International Journal Multidisciplinary Science
IJML Vol 5 No. 3 October 2026 | ISSN: 2963-8119 (print), ISSN: 2963-7821 (online), Page 01-17

Development of a Web-Based E-Commerce Product Review Sentiment Analysis System: A Comparative Study of Naive Bayes and Support Vector Machine

Robby Tri Ardianto, Dia Ragasari*
Gunadarma University, Jakarta, Indonesia

Article History
Received : 15-08-2026
Revised : 20-09-2026
Accepted : 15-10-2026
Published : 17-10-2026

Corresponding author*:
dia_ragasari@staff.gunadarma.ac.id

DOI:
https://doi.org/10.56127/ijml.v5i3.3132

Abstract: The increasing volume of customer reviews on e-commerce platforms presents challenges for micro, small, and medium enterprises (MSMEs) in identifying customer perceptions and translating feedback into actionable product improvements. This study aims to develop a web-based sentiment analysis system that compares the performance of Naive Bayes and Support Vector Machine (SVM) algorithms while providing automated product improvement recommendations. The system was developed using the Waterfall methodology, incorporating requirements analysis, system design, implementation, and functional testing. A total of 5,397 Indonesian-language product reviews from the PRDECT-ID dataset were processed using text preprocessing and Term Frequency–Inverse Document Frequency (TF-IDF) feature extraction. Sentiment classification was supported by a rule-based lexicon approach for handling neutral sentiment. Model performance was evaluated using accuracy, precision, recall, and F1-score, while system functionality was assessed through black-box testing. The experimental results demonstrate that SVM outperformed Naive Bayes, achieving an accuracy of 94.44%, precision of 94.46%, recall of 94.44%, and F1-score of 94.44%, compared with 93.98%, 94.13%, 93.98%, and 93.97%, respectively, for Naive Bayes. The resulting system, SentimenUMKM, was implemented using the Flask framework and provides sentiment visualization, individual and batch review analysis, and rule-based product improvement recommendations. These findings indicate that SVM offers superior classification performance for the evaluated dataset and demonstrate the practical potential of integrating sentiment classification with automated recommendations to support data-driven product improvement decisions among MSMEs.

Keywords: E-Commerce; Sentiment Analysis; Support Vector Machine; Naive Bayes; TF-IDF; Product Improvement Recommendation.

INTRODUCTION

The continuous growth of electronic commerce (e-commerce) has transformed how businesses interact with consumers, distribute products, and evaluate market demand. Digital marketplaces provide opportunities for micro, small, and medium enterprises (MSMEs) to expand their customer reach while generating substantial amounts of consumer feedback. Among the various forms of digital feedback, online product reviews are particularly important because they contain customers' evaluations of product quality, pricing, delivery services, and overall purchasing experiences. These reviews can influence purchasing decisions and provide businesses with valuable information for improving products and services. However, their increasing volume and unstructured textual

© 2026 by the author. This is an open-access article under the Creative Commons Attribution ShareAlike license (CC BY SA)

characteristics make manual interpretation difficult, particularly for businesses with limited analytical resources. Jain et al. (2021) identified the growing relevance of machine learning in processing online consumer reviews, while Gooljar et al. (2024) highlighted the role of sentiment-based predictive models in understanding online purchasing behavior. In the Indonesian context, Romadhony et al. (2024) demonstrated the potential of computational methods for analyzing product reviews written in Indonesian.

The increasing availability of customer-generated data has also created opportunities for MSMEs to adopt more systematic and evidence-based business practices. Customer reviews provide direct descriptions of experiences that may not be fully represented by numerical ratings alone. For example, a review containing a low rating may refer to poor packaging, delayed delivery, inappropriate product dimensions, or unsatisfactory customer service. Without further textual analysis, these distinct problems may be interpreted as equivalent expressions of dissatisfaction, potentially limiting the usefulness of customer feedback for business improvement. Automated text analytics can help distinguish patterns within consumer opinions and provide a more structured interpretation of recurring complaints. Mao et al. (2024) emphasized the importance of sentiment analysis for extracting opinions from large collections of online comments, whereas Kamal and Astri (2025) examined the application of deep learning to sentiment analysis in e-commerce environments. These developments indicate the importance of analytical systems that not only process customer opinions efficiently but also provide interpretable information for business decision-making.

Sentiment analysis is an important area of natural language processing (NLP) that aims to identify and classify subjective information, including opinions, attitudes, and emotional orientations expressed in textual data. Its applications include customer satisfaction analysis, social media monitoring, service evaluation, and product review classification. In e-commerce applications, sentiment analysis commonly identifies positive and negative evaluations, although neutral sentiment may also be necessary to accommodate statements that do not express a clear evaluative orientation. The development of sentiment analysis methods has progressed from lexicon-based approaches and conventional machine learning toward neural networks and transformer-based architectures. Mao et al. (2024) reviewed the methods, applications, and challenges of sentiment analysis, emphasizing the diversity of computational approaches and the continuing difficulties associated with language interpretation. Similarly, Chauhan et al. (2023) reviewed deep learning approaches to aspect-based sentiment analysis and demonstrated the importance of identifying sentiment at a more detailed level than overall document polarity. These findings suggest that sentiment analysis systems should be designed according to their intended analytical objectives, data characteristics, and practical deployment requirements.

Conventional supervised machine learning algorithms remain relevant for sentiment classification because they can be implemented using relatively straightforward feature representations and evaluated through established classification metrics. Two frequently investigated algorithms are Naive Bayes and Support Vector Machine (SVM). Naive Bayes is a probabilistic classifier based on Bayes' theorem and an assumption of conditional independence among features. Its relatively simple computational structure makes it applicable to text classification tasks involving high-dimensional feature representations. SVM, in contrast, seeks a separating hyperplane that maximizes the margin between different classes and is particularly relevant to sparse textual feature spaces. Romadhony et al. (2024) evaluated multiple classification approaches, including Naive Bayes and SVM, using Indonesian product review data. Yolanda and Tri Mulya (2024) investigated SVM for sentiment analysis of Sayurbox application reviews and reported an accuracy of

89.29%. Meanwhile, Tarigan et al. (2024) examined Naive Bayes for Indonesian-language text classification in a different application domain. Collectively, these studies support the relevance of conventional machine learning algorithms while indicating that classification performance should be evaluated in relation to the characteristics of the dataset and experimental configuration.

