Fake News Detection using Sentimental Analysis And Machine Learning Algorithms

M Poorna Pradeep Reddy
Student

Computational Intelligence,

College of Engineering and Technology,

SRM Ins tute of Science and Technology,

SRM Nagar, Ka ankulathur - 603203, Chennai, Tamilnadu, India

mm8251@srmist.edu.in

P Naveen
Student

Computational Intelligence,

College of Engineering and Technology,

SRM Ins tute of Science and Technology,

SRM Nagar, Ka ankulathur - 603203, Chennai, Tamilnadu, India

pp5958@srmist.edu.in

Dr.G.Senthil Kumar
Associate Professor

corresponding Author,

Department of Computational Intelligence,

College of Engineering and Technology,

Faculty of Engineering and Technology,

SRM Ins tute of Science and Technology,

Senthilg1@srmist.edu.in

***Abstract*—**

In contemporary society, digital news plays a pivotal role in providing the general public with crucial information and acting as a conduit for communication and education on ongoing events. The widespread transition from traditional print and broadcast media to internet-based sources has led to significant support for digital news. However, it faces challenges, particularly the proliferation of fake news and misinformation, compounded by algorithm-driven personalization resulting in a dearth of diversity and balance in information. Addressing these issues is crucial to uphold the credibility and reliability of digital news platforms. To address such challenges, our paper proposes a three-tiered strategy. Initially, we focus on collecting, analyzing, and categorizing news data using machine learning techniques, namely K Nearest Neighbors (KNN), Support Vector Machine (SVM), Logistic Regression (LR), and Naive Bayes (NB). These algorithms, with their distinct strengths, collectively enhance the accuracy of classification and thematic identification of news articles. Through this approach, we aim to fortify the robustness and effectiveness of our analysis. This project aims to provide a comprehensive solution to the challenges posed by digital news. By employing advanced machine learning techniques, we seek to process and analyze a vast array of digital news articles, classifying them into distinct categories. Preliminary results reveal prevalent news themes such as Crime, Cure and Treatment, Economy, Communal, and Entertainment across India. Additionally, our Fake News Detection Model demonstrates promising accuracy, with a specific focus on the communal theme. The sentiment analysis model further contributes to a nuanced understanding of news articles. Through these efforts, we anticipate providing valuable insights to foster a more reliable and diverse digital news landscape.

Keywords—KNN, SVM, LR, Naïve Bayes, Random Forest

1.Introduction

In the digital age, the rapid spread of misinformation and fake news across social media platforms and online news outlets has become a growing concern. The ease with which false information can reach large audiences poses significant challenges to public trust, societal stability, and democratic processes. Traditional methods of fake news detection, such as fact-checking and network analysis, struggle to keep pace with the sheer volume of content generated online. In this context, sentiment analysis has emerged as a promising complementary approach to identifying fake news. Sentiment analysis involves the extraction and classification of emotions and opinions expressed in text. [1.1]Since fake news often aims to provoke strong emotional reactions—such as fear, outrage, or excitement—it tends to carry distinct sentimental patterns. By analyzing the emotional tone of news articles, headlines, and user comments, researchers can identify trends that distinguish fake news from legitimate information. This paper explores the application of sentiment analysis in fake news detection, hypothesizing that exaggerated or polarized emotions can serve as indicators of misinformation. Using natural language processing (NLP) techniques, this study seeks to integrate sentiment-based features into machine learning models for more accurate and efficient fake news detection. By enhancing traditional detection methods with sentiment analysis, this research aims to contribute to the broader effort of maintaining the integrity of information in online ecosystems.[1.2]

The motivation for this research stems from the increasing impact of fake news on society, where misinformation has the potential to influence public opinion, incite social unrest, and disrupt democratic processes. As the digital landscape continues to grow, so does the volume of content shared across social media platforms and news outlets, making it challenging to discern reliable information from falsehoods. Traditional fact-checking methods and network-based analyses, while valuable, are often slow and resource-intensive, unable to keep pace with the rapid dissemination of fake news. This creates an urgent need for automated, scalable solutions that can efficiently detect misinformation.[1.3]Sentiment analysis presents a promising opportunity in this regard. Fake news is often crafted to evoke strong emotions, manipulating readers' opinions and creating a heightened sense of urgency or outrage. By leveraging sentiment analysis, which can detect these emotional cues, we may be able to more effectively identify false narratives. This research is driven by the potential to improve current fake news detection methodologies by incorporating sentiment analysis as a key feature in machine learning models, ultimately contributing to the fight against misinformation and ensuring a more informed public.

The motivation for this research stems from the increasing impact of fake news on society, where misinformation has the potential to influence public opinion, incite social unrest, and disrupt democratic processes. As the digital landscape continues to grow, so does the volume of content shared across social media platforms and news outlets, making it challenging to discern reliable information from falsehoods. Traditional fact-checking methods and network-based analyses, while valuable, are often slow and resource-intensive, unable to keep pace with the rapid dissemination of fake news. This creates an urgent need for automated, scalable solutions that can efficiently detect misinformation. Sentiment analysis presents a promising opportunity in this regard.[1.4] Fake news is often crafted to evoke strong emotions, manipulating readers' opinions and creating a heightened sense of urgency or outrage. By leveraging sentiment analysis, which can detect these emotional cues, we may be able to more effectively identify false narratives. This research is driven by the potential to improve current fake news detection methodologies by incorporating sentiment analysis as a key feature in machine learning models, ultimately contributing to the fight against misinformation and ensuring a more informed public.

