Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1164−1173
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8072

Sentiment Classification and Interpretation of Tokopedia Reviews: A Machine Learning, IndoBERT, and LIME Approach

Adrian Yoris Mbake Woka, Mahendra Dwifebri Purbolaksono, Dody Qori Utama*

Faculty of Informatics, Informatic, Telkom University, Bandung, Indonesia
Email: [1]oiswoka@students.telkomuniversity.ac.id, [2]mahendradp@telkomuniversity.ac.id, [3,*]dodyqori@telkomuniversity.ac.id
Correspondence Author Email: dodyqori@telkomuniversity.ac.id
Submitted: **21/07/2025**; Accepted: **01/09/2025**; Published**: 02/09/2025**

Abstract−Sentiment classification of user reviews plays a vital role in business decision-making, especially on e-commerce platforms like Tokopedia. This study evaluates the performance of various sentiment classification models such as Logistic Regression LinearSVC, and BERT models, both baseline and fine-tuned. Evaluation metrics used include accuracy, precision, recall, and F1-score, applied to Tokopedia review data labelled based on user ratings. The result is fine-tuned BERT model has the best and consistent result, with 92% accuracy and 0.92 f1-score for each class. This shows that fine-tuned BERT can effectively capture the semantic context of user reviews. Its consistent performance across classes makes it suitable for reliable sentiment classification in real-world applications. Furthermore, fine-tune BERT model is visualized by Local Interpretable Model-agnostic Explanation to identify features – in this case is word – that indicates sentiment as positive or negative. It will show as color, orange for positive and blue as negative. This method will make the model more transparent and more reliable.

Keywords: Sentiment Classification; Tokopedia; Machine Learning; BERT; XAI; LIME

1. INTRODUCTION

Electronic Commerce (e-commerce) represents the transformation from conventional to digital-based trade [1]. In Indonesia, the growth in user numbers increased by 53% from 2020 to 2023 [2]. It reflects on many ecommerce platforms that apppear in Indonesia, such as Tokopedia. Tokopedia was the most visited e-commerce platform in the second quarter of 2022, recording 158.35 million website visitors [2]. This number has led to various user experiences that expressed through reviews on platforms like Google Play Store. These reviews reflect sentiments that separate into three categories: positive, negative, and neutral [3]. The process of identifying and classifying these reviews or opinions is known as sentiment analysis. Sentiment analysis is a subfield of Natural Language Processing (NLP) aimed at extracting and determining the sentiment contained in text [1] [3]. Sentiment analysis is essential for capturing direct user experiences and providing strategic insights for feature development and overall business direction [4]. By analyzing user sentiments, companies can identify which features are effective, underperforming, and should be improved or newly developed.

Machine learning and deep learning models have demonstrated strong performance in sentiment analysis. Algorithms such as Logistic Regression, Support Vector Machine (SVM), and BERT are widely adopted due to their effectiveness in handling sentiment classification tasks. Several studies have explored sentiment analysis using those algorithms. For instance, a study by Youga Pratama et al. titled *"Analisis Sentimen Kendaraan Listrik Pada Media Sosial Twitter Menggunakan Algoritma Logistic Regression dan Principal Component Analysis"* applied Logistic Regression to 1,874 tweet data samples [5]. The model achieved its highest accuracy of 87.9% using a 90% training and 10% testing split, and further improved to 90% accuracy with Principal Component Analysis (PCA) optimization. Another study by Miftahul Qorib et al., titled *"Covid-19 vaccine hesitancy: Text mining, sentiment analysis and machine learning on COVID-19 vaccination Twitter dataset"*, experimented with several models, including Random Forest, Logistic Regression, Decision Tree, LinearSVC, and Naïve Bayes, in combination with vectorization techniques such as Doc2Vec, CountVectorizer, and TF-IDF [6]. The best result was achieved using TF-IDF with LinearSVC, yielding 94.68% accuracy. Another study, *"Fine-tune BERT based on Machine Learning Models For Sentiment Analysis"* by Nadia Smairi et al. employed the IMDB review dataset to compare the performance of BERT-based models [7]. The highest accuracy, 91%, was obtained by combining BERT embeddings with a Genetic Algorithm-optimized SVM (G-SVM). Further, research by Muhammad Zainottah et al. titled *"Critical Sentiment Analysis of Tokopedia Electronic Products Using SVM-Logistic & TF-IDF Ensemble Methods"* focused specifically on Tokopedia product reviews. Their model, which combined SVM and Logistic Regression with TF-IDF, achieved an accuracy of 89% [1]. Another study titled *"Explainable Artificial Intelligence (XAI) towards Model Personality in NLP task"* by Dimas Adi and Nadhila Nurdin used a dataset of tweets related to flights in the United States and applied a Bi-LSTM model that reached 78% accuracy [8]. The study also incorporated Local Interpretable Model-agnostic Explanations (LIME), Shapley Additive Explanations (SHAP), and Anchor for model explanation, demonstrating the growing importance of interpretability in NLP-based sentiment classification tasks. Lastly, the study titled "*An Explainable AI Model for Hate Speech Detection on Indonesian Twitter*", by M. A. Ibrahim et al. use Logistic Regression, Multinomial Naive Bayes, Random Forest, and XGBoost as the machine learning model to classify 13,169 tweets [9]. The result is Logistic Regression, Multinomial Naive Bayes, and XGBoost show a same performance with 83% accuracy and 79% F1-score. This study also visualize the result in XAI through LIME explanation. While several studies have achieved high accuracy and employed Explainable AI (XAI) methods, none have specifically applied machine learning and deep learning models to real user reviews on Tokopedia. Moreover, there is a lack of research

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1164−1173
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8072

analyzing real-time, the Tokopedia review data with an emphasis on interpretability using XAI techniques such as LIME, which this study aims to address.