An important stage in conventional machine learning-based sentiment analysis is the transformation of textual information into numerical features. Product reviews typically contain informal language, abbreviations, inconsistent spelling, and expressions whose meanings depend on context. Text preprocessing is therefore necessary to prepare the data for subsequent analysis. Common procedures include case normalization, tokenization, removal of unnecessary textual elements, and other language-specific transformations. Following preprocessing, Term Frequency–Inverse Document Frequency (TF-IDF) can be employed to represent textual documents according to the relative importance of their constituent terms. TF-IDF is particularly useful for conventional text classifiers because it generates weighted numerical features that can be processed by algorithms such as Naive Bayes and SVM. Nevertheless, the effectiveness of these representations depends on vocabulary characteristics, document distributions, and preprocessing choices. The systematic review by Jain et al. (2021) demonstrated the variety of machine learning approaches employed in online review analysis, while Romadhony et al. (2024) provided evidence of the applicability of multiple classification methods to Indonesian product review datasets.

The availability and characteristics of labeled datasets are equally important for developing reliable sentiment classification models. In low-resource or domain-specific language settings, publicly available datasets can facilitate reproducible experimentation and comparative evaluation. Sutoyo et al. (2022) introduced the PRDECT-ID dataset, consisting of 5,400 Indonesian-language product reviews collected from Tokopedia across 29 product categories. The dataset includes emotion annotations and attributes that support sentiment classification research. Its availability provides a relevant empirical foundation for evaluating machine learning approaches to Indonesian e-commerce reviews. However, an important distinction must be maintained between the dataset's original annotation structure and the classification requirements of a deployed application. Although the dataset includes positive and negative sentiment labels, a practical system may need to identify neutral expressions that are not represented as a separate class in the supervised training labels. This limitation motivates the integration of an additional rule-based lexicon mechanism to support neutral sentiment detection rather than assuming that a binary classifier has been trained directly to recognize three sentiment classes.

Beyond document-level sentiment classification, recent research has emphasized the importance of extracting more specific information from customer opinions. Aspect-based sentiment analysis aims to identify the entities or attributes being discussed and determine the sentiment associated with each aspect. Brauwers and Frasincar (2023) reviewed aspect-based sentiment classification methods and highlighted the relevance of fine-grained sentiment interpretation. Zhang et al. (2023) further examined the tasks, methods, and challenges associated with aspect-based sentiment analysis, demonstrating that identifying specific opinion targets is an important component of advanced sentiment understanding. Zhao et al. (2023) investigated multitask learning for aspect-based sentiment analysis of online reviews, illustrating the relationship between aspect extraction and sentiment polarity classification. These developments indicate that overall sentiment labels may be insufficient when businesses need to understand the specific causes of customer dissatisfaction. Nevertheless, advanced aspect-based approaches can introduce additional

requirements concerning annotated data, computational resources, and model complexity. For practical MSME-oriented applications, rule-based extraction of recurring negative-review keywords offers a simpler mechanism for linking sentiment outcomes with predefined product improvement categories, although it should not be regarded as equivalent to a fully validated aspect-based sentiment analysis model.

The practical implementation of sentiment analysis also requires attention to accessibility, interpretability, and the presentation of analytical results. A classification model may achieve satisfactory predictive performance but provide limited operational value if its outputs remain inaccessible to intended users. Web-based systems can reduce this limitation by providing interfaces for review submission, batch processing, sentiment visualization, and the presentation of analytical findings. For MSMEs, an accessible application may be more practical than a standalone machine learning experiment requiring direct interaction with programming environments. Furthermore, the usefulness of sentiment analysis depends not only on identifying favorable or unfavorable opinions but also on communicating patterns that can support product development and service improvement. Gooljar et al. (2024) discussed the relevance of sentiment-based models for understanding online purchasing behavior, while Brauwers and Frasincar (2023) emphasized the importance of more detailed sentiment interpretation. These perspectives support the development of systems that connect automated classification with understandable analytical outputs and business-oriented recommendations.

Despite substantial progress in sentiment analysis research, an application-oriented gap remains between the evaluation of classification algorithms and the development of integrated tools for interpreting customer feedback. Existing research has provided valuable evidence regarding machine learning classification, Indonesian product review datasets, and fine-grained sentiment interpretation (Sutoyo et al., 2022; Romadhony et al., 2024; Zhang et al., 2023). However, these research directions do not necessarily address the complete workflow required by MSMEs, from submitting customer reviews to comparing classification results and identifying actionable improvement priorities. Moreover, algorithmic comparisons reported across different studies cannot establish the superiority of one classifier over another because performance is influenced by dataset composition, preprocessing procedures, feature representations, and evaluation protocols. An application-level comparison using a common dataset and consistent evaluation metrics is therefore necessary. A further requirement is to address neutral sentiment in a system trained on binary sentiment labels and to transform recurring negative-review expressions into structured recommendations. The research gap addressed in this study is thus the integration of these complementary capabilities within a single accessible web-based platform, rather than the introduction of a fundamentally new classification algorithm.

