Jurnal Sisfotek Global

ISSN (Online): 2721 - 3161, ISSN (Print): 2088 – 1762
DOI: http://dx.doi.org/10.38101/sisfotek.v15i1.15788
Vol. 15, No. 1, March 2025, pp. 20-27

Sentiment Analysis on User Reviews of the Edlink Application Using the Random Forest Classifier Method

Sebastianus A.S. Mola[1], Dian Putri Novita Polly[2] & Nelcy D. Rumlaklak[3]

[1,2,3]Nusa Cendana University, Kupang, Indonesia, 85001
E-mail: [1]adimola@staf.undana.ac.id, [2]dianpolly2@gmail.com, [3]dessyrumlaklak@staf.undana.ac.id

ARTICLE HISTORY

Received : January 14th, 2025
Revised : February 15th, 2025
Accepted : March 25th, 2025

KEYWORDS

Edlink Application
Sentiment Analysis
Random Forest Classifier
SMOTE
K-Fold Cross Validation

ABSTRACT

Edlink is a learning platform developed by PT. Sentra Vidya Utama (SEVIMA), established in 2004. Although it offers useful features, some aspects need improvement based on user reviews on Google Play Store. This study aims to accurately classify user sentiment to identify areas that need enhancement. The main challenges include language diversity, sentiment class imbalance, and the need for a reliable classification method. The random forest classifier method was chosen for its ability to handle overfitting and optimize performance. The dataset consists of 1,117 reviews divided into three classes: 385 negative, 118 neutral, and 614 positive. Data was collected through web scraping and processed using cleaning, normalization, tokenizing, stemming, negation conversion, and stopword removal, then weighted using TF-IDF. Testing results showed an accuracy of 86% using 5-Fold cross-validation and SMOTE. The 10-Fold cross-validation test demonstrated that this method outperforms other classification methods with 90% accuracy.

1. Introduction

Learning Management System (LMS) is software designed to create, distribute, and manage the delivery of learning materials online. LMS applications play a crucial role in supporting online learning. Many higher education institutions in Indonesia have adopted various LMS applications, such as Edlink, Google Classroom, Moodle, Schoology, Atutor, and others.

Edlink is an LMS application developed by PT. Sentra Vidya Utama (SEVIMA). Unlike most LMS applications originating from abroad, Edlink is a local creation tailored to Indonesia's learning culture. Edlink’s development stemmed from an initiative to provide broader access to information among higher education institutions. In 2020, Edlink was introduced as an LMS to assist universities during the COVID-19 pandemic. Edlink enables lecturers and students to interact via teleconferencing features integrated with Zoom and Google Meet and connects to academic systems for efficient class and course management. Data shows that as of 2023, Edlink has served over 700 higher education institutions, with a total of 3 million users, including students, lecturers, and campus operators [1].

Despite its usefulness, Edlink has certain shortcomings. One frequently reported issue by users is delayed notifications, with a complaint rate of 75% [2]. User experiences with Edlink are often expressed in the review section of the Google Play Store, encompassing both criticisms and expressions of satisfaction. Given the varying opinions and perspectives, sentiment analysis is needed to analyze user feedback on Edlink’s services. Sentiment analysis provides insights into user experiences and helps companies manage user perceptions, which can aid in improving product or service quality.

Sentiment analysis has become a extensively studied topic as it seeks to extract insights from unstructured data [3]. It involves removing irrelevant words and symbols from data and converting qualitative data into quantitative forms. User reviews are then classified to identify whether they are positive or negative [4].

Numerous studies on sentiment analysis have been conducted. Research [5] analyzed the sentiment of the Dana application reviews using the random forest method, achieving an accuracy of 84% with a tree depth of 65 and 40 trees. Another study on public sentiment regarding COVID-19 vaccination on Twitter used the random forest classifier method on 1,500 tweets. Results showed that public sentiment toward Sinovac vaccination was positive, with the model predicting tweet sentiment with 79% accuracy [6]. Furthermore, study [2] applied SVM for sentiment analysis on Edlink using confusion matrix testing, achieving an accuracy rate of 82%.

