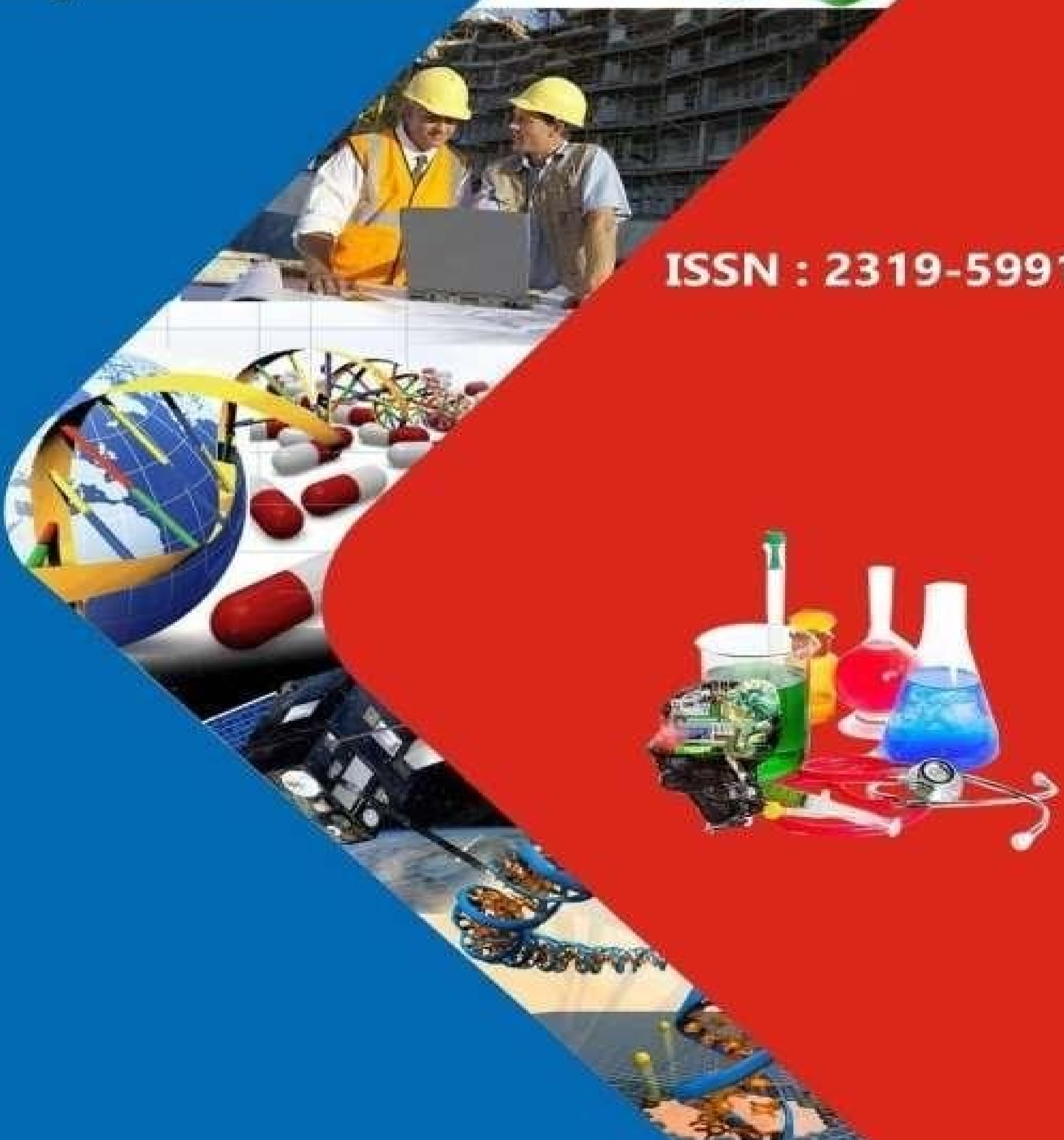
IJERST

International Journal of
Engineering Research and Science & Technology



ISSN : 2319-5991

www.ijerst.com

Email: editor@ijerst.com or editor.ijerst@gmail.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

A Unified Framework for Depression Detection Using ML Algorithms

J. Venkata Nandini[1],K.Faizz Ahmad[2] ,A. Dharani[3],L. Sai Chaitanya[4],B. Jagadeesh[5].

[1234]Department of Computer Science and Engineering, Acharya Nagarjuna University, Andhra Pradesh, India .

[5]B.Tech, M.Tech. Assistance Prof. Department of Computer Science and Engineering,

Acharya Nagarjuna University, Andhra Pradesh

Email ID's:venkatanandinijajula@gmail.com[1], faizzkhadri@gmail.com[2]·dharaniamp6666@gmail.com[3], lammatha.saichaitanya9177@gmail.com[4], b.jagadeeshanu@gmail.com[5].

