Available online at https://journal.rescollacomm.com/index.php/ijqrm/index

International Journal of Quantitative Research and Modeling

e-ISSN 2721-477X
p-ISSN 2722-5046

Vol. 5, No. 3, pp. 331-340, 2024

Sentiment Analysis of Maxim App User Reviews in Indonesia Using Machine Learning Model Performance Comparison

Rifki Saefullah[1*], Setyo Luthfi Okta Yohandoko[2], Agung Prabowo[3]

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

**Corresponding author email: rifki23008@mail.unpad.ac.id*

Abstract

User reviews can vary widely in language and writing style, which can make accurate sentiment modeling difficult. Selecting the right machine learning model and comparing performance between models can be challenging, given that each model has its own strengths and weaknesses. The method used involved data collection by scraping 5000 reviews from the Google Play Store, followed by data pre-processing including data cleaning, tokenization, stemming, and feature engineering using TF-IDF. The data was divided into training (70%) and testing (30%) sets, with the SMOTE oversampling technique applied to address class imbalance. Three machine learning models were used: Random Forest, Support Vector Machine (SVM), and Naive Bayes. The results showed that the majority of reviews were positive, with a high average app rating. Word cloud analysis revealed that "service", "driver", "price", and "time" were the most frequently discussed aspects in the reviews. In terms of model performance, SVM performed the best with an accuracy of 91.3%, followed by Random Forest (89%) and Naive Bayes (78%). Maxim was generally well received by users in Indonesia, with the majority of reviews being positive. The SVM model proved to be the most effective in classifying review sentiment, outperforming other models in accuracy and precision.

Keywords: Sentiment analysis, maxim application, machine learning, support vector machine (SVM), random forest, naive bayes.

1. Introduction

In the ever-evolving digital era, ride-hailing apps have become an integral part of the daily lives of urban Indonesians. One of the emerging players in this industry is Maxim, an app that offers a variety of transportation and delivery services (Nurhalifah et al., 2023). Maxim's presence in the competitive Indonesian market has elicited a variety of responses from users, which are reflected in their reviews on app distribution platforms such as the Google Play Store (Panggarra and Mustika, 2024).

User reviews are an invaluable source of information for app developers and alike researchers. They provide first-hand insights into user experiences, preferences, and areas for improvement (Sadiq et al., 2021). However, with the large volume of reviews, manual analysis becomes impractical and prone to subjective bias. This is where machine learning-based sentiment analysis becomes a crucial tool.

Sentiment analysis, a branch of natural language processing (NLP), allows us to automatically extract and categorize opinions from text. In the context of app reviews, this can help identify overall sentiment trends, uncover features that users like or dislike the most, and even detect emerging issues before they become more serious (Solangi et al., 2018).

This research focuses on the Maxim app in Indonesia for several reasons. First, Indonesia is one of the largest markets for ride-hailing apps in Southeast Asia (Fauzi and Sheng, 2021), with a large population and high technology adoption rates. Second, as a relative newcomer to the market, Maxim faces unique challenges in building a user base and competing with established players. Understanding user sentiment can be key to their growth and customer retention strategies (Wulansari et al., 2024).

The methodology used in this study combines data scraping techniques for review collection (Li et al., 2019), comprehensive text pre-processing, and the application of three different machine learning models Random Forest, Support Vector Machine (SVM), and Naive Bayes (Pranckevičius and Marcinkevičius, 2017). The use of multiple models allows for performance comparison and provides a deeper understanding of the strengths and weaknesses of each approach in the context of sentiment analysis.

One important aspect of this study is the handling of class imbalance, a common problem in sentiment analysis where one sentiment category may be significantly more numerous than the others. To address this, oversampling techniques such as SMOTE (Synthetic Minority Over-sampling Technique) are applied (Tarawneh et al., 2020), ensuring that the trained models can accurately recognize and classify all sentiment categories.

Furthermore, this study not only aimed to classify sentiment, but also to extract key themes and specific aspects of Maxim's service that users commented on most frequently. This information is invaluable to product development and customer service teams, allowing them to focus on areas that most impact user satisfaction.

The implications of this study extend beyond the context of Maxim or even the ride-hailing industry (Subriadi and Baturohmah, 2023). The methodology and insights generated can be applied to a variety of mobile applications and other digital platforms. In a rapidly changing digital landscape, where user feedback can influence the success or failure of a product, the ability to quickly and accurately analyze user sentiment becomes a significant competitive advantage.

