ISSN: 2623-1662 (online), 2460-092X (printed)
DOI: https://doi.org/10.19109/jusifo.v11i1.27925
Vol. 11, No. 1 (2025), p.43-52

JUSIFO
JURNAL SISTEM INFORMASI

Research Article [OPEN ACCESS]

Sentiment Analysis on WeTV Application Reviews Using Naïve Bayes: A Study of Preprocessing, Balancing, and Model Performance

Wilis Brawijaya, Khothibul Umam*, Siti Nur'aini, Maya Rini Handayani

ABSTRACT

This study investigates the application of the Naïve Bayes classification algorithm for sentiment analysis of user-generated reviews on the WeTV application available on the Google Play Store. A structured methodology was employed, consisting of data scraping, sentiment labeling based on heuristics, multi-stage preprocessing, class balancing using Synthetic Minority Over-sampling Technique (SMOTE), and performance evaluation through standard metrics. Prior to balancing, the model exhibited strong performance on the dominant class but underperformed on the minority class. The introduction of SMOTE led to improved F1-scores, particularly for positive sentiment, increasing from 61% to 64%, while maintaining overall accuracy around 71%. These findings confirm that Naïve Bayes, when supported by effective preprocessing and data balancing, can deliver robust and interpretable classification results in text mining tasks. This research contributes to the growing literature on machine learning for opinion mining and provides practical implications for developers aiming to extract structured insights from large-scale user reviews.

Keyword: Naïve bayes, sentiment analysis, smote balancing

Received: April 18, 2025; Revised: May 17, 2025; Accepted: June 29, 2025

Corresponding Author: Khothibul Umam, Department of Information Technology, Universitas Islam Negeri Walisongo Semarang, Indonesia, khothibul_umam@walisongo.ac.id
Authors: Wilis Brawijaya, Department of Information Technology, Universitas Islam Negeri Walisongo Semarang, Indonesia, 2208096069@student.walisongo.ac.id; Siti Nur'aini, Department of Information Technology, Universitas Islam Negeri Walisongo Semarang, Indonesia, siti_nuraini@walisongo.ac.id; Maya Rini Handayani, Department of Information Technology, Universitas Islam Negeri Walisongo Semarang, Indonesia, maya@walisongo.ac.id

1. INTRODUCTION

The rapid advancement of information technology has significantly transformed content consumption behavior, particularly through digital platforms. One of the most impactful innovations in this domain is the development of Video on Demand (VOD) services, which provide flexible access to various entertainment content such as films, series, and online dramas. In Indonesia, WeTV has emerged as a prominent VOD platform, with over 100 million downloads and more than 500,000 user reviews on the Google Play Store. The abundance of user-generated content presents an opportunity to understand user sentiment at scale; however, it also introduces challenges in extracting structured insights from unstructured textual data.

A critical issue faced by application developers is the difficulty in manually analyzing large volumes of diverse user reviews, which often contain informal language, abbreviations, and subjective expressions. These reviews not only reflect user satisfaction but also include complaints and suggestions related to service quality and application features. Manual sentiment classification is not only inefficient but also susceptible to bias (Adib et al., 2024). To address this, sentiment analysis has become a standard computational approach for identifying the polarity of user opinions. Among various classification algorithms, Naïve Bayes has shown favorable performance in multiple studies, surpassing Support Vector Machine (SVM) and Random Forest in classification accuracy (Madyatmadja et al., 2024). Prior research

also supports its application in similar contexts, with reported accuracies of 80.28% and 73% for the Video and Threads applications, respectively (Nurzaman et al., 2024; Siregar et al., 2024).

Although several studies have examined sentiment analysis on WeTV application reviews (Allorerung & Rismayani, 2023; Kulsum et al., 2022; Lestari et al., 2023; Mareby & Desanti, 2024), many of them either focus on algorithm comparison without a structured preprocessing pipeline or lack a comprehensive evaluation of model performance and class imbalance handling. These elements are particularly relevant in cases involving noisy data, where the proportion of positive and negative reviews is uneven. Moreover, inadequate preprocessing and imbalanced datasets can reduce model reliability, resulting in skewed classification outcomes (Ali et al., 2019; Felix & Lee, 2019; Hussein et al., 2019; Wojciechowski & Wilk, 2017).

