Available online at http://www.iorajournal.org/index.php/ijgor/index

International Journal of Global Operations Research

e-ISSN: 2722-1016
p-ISSN: 2723-1739

Vol. 5, No. 4, pp. 275-287, 2024

Sentiment Analysis of Tiktok App Reviews on Google Play using Several Machine Learning Methods

Nurnisaa binti Abdullah Suhaimi[1*], Mugi Lestari[2]

[1,2] *Master's Program of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Padjadjaran, Jatinangor, West Java, Indonesia*

**Corresponding author email: nurnisaa23001@mail.unpad.ac.id*

Abstract

Sentiment analysis has become increasingly important in understanding user perceptions of digital platforms. This study focuses on analyzing TikTok application reviews from the Google Play Store in Indonesia using machine learning techniques. The research aims to investigate sentiment distribution and compare the performance of three popular machine learning models: Random Forest, Support Vector Machine (SVM), and Naive Bayes. The study employed a comprehensive methodology involving data collection, preprocessing, feature extraction, and model evaluation. A dataset of 10,000 TikTok reviews was collected and preprocessed using techniques such as case folding, tokenization, and stopword removal. The sentiment labeling process categorizes reviews into positive, negative, and neutral sentiments based on user ratings. The TF-IDF algorithm was used for feature extraction, and the SMOTE technique addressed class imbalance. Results revealed a predominance of negative sentiment (53.5%), followed by neutral (32.1%) and positive (14.4%) sentiment. Model performance comparisons at different data sharing ratios (80/20 and 70/30) demonstrated that Random Forest and SVM consistently outperformed Naive Bayes. At the 80/20 ratio, Random Forest achieved the highest accuracy of 83.73%, highlighting its effectiveness in sentiment classification. The research contributes to the field of sentiment analysis and natural language processing by providing insights into user experiences with the TikTok application in Indonesia. The findings can guide application developers in understanding user perceptions and improving user experience.

Keywords: Sentiment analysis, tiktok, machine learning, random forest, support vector machine, naive bayes.

1. Introduction

In the increasingly developing development, social media is closely related to everyday life and has changed the way people interact (Ausat, 2023). One of the plat-forms that is currently getting a lot of attention is the Tik-Tok application, TikTok is a mobile application and social media platform that allows users to create and share short videos (Quiroz, 2020). According to Newman et al., (2024), the number of TikTok users has increased rapidly in various countries, especially in Thailand, increasing by 39%, in Kenya 36% and in Indonesia increasing by 29%. This growth is because the TikTok ap-plication provides innovative content, presents entertainment content and can be a source of income for its users (Dewi et al., 2023).

Although TikTok users increase every year, this application is banned in several countries. Pathak, (2024) stated that there are at least 13 countries that ban this appli-cation such as Neval, Belgium, Canada, India (Newman et al., 2024). TikTok users are 77% under 30, 15% between 31 and 40, and 8% over 41 (Liu 2022). TikTok users are 77% under 30, 15% between 31 and 40, and 8% above 41. Around 45.2 percent of global TikTok users are female while male users on the popular social video platform make up 54.8 percent of the total users (Ceci, 2024).

The use of TikTok also brings various responses and perceptions from its users, the opinions expressed can be used as the main indicator to measure the level of public satisfaction. The information to be obtained involves questions, criticism, suggestions, and input.

Technology has enabled individuals to express their opinions on social media in response to various events. The reviews provided can be a very useful indicator to assess community satisfaction. The information collected includes questions, suggestions, criticisms, and appreciations. However, the challenge that arises is how to automatically categorize these opinions into positive, negative, and neutral classes. Therefore, it is important to develop an effective sentiment analysis method to accurately group these reviews. The problem that needs to be solved is how the

application of various machine learning models with varying training data ratios affects the model's ability to accurately classify review sentiments.

This study aims to conduct an in-depth sentiment analysis of TikTok user reviews in Indonesia, the focus of this study is on the comparison of three popular machine learning models, namely Random Forest, support vector Machine (SVM) and Naive Bayes. In addition, this study investigates how variations in the data sharing ratio of 80/20 to 70/30 affect the accuracy and reliability of this model. The main goal is not only to reveal insights into TikTok user sentiment in Indonesia, but also to provide method innovation in the field of sentiment analysis and machine learning.

This research is expected to provide significant contributions in the field of sentiment analysis and natural language processing, especially in the context of social media applications in Indonesia. By comparing the performance of various models and the ratio of training data, the results of this study can provide useful guidance for researchers and practitioners in choosing the optimal model for sentiment analysis. The limitations of this study are that it relies on review data available on the Google Play Store and may not cover all types of reviews or opinions outside the platform. In addition, variations in writing style and language can affect the accuracy of the analysis results.

2. Materials and Method

2.1. Data Collection

The data used in this research are comments (reviews) and ratings from TikTok application users on the Google Play Store website which were taken by scraping (Putra et al., 2019), involving 10.000 reviews (Hasanah, 2024). Web scraping is a technique used to collect data from websites automatically, this process involves using software to extract information from web pages, this technique allows for the collection of large amounts of data with greater efficiency than manual data collection (ten Bosch et al., 2018). The focus of data collection is directed at reviews written in Indonesian and originating from users in Indonesia (Uliniansyah et al., 2024), this parameter was chosen to obtain an accurate representation of the TikTok user experience in Indonesia.

2.2. Pre-processing

Data Preprocessing is the process of changing raw data into a form that is easier to understand, which aims to facilitate the mining process, make data easier to read, minimize mining time and simplify the data analysis process in machine learning (Roy et al., 2018; Pandey et al., 2020). In this study, the Preprocessing process includes Cleaning, Case Folding, Removing Special Characters, Tokenization and Remove Stopwords (Uysal and Gunal, 2014).