By combining sentiment analysis, machine learning, and a deep understanding of business and social contexts, this study aims to provide actionable and meaningful insights into the Maxim app user experience in Indonesia. The results and methodology used are expected to contribute not only to the development of the Maxim app, but also to a broader understanding of consumer preferences and behavior in Indonesia's growing digital economy.

2. Methodology

2.1. Data collection

In data collection, reviews were taken from users of the Google Classroom application (Alim et al., 2019; Kumar and Bervell, 2019), namely by scraping on the Google Play Store using python to automatically scrape data from websites (Aldabbas et al., 2020). A total of 5000 reviews were used consisting of a collection of reviews or comments that had been labeled with sentiment.

2.2. Pre-Processing

In the data pre-processing stage, several important steps are taken to ensure that the data review is ready to be used in sentiment analysis. In this study, data pre-processing will be carried out as shown in Figure 1.

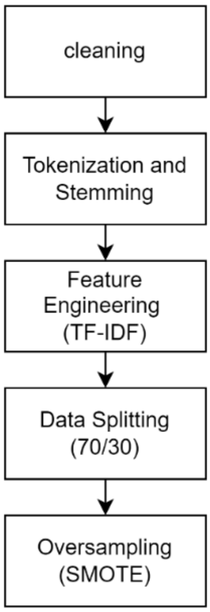

Figure 1: Pre-processing stages

2.2.1. Data cleaning

Data cleaning is the first step in data analysis, especially for text data. In this process, all irrelevant or insignificant elements are removed from the text to improve the quality of the data. For example, special characters such as punctuation, numbers, and symbols that do not directly contribute to understanding sentiment are removed (Van der Loo and De Jonge, 2018; Murshed et al., 2021). This is done to remove noise from the text, so that the machine learning model can focus on more meaningful content. In addition, elements that may be irrelevant, such as double spaces or unstructured text formats, are also cleaned.

2.2.2. Tokenization and Stemming

Tokenization is the process of breaking down text into smaller units called “tokens.” Tokens can be words, phrases, or even characters depending on the purpose of the analysis (Song et al., 2020). In the context of sentiment analysis or text processing in general, tokenization helps isolate important words or phrases from a text, separating them from the larger context so they can be analyzed individually (Alyafeai et al., 2023). For example, the sentence “I like this app” would be broken down into separate tokens: [“I”, “like”, “app”, “this”]. This process is important because machine learning models have an easier time processing data in the form of separate tokens than in the whole text.

Stemming is the process of reducing a word to its base form or "stem" (Singh and Gupta, 2017). The goal of stemming is to convert variations of a word with the same meaning, such as the words "run", "running", and "running", all into "run" (Rianto et al., 2021). This process helps reduce the complexity of the text, allowing machine learning models to recognize words with the same root as the same entity, regardless of their form. Stemming improves efficiency and accuracy in sentiment analysis because the model does not have to deal with multiple forms of a word that have similar meanings.

2.2.3. Feature engineering

At this stage the text is converted into a numeric vector representation using methods such as TF-IDF (Term Frequency-Inverse Document Frequency). TF-IDF calculates how important a word in a single document is relative to the entire corpus of review data (Ririanti and Purwinarko, 2021). The result is a feature matrix with thousands of dimensions, where each dimension represents a word or token from the review (Naeem et al., 2022). The formulas of TF-IDF are as follows (Xu et al., 2019):

$$w(t_i,d)=\frac{tf(t_i,d)\times idf(t_i)}{\sqrt{\sum_{t_i\in d}\left[tf(t_i,d)\times idf(t_i)\right]^2}} \tag{1}$$

$$idf(t_i)=\log\left(\frac{N}{nt_i}\right)+1 \tag{2}$$

where:
$w(t_i,d)$: weight of word t_i in document d
$tf(t_i,d)$: frequency of word t_i in document d
N : total number of documents
nt_i : number of documents in which word t_i appears

2.2.4. Data Splitting

In the data splitting stage, the Maxim app review data was divided into two main sets, namely, the training set and the testing set. This division was carried out with a proportion of 70% for the training set and 30% for the testing set (Nguyen et al., 2021). The training set is used to train the model, allowing the algorithm to learn from the data and make generalizations, while the testing set is used to evaluate the performance of the trained model (Xu and Goodacre, 2018). This division aims to ensure that the resulting model can be objectively evaluated on data that has never been seen before, providing an accurate indication of the model's ability to deal with real data.

