Detecting Depression in Twitter's Users Using Text Classification Models

Gabriel Cesário Silva Martins
Departamento de Ciência da Computação
University of Brasília
Brasília, Brazil
gcesario031@gmail.com

***Abstract*—This document presents the exploration of NLP techniques for classifying tweet texts. More specifically, it uses machine learning models to classify tweet's authors as depressive or non-depressive. The models used in this task are the follow: LSTM, CNN and Logistic Regression. A review of the available literature is carried out, verifying which approaches were previously made in this specific task, to assess which are the best methods to be used. Checking the actual state-of-art in this area is also done. The data used contain 20000 tweets, which are analyzed and preprocessed, according to some criteria. A brief explanation of the approach adopted is made, as well as the resources and values chosen for each model. Cross-validation is also used, in the adjustment of a hyper-parameter. Also, the manual adjustment of some hyper-parameters is done. Results shown that the training process is successful. The classification for each model in the test set is presented, giving good results. All models performed similarly. The accuracy for the best model is shown: for the given data, its value was around 80%.**

***Index Terms*—NLP, Twitter, Depression, Machine Learning, Logistic Regression, RNN, CNN, LSTM, Random Forest, SVM.**

I. Introduction

In the current context, it is not novel that humanity faces a varied amount of problems, at most different scales. Wars, disease and violence are some of these present at regional levels or affect the entire globe. One problem, in particular, is depression. According to [McCarron et al. 2021], depression is "a mood disorder characterized by a persistent feeling of sadness and/or an inability to experience pleasure, with associated deficits in daily functioning". This disease can affect anyone, of any age, and is present in many countries around the world. As cited by [Marcelo Basso de Sousa 2013], the World Health Organization estimates that 10 to 25% of women will have the disease at some point in their lives, and in men, this rate is 5 to 12%. It is also known that, if left untreated, it can have serious consequences for the patient. Also according to [McCarron et al. 2021], it is one of the main factors that lead to suicide, being the motive for an increase in 35% in suicide rates in the United States. Knowing that it is serious, what are the factors that lead someone to have depression? Unfortunately, this is a difficult question to answer. According to [McCarron et al. 2021], the pathological causes that lead someone to have depression are unknown, and currently, there are no tests that make a correct diagnosis. Thus, for the correct detection of the disease, an individual assessment of the patient must be carried out by a trained professional.

Despite the barrier of correct diagnosis, alternatives have been developed to overcome this problem recently. One that can be highlighted is the analysis of texts from users on Twitter, to verify the possible presence of the disease. Briefly, Twitter is a social network for communication, which is used by millions of people all over the world. Tweets, according to [Twitter 2022], are messages containing text, photos, GIFs, or videos. Tweets are the primary form of communication used on Twitter. That said, the mentioned analysis is done on tweets of the network users, more specifically, on the tweets that contain texts, in order to predict whether the user has depression or not. This can be performed by predictive machine learning models, which do the prediction. Such analysis will be the focus of this work.

About this method, it has already been studied before, with very optimistic results. In previous work, varied models were developed to predict depression, using random forests [Reece et al. 2017], support vector machines [Hatoon S AlSagri e Mourad Ykhlef 2020], logistic regression [Lin et al. 2016], etc. In the cited works, the results showed that this technique is valid to detect users who have depression, with a good rate of success. Furthermore, analyses show that such a technique is a good and valid way to predict depressed users [Kim et al. 2021].

Since this is a good way to detect users with depressive disorder, the present work seeks to develop machine learning models to solve this task, in a way that helps professionals in the area in the diagnosis of the disease. More specifically, developing models not found in the literature for this task, new approaches are made, namely: LSTM network, or *Long Short-Term Memory*, and CNN, or *Convolutional Neural Network*. In addition to these, for comparison purposes, logistic regression is also adopted.

