BIMA
BULLETIN OF INTELLIGENT MACHINES AND ALGORITHMS
Vol. 2 No.1 November 2026
E-ISSN: 3132-5115
DOI:10.65780/bima.v2i1.41

Comparison of Machine Learning Algorithms for Sentiment Classification of Spotify App Reviews on the Google Play Store

Icha Friska Ameylia[1*]; Budiman[2]; Venia Restreva Danestiara[3]

Informatics[1,2,3]
Universitas Informatika dan Bisnis Indonesia, Bandung, Indonesia [1,2,3]
https://unibi.ac.id/ [1,2,3]
icha.fa22@student.unibi.ac.id [1*]

Abstract. The large number of user reviews of the Spotify application on the Google Play Store requires an analysis method capable of processing text data automatically and objectively. This study utilized 5,000 review data collected through scraping, which were then processed into 4,605 labeled data consisting of positive and negative sentiments. The methods applied include text Preprocessing, TF-IDF feature extraction, and the implementation of several Machine Learning algorithms, namely Naïve Bayes, Support Vector Machine (SVM), Logistic Regression, Random Forest, Decision Tree, and K-Nearest Neighbor (KNN), with data balancing using SMOTE. The results show that the application of SMOTE improves model performance, particularly in recognizing the minority class. The Random Forest algorithm achieved the best performance with an accuracy of 84.58% and an F1-score of 81.71%, followed by Logistic Regression with competitive results. Further analysis using SHAP and Learning Curves indicates that the model has good generalization ability and can identify important features influencing sentiment classification. This study demonstrates that the proposed methods are effective for sentiment analysis of Spotify user reviews.

Key words: Sentiment Analysis, Machine Learning, Random Forest, TF-IDF, SMOTE, Spotify

1. INTRODUCTION

Among the millions of applications available through mobile application marketplaces, music streaming services have become an important part of contemporary digital entertainment [1]. The rapid adoption of mobile applications has generated substantial volumes of user-generated reviews on platforms such as the Google Play Store, providing information about users' perceptions, experiences, satisfaction, and complaints regarding the applications they use [2]. Previous research has demonstrated that app reviews can contain valuable information for developers, including user requirements, suggestions for improvement, opinions about specific application features, and descriptions of user experiences [3], [4]. In the case of Spotify, user reviews on the Google Play Store have been increasingly investigated as a source for understanding user perceptions and evaluating application quality [5], [6]. These reviews may reveal a wide range of issues, including satisfaction with application features, technical problems, service limitations, and other aspects of the user experience [7]. However, the large volume of reviews makes comprehensive manual analysis difficult and time-consuming, particularly when developers need to identify recurring issues and distinguish positive and negative user perceptions.

Therefore, an automated approach is required to process and classify the sentiment of user reviews efficiently and systematically [8]. Sentiment analysis enables the automatic identification of opinions, attitudes, and emotions expressed in large collections of textual data and has become an important application of Natural Language Processing (NLP) for analyzing user-generated content [9], [10]. In general, sentiment analysis approaches can be categorized into rule-based, machine learning-based, and hybrid methods [11]. While rule-based approaches rely on predefined sentiment lexicons and linguistic rules, machine learning methods learn sentiment patterns from labeled data and can be adapted to different application domains [6], [9]. In the context of application reviews, the selection of a classification algorithm may substantially influence the resulting performance because different algorithms have different characteristics in handling textual features and data distributions [12], [13].

Published by Maheswari Publisher
Creation disseminated under Creative Commons Attribution 4.0 International License

Vol. 2 No.1 November 2026
E-ISSN: 3132-5115
DOI:10.65780/bima.v2i1.41

Based on this background, this study aims to conduct a comparative analysis of six Machine Learning algorithms for binary sentiment classification of Spotify user reviews collected from the Google Play Store, namely Naïve Bayes, Support Vector Machine (SVM), Logistic Regression, Random Forest, Decision Tree, and K-Nearest Neighbor (KNN). In addition to evaluating conventional classification metrics, this study integrates Explainable Artificial Intelligence (XAI) using SHapley Additive exPlanations (SHAP) to improve the interpretability of the classification model and identify the contribution of individual textual features to model predictions. By combining comparative evaluation with model interpretability analysis, this study seeks not only to determine the best-performing algorithm but also to provide insights into the features that influence the model's sentiment classification decisions.

