International Journal of Computer, Network Security and Information System (IJCONSIST)
Vol. 7 No. 1, September 2025

Sentiment Analysis of User Reviews for the LinkedIn Application Using Support Vector Machine and Naïve Bayes Algorithm

Nurissaidah Ulinnuha
Faculty of Sains and Technology
UIN Sunan Ampel, Indonesia
nuris.ulinnuha@uinsa.ac.id

Aisyah Pertiwi
Faculty of Computer Science
University of Pembangunan Nasional
Veteran Jawa Timur, Indonesia
aisyahicha728@gmail.com

Athiyah Fitriyani Basuki
Faculty of Computer Science
University of Pembangunan Nasional
Veteran Jawa Timur, Indonesia
athiyah.fb@gmail.com

Beni Tiyas Kristanti
Faculty of Computer Science
University of Pembangunan Nasional
Veteran Jawa Timur, Indonesia
tiyaskristanti12@gmail.com

Addien Haniefardy
Faculty of Computer Science
University of Pembangunan Nasional
Veteran Jawa Timur, Indonesia
haniefardy.if@upnjatim.ac.id

Muhamad Aris Burhanudin
Faculty of Computer Science
University of Pembangunan Nasional
Veteran Jawa Timur, Indonesia
muhamad_aris.bd@upnjatim.ac.id

Vinza Hedi Satria
Faculty of Computer Science
University of Pembangunan Nasional
Veteran Jawa Timur
Surabaya, Indonesia
vinzasatria.if@upnjatim.ac.id

Corresponding author email: haniefardy.if@upnjatim.ac.id

***Abstract*—Social Networking Sites (SNS) have become integral communication platforms for knowledge sharing and professional connections. LinkedIn, a leading professional network, is widely utilized in today's digital era, primarily by professionals and the business community. This research focuses on analyzing user sentiment on LinkedIn through the application of the Support Vector Machine (SVM) and Naive Bayes methods. Understanding user opinions and satisfaction is important, and sentiment analysis serves as a key tool for this purpose. This study is a comparative analysis of Support Vector Machine (SVM) and Naïve Bayes algorithm for classifying user reviews of the LinkedIn application. Drawing on data from Google Play reviews, this research explores a range of user sentiment towards the LinkedIn platform, including positive, negative and neutral reviews. The application of SVM and Naive Bayes algorithms successfully classifies reviews into relevant sentiment categories. Analyzing 2000 review datasets with an 80% training and 20% testing data split, Support Vector Machines demonstrate an 80% accuracy rate, while Naïve Bayes achieves a 70% accuracy rate. The Support Vector Machines (SVM) algorithm has better accuracy than the Naïve Bayes algorithm based on the test scenarios that have been carried out.**

Keywords— Sentiment Analysis, LinkedIn, Support Vector Machine (SVM), Naive Bayes

I. INTRODUCTION (*HEADING 1*)

The LinkedIn social networking application has witnessed impressive growth in recent years and has become one of the highly influential platforms in the business and professional world. LinkedIn users actively provide reviews covering various aspects, ranging from user interface to functionality and user experience.

Given the abundance of reviews, sentiment analysis is imperative to understand how users respond to and perceive this platform. Two main methods frequently employed in sentiment analysis are Support Vector Machine (SVM) and Naïve Bayes.

SVM, a machine learning algorithm, has proven effective in handling complex classification tasks. By constructing a hyperplane to separate different classes, SVM can recognize intricate patterns in data. Meanwhile, Naïve Bayes, despite its simplicity, offers a robust approach by using Bayesian probability principles to classify text. The main advantage of Naïve Bayes lies in its ability to address class imbalances and its speed in the classification process.

Previous research indicates that the choice of sentiment analysis method should be tailored to the specific task context and the type of data faced. By evaluating their performance in the LinkedIn application context, a deeper understanding of the strengths and weaknesses of each method can be gained. This research also provides practical guidance for the development of improved sentiment analysis tools, enhancing the understanding of the needs and preferences of LinkedIn users.

Hendriyanto and his colleagues have conducted a similar study, focusing on sentiment analysis of reviews for the Mola application on the Google Play Store. This research utilized the Support Vector Machine (SVM) algorithm with a dataset consisting of 520 entries. The sentiment classification results revealed 312 positive sentiments and 208 negative sentiments [1].

