

Evaluating the Trade-Off Between Predictive Performance and Computational Efficiency: The Effect of Stemming on TF-IDF-Based IMDb Sentiment Analysis

Nurfiqih¹, Aa Kurniawan²

^{1,2}Study Program of Informatics Engineering, Universitas Pamulang, Tangerang Selatan, Indonesia

Article Info

Article history:

Received 18/07/2026

Revised 24/08/2026

Accepted 05/09/2026

Keywords:

IMDb

Machine Learning

Sentiment Analysis

Stemming

TF-IDF

ABSTRACT

Sentiment analysis has become an important approach to extracting opinions from textual data. One frequently used preprocessing technique is stemming, which aims to reduce words to their root forms. However, the impact of stemming on classification performance, feature complexity, and computational efficiency remains an open issue. This study evaluated the effect of stemming on IMDb sentiment analysis using four machine learning algorithms: Naïve Bayes, Support Vector Machine (SVM), Decision Tree, and Random Forest. The dataset consisted of 50,000 IMDb movie reviews represented through TF-IDF feature extraction. Performance evaluation was conducted using accuracy, precision, recall, F1-score, feature complexity analysis, and computational runtime measurements. The results demonstrated that stemming successfully reduced TF-IDF features from 99,249 to 68,991, representing a feature reduction of 30.49%. However, stemming slightly decreased classification performance across all evaluated algorithms. SVM achieved the highest accuracy of 89.74% without stemming and 89.32% with stemming. Despite the minor decrease in accuracy, stemming improved computational efficiency for tree-based algorithms, particularly Decision Tree and Random Forest. These findings indicate that stemming does not always improve sentiment classification performance but can significantly reduce feature complexity and computational costs.



This work is licensed under a [CC-BY-NC](https://creativecommons.org/licenses/by-nc/4.0/)

Corresponding Author:

Aa Kurniawan,

Informatics Engineering,

Universitas Pamulang,

Tangerang Selatan, Banten.

Email: dosen02361@unpam.ac.id

1. INTRODUCTION

The rapid growth of user-generated content on digital platforms has led to a substantial increase in textual data containing opinions, experiences, and evaluations of products and services. As a result, *sentiment analysis* has become one of the most important research areas in *Natural Language Processing (NLP)* because it enables automatic identification of public sentiment from textual information [1]. Among various publicly available datasets, the Internet Movie Database (*IMDb*) has become one of the most widely adopted

benchmarks for sentiment classification due to its balanced distribution of positive and negative reviews and its linguistic diversity [2]. The large vocabulary and morphological variations found in IMDb reviews present significant challenges for text classification models, making data preprocessing an essential stage before machine learning algorithms are applied.

Text preprocessing aims to transform unstructured textual data into a structured numerical representation suitable for computational analysis. Common preprocessing procedures include *case folding*, tokenization, stop-word removal, and stemming [3]. Among these techniques, *stemming* has received considerable attention because it reduces inflected or derived words into their root forms, thereby decreasing vocabulary size and simplifying feature representation [4]. Theoretically, reducing redundant lexical variations is expected to improve the effectiveness of feature extraction methods such as *Term Frequency–Inverse Document Frequency (TF-IDF)* while simultaneously lowering computational complexity.

Despite its widespread adoption, the effectiveness of stemming remains inconclusive. Several previous studies have reported improvements in text classification performance after applying stemming because the reduction of morphological variations produces a more compact feature space [5]. Conversely, other studies have demonstrated that stemming may remove important semantic information embedded in different word forms, resulting in reduced classification accuracy [6]. This inconsistency suggests that the effectiveness of stemming is highly dependent on dataset characteristics, language structure, feature representation, and the classification algorithm employed.

Most previous studies primarily evaluate preprocessing techniques based on conventional predictive performance metrics, including accuracy, precision, recall, and F1-score [7], [8]. Although these metrics are essential for assessing classification capability, they do not fully explain the overall impact of preprocessing on machine learning models. In practical applications involving high-dimensional textual data, feature dimensionality and computational efficiency are equally important because they directly influence memory consumption, processing time, and model scalability [9]. Nevertheless, comprehensive investigations that simultaneously analyze classification performance, feature complexity, and computational efficiency remain relatively limited.

Another limitation observed in the existing literature is that most comparative studies evaluate only one or two machine learning algorithms, making it difficult to generalize the influence of preprocessing techniques across different learning paradigms [10]. Probabilistic methods such as Naïve Bayes, margin-based classifiers such as Support Vector Machine (SVM), tree-based methods such as Decision Tree, and ensemble learning approaches such as Random Forest possess fundamentally different learning mechanisms. Consequently, the influence of stemming may vary substantially among these algorithms, particularly when processing high-dimensional TF-IDF feature representations.

