MJI
Media Jurnal Informatika
Vol. 17, No. 2, December 2025
p-issn : 2088-2114, e-issn : 2477-2542

Evaluating Machine Learning Models Across Feature Extraction and Data Balancing Scenarios for Coretax Sentiment Analysis

Subhan Syah Putra [a,1], Desti Riminarsih [b,2,*]

[a,b]Program Studi Informatika, Fakultas Teknologi Industri, Universitas Gunadarma, Jl. Margonda Raya No. 100, Depok, Jawa Barat, 16424
[1] subhansyahsp@gmail.com; [2] destimath@staff.gunadarma.ac.id*
* corresponding author

ARTICLE INFO

Article history
Received 2025/12/09
Revised 2025/12/24
Accepted 2025/12/28

Keywords
Coretax
Data Balancing
Feature Extraction
Machine Learning
Sentiment Analysis

ABSTRACT

The implementation of the Core Tax Administration System (Coretax) by the Indonesian Directorate General of Taxes has generated diverse public responses on social media, particularly on platform X, making sentiment analysis a relevant approach to assess public perception of this policy. This study aims to evaluate the performance of machine learning classifiers across different feature extraction and data balancing scenarios. Three machine learning classifiers, namely Multinomial Naïve Bayes, Support Vector Machine (SVM), and Logistic Regression were evaluated under four experimental scenarios combining two feature extraction methods, namely Term Frequency–Inverse Document Frequency (TF-IDF) and Bag of Words (BoW), with original and balanced data distributions. A dataset of more than 50,000 Coretax-related posts collected from platform X was preprocessed and automatically labeled into positive, negative, and neutral sentiment classes using a pretrained IndoBERT sentiment model. A brief manual inspection of a random subset indicates moderate agreement between automatic and manual labels, highlighting potential noise while supporting the use of automatic labeling for comparative analysis. The results show that performance is shaped by the combined effects of representation and data distribution rather than algorithm choice alone. Logistic Regression consistently achieved the most stable and competitive performance across all scenarios, with accuracy values ranging from approximately 0.80 to 0.83 and macro F1-scores around 0.72–0.73. TF-IDF generally provided more stable performance, while data balancing improved prediction fairness for minority sentiment classes despite a slight decrease in overall accuracy. These findings demonstrate that Logistic Regression is the most robust model for Coretax sentiment analysis across varying feature extraction and data balancing conditions and provide practical insights into the influence of data representation and distribution on sentiment classification performance.

1. Introduction

Taxes constitute a fundamental source of governmental revenue utilized to finance different sectors of growth, including education, healthcare, and infrastructure[1]. Effective tax management substantially enhances public welfare while concurrently stimulating national economic growth. Nonetheless, Indonesia's tax system has certain obstacles, such as inadequate taxpayer compliance, administrative intricacy, and insufficient public comprehension of legislation. The intricate reporting and payment procedures frequently deter individuals from actively participating in the tax system[2].

Regulated by Presidential Regulation Number 40 of 2018, the Directorate General of Taxes (DGT) initiated the Core Tax Administration System (Coretax) as part of the Core Tax Administration System Renewal

Project (Pembaruan Sistem Inti Administrasi Perpajakan (PSIAP))[3]. The goal of Coretax is to improve accountability, effectiveness, and transparency by combining services such as registration, reporting, payment, and verification into a single digital portal [4]. The public will respond to the full implementation of Coretax starting January 1, 2025. It is very important to evaluate user experience and public perception of this system due to the large number of complaints regarding the interface and system response.

Social media has become a constantly evolving representation of public opinion, where individuals freely express their views in real time. Platforms such as X, formerly known as Twitter, enable users to share opinions and experiences that generate a large volume of textual data suitable for analysis using sentiment analysis methods. Previous studies have demonstrated the effectiveness of Twitter-based data in capturing public opinions, including research that analyzed public perceptions of KRL Commuterline services based on user comments on Twitter [5]. Building on this evidence, evaluating public perception of government policies, including the implementation of the Core Tax Administration System (Coretax), can be effectively conducted by utilizing social media platforms as a data source.

Research has compared the performance of classification algorithms for social media sentiment analysis. The Bernoulli Naïve Bayes variants (Bernoulli, Multinomial, and Gaussian) were tested on tweet data related to interest in paying taxes, and Bernoulli Naïve Bayes with SMOTE showed an accuracy of 91.03% [4]. An analysis of public opinion regarding the discourse on value-added tax (Pajak Pertambahan Nilai (PPN)) for staple food commodities and educational services using Social Network Analysis and the Naïve Bayes classifier achieved an accuracy of 74.87%, although the results tended to be negatively biased due to imbalanced data distribution [6]. A comparative study involving Naïve Bayes, SVM, and K-Nearest Neighbor (K-NN) for sentiment classification of user reviews of a vehicle tax application demonstrated that SVM outperformed the other algorithms with an accuracy of 76.5%, followed by Naïve Bayes at 72.3% and K-NN at 59.1% [7]. Another study on sentiment analysis related to the discourse on the relocation of Indonesia's capital city evaluated four algorithms—Naïve Bayes, Logistic Regression, SVM, and K-NN—and again confirmed the dominance of SVM, achieving an accuracy of 97.72% [8]. Overall, previous studies highlight the strong performance of SVM in sentiment analysis tasks, while Naïve Bayes remains widely used due to its simplicity and computational efficiency. Logistic Regression has also shown consistently competitive performance and is frequently employed as a benchmark model in text classification studies.

Although previous studies have reported competitive accuracy in sentiment analysis of public policy discourse, most evaluations were conducted under a single experimental configuration. In particular, earlier works typically relied on one feature extraction method and did not systematically account for class imbalance, causing accuracy results to be dominated by the majority class and limiting insight into minority sentiment detection. Moreover, algorithm comparisons rarely considered variations in both feature representation and data distribution. To address this limitation, this study systematically evaluates multiple experimental scenarios by combining different feature extraction techniques (TF-IDF and Bag of Words) with original and balanced data distributions, thereby emphasizing the novelty of the proposed research objective.

There is not much research available on the context of Coretax. Using the Naive Bayes algorithm, recent research investigated public perception of Coretax on the X platform, showing an accuracy of approximately 77% [9]. The model is better for negative and neutral sentiment, but due to the imbalanced data distribution, it's difficult to find positive sentiment. Although the model demonstrated better performance in identifying negative and neutral sentiments, it encountered difficulties in recognizing positive sentiment due to imbalanced data distribution. While these findings provide an important initial insight into public sentiment toward Coretax, the study was restricted to a single classification algorithm and did not consider comparisons with other widely used and more robust machine learning methods. Furthermore, the impact of different feature extraction techniques and data balancing strategies on classification performance was not examined. This limitation highlights a clear research gap and underscores the need for a more comprehensive evaluation that systematically compares multiple machine learning models under various feature extraction and data balancing scenarios to obtain a more robust and fair assessment of Coretax sentiment analysis.

Based on these differences, this study will focus on evaluating and comparing three text classification algorithms commonly used in sentiment analysis: Multinomial Naive Bayes, SVM, and Logistic Regression. These algorithms were chosen because they have advantages in terms of efficiency, accuracy, and performance stability. In addition to comparing algorithms, this study also examines how variations in feature representation, data distribution, and hyperparameter settings impact different modeling scenarios. As a result, this research is expected to provide a more comprehensive picture of how effective sentiment classification methods are in assessing public opinion about Coretax on the X platform.

2. Method

This study aims to evaluate the performance of several machine learning algorithms for sentiment analysis on the Coretax topic using data from the social media platform X. The research workflow was conducted in a structured manner, starting from data collection, exploratory data analysis, text preprocessing, automatic labeling, dataset splitting, feature extraction, data balancing, to the training and evaluation of classification models. All research stages were implemented using the Python programming language with the support of various libraries, as illustrated in Fig 1.

