Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th February 2026. Vol. 5. No. 2; e-ISSN: 2808-4519

Sentiment Analysis of Mamikos Application User Reviews using the Naive Bayes Classifier Algorithm

Amsal Tampubolon[1]*, Erich Ricardo[2], Sardo Pardingotan Sipayung[3]

[1,2,3]*Universitas Katholik Santo Thomas*
amsaltampubolon24@gmail.com[1*], ericmarbun3@gmail.com[2]

Abstract

Mamikos, a boarding house search application, has grown to be among Indonesia's most well-liked sites. User reviews on the Google Play Store provide valuable data regarding user satisfaction. However, the large volume of reviews makes it difficult for developers to analyze sentiments manually. This study seeks to do sentiment analysis on evaluations of the Mamikos application to categorize user thoughts as positive or negative sentiments. The method used is the Naive Bayes Classifier with TF-IDF (Term Frequency-Inverse Document Frequency) feature weighting. The research stages include data collection (scraping), text preprocessing (including cleaning and stopword removal using the Sastrawi library), and classification. The test results show that the Naive Bayes algorithm can classify sentiments with an accuracy of 87.96 %. This research is expected to provide insights for application developers regarding aspects that need improvement based on user complaints.

Keywords: *Sentiment Analysis; Mamikos; Naive Bayes; TF-IDF; User Reviews.*

1. Introduction

The swift progression of information technology has markedly altered numerous sectors in Indonesia, including the property and accommodation rental industry. Traditionally, finding a boarding house (*kost*) was a time-consuming process that required physical surveys and manual comparisons. Today, mobile applications have simplified this process by connecting property owners with potential tenants digitally. *Mamikos* has emerged as one of the leading platforms in this domain, providing comprehensive data on boarding houses across Indonesia [1].

As the user base grows, the volume of user feedback generated on distribution platforms like the Google Play Store increases exponentially. User reviews are a critical source of data for developers, as they contain valuable information regarding user satisfaction, feature requests, and technical issues such as login failures or application bugs [2]. However, this feedback is unstructured and massive in volume. Manually reading and analyzing thousands of reviews to extract meaningful insights is inefficient, prone to human error, and impractical for real-time decision-making.

To address this challenge, Sentiment Analysis, also known as Opinion Mining, is employed. Sentiment analysis is a computational study of people's opinions, sentiments, evaluations, attitudes, and emotions from written language [3]. By utilizing Natural Language Processing (NLP) techniques, unstructured text data can be automatically classified into sentiment categories, such as positive or negative.

Various algorithms can be used for text classification, including Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Naive Bayes. Among these, the Naive Bayes Classifier is widely chosen for text mining tasks due to its computational efficiency and effectiveness in handling high-dimensional data [4]. Naive Bayes is a probabilistic classifier derived from Bayes' theorem, predicated on the premise of feature independence. Despite its simplicity, it often outperforms more complex models in text classification scenarios, especially when combined with TF-IDF (Term Frequency-Inverse Document Frequency) for feature weighting [5].

Several previous studies have applied sentiment analysis to mobile application reviews. However, specific research focusing on the Mamikos application using the Naive Bayes method with the latest dataset is still relevant to understand current user satisfaction trends. Therefore, this study aims to implement the Naive Bayes Classifier algorithm to analyze the sentiment of Mamikos application users. The specific objective is to measure the accuracy of the algorithm in classifying reviews into positive and negative sentiments, thereby providing actionable insights for application developers to improve service quality.

2. Method

Research Flow The research methodology follows a systematic approach. The process begins with data acquisition from the Google Play Store, followed by data labeling and preprocessing. The preprocessed data is then transformed into numerical vectors using TF-IDF weighting. Finally, the Naive Bayes algorithm is applied to classify the sentiments, and the model is evaluated using confusion matrix metrics.

Fig 1: Research Methodology Flowchart

Naive Bayes Classifier Theory Naive Bayes is a probabilistic classification method based on Bayes' Theorem. It assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature (independence assumption) [6]. The probability of a class c given a document d is calculated using Equation (1):

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)} \quad (1)$$

Where:

1. $P(c|d)$ is the posterior probability of class c given predictor (document) d.
2. $P(c)$ is the prior probability of class c.
3. $P(d|c)$ is the likelihood which is the probability of predictor d given class c.
4. $P(d)$ is the prior probability of predictor d.