To address these research gaps, this study performs a comprehensive evaluation of the impact of stemming on IMDb sentiment analysis using four widely adopted machine learning algorithms, namely Naïve Bayes, Support Vector Machine (SVM), Decision Tree, and Random Forest. Movie reviews are transformed into numerical vectors using the TF-IDF feature extraction technique, while model performance is evaluated using an 80:20 train-test split. A separate 10-fold stratified cross-validation procedure is employed for statistical significance testing using the Wilcoxon Signed-Rank Test. The evaluation framework incorporates three complementary perspectives: classification performance (accuracy, precision, recall, and F1-score), feature complexity measured by TF-IDF dimensionality, and computational efficiency measured by training and testing runtime.

The primary novelty of this research lies in its multidimensional evaluation framework. Unlike most previous studies that emphasize predictive accuracy as the sole performance indicator, this research simultaneously investigates the trade-off between predictive performance, feature complexity, and computational efficiency after the application of stemming. Furthermore, statistical significance testing using the Wilcoxon Signed-Rank Test is employed to determine whether the observed performance differences are statistically meaningful rather than resulting from random variation. This comprehensive evaluation provides a deeper understanding of the practical implications of stemming for sentiment analysis systems operating on large-scale textual datasets.

The contributions of this study can be summarized as follows. First, it provides an empirical evaluation of stemming across four machine learning algorithms representing different computational paradigms. Second, it quantitatively measures the influence of stemming on TF-IDF feature dimensionality and computational runtime in addition to conventional classification metrics. Third, it investigates the trade-off between predictive performance and computational efficiency using statistical significance analysis. The findings of this study are expected to provide practical guidance for researchers and practitioners in selecting appropriate preprocessing strategies for sentiment analysis applications, particularly when balancing classification effectiveness and computational resource requirements.

2. METHOD

2.1 Research Design

This study employed a quantitative experimental research design to investigate the impact of stemming on sentiment classification performance, feature complexity, and computational efficiency in IMDb movie reviews. The experimental approach was selected because it enables objective comparison of multiple machine learning algorithms under identical preprocessing and evaluation settings. Similar experimental designs have been widely adopted in recent sentiment analysis research to assess the effectiveness of text preprocessing techniques and classification models [11].

The research consisted of six sequential stages: data acquisition, text preprocessing, feature extraction, machine learning classification, model validation, and performance evaluation. Each stage was designed to ensure that the influence of stemming could be objectively measured without introducing variations from other experimental components. The overall research framework is presented in Figure 1.



Figure 1. Research Framework

2.2 Data Acquisition

The dataset used in this study was the IMDb 50K Movie Reviews dataset obtained from the Kaggle repository. The dataset consists of 50,000 English-language movie reviews, equally distributed into 25,000 positive reviews and 25,000 negative reviews. Each record contains two attributes: **review**, representing the textual content, and **sentiment**, representing the corresponding sentiment label.

Prior to model development, an Exploratory Data Analysis (EDA) was conducted to identify the characteristics of the dataset, including data completeness, class distribution, and text characteristics. The analysis indicated that the dataset contains no missing values and exhibits a balanced class distribution, making it suitable for supervised sentiment classification. Similar data acquisition and exploratory analysis procedures have been implemented in recent sentiment analysis studies to ensure dataset quality before model construction [12].

Table 1 summarizes the characteristics of the dataset used in this research.

Table 1. Dataset Characteristics

Characteristics	Description
Dataset	IMDb 50k Movie Reviews
Source	Kaggle
Number of Documents	50,000
Positive Reviews	25,000
Negative Reviews	25,000
Attribute	Review, Sentiment

2.3 Text Preprocessing

Text preprocessing was performed to convert raw textual reviews into structured data suitable for machine learning algorithms. The preprocessing pipeline consisted of HTML tag removal, case folding, tokenization, stop-word removal, and stemming. These preprocessing techniques have been widely employed in recent sentiment analysis research to improve text quality prior to feature extraction [13].

To investigate the influence of stemming, two preprocessing scenarios were designed.

Scenario I applied HTML tag removal, case folding, tokenization, and stop-word removal without performing stemming.

Scenario II applied the same preprocessing pipeline followed by stemming using the Porter Stemmer algorithm before feature extraction.

The comparison between these two scenarios enabled the effect of stemming on feature representation and model performance to be objectively evaluated while maintaining identical experimental conditions.

