JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 10, No. 1, JANUARY, 2025, Pp. 74 – 86
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Application of SMOTE in Sentiment Analysis of MyXL User Reviews on Google Play Store

Badriyah [(1)*], Totok Chamidy [(2)], Suhartono [(3)]
Magister Informatika, Fakultas Sains dan Teknologi, UIN Maulana Malik Ibrahim, Malang, Indonesia
e-mail : 200605220010@student.uin-malang.ac.id, {to2k2013,suhartono}@ti.uin-malang.ac.id.
* Corresponding author.
This article was submitted on 30 July 2024, revised on 26 September 2024, accepted on 26 September 2024, and published on 31 January 2025.

Abstract

Texts that express customer opinions about a product are important input for companies. Companies obtain valuable information from consumer perceptions of marketed products by conducting sentiment analysis. However, real-world text datasets are often unbalanced, causing the prediction results of classification algorithms to be biased towards the majority class and ignore the minority class. This study analyzes the sentiment of MyXL user reviews on the Google Play Store by comparing the performance of the Logistic Regression and Support Vector Machine algorithms in the SMOTE implementation. This analysis uses TF-IDF to extract feature and GridSearchCV to optimize the accuracy, precision, recall, and F1 score evaluation metrics. This study follows several scenarios of dividing training data and test data. SVM implementing SMOTE is the algorithm with the best performance using the division of training data (90%) and test data (10%), resulting in accuracy (73.00%), precision (67.13%), recall (65.82%) and F1 score (66.30%).

Keywords: Sentiment Analysis, Logistic Regression, Support Vector Machine, GridSearchCV, SMOTE

Abstrak

Teks yang mengungkapkan opini pelanggan tentang suatu produk merupakan masukan penting bagi perusahaan. Perusahaan memperoleh informasi berharga dari persepsi konsumen terhadap produk yang dipasarkan dengan melakukan analisis sentimen. Namun, kumpulan data teks dunia nyata seringkali tidak seimbang sehingga menyebabkan hasil prediksi algoritma klasifikasi menjadi bias terhadap kelas mayoritas dan mengabaikan kelas minoritas. Penelitian ini menganalisis sentimen ulasan pengguna MyXL di Google Play Store dengan membandingkan kinerja algoritma Logistic Regression dan Support Vector Machine pada implementasi SMOTE. Analisis ini menggunakan TF-IDF untuk ekstraksi fitur dan GridSearchCV untuk mengoptimalkan metrik evaluasi akurasi, presisi, recall, dan skor F1. Penelitian ini mengikuti beberapa skenario pembagian data latih dan data uji. SVM yang mengimplementasikan SMOTE merupakan algoritma dengan performa terbaik dengan menggunakan pembagian data latih (90%) dan data uji (10%), menghasilkan akurasi (73,00%), presisi (67,13%), recall (65,82%) dan skor F1 (66,30%)**.**

Kata Kunci: Analisis Sentimen, *Logistic Regression*, *Support Vector Machine*, GridSearchCV, SMOTE

1. INTRODUCTION

Customer opinions are one of the main indicators for evaluating a product's success. In a highly competitive world, listening to the voice of customers is crucial to gaining deeper insights into what consumers truly desire and how they respond to changes made by companies. Sentiment analysis allows businesses to understand users' perceptions of their products or services. By analyzing sentiment, companies can identify strengths and weaknesses from the customer's perspective. This identification helps in improving services and developing products to better align with market needs. Additionally, customer opinions serve as valuable information for other customers (Hasibuan & Heriyanto, 2022). A survey of over 7,000 consumers across 11 Asia-

Pacific regions revealed that 76% of consumers seek reviews to validate a company before making a purchase (Cheng & Mani, 2024). Customer experiences significantly influence their purchasing decisions. Companies must actively respond to both positive and negative reviews professionally and promptly, demonstrating that they value customer feedback and are willing to address shortcomings.

Customers express opinions in the form of text, commonly found on social media, online marketplaces, and applications in the Google Play Store. These text data serve as input for machine learning algorithms to analyze and classify sentiment as positive, negative, or neutral. In real-world conditions, datasets often exhibit imbalances in the distribution of data among classes. This imbalance greatly affects the accuracy and reliability of sentiment analysis results, as the classification tends to be biased toward the majority class. For instance, if most training data consists of positive sentiment, the model is likely to be more accurate in detecting positive sentiment but less accurate in identifying neutral or negative sentiment. Such data imbalance causes the model to ignore or misclassify minority class predictions (Khushi et al., 2021). Hence, research is needed to develop and evaluate methods capable of effectively handling data imbalance, ensuring accurate predictions across all classes.

SMOTE is frequently used for sentiment analysis in datasets with imbalanced class distributions. For instance, sentiment analysis of Twitter data about IndihomeCare used SMOTE alongside Support Vector Machine (SVM), AdaBoost, and Particle Swarm Optimization algorithms (Syah et al., 2023). The dataset comprised 1,000 records, with 653 positive reviews and 570 negative reviews. In this study, the application of SMOTE and SVM achieved the highest evaluation scores, proving effective for the given dataset.

Similarly, sentiment analysis of public opinions about antibiotic use in Indonesia employed SVM (Darwis et al., 2023). Out of 1,889 tweets collected through web scraping, 1,631 were negative sentiments, and 258 were positive sentiments. This study implemented SVM with linear, RBF, and polynomial kernels, using RoBERTa-based labeling, cross-validation training, and bigram tokenization methods. Three different text preprocessing scenarios were tested, including TF-IDF feature extraction and SMOTE for addressing class imbalance. The results demonstrated a significant improvement in SVM performance after applying SMOTE.

