapist to see why a decision was made in the assessment is something we consider important to the final product. Another disadvantage of the system is that the algorithm is computationally expensive. Trialling and developing aspects of the algorithm take a long time and so the developmental time for a system using this algorithm will be long and may limit the accuracy of the final system if the development is limited by a deadline.

The algorithm can also be highly beneficial. Due to the aforementioned 'law of large numbers', this algorithm can reduce the chances of 'overfitting' to the training data, a common flaw in machine learning where the model fits the training data so exactly that it cannot perform well on unseen data, defeating the purpose of the model (Breiman, 2001; Densiko & Hoffman, 2018). The ensemble algorithm has also been seen to have high predictive accuracy despite the often poor performance of a sole decision tree (Densiko & Hoffman, 2018). Due to this, they are commonly used in many fields, including the medical field where accuracy is paramount for roles such as predicting the response of a cancer cell to a new drug, identifying

proteins, and locating cancerous tissues (Densiko & Hoffman, 2018). This, combined with the use of the algorithm to great effect in speech and writing analyses makes the algorithm an ideal candidate for inclusion in the current system we are creating (Amato *et al*., 2021; Densiko & Hoffman, 2018).

3.2.2 Neural Network (Multi-Layer Perceptron, MLPC)

Neural networks are methods of machine learning that are analogous to the neurons of a brain. The nodes are the 'neurons' and the connections between these nodes are the 'synapses'. Each neuron takes in information, applies a function to this data, and then passes that output through a synapse to the next neuron. Weights are also applied to the outputs at each stage and it is these weightings that make up the training phase of the neural network (Denœux, 2000).

In this system the neural network used is a multi-layer perceptron classifier (MLPC), a class of neural networks where information is fed strictly from one node to the next and does not return to a previous node at any point, i.e. feedforward. The term 'multi-layer' in MLPC refers to the existence of one or more hidden layer and all MLPC's consist of at least three layers:

a) the input layer, which only operates to move the data to the first (or only) hidden layer.
b) a hidden layer, where weights are transferred to another hidden layer or the output layer.
c) the output layer, where a nonlinear activation function is used that determines the final output value dependent upon the input.

An MLPC is defined in part by its use of nonlinear activation functions. Nonlinear functions allow networks to give more complex results than the binary classifications of linear activation functions. These nonlinear functions can compute more complex classifications using a smaller number of nodes that should reduce the computational cost. The neural network 'learns' to adjust the weights and thresholds used within the system to give improved outputs (Fuomo, 2017).

An issue with the method is the high processing power needed to run the systems due to the complex nature of the algorithm itself (Mijwel, 2018). A neural network can also require a larger training dataset than other machine learning algorithms and can also, as with RF, suffer from the obscurity of its decisions (Li *et al*., 2016; Mijwel, 2018). The neural net will make classifications with no clear basis for the decisions made which, while the decisions may be highly accurate, would be unhelpful for a use case such as this where the reasoning for the classifications is a feature desired by the therapists using the system. However there is also much research currently being done into ways to automatically interpret the decisions made by the neural net, and so with correct implementation of such a system this issue may be avoided (Belinkov, Gehrmann & Pavlick, 2020; Zhang, Wu & Zhu, 2018).

The key benefit of a neural network for machine learning classification problems is that they are a powerful classifier for complex data. As a network can be many hidden layers deep, the network can classify a piece of data into a wide range of final classifications based on a complex array of factors, and the non-linearity of the activation function is often associated with high levels of success (Goldberg, 2017).

3.2.3 Gradient Boost Classifier (XGB)

A gradient boost classifier is another example, like the RF method, of an ensemble method, combining multiple weaker decision trees into a more reliable model for classification. Unlike the RF method where many trees are built using random subsets of the data, gradient boosting classifiers involve building trees sequentially where each tree attempts to correct for errors found in the previous. This can lead to more accurate classifications that can be particularly effective on 'unbalanced' datasets (Pedregosa *et al.,* 2011). A 2021 study by Sachdeva and Kumar found that when comparing these two ensemble methods the gradient boost classifier outperformed the RF by 8% (79% accuracy vs. 71%) and another found that the gradient boost classifier was the most accurate of the main tree-based classifiers (Ghanem, Rosso & Rangel, 2018).

As with the RF classifier, the gradient boost classifier sacrifices interpretability for higher success rates and, additionally, the model is associated with a high computational demand. The time cost can also be higher for the gradient boost as the trees must be made sequentially and multiple cannot be made at once. They can also suffer more significantly from overfitting as the process involves 'boosting' errors in the dataset for easier detection, but can then overfit smaller differences (Pedregosa *et al.,* 2011).

Positively, XGB classifiers can be extremely powerful classification algorithms. The same 'boosting' that can lead to overfitting also allows the algorithm to detect small differences in complex datasets (Sachdeva & Kumar, 2021). The sequential nature of the building of the decision trees can also reduce the effect of bias in a training dataset as variance is removed in the improvement of one tree on another and also in the aggregation of multiple trees together (Sachdeva & Kumar, 2021).

3.2.4 Support Vector Machines with Stochastic Gradient Descent (SVM, SGD)

Support vector machines are machine learning algorithms that classify subjects by finding a 'separating line' that distinguishes groups from one another. A stochastic gradient descent is a method of randomly selecting a possible solution, looking at the success of that solution, and then iteratively moving towards the best solution. Thus, the combination of these two concepts is the process of creating arbitrary separations of data, looking at the success of these delineations, and then modifying them until the most useful version is created.

The algorithm suffers from oversensitivity to feature scaling, i.e. the system would struggle to differentiate between two factors with very different scales, but feature scaling, or normalising the data, would allow the system to more easily compare the factors. The issue is that some methods of normalisation can cause the SVM to misinterpret the final classification.

An advantage of the method is the ease with which the process can be modified for its specific purpose with the specifics of the stochastic gradient descent relatively simple to both understand and modify. SVMs alone, like the other algorithms discussed thus far, also suffer from being computationally expensive; however, using stochastic gradient descent can reduce that cost as it is a more simplified method of assessing the success of a classification (Lopes *et al*., 2019). The algorithm is also a fairly efficient process that is much less costly computationally than other, equivalent, algorithms and the time

taken for the algorithm to be run through does not tend to increase as the training set gets larger (Pedregosa *et al.,* 2011).

3.2.5 Bayes Classifiers

Bayes classifiers are commonly used probabilistic classifiers in the field of machine learning for minimising the chances for misclassification. The model is based on Bayes' Theorem, a method of calculating a conditional probability based on the probability of another known variable. The classifier can be computationally expensive as it assumes a relationship that can be calculated between all variables. The classifier also relies on the dataset on which the system is trained being representative of all proportional outcomes in the true data. This is rarely feasible and so 'Naive Bayes' classifiers are often used instead.

Naive Bayes classifiers (simple Bayes, independence Bayes) are essentially a simplified form of the Bayes classifier. The Naive Bayes classifiers don't assume the dependency of a variable on all others and treat them as independent. These are more effective on smaller datasets than the Bayes classifiers as the proportionalities of the sample data does not need to be as directly representative of the true data. Despite their 'simplicity' Naive Bayes are still considered highly effective and efficient machine learning algorithms even when the assumption of independence of the variables is not met (Ranganathan *et al.*, 2019). The three most popular, particularly for document classification, of the Naive Bayes classifiers are the Bernoulli Naive Bayes classifier, the Multinomial Naive Bayes classifier, and the Gaussian Naive Bayes classifier (Singh *et al*., 2019).

3.2.5.1 Bernoulli Naive Bayes (BNB)

Bernoulli Naive Bayes classifiers look at whether or not a keyword appears in a document and modify the text classification based on this factor (Singh *et al.*, 2019). This additional information can be particularly important when some keywords can change the meaning or tone of the rest of the document. In this case of therapy sessions keywords and phrases related to wellbeing and attitudes can greatly affect the way the session is interpreted.

3.2.5.2 Multinomial Naive Bayes (MNB)

Like BNB, the Multinomial Naive Bayes classifier looks at whether keywords appear in the text. This method, however, also takes into account how often those words appear in the text, i.e. the term frequency (Singh *et al.*, 2019). In the case of therapy sessions the occurrence rate of a word can have a sizable impact on the way the session is interpreted, for example many repetitions of a positive word, or series of positive words, may be more important to the interpretation of the text than one occurrence of a negative term. It is important in this analysis that stopwords (these can be any words chosen by the analyst but are usually words that would not affect the classification of the text such as 'the', 'also', or 'was') are not weighted as they can appear many times in a text but add no weight to the interpretation; this would apply to any words not automatically excluded by NLTK. One study by Singh *et al.* (2019) on a comparison of the Bernoulli and Multinomial Naive Bayes classifiers on news article interpretations found that the Multinomial classifier at a 73% success rate was slightly better than the Bernoulli at 69%. However, the success rate of the Multinomial Naive Bayes classifier at 73% was classified by the researchers as 'not very effective'. They estimated that this was due to a small dataset in the training.

3.2.5.3 Gaussian Naive Bayes (GNB)

It can often be assumed that a continuous variable, when the dataset is large enough, will fall in a Gaussian, or 'Normal', distribution. In a comparison of the Bernoulli and Gaussian Naive Bayes classifiers in detecting fake news Singh *et al.* (2020) found that the Gaussian classifier gave an accuracy of approximately 72%, but this was outperformed by a success rate of 83% when using the Bernoulli Naive Bayes classifier. Whilst we assumed a multinomial distribution of the data, this method was used early in the design phases for testing purposes. Over 100 tests, the f-score average was < 0.2 and therefore underperforming as compared to the dummy classifier and removed from consideration for the project.

