Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871

Vol. 6, No. 5, October 2025, Page. 3033-3050
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5238

Improving the Performance of Machine Learning Classifiers in Sentiment Analysis of Jenius Application Using Latent Dirichlet Allocation in Text Preprocessing

Vincentius Riandaru Prasetyo*[1], Njoto Benarkah[2], Bayu Aji Hamengku Rahmad[3]

[1,2,3] Department of Informatics Engineering, University of Surabaya, Surabaya, Indonesia

Email: [1]vincent@staff.ubaya.ac.id

Received : Aug 5, 2025; Revised : Aug 24, 2025; Accepted : Aug 27, 2025; Published : Oct 16, 2025

Abstract

Sentiment analysis aims to classify a person's opinion into a specific sentiment, such as positive or negative. The choice of preprocessing used can influence the performance of a sentiment analysis model. The Latent Dirichlet Allocation (LDA) method, commonly used for topic modelling, can be employed as an additional preprocessing step to identify relevant words associated with a particular sentiment label. This study aims to assess whether the LDA method, implemented in the preprocessing stage, can enhance the performance of machine learning models, including Naïve Bayes, Decision Tree, KNN, Logistic Regression, and SVM. This study utilized a dataset comprising 1,800 reviews, with 900 labelled as positive and 900 as negative. Words with an LDA score of at least 0.15 were given additional weight in the TF-IDF stage before model training. After the model was developed, evaluation was carried out by calculating accuracy, precision, recall, and F1-score. The use of LDA in preprocessing improved the performance of all classification models by 1-3% across most evaluation metrics. Specifically, the Logistic Regression model achieved the best performance, followed by SVM and KNN. This performance improvement is aligned with the use of LDA to reduce semantic noise and improve feature representation. Furthermore, this research is also helpful for monitoring customer opinions in the digital banking sector, enabling the rapid and accurate identification of priority issues. Further research could explore the comparison of performance with other topic modelling and feature extraction methods, as well as expanding the dataset and utilizing multiclass models.

Keywords : *Digital Banking, Jenius Application, Latent Dirichlet Allocation, Machine Learning Classifiers, Sentiment Analysis, Text Preprocessing.*

1. INTRODUCTION

The modern era's surge in smartphone users has spurred app developers to create applications that cater to various user needs. One such standout is the Jenius app, a digital banking application in Indonesia. Developed and launched by BTPN, the Jenius app offers a unique feature: users can use the app without needing a BTPN account. This app, with its transformative potential, not only promises a more convenient and efficient way of banking but also serves as a beacon of hope in the digital era, inspiring optimism about the future of digital banking [1]. It enables users to perform a range of financial activities, including opening and closing accounts, as well as making transactions, all within a single application, thereby eliminating the need to visit a physical bank. The Jenius app, with its transformative potential, sparks inspiration and optimism about the future of digital banking [2].

Users often express the opinions about the Jenius app on X and the Google Play Store. These platforms, unlike Instagram or Facebook, are more text centric [3]. The Jenius app, with over 10 million users and 194,000 reviews, has an average rating of 3.6. While this rating is lower than that of some other digital banking apps with similar user numbers, the sheer volume of reviews for the Jenius app suggests that potential users may struggle to accurately assess its quality and service level without text

Vol. 6, No. 5, October 2025, Page. 3033-3050
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5238

data processing [4]. This highlights the crucial role of sentiment analysis in understanding these comments.

Sentiment analysis is an opinion mining technique that extracts and identifies desired information based on the source. Sentiment analysis aims to classify a review's sentiment as either positive or negative [5]. The performance of sentiment analysis models is significantly influenced by the preprocessing techniques used. Research by [6] confirms that aligning these techniques with the data's character, which refers to the unique features and characteristics of the dataset, such as the language used, the type of text, and the context of the text, is a key factor in shaping the model's accuracy. The findings underscore the significant impact of noise removal and representation expansion on the accuracy of various classifiers. This emphasis on aligning preprocessing techniques with the data's character ensures that the user is equipped with the knowledge of best practices in sentiment analysis, enabling the user to make informed decisions and feel knowledgeable and well-prepared.

Another study conducted by [7] showed that the complete pipeline of preprocessing techniques such as case folding, cleaning, normalization, removing stop words, and stemming, decreased the accuracy of IndoBERT from 88.81% to 85.35% and IndoBERTweet from 92.54% to 88.28%. The loss of emotionally valuable tokens such as "*tidak*," "*namun*," or the original word form prevented the model from capturing the nuances of polarity and led to misclassification. This loss of emotionally valuable tokens can be considered "noise" in the context of preprocessing. In this context, "noise" refers to the removal of tokens that carry emotional or sentiment value, which can hinder the accuracy of sentiment analysis. Here, "sentiment value" refers to the emotional or opinionated content of the text, which is crucial for sentiment analysis. Indonesian language transformers tend to benefit from minimal cleaning to preserve sub word representations, while classical algorithms or English language data still benefit significantly from noise reduction. This stress on the need for caution in preprocessing ensures that the audience is aware of the potential risks and challenges in data preparation, fostering a sense of caution and attentiveness.

Similarly, research conducted by [8] shows that the preprocessing stage is a crucial step before model training. This can impact text classification accuracy, as seen in the XLNet model, a transformer-based language model, with the IMDB dataset, a popular dataset for sentiment analysis in movie reviews, where a simple Naïve Bayes model outperformed it by 2%. This also confirms that input quality is often more important than architectural complexity. The authors also demonstrate that stop-word removal, lowercasing, or stemming processes can eliminate important semantic information, leading to classification errors. Preprocessing that aligns with the data character, which refers to the unique features and characteristics of the dataset, such as the language used, the type of text, and the context of the text, not only improves accuracy but also reduces training time. [8] study shows that the Transformer model is most sensitive to preprocessing variations, while deep networks without pretrained embedding are relatively stable.

Latent Dirichlet Allocation (LDA) is a generative probabilistic model that views each document as a mixture of several latent topics, with each topic represented as a multinomial distribution over words in the corpus [9]. In research related to sentiment analysis, LDA is more often used to group texts resulting from sentiment analysis into specific topics. This is to identify topics frequently discussed in positive and negative sentiments [10] [11]. In addition to being used for topic modelling, LDA can also be used as a feature extraction in sentiment analysis models. LDA is useful for providing semantic context that enriches the interpretation of sentiment scores and reduces the dimensionality of the corpus in the dataset. The use of LDA for feature extraction produces promising performance in sentiment analysis models [12] [13]. For example, research conducted by [14] utilized LDA to extract and cluster product features from reviews, enabling the identification of user needs and the classification of

Vol. 6, No. 5, October 2025, Page. 3033-3050
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5238

documents according to emerging topics. The BERT model uses the weights obtained from the LDA calculation to perform sentiment analysis.

Another study, conducted by [15] and [16], implemented LDA to remove irrelevant words within a class, thereby making the corpus more concise for further processing. In contrast to previous research, [17] utilized the results of LDA processing, where the weighted values of a word would be used in a machine learning method to build a classification model. Other research conducted by [18] and [19] combined the scores generated by LDA with the cosine similarity scores, resulting in a more effective feature representation for the classification model used. Similar to [18] and [19], research conducted by [20] also utilizes LDA to produce more effective feature representations. However, [20] combines LDA with Word2Vec and Doc2Vec. In contrast to the three previous studies, which combined LDA with a word embedding approach, the study conducted by [21] combined LDA with Fuzzy, while [22] and [23] combined it with another topic modeling method, namely Latent Semantic Analysis (LSA).

LDA is not only used to extract relevant features, but research conducted by [24] also utilizes LDA to assist VADER in labeling datasets. LDA is useful because it enables VADER to focus more on words with high weights, resulting in better labeling results. In another study [25], LDA was implemented to perform feature substitution, reducing dimensions while maintaining semantic context, prior to the classification process.

Unlike previous studies that apply LDA post-sentiment classification [10], [11], feature extraction [12], [13], [14], feature representation [15], [16], [17], [18], [19], [20], [21], [22], [23], or assisting the dataset labeling process [24], this research integrates LDA in the preprocessing stage. This research uses LDA to reduce semantic noise, addressing gaps in handling Indonesian slang and noise in app reviews. This study aims to evaluate the impact of LDA in preprocessing on machine learning-based sentiment analysis models for Jenius app reviews, contributing to a better understanding of user sentiments in Indonesian digital banking.

