JASMINE
Journal of Intelligent Systems and Machine Learning
https://journals.telkomuniversity.ac.id/jasmine

Vol. 1, Issue 1, May 2026, pp: 57-67

Comparison of SVM, Naive Bayes, and Logistic Regression for LinkedIn Reviews Sentiment Analysis

Nadhilah Hazrati[1], Alam Rahmatulloh[1,*]
[1]Department of Informatics, Faculty of Engineering, Siliwangi University, Tasikmalaya, Indonesia

*Author to whom any correspondence should be addressed.
E-mail: alam@unsil.ac.id

Received: December 1, 2025
Accepted for publication: January 13, 2026
Published: May 11, 2026

ABSTRACT
The rapid development of digital technology has transformed how people search for jobs, with LinkedIn emerging as the largest professional social media platform. Users express opinions about the application through Google Play Store reviews, reflecting both positive and negative sentiments. This study analyzes 5,000 LinkedIn reviews in the Indonesian context using sentiment analysis, comparing three classification algorithms: Support Vector Machine (SVM), Naïve Bayes, and Logistic Regression. The research process includes data collection, text preprocessing, feature extraction, and model evaluation using accuracy, precision, recall, and F1-score metrics. The results show that all models perform effectively, with Logistic Regression achieving the highest performance (accuracy 88.53%, precision 94% for negative and 83% for positive reviews, recall 84% for negative and 94% for positive reviews). These findings demonstrate the study's contribution in providing a large-scale, context-specific analysis of user sentiments, offering insights to improve the LinkedIn application.

Keywords: Feature Extraction; Model Evaluation; Text Preprocessing; User Perceptions; User Review.

I. Introduction

In the digital era, the job search process has experienced a major transformation. In the past, people relied on printed media such as newspapers, pamphlets, or word-of-mouth information to find job vacancies. Now, everything can be accessed more easily through the internet. One of the most popular platforms for job searching and professional networking is LinkedIn. Founded in 2002 by Reid Hoffman and launched in 2003, LinkedIn is a professional social media platform that enables users to build professional networks, making it an essential tool for career management and recruitment [1].

The LinkedIn platform serves as one of the best ways for users to build professional networks for both job searching and recruiting skilled professionals. LinkedIn users can share content related to their achievements and professional skills on their personal profiles. In addition, users can set the "Open to Work" status on their profiles, allowing recruiters to view this status and making it easier for them to find potential job seekers [1].

The development of digital technologies such as LinkedIn also supports the achievement of the Sustainable Development Goals (SDGs), especially Goal 8 "Decent Work and Economic Growth" [2]. This platform plays a role in creating a more open and inclusive work ecosystem, expanding access to decent employment opportunities, and increasing efficiency in the recruitment process. Thus, the use of digital technology in the workplace not only makes it easier for individuals to find work but also contributes to sustainable economic growth and equal employment opportunities across all levels of society [3].

User reviews on the Google Play Store reflect a wide range of user opinions, including complaints, criticism, praise, and suggestions regarding an application. Similar patterns are also observed in user reviews of the LinkedIn application. In this study, LinkedIn user reviews collected from the Google Play Store are analyzed with the aim of categorizing the expressed sentiments and evaluating the performance of three classification methods. The analysis focuses on identifying whether each review conveys a positive or negative sentiment. Based on this objective, this study compares the performance of Support Vector Machine (SVM), Naïve Bayes, and Logistic Regression in classifying the sentiment of LinkedIn user reviews.

The comparison of these three methods is important because each algorithm has different ways of working and characteristics in processing text. SVM is known to be strong in handling high-dimensional data such as text, Naïve Bayes often gives good results on text data due to its simplicity, while Logistic Regression is widely used as a stable baseline model. By comparing the three, this research can determine the most effective method for analyzing the sentiment of LinkedIn user reviews, thereby producing more accurate interpretations and supporting data-driven decision making.

