Cureus Journal of Computer Science

Open Access Original Article

Live News Classification Using Naive Bayes Classifier

Dhruv S. Kashid [1] , Janhavi D. Patil [1] , Amar Buchade [1]

1. Artificial Intelligence and Data Science, Vishwakarma Institute of Information Technology, Pune, IND

Corresponding authors: Dhruv S. Kashid, dhruvkashid1027@gmail.com, Janhavi D. Patil, janhavipatil9702@gmail.com, Amar Buchade, amar.buchade@viit.acin

Received 10/28/2024
Review began 10/30/2024
Review ended 01/07/2025
Published 01/08/2025

DOI: https://doi.org/10.7759/s44389-024-01030-8

Abstract

In today's fast-paced digital environment, where information spreads at an unprecedented rate, the accurate and efficient classification of live news streams has become crucial. With the rise of various online platforms, users demand quick access to trustworthy, topic-specific news while reducing their exposure to misinformation. This paper investigates the categorization of live news articles into various topics, including sports, politics, technology, and entertainment, utilizing the naive Bayes classifier due to its ease of implementation and computational efficiency. The naive Bayes algorithm is particularly well suited for real-time applications, allowing for rapid processing with minimal delay.

By comparing naive Bayes with other machine learning techniques, such as support vector machines and decision trees, the study highlights the competitive accuracy of naive Bayes and its notable advantages in speed, affirming its appropriateness for real-time use. The findings indicate that naive Bayes is both scalable and lightweight, effectively addressing the unique challenges associated with live news streams, including rapidly evolving content, noisy data, and ambiguous terminology. These insights provide valuable implications for real-time news systems and suggest avenues for future improvements to enhance adaptability and robustness in continuously changing news environments.

Categories: Multi-Agent Systems, AI applications, Computer Architecture
Keywords: news categorization, sentiment analysis, topic modeling, real-time news analysis, event detection, breaking news identification, fake news detection

Introduction

The rapid growth of digital news platforms has led to an overwhelming influx of news content for readers each day. Although this abundance of information is beneficial, it creates a significant challenge in helping users locate articles that match their interests. Efficient news classification - categorizing articles into segments such as politics, sports, entertainment, health, technology, and education - is essential for creating a more organized and tailored news consumption experience. This study presents a method aimed at optimizing and refining the sorting of news articles by utilizing the naive Bayes classifier in conjunction with the CountVectorizer, with the goal of enhancing both the accuracy and efficiency of news categorization.

A literature survey on news classification with naive Bayes highlights its significance at the intersection of natural language processing (NLP) and machine learning. Naive Bayes is recognized for its straightforwardness and effectiveness, making it a strong candidate for news categorization tasks. This study also examines various naive Bayes variants, such as Multinomial and Bernoulli Naive Bayes, while tackling challenges associated with large-scale, multi-class datasets and addressing issues related to class imbalances. The combination of naive Bayes with other techniques, such as deep learning, has shown promise in improving news classification. This survey summarizes trends, challenges, and future directions in naive Bayes-based news classification, reaffirming its relevance in the evolving field of information categorization. Dadgar et al. [1] present a high precision rates (97.48% for BBC and 94.93% for 20Newsgroup datasets), identifying sports and politics as the best-performing categories. Term frequency-inverse document frequency (TF-IDF) and text classification using support vector machine (SVM) is used. The study by Irfani et al. [2] represents that adding queries in the form of hypernyms and hyponyms to tweets for news classification using naive Bayes reduces accuracy, possibly due to translation issues and a lack of contextual relevance. This highlights the need for language-specific resources, such as an Indonesian-based WordNet. The method was implemented using naive Bayes and hypernym-hyponym-based feature expansion. A key challenge identified in the research conducted by Shahi and Pant [3] is the presence of imbalanced datasets, which can result in underflow and overflow problems, ultimately impacting the accuracy of fake news detection. Yasawi et al. [4] combine machine learning algorithms, NLP, and evaluation metrics to develop a system for detecting fake news. These technologies and methods are commonly used in text classification tasks and aim to provide a reliable solution for discerning between fake and real news articles. Machine learning methods, particularly neural networks and CNNs, are used in these tasks. Minaee et al. [5] emphasize the use of deep learning models to address various text classification tasks and

How to cite this article
Kashid D S, Patil J D, Buchade A (January 08, 2025) Live News Classification Using Naive Bayes Classifier . Cureus J Comput Sci 2 : es44389-024-01030-8. DOI https://doi.org/10.7759/s44389-024-01030-8

provide insights into their technical contributions and performance on benchmark datasets. It contributes to the ongoing research in NLP and deep learning for text analysis. The observation [6] here underscores that the micro-averaging F-score is more influenced by classification performance on common classes, and the macro-averaging score is influenced by classification performance on rare classes. Least squares twin support vector machine has better generalization ability and computational speed as compared to least square support vector machine and twin support vector machine for news categorization problem [7].

