UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA BIOMÉDICA

STUDY OF DEPRESSION DETECTION IN ONLINE SOCIAL NETWORKS USING NATURAL LANGUAGE PROCESSING AND MACHINE LEARNING

AUTOR: JIMÉNEZ CAMPFENS, JOSE NÉSTOR

TUTOR: SELMO, CARLOS

COTUTORA: NARANJO ORNEDO, VALERIANA

Curso Académico: 2019-20

Contents

Abstract

As many studies and indicators suggest, the prevalence of depressive disorders is on the rise in the last years. Furthermore, the use of Online Social Networks has risen on a yearly basis since their appearance and the predictions seem to suggest this trend will go on for many years. These two facts seem to be forming a thriving niche of work in which the huge data produced by users online can be used to predict certain conditions, of which mental illness seems to be one of the most promising. With the use of Natural Language Processing techniques, Machine Learning algorithms can be used as a powerful tool in this goal.

This work makes an analysis of the reasons why such a tool could be useful, talking about concrete depressive disorders statistics and the use of online social networks. Then, it tries to explain how Natural Language Processing tools and Machine Learning Algorithms work and give an idea about their adequacy to the problem in question.

Finally, we make a detailed comparison of the different methods designed to perform the depression detection and explain the results obtained with the different combinations and a general idea of the reasons of these results.

Concretely, we were able to design some potent prediction algorithms (including *Multinomial Naive Bayes, Logistic Regression, Multi Layer Perceptron* and *Convolutional Neural Network* models) capable of correctly classifying the subjects between a depressed and control group using as input the texts posted on the famous social network *Reddit*, obtaining a 0.94 accuracy, 0.72 f1-score, 0.75 recall and 0.79 AUC-score.

Key-Words

Depression, Detection, Online Social Networks, Natual Language Processing, Machine Learning

1. *Introduction*

Within this chapter, and in order to outline the importance of the topic of this work, we will try and provide the reader with enough background information to be able to understand the context in which this research seats. With this aim, we will, first, try to describe the essential knowledge and building blocks needed to understand and deal with the problem of using Machine Learning algorithms to detect depressed subjects based on Social Networks publications.

1.1 Depression

1.1.1 Concept

According to the Medical Subject Heading (MeSH) definition introduced in 1981 by the National Center for Biotechnology Information (NCBI) of the U.S. National Library of Medicine, a depressive disorder is a prominent and relatively persistent affective disorder consisting of either a dysphoric mood or loss of interest or pleasure in usual activities [1].

Depending on the severity of symptoms shown by depressive patients, depression can be graded as mild, moderate or severe; if the lapse of time the disorder appears is taken into account, it can be considered an accute or chronic disorder; and a distinction is made between patients with present or absent manic episodes. As well, a distinction is made between recurrent depressive and bipolar affective disorders, when repeated depressive episodes appear in the patient where a minimum 2 weeks period of depressive mood is suffered by the subject in the former type, and when both manic and depressive periods appear separated by normal mood periods in the latter type [2].

1.1.2 Statistics

It has been studied that psychiatric disorders are the leading causes of disability worldwide, accounting for 37% of all healthy life years lost through disease. They are the most disabling disorders even in low-income and middle-income countries [3]. Within the group of mental illnesses, depression can be considered very common, with more than 264 million people affected worldwide [4]. Furthermore, it has been studied that the prevalence of depression increased in a significant way in the United States from 2005 to 2015, especially among young individuals when compared with the rest of the population [5].

1.2 Online Social Networks

1.2.1 Concept

As defined in [6], social network sites are web based services which give users of the service the ability to:

1. Construct a public or semi-public profile within the site system.
2. Create a list of other users with whom a connection exists (based on features like a shared interest, political view or activity).
3. Check their list of connections and those made by different users of the network.

It has been stated that the uniqueness of such Online Social Networks is that they allow the user to show or make their social networks visible, rather than the ability to meet or communicate with strangers, as the connections made within the site are normally already present offline [7].

1.2.2 Statistics

With regard to the use of Online Social Networks among the population, by looking at Table 1.1 we can get an idea of the quantities managed by such services by looking at the number of monthly users and its age distribution of Facebook, Twitter and Reddit (three of the largest currently offered services) in the United States during the year 2019.

	U.S. adults using social networks				**U.S monthly users (millions)**
Age	18-29	30-49	50-64	65+	
Facebook	79%	79%	68%	40%	169.76
Twitter	38%	26%	17%	7%	81.47
Reddit	22%	14%	6%	1%	47.87

Table 1.1: Percentage of surveyed subjects using one of the selected Social Networks as of Februrary, 2019 [8][9][10], and total number of monthly users as of September, 2019 [11].

1.3 Machine-Learning

1.3.1 Concept

If we attend the definition of an expert, we should quote Ian Goodfellow et al., when describing Arificial Intelligence [12]:

> *The true challenge to artificial intelligence proved to be solving the tasks that are easy for people to perform but hard for people to describe formally—problems that we solve intuitively, that feel automatic, like recognizing spoken words or faces in images.*

Attending to other expert definitions, we also see fit to quote Tom Mitchell's definition of Machine Learning stated in his book with same name [13]:

> *The field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience.* And the formalism: *A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.*

1.3.2 Classification

Supervised Learning

Supervised Learning consists of finding the model that best describes some data in order to predict an output. In other words, to find the function for a given input that makes the best estimation of an output. The main characteristic of Supervised Learning is that we know the ground-truth output beforehand and that we are able to assess and evaluate the efficacy of the method.
Some algorithms that use this kind of approach are Naive Bayes, Decision Trees and Neural Networks.
The main uses of this kind of approach consist of Classification and Regression problems.

Unsupervised Learning

Unsupervised Learning consists of finding descriptive patterns in unlabelled data (where a specific classification is not available). This kind of algorithms try to find patterns and relationships in the data and model them in order to be applied to new data later.
A classic unsupervised learning algorithm is the k-means clustering method, which uses the distance in different dimensions to find clusters in data.
The main uses of this kind of approach consist of Clustering and Association problems.

Semi-Supervised Learning

Semi-Supervised Learning uses a combination of both of the methods above mentioned. In this type of algorithms both labelled and unlabelled data is used, with the assumption that unlabelled data carries important information about the group parameters although its labels are unknown.

Reinforcement Learning

Reinforcement Learning algorithms are designed with the goal of learning the best actions possible in order to attain certain goal. The general idea of these algorithms is that it rewards the system when it implements actions that achieve a better result while it penalizes actions that achieve a worse result, thus reinforcing its learning.

1.3.3 Deep Learning

Based on the work of Yann Lecun, et al., covered in [14].

Deep Learning allows computational models to learn representation of data with a high level of abstraction, suppressing the need of using hand-engineered features to describe the data. Normally using Neural Networks in the form of connected stacked layers of neurons that extract information from the data, Deep Learning algorithms use the backpropagation algorithm to best fit the parameters of the layers to discover the intricate structure of large datasets. This is achieved by minimizing a specific cost function.

Multi-Layer Perceptron (MLP)

For explanation purposes, we will first try to define how an MLP network works as the simplest form of Neural Network (see Fig. 1.1).

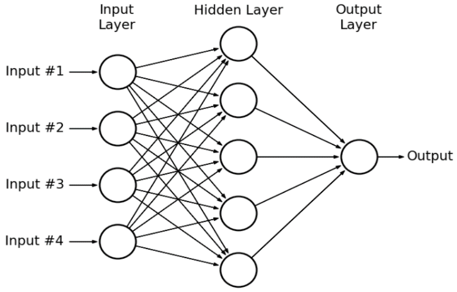


Figure 1.1: Basic architecture of a Multi Layer Perceptron net (obtained from [15])

In this type of algorithm, the input data is fed to the network through its input layer, where the data is weighted (multiplied by weight of the input layer) and passed to the subsequent layer where the sum of the weighted inputs is activated (passed through an activation function). This process is repeated in the following layers until the data comes out the output layer, where the activation corresponds to the prediction. An error is then calculated comparing the prediction with the Ground-Truth value of the data. Finally, the gradient of the cost function with respect to each set of weights is calculated in order to update the weights (using backpropagation), with the aim of decreasing the calculated cost. This process is then repeated, recursively adjusting the weights to the input data.

