Detection to Intervention: Comparative Analysis of Text Classification Models for Suicide Prevention

Syeda Tanjuma Tasnim
Id: 20200204093
Group: B2 (03)
Department of Computer Science
and Engineering

Ekramul Huda Chowdhury
Id: 20200204103
Group: B2 (03)
Department of Computer Science
and Engineering

Hasan Farabi
Id: 20200204106
Group: B2 (03)
Department of Computer Science
and Engineering

***Abstract*—The growing need for mental health awareness, particularly in preventing suicide, has prompted the exploration of machine learning methods for the early detection of suicidal tendencies. Our paper presents a comparative analysis of multiple machine learning classifiers: Naive Bayes, Logistic Regression, Random Forest, K-Nearest Neighbour (KNN), and Support Vector Machine (SVM) for classifying text data related to suicide. We utilized a dataset that was compiled using the comments in social media and mental health forums to train and evaluate these classifiers. Performance metrics such as accuracy, precision, recall, and F1-score were used to measure model effectiveness, with Logistic Regression and SVM emerging as the top performers.**

***Index Terms*—Naive Bayes, Logistic Regression, Random Forest, Support Vector Machine, K-Nearest Neighbour, Text Classification, Suicide Prevention, Mental Health.**

I. Motivation

The rising prevalence of suicide, particularly among young adults, has emerged as a significant public health crisis. Suicide is often linked to untreated mental health issues, which can be identified through behavioral cues, especially in online forums and social media platforms where individuals express their thoughts and emotions. Detecting suicidal intent early using automated systems that analyze text-based data can enable timely intervention and save lives. This research is motivated by the need for such automated tools, leveraging machine learning to analyze and classify text data that may indicate suicide risk.

II. Introduction

Suicide prevention is a crucial element of public health. The World Health Organization (WHO) estimates that over 700,000 people die by suicide each year. Mental health issues like depression, anxiety, and trauma are key contributors to suicidal thoughts. In many cases, individuals who are struggling, share their feelings in online spaces, providing a window into their mental state. Machine learning offers a promising avenue for the development of systems that can automatically detect signs of suicidal intent in written text, potentially providing alerts to healthcare professionals for intervention and save lives.

Our paper presents a comparative analysis of five popular machine learning algorithms: Naive Bayes, Logistic Regression, Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbour (KNN) for classifying text into ”suicide” and ”non-suicide” categories. By evaluating the performance of each classifier, this study aims to identify the most effective model for detecting suicidal tendencies in written content.

III. Literature Review

Sentiment analysis, a sub-field of Natural Language Processing (NLP), has garnered significant attention in recent years due to its wide array of applications in areas such as mood analysis, depression detection, and suicide risk management. Our paper solely focuses on suicide risk management but the development and application of machine learning and deep learning models in sentiment analysis have been extensively explored in various studies.

At first, in the context of mood analysis and depression detection, Zohuri et al. (2020) discuss the utility of artificial intelligence (AI) in mood analysis, depression detection, and suicide risk management. Their study emphasizes the potential of AI to enhance mental health care through early detection and intervention strategies [1].

Similarly, Chary et al. (2020) highlight the significance of feature extraction techniques in sentiment analysis of textual data. Their research underscores the importance of effective feature selection in improving the performance of sentiment analysis models [2].

The review by Hussain et al. (2020) delves into the various methodologies and approaches used in sentiment analysis, including machine learning and deep learning techniques. They provide a comprehensive overview of the state-of-the-art methods and their applications in different domains [3].

In a study focused on healthcare applications, Al-Mosaiwi et al. (2020) examine the use of sentiment analysis in detecting depression from social media texts. Their findings suggest that sentiment analysis can be a valuable tool in identifying individuals at risk of depression based on their online activity [4].

Moreover, the work by O’Shea et al. (2020) explores the challenges and opportunities in sentiment analysis of healthcare data. They discuss the various machine learning and deep learning models employed in this domain and highlight the potential benefits of integrating sentiment analysis with electronic health records [5].

The research by Cambria et al. (2020) presents a novel approach to sentiment analysis by combining deep learning models with linguistic features. Their study demonstrates the effectiveness of this hybrid approach in improving the accuracy of sentiment classification for screening suicide risk [6].

In another study, Kaur and Sharma (2020) review the applications of sentiment analysis in the context of social media. They discuss the various machine learning algorithms used for sentiment classification and their performance in different social media platforms [7].

Furthermore, the paper by Saha et al. (2020) investigates the use of sentiment analysis in monitoring public health trends. Their findings indicate that sentiment analysis can be an effective tool in tracking the spread of diseases and public sentiment towards health policies [8].