2.4 Feature Extraction

Following preprocessing, each movie review was transformed into numerical feature vectors using the Term Frequency–Inverse Document Frequency (TF-IDF) weighting scheme. TF-IDF assigns higher weights to terms that frequently occur within a document but appear less frequently across the entire corpus, thereby improving the discriminative capability of textual features [14].

$$TF(t, d) = \frac{f_{t,d}}{\sum t' f_{t',d}} \quad (1)$$

$$IDF(t) = \log \left(\frac{N}{DF(t)} \right) \quad (2)$$

$$TFIDF(t, d) = TF(t, d) \times IDF(t) \quad (3)$$

Where $f(t, d)$ denotes the raw frequency of term t in document d , N represents the total number of documents, and $DF(t)$ indicates the number of documents containing term t . In addition to generating feature vectors for classification, the total number of TF-IDF dimensions was recorded to evaluate feature complexity under each preprocessing scenario.

2.5 Machine Learning Classification

Four supervised machine learning algorithms representing different learning paradigms were employed in this study, namely Naïve Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT), and Random Forest (RF). These algorithms were selected because they are among the most widely used methods in sentiment analysis research and demonstrate different learning characteristics when handling high-dimensional textual data [15].

Naïve Bayes represents a probabilistic learning approach suitable for sparse textual features. Support Vector Machine constructs an optimal separating hyperplane and has consistently demonstrated superior performance in text classification tasks. Decision Tree generates hierarchical decision rules based on feature selection criteria, whereas Random Forest combines multiple decision trees using the ensemble learning strategy to improve prediction stability and reduce overfitting.

To ensure a fair comparison, identical TF-IDF representations, training data partitions, and validation strategies were applied to all classification algorithms.

2.6 Research Procedure

The research procedure implemented in this study is summarized in Figure 2.

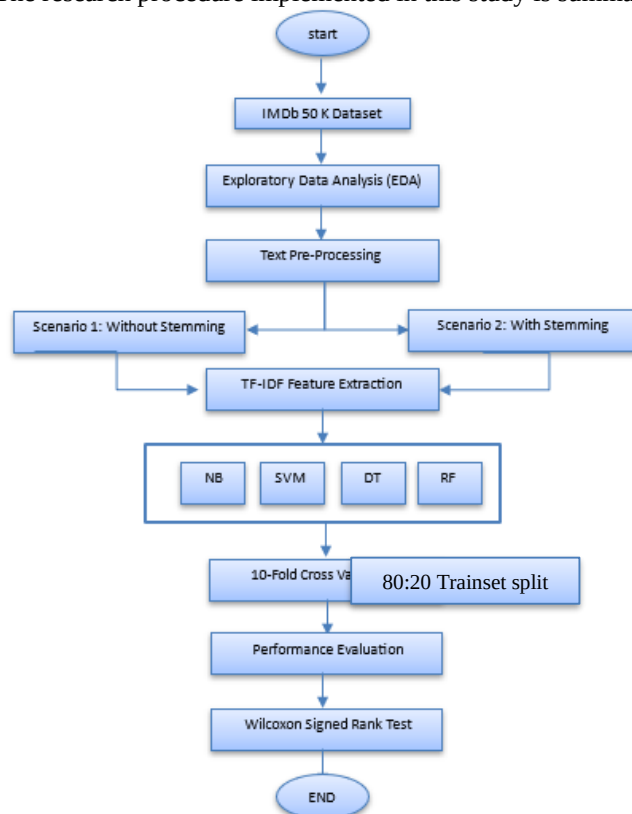


Figure 2. Research Procedure

Figure 2 illustrates the research procedure employed in this study. The research began with the acquisition of the IMDb 50K Movie Reviews dataset, which was subsequently analyzed using *Exploratory Data Analysis* (EDA) to identify its characteristics, including data quality, class distribution, and textual properties. Subsequently, two text preprocessing scenarios were implemented, namely preprocessing without

stemming and preprocessing with stemming. In both scenarios, textual data were transformed into numerical feature representations using the *Term Frequency-Inverse Document Frequency* (TF-IDF) method. The resulting feature vectors were then used as input for four supervised machine learning algorithms: Naïve Bayes, Support Vector Machine, Decision Tree, and Random Forest.

The primary classification experiments were conducted using an 80:20 train-test split. A separate 10-fold stratified cross-validation procedure was subsequently performed to obtain fold-level accuracy scores for the Wilcoxon Signed-Rank Test. Model performance was evaluated based on classification metrics, including accuracy, precision, recall, and F1-score, as well as feature complexity measured by the number of TF-IDF dimensions and computational efficiency measured by training and testing time. Finally, the Wilcoxon Signed-Rank Test was performed to determine whether the performance differences observed between the stemming and non-stemming scenarios were statistically significant.