2.2.5. Oversampling

Certain classes, such as negative or neutral sentiment, had a much smaller number of samples compared to other classes. To overcome this problem, the SMOTE (Synthetic Minority Over-sampling Technique) oversampling technique was applied (Mansourifar and Shi, 2020). SMOTE works by creating new examples of the minority class by synthesizing data from existing examples, rather than simply copying the data. In this way, SMOTE increases the amount of data from the underrepresented class without reducing the data from the majority class (Tarawneh et al., 2020; Santoso et al., 2017). This technique aims to improve the balance between classes in the training set, so that the model can be better trained to detect previously underrepresented minority classes (Xu et al., 2022).

2.3. Modeling

2.3.1. Random Forest

Random Forest is an ensemble model that combines the results of many decision trees to improve prediction accuracy and reduce overfitting. Each decision tree works independently, and the final result is an aggregation of the predictions of all trees. After all decision trees are built, Random Forest votes to determine the final class of the prediction. The class that gets the most votes from all decision trees will be the final prediction. If T is the number of trees in the forest, then the final prediction for the input x can be expressed as $y = argmax_k \sum_{j=1}^{T} I(h_j(x) = k)$ where $h_j(x)$ is the prediction of the j tree for input x, and i is an indicator function that has a value of 1 if the prediction is equal to class k and 0 otherwise (Nurbagja et al., 2023).

2.3.2. Support Vector Machine (SVM)

Support Vector Machine (SVM), works by finding a hyperplane that separates data classes with maximum margin, making it suitable for classification problems with high feature dimensions. The formulas Support Vector Machine (SVM) are as follows (Han et al., 2020).

$$\frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i$$

with the provision of:

$$y_i(w.x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

where:

w : weight vector
b : bias
C : regulatory parameters that control the trade-off between larger margins and misclassification
ξ_i : slack variables that allow for some misclassification
x_i : features of the training data
y_i : class labels from training data

2.3.3. Naive Bayes

Naive Bayes is a probabilistic model based on the principle of Bayes' theorem, assuming that features are independent of each other. Although this assumption is rarely true in reality, Naive Bayes often gives good results in text classification tasks.

$$Pr(c|d) \ \alpha \ \Pr(c)\prod_{w\in d}\Pr(w|c)$$

Where $Pr(c|d)$ posterior probability that a document d included in the class c, α shows that the probability $Pr(c|d)$ is comparable to the expression on the right. $Pr(c)$ shows how often class c appears in the dataset. $\prod_{w\in d}\Pr(w|c)$ the probability of word w appearing in document d for each word w present in the document (Abbas et al., 2019).

2.4. Model Evaluation

After the model is trained, it is evaluated using several metrics, including accuracy, precision, recall, and F1-Score. Accuracy measures how often the model makes correct predictions. Precision indicates the proportion of correct predictions of a particular class compared to the total predictions for that class. Recall measures how well the model finds all instances of a particular class. F1-Score is a compromise between precision and recall, providing a more balanced view of the model's performance when there is class imbalance. The formulas accuracy, precision, recall, and F1-Score are as follows (George and Srivindhya, 2022):

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{3}$$

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

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{6}$$

where:
TP : true positive
TN : true negative
FP : false positive
FN : false negative

3. Results and Discussion

3.1. Sentiment Distribution

The distribution of sentiment in Maxim app user reviews shows how reviews are divided into three categories: Negative, Neutral, and Positive. From the analysis of 1500 reviews, the majority are in the Positive category, followed by the Negative category, with the fewest Neutral reviews. Visualization in the form of a bar graph or pie chart helps to clearly illustrate the dominance of positive reviews, while neutral sentiment shows a relatively low number. This reflects the general perception of users towards the Maxim app, with a tendency to be more positive than negative or neutral.