In the upcoming sections of this paper, we will delve into the methodology of employing naive Bayes and CountVectorizer to effectively classify news articles into their respective categories. By leveraging NLP techniques and rigorous statistical analysis, our aim is to construct a news classification model that not only enriches the user experience but also aids content providers and researchers in comprehending news consumption patterns.

Materials And Methods

Dataset creation

The dataset for news classification was created manually by referring to the following newspapers: Hindustan Times, The Times of India, Indian Express, and News Today.

The attributes of the dataset include date, news, news category, and newspaper name. The news attribute includes the news statement. The category attribute includes sports, business, health, lifestyle, technology, crime, world, political, India news, UPSC specials, etc. The news was collected from 25th September 2023 to 25th November 2023. The total number of news items collected each day varied.

Methodology for real-time news article classification using naive Bayes is outlined in the following subsections.

Data collection

To maintain a high-quality dataset for sentiment and topic classification, data are collected from reliable sources such as leading news agencies (e.g., Reuters, BBC, Associated Press) through web scraping or API integration. For this process, popular RSS (Really Simple Syndication) feeds or open APIs, like the Google News API, may also be used to ensure real-time data acquisition and timeliness of the dataset.

Challenges: Real-time data collection must address issues of API rate limits, data consistency, and reliability.

Tools: Python libraries like BeautifulSoup and Scrapy, or APIs such as NewsAPI, offer efficient methods for gathering textual data from diverse sources.

Data preprocessing

Effective text preprocessing is essential for reducing noise and preparing text data for reliable feature extraction and classification. Standard preprocessing methods are as follows:

HTML tag removal: Stripping away HTML and XML tags ensures that only the primary content remains, removing irrelevant markup.

Tokenization: The text is divided into individual tokens - words or phrases - enabling a detailed analysis at the word level.

Stopword removal: Common, non-informative words (such as "the" and "and") are removed using libraries like NLTK or spaCy to enhance the signal of meaningful terms.

Stemming and lemmatization: This technique involves reducing words to their fundamental or root forms, which helps the model recognize different variations of a word as equivalent.

Removal of special characters and punctuation: This process entails eliminating special characters and punctuation marks to reduce noise and enhance the accuracy of the model.

Such a preprocessing strategy enhances the reliability and consistency of machine learning models, facilitating more precise feature extraction and classification in subsequent stages.

Feature extraction

To convert textual information into numerical features suitable for machine learning, specific techniques must be applied:

Bag-of-words (BoW): This approach transforms each document into a vector reflecting word frequencies, capturing the frequency of words present in the documents, which aids classification efforts.

TF-IDF: This method assigns importance to words based on their frequency across documents, thereby mitigating the influence of frequently occurring but less informative terms.

These transformations generate a numerical representation that conveys the thematic core of each document, facilitating effective text analysis and classification.

Naive Bayes classifier for text classification

The naive Bayes algorithm, celebrated for its straightforwardness and efficacy in text classification, is trained on labeled datasets that categorize different news topics.

Training process: The classifier is trained using a labeled dataset with predefined categories (e.g., politics, health, technology). The model calculates the prior probability for each category.

Likelihood calculation: By computing word-category likelihoods, the model assesses the probability of specific words appearing in each category.

Classification: Using Bayes' theorem, the classifier predicts the category of new, unseen articles based on calculated probability estimates.

Tools: The model is implemented with Python’s scikit-learn library, offering a robust, efficient naive Bayes classifier suited for text data.

Model evaluation

Assessing the performance of a model is vital for validating its accuracy and dependability in practical scenarios:

Cross-validation: This technique consists of splitting the dataset into training and testing groups, allowing for an evaluation of the model's ability to generalize across various data samples.

Performance metrics: Employing metrics such as accuracy, precision, recall, F1-score, and the confusion matrix provides a comprehensive understanding of the model's performance. These metrics are critical for assessing the model's suitability for real-world applications.

