Available online to ejournal.pnc.ac.id

Journal of Innovation Information Technology and Application

Journal Page is available to https://ejournal.pnc.ac.id/index.php/jinita

Sentiment Analysis of Universitas Jember's Sister for Student Application Using Gaussian Naive Bayes and N-Gram

Mochamad Bagoes Alfarazi [1], Muhammad 'Ariful Furqon [2], Harry Soepandi [3]*
[1,3] Department of Information Systems, Universitas Jember, Jember
[2] Department of Informatics, Universitas Jember, Jember
email: [1] 172410101039@mail.unej.ac.id, [2] ariful.furqon@unej.ac.id, [3] harrysupandi@unej.ac.id*

ARTICLE INFO

Article history:
Received 24 July 2024
Revised 29 October 2024
Accepted 9 December 2024
Available online 30 December 2024

Keywords:
Gaussian Naive Bayes
Cross Validation
SFS
Application Review Analysis
N-Gram

IEEE style in citing this article :
M. B. Alfarazi , M. 'Ariful Furqon , and H. Soepandi , "Sentiment Analysis of Universitas Jember's Sister for Student Application Using Gaussian Naive Bayes and N-Gram ," Journal of Innovation Information Technology and Application (JINITA), vol. 6, no. 2, pp. 101–108, Dec. 2024..

ABSTRACT

This research aims to classify sentiment in reviews of the Universitas Jember Sister for Student application on Google Play Store, a vital student platform. The primary challenge tackled is the automated identification of positive and negative user sentiments. The study employs the Gaussian Naive Bayes method for classification and uses N-Gram techniques for sentiment analysis. The dataset consists of 1097 reviews, with 673 negative and 424 positive reviews, after removing 86 neutral spam reviews. The data is divided into 80% training data (877 reviews) and 20% test data (220 reviews). Gaussian Naive Bayes is used for modeling and combined with TF-IDF vectorization. The findings reveal that the Gaussian Naive Bayes model achieves an accuracy of 68%, precision of 68%, and recall of 71% on the test data. N-Gram analysis shows frequent occurrences of words like "bisa", "bagus", and "aplikasi" in positive sentiments, while "bisa", "hp", and "absen" are prevalent in negative sentiments. The study concludes that the Gaussian Naive Bayes model effectively classifies sentiment in application reviews, with the potential for further performance improvements.

1. INTRODUCTION

In the current era of information technology, mobile applications have become an essential part of daily life, especially in education [1]. Mobile applications help students access information even when they are in different locations. Universitas Jember also employs mobile applications to assist students, faculty, and staff in their educational system. One application is Sister for Student, which includes course registration (KRS), course planning, payment information, academic results, and attendance tracking [2]. However, the sister for Student application has several drawbacks and weaknesses, including bugs that disrupt its usage. Many students are dissatisfied with the application, citing sudden implementation, frequent crashes of features, and inaccurate information. Reviews on the Google Play store give it a rating of 3.2 out of 3,880 reviews, indicating widespread complaints about the application's performance.

User reviews on the Google Play Store consist of ratings and text comments, each offering unique insights into the overall user experience [3]. While ratings provide a quick numerical summary of user satisfaction, text reviews delve deeper, detailing specific experiences and feedback [4]. In sentiment analysis, the primary objective is to classify the polarity of such textual data—whether a review is positive or negative—at the document, sentence, or opinion level [5], [6]. Various machine learning techniques are commonly employed, including Naïve Bayes, Maximum Entropy, Support Vector Machines, and K-Nearest Neighbor [7]. Among these, Gaussian Naive Bayes (GNB) has shown notable effectiveness, consistently achieving high accuracy levels, with studies reporting an average accuracy rate of 97.48% [8]. This performance makes GNB a compelling choice for accurately capturing user sentiment in diverse applications.