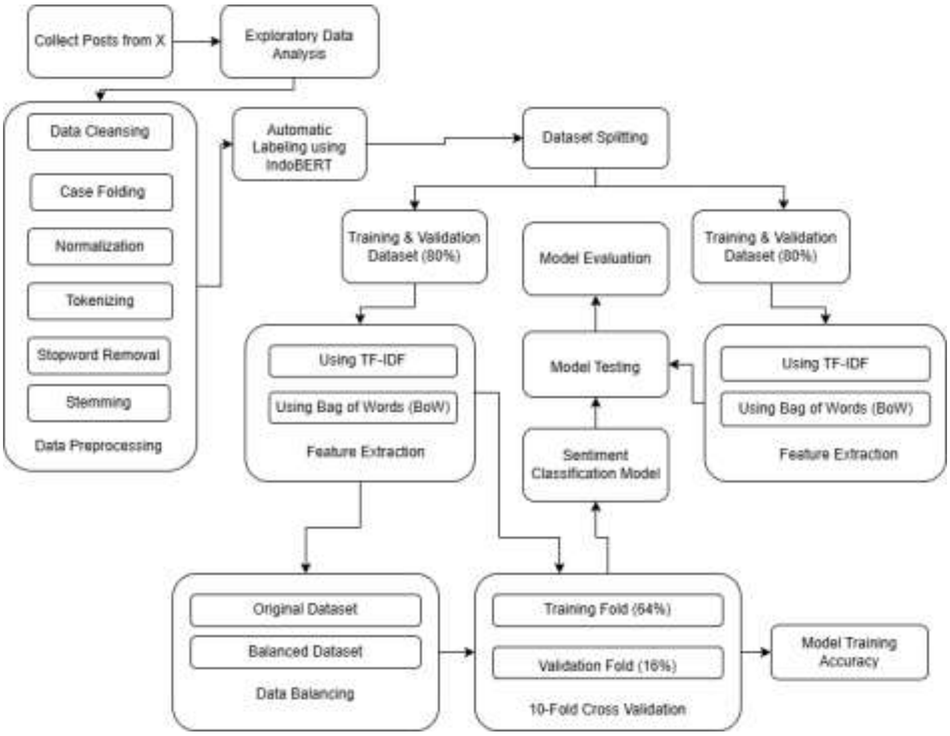

Fig 1. Research Flowchart

2.1 Data Collection

The data used in this study was collected from the social media platform X, which was chosen because it is one of the most widely used social media platforms in public discussions [10]. Web scraping techniques, an automated process for extracting data from websites commonly used in social media-based research, were used to collect this data [11]. The scraping process was implemented using the Twikit library in the Python programming language. The search keywords used were “coretax” and “#coretax” over the period January–April 2025. The collected data were subsequently stored in Comma Separated Values (CSV) format to facilitate further processing.

2.2 Exploratory Data Analysis

At this stage, data exploration analysis is performed to gain an understanding of the initial characteristics of the dataset, such as patterns, trends, and anomalies. This will be useful for determining further preprocessing strategies and the design of machine learning pipelines [12]. The analysis was performed using the Python collections, matplotlib, and pandas libraries. Some of the elements evaluated include text length (number of words per entry), the identification of the most frequent words in the raw data, and the use of non-standard words or abbreviations based on a normalization dictionary (slang dictionary). Additionally, the proportion of data containing non-text elements such as URLs, mentions (@username), and hashtags was calculated. This also checks for duplicate data, text that is too short (less than 2 words) or too long (more than 60 words), data with a word uniqueness ratio less than 0.4, and specific account auto-reply patterns.

2.3 Data Preprocessing

An important stage in the text processing pipeline is the text preprocessing stage, which allows for the transformation of unstructured data into a more consistent and analysis-ready format [13]. This stage is crucial in sentiment analysis as it involves preparing the text for numerical representation thru feature extraction methods [14]. Cleaning (removing URLs, mentions, hashtags, and punctuation), case folding, slang normalization, space-based tokenization, stopword removal, and stemming using Sastrawi are some of the processes used on Indonesian-language data[14]. Additionally, extra filters are used to remove duplicate data, repetitive entries, text that is too short or too long, and auto-reply patterns.

In addition, several filtering rules were applied to improve data quality, including the removal of duplicate and repetitive entries, auto-reply patterns, excessively short or long texts, and texts with a low word uniqueness ratio. The exclusion of very short texts helps reduce noise from ambiguous or context-poor posts that often lack sufficient semantic information for reliable sentiment classification. Meanwhile, the uniqueness ratio filter aims to eliminate repetitive or spam-like content that may bias the model toward dominant sentiment patterns. Although these filtering steps may reduce the number of minority-class samples, they are intended to preserve meaningful contextual sentiment signals and improve overall model robustness by ensuring that retained texts contain sufficient linguistic information for effective classification.

2.4 Data Labeling

All preprocessed data were assigned sentiment labels using a pretrained IndoBERT-Sentiment model through the HuggingFace Transformers library. This model was fine-tuned from IndoBERT Base P1 on the SmSA dataset from IndoNLU [15], [16] , achieving a validation accuracy of 94.52% [17]. The sentiment labels were categorized into three classes: positive (appreciation/support), negative (criticism/dissatisfaction), and neutral (informational content without polarity). As a general-purpose model, it may exhibit limitations in capturing sentiment nuances specific to the Coretax context.

To assess labeling reliability, a random subset of 200 tweets was manually inspected and compared with the IndoBERT-generated labels. The agreement rate was 75.5%, indicating a moderate level of consistency. This result reflects the inherent ambiguity and noise of social media text, particularly for sentiment classification involving sarcasm and implicit opinions. Therefore, the automatically generated labels are treated as a silver standard and used consistently across all experimental scenarios.

2.5 Data Splitting

The dataset was split using an 80% training and 20% testing ratio, following common practices in machine learning research [18]. The splitting was performed in a stratified manner to preserve the proportion of each sentiment class across both subsets. The training data were used exclusively for model training and validation, while the testing data were held out and used only once during the final evaluation stage. Consequently, the evaluation results provide an unbiased estimate of the model's ability to generalize to previously unseen data.

2.6 Feature Extraction

The feature extraction stage was conducted to transform textual data into numerical representations suitable for classification algorithms. Two approaches were employed: Term Frequency–Inverse Document Frequency (TF–IDF) and Bag of Words (BoW), both of which are widely used in sentiment analysis. TF–IDF assigns higher weights to words that frequently appear in a document but rarely occur across other documents, making it more informative for highlighting important terms. In contrast, BoW represents documents as vectors of word frequencies from the existing vocabulary, without considering word order [19].

In this study, both methods were applied using unigram and bigram representations, allowing the models to capture not only single-word semantics but also two-word combinations that are more representative in sentiment analysis. The extracted features were subsequently used as inputs for the model training pipeline described in the following subsection.

2.7 Data Balancing

Class imbalance is a common issue in classification tasks, where certain classes contain significantly more samples than others. This condition may cause models to be biased toward the dominant class and degrade prediction performance on minority classes [14]. Therefore, data balancing strategies are required to reduce the risk of overfitting or underfitting due to disproportionate class distributions.

To ensure the evaluation results reflect the actual conditions, balancing is only applied to the training data. Test data, on the other hand, remains in its original distribution. This method uses oversampling for the minority class and undersampling for the majority class, with the inter-class distribution adjusted to the median value. Therefore, the training data becomes more balanced, allowing the model to learn more representatively without reducing the validity of the evaluation. Data balancing was applied only to the training set, since balancing the test data would alter the natural class distribution and potentially produce overly optimistic performance estimates that do not reflect real-world conditions.

2.8 Model Training

This study employed three classification algorithms, namely Logistic Regression, Multinomial Naive Bayes, and SVM (Linear SVM). Multinomial Naive Bayes is an algorithm based on Bayes' Theorem with the assumption of feature independence, and it is simple and effective for large datasets [14]. To separate classes with maximum margin, SVM searches for the ideal hyperplane [14]. Logistic Regression models the probability of class membership using a sigmoid function over a linear combination of features, which has the advantages of being simple and interpretable [20].

For validation, 10-Fold Stratified Cross Validation was employed, where the training data were divided into ten balanced subsets; in each iteration, one subset was used as validation data and the remaining nine as training data. This approach preserves class distribution and enhances the reliability of performance estimation [21]. Experimental evaluations were designed under four scenarios combining feature extraction methods and data balancing strategies, as presented in Table 1.

Table 1. Model Training Scenarios

Scenario	Feature Extraction Method	Data Balancing
S1	TF–IDF (unigram + bigram)	No
S2	TF–IDF (unigram + bigram)	Yes
S3	BoW (unigram + bigram)	No
S4	BoW (unigram + bigram)	Yes

The model training pipeline is built using the scikit-learn and imbalanced-learn libraries. The pipeline components are arranged sequentially as follows:

1. feature extraction using TF-IDF Vectorizer or CountVectorizer;
2. data balancing using a combination strategy of RandomOverSampler and RandomUnderSampler to adjust class distribution to the median (only in balanced scenarios);
3. classification using MultinomialNB, LinearSVC, and LogisticRegression.