II. Related work

Several previous studies have applied machine learning and deep learning techniques for sentiment analysis of LinkedIn reviews. Study [4] utilized the Support Vector Machine (SVM) method with TF-IDF and achieved 90% accuracy, although no comparison with other models was conducted. Research [4] compared the Naïve Bayes and SVM algorithms, showing that SVM performed better with 90% accuracy compared to 88% for Naïve Bayes however, their dataset was limited to 2,000 reviews and did not include neutral sentiments. Study [1] employed the Naïve Bayes Classifier and achieved 84% accuracy, with most reviews categorized as negative. Meanwhile, research [5] implemented the k-Nearest Neighbors (k-NN) algorithm on Android game reviews, obtaining 75.50% accuracy and an AUC of 0.825. More recently, study [6] combined Lexicon-based labeling with a Long Short-Term Memory (LSTM) model, achieving 88.21% accuracy and demonstrating improved contextual understanding. Additionally, a previous study proposed a hybrid evolutionary SVM-based approach for sentiment analysis under imbalanced data distributions, highlighting the robustness of SVM in high-dimensional text data while still focusing on binary sentiment classification [7]. Another study applied a Naïve Bayes classifier to agritech-related social media reviews, showing effective classification of positive and negative sentiments in domain-specific text data [8]. Overall, these studies highlight the effectiveness of SVM and LSTM-based models for sentiment analysis. However, they also reveal several limitations, including the use of relatively small datasets, the exclusion of neutral sentiment classification, and limited comparisons across different algorithms.

III. Material and Methods

A. System Architecture

Figure 1 represents the system architecture developed in this project, which consists of several main integrated stages designed to perform sentiment classification on user review texts [9].

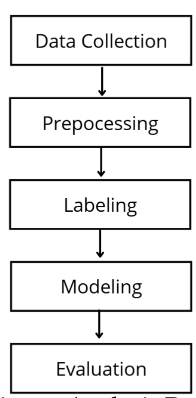

Figure 1. Sentiment Analysis Process Flow.

The process begins with the data collection stage, where user review data from the Google Play Store is gathered. Data collection through web scraping starts by accessing the LinkedIn application page on the Google Play Store and collecting user reviews from 2023 to 2025. The collected data includes users opinions on various aspects such as ease of use, available features, service quality, as well as technical issues or updates that may affect user experience. As a key contribution, this research utilizes a relatively large dataset consisting of 5,000 user reviews, which allows for a more robust and representative sentiment analysis. In addition, the study specifically focuses on the Indonesian context of LinkedIn users, providing insights into user perceptions and experiences that are rarely explored in previous studies. These contributions enable a more comprehensive understanding of user sentiment toward the LinkedIn application within a specific regional and temporal scope. After data collection, the text data undergo a preprocessing stage that includes text cleaning, case folding, tokenization, stopword removal, and stemming. These standard natural language processing steps are applied to normalize the data and reduce noise prior to sentiment classification, following established practices in previous studies [10].

All of these stages are carried out to produce clean and well-prepared data for the labeling process. Next, the pre-processed data is assigned sentiment labels, namely positive or negative. The labeled data is then converted into a numerical form using the TF-IDF (Term Frequency–Inverse Document Frequency) method, which represents the text as a numerical vector based on the frequency of word occurrences [11]. After vectorization, the data is divided into two parts: training data and testing data. The positive class consists of 1,662 entries, while the negative class contains 1,940 entries. The use of only two classes positive and negative without including a neutral class was intended to simplify the sentiment analysis process and address data distribution imbalance. In the original dataset, the score values ranged from 1 to 5, where a score of 3 was often considered neutral [12]. However, if the proportion of neutral scores is too large, the analysis model may become biased toward the neutral class and overlook the emotional nuances present in other data. The training data is then used to train the classification model using three machine learning algorithms: Support Vector Machine (SVM), Naïve Bayes, and Logistic Regression.

The trained models are then evaluated using the testing data to measure the performance of each algorithm. The evaluation is carried out based on accuracy and other evaluation metrics to determine how well each model predicts sentiment. After the evaluation process, a comparison among the models is conducted to identify which algorithm provides the best results. The final stage of this architecture is to present the classification and evaluation results as the system's output. The entire process is designed to run efficiently on the Google Colab platform, utilizing lightweight software and hardware resources while still being capable of accurately and optimally handling medium-sized datasets.

1) *Support Vector Machine (SVM)*

Mathematical Formulation of the Support Vector Machine (SVM) Classification Methods as shown in Equation 1, where n is total number of LinkedIn reviews used in the training dataset, $x_i \in \mathbb{R}^d$ refers to feature vector of the i-th LinkedIn review represented in a d-dimensional space, obtained from text feature extraction methods such as TF-IDF or Bag-of-Words, d refers to number of features generated from the text representation, $y_i \in \{-1, +1\}$ refers sentiment label of the i-th LinkedIn review, where -1 denotes negative sentiment and $+1$ denotes positive sentiment.

$$\{(x_i, y_i)\}_{i=1}^{n}, x_i \in \mathbb{R}^d, \ y_i \in \{-1, +1\} \tag{1}$$

The objective function of the SVM formula is shown in Equation 2, where w refers a weight vector defining the separating hyperplane between positive and negative sentiments, b refers bias term of the hyperplane, $\parallel w \parallel^2$ refers squared norm of the weight vector, which is minimized to maximize the margin between classes, ξ_i refers slack variable that allows misclassification of difficult-to-separate LinkedIn reviews, C refers regularization parameter controlling the trade-off between margin maximization and classification error.

$$\min_{w,b,\xi} \frac{1}{2} \parallel w \parallel^2 + C \sum_{i=1}^{n} \xi_i \tag{2}$$

The constraint of SVM is shown in Equation 3, where $w \cdot x_i$ refers dot product between the weight vector and the feature vector of the i-th review. The first constraint ensures correct classification with a margin, while allowing violations through ξ_i and the second constraint enforces non-negativity of the slack variables.