Previous studies still have several limitations. Research on sentiment analysis of the LMS Edlink application using the random forest method is still very limited. Most studies focus more on sentiment analysis of commercial applications or social media. Moreover, previous research using the SVM method on Edlink has not explored hyperparameter optimization to improve model accuracy. Therefore, this study aims to fill this gap by applying the random forest method in sentiment analysis of user reviews of the Edlink application and performing hyperparameter optimization to enhance model performance.

The advantage of this study compared to previous research lies in the use of random forest, which has the ability to combine predictions from multiple decision trees into a single model, making it more stable and less prone to overfitting. Additionally, this study will implement hyperparameter optimization techniques, such as selecting the optimal number of trees and the appropriate tree depth, to improve model accuracy [7]. Thus, the results of this research are expected to provide deeper insights into user perceptions of Edlink and offer more accurate recommendations for developers to enhance the quality of the application.

2. Method

This study employs a methodological approach to analyze sentiments in user reviews of the Edlink application. The research stages include data collection, followed by text preprocessing, TF-IDF weighting, K-fold cross-validation, random forest classification, and accuracy testing. The details of the research stages are illustrated in Figure 1.

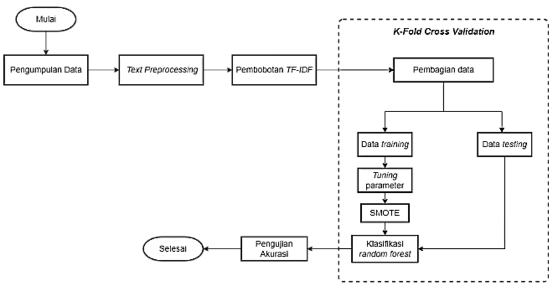

Figure 1. Research Stages

The research process in Figure 1 begins with data collection, followed by text preprocessing to clean and prepare the data for analysis. Next, the term frequency-inverse document frequency (TF-IDF) weighting is applied to transform textual data into numerical form. The research then implements K-Fold Cross-Validation, where the dataset is split into training and testing sets. The training data undergoes parameter tuning to optimize model performance, followed by Synthetic Minority Over-sampling Technique (SMOTE) to handle class imbalances. The processed data is then classified using the Random Forest algorithm. Finally, accuracy testing is conducted to evaluate the model's performance, marking the end of the research process.

2.1 Data Collection

The dataset used in this study is sourced from user reviews of the Edlink application on the Google Play Store, a trusted platform for providing direct user feedback. The data collection process was carried out using a scraping technique with Google Colab to ensure consistency in data retrieval. The total dataset obtained through scraping consists of 1,117 reviews spanning from February 2017 to December 2023. For data validity and reliability, the labeling process was conducted manually by linguistic experts, considering applicable language rules. Additionally, to maintain consistency in classification results, validation based on a machine learning model trained with labeled data was used to disseminate the manual labeling results, thereby minimizing subjectivity in assigning positive, negative, and neutral labels. The collected and labeled dataset is illustrated in Figure 2.

	username	at	label	ulasan
0	Intan Safitri	12/27/2023 9:41	Negatif	Tolong kepada pihak apk sevima, akun saya saat...
1	Ibnu Najaib	12/25/2023 23:50	Negatif	aneh kenapa tiba tiba keluar sendiri dari akun...
2	perdy imam	12/22/2023 10:58	Negatif	Suruh kasih izin mulu, padahal udah dikasih iz...
3	Erni Kedafota	12/15/2023 3:13	Netral	Aplikasinya bagus tapi tiba-tiba tidak bisa me...
4	Andi Nurfadillah	12/14/2023 8:15	Negatif	Kenapa kodenya tidak muncul" di Gmail yang di ...
...	...	...	...	...
1112	Pengguna Google	9/11/2017 9:30	Netral	Oke
1113	Pengguna Google	8/18/2017 11:11	Netral	Apakah untuk gofedder nya sudah berjalan denga...
1114	Pengguna Google	5/3/2017 5:29	Netral	Apa ini hanya untuk dosen saja?bagaimana untuk...
1115	Pengguna Google	3/30/2017 1:11	Netral	Coba dulu
1116	Pengguna Google	2/23/2017 3:37	Positif	Sangat bermanfaat untuk menunjang proses perku...

