DEPRESSION SIGNS DETECTION WITH NLP

A Machine Learning Approach to Detect Depression Signs in Social Media Text

By

Paula García Serrano

A THESIS

Submitted in partial fulfillment of the requirements for the degree of

BACHELOR IN DATA AND BUSINESS ANALYTICS

IE UNIVERSITY

May 2022

This thesis has been approved in partial fulfillment of the requirements for the Degree of BACHELOR IN DATA AND BUSINESS ANALYTICS.

School of Science and Technology

Thesis Advisor: *Noa Cruz Díaz, PhD.*

Acknowledgments

To all my family and friends who supported me during this study.

Thank you to Noa for being there whenever I needed her help and guiding me during this unique and challenging process.

Abstract

This paper explores how artificial intelligence (AI) algorithms can be applied to detect signs of depression from social media text in English. It does so by training the models on a very recent corpus, extracted from a Codalab competition: Detecting Signs of Depression from Social Media Text-LT-EDI@ACL 2022. The study covers classical machine learning, deep learning and transfer learning algorithms to solve the task, and compares the results among the three categories. Lastly, a website has been created so that any individual can test the predictions of the model over the desired text, regardless of his or her programming capabilities.

1. Introduction

According to the World Health Organization (2021), 3.8% of the population, approximately 280 million people, suffer from depression globally. In addition, more than 700,000 people commit suicide every year, making it the fourth leading cause of death for people between 15 and 29 year-olds as of 2019. Therefore, it becomes crucial to identify depressed individuals because when their depression is recurrent and of moderate to severe intensity, it can lead to suicide. Consequently, we must identify depressed individuals on time and act accordingly, to reduce these figures for the years to come.

To do so, there are some ongoing initiatives, such as the Mental Health Gap Action Program (mhGAP), which goals to assist people with mental, neurological, and substance use disorders through care provided by health workers who are not mental health specialists; or MenteScopia[1], a multimedia project to spread information about mental illnesses and their prevention in Spain. Other non-public initiatives include the service that Instagram launched in 2022. By searching for the tags #anxiety and #depression, users will have a question popping up from the Instagram app: "Can we help you?". This new window allows users to choose from three options: generalist advice that can help, contact a friend or family member and, finally, call the Hope Phone, a telephone psychological consultation service.

Among the symptoms of depression, one can find a loss of enjoyment or interest in activities and thoughts of death or suicide, among many others. This fact suggests that social media might be an excellent source to obtain the data from these people and analyze whether these symptoms are reflected in the postings.

[1] More about the MenteScopia initiative can be found on https://psynal.eu/mentescopia/

Furthermore, current treatments include face-to-face psychological individual or group treatments, such as behavioral activation, cognitive behavioral therapy, interpersonal psychotherapy, and antidepressant medications dispensed by professionals and supervised by non-specialized therapists. As perceived, the antidepressants administered vary between the different levels of depression and should never be the first choice for mild depression. Therefore, it is essential to detect the signs of depression and quantify their level in three different classes: mild, moderate, and severe.

A spike in depression episodes should be expected in moments of crisis, such as financial crises, pandemics, wars, relationship break-ups, or chronic pain and illness. Although some are personal, such as relationship break-ups or chronic pain and illness, others are public, such as financial crises, pandemics, and wars. It would be beneficial for administrations around the globe to have a tool to detect these depression episodes spikes and act fast to prevent them from becoming more dangerous. Having an AI algorithm that analyzes the social media texts from citizens and detects the level of signs of depression present would help to act more efficiently.

Therefore, this paper is dedicated to developing the aforementioned AI algorithm and making it publicly available for anyone to use and detect signs of depression in the provided text.

2. Literature Review

When looking at existing research regarding automated depression detection, there are different methods that researchers can follow when extracting the data needed.

The first class is patient-interaction-dependent, which means that we need direct interaction with the patient to extract the data. These include clinical interviews (AI Hanai et al., 2018),

physical exams with depression scales (Havigerová et al., 2019), and monitoring of facial and speech modulations (Nasir et al., 2016).

However, some others belonging to the patient-interaction-independent class do not require direct contact with the patient. These include videos and audio (Morales & Levitan, 2016), live journals (Nguyen et al., 2014), blog posts (Tyshchenko, 2018), and the core of this paper: social media. This second data extraction method is handy in targeting mental health disorders. There is a lot of stigma around acknowledging suffering from a mental illness. Therefore, the data retrieved might be more natural if the patient can avoid visiting a clinic in-person. However, it is essential to point out that the patient should always rely on an expert's diagnosis.

When relying on social media to train depression detectors, most of the research done up until today works with the same dataset, the E-Risk@CLEF-2017 (Losada et al., 2017). However, other datasets exist for the same purpose, represented in Table 1.

Existing Corpus	**Social Media Source**	**Type of classification**	**Class Labels**
Eichstaedt et al. (2018)	Facebook	Binary	Depressed; Not Depressed
Nguyen et al. (2014)	Live Journal	Binary	Depressed; Control
Tyshchenko et al. (2018)	Blog post	Binary	Clinical; Control
Deshpande & Rao (2017)	Twitter	Binary	Neutral; Negative
Lin et al. (2020)	Twitter	Binary	Depressed; Not Depressed
Reece et al. (2017)	Twitter	Binary	Post-Traumatic Stress Disorder; Depression
Tsugawa et al. (2015)	Twitter	Binary	Depressed; Not Depressed
Losada et al. (2017)	Reddit	Binary	Depression; Not Depression
Wolohan et al. (2018)	Reddit	Binary	Depressed; Not Depressed
Tadesse et al. (2019)	Reddit	Binary	Depression Indicative; Standard
Pirina & Çöltekin (2018)	Reddit	Binary	Positive; Negative

Existing Corpus	Social Media Source	Type of classification	Class Labels
Yao et al. (2020)	Reddit	Four classes	Depression; Suicide Watch; Control; Opiates

Table 1: Existing Corpus for Depression Detection

Furthermore, when choosing social media as the data source for this kind of research, it is important to specify the source platform. These platforms include Facebook (Eichstaedt et al., 2018) and Instagram (Reece and Danforth, 2017). However, one of the most popular ones tends to be Twitter. It offers a convenient API to connect to a Twitter Developer account, allowing users to retrieve tweets easily. The data from this platform can be retrieved by querying specific tweets. Coppersmith et al. (2015) decided to look for tweets that explicitly stated, “I was just diagnosed with depression”. In contrast, Deshpande & Rao (2017) opted for looking at some keywords, including “depressed”, “hopeless”, and “suicide”. Another interesting platform to retrieve the data from is Reddit. Reddit possesses a large amount of text discussion. As for the case of Twitter, the data from this platform can be retrieved by filtering subreddits. In the study by Wolohan et al. (2018), the chosen keywords were “r/depression help, r/aww, r/AskReddit, r/news, r/Showerthoughts, r/pics, r/gaming, r/depression, r/videos r/todayilearned r/funny", whereas Pirina & Çöltekin (2018) used “r/anxiety, r/depression and r/depression_help", and Yao et al. (2020) opted for “r/suicidewatch, r/depression".