This study addresses the identified gap by applying the Naïve Bayes algorithm to classify user sentiment on the WeTV application, incorporating systematic text preprocessing and data balancing using the Synthetic Minority Oversampling Technique (SMOTE). Drawing from previous findings (Friadi & Kurniawan, 2024; Kaburuan et al., 2022; Razaq et al., 2023), this research validates the capability of Naïve Bayes in handling real-world textual data and assesses its classification performance based on established evaluation metrics. The outcomes provide insights that support the development of automated sentiment analysis systems and enhance understanding of user feedback in large-scale digital platforms.

2. MATERIALS AND METHODS

2.1 Materials

The dataset used in this study comprises 2,000 user reviews of the WeTV application collected from the Google Play Store. The reviews were retrieved using a Python-based web scraping tool operated through Google Colab. Each data entry includes the username, review date, star rating, and review content. The data collection period spanned from October 2024 to March 2025.

Sentiment labeling was conducted using the Natural Language Toolkit (NLTK) library. A rating-based classification approach was applied, in which reviews with 3–5 stars were labeled as positive, while those with 1–2 stars were labeled as negative (Güner et al., 2019). The initial dataset was imbalanced, containing 57.85% negative reviews and 42.15% positive reviews. This class imbalance was addressed during preprocessing using the Synthetic Minority Oversampling Technique (SMOTE), which synthetically generates samples for the minority class (Elreedy et al., 2024).

2.2 Methods

The methodological design of this study consists of six sequential stages as illustrated in Figure 1: data collection, sentiment labeling, text preprocessing, sentiment classification using the Naïve Bayes algorithm, and model evaluation.

Figure 1. Research methodology

The dataset was obtained through an automated scraping process of publicly available user reviews on the Google Play Store. This approach ensured a consistent and systematic data acquisition workflow. Sentiment labeling was conducted based on a heuristic, rating-based classification scheme, where user ratings were automatically mapped to sentiment categories. This method has been validated in previous studies and shown to be reliable for large-scale opinion mining tasks (Güner et al., 2019).

Text preprocessing was implemented through a five-stage pipeline, as suggested by established best practices in natural language processing (Alasadi & Bhaya, 2017; Alexandropoulos et al., 2019; García et al., 2015; Saleem et al., 2014). The process began with case folding to convert all letters into lowercase, ensuring lexical consistency. Noise cleaning was applied to eliminate irrelevant characters such

as punctuation marks, digits, symbols, and embedded hyperlinks. Subsequently, stopwords—frequently occurring words with minimal semantic contribution—were removed. Tokenization segmented each review into individual tokens or word units, and stemming transformed each word into its root form using an Indonesian language stemmer.

Following preprocessing, the dataset was split into training and testing subsets with an 80:20 ratio, resulting in 1,600 entries for training and 400 for testing. This division allowed for robust evaluation while minimizing overfitting (Musu et al., 2021). The preprocessed textual data was transformed into numerical feature vectors using the Term Frequency–Inverse Document Frequency (TF-IDF) method. This transformation is essential to enable the Naïve Bayes algorithm to process textual inputs by converting them into structured numerical representations that reflect the relative importance of each term within individual reviews and across the corpus. To prevent data leakage and ensure unbiased model evaluation, the TF-IDF vectorization was performed after the dataset was split. The vectorizer was fitted to the training data using *"fit_transform"* and subsequently applied to the test data using *"transform"*, thereby ensuring that no information from the test set influenced the training process.

Given the initial class imbalance, the Synthetic Minority Oversampling Technique (SMOTE) was applied exclusively to the training dataset. SMOTE generates synthetic samples for the minority class, creating a more balanced class distribution and enhancing the generalizability of the model. Importantly, data balancing was not performed on the testing set to preserve its integrity for unbiased evaluation.

The classification task was performed using the Naïve Bayes algorithm, which applies probabilistic computations to estimate the most likely class membership for each input based on observed features (Yang, 2018). The model was trained on the balanced training data and evaluated using the unaltered testing subset. Performance assessment was carried out using standard evaluation metrics derived from the confusion matrix, including accuracy, precision, recall, and F1-score. These metrics were calculated separately for each sentiment class to provide a comprehensive assessment of the model's effectiveness in classifying user-generated content with varied lexical and emotional characteristics.

