



Optimal Preprocessing and Feature Engineering Combination for DANA App Sentiment Classification: A Full Factorial Study

Samsudin^{1*}, Linda Putriani²

^{1,2}Department of Information System, Universitas Islam Indragiri, Indonesia

Abstract.

Purpose: Selecting optimal text-processing methods for sentiment analysis of digital financial application reviews, especially in Indonesian, is challenging. Previous studies have typically evaluated preprocessing techniques, feature engineering methods, and classification models in isolation, leaving the question of their optimal combination unresolved. This study aims to systematically determine the best pipeline for sentiment classification of DANA application reviews on the Google Play Store.

Methods: A full factorial experimental design was employed, generating 24 combinations across four preprocessing configurations (no preprocessing, stopword removal, stemming, and stopword removal combined with stemming), three feature extraction methods (Bag of Words, TF-IDF, and N-gram), and two classifiers (Naive Bayes and Logistic Regression). A dataset of 57,115 Indonesian reviews, comprising 34,435 positive (60.3%) and 22,680 negative (39.7%) instances, was used, with labels based on user star ratings.

Result: From the experimental results, it was found that the best setting is one that does not involve any pre-processing, the use of N-grams, and Logistic Regression, giving an F1-Score of 0.8772 and surpassing the baseline F1-Score by 1.82 percent. Logistic Regression has been found to be significantly better than Naive Bayes using the Mann-Whitney U test ($p=0.0001$). Preprocessing and feature extraction independently had significant effects on model performance, as indicated by the Kruskal-Wallis test ($p=0.0274$ for preprocessing and $p=0.0269$ for feature extraction). N-gram was proven superior to TF-IDF in classifying the negative class based on the Wilcoxon Signed-Rank test ($p=0.0039$). However, preprocessing and feature extraction did not jointly exert a significant effect on model performance based on the Scheirer-Ray-Hare test ($H=6.0117$, $p=0.3051$), and no significant interaction effect was found between both factors ($H=0.3683$, $p=0.9991$), implying that both factors can be independently optimized.

Novelty: This study contributes a systematic full factorial evaluation framework for Indonesian NLP pipelines in the fintech domain. Unlike previous studies that tested variables in isolation, this approach enables a fair and comprehensive comparison across method combinations, offering practitioners evidence-based guidance for sentiment analysis method selection. Future research may explore deep learning or large language model-based approaches to further improve classification performance.

Keywords: Sentiment analysis, Full factorial experiment, Feature engineering, Text preprocessing, Fintech app reviews

Received July 2026 / **Revised** September 2026 / **Accepted** October 2026

This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).



INTRODUCTION

The development of digital financial technology (fintech) in Indonesia has experienced significant growth in recent years. Digital wallet platforms such as GoPay, OVO, ShopeePay, and DANA have become the most popular payment methods among Indonesians, with usage penetration reaching 80% in 2025 [1]. This rapid growth has been driven by increasing internet penetration, the expansion of inclusive financial services, and post-pandemic shifts in consumer behavior toward digital transactions [2]. The high adoption of digital wallet applications generates millions of user reviews on the Google Play Store that serve as strategic information sources for developers to evaluate service quality, identify user complaints, and formulate feature development priorities [3].

Several studies have explored sentiment analysis on Indonesian digital application reviews using various approaches. Anggraini et al. [4] compared Naive Bayes and Logistic Regression using TF-IDF N-gram on Google Play Store reviews and found that Logistic Regression outperformed Naive Bayes with an F1-score of 87.91%; however, this study only tested a single preprocessing configuration and a single feature

* Corresponding author.

Email addresses: khozaimi@trunojoyo.ac.id (Khozaimi)

DOI: [10.15294/sji.v13i4.58535](https://doi.org/10.15294/sji.v13i4.58535)

extraction method. Syaputra et al. [5] classified sentiment from DANA, GoPay, and OVO digital wallet reviews using TF-IDF and Random Forest with SMOTE balancing, but did not systematically compare preprocessing variations. Fitriyanto and Purwanto [6] compared four classifiers (SVM, Logistic Regression, Naïve Bayes, and KNN) using TF-IDF on DANA application reviews from the Google Play Store and found that Naïve Bayes delivered the most stable performance with an accuracy of 79.80%; however, the study used only a single preprocessing configuration and a single feature extraction method without systematic comparison across variations. Ramadhan and Sugianto [7] classified sentiment from DANA application reviews on the Google Play Store using a Naïve Bayes Classifier on a dataset of 2,000 reviews and achieved an accuracy of 86%; however, this study did not compare variations in preprocessing or feature extraction methods systematically. Sianipar and Devega [8] analyzed the sentiment of DANA and LinkAja reviews using Naïve Bayes with only a single preprocessing configuration, without comparing alternative feature extraction methods. Munandar et al. [9] applied K-Nearest Neighbor to analyze sentiment in mobile banking application reviews, but focused solely on a single algorithm without feature variations. Nurian [10] tested Naïve Bayes on Google Play Store application reviews and demonstrated its effectiveness, yet without comparing different preprocessing configurations. Gerliandeva et al. [11] demonstrated that N-gram selection significantly affects sentiment classification accuracy when combined with TF-IDF and Naïve Bayes, yet did not integrate preprocessing variations into the experimental design.

These studies collectively reveal a significant gap in the literature: most existing works evaluate preprocessing techniques, feature extraction methods, and classification models in isolation or in limited combinations, making it impossible to determine their optimal joint configuration [12]. No study has systematically evaluated all 24 combinations from four preprocessing configurations, three feature extraction methods, and two classifiers simultaneously in the context of Indonesian fintech app reviews using a full factorial experimental design with rigorous statistical validation. Therefore, this research aims to determine the optimal sentiment analysis pipeline for DANA application reviews by employing a full factorial experimental design ($4 \text{ preprocessing} \times 3 \text{ feature engineering} \times 2 \text{ models} = 24 \text{ combinations}$), with statistical validation using Kruskal-Wallis, Mann-Whitney U, Wilcoxon Signed-Rank, and Scheirer-Ray-Hare tests to assess the significance of each experimental factor.

METHODS

Research design

This study employs a full factorial experiment design to evaluate the effect of combinations of three independent variables on sentiment classification performance, namely text preprocessing techniques with four levels (X1), feature extraction methods with three levels (X2), and classification models with two levels (X3), resulting in $4 \times 3 \times 2 = 24$ experimental combinations executed systematically. This approach was selected because it enables the simultaneous evaluation of the main effect of each factor as well as the interaction effects between factors. As presented in Figure 1, the overall research workflow encompasses seven interconnected stages, ranging from data collection, sentiment labeling, basic preprocessing, application of the full factorial experiment design, data splitting, model evaluation, and statistical hypothesis testing (H1–H8) in order to determine the optimal treatment combination.

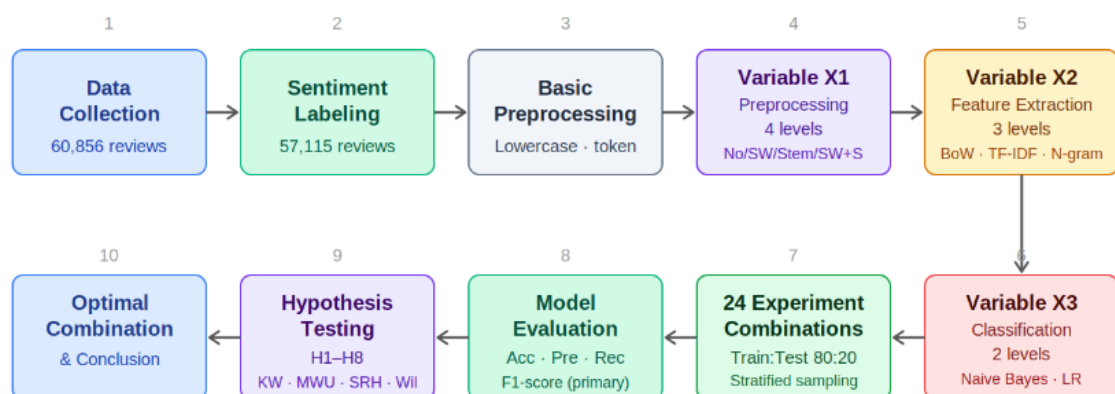


Figure 1. Research flowchart

Dataset

The dataset used in this study was sourced from Kaggle under the name `dana_reviews_raw.csv`, which contains user reviews of the DANA application on the Google Play Store in Indonesian [13]. Sentiment classification was done using the ratings. This means that ratings ranging between 4 to 5 stars were considered positive while those that were between 1 to 2 stars were considered negative. Rating of 3 stars was not taken into consideration due to neutrality [14], [15]. Out of a total of 60,856 initial records, 3,741 reviews with a 3-star rating were eliminated, resulting in a final dataset of 57,115 reviews, consisting of 34,435 positive reviews (60.3%) and 22,680 negative reviews (39.7%).

