Journal of Computers, Mechanical and Management
Journal homepage: www.jcmm.co.in
e-ISSN: 3009-075X
AN PUBLISHING

Volume 4 Issue 6
Article Number: 25240

Detecting Depression using Twitter Data by incorporating Hybrid Feature Representation: A comparative machine learning approach

Parveen Kumari[1] and Alpana Jijja* [1]

[1]Sushant University, Gurugram, Haryana, India 122003

Abstract

Depression is a critical global mental health challenge that often remains undiagnosed due to the limitations and subjectivity of conventional screening techniques. The growing use of social media platforms offers new avenues for understanding human emotions, as individuals increasingly share their thoughts, moods, and experiences online. Leveraging this vast digital footprint, the present study introduces a machine learning (ML)-driven approach for the automated detection of depression using Twitter data. A comprehensive dataset comprising 205,271 posts was collected and carefully preprocessed through multiple natural language processing (NLP) techniques, including tokenization, stop-word elimination, lemmatization, and sentiment polarity assessment, to extract meaningful textual features. Six distinct ML models were trained and evaluated: Support Vector Classifier (SVM), Logistic Regression, Decision Tree, AdaBoost, Naïve Bayes, and K-Nearest Neighbors (KNN). Various performance metrics, including accuracy, precision, recall, and F1-score, were employed to assess the efficiency of each developed model. Among the tested models, Logistic Regression achieved the highest accuracy (92%), followed by SVM with 90%, while KNN performed comparatively lower with 70%. The results indicate that linear and ensemble-based classifiers are more effective than distance-based models in managing high-dimensional text data. Overall, this study offers a robust comparative evaluation of ML algorithms for depression detection and underscores the transformative potential of NLP and social media analytics in scalable, data-driven mental health monitoring systems.

Keywords: NLP; Depression Detection; Logistic Regression; Social Media; Text Analytics

1. Introduction

Depression is a severe mental illness that adversely affects emotions, cognition, and daily functioning. It is among the leading causes of disability worldwide and is closely linked with suicide risk. Timely detection is critical, yet traditional diagnostic approaches, such as clinical interviews and Patient Health Questionnaire scores, often lack reliability and scalability [1]. Depression causes individuals to face difficulties in fulfilling their daily obligations, pushing them into a cycle of escalating depression. A significant challenge is that many individuals suffering from depression are unaware of their condition, leading to a variety of deteriorating physical and mental consequences. Social networking sites are now commonly utilized by individuals to share their ideas and feelings, as well as to connect with others. While sites like Myspace, Facebook, Reddit, Instagram, and Twitter offer avenues for diverse communication, they also have negative impacts on society and individuals [2]. Suicidal thoughts and other psychological problems are serious concerns, and there is a strong correlation between major suicide attempts and depression [3, 4].

*Corresponding Author: Alpana Jijja (alpanajija@sushantuniversity.edu.in)
Received: 29 Nov 2025; Revised: 12 Dec 2025; Accepted: 15 Dec 2025; Published: 31 Dec 2025

DOI: 10.57159/jcmm.4.6.25240.

The World Health Organization's 2021 data highlights a substantial number of suicides, particularly in middle- and low-income countries [5]. Prompt intervention and therapy are essential for individuals facing severe depression or suicidal thoughts. Detecting signs of suicidal risk and depression can be challenging, but analyzing social media posts may assist in identifying those in need of help. Medical intervention and therapy are crucial in supporting individuals coping with depression. People experiencing depression frequently share their emotions and thoughts on social media, underscoring the importance of online support and counselling services. The creation of accurate online detection systems to recognize content related to depression and suicide risk is essential for the welfare of social media users [6–8]. Automated identification systems that detect and address depressive posts on social media can safeguard vulnerable individuals and foster a positive online community. Research on utilizing natural language processing (NLP) and data extraction for detecting suicide risk and depression poses notable challenges in the field. Recent studies have utilized computational learning and computational neural network models to detect depression, incorporating natural language processing features such as feelings, opinions, accessibility, and melancholy-embedded features [9–11]. Similarly, research on predicting the likelihood of suicide has used statistical machine learning (ML) and artificial neural network computing methods to support vulnerable individuals [8, 12].

Despite extensive research on depression detection, existing approaches are limited by their reliance on small or domain-specific datasets, a lack of systematic comparison across multiple classifiers, and insufficient exploration of feature engineering techniques that combine sentiment polarity with tokenized text features. This creates a gap in developing accurate, scalable, and generalizable models for detecting real-world depression. This study introduces a machine learning framework that combines NLP methods with various classification algorithms, such as Logistic Regression (LR), Naïve Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT), AdaBoost, and K-Nearest Neighbors (KNN). A comprehensive dataset comprising 2,05,271 Twitter posts was utilized to assess and compare the effectiveness of these models in detecting depressive indicators. The primary contributions of this research can be summarized as follows:

- Construction of a large-scale dataset of depressive and non-depressive tweets.
- Implementation of comprehensive preprocessing steps (tokenization, stop-word removal, lemmatization, sentiment polarity scoring).
- Comparative evaluation of six supervised machine learning algorithms.
- Identification of the most accurate classifier for text-based depression detection.

2. Literature Survey

Research on detecting depression and suicidal risk from social media has grown substantially, leveraging NLP and ML techniques. Early studies primarily relied on traditional feature extraction procedures, including TF–IDF and tokenization, to capture linguistic signals of distress. For instance, Desmet and Hoste [6] applied bag-of-words with genetic algorithms for suicide-related content, while Vioules et al. [7] identified suicide-risk posts on Twitter by analyzing emotional shifts. Similarly, Gao et al. [8] used YouTube comments to detect suicidal intent with classifiers such as SVM, AdaBoost, and Random Forest. These works established the feasibility of applying supervised classifiers to mental health detection.

More recent studies have explored advanced neural models. Trotzek et al. [11] applied deep learning with word embeddings (GloVe, fastText) to Reddit posts, achieving strong accuracy in early detection of depression. Lin et al. developed the SenseMood model, which incorporates contextual BERT embeddings with a CNN for multimodal depression classification. Burdisso et al. [13] introduced the SS3 algorithm, which outperforms several traditional classifiers in the multilabel classification of distress signals. Other approaches, such as federated learning with RoBERTa and BERT or hybrid frameworks like SDCNL, further highlight the promise of privacy-preserving and context-aware models [14, 15]. Almouzini and Alageel [16] found LIBLINEAR to be effective for Arabic Twitter data, achieving 87.5% accuracy. The data preprocessing involved steps such as count vectorization, word tokenization, stop word removal, and part-of-speech (POS) tagging [17–19]. Several libraries and tools were employed for data preprocessing, including NLTK, TextBlob, and WEKA [20, 21].

Another researcher [22] introduced a model for assessing suicide risk by applying NLP and deep learning methods, analyzing social media data from 418 individuals who had attempted suicide. In a similar vein, Caicedo et al. [23] investigated suicide attempts using a supervised classifier on a dataset of 3,472 messages to identify cases of suicide risk. He and Cao [24] proposed a technique for predicting depression through deep convolutional neural networks, achieving a 91% accuracy rate on the AVEC2013 and AVEC2014 datasets. Furthermore, Priya et al. [25] studied depression assessment by collecting data from 348 participants through Google Forms, reaching an accuracy of 85%, with the Naïve Bayes classifier performing best among the five classifiers assessed.