Another example is sentiment analysis of netizen opinions on various international bag brands, utilizing SMOTE for classification model optimization (Huda et al., 2023). The cleaned dataset from Twitter reviews contained 2,881 reviews, including 1,083 positive, 374 negative, and 1,424 neutral sentiments. This study compared the performance of several algorithms, including Logistic Regression, Multinomial Naïve Bayes, Decision Tree, K-Nearest Neighbors, Random Forest, and SVM. SVM achieved the best performance with an accuracy of 69%. After applying SMOTE, the SVM model's accuracy improved to 82%.

A sentiment analysis study on the metaverse compared Naïve Bayes and Logistic Regression algorithms using SMOTE optimization (Ramadhani & Suryono, 2024). This research analyzed 6,728 comments about the metaverse on the X (formerly Twitter) social media platform using a text mining approach. The optimization results showed that Logistic Regression outperformed Naïve Bayes, achieving a higher accuracy of 95% compared to 91%. The studies mentioned and the planned research are summarized in Table 1.

Logistic Regression and SVM are highly popular algorithms for text classification and are widely used in sentiment analysis. Both algorithms were originally designed for binary classification tasks but have been developed to handle multiclass classification effectively. This study aims to compare the performance of these two algorithms in applying SMOTE for sentiment analysis using an imbalanced multiclass dataset. The novelty of this research lies in comparing the performance of Logistic Regression and SVM using SMOTE, TF-IDF, and GridSearchCV hyperparameter tuning in sentiment analysis of user reviews of the MyXL application on Google Play Store.

Table 1 Related Research

No.	Researchers	Title	Research Features	Research Plan
1	Syah et al. (2023)	Sentiment Analysis of IndihomeCare Twitter Using Comparison of SMOTE, Support Vector Machine, and AdaBoost Algorithms	- Comparison of: 1. SMOTE, SVM 2. SMOTE, SVM, and AdaBoost 3. SMOTE, Particle Swarm Optimization - Twitter Crawling - RapidMiner	- Comparing SMOTE application on Logistic Regression and SVM - GridSearchCV - MyXL user reviews on Google Play Store - Python
2	Darwis et al. (2023)	Support Vector Machine for Public Sentiment Analysis on Antibiotic Use in Indonesia	- Comparing SMOTE and non-SMOTE on SVM with linear, RBF, and polynomial kernels - Preprocessing comparison: slang words by Pujangga and Ramaprokoso, stopwords by NLTK and Sastrawi - TF-IDF - Twitter Crawling - RapidMiner	- Comparing SMOTE and non-SMOTE application on Logistic Regression and SVM - TF-IDF - GridSearchCV - MyXL user review dataset from Google Play Store - Python
3	Huda et al. (2023)	Optimization of Netizen Sentiment Classification Model for Foreign Brand Bags	- Comparing SMOTE application on Logistic Regression, Multinomial Naïve Bayes, Decision Tree, KNN, Random Forest, and SVM - SMOTE applied to the entire dataset before train-test split - TF - Twitter Crawling	- Comparing SMOTE and non-SMOTE application on Logistic Regression and SVM - SMOTE on training data - TF-IDF - GridSearchCV - MyXL user review dataset from Google Play Store
4	Ramadhani & Suryono (2024)	Comparison of Naïve Bayes and Logistic Regression Algorithms for Sentiment Analysis of the Metaverse	- Comparing SMOTE and non-SMOTE on Naïve Bayes and Logistic Regression - SMOTE applied to the entire dataset before train-test split - TF-IDF - Crawling X (formerly Twitter)	- Comparing SMOTE and non-SMOTE application on Logistic Regression and SVM - SMOTE on training data - TF-IDF - GridSearchCV - MyXL user review dataset from Google Play Store

2. METHODS

2.1 Dataset Preparation

The object of this study is the user review dataset for the MyXL application on Google Play Store, comprising 1,000 data points. The dataset contains 613 negative sentiment reviews, 226 neutral sentiment reviews, and 161 positive sentiment reviews. The imbalanced distribution of classes in this dataset is ideal for testing machine learning algorithms designed to handle imbalanced data. The MyXL user review dataset can be downloaded from Kaggle (Audiansyah, 2022). An example of user review data is presented in Table 2.

Table 2 Sample Data of MyXL User Reviews on Google Play Store

Data	Review	Sentiment
Review 1	Tolong dong masalah jaringan hampir setiap hari leg parahh!! Jangan buat kecewa costumer lah	Negative
Review 2	..... kalau bisa masa aktif kartunya di tingkatkan jadi lebih lama. terima kasih	Positive
Review 3	Bisa cek kuota dgn simpel terbaikkkkkkkkkkkk!!!	Positive

2.2 System Design

The system design for this study, as shown in Figure 1, begins with preparing the MyXL user review dataset. Next, the dataset undergoes text preprocessing and keyword weighting using TF-IDF. For model testing, the dataset is split into training and testing datasets. SMOTE oversampling is applied to balance class distributions in the training data. The Logistic Regression and Support Vector Machine classification algorithms are developed with hyperparameter tuning using the GridSearchCV object from the Scikit-learn library. Predictions are made on the testing data, and evaluation is conducted by comparing the predictions against the actual classes.

