**Jurnal Teknologi Sistem Informasi dan Aplikasi** **ISSN: 2654-3788**
**Penerbit: Program Studi Teknik Informatika Universitas Pamulang** **e-ISSN: 2654-4229**
**Vol. 7, No. 4, Oktober 2024 (1603-1610)** **DOI: 10.32493/jtsi.v7i4.45117**

# Enhancing Usability Testing Through Sentiment Analysis: A Comparative Study Using SVM, Naive Bayes, Decision Trees and Random Forest

**Hasan Basri[1], Mochamad Bagoes Satria Junianto[2], Irpan Kusyadi[3]**

Department of Information System, Faculty of Science and Technology, Open University, Indonesia
e-mail: hasan.basri@ecampus.ut.ac.id[1], mochamad.bagoes@ecampus.ut.ac.id[2],
irpan.kusyadi@ecampus.ut.ac.id[3]

Submitted Date: October 02nd, 2024 Reviewed Date: October 28th, 2024
Revised Date: October 31st, 2024 Accepted Date: October 31st, 2024

**Abstract**

In the digital age, mobile applications have become an integral part of everyday life, making usability testing an essential factor in ensuring a seamless user experience. Traditional usability testing methods often demand considerable resources, including time and cost, which calls for more efficient and automated alternatives. This study explores the use of sentiment analysis as an innovative approach to evaluate the usability of mobile applications. By analyzing user reviews from the Google Play Store, the research compares the effectiveness of four machine learning algorithms—Support Vector Machine (SVM), Naive Bayes, Decision Tree, and Random Forest—in classifying sentiment and evaluating usability. A dataset consisting of 2,000 reviews from a banking app was collected and labeled based on usability criteria, such as efficiency, user satisfaction, learnability, memorability, and error rates. The feature extraction process utilized Term Frequency-Inverse Document Frequency (TF-IDF) to enhance the relevance of the review texts for sentiment analysis. The findings reveal that Random Forest achieved the highest accuracy (68.15%) and demonstrated the best performance in terms of F1 Score, precision, and recall, although it had the longest processing time. In contrast, Naive Bayes, while the fastest, showed lower accuracy and F1 Score, making it suitable for applications with large datasets or limited processing time. Decision Tree and SVM offered a balanced trade-off between speed and accuracy. The study concludes that Random Forest is the preferred choice when high accuracy and prediction performance are crucial, despite its longer processing time. Meanwhile, Naive Bayes is more appropriate for scenarios demanding rapid data processing, and SVM and Decision Tree are recommended when a balance between speed and accuracy is needed.

Keywords: Usability Testing; Sentiment Analysis; ML4SE; Model Comparison

## 1 Introduction

In today's digital age, mobile applications have become an integral part of everyday life. Therefore, evaluating the quality and ease of use (usability) of an application becomes very important to ensure an optimal user experience (Huang & Benyoucef, 2023). Usability testing is a process used to evaluate how easily and effectively an application can be used by end users. The results of usability testing not only provide insight into how users interact with the application, but also assist developers in improving the quality of their products. A good evaluation can improve user retention, minimize churn rates, and increase overall user satisfaction (Hajesmaeel-Gohari et al., 2022).

However, conventional usability testing methods often require large resources, both in terms of cost and time. The process typically involves collecting data through direct observation, interviews, and user surveys, all of which can be costly and time-consuming, especially if conducted on a large scale (Weichbroth, 2024). These challenges trigger the need for a more efficient and automated approach to evaluating the usability of mobile applications.

Sentiment analysis is emerging as one of the potential solutions in the context of Machine Learning for Software Engineering (ML4SE) in

ISSN: 2654-3788
e-ISSN: 2654-4229
DOI: 10.32493/jtsi.v7i4.45117

software engineering evaluation. By analyzing user reviews available on platforms such as the Google Play Store, sentiment analysis can provide valuable insights into how users experience and feel about apps. This technique uses Natural Language Processing (NLP) to identify and classify user opinions into specific sentiment categories, such as positive, negative or neutral. As such, ML4SE enables faster and more automated evaluation, and provides real-time data that can be immediately used for app improvement.