*) Corresponding Author : ariful.furqon@unej.ac.id

This research adopts the GNB method as the basis for sentiment classification for several compelling reasons supported by previous studies. Previous research demonstrates that GNB has high accuracy in sentiment analysis related to COVID-19 vaccines, reaching 97.48%, proving its effectiveness in text classification [9]. Additionally, a study highlights GNB's superiority in ranking Yelp reviews with an accuracy of 86.7%, underscoring GNB's consistent performance in various application contexts [10]. Meanwhile, other studies emphasize the variation in machine learning algorithm performance depending on the dataset used, with GNB often showing competitive results [11]. However, this study introduces a novel approach by combining sentiment analysis using N-Gram techniques and TF-IDF vectorization to enhance understanding of dominant words or phrases in user reviews. Moreover, the implementation of grid search and 10-fold cross-validation is used to optimize model parameters [12], differing from conventional approaches in previous research. The focus on the sister for Student application also adds a unique context, given its relevance in the educational world and actual use by students, faculty, and staff at Universitas Jember.

This study addresses a gap by applying the GNB method to analyze user sentiment specifically within the educational sector, focusing on the Sister for Student application used by students, faculty, and staff at Universitas Jember. Prior studies have demonstrated the effectiveness of GNB in sentiment analysis across domains such as social media and consumer product reviews; however, few have explored its utility in educational technology. Additionally, previous research on GNB often lacks integrated model optimization approaches, such as grid search and 10-fold cross-validation, implemented here to enhance the model's performance metrics—accuracy, precision, and recall. This study uses TF-IDF vectorization, N-Gram analysis, and optimization techniques to refine the sentiment classification, specifically addressing educational app users' unique feedback patterns and requirements. This research contributes to the existing body of knowledge by providing a domain-specific, optimized sentiment analysis model for an academic application.

2. METHOD

This study employs a quantitative and sentiment analysis approach to classify user reviews of the sister for Student application on the Google Play store. The Gaussian Naive Bayes (GNB) method was chosen as the primary classification algorithm due to its simplicity and efficiency in handling text classification problems [13]. The TF-IDF technique is also used for feature extraction [14], and the Confusion Matrix is employed for model evaluation [15]. N-gram analysis is also applied to gain insights into frequently occurring words or phrases in positive and negative reviews [16]. The research stages are illustrated in the diagram in Figure 1.

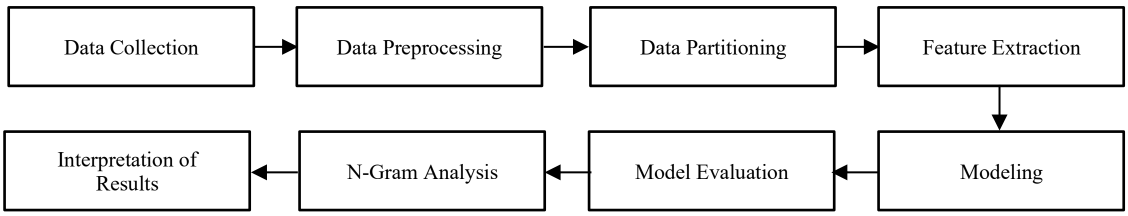


Figure 1. Research Stages

3.1. Data Collection

The data used in this study was collected through crawling user reviews of the Sister for Student (SFS) application, utilized at Universitas Jember and available on the Google Play store platform. This crawling process aimed to gather user reviews from May 26, 2018, to June 22, 2023. A total of 1147 reviews were collected for analysis in this study.

3.2. Data Preprocessing

1. Actual Value Validation

   Actual value validation is a crucial stage in the sentiment analysis of the Sister for Student application reviews. This process assigns class labels to each review based on the contained sentiment. Experts conduct the assessment proficiently to understand the nuances and meanings of reviews. The experts label reviews as “Positive” if the review is satisfactory, “Negative” if it

DOI: doi.org/10.35970/jinita.v6i2.2400

expresses dissatisfaction or criticism, and “Neutral” for reviews outside the context of the application. The actual value validation results show the distribution pattern of user sentiments towards the Sister for Student application. Most reviews are negative, particularly in one-star ratings with 420 reviews and three-star ratings with 109 reviews, indicating numerous user complaints. Conversely, positive sentiment reviews are most prevalent in five-star ratings, with 339 reviews, and four-star ratings, with 58 reviews, indicating user satisfaction. Neutral sentiment reviews are relatively fewer compared to positive and negative sentiments.