1117 rows × 4 columns

Figure 2. Research Dataset

2.2 *Text Preprocessing*

Text preprocessing is the initial stage in text analysis that includes semantic and syntactic analysis to prepare the text into high-quality data ready for further processing [8]. The stages in text preprocessing include:

1) Cleaning: Performed to reduce noise in the data or comments.
2) Case Folding: Adjusts the text to a uniform format, specifically lowercase.
3) Normalization: Converts abbreviations or slang words into their standard forms.
4) Tokenizing: Breaks the dataset into tokens or smaller word segments.
5) Stemming: Reduces words to their root or base form.
6) Convert Negation: Adds a dependency marker "_neg" to words following those with negative connotations.
7) Stopword Removal: Removes words considered less relevant for analysis.

2.3 Weighting TF-IDF

The Term Frequency-Inverse Document Frequency (TF-IDF) method is a technique used to measure the importance of a word (term) in a document by

assigning a weight to each word [9]. The TF-IDF method is a combination of two weighting concepts, namely TF (Term Frequency) and IDF (Inverse Document Frequency). TF is the frequency of a term (word/phrase) appearing in the relevant review. The TF formula can be seen
in Equation 1.

$$TF_{(t_i u_j)} = \frac{f(t_i u_j)}{n} \quad (1)$$

Meanwhile, IDF refers to the document that contains the term. The IDF formula can be seen in Equation 2.

$$IDF_{(t_i)} = ln\left(\frac{m+1}{1+df_{(t_i)}}\right) + 1 \quad (2)$$

Description:
t = Term
i = Term index (1,2,3,…, n)
u = Review
j = Review index (1,2,3,…., m)
m = Total of the review
n = Total of the term in u_j
$f(t_i, u_j)$ = Term frequency of-i in the review of-j
$df_{(t_i)}$ = Term review frequency of-i

The calculation of TF-IDF weights in this study uses the TfidfVectorizer() class from the scikit-learn library. Subsequently, the TF-IDF weight values will be normalized using the Euclidean Norm [10]. The calculation of TF-IDF weights and the normalization of TF-IDF weights can be seen in Equation 3 and Equation 4.

$$v_{(t,u)} = TF_{(t,u)} * IDF_{(t)} \quad (3)$$

$$v_{norm_{((t,u)}} = \frac{v_{(t,u)}}{\sqrt{v^2_{(1,u)} + v^2_{(2,u)} \dots + v^2_{(n,u)}}} \quad (4)$$

Description:
$v_{(t,u)}$ = TF-IDF weight value
$v_{norm_{((t,u)}}$ = Value of each normalized term

2.4 Syintetic Minority Oversampling Technique (SMOTE)

In research with a large amount of data, class imbalance issues often arise due to data that is inadequate or poorly structured. Therefore, a method is needed to address this issue. SMOTE is a widely used technique to address class imbalance by generating new samples from the minority class, thereby helping to balance the data distribution through the resampling process [11].

2.5 Random Forest

Random forest is a classification method that constructs multiple decision trees using random samples and training data variables, with the aim of improving model accuracy [12]. The following are the steps for classification using the random forest method [13]:

1) Creating a bootstrap sample involves randomly drawing a sample of size N observations from the training data with replacement.
2) The next step is random feature selection where in this step the tree is built so that it reaches the maximum size (without pruning).
3) Repeat the step 1-3 as much as k times, so that it forms a forest that includes k tree
4) Majority vote to classification

In the random forest method, there are parameters that must be adjusted to achieve an optimal model, referred to as hyperparameters. One method to determine effective hyperparameters is by using the grid search method. Grid search is an approach used to find the best combination of hyperparameters in a model, so the model can produce accurate predictions on the data used. There are many hyperparameters in the random forest algorithm that can be tuned. In this study, the focus is on several hyperparameters, including "n_estimators" (the number of trees in the forest), "max_depth" (the maximum depth of the trees), "min_samples_split" (the minimum number of samples required to split a node), "min_samples_leaf" (the minimum number of samples required to be at a leaf node), and "max_features" (the number of features considered when searching for the best split).

2.6 K-Fold Cross Validation

K-fold cross-validation begins by dividing the data into a specified number of k-folds as needed. This method splits the data into k equally sized parts, such as Fold1, Fold2, Fold3, ..., Foldn. Subsequently, the training and testing process is performed k times, where each iteration uses a different part as the test data, while the remaining parts are used as training data [14].