Implementation

Key libraries in Python streamline the process:

NLTK: Aids in preprocessing (tokenization, stopword removal).

scikit-learn: Provides the naive Bayes classifier and evaluation metrics.

spaCy: Enhances preprocessing steps with its robust linguistic processing capabilities.

Optimization

Optimization strategies improve model accuracy and efficiency:

Hyperparameter tuning: Parameters like smoothing factors in naive Bayes are fine-tuned for optimal performance.

Feature selection: Selection techniques (e.g., chi-square test) are used to identify the most discriminative features, enhancing classification accuracy.

Real-time classification and integration

With an optimized model, the next step is to deploy the model for real-time classification.

API integration: The model is linked with a live news API or RSS feeds to classify incoming articles dynamically.

Real-time processing: As new articles are ingested, the model categorizes them immediately, providing real-time insights.

Figures *1*, *2*, *3*, *4*, and 5 (source: Word document and unified modeling language) depict the system architecture, dataflow diagram, class diagram, use case diagram, and sequence diagram of live news classifier.

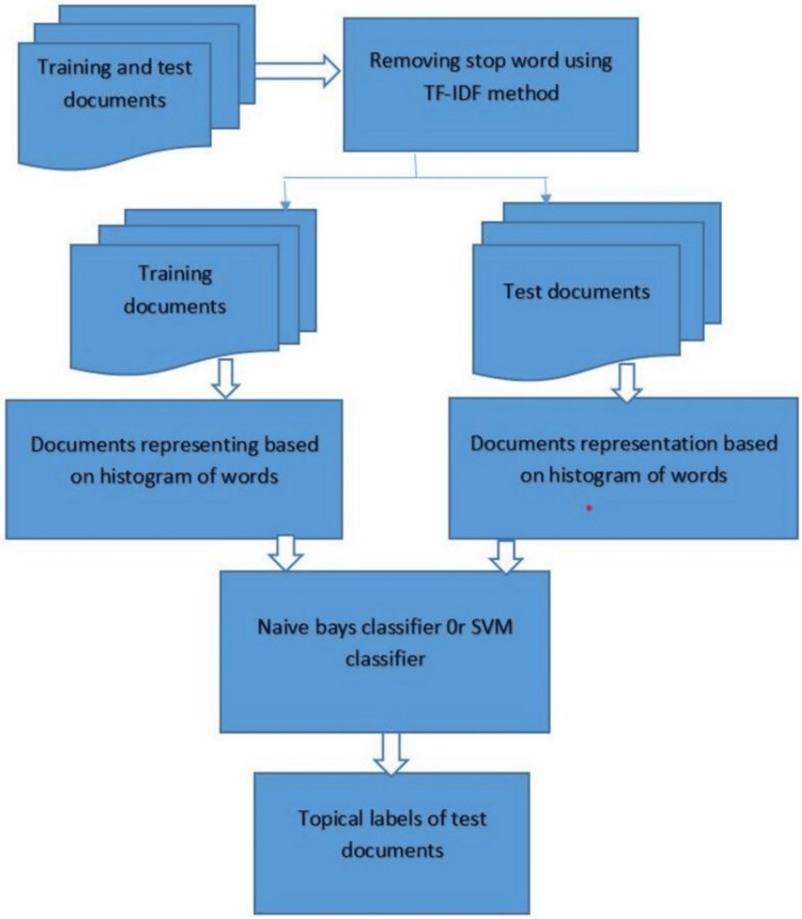

FIGURE 1: Dataflow Diagram

Figure *1* depicts a classification pipeline where training and test documents undergo stop-word removal and TF-IDF weighting to enhance key term relevance. Each document is represented as a histogram of words, with TF-IDF scores prioritizing contextually important terms. These TF-IDF-weighted histograms serve as inputs for a naive Bayes or SVM classifier, which assigns topical labels based on patterns learned during training. This structured approach improves the classifier's accuracy in categorizing documents by emphasizing distinctive vocabulary.