Accordingly, this study aims to develop SentimenUMKM, a web-based e-commerce product review sentiment analysis system that integrates Naive Bayes and SVM classification with rule-based neutral sentiment detection and automated product improvement recommendations. The system employs the PRDECT-ID dataset, with 5,397 usable Indonesian-language product reviews retained after preprocessing, and applies TF-IDF for numerical feature extraction. The classification models are evaluated using accuracy, precision, recall, and F1-score to establish their relative performance under the study's experimental conditions. The application is developed using the Waterfall software development methodology and the Python Flask framework, with interfaces supporting individual review analysis, CSV-based batch processing, sentiment visualization, model comparison, and recommendation generation. The principal contribution of this study is the integration of comparative machine learning evaluation, supplementary neutral

sentiment identification, and rule-based product improvement recommendations into a web-accessible application. By connecting sentiment classification results with interpretable feedback categories, the proposed system is intended to support MSMEs in identifying recurring customer concerns and making more informed product improvement decisions.

RESEARCH METHOD

Research Design

This study employed a quantitative experimental approach integrated with web-based information system development to investigate the classification performance of Naive Bayes and Support Vector Machine (SVM) algorithms for Indonesian-language e-commerce product reviews. The research focused on the development of SentimenUMKM, a sentiment analysis application designed to assist micro, small, and medium enterprises (MSMEs) in interpreting customer feedback and identifying potential product improvements. The experimental component involved text preprocessing, feature extraction, supervised model training, and comparative performance evaluation. The system development component involved requirements analysis, system design, implementation, functional testing, deployment, and maintenance.

The application was developed using the Waterfall software development methodology, which organizes development activities into sequential stages. During requirements analysis, the functional and non-functional requirements were identified, including individual text analysis, CSV-based batch processing, sentiment classification, model comparison, sentiment visualization, and automated product improvement recommendations. The design stage involved system architecture development, navigation structure design, flowchart modeling, Unified Modeling Language (UML) diagrams, and user interface design. Subsequently, the application was implemented using Python and the Flask framework, followed by functional testing and online deployment.

The proposed system integrates supervised machine learning with a rule-based lexicon mechanism. Naive Bayes and SVM are used to classify product reviews into positive and negative sentiment categories, whereas the rule-based lexicon mechanism provides supplementary identification of neutral expressions. In addition, a rule-based recommendation module analyzes keywords associated with negative customer reviews and maps them to predefined product improvement categories.

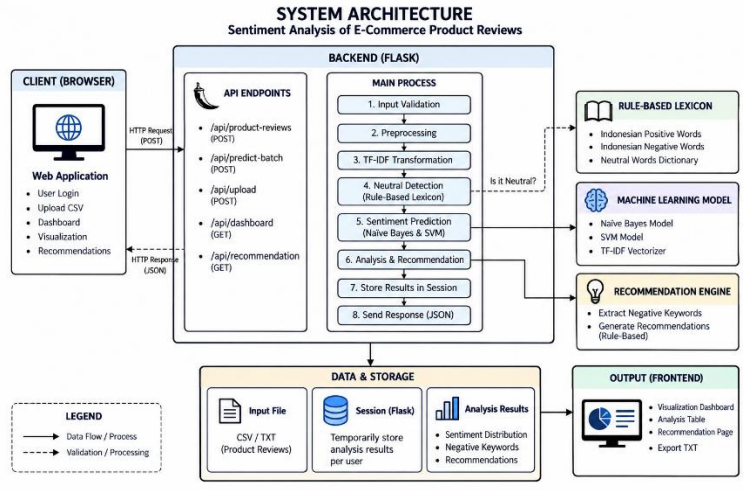


Figure 1. Architecture of the proposed web-based sentiment analysis and product improvement recommendation system.

Dataset and Data Preparation

The dataset employed in this research was the PRDECT-ID (Indonesian Product Reviews Dataset for Emotions Classification Tasks), which contains Indonesian-language product reviews collected from the Tokopedia e-commerce platform. The original dataset comprises 5,400 reviews covering 29 product categories. The dataset was selected because it represents consumer-generated textual data in the Indonesian e-commerce environment and provides labeled information suitable for sentiment classification experiments.

The initial dataset consisted of 5,400 review records. During preprocessing, three records were excluded because the text became empty after the removal of stopwords and other preprocessing operations. Consequently, 5,397 usable reviews were retained for model training and evaluation. The supervised classification experiment used the positive and negative sentiment labels provided in the dataset. Neutral sentiment was handled separately through the rule-based lexicon mechanism during application-level prediction because neutral labels were not included as a distinct supervised training category.

Table 1. Characteristics and sentiment distribution of the PRDECT-ID dataset.

Sentiment Class	Training Data (80%)	Testing Data (20%)	Total
Positive	2,583	548	3,131
Negative	1,734	467	2,201
Neutral	0	0	65[o]
Total	4,317	1,080	5,397

The dataset was divided into training and testing subsets using a stratified sampling procedure to maintain the proportional representation of sentiment classes. An 80:20 training-to-testing ratio was applied, resulting in 4,317 training records and 1,080 testing records. The same data partition was used for both classification algorithms to support a consistent comparison of predictive performance.

Text Preprocessing

Text preprocessing was conducted to transform unstructured Indonesian-language product reviews into standardized textual representations suitable for machine learning. The preprocessing procedure was implemented in a dedicated Python module and applied consistently to the training data and incoming user reviews.

The preprocessing sequence consisted of case folding, noise removal, tokenization, stopword removal, and stemming. Case folding converted uppercase characters into lowercase characters to reduce variations in word representation. Noise removal eliminated unnecessary textual elements, including URLs, mentions, numerical characters, and punctuation marks. Tokenization separated the cleaned text into individual word tokens, enabling subsequent lexical processing.

Stopword removal was performed using an Indonesian-language stopword list supplemented with selected informal expressions commonly found in customer reviews. This process reduced the influence of frequently occurring words considered less informative for classification. Stemming was subsequently conducted using the PySastrawi library to convert inflected Indonesian words into their base forms. The resulting cleaned text was used as input for TF-IDF feature extraction.

