Evaluating Student Depression Detection from Social Media Post Using Multinomial Naïve Bayes Text Vectorization and Randomizedsearchcv

Roman B. Villones, John Joshua E. Mendoza, Rhayz Steven Kyle P. Bautista*, Alfred Brian C. Bautista, Julius P. Claour

Graduate School, La Consolacion University Philippines, Malolos, Bulacan, Philippines

***Corresponding Author**

DOI : https://doi.org/10.51583/IJLTEMAS.2025.1411000089

Received: 06 December 2025; Accepted: 11 December 2025; Published: 18 December 2025

ABSTRACT

Purpose - The complexities of recognizing student depression make it a promising area for exploring the effectiveness of various text vectorization techniques such as `CountVectorizer`,

`TfidfVectorizer`, and `HashingVectorizer` in machine learning applications. These techniques can help identify patterns and symptoms by analyzing large datasets like social media activity which may reveal indicators of depression that traditional diagnostic methods might miss.

Method - In recognizing student depression through machine learning, the methodology follows the Knowledge Discovery in Databases (KDD) process which guides each stage of data handling and analysis.

Results - Based on the result of the evaluation of different text vectorization techniques, the `CountVectorizer` achieved the highest classification accuracy at 87%, ranking it first. `TfidfVectorizer` follows closely with an accuracy of 86%, ranking second. Lastly, `HashingVectorizer` achieved a lower accuracy of 77%, placing it third in performance.

Conclusion – The study reaffirms that traditional vectorization methods like `CountVectorizer` and `TfidfVectorizer` remain highly effective for text classification tasks and that method selection should be guided by the specific needs of the application and characteristics of the data.

Recommendation - Future work will center on combining these techniques or probing more advanced embedding methods which better capture deeper semantic relations. The generalization of such vectorizers on other datasets and domains will be further illustrated in detail by their robustness and performance.

Keywords – Depression, Text Vectorization, Count Vectorizer, Tfidf Vectorizer, Hashing Vectorizer.

INTRODUCTION

Mental health has emerged as a critical component of global well-being, particularly among vulnerable populations such as students. The increasing prevalence of depression among youth poses significant challenges to educational institutions and society at large. For instance, research conducted in the United Kingdom revealed that the proportion of undergraduate students reporting mental health difficulties has nearly tripled over the past seven years, rising from 6% in 2016/17 to 18% in 2023/24 (Havergal, 2024). Similarly, a study from Saudi Arabia found that 81.5% of surveyed university students reported symptoms of depression, highlighting the global nature of this crisis (Alzahrani et al., 2024). Depression is perhaps one of the most seen disorders in psychiatry and affects the practice of specialists and general practitioners equally; therefore, it is very important to understand the basics of the subject (Paykel, 2008).

Machine Learning methods are successful in finding good solutions for psychological problems of Facebook users (Islam et al., 2018). However, there are other factors like privacy, ethical data usage, and quality of mental health responses, which must be considered to ensure that AI-driven insights really benefit users while not risking their private data. In addition, experimental findings point toward the effectiveness of the presented method at good reliability identifying symptoms of depression in the posts of social networks-even without having the explicit use of such words as 'depression' or 'diagnose' in the training dataset-and on testing datasets with which it hasn't previously been exposed and that it would have absolutely no notion about (Chiong, 2021). However, both linguistically as well as culturally diverse deployment, as well as user information confidentiality issues, continue to have significant importance as considerations when it comes to the responsibly applied use of such a method. On the other hand, as the fast growth of social media sites continues to occur, it has emerged as a 'diary' in which users state their feelings and thoughts. Many studies have been performed to identify symptoms of depression in the posts of the users through machine learning algorithms (Govindasamy and Palanichamy, 2021). As social media continues to growth of digital "diaries", their role in mental health diagnostics will likely grow, both offering opportunities and challenges to the field.