It is important to note that this method has limitations. First, only people who use Twitter are evaluated in the model, and furthermore, only the data of users who allow it can be used in the model (due to respect for their privacy) [Hatoon S AlSagri e Mourad Ykhlef 2020]. Other limitations are known but will not be described here. It is hoped that, with the development of this work, a faster and cheaper way to detect depression will be developed, in a way that helps professionals in the

area in its treatment.

In the next sections, possible models will be examined, in addition to the description of the data used and the training of a specific model. In the *Related Works* section, some works will be analyzed, in order to verify which models were developed. In the *Methodology* section, it is explained how the model was chosen, in addition to its training. In *Experimental Results*, the results obtained are presented, in addition to an analysis and discussion about them. Finally, in the *Conclusion* section, a recap of the work is made, explaining whether the results obtained were satisfactory or not.

II. Related Works

In order to implement a model that could resolve the task, a research was made in different sources. Some of the found works are presented here.

First, the work done on [Reece et al. 2017] is shown. Some models were developed to predict users with and without depression, as well as people who might have post-traumatic stress disorder. To do this, the authors collected data from 204 Twitter users, as well as their history of depression (for users who had depression). Among the data, about 279,951 tweets were collected. Similarly, the history of 243,775 tweets were collected from 174 users to predict post-traumatic stress disorder. With this, random forest models were developed to make the prediction of possible sick users, and these were applied to the data (i.e., trained). In their results, it was found that the models developed, after training, were able to predict with a high success rate, which users could have depression and which ones did not. Incredibly, the models were able to be more accurate than the doctors in terms of diagnosing the disease. Basically, by analyzing a user's tweets, the model was able to detect depression months before the actual diagnosis. Such results were similar with post-traumatic stress disorder. In addition, other models (Hidden Markov Models) developed in this work also had good results.

Similarly, in [Hatoon S AlSagri e Mourad Ykhlef 2020], to investigate users with depression, the collection of tweets was done, so that the data were used to train machine learning models, which in turn made the prediction. For this, some models commonly used in natural language processing were tested, notably, decision trees, support vector machines, and a Naive Bayes classifier. Briefly, to do the task, data preparation, feature extraction, and classification were done using various R libraries, as described by [Hatoon S AlSagri e Mourad Ykhlef 2020]. In addition, cross-validation was applied to the models. Of the results obtained, support vector machines were the best-performing model. The accuracy got was around 82%, with a recall of 0.85. With this, the authors conclude that their results were better than models already implemented previously (such models are highlighted in this work, but will not be shown here). In addition, it is noted that the use of tweets in conjunction with user activity is important to make a good prediction.

III. Proposed Method

In an attempt to solve the proposed problem, some steps were applied to prepare the data and train the models that would make the classification. Such measures are explained here.

A. The Data Used

The dataset used for the task was obtained from [Depression: Twitter Dataset + Feature Extraction]. Describing it, we have a "table" with 20000 rows. Each line describes the id, date, and text of a tweet. In addition, user data is present, such as id, the number of followers, friends, favorites, and status, the number of times the tweet was shared (retweet), and finally, a classification label. On the latter, there are two possible values: 0 for non-depressive and 1 for depressive.

1) Exploratory Data Analysis: After exploration, some data were obtained from the dataset. The 20000 tweets were collected from a total of 72 users. There is no presence of tweets from both classes for the same user. Knowing this, 54 users were classified as depressive, while the remaining 18 are non-depressive. Despite the discrepancy, interesting data shows an average of 185 tweets made by depressive users, while there are 556 tweets by non-depressed users. This shows that depressed users tweet less than expected. This fact was already observed in [Reece et al. 2017], being a good feature for classification.

Another piece of information is that the dataset is well distributed: half of the tweets are classified as depressive (i.e., tweets where the user is depressed, but for succinct writing, tweets are directly described), and the other half as non-depressive.

Finally, the tweets are not cleaned, i.e., they contain symbols and emotes commonly used in social networks, indications that the tweet is a retweet, links, etc.; Knowing this, a pre-processing of the text is necessary to move forward, which is done in the next section.

