Journal of Deep Learning, Computer Vision and Digital Image Processing
E-ISSN: 2986-8939 P-ISSN: 2986-8920
Vol. 4, No. 1, March 2026
DOI. https://doi.org/10.61255/decoding.v4i1.948

Sentiment Analysis of Gojek Driver Application Reviews Using Support Vector Machine and Naïve Bayes with Optuna-Based Hyperparameter Tuning

Nadilla Madjid
Universitas Trilogi, Jakarta, Indonesia

Rudi Setiawan✉
Universitas Trilogi, Jakarta, Indonesia

ARTICLE HISTORY
Received: 6 February, 2026
Revised: 3 March, 2026
Accepted: 27 March, 2026

KEYWORDS
Hyperparameter Tuning;
Naïve Bayes;
Optuna;
Sentiment Analysis;
Support Vector Machine

ABSTRACT

Purpose – This study aims to analyze user review sentiment toward the Gojek Driver application and compare the performance of two classification algorithms, Support Vector Machine (SVM) and Naïve Bayes, using Optuna as a framework for hyperparameter tuning.
Methods – The study collected and labeled user review data into positive and negative sentiment categories. Text preprocessing involved cleaning, case folding, normalization, tokenization, stopword removal, and stemming. Features were represented using TF-IDF. The dataset was then divided into training and testing sets, and SVM and Naïve Bayes models were trained using automated hyperparameter optimization with Optuna. Model performance was evaluated using accuracy, precision, recall, F1-score, and confusion matrix.
Findings – The application of SMOTE to the Optuna-tuned SVM model produced better performance than the other models tested in this study. The best model achieved an accuracy of 0.868, a highest cross-validation accuracy of 92.72%, and a weighted average F1-score of 0.87. These results indicate that SVM was more effective in handling high-dimensional TF-IDF features and complex decision boundaries.
Research implications – The findings support the use of automated sentiment analysis to assist operational decision-making and improve the quality of Gojek Driver services. The proposed approach can accelerate the identification of service-related issues and provide a basis for proactive responses to user feedback.
Originality – This study offers an original contribution by directly comparing SVM and Naïve Bayes on a Gojek Driver review dataset while applying Optuna-based hyperparameter tuning. It highlights the effect of automated tuning on both algorithms within a TF-IDF representation framework for ride-hailing service data, a topic that remains underexplored in the specific context of Gojek Driver within the local literature.

Correspondence Author: ✉rudi@trilogi.ac.id
To cite this article : N. Madjid & R. Setiawan. (2026). Sentiment Analysis of Gojek Driver Application Reviews Using Support Vector Machine and Naïve Bayes with Optuna-Based Hyperparameter Tuning. *Journal of Deep Learning, Computer Vision and Digital Image Processing*, *4*(1), 1-14. https://doi.org/10.61255/decoding.v4i1.948

INTRODUCTION

Digital service applications such as Gojek Driver have become an important part of everyday life in Indonesia, particularly in supporting mobility, platform-based work, and service interaction between drivers and users. As a large-scale digital platform, Gojek Driver receives substantial user feedback through platforms such as the Google Play Store. These reviews provide valuable information about service quality, user experience, technical problems, and perceived platform performance[1], [2], [3], [4]. However, manually evaluating large volumes of user reviews is inefficient and difficult to sustain, especially when the data continue to grow over time [5], [6]. Automated sentiment analysis is therefore necessary to classify user opinions into positive and negative sentiment categories and to support service quality improvement in digital platform ecosystems.

Previous studies have applied various machine learning algorithms for sentiment analysis in digital application contexts. Naïve Bayes has been widely used in sentiment classification because of its simplicity and efficiency [7], [8], [9]. Decision Tree algorithms have also been applied in several sentiment analysis studies involving user feedback and application reviews [10], [11], [12]. Other studies have employed K-Nearest Neighbor to classify sentiment patterns in user-generated textual data [13], [14], [15], while Support Vector Machine has frequently been used because of its strong performance in text classification tasks [16], [17], [18]. A comparative study on sentiment classification for the Satu Sehat application reported that SVM achieved higher accuracy, at 87.95%, than Naïve Bayes, which reached 81.65% [19]. This finding suggests that SVM can perform better in sentiment classification, particularly when dealing with imbalanced data, which remains one of the main challenges in sentiment analysis.

Although prior studies have contributed to the development of sentiment analysis for digital applications, several methodological limitations remain. Many existing studies have not fully integrated lexicon-based automatic labeling, imbalance handling through the Synthetic Minority Over-sampling Technique, and automated hyperparameter optimization using frameworks such as Optuna. These components are important because sentiment datasets from application reviews often contain unequal class distributions, noisy informal language, and high-dimensional textual features. Without proper imbalance handling and parameter optimization, classification models may produce biased performance, particularly toward the majority class. This issue is especially relevant in the context of user reviews, where negative and positive sentiment may not be evenly distributed.

In this context, the Gojek Driver application provides a relevant case for sentiment analysis. The application has a large and diverse user base, and its review data reflect various user perceptions of platform reliability, service quality, technical functionality, and operational experience. This study analyzes user sentiment toward the Gojek Driver application by comparing two widely used classification algorithms, namely Naïve Bayes and Support Vector Machine. Naïve Bayes was selected because of its ability to perform classification efficiently even with limited data [20], [21]. SVM was selected because of its capability to construct strong decision margins and its suitability for imbalanced text classification problems [22], [23], [24]. Previous studies also indicate that SVM can achieve strong performance when combined with SMOTE for imbalance handling [25], [26].

This study further integrates Optuna-based hyperparameter tuning to automatically optimize model parameters and improve classification performance. By combining SMOTE and stratified train-test splitting, this study examines whether the proposed methodological configuration can produce more accurate sentiment classification models than approaches without optimization. The main contribution of this study lies in its comparative evaluation of Naïve Bayes and SVM for Gojek Driver review sentiment analysis, with attention to class imbalance handling and automated hyperparameter optimization.

This study seeks to address the existing research gap by offering an optimized sentiment analysis approach for digital application reviews, particularly in the case of Gojek Driver. The findings are expected to contribute academically to the development of machine learning-based sentiment analysis in Indonesian digital service contexts. Practically, the results may assist platform managers and service developers in identifying user concerns more efficiently, monitoring service quality, and responding proactively to user feedback.