2. Text Processing
Data preprocessing is done to clean and prepare the data before further analysis. The preprocessing stages include:
    a. Tokenization: Breaking down the review text into words or tokens.
    b. Normalization: Convert all text to lowercase and remove special characters.
    c. Stopwords Removal: Removing common words that do not have significant meaning in sentiment analysis, such as “and,” “or,” “that.”
    d. Stemming: Converting words to their base form to reduce word variations.

3.3. Data Partitioning

Data partitioning is a crucial stage in preparing sentiment analysis for the sister for Student application, where reviews are divided into training and test data. From an initial total of 1183 reviews, consisting of 673 negative reviews, 424 positive reviews, and 86 neutral reviews (removed due to spam), 1097 reviews remain for analysis. The data is proportionally divided, with 80% used as training data and 20% as test data. The result of the data partitioning process is illustrated below.

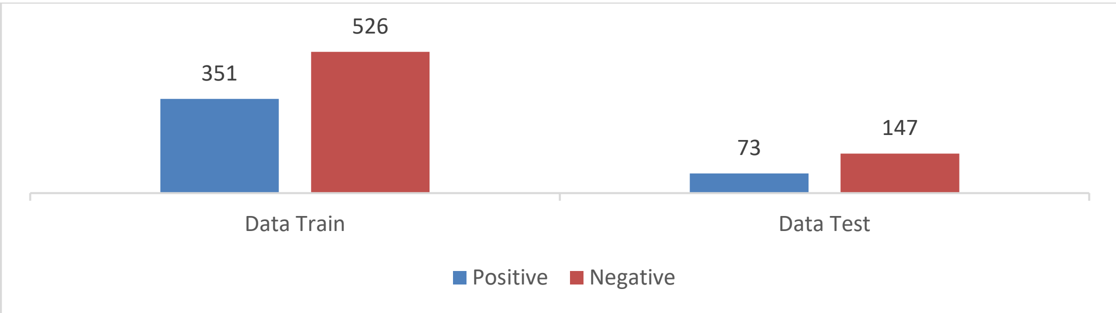


Figure 2. Results of Split Testing Data and Training Data

The data labeling process for sentiment analysis in the SISTER for Student application involves structured steps to ensure accurate sentiment classification. This systematic labeling process supports effective sentiment analysis by ensuring that training and test data are differentiated into “positive” and “negative” sentiments, creating a robust foundation for building an accurate SISTER for Student application model.

3.4. Feature Extraction

Feature extraction transforms raw text data into a numerical format that machine learning algorithms can process. The goal is to identify and filter significant information from the text that can be used as input for the model. Feature extraction is performed using the Term Frequency-Inverse Document Frequency (TF-IDF) method, which evaluates the importance of a word in a document relative to the entire corpus. TF-IDF calculates the weight of each word based on two main components:

1. Term Frequency (TF): Measures how often a word appears in a document. The more frequently a word appears, the higher its TF value.
2. Inverse Document Frequency (IDF): Measures the importance of a word by calculating the inverse of the number of documents containing that word. Common words across documents (such as “and,” “or”) will have a low IDF value, while rare words will have a high IDF value.

3.5. Modeling

DOI: doi.org/10.35970/jinita.v6i2.2400

Modeling is conducted using the Gaussian Naive Bayes (GNB) algorithm. Gaussian Naive Bayes is a variant of the algorithm for classifying data based on Gaussian distribution, assuming independence between each feature to predict the output variable. This algorithm combines predictions from each feature to generate the probability of the output variable within each class, and it is suitable for numerical or continuous data. Known for its speed and ease of implementation, Gaussian Naive Bayes is frequently used in artificial intelligence applications [17]. The following equations are used in the Gaussian Naive Bayes classification process, including the calculation of prior probability [18] as explained in Equation 2 below.