Activation Function

These are non-linear functions applied to the input data to each layer. The end of this function is to represent non-linearities of the data to best represent its structure. In the group of activation functions we can find the *Sigmoid* function, the *Hyperbolic Tangent (tanh)* function, the *Rectified Linear Unit (ReLU)* function, and the *Softmax* function, between others.
In Table 1.2 we introduce some examples of activation functions in order to get an idea on how they modify the input to the function.

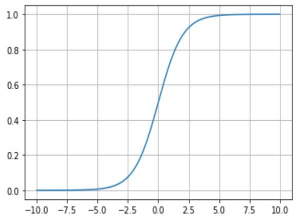
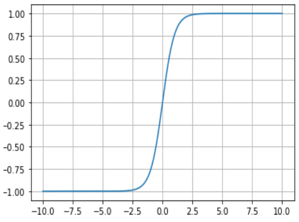
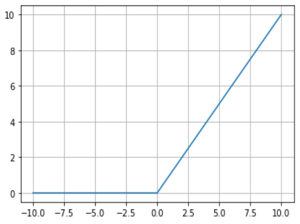
Sigmoid	**Tanh**	**ReLU**
$f(x) = \frac{1}{1+e^{-x}}$	$f(x) = tanh(x)$	$f(x) = max(0, x)$
		
Scales the input between 0 and 1.	Scales the input between -1 and 1.	Identity for positive values and 0 for negative values.

Table 1.2: Activation Functions

Cost Function

Cost functions are used to assess the accuracy or error of a neural net by comparing the output of the model to the ground-truth output expected. This function is used during training the model to follow the progress of this stage and fit the parameters with the backpropagation algorithm, and during testing to assess how good the model responds to new data. In the group of cost functions we can find the *Mean Absolute Error (MAE)*, *Mean Squared Error (MSE)*,

Cross-Entropy (CE) and *Cosine Proximity*, between others.
In Table 1.3 we introduce some examples of cost functions in order to get an idea on they work.

Mean Absolute Error:	$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} \lvert y_i - \hat{y}_i \rvert$	i = index of sample
		$\hat{y}$ = predicted output
		y = real output
		n = length of dataset

Mean Square Error:	$MSE(y, \hat{y}) = (\frac{1}{n}) \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$	
Cross Entropy:	$CE(y, \hat{y}) = - \sum_{i=1}^{n} y_i * log\hat{y}_i$	

Table 1.3: Cost Functions

Gradient Descent and Backpropagation

In supervised learning, in order to train the model, the error of the output with respect to the ground-truth is calculated. Then, the gradient of the cost function with respect to each weight is calculated. With this gradient, the values of the weights are updated by subtracting the value of the gradient of the cost function with respect to that particular weight multiplied by a factor (known as the learning rate). This optimization algorithm known as Gradient Descent is recursively applied with new data to fit the weights of the model following the equation 1.1:

$$\omega^{t+1} = \theta^t - \alpha * \frac{\delta C(X, \omega^t)}{\delta \omega} \tag{1.1}$$

, where X represents the input, ω^t denotes the weights of the model at iteration t, C denotes the cost function (which depends of X and ω), and α is the learning rate hyperparameter.

The name of the method comes from the fact that the first gradient to be calculated is that of the last (or output) layer of weights, propagating the error backwards through the net, to end calculating the gradient of the first layer of weights.

Special types of Neural Networks

The MLP-type of network represented in 1.1 can be very useful to solve certain problems but has one main flaw: it treats each input item as independent. Whereas in some cases this approach may be very useful, in some cases, the data to work with has a particular structure which can be dealt with using a different approach. Although there may be more particular types of Neural Networks, in this section we will introduce Convolutional Neural Networks and Recurrent

Neural Networks, which are used to work with datasets with a spatial connection (e.g. images), and with a temporal connection (e.g. speech), respectively.

Convolutional Neural Networks (CNNs). This type of networks use spatial information or patrons from the data to extract characteristics. It does so by recurrently applying the convolution operation of kernels over the input (along with pooling and activation operations) and use an algorithm to adapt the parameters of these kernels for the specific task of interest. These kernels can be flat when dealing with one-dimensional data like an Electro Cardiogram (ECG) signal, two-dimensional when dealing with 2D data like images or three-dimensional when dealing with volumetric data such as Magnetic Resonance Images (MRI).
This kind of neural nets can be used for classification or segmentation of the input data.

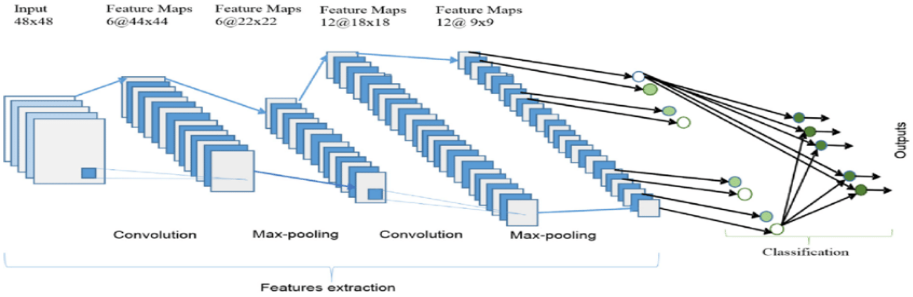


Figure 1.2: Basic architecture of a Convolutional Neural Network (obtained from [16])

Recurrent Neural Networks (RNNs). This approach is used to deal with temporal series of data, where one point of the data is normally related to the ones in its vicinity in a temporal dimension. This type of networks are somewhat like normal feed-forward networks in which an internal state (or memory) is added to hold sequential information of the data. This gives the advantage of assuming each sample to be dependant on previous samples of the data.
Some of the uses of this kind of networks is in regression of temporal data such as trajectories, speech recognition or processing of natural language.

1.4 Natural Language Processing

1.4.1 Concept

Natural Language Processing (NLP) is an area of study including the understanding of human spoken or written language. It concerns the area of research and application that explores how language can be understood and manipulated

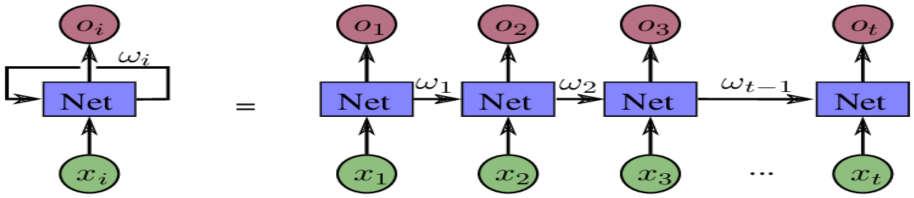


Figure 1.3: Basic architecture of a Recurrent Neural Network (obtained from [17])

automatically by computers [18].
The use of Natural Language Processing has been proved useful in several tasks including chat-bots, speech recognition and machine translation.

1.4.2 Techniques

Based on Frequency of Appearance

This kind of representations tell the number of occurrences of words within a text.

Bag of Words (BOW) This approach counts the number of different words in a text and saves the frequency of each of the vocabulary words into a vector. The problem with this approach appears when new words appear which are not in the initial vocabulary, as they will not have a representation in the encoded vector.