Finally, the work by Zhang et al. (2020) focuses on the application of deep learning models in sentiment analysis of biomedical texts. Their research highlights the potential of deep learning in extracting meaningful insights from large volumes of biomedical literature. Their work gives us a new insight into suicide and mental health challenges from the biomedical literature. [9] [10].

IV. Dataset

The dataset used for this study consists of anonymized social media posts and comments from various mental health-related forums. The data was labeled manually into two classes: ”suicide” and ”non-suicide.” The ”suicide” class includes posts where individuals express suicidal thoughts or ideation, while the ”non-suicide” class includes general mental health discussions without explicit suicidal intent.

V. Methodology

This section outlines the approach taken to develop and evaluate various machine-learning models for suicide detection in text data. It includes details of the pre-processing techniques used to prepare the dataset, as well as the implementation of five classification models: Naive Bayes, Logistic Regression, Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbour (KNN). Each model's performance was evaluated using standard classification metrics.

A. Preprocessing Techniques

The raw dataset obtained from social media platforms and various mental health forums contained various challenges typical of textual data, such as noise, irrelevant words, and inconsistent formats. To ensure that the data was in a suitable form for machine learning models, we applied the following preprocessing steps:

- **Data Cleaning**: First, we removed non-textual elements such as HTML tags, special characters, numbers, and URLs from the raw text. Additionally, any extra whitespace and stopwords were eliminated.
- **Tokenization**: This process splits the text into individual units or ”tokens” (typically words). In this study, tokenization was performed at the word level. For example, the sentence *”I am feeling hopeless”* would be split into tokens: `["I", "am", "feeling", "hopeless"]`.
- **Lemmatization**: Each word was reduced to its base or dictionary form. For example, ”running” was transformed into ”run,” and ”better” became ”good.” This helped reduce the dimensionality of the feature space by treating words with the same root meaning as identical features.
- **Stop Words Removal**: Commonly used words such as ”the,” ”and,” and ”is” were removed as they do not add significant meaning to the analysis. The NLTK library's list of English stop words was used for this task.
- **Vectorization (Text Representation)**: Machine learning models require numerical inputs, so the textual data was transformed into numerical representations using three methods:
  - **Bag of Words (BoW)**: In this method, each unique word in the corpus is treated as a feature, and the text is represented as a frequency count of the words in the document. However, this method does not account for word importance or relationships between words.
  - **Term Frequency-Inverse Document Frequency (TF-IDF)**: TF-IDF was applied to weight the words in the corpus by their importance, reflecting how frequently a word occurs in a document relative to its frequency in the entire corpus. This helped reduce the impact of common words appearing frequently across many posts.
  - **N-Gram**: N-Grams capture a word or character sequences to understand the context. In our tests, we have used three instances of *n*, Unigrams for focusing on individual words, while bigrams and trigrams were used to capture pairs and triplets. Using N-grams helps models grasp word dependencies, improving tasks like text classification, sentiment analysis, and language modeling.

The processed dataset was then split into training (80%) and testing (20%) sets for model training and evaluation.

B. Models Used

We implemented and compared the performance of five distinct machine learning models for the binary classification task (suicide-related vs. non-suicide-related content). These models were selected based on their effectiveness in text classification tasks and interpretability.

1) Naive Bayes (Multinomial Naive Bayes): Naive Bayes is a probabilistic classifier based on Bayes' theorem, which assumes independence between the features (in this case, the words). Despite the simplicity of this assumption, Naive Bayes performs exceptionally well in text classification problems due to the high dimensionality of text data.

$$P(C \mid X) = \frac{P(X \mid C)P(C)}{P(X)}$$

where $P(C \mid X)$ is the probability of class C given the feature vector X (the document), and $P(X \mid C)$ is the likelihood of observing the feature vector given class C.

We used Multinomial Naive Bayes, ideal for discrete features like word counts. It assigns class probabilities based on word frequency, making it efficient for text classification. While it handles large feature spaces well, its assumption of feature independence can be overly simplistic.

2) Logistic Regression: Logistic Regression is a discriminative classifier that models the probability of a binary outcome based on the input features. It assumes a linear relationship between the input features and the log-odds of the outcome.

The logistic function used for prediction is:

$$P(y = 1 \mid X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n)}}$$

where β are the coefficients learned from the data, and X represents the features (TF-IDF vectors in our case).

Logistic Regression is used for binary classification and works well with a linear relationship between features and the target. It's valued for its interpretability, providing probabilities to rank predictions. While it handles high-dimensional data, it struggles with non-linear boundaries and may require techniques like class weighting or oversampling for imbalanced classes.