To provide a concrete illustration of the data characteristics, examples of positive reviews in the dataset include *"Aplikasi sangat membantu, transaksi cepat dan mudah digunakan"* (The app is very helpful, transactions are fast and easy to use, rating 5★) and *"Mantap banget, saldo langsung masuk tidak perlu nunggu lama"* (Really great, balance is instantly credited without waiting long, rating 4★), while examples of negative reviews include *"Saldo saya hilang tiba-tiba, sudah lapor tapi tidak ada respon dari cs"* (My balance suddenly disappeared, I reported it but received no response from customer service, rating 1★) and *"Kecewa banget, transfer gagal tapi saldo sudah terpotong"* (Very disappointed, transfer failed but balance was already deducted, rating 1★). These samples reflect the typical characteristics of Indonesian-language review text, namely the widespread use of informal language, abbreviations, and spelling variations, as well as the dominance of negation words such as *"tidak bisa"* (cannot) and *"tidak ada"* (there is no) in negative reviews, which serve as the primary justification for employing N-gram as a feature extraction method in this study.

As illustrated in Figure 2, exploratory data analysis was conducted to understand the fundamental characteristics of the dataset prior to the modeling process, covering three key aspects: sentiment class distribution, review length distribution in characters, and word count distribution per review. The class distribution analysis was performed to determine whether data balancing techniques such as SMOTE were required prior to model training, where the results revealed a moderate class imbalance (60.3% versus 39.7%), indicating that additional balancing techniques were not necessary. The review length and word count distribution analysis was conducted to understand the linguistic characteristics of the text to be processed, where a median review length of 49 characters and a median of 8 words per review indicate that the majority of reviews are concise and succinct. These findings carry direct implications for the selection of feature extraction methods, as short and concise reviews dominated by negation words tend to lose meaningful context when represented solely by unigrams, thereby reinforcing the justification for employing N-gram (unigram+bigram), which is capable of capturing phrase patterns and negation context in Indonesian-language text.

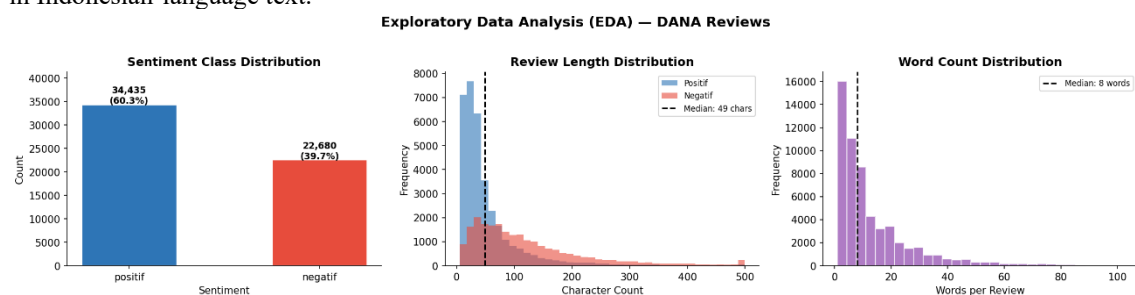


Figure 2. Exploratory data analysis 'dana' application reviews

Text preprocessing

The preprocessing stage aims to clean and normalize review text before it is extracted into numerical features [16]. Basic preprocessing applied across all experiments includes: converting text to lowercase (lowercasing), removal of URLs, punctuation, special characters, and numbers, as well as tokenization [17]. Subsequently, four preprocessing configurations were tested as variable X1, namely: (1) no additional preprocessing (No Prep), (2) stopword removal only using an Indonesian stopwords list from NLTK [18], (3) stemming only using the PySastrawi library specifically designed for the Indonesian language [19], and (4) a combination of stopword removal and stemming (SW+Stem) [20]. The effect of these four configurations on model performance was evaluated empirically following the approach proposed by Artanto et al. [21].

Feature extraction

Three feature extraction methods were used as variable X2 [22]. First, Bag of Words (BoW) represents text as a vector of word occurrence frequencies using CountVectorizer [23]. Second, Term Frequency-Inverse Document Frequency (TF-IDF) assigns a weight to each word based on its frequency of occurrence in a document relative to the entire corpus using TfidfVectorizer [24]. Third, the N-gram (TF-IDF unigram+bigram) feature expansion is a method to extends the TF-IDF model using unigrams (single token features), $n=1$, and bigrams (word pairs), $n=1-2$, through the parameter `gram_range=(1,2)`. This helps the model understand context and negations that are typical of Indonesian-language reviews and can't be identified by unigram models alone [25]. The maximum number of features (*max_features*) was uniformly set to 10,000 across all feature extraction methods. To prevent data leakage, all vectorizers were fitted exclusively on the training data and subsequently applied (*transform*) to the test data without learning any word distribution from the test set, ensuring that the evaluation results reflect genuine model performance on previously unseen data.

Classification models

Two classification models were used as variable X3. First, Multinomial Naive Bayes (NB) was selected due to its efficiency in text classification and its ability to handle high-dimensional data [26]. This model operates by computing the conditional probability of each word given a sentiment class based on Bayes' theorem, where each feature is assumed to be independent of one another (*naive assumption*). Second, Logistic Regression (LR) was chosen as a comparative model because of its competitive performance in binary text classification tasks [27]. This model learns optimal weights for each feature through a sigmoid function that maps a linear combination of features to class probabilities. Both models were implemented using the Scikit-learn library with default parameters, except for the `max_iter` parameter in Logistic Regression, which was set to 1,000 to ensure convergence, and `random_state=42` to ensure reproducibility [28]. The workflow of both classification models from feature input to sentiment prediction output is presented in Figure 3.

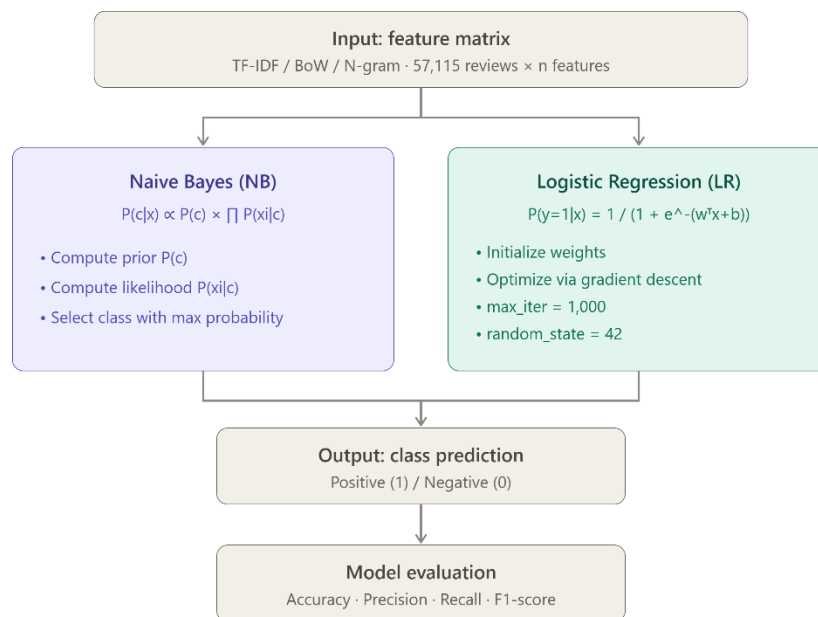


Figure 3. Classification model workflow naive bayes and logistic regression

Data splitting

The dataset was partitioned into training and test sets at an 80:20 ratio using stratified sampling to preserve class proportions across both subsets, with `random_state=42` consistently applied to both the `train_test_split()` function and the Logistic Regression model initialization to ensure reproducibility [29]. The split was performed once and applied consistently across all 24 experiments to ensure the comparability of results. It should be explicitly noted that the 24 F1-score values reported in Table 2 are the results of single-run evaluation on the test set (20%) with a fixed `random_state=42`, not averages derived from 5-fold

cross-validation. The samples used in statistical hypothesis testing (H1–H4, H6–H8) were formed by regrouping the F1-score values from the 24 single-run experiments according to the factor under examination. For instance, to test the effect of preprocessing (H2), the 24 F1-score values were grouped into four subsets based on preprocessing configuration (No Prep, Stopword, Stemming, SW+Stem), each consisting of six values derived from the combination of three feature extraction methods and two classification models. The one-shot design with a predetermined random state was intentionally chosen to reduce variability and thus ensure that any differences in the F1 scores between the groups were due to the factor manipulation. As a separate additional validation from the primary evaluation, 5-fold cross-validation was conducted on a selected subset of experiments solely to confirm model performance stability and generalizability, and its resulting values were not used as the source of F1-score values in Table 2 nor as samples in the statistical hypothesis tests.