After extracting the data of interest, it then needs to be annotated to be able to use an AI model to automatically detect depression signs. Previous research suggests using surveys (Reece et al., 2017), questionnaires (Tsugawa et al., 2015), or having it annotated by two annotators (Wolohan et al., 2018).

Once the data is collected and annotated, researchers can follow different approaches to creating the target AI model. First of all, features need to be extracted from the text data. To

do so, Paul et al. (2018) suggest the Bag Of Words (BOW) vectorizer, Stankevich et al. (2018) suggest using the Term Frequency - Inverse Document Frequency (TF-IDF) vectorizer, Word Embeddings, and Bigrams and Coello Guilarte (2019) advocate for using Word2Vec and GloVe. On a more innovative approach, Villatoro Tello et al. (2019) present the use of graph-based representations to extract these features, as they can reflect contextual information. It does so by denoting the co-occurrence of terms through the graph's edges.

After extracting the features, researchers can use different algorithms to classify the data. These can be divided into machine learning, lexicon-based, and transfer learning.

The machine learning models research for detecting depression suggests using Logistic Regression, Multi-Layer Perceptron, Convolutional Neural Network, and Long Short-Term Memory Network (Jiménez Campfens, 2020), Multinomial Naïve Bayes, Decision Tree, and Support Vector Machine (Siurob Palomero, 2019) and a Multitask Learning framework (Benton et al., 2017).

Furthermore, the lexicon-based models are a collection of recognized and precompiled sentiment terms. This approach is further divided into the dictionary-based and corpus-based methods (Razak et al., 2020), making these models a good fit for sentiment analysis tasks. Some models included in this approach for detection of depression are a lexicon-based Comments-oriented News Sentiment Analyzer (LCN-SA) (Moreo et al., 2012), Linguistic Inquiry and Word Count (LIWC), and Latent Dirichlet Allocation (LDA) (Eichstaedt et al., 2018). In addition, Resnik et al. (2015) proved that LDA serves to extract the latent structure for automatic identification of relevant topics for depression detection. The reason why these models have not been tested in the proposed methodology of this paper is that they require expert knowledge to build the rules and are difficult to maintain, as the

rules need to be updated constantly, which makes them difficult to apply to data extracted from social media, where the content and forms of expression are continually changing.

On the other hand, the latest trends in the Natural Language Processing (NLP) area highlight transfer learning to create systems for these tasks. Transfer learning consists of the relocation of knowledge from an already learned task to the new task at hand (Torrey & Shavlik, 2010), and one way of doing so is by using pre-trained models. As Qiu et al. (2020) explain, pre-trained models can learn universal language representations, making them beneficial for downstream NLP tasks and reusing them as part of transfer learning. The first generation of these pre-trained models was similar to what deep learning is nowadays because they were trained using word embeddings and were context-free. However, the second generation has shown promising results because they can learn contextual word embeddings. Overall, the training of these pre-trained models involves two steps: first, the model is trained on a huge amount of unlabeled datasets, and second, the model is fine-tuned on labeled data for a specific downstream NLP task (Ruder et al., 2019).

To evaluate the performance of all the models explained above, researchers can use multiple metrics. Apart from the traditional ones, including accuracy, precision, recall, area under the curve, or f1-score, Villatoro Tello et al. (2019) propose the use of ERDE (Early Risk Detection Error), a metric that accounts not only for the exactness of the system but also for the time the system takes to classify the instances.

After investigating the existing research around automatic depression detection, this paper aims first to use a different and more updated dataset than the majority of the studies use. Second, explore all the three modeling approaches to see which one drives better results. Furthermore, the ultimate purpose of this research study is for society to be able to benefit

from it. For that purpose, a website will be created so that individuals can see the model's predictions for their proposed text regardless of their programming capabilities.

3. Data and Methodology

3.1. Data Source

As explained in Table 1, up until today, the existing datasets only propose a binary classification for the corpus instances: depressed or not depressed.

Therefore, the author of this paper chose the corpus used in this study because it is the first one created to fill the gap in existing research: it allows researchers to classify the data into three different classes, quantifying the level of the depression signs found. This corpus belongs to an ongoing competition (as of May 2022) hosted on Codalab: "Detecting Signs of Depression from Social Media Text-LT-EDI@ACL 2022"[2], which focus is detecting the signs of depression in a person from their social media postings in English. The expected result of the competition is a system that classifies the signs of depression into not depressed, moderately depressed, and severely depressed, using the following labels: "not depression", "moderate", and "severe" (Kayalvizhi et al., 2022). This proposed study will be beneficial as the model will not only detect depression but also denote the level of signs of depression, which will allow for a more personalized and accurate action plan for the person designated as depressed.

3.2. Data Extraction

According to Kayalvizhi et al. (2022), the corpus mentioned above was created following these steps:

[2] The competition can be found on: https://competitions.codalab.org/competitions/36410

Firstly, a suitable social media platform needed to be chosen. For this purpose, researchers determined that Reddit was an appropriate platform since it is open source and has more textual data than any other social media platform among the ones described in the literature review. Furthermore, the content inside this platform is organized into postings, including one or more user statements.

Secondly, researchers need to define a way to extract the data from that source once the data source is chosen. To do so, they decided to use the Python Reddit API Wrapper (PRAW)[3]. However, authentication from the Reddit platform is required to use the API. Therefore, researchers needed to create a unique client key to which a unique client id is assigned.

Thirdly, researchers need to select the subreddits of interest when the data source is chosen, and the authentication is working. Finally, it is fundamental to define the researcher's keywords to query the platform to extract the data of interest. In this case, as the objective was to collect people's discussions about their mental health, the scraped keywords were: r/Mental Health, r/depression, r/loneliness, r/stress, r/anxiety.

Fourthly, when executing the query defined, researchers get access to the following information about each subreddit: post ID, title, URL, publish date, name of the subreddit, the score of the post, and the total number of comments, from which they decided to only keep: PostID, title, text, URL, date and subreddit name.

Lastly, the data is stored in Comma Separated Values (.csv) format and uploaded to the Codalab competition, where participants can retrieve it, as long as they are registered for the competition.

[3] More information about the PRAW library can be found on: https://praw.readthedocs.io/en/stable/

3.3. Data Annotation

Once the data is extracted using the method described in the section above, it needs to be annotated to be able to train a predictive model with it. For this specific use case, researchers found the annotation guidelines challenging to frame, as the mental health of an individual needs to be analyzed using single postings. However, they developed the following guidelines (Kayalvizhi et al., 2022):

An instance of the dataset is labeled as “not depression” if:

- The statements have only one or two lines about relevant topics
- The statements reflect momentary feelings of a present situation
- The statements are about asking questions about any medication
- The statements are about asking/seeking help for friend’s difficulties

An example of a text classified as “not depression” is:

> *The ups and downs : I have a habit, or maybe more of a tactic, of avoidance. I distract myself between breakdowns and never actually face any of my issues. I make lists of things I should do to make it better and then never accomplish them. Theres constantly this feeling that i am drowning but I'm holding my own head underwater. I dont know the best way to approach this and the wya I process and deal with things emotionally ranges from a complete denial and invalidation of my own feelings to depressive/anxious episodes that last for days, sometimes involving me getting so upset I incessantly punch my legs or stay in bed for hours and hours on end (I enjoy working out and so consequently I'll beat myself up for neglecting that). How do you stop yourself when you know you're spiraling?*

On the other hand, an instance of the dataset is labeled as “moderate” if:

- The statements reflect a change in feelings (feeling low for some time and feeling better for some time)
- The statement shows that they aren’t feeling completely immersed in any situations
- The statements show that they have hope for life

An example of the “moderate” class is:

I will probably end it when my mum isn't around anymore. : I can say with certainty that I have tried hard to make my life one worth living but my ongoing depression is more present and crippling than ever recently.

