Vol. 8, No. 4, August 2026

ISI Journal of Information Systems and Informatics

Published By Asosiasi Doktor Sistem Informasi Indonesia

Comparative Sentiment Analysis of Profits Anywhere Application Reviews Using KNN and Naïve Bayes

Yurindra[1], Chandra Kirana[2], Dian Novianto[3], Tri Sugihartono[4], Muhammad Ilham[5]

[1,2,3,4,5]Informatics Departement, Institut Sains dan Bisnis Atma Luhur, Pangkalpinang, Indonesia

Received:
February 11, 2026
Revised:
May 17, 2026
Accepted:
June 24, 2026
Published:
August 22, 2026

Corresponding Author:
Author Name*:
Yurindra
Email*:
yurindra@atmaluhur.ac.id

DOI:
10.63158/journalisi.v8i4.1658

Abstract. User reviews on Google Play Store provide valuable information regarding application service quality and user satisfaction. This study aims to compare the performance of the K-Nearest Neighbor (KNN) and Naïve Bayes algorithms for sentiment analysis of reviews of the Profits Anywhere application, a digital business platform offering affiliate marketing and financial services. A total of 1,037 reviews were collected through web scraping and subsequently underwent data cleaning and validation, resulting in 957 valid reviews used in the analysis, consisting of 674 positive and 283 negative reviews. The dataset was subjected to a comprehensive text preprocessing pipeline, including text cleaning, case folding, tokenization, stopword removal, and stemming, followed by feature representation using the Term Frequency–Inverse Document Frequency (TF-IDF) technique. The data were partitioned using an 80:20 stratified train–test split, while hyperparameter optimization was conducted using GridSearchCV with 5-fold stratified cross-validation. Experimental results demonstrate that KNN outperformed Naïve Bayes on the evaluated dataset and experimental configuration, achieving an accuracy of 92.19%, weighted precision of 92.23%, weighted recall of 92.19%, and weighted F1-score of 92.21%. In contrast, Naïve Bayes achieved an accuracy of 88.54%, weighted precision of 89.04%, weighted recall of 88.54%, and weighted F1-score of 88.70%. These findings provide empirical evidence regarding algorithm selection for sentiment analysis of application reviews in the fintech and digital business domains.

Keywords: Sentiment Analysis, K-Nearest Neighbor, Naïve Bayes, Fintech Application, Google Play Store Reviews

1. INTRODUCTION

The rapid advancement of digital technology has driven the transformation of various business sectors through the adoption of mobile-based applications. Today, mobile applications are utilized not only as communication tools but also as platforms that support business operations, digital marketing activities, financial transactions, and business management in a more effective and efficient manner. As the number of application users continues to grow, user reviews submitted on application distribution platforms such as Google Play Store have become a valuable source of information for evaluating service quality, identifying user needs, and supporting data-driven decision-making for application developers [1]. The development of sentiment analysis for mobile application reviews has become an increasingly important research topic, as it enables developers to automatically understand user experiences through the analysis of large volumes of review data. Previous studies have demonstrated that machine learning approaches based on Term Frequency–Inverse Document Frequency (TF-IDF) remain effective and widely adopted for sentiment classification of Google Play Store reviews due to their ability to generate lightweight, interpretable, and computationally efficient feature representations. [2], [3]. Furthermore, application reviews have been recognized as an important source of information for identifying functional issues, evaluating service quality, and assessing user satisfaction with digital applications [4].

Profits Anywhere is a digital business application that provides various services to support affiliate marketing activities, digital business management, and transactions made through mobile devices [5]. As an application that interacts directly with users, the level of user satisfaction is one of the important factors that affect the success and sustainability of services. Users generally convey experiences, complaints, criticisms, and appreciation through the review feature on the Google Play Store. These reviews contain valuable information to understand user perceptions of the quality of services provided. The increasing number of reviews causes the manual analysis process to be less effective because it requires a large amount of time and resources. Therefore, sentiment analysis is one of the widely used approaches to automatically identify user opinions [6]. Sentiment analysis allows the process of classifying opinions into categories of positive and negative sentiment so that it can help application developers understand the level of user satisfaction and identify aspects of the service that need to be improved [7] [1].

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

Various machine learning methods have been used in sentiment analysis research. Naïve Bayes is one of the most widely used algorithms because it has a simple, efficient computational process, and is able to produce good performance in text classification [8] [9] [10]. On the other hand, K-Nearest Neighbor (KNN) is also widely applied to sentiment analysis because it is able to classify based on the similarities between documents in the feature space [11] [12]. Several studies have shown that KNN is able to produce better performance on datasets that have strong local similarity patterns [13], while other studies show that Naïve Bayes has an advantage in terms of computational efficiency and classification stability [14] [15] . These differences in results show that the algorithm's performance is greatly influenced by the characteristics of the dataset, feature representation, and problem domain used.

Previous sentiment analysis research generally focused on reviews of e-commerce applications, online transportation, social media, financial services, and educational applications [16] [17] [18] [19]. In addition, various studies have compared the performance of several classification algorithms using TF-IDF-based feature representations [12]. However, studies that specifically analyze the sentiment of user reviews of digital business applications such as Profits Anywhere are still relatively limited. In addition, many previous studies used different preprocessing procedures, feature extraction methods, and evaluation configurations so that the results obtained are difficult to compare directly.

Based on these conditions, this study uses a uniform experimental framework through the stages of text preprocessing, feature extraction using TF-IDF, parameter optimization using GridSearchCV, and the same evaluation procedure on the KNN and Naïve Bayes algorithms. This approach is carried out to obtain a more objective comparison of performance on the same dataset and experimental conditions. Based on the research gaps that have been identified, this study asks the research question which classification algorithm, K-Nearest Neighbor (KNN) or Naïve Bayes, achieves better sentiment classification performance on Profits Anywhere user reviews when both models are evaluated under the same pre-processing pipeline and TF-IDF feature representation?. Therefore, the purpose of this study is to compare the performance of K-Nearest Neighbor (KNN) and Naïve Bayes algorithms in performing sentiment classification on Profits Anywhere app user reviews. The contribution of this study lies in the comparative

evaluation of two classification algorithms that have been optimized using GridSearchCV on the digital business application review dataset with a TF-IDF feature representation and consistent evaluation procedures. The results of the study are expected to provide empirical evidence supporting the selection of sentiment classification algorithms in the digital business application domain.

2. METHODS

2.1 Research Framework

This study uses an experimental quantitative approach to compare the performance of the K-Nearest Neighbor (KNN) and Naïve Bayes algorithms in the sentiment classification of Profits Anywhere application reviews. The research stages consist of data collection, sentiment labeling, text preprocessing, feature extraction using TF-IDF, model training and optimization, and model performance evaluation. The research flow used can be seen in Figure 1.

