J-INTECH (Journal of Information and Technology)
Accredited Sinta 4 Ministry of Higher Education, Science and Technology Republic of Indonesia SK No. 10/C/C3/DT.05.00/2025
E-ISSN: 2580-720X || P-ISSN: 2303-1425

Application of K-Nearest Neighbor Algorithm for Sentiment Analysis on Free Fire Online Game Based on Google Play Store Reviews

Dimas Raya[1]*, Amelia Yusnita[2], Ivan Haristyawan[3]

[1,2,3]STMIK Widya Cipta Dharma, Informatics Engineering, Jl. M. Yamin No.25, Gn. Kelua, Kec. Samarinda Ulu, Kota Samarinda, Kalimantan Timur 75123, Indonesia

Keywords

Free Fire; Google Play Store; K-Nearest Neighbor; Sentiment Analysis; TF-IDF.

Corresponding Author:
2143037@wicida.ac.id

Abstract

The swift expansion of the digital gaming sector, especially online games like Free Fire, has produced extensive user feedback via platforms like the Google Play Store. This research utilizes the K-Nearest Neighbor (KNN) algorithm to conduct sentiment analysis on 5,000 user reviews, with the goal of assessing its classification effectiveness. Following preprocessing (case folding, Text Cleaning, tokenization, stopword Removal, stemming), the data was converted using TF-IDF and balanced through SMOTE. Experimental findings indicate that KNN attained a peak accuracy of merely 36.53% (at k = 14), reflecting weak performance with high-dimensional textual data. In contrast, Logistic Regression attained a notably higher accuracy of 88%, showcasing its dominance for this task. The results offer perspectives for game developers to assess user feelings and emphasize the significance of selecting suitable machine learning models. Future research should investigate advanced classifiers like SVM, Random Forest, or deep learning methods to enhance accuracy.

1. Introduction

The rapid advancement of information technology has significantly influenced the growth of the digital gaming industry, particularly in the mobile platform sector. One of the most popular games in this domain is Free Fire, which has received millions of user reviews on the Google Play Store. These reviews are rich in user sentiments and often contain critical insights regarding gameplay, user interface, performance, and overall satisfaction. Despite the abundance of such data, there is a noticeable lack of research that focuses on evaluating the performance of traditional classification algorithms, such as K-Nearest Neighbor (KNN), in the context of sentiment analysis for mobile game reviews. This presents a significant research gap that deserves attention. Sentiment analysis, a technique within the field of Natural Language Processing (NLP), aims to classify textual opinions into categories such as positive, negative, or neutral [1]. While a variety of machine learning algorithms have been explored in prior sentiment analysis studies—such as Naïve Bayes, Support Vector Machine (SVM), and Logistic Regression—traditional models like KNN are often overlooked due to their limitations in handling high-dimensional text data, even though they offer simplicity and interpretability [2], [3]. To address this gap, this study seeks to answer the following research question: Can the K-Nearest Neighbor algorithm effectively classify sentiments in Free Fire user reviews from the Google Play Store? The objective of this research is to evaluate the performance of KNN in sentiment classification and compare it with Logistic

Regression, which has been proven to outperform KNN in managing sparse and high-dimensional feature spaces in previous studies [3], [4]. This research applies a structured methodology, including data collection via web scraping, preprocessing (case folding, text cleaning, tokenization, stopword removal, stemming), feature extraction using Term Frequency-Inverse Document Frequency (TF-IDF), class balancing using Synthetic Minority Over-sampling Technique (SMOTE), and performance evaluation using standard classification metrics. By conducting this comparative analysis, the study not only contributes empirical findings but also reinforces the conclusions of earlier works that suggest Logistic Regression is more suitable for sentiment analysis in the context of mobile app reviews [2], [3], [4], [5].

2. Research Method

The method used for this research goes through the following stages. This research employs a web scraping method to collect 5,000 user reviews of the Free Fire game from the Google Play Store. Each review is then assigned a sentiment label—positive, negative, or neutral—using a rating-based approach. Specifically, reviews with 4 or 5 stars are classified as positive, those with 1 or 2 stars as negative, and 3-star reviews are considered neutral. This automated labeling technique provides a scalable and efficient way to categorize large volumes of data without requiring manual annotation.