Various machine learning methods have been developed to improve the accuracy and effectiveness of sentiment analysis (BAYAT & IŞIK, 2023). Each classification method, such as Support Vector Machine (SVM), Naive Bayes, Decision Trees, and Random Forest, has its own advantages and disadvantages. Therefore, it is important to conduct a comparison between these methods to determine which one is most effective in the context of sentiment analysis-based usability testing. This study aims to evaluate and compare the performance of various classification techniques in analyzing sentiment from user reviews, in order to find the most optimal method for improving usability evaluation of mobile applications.

## 2 Research Method

The following is a research methodology diagram that describes the steps that will be taken in this research, can be seen in the figure 1. The first step in this research is to collect mobile app user review data from sources such as Google Play Store. These reviews will be the dataset analyzed to evaluate the usability of the app. The data pre-processing stage involves labeling and text processing. The labeling process includes analyzing the usability testing criteria and labeling the data according to those criteria (positive, negative). Next, the review text is processed through tokenization, stopword removal, lemmatization, and normalization to ensure consistency.

After the data is processed, important features are extracted from the text for use in machine learning models. Techniques such as TF-IDF are used for text feature representation. The processed data is then divided into two sets: training data (80%) and test data (20%). Several machine learning models such as SVM, Naive Bayes, Decision Trees, and Random Forest will be trained using the training data. Each model is trained and tested separately, and evaluation metrics are recorded for each model.

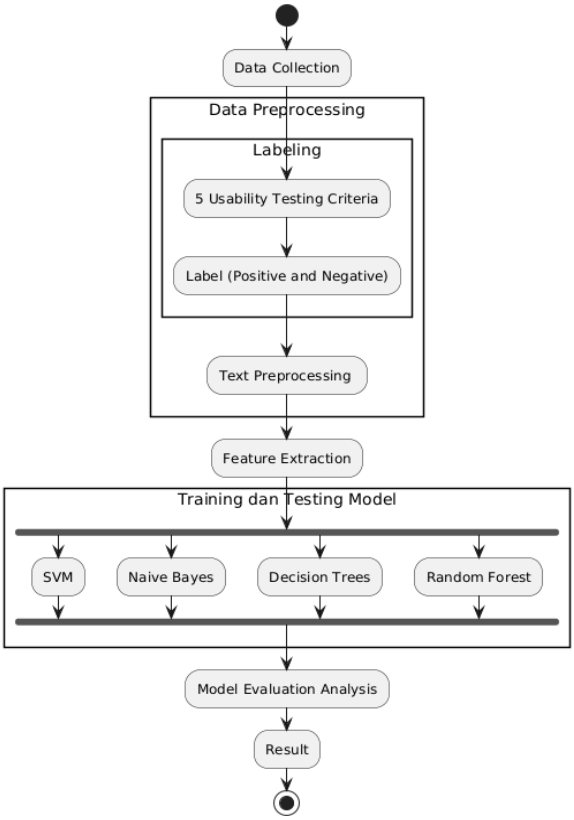


Figure 1. Research Procedure

The evaluation results of each model are compared to determine which one is most effective. The metrics used include Accuracy, Precision, Recall, F1-Score, and Processing time. This step is important to understand how good each model is at classifying sentiment from user reviews. A good evaluation can help developers improve app quality and overall user experience.

## 3 Results and Discussion

### 3.1 Data Collection and Data Preparation

This study uses review data from one banking company's app available on the Google Play Store, with a total of around 1.5 million reviews. This huge number of reviews provides significant potential to be used in research related to sentiment analysis and usability testing (Sarker & Roy, 2020). This review data reflects users' opinions on their experience using banking apps, specifically in terms of Efficiency, User

Satisfaction, Learnability, Memorability, and Error Rate. A total of 2,000 reviews were taken from the entire dataset to be further analyzed based on Usability Testing criteria. The labeling process is done by referring to the five main criteria of usability testing based on relevant literature (Jakob Nielsen, 2012).The five main aspects include Efficiency, User Satisfaction, Learnability, Memorability, and Error Rate, Labeling is also done with reference to sentiment analysis criteria.

A multi-label approach in the process of labeling review data is used, involving two annotators working collaboratively (Elghannam, 2023). The first annotator (called Annotator 'A') is responsible for labeling the reviews based on sentiment analysis with two categories, namely Positive and Negative. Once the sentiment labels are determined, the second Annotator (Annotator 'B') labels them based on Usability Testing criteria. In addition, Annotator 'B' also checks the sentiment labels that have been given by Annotator 'A' to ensure there are no errors in the labeling process. As a final step, Annotator 'A' again checks the labels that have been given by Annotator 'B' on the Usability Testing criteria to ensure the consistency and accuracy of the labeling results. After labeling and data cleaning, 6 classes are obtained for the classification process in the next classification method comparison stage.