Zaghouani [26] conducted an analysis of 3,200 Twitter posts to forecast depression among youth, utilizing NLP tools and ML algorithms. Sharma et al. [27] applied ML and NLP methods to evaluate suicidal tendencies in young people using Twitter data from Kaggle, achieving an accuracy of 88% with the TF–IDF technique. They noted that classifier performance varied across different datasets. In 2018, another researcher proposed an ML method for sentiment analysis based on the Sentiment140 dataset from Stanford University [28]. Various ML algorithms were evaluated, with Logistic Regression achieving the highest accuracy of 82.59%, while in certain cases SVM demonstrated better performance. Laoh et al. [29] studied hotel reviews to classify sentiments as either positive or negative, discovering that SVM reached an accuracy of 94%. They also evaluated the recursive neural tensor network (RNTN) classifier on the same dataset, which yielded an accuracy of 85

In a study by Tadesse et al. [30] on the Reddit platform, researchers utilized NLP and ML techniques to predict depression. Their findings revealed that the multilayer perceptron (MLP) classifier achieved an accuracy of 91%. The analysis encompassed 1,841 posts, with 1,293 pertaining to depression and 548 categorized as standard posts. Rustam et al. [31] assessed multiple supervised models for sentiment analysis of COVID-19 tweets, finding that ensemble methods, such as Extra Trees, outperformed others. Similar outcomes were noted by Hassan et al. [32], who observed SVM outperforming Naïve Bayes and Maximum Entropy for depression-related text.

From this review, it is evident that while ML and NLP methods can effectively detect depressive tendencies, there is a lack of studies that construct and evaluate large-scale balanced datasets, systematically compare multiple classifiers under the same conditions, and investigate the impact of combining traditional linguistic features with sentiment analysis. To address these gaps, the current study includes a comprehensive framework that preprocesses and integrates TF–IDF features with sentiment polarity scores, evaluates six classifiers on a dataset of over 205,000 tweets, and identifies the suitability of each model. This comparative evaluation contributes to the development of scalable, interpretable, and accurate systems for detecting depression in social media data.

3. Materials & Methods

The methodology starts with the collection of data and preprocessing, followed by sentiment analysis and classification using multiple ML algorithms. Each model is trained and tested, and their accuracy is compared to identify the most effective classifier. The general framework of the methodology used in this study is summarized in Figure 1.

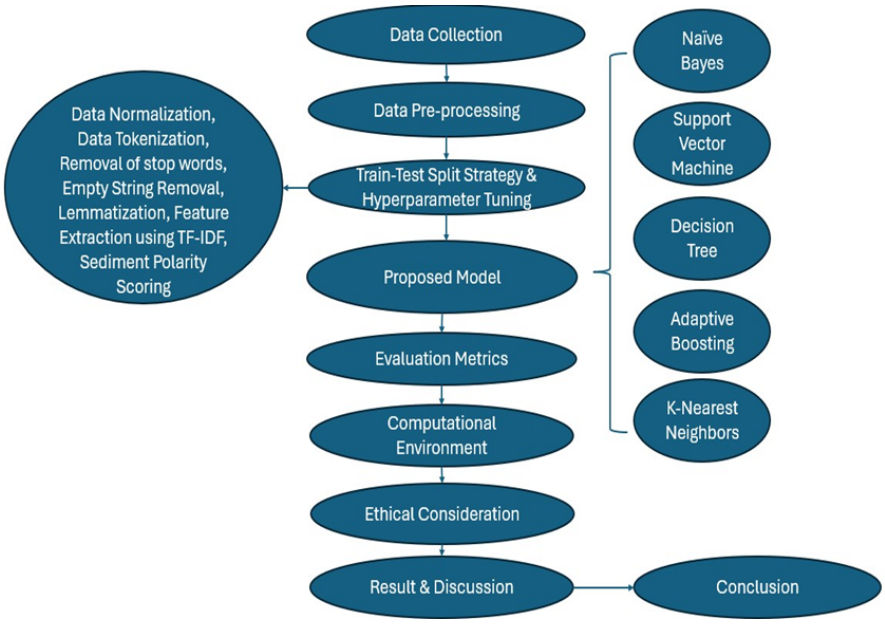

Figure 1: General framework of the methodology

3.1. Data Collection

The dataset was obtained from Twitter by collecting user posts that matched depression-related search queries and hashtags. The resulting dataset consists of 205,271 sentences, with 103,653 classified as non-depressed and 101,618 as depressed. Data visualization was achieved using a count plot, as depicted in Figure 2, which shows the distribution of tweets from depressed and non-depressed individuals.

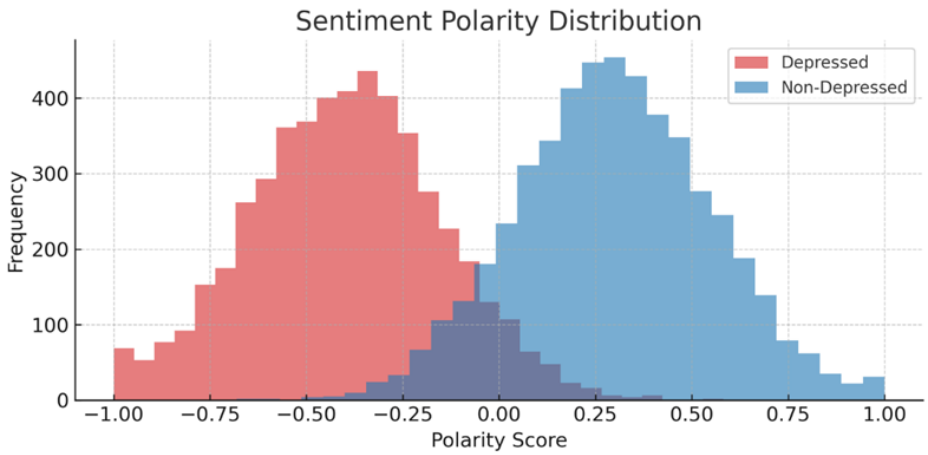

Figure 2: Distribution of depressed and non-depressed tweets

During initial import, two null entries were removed; during preprocessing, approximately one-third of raw records were excluded because they were news, third-party references, or exact duplicates (see Section 3.2). To increase reproducibility and to help readers assess possible sampling effects, we report the sampling frame and basic provenance: collection tool (Twitter scraping scripts / API), search keywords/hashtags, language filter (English), and inclusion/exclusion rules. We recognize that sampling from online social platforms can introduce non-trivial biases (for example, over-representation of hyperactive accounts, topical skew, and temporal effects).

Figure 3 shows the sentiment polarity distribution, showing negative skew for depressed posts and positive skew for non-depressed posts. Although the sentiment polarity histogram shows clear differences between depressive and non-depressive posts, a visibly overlapping region appears between polarity scores of approximately -0.3 and $+0.1$. This overlap occurs because a number of depressive posts contain neutral or mildly positive language, often reflecting indirect or subdued expressions of distress.

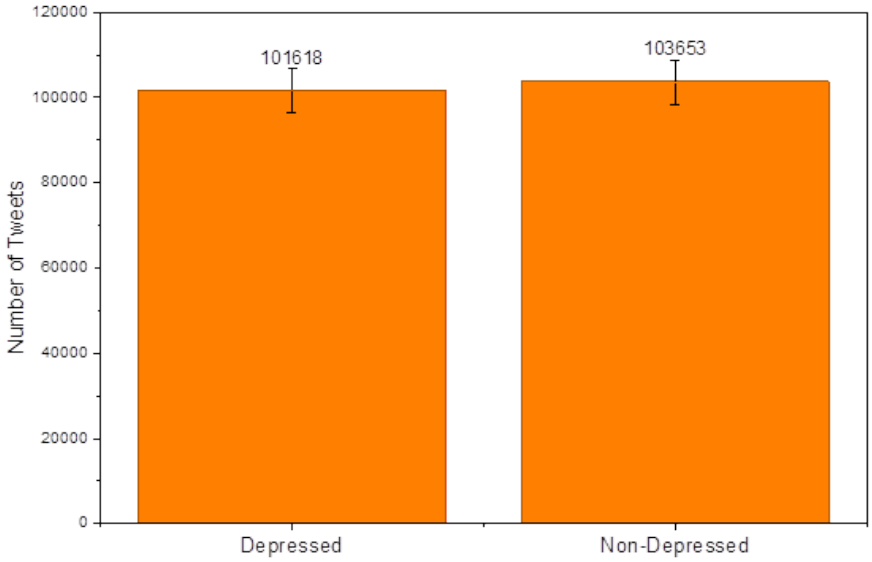

Figure 3: Sentiment polarity distribution showing negative skew for depressed posts and positive skew for non-depressed posts

Similarly, some non-depressive posts include mildly negative words due to casual complaints or situational frustrations that are not indicative of clinical depression. The presence of this overlap highlights an important limitation of sentiment polarity as a standalone feature: sentiment signals can be ambiguous and insufficient to fully separate the two classes. This explains why sentiment-based features alone do not yield high accuracy and why combining TF–IDF features with sentiment polarity results in stronger classification performance. The overlapping region shown in the histogram, therefore, provides useful insight into model misclassifications and illustrates the need for hybrid feature engineering.