2) Data Pre-processing: At first, the classification of a tweet in retweet is made. This data was used in [Reece et al. 2017], and given that the trained models obtained good results, this feature will be preserved for the dataset of this work.

To do this classification, regular expressions were used to find the patterns within each tweet. It is expected that the result of this process to have high accuracy (close to 100%), given the characteristics of retweets.

With such a feature extracted, more pre-processing is done. For each tweet, text normalization is done, removing special characters and expressions. Retweet markers, links, emotes, punctuation characters, and mentions are taken out. There is also a transformation of emotes and language expressions (for example, in the phrase *R u okay?*, it could be expanded to *Are you okay?*) done. In addition, letters are transformed into lowercase form. Following, a tokenization process is also performed. For this, the use of the `nltk` package for python was employed, in order to complete the pre-processing, i.e.,

remove characters/symbols that are still left over from the previous step.

From this process, the removal of stop-words is made on the text, using the set of stop-words from the `nltk`. When done, first-person pronouns were kept, as they can influence the prediction of the models. For the next step, lemmatization with the `WordNetLemmatizer` is applied to text, in order to reduce variations from a word into a single one. In the end, empty tweets are removed, resulting in a dataset with 19929 tweets (total removal of 71).

3) Data Separation: The last step, before the use in the model, is the data separation. For this, the 19929 tweets from the dataset were separated into 3 subsets: validation, training, and testing, using functions from the `scikit-learn` package. The training set has 80%, or 15943 of the total, while the validation and test sets have 10% (or 1993 tweets) each.

B. Feature Extraction

Once the data is prepared, it is necessary to obtain a representation of the texts so that they can be submitted to the models, since they do not understand them directly. For this, some approaches were applied: *Term Frequency-Inverse Document Frequency* (TF-IDF) and *Sentence Embedding*, to map the texts into objects understandable by the models; and *One-Hot Encoding*, to map the classes of each tweet.

1) TF-IDF: This feature computes the frequency of a word in a document, and weights this measure by its presence throughout a dataset. According to [tf–idf 2022], is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.

In the dataset, its application was made through the use of the `TfidfVectorizer` class from the `scikit-learn` package, generating a sparse matrix with 16158 entries, for each tweet of the training set.

2) Sentence Embedding: In this feature, a given sentence is mapped to a vector of real numbers [Sentence embedding 2022].With its use, the semantic relationship between the words of a sentence is maintained in the conversion, something that does not happen in other representations (as in TF-IDF). Such a relationship is adjusted during the training of a model. Its use was employed in this work for this characteristic.

For its application, the vectorization of each text was made, so that the words of a sentence were mapped in indices. This can lead to vectors of different sizes, given the different length of the tweets. So to make a fixed size for all vectors, an estimation is made. The max tweet size in words is 27. Knowing this, all generated vectors, with a size smaller than that count, were padded with 0s. So, when submitted to the models, all tweet's vectors have 27 in length.

Then, these vectors are submitted to the model through classes from the `TensorFlow` package.

3) One-Hot Encoding: For the use of this feature, the generation of vectors of size n is done, for all tweets. Here, n represents the number of classes (in this work, $n = 2$). For an element of class i, the array at index i will be 1, and 0 in all other positions. Its use was employed in this work due to the characteristics of the models made. Here, the label for each tweet is transformed, given the explained process, generating the vectors.

C. Machine Learning Models

For the classification task, three models were chosen and trained, namely: **Long Shot-Term Memory**, or **LSTM** a recurrent neural network; **Logistic Regression**; and finally, **Convolutional Neural Network**, or **CNN**. With the intention of using models not previously found in the literature, LSTM and CNN were chosen, in order to verify the result of their application in the dataset. For comparison purposes, the Logistic Regression model was trained.

With this, a brief description of the approaches used is made.

1) Long Shot-Term Memory: Described in [DSA 2019], LSTM is a recurrent neural network (RNN) architecture that "remembers" values at arbitrary intervals. LSTM is well suited for classifying, processing, and predicting time series with time intervals of unknown duration.