2.7 Confusion Matrix

The confusion matrix is a tool for assessing the performance of a classification algorithm. It provides various measurable metrics, including accuracy, precision, recall, specificity, and F1-Score, to determine the effectiveness of the model [15]. During the evaluation phase, the confusion matrix is illustrated in Table 1, which consists of the number of test data points correctly and incorrectly predicted or classified by the classification model.

Table 1. Confusion Matrix

		Predicted Class		
		Negative	**Neutral**	**Positive**
Actual Class	**Negative**	TNegNeg	NegFNet	NegFP
	Neutral	NetFNeg	TNetNet	NetFP

	Posit	PFNeg	PFNet	TPP

Explanation:

TNegNeg (True Negative Negative): The number of data points where the actual value is negative and the predicted value is also negative.

TNetNet (True Neutral Neutral): The number of data points where the actual value is neutral and the predicted value is also neutral.

TPP (True Positive Positive): The number of data points where the actual value is positive and the predicted value is also positive.

PFNeg (Positive False Negative): The number of data points where the actual value is positive but the predicted value is negative.

PFNet (Positive False Neutral): The number of data points where the actual value is positive but the predicted value is neutral.

NegFP (Negative False Positive): The number of data points where the actual value is negative but the predicted value is positive.

NegFNet (Negative False Neutral): The number of data points where the actual value is negative but the predicted value is neutral.

NetFP (Neutral False Positive): The number of data points where the actual value is neutral but the predicted value is positive.

NetFNeg (Neutral False Negative): The number of data points where the actual value is neutral but the predicted value is negative.

The confusion matrix also serves as the basis for calculating accuracy, precision, and recall. Accuracy indicates the proportion of correct predictions compared to the actual conditions. Precision measures the accuracy or exactness of the test results. Recall measures the proportion of correctly identified values [16]. The calculations for accuracy, precision, and recall are based on the following equations:

$$Accuracy = \frac{TNegNeg + TNetNet + TPP}{Total\ prediksi} x100\% \quad (5)$$

$$Precision = \frac{TPP}{TPP + NegFP + NetFP} x100\% \quad (6)$$

$$Recall = \frac{TPP}{TPP + PFNeg + PFNet} x100\% \quad (7)$$

3. Result and Discussion

The research results include a description of each stage or process carried out in this study.

3.1Text Preprocessing

The review data obtained from the Google Play Store application consists of textual data containing unstructured characters or words. Therefore, in this process, these characters and words will be removed or modified to make classification easier. The initial stage in text preprocessing for this study is cleaning. The cleaning stage is performed to remove punctuation marks, special characters, numbers, URLs, emoticons, hashtags (#), mentions (@), and links from the review texts. The results of the cleaning process can be seen in Table 2.

Table 2. Cleaning Results

Input	*Output*
This app is good	This app is good
Does not support study packages	Does not support study packages

The case folding stage converts all letters in the text into a uniform form, which is lowercase letters. The results of case folding can be seen in Table 3.

Table 3. Case Folding Results

Input	*Output*
This app is good	this app is good
Does not support study packages	does not support study packages

The data that has undergone the case folding stage is then normalized by correcting non-standard words into standard words. The results of the normalization can be seen in Table 4.

Table 4. Normalization Results

Input	*Output*
this app is good	**application** is good
does not support study packages	does not support study packages

Next, in the tokenizing stage, the text is split based on spaces. The results of tokenizing are shown in Table 5.

Table 5. Tokenizing Results

Input	*Output*
application is good	[application, is, good]
does not support study packages	[does, not, support, study, packages]

The stemming stage is performed to remove all affixes. The results of stemming can be seen in Table 6.

Table 6. Stemming Results

Input	*Output*
[application, is, good]	application is good
[does, not, support, study, packages]	not **support** study packages

In the convert negation stage, a dependency marker "_neg" is added to words with negative connotations. The results of convert negation can be seen in Table 7.

Table 7. Convert Negation Results

Input	*Output*
application is good	application is good
not support study packages	**not support_neg** study packages

The final stage of text preprocessing in this study is stopword removal, which is performed to eliminate words that are considered less relevant. The results of stopword removal can be seen in Table 8.