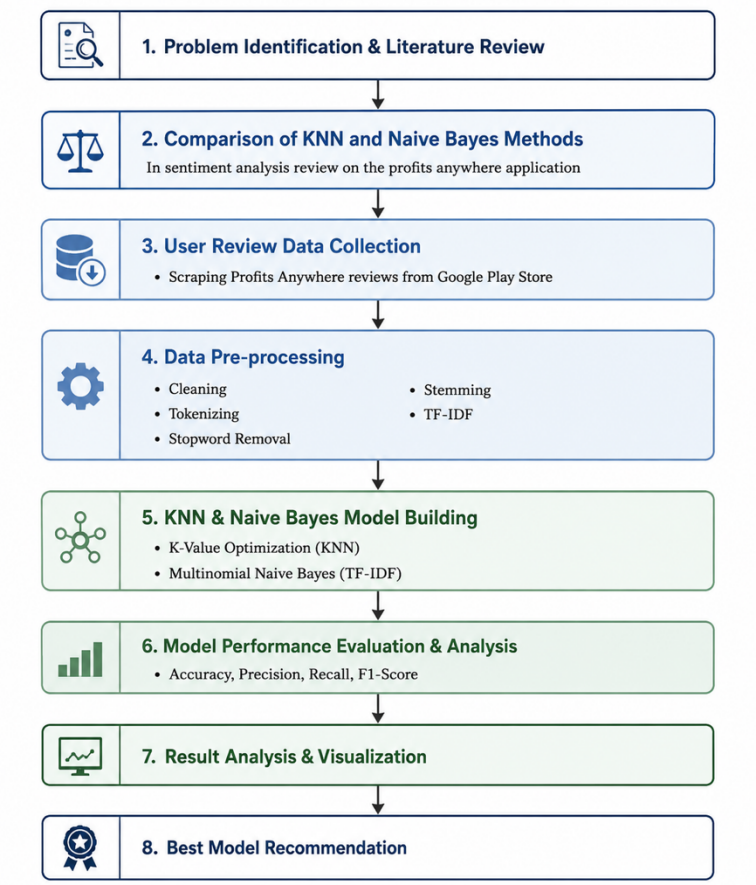

Figure 1. Research Flow Chart

2.2 Data Collection

The research data were collected from user reviews of the Profits Anywhere application available on Google Play Store using the Google Play Scraper library. The data collection process yielded a total of 1,037 user reviews. The collected data subsequently underwent a selection and validation process based on the following criteria: (1) removal of duplicate reviews, (2) removal of empty reviews, (3) removal of reviews containing only symbols or numerical characters, (4) removal of reviews with insufficient textual information, and (5) exclusion of reviews that could not be adequately interpreted due to the use of mixed languages or languages other than Indonesian. Following the data cleaning and validation procedures, a total of 957 reviews were retained for analysis in this study. Table 1 presents a sample of the initial dataset used in the research. The final dataset consisted of 674 positive reviews (70.43%) and 283 negative reviews (29.57%).

Table 1. Initial Dataset

No	content	score
1	The application often errors. For real-time trading, old mistakes at the time of trading are fatal. If you are on vacation, it is not a problem. This is in trading hours and is not even accessible. less recommended. If you're looking for a low-cost broker, look for someone else.	2
2	1 star used to be until there was a reply and a rebuttal…	1
3	Profit Anywhere is an app that's very easy to use.	5
4	Isn't there a Portfolio Share feature? You can't show it, you can't tell everyone.	2
5	Often error when logging in, klau deposit gk langsuk sometimes take a long time to be fixed	2
.....	.....	.....
1033	The better the interface and the important features are already included	5
1034	The application often errors. for real-time trading.	2
1035	The profits are well worth it	5
1036	With a friendly appearance, but maximum functionality	5
1037	The worse the performance is	1

2.3 Sentiment Labelling

Sentiment labeling was conducted based on the star ratings available on Google Play Store, which were collected simultaneously during the data scraping process. This approach has been widely adopted in sentiment analysis research because user-provided ratings directly reflect their level of satisfaction with an application. In this study, reviews with 4- and 5-star ratings were classified as positive sentiment, while reviews with 1- and 2-star ratings were classified as negative sentiment. Reviews with a 3-star rating were excluded from the analysis because they were considered to represent neutral or ambiguous opinions that could not be reliably assigned to either the positive or negative category. Following the labeling and data validation processes, a total of 957 reviews were retained, consisting of 674 positive reviews (70.43%) and 283 negative reviews (29.57%). This class distribution served as the basis for the training and evaluation of the sentiment classification models.

Table 2. Data Labelling Results

No	content	score	Label
1	The application often errors. For real-time trading, old mistakes at the time of trading are fatal. If you are on vacation, it is not a problem. This is in trading hours and is not even accessible. less recommended. If you're looking for a low-cost broker, look for someone else.	2	Negative
2	1 star used to be until there was a reply and a rebuttal...	1	Negative
3	Profit Anywhere is an app that's very easy to use.	5	Positive
4	Isn't there a Portfolio Share feature? You can't show it, you can't tell everyone.	2	Negative
.....	.....	.....	
1034	The application often errors. for real-time trading.	2	Negative
1035	The profits are well worth it	5	Positive
1036	With a friendly appearance, but maximum functionality	5	Positive
1037	The worse the performance is	1	Negative

2.4 Text Preprocessing

The text preprocessing stage consisted of several steps, including text cleaning, case folding, informal word normalization, tokenization, stopword removal, and stemming. Informal expressions commonly found in user reviews, such as *"ga," "gak," "nggak,"* and *"ngga,"* were normalized to the standard Indonesian form *"tidak"* ("not"). Negation terms, including *"tidak"* (not), *"bukan"* (not/other than), *"jangan"* (do not), and *"belum"* (not yet), were preserved during the stopword removal process because they have a direct influence on sentiment polarity. Emojis, emoticons, URLs, numerical values, and special characters were removed during the cleaning stage, as they were not utilized as features in this study. These preprocessing procedures follow commonly adopted practices in machine learning–based sentiment analysis research, as they effectively reduce textual noise and improve the quality of text feature representations [20].