Describing the model, it use was possible through the API provided by the `TensorFlow` package. The model is composed of an Embedding layer (the one that does the sentence embedding), and 3 LSTM layers, each with 256, 128, and 64 units. In an attempt to improve the prediction, 3 fully connected dense layers are stacked, each with 128, 64, and 32 neurons each, with ReLU activation function. The last layer with two neurons and a softmax activation function is used to predict the classes. The number of units in each layer, as well as the activation functions, were chosen arbitrarily.

To feed the data to the model, the tweets were transformed through sentence embedding. The data used in the embedding adjustment are provided only from the training subset. Only the texts of the tweets were used to make the prediction. This occurred because there was little understanding of how to use the other features within the model (such as user activity, for example). This was the same for the other models used in this work.

2) Logistic Regression: As [Lawton, Burns e Rosencrance] define, logistic regression is a statistical analysis method to predict a binary outcome, such as yes or no, based on prior observations of a data set. It predicts a dependent data variable by analyzing the relationship between one or more existing independent variables and can take into consideration multiple input criteria.

For the model conception, the API from the `scikit-learn` package was used. Since the model used has a regularization hyper-parameter, 10-fold stratified cross-validation was used to adjust it, using values chosen on a logarithmic scale. Classification accuracy was used to train the model, in 3000 iterations.

Transforming the input, a TF-IDF matrix was created with the training data. She was directly submitted to the model. The validation and test subsets were also transformed.

3) Convolutional Neural Network: A Convolutional Neural Network is a deep learning algorithm, which is known for its

application in image prediction. With some adaptations, this model can also be applied to the text classification task.

For this work, again, the API from the `TensorFlow` package is used. For the conception, a sequential model is used. The first layer is a sentence embedding layer. After this one, 3 Convolutional 1-dimensional layers are used with 128 filters each, and the ReLU activation function. In the first one, 4 strides are used. The kernel size used for each layer is 3, 6, and 12, from the first to the last layer. In the first two, a BatchNormalization with Max pooling is used. For the last one, a Global max pooling layer is applied. In an attempt to improve the prediction, 3 dense layers are used, each with 512, 256, and 16 neurons and a ReLU activation function. The last layer has 2 neurons and a softmax activation function. It is used to make the classification. The selection of the cited values, as well as the choice of which layers to use, were made arbitrarily.

To apply the tweets in the model, the same process used in the LSTM network was adopted here.

IV. Experimental Results

In this section, a presentation of the results is made. In addition, a comparison is also made.

A. LSTM

For the LSTM model, some attempts to improve the result of the training phase were made. For this, in addition to the manual adjustment of the hyper-parameters, changes in approaches were made, in order to try to obtain better results. In the training phase, an Adam optimizer was used. As described by [Team], this is the stochastic gradient descent method, which makes adaptive estimations of its parameters. In addition, the main metric for training was hit accuracy.

Among the measures adopted, it can be mentioned the change in training epochs (from 100 to 50, which slightly improved the results), the change in the learning rate (the model quickly converged to the best solution with a rate of 0.002), as well as changing the network used. For this, of the 4 dense layers of neurons present in the model, two were removed, leaving only the layers with 32 and 2 neurons. This helped to improve a little the results obtained. Incredibly, if more neurons, or more dense layers, were added, the model's performance would drop dramatically.

Furthermore, although 50 epochs were used, the model reached approximately maximum accuracy in the training set in the 24th iteration. Thus, the maximum accuracy of the validation set was approximately 77%. The image 1 shows the history of the loss function during training.

As shown, for the training set, the loss is close to zero, while on the validation set, it is close to 1.5. It seems like it would go higher if there were more epochs in the training, so reducing the epochs from 100 to 50 was a good approach.

Fig. 1. Loss function history for the LSTM model

B. CNN

Similar to the previous model, some attempts to improve the training results were made. What can be said is that this model performed similarly to the LSTM network. For the training phase, an Adam optimizer was used again, with a learning rate of 0.002, using precision accuracy as a training measure, and 50 epochs.