Table 8. Stopword Removal Results

Input	*Output*
application is good	application good
not support_neg study packages	**support_neg** study packages

3.2 TF – IDF Weighting

After preprocessing, TF-IDF weighting is applied to assign values to each word in a review. The results of TF-IDF weighting are shown in Figure 3.

```
aplikasi: 0.6139675966047301
bagus: 0.789331229788491

ajar: 0.5773502691896257
dukung_neg: 0.5773502691896257
paket: 0.5773502691896257
```

Figure 3. TF-IDF Results

3.3 Random Forest Classification Results

The best parameter values from the tuning of the random forest parameters using GridSearchCV for classification are shown in Table 9.

Table 9. GridSearchCV Parameter Tuning Results

Parameter	*Grid Search Values*	*Best* Parameter
n_estimators	100, 200, 300, 400, 500	200
max_depth	50, 60, 70, 80, 90, 100	100
criterion	*entropy*	*entropy*
min_samples_split	2, 5, 10	5
min_samples_leaf	1, 2, 3, 4, 5	1
max_features	*sqrt, log2*	*sqrt*

The testing scenario is presented in two ways: testing without the SMOTE method and testing with the SMOTE method.

a. Results of testing without SMOTE

The total number of dataset entries obtained from scraping and that have gone through the text preprocessing stages is 1,117 reviews. The data is then divided into 5 Folds, where in each iteration, 1 Fold is selected as the training data, while the remaining 4 Folds are used as testing data. This process continues until each Fold has been used once as the testing data. The results of the dataset division into 5 Folds can be seen in Table 10.

Table 10. Dataset Division Results

Fold$_i$	*Training*				*Testing*			
	Neg	Net	Pos	Total	Neg	Net	Pos	Total
Fold$_1$	307	92	494	893	78	26	120	224
Fold$_2$	314	95	484	893	71	23	130	224
Fold$_3$	306	93	495	894	79	25	119	223
Fold$_4$	303	95	496	894	82	23	118	223
Fold$_5$	310	97	487	894	75	21	127	223

After the dataset division, testing is performed, and the results of the confusion matrix are obtained, showing the highest accuracy in fold K4, as shown in Figure 4.

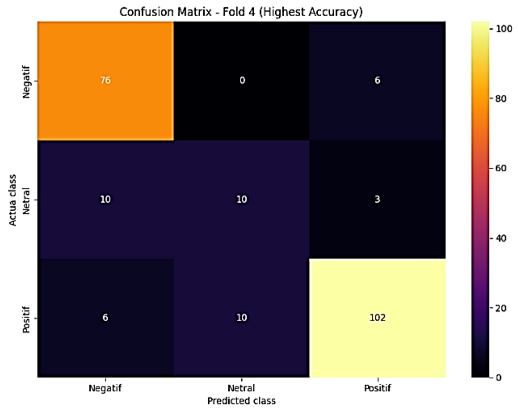

Figure 4. Confusion Matrix Results for Fold K4

From the confusion matrix results, the accuracy, precision, and recall are obtained and can be seen in Table 11.

Table 11. Accuracy, Precision, and Recall Results

Fold	Precision			*Recall*			Accuracy
	Neg	Net	Pos	Neg	Net	Pos	
K1	84%	52%	89%	86%	46%	90%	83%
K2	89%	60%	84%	79%	52%	91%	83%
K3	82%	100%	84%	91%	12%	93%	83%
K4	83%	50%	92%	93%	43%	86%	84%
K5	75%	47%	95%	93%	43%	83%	83%
Rata-rata							83%

From the results, Fold K4 shows excellent performance with the highest accuracy of 84% in classifying negative and positive data, along with high precision and recall values. However, the neutral class remains a challenge, as evidenced by the lower precision and recall values. Thus, although the model achieves high accuracy, its performance in classifying reviews into each label class varies significantly. In fact, its performance is quite low for the neutral label class.

b. Results of testing with SMOTE

Based on the issue of imbalanced distribution of reviews in each label class, the SMOTE method is used to increase the number of data in the minority class and address the imbalance in the dataset. SMOTE is applied to the training data, and the results of the dataset division can be seen in Table 12.

