

Sentiment Analysis of Generative AI Application Reviews Using Random Forest and SVM with SMOTE-Based Class Imbalance Handling

Istiqomah¹, Alamsyah^{2*}

^{1,2}Computer Science Department, Faculty of Mathematics and Natural Sciences,
Universitas Negeri Semarang, Indonesia

Abstract. The rapid adoption of Generative Artificial Intelligence (Gen-AI) has generated extensive user reviews, offering valuable insights into user perceptions. This study aims to classify review sentiment using Random Forest and Support Vector Machine (SVM), evaluate the impact of SMOTE on class imbalance, and analyze review themes. This study analyzed 50,000 reviews from five popular Gen-AI applications: ChatGPT, Microsoft Copilot, Google Gemini, Perplexity, and Claude. The models were evaluated using accuracy, weighted precision, weighted recall, weighted F1-score, and confusion matrix. The results showed that SVM without SMOTE achieved the highest overall accuracy of 84% and a weighted F1-score of 0.81. However, its F1-score for the neutral class was only 0.06. Applying SMOTE reduced the overall accuracy of Random Forest from 83% to 81% and SVM from 84% to 76%. Nevertheless, SMOTE increased the neutral-class F1-score from 0.00 to 0.06 for Random Forest and from 0.06 to 0.20 for SVM. These results indicate that SMOTE improved minority-class recognition, particularly for SVM, but introduced a trade-off with overall predictive performance. Ultimately, these findings provide practical insights for improving Gen-AI applications while also serving as a reference for selecting sentiment classification models in imbalanced multiclass sentiment analysis.

Purpose: This study aims to classify the sentiment of user reviews on Gen-AI applications using Random Forest and Support Vector Machine (SVM), evaluate the impact of SMOTE on imbalanced multiclass data, and identify key review themes influencing user perceptions.

Methods/Study design/approach: A dataset of 50,000 user reviews from five popular Gen-AI applications was analyzed using text preprocessing, TF-IDF feature extraction, rating-based sentiment labeling, and SMOTE applied to the training data. Random Forest and SVM were evaluated using accuracy, weighted precision, weighted recall, weighted F1-score, and confusion matrix.

Result/Findings: SVM without SMOTE achieved the best overall performance, with an accuracy of 84% and a weighted F1-score of 0.81. Although SMOTE reduced overall accuracy, it improved the neutral-class F1-score from 0.00 to 0.06 for Random Forest and from 0.06 to 0.20 for SVM, demonstrating better recognition of minority-class samples.

Novelty/Originality/Value: This study provides a comparative evaluation of Random Forest and SVM with and without SMOTE for multiclass sentiment analysis of Gen-AI user reviews. It also integrates review theme analysis to identify factors affecting user perceptions, offering practical guidance for selecting classification models and improving Gen-AI applications.

Keywords: Generative AI, Random Forest, Sentiment Analysis, SMOTE, Support Vector Machine

Received July 2026 / **Revised** August 2026 / **Accepted** September 2026

This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).



INTRODUCTION

Artificial Intelligence (AI) has advanced rapidly in recent years. As one of the key technologies driving digitalization and digital transformation, AI is expected to play an increasingly important role in the future [1]. These advancements have enabled AI systems to automate a growing range of tasks that were previously performed exclusively by humans [2]. One of the most significant developments in AI is the emergence of Generative Artificial Intelligence (Gen-AI), which has transformed the way people work, learn, and interact with technology [3]. Unlike conventional AI systems that primarily focus on analysis or prediction, Gen-AI is capable of generating adaptive and goal-oriented outputs across various content formats, including text, images, audio, video, and source code [4]. Furthermore, Gen-AI can generate new content that closely resembles human creativity by learning patterns from large-scale datasets [5][6]. This

*Corresponding author

Email addresses: alamsyah@mail.unnes.ac.id (Alamsyah)

DOI: 10.15294/rji.v4i2.60530

capability has made Gen-AI highly applicable across a wide range of domains, including Natural Language Processing (NLP), computer vision, education, healthcare, finance, business, and the creative industries [7]. Consequently, Gen-AI has become a transformative technology that continues to drive innovation across multiple sectors. The rapid advancement of Gen-AI has also led to the emergence of numerous applications that enable users to interact with intelligent, context-aware, and intuitive natural language-based systems. Applications such as ChatGPT, Microsoft Copilot, Google Gemini, Perplexity, and Claude are widely used Gen-AI tools for academic, professional, and everyday activities.

As the popularity of Gen-AI applications continues to grow, the volume of user reviews on digital platforms such as Google Play Store has increased substantially. These reviews contain diverse opinions, user experiences, levels of satisfaction, and complaints regarding service quality and application usability. Collectively, they provide valuable insights into users' emotions and perceptions, offering a reliable basis for assessing public sentiment toward Gen-AI applications [8][9][10]. User reviews serve as an important source of information for developers to better understand user needs, expectations, and feedback for evaluation purposes. The opinions expressed in textual reviews also provide valuable guidance for continuously improving and refining the quality of Gen-AI applications and their services [11][12]. The volume of user reviews on Gen-AI applications continues to increase as their adoption becomes more widespread. The growing amount of review data makes manual analysis increasingly impractical. Such an approach is time-consuming, difficult to perform comprehensively, and susceptible to subjective interpretation [13]. These limitations highlight the need for a computational approach capable of automatically processing and analyzing large volumes of textual data. Natural Language Processing (NLP) is one of the key technologies for understanding human language and processing large-scale textual data [14][15]. Sentiment analysis, a prominent NLP technique, is widely used to identify and classify the polarity of textual data into positive, negative, or neutral sentiment classes. This technique enables the extraction of subjective information from text by considering the context and underlying meaning of user reviews [16]. Numerous studies have employed machine learning approaches for sentiment analysis because of their ability to achieve strong classification performance on textual review data. Various algorithms have also been proposed and compared to identify classification models with optimal predictive performance.

A study by Kurniawan et al. (2025) successfully captured user perceptions using the Naïve Bayes algorithm, providing actionable insights for evaluating and improving the service quality of the GetContact application [17]. Similarly, Nurhasanah et al. (2026) conducted sentiment analysis on Gen-AI application reviews using a combination of TF-IDF and the Naïve Bayes algorithm, achieving an average accuracy of 82.31%. However, the model struggled to classify neutral sentiments due to class imbalance [18]. Imanuddin et al. (2023) conducted a sentiment analysis of Satushat application reviews using Support Vector Machine (SVM). The algorithm achieved an accuracy of 91%, demonstrating its effectiveness in classifying user sentiments [19]. Using Shopee user reviews, Rahman et al. (2024) applied the Random Forest algorithm with TF-IDF feature extraction, with performance metrics of 94% accuracy, 91% precision, 91% recall, and 93% F1-score, demonstrating strong classification performance [20]. Pratama et al. (2025) compared the performance of Random Forest, SVM, and Naïve Bayes for sentiment analysis of Apple Vision Pro user reviews. Random Forest achieved the highest accuracy of 83%, followed by SVM with 80% and Naïve Bayes with 75%. Based on these findings, the authors recommended the adoption of ensemble learning and data balancing techniques to further improve classification performance [21]. In contrast, Shobil et al. (2026) evaluated user reviews of the Solar Energy Monitoring App and reported that Random Forest achieved an accuracy of only 68%, whereas SVM outperformed it with an accuracy of 91% [22]. Meanwhile, Kartika et al. (2026) incorporated TF-IDF, Synthetic Minority Over-sampling Technique (SMOTE), and N-Gram into sentiment analysis of M-Passport application reviews using the Naïve Bayes algorithm. The results showed that the integration of SMOTE and N-Gram improved the model's ability to classify the minority class and increased the accuracy from 61% to 77.51% [23]. Similarly, Putra (2025) combined SMOTE with a stacking ensemble method to address class imbalance in sentiment analysis of Traveloka hotel reviews. The findings demonstrated that SMOTE effectively balanced the sentiment class distribution, particularly for the minority class, resulting in more stable predictions and an improved model accuracy of 88% [24].

Previous studies have shown that Naïve Bayes, Random Forest, and SVM are effective for sentiment analysis and achieve strong classification performance across various datasets. In addition, TF-IDF and SMOTE have been shown to improve classification performance [25]. However, the effectiveness of these algorithms may vary depending on the characteristics of the dataset. Although sentiment analysis has been

successfully applied across various application domains, previous studies have mainly concentrated on user reviews from public service and e-commerce applications. In contrast, studies investigating user reviews of multiple Gen-AI chatbot applications remain limited, despite the rapid growth in Gen-AI adoption and the increasing volume of user reviews that can be utilized to evaluate service quality. Most existing studies have also focused primarily on comparing classification algorithms without addressing the issue of class imbalance. Furthermore, studies that specifically compare the performance of Random Forest and SVM with the application of SMOTE on user reviews from multiple Gen-AI applications remain scarce. Accordingly, the objective is to classify user reviews from various Gen-AI applications into three sentiment categories while addressing class imbalance.