2. METHOD

The research flow, as illustrated in figure 1, is a systematic process designed to evaluate the impact of LDA and other preprocessing techniques on the accuracy of sentiment analysis. It begins with the collection and labelling of text data, followed by the preprocessing stage. This stage comprises several processes, including special character cleaning, slang conversion, stop-word removal, stemming, and the application of Latent Dirichlet Allocation (LDA) to reduce semantic noise.

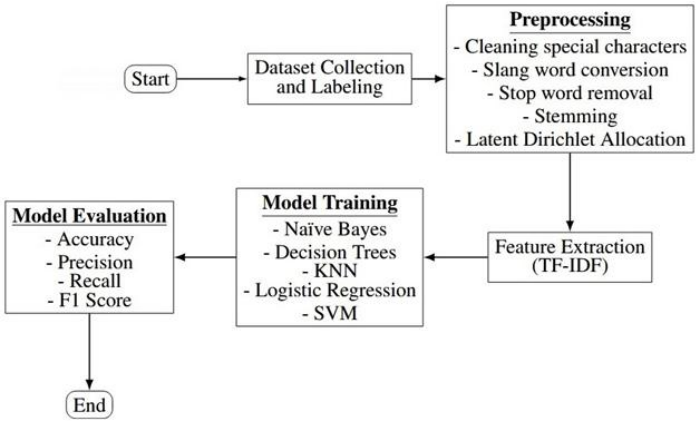


Figure 1. Research Steps

The resulting clean text is then converted into a numeric vector through TF-IDF feature extraction, which serves as input for five machine learning algorithms, such as Naïve Bayes, Decision Tree, K-Nearest Neighbour, Logistic Regression, and Support Vector Machine, in the model training phase. The

Vol. 6, No. 5, October 2025, Page. 3033-3050
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5238

performance of each model is evaluated using accuracy, precision, recall, and F1-score metrics. The resulting evaluation results will provide a comprehensive overview of the effectiveness of LDA in improving the performance of sentiment analysis models. This research flow is a roadmap that guides the reader through the study's methodology, ensuring a clear understanding of the steps taken and the rationale, and the contribution to answering the research question.

2.1. Dataset Collection and Labelling

The dataset used for this study comprises review data collected from social media platforms, including Twitter and the Google Play Store. The dataset was compiled using the Selenium library for review data originating from X and the Google Play Store scraper API for review data from the Google Play Store. The primary advantage of Selenium is its ability to execute websites that contain JavaScript, allowing it to extract content from modern sites based on AJAX or SPA (Single-Page Application) frameworks. Selenium also supports various browsers (Chrome, Firefox, Edge, Safari) and headless mode for efficiency [26]. Meanwhile, the Google Play Scraper API can be used for free without requiring a login to a Google account. This API enables continuous batch retrieval up to a maximum limit per application, providing more efficient and precise results [27]. The collected dataset consisted of 1,800 comments, divided into two classes: 900 positive comment and 900 negative comments. Nur Hany Choirotinnisa, an expert in the Indonesian language, assisted with labelling the dataset. The distribution of the number of datasets for each platform and label is shown in figure 2.

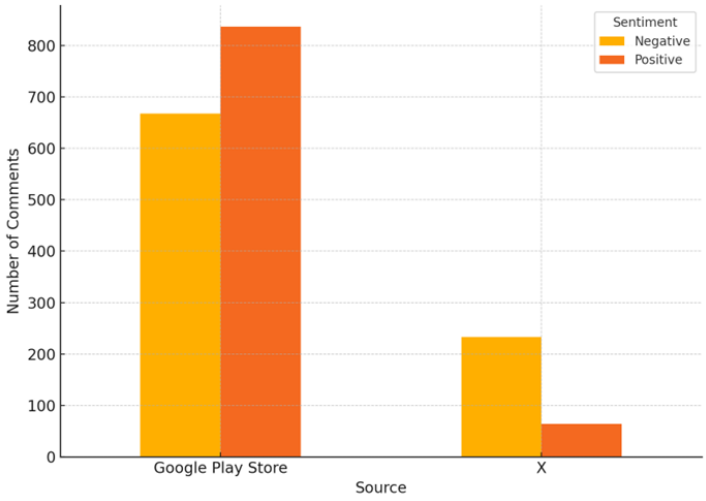


Figure 2. Dataset Distribution

Based on figure 2, the dataset collected from Google Play Store comprises 836 positive comments and 667 negative comments, totaling 1,503 reviews. In contrast, X contributed only 64 positive comments and 233 negative comments, totaling 297 reviews. This pattern indicates that Google Play Store not only contains a higher volume of reviews, but also a relatively higher proportion of positive sentiment compared to X, where reviews tend to be dominated by complaints. The much larger amount of data on the Google Play Store is because every user who installs or updates an application is often prompted to provide a rating directly on the Google Play Store. In contrast, X is not a dedicated review platform, so only a small portion of users are encouraged to express the experiences, typically when encountering problems, resulting in a stronger negative bias. Furthermore, X's limited character policy results in shorter reviews that are not always clearly labelled with sentiment. Some examples of sample datasets are presented in table 1.

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 5, October 2025, Page. 3033-3050
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5238

Table 1. The Examples of Sample Datasets

Comments	Platform	Sentiment
Kenapa susah masuk akun...Katanya koneksi terputus .padahal signal tempat ku bagus..full 4G	Google Play Store	Negative
Aplikasi nya sangat membantu dan mempermudah menabung ðŸ¥°	Google Play Store	Positive
Gimana sih lagi butuh topup ewallet & transfer tapi malah gak bisa?! @JeniusConnect	X	Negative
Gegara drama hp kena hujan krmarin, akhirnya dapat pengalaman menyenangkan dg @JeniusConnect. Ganti device semudah itu, gaes. Kirim email ke cs sat set set udah bisa transaksi lagi deh. Yg penting no hp dan email aktif ya. Thank u Jenius. Muah.	X	Positive

2.2. Preprocessing

Text preprocessing in sentiment analysis aims to provide clean and informative input to the model, ensuring accurate and reliable results. This preprocessing results in a more concise and standardized feature representation, thereby improving the accuracy and generalizability of the sentiment analysis system [28]. The preprocessing stages in this study consist of cleaning special characters, converting slang words, removing stop words, stemming, and applying LDA. The first preprocessing is removing punctuation, hashtags (#), mentions (@), and numbers from the text. This is because these symbols have little impact on the classification. Additionally, all characters will be converted to lowercase. This process is due to the case-sensitive nature of sentiment analysis models, meaning that words written in uppercase and lowercase are considered the same entity [29]. Table 2 shows an example of the results of the special character removal process in this study.

Table 2. The Example of Cleaning Special Characters Results

Original Comments	After Cleaning Special Characters
Kenapa susah masuk akun...Katanya koneksi terputus .padahal signal tempat ku bagus..full 4G	*kenapa susah masuk akun katanya koneksi terputus padahal signal tempat ku bagus full g*
Aplikasi nya sangat membantu dan mempermudah menabung ðŸ¥°	*aplikasi nya sangat membantu dan mempermudah menabung*
Gimana sih lagi butuh topup ewallet & transfer tapi malah gak bisa?! @JeniusConnect	*gimana sih lagi butuh topup ewallet transfer tapi malah gak bisa jeniusconnect*
Gegara drama hp kena hujan krmarin, akhirnya dapat pengalaman menyenangkan dg @JeniusConnect. Ganti device semudah itu, gaes. Kirim email ke cs sat set set udah bisa transaksi lagi deh. Yg penting no hp dan email aktif ya. Thank u Jenius. Muah.	*gegara drama hp kena hujan krmarin akhirnya dapat pengalaman menyenangkan dg jeniusconnect ganti device semudah itu gaes kirim email ke cs sat set set udah bisa transaksi lagi deh yg penting no hp dan email aktif ya thank u jenius muah*

The next step in preprocessing is converting slang words, such as slang, abbreviations, and micro-text, to the standard forms. This is done to prevent the model from misinterpreting words with the same meaning as different words [30]. In this study, the list of slang words and the standard forms was taken from [31]. The list consists of two columns: the first column represents the slang word, and the second column represents its standard form. This slang word conversion process is carried out by checking each word in a comment to see if it is in the first column of the slang word list. If so, it is converted to its standard form. Table 3 shows an example of the slang word conversion results.

