INTERNATIONAL BURCH UNIVERSITY
FACULTY OF ENGINEERING, NATURAL AND MEDICAL SCIENCES
DEPARTMENT OF INFORMATION TECHNOLOGIES

MENTAL HEALTH PREDICTION USING SENTIMENT ANALYSIS

PROJECT PAPER

Students:
Arnela Ombaša, Amina Hamzić

Supervisor
Prof. Dr. Nermina Durmić

SARAJEVO
January 2024

ABSTRACT

In the digital age, social media influences society well-being, especially concerning mental health. This research focuses on using sentiment analysis on data, to predict mental health status. Early detection of these conditions is very important to prevent health issues and to help develop better communities on the internet.

The study uses methods like Random Forest and Naive Bayes, supported by a confusion matrix, to analyze data from different sources on social media. It strives to create a framework that harnesses digital platforms to address mental health challenges, offering a proactive and technology-driven strategy for a healthier society.

Keywords: Mental Health, Python, Sentiment Analysis, Natural Language Processing

INTRODUCTION

In today's digital world, how we feel emotionally is really important. People share their thoughts and feelings on social media a lot. With more and more people, especially young people, facing mental health challenges, we need new and smart ways to spot and help early.

Our digital age is a special time where technology and our feelings meet. Social media, like a giant canvas, shows us what people are going through. It tells stories of happiness, struggles, and hints about mental health.

In simple terms, this study explores the changing relationship between technology and mental health. It recognizes the powerful role of digital platforms and how they can create a kind and understanding online space.

LITERATURE REVIEW

From the research paper (Tiwari et al, 2021) analyzing real-time data collected in social media platforms such as Twitter using sentiment analysis has potentially predicted ailments like depression anorexia and associated mental illnesses among young individuals. The early detection of depression plays a crucial role, for the reasons it is underneath so many health problems as well and that causes suicides can be minimized. The aim of this study is to detect depression and Post Traumatic Stress Disorder (PTSD) in users of Twitter.

Today's research often involves the use of sentiment analysis. According to (Zucco, Calabrese & Cannataro 2017) sentiment analysis is an essential tool that can be utilized in research for the purpose of understanding and analyzing people's opinions, attitudes, emotions towards certain topics or products. It includes the utilization of natural language processing and content mining methods to determine subjective information from semi-structured data in text form including social media posts, online reviews , or surveys.

(Zucco, Calabrese & Cannataro 2017) also note that sentiment analysis could be integrated with other data analyzing approaches like machine learning or visualizing the data in order to get a more complete picture on the nature of information. This will help us to find trends, patterns and correlations that may not be obvious at first glance from the raw data.

One of the most important parts of analyzing data is preprocessing. The established practice is that after importing the .csv file, data cleaning is performed, as seen in the (Tiwari et al 2021) research paper, where the tweets were processed to remove emoticons and punctuation marks, retaining only the text and user IDs for further analysis.

One of the most important parts of every research paper is how to present data. The confusion matrix gives a detailed segmented view for accurate and flawed predictions carried out by such modeling classification which helps the researchers to study true positive, true negative, false positive as well as False Negative . This analysis allows understanding each classifier's strengths and weaknesses when identifying sentiments tied to mental illness in data. Also based on (Tiwari et al, 2021) paper, the confusion matrix was used to evaluate the performance of different classifiers in predicting the sentiment and likelihood of mental illness based on Twitter data, which is a good sign for our choice.

Natural language processing and machine learning can have wide applications in prediction of mental health problems. As seen in (Calvo, Milne, Hussain & Christensen, 2017) paper, analyzing language patterns and sentiment in non-clinical texts, NLP and ML can help in early detection of mental health issues, allowing for timely intervention and support.