$$P(C_k) = \frac{N_k}{N} \tag{1}$$

$$\mu_{k,j} = \frac{1}{N_k} \sum_{i=1}^{N} x_{i,j}\, I(y_i = k) \tag{2}$$

$$\sigma_{k,j}^2 = \frac{1}{N_k} \sum_{i=1}^{N} \left(x_{i,j} - x_{k,j}\right)^2 I(y_i = k) \tag{3}$$

$$P(x_{i,j}|C_k) = \frac{1}{\sqrt{2\pi\sigma_{k,j}^2}} exp\left(-\frac{(x_{i,j} - \mu_{k,j})^2}{2\sigma_{k,j}^2}\right) \tag{4}$$

$$P(C_k|x_i) = \prod_{j=1}^{d} P(x_{i,j}|C_k) \tag{5}$$

Gaussian Naive Bayes uses several key equations in its classification process. First, equation (1) calculates the prior probability $P(C_k)$, which represents the probability of class k before considering evidence from the data, where N_k is the number of data points in class k and N is the total number of data points. Next, equations (2) and (3) calculate the mean $\mu_{k,j}$ and variance $\sigma_{k,j}^2$ of feature j in class k, with $x_{i,j}$ being the value of feature j in data point i, y_i being the class of data point i, and $I(y_i = k)$ being the indicator function. Equation (4) describes the likelihood probability $P(x_{i,j}|C_k)$ of feature j in data point iii for class k, using a Gaussian distribution with the values $\mu_{k,j}$ and $\sigma_{k,j}^2$. Finally, equation (5) calculates the posterior probability $P(x_{i,j}|C_k)$, which is the probability of class k after considering all features x_i.
Modeling Steps

1. Implementation of Gaussian Naive Bayes: Constructing the classification model with GNB using the training data.
2. Grid Search: Optimizing model parameters using grid search to find the best value for the parameter classifier__var_smoothing, which varies from 1×10−91×10−9 to 1×10−51×10−5.
3. 10-Fold Cross Validation: Performing 10-fold cross-validation to find the optimal parameters and avoid overfitting.

3.6. Model Evaluation

Model evaluation is conducted to assess the classification performance using test data. The evaluation methods include several key metrics:

1. Accuracy: Measures the percentage of correct predictions out of the total predictions.
2. Precision: Calculates the percentage of true positive predictions out of the total positive predictions.
3. Recall: Calculates the percentage of true positive predictions out of the total actual positive data.

A confusion matrix is also used in this evaluation. A confusion matrix is a table that allows for the visualization of model performance by comparing the actual values and predicted values of a classification, dividing them into four categories: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The diagram below illustrates the confusion matrix used to categorize classification results [19].

3.7. N-Gram Analysis

N-gram analysis is conducted to understand the patterns of word or phrase occurrences in user reviews. An N-gram is a method for identifying and counting the frequency of word or phrase occurrences in a text based on the number of words used as a single unit. The stages of N-gram analysis include [20]:

1. Unigram: Identifying and counting the frequency of single-word occurrences in the reviews.
2. Bigram: Identifying and counting the frequency of pairs of consecutive words in the reviews.
3. Trigram: Identifying and counting the frequency of sequences of three consecutive words in the reviews.

3.8. Interpretation of Results

The results of the above stages are interpreted to provide a comprehensive understanding of user sentiment towards the sister for Student application. These results include:

1. Model Performance: Analysis of the Gaussian Naive Bayes model's accuracy, precision, and recall measurements.
2. N-Gram Analysis Results: Identification of the most frequently occurring words or phrases in positive and negative reviews.
3. User Insights: Conclusions about user sentiment and areas for improvement in the application based on user reviews.

3. RESULTS AND DISCUSSION

3.1. Modeling

In this section, we briefly explain what a model is, the modeling process performed, and the scenarios implemented. A model in machine learning is an algorithmic representation that learns patterns from data to make predictions or decisions without being explicitly programmed to perform the task. This study used the Gaussian Naive Bayes (GNB) algorithm to classify the sentiment of Sister for Student application reviews. The modeling process involved several steps: data preprocessing, feature extraction using TF-IDF, and training the GNB model on the training dataset. Grid search and 10-fold cross-validation were employed to optimize the model parameters. The scenarios implemented included different values for the var_smoothing parameter, ranging from 1e-9 to 1e-5. The obtained modeling results are presented in the tabulated table below.