Performance evaluation for each fold was conducted using accuracy as the primary metric, with additional analysis of result distribution across folds visualized thru boxplots to assess performance stability. In addition to basic experiments, limited hyperparameter tuning was performed thru a simple grid search. Adjustable parameters include α for Multinomial Naïve Bayes, the value of C and class_weight for Linear SVM, and combinations of C, class_weight, and solver values for Logistic Regression. The candidate range is focused on small to medium values commonly used in text-based sentiment analysis to keep experiments efficient while still sufficiently differentiating the baseline model from the tuned variants.

Table 2. Hyperparameter Configurations Evaluated for Each Classification Model

Model	Hyperparameter	Evaluated Value
Multinomial Naïve Bayes	α(smoothing)	(baseline, 0.1, 0.5)
	prior	(True, False)
SVM	C	(baseline, 1.5, 2.0)
	class_weight	(None, Balanced)
Logistic Regression	C	(baseline, 1.5, 2.0)
	class_weight	(None, Balanced)
	solver	(lbfgs, liblinear)

Table 2 summarizes the hyperparameter configurations evaluated for each classification model across all experimental scenarios. Only parameters that were explicitly tested and reported in the experimental results are included to ensure full reproducibility and transparency. Model performance during cross-validation was primarily assessed using accuracy, with the distribution of results across folds visualized using boxplots to evaluate model stability. After the training and validation stage, the best-performing configuration from each scenario was selected and further evaluated on the held-out test set, as described in the following subsection.

2.9 Model Testing

After the cross-validation stage, the model was tested with twenty percent test data, which was completely different from the training process. The distribution of model predictions against actual classes is illustrated thru a confusion matrix for evaluation [22]. Four main metrics are calculated from this matrix: accuracy, precision, recall, and F1 score. Accuracy measures the proportion of correct predictions out of all the data, precision assesses the accuracy of positive predictions, and recall is the harmonic mean score between Precision and Recall to maintain a balance between the two. To ensure that each class has the same evaluation value, all metrics are calculated in macro form. This is done to ensure that the model's performance can be reviewed fairly across uneven class distributions.

3. Results and Discussion

The initial dataset was obtained by scraping the social media platform X, resulting in 73,797 entries related to Coretax. The results of the initial exploratory data analysis revealed several key characteristics of the dataset.

a. the text length ranged from 1 to 59 words, with an average length of 25.97 words per entry
b. the term *"coretax"* appeared most frequently, with 59,732 occurrences, followed by conjunctions and commonly used greeting words.
c. approximately 75.8% of the entries contained user mentions, 19.0% included URLs, and 5.1% contained hashtags.
d. the high variation in informal language and the frequent repetition of words highlighted the necessity of applying comprehensive text preprocessing before further analysis.

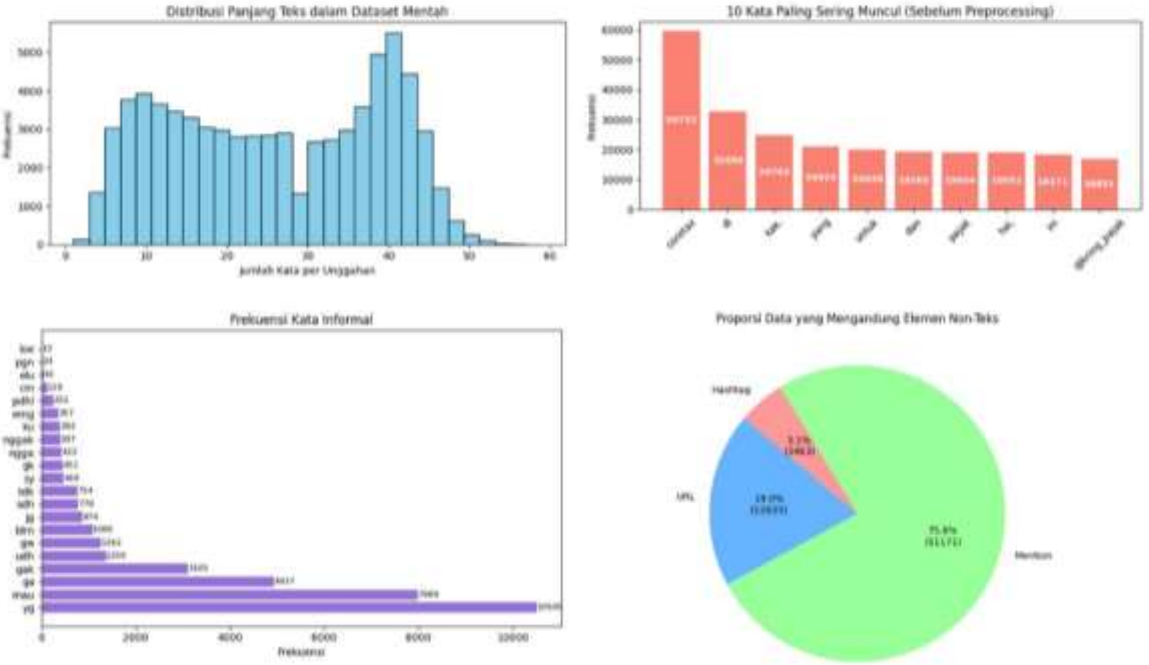

Fig 2. Results of Exploratory Data Analysis

Figure 2 presents the results of the exploratory data analysis conducted on the raw Coretax-related dataset collected from platform X. The first subplot illustrates the distribution of text length, showing that the number of words per post varies widely, with most entries concentrated between short to medium-length texts, which is typical of social media content. The second subplot displays the ten most frequently occurring words before preprocessing, where the term *"coretax"* dominates the corpus, followed by commonly used function words and general expressions, indicating a strong topical focus on Coretax-related discussions. The third subplot highlights the frequency of informal words and slang expressions, reflecting the conversational and non-formal nature of user-generated content on social media. This characteristic emphasizes the necessity of normalization and text preprocessing to reduce noise in the dataset. The fourth subplot shows the proportion of non-textual elements, revealing that a large portion of the posts contain user mentions, URLs, and hashtags. The high presence of these elements further justifies the need for comprehensive text preprocessing prior to feature extraction and model training. Overall, the exploratory analysis provides an initial understanding of the dataset characteristics and serves as the basis for designing the subsequent preprocessing and sentiment classification pipeline. After preprocessing, the dataset was reduced to a net 52,095 entries, ready for labeling and model training.

3.1 Data Labelling Result

A total of 52,095 entries were successfully labeled automatically. The label distribution shows class imbalance (Fig 3.): negative 26,088, neutral 22,870, and positive 3,137. This imbalance underscores the need for data balancing strategies during the training phase.

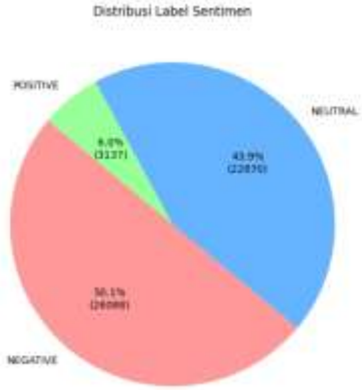

Fig 3. Sentiment Label Distribusions

3.2 Feature Representation & Data Balancing

The label distribution in the training and test sets remained consistent after the stratified split (Fig 4.), although class imbalance was still evident.

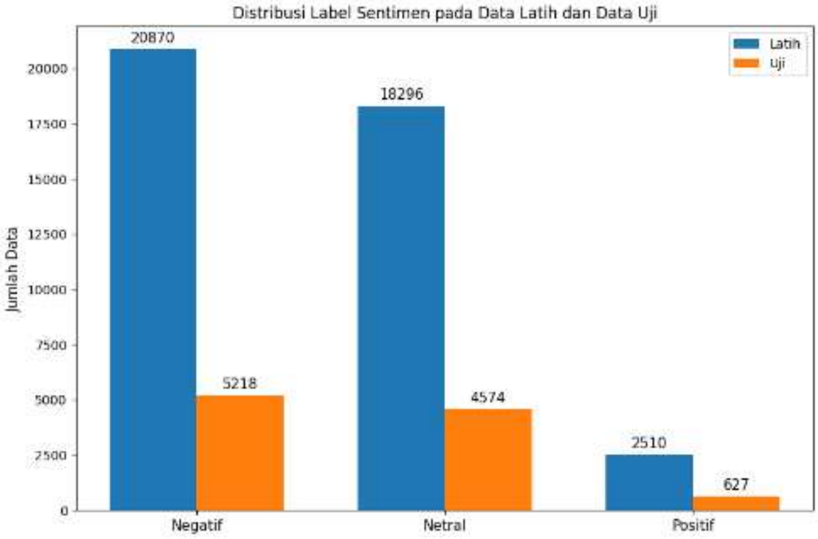

Fig 4. Sentiment Class Distribution for Training and Test Data

The text feature successfully highlighted important words in the corpus. Both TF-IDF (Fig 5.) and BoW (Fig 6.) show the dominance of the word "coretax" and tax-related terms, reflecting the main topics of public discussion.