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i \tag{3}$$
$$\xi_i \geq 0$$

2) *Naïve Bayes*

The posterior probability formula of Naïve Bayes is shown in Equation 4, where $x = (x_1, x_2, \ldots, x_d)$ refers to a feature vector representing a LinkedIn review, x_j refers to the frequency of the j-th word in the review, w_j refers the j-th word in the vocabulary, d is size of the vocabulary, y is sentiment class (positive or negative), $P(y)$ refers to prior probability of class y, $P(w_j \mid y)$ refers conditional probability of word, w_j given sentiment class y, and $P(y \mid x)$ refers to posterior probability that a review belongs to class y.

$$P(y \mid x) \propto P(y) \prod_{j=1}^{d} P(w_j \mid y)^{x_j} \tag{4}$$

The classification rule formula of Naïve Bayes is shown in Equation 5, where $\hat{y}$ refers to the predicted sentiment label of a LinkedIn review and Y refers to a set of sentiment classes. The logarithmic form is used to improve numerical stability.

$$\hat{y} = \arg\max_{y \in Y} \left[\log P(y) + \sum_{j=1}^{d} x_j \log P(w_j \mid y)\right] \tag{5}$$

3) Logistic Regression

The Probabilistic model formula of Logistic Regression is shown in Equation 6, where $x \in \mathbb{R}^d$ refers to a feature vector of a LinkedIn review obtained from TF-IDF or Bag-of-Words, w refers to weight vector representing the contribution of each textual feature, b refers to bias term, and $P(y = 1 \mid x)$: probability that a LinkedIn review expresses positive sentiment.

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-(w \cdot x + b)}} \tag{6}$$

The loss function formula is shown in Equation 7, where $y_i \in \{0,1\}$ refers to true sentiment label of the i-th LinkedIn review, $\hat{y}_i$ refers to a predicted probability of positive sentiment for the i-th review, $\mathcal{L}(w)$: cross-entropy loss function measuring classification error, n: total number of training reviews.

$$\mathcal{L}(w) = -\sum_{i=1}^{n} [y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{7}$$

The optimization formula of Logistic Regression is shown in Equation 8, where w refers to weight vector of the logistic regression model representing the contribution of textual features extracted from LinkedIn reviews, and $\mathcal{L}$(w) is cross-entropy loss function measuring the classification error in LinkedIn reviews sentiment analysis.

$$\min_{w} \ \mathcal{L}(w) \tag{8}$$

IV. Results and Discussion

A. Collection and Preprocessing Data

Data collection through web scraping began by accessing the LinkedIn app page on Google Play Store and collecting 5,000 user reviews from 2023 to 2025. The summary of the collected data is presented in Table 1, which provides an overview of the total number of reviews retrieved during the data collection process. According to Figure 2, the flow of data is processed through a series of steps prior to analysis.

Table 1. Data Collection.

No	Text
1	*aplikasi maksa update*
2	*hanya bertulisakan "Mari lakukan pemeriksaan keamanan cepat" semenjak saya pasang authenticator app untuk keamanan ekstra setelah di hack sebelumnya. sekarang menjadi stuck bertuliskan LinkedIn ©2023. tolong perbaiki akun atas email saya*
3	*lowongan yg di terbitkan yg kadaluarsa dan sangat sulit cara melamar*
4	*App bodo, tbtb keluar, pas mau login lagi malah stuck di pemeriksaan keamanan. tolong lah ini developer applikasinya dibetulin, acuh amat sama applikasi sendiri*
5	*terimaksih sudh mebantu kami*
…	…
5000	*Waktu pertama kali install, AlhamduLillah ndak ada keribetan pas mau login, malah ada fitur login dengan akun google, praktis & selesai. Device: vivo 2019*

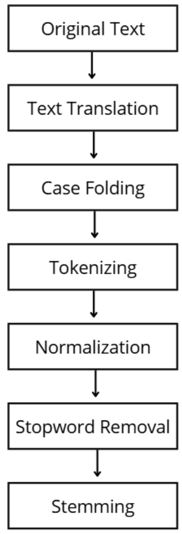

Figure 2. Preprocessing Data Flow.