I would never do anything while my mum is still alive because I know she wouldn't cope. I'm her only daughter and we are very close. But I'm clueless as to how to live with this fog that has followed me since my teenage years, especially if I don't have her in my life.

It's as if the depression becomes heavier the older I get, despite the sertraline and hours and hours of therapy. I just can't see myself living a long and fulfilled life.

Similarly, an instance of the dataset is labeled as “severe” if:

- The statements express more than one disorder condition
- The statements explain the history of suicide attempts

An example of this class is the following:

Money... : So, I just got a student loan and they originally told me that I’d be paid a monthly living allowance, which would be fine. That would be perfect because I could pay my bills and still have a little bit left over each month. I still haven’t been a paid a living allowance, and thus when I called them; they told me that I was given the WRONG information... and that it was all in a lump sum. So now, I’ve paid my bills... and according to them, $1,000 is enough to live on for 4 months at a time. Paying for bills, food, and rent... so now I’m even deeper into my depression because I’m not going to be able to afford my antidepressants... much less barely being able to afford my bills

I hate life

Once the annotation guidelines have been established, other individuals must annotate the corpus independently, according to the agreed rules. The purpose of this is that there is consistency in the annotation decisions. This research measured the consistency using the inter-rater agreement with Cohen's kappa coefficient (Cohen, 1960) and estimated using a per-annotator empirical over the class labels (Artstein and Poesio, 2008). This measurement represents the degree of agreement among the annotators, resulting in a kappa value (k) of 0.686 (Kayalvizhi et al., 2022).

Taking Landis and Koch's (1977) measurement table as a reference, represented in Table 2, the obtained k value by the researchers is considered to represent a substantial agreement between the annotators.

Kappa Value (k)	Strength of agreement
< 0	Poor
0.01 - 0.20	Slight
0.21 - 0.40	Fair
0.41 - 0.60	Moderate
0.61 - 0.80	Substantial
0.81 - 0.99	Almost perfect agreement

Table 2. Landis & Koch measurement table of inter-rater agreement

3.4. Data Preprocessing

When building the competition datasets, the researchers performed a basic cleaning of the data by removing non-ASCII characters and emoticons. After this, to fill the missing values, they combined the “title” and “text” columns into a single “text data” column (Kayalvizhi et al., 2022).

As the datasets come from social media, a more profound cleaning was performed: all the text was converted to lowercase, the full expressions substituting abbreviations (e.g., don't → do not, it's → it is, he'll → he will) and unwanted characters were removed, including tags or mentions (e.g., @name), hashtags, weblinks, remaining emojis, punctuation, trailing whitespaces and stop words. Furthermore, all the text was tokenized and stemmed using Porter stemming (Porter, 1980).

3.5. Feature Extraction

After having the data cleaned, numerical features need to be extracted out of the raw text for the models to understand this text data. The feature extraction step is necessary for machine learning models but not for pre-trained models. That is because pre-trained models include an encoding step, in which text is translated into numbers and, therefore, do not need

vectorizers. According to the suggestions in the literature review section, different vectorizers have been tested according to the model categories.

For the traditional machine learning models, the vectorizers tested were:

- **BoW**: When using BoW, "each document in a corpus is represented by a vector whose length equals the number of unique terms, also known as vocabulary. The number of documents of the corpus is denoted by N and the number of terms of the vocabulary by n. The number of times the i^{th} term t_i occurs in the j^{th} document is denoted by tf_{ij}, where $i = 1, 2, ..., n$ and $j = 1, 2, ..., N$" (Paul et al., 2018, p. 4).
- **TF-IDF**: in this case, "the number of documents in which a particular term appears is knowns as document frequency, denoted by df_i, and how frequently a term occurs in a corpus is known as inverse document frequency, defined as $idf_i = \log(N/df_i)$. The weight of the i^{th} term in the j^{th} document, denoted by w_{ij}, is calculated by combining the term frequency with the inverse document frequency as follows: $w_{ij} = tf_{ij} \times idf_i = tf_{ij} \times \log(N/df_i)$, $\forall\ i = 1, 2, ..., n$ and $\forall\ j = 1, 2, ..., N$. Each document of the corpus (d_j) is considered to be a vector d_j, where the i^{th} component of the vector is w_{ij}, i.e., $d_j = (w_{1j}, w_{2j}, ..., w_{nj})$" (Paul et al., 2018, p. 5).
- **N-grams**: consist of contiguous sequences of items, which can be phonemes, syllables, letters, words, or base pairs (Stankevich et al., 2018), that inspect the co-occurrence of terms. In this specific implementation, (1,2) n-grams were added to the BoW and the TF-IDF vectorizers when training classical machine learning models, which account for unigrams and bigrams.

In the case of the deep learning algorithms, the technique tested has been word embeddings, where words are represented as real-value vectors in a predefined vector space. Therefore, it is frequently used with deep learning algorithms, as each term is mapped to one vector, and

the vector values represent a neural network. This representation is based on the usage of words, which facilitates that terms used in similar ways are represented similarly (Brownlee, 2019). Different embeddings sizes were tested in the experiments, including 50, 100, 200, and 300 dimensions. Two famous implementations of word embeddings are:

- **Word2Vec**: as developed by Mikolov et al. (2013), the word embedding is learned together with the neural network model. The vector is initiated with small random values, and the layer is fit using a backpropagation algorithm. This approach learns the embedding targeted to the specific data and specific task.
- **GloVe**: consists of pre-trained word embeddings released under a public domain license by Pennington et al. (2014) that can be applied to other deep learning tasks. These word embeddings were trained on a dataset of one billion tokens with a vocabulary of 400 thousand words with different embedding sizes, including 50, 100, 200, and 300 dimensions.

In addition to these implementations, word embeddings can also be learned as part of the deep learning model. The embedding layer is initialized with random weights and then learns embeddings for all the train set words.

3.6. Model Creation

After deciding on the techniques used to extract the numerical features from the raw text, the modeling approach needs to be established. This approach has been divided into classical machine learning, deep learning, and transfer learning.