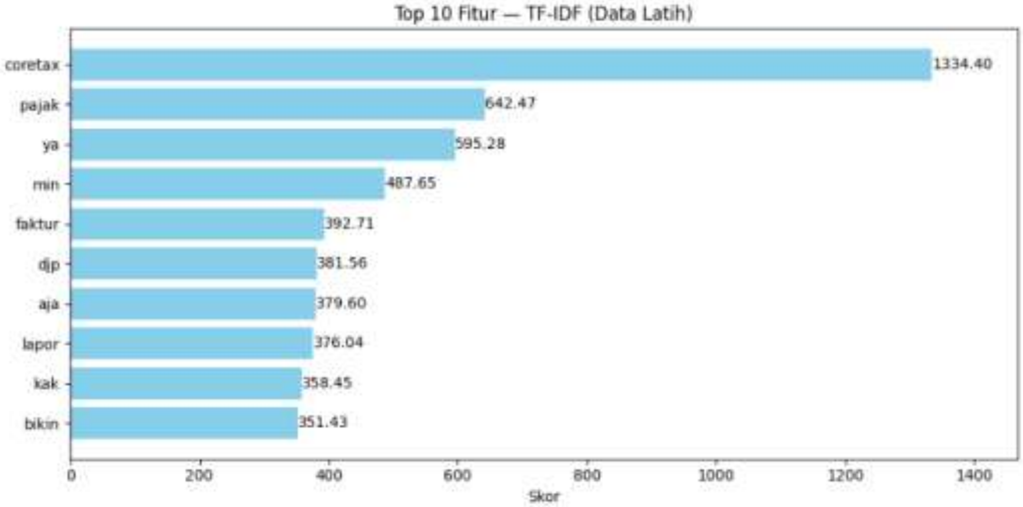

Fig 5. Ten Features with the Highest Frequency in BoW (Training Data)

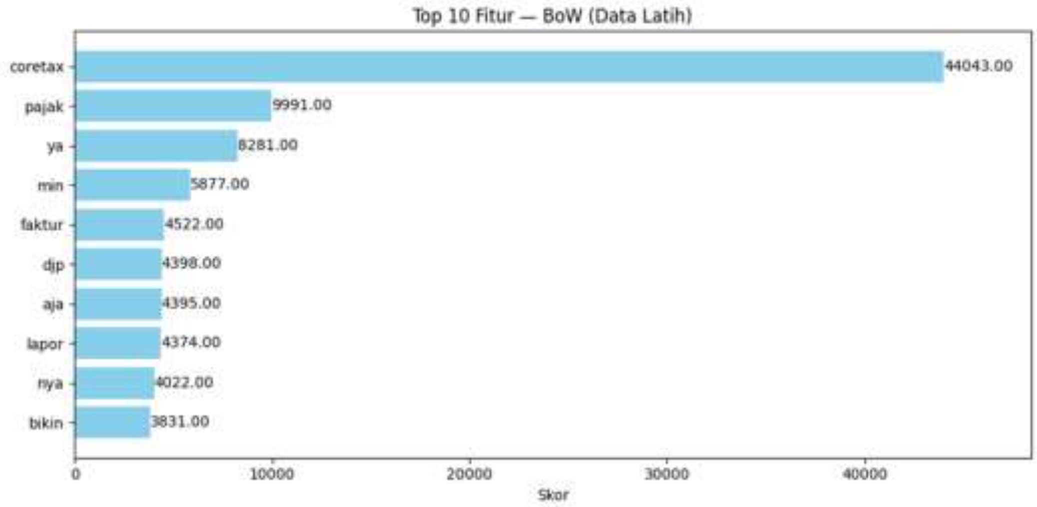

Fig 6. Ten Features with the Highest Frequency in BoW (Training Data)

After data balancing was applied to the training set, the class distribution became uniform according to the median, while the test set retained its original distribution (Fig 7.). This ensures fair learning models without changing the evaluation conditions.

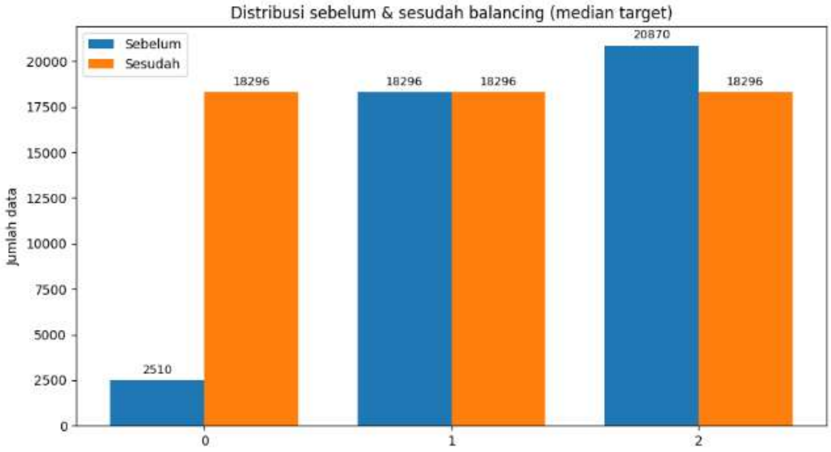

Fig 7. Data Distribution Before and After Balancing (Median Target)

3.3 Result of Training Model

The model performance evaluation results show a clear difference between the algorithms. Generally, Logistic Regression and SVM consistently show higher accuracy and F1-score compared to Naïve Bayes across all scenarios. The boxplot of cross-validation results illustrates the performance distribution of each model, with mean accuracies relatively close for Logistic Regression and SVM, indicating stability across folds, while Naïve Bayes tends to be more fluctuating.

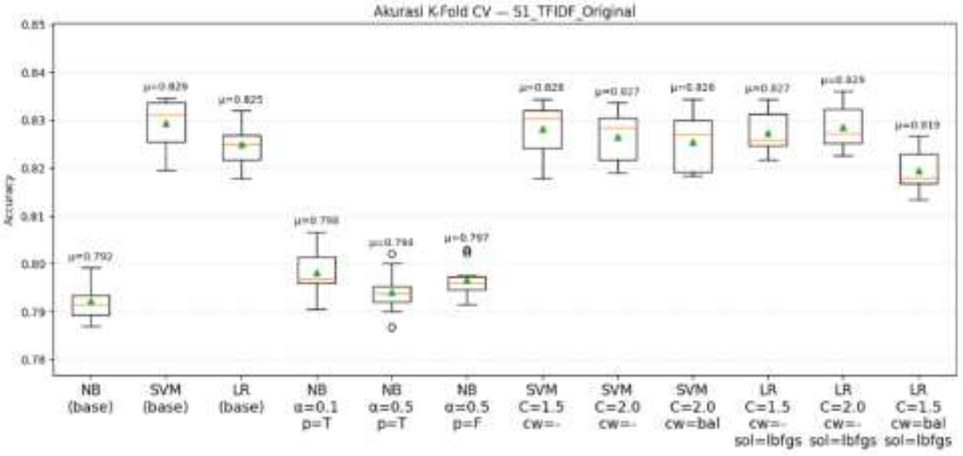

Fig 8. Boxplot of Cross-Validation Accuracy Results for Scenario 1 (Original TF-IDF)

The validation results for Scenario 1 (Original TF-IDF) are shown in Fig 8., where SVM recorded the highest average accuracy of 0.829, followed by Logistic Regression at 0.825, and Naïve Bayes at 0.792. In the hyperparameter variation experiment, changing the smoothing (α) or prior in Naïve Bayes only provided a marginal improvement, while SVM and Logistic Regression were relatively stable to variations in regularization parameters and solvers. Although the absolute performance differences among the top configurations are relatively small (within 1–2%), the cross-validation results reveal clear differences in performance stability. As illustrated in Fig 8., Logistic Regression consistently achieves higher median accuracy with a narrower interquartile range across folds compared to SVM and Naïve Bayes. This indicates lower variance and greater robustness to data partitioning, supporting the selection of Logistic Regression as the most stable model rather than relying solely on marginal accuracy differences.