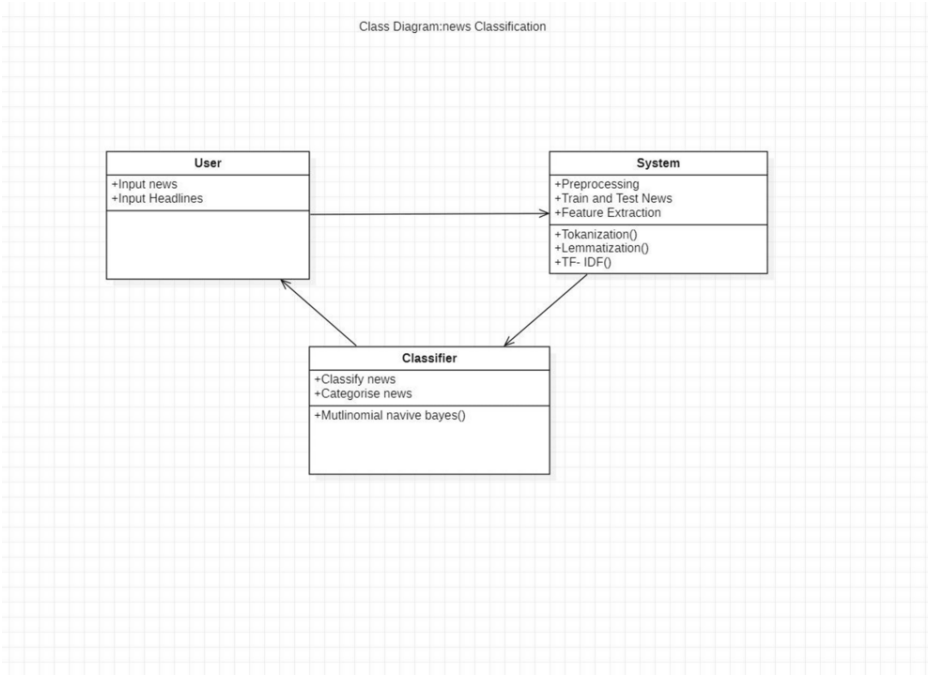

FIGURE 2: Class Diagram

Figure *2* illustrates a structured approach to news classification, showing interactions among the User, System, and Classifier for automated news categorization. The User inputs news data, including headlines, which the System processes through Preprocessing tasks like tokenization, lemmatization, and applying the TF-IDF method to generate meaningful numerical features. The Classifier, using algorithms like multinomial naive Bayes, categorizes the news based on trained patterns, effectively assigning topical labels. This diagram provides a clear pathway for efficiently handling and classifying news data.

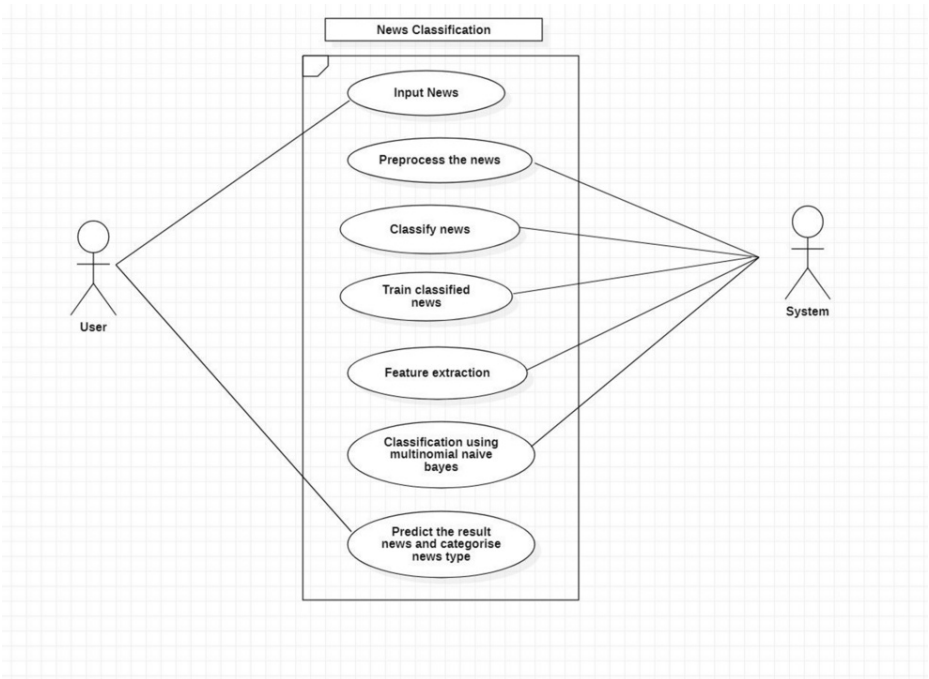

FIGURE 3: Use Case Diagram

Figure *3* outlines a workflow for news classification that starts with the Input News phase, where users submit raw articles. The Preprocess News step cleans and prepares the text for analysis. Next, during Feature Extraction, significant terms and patterns are identified, structuring the text for machine learning. The Train

Classified News function then develops a predictive model using algorithms like multinomial naive Bayes. Finally, the Predict the Result phase classifies the input news into specific categories based on the model's outputs, delivering categorized news types to users. This diagram effectively highlights the critical processes involved in automating reliable and accurate news classification.