The traditional machine learning algorithms tested are the following:

- **Logistic Regression**, initially proposed by Berkson (1944), helps understand the relationship between the dependent and independent variables by estimating probabilities using a logistic regression equation.
- **Decision Tree**, developed by Morgan & Sonquist (1963), predicts the target value based on the decision rules, formed using the Gini index and entropy for information gain and features to identify the target variable.
- **Random Forest**, created by Ho (1995), combines many decision trees to generate the final prediction by bootstrap aggregation and bagging.
- **eXtreme Gradient Boosting (XGBoost)**, an implementation of gradient boosted decision trees designed for speed and performance, originated by Chen & Guestrin (2016).
- **Adaptive Boosting (AdaBoost)**, created by Freund & Schapire (1997), consists of a collection of N estimator models that assigns higher weights to the misclassified samples in the next model.
- **Multinomial Naïve Bayes**: the Naïve Bayes algorithm uses the Bayes Theorem (Bayes, 1958) to predict belonging probabilities for each class. The Multinomial variant is implemented with discrete features (e.g., word counts for text classification).
- **K-Nearest Neighbors (KNN)**, attributed to Fix & Hodges (1951), the algorithm classifies each data instance by plotting it and finding the similarity between K data points according to a specified distance metric.
- **Support Vector Machine (SVM)**, developed by Vapnik & Lerner (1963), projects the data into higher dimensions to classify it using hyper-planes. Then, the specified kernel transforms the input data into the required form. In this case, the linear and radial kernels were tested.

- **Multilayer Perceptron (MLP)**, an artificial neural network trained with the backpropagation of the error, was developed by Rosenblatt (1958).

Furthermore, the deep learning algorithms tested for this kind of task were:

- **Convolutional Neural Networks (CNN)**: initially developed by LeCun & Bengio (1995), they are generally used in computer vision. However, in recent years, this algorithm's results in NLP have been quite promising. The algorithm is trained to recognize patterns across space, where when a particular pattern is detected, each convolution returns the corresponding result. Furthermore, given the kernel parameter of this algorithm, patterns of different sizes can be identified, which allows for identifying different sequences of words.
- **Recurrent Neural Network (RNN):** this class of neural networks, originated by Rumelhart et al. (1985), is trained to recognize patterns across time. The algorithm works by using sequential data, which allows identifying relationships with previous steps in time. This algorithm is beneficial for NLP tasks because it helps preserve the meaning of the sequential text data.

Within RNNs, further variants of the algorithm exist. However, the Long Short-Term Memory and the Gated Recurrent Units variants stand out because they help mitigate the vanishing gradient problem. The gradient reaches such small values that these are no longer useful to train the model. Therefore, these variants were also tested:

- **Long Short-Term Memory (LSTMs)**: developed by Hochreiter & Schmidhuber (1996), consists of a memory cell that stores information by computing an input gate, a forget gate, and an output gate to manage this memory. In this case, they help capture long-distance dependencies in the instances of Reddit posts.

- **Gated Recurrent Units (GRUs)**: proposed by Cho et al. (2014), they modulate the flow of information using gating units without having a separate memory cell. They compute an update gate and a reset gate, which control the flow of information through each hidden unit.

As seen by Qiu et al. (2020), pre-trained models show promising results for NLP tasks. Therefore, the Roberta model was fine-tuned over the subject study of this paper. Furthermore, Roberta is an improved version of Bert because, according to Liu et al. (2019), Bert was significantly undertrained when evaluating the effects of hyperparameter tuning and training set size. Therefore, the implemented improvements to create Roberta were:

- Training the model longer, with bigger batches, over more data.
- Removing the next sentence prediction objective.
- Training on longer sequences.
- Dynamically changing the masking pattern applied to the training data.

Developed by Devlin et al. (2018) at the Google AI Language laboratory, Bert is an algorithm for deep preliminary learning of bidirectional text representation. It stands out because it is easy to use, as the user only needs to add one output layer to the existing architecture to obtain the wanted predictions (Koroteev, 2021). Before the creation of Bert, models such as GPT were limited to left-to-right analyses, which did not allow for analyzing subsequent tokens. Therefore, when creating Bert, scientists were careful enough to use a masked language approach, which allows for predicting randomly selected and masked words in a text by only considering the surrounding context.

3.7. Model Evaluation

The dataset needs to be divided to evaluate the performance of these models over unseen data. In this specific case, the competition organizers already divided the data into three sets following the traditional train-test split approach, whose characteristics are explained in Table 3.

One can observe that the 'Label' feature is only present in two of the three datasets, which only allows the use of the train and development sets to build the predictive model. As the performance of the models cannot be evaluated in the proposed test set, the development set will act as the test set, and the train set will be divided into train and development itself, following a 70-30 stratified ratio.

	Nº observations	**Column names**	**Nº instances "not depression"**	**Nº instances "moderate"**	**Nº instances "severe"**
Train	8891	"Text_data", "Label"	1971	6019	901
Development	4496	"Text data", "Label"	1830	2306	360
Test	3245	"text data"	-	-	-

Table 3. Components of the provided datasets

It is also relevant to observe in Table 3 that there is a class imbalance in the proposed datasets, which can lead to lower performance of the models trained. Therefore, all the algorithms will be tested over the original imbalance dataset and an undersampled dataset when carrying out the experiments. When building this unbalanced dataset, the attention needs to be on the number of instances N belonging to the minority class, in this case, the "severe" class. In doing so, a random sample of N instances is extracted from the other two classes (i.e., N instances of the "moderate" class and N instances of the "not depression"

class). Therefore, the resulting undersampled dataset consists of the same number of instances for each class.

Table 4 shows some critical metrics of the proposed datasets, divided by class, which include: the number of sentences, the average document length (calculated in terms of sentences), the number of words, and the average sentence length (calculated in terms of words). As observed, the characteristics for each class are not consistent across tables. For example, if we look at the average document length and the average sentence length, these are bigger for the "severe" class in the train set but are more extensive for the "moderate" class in the development set.

Dataset	Class	Nº sentences	Avg. document length (in sentences)	Nº words	Avg. sentence length (in words)
Train	not depression	7,884	4	153,738	78
	moderate	36,114	6	601,900	100
	severe	9,911	11	126,140	140
Development	not depression	3,660	2	10,980	6
	moderate	66,874	29	804,794	349
	severe	2,880	8	75,240	209
Test		29,205	9	369,930	114

Table 4. Characteristics of the provided datasets

Once the data is split, a standard metric to evaluate all the experiments must be chosen. However, as this dataset is proposed within a competition, the metric has already been established as the macro f1-score. To understand this measure, it is helpful to first look at the f1-score, the harmonic average of precision and recall. The precision is the percentage of true positive instances from all the positive classified instances, whereas the recall is the percentage of positive classified instances that are true positives. When looking at these

measures on a macro level, the observed results derive from averaging each label, making these macro-measures very useful for multiclass classification problems. The macro f1-score is then calculated as the harmonic mean of macro-precision and macro-recall (Santos et al., 2011).

4. Results

The best results for each modeling approach have been included in the tables of this section. In addition, the results of each proposed model configuration are displayed in the more detailed tables in the Appendix. All the results are expressed in macro f1-score and obtained using the Scikit-Learn[4] confusion matrix function (Pedregosa et al., 2011).