3.3 Augmentations

The success and accuracy in NLP through machine learning is highly dependent on the size and variation of the training dataset. Thus, using automatic data augmentation for increasing the size of a test dataset can provide significant benefits to the accuracy of the final program. The data will be augmented to the following breakpoints: 0%, 30%, 60% and 90% extra data added and, in the case of the cross augmentation technique, 108% total data is added; 60% initially plus 30% of the new total augmented data.

The amount that each augmentation method would augment was determined by not specifying the aug_max; the maximum number of words in a sentence to augment. NLPAUG uses an in-built method called aup_p to determine a standardized amount to augment depending on the length of the sentence, helping to ensure underlying classes aren't affected by the augmentation. The exception to this is RSR and can be seen in figure 1. We select a minimum, and a maximum however, this can be volatile in one-word sentences or completely redundant in those which are much longer. A standardized percentage of total words in a sentence to be augmented, like aup_p, would be much more appropriate. We kept the RSR technique in a potentially weaker state (Section 3.3.1; Wei and Zou, 2019) to test how much difference it had against BERT's RWS deliberately. This decision was made because the methods work in similar ways and we wanted to increase the robustness of the results this way, rather than potentially replicating RWS and RSR with one another. The differences between the results using RSR and RWR were not significant but if they had been, and if there was more time, this parameter would have been experimented upon or, at the very least, standardized.

The augmentations were initialized (figure 1) in a way not dissimilar from a common Python factory pattern. The lack of inherent lazy loading in python meant it could not be implemented as cleanly as the algorithm factory pattern and because of this only one single augmentation, or multiple at the same time if using cross-augmentation, can be run per experimental loop; slowing down the experimental process.

```
class AugmentFactory():
  def __init__(self):
      self._synRep = None
      self._randLet = None
      self._nonRandLet = None
      self._randWordInsert = None
      self._randWordSub = None
      self._randWordDel = None
```

Figure 1: Factory implementation of augmentation technique initializers

The augmentation type and percentage to be augmented are thereafter selected for in the main experimental loop, see figure 2. When we want to use cross augmentation, we simply uncomment the excepted line and manipulate it as we see fit.

```
aug = AugmentFactory().RANDOM_WORD_DELETION()
newDF = DataAugmentor().augment(data, 60, aug) #int = the amount of data to be augmented
#newDF = DataAugmentor().augment(newDF, 30,
AugmentFactory().RANDOM_WORD_SUBSTITUTION())
```

Figure 2: Implementation of augmentation and its manipulation in the experimental loop.

3.3.1 Random Synonym Replacement and Substitution (RSR and RWS)

Synonym replacement for data augmentation involves randomly choosing a selection of words that are not stop words, which are then replaced with a synonym of that word; also chosen at random. The process can be computationally expensive as the system is required to search through a large database of words for the right synonym (Rizos & Schuller, 2019). The success of this method can be varied with one study (Kolomiyets, Bethard & Moens, 2011) seeing error reductions as a result of synonym replacement of 10.6% and 23.3%, and final F-scores of 0.796 and 0.877, depending on the training data source. As sentence lengths vary in the transcripts, using a flat number of substitutions per sentence may not be the most optimal method for augmentation. Wei and Zou (2019) have found that varying the number of these augmentation events according to the sentence length can improve the success of the system. Importantly, the 2011 study (Kolomiyets, Bethard & Moens) found that RSR alone did not improve the value of the dataset and that it was only in combination with the Latent Words Language Model (LWLM), that predicts the most valuable synonym according to context, that an improvement was seen. The system being built here used two methods of word selection that weighted the replacement words towards those most likely to fit the sentence.

3.3.1.1 Paraphrase Database (PPDB)

For random synonym replacement (RSR), the system selects the most appropriate word for synonym replacement from the PPDB database of multilingual paraphrases based on the highest score for matching with the original word (Pavlick *et al*., 2015). PDBB RSR was defined in figure 3.

```
def SYNONYM_REPLACEMENT(self):
  if (self._synRep == None):
     self._synRep = naw.SynonymAug(aug_src='ppdb',
model_path=os.environ.get("MODEL_DIR") + 'ppdb-2.0-tldr', aug_min=1, aug_max=3)
  return self._synRep
```

Figure 3: Definition of the RSR method for use in augmentation.

3.3.1.2 Bidirectional Encoder Representations from Transformer (BERT)

For random word substitution (RWS), a synonym of a random non-stop-word is substituted in place of the original word. This synonym is chosen using BERT to search for the most suitable word for augmentation (Devlin *et al*., 2019). BERT's RWS was defined in figure 4.

```
def RANDOM_WORD_SUBSTITUTION(self):
  if (self._randWordSub == None):
     self._randWordSub = naw.ContextualWordEmbsAug(model_path='bert-base-uncased',
action="substitute")
  return self._randWordSub
```

Figure 4: Definition of the RWS method for use in augmentation.

3.3.2 Random Letter Replacement (RLR)

This augmenter applies random character errors to the text. The generation of these errors is through random value replacements of letters. This is used as it is common for a person to mis-type and thus spell words incorrectly and so prepares the model to be used on a dataset that may contain such errors. The disadvantage of this method is that it can often create unrelated or nonsense words that, while expanding the size of the dataset, do not increase the number of valuable sentences on which the algorithm can train. NLPAUG's RLR was defined in figure 5.

```
def RANDOM_LETTER_REPLACEMENT(self):
  if (self._randLet == None):
     self._randLet = nac.RandomCharAug(action="substitute")
  return self._randLet
```

Figure 5: Definition of the RLR method for use in augmentation.

3.3.3 Random Word Insertion (RWI)

Random word insertion involves finding a synonym at random for a word in the sentence and then adding that word in at a random point in the sentence, rather than replacing the original word with the synonym as with RSR and RWS. This synonym, like RWS, is found through using the BERT model for language modelling to select an appropriate synonym given the context. This can be done a specified number of times per sentence. BERT's RWI was defined in figure 6.

```
def RANDOM_WORD_INSERTION(self):
  if (self._randWordInsert == None):
     self._randWordInsert = naw.ContextualWordEmbsAug(model_path='bert-base-uncased',
action="insert")
  return self._randWordInsert
```

Figure 6: Definition of the RWI method for use in augmentation.

3.3.4 Random Word Deletion (RWD)

This involves the random deletion of a word from the sentence which can simulate words that may be missed in real-world transcriptions as well as mis-speaking on the part of the patient or therapist. This has the benefit of increasing the dataset while not increasing the chance of training the model on unrelated words and phrases created by other augmentation methods. However, this is also a limitation of the method as it does not create a wide range of additional words on which the model can train. NLPAUG's RWD was defined in the figure 7:

```
def RANDOM_WORD_DELETION(self):
  if (self._randWordDel == None):
     self._randWordDel = naw.RandomWordAug()
  return self._randWordDel
```

Figure 7: Definition of the RWD method for use in augmentation.

3.4 Evaluation

The machine learning algorithms will be trained and assessed using K fold cross validation, i.e. the process of estimating the skill of the machine learning model by training the model on approximately ⅔ of the dataset and testing it on the remaining ⅓. This will give information not only on the value of the design and implementation that is best for this specific project, but also on the value of the designs and implementations on smaller datasets; their value using text classification from spoken transcripts in general; and how size affects the way the algorithms run. This will be especially useful in the future for projects that may only have access to smaller datasets and to see if increased dataset size, especially those increased artificially with data augmentation, actually affects the value of the algorithm.

After each fold is completed using the aforementioned method, an F-score will be returned to us which will help us determine the effectiveness of the model. The F-score is a measure of the model's accuracy as

a function of the harmonic mean of precision and recall, with optimal values of 1 and least optimal at 0, and provides us with more useful information than just the accuracy of predicting the right class for a sentence alone (Baeza-Yates and Ribeiro-Neto, 2011). We calculate the F-score using the following equation:

$$F1 = \frac{2}{recall^{-1} + precision^{-1}} = 2.\frac{precision.recall}{precision + recall} = \frac{tp}{tp + \frac{1}{2}(rp + fn)}$$

For each section of the data there are 100 folds, i.e. 100 data points. We are attempting to discern overall differences between factors with large ranges in results (e.g. the F-score for RF at 0% augmentation ranges from 0.387 to 0.633) and whose ranges widely overlap. For this situation an analysis of variance (ANOVA) and its variations are the most appropriate tests. Prior to this, we must test that our data meets the assumptions of an ANOVA:

1. That the residuals are normally distributed
    a. We have done this visually through the creation of a histogram of residuals which should be approximately bell-shaped. Formal tests for normality at sample sizes above 30-40 become less important as the importance of normality is reduced at large sample sizes (Pallant, 2007).
2. That there is homogeneity of variances
    a. This is tested for using a Levene's test, widely regarded as the more appropriate test for homogeneity than the Bartlett's test which is considered oversensitive to departures from homogeneity that don't affect the final calculations.

Where these assumptions are met the calculations proceed, analysing the data through one-way or two-way ANOVAs. Where there are more than two factor levels, these initial tests are then followed by Tukey's post-hoc tests to determine which factor pairs are significantly different ($P<0.05$). Where the assumptions for an ANOVA are not met, the ANOVA will be replaced with a nonparametric Kruskall-Wallis test, and the post-hoc will be carried out through a Dunn's pairwise test with a Bonferroni correction.

---

4. Method

This section will describe the implementations of the algorithms and augmentations used: the basics of the process and the external resources used.

4.1 Implementation and Experimental Loop

Following the data sanitization (Section 3.1) we perform a method for splitting the data into a training set and a testing set, proportionally ⅔ to ⅓ respectively (Section 3.4.1) with K-Fold Cross Validation. X and Y variables were made for testing and training data sets, where X is the sentences and Y is the labels; e.g. IDI. Next, the models are trained. It takes the training sentences (x_train) and the training labels (y_train) and uses them on the model we've chosen - the train method returns the trained model.