The study aims to contribute to this global agenda by exploring the use of machine learning in the early identification of depressive tendencies through textual data of social media. By focusing on student populations, recognizing that mental health is integral to effective learning and academic success. The deployment of accessible, automated detection systems has the potential to reduce disparities in mental health support. Furthermore, through the integration of machine learning and natural language processing, this study underscores the interdisciplinary approach required to address complex social and health challenges. It positions technology not only as a tool for innovation but as a mechanism for promoting sustainable, inclusive, and equitable well-being among learners.

LITERATURE REVIEW

Diagnosis of depression is, for the most part, done by means of doctor-patient conversation and assessment scales. However, these are not without their downsides: denial by patients, lack of sensitivity, subjective judgment, and possibly incorrect findings (Li et al., 2019). The application of technology as a supplement or enhancement of the traditional approach in the diagnosis of mental illnesses would make this task more precise and accessible. Bias, denial, and lack of sensitivity might then be alleviated when diagnosing depression.

Platforms like Facebook, Twitter, and Instagram have transformed society by fostering unprecedented levels of connection and allowing individuals to showcase their digital identities. While these platforms offer numerous benefits, they also come with notable drawbacks (AlSagri and Ykhlef, 2020). Promoting balanced usage and fostering supportive online environments may be key steps in mitigating these adverse effects, enabling social media to serve as a positive influence on mental health.

Advancements in technology have led to the development of AI-driven methods aimed at making machines emotionally aware, capable of identifying human emotions. For instance, text- based emotion recognition, such as sentiment analysis of tweets and social media posts, can gauge users' moods and emotions (Joshi and Kanoongo, 2022). However, using AI in this sensitive area demands careful attention to privacy, accuracy, and ethical considerations. Ensuring that these tools provide appropriate, reliable support while protecting user data is critical to their responsible use and acceptance in mental health care.

We could conclude the mental status of the user, especially concerning depression, by trying various algorithms of machine learning techniques on the data coming through social media. One of the most efficient methods to identify depression is by analyzing texts showing negative sentiments (Dabhane and Chawan, 2020). By applying NLP and sentiment analysis techniques, researchers and mental health professionals can understand the emotional state of users and flag people who may need support.

Machine learning integration into the medical field contributes to the more accurate, precise, and analytical diagnostics of patients, thereby reducing uninteresting work. There has been mounting evidence that such machine learning can identify serious mental disorders like depression (Ashraf et al., 2020). Machine learning adds accuracy and thus efficiency into diagnostic procedures, thereby making the approach more reliable and less sensitive to human error.

The algorithm checks for emotional readings in the input text from the user and then decides if there are signs of depression. Several classification algorithms have been used in the detection model, such as K-Nearest Neighbors, Naïve Bayes, Decision Tree, and Random Forest, so we can find out which algorithm gives the highest accuracy (Sudha et al., 2020). While algorithmic accuracy is key, so is interpretability in such a way that the result makes sense and can be taken forward by mental health professionals.

The Support Vector Machine and Naive Bayes classifiers have been used in the class prediction model. Grading of the results uses the key classification metric: F1-score, accuracy, and confusion matrix (Deshpande and Rao, 2017). Although these metrics are very useful, the interpretation in the context of the specific application is critical. For example, in mental health scenarios, the importance of recall over precision might be such that false negatives must be avoided so that patients suffering from depression are detected accurately at the cost of some loss in precision. A balanced approach to model evaluation and continuous refinement based on these metrics will thus be key to developing effective and reliable predictive tools in mental health.

METHODOLOGY

Knowledge Discovery in Databases (KDD Process)

Knowledge Discovery in Databases or KDD is the whole process of extracting knowledge from databases. This process involves evaluation and possibly interpretation of patterns for what can be considered as knowledge. It also encompasses encoding methods, preprocessing, sampling, and data projections before actual mining of data (Fayyad et al, 1996).