Table 5 shows the best results of each of the traditional machine learning algorithms. These results have been achieved using the Scikit-Learn implementation of the algorithms (Pedregosa et al., 2011), except for the case of XGBoost, which was implemented using the Xgboost library[5]. Random forest drives the highest macro f1-score results when fed with the undersampled data and having the features extracted with the TF-IDF vectorizer with N-grams. The configuration for this model was 100 estimators, Gini criterion, two minimum samples split, one minimum sample leaf, zero minimum weight fraction leaf, auto maximum features, zero minimum impurity decrease, random split at ten, with bootstrap, no out-of-bag samples, and no warm start.

[4] The Scikit-Learn library can be found on: https://github.com/scikit-learn/scikit-learn
[5] The Xgboost library can be found on: https://github.com/dmlc/xgboost

Model	Feature Extraction	Dataset	Macro F1-Score
Logistic Regression	BoW with N-grams	Original	0.51
Decision Tree	BoW	Undersampled	0.41
Random Forest	**TF-IDF with N-grams**	**Undersampled**	**0.52**
XGBoost	TF-IDF	Original	0.49
Ada Boost	BoW with N-grams	Undersampled	0.44
Multinomial Naïve Bayes	BoW with N-grams	Original	0.37
KNN	BoW	Original	0.39
SVM with Linear Kernel	BoW with N-grams	Original	0.50
SVM with Radial Kernel	TF-IDF	Undersampled	0.49
MLP	BoW with N-grams	Undersampled	0.48

Table 5: Best Traditional Machine Learning Results

Similarly, Table 6 shows the best results for the deep learning algorithms. In this case, all of them were implemented using the Keras library[6], part of the TensorFlow framework, and were trained using five epochs. As displayed in the table, the CNN algorithm was the best performing one, configured using a GloVe embedding layer of 200 dimensions, a convolutional layer with 32 filters and relu activation, a max-pooling layer of pool size of two, a flattening layer, a dense layer of size ten with relu activation and a dense layer of size three with sigmoid activation.

It is important to note that neither of the configurations suggested in the methodology section was enough for the LSTM nor the GRU algorithms to predict well. As shown in Table 10 and Table 11 in the Appendix, the performance of both algorithms was always below 0.27 in terms of macro f1-score. Moreover, in most cases, they were only predicting just one class.

[6] The Keras library can be found on: https://github.com/keras-team/keras

Model	Embedding Type	Embedding Size	Dataset	Macro F1-Score
CNN	**GloVe**	**200**	**Original**	**0.44**

Table 6: Best Deep Learning Results

Lastly, Table 7 shows the best results of each transfer learning experiment. In this category, the models were trained with five epochs and using the PyTorch[7] (Paszke et al., 2019) implementation of the Transformers[8] library (Wolf et al., 2019). Furthermore, the Roberta adaptation to Twitter text developed by the Research Group in Natural Language Processing at Cardiff University[9] was tested. This adaptation was trained on 128.06M tweets until the end of March 2022, making it more updated for social media text. This Roberta adaptation drove the best results for the proposed task, with no cleaning of the original imbalance dataset.

Model	Preprocessing	Dataset	Macro F1-Score
roberta-base	No cleaning	Original	0.53
twitter-roberta-base-mar2022	**No cleaning**	**Original**	**0.54**

Table 7: Best Transfer Learning Models Results

To conclude, the best model in terms of the metric performance turned out to be the twitter-roberta-base-mar2022, with a score of 0.54. However, when deciding what model to use for the task, the metric, the latency, and the explainability of the model need to be taken into account altogether. When looking at the latency and the explainability, the best model turns out to be the random forest, which additionally has a score of 0.52 in terms of macro f1-score. Therefore, taking the evaluation factors into account altogether, the random forest is the best overall to complete the task of depression signs detection. The reason for it is that,

[7] The PyTorch library can be found on: https://github.com/pytorch/pytorch
[8] The Transformers library can be found on: https://github.com/huggingface/transformers
[9] The CardiffNLP model can be found on: https://huggingface.co/cardiffnlp/twitter-roberta-base-mar2022

when speaking about mental health disorders detection, the researcher needs to be able to offer transparency and explain how the prediction turned out to be what it was.

Therefore, the files used to develop this random forest model have been added to this GitHub repository: https://github.com/paulagarciaserrano/depression-detection-system. In addition, the best configured Roberta model was uploaded to the Hugging Face platform and found on this link for research purposes: https://huggingface.co/paulagarciaserrano/Roberta-depression-detection.

5. Model Serving

The primary purpose of this research study was for society to be able to consume the final result, so a website has been made publicly available to ease that task. This website allows every individual to test the proposed model on their preferred text, regardless of their programming capabilities. Furthermore, it explains to the users what the model consists of and allows them to input their text and instantly get the model prediction. The website is hosted on the pythonanywhere.com platform and can be accessed through this link: http://paulagarciaserrano.pythonanywhere.com/.

A demo of the website can be found in this tutorial: https://youtu.be/z5P1FTkgq24.

6. Discussion

The results section shows that all the best models' results reach a plateau at a macro f1-score of between 0.5 and 0.54. This fact suggests there might be a problem with the quality of the data provided; the problem might rely on the datasets these different algorithms are being trained with, not on the algorithms themselves.

As Gupta et al. (2021) explain, although researchers and practitioners have shown interest in improving the performance of models, there are little efforts towards improving the data quality when facing machine learning tasks. Suppose the data quality issues are not solved before inputting that data into the models. In that case, these can substantially impact the efficiency and accuracy of the machine learning algorithms, resulting in inaccurate analytics and unreliable decisions.

The suspicion around the data quality issues first arises when inspecting the inter-rater agreement explained in the data annotation section. This inter-rater agreement achieved a kappa value of 0.686, which lies in the "Substantial Agreement" section of Table 2. However, as observed in this same table, the Substantial section's lower bound is 0.61, which is very close to the kappa value achieved by the annotators of this corpus. The interpretation of this kappa is that the annotators have only agreed on 68.6% of instances during the data annotation process, which suggests that the overall task is complicated, making the data very difficult to annotate. Therefore, substantial efforts are required to improve the data annotation process (e.g., improving the guidelines or adding more annotators to the process).

In this line of research, Landing AI has developed what they call Data-Centric AI[10], which suggests systems programmed with a focus on data rather than code. The approach provides a systematic method for improving data, reaching a consensus on the data, and cleaning up inconsistent data.

Apart from this, as one can observe in Table 3, the text data column is named differently in each of the given datasets: "Text_data" in the training dataset, "Text data" in the development dataset, and "text data" in the test dataset. Although this does not directly impact the model's performance, it gives a ground base to think that there might be other small mistakes

[10] More about the Data-Centric AI initiative can be found on: https://landing.ai/data-centric-ai/

impacting the consistency of the proposed datasets.

Unfortunately, the proposed methodology results cannot be compared to those of other participants from the competition because the author of this paper has not been granted access to the test dataset used to evaluate the models of other participants. However, by looking at the competition's ranking to date (May 2022) in the Codalab platform, the best model has a macro f1-score of 0.5830, the second-best 0.5523, and the third-best 0.5467. The complete ranking of the competition can be found in Table 14 of the Appendix.