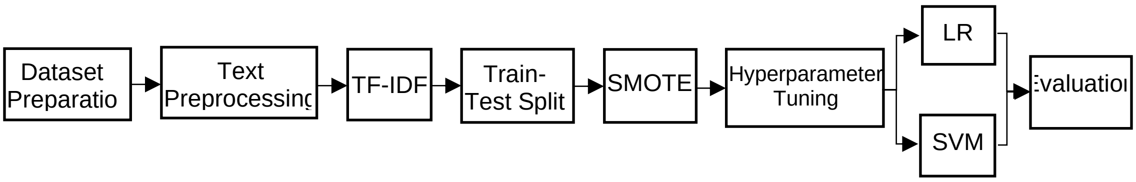

Figure 1 Research System Design

2.3 Text Preprocessing

Text preprocessing involves a series of techniques to transform raw text data into a format suitable for processing and analysis (Haikal et al., 2024). The MyXL user review dataset from Google Play Store, as unstructured raw data, contains various writing styles and language variations. To enable machine learning on this data, it must be converted into a structured format through preprocessing. Table 3 displays examples of MyXL user reviews that have been preprocessed using these techniques.

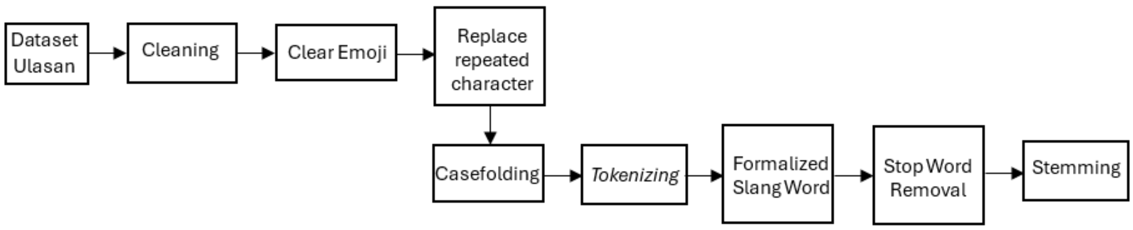

Figure 2 Text Preprocessing Steps

Table 3 Sample Results of Text Preprocessing

Stage	Review 1	Review 2	Review 3
Original Review	Tolong dong masalah jaringan hampir setiap hari leg parahh!! Jangan buat kecewa costumer lah	..... kalau bisa masa aktif kartunya di tingkatkan jadi lebih lama. terima kasih	Bisa cek kuota dgn simpelterbaikkkkkkkkkkkk!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! ! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Cleaning	Tolong dong masalah jaringan hampir setiap hari leg parahh Jangan buat kecewa costumer lah	. kalau bisa masa aktif kartunya di tingkatkan jadi lebih lama terima kasih	Tolong dong masalah jaringan hampir setiap hari leg parahh Jangan buat kecewa costumer lah
Clear Emoji	Tolong dong masalah jaringan hampir setiap hari leg parahh Jangan buat kecewa costumer lah	kalau bisa masa aktif kartunya di tingkatkan jadi lebih lama terima kasih	Bisa cek kuota dgn simpel terbaikkkkkkkkkkk
Replace Repeated Characters	Tolong dong masalah jaringan hampir setiap hari leg parahh Jangan buat kecewa costumer lah	kalau bisa masa aktif kartunya di tingkatkan jadi lebih lama terima kasih	Bisa cek kuota dgn simpel terbaik
Casefolding	tolong dong masalah jaringan hampir setiap hari leg parahh jangan buat kecewa costumer lah	kalau bisa masa aktif kartunya di tingkatkan jadi lebih lama terima kasih	bisa cek kuota dgn simpel terbaik
Tokenizing	'tolong', 'dong', 'masalah', 'jaringan', 'hampir', 'setiap', 'hari', 'leg', 'parahh', 'jangan', 'buat', 'kecewa', 'costumer', 'lah'	'kalau', 'bisa', 'masa', 'aktif', 'kartunya', 'di', 'tingkatkan', 'jadi', 'lebih', 'lama', 'terima', 'kasih'	'bisa', 'cek', 'kuota', 'dgn', 'simpel', 'terbaik'
Formalizing Slang Words	'tolong', 'dong', 'masalah', 'jaringan', 'hampir', 'setiap', 'hari', 'lelet', 'parah', 'jangan', 'buat', 'kecewa', 'konsumen', 'lah'	'kalau', 'bisa', 'masa', 'aktif', 'kartunya', 'di', 'tingkatkan', 'jadi', 'lebih', 'lama', 'terima', 'kasih'	'bisa', 'cek', 'kuota', 'dengan', 'simpel', 'terbaik'
Removing Stopwords	'tolong', 'jaringan', 'lelet', 'parah', 'kecewa', 'konsumen'	'aktif', 'kartunya', 'tingkatkan', 'terima', 'kasih'	'cek', 'kuota', 'simpel', 'terbaik'
Stemming	'tolong', 'jaring', 'lelet', 'parah', 'kecewa', 'konsumen'	'aktif', 'kartu', 'tingkat', 'terima', 'kasih'	'cek', 'kuota', 'simpel', 'baik'

The text preprocessing steps, illustrated in Figure 2, include:

a) Text cleaning: Removing noise such as mentions, hashtags, numbers, and specific characters, replacing them with spaces, and trimming leading or trailing spaces.
b) Emoji removal: Removing emojis from the text.
c) Character reduction: Eliminating repeated characters that appear three times or more.
d) Case folding: Converting all text to lowercase to simplify the text features.
e) Tokenization: Splitting text into tokens (words) using the Natural Language Toolkit (NLTK) library.
f) Slang formalization: Replacing slang terms with formal equivalents using a predefined slang dictionary stored in a .txt file.

g) Stopword removal: Eliminating Indonesian stopwords listed in the NLTK library.
h) Stemming: Converting words to their root forms using the Sastrawi stemmer, with the Swifter library speeding up DataFrame operations in pandas.