The initial stage in this research involves preprocessing the user review dataset to ensure the data is clean and suitable for further analysis. Several preprocessing techniques were applied. The first step is case folding, which converts all characters in the text to lowercase to eliminate inconsistencies caused by differences between uppercase and lowercase letters. Next, text cleaning is performed to remove irrelevant characters such as numbers, symbols, punctuation marks, and other special characters, thereby producing cleaner and more uniform text. This is followed by tokenization, a process of splitting the text into smaller units called tokens, which may consist of words, sentences, or punctuation. The fourth step is stopword removal, which eliminates frequently occurring words (e.g., "and", "or", "which") that do not carry significant meaning and may interfere with the analysis. Finally, stemming is applied to reduce words to their root or base forms, thereby unifying different variations of the same word and simplifying the text structure for more effective analysis.

In this study, the preprocessed user reviews were converted into numerical vectors using the Term Frequency-Inverse Document Frequency (TF-IDF) method. To address class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was applied, ensuring balanced sentiment classes during the model training phase. The K-Nearest Neighbor (K-NN) algorithm was then implemented by determining the optimal value of *k*, representing the number of nearest neighbors. The model was trained using the training data and evaluated on the testing data. To assess the model's performance, evaluation metrics such as accuracy, precision, recall, and F1-score were used, measuring how well the model could classify the reviews into the three sentiment classes: positive, negative, and neutral.

Figure 1 illustrates the complete research process for analyzing user review sentiments regarding the Free Fire mobile game utilizing the K-Nearest Neighbor (KNN) algorithm. The research starts with the data gathering phase, during which 5,000 user reviews were obtained from the Google Play Store through web scraping. These evaluations act as the main dataset for sentiment analysis.

After data collection, a sentiment tagging procedure was carried out. In this research, sentiment tags were automatically given according to the star rating of the review. Reviews with ratings of 4 or 5 stars were classified as positive, those with 1 or 2 stars as negative, while 3-star reviews were deemed neutral. This labeling method based on ratings was selected for its effectiveness and ability to scale when handling extensive amounts of user-generated data without requiring manual tagging or outside lexicons.

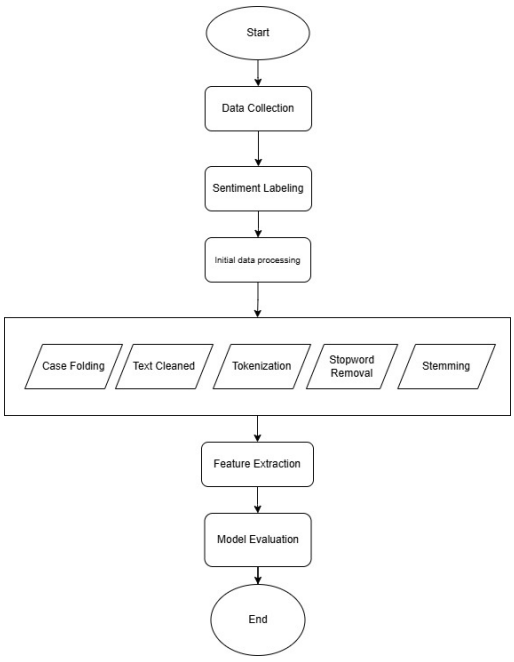

Figure 1. Flowchart of KNN implementation

The labeled dataset underwent initial data preprocessing, which included five successive text normalization procedures: case folding, text cleaning, tokenization, removal of stopwords, and stemming. Case folding meant changing all text to lowercase to remove inconsistencies created by capitalization. Text cleaning eliminated unnecessary characters like numbers, punctuation marks, and special symbols. Tokenization divided each review into separate tokens or words. Stopword removal Removal frequent words with minimal semantic importance, while stemming simplified words to their base forms, allowing for the categorization of word variants and decreasing feature dimensionality.

Subsequently, the feature extraction phase was carried out by transforming the preprocessed text into numerical vectors using the Term Frequency–Inverse Document Frequency (TF-IDF) method. This technique assigns weights to terms based on their relevance and frequency, enhancing the discriminative power of the resulting features. To address class imbalance in the dataset, the Synthetic Minority Over-sampling Technique (SMOTE) was applied. SMOTE generates synthetic instances of minority classes, ensuring that the training data for each sentiment category is balanced and representative.

In the model evaluation phase, the KNN algorithm was implemented to perform sentiment classification. Several values of the hyperparameter k were tested to determine the optimal neighborhood size. However, given the high dimensionality of TF-IDF vectors and the sparsity of the data, KNN exhibited suboptimal performance. As a result, a Logistic Regression model was introduced as a comparative benchmark. Logistic Regression was selected due to its robustness in handling sparse, high-dimensional text data and its ability to produce more stable performance metrics.