2.7 Model Validation

The primary classification experiments used an 80:20 train-test split with a fixed random state of 42. The same data partition was applied to the four classification algorithms under both preprocessing scenarios to ensure a consistent comparison. Classification performance was subsequently evaluated on the held-out test set using accuracy, precision, recall, and F1-score.

In addition to the primary train-test evaluation, a separate 10-fold stratified cross-validation procedure was conducted for statistical analysis. The cross-validation results were used to obtain accuracy scores for each fold under the stemming and non-stemming scenarios. These paired fold-level scores were subsequently compared using the Wilcoxon Signed-Rank Test.

2.8 Performance Evaluation

The proposed models were evaluated using three complementary evaluation dimensions.

The first dimension was classification performance, which was measured using Accuracy, Precision, Recall, and F1-score. These metrics provide comprehensive information regarding the predictive capability of each classification algorithm [16].

The second dimension was feature complexity, measured by the total number of TF-IDF features generated after each preprocessing scenario. This evaluation was conducted to determine the effectiveness of stemming in reducing feature dimensionality.

The third dimension was computational efficiency, evaluated by measuring model training time and testing time. Lower computational cost indicates greater efficiency, particularly when processing high-dimensional textual datasets.

Finally, statistical significance analysis was performed using the Wilcoxon Signed-Rank Test with a significance level of $\alpha = 0.05$. A p-value lower than 0.05 indicates that the observed differences between the stemming and non-stemming scenarios are statistically significant rather than resulting from random variation.

2.9 Computational Environment

The experiments were conducted using Google Colaboratory (Google Colab) as the computational platform. The experimental environment used Python 3.12.13, Scikit-learn 1.6.1, and NLTK 3.9.1. The runtime environment was based on an Intel(R) Xeon(R) CPU @ 2.20 GHz with 2 CPU cores and approximately 12.67 GB of RAM. No NVIDIA GPU was available during the runtime environment used for the experiment.

3. RESULT AND DISCUSSION

A series of comparative experiments was conducted using the IMDb 50K Movie Reviews dataset to investigate the impact of stemming on sentiment classification. The evaluation considered four complementary aspects, namely classification performance, feature complexity, computational efficiency, and statistical significance. By comparing preprocessing scenarios with and without stemming, this study provides a comprehensive assessment of the influence of lexical normalization on machine learning-based sentiment analysis.

3.1 Results

3.1.1 Classification Performance

The predictive performance of the four supervised machine learning algorithms was evaluated using an 80:20 train-test split with a fixed random state of 42. The obtained results, including Accuracy, Precision, Recall, and F1-score, are presented in Table 2.

Table 2. Classification Performance Comparison

Algorithm	Condition	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Naïve Bayes	Without Stemming	86.64	87.78	85.37	86.56
Naïve Bayes	With Stemming	86.06	87.05	84.98	86.00
SVM	Without Stemming	89.74	89.20	90.61	89.90
SVM	With Stemming	89.32	88.47	90.61	89.53
Decision Tree	Without Stemming	72.57	73.17	71.94	72.55
Decision Tree	With Stemming	71.76	72.15	71.60	71.87
Random Forest	Without Stemming	86.13	86.56	85.79	86.18
Random Forest	With Stemming	85.30	85.30	85.57	85.44

To facilitate the comparison of classification performance across all evaluation metrics, Figure 3 illustrates the Accuracy, Precision, Recall, and F1-score obtained by each classifier under the two preprocessing scenarios.