TF-IDF Weighting Term Frequency-Inverse Document Frequency (TF-IDF) is used to determine the weight of each word in the document. It combines Term Frequency (TF), which measures how frequently a term appears in a document, and Inverse Document Frequency (IDF), which measures how important a term is [7]. The formula is given in Equation (2):

$$W_{i,j} = TF_{i,j} \times \log\left(\frac{N}{DF_i}\right) \quad (2)$$

Where N is the total number of documents and $\mathrm{DF_i}$ is the number of documents containing term i

Getting the Data The sample used in this study is made up of reviews of the Mamikos app that were gathered from the Google Play Store. The data was acquired using the *google-play-scraper* library in Python. The attributes retrieved include the username, rating score (1-5), and the review content. The scraping process targeted the most recent reviews to ensure the relevance of the data.

Data Labeling The labeling process was performed automatically based on the rating score given by the user (Lexicon-based labeling). Reviews with a rating of 4 or 5 were labeled as "Positive", while reviews with a rating of 1 or 2 were labeled as "Negative". Reviews with a rating of 3 (neutral) were excluded to focus on binary classification.

Preprocessing Before classification, the textual data underwent several preprocessing steps to reduce noise and improve model performance:

1. Case Folding: Converting all characters to lowercase to ensure uniformity.
2. Cleaning: Removing numbers, punctuation, special characters, and emojis using Regular Expressions (Regex), leaving only alphabetic characters.
3. Stopword Removal: Removing common Indonesian words that do not carry significant meaning (e.g., "dan", "yang", "di") using the *Sastrawi* library [8].

Feature Extraction (TF-IDF) The cleaned text data was converted into numerical vectors using the Term Frequency-Inverse Document Frequency (TF-IDF) method. TF-IDF evaluates how important a word is to a document in a collection or corpus. It increases the weight of words that appear frequently in a document but decreases the weight of words that are used a lot in all papers.

Classification The classification was performed using the Multinomial Naive Bayes algorithm. This variant of Naive Bayes is specifically designed for text classification with discrete features (like word counts or TF-IDF vectors). The train_test_split function from the scikit-learn library was used to divide the dataset into training and testing sets with an 80:20 split.

3. Results and Discussion

Dataset Distribution The analysis of the dataset reveals a significant class imbalance. As observed in the testing set, the number of good reviews is much higher than the number of bad reviews. This reflects that the majority of users are satisfied with the application, although a notable portion still experiences issues.

Word Analysis To understand the context of the reviews, a WordCloud visualization was generated.

Fig 2: WordCloud of Mamikos User Reviews.

Fig 1 shows the most frequent words appearing in the reviews. Words such as "bagus" (good), "mudah" (easy), and "bantu" (helpful) are dominant, indicating positive sentiment. Meanwhile, words related to technical issues may appear in smaller frequencies but are crucial for negative sentiment detection.

How the Model Works The Confusion Matrix and measures like Accuracy, Precision, Recall, and F1-Score were used to test how well the Naive Bayes classifier worked.

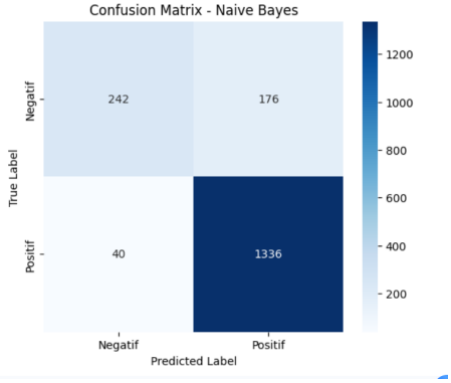

Fig 3: Confusion Matrix of Naive Bayes Classifier

The classification results are summarized in Table 1.

Table 1. Classification Report

Class	Precision	Recall	F1-Score	Support
Negative	0.86	0.58	0.69	418
Positive	0.88	0.97	0.93	1376
Accuracy			0.88	1794

Detailed Analysis of Classification Based on the Confusion Matrix in Figure 2, the model successfully predicted 1,336 reviews correctly as Positive (True Positive) and 242 reviews correctly as Negative (True Negative). However, there were 176 False Positives, where negative reviews were incorrectly classified as positive. This error typically occurs when a negative review contains polite words or ambiguous phrases that are often found in positive reviews (e.g., "The app is good but needs fix"). Conversely, the False Negative rate was relatively low (40 data).