Model evaluation

Model performance was evaluated using four metrics: accuracy, precision, recall, and F1-score (macro average) [30]. Accuracy measures the overall proportion of correct predictions, precision measures the exactness of positive predictions, and recall measures the model's ability to detect all actual positive instances. The F1-score was selected as the primary evaluation metric because it provides a balance between precision and recall and is more robust to class imbalance [31]. All four metrics were computed using the macro average approach, which assigns equal weight to each class regardless of the number of samples per class, ensuring that performance on the minority class (negative, 39.7%) receives equal consideration to the majority class (positive, 60.3%).

Statistical hypothesis testing

To statistically validate the significance of each experimental factor, eight hypotheses were formulated and tested using non-parametric statistical methods as data distribution was not assumed to be normal [32], namely: (H1) preprocessing and feature engineering jointly exert a significant effect on model performance; (H2) preprocessing significantly affects model performance; (H3) feature engineering significantly affects model performance; (H4) there is a significant performance difference between Naive Bayes and Logistic Regression; (H5) the best parameter combination can be identified through experimentation; (H6) there is a significant interaction effect between preprocessing and feature engineering; (H7) stopword removal produces better performance than stemming; and (H8) N-gram produces a higher negative class F1-score compared to TF-IDF. The Scheirer-Ray-Hare test was used to evaluate the joint effect of preprocessing and feature extraction on model performance (H1) as well as the interaction effect between both factors (H6) [33]. The Kruskal-Wallis test was used to assess the independent main effects of preprocessing (H2) and feature extraction (H3) on F1-score [34]. The Mann-Whitney U test was used to compare the performance between Naive Bayes and Logistic Regression (H4) and to assess the marginal effect of stopword removal compared to stemming (H7) [35]. H5 was verified descriptively by identifying the experimental combination that produced the highest F1-score among all 24 full factorial combinations. The Wilcoxon Signed-Rank test was used to compare N-gram and TF-IDF for negative class classification (H8) [36]. All statistical tests were conducted at a significance level of $\alpha = 0.05$.

RESULT AND DISCUSSION

Labeling results and dataset distribution

The sentiment labeling process produced a final dataset of 57,115 reviews after eliminating 3,741 neutral-rated reviews (3 stars). The class distribution is presented in Table 1. The class imbalance of 60.3% against 39.7% is moderate, hence manageable through machine learning algorithms without need for balance measures, similar to findings that moderate class imbalances in large datasets may not significantly benefit from sampling methods like SMOTE [37].

Table 1. Sentiment class distribution.

Class	Count	Percentage
Positive (4–5★ Rating)	34,435	60.3%
Negative (1–2★ Rating)	22,680	39.7%
Neutral/Removed (3★ Rating)	3,741	—
Final Total	57,115	100%

Full factorial experiment results

All 24 experimental combinations were successfully executed and evaluated. Table 2 presents the complete results of all experiments, ranked in descending order by F1-score. The obtained F1-scores ranged from 0.8590 to 0.8772, indicating that all configurations achieved relatively strong classification performance. Nevertheless, the variation in F1-scores suggests that the choice of preprocessing technique, feature extraction method, and classification algorithm still influenced the overall model performance.

Table 2. Complete results of 24 full factorial experiments

Exp	Preprocessing	Feature Extraction	Model	Accuracy	Precision	Recall	F1-Score
6	No Preprocessing	N-gram	Logistic Regression	0.8528	0.8826	0.8719	0.8772
18	Stemming	N-gram	Logistic Regression	0.8525	0.8827	0.8711	0.8769
16	Stemming	TF-IDF	Logistic Regression	0.8484	0.8753	0.8728	0.8741
4	No Preprocessing	TF-IDF	Logistic Regression	0.8475	0.8749	0.8716	0.8733
2	No Preprocessing	Bag of Words	Logistic Regression	0.8397	0.8487	0.8934	0.8705
14	Stemming	Bag of Words	Logistic Regression	0.8384	0.8466	0.8940	0.8696
24	Stopword Removal + Stemming	N-gram	Logistic Regression	0.8429	0.8705	0.8686	0.8695
12	Stopword Removal	N-gram	Logistic Regression	0.8422	0.8704	0.8676	0.8690
10	Stopword Removal	TF-IDF	Logistic Regression	0.8412	0.8661	0.8714	0.8687
22	Stopword Removal + Stemming	TF-IDF	Logistic Regression	0.8388	0.8631	0.8708	0.8669
1	No Preprocessing	Bag of Words	Naive Bayes	0.8414	0.8841	0.8481	0.8657
8	Stopword Removal	Bag of Words	Logistic Regression	0.8317	0.8352	0.8979	0.8654
13	Stemming	Bag of Words	Naive Bayes	0.8398	0.8802	0.8500	0.8648
5	No Preprocessing	N-gram	Naive Bayes	0.8436	0.9040	0.8287	0.8647
20	Stopword Removal + Stemming	Bag of Words	Logistic Regression	0.8296	0.8327	0.8976	0.8640
17	Stemming	N-gram	Naive Bayes	0.8423	0.9011	0.8295	0.8638
15	Stemming	TF-IDF	Naive Bayes	0.8393	0.8910	0.8356	0.8624
3	No Preprocessing	TF-IDF	Naive Bayes	0.8393	0.8931	0.8332	0.8621
11	Stopword Removal	N-gram	Naive Bayes	0.8364	0.8780	0.8462	0.8618
23	Stopword Removal + Stemming	N-gram	Naive Bayes	0.8348	0.8745	0.8477	0.8609
19	Stopword Removal + Stemming	Bag of Words	Naive Bayes	0.8317	0.8587	0.8628	0.8607
7	Stopword Removal	Bag of Words	Naive Bayes	0.8315	0.8611	0.8590	0.8601
21	Stopword Removal + Stemming	TF-IDF	Naive Bayes	0.8333	0.8717	0.8484	0.8599
9	Stopword Removal	TF-IDF	Naive Bayes	0.8324	0.8714	0.8470	0.8590

The heatmap for the F1-score for all 24 configurations is shown in Figure 4. The best configuration varied from one classifier to another. For the Naive Bayes classifier, the best F1-score was obtained from the configuration without preprocessing and Bag of Words (0.8657), while for the Logistic Regression classifier, the best F1-score was obtained from the configuration without preprocessing and N-gram (0.8772).

The difference in optimal configurations between both models indicates that Naive Bayes is more sensitive to simple frequency-based features (BoW), while Logistic Regression is able to more effectively leverage the richer contextual information provided by N-gram. Furthermore, Logistic Regression consistently outperformed Naive Bayes across nearly all preprocessing and feature extraction configurations tested.

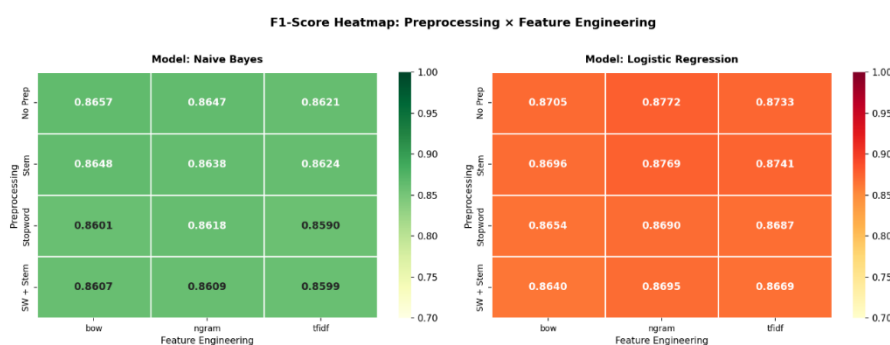


Figure 4. F1-score heatmap: preprocessing × feature engineering

These findings are further reinforced by Figure 5, which compares four evaluation metrics: accuracy, precision, recall, and F1-score across all 24 experiments, wherein Logistic Regression consistently achieved higher recall, while Naive Bayes tended to produce higher precision, with the best-performing experiment in each model highlighted by a light green area.

