# CLASSIFICATION OF DEPRESSION USING TEMPORAL TEXT ANALYSIS IN SOCIAL NETWORK MESSAGES

Gabriel Melo, KaykeBonafé and Guilherme Wachs-Lopes

Department of Computer Science,
University Center of FEI, São Paulo, Brazil

***ABSTRACT***

*In recent years, depression has gained increasing attention. As with other disorders, early detection of depression is an essential area of study, since severe depression can result in suicide. Thus, this study develops, implements, and analyzes a computational model based on natural language processing to identify the depression tendencies of Twitter users over time based on their tweets. Consequently, an F-measure of 83.58 % was achieved by analyzing both the textual content and the emotion of the papers. With these data, it is possible to determine whether constant fluctuation of emotions or the message in the text is a more accurate indicator of depression.*

***KEYWORDS***

*Depression, Natural Language Processing, Machine Learning.*

## 1. INTRODUCTION

According to the WHO [1] more than 294 million people worldwide suffer from depression. Works, such as [2] and [3], states that early diagnosis is a crucial aspect in determining the efficacy of a treatment, therefore reducing the probability of the disease escalating to severe depression or even suicide.

In this sense, social networks can be a great field of study for the early detection of depressive behaviors. An example of this is the study of [4], which proposes to infer whether there is a relationship between the use of multiple social networks and the development of conditions such as anxiety and depression. The results of this study showed that there is a correlation between these factors. Another is the study of [5], which is a model of text classification that aims at early detection of depression in social media streams. The results of this paper show that the model proposed by the authors outperforms the most used machine learning models, like SVM.

Following this point, it is possible to create a model that can show the information related to the feelings contained in the text, in this case, the variation of feelings, so that it is possible to infer how much influence this variation affects the classification.

Therefore, the goal of this work is to propose, implement, and evaluate a computational model based on natural language processing to classify depressive tendencies of Twitter users through their posts over time. The second goal is to discuss how sentiment analysis and the textual content itself influence the classification score.

David C. Wyld et al. (Eds): ARIA, SIPR, SOFEA, CSEN, DSML, NLP, EDTECH, NCWC - 2022
pp. 167-177, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.121514

## 2. METHODOLOGY

The proposed methodology pipeline starts with dataset formatting and data pre-processing. Then, the pipeline proceeds to the vectorization step and sentiment analysis. Finally, the last step is the classification output. This pipeline can be seen in Figure 1.

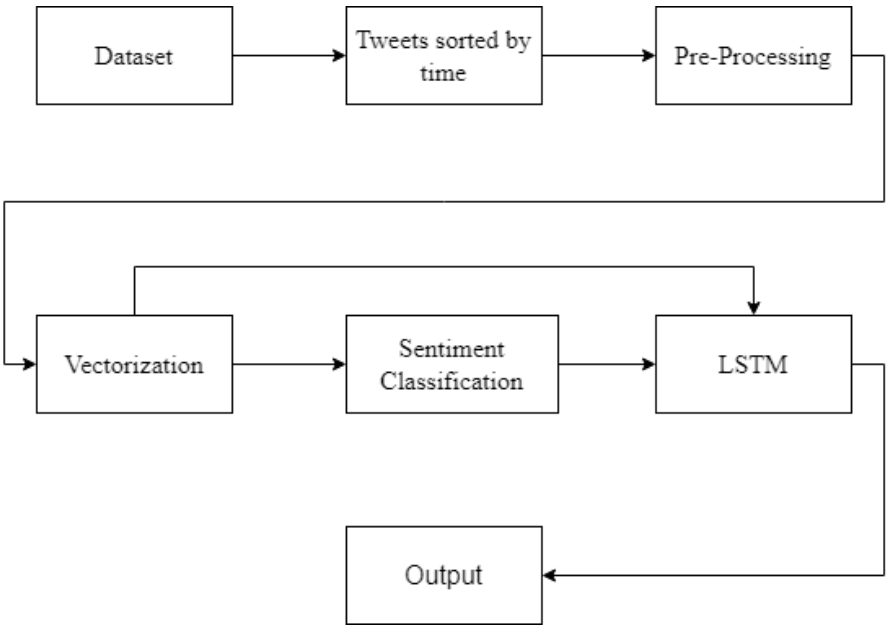


Figure 1. Methodology Pipeline

### 2.1. Dataset and Data Pre-Processing

The dataset used in this study was developed by [6]. It consists of textual data collected from Twitter from 2009 to 2016 and divided into 3 different categories: **D1,** which consists of users who explicitly have depression. This classification is done if the user has a post with a pattern similar to "(I'm/ I was/ I am/ I've been) diagnosed depression"; **D2** consists of users who have not been classified as depressed. This happens if the term 'depress' was not found in any publication; **D3** consists of unclassified data. This occurs if the term 'depress' is found, but it is not identified in the pattern used in D1.