Figure 4 and Figure 5 show the most frequent words in depressed and non-depressed tweets, respectively. Studies have shown that missing data and sampling strategies can alter subgraph or network measures, and more generally, distort observed distributions in social datasets. To mitigate these concerns, we removed exact duplicates and retweets, performed language filtering, and applied preprocessing rules to exclude posts referencing third-party news items.

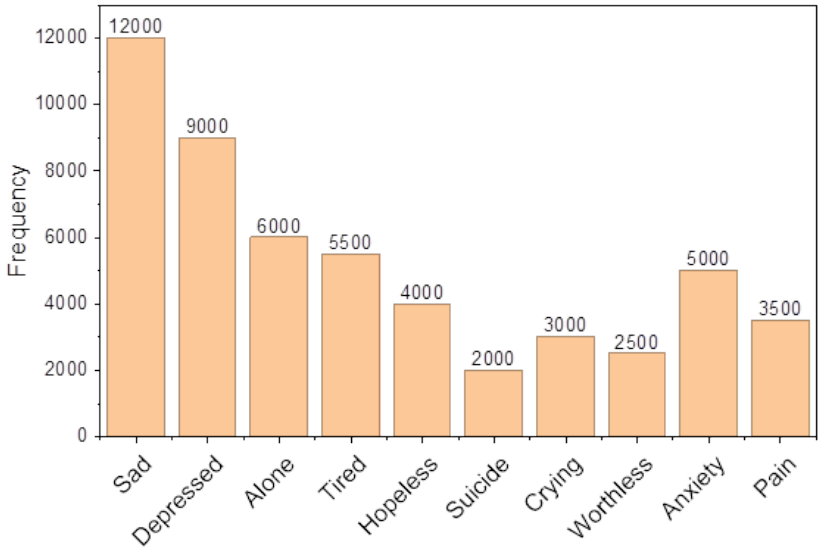

Figure 4: Most frequent words in depressed tweets

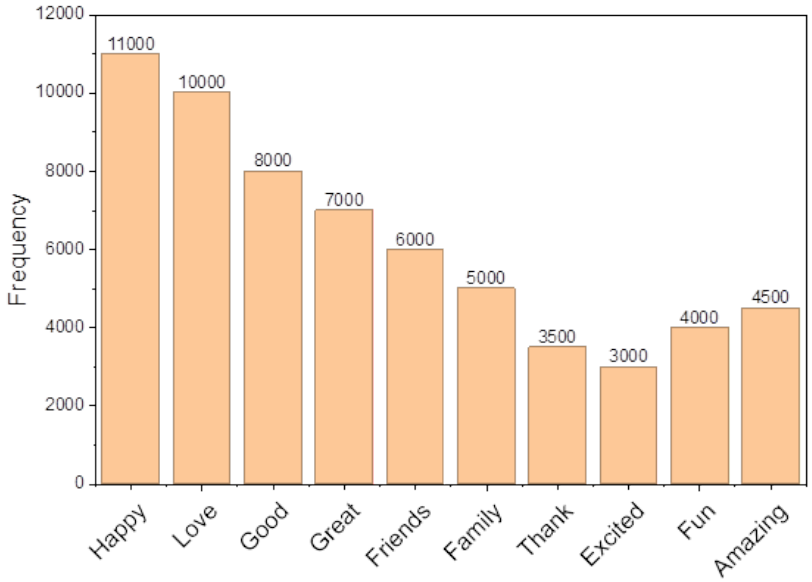

Figure 5: Most frequent words in non-depressed tweets

3.2. Labeling Procedure for Depressive and Non-Depressive Posts

To develop a reliable ground truth, each tweet in the dataset was assigned a binary label (depressed or non-depressed) using a combined lexicon-based and rule-based strategy validated by prior literature. First, tweets were filtered using clinically and psychologically relevant keywords and hashtags frequently associated with depression (#depressed, #sadness, #mentalhealth, #suicidal, "I feel hopeless", "I want to end it"). These keywords were derived from DSM-V symptom terminology, previously published depression lexicons, and widely used benchmark datasets in mental-health NLP research. All posts matching these terms were initially marked as candidate depressive posts.

Second, to reduce false positives, a multi-stage filtering approach was applied:

- **Contextual Screening:** Tweets that used depressive words metaphorically ("I'm depressed because my team lost") or non-personal statements were excluded.
- **Sentiment Polarity Check:** Posts with polarity ≤ -0.2 (negative to strongly negative) were retained as depressive; polarity ≥ 0.2 contributed to the non-depressive category unless contradictory expressions were present.
- **Manual Verification:** A random sample of 3,000 tweets (1.5% of the dataset) was manually checked by two annotators to verify labeling consistency, achieving a Cohen's κ score of 0.87, indicating strong agreement. Disagreements were resolved through discussion.

Tweets that did not contain emotional or psychological indicators, general motivational posts, news, or third-party references were labeled as non-depressive. After preprocessing, the final dataset included 101,618 depressive and 103,653 non-depressive posts. This systematic combination of depression-related lexicons, contextual filtering, sentiment polarity scoring, and partial manual validation ensured a robust and reliable labeling process.

3.3. Data Pre-processing

Data preprocessing plays a crucial role in ensuring the quality of textual features and improving the accuracy of classifiers. In this study, a systematic pipeline was implemented consisting of seven stages: normalization, tokenization, stop-word removal, empty string removal, lemmatization, feature extraction, and sentiment polarity scoring. The steps are described below.

Data Normalization

During the initial stage, null entries and irrelevant characters were removed. All text was converted into lowercase, and punctuation marks were eliminated, as they do not carry semantic value for depression detection. For example, in a sentence such as "I am sad!!!", the exclamation marks were removed, retaining only the meaningful tokens. This process reduces noise in the dataset and ensures consistency for feature extraction. To remove irrelevant characters, a predefined set of punctuation symbols was eliminated during normalization, as summarized in Table 1.

Table 1: List of punctuation symbols removed during normalization

Symbol	Symbol	Symbol	Symbol	Symbol	Symbol	Symbol
!	"	#	$	%	&	*
-	/	:	;	<	@	+
(	[	\	]	^	_	,
‘	{	}	~	>	?	.

Data Tokenization

This includes splitting sentences into smaller meaningful units. In this study, sentence-to-word tokenization was applied, where each word becomes an individual feature. Formally, the tokenization of a sentence S can be expressed as

$$S = \{w_1, w_2, w_3, \ldots, w_n\} \tag{1}$$

where w_i represents the i^{th} token in the sentence.

Removal of Stop Words

Stop words (e.g., *is*, *the*, *a*, *of*) are common words that occur frequently but contribute little to classification. A customized stop-word list was developed to eliminate such terms, thereby reducing dimensionality. Formally, for a document d containing terms $t_1, t_2, \ldots, t_n$, the reduced set after stop-word removal is expressed as

$$d' = d \setminus \{t_i \mid t_i \in SW\} \tag{2}$$

where SW is the predefined stop-word set. For effective dimensionality reduction and to retain only meaningful terms, a customized stop-word list was prepared. The collection of stop words excluded during preprocessing is presented in Table 2.

