

Enhancing K-NN Performance with SMOTE for Sentiment Analysis of Streaming App Reviews

Ni Putu Eka Patrycia Dewi^{1*}, Christina Purnama Yanti², I Made Marthana Yusa³

¹ Technology and Informatics, Informatics Engineering, Institut Bisnis dan Teknologi Indonesia, Bali, Indonesia

² Business and Creative Design, Visual Communication Design, Institut Bisnis dan Teknologi Indonesia, Bali, Indonesia

Email: ^{1*}niputu_eka_patrycia@instiki.ac.id, ²christinapy@instiki.ac.id, ³made.marthana@instiki.ac.id

Correspondence Author Email: niputu_eka_patrycia@instiki.ac.id

Submitted: 03/11/2024; Accepted: 29/12/2024; Published: 30/12/2024

Abstract—This research aims to analyze the sentiment of user reviews for a popular streaming app on both the Play Store and App Store using the K-Nearest Neighbor (K-NN) method. As the user base expands, reviews increasingly influence app development, guiding improvements and optimizing user experience. However, the large volume of reviews renders manual analysis inefficient and prone to inconsistencies, underscoring the necessity of sentiment analysis to quickly and accurately capture user perceptions. Review data were collected from both platforms, with preprocessing steps such as data cleaning, tokenization, and normalization applied to ensure data consistency. The Synthetic Minority Over-sampling Technique (SMOTE) was used to address class imbalance, enhancing the reliability of classification results. Findings indicate that SMOTE improved model accuracy, raising it from 74% to 82.9% for Play Store data and from 79% to 84.1% for App Store data. Furthermore, a notable difference in sentiment dominance was observed, with positive sentiment prevailing on the Play Store, while negative sentiment was more prevalent on the App Store. These insights reveal that, overall, the app is well received, although certain areas highlighted in negative reviews require further attention to improve user satisfaction.

Keywords: Sentiment Analysis, K-Nearest Neighbor (KNN), SMOTE, Streaming App, Pre-processing

1. INTRODUCTION

The percentage of Indonesians who use the internet has increased to 73.7% in 2019, and 58.8% of internet users watch movies, TV series, and videos on the internet. This activity ranks in the top ten among the most frequent activities performed by Indonesians when using the Internet [1]. With the increasing use of the internet, the use of streaming applications has also increased, streaming applications make it easy for users to download or re-watch the TV series and movies they want, and provide free access to choose both the movies and TV series they want [2]. Streaming media has become popular as a technology that allows real-time distribution of audio, video, and multimedia data over the internet [3]. One such streaming application is RCTI+, which was launched on August 24, 2019, becoming one of the popular platforms as it offers a wide array of television content, movies, and sporting events. RCTI+ is named after one of MNC Group's television stations, RCTI. The first private television station in Indonesia was RCTI, also known as Rajawali Citra Televisi Indonesia. However, the “+” sign in the name RCTI+ indicates the shift from free-to-air TV media to digital [4].

As users of the RCTI+ application grow, users often provide reviews about the services provided by the application. RCTI+ is one of Indonesia's local streaming applications that not only provides live broadcast services of television programs but also has exclusive content that is only available on the RCTI+ platform. Application users provide a variety of reviews, both positive and negative. User reviews in the Play Store and App Store are an important source for app improvement [5]. However, a large volume of reviews can consume time and effort to read and analyze them if done manually and cumbersome. Therefore, sentiment analysis on RCTI+ user reviews is crucial to identify user perceptions quickly and accurately. Sentiment analysis is the computational method used to identify and categorize opinions in text, helping to determine if the author's view on a specific topic, product, or service is positive, negative, or neutral [6].

There are previous studies that also discuss sentiment analysis. The first study analyzed sentiment towards digital payment services in Indonesia using Twitter data and found that KNN outperformed Naïve Bayes, with OVO experiencing more negative sentiment [7]. The second study evaluated sentiment toward Indonesian television episodes on Twitter combining Lexicon-based methods with K-NN algorithms, the best results were obtained at k=3 with 74% accuracy [8]. The third study focused on COVID-19-related news in Arabic on Twitter and found mostly negative sentiments in the Gulf countries [9]. The fourth study analyzed public sentiment towards global recession news in Indonesia, balancing nearly 300,000 tweets using SMOTE, achieving 84% accuracy with the Bernoulli Naive Bayes algorithm [10]. The fifth study analyzed the sentiment of Vidio application user reviews with the K-Nearest Neighbor algorithm with 70% accuracy using Microsoft Excel and 50% accuracy using Rapidminer [11]. Previous research has similarities in general objectives with the current research, namely to conduct sentiment analysis of an application or service and determine the accuracy of the methods used. The difference in the above research lies in the object of research where this research is RCTI +, which uses data sources from the App Store and Play Store, the tools for data processing and the study results.

The classifier used in this study is called K-Nearest Neighbors (K-NN), which is a classification technique that uses the distance between the object to be evaluated and its K closest neighbors in the training data to determine the object's classification [12]. When compared to other algorithms like Naïve Bayes, this approach has demonstrated a

high degree of accuracy [13]. Furthermore, a prevalent issue in machine learning is data imbalance. The classification process may be biased in favor of the majority class since the minority class may be more challenging to categorize than the majority class due to the uneven amount of data. The research will use the Synthetic Minority Oversampling Technique (SMOTE), which takes samples from the minority class and adds synthetic samples to increase the amount of data about the minority class [14]. This research is expected to produce a series of information based on RCTI+ application user reviews, which can be used as one of the guidelines or consideration tools for companies in making decisions to improve the application. Thus, it is expected to reduce the level of low ratings and retain users to keep using RCTI+ without switching to other streaming platforms.