3. RESULTS AND DISCUSSION

3.1 Data Collection

The first step of this study involved collecting 2,000 user reviews from the WeTV application on the Google Play Store using the Python programming language via Google Colab. The reviews were extracted automatically using a scraping technique and stored in CSV format. This process was conducted over a six-month period from October 2024 to March 2025, ensuring that the dataset was up-to-date and representative of user sentiment (Figure 2).

```
Installing collected packages: google-play-scraper
Successfully installed google-play-scraper-1.2.7
```

```
[ ] from google_play_scraper import app

    import pandas as pd

    import numpy as np
```

```
from google_play_scraper import Sort, reviews_all
import pandas as pd

# ID aplikasi yang ingin di-scrape
app_id = 'com.tencent.qqlive118n'

# Mendapatkan semua ulasan terbaru
reviews = reviews_all(
    app_id,
    sleep_milliseconds=0,  # Delay antara permintaan
    lang='id',  # Bahasa ulasan yang diinginkan
    country='id',  # Negara ulasan yang diinginkan
)

# Konversi hasil ke DataFrame
reviews_df = pd.DataFrame(reviews)

# Tampilkan 2000 ulasan pertama
reviews_df = reviews_df.head(2000)

# Memilih hanya atribut yang diinginkan
filtered_reviews = reviews_df[['userName', 'score', 'at', 'content']]

# Simpan ke file CSV
filtered_reviews.to_csv('wetv.csv', index=False, encoding='utf-8')

print(f"Jumlah ulasan yang diambil: {len(filtered_reviews)}")
```

```
Jumlah ulasan yang diambil: 2000
```

Figure 2. Data collection process script

3.2 Sentiment Labeling

Following data acquisition, sentiment labeling was performed using the Natural Language Toolkit (NLTK). Reviews were categorized into positive or negative sentiment classes based on the associated star ratings. Specifically, reviews rated 3, 4, or 5 stars were labeled as positive, while those with 1 or 2 stars were labeled as negative (Figure 3).

```
[ ] def pelabelan(score):
      if score < 3:
        return 'Negatif'
      elif score >=3 :
        return 'Positif'
    df['Label'] = df ['score'].apply(pelabelan)
    df.head(150)
```

	userName	score	at	content	Label
0	Pengguna Google	4	2025-03-25 12:25:20	iklannya kurangin lah parah iklannya	Positif
1	Pengguna Google	1	2025-03-25 11:27:00	padahl sudah di hapus akun metode pembayaran d...	Negatif
2	Pengguna Google	1	2025-03-25 10:39:37	iklannya banyak bgt, aku mau nonton kelanjutan...	Negatif
3	Pengguna Google	3	2025-03-25 08:57:19	saldo saya terpotong sendiri	Positif
4	Pengguna Google	5	2025-03-25 07:52:14	enak dan bagus bagus film nya 🤗	Positif
...	...	...	...	...	...
145	Pengguna Google	1	2025-03-15 14:44:32	WOIILAHH TOLONG KALI INII GA BISA FULLSCREEN, ...	Negatif
146	Pengguna Google	5	2025-03-15 14:19:48	Aplikasi yang sangat bagus	Positif
147	Pengguna Google	1	2025-03-15 13:28:59	Tiap bulan berlangganan VIP tapi baru bulan in...	Negatif
148	Pengguna Google	4	2025-03-15 13:19:18	pakai wetv karena dpt gratisan vip sebulan dar...	Positif
149	Pengguna Google	2	2025-03-15 11:12:21	Saya sebagai pengguna premium sangat terganggu...	Negatif

150 rows × 5 columns

Figure 3. Labeling process

The result of the labeling showed that 42.15% of reviews were classified as positive, while 57.85% were negative. The sentiment distribution is illustrated in Figure 4.