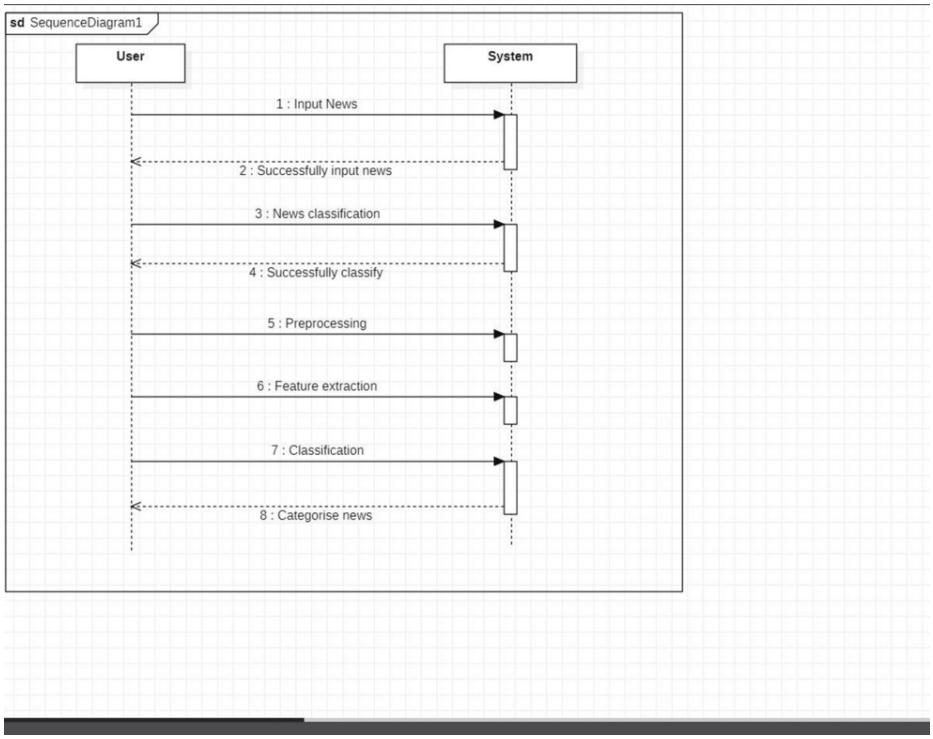

FIGURE 4: Sequence Diagram

Figure *4* explains the interaction between the User and the System in the news classification process. It begins with the User inputting news articles, followed by the System confirming successful input. The System then initiates the News Classification process, which includes Preprocessing the data and performing Feature Extraction. After extracting features, the System moves to the Classification stage and ultimately categorizes the news articles. This diagram highlights the systematic workflow from user input to final categorization, showcasing the structured processes involved in automating news classification.

Results

This study examines the effectiveness of the naive Bayes classifier in classifying live news articles across several categories, including politics, sports, technology, health, and entertainment. The dataset consists of approximately 10,000 news articles collected from various online sources. A comprehensive preprocessing stage was implemented to remove irrelevant information, tokenize the content, and transform it into a bag-of-words format for efficient analysis.

The performance of the classifier was assessed using essential metrics such as precision, recall, and F1-score. These metrics were calculated based on a 10-fold cross-validation procedure, offering an in-depth evaluation of the classifier's accuracy and reliability in categorizing news articles. A summary of the evaluation results is presented in Table *1*.

Category	Precision	Recall	F1-Score
Politics	0.75	0.7	0.72
Sports	0.82	0.8	0.81
Technology	0.78	0.76	0.77
Health	0.65	0.6	0.62
Entertainment	0.7	0.68	0.69
Average	0.74	0.71	0.72

TABLE 1: Performance Metrics

In Table *1* (source: Excel Sheet), the naive Bayes classifier achieved an average precision of 0.74, recall of 0.71, and an F1-score of 0.72 across all categories, indicating solid performance for the live news classification task.