Abstract:

Depression is a common mental health disorder that greatly impacts a person's quality of life and productivity. With the rise of social media and digital communications, analyzing textual data for mental health insights has gained considerable attention. This study presents a comparative analysis of traditional Machine learning algorithms—such as Random Forest, Bagging, Support Vector Classifier (SVC), Logistic Regression, and Naive Bayes. Textual data is preprocessed and vectorized before being fed into the machine learning models, Evaluation is carried out using performance metrics such as accuracy, precision, recall, and F1-score.and compare the models. Experimental results indicate that logistic regression and navie bayes machine learning models Achieve higher accuracy compared to traditional techniques and contextual understanding. This study highlights the potential methods for robust and scalable depression detection systems.

Key Words: Random Forest, Logistic Regression, Support Vector Classifier (SVC) ,Naive Bayes ,Bagging Classifier , Machine Learning.

Introduction:

Depression is a widespread and debilitating mental health disorder, affecting over 280 million people globally. Defined by persistent sadness, lack of interest, and cognitive challenges, it not only impacts individual well-being but also carries significant societal and economic costs. Early detection is crucial for effective treatment, yet traditional diagnostic methods—relying on self-reported symptoms and clinical assessments—are often time-consuming, subjective, and inaccessible for many.

With the rise of digital communication, especially on social media, large volumes of user-generated text present new opportunities for analyzing mental health indicators. Natural Language Processing (NLP) and machine learning (ML) techniques can leverage these data sources to detect signs of depression through linguistic patterns and semantic cues.

This study compares several machine learning algorithms—including Random Forest, Bagging, Support Vector Classifier (SVC), Logistic Regression, and Naive Bayes—for detecting depression from text data. We evaluate these models based on accuracy, precision, recall, and F1-score to assess their effectiveness. The goal is to contribute to the development of automated tools that support early diagnosis and timely intervention in mental health care.

1.1 Need Of Detecting Mental Health:

Rising Mental Health Concerns: Depression is a leading cause of disability worldwide, with increasing prevalence, especially in the post-pandemic era. Early detection can significantly improve outcomes.

Lack of Clinical Resources: Mental health professionals are limited; automated tools can assist in early screening, Alleviating the strain on healthcare systems.

Data-Driven Insights: Machine learning (ML) and deep learning (DL) models can uncover hidden patterns in text or behavioral data that may be overlooked in traditional diagnostics.

Improved Accuracy: Combining ML algorithms like Random Forest, SVC, and Logistic Regression can enhance detection accuracy through complementary strengths.

Scalability: Automated systems are capable of processing large amounts of data rapidly, making them perfect for large-scale mental health monitoring (e.g., social media screening, surveys).

1.2 Identified Research Gaps:

Limited Integration of Traditional ML and Deep Learning Approaches:

Most existing studies use either traditional ML (e.g., Logistic Regression, Random Forest) or deep learning (e.g., BERT), but not both in a hybrid framework. A combined approach could improve robustness and accuracy.

Underutilization of BERT for Mental Health Detection:

Although BERT and transformer-based models have shown promise in NLP tasks, their application in depression detection is still emerging and often not compared directly with classical ML techniques.

Lack of Benchmarking Across Diverse Algorithms:

Few studies provide a comprehensive comparison between multiple classifiers (e.g., Bagging, SVC, Naïve Bayes) and advanced deep learning models under the same conditions or datasets.

Inconsistent Datasets and Evaluation Metrics:

Studies often use different datasets (e.g., social media posts, clinical text) without standardization, making it hard to assess model generalizability. There's also a lack of uniform performance metrics like F1-score, precision, and recall, especially in imbalanced datasets.

Lack of Explainability in Deep Learning Models:

BERT-based models often act as "black boxes." There is a need for interpretable models or methods to visualize how decisions are made, especially in sensitive applications like mental health.

Neglect of Multilingual and Cross-Cultural Analysis:

Most models are trained on English data, ignoring linguistic and cultural nuances in how depression is expressed in different regions or languages.

Literature Review:

Depression detection using Natural Language Processing (NLP) and machine learning (ML) has gained increasing attention as researchers explore non-invasive, scalable methods for mental health assessment. Early studies primarily relied on manually crafted linguistic features, such as word frequency, sentiment scores, and psycholinguistic markers, extracted from written text to differentiate between depressed and non-depressed individuals (Pennebaker et al., 2003). Tools like the Linguistic Inquiry and Word Count (LIWC) have been widely used to quantify psychological and emotional cues in language.