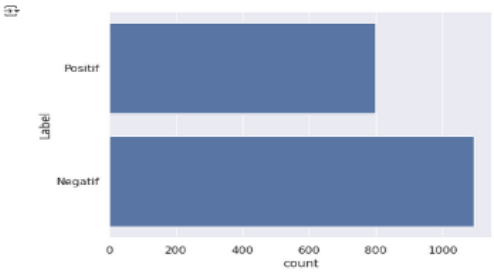

Figure 4. Labeling proportion result

3.3 Text Preprocessing

Text preprocessing was conducted to enhance data quality and prepare it for classification. This stage included five sequential steps:

1. Case folding and cleaning

   All characters in the review texts were converted to lowercase to ensure uniformity (case folding). Unnecessary characters such as punctuation, numbers, symbols, and URLs were then removed (cleaning). The resulting text was more consistent and structured for further processing. The outcome of this stage is summarized in Table 1.

Table 1. Case folding and cleaning results

Username	Score	At	Content	Text_Lower	Text_Clean
Pengguna Google	5	2025-03-25 07:52:14	enak dan bagus bagus film nya 🤗	enak dan bagus bagus film nya 🤗	enak dan bagus bagus film nya
Pengguna Google	3	2025-03-25 06:49:20	udah VIP tapi tiap nonton HD seperti putus-put...	udah vip tapi tiap nonton hd seperti putus-put...	udah vip tapi tiap nonton hd seperti putusputu...
Pengguna Google	2	2025-03-25 06:17:58	emng ini seru ga?	emng ini seru ga?	emng ini seru ga
...	...	...	...	...	...

2. Stopword removal

Common non-informative words such as “dan”, “saya”, and “dari” were removed to reduce noise in the data and retain only semantically important tokens. Examples of the results after stopword elimination are shown in Table 2.

Table 2. Stopword removal results

Content_clean	Text_stopwords
enak dan bagus bagus film nya	enak bagus bagus film nya
udah vip tapi tiap nonton hd seperti putusputu...	udah vip nonton hd putusputus video nya sinyal...
emng ini seru ga	emng seru ga
…	…

3. Tokenization

Tokenization segmented the cleaned texts into individual words or tokens for subsequent analysis. The tokenized version of the cleaned text is presented in Table 3.

Table 3. Tokenization results

Text_stopwords	Tokens
enak bagus bagus film nya	[enak, bagus, bagus, film, nya]
udah vip nonton hd putusputus video nya sinyal...	[udah, vip, nonton, hd, putusputus, video, nya, ...]
emng seru ga	[emng, seru, ga]
…	…

4. Stemming

Words were reduced to their base forms using an Indonesian language stemmer. This process enhanced consistency in sentiment classification. The results of this stemming process are detailed in Table 4.

Table 4. Stemming results

Tokens	Stemmed_tokens
[enak, bagus, bagus, film, nya]	[enak, bagus, bagus, film, nya]
[udah, vip, nonton, hd, putusputus, video, nya, ...]	[udah, vip, nonton, hd, putusputus, video, nya, ...]
[emng, seru, ga]	[emng, seru, ga]
…	…

Following the stemming process, the cleaned tokens were transformed into numerical feature vectors using the Term Frequency–Inverse Document Frequency (TF-IDF) method. This transformation quantifies the importance of each term relative to both the individual document and the overall corpus. To ensure unbiased model evaluation and avoid data leakage, the TF-IDF vectorizer was trained exclusively on the training data using *“fit_transform”*, and subsequently applied to the test data using *“transform”*. The resulting TF-IDF matrix served as the input for the classification algorithm.

3.4 Evaluation Before Data Balancing

Prior to applying the SMOTE technique, the sentiment dataset exhibited a significant class imbalance, with negative sentiment dominating the reviews. This imbalance could potentially bias the model and hinder its ability to accurately classify underrepresented sentiment categories (Figure 4). The performance

of the Naïve Bayes classifier was first evaluated using the original, unbalanced dataset. The confusion matrix in Figure 5 presents the model's classification outcomes for both training and testing subsets.