[INSERT TABLE 2 HERE — Adapt from Tables 3.4, 3.5, and 3.6, pages 52–54 of the original document. Combine the positive, negative, and neutral preprocessing examples into one compact table.]

Table 2. Examples of Indonesian product review preprocessing across different sentiment categories.

Preprocessing Stage	Result
Original Text	The product arrived quickly, and the packaging was very secure! Highly recommended seller.
Case Folding	the product arrived quickly, and the packaging was very secure! highly recommended seller.
Noise Removal	the product arrived quickly and the packaging was very secure highly recommended seller
Tokenization	['the', 'product', 'arrived', 'quickly', 'and', 'the', 'packaging', 'was', 'very', 'secure', 'highly', 'recommended', 'seller']
Stopword Removal	['product', 'arrived', 'quickly', 'packaging', 'secure', 'highly', 'recommended', 'seller']
Stemming	['product', 'arriv', 'quick', 'packag', 'secur', 'highli', 'recommend', 'seller']
Final Preprocessed Text	product arriv quick packag secur highli recommend seller

The same preprocessing procedure was applied to both Naive Bayes and SVM to ensure that differences in classification performance were not caused by different text preparation methods. Records producing empty strings after preprocessing were excluded from subsequent model training and evaluation.

Sentiment Classification Using Naive Bayes and Support Vector Machine

Two supervised machine learning algorithms, Multinomial Naive Bayes and Support Vector Machine, were implemented and compared using the same preprocessed dataset and TF-IDF feature representation.

Multinomial Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. The method estimates the probability of a sentiment class given the observed textual features and assigns the review to the class with the highest posterior probability. In this research, the MultinomialNB implementation from Scikit-learn was used with the default smoothing parameter, alpha = 1.0.

Support Vector Machine is a supervised learning algorithm that identifies a decision boundary separating observations from different classes. The present study employed LinearSVC, which implements a linear SVM classifier suitable for high-dimensional text representations. The regularization parameter was set to C = 1.0, corresponding to the default configuration adopted in the source implementation.

The two algorithms were trained using identical training records and evaluated using the same testing subset. Default parameter settings were retained because the research

objective was to compare the classification performance of the two algorithms under their selected baseline configurations rather than conduct hyperparameter optimization. Consequently, no separate optimization procedure was included in the experimental design.

The classification experiment focused on positive and negative sentiment because these were the sentiment categories represented in the supervised training labels. The outputs of the two models were subsequently integrated into the web application, allowing users to compare classification results.

Rule-Based Lexicon for Neutral Sentiment Detection

A supplementary rule-based lexicon mechanism was implemented to identify neutral sentiment in incoming customer reviews. This mechanism was necessary because the supervised classification dataset did not provide a separate neutral sentiment class. Therefore, the neutral category was not learned directly by either the Naive Bayes or SVM model.

The rule-based lexicon mechanism examined the preprocessed tokens of each review and compared them with predefined positive and negative sentiment word lists. When no sentiment-bearing terms were identified, the review was assigned a neutral sentiment label. Conversely, when positive or negative lexical indicators were detected, the review was forwarded to the trained Naive Bayes and SVM classifiers for binary sentiment prediction. This hybrid arrangement enabled the web application to present three sentiment categories while retaining the binary classification framework used for machine learning evaluation. Importantly, the lexicon-based neutral detection mechanism was treated as a supplementary application component rather than as a third class predicted by the supervised models.

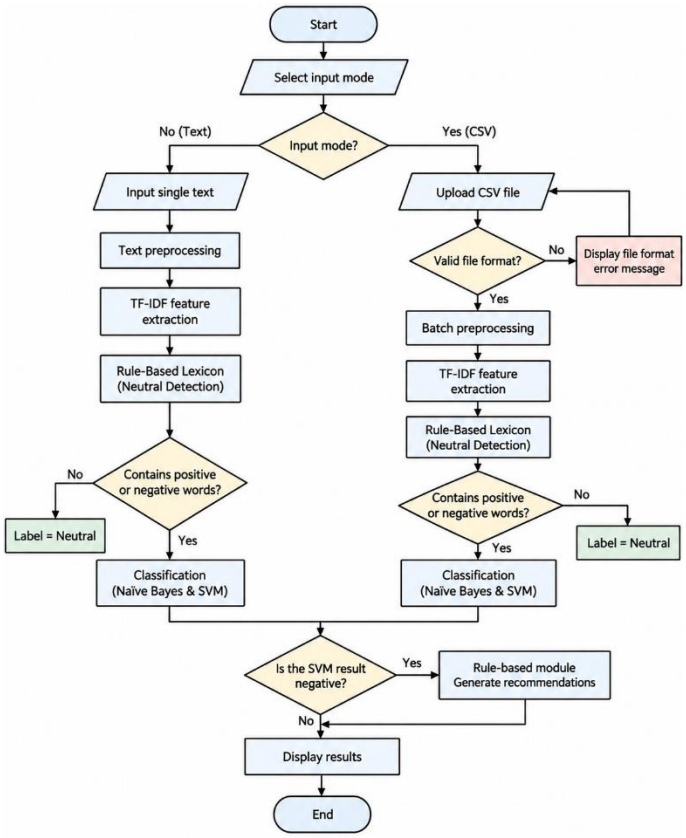


Figure 2. Sentiment analysis workflow integrating rule-based neutral detection and machine learning classification.

RESULT AND DISCUSSION

Dataset Preparation and Distribution

The experimental analysis was conducted using the PRDECT-ID dataset, which initially consisted of 5,400 Indonesian-language e-commerce product reviews. After text preprocessing and validation, three records were excluded because their textual content became empty. Consequently, 5,397 valid reviews were retained for subsequent feature extraction, model training, and evaluation. The exclusion of these records represented approximately 0.06% of the original dataset, indicating that preprocessing preserved nearly all available review observations.

