

Evaluation of TF-IDF and Word2Vec Representations Using Support Vector Machine for Multiclass Product Category Classification of Tokopedia Reviews

Anggi Dwi Septian¹, Primandani Arsi², Pungkas Subarkah³

^{1,2,3}Department of Informatics, Faculty of Computer Science, Universitas Amikom Purwokerto, Purwokerto, Indonesia

Received:

June 6, 2026

Revised:

July 21, 2026

Accepted:

September 3, 2026

Published:

October 24, 2026

Corresponding Author:

Author Name*:

Primandani Arsi

Email*:

ukhti.prima@amikompurwokerto.ac.id

DOI:

10.63158/journalisi.v8i5.1820

© 2026 Journal of Information Systems and Informatics. This open access article is distributed under a (CC-BY License)



Abstract. As electronic commerce continues to expand, customer reviews have become an important source for automated text classification. This research evaluates and compares TF-IDF and Word2Vec representations for multiclass product category classification with Support Vector Machine (SVM). The experiment used the Tokopedia Product Reviews 2025 dataset containing 65,543 reviews across six product categories. After text preprocessing and keyword-based text refinement, 11,239 relevant reviews were retained. The dataset was divided into 80% training sets and 20% hold-out test sets. The reviews were transformed using TF-IDF and Word2Vec, while SMOTE was applied to the training data to handle the imbalance among product categories. Classification performance was assessed through five-fold cross-validation and a hold-out test set based on accuracy, macro precision, macro recall, and macro F1-score. Five-fold cross-validation showed that TF-IDF achieved mean accuracy, macro precision, macro recall, and macro F1-score of 95.95%, 93.13%, 92.78%, and 92.93%, respectively, compared with 90.80%, 84.58%, 86.02%, and 85.19% for Word2Vec. On the hold-out test set, TF-IDF achieved 95.42% accuracy and 91.88% macro F1-score, while Word2Vec achieved 91.01% and 84.91%, respectively. The Wilcoxon signed-rank test yielded a p-value of 0.0625, indicating no statistically significant difference at the 5% significance level. Nevertheless, TF-IDF consistently achieved higher descriptive performance than Word2Vec.

Keywords: Text Classification, TF-IDF, Word2Vec, Support Vector Machine, Tokopedia Reviews, Product Category Classification.

1. INTRODUCTION

The continuous advancement of digital technology has contributed to the rapid expansion of electronic commerce (e-commerce), leading to the generation of extensive product review data. Customer reviews are increasingly recognized as a valuable source of information because they reflect users' experiences with purchased products and may influence the purchasing decisions of other consumers [1]. The increasing availability of review data on e-commerce platforms creates opportunities to analyze consumer perceptions and support data-driven decision-making processes [2]. However, manual analysis becomes less practical as the volume of textual data increases, creating the need for automated computational approaches to extract useful information from customer reviews [3]. Product category classification from review text is practically important because it enables customer reviews to be organized and analyzed according to product categories. This facilitates category-specific analysis of customer feedback and supports marketplace analytics and data-driven decision-making. Prior research has shown the potential of machine learning methods for marketplace product classification, highlighting its potential for organizing large-scale e-commerce data and supporting intelligent marketplace services [4].

Text classification is one of the commonly adopted approaches for processing review data through Natural Language Processing (NLP) and machine learning methods [5]. This process seeks to categorize textual documents into predefined classes based on their underlying characteristics [6]. In text classification, the text representation stage plays an essential role because it determines how textual information is transformed into numerical features that can be utilized by machine learning algorithms [7]. Among the commonly applied representation methods are Term Frequency-Inverse Document Frequency (TF-IDF) and Word2Vec. TF-IDF is a weighting method used to estimate the importance of terms according to their occurrence within documents [8]. In contrast, Word2Vec generates continuous vector representations to capture semantic relationships among words [9].

The resulting text representations are subsequently processed using classification algorithms. Support Vector Machine (SVM) is commonly employed for text classification because it effectively handles high-dimensional feature spaces while maintaining robust

predictive performance [10]. Nevertheless, classification effectiveness may decline when the dataset suffers from class imbalance, where certain classes contain substantially more instances than others, causing the model to favor majority classes during prediction [11]. To mitigate this issue, the Synthetic Minority Oversampling Technique (SMOTE) can be used to generate synthetic minority samples, resulting in a more balanced class distribution [12].