The research process concludes at the final stage, where model performance is assessed based on standard classification metrics, including accuracy, precision, recall, and F1-score. Each step in the proposed framework is systematically aligned with the flowchart, providing a coherent and reproducible methodology for sentiment analysis in mobile gaming applications.

3. Results and Discussions

This research uses the web scraping method to collect 5,000 Free Fire user reviews from the Google Play Store. The data is then processed through preprocessing stages, including case folding, tokenization, stopword

removal, and stemming. This research utilizes user review data of the Free Fire game application retrieved from its Google Play Store page. The data was collected using the web scraping method, resulting in a total of 5,000 reviews. These reviews contain various sentiments, primarily categorized as positive and negative.

Below are examples of positive reviews and negative reviews of the Free Fire application. Example of positif review is like this. *The game is good cool, I used to play ram 2 hp and it really really ngeleg really often exit itself when playing, anyway ngelag ngelag and I delete the ff . . . But it's still good and since I changed my phone I downloaded ff again, and really it's really good ff from the graphics already hd in my opinion, lots of new weapons, so I want to top up continuously.* Example of negative review is like this. *2 stars first. So here's the story. there was a bug that the bug made my character suddenly unable to move, and my friend's character was in a running position, but strangely he was suddenly stiff like a running statue. Now strangely enough, buttons such as med kits, maps, etc. disappear when pressed. strangely enough when the game is over / game time is up, I am stuck in the match and cannot get out. but my friend can get out, even though the ping is 50. [please fix garena] :(.*

The initial stage in the classification process is case folding. This step involves converting all the text data into a uniform format, typically by changing all characters to lowercase. By standardizing the text, case folding helps reduce redundancy caused by different letter cases (e.g., "Free" and "free" are treated the same), thereby simplifying further text processing steps such as tokenization, filtering, and feature extraction.

Table 1. Case Folding

Text Before Case Folding	Text after Case Folding
untuk pihak Garena , game nya udah bagus dari segi gameplay , tetapi jaringan nya kok bisa ngelag , pdhl wifi sy bagus jaringannya , dan juga ketika maen game online lain bagus , tapi ketika main ff jaringan nya 999+ terus , gw ganti pake data pun sama tetap 999+ , tapi kalo game online lain lancar bgt jaringannya , MOHON UNTUK PIHAK GARENA DI FIX YA !! 🙏🙏 TERIMA KASIH !!	untuk pihak garena , game nya udah bagus dari segi gameplay , tetapi jaringan nya kok bisa ngelag , pdhl wifi sy bagus jaringannya , dan juga ketika maen game online lain bagus , tapi ketika main ff jaringan nya 999+ terus , gw ganti pake data pun sama tetap 999+ , tapi kalo game online lain lancar bgt jaringannya , mohon untuk pihak garena di fix ya !! 🙏🙏 terima kasih !!

This table demonstrates the case folding process applied to the user review text. Case folding is the first step in text preprocessing, where all characters are converted to lowercase. This process is essential to eliminate inconsistencies due to capitalization, ensuring that words such as "Game" and "game" are treated identically during analysis. As shown in the table, the text content before and after case folding remains semantically the same, but all uppercase letters have been standardized to lowercase.

Table 2. Text Cleaning

Text before Cleaning	Text after Cleaning
untuk pihak garena , game nya udah bagus dari segi gameplay , tetapi jaringan nya kok bisa ngelag , pdhl wifi sy bagus jaringannya , dan juga ketika maen game online lain bagus , tapi ketika main ff jaringan nya 999+ terus , gw ganti pake data pun sama tetap 999+ , tapi kalo game online lain lancar bgt jaringannya , mohon untuk pihak garena di fix ya !! 🙏🙏 terima kasih !!	untuk pihak garena game nya udah bagus dari segi gameplay tetapi jaringan nya kok bisa ngelag pdhl wifi sy bagus jaringannya dan juga ketika maen game online lain bagus tapi ketika main ff jaringan nya terus gw ganti pake data pun sama tetap tapi kalo game online lain lancar bgt jaringannya mohon untuk pihak garena di fix ya terima kasih

Table 2 illustrates the text cleaning stage, which involves removing irrelevant characters such as punctuation marks, emojis, special symbols, and numeric values. This step is critical to produce a clean and uniform dataset

that is suitable for further processing. By eliminating these extraneous elements, the model can focus on the meaningful content of each review. The comparison in the table shows how the noisy elements are successfully removed in the cleaned version.