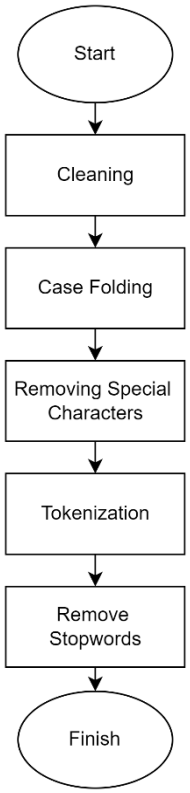


Figure 1: Pre-processing flow diagram

2.2.1. Cleansing

Data cleansing is the first stage in the data preprocessing process and the process of correcting or removing inaccurate, incomplete, or irrelevant data (Jamshed et al., 2019; Sarpong and Arthur, 2013). Data cleansing serves to handle missing values (Manimekalai and Kavitha, 2018), normalize noisy data (Roy et al., 2021), and identify and remove inconsistent or duplicate data (Kothandapani, 2021). The goal is to produce high-quality data (Ridzuan and Zainon, 2019) and increase overall productivity (Rahmani et al., 2021). In addition, unclean raw data can affect the accuracy of Machine Learning models (Borrohou et al., 2022).

2.2.2. Case Folding

Case Folding is one of the pre-processing stages that aims to change all letters in a document to lowercase (Yudhana et al., 2019), only letters "a" to "z" are accepted (Riyanto and Azis, 2022: RH, 2023). Characters other than letters (Nurkholis et al., 2022) and numbers such as punctuation and spaces are removed and considered delimiters (Mawardi et al., 2018. This delimiter can also be removed or ignored using the command in Python.

2.2.3. Tokenization

Tokenizers break down unstructured data and natural language text into pieces of information that can be treated as individual elements. The occurrence of a token in a document can be used directly as a vector representing the document. This method is useful for separating elements that make up text data and removing irrelevant characters, such as punctuation, spaces, or numbers.

2.2.4. Stopwords

Stopwords are common words that often appear in large numbers and are considered to have no significant meaning (Sarica and Luo, 2021). Examples of stopwords in Indonesian include "yang", "dan", "di", "dari", and so on (Pradana and Hayaty, 2019). The purpose of using stopwords is to remove words that have low information value from a text (Ladani and Desai, 2020).

2.3. Labeling Process

The obtained review data is then processed through the labeling stage based on the calculated sentiment score. Each review is labeled Positive, Negative and Neutral based on the score given by the user. Scores of 4 and 5 will be given a Positive sentiment value, a score of 3 will be given a Neutral sentiment value and a score of less than 3 will be given a Negative sentiment value (Imran et al., 2022). The labeling process in sentiment analysis aims to identify emotions, attitudes and opinions towards a particular subject (Lee and Kim, 2017). In addition, labeling also aims to determine whether reviews fall into the Positive, Negative or Neutral sentiment category (Vitianingsih, 2024).

2.4. Feature Extraction

The use of the TF-IDF algorithm aims to give weight to words in a document (AlShammari, 2023) and assess the importance of those words in text classification (Liang and Niu, 2022). TF-IDF calculates weights based on the frequency of occurrence of words in a document (TF) and the distribution of those words across documents (IDF) (Guo and Yang, 2016). Fan and Qin (2018) state that the importance of a word increases proportionally to the number of times it appears in a document, but at the same time it decreases inversely to the frequency of its occurrence in the corpus. The number of common words in a text is often high and often includes some irrelevant words. Usually, based on a certain filtering strategy, entries that have a significant contribution to the classification are selected to classify the text. The step that ought to be done is to calculate the Term Recurrence (TF) and Report Recurrence (DF), at that point calculate the number of reverse frequencies that can be seen from the taking after condition (Wardani et al., 2022):

$$w(t,d) = tf(t.d) \times idf(t) = tf(t,d) \times log\frac{N}{df(t)} \quad (1)$$

where

$w(t,d)$: TF-IDF weight or tern (t) wight in document (d)
$tf(t,d)$: number of occurrencesof term (t) in document (d)
$idf(t)$: number of document frequency inverses per word
$df(t)$: number of document frequencies per word
N : total number of document

2.5. Data processing

Class Imbalance in machine learning oversampling is a common problem in machine learning, especially in classification problems, imbalanced data can hamper model accuracy. In this study, the Synthetic Minority Over-sampling Technique (SMOTE) will be used (Larse, 2022; Pears et al., 2014), which aims to generate new synthetic samples for the minority class, which aims to improve the balance between the majority class and the minority class (Fauzan et al., 2023).

2.6. Data Splittng

The purpose of dividing the data into training and testing sets is to prevent overfitting, where the model learns too much from the training data, including noise and irrelevant details (Ying, 2019). It is recommended to divide the data into training and testing data with a proportion of 70-80% for training data and 20-30% for testing data to get valid estimates (Gholamy et al., 2018). In this study, data distribution will be carried out with various ratios to ensure an optimal model and to evaluate model performance in various data proportions.

2.7. Model Implementation

2.7.1. Naïve Bayes

Naive Bayes works on the principle of Bayes' Theorem, which calculates the probability of a text being included in a class based on the words that appear in the text. This method assumes that the features (words) are independent of each other, making it possible to calculate the joint probability of all features in the text.

2.7.2. Support Vector Machine (SVM)

Support Vector Machine (SVM) is one of the machine learning algorithms used for classification and regression. SVM works by finding the best hyperplane that separates data into different classes in a feature space (Awad et al., 2015). The main concept of SVM is to find the maximum margin between the hyperplane and the nearest data point of each class. A larger margin usually results in a better model in terms of generalization to new data. SVM is effective in handling high-dimensional data and can be used for both linear and non-linear classification by using the kernel trick to map data to a higher feature space.