Table 1. Gaussian Naive Bayes Modeling Results

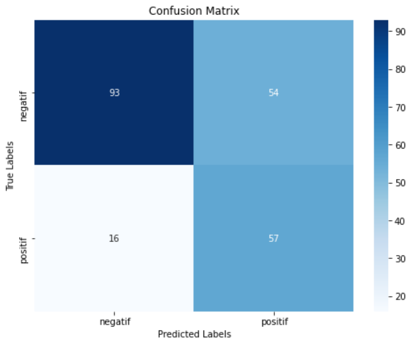
No.	Parameter	average accuracy	average precision	average recall
1	0.00001	0.723	0.746	0.723
2	0.000001	0.708	0.735	0.708
3	0.0000001	0.695	0.729	0.695
4	0.00000001	0.689	0.725	0.689
5	0.000000001	0.680	0.718	0.680

Based on varying var_smoothing parameters for the Gaussian Naive Bayes (GNB) algorithm, the modeling accuracy results indicate significant findings. The highest average accuracy of 0.723 and the highest average precision of 0.746 and recall of 0.723 was achieved with a var_smoothing value of 0.00001. This demonstrates that this parameter setting is the most effective among those tested for classifying user reviews accurately. As the var_smoothing value decreases from 0.00001 to 0.000000001, there is a notable decline in all three metrics: accuracy, precision, and recall. For instance, with a var_smoothing value of 0.000001, the average accuracy drops to 0.708, and further decreases are observed with lower var_smoothing values, culminating in an average accuracy of 0.680 at 0.000000001. This trend suggests that smaller values of var_smoothing are less effective, likely due to overfitting or insufficient smoothing of the data variance.

Despite the consistent decrease in performance metrics with lower var_smoothing values, the precision remains slightly higher than recall across all parameters. This indicates that the model is more conservative in predicting positive classes, resulting in fewer false positives but slightly more false negatives. Overall, the analysis underscores the critical importance of parameter tuning in Gaussian Naive Bayes modeling. The optimal var_smoothing value of 0.00001 should be preferred for future modeling tasks on similar datasets to ensure a balanced and reliable sentiment classification, providing the best accuracy, precision, and recall performance.

3.2. Model Evaluation

In this section, we present the best parameters obtained from the modeling process and their application to the test data. The optimal parameter found for the GNB model was the var_smoothing value, which was determined using grid search and 10-fold cross-validation. This optimized model was then applied to the test dataset to evaluate its performance. The evaluation metrics used included accuracy, precision, recall, and F1-score, all of which were derived from the confusion matrix. The results of the evaluation are summarized in the following classification report.



	precision	recall	f1-score	support
negatif	0.85	0.63	0.73	147
positif	0.51	0.78	0.62	73
accuracy			0.68	220
macro avg	0.68	0.71	0.67	220
weighted avg	0.74	0.68	0.69	220

Figure 3. (left) Confusion Matrix Test Results (right) Classification Report Evaluation Results of the GNB Model

The evaluation results of the Gaussian Naive Bayes model on the test dataset reveal several key insights regarding its performance in classifying user reviews into positive and negative sentiments. Firstly, the model's overall accuracy is 0.68, indicating that 68% of the reviews were correctly classified. This suggests a moderately effective performance with room for improvement. When examining each class's precision, recall, and F1-score metrics, a more detailed understanding of the model's strengths and weaknesses emerges. The model achieved a high precision of 0.85 for the negative sentiment class. This means that 85% of the reviews predicted as negative were indeed negative, indicating a solid ability to identify true negative reviews. However, the recall for the negative class is 0.63, suggesting that the model only captured 63% of all actual negative reviews, potentially missing a significant portion of negative sentiment.

Conversely, for the positive sentiment class, the precision is lower at 0.51, meaning just over half of the reviews predicted as positive were truly positive. Despite this, the recall for the positive class is relatively high at 0.78, indicating that the model successfully identified 78% of all positive reviews. This discrepancy between precision and recall for the positive class indicates the model tends to overpredict positive sentiment, resulting in a higher number of false positives. The macro average, the arithmetic means of precision and recall for both classes, shows values of 0.68 for precision and 0.71 for recall. This balanced performance across classes highlights that, while the model is not heavily biased towards one class, improvements can still be made, particularly in harmonizing precision and recall.