Once the data is split and augmented, term frequency–inverse document frequency (TF-IDF) is used as a scoring measure of the words within the sentence - with the intention to classify how relevant it is within the sentence itself. The relevance of a word can be defined in this context as the proportional amount of information the word supplies about its context within the sentence. This is achieved by the frequency a word is used: the more frequent the more relevant it is to this particular sentence. Naturally, because a word occurs frequently does not make it integral or useful and is one of the reasons, at this stage, stop-words are removed. The TF part of this method specifically tallies the frequency of relevant words to the sentences across the document and provides us with an initial estimate. To refine further what words carry the most weight or relevance, the second half of the TF-IDF method is deployed. IDF will take the TF approximation of relevance and then penalise or eliminate words that are frequent across the whole document, classifying them as generic words or noise. The max features, or the maximum number of words in the pool returned, was set to 500 for the purpose of this experiment. This is not the optimum by any means for any of our algorithms; however, because of the time constraints of this project and the extensive amount of data being collected over the experimental loop(s), it was one of the most necessary factors to limit. Further research (Section 5.5.1) explores the results of lifting this max feature cap to 2000. Also of note is that this change of max features within TF-IDF was only feasible as added research once the bulk of the data was collected as RF, with this new limit, took 12 hours to run one experimental loop at one augmentation increment and type meaning a total of 7.5 full days would be needed to calculate RF alone.

This project employed the use of factory patterns to the dependent variable classes, such as algorithm and data augmentation, to allow them to run one after the other without the need for user input. These factory API-like code implementations allowed easy enabling or disabling of particular methods within the experimental loop and was especially useful for some of the longer-to-run methods, such as MLPC and RF, as it allowed them to run overnight. The K-Fold Cross Validation method was also implemented as a class to allow it to be called simply within the experimentation loop and defined by a hard-coded integer variable, “n_splits = *int*”, for how many repetitions, or folds, of each algorithm under each augmentation technique would be run. Sadly, because Python does not incorporate “lazy loading”, and the sheer size of BERT and PPDB augmentation libraries, the augmentation factory was not implemented to full effect and only one augmentation could be used at a time per user-instigated experimental loop.

At the end of each fold within an experimental loop, our Evaluate method is called. This method takes the trained model and the test data (x_test and y_test) and then our evaluate method provides us with a printout of that model's F-score, which is then used for analysis (Section 5).

4.2 Additional Implementation Information

During the early stages of experimentation, the default parameters for the algorithms were used. Once the experimental loop design was complete, the parameters for the algorithms were then hypertuned. The algorithm parameters were hypertuned using the SKLearn “GridSearchCV” (grid search) method. In combination with K-Fold Cross Validation, we were able to specify what parameters we wanted to search for and at what increments, or values, within that parameter we wanted to test. We are also able to select for multiple parameters at once allowing the grid search to automatically test these combinations. Finally, grid search also implements a score and fit method providing us with an estimator for the best combination of parameters and their individual best value within. Naturally, this exhaustive search takes a considerable amount of time and with the limited time available within this experiment, only the parameters identified as the most important, through literature, were entered into the grid search. For the purposes of this project and its given timeframe, grid-searching for hyperparameter tuning was done using the base dataset. Ideally this tuning would have taken place at each augmentation percentage per each technique for all the algorithms as certain algorithms and their parameters are sensitive to the amount of data.

Early stopping was a vital feature for saving time during this project by preventing overfitting. Overfitting for a model is defined as the point at which the algorithm continues to attempt to improve on the training set but is producing worse results on the testing set. The optimum relationship between training and test errors for effective learning are where they are both seen to be low. When this relationship sees low training errors and high test errors, or variance, then we know our model is overfitting. Early stopping is a feature that allows the model to finish its training and the optimum point, provide its F-score, and move on to the next fold. For models such as XGB and MLPC, this, in some cases, halved exhaustive experimental loop times.

Due to the sensitivity of the parameters being hypertuned for the data itself, no hard coded examples are provided. Instead, a set of recommendations are laid out for the user to apply this research further at the end of the conclusion (Section 7.3). It is the firm position of this research that the parameters be hypertuned, in tandem with k-fold cross validation, by the user for their specific data set.

---

5. Results

This section will detail the statistical analyses used to assess each research question and interpretations of the results, along with figures and tables to more clearly display the data. This section goes on to detail the extra research done as a result of the answers to the research questions and statistical assessment of the progress made through this research.

5.1 Best Classification Algorithm without Augmentation

> *RQ1: Is there a classification algorithm that outperforms the rest without the use of any data augmentation?*

While the test for normality of residuals was met, a Levene's test found the variances to be not homogenous across the algorithms. An independent-samples Kruskal-Wallis test was thus carried out and found strong evidence ($P<0.001$) of a difference in the F-scores of the algorithms. This was followed up with a pairwise Dunn's test with a Bonferroni correction that found that all algorithms were significantly different ($P<0.05$) from one another with the exception of BNB with RF, XGB with SGD, and MLPC with MNB (figure 8).

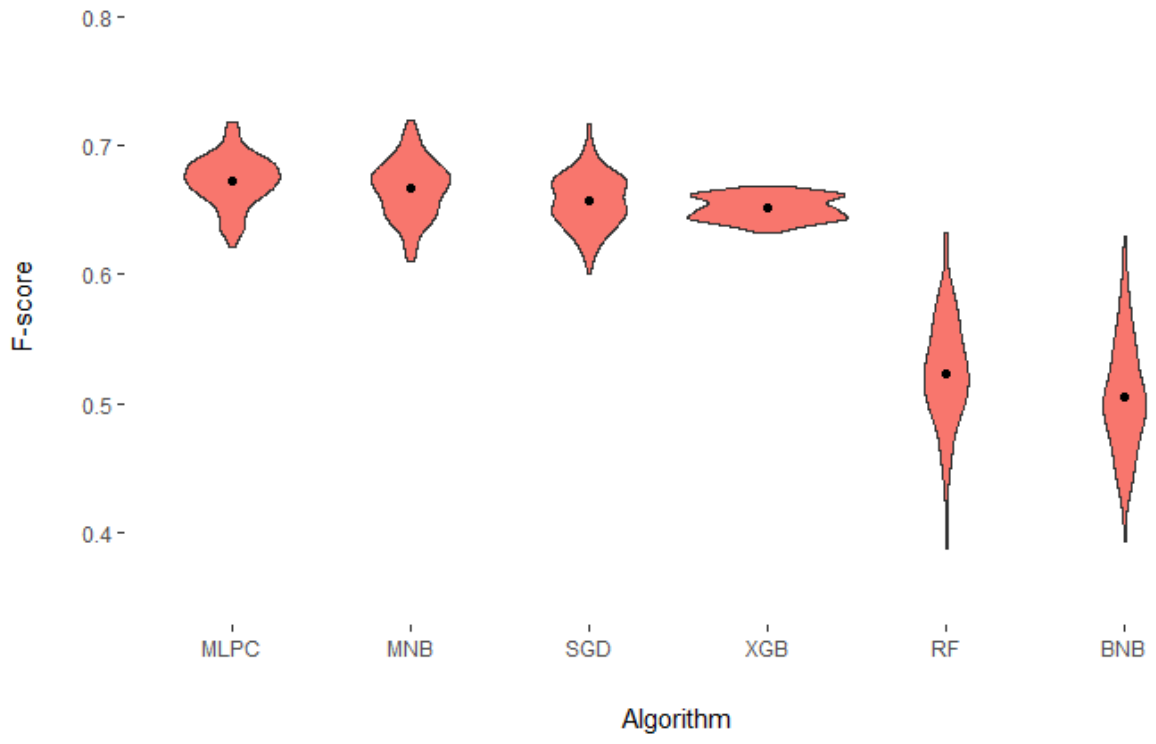

Figure 8: The mean F-scores and the distribution of data of all algorithms tested with no data augmentation.

5.2 Best Algorithm and Augmentation Pair

> *RQ2: Is there an optimal data augmentation technique and classification algorithm pair that outperforms the rest?*

The assumptions for an ANOVA were not met as a Levene's test found the variances to be heterogeneous ($P<0.001$). Thus, an independent-samples Kruskal-Wallis test was carried out that found that the data augmentation technique and classification algorithm combinations were significantly different from one another ($P<0.001$). This was followed by a Dunn's post-hoc test with a Bonferroni correction that found that the F-scores of the SGD and MNB algorithms were not significantly affected by the augmentation paired with it ($P>0.05$). The SGD, MNB and XGB algorithms were also consistently significantly lower ($P<0.05$) than any combination of RF or MLPC with an augmentation and so these were removed from the dataset for the next analysis (table 1). The BNB consistently performed below the value of the dummy classifier and was also removed from the results table.

The chance of a Type 1 error increases the more comparisons are made. A post-hoc test can mitigate this increase by introducing a family-wise error-rate of 0.05, but that then reduces the power of each comparison within the family. Therefore, the test was run again on just the RF and MLPC algorithms with each data augmentation pairings to increase the power of these individual comparisons. Significant differences were found between these combinations ($P=0.007$) (table 1) and a Dunn's post-hoc test with a Bonferroni correction found that no combinations were significantly different from one another ($P>0.05$) with the exception of MLPC with RLR which was significantly lower than MLPC with RWI and RWD.

We can see that there is no classification algorithm and data augmentation technique pairing that significantly outperforms the rest given this dataset. Instead, for further development of the model in Research Questions 3 and 4, we will be looking into more detail on the differences in performance of the data augmentation techniques to select the primary candidates.