Users with more than 15,000 followers were excluded from this dataset because they may flag accounts of organizations, notable persons, or bots. Tweets that describe a narrative or a narrative have also been deleted, since they may indicate someone else's narrative or a fake story. Additionally, it was eliminated outdated information, such as "I was diagnosed with depression when I was 10."

The data is in JSON (JavaScript Object Notation) format and divided into multiple folders, where each tweet can be viewed individually or the entire user timeline, as well as data associated to their accounts such as the user's username and profile description (for security reasons, no personal data was revealed). The most significant data in each tweet structure are its creation date and textual content. The texts are not arranged in chronological order.

Considering the pre-processing step, the first goal is to increase the quality of the data obtained from the dataset, where some techniques can be applied and combined in order to make the texts

denser (word uniformity and with greater cooccurrence) so that the created model can process the data more accurately. In order to decrease textual complexity, we removed special characters, emojis, and nonlatin alphabet characters, like Japanese Kanji.

After the tokenization process, it is verified that some tokens have no semantic value, being useful only for the formalism and rules of the language, not adding any relevant information. These tokens are known as stop words and are part of the so-called stop list, which is a list of predefined words. These stop-lists are removed in order to improve data quality.

## 2.2. Vectorization and Feature Extraction

In the vectorization stage, the pre-processed texts are transformed into valued vectors using Doc2Vec. In addition, we also use sentiment analysis to study how this feature can contribute to the final classification.

During the vectorization step, text that has already been pre-processed is converted to value vectors by using the Doc2Vec algorithm. In addition to that, we also make use of sentiment analysis in order to investigate the potential role that this attribute plays in the overall categorization.

An SVM classifier is used to perform sentiment analysis. In the first step of the vectorization process, each document is processed through a word counter (BoW) and a standard scaler. We carried out an SVD reduction in conformity with the results shown in Table 1 to accomplish the dual goals of decreasing the matrix's dimensions and raising its density (Section 4.1). Following this step, the produced vectors are fed into a support vector machine (SVM) in order to classify the polarity of sentiment for the sentences that were collected from Twitter.

Since the dataset employed in this research lacks a sentiment ground truth, a second dataset is required. This dataset was created by [7], and all tweets were classified as having either a positive or negative sentiment polarity using many techniques, including SVM.

## 2.3. Training and Classification

During the training and classification stage, the collected preprocessed data is structured as input parameters for the SVM and LSTM.

SVM is used for sentiment analysis. However, as described in the vectorization section, this training cannot be done on the depression dataset, since there is no supervised information on feelings. As a result, the model is trained using the dataset from [7]. The texts from the Twitter dataset will be inferred from the vectorized documents using BoW and their dimensions will be decreased using SVD, as outlined in Section 2.2.

After entering the vectorized BoW data into SVM, the output will be a vector that represents how much positive feeling is present in the analyzed text. The trained algorithm is used in the dataset developed by [6] that classifies tweets as depressive and non-depressive, as presented above.

At this point, there are two vector representations for each tweet. The first is obtained by Doc2Vec and the second is the sentiment polarity classified by SVM. The final classification of the data occurs through the LSTM network that is fed with the vectorized sentences of the dataset (tweets). Each of these sentences enters the timestep dimension of the LSTM.

The result of this process is given by a numerical value between 0 and 1, which indicates the level of depression in the sentence, with 1 being the presence of depression and 0 the absence. This process is illustrated in Figure 2.