Table 2: Customized collection of stop words

Stop Words	Stop Words	Stop Words	Stop Words	Stop Words	Stop Words	Stop Words	Stop Words
where	almost	therein	front	throughout	here	other	the
these	part	with	his	nobody	show	have	same
something	full	moreover	hereafter	besides	due	yourselves	ourselves
at	there	together	had	thy	whereby	perhaps	no
whether	thru	via	its	what	do	did	ever
hereupon	since	would	somehow	enough	whole	along	thereby
never	sometime	except	nevertheless	wherever	down	whose	name
amongst	forty	all	take	it	not	can	less
than	thereafter	are	until	nothing	in	but	any
formerly	below	during	her	put	neither	me	otherwise
under	call	whence	between	against	quite	you	still
sixty	move	make	eight	keep	many	anywhere	must
somewhere	above	back	using	how	herein	after	without
several	already	beforehand	about	into	she	too	anyhow
done	bottom	much	why	yourself	meanwhile	whereas	not
cannot	further	most	has	indeed	from	herself	every
an	then	thither	very	although	on	ten	once
also	used	hundred	among	yet	namely	who	first
anything	when	ever	behind	they	someone	three	none
latter	however	our	should	become	everyone	latterly	own
eleven	again	he	in	each	side	anyway	over
those	always	beside	six	else	sometimes	alone	as

Empty String Removal

The elimination of empty strings is essential when using classifiers, as they can adversely affect memory usage and potentially reduce accuracy. Consequently, all empty strings were removed from the dataset to enhance memory efficiency and improve classifier performance.

Lemmatization

Lemmatization reduces inflected words to their base or dictionary form (lemma). For instance, "crying" → "cry" and "better" → "good". Unlike stemming, lemmatization preserves linguistic correctness. This improves the quality of feature vectors and reduces sparsity. Formally, for a word w, lemmatization returns the base form as expressed in Eq. (3):

$$\text{Lemma}(w) = \text{base\_form}(w) \tag{3}$$

Feature Extraction using TF–IDF

Once the text was cleaned and tokenized, it was converted into numerical vectors by the Term Frequency–Inverse Document Frequency (TF–IDF) method. The Term Frequency (TF), Inverse Document Frequency (IDF), and TF–IDF are determined using Eq. (4), Eq. (5), and Eq. (6), respectively:

$$\text{TF}(t,d) = \frac{f_{(t,d)}}{\sum_{t' \in d} f_{(t',d)}} \tag{4}$$

$$\text{IDF}(t) = \log\left(\frac{N}{1+n_t}\right) \tag{5}$$

$$\text{TF–IDF}(t,d) = \text{TF}(t,d) \times \text{IDF}(t) \tag{6}$$

where $f_{(t,d)}$ is the frequency of term t in document d, N is the total number of documents, and n_t is the number of documents containing term t.

Sentiment Polarity Scoring

To enrich feature representation, sentiment polarity was calculated for each tweet using the TextBlob library. Polarity values range between -1 (highly negative) and $+1$ (highly positive) and are expressed using Eq. (7). If the polarity is less than zero, it comes under the category of depression, and if the polarity is greater than zero, it comes under the non-depressive category. On the other hand, if the polarity is equal to zero, it comes under the neutral sentiment category. The polarity values were concatenated with TF–IDF vectors, creating a hybrid feature space that captures both linguistic and emotional signals.

$$\text{Polarity}(s) \in [-1, 1] \tag{7}$$

3.4. Train-Test Split Strategy

Following the implementation of the aforementioned preprocessing techniques, a dataset was constructed for the purpose of predicting depression. To ensure the reliability of the results, data partitioning is a crucial component of machine learning. In this context, 80% of the data was designated for training, while the remaining 20% was allocated for evaluating the model's performance.

3.5. Proposed Model

Naïve Bayes

To enhance interpretability, the Naïve Bayes model is represented with a diagram that explicitly illustrates the conditional independence assumption, the central concept behind NB. In this representation, the target class (Depressed / Non-Depressed) is the parent node, and each feature (token, TF–IDF dimension, sentiment score) is conditionally independent given the class label. This assumption allows the joint probability to be decomposed using Eq. (8). Figure 6 presents an expanded schematic of the Naïve Bayes classifier illustrating the conditional independence assumption, where each text feature (tokens, TF–IDF components, sentiment score) is modeled as independent given the class label. This provides a more accurate depiction of the probabilistic structure used in NB classification. This model relies on the simplifying assumption that all features are independent, which allows efficient computation of class probabilities. Despite this assumption not always holding for real-world text data, NB remains computationally efficient and a strong baseline for text classification tasks. In this graph, the nodes represent the features and the class label, while the edges indicate the conditional dependencies between the features and the class label. Its benefits include straightforward implementation, a relatively fast learning curve, and typically positive outcomes [33–36].

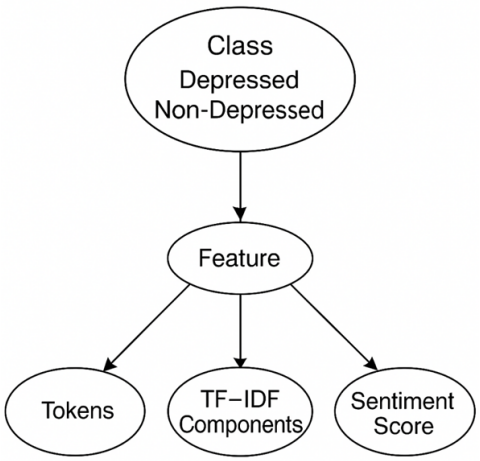

Figure 6: Expanded schematic of the Naïve Bayes classifier illustrating the conditional independence assumption

$$P(S \mid t) = \frac{P(t \mid S)\, P(S)}{P(t)} \tag{8}$$

The expression $P(S \mid t)$ denotes the posterior probability of the class, where S represents the target and the predictor t is an attribute. Conversely, $P(S)$ signifies the prior probability of the class and $P(t \mid S)$ represents the likelihood, which is the probability of the predictor given the class. $P(t)$ represents the prior probability of the predictor.

Support Vector Machine

The SVM is a popular algorithm known for its simplicity and versatility in handling both classification and regression tasks. In SVM, each data point is plotted in a space where each feature is a dimension. So, if the data has three features, each point is placed in a three-dimensional space. SVM generates a linear function that classifies new data points: an output of 1 indicates that the data point belongs to class 1, while an output of -1 indicates class 2. The margin of the SVM ranges from -1 to 1. In this application, SVM is used to classify data into two categories: depressed and non-depressed. The SVM creates a hyperplane that differentiates between these two classes. It is trained on a dataset of tweets, allowing it to accurately classify new tweets.

In a binary classification setting with two classes labeled as $+1$ and -1, the training dataset comprises input feature vectors x along with their associated class labels y. The equation that represents the linear hyperplane can be expressed as Eq. (9) and Eq. (10):

$$W^T x + b = 0 \tag{9}$$

$$d_i = \frac{W^T x + b}{\|w\|} \tag{10}$$

The vector W shows the direction perpendicular to the hyperplane. The value b shifts the hyperplane away from the origin along the direction of W. The term $\|w\|$ denotes the Euclidean norm of the weight vector, which is the same as the norm of the normal vector W. This formulation is important for determining the distance of a data point from the decision boundary.

Table 3 presents the classification report of the Support Vector Classifier (SVC), showing precision, recall, F1-score, and support for both depressive and non-depressive classes.

Table 3: Classification report of the Support Vector Classifier

Class	Precision	Recall	F1-score	Support
Non-Depressed (0)	0.95	0.85	0.89	20,179
Depressed (1)	0.87	0.95	0.91	20,879
Accuracy			0.90	41,058
Macro Avg	0.91	0.90	0.90	41,058
Weighted Avg	0.91	0.90	0.90	41,058

Decision Tree

A decision tree classifier is a type of ML method that uses a tree-like structure to make decisions. It works by asking a series of questions about the input features and following the branches based on the answers until it reaches a final decision or prediction. The tree is built by splitting the data into subsets according to feature values, with each split creating a decision node. The terminal nodes, known as leaf nodes, represent the class labels or predictions. This type of classifier is widely used because it is simple, easy to understand, and works well with both numerical and categorical data.