2.7.3. Random Forest

Random Forest is an ensemble learning algorithm that combines multiple decision trees to improve prediction accuracy. In Random Forest, each decision tree is trained on a random subset of the training data, and the final prediction is generated by combining the results of all decision trees. This technique reduces the risk of overfitting that often occurs with a single decision tree by creating a more robust and accurate model. Random Forest can also handle data with many features and provides information about the importance of features in the classification process.

2.8. Model Evaluation

2.8.1. Accuracy

Accuracy is a metric that shows the proportion of correct predictions compared to the total predictions made. It measures how often a model makes correct predictions across the entire data set. Accuracy is often used as a primary measure of model performance, but it can be misleading if the data is imbalanced.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

where

TP : true positives
TN : true negatives
FP : false positives
FN : false negatifes

2.8.2. Precision

Precision measures the proportion of correct positive predictions out of all positive predictions made by the model. This is important when the cost of false positives is high, such as in medical diagnosis. High precision means the model rarely produces false positives.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

where
TP : true positives
FP : false positives

2.8.3. Recall

Recall is a metric that shows the proportion of all true positive cases that the model correctly predicted. It is useful when the cost of false negatives is high, such as in spam detection. High recall means the model was able to detect most of the positive cases.

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

where
TP : true positives
FN : false negatifes

2.8.4. F1-Score

F1-Score is a performance measure that is the harmonic mean of Precision and Recall. F1-Score is particularly useful when there is class imbalance, i.e. when one class has more data than the other. It helps combine both metrics (Precision and Recall) to provide one overall score that reflects the overall performance of the model.

$$F1 = 2 \times \frac{Preccision \times Recall}{Precision + Recall} \tag{5}$$

3. Results and Discussion

3.1. TikTok review scores

Although TikTok has an overall rating of 4,2 stars from 19 million reviews on the Google Play Store, an analysis of 10,000 recent reviews from users in Indonesia shows different results. The results of this study clearly show the difference, as can be seen in Figure 2.

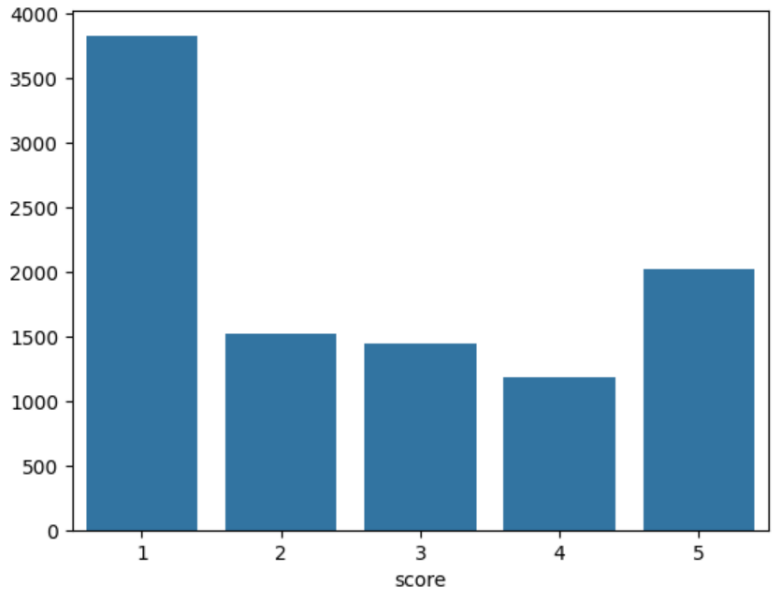


Figure 2: Tiktok application review rating

The reviews collected are mostly dominated by a 1-star score or around 4000 reviews. Reviews with a score of 2 and 3 stars have almost the same number, ranging from 1300 to 1600 reviews. Meanwhile, reviews with a score of 4

stars are fewer, indicating that only a few users give this application a fairly good rating. A score of 5 stars, which represents the highest satisfaction, is in second place after a score of 1, with around 2000 reviews. This difference could be due to several factors, such as recent changes to the application that users may not like, specific issues experienced in Indonesia, or bias in the latest review sample that is more directed at users who have had negative experiences.

3.2. Sentiment Distribution

A score of 1 is given to text that shows positive sentiment, such as praise or user satisfaction with a product or service. For example, a review that says, "This app is great and helpful!" would be given a score of 1 because it shows user satisfaction. Conversely, a score of -1 is used for text that contains negative sentiment, such as complaints, disappointment, or criticism. For example, a comment like "The app often crashes and is slow" is given a score of -1 because it reflects a negative user experience. Meanwhile, a score of 0 is given to neutral text that has no emotional tendencies, either positive or negative, such as statements that contain factual information or opinions that do not emphasize feelings, such as "This app works well, but it's just so-so".

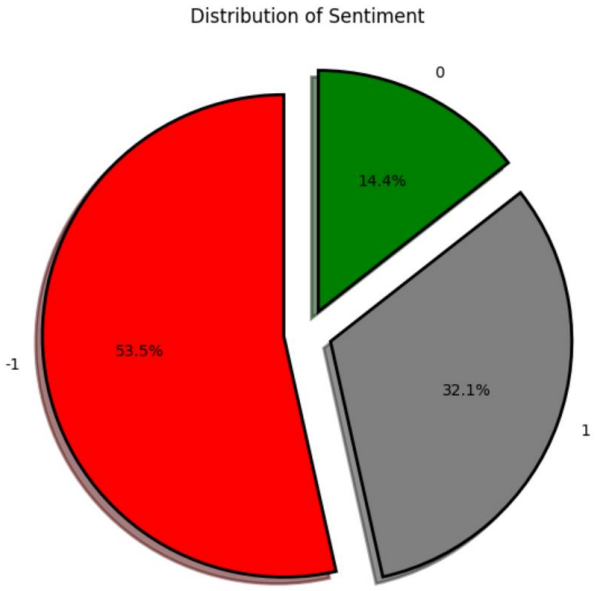


Figure 3: Distribution of sentiment

Based on the graph above, the distribution of TikTok application review sentiment, negative sentiment dominates with a proportion of 53.5%, followed by neutral sentiment at 32.1%, and the least positive sentiment, which is 14.4%. The high proportion of negative sentiment indicates that the majority of users who provide reviews are dissatisfied with their experience when using the TikTok application. Neutral sentiment which reaches one third of the total reviews indicates that there is a group of users who have a neutral view, may not be too impressed but also do not experience serious problems. Meanwhile, only a small number of users provide positive reviews, indicating that only a few users are very satisfied with this application.