2. RESEARCH METHODOLOGY

2.1 Research Stages

To achieve the expected results, this research undertakes a number of steps or processes. The process begins with data collection, preprocessing to clean the data, application of the algorithm, and evaluation of the results. Each of these steps was designed in a way that allowed the analysis to yield a deep understanding of the research objectives. The several stages of this research are illustrated in Figure 1.

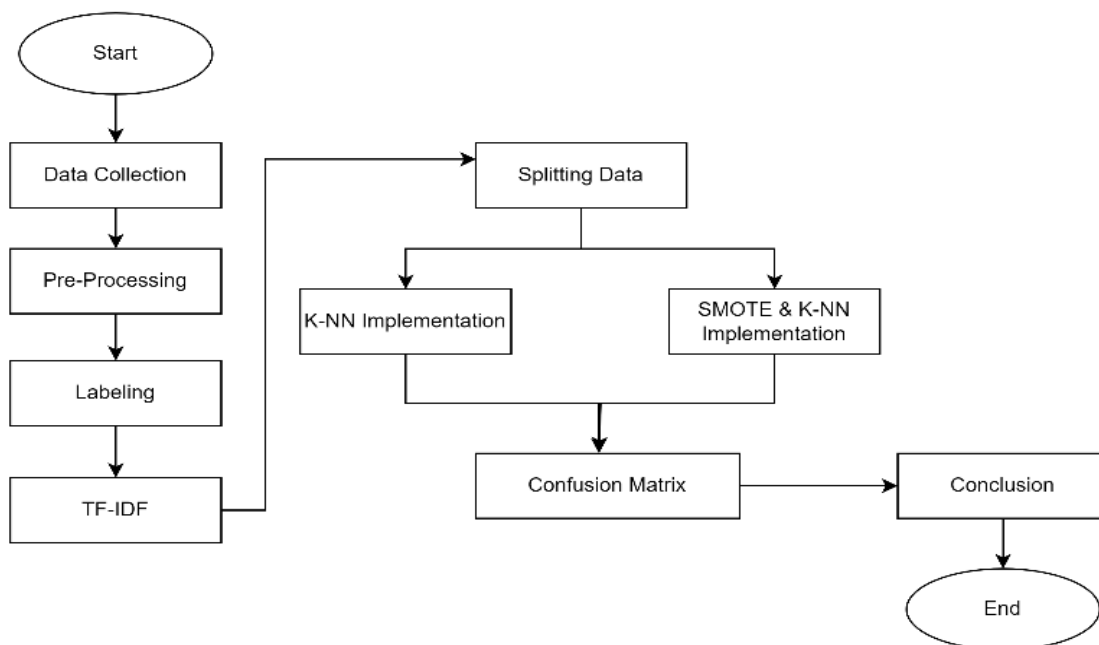


Figure 1. Research Stages

2.2 Data Collection

Data collection is conducted to gather information relevant to the research problems. In this phase, data is sourced from RCTI+ reviews on both the Play Store and App Store through a web scraping process implemented in Python, utilizing Google Colab as the platform [15]. The web scraping process yields a total of 2000 data points, with 1000 reviews collected from the Play Store and 1000 reviews from the App Store. By leveraging automated data collection techniques, the research efficiently compiles a substantial volume of data for subsequent analysis.

2.2 Pre-processing

After obtaining data from the Play Store and App Store, the preprocessing stage was continued. At this stage, data cleaning and attribute reduction are performed. The scraped data contains a lot of noise, so this cleaning and reduction is necessary to avoid errors in the classification process [16]. Pre-processing involves several key steps to prepare data for analysis. First, cleaning removes irrelevant features like punctuation and emojis, case folding converts all text to lowercase, normalization standardizes slang and non-standard words using the dictionary, tokenizing segments text into smaller units, stopword removal eliminates insignificant words, and stemming reduces words to their root forms by removing prefixes and suffixes [17]. The sequence of preprocessing steps can be seen in Figure 2.

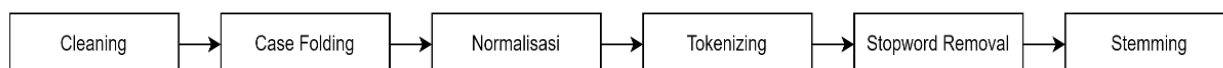


Figure 2. Pre-processing Steps

2.3 Labeling

Data that has gone through the preprocessing process will then be labeled or determined by class. This stage uses the results of stemming in preprocessing which will be labeled with the division of 2 sentiment classes, namely positive and negative. Class determination is done with a dictionary of positive and negative words in Indonesian to label sentences containing positive and negative words according to the words in the dictionary [18].

2.4 TF-IDF Extraction

Data that already has a sentiment class label will then be word-weighted. In this study, word weighting is performed using TF-IDF (Term Frequency-Inverse Document Frequency), assigning weights based on the frequency of word occurrences within a document. One method for assessing the importance of words in documents is inverse document frequency. To generate features, it is necessary to calculate the weight of each word using the TF-IDF method. This results in a value for each word that has been extracted previously [19].