Customer reviews on e-commerce platforms like Tokopedia provide valuable insights for companies from user satisfaction and experiences. However, traditional sentiment analysis models operate as "black boxes" [10], offering predictions without explaining their reason. This lack of transparency limits business understanding of the reason why customers feel a certain way. XAI addresses this problem by providing an interpretable system. Instead of just classifying reviews, XAI can combine with machine learning or deep learning model to explain which specific words or phrases that indicate the sentiment. This transparency helps businesses make informed strategic decisions and builds trust in AI systems [11]. This study applies LIME as one of the approaches in XAI to sentiment analysis of Tokopedia reviews using machine learning models such as Logistic Regression and LinearSVC, also transformer-based BERT. This study evaluates the model and visualizes the best model result with LIME. Our approach delivers not only sentiment classification but also clear explanations for each prediction.

2. RESEARCH METHODOLOGY

2.1 Research Stages

The aim of this study is to compare Logistic Regression, SVM, and BERT in classifying positive and negative reviews. The research process begins with dataset collection and ends with model explanation using LIME. The steps taken in this study are illustrated in Figure 1.

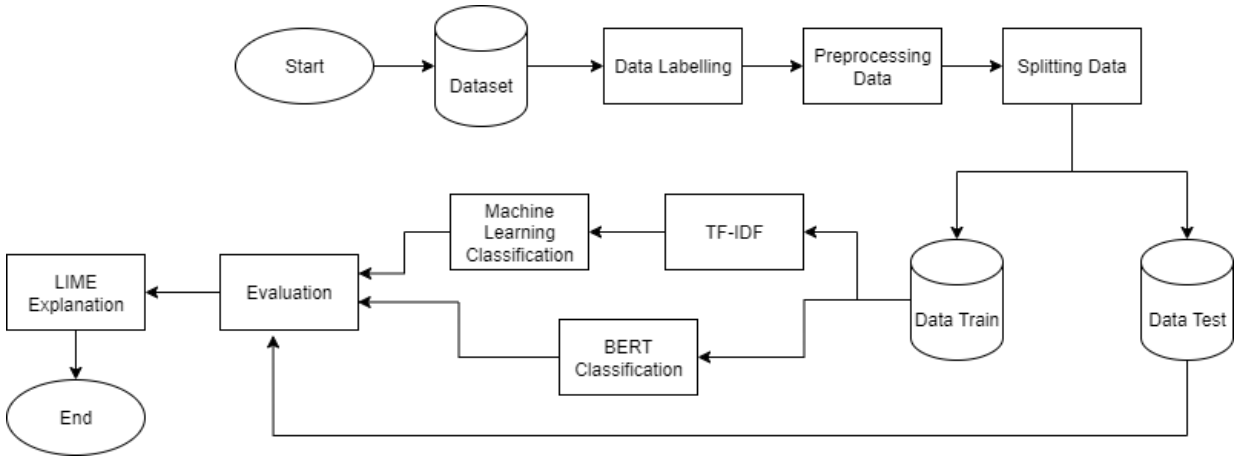

Figure 1. Research Flow Diagram

Figure 1 illustrates the system workflow implemented in this study. The process begins with the dataset collection stage, which serves as the primary source for the entire classification process. The dataset then labeled, where each entry is categorized into one of two classes: positive and negative. After labeling is complete, every data entry goes through preprocessing, such as lowercasing, cleaning, tokenizing, normalizing, removing stopword, and stemming. The processed data is then split into two subsets: data train, which is used to build the model, and data test, which is used to evaluate the performance of the trained model. Next, two classification approaches are used. In the first approach, the training data is transformed into a numerical form using the TF-IDF technique. The resulting transformation is then used to train a traditional machine learning classification model. In the second approach, the training data is processed using a BERT-based embedding technique. The vector representations produced by the embedding are then used in the classification process with BERT. After both approaches complete the classification process, the results are evaluated in the performance evaluation stage, which includes accuracy measurement and other metrics. As the final stage, the classification results from the best-performing model are visualized using the LIME technique.

2.2 Dataset

This study uses a dataset obtained from the Kaggle website, consisting of Tokopedia app reviews from the Google Play Store [12]. The data was collected from September 13, 2018, to October 25, 2024, containing a total of 71,241 entries.

2.3 Data Labelling

Data labelling process was based on the review scores provided by users. Reviews with scores of 4 and 5 are positive sentiment, while scores of 1 and 2 are negative sentiment. To avoid ambiguity or noise during the model training process, reviews with a score of 3 were removed from the dataset, as they are considered as neutral. The dataset contains 35,599 entries labelled as positive sentiment and 31,058 entries labelled as negative sentiment. To balance

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1164−1173
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8072

the dataset, the undersampling method was applied. Undersampling is a technique that reduces the number of data points in the majority class to match the number in the minority class [13].

2.4 Data Preprocessing

At this stage, since the initial dataset contains a significant amount of noise, the data needs to be processed beforehand to enable the classification model to produce more optimal results. The preprocessing steps in this study include lowercasing, cleansing, tokenizing, normalizing, removing stopword, and stemming. Examples of the output from each preprocessing step are shown in Table 1.