Vol. 6, No. 5, October 2025, Page. 3033-3050
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5238

Table 3. The Example of Slang Word Conversion Results

After Cleaning Special Characters	After Slangword Conversion
kenapa susah masuk akun katanya koneksi terputus padahal signal tempat ku bagus full g	*kenapa susah masuk akun katanya koneksi terputus padahal signal tempat ku bagus full tidak*
aplikasi nya sangat membantu dan mempermudah menabung	*aplikasi nya sangat membantu dan mempermudah menabung*
gimana sih lagi butuh topup ewallet transfer tapi malah gak bisa jeniusconnect	*gimana sih lagi butuh topup ewallet transfer tapi malah tidak bisa jeniusconnect*
gegara drama hp kena hujan krmarin akhirnya dapat pengalaman menyenangkan dg jeniusconnect ganti device semudah itu gaes kirim email ke cs sat set set udah bisa transaksi lagi deh yg penting no hp dan email aktif ya thank u jenius muah	*gegara drama hp kena hujan kemarin akhirnya dapat pengalaman menyenangkan dengan jeniusconnect ganti device semudah itu gaes kirim email ke cs sat set set sudah bisa transaksi lagi sudah yang penting no hp dan email aktif ya thank kamu jenius muah*

The following preprocessing step is stop word removal, which aims to remove frequently occurring words that have little impact on the sentiment analysis model. Stop words include prepositions and conjunctions such as "*yang*," "*dan*," "*ke*," and "*dengan*," among others [28]. In this study, the stop word removal process is implemented using the Sastrawi library, as the dataset consists predominantly of Indonesian text. Table 4 shows an example of stop word removal results.

Table 4. The Example of Stop Word Removal Results

After Slangword Conversion	After Stop Word Removal
kenapa susah masuk akun katanya koneksi terputus padahal signal tempat ku bagus full tidak	*kenapa susah masuk akun katanya koneksi terputus padahal signal tempat bagus full tidak*
aplikasi nya sangat membantu dan mempermudah menabung	*aplikasi membantu mempermudah menabung*
gimana sih lagi butuh topup ewallet transfer tapi malah tidak bisa jeniusconnect	*gimana butuh topup ewallet transfer malah tidak jeniusconnect*
gegara drama hp kena hujan kemarin akhirnya dapat pengalaman menyenangkan dengan jeniusconnect ganti device semudah itu gaes kirim email ke cs sat set set sudah bisa transaksi lagi sudah yang penting no hp dan email aktif ya thank kamu jenius muah	*gegara drama hp kena hujan kemarin akhirnya pengalaman menyenangkan jeniusconnect ganti device semudah gaes kirim email cs sat set set transaksi penting no hp email aktif thank kamu jenius muah*

Table 5. The Example of Stemming Results

After Stop Word Removal	After Stemming
kenapa susah masuk akun katanya koneksi terputus padahal signal tempat bagus full tidak	*kenapa susah masuk akun kata koneksi putus padahal signal tempat bagus full tidak*
aplikasi membantu mempermudah menabung	*aplikasi bantu mudah tabung*
gimana butuh topup ewallet transfer malah tidak jeniusconnect	*gimana butuh topup ewallet transfer malah tidak jeniusconnect*
gegara drama hp kena hujan kemarin akhirnya pengalaman menyenangkan jeniusconnect ganti device semudah gaes kirim email cs sat set set transaksi penting no hp email aktif thank kamu jenius muah	*gara drama hp kena hujan kemarin akhir pengalaman senang jeniusconnect ganti device mudah gaes kirim email cs sat set set transaksi penting no hp email aktif thank kamu jenius muah*

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 5, October 2025, Page. 3033-3050
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5238

The next process is stemming which aims to return words to the basic form (stem or root). This process aims to reduce duplication of words with similar meanings, such as "*membantu*," "*dibantu*," and "*bantu*." The results of this process are expected to improve computational efficiency. Similar to stop word removal, this process will utilize the Sastrawi library because the dataset is predominantly Indonesian text [29]. Table 5 shows an example of stemming results.

The following process is Latent Dirichlet Allocation (LDA) calculation. LDA is a generative probabilistic model often used in topic modeling. This model aims to identify information stored in a document, where the document can belong to a specific topic. Each topic is represented by words related to that topic in the document [11]. In this study, LDA is used to select important words in a document (comments) related to existing sentiment labels, which are then taken into the feature extraction process. In the feature extraction process, these important words will receive greater weight. In general, the way LDA works is by first determining the number of desired topics; in this study, it is two. This is because only two sentiment labels are used: positive and negative. Next, the probability value of a word about a topic within a document will be calculated. This probability calculation allows for conditions where a word can fall into two different topics. To overcome this condition, the collapsed Gibbs method can be used [10]. The general LDA model is illustrated in figure 3, where α is the Dirichlet parameter for the distribution of topics with index Z in a document (θ) in a set of documents (M). While β is the Dirichlet parameter for the distribution of words to topics, and φ is the distribution of words (W) in a set of words (N) to a set of topics (K) in the corpus [32].

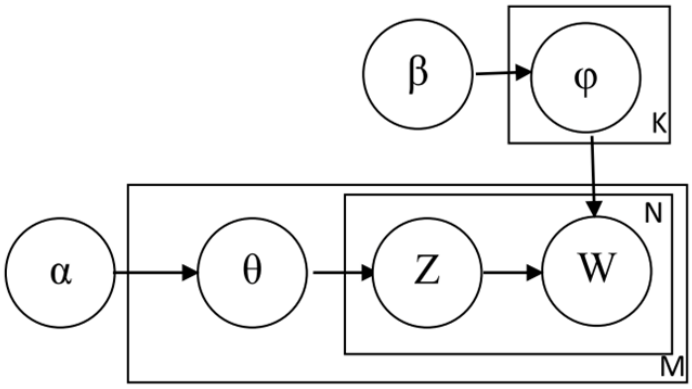


Figure 3. General Model of LDA

The collapsed Gibbs method begins by measuring how close a word is to a particular topic, calculated by equation (1), where $C_{w_{t,j}}^{w_t}$ is the number of occurrences of word w_t that have been assigned to topic j. The calculation in the denominator is useful for ensuring the normalized probability in topic j, given the vocabulary size (W) [10].

$$C_w = \frac{C_{w_{t,j}}^{w_t} + \beta}{\sum_{w=1}^{W} C_{w_j}^{w_t} + W\beta} \quad (1)$$

After calculating how close a word is to a topic, the next step in the collapsed Gibbs method calculation is to measure whether a topic is representative of a document. This measurement is done using (2), where $C_{d_{t,j}}^{D_t}$ is the number of words in document dt that fall into topic j. Similar to calculation (1), the denominator is useful for normalizing the frequency of topics in a document. After that, calculations (1) and (2) will be combined, then the latent variable z_i will be updated, namely the topic that has a word, while assuming all topics for other words z_{-i} remain constant, so that the final calculation becomes (3) [10].

Vol. 6, No. 5, October 2025, Page. 3033-3050
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5238

$$C_D = \frac{C_{d_{t,j}}^{D_t} + \alpha}{\sum_{j'=1}^{T} C_{d_{t,j'}}^{D_t} + T\alpha} \quad (2)$$

$$p(z_i = j \mid z_{-i}, w_i, d_t) \propto C_w \; x \; C_D \quad (3)$$

For example, based on the stemming results in Table 5, further processing using LDA yields the results presented in table 6. In this study, the LDA implementation was carried out using the Gensim library. The results in table 6 were obtained using a smaller α value than the default in the Gensim library, which is 0.1. This value is used to focus the topic distribution, where documents are more inclined towards one dominant topic, positive or negative [33].