RESEARCH QUESTION

With this research paper, we aim to explore whether sentiment analysis of textual data can effectively predict mental health status. We intend to examine what people write, particularly on social media, to determine if certain words or patterns can indicate if someone is feeling sad, anxious, or experiencing other mental health issues. The objective is to identify ways to provide early assistance and support to individuals who might be going through challenging times based on their online expressions.

METHODS AND MATERIALS

DATASET

The dataset used in this research, sourced from Kaggle, has a body of text data categorized for sentiment, which serves as an indicator of the user's mental health status. This dataset encapsulates a broad spectrum of expressions, ranging from daily experiences to explicit references to mental health challenges. The preprocessing stage involved standardizing the text data using cleaning techniques, eliminating irrelevant characters and stopwords. Subsequently, the data underwent tokenization and vectorization, transforming the textual content into a numerical representation conducive to analysis.

The Mental Health Corpus is composed of data from individuals dealing with anxiety, depression, and various other mental health issues. This corpus has two columns: the first containing comments and the second featuring labels classifying whether the comments are expressing mental health concerns or not. It has 27977 records which have 2043667 words and 72650 unique words.

RESEARCH INSTRUMENTS AND MATERIALS

For processing and analyzing the textual data, we used the Python programming language, along with several libraries. The Natural Language Toolkit (nltk) was used for text processing and sentiment analysis. Matplotlib library for creating visualizations in Python, a spaCy library for advanced text processing. Also we used the NumPy library for numerical operations in Python, providing support for large, multi-dimensional arrays and matrices, along with mathematical functions, Pandas and Seaborn.

Machine learning models, including the Random Forest Classifier and Naive Bayes classifier from the scikit-learn library, were the primary tools for prediction analysis.

DATA PREPROCESSING

Process started with importing dataset into dataframe, after which we tested if a csv file is imported and if there are some missing values in the dataset.

The following code snippet analyzes and visualizes the distribution of labels (0 or 1) in a dataset. It uses the matplotlib.pyplot library to create a bar plot, where the height of each bar represents the frequency of each label. The distribution is calculated using the value_counts() function.

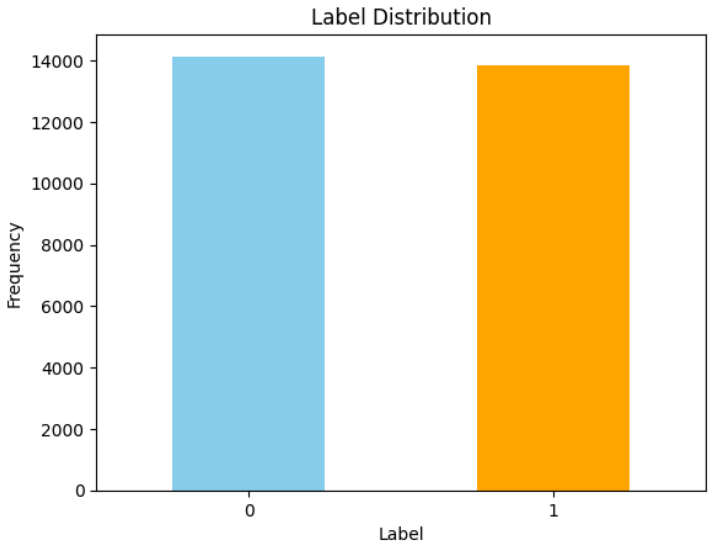

Figure 1: The Distribution Of Labels (0 or 1) In The Dataset

After label distribution, we started with data preprocessing, everything with the goal to prepare data for machine learning algorithms. First of all we created a list of sentences, and joined them in one raw text. After that, we splitted text so we can count the total number of words in the dataset and total number of unique words, as well as frequent distribution for finding the most used words. We created a function remove_stopwords to remove all stopwords and non alpha words, returning them in lowercase. The next step was creating a new column in a dataset without stopwords in order to see exact numbers of unique words and stopwords in text. We are removing stop words from text because they do not add any value to our data, so it is better to remove them. We reduced our total word count from over 2 million to 1.90 million, but unique words went from 72,650 to 72,092.