Term Frequency-Inverse Document Frequency (TF-IDF) This approach has the aim of statistically representing the importance of a word with respect to a document in a group of documents. The TF-IDF score of a word w in a document d from de set of documents D, with size N is calculated by equation 1.2, ponderating the frequency of a word within a document (*tf*) and the frequency of a word within all documents (*idf*):

$$tfidf(w,d,D) = tf(w,d) * idf(w,D) \tag{1.2}$$

where:

$$tf(w,d) = log(1+freq(w,d)) \qquad idf(w,D) = log(\frac{N}{count(d \in D: w \in d})$$

N-Grams This technique uses the frequency of groups of words that appear in a sequence. The BOW technique could be also be named a 1-gram or unigram. This technique can be used to find the most common expressions in a text or group of texts.

Based on Context

A more ”natural” way to approach language understanding is by looking at the surroundings (context) of the word to be encoded to see its relations with it.

> **Word Embedding** This technique allows machine learning algorithms understand words with similar meaning. Word embedding encodes words into vectors similar to vectors of words used in the same context. This means that the word *red* will have a similar vector representation as the word *green* as both of them represent a color. In a different way, the word *red*'s vector will be closer to the vector of the word *tomato* than to the vector of the word *cucumber*, as it describes a principal characteristic of it.
> Some of the most popular tools of word embeddings are *Word2Vec* [19], a shallow neural network algorithm to learn word embeddings, *fastText* [20] and *GloVe* [21].
>
> **Document Embedding** A similar approach as word embedding can be followed to encode full texts. Whereas Word2Vec learns to project words into a n-dimensional output space, Paragraph Vector (or Doc2Vec) does the same with whole paragraphs or texts.

2. *State of the Art*

The study of mental health through the analysis of language has been a topic of interest since long ago. In 1901, Freud wrote about the detection of feelings of literary authors through grammatical mistakes in their writings and books [22]. In the same line, Stirman and Pennebaker studied the language used by suicidal and non-suicidal poets in their literary work [23], proving a more common use of singular first-person pronouns in the first group of artists. In another study, researchers centered on the analysis of language of depressed, non-depressed and former-depressed subjects [24], finding that members of the depressed group were more prone to negative emotion words.

With the appearance of the Internet and Online Social Networks, new thriving challenges appeared in research in the field of language and mental health. In classic psychiatric studies researchers had to individually contact the subjects, making the task of collecting data very tedious. In this line, Online Social Networks proved to be a useful data source to researchers, especially using their APIs (Application Programming Interfaces) to make targeted searches over the entire site's data. Using these tools to search for user publications including words or sentences of interest (e.g. "I was diagnosed with depression"), big study datasets could be created, whereas, before this tools existed, generation of this kind of datasets could be a huge headache.
Within this perspective and using Twitter as data source, De Choudry et al. [25] used posts to create and train models capable of identify mothers at risk of post-partum depression; and Tsugawa et al. [26] used posts to obtain features from the activity on the social network to detect the degree of depression of Japanese users, obtaining an accuracy of 69%.

Using Facebook as data source, several studies have been made. Eichstaedt et al. [27] obtained a 72% accuracy predicting depression using posts from 6 months immediately prior to the clinical diagnosis in a medical center, extracting 200 topics using Latent Dirichlet Allocaton (LDA). Straton et al. [28] used Facebook posts labeled by experts in order to train a model able to predict stigmatized behaviour in users participating in vaccination discussions on the site; they trained several model architectures of which fastText, Bilateral-LSTM and CNN outperformed the rest of approaches used.

Another source of data in the same line is Reddit, an anonymous social network divided into communities (as they officially name the different themed forums in the site) or most commonly known "subreddits". The fact of anonymity encourages users to talk openly about delicate issues like mental health and depression. This advantage is especially patent when looking at the fact that there exists a community specifically focused on depression [29]. In this regard, Bagroy et al. [30] built a classification tool with 97% accuracy to detect mental health expressions using student posts on Reddit (concerning more than 100 universities). They as well revealed some interesting temporal patterns of mental health, finding that depression increases over the course of the academic year.

One main drawback of working with Twitter and Facebook data is that these sites do not allow redistribution of its data. This issue makes it difficult to compare different methods used as the data to work with will not be the same in all cases. This fact erases the possibility of creating publicly available datasets. Here is where Reddit data presents its main advantage over the rest of Social Networks. Within this framework Losada and Crestani [31] created a publicly available dataset consisting of Reddit posts classified by diagnosed and non-diagnosed with depression users. This dataset is used to train and test the models presented to the Conference and Labs of Evaluation Forum for Early Risk Predicition (CLEF eRISK), a public competition that encourages scientists from different fields to show their contribution to early detection of mental health issues. This competiton has been held on an anual basis continuously since 2017 with the title "Early Risk Prediction on the Internet".
Within the scope of this competition, many research centers have developed Machine Learning algorithms to predict the risk of depression as well as including a measure introduced by Losada and Crestani [31] called *Early Risk Detection Error* (ERDE), which measures how soon the model is able to correctly predict the risk of depression.
In the 2018 edition, Paul et al. [32] compared the efficacy of different methods from which the Adaptative Boosting (AdaBoost) and Support Vector Machine (SVM) classifiers using Bag of Words model outperformed other techniques.
In the same edition, Trotzek et al. [33] studied the application of CNNs between others,to the same problem, from which a Logistic Regression method based on metadata features obtained the highest F1-score.

3. *Scope of Study*

General

On one side, we face the fact that Online Social Networks are currently been used by hundreds of millions of users all across the world on a daily basis, and with the perspective that the use of these is constantly increasing, where users post their thoughts and feelings and that these sites are mainly used by young subjects; on another side we face the fact that psychiatric disorders are the leading causes of disability worldwide, especially among young individuals in comparison with the rest of the population; besides, in recent years a considerable amount of new techniques involving Machine Learning have been developed to try to make computers learn how humans communicate and the inherent structure of Language, specifically Natural Language Processing techniques.

With this in mind, this work studies the application of Machine Learning Algorithms and Natural Language Techniques to detect mental health disorders, concretely depression, using Online Social Network gathered data.

Specific

Exploratory Data Analysis We will first explain and explore the data selected for the study. Then, from a macro perspective, we will describe how the data is classified between the two different groups. Last from a more detailed perspective, we will describe some features extracted that could be of interest for our study.

Prediction We will study different approaches and study different models to try and solve the problem of detecting the presence of depression from the gathered data.

Comparison We will compare the performance of the different approaches used in this study and compare it to what other authors have achieved with the same data.

4. *Methods and Materials*

4.1 Data

4.1.1 Acquisition

As every Machine Learning algorithms developer knows, the first step in this kind of projects is the data gathering phase. In order to train a model and assure it is performing the task it was designed for, we should make sure the data to work with is optimally selected.

First of all, we need to define the characteristics of the data we need to fulfill the task object of this study: We need a dataset composed of textual data posted in Online Social Networks, labelled and classified between texts written by depressed users and texts written by non-depressed users.

For this task, different approaches could be followed, of which we are going to mention three to, after, explain the advantages and disadvantages of each one and our final decision:

1. Make an agreement with one or various mental health facilities or clinics and confirmed depressed patients of such clinics to gather texts posted on their Online Social Network feeds. Then gather posts of random non-depressed users to complete the dataset.
2. Use the Online Social Networks Application Programming Interfaces (APIs) to make a personalized query inferring the classification between depressed and normal subjects by modifying the parameters with which the search is made.
3. Search for an existing well-known published dataset which can fulfill our criteria with small or no changes.

The two metrics by which the data adequacy can be measured is fitness to represent the problem (i.e. the data really represents the problem object of study) and availability (i.e. possibility of gathering a dataset of considerable size).