7. Conclusion

This paper has presented an end-to-end methodology to extract, annotate and classify social media texts to identify signs of depression. It does so by using a very recent corpus obtained from a Codalab competition, which allows for identifying signs of depression in the texts and quantifying them.

Different modeling approaches have been proposed in the proposed methodology, such as classical machine learning, deep learning, and transfer learning, which give a holistic approach to the task at hand.

The promising results show how difficult detecting signs of depression can be, even for humans, and the following section includes recommendations on how the obtained results could be improved.

8. Recommendations

The suggestions can be divided into two sections to improve the proposed methodology results: dataset creation and modeling.

For the dataset creation section, the suggestions are:

- Diversify the platform from where the data is extracted.
- To generalize better, use a broader keyword combination to query the platforms that not only includes words related to depression.
- Create more specific guidelines for the annotators to follow.
- Include more annotators in the process.
- Try data augmentation techniques.

On the other hand, for the modeling stage, the suggestions are:

- Try the Doc2Vec vectorizer.
- Include lexical and syntactical information in the classical machine learning models.
- Try word embeddings for classical machine learning models.
- See the performance of alternative pre-trained models.
- Use a loss function that penalizes the incorrectly classified samples of the minority class.
- Test a two-step model that first classifies instances into “depression” and “no depression” and then further classifies the depressed into “severe” and “moderate”.
- Use cross-validation for the evaluation of the models.
- Use a metric that accounts for class imbalance, such as the weighted f1-score.

Bibliography

Al Hanai, T., Ghassemi, M. M., & Glass, J. R. (2018, September). Detecting Depression with Audio/Text Sequence Modeling of Interviews. In *Interspeech* (pp. 1716-1720).

Artstein, R., & Poesio, M. (2008). Inter-coder agreement for computational linguistics. *Computational linguistics, 34*(4), 555-596.

Bayes, T. (1958). An essay towards solving a problem in the doctrine of chances. Biometrika, 45(3-4), 296-315.

Benton, A., Mitchell, M., & Hovy, D. (2017). Multi-task learning for mental health using social media text. *arXiv preprint arXiv:1712.03538*.

Berkson, J. (1944). Application of the logistic function to bio-assay. *Journal of the American statistical association, 39*(227), 357-365.

Brownlee, J. (2019, August 7). *What Are Word Embeddings for Text?* Machine Learning Mastery. https://machinelearningmastery.com/what-are-word-embeddings/

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Coello Guilarte, D. L. (2019, August). *Clasificación translingüe para la detección de depresión en usuarios de Twitter* (Doctoral dissertation, Universidad Politécnica de Tulancingo).

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement, 20*(1), 37-46.

Coppersmith, G., Dredze, M., Harman, C., & Hollingshead, K. (2015). From ADHD to SAD: Analyzing the language of mental health on Twitter through self-reported diagnoses. In *Proceedings of the 2nd workshop on computational linguistics and clinical psychology: from linguistic signal to clinical reality* (pp. 1-10).

Deshpande, M., & Rao, V. (2017, December). Depression detection using emotion artificial intelligence. In *2017 international conference on intelligent sustainable systems (iciss)* (pp. 858-862). IEEE.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dinkel, H., Wu, M., & Yu, K. (2019). Text-based depression detection on sparse data. *arXiv preprint arXiv:1904.05154*.

Eichstaedt, J. C., Smith, R. J., Merchant, R. M., Ungar, L. H., Crutchley, P., Preoţiuc-Pietro, D., ... & Schwartz, H. A. (2018). Facebook language predicts depression in medical records. *Proceedings of the National Academy of Sciences, 115*(44), 11203-11208.

Fix, E., & Hodges, J. L. (1951). Nonparametric discrimination: consistency properties. Randolph Field, Texas, Project, 21-49.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. J*ournal of computer and system sciences, 55*(1), 119-139.

Gupta, N., Mujumdar, S., Patel, H., Masuda, S., Panwar, N., Bandyopadhyay, S., ... & Munigala, V. (2021, August). Data quality for machine learning tasks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (pp. 4040-4041).

Havigerová, J. M., Haviger, J., Kučera, D., & Hoffmannová, P. (2019). Text-based detection of the risk of depression. *Frontiers in psychology, 10*, 513.

Ho, T. K. (1995, August). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition* (Vol. 1, pp. 278-282). IEEE.

Hochreiter, S., & Schmidhuber, J. (1996). LSTM can solve hard long time lag problems. *Advances in neural information processing systems, 9*.

Jiménez Campfens, J. N. (2020). *Estudio de Detección de Depresión en Redes Sociales mediante Procesamiento de Lenguaje Natural y Aprendizaje Automático* (Doctoral dissertation, Universitat Politècnica de València).

Kayalvizhi, S., & Thenmozhi, D. (2022). Data set creation and empirical analysis for detecting signs of depression from social media postings. *arXiv preprint arXiv:2202.03047*.

Koroteev, M. V. (2021). BERT: a review of applications in natural language processing and understanding. *arXiv preprint arXiv:2103.11943*.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *biometrics*, 159-174.

LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks, 3361*(10), 1995.

Lin, C., Hu, P., Su, H., Li, S., Mei, J., Zhou, J., & Leung, H. (2020, June). Sensemood: Depression detection on social media. In *Proceedings of the 2020 International Conference on Multimedia Retrieval* (pp. 407-411).

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Losada, D. E., Crestani, F., & Parapar, J. (2017, September). eRISK 2017: CLEF lab on early risk prediction on the internet: experimental foundations. In *International Conference of the Cross-Language Evaluation Forum for European Languages* (pp. 346-360). Springer, Cham.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Morales, M. R., & Levitan, R. (2016, December). Speech vs. text: A comparative analysis of features for depression detection systems. In 2*016 IEEE spoken language technology workshop (SLT)* (pp. 136-143). IEEE.

Moreo, A., Romero, M., Castro, J. L., & Zurita, J. M. (2012). Lexicon-based comments-oriented news sentiment analyzer system. *Expert Systems with Applications, 39*(10), 9166-9180.

Morgan, J. N., & Sonquist, J. A. (1963). Problems in the analysis of survey data, and a proposal. *Journal of the American statistical association, 58*(302), 415-434.

Nasir, M., Jati, A., Shivakumar, P. G., Nallan Chakravarthula, S., & Georgiou, P. (2016, October). Multimodal and multiresolution depression detection from speech and facial landmark features. In *Proceedings of the 6th international workshop on audio/visual emotion challenge* (pp. 43-50).

Nguyen, T., Phung, D., Dao, B., Venkatesh, S., & Berk, M. (2014). Affective and content analysis of online depression communities. *IEEE Transactions on Affective Computing, 5*(3), 217-226.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems, 32*.

Paul, S., Jandhyala, S. K., & Basu, T. (2018, August). Early Detection of Signs of Anorexia and Depression Over Social Media using Effective Machine Learning Frameworks. *In CLEF (Working notes)*.

Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research, 12*, 2825-2830.

Pirina, I., & Çöltekin, Ç. (2018, October). Identifying depression on reddit: The effect of training data. In *Proceedings of the 2018 EMNLP Workshop SMM4H: The 3rd Social Media Mining for Health Applications Workshop & Shared Task* (pp. 9-12).

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*.

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., & Huang, X. (2020). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences, 63*(10), 1872-1897.

Razak, C. S. A., Zulkarnain, M. A., Hamid, S. H. A., Anuar, N. B., Jali, M. Z., & Meon, H. (2020). Tweep: a system development to detect depression in twitter posts. In *Computational Science and Technology* (pp. 543-552). Springer, Singapore.

Reece, A. G., Reagan, A. J., Lix, K. L., Dodds, P. S., Danforth, C. M., & Langer, E. J. (2017). Forecasting the onset and course of mental illness with Twitter data. *Scientific reports, 7*(1), 1-11.

Reece, A. G., & Danforth, C. M. (2017). Instagram photos reveal predictive markers of depression. *EPJ Data Science, 6*(1), 15.

Resnik, P., Armstrong, W., Claudino, L., Nguyen, T., Nguyen, V. A., & Boyd-Graber, J. (2015). Beyond LDA: exploring supervised topic modeling for depression-related language in Twitter. In *Proceedings of the 2nd workshop on computational linguistics and clinical psychology: from linguistic signal to clinical reality* (pp. 99-107).

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review, 65*(6), 386.

Ruder, S., Peters, M. E., Swayamdipta, S., & Wolf, T. (2019, June). Transfer learning in natural language processing. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Tutorials* (pp. 15-18).

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning internal representations by error propagation*. California Univ San Diego La Jolla Inst for Cognitive Science.

Santos, A., Canuto, A., & Neto, A. F. (2011). A Comparative Analysis of Classification Methods to Multi-label Tasks in Different Application Domains. *International Journal of Computer Information Systems and Industrial Management Applications*, *3*, 218–227.

Stankevich, M., Isakov, V., Devyatkin, D., & Smirnov, I. V. (2018). Feature Engineering for Depression Detection in Social Media. In *ICPRAM* (pp. 426-431).

Siurob Palomero, T. A. (2019, July). Textos de usuarios con depresión: atributos que los representan y posible detección.

Tadesse, M. M., Lin, H., Xu, B., & Yang, L. (2019). Detection of depression-related posts in reddit social media forum. *IEEE Access, 7*, 44883-44893.

Torrey, L., & Shavlik, J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques* (pp. 242-264). IGI global.

Tsugawa, S., Kikuchi, Y., Kishino, F., Nakajima, K., Itoh, Y., & Ohsaki, H. (2015, April). Recognizing depression from twitter activity. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems* (pp. 3187-3196).

Tyshchenko, Y. (2018). Depression and anxiety detection from blog posts data. *Nature Precis. Sci., Inst. Comput. Sci., Univ. Tartu, Tartu, Estonia.*

Vapnik, V., & Lerner, A. (1963). Pattern recognition using generalized portrait method. Automation and remote control.

Villatoro Tello, E., Ramírez de la Rosa, G., & Jiménez Salazar, H. (2019). Detección anticipada de usuarios con depresión. *MADIC A cinco años de su creación: pasado, presente y futuro, 79* (pp. 79-94).

WHO - World Health Organization. (2021, September 13). *Depression*. Retrieved from: https://www.who.int/en/news-room/fact-sheets/detail/depression

WHO - World Health Organization. (2021, June 17). *Suicide*. Retrieved from: https://www.who.int/news-room/fact-sheets/detail/suicide

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Wolohan, J. T., Hiraga, M., Mukherjee, A., Sayyed, Z. A., & Millard, M. (2018, August). Detecting linguistic traces of depression in topic-restricted text: Attending to self-stigmatized depression with NLP. In *Proceedings of the first international workshop on language cognition and computational models* (pp. 11-21).

Yao, H., Rashidian, S., Dong, X., Duanmu, H., Rosenthal, R. N., & Wang, F. (2020). Detection of suicidality among opioid users on reddit: Machine learning–based approach. *Journal of medical internet research, 22*(11), e15293.

Appendix

Table 8: Experiments results of traditional machine learning with the original dataset

Model	Feature Extraction	Validation Macro F1-Score	Test Macro F1-Score
Logistic Regression	BoW	0.81	
	TF-IDF	0.79	
	BoW with N-grams	**0.85**	**0.51**
	TF-IDF with N-grams	0.80	
Decision Tree	BoW	0.81	0.41
	TF-IDF	0.80	
	BoW with N-grams	0.81	
	TF-IDF with N-grams	0.80	
Random Forest	BoW	0.81	0.41
	TF-IDF	0.81	
	BoW with N-grams	0.81	
	TF-IDF with N-grams	0.81	
XGBoost	BoW	0.82	
	TF-IDF	0.83	0.49
	BoW with N-grams	0.83	
	TF-IDF with N-grams	0.83	
Ada Boost	BoW	0.46	
	TF-IDF	0.54	0.4
	BoW with N-grams	0.53	
	TF-IDF with N-grams	0.53	
Multinomial Naïve Bayes	BoW	0.58	
	TF-IDF	0.59	
	BoW with N-grams	0.80	0.4

Model	Feature Extraction	Validation Macro F1-Score	Test Macro F1-Score
	TF-IDF with N-grams	0.80	
KNN	BoW	0.78	0.39
	TF-IDF	0.77	
	BoW with N-grams	0.78	
	TF-IDF with N-grams	0.78	
SVM with Linear Kernel	BoW	0.81	
	TF-IDF	0.82	
	BoW with N-grams	0.84	0.5
	TF-IDF with N-grams	0.83	
SVM with Radial Kernel	BoW	0.72	
	TF-IDF	0.82	0.49
	BoW with N-grams	0.74	
	TF-IDF with N-grams	0.81	
MLP	BoW	0.82	
	TF-IDF	0.83	
	BoW with N-grams	0.83	0.48
	TF-IDF with N-grams	0.82	

Table 9: Experiments results of traditional machine learning with undersampling

Model	Feature Extraction	Validation Macro F1-Score	Test Macro F1-Score
Logistic Regression	BoW	0.70	
	TF-IDF	0.67	
	BoW with N-grams	0.72	0.46
	TF-IDF with N-grams	0.70	
Decision Tree	BoW	0.69	0.41

Model	Feature Extraction	Validation Macro F1-Score	Test Macro F1-Score
	TF-IDF	0.69	
	BoW with N-grams	0.69	
	TF-IDF with N-grams	0.66	
Random Forest	BoW	0.75	
	TF-IDF	0.75	
	BoW with N-grams	0.76	
	TF-IDF with N-grams	**0.77**	**0.52**
XGBoost	BoW	0.70	
	TF-IDF	0.71	
	BoW with N-grams	0.71	
	TF-IDF with N-grams	0.72	0.44
Ada Boost	BoW	0.54	
	TF-IDF	0.53	
	BoW with N-grams	0.56	0.44
	TF-IDF with N-grams	0.55	
Multinomial Naïve Bayes	BoW	0.57	
	TF-IDF	0.58	
	BoW with N-grams	0.65	0.34
	TF-IDF with N-grams	0.64	
KNN	BoW	0.54	
	TF-IDF	0.57	0.36
	BoW with N-grams	0.52	
	TF-IDF with N-grams	0.53	
SVM with Linear Kernel	BoW	0.69	
	TF-IDF	0.69	
	BoW with N-grams	0.71	