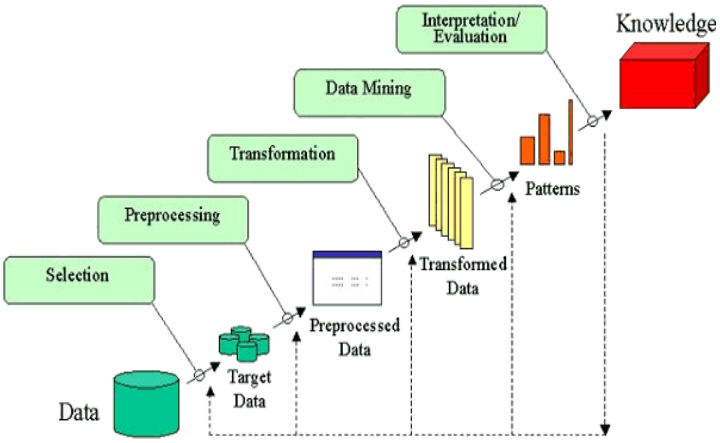

Figure 1. Knowledge Discovery in Databases Figure 1 shows the evaluation and interpretation of patterns must have a judgmental aspect to it, with the proper context, to distinguish between mere data and actionable knowledge.

Selection

In this study, we used a dataset from an open-source platform particularly in Reddit social media student posts which means it’s freely available for anyone to access, use, and even modify under an open license. Open-source datasets are often hosted on platforms from Kaggle that support open science and collaboration.

CSV files are used to load data into a Data Frame for data analysis. We are working with a dataset related to depression from Reddit posts, which might be useful for sentiment analysis, natural language processing, or mental health research.

Preprocessing

Since we only select the columns for `body` and `label`, we are essentially focusing on what is most relevant in our data. In this instance, `body` likely represents the content of the text of posts on Reddit, while `label` likely represents the classification-in this case, depressed or not depressed. This may be common in any kind of Natural Language Processing research or any kind of student mental health study that basically analyzes text and corresponding classification for building models of detecting depression.

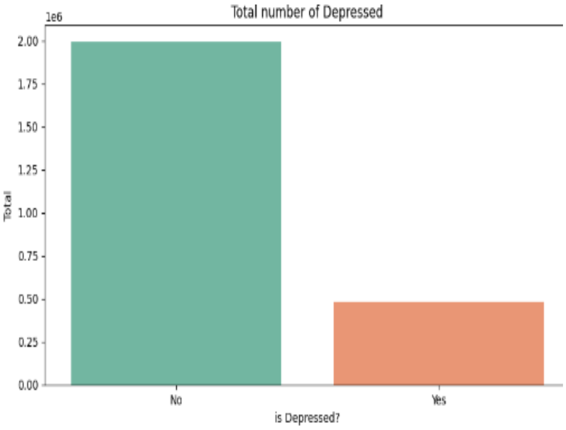

Figure 2. Total Count of Student Depressed Post

Figure 2 shows that we are going through the head and tail of the dataset (2,470,777). This should confirm that data was loaded correctly, that columns `body` and `label` contain content and format as expected and that nothing obviously is amiss, for example missing or poorly formatted data. We performed sentiment analysis on body text to understand themes posts with the label depression refer to. We might develop a classifier that makes predictions based on the `body` content to contribute toward research in detecting student mental health issues from online discussions. This kind of data selection and inspection is an essential first step in any data-driven research study that wants to ensure we're working with the right subset of data.

Transformation

By removing rows with missing values, the analysis ensures the dataset is complete. This reduces the likelihood of introducing biases or errors that could affect the model's accuracy and reliability, leading to more trustworthy conclusions about patterns in depressive language. The code balances the dataset by extracting an equal number of records from both classes, i.e., posts labeled as `No` and

`Yes`. This balance is crucial in research involving classification tasks (such as distinguishing depressive posts) because an imbalanced dataset can lead to a biased model that favors the majority class. Here, by equalizing the classes, the model can better generalize and produce fairer, more accurate predictions for both classes.