The novelty of this study lies in its comparative sentiment analysis by evaluating the performance of Random Forest and SVM on user reviews of multiple Gen-AI applications while addressing potential class imbalance through the application of SMOTE prior to the classification process. In addition, this study presents a descriptive analysis of sentiment distribution across applications and review themes to identify the aspects that receive the most user complaints and appreciation. The study uses user reviews of Gen-AI applications that reflect users' perceptions of Gen-AI technology across widely used applications, including ChatGPT, Gemini, Claude, Copilot, and Perplexity. The dataset was obtained from the public Kaggle dataset entitled "The Generative AI Ecosystem: 50K User Reviews 2026", which contains 50,000 user reviews. The data were processed through text preprocessing, rating-based labeling, feature extraction, class imbalance handling using SMOTE, and sentiment classification using Random Forest and SVM. The findings are expected to provide insights into user perceptions of Gen-AI technology and serve as a reference for selecting effective sentiment classification methods within this domain.

METHODS

This study was conducted through a series of systematic stages to produce accurate sentiment analysis results. The process begins with collecting a dataset of Gen-AI application user reviews from Kaggle. The collected data then underwent data selection and cleaning, text preprocessing, sentiment labeling, train-test splitting, and TF-IDF feature extraction, which was performed exclusively on the training data. Two experimental scenarios were conducted: training data with SMOTE for class imbalance handling and training data without SMOTE. In each scenario, sentiment classification was performed using Random Forest and SVM, followed by performance evaluation using accuracy, weighted precision, weighted recall, weighted F1-score, and confusion matrix. Figure 1 presents the overall research workflow.

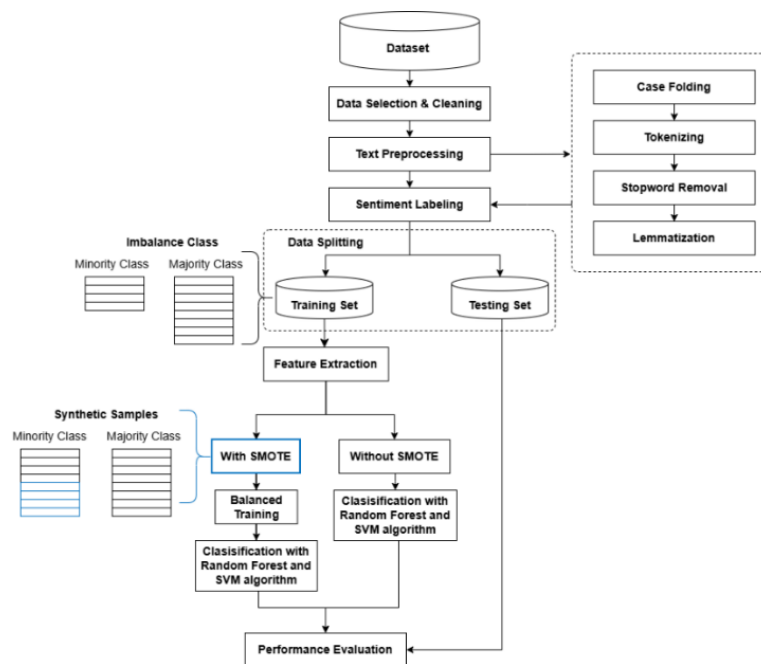


Figure 1. Research Flow

All research procedures were implemented in Python using the Google Colaboratory platform. Several libraries were utilized, including pandas for data manipulation, NLTK (Natural Language Toolkit) for text

preprocessing, and scikit-learn for TF-IDF feature extraction, sentiment classification, and SMOTE implementation. In addition, Matplotlib and Seaborn were used for data visualization, including heatmap generation for the descriptive analysis.

Dataset

This study uses a publicly available dataset obtained from Kaggle entitled "*The Generative AI Ecosystem: 50K User Reviews 2026*" [26]. A total of 50,000 user reviews from five Gen-AI applications: ChatGPT, Microsoft Copilot, Google Gemini, Perplexity, and Claude, were collected from the U.S. Google Play Store in early 2026. The dataset includes several attributes, namely App, Review_Date, Star_Rating, Review_Text, Word_Count, Review_Length_Chars, Thumbs_Up_Count, App_Version, Sentiment_Polarity, and Review_Theme. In this study, only Review_Text and Star_Rating were used. Review_Text serves as the textual data for sentiment analysis, Star_Rating is used for sentiment labeling, while App and Review_Theme attributes are used in the descriptive analysis to examine the distribution of sentiments across Gen-AI applications and review themes. This dataset was selected because it contains a relatively large number of user reviews with diverse opinions, making it suitable for evaluating the classification performance of Random Forest and SVM, particularly before and after addressing class imbalance using SMOTE. The dataset has also been utilized in previous research on sentiment analysis of Gen-AI user reviews using a Python-based Naïve Bayes algorithm [18].

Data Selection & Cleaning

Ensuring high-quality data through data cleaning is a fundamental part of the Natural Language Processing (NLP) workflow, as it directly affects model performance and accuracy of NLP models [11]. The dataset obtained from Kaggle first underwent attribute selection, where only the Review_Text and Star_Rating attributes were used for model training and sentiment classification. A duplicate check was then performed, confirming that the dataset contained no duplicate records. Subsequently, the Review_Text attribute was cleaned by removing unnecessary characters, including punctuation, mentions, emojis, URLs, hashtags, and excessive whitespace. Numerical characters were also removed to reduce irrelevant token variations that could affect sentiment classification. This process aims to produce cleaner and more consistent textual data for the subsequent analysis stages [17][27].

Text Preprocessing

Text preprocessing is a fundamental stage in Natural Language Processing (NLP) that aims to produce cleaner, more structured textual data for effective machine learning model development [28]. In this study, text preprocessing was performed sequentially through four main stages: case folding, tokenization, stopwords removal, and lemmatization. In the case folding stage, uppercase letters were transformed into lowercase to maintain text consistency and standardize word representation. Next, tokenization was performed by splitting each review into individual words or tokens, allowing each word to be represented independently. The resulting tokens then underwent stopwords removal to eliminate common words that contributed little to the sentiment classification process [29]. The NLTK stopwords list was customized by retaining negation words, including *not*, *no*, *nor*, *never*, and *neither*, to preserve sentiment polarity. The final stage was lemmatization, which transforms inflected words into their base forms (lemmas) based on their vocabulary entries and grammatical categories, thereby reducing lexical variations. Lemmatization was selected because it normalizes words according to linguistic rules while preserving their semantic meaning more effectively than other normalization techniques [30]. To further improve the lemmatization process, this study incorporated Part-of-Speech (POS) tagging to identify grammatical categories, such as nouns, verbs, adjectives, and adverbs, enabling more accurate determination of each word's base form.

Sentiment Labeling

Sentiment labeling assigns sentiment categories to data, allowing machine learning models to classify sentiment effectively [31]. Sentiment labels were assigned using a rating-based labeling approach based on the Star_Rating attribute provided in the dataset as a practical proxy. This approach was adopted because user ratings are strongly correlated with the sentiment expressed in textual reviews [32]. Each review was categorized into one of three sentiment classes according to its rating: negative, neutral, or positive. Based on the star rating, reviews were classified into three sentiment categories: ratings of 1–2 as negative, 3 as neutral, and 4–5 as positive. This labeling scheme follows the rating-based labeling approach that has been widely adopted as the target labeling strategy in sentiment classification studies [33].

Data Splitting

Data splitting is the process of dividing a dataset into a training set and a testing set prior to model development [34]. The dataset was partitioned using a stratified train-test split method (stratified random sampling) with an 80:20 ratio to preserve the proportion of each sentiment class in the training and testing sets while maintaining class proportions consistent with the original dataset. The 80:20 split was selected because it provides a balanced allocation of data for model training and performance evaluation [35]. A random_state value of 42 was used to ensure reproducibility of the data splitting process. Additionally, stratify=y was used to preserve sentiment class distribution across the training and testing sets.