METHOD

Data Collection

The data used in this study were obtained from user reviews of the Gojek Driver application published on the Google Play Store. Data collection was conducted in February 2025 using the Python library google-play-scraper, which enables the extraction of review data based on the application ID. The application ID used in this study was com.gojek.partner. A total of 14,990 user reviews were collected and used as the dataset for sentiment classification.

Text Preprocessing

Text preprocessing was conducted to prepare the review data for sentiment labeling and model development. The preprocessing process began with case folding, in which all text was converted into lowercase to ensure consistency. The data were then cleaned by removing URLs, numbers, HTML tags, emojis, symbols, and non-alphabetic characters. After the cleaning stage, informal words and abbreviations were normalized into standard forms using a lexicon-based normalization dictionary. Tokenization was then applied to split each review into individual word tokens. Common words that did not carry substantial sentiment meaning were removed through stopword removal, while stemming was conducted using Sastrawi to reduce Indonesian words to their root forms.

After preprocessing, the textual data were transformed into numerical features using Term Frequency-Inverse Document Frequency (TF-IDF). To prevent data leakage, the TF-IDF representation was fitted only on the training data and then applied to the corresponding validation or testing data.

Lexicon-Based Sentiment Labeling

Sentiment labeling was performed using a lexicon-based approach. Each token in the review data was mapped to the Indonesian sentiment lexicon, Lexicon InSet, which contains word polarity scores. The sentiment score of each review was calculated by summing the polarity scores of all tokens in the review. Reviews with a total sentiment score greater than or equal to zero were categorized as positive, while reviews with a total sentiment score below zero were categorized as negative.

After the labeling process, the dataset consisted of 4,825 positive reviews and 10,165 negative reviews, with a total of 14,990 labeled reviews. This distribution indicates a class imbalance, as negative reviews constituted the majority class.

Data Splitting and Class Imbalance Handling

The dataset was divided into training and testing sets using an 80:20 split. To examine the effect of class distribution control during data partitioning, this study evaluated two experimental scenarios. The first scenario used a non-stratified train-test split, while the second scenario used a stratified train-test split to preserve the original class distribution in both the training and testing sets. A random state value of 42 was applied to ensure reproducibility.

To address class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was applied only to the training data. SMOTE was not applied to the testing data because the testing set must retain the original data distribution to provide an unbiased evaluation of model performance. During cross-validation, Stratified K-Fold with 10 folds was applied to the training data. In each fold, TF-IDF was fitted only on the training portion, and SMOTE was applied only to the training fold. This procedure was used to minimize the risk of data leakage during model validation..

Hyperparameter Optimization Using Optuna

Hyperparameter optimization was conducted using Optuna to improve the performance of the classification models. The objective function was based on accuracy, calculated as the average score across all cross-validation folds on the training data. For the SVM model, the search space included the regularization parameter C from 1e-3 to 1e2 using a log-uniform distribution, the kernel type consisting of linear and rbf, and gamma from 1e-4 to 1e1 using a log-uniform distribution. The gamma parameter was evaluated only when a non-linear kernel was selected because gamma does not affect the decision boundary of a linear kernel. For the Naïve Bayes model, the search space included the alpha parameter from 1e-3 to 10 using a log-uniform distribution. The complete hyperparameter search space is shown in Table 1.

Table 1. Hyperparameter Search Space

Algorithm	Hyperparameter Search Space
SVM	C: 1e-3 to 1e2, log-uniform; kernel: linear and rbf; gamma: 1e-4 to 1e1, log-uniform
Naïve Bayes	alpha: 1e-3 to 10, log-uniform

Optuna performed 50 trials to identify the best hyperparameter combination for each model. The objective function was evaluated based on the accuracy score obtained during cross-validation. After the best hyperparameters were identified, the selected model was retrained on the training data and then evaluated on the testing data.

Model Evaluation

Model evaluation was conducted on the testing data that had been separated before model training. The evaluation metrics included accuracy, precision, recall, F1-score, and confusion matrix. Accuracy was used to measure the overall proportion of correct predictions. Precision was used to evaluate the proportion of correctly predicted sentiment labels among all predicted labels for each class. Recall was used to assess the model's ability to identify actual class instances. F1-score was used to balance precision and recall, particularly under imbalanced class conditions. The confusion matrix was used to provide a more detailed view of correct and incorrect predictions for each sentiment class.

RESULTS AND DISCUSSION

Results

Training and Evaluation Results of SVM Without Stratified Train-Test Split

Table 2 presents the hyperparameter tuning results of the Support Vector Machine (SVM) model using Optuna without applying the stratified train-test split technique. The best parameters obtained were a C value of 9.9034, a linear kernel, and gamma set to scale. It should be noted that, for a linear kernel, the decision hyperplane is computed based on the linear dot product between features. Therefore, the gamma parameter does not affect the decision boundary when the selected kernel is linear. In this configuration, the model achieved the highest cross-validation accuracy of 0.9272, indicating strong validation performance.

Table 2. Hyperparameter Tuning Results of the SVM Model Using Optuna

Parameter	Value
C	*9.9034*
Kernel	*linear*
Gamma	*scale*
Highest CV Accuracy	*0.9272*

Figure 3 shows that the SVM training accuracy remained consistently high and approached 100%, while the validation accuracy increased as the amount of training data expanded. The validation accuracy reached 92.72%, suggesting that the model had good generalization ability. The learning curve indicates that the model was able to learn the training data effectively while still maintaining stable validation performance.