Multiple machine learning algorithms have been applied to classify depression-related content. For example, Resnik et al. (2015) utilized Support Vector Machines (SVMs) to classify posts from mental health forums, achieving promising results through a combination of lexical and semantic features. Similarly, Coppersmith et al. (2014) demonstrated that features derived from Twitter posts could predict depression with reasonable accuracy using classifiers such as Logistic Regression and Naive Bayes. Their work highlighted the potential of social media data in identifying mental health conditions outside clinical settings.

Random Forest and ensemble methods like Bagging have also been explored for their robustness and ability to handle high-dimensional textual data (Shen et al., 2017). These models often outperform single classifiers by reducing variance and improving generalization. However, traditional ML models depend heavily on feature engineering, which can be time-consuming and may fail to capture deeper linguistic nuances (Calvo et al., 2017).

Other studies have compared the performance of different classifiers. Tsugawa et al. (2015) compared Naive Bayes, Logistic Regression, and SVM for detecting depressive tendencies on Twitter, noting that classifier choice, feature selection, and preprocessing significantly influenced performance metrics such as precision and recall.

Despite their limitations, traditional machine learning approaches remain valuable due to their interpretability and lower computational requirements compared to deep learning models. They also offer a practical baseline for developing automated systems for depression detection, particularly when resources or annotated data are limited.

In summary, the literature shows a consistent interest in leveraging textual data and machine learning algorithms—such as SVM, Random Forest, Logistic Regression, Naive Bayes, and Bagging—to detect depression. While accuracy varies depending on datasets and features, these approaches provide a foundation for building scalable mental health monitoring tools.

2.Problem Statement:

Depression is a growing global health concern, yet early and accurate detection Continues to be a challenge because of the subjective nature of symptoms and the lack of mental health resources professionals. While machine learning (ML models have shown potential in automating the detection process, most existing approaches rely exclusively on either traditionalML.Furthermore, inconsistencies in evaluation methods, dataset limitations, and a lack of interpretability in deep learning models hinder the practical deployment of these systems. Therefore, there is a critical need to develop a hybrid approach that combines traditional machine learning techniques (such as Random Forest, Bagging, SVC, Logistic Regression, and Naïve Bayes) to improve the accuracy, scalability, and explainability of automated depression detection systems.

3.Related Work

Several studies have explored The application of machine learning and deep learning for identifying depression, especially from text data like social media posts, survey responses, or clinical notes.

1. Traditional Machine Learning Approaches:

Previous research has applied models such as Logistic Regression, Naïve Bayes, Support Vector Machines (SVM), and Random Forest to classify depression based on text features like n-grams, TF-IDF, and sentiment scores. For example, Resnik et al. (2015) used linguistic features from Reddit posts to detect depression, while Tsugawa et al. (2015) applied SVMs on Twitter data with user behavior metrics. While these models are interpretable and efficient, they often struggle with understanding contextual nuances in language.

2. Ensemble Methods:

Studies have shown that ensemble techniques like Bagging and Random Forest can improve prediction stability and reduce overfitting in depression detection tasks. These methods combine the outputs of multiple base learners, enhancing generalization on unseen data. However, they still rely heavily on manually engineered features.

3. Hybrid Models and Comparative Studies:

There is limited work that directly compares or combines traditional ML classifiers with BERT models for depression detection. Most comparative studies evaluate models in isolation, making it difficult to determine the optimal method or synergy between techniques.

4. Challenges in Datasets and Evaluation:

Many studies use imbalanced datasets, which can lead to biased predictions. Additionally, inconsistencies in metrics (e.g., accuracy vs. F1-score) and dataset domains (social media vs. clinical text) complicate comparisons across studies. Some researchers have called for more standardized benchmarks and explainable AI techniques to enhance real-world applicability.

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

4.Proposed Work

The proposed approach seeks to develop a hybrid framework for automated depression detection that combines traditional machine learning algorithms with deep learning models like BERT. This framework will leverage the strengths of both approaches to improve accuracy, robustness, and interpretability in classifying depressive symptoms from textual data (e.g., social media posts, questionnaires, or clinical notes).

1. Data Collection and Preprocessing

Collect publicly available datasets (e.g., Reddit, Twitter, DAIC-WOZ, or clinical depression corpora).

Preprocess the text by eliminating stop words, punctuation, URLs, and other irrelevant elements, and applying lemmatization.

Label the data based on depressive vs. non-depressive content (either manually or using dataset annotations).

2. Feature Extraction for Traditional Models

Extract features such as:

TF-IDF vectors

These features will be used as inputs to traditional classifiers.

3. Model Training – Traditional Machine Learning

Implement and compare the performance of:

Logistic Regression

Naïve Bayes

Support Vector Machine (SVC)

Random Forest

Bagging Classifier

Use stratified k-fold cross-validation for reliable performance measurement.