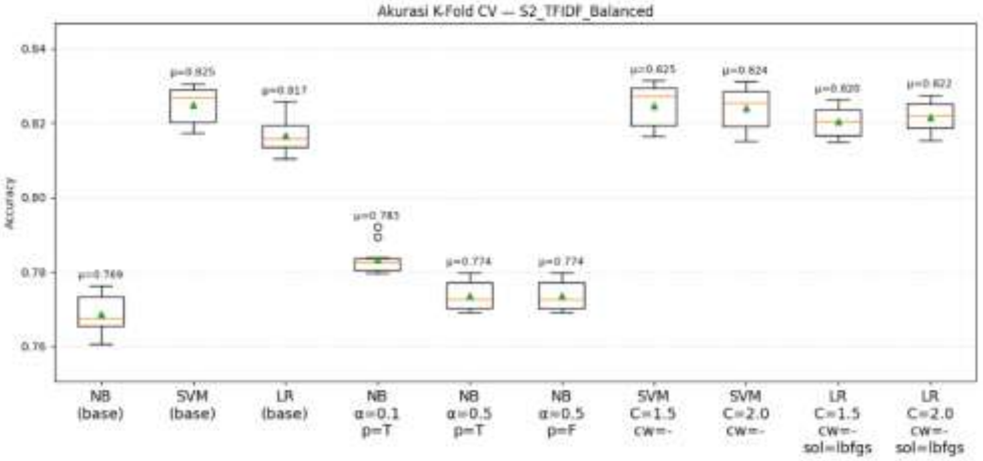

Fig 9. Boxplot of Cross-Validation Accuracy Results for Scenario 2 (TF-IDF Balanced)

In Scenario 2 (TF-IDF Balanced), data balancing improves the distribution of predictions across minority classes, especially the Positive class, although the average accuracy slightly decreases. The accuracy distribution for each fold remained stable for SVM and Logistic Regression (Fig 9.). Naive Bayes remains the lowest-performing algorithm, showing outliers in some folds. In the balanced TF-IDF scenario (S2), although the mean accuracy differences among Logistic Regression and SVM remain within 1–2%, the cross-validation boxplots reveal clear differences in stability. Logistic Regression exhibits the narrowest interquartile range and minimal outliers, indicating lower variance and greater robustness to data partitioning. In contrast, SVM shows slightly higher variability across folds, while Naïve Bayes experiences a noticeable performance degradation under data balancing.

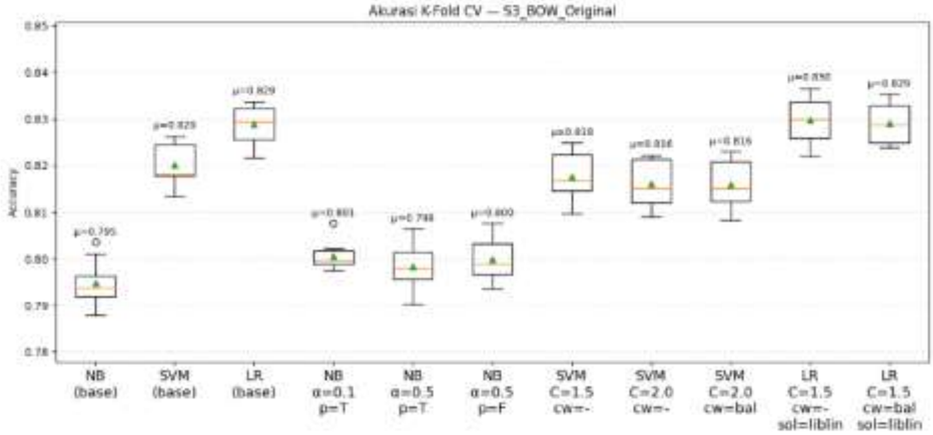

Fig 10. Boxplot of Cross-Validation Accuracy Results for Scenario 3 (Original BoW)

Scenario 3 (BoW Original) shows a similar pattern. Logistic Regression recorded the highest accuracy of 0.830 with regularization configuration C=1.5 and solver liblinear, SVM was stable in the range of 0.818, while Naïve Bayes was at 0.798–0.801 (Fig 10.). Logistic Regression consistently achieves the highest median accuracy with a relatively narrow interquartile range, indicating both strong performance and low variance across cross-validation folds. While SVM exhibits competitive accuracy, its slightly wider variability suggests greater sensitivity to data partitioning. Naïve Bayes remains consistently below both models, confirming its role as a baseline rather than a competitive classifier in this setting.

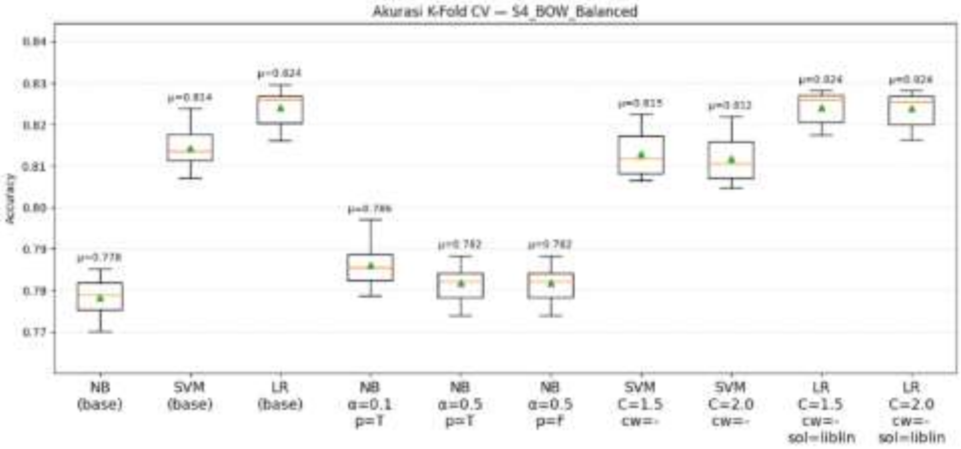

Fig 11. Boxplot of Cross-Validation Accuracy Results for Scenario 4 (BoW Balanced)

In Scenario 4 (BoW Balanced), the mean accuracy of Logistic Regression remained at 0.821–0.824, indicating stable performance across folds, SVM was stable at 0.808–0.814, and Naïve Bayes remained low at 0.778–0.786 (Fig 11.). The cross-validation results further confirm the importance of model stability under distributional shifts. Logistic Regression consistently achieves the highest median accuracy (approximately 0.824) with a relatively narrow interquartile range across folds, indicating strong robustness to data partitioning when combined with balanced training data. Although SVM demonstrates competitive performance, its slightly wider variability suggests greater sensitivity to fold variation. In contrast, Naïve Bayes experiences a noticeable reduction in accuracy under data balancing, highlighting its limited adaptability to altered class distributions. These results indicate that the superior performance of Logistic Regression in the balanced BoW setting is driven not by marginal accuracy gains alone, but by its consistently stable behavior across folds, resulting in a favorable trade-off between accuracy, stability, and fairness. Balancing improves recall for minority classes, especially negative ones, although overall accuracy decreases slightly.

3.4 The Result of Testing Model

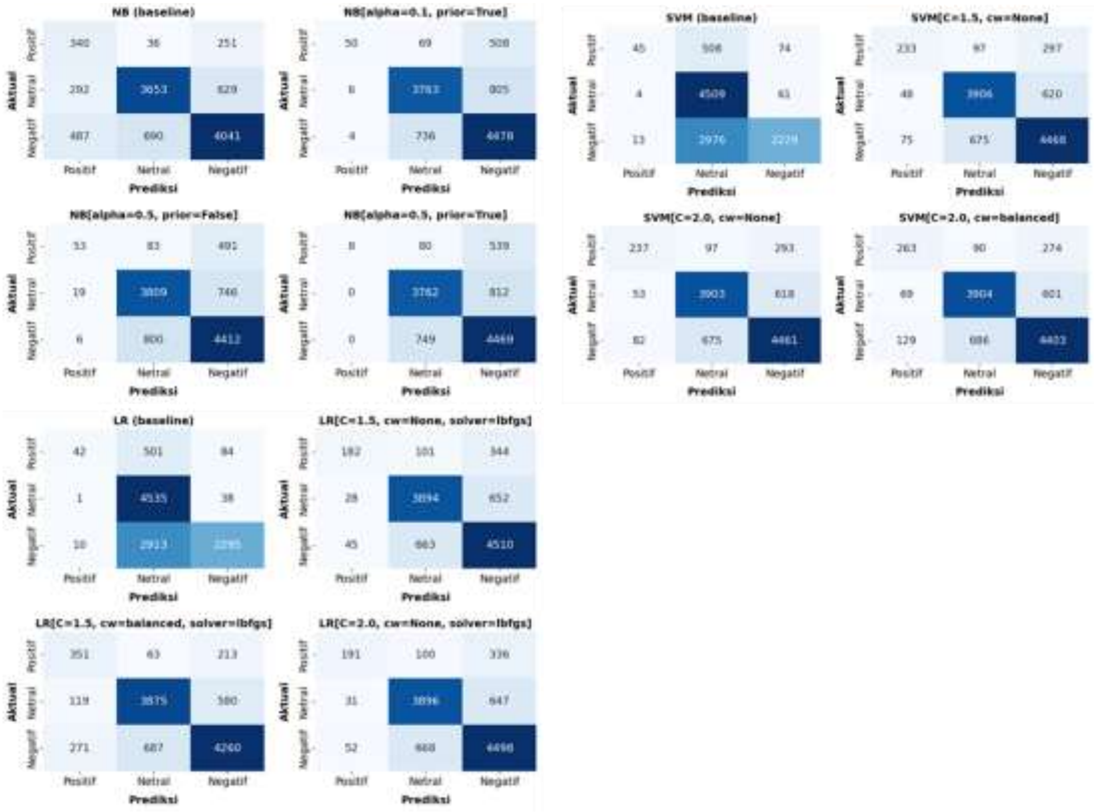

Fig 12. Confusion Matrix for Scenario 1 (Original TF-IDF)