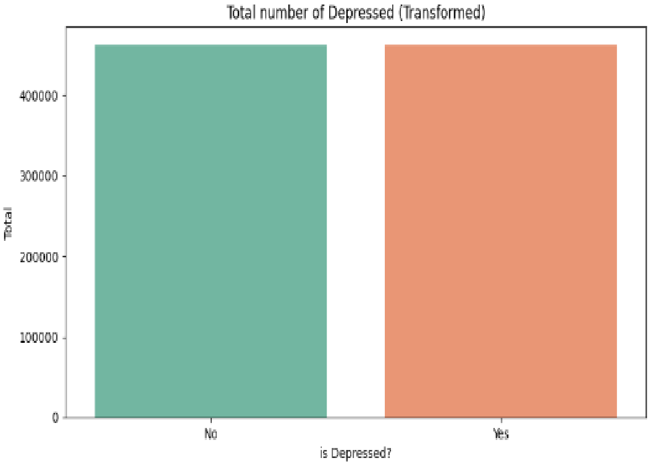

Figure 3. Transformed Dataset

Figure 3 shows that by specifying the desired number of records (461,744) per class, the code provides a substantial and balanced sample for analysis, which is critical in drawing statistically significant and generalizable conclusions. This large sample size also enhances the robustness of the findings, making them more representative of broader trends in the data. And it helps ensure that the data is high-quality and unbiased, forming a solid foundation for subsequent analysis or machine learning model training. The balanced dataset enhances the model's capability to learn effectively from both classes, improving the reliability and interpretability of any predictive models developed from this research.

Data Mining

Naïve Bayes Algorithm

Among these popular algorithms of data mining, Naïve Bayes is one where one of the characteristics is also the efficiency of this algorithm due to the assumption about independence of attributes (Chen et al., 2020). Yet, in many real datasets, this assumption gets violated. To overcome this weakness, several strategies have been applied, and one such strategy is attribute selection, but conventional methods of attribute selection in Naïve Bayes also have heavy computational overhead. In addition, it decreases the learning process due to the independence of features conditional on the class. Yet this is often unrealistic very much (Rish, 2001). Despite that fact, in practice, Naïve Bayes still often equals and even outperforms much more complicated classifiers.

Multinomial Naive Bayes

Automatic text classification, also known as text categorization, is a subdiscipline of machine learning that is growing in importance with the rapid increase in electronically stored text data. It is a method of supervised learning in which a new document is classified into one or more predefined categories by assigning it one or more labels from a given set of categories. The learning algorithm applied to train the system accurately labels training documents. The effectiveness of automatic text classification is strongly dependent on the quality and representativeness of the training data (Kibriya et al., 2005). The effectiveness of automatic text classification is strongly dependent on the quality and representativeness of the training data. Poorly labeled or biased datasets are likely to result in false classifications, which can seriously reduce the utility of the model in real-world applications.

CountVectorizer

`CountVectorizer` is perhaps one of the most common methods used in NLP that transforms text documents into numerical vectors. It is a method that generates a sparse matrix where each column is a unique word within the entire text corpus, and each row is an individual document. The values in this matrix are the frequency of occurrence of each word in a document. This technique is essential for text classification since it makes it possible to process textual data in terms of numbers. `CountVectorizer` aids in the processing and analysis of inherently unstructured text by converting textual information into numerical vectors (Daiv et al., 2020). Transformation is crucial for enabling multiple machine learning tasks, especially the ability to classify texts, wherein algorithms will learn and make predictions based on structured numerical inputs.

TfidfVectorizer

Another technique widely used in NLP to convert text into numerical representations is Term Frequency-Inverse Document Frequency Vectorizer, or `TfidfVectorizer`. Unlike `CountVectorizer`, which counts the frequencies of only words, `TfidfVectorizer` considers how important a word is relative to the entire document set. It assigns higher weights to words that often appear in a single document but not across many documents. This effectively means the effect of word tokens of high frequency low meanings, and makes other very informative words, or whose co- occurrences reflect in a higher density way (Daiv et al., 2020). This section will point to some of the advances made in text-representative techniques, where instead of considering how often these word tokens appear in one's given document, they also count how often they occurred throughout the data that leads in emphasizing more informed word tokens.