3.3. Word Cloud Analysis

Word Cloud itself is a visualization technique that depicts a visual representation of word frequency. Word Cloud itself is quite popular in text mining because it can provide a visual display of the words to be analyzed but remains informative. In this study, Word Cloud frequency is separated into positive, negative and neutral.

Figure 4: Positive sentiment

Figure 6: Negative sentiment

Figure 7: Netral sentiment

3.4. Model comparison

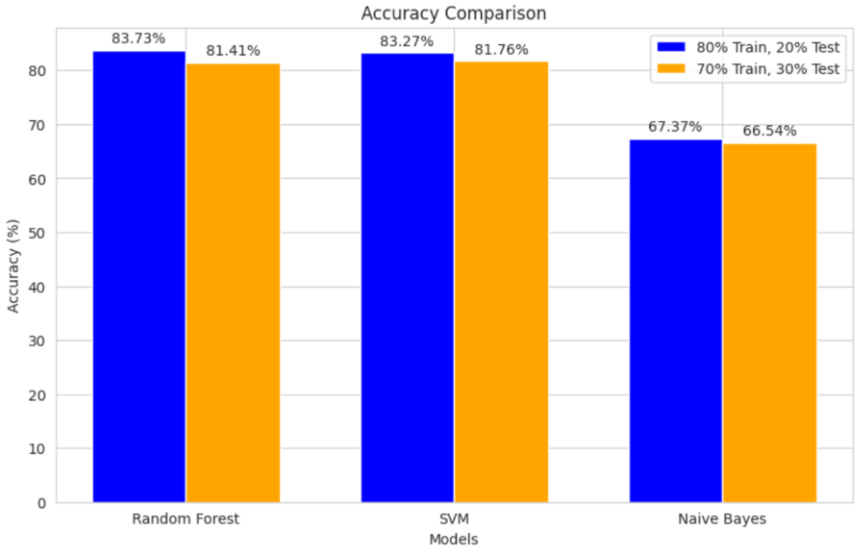


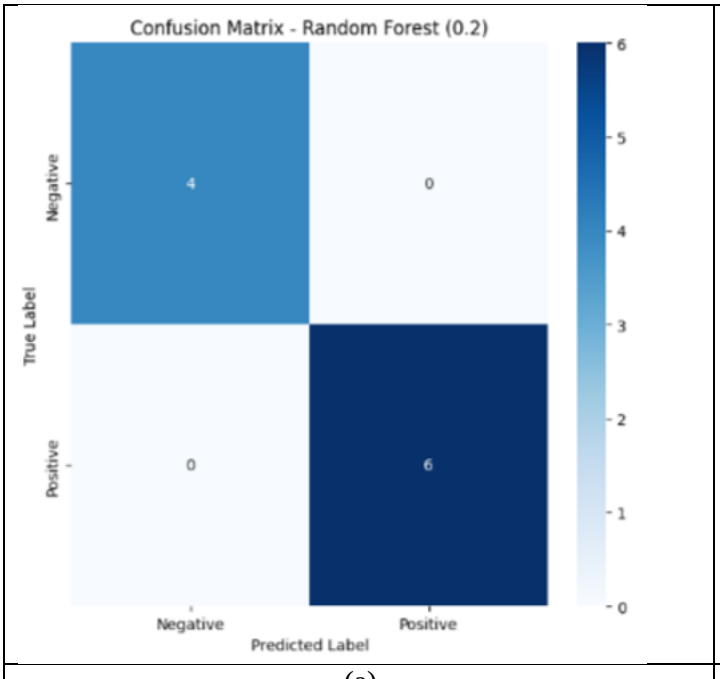
Figure 8: Accuracy comparison of each machine learning modality

The graph shows the accuracy results for three classification algorithms, namely Random Forest, Support Vector Machine (SVM), and Naive Bayes. At two different training and testing data sharing ratios: 80% training and 20% testing, and 70% training and 30% testing. At a ratio of 80% training and 20% testing, Random Forest showed the highest accuracy among the three models, which was 83.73%, followed by SVM with an accuracy of 83.27%. Naive Bayes recorded a lower accuracy, which was 67.37%.