2.5 Feature Extraction Using TF-IDF

Text features were extracted using TF-IDF. The vectorizer was configured using unigram and bigram representations (ngram_range = (1,2)), minimum document frequency (min_df = 2), maximum document frequency (max_df = 0.9), and sublinear term-frequency scaling. Vocabulary fitting and inverse document frequency estimation were performed exclusively on the training data to prevent information leakage. Formally, the TF-IDF weight of a term $t$in a document $d$is defined as shown in Equation 1.

$$\text{TF-IDF}_{t,d} = \text{TF}_{t,d} \times \text{IDF}_t \tag{1}$$

where the term frequency TF(t,d)is calculated as shown in Equation 2.

$$TF(t,d) = \frac{f_{t,d}}{\sum_k f_{k,d}} \tag{2}$$

and the inverse document frequency IDF(t)is defined as shown in Equation 3.

$$IDF(t) = \log\left(\frac{N}{n_t}\right) \tag{3}$$

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

2.6 Experimental Configuration

To ensure research replication and transparency of the experimental process, Table 7 presents a summary of the configurations used in this study.

Table 3. Experiment Configuration Summary

Components	Configuration
Number of Datasets	957 Reviews
Positive Reviews	674
Negative Reviews	283
Labeling Methods	Google Play Store rating-based labeling
Data Training	80% (765 Reviews)
Data Testing	20% (192 Reviews)
Random_state	42
Methods Split	Stratified Train-Test Split
Feature Extraction	TF-IDF
ngram_range	-1,2
min_df	2
max_df	0,9
sublinear_tf	TRUE
Cross-Validation	Stratified 5-Fold Cross Validation
Optimasi Parameter	GridSearchCV
KNN Search Space	k={3,5,7,9}
KNN Metric	cosine, euclidean
KNN Weighting	uniform, distance
Naïve Bayes Alpha	{0,1; 0,3; 0,5; 1,0}
Evaluation Metrics	Accuracy, Precision, Recall, F1-Score

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

2.7 Model Development and Hyperparameter Optimization

The dataset was divided into training and testing sets using an 80:20 stratified split with random_state = 42. Hyperparameter optimization for KNN was performed using GridSearchCV within the training partition only. The search space as shown in Table 4.

Tabel 4. Parameter KNN

Parameter	Value
n_neighbors	3, 5, 7, 9
weights	uniform, distance
metric	cosine, euclidean

Model selection was conducted using stratified 5-fold cross-validation with accuracy as the optimization criterion. The optimal KNN configuration was: k = 7, cosine similarity, and distance weighting. For Multinomial Naïve Bayes, the smoothing parameter α was optimized using GridSearchCV as shown in Table 5. Therefore, α = 0.1 was obtained through hyperparameter optimization and was not manually selected. The entire Grid Search process is done only on training data to avoid evaluation bias.

Tabel 5. Parameter Naïve Bayes

Parameter	Value
alpha	0,1 ; 0,3 ; 0,5 ; 1,0

2.8 Evaluation Metrics

Model performance was evaluated using Accuracy, Precision, Recall, and F1-Score as shown in Equation 4 to 7. Weighted-average metrics were reported to account for class imbalance. The confusion matrix was also used to analyze classification performance for each sentiment class.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (5)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (6)$$

$$F1 = 2\ x \frac{Precision\ x\ Recall}{Precision+Recall} \quad (7)$$

3. RESULTS AND DISCUSSION

3.1 Dataset Description

The dataset used in this study was obtained from user reviews of the Profits Anywhere mobile application available on the Google Play Store. Initially, a total of 1,037 reviews were collected. Because raw online reviews may contain duplicated entries, incomplete content, irrelevant text, or invalid records, a systematic data-cleaning procedure was performed before the reviews were used for sentiment classification. The filtering process included duplicate removal, elimination of incomplete or empty reviews, and validation of textual content to ensure that each record contained meaningful information suitable for further analysis. After the cleaning and validation stages, 957 valid reviews were retained as the final dataset. Based on their sentiment labels, the dataset consisted of 674 positive reviews (70.43%) and 283 negative reviews (29.57%). This distribution indicates a moderate class imbalance, with positive reviews representing approximately 2.38 times the number of negative reviews. Nevertheless, both sentiment categories contained sufficient observations to support supervised classification and performance evaluation.

To ensure that the original class proportions were maintained during model development, the dataset was divided using stratified sampling. This strategy is particularly important for imbalanced datasets because a conventional random split may produce substantially different sentiment distributions between the training and testing subsets. By preserving approximately the same positive-to-negative ratio in both subsets, the resulting evaluation provides a more representative assessment of the model's ability to classify each sentiment category. An 80:20 training–testing split was applied. Consequently, 765 reviews were assigned to the training set, while the remaining 192 reviews were used as the independent testing set. The training set contained 539 positive and 226 negative reviews, whereas the testing set contained 135 positive and 57 negative reviews. The training data were used to learn the relationship between textual features and sentiment labels, while the testing data were reserved for evaluating the model's generalization performance on previously unseen reviews. The complete dataset distribution is presented in Table 6.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

Table 6. Dataset Distribution

Dataset	Positive	Negative	Total
Training Set	539	226	765
Testing Set	135	57	192
Total	674	283	957

The use of a stratified partition also ensured that approximately 70% of the reviews in both subsets remained positive and 30% remained negative, closely reflecting the distribution of the original cleaned dataset. Maintaining this distribution reduces the risk that differences in model performance are caused by disproportionate class representation between the training and testing data.

3.2 Preprocessing

User-generated reviews obtained from online application platforms commonly contain linguistic noise, such as inconsistent capitalization, punctuation, repeated symbols, informal expressions, abbreviations, and non-standard vocabulary. If these variations are processed directly, they may unnecessarily increase the dimensionality of the feature space and reduce the effectiveness of the sentiment classification model. Therefore, a structured text preprocessing pipeline was applied to transform the raw reviews into a more standardized representation before feature extraction and classification. The preprocessing procedure consisted of text cleaning, case folding, normalization, tokenization, stopword removal, and stemming. Each stage was designed to reduce irrelevant textual variation while preserving words that contribute to sentiment information.

First, text cleaning was performed to remove elements that did not contribute directly to sentiment interpretation, including unnecessary punctuation marks, special characters, URLs, excessive whitespace, and other non-textual symbols. For example, punctuation such as exclamation marks was removed so that different surface forms of the same expression would be represented consistently. Second, case folding converted all alphabetic characters into lowercase. Without this operation, terms such as Good, GOOD, and good could be interpreted as different features even though they have the same semantic meaning. Converting the text into lowercase therefore reduces redundant

vocabulary and produces a more consistent feature representation. Third, word normalization was applied to convert informal, abbreviated, or non-standard terms into their standard lexical forms. This step is important for reviews written in informal language because users frequently employ shortened expressions or colloquial vocabulary. For example, Indonesian informal negative expressions such as “ga” and “gak” can be normalized into the standard form “tidak”. Normalization allows semantically equivalent words to be represented by a common token.