### 3.2 Feature Extraction

TF-IDF (Term Frequency-Inverse Document Frequency) was chosen as the feature extraction method in this study due to its ability to highlight relevant words in the review text. This method works by giving more weight to words that appear frequently in a review (Setiawan et al., 2022). In this way, TF-IDF can capture terms that have meaning specific to the review context, such as words that reflect user satisfaction or app efficiency. This is crucial to ensure that the generated features are relevant to the tasks of sentiment analysis and usability testing.

The main advantage of TF-IDF is its ability to reduce the influence of common words that often appear in most reviews but have no informative value, such as “and”, “in”, or “is”. By giving these words a low weight, TF-IDF helps classification algorithms focus on more significant words, such as “slow”, “easy to use”, or “dissatisfied”. This makes TF-IDF particularly suitable for analyzing Google Play Store review datasets that vary in text length and context.With this capability, TF-IDF becomes an effective solution to capture the relationship between review texts and labels, both in sentiment analysis and usability testing.

### 3.3 Model Comparison

The evaluation results of each model were compared to determine which was most effective.

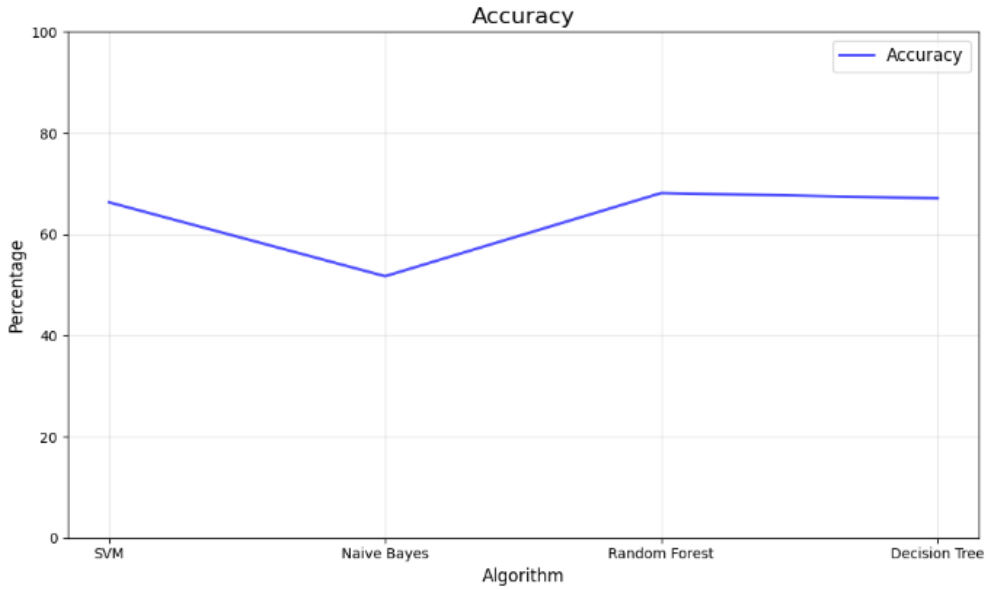


Figure 2. Accuracy

The Figure 2 above shows the accuracy comparison of the four machine learning algorithms applied in this study, namely Support Vector Machine (SVM), Naive Bayes, Random Forest, and Decision Tree. Based on the analysis results, the Random Forest algorithm shows the highest accuracy of 68.15%. This shows that Random Forest is able to classify data better than other algorithms in the context of this research.

The Decision Tree algorithm ranked second with an accuracy of 67.1%, which is only slightly lower than Random Forest. This result indicates that Decision Tree is also quite effective in processing data despite having a slight disadvantage compared to Random Forest, especially since Random Forest is a development of the Decision Tree method.

Meanwhile, SVM achieved an accuracy of 66.32%. Although not as good as Random Forest and Decision Tree, SVM is still able to provide competitive results. However, the Naive Bayes algorithm showed the lowest performance with an accuracy of 51.7%, which may be due to the assumption of independence between features that was not fully met in the dataset.