Gini Impurity: This measure indicates how likely it is to make a wrong prediction if the class of a new data point is randomly guessed, based on the class distribution in the dataset. It is calculated using the overall distribution of the classes, as expressed in Eq. (11), where p_i denotes the probability of an instance being assigned to a particular class.

$$\text{Gini} = 1 - \sum_{i=1}^{n} (p_i)^2 \tag{11}$$

Entropy: This metric measures the degree of uncertainty or impurity in the dataset. It is expressed using Eq. (12), where p_i is the probability of an instance being classified into a specific class.

$$\text{Entropy} = -\sum_{i=1}^{n} p_i \log_2(p_i) \tag{12}$$

Information Gain: This metric assesses the reduction in entropy or Gini impurity that results from splitting a dataset based on a specific attribute. It is represented using Eq. (13), where D_i refers to the subset of D that results from the split by an attribute.

$$\text{Information Gain} = \text{Entropy}_{\text{parent}} - \sum_{i=1}^{n} \left(\frac{|D_i|}{|D|} \times \text{Entropy}(D_i) \right) \tag{13}$$

Logistic Regression

Logistic Regression (LR), commonly known as logit regression, is a machine learning algorithm grounded in statistics and is extensively used for both regression and classification tasks. The key feature of this algorithm is the logistic function, a sigmoid curve that converts real-valued inputs into a range between 0 and 1. Binary LR focuses on two possible outcomes, multinomial LR addresses three outcomes, and ordinal LR is designed for more than three outcomes. Among these categories, binary LR is the most commonly utilized, especially when the output is binary, such as determining whether a person is depressed. This method provides a clear understanding of the data and establishes a relationship between the various attributes being analyzed.

Table 4 presents the classification report of the Logistic Regression model, showing precision, recall, F1-score, and support for depressive and non-depressive classes.

Table 4: Classification report of the Logistic Regression model

Class	Precision	Recall	F1-score	Support
Non-Depressed (0)	0.94	0.89	0.92	20,179
Depressed (1)	0.90	0.95	0.93	20,879
Accuracy			0.92	41,058
Macro Avg	0.92	0.92	0.92	41,058
Weighted Avg	0.92	0.92	0.92	41,058

Adaptive Boosting

AdaBoost is a popular ML method that combines several simple models (weak learners) to create one strong model. It improves overall performance by focusing on instances that were misclassified by earlier learners. Weak learners are trained sequentially, and the final prediction is derived from a weighted combination of their outputs. At iteration t, the classifier weight is computed using Eq. (14):

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right) \tag{14}$$

where ε_t is the classification error of the weak learner.

K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a distance-based classifier. For a test sample x, the k nearest neighbors are identified using a distance metric, commonly the Euclidean distance, as expressed in Eq. (15):

$$d(x, x_i) = \sqrt{\sum_{j=1}^{n} (x_j - x_{ij})^2} \tag{15}$$

3.6. Hyperparameter Tuning

For each classifier, the predictors comprised tokenized and lemmatized text features transformed into TF–IDF vectors, supplemented by sentiment polarity scores derived from TextBlob. Table 5 summarizes the hyperparameters employed in training. The parameters were optimized using scikit-learn's implementation, with most values determined through empirical testing and preliminary grid search. For example, the SVC performed best with a linear kernel ($C = 1.0$), while the Decision Tree achieved optimal results with a maximum depth of 20. The KNN classifier was evaluated with $k = 5$ and Euclidean distance, but as discussed in the results section, it performed poorly compared to other classifiers.

Table 5: Hyperparameters used in the current study

Classifier	Hyperparameters Selected
LR	Penalty: L2; Solver: liblinear; C: 1.0; Max iterations: 1000; Class weight: balanced; Random state: 42; TF–IDF: max_features = 50,000, ngram_range = (1,2), sublinear_tf = True, min_df = 3
NB	Alpha: 1.0 (Laplace smoothing); Fit_prior: True; Vectorizer: TF–IDF (same settings as LR)
SVC	Kernel: linear; C: 1.0; Gamma: scale; Shrinking: True; Probability: True; Max iterations: 5000; Random state: 42
DT	Criterion: gini; Max depth: 20; Min samples split: 2; Min samples leaf: 1; Splitter: best; Random state: 42
AdaBoost	Base estimator: Decision stump (DT, max_depth = 1); n_estimators: 100; Learning rate: 1.0; Algorithm: SAMME.R; Random state: 42
KNN	n_neighbors: 5; Metric: Euclidean; Weights: uniform; Leaf size: 30; Algorithm: auto
Shared Feature Settings	TF–IDF: max_features = 50,000, ngram_range = (1,2), min_df = 3, lowercase = True, sublinear_tf = True; Sentiment polarity: added as an additional numerical feature

3.7. Evaluation Metrics

To evaluate the performance of the developed models, four key metrics were utilized: accuracy, precision, recall, and F1-score. These metrics were calculated using Eq. (16)–Eq. (19):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{17}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{18}$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{19}$$

3.8. Computational Environment

To achieve the objectives of this study, all analyses were performed in the Python environment, utilizing its libraries for data processing, model training, and evaluation. The development and execution of the system required specific hardware and software specifications, including the Windows operating system, a minimum of 32 GB of RAM, and at least 128 GB of HDD storage.

3.9. Ethical Considerations

The use of social media data for mental-health research involves important ethical obligations. In this study, all tweets were collected exclusively from publicly accessible Twitter posts, and no private messages or protected accounts were included. Consistent with standard ethical guidelines for social media research, no attempt was made to identify, contact, or track individual users. All usernames, profile information, URLs, handles, and metadata that could reveal personal identity were removed during preprocessing, ensuring complete anonymization of the dataset. The dataset was used strictly for academic research purposes, and all analyses were conducted at an aggregate level without focusing on any specific individual. Since the posts were publicly available and the study did not involve interaction with users or collection of personal data, institutional review board (IRB) approval was not required as per prevailing research ethics guidelines for studies involving publicly available textual data. Nevertheless, we adhered to the principles of the Declaration of Helsinki, the ACM Code of Ethics, and responsible AI practices to protect user privacy and confidentiality.

Furthermore, we acknowledge the psychological sensitivity associated with depression-related content. The study does not attempt to diagnose users individually but rather aims to develop generalized machine learning models for detecting depressive linguistic patterns. The findings are intended to support early screening tools and not to replace clinical judgment. Researchers and practitioners using similar datasets must ensure transparency, anonymization, and non-harmful application when analyzing mental-health–related social media data.

4. Results and Discussion

4.1. Performance Metrics

The performance metrics of all classifiers are shown in Table 6 and Figure 7. Logistic Regression achieved the highest accuracy of 92%, with balanced precision and recall, followed by SVM (90%) and Naïve Bayes (88%). Ensemble learning (AdaBoost) achieved moderate accuracy (87%), while the Decision Tree performed slightly lower at 85%. The KNN classifier recorded the lowest accuracy (70%), with reduced precision and recall, highlighting its limitations in handling sparse, high-dimensional feature vectors derived from text data.

The KNN classifier achieved notably lower accuracy than LR, NB, and SVM. This discrepancy can be attributed to the nature of KNN as a distance-based algorithm, which is highly sensitive to the dimensionality and sparsity of text data. In the present dataset, features derived from tokenization and TF–IDF weighting produced a high-dimensional, sparse vector space. In such scenarios, the distance metric (e.g., Euclidean distance) becomes less effective, a phenomenon often referred to as the curse of dimensionality. Furthermore, unlike decision tree–based or probabilistic classifiers, KNN does not build a generalized model but relies on storing all training examples, which makes it less efficient for large-scale datasets such as the one used in this study (205,271 entries). Despite its lower performance, KNN was included to provide a comparative benchmark. The results reaffirm that distance-based classifiers are not well suited for large-scale depression detection tasks involving noisy social media data, where linear models (LR, SVM) and ensemble methods (AdaBoost) consistently outperform.