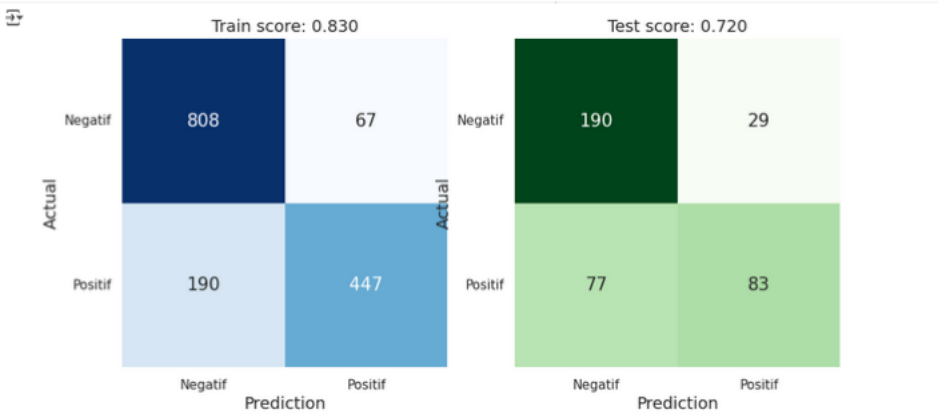

Figure 5. Confusion matrix plot before smote

Metrics Calculation (Training):

- $\boldsymbol{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)} = \frac{(447+808)}{(447+808+67+190)} = 83\%$
- $\boldsymbol{Precision\ (Negative)} = \frac{(TN)}{(TN+FN)} = \frac{(808)}{(808+190)} = 81\%$
- $\boldsymbol{Recall\ (Negative)} = \frac{(TN)}{(TN+FP)} = \frac{(808)}{(808+67)} = 92\%$
- $\boldsymbol{F1-Score\ (Negative)} = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} = 86\%$
- $\boldsymbol{Precision\ (Positive)} = \frac{(TP)}{(TP+FP)} = \frac{(447)}{(447+67)} = 70\%$
- $\boldsymbol{Recall\ (Positive)} = \frac{(TP)}{(TP+FN)} = \frac{(808)}{(808+190)} = 87\%$
- $\boldsymbol{F1-Score\ (Positive)} = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} = 78\%$

Metrics Calculation (Testing):

- $\boldsymbol{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)} = \frac{(83+190)}{(83+190+29+77)} = 72\%$
- $\boldsymbol{Precision\ (Negative)} = \frac{(TN)}{(TN+FN)} = \frac{(190)}{(190+77)} = 71\%$
- $\boldsymbol{Recall\ (Negative)} = \frac{(TN)}{(TN+FP)} = \frac{(190)}{(190+29)} = 87\%$
- $\boldsymbol{F1-Score\ (Negative)} = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} = 78\%$
- $\boldsymbol{Precision\ (Positive)} = \frac{(TP)}{(TP+FP)} = \frac{(83)}{(83+29)} = 74\%$
- $\boldsymbol{Recall\ (Positive)} = \frac{(TP)}{(TP+FN)} = \frac{(83)}{(83+77)} = 52\%$
- $\boldsymbol{F1-Score\ (Positive)} = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} = 61\%$

3.5 Evaluation After Data Balancing Using SMOTE

To mitigate the effect of class imbalance, the SMOTE (Synthetic Minority Over-sampling Technique) method was applied to the training data. This technique synthetically increased the number of samples in the minority class while preserving the original data in the majority class, resulting in a more balanced dataset as shown in Figure 6. Subsequent evaluation of the Naïve Bayes classifier using the balanced dataset demonstrated improved classification performance. The resulting confusion matrix is presented in Figure 4.