Tokenization and lemmatization are two fundamental techniques in natural language processing that we used in text preprocessing before applying machine learning algorithms. Tokenization is the process of breaking down text into smaller parts, typically words. For example, the sentence "I am studying at IBU" would be tokenized into the tokens "I", "am", "studying ", "at " and

“IBU”. In sentiment analysis and mental health prediction, that we are doing, tokenization allows the algorithm to process individual words or phrases. This is crucial because the sentiment or emotional tone of the text is often conveyed at the word level. To remove special characters and punctuation, we used the 're' library.

Lemmatization is a more sophisticated approach to reducing words to their base or root form. Unlike stemming, which crudely chops off word endings, lemmatization considers the context and morphological analysis of words to bring them back to their base or dictionary form. For example, "running", "ran", and "runs" would all be lemmatized to "run".
In our sentiment analysis, lemmatization helps in standardizing words to their base form, reducing the complexity of the text data and improving the performance of the machine learning models. This is especially useful in sentiment analysis, as it helps in accurately capturing the sentiment by grouping together various forms of the same word. For instance, "happiness", "happy", and "happily" would all contribute similarly to the sentiment analysis after lemmatization. But during this process, we had an issue when using lemmatization from the NLTK library. Therefore, to achieve better results, we used lemmatization from the spaCy library.

After text preprocessing, text is ready for further analysis and applying machine learning algorithms. In this part we used Random Forest Classifier and Naive Bayes Classifier.
Random Forest Classifier is an ensemble learning method that operates by constructing a multitude of decision trees during training and outputs the class that is the mode of the classes or mean prediction of the individual trees. We used it to predict mental health status based on sentiment analysis of textual data. Beside random forest classifier, Naive Bayes Classifier was used which is a probabilistic classifier based on Bayes' theorem, which assumes that the features used to describe an observation are conditionally independent given the class label. Naive Bayes was a good choice because it captures probabilities associated with certain words or phrases indicating mental health states.

RESULT AND DISCUSSION

As for results, we will first talk about identifying the most prevalent words in our dataset. The word 'not' appeared 19,447 times, suggesting a significant presence of negation in the texts related to class 1, which often signals heightened emotion or stress. Similarly, the word 'like', with 7,319 appearances, was the most common in class 0 texts, indicating a comparative or simulative context in these entries. We used the NLTK library's frequency distribution for this part.