Table 6. The Example of LDA Calculation Results

Topics	Word Distribution
Positive	*mudah (0.25), bantu (0.22), tabung (0.19), aplikasi (0.15)*
Negative	*koneksi (0.30), putus (0.28), drama (0.22), masalah (0.15)*

2.3. Feature Extraction

Feature extraction is a process that converts text into a numerical representation, enabling it to be processed by a model. One feature extraction method commonly used in sentiment analysis is the Term Frequency-Inverse Document Frequency (TF-IDF) method. This method measures the importance of a word within a document relative to the entire document collection. This method consists of two important parts: TF, which measures how often word *t* appears in document *d*, and IDF, which is useful for reducing the weight of word *t* that often appears in almost all documents. IDF is calculated using (4), where *N* is the number of documents, in this case, text comments, and $df_{t,d}$ is the number of documents that contain a particular word. After the TF and IDF values are obtained, the two values will be multiplied to get the final value of TF-IDF (5) [29]. After the TF-IDF calculation for all words is performed, words with high probabilities from the previous LDA calculation will have an extra value added to increase the weight of the word.

$$IDF_{t,d} = \frac{N}{df_{t,d}} \quad (4)$$

$$TF - IDF_{t,d} = TF_{t,d} \; x \; IDF_{t,d} \quad (5)$$

2.4. Model Training

The next step after feature extraction is training a sentiment analysis model using machine learning methods. In this study, we tested several machine learning methods, including Naïve Bayes, Decision Trees, KNN, Logistic Regression, and SVM, to find the best sentiment analysis model. Naïve Bayes is a probabilistic method based on Baye's Theorem that calculates the probability of a document (or text) belonging to a particular sentiment class, based on the weights of its words. This method assumes that words are independent of each other. The advantages of this method are its fast training process and its robustness to noise [34]. There are two ways to classify based on the type of data in an attribute: numeric and non-numeric. In this study, the Naïve Bayes method will be used to calculate numeric data, specifically the TF-IDF values obtained from the feature extraction process. In Naïve Bayes calculations with numeric data, we assume each data item has a normal distribution. The Naïve Bayes calculation is presented in (6), where μ_{ck} represents the average feature for each class and σ_{ck} denotes the standard deviation [35].

$$P(x_i|y = c_k) = \frac{1}{\sqrt{2\pi\sigma_{ck}}} e^{-\frac{(x_i-\mu_{ck})^2}{2\sigma_{ck}^2}} \quad (6)$$

Decision Tree is a collection of attributes arranged into a structured tree that can be tested and aims to predict its output. The decision tree structure begins with the root node, the first node, which represents the dataset split. It then proceeds to the inner node, or internal node, and the leaf node, or end node, which contains the class label used for sentiment classification. In determining the nodes of the decision tree, the concepts of Entropy (7) and Gain (8) are used, where $p(w_i|s)$ is the probability of the i-th class on all training data processed at node s. The selected attribute is the attribute that provides the highest gain or the lowest entropy after the split. The child nodes after the calculation are done will be recalculated recursively for each new child node. This process stops when there are no more splits [36].

$$E_s = -\sum_{i=1}^{m} p(w_i|s)\, log_2\, p(w_i|s) \quad (7)$$

$$G_{s,i} = E_s - \sum_{i=1}^{n} p(w_i|s)\; x\; E_{s,i} \quad (8)$$

K-Nearest Neighbour (KNN) is a method that classifies a document (text) based on its k nearest neighbours. Each document is represented as a point in the feature space, and the label of a document is determined by the majority label among its k nearest neighbors. The advantage of this method is that it can work quickly because it does not require a parameter training phase [37]. In general, the selection of the k value is determined using (9), where n is the number of documents in the dataset. Meanwhile, to measure the distance to k neighbors, the Euclidean approach is used (10), where x_i and y_i are the data whose proximity is to be calculated [29].

$$k = \sqrt{n} \quad (9)$$

$$d_i = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (10)$$

Unlike linear regression, which is used to predict continuous data, logistic regression is used to predict discrete data. This method maps linear feature scores to class probabilities through the sigmoid (σ) function using (11). Probabilistic (p) values are expressed through odds and logit, which is the log-odds transformation of the probability of a positive class, making the relationship with the feature linear using (12), where w is the parameters learned and x is the feature vector, in this case, TF-IDF values [38]. The final decision of the logistic regression model is determined through the decision boundary, which is a hyperplane in the feature space that separates classes based on the model's output value. If $w^Tx \geq 0$, then the document is classified into class 1 (i.e., positive sentiment), while if $w^Tx < 0$, then it is classified into class 0 (negative). The dividing boundary is the hyperplane $w^Tx = 0$, where the condition $w^Tx = 0$ is equivalent to the probability $p(y=1|x) = 0$, because the sigmoid function is 0.5 when the input is zero [39].

$$\sigma(a) = \frac{1}{1+e^{-a}} \quad (11)$$

$$logit(p) = ln(odds(p) = w^T x \quad (12)$$

The Support Vector Machine (SVM) is a classification algorithm renowned for its superior performance, particularly due to its ability to process high-dimensional data and achieve a high level of accuracy. Similar to logistic regression, SVM classifies documents by finding a separating hyperplane that maximizes the margin between positive and negative classes. The basic linear model in SVM is calculated using (13), where w represents the learned parameters, x denotes the feature vector (in this

Vol. 6, No. 5, October 2025, Page. 3033-3050
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5238

case, TF-IDF values), and *b* represents the bias. If $f(x) \geq 0$, then the sentiment label is positive; if $f(x) < 0$, then the sentiment label is negative. Another advantage of the SVM method is its robustness against noise and overfitting [40].

$$f(x) = w^T x + b \quad (13)$$

2.5. Model Evaluation

After all models are generated, the models will be evaluated by calculating the accuracy, precision, recall, and F1-score metrics. These four metrics are calculated by considering the values of correctly detected positive comments (TP), incorrectly predicted negative comments as positive (FP), predicted positive comments as negative (FN), and correctly detected negative comments (TN). The accuracy, precision, recall, and F1-score metrics are calculated using (14), (15), (16), and (17), respectively [28].

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (14)$$

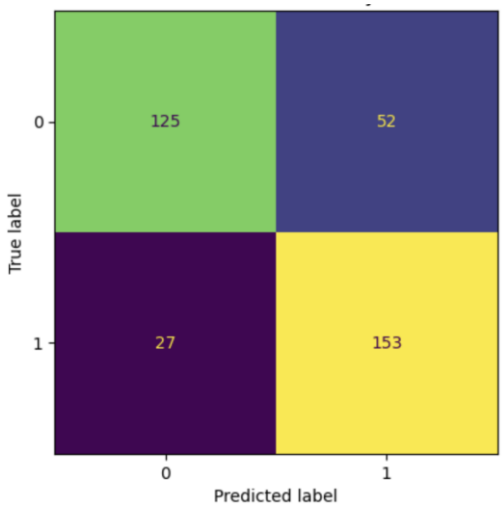
$$Precision = \frac{TP}{TP+FP} \quad (15)$$

$$Recall = \frac{TP}{TP+FN} \quad (16)$$

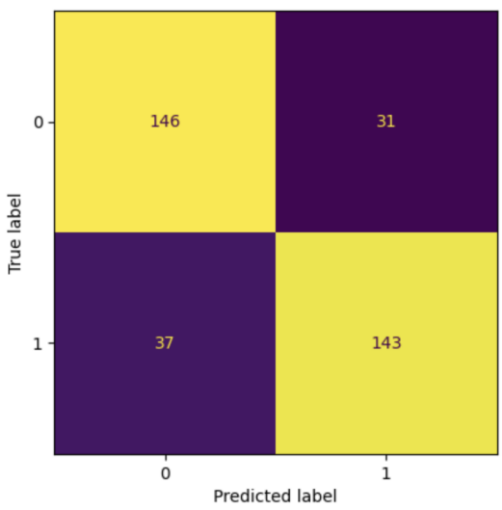
$$F1 - score = \frac{2\ x\ Precision\ x\ Recall}{Precision+Recall} \quad (17)$$

3. RESULT

This study comprises three parts: evaluation of models using unpreprocessed datasets, preprocessed datasets without LDA, and preprocessed datasets incorporating LDA. The existing dataset is divided into two parts: training data and testing data, with a proportion of 80% training data and 20% testing data. The creation of each machine learning model will use 80% of the existing training data, and evaluation will use 20% of the existing testing data. The evaluation is carried out by creating a confusion matrix to view all the resulting true positive (TP), false positive (FP), false negative (FN), and true negative (TN) values. The confusion matrix for each machine learning model using an unpreprocessed dataset is shown in figure 4.