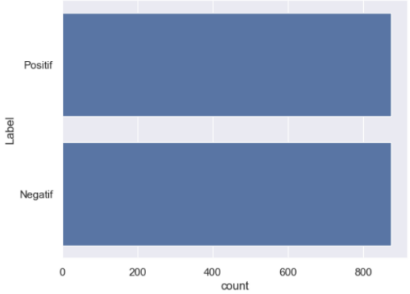

Figure 6. Data distribution after smote application

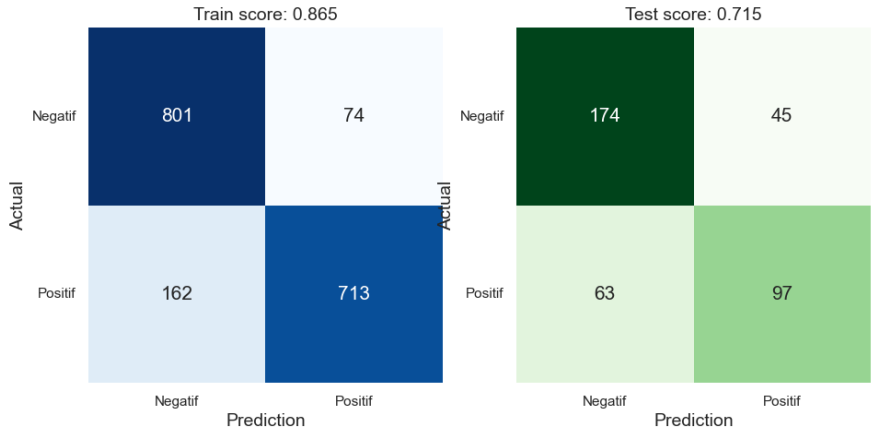

Figure 7. Confusion matrix plot after smote

Metrics Calculation (Training):

- $\boldsymbol{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)} = \frac{(713+801)}{(713+801+74+162)} = 86\%$
- $\boldsymbol{Precision\ (Negative)} = \frac{(TN)}{(TN+FN)} = \frac{(801)}{(801+162)} = 83\%$
- $\boldsymbol{Recall\ (Negative)} = \frac{(TN)}{(TN+FP)} = \frac{(801)}{(801+74)} = 91\%$
- $\boldsymbol{F1-Score\ (Negative)} = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} = 86\%$
- $\boldsymbol{Precision\ (Positive)} = \frac{(TP)}{(TP+FP)} = \frac{(713)}{(713+74)} = 90\%$
- $\boldsymbol{Recall\ (Positive)} = \frac{(TP)}{(TP+FN)} = \frac{(713)}{(713+162)} = 81\%$
- $\boldsymbol{F1-Score\ (Positive)} = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} = 85\%$

Metrics Calculation (Testing):

- $\boldsymbol{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)} = \frac{(97+174)}{(97+174+45+63)} = 71\%$
- $\boldsymbol{Precision\ (Negative)} = \frac{(TN)}{(TN+FN)} = \frac{(174)}{(174+63)} = 73\%$
- $\boldsymbol{Recall\ (Negative)} = \frac{(TN)}{(TN+FP)} = \frac{(174)}{(174+45)} = 79\%$
- $\boldsymbol{F1-Score\ (Negative)} = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} = 76\%$
- $\boldsymbol{Precision\ (Positive)} = \frac{(TP)}{(TP+FP)} = \frac{(97)}{(97+45)} = 68\%$
- $\boldsymbol{Recall\ (Positive)} = \frac{(TP)}{(TP+FN)} = \frac{(97)}{(97+63)} = 61\%$
- $\boldsymbol{F1-Score\ (Positive)} = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} = 64\%$

3.6 Discussion

The findings of this study demonstrate that the application of the Naïve Bayes algorithm, combined with a structured preprocessing pipeline and the SMOTE balancing technique, yields promising results in classifying user sentiment in WeTV application reviews. Prior to data balancing, the classification model showed limitations, particularly in identifying underrepresented classes. As evidenced by the confusion matrix, the positive sentiment class exhibited a relatively low recall, indicating the model's difficulty in capturing minority class patterns in imbalanced data.

The application of SMOTE significantly improved the model's ability to generalize across classes by synthesizing additional data for the underrepresented sentiment category. The improvement in evaluation metrics—such as accuracy increasing from 83% to 86% on training data and stable performance at 71–72% on test data—confirms that balancing techniques are crucial for enhancing classification robustness. Furthermore, the F1-score for the positive class increased from 0.78 to 0.85, reflecting improved precision and recall in detecting positive sentiments.

This study also reinforces the practicality of Naïve Bayes for sentiment analysis tasks involving user-generated reviews. Despite its simplifying assumption of feature independence, the algorithm showed competitive performance and computational efficiency, aligning with findings from related research (e.g., Madyatmadja et al., 2024; Siregar et al., 2024). These results validate that, when paired with proper data preprocessing and balancing, Naïve Bayes remains a viable and effective classifier for real-world opinion mining applications.