Approach 1 would be fit to assure the data's adequacy to the problem. By using data of diagnosed patients gathered from the source of such diagnostic (the clinic) we can assure a correct distinction of the data between depressed and non-depressed subjects. On the other hand, this task does offer two main disadvantages: the amount gathered is going to be restricted to the number of patients within the clinic which fulfill the criteria, which possibly avoids the data to have a valuable and representative size; this approach can be very time-consuming as complicated agreements concerning data privacy should be signed with the clinics and patients.

Approach 2 has opposite implications in comparison to option 1. On one side, it presents the advantage of gathering a big dataset as it will get all the publications of users that match our search criteria. On the other side, we cannot assure the adequacy of the data to the problem in question as we cannot know the concrete diagnostic of every user. Although we can fine-tune our search criteria, we will never be 100% sure of the fitness of it.

Approach 3 could be considered a middle point between the two metrics proposed. On one hand, it exploits the option of gathering a big dataset using the Online Social Networks APIs and, as a consequence of being a well-known dataset, being tested by several developers for similar tasks through the years with good results, we can suppose this dataset to be representative of the problem object of study.

Taking all this into account we decided to use the Losada and Crestani dataset gathered in [31] which serves as base for the CLEF eRISK conference. This dataset consists of posts retrieved from Reddit users using the site's API. The dataset consists of posts of depressed users extended with a set of control subjects. The criteria to build this dataset was the following:

Depressed group Posts with self-expressions of depression diagnoses running searches through the Reddit API (e.g. "I was diagnosed with depression") to after manually reviewing the posts to verify the presence of such expressions. Only expressions with a clear mention to a diagnostic were included. With this aim, expressions of subjective opinions about self depression were not included (e.g. "I have depression", "I am depressed"). Older posts of targeted users were included and the post were the reference to the diagnostic was made was deleted. For each user a maximum of 1000 posts and 1000 comments were retrieved (due to Reddit's API limits), consequently collecting a maximum of 2000 items for the most active users.

Control group Posts of random users of Reddit were included into the control set, extracting them from various *subreddits*. Furthermore, active users in the depression *subreddit* were included, where posts usually talk about depression and mental health related issues. For this reason, the authors claim that we cannot rule out the possibility of accidentally including depressed subjects into the control group which have not clearly stated they had a depression diagnostic.

4.1.2 Exploratory analysis

The first thing we should take note about the data is the columns in which it is sorted. These are: *Patient ID, Title* of the post, *Date* of submission of the post, *Text* posted and the *Label (Depressed or Control)*

If we look at the data from a descriptive perspective we can see that we encounter a total of 892 subjects from which 755 are included in the control group and 137 in the depression group. If we calculate some statistics about the data, we can encounter the results covered in table 4.1, extracted from [31].

Group	*Control*	*Depression*
N. of Subjects	755	137
N. of Posts	481,873	49,580
Avg. Posts/Subject	638.2	361.9
Avg. Words/Post	36.7	27.4

Table 4.1: Statistics of data

We can see the data is unbalanced between the depressed and control groups and the fraction in each froup can be explored in figure 4.1.

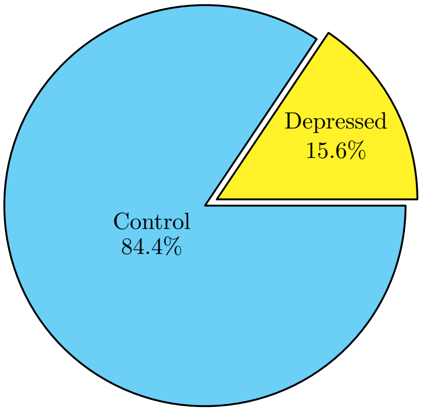


Figure 4.1: Distribution of data

If we take a deeper look into the data we can find the statistics covered in table 4.2, and find some interesting features which can be summarized in the following items:

- The percentage of negative users defining a title for their posts is more than double the percentage of depressed users.
- The ratio of posts including text is more than a 20% more common between depressed users than between users in the control group. This could

be explained by their higher tendency to posting comments (which lack a title) to other posts.

- The ratio of posts with both a title and a text nearly doubles between depressed users when compared to the control group.
- Non-depressed users tend to use less words in the text of the posts but more in the title of the posts, compared to depressed users.
- Depressed users have a higher tendency of using the word ”I” in their posts, almost doubling that of non-depressed users.

Group	*Depression*	*Control*
Posts with title (%)	13.79	32.61
Posts with text (%)	92.44	71.33
Posts with title and text (%)	6.25	3.96
Words/Title (mean)	11.77	13.91
Use of word ”I” (%)	3.91	2.32

Table 4.2: Statistics of data

4.1.3 Pre-processing

Input - Output format

Two different approaches have been used to address the classification problem. For the first approach, every post has been considered separately and independently from the rest of the posts (even from posts by the same user) and labeled with a depression or non-depression output. With this approach, the task of classification is centered into every single post. In the second approach, all posts from a same user have been considered as one large post, uniting the separate posts into one, and labeled together as depressed or non-depressed. In this way, the classification task is centered in the language and semantics used by every user in a global way along different periods of time.

Text Filtering

The texts have been filtered with the aim of normalizing the use of language between different users and posts. With this aim the texts have been put through different filtering stages consisting of the following operations:

1. **Tokenization:** Separate each word into a different token to be considered independently from its surroundings.
2. **Lower-Case Transformation:** Eliminate upper-case letters to treat same words as equal, without regarding the symbol used.

3. **Lemmatization and Stemming:** Reducing inflection in words to their stem (root).

4. **Non-Words Elimination:** Elimination of words containing non-alphabet symbols.

In this sense, two paths have been followed. The first using operations 1, 2, 3 and 4 (Pre-processing 1); and the second using operations 1, 2 and 4 (Pre-processing 2).
This step has been performed using the *Natural Language ToolKit (NLTK)*[34] package of *Python*.

Count Vectorization

A Bag of Words (BOW) is created and a vector is calculated with count of words for each post. In this stage, only words appearing in less than 80% posts and that appear at least 100 times are considered). A grid-search is performed to select the best option between considering only uni-grams or uni-grams (words) along with bi-grams (pairs of words), and to select the best option from applying TF-IDF transformation or not applying it.
This step has been performed using the *Scikit-learn (sklearn*[35] package of *Python*.

Text Embedding

In order to perform the embedding, an approach of embedding the entire text has been followed. For this step a *Document to Vector (Doc2Vec)* approach has been followed. In this regard, the selected hyper-parameters chosen to create the embedding were the following: *dm*=1, to use the *distributed memory (PV-DM)* training algorithm, an output vector length of *s*=300, *n*=5 in order to use negative sampling using 5 noise words, *hs*=0 in order to assure negative sampling, *min_count*=2 to only take into account words appearing at least this number of times, an initial learning rate of *alpha*=0.025 and a minimum learning rate of *alpha*=0.001. An embedding for each data representation (united or separated texts with pre-processing 1 or 2) has been trained during 30 epochs each.
This step has been performed using the *Gensim* [36] package of *Python*.

Training - Testing splitting

Finally the data corpus is shuffled and split into train and test sets, with 80% of the data going to the former set and 20% going to the latter.

4.2 Models

In this section we will list each of the methods used to predict whether a user is depressed based on the text posted on Reddit. Furthermore, we will briefly discuss the selection of the models for this particular task and the purpose and interest of using each kind of method.

4.2.1 Aleatory model

In order to evaluate the performance of each Machine Learning method, we first developed an algorithm outputting a an aleatory decision (depressed or non-depressed) randomly with 50% chance each value. With this we aim to create a reference to compare the rest of the methods.