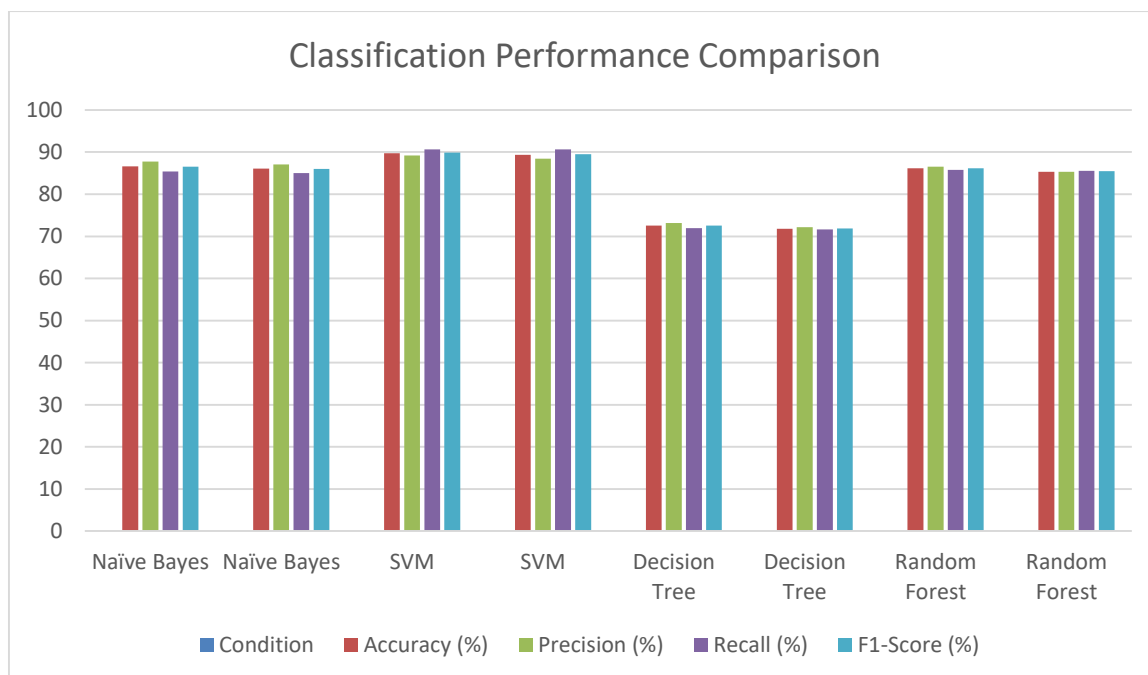


Figure 3. Classification Performance Comparison

The experimental results demonstrate that Support Vector Machine (SVM) consistently achieved the highest predictive performance under both preprocessing scenarios, reaching an accuracy of 89.74% without stemming. Conversely, Decision Tree yielded the lowest classification accuracy among the evaluated algorithms. Although the reductions in performance were relatively small, all classifiers experienced a decline in accuracy after stemming was applied. This observation indicates that lexical normalization through stemming did not improve sentiment classification performance on the IMDB dataset and, instead, slightly reduced the discriminative capability of the extracted textual features.

3.1.2 Feature Complexity Analysis

To evaluate the influence of stemming on feature representation, the dimensionality of the TF-IDF feature space was analyzed for both preprocessing scenarios. The resulting statistics are summarized in Table 3.

Table 3. TF-IDF Feature Complexity

Condition	Number of Features
Without Stemming	99,249
With Stemming	68,991
Feature Reduction	30.49%

The application of stemming substantially reduced the dimensionality of the TF-IDF feature space, decreasing the number of extracted features by 30.49%. This reduction was achieved by transforming multiple inflected and derived word forms into common root forms, thereby eliminating redundant lexical variants and reducing vocabulary size. The resulting compact feature representation is expected to improve computational efficiency while maintaining the essential semantic information required for classification.

3.1.3 Computational Efficiency

The reduction in feature dimensionality also influenced the computational cost of model construction and prediction. Table 4 presents the training and testing times obtained for each classification algorithm.

Table 4. Runtime Comparison

Algorithm	Condition	Training Time (s)	Testing Time (s)
Naïve Bayes	Without Stemming	0.159	0.035
Naïve Bayes	With Stemming	0.107	0.013
SVM	Without Stemming	1.974	0.006
SVM	With Stemming	2.098	0.005
Decision Tree	Without Stemming	140.104	0.062
Decision Tree	With Stemming	111.228	0.054
Random Forest	Without Stemming	342.067	1.435
Random Forest	With Stemming	279.889	1.125

The experimental results indicate that stemming generally improved computational efficiency for Naïve Bayes, Decision Tree, and Random Forest, as reflected by reduced training and testing times. The most substantial improvements were observed in tree-based algorithms, where Decision Tree and Random Forest reduced their training times by approximately 20.6% and 18.2%, respectively. In contrast, SVM showed a slight increase in training time after stemming, although its testing time decreased slightly.

3.1.4 Statistical Significance Analysis Using the Wilcoxon Signed-Rank Test

To verify whether the observed performance differences between the stemming and non-stemming scenarios were statistically significant, the Wilcoxon Signed-Rank Test was performed on the accuracy scores obtained from the 10-fold cross-validation procedure. A significance level of $\alpha = 0.05$ was adopted, where a p-value below 0.05 indicates a statistically significant difference.