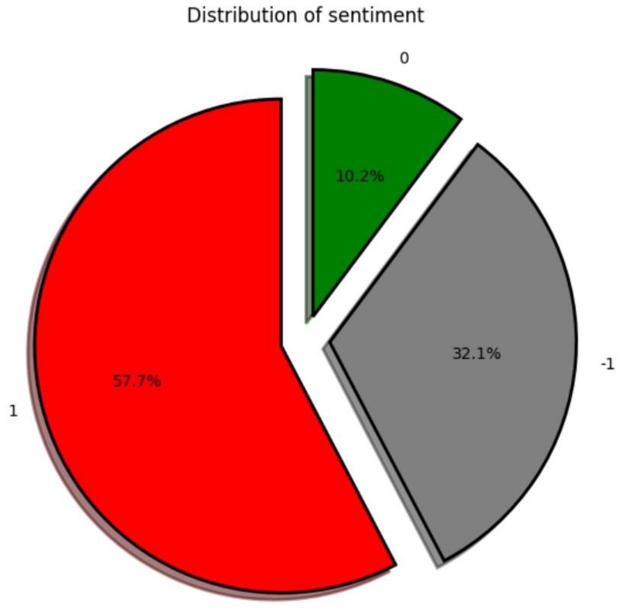

Figure 2: The distribution of sentiment in maxim app

3.2. App Rating

The rating of the Maxim app, based on the analyzed reviews, shows that on average users give this app a very good rating with many giving it 5 stars. The average rating of user reviews provides an overview of the quality and user satisfaction with the app. Although positive reviews dominate, the presence of negative reviews indicates some issues that need to be addressed by app developers to improve the overall user experience.

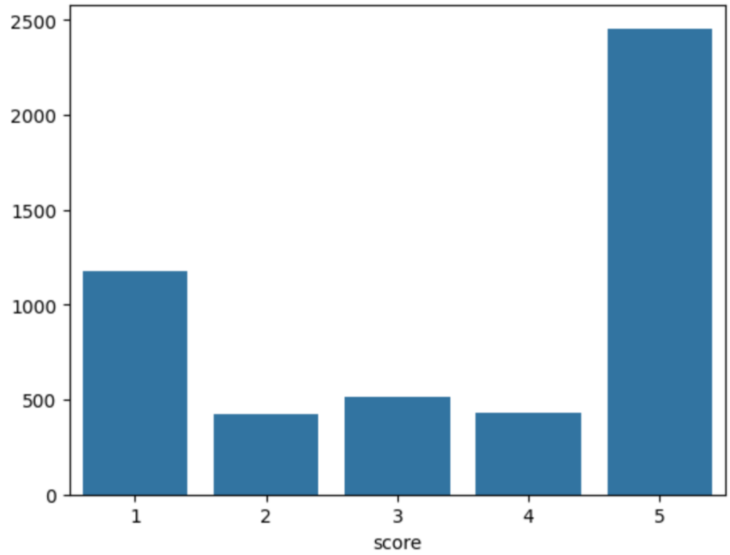

Figure 3: Rating reviews on the Maxim application

3.3. Word Cloud

The word cloud visualization shows the words that appear most often in user reviews (Ullah et al., 2019). Words such as "service," "driver," "price," and "time" were frequently mentioned in reviews, reflecting the main aspects that users care about. The frequency of these words provides important insights into the elements of the app that users discuss the most, both in a positive and negative context. This visualization helps understand the patterns and key topics highlighted in reviews.

Figure 4: Word Cloud

3.4. Model evaluation results

The results of the model evaluation including Random Forest, SVM, and Naive Bayes show that SVM provides the best performance with an accuracy of 91%. Followed by Random Forest with an accuracy of 89%, while Naive Bayes obtained an accuracy of 78%. However, each model shows different performance for each sentiment class.