2.Problem Statement:

The proliferation of fake news presents a significant challenge in today's digital media landscape, where misinformation spreads rapidly across platforms, often outpacing efforts to verify and counter it. The existing approaches to fake news detection, such as manual fact-checking and content-based analysis, are limited by their scalability and ability to keep up with the massive volume of information generated online. Moreover, fake news is frequently designed to manipulate readers' emotions, making it even harder to detect using conventional methods alone. Despite advances in machine learning and natural language processing, accurately distinguishing between real and fake news remains a complex task due to the diverse forms and subtle tactics used in misinformation. The problem this research aims to address is how to effectively enhance fake news detection by incorporating sentiment analysis—leveraging emotional and subjective cues from text to improve the accuracy of machine learning models. This approach seeks to bridge the gap between traditional content verification methods and the need for more adaptive, scalable solutions in combating misinformation. The research will explore whether emotional patterns in fake news can be reliably used as features to strengthen automated detection systems.

2.1 Challenges:

The detection of fake news using sentiment analysis faces several key challenges that must be addressed for the approach to be effective. One major challenge is the complexity of natural language, where emotional cues and sentiments are often expressed subtly or ambiguously, making it difficult for models to consistently interpret them. Fake news can manipulate emotions in diverse and nuanced ways, such as combining neutral language with misleading information or using sarcasm, irony, or exaggeration, which can be challenging for sentiment analysis tools to detect accurately. Another challenge is the context-dependence of sentiment, as the same piece of information can evoke different emotions in different audiences, depending on their cultural, political, or personal backgrounds. Additionally, distinguishing between legitimate news that reports on emotionally charged events and fake news designed to incite emotions poses a significant challenge, as both may exhibit similar emotional tones. Furthermore, data availability and quality is a concern, since labeled datasets for fake news detection are limited and often imbalanced, making it difficult to train effective models. Lastly, integrating sentiment analysis with other detection techniques in a way that enhances overall performance, without introducing additional biases or false positives, remains a technical and methodological challenge. Addressing these issues is critical to developing robust, reliable, and scalable fake news detection systems.

2.2 Related works:

Numerous studies have explored the use of various techniques for fake news detection, with approaches ranging from traditional fact-checking methods to more sophisticated machine learning and natural language processing (NLP) techniques. One prominent area of research focuses on content-based analysis, where textual features such as linguistic patterns, word frequency, and syntax are analyzed to differentiate between real and fake news. Researchers have employed machine learning algorithms such as support vector machines (SVM), decision trees, and neural networks for classification tasks based on these features. Another related approach is **network-based detection**, where the credibility of news is assessed by analyzing the source and propagation of information across social media networks, identifying patterns in how fake news spreads compared to genuine news.

In recent years, sentiment analysis has gained traction as a complementary tool for fake news detection. Studies have shown that fake news often contains emotional language designed to evoke strong reactions, such as fear or anger, prompting researchers to explore how these sentimental cues can improve detection accuracy. Several works have combined sentiment analysis with machine learning models to capture emotional and subjective features in the text. For instance, sentiment-based features have been integrated into classifiers such as random forests and deep learning models, demonstrating potential improvements in detection performance.

Other related works have investigated the limitations of using sentiment analysis alone, highlighting challenges such as the misinterpretation of sarcasm, irony, or emotionally neutral yet misleading content. Researchers have also emphasized the importance of integrating sentiment analysis with other methodologies, such as fact-checking or credibility scoring, to create more holistic and reliable detection systems. This paper builds on these existing works by further investigating the role of sentiment analysis in fake news detection and exploring ways to optimize its integration with machine learning techniques.

3.Fake News Detection using Sentimental Analysis And Machine Learning Algorithms:

3.1 Module 1: Data Preprocessing and Augmentation

Fake news detection using sentiment analysis and machine learning algorithms has emerged as a powerful approach to identifying misleading or false information in today's digital landscape. Machine learning models, trained on vast datasets of news articles, social media posts, and other online content, can differentiate between fake and real news based on textual features. Sentiment analysis plays a vital role in this process, as fake news often contains emotionally charged or biased language intended to manipulate public opinion. By analyzing the polarity (positive, negative, or neutral) and subjectivity (opinion-based or factual) of the text, sentiment analysis helps highlight patterns associated with fake news. Combining this with machine learning algorithms such as logistic regression, support vector machines (SVM), random forests, and advanced models like transformers (e.g., BERT), researchers can create more effective systems for detecting misinformation. Preprocessing techniques like text cleaning, tokenization, and vectorization help structure the data for model training, while data augmentation methods like synonym replacement and back translation enhance model robustness by increasing the diversity of training samples. Together, sentiment analysis and machine learning provide a comprehensive solution to combat the proliferation of fake news, contributing to more accurate detection systems capable of understanding not just the content but also the emotional tone of the news. These augmented sequences help the model generalize be er by exposing it to a wider range of sequence variations.