HashingVectorizer

Hashing vectorizer (HV) is a feature extraction method in NLP for text, which maps text documents into matrices with each entry representing the frequency of individual tokens. This technique permits every word to be treated uniquely because it uses the hash function to map words as unique indices; it finds advantage about managing large datasets because of its scalability (Roshan et al., 2023). Even though hash collisions can occur in which several words share the same index, the performance impact is negligible in general, and this makes HV a promising candidate for managing high- dimensional text data of massive datasets.

RESULTS

Interpretation/ Evaluation

With the increase of machine learning algorithms into a wide variety of applications, the need for suitable measurement and evaluation is significantly amplified when classifiers are to be used in real-life situations. Hence, what type of evaluation metrics will most appropriately evaluate the performance of binary, multi-class, and multi- label classifiers is very important (Naidu et al., 2023). The algorithm of the future will therefore rely on accuracy and reliability in the overall decision- making process as technology integrates into real procedures, hence depending on proper evaluation practices using Confusion Matrix, Precision, Recall, F1-score/ Model Accuracy to get the Classification Report using different Text Vectorization.

CountVectorizer Result

Depressed	Precision	Recall	F1- Score	Support
\`No\`	0.92	0.81	0.86	138524
\`Yes\`	0.83	0.93	0.88	138523
Accura cy			0.87	277047
Macro Avg.	0.87	0.87	0.87	277047
Weigth ed Avg.	0.87	0.87	0.87	277047

Table 1. Classification Report for CountVectorizer Table 1 shows the overall accuracy of the

\`CountVectorizer\` is 0.87, meaning it got 87% of all instances correct. For both classes, the macro average, or unweighted average, for precision, recall, and F1-score, was 0.87, indicating balanced performance. For the weighted average, considering the support or frequency of each class, precision, recall, and F1-score all equaled 0.87. This shows the model performs well with consistent performance across both classes at a high level of accuracy and balanced classification.

TfidfVectorizer Result

Depressed	Precision	Recall	F1- Score	Support
\`No\`	0.94	0.76	0.84	138524
\`Yes\`	0.80	0.95	0.87	138523
Accuracy			0.86	277047
Macro Avg.	0.87	0.86	0.85	277047
Weigthed Avg.	0.87	0.86	0.85	277047

Table 2 shows the total accuracy of the model or proportion of all correct predictions is 0.86, which means the model is 86% accurate. The macro average gives the unweighted mean of each metric

DISCUSSION

In this paper, we evaluate different vectorization techniques for text preparation such as across classes; this results in a precision of 0.87, recall of 0.86, and F1-score of 0.85. The weighted average, considering support, or the number of instances in each class, is the same for precision, recall, and F1-score, which are roughly around 0.87, 0.86, and 0.85, respectively. This shows that it is a very balanced performance for both the classes.

HashingVectorizer Result

Depressed	Precision	Recall	F1- Score	Support
\`No\`	0.97	0.57	0.72	138524
\`Yes\`	0.69	0.98	0.81	138523
Accuracy			0.77	277047
Macro Avg.	0.83	0.77	0.76	277047
Veigthed Avg.	0.83	0.77	0.76	277047

Table 3. Classification Report for HashingVectorizer

Table 3 shows the classification report talks about the performance of a \`HashingVectorizer\` in the case of binary classification. So, the accuracy here is 77%. In other words, it was correct for 77%. Macro and weighted averages for precision, recall, and F1-score are also available that are around 0.83, 0.77, and 0.76 respectively showing that the model's performance is quite balanced and proper over both classes.

Ranking of Classification Report

Rank	Model	Scores
1	CountVectorizer	87%
2	TfidfVectorizer	86%
3	HashingVectorizer	77%

Table 4. Rank per Text Vectorization

Table 4 shows the comparison of the performances of three different text vectorization techniques: \`CountVectorizer\`, \`TfidfVectorizer\`, and \`HashingVectorizer\` used to transform text data prior to classification. The performance of each vectorization method has been assessed by using accuracy metric from the classification report, measuring the overall percentage of correct predictions.

\`CountVectorizer\`, \`TfidfVectorizer\` and \`HashingVectorizer\`.We will measure the performance for every vectorization technique using accuracy from the classification report, a simple measure of how good each model is getting at instances classified in a dataset.