2.5 Splitting Data

The next stage is the division of datasets that have gone through the TF-IDF weighting process. Based on the Pareto principle, the commonly used ratio is 80:20. The dataset is divided into 20% test data and 80% training data in the dataset. The classification model is trained using the training data, and its performance in categorizing new data is assessed using the test data. [20].

2.6 K-Nearest Neighbor(K-NN) Classifier

To perform sentiment classification, this research adopts the K-Nearest Neighbor (K-NN) algorithm. The working principle of K-NN is to compare new documents with the closest training data based on a predetermined K value [21]. The odd K value is selected to minimize the risk of overfitting and underfitting and improve classification accuracy. To determine whether neighbors are close or far away, the Euclidean distance is typically used. It can also be used to calculate the distance between two objects to assess their proximity to each other [22]. The eucliden formula as shown below.

$$d(x, y) = \sqrt{\sum_{k=1}^r (x_k - y_k)^2} \quad (1)$$

2.7 Synthetic Minority Oversampling Technique(SMOTE)

One resampling method aimed at equalizing class distributions is the SMOTE technique. SMOTE involves sampling from minority classes and inserting synthesized samples to increase the amount of data for these minority classes [23]. By giving minority classes a greater chance to be learned, the SMOTE method can improve their classification accuracy. Additionally, since there is no data reduction involved, class information is preserved. To solve the problem of class imbalance in sentiment data, the SMOTE technique is applied to the K-NN algorithm. This makes it possible to improve the representation of the minority class without significantly changing the distribution of the majority class [24].

2.8 Confusion Matrix

The Confusion Matrix is used for testing. The performance of the developed classification model is assessed using a confusion matrix. In the Confusion Matrix, the predicted class results will be compared with the actual data class. The accuracy, precision, and recall values will then be calculated based on these results [25].

3. RESULT AND DISCUSSION

Training and testing of the K-NN model was conducted using various odd k values, namely 3, 5, 7, 9, 11, 13, 15, 17, and 19. Each of these k values was systematically tested to find the most optimal k value, which is the k value that produces the highest accuracy among all tested values. After the search for the optimal k value is complete, the next step is implementing the SMOTE method. SMOTE is used to resample the current classes in order to maximize the performance of the K-NN model when there is an imbalance in the amount of data between classes. The following are the outcomes of the K-NN implementation at every stage of the study, both prior to and following the use of SMOTE.

3.1 Data Collection

Review data provided by users of the RCTI+ app on the Play Store and App Store is collected through the data scraping method, which is an automated technique of retrieving data from web pages through a specific program. As a result, 1,000 review data from the Play Store and 1,000 review data from the App Store were obtained. This data shows the development of user perception of the RCTI+ app during the period from September 17, 2019, to April 17, 2024. To facilitate data processing in the next stage of analysis, the collected review data was saved in CSV file format. The results obtained are CSV files, which contain the source, rating, review date, and review description as shown in Table 1 and Table 2.



Table 1. Playstore Data

source	rating	review date	Review deskripsi
Google Play	1	13/09/2020 00:59	aplikasinya sangat gelap. harus meningkatkan kontras hp itu akan memakan daya batrai (boros_batrai) saat mengunduh dan tiba-tiba aplikasi keluar sendiri, unduhan mengulang dari awal.
Google Play	4	16/09/2020 04:01	Aplikasinya lumayan bagus kok.... Tapi yang saya perhatikan kok setiap baru mau mulai acaranya selalu ada iklan dari aplikasinya truss kan udh ada juga iklan dari siaran tersebut tolong dong di perbaiki lagi donggg
Google Play	5	09/03/2022 09:10	Keren banget, semenjak ada aplikasi ini jadi gaperlu khawatir lagi ketinggalan film. Bisa nonton streaming nya jadi gampang, gambarnya juga jernih, suaranya jelas. Mantep pokonya.

Table 2. Appstore Data

source	rating	review date	Review deskripsi
App Store	5	02/12/2021 01:37	Aplikasi sangat membantu. Gak ketinggal acara-acara di rcti. Bisa nonton kapanpun dan dimanapun sama siapapun. Aksesnya sangat mudah gak ribet. Kalau TV lagi gangguan, tetep bisa nonton lewat HP dengan aplikasi ini. Mantap jiwa 🍷
App Store	3	12/07/2020 02:26	Aplikasinya sangat bagus ,tapi sayang kebanyakan iklan dari aplikasi mis*er ala**n.
App Store	1	19/07/2021 08:58	setelah update aplikasi terbaru saya buka seri favorit saya, tiap mau tap videonya karena mau memperbesar tiba2 ketutup sendiri.

3.2 Pre-Processing

Pre-processing comes next when the data has been satisfactorily gathered. This stage analyzes data, especially text data, to make the data more ready for further analysis. By going through the preprocessing process, the initially raw and diverse data will be transformed into a more structured format and ready for the next stage of analysis. The six steps of this preprocessing are cleaning, case folding, normalization, tokenizing, stopword removal, and stemming. Table 3 shows examples of raw data obtained from the Play Store and App Store.

Table 3. Sample Data

Process	Play Store	App Store
Data	Sukaa bgt sama aplikasi ini. Aku bisa nonton series yg kusuka, dengerin musik favorit, baca baca berita yg sedang update sekarang.. 1 aplikasi bisa untuk semua hal.. 🍷🍷	aplikasinya bagus, bermanfaat, memberi banyak sekali informasi yang berguna, smga bisa berkembang lagi lebih baik 🍷🍷🍷

a. Cleaning

The cleaning stage is the process of cleaning user reviews on the Play Store and App Store from noise by removing features such as punctuation, blank characters, and emojis that do not affect classification. The cleaning process has been implemented on raw data from the Play Store and App Store, and the results are shown in Table 4.