Table 5. Wilcoxon Signed-Rank Test Results

Algorithm	Accuracy Without Stemming (%)	Accuracy With Stemming (%)	Difference (%)	P-value	Conclusion
Naïve Bayes	86.87	86.17	-0.70	0.001953	Significant
SVM	89.93	89.41	-0.52	0.003906	Significant
Decision Tree	72.39	72.29	-0.10	0.716797	Not significant
Random Forest	85.83	85.37	-0.46	0.019531	significant

The statistical analysis confirms that the influence of stemming varied across the evaluated classifiers. Naïve Bayes, Support Vector Machine, and Random Forest produced p-values below the significance threshold, indicating that the observed differences in classification performance were statistically significant despite their relatively small magnitude. These findings suggest that changes in lexical representation introduced by stemming had a measurable impact on the learning process of these algorithms. In contrast, Decision Tree produced a p-value considerably greater than 0.05, indicating that its slight reduction in accuracy was not statistically significant. This result implies that Decision Tree was comparatively less sensitive to the feature representation changes introduced by stemming.

3.2 Discussion

The findings of this study demonstrate that the effectiveness of stemming cannot be evaluated solely from the perspective of classification accuracy. Instead, its impact should be examined through multiple complementary dimensions, including predictive performance, feature complexity, computational efficiency, and statistical significance. This multidimensional evaluation provides a more comprehensive understanding of how text normalization influences machine learning performance in sentiment analysis.

The experimental results reveal a clear trade-off between predictive performance and computational efficiency. The application of stemming successfully reduced the TF-IDF feature space from 99,249 to 68,991 features, representing a 30.49% reduction in feature dimensionality. This substantial reduction confirms that stemming effectively merges multiple inflected and derived word forms into common root forms, thereby eliminating redundant lexical representations. From a computational perspective, a more compact feature space decreases memory requirements and simplifies matrix operations during both training and prediction. Consequently, stemming contributes positively to computational efficiency, particularly when processing large-scale textual datasets.

Despite the significant reduction in feature dimensionality, none of the evaluated machine learning algorithms exhibited improved classification performance after stemming was applied. Instead, all models experienced slight decreases in accuracy, indicating that lexical normalization does not always benefit sentiment classification. This finding suggests that several morphological variations preserved meaningful semantic information within IMDB movie reviews. Words sharing the same lexical root may still convey different contextual nuances, emotional intensity, or grammatical functions. When these variations are reduced to a single root form, part of the discriminative information useful for distinguishing positive and negative sentiments may be lost, resulting in a slight decline in predictive performance.

Among the evaluated algorithms, Support Vector Machine consistently achieved the highest classification performance under both preprocessing scenarios. This result demonstrates the robustness of SVM when handling high-dimensional TF-IDF representations. Unlike tree-based learning methods, SVM constructs an optimal separating hyperplane by maximizing the margin between classes, enabling it to maintain strong predictive performance even when the feature space contains a large number of dimensions. The relatively small decrease in SVM accuracy after stemming indicates that its decision boundary remains stable despite changes in lexical representation, confirming its suitability for sentiment classification involving sparse, high-dimensional textual features.

In contrast, Decision Tree and Random Forest exhibited greater sensitivity to stemming. Although Random Forest maintained competitive predictive performance, it experienced the largest reduction in accuracy among all evaluated classifiers. Decision Tree also showed a decline in predictive performance, although the Wilcoxon Signed-Rank Test indicated that this decrease was not statistically significant. These observations suggest that tree-based learning algorithms rely more heavily on lexical diversity during the attribute-selection process. When multiple word variations are merged into identical root forms, the availability of discriminative splitting attributes decreases, potentially reducing the quality of the generated decision rules. However, the non-significant statistical result obtained for Decision Tree also indicates that its observed performance reduction may be attributable to natural variation rather than a systematic effect of stemming.

Although stemming did not improve predictive accuracy, its contribution to computational efficiency was substantial. The reduction in TF-IDF dimensionality accelerated model construction, particularly for tree-based algorithms. Decision Tree achieved a training-time reduction of approximately 20.6%, while Random Forest reduced training time by approximately 18.2%. These improvements indicate that reducing the number of candidate features substantially decreases the computational burden associated with repeated feature evaluation during node splitting. Consequently, the computational advantages of stemming become increasingly important for large-scale sentiment analysis systems where processing speed and resource utilization are critical considerations.

The statistical significance analysis further strengthens the reliability of these findings. The Wilcoxon Signed-Rank Test produced statistically significant results for Naïve Bayes, Support Vector Machine, and Random Forest, indicating that the observed performance differences between the stemming and non-stemming scenarios were unlikely to have occurred by chance. Conversely, the non-significant result obtained for Decision Tree suggests that this algorithm is comparatively more robust to changes in feature representation introduced by stemming. These findings demonstrate the importance of complementing conventional evaluation metrics with statistical hypothesis testing to distinguish genuine methodological effects from random experimental variation.