Table 1. Preprocessing Data

Review	Lowercasing	Cleansing	Tokenizing	Normalizing	Removing Stopword	Stemming
Aplikasinya mulai lemott	aplikasinya mulai lemott	aplikasinya mulai lemott	[aplikasinya, mulai, lemott]	[aplikasinya, mulai, lemott]	[aplikasinya, lemott]	aplikasi lemott

This preprocessing includes six steps based on Table 1, those are:

a. Lowercasing is the process of converting all text data to lowercase letters.
b. Cleansing refers to the removal of special characters, excessive spaces, URLs, punctuation marks, and duplicated characters within a word. Cleaning process is critical since those things do not have meaning and will not affect their sentiment value [6]. Cleaning up review texts includes removing non-useful signs and punctuation, such as #, /, \, using python re package.
c. Tokenizing is the preprocessing step that splits long tweets into words [14]. These words are referred to as *tokens*, as each token represents a single word [15].
d. Normalizing is the process of converting slang words or typographical errors into their standard forms. Collection of slang and non-standard words are converted into their official [6]. In this study, the word list used for normalization was created by the authors.
e. Removing Stopwords involves eliminating unimportant words, typically conjunctions or connecting words [1]. 'NLTK' library stopword was used for this process.
f. Stemming is the process of reducing words to their root forms [6]. Occasionally, some of the words might not be valid in the language. 'Sastrawi' library stemming was used for the experiment.

2.5 Splitting Data

After preprocessing, the dataset was split into two subsets, those are 80% for data train and 20% for data test. This split was performed randomly to ensure reproducibility of the results. The resulting datasets were saved to ensure that all models were trained and tested using the same data. The training set was used to train the classification models, while the testing set was used to evaluate their performance.

2.6 Term Frequency Inverse Document Frequency (TF-IDF)

Since text data is unstructured, a representation method is required to convert it into a numerical format. In this study, the Term Frequency–Inverse Document Frequency (TF-IDF) method was used to transform textual data into numerical form. TF-IDF is a technique that balances frequently occurring words with more specific, informative ones [1]. The formula for TF-IDF is defined as follows [14]:

$$TF = \frac{Number\ of\ times\ terms\ appear\ in\ a\ document}{Number\ of\ Terms\ in\ a\ document} \quad (1)$$

$$IDF = log\frac{Total\ number\ of\ documents\ each\ term\ appears}{Total\ number\ of\ documents} \quad (2)$$

where *TF* represents term frequency and *IDF* represents inverse document frequency. In this study, I'm using trigram approach of TF-IDF, which means the

2.7 Logistic Regression

Logistic Regression is a statistical method used for multivariate analysis and modeling of binary variables [5]. This algorithm is used to estimate the probability of a binary event. Logistic Regression models a linear relationship with the logit function, which is the natural logarithm of the odds of an outcome occurring. The algorithm can be represented by the following equation [16]:

$$\ln\left(\frac{P}{1-P}\right) = b_0 + b_1 X \quad (3)$$

In this equation, P is the probability of the event (e.g., positive sentiment), and $\frac{P}{1-P}$ represents the odds, the logit is the natural log of the odds, b_0 is the intercept, and b_1 is the coefficient showing how changes in X, the input variable, affect the log-odds [16]. This transformation ensures predicted probabilities stay between 0 and 1 and allows the model to handle classification effectively. In this study, Logistic Regression is used as one of the classification

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1164−1173
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8072

models. This model will train with baseline model and with hyperparameter tuning performed using GridSearchCV to optimize its performance. The hyperparameter values explored during the tuning process are summarized in Table 2. This tuning aimed to help the model identify the configuration that yields the best evaluation.

Table 2. Tuning Hyperparameter Logistic Regression

Hyperparameter	Value
C	{0.01, 0.1, 1, 10}
penalty	{'l1', 'l2', 'elasticnet', 'none'}
solver	{'lbfgs', 'newton-cg', 'liblinear', 'sag', 'saga'}

The hyperparameters tuned in this process as explained in Table 2 include *C*, *penalty*, and *solver*. These parameters influence how the Logistic Regression model learns from the data. By exploring different combinations, the goal was to find the settings that produce the most accurate and reliable classification results.

2.8 LinearSVC

LinearSVC is a classification method based on Support Vector Machine (SVM) that directly supports both dense and sparse input formats, offering flexibility in choosing penalty and loss functions [6]. LinearSVC is known for its efficiency on large-scale datasets, as it tends to converge faster as the number of samples increases [17]. With its capability to handle high-dimensional data, LinearSVC is well-suited for text classification tasks, particularly for large volumes of review data. In this study, hyperparameter tuning was performed using GridSearchCV to improve model performance. The hyperparameter values explored during the tuning process are summarized in Table 3.

Table 3. Tuning Hyperparameter LinearSVC

Hyperparameter	Value
C	{0.01, 0.1, 1, 10, 100}
loss	{'hinge', 'squared_hinge'}
max_iter	{1000, 2000, 5000}

The hyperparameters tuned in this process as explained in Table 3 include *C*, *loss*, and *max_iter*. These parameters influence how the LinearSVC model build based on data. By exploring different combinations, the goal was to find the settings that produce the most accurate and reliable classification results.