B. Text Translation

Text translation was performed because many reviews in the dataset were written in English. To ensure consistent processing, all English reviews were translated into Indonesian. The translation process was carried out using the *deep-translator* library. To verify translation quality, a manual inspection was conducted on a representative sample by comparing the original English texts with their Indonesian translations. The evaluation focused on sentiment-bearing terms and overall polarity, and the results showed that sentiment polarity was preserved in the vast majority of cases, with no significant changes observed, and the details of this step are summarized in Table 2.

Table 2. Text Translation.

No	Original Text	Translation Results
1	*naah, login page error*	*Naah, kesalahan halaman login*
2	*Nice*	*Bagus*
3	*very good*	*Sangat Bagus*
4	*Good*	*Bagus*
5	*very good for conection and karier*	*sangat bagus untuk conection dan karier*

C. Case Folding

Case folding is used to convert all text by changing all uppercase letters to lowercase letters[13]. The main purpose of this process is to standardize the text, so that words such as "Data", 'data', and "DATA" are treated as the same entity. Without case folding, text analysis models or systems may consider these three forms as different words, which can cause redundancy and interfere with the accuracy of the analysis. By performing case folding, data complexity is reduced, data consistency is improved, and subsequent processes such as tokenization, normalization, and model creation can be carried out more effectively. The implementation of the case folding step is summarized in Table 3.

Table 3. Case Folding.

No	Case Folding
1	*aplikasi maksa update*
2	*hanya bertuliskan "mari lakukan pemeriksaan keamanan cepat" semenjak saya pasang authenticator app untuk keamanan ekstra setelah di hack sebelumnya. sekarang menjadi stuck bertuliskan linkedin ©2023. tolong perbaiki akun atas email saya*
3	*lowongan yg di terbitkan yg kadaluarsa dan sangat sulit cara melamar*
4	*app bodo, tbtb keluar, pas mau login lagi malah stuck di pemeriksaan keamanan. tolong lah ini developer applikasinya dibetulin, acuh amat sama applikasi sendiri*
5	*terimaksih sudah mebantu kami*

D. Tokenizing

Tokenizing is the process of breaking long texts into smaller parts called "tokens"[14]. These small units can be words, phrases, or punctuation marks, depending on how the breakdown is done. For example, words such as "The LinkedIn app is very useful" will consist of words such as "app", "LinkedIn", 'very', and "useful". Because the text data is now structured into elements, tokenization facilitates the analysis process. The implementation of this step is presented in Table 4.

Table 4. Tokenizing.

No	Original Text	Tokenizing
1	*aplikasi maksa update*	*['aplikasi', 'maksa', 'update']*
2	*hanya bertuliskan mari lakukan pemeriksaan keamanan cepat semenjak saya pasang authenticator app untuk keamanan ekstra setelah di hack sebelumnya sekarang menjadi stuck bertuliskan linkedin tolong perbaiki akun atas email saya*	*['hanya', 'bertuliskan', 'mari', 'lakukan', 'pemeriksaan', 'keamanan', 'cepat', 'semenjak', 'saya', 'pasang', 'authenticator', 'app', 'untuk', 'keamanan', 'ekstra', 'setelah', 'di', 'hack', 'sebelumnya', 'sekarang', 'menjadi', 'stuck', 'bertuliskan', 'linkedin', 'tolong', 'perbaiki', 'akun', 'atas', 'email', 'saya']*
3	*lowongan yg di terbitkan yg kadaluarsa dan sangat sulit cara melamar*	*['lowongan', 'yg', 'di', 'terbitkan', 'yg', 'kadaluarsa', 'dan', 'sangat', 'sulit', 'cara', 'melamar']*
4	*app bodo tbtb keluar pas mau login lagi malah stuck di pemeriksaan keamanan tolong lah ini developer applikasinya dibetulin acuh amat sama applikasi sendiri*	*['app', 'bodo', 'tbtb', 'keluar', 'pas', 'mau', 'login', 'lagi', 'malah', 'stuck', 'di', 'pemeriksaan', 'keamanan', 'tolong', 'lah', 'ini', 'developer', 'applikasinya', 'dibetulin', 'acuh', 'amat', 'sama', 'applikasi', 'sendiri']*
5	*terimaksih sudh mebantu kami*	*['terimaksih', 'sudh', 'mebantu', 'kami']*

E. Normalization

This normalization uses a dictionary called normalization_dict that contains pairs of non-standard words as keys and their standard equivalents as values. For example, the words *"gak"*, *"ga"*, and *"tdk"* are normalized to *"tidak"*, the word *"bgt"* to *"banget"*, and the word *'yg'* to *"yang"*. In addition, there are also expressions such as *"sih"*, *"deh"*, *"dong"*, *'lah'*, and *"kok"* that are removed by changing them to empty strings. The list of normalized terms and their standardized forms is presented in Table 5.