The weighted average, which considers the support (number of instances) for each class, presents a precision of 0.74, a recall of 0.68, and an F1-score of 0.69. These metrics reflect the model's overall performance more accurately, accounting for the dataset's higher prevalence of negative reviews. In summary, the Gaussian Naive Bayes model performs better in identifying negative reviews than positive ones, with a notable trade-off between precision and recall in both sentiment classes. The overall accuracy of 68% indicates moderate effectiveness, with precision and recall values suggesting areas for further refinement to enhance the model's predictive capabilities.

3.3. Sentiment Analysis Results

This section presents the results of the N-Gram implementation on sentiment analysis to identify the most frequently occurring words or phrases in positive and negative reviews using unigrams, bigrams, and trigrams. The analysis provides insights into users' language patterns and key expressions when expressing positive or negative sentiments about the Sister for Student application.

1. Unigram Analysis: Single words that frequently appear in positive and negative reviews.
2. Bigram Analysis: Pairs of consecutive words commonly appearing in reviews.
3. Trigram Analysis: Sets of three consecutive words frequently occurring in the reviews.

The 10 most frequently occurring words or phrases for each sentiment across unigram, bigram, and trigram analyses are presented in the following diagrams.

```
[(('bisa',), 72),
 (('bagus',), 65),
 (('aplikasi',), 49),
 (('sangat',), 45),
 (('fitur',), 39),
 (('good',), 36),
 (('mahasiswa',), 36),
 (('sudah',), 35),
 (('membantu',), 34),
 (('baik',), 31)]
```

```
[(('sangat', 'membantu'), 24),
 (('sangat', 'bagus'), 8),
 (('terima', 'kasih'), 7),
 (('sudah', 'bagus'), 6),
 (('upt', 'ti'), 6),
 (('sangat', 'bermanfaat'), 5),
 (('bisa', 'absen'), 5),
 (('mata', 'kuliah'), 5),
 (('membantu', 'mahasiswa'), 5),
 (('jadwal', 'kuliah'), 5)]
```

```
[(('sangat', 'membantu', 'dalam'), 4),
 (('sangat', 'membantu', 'mahasiswa'), 4),
 (('alhamdulillah', 'sangat', 'membantu'), 3),
 (('aplikasi', 'sangat', 'bagus'), 3),
 (('notifikasi', 'jadwal', 'perkuliahan'), 2),
 (('jadwal', 'mata', 'kuliah'), 2),
 (('sudah', 'cukup', 'bagus'), 2),
 (('aplikasi', 'sudah', 'bagus'), 2),
 (('semoga', 'semakin', 'baik'), 2),
 (('mahasiswa', 'universitas', 'jember'), 2)]
```

Figure 4. (left) Unigram Results (middle) Bigram Results (right) Trigram Results

The unigram analysis reveals that the most common single words in the reviews are “bisa” (72 occurrences), “bagus” (65 occurrences), and “aplikasi” (49 occurrences). Other frequently mentioned words include “sangat” (45 occurrences), “fitur” (39 occurrences), “good” (36 occurrences), “mahasiswa” (36 occurrences), “sudah” (35 occurrences), “membantu” (34 occurrences), and “baik” (31 occurrences). These words indicate that users frequently discuss the functionality and quality of the application, with “bagus,” “baik,” and “membantu” suggesting a positive sentiment towards the application's helpfulness and overall quality.

The bigram analysis shows the prevalence of common two-word phrases. The most frequent bigrams are “sangat membantu” (24 occurrences), “sangat bagus” (8 occurrences), and “terima kasih” (7 occurrences), indicating expressions of gratitude and positive feedback. Other notable bigrams include “sudah bagus” (6 occurrences), “upt ti” (6 occurrences), “sangat bermanfaat” (5 occurrences), “bisa absen” (5 occurrences), “mata kuliah” (5 occurrences), “membantu mahasiswa” (5 occurrences), and “jadwal kuliah” (5 occurrences). These bigrams suggest that users appreciate the app's assistance with attendance, course scheduling, and its benefits to students.