After it was checked that assumptions for normality and homogeneity of variances were met, we ran a single-factor ANOVA and found no significant differences in the F-scores across the data augmentation techniques across percentages, or within the percentage bands with the exception of RLR which consistently significantly performed more poorly ($P<0.05$). Numerically, if not significantly, RSR and RWD had the highest F-scores overall, regardless of the percentage (0.664 and 0.664 respectively, table 2) and so, due to time constraints limiting the ability to run multiple combinations, these augmentation types were chosen for the next stage of the project.

Table 1: The mean F-score of the RF and MLPC algorithms combined with each augmentation type at 90% augmentation (57,000 sentences).

Algorithm	**Augment**	**F-score**
MLPC	RWI	0.7402
MLPC	RWD	0.7342
RF	RSR	0.7309
MLPC	RWS	0.7307
RF	RWI	0.7298
MLPC	RSR	0.7282
RF	RLR	0.7266
RF	RWD	0.7265
RF	RWS	0.724
MLPC	RLR	0.719

Table 2: The average F-scores of the augmentation types across all algorithms at different percentages and overall, the average across percentages.

Percentage	**Random Letter Replacement**	**Random Synonym Replacement**	**Random Word Deletion**	**Random Word Insertion**	**Random Word Substitution**
30	0.658	0.663	0.665	0.666	0.663
60	0.659	0.666	0.666	0.660	0.663
90	0.656	0.663	0.665	0.666	0.662
Overall	**0.659**	**0.664**	**0.664**	**0.650**	**0.662**

5.3 The Optimum Percentage Amount of Augmentation

RQ3: Is there an optimal percentage of data augmentation for this data augmentation technique and classification algorithm pair?

Due to the lack of a significantly superior data augmentation technique and classification algorithm pair in Research Question 2, we will be answering Research Question 3 by comparing the augmentation percentages of both RF and MLPC with either RSR or RWD. The assumptions of normality of residuals and homogeneity of variances were both met across the augmentation type and percentage of augmentation. A one-way ANOVA found that there was a significant effect of the percentage of augmentation on the final F-score ($P<0.001$). A subsequent Tukey post-hoc test found that the only significant difference was between no augmentation and any level of augmentation (30%, 60%, 90%) ($P<0.05$) (figure 9).

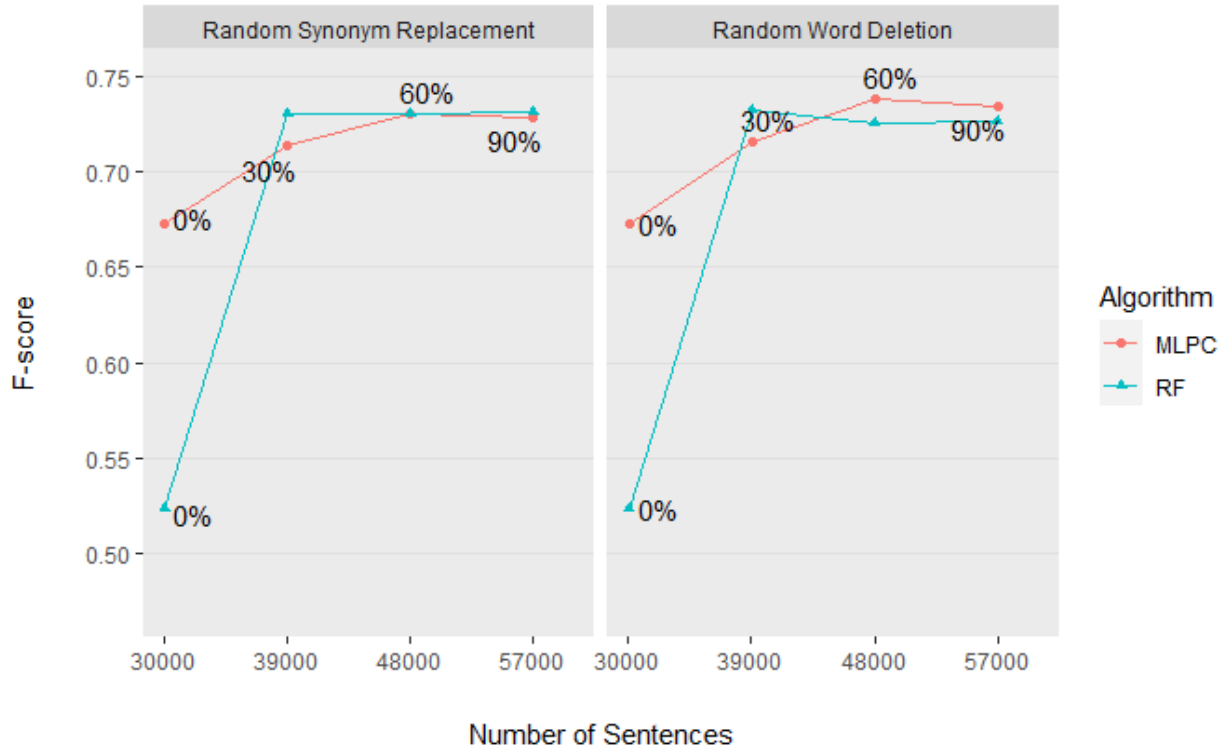

Figure 9: The effect of the percentage of augmented data on the final F-score of the MLPC and RF algorithms with either RSR or RWD augmentation techniques.

5.4 Optimum Cross-Augmentation Techniques

RQ4: Is there a combination of data augmentation techniques that improves the best model found in Research Question 3; section 5.3?

In table 1 we can see that the top ten performing classification algorithm and data augmentation technique combinations involve either RF or MLPC and in table 2 we can see that the top two data augmentation techniques are RSR and RWD. We attempted to use this data to improve the model development by combining RSR (used to augment 60% of the dataset) and RWD (used to further augment 30% of the dataset) to a total of 108% augmentation with MLPC and RF to see if they achieved a greater F-score. An independent-samples Kruskal-Wallis test found significant differences between the tests ($P<0.001$) and a post-hoc Dunn's test with a Bonferroni correction found that the classification algorithms with cross-augmentation always performed significantly better than their counterparts with a single augmentation at 90% ($P<0.05$) (figure 10). Due to time constraints we were unable to test this new cross-augmentation method at all percentage intervals of augmentation tested previously, but we can still see here that combining augmentations can improve the F-score significantly.

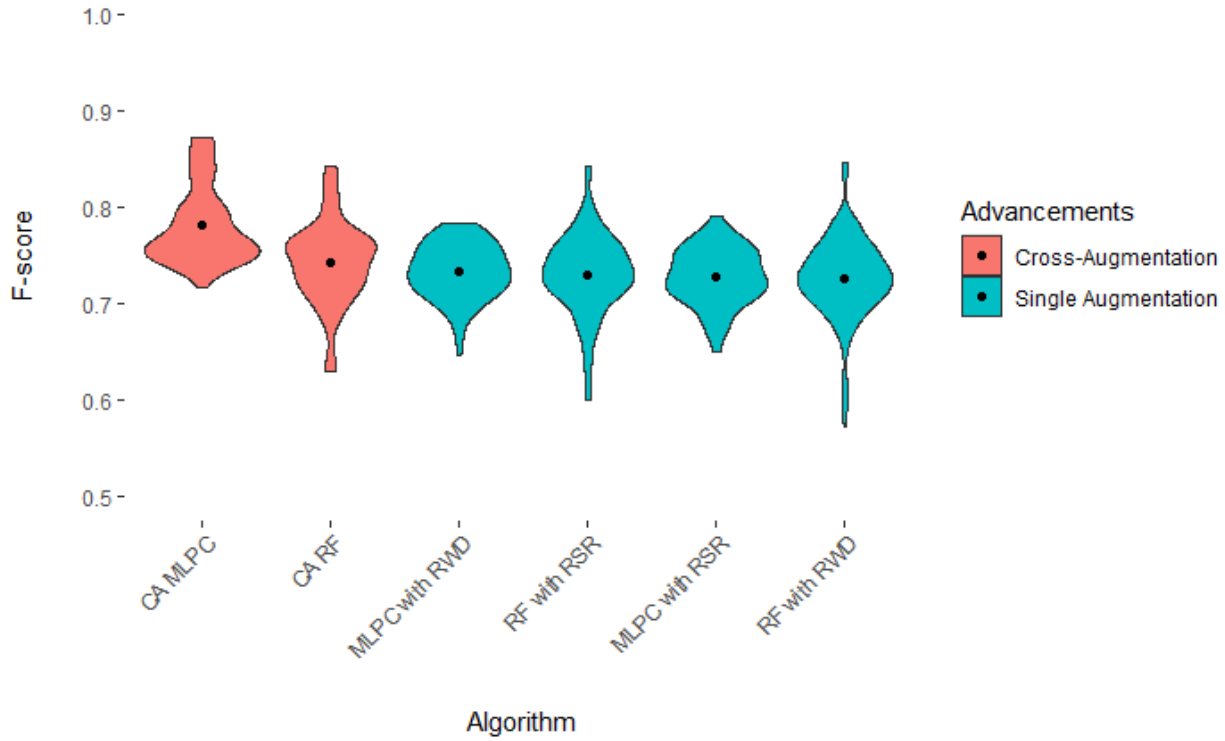

Figure 10. The means and the distributions of the F-scores of the different classification algorithms, RF and MLPC and the data augmentation techniques (RWD and RSR, both at 90% augmentation), separated in colour by adjustments to the augmentation type (Cross-Augmentation (CA) referring to 60% RSR and 30% RWD totalling 108% data augmentation).