1. Category-specific performance: The category-specific performances are as follows:

- Sports emerged as the best-performing category with a precision of 0.82 and an F1-score of 0.81, suggesting that sports articles often have distinct vocabulary and clear context.

- Politics and technology also performed well, with F1-scores of 0.72 and 0.77, respectively.

- Health showed the lowest performance with an F1-score of 0.62, indicating challenges in distinguishing health-related articles due to overlapping terminology with other categories.

2. Impact of text preprocessing: The implementation of TF-IDF weighting significantly enhanced the model's ability to identify key features, resulting in improved overall accuracy.

3. Computational efficiency: The naive Bayes classifier displayed high computational efficiency, characterized by swift training and quick classification times, making it particularly well suited for applications that require real-time categorization of news content. The average classification time per article was approximately 0.02 seconds, which is highly efficient compared to more complex models.

The results highlight the inherent limitation of the naive Bayes classifier, primarily its assumption of feature independence. This can lead to misclassification in categories where the context is crucial for understanding, particularly in articles that may share similar keywords across different contexts. It underscore the viability of the naive Bayes classifier for live news classification, providing a balance between efficiency and performance. While the model excels in well-defined categories, there is potential for further improvement, particularly in handling overlapping or nuanced categories. Future work should consider integrating more advanced machine learning techniques and exploring additional feature representation methods to enhance classification accuracy across all news categories.

The calculation of precision, recall, and F1-score is as follows:

1. Precision: Precision is the ratio of correctly predicted positive observations to the total predicted positives. In a multi-agent system, precision can be computed individually for each agent or aggregated across agents.

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

In multi-agent systems, we can compute precision per agent or overall,

$$\text{Overall Precision} = \frac{\sum_i \text{TP}_i}{\sum_i (\text{TP}_i + \text{FP}_i)} \qquad (1)$$

In Equation (1),

TP_i = True positives by agent i

FP_i = False positives by agent *i*

2. Recall: Recall is the ratio of correctly predicted positive observations to all observations in the actual class.

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

For multi-agent systems, the overall recall is calculated similarly,

$$\text{Overall Recall} = \frac{\sum_i \text{TP}_i}{\sum_i (\text{TP}_i + \text{FN}_i)} \qquad (2)$$

In Equation (2),

FN_i = False negatives by agent *i*

3. The F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall and is particularly useful when the dataset is imbalanced.

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

For multi-agent systems, it can be calculated per agent or across agents,

$$\text{Overall F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \qquad (3)$$

In Equation (3), *i* represents each individual agent or classifier in the system.

Precision_i and Recall_i are the precision and recall values for the i^{th} agent.

The calculations are presented in Table *2* and Figure *5* (graph).

Precision	Recall	F1-Score
0.4	0.5	0.44
0.23	0.43	0.3
0.38	1	0.55
0	0	0
0	0	0
0.44	0.4	0.42
0	0	0
1	0.4	0.57
0.5	0.5	0.5
0.25	0.5	0.33
0	0	0
0.6	0.25	0.35

TABLE 2: Naive Bayes Classifier Performance Metrics

Table *2* (source: Excel Sheet) presents the precision, recall, and F1-score metrics for different classes or

agents in a classification task related to a multi-agent system.

1. Precision: This measures the proportion of true positive predictions out of all positive predictions made by the model or agent. A higher precision indicates fewer false positives. In the first row, the precision is 0.4, meaning that 40% of the predicted positives are correct.

2. Recall: This measures the proportion of actual positives that the model or agent correctly identified. It shows how well the model recalls relevant instances from the actual positives. In the second row, a recall of 0.43 indicates that the model accurately predicted 43% of the actual positive cases.

3. F1-score: The F1-score is a metric that combines precision and recall into a single value, representing their harmonic mean. This score provides a balanced assessment of both metrics, making it an effective indicator of a model's overall performance in classification tasks, especially in scenarios with imbalanced classes. For instance, the F1-score of 0.44 in the first row reflects a balance between a precision of 0.4 and a recall of 0.5.

Row 1: The precision of 0.4 and recall of 0.5 demonstrate that while the model successfully identifies 50% of the actual positive instances, it also indicates that 60% of its predictions were incorrect.

Row 2: This row shows lower precision (0.23), meaning more false positives, but slightly better recall (0.43). The F1-score (0.3) reflects this imbalance between the two metrics.

Row 3: Here, the recall is perfect (1.0), meaning the agent identified all true positive cases. However, the precision (0.38) is quite low, indicating that the system made many false positives. The F1-score (0.55) reflects the trade-off between these two.