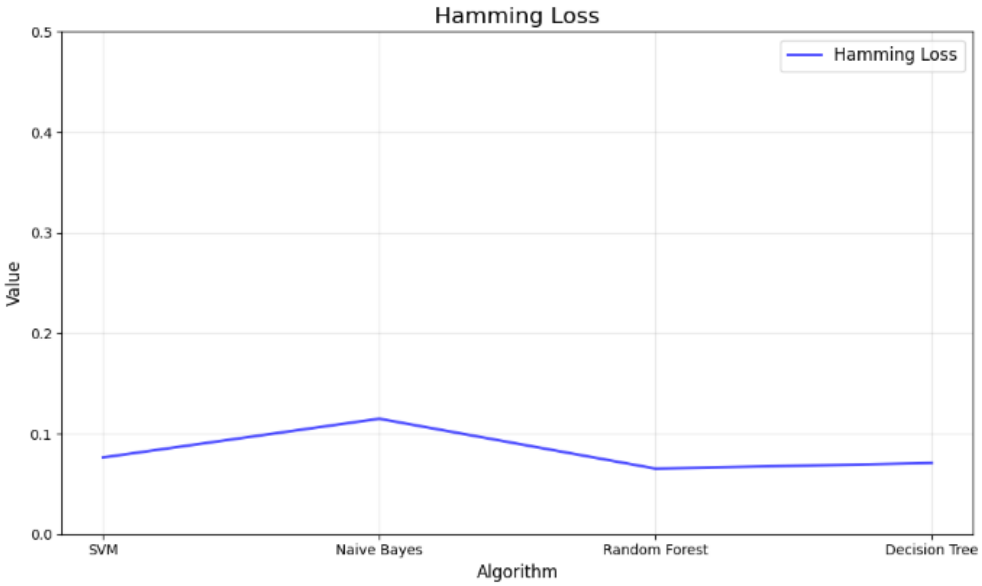


Figure 3. Hamming Loss

The Figure 3 above illustrates the comparison of Hamming Loss values of the four machine learning algorithms used, namely Support Vector Machine (SVM), Naive Bayes, Random Forest, and Decision Tree. Hamming Loss is used as an evaluation metric to measure classification error in the context of multilabel data.

The analysis shows that the Random Forest algorithm has the lowest Hamming Loss of 0.0653. This value reflects that Random Forest has the smallest misclassification rate compared to other algorithms, so it can be considered as the most reliable algorithm for the data in this study.

On the other hand, the Decision Tree algorithm performed quite well with a Hamming Loss value close to Random Forest. SVM showed a slightly lower performance, while Naive Bayes recorded the highest Hamming Loss. The high Hamming Loss value of Naive Bayes indicates that this model has a larger misclassification error, possibly due to the assumption of independence between features that is not suitable for this dataset.

Random Forest stands out as the best algorithm in reducing classification error, with Decision Tree as a moderately competitive alternative. Meanwhile, SVM and Naive Bayes may need additional parameter or feature adjustments to improve their performance.