The trigram analysis identifies frequent three-word phrases, which often provide more context. The most common trigrams are “sangat membantu dalam” (4 occurrences) and “sangat membantu mahasiswa” (4 occurrences), highlighting the application's significant role in assisting students. Other trigrams such as “alhamdulillah sangat membantu” (3 occurrences), “aplikasi sangat bagus” (3 occurrences), and “notifikasi jadwal perkuliahan” (2 occurrences) further emphasize the application's positive impact and its utility in academic scheduling. Additional trigrams like “jadwal mata kuliah” (2 occurrences), “sudah cukup bagus” (2 occurrences), “aplikasi sudah bagus” (2 occurrences), “semoga semakin baik” (2 occurrences), and “mahasiswa Universitas Jember” (2 occurrences) reflect users' satisfaction and their hopes for continued improvement.

4. CONCLUSION

The GNB model effectively classified user sentiment in reviews for the “Sister for Student” application, with grid search optimization identifying 0.00001 as the best smoothing parameter. This optimization led to a mean accuracy of 72.3%, a mean precision of 74.6%, and a mean recall of 72.3%, indicating that the GNB model has a strong capacity for text classification tasks. Testing withheld-out data produced an overall accuracy of 68%, with a precision of 0.85 and recall of 0.63 for negative sentiment. In contrast, positive sentiment achieved a lower precision of 0.51 and a higher recall of 0.78. This suggests the model is particularly adept at identifying negative sentiments but struggles more with accurately identifying positive instances. The confusion matrix confirms a higher true positive rate for negative reviews than positive ones, hinting at the GNB model’s limitation in its independence assumption, which may not hold in natural language and could lead to misclassifications. Improving precision for positive sentiment classification could be achieved by additional tuning or incorporating features. An N-gram analysis provided insights into user feedback, with unigrams like "bisa," "bagus," "aplikasi," and "membantu" highlighting user focus on functionality and quality. Bigrams like “sangat membantu” and trigrams like “sangat membantu mahasiswa” emphasize the app's supportive role, reflecting positive user appreciation.

Based on the N-Gram results, several recommendations can be made. Firstly, developers should continue enhancing the application's helpful features, as users highly appreciate these. Secondly, addressing issues related to app crashes or bugs could improve user satisfaction, as indicated by some negative sentiments. Lastly, expanding features that assist with academic tasks and integrating user feedback into updates can help maintain and increase positive sentiments. Focusing on the key aspects highlighted in user reviews will improve the “Sister for Student” application and better meet user needs.

REFERENCES

[1] S. Criollo-C, A. Guerrero-Arias, Á. Jaramillo-Alcázar, and S. Luján-Mora, “Mobile Learning Technologies for Education: Benefits and Pending Issues,” *Applied Sciences 2021, Vol. 11, Page 4111*, vol. 11, no. 9, p. 4111, Apr. 2021, doi: 10.3390/APP11094111.

[2] M. I. N. Ardiansyah, “Analisa Faktor-faktor yang Mempengaruhi Kepuasan Penggunan Aplikasi Sister For Student Menggunakan Metode End User Computing Satisfaction,” *Jurnal Sistem Informasi*, p. 95, 2019.

[3] B. Fu, J. Lin, L. Liy, C. Faloutsos, J. Hong, and N. Sadeh, “Why people hate your App - Making sense of user feedback in a mobile app store,” *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. Part F128815, pp. 1276–1284, Aug. 2013, doi: 10.1145/2487575.2488202.

[4] E. Indrayuni, “Analisa Sentimen Review Hotel Menggunakan Algoritma Support Vector Machine Berbasis Particle Swarm Optimization,” *Jurnal Evolusi Volume 4 Nomor 2 - 2016*, vol. 4, no. 2, pp. 20–27, 2016.

[5] N. F. Najwa, M. A. Furqon, E. S. Sintiya, and A. C. Puspitaningrum, “AKUISISI DATA MEDIA SOSIAL PEMERINTAH UNTUK MENGANALISIS KETERBUKAAN INFORMASI PENYEBARAN COVID-19”.