Feature Extraction

Feature extraction converts textual data into numerical features that can be interpreted by machine learning algorithms [36]. In this study, the Term Frequency–Inverse Document Frequency (TF-IDF) method was employed to generate feature vectors by assigning weights to words according to their frequency of occurrence and their relative importance within the corpus. TF-IDF was selected because it has consistently demonstrated strong performance in machine learning-based sentiment analysis studies [37]. Term Frequency (TF) measures the frequency of each word in a document, whereas Inverse Document Frequency (IDF) emphasizes informative words by decreasing the weights of terms that are common across multiple documents. Consequently, more representative terms are assigned higher weights during the feature extraction process. The TF value is calculated using Equation (1).

$$TF(i, j) = \frac{n(i, j)}{\sum_k n(k, j)} \quad (1)$$

The IDF weighting is defined by Equation (2).

$$IDF(i) = \log \left(\frac{1 + N}{1 + df(i)} \right) + 1 \quad (2)$$

In Equation (1), $n(i, j)$ indicates the number of occurrences of term i in document j , while the denominator represents the total number of terms in document j . In Equation (2), N refers to the total number of documents in the corpus, and $df(i)$ represents the number of documents containing term i . Based on Equations (1) and (2), the TF-IDF weight is computed using Equation (3).

$$TFIDF(i, j) = TF(i, j) \times IDF(i) \quad (3)$$

In this study, TF-IDF features were generated using the TfidfVectorizer class provided by the scikit-learn library. The parameters were configured as min_df = 5 to reduce noise, max_df = 0.95 to exclude terms appearing in more than 95% of the documents, and ngram_range = (1,1) to extract unigram features. All remaining parameters were set to their default values. The vocabulary and IDF values were learned exclusively from the training data using the fit_transform() method, while the transform() method was applied to the testing data. This approach prevents information from the testing set from influencing the model training process, thereby avoiding data leakage.

SMOTE Data Balancing

Synthetic Minority Over-sampling Technique (SMOTE) is one of the most widely used oversampling methods for addressing class imbalance. Unlike conventional oversampling techniques that simply duplicate minority class instances, SMOTE generates new synthetic samples through interpolation based on the K-Nearest Neighbors (KNN) algorithm [38]. The method selects neighboring samples from the minority class and generates synthetic instances along the line segments connecting them. As a result, the generated samples preserve the characteristics of the minority class while introducing greater diversity, thereby reducing the risk of overfitting [39]. The general formulation of SMOTE is presented in Equation (4).

$$x_{new} = x_i + \lambda(x_j - x_i) \quad (4)$$

where x_{new} is defined as the generated synthetic sample, x_i represents a minority class sample, and x_j denotes one of the k-nearest neighbors of x_i within the minority class. The parameter λ is a random value in the range of [0,1] that determines the interpolation position between x_i and x_j . Two experimental

scenarios were designed to evaluate the effect of SMOTE on the classification performance. The first scenario used the original training data without class balancing, whereas the second scenario applied SMOTE to the training data to obtain a more balanced class distribution. In the second scenario, oversampling was performed only on the training data after TF-IDF feature extraction, while the testing data remained unchanged to preserve its original class distribution. SMOTE was implemented using the imbalanced-learn library with `sampling_strategy='auto'` to balance the class distribution, `k_neighbors=5` to generate synthetic samples based on the five nearest neighbors, and `random_state=42` to ensure reproducibility.

Classification

The classification stage involved training models to categorize Gen-AI user reviews into three sentiment classes by learning patterns from the training data. Two supervised machine learning algorithms, Random Forest and SVM, were implemented with the Python programming language and the scikit-learn library.

Random Forest

Random Forest is a widely used machine learning algorithm for classification and prediction, recognized for its robust performance [40]. The algorithm constructs an ensemble of Decision Trees using different subsets of the training data, enabling it to produce more accurate and robust predictions than a single Decision Tree [41]. Random Forest increases the diversity among Decision Trees through bootstrap sampling and random feature selection [20]. Bootstrap sampling randomly selects samples from the TF-IDF feature matrix with replacement to generate a different training subset for each Decision Tree. As a result, a sample may be selected multiple times, while others may not be included in a particular bootstrap sample [42]. During the construction of each Decision Tree, Random Forest randomly selects a subset of features at each node as candidates for splitting. The optimal feature is then determined using the Gini Index criterion. The Gini Index measures the impurity of a node by estimating the probability of incorrectly classifying a randomly selected sample. A lower Gini Index indicates a better split [43]. The Gini Index is calculated using Equation (5), where C denotes the number of classes and p_i represents the proportion of samples belonging to the (i) -th class.

$$Gini(D) = 1 - \sum_{i=1}^C (p_i)^2 \quad (5)$$

The predictions generated by individual Decision Trees are then aggregated using the majority voting mechanism, as defined in Equation (6).

$$\hat{y} = mode\{h_1(x), h_2(x), \dots, h_T(x)\} \quad (6)$$

The `RandomForestClassifier` class from the scikit-learn library was used to implement the Random Forest model. The model was configured with `n_estimators=100`, `criterion='gini'`, `max_depth=None`, `min_samples_split=2`, `class_weight=None`, `max_features='sqrt'`, and `random_state=42`. These settings indicate that the maximum tree depth was not restricted and that no class weighting was applied during model training.

Support Vector Machine (SVM)

SVM is a supervised machine learning algorithm that learns decision boundaries from training data for classification tasks [44]. The algorithm operates by identifying the optimal hyperplane that separates different classes within the feature space [45]. This hyperplane is selected to maximize the margin between the closest data points from each class, known as support vectors. The general decision function of SVM is expressed in Equation (7)

$$f(x) = w^T \cdot x + b \quad (7)$$

where $f(x)$ is the decision function, w is the weight vector, x is the feature vector, and b is the bias term. The TF-IDF features generated in this study are represented in a high-dimensional feature space, where linear decision boundaries are generally effective for separating sentiment classes. Therefore, the `LinearSVC` implementation from the scikit-learn library was employed with `C=1.0` to control the regularization strength and `class_weight=None`, as class imbalance was addressed using SMOTE. In addition to providing strong classification performance, `LinearSVC` offers higher computational efficiency than kernel-based SVM models. For multiclass classification, `LinearSVC` adopts the one-vs-rest strategy

by constructing one binary classifier for each class using `multi_class='ovr'`. The final sentiment class is determined by the highest decision function score, classifying Gen-AI reviews as negative, neutral, or positive.

Model Evaluation

Model performance was assessed using the testing set, which was not involved in model training, to evaluate the models' ability to classify previously unseen data [46]. A confusion matrix was employed as the primary evaluation method because it provides a comprehensive assessment of classification performance and is widely used for multiclass classification problems [47]. In this study, the confusion matrix was used to evaluate three sentiment classes: negative, neutral, and positive. It compares the predicted labels with the ground-truth labels and serves as the basis for calculating the evaluation metrics, including accuracy, precision, recall, and F1-score. Accuracy measures the overall correctness of the model, whereas precision, recall, and F1-score evaluate its performance for each sentiment class. Higher evaluation metric values indicate better classification performance. The formulas for these evaluation metrics are presented in Equations (8)–(11).

$$Accuracy = \frac{\sum_{i=1}^C CM_{ii}}{\sum_{i=1}^C \sum_{j=1}^C CM_{ij}} \quad (8)$$

$$Precision = \frac{TP}{TP + FP} \quad (9)$$

$$Recall = \frac{TP}{TP + FN} \quad (10)$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (11)$$

The accuracy metric in Equation (8) was computed using the multiclass confusion matrix formulation, as this study involves three sentiment classes. For multiclass evaluation, precision, recall, and F1-score were calculated for each class using the one-vs-rest (OvR) approach and subsequently summarized using macro averaging and weighted averaging. Macro averaging assigns equal importance to each class by computing the unweighted mean of the evaluation metrics, whereas weighted averaging accounts for the proportion of samples in each class. In addition, balanced accuracy was reported as the average recall across all classes, providing a more representative evaluation for imbalanced datasets.

Hyperparameter Configuration

The hyperparameter configurations used throughout the study, including the data splitting strategy, TF-IDF feature extraction parameters, SMOTE configuration, and classification model hyperparameters, are presented in Table 1 to support the reproducibility of the study.

Stage	Hyperparameter
Data Splitting	<code>test_size = 0.2</code>
	<code>random_state = 42</code>
Feature Extraction (TF-IDF)	<code>stratify = y</code>
	<code>ngram_range = (1,1)</code>
	<code>max_df = 0.95</code>
	<code>min_df = 5</code>
	<code>k_neighbors = 5</code>
SMOTE	<code>random_state = 42</code>
	<code>sampling_strategy = auto</code>
	<code>n_estimators = 100</code>
	<code>max_depth = None</code>
	<code>max_features = sqrt</code>
	<code>min_samples_split = 2</code>
	<code>random_state = 42</code>
SVM (LinearSVC)	<code>C = 1.0</code>
	<code>kernel = linear</code>
	<code>multi_class = ovr</code>
	<code>tol = 0.0001</code>
	<code>max_iter = 1000</code>

RESULT AND DISCUSSION

Result

Data Preparation

The data preparation stage was conducted to prepare the dataset before feature extraction and sentiment classification. This stage consisted of dataset characteristic analysis, text preprocessing, and sentiment labeling.