The results of this study also address the research gaps identified in the Introduction. Previous sentiment analysis studies have generally evaluated stemming using conventional performance metrics such as accuracy, precision, recall, and F1-score. In contrast, this study demonstrates that the effectiveness of

preprocessing techniques should be assessed from a broader perspective that includes feature complexity, computational efficiency, and statistical significance. By integrating these complementary evaluation dimensions, the proposed framework provides a more comprehensive assessment of preprocessing effectiveness than studies that rely exclusively on predictive performance.

Overall, the findings indicate that the decision to apply stemming should be aligned with the primary objective of the sentiment analysis system. When maximizing classification accuracy is the principal objective, stemming may not be necessary because lexical normalization can remove informative semantic cues embedded in textual reviews. Conversely, when computational efficiency, memory optimization, and feature-space reduction are prioritized, stemming represents an effective preprocessing strategy. Therefore, the contribution of stemming should be interpreted as an optimization technique that balances predictive performance against computational resource requirements rather than as a preprocessing step intended solely to improve classification accuracy.

The principal contribution of this study lies in demonstrating that stemming introduces a measurable trade-off between predictive performance and computational efficiency in TF-IDF-based sentiment analysis. Unlike previous studies that focused predominantly on classification accuracy, this research provides empirical evidence that preprocessing techniques should be evaluated using multiple complementary dimensions. This comprehensive evaluation framework offers a broader understanding of how lexical normalization influences machine learning performance and provides practical guidance for selecting preprocessing strategies according to specific application requirements.

4. CLOSING

This study investigated the impact of stemming on sentiment classification by evaluating its influence on classification performance, TF-IDF feature complexity, computational efficiency, and statistical significance using the IMDb 50K Movie Reviews dataset. Four supervised machine learning algorithms, namely Naïve Bayes, Support Vector Machine (SVM), Decision Tree, and Random Forest, were employed to compare two preprocessing scenarios: without stemming and with stemming.

The experimental results indicate that the application of stemming successfully reduced the TF-IDF feature space from 99,249 to 68,991 features, representing a 30.49% reduction in feature dimensionality. This reduction improved computational efficiency by decreasing model training time, particularly for tree-based algorithms, where Decision Tree and Random Forest achieved training-time reductions of approximately 20.6% and 18.2%, respectively. These findings confirm that stemming is an effective preprocessing technique for simplifying textual feature representations and reducing computational cost.

However, the reduction in feature complexity was not accompanied by improvements in predictive performance. All evaluated classifiers experienced slight decreases in classification accuracy after stemming was applied. Among the evaluated models, Support Vector Machine consistently achieved the highest classification performance under both preprocessing scenarios, while Decision Tree produced the lowest predictive performance. Furthermore, the Wilcoxon Signed-Rank Test confirmed that the performance differences observed for Naïve Bayes, Support Vector Machine, and Random Forest were statistically significant, whereas the changes observed for Decision Tree were not statistically significant. These findings demonstrate that the influence of stemming varies across machine learning algorithms and should not be generalized without empirical evaluation.

The principal contribution of this study is the comprehensive evaluation of stemming from multiple complementary perspectives rather than relying solely on classification accuracy. The findings demonstrate that stemming introduces a measurable trade-off between predictive performance and computational efficiency. While lexical normalization effectively reduces feature dimensionality and computational requirements, it may simultaneously remove discriminative lexical information that remains valuable for sentiment classification. Therefore, the effectiveness of preprocessing techniques should be assessed using multidimensional evaluation criteria, including predictive performance, feature complexity, computational efficiency, and statistical significance.

From a practical perspective, the findings suggest that the selection of preprocessing strategies should be aligned with the primary objective of the sentiment analysis system. If maximizing classification accuracy is the primary concern, stemming may not always be necessary. Conversely, if computational efficiency, memory optimization, and feature-space reduction are prioritized, stemming represents an appropriate preprocessing technique for large-scale text mining applications.

Future research may extend this work by investigating alternative text normalization techniques, such as lemmatization or contextual normalization, and by integrating advanced feature representation methods, including word embeddings and transformer-based language models. Furthermore, evaluating these preprocessing techniques across multilingual datasets and diverse application domains would provide broader

insights into the relationship between lexical normalization, feature representation, and sentiment classification performance.