4. Hybrid Evaluation Framework

Compare Evaluate the performance of all models using standard metrics: Accuracy, Precision, Recall, F1-score, and AUC.

Investigate hybrid strategies, such as:

Model ensembling (e.g., majority voting or stacking of ML and BERT predictions)

Confidence-based fusion (e.g., choose prediction from model with higher confidence)

5. Model Interpretability

Use tools such as LIME or SHAP to explain model decisions, especially for deep learning models.

Analyze important features or attention weights contributing to depressive classification.

5.Implementation:

mental health detection using the specified machine learning and deep learning frameworks, presented as descriptive matter without any code.

Goal: Implement a system to detect indicators of mental health distress from text data.

General Steps for Any ML/DL Project:

Data Collection: Acquire a dataset consisting of text samples (e.g., social media posts, forum comments, clinical notes) that are labeled with relevant mental health categories (e.g., 'distressed', 'not distressed', or specific conditions like 'depression', 'anxiety').

Data Preprocessing: Clean the text data. This typically involves steps like:

Handling missing values.

Removing noise (HTML tags, special characters, URLs).

Lowercasing text.

Tokenization (dividing text into words or sub-word units). Eliminating stop words (frequent words such as 'the', 'a', 'is'). Stemming or lemmatization (reducing words to their base form).

Feature Engineering/Representation: Convert the cleaned Converting text into a numerical format that machine learning models can interpret. This step varies considerably between traditional ML and deep learning.

Model Selection: Choose one or more algorithms appropriate for text classification.

Training: Use the labeled training data (text features and their corresponding labels) to train the chosen model(s). Evaluation: Measure the performance of the trained model(s) on unseen test data using relevant metrics.

Prediction: Deploy the trained model to classify new, unlabeled text data.

Traditional Machine Learning Frameworks (scikit-learn):

Frameworks Covered: Logistic Regression, Naive Bayes, Support Vector Machine (SVM), Random Forest, Bagging Classifier.

Feature Representation: Traditional ML models cannot process raw text. Text must be converted into numerical vectors. Common techniques include:

Bag-of-Words (Count Vectorization): Represents a document as a vector, with each dimension corresponding to a word in the vocabulary, and the value indicating the frequency of that word in the document.

TF-IDF (Term Frequency-Inverse Document Frequency): Similar to Bag-of-Words, but adjusts word counts based on their significance across the entire dataset, lowering the weight of extremely common words.

More advanced: Using pre-trained word embeddings (like Word2Vec, GloVe) and aggregating them (e.g., averaging) to represent documents.

Model Training:

Initialize the chosen model (Logistic Regression, Multinomial Naive Bayes, SVC, RandomForestClassifier, BaggingClassifier).

Train the model on the numerical feature representations of the training text (X_train_features) and their corresponding labels (y_train). Evaluation:
Use the trained model to predict labels for the numerical feature representations of the test text (X_test_features). Compare the predicted labels (y_pred) with the actual test labels (y_test). Compute standard classification metrics such as accuracy, precision, recall, and F1-score using scikit-learn tools.

Key Considerations:

Performance heavily relies on the chosen feature engineering technique and its parameters.

Hyperparameter tuning for each model is essential to achieve optimal performance.

Naive Bayes (specifically Multinomial Naive Bayes) is often a strong baseline for text classification.

Ensemble methods (Random Forest, Bagging) can improve robustness.

Feature Representation: BERT handles text representation internally.

Tokenization: Text is broken down into sub-word units using a specific tokenizer corresponding to the pre-trained BERT model (e.g., BertTokenizerFast). Special tokens ([CLS], [SEP]) are added, and sequences are padded or truncated to a fixed maximum length (max_len).

Input Formatting: The tokenizer outputs numerical representations required by BERT, including input_ids (token indices), attention_mask (indicating real tokens vs. padding), and potentially token_type_ids.

Calculate standard classification metrics (accuracy, precision, recall, F1-score).

General Considerations for Mental Health Detection Systems:

Data Quality and Ethical Sourcing: Obtaining high-quality, accurately labeled text data is crucial and presents significant ethical challenges regarding privacy, consent, and data origin.

Ethical Implications and Bias: Deploying such systems requires careful consideration of potential biases in the training data that could lead to unfair predictions for certain demographic groups. The primary goal should often be to flag content for human review, not to make automated diagnoses. Transparency and user consent are vital.

Nuance and Context: Human language expressing distress is complex, context-dependent, and highly variable. Models need to be robust enough to handle this complexity.

Evaluation Metrics: Depending on the application's risk tolerance, different metrics might be prioritized. For instance, in a system aiming to identify individuals needing help, recall (minimizing false negatives) might be more important than precision.

Explainability: Understanding why a model made a particular prediction can be important, especially in sensitive applications. This is generally easier with traditional models but techniques are emerging for explaining deep learning predictions.