Table 3. Tokenization

Text before Tokenization	Text after Tokenization
untuk pihak garena game nya udah bagus dari segi gameplay tetapi jaringan nya kok bisa ngelag pdhl wifi sy bagus jaringannya dan juga ketika maen game online lain bagus tapi ketika main ff jaringan nya terus gw ganti pake data pun sama tetap tapi kalo game online lain lancar bgt jaringannya mohon untuk pihak garena di fix ya terima kasih	[untuk, pihak, garena, game, nya, udah, bagus, dari, segi, gameplay, tetapi, jaringan, nya, kok, bisa, ngelag, pdhl, wifi, sy, bagus, jaringannya, dan, juga, ketika, maen, game, online, lain, bagus, tapi, ketika, main, ff, jaringan, nya, terus, gw, ganti, pake, data, pun, sama, tetap, tapi, kalo, game, online, lain, lancar, bgt, jaringannya, mohon, untuk, pihak, garena, di, fix, ya, terima, kasih]

Table 3 presents the results of the tokenization process. Tokenization involves splitting the cleaned text into smaller units called tokens, which typically represent individual words. This process facilitates word-level analysis, which is foundational for subsequent techniques such as stopword removal and stemming. The table clearly shows how a continuous review sentence is segmented into a structured list of tokens.

Table 4. Stopword Removal

Text before Stopword Removal	Text after Stopword Removal
[untuk, pihak, garena, game, nya, udah, bagus, dari, segi, gameplay, tetapi, jaringan, nya, kok, bisa, ngelag, pdhl, wifi, sy, bagus, jaringannya, dan, juga, ketika, maen, game, online, lain, bagus, tapi, ketika, main, ff, jaringan, nya, terus, gw, ganti, pake, data, pun, sama, tetap, tapi, kalo, game, online, lain, lancar, bgt, jaringannya, mohon, untuk, pihak, garena, di, fix, ya, terima, kasih]	pihak garena game nya udah bagus segi gameplay jaringan nya kok ngelag pdhl wifi sy bagus jaringannya juga maen game online bagus ketika main ff jaringan nya terus gw ganti pake data sama tetap kalo game online lancar bgt jaringannya mohon pihak garena fix terima kasih

Table 4 describes the impact of removing stopwords from the tokenized text. Stopwords are frequently occurring words (e.g., "dan", "yang", "untuk") that do not carry significant meaning and can dilute the effectiveness of sentiment analysis. The table compares the token list before and after stopword removal, highlighting how non-essential words have been excluded to improve the quality and relevance of the remaining data.

Table 5. Stemming

Text before Stemming	Text after Stemming
pihak garena game nya udah bagus segi gameplay jaringan nya kok ngelag pdhl wifi sy bagus jaringannya juga maen game online bagus ketika main ff jaringan nya terus gw ganti pake data sama tetap kalo game online lancar bgt jaringannya mohon pihak garena fix terima kasih	pihak garena game nya udah bagus segi gameplay jaring nya kok ngelag pdhl wifi sy bagus jaring juga maen game online bagus ketika main ff jaring nya terus gw ganti pake data sama tetap kalo game online lancar bgt jaring mohon pihak garena fix terima kasih

Table 5 shows the stemming process, where each word is reduced to its base or root form. Stemming helps consolidate word variants that have similar meanings (e.g., "jaringan", "jaringannya" → "jaring") into a single

representative form. This process reduces dimensionality and enhances the consistency of features used in model training. The table illustrates how words are normalized to their root forms while retaining the original context.

After completing the preprocessing stage, the next step in the sentiment analysis pipeline is feature extraction. In this study, the Term Frequency–Inverse Document Frequency (TF-IDF) method is employed to convert the textual data into numerical vectors that can be processed by machine learning algorithms. TF-IDF is a widely used technique in natural language processing, where each term's importance is quantified based on its frequency in a specific document relative to its frequency across the entire corpus. Words that appear frequently in a document but less frequently across other documents receive higher weights, making TF-IDF highly effective in distinguishing relevant terms for classification tasks [1]. This transformation is particularly crucial when working with high-dimensional, sparse text data such as user reviews. By applying TF-IDF, the dataset becomes more structured and informative, allowing the classification model to identify patterns associated with different sentiment classes [6]. Previous studies have demonstrated the effectiveness of TF-IDF in enhancing classification accuracy when combined with algorithms such as KNN and Logistic Regression [2], [3].