3) Random Forest: Random Forest is a supervised learning method that uses a combination of many decision trees and their output to make a prediction. To decide which sample of data to use for training each node of a decision tree, every decision tree is trained with a bootstrap sample of the data collected, and at any given branch of the tree, only a random subset of features is considered.

It generates multiple decision trees and the final output is an average or a mode of the generated trees to minimize overfitting. It is scalable with high-dimensional data, it can handle interactions between the features and can handle non-linear relationships. It also has an assessment of relative feature importance for the identification of important predictors. However, it's computationally intensive and may not handle sparse text data as effectively as simpler models such as Logistic Regression or Naive Bayes.

4) Support Vector Machine (SVM): Support Vector Machine is a strong classification model that can find the hyperplane with the largest margin to separate the classes of the data. However, text classification is often faced with high dimensionality of the feature space, and SVM is a good fit for this data type.

It creates a hyperplane that best establishes a maximum margin between the set classes. it is optimized to nonlinear equations, for high-dimensional sparse text data using a dense linear kernel. SVM operates on high-dimensional spaces like the ones created by BoW, TF-IDF, or any other feature extraction technique and does not suffer from overfitting. Although effective for text classification, this method can be relatively costly in terms of time and feature space and depends on the choice of C and the kernel type.

5) K-Nearest Neighbour (KNN): K-Nearest Neighbors (KNN) algorithm is employed for text classification mainly because of its simplicity and efficiency. KNN categorizes text by identifying the *k* most similar texts to an input using distance measures such as cosine similarity measures on text that is first represented numerically, such as with TF-IDF or word vectors. Instead, this enables the algorithm to label it based on the majority of the class of neighbors that are closest to it.

It is especially beneficial in tasks involving text similarity, such as sentiment analysis and document categorization. This makes it efficient and flexible since it only requires a small number of training samples due to its non-parametric nature.

C. Evaluation Metrics

The performance of all models was evaluated using the following metrics:

- **Accuracy**: The percentage of correctly classified instances.
- **Precision**: The proportion of true positives among all predicted positives. It measures the model's ability to avoid false positives.
- **Recall (Sensitivity)**: The proportion of true positives among all actual positives, measuring the model's ability to detect all instances of suicide-related content.
- **F1-Score**: The harmonic mean of precision and recall, providing a single measure of the model's performance when there's an imbalance between classes.

All models were implemented in Python using the Scikit-learn library. Cross-validation was performed to ensure the robustness of results, and hyperparameters were tuned using grid search to optimize model performance.

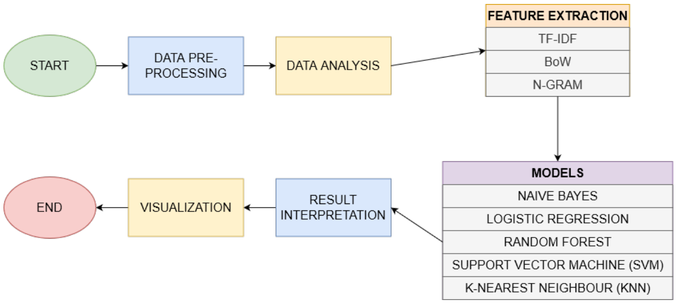


Fig. 1. Working Methodology

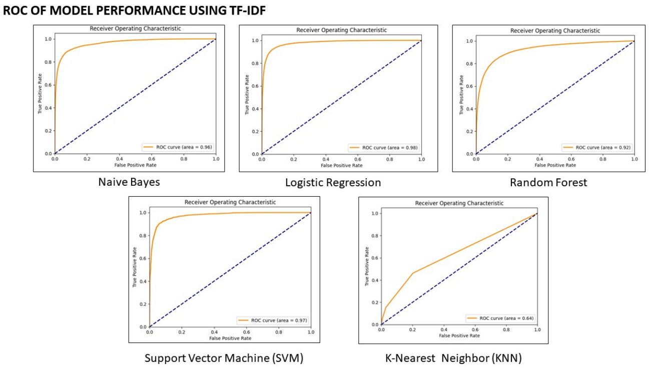
VI. Result Analysis

The models were evaluated on a test set comprising 20% of the original dataset. We have compared the results of all five models using the three pre-processing techniques in the following.