The cleaned dataset was divided into training and testing subsets using an 80:20 holdout strategy with stratified sampling. This procedure produced 4,317 training observations and 1,080 testing observations. Stratification was applied to preserve the relative representation of positive and negative sentiment classes in both subsets. The resulting data partition provided a common experimental basis for comparing Naive Bayes and SVM under equivalent training and testing conditions.

The dataset preparation results also revealed a limitation associated with generic Indonesian stopword removal. Certain words that may carry sentiment information in product review contexts can be removed when treated as common stopwords. For example, expressions relating to delivery duration or product quality may lose important evaluative terms during preprocessing. Although only three records were excluded in the present dataset, this observation indicates that preprocessing rules should be carefully adapted to the linguistic characteristics of e-commerce reviews. The retained dataset was nevertheless sufficient for the comparative classification experiment conducted in this study.

Text Preprocessing and Feature Representation

The preprocessing procedure transformed raw Indonesian-language product reviews into standardized textual representations through case folding, noise removal, tokenization, stopword removal, and stemming. The implementation was examined using examples representing positive, negative, and neutral review expressions. The preprocessing results demonstrated that informal sentences could be reduced to normalized word sequences while retaining important terms associated with customer evaluations.

For positive reviews, expressions describing rapid delivery, secure packaging, and satisfactory seller performance were transformed into simplified textual features. Negative reviews containing terms associated with damaged products, inadequate packaging, and customer disappointment were similarly standardized. Meanwhile, descriptive reviews without explicit positive or negative lexical indicators were processed into representations suitable for supplementary neutral sentiment detection. These examples illustrate how preprocessing reduced textual variation before feature extraction and sentiment classification.

The processed documents were converted into numerical vectors using TF-IDF. The implementation employed unigram and bigram features, a maximum feature limit of 10,000, and a minimum document frequency of two. The resulting feature representation contained 6,496 vocabulary features after training on the selected dataset. These features provided a shared input representation for both machine learning algorithms, ensuring that differences in their predictive performance were evaluated using the same textual information.

The use of a common TF-IDF representation is important because Naive Bayes and SVM employ different classification principles. Naive Bayes estimates class probabilities from feature distributions, whereas SVM determines a linear decision boundary between

sentiment classes. Maintaining consistent input features therefore allows the comparison to focus on differences between the selected classification algorithms rather than differences in feature extraction procedures.

Comparative Performance of Naive Bayes and Support Vector Machine

The classification performance of Naive Bayes and SVM was evaluated using 1,080 testing observations that were not included in model training. Four evaluation metrics were employed: accuracy, precision, recall, and F1-score. The experimental results demonstrated that both algorithms achieved classification accuracy above 93%, indicating their ability to distinguish positive and negative sentiment patterns in the evaluated Indonesian-language product reviews.

Table 6. Comparative classification performance of Naive Bayes and SVM

Evaluation Metrics	Naive Bayes	SVM (LinearSVC)
Accuracy	93.98%	94.44%
Precision	94.13%	94.46%
Recall	93.98%	94.44%
F1-Score	93.97%	94.44%

The results indicate that SVM achieved slightly higher performance than Naive Bayes across all four evaluation metrics. SVM obtained an accuracy of 94.44%, compared with 93.98% for Naive Bayes, corresponding to an improvement of 0.46 percentage points. SVM also achieved a precision of 94.46%, exceeding the Naive Bayes precision of 94.13% by 0.33 percentage points. The difference in recall was 0.46 percentage points, while the F1-score improvement was 0.47 percentage points.

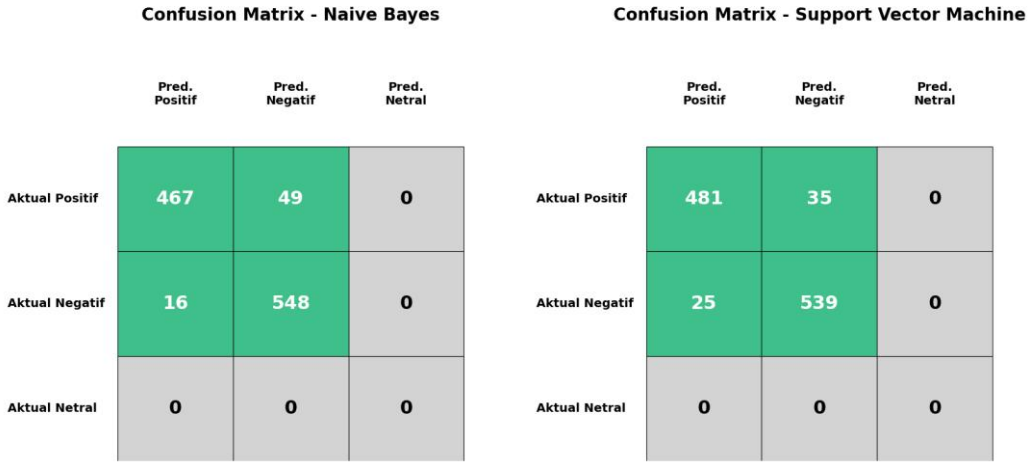
Although SVM consistently outperformed Naive Bayes, the differences were relatively small. This suggests that both algorithms were effective for the binary sentiment classification task under the selected experimental conditions. The stronger SVM performance may be associated with the suitability of linear decision boundaries for high-dimensional, sparse TF-IDF representations. However, the present experiment does not independently establish the cause of the performance difference, and the observed advantage should be interpreted specifically within the evaluated dataset and model configurations.

Both classifiers exceeded the minimum accuracy requirement of 80% specified in the system's non-functional requirements. The results therefore satisfied the predefined classification performance criterion. Nevertheless, the comparison was conducted using default hyperparameter configurations rather than independently optimized models. Consequently, the findings establish the relative performance of the implemented baseline configurations and should not be generalized as evidence that SVM will consistently outperform Naive Bayes across other datasets or parameter settings.