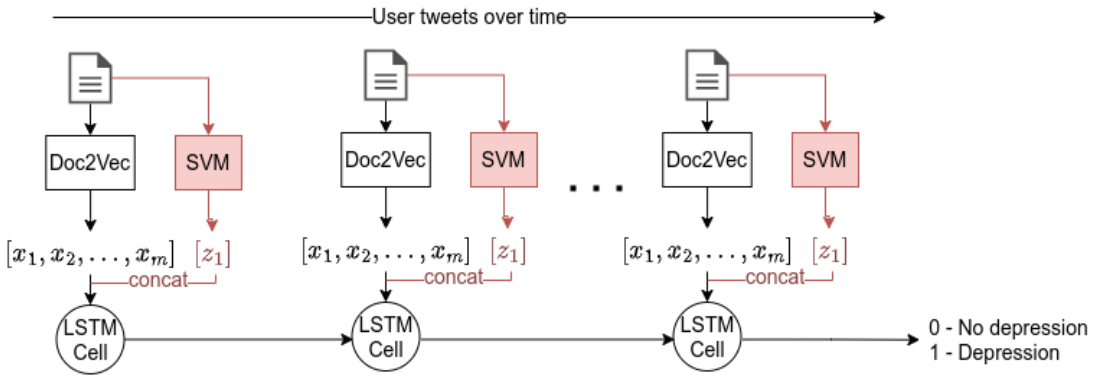


Figure 2. LSTM usage

## 3. EXPERIMENTS

### 3.1. Sentiment Classification

The SVM is trained independently using the dataset that has sentiment classifications of texts, described in Section 2.3, to be used in the classification of the LSTM. To carry out the training, 70% of the dataset is used as training data and 30% as input data for classification and subsequently tool verification. The input is the text of the post in Twitter and the output is the positive or negative value detected in the tweet. To check the score level, four metrics are used: precision, recall, F-Measure, and accuracy.

### 3.2. Vectorization

Since Doc2Vec has some tuning parameters, such as window size and embeddings vector dimension, this experiment consists of analysing how they contribute to describing document contents as vectors. For instance, if the embedding size is too high, this can lead to high memory usage, and LSTM network may receive inputs with higher dimensions. However, if the embedding size is too small, Doc2Vec could not have enough vector space to describe all the documents from the dataset.

Therefore, the goal of this experiment is to find a balance between the size of the embeddings and the discrimination of the documents. The first step is to choose $n$ randomly tweets from the dataset $D$ and store in a list $L$. Then, Doc2Vec is trained with different values of window size and embedding size. For each training, we choose a random generated database, $R = L \cup \{x \mid x \in s(D, z)\}$ where $s$ is a function that returns $z$ sample tweets.

Finally, for each document $d \in L$ we infer document embedding from Doc2Vec and compare it for each document in $R$ dataset using cosine similarity. This process generates a list of tweets sorted by similarity with respect to document $d$. When the most similar document in $R$ dataset is the document $d$ it means that the Doc2Vec could generate discriminant vectors. However, the farther $d$ is from first position, the lower is the discrimination between documents. Figure 3 illustrates the whole process.

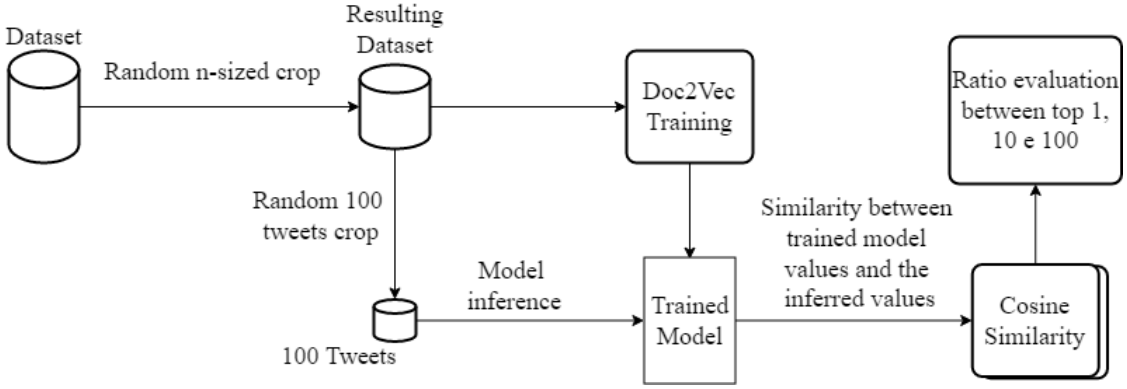


Figure 3. Visual representation of vectorization experiment

The results are computed counting how many documents were found into Top 1, 10 and 100 positions of distance sorted list.