Table 4. Cleaning Result

Process	Play Store	App Store
Cleaning	Sukaa bgt sama aplikasi ini Aku bisa nonton series yg kusuka dengerin musik favorit baca baca berita yg sedang update sekarang aplikasi bisa untuk semua hal	aplikasinya bagus bermanfaat memberi banyak sekali informasi yang berguna smga bisa berkembang lagi lebih baik



b. Case Folding

One method of data standardization in text processing is case folding. This method aims to equalize the form of letter writing by converting all letters in the text to lowercase. This helps to reduce variation in the text and improve accuracy in subsequent processes. Implementation of case folding is done after the data is successfully cleaned, here are the results of the case folding implementation that can be seen in Table 5.

Table 5. Case Folding Result

Process	Play Store	App Store
Case Folding	sukaa bgt sama aplikasi ini aku bisa nonton series yg kusuka dengerin musik favorit baca baca berita yg sedang update sekarang aplikasi bisa untuk semua hal	aplikasinya bagus bermanfaat memberi banyak sekali informasi yang berguna smga bisa berkembang lagi lebih baik

c. Normalization

Normalization is the stage of replacing sentences consisting of slang or non-standard words with standard words according to the Big Indonesian Dictionary (KBBI). The data that has passed the case folding stage is then implemented in the normalization process. The results can be seen in Table 6, where the words marked in bold are converted into standard words.

Table 6. Normalization Result

Process	Play Store	App Store
Normalization	suka sekali sama aplikasinya memudahkan saya untuk menonton siaran ulang ataupun live apalagi untuk nonton bola lancar sekali	aplikasinya bagus bermanfaat memberi banyak sekali informasi yang berguna semoga bisa berkembang lagi lebih baik

d. Tokenizing

Tokenizing is a process that breaks sentences or strings into small parts in the form of characters or words. This process is also known as text segmentation or lexical analysis. The tokenization implementation is done after passing the normalization process. The results of tokenizing implementation on Play Store and App Store data can be seen in Table 7.

Table 7. Tokenizing Result

Process	Play Store	App Store
Tokenizing	['suka', 'sekali', 'sama', 'aplikasinya', 'memudahkan', 'saya', 'untuk', 'menonton', 'siaran', 'ulang', 'ataupun', 'live', 'apalagi', 'untuk', 'nonton', 'bola', 'lancar', 'sekali']	['aplikasinya', 'bagus', 'bermanfaat', 'memberi', 'banyak', 'sekali', 'informasi', 'yang', 'berguna', 'semoga', 'bisa', 'berkembang', 'lagi', 'lebih', 'baik']

e. Stopword Removal

At this stage, words that have no significance such as conjunctions, prepositions, and pronouns are removed. Stopword Removal aims to focus on data that has meaning. The implementation of stopword removal is done after the data has passed the tokenizing stage. The results of the stopword removal process can be seen in Table 8.

Table 8. Stopword Removal Result

Process	Play Store	App Store
Stopword Removal	['suka', 'sekali', 'aplikasinya', 'memudahkan', 'menonton', 'siaran', 'ulang', 'nonton', 'bola', 'lancar', 'sekali']	['aplikasinya', 'bagus', 'bermanfaat', 'informasi', 'berguna', 'semoga', 'berkembang']

f. Stemming

Stemming is the process of removing prefixes or suffixes from words, as well as converting conjunctions, prepositions, and pronouns into basic words. Stemming implementation is done after the stopword removal process. Data from the Play Store and App Store that has gone through the stopword removal stage is then processed using stemming, and the results can be seen in Table 9.

Table 9. Stemming Result

Process	Play Store	App Store
Stemming	suka sekali aplikasi mudah tonton siar ulang tonton ulang bola lancar sekali	aplikasi bagus manfaat informasi guna semoga kembang

3.3 Labeling

After the data has gone through preprocessing, data labeling is then carried out which is divided into two classes, namely positive and negative. Class determination is done with a dictionary of positive and negative words in Indonesian so that it can label sentences containing positive and negative words according to the words in the dictionary. With this approach, sentences containing words with positive nuances will be automatically labeled as positive, while sentences containing words with negative nuances will be labeled as negative. This process enables consistent labeling of data according to established standards and helps prepare the data for further stages of sentiment analysis.

Based on the labeling process carried out using a dictionary of positive and negative words on user reviews of streaming applications on the Play Store, the results show that as many as 618 review data are categorized as reviews with positive sentiments, while 382 other review data are categorized as reviews with negative sentiments. These labeling results provide an initial picture of user perceptions of these apps on the Play Store platform, where positive sentiment appears to be more dominant compared to negative sentiment. A visualization of these sentiment labeling results is presented in Figure 3.