3.5. Confusion matrix

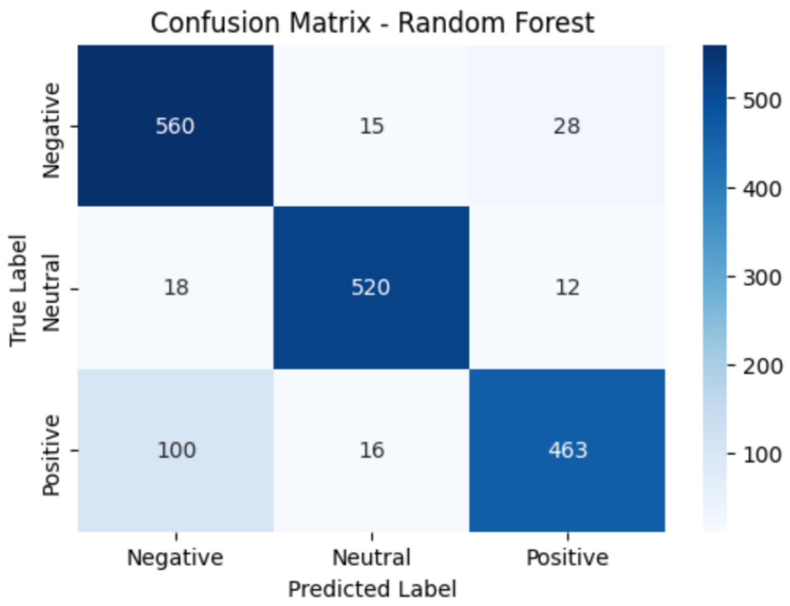

Figure 5: Confusion matrix Random Forest

The Random Forest model successfully identified 560 negative reviews, 520 neutral reviews, and 463 positive reviews, but struggled with 28 negative reviews being incorrectly classified as positive and 15 neutral reviews being incorrectly classified as negative. This suggests that while the model is effective, there is room for improvement in distinguishing between categories.

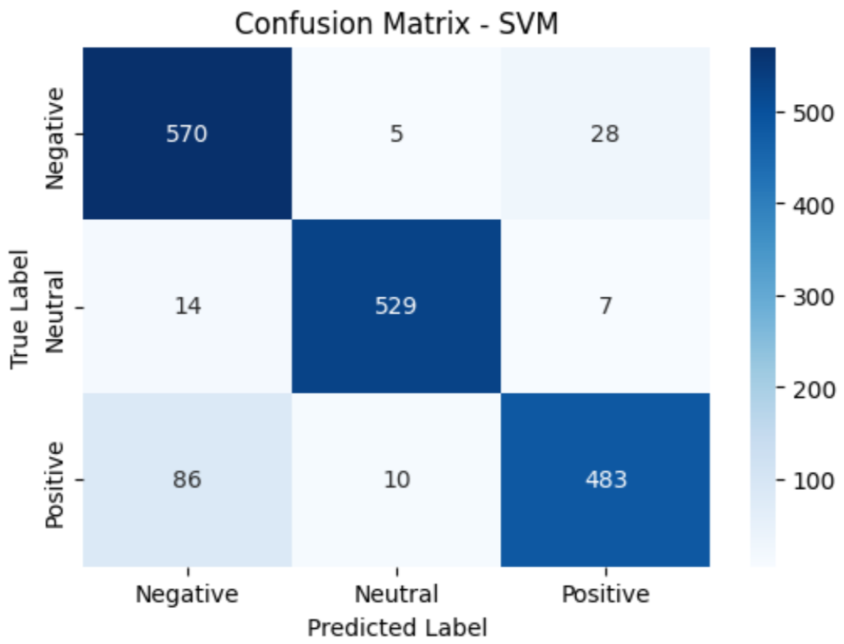

Figure 6: Confusion matrix SVM

This model shows quite good performance with a high level of accuracy, but there are still some prediction errors. For the Negative class, the model is able to correctly classify 570 samples out of 603 samples that are actually negative, but there are still 28 samples that are misclassified as Positive and 5 samples as Neutral. In the Neutral class, out of 550 neutral samples, 529 samples are correctly classified, while 14 samples are misclassified as Negative and 7 samples as Positive. For the Positive class, out of 579 samples that are actually positive, the model is able to correctly classify 483 samples, but there are significant errors where 86 samples are misclassified as Negative and 10 samples as Neutral. The largest prediction error occurs in the Positive class which is often misclassified as Negative.