2. RELATED WORK

Recent studies have increasingly explored user reviews from application stores as a valuable source for understanding user perceptions, requirements, and experiences. App reviews contain textual feedback that can support software developers in identifying user needs, feature-related opinions, and problems encountered during application use, while the large volume of reviews makes comprehensive manual analysis impractical [3], [4]. In the context of Spotify, Madyatmadja et al. [6] analyzed 14,296 reviews collected from the Google Play Store and App Store using Naïve Bayes, Support Vector Machine (SVM), and Random Forest. Their results showed that SVM achieved the highest accuracy of 87.4%, followed by Naïve Bayes at 85.7% and Random Forest at 85.6%. More recent research has continued to demonstrate that the relative performance of classification algorithms varies according to the dataset and experimental setting. For example, Olivian et al. compared SVM, Neural Network, and Random Forest for Spotify reviews and reported that SVM and Neural Network achieved an accuracy of 93.7%, substantially exceeding Random Forest at 85.3% [14]. These findings suggest that algorithm performance should not be assumed to be universally superior across different application-review datasets.

Beyond the choice of classification algorithm, data characteristics also influence the reliability of sentiment classification. Application-review datasets commonly exhibit uneven class distributions, which can cause conventional classifiers to favor the majority class and perform poorly when identifying minority instances. Chawla et al. [15] introduced the Synthetic Minority Over-sampling Technique (SMOTE) to address this problem by generating synthetic samples for the minority class, providing an alternative to simple replication-based oversampling. In addition, recent studies on app-review mining indicate that Natural Language Processing and Machine Learning are among the dominant techniques used to extract information from application reviews [15]. Consequently, comparing multiple algorithms under the same preprocessing, feature representation, data-balancing, and evaluation conditions is important because differences in experimental settings can otherwise obscure the actual contribution of the classifier itself. This consideration motivates the use of a unified experimental framework in the present study.

Based on these findings, a gap remains in the comprehensive evaluation of multiple classical Machine Learning algorithms for binary sentiment classification of Spotify reviews while simultaneously considering class imbalance, model generalization, and interpretability. Existing Spotify-oriented studies have generally focused on a limited number of algorithms, making it difficult to establish a broader comparison among probabilistic, linear, tree-based, ensemble, and instance-based classifiers [6], [14]. Therefore, this study compares six algorithms—Naïve Bayes, Support Vector Machine, Logistic Regression, Random Forest, Decision Tree, and K-Nearest Neighbor—under a common experimental framework and applies SMOTE only to the training data to reduce the risk of data leakage [15]. Furthermore, Learning Curve analysis is incorporated to examine generalization behavior, while SHAP is employed to identify the contribution of individual textual features to model predictions. SHAP provides feature-level attribution for individual predictions and was specifically developed to improve the interpretability of complex predictive models [16]. Through this combination, the study evaluates not only which algorithm produces the strongest classification performance, but also how reliably the model generalizes and which textual features influence its decisions.

3. METHODS

This study adopts a quantitative comparative-experimental design within a supervised learning framework to examine sentiment classification of Spotify user reviews collected from the Google Play Store. The research workflow consists of eleven sequential stages, as illustrated in Figure 1, beginning with data collection and ending with the comparison of classification algorithms and discussion of the results. A total of 5,000 Spotify reviews were collected using the google-play-scraper library, with each review accompanied by a 1–5 star rating. Sentiment labels were assigned automatically according to the rating: reviews with 1–2 stars were categorized as negative, whereas reviews with 4–5 stars were categorized as positive. Reviews with a rating of 3 stars were excluded because they may represent an ambiguous sentiment and could reduce the consistency of binary classification. The remaining reviews underwent text preprocessing consisting of cleaning, tokenization, normalization, stopword removal, and stemming. The processed text was subsequently transformed into numerical feature representations using Term Frequency–Inverse Document Frequency (TF-IDF) with a maximum of 5,000 features and unigram–bigram representation. The resulting dataset was divided into training and testing sets

Published by Maheswari Publisher
Creation disseminated under Creative Commons Attribution 4.0 International License

Vol. 2 No.1 November 2026
E-ISSN: 3132-5115
DOI:10.65780/bima.v2i1.41

using an 80:20 stratified split so that the class proportions were maintained between the two subsets.