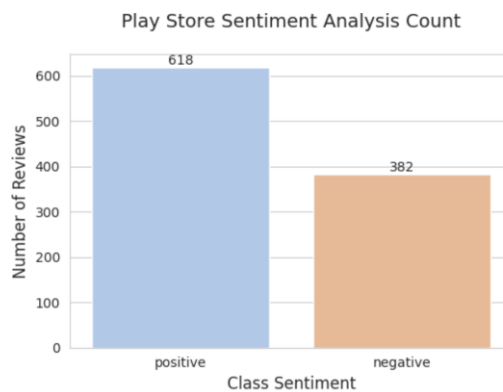


Figure 3. Visualisasi Labeling Play Store Data

Then, based on labeling using a word dictionary that identifies positive and negative words, the labeling results of user reviews of streaming apps on the App Store show that 371 reviews are categorized as having positive sentiment, while 629 reviews are categorized as having negative sentiment. Visualization of labeling on the Play Store can be seen in the Figure 4.

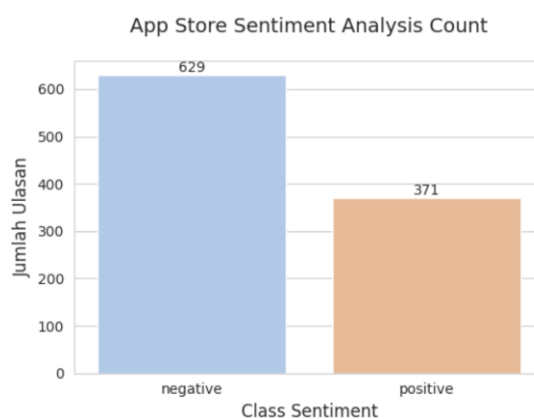


Figure 4. Visualisasi Labeling App Store Data

3.4 TF-IDF Extraction

This weighting stage converts review data into numbers or numeric to facilitate the processing of classification models, and to create data features, stemming will be used as a reference in this weighting. To generate features, it is necessary to calculate the weight of each word using the Term Frequency-Inverse Document Frequency (TF-IDF) method. This generates a value for each word that has been extracted previously. By converting text into numbers, the model can understand patterns and information and make better predictions.

3.5 Splitting Data

This research will use 2000 datasets. Based on the *Pareto Principle*, the commonly used ratio is 80:20. Therefore, the dataset is divided into 80% training data. The data splitting can be seen in Table 10.

Table 10. Splitting Data

	Training Data	Test Data
Play Store	800	200
App Store	800	200

3.6 K-Nearest Neighbor Classification

At this stage, testing of the parameter k is carried out on various KNN methods to determine the optimal parameter value. This research uses a variety of parameter values k, or the number of nearest neighbors, namely odd numbers 3 to 19. The following are the results of the implementation of the KNN method on Playstore and Appstore which can be seen in Table 11 and Table 12.

Table 11. K-NN Result on Play Store Data

K	Accuracy
3	63,5%
5	68,0%
7	70,0%
9	73,5%
11	70,5%
13	71,5%
15	73,5%
17	72,5%
19	74,0%

Table 12. K-NN Result on App Store Data

K	Accuracy
3	79,0%
5	76,0%
7	77,5%
9	79,0%
11	79,0%
13	77,5%
15	79,0%
17	79,0%
19	79,0%

Based on the results displayed in Table 11 and Table 12, it can be seen that the accuracy results generated by the K-NN model vary for each K value applied. In Table 11, which contains the model evaluation results using data from the Play Store, it can be seen that the optimal K value is found at K = 19 with an accuracy rate of 74%. Meanwhile, in Table 12, which presents the evaluation results for data from the App Store, the optimal K value was achieved at K = 15 with an accuracy of 79%. This suggests that choosing the K value depends not only on the K-NN method but also on the distribution of the data and the particulars of each source.

3.7 SMOTE Implementation

In this stage, the SMOTE method is applied to optimize the performance of the K-Nearest Neighbors (KNN) model to balance the data's sentiment class distribution. By using SMOTE, the KNN model can be more effective in recognizing patterns from both classes equally, reducing the risk of bias towards the majority class which can lead to less accurate predictions. This is expected to result in a more stable and accurate model performance in classifying sentiment, both for positive and negative classes. The results of SMOTE are depicted in Figure 5 and Figure 6.

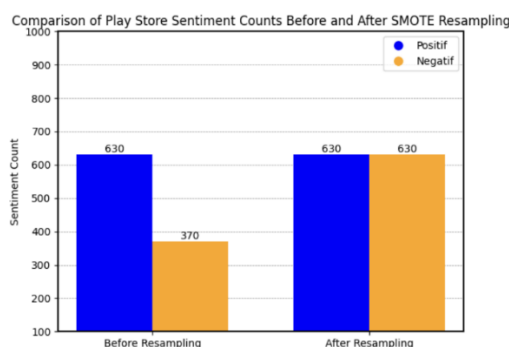


Figure 5. SMOTE Result on Play Store Data

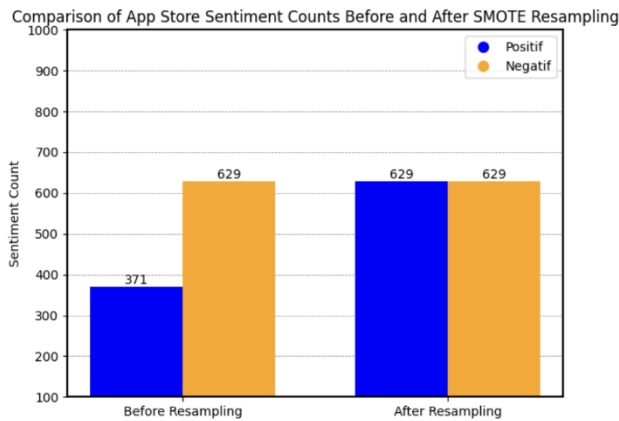


Figure 6. SMOTE Result on App Store Data

Based on the Figure 5 and Figure 6, after the resampling process, the number of sentiments is balanced, with 630 reviews for the Play Store and 629 reviews for the App Store for each sentiment, so the dataset for the Play Store is 1,260 and the dataset for the app store is 1,258. The following are the results of training the SMOTE optimization model on K-NN after the data distribution is balanced which can be seen in Table 13 and Table 14.