The normalized text was subsequently processed through tokenization, in which each review sentence was divided into individual lexical units or tokens. These tokens constitute the basic units used during subsequent text processing and feature extraction. Next, stopword removal was performed to eliminate frequently occurring functional words that generally provide limited discriminatory information for sentiment classification. However, this process was implemented carefully because some commonly occurring terms may carry important sentiment information. In particular, negation terms were retained, because words such as “not” or “tidak” can reverse the polarity of an expression. For example, “good” generally indicates positive sentiment, whereas “not good” conveys negative sentiment. Removing negation words could therefore alter the semantic meaning of a review and adversely affect classification performance. Finally, stemming was applied to reduce morphological variants of words to their corresponding root forms. For instance, words with different affixes but similar semantic bases can be represented by a single root term. This process reduces vocabulary redundancy and helps the feature extraction method identify related expressions as the same underlying feature. An example of the transformation performed at each preprocessing stage is presented in Table 7.

Table 7. Text Preprocessing

Stage	Output
Original Review	Very good application, very helpful for my business!!
Cleaning	Very good application very helpful for my business
Case Folding	Very good application very helpful in my business
Tokenization	app, good, really, very, helpful, business, me
Stopword Removal	app, good, helpful, business
Stemming	Application, Good, Help, Business

The resulting tokens provide a cleaner and more compact textual representation than the original reviews. The preprocessing procedure reduces unnecessary lexical variation while retaining sentiment-bearing information required by the classification model. The processed corpus was subsequently used as the input for text feature extraction, enabling each review to be transformed into a numerical representation suitable for machine-learning-based sentiment classification.

A particularly important consideration in this preprocessing pipeline is the treatment of negation. Informal negative terms such as "ga" and "gak" were normalized into their standard equivalent, "tidak", rather than being discarded. Similarly, negation words were excluded from the stopword-removal list. This decision was made because negation directly influences sentiment polarity and can substantially change the interpretation of adjacent words. Consequently, preserving normalized negation terms helps maintain the semantic meaning of user reviews and improves the reliability of the sentiment representation used in subsequent classification stages.

3.3 TF-IDF Feature Representation

Following the text preprocessing stage, the cleaned reviews were converted into numerical feature vectors using Term Frequency–Inverse Document Frequency (TF-IDF). This transformation was required because conventional machine-learning algorithms cannot directly process textual data. TF-IDF assigns a numerical weight to each term according to its frequency within a particular document and its relative rarity across the entire collection of documents. Consequently, terms that occur frequently in a specific review but are less common across the corpus receive higher weights, whereas highly frequent terms appearing in many reviews receive lower weights.

In this study, the TF-IDF vectorizer was configured using both unigram and bigram features with ngram_range = (1,2). Unigrams represent individual words, whereas bigrams capture combinations of two consecutive words. The inclusion of bigrams allows the model to retain a limited amount of contextual information that may be lost when individual words are considered independently. This is particularly relevant to sentiment analysis because expressions consisting of multiple words can carry different semantic meanings from their individual components.

To reduce sparse and potentially uninformative features, min_df = 2 was applied, meaning that a term had to occur in at least two review documents to be included in the vocabulary. In addition, max_df = 0.9 excluded terms occurring in more than 90% of the documents because such terms generally provide limited discriminatory information. The parameter sublinear_tf = True was also employed to replace raw term frequency with logarithmically scaled term frequency. This prevents highly repeated terms within a single review from disproportionately dominating the resulting feature representation.

After feature extraction, the resulting TF-IDF matrix had dimensions of 957 × 1,767, indicating that the 957 preprocessed reviews were represented using 1,767 textual features. Each row of the matrix corresponds to one review, while each column represents a unigram or bigram feature identified from the corpus. The resulting matrix was subsequently used as the numerical input for the sentiment classification model. Figure 2 presents the 20 most prominent TF-IDF features identified from the review corpus.

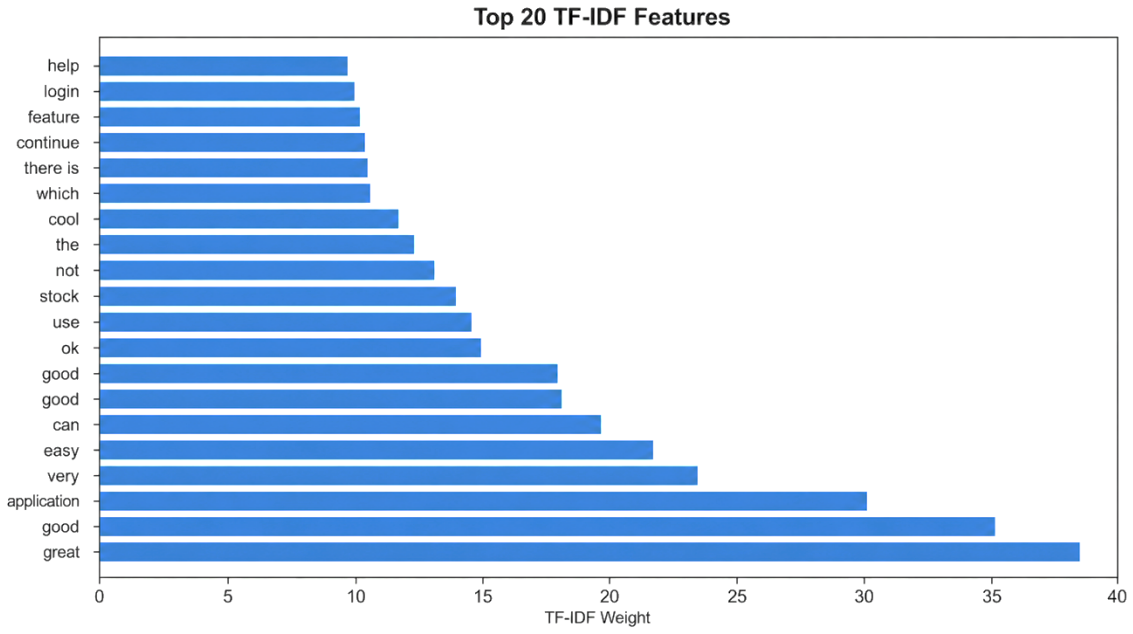

Figure 2. Top 20 Features of TF-IDF

The distribution shown in Figure 2 indicates that several terms received relatively high TF-IDF weights. Among the most prominent terms were “great,” “good,” “application,” “very,” “easy,” and “can.” Their relatively high weights indicate that these expressions occur prominently within particular reviews while remaining sufficiently distinctive across the

overall corpus. Several of these terms also have clear semantic relationships with users' evaluations of the Profits Anywhere application. For instance, terms such as “great,” “good,” and “easy” are commonly associated with favorable user experiences, whereas “application” represents the principal object being evaluated.