### 3.3. Doc2Vec and SVM

As described in Section 2.3 and illustrated in Figure 2, there are two mechanisms to extract features from tweets: Doc2Vec and Sentiment Analysis. Doc2Vec is used for textual analysis in order to vectorize the tweets in the text dataset in order to comply with the input format expected by LSTM network. For sentiment analysis, SVM is trained to evaluate sentiment from texts analyzed in the dataset.

As one of the goals of this work is to study how sentiment analysis can influence depression classification, we propose an experiment that compares two classifiers: with both information (Doc2Vec + SVM); and with only information on document content (Doc2Vec).

After performing both training sessions, it is possible to measure whether the detection of depression is more related to the constant variation of emotions or the final message conveyed by the text.

### 3.4. Stemming Validation

In this step, an experiment is carried out in which the sentiment dataset are submitted to a classification evaluation through the SVM to validate the impact of using stemming on the result.

## 4. RESULTS

This paper achieved three main results: sentiment classification with SVM, vectorization with Doc2Vec, and depressive tendencies classification. The following sections discuss the results.

### 4.1. Sentiment Classification

The method used for the sentiment classification was the SVM. Several executions of the training have been performed by varying some parameters. Beyond that, two different datasets were used, with and without stemming. Another parameter on the execution is the words sparse matrix dimensionality, generated by the BoW. The starting dimensions were 25, going up to 300. When validating with the F-Score measure, the best results got a precision of 70.11%, a recall of 78.35%, an accuracy of 72.48, and an F-Score of 74%. The results are shown in Table 1.

Table 1. Sentiment classification results

| Dataset | BoW Dimensions | Precision | Recall | Accuracy | F-Measure |
|---|---|---|---|---|---|
| With Stemming | 25 | 60.96% | 68.83% | 62.24% | 64.65% |
| | 50 | 63.09% | 73.82% | 65.46% | 68.04% |
| | 75 | 65.60% | 75.72% | 68.01% | 70.30% |
| | 100 | 66.95% | 75.65% | 69.08% | 71.04% |
| | 200 | 68.75% | 78.11% | 71.32% | 73.14% |
| | 300 | 70.11% | 78.35% | 72.48% | 74.00% |
| Without Stemming | 25 | 57.18% | 72.39% | 59.04% | 63.89% |
| | 50 | 62.59% | 72.35% | 64.51% | 67.11% |
| | 75 | 63.06% | 74.80% | 65.63% | 68.43% |
| | 100 | 65.81% | 76.64% | 68.41% | 70.82% |
| | 200 | 68.01% | 76.87% | 70.33% | 72.17% |
| | 300 | 69.60% | 77.77% | 71.83% | 73.46% |

According to Table 1, we can observe that the bigger the dimensionality, the better the result. The conclusion here is that the higher dimensions the bigger is the vector space used to describe a word. Therefore, more details of the document are captured. The outcome would have been better if there had been additional dimensionality added, however due to time and hardware constraints, only 300 dimensions were used.

## 4.2. Vectorization

On the text vectorization, the method used was Doc2Vec. Seven runs were conducted during this stage, all with a dataset percentage, starting with 1000 tweets going up until 3,192,403. For this experiment, the used dimensionality was 300. When tested, the described model reached an average of 95.91% of similarity between the generated vectors on the training and the vectors generated for the validation.

The results of the experiments are shown in Figure 4, where the **x** axes represent the amount of data used, the left **y** axes represent the percentage of times that the model's first result is the same as the used validation vector and the second **y** axes represent the training time. On the first 6 experiments, all searched tweets were between the first 10 to 100 positions, only on the last experiment that this number goes to 91 on the first 10 positions and 98 on the first 100. In the first case, this happens because of the difference between the vectorial representation of the words. On the last two, this decrease is justified by the amount of data used.