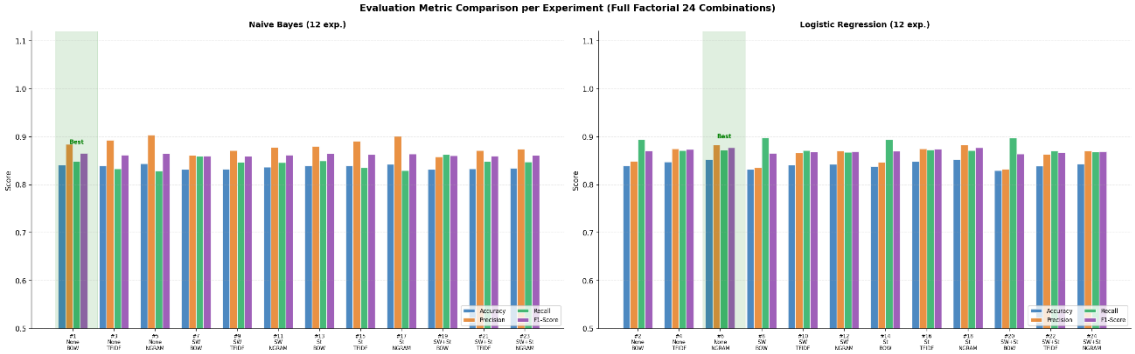


Figure 5. Evaluation metrics comparison per experiment (full factorial 24 combinations)

Optimal combination and performance analysis

The best combination from 24 experiments was Experiment #6: No Preprocessing + N-gram + Logistic Regression with an F1-score of 0.8772, surpassing the baseline (Experiment #9: Stopword + TF-IDF + Naive Bayes, F1=0.8590) by +1.82%, a finding consistent with Aryanusa et al. [38] that Logistic Regression substantially outperforms Naive Bayes on sentiment classification tasks for Indonesian-language mobile application reviews, while the superiority of N-gram as the best-performing feature extraction method can be attributed to its ability to capture phrase context and negation patterns in text such as "tidak bagus" or "kurang memuaskan" that cannot be captured by unigram BoW or standard TF-IDF [39],[40].

A particularly noteworthy finding is that the configuration without additional preprocessing (No Prep) yielded the best performance, indicating that for large-scale Indonesian-language application review datasets, the lexical information removed by stopwords removal and stemming actually contributes to sentiment feature discrimination, consistent with Pradana and Hayaty [41] who reported that the application of stemming and stopwords removal does not significantly improve sentiment analysis accuracy on Indonesian text and in fact tends to reduce it, as further confirmed by the confusion matrix in Figure 6 showing that the model successfully classified 6,005 positive reviews correctly (True Positive) and 3,737 negative reviews correctly (True Negative) with 882 False Negative cases and 799 False Positive cases, wherein the slightly higher False Negative rate indicates that the model more frequently misclassified positive reviews as negative, likely due to positive reviews containing partially negative phrases.

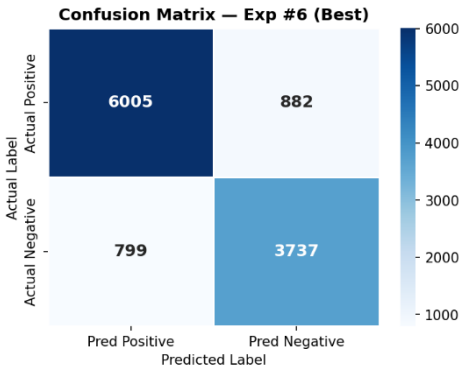


Figure 6. Confusion matrix experiment #6 (best)

Figure 7 and Figure 8 present word cloud visualizations of positive and negative reviews in the DANA dataset, where the size of each word proportionally represents its frequency of occurrence within each respective class. The word "dana" dominates both classes, yet a striking vocabulary contrast is evident as (Figure 7) is dominated by words such as "sangat" (very), "membantu" (helpful), "mudah" (easy), "mantap" (great), and "cepat" (fast), reflecting user satisfaction with the ease of use and speed of service, while

Indonesian-language users on digital review platforms, while negative features are dominated by profanity and extreme disappointment expressions.

The difference in feature characteristics between Naive Bayes (Figure 12) and Logistic Regression (Figure 11) reflects the difference in learning mechanisms between the two models, wherein Naive Bayes relies on independent word occurrence frequencies making it susceptible to informal spelling variations, while Logistic Regression learns feature weights discriminatively and produces more semantically meaningful features, further explaining why Logistic Regression is statistically superior to Naive Bayes across all experimental configurations ($p=0.0001$).

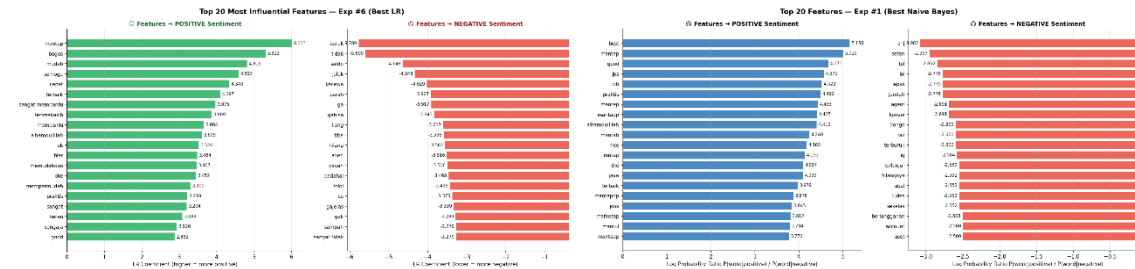


Figure 11. Top 20 most influential features experiment #6 (best lr model)

Figure 12. Top 20 most influential features experiment #1 (best nb model)

While Figure 13 reveals that reviews misclassified as False Negatives have a substantially higher median length (86 characters) compared to correct predictions (46 characters), indicating that longer and more complex reviews tend to be more difficult for the model to classify correctly.

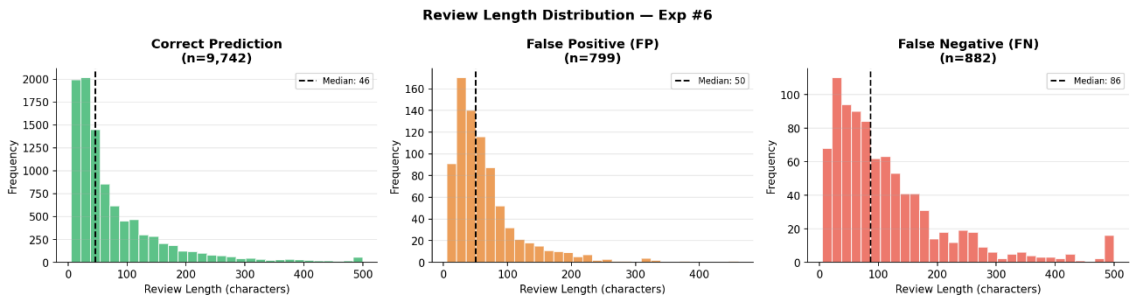


Figure 13. Review Length Distribution: Correct Predictions, False Positives, and False Negatives Experiment #6

Average F1-score comparison per factor

Table 3 summarizes the average F1-scores across each level of the three experimental factors.

Table 3. Average F1-score per factor level		
Factor	Level	Average F1-Score
Preprocessing (X1)	No Preprocessing	0.8742
	Stopword Removal	0.8676
	Stemming	0.8738
	Stopword Removal + Stemming	0.8657
Feature Extraction (X2)	Bag of Words (BoW)	0.8651
	TF-IDF	0.8658
	N-gram	0.8680
	Naive Bayes	0.8622
Model (X3)	Logistic Regression	0.8704

Logistic Regression consistently produced a higher average F1-score (0.8704) compared to Naive Bayes (0.8622) across all 12 experiments each. This difference is consistent with the findings of Monica dan

Purwanto [42], who compared multiple classifiers on Indonesian-language application reviews, although that study did not systematically isolate the effect of preprocessing.

As presented in Figure 14 through an individual effect analysis of each independent variable on the average F1-score, the left panel shows that the application of stopwords removal reduced the average F1-score by 0.0049 compared to no stopwords removal, the middle panel indicates that stemming had a marginal effect approaching zero (−0.0003), while the right panel confirms that N-gram yielded the highest average F1-score for both models, reinforcing the superiority of bigram-based feature extraction methods on Indonesian-language review datasets.

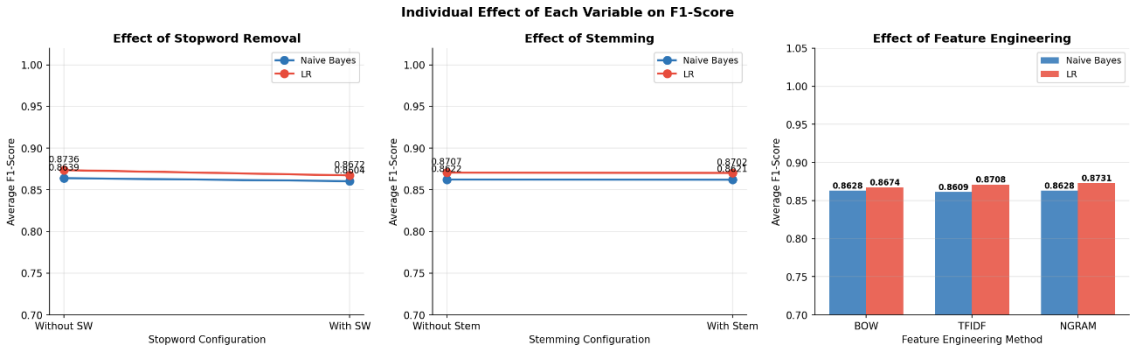


Figure 14. Individual effect of each variable on F1-score

Statistical hypothesis testing results (H1–H8)

Table 4 presents a summary of the eight hypothesis test results.