Earlier research has investigated the comparative performance of TF-IDF and Word2Vec in text classification tasks. Efrizoni et al. reported that TF-IDF achieved higher classification performance than Word2Vec when combined with traditional machine learning algorithms. In contrast, Rifaldy et al. demonstrated that Word2Vec outperformed TF-IDF when applied to sentiment classification using data collected from online investment platforms and classified using SVM [13]. Other studies conducted by Zhou et al. and Al Tawil et al. demonstrated that classification outcomes are influenced by both text representation methods and classification algorithms [14], [15]. Furthermore, Sumantiawan et al. showed that applying SMOTE-based balancing techniques can improve classification performance on imbalanced datasets [16].

Previous research has also explored multiclass marketplace product classification using machine learning techniques. Aditama et al. classified products from Tokopedia and Shopee into predefined commodity categories and found that machine learning models can effectively organize marketplace products based on product information [4]. However, their study relied on product attributes rather than customer review texts. In contrast, the present study focuses on multiclass product category classification using customer review texts, enabling category-specific analysis based on user-generated content.

Although numerous studies have investigated the comparative performance of TF-IDF and Word2Vec, the reported findings remain inconclusive due to differences in dataset characteristics, preprocessing strategies, text representation approaches, and classification methods. Moreover, studies focusing on multiclass product category classification using Indonesian-language Tokopedia reviews while addressing class imbalance remain limited. Most previous studies have concentrated on sentiment analysis or employed different research objects, leaving the comparative performance of TF-IDF

and Word2Vec for multiclass product category classification insufficiently examined. Therefore, this study evaluates TF-IDF and Word2Vec for multiclass product category classification using Indonesian-language Tokopedia reviews while addressing class imbalance through SMOTE. By evaluating both text representation methods under the same experimental settings, this study provides empirical evidence regarding their effectiveness for multiclass product category classification.

Considering the limitations identified in previous studies, this study seeks to determine which text representation method, TF-IDF or Word2Vec, provides better classification performance for multiclass Tokopedia product category classification using Support Vector Machine (SVM) classifier. To answer this research question, TF-IDF and Word2Vec were evaluated using the SVM classifier. To handle class imbalance, SMOTE was applied to the training data. Classification performance was then assessed through 5-fold cross-validation using accuracy, macro precision, macro recall, and macro F1-score.

2. METHODS

2.1. Literature Review

The study began by examining previous studies related to text classification using Natural Language Processing and machine learning. The review focused on several aspects relevant to this study, including TF-IDF and Word2Vec for text representation, SVM for classification, and SMOTE as a technique for handling class imbalance. The reviewed literature was collected from journal articles and scientific publications discussing text classification, feature representation, and model evaluation in text-based review data. This stage was conducted to understand the implementation of methods in prior studies, identify existing research limitations, and establish the research workflow and methodological procedures applied in this study. In accordance with the results of the literature review, the research stages were systematically designed to support the implementation of the proposed methods.

2.2. Data Collection

This study utilized the Tokopedia Product Reviews 2025 dataset collected from the Kaggle platform [17]. The dataset initially contained 65,543 Tokopedia product reviews in

CSV format. The reviews were written in Indonesian and contain both formal and informal language variations, including abbreviations, slang expressions, and emojis.