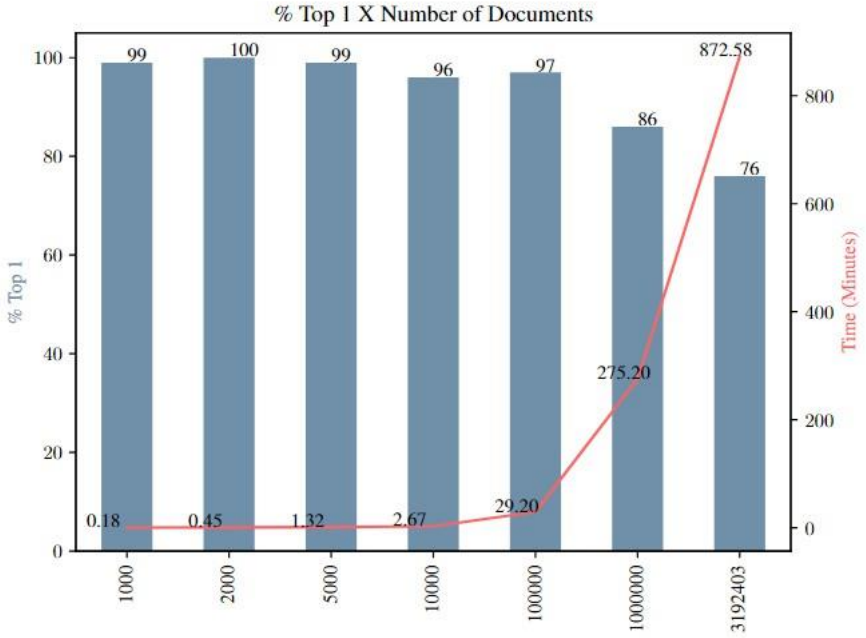


Figure 4. Number of documents versus the percentage of times that the first result of the model was identical to the result of the inference

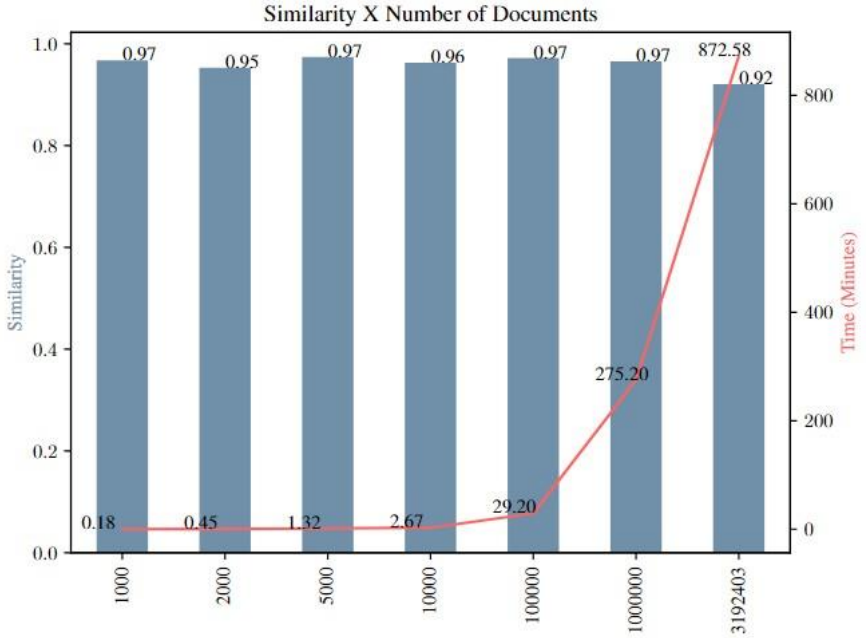


Figure 5. Similarity versus Number of Documents

Last, the same observation can be made if we analyze the average similarity between the tweet used for the validation and the existing in the model, which decreases slightly, as can be seen in Figure 5. In this Figure, the *x* axes represent the amount of data used, the first *y* axes represent the similarity and the second one represents the time used on the training.

### 4.3. Depressive Tendencies Classification

For this purpose, the LSTM was executed with two different strategies: using textual data from Doc2Vec (textual); and using both sentiment analysis and combined textual data (concatenated), this being a transfer learning approach. Two different executions of the experiments occurred, each with random data from the dataset, both on the training and on the validation. The hidden state size was one and two times the input size.

It is possible to see the variation of F-Score in the three strategies on image, as well as the hidden state sizes. It is noticeable that increasing the hidden size increases the F-Score, in most cases.

The best F-Score was obtained using the concatenated data as well as the hidden size being two times the input size. As shown in Table 2, the best result obtained a precision of 82.87%, recall of 84.31%, accuracy of 83.46%, and an F score of 83.58%. The threshold used for the metrics was found by maximizing the Geometric Mean using a ROC curve, where the best AUC was 0.90. The curve can be seen in Figure 6.