Confusion Matrix Analysis

The confusion matrix was used to examine the distribution of correct and incorrect predictions generated by the two classification models. Unlike overall accuracy, the confusion matrix provides information about the types of classification errors, distinguishing correctly classified positive and negative reviews from observations assigned to the opposite class.

Figure 3. Confusion matrices of the Naive Bayes and SVM classifiers on the testing dataset.



The classification metrics indicate that SVM produced a slightly lower overall error rate than Naive Bayes. Based on the reported accuracies, the respective error proportions were approximately 5.56% and 6.02%. These differences are consistent with the higher precision, recall, and F1-score achieved by SVM.

The confusion matrices should be interpreted with particular attention to false-positive and false-negative classifications because these errors can affect the interpretation of customer feedback. A negative review incorrectly classified as positive may prevent a potentially important complaint from being recognized, while a positive review incorrectly classified as negative may generate unnecessary concern regarding product quality or service performance. In an MSME-oriented decision-support application, both error types are relevant because the classification output may influence subsequent recommendation generation.

Importantly, the confusion matrix evaluation concerns the positive and negative sentiment categories represented in the supervised training dataset. The neutral category introduced by the supplementary lexicon mechanism was not evaluated as an independently learned third class in the reported model comparison. Therefore, the reported accuracy and related metrics should not be interpreted as an overall three-class classification performance measure.

Rule-Based Lexicon Implementation for Neutral Sentiment

The neutral sentiment detection mechanism was implemented to address the absence of a neutral sentiment class in the supervised training labels. The application examined preprocessed review tokens against predefined positive and negative lexical dictionaries. Reviews containing no recognized positive or negative terms were assigned a neutral label, whereas reviews containing recognized sentiment terms were forwarded to the trained machine learning classifiers.

The implementation demonstrated that the system could produce three sentiment labels at the application level despite using binary supervised classifiers. Positive expressions containing terms such as "baik", "aman", and "ramah" were processed by the machine learning models, while negative expressions containing terms such as "kecewa" and "rusak" were similarly forwarded for classification. Reviews without matching lexical indicators were labeled neutral by the rule-based mechanism.

Table 7. Examples of neutral sentiment detection using the rule-based lexicon mechanism.

No.	Original Review (PRDECT-ID)	Lexicon Dictionary Status	System-Generated Sentiment Label
1	Thank God, the product works properly. The packaging is secure. The seller responds quickly and is friendly.	Contains positive words (works, properly, secure, friendly).	Processed by the machine learning model; classified as Positive.
2	Very disappointed. The scroll wheel stopped working after only four months.	Contains negative words (disappointed, damaged).	Processed by the machine learning model; classified as Negative.
3	The mouse is not very responsive, so I have to press it quite firmly.	No matching words found in either the positive or negative lexicon dictionary.	Neutral (directly assigned by the Rule-Based Lexicon).

The implementation also revealed an important limitation of dictionary-based sentiment identification. In one example, a customer complained that a computer mouse was insufficiently responsive and required excessive pressing force. Although the original dataset labeled the review as negative, the rule-based mechanism classified it as neutral because the relevant complaint expression was not represented in the predefined negative lexicon. This result demonstrates that a lexical matching approach may fail to recognize implicit or context-dependent negative sentiment.

Accordingly, the neutral detection mechanism should be interpreted as a supplementary heuristic rather than a fully validated three-class sentiment classifier. Its performance depends on the coverage and suitability of the predefined sentiment dictionaries. Further development could include expansion of domain-specific lexical resources, improved handling of negation, and evaluation using a separately annotated three-class dataset.

Implementation of Product Improvement Recommendations

The product improvement recommendation module was developed to translate negative customer feedback into predefined categories of corrective action. The module extracted frequently occurring terms from negative reviews and matched them with rule-based recommendation categories. Six categories were implemented: product quality, delivery speed, product information accuracy, customer service, pricing, and product sizing.

Table 8. Mapping of negative review keywords to product improvement recommendations.

Detected Keywords	Category	Recommended Improvement Actions
damaged, defective, dented, broken, destroyed	Product Quality	Improve packaging standards and product handling procedures before shipment.
slow, long delivery time, late, delayed	Delivery Speed	Evaluate logistics partners and provide additional express delivery options.

Detected Keywords	Category	Recommended Improvement Actions
not as described, different, counterfeit, misleading	Product Information Accuracy	Update product images and descriptions to accurately reflect the actual product condition and specifications.
response, reply, slow response, unresponsive	Customer Service	Establish a maximum response time of one hour during business hours.
expensive, overpriced, not worth the price	Product Pricing	Evaluate pricing strategies and consider implementing promotional discounts.
too small, too large, improper fit, incorrect size	Size Accuracy	Provide more detailed and comprehensive sizing guides on product pages.

The implemented rules associated terms indicating damage, defects, dents, or broken products with product quality improvement. Delivery-related complaints involving delayed or slow shipments were mapped to logistics evaluation and possible improvements in delivery service options. Expressions indicating inaccurate product descriptions or differences between advertised and received products were associated with improvements in product information accuracy. Other rules addressed customer service responsiveness, product pricing, and size information.

These results demonstrate the operational integration of sentiment classification with predefined recommendation generation. Instead of presenting negative sentiment labels alone, the system provides users with suggested areas of improvement corresponding to detected complaint keywords. This functionality extends the practical usefulness of the application by connecting analytical outputs with possible business responses.

However, the recommendations are generated through predefined keyword mappings rather than through an independently trained recommendation model. Their relevance is therefore dependent on the completeness of the keyword lists and the appropriateness of the associated recommendations. The original study demonstrates functional implementation of the module but does not report a separate quantitative evaluation of recommendation relevance, user acceptance, or actual business improvement outcomes. Consequently, its effectiveness should be described as demonstrated functionality rather than empirically verified improvement in MSME business performance.