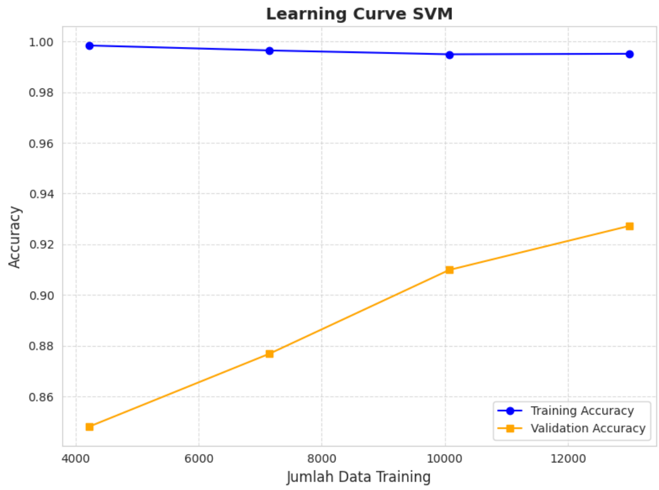


Figure 3. Learning Curve of the SVM Model Without Stratified Train-Test Split

Figure 4 presents the validation curve for the C parameter. The curve shows that accuracy increased substantially at lower C values, particularly around 0.01 to 1, before stabilizing at high training accuracy and approximately 92% validation accuracy when C exceeded 10. This pattern indicates that the selected C value of 9.9034 was located within a favorable parameter range. A smaller C value made the model more tolerant of classification errors, while a larger C value could increase the risk of overfitting because the model becomes stricter in separating classes. Overall, the combination of the learning curve and validation curve suggests that the SVM model without stratified train-test split achieved high and stable performance.

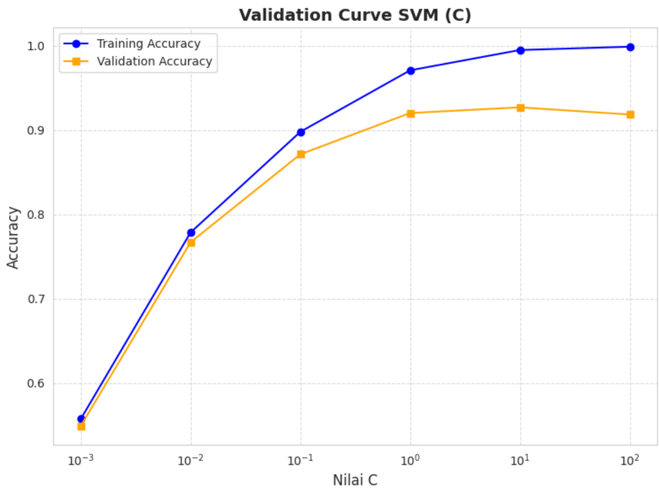


Figure 4. Validation Curve of the SVM Model Without Stratified Split

Training and Evaluation Results of Naïve Bayes Without Stratified Train-Test Split

Table 3 shows the hyperparameter tuning results of the Naïve Bayes model using Optuna without applying the stratified train-test split technique. The best alpha value was 0.5828, which produced the highest cross-validation accuracy of 0.8314. Although this performance was lower than that of SVM, the result indicates that Naïve Bayes still achieved acceptable performance after hyperparameter optimization.

Table 3. Hyperparameter Tuning Results of the Naïve Bayes Model Using Optuna

Best Parameter	Highest CV Accuracy
alpha: 0.5828	0.8314

Figure 5 shows that the training accuracy of Naïve Bayes ranged from approximately 0.75 to 0.94, while the validation accuracy increased from around 0.50 to 0.83 as the amount of training data increased. This pattern indicates improved generalization as more data became available. However, the initial gap between training and validation accuracy suggests that the model required a larger amount of data to stabilize its performance.

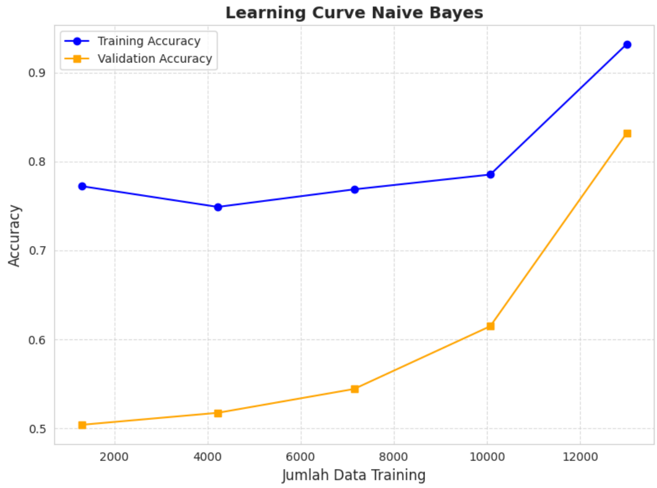


Figure 5. Learning Curve of the Naïve Bayes Model Without Stratified Train-Test Split

Figure 6 presents the validation curve for the alpha parameter. As alpha increased from 0 to 2, the training accuracy gradually decreased from around 0.95 to 0.91, while the validation accuracy remained relatively stable between 0.82 and 0.83. This result suggests that a very large alpha value may reduce the model's ability to fit the training data, although its effect on validation performance was limited. Therefore, selecting an appropriate alpha value is important to balance training accuracy and validation performance.

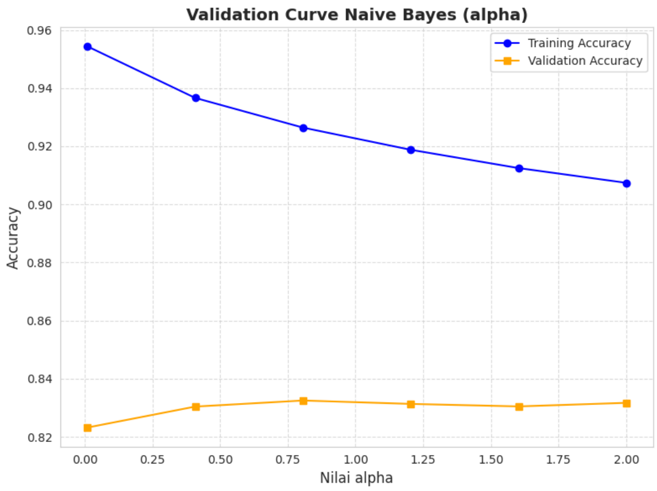


Figure 6. Validation Curve of the Naïve Bayes Model Without Stratified Train-Test Split

Training and Evaluation Results of SVM With Stratified Train-Test Split

Table 4 presents the tuning results of the SVM model after applying the stratified train-test split technique. The optimal parameters were a C value of 81.3586, a linear kernel, and gamma set to auto. The model achieved the highest cross-validation accuracy of 0.9172. This result indicates that the stratified configuration still produced strong performance, although the cross-validation accuracy was slightly lower than that of the SVM model without stratified train-test split.