REFERENCES

- [1] Andritri, R. Firliana dan A. Nugroho, "Analisis Sentimen Pada Ulasan IMDb Dengan Algoritma Naïve Bayes dan SVM," *Jurnal Informatika, Komputer dan Bisnis (JIKOBIS)*, vol. 5, no. 1, 2025.
- [2] D. Subedi, N. Lamichhane dan N. Subedi, "Sentiment Analysis of IMDb Movie Reviews Using SVM and Naive Bayes Classifier," *Journal of Engineering and Sciences*, vol. 4, no. 1, pp. 56-68, 2025.
- [3] M. Fikri dan Z. Abidin, "Analysis Of The Use Of Nazief-Adriani Stemming And Porter Stemming In Covid-19 Twitter Sentiment Analysis With Term Frequency-Inverse Document Frequency Weighting Based On K-Nearest Neighbor Algorithm," *Recursive Journal of Informatics*, vol. 2, no. 2, pp. 80-87, 2024.
- [4] N. V. Pusean, N. Charibaldi dan B. Santosa, "Comparison of Scenario Pre-processing Performance on Support Vector Machine and Naïve Bayes Algorithms for Sentiment Analysis," *Inform: Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi*, vol. 8, no. 1, pp. 57-63, 2023.
- [5] Asro dan Solihin, "Comparative Evaluation of Preprocessing Techniques in Twitter Sentiment Analysis for Indonesia's 2024 Regional Elections," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 11, no. 1, pp. 150-159, 2024.
- [6] E. V. Arends, "Perbandingan Kinerja Metode Pembelajaran Mesin untuk Analisis Sentimen Ulasan Film," *Jurnal Ilmu Komputer dan Sistem Informasi*, vol. 13, no. 1, 2025.
- [7] A. Fitrianto, S. A. R. Manaf dan A. M. Soleh, "Perbandingan Algoritma Pohon dengan Beberapa Skenario Pelabelan untuk Analisis Sentimen pada Aplikasi Milik Pemerintah/BUMN," *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, vol. 10, no. 1, pp. 24-32, 2024.
- [8] M. Y. Alghifari, M. R. Sanjaya, D. R. Indah dan E. L. Ruskan, "Comparison of SVM and Naive Bayes Algorithms in Sentiment Analysis of User Reviews on Bukalapak," *INOVTEK Polbeng – Seri Informatika*, vol. 10, no. 3, pp. 1623-1633, 2025.
- [9] W. Ningsih, B. Alfianda, Rahmadden dan D. Wulandari, "Perbandingan Algoritma SVM dan Naïve Bayes dalam Analisis Sentimen Twitter pada Penggunaan Mobil Listrik di Indonesia," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 2, pp. 556-562, 2024.
- [10] F. Anisa dan A. Purwanto, "Design of a Sentiment Analysis System for Indonesian-Language International Conflict News using Naïve Bayes and SVM," *SISTEMASI: Jurnal Sistem Informasi*, vol. 15, no. 5, pp. 1861-1872, 2026.
- [11] H. Barus, I. N. Fajri dan Y. Pristyanto, "Sentiment Classification Analysis of Tokopedia Reviews Using TF-IDF, SMOTE, and Traditional Machine Learning Models," *JAIC*, vol. 9, no. 5, pp. 2552-2561, 2025.
- [12] Asro dan Solihin, "Comparative Evaluation of Preprocessing Techniques in Twitter Sentiment Analysis for Indonesia's 2024 Regional Elections," *INOVTEK POLBENG-Seri Informatika*, vol. 11, no. 1, pp. 150-159, 2026.
- [13] Y. A. Wahid, Sanatang dan D. D. Andayani, "Performance Comparison of Svm and Naïve Bayes For Indonesian-Language Sentiment Analysis On Free Fire Reviews Using Tf-Idf And Smote," *JESSI*, vol. 6, no. 4, pp. 674-689, 2025.
- [14] J. Sutrisno dan J. G. E. Putri, "Analisis Sentimen Pada Aplikasi Livin' By Mandiri Menggunakan Metode Tf-Idf Dan Naive Bayes," *JIEMS*, vol. 17, no. 1, pp. 1-10, 2024.
- [15] I. G. B. A. Budaya dan I. K. P. Suniantara, "Comparison of Sentiment Analysis Algorithms with SMOTE Oversampling and TF-IDF Implementation on Google Reviews for Public Health Centers," *Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 3, pp. 1077-1086, 2024.
- [16] A. S. Muliana, D. Lestarini dan S. P. Raflesia, "Analysis of Public Sentiment on Election Results using Naïve Bayes in Social Media X," *SISTEMASI*, vol. 13, no. 6, pp. 2467-2478, 2024.
- [17] N. C. Ramadani, I. Tahyudin dan A. S. Barkah, "Perbandingan Algoritma Support Vector Machine, Decision Tree, dan Logistic Regression pada Analisis Sentimen Ulasan Aplikasi Netflix," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 10, no. 2, 2024.