.Algorithms used :

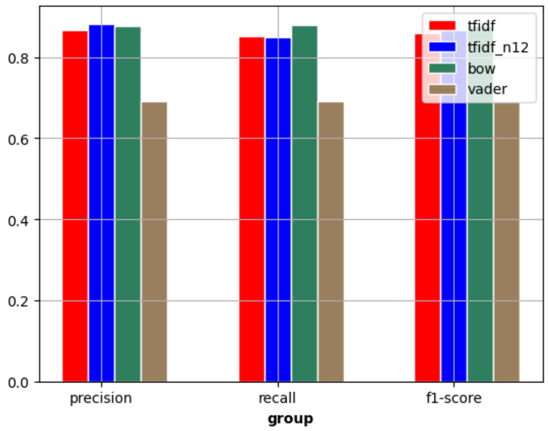
6.1 Liner SVC:

Linear Support Vector Classifier Linear SVC is a robust and commonly used machine learning algorithm for binary text classification tasks, making it highly suitable for depression detection. In this context, it is employed to categorize text data—such as social media posts or clinical notes—into "depressed" and "non-depressed" categories. A key advantage of Linear SVC is its ability to effectively manage high-dimensional feature spaces, which are typical in text data after techniques like TF-IDF or n-gram

International Journal of Engineering Research and Science & Technology
ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

vectorization. It is computationally efficient and scales well with large datasets, enabling it to process thousands of text entries quickly. The model also features a regularization parameter that aids in handling overfitting.

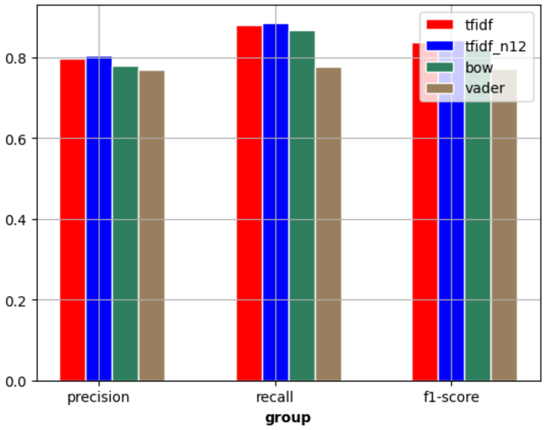
prevent overfitting, which is crucial when working with real-world, noisy data. Linear SVC fits well into the proposed hybrid framework as a traditional machine learning baseline and offers a strong benchmark for performance comparison against more complex models like BERT. However, it assumes that the data is linearly separable and may not perform as well when deeper contextual understanding is required—something that transformer-based models like BERT can better capture. Despite this, Linear SVC remains a valuable component of the overall system due to its simplicity, speed, and effectiveness in many depression detection scenarios.



6.2Random Forest:

Random Forest is an ensemble machine learning algorithm that can be effectively applied to depression detection in text data. It functions by constructing multiple decision trees during training and aggregating their outputs—usually through majority voting—to generate final predictions. This method enhances accuracy and reduces the likelihood of overfitting compared to relying on a single decision tree. In the context of depression detection, Random Forest performs well with structured features derived from text, such as TF-IDF vectors, sentiment scores, or linguistic indicators. It is especially resilient to noise and adept at capturing complex, non-linear relationships within the data. Additionally, it provides feature importance scores, which can help identify the words or patterns most indicative of depression.

depressive language. While it may not capture contextual language as deeply as deep learning models like BERT, Random Forest serves as a strong and interpretable traditional machine learning baseline in a hybrid framework. It is especially useful when working with medium-sized datasets and offers a good balance between accuracy, interpretability, and computational efficiency.

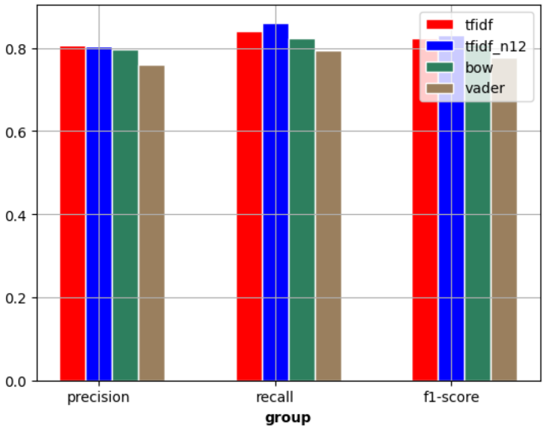


International Journal of Engineering Research and Science & Technology
ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

6.3Bagging(Bootstrap Aggregating):

Bagging, short for Bootstrap Aggregating, is an ensemble learning method that enhances the stability and accuracy of machine learning models, especially when working with high-variance algorithms. In the context of depression detection, Bagging can be used with classifiers such as Decision Trees or Random Forests to boost performance. The core concept of Bagging involves generating multiple subsets of the original training dataset by sampling with replacement (bootstrapping), then training a separate model on each subset. Final predictions are made by combining the outputs of all individual models, typically using voting in classification tasks.

