# Depression Detection By Analyzing Social Media Text Using Natural Language Processing

Abid Hossain Ashik
ID: 20201162
*Department of Computer Science and Engineering, BRAC University*
*abid.hossain.ashik@g.bracu.ac.bd*

Md. Yasin
ID: 20201157
*Department of Computer Science and Engineering, BRAC University*
*md.yasin1@g.bracu.ac.bd*

Zahin Zaima
ID: 20201147
*Department of Computer Science and Engineering, BRAC University*
*zahin.zaima@g.bracu.ac.bd*

Ahmed Wasi Bin Faruque
ID: 20101352
*Department of Computer Science and Engineering, BRAC University*
*ahmed.wasi.bin.faruque@g.bracu.ac.bd*

***Abstract*—The goal of this research is to create a machine learning model for spotting suicidal behavior on social media platforms online. The prevention and early detection of suicidal behavior are extremely important since suicide is a serious public health concern. Social media is now widely used, making it a viable tool for identifying and preventing suicide behavior. The suggested machine learning model analyzes text data from social media posts using methods of natural language processing to spot potential suicidal behavior in people. A dataset of social media posts containing both suicidal and non-suicidal behaviour are used to train the model. Precision, recall, and F1-score are just a few of the performance indicators that are used to assess the model's correctness.* [2]**

## Introduction

With over 700,000 fatalities from suicide each year worldwide, it is a significant public health issue. Strategies for preventing suicide have centered on spotting at-risk persons and taking action to stop suicidal behavior. Social networking sites have become a possible resource for spotting and addressing suicidal behavior in recent years. Social media sites offer a plethora of information that can be evaluated to spot possible suicidal behavior indicators, such as changes in language and behavior patterns.

The suggested machine learning model analyzes text data from social media posts using methods of natural language processing to spot potential suicidal behavior in people. A dataset of social media posts containing both suicidal and non-suicidal behaviour are used to train the model. The model can be used to automatically identify people who may be at risk and highlight them for additional assessment and assistance. Suicides can be avoided and effective interventions can be made when people at risk for suicide are identified early.

The objective of this project is to create a machine learning model that can precisely identify suicidal behavior in social media posts. We will examine numerous approaches to natural language processing, such as feature extraction, text cleaning, and machine learning algorithms. [1]

## Motivation

Almost every week in and week out, we come across the news of suicides, suicidal attempts in the social medias. This is a very concerning issue as we are losing so many bright lives just because there is no one to help them out of depression. Therefore, if we can predict whether a person is suicidal beforehand from their social media posts, we might be able to reach them out and try to help them and thus it might save a lot of lives. This thought has motivated us to build this project.

## Methodology

Due to the massive advancement of technology, now researchers can work with linguistic analysis. Now, it is possible to extract a pattern from a text which helps to identify the sentiment of the text. There are lots of techniques to analyze text. We have selected natural language processing(NLP) to identify symptoms of depression from social media posts.

***Data Acquisition***

Data collection is the first step. We have used a secondary dataset which is collected from kaggle[10]. There are two columns, one contains the posts and the other contains the decision. The data is collected from "SuicideWatch" and "depression" subreddits of the Reddit platform.

***Data Pre-processing***

The collected data includes lots of errors and unnecessary words which might cause problems during the model development. Therefore, these words must be removed. The NLTK package of python helps to do this easily. Text processing steps we have used.

- ☐ Symbol removal: Elimination of unnecessary meaningless symbols.
- ☐ Lower casing: Lower case each post to minimize complications.
- ☐ Tokenization: It creates tokens as words from the sentences.
- ☐ Grammatical correction: It corrects spelling mistakes.
- ☐ Stemming & Lemmatization: It converts each word to its root form.

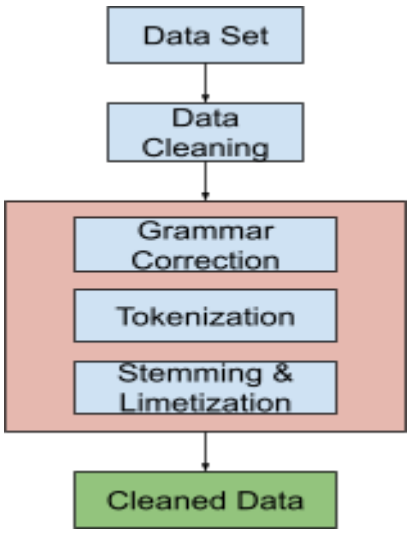



*Figure 1: Procedure for pre-processing the dataset[11]*

**Vectorization & Train-Test Split**

By applying the above mentioned technique, the initial dataset will be cleaned. But we all know that machines can not understand text data, they only understand zeros and ones. Therefore, the text needs to be converted to vector form so that the machine can understand it. There are lots of techniques to convert text to vector form. We have used the bag of word technique which is also known as countvectorizer. In this technique, all the unique words are collected to create a list which is called grammar. Compared with this grammar all posts are converted to vector form.