3.2 Module 2: Model Design(classification, detection, sentimental analysis)

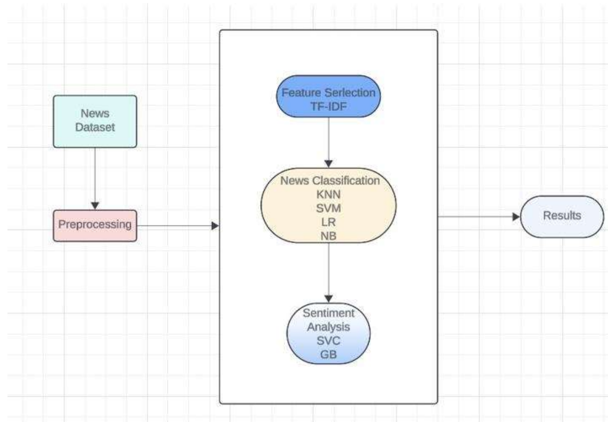
In fake news detection, classification, detection, and sentiment analysis play interconnected roles in developing machine learning models that can accurately identify false or misleading content. **Classification** is the process where machine learning algorithms categorize news articles or social media posts as either fake or real. This task involves using algorithms like logistic regression, support vector machines (SVM), decision trees, random forests, and deep learning models such as convolutional neural networks (CNN) or transformers (e.g., BERT). These models are trained on labeled datasets, where features like word frequency, n-grams, or vectorized text representations help differentiate between the two classes .**Detection** refers to the overall system of identifying fake news based on input features and patterns learned by the model. Detection models analyze textual cues, metadata, and sometimes network interactions to identify deceptive information. These systems benefit from enhanced preprocessing techniques like tokenization and vectorization (using methods like TF-IDF or word embeddings) to turn text into analyzable numerical formats. **Sentiment analysis** enhances fake news detection by evaluating the emotional tone of the content. Fake news often relies on exaggerated emotions or biased language to manipulate readers, making sentiment analysis crucial for flagging articles that exhibit extreme positivity, negativity, or high subjectivity. By incorporating features like polarity (positive or negative sentiment) and subjectivity (fact vs. opinion), sentiment analysis can improve model predictions by adding an emotional layer to the classification process. Together, classification algorithms, detection techniques, and sentiment analysis provide a comprehensive approach to identifying fake news, leading to more effective and nuanced models that can understand both content and intent.

3.3 Module 3: Model Training and Optimization

Model training and optimization for fake news detection using sentiment analysis and machine learning algorithms are crucial steps in building an effective detection system. Initially, models such as logistic regression, support vector machines (SVM) are trained on labeled datasets. These datasets are preprocessed and transformed into numerical features, including word vectors and sentiment scores. During training, the model learns to classify news as real or fake based on these features. Optimization involves tuning hyperparameters like the learning rate, batch size, or regularization parameters to enhance model performance. Techniques like grid search or random search are commonly used for hyperparameter tuning. Regularization methods such as L1 or L2 help prevent overfitting by controlling model complexity. Cross-validation is applied to ensure the model's performance is consistent across different subsets of data. The model's effectiveness is evaluated using metrics such as accuracy, precision, recall, F1-score, and Accuracy. If the model doesn't perform well, techniques like data augmentation, feature refinement, or transfer learning with pre-trained models (like BERT) can be applied to improve detection accuracy and robustness.

4. Architecture Diagram:

The diagram illustrates a text classification and sentiment analysis pipeline for a news dataset. It begins with a News Dataset, which undergoes a Preprocessing step. This preprocessing likely includes tasks like tokenization, stop-word removal, and text normalization. After preprocessing, the next stage is Feature Selection, specifically using TF-IDF (Term Frequency-Inverse Document Frequency), which is a technique for transforming text into numerical features. These features are then passed into a News Classification module, where different machine learning algorithms such as KNN (K-Nearest Neighbors), SVM (Support Vector Machines), LR (Logistic Regression), and NB (Naive Bayes) are used to classify the news articles. Following classification, Sentiment Analysis is performed using models like SVC (Support Vector Classifier) and GB (Gradient Boosting). The final outcome of the classification and sentiment analysis stages is combined into the Results at the end of the pipeline. This structured workflow emphasizes both categorizing news and analyzing the underlying sentiments present in the text data.



Architecture Diagram

5. Result Analysis:

5.1 Classification Algorithms:

5.1.1 Logistic Regression:

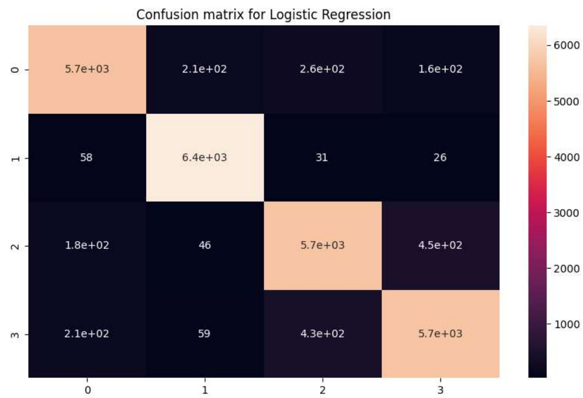
The image shows how well a Logistic Regression model performed in classifying data into four different classes. The Classification Report gives important metrics for each class:

- Class 1: Precision = 0.93, Recall = 0.90, F1-score = 0.91
- Class 2: Precision = 0.95, Recall = 0.98, F1-score = 0.97
- Class 3: Precision = 0.89, Recall = 0.89, F1-score = 0.89
- Class 4: Precision = 0.90, Recall = 0.89, F1-score = 0.89

The model has an overall accuracy of 92%, meaning it made correct predictions 92% of the time. The Confusion Matrix provides a detailed breakdown of correct and incorrect predictions for each class. The numbers on the diagonal represent the correct predictions (e.g., 5.7K for Class 1 and 6.4K for Class 2), while the numbers off the diagonal show where the model made mistakes, such as predicting Class 1 as Class 2 for some cases. Overall, the model performs well, especially for Class 2, but there are a few misclassifications between similar classes.