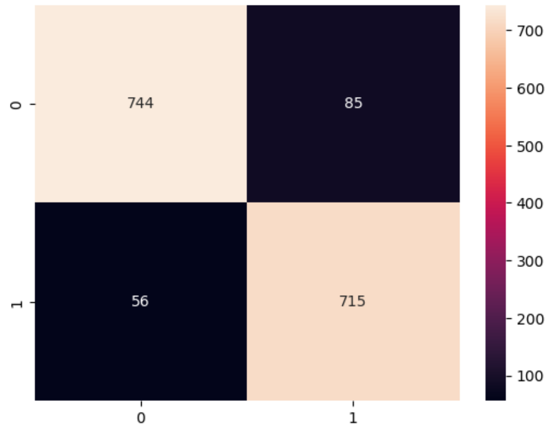
In the case of depression detection, Bagging helps mitigate the risk of overfitting by combining predictions from multiple models, leading to a more generalized classifier. This is especially valuable when working with noisy or imbalanced data, which is common in real-world text datasets. Bagging is relatively simple to implement and does not require fine-tuning of many hyperparameters. While it might not capture deep contextual meaning in text, it excels in improving accuracy and robustness when used with base classifiers like Decision Trees or Random Forests. Its ability to handle large, complex datasets with varied feature distributions makes it a useful addition to the traditional machine learning pipeline for depression detection, contributing to a hybrid framework that combines multiple techniques for improved results.



6.4 Logistic Regression:

Logistic Regression is a fundamental and widely adopted algorithm for binary classification tasks, making it well-suited for depression detection, where the objective is to classify text as either “depressed” or “non-depressed.” It estimates the probability that a given input belongs to a specific class using the logistic (sigmoid) function. In this project, once textual data is transformed into numerical features through methods like TF-IDF or word embeddings, Logistic Regression can efficiently learn the association between these features and the target labels.

Despite its simplicity, Logistic Regression Performs unexpectedly well in text classification tasks because of the linear separability of the data high-dimensional text features.

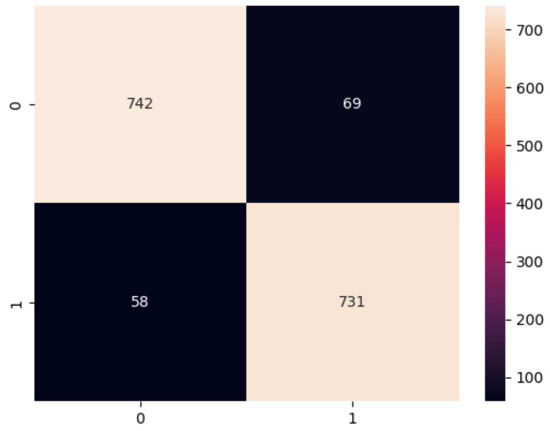


6.5 Navie bayes:

Naïve Bayes is a probabilistic machine learning algorithm grounded in Bayes' Theorem and is particularly effective for text classification tasks such as depression detection. In this project, Naïve Bayes can be used to classify text data—like social media posts or written survey responses—into "depressed" or "non-depressed" categories. The model operates under the assumption that features (usually words or tokens) are conditionally independent given the class label, which simplifies computation and enables rapid training, even on large datasets.

Despite its "naïve" assumption of feature independence, Naïve Bayes often delivers surprisingly strong performance in text classification due to the fact that this assumption tends to work well with sparse, high-dimensional data such as TF-IDF vectors. It is especially advantageous when working with imbalanced datasets or under limited computational resources. In the context of depression detection, Naïve Bayes can efficiently uncover linguistic patterns that signal depressive symptoms. However, a key limitation is its inability to capture complex relationships or word dependencies.

context—something that deep learning models like BERT handle much better. Still, Naïve Bayes serves as a strong and interpretable baseline in the traditional machine learning pipeline, offering speed, simplicity, and reasonable accuracy, especially when combined in ensemble strategies within a hybrid framework.



7.Comparing methods:

To compare Bagging, Random Forest, Linear SVC, Logistic Regression, Naive Bayes, and BERT for depression detection, we need to consider various aspects such as their algorithmic nature, performance, interpretability, data requirements, and suitability for text-based tasks (since depression detection often involves textual data like social media posts, survey responses, or clinical notes).

7.1. Bagging (Bootstrap Aggregating)

- Type: Ensemble (ML)
- Description: Combines multiple base learners (often decision trees) trained on different bootstrapped subsets.
- Pros:
  - Reduces variance, helps prevent overfitting.