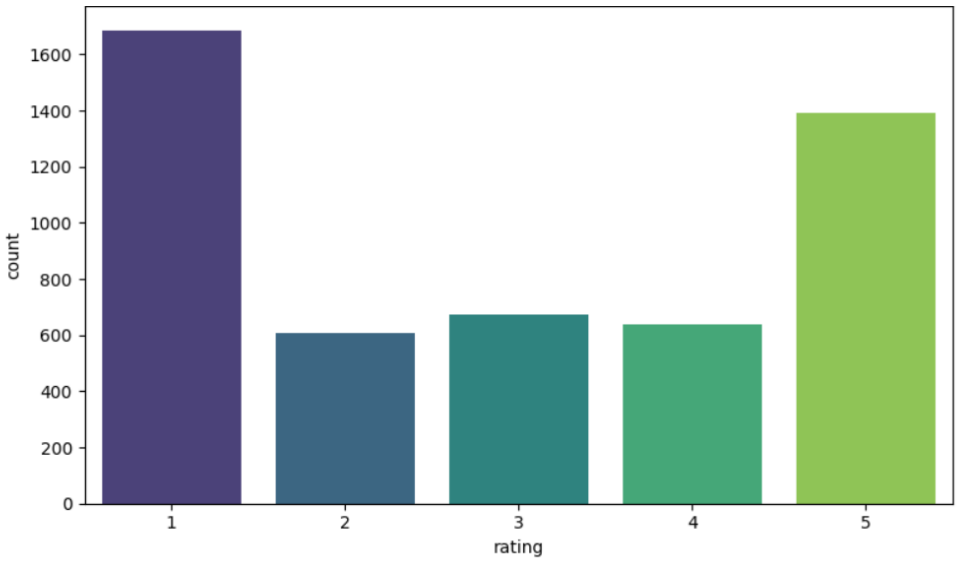

Figure 2. Free Fire Reviews Rating sample chart

Based on Figure 2, it can be observed that the distribution of user ratings is highly varied. Ratings with a value of 1 appear most frequently, totaling 1,687 reviews, while ratings with a value of 5 also appear significantly, with 1,393 occurrences. This distribution illustrates the diversity of user sentiment toward the *Free Fire* game application and highlights the need for automated sentiment analysis. Such analysis can provide meaningful insights into user satisfaction and dissatisfaction, which are crucial for application developers and companies in making strategic decisions to maintain and improve service quality [7], [2]. The application of the K-Nearest Neighbor (KNN) method in classifying *Free Fire* user reviews is feasible but presents certain limitations. If the model accuracy falls below 50%, several contributing factors may be involved, such as an imbalanced dataset, suboptimal selection of the *k* parameter, or poor feature normalization [3], [4].

To address these challenges, this study enhances classification performance through the integration of the Term Frequency-Inverse Document Frequency (TF-IDF) technique and the Synthetic Minority Over-sampling Technique (SMOTE). TF-IDF is used in the feature extraction stage to convert text data into numerical vectors that reflect the relative importance of each term across the corpus [1]. SMOTE, on the other hand, addresses the common issue of class imbalance in sentiment datasets by generating synthetic samples for minority classes, ensuring a more balanced training set [3]. The combination of TF-IDF and SMOTE has proven effective

in improving the performance of KNN, as evidenced by the more consistent and balanced evaluation metrics, including accuracy, precision, and recall, across all sentiment categories [8], [3].

Table 6. Evaluation of KNN Results with various values of K

K	Accuracy	Precision	Recall
14	36.53%	42.98%	36.53%
16	35.84%	43.93%	35.83%
22	34.47%	41.92%	34.47%
26	33.95%	40.59%	33.95%

Based on the test results of several variations of the k value in the K-Nearest Neighbor (KNN) algorithm, the model performance tends to be low and inconsistent. At k = 14, the maximum accuracy recorded was 36.53%, with 42.98% precision and 36.53% recall. However, as the value of k increases, the performance decreases. A value of k = 16 gave an accuracy of 35.84%, while at k = 22 and k = 26, the accuracy decreased to 34.47% and 33.95% respectively. This decrease is also accompanied by a decrease in precision and recall. The low performance of KNN at all variations of the k value indicates that the algorithm is unable to classify sentiment well in the context of user review text data. This is most likely due to the KNN's inability to handle the high-dimensional data generated from the TF-IDF representation, as well as the algorithm's sensitivity to the distance between data in a complex feature space [8], [3], [9], [10], [11]. Therefore, this finding serves as a reference to compare and direct the application of more appropriate alternative classification models, such as Logistic Regression, in this study. This limitation is also highlighted in other studies comparing KNN with probabilistic classifiers like Naïve Bayes, where KNN often lags behind due to its sensitivity to feature space dimensionality [12].