Web-Based Application Implementation

The SentimenUMKM application was successfully implemented using Python Flask as the backend framework, with HTML, CSS, and JavaScript supporting the user interface. The application provides several integrated functions, including individual text analysis, CSV-based batch processing, sentiment visualization, model comparison, and product improvement recommendations.

The individual review analysis interface allows users to submit a single product review and obtain sentiment classification results. The CSV processing function enables the analysis of multiple review records, supporting more efficient processing of larger collections of customer feedback. The dashboard presents sentiment distribution information and recurring keywords, while the recommendation page provides suggestions based on the identified negative-review patterns.



Figure 8. Comparative performance visualization of Naive Bayes and SVM.

The implementation demonstrates that the experimental classification models were successfully incorporated into a web-based application accessible through a graphical interface. This integration is relevant to MSMEs because users can interact with the analytical functions without directly executing Python scripts or managing machine learning libraries.

The developed application was deployed using the PythonAnywhere hosting platform. Successful deployment confirmed that the system could be accessed through an online environment and that its primary analytical functions were operational outside the local development environment.

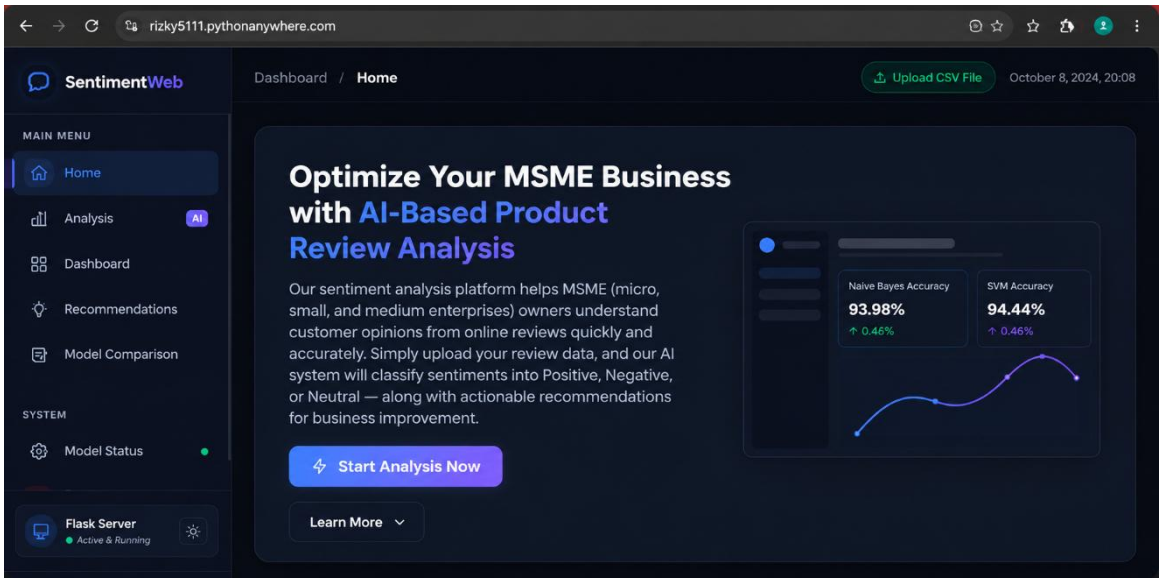


Figure 9. Deployed SentimenUMKM web application.*

Discussion of Research Findings

The findings demonstrate that both Naive Bayes and SVM can provide effective binary sentiment classification for the evaluated Indonesian-language e-commerce product reviews. SVM achieved the highest accuracy, precision, recall, and F1-score, although the improvement over Naive Bayes was relatively small. This result is consistent with the

broader relevance of SVM and Naive Bayes in Indonesian text classification research reported by Romadhony et al. (2024) and Yolanda and Tri Mulya (2024). Nevertheless, direct numerical comparisons with previous studies should be interpreted cautiously because differences in datasets, annotation procedures, preprocessing methods, and experimental settings can influence classification performance.

The slightly stronger performance of SVM suggests that its linear decision boundary was suitable for the TF-IDF representation employed in this experiment. However, the use of default hyperparameters limits the conclusions that can be drawn about the relative optimal performance of the two algorithms. Hyperparameter optimization, repeated experimental runs, or cross-validation could provide additional evidence regarding the stability and generalizability of the observed differences.

The integration of rule-based neutral sentiment detection represents an application-level solution to the absence of neutral labels in the supervised training dataset. This mechanism enabled the application to display positive, negative, and neutral outcomes without retraining the two supervised classifiers on an independently labeled three-class dataset. Nevertheless, the documented misclassification of a negative review as neutral demonstrates the limitations of lexicon coverage and literal keyword matching. The hybrid mechanism therefore improves the range of outputs available to users but introduces a separate source of classification uncertainty that is not captured by the reported binary model evaluation metrics.

The recommendation module constitutes another practical component of the proposed application. By mapping recurring negative-review keywords to predefined improvement categories, SentimenUMKM connects sentiment classification with interpretable business suggestions. This functionality addresses an application-oriented need that extends beyond reporting classification labels. However, the present study does not include an independent evaluation of recommendation precision, expert agreement, user satisfaction, or the effects of recommendations on product performance. The module should therefore be understood as a rule-based decision-support feature whose operational implementation has been demonstrated, rather than a system whose business impact has been established.

Overall, the research contributes an integrated web-based application combining conventional machine learning classification, supplementary neutral sentiment detection, visualization, and product improvement recommendations. The experimental results establish the comparative performance of the implemented Naive Bayes and SVM models, while functional and browser testing demonstrate the operation of the developed application under the specified testing conditions. Future research should focus on evaluating neutral sentiment detection using a properly annotated three-class dataset, optimizing classification parameters, expanding Indonesian domain-specific lexicons, and assessing the practical usefulness of generated recommendations through user-centered evaluation.