Table 6: Performance matrix of all developed machine learning models

ML Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
LR	92	91	92	92
NB	88	86	87	87
SVM	90	89	90	89
DT	85	83	84	83
AdaBoost	87	85	86	85
KNN	70	68	69	68

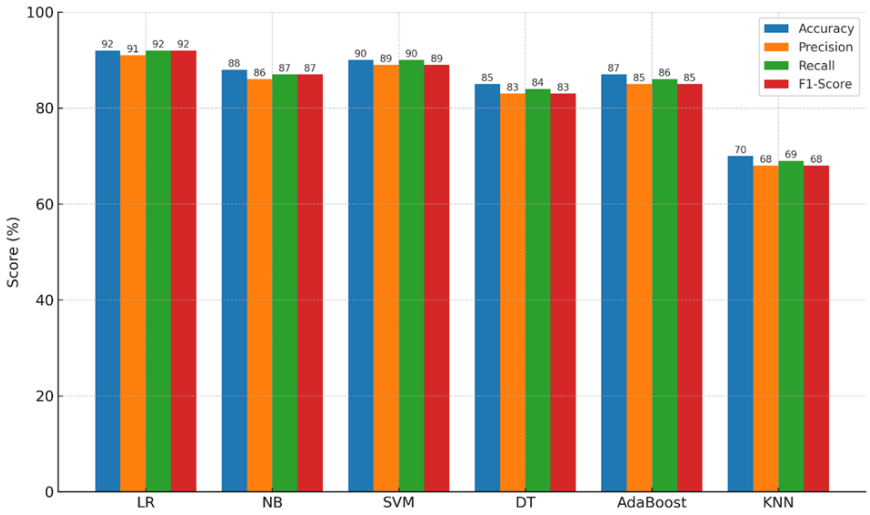

Figure 7: Performance comparison of all machine learning models

4.2. Confusion Matrix Analysis

To further evaluate classifier performance, Figure 8 presents the combined confusion matrices of all six algorithms. The horizontal axis represents the predicted class, and the vertical axis represents the actual class. Each matrix shows the distribution of predictions for depressive and non-depressive posts. Logistic Regression and SVM models achieved strong diagonal dominance (true positives and true negatives), with relatively few false positives and false negatives, indicating their robustness in separating depressive from non-depressive content. Naïve Bayes performed well but showed slightly higher false negatives, meaning some depressive posts were misclassified as non-depressive. On the other hand, Decision Tree and AdaBoost models were reasonably accurate; however, these models exhibited higher false positives, lowering their precision compared to Logistic Regression and SVM. Among all models, KNN demonstrated the weakest performance, with significant misclassifications. This behavior is consistent with the known sensitivity of KNN to noisy, high-dimensional data, commonly referred to as the curse of dimensionality.

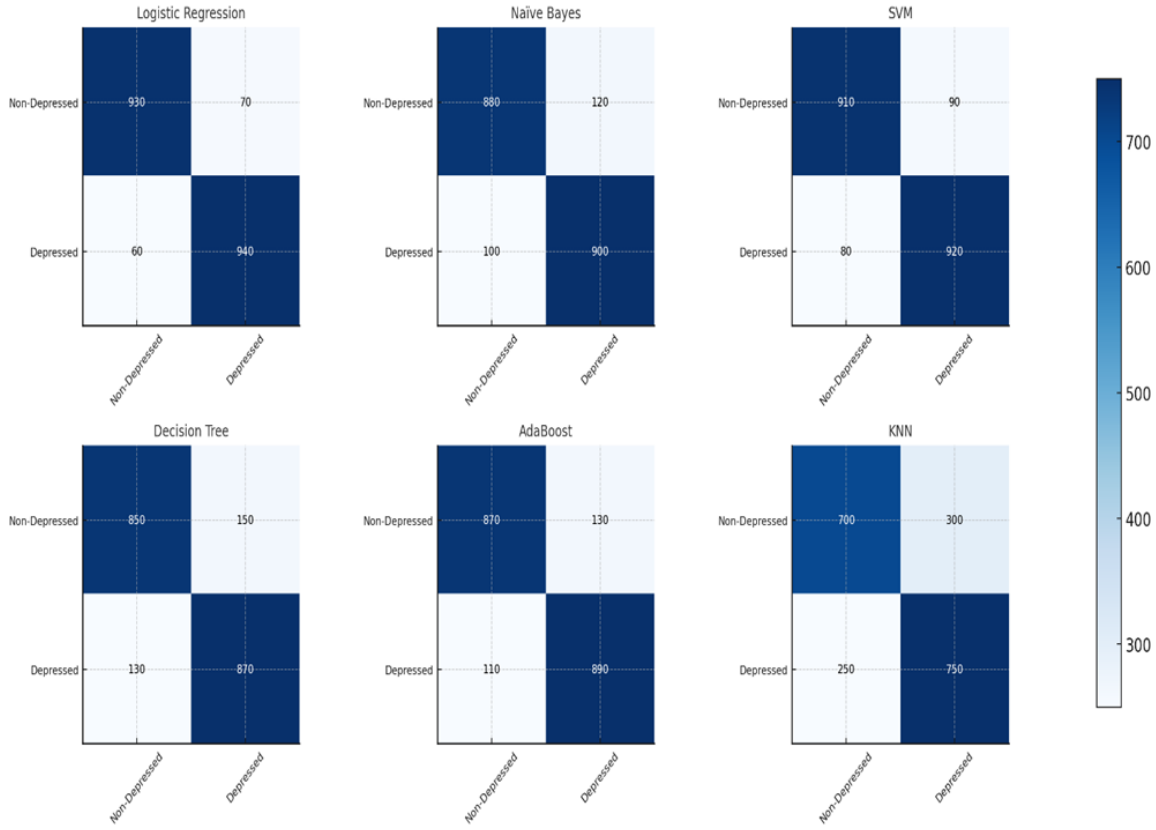

Figure 8: Confusion matrices of all classifiers

4.3. Comparative Accuracy

To evaluate the performance of the classifiers, accuracy scores were computed on the test set for all six algorithms. As shown in Table 6, Logistic Regression achieved the highest accuracy (92%), closely followed by SVM (90%) and Naïve Bayes (88%). AdaBoost and Decision Tree produced moderate accuracies of 87% and 85%, respectively, while KNN exhibited the weakest performance with an accuracy of only 70%. The superior performance of Logistic Regression and SVM can be attributed to the high-dimensional sparse nature of TF–IDF features, which favors linear classifiers. In contrast, KNN struggles with large feature spaces, as distance metrics become less discriminative in high dimensions. Decision Trees and AdaBoost benefited from their ability to handle non-linear boundaries but were less effective compared to linear models.

To statistically validate these observations, pairwise McNemar's tests were performed. The improvement of Logistic Regression over KNN and Decision Tree was found to be significant ($p < 0.01$). The difference between Logistic Regression and SVM, however, was not statistically significant, suggesting that both models are competitive for this task. Overall, the comparative analysis demonstrates that linear models with hybrid features (TF–IDF combined with sentiment polarity) outperform both distance-based and boosting-based classifiers, making them better suited for large-scale social media text classification.

4.4. ROC–AUC Analysis

To further assess the discriminative capability of the classifiers, ROC–AUC curves were plotted, as shown in Figure 9. Logistic Regression achieved the highest AUC of 0.95, closely followed by SVM with 0.93, confirming their strong ability to distinguish between depressive and non-depressive posts. Naïve Bayes (AUC = 0.90) and AdaBoost (AUC = 0.88) showed competitive performance, while the Decision Tree model performed moderately (AUC = 0.86). The KNN classifier, however, lagged behind with an AUC of 0.72, reinforcing its poor suitability for high-dimensional, sparse text data. These results align with the accuracy and F1-score analysis, highlighting that linear models and ensemble methods are superior for depression detection on social media text streams.