2.9 Bidirectional Encoder Representations from Transformers (BERT)

BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based language model developed by Google AI at the end of 2018 [18]. BERT's innovation lies in its use of a bi-directional Transformer encoder [19]. This can get information in both the left and right direction to capture the richer meaning of a sentence. This also makes BERT highly effective in various Natural Language Processing (NLP) tasks such as text classification, sentiment analysis, question answering, and entity extraction [19]. It is because BERT can learn the semantics and structure of a language. Pre-trained on large-scale corpora like Wikipedia and BookCorpus, BERT can generate rich and contextual word representations [18]. The BERT mode consists of three embedding layers—token embedding, segment embedding, and position embedding—and is pre-trained using two unsupervised tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) [7].

In this study, BERT was applied using two approaches: baseline and fine-tuning. In the baseline approach, BERT was used solely as a feature extractor, where the contextual vector representations of each input text were obtained from the frozen BERT model and passed to a Logistic Regression model. This method does not involve updating BERT's internal parameters. The fine-tuning approach trains the entire BERT architecture end-to-end using the same training dataset. The model was optimized using a cross-entropy loss function, with an initial learning rate of 5e-5, over three epochs, and a batch size of 32. This approach allows BERT's internal weights to adapt to the specific characteristics of the sentiment classification task based on Tokopedia review data.

2.10 Evaluation

After the classification process, the next step is evaluating the performance of each model. This evaluation aims to determine how well the model classifies data and to identify potential issues such as overfitting or underfitting. The evaluation was using a confusion matrix, which presents in four categories: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The confusion matrix is visualized in Figure 2 to facilitate interpretation of the classification outcomes.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1164−1173
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8072

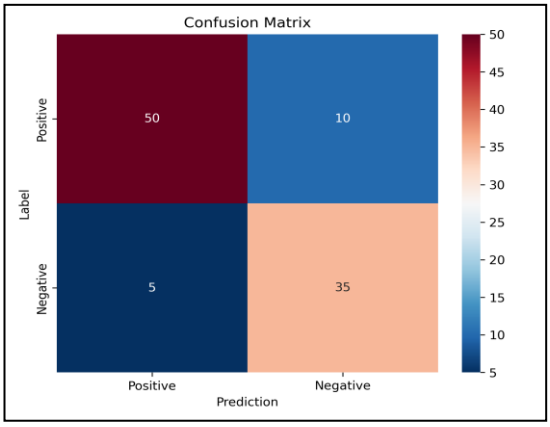

Figure 2. Confusion Matrix

From Figure 2, True Positive (TP) represents instances that are positive and correctly predicted as positive. In Figure 2, the TP is at the top left, with a value of 50. False Positive (FP) represents instances that are negative but incorrectly predicted as positive. In Figure 2, the FP is at the bottom left, with a value of 5. False Negative (FN) represents instances that are positive but incorrectly predicted as negative. In Figure 2, the FN is at the top right, with a value of 10. Lastly, True Negative (TN) represents instances that are negative and correctly predicted as negative. In Figure 2, the TN is at the bottom right, with a value of 35. From this matrix, four key evaluation metrics are derived: Accuracy, Precision, Recall, and F1-score, providing a comprehensive view of model performance. These metrics are consistently applied across all models used in this study, including both machine learning models, those are Logistic Regression, LinearSVC and deep learning model which is BERT.
The confusion matrix in Figure 2 is used to compute the following metrics:

a. Accuracy is the proportion of correct predictions among all predictions made [1]. Accuracy is defined by following equation [20]:

$$Accuracy = \frac{TP+TN}{Total\ All\ Matrix} \tag{4}$$

b. Precision is the proportion of true positives among all predicted positives [1]. This metric is crucial when minimizing false positives is a priority. Precision is defined by following equation [20]:

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

c. Recall is the proportion of true positives correctly identified from actual positives [1]. Recall is defined by following equation [20]:

$$Recall = \frac{TP}{TP+FN} \tag{6}$$

d. F1-score is the harmonic mean of Precision and Recall, especially useful in imbalanced datasets [1]. F1-score is defined by following equation [20]:

$$F1-score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \tag{7}$$

2.11 Local Interpretable Model-agnostic Explanations (LIME)

LIME (Local Interpretable Model-Agnostic Explanations) is a prominent XAI technique designed to provide local interpretation of predictions made by any classification or regression model. It works by creating perturbations around the instance of interest and training a simple, interpretable surrogate model (such as linear regression or a small decision tree) weighted by similarity to approximate the original model's behavior in that local region—ensuring local fidelity while remaining model-agnostic [21]. Because LIME is independent of the underlying model (black-box), it can be employed with complex architectures like BERT, without requiring access to internal model structure. The approach highlights how each input feature contributes to an individual prediction, offering intuitive visualizations that help uncover biases, increase transparency, and foster trust in AI systems [21] [22].

In this study, LIME is used to visualize words that indicate a review as positive or negative. The method generates two types of output: a bar chart showing the direction and magnitude of each word's contribution to the sentiment prediction, and a color-highlighted version of the input text, where different shades represent the level of influence each word has on the model's decision. This visual representation shows the reason why a model classifies reviews as positive or negative. It also helps identify cases where the model might rely on irrelevant or misleading features. By providing this interpretability layer, LIME ensures that the model's predictions are not only quantitatively accurate but also justifiable. This method will increase transparency of a model and gain user trust in the AI system.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1164−1173
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8072

3. RESULT AND DISCUSSION

3.1 Logistic Regression Evaluation

This study evaluated the performance of the Logistic Regression model under two scenarios: the baseline model without any hyperparameter adjustments, and the model after hyperparameter tuning. The result of classification report of baseline model is shown in Table 4.