4.2.2 Naive Bayes (NB)

This algorithm makes use of the Bayes Theorem which uses the prior probability to calculate the posterior probability of an event. In other words, it calculates the conditional probability of pertaining to a class given a feature (Posteriori) using the known distribution of the feature among each class (Priori). This probability can be calculated using the following equation:

$$P(Y|X) = \frac{P(Y)P(X|Y)}{P(X)} \tag{4.1}$$

where:

$$\textit{P(X) = probability of feature X}$$
$$\textit{P(Y) = probability of class Y}$$
$$\textit{P(X|Y) = probability of feature X given class Y (Priori)}$$

This algorithm is fit to the distribution of the feature in each class of a training set and to predict on new data, the class (Y) which maximizes P(Y—X). In other words, starting from a set of observations for each class, we can predict the class which maximizes the probability of a new observation.

Finally, when we consider more than one feature, we assume independence between features (therefore *Naive*) and the class predicted will be the one that maximizes the result of the following equation:

$$P(Y_j|X) = \frac{P(Y_j)\Pi_i P(X_i|Y_j)}{P(X)} \tag{4.2}$$

where:

$$X_i \textit{ = feature i, } Y_j \textit{ = class j}$$

In this case, a grid Search has been put in practice to select the best hyperparameters of the model from the following: alpha value of the model (10^{-2} and 10^{-3}).
This algorithm was designed using the *Scikit-learn (sklearn)*[35] package of *Python.*

4.2.3 Logistic Regression (LR)

This algorithm is used as a predictive model when the target (dependant variable) is categorical, hence the adequacy to our problem (depressed vs. non-depressed). This model is computed in 3 steps as follows:

$$\begin{gathered} z = X * w + b \\ \hat{y} = Sigmoid(z) = \frac{1}{1+e^{-z}} \\ LRmodel = MLE(y, \hat{y}) \end{gathered} \tag{4.3}$$

This model, as a linear regression, multiplies the inputs by a set of weights (w) and adds a bias (b) term to the result (z). Then, to maximize the separability of the two classes, the result is passed through the sigmoid function (eq. 4.3). Finally, the Maximum Likelihood Estimation (*MLE*) method is applied to fit the weight and bias values. A big advantage of this method is that it permits the treatment of unbalanced data by stating the class weights. In this case, a grid Search has been put in practice to select the best hyper-parameters of the model from the following: norm used in penalization (*l1* and *l2*) and the inverse regularization strength ($2^0, 2^1, 2^2, 2^3, 2^4$ and 2^5).
This algorithm was designed using the *Scikit-learn (sklearn)*[35] package of *Python*.

4.2.4 Multi-Layer Perceptron (MLP)

The first Deep Learning model used is based on a *Fully Connected Neural Network* (or MLP) with descending number of neurons. Starting with an input layer of the size of the input vector (300 when using the embedding approach), 4 hidden layers with 128, 64, 16 and 4 neurons, respectively, and an output layer of 1 neuron (which outputs a 0 or a 1, depending on the input).

We used the *ReLU* activation function (see table 1.2) for every fully connected layer except for the last one, for which we used the *Sigmoid* activation function (see table 1.2), to calculate the probability of each sample to belong to each class.

We initialised the batch size to β=256 in order to update the weight parameters of our model after forward propagating 256 samples through the net and calculating its error using the binary cross-entropy loss function (see table 1.3).

Furthermore, we applied the *Adam* optimiser [37] in order to update the network weights after forwarding the inputs of each batch and calculating the error.

4.2.5 Convolutional Neural Network (CNN)

The second Deep Learning model consisted of a Convolutional Neural Network, formed by the following architecture: an input layer of the size of the input vector (300 when using the embedding approach), followed by 4 hidden one-dimensional convolutional layers made of 16, 32, 64 and 128 filters of kernels with size 11, 7, 5 and 3, respectively with kernels of size 5, a *Global Max Pooling* layer and an output layer of 1 neuron (which outputs a 0 or a 1, depending on the input).

As in the MLP network, We used the *ReLU* activation function (see table 1.2) for every fully connected layer except for the last one, for which we used the *Sigmoid* activation function (see table 1.2), to calculate the probability of each sample to belong to each class.

Likewise, we initialised the batch size to β=256 in order to update the weight parameters of our model after forward propagating 256 samples through the net and calculating its error using the binary cross-entropy loss function (see table 1.3). And we applied the *Adam* optimiser [37] in order to update the network weights after forwarding the inputs of each batch and calculating the error.

4.2.6 Long Short-Term Memory Network (LSTM)

The third and last Deep Learning method used was a LSTM-type Recurrent Neural Network (see figure 1.3). This model was designed with the following architecture: an LSTM layer formed of 300 units (one for each item of the embedded input text data) connected to a dense output layer of 1 neuron (which outputs a 0 or a 1, depending on the input).

We used the *ReLU* activation function (see table 1.2) for the LSTM layer and the *Sigmoid* activation function (see table 1.2) in the output layer to calculate the probability of each sample to belong to each class.

Like in the previous DL models, We initialised the batch size to β=256 in order to update the weight parameters of our model after forward propagating 256 samples through the net and calculating its error using the binary cross-entropy loss function (see table 1.3). And we applied the *Adam* optimiser [37] in order to update the network weights after forwarding the inputs of each batch and calculating the error.

4.3 Work Pipeline

In figure 4.2, we can check the path followed throughout the work of this project, where the different steps (pre-processing, training and testing) and their timings are shown along with the different possibilities in each step:

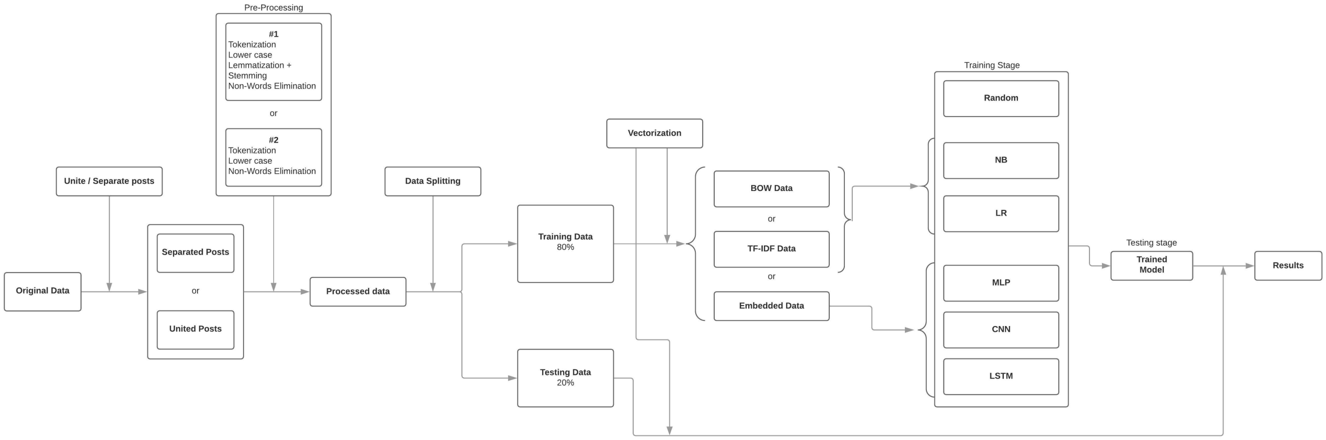


Figure 4.2: Project pipeline

5. *Results*

In this section, we will expose the general scores achieved by each model introduced in the *Methods and Materials* chapter, adding the results of the random model as a baseline to compare the results of each model.