In addition to the improvements yielded by the SMOTE balancing technique, this study reveals nuanced insights regarding model behavior in the presence of imbalanced and preprocessed text data. Although the overall testing accuracy remained relatively stable after SMOTE application (from 72% to 71%), the notable increase in F1-score for the positive sentiment class (from 61% to 64%) indicates a more equitable classification performance. This highlights that improvements in minority class performance can be achieved without compromising the predictive capability for the dominant class. Thus, SMOTE proves to be not merely a balancing tool, but a mechanism for enhancing model generalization in sentiment classification tasks.

Moreover, the stepwise preprocessing pipeline—consisting of case folding, noise removal, stopword elimination, tokenization, and stemming—was found to be essential in preparing the textual data for effective classification. Each step contributed to reducing noise and linguistic redundancy, enabling the Naïve Bayes classifier to operate on a cleaner and more representative feature space. The importance of preprocessing is especially pronounced in sentiment analysis involving informal and user-generated content, which is often characterized by spelling variations, colloquialisms, and emoji usage.

Lastly, this research reinforces the methodological value of lightweight and interpretable models such as Naïve Bayes in real-world sentiment analysis applications. While more complex models like deep neural networks may offer higher accuracy under certain conditions, their interpretability and computational cost remain barriers for many practical scenarios. The results of this study suggest that, when combined with proper preprocessing and balancing, Naïve Bayes can achieve performance levels that are not only adequate but also efficient and explainable.

4. CONCLUSION

This research contributes to the field of sentiment analysis by investigating the application of the Naïve Bayes classification algorithm to user reviews of the WeTV application on the Google Play Store. The methodology integrated a multi-step preprocessing pipeline, class balancing using SMOTE, and systematic model evaluation through standard performance metrics. The study reveals that while the baseline Naïve Bayes model performs reasonably well on imbalanced data, its effectiveness substantially increases when supported by balanced training sets and optimized preprocessing.

The results demonstrate an improved F1-score and accuracy in detecting both positive and negative sentiment after data balancing, validating the importance of addressing class imbalance in text classification tasks. These insights offer practical implications for developers and analysts aiming to derive structured insights from unstructured user-generated data, particularly in entertainment platforms.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

Adib, khoirul, Handayani, M. R., Yuniarti, W. D., & Umam, K. (2024). Opini publik pasca-pemilihan presiden: eksplorasi analisis sentimen media sosial x menggunakan svm. *Sintech (Science and Information Technology) Journal*, *7*(2), 80–91. https://doi.org/10.31598/sintechjournal.v7i2.1581

Alasadi, S. A., & Bhaya, W. S. (2017). Review of data preprocessing techniques in data mining. *Journal of Engineering and Applied Sciences*, *12*(16), 4102–4107. https://doi.org/10.3923/JEASCI.2017.4102.4107

Alexandropoulos, S. A. N., Kotsiantis, S. B., & Vrahatis, M. N. (2019). Data preprocessing in predictive data mining. *The Knowledge Engineering Review*, *34*, e1. https://doi.org/10.1017/S026988891800036X

Ali, H., Salleh, M. N. M., Hussain, K., Ahmad, A., Ullah, A., Muhammad, A., Naseem, R., & Khan, M. (2019). A review on data preprocessing methods for class imbalance problem. *International Journal of Engineering and Technology*, *8*(3), 390–397.

Allorerung, P. P., & Rismayani, R. (2023). Sentiment analysis on wetv app reviews on google play store using nbc and svm algorithms. *Sistemasi*, *12*(2), 404–414. https://doi.org/10.32520/STMSI.V12I2.2518

Elreedy, D., Atiya, A. F., & Kamalov, F. (2024). A theoretical distribution analysis of synthetic minority oversampling technique (smote) for imbalanced learning. *Machine Learning*, *113*(7), 4903–4923. https://doi.org/10.1007/S10994-022-06296-4

Felix, E. A., & Lee, S. P. (2019). Systematic literature review of preprocessing techniques for imbalanced data. *IET Software*, *13*(6), 479–496. https://doi.org/10.1049/IET-SEN.2018.5193

Friadi, J., & Kurniawan, D. E. (2024). Analisis sentimen ulasan wisatawan terhadap alun-alun kota batam: perbandingan kinerja metode naive bayes dan support vector machine. *Jurnal Sistem Informasi Bisnis*, *14*(4), 403–407. https://doi.org/10.21456/VOL14ISS4PP403-407

García, S., Luengo, J., & Herrera, F. (2015). *Data preprocessing in data mining* (Vol. 72). Springer International Publishing. https://doi.org/10.1007/978-3-319-10247-4

Güner, L., Coyne, E., & Smit, J. (2019). *Sentiment analysis for amazon.com reviews*.