ISSN: 2654-3788
e-ISSN: 2654-4229
DOI: 10.32493/jtsi.v7i4.45117

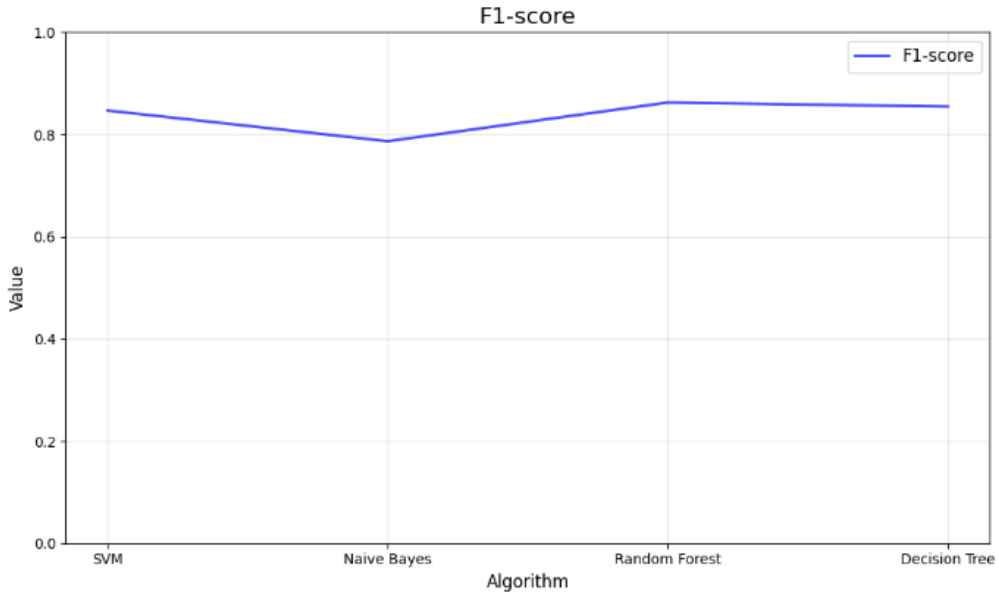


Figure 4. F1-Score

In addition to accuracy and Hamming Loss analysis, F1 Score is used to evaluate the balance between precision and recall of four machine learning algorithms, namely Support Vector Machine (SVM), Naive Bayes, Random Forest, and Decision Tree.

The analysis results show that Random Forest has the highest F1 Score of 0.8624. This value indicates that Random Forest is able to provide the best balance between precision (the ability of the model to minimize false positives) and recall (the ability of the model to capture all relevant instances), making it the most effective algorithm in this study.

Decision Tree came in second with a slightly lower F1 Score, followed by SVM, which showed a fairly competitive performance. Naive Bayes, once again, took the lowest position in terms of F1 Score, indicating that this model is less than optimal in balancing precision and recall, most likely due to the assumption of independence between features not being fully met on the dataset.