Regarding the measures adopted, one that stands out is the adjustment of the model's learning rate. Experimenting with different values, when the learning rate was low (about 1e-5), the model's accuracy rate was no more than 70% in training, and in the validation set, it was around 60%. When the learning rate was raised to $2*10^{-2}$, the accuracy increased dramatically in the training set (close to 100%) and reached 73% in the validation set. The image 2 shows the loss history using this training rate.

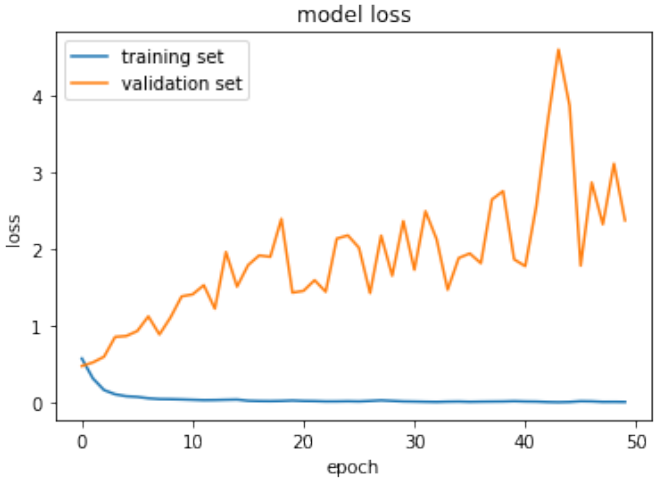

Fig. 2. Loss function history for the CNN model

Similar to the image 1, the more training epochs, the greater the loss. Also, during training, this rate went from 4, ending in 3 of loss for the validation set.

C. Logistic Regression

In this model, a little adjustment in the hyperparameters was necessary, since, in the only adjustable one, cross-validation was applied to choose the best possible value. With this, the results for this model are described. With the training done, the next step was to test the network. As the validation set had not been used before, it was submitted for testing in the model. With this, the accuracy obtained was around 78%, a very high rate. Finally, the model was submitted for a final test.

D. Final Results

Table I shows the prediction result for the three models in the test set. In it, the values of Precision, recall, and measure f1 are shown. The evaluation measures are very close, for all models. This is because the average of the values for the two classes is considered. Thus, the model with the lowest performance was the CNN network, with 0.74 accuracy, recall, and f1. The second best model was the LSTM network, with logistic regression being the best-trained model. Nevertheless, it is important to emphasize that the results obtained are very close. It can be said that the models acted in a very similar way.

Comparing the results obtained with the previous ones in the literature, the approaches made here show results as good as those already obtained. Citing the best model of this work, the accuracy presented here was around 78% for the validation set. In the test set, this value reached 79%. In [Hatoon S AlSagri e Mourad Ykhlef 2020], this value was around 82%. Furthermore, a notable difference is that, in the cited work, not only the text but the user activity was used to make the predictions, whereas here, this is not the case. Possibly, the use of user activities, as well as other features extracted from the dataset can help to improve the accuracy of the predictions. What can be concluded is that the models used here performed as well as approaches already made in other works, complying with the expected. Given an accuracy rate close to 80%, the developed models have a good performance.

TABLE I
TEST EVALUATION MEASURES FOR MODELS TRAINING

	Precision	Recall	F1
CNN	0.740	0.740	0.740
LSTM	0.782	0.782	0.782
LogReg	0.794	0.793	0.793

V. CONCLUSION

In this work, some machine learning approaches were adopted to classify texts. More specifically, it was necessary to classify user tweets as depressive or non-depressive. For this, 3 different approaches were used, namely: Long Shot-Term Memory, Convolutional Neural Network, and Logistic Regression. As shown in the results section, the tests carried out achieved good results, as expected.