Table 4. Tuning Results of the SVM Model

Parameter	Value
C	81.3586
Kernel	linear
Gamma	Auto (default)
Highest CV Accuracy	0.9172

Figure 7 shows that the SVM training accuracy remained close to 100% across different training sizes, indicating that the model learned the training data very effectively. Meanwhile, the validation accuracy increased from approximately 85% to nearly 92% as the amount of training data increased. This pattern suggests improved generalization and reduced variance as more training data were used.

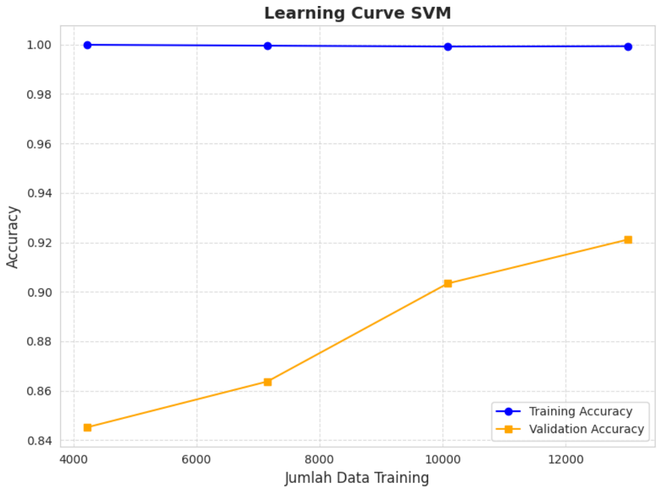


Figure 7. Learning Curve of the SVM Model With Stratified Train-Test Split

Figure 8 shows the validation curve for the C parameter. At very small C values, from 10^{-3} to 10^{-2}, both training and validation accuracy were low, indicating underfitting. As C increased to the range of 1 to 10, accuracy improved substantially, with validation accuracy reaching approximately 92%. However, when C exceeded 10, validation accuracy began to decline while training accuracy remained high, indicating a possible tendency toward overfitting. The selected C value of 81.3586 still produced high performance, but the curve suggests that careful control of the regularization parameter remains necessary to balance bias and variance.

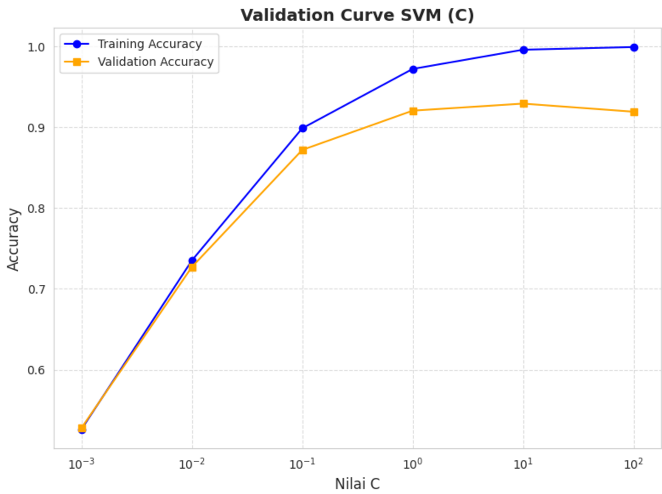


Figure 8. Validation Curve of the SVM Model With Stratified Train-Test Split

Training and Evaluation Results of Naïve Bayes With Stratified Train-Test Split

Table 5 presents the tuning results of the Naïve Bayes model after applying the stratified train-test split technique. The optimal alpha value was 0.9752, with the highest cross-validation accuracy of 0.8320. This result shows a slight improvement compared with the non-stratified scenario, although the performance remained substantially lower than that of SVM.

Table 5. Tuning Results of the Naïve Bayes Model

Best Parameter	Highest CV Accuracy
alpha: 0.9752	0.8320

Figure 9 shows that the training accuracy of Naïve Bayes initially remained within the range of approximately 0.70 to 0.75 when the training size was small, then increased sharply to around 0.93 as the dataset approached its maximum size. The validation accuracy also increased substantially from around 0.50 to approximately 0.84. This pattern indicates that Naïve Bayes benefited from larger training data, although its performance remained constrained compared with SVM.

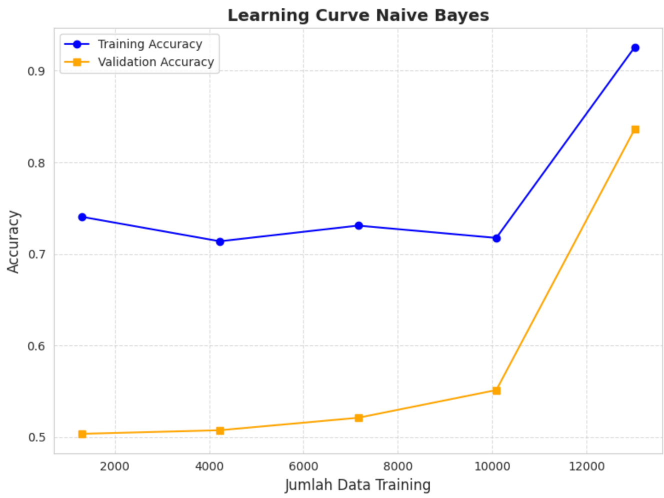
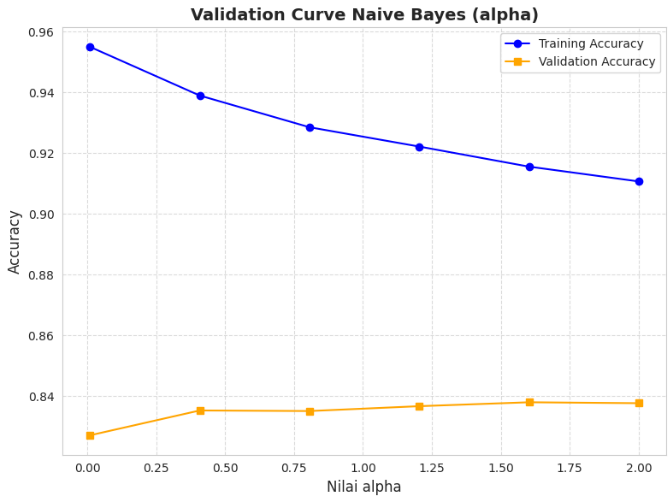


Figure 9. Learning Curve of the Naïve Bayes Model With Stratified Train-Test Split

Figure 10 shows that the highest training accuracy, approximately 0.955, occurred when alpha was close to zero. As alpha approached 2, training accuracy decreased to approximately 0.91. Meanwhile, validation accuracy increased from approximately 0.82 to 0.839, with the best balance between training and validation performance occurring when alpha was around 0.9 to 1.0. This result indicates that the stratified train-test split improved the stability of Naïve Bayes, although a performance gap between training and validation accuracy remained visible.