Tabel 4. Summary of statistical hypothesis testing results H1–H8

Hypothesis	Statement	Test Method	Statistic	p-value	Decision
H1	Joint effect of preprocessing and feature engineering on model performance	Scheirer-Ray-Hare	H = 6.0117	0.3051	✗ Fail to Reject H_0
H2	Preprocessing significantly affects model performance	Kruskal-Wallis	H = 9.1486	0.0274	☑ Reject H_0
H3	Feature engineering significantly affects model performance	Kruskal-Wallis	H = 7.2329	0.0269	☑ Reject H_0
H4	Naive Bayes and Logistic Regression differ significantly in performance	Mann-Whitney U	U = 4.0000	0.0001	☑ Reject H_0
H5	The optimal parameter combination can be identified through experimentation	Descriptive Analysis	—	—	☑ Confirmed
H6	The interaction effect between preprocessing (X1) and feature engineering (X2) is significant	Scheirer-Ray-Hare	H = 0.3683	0.9991	✗ Fail to Reject H_0
H7	Stopword removal outperforms stemming (marginal effect)	Mann-Whitney U	U = 0.0000	1.0000	✗ Fail to Reject H_0
H8	N-gram outperforms TF-IDF for the negative sentiment class	Wilcoxon Signed-Rank	W = 36.0000	0.0039	☑ Reject H_0

H1 failed to be rejected based on the Scheirer-Ray-Hare test ($H=6.0117$, $p=0.3051$), indicating that preprocessing and feature engineering did not jointly exert a significant effect on model performance, implying that the contribution of both factors is independent and does not produce a synergistic effect when combined. H2 was successfully rejected based on the Kruskal-Wallis test ($H=9.1486$, $p=0.0274$), confirming that preprocessing independently exerted a significant effect on model performance. H3 was also successfully rejected based on the Kruskal-Wallis test ($H=7.2329$, $p=0.0269$), confirming that the feature extraction method independently exerted a significant effect on model performance. H4 was successfully rejected based on the Mann-Whitney U test ($U=4.0000$, $p=0.0001$), revealing a highly significant performance difference between Logistic Regression and Naive Bayes, statistically confirming the superiority of Logistic Regression and reinforcing the findings of Palomino & Aider [43], who demonstrated that classifier selection significantly influences sentiment analysis model performance on Indonesian-language text. H5 was confirmed descriptively, where the best combination among all 24 full

factorial experiments was successfully identified as No Prep + N-gram + Logistic Regression, achieving the highest F1-score of 0.8772 and surpassing the baseline by 1.82%.

Conversely, H6 failed to be rejected based on the Scheirer-Ray-Hare test ($H=0.3683$, $p=0.9991$), indicating the absence of a significant interaction effect between preprocessing and feature extraction, implying that the effect of each preprocessing technique is consistent and additive across all feature extraction methods, thereby allowing independent optimization of both factors. H7 also failed to be rejected based on the one-sided Mann-Whitney U test ($U=0.0000$, $p=1.0000$), demonstrating that stopwords removal did not produce a greater F1-score improvement than stemming, with both techniques tending to reduce performance compared to no preprocessing, consistent with findings by Hadwan et al. [44], who confirmed that preprocessing techniques do not consistently improve performance in sentiment classification.

H8 was successfully rejected based on the Wilcoxon Signed-Rank test ($W=36.000$, $p=0.0039$), proving that N-gram significantly outperformed unigram TF-IDF in negative class classification, attributed to N-gram's capacity to capture negation phrases such as "tidak bisa" (cannot), "tidak mau" (will not), and "tidak pernah" (never), which frequently appear in negative Indonesian-language reviews but are lost when using only unigrams [45].

As presented in Figure 15, the nearly parallel lines in both panels of the interaction plot between preprocessing and feature engineering variables confirm the absence of a significant interaction effect ($p=0.9991$), indicating that the influence of each preprocessing technique is consistent and additive across all tested feature extraction methods.

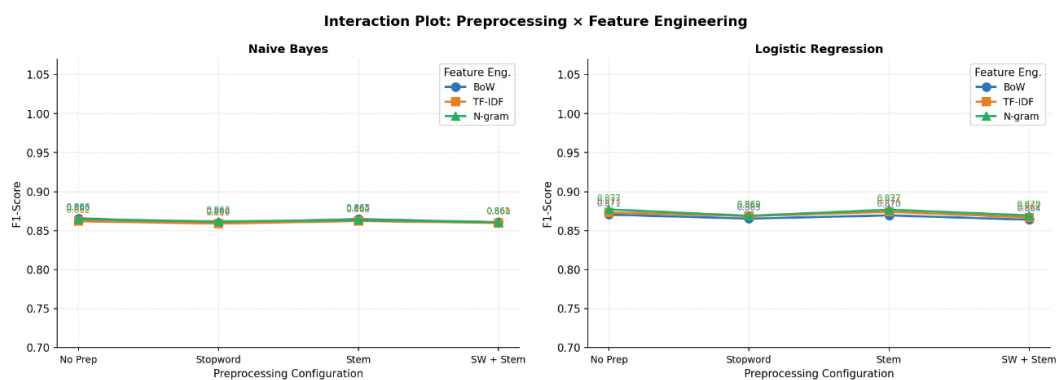


Figure 15. Interaction plot: preprocessing × feature engineering

Figure 16 further illustrates the H7 hypothesis testing results, where the left panel shows that stopwords removal consistently reduced F1-score (mean delta = -0.0049) while stemming had a near-zero effect, and the right panel reveals that stopwords removal exerted the greatest negative impact on N-gram (-0.0053), suggesting that the removal of stopwords eliminates important contextual information needed by N-gram to capture phrase patterns.

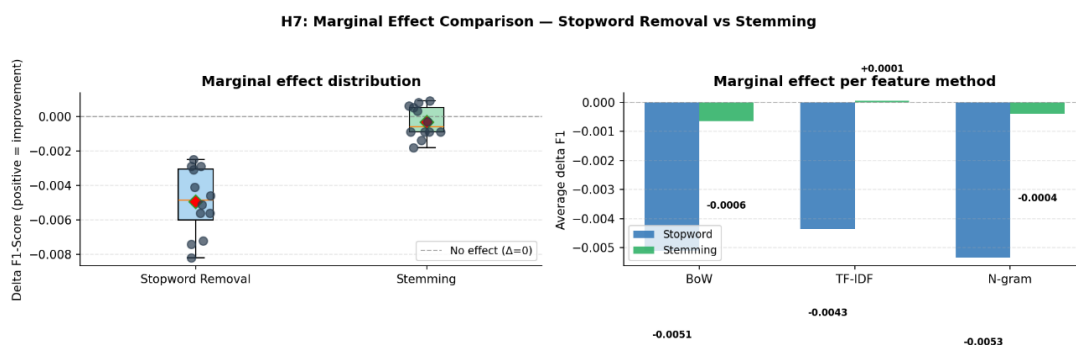


Figure 16. Marginal effect comparison of stopwords removal and stemming

Figure 17 presents the H8 hypothesis testing results comparing N-gram and TF-IDF performance specifically for negative class classification, where all 8 green lines in the paired comparison panel point upward from TF-IDF to N-gram, confirming that N-gram consistently achieved higher F1-scores across all experimental pairs, with the right panel numerically confirming this superiority through a mean negative class F1-score of 0.8076 for N-gram compared to 0.8018 for TF-IDF ($p=0.0039$).