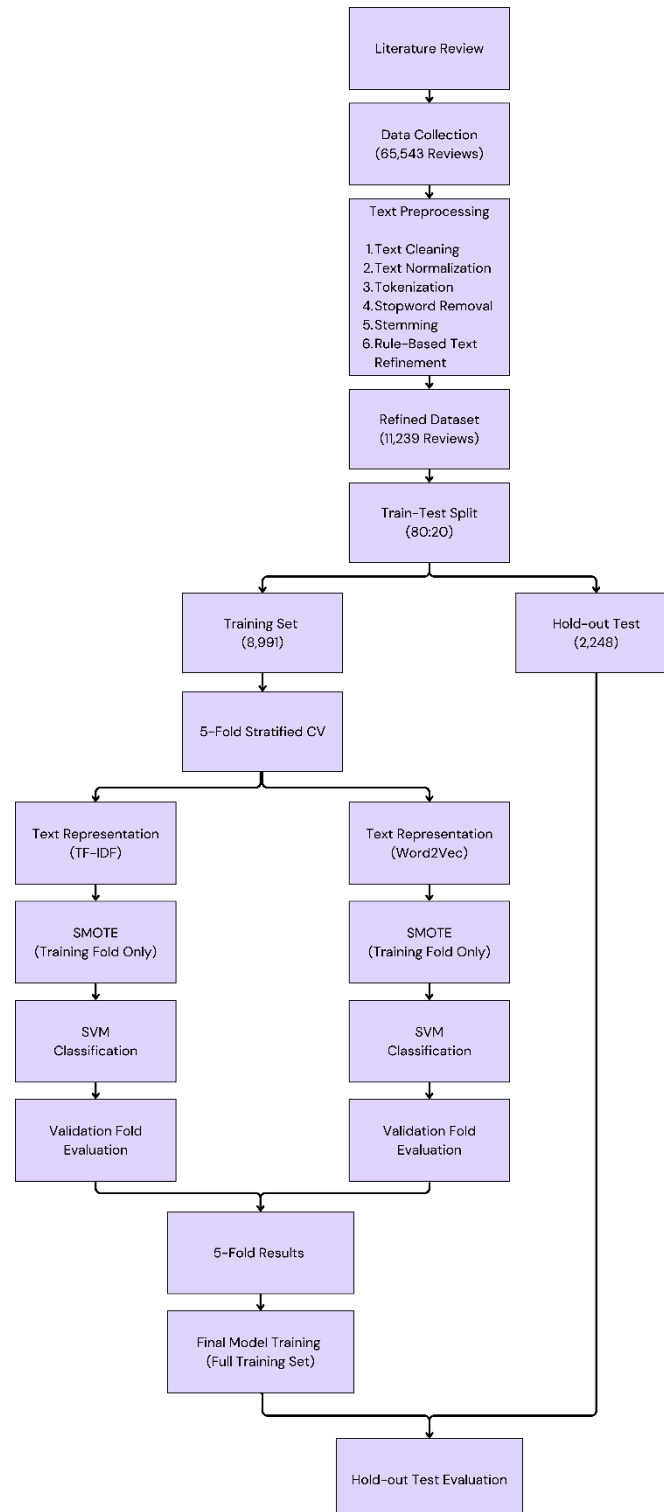


Figure 1. Research Workflow

The dataset covers six product categories: Electronics, Mobile Phones & Tablets, Health, Food & Beverages, Sports, and Hardware & Tools. In this study, the `review_text` attribute was used as the input feature, while `product_category` served as the classification label. Other attributes, including `rating`, `product_name`, and `sold_count`, were excluded from the modeling process. The distribution of review data across product categories is presented in Figure 2. Based on the distribution, the number of reviews across categories was not evenly distributed, indicating an imbalance class distribution in the dataset.