2.4 TF_IDF

The MyXL user review dataset consists of text data in string format, which cannot be directly processed by machine learning algorithms. It must first be transformed into numerical or vector representations, a process known as feature extraction.

In this study, TF-IDF (Term Frequency-Inverse Document Frequency) was used for keyword extraction. TF-IDF calculates the weight of a term in a document by considering its frequency and importance across the entire corpus (Febrianti et al., 2023).

a) **TF (Term Frequency):** Measures how often a term appears in a document (tf_{ij} is the frequency of term i in document j).
b) **IDF (Inverse Document Frequency):** Assesses the significance of a term in the corpus, as shown in Equation (1), where N is the total number of documents, and df_i is the number of documents containing term i.
c) **TF-IDF:** A combination of TF and IDF, obtained by multiplying them, as shown in Equation (2).

$$idf_i = \log\left(\frac{N}{df_i}\right) + 1 \tag{1}$$

$$\begin{aligned} w_{ij} &= tf_{ij} * idf_i \\ &= tf_{ij} * (\log\left(\frac{N}{df_i}\right) + 1) \end{aligned} \tag{2}$$

The MyXL user review dataset contains 1,000 records and 1,589 features. Table 4 presents the average TF-IDF values for all features, including “aamiin” (0.2507), “abal” (0.8652), and “yutube” (0.3249).

Table 4 Average TF-IDF Values of All Features

Index	0	1	2	3	…	1,855	1,856	1,857	1,858
Feature	Aamiin	abal	abang	abg	…	yt	yth	yuk	yutube
TF-IDF	0.2507	0.8652	0.1933	0.3206	…	1.1781	0.4507	0.5644	0.3249

2.5 Train-Test Split

The train-test split is a simple, commonly used validation method that divides the dataset into training and testing sets. This division is necessary for training and evaluating the algorithm. The numerical dataset from feature extraction is typically split using ratios of 90:10, 80:20, or 70:30. The split is done randomly while maintaining class proportions in both sets, mirroring the original class distribution. Table 5 shows the training and testing data counts for a 90:10 split.

Table 5 Data Splitting Using a 90:10 Ratio

Review Category	Negative Class	Neutral Class	Positive Class	Total	%
Training Data	552	203	145	900	90
Testing Data	61	23	16	100	10

2.6 Synthetic Minority Oversampling Technique (SMOTE)

SMOTE is a widely used oversampling technique that synthesizes new samples for minority classes to increase their representation before classification. SMOTE works by selecting a sample from the minority class and identifying its k-nearest neighbors. Synthetic samples are then generated along the line segments connecting the original sample to its neighbors, based on the required level of oversampling (Chawla et al., 2002).

SMOTE aims to address class imbalance during model training by balancing the class distributions in the training data, allowing the model to learn the minority class patterns effectively. As such, SMOTE is applied only to the training data, leaving the testing data imbalanced to reflect real-world conditions. Testing on imbalanced data provides valid, objective, and accurate model performance evaluation. Applying SMOTE to the entire dataset before splitting would introduce data leakage, as the model would have access to synthetic patterns from the testing data during training, invalidating the evaluation. Table 6 displays the training data distribution after SMOTE application.

Table 6 Number of Data After Applying SMOTE on Training Data

Review Category	Negative Class	Neutral Class	Positive Class	Total
Before SMOTE	552	203	145	900
	61.3%	22.6%	16.1%	100%
After SMOTE	552	552	552	1,656
	33.3%	33.3%	33.3%	100%

2.7 Hyperparameter Tuning

Hyperparameters are parameters that control the learning process of a machine learning algorithm (Nishat et al., 2022). Hyperparameter tuning involves adjusting these parameters to find the optimal combination for maximizing model performance. This study employed GridSearchCV to train the algorithm and identify the best model by exploring all possible hyperparameter combinations.

GridSearchCV performs cross-validation by dividing the training data into 10 subsets. The model is trained on nine subsets and validated on one, with the process repeated until each subset has served as a validation set. This approach optimizes hyperparameter tuning, identifying the best parameters and achieving the highest cross-validation score. For Logistic Regression, the hyperparameters include C and penalty values, while for SVM, they include C, gamma, and kernel.

2.8 Logistic Regression

Logistic Regression predicts the relationship between independent variables and categorical dependent variables, which may be either nominal or ordinal. For datasets where the dependent variable is nominal with more than two categories, Multinomial Logistic Regression is used (Harahap et al., 2023).

The formula for Multinomial Logistic Regression is expressed in Equation (3). It predicts the probability of a particular observation i belonging to a given class in a dataset. In this formula, π (X_i) represents the estimated probability of the i-th observation, which is calculated based on the independent variables associated with that observation. The equation incorporates β_0, which is the constant or intercept term that adjusts the baseline probability for all observations. Additionally, β_k denotes the coefficient for the k-th independent variable, which measures the influence of that variable on the predicted probability. X_{ik} represents the value of the k-th independent variable for the i-th observation. Together, these components define the relationship between the independent variables and the estimated probabilities, enabling classification into multiple categories.

$$\pi(X_i) = \frac{\exp\,(\beta_0 + \sum_{k=1}^{n} \beta_k X_{ik})}{1 + \exp\,(\beta_0 + \sum_{k=1}^{n} \beta_k X_{ik})} \tag{3}$$

2.9 Support Vector Machine

Support Vector Machine (SVM) is a supervised learning algorithm that uses a linear function hypothesis in high-dimensional spaces, trained through optimization algorithms that apply learning biases derived from statistical theory (Ovirianti et al., 2022). The primary goal of SVM is to create an optimal separating function that can be used for classification tasks.