To address the imbalance between positive and negative reviews, SMOTE was applied exclusively to the training data, while the testing data were retained in their original distribution. This separation was maintained to prevent information from the testing set from influencing the training process and, consequently, to provide a more reliable estimate of model performance. Six supervised Machine Learning algorithms were then evaluated under the same experimental conditions: Naïve Bayes, Support Vector Machine (SVM), Logistic Regression, Random Forest, Decision Tree, and K-Nearest Neighbor (KNN). For KNN, the value of k was systematically examined from 1 to 100 using the balanced training data, and the value producing the highest performance was selected for subsequent evaluation. Model training and validation were performed using 5-Fold Cross Validation with macro F1-score as the primary validation criterion. The trained models were subsequently evaluated on the independent testing set using accuracy, precision, recall, F1-score, and confusion matrix. This common experimental setting enables the performance differences to be attributed more consistently to the characteristics of the algorithms rather than to differences in data preparation or evaluation procedures.

The final stages focused on examining model behavior beyond conventional performance metrics. As presented in Figure 1, Learning Curve analysis was conducted to assess the relationship between the amount of training data and model performance, particularly to identify indications of overfitting or underfitting. The best-performing model was then further examined using Explainable Artificial Intelligence (XAI) through SHAP (SHapley Additive exPlanations) to identify textual features that contributed to its classification decisions. Finally, the six algorithms were compared based on their predictive performance, cross-validation stability, generalization behavior, and interpretability analysis. This procedure provides a broader basis for selecting the most suitable classifier for Spotify review sentiment classification rather than relying solely on a single evaluation metric.

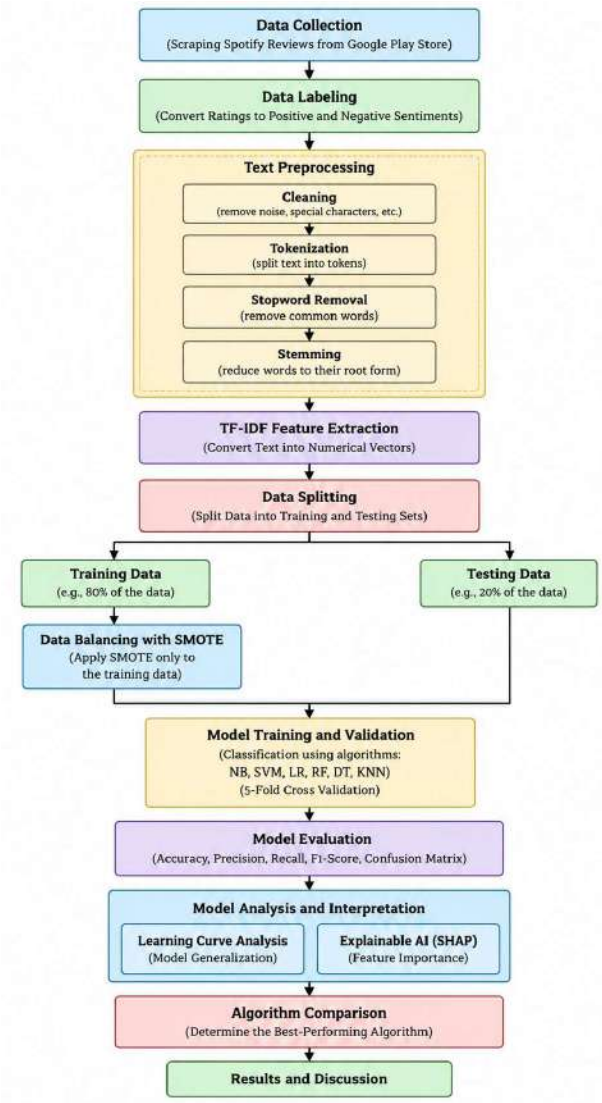


Fig. 1. Research Stages

4. RESULTS AND DISCUSSIONS

4.1. Results

This section presents the results of sentiment classification of Spotify application user reviews on the Google Play Store using six Machine Learning algorithms, namely Naïve Bayes, Support Vector Machine (SVM), Logistic Regression, Random Forest, Decision Tree, and K-Nearest Neighbor (KNN), to address the three research questions: (1) the performance of each algorithm, (2) the comparison of accuracy, precision, recall, and F1-score metrics, and (3) the identification of the most optimal algorithm to be recommended.