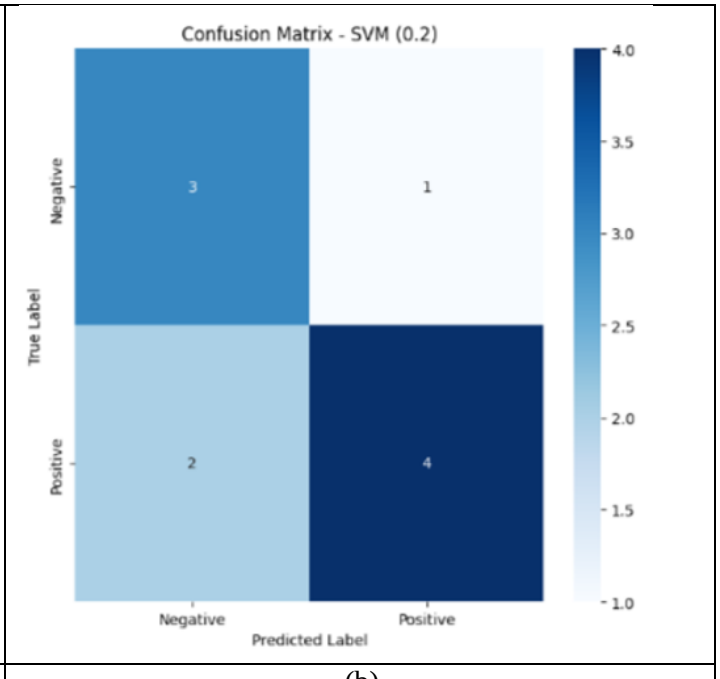
When the data sharing ratio was changed to 70% training and 30% testing, Random Forest's accuracy decreased slightly to 81.41%, while SVM's accuracy increased to 81.76%. Naive Bayes also experienced a decrease in accuracy to 66.54%. Overall, the results show that Random Forest and SVM consistently provide better performance compared to Naive Bayes, although the difference between Random Forest and SVM is relatively small. Naive Bayes showed less stable performance, with lower accuracy than the other two models at both data sharing ratios.

3.5. Confusion Matrix

The results of the performance evaluation of three classification models, namely Random Forest, SVM (Support Vector Machine), and Naive Bayes, with two data sharing ratios (20% and 30%) are displayed through the confusion matrix in Figure 9.



(a)



(b)

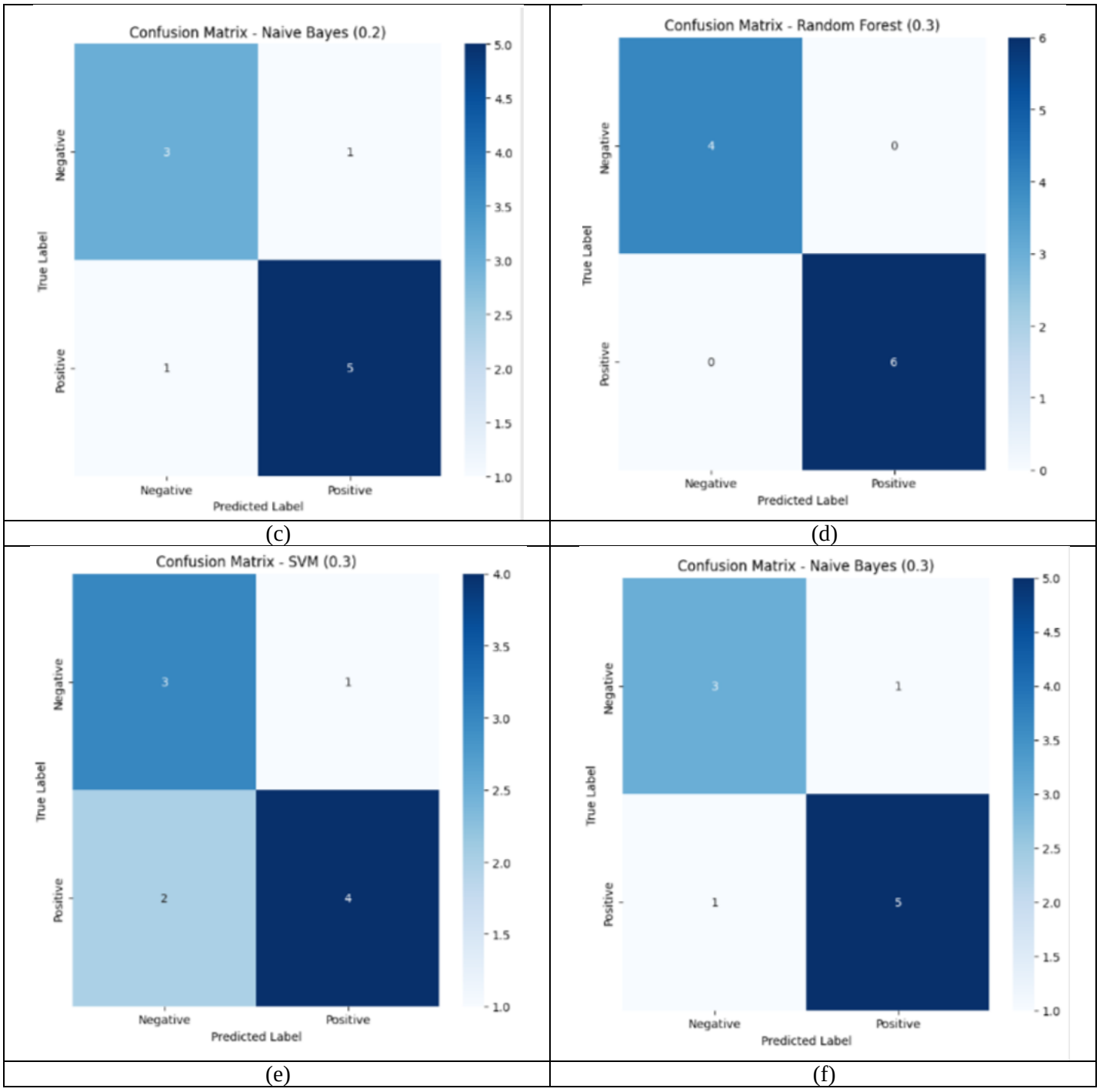


Figure 9: (a) Confusion Matrix in random forest (0.2), (b) Confusion Matrix in SVM (0.2), (c) Confusion Matrix in Naive Bayes (0.2), (d) Confusion Matrix in Random Forest (0.3), (e) Confusion Matrix in SVM (0.3), (f) Confusion Matrix in Naive Bayes (0.3)

The confusion matrix for Random Forest with a data sharing ratio of 20% in figure (a) shows that this model has a very good performance without any prediction errors. The model successfully predicted four negative reviews and six positive reviews correctly, resulting in a True Negative (TN) value of 4 and a True Positive (TP) of 6. There were no False Positive (FP) or False Negative (FN) in this prediction, indicating that Random Forest at this ratio was able to distinguish negative and positive reviews with perfect accuracy. These results indicate that Random Forest is a very reliable model for sentiment analysis on this data sharing.