Table 12. Results of Dataset Division After SMOTE

Fold$_i$	*Training*	*Testing*

	Neg	Net		Neg	Net		Neg	Net
$Fold_1$	494	494	***$Fold_1$***	494	494	***$Fold_1$***	494	494
$Fold_2$	484	484	***$Fold_2$***	484	484	***$Fold_2$***	484	484
$Fold_3$	495	495	***$Fold_3$***	495	495	***$Fold_3$***	495	495
$Fold_4$	496	496	***$Fold_4$***	496	496	***$Fold_4$***	496	496
$Fold_5$	487	487	***$Fold_5$***	487	487	***$Fold_5$***	487	487

After the dataset division, testing is performed, and the results of the confusion matrix are obtained, showing the highest accuracy in fold K3, as shown in Figure 5.

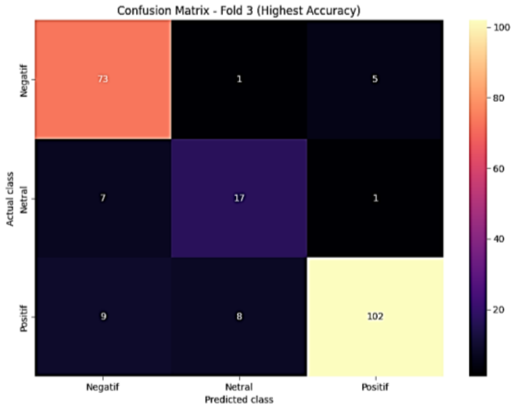

Figure 5. Confusion Matrix Results for Fold K3

From the confusion matrix results, the accuracy, precision, and recall are obtained and can be seen in Table 13.

Table 13. Accuracy, Precision, and Recall Results with SMOTE

Fold	Precision			*Recall*			Accuracy
	Neg	Net	Pos	Neg	Net	Pos	
K1	81%	43%	95%	91%	46%	86%	83%
K2	88%	59%	87%	80%	61%	91%	84%
K3	82%	65%	94%	92%	68%	86%	86%
K4	81%	48%	96%	96%	48%	84%	85%
K5	74%	45%	95%	93%	43%	81%	82%
Rata-rata							84%

In Table 13, it can be seen that the average accuracy of the model increased by 1% after applying SMOTE, resulting in an accuracy of 84%. Fold K3 achieved the highest accuracy of 86%. From these results, Fold K3 demonstrates excellent performance across all metrics, with high precision and recall values for all categories (Negative, Neutral, Positive). Therefore, SMOTE has contributed to improving the overall performance of the model, particularly in addressing class imbalance and enhancing precision and recall for certain classes that initially had lower performance.

The correlation matrix of precision, recall, and accuracy can be seen in the Figure 6.

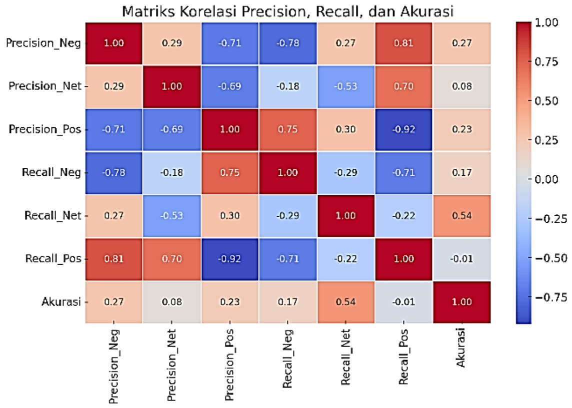

Figure 6. Correlation matrix of precision, recall, and accuracy.

The correlation matrix between precision, recall, and accuracy shows the relationships between evaluation metrics. Some key points from the correlation results are:

Recall_Pos and Precision_Pos have a strong negative correlation (-0.92), indicating that an increase in one may lead to a decrease in the other.

Precision_Neg and Recall_Pos have a strong positive correlation (0.81), meaning that if precision for the negative class increases, recall for the positive class also tends to increase.

Accuracy has a relatively low correlation with precision and recall, suggesting that an increase in accuracy does not necessarily mean all evaluation metrics improve proportionally.

The ANOVA test results show that the p-values for all metrics are greater than 0.05, indicating that there is no significant difference between the evaluation metrics before and after using SMOTE. The ANOVA test results can be seen in Table 14.