Gambar 10. *Validation Curve Model Naïve Bayes* dengan Teknik *Stratified Train-Test Split*

Model Performance Comparison

Table 6 compares the performance of SVM and Naïve Bayes based on cross-validation accuracy and test set accuracy under two imbalance-handling scenarios. The results show that SVM with SMOTE achieved the highest performance, with a cross-validation accuracy of 0.9272 and a test set accuracy of 0.8679. SVM with stratified train-test split and SMOTE achieved a slightly lower test set accuracy of 0.8502. In contrast, Naïve Bayes achieved lower performance in both scenarios, with test set accuracy values of 0.7545 and 0.7598. These findings indicate that SVM consistently outperformed Naïve Bayes in sentiment classification for the imbalanced Gojek Driver review dataset.

Table 6. Comparison of Cross-Validation Accuracy and Test Set Accuracy

Model	Imbalanced Dataset Technique	CV Accuracy	Test Set Accuracy
SVM	SMOTE	0.9272	0.8679
	Stratified Train-Test Split, SMOTE	0.9172	0.8502
Naïve Bayes	SMOTE	0.8315	0.7545
	Stratified Train-Test Split, SMOTE	0.8320	0.7598

Comparison of Accuracy, Precision, Recall, and F1-Score

Table 7 compares the test set performance of SVM and Naïve Bayes using accuracy, precision, recall, and F1-score with the weighted average method. The results show that SVM with SMOTE achieved the best performance across all metrics, with an accuracy of 0.8679, precision of 0.87, recall of 0.87, and F1-score of 0.87. SVM with stratified train-test split and SMOTE achieved slightly lower values across all metrics, with an accuracy of 0.8502 and an F1-score of 0.85. Naïve Bayes produced weaker results, with its highest F1-score reaching only 0.76. These results indicate that SVM was more consistent and more accurate in classifying sentiment, particularly under imbalanced data conditions.

Table 7. Model Performance Comparison on the Test Data

Model	Imbalanced Dataset Technique	Accuracy	Precision	Recall	F1-Score
SVM	SMOTE	0.8679	0.87	0.87	0.87

	Stratified Train-Test Split, SMOTE	0.8502	0.85	0.85	0.85
Naïve Bayes	SMOTE	0.7545	0.77	0.75	0.76
	Stratified Train-Test Split, SMOTE	0.7598	0.78	0.76	0.76

The confusion matrix results for both SVM and Naïve Bayes are presented in Figure 11 and Figure 12. The comparison was conducted between models without and with the stratified train-test split technique.

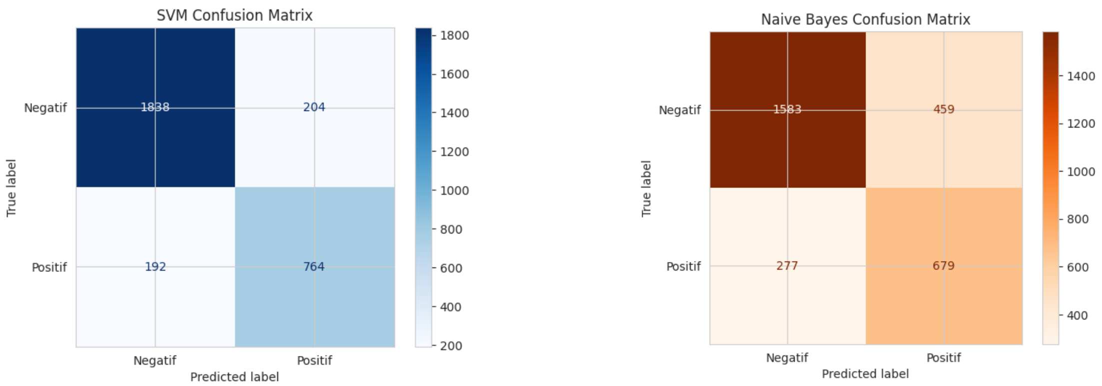


Figure 11. Confusion Matrix of the Models Without Stratified Train-Test Split

As shown in Figure 11, the SVM model without stratified train-test split produced 2,602 correct predictions, consisting of 1,838 negative reviews and 764 positive reviews, while 396 predictions were incorrect. The Naïve Bayes model produced 2,262 correct predictions, consisting of 1,583 negative reviews and 679 positive reviews, while 736 predictions were incorrect. This comparison confirms that SVM produced fewer misclassifications than Naïve Bayes in the non-stratified scenario.