The scores selected to measure the performance of the models with different pre-processing methods were the following:

- **Accuracy**: As the fraction of correct predictions from the whole testing set:

$$\frac{n_{\text{correct predictions}}}{\textit{length}(\text{test set})} \tag{5.1}$$

- **F1-score**: As the weighted harmonic mean of the model's precision and accuracy, calculated by equation:

$$\frac{\text{precision*recall}}{\text{precision+recall}} \tag{5.2}$$

- **Recall**: As the fraction of positive subjects, correctly classified as positive by the model:

$$\frac{\text{True Positives}}{\text{True Positives + False Negatives}} \tag{5.3}$$

- **Area Under the Curve (AUC)**: As the Area Under the *Receiver Operating Characteristic Curve (ROC)*, which represents the diagnostic ability of a binary classifier. The higher the AUC value, the higher this capacity, with a 1 representing perfect classification capacity, 0.5 representing null classification capacity and 0 representing perfect capacity to classify incorrectly.

Furthermore, the results of the Grid-Search performed in the different models when using vectorization are shown.

5.1 Vectorization

5.1.1 Separated texts + Pre-processing 1

For the Multinomial Naive Bayes model with a BOW Count Vectorizer, the Grid-Search yielded a value of *alpha* of the classifier of 10^{-2} and the use of unigrams in the vectorization.
For the Multinomial Naive Bayes model with a TF-IDF Vectorizer, the Grid-Search yielded a value of *alpha* of the classifier of 10^{-2} and the use of unigrams along with bigrams in the vectorization.
For the Logistic Regression model with a BOW Count Vectorizer, the Grid-Search yielded the use of unigrams along bigrams for the vectorizer and a *l2* penalty and a inverse regularisation strength C=2^2.
For the Logistic Regression model with a TF-IDF Vectorizer, the Grid-Search yielded the use of unigrams along bigrams for the vectorizer and a *l2* penalty and a inverse regularisation strength C=2^5.

Results are shown in table 5.1.

Model	*Accuracy*	*F1*	*Recall*	*AUC*
Random (50%)	0.50	0.19	0.51	0.50
MNB + BOW Vectorizer	0.85	0.29	0.26	0.6
MNB + TF-IDF Vectorizer	0.88	0.07	0.04	0.52
LR + BOW Vectorizer	0.88	0.07	0.04	0.52
LR + TF-IDF Vectorizer	0.88	0.01	0.01	0.5
MLP + BOW Vectorizer	0.88	0.16	0.09	0.54
MLP + TF-IDF Vectorizer	0.88	0.16	0.09	0.54

Table 5.1: Model scores with separated texts and pre-processing 1

5.1.2 Separated texts + Preprocessing 2

For the Multinomial Naive Bayes model with a BOW Count Vectorizer, the Grid-Search yielded a value of *alpha* of the classifier of 10^{-3} and the use of unigrams in the vectorization.
For the Multinomial Naive Bayes model with a TF-IDF Vectorizer, the Grid-Search yielded a value of *alpha* of the classifier of 10^{-2} and the use of unigrams along with bigrams in the vectorization.
For the Logistic Regression model with a BOW Count Vectorizer, the Grid-Search yielded the use of unigrams along bigrams for the vectorizer and a *l2*

penalty and a inverse regularisation strength C=2^5.
For the Logistic Regression model with a TF-IDF Vectorizer, the Grid-Search yielded the use of unigrams along bigrams for the vectorizer and a *l2* penalty and a inverse regularisation strength C=2^5.

Results are shown in table 5.2.

Model	*Accuracy*	*F1*	*Recall*	*AUC*
Random (50%)	0.50	0.19	0.51	0.50
MNB + BOW vectorizer	0.85	0.29	0.26	0.59
MNB + TF-IDF vectorizer	0.88	0.06	0.03	0.52
LR + BOW vectorizer	0.88	0.08	0.04	0.52
LR + TF-IDF vectorizer	0.88	0.02	0.01	0.5
MLP + BOW Vectorizer	0.88	0.18	0.1	0.55
MLP + TF-IDF Vectorizer	0.88	0.14	0.08	0.54

Table 5.2: Model scores with separated texts and pre-processing 2

5.1.3 United texts + Preprocessing 1

For the Multinomial Naive Bayes model with a BOW Count Vectorizer, the Grid-Search yielded a value of *alpha* of the classifier of 10^{-2} and the use of unigrams, along bigrams in the vectorization.
For the Multinomial Naive Bayes model with a TF-IDF Vectorizer, the Grid-Search yielded a value of *alpha* of the classifier of 10^{-2} and the use of unigrams along with bigrams in the vectorization.
For the Logistic Regression model with a BOW Count Vectorizer, the Grid-Search yielded the use of unigrams along bigrams for the vectorizer and a *l2* penalty and a inverse regularisation strength C=2^5.
For the Logistic Regression model with a TF-IDF Vectorizer, the Grid-Search yielded the use of unigrams along bigrams for the vectorizer and a *l2* penalty and a inverse regularisation strength C=2^5.

Results are shown in table 5.3.

Model	*Accuracy*	*F1*	*Recall*	*AUC*
Random (50%)	0.49	0.25	0.62	0.55
MNB + BOW vectorizer	0.78	0.48	**0.75**	0.77
MNB + TF-IDF vectorizer	**0.94**	**0.72**	0.58	**0.79**
LR + BOW vectorizer	0.91	0.57	0.42	0.7
LR + TF-IDF vectorizer	0.90	0.40	0.25	0.62
MLP + BOW Vectorizer	0.89	0.47	0.38	0.67
MLP + TF-IDF Vectorizer	0.86	0.00	0.00	0.50

Table 5.3: Model scores with united texts and pre-processing 1

5.1.4 United texts + Preprocessing 2

For the Multinomial Naive Bayes model with a BOW Count Vectorizer, the Grid-Search yielded a value of *alpha* of the classifier of 10^{-2} and the use of unigrams, along bigrams in the vectorization.
For the Multinomial Naive Bayes model with a TF-IDF Vectorizer, the Grid-Search yielded a value of *alpha* of the classifier of 10^{-2} and the use of unigrams along with bigrams in the vectorization.
For the Logistic Regression model with a BOW Count Vectorizer, the Grid-Search yielded the use of unigrams along bigrams for the vectorizer and a *l2* penalty and a inverse regularisation strength $C=2^4$.
For the Logistic Regression model with a TF-IDF Vectorizer, the Grid-Search yielded the use of unigrams for the vectorizer and a *l2* penalty and a inverse regularisation strength $C=2^5$.

Results are shown in table 5.4.

Model	Accuracy	F1	Recall	AUC
Random (50%)	0.49	0.25	0.62	0.55
MNB + BOW vectorizer	0.78	0.49	**0.75**	0.77
MNB + TF-IDF vectorizer	**0.94**	**0.72**	0.58	**0.79**
LR + BOW vectorizer	0.90	0.50	0.38	0.68
LR + TF-IDF vectorizer	0.90	0.40	0.25	0.62
MLP + BOW Vectorizer	0.83	0.51	0.62	0.75
MLP + TF-IDF Vectorizer	0.86	0.00	0.00	0.50

Table 5.4: Model scores with united texts and pre-processing 2

5.2 Embedding

Using the Doc2Vec tool for embedding along with the selected models and the text pre-processing techniques, the following results have been obtained:

5.2.1 Separated texts + Pre-processing 1

First, we embedded each post separately and pre-processed with stemming and lemmatizing.

Results are shown in table 5.5.

Model	*Accuracy*	*F1*	*Recall*	*AUC*
Random (50%)	0.50	0.19	0.51	0.50
LR	0.88	0.19	0.11	0.55
MLP	0.88	0.25	0.18	0.57
CNN	0.88	0.09	0.05	0.51
LSTM	0.88	0.00	0.00	0.50

Table 5.5: Model scores with separated texts, pre-processing 1 and embedding

5.2.2 Separated texts + Pre-processing 2

Secondly, we embedded each post separately and pre-processed without stemming and lemmatizing.