The results of the assessment of the K-Nearest Neighbor (KNN) algorithm show that the model accuracy rate is below 50%. This indicates that KNN is not effective enough in classifying sentiment in Free Fire game user reviews on the Google Play Store. This poor performance may be due to several reasons, including KNN's sensitivity to data distribution, non-linear data distance, and the high complexity of unstructured text data [3], [13]. Although the TF-IDF technique has been applied as a means of feature extraction and SMOTE is used to balance the number of classes, KNN still faces difficulties in accurately determining the nearest neighbor in high-dimensional spaces [4]. Similar challenges were identified in other comparative studies, where Naïve Bayes outperformed KNN in sentiment classification tasks involving short text on platforms like Twitter [12].

To further demonstrate model effectiveness, Figure 3 showcases the confusion matrices for the K-Nearest Neighbor and Logistic Regression models, respectively. The KNN matrix exhibits considerable misclassification in all sentiment categories, often misidentifying positive and neutral sentiments as negative. This reinforces the idea that KNN has difficulty with sparse, high-dimensional TF-IDF vectors because of its dependence on distance measures [3], [14], [15], [16], [17].

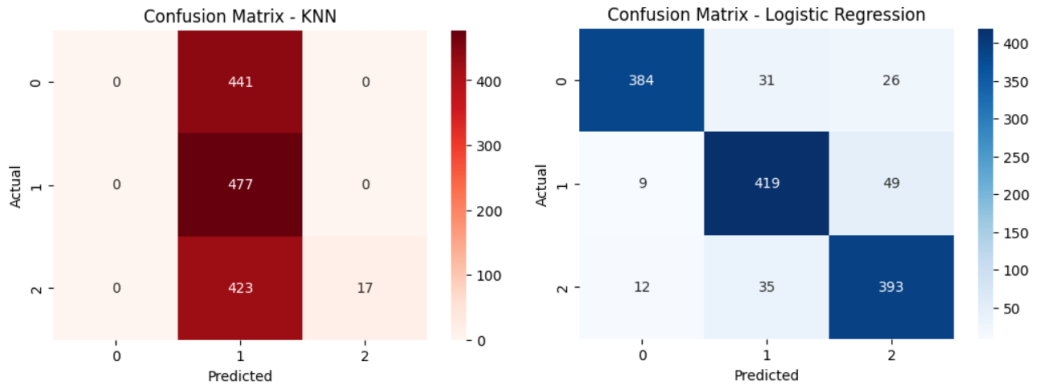

Figure 3. results of confusion matrix from KNN and Logistic regression

Table 7. Comparison between KNN and Logistic Regression Accuracy

Matrix	*KNN*	*Logistic Regression*
Accuracy	*34.14%*	*92.49%*
Precision	*41.92%*	*92.68%*
Recall	*35.07%*	92.52%
F1-Score	*38.22%*	*92.60%*

In comparison and as an alternative, the Logistic Regression algorithm is utilized to address the limitations of KNN. Logistic Regression was selected as it is a linear classification model that proves more effective in handling text data represented as vectors through TF-IDF. This model operates by estimating the probability that a data point belongs to a specific class, enhancing its efficiency in handling linear relationships between features. The evaluation outcomes indicate that Logistic Regression outperforms KNN regarding accuracy, precision, and recall, leading to the conclusion that Logistic Regression is more suitable for sentiment analysis in this study's context.

A more in-depth technical rationale for KNN's inadequate performance can be linked to its sensitivity to the arrangement of the feature space, especially when employing TF-IDF vectorization. TF-IDF converts text data into high-dimensional and sparse vectors, with the majority of features having zero values. In these environments, the concept of "proximity" employed by KNN loses significance, as the distances among data points often equalize, complicating the identification of genuine nearest neighbors. Moreover, KNN does not create a model from the training data; it merely stores it, leading to inadequate generalization for noisy, high-dimensional inputs. These factors together impede KNN's capacity to accurately classify sentiment in user reviews, particularly in contrast to stronger algorithms such as Logistic Regression.

These findings are in line with the results of this study, supporting the conclusion that Logistic Regression is better suited for sentiment classification tasks involving user-generated text in mobile applications. Furthermore, Muktafin et al. also showed that Logistic Regression handled sparse and high-dimensional datasets more effectively than distance-based models like KNN [3]. In another study, Wahyudi and Kusumawardana compared KNN, Naïve Bayes, SVM, and Logistic Regression for Gojek application reviews and concluded that Logistic Regression outperformed other models [2]. Several recent studies have evaluated the performance of KNN and other algorithms in analyzing sentiment from mobile application reviews. For instance, a study by Sari et al. applied KNN to analyze reviews of the game Clash of Clans and achieved high accuracy using TF-IDF weighting.