Dataset Distribution

The dataset used in this study consists of 50,000 user reviews collected from five Gen-AI applications. Each application contributes an equal number of reviews (10,000 reviews or 20% of the total dataset), resulting in a balanced distribution across applications and reducing potential bias toward any particular application. However, the overall distribution of the Star_Rating attribute is imbalanced. Five-star ratings dominate the dataset with 28,414 reviews, whereas three-star ratings have the fewest reviews (3,003), as presented in Table 2.

Table 2. Dataset Distribution						
App	Rating					Total
	1	2	3	4	5	
ChatGPT	2085	397	599	1210	5709	10000
Claude	1980	570	724	1114	5612	10000
Google Gemini	2739	524	697	1039	5001	10000
Microsoft Copilot	1127	280	556	1235	6802	10000
Perplexity	3137	436	454	683	5290	10000
Total	11068	2207	3030	5281	28414	50000

Preprocessing Data

The Review_Text attribute underwent text cleaning, followed by case folding, tokenization, stopwords removal, and lemmatization. The preprocessing results are presented in Table 3.

Table 3. Preprocessing Result	
Preprocessing	Result
Dataset	"ChatGPT is an amazing app for anyone who wants quick answers, creative ideas, and helpful guidance. It's user-friendly, fast, and very reliable. I use it daily for content creation and problem-solving. Great experience overall!" 😊 🌸 ❤️
Cleansing	ChatGPT is an amazing app for anyone who wants quick answers creative ideas and helpful guidance Its user friendly fast and very reliable I use it daily for content creation and problem solving Great experience overall
Case Folding	chatgpt is an amazing app for anyone who wants quick answers creative ideas and helpful guidance its user friendly fast and very reliable i use it daily for content creation and problem solving great experience overall
Tokenizing	[chatgpt, is, an, amazing, app, for, anyone, who, wants, quick, answers, creative, ideas, and, helpful, guidance, its, user, friendly, fast, and, very, reliable, i, use, it, daily, for, content, creation, and, problem, solving, great, experience, overall]
Stopwords	[chatgpt, amazing, anyone, wants, quick, answers, creative, ideas, helpful, guidance, user, friendly, fast, reliable, use, daily, content, creation, problem, solving, great, experience, overall]
Lemmatizing	[chatgpt, amaze, anyone, want, quick, answer, creative, idea, helpful, guidance, user, friendly, fast, reliable, use, daily, content, creation, problem, solve, great, experience, overall]

Sentiment Labeling

Sentiment labeling was performed using a practical rating-based labeling approach, where ratings of 1–2 were classified as negative, a rating of 3 classified as neutral, and ratings of 4–5 classified as positive. The distribution of the sentiment labels is illustrated in Figure 2.

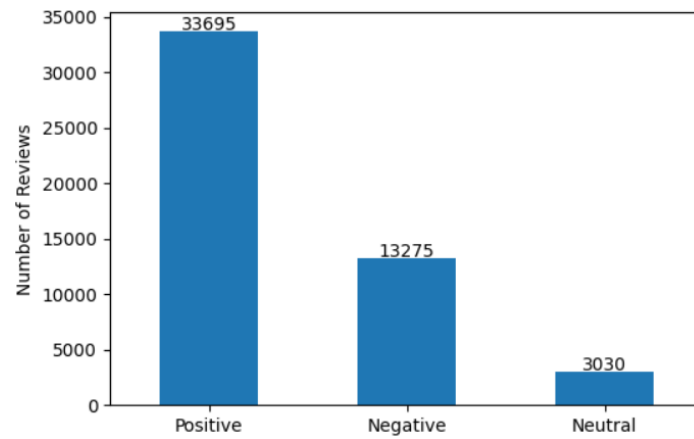


Figure 2. Sentiment Class Distribution

The sentiment labeling results revealed an imbalanced class distribution among the three sentiment categories. The positive class contained the largest number of reviews, with 33,695 instances, followed by the negative class with 13,275 reviews, while the neutral class contained only 3,030 reviews. This distribution indicates a substantial class imbalance, where the number of instances across sentiment classes was not proportional.

Data Splitting and TF-IDF Result

After preprocessing, the dataset was split into training and testing sets using an 80:20 ratio, with 80% of the data used for model training and the remaining 20% used for model testing. The training set consisted of 40,000 reviews, comprising 26,956 positive reviews (67.38%), 2,424 neutral reviews (6.06%), and 10,620 negative reviews (26.55%). The testing set contained 10,000 reviews with the same class distribution: 6,739 positive reviews (67.38%), 606 neutral reviews (6.06%), and 2,655 negative reviews (26.55%), confirming that the stratified train-test split preserved the original class proportions. Subsequently, TF-IDF feature extraction generated 5,589 features from the training data. These features served as the numerical representation for sentiment classification using the Random Forest and SVM models.

Experimental Scenarios

Two experimental scenarios were evaluated: the first without SMOTE and the second with SMOTE applied to the training data for class imbalance handling. In the first scenario, the models were trained on the original imbalanced training set, consisting of 26,956 positive reviews, 10,620 negative reviews, and 2,424 neutral reviews. After applying SMOTE, the training data became balanced, with each sentiment class containing 26,956 samples. The resulting training data distributions for both scenarios are presented in Figure 3.

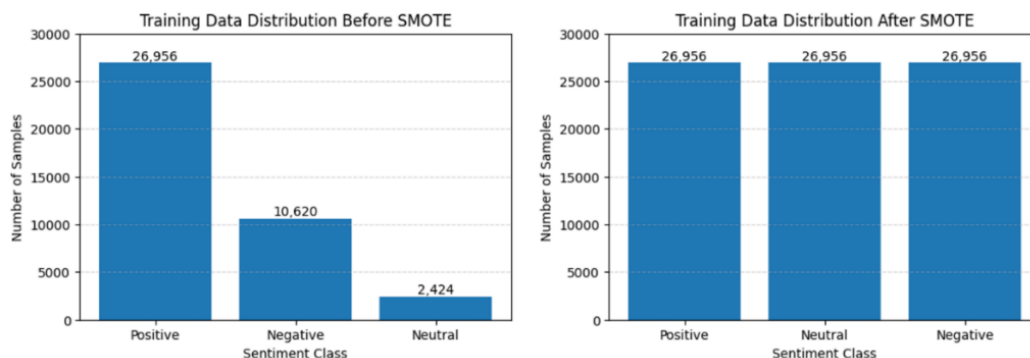


Figure 3. Distribution of Training Data Before and After SMOTE

Following model training, the performance of each model was evaluated using a confusion matrix, from which the accuracy, precision, recall, and F1-score metrics were computed.

Model Comparison

Random Forest Performance with and without SMOTE

The confusion matrices for the Random Forest model under both experimental scenarios are presented in Figure 4. Without SMOTE, the model achieved the highest classification performance for the positive sentiment class, correctly classifying 94.24% of positive reviews (6.351). For the negative class, 73.60% of reviews (1.954) were correctly classified. In contrast, the model showed difficulty of correctly classify any neutral reviews (0%). Instead, 58.25% of neutral reviews (353) were misclassified as positive, while the remaining 41.75% (253) were misclassified as negative.

After applying SMOTE, the Random Forest model showed an improved ability to recognize the neutral class, with the correct classification rate increasing to 3.96% (24). However, most neutral reviews were still misclassified as positive (48.18%, 292 reviews) or negative (47.85%, 290 reviews). The classification performance for the negative class also improved, with the proportion of correctly classified reviews increasing to 77.74% (2.064 reviews). In contrast, the correct classification rate for the positive class decreased from 94.24% to 89.26%, corresponding to 6.015 correctly classified reviews.

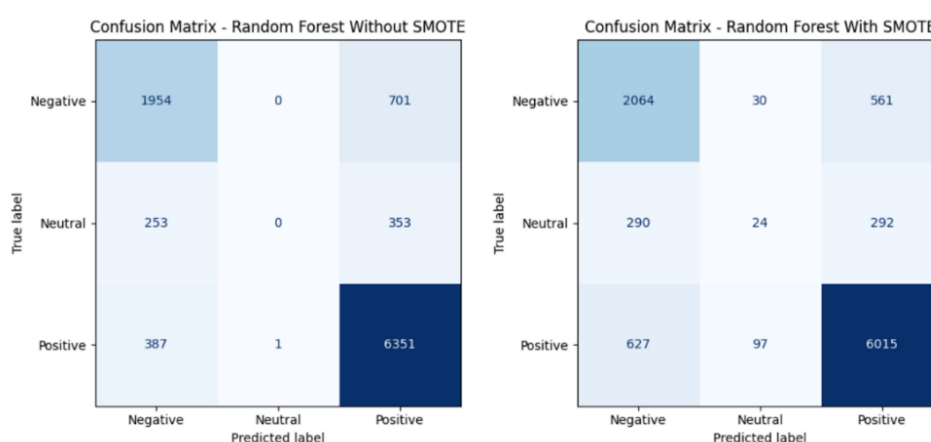


Figure 4. Confusion Matrix of Random Forest Model With and Without SMOTE

As summarized in Table 4, the Random Forest model achieved an overall accuracy of 83% without SMOTE. The best performance was observed for the positive sentiment class, with a precision of 0.86, a recall of 0.94, and an F1-score of 0.90. In contrast, the neutral class obtained precision, recall, and F1-score values of 0.00, indicating that the model was unable to correctly identify any neutral reviews due to the severe class imbalance. After applying SMOTE, the overall accuracy decreased slightly to 81%. The positive class achieved a precision of 0.88, a recall of 0.89, and an F1-score of 0.88. Meanwhile, the neutral class showed improved performance, with the precision, recall, and F1-score increasing to 0.16, 0.04, and 0.06, respectively. These results indicate that SMOTE enabled the model to recognize a portion of the neutral class, although its classification performance remained limited.