After this we have splitted the total dataset to testset and trainset. Using trainset we have trained all the models and using the testset we have tested the accuracy of the model.

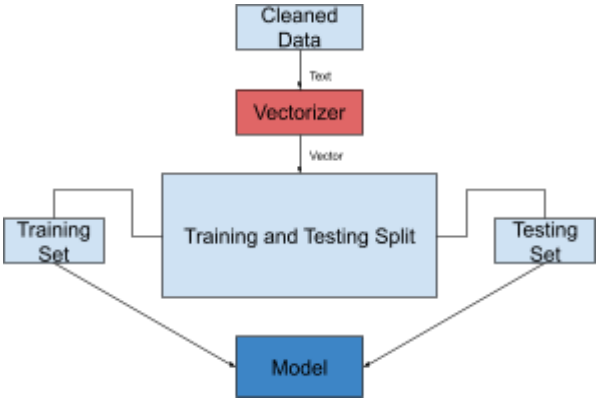



*Figure 2: Procedure for convert text to vector and splitting to train-test set*

***Model Description***

We have used 6 different models.The description is given below about how these models work.

***Logistic Regression***

Logistic Regression is a common machine learning algorithm used for text classification which belongs to the class of Generalized Linear Models. The probabilities describing outcome of a trial is modeled using logistic regression[8]. Basically, it models the probability of an input and depending on the probability this model decides the output.

***K-Nearest Neighbour (KNN)***

The K-Nearest Neighbor Algorithm places the new instance in the category that is most similar to the existing categories on the presumption that the new case and the existing cases are comparable.After storing all the previous data, a new data point is categorized using the K-NN algorithm based on similarity. This indicates that new data can be reliably and quickly categorized using the K-NN approach.[8]

***Decision Tree***

The Decision Tree Classifier is a well-known machine learning technique for classification problems, and it serves as the model in our project[9]. The fundamental concept is to segment the dataset into smaller groups while also progressively building the related tree. This works with both numerical and category data. Gini index and information gain parameter can both be used to choose which

characteristic will be utilized to further divide the dataset. When using the Gini index, the decision tree is referred to as a CART (classification and regression tree), and when using information gain, it is referred to as an ID3. We used the CART method [13].

***Random Forest***

It is a collection of decision tree techniques that may be applied to both regression and classification. More trees often translate into improved performance and efficiency in this method. Extract a sample set of data points from a specified training set using the bootstrap method. Create a decision tree based on the results of the previous step after this. Applying the first two steps will give us the number of trees (default is 100). Every tree that is built will cast a vote for a data point. Calculate the decision tree classifiers' overall majority voting. [1]

***Naive Bayes***

Naive Bayes is a classification algorithm based on Bayes' Theorem with a 'naive' assumption that features of the dataset are independent of Each other[6]. Naive Bayes is a probabilistic algorithm that can be utilized for classification tasks. It works by calculating the probability of a given data point belonging to each possible class, and then selecting the class with the highest probability as the predicted class for that data point. Usually when we give some data to train our model then the Naive Bayes method entirely makes a histogram of all the words that occur in the sentiment text. The histogram is used to calculate the probabilities of seeing each word in the text and trained to see the sentiment output of the train data. Using the learned probabilities, the algorithm makes predictions on the testing set by calculating the probability of each class given the features of the data point. The class with the highest probability is selected as the predicted class for that data point. Naive Bayes is a simple yet powerful algorithm that can work well with large datasets and high-dimensional feature spaces. However, it assumes that the features are independent of each other. In our model, we use two types of Naive Bayes. (1)Gaussian Naive Bayes (2) Multinomial Naive Bayes. Gaussian Naive Bayes is a specific type of Naive Bayes algorithm that assumes the features in the data follow a Gaussian (normal) distribution. This algorithm is typically used when the features are continuous variables, such as the length of a sentence or the frequency of a word in a text. Multinomial Naive Bayes, on the other hand, gives more accurate results when the features are discrete variables. But in both cases, it can be used. The difference from the Gaussian Naive Bayes is when Multinomial Naive Bayes is called it adds any numeric value of the count of each of the word's histogram. For this, the probability of any new word is not zero. This approach to overcome the 'zero-frequency problem'[5].