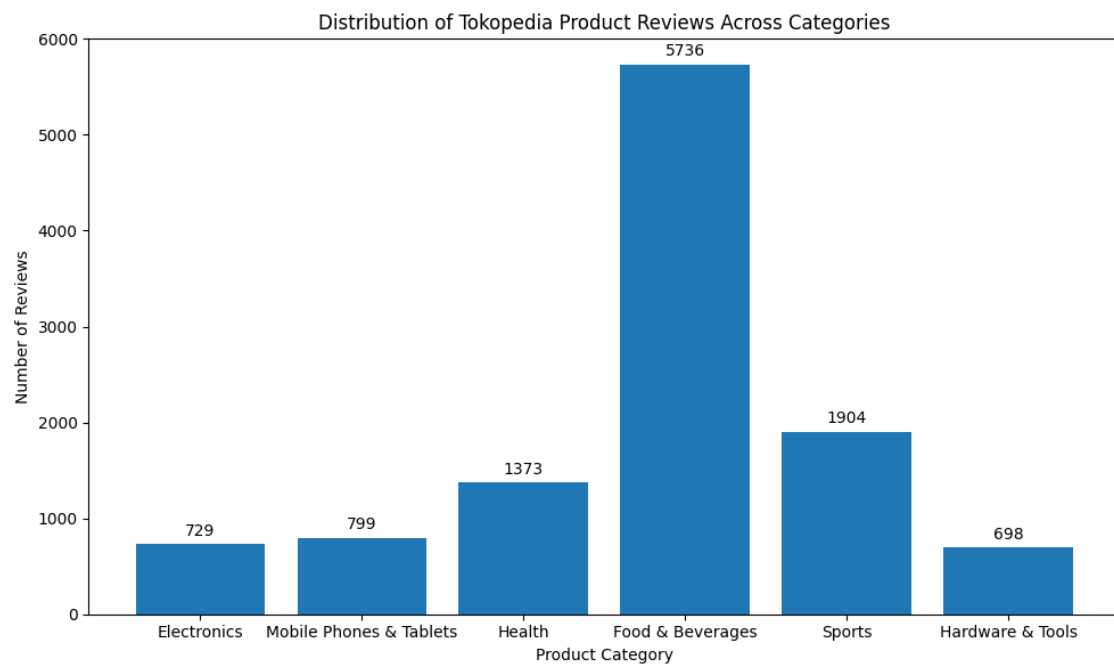


Figure 2. Distribution of Review Data Across Product Categories

Before model development, the dataset underwent a text preprocessing stage to improve data quality and retain reviews suitable for product category classification. This process included text cleaning, normalization, tokenization, stopword removal, stemming, and rule-based text refinement. During the refinement stage, reviews that did not satisfy the predefined filtering criteria were removed, resulting in a reduction of the dataset from 65,543 to 11,239 reviews. The resulting dataset was then used for text representation, model training, and performance evaluation.

2.3. Text Preprocessing

Prior to feature representation and model classification, a text preprocessing stage was implemented to refine and standardize the review data [18]. This process aimed to reduce

noise and improve the quality of textual features used for model training. To prepare the review dataset for subsequent analysis, the following preprocessing stages were carried out:

- 1) Text Cleaning, which removed URLs, emojis, punctuation, numbers, special symbols, and other non-informative characters using regular expressions (regex).
- 2) Text Normalization, which included converting all text into lowercase, correcting repeated characters, standardizing non-standard words into their formal forms, and refining spacing inconsistencies.
- 3) Tokenization, which segmented each review into individual word tokens using whitespace-based tokenization [19].
- 4) Stopword Removal, which removed common Indonesian stopwords using the Sastrawi StopWordRemoverFactory. In addition, a custom stopwords list was incorporated to eliminate domain-specific and non-informative terms, including common slang, filler words, social expressions, and general evaluative words that contributed minimally to product category classification.
- 5) Stemming was performed using the Sastrawi Stemmer library to reduce words to their root forms [20].
- 6) Text Refinement, which was performed automatically using a rule-based approach implemented in Python. This stage aimed to retain informative reviews relevant to product category classification. Generic reviews containing only short expressions with insufficient category-related information were removed. Subsequently, a predefined list of category-specific keywords was manually developed based on frequently occurring terms identified by the authors through exploratory analysis of the Tokopedia review dataset. Each review was retained only if it contained at least one keyword corresponding to its original product category. The keyword list was used solely for preprocessing to retain informative reviews and was not used to assign or modify the original product category labels. As a result, the dataset was reduced from 65,543 to 11,239 reviews, which were subsequently used for feature extraction, model training, and evaluation.

Table 1. Category Keywords Used in Rule-Based Text Refinement

Product Category	Keywords (Indonesia)
Electronics	<i>kabel, suara, printer, listrik, mesin</i>
Mobile Phones & Tablets	<i>handphone, smartphone, garansi, performa, cas, ipad</i>
Health	<i>masker, wangi, minyak, sehat, gizi, diet, protein, obat</i>
Food & Beverages	<i>telur, rasa, enak, roti, sayur, segar, daging, manis</i>
Sports	<i>bola, pingpong, billiard, olahraga, raket, yoga</i>
Hardware & Tools	<i>kayu, alat, pipa, plastik, material</i>

Table 1 presents the category-specific keywords used for rule-based text refinement. These keywords were applied only during preprocessing to retain informative reviews and did not determine or modify the original product category labels. Examples of the preprocessing results are presented in Table 2.