Zero values: Several rows show zero values for precision, recall, and F1-score. This indicates that for those categories or agents, the classifier either did not make any positive predictions (leading to zero precision) or failed to recall any positive instances, resulting in no true positives. This can suggest misclassification or absence of relevant data for those classes.

High precision/low recall: Certain instances, such as the final row with a precision of 0.6 and recall of 0.25, indicate a higher accuracy in predicting positives but a significant number of missed actual positives. This imbalance results in a reduced F1-score of 0.35, highlighting the classifier's tendency to favor precision over recall in these cases.

Hence, it represents the performance of individual agents or models in a multi-class or multi-agent system. Each row could correspond to a different class in a multi-class classification or different agents contributing to predictions. The variability in precision, recall, and F1-score indicates that some models or agents perform well (e.g., row 3 with perfect recall), while others underperform (e.g., rows with zeros). The results suggest potential issues with class imbalance or varying data complexity. Some classes/agents perform well in recall but suffer in precision, or vice versa, leading to lower F1-scores. Rows with low or zero precision/recall indicate areas where the model fails to make meaningful predictions. In summary, this table highlights the varying performance of a classification system across different agents or classes, with room for improvement in balancing precision and recall across all categories to achieve higher F1-scores and overall better classification.

FIGURE 5: Agent Performance Evaluation

Figure *5* (source: Excel Sheet) represents the precision, recall, and F1-score trends for different categories or agents across 12 data points. The three lines illustrate how these metrics vary across each data point.

The experimental results indicate notable variability in the naive Bayes classifier's performance across different categories or agents, suggesting inherent limitations in applying this model within a multi-agent news classification system. While precision, recall, and F1-scores reached favorable levels for certain categories with distinct vocabulary profiles, performance significantly dropped in categories with ambiguous or overlapping vocabulary, underscoring naive Bayes' assumption of feature independence as a limiting factor.

In categories with high recall but low precision (e.g., point 3), the classifier demonstrated its ability to identify true positive instances but at the expense of increased false positives, likely due to overlaps in the lexicon shared across classes. Alternatively, the model's performance in categories with high precision but low recall (e.g., point 8) indicates the presence of strict but narrow prediction criteria, leading to missed relevant instances. This highlights that while the naive Bayes classifier can accurately identify some categories, it struggles with broader generalizations where vocabulary overlaps or context-dependency is prominent.

The occurrence of zero values across metrics (observed at points 4, 5, and 11) further reveals critical points of failure within the model. Such zero values may suggest complete misclassification or the absence of positive predictions, potentially resulting from data sparsity, class imbalance, or a lack of distinguishable terms within specific categories. These findings align with known theoretical limitations of naive Bayes, particularly in failing to account for feature dependencies, which may hinder its effectiveness in complex, real-time classification tasks.

To mitigate these challenges, future research should consider hybrid models that combine the simplicity and efficiency of naive Bayes with models capable of capturing more complex feature interdependencies, such as SVMs or transformer-based architectures. Additionally, implementing advanced feature engineering techniques, such as word embeddings, term weighting, or context-based embeddings, could enhance the model's ability to discern subtle linguistic cues and improve performance across categories with ambiguous terminology. Integrating such methods could help address both precision-recall imbalances and issues with data sparsity.

The current study demonstrates that while naive Bayes can perform effectively in multi-agent news classification tasks with distinct categories, there are notable limitations in handling overlapping or complex vocabulary structures. These findings highlight the need for hybrid or enhanced feature engineering approaches to optimize performance consistency and accuracy. Future advancements in combining naive Bayes with more context-aware models could provide significant improvements, particularly in the real-time classification of high-dimensional, dynamically evolving text data.

Discussion

The experimental analysis provides insights into the performance of a naive Bayes classifier applied in a multi-agent news classification context. Key metrics, including precision, recall, and F1-score, were assessed

across several agents or categories, with variations indicating different levels of classification accuracy. Each metric captures specific aspects of the classifier's performance, informing the strengths and weaknesses in predicting certain categories.

For precision, values indicate the ratio of true positives among predicted positives, shedding light on the classifier's ability to minimize false positives across agents. Recall reflects the model's capacity to retrieve actual positives within each category, indicating its efficiency in identifying relevant instances. The F1-score, a harmonic mean of precision and recall, provides a balanced perspective on the classifier's ability to perform in scenarios with imbalanced data distributions or varying class complexities.