CONCLUSION

This study successfully developed SentimenUMKM, a web-based sentiment analysis system integrating Naive Bayes, Support Vector Machine (SVM), TF-IDF feature extraction, rule-based neutral sentiment detection, and automated product improvement recommendations to support micro, small, and medium enterprises (MSMEs) in analyzing Indonesian-language e-commerce product reviews. Using 5,397 preprocessed reviews from the PRDECT-ID dataset, the experimental results demonstrated that SVM achieved superior classification performance, with an accuracy of 94.44%, precision of 94.46%,

recall of 94.44%, and F1-score of 94.44%, compared with Naive Bayes, which achieved 93.98%, 94.13%, 93.98%, and 93.97%, respectively. Although the accuracy difference was relatively small at 0.46 percentage points, SVM consistently outperformed Naive Bayes across all evaluation metrics. The developed application also successfully integrated individual and batch review analysis, sentiment visualization, and rule-based recommendations covering product quality, delivery, pricing, customer service, product information, and sizing. Functional testing across ten black-box scenarios and compatibility testing on three web browsers confirmed that the application operated according to its specified requirements. Nevertheless, the reported classification performance was limited to positive and negative sentiment, while neutral sentiment detection relied on predefined lexical rules that may not adequately capture contextual expressions. Future research should therefore focus on optimizing classification models, developing and evaluating a fully supervised three-class sentiment classification approach, expanding domain-specific sentiment lexicons, and assessing the practical effectiveness of the recommendation system. Overall, the proposed system demonstrates the potential of integrating machine learning and rule-based techniques into an accessible decision-support platform for improving customer feedback analysis in Indonesian e-commerce businesses.

REFERENCES

Abighail, B., Fachrifansyah, Firmanda, M., Anggreainy, M., Harvianto, & Gintoro. (2023). Sentiment analysis e-commerce review. Procedia Computer Science, 227, 1039–1045. https://doi.org/10.1016/j.procs.2023.10.613

Badulescu, Y., Cañas, F., & Cheikhrouhou, N. (2024). Judgmental adjustment of demand forecasting models using social media data and sentiment analysis within Industry 5.0 ecosystems. International Journal of Information Management Data Insights, 4(2), 100272. https://doi.org/10.1016/j.jjimei.2024.100272

Brauwers, G., & Frasincar, F. (2023). A survey on aspect-based sentiment classification. ACM Computing Surveys, 55(4), Article 65. https://doi.org/10.1145/3503044

Chauhan, G. S., Nahta, R., Meena, Y. K., & Gopalani, D. (2023). Aspect based sentiment analysis using deep learning approaches: A survey. Computer Science Review, 49, 100576. https://doi.org/10.1016/j.cosrev.2023.100576

Gooljar, V., Issa, T., Hardin-Ramanan, S., & Abu-Salih, B. (2024). Sentiment-based predictive models for online purchases in the era of marketing 5.0: A systematic review. Journal of Big Data, 11, Article 107. https://doi.org/10.1186/s40537-024-00947-0

Jain, P. K., Pamula, R., & Srivastava, G. (2021). A systematic literature review on machine learning applications for consumer sentiment analysis using online reviews. Computer Science Review. [Volume, article number, dan DOI perlu diverifikasi.]

Kamal, A., & Astri, R. (2025). Eksplorasi sentimen pengguna pada aplikasi e-commerce dengan deep learning. Jurnal Teknologi dan Sistem Informasi Bisnis, 7(3), 435–441. https://doi.org/10.47233/jteksis.v7i3.2010

Mao, Y., Liu, Q., & Zhang, Y. (2024). Sentiment analysis methods, applications, and challenges: A systematic literature review. Journal of King Saud University – Computer and Information Sciences, 36(4), 102048. https://doi.org/10.1016/j.jksuci.2024.102048

Nurmadewi, D., Jailani, Z. F., Rafi, H., Anggoro, D. A., & Setiowati, D. (2026). Perbandingan Support Vector Machine dan Naïve Bayes untuk klasifikasi sentimen

ulasan e-commerce. MALCOM: Indonesian Journal of Machine Learning and Computer Science, 6(2). https://doi.org/10.57152/malcom.v6i2.2648

Romadhony, A., Al Faraby, S., Rismala, R., Wisesty, U. N., & Arifianto, A. (2024). Sentiment analysis on a large Indonesian product review dataset. Journal of Information Systems Engineering and Business Intelligence, 10(1), 167–178. https://doi.org/10.20473/jisebi.10.1.167-178

Sutoyo, R., Achmad, S., Chowanda, A., Andangsari, E. W., & Isa, S. M. (2022). PRDECT-ID: Indonesian product reviews dataset for emotions classification tasks. Data in Brief, 44, 108554. https://doi.org/10.1016/j.dib.2022.108554

Tarigan, A., Sahputra, I., & Multazam, T. (2024). Analisis algoritma Naive Bayes Classifier untuk mendeteksi berita hoax pada Dinas Komunikasi Informatika dan Persandian. SISFO: Jurnal Ilmiah Sistem Informasi, 8(1), 1–13. https://doi.org/10.29103/sisfo.v8i1.18083

Yolanda, A. M., & Tri Mulya, R. (2024). Implementasi metode Support Vector Machine untuk analisis sentimen pada ulasan aplikasi Sayurbox di Google Play Store. VARIANSI: Journal of Statistics and Its Application on Teaching and Research, 6(2), 76–83. https://doi.org/10.35580/variansiunm258

Zhang, W., Li, X., Deng, Y., Bing, L., & Lam, W. (2023). A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. IEEE Transactions on Knowledge and Data Engineering, 35(11), 11019–11038. https://doi.org/10.1109/TKDE.2022.3230975

Zhao, G., Luo, Y., Chen, Q., & Qian, X. (2023). Aspect-based sentiment analysis via multitask learning for online reviews. Knowledge-Based Systems, 264, 110326. https://doi.org/10.1016/j.knosys.2023.110326.