Based on the confusion matrix results for Scenario 1, as shown in Fig 12., the Neutral class was the easiest to identify for all algorithms. For the Positive class, both SVM and Logistic Regression tended to misclassify instances as Neutral, whereas Naïve Bayes more frequently shifted them to the Negative class.In Table 3 we can see that Logistic Regression and SVM achieved a more balanced performance across sentiment classes compared to Naïve Bayes. Variations in hyperparameter settings did not alter this overall pattern.

Table 3. Summary of Model Testing Results on Test Data Scenario 1

Skenario	**Model**	***Hyperparameter***	***Accuracy***	***Precision (macro)***	***Recall (macro)***	***F1-score (macro)***
S1 TF-IDF Original	***Naïve Bayes***	*baseline*	0.771	0.653	0.705	0.668
		α=0.1, prior=True	0.796	0.810	0.587	0.594
		α=0.5, prior=False	0.794	0.757	0.588	0.595
		α=0.5, prior=True	0.791	0.862	0.564	0.552
	SVM	*baseline*	0.651	0.744	0.495	0.479
		C=1.5, cw=None	0.826	0.773	0.694	0.720
		C=2.0, cw=None	0.826	0.767	0.695	0.720
		C=2.0, cw=balanced	0.823	0.746	0.706	0.722
	Logistic Regression	*baseline*	0.660	0.771	0.499	0.483
		C=1.5, cw=None, solver=lbfgs	0.824	0.790	0.669	0.699
		C=2.0, cw=None, solver=lbfgs	0.824	0.784	0.673	0.703
		C=1.5, cw=balanced, solver=lbfgs	0.814	0.718	0.741	0.728

To provide deeper insight beyond model ranking, the consistent robustness of Logistic Regression across all experimental scenarios can be explained by its linear probabilistic formulation and regularization mechanism, which are well suited for high-dimensional and sparse text representations such as TF-IDF and BoW. Unlike Naïve Bayes, which relies on the strong assumption of feature independence, Logistic Regression directly optimizes class decision boundaries using weighted feature combinations, allowing it to better capture discriminative patterns across sentiment classes, particularly when feature correlations are present.

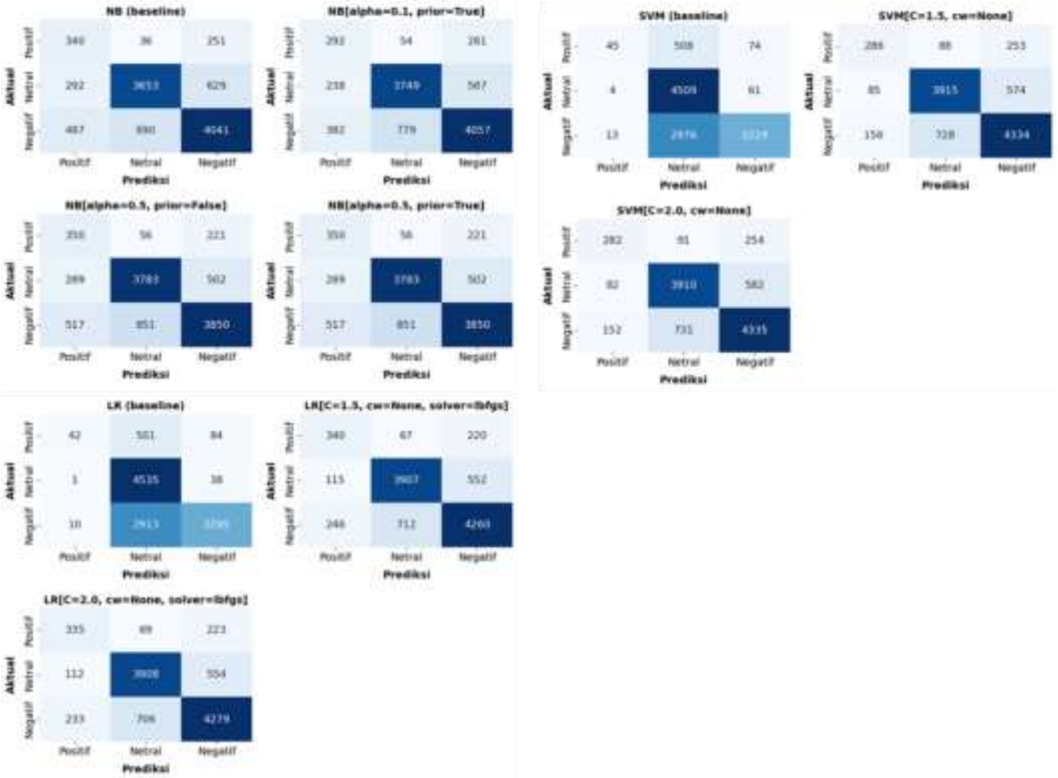

Fig 13. Confusion Matrix for Scenario 2 (TF-IDF Balanced)

The confusion matrix results for Scenario 2 are presented in Figures 4.17, 4.18, and 4.19. Based on these confusion matrices, it can be observed that the application of data balancing leads to more evenly distributed predictions for the Positive and Negative classes. The confusion matrices indicate an improvement in recall for minority classes, although precision slightly decreases due to an increase in false positives. This pattern is also reflected in the classification report (Table 4), where Logistic Regression and SVM remain consistent with macro F1-scores of approximately 0.72–0.73, while Naïve Bayes continues to lag behind at around 0.66. Variations in hyperparameter settings do not alter the overall pattern, leading to the conclusion that data balancing has a greater impact on prediction distribution than on the metric values.

Table 4. Summary of Model Testing Results on Test Data Scenario 2

Skenario	Model	*Hyperparameter*	*Accuracy*	*Precision (macro)*	*Recall (macro)*	*F1-score (macro)*
S2 TF-IDF Balanced	***Naïve Bayes***	*baseline*	0.771	0.653	0.705	0.668
		α=0.1, prior=True	0.777	0.654	0.688	0.666
		α=0.5, prior=False	0.766	0.650	0.708	0.665
		α=0.5, prior=True	0.766	0.650	0.708	0.665
	SVM	*baseline*	0.651	0.744	0.495	0.479

Skenario	Model	*Hyperparameter*	*Accuracy*	*Precision (macro)*	*Recall (macro)*	*F1-score (macro)*
		C=1.5, cw=None	0.819	0.737	0.714	0.724
		C=2.0, cw=None	0.818	0.737	0.712	0.723
	Logistic Regression	*baseline*	0.660	0.771	0.499	0.483
		C=1.5, cw=None, solver=lbfgs	0.816	0.722	0.738	0.729
		C=2.0, cw=None, solver=lbfgs	0.818	0.724	0.736	0.730