The disparity between Precision (0.86) and Recall (0.58) for the Negative class indicates that while the model is precise when it predicts negative sentiment, it struggles to find all negative instances. This is a common phenomenon in imbalanced datasets where the majority class (Positive) dominates the learning process.

Table 2. Sample of Classification Results vs Actual Labels

Review_Text	Actual_Label	Predicted_Label	Status
aplikasi mamikos diandalkan mencari koskosan ribet aplikasi sudah komplit			
mudah digunakan merasa sangat terbantu aplikasi untuk menemukan koskosan saya inginkan harus ribet bisa memangkas biaya yang tidak perlu	Positif	Positif	Correct
sangat membantu nyari kos sesuai budget keinginan mahasiswa rantau saya aplikasi keren sangat membantu trims	Positif	Positif	Correct
apknya bagus nih gk perlu susah kesana kemari nyari alamat buat cari kosan cukup download apk tinggal jalankkan apk kita cari tempat sewa kosan ada seluruh indonesia pokoknya buruan download recomendlah guys	Positif	Positif	Correct
aplikasi aktifbalasan chat tidak pernah balasakun terblock bagaimanaterima kasih	Negatif	Positif	Incorrect
banyak penyewa kost bodong cuma buat habisin saldo iklan biayanya sesuai performa mamikos	Negatif	Positif	Incorrect

Table 2 demonstrates samples of the classification results. While the model correctly identifies clear sentiments (samples 1-3), it faces challenges with mixed sentiments, as seen in sample 4-5.

4. Conclusion

This study successfully implemented the Naive Bayes Classifier for the sentiment analysis of Mamikos application reviews. The model achieved a high accuracy of 87.96%. The results indicate that the majority of users have a positive perception of the application. The use of TF-IDF and basic preprocessing proved effective in handling Indonesian text data. However, the lower recall on negative data suggests that future research could benefit from handling class imbalance using techniques like SMOTE (Synthetic Minority Over-sampling Technique) or using more complex algorithms like SVM or LSTM to better capture negative sentiments. Nevertheless, the current system is sufficiently effective to provide an automated overview of user satisfaction.

References

[1] Fitri Nur Utami, Hani Dewi Ariessanti, Riya Widayanti, and Arief ichwani, "Klasifikasi Sentimen untuk Prediksi Churn Pengguna Aplikasi Mamikos Menggunakan Support Vector Machine (SVM) dan Naïve Bayes," *J. Ilmu Multidisiplin*, vol. 4, no. 4, pp. 2213–2223, 2025.

[2] K. Vasa, "Sentiment Analysis of Google Play Store Reviews Using Naive Bayes and Decision Tree," *Int. J. Comput. Appl.*, vol. 182, no. 48, pp. 15–19, 2019.

[3] B. Liu, *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 2012.

[4] V. Narayanan, I. Arora, and A. Bhatia, "Fast and Accurate Sentiment Classification Using an Enhanced Naive Bayes Model," in *Intelligent Data Engineering and Automated Learning -- IDEAL 2013*, 2013, pp. 194–201.

[5] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[6] J. Han, M. Kamber, and J. Pe, *Data Mining: Concepts and Techniques*. 2011. doi: 10.1016/C2009-0-61819-5.

[7] M. Allahyari *et al.*, "A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques," 2017, [Online]. Available: http://arxiv.org/abs/1707.02919

[8] F. Z. Tala, "A Study of Stemming Effects on Information Retrieval in Indonesian Language," Institute for Logic, Language and Computation, Universiteit van Amsterdam, Amsterdam, Netherlands, 2003.

[9]. Muammar Khadapi, & Pakpahan, V. M. (2024). Analisis Sentimen Berbasis Jaringan LSTM dan BERT terhadap Diskusi Twitter tentang Pemilu 2024. *JUKI : Jurnal Komputer Dan Informatika*, *6*(2), 130–137. Retrieved from https://ioinformatic.org/index.php/JUKI/article/view/681