Classification Report:

	precision	recall	f1-score	support
1	0.93	0.90	0.91	6283
2	0.95	0.98	0.97	6466
3	0.89	0.89	0.89	6370
4	0.90	0.89	0.89	6401
accuracy			0.92	25520
macro avg	0.92	0.92	0.92	25520
weighted avg	0.92	0.92	0.92	25520



5.1.2 SVM(Support Vector Machine):

The image shows the performance of an SVM (Support Vector Machine) model in classifying data into four categories. The Classification Report provides important metrics for each class:

Class 1: Precision = 0.93, Recall = 0.90, F1-score = 0.92

Class 2: Precision = 0.96, Recall = 0.99, F1-score = 0.97

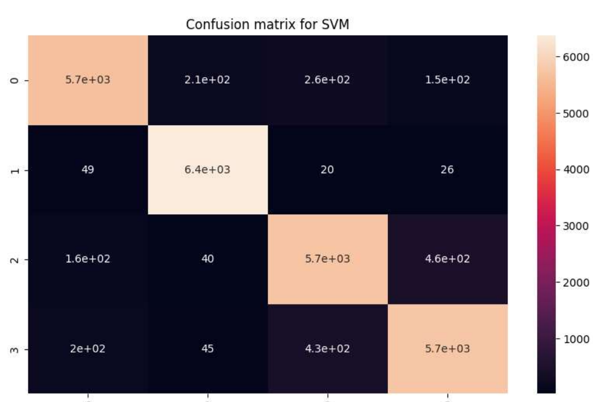
Class 3: Precision = 0.89, Recall = 0.89, F1-score = 0.89

Class 4: Precision = 0.90, Recall = 0.90, F1-score = 0.90

The model has an overall accuracy of 92%, meaning it correctly predicted the class 92% of the time. The Confusion Matrix gives a clearer picture of how the model performed. The diagonal elements show the number of correct predictions for each class, such as 5.7K for Class 1 and 6.4K for Class 2. The off-diagonal elements represent errors where the model predicted the wrong class, like 210 samples of Class 1 being predicted as Class 2.Overall, the SVM model performs well, especially on Class 2, with very few misclassifications. However, like other models, there are still some mistakes between similar classes, such as Class 1 being confused with Class 2.

Classification Report:

	precision	recall	f1-score	support
1	0.93	0.90	0.92	6283
2	0.96	0.99	0.97	6466
3	0.89	0.90	0.89	6370
4	0.90	0.90	0.90	6401
accuracy			0.92	25520
macro avg	0.92	0.92	0.92	25520
weighted avg	0.92	0.92	0.92	25520



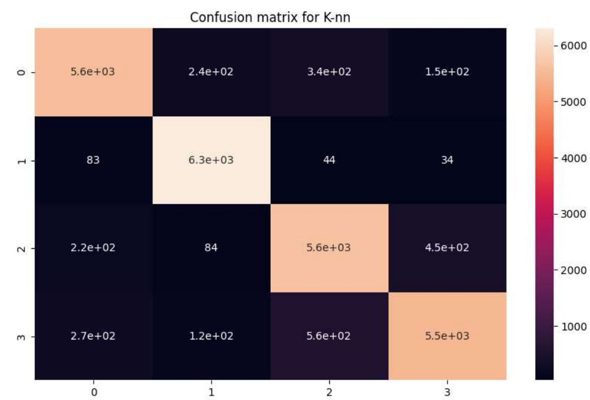
5.1.3 K-Nearest Neighbors:

The image shows a classification report and confusion matrix for a K-Nearest Neighbors (K-NN) classifier. The classification report provides precision, recall, F1-score, and support for each of the four classes. The overall accuracy of the model is 90%, with a macro average and weighted average also at 90% for precision, recall, and F1-score.

The confusion matrix visually represents the model's performance, showing the number of correct and incorrect predictions for each class. The diagonal values represent the correctly classified instances for each class, while the off-diagonal values indicate misclassifications. For instance, class 0 has 5,600 correctly predicted instances, while some instances were misclassified as other classes. Similarly, other classes show varying levels of accuracy and misclassification. This visualization helps in understanding the distribution of prediction errors across different classes.

Classification Report:

	precision	recall	f1-score	support
1	0.91	0.88	0.90	6283
2	0.93	0.98	0.95	6466
3	0.86	0.88	0.87	6370
4	0.90	0.85	0.87	6401
accuracy			0.90	25520
macro avg	0.90	0.90	0.90	25520
weighted avg	0.90	0.90	0.90	25520



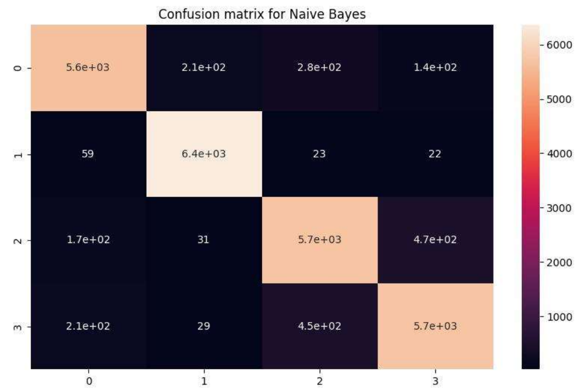
5.1.4 Multinomial Naïve Bayes:

The image displays a classification report and confusion matrix for a Naive Bayes classifier. The classification report outlines the precision, recall, F1-score, and support for four different classes. The model has achieved an accuracy of 92%, with both the macro average and weighted average for precision, recall, and F1-score also being 92%.