Table 2. Example of Text Preprocessing on Customer Reviews (Indonesia)

Preprocessing	Sample Review 1	Sample Review 2
Original Review Data	<i>Puaass bgt! Semuanya segar, packing nya rapi n cpt bgt sampe nya</i>	<i>aman tentram..... pokoknya aman dah sampenya. packing okay.</i>
Text Cleaning	<i>puaass bgt semuanya segar packing nya rapi n cpt bgt sampe nya</i>	<i>aman tentram pokoknya aman dah sampenya packing okay</i>
Text Normalization	<i>puas banget semuanya segar kemasan nya rapi dan cepat banget sampai nya</i>	<i>aman tentram pokoknya aman sudah sampai packing oke</i>
Tokenization	<i>[puas, banget, semuanya, segar, kemasan, nya, rapi, dan, cepat, banget, sampai, nya]</i>	<i>[aman, tentram, pokoknya, aman, sudah, sampai, packing, oke]</i>
Stopword Removal	<i>[puas, segar, kemasan, rapi, cepat]</i>	<i>[aman, tentram, aman, sampai, packing, oke]</i>
Stemming	<i>[puas, segar, kemasan, rapi, cepat]</i>	<i>[aman, tentram, aman, sampai, packing, oke]</i>
Text Refinement	<i>[puas, segar, kemasan, rapi, cepat]</i>	-

Based on Table 2, each text preprocessing stage resulted in modifications to the review data to improve data quality prior to the classification process. The text cleaning, normalization, tokenization, stopword removal, and stemming stages were performed to clean and standardize the textual content, whereas the text refinement stage was applied to retain reviews that were relevant to the product categories. Reviews that did not meet the filtering criteria were removed from the dataset, resulting in more representative data for the subsequent text representation and classification processes.

2.4. Data Splitting

The refined dataset was partitioned using an 80:20 stratified split, following a data-splitting approach applied in previous Tokopedia review classification research [21]. This resulted in 8,991 reviews for training and 2,248 reviews for hold-out testing. The training data were used for model development and five-fold cross-validation, whereas the hold-out test data were kept separate for the final evaluation. Stratification was employed to maintain the class distribution in both subsets. The resulting data partition is presented in Table 3.

Table 3. Dataset Split

Category	Training Data	Hold-out Test Data
Electronics	583	146
Mobile Phones & Tablets	639	160
Health	1,098	275
Food & Beverages	4,589	1,147
Sports	1,523	381
Hardware & Tools	559	139
Total	8,991	2,248

This partitioning scheme was applied consistently to both TF-IDF and Word2Vec representations. For five-fold cross-validation, the training data were further divided into training and validation folds. TF-IDF was fitted and Word2Vec was trained only on the corresponding training portion of each fold, while the validation portion was used exclusively for evaluation. The hold-out test set was excluded from cross-validation, feature learning, data balancing, and model training, and was used only for the final assessment of the trained models.

2.5. Text Representation

Following data splitting, the textual data in each training fold were transformed into numerical representations for classification. In this study, two text representation methods were employed Term Frequency–Inverse Document Frequency (TF-IDF) and Word2Vec.

2.5.1. TF-IDF

The TF-IDF method was applied to calculate the weight of each term based on the number of occurrences within a document and its relative importance throughout the entire document collection [22]. In this study, TF-IDF was implemented using the Scikit-learn library with a maximum of 5,000 features and a unigram setting. During five-fold cross-validation, the TF-IDF vectorizer was fitted exclusively on each training fold and subsequently used to transform the corresponding validation fold. This procedure prevented information from the validation fold from being used during feature construction.

1) Term Frequency (TF)

Term Frequency (TF) represents how often a particular term occurs in a document. Its value is calculated as the number of occurrences of the term divided by the total number of terms contained in that document, as presented in Equation (1) [23].

$$TF(t, d) = \frac{\text{number of occurrences of a term in a document}}{\text{total number of terms in the document}} \quad (1)$$

where $TF(t, d)$ denotes the term frequency of term (t) in document (d). The numerator represents the number of occurrences of term (t) in document (d), while the denominator indicates the total number of terms in document (d).