Table 13. SMOTE & K-NN Result on Play Store Data

K	Accuracy
3	79,4%
5	79,4%
7	81,7%
9	81,7%
11	81,3%
13	81,0%
15	81,7%
17	82,9%
19	80,6%

Table 14. SMOTE & K-NN Result on App Store Data

K	Accuracy
3	84,1%
5	84,1%
7	80,2%
9	82,1%
11	83,7%
13	81,7%
15	81,3%
17	81,0%
19	82,9%

From the Table 13 and 14, it can be seen the accuracy results generated from each K value. In the implementation of SMOTE and K-NN, the optimal K value for Playstore data (table 13) is K = 17 with 82.9% accuracy, while for Appstore data (table 14) is K = 5 with 84.1% accuracy. After the K-NN and SMOTE models are successfully implemented and the optimal K value is obtained on Playstore and Appstore data, the next step is to evaluate using the confusion matrix as shown in the Figure 7 and Figure 8.

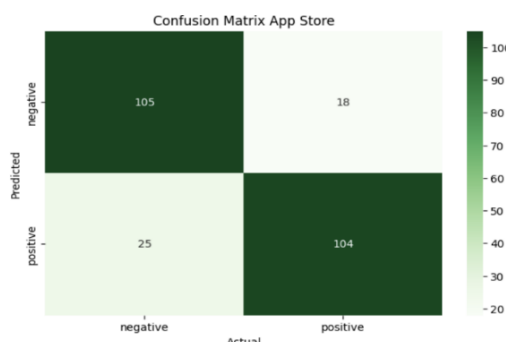


Figure 7. Confusion Matrix Play Store Data

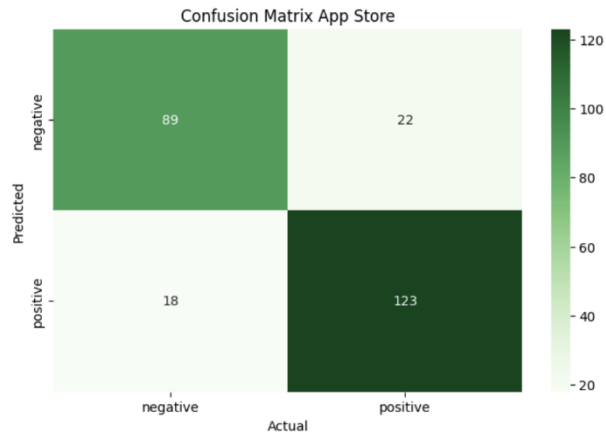


Figure 8. Confusion Matrix App Store Data

3.8 Comparison Accuracy

After obtaining the results of the K-NN model before and after the application of SMOTE, and evaluating the overall performance of the model using the confusion matrix, it is now possible to further analyze and compare the effectiveness of the model under these two different conditions. This comparison process is essential to understand the impact that SMOTE has on the model's performance, especially in addressing data imbalances that may affect the prediction results. Thus, it can evaluate the extent to which SMOTE can successfully improve the accuracy and precision of the model on minority data without compromising the performance of majority data.

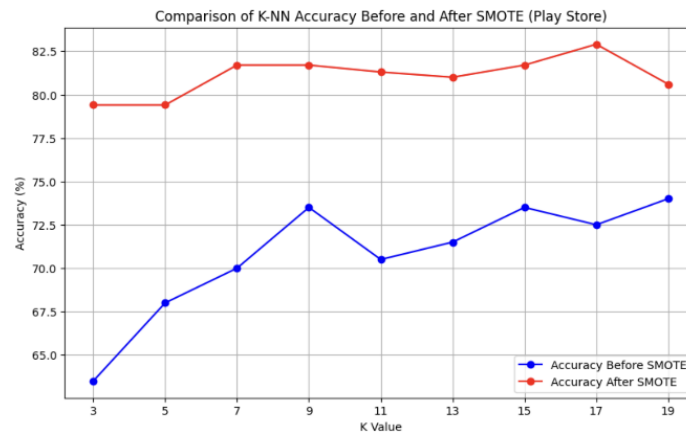


Figure 9. Comparison of K-NN Accuracy (Play Store)

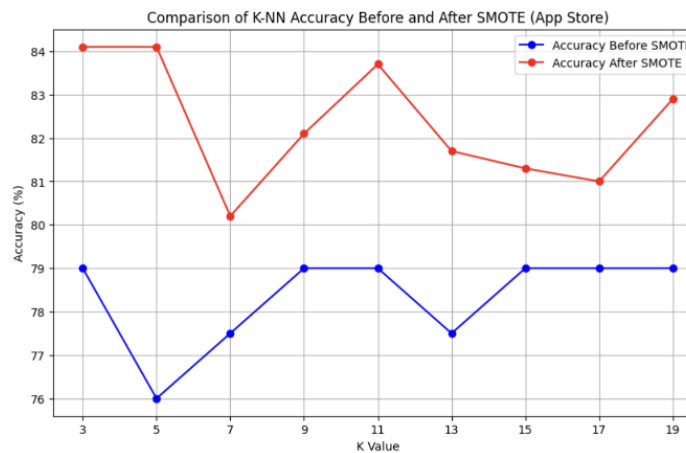


Figure 10. Comparison of K-NN Accuracy (App Store)