Several previous studies have also delved into sentiment analysis. In 2019, Chaithra VD conducted research to analyze user responses on YouTube to a specific smartphone model, the LG G7. The dataset comprised 6,248 comments extracted from a mobile phone unboxing video uploaded on YouTube. Chaithra VD employed a hybrid method combining the Naïve Bayes algorithm and the VADER Sentiment method. VADER Sentiment was used in the data labeling process, while the Naïve Bayes algorithm classified comments into three classes: positive, negative, and neutral. The research results showed an accuracy rate of 79.78% [2].

With the growing importance of sentiment analysis in the professional social networking context, this research not only contributes to academic literature but also opens opportunities for the development of more advanced applications to understand and enhance user experiences on professional platforms. Therefore, it is anticipated that this research will not only be a valuable contribution to academic literature but will also unlock the potential for the development of more sophisticated applications in understanding and improving user interactions in a professional environment.

II. An Overview of Support Vector Machine and Naïve Bayes

A. Support Vector Machine

Support Vector Machine (SVM) is a technique for making predictions, both in regression and classification cases [3]. The SVM technique is employed to obtain an optimal separator function (hyperplane) to distinguish observations with different target variable values [4]. This hyperplane can take the form of a line in two dimensions and a flat plane in multiple dimensions.

B. Naïve Bayes Classifier

The Naïve Bayes Classifier is a classification method rooted in Bayes' theorem. It utilizes the probability and statistics methods proposed by the English scientist Thomas Bayes. The theorem involves predicting future probabilities based on past experiences, earning it the name Bayes' Theorem. The main characteristic of the Naïve Bayes Classifier is its strong (naïve) assumption of independence among each condition/event.

C. Related Research

The use of support vector machine and naive Bayes algorithms is used to analyze sentiment data on user reviews of the MySAPK BKN application on the Google Play Store. The results show that the majority of reviews have negative sentiments (55.7%), while reviews with positive sentiments reach 44.3%. Naïve Bayes achieved an accuracy rate of 92.47%, while SVM achieved a higher accuracy rate of 94.14% [5].

In previous research, the support vector machine method showed good performance when analyzing by.u provider sentiment. Data was collected through web scraping of more than 8,900 reviews. Data was labeled with sentiment based on review ratings, with ratings 1 and 2 categorized as negative, ratings 4 and 5 as positive, and rating 3 labeled manually. The results show that reviews tend to be positive with expressions of satisfaction with by.U services, including discussions about cheap quota prices and internet speed. Meanwhile, negative reviews contain complaints about purchasing internet packages, slow delivery of SIM cards, and problems with the by.U application. The use of the TF-IDF and SVM methods in sentiment classification produces good performance, with an average accuracy of around 84.7%, precision around 84.9%, recall around 84.7%, and f-measure around 84.8%. The highest accuracy was achieved on the second fold, reaching 86.1% [6].

III. Methodology

A. Data Collection

The data used in this research are reviews of the LinkedIn application obtained through reviews on Google Play. To retrieve a dataset, the process begins with installing packages and importing basic libraries such as pandas and numpy to help with the dataset retrieval process. Review data collection is limited to Indonesian language reviews, and only two thousand of the most relevant data is taken. The data taken consists of four columns, namely the username column which contains the user name of the review author, the score column which contains the rating of the application given by the user, the at column which contains the time the review was written and the content column contains reviews from Linkedin application users.

B. Data Preprocessing

Preprocessing is a crucial stage in data mining aimed at cleaning data. This involves handling incomplete, noisy, and inconsistent data. Data Cleansing is an essential process in cleaning review data for model development. The cleaning steps involve handling abbreviations, removing punctuation, eliminating numbers, and discarding single letters without sentiment value. Detailed explanations for each step are as follows:

- Abbreviation Handling: Abbreviations in the text are often challenging for systems to understand. In this study, we utilize a script code containing a list of commonly used abbreviations in Indonesian reviews. These abbreviations are then converted to their corresponding base words.
- Punctuation Removal: Punctuation in the text often lacks relevance in text analysis and needs to be removed.
- Number Removal: Numbers in the text indicate variable quantities but do not provide useful sentiment information. Therefore, these numbers need to be removed.
- Single Letter Removal: Single letters are often a result of pre-processing or user writing errors. These single letters do not contribute to sentiment analysis and should be eliminated.