Table 14. ANOVA test results

Matrix	F-Statistic	p-value
Precision (Neg)	0.1960	0.6697
Precision (Net)	0.8422	0.3856
Precision (Pos)	2.8595	0.1293
Recall (Neg)	0.0862	0.7765
Recall (Net)	2.7192	0.1378
Recall (Pos)	1.5203	0.2526
Accuracy	1.1852	0.3080

Based on the ANOVA test results table, since the p-value > 0.05, there is not enough evidence to state that SMOTE produces a significant difference in the evaluation metrics.

In addition, testing was also conducted on the same dataset using 10-fold cross-validation with several other classification methods, namely Naïve Bayes Classifier (NBC), K-Nearest Neighbor (KNN), and Xtreme Gradient Boosting (XGB), with the results presented in Table 15.

Table 15. Comparison of Classification Algorithm Performance

Metric	Class	NBC	KNN	XGB	RFC
Precision	Negative	88%	78%	82%	88%
	Neutral	38%	25%	66%	66%
	Positive	94%	100%	98%	95%
Recall	Negative	83%	68%	94%	97%
	Neutral	50%	100%	76%	60%
	Positive	94%	56%	87%	88%
Accuracy		85%	65%	88%	**90%**

Based on the table above, the Naïve Bayes Classifier (NBC) demonstrates relatively stable performance with an accuracy of 85%. Precision and recall for the positive class are very high (94%), but for the neutral class, both precision and recall remain low (38% and 50%). K-Nearest Neighbor (KNN) performs worse than the other methods, achieving only 65% accuracy. While precision and recall for the negative and positive classes are fairly good, performance for the neutral class is significantly poor (25% precision, 10% recall). On the other hand, Xtreme Gradient Boosting (XGB) exhibits strong performance with an accuracy of 88%. Although precision and recall for the positive class are lower than those of NBC, they remain relatively good.

However, RFC achieves the highest accuracy (90%), with more balanced precision and recall across all classes, particularly for the neutral class, where it outperforms other methods. These findings highlight the importance of selecting the right algorithm to enhance sentiment analysis accuracy, especially when dealing with datasets that have an imbalanced class distribution.

4. Conclusion and Suggestions

Based on accuracy testing using the confusion matrix, 5-fold cross-validation and 10-fold cross-validation for performance comparison with several other classification methods, the implemented Random Forest classification model has shown good performance in categorizing reviews appropriately. The use of the SMOTE method has proven effective in addressing dataset imbalance issues, thereby improving model performance, with the highest accuracy reaching 90%, while precision and recall vary for each label class. However, model performance variations are still influenced by suboptimal text preprocessing steps, particularly in aspects such as cleaning, normalization, stemming, negation conversion, and stopword removal, which can impact classification accuracy. Overall, positive sentiment reviews dominate based on evaluation metrics, followed by negative sentiment, while neutral sentiment has the lowest proportion, although it increased after SMOTE was applied.

The implications of this study indicate that the application of machine learning models in sentiment analysis of app reviews can provide deeper insights into user satisfaction, which can be utilized by app developers to enhance service quality. For future research, it is recommended to improve text preprocessing steps to optimize data quality before entering the model training phase. Additionally, the analysis can be expanded by considering other factors such as user ratings or the temporal aspects of reviews to achieve more accurate results. Model development can also be directed toward deep learning approaches to further enhance classification performance. Finally, implementing a classification system capable of handling new, unlabeled data would be a valuable next step for real-world applications.

References

[1] Erna, “Apa itu SEVIMA? Profil dan Sejarah Edtech yang Sudah 2 Dekade Merevolusi Pendidikan Tinggi Indonesia | SEVIMA.” Accessed: Nov. 21, 2023. [Online]. Available: https://sevima.com/apa-itu-sevima-profil-dan-sejarah-edtech-yang-sudah-2-dekade-merevolusi-pendidikan-tinggi-indonesia/

[2] G. Rininda, I. H. Santi, and S. Kirom, “PENERAPAN SVM DALAM ANALISIS SENTIMEN PADA EDLINK MENGGUNAKAN PENGUJIAN CONFUSION MATRIX,” *JATI J. Mhs. Tek. Inform.*, vol. 7, no. 5, pp. 3335–3342, 2023.