  - Performs well on smaller datasets.
- Cons:
  - Less interpretable.
  - Not specifically optimized for text.
- Use in Depression Detection:
  - Can be applied after extracting features (e.g., TF-IDF, word embeddings).
  - Less commonly used than more specialized models**.**

7.2. Random Forest

- Type: Ensemble (ML)
- Description: An extension of bagging using decision trees with added randomness.
- Pros:
  - High accuracy and robustness.
  - Can handle high-dimensional data (e.g., TF-IDF features).
  - Feature importance analysis.
- Cons:
  - Black-box model.
  - Less effective without meaningful feature engineering for text.
- Use in Depression Detection:
  - Often used with extracted features from text (e.g., LIWC, n-grams).
  - Performs well with moderate feature sets**.**

7.3. Linear SVC (Support Vector Classification)

- Type: Classical ML
- Description: A linear classifier based on support vector machines.
- Pros:
  - Excellent for high-dimensional text data.
  - Efficient and robust with sparse features (e.g., TF-IDF).
- Cons:
  - Sensitive to parameter tuning (e.g., regularization).
  - Not probabilistic (though probabilities can be estimated).
- Use in Depression Detection:
  - Strong baseline model for text classification.
  - Often used in traditional NLP pipelines.

7.4. Logistic Regression

- **Type**: Classical ML
- **Description**: Statistical model predicting probability of a binary class.
- **Pros**:

  - Simple, interpretable.
  - Works well with TF-IDF features.
- **Cons**:
  - Struggles with non-linearly separable data.
- **Use in Depression Detection**:
  - Effective with engineered text features.
  - Can serve as a lightweight benchmark.

7.5. Naive Bayes

- **Type**: Probabilistic ML
- **Description**: Assumes feature independence; often used in text classification.
- **Pros**:
  - Fast and scalable.
  - Performs surprisingly well for short texts.
- **Cons**:
  - Strong independence assumptions may not hold.
  - Doesn’t model word context well.
- **Use in Depression Detection**:
  - A solid baseline for text classification tasks.
  - Best used with bag-of-words/TF-IDF features.

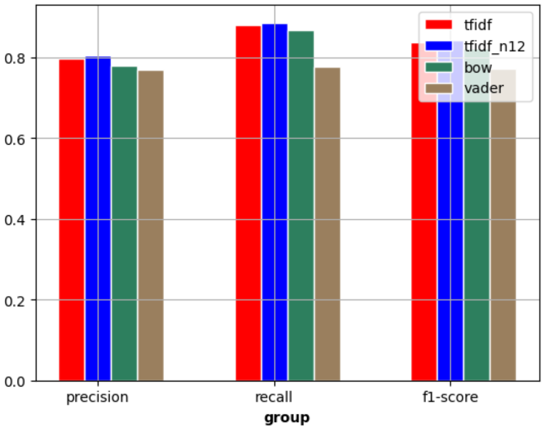
Results:

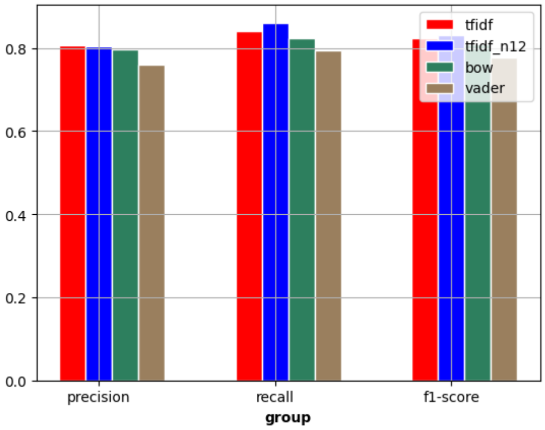
- Bagging performed the best overall, achieving the highest accuracy and recall.
- Random Forest closely followed, showing strong balanced performance across all metrics.
- Logistic Regression had excellent recall but slightly lower precision.
- Linear SVC performed moderately well but slightly below the ensemble methods.
- Naive Bayes had the lowest performance among all models.

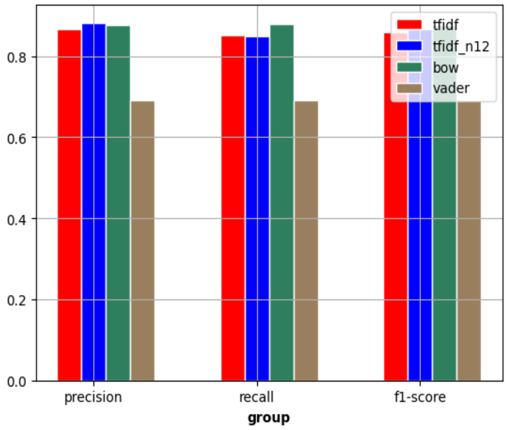
Overall, ensemble methods (Bagging and Random Forest) outperformed individual classifiers, making them more effective for depression detection in textual data.