2) Inverse Document Frequency (IDF)

Inverse Document Frequency (IDF) determines the importance of a term based on its distribution across the document collection. Terms appearing in fewer documents receive higher weights, whereas terms occurring in many documents receive lower weights. This study adopted the smoothed IDF formulation provided by Scikit-learn's `TfidfVectorizer`, as shown in Equation (2) [24].

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

where $IDF(t)$ represents the inverse document frequency of term (t), (N) indicates the total number of documents in the training corpus, and $df(t)$ refers to the number of documents containing term (t).

3) TF-IDF Weight

The TF-IDF value is calculated from the product of the term frequency (TF) and the corresponding inverse document frequency (IDF), as shown in Equation (3) [23].

$$TF - IDF(t, d) = TF(t, d) \times IDF(t) \quad (3)$$

Where $TF-IDF(t, d)$ represents the weight assigned to term (t) in document (d), $TF(t, d)$ denotes the term frequency of (t) in (d), and $IDF(t)$ represents the inverse document frequency of term (t). The resulting TF-IDF values are used as numerical features for the SVM classifier.

2.5.2. Word2Vec

Word2Vec method was used to generate vector representations capable of capturing semantic relationships among words [25]. The implementation used the Gensim library with the Skip-gram architecture. During five-fold cross-validation, Word2Vec was trained exclusively on the training portion of each fold to avoid incorporating information from the corresponding validation data. Each review was then converted into a fixed-length document representation by averaging the vectors of all words contained in the review [26]. The mean document vector is calculated using Equation (4).

$$\vec{d} = \frac{1}{n} \sum_{i=1}^n \vec{w}_i \quad (4)$$

where $\vec{d}$ denotes the document vector representing a review, (n) denotes the number of words in the review that have corresponding Word2Vec vectors, and $\vec{w}_i$ denotes the vector representation of the (i)-th word in the review.

The Word2Vec model was configured with a vector size of 100, a window size of 5, and a minimum word frequency of 2. The model used negative sampling with 5 negative samples and was trained for 10 epochs using 4 worker processes. A random seed of 42 was used to ensure reproducibility. The resulting word vectors were averaged to obtain

a fixed-length representation for each review, which was subsequently used as the input to the SVM classifier.

During five-fold cross-validation, TF-IDF was fitted and Word2Vec was trained exclusively on the training fold, while the corresponding validation fold was transformed using the representation model obtained from the training fold. The parameters used for each text representation method are presented in Table 4.

Table 4. Text Representation Parameters

Parameter	Text Representation	
	TF-IDF	Word2Vec
Library	scikit-learn	Gensim
Max Features	5,000	–
N-gram Range	(1,1)	–
Vector Size	–	100
Window Size	–	5
Min Count	–	2
Training Method	–	Skip-gram
Negative Sampling	–	5
Epochs	–	10
Workers	–	4
Random Seed	–	42
Training Data	Training Fold	Training Fold

2.6. Data Balancing (SMOTE)

For each training fold, the text data were first represented using either TF-IDF or Word2Vec. SMOTE was then applied to the training portion to address the imbalance between classes. The validation portion was kept unchanged, while the hold-out test set was excluded from the oversampling procedure. This arrangement ensured that synthetic samples were generated only from training data and that the original class distributions of the validation and test sets were preserved [12].

SMOTE generated synthetic samples for the minority classes based on their nearest-neighbor relationships, increasing their representation during model training. In this

study, SMOTE was implemented using `sampling_strategy='auto'`, `k_neighbors=5`, and `random_state=42`. The oversampling process was performed separately within each training fold after text representation. Figure 3 presents the distribution of samples across classes before and after applying SMOTE.