(a)



(b)

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 5, October 2025, Page. 3033-3050
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5238

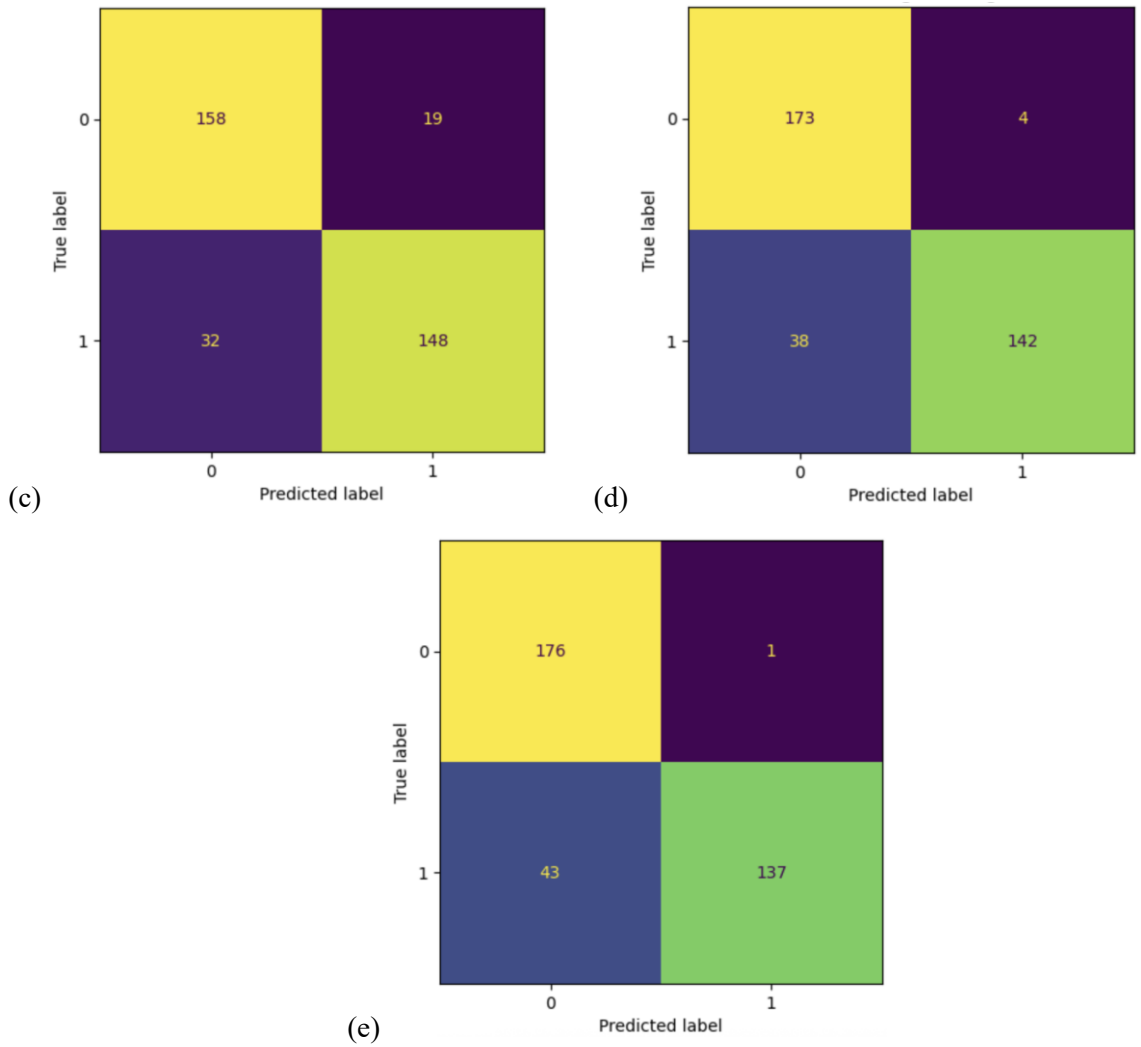


Figure 4. Confusion Matrix for Each Model without Preprocessing (a) Naïve Bayes; (b) Decision Trees; (c) KNN; (d) Logistic Regression; (e) SVM

Based on the confusion matrix shown in figure 4, the accuracy, precision, recall, and F1-score values of each existing machine learning model can be calculated. The results of each metric calculation are presented in table 7, where the Logistic Regression and SVM models demonstrate superior performance compared to the Decision Tree and Naïve Bayes models. The Logistic Regression model slightly outperforms the SVM model, with a recall value that is 1% higher.

Table 7. The Test Results Without Preprocessing

Models	Accuracy	Precision	Recall	F1-Score
Naïve Bayes	78%	79%	78%	78%
Decision Trees	81%	81%	81%	81%
KNN	86%	86%	86%	86%
Logistic Regression	88%	90%	89%	88%
SVM	88%	90%	88%	88%

The next evaluation was conducted on machine learning models trained with preprocessed datasets, but without LDA. This was done to demonstrate the effect of preprocessing on machine learning models in sentiment analysis. The confusion matrix from the evaluation results is shown in figure 5.

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 5, October 2025, Page. 3033-3050
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5238

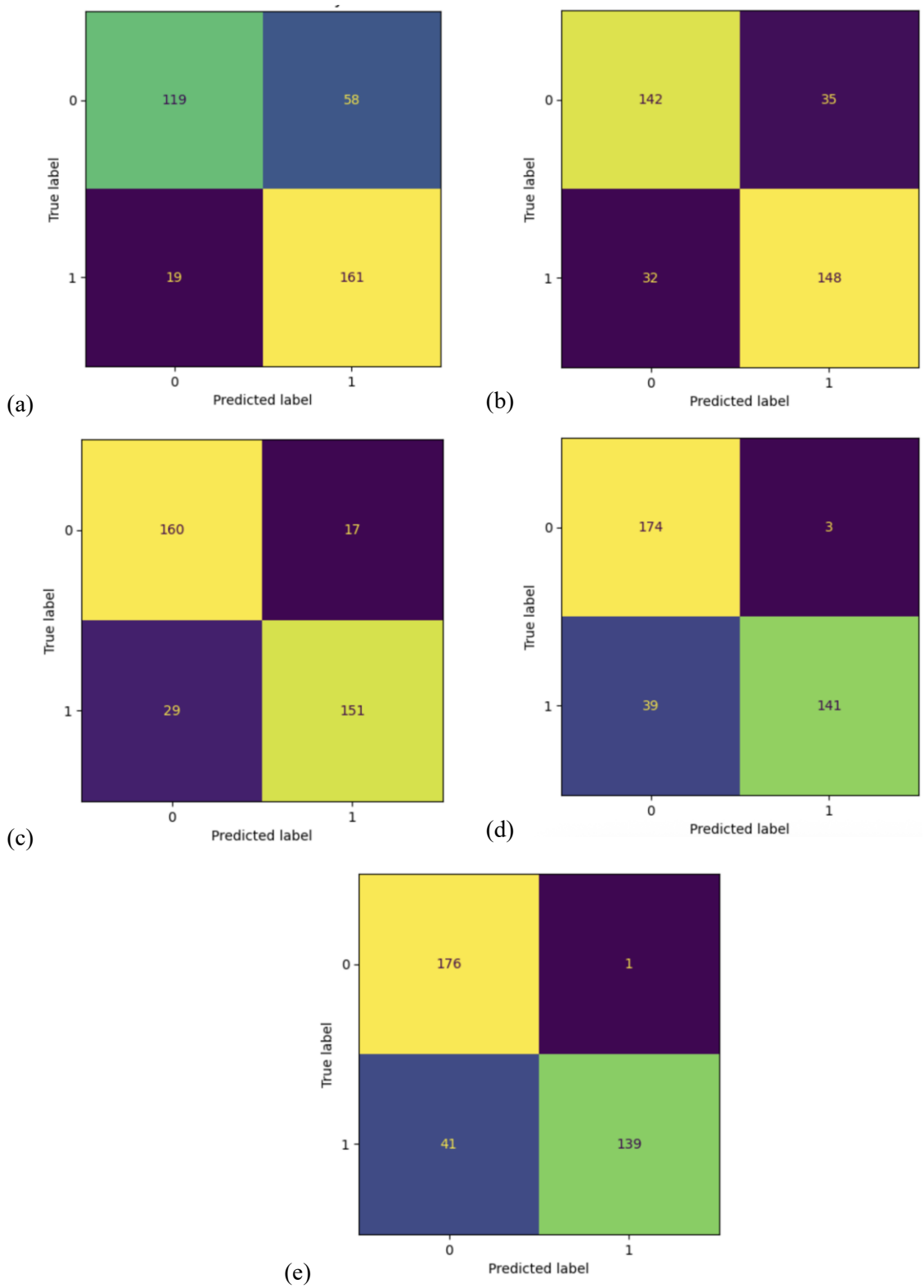


Fig. 5 Confusion Matrix for Each Model with Preprocessing (Without LDA) (a) Naïve Bayes; (b) Decision Trees; (c) KNN; (d) Logistic Regression; (e) SVM