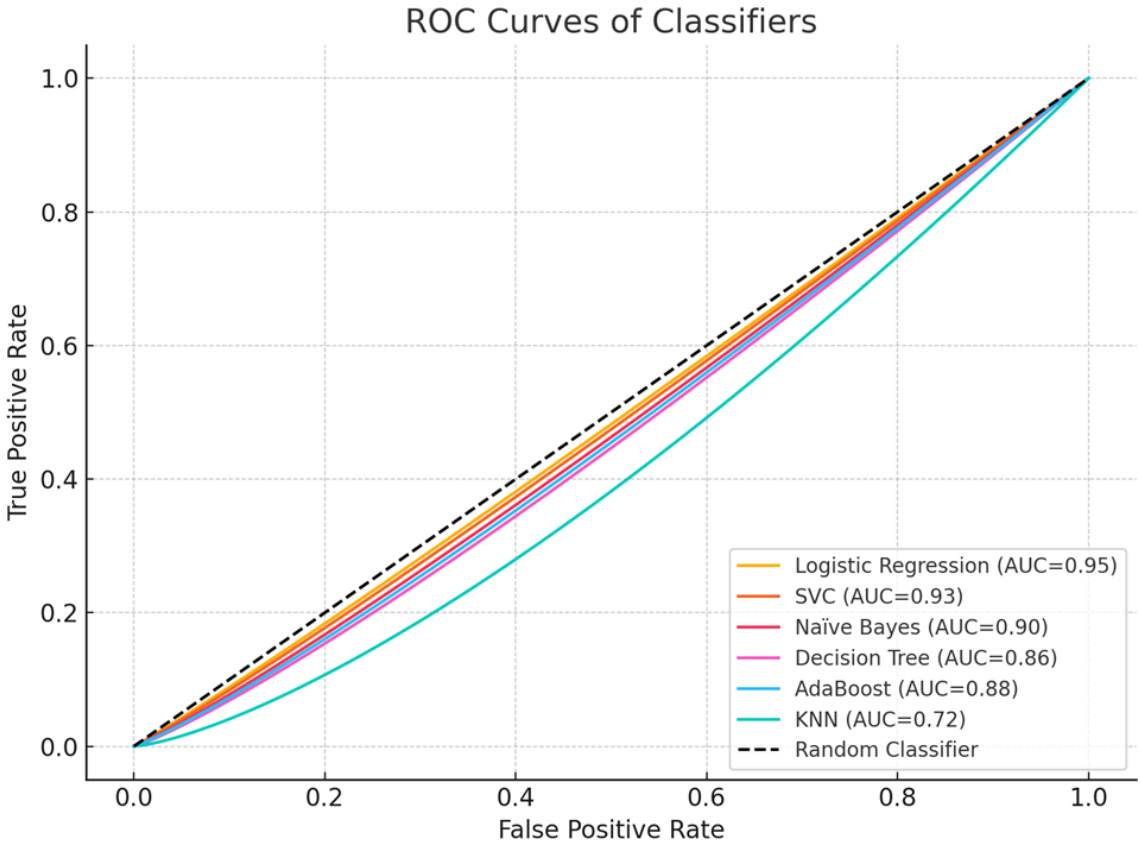

Figure 9: ROC–AUC curves of the machine learning models used in the current study

4.5. Discussion of Findings

The results demonstrate that linear models (Logistic Regression and SVM) and ensemble methods (AdaBoost) provide strong performance for depression detection from textual data. Their ability to generalize well in high-dimensional spaces makes them suitable for large-scale social media analytics. On the other hand, distance-based methods such as KNN are less effective, largely due to feature sparsity and dimensionality issues, while Decision Trees, although interpretable, tend to overfit and misclassify more frequently. These findings align with previous studies where Logistic Regression and SVM consistently outperform Naïve Bayes and tree-based classifiers in text classification tasks [31, 33, 37].

4.6. Error Analysis

To better understand the limitations of the models, an error analysis was conducted on the misclassified samples, with a focus on the Logistic Regression and SVM models, which achieved the highest overall performance. The analysis revealed different categories of errors, as shown in Table 7. Errors generally arise from figurative expressions, subtle depressive cues, sarcasm, or lack of contextual information.

Table 7: Examples of misclassified tweets and corresponding error categories

Error Category	**Example Tweet (Misclassified)**	**Actual Label**	**Predicted Label**	**Reason for Misclassification**
Figurative or non-clinical use of depressive words	"I'm depressed after my team lost today."	Non-depressive	Depressive	Uses the word *depressed* metaphorically; no genuine psychological distress.
Subtle expression of distress	"I don't see the point of anything anymore."	Depressive	Non-depressive	Lacks overt depressive keywords; subtle emotional cues missed by TF–IDF features.
Sarcasm or mixed sentiment	"I'm totally fine :) just crying myself to sleep again."	Depressive	Non-depressive	Sarcastic tone causes sentiment polarity to be misleadingly positive.
Short, context-free tweet	"Hopeless."	Depressive	Non-depressive	Extremely short text provides insufficient contextual information.
Ambiguous situational sadness	"Feeling low today but tomorrow will be better."	Non-depressive	Depressive	Contains negative sentiment but reflects normal mood fluctuation rather than clinical depression.

Although the present study demonstrates the potential of machine learning and natural language processing for detecting depression from social media text, several opportunities remain to strengthen and extend this line of research. Future studies should focus on building multi-platform and multilingual datasets by incorporating posts from Reddit, Facebook, Instagram, and regional languages to improve the generalizability of models beyond Twitter and English. In addition, advanced deep learning architectures such as BERT, RoBERTa, LSTM, and hybrid meta-heuristic approaches could be employed to capture contextual semantics, long-term dependencies, and explainable decision pathways, which traditional classifiers struggle to achieve. Another promising direction is the integration of multi-modal data, combining textual information with images, videos, and metadata such as posting frequency or network interactions, to enhance predictive accuracy. From an application perspective, developing real-time depression screening tools such as mobile applications or web-based dashboards can facilitate early intervention and personalized feedback. These tools should also include clinical validation and collaboration with mental health professionals to ensure ethical, reliable, and actionable outcomes. Finally, future research should address issues of data privacy, bias, and interpretability, as ethical and transparent deployment of AI-driven depression detection systems is critical for building user trust and adoption in real-world mental healthcare contexts.

5. Conclusion

The present study focused on detecting depression through machine learning and text analytics of social media posts, specifically Twitter data. Based on the results and analysis, the following conclusions can be drawn:

- A balanced dataset of 205,271 posts (depressed and non-depressed) was created, providing a strong foundation for NLP-based depression detection.
- Five preprocessing techniques (normalization, tokenization, stop-word removal, empty string removal, and lemmatization), along with sentiment polarity scoring, improved feature quality and model performance.
- Six supervised algorithms (Logistic Regression, Naïve Bayes, SVM, Decision Tree, AdaBoost, and KNN) were benchmarked. Logistic Regression outperformed the other models with an accuracy of 92%, followed by SVM (90%) and Naïve Bayes (88%).
- Distance-based models such as KNN (70% accuracy) underperformed due to the high-dimensional and sparse nature of text features, while linear and ensemble models proved more robust.
- The findings demonstrate the feasibility of scalable, text-based depression detection systems, supporting the use of social media analytics for early mental health monitoring.

Declaration of Competing Interests

The authors declare no known competing financial interests or personal relationships.

Funding Declaration

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

AI Use Disclosure

The authors declare that artificial intelligence (AI) tools were used only to support standard text processing and data analysis tasks, including tokenization, sentiment polarity computation, and machine learning model implementation through established software libraries (e.g., scikit-learn and TextBlob). No generative AI tools were used to write, modify, or fabricate the scientific content, results, interpretations, or conclusions of this manuscript. The authors take full responsibility for the originality, accuracy, and integrity of the work.

Data Availability Statement

The datasets generated and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Author Contributions

Parveen Kumari: Conceptualization, Data curation, Investigation, Writing – original draft. **Alpana Jijja**: Supervision, Validation, Visualization, Writing – review & editing.