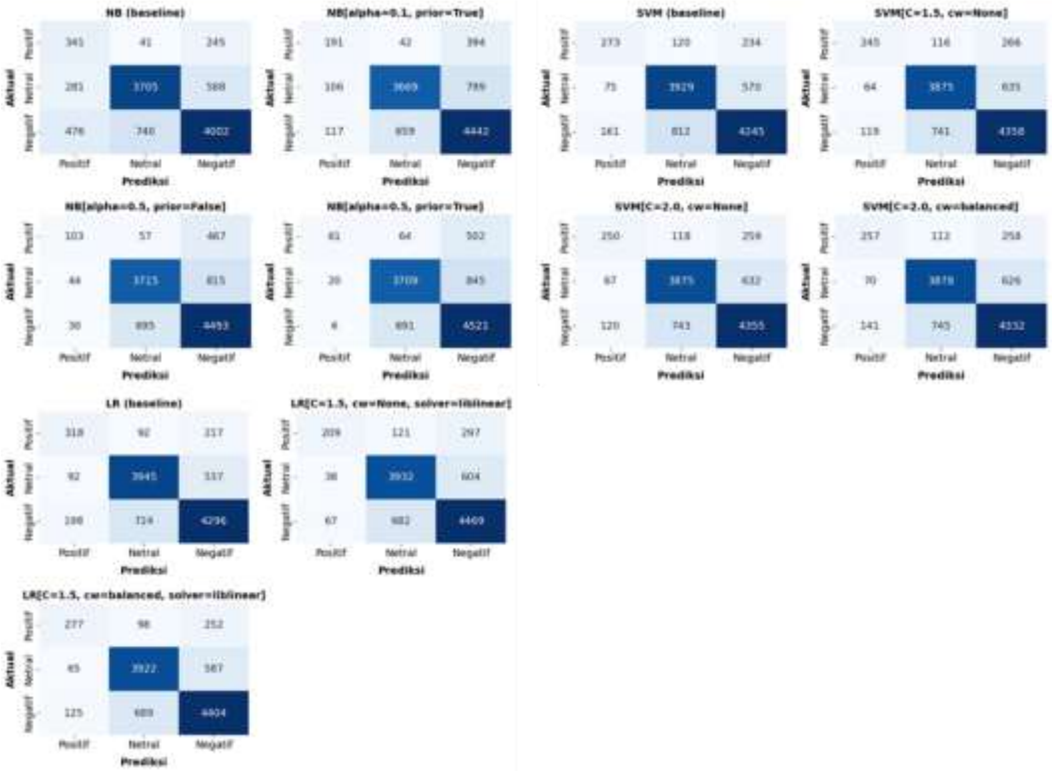

Fig 14. Confusion Matrix for Scenario 3 (Original BoW)

The confusion matrix results for Scenario 3 are presented in Fig 14. Based on these confusion matrices, Logistic Regression produces the most balanced distribution of predictions across sentiment classes. SVM still tends to experience difficulties in distinguishing between the Neutral and Negative classes, while Naïve Bayes again records the highest number of errors, particularly for the Positive class, which is frequently misclassified as Negative. These findings are reinforced by the classification report in Table 5., where Logistic Regression achieves a macro F1-score of approximately 0.73, SVM ranges between 0.71 and 0.72, and Naïve Bayes attains only 0.60–0.67. Variations in hyperparameter settings, such as the use of the liblinear solver in Logistic Regression, have only a minor influence on precision and recall and do not alter the main trend that Logistic Regression remains superior in this scenario.

Table 5. Summary of Model Testing Results on Test Data Scenario 3

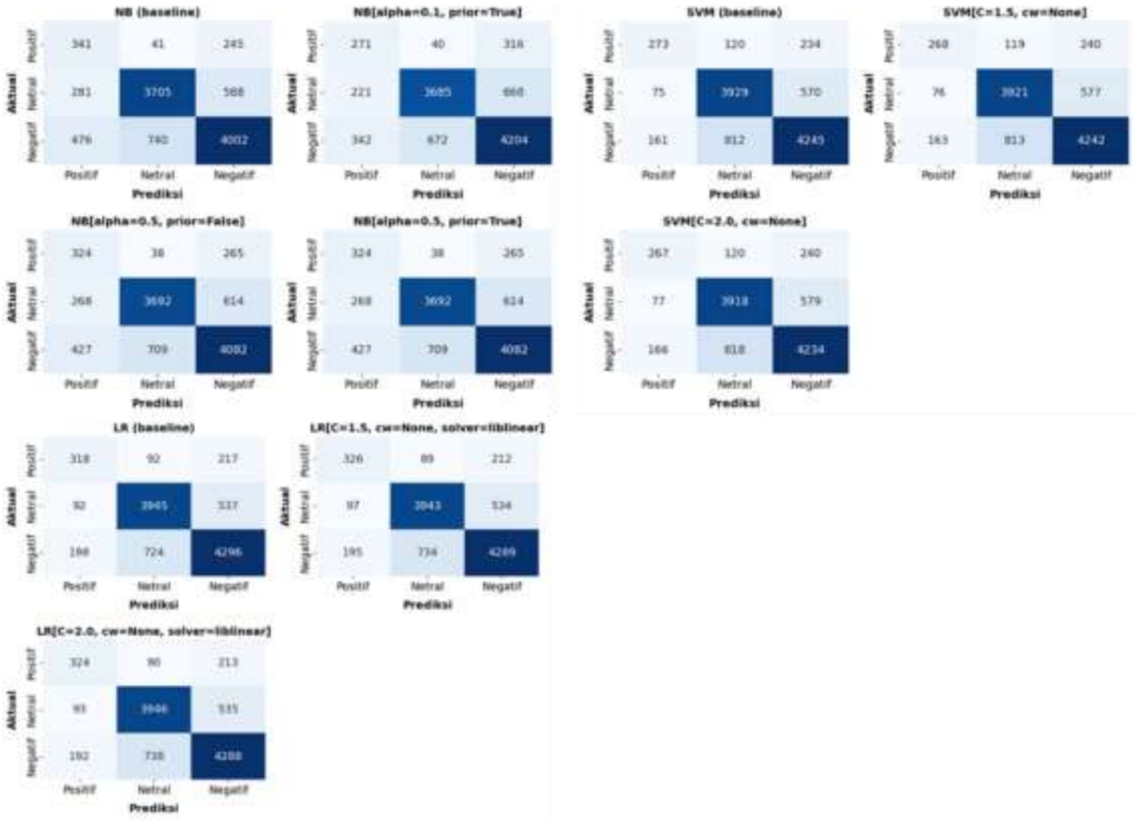

Fig 15. Confusion Matrix for Scenario 4 (BoW Balanced)

The confusion matrix results for Scenario 4 are presented in Fig 15. Based on these confusion matrices, data balancing again helps improve recall for minority classes, particularly the Negative class, although the overall accuracy slightly decreases compared to the original condition. The confusion matrices indicate a more evenly distributed set of predictions, with Logistic Regression showing the most consistent performance across sentiment classes. SVM also remains relatively stable, although misclassifications between the Neutral and Negative classes are still observed, while Naïve Bayes continues to exhibit the highest error rate. These findings are consistent with the classification report in Table 6, where Logistic Regression records the highest macro F1-score at approximately 0.735, SVM achieves values around 0.71, and Naïve Bayes attains only 0.66–0.67. Variations in hyperparameter settings, such as different values of C in SVM and Logistic Regression, do not produce meaningful differences and therefore do not alter the main conclusions.

Table 6. Summary of Model Testing Results on Test Data Scenario 4

Model	*Hyperparameter*	*Accuracy*	*Precision (macro)*	*Recall (macro)*	*F1-score (macro)*
Naïve Bayes	*baseline*	0.772	0.655	0.707	0.670
	α=0.1, prior=True	0.783	0.658	0.681	0.667
	α=0.5, prior=False	0.777	0.657	0.702	0.672
	α=0.5, prior=True	0.777	0.657	0.702	0.672
SVM	*baseline*	0.811	0.728	0.703	0.713
	C=1.5, cw=None	0.809	0.725	0.699	0.710
	C=2.0, cw=None	0.808	0.723	0.698	0.708
Logistic Regression	*baseline*	0.821	0.734	0.731	0.732
	C=1.5, cw=None, solver=liblinear	0.821	0.736	0.735	0.735
	C=2.0, cw=None, solver=liblinear	0.821	0.737	0.734	0.735

The confusion matrix for each scenario is shown the Neutral class is the easiest for all algorithms to recognize. Positive and negative classes are more frequently confused in Naïve Bayes, while Logistic Regression and SVM show a more balanced distribution of predictions between classes. Applying data balancing improves the recall of the minority class, but precision slightly decreases due to an increase in false positives, which aligns with the research objective of evaluating model stability and fairness.

Logistic Regression consistently emerges as the most stable and robust algorithm, followed by SVM, while Naïve Bayes is less recommended. Overall, TF-IDF representations tend to maintain stable performance across scenarios, while BoW with Logistic Regression recorded the highest accuracy. Hyperparameter variations only have a limited impact, so the performance ranking between algorithms does not change. This finding confirms that Logistic Regression with BoW Balanced is the best variant configuration for Coretax sentiment analysis on Platform X.