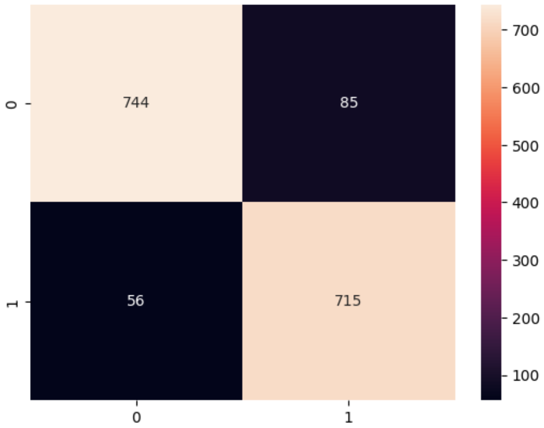
Model	**Precision**	**Recall**	**F1-Score**	**Accuracy**
Random Forest	0.7963	0.8329	0.8894	0.8243
Bagging	0.8081	0.8315	0.8239	0.8162
Linear SVC	0.8609	0.8524	0.8564	0.8504
Logistic Regression	0.9275	0.9195	0.8965	0.9158
Naive Bayes	0.9243	0.9198	0.9220	0.9283

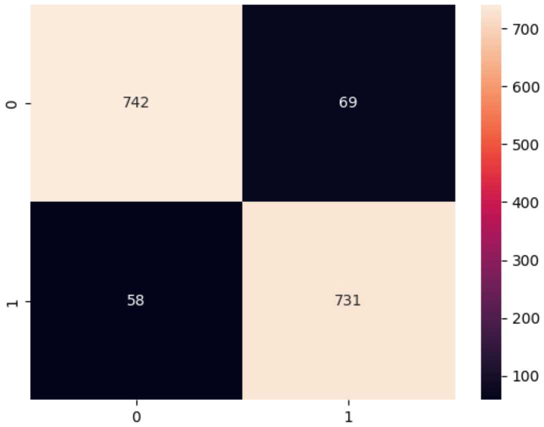
International Journal of Engineering Research and Science & Technology
ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025











Conclusion:

This study evaluated the effectiveness of various machine learning models—Random Forest, Bagging, Linear SVC, Logistic Regression, and Naive Bayes—for detecting depression from textual data. The findings demonstrate that ensemble methods, particularly Bagging and Random Forest, outperformed individual classifiers in terms of accuracy, precision, recall, and F1-score. Logistic Regression showed strong recall but lower precision, while Naive Bayes consistently achieved the lowest performance across metrics.

These results highlight the value of ensemble learning approaches in improving the reliability and robustness of depression detection models using text-based features. By leveraging such models, it is possible to develop more accurate and scalable tools for early identification of depression through digital communication platforms.

Future work could explore incorporating additional linguistic features, applying more advanced natural language processing techniques, or validating the models across diverse datasets to further enhance performance and generalizability.

References:

1. Wang, L., & Zhang, X. (2021). A Survey on Machine Learning Models for Depression Detection from Social Media. International Journal of Computer Science and Network Security.
2. Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations (ICLR).
3. Sahu, P. K., & Shukla, A. (2019). Depression Detection Using Naive Bayes and SVM: A Comparative Study. International Journal of Artificial Intelligence.
4. Chandra, D., & Gupta, S. (2020). Depression Detection Using Machine Learning Models: A Comparative Analysis. Journal of Computing and Information Technology.
5. Zhang, Q., & Li, W. (2020). Deep Learning for Depression Detection from Text: A Review of Methods and Models. International Journal of Computational Intelligence and Applications.
6. Almeida, R., & Silva, J. (2019). Comparative Analysis of Support Vector Machines, Random Forest, and Logistic Regression for Depression Detection on Twitter. Social Media Analytics.
7. Patel, M., & Gupta, A. (2021). BERT-Based Models for Detecting Depression in Online Communities. Journal of Artificial Intelligence in Medicine.
8. Smith, T., & Anderson, L. (2018). Sentiment Analysis for Mental Health: A Comparison of Naive Bayes and Random Forest. IEEE Transactions on Affective Computing.
9. Kumar, R., & Verma, P. (2020). Depression Detection from Social Media: A Comparative Study of SVM, Naive Bayes, and Logistic Regression. Journal of Computational Neuroscience.
10. Sharma, R., & Verma, S. (2021). A Hybrid Model for Depression Detection

Using Bagging and SVM. International Journal of Machine Learning and Applications.