5.5 Additional Research with TF-IDF

Following on from the initial research questions and our results at this stage we decided to modify other factors in an attempt to maximise the F-score of the final chosen model. This section will explore the most prominent adjustment and justifications for the developments made and the results found.

5.5.1 Increasing the max features of TF-IDF

Due to the time constraints of the project we were unable to test all factors at the maximum system requirements. TF-IDF's max features were limited to 500 during the previous designs despite this having the effect of lowering the final F-scores as raising it to just 1000 approximately doubled the computation time of most algorithms. Now that we have limited the models to MLPC and RF and the augmentations to a combination of 60% RSR and 30% RWD totalling 108% total augmentation, we can raise the TF-IDF to 2000 and see how high the F-scores would be under normal processing.

The data do not meet the assumptions of an ANOVA, and so an independent-samples Kruskal-Wallis test was carried out that found a strong indication of difference between the algorithms and variations ($P<0.001$). A Dunn's post-hoc with pairwise comparisons found no significant differences between MLPC and RF when using only one augmentation type. However, significant differences ($P<0.05$) were found between single-augmented and cross-augmented models, and again between the raising and non-raising of the max features of the TF-IDF to 2000 ($P<0.05$). This pairwise comparison also found that the most successful model was MPLC with cross-augmentation at 2000 TF-IDF which was significantly higher ($P=0.001$) at a mean F-score of 0.825 compared to the next best: RF with cross-augmentation at 2000 TF-IDF with a mean F-score of 0.789 (figure 11).

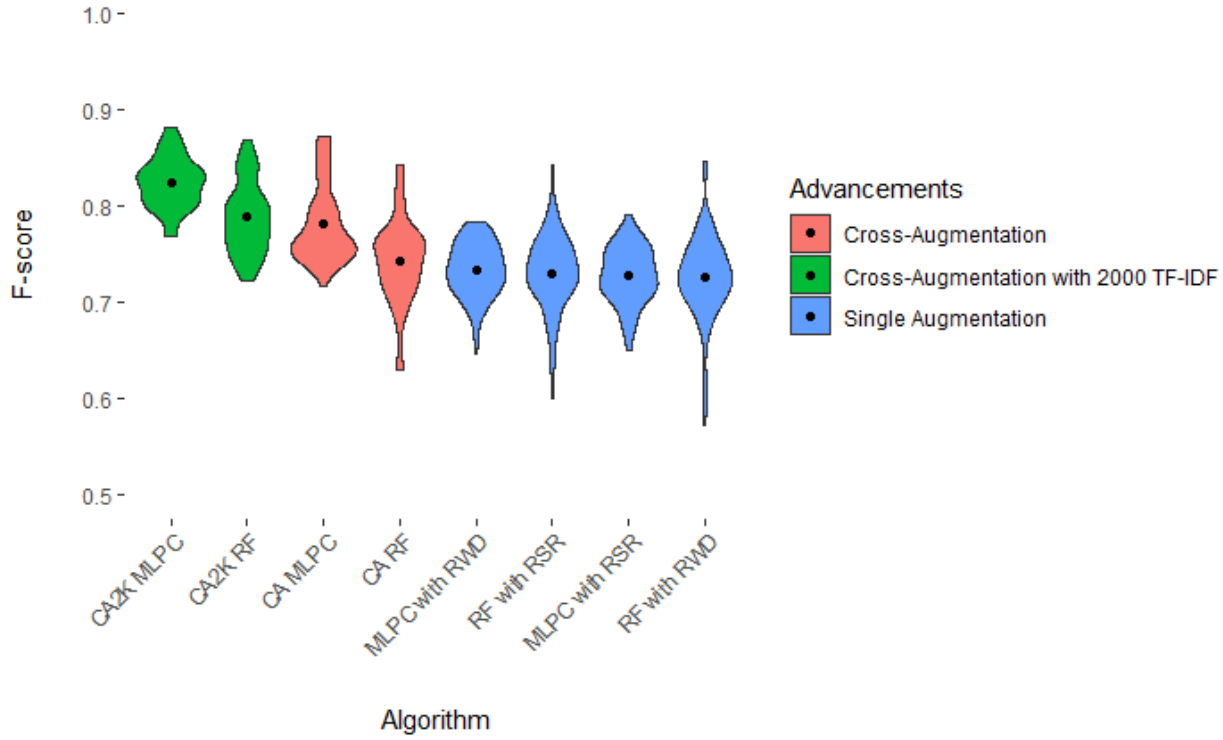

Figure 11. The means and the distributions of the F-scores of the different classification algorithms and the data augmentation techniques (RWD and RSR, both at 90% augmentation), separated in colour by adjustments to the augmentation type (Cross-Augmentation (CA) referring to 60% RSR and 30% RWD totalling 108% augmentation in total) and the 'max features' of the TF-IDF of 2000.

6. Discussion

This section will be exploring the results (Section 5) and looking at these data within the perspective of existing research explored in the literature review of this topic (Section 2). We will then go on to talk about how these results could be improved and where, based on the data we have investigated above, research could be targeted. We also explain what we believe would be 'best practice' for the use of these data in the final creation of a therapeutic assessment system in a medical setting.

6.1 The Discussion

Overall it was found that MLPC and MNB were the strongest classification algorithms for the dataset when no data augmentation occurred (figure 8). This was somewhat unexpected as neural networks (MLPC) are commonly associated with a demand for a larger dataset before consistently successful models can be made (Li *et al*., 2016). Importantly, the MLPC was only numerically more successful than the MNB, but not significantly higher, and so it is possible that this difference was merely down to chance, and the MNB should be considered as a possible equal. The data were assumed to have a multinomial distribution and so it was as expected that MNB outperformed GNB and BNB which were likely to be making predictions about the distribution of the datasets that did not match the reality.

Table 1 and figure 9 show how significant an impact data augmentation can make on the success of an algorithm. RF classification, previously performing at an F-score of 0.523 with 0% augmentation jumped up to 0.714 with any augmentation type, peaking at an average of 0.729 when combined with random synonym replacement at 90% data augmentation. On the other hand, the BNB classifier was not significantly improved by augmentation and consistently performed below the baseline of the dummy classifier regardless of the augmentation used. Previous research has found that BNB classifiers can be useful and effective forms of machine learning (Singh *et al*., 2019) and so it is likely that the issue with this algorithm is that it was not suited for the distribution of this specific dataset. Augmentation paired with all classification algorithms also led to MNB no longer being one of the top classification algorithms. This could be due to augmentation having a larger benefit to RF and MLPC, or could be due to augmentation reducing the association of the dataset to a multinomial distribution, causing the algorithm to less closely match the data.

We can see from figure 9 and table 2 that the level of data augmentation is much less important than the presence of data augmentation at all. It seems that, since there were no significant differences between the data augmentation amounts, that a minimum of 30% augmentation, 39,000 sentences, is enough to reliably produce a successful model with a good algorithm. This is especially useful as it reduces the computational cost of producing a model for use in a clinical setting, and can make further research faster and easier. This result in 5.3 is good food for thought as we begin to consider why it is that at the smallest augmentation percentage there is such a significant improvement, but no significance is seen in increasing from that amount as it then plateaus between 30%, 60% and 90%. We posit that, within this data, this is due to the creation of extra noise around each data point through augmentation, the models see significant improvement in their predictions. However, it is likely that only a small increase in the number of data points is required to allow the models to make the necessary connections and delineations. This noise doesn't affect the underlying class but simply improves the efficacy of the models.

When sorted by the F-score, and augmented to 90%, we can see that RF and MLPC make up the top ten classification algorithms, while the best augmentation type is less consistent. RF may be a strong performer in this context due to its robustness against outliers, unbalanced data, and non-linear data. As the original dataset of spoken word includes many broken sentences and interruptions (see Section 6.2.1 for more details), these features of the RF algorithm may allow it to better handle and classify this data as each decision tree only looks at a small subset in isolation. Similarly, neural networks have a high tolerance for noisy data and so, where RF can deal with inconsistencies through the breaking up of the data into smaller chunks, MLPC can stack many hidden layers together until the correct level of complexity is reached to match the complexity of the data.

Table 2 suggests that the best augmentations are RSR and RWD, both at F-scores of 0.664; however, the only significant difference between the augmentations was that the RLR was worse than the rest. The poor performance of RLR is likely due to the fact that it doesn't create as substantial an increase in relevant terms or usable sentences as compared to the other augmentations as it has a higher chance of creating nonsense words; potentially being miscategorized or poorly weighted by TF-IDF. RSR and RWD were chosen for further development in the model due to their numerically higher performance, but due to a lack of significant differences between the augmentations, RWS and RWI should be treated as equal performers if the opportunity arises for further investigation into the augmentations. This is especially important if considering the inclusion of a third or fourth augmentation into the cross-augmentation method to investigate if increasing the variation in augmentations can continue to improve the model on top of the improvements already seen after the cross-augmentation of two data augmentation techniques.

Through the additional research, beyond the initial research questions, we can begin to get an idea for what the most promising direction for the development of such a system would be. A mean F-score of 0.825 was reached with the combination of MLPC and cross-augmented RWD and RSR when the 'max features' of the TF-IDF were raised to 2000. Comparing this to the initial results of the Rawsthorne *et al*. (2020) position paper where initial findings of the label classification system had 74% accuracy, we can confidently state that we have made an improvement to the model as well as providing a robust series of recommendations for the further improvement of the model (Section 6.2) and for the implementation of the findings so far in the final program (Section 7.3).

6.2 Further Work

In this section we will explore what the results of this project have indicated to be promising fields for further research. This will begin with an insight to the drawbacks of this experiment's outcomes in terms of interpretability and then move on to how improved data preprocessing may improve the final result, and then how this, combined with a different data augmentation technique, back translation, may further improve the model.