The review data were collected through scraping from the Google Play Store, resulting in 5,000 reviews, which were then labeled as binary sentiment based on ratings (1–2 stars = negative, 4–5 stars = positive; 3-star ratings were removed because they were semantically ambiguous),

Published by Maheswari Publisher
Creation disseminated under Creative Commons Attribution 4.0 International License

Vol. 2 No.1 November 2026
E-ISSN: 3132-5115
DOI:10.65780/bima.v2i1.41

resulting in 4,717 reviews (3,495 positive and 1,222 negative, class ratio of 2.86×). After undergoing the preprocessing stages (cleaning, tokenization, normalization, stopword removal, and stemming) and TF-IDF feature extraction (max_features=5000, ngram_range = (1,2)), 4,605 data instances were obtained and were ready for modeling (3,391 positive and 1,214 negative; class ratio of 2.79×), which were divided into training data (80%) and testing data (20%). SMOTE was applied only to the training data, while the testing data remained in its original distribution.

The optimal k value for the KNN algorithm was determined systematically within the range of k=1 to k=100 using the SMOTE-balanced training data and evaluated on the testing data. The search results showed that k=51 produced the highest accuracy of 82.74%; therefore, this value was used throughout all subsequent KNN training and evaluation processes. The six algorithms were trained using the SMOTE-balanced training data and validated using 5-Fold Cross Validation based on the macro F1-score metric. The validation results are presented.

TABLE 1. Cross-Validation Results

Algorithm	CV F1 (Mean)	CV F1 (±Std)
Naïve Bayes	0,8919	±0,0105
SVM	0,8934	±0,0125
Logistic Regression	0,8818	±0,0167
Random Forest	0,8968	±0,0060
Decision Tree	0,8183	±0,0128
KNN	0,8327	±0,0087

Random Forest achieved the highest cross-validation F1-score (0.8968) with the smallest standard deviation (±0.0060), indicating the most stable and consistent performance across folds, while Decision Tree obtained the lowest score (0.8183). The final evaluation was conducted on the testing data (20%) using four evaluation metrics: accuracy, precision, recall, and F1-score. The evaluation results are presented in Table 1, which address the first and second research questions of this study.

TABLE 2. Evaluation Results

Algorithm	Accuracy	Precision	Recall	F1
Naïve Bayes	0,8306	0,7841	0,8282	0,7991
SVM	0,8089	0,7610	0,8042	0,7747
Logistic Regression	0,8371	0,7954	0,8537	0,8116
Random Forest	0,8458	0,8011	0,8478	0,8171
Decision Tree	0,8154	0,7626	0,7796	0,7701
KNN	0,8274	0,7877	0,8497	0,8026

Based on Table 2, Random Forest achieved the best performance with an accuracy of 84.58% and an F1-score of 81.71%, followed by Logistic Regression (accuracy of 83.71%; F1-score of 81.16%), with a difference of only 0.55%. KNN ranked third (accuracy of 82.74%; F1-score of 80.26%), while Decision Tree and SVM achieved the lowest performance.

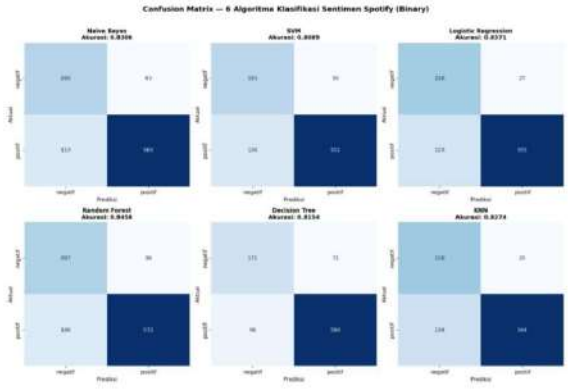


Fig. 2. Confusion Matrix For 6 Classification Algorithms

The confusion matrix in Figure 2 shows that Random Forest has the highest number of correct predictions (true positive and true negative). Decision Tree and SVM have a higher number of false negatives, indicating that these two models more frequently misclassified negative reviews as positive, while KNN and Logistic Regression demonstrated good recall for the negative class. To evaluate the contribution of data balancing, all models were retrained using the original imbalanced training data without SMOTE. The comparison of F1-scores is presented in Table 2.