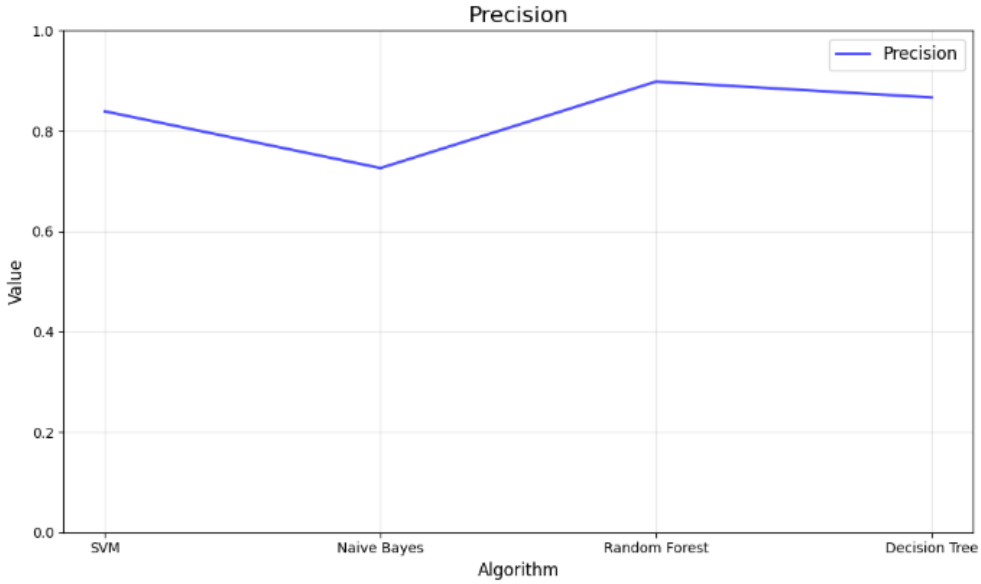


Figure 5. Precision

ISSN: 2654-3788
e-ISSN: 2654-4229
DOI: 10.32493/jtsi.v7i4.45117

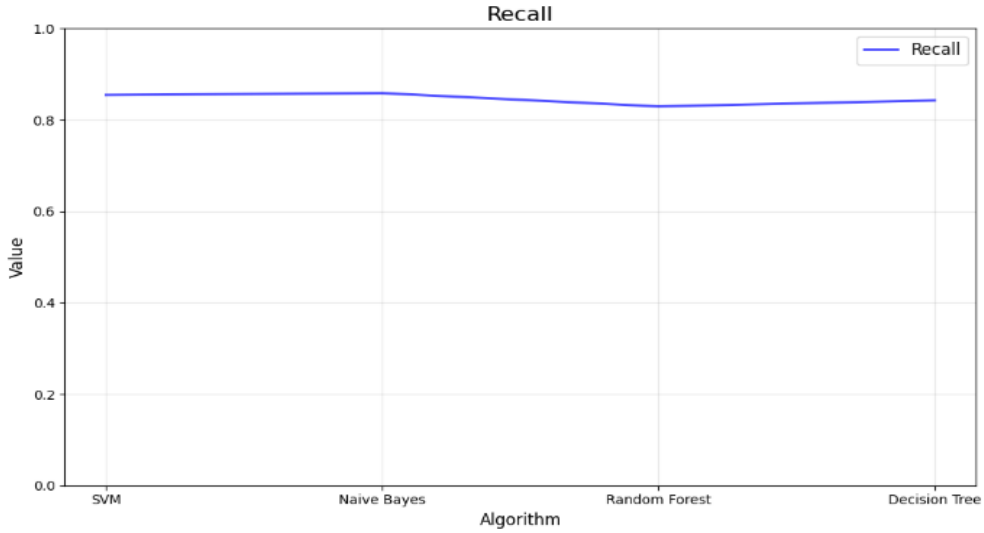


Figure 6. Recall

Further evaluation is done using precision and recall metrics to understand the specific performance of each machine learning algorithm. Precision measures the ability of the model to avoid false positive predictions, while recall measures the extent to which the model can capture relevant positive instances (true positives). The analysis results show that Random Forest has the highest precision value of 0.8983. This reflects that Random Forest is able to minimize false positive predictions very well, making it the most reliable algorithm in producing accurate predictions.

In contrast, the Naive Bayes algorithm recorded the highest recall value of 0.8585, which demonstrates its ability to capture most of the relevant positive instances. However, although the recall of Naive Bayes is high, it is not in line with its precision value, so it tends to produce more false positives than other algorithms. Decision Tree and SVM showed a good balance between precision and recall, but still underperformed Random Forest for both metrics.

In addition to evaluating performance based on accuracy, F1 Score, precision, and recall, this study also analyzed the processing time of each machine learning model.

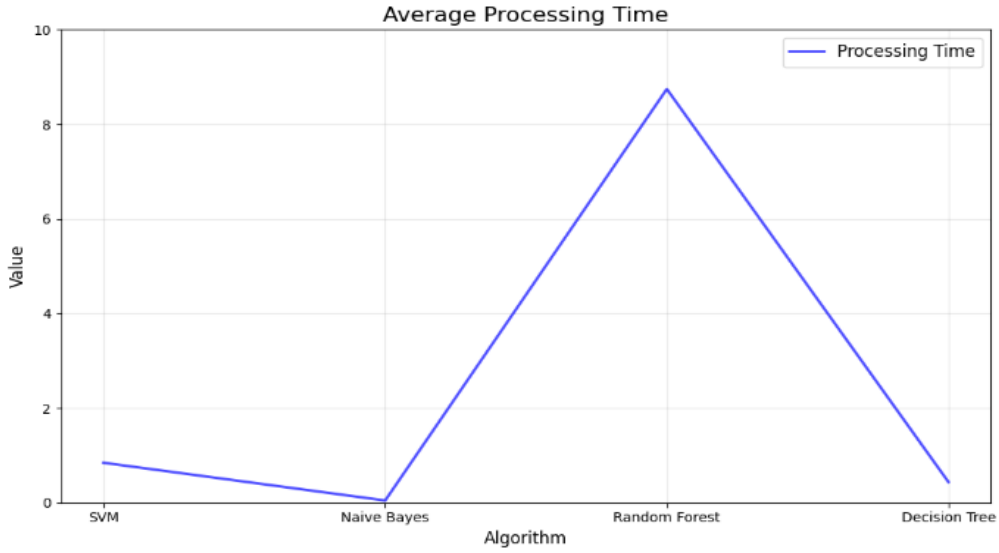


Figure 7. Average Processing Time

Processing time evaluation was conducted to compare the efficiency of the four machine learning algorithms used, namely Naive Bayes, Support Vector Machine (SVM), Random Forest, and Decision Tree. The processing time is calculated as the average time required for each algorithm to process the dataset after 10 trials. The analysis results show that Naive Bayes has the lowest average processing time, which is 0.04528 seconds. This makes Naive Bayes the most efficient model in terms of processing time, mainly because this algorithm has a simple and lightweight computational structure.