In Random Forest with a data sharing ratio of 30% shown in figure (d), the model again shows very good performance with perfect accuracy, the same as at a ratio of 20%. The model successfully predicted all reviews correctly, both negative and positive, resulting in four True Negatives (TN) and six True Positives (TP), with no False Positives (FP) or False Negatives (FN). This indicates that Random Forest is very effective and consistent in correctly classifying different data splits.

The confusion matrix results show no prediction errors at both data split ratios, namely 0.2 and 0.3. All predictions were successfully classified correctly. There are 4 True Negatives (TN) which means the model successfully

identified 4 negative samples correctly, and 6 True Positives (TP) which indicate that the model can also classify all positive samples accurately. Since there are no False Positives (FP) or False Negatives (FN), this indicates that the model never misclassifies negative samples as positive, or vice versa. As a result, all evaluation metrics such as accuracy, precision, recall, and F1-score reach a value of 1.0 (100%). This means that the Random Forest model at data splits of 0.2 and 0.3 has perfect performance in classifying data without any errors.

In the SVM (Support Vector Machine) shown in figure (b) with a data sharing ratio of 20%, the model showed some errors in prediction. Out of four negative reviews, three were predicted correctly, while one review was incorrectly predicted as positive. In addition, out of six positive reviews, only four were predicted correctly while the remaining two were incorrectly predicted as negative. This resulted in one False Positive (FP) and two False Negative (FN). These results indicate that SVM has difficulty in accurately separating positive and negative reviews, so its performance is not as good as Random Forest at the same ratio.

The SVM with a 30% data split ratio in figure (e) shows similar results to the 20% ratio, with the same error rate. Out of the four negative reviews, three were correctly predicted while one negative review was incorrectly predicted as positive, resulting in one False Positive (FP). Out of the six positive reviews, four were correctly predicted while the other two were incorrectly predicted as negative, resulting in two False Negatives (FN). This shows that although the SVM was able to correctly classify some reviews, the model consistently made errors that reduced its effectiveness in separating positive and negative reviews.

In the SVM model with a data split ratio of 0.2 and 0.3, the confusion matrix results show some prediction errors. There are 3 True Negatives (TN) and 4 True Positives (TP), which means the model is able to correctly identify most of the negative and positive samples. However, there is 1 False Positive (FP) which means the model incorrectly classifies 1 negative sample as positive, and 2 False Negatives (FN) which indicate 2 positive samples are incorrectly classified as negative. As a result of these prediction errors, the model's accuracy drops to around 70%. Precision is recorded at 0.8, which means that out of all the positive predictions generated by the model, 80% of them are truly positive. However, the lower recall at 0.6667 indicates that the model is not able to accurately capture all positive samples, as there are some positive samples that are incorrectly classified as negative. The combination of precision and recall produces an F1-score of 0.7273, which illustrates the balance between the two metrics.

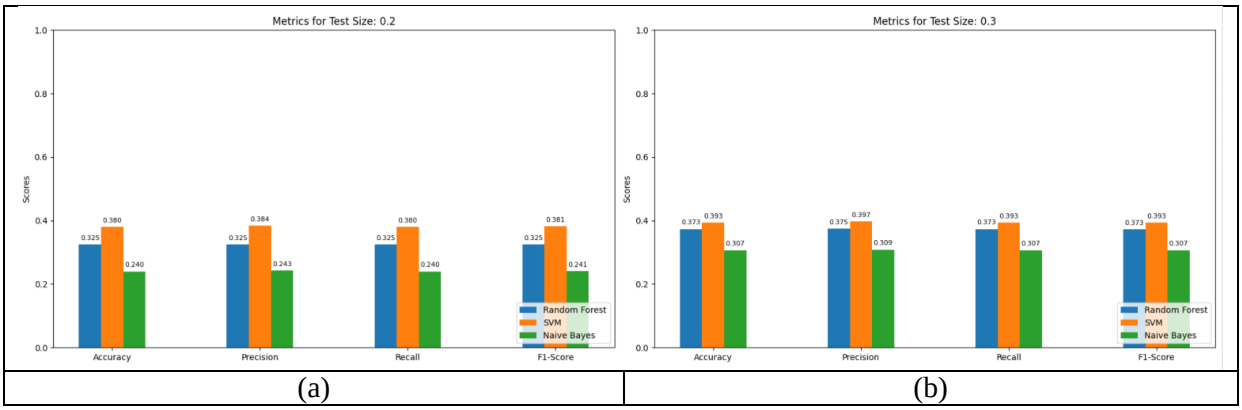
Naive Bayes at a data sharing ratio of 20% in figure (c) shows quite good performance but there are still errors. Of the four negative reviews, three were successfully predicted correctly, while one negative review was incorrectly predicted as positive, resulting in one False Positive (FP). Of the six positive reviews, five were correctly predicted and one review was incorrectly predicted as negative, resulting in one False Negative (FN). Although more accurate than SVM, the Naive Bayes model still shows room for improvement, especially in minimizing prediction errors between classes.

Naive Bayes at a data sharing ratio of 30% in figure (f) also shows similar results to the ratio of 20%, with three True Negatives (TN) and five True Positives (TP). Similar to before, one negative review was incorrectly predicted as positive, resulting in one False Positive (FP), and one positive review was incorrectly predicted as negative, resulting in one False Negative (FN). This shows that the performance of Naive Bayes is relatively stable but there are still errors that need to be addressed, especially in distinguishing reviews that are on the border between the positive and negative classes.