The results also show that \`CountVectorizer\` was the best performing, 87% accuracy, followed closely by \`TfidfVectorizer\` at 86%. Both methods led \`HashingVectorizer\` with a far lower accuracy rate at only 77%. The choice of vectorization method thus apparently has an important implication for classification performance.`CountVectorizer` and `TfidfVectorizer` make word frequency and term importance explicit and may provide more context and readily interpretable representations of text data than `HashingVectorizer`.

This is because its technique for reducing dimensionality is somewhat efficient but with a side effect of information loss from hash collisions. Although it is less interpretable, because it does not store vocabulary, `HashingVectorizer` also is limited for some applications requiring nuanced representations of words. Thus, to understand the strength and limitations of various vectorization techniques and select the proper vectorization techniques for any task with corresponding computational resource requirements, it is significant to recognize that the correct selection of the vectorization techniques plays an important role in making this happen.

More generally, the results agree with other literature suggesting the usefulness of `TfidfVectorizer` weighting for tasks where semantic understanding is important. In fact, although both algorithms provide good accuracy, a very small edge for `CountVectorizer` suggests that frequency-based features may be more informative than weighted term frequencies for this dataset. But a performance difference of only 1% between

`CountVectorizer` and `TfidfVectorizer` means that one may just as well use either one, depending on what your model's objectives are.

CONCLUSIONS AND RECOMMENDATIONS

In conclusion, the study confirms the fact that traditional methods such as `CountVectorizer` and `TfidfVectorizer` are still well effective for text classification and that the choice of the method should be made considering the specific requirements of the application and nature of the data.

Future work will center on combining these techniques or probing more advanced embedding methods which better capture deeper semantic relations. The generalization of such vectorizers on other datasets and domains will be further illustrated in detail by their robustness and performance.

ACKNOWLEDGMENT

We would like to thank Kaggle.com for sharing the publicly available datasets for this study. Their platform made it possible for me to carry out the analysis and evaluation of the machine learning algorithms presented here. We are also grateful for the support provided by the University Research Department in helping with their support throughout this research.

Funding

This study did not get funding from any institution.

Declarations

Conflict of Interest

The author declared that there is no conflict of interest.

Informed Consent

Since the datasets used in this research were obtained from Kaggle.com, a public dataset sharing website, informed consent does not apply. The datasets are public, and the terms of use of Kaggle state that data uploaded to the site can be used by the research community.

Ethics Approval

Since the data used here is from Kaggle.com, ethics approval is not required for their use since datasets on this public platform are made available to research communities, and this dataset does not contain sensitive information.