TABLE 3. Comparison of Training Data Using SMOTE

Algorithm	**Without SMOTE**	**With SMOTE**	**Difference**
Naïve Bayes	0,7823	0,7991	+0,0168↑
SVM	0,7591	0,7747	+0,0156↑
Logistic Regression	0,7942	0,8116	+0,0174↑
Random Forest	0,8043	0,8171	+0,0128↑
Decision Tree	0,7489	0,7701	+0,0212↑
KNN	0,7801	0,8026	+0,0225↑

The application of SMOTE was proven to improve the F1-score of all algorithms, with the largest improvement observed in KNN (+0.0225) and Decision Tree (+0.0212), while the smallest improvement was observed in Random Forest (+0.0128). This demonstrates that SMOTE is effective in improving the models' ability to recognize the negative class, which was previously underrepresented.

As a specific comparison between tree-based algorithms, Table 3 shows that the ensemble approach of Random Forest achieved higher and more stable performance than a single Decision Tree across all evaluation metrics.

TABLE 4. Results of Tree-Based Algorithms (Random Forest and Decision Tree)

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1**
Random Forest	0,8458	0,8011	0,8478	0,8171
Decision Tree	0,8154	0,7626	0,7796	0,7701

The Learning Curve analysis shows that Naïve Bayes, SVM, Logistic Regression, and KNN have good generalization capability, as indicated by the decreasing gap between the

Published by Maheswari Publisher
Creation disseminated under Creative Commons Attribution 4.0 International License

Vol. 2 No.1 November 2026
E-ISSN: 3132-5115
DOI:10.65780/bima.v2i1.41

training score and the validation score as the amount of data increases. Random Forest and Decision Tree show indications of overfitting because their training scores remain high (>0.97) while their validation scores are lower; however, for Random Forest, the validation score remains stable at a high level (~0.90), indicating that its tendency toward overfitting is still within an acceptable range. The Explainable AI analysis using SHAP (SHapley Additive exPlanations) on the best-performing model (Random Forest) shows that the words contributing most to the negative class are related to complaints about advertisements, limitations of free features, and technical issues within the application, whereas the words contributing to the positive class express satisfaction with the quality of the application and the convenience of listening to music. These findings are consistent with the results of the exploratory data analysis, which showed the dominance of the words "iklan" and "premium" in negative reviews, thereby confirming that the model successfully captured relevant semantic patterns.

Analysis and Comparison of Algorithms

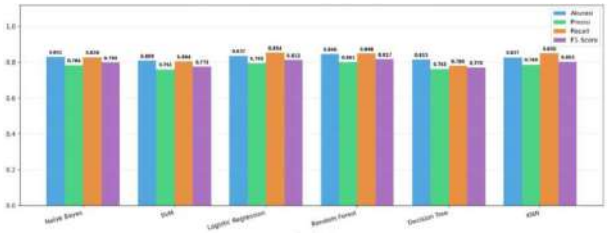


Fig. 3. Comparison Results Of 6 Algorithms

Based on the overall evaluation results presented, Random Forest was selected as the most optimal algorithm, achieving an F1-score of 81.71% and an accuracy of 84.58%, consistent with the Cross Validation results, which also identified Random Forest as the best and most stable model (CV F1-score of 0.8968; ±0.0060). This superiority is explained by the algorithm's characteristics as an ensemble method that combines the predictions of 200 decision trees through majority voting, making it more resistant to overfitting than a single Decision Tree and capable of capturing non-linear patterns in high-dimensional TF-IDF data. Logistic Regression, as the baseline model, ranked second with only a slight difference in F1-score (0.55%), indicating that the TF-IDF features used were sufficiently representative even for a simple linear model. Therefore, the selection of the best algorithm in this study was based on a combination of evaluation metrics, Cross Validation stability, and the suitability of the algorithm's characteristics to the dataset, rather than on a single evaluation metric alone.

4.2. Discussion

The results indicate that Random Forest achieved the best overall performance in the present study, with an accuracy of 84.58% and an F1-score of 81.71%, followed closely by Logistic Regression with an accuracy of 83.71% and an F1-score of 81.16%. This finding differs from previous studies on Spotify reviews. Madyatmadja et al. reported that SVM achieved the highest accuracy (87.4%), followed by Naïve Bayes (85.7%) and Random Forest (85.6%) [6]. Similarly, Olivian et al. found that SVM and Neural Network achieved 93.7% accuracy, whereas Random Forest obtained 85.3% [14]. These differences suggest that the optimal classifier is not determined solely by the application domain, but can also be influenced by dataset composition, sentiment labeling, class distribution, preprocessing, feature representation, and experimental configuration. Therefore, the present result should be interpreted as evidence that Random Forest was the most suitable model under the experimental conditions of this study, rather than as a universally superior classifier.