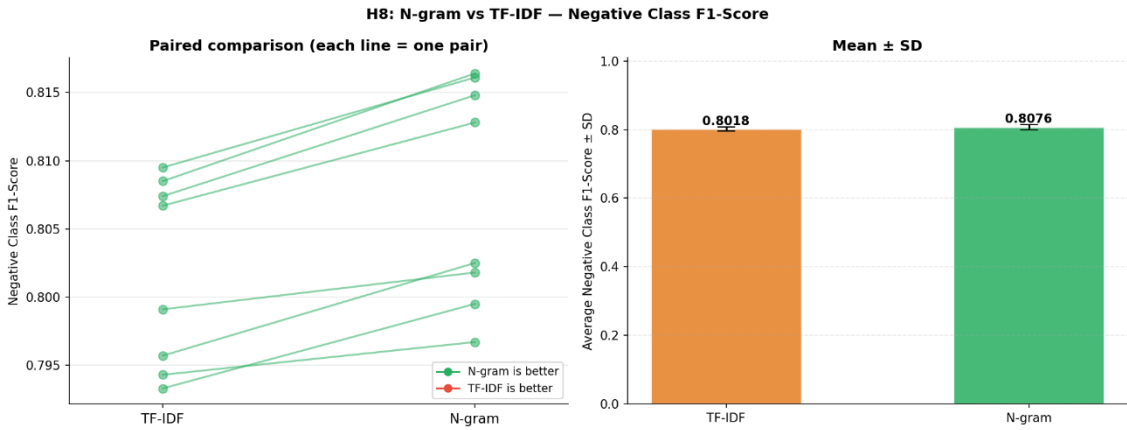


Figure 17. Negative class F1-score comparison: N-gram vs TF-IDF

Finally, Figure 18 displays the ROC curves of all 24 experiments, wherein the best Logistic Regression experiment (Exp #6: No Prep + N-gram) achieved the highest AUC of 0.929 and the best Naive Bayes experiment (Exp #5: No Prep + N-gram) reached an AUC of 0.920, with all experiments yielding AUC values above 0.890, indicating strong discriminative capability across all tested combinations.

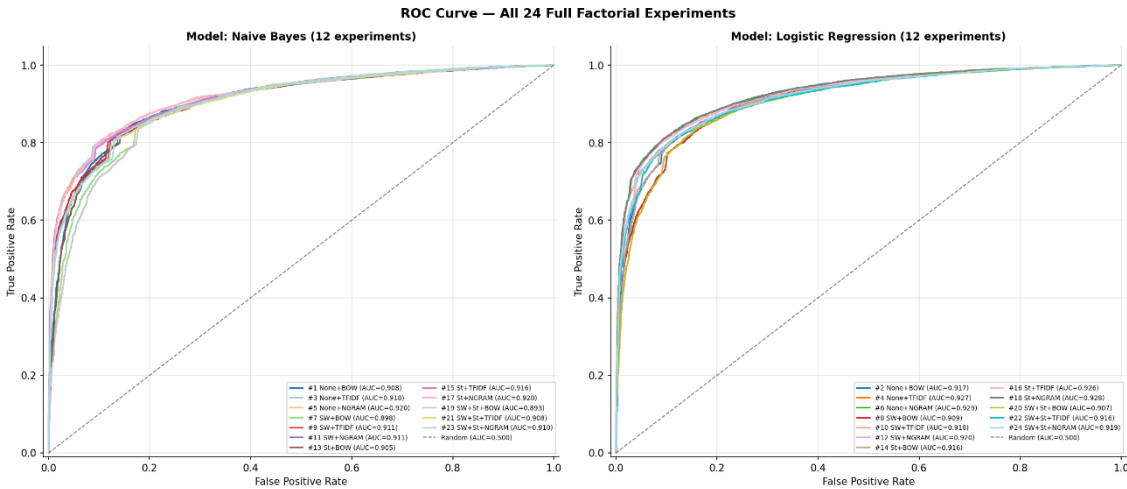


Figure 18. ROC curve all 24 full factorial experiments

Research implications

The findings of this study carry practical implications for developers of sentiment analysis systems for Indonesian-language fintech applications. First, the use of Logistic Regression with N-gram and no additional preprocessing is confirmed as the most optimal configuration and can be directly adopted as an efficient baseline pipeline [46]. Second, intensive preprocessing is unnecessary for large-scale Indonesian app review datasets, significantly reducing computational overhead [47]. Third, the absence of interaction effects between preprocessing and feature extraction (H6) simplifies system optimization, as both factors can be independently tuned without concern for confounding interactions. Future research may explore transformer-based models such as IndoBERT to further improve classification performance on Indonesian-language fintech reviews.

As illustrated in Figure 19, 5-fold cross-validation was conducted as a separate additional validation from the primary single-run evaluation that produced the F1-score values reported in Table 2, solely to confirm model performance stability and generalizability. The resulting F1-score distribution reveals narrow value ranges across all boxplots, where the left panel depicts the effect of preprocessing on Logistic Regression and Naive Bayes using TF-IDF while the right panel shows the effect of feature extraction methods using the SW+Stem configuration. These narrow value ranges indicate that both models exhibit low variance and stable performance across folds, thereby confirming that the single-run evaluation results in Table 2 are not artifacts of any particular data split, but rather reflect consistent and reliable model performance.

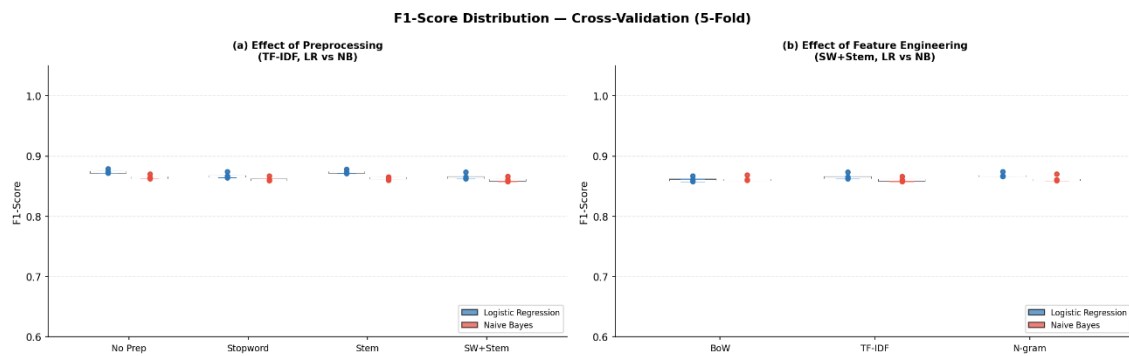


Figure 19. F1-score distribution 5-fold cross-validation

Figure 20 further presents the learning curves of the best-performing model from each classifier, wherein for both models the training score and validation score curves converge as the amount of training data increases, indicating the absence of significant overfitting or underfitting symptoms, although Logistic Regression demonstrates a slightly larger gap between the training score (~ 0.92) and validation score (~ 0.87), this gap narrows as data increases, confirming the model's good generalization capability.

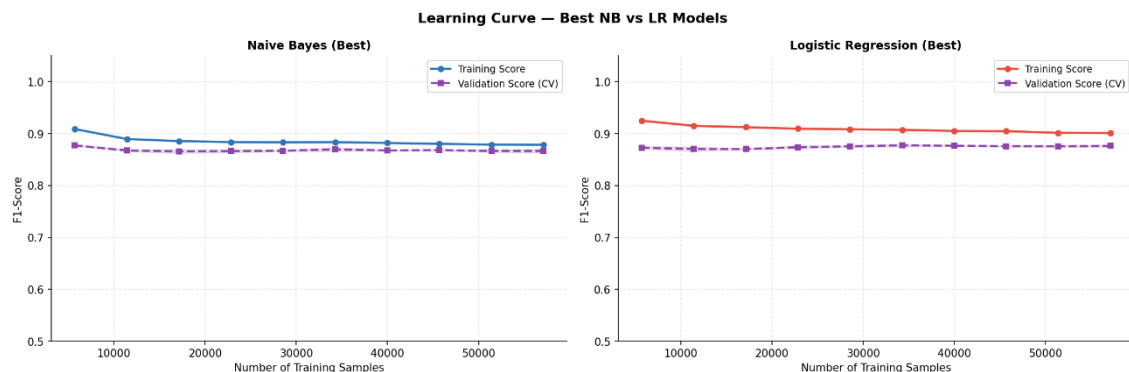


Figure 20. Learning curve best naive bayes and logistic regression models

CONCLUSION

This study successfully demonstrated that the joint selection of text preprocessing configuration, feature extraction method, and classification model significantly influences sentiment analysis performance on Indonesian-language DANA application reviews, as statistically confirmed by the Kruskal-Wallis test ($p=0.0274$). Among all 24 systematically executed full factorial experimental combinations, the optimal pipeline was identified as no additional preprocessing, N-gram as the feature extraction method, and Logistic Regression as the classifier, achieving the highest F1-score of 0.8772 and surpassing the baseline by 1.82%. Logistic Regression was statistically proven superior to Naive Bayes across all configurations based on the Mann-Whitney U test ($p=0.0001$), while N-gram significantly outperformed unigram TF-IDF in negative sentiment classification based on the Wilcoxon Signed-Rank test ($p=0.0039$), owing to its capacity to capture phrase context and negation patterns in Indonesian text. A further critical finding was the absence of a significant interaction effect between preprocessing and feature extraction based on the Scheirer-Ray-Hare test ($p=0.9991$), implying that both factors can be independently optimized in sentiment analysis system development. This research contributes a systematic full factorial evaluation framework that can be adopted by researchers and practitioners as evidence-based guidance for selecting optimal

sentiment analysis pipelines for Indonesian-language fintech applications. Nevertheless, this study acknowledges several limitations that warrant consideration: the dataset was sourced exclusively from a single platform (Google Play Store), limiting the generalizability of results to other platforms such as the App Store; sentiment labeling was performed automatically based on star ratings without manual verification, potentially introducing inaccurate labels for ambiguous reviews; the experiments were restricted to two traditional machine learning classifiers, thus not reflecting the potential of deep learning or transformer-based models; and the N-gram implementation was limited to bigrams ($n=1-2$), leaving trigrams and higher-order n-gram combinations unexplored. For future research, it is recommended to explore transformer-based models such as IndoBERT or RoBERTa pretrained on Indonesian corpora, integrate data augmentation techniques to address more extreme class imbalances, extend the dataset to include reviews from additional platforms such as the App Store, and apply manual or semi-automatic sentiment labeling to improve data quality and model generalizability.