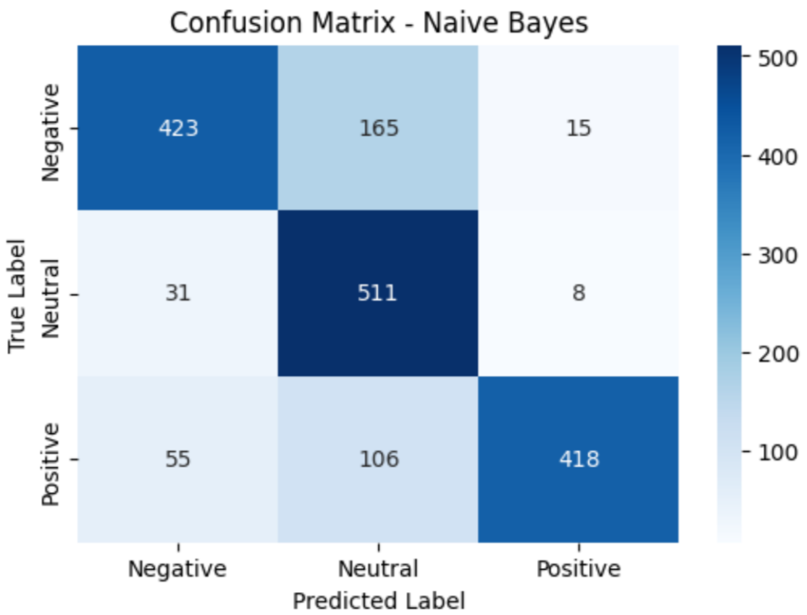

Figure 7: Confusion matrix Naive Bayes

In the Negative class, the model was able to correctly classify 423 samples out of 603 samples, but there were 165 samples that were misclassified as Neutral and 15 samples as Positive. The Neutral class performed better, where out of 550 samples, 511 samples were correctly classified, while 31 samples were misclassified as Negative and 8 samples as Positive. The largest error occurred in the Positive class, where out of 579 samples, only 418 were correctly classified, while 55 samples were misclassified as Negative and 106 samples were misclassified as Neutral. The largest error of the Naive Bayes model was seen in the difficulty of distinguishing between the Neutral and Positive classes, with many Positive samples being miscategorized as Neutral.

3.6. Model Performance Analysis

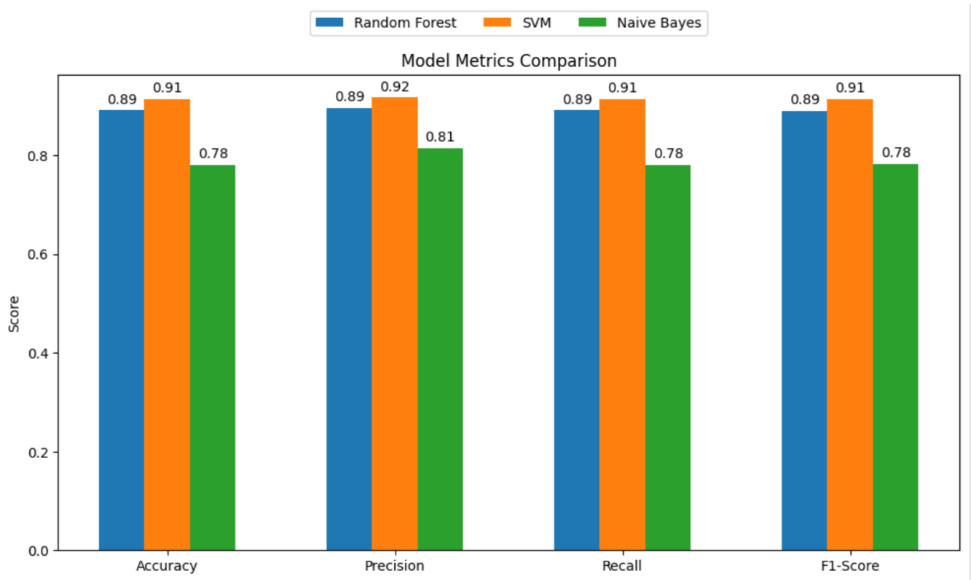

Figure 8: Model performance evaluation

Based on the results of the model performance evaluation, Random Forest showed an accuracy of 89%, with a precision of around 89.5%, a recall of 89%, and an F1-Score of 89%. This indicates that this model has a fairly good performance in classifying sentiment from user reviews. Although the error rate is relatively low, there is little difference between precision and recall, indicating that this model is quite balanced in detecting existing classes. Meanwhile, Support Vector Machine (SVM) recorded the highest accuracy, which was 91.3%, with a precision above 91.6%, and recall and F1-Score were also around 91.3%. This shows that SVM has very good performance compared to other models, and is able to separate sentiment classes accurately. Naive Bayes has an accuracy of 78%, which is lower than the other two models. Although the precision reaches 81%, the lower recall and F1-Score values indicate that this model tends to be less than optimal in handling complex sentiment class distributions.