Table 4. Random Forest Classification Metrics

Model		Evaluation Metrics			
		Accuracy	Precision	Recall	F1-score
Without SMOTE	Negative	0.83	0.75	0.74	0.74
	Neutral		0.00	0.00	0.00
	Positive		0.86	0.94	0.90
With SMOTE	Negative	0.81	0.69	0.78	0.73
	Neutral		0.16	0.04	0.06
	Positive		0.88	0.89	0.88

SVM Performance with and without SMOTE

The confusion matrices for the SVM model under both experimental scenarios are presented in Figure 5. Without SMOTE, the model correctly classified 92.98% of positive reviews (6.265) and 77.89% of negative reviews (2.068). For the neutral class, only 3.30% of neutral reviews (20) were correctly classified, whereas 53.47% of neutral reviews (324) were misclassified as positive and 43.23% (262) were misclassified as negative.

After applying SMOTE, the SVM model showed a substantial improvement in recognizing the neutral class, with the correct classification rate increasing to 30.53% (185 reviews). The proportion of misclassified neutral reviews also decreased, with 31.85% (193 reviews) classified as positive and 37.62% (228 reviews) classified as negative. However, this improvement was accompanied by a decline in the classification performance of the positive and negative classes. The correct classification rate decreased to 80.69% of positive reviews (5.438) and 73.97% of negative reviews (1.964).

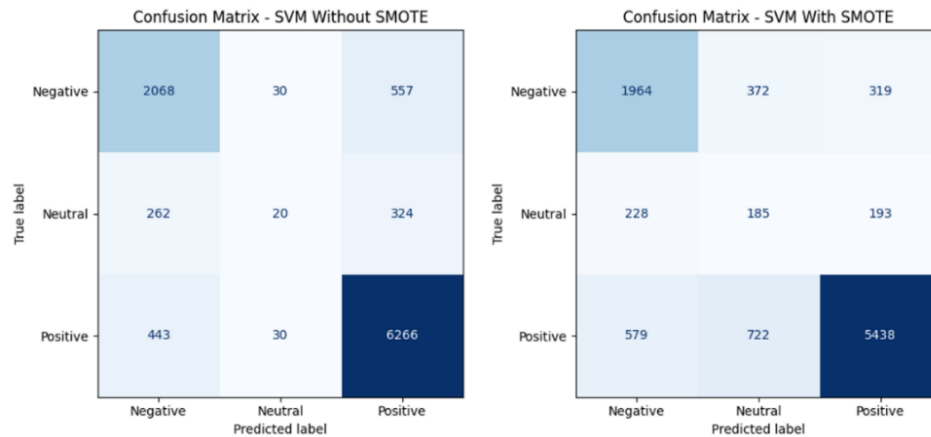


Figure 5. Confusion Matrix of SVM Model With and Without SMOTE

As summarized in Table 5, the SVM model achieved an overall accuracy of 84% without SMOTE. The best performance was observed for the positive sentiment class, with a precision of 0.88, a recall of 0.93, and an F1-score of 0.90. For the neutral class, the model obtained a precision of 0.25, a recall of 0.03, and an F1-score of 0.06, indicating that it was able to recognize only a small proportion of neutral reviews. After applying SMOTE to address class imbalance, the overall accuracy decreased to 76%. The positive class achieved a precision of 0.91, a recall of 0.81, and an F1-score of 0.86. In contrast, the neutral class showed notable improvement, with the precision, recall, and F1-score increasing to 0.14, 0.31, and 0.20, respectively. These results indicate that SMOTE substantially improved the model's ability to recognize neutral reviews, although this improvement was accompanied by a reduction in overall classification accuracy.

Table 5. SVM Classification Metrics

Model		Evaluation Metrics			
		Accuracy	Precision	Recall	F1-score
Without SMOTE	Negative	0.84	0.75	0.78	0.76
	Neutral		0.25	0.03	0.06
	Positive		0.88	0.93	0.90
With SMOTE	Negative	0.76	0.71	0.74	0.72
	Neutral		0.14	0.31	0.20
	Positive		0.91	0.81	0.86

Comparison of All Experimental Scenarios

Table 6 presents a comparison of the performance of all models across the experimental scenarios using the following metrics: Accuracy (Acc), Balanced Accuracy (B-Acc), Weighted Precision (W-Prec), Weighted Recall (W-Rec), Weighted F1-score (W-F1), Macro Precision (M-Prec), Macro Recall (M-Rec), and Macro F1-score (M-F1).

Table 6. Comparison of All Experimental Result

Experiment	Acc	B-Acc	W-Prc	W-Rec	W-F1	M-Prec	M-Rec	M-F1
RF	0.83	0.56	0.78	0.83	0.80	0.54	0.56	0.55
RF + SMOTE	0.81	0.57	0.78	0.81	0.79	0.58	0.57	0.56
SVM	0.84	0.58	0.80	0.84	0.81	0.62	0.58	0.57
SVM + SMOTE	0.76	0.62	0.81	0.76	0.78	0.59	0.62	0.59

Based on the comparison of all experimental scenarios, the SVM model without SMOTE achieved the best overall performance, with an accuracy of 0.84, a weighted recall of 0.84, and a weighted F1-score of 0.81. The Random Forest model without SMOTE also demonstrated competitive performance, achieving an

accuracy of 0.83, a weighted recall of 0.83, and a weighted F1-score of 0.80. After applying SMOTE, both models exhibited a decline in overall accuracy and weighted evaluation metrics. The accuracy of the Random Forest model decreased from 0.83 to 0.81, whereas the SVM model showed a more substantial reduction from 0.84 to 0.76. Similarly, the weighted F1-score decreased from 0.80 to 0.79 for Random Forest and from 0.81 to 0.78 for SVM.

In contrast, balanced accuracy and the macro evaluation metrics improved after applying SMOTE. For the Random Forest model, balanced accuracy increased from 0.56 to 0.57, while the macro F1-score improved from 0.55 to 0.56. A more pronounced improvement was observed for the SVM model, with balanced accuracy increasing from 0.58 to 0.62 and the macro F1-score from 0.57 to 0.59.

Word Cloud Visualization



Figure 6. Word Cloud Visualization

The word cloud provides a visual representation of the dominant terms across positive, negative, and neutral Gen-AI user reviews. As shown in Figure 6, the dominant terms are primarily associated with users' experiences and perceptions of Gen-AI application performance. In the positive sentiment class, words such as “good”, “best”, “well”, “thank”, and “love” appear with high frequency, reflecting favorable user opinions. In contrast, the negative sentiment class is characterized by words including “bad”, “don’t”, “doesn’t”, “response”, and “limit” suggesting that many negative reviews are related to inaccurate responses, technical issues, and user dissatisfaction with the provided services. Meanwhile, the neutral sentiment class is dominated by terms such as “good”, “time”, “can’t”, “well”, and “answer” indicating that neutral reviews contain a mixture of appreciative and critical expressions. Overall, the word cloud visualization provides an intuitive overview of the vocabulary characteristics associated with each sentiment class and complements the quantitative sentiment classification results.

In addition to opinion-related words, the word cloud visualization also highlights the frequent occurrence of Gen-AI application names across different sentiment classes. Terms such as “chatgpt”, “claude”, and “copilot” frequently appear in the positive class, whereas “gemini”, “chatgpt”, and “perplexity” are common in the negative class. In the neutral class, “gemini”, “claude”, and “chatgpt” are among the most frequent terms. This finding indicates that users often mention application names explicitly when expressing their opinions. Therefore, a more detailed analysis based on the App and Review_Theme attributes was conducted to identify the dominant review themes for each Gen-AI application.

Descriptive Analysis of User Sentiment Based on Application and Review Theme Visualization

As one of the novel contributions of this study, the review aspects receiving the greatest number of complaints and positive feedback were further analyzed using the App and Review_Theme attributes available in the dataset. Their distributions are more clearly illustrated through the heatmaps presented in Figures 7–9.