REFERENCES

1. Acheampong, F. A., Wenyu, C., & Nunoo-Mensah, H. (2020). Text-based emotion detection: Advances, challenges, and opportunities. Engineering Reports, 2(7), e12189.
2. Al Asad, N., Pranto, M. A. M., Afreen, S., & Islam, M. M. (2019, November). Depression detection by analyzing social media posts of user. In 2019 IEEE international conference on signal processing, information, communication & systems (SPICSCON) (pp. 13-17). IEEE.
3. AlSagri H. S. & Ykhlef M. (2020). Machine learning-based approach for depression detection in twitter using content and activity features. IEICE Transactions on Information and Systems, 103(8), 1825-1832.
4. Alzahrani, A., Alosaimi, N., Alzahrani, M., Aljohani, N., & Alshehri, S. (2024). Prevalence and associated factors of depression among university students in Saudi Arabia: A cross-sectional study. Frontiers in Public Health, 12, 1441695. https://doi.org/10.3389/fpubh.2024.144169 5
5. Ashraf A., Gunawan T. S., Riza B. S., Haryanto E. V., & Janin Z. (2020). On the review of image and video-based depression detection using machine learning. Indonesian Journal of Electrical Engineering and Computer Science (IJEECS), 19(3), 1677-1684
6. Chatterjee, R., Gupta, R. K., & Gupta, B. (2021). Depression detection from social media posts using multinomial naive theorem. In IOP conference series: materials science and engineering (Vol. 1022, No. 1, p. 012095). IOP Publishing.
7. Chen S., Webb G. I., Liu L., & Ma X. (2020). A novel selective naïve Bayes algorithm. Knowledge-Based Systems, 192, 105361.
8. Chiong R., Budhi G. S., Dhakal S., & Chiong F. (2021). A textual-based featuring approach for depression detection using machine learning classifiers and social media texts. Computers in Biology and Medicine, 135, 104499.
9. Dabhane S. & Chawan, P. M. (2020). Depression detection on social media using machine learning techniques: a survey. Int. Res. J. Eng. Technol, 7(11), 97-100.
10. Daiv K., Lachake M., Jagtap P., Dhariwal S., & Gutte V. (2020). An approach to detect fake reviews based on logistic regression using review-centric features. Int. Res. J. Eng. Technol.(IRJET), 7(06), 2107-2112.
11. Deshpande M. & Rao V. (2017, December). Depression detection using emotion artificial intelligence. In 2017 international conference on intelligent sustainable systems (iciss) (pp. 858-862). IEEE.
12. Fayyad U., Piatetsky-Shapiro G., & Smyth P. (1996). From Data Mining to Knowledge Discovery in Databases. AI Magazine, 17(3), 37. https://doi.org/10.1609/aimag.v17i3.1230 Govindasamy K. A. & Palanichamy N. (2021, May).
13. Depression detection using machine learning techniques on twitter data. In 2021 5th international conference on intelligent computing and control systems (ICICCS) (pp. 960-966). IEEE.
14. Havergal, C. (2024, February 22). Number of UK students reporting mental health difficulties triples. Times Higher Education. Retrieved from https://www.timeshighereducation.com/ne ws/number-uk-students-reporting-mental- health-difficulties-triples
15. Islam M. R., Kabir M. A., Ahmed A., Kamal A. R. M., Wang H., & Ulhaq A. (2018). Depression detection from social network data using machine learning techniques. Health information science and systems, 6, 1-12.
16. Joshi M. L. & Kanoongo N. (2022). Depression detection using emotional artificial intelligence and machine learning: A closer review. Materials Today: Proceedings, 58, 217-226.
17. Kibriya A. M., Frank E., Pfahringer B., & Holmes G. (2005). Multinomial naive bayes for text categorization revisited. In AI 2004:
18. Advances in Artificial Intelligence: 17th Australian Joint Conference on Artificial Intelligence, Cairns, Australia, December 4-6, 2004. Proceedings 17 (pp. 488-499). Springer Berlin Heidelberg
19. Li X., Zhang X., Zhu J., Mao W., Sun S., Wang Z., Xia C., & Hu B. (2019). Depression recognition using machine learning methods with different feature generation strategies. Artificial intelligence in medicine, 99, 101696.
20. Naidu G., Zuva T., & Sibanda E. M. (2023, April). A review of evaluation metrics in machine learning algorithms. In Computer Science On-line Conference (pp. 15-25). Cham: Springer International Publishing.

21. Paykel, E. S. (2008). Basic concepts of depression. Dialogues in clinical neuroscience, 10(3), 279-289.
22. Rish I. (2001, August). An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, No. 22, pp. 41-46).
23. Roshan R., Bhacho I. A., & Zai S. (2023). Comparative Analysis of TF–IDF and Hashing Vectorizer for Fake News Detection in Sindhi: A Machine Learning and Deep Learning Approach. Engineering Proceedings, 46(1), 5.
24. Sudha K., Sreemathi S., Nathiya B., & RahiniPriya D. (2020). Depression detection using machine learning. In International Journal of Research and Advanced Development. on AICTE Sponsored International Conference on Data Science & Big Data Analytics for Sustainability.

Author’s Biography

Roman B. Villones is currently working as an Assistant Professor in the College of Engineering and information Sciences of Trinity University of Asia. He is also a Research Coordinator under the University Research and Development Center at the same university. He holds a Master in Information Technology and he is also a candidate of Doctor in Information Technology. His field of specialization includes Software Engineering and Machine Learning.