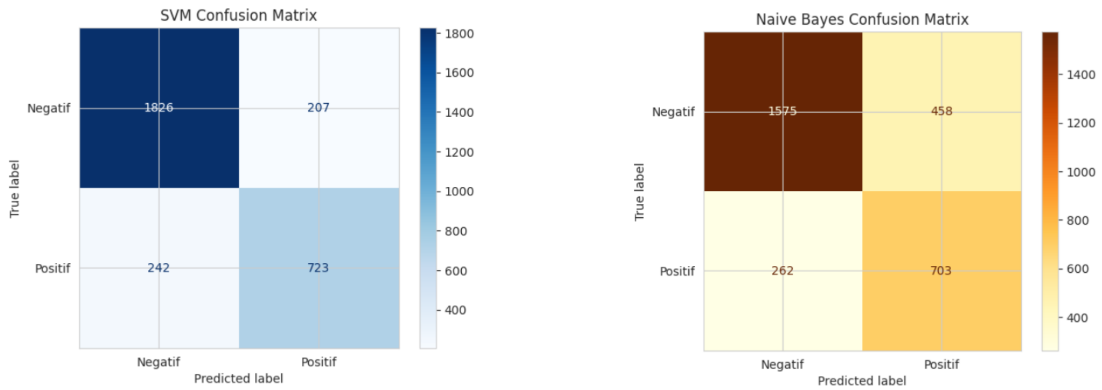


Figure 12. Confusion Matrix of the Models With Stratified Train-Test Split

As shown in Figure 12, the SVM model with stratified train-test split produced 2,549 correct predictions, consisting of 1,826 negative reviews and 723 positive reviews, while 449 predictions were incorrect. The Naïve Bayes model produced 2,278 correct predictions, consisting of 1,575 negative reviews and 703 positive reviews, while 720 predictions were incorrect. These results show that stratification provided a more controlled data partitioning strategy, but it did not necessarily improve SVM performance compared with the non-stratified scenario. For Naïve Bayes, the stratified scenario produced a small improvement in total correct predictions and test set accuracy.

Discussion

The findings show that SVM consistently outperformed Naïve Bayes in the sentiment classification of Gojek Driver application reviews. This superiority was observed across cross-validation accuracy, test set accuracy, precision, recall, F1-score, and confusion matrix results. The best-performing model was SVM with SMOTE, which achieved a cross-validation accuracy of 0.9272, a test set accuracy of 0.8679, and a weighted average F1-score of 0.87. These results suggest that SVM was better suited to handling the high-dimensional TF-IDF feature space and the imbalanced distribution of positive and negative reviews. This finding is consistent with previous studies reporting that SVM performs strongly in text classification and imbalanced sentiment analysis contexts [27], [28].

The performance difference between SVM and Naïve Bayes can be explained by the nature of the two algorithms. SVM is designed to identify an optimal separating hyperplane with a maximum margin between classes. This mechanism makes it effective for sparse and high-dimensional text data, such as TF-IDF representations. In contrast, Naïve Bayes relies on the conditional independence assumption among features. This assumption is often too restrictive for natural language data because words in user reviews are semantically and contextually related. Indonesian application reviews also frequently contain informal expressions, abbreviations, mixed sentiment, and context-dependent wording. These characteristics can weaken Naïve Bayes performance, particularly when sentiment is not expressed through isolated words but through combinations of terms.

The results also indicate that SMOTE played an important role in improving model performance under class imbalance. The dataset contained 10,165 negative reviews and 4,825 positive reviews, which means that negative reviews dominated the classification task. Without proper imbalance handling, the model could become biased toward the majority class. Applying SMOTE only to the training data helped reduce this risk while preserving the original distribution of the testing data for unbiased evaluation. The stronger performance of SVM with SMOTE suggests that synthetic minority oversampling was more effectively exploited by SVM than by Naïve Bayes.

The comparison between non-stratified and stratified train-test split scenarios requires careful interpretation. Stratified splitting is methodologically useful because it preserves class proportions in both training and testing sets. However, in this study, stratification did not automatically produce the highest SVM performance. SVM with SMOTE but without stratified train-test split achieved higher test set accuracy than SVM with stratified train-test split and SMOTE. This does not mean that stratification is methodologically inferior. Rather, it shows that performance differences may depend on the specific data partition, class distribution in each subset, and interaction between SMOTE, TF-IDF features, and model hyperparameters. For Naïve Bayes, stratified splitting produced a slight improvement, with test set accuracy increasing from 0.7545 to 0.7598. This indicates that Naïve Bayes may benefit modestly from more stable class distribution, although its overall performance remained limited.

Naïve Bayes appeared to reach a performance ceiling despite hyperparameter optimization. The best Naïve Bayes configuration achieved cross-validation accuracy values of 0.8314 and 0.8320 in the two scenarios, which were relatively close. This pattern suggests that the model had limited room for further improvement under the current feature representation and labeling approach. The likely reason is that Naïve Bayes is less flexible in modeling complex decision boundaries within high-dimensional text data. Even after applying SMOTE and Optuna-based tuning, the model remained sensitive to uneven word distributions and the independence assumption among terms. This explains why its F1-score remained around 0.76, while the best SVM configuration reached 0.87.

From an applied perspective, the results suggest that SVM with SMOTE is the more reliable option for sentiment analysis of Gojek Driver reviews. The model can support the automatic identification of user sentiment patterns and help detect service-related problems more efficiently. For platform managers, this approach may assist in monitoring user concerns, prioritizing technical issues, and generating evidence-based feedback for service improvement. Naïve Bayes may still be useful when computational simplicity and faster training are more important than maximum classification accuracy. However, for applications requiring stronger predictive performance on imbalanced review data, the findings clearly favor SVM.

This study also has several methodological limitations. The sentiment labels were generated using a lexicon-based approach, which may introduce labeling errors because lexicons cannot always capture sarcasm, context, domain-specific expressions, or mixed sentiment in user reviews. The study also used only classical machine learning algorithms, namely SVM and Naïve Bayes. Although these models are useful for comparative analysis and interpretable baseline classification, transformer-based models such as IndoBERT, BERT, or RoBERTa may provide better contextual representation for Indonesian text. In addition, this study did not examine temporal sentiment patterns, even though user sentiment toward an application such as Gojek Driver may change over time due to policy updates, technical changes, service disruptions, or platform feature modifications.

Future studies should consider manual annotation or expert validation on a subset of reviews to improve label reliability and estimate the accuracy of lexicon-based labeling. Further research may also compare SVM and Naïve Bayes with transformer-based models to determine whether contextual embeddings improve classification performance in Indonesian ride-hailing review data. Temporal sentiment analysis would also be valuable for understanding how user perceptions change across application updates or service policy changes. Ensemble learning may provide another promising direction, particularly if future studies aim to combine the interpretability of classical models with the predictive strength of more advanced architectures.

CONCLUSION

This study contributes to the methodological development of sentiment analysis for digital service applications in Indonesia, particularly in addressing class imbalance in local user review datasets. By comparing Support Vector Machine and Naïve Bayes under different imbalance-handling scenarios, this study demonstrates that model performance is strongly influenced by the interaction between feature representation, data distribution, oversampling strategy, and hyperparameter optimization.