However, a high TF-IDF weight should not be interpreted directly as evidence that a feature is exclusively associated with either positive or negative sentiment. TF-IDF measures the statistical prominence of a term within the corpus rather than its direct relationship with a sentiment class. The actual contribution of these features to positive or negative sentiment classification is determined during the subsequent model-training process. Thus, TF-IDF serves primarily as a feature representation mechanism that transforms textual reviews into informative numerical vectors from which the classifier learns sentiment-related patterns.

The use of both unigram and bigram representations is particularly beneficial when sentiment depends on short contextual expressions. For example, an individual term such as “good” may indicate positive sentiment, whereas its combination with a negation term, such as “not good,” may represent negative sentiment. Because negation terms were retained during preprocessing and bigram features were incorporated into the TF-IDF representation, such contextual relationships can potentially be preserved in the feature space. This design supports a more meaningful representation of user opinions prior to sentiment classification.

3.4 K-Nearest Neighbor Classification

The K-Nearest Neighbor (KNN) classifier was employed to classify the TF-IDF representations of user reviews into positive and negative sentiment categories. Because KNN performance is highly dependent on the number of neighbors, distance measurement, and neighbor-weighting strategy, hyperparameter optimization was performed before evaluating the classifier on the independent testing set.

Hyperparameter tuning was conducted using GridSearchCV with five-fold stratified cross-validation applied exclusively to the training data. Stratified cross-validation was selected to preserve approximately the same proportion of positive and negative reviews across each validation fold. This procedure provides a more reliable estimate of model

performance during hyperparameter selection, particularly because the dataset contains a moderate class imbalance. During the optimization process, several combinations of KNN parameters were evaluated. The best-performing configuration consisted of seven nearest neighbors (n_neighbors = 7), cosine distance, and distance-based weighting (weights = distance), as presented in Table 8.

Table 8. Best KNN Configuration

Parameters	Value
n_neighbors	7
metric	cosine
weights	distance

The selection of seven neighbors provides a balance between sensitivity to local observations and robustness against noise. A very small number of neighbors may cause the classifier to become overly influenced by individual training samples, whereas an excessively large neighborhood may reduce its ability to capture local sentiment patterns. The use of cosine distance is particularly appropriate for TF-IDF representations because textual documents are represented as high-dimensional sparse vectors. Cosine-based similarity focuses primarily on the orientation of two document vectors rather than their absolute magnitude. Consequently, reviews containing similar patterns of important words may be considered close even when they differ in length or overall word frequency.

Furthermore, the use of distance-based weighting gives greater influence to neighboring training samples that are more similar to the review being classified. Therefore, closer documents contribute more strongly to the sentiment prediction than more distant neighbors. After determining the optimal hyperparameters, the selected KNN configuration was retrained using the complete training set and subsequently evaluated on the 192 previously unseen testing reviews. The resulting confusion matrix is presented in Figure 3.

The confusion matrix provides a class-level representation of the classifier's predictions by distinguishing correctly classified positive and negative reviews from misclassified observations. This analysis complements the aggregate performance metrics by showing

how classification errors are distributed between the two sentiment categories. The overall classification performance of KNN is summarized in Table 10.

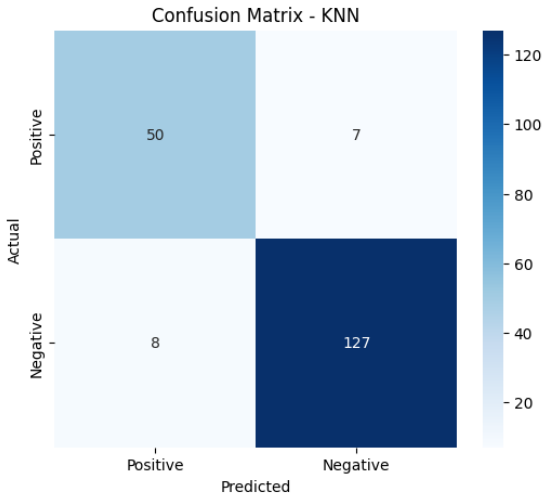

Figure 3. Confusion of Matrix-KNN

Table 10. KNN Classification Results

Metrics	Value (%)
Accuracy	92,19
Precision (Weighted)	92,23
Recall (Weighted)	92,19
F1-Score (Weighted)	92,21

The KNN classifier achieved an accuracy of 92.19%, indicating that more than nine out of ten testing reviews were correctly assigned to their corresponding sentiment classes. The model also produced a weighted precision of 92.23%, weighted recall of 92.19%, and weighted F1-score of 92.21%. The relatively similar values across accuracy, weighted precision, weighted recall, and weighted F1-score indicate stable overall predictive performance. In particular, the weighted F1-score demonstrates that the model maintains a favorable balance between precision and recall after accounting for the different numbers of samples in each sentiment class.

These results suggest that the combination of TF-IDF representation and cosine-based KNN provides an effective mechanism for identifying similarities among user reviews.

Reviews containing comparable lexical patterns are positioned closer to one another within the TF-IDF feature space, enabling KNN to infer sentiment based on the labels of neighboring documents. Nevertheless, because weighted metrics are influenced by class frequency, they should not be interpreted as direct evidence that positive and negative reviews are classified equally well. The confusion matrix and class-specific precision, recall, and F1-scores provide more appropriate evidence for assessing performance on the minority negative class.

3.5 Naïve Bayes Classification Results

A Naïve Bayes classifier was also evaluated as an alternative probabilistic approach for sentiment classification. Naïve Bayes is widely employed in text classification because it can efficiently process high-dimensional and sparse textual representations such as TF-IDF features. Similar to the KNN experiment, hyperparameter optimization was conducted using GridSearchCV on the training data. The tuning process was performed independently of the testing data to prevent information leakage and ensure that the final evaluation reflected the model's ability to generalize to unseen reviews. The optimization process identified α=0.1 as the best smoothing parameter, as shown in Table 11..

Table 11. Best Naïve Bayes Configurations

Parameters	Value
Alpha	0.1

The parameter α controls the degree of additive smoothing applied by the Naïve Bayes model. Smoothing is important in text classification because some terms may not occur in particular sentiment classes within the training data. Without smoothing, unseen feature-class combinations could be assigned zero probability, negatively affecting classification. The selected value of 0.1 provides relatively light smoothing, allowing the model to preserve the influence of observed term distributions while still accounting for rarely occurring features. After hyperparameter selection, the optimized classifier was trained using the complete training dataset and evaluated against the independent testing subset. Figure 4 presents the resulting confusion matrix.

The confusion matrix illustrates the distribution of correct and incorrect predictions across positive and negative sentiment classes. It provides additional information beyond

aggregate metrics by showing which sentiment category contributes more strongly to the model's classification errors. The performance results of the Naïve Bayes classifier are presented in Table 12.