Hussein, A. S., Li, T., Yohannese, C. W., & Bashir, K. (2019). A-smote: a new preprocessing approach for highly imbalanced datasets by improving smote. *International Journal of Computational Intelligence Systems*, *12*(2), 1412–1422. https://doi.org/10.2991/ijcis.d.191114.002

Kaburuan, E. R., Sari, Y. S., & Agustina, I. (2022). Sentiment analysis on product reviews from shopee marketplace using the naïve bayes classifier. *Lontar Komputer: Jurnal Ilmiah Teknologi Informasi*, *13*(3), 150–159. https://doi.org/10.24843/LKJITI.2022.V13.I03.P02

Kulsum, U., Jajuli, M., & Sulistiyowati, N. (2022). Analisis sentimen aplikasi wetv di google play store menggunakan algoritma support vector machine. *Journal of Applied Informatics and Computing*, *6*(2), 205–212. https://doi.org/10.30871/JAIC.V6I2.4802

Lestari, N., Haerani, E., & Candra, R. M. (2023). Analisa sentimen ulasan aplikasi wetv untuk peningkatan layanan menggunakan metode naïve bayes. *Journal of Information System Research (JOSH)*, *4*(3), 874–882. https://doi.org/10.47065/JOSH.V4I3.3355

Madyatmadja, E. D., Candra, H., Nathaniel, J., Jonathan, M. R., & Rudy, R. (2024). Sentiment analysis on user reviews of threads applications in indonesia. *Journal Europeen Des Systemes Automatises*, *57*(4), 1165–1171. https://doi.org/10.18280/JESA.570423

Mareby, Y. S. P., & Desanti, R. I. (2024). Exploring wetv application with naïve bayes, decision tree, and random forest classifiers for sentiment analysis. *2024 International Visualization, Informatics and Technology Conference, IVIT 2024*, 35–42. https://doi.org/10.1109/IVIT62102.2024.10692731

Musu, W., Ibrahim, A., & Heriadi, H. (2021). Pengaruh komposisi data training dan testing terhadap akurasi algoritma c4.5. *Sisiti: Seminar Ilmiah Sistem Informasi Dan Teknologi Informasi*, *10*(1), 186–195. https://doi.org/10.36774/SISITI.V10I1.802

Nurzaman, N., Suarna, N., & Prihartono, W. (2024). Analisis sentimen ulasan aplikasi threads di google playstore menggunakan algoritma naïve bayes. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *8*(1), 967–974. https://doi.org/10.36040/JATI.V8I1.8708

Razaq, M. T., Nurjanah, D., & Nurrahmi, H. (2023). Analisis sentimen review film menggunakan naive bayes classifier dengan fitur tf-idf. *EProceedings of Engineering*, *10*(2), 45–49. https://doi.org/10.5120/IJCA2017916005

Saleem, A., Asif, K. H., Ali, A., Awan, S. M., & Alghamdi, M. A. (2014). Pre-processing methods of data mining. *Proceedings - 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, UCC 2014*, 451–456. https://doi.org/10.1109/UCC.2014.57

Siregar, M. Y., Wiranata, A. D., & Saputra, R. A. (2024). Analisis Sentimen Pada Ulasan Pengguna Aplikasi Streaming Vidio Menggunakan Metode Naïve Bayes. *KLIK: Kajian Ilmiah Informatika Dan Komputer*, *4*(5), 2419–2429. https://doi.org/10.30865/KLIK.V4I5.1787

Wojciechowski, S., & Wilk, S. (2017). Difficulty factors and preprocessing in imbalanced data sets: an experimental study on artificial data. *Foundations of Computing and Decision Sciences*, *42*(2), 149–176. https://doi.org/10.1515/FCDS-2017-0007

Yang, F. J. (2018). An implementation of naive bayes classifier. *Proceedings - 2018 International Conference on Computational Science and Computational Intelligence, CSCI 2018*, 301–306. https://doi.org/10.1109/CSCI46756.2018.00065