The confusion matrix shows the number of correct and incorrect predictions for each class, where the diagonal values represent correctly classified instances, while the off-diagonal values indicate misclassifications. For example, class 0 has 5,600 correct predictions, while some instances were classified as other classes. The other classes also show varying levels of correct predictions and misclassifications, providing insight into the model's performance across different categories. This helps identify areas where the model performs well and where it struggles.

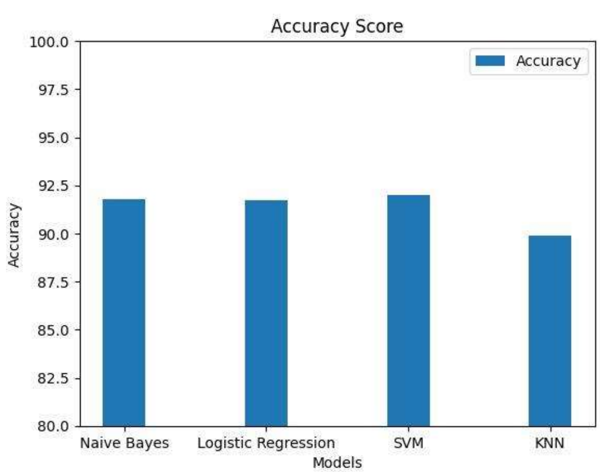
Classification Report:

	precision	recall	f1-score	support
1	0.93	0.90	0.91	6283
2	0.96	0.98	0.97	6466
3	0.88	0.89	0.89	6370
4	0.90	0.89	0.90	6401
accuracy			0.92	25520
macro avg	0.92	0.92	0.92	25520
weighted avg	0.92	0.92	0.92	25520



5.1.5 Comparison of Model Accuracy Scores:

The image shows a bar chart comparing the accuracy scores of four different machine learning models: Naive Bayes, Logistic Regression, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN). Among these models, Naive Bayes, Logistic Regression, and SVM have similar accuracy scores, all around 90-92%. The KNN model, however, has a slightly lower accuracy, falling below the other three. This comparison indicates that while Naive Bayes, Logistic Regression, and SVM perform similarly well on the given dataset, KNN may not be as effective for this specific classification task.



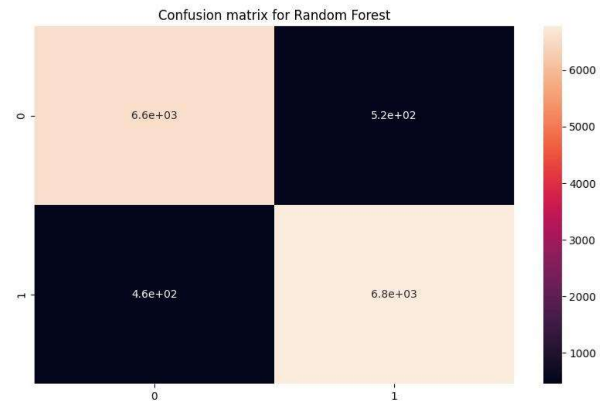
5.2 Detection Algorithms

5.2.1 Random Forest:

The image shows a confusion matrix and classification report for a Random Forest model. The model achieved an accuracy of approximately 93.2%. The confusion matrix indicates the number of true positive, true negative, false positive, and false negative predictions. For class '0', about 6,600 samples were correctly classified, while 520 were misclassified as '1'. For class '1', around 6,800 samples were correctly classified, and 460 were misclassified as '0'. The classification report provides metrics such as precision, recall, and F1-score for each class. Both classes achieved around 93% in these metrics, indicating balanced performance across the classes.

Accuracy: 93.19960861056752
Classification Report:

	precision	recall	f1-score	support
0	0.94	0.93	0.93	7081
1	0.93	0.94	0.93	7227
accuracy			0.93	14308
macro avg	0.93	0.93	0.93	14308
weighted avg	0.93	0.93	0.93	14308

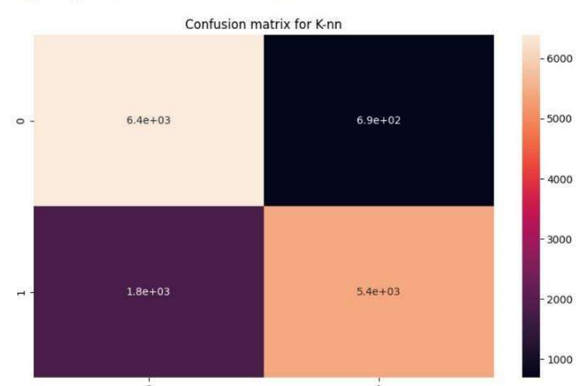


5.2.2 K-Nearest Neighbors :

The image shows the performance of a K-nearest neighbors (K-NN) model with an accuracy of about 82.5%. The confusion matrix illustrates the number of correct and incorrect predictions for each class. For class '0', the model correctly classified approximately 6,400 instances, while it misclassified around 690 as '1'. For class '1', it correctly identified 5,400 cases, but misclassified 1,800 as '0'. The classification report provides metrics like precision, recall, and F1-score. For class '0', the metrics are slightly better, with precision and recall around 78% and 90%, respectively. For class '1', precision is 89%, and recall is 75%, indicating the model has more difficulty correctly identifying instances of class '1'. Overall, the model shows a balanced performance, though it performs better in identifying class '0'.