Furthermore, the accuracy, precision, recall, and F1-score values based on figure 5 are presented in table 8. According to the evaluation results, the performance of the SVM model remained unchanged from the previous one. Meanwhile, the performance of the Naïve Bayes and Decision Tree models

Vol. 6, No. 5, October 2025, Page. 3033-3050
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5238

improved by 1% in terms of precision and F1-score metrics. In contrast to the previous two models, the KNN model showed a performance increase of 1-2% in all metrics, while the Logistic Regression model experienced a slight decrease in performance, specifically in the recall metric, by 1%.

Table 8. The Test Results with Preprocessing (Without LDA)

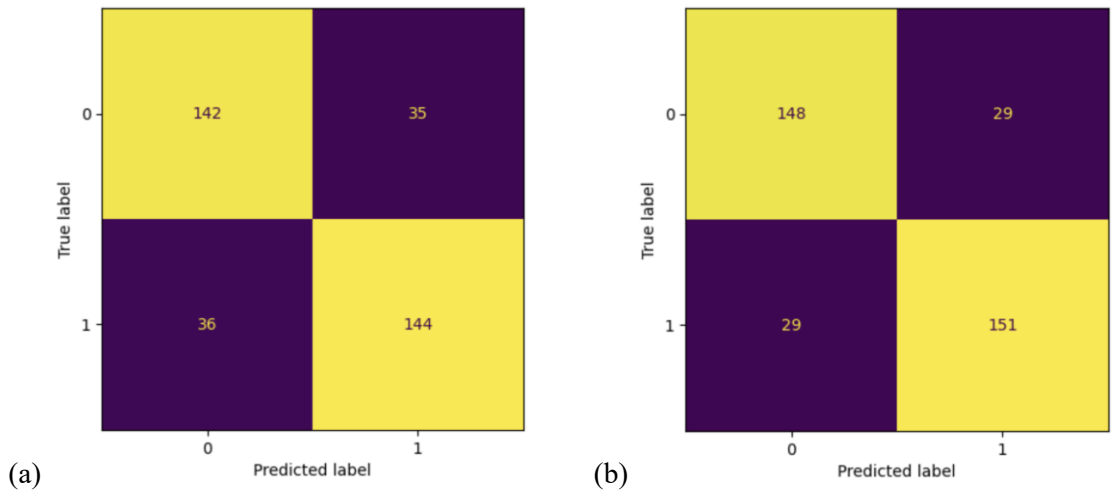
Models	Accuracy	Precision	Recall	F1-Score
Naïve Bayes	78%	80%	78%	79%
Decision Trees	81%	82%	81%	82%
KNN	87%	88%	87%	87%
Logistic Regression	88%	90%	88%	88%
SVM	88%	90%	88%	88%

The final evaluation was conducted on machine learning models trained with preprocessed datasets using the LDA process. This was done to demonstrate the effect of LDA on the performance of the resulting sentiment analysis model in the preprocessing stage. Words with high influence on the dataset, as determined by the LDA calculation results, are listed in table 9. These results will determine the subsequent feature extraction process, where words in table 9, specifically those in the Words Distribution column, will have a higher TF-IDF value than in the original TF-IDF calculation. Words selected from the LDA calculation results for the feature extraction process are words with an LDA value of at least 0.15.

Table 9. LDA Calculation Results on All Dataset

Topics	Word Distribution
Positive	*aplikasi (0.57), mudah (0.41), bantu (0.33), tabung (0.27), bagus (0.17), top (0.15)*
Negative	*login (0.49), koneksi (0.43), putus (0.35), drama (0.27), masalah (0.17), transfer (0.15)*

The evaluation results of machine learning models involving LDA in the dataset preprocessing stage are presented in the confusion matrix in figure 6 and table 10. Based on the evaluation results, it can be seen that all machine learning models experienced a performance increase of between 1% and 3% in all metrics, except for the Naïve Bayes and SVM models, where the precision metric remained unchanged.



Vol. 6, No. 5, October 2025, Page. 3033-3050
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5238

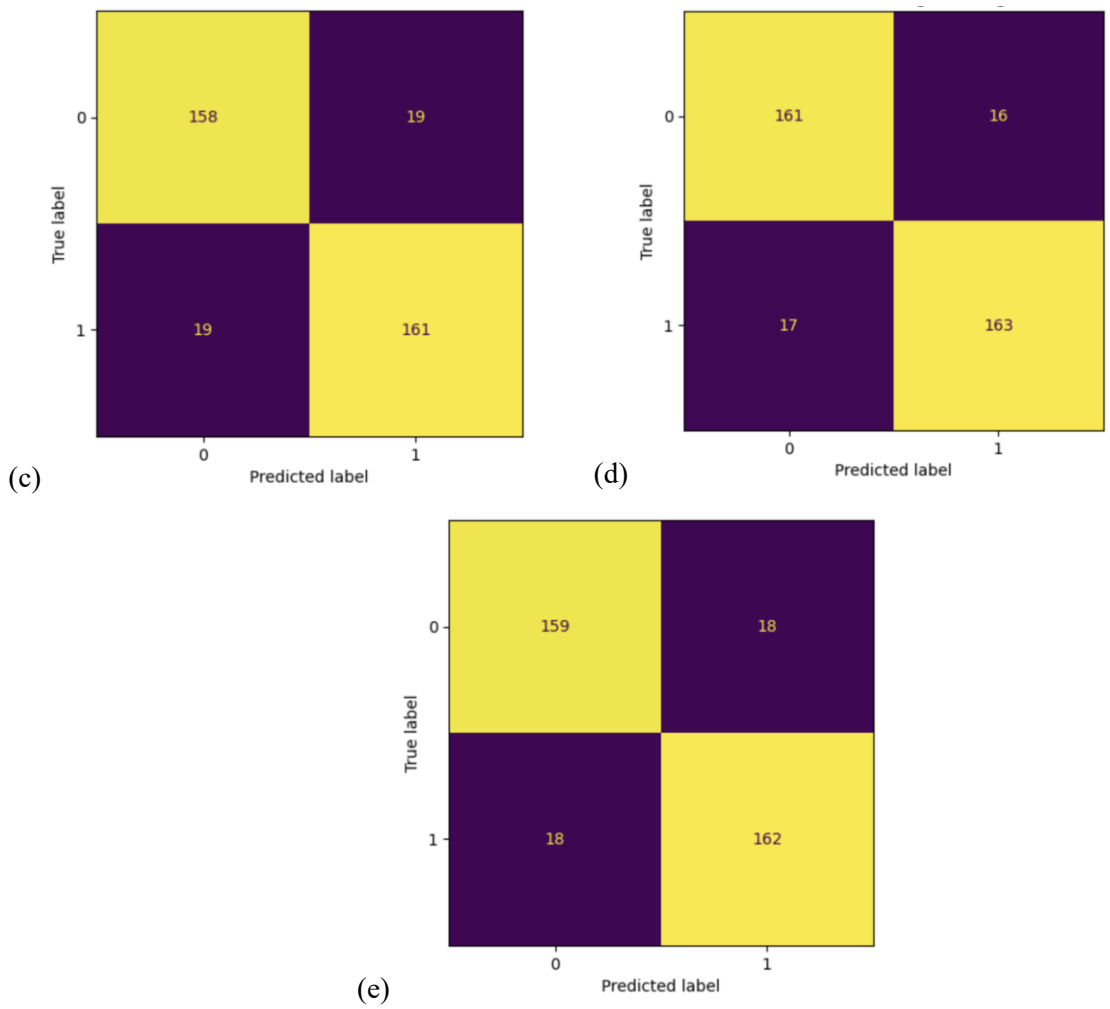


Fig. 6 Confusion Matrix for Each Model with Preprocessing (using LDA) (a) Naïve Bayes; (b) Decision Trees; (c) KNN; (d) Logistic Regression; (e) SVM