These findings demonstrate that the Logistic Regression algorithm is significantly more suitable than KNN for sentiment classification on text data, especially when using TF-IDF features. The strong performance of Logistic

Regression in this study aligns with results from prior research. For instance, Wahyudi and Kusumawardana reported that Logistic Regression outperformed KNN in analyzing app reviews using Support Vector Machine and Logistic Regression as baselines [2]. Similarly, Muktafin et al. highlighted the effectiveness of Logistic Regression in handling sparse and high-dimensional datasets compared to distance-based classifiers like KNN [3]. These supporting studies reinforce the conclusion that Logistic Regression provides a more robust and scalable approach for sentiment analysis in mobile app reviews

4. Conclusions and Future Work

This research sought to assess how well the K-Nearest Neighbor (KNN) algorithm performs in analyzing sentiment in user reviews of the Free Fire mobile game obtained from the Google Play Store. The study employed a systematic approach, beginning with extensive preprocessing that involved case folding, text cleaning, tokenization, removal of stopwords, and stemming, followed by feature extraction with the TF-IDF technique and class balancing via SMOTE.

The results show that although KNN is straightforward and easy to understand, it struggled with high-dimensional, sparse text data, reaching a maximum accuracy of merely 36.53%. The model also had difficulty providing acceptable outcomes regarding precision, recall, and F1-score. These constraints are especially evident when contrasted with the Logistic Regression model, which greatly exceeded KNN, achieving over 92% across all assessment metrics. This result is in line with previous research conducted by Wahyudi and Kusumawardana [2], as well as Muktafin et al. [3], who also demonstrated the superior performance of Logistic Regression over KNN in similar contexts. This outcome highlights that KNN is suboptimal for tasks dealing with complex, unstructured text data, especially when utilizing TF-IDF representation.

To improve upcoming studies in sentiment analysis, especially those that examine user-generated content with comparable traits, various approaches can be contemplated for additional investigation. Future research should investigate Transformer-based models like BERT (Bidirectional Encoder Representations from Transformers), which excel at grasping intricate contextual relationships in text and have consistently shown better performance in multiple NLP tasks. Researchers might utilize hybrid models that merge conventional feature extraction methods like TF-IDF with deep learning structures such as Long Short-Term Memory (LSTM) or Convolutional Neural Networks (CNN), or they could integrate various classifiers to achieve a balance between interpretability and predictive effectiveness.

Third, assessing different algorithms apart from KNN and Logistic Regression, like Support Vector Machine (SVM), Random Forest, or Naïve Bayes, is beneficial as they might provide improved accuracy in handling high-dimensional feature spaces. Moreover, enhancing preprocessing methods—like adding slang normalization, managing negations, and utilizing domain-specific sentiment lexicons—can greatly enhance the quality and applicability of textual attributes. To more effectively tackle class imbalance, future studies could investigate alternative oversampling or ensemble methods like ADASYN or hybrid resampling techniques, which may provide stronger model performance than traditional methods such as SMOTE. Together, these suggestions seek to promote the creation of more precise, scalable, and flexible sentiment classification systems in upcoming research.

5. References

[1] S. Rahayu, Y. MZ, J. E. Bororing, and R. Hadiyat, "Implementasi Metode K-Nearest Neighbor (K-NN) untuk Analisis Sentimen Kepuasan Pengguna Aplikasi Teknologi Finansial FLIP," *Edumatic: Jurnal Pendidikan Informatika*, vol. 6, no. 1, pp. 98–106, Jun. 2022, doi: 10.29408/edumatic.v6i1.5433.

[2] R. Wahyudi and G. Kusumawardana, "Analisis Sentimen pada Aplikasi Grab di Google Play Store Menggunakan Support Vector Machine," *Jurnal Informatika*, vol. 8, no. 2, pp. 200–207, Sep. 2021, doi: 10.31294/ji.v8i2.9681.

[3] E. H. Muktafin, K. Kusrini, and E. T. Luthfi, "Analisis Sentimen pada Ulasan Pembelian Produk di Marketplace Shopee Menggunakan Pendekatan Natural Language Processing," *Jurnal Eksplora Informatika*, vol. 10, no. 1, pp. 32–42, Sep. 2020, doi: 10.30864/eksplora.v10i1.390.