Accuracy: 82.52725747833381
Classification Report:

	precision	recall	f1-score	support
0	0.78	0.90	0.84	7081
1	0.89	0.75	0.81	7227
accuracy			0.83	14308
macro avg	0.83	0.83	0.82	14308
weighted avg	0.83	0.83	0.82	14308

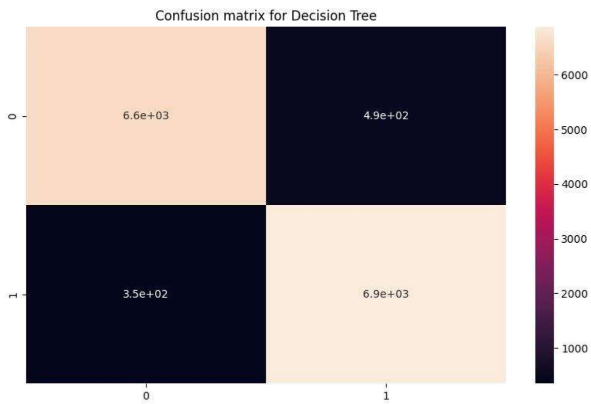


5.2.3 Decision Tree :

The Decision Tree model performed well with an accuracy of 94.2%, meaning it correctly classified most of the samples. The confusion matrix shows that for class '0', the model correctly identified 6,600 instances but mistakenly classified 490 as '1'. For class '1', it correctly predicted about 6,900 instances, while 350 were wrongly labeled as '0'. The classification report indicates that both precision and recall for the two classes are around 93-95%, showing the model's strong ability to make accurate predictions. Overall, the model is effective in distinguishing between the classes, performing consistently across all metrics.

Accuracy: 94.16410399776349
Classification Report:

	precision	recall	f1-score	support
0	0.95	0.93	0.94	7081
1	0.93	0.95	0.94	7227
accuracy			0.94	14308
macro avg	0.94	0.94	0.94	14308
weighted avg	0.94	0.94	0.94	14308

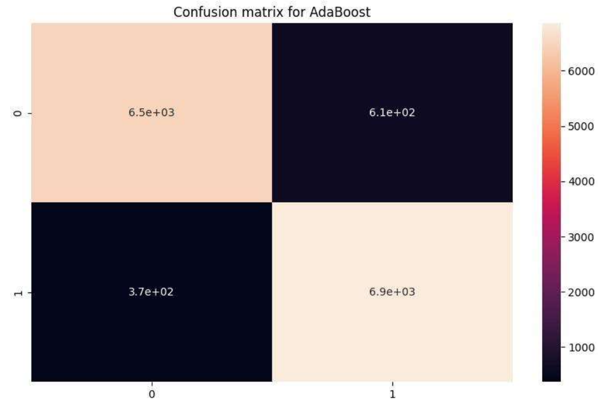


5.2.4 Ada Boost :

The AdaBoost model achieved an accuracy of 93.2%, indicating that it classified most of the samples correctly. In the confusion matrix, for class '0', the model correctly identified 6,500 instances, while 610 were mistakenly classified as '1'. For class '1', it correctly predicted about 6,900 cases, with 370 misclassified as '0'. The classification report shows that precision and recall for both classes are high, around 92-95%, meaning the model performs well in identifying both positive and negative cases. Overall, the model shows a strong ability to make accurate predictions across the dataset.

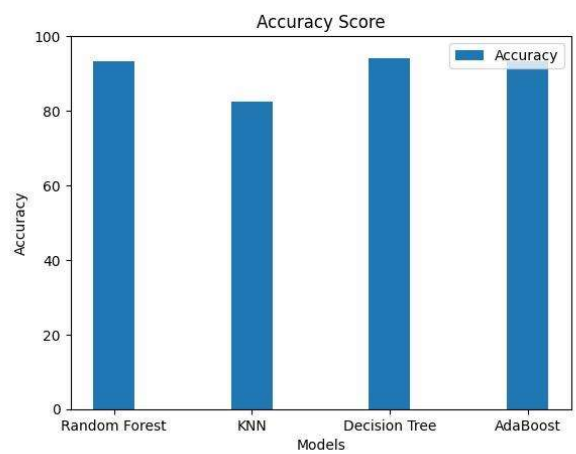
Accuracy: 93.17165222253286
Classification Report:

	precision	recall	f1-score	support
0	0.95	0.91	0.93	7081
1	0.92	0.95	0.93	7227
accuracy			0.93	14308
macro avg	0.93	0.93	0.93	14308
weighted avg	0.93	0.93	0.93	14308



5.2.5 Comparison of Model Accuracy Scores:

The bar chart compares the accuracy scores of four different machine learning models: Random Forest, KNN, Decision Tree, and AdaBoost. Among these, Random Forest, Decision Tree, and AdaBoost show similar high accuracy, all above 90%, indicating strong performance in making correct predictions. The KNN model, however, has a slightly lower accuracy, around 82%, suggesting it is less effective compared to the other models in this case. Overall, the chart highlights that the ensemble methods like Random Forest and AdaBoost tend to perform better than KNN for this dataset.