Table 5. Normalization.

No	Text	Normalization Results
1	*aplikasi maksa update*	*aplikasi maksa update*
2	*hanya bertuliskan mari lakukan pemeriksaan keamanan cepat semenjak saya pasang authenticator app untuk keamanan ekstra setelah di hack sebelumnya sekarang menjadi stuck bertuliskan linkedin tolong perbaiki akun atas email saya*	*hanya bertuliskan mari lakukan pemeriksaan keamanan cepat semenjak saya pasang authenticator app untuk keamanan ekstra setelah di hack sebelumnya sekarang menjadi stuck bertuliskan linkedin tolong perbaiki akun atas email saya*
3	*lowongan yg di terbitkan yg kadaluarsa dan sangat sulit cara melamar*	*lowongan yang di terbitkan yang kadaluarsa dan sangat sulit cara melamar*
4	*app bodo tbtb keluar pas mau login lagi malah stuck di pemeriksaan keamanan tolong lah ini applikasinya dibetulin acuh amat sama applikasi sendiri*	*app bodo tbtb keluar pas mau login lagi malah stuck di pemeriksaan keamanan tolong ini applikasinya dibetulin acuh amat sama applikasi sendiri*
5	*terimaksih sudh mebantu kami*	*terimaksih sudh mebantu kami*

F. Stopword Removal

At this stage, words that are considered irrelevant or do not add meaning are removed from the text, such as conjunctions (like "and," 'or') or personal pronouns (like "I," "he"). These stop words are words that frequently appear in sentences but do not provide information necessary for analysis. The list of stop words used in this process is presented in Table 6.

G. Stemming

Stemming is the process of converting words to their base form by removing unnecessary suffixes or affixes [15]. Stemming uses a literary library. For example, the term *"bertuliskan"* will be converted to its original form, which is *"tulis"*. As shown in Table 7, stemming is used to facilitate the analysis of words that have the same meaning.

Table 6. Stopword Removal.

No	Text	Stopword Removal
1	*aplikasi maksa update*	*aplikasi maksa update*
2	*hanya bertuliskan mari lakukan pemeriksaan keamanan cepat semenjak saya pasang authenticator app untuk keamanan ekstra setelah di hack sebelumnya sekarang menjadi stuck bertuliskan linkedin tolong perbaiki akun atas email saya*	*bertuliskan mari lakukan pemeriksaan keamanan cepat semenjak pasang authenticator app keamanan ekstra hack stuck bertuliskan linkedin tolong perbaiki akun email*
3	*lowongan yang di terbitkan yang kadaluarsa dan sangat sulit cara melamar*	*lowongan terbitkan kadaluarsa sulit melamar*
4	*app bodo tbtb keluar pas mau login lagi malah stuck di pemeriksaan keamanan tolong ini developer applikasinya dibetulin acuh amat sama applikasi sendiri*	*app bodo tbtb pas login stuck pemeriksaan keamanan tolong developer applikasinya dibetulin acuh applikasi*
5	*terimaksih sudh mebantu kami*	*terimaksih sudh mebantu*

Table 7. Stemming.

No	Text	Stemming
1	*aplikasi maksa update*	*aplikasi maksa update*
2	*bertuliskan mari lakukan pemeriksaan keamanan cepat semenjak pasang authenticator app keamanan ekstra hack stuck bertuliskan linkedin tolong perbaiki akun email*	*tulis mari laku periksa aman cepat semenjak pasang authenticator app aman ekstra hack stuck tulis linkedin tolong baik akun email*
3	*lowongan terbitkan kadaluarsa sulit melamar*	*lowong terbit kadaluarsa sulit lamar*
4	*app bodo tbtb pas login stuck pemeriksaan keamanan tolong developer applikasinya dibetulin acuh applikasi*	*app bodo tbtb pas login stuck periksa aman tolong developer applikasinya dibetulin acuh applikasi*
5	*terimaksih sudh mebantu*	*terimaksih sudh mebantu*

H. Data Splitting

Data splitting is performed to prepare data before it is used in the machine learning model training and testing process. First, the feature used is the content_translation column, while the target label is the score column. After that, the dataset is divided into two parts, namely training data and testing data, with a proportion of 70% for training and 30% for testing, using the train_test_split function from the sklearn library.