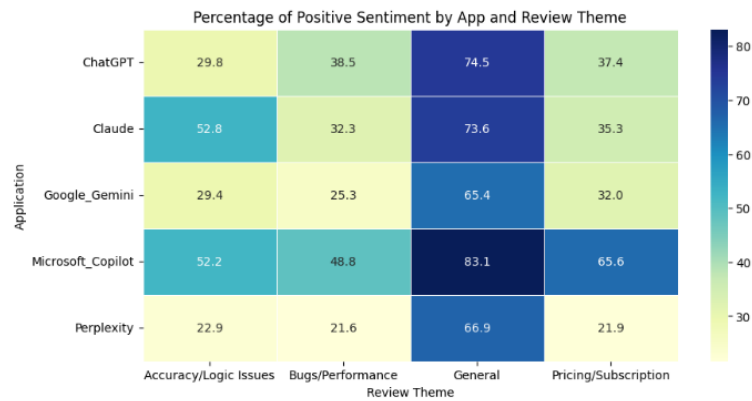


Figure 7. Heatmap of Positive Sentiment Distribution by Application and Review Theme

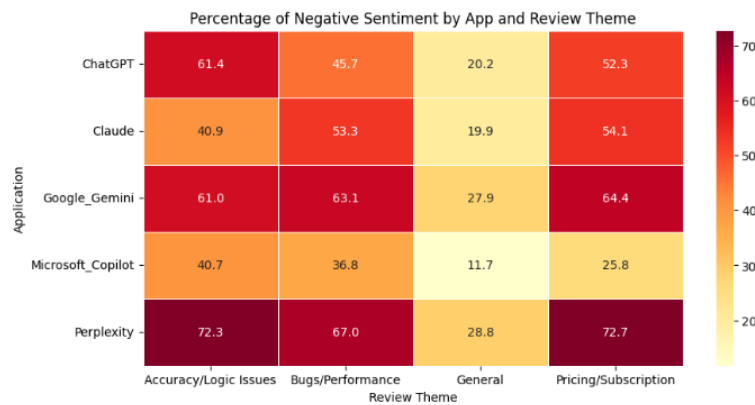


Figure 8. Heatmap of Negative Sentiment Distribution by Application and Review Theme

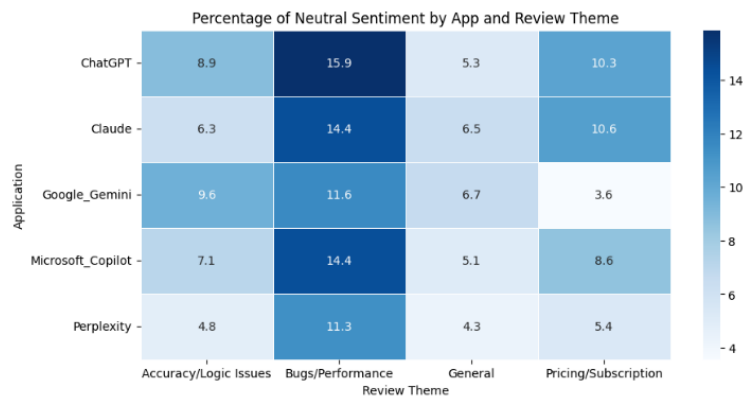


Figure 9. Heatmap of Neutral Sentiment Distribution by Application and Review Theme

In the positive sentiment class, the General theme accounted for the highest proportion of reviews across all applications, with the highest percentage observed for Microsoft Copilot (83.1%), followed by ChatGPT (74.5%) and Claude (73.6%). An examination of representative reviews within the General theme indicates that these positive reviews primarily reflect users' overall experiences with the applications rather than discussing specific issues related to Accuracy/Logic Issues, Bugs/Performance, or Pricing/Subscription. In contrast, the Accuracy/Logic Issues, Bugs/Performance, and Pricing/Subscription themes exhibited relatively higher proportions of negative sentiment. Among the evaluated applications, Perplexity recorded the highest percentage of negative reviews in all three themes, with 72.7% for Pricing/Subscription, 72.3% for Accuracy/Logic Issues, and 67.0% for Bugs/Performance. Meanwhile, the neutral sentiment class generally accounted for less than 16% of reviews across all applications and review themes, indicating a relatively small proportion overall. This finding is consistent with the dataset distribution, in which neutral reviews represent only a small fraction of the entire dataset.

Discussion

Based on the evaluation and comparative analysis of the models, the SVM model without SMOTE achieved the highest overall accuracy and weighted F1-score across all experimental scenarios. These results indicate that SVM is more effective for classifying TF-IDF-based text representations, as it can construct an optimal decision boundary in a high-dimensional and sparse feature space. SVM without SMOTE achieved the highest overall accuracy and weighted F1-score, but its performance on the neutral class remained poor. This finding indicates that the model's overall performance does not fully reflect its ability to recognize all sentiment classes, particularly the minority class, as evidenced by the low recall and F1-score for neutral sentiment.

In contrast, applying SMOTE to address class imbalance in the training data did not improve the overall classification accuracy and instead reduced the accuracy of both models. Nevertheless, oversampling improved the ability of both models to recognize the neutral sentiment class, as reflected by the higher recall and macro F1-score. These findings experimentally demonstrate the trade-off associated with applying SMOTE to text classification models, where improvements in minority-class recognition are achieved at the expense of overall classification performance.

The decrease in classification accuracy under the second experimental scenario is likely attributable to the feature representation characteristics of the neutral sentiment class, which was the most difficult for the models to learn. SMOTE operates in the numerical TF-IDF feature space rather than generating linguistically valid textual reviews. Therefore, synthetic neutral vectors may overlap with positive and negative feature regions. As a result, when the TF-IDF feature representations of neutral reviews are located closer to those of the other sentiment classes, the generated synthetic samples may enlarge the overlap between classes, making the decision boundaries less distinct and increasing prediction bias. This interpretation is consistent with the confusion matrix results for both models, where most misclassifications occurred in the neutral sentiment class, which was frequently predicted as either positive or negative. Table 7 presents examples of these misclassifications.

Table 7. Examples of Misclassification Neutral Reviews

Review	True Label	Predicted Label	Interpretation
this app is amazing but sometimes it forgets the previous information	Neutral	Positive	Contains strong positive words followed by a mild complaint
chatgpt is really good but for my imagination it's slow but its best 🤖👍👍👍👍👍	Neutral	Positive	Dominated by positive expressions despite containing criticism
i have been using it for some time. There wasnt any issues until yesterday. i like the app, but it's been stuck on telling me it's out of messages until 5 pm since yesterday. Please fix it	Neutral	Negative	Contains past positive experience followed by multiple complaints
good but sametime it didn't get what I want which annoyed me	Neutral	Negative	Contains both appreciation and dissatisfaction, with negative expressions dominating

In addition to comparing the performance of sentiment classification algorithms, this study provides a more comprehensive understanding of user perceptions through an analysis of the Review_Theme distribution across five Gen-AI applications. The results indicate that positive reviews are predominantly associated with the General theme, suggesting that most users express overall satisfaction with their experience rather than commenting on specific aspects of the applications. In contrast, negative reviews are primarily related to Accuracy/Logic Issues, Bugs/Performance, and Pricing/Subscription, indicating that these aspects represent the main concerns reported by users. Meanwhile, neutral reviews are dominated by the Bugs/Performance theme, suggesting that users often express both satisfaction and dissatisfaction with application performance in a balanced manner. These findings provide practical implications for Gen-AI application developers by highlighting the aspects that should be prioritized for improvement. Furthermore, they demonstrate that 3-star reviews should not be overlooked, as they often contain constructive criticism and suggestions that can support continuous application improvement.

Compared with the study by Nurhasanah *et al.* [18], which used the same dataset, the present study achieved a higher accuracy of 84% using the SVM algorithm, outperforming the 82% accuracy reported for the Naïve Bayes classifier. Furthermore, this study extends the previous work by not only evaluating overall

classification accuracy but also investigating the effect of SMOTE on the models' ability to recognize the minority class. The findings are also consistent with those reported by Imanuddin *et al.* [19] and Shobil *et al.* [22] who demonstrated that SVM provides strong classification performance for high-dimensional datasets by constructing an optimal hyperplane over TF-IDF feature representations while maintaining computational efficiency. However, unlike several previous studies that reported improved classification accuracy after applying SMOTE [23][24], the results of this study are consistent with those of Pambudi *et al.* [48] which showed that SMOTE may reduce overall classification accuracy despite producing a more balanced prediction distribution for the minority class. Therefore, the effectiveness of SMOTE cannot be generalized to all sentiment analysis tasks but should be evaluated in the context of the linguistic characteristics of the review data.

A limitation of this study is the use of rating-based labeling to assign sentiment labels. This approach assumes that user ratings accurately reflect the sentiment expressed in the review text, although discrepancies between ratings and textual content may occur in practice. Neutral reviews frequently include a mixture of positive and negative opinions, which complicates the classification process. This study is also limited to a comparison of two traditional machine learning models, namely Random Forest and SVM. Furthermore, the experimental results are based on a single stratified train–test split and were not supported by statistical significance testing.