In the Naive Bayes model, the confusion matrix shows quite stable results at data sharing ratios of 0.2 and 0.3. There are 3 True Negatives (TN) and 5 True Positives (TP), indicating that most of the negative and positive predictions are correctly classified. However, there is 1 False Positive (FP) and 1 False Negative (FN), meaning that there is an error in classifying one negative sample as positive, and one positive sample as negative, respectively. The model accuracy reaches 80%, indicating that 80% of the total predictions generated by the model are correct. Precision and recall are each 0.8333, meaning that the model can identify positive samples well, both in terms of positive predictions and the ability to capture all positive samples. The F1-score value is also 0.8333, indicating a fairly good balance between precision and recall in this model.

3.6. Visualization

In this study, the visualization will be displayed in a bar graph. Where, the blue color displays the results of random forest, the orange color displays the results of Support Vector Machine (SVM) and the green color displays the results of Naïve bayes.



(a) (b)

Figure 10: (a) Analysis of model performance results on test size 0.2, (b) Analysis of model performance results on test size 0.3

Based on the visualization shown, it can be seen that the performance of the Random Forest, SVM, and Naive Bayes models is compared using four main metrics Accuracy, Precision, Recall, and F1-Score. On the test set size of 0.2 (left graph), Random Forest consistently shows the best performance in all metrics, especially in Accuracy, Recall, and F1-Score, with the highest value compared to the other two models. SVM stands out in the Precision metric, but for other metrics, its performance is below Random Forest. Naive Bayes has the lowest performance in all metrics on this test set size.

On the test set size of 0.3 (right graph), Random Forest still shows superior performance in the Accuracy metric, although the Precision and Recall values are quite balanced with Naive Bayes. On this test set, Naive Bayes shows an increase in performance, especially in Precision and F1-Score, which are almost on par with Random Forest. SVM, on the other hand, has less than optimal performance in terms of Recall and F1-Score, although it is still able to compete in Precision.

4. Conclusion

Explain what has been done, and draw conclusions in accordance with the objectives of the research that has been determined. The conclusions are delivered narratively, do not contain equations, tables, and figures. The Random Forest model consistently outperforms SVM and Naive Bayes across different test set sizes, particularly excelling in Accuracy, Recall, and F1-Score. On the 0.2 test set size, Random Forest achieves the highest values across all metrics except Precision, where SVM performs best. However, Naive Bayes has the weakest performance across all metrics for this test set size. On the 0.3 test set size, Random Forest maintains its dominance in Accuracy, while Naive Bayes shows noticeable improvement, particularly in Precision and F1-Score, becoming comparable to Random Forest. In contrast, SVM's performance is less competitive in Recall and F1-Score, although it still demonstrates strength in Precision. These results highlight the robustness of the Random Forest model in achieving balanced performance across metrics.

References

AlShammari, A. F. (2023). Implementation of Keyword Extraction using Term Frequency-Inverse Document Frequency (TF-IDF) in Python. *International Journal of Computer Applications*, *975*, 8887.

Ausat, A. M. A. (2023). The role of social media in shaping public opinion and its influence on economic decisions. *Technology and Society Perspectives (TACIT)*, *1*(1), 35-44.

Awad, M., Khanna, R., Awad, M., & Khanna, R. (2015). Support vector machines for classification. *Efficient learning machines: Theories, concepts, and applications for engineers and system designers*, 39-66.

Borrohou, S., Fissoune, R., Badir, H., & Tabaa, M. (2022, May). Data Cleaning in Machine Learning: Improving Real Life Decisions and Challenges. In *International Conference on Advanced Intelligent Systems for Sustainable Development* (pp. 627-638). Cham: Springer Nature Switzerland.

Ceci, L. (2024). Global TikTok user distribution by gender 2024. Statista. https://www.statista.com/statistics/1299785/distribution-tiktok-users-gender/

Dewi, I. A. S. K., Cahyaningrum, F. S., Darmawan, Y., & Setyono, V. I. (2023). The factors that influence TikTok popularity as a

digital marketing technique to grow customer engagement. *International Journal of Economics, Business and Innovation Research, 2*(02), 103-111.

Fan, H., & Qin, Y. (2018, May). Research on text classification based on improved tf-idf algorithm. In *2018 International Conference on Network, Communication, Computer Engineering (NCCE 2018)* (pp. 501-506). Atlantis Press.

Fauzan, I. K., Makhtar, M., Rosly, R., & Sambas, A. (2023). Performance evaluation of classifiers for the COVID-19 symptom-based dataset using different feature selection methods. *International Journal of Advanced Technology and Engineering Exploration, 10*(103), 741.

Gholamy, A., Kreinovich, V., & Kosheleva, O. (2018). Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation. Int. J. Intell. *Technol. Appl. Stat, 11*(2), 105-111.

Guo, A., & Yang, T. (2016). Research and improvement of feature words weight based on TFIDF algorithm. In *2016 IEEE information technology, networking, electronic and automation control conference* (pp. 415-419). IEEE.

Hasanah, K. (2024). Comparison of Sentiment Analysis Model for Shopee Comments on Google Play Store. *Jurnal Sisfokom (Sistem Informasi dan Komputer),* 13(1), 21-30.

Imran, A. S., Yang, R., Kastrati, Z., Daudpota, S. M., & Shaikh, S. (2022). The impact of synthetic text generation for sentiment analysis using GAN based models. *Egyptian Informatics Journal, 23*(3), 547-557.

Jamshed, H., Khan, S. A., Khurrum, M., Inayatullah, S., & Athar, S. (2019). Data Preprocessing: A preliminary step for web data mining. *3c Tecnología: glosas de innovación aplicadas a la pyme, 8*(1), 206-221.