SVM's basic principle is linear classification, initially limited to handling binary class problems. However, its capabilities have been enhanced through the kernel concept, allowing it to address non-linear problems and multiclass classification. Important parameters in the SVM algorithm include the penalty (L) and the kernel (Atmanegara & Purwa, 2021).

The equations for the **linear** and **polynomial kernels** in Support Vector Machine (SVM) help define the transformation of input data into a higher-dimensional feature space where it becomes easier to separate classes. In these equations, x_i and x_j represent the dot product of two feature vectors, which quantifies their similarity in the feature space. The parameter γ acts as a scale control, commonly set to $1/number\ of\ features$, and influences the flexibility of the decision boundary. The parameter r serves as a bias term, adjusting the output of the kernel function to improve fit. For the polynomial kernel, an additional parameter d specifies the degree of the polynomial, where higher degrees allow for more complex decision boundaries. These components collectively enable SVM to adapt to both linear and non-linear classification tasks by adjusting how input data is mapped and classified in the transformed feature space.

$$K_{linear}(x_{i,}x_j) = x_i^T\ x_j \quad (4)$$

$$K_{polynomial}(x_{i,}x_j) = \left(\gamma x_i^T\ x_j + r\right)^d, \gamma > 0 \quad (4)$$

2.10 Research Scenarios

Table 7 Research Scenarios

Scenario	Training Data (%)	Testing Data (%)	Training Data Balance	Algorithm
1	90	10	Before SMOTE	Logistic Regression
			Before SMOTE	Support Vector Machine
			After SMOTE	Logistic Regression
			After SMOTE	Support Vector Machine
2	80	20	Before SMOTE	Logistic Regression
			Before SMOTE	Support Vector Machine
			After SMOTE	Logistic Regression
			After SMOTE	Support Vector Machine
3	70	30	Before SMOTE	Logistic Regression
			Before SMOTE	Support Vector Machine
			After SMOTE	Logistic Regression
			After SMOTE	Support Vector Machine
4	60	40	Before SMOTE	Logistic Regression
			Before SMOTE	Support Vector Machine
			After SMOTE	Logistic Regression
			After SMOTE	Support Vector Machine
5	50	50	Before SMOTE	Logistic Regression
			Before SMOTE	Support Vector Machine
			After SMOTE	Logistic Regression
			After SMOTE	Support Vector Machine

This study evaluates the performance of Logistic Regression and SVM, employing SMOTE to address data imbalance in sentiment analysis of MyXL reviews. It involves hyperparameter tuning with GridSearchCV across five experimental scenarios, as shown in Table 7.

2.11 Evaluasi

Evaluation measures classification accuracy to assess algorithm performance. Multiclass sentiment classification accuracy is calculated using metrics such as True Positive (TP), False Negative (FN), True Negative (TN), and False Positive (FP) for each class C_i. Overall accuracy is calculated as a macro average of all classes, providing an unbiased performance summary across classes (Grandini et al., 2020). The equations for macro-average metrics are shown in Table 8.

Table 8 Formula for Macro Average Classification Accuracy

Metric	Formula
Accuracy rata-rata	$\frac{\sum_{i=1}^{n} \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i}}{n}$
Precision	$\frac{\sum_{i=1}^{n} \frac{tp_i}{tp_i + fp_i}}{n}$
Recall	$\frac{\sum_{i=1}^{n} \frac{tp_i}{tp_i + fn_i}}{n}$
F score	$2 * \frac{Precision * Recall}{Precision + Recall}$

3. RESULTS AND DISCUSSION

Table 9 Hyperparameter Tuning GridSearchCV Result

	Algorithm	Before SMOTE	After SMOTE
Scenario 1	LR	C = 1,1263157894736844 Penalty = L2	C = 4,0 Penalty = L2
	SVM	C = 1,0 Gamma = 0,2 Kernel = linear	C = 1,0 Gamma = scale Kernel = poly
Scenario 2	LR	C = 1,9473684210526316 Penalty = L2	C= 3,3842105263157896 Penalty = L2
	SVM	C = 1,0 Gamma = scale Kernel = poly	C = 1,0 Gamma = scale Kernel = poly
Scenario 3	LR	C = 0,5105263157894737 Penalty = L2	C = 4,0 Penalty = L1
	SVM	C = 1,0 Gamma = 0,2 Kernel = linear	C = 1,0 Gamma = scale Kernel = poly
Scenario 4	LR	C = 1,3315789473684212 Penalty = L2	C = 3,3842105263157896 Penalty = L2
	SVM	C = 1,0 Gamma = 0,2 Kernel = linear	C = 1,0 Gamma = scale Kernel = poly
Scenario 5	LR	C = 3,3842105263157896 Penalty = L2	C = 4,0 Penalty = L2
	SVM	C = 1,0 Gamma = 0.2 Kernel = linear	C = 1,0 Gamma = scale Kernel = poly

Hyperparameter tuning using GridSearchCV for training Logistic Regression (LR) and Support Vector Machine (SVM) produced the parameter combinations shown in Table 9. These parameters resulted in the best sentiment classification models for user reviews of the MyXL application on Google Play Store. Multiclass classification accuracy for the scenarios is detailed in Table 10.