Future research may improve sentiment labeling by validating rating-based labels using lexicon-based approaches such as VADER (Valence Aware Dictionary and sEntiment Reasoner) to enhance the consistency between sentiment labels and review content. In addition, deep learning approaches, including transformer-based models and word embedding techniques such as Word2Vec, BERT, and FastText, could be explored to better capture semantic relationships and contextual information in review texts. Model evaluation could also be strengthened through Stratified k-Fold Cross-Validation, repeated experiments with multiple random seeds, and statistical significance testing using methods such as McNemar's test, bootstrap, or confidence intervals. Finally, the effectiveness of alternative class imbalance handling techniques, including Adaptive Synthetic Sampling (ADASYN), Random Oversampling (ROS), Borderline-SMOTE, undersampling, class-weighted learning, and cost-sensitive SVM, should be compared with SMOTE.

CONCLUSION

This study successfully classified user sentiments toward Gen-AI applications using the Random Forest and SVM algorithms. A dataset consisting of 50,000 user reviews from ChatGPT, Microsoft Copilot, Google Gemini, Perplexity, and Claude was categorized into three sentiment classes: positive, neutral, and negative. The results demonstrate that both algorithms achieved strong classification performance, with Random Forest attaining an accuracy of 83% and SVM achieving 84%. Based on the comparison of the two experimental scenarios, SVM without SMOTE achieved the highest overall accuracy and weighted F1-score, but its ability to identify neutral reviews remained limited. SMOTE improved the neutral-class recall and F1-score, especially for SVM, although it reduced overall accuracy. Therefore, the preferred model depends on the evaluation objective: SVM without SMOTE is more appropriate when overall predictive accuracy is prioritized, whereas SVM with SMOTE may be considered when improved neutral-class detection is required. The results reveal a trade-off between overall classification performance and minority-class recognition when SMOTE is used for imbalanced sentiment classification. Furthermore, the findings may assist Gen-AI application developers in understanding user perceptions, recognizing areas for improvement, and informing data-driven efforts to enhance application quality and services.

REFERENCES

- [1] A. Holzinger, K. Keiblinger, P. Holub, K. Zatloukal, and H. Müller, "AI for life: Trends in artificial intelligence for biotechnology," *New Biotechnology*, vol. 74, pp. 16–24, 2023, doi: 10.1016/j.nbt.2023.02.001.
- [2] P. Ågerfalk, P. J., Conboy, K., Crowston, K., Eriksson Lundström, J. S., Jarvenpaa, S., Ram, S., & Mikalef, "Artificial Intelligence in Information Systems: State of the Art and Research Roadmap," *Communications of the Association for Information Systems*, vol. 50, no. 1, pp. 420–438, 2022, doi: 10.17705/1CAIS.05017.
- [3] M. F. Shahzad, S. Xu, X. An, and M. Asif, "Are Generative AI Technologies Transforming Education for the 21st Century? Research Trends, Challenges, and Benefits," *Sage Open*, vol. 15, no. 3, Art. no. 21582440251368594, 2025, doi: 10.1177/21582440251368594.
- [4] M. Trigka and E. Dritsas, "The Evolution of Generative AI: Trends and Applications," *IEEE Access*, vol. 13, pp. 98504–98529, 2025, doi: 10.1109/ACCESS.2025.3574660.
- [5] S. Bouguettaya, F. Pupo, M. Chen, and G. Fortino, "A Meta-Survey of Generative AI in Education: Trends, Challenges, and Research Directions," *Big Data and Cognitive Computing*, vol. 9, no. 9, p. 237, 2025, doi: 10.3390/bdcc9090237.

- [6] S. Feuerriegel, J. Hartmann, C. Janiesch, and P. Zschech, "Generative AI," *Business and Information Systems Engineering*, vol. 66, no. 1, pp. 111–126, 2024, doi: 10.1007/s12599-023-00834-7.
- [7] X. Sun and W. He, "Exploring the Frontier of Generative AI: Evolution, Applications, and Future Trends," *International Journal of Crowd Science*, vol. 9, no. 4, pp. 269–277, 2025, doi: 10.26599/IJCS.2024.9100032.
- [8] R. Alabduljabbar, "User-centric AI: Evaluating the Usability of Generative AI Applications Through User Reviews on App Stores," *PeerJ Computer Science*, vol. 10, Art. no. e2421, 2024, doi: 10.7717/peerj-cs.2421.
- [9] T. Chen, P. Samaranayake, X. Cen, M. Qi, and Y.-C. Lan, "The Impact of Online Reviews on Consumers' Purchasing Decisions: Evidence From an Eye-Tracking Study," *Frontiers in Psychology*, vol. 13, Art. no. 865702, 2022, doi: 10.3389/fpsyg.2022.865702.
- [10] F. Muttakin, N. Andrika, and Salsabila, "Sentiment Analysis of Shoe Product Reviews on Indonesian E-Commerce Platform Using Lexicon Based and Support Vector Machine," *Jurnal Teknik Informatika (JUTIF)*, vol. 6, no. 2, pp. 839–854, 2025, doi: 10.52436/1.jutif.2025.6.2.3800.
- [11] H. Imaduddin, F. Y. A'la, and Y. S. Nugroho, "Sentiment Analysis in Indonesian Healthcare Applications using IndoBERT Approach," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 8, pp. 113–117, 2023, doi: 10.14569/IJACSA.2023.0140813.
- [12] M. D. R. Haque and S. Rubya, "An Overview of Chatbot-Based Mobile Mental Health Apps: Insights From App Description and User Reviews," *JMIR mHealth uHealth*, vol. 11, Art. no. e44838, 2023, doi: 10.2196/44838.
- [13] J. Sangeetha and U. Kumaran, "Sentiment Analysis of Amazon User Reviews Using a Hybrid Approach," *Measurement: Sensors*, vol. 27, Art. no. 100790, 2023, doi: 10.1016/j.measen.2023.100790.
- [14] A. Albladi, M. Islam, and C. Seals, "Sentiment Analysis of Twitter Data Using NLP Models: A Comprehensive Review," *IEEE Access*, vol. 13, pp. 30444–30468, 2025, doi: 10.1109/ACCESS.2025.3541494.
- [15] S. Dahl, M. Bøgsted, T. Sagi, and C. Vesteghem, "Performance of Natural Language Processing for Information Extraction From Electronic Health Records Within Cancer: Systematic Review," *JMIR Medical Informatics*, vol. 13, Art. no. e68707, 2025, doi: 10.2196/68707.
- [16] F. Aftab *et al.*, "A Comprehensive Survey on Sentiment Analysis Techniques," *International Journal of Technology*, vol. 14, no. 6, pp. 1288–1298, 2023, doi: 10.14716/ijtech.v14i6.6632.
- [17] R. D. Kurniawan, A. Yohannis, and W. T. Atmojo, "Sentiment Analysis of Getcontact Application Reviews on Google Play Store Using Naive Bayes Algorithm," *Jurnal Teknik Informatika (JUTIF)*, vol. 6, no. 4, pp. 2848–2858, 2025, doi: 10.52436/1.jutif.2025.6.4.5248.
- [18] R. Nurhasanah and V. A. Elyakim, "Analisis Sentimen Ulasan Pengguna Generative AI Menggunakan Naïve Bayes Berbasis Python," *INSOLOGI: Jurnal Sains dan Teknologi*, vol. 5, no. 3, pp. 875–890, 2026, doi: 10.55123/insologi.v5i3.8050.
- [19] S. H. Imanuddin, K. Adi, and R. Gernowo, "Sentiment Analysis on Satuschat Application Using Support Vector Machine Method," *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 5, no. 3, pp. 143–149, 2023, doi: 10.35882/jeemi.v5i3.304.
- [20] M. R. Rahman, A. F. Diansyah, and Hanafi, "Sentiment Analysis on the Shopee Application on Playstore Using the Random Forest Classification Method," *Inform : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi*, vol. 9, no. 1, pp. 20–24, 2024, doi: 10.25139/inform.v9i1.5465.
- [21] R. R. Pratama and R. R. Suryono, "Performance Comparison of Naive Bayes, Support Vector Machine and Random Forest Algorithms for Apple Vision Pro Sentiment Analysis," *Jurnal Teknik Informatika (JUTIF)*, vol. 6, no. 1, pp. 31–40, 2025, doi: 10.52436/1.jutif.2025.6.1.4035.
- [22] A. Shobil and S. Al-Haggee, "User Sentiment Analysis of Solar Energy Monitoring Apps: Insights From Google Play Reviews," *IEEE Access*, vol. 14, pp. 38421–38433, 2026, doi: 10.1109/ACCESS.2026.3670366.
- [23] D. Kartika and S. Defit, "Optimizing Naive Bayes for Sentiment Analysis of M-Passport Reviews Using N-Gram and Synthetic Minority Over-sampling Technique," *Jurnal Teknik Informatika (JUTIF)*, vol. 7, no. 3, pp. 2798–2811, 2026, doi: 10.52436/1.jutif.2026.7.3.5853.
- [24] P. P. Putra, M. K. Anam, A. S. Chan, A. Hadi, N. Hendri, and A. Masnur, "Optimizing Sentiment Analysis on Imbalanced Hotel Review Data Using SMOTE and Ensemble Machine Learning Techniques," *Journal of Applied Data Sciences*, vol. 6, no. 2, pp. 936–951, 2025, doi: 10.47738/jads.v6i2.618.
- [25] Badriyah, T. Chamidy, and Suhartono, "Application of SMOTE in Sentiment Analysis of MyXL User Reviews on Google Play Store," *JISKA (Jurnal Informatika Sunan Kalijaga)*, vol. 10, no. 1, pp. 74–86, 2025, doi: 10.14421/jiska.2025.10.1.74-86.
- [26] J. Kachhia, "The Generative AI Ecosystem: 50K User Reviews 2026," *Kaggle*, 2026. <https://www.kaggle.com/datasets/jahnavikachhia23/the-generative-ai-ecosystem-50k-user-reviews-2026> (accessed Jun. 25, 2026).
- [27] H. Syah and A. Witanti, "Analisis Sentimen Masyarakat Terhadap Vaksinasi COVID-19 pada Media Sosial Twitter Menggunakan Algoritma Support Vector Machine (SVM)," *Jurnal Sistem Informasi dan Informatika (SIMIKA)*, vol. 5, no. 1, pp. 59–67, 2022, doi: 10.47080/SIMIKA.V5I1.1411.
- [28] I. Najiyah and M. Rizal, "Analisis Sentimen Media Sosial X Terhadap Kebijakan Presiden Republik Indonesia Prabowo Subianto," *JURNAL FASILKOM*, vol. 15, no. 3, pp. 443–455, 2025, doi: 10.37859/jf.v15i3.10385.
- [29] K. Munawaroh and Alamsyah, "Performance Comparison of SVM, Naïve Bayes, and KNN Algorithms for Analysis of Public Opinion Sentiment Against COVID-19 Vaccination on Twitter," *Journal of Advances in Information Systems and Technology*, vol. 4, no. 2, pp. 113–125, 2023, doi: 10.15294/jaist.v4i2.59493.
- [30] N. W. S. Saraswati, C. P. Yanti, I. D. M. K. Muku, and A. P. R. D. Dewa, "Evaluation Analysis of the Necessity of Stemming and Lemmatization in Text Classification," *MATRIK: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, vol. 24, no. 2, pp. 321–332, 2025, doi: 10.30812/matrik.v24i2.4833.
- [31] E. F. Baharsyah, Armanto, T. H. B. Aviani, and C. Wulandari, "Analisis Sentimen Pengguna Terhadap Aplikasi Belajar Online Ruang Guru Pada Ulasan Google Play Store Menggunakan Algoritma Naïve Bayes dan Support Vector Machine," *Innovative: Journal Of Social Science Research*, vol. 4, no. 3, pp. 2965–2979, 2024, doi: 10.31004/innovative.v4i3.10769.
- [32] K. Hossain, L. Li, U. Ghani, and M. S. Hossain, "Unveiling the Impact of Service Attributes and Review Scores on Sentiment: A Deep Learning and Feature Engineering Approach to UberEats Reviews," *International Journal of Engineering Business Management*, vol. 17, pp. 1–14, 2025, doi: 10.1177/18479790251341980.
- [33] S. A. H. Bahtiar, C. K. Dewa, and A. Luthfi, "Comparison of Naïve Bayes and Logistic Regression in Sentiment Analysis on Marketplace Reviews Using Rating-Based Labeling," *Journal of Information Systems and Informatics*, vol. 5, no. 3, pp. 915–