The empirical results show that Support Vector Machine, particularly when combined with SMOTE and Optuna-based hyperparameter tuning, outperformed Naïve Bayes across the main evaluation metrics. The best SVM configuration achieved the highest cross-validation accuracy of 92.72%, a test set accuracy of 86.79%, and a weighted average F1-score of 0.87. In comparison, Naïve Bayes produced lower performance, with the best cross-validation accuracy remaining around 83.20% and the highest F1-score reaching only 0.76. These results indicate that SVM was more effective in handling high-dimensional TF-IDF features and imbalanced sentiment classes in Gojek Driver application reviews.

The findings suggest that SVM with SMOTE can serve as a reliable baseline model for automated sentiment analysis in ride-hailing application review data. Practically, this model can help platform managers identify user concerns, detect service-related issues, and support more responsive decision-making based on large-scale user feedback. Naïve Bayes may still be useful in contexts where computational simplicity and faster model training are prioritized, but its predictive performance is less competitive for the dataset examined in this study.

This study has several limitations. The sentiment labels were generated using a lexicon-based approach, which may be affected by the inability of lexicons to capture sarcasm, contextual meaning, mixed sentiment, and informal expressions in Indonesian user reviews. The study also focused only on two classical machine learning algorithms, namely SVM and Naïve Bayes. Future research should consider manual annotation or expert validation to improve label reliability, compare classical models with transformer-based models such as IndoBERT, BERT, or RoBERTa, and incorporate temporal sentiment analysis to examine how user perceptions change over time in response to application updates, service policies, or platform-related issues.

AUTHOR CONTRIBUTION STATEMENT

NM contributed to data collection, lexicon-based sentiment labeling, model development, hyperparameter optimization using Optuna, and the application of SMOTE for class imbalance handling. NM also contributed to the implementation of train-test splitting procedures and the interpretation of the model results. RS contributed to the research design, model evaluation, experimental analysis, and validation strategy. RS also led the writing of the results and discussion sections and reviewed the manuscript for conceptual and methodological coherence.

AI DISCLOSURE STATEMENT

The authors used ChatGPT during the preparation of this manuscript to obtain writing feedback, refine ideas, improve argument clarity, and support language editing. After using this tool, the authors carefully reviewed, revised, and edited the generated content as needed. The authors take full responsibility for the accuracy, integrity, and final content of this manuscript.

REFERENCES

[1] S. Biswas and G. Poornalatha, "Opinion Mining Using Multi-Dimensional Analysis," *IEEE Access*, vol. 11, pp. 25906–25916, 2023, doi: 10.1109/ACCESS.2023.3256521.

[2] T. Sugihartono and R. R. C. Putra, "PENERAPAN METODE SUPPORT VECTOR MACHINE DALAM CLASSIFIKASI ULASAN PENGGUNA APLIKASI MOBILE JKN," *SKANIKA Sist. Komput. Dan Tek. Inform.*, vol. 7, no. 2, pp. 144–153, Jul. 2024, doi: 10.36080/skanika.v7i2.3193.

[3] A. Suryana, A. I. Purnamasari, and I. Ali, "MENGOPTIMALKAN KEPUASAN PENGGUNA: ANALISIS SENTIMEN REVIEW APLIKASI GRAB DI INDONESIA," *JATI J. Mhs. Tek. Inform.*, vol. 8, no. 3, pp. 3396–3404, May 2024, doi: 10.36040/jati.v8i3.9688.

[4] N. Amat-Lefort, F. Barravecchia, and L. Mastrogiacomo, "Quality 4.0: big data analytics to explore service quality attributes and their relation to user sentiment in Airbnb reviews," *Int. J. Qual. Reliab. Manag.*, vol. 40, no. 4, pp. 990–1008, Mar. 2023, doi: 10.1108/IJQRM-01-2022-0024.

[5] S. V, G. V. Varshini, N. M, D. R, V. K. Kaliappan, and J. Li, "Customer Feedback Automation and Classification using Sentiment Analysis, Robotic Process Automation and Uipath," in *Recent Trends in Data Science and its Applications*, Kattankulathur, Tamil Nadu, India: River Publishers, 2023, pp. 899–907. doi: 10.13052/rp-9788770040723.175.

[6] K. Huda, S. D. Pohan, and Y. Herlina, "PENERAPAN PEMBOBOTAN TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY DAN ALGORITMA K-NEAREST NEIGHBOR UNTUK ANALISIS ULASAN HOTEL DI SITUS TRIPADVISOR," *J. Inform. Dan Tek. Elektro Terap.*, vol. 12, no. 3, Aug. 2024, doi: 10.23960/jitet.v12i3.4800.

[7] S. Siswanto, Z. Mar'ah, A. S. D. Sabir, T. Hidayat, F. A. Adhel, and W. S. Amni, "The Sentiment Analysis Using Naïve Bayes with Lexicon-Based Feature on TikTok Application," *J. Varian*, vol. 6, no. 1, pp. 89–96, Nov. 2022, doi: 10.30812/varian.v6i1.2205.

[8] A. Safira and F. N. Hasan, "ANALISIS SENTIMEN MASYARAKAT TERHADAP PAYLATER MENGGUNAKAN METODE NAIVE BAYES CLASSIFIER," *ZONAsi J. Sist. Inf.*, vol. 5, no. 1, pp. 59–70, Jan. 2023, doi: 10.31849/zn.v5i1.12856.

[9] A. Jaha, N. Satya Deepthi, G. Suryakanth, and G. S. K. Eswari, "Text Sentiment Analysis Using Naïve Baye's Classifier," *Int. J. Comput. Trends Technol.*, vol. 68, no. 4, pp. 261–265, Apr. 2020, doi: 10.14445/22312803/IJCTT-V68I4P141.

[10] K. Hanif, A. Fadllullah, N. R. Muntiari, and I. A. Fahrezi, "A Comparative Sentiment Analysis of Computer Engineering Student Feedback Using Decision Trees and SVM," *J. Inotera*, vol. 10, no. 1, pp. 71–82, Feb. 2025, doi: 10.31572/inotera.Vol10.Iss1.2025.ID436.