Table 4. Classification Report Logistic Regression Baseline Model

Label	Precision	Recall	F1-score	Support
Negative	0.90	0.94	0.92	6212
Positive	0.94	0.89	0.91	6212

Based on Table 4, the Logistic Regression baseline model demonstrates strong performance in classifying both negative and positive sentiments. For the negative sentiment class, the model achieves a precision of 0.90, indicating a low false positive rate, and a recall of 0.94, which reflects the model's ability to correctly identify most instances of negative sentiment. This yields an F1-score of 0.92, suggesting a balanced trade-off between precision and recall. Then for the positive sentiment class, the model attains a precision of 0.94 and a recall of 0.89, resulting in an F1-score of 0.91. These values confirm the model's capability to detect positive sentiment effectively, with slightly higher precision than recall. Given the balanced distribution of test data (6,212 samples per class), these findings illustrate that the Logistic Regression baseline model can perform sentiment classification reliably and without introducing bias toward either sentiment category.

Furthermore, the optimal performance for the Logistic Regression model was achieved using a regularization parameter *C* set to 1, *penalty* term set to L2, and *solver* set to sag. The classification report for this hyperparameter configuration is presented in Table 5.

Table 5. Classification Report Logistic Regression Hyperparameter-tuned Model

Label	Precision	Recall	F1-score	Support
Negative	0.89	0.94	0.92	6212
Positive	0.94	0.89	0.91	6212

From Table 5, the Logistic Regression model with hyperparameter tuning demonstrates only a slight improvement over the baseline model. For the negative sentiment class, the tuned model achieves a precision of 0.89 and a recall of 0.94, resulting in an F1-score of 0.92—identical to the baseline. Meanwhile, for the positive sentiment class, the model reaches a precision of 0.94 and recall of 0.89, also matching the F1-score of 0.91 from the baseline. Despite minor differences in the precision and recall values, particularly in how the model balances false positives and false negatives, the overall macro and weighted average metrics remain at 0.92, indicating no substantial gain in classification performance from hyperparameter tuning. These findings suggest that while tuning offers slight refinement in class-level precision and recall, the overall impact on model effectiveness is minimal. Hence, the baseline Logistic Regression model already performs near-optimally for this sentiment classification task.

3.2 LinearSVC Evaluation

Same as Logistic Regression, this model also evaluates two scenarios, those are baseline and hyperparameter-tuned. This aimed to find the best tuning model to classify reviews as positive and negative. The results show that the hyperparameter-tuned for LinearSVC model has a better accuracy than the baseline model. For the baseline model, the accuracy is 91%, but in hyperparameter-tuned LinearSVC model, the accuracy is 92%. The result of classification report of LinearSVC baseline model is shown in Table 6.

Table 6. Classification Report LinearSVC Baseline Model

Label	Precision	Recall	F1-score	Support
Negative	0.90	0.93	0.92	6212
Positive	0.93	0.90	0.91	6212

As presented in Table 6, the LinearSVC baseline model shows strong performance in sentiment classification across both classes. For the negative sentiment class, the model achieves a precision of 0.90, recall of 0.93, and an F1-score of 0.92. For positive sentiment class, it records a precision of 0.93, recall of 0.90, and an F1-score of 0.91. These results show that the model performs well in identifying both sentiment categories, with slightly higher precision for positive sentiment and better recall for negative sentiment. The balanced F1-scores across both classes reflect the model's robustness and stability, especially given the symmetric distribution of the test data (6212 instances per class).

Furthermore, after performing hyperparameter tuning, the LinearSVC model achieved its best performance with the following parameter configuration: $C = 0.1$, *loss* = squared hinge, and *max_iter* = 1000. The classification report for hyperparameter-tuned models is shown in Table 7.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1164−1173
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8072

Table 7. Classification Report LinearSVC Hyperparameter-tuned Model

Label	Precision	Recall	F1-score	Support
Negative	0.89	0.95	0.92	6212
Positive	0.94	0.89	0.91	6212

As shown in Table 7, the tuned model yields precision, recall, and F1-score matrix. For the negative sentiment class, recall achieve score in 0.95 and precision is in 0.89. For the positive sentiment class, precision is in 0.94 and recall is in 0.89. Compared to baseline model, there are slight improvements and decrease from precision and recall. These observations illustrate precision-recall trade-off, where improving recall in one class may inadvertently reduce precision. It means model is better when classifying positive instances, but worse when classifying negative instances. Despite these adjustments, the overall macro-average F1-score remains stable, suggesting that hyperparameter tuning contributes only marginal refinement in model behavior rather than a substantial performance gain.

3.3 BERT Evaluation

This study also evaluates the performance of BERT using two distinct approaches: baseline embedding and fine-tuning. The results show that the accuracy of both approaches are same, it is 92%. The classification report for baseline model is shown in Table 8.