TABLE 4. Comparison of the Present Study with Previous Sentiment Analysis Studies

Study	Dataset / Application	Algorithms Compared	Best Result	Comparison with This Study
[6]	14,296 Spotify reviews from Google Play Store and App Store	NB, SVM, RF	SVM: 87.4% accuracy; 0.875 F1	SVM was superior in their setting, whereas RF was the best model in the present study
[14]	10,000 Indonesian Spotify reviews	SVM, Neural Network, RF	SVM & NN: 93.7% accuracy; RF: 85.3%	The present RF accuracy (84.58%) is slightly below their RF result, while their SVM performed substantially better
[17]	Spotify reviews	VADER-based sentiment analysis	77% accuracy reported	The present study evaluates six supervised ML classifiers and provides a broader algorithmic comparison
This study	**4,605 Spotify reviews, binary sentiment**	**NB, SVM, LR, RF, DT, KNN**	**RF: 84.58% accuracy; 81.71% F1**	**Compares six classical ML algorithms under a common TF-IDF + SMOTE framework**

The comparison also shows in Table 4 that the present study extends previous work beyond simply identifying the classifier with the highest accuracy. The use of SMOTE only on the training data improved the F1-score of all six models, with the largest gains observed for KNN (+2.25 percentage points) and Decision Tree (+2.12 percentage points). This result is consistent with previous research showing that class imbalance can bias sentiment classifiers toward the majority class and that SMOTE can improve recognition of minority instances [15]. In addition, the present study incorporates Learning Curve and SHAP analyses, allowing model generalization and feature-level decision behavior to be examined alongside conventional performance metrics. Consequently, the

Published by Maheswari Publisher
Creation disseminated under Creative Commons Attribution 4.0 International License

Vol. 2 No.1 November 2026
E-ISSN: 3132-5115
DOI:10.65780/bima.v2i1.41

main contribution is not merely the selection of Random Forest as the best-performing classifier, but the integration of comparative evaluation, imbalance handling, generalization analysis, and model interpretability within a single experimental framework.

5. CONCLUSIONS

Based on the results of this study on sentiment analysis of Spotify application user reviews on the Google Play Store using six Machine Learning algorithms, it can be concluded that the sentiment classification process was successfully carried out through the stages of data collection using the scraping technique, rating-based labeling, text preprocessing, feature extraction using TF-IDF, data balancing with SMOTE, as well as the training and evaluation of six classification algorithms. The comparison results using the accuracy, precision, recall, and F1-score metrics indicate that each algorithm has different performance characteristics in classifying user review sentiment. Among all the algorithms tested, Random Forest demonstrated the best and most consistent performance, supported by the model evaluation results, the characteristics of the algorithm, and its ability to recognize patterns in the Spotify review dataset. Although the Learning Curve analysis indicated the presence of overfitting, the Random Forest model was still able to maintain good generalization capability on the testing data and is therefore considered the best algorithm in this study.

REFERENCES

[1] B. J. Hracs and J. Webster, 'From selling songs to engineering experiences: exploring the competitive strategies of music streaming platforms', *Journal of Cultural Economy*, vol. 14, no. 2, pp. 240–257, Mar. 2021, doi: 10.1080/17530350.2020.1819374.

[2] P. Gupta and S. Srivastava, 'Exploring customer attitude towards neo-banking apps: a thematic analysis using Google Play and Apple App Store reviews', *Qualitative Research in Financial Markets*, vol. 18, no. 4, pp. 987–1013, Jun. 2026, doi: 10.1108/QRFM-07-2024-0198.

[3] E. Guzman and W. Maalej, 'How Do Users Like This Feature? A Fine Grained Sentiment Analysis of App Reviews', in *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, Aug. 2014, pp. 153–162. doi: 10.1109/RE.2014.6912257.

[4] J. Dąbrowski, E. Letier, A. Perini, and A. Susi, 'Analysing app reviews for software engineering: a systematic literature review', *Empir Software Eng*, vol. 27, no. 2, p. 43, Jan. 2022, doi: 10.1007/s10664-021-10065-7.