REFERENCES

- [1] I. S. Widiyanto, Y. R. Ramadhan, and M. A. Komara, "Analisis Sentimen E-Wallet Gopay, ShopeePay, dan OVO Menggunakan Algoritma Naive Bayes," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 3S1, 2024, doi: 10.23960/jitet.v12i3S1.5277.
- [2] D. F. Nawulansih, N. C. Santi, and I. A. Sa'ida, "Analisis Sentimen Ulasan Aplikasi DANA di Google Play Store: Penerapan Support Vector Machine dan Synthetic Minority Over-sampling Technique," *J. Pendidik. dan Teknol. Indones.*, vol. 5, no. 9, pp. 2660–2671, 2025, doi: 10.52436/1.jpti.1053.
- [3] A. L. Fuadi, A. H. Yunial, A. Hidayat, A. Napila, and A. Fauzi, "Analisis Sentimen pada Ulasan Aplikasi Fintech di Indonesia: Studi Komparatif Model Machine Learning dan Deep Learning," *J. Innov. Creat.*, vol. 6, no. 1, pp. 1858–1874, 2026, doi: 10.31004/joecy.v6i1.6939.
- [4] A. Bachtiar, M. J. Hawali, N. C. Simbolon, D. A. Maulana, and R. A. Anggraini, "Analisis Sentimen Ulasan Free Fire Menggunakan Naive Bayes Logistic Regression," *J. Inf. Syst. Manag.*, vol. 7, no. 2, 2026, doi: 10.24076/joism.2026v7i2.2433.
- [5] M. R. Syaputra, M. Arifin, and D. L. Fithri, "Klasifikasi Sentimen Ulasan E-Wallet Menggunakan TF-IDF dan Random Forest dengan Penyeimbangan Data SMOTE," *Edumatic J. Pendidik. Inform.*, vol. 9, no. 2, pp. 532–541, 2025, doi: 10.29408/edumatic.v9i2.31084.
- [6] A. Fitriyanto and P. Purwanto, "Analisis Sentimen Ulasan DANA dari Play Store dengan Metode SVM, Logistic Regression, Naive Bayes dan KNN," *Build. Informatics, Technol. Sci.*, vol. 7, no. 3, pp. 1887–1899, 2025, doi: 10.47065/bits.v7i3.8769.
- [7] G. R. Ramadhan and C. A. Sugianto, "Analisis Sentimen Ulasan Aplikasi DANA di Google Play Store Menggunakan Algoritma Naive Bayes," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 8, no. 5, pp. 9849–9857, 2024, doi: 10.36040/jati.v8i5.10732.
- [8] A. N. Sianipar, Yuhelmi, and M. Devega, "Analisis Sentimen Ulasan Pengguna Aplikasi Dana dan LinkAja Menggunakan Metode Naive Bayes," *Zo. J. Sist. Inf.*, vol. 6, no. 2, pp. 510–522, 2024, doi: 10.31849/zn.v6i2.19907.
- [9] D. Munandar, M. Afdal, Zarnelly, and R. Novita, "Analisis Sentimen Ulasan Pengguna Aplikasi Mobile Banking Menggunakan Algoritma K-Nearest Neighbor," *J. Teknol. Sist. Inf. dan Apl.*, vol. 7, no. 3, pp. 1309–1318, 2024, doi: 10.32493/jtsi.v7i3.41409.
- [10] A. Nurian, "Analisis Sentimen Ulasan Pengguna Aplikasi Google Play Menggunakan Naive Bayes," *J. Inform. dan Tek. Elektro Terap.*, vol. 11, no. 3S1, pp. 829–835, 2023, doi: 10.23960/jitet.v11i3s1.3348.
- [11] A. Gerliandeva, Y. H. Chrisnanto, and H. Ashaury, "Optimasi Klasifikasi Sentimen pada Komentar Online Menggunakan Multinomial Naive Bayes dan Ekstraksi Fitur TF-IDF serta N-Grams," *J. Pekommas*, vol. 9, no. 2, pp. 259–272, 2024, doi: 10.56873/jpkm.v9i2.5585.
- [12] C. Dometian and S. Jatmiko, "Analisis Sentimen Ulasan Google Play Store: Studi Komparatif Algoritma SVM, Naive Bayes, dan Logistic Regression," *J. FASILKOM (teknologi Inf. dan Ilmu KOMputer)*, vol. 15, no. 3, pp. 610–620, 2025, doi: 10.37859/jf.v15i3.10016.
- [13] N. A. Hapsari and A. D. Indriyanti, "Analisis Sentimen pada Aplikasi Dompot Digital Menggunakan Algoritma Random Forest," *J. Emerg. Inf. Syst. Bus. Intell.*, vol. 4, no. 3, pp. 186–192, 2023, doi: 10.26740/jeisbi.v4i3.55696.
- [14] A. I. Tanggraeni and M. N. N. Sitokdana, "Analisis Sentimen Aplikasi E-Government pada Google Play Menggunakan Algoritma Naive Bayes," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 2, pp. 785–795, 2022, doi: 10.35957/jatisi.v9i2.1835.
- [15] R. Maulana, A. Voutama, and T. Ridwan, "Analisis Sentimen Ulasan Aplikasi MyPertamina pada