Kothandapani, H. P. (2021). A benchmarking and comparative analysis of python libraries for data cleaning: Evaluating accuracy, processing efficiency, and usability across diverse datasets. *Eigenpub Review of Science and Technology, 5*(1), 16-33.

Kurniawan, S. A., Rimenda, T., Buntoro, A., & Mirati, R. E. (2023). Consumer Reviews on TikTok and Their Impact on Millennial Saving Decisions. *Kontigensi: Jurnal Ilmiah Manajemen, 11*(2), 483-490.

Ladani, D. J., & Desai, N. P. (2020, March). Stopword identification and removal techniques on tc and ir applications: A survey. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)* (pp. 466-472). IEEE.

Larsen, B. S. (2022). Synthetic minority over-sampling technique (SMOTE). GitHub (https://github. com/dkbsl/matlab_smote/releases/tag/1.0).

Lee, S., & Kim, W. (2017). Sentiment labeling for extending initial labeled data to improve semi-supervised sentiment classification. *Electronic Commerce Research and Applications, 26*, 35-49.

Liang, M., & Niu, T. (2022). Research on text classification techniques based on improved TF-IDF algorithm and LSTM inputs. *Procedia Computer Science, 208*, 460-470.

Liu, K. (2022). Research on the core competitiveness of short video industry in the context of big data—a case study of tiktok of bytedance company. *American Journal of Industrial and Business Management, 12*(4), 699-730.

Manimekalai, K., & Kavitha, A. (2018). Missing value imputation and normalization techniques in myocardial infarction. *ICTACT Journal on Soft Computing, 8*(03), 8.

Mawardi, V. C., Susanto, N., & Naga, D. S. (2018). Spelling correction for text documents in Bahasa Indonesia using finite state automata and Levinshtein distance method. In *MATEC web of conferences* (Vol. 164, p. 01047). EDP Sciences.

Newman, N., Fletcher, R., Robertson, C. T., Ross Arguedas, A., & Nielsen, R. K. (2024). Reuters Institute digital news report 2024.

Nurkholis, A., Alita, D., & Munandar, A. (2022). Comparison of kernel support vector machine multi-class in PPKM sentiment analysis on twitter. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 6*(2), 227-233.

Pandey, N., Patnaik, P. K., & Gupta, S. (2020). Data pre-processing for machine learning models using python libraries. Int. J. Eng. Adv. *Technol, 9*(4), 1995-1999.

Pathak, S. (2024). Cyber warfare, influence operations, and TikTok bans. *Hindustan Times.*

Pears, R., Finlay, J., & Connor, A. M. (2014). Synthetic Minority over-sampling technique (SMOTE) for predicting software build outcomes. arXiv preprint arXiv:1407.2330.

Pradana, A. W., & Hayaty, M. (2019). The effect of stemming and removal of stopwords on the accuracy of sentiment analysis on indonesian-language texts. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 375-380.

Putra, R. R., Johan, M. E., & Kaburuan, E. R. (2019). A naïve bayes sentiment analysis for fintech mobile application user review in Indonesia. *International Journal of Advanced Trends in Computer Science and Engineering, 8*(5), 1856-1860.

Quiroz, N. T. (2020). TikTok. Revista argentina de estudios de juventud.

Rahmani, D. A., Kusumasari, T. F., & Alam, E. N. (2021, October). Addition of Process Decomposition in Open Source Tools-Based Cleansing Data Modules. In *2021 International Conference on Data Science and Its Applications (ICoDSA)* (pp. 129-134). IEEE.

RH, S. B. (2023). Stemming Algorithm Modification for Overstemming Cases. *Journal of Computers for Society, 4*(2), 105-112.

Ridzuan, F., & Zainon, W. M. N. W. (2019). A review on data cleansing methods for big data. *Procedia Computer Science, 161*, 731-738.

Riyanto, R., & Azis, A. (2022). Implementation of the Jaccard Similarity Algorithm on Answer Type Description. *International Journal of Informatics and Information Systems, 5*(2), 76-83.

Roy, A., Ghosh, S., Ghosh, K., & Ghosh, S. (2021). An unsupervised normalization algorithm for noisy text: a case study for information retrieval and stance detection. *Journal of Data and Information Quality (JDIQ), 13*(3), 1-25.

Roy, S., Sharma, P., Nath, K., Bhattacharyya, D. K., & Kalita, J. K. (2018). Pre-processing: a data preparation step. *Encyclop Bioinform Comput Biol ABC Bioinform, 463*.

Sarica, S., & Luo, J. (2021). Stopwords in technical language processing. *Plos one, 16*(8), e0254937.

Sarpong, K. A. M., & Arthur, J. K. (2013). Analysis of data cleansing approaches regarding dirty data-a comparative study. *International Journal of Computer Applications, 76*(7).

ten Bosch, O., Windmeijer, D., van Delden, A., & van den Heuvel, G. (2018. Web scraping meets survey design: combining forces. In *Big Data Meets Survey Science Conference, Barcelona, Spain.*

Uliniansyah, M. T., Budi, I., Nurfadhilah, E., Afra, D. I. N., Santosa, A., Latief, A. D., ... & Sampurno, T. (2024). Twitter dataset on public sentiments towards biodiversity policy in Indonesia. *Data in Brief, 52*, 109890.

Vitianingsih, A. V. (2024). Sentiment Analysis of Brand Ambassador Influence on Product Buyer Interest Using K-NN and SVM. *Indonesian Journal of Artificial Intelligence and Data Mining (IJAIDM), 7*(2), 327-336.

Ying, X. (2019). An overview of overfitting and its solutions. In *Journal of physics: Conference series* (Vol. 1168, p. 022022). IOP Publishing.

Yudhana, A., Fadlil, A., & Rosidin, M. (2019). Indonesian words error detection system using nazief adriani stemmer algorithm. *International Journal of Advanced Computer Science and Applications, 10*(12).