Initially, the entire dataset was being used, having more than 3 million documents. However, when validation was being done, the results had around 40% F-Score and 8% of accuracy, which was evidence of an unbalanced dataset. The initial proportion was around 9 nondepressive users to 1 depressive user. After the data were balanced, the dataset had 50% depressive and non-depressive users, totaling around 900 thousand documents. The final results can be seen in Table 2.

Table 2. LSTM Results

| | Dataset | hidden size | Precision | Recall | Accuracy | F-Measure | Threshold |
|---|---|---|---|---|---|---|---|
| Execution 1 | Textual | 300 | 80.87% | 82.02% | 81.34% | 81.44% | 54.01% |
| | | 600 | 82.48% | 84.15% | 83.15% | 83.31% | 52.41% |
| | Textual + Sentiment | 301 | 82.25% | 81.93% | 82.14% | 82.09% | 92.67% |
| | | 602 | 82.41% | 84.04% | 83.06% | 83.22% | 42.47% |
| Execution 2 | Textual | 300 | 81.62% | 82.75% | 82.12% | 82.19% | 62.35% |
| | | 600 | 82.65% | 84.21% | 83.30% | 83.42% | 72.41% |
| | Textual + Sentiment | 301 | 81.89% | 82.13% | 81.99% | 82.01% | 80.44% |
| | | 602 | 82.87% | 84.31% | 83.46% | 83.58% | 60.93% |
| Execution 3 | Textual | 300 | 82.28% | 82.63% | 82.43% | 82.45% | 76.66% |
| | | 600 | 83.70% | 84.36% | 83.98% | 84.03% | 64.59% |
| | Textual + Sentiment | 301 | 80.21% | 82.47% | 81.43% | 81.64% | 35.14% |
| | | 602 | 83.15% | 84.96% | 83.84% | 84.04% | 28.91% |
| Execution 4 | Textual | 300 | 82.08% | 83.37% | 82.56% | 82.72% | 60.15% |
| | | 600 | 83.69% | 84.50% | 84.00% | 84.10% | 74.49% |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | Textual + Sentiment | 301 | 82.27% | 82.81% | 82.46% | 82.54% | 59.68% |
| | | 602 | 83.66% | 83.95% | 83.73% | 83.81% | 61.08% |
| Execution 5 | Textual | 300 | 82.39% | 83.52% | 82.80% | 82.95% | 81.41% |
| | | 600 | 83.68% | 84.20% | 83.85% | 83.94% | 75.71% |
| | Textual + Sentiment | 301 | 82.59% | 82.83% | 82.65% | 82.71% | 49.76% |
| | | 602 | 83.10% | 84.56% | 83.71% | 83.83% | 49.77% |

When comparing the F1 scores of the two techniques using the Student T Test, we can see that there is no significant difference between the classification test that uses only vectorized textual data and the test that uses vectorized textual data and sentiment information.

The test results show a p-value of 65.37% for the LSTM with hidden size of 300 (for the textual data) and 301 (for the textual with sentiment analysis data). Furthermore, the p-value of 77.27% for the LSTM with hidden size of 600 (for textual data) and 602 (for the textual with sentiment analysis data). These test results mean a non-trustable confidence interval, confirming that the sentiment analysis does not make any significant improvement in the overall classification.

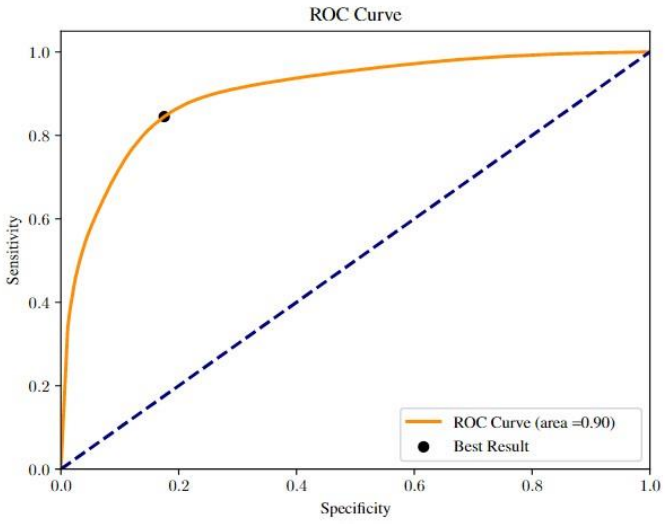


Figure 6. ROC Curve of the best result

## 5. CONCLUSIONS

This paper proposed a computational model based on natural language processing to classify depressive tendencies in tweets. For this goal, two different approaches were used: classification through text analysis; and classification using both described methods.