Table 10. The Test Results with Preprocessing (Using LDA)

Models	Accuracy	Precision	Recall	F1-Score
Naïve Bayes	78%	80%	78%	79%
Decision Trees	81%	82%	81%	82%
KNN	87%	88%	87%	87%
Logistic Regression	88%	90%	88%	88%
SVM	88%	90%	88%	88%

4. DISCUSSIONS

The results of this study indicate that implementing LDA in the preprocessing stage, specifically to select important words that correspond to sentiment labels, will be used to increase the weight of these words in the TF-IDF calculation stage, providing a 1 to 3% improvement in most models, with the Logistic Regression model achieving the best performance across all metrics. This 1-3% improvement outperforms preprocessing without LDA, contributing to informatics by enhancing noise reduction in Indonesian text, thereby supporting better app development decisions. This finding is reinforced by the evaluation, which found that all models experienced improved performance after using LDA, suggesting that topic-based feature selection can strengthen feature representation in the text. This contrasts with previous studies [10], [11] that used LDA for topic modeling after classification, as well as using LDA to perform feature extraction [12], [13], [14], and also assist in the labeling process [24].

Vol. 6, No. 5, October 2025, Page. 3033-3050
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5238

Furthermore, the composition of the dataset can also influence the performance of the classification model, as most reviews originate from the Google Play Store platform, which tends to have more positive reviews, compared to platform X, which contains relatively more negative complaints. This imbalance in data size can explain why models with linear decision lines, such as Logistic Regression and SVM, outperform when keywords like "*login*", "*koneksi*", "*mudah*", and "*bantu*" are highlighted by LDA, resulting in a more apparent hyperplane separation between the two sentiment labels. In other words, the Google Play Store platform, as a product feedback channel, can support the effectiveness of LDA calculations more effectively than X, which serves as a complaint channel.

This study also has limitations, including an imbalanced dataset between platforms, where the raw data is biased toward the Google Play Store and particular sentiments. This can weaken generalizability to the real world. Another limitation is the limited number of topics in the LDA calculation, which is constrained to two topics due to the use of sentiment labels. This topic selection may overlook comments that tend to be neutral in nature. A further limitation is the LDA implementation, which uses a smaller α value than the default to focus on topic distribution. While this α value was practical in this study, it has not been tuned for topic coherence. In this study, labelling was performed with the assistance of a single Indonesian linguist, which could potentially lead to annotator bias. Finally, the feature extraction process relied solely on TF-IDF as a text representation, which could result in the model's inability to capture semantic concepts effectively.

To address these limitations, future research could include hyperparameter tuning for alpha and beta values in the LDA calculation in the Gensim library. Furthermore, future research could compare LDA with other topic modelling methods such as Biterm Topic Modelling, Neural Topic Models, and BERTopic. Furthermore, TF-IDF can be compared with other text representation methods, such as fastText or Word2Vec, before model training. Finally, future research could expand the data coverage to include banking applications and other platforms, as well as the use of multiple classes in the classification process.

5. CONCLUSION

This research advances computer science in sentiment analysis by demonstrating the efficacy of LDA in preprocessing noisy Indonesian datasets, enabling more accurate user feedback processing in digital applications. This study concludes that LDA used in the preprocessing stage to select important words that correspond to sentiment labels to increase word weights in the TF-IDF calculation in the feature extraction stage can improve performance between 1-3% in most machine learning models, with Logistic Regression being the best model, followed by SVM and KNN models. This LDA implementation differs from previous studies, which only used topic modeling after the classification process was completed. The main limitations of this study include the imbalance of sources in the raw data and the number of topics used. Future research can be compared with other topic modeling and text representation methods, expand cross-platform datasets, and explore multi-class classification.

CONFLICT OF INTEREST

The authors declares that there is no conflict of interest between the authors or with research object in this paper.

REFERENCES

[1] C. L. Rithmaya, H. Ardianto and E. Sistiyarini, "Gen Z and The Future of Banking: An Analysis of Digital Banking Adoption," *Jurnal Manajemen Dan Kewirausahaan*, vol. 26, no. 1, pp. 64–78, 2024. doi: 10.9744/jmk.26.1.64-78.

Vol. 6, No. 5, October 2025, Page. 3033-3050
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5238

[2] F. N. Styaningsih and Z. Abidin, "The Influence of Personal Selling and Service Quality on Jenius Application User Satisfaction and Loyalty Using the E-Servqual Model," *Journal of Advances in Information Systems and Technology*, vol. 7, no. 1, 2025. doi: 10.15294/jaist.v7i1.13259.

[3] R. Alawaji and A. Aloraini, "Sentiment Analysis of Digital Banking Reviews Using Machine Learning and Large Language Models," *Electronics*, vol. 14, no. 11, 2025. doi: 10.3390/electronics14112125.

[4] A. A. Mulyadi, S. H. Wijoyo and H. M. Az-Zahra, "Analisis Pengaruh Kualitas Layanan Terhadap Kepuasan Pelanggan dan Loyalitas Pengguna Aplikasi Jenius Menggunakan Model E-S- Qual dan E- Recs- QUal (Studi Kasus: Pengguna Aplikasi Jenius Kota Malang)," *Jurnal Teknologi Informasi Dan Ilmu Komputer*, vol. 9, no. 6, pp. 1145-1154, 2022. doi: 10.25126/jtiik.2022934937.

[5] Jimmy and V. R. Prasetyo, "Sentiment analysis on feedback of higher education teaching conduct: An empirical evaluation of methods," in *Proceedings The 3rd International Conference on Informatics, Technology and Engineering*, 2022. doi: 10.1063/5.0080182.

[6] M. IŞIK and H. DAĞ, "The impact of text preprocessing on the prediction of review ratings," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 28, p. 1405 – 1421, 2020. doi: 10.3906/elk-1907-46.

[7] U. Khairani, V. Mutiawani and H. Ahmadian, "Pengaruh Tahapan Preprocessing Terhadap Model IndoBERT dan IndoBERTTweet Untuk Mendeteksi Emosi pada Komentar Akun Berita Instagram," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 4, pp. 887-894, 2024. doi: 10.25126/jtiik.1148315.

[8] M. Siino, I. Tinnirello and M. L. Cascia, "Is text preprocessing still worth the time? A comparative survey on the influence of popular preprocessing methods on Transformers and traditional classifier," *Information Systems*, vol. 121, 2024. doi: 10.1016/j.is.2023.102342.

[9] R. Egger and J. Yu, "A Topic Modeling Comparison Between LDA, NMF, Top2Vec, and BERTopic to Demystify Twitter Posts," *Frontiers in Sociology*, vol. 7, no. 886498, 2022. doi: 10.3389/fsoc.2022.886498.

[10] T. Ali, B. Omar and K. Soulaimane, "Analyzing tourism reviews using an LDA topic-based sentiment analysis approach," *MethodsX*, vol. 9, 2022. doi: 10.1016/j.mex.2022.101894.

[11] N. M. K. Sedana, I. N. S. Wijaya and I. K. R. Artana, "Analisis Sentimen Berbahasa Inggris Dengan Metode LSTM Studi Kasus Berita Online Pariwisata Bali," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 6, pp. 1325-1334, 2024. doi: 10.25126/jtiik.2024118792.

[12] K. Chen and G. Wei, "Public sentiment analysis on urban regeneration: A massive data study based on sentiment knowledge enhanced pre-training and latent Dirichlet allocation," *PLoS ONE*, vol. 18, no. 4, 2023. doi: 10.1371/journal.pone.0285175.

[13] D. Voskergian, R. Jayousi and M. Yousef, "Topic selection for text classification using ensemble topic modeling with grouping, scoring, and modeling approach," *Scientific Reports*, vol. 14, no. 23516, 2024. doi: 10.1038/s41598-024-74022-2.