Something to be highlighted is that the method developed here can be improved. A clear reason for this is that some of the approaches taken here were made this way because there was no complete knowledge of some of the resources used. For example, more features could have been used to feed the models, not just the text. Furthermore, cross-validation could have been used, for example, to adjust the hyperparameters of the LSTM and CNN networks. Such measures, if taken, would result in models with a higher rate of precision and accuracy. In future work, these approaches can be taken, as well as the training of new networks, to improve the method presented here.

Finally, despite these conditions, the method presented here proved to be a good measure to predict Twitter users with depression. The models presented could be used to carry out the task on a larger scale, given their agility and cost, to reduce the workload of professionals in the area. With the supervision of a professional, the correct diagnosis can be made faster and cheaper.

REFERENCES

[Depression: Twitter Dataset + Feature Extraction]DEPRESSION: Twitter Dataset + Feature Extraction. Disponível em: <https://www.kaggle.com/datasets/infamouscoder/mental-health-social-media>.

[DSA 2019]DSA, E. *Capítulo 51 - Arquitetura de Redes Neurais Long Short Term Memory (LSTM)*. ago. 2019. Disponível em: <https://www.deeplearningbook.com.br/arquitetura-de-redes-neurais-long-short-term-memory/>.

[Hatoon S AlSagri e Mourad Ykhlef 2020]Hatoon S AlSagri; Mourad Ykhlef. Machine Learning-based Approach for Depression Detection in Twitter Using Content and Activity Features. *arXiv.org*, 2020. Place: Ithaca Publisher: Ithaca: Cornell University Library, arXiv.org.

[Kim et al. 2021]KIM, J. et al. A Systematic review of the validity of screening depression through Facebook, Twitter, Instagram, and Snapchat. *Journal of Affective Disorders*, v. 286, p. 360–369, maio 2021. ISSN 0165-0327. Disponível em: <https://www.sciencedirect.com/science/article/pii/S016503272100135X>.

[Lawton, Burns e Rosencrance]LAWTON, G.; BURNS, E.; ROSENCRANCE, L. *What is Logistic Regression? - Definition from SearchBusinessAnalytics*. Disponível em: <https://www.techtarget.com/searchbusinessanalytics/definition/logistic-regression>.

[Lin et al. 2016]LIN, L. y. et al. ASSOCIATION BETWEEN SOCIAL MEDIA USE AND DEPRESSION AMONG U.S. YOUNG ADULTS. *Depression and anxiety*, v. 33, n. 4, p. 323–331, 2016. ISSN 1091-4269. Place: United States Publisher: United States: Blackwell Publishing Ltd.

[Marcelo Basso de Sousa 2013]Marcelo Basso de Sousa. Depressão. *Clinical and Biomedical Research*, v. 32, n. 4, 2013. Publisher: Hospital de Clinicas de Porto Alegre.

[McCarron et al. 2021]MCCARRON, R. M. et al. Depression. *Annals of internal medicine*, v. 174, n. 5, p. ITC65–ITC80, maio 2021. ISSN 1539-3704 0003-4819. Place: United States.

[Reece et al. 2017]REECE, A. G. et al. Forecasting the onset and course of mental illness with Twitter data. *Scientific reports*, v. 7, n. 1, p. 13006–11, 2017. ISSN 2045-2322. Place: England Publisher: England: Nature Publishing Group.

[Sentence embedding 2022]SENTENCE embedding. set. 2022. Page Version ID: 1109035949. Disponível em: <https://en.wikipedia.org/w/index.php?title=Sentence$_embeddingoldid$ = 1109035949¿.

[Team]TEAM, K. *Keras documentation: Adam*. Disponível em: <https://keras.io/api/optimizers/adam/>.

[tf–idf 2022]TF–IDF. jul. 2022. Page Version ID: 1098671947. Disponível em: <https://en.wikipedia.org/w/index.php?title=Tf

[Twitter 2022]Twitter. *About different types of Tweets*. ago. 2022. Disponível em: <https://help.twitter.com/en/using-twitter/types-of-tweets>.