Table 8. Classification Report BERT Baseline Model

Label	Precision	Recall	F1-score	Support
Negative	0.90	0.94	0.92	6212
Positive	0.94	0.89	0.91	6212

As presented in Table 8, the BERT baseline model also demonstrates strong performance in sentiment classification for both sentiment classes. For the negative sentiment class, the model achieves a precision of 0.90, a recall of 0.94, and an F1-score of 0.92. Meanwhile, for the positive sentiment class, it records a precision of 0.94, recall of 0.89, and an F1-score of 0.91. These results suggest that the model is highly effective in distinguishing between positive and negative sentiments, with particularly strong precision in identifying positive sentiment and higher recall in detecting negative sentiment. The relatively balanced F1-scores indicate that the model maintains robustness across both classes. This consistency is further supported by the equal distribution of test samples (6212 instances per class), which ensures fair evaluation across sentiment categories.

Next, for the fine-tuning model, the model was optimized using a cross-entropy loss function, with an initial learning rate of 5e-5, over three epochs, and a batch size of 32. The classification report of this is shown in Table 9.

Table 9. Classification Report BERT Fine-tune Model

Label	Precision	Recall	F1-score	Support
Negative	0.91	0.93	0.92	6212
Positive	0.93	0.91	0.92	6212

As presented in Table 9, the fine-tuned BERT model achieves consistently high performance across both sentiment classes. For the negative sentiment class, it records a precision of 0.91, recall of 0.93, and an F1-score of 0.92. Similarly, the positive sentiment class achieves a precision of 0.93, recall of 0.91, and an F1-score of 0.92. These results indicate a good ability in identifying both positive and negative sentiments, with minimal trade-off between precision and recall. The same score of F1-scores across classes suggest that the fine-tuned model performs consistently on the classification task. Compared to the baseline BERT model, the fine-tuned model shows both improvement and decrease in overall balance. The fine-tuned model demonstrates greater consistency, with identical F1-scores for both classes. This indicates that fine-tuning the BERT architecture allows the model to adapt more effectively to the dataset's characteristics, reducing the disparity between class performances and improving overall generalization. The reduced trade-off between precision and recall suggests that fine-tuning yields a more stable classification output.

3.4 Logistic Regression, LinearSVC, and BERT Evaluation

After conducting research on the three models with two approaches of each model, the evaluation compared to each model. The results show that the macro average of each metric is identical, namely 92%. However, since the macro average is likely the same across all models, we turn to the distribution of the confusion matrix to find the best model. Firstly, true predictions confusion matrix is shown in Figure 3.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1164−1173
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8072

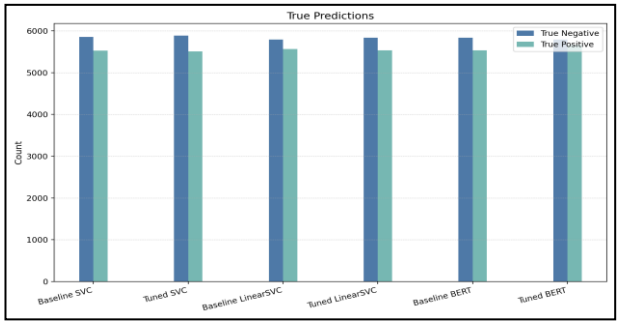

Figure 3. Distribution of True Prediction Confusion Matrix of Each Model

As shown in Figure 3, the true prediction counts, which are True Positive and True Negative are relatively high across all models, with only slight variations. True Negative counts are consistently higher than True Positive counts across the board, indicating that all models are slightly better at correctly identifying negative samples. Notably, the fine-tuned BERT model demonstrates stable and high true prediction counts, comparable to or slightly better than other models. Next, false prediction confusion matrix is shown in Figure 4.

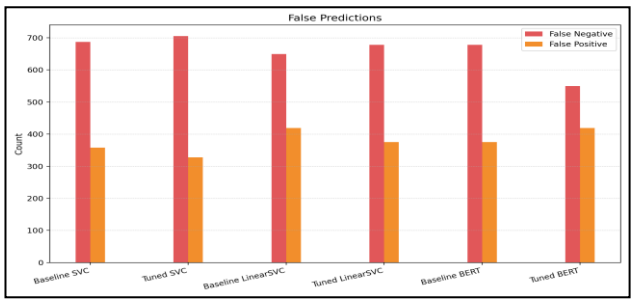

Figure 4. Distribution of False Prediction Confusion Matrix of Each Model

As shown in Figure 4, the false prediction counts (i.e., False Positive and False Negative) reveal more distinct differences. The fine-tuned BERT model exhibits the lowest number of false negatives, suggesting better sensitivity in correctly identifying the positive class. While its false positives are slightly higher than in some other models, the significant reduction in false negatives makes it favorable for sentiment classification tasks where missing positive sentiments may be more critical than incorrectly flagging a neutral or negative sentiment as positive.

Therefore, the fine-tuned BERT model is selected as the best model for the subsequent interpretation stage using the LIME method. Although all three models have identical macro average values, the selection of BERT is based on its performance stability across classes and its ability to capture richer semantic context [7], which enables more meaningful and contextually relevant prediction interpretations. BERT that is trained on large-scale data, demonstrates consistency across classes, meaning that the model can maintain balanced performance in identifying both sentiment categories.

3.5 LIME Explanation

Visualization using LIME is applied to the model's prediction results to observe the features that influence the classification. LIME converts any model into a linear local model and then reports the coefficient values which represent the weights of the features in the model [23] and then visualizes it with probabilities and highlight texts with colors. The visualization covers four types of cases based on the confusion matrix: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), to examine the model's behavior in various scenarios. The model used is the fine-tuned BERT model due to its consistency in capturing semantic meaning. In the LIME visualization, two colors are used to indicate the tendency of a word toward a particular class: orange for the positive class and blue for the negative class. The more intense the color, the stronger the word's indication toward that class. For each example, it is shown in Table 10.