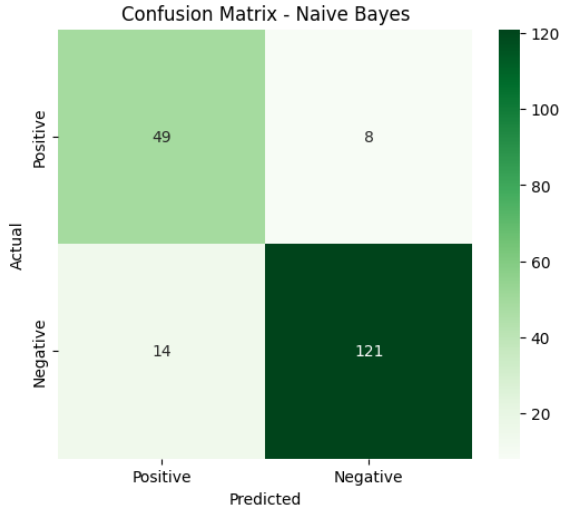

Figure 4. Matrix of Confusion – Naïve Bayes

Table 12. Results of Bayes' Naïve Classification

Metrics	Value (%)
Accuracy	88,54
Precision (Weighted)	89,04
Recall (Weighted)	88,54
F1-Score (Weighted)	88,70

The Naïve Bayes classifier achieved an accuracy of 88.54%, with a weighted precision of 89.04%, weighted recall of 88.54%, and weighted F1-score of 88.70%. These results indicate that the probabilistic classifier was capable of identifying the general sentiment patterns contained in the TF-IDF representation, although its performance remained below that obtained using KNN. The weighted precision value of 89.04% indicates that the predicted sentiment labels were generally reliable, whereas the weighted recall of 88.54% reflects the proportion of actual class instances that were correctly identified after accounting for class distribution. The weighted F1-score of 88.70% further demonstrates a reasonably balanced relationship between precision and recall.

Although Naïve Bayes performs effectively for many text-classification problems, its underlying conditional-independence assumption may limit its ability to represent more complex relationships among textual features. In user reviews, sentiment is frequently expressed through combinations of words and contextual expressions rather than completely independent terms. This characteristic may partially explain the difference between Naïve Bayes and the similarity-based KNN approach observed in the present experiment.

3.6 Comparative Performance

To determine which classification approach was more effective for the Profits Anywhere review dataset, the performance of the optimized KNN and Naïve Bayes classifiers was compared using the same independent testing set. Accuracy, weighted precision, weighted recall, and weighted F1-score were employed as the primary evaluation measures. The comparison is presented in Table 13.

Table 13. Model Performance Comparison

Metrics	KNN (%)	Naïve Bayes (%)
Accuracy	**92,19**	**88,54**
Precision	92,23	89,04
Recall	92,19	88,54
F1-Score	92,21	88,70

The comparative results demonstrate that KNN consistently outperformed Naïve Bayes across all four evaluation metrics. KNN achieved an accuracy of 92.19%, compared with 88.54% for Naïve Bayes, corresponding to an absolute improvement of 3.65 percentage points. A similar pattern was observed for the remaining metrics. Weighted precision increased from 89.04% to 92.23%, representing an improvement of 3.19 percentage points, while weighted recall improved by 3.65 percentage points. The weighted F1-score increased from 88.70% for Naïve Bayes to 92.21% for KNN, representing an improvement of 3.51 percentage points.

The superior performance of KNN suggests that similarity among review documents provides particularly useful information for sentiment classification in this dataset. With cosine distance, the classifier evaluates the similarity between the direction of TF-IDF

vectors rather than relying solely on their magnitude. Reviews containing similar combinations of sentiment-bearing terms can therefore be grouped within the feature space, even when the reviews differ in length. The distance-weighting mechanism may further contribute to this performance because highly similar neighboring documents exert a greater influence on the predicted sentiment label. As a result, the classifier can exploit local structures in the TF-IDF feature space that may correspond to recurring patterns of user opinion.

In contrast, the Naïve Bayes classifier estimates sentiment probabilities based on the statistical distribution of textual features and generally assumes conditional independence among them. While this assumption enables efficient classification, it may oversimplify relationships among terms in user reviews. Sentiment expressions often depend on combinations of words, including modifiers and negations, which may not be fully represented under a strong independence assumption.

Another important finding is the relatively small difference among the KNN accuracy, weighted precision, weighted recall, and weighted F1-score values. This consistency indicates that the model's performance is not driven exclusively by one evaluation criterion. The 92.21% weighted F1-score is particularly relevant given the moderately imbalanced distribution of positive and negative reviews because it simultaneously considers precision and recall while weighting each class according to its support. Overall, the experimental results indicate that the optimized KNN classifier using seven neighbors, cosine distance, and distance-based weighting provides the strongest performance among the two evaluated approaches. Within the conditions of this experiment, KNN therefore represents the more effective classifier for sentiment analysis of Profits Anywhere application reviews using TF-IDF features.

However, the observed superiority should be interpreted specifically within the evaluated dataset and experimental configuration rather than as evidence that KNN universally outperforms Naïve Bayes. Model effectiveness may vary according to dataset size, vocabulary characteristics, sentiment distribution, preprocessing choices, and feature representation. Additional evaluation using other classifiers and statistical significance testing could further strengthen the comparative conclusions.

3.7 Discussion

The experimental results demonstrate that the K-Nearest Neighbor (KNN) classifier provided better sentiment-classification performance than Naïve Bayes for the Profits Anywhere review dataset under the experimental configuration adopted in this study. KNN achieved an accuracy of 92.19% and a weighted F1-score of 92.21%, whereas Naïve Bayes obtained an accuracy of 88.54% and a weighted F1-score of 88.70%. Thus, KNN produced an improvement of 3.65 percentage points in accuracy and 3.51 percentage points in weighted F1-score. Similar improvements were also observed for weighted precision and weighted recall, indicating that the advantage of KNN was consistent across the principal evaluation metrics rather than being limited to a single measure.

One possible explanation for this result is the compatibility between the TF-IDF feature representation and the similarity-based learning mechanism of KNN. Each review was represented as a high-dimensional TF-IDF vector containing unigram and bigram information. With cosine distance, KNN determines similarity primarily from the orientation of document vectors rather than their absolute magnitude. This characteristic is advantageous for textual data because two reviews may express similar opinions even when their lengths and term frequencies differ substantially. In addition, the use of distance-based weighting enables observations located closer to the query review to exert greater influence on the predicted class. Consequently, reviews containing similar combinations of sentiment-bearing words and phrases can contribute more strongly to the final classification decision.