6.2.1 Model and Prediction Interpretability

This project aims to build on the foundations laid out in Rawsthorne *et al.,*'s 2021 position paper ExTRA. One of the fundamental principles laid out in this paper was the desire that the decisions made in the algorithms be interpretable. Neither RF, nor the MLPC Neural Network are associated with easy interpretability of the decisions the algorithm has made. It is an aim of the final production of this therapeutic assessment program that not only will the system tell the user the decision that it has made regarding a session, but also why it has made that decision, i.e., what is in the text that led the system to interpret it the way that it did. In the world of neural networks, much research is being done into methods of elucidation of the decision process that is usually hidden in the 'hidden' layers and convoluted by the many nodes the information is passed through (Belinkov, Gehrmann & Pavlick, 2020; Zhang, Wu & Zhu, 2018). Pereira *et al*. (2018) present a novel method of increasing the interpretability of a RF algorithm, but it is important that whichever algorithm is chosen to be further developed for implementation, that methods for interpretability are also included.

6.2.2 API for text normalization

One piece of information that we have learned from this experimental process is the importance of the style and cleanliness of the base dataset. Being that this dataset is formed of transcripts of spoken English, the syntax and grammar is strongly different to the full sentences and correct grammar on which these classification algorithms are commonly developed. People interrupt each other in this dataset, many answers may have come in the form of body language such as shrugs and nods that don't get recorded in a transcript and it is common for someone to say something different to what they mean exactly in a way that can be interpreted with more complex in-person cues. What this means from a machine learning perspective is that the model must interpret meanings from sentences with more complex cues than in the interpretation of other common subjects such as news media, legal proceedings, or even tweets which are more commonly written in full sentences, with correct grammar, and with more thought in the word choice than the spoken word (Ayo *et al*., 2020; Singh *et al*., 2020).

We posit that one of the greatest constraints within this project then, is the structure of the sentences in the dataset. They are a direct transcription of what was said and, despite removing stop words, have a lack of any structure in many cases. Furthermore, a regulated sentence structure could also dramatically improve the effectiveness of TF-IDF which, following the additional research in 5.5.1, has a statistically significant impact on the results.

Therefore, we propose a big step forward would be to create an API capable of taking spoken sentence structures from session transcriptions and somewhat normalizing them for the use within the prediction algorithms of this project. We believe this is evidenced by the comparable success of the models in Youzhi (2009) at 99.83% accuracy, Elhadj (2009) at 96% accuracy, and Khan *et al*. (2019) at 98.22% where their sentences are normalized written sentence structures. It was highly sought after that this would be an available goal to achieve during this project however the time constraints built into this project, predominantly experimental computational time, made it impossible.

6.2.3 Back Translation Augmentation

We further posit the best augmentation method will be back-translation, for the purposes of text classification from therapeutic text. We believe the best language to use will be translating to French and back to English as the sentence structure and sentiment are kept very similar. Back translation was not used in this project because of the spoken sentence structure outlined in 6.2.2, in addition to being very computationally expensive. If an API for normalized sentence structure is developed, we believe a combination of it and back translation, using the French language, could produce the most significant results and be a substantial leap forward to the desired result of this project.

6.2.4 Augmentation Structures

One of the most novel features of this research that did not come to fruition, due to a lack of time, was the testing of what we called “augmentation structures”. The premise for this is simple: for every sentence we augment, we augment a minimum percentage on the first pass and then with each successive pass we incrementally augment a greater percentage of the original sentence. An experimental phase would be required to determine the optimum number of passes. We posit this is an avenue of research worth pursuing based on the results found in 5.3 which found significant improvements by adding noise around the data points. It is theorised that by increasing the amount of noise around data points, i.e. building augmentation structures, we amplify the success seen between 0% and 30% and step away from the lack of significant results in the augmentation amounts between 30%, 60% and 90%.

---

7. Conclusion

This section summarises the contents of the paper, the results and their relevance. We also take time here to reflect on the decisions made throughout the project, the practicalities of the project, the effectiveness of our own process and time-management and, finally, recommendations for the application of this research in terms of algorithm selection and augmentation techniques. We will start by looking at the data we have collected and assess our success in the context of the data type that we are working with and the limitations of the methods used. We will then assess our project management, comparing the expected and actual timescales and then how inexperience and computer processing time caused these changes. Finally, we will lay out our recommendations for implementation of the results of this research in the creation of the final therapeutic assessment program.

7.1 The Conclusion

This paper presents a series of tests into the effectiveness of multiple classification algorithms and data augmentation methods alone, in combination, and at different factor levels. The larger the dataset on which an algorithm can train, the more successfully the model can perform and so artificially inflating the datasets using the methods discussed in this paper present a method for increasing model accuracy without expending further time and/or effort to source more data.

It is unlikely that the processes explored above: algorithm development and data augmentation methods, will be enough to reach the high success rates required for a system used in a medical setting without additional cleaning to the base dataset for sentence normalisation and smoothing. However, we have seen that we can achieve high F-scores with these features alone and can see the most promising candidates with which to do so. Algorithm and augmentation development are only one step towards the final goal of a full system for the use of therapists to improve their clinical performance, but they are a key one to ensure that the classifications made by the final model are accurate, useful, and understandable.

At the conclusion of the experimentation and the analysis of the results, we identified that this project has found immense success in answering its research questions and finding invaluable significance in doing so. Whilst the depth of individual elements such as the algorithm parameter hypertuning or the lack of testing with back translation left us wanting, we can say with confidence the balance between the depth this experiment committed to, given the time-frame of the project, and the breadth it covered brought about great success and an unprecedented stepping-off point for future research.

7.2 Project Management and Reflections

As a life-science graduate, I am no stranger to experiments with large data or results and a lengthy scientific write-up; however this project handed me an entirely novel challenge. That challenge was deciding the depth of experimentation per individual element, i.e. algorithm or augmentation parameters, and begin with the breadth of testing. This was especially difficult as It was impossible to predict total run-time for the experiments. Throughout this entire project, there was a constant battle present in deciding when to stop experimenting and expanding and begin the full experimental implementation

itself. In my experience in life science, this has always been much more black and white and this was a particularly valuable experience.

Coming into this project, I was a complete novice within the field of machine learning having no prior experience and whilst I have learned a great deal, both in terms of machine learning and computational experiments, this inexperience had a few knock-on hindrances. Firstly, I grossly overestimated how much code was necessary, predominantly because of my inexperience with SKLearn in particular, and did not leave enough time for the experiments to be done at the level I, as a life science graduate, would have liked. Additionally, the final code for the experiment is clean and easy to use; however, it wasn't this way during the learning and development stage of the project and, upon reflection, better planning of what I wanted the code to look like before starting would have saved a lot of time. Finally, this inexperience also meant I grossly underestimated the time required for the experiments themselves. Before the additional research and advancing the research question answers, I had to collect 9,600 pieces of data. This took the best part of six weeks and included multiple alarms in the middle of the night to set off the next experimental loop. By the end of this project, the total amount of data collected is 12,001 pieces and, had I had experience in the field, I would have proportioned the time allocated to this task in the project plan much more accurately and professionally. Figure 12 highlights the disparities in how long I originally thought parts of the project would take against how long things took in reality. For example, we can see my original expectations for code implementation were four weeks but in actual fact it only took the best part of only two; before code optimisation. Additionally, we can see I had initially planned approximately three weeks for experimental testing but in reality it was much closer to six to seven weeks.

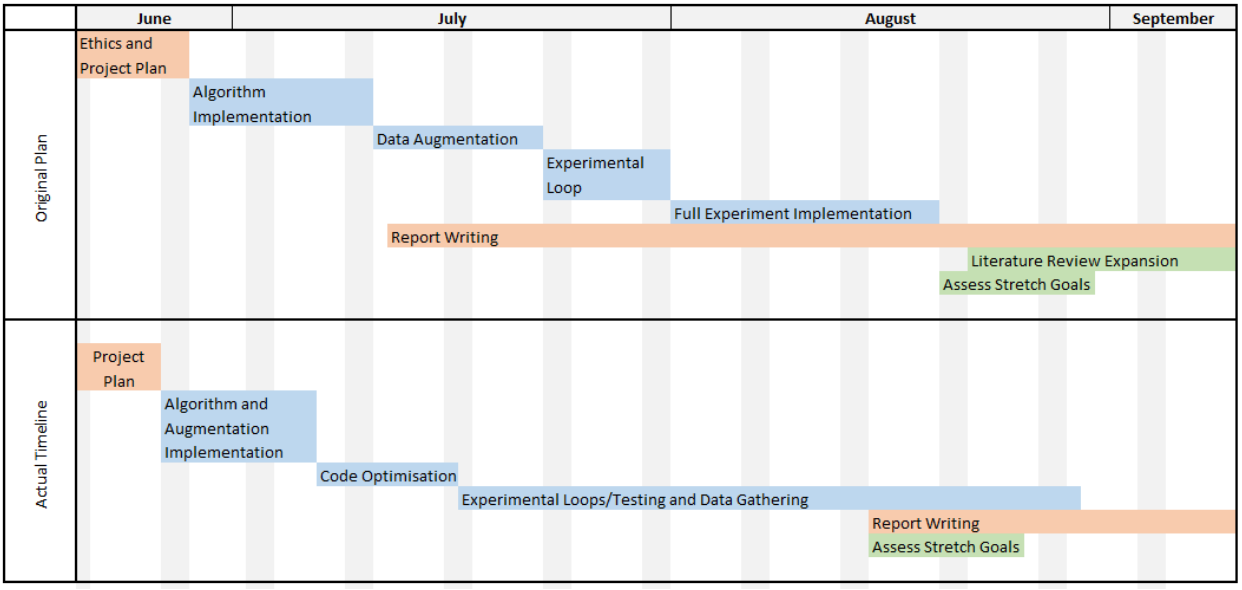

Figure 12: The original Gannt chart plan for time management for this project (above) compared to the final breakdown of the time taken throughout the project (below).