4. Conclusion

The sentiment distribution shows a predominance of positive reviews and generally high app ratings, with many users giving 5 stars. A comparison of the performance of machine learning models revealed that Support Vector Machine (SVM) performed best in classifying sentiment, with an accuracy of 91.3%. Random Forest also performed well with an accuracy of 89%, while Naive Bayes, although with a lower accuracy (78%). Overall, this study not only provides a comprehensive picture of user perceptions of the Maxim app in Indonesia, but also demonstrates the effectiveness of machine learning techniques in sentiment analysis.

References

Abbas, M., Memon, K. A., Jamali, A. A., Memon, S., & Ahmed, A. (2019). Multinomial Naive Bayes classification model for sentiment analysis. *IJCSNS Int. J. Comput. Sci. Netw. Secur*, *19*(3), 62.

Aldabbas, H., Bajahzar, A., Alruily, M., Qureshi, A. A., Amir Latif, R. M., & Farhan, M. (2020). Google play content scraping and knowledge engineering using natural language processing techniques with the analysis of user reviews. *Journal of Intelligent Systems, 30*(1), 192-208.

Alim, N., Linda, W., Gunawan, F., & Saad, M. S. M. (2019). The effectiveness of Google classroom as an instructional media: A case of state islamic institute of Kendari, Indonesia. *Humanities & Social Sciences Reviews, 7*(2), 240-246.

Alyafeai, Z., Al-shaibani, M. S., Ghaleb, M., & Ahmad, I. (2023). Evaluating various tokenizers for Arabic text classification. *Neural Processing Letters, 55*(3), 2911-2933.

Fauzi, A. A., & Sheng, M. L. (2021). Ride-hailing apps' continuance intention among different consumer groups in Indonesia: the role of personal innovativeness and perceived utilitarian and hedonic value. *Asia Pacific Journal of Marketing and Logistics, 33*(5), 1195-1219.

George, S., & Srividhya, V. (2022). Performance evaluation of sentiment analysis on balanced and imbalanced dataset using ensemble approach. *Indian Journal of Science and Technology*, *15*(17), 790-797.

Han, K. X., Chien, W., Chiu, C. C., & Cheng, Y. T. (2020). Application of support vector machine (SVM) in the sentiment analysis of twitter dataset. *Applied Sciences*, *10*(3), 1125.

Kumar, J. A., & Bervell, B. (2019). Google Classroom for mobile learning in higher education: Modelling the initial perceptions of students. *Education and Information Technologies, 24*, 1793-1817.

Li, T., Higgins, J. P., & Deeks, J. J. (2019). Collecting data. *Cochrane handbook for systematic reviews of interventions,* 109-141.

Mansourifar, H., & Shi, W. (2020). Deep synthetic minority over-sampling technique. *arXiv preprint arXiv:2003.09788.*

Murshed, B. A. H., Mallappa, S., Ghaleb, O. A., & Al-ariki, H. D. E. (2021). Efficient twitter data cleansing model for data analysis of the pandemic tweets. *Emerging technologies during the era of COVID-19 pandemic*, 93-114.

Naeem, M. Z., Rustam, F., Mehmood, A., Ashraf, I., & Choi, G. S. (2022). Classification of movie reviews using term frequency-inverse document frequency and optimized machine learning algorithms. *PeerJ Computer Science, 8,* e914.

Nguyen, Q. H., Ly, H. B., Ho, L. S., Al-Ansari, N., Le, H. V., Tran, V. Q., ... & Pham, B. T. (2021). Influence of data splitting on performance of machine learning models in prediction of shear strength of soil. *Mathematical Problems in Engineering, 2021*(1), 4832864.

Nurbagja, K., Saputra, N., Riyadi, A., & Tentua, M. N. (2023). Sentiment Analysis of the Increase in Fuel Prices Using Random Forest Classifier Method. *Buletin Ilmiah Sarjana Teknik Elektro*, *5*(1), 132-144.

Nurhalifah, D., Musa, M. I., Dipoatmodjo, T. S., & Heruddin, M. I. W. (2023). The Influence of Service Quality on Customer Satisfaction With Maxim Services: Case Study of Management Students, Faculty of Economics and Business, Makassar State University. *Current Research on Practice Economics and Sharia Finance (CAPITAL), 1*(2), 27-38.