The selected value of k=7 may also have contributed to the performance of the KNN classifier. A neighborhood consisting of several observations can reduce sensitivity to isolated or noisy training samples while maintaining sufficient locality to capture recurring sentiment patterns. In this dataset, this configuration appears to provide an appropriate balance between local discrimination and prediction stability. Nevertheless, this interpretation is specific to the present dataset because the optimal number of neighbors may change with dataset size, vocabulary characteristics, and class distribution.

The performance of KNN may additionally be related to the use of unigram and bigram TF-IDF features. Sentiment in application reviews is not always expressed through individual words in isolation. Short expressions containing modifiers or negations can

alter the meaning of otherwise positive or negative terms. For example, the sentiment represented by "good" differs substantially from that represented by "not good." Because the preprocessing procedure retained negation terms and the TF-IDF representation incorporated bigrams, some of these local contextual patterns could be preserved in the feature space. Reviews containing similar lexical structures may therefore become closer under cosine similarity, supporting the neighborhood-based decision mechanism used by KNN.

In contrast, the Naïve Bayes classifier relies on a probabilistic representation of feature distributions and is generally based on a conditional-independence assumption among features given the class. Although this assumption makes the algorithm computationally efficient and suitable for high-dimensional text data, natural-language expressions frequently contain dependencies among words. The meaning of a term may be influenced by neighboring words, modifiers, or negations. Such relationships may not be fully represented when textual features are modeled as conditionally independent. This characteristic may partly explain why Naïve Bayes produced lower performance than KNN in the present experiment.

Nevertheless, the Naïve Bayes results should not be regarded as poor. An accuracy of 88.54% and a weighted F1-score of 88.70% indicate that the model was still capable of identifying substantial sentiment-related patterns within the TF-IDF representation. Its relatively strong performance also confirms that probabilistic text classification remains a viable baseline for this dataset. The comparison therefore suggests not that Naïve Bayes is unsuitable for sentiment analysis, but rather that the similarity structure captured by optimized KNN was more beneficial for the characteristics of the Profits Anywhere reviews evaluated in this study.

Another relevant aspect concerns the distribution of sentiment classes. The final dataset contained 674 positive reviews (70.43%) and 283 negative reviews (29.57%), indicating a moderate class imbalance. Stratified sampling was used to maintain approximately the same class proportions in the training and testing subsets, thereby reducing the risk of an unrepresentative data partition. However, the reported precision, recall, and F1-score values are weighted averages, meaning that the contribution of each class is proportional to its number of observations. Consequently, the majority positive class has a greater

influence on the aggregate scores. For this reason, the weighted F1-score of 92.21% indicates strong overall performance but does not, by itself, demonstrate that positive and negative reviews were classified equally well. The confusion matrices provide additional class-level information, while future evaluations should also report macro-averaged and class-specific precision, recall, and F1-scores, particularly for the minority negative class.

The findings also highlight the importance of hyperparameter optimization in sentiment-classification experiments. The best KNN configuration—seven neighbors, cosine distance, and distance weighting—was identified using GridSearchCV with five-fold stratified cross-validation on the training data. Similarly, the Naïve Bayes smoothing parameter was optimized before final testing. This procedure reduces dependence on arbitrarily selected parameter values and provides a more systematic basis for comparing the two algorithms. Importantly, the independent testing data were not used during hyperparameter selection, thereby helping to reduce information leakage between model development and final performance evaluation.

Although KNN achieved the best performance in this study, the findings should be interpreted within the boundaries of the experimental design. The observed superiority of KNN does not imply that it will consistently outperform Naïve Bayes for all sentiment-analysis datasets. Classification performance depends on several factors, including corpus size, vocabulary characteristics, language variation, feature representation, class balance, labeling quality, preprocessing decisions, and hyperparameter settings. In a larger or substantially different corpus, the relative performance of the two methods may change. Therefore, the results demonstrate the suitability of KNN specifically for the Profits Anywhere review dataset and the TF-IDF configuration examined in this study, rather than establishing a universally superior classifier.

This research has several limitations. First, sentiment labeling is done based on Google Play Store star ratings so that it does not involve independent human annotations as a reference standard. Second, the study used only one dataset derived from the Profits Anywhere application, so generalization of results to other application domains needs to be done carefully. Third, the dataset had a moderate class imbalance with a predominance of positive reviews. Fourth, model evaluation was carried out using one test-study data

sharing (80:20) so that the stability of the model in various data sharing scenarios has not been evaluated in depth. Therefore, further research may consider the use of independent human annotations, repeated cross-validation, external datasets, alternative feature representations, and more robust classification models to improve the validity and generalization of the research results.

4. CONCLUSION

This study compares the performance of the K-Nearest Neighbor (KNN) and Naïve Bayes algorithms in the sentiment classification of user reviews of the Profits Anywhere application using the TF-IDF feature representation. Based on the dataset of 957 reviews and evaluation protocols used, KNN achieved better classification performance than Naïve Bayes with an accuracy of 92.19%, weighted precision of 92.23%, weighted recall of 92.19%, and a weighted F1-score of 92.21%. The results show that the TF-IDF preprocessing pipeline and representation provide a structured feature space for sentiment classification in application reviews. These findings provide empirical evidence regarding the selection of sentiment classification algorithms in the domain of digital business and fintech applications. However, the results of the study need to be interpreted within the limitations of the use of Google Play Store rating-based labeling, the use of one application dataset, moderate class imbalances, and evaluations carried out using one data sharing of test-based data. Further research can develop this study by expanding the number of datasets, adding variations of text features, and comparing other classification algorithms to obtain more optimal performance.

ACKNOWLEDGMENT

The author would like to express his sincere gratitude to the Kemdiktisaintek for the support and facilitation provided for this research.

REFERENCES

[1] M. Jorayeva, A. Akbulut, C. Catal, And A. Mishra, "Machine Learning-Based Software Defect Prediction For Mobile Applications: A Systematic Literature Review," *Sensors*, Vol. 22, No. 7, Pp. 2–17, 2022, Doi: 10.3390/S22072551.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

[2] A. A. Qureshi, M. Ahmad, S. Ullah, M. N. Yasir, F. Rustam, And I. Ashraf, "Performance Evaluation Of Machine Learning Models On Large Dataset Of Android Applications Reviews," *Multimed. Tools Appl.*, Vol. 82, No. 24, Pp. 37197–37219, 2023, Doi: 10.1007/S11042-023-14713-6.

[3] O. Aljeezani, D. Alomari, And I. Ahmad, "Arabic App Reviews: Analysis And Classification," *Acm Trans. Asian Low-Resour. Lang. Inf. Process.*, Vol. 24, No. 2, Feb. 2025, Doi: 10.1145/3708987.