Table 10. LIME Result

Confusion Matrix	Value
TP	pengiriman sudah cepat hari sampe klaim asuransi beneran memuaskan utk produk basah yg rusak akibat expedisi lgs dicover jd gk ribut dgn penjual chat tokpedcare selalu continue sekalipun keluar room gk dianggap endchat
TN	semakin mahal ongkos kirimnya di imingi dgn potongan ongkir tp dia kasi naik ongkir jadix sama sj tdk kena potongan ongkir pdhl sdh jd pelanggan setia dan sdh lama sy gunakan tokped knp smakin mahal ongkir bgini jadix mls kecewa berat

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1164−1173
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8072

Confusion Matrix	Value
FP	saya dapat pemberitahuan dapat hadiah handphone dan menginstal aplikasi ini pas udah saya instal tetap aja dapat pemberitahuan lagi untuk menginstal pdhl udah aq instal berulang kali kek gini sebenarnya hadiah tersebut ada tidak ya maaf sebelumnya
FN	nomer hp saya yang terdaftar di tokopedia sudah tidak aktif pengajuan perubahan nomer hp sudah kali ditolak jika yang ke kali ini ditolak lagi ya uninstall aja ganti yg lain.

In Table 10, especially for matrix TP, the interpretation results using LIME show that words such as *"memuaskan"* and *"cepat"* have the most intense orange color, indicating the highest positive contribution to the prediction of the positive class. This suggests that the model is not only capable of making correct classifications but also able to recognize key words that are semantically strongly associated with positive sentiment. Next, for matrix TN, The visualization results with LIME indicate that words such as *"mls"*, *"imingi"*, and *"mahal"* exhibit the deepest shade of blue, signifying a strong contribution to the prediction of the negative class. This demonstrates that the model successfully identifies negatively charged words accurately during the classification process. For matrix FN, based on the LIME visualization, words such as *"gini"*, *"pemberitahuan"*, and *"menginstal"* are highlighted in a deep shade of blue, indicating their contribution to negative sentiment. However, words like *"dapat"*, *"ya"*, and *"pdhl"* are instead marked in orange, which influenced the model to incorrectly predict a positive sentiment. This suggests that although the overall context is negative, the presence of neutral or ambiguous words such as *"dapat"* and *"ya"* can lead to misclassification. Lastly for matrix FN, LIME visualization shows that words such as *"uninstall"*, *"kali"*, and *"ditolak"* are highlighted in a deep blue color, indicating a strong contribution to negative sentiment. However, the model classified this text as positive. This reflects a potential bias in the data labeling process, where reviews containing complaints are labeled as positive due to high star ratings. This situation highlights the strength of LIME in revealing contextual mismatches between the review content and its assigned label.

4. CONCLUSION

This study evaluates the performance of Logistic Regression, LinearSVC, and BERT models for sentiment classification on Tokopedia user reviews, using both baseline and tuned systems. All models achieved 92% of macro average across accuracy, precision, recall, and F1-score. Among them, the fine-tuned BERT model showed the most consistent performance, especially in maintaining balanced F1-scores across sentiment classes. Therefore, it was selected for further analysis using LIME for model interpretability. LIME revealed that the model often relied on frequently occurring words that associate on sentiment in the training data. However, several misclassifications occurred. For example, some incorrect label reviews that containing complaints labeled as positive due to high star ratings. These mismatches led to false positives and negatives, highlighting that model performance is influenced by label quality. The results suggest that relying solely on text content is insufficient when review content and star ratings are misaligned. Future work should consider using manually labeled data to reduce noise, or maybe using VADER to do the labelling process, applying aspect-based sentiment analysis for more granular insights, and enhancing interpretability with techniques like SHAP alongside LIME to build more robust sentiment analysis systems.

REFERENCES

[1] M. Zainottah, R. Rengga, Y. Yustian, and I. Isa, "Critical Sentiment Analysis of Tokopedia Electronic Products Using SVM-Logistic & TF-IDF Ensemble Methods," *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, vol. 4, pp. 2476–2482, Jul. 2025, doi: 10.59934/jaiea.v4i3.1194.

[2] BPS-Statistics Indonesia, *Statistik E-Commerce 2023 / E-Commerce Statistics 2023*, Jakarta, Indonesia, Publikasi No. 06300.25001, Jan. 30, 2025. [Online]. Available: https://www.bps.go.id/id/publication/2025/01/30/d52af11843aee401403ecfa6/statistik-e-commerce-2023.html

[3] M. Birjali, M. Kasri, and A. Beni-Hssane, "A comprehensive survey on sentiment analysis: Approaches, challenges and trends," *Knowl Based Syst*, vol. 226, Aug. 2021, doi: 10.1016/j.knosys.2021.107134.

[4] H. Huang, A. Asemi, and M. Mustafa, "Sentiment Analysis in E-Commerce Platforms: A Review of Current Techniques and Future Directions," *IEEE Access*, vol. 11, p. 1, Jul. 2023, doi: 10.1109/ACCESS.2023.3307308.

[5] Y. Pratama, D. Murdiansyah, and K. Lhaksmana, "Analisis Sentimen Kendaraan Listrik Pada Media Sosial Twitter Menggunakan Algoritma Logistic Regression dan Principal Component Analysis," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 7, no. 1, p. 529, Jul. 2023, doi: 10.30865/mib.v7i1.5575.