Figures 9 and 10 show that the application of the SMOTE resampling technique successfully improves the performance of the classification model on the Play Store and App Store datasets. Overall, there is a significant increase in accuracy after applying SMOTE, indicating that the method effectively corrects the class imbalance and enhances the model's ability to classify data. This comparison aims to highlight the positive impact of SMOTE on model accuracy by demonstrating how resampling can help models better handle imbalanced data.

4. CONCLUSION

The model classification results show that the use of the SMOTE method successfully improves the performance of the K-Nearest Neighbor (K-NN) model in analyzing user sentiment towards the RCTI+ application. In Playstore data, the model accuracy increased from 74% to 82.9%, while in Appstore data, the accuracy increased from 79% to 84.1% after SMOTE implementation. This improvement shows that the K-NN method effectively recognizes sentiment patterns based on user reviews. However, several factors still affect the accuracy of the model, such as a sufficient amount of training data and the selection of relevant features. Therefore, further optimization through setting model parameters and selecting more representative features is expected to significantly improve model performance in sentiment analysis. In addition, the analysis revealed differences in user perceptions between the two platforms with positive sentiment dominating reviews on the Play Store, while negative sentiment was more prevalent in reviews on the App Store. This conclusion suggests that streaming apps are generally well-received by users, especially Playstore users, who show a high level of satisfaction. However, the criticisms contained in negative reviews on the App Store are also important to note in order to improve the quality of apps in the future. By understanding user feedback through this sentiment analysis, developers will be able to continuously improve and customize app features to meet user expectations and enhance the overall user experience. However, this study has some limitations, such as data coverage limited to reviews, as well as the sensitivity of the K-NN model to parameter and feature selection, which may affect the accuracy of the results. Future research is recommended to expand the data source, explore other more adaptive algorithms, and refine language processing to handle variations in expressions in reviews.