From Table 10, Scenario 1 achieved the highest accuracy across all models before applying SMOTE: LR (71.00%) and SVM (70.00%). The same SVM accuracy (70.00%) was also observed in Scenario 4. However, accuracy metrics in imbalanced datasets may not fully reflect the model's true performance. For a more comprehensive evaluation, the F1-score, a harmonic mean of precision and recall, must be considered. A higher F1-score indicates better precision-recall balance. In Scenario 1, SVM achieved a higher F1-score (65.10%) than its accuracy (62.75%), highlighting that SVM Scenario 1 outperformed SVM Scenario 4. Furthermore, Scenario 1 yielded the highest accuracy and F1-score for all models after applying SMOTE, with LR accuracy at 72.00%, SVM accuracy at 73.00%, LR F1-score at 63.94%, and SVM F1-score at 66.30%. This makes Scenario 1 the most effective configuration for all algorithms compared to other scenarios. The classification accuracy results for all scenarios are visualized in Figure 3.

Table 10 Macro Average Classification Accuracy of LR and SVM

	Algorithm	Before SMOTE				After SMOTE			
		Acc (%)	Prec (%)	Rec (%)	F1 (%)	Acc (%)	Prec (%)	Rec (%)	F1 (%)
Scenario 1	LR	71,00	62,95	63,83	63,01	72,00	64,12	64,38	63,94
	SVM	70,00	64,44	65,99	65,10	73,00	67,13	65,82	66,30
Scenario 2	LR	69,50	61,49	62,61	61,72	69,50	61,74	62,61	62,00
	SVM	69,00	61,31	63,75	62,34	68,50	60,33	60,36	60,27
Scenario 3	LR	66,67	58,64	59,67	59,12	67,33	58,19	59,61	58,71
	SVM	66,00	60,03	61,05	60,42	68,67	60,69	60,65	60,63
Scenario 4	LR	69,50	60,46	60,58	60,34	69,50	60,54	61,20	60,44
	SVM	70,00	62,98	62,54	62,75	69,25	62,50	60,37	61,30
Scenario 5	LR	67,20	58,26	57,99	58,10	68,20	59,70	58,78	59,13
	SVM	66,00	59,49	58,77	58,98	67,00	60,06	57,21	58,39

In Scenario 1, Logistic Regression accuracy after SMOTE (72.00%) surpassed its pre-SMOTE accuracy (71.00%). Other metrics showed similar improvements: precision increased from 62.95% to 64.12%, recall from 63.83% to 64.38%, and F1-score from 63.01% to 63.94%. The increases—accuracy (+1%), precision (+1.17%), recall (+0.55%), and F1-score (+0.93%)—demonstrate that SMOTE consistently improved Logistic Regression's performance. These improvements indicate that the model became better at identifying patterns in minority classes and achieving balanced classifications after data distribution was equalized using SMOTE.

In Scenario 1, SVM accuracy after SMOTE (73.00%) exceeded its pre-SMOTE accuracy (70.00%). Precision increased from 64.44% to 67.13%, and F1-score rose from 65.10% to 66.30%. Accuracy improved by 3%, precision by 2.69%, and F1-score by 1.2%. These substantial improvements show that SMOTE significantly enhanced SVM's ability to classify data accurately and reduce false positives. However, SVM's recall decreased slightly after SMOTE, from 65.99% to 65.82%, a minor drop of 0.17%. Despite this, the F1-score increase (1.2%) demonstrates that SMOTE effectively improved SVM's overall performance in handling imbalanced data.

In Scenario 1, before SMOTE, SVM had slightly lower accuracy (70.00%) than LR (71.00%), but SVM outperformed LR in precision (64.44%), recall (65.99%), and F1-score (65.10%). This indicates that SVM was better at identifying minority classes before SMOTE. After SMOTE, SVM surpassed LR across all metrics, demonstrating that SMOTE enabled SVM to more effectively identify and classify minority classes. The evaluation results for Scenario 1 are visualized in Figure 4.