[11] F. Fersellia, E. Utami, and A. Yaqin, "Sentiment Analysis of Shopee Food Application User Satisfaction Using the C4.5 Decision Tree Method," *Sinkron*, vol. 8, no. 3, pp. 1554–1563, Jul. 2023, doi: 10.33395/sinkron.v8i3.12531.

[12] Y. Liu and H. Kou, "Student career guidance using sentiment analysis and decision tree models," *Int. J. Inf. Commun. Technol.*, vol. 26, no. 38, pp. 1–21, 2025, doi: 10.1504/IJICT.2025.149290.

[13] A. A. Sulaeman, M. Danny, S. Butsianto, and S. Pratama, “Sentiment Analysis on Social Media X (Twitter) Against ChatGBT Using the K-Nearest Neighbors Algorithm,” *Brill. Res. Artif. Intell.*, vol. 4, no. 1, pp. 265–275, Jul. 2024, doi: 10.47709/brilliance.v4i1.4105.
[14] D. Wijati, P. D. Atika, S. Setiawati, and R. Rasim, “Sentiment Analysis of Application Reviews using the K-Nearest Neighbors (KNN) Algorithm,” *PIKSEL Penelit. Ilmu Komput. Sist. Embed. Log.*, vol. 12, no. 1, pp. 209–218, Mar. 2024, doi: 10.33558/piksel.v12i1.9490.
[15] K. Nugroho, E. Winarno, D. R. I. M. Setiadi, and O. Farooq, “Enhanced multi-lingual Twitter sentiment analysis using hyperparameter tuning k-nearest neighbors,” *Bull. Electr. Eng. Inform.*, vol. 13, no. 6, pp. 4327–4334, Dec. 2024, doi: 10.11591/eei.v13i6.7265.
[16] A. Pambudi and S. Suprapto, “Effect of Sentence Length in Sentiment Analysis Using Support Vector Machine and Convolutional Neural Network Method,” *IJCCS Indones. J. Comput. Cybern. Syst.*, vol. 15, no. 1, p. 21, Jan. 2021, doi: 10.22146/ijccs.61627.
[17] Muthia Sakhdiah, Admi Salma, Dony Permana, and Dina Fitria, “Sentiment Analysis Using Support Vector Machine (SVM) of ChatGPT Application Users in Play Store,” *UNP J. Stat. Data Sci.*, vol. 2, no. 2, pp. 151–158, May 2024, doi: 10.24036/ujsds/vol2-iss2/158.
[18] T. A. Khan, R. Sadiq, Z. Shahid, M. M. Alam, and M. Mohd Su’ud, “Sentiment Analysis using Support Vector Machine and Random Forest,” *J. Inform. Web Eng.*, vol. 3, no. 1, pp. 67–75, Feb. 2024, doi: 10.33093/jiwe.2024.3.1.5.
[19] F. M. Sarimole and K. Kudrat, “Analisis Sentimen Terhadap Aplikasi Satu Sehat Pada Twitter Menggunakan Algoritma Naive Bayes Dan Support Vector Machine: Analisis Sentimen,” *J. Sains Dan Teknol.*, vol. 5, no. 3, pp. 783–790, Feb. 2024, doi: 10.55338/saintek.v5i3.2702.
[20] R. Setiawan and F. Dewi, “Analysis of Twitter User Sentiment on Presidential Candidate Anies Baswedan Using Naïve Bayes Algorithm,” *Ilk. J. Ilm.*, vol. 15, no. 3, pp. 473–487, Dec. 2023, doi: 10.33096/ilkom.v15i3.1775.473-487.
[21] W. Saidi, A. E. Abderrahmani, and K. Satori, “EFFECTIVE COMPARATIVE EVALUATION OF SENTIMENT ANALYSIS USING PAIRED T-TEST: A PERFORMANCE STUDY OF SUPERVISED METHODS,” *J. Southwest Jiaotong Univ.*, vol. 58, no. 5, 2023, doi: 10.35741/issn.0258-2724.58.5.28.
[22] A. Thumpati and Y. Zhang, “Towards Optimizing Performance of Machine Learning Algorithms on Unbalanced Dataset,” in *Artificial Intelligence & Applications*, Academy & Industry Research Collaboration Center, Oct. 2023, pp. 169–183. doi: 10.5121/csit.2023.131914.
[23] S. N. Almuayqil, M. Humayun, N. Z. Jhanjhi, M. F. Almufareh, and N. A. Khan, “Enhancing Sentiment Analysis via Random Majority Under-Sampling with Reduced Time Complexity for Classifying Tweet Reviews,” *Electronics*, vol. 11, no. 21, p. 3624, Nov. 2022, doi: 10.3390/electronics11213624.
[24] E. Hokijuliandy, H. Napitupulu, and Firdaniza, “Application of SVM and Chi-Square Feature Selection for Sentiment Analysis of Indonesia’s National Health Insurance Mobile Application,” *Mathematics*, vol. 11, no. 17, p. 3765, Sep. 2023, doi: 10.3390/math11173765.
[25] R. Rahmanda and E. B. Setiawan, “Word2Vec on Sentiment Analysis with Synthetic Minority Oversampling Technique and Boosting Algorithm,” *J. RESTI Rekayasa Sist. Dan Teknol. Inf.*, vol. 6, no. 4, pp. 599–605, Aug. 2022, doi: 10.29207/resti.v6i4.4186.
[26] D. A. Kristiyanti, S. A. Sanjaya, V. C. Tjokro, and J. Suhali, “Dealing imbalance dataset problem in sentiment analysis of recession in Indonesia,” *IAES Int. J. Artif. Intell. IJ-AI*, vol. 13, no. 2, p. 2060, Jun. 2024, doi: 10.11591/ijai.v13.i2.pp2060-2072.
[27] R. Obiedat *et al.*, “Sentiment Analysis of Customers’ Reviews Using a Hybrid Evolutionary SVM-Based Approach in an Imbalanced Data Distribution,” *IEEE Access*, vol. 10, pp. 22260–22273, 2022, doi: 10.1109/ACCESS.2022.3149482.
[28] G. Ryan, P. Katarina, and D. Suhartono, “MBTI Personality Prediction Using Machine Learning and SMOTE for Balancing Data Based on Statement Sentences,” *Information*, vol. 14, no. 4, p. 217, Apr. 2023, doi: 10.3390/info14040217.