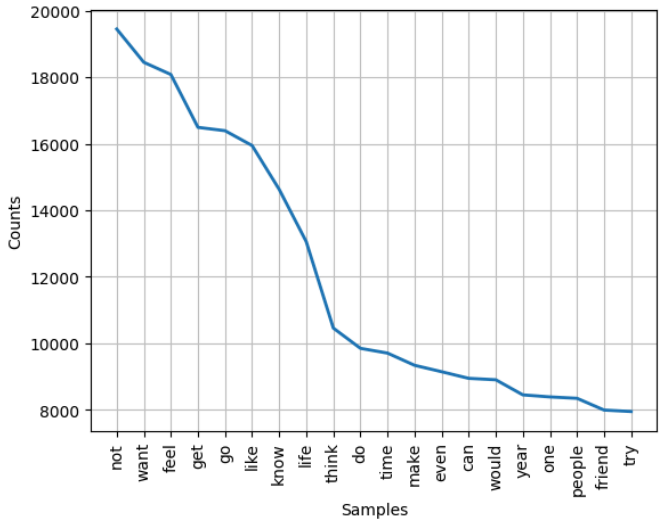

Figure 2: Most Prevalent Words In Our Dataset From Class 1

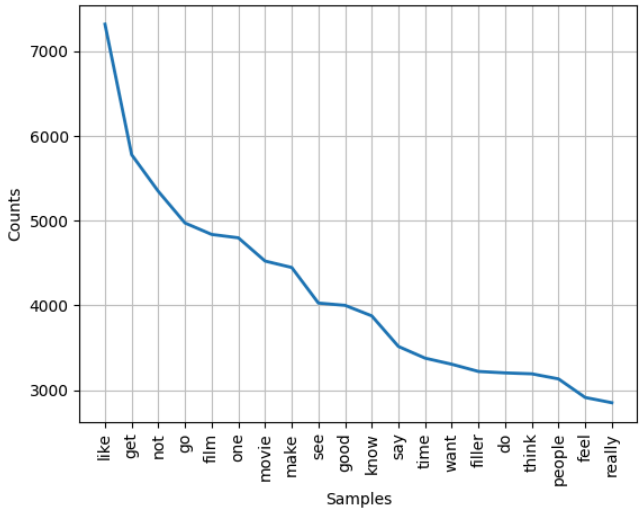

Figure 3: Most Prevalent Words In Our Dataset From Class 0

Initially, the corpus contained a large total number of words that significantly decreased after deleting stopwords – common words that provide very little informational value. Finally, the removal of unnecessary characters and patterns using regexes resulted in another reduction to word count. Despite such reductions, the number of distinct words — those that contributed to textual diversity – shrank only slightly. This implies that the cleaning process managed to reduce redundancy while maintaining lexicality richness of corpus. It is obvious from the bar chart illustration that data cleaning has a significant impact on the corpus; thus, it manages to strike one fine balance between simplification of information and preserving weighty content which can be taken into analytical account.

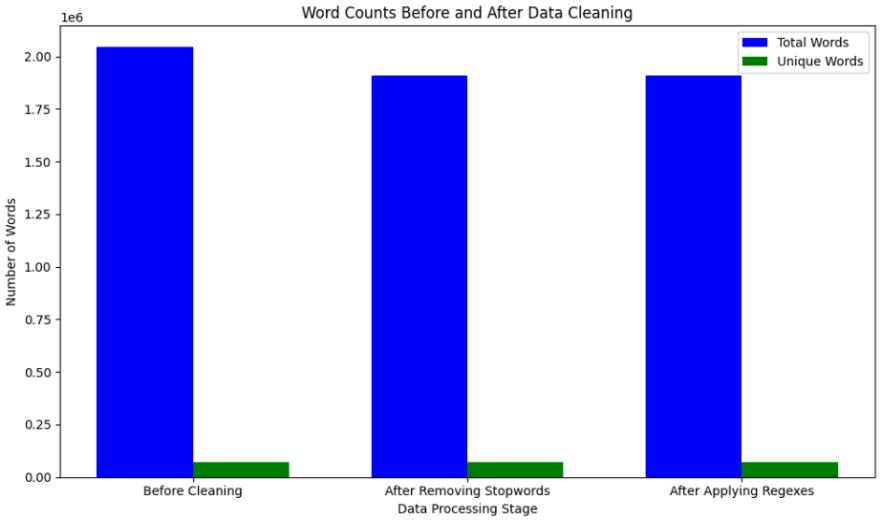

Figure 4: Impact of Data Cleaning on Corpus Word Count and Diversity

For the machine learning aspect, we first transformed our textual data into a numerical format, using the CountVectorizer from Scikit-learn, which provided us with a feature matrix representing the occurrence count of words. The 'cleaned_text' column became our feature set X, and the 'label' column our target variable y.

	aa	aaa	aaaa	aaaaa	aaaaaa	aaaaaaa	aaaaaaaa	aaaaaaaaa	aaaaaaaaaaaa	aaaaaaaaaaaaa	...	zuess	zula	zulaaynurmzeyyan
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0

Figure 5: Final Data Set

When preparing our data for the models, we divided it into training and testing sets, keeping its stratification to a class distribution of the target variable to preserve the proportion of classes. Then, the Random Forest Classifier was trained with an accuracy of about *88.6%* on test data. This was a good outcome, though it needed more detailed analysis, specifically the lens of a confusion matrix to know its precise and recall percentages.