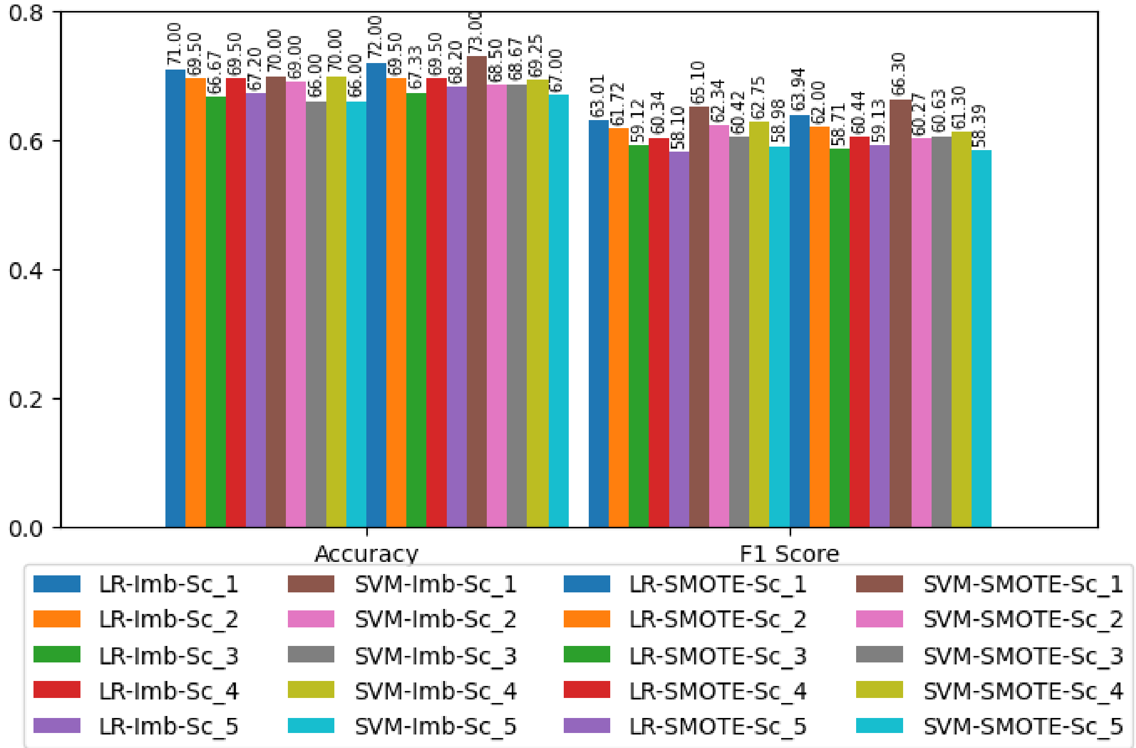

Figure 3 Evaluation Results of All Research Scenarios

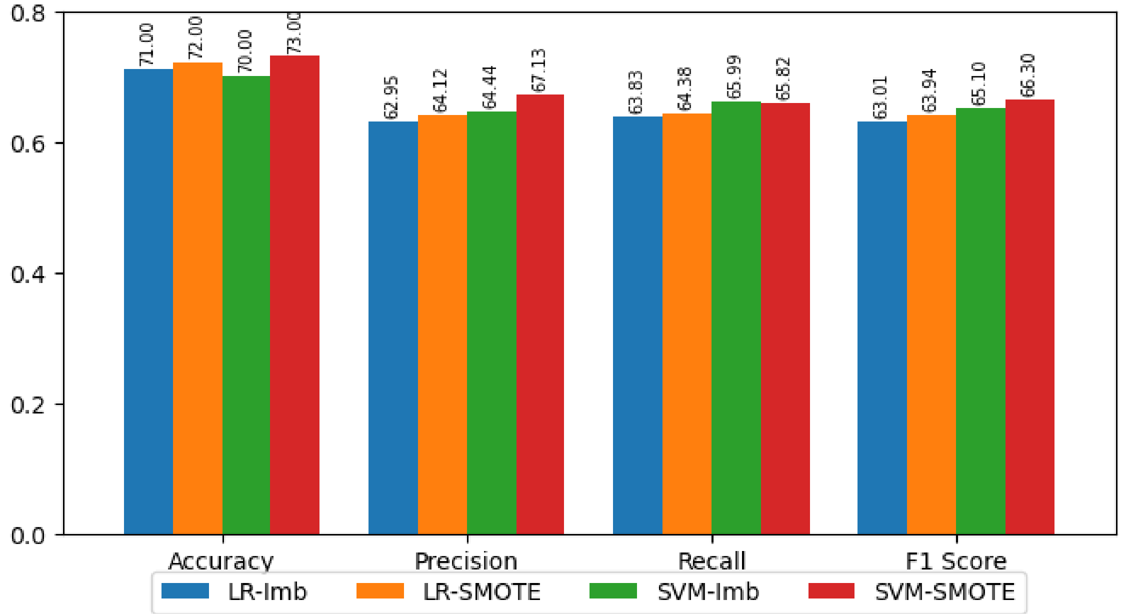

Figure 4 Comparison of Evaluation Results in Scenario 1

Post-SMOTE, the SVM algorithm achieved the highest classification accuracy with a 90% training and 10% testing split, using the parameter combination $C = 1.0$, $\gamma = scale$, and $kernel = poly$. Predictions on the test data produced a confusion matrix shown in Table 11. From this, True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values for each class were calculated and are presented in Table 12. Table 13 displays the overall multiclass classification accuracy, calculated using the macro-average formula from Table 8.

Table 11 Confusion Matrix of SVM After SMOTE in Scenario 1

Actual Class	Predicted Class		
	Negative	Neutral	Positive
Negative	52	7	2
Neutral	10	10	3
Positive	3	2	11

Table 12 Classification Accuracy per Class for SVM After SMOTE in Scenario 1

Class	TP	FP	TN	FN
Negative	52	13	26	9
Neutral	10	9	68	13
Positive	11	5	79	5

Table 13 Macro Average Classification Accuracy of SVM After SMOTE in Scenario 1

Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
73,00	67,13	65,82	66,30

4. CONCLUSIONS

This study compares the performance of Logistic Regression (LR) and Support Vector Machine (SVM) algorithms using SMOTE, TF-IDF, and GridSearchCV for sentiment analysis on the MyXL user review dataset from Google Play Store. The GridSearchCV object from the Scikit-learn library played a crucial role in hyperparameter tuning for both algorithms. The parameter combinations yielding the best models were applied to the algorithms for evaluation. The 90% training and 10% testing data split demonstrated the highest performance for both LR and SVM models, both before and after applying SMOTE. In the context of imbalanced data, SVM outperformed LR in identifying minority classes. Applying SMOTE enhanced the performance of both algorithms, with SVM continuing to show superior capabilities in recognizing and classifying minority classes.

The SVM algorithm achieved the best performance using SMOTE with parameter combinations $C = 1.0$, $\gamma = scale$, and $kernel = poly$, resulting in a classification accuracy of 73.00%, precision of 67.13%, recall of 65.82%, and F1-score of 66.30%.