- Duplicate Letter Removal: In datasets, character repetition within words is common. This repetition can introduce noise into the analyzed data, so it needs to be addressed. To remove URLs from other sites, we use the re library package and the sub function. In this step, we specify the string pattern representing repeated letters, the replacement character for repeated letters, and the variable to be applied.

An example of the cleaning process can be seen in Table I.

Case folding is a procedure used to standardize the use of uppercase and lowercase letters in a text document. Its goal is to simplify the search process, as text in documents often has inconsistencies in the use of uppercase or lowercase letters. Therefore, case folding is crucial in transforming the entire text in the document into a standard format, namely lowercase. An example of case folding application can be seen in Table II.

TABLE I. EXAMPLE OF REVIEW DATA BEFORE AND AFTER CLEANING

No	Before Cleaning	After Cleaning
1.	Aplikasi yg sangat bagus ,sangat membantu banget ðŸ‘•	Aplikasi yg sangat bagus sangat membantu banget
2.	2 Kali download 2 kali daftar hasilnya sangat mengecewakan ga ada hasilnya	Kali download kali daftar hasilnya sangat mengecewakan ga ada hasilnya
3.	Perlu perbaikan, karena kadang susah dibuka	Perlu perbaikan karena kadang susah dibuka

TABLE II. EXAMPLE OF REVIEW DATA BEFORE AND AFTER CASE FOLDING

No	Before Case Folding	After Case Folding
1.	Aplikasi yg sangat bagus sangat membantu banget	Aplikasi yg sangat bagus sangat membantu banget
2.	aplikasi yg sangat bagus sangat membantu banget	aplikasi yg sangat bagus sangat membantu banget
3.	Kali download kali daftar hasilnya sangat mengecewakan ga ada hasilnya	Kali download kali daftar hasilnya sangat mengecewakan ga ada hasilnya

Tokenizing is the step where sentences are segmented into separate word units. Tokenizing is crucial in supporting the model to predict sentiment labels for each review. It breaks down each sentence into individual words, allowing the model to review each word separately, ultimately enhancing understanding of the meaning and context of the review more comprehensively. Tokenizing allows the model to break down the review into more understandable and interpretable elements. An example of tokenizing implementation can be seen in Table III.

TABLE III. EXAMPLE OF REVIEW DATA BEFORE AND AFTER TOKENIZING

No	Before Tokenizing	After Tokenizing
1	aplikasi yg sangat bagus sangat membantu banget	['aplikasi', 'yg', 'sangat', 'bagus', 'sangat', 'membantu', 'banget']
2	kali download, kali daftar, hasilnya sangat mengecewakan, ga ada hasilnya	['kali', 'download', 'kali', 'daftar', 'hasilnya', 'sangat', 'mengecewakan', 'ga', 'ada', 'hasilnya']
3	perlu perbaikan karena kadang susah dibuka	['perlu', 'perbaikan', 'karena', 'kadang', 'susah', 'dibuka']

The next step is stopword removal, which involves eliminating stopwords from the reviews. Stopwords are frequently occurring words in the text that do not have specific meaning in the context of sentiment analysis. Therefore, stopwords do not significantly contribute to determining the sentiment of the review. An example of stopword removal can be seen in Table IV.

The final preprocessing step is stemming, which involves removing prefixes or suffixes (both at the beginning and end of words) to reduce words to their base form, even though this base form may not have the same meaning as the original word. An example of stemming implementation can be seen in Table V.