In addition, Logistic Regression demonstrates greater stability under varying data distributions due to its compatibility with class weighting strategies. When data balancing is applied, Logistic Regression effectively adjusts decision thresholds through class_weight without substantially altering the learned feature space. This behavior explains why, across balanced scenarios, the model maintains relatively stable macro F1-scores while improving recall for minority classes, despite a slight reduction in overall accuracy. In contrast, Naïve Bayes exhibits higher sensitivity to changes in class distribution, leading to unstable precision–recall trade-offs, while SVM shows competitive performance but greater variability across scenarios due to its margin-based optimization, which is more sensitive to changes in class composition.

Furthermore, the robustness of Logistic Regression is reinforced by its moderate sensitivity to hyperparameter variations. As observed in Table 2, changes in the regularization parameter (C), solver choice, or class weighting produce only incremental performance differences, indicating that the model operates within a relatively flat performance landscape. This characteristic makes Logistic Regression less prone to overfitting and more reliable when applied across different feature representations and data distributions. Overall, these findings suggest that the superior performance of Logistic Regression in this study is not merely a result of higher accuracy values, but rather a consequence of its balanced bias–variance trade-off, adaptability to data imbalance, and resilience to representation changes, which collectively contribute to its consistent robustness across all evaluated scenarios.

4. Conclusion

Based on the evaluation across four experimental scenarios, the results show that sentiment analysis performance and stability are determined by the interaction between feature representation, data distribution, and classification model. In Scenario 1, Logistic Regression and SVM outperform Naïve Bayes, achieving average accuracy values of approximately 0.82–0.83 and macro F1-scores around ±0.72. In Scenario 2, Logistic Regression remains the most stable model, although data balancing reduces average accuracy to about 0.80, macro F1-scores remain consistent in the range of 0.72–0.73 that indicating improved performance for minority sentiment classes. Similar trends are observed with Bag of Words representations, where Logistic Regression achieves an accuracy of approximately ±0.81 and a macro F1-score of around 0.73 in Scenario 3 (BoW Original), and records the best results in Scenario 4 (BoW Balanced) with an accuracy of 0.821 and a macro F1-score of 0.732, alongside more balanced prediction distributions.

Overall, these results confirm that no single component independently determines the success of sentiment analysis. Instead, the combined effects of representation, data distribution, and model characteristics govern performance stability and fairness. Within this interaction framework, Logistic Regression demonstrates the highest robustness to changes in feature representation and data distribution, making it the most reliable choice for Coretax sentiment analysis on platform X. SVM ranks second with competitive and consistent performance, whereas Naïve Bayes remains relevant as a simple baseline model despite its notable limitations under imbalanced conditions. TF-IDF representations tend to provide more stable performance, while the combination of BoW with data balancing and Logistic Regression yields the best overall trade-off between accuracy, stability, and fairness. Variations in hyperparameters result in only small to moderate performance differences and do not alter the primary patterns driven by the interaction between representation, distribution, and model.

Future research may explore more advanced representation and modeling approaches, particularly transformer-based models such as IndoBERT or IndoBERTweet, to further investigate how contextual representations interact with data distribution in sentiment classification. Ensemble strategies that combine linear classifiers with neural models may also be explored to enhance robustness and fairness across sentiment classes. In addition, future studies could incorporate more domain-specific labeling strategies or partial manual annotation to reduce potential noise introduced by automatic labeling. More in-depth analyses, such as aspect-based and temporal sentiment analysis, may provide richer insights into public perceptions of Coretax by examining specific system features and sentiment dynamics over time.

References

[1] A. Purwitasari, B. Mutafarida, and Yuliani, "Urgensi pajak dalam mendorong pembangunan infrastruktur dan pertumbuhan ekonomi di Indonesia," *Jurnal Ilmiah Ekonomi dan Manajemen*, vol. 2, no. 6, 2024.

[2] R. S. Aliyudin, E. F. Ahmad, and N. Nizhan, "Pengaruh sistem perpajakan, diskriminasi, teknologi dan informasi perpajakan terhadap persepsi wajib pajak mengenai penggelapan pajak," *J-AKSI : Jurnal Akuntansi Dan Sistem Informasi*, vol. 2, no. 2, 2021, doi: 10.31949/j-aksi.v2i2.1615.

[3] Republik Indonesia, *Peraturan Presiden (Perpres) Nomor 40 Tahun 2018 tentang Pembaruan Sistem Administrasi Perpajakan*. 2018.

[4] G. Naufal Wala and R. Tesalonika, "Transformasi Administrasi Perpajakan Melalui Coretax: Analisis Hukum dan Akuntansi," *Jurnal Komunikasi dan Ilmu Sosial*, vol. 2, no. 4, 2024, doi: 10.38035/jkis.v2i4.1479.

[5] M. Saraswati and D. Riminarsih, "Analisis sentimen terhadap pelayanan KRL Commuterline berdasarkan data Twitter menggunakan algortima Bernoulli Naive Bayes," *Jurnal Ilmiah Informatika Komputer*, vol. 25, no. 3, 2020, doi: 10.35760/ik.2020.v25i3.3256.

[6] J. A. Nursiyono and C. Chotimah, "Analisis Sentimen Netizen Twitter terhadap Pemberitaan PPN Sembako dan Jasa Pendidikan dengan Pendekatan Social Network Analysis dan Naive Bayes Classifier," *J Statistika: Jurnal Ilmiah Teori dan Aplikasi Statistika*, vol. 14, no. 1, 2021, doi: 10.36456/jstat.vol14.no1.a3868.

[7] M. I. Fauzy and F. F. Abdulloh, "Sentiment Analysis of Online Vehicle Tax Renewal Application Users Using SVM Algorithm," *Journal of Applied Informatics and Computing*, vol. 8, no. 2, 2024, doi: 10.30871/jaic.v8i2.8654.

[8] E. Sutoyo and A. Almaarif, "Twitter sentiment analysis of the relocation of Indonesia's capital city," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 4, 2020, doi: 10.11591/eei.v9i4.2352.

[9] f. Fathoni, A. Faradhisa Ansori, I. Nailah Ramadhani, C. Rahmi Anissa, and S. Amelia Putri, "Analisis sentimen Masyarakat Indonesia di Twitter terhadap Sistem Perpajakan 'Coretax' menggunakan metode Naïve Bayes," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 4, 2025, doi: 10.36040/jati.v9i4.14214.

[10] A. K. -, "Sentiment Analysis of X(twitter) Data-a Review Study," *International Journal For Multidisciplinary Research*, vol. 6, no. 2, 2024, doi: 10.36948/ijfmr.2024.v06i02.15636.

[11] M. A. Khder, "Web scraping or web crawling: State of art, techniques, approaches and application," *International Journal of Advances in Soft Computing and its Applications*, vol. 13, no. 3, 2021, doi: 10.15849/ijasca.211128.11.

[12] S. Dhummad, “The Imperative of Exploratory Data Analysis in Machine Learning,” *Scholars Journal of Engineering and Technology*, vol. 13, no. 01, 2025, doi: 10.36347/sjet.2025.v13i01.005.

[13] Y. Findawati, *Buku Ajar Text Mining*. 2020. doi: 10.21070/2020/978-623-6833-19-3.

[14] D. Purnamasari *et al.*, *Pengantar Metode Analisis Sentimen*. 2024.

[15] B. Wilie *et al.*, “IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding,” 2024. doi: 10.18653/v1/2020.aacl-main.85.

[16] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 2019.

[17] crypter70, “IndoBERT-Sentiment-Analysis,” https://huggingface.co/crypter70/IndoBERT-Sentiment-Analysis.

[18] V. R. Joseph, “Optimal ratio for data splitting,” *Stat Anal Data Min*, vol. 15, no. 4, 2022, doi: 10.1002/sam.11583.

[19] T. P. Kurniawan, M. A. Hariyadi, and C. Crysdian, “Perbandingan feature extraction TF-IDF dan BoW untuk analisis sentimen berbasis SVM,” *Jurnal Cahaya Mandalika*, vol. 3, no. 2, 2023.

[20] N. A. Saran and F. Nar, “Fast binary logistic regression,” *PeerJ Comput Sci*, vol. 11, 2025, doi: 10.7717/PEERJ-CS.2579.

[21] S. Bates, T. Hastie, and R. Tibshirani, “Cross-Validation: What Does It Estimate and How Well Does It Do It?,” *J Am Stat Assoc*, vol. 119, no. 546, 2024, doi: 10.1080/01621459.2023.2197686.

[22] M. Hasnain, M. F. Pasha, I. Ghani, M. Imran, M. Y. Alzahrani, and R. Budiarto, “Evaluating Trust Prediction and Confusion Matrix Measures for Web Services Ranking,” *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2994222.