The results show that depressive tendencies can be detected using textual content and textual content combined with sentiment analysis. Furthermore, another finding was that textual models were more relevant to the classification than to the sentiment analysis. This indicates that the existing sentiment on the text is not a piece of discriminant information for classification.

Remarkably, both the textual data and both textual and sentiment together got similar results, making the use of sentiment classification unnecessary, when considering the computational cost.

As a contribution to this project, the proposed method got an F-Score of 83.58% using only textual information, in comparison, the state-of-the-art related to depression detection has an F-Score of 97% on [8], research that used not only textual content but images as well, which is not used in this paper. The model as well as the code can be found on Github, available at: https://github.com/gabrielomelo/tcc-lstm.

In future works, it is suggested to increase the density number of the data to improve the model precision as well as the sentiment analysis data precision as a way to aggregate the most accurate pieces of information to the concatenated model. Another suggestion in the sentiment analysis model could be the change from linear to the RBF Kernel in order to increase the classification accuracy.

In addition to this point, there is a possibility of doing a study on how the textual serialized information by time contributed to improving the classification quality from the proposed model. Thus, it is suggested to create a simpler model that considers only one tweet from the user. The main hypothesis here is that the textual serialized information by time contributes to the improvement of the F score.

To conclude, for more evidence about the model efficiency, it is expected that this line of study and investigate the model quality will be investigated through other statistical tests such as K-fold.

## REFERENCES

[1] WORLD HEALTH ORGANIZATION (2021) Depression. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/depression

[2] W. Yu-Tseng, H. Hen-Hsen and H. Chen, (2018) A Neural Network Approach to Early Risk Detection of Depression and Anorexia on Social Media Text. Avignon, France: CLEF 2018.

[3] Paul, S.; Jandhyala, S. K.; Basu, T. (2018) Early detection of signs of anorexia and depression over social media using effective machine learning frameworks. West Bengal, India: CLEF 2018.

[4] B. A. Primack, S. Ariel, C. G. Escobar-Viera et al., (2017) Use of multiple social media platforms and symptoms of depression and anxiety: A nationally representative study among U.S. young adults. Pittsburgh, United States of America: Computers in Human Behaviour.

[5] Burdisso, S.G., Errecalde, M.L., & Montes-y-Gómez, M. (2019). A Text Classification Framework for Simple and Effective Early Depression Detection Over Social Media Streams. San Luís: Argentina: Expert Syst. Appl. 2019.

[6] G. Shen and J. Jia and L. Nie et al., (2017) Depression Detection via Harvesting Social Media: A Multimodal Dictionary Learning Solution. Hefei, China: IJCAI-17.

[7] A. Go and R.Bhayani and l. Huang, (2009) Twitter sentiment classification using distant supervision. California, United States: CS224N.

[8] R. Kumar and S. K. Nagar and A. Shrivastava, (2020) Depression Detection Using Stacked Autoencoder from Facial Features and NLP. Bhopal, India: SMART MOVES JOURNAL.

## AUTHORS

**Gabriel Melo** has a Computer Science bachelor degree (University Center of FEI, 2021), is interested in the following topics: complex networks, neural networks, natural language

processing and cyber security. Currently works as an information security analyst developing cyber intelligence tools (Itaú Unibanco S.A.).

**Kayke Bonafé** has a degree in Computer Science (University Center of FEI, 2021), is interested in the following topics: artificial intelligence, neural networks, natural language processing and machine learning. Currently works as a data scientist developing reports and models (Monett Conteúdo Digital LTDA.).

**Prof. Dr. Guilherme Wachs** has a Computer Science bachelor degree, master in Artificial Intelligence and PhD in Signal Processing Area. Currently, is a researcher and professor of A.I. group of University Center of FEI, and interested in NLP, IoT and Computer Vision.