TABLE IV. EXAMPLE OF REVIEW DATA BEFORE AND AFTER STOPWORD REMOVAL

No	Before Stopword Removal	After Stopword Removal
1	['aplikasi', 'yg', 'sangat', 'bagus', 'sangat', 'membantu', 'banget']	['aplikasi', 'bagus', 'membantu']
2	['kali', 'download', 'kali', 'daftar', 'hasilnya', 'sangat', 'mengecewakan', 'ga', 'ada', 'hasilnya']	['kali', 'download', 'kali', 'daftar', 'hasilnya', '', 'mengecewakan', '', '', 'hasilnya']
3	['perlu', 'perbaikan', 'karena', 'kadang', 'susah', 'dibuka']	['perlu', 'perbaikan', 'karena', 'kadang', 'susah', 'dibuka']

TABLE V. EXAMPLE OF REVIEW DATA BEFORE AND AFTER STEMMING

No	Before Stemming	After Stemming
1	['aplikasi', 'bagus', 'membantu']	['aplikasi', 'bagus', 'bantu']
2	['kali', 'download', 'kali', 'daftar', 'hasilnya', '', 'mengecewakan', '', '', 'hasilnya']	['kali', 'download', 'kali', 'daftar', 'hasil', 'mengecewakan', 'hasil']
3	['perlu', 'perbaikan', 'karena', 'kadang', 'susah', 'dibuka']	['perlu', 'perbaikan', 'karena', 'kadang', 'susah', 'buka']

After going through the cleaning, case folding, tokenizing, stopword removal, and stemming stages, sentiment labeling is performed on the data using VADER Sentiment. In the labeling process, a function will be defined to analyze sentiment based on the compound value. If the compound value is greater than or equal to 0.05, the sentiment is considered positive. If the compound value is between -0.05

and 0.05, the sentiment is considered neutral, and if the polarity value is less than or equal to -0.05, the sentiment is considered negative. Next, the TF-IDF (Term Frequency-Inverse Document Frequency) weighting stage is carried out. An example of sentiment labeling implementation can be seen in Table VI.

TABLE VI. SENTIMENT LABELING RESULTS

No	Data	Label
1	good app helps	Positive
2	times download times list of results disappointing results	Negative
3	needs repair because sometimes it's hard to open	Negative

C. Modelling

After going through the data preparation process in the previous stage, the data will be used as input for the classification algorithm. Data will be divided into two types: training data and testing data. The training data will cover 80% of the total data, while the testing data will cover 20%. The dataset will be categorized into three different sentiment categories: positive sentiment, neutral sentiment, and negative sentiment. In this research, two types of algorithms will be used for comparison, namely Naïve Bayes and SVM.

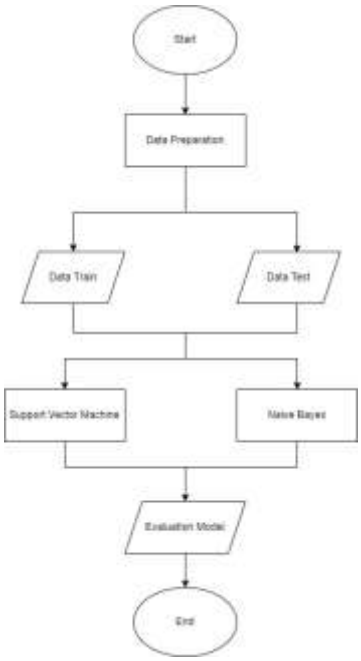

Fig. 1. Classification Process

D. Evaluation

The evaluation phase is the stage of evaluating the performance of the modeling and calculations that have been carried out by measuring the performance of the classification algorithm used. In this research, performance is measured using accuracy, precision, recall, F1-score. These results will show the use of the most appropriate algorithm for the modeling that has been proposed previously.

IV. RESULT

After obtaining the required 2000 data using crawling methods, the data is then categorized into three sentiment categories. There are 1156 positive data, 569 negative data, and 275 neutral data, totaling 2000 data in the entire dataset. Next, the data is divided using an 80% to 20% ratio for training and testing data, respectively. The next step involves the classification of data using the Support Vector Machine and Naïve Bayes Classifier algorithms. The Support Vector Machine classification begins with searching for the best parameters using Grid Search, employing SVM as the model in a pipeline. The explored parameters include the kernel type (linear or rbf), the value of C, gamma, and the decision function shape. The results from Grid Search are used to train the SVM model on the training data, and the best-found model is saved for further use. Subsequently, Naïve Bayes classification is carried out by training the model using MultinomialNB from scikit-learn. After completing the model training process for the Support Vector Machine and Naïve Bayes algorithms, the results are obtained and can be seen in Table VII and VIII.