The stretch goals outlined in the project plan were assessed in mid-August however, due to the fact that the experimental phase had not yet concluded, it was decided they were not going to be chased for completion by the deadline of this project. Similarly, because of the extensive experimental phase of this

project, the writing for the project began in earnest later than originally anticipated. Whilst the writing was finished ahead of the deadline, its delayed start also played its part in the abandonment of stretch goals

Despite these setbacks, the progress made in this project, within the context of the much wider focus of this work, feels very positive. The experiments that ran in this project have found significance and isolated the best models for text classification within a therapeutic setting and, additionally, provided strong evidence for the success of data augmentation, the best technique and an optimum amount to use. With all this in mind, this project has set an incredible precedent for future projects in the grand scope of this topic and I am extremely proud of what has been achieved.

7.3 Recommendations for Applications of this Research

Here we lay out our recommendations for applications of this research. It is our educated opinion that these recommendations fall into two categories. Either what we have called Accelerated Testing, which is computationally inexpensive but, critically, not as effective. This method is useful when making changes outside of algorithm tuning to assess the value of what has changed. The second method we have called Best Performance Testing. This method will likely produce the best results possible, as it did with our data but, critically, it is extremely computationally expensive.

7.3.1 Accelerated Testing

If the user's purpose is to build on this research and is looking for a good starting point experimentally that is not computationally expensive then we would make the following brief recommendation. In terms of augmentation, RWD at 30% is inexpensive and found to be significantly one of the best augmentation techniques. For classification algorithms, we recommend MNB; or the bayes test that suits the user’s data. This paradigm did not produce the most impressive models but it was considerably faster, potentially 100 repetitions in just a few minutes, and provided a good baseline during the experimental process as to whether the value of a factor, not related to the classification algorithms, was an improvement, or not. Finally, we would strongly recommend limiting the TF-IDF max features to no more than 500. This value is still relatively computationally expensive but we found it to be the best trade-off between computational requirements and reward.

7.3.2 Best Performance Testing

My informed and professional recommendation, to build upon and adapt this research to the grander scope of its premise, and other research, would be to work with the MLPC algorithm exclusively. Similar models, in style and success, can be produced with RF however it has a much higher variance and therefore is much less accurate and reliable. In order to achieve the best results, as found in this project, we further lay out the following recommendations. Firstly, early-stopping will be principal in order to avoid overfitting and take some time off of producing the results, which will be extensive at best. Next, whilst other methods are available, it will be imperative that a grid search is used to find the best combination of parameter values for max iterations and hidden layer sizes based on the user’s data; we also recommend grid searching for tuning all hyperparameters. We found the greatest success setting the method alpha to 0.0001 and using the “adam” solver method. As mentioned previously, TF-IDF max

features were limited to 500 for the experimental phase because, despite producing lower F-scores, we had to be realistic with the time available for this project. Raising these to 2000 (Section 5.5.1) would be my recommendation, as we found significant differences ($P<0.05$); to its 500 counterpart.

We also found there was always a significant improvement to the models produced when using cross-augmentation methods and, though not statistically tested, that using cross-augmentation reduced the variance by increasing the lowest value F-score models. It would also be my recommendation to use an approach similar to this over single-factor augmentation. As for the augmentation techniques advised, BERT types are the most computationally expensive but preserve sentiment the best and so we would recommend testing, where possible, with a combination of BERT methods primarily. If testing time is an issue, we found significant success using 60% RSR on the original data set and 30% RWD on the new data set with RSR included. Neither of these use BERT however we once again had to make a sensible decision based on how much time is available in this project and how much can be done without the results.

These recommendations, with this data, will lead to the best possible results found in this project with individual model F-scores >0.9 and an average of 0.825 however, with all these features in place the processing time of these experiments, to achieve 100 repetitions as to keep with the implementation of this project, would be vast. This project was run on a single personal computer with 99 percentile components for personal use computers, based on data at the system requirements lab (System Requirements Lab, 2021), and this type of loop for 100 iterations could take 48 hours or more. For a grand perspective on time, that would take approximately a minimum of 32 days per algorithm using each augmentation type and invariably significantly more if testing cross augmentation combinations and their implementation amount combinations.

Whilst, for the purpose of this experiment, these specifications were not achievable because of the time constraints, if time is not an issue and the user has access to a vastly improved, or multiple, computer(s) these are my strongest recommendations for applications of this research in order to produce the best models possible.

---

8. Glossary

A glossary of all abbreviations used.

Context Terms	**Abbreviation**
Natural language processing	NLP
Paraphrase database	PPDB
Bidirectional Encoder Representations from Transformer	BERT
Natural language toolkit	NLTK

Algorithm Terms	**Abbreviation**
Random forest	RF
Multi-layer perceptron neural network	MLPC
Gradient boost classifier	XGB
Support vector machine	SVM
Stochastic gradient descent	SGD
Bernoulli naive bayes classifier	BNB
Multinomial naive bayes classifier	MNB
Gaussian naive bayes classifier	GNB
Convolutional neural networks	CNN

Augmentation Terms	**Abbreviation**
Random synonym replacement	RSR
Random word substitution	RSS
Random letter replacement	RLR
Random word insertion	RWI
Random word deletion	RWD

9. Bibliography

Amato, F., Coppolino, L., Cozzolino, G., Mazzeo, G., Moscato, F. and Nardone, R., 2021. Enhancing random forest classification with NLP in DAMEH: A system for Data Management in eHealth Domain. Neurocomputing, 444, pp.79-91.

Ayo, F., Folorunso, O., Ibharalu, F. and Osinuga, I., 2020. Machine learning techniques for hate speech classification of twitter data: State-of-the-art, future challenges and research directions. Computer Science Review, 38, p.100311.

Baeza-Yates, R. and Ribeiro-Neto, B., 2011. Modern Information Retrieval. Addison Wesley, pp. 327-328

Belinkov, Y., Gehrmann, S. and Pavlick, E., 2020. Tutorial Proposal: Interpretability and Analysis in Neural NLP. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. [online] 2020 Association for Computational Linguistics, pp.1-5. Available at: <https://doi.org/10.18653/v1/P17> [Accessed 20 August 2021].

Bing, L., Chaudhari, S., Wang, R. and Cohen, W., 2015. Improving Distant Supervision for Information Extraction Using Label Propagation Through Lists. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing,.

Breiman, L, 2001. Random Forests. Machine Learning 45, 5–32

Chen, S., Jones, P., Underwood, B., Moore, A., Bullmore, E., Banerjee, S., Osimo, E., Deakin, J., Hatfield, C., Thompson, F., Artingstall, J., Slann, M., Lewis, J. and Cardinal, R., 2020. The early impact of COVID-19 on mental health and community physical health services and their patients' mortality in Cambridgeshire and Peterborough, UK. Journal of Psychiatric Research, 131, pp.244-254.

Chowdhury, G. (2003) Natural language processing. Annual Review of Information Science and Technology, 37. pp. 51-89.

Denœux, T., 2000. A neural network classifier based on Dempster-Shafer theory. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 30(2), pp.131-150.

Denisko, D. and Hoffman, M., 2018. Classification and interaction in random forests. Proceedings of the National Academy of Sciences, 115(8), pp.1690-1692.

Devlin, J., Chang, M., Lee, K. and Toutanova, K., 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Elhadj, Y., 2009. Statistical Part-of-Speech Tagger for Traditional Arabic Texts. Journal of Computer Science, 5(11), pp.794-800.

Fuomo, D., 2017. A Gentle Introduction To Neural Networks Series — Part 1. [online] Medium. Available at: <https://towardsdatascience.com/a-gentle-introduction-to-neural-networks-series-part-1-2b90b87795bc> [Accessed 22 August 2021].

Ghanem, B., Rosso, P. and Rangel, F., 2021. Stance Detection in Fake News: A Combined Feature Representation. In: Proceedings of the

First Workshop on Fact Extraction and VERification (FEVER). Association for Computational Linguistics, pp.66-71.

Goldberg, Y. and Hirst, G., 2017. Neural Network Methods in Natural Language Processing. San Rafael: Morgan & Claypool Publishers.

House of Commons Library, 2021a. Mental Health Policy in England. CBP 07547

House of Commons Library, 2021b. Mental Health Statistics for England, prevalence, services and funding. House of Commons Library.

Hussain, S., Fatima, T., Riaz, R., Shahla, S., Riaz, F. and Jin, S., 2019. A Comparative Study of Supervised Machine Learning Techniques for Diagnosing Mode of Delivery in Medical Sciences. International Journal of Advanced Computer Science and Applications, 10(12).

Jacob, L., Smith, L., Armstrong, N., Yakkundi, A., Barnett, Y., Butler, L., McDermott, D., Koyanagi, A., Shin, J., Meyer, J., Firth, J., Remes, O., López-Sánchez, G. and Tully, M., 2021. Alcohol use and mental health during COVID-19 lockdown: A cross-sectional study in a sample of UK adults. Drug and Alcohol Dependence, 219, p.108488.

Imel, Z., Steyvers, M. and Atkins, D., 2015. Computational psychotherapy research: Scaling up the evaluation of patient–provider interactions. Psychotherapy, 52(1), pp.19-30.

Kam Ho, T. "Random decision forests," Proceedings of 3rd International Conference on Document Analysis and Recognition, 1995, pp. 278-282 vol.1, doi: 10.1109/ICDAR.1995.598994.