[4] R. Wahyudi and G. Kusumawardana, "Analisis Sentimen pada Aplikasi Grab di Google Play Store Menggunakan Support Vector Machine," *Jurnal Informatika*, vol. 8, no. 2, pp. 200–207, Sep. 2021, doi: 10.31294/ji.v8i2.9681.

[5] S. G. Setyorini and Mustakim, "Application of the Nearest Neighbor Algorithm for Classification of Online Taxibike Sentiments in Indonesia in the Google Playstore Application," *J Phys Conf Ser*, vol. 2049, no. 1, p. 12026, 2021.

[6] A. Nurian and B. N. Sari, "Analisis Sentimen Ulasan Pengguna Aplikasi Google Play Menggunakan Naïve Bayes," *JITET (Jurnal Informatika dan Teknik Elektro Terapan)*, pp. 829–835, 2023.

[7] M. Raffi, A. Suharso, and I. Maulana, "Analisis Sentimen Ulasan Aplikasi Binar Pada Google Play Store Menggunakan Algoritma Naïve Bayes," *INTECOMS: Journal of Information Technology and Computer Science*, vol. 6, no. 1, pp. 450–462, Jun. 2023, doi: 10.31539/intecoms.v6i1.6117.

[8] R. Ramadhan, M. Afdal, I. Permana, and M. Jazman, "Analisis Sentimen pada Ulasan Aplikasi Maxim di Google Play Store dengan K-Nearest Neighbor," *JURIKOM (Jurnal Riset Komputer)*, vol. 10, no. 3, 2023.

[9] D. Y. Lakoro, E. Utami, and D. Ariatmanto, "Sentiment Analysis of BNI Mobile Application Using The K-Nearest Neighbor Algorithm (KNN) With Particle Swarm Optimization (PSO) Feature Selection," *INTECOMS: Journal of Information Technology and Computer Science*, vol. 6, no. 2, pp. 948–953, Nov. 2023, doi: 10.31539/intecoms.v6i2.7912.

[10] A. Setiawan, "Perbandingan Penggunaan Jarak Manhattan, Jarak Euclid, dan Jarak Minkowski dalam Klasifikasi Menggunakan Metode KNN pada Data Iris," *Jurnal Sains dan Edukasi Sains*, vol. 5, no. 1, pp. 28–37, May 2022, doi: 10.24246/juses.v5i1p28-37.

[11] M. K. Anam, B. N. Pikir, and M. B. Firdaus, "Penerapan Na¨ıve Bayes Classifier, K-Nearest Neighbor (KNN) dan Decision Tree untuk Menganalisis Sentimen pada Interaksi Netizen danPemeritah," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 21, no. 1, pp. 139–150, Nov. 2021, doi: 10.30812/matrik.v21i1.1092.

[12] A. F. N. Azizah and V. P. Ramadhan, "Comparison of Naïve Bayes and K-N-N in Sentiment Analysis on Twitter Regarding the Victory of Candidate Pair 02," *J-Intech: Journal of Information and Technology*, vol. 12, no. 2, pp. 228–237, Dec. 2024.

[13] M. Furqan, Sriani, and S. M. Sari, "Analisis Sentimen Menggunakan K-Nearest Neighbor Terhadap New Normal Masa Covid-19 di Indonesia," *Techno.COM*, vol. 21, pp. 52–61, Feb. 2022.

[14] S. Alfaris and Kusnawi, "Komparasi Metode KNN dan Naive Bayes Terhadap Analisis Sentimen Pengguna Aplikasi Shopee," *Indonesian Journal of Computer Science*, vol. 12, no. 5, Oct. 2023, doi: 10.33022/ijcs.v12i5.3304.

[15] G. Darmawan, S. Alam, and M. I. Sulistyo, "Analisis Sentimen Berdasarkan Ulasan Pengguna Aplikasi Mypertamina Pada Google Playstore Menggunakan Metode Naïve Bayes," *STORAGE – Jurnal Ilmiah Teknik dan Ilmu Komputer*, vol. 2, no. 3, pp. 100–108, 2023.

[16] M. Riski, M. Fikry, and Yusra, "Klasifikasi Sentimen Ulasan Aplikasi WhatsApp di Play Store Menggunakan Metode K-Nearest Neighbor," *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 4, no. 1, pp. 438–444, 2023, doi: 10.30865/klik.v4i1.1050.

[17] P. Astuti and N. Nuris, "Penerapan Algoritma KNN Pada Analisis Sentimen Review Aplikasi Peduli Lindungi," *Computer Science (CO-SCIENCE)*, vol. 2, no. 2, pp. 137–142, Jul. 2022, doi: 10.31294/coscience.v2i2.1258.