[5] P. H. C. Samanmali and R. a. H. M. Rupasingha, 'Sentiment analysis on google play store app users' reviews based on deep learning approach.', *Multimedia Tools & Applications*, vol. 83, no. 36, p. 84425, Nov. 2024, doi: 10.1007/s11042-024-19185-w.

[6] E. D. Madyatmadja, F. Felix, I. G. K. Edrick, J. W. Indarto, and D. J. M. Sembiring, 'Harmonizing sentiments: Analyzing user reviews of Spotify through sentiment analysis', *JIPD*, vol. 8, no. 9, p. 7101, Sep. 2024, doi: 10.24294/jipd.v8i9.7101.

[7] G. Lu, S. Qu, and Y. Chen, 'Understanding user experience for mobile applications: a systematic literature review', *Discov Appl Sci*, vol. 7, no. 6, p. 587, Jun. 2025, doi: 10.1007/s42452-025-07170-3.

[8] N. Alsaleh, R. Alnanih, and N. Alowidi, 'Hybrid Deep Learning Approach for Automating App Review Classification: Advancing Usability Metrics Classification with an Aspect-Based Sentiment Analysis Framework', *Computers, Materials & Continua*, vol. 82, no. 1, pp. 949–976, Jan. 2025, doi: 10.32604/cmc.2024.059351.

[9] M. Wankhade, A. C. S. Rao, and C. Kulkarni, 'A survey on sentiment analysis methods, applications, and challenges', *Artif Intell Rev*, vol. 55, no. 7, pp. 5731–5780, Oct. 2022, doi: 10.1007/s10462-022-10144-1.

[10] Y. Jin, K. Cheng, X. Wang, and L. Cai, 'A Review of Text Sentiment Analysis Methods and Applications', *FBEM*, vol. 10, no. 1, pp. 58–64, Jul. 2023, doi: 10.54097/fbem.v10i1.10171.

[11] K. Naithani and Y. P. Raiwani, 'Realization of natural language processing and machine learning approaches for text-based sentiment analysis', *Expert Systems*, vol. 40, no. 5, p. e13114, Jun. 2023, doi: 10.1111/exsy.13114.

[12] S. F. Taskiran, B. Turkoglu, E. Kaya, and T. Asuroglu, 'A comprehensive evaluation of oversampling techniques for enhancing text classification performance', *Sci Rep*, vol. 15, no. 1, p. 21631, Jul. 2025, doi: 10.1038/s41598-025-05791-7.

[13] M. A. Hadi and F. H. Fard, 'Evaluating pre-trained models for user feedback analysis in software engineering: a study on classification of app-reviews', *Empir Software Eng*, vol. 28, no. 4, p. 88, May 2023, doi: 10.1007/s10664-023-10314-x.

[14] F. Olivian, S. Bariyah, G. C. Budiyanto, R. Annisa, L. A. Fitriana, and W. S. Dharmawan, 'Performance Evaluation of Machine Learning Algorithms in Sentiment Analysis of Spotify Reviews', *j. of artif. intell. and eng. appl.*, vol. 5, no. 3, pp. 4150–4159, Jun. 2026, doi: 10.59934/jaiea.v5i3.2362.

[15] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, 'SMOTE: Synthetic Minority Over-sampling Technique', *jair*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.

[16] S. M. Lundberg and S.-I. Lee, 'A Unified Approach to Interpreting Model Predictions', in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2017. Accessed: Oct. 20, 2026. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

[17] Nur Hasanah Habibillah, Rahma Nurlaela, Riska Dwi Lestari, Faradina Mosyarrina, and Nabilla Najma Tsuraya, 'Analisis Sentimen Terhadap Aplikasi Spotify Berdasarkan Ulasan di Google Play Store', *JIITE*, vol. 2, no. 1, pp. 300–306, Apr. 2025, doi: 10.63547/jiite.v2i1.47.

[18] T. Setyani, K. Sari, H. N. Heidy, and R. R. Suryono, 'Analisis Sentimen Pengguna Aplikasi Jamsostek Mobile Berdasarkan Ulasan Google Play Store Menggunakan Algoritma Support Vector Machine dan Naive Bayes: Sentiment Analysis of Jamsostek Mobile Application Reviews on Google Play Store Using Support Vector Machine and Naive Bayes Algorithms', *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 6, no. 1, pp. 373–384, Feb. 2026, doi: 10.57152/malcom.v6i1.2526.

Published by Maheswari Publisher
Creation disseminated under Creative Commons Attribution 4.0 International License