927, 2023, doi: 10.51519/journalisi.v5i3.539.

- [34] D. Shalikhah and Alamsyah, "Improved Accuracy of Naïve Bayes Algorithm and Support Vector Machine Using Particle Swarm Optimization for Menstrual Cup Sentiment Analysis on Twitter," *Journal of Advances in Information Systems and Technology*, vol. 4, no. 2, pp. 139–148, 2023, doi: 10.15294/jaist.v4i2.59561.
- [35] M. Sivakumar, S. Parthasarathy, and T. Padmapriya, "Trade-off between Training and Testing Ratio in Machine Learning for Medical Image Processing," *PeerJ Computer Science*, vol. 10, Art. no. e2245, 2024, doi: 10.7717/peerj-cs.2245.
- [36] L. Zhang, "Features Extraction Based on Naive Bayes Algorithm and TF-IDF for News Classification," *PLoS One*, vol. 20, no. 7, Art. no. e0327347, 2025, doi: 10.1371/journal.pone.0327347.
- [37] N. A. Semaary, W. Ahmed, K. Amin, P. Pławiak, and M. Hammad, "Enhancing Machine Learning-Based Sentiment Analysis Through Feature Extraction Techniques," *PLoS One*, vol. 19, no. 2, Art. no. e0294968, 2024, doi: 10.1371/journal.pone.0294968.
- [38] R. Obiedat *et al.*, "Sentiment Analysis of Customers' Reviews Using a Hybrid Evolutionary SVM-Based Approach in an Imbalanced Data Distribution," *IEEE Access*, vol. 10, pp. 22260–22273, 2022, doi: 10.1109/ACCESS.2022.3149482.
- [39] Y. Li, Y. Yang, P. Song, L. Duan, and R. Ren, "An Improved SMOTE Algorithm for Enhanced Imbalanced Data Classification by Expanding Sample Generation Space," *Scientific Reports*, vol. 15, no. 1, Art. no. 23521, 2025, doi: 10.1038/s41598-025-09506-w.
- [40] H. A. Salman, A. Kalakech, and A. Steiti, "Random Forest Algorithm Overview," *Babylonian Journal of Machine Learning*, vol. 2024, pp. 69–79, 2024, doi: 10.58496/BJML/2024/007.
- [41] M. . A. Azkiya and Y. N. Ifriza, "Analysis of Coding Stress Impact on Students Programming Skills with Random Forest and C4.5 Algorithms," *Journal of Computer Science and Information*, vol. 18, no. 2, pp. 167–180, 2025, doi: 10.21609/jiki.v18i2.1487.
- [42] J. Hu and S. Szymczak, "A Review on Longitudinal Data Analysis with Random Forest," *Briefings in Bioinformatics*, vol. 24, no. 2, Art. no. bbad002, 2023, doi: 10.1093/bib/bbad002.
- [43] S. Lee, C. Lee, K. G. I. Mun, and D. Kim, "Decision Tree Algorithm Considering Distances Between Classes," *IEEE Access*, vol. 10, pp. 69750–69756, 2022, doi: 10.1109/ACCESS.2022.3187172.
- [44] D. Valkenborg, A. Rousseau, M. Geubbelmans, and T. Burzykowski, "Support Vector Machines," *American Journal of Orthodontics and Dentofacial Orthopedics*, vol. 164, no. 5, pp. 754–757, 2023, doi: 10.1016/j.ajodo.2023.08.003.
- [45] O. I. Gifari, M. Adha, I. R. Hendrawan, and F. F. S. Durrand, "Analisis Sentimen Review Film Menggunakan TF-IDF dan Support Vector Machine," *JIFOTECH (Journal of Information Technology)*, vol. 2, no. 1, pp. 36–40, 2022, doi: 10.46229/jifotech.v2i1.330.
- [46] A. Safitri, M. Risaldi, M. N. R. Alwi, D. F. Surianto, N. Fadilah, and J. M. Parenreng, "Improving Sentiment Classification of Kredit Pintar Reviews Using IndoBERT, SMOTE, and Stacking Ensemble," *Jurnal Teknik Informatika (JUTIF)*, vol. 7, no. 3, pp. 2411–2424, 2026, doi: 10.52436/1.jutif.2026.7.3.5342.
- [47] S. Sathyanarayanan and B. R. Tantri, "Confusion Matrix-Based Performance Evaluation Metrics," *African Journal of Biomedical Research*, vol. 27, no. 4S, pp. 4023–4031, 2024, doi: 10.53555/AJBR.v27i4S.4345.
- [48] S. Pambudi, P. Setiaji, and W. A. Triyanto, "Sentiment Analysis of Fizzo Novel Application Using Support Vector Machine and Naïve Bayes Algorithm with SEMMA Framework," *Jurnal Teknik Informatika (JUTIF)*, vol. 6, no. 4, pp. 1861–1880, 2025, doi: 10.52436/1.jutif.2025.6.4.4875.