Results are shown in table 5.6.

Model	*Accuracy*	*F1*	*Recall*	*AUC*
Random (50%)	0.50	0.19	0.51	0.50
LR	0.88	0.2	0.13	0.55
MLP	0.87	0.24	0.17	0.57
CNN	0.88	0.017	0.01	0.50
LSTM	0.88	0.00	0.00	0.50

Table 5.6: Model scores with separated texts, pre-processing 2 and embedding

5.2.3 United texts + Pre-processing 1

Thirdly, we embedded all posts togheter and pre-processed the joint text with stemming and lemmatizing.

Results are shown in table 5.7.

Model	*Accuracy*	*F1*	*Recall*	*AUC*
Random (50%)	0.49	0.25	0.62	0.55
LR	0.89	0.54	0.46	0.71
MLP	0.90	0.54	0.42	0.70
CNN	0.90	0.54	0.42	0.70
LSTM	0.86	0.00	0.00	0.50

Table 5.7: Model scores with united texts, pre-processing 1 and embedding

5.2.4 United texts + Pre-processing 2

Last, we embedded all posts togheter and pre-processed the joint text without stemming and lemmatizing.

Results are shown in table 5.8.

Model	*Accuracy*	*F1*	*Recall*	*AUC*
Random (50%)	0.49	0.25	0.62	0.55
LR	0.89	0.5	0.42	0.69
MLP	0.89	0.42	0.29	0.64
CNN	0.87	0.26	0.17	0.57
LSTM	0.86	0.00	0.00	0.50

Table 5.8: Model scores with united texts, pre-processing 2 and embedding

6. *Discussion*

We will first compare the results obtained within this study by method of comparison of the results obtained with each pre-processing method and prediction model. We, then, will compare our results to other author's work.

6.1 Remark on results

In many cases, the accuracy obtained while testing achieves a value of close to 88%. At first sight it may be promising to find a model with this result. Nevertheless, if we take a deeper look into the data structure and partition (explained in chapter 4), we will see that the dataset is unbalanced having several more cases of non-depressed users than depressed. With this in mind, we decided to consider different measures which could better explain the reality of the predictions (i.e. f1-score, recall and AUC).

6.2 Results Discussion

6.2.1 Text strategy: Separated vs. United texts

With regard to the strategy of using the posts of a user combined in one or to use them separately from each other, we can undoubtedly affirm that it is highly preferable to unite them into one big text. This argument can be supported by looking at the highest scores obtained with each strategy, shown on table 6.1:

	Separated texts	*United Texts*
Accuracy	0.88	0.94
F1-score	0.29	0.72
Recall	0.26	0.75
AUC	0.6	0.79

Table 6.1: Best scores related to text strategy

Furthermore, if we compare the performance of the different strategies to the results obtained with the baseline model, we can find that in the *Separated*

Texts strategy the results obtained do not suppose a significant improvement to a Random approach, even obtaining some poorer results in some scores.

6.2.2 Pre-processing strategy: Use of Lemmatization and Stemming

Taking into account the using of pre-processing 1 (with Stemming and Lemmatization) or pre-processing 2 (without Stemming and Lemmatization), we can see that the results obtained do not suppose a significant difference between the two methods. In fact, the results obtained with these two processing strategies only differ in a couple decimals in the majority of the cases. The highest scores obtained with both methods are the same in all measures considered, as we can see in table 6.2:

	Pre-processing 1	*Pre-processing 2*
Accuracy	0.94	0.94
F1-score	0.72	0.72
Recall	0.75	0.75
AUC	0.79	0.79

Table 6.2: Best scores related to pre-processing strategy

This fact can induce the idea that the suppression of inflected language to standardize the texts does not suppose a significant change in the classification task.

6.2.3 Vectorization strategy: Count vs. Embedding

In general, we find that models that make use of Count Vectorization techniques (*Bag Of Words* and *TF-IDF*) throw better results than those based on the embedding technique (*Doc2Vec*). In fact, the two models used with both techniques (*LR* and *MLP*) always show better results with a Count Vectorizing approach than with the Embedding approach, independent of the rest of strategies.

At this point we should state that the possible reason why the Embedding strategy has not been successful could be that we trained an Embedding model from scratch with our data, which could be limited and could lack the ability to perform in an optimised manner. In this regard and with the aim of improving this strategy, other pre-trained embedding techniques could be used like *GloVE* [21] and *fastText* [20]. This approach could yield better results as they are trained on much larger datasets that can be fitter to our problem (i.e. the *GloVE* has a pretrained embedding model trained on 2 billion tweets). Nevertheless, this falls out of the scope of this study and should be addressed in posterior works.

6.2.4 Models comparison

In general, the best performing algorithm has been the Multinomial Naive Bayes model, obtaining the highest AUC score, regardless of the rest of strategies. Moreover, this model achieves the highest value in each measure, regardless of the rest of strategies.

Moreover, looking at the results obtained with the Deep Learning models (i.e. MLP, CNN and LSTM), we see that the MLP and CNN models have a reasonably good performance when used with embedding (even when the embedding does not perform efficiently). Whereas the use of LSTM has a very poor performance, predicting the majority class as output, thus yielding the bad results shown.

With special concern about the use of LSTM networks, it could be argued that, rather than using a *Document to Vector* embedding technique, a type of embedding that conserves the sequential information of the texts (i.e. a word embedding technique) would have been of a more logical use as a previous step to feeding LSTM neural networks, which exploit the sequential structure of the data. This fact shoud be addressed in future work with the aim of studying the performance of Recurrent Neural Networks to address the problem of this work.

6.2.5 Training time

In terms of text processing, the embedding took a considerably bigger amount of time to be performed when compared with the Count Vectorization techniques.

With regard to model training, the MLP and CNN models were, by far, the fastest to be trained. On the other hand, the LSTM model took the longest to be trained, followed by the MNB and LR models.

6.3 Related work

In this study, we found some interesting facts that correspond to the findings of other scientists within the same topic.
If we look at the exploratory analysis of the data, we found that the use of the first person pronoun "I" was used twice as more between depressed subjects, when compared with the control group. This is lined up with the findings of Stirman and Pennebaker [23], where they encountered a more common use of this kind of words between suicidal poets, in comparison with non-suicidal authors.

On one hand, With regard to the prediction task, in the *CLEF eRISK* edition of 2018 Paul et al. [32] found that the best performing algorithm to predict depressed users was a Logistic Regression model combined with a Bag of Words strategy. This fact, taking into account that the study was developed with the same type of data as our study, stands in line with our findings, where the best performing algorithms made use of a Count Vectorization strategy.

On the other hand, our findings contrast with the results obtained by Straton et al. [28], which found CNN and Bilateral-LSTM models to be outperforming the rest of the studied models. In this regard, we can make an assumption that the pre-processing stratgy followed was not optimal and more sofisticated ways of vectorizing the data should be considered.

7. *Conclusions*

As many studies and indicators suggest, the prevalence of depressive disorders is on the rise in the last years. Furthermore, the use of Online Social Networks has increased on a yearly basis since their appearance and the predictions seem to suggest this trend will go on for many years. These facts make Machine Learning algorithms, along NLP tools, very adequate in the task of predicting depressed users, making use of the huge amount of data the social networks produce.

With the results commented in the different subsections of the discussion we are able to select the best pathway to address the problem of depression detection in online social networks, based on text posts published by users. The paragraphs that follow are a set of recommendations based on the obtained results.