## RESULTS

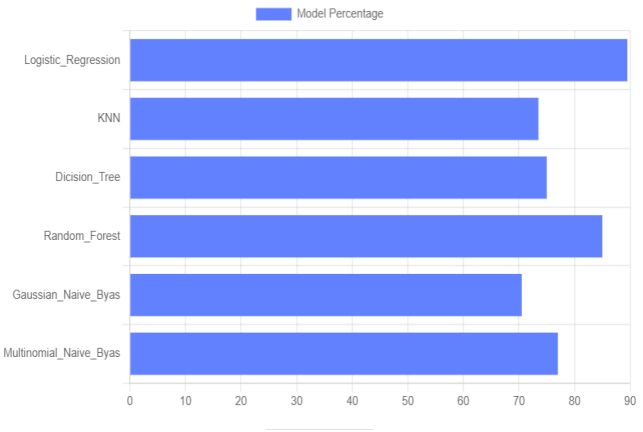



*Figure:3*

We used six models in our project. We can see that the models give different accuracy output for different data sets.If we look at the bar chart it is seen that LR(Logistic Regression) and Random Forest gives us more than 80% accuracy but LR model is giving the maximum accuracy output. Meanwhile we also see that KNN, Decision Tree, Gaussian NB and Multinomial NB give accuracy between 70 to 80 percent (Figure-3).

List of Percentage of different amounts of data

| Model Name | 250 | 500 | 750 | 1000 |
|---|---|---|---|---|
| Logistic Regression | 74% | 87% | 86% | 89.5% |
| KNN | 60% | 74% | 66% | 73.5% |
| Decision Tree | 82% | 87% | 78.6% | 75% |
| Random Forest | 80% | 87% | 79.3% | 85% |
| Gaussian NB | 76% | 66% | 60.7% | 70.5% |
| Multinomial NB | 76% | 77% | 72% | 77% |

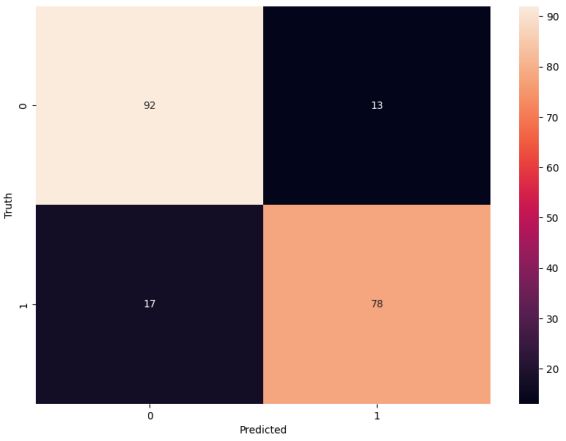
*Table-1*

We can observe that in table-1, as we increase the data, the accuracy of the regression model does not go down. This scenario also seen for KNN, Multinomial NB but in LR, the precision is very high than KNN and Multinomial NB. For others, different accuracies are coming for different data. Since LR gives us higher accuracy, LR's 89% is our project final accuracy of 1000 data points. This will give us higher accuracy for more data as we have more than 200K data. We couldn't all the data's for time consumption.

**Visualization(Confusion Matrix):**

A confusion matrix is a table that is used to define the performance of a classification algorithm. A confusion matrix helps to visualize and summarize the performance of a classification algorithm or model. A classification model gives an accuracy in a percentage. If we want to see the numeric value of how many outputs are right and wrong from testing data in a matrix form, we use a confusion matrix. In our model, the confusion matrix's left to right

diagonal gives me true positive prediction number and right to left diagonal give me false negative & false positive prediction number. Mainly, right to left diagonal give me the numeric value of wrong prediction. As a example of a confusion matrix you apply it in Random Forest model(Figure-4) and Multinomial Naive Bayes model(Figure-5) for 1000 dataset.




a.