TABLE I
Performance Metrics of Machine Learning Models using TF-IDF pre-processing technique

Models	Accuracy	Precision	Recall	F1-Score
Naive Bayes	0.86	0.80	0.95	0.87
Logistic Regression	0.92	0.93	0.91	0.92
Random Forest	0.84	0.89	0.78	0.83
Support Vector Machine	0.92	0.93	0.90	0.91
K-Nearest Neighbour	0.52	0.88	0.05	0.10

Table 1 shows the result of the models when using TF-IDF pre-processing technique, Logistic Regression and SVM perform the best, with high accuracy (0.92), precision (0.93), and F1-Scores (0.92 and 0.91, respectively), while KNN significantly under-performs with the lowest scores, especially in recall and F1-Score making it the least effective model.



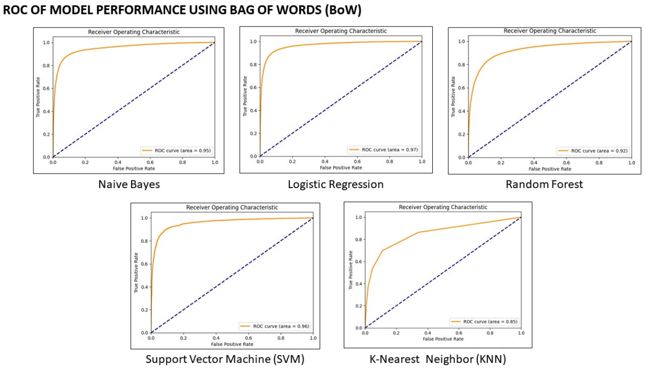
Naive Bayes
Logistic Regression
Random Forest
Support Vector Machine (SVM)
K-Nearest Neighbor (KNN)

Fig. 2. ROC of various models using TF-IDF

TABLE II
Performance Metrics of Machine Learning Models using Bag of Words (BoW) pre-processing technique

Models	Accuracy	Precision	Recall	F1-Score
Naive Bayes	0.84	0.78	0.94	0.85
Logistic Regression	0.90	0.94	0.86	0.93
Random Forest	0.85	0.89	0.79	0.84
Support Vector Machine	0.90	0.92	0.86	0.89
K-Nearest Neighbour	0.74	0.91	0.53	0.67

Table 2 shows the results of the five models when using Bag of Words (BoW) as the pre-processing technique. Logistic Regression and SVM lead with 0.90 accuracy and high F1-Scores (0.93 and 0.89). Naive Bayes and Random Forest show moderate performance, while KNN underperforms with the lowest accuracy (0.74) and F1-Score (0.67).



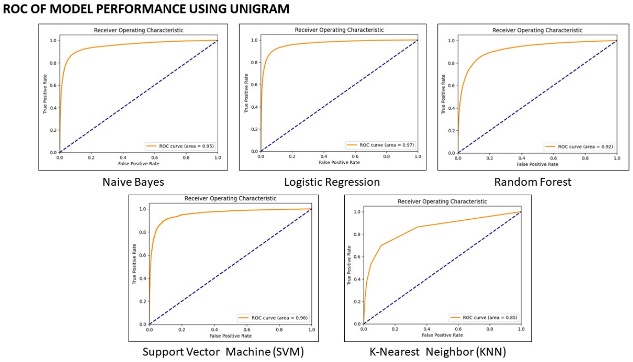
Naive Bayes
Logistic Regression
Random Forest
Support Vector Machine (SVM)
K-Nearest Neighbor (KNN)

Fig. 3. ROC of various models using BoW

TABLE III
Performance Metrics of Machine Learning Models using N-Gram (Uni Gram) pre-processing technique

Models	Accuracy	Precision	Recall	F1-Score
Naive Bayes	0.84	0.78	0.94	0.85
Logistic Regression	0.90	0.94	0.86	0.90
Random Forest	0.85	0.89	0.79	0.84
Support Vector Machine	0.90	0.92	0.86	0.89
K-Nearest Neighbour	0.74	0.91	0.53	0.67

Lastly, Table 3 shows the results using the N-Gram (Uni Gram) pre-processing technique. Here, Logistic Regression and SVM lead with 0.90 accuracy and high F1-Scores (0.90 and 0.89). Naive Bayes and Random Forest perform moderately, while KNN underperforms with the lowest accuracy (0.74) and F1-Score (0.67).



Naive Bayes
Logistic Regression
Random Forest
Support Vector Machine (SVM)
K-Nearest Neighbor (KNN)

Fig. 4. ROC of various models using Unigram

Overall, we can conclude that after testing five different models with three distinct preprocessing techniques, Logistic Regression consistently delivered the best results across all cases. However, the Support Vector Machine (SVM) closely followed, with its performance being nearly on par with Logistic Regression. Among all the models tested, K-Nearest Neighbors (KNN) produced the poorest results.