The test results show that the support vector machine method has better accuracy, precision, recall and f1-score results compared to the naive Bayes method. the accuracy of the Support Vector Machine is greater with an accuracy level of 0.80, while the accuracy of Naïve Bayes is 0.70.

TABLE VII. CLASSIFICATION REPORT SUPPORT VECTOR MACHINE

Class	Precision (%)	Recall (%)	F1-Score (%)
Negative	0.81	0.73	0.77
Neutral	0.67	0.39	0.49
Positive	0.81	0.93	0.87
Accuracy (%)			0.80

TABLE VIII. CLASSIFICATION REPORT NAÏVE BAYES

Class	Precision (%)	Recall (%)	F1-Score (%)
Negative	0.64	0.64	0.64
Neutral	0.77	0.77	0.77
Positive	0.70	0.70	0.70
Accuracy (%)			0.70

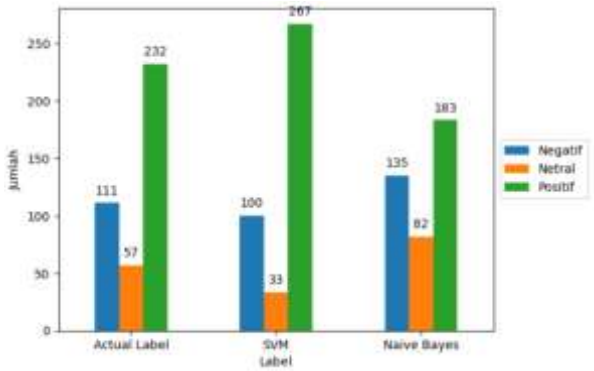

Fig. 2. Instance comparison visualization

In terms of the number of instances successfully predicted for each class, SVM has different results from Naive Bayes. In the Negative class, SVM succeeded in predicting 100 instances, while Naive Bayes managed 135 instances. For the Neutral class, SVM predicts 33 instances, while Naive Bayes predicts 82 instances. In the Positive class, SVM has predictions of 267 instances, while Naive Bayes has 183 instances. The results of this comparison show that SVM is better than Naive Bayes because it succeeded in predicting more instances overall.

V. CONCLUSION

This paper shows that the Support Vector Machines (SVM) algorithm has better performance than the Naïve Bayes algorithm based on test scenarios that have been carried out on 2000 LinkedIn application review data, with an 80:20 data division, the accuracy of the Support Vector Machines algorithm is 80%, while the Naïve Bayes algorithm has an accuracy value of 70%.

REFERENCES

[1] M. D. Hendriyanto, A. A. Ridha and U. Enri, "Analisis Sentimen Ulasan Aplikasi Mola Pada Google Play Store Menggunakan Algoritma Support Vector Machine," *INTECOMS,* vol. 5, no. 1, pp. 1-7, 2022.

[2] C. V. D., "Hybrid approach: naive bayes and sentiment VADER for analyzing sentiment of mobile unboxing video comments," *International Journal of Electrical and Computer Engineering (IJECE),* vol. 9, no. 5, pp. 4452-4459, 2019.

[3] S. Fachrurrazi, "Penggunaan Metode Support Vector Machine (SVM) untuk Mengklasifikasi dan Memprediksi Angkutan Udara Jenis Penerbangan Domestik dan Penerbangan Internasional di Banda Aceh," 2011.

[4] G. Williams, Data Mining with Ratle and R: The Art of Excavating Data for Knowledge Discovery, NEW YORK: Springer-Verlag, 2011.

[5] R. I. Alhaqq, I. M. . K. Putra and Y. Ruldeviyani, "Sentiment Analysis toward the Use of MySAPK BKN Application in Google Play Store," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi,* vol. 11, no. 2, pp. 105-113, 2022.

[6] S. Fransiska, R. and A. . I. Gufroni , "Sentiment Analysis Provider by.U on Google Play Store Reviews with TF-IDF and Support Vector Machine (SVM) Method," *Scientific Journal of Informatics,* vol. 7, no. 2, 2020.