We used Random Forest Classifier as it was used in (Tiwari et al, 2021) paper, which includes its application as a machine learning algorithm for sentiment analysis and the prediction of mental illness based on Twitter data. The classifier was trained and tested using the Twitter data to predict the likelihood of depression and PTSD among users. We used it on our data set to predict mental health status.

On the other hand, the Naive Bayes model had an accuracy of about *84.5%*. Though slightly lower, it was helpful in giving valuable insights especially when comparing the rates of false positive and false negative readings through its confusion matrix.

The Random Forest algorithm provides a two-class confusion matrix quantitative evaluation of the model's performance. For class 0, the matrix shows a large number of true negatives and true positives which means this category is correctly classified by the matrix. On the other hand, for class 1, the matrix shows once more a high number of correctly classified both negatives and positives. The number of false positive and false negative cases across both classes is very low, showing that the model is competent in determining the right class with confidence. These findings suggest that this model performs well and has a good predictive capacity for the given classification task.

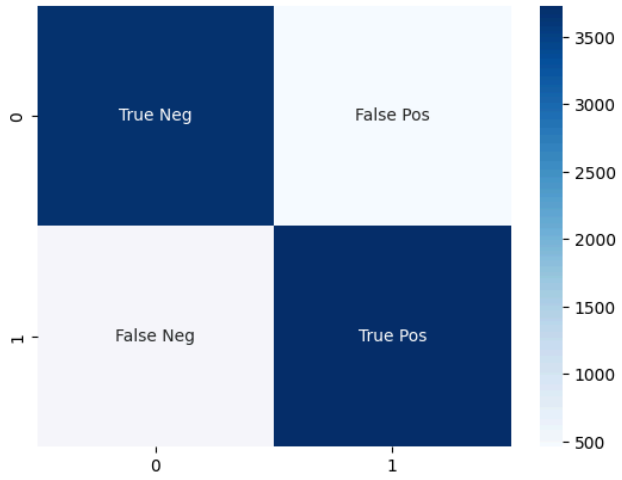

Figure 6: Confusion Matrix - Random Forest (Class 1)

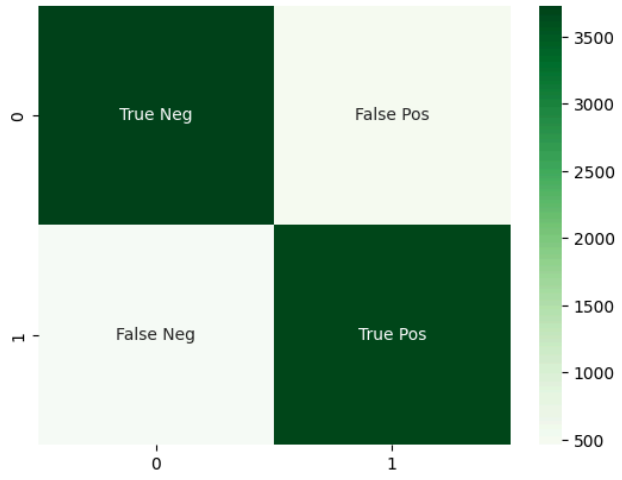

Figure 7: Confusion Matrix - Random Forest (Class 2)

The Naive Bayes confusion matrices for the first and second classes show how well this model can predict. For the first class, we find a significant number of true negatives and true positives with relatively few false ones in each case resulting in very good performance for our model on this class. Similarly, the second class shows a high true negative count and strong positive rate which validates the model. These matrices show that the Naive Bayes classifier has a high accuracy rate in classification of this dataset.