[6] M. Qorib, T. Oladunni, M. Denis, E. Ososanya, and P. Cotae, "Covid-19 Vaccine Hesitancy: Text Mining, Sentiment Analysis and Machine Learning on COVID-19 Vaccination Twitter Dataset," *Expert Syst Appl*, vol. 212, p. 118715, Jul. 2022, doi: 10.1016/j.eswa.2022.118715.

[7] N. Smairi, H. Abadlia, H. Brahim, and W. L. Chaari, "Fine-tune BERT based on Machine Learning Models For Sentiment Analysis," *Procedia Comput Sci*, vol. 246, no. C, pp. 2390–2399, Jan. 2024, doi: 10.1016/J.PROCS.2024.09.531.

[8] N. Nurdin and A. Dimas, "Explainable Artificial Intelligence (XAI) towards Model Personality in NLP task," 2021. doi: 10.12962%2Fj23378557.v7i1.a8989.

[9] M. A. Ibrahim *et al.*, "An Explainable AI Model for Hate Speech Detection on Indonesian Twitter," *CommIT (Communication and Information Technology) Journal*, vol. 16, no. 2, Jul. 2022, doi: 10.21512/commit.v16i2.8343.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1164−1173
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8072

[10] T. Thogesan, A. Nugaliyadde, and K. W. Wong, "Integration of Explainable AI Techniques with Large Language Models for Enhanced Interpretability for Sentiment Analysis," 2025. [Online]. Available: https://arxiv.org/abs/2503.11948
[11] D. S. Parmar and H. K. Saran, "Empirical Study on The Role of Explainable AI (XAI) in Improving Customer Trust in AI-Powered Products," *International Journal of Computer Trends and Technology*, vol. 73, no. 2, pp. 48–57, Feb. 2025, doi: 10.14445/22312803/ijctt-v73i2p106.
[12] Rezky Yayang Yakhamid, "Reviews of Indonesian Startup Apps on Playstore." Accessed: Jul. 07, 2025. [Online]. Available: https://www.kaggle.com/datasets/rezkyyayang/reviews-of-indonesian-app-startups-on-playstore/data?select=tokopedia.csv
[13] L. Qadrini, "Undersampling dan K-Fold Random Forest Untuk Klasifikasi Kelas Tidak Seimbang," *Building of Informatics, Technology and Science (BITS)*, vol. 4, no. 4, Jul. 2023, doi: 10.47065/bits.v4i4.3141.
[14] G. Popoola, K.-K. Abdullah, G. Shu Fuhnwi, and J. Agbaje, "Sentiment Analysis of Financial News Data using TF-IDF and Machine Learning Algorithms," Jul. 2024, pp. 1–6. doi: 10.1109/ICAIC60265.2024.10433843.
[15] M. Nasir and S. Hidayat, "Analisis Sentimen Ulasan Film Menggunakan Metode BiLSTM," *Jurnal Informatika dan Teknologi Komputer (J-ICOM)*, vol. 5, no. 2, pp. 126–132, Jul. 2024, doi: 10.55377/j-icom.v5i2.8871.
[16] P. Schober and T. Vetter, "Logistic Regression in Medical Research," *Anesth Analg*, vol. 132, pp. 365–366, Jul. 2021, doi: 10.1213/ANE.0000000000005247.
[17] N. Ashraf, R. Iqbal, S. Bano, H. M. Azeem, and S. Naz, "Enhancing MBTI Personality Prediction from Text Data with Advance Word Embedding Technique.," *VFAST Transactions on Software Engineering*, vol. 12, p. 35, Jul. 2024, doi: 10.21015/vtse.v12i3.1864.
[18] V. Chakkarwar, S. Tamane, and A. Thombre, "A Review on BERT and Its Implementation in Various NLP Tasks," pp. 112–121, 2023, doi: 10.2991/978-94-6463-136-4_12.
[19] Y. Wu, Z. Jin, C. Shi, P. Liang, and T. Zhan, "Research on the application of deep learning-based BERT model in sentiment analysis," *Applied and Computational Engineering*, vol. 71, pp. 14–20, Jul. 2024, doi: 10.54254/2755-2721/71/2024MA.
[20] M. Tripathi, "Sentiment Analysis of Nepali COVID19 Tweets Using NB, SVM AND LSTM," *Journal of Artificial Intelligence and Capsule Networks*, vol. 3, no. 3, pp. 151–168, Jul. 2021, doi: 10.36548/jaicn.2021.3.001.
[21] S. Rao, S. Mehta, S. Kulkarni, H. Dalvi, N. Katre, and M. Narvekar, "A Study of LIME and SHAP Model Explainers for Autonomous Disease Predictions," in *2022 IEEE Bombay Section Signature Conference (IBSSC)*, Mumbai, India, Dec. 2022, pp. 1-6, doi: 10.1109/IBSSC56953.2022.10037324.
[22] C. V Roberts, E. Elahi, and A. Chandrashekar, "On the Bias-Variance Characteristics of LIME and SHAP in High Sparsity Movie Recommendation Explanation Tasks," 2022. [Online]. Available: https://arxiv.org/abs/2206.04784
[23] A. Salih *et al.*, "A Perspective on Explainable Artificial Intelligence Methods: SHAP and LIME," *Advanced Intelligent Systems*, vol. 7, Jul. 2024, doi: 10.1002/aisy.202400304.