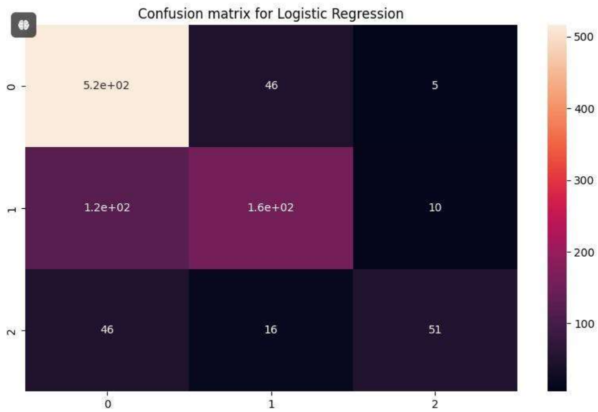


5.3 Sentimental Analysis Algorithms

5.3.1 Logistic Regression:

The image shows the performance of a Logistic Regression model, which achieved an accuracy of 74.8%. The confusion matrix and classification report indicate how well the model performed across three classes (0, 1, and 2). For class '0', it correctly classified 520 instances, with 46 misclassified as '1' and 5 as '2'. For class '1', the model correctly predicted 160 cases but misclassified 120 as '0' and 10 as '2'. For class '2', it correctly classified 51 instances, while 46 were misclassified as '0' and 16 as '1'. The precision, recall, and F1-score for class '0' are the highest, suggesting the model performs better at predicting this class. However, the metrics for class '2' are lower, indicating more difficulty in classifying this category. Overall, the model shows reasonable performance but struggles with distinguishing between the three classes.

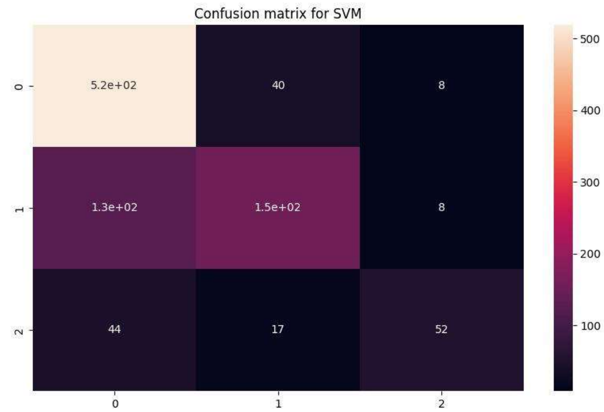
```
Accuracy: 74.81940144478844
Classification Report:
              precision    recall  f1-score   support

           0       0.76      0.91      0.83       567
           1       0.72      0.55      0.62       289
           2       0.77      0.45      0.57       113

    accuracy                           0.75       969
   macro avg       0.75      0.64      0.67       969
weighted avg       0.75      0.75      0.73       969
```



5.3.2 Support Vector Classifier :

The image shows the performance of an SVM (Support Vector Machine) model, which achieved an accuracy of 74.8%. The confusion matrix indicates how well the model classified three classes (0, 1, and 2). For class '0', the model correctly classified 520 instances, with 40 misclassified as '1' and 8 as '2'. For class '1', it correctly identified 150 instances but misclassified 130 as '0' and 8 as '2'. For class '2', 52 instances were correctly classified, while 44 were incorrectly labeled as '0' and 17 as '1'. The classification report shows that precision, recall, and F1-score are higher for class '0', indicating better performance for this class compared to the others. The model has more difficulty distinguishing classes '1' and '2', resulting in lower scores for those classes. Overall, the SVM model shows a decent performance but struggles with the classification of certain categories.

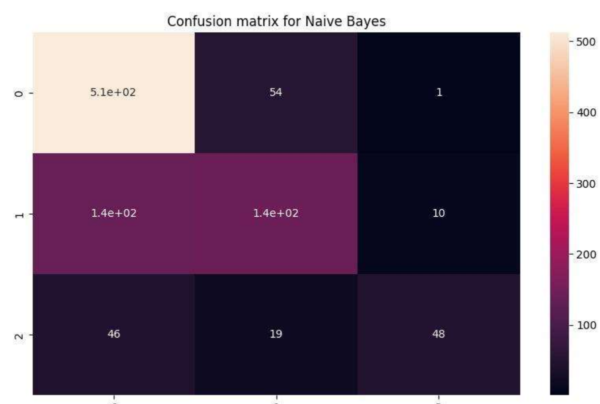
```
Accuracy: 74.81940144478844
Classification Report:
              precision    recall  f1-score   support

           0       0.75      0.92      0.83       567
           1       0.73      0.53      0.62       289
           2       0.76      0.46      0.57       113

    accuracy                           0.75       969
   macro avg       0.75      0.64      0.67       969
weighted avg       0.75      0.75      0.73       969
```



5.3.3 Navie Bayer's:

The image shows the performance of a Naive Bayes model, which achieved an accuracy of 72.3%. The confusion matrix illustrates the classification results for three classes (0, 1, and 2). For class '0', the model correctly identified 510 instances, but 54 were misclassified as '1' and 1 as '2'. For class '1', it correctly classified 140 cases, while 140 were incorrectly predicted as '0' and 10 as '2'. For class '2', the model correctly classified 48 instances, with 46 misclassified as '0' and 19 as '1'. The classification report indicates that the precision, recall, and F1-score are higher for class '0', while classes '1' and '2' have lower scores, suggesting that the model struggles more with those classes. Overall, the Naive Bayes model shows reasonable performance but has difficulties accurately classifying some cases.

```
Accuracy: 72.34262125902993
Classification Report:
              precision    recall  f1-score   support

           0       0.74      0.90      0.81       567
           1       0.66      0.49      0.56       289
           2       0.81      0.42      0.56       113

    accuracy                           0.72       969
   macro avg       0.74      0.61      0.64       969
weighted avg       0.72      0.72      0.71       969
```