REFERENCES

- [1] U. Kulsum, M. Jajuli, and N. Sulistiyowati, "Analisis Sentimen Aplikasi WETV di Google Play Store Menggunakan Algoritma Support Vector Machine," *J. Appl. Informatics Comput.*, vol. 6, no. 2, pp. 205–212, 2022, doi: 10.30871/jaic.v6i2.4802.
- [2] Bagas hutauruk dkk., "Hubungan antara Intensitas Penggunaan Aplikasi Streaming Berbayar dan Tingkat Kekayaan Konten (content richness) yang Ditawarkan dengan Loyalitas Konsumsi Aplikasi Streaming Berbayar GoPlay," *J. SUARA MERDEKA*, pp. 410–421, 2023, doi: 10.48175/ijarsct-13062.
- [3] A. D. Yuliani, "Peranan Media Streaming dalam Menggantikan Televisi Konvensional di Kalangan Masyarakat," *J. Ris. Manaj. Komun.*, vol. 3, no. 2, pp. 109–114, 2023, doi: 10.29313/jrmk.v3i2.3140.
- [4] R. R. Putra and Z. Hidayat, "Komunikasi Pemasaran Layanan Video Streaming Dan on Demand Mnc Group (Studi Kasus: Aplikasi Reti+)," *JISIP (Jurnal Ilmu Sos. dan Pendidikan)*, vol. 6, no. 1, pp. 2255–2269, 2022, doi: 10.58258/jisip.v6i1.2813.
- [5] Y. Du, S. Choe, J. Vega, Y. Liu, and A. Trujillo, "Listening to Stakeholders Involved in Speech-Language Therapy for Children With Communication Disorders: Content Analysis of Apple App Store Reviews," *JMIR Pediatr. Parent.*, vol. 5, no. 1, 2022, doi: 10.2196/28661.
- [6] J. A. Rieuwpassa, S. Sugito, and T. Widiharih, "Implementasi Metode Naive Bayes Classifier Untuk Klasifikasi Sentimen Ulasan Pengguna Aplikasi Netflix Pada Google Play," *J. Gaussian*, vol. 12, no. 3, pp. 362–371, 2024, doi: 10.14710/j.gauss.12.3.362-371.
- [7] H. Wisnu, M. Afif, and Y. Ruldevyani, "Sentiment analysis on customer satisfaction of digital payment in Indonesia: A comparative study using KNN and Naive Bayes," *J. Phys. Conf. Ser.*, vol. 1444, no. 1, 2020, doi: 10.1088/1742-6596/1444/1/012034.
- [8] D. K. and S. A. K. Hulliyah, A. M. Almaisah, F. Mintarsih, S. U. Masrurah, "Analysis of Public Sentiment Using The K-Nearest Neighbor (k-NN) Algorithm and Lexicon Based on Indonesian Television Shows on Social Media Twitter," *Int. Conf. Cyber IT Serv. Manag.*, 2022, doi: 10.1109/CITSMS56380.2022.9936011.
- [9] S. Albahli, "Twitter sentiment analysis: An Arabic text mining approach based on COVID-19," *Front. Public Heal.*, vol. 10, no. October, pp. 1–13, 2022, doi: 10.3389/fpubh.2022.966779.
- [10] D. A. Kristiyanti, S. A. Sanjaya, V. C. Tjokro, and J. Suhali, "Dealing imbalance dataset problem in sentiment analysis of recession in Indonesia," *IAES Int. J. Artif. Intell.*, vol. 13, no. 2, pp. 2058–2070, 2024, doi: 10.11591/ijai.v13.i2.pp2060-2072.
- [11] F. F. Supeli and Setiaji, "Klasifikasi Sentimen Positif Dan Negatif Pada Aplikasi Vidio Dengan Algoritma K-Nearest Neighbor," *Indones. J. Comput. Sci.*, vol. 2, no. 1, pp. 7–15, 2023, doi: 10.31294/ijcs.v2i1.1874.
- [12] S. Nuraeni, S. P. A. Syam, M. F. Wajdi, B. Firmansyah, and M. Malkan, "Implementasi Metode K-NN Untuk Menentukan Jurusan Siswa di SMAN 02 Manokwari," *G-Tech J. Teknol. Terap.*, vol. 7, no. 1, pp. 89–95, 2023, doi: 10.33379/gtech.v7i1.1905.
- [13] M. F. El Firdaus, N. Nurfaizah, and S. Sarmini, "Analisis Sentimen Tokopedia Pada Ulasan di Google Playstore Menggunakan Algoritma Naive Bayes Classifier dan K-Nearest Neighbor," *JURIKOM (Jurnal Ris. Komputer)*, vol. 9, no. 5, p. 1329, 2022, doi: 10.30865/jurikom.v9i5.4774.
- [14] O. Chamorro-Atalaya *et al.*, "Student Satisfaction Classification Algorithm Using the Minority Synthetic Oversampling Technique," *Int. J. Inf. Educ. Technol.*, vol. 13, no. 7, pp. 1094–1100, 2023, doi: 10.18178/ijiet.2023.13.7.1909.
- [15] J. Soni, S. Xavier, R. Saxena, and P. Waghela, "Leveraging Web Scraping for Aspect-Based Sentiment Analysis: A Case Study on Flipkart Customer Reviews of Mobile Phones," *Int. J. Res. Publ. Rev.*, vol. 5, no. 6, pp. 6747–6752, 2024, doi: 10.55248/gengpi.5.0624.1638.
- [16] S. Raheja and A. Asthana, "Sentiment Analysis of Tweets During the COVID-19 Pandemic Using Multinomial Logistic Regression," *Int. J. Softw. Innov.*, vol. 11, no. 1, pp. 1–16, 2022, doi: 10.4018/IJSI.315740.
- [17] K. Adrian Manurung, "Sentiment Analysis of Tourist Attraction Review from TripAdvisor Using CNN and LSTM," *Int. J. Inf. Commun. Technol.*, vol. 9, no. 1, pp. 73–85, 2023, doi: 10.21108/ijoi.v9i1.756.
- [18] R. L. Mustofa and B. Prasetyo, "Sentiment analysis using lexicon-based method with naive bayes classifier algorithm on



- #newnormal hashtag in twitter,” *J. Phys. Conf. Ser.*, vol. 1918, no. 4, 2021, doi: 10.1088/1742-6596/1918/4/042155.
- [19] F. Westin, “Time Period Categorization in Fiction: A Comparative Analysis of Machine Learning Techniques,” *Cat. Classif. Q.*, vol. 62, no. 2, pp. 124–153, 2024, doi: 10.1080/01639374.2024.2315548.
- [20] E. Serradell-Lopez, P. Lara-Navarra, and S. Martínez-Martínez, “The Pareto Principle in virtual communities of learning,” *Comput. Human Behav.*, vol. 138, no. August 2022, p. 107444, 2023, doi: 10.1016/j.chb.2022.107444.
- [21] H. Herman, I. Riadi, and Y. Kurniawan, “Vulnerability Detection With K-Nearest Neighbor and Naïve Bayes Method using Machine Learning,” *Int. J. Artif. Intell. Res.*, vol. 7, no. 1, p. 10, 2023, doi: 10.29099/ijair.v7i1.795.
- [22] S. Bengesi, T. Oladunni, R. Olusegun, and H. Audu, “A Machine Learning-Sentiment Analysis on Monkeypox Outbreak: An Extensive Dataset to Show the Polarity of Public Opinion From Twitter Tweets,” *IEEE Access*, vol. 11, no. February, pp. 11811–11826, 2023, doi: 10.1109/ACCESS.2023.3242290.
- [23] N. Sigeeef, “An Oversampling Algorithm combining SMOTE and RF for Imbalanced Medical Data,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 11, no. 6, pp. 2429–2434, 2023, doi: 10.22214/ijraset.2023.54074.
- [24] A. X. Wang, S. S. Chukova, and B. P. Nguyen, “Synthetic minority oversampling using edited displacement-based k-nearest neighbors,” *Appl. Soft Comput.*, vol. 148, no. October, p. 110895, 2023, doi: 10.1016/j.asoc.2023.110895.
- [25] M. Rahardi, A. Aminuddin, F. F. Abdulloh, and R. A. Nugroho, “Sentiment Analysis of Covid-19 Vaccination using Support Vector Machine in Indonesia,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 6, pp. 534–539, 2022, doi: 10.14569/IJACSA.2022.0130665.