[14] L. Liu and B. Ma, "CA-VAR-Markov model of user needs prediction based on user generated content," *Scientific Reports*, vol. 15, no 7716, 2025. doi: 10.1038/s41598-025-92173-8.

[15] Q. Zhou, D. Yang, S .Zheng and S. Cheng, "Research on Sentiment Analysis Techniques for Online Ideological and Political Education," *Applied Mathematics and Nonlinear Sciences*, vol. 9, no. 1, pp. 1-19, 2024. doi: 10.2478/amns.2023.2.00338.

[16] N. Jacob and V. M. Viswanatham, "Sentiment Analysis Using Improved Atom Search Optimizer With a Simulated Annealing and ReLU Based Gated Recurrent Unit," *IEEE Access*, vol. 12, pp. 38944-38956, 2024. doi: 10.1109/ACCESS.2024.3375119.

[17] Z. A. Guven, B. Diri and T. Cakaloglu, "Impact of N-Stage Latent Dirichlet Allocationon Analysis of Headline Classification," *Computer Science*, vol. 23, no. 3, pp. 375-394, 2022. doi: 10.7494/csci.2022.23.3.4622.

[18] Y. Su and Z. J. Kabala, "Public Perception of ChatGPT and Transfer Learning for Tweets Sentiment Analysis Using Wolfram Mathematica," *Data*, vol. 8, no. 180, 2023. doi: 10.3390/data8120180.

[19] S. Yue, "Research on Microblog Comment Clustering Algorithm Based on Emotional Topic Feature Word Weighting," *Proceedings IEEE 2nd International Conference on Control,*

Vol. 6, No. 5, October 2025, Page. 3033-3050
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5238

Electronics and Computer Technology (ICCECT), 2024. doi: 10.1109/ICCECT60629.2024.10545884.

[20] N. N. Hidayati, "Improving Aspect-Based Sentiment Analysis for Hotel Reviews with Latent Dirichlet Allocation and Machine Learning Algorithms," *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, vol .9, no. 2, pp. 144-159, 2023. doi: 10.26594/register.v9n2.3441.

[21] S. Song and A. P. Johnson, "Predicting Drug Review Polarity Using the Combination Model of Multi-Sense Word Embedding and Fuzzy Latent Dirichlet Allocation (FLDA)," *IEEE Access*, vol. 11, pp. 118538-118546, 2023. doi: 10.1109/ACCESS.2023.3326757.

[22] S. R. Kothuri and N. R. RajaLakshmi, "MALO-LSTM: Multimodal Sentiment Analysis Using Modified Ant Lion Optimization with Long Short Term Memory Network," *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 5, 2022. doi: 10.22266/ijies2022.1031.29.

[23] A. Pradhan, M. R. Senapati and P. K. Sahu, "Improving sentiment analysis with learning concepts from concept, patterns lexicons and negations," *Ain Shams Engineering Journal*, vol. 13, 2022. doi: 10.1016/j.asej.2021.08.004.

[24] N. Aslam, K. Xia, F. Rustam, A. Hameed and I. Ashraf, "Using Aspect-Level Sentiments for Calling App Recommendation with Hybrid Deep-Learning Models," *Applied Sciences*, vol. 12, no. 17, 2022. doi: 10.3390/app12178522.

[25] N. M. N. Mathivanan, R. M. Janor, S. A. Razak, and N. A. M Ghani, "Feature Substitution Using Latent Dirichlet Allocation for Text Classification," *International Journal of Advanced Computer Science and Applications*, vol. 16, no. 1, pp. 1087-1098, 2025. doi: 10.14569/IJACSA.2025.01601105.

[26] S. Yuan, "Design and Visualization of Python Web Scraping Based on Third-Party Libraries and Selenium Tools," *Academic Journal of Computing & Information Science*, vol. 6, no. 9, pp. 25-31, 2023. doi: 10.25236/AJCIS.2023.060904.

[27] E. D. Madyatmadja, H. Candra, J. Nathaniel, M. R. Jonathan and Rudy, "Sentiment Analysis on User Reviews of Threads Applications in Indonesia," *Journal Européen des Systèmes Automatisés*, vol. 57, no. 4, pp. 1165-1171, 2024. doi: 10.18280/jesa.570423.

[28] V. R. Prasetyo, M. F. Naufal and K. Wijaya, "Sentiment Analysis of ChatGPT on Indonesian Text using Hybrid CNN and Bi-LSTM," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 9, no. 2, pp. 327-333, 2025. doi: 10.29207/resti.v9i2.6334.

[29] V. R. Prasetyo and A. H. Samudra, "Hate Speech Content Detection System on Twitter using K-Nearest Neighbor Method," in *Proceedings The 3rd International Conference on Informatics, Technology and Engineering*, 2022. doi: 10.1063/5.0080185.

[30] M. I. Raif, N. N. Hidayati and T. Matulatan, "Otomatisasi Pendeteksi Kata Baku Dan Tidak Baku Pada Data Twitter Berbasis KBBI," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 2, pp. 337-348, 2024. doi: 10.25126/jtiik.20241127404.

[31] M. O. Ibrohim and I. Budi, "A Dataset and Preliminaries Study for Abusive Language Detection in Indonesian Social Media," *Proceedings in 3rd International Conference on Computer Science and Computational Intelligence,* 2018. doi: 10.1016/j.procs.2018.08.169.

[32] R. Yasutomi, S. Yamada and T. Onoda, "Examination of Document Clustering Based on Independent Topic Analysis and Word Embeddings," in *Proceedings 17th International Conference on Agents and Artificial Intelligence*, 2025. doi: 10.5220/0013104100003890

[33] M. Zhang, L. Sun, Y. Li, G. A. Wang and H. Zhen, "Using supplementary reviews to improve customer requirement identification and product design development," *Journal of Management Science and Engineering*, vol. 8, no. 4, pp. 584-597, 2023. doi: 10.1016/j.jmse.2023.03.001.

[34] A. Vitetta, "Sentiment Analysis Models with Bayesian Approach: A Bike Preference Application in Metropolitan Cities," *Journal of Advanced Transportation*, vol. 2022, no. 1, 2022. doi: 10.1155/2022/2499282.

[35] J. P. Arisula and P. Parjito, "Comparison Of Naive Bayes And Random Forest Methods In Sentiment Analysis On The Getcontact Application", *Jurnal Teknik Informatika (JUTIF)*, vol. 5, no. 5, pp. 1221-1230, 2024. doi: 10.52436/1.jutif.2024.5.5.2004.

Vol. 6, No. 5, October 2025, Page. 3033-3050
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.5238

[36] M. Yang, "English Sentiment Analysis And Its Application In Translation Based On Decision Tree Algorithm," *International Journal of Maritime Engineering*, vol. 1, no. 1, pp. 395-407, 2024. doi: 10.5750/ijme.v1i1.1371.

[37] R. K. Ramasamy, M. Muniandy and P. Subramanian, "A Predictive Framework for Sustainable Human Resource Management Using tNPS-Driven Machine Learning Models," *Sustainability*, vol. 17, no. 13, 2025. doi: 10.3390/su17135882.

[38] O. I. Villanueva, K. E. Linares, R. O. F. Castañeda and M. C. Carbonell, "Application of Machine Learning Models for Early Detection and Accurate Classification of Type 2 Diabetes," *Diagnostics*, vol. 13, no. 14, 2023. doi: 10.3390/diagnostics13142383.

[39] M. Aslam, D. Ye, A. Tariq, M. Asad, M. Hanif, D. Ndzi, S. A. Chelloug, M. A. Elaziz, M. A. A. Al-Qaness and S. F. Jilani, "Adaptive Machine Learning Based Distributed Denial-of-Services Attacks Detection and Mitigation System for SDN-Enabled IoT," *Sensors*, vol. 22, no. 7, 2022. doi: 10.3390/s22072697.

[40] Y. J. Chang, Y. L. Lin and P. F. Pai, "Support Vector Machines with Hyperparameter Optimization Frameworks for Classifying Mobile Phone Prices in Multi-Class," *Electronics*, vol. 14, no. 11, 2025. doi: 10.3390/electronics14112173.