1) Labeling

After performing the text preprocessing steps, data labeling was conducted. As shown in Table 8, the classes generated in the labeling consisted of two categories, namely Positive and Negative. There were 1,662 Positive instances and 1,940 Negative instances. The use of only two classes, Positive and Negative, without including a Neutral class, was intended to simplify the sentiment analysis process. This approach allows the model to focus on clear sentiment polarity and improves classification effectiveness. In the initial dataset, the score values ranged from 1 to 5, where a score of 3 was commonly considered Neutral. If the proportion of neutral scores is too large, it may lead to data imbalance and cause the model to be biased toward the neutral class, thereby reducing its ability to capture the emotional nuances present in positive and negative sentiments.

Table 8. Distribution of Positive and Negative Scores.

Positive	Negative
53,9%	46,1%

2) Modeling

a. Support Vector Machine

Table 9 presents the performance of the Support Vector Machine (SVM) model applied in this study for sentiment classification of LinkedIn reviews. The SVM model is formulated based on the training dataset defined in Equation (1), where each LinkedIn review is represented as a feature vector with a corresponding sentiment label. The optimization process follows the objective function in Equation (2), which aims to maximize the margin between positive and negative sentiment classes while allowing limited misclassification through slack variables. The classification boundaries are enforced by the constraints defined in Equation (3).

Based on this formulation, the SVM model achieved an accuracy of 87.79%, indicating that nearly 88% of the LinkedIn reviews were correctly classified into positive or negative categories. This result demonstrates that the margin-based optimization defined in Equations (2) and (3) enables the model to reliably distinguish between the two sentiment classes. In terms of precision, the model achieved 93% for the negative class, indicating a low number of false positive predictions, while the precision for the positive class reached 83%, suggesting a slightly higher tendency toward false positives. Furthermore, the recall values show that the SVM model successfully identified 83% of actual negative reviews and 93% of positive reviews, reflecting the effectiveness of the constraint-based classification mechanism in Equation (3).

Table 9. Result of the SVM model.

	Precision	Recall	F1- Score	Support
Negative	0.93	0.83	0.88	587
Positive	0.83	0.93	0.87	494
Accuracy	-	-	0.88	1081
Macro avg	0.88	0.88	0.88	1081
Weighted avg	0.88	0.88	0.88	1081

b. Naïve Bayes

Table 10 presents the performance of the Multinomial Naïve Bayes classification model applied in this study. The model estimates the posterior probability of each sentiment class based on the probabilistic formulation defined in Equation (4), where the sentiment label is inferred from the distribution of words in each LinkedIn review. The final sentiment classification is determined using the decision rule in Equation (5), which selects the class with the maximum log-posterior probability.

Table 10. Result of the Naïve Baiyes Model.

	Precision	Recall	F1- Score	Support
Negative	0.80	0.93	0.86	587
Positive	0.89	0.72	0.80	494
Accuracy	-	-	0.83	1081
Macro avg	0.85	0.83	0.83	1081
Weighted avg	0.84	0.83	0.83	1081

Based on this probabilistic framework, the classification model achieved an accuracy of 83.44%, indicating that approximately 83 out of 100 predictions were correctly classified. For the negative class, the model obtained a precision of 80% and a recall of 93%, resulting in an F1-score of 0.86, which shows that the model is highly effective in identifying negative reviews with minimal false negatives. In contrast, the positive class achieved a higher precision of 89% but a lower recall of 72%, indicating that while positive predictions are generally accurate, some positive reviews were not successfully detected. Consequently, the F1-score for the positive class decreased to 0.80, reflecting the trade-off inherent in the probabilistic assumptions of the Naïve Bayes model defined in Equations (4) and (5).

c. Logistic Regression

Table 11 presents the performance evaluation of the Logistic Regression classification model used in this study. The model estimates the probability of a LinkedIn review expressing positive sentiment using the probabilistic formulation defined in Equation (6). The learning process minimizes the cross-entropy loss function described in Equation (7) through the optimization objective shown in Equation (8), ensuring an optimal set of model parameters for sentiment classification.

Table 11. Result of the Logistic Regression.

	Precision	Recall	F1- Score	Support
Negative	0.94	0.84	0.89	587
Positive	0.83	0.94	0.88	494
Accuracy	-	-	0.89	1081
Macro avg	0.89	0.89	0.89	1081
Weighted avg	0.89	0.89	0.89	1081

Based on this formulation, the model achieved an overall accuracy of 89%, indicating that nearly 89 out of every 100 predictions were classified correctly. For the negative class, the model obtained a high precision of 94%, showing that most reviews predicted as negative were indeed negative. Meanwhile, the positive class achieved a precision of 83%, indicating reliable performance in identifying positive sentiment, although slightly lower than that of the negative class. In terms of recall, the model demonstrated strong capability, achieving 84% for the negative class and 94% for the positive class, which indicates excellent sensitivity in detecting actual positive reviews. The resulting F1-scores of 0.89 for the negative class and 0.88 for the positive class confirm that the Logistic Regression model, as defined in Equations (6)–(8), provides a strong balance between precision and recall for LinkedIn sentiment analysis.