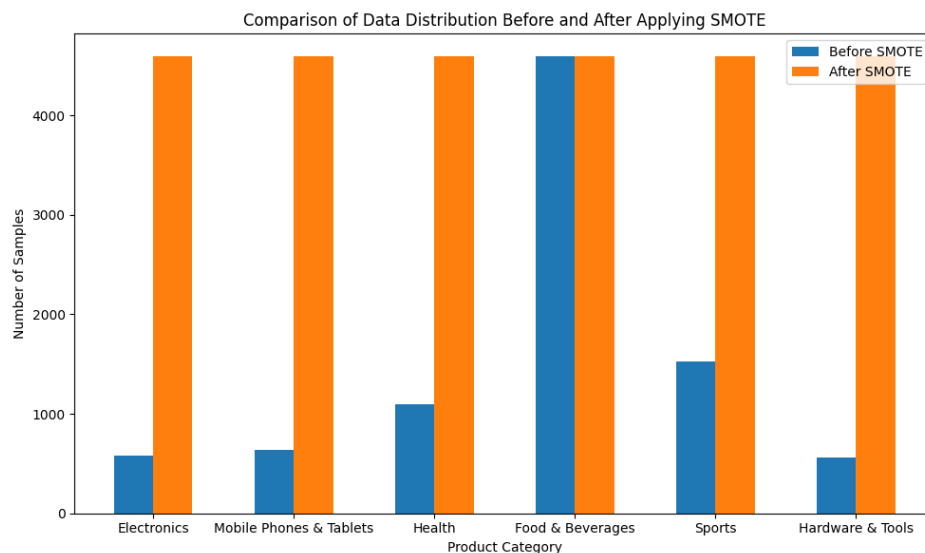


Figure 3. Data Distributions Before and After SMOTE

Based on the class distribution before and after SMOTE, the minority classes received additional synthetic samples to achieve a more balanced class distribution within the training data. The resulting balanced training folds were subsequently used to train the SVM classifier, while the validation folds and hold-out test set remained unchanged.

2.7. Model Classification

The Support Vector Machine (SVM) algorithm was employed to classify Tokopedia customer reviews into predefined product categories using numerical features generated during the text representation stage. SVM has been widely applied to text classification tasks, including Indonesian-language text classification [27]. Within each iteration of the five-fold stratified cross-validation, the classifier was trained using the balanced training fold obtained after applying SMOTE. The model development process was conducted separately for the TF-IDF and Word2Vec representations using the same classifier configuration to provide a consistent basis for comparison.

In this study, the LinearSVC classifier was selected for both text representations, as linear SVM has demonstrated strong performance in Indonesian text classification with high-dimensional TF-IDF features [28]. The regularization parameter (C) was set to 1.0, class_weight was set to None, and random_state was set to 42. As LinearSVC uses a linear decision function, the kernel and gamma parameters were not applicable. No hyperparameter optimization was performed, and the models were trained using the specified parameter settings.

2.8. Validation and Evaluation

The classification models were validated and evaluated to assess their ability to assign text reviews to the appropriate categories. Model performance was measured using accuracy, macro precision, macro recall, and macro F1-score. Since this study addresses multiclass classification, macro-averaged metrics were used to give equal consideration to all product categories [29].

To assess model robustness and the consistency of classification performance, five-fold stratified cross-validation was performed exclusively on the training set [30]. The 8,991 training reviews were divided into five stratified folds, with four folds used for training and the remaining fold for validation. This process was repeated five times so that each fold served once as the validation set. In each iteration, TF-IDF was fitted or Word2Vec was trained only on the training portion, followed by SMOTE applied exclusively to that portion. The validation data were not subjected to oversampling. After each iteration, the trained model was evaluated on the corresponding validation fold. The cross-validation results were summarized using the mean and standard deviation of the performance metrics.

After five-fold cross-validation was completed, the final model was trained using the complete training set of 8,991 reviews. The separate hold-out test set containing 2,248 reviews was not used during cross-validation, text representation learning, SMOTE, or model training. It was used exclusively for the final evaluation of the trained model on previously unseen data. The results obtained from the TF-IDF and Word2Vec models were then compared based on both cross-validation and hold-out test performance to determine the more effective text representation method for Tokopedia product review category classification.

3. RESULTS AND DISCUSSION

3.1. Dataset Characteristics

In addition to the category distribution and data balancing process described in the Methods section, an analysis of review length was conducted to better understand the characteristics of the textual data used in this study. Figure 4 illustrates the distribution of review lengths measured by the number of tokens.