Decision Tree also shows good time efficiency with an average processing time of 0.43595 seconds. This efficiency is due to the relatively simple algorithm structure in creating the decision tree. SVM requires an average processing time of 0.84302 seconds, slightly longer than Decision Tree. This can be attributed to the kernel optimization process in SVM, which requires more computation. Random Forest has the highest average processing time of 8.74353 seconds. The high processing time is due to the complexity of the algorithm which involves merging the results of many decision trees. Although it takes longer, Random Forest provides the best performance in terms of accuracy, F1 Score, and precision.

This analysis shows a trade-off between performance and processing time efficiency. Naive Bayes and Decision Tree are the best choices for fast computing requirements, while Random Forest is more suitable when prediction performance is a top priority despite requiring longer processing time. SVM, with its moderate processing time, can be an alternative choice when a balance between time and performance is required.

## 4 Conclusion

Based on the analysis conducted, it can be concluded that Random Forest is the most accurate model, with the highest F1 Score and the best performance in terms of precision and recall. However, this model requires longer processing time than other algorithms. Therefore, Random Forest is suitable for use in situations where accuracy and prediction performance are top priorities, although there is a trade-off in processing time. Meanwhile, Naive Bayes is the most efficient model in terms of processing time, with the lowest average time. However, Naive Bayes has lower accuracy and F1 Score than other models. This model is more suitable for applications that require speed in data processing, such as for very large datasets or in conditions with tight time constraints. SVM and Decision Tree offer a good balance between performance and processing time. Both are more efficient than Random Forest, with a more moderate processing time. SVM provides a balance between accuracy and efficiency, while Decision Tree is simpler and more time efficient.

Recommendations on the use of models can be tailored to the specific needs of the application. If accuracy and F1 Score are the top priorities in the application, then Random Forest is the best choice even though it requires longer processing time. However, if time efficiency is more important and facing computational limitations or having to handle large amounts of data, then Naive Bayes could be more suitable. For applications that require a balance between speed and accuracy, SVM and Decision Tree can be good alternatives.

## References

Bayat, S., & IŞIK, G. (2023). Evaluating the Effectiveness of Different Machine Learning Approaches for Sentiment Classification. *Iğdır Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, *13*(3), 1496–1510. https://doi.org/10.21597/jist.1292050

Elghannam, F. (2023). Multi-Label Annotation and Classification of Arabic Texts Based on Extracted Seed Keyphrases and Bi-Gram Alphabet Feed Forward Neural Networks Model. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *22*(1), 1–16. https://doi.org/10.1145/3539607

Hajesmaeel-Gohari, S., Khordastan, F., Fatehi, F., Samzadeh, H., & Bahaadinbeigy, K. (2022). The most used questionnaires for evaluating satisfaction, usability, acceptance, and quality outcomes of mobile health. *BMC Medical Informatics and Decision Making*, *22*(1), 22. https://doi.org/10.1186/s12911-022-01764-2

Huang, Z., & Benyoucef, M. (2023). A systematic literature review of mobile application usability: addressing the design perspective. *Universal Access in the Information Society*, *22*(3), 715–735. https://doi.org/10.1007/s10209-022-00903-w

Jakob Nielsen. (2012). *Usability 101: Introduction to Usability*. https://www.nngroup.com/articles/usability-101-introduction-to-usability/

Sarker, P., & Roy, S. (2020). Multi-Classifier based Sentiment Analysis for Opinionated Data Posted in Social Networking. *International Journal of Computer Science and Mobile Computing*, *9*(12), 68–75. https://doi.org/10.47760/ijcsmc.2020.v09i12.009

Setiawan, Y., Gunawan, D., & Efendi, R. (2022). Feature Extraction TF-IDF to Perform Cyberbullying Text Classification: A Literature Review and Future Research Direction. *2022 International Conference on Information Technology Systems and Innovation (ICITSI)*, 283–288. https://doi.org/10.1109/ICITSI56531.2022.9970942

Weichbroth, P. (2024). Usability Testing of Mobile Applications: A Methodological Framework. *Applied Sciences*, *14*(5), 1792. https://doi.org/10.3390/app14051792