I. Evaluation

Based on the figure 3 test results shown in the graph below, three classification models, namely Support Vector Machine (SVM), Naive Bayes, and Logistic Regression, showed satisfactory performance with accuracy and F1 scores above 0.8. However, there were differences in the accuracy levels of each model. The Logistic Regression model ranks first with an accuracy and F1 score of around 0.89. These results show that this model is capable of providing accurate predictions while maintaining a balance between precision and recall. This means that this model can recognize patterns in the data well without favoring a particular class. Logistic Regression outperforms SVM and Naive Bayes because it effectively models linear relationships between features and sentiment probabilities, does not assume feature independence (unlike Naive Bayes), and provides calibrated probability estimates, which help handle imbalanced datasets. The SVM model ranks second with an accuracy and F1 score between 0.87 and 0.88. Although slightly lower than Logistic Regression, this model still shows stable and consistent performance in classifying data. The Naive Bayes model has an accuracy value of around 0.83 and an F1 score of around 0.80, which is the lowest result compared to the other two models. This difference in values shows that the Naive Bayes model is less effective in dealing with data that has complex feature variations. Overall, it can be concluded that Logistic Regression provides the most accurate results in this study, followed by SVM, while Naive Bayes has the lowest performance.

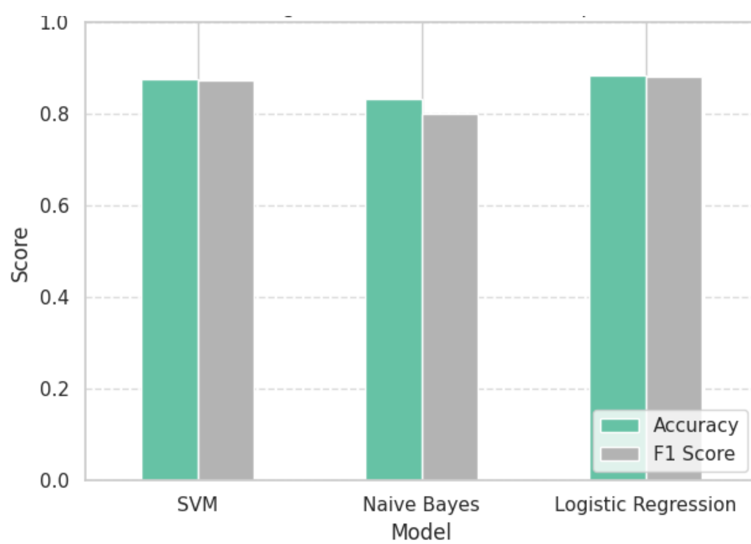

Figure 3. Comparison of Accuracy for SVM, Naive Bayes, and Logistic Regression.

V. Conclusion

Logistic Regression showed the best performance with the highest accuracy and F1-Score, approaching 0.89 to 0.90, indicating a strong balance between precision and recall for both classes. Support Vector Machine (SVM) also provided excellent results with almost equivalent values, although slightly lower. In contrast, Naive Bayes had the lowest performance with lower accuracy and F1-Score, indicating the limitations of this method in handling data imbalance. For further development, it is recommended to add an automatic bilingual feature that can translate user comments bidirectionally between English and Indonesian. This feature will facilitate cross-language understanding and improve the accuracy of sentiment analysis. In addition, the integration of multilingual BERT models such as mBERT or XLM-

RoBERTa is highly recommended to capture complex linguistic contexts and nuances, including sarcasm and irony, in greater depth. With the implementation of bilingual features and the natural language processing capabilities of the BERT model, this project can evolve into a more intelligent, adaptive, and accurate multilingual sentiment analysis system.

Conflicts of Interest

The authors declare no conflicts of interest.

Author Contributions Statement

Author 1 contributed to the conception, design, data collection, analysis, and writing of the manuscript as part of their research project. Author 2 served as the supervising lecturer, providing guidance, feedback, and critical revisions throughout the research and manuscript preparation process. All authors have read and approved the final version of the manuscript for publication.