Evaluating these metrics in the experimental setup reveals how certain categories or agents may experience challenges due to factors like vocabulary overlap, class imbalance, and context dependencies. These factors can influence the classifier's ability to generalize across different categories, potentially impacting its precision and recall in specific instances.

The occurrence of zero metrics in certain rows suggests issues such as data sparsity, lack of distinguishable features for certain categories, or complete misclassification within those cases. These results point to the need for enhanced methods that address the limitations of naive Bayes when dealing with overlapping feature spaces and complex classification environments.

This analysis highlights the experimental results of using naive Bayes within a multi-agent system and demonstrates its potential applicability in real-time classification tasks.

Conclusions

This study demonstrates the effectiveness of the naive Bayes classifier for live news categorization, specifically within high-dimensional, real-time text environments. Despite its inherent simplicity, naive Bayes showed strong performance across distinct news categories, as evidenced by precision, recall, and F1-scores. Categories with clearly defined, non-overlapping vocabularies achieved higher classification accuracy, while those with ambiguous or overlapping vocabularies presented challenges, highlighting naive Bayes' limitations due to its assumption of feature independence.

Future research should consider hybrid models that integrate naive Bayes with more sophisticated algorithms, such SVMs or transformer-based deep learning architectures. These approaches can leverage complementary strengths to enhance classification robustness, particularly in categories with nuanced, overlapping vocabularies. Advanced feature engineering techniques, including word embeddings and context-aware NLP models, could further strengthen the model's ability to capture semantic intricacies and interdependencies within live data streams. Such enhancements hold promise for improving classification precision in complex categories, supporting naive Bayes' applicability in real-time, high-dimensional text classification across diverse domains.

Additional Information

Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

Concept and design: Dhruv S. Kashid, Janhavi D. Patil, Amar Buchade

Acquisition, analysis, or interpretation of data: Dhruv S. Kashid, Janhavi D. Patil, Amar Buchade

Drafting of the manuscript: Dhruv S. Kashid, Janhavi D. Patil, Amar Buchade

Critical review of the manuscript for important intellectual content: Dhruv S. Kashid, Janhavi D. Patil, Amar Buchade

Supervision: Amar Buchade

Disclosures

Human subjects: All authors have confirmed that this study did not involve human participants or tissue. **Animal subjects:** All authors have confirmed that this study did not involve animal subjects or tissue. **Conflicts of interest:** In compliance with the ICMJE uniform disclosure form, all authors declare the following: **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work. **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work. **Other relationships:** All authors have declared that there are no

other relationships or activities that could appear to have influenced the submitted work.

Acknowledgements

Dhruv Kashid and Janhavi Patil contributed equally to the work and should be considered co-first authors.

References

1. Dadgar SMH, Araghi MS, Farahani MM: A novel text mining approach based on TF-IDF and Support Vector Machine for news classification. 2016 IEEE International Conference on Engineering and Technology (ICETECH), Coimbatore, India. 2016, 112-116. 10.1109/ICETECH.2016.7569223
2. Irfani FF, Fauzi MA, Sari YA: News classification on Twitter using naive Bayes and hypernym-hyponym based feature expansion. 2018 International Conference on Sustainable Information Engineering and Technology (SIET), Malang, Indonesia. 2018, 317-321. 10.1109/SIET.2018.8693213
3. Shahi TB, Pant AK: Nepali news classification using Naïve Bayes, Support Vector Machines and Neural Networks. 2018 International Conference on Communication information and Computing Technology (ICCICT), Mumbai, India. 2018, 1-5. 10.1109/ICCICT.2018.8325883
4. Yasaswi K, Kambala VK, Pavan PS, Sreya M, Jasmika V: News classification using natural language processing. 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom. 2022, 63-67. 10.1109/ICIEM54221.2022.9853174
5. Minaee S, Kalchbrenner N, Cambria E, Nikzad N, Chenaghlu M, Gao J: Deep learning--based text classification: A comprehensive review. ACM Computing Surveys. 2021, 54:1-40. 10.1145/3439726
6. Fu Y, Ke W, Mostafa J: Automated text classification using a multi-agent framework . JCDL '05: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries. Association for Computing Machinery, New York, NY, USA; 2005. 157-158. 10.1145/1065385.1065420
7. Saigal P, Khanna V: Multi-category news classification using Support Vector Machine based classifiers . SN Applied Sciences. 2020, 2:458. 10.1007/s42452-020-2266-6