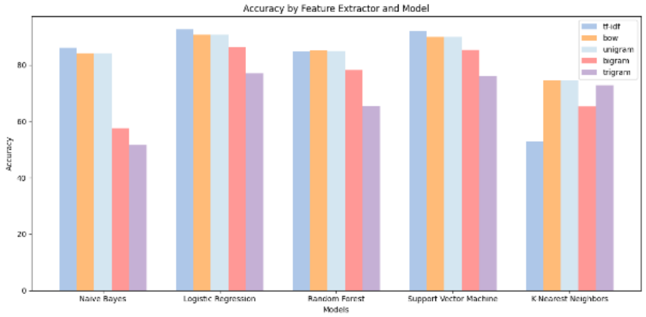


Fig. 5. Overall Comparison of Results

VII. Research Gap

While this study provides valuable insights into the performance of classical machine learning models for detecting suicidal language, it is important to acknowledge several limitations that could impact the generalizability and robustness of the findings. First, the dataset used in this research is relatively small, which may limit its ability to capture the full diversity and nuances of suicidal expressions across different demographic groups, platforms, and cultural contexts. A more extensive and diverse dataset could yield more accurate and reliable results, especially in identifying subtle variations in language that may indicate suicidal intent. Furthermore, this dataset is solely focused on the ”English” language, making the results applicable primarily to regions where English is the dominant or native language. This language-specific limitation raises concerns about the applicability of the findings in non-English-speaking countries, where cultural and linguistic differences may influence the expression of suicidal thoughts.

Second, the study did not explore the use of deep learning models such as Recurrent Neural Networks (RNN) and Transformer-based models like BERT, which have demonstrated superior performance in natural language processing tasks. These advanced models are particularly well-suited for capturing complex patterns and long-term dependencies in text, which may be crucial in identifying nuanced emotional and psychological states in social media posts. By focusing solely on classical machine learning models, the study may have missed the opportunity to leverage the strengths of these deep learning approaches, which could have significantly enhanced the accuracy and robustness of the detection system. Consequently, future research should consider incorporating deep learning techniques to further improve the performance and scalability of suicidal language detection systems across diverse languages and cultural contexts.

VIII. Conclusion

This study conducted a comprehensive comparison of five machine learning models—Naive Bayes, Logistic Regression, Random Forest, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM)—for classifying text related to suicide. Among the models evaluated, SVM and Logistic Regression stood out as the most effective, achieving the highest accuracy and F1-scores. These results underscore the robustness of these models in identifying suicide-related content. Looking ahead, we plan to extend our research by incorporating larger and more diverse datasets, which will allow us to capture a wider range of textual nuances. Additionally, we will explore advanced deep learning models to potentially further improve the accuracy and reliability of suicide detection in textual data, aiming to enhance the effectiveness of preventative measures and support systems.

References

[1] N. B. Zohuri and N. S. Zadeh, “The Utility of Artificial Intelligence for Mood Analysis, Depression Detection, and Suicide Risk Management,” *Journal of Health Science*, vol. 8, no. 2, 2020. doi: 10.17265/2328-7136/2020.02.003.

[2] P. Chary, *Feature Extraction Techniques in Sentiment Analysis*, 2020. [Online]. Available: https://sci-hub.se/https://link.springer.com/chapter/10.1007/978-3-319-70284-1_34

[3] A. Hussain, et al., *Sentiment Analysis Using Machine Learning and Deep Learning*, 2020. [Online]. Available: https://www.mdpi.com/1999-4893/13/1/7

[4] J. Al-Mosaiwi, et al., *Sentiment Analysis in Detecting Depression from Social Media Texts*, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6918275

[5] K. O’Shea, et al., *Challenges and Opportunities in Sentiment Analysis of Healthcare Data*, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9199553

[6] E. Cambria, et al., *Combining Deep Learning Models with Linguistic Features for Sentiment Analysis*, 2020. [Online]. Available: https://journals.sagepub.com/doi/full/10.1177/1178222618792860

[7] H. Kaur and A. Sharma, *Applications of Sentiment Analysis in Social Media*, 2020. [Online]. Available: https://sci-hub.se/https://link.springer.com/article/10.1007/s10916-020-01669-5

[8] S. Saha, et al., *Sentiment Analysis in Monitoring Public Health Trends*, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667096822000465

[9] X. Zhang, et al., *Application of Deep Learning Models in Sentiment Analysis of Biomedical Texts*, 2020. [Online]. Available: https://www.mdpi.com/1660-4601/19/19/12635

[10] X. Zhang, et al., *Sentiment Analysis in Biomedical Literature*, 2020. [Online]. Available: https://journals.sagepub.com/doi/abs/10.1177/0261927X211036171