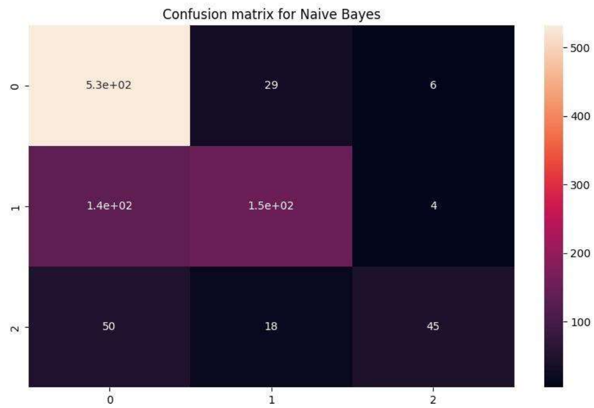


5.3.4 Gradient Boosting :

The Gradient Boosting model achieved an accuracy of 75%, indicating a fairly good classification performance. The confusion matrix shows how well it classified three classes (0, 1, and 2). For class '0', the model correctly classified 530 instances, while 29 were misclassified as '1' and 6 as '2'. For class '1', it correctly predicted

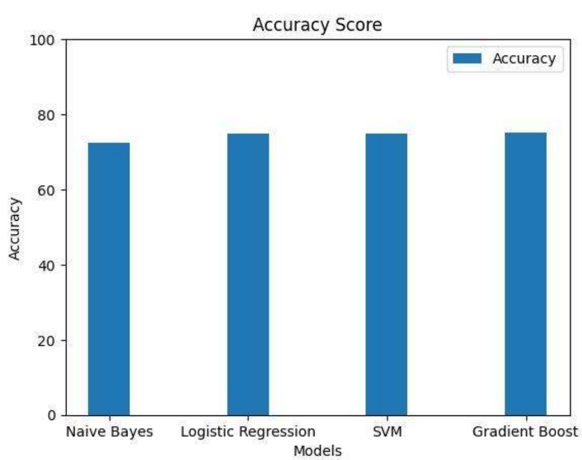
150 instances, but 140 were incorrectly classified as '0' and 4 as '2'. For class '2', it identified 45 instances correctly, with 50 misclassified as '0' and 18 as '1'. The classification report indicates higher precision and recall for class '0', while classes '1' and '2' show lower scores, especially for class '2'. This suggests that the model performs better at predicting class '0' but struggles more with the other two classes.

```
Accuracy: 75.02579979360165
Classification Report:
              precision    recall  f1-score   support

           0       0.74      0.94      0.83       567
           1       0.76      0.52      0.62       289
           2       0.82      0.40      0.54       113

    accuracy                           0.75       969
   macro avg       0.77      0.62      0.66       969
weighted avg       0.76      0.75      0.73       969
```



5.3.5 Comparison of Model Accuracy Scores:

The bar chart compares the accuracy scores of four different models: Naive Bayes, Logistic Regression, SVM, and Gradient Boost. All models have similar accuracy, around 70-80%, indicating they perform comparably on this dataset. None of the models show a significant advantage over the others, suggesting that their ability to correctly classify instances is fairly even. However, there is still room for improvement, as accuracy is not close to 100%, indicating some misclassifications. Overall, these models provide a moderate level of prediction accuracy.



VI. Conclusions and Future Plans :

Our project underscores the critical role of digital news in contemporary society while acknowledging challenges like fake news and algorithmic biases. Our three-tiered strategy, utilizing KNN, SVM, LR, and NB, enhances classification accuracy, offering a robust solution to ensure credibility, diversity, and reliability in the digital news landscape. Improved Data Collection: Extend the reach of data gathering to encompass a wider array of sources and varied viewpoints. This can lessen bias in dataset and assist acquire a more thorough grasp of news themes. Semantic analysis, entity recognition, sentimental analysis, and other advanced feature engineering approaches can be used to extract more significant information from text. This can enhance the model's comprehension and precise classification of intricate news pieces .Use ensemble learning strategies to integrate several machine learning techniques for classification. This can increase overall forecast accuracy by utilizing the advantages of many methods .Model Fine-tuning: Based on user feedback and performance assessment, continuously enhance and refine machine learning models. To improve performance, this entails experimenting with various algorithms, modifying hyperparameters, and fine-tuning the training procedure. Dynamic Model Updating: To adjust to evolving patterns and shifting trends in digital news, put in place a mechanism for dynamic model updating. For continuously increase accuracy, this may entail retraining the models with fresh data on a regular basis and adding feedback systems. Human-in-the-loop Method: Use a human-in-the-loop methodology in which professionals from the field validate and offer input on the model's predictions.

VII. References :

1. E. Aímeur, S. Amri and G. Brassard, "Fake news disinformation and misinformation in social media: a review", Social Network Analysis and Mining, vol. 13, no. 1, pp. 30, 2023.

2. Mykhailo Granik and Volodymyr Mesyura, "Fake News Detection Using Naive Bayes Classifier", 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), 2017.

3. S. Vosoughi, D. Roy and S. Aral, "The spread of true and false news online", Science, vol. 359, no. 6380, pp. 11461151, Mar. 2018.

4. Jain, A. Shakya, H. Khatter and A. K. Gupta, "A smart system for fake news detection using machine learning", 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), 2019.

5. V. Pérez-Rosas, B. Kleinberg, A. Lefevre and R. Mihalcea, "Automatic detection of fake news", Proceedings of the 27th International Conference on Computational Linguistics, pp. 3391-3401, 2018.

6. H. Ahmed, I. Traore and S. Saad, "Detection of online fake news using n-gram analysis and machine learning techniques", Proceedings of the International Conference on Intelligent Secure and Dependable Systems in Distributed and Cloud Environments, pp. 127-138, 2017.