[4] S. A. Memon, M. A. Mahar, A. Kehar, S. Oad, And I. Ahmed, "User Experience Enhancement Through Sentiment Analysis: A Machine Learning Approach To App Reviews," *Int. J. Inf. Syst. Comput. Technol.*, Vol. 4, No. 2, Pp. 37–46, 2025, Doi: 10.58325/Ijisct.004.02.0131 User.

[5] A. Arora, V. Soni, M. Sharma, P. Kulhari, And P. Vats, "The Role Of Digital Marketing Analytics In Enhancing Passive Income From E-Commerce Platforms," In *In: Atulya K., N., Dharm Singh, J., Durgesh Kumar, M., Joshi, A. (Eds) Intelligent Sustainable Systems. Worlds4 2025. Lecture Notes In Networks And Systems*, 2026, Pp. 1–11. Doi: 10.1007/978-3-032-11521-8_1.

[6] L. R. Ali, B. N. Shaker, And S. A. Jebur, "An Extensive Study Of Sentiment Analysis Techniques: A Survey," In *Aip Conference Proceedings*, 2023, Vol. 2591, P. 2591. Doi: 10.1063/5.0119604.

[7] Y. Yuniarthe, A. Syarif, I. M. Shofi, And Fatimah Fahurian, "Sentiment Analysis Of Twitter Discussions About Lampung Robusta Coffee: A Comparative Study Of Machine Learning Algorithms With Svm As The Optimal Model," *J. Tek. Inform.*, Vol. 18, No. 2, Pp. 339–349, 2025, [Online]. Available: Https://Journal.Uinjkt.Ac.Id/Index.Php/Ti/Article/View/41316

[8] P. Rindiani *Et Al.*, "Penerapan Machine Learning Untuk Analisis Sentimen Agoda Dengan Algoritma Knn , Naive Bayes Dan Svm," *Jumistik J. Manaj. Inform. Sist. Inf. Dan Teknol. Inform.*, Vol. 4, No. 2, Pp. 664–672, 2025, Doi: 10.70247/Jumistik.V4i2.232.

[9] A. A. Purnama And Y. R. Sipayung, "Sentiment Analysis Of Public Service Using Naïve Bayes Classifier," *J. Inf. Syst. Informatics*, Vol. 7, No. 3, Pp. 2439–2457, 2025, Doi: 10.51519/Journalisi.V7i3.1207.

[10] A. Nurfaida, R. Y. A. Syabana, T. Abimanyu, And N. Husufa, "Web-Based Inventory Application And Prediction Using Naive Bayes Algorithm (Case Study: Puskesmas Kecamatan Sawah Besar)," *J. Inf. Syst. Informatics*, Vol. 4, No. 4, Pp. 864–878, 2022, Doi: 10.51519/Journalisi.V4i4.348.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

[11] A. K. Laturiuw And Y. A. Singgalen, "Sentiment Analysis Of Raja Ampat Tourism Destination Using Crisp-Dm: Svm, Nbc, Dt, And K-Nn Algorithm," *J. Inf. Syst. Informatics*, Vol. 5, No. 2, Pp. 518–535, 2023, Doi: 10.51519/Journalisi.V5i2.490.

[12] M. Febima And L. Magdalena, "Predictive Analytics On Shopee For Optimizing Product Demand Prediction Through K-Means Clustering And Knn Algorithm Fusion," *J. Inf. Syst. Informatics*, Vol. 6, No. 2, Pp. 751–765, 2024, Doi: 10.51519/Journalisi.V6i2.720.

[13] A. K. Iman And E. I. H. Ujianto, "Analisis Sentimen Pemindahan Ibu Kota Indonesia Menggunakan K-Nearest Neighbor Sentiment Analysis Of The Relocation Of Indonesia ' S Capital Using K-Nearest Neighbor," *J. Pendidik. Dan Teknol. Indones.*, Vol. 4, No. 12, Pp. 759–768, 2024, Doi: 10.52436/1.Jpti.546.

[14] T. M. Fadli And A. S. Puspaningrum, "Analisis Sentimen Terhadap Kenaikan Gaji Guru Honorer Menggunakan Algoritma Naïve Bayes Sentiment Analysis Towards The Increase In Honorary Teachers ' Salary Using The Naïve Bayes Algorithm," *J. Pendidik. Dan Teknol. Indones.*, Vol. 5, No. 3, Pp. 635–644, 2025, Doi: 10.52436/1.Jpti.689.

[15] A. Bhalla, "Comparative Analysis Of Text Classification Using Svm, Naive Bayes, And Knn Models," In *2023 5th International Conference On Inventive Research In Computing Applications (Icirca)*, 2023, Pp. 878–883. Doi: 10.1109/Icirca57980.2023.10220728.

[16] P. C. E. And S. Mata-Domingo, "Sentiment Analysis On Product Reviews: A Machine Learning Approach For E- Commerce Platforms," *Int. J. Manag. Entrep. Res.*, Vol. 8, No. 1, Pp. 148–161, 2026, Doi: 10.51594/Ijmer.V8i1.2184.

[17] N. L. W. S. R. Ginantra, C. P. Yanti, G. D. Prasetya, I. B. G. Sarasvananda, And I. K. A. G. Wiguna, "Analisis Sentimen Ulasan Villa Di Ubud Menggunakan Metode Naive Bayes, Decision Tree, Dan K-Nn," *Janapati (Jurnal Nas. Pendidik. Tek. Inform.*, Vol. 11, No. 3, Pp. 205–216, 2024, Doi: 10.23887/Janapati.V11i3.49450.

[18] S. Thiang, W. Chandra, Ferawaty, And M. A. S. Manulang, "Comparison Of Naïve Bayes Classifier And K-Nearest Neighbor Algorithms In Sentiment Analysis On Social Media X With Vader Lexicon," *Jiko (Jurnal Inform. Dan Komputer)*, Vol. 8, No. 2, Pp. 85–93, 2025, Doi: 10.33387/Jiko.V8i2.9865.

[19] V. And P. S. Atina, "Sentiment Analysis Of Grab App Reviews With Machine Learning Approach," In *Proceeding Of International Conference On Science, Health, And Technology*, 2024, Pp. 395–402. Doi: 10.47701/Icohetech.V5i1.4218.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

[20] S. Putri, A. Ati, P. Muharani, And R. Qori, "Sentiment Analysis Of Gojek User Reviews Using Tf-Idf And Machine Learning In Orange Platform," *Jsce (Journal Syst. Comput. Eng.*, Vol. 6, No. 4, Pp. 296–305, 2025, Doi: 10.61628/Jsce.V6i4.2062.