*Figure-4*

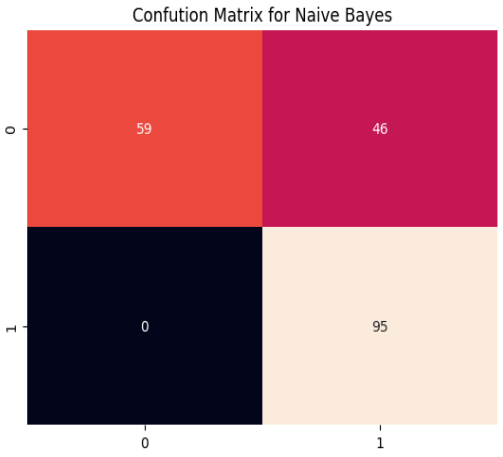



*Figure-5*

## Compare

Several factors, including variations in data preparation, feature engineering, model selection, and hyperparameter tuning, might account for the accuracy gap between the two research. However, one obvious distinction is the application of various text vectorization strategies. Our project employed the CountVectorizer, whereas the original study used the TF-IDF vectorizer. While CountVectorizer is regarded to be quicker and easier than TF-IDF, both vectorization approaches offer benefits and drawbacks. Additionally, CountVectorizer excels on datasets with brief sentences, which might account for the increased accuracy attained in our study.[4]

## Conclusion

This paper defines a depression analysis of whether a person is depressed, based on his tweets on his Twitter profile. To do this we use six different machine learning algorithms or methods. We came to the final conclusion that the Logistic regression is a better choice compared to KNN, Decision Tree, Random Forest, Gaussian Naive Bayes, and Multinomial Naive Bayes for a large amount of data of the depression analysis. It is computationally efficient and has low memory requirements, making it suitable for large datasets. Logistic regression is less prone to overfitting than decision trees and random forests. On the other hand, Naive Bayes ignores the relationship between text words. If high dependency on features the Naive Bayes is inefficient. Similarly, KNN's performance decreases as the number of features increases, which is known as the "curse of dimensionality." This makes it less suitable for high-dimensional data, which is common in many real-world applications. When we run our model for 250,500,750,1000 datasets, we can see the variation and analyze these problems. So the logistic regression model is better for depression analysis. Because logistic regression is a simpler and more interpretable model that can handle high-dimensional data, is less sensitive to irrelevant features, and can provide a probabilistic interpretation of the prediction. However, it is important to note that each algorithm has its own strengths and weaknesses and the choice of algorithm should depend on the specific problem and the characteristics of the data.

## References


[1] Breiman, L., Random forests. Mach. Learn. 45(1):5–32, 2001.

[2] Biddle, P., Roberts, T., & Rosenfeld, J. (2018). Applying machine learning to detect suicidal behavior among college students. Journal of American college health, 66(5), 298-305

[3] Fox, S., & Duggan, M. (2013). Pew Internet & American Life Project. Health Online 2013. Available at: http://www. pewinternet. org/2013/01/15/health-online-2013/

[4] J. Singh and P. Tripathi, "Sentiment analysis of Twitter data by making use of SVM, Random Forest and Decision Tree algorithm," *2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT)*, Bhopal, India, 2021, pp. 193-198, doi: 10.1109/CSNT51715.2021.9509679.

[5] KDnuggets & Chauhan, N. (2022, April 8). *Naïve Bayes Algorithm*. https://www.kdnuggets.com/2020/06/naive-bayes-algorithm-everything.html#:~:text=An%20approach%20to%20overcome%20this,occur%20with%20every%20class%20value.&text=This%20is%20how%20we'll,of%20getting%20a%20zero%20probability.

[6] Level Up Coding. (n.d.). *Classification using Gaussian Naive Bayes from scratch*. https://levelup.gitconnected.com/classification-using-gaussian-naive-bayes-from-scratch-6b8ebe830266

[7] Mergel, I. (2014). Social media in the public sector: A guide to participation, collaboration and transparency in the networked world. John Wiley & Sons.

[8] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830.

[9] Sharma, A. (2021). Machine Learning 101: Decision Tree Algorithm for Classification. *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2021/02/machine-learning-101-decision-tree-algorithm-for-classification/

[10] *Suicide and Depression Detection*. (2021, May 19). Kaggle. https://www.kaggle.com/datasets/nikhileswarkomati/suicide-watch

[11] Tejaswini, V., Babu, K. S., & Sahoo, B. (2022). Depression Detection from Social Media Text Analysis using Natural Language Processing Techniques and Hybrid Deep Learning Model. *ACM Transactions on Asian and Low-resource Language Information Processing*. https://doi.org/10.1145/3569580

[12] World Health Organization. (2020). Suicide Prevention. Retrieved from https://www.who.int/health-topics/suicide#tab=tab_1

[13] YAcharya, U. R., Molinari, F., Sree, S. V., Chattopadhyay, S., Ng, K. H., and Suri, J. S., Automated diagnosis of epileptic EEG using entropies. Biomed. Signal Process. Control 7(4):401–408, 2012.