- Google Play Store Menggunakan Algoritma NBC,” *J. Teknol. Terpadu*, vol. 9, no. 1, pp. 42–48, 2023, doi: 10.54914/jtt.v9i1.609.
- [16] H. H. Mubaroroh, H. Yasin, and A. Rusgiyono, “Analisis Sentimen Data Ulasan Aplikasi Ruangguru pada Situs Google Play Menggunakan Algoritma Naïve Bayes Classifier dengan Normalisasi Kata Levenshtein Distance,” *J. Gaussian*, vol. 11, no. 2, pp. 248–257, 2022, doi: 10.14710/j.gauss.v11i2.35472.
 - [17] S. J. Angelina, A. Bijaksana, P. Negara, and H. Muhandi, “Analisis Pengaruh Penerapan Stopword Removal pada Performa Klasifikasi Sentimen Tweet Bahasa Indonesia,” *JUARA J. Apl. dan Ris. Inform.*, vol. 2, no. 1, pp. 165–173, 2023, doi: 10.26418/juara.v2i1.69680.
 - [18] A. Bijaksana and P. Negara, “The Influence of Applying Stopword Removal and SMOTE on Indonesian Sentiment Classification,” *Lontar Komput. J. Ilm. Teknol. Inf.*, vol. 14, no. 3, pp. 172–185, 2023, doi: 10.24843/lkjiti.2023.v14.i03.p05.
 - [19] R. Ulgasesa, A. Bijaksana, and P. Negara, “Pengaruh Stemming terhadap Performa Klasifikasi Sentimen Masyarakat tentang Kebijakan New Normal,” *JUSTIN J. Sist. dan Teknol. Inf.*, vol. 10, no. 3, 2022, doi: 10.26418/justin.v10i3.53880.
 - [20] A. Santosa, I. Purnamasari, and R. Mayasari, “Pengaruh Stopword Removal dan Stemming terhadap Performa Klasifikasi Teks Komentar Kebijakan New Normal Menggunakan Algoritma LSTM,” *J-SAKTI (Jurnal Sains Komput. dan Inform.*, vol. 6, no. 1, pp. 81–93, 2022, doi: 10.30645/j-sakti.v6i1.427.
 - [21] R. F. P. Ichsansyah and Subektiningsih, “Perbandingan Algoritma Naïve Bayes dan Random Forest dalam Klasifikasi Sentimen Ulasan Pengguna Kredivo pada Play Store,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 13, no. 2, pp. 297–308, 2026, doi: 10.25126/jtiik.132.
 - [22] and R. R. A. D. M. Putri, N. Sulistianingsih, “Pengaruh Teknik Representasi Teks Bag-of-Words dan TF-IDF terhadap Akurasi Klasifikasi Sentimen Teks Multi-Domain,” *JTIM J. Teknol. Inf. dan Multimed.*, vol. 7, no. 4, pp. 675–688, 2025, doi: 10.35746/jtim.v7i4.756.
 - [23] M. S. Al Markas, S. Anraeni, and L. B. Ilmuwan, “Implementasi Fitur Vector Bag of Word dan TF-IDF untuk Analisis Sentiment,” *LINIER Lit. Inform. dan Komput.*, vol. 2, no. 2, pp. 136–146, 2025, doi: 10.33096/linier.v2i2.3104.
 - [24] R. Wati, S. Ernawati, and H. Rachmi, “Pembobotan TF-IDF Menggunakan Naïve Bayes pada Sentimen Masyarakat Mengenai Isu Kenaikan BIPIH,” *J. Manaj. Inform.*, vol. 13, no. 1, pp. 84–93, 2023, doi: 10.34010/jamika.v13i1.9424.
 - [25] and I. S. W. Iriananda, R. W. Budiawan, A. Y. Rahman, “Optimasi Klasifikasi Sentimen Komentar Pengguna Game Bergerak Menggunakan SVM, Grid Search dan Kombinasi N-Gram,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 11, no. 4, pp. 743–752, 2024, doi: 10.25126/jtiik.1148244.
 - [26] and D. L. P. D. D. T. Situngkir, Anita, “Klasifikasi Sentimen Ulasan Gofood di Google Play Store dengan Metode Naïve Bayes,” *J. Algoritm.*, vol. 23, no. 1, pp. 1589–1600, 2026, doi: 10.33364/algoritma/v.23-1.3479.
 - [27] M. M. Rachmatullah, B. Irawan, A. Faqih, D. Pratama, and D. A. Kurnia, “Analisis Komparatif Multinomial Naïve Bayes dan Logistic Regression untuk Klasifikasi Sentimen Ulasan Pengguna Aplikasi TIX ID,” *JSI (Jurnal Sist. Informasi) Univ. Suryadarma*, vol. 13, no. 1, pp. 62–70, 2026, doi: 10.35968/jsi.v13i1.1723.
 - [28] R. P. and I. A. G. S. Putra, “Analisis Sentimen Ulasan Aplikasi dengan Multinomial Naïve Bayes, Logistic Regression, dan SVM,” *urnal Nas. Teknol. Inf. dan Apl.*, vol. 4, no. 1, pp. 63–72, 2025, doi: 10.24843/JNATIA.2025.v04.i01.p08.
 - [29] Y. A. Prasetyo, E. Utami, and A. Yaqin, “Pengaruh Komposisi Split Data terhadap Performa Akurasi Analisis Sentimen Algoritma Naïve Bayes dan SVM,” *J. Elektro dan Mesin Terap.*, vol. 6, no. 2, pp. 382–390, 2024, doi: 10.33650/jeeecom.v4i2.
 - [30] D. A. Tarigan, Z. Situmorang, and R. Rosnelly, “Analisis Sentimen Aplikasi Playstore Sirekap 2024 Pasca Pilpres dengan Perbandingan Metode Support Vector Machine (SVM), Naïve Bayes Classifier, dan Random Forest,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 12, no. 3, pp. 661–670, 2025, doi: 10.25126/jtiik.2025129608.
 - [31] D. Chicco, A. Sichenze, and G. Jurman, *A Simple Guide to the Use of Student's T-Test, Mann-Whitney U Test, Chi-Squared Test, and Kruskal-Wallis Test in Biostatistics*. London: BioMed Central, 2025. doi: 10.1186/s13040-025-00465-6.
 - [32] and A. P.-G. A. Garrocho-Rangel, S. Aranda-Romo, R. Martínez-Martínez, V. Zavala-Alonso, J. C. Flores-Arriaga, “Fundamentals of Nonparametric Statistical Tests for Dental Clinical Research,” *Dent. J.*, vol. 12, no. 10, p. 314, 2024, doi: 10.3390/dj12100314.
 - [33] L. A. Elkin, M. Kay, J. J. Higgins, and J. O. Wobbrock, *An Aligned Rank Transform Procedure for*

- Multifactor Contrast Tests*, vol. 1, no. 1. Association for Computing Machinery, 2021.
- [34] M. Abid, M. Sun, A. Shabbir, M. E. Bakr, and T. Abbas, "An enhanced nonparametric quality control chart with application related to industrial process," *Sci. Rep.*, vol. 14, pp. 1–15, 2024, doi: 10.1038/s41598-024-64084-7.
 - [35] R. W. Emerson, "Mann-Whitney U Test and T-Test," *J. Vis. Impair. Blind.*, vol. 117, no. 1, pp. 99–100, 2023, doi: 10.1177/0145482x221150592.
 - [36] S. F. Taskiran, B. Turkoglu, E. Kaya, and T. Asuroglu, "A Comprehensive Evaluation of Oversampling Techniques for Enhancing Text Classification Performance," *Sci. Rep.*, vol. 15, no. 1, p. 21631, 2025, doi: 10.1038/s41598-025-05791-7.
 - [37] A. Aryanusa, "Evaluasi Kinerja Logistic Regression dan Multinomial Naïve Bayes dalam Klasifikasi Sentimen Mobile Banking," *J. Inform. dan Tek. Elektro Terap.*, vol. 13, no. 3S1, pp. 463–474, 2025, doi: 10.23960/jitet.v13i3S1.7687.
 - [38] N. A. Semary, W. Ahmed, K. Amin, and P. P. Id, "Enhancing Machine Learning-Based Sentiment Analysis Through Feature Extraction Techniques," *PLoS One*, vol. 19, no. 2, p. e0294968, 2024, doi: 10.1371/journal.pone.0294968.
 - [39] M. Mylarappa, S. K. B. N, T. J. Gowda, and V. K. Rajuk, "Enhanced Negation Handling for Sentiment Analysis on Twitter Using Deep Neural Networks," *IJECS Indones. J. Electr. Eng. Comput. Sci.*, vol. 32, no. 3, pp. 1736–1745, 2023, doi: 10.11591/ijeecs.v32.i3.pp1736-1745.
 - [40] A. W. Pradana and M. Hayaty, "The Effect of Stemming and Removal of Stopwords on the Accuracy of Sentiment Analysis on Indonesian-language Texts," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 3, pp. 375–380, 2019, doi: 10.22219/kinetik.v4i4.912.
 - [41] and M. I. M. D. Y. Saraswati, M. R. Handayani, K. Umam, "Sentiment Classification of MyPertamina Reviews Using Naïve Bayes and Logistic Regression," *J. Appl. Informatics Comput.*, vol. 9, no. 4, pp. 1319–1325, 2025, doi: 10.47065/jaic.v9i4.9723.
 - [42] M. Monica and A. Purwanto, "Comparative Performance Evaluation of Naïve Bayes and Logistic Regression for Indonesian YouTube Comment Sentiment Classification," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 6, no. 2, pp. 934–943, 2026, doi: 10.57152/malcom.v6i2.2639.
 - [43] M. A. Palomino and F. Aider, "Evaluating the Effectiveness of Text Pre-Processing in Sentiment Analysis," *Appl. Sci.*, vol. 12, no. 17, p. 8765, 2022, doi: 10.3390/app12178765.
 - [44] M. Hadwan, M. Al-Sarem, and F. Saeed, "An Improved Sentiment Classification Approach for Measuring User Satisfaction Toward Governmental Services' Mobile Apps Using Machine Learning Methods with Feature Engineering and SMOTE Technique," *Appl. Sci.*, vol. 12, no. 11, p. 5547, 2022, doi: 10.3390/app12115547.
 - [45] N. Latifah, D. H. E. Putra, and F. Nugraha, "Time-Series Monitoring of Sentiment Dynamics in Reviews of Four Indonesian E-Wallet Applications Using a Hybrid TF-IDF and Bi-LSTM Framework," *J. Inf. Syst. Informatics*, vol. 8, no. 2, pp. 1958–1983, 2026, doi: 10.63158/journalisi.v8i2.1488.
 - [46] D. E. Cahyani and I. Patasik, "Performance Comparison of TF-IDF and Word2Vec Models for Emotion Text Classification," *Bull. Electr. Eng. Informatics*, vol. 10, no. 5, pp. 2780–2788, 2021, doi: 10.11591/eei.v10i5.3157.
 - [47] B. Setiawan, "A Review of Sentiment Analysis Applications in Indonesia Between 2023-2024," *J. Inf. Eng. Educ. Technol.*, vol. 8, no. 2, pp. 71–83, 2025, doi: 10.26740/jieet.v8n2.p71-83.