Kensert, A., Alvarsson, J., Norinder, U. and Spjuth, O., 2018. Evaluating parameters for ligand-based modeling with random forest on sparse data sets. Journal of Cheminformatics, 10(1).

Kilbourne, A., Beck, K., Spaeth-Rublee, B., Ramanuj, P., O'Brien, R., Tomoyasu, N. and Pincus, H., 2018. Measuring and improving the quality of mental health care: a global perspective. World Psychiatry, 17(1), pp.30-38.

W., Khan, Daud, A., Khan, K., Nasir, J. A., Basheri, M., Aljohani, N., Alotaibi, F. S. 2019, "Part of Speech Tagging in Urdu: Comparison of Machine and Deep Learning Approaches," in IEEE Access, vol. 7, pp. 38918-38936

Kluegl, P., Toepfer, M., Beck, P., Fette, G. and Puppe, F., 2014. UIMA Ruta: Rapid development of rule-based information extraction applications. Natural Language Engineering, 22(1), pp.1-40.

Kolomiyets, O., Bethard, S. and Moens, M., 2011. Model-Portability Experiments for Textual Temporal Analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: shortpapers. [online] Association for Computational Linguistics, pp.271-276. Available at: <https://dl.acm.org/doi/pdf/10.5555/2002736.2002793> [Accessed 16 August 2021].

Liddy, E.D. 2001. Natural Language Processing. In Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc.

Lopes, F., Ferreira, J. and Fernandes, M., 2019. Parallel Implementation on FPGA of Support Vector Machines Using Stochastic Gradient Descent. Electronics, 8(6), p.631.

Meehl, P., 1978. Theoretical risks and tabular asterisks: Sir Karl, Sir Ronald, and the slow progress of soft psychology. Journal of Consulting and Clinical Psychology, 46, pp.806-838.

Melville, P., Gryc, W. and Lawrence, R., 2009. Sentiment analysis of blogs by combining lexical knowledge with text classification. Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09,.

MHCYP, NHS Digital. 2018. Mental Health of Children and Young People in England, 2017 [PAS] - NHS Digital. [online] Available at: <https://digital.nhs.uk/data-and-information/publications/statistical/mental-health-of-children-and-young-people-in-england/2017/2017> [Accessed 24 May 2021].

MHCYP, NHS Digital. 2020. Mental Health of Children and Young People in England, 2020: Wave 1 follow up to the 2017 survey - NHS Digital. [online] Available at: <https://digital.nhs.uk/data-and-information/publications/statistical/mental-health-of-children-and-young-people-in-england/2020-wave-1-follow-up> [Accessed 24 May 2021].

Mijwel, M., 2018. Artificial Neural Networks Advantages and Disadvantages. [online] Available at: <https://www.researchgate.net/profile/Maad-Mijwil/publication/323665827_Artificial_Neural_Networks_Advantages_and_Disadvantages/links/5aa2c01faca272d448b5a23d/Artificial-Neural-Networks-Advantages-and-Disadvantages.pdf> [Accessed 13 August 2021].

Mykowiecka, A., Marciniak, M. and Kupść, A., 2009. Rule-based information extraction from patients' clinical data. Journal of Biomedical Informatics, 42(5), pp.923-936.

NHS Digital, 2016. Adult Psychiatric Morbidity Survey: Survey of Mental Health and Wellbeing, England, 2014.

O'Connor, R., Wetherall, K., Cleare, S., McClelland, H., Melson, A., Niedzwiedz, C., O'Carroll, R., O'Connor, D., Platt, S., Scowcroft, E., Watson, B., Zortea, T., Ferguson, E. and Robb, K., 2020. Mental health and well-being during the COVID-19 pandemic: longitudinal analyses of adults in the UK COVID-19 Mental Health & Wellbeing study. The British Journal of Psychiatry, pp.1-8.

Pallant J, 2007. SPSS survival manual, a step by step guide to data analysis using SPSS for windows. 3 ed. Sydney: McGraw Hill; pp. 179–200.

Pavlick, E., Rastogi, P., Ganitkevitch, J., Van Durme, B. and Callison-Burch, C., 2015. {PPDB} 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). [online] Association for Computational Linguistics, pp.425-430. Available at: <http://paraphrase.org/#/download> [Accessed 17 August 2021].

Pedregosa, F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot., M and Duchesnay E., 2011. Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830, 2011

Pereira, S., Meier, R., McKinley, R., Wiest, R., Alves, V., Silva, C. and Reyes, M., 2018. Enhancing interpretability of automatically

extracted machine learning features: application to a RBM-Random Forest system on brain lesion segmentation. Medical Image Analysis, 44, pp.228-244.

Pierce, M., Hope, H., Ford, T., Hatch, S., Hotopf, M., John, A., Kontopantelis, E., Webb, R., Wessely, S., McManus, S. and Abel, K., 2020. Mental health before and during the COVID-19 pandemic: a longitudinal probability sample survey of the UK population. The Lancet Psychiatry, 7(10), pp.883-892.

Ranganathan, S., Gribskov, M., Nakai, K. and Schönbach, C., 2019. Encyclopedia of bioinformatics and computational biology. 1st ed. Amsterdam: Elsevier, pp.403-412.

Reback J., *et al.,* (2020). pandas-dev/pandas: Pandas 1.0.3 (v1.0.3). Zenodo. https://doi.org/10.5281/zenodo.3715232

Rawsthorne, M., Jilani, T., Andrews, J., Long, Y., Clos, J., Malins, S. and Hunt, D., 2020. ExTRA: Explainable Therapy-Related Annotations. In: Proceedings of the 2nd Workshop on Interactive Natural Language Technology for Explainable Artificial Intelligence (NL4XAI 2020). [online] Dublin: Association for Computational Linguistics, pp.11-15. Available at: <https://www.aclweb.org/anthology/2020.nl4xai-1.4.pdf> [Accessed 24 May 2021].

Razno, M., 2019. Machine Learning Text Classification Model with NLP Approach. In: Proceedings of the 3d International Conference Computational Linguistics And Intelligent Systems. [online] Kharkiv: Kharkiv Polytechnic Institute, pp.71-73. Available at: <http://ena.lp.edu.ua/bitstream/ntb/45487/2/2019v2___Proceedings_of_the_3nd_International_conference_COLINS_2019_Workshop_Kharkiv_Ukraine_April_18-19_2019_Razno_M-Machine_learning_text_classification_71-73.pdf> [Accessed 24 May 2021].

Rizos, G., Hemker, K. and Schuller, B., 2019. Augment to Prevent. Proceedings of the 28th ACM International Conference on Information and Knowledge Management, [online] pp.991-1000. Available at: <https://dl.acm.org/doi/abs/10.1145/3357384.3358040?casa_token=IalVYZO_Vm8AAAAA:QFnAnsAgzXSye1vRn3ILVH0fndxen5cxjX4_ekVxvER5QFpLiZx7h7hqGkN7oSkoDH1dvU1ViqaK> [Accessed 16 August 2021].

Sachdeva, S., Kumar, B., 2021 Comparison of gradient boosted decision trees and random forest for groundwater potential mapping in Dholpur (Rajasthan), India. Stoch Environ Res Risk Assess 35, 287–306.

Shalev-Shwartz, S. and Ben-David, S., 2014. Understanding Machine Learning From Theory To Algorithms. New York: Cambridge University Press, pp.1-30.

Singh, G., Kumar, B., Gaur, L. and Tyagi, A., 2019. Comparison between Multinomial and Bernoulli Naïve Bayes for Text Classification. In: 2019 International Conference on Automation, Computational and Technology Management (ICACTM). [online] London, UK: IEEE. Available at: <https://ieeexplore-ieee-org.ezproxy.nottingham.ac.uk/abstract/document/8776800/authors#authors> [Accessed 12 August 2021].

Singh, M., Bhatt, M., Bedi, H. and Mishra, U., 2020. Performance of bernoulli's naive bayes classifier in the detection of fake news. In: Materials Today: Proceedings. [online] Available at: <https://doi.org/10.1016/j.matpr.2020.10.896.> [Accessed 12 August 2021].

Stevenson, D., Farmer, P. 2017. Thriving at Work: The Stevenson/Farmer review of mental health and employers.

Systemrequirementslab.com. 2021. *System Requirements Lab*. [online] Available at: <https://www.systemrequirementslab.com/Marketing/Home.html> [Accessed 8 September 2021].

Tseng, S., Baucom, B. and Georgiou, P., 2017. Approaching Human Performance in Behavior Estimation in Couples Therapy Using Deep Sentence Embeddings. Interspeech 2017.

Wang, P., Xu, B., Xu, J., Tian, G., Liu, C. and Hao, H., 2016. Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. Neurocomputing, 174, pp.806-814.

Wang, Y., Sohn, S., Liu, S., Shen, F., Wang, L., Atkinson, E., Amin, S. and Liu, H., 2019. A clinical text classification paradigm using weak supervision and deep representation. BMC Medical Informatics and Decision Making, 19(1).

Wei, J. and Zou, K., 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).

Zhang Youzhi, "Research and implementation of part-of-speech tagging based on Hidden Markov Model," 2009 Asia-Pacific Conference on Computational Intelligence and Industrial Applications (PACIIA), 2009, pp. 26-29

Zhang, Q., W, Y. and Zhu, S., 2021. Interpretable Convolutional Neural Networks. In: Conference on Computer Vision and Pattern Recognition. [online] pp.8827-8836. Available at: <https://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_Interpretable_Convolutional_Neural_CVPR_2018_paper.html> [Accessed 24 August 2021].

Zolfaghar, K. and Aghaie, A., 2011. Evolution of trust networks in social web applications using supervised learning. Procedia Computer Science, 3, pp.833-839.