Panggarra, R., & Mustika, S. M. W. (2024). Comparison of Transportation Service Tariffs Between Maxim and Gojek Transportation Services (Study on Students, Faculty of Economics and Business, Victory University, Sorong). *Indonesian Journal of Contemporary Multidisciplinary Research, 3*(4), 729-740.

Pranckevičius, T., & Marcinkevičius, V. (2017). Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification. *Baltic Journal of Modern Computing, 5*(2), 221.

Rianto, Mutiara, A. B., Wibowo, E. P., & Santosa, P. I. (2021). Improving the accuracy of text classification using stemming method, a case of non-formal Indonesian conversation. *Journal of Big Data, 8*, 1-16.

Ririanti, N. P., & Purwinarko, A. (2021). Implementation of Support Vector Machine Algorithm with Correlation-Based Feature Selection and Term Frequency Inverse Document Frequency for Sentiment Analysis Review Hotel. Sci. J. *Informatics, 8*(2), 297-303.

Sadiq, S., Umer, M., Ullah, S., Mirjalili, S., Rupapara, V., & Nappi, M. (2021). Discrepancy detection between actual user reviews and numeric ratings of Google App store using deep learning. *Expert Systems with Applications, 181*, 115111.

Santoso, B., Wijayanto, H., Notodiputro, K. A., & Sartono, B. (2017, March). Synthetic over sampling methods for handling class imbalanced problems: A review. In *IOP conference series: earth and environmental science* (Vol. 58, No. 1, p. 012031). IOP Publishing.

Singh, J., & Gupta, V. (2017). A systematic review of text stemming techniques. *Artificial Intelligence Review, 48*, 157-217.

Solangi, Y. A., Solangi, Z. A., Aarain, S., Abro, A., Mallah, G. A., & Shah, A. (2018, November). Review on natural language processing (NLP) and its toolkits for opinion mining and sentiment analysis. In *2018 IEEE 5th international conference on engineering technologies and applied sciences (ICETAS)* (pp. 1-4). IEEE.

Song, X., Salcianu, A., Song, Y., Dopson, D., & Zhou, D. (2020). Fast wordpiece tokenization. *arXiv preprint arXiv:2012.15524.*

Subriadi, A. P., & Baturohmah, H. (2023). The Effect of Purchase Intention and Customer Behavior during the Covid-19 Pandemic on Ride-Hailing Application Loyalty. *International Journal on Advanced Science, Engineering & Information Technology, 13*(4).

Tarawneh, A. S., Hassanat, A. B., Almohammadi, K., Chetverikov, D., & Bellinger, C. (2020). Smotefuna: Synthetic minority over-sampling technique based on furthest neighbour algorithm. *IEEE Access, 8*, 59069-59082.

Tarawneh, A. S., Hassanat, A. B., Almohammadi, K., Chetverikov, D., & Bellinger, C. (2020). Smotefuna: Synthetic minority over-sampling technique based on furthest neighbour algorithm. *IEEE Access, 8*, 59069-59082.

Ullah, Z., Uzair, M., & Mahmood, A. (2019). Extraction of Key Motifs as a Preview from 2017 Nobel Prize Winning Novel,'Never Let Me Go': An Interactive Word Cloud Study. *Journal of Research in Social Sciences, 7*(2), 83-98.

Van der Loo, M., & De Jonge, E. (2018). *Statistical data cleaning with applications in R.* John Wiley & Sons.

Wulansari, A. D., Novianto, A., Keban, Y. T., & Hernawan, A. (2024). Hiding behind the platform: the myth of flexibility for gig workers in Indonesia. *South East Asia Research, 32*(1), 22-40.

Xu, G., Meng, Y., Qiu, X., Yu, Z., & Wu, X. (2019). Sentiment analysis of comment texts based on BiLSTM. *Ieee Access*, *7*, 51522-51532.

Xu, Y., & Goodacre, R. (2018). On splitting training and validation set: a comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. *Journal of analysis and testing, 2*(3), 249-262.

Xu, Z., Shen, D., Kou, Y., & Nie, T. (2022). A synthetic minority oversampling technique based on Gaussian mixture model filtering for imbalanced data classification. *IEEE Transactions on Neural Networks and Learning Systems, 35*(3), 3740-3753.