Acknowledgment

The author would like to express sincere gratitude to the supervising lecturer for the guidance and valuable input provided during this research. Thanks, are also extended to all lecturers in the Department of Informatics, Faculty of Engineering, Siliwangi University, for the knowledge, support, and encouragement provided throughout the study process, enabling this research to be completed successfully.
The author would also like to express his deep gratitude to the Department of Informatics, Faculty of Engineering, Siliwangi University, for the adequate academic facilities and learning environment that supported the process of scientific development and research.
Last but not least, the author would also like to thank his colleagues and family for their support, prayers, and encouragement throughout the process of preparing this research.

References

[1] L. Kusneti, A. Ratu, and A. Wijaya, ‘Analisis Sentimen Ulasan Aplikasi Linkedin dalam Google Play Store dengan Model Naïve Bayes’, *Jurnal Teknologi Informasi*, vol. 4, no. 2, 2023, doi: 10.46576/djtechno.

[2] A. Raihan, ‘A review of the digitalization of the small and medium enterprises (SMEs) toward sustainability’, *Global Sustainability Research*, vol. 3, no. 2, pp. 1–16, Mar. 2024, doi: 10.56556/gssr.v3i2.695.

[3] F. S. Gilbert and W. Setiyani, ‘GLOBAL KOMUNIKA Pemberdayaan UMKM Sebagai Penggerak Pemulihan Ekonomi Nasional Melalui Optimalisasi Digital 4.0’, 2022.

[4] P. Silitonga, D. El, R. Purba, F. Ojak, and I. Pardede, ‘Analisis Sentimen Ulasan Pengguna Linkedin di Google Playstore Menggunakan Metode Support Vecotre Machine (SVM)’, 2024.

[5] S. Sahara, R. Agung Permana, U. Bina Sarana Informatika, and S. Antar Bangsa, ‘METODE KNN PADA SENTIMENT ANALISIS REVIEW PRODUK GAME ANDROID’, Online, 2022. [Online]. Available: www.zdnet.com

[6] R. Putra Kurniawan and S. Wahyu Iriananda, ‘Analisis Sentimen Ulasan Aplikasi Linkedin Berbasis Lexicon dan Long Short-Term Memory (LSTM)’, 2025.

[7] R. Obiedat *et al.*, ‘Sentiment Analysis of Customers’ Reviews Using a Hybrid Evolutionary SVM-Based Approach in an Imbalanced Data Distribution’, *IEEE Access*, vol. 10, pp. 22260–22273, 2022, doi: 10.1109/ACCESS.2022.3149482.

[8] N. Kewsuwun and S. Kajornkasirat, ‘A sentiment analysis model of agritech startup on Facebook comments using naive Bayes classifier’, *International Journal of Electrical and Computer Engineering*, vol. 12, no. 3, pp. 2829–2838, Jun. 2022, doi: 10.11591/ijece.v12i3.pp2829-2838.

[9] R. Hasanah, Z. Hasanah, S. Ardi Wijaya, D. Abdennasser, and A.-N. Sharkawy, ‘Play Store Data Scrapping and Preprocessing done as Sentiment Analysis Material’, 2025. [Online]. Available: https://journal.abhinaya.co.id/index.php/IJMST

[10] I. Muslim *et al.*, ‘Implementasi Text Summarization Pada Review Aplikasi Digital Library System Menggunakan Metode Maximum Marginal Relevance’, 2024.

[11] P. R. Sari *et al.*, ‘Sistemasi: Jurnal Sistem Informasi Comparison of Naive Bayes and SVM Algorithms for Sentiment Analysis of PUBG Mobile on Google Play Store’. [Online]. Available: http://sistemasi.ftik.unisi.ac.id

[12] M. Haris, A. Suharso, and E. Haodudin Nurkifli, ‘Analisis Sentimen pada Game EFootball di Google Play Store Menggunakan Algoritma IndoBERT’, 2024.

[13] G. Van Transco and A. Herliana, 'Journal of Artificial Intelligence and Engineering Applications Public Sentiment Analysis on Police Service Satisfaction Using Twitter Dataset Based on NLP and SVM', 2025. [Online]. Available: https://ioinformatic.org/

[14] S. A. Mahira, I. Sukoco, C. S. Barkah, N. Jamil, A. Novel, and J. A. Bisnis, 'Teknologi Artificial Intelligence dalam Analisis Sentimen: Studi Literatur Pada Perusahaan KATA.AI', 2023.

[15] R. A. Fauzianto and Supatman, 'Analisis Sentimen Opini Masyarakat Terhadap Tech Winter Pada Twitter Menggunakan Natural Language Processing', *Jurnal Syntax Admiration*, vol. 3, no. 9, pp. 1577–1585, Sep. 2023, doi: 10.46799/jsa.v3i9.909.