[3] S. Kumar, P. P. Roy, D. P. Dogra, and B.-G. Kim, “A Comprehensive Review on Sentiment Analysis: Tasks, Approaches and Applications,” Nov. 19, 2023, *arXiv*: arXiv:2311.11250. doi: 10.48550/arXiv.2311.11250.

[4] R. Maulana, A. Voutama, and T. Ridwan, “Analisis Sentimen Ulasan Aplikasi MyPertamina pada Google Play Store menggunakan Algoritma NBC,” *J. Teknol. Terpadu*, vol. 9, no. 1, pp. 42–48, 2023.

[5] F. A. Larasati, D. E. Ratnawati, and B. T. Hanggara, “Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest,” 2022.

[6] M. Y. Aldean, P. Paradise, and N. A. Setya Nugraha, “Analisis Sentimen Masyarakat Terhadap Vaksinasi Covid-19 di Twitter Menggunakan Metode Random Forest Classifier (Studi Kasus: Vaksin Sinovac),” *J. Inform. Inf. Syst. Softw. Eng. Appl. INISTA*, vol. 4, no. 2, Art. no. 2, Jun. 2022, doi: 10.20895/inista.v4i2.575.

[7] H. Yuliana *et al.*, “Hyperparameter Optimization of Random Forest Algorithm to Enhance Performance Metric Evaluation of 5G Coverage Prediction,” *Bul. Pos Dan Telekomun.*, vol. 22, no. 1, Art. no. 1, Jun. 2024, doi: 10.17933/bpostel.v22i1.390.

[8] B. Z. Ramadhan, R. I. Adam, and I. Maulana, “Analisis Sentimen Ulasan pada Aplikasi E-Commerce dengan Menggunakan Algoritma Naïve Bayes,” *J. Appl. Inform. Comput.*, vol. 6, no. 2, Art. no. 2, Dec. 2022, doi: 10.30871/jaic.v6i2.4725.

[9] A. Widianto, E. Pebriyanto, F. Fitriyanti, and M. Marna, “Document Similarity using Term Frequency-Inverse Document Frequency Representation and Cosine Similarity,” *J. Dinda Data Sci. Inf. Technol. Data Anal.*, vol. 4, no. 2, pp. 149–153, 2024.

[10] A. Kautsar and M. Syafrullah, “Implementasi Algoritme Multinomial Naïve Bayes Pada Analisis Sentimen Terhadap Isu Presiden 3 Periode,” in *Prosiding Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI)*, 2022, pp. 675–682. Accessed: Mar. 01, 2024. [Online]. Available: http://senafti.budiluhur.ac.id/index.php/senafti/article/view/77

[11] J. H. Joloudari, A. Marefat, M. A. Nematollahi, S. S. Oyelere, and S. Hussain, “Effective class-imbalance learning based on SMOTE and convolutional neural networks,” *Appl. Sci.*, vol. 13, no. 6, p. 4006, 2023.

[12] M. Story and R. G. Congalton, “Accuracy assessment: a user’s perspective,” *Photogramm. Eng. Remote Sens.*, vol. 52, no. 3, Art. no. 3, 1986.

[13] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, pp. 5–32, 2001.

[14] O. Chamorro-Atalaya *et al.*, “K-Fold Cross-Validation through Identification of the Opinion Classification Algorithm for the Satisfaction of University Students.,” *Int. J. Online Biomed. Eng.*, vol. 19, no. 11, 2023, Accessed: Feb. 22, 2025. [Online]. Available: https://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=26268493&AN=169957647&h=mvLfH5TxZM4cCGt5xauQqplvWeca%2FazN9k4Mk2nWIuexjJy3PJFYQ8vzsBJhzVcDuXIkxZQxo5nmgzpIth8Tjw%3D%3D&crl=c

[15] S. Sathyanarayanan and B. R. Tantri, “Confusion matrix-based performance evaluation metrics,” *Afr. J. Biomed. Res.*, pp. 4023–4031, 2024.

[16] R. Aryanti, T. Misriati, and A. Sagiyanto, “Analisis Sentimen Aplikasi Primaku Menggunakan Algoritma Random Forest dan SMOTE untuk Mengatasi Ketidakseimbangan Data,” *J. Comput. Syst. Inform. JoSYC*, vol. 5, no. 1, pp. 218–227, 2023.