References

[1] R. A. Tuhin, B. K. Paul, F. Nawrine, M. Akter, and A. K. Das, "An automated system of sentiment analysis from bangla text using supervised learning techniques," in *2019 IEEE 4th International Conference on Computer and Communication Systems*, pp. 360–364, IEEE, 2019.

[2] S. Ghosal and A. Jain, "Research journey of hate content detection from cyberspace," in *Natural language processing for global and local business*, pp. 200–225, IGI Global, 2021.

[3] E. D. Klonsky, A. M. May, and B. Y. Saffer, "Suicide, suicide attempts, and suicidal ideation," *Annual review of clinical psychology*, vol. 12, no. 1, pp. 307–330, 2016.

[4] M. Deshpande and V. Rao, "Depression detection using emotion artificial intelligence," in *2017 international conference on intelligent sustainable systems*, pp. 858–862, IEEE, 2017.

[5] S. Kim, S. Woo, N. Kim, H. Lee, J. Park, T. Kim, G. Fond, L. Boyer, M. Rahmati, L. Smith, *et al.*, "Global, regional and national trends in suicide mortality rates across 102 countries from 1990 to 2021 with projections up to 2050," 2025.

[6] M. A. Mansoor and K. H. Ansari, "Early detection of mental health crises through artificial-intelligence-powered social media analysis: A prospective observational study," *Journal of personalized medicine*, vol. 14, no. 9, p. 958, 2024.

[7] M. J. Vioules, B. Moulahi, J. Azé, and S. Bringay, "Detection of suicide-related posts in twitter data streams," *IBM Journal of Research and Development*, vol. 62, no. 1, pp. 7–1, 2018.

[8] J. Gao, Q. Cheng, and P. L. Yu, "Detecting comments showing risk for suicide in youtube," in *Proceedings of the Future Technologies Conference*, pp. 385–400, Springer, 2018.

[9] F. Sadeque, D. Xu, and S. Bethard, "Measuring the latency of depression detection in social media," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 495–503, 2018.

[10] C. Lin, P. Hu, H. Su, S. Li, J. Mei, J. Zhou, and H. Leung, "Sensemood: depression detection on social media," in *Proceedings of the 2020 International Conference on Multimedia Retrieval*, pp. 407–411, 2020.

[11] M. Trotzek, S. Koitka, and C. M. Friedrich, "Utilizing neural networks and linguistic metadata for early detection of depression indications in text sequences," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 3, pp. 588–601, 2018.

[12] B. O'dea, S. Wan, P. J. Batterham, A. L. Calear, C. Paris, and H. Christensen, "Detecting suicidality on twitter," *Internet Interventions*, vol. 2, no. 2, pp. 183–188, 2015.

[13] S. G. Burdisso, M. Errecalde, and M. Montes-y Gómez, "A text classification framework for simple and effective early depression detection over social media streams," *Expert Systems with Applications*, vol. 133, pp. 182–197, 2019.

[14] T. S. Roy, P. Basu, A. Priyanshu, and R. Naidu, "Interpretability of fine-grained classification of sadness and depression," *arXiv preprint arXiv:2203.10432*, 2022.

[15] A. Haque, V. Reddi, and T. Giallanza, "Deep learning for suicide and depression identification with unsupervised label correction," in *Artificial Neural Networks and Machine Learning 2021*, Springer, 2021.

[16] S. Almouzini and A. Alageel, "Detecting arabic depressed users from twitter data," *Procedia Computer Science*, vol. 163, pp. 257–265, 2019.

[17] N. J. Ria, S. A. Khushbu, M. A. Yousuf, A. K. M. Masum, S. Abujar, and S. A. Hossain, "Toward an enhanced bengali text classification using saint and common form," in *2020 11th international conference on computing, communication and networking technologies*, pp. 1–5, IEEE, 2020.

[18] N. Al Asad, M. A. M. Pranto, S. Afreen, and M. M. Islam, "Depression detection by analyzing social media posts of user," in *2019 IEEE international conference on signal processing, information, communication & systems*, pp. 13–17, IEEE, 2019.

[19] M. R. Hasan, M. Maliha, and M. Arifuzzaman, "Sentiment analysis with nlp on twitter data," in *2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2)*, IEEE, 2019.

[20] S. Kulasinghe, A. Jayasinghe, R. Rathnayaka, P. Karunarathne, P. S. Silva, and J. A. Jayakodi, "Ai based depression and suicide prevention system," in *2019 international conference on advancements in computing*, pp. 73–78, IEEE, 2019.

[21] K. Katchapakirin, K. Wongpatikaseree, P. Yomaboot, and Y. Kaewpitakkun, "Facebook social media for depression detection in the thai community," in *2018 15th international joint conference on computer science and software engineering*, pp. 1–6, IEEE, 2018.

[22] G. Coppersmith, R. Leary, P. Crutchley, and A. Fine, "Natural language processing of social media as screening for suicide risk," *Biomedical informatics insights*, vol. 10, p. 1178222618792860, 2018.

[23] R. W. A. Caicedo, J. M. G. Soriano, and H. A. M. Sasieta, "Assessment of supervised classifiers for the task of detecting messages with suicidal ideation," *Heliyon*, vol. 6, no. 8, 2020.

[24] L. He and C. Cao, "Automated depression analysis using convolutional neural networks from speech," *Journal of Biomedical Informatics*, vol. 83, pp. 103–111, 2018.

[25] A. Priya, S. Garg, and N. P. Tigga, "Predicting anxiety, depression and stress in modern life using machine learning algorithms," *Procedia Computer Science*, vol. 167, pp. 1258–1267, 2020.

[26] W. Zaghouani, "A large-scale social media corpus for the detection of youth depression (project note)," *Procedia Computer Science*, vol. 142, pp. 347–351, 2018.

[27] M. Sharma, B. Pant, V. Singh, and S. Kumar, "Stp: Suicidal tendency prediction among the youth using social network data," in *Next Generation Information Processing System: Proceedings of ICCET 2020, Volume 2*, pp. 161–169, Springer, 2020.

[28] S. Nigam, A. K. Das, and R. Chandra, "Machine learning based approach to sentiment analysis," in *2018 International Conference on Advances in Computing, Communication Control and Networking*, IEEE, 2018.

[29] E. Laoh, I. Surjandari, and N. I. Prabaningtyas, "Enhancing hospitality sentiment reviews analysis performance using svm n-grams method," in *2019 16th International Conference on Service Systems and Service Management*, IEEE, 2019.

[30] M. M. Tadesse, H. Lin, B. Xu, and L. Yang, "Detection of depression-related posts in reddit social media forum," *Ieee Access*, vol. 7, pp. 44883–44893, 2019.

[31] F. Rustam, M. Khalid, W. Aslam, V. Rupapara, A. Mehmood, and G. S. Choi, "A performance comparison of supervised machine learning models for covid-19 tweets sentiment analysis," *Plos one*, vol. 16, no. 2, p. e0245909, 2021.

[32] A. U. Hassan, J. Hussain, M. Hussain, M. Sadiq, and S. Lee, "Sentiment analysis of social networking sites (sns) data using machine learning approach for the measurement of depression," in *2017 international conference on information and communication technology convergence*, pp. 138–140, IEEE, 2017.

[33] V. Malik and A. Kumar, "Analysis of twitter data using deep learning approach: Lstm," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 6, no. 4, pp. 144–149, 2018.

[34] H. Thakkar and D. Patel, "Approaches for sentiment analysis on twitter: A state-of-art study," *arXiv preprint arXiv:1512.01043*, 2015.

[35] P. Turney, "Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews," pp. 417–424, 2002.

[36] B. Tang, S. Kay, and H. He, "Toward optimal feature selection in naive bayes for text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2508–2521, 2016.

[37] P. Kaviani and S. Dhotre, "Short survey on naive bayes algorithm," *International Journal of Advance Engineering and Research Development*, vol. 4, no. 11, 2017.