> In general, we can assume that a strategy based on the whole history of texts posted on a Social Networks united into one big text will contain more useful information in the task of depression detection. With this strategy, the results improve in all considered measures.
>
> The use of Lemmatization and Stemming in the data pre-processing stage does not contribute to a significant improvement in the classification task, and, in consequence, the elimination of inflected language in not a necessary step.
>
> With the data we worked with proposed in [31], the use of the *Document to Vector* embedding strategy trained from scratch does not suppose an improvement in comparison to classic Count Vectorization techniques (i.e. *Bag of Words* and *TF-IDF*). With the results of this study, we recommend the use of the latter strategy, although we affirm that trials with pre-trained embeddings (i.e. *GloVe* and *fastText*) should be studied in future works.
>
> In general the best performing algorithm was the Multinomial Naive Bayes, in all measures considered, when compared to the rest of models.

With all, the best performing algorithm concerning each of the considered measures and its hyper-parameters were the following:

The Multinomial Naive Bayes model applied to the user's united texts, with pre-processing 1 or 2, making use of unigrams and bigrams in the vectorization and *BOW* Vectorization strategy in order to maximise the Recall score, and *TF-IDF* Vectorization strategy in order to maximise the Accuracy, F1-score and AUC.

Bibliography

[1] National Center for Biotechnology Information U.S. National Library of Medicine. *Depressive dissorder*. 1981. URL: `https://www.ncbi.nlm.nih.gov/mesh/68003866`.

[2] World Health Organization. *Depression*. 2020. URL: `https://www.who.int/news-room/fact-sheets/detail/depression`.

[3] Philip S Wang et al. "Use of mental health services for anxiety, mood, and substance disorders in 17 countries in the WHO world mental health surveys". In: *The Lancet* 370.9590 (2007), pp. 841–850.

[4] Spencer L James et al. "Global, regional, and national incidence, prevalence, and years lived with disability for 354 diseases and injuries for 195 countries and territories, 1990–2017: a systematic analysis for the Global Burden of Disease Study 2017". In: *The Lancet* 392.10159 (2018), pp. 1789–1858.

[5] AH Weinberger et al. "Trends in depression prevalence in the USA from 2005 to 2015: widening disparities in vulnerable groups". In: *Psychological medicine* 48.8 (2018), pp. 1308–1315.

[6] Danah M Boyd and Nicole B Ellison. "Social network sites: Definition, history, and scholarship". In: *Journal of computer-mediated Communication* 13.1 (2007), pp. 210–230.

[7] Caroline Haythornthwaite. "Social networks and Internet connectivity effects". In: *Information, Community & Society* 8.2 (2005), pp. 125–147.

[8] Pew Research Center. *Percentage of U.S. Adults Who Use Facebook as of February 2019, by Age Group.* 2019. URL: `https://www.statista.com/statistics/246221/share-of-us-internet-users-who-use-facebook-by-age-group/`.

[9] Pew Research Center. *Percentage of U.S. Adults Who Use Twitter as of February 2019, by Age Group.* 2019. URL: `https://www.statista.com/statistics/265647/share-of-us-internet-users-who-use-twitter-by-age-group/`.

[10] Pew Research Center. *Percentage of U.S. Adults Who Use Reddit as of February 2019, by Age Group.* 2019. URL: `https://www.statista.com/statistics/261766/share-of-us-internet-users-who-use-reddit-by-age-group/`.

[11] Verto Analytics. *Most Popular Mobile Social Networking Apps in The United States as of September 2019, by Monthly Users (in Millions).* 2019. URL: `https://www.statista.com/statistics/248074/most-popular-us-social-networking-apps-ranked-by-audience/`.

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* `http://www.deeplearningbook.org`. MIT Press, 2016.

[13] T.M. Mitchell. *Machine Learning.* McGraw-Hill International Editions. McGraw-Hill, 1997. ISBN: 9780071154673. URL: `https://books.google.es/books?id=EoYBngEACAAJ`.

[14] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep Learning". In: *Nature* 521.7553 (2015), pp. 436–444. DOI: `10.1038/nature14539`. URL: `https://doi.org/10.1038/nature14539`.

[15] Hassan Hassan et al. "Assessment of artificial neural network for bathymetry estimation using high resolution satellite imagery in shallow lakes: case setudy El Burullus Lake." In: *International Water Technology Journal* 5 (Dec. 2015).

[16] Md. Zahangir Alom et al. "A State-of-the-Art Survey on Deep Learning Theory and Architectures". In: *Electronics* 8 (Mar. 2019), p. 292. DOI: `10.3390/electronics8030292`.

[17] Lei Tai and Ming Liu. *Deep-learning in Mobile Robotics - from Perception to Control Systems: A Survey on Why and Why not.* Dec. 2016.

[18] Gobinda G Chowdhury. "Natural language processing". In: *Annual review of information science and technology* 37.1 (2003), pp. 51–89.

[19] Tomas Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems.* 2013, pp. 3111–3119.

[20] Armand Joulin et al. "Bag of tricks for efficient text classification". In: *arXiv preprint arXiv:1607.01759* (2016).

[21] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP).* 2014, pp. 1532–1543.

[22] Sigmund Freud. *The Psychopathology of Everyday Life.* London, U.K.: Hogarth, 1901.

[23] Shannon Stirman and James Pennebaker. "Word Use in the Poetry of Suicidal and Nonsuicidal Poets". In: *Psychosomatic medicine* 63 (July 2001), pp. 517–22. DOI: `10.1097/00006842-200107000-00001`.

[24] Jörg Zinken et al. "Analysis of syntax and word use to predict successful participation in guided self-help for anxiety and depression". In: *Psychiatry research* 179 (Sept. 2010), pp. 181–6. DOI: 10.1016/j.psychres.2010.04.011.

[25] Munmun De Choudhury, Scott Counts, and Eric Horvitz. "Predicting postpartum changes in emotion and behavior via social media". In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2013, pp. 3267–3276.

[26] Sho Tsugawa et al. "Recognizing depression from twitter activity". In: *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. 2015, pp. 3187–3196.

[27] Johannes C Eichstaedt et al. "Facebook language predicts depression in medical records". In: *Proceedings of the National Academy of Sciences* 115.44 (2018), pp. 11203–11208.

[28] Nadiya Straton et al. "Predictive modelling of stigmatized behaviour in vaccination discussions on Facebook". In: *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE. 2019, pp. 2561–2568.

[29] *Depression, because nobody should be alone in a dark place*. https://www.reddit.com/r/depression/. [Online; accessed 10-04-2020].

[30] Shrey Bagroy, Ponnurangam Kumaraguru, and Munmun De Choudhury. "A Social Media Based Index of Mental Well-Being in College Campuses". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, 1634–1646. ISBN: 9781450346559. DOI: 10.1145/3025453.3025909. URL: https://doi.org/10.1145/3025453.3025909.

[31] David E Losada and Fabio Crestani. "A test collection for research on depression and language use". In: *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer. 2016, pp. 28–39.

[32] Sayanta Paul, Sree Kalyani Jandhyala, and Tanmay Basu. "Early Detection of Signs of Anorexia and Depression Over Social Media using Effective Machine Learning Frameworks." In: *CLEF (Working Notes)*. 2018.

[33] Marcel Trotzek, Sven Koitka, and Christoph M Friedrich. "Utilizing neural networks and linguistic metadata for early detection of depression indications in text sequences". In: *arXiv preprint arXiv:1804.07000* (2018).

[34] Edward Loper and Steven Bird. "NLTK: The Natural Language Toolkit". In: *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*. 2002.

[35] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python ". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[36] Radim Řehůřek and Petr Sojka. "Software Framework for Topic Modelling with Large Corpora". English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. http://is.muni.cz/publication/884893/en. Valletta, Malta: ELRA, May 2010, pp. 45–50.

[37] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).