The lack of significant improvement in evaluations before and after SMOTE might stem from the nature of SMOTE, which performs well in certain cases but does not always produce substantial improvements in all situations. To address this, more in-depth hyperparameter tuning is necessary after applying SMOTE. While GridSearchCV explores all possible hyperparameter combinations, making it computationally intensive, RandomizedSearchCV can serve as an alternative. This method conducts random searches within a large parameter space, offering greater efficiency in terms of time.

REFERENCES

Audiansyah, D. D. (2022, July 5). *Data Ulasan Terlabel*. Kaggle. https://www.kaggle.com/datasets/dimasdiandraa/data-ulasan-terlabel?select=Ulasan+My+XL+1000+Data+Labelled.csv

Atmanegara, E., & Purwa, T. (2021). Hybrid Support Vector Machine and Logistic Regression for Multiclass Classification: A Case Study on Wine Dataset. *Indonesian Journal of Data Science*, *1*(1), 1–7. https://www.researchgate.net/publication/353211298

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

Cheng, M., & Mani, R. (2024, June 24). *Voice of the Consumer Survey 2024: Asia Pacific*. PWC Indonesia. https://www.pwc.com/id/en/pwc-publications/industries-publications/consumer-and-industrial-products-and-services/consumer-survey-2024-asia-pacific.html

Darwis, H., Wanaspati, N., & Anraeni, S. (2023). Support Vector Machine untuk Analisis Sentimen Masyarakat Terhadap Penggunaan Antibiotik di Indonesia. *The Indonesian Journal of Computer Science*, *12*(4), 12. https://doi.org/10.33022/ijcs.v12i4.3320

Hasibuan, E., & Heriyanto, E. A. (2022). Analisis Sentimen pada Ulasan Aplikasi Amazon Shopping di Google Play Store Menggunakan Naive Bayes Classifier. *Jurnal Teknik Dan Science*, *1*(3), 13–24. https://doi.org/10.56127/jts.v1i3.434

Febrianti, F. A. D. P., Hamami, F., & Fa'rifah, R. Y. (2023). Aspect-Based Sentiment Analysis Terhadap Ulasan Aplikasi Flip Menggunakan Pembobotan Term Frequency-Inverse Document Frequency (TF-IDF) Dengan Metode Klasifikasi K-Nearest Neighbors (K-NN). *Jurnal Indonesia : Manajemen Informatika Dan Komunikasi*, *4*(3), 1858–1873. https://doi.org/10.35870/jimik.v4i3.429

Grandini, M., Bagli, E., & Visani, G. (2020). *Metrics for Multi-Class Classification: an Overview*. http://arxiv.org/abs/2008.05756

Haikal, M., Martanto, M., & Hayati, U. (2024). Analisis Sentimen Terhadap Penggunaan Aplikasi Game Online PUBG Mobile Menggunakan Algoritma Naive Bayes. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *7*(6), 3275–3281. https://doi.org/10.36040/jati.v7i6.8174

Harahap, F. H., Sutarman, S., Darnius, O., & Sembiring, P. (2023). Klasifikasi Menggunakan Model Regresi Logistik Multinomial dan Regresi Logistik Multinomial Komponen Utama. *IJM: Indonesian Journal of Multidisciplinary*, *1*(2), 632–642. https://journal.csspublishing.com/index.php/ijm/article/view/183

Huda, M. N., Fauzan, D. A., Pamungkas, M. R. S. P., Ratnadewi, N. S., & Vahendra, A. A. (2023). Optimalisasi Model Klasifikasi Sentimen Netizen Terhadap Merek Tas Luar Negeri. *Jurnal KomtekInfo*, 21–28. https://doi.org/10.35134/komtekinfo.v10i1.360

Khushi, M., Shaukat, K., Alam, T. M., Hameed, I. A., Uddin, S., Luo, S., Yang, X., & Reyes, M. C. (2021). A Comparative Performance Analysis of Data Resampling Methods on Imbalance Medical Data. *IEEE Access*, *9*, 109960–109975. https://doi.org/10.1109/ACCESS.2021.3102399

Nishat, M. M., Faisal, F., Ratul, I. J., Al-Monsur, A., Ar-Rafi, A. M., Nasrullah, S. M., Reza, M. T., & Khan, M. R. H. (2022). A Comprehensive Investigation of the Performances of Different Machine Learning Classifiers with SMOTE-ENN Oversampling Technique and Hyperparameter Optimization for Imbalanced Heart Failure Dataset. *Scientific Programming*, *2022*, 1–17. https://doi.org/10.1155/2022/3649406

Ovirianti, N. H., Zarlis, M., & Mawengkang, H. (2022). Support Vector Machine Using A Classification Algorithm. *Jurnal Dan Penelitian Teknik Informatika*, *6*(3). https://doi.org/10.33395/sinkron.v7i3

Ramadhani, B., & Suryono, R. R. (2024). Komparasi Algoritma Naïve Bayes dan Logistic Regression Untuk Analisis Sentimen Metaverse. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, *8*(2), 714. https://doi.org/10.30865/mib.v8i2.7458

Syah, F., Fajrin, H., Afif, A. N., Saeputra, M. R., Mirranty, D., & Saputra, D. D. (2023). Analisa Sentimen Terhadap Twitter IndihomeCare Menggunakan Perbandingan Algoritma Smote, Support Vector Machine, AdaBoost dan Particle Swarm Optimization. *Jurnal JTIK (Jurnal Teknologi Informasi Dan Komunikasi)*, *7*(1), 53–58. https://doi.org/10.35870/jtik.v7i1.686