Model	Feature Extraction	Validation Macro F1-Score	Test Macro F1-Score
	TF-IDF with N-grams	0.72	0.42
SVM with Radial Kernel	BoW	0.60	
	TF-IDF	**0.77**	0.49
	BoW with N-grams	0.63	
	TF-IDF with N-grams	0.76	
MLP	BoW	0.70	
	TF-IDF	0.71	
	BoW with N-grams	0.71	
	TF-IDF with N-grams	0.72	0.44

Table 10: Experiments results of deep learning with the original dataset

Model	Embedding Type	Embedding Size	Validation Macro F1-Score	Test Macro F1-Score
CNN	Trainable	50	0.77	
		100	0.81	
		200	0.77	
		300	0.80	
	Word2Vec	50	0.73	
		100	0.66	
		200	0.68	
		300	0.69	
	GloVe	50	0.80	
		100	0.79	
		200	**0.82**	**0.44**
		300	0.81	
LSTM	Trainable	50	0.27	

Model	Embedding Type	Embedding Size	Validation Macro F1-Score	Test Macro F1-Score
		100	0.27	
		200	0.27	
		300	0.27	
	Word2Vec	50	0.27	
		100	0.27	
		200	0.27	
		300	0.27	
	GloVe	50	0.27	
		100	0.27	
		200	0.27	
		300	0.27	
GRU	Trainable	50	0.27	
		100	0.27	
		200	0.27	
		300	0.27	
	Word2Vec	50	0.27	
		100	0.27	
		200	0.27	
		300	0.27	
	GloVe	50	0.27	
		100	0.27	
		200	0.27	
		300	0.27	

Table 11: Experiments results of deep learning with undersampling

Model	Embedding Type	Embedding Size	Validation Macro F1-Score	Test Macro F1-Score
CNN	Trainable	50	0.68	
		100	0.70	
		200	0.70	
		300	**0.72**	**0.45**
	Word2Vec	50	0.60	
		100	0.26	
		200	0.57	
		300	0.53	
	GloVe	50	0.65	
		100	0.63	
		200	0.67	
		300	0.71	
LSTM	Trainable	50	0.12	
		100	0.06	
		200	0.06	
		300	0.27	
	Word2Vec	50	0.07	
		100	0.06	
		200	0.15	
		300	0.15	
	GloVe	50	0.27	
		100	0.27	
		200	0.27	
		300	0.27	
GRU	Trainable	50	0.06	
		100	0.12	

Model	Embedding Type	Embedding Size	Validation Macro F1-Score	Test Macro F1-Score
		200	0.12	
		300	0.12	
	Word2Vec	50	0.16	
		100	0.10	
		200	0.15	
		300	0.06	
	GloVe	50	0.12	
		100	0.27	
		200	0.12	
		300	0.12	

Table 12: Experiments results of transfer learning with the original dataset

Model	Preprocessing	Validation Macro F1-Score	Test Macro F1-Score
roberta-base	Cleaning	0.83	
	No cleaning	0.86	0.53
twitter-roberta-base-mar2022	Cleaning	0.85	
	No cleaning	**0.87**	**0.54**

Table 13: Experiments results of transfer learning with undersampling

Model	Preprocessing	Validation Macro F1-Score	Test Macro F1-Score
roberta-base	Cleaning	0.73	**0.49**
	No cleaning	0.73	
twitter-roberta-base-mar2022	Cleaning	0.72	
	No cleaning	**0.75**	0.46

Table 14: Competition Ranking as of May 2022[11]

Shared Task on DepSign-LT-EDI ACL@2022						
TEAM WISE RESULTS						
Team Name	**Accuracy**	**Recall**	**Precision**	**Weighted F1-score**	**Macro F1-score**	**Rank (based on Macro F1 score)**
OPI	0.6582	0.5912	0.5860	0.6660	0.5830	1
NYCU_TWD	0.6330	0.5732	0.5394	0.6419	0.5523	2
ARGUABLY	0.6253	0.5720	0.5303	0.6333	0.5467	3
BERT 4EVER	0.6250	0.5806	0.5218	0.6318	0.5426	4
KADO	0.6179	0.5704	0.5263	0.6285	0.5422	5
UMUTeam	0.6250	0.5575	0.5248	0.6321	0.5382	6
DeepBlues	0.6515	0.5431	0.5374	0.6442	0.5374	7
Titowak	0.6706	0.5146	0.5710	0.6580	0.5356	8
E8@IJS	0.6015	0.5714	0.5149	0.6140	0.5334	9
SSN	0.6357	0.5334	0.5284	0.6380	0.5308	10
Ablimet	0.6228	0.5650	0.5118	0.6286	0.5299	11
Vishwaas	0.6089	0.5418	0.5132	0.6186	0.5236	12
sclab@cnu	0.6419	0.4947	0.5168	0.6300	0.5028	13
ai901@cnu	0.6123	0.5387	0.4901	0.6122	0.4957	14
Beast	0.5504	0.5690	0.4791	0.5693	0.4950	15
Unibuc_NLP	0.5686	0.5407	0.4688	0.5848	0.4862	16
BFCAI	0.6327	0.4739	0.4985	0.6220	0.4837	17
MUCS	0.6117	0.4969	0.4745	0.6096	0.4794	18
DepressionOne	0.6018	0.4733	0.4917	0.6061	0.4782	19
SSN_MLRG3	0.5729	0.5156	0.4576	0.5846	0.4726	20
niksss	0.5242	0.5567	0.4545	0.5454	0.4674	21
UAGD	0.5766	0.4685	0.4701	0.5859	0.4637	22
scubeMSEC	0.5106	0.5193	0.4610	0.5272	0.4571	23
kecsaiyans	0.5840	0.4605	0.4468	0.5855	0.4526	24
KUCST	0.5464	0.4728	0.4321	0.5623	0.4429	25
IISERB	0.5304	0.4811	0.4273	0.5499	0.4378	26
SSN_MLRG1	0.5846	0.4030	0.4362	0.5764	0.4117	27
KEC_Deepsign_ ACL2022	0.5692	0.3991	0.3984	0.5743	0.3982	28
RACAI	0.6709	0.3844	0.3952	0.6134	0.3721	29
GA	0.5131	0.3726	0.3648	0.5273	0.3635	30
FilipN	0.5858	0.3477	0.2564	0.5377	0.2914	31

[11] The competition raking can be found on:
https://drive.google.com/file/d/1LWr0VE0FYN9uGkfxxjQx0jlX5Nm0hhiw/view?usp=sharing