[6] M. A. Furqon, D. Hermansyah, R. Sari, A. Sukma, Y. Akbar, and N. A. Rakhmawati, “Analisis sosial media pemerintah daerah di indonesia berdasarkan respons warganet,” *Jurnal Sosioteknologi*, vol. 17, no. 2, pp. 2–4, 2018.

[7] P. Borele and D. A. Borikar, “An Approach to Sentiment Analysis using Artificial Neural Network with Comparative Analysis of Different Techniques,” *IOSR Journal of Computer Engineering (IOSR-JCE)*, vol. 18, no. 2, pp. 64–69, 2016.

[8] N. G. Ramadhan and F. D. Adhinata, “Sentiment analysis on vaccine COVID-19 using word count and Gaussian Naïve Bayes,” *Indonesian Journal of Electrical Engineering and Computer Science*, 2022.

[9] N. G. Ramadhan and F. D. Adhinata, “Sentiment analysis on vaccine COVID-19 using word count and Gaussian Na\"\ive Bayes,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 26, no. 3, pp. 1765–1772, 2022.

[10] H. Faisol, K. Djajadinata, and M. Muljono, “Sentiment analysis of yelp review,” *Proceedings - 2020 International Seminar on Application for Technology of Information and Communication: IT Challenges for Sustainability, Scalability, and Security in the Age of Digital Disruption, iSemantic 2020*, pp. 179–184, 2020, doi: 10.1109/iSemantic50169.2020.9234213.

[11] S. U. Hassan, J. Ahamed, and K. Ahmad, “Analytics of machine learning-based algorithms for text classification,” *Sustainable Operations and Computers*, vol. 3, pp. 238–248, 2022, doi: 10.1016/j.susoc.2022.03.001.

[12] S. M. Malakouti, M. B. Menhaj, and A. A. Suratgar, “The usage of 10-fold cross-validation and grid search to enhance ML methods performance in solar farm power generation prediction,” *Clean Eng Technol*, vol. 15, p. 100664, Aug. 2023, doi: 10.1016/J.CLET.2023.100664.

[13] S. U. Hassan, J. Ahamed, and K. Ahmad, “Analytics of machine learning-based algorithms for text classification,” *Sustainable Operations and Computers*, vol. 3, pp. 238–248, Jan. 2022, doi: 10.1016/J.SUSOC.2022.03.001.

[14] A. I. Kadhim, “Term Weighting for Feature Extraction on Twitter: A Comparison between BM25 and TF-IDF,” *2019 International Conference on Advanced Science and Engineering, ICOASE 2019*, pp. 124–128, Apr. 2019, doi: 10.1109/ICOASE.2019.8723825.

[15] A. Fakhruddin, J. S. Kom, A. Ridwan, and S. Mmsi, “Performance Measurement of Confusion Matrix Accuracy in Sentiment Analysis with Decision Trees, Naïve Bayes, K-Nearest Neighbor methods Using Rapidminer,” *International Research Journal of Advanced Engineering and Science*, vol. 8, no. 4, pp. 123–127, 2023.

[16] A. Tripathy, A. Agrawal, and S. K. Rath, “Classification of sentiment reviews using n-gram machine learning approach,” *Expert Syst Appl*, vol. 57, pp. 117–126, Sep. 2016, doi: 10.1016/J.ESWA.2016.03.028.

[17] C. Martins, “Gaussian Naive Bayes Explained and Hands-On with Scikit-Learn | by Carla Martins | Towards AI,” 2022.

[18] A. Ashari Muin and Syarli, “Metode Naive Bayes Untuk Prediksi Kelulusan (Studi Kasus: Data Mahasiswa Baru Perguruan Tinggi),” *Jurnal Ilmiah Ilmu Komputer*, vol. 2, no. 1, 2016.

[19] N. Sarang, “Understanding Confusion Matrix,” *Towards Data Science*, 2018.

[20] P. Kumar, “An Introduction to N-grams: What Are They and Why Do We Need Them?,” *XRDS Crossroads - The ACM Magazine for Students*, pp. 1–6, 2017.