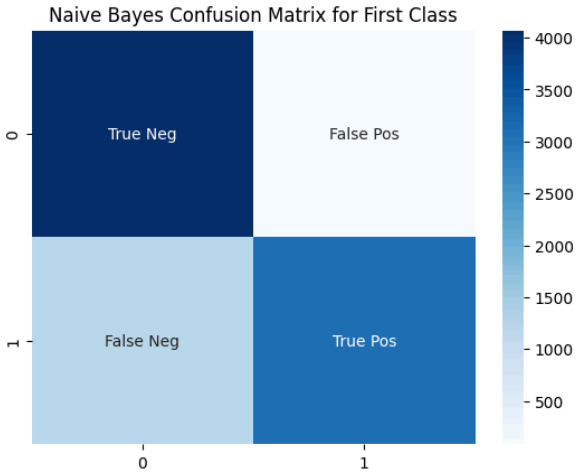

Figure 8: Confusion Matrix - Naive Bayes (Class 1)

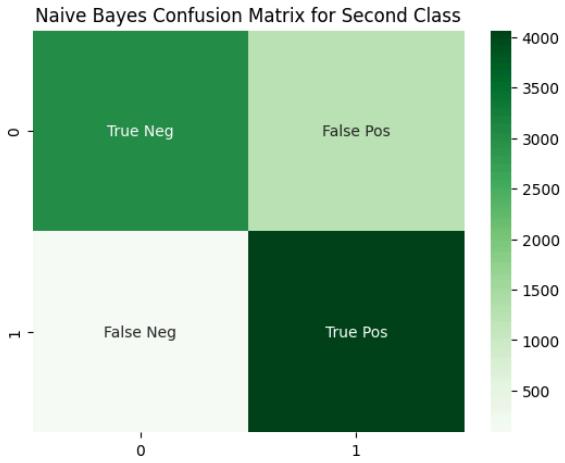

Figure 9: Confusion Matrix - Naive Bayes (Class 2)

When comparing the Random Forest and Naive Bayes confusion matrices, we focus on balance between true positives, true negatives, false positive and false negative . The Random Forest algorithm is relatively accurate as the true classifications are balanced between both classes. Slightly less accurate, Naive Bayes still can hold a decent number of true classifications. Overall,

the Random Forest classifier demonstrates a slight superiority in accuracy compared to computing models for this specific dataset and classification task.

But also, we need to mention that these results are not always the same, meaning that applied on different datasets, different algorithms give different results. For example, in (Tiwari et al, 2021) paper, the Random Forest algorithm did not provide better results than the Naïve Bayes algorithm. In fact, the paper reports that the Naïve Bayes algorithm had a higher accuracy (87.13%) than the Random Forest algorithm (51.35%).

CONCLUSION

Overall, this paper has successfully proven the suitability of sentiment analysis for mental health prediction. Data preprocessing is an important step, which made it possible to have the Random Forest Classifier be a pretty reliable tool with minimal classification errors. Though the Naive Bayes classifier did not achieve same or higher accuracy, it provided useful probabilistic insights and was very helpful in particular cases.

The results indicate that no one model is better than the other, but rather a given model should be selected based on a specific dataset. This paper makes for a convincing argument that sentiment analysis could be used on a much larger scale in monitoring mental health, and shows its potential to enable proactive support of good mental state. As the field evolves, these models could play a vital role in creating real-time monitoring tools on digital platforms that have promisingly pointed towards more effective mental health support systems.

REFERENCES

[1] Zucco, C., Calabrese, B., & Cannataro, M. (2017). Sentiment Analysis and Affective Computing for Depression Monitoring. CR Data Analytics. Department of Medical and Surgical Sciences, University "Magna Græcia".

[2] Tiwari, P. K., et al. (2021). A Study on Sentiment Analysis of Mental Illness Using Machine Learning Techniques. IOP Conference Series: Materials Science and Engineering, 1099, 012043.

[3] Calvo, R. A., Milne, D. N., Sazzad Hussain, M., & Christensen, H. (2017). Natural language processing in mental health applications using non-clinical texts. Cambridge University Press 2017.