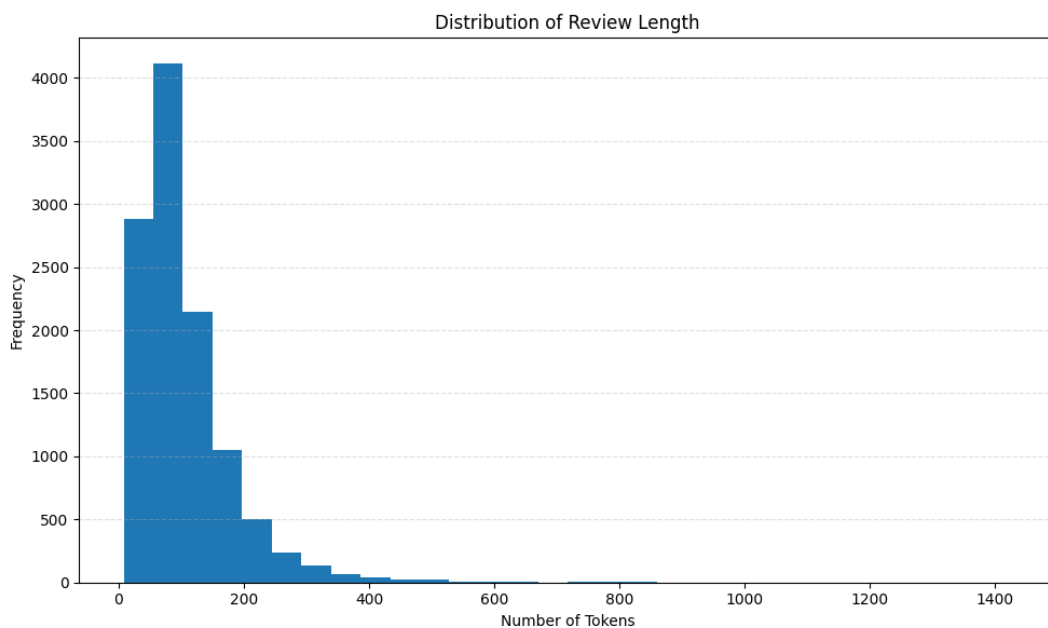


Figure 4. Distribution of Review Length

Based on the distribution, most reviews contained a relatively small number of tokens, while only a limited portion of reviews exhibited substantially longer text lengths, resulting in a skewed distribution pattern. To support this analysis of dataset characteristics, descriptive statistics of review length are presented in Table 5.

Table 5. Review Length Statistics

Statistic	Value
Number of Reviews	11,239
Mean	8.30
Standard Deviation	7.20
Minimum	1
Median	6
Maximum	114

Table 5 presents the descriptive statistics of review length measured by the number of tokens after the text preprocessing stage. The results indicate that the average review length was 8.30 tokens, with a median value of 6 tokens. These findings suggest that most reviews were relatively short, although a small proportion contained substantially more tokens. This characteristic was used as one of the considerations in interpreting the performance of the text representation methods during the classification process.

3.2. Text Representation Results

Following preprocessing, the textual data were converted into numerical feature representations to enable processing by the classification model. In this study, two text representation methods were employed: TF-IDF and Word2Vec. To provide an overview of the representation process, a simulation was conducted using several sample reviews before applying the methods to the entire dataset. The review examples used in the representation simulation are presented in Table 6.

Table 6. Example Reviews for Text Representation Simulation

Document	Review (Indonesia)	Category
D1	<i>bahan tebal mudah cuci</i>	Sports
D2	<i>mesin pel hasil bersih operasi mudah</i>	Electronics
D3	<i>salmon enak potong pas tebal mudah matang</i>	Food & Beverages

3.2.1. TF-IDF

Based on the review examples shown in Table 6, a simulation was performed using the TF-IDF method to demonstrate the generating of numerical representations for each document. The vector construction stages of the TF-IDF method are presented as follows:

1) Vocabulary Generation

The first stage of TF-IDF weighting involves creating a vocabulary, which consisting of unique terms appearing across the document collection. Based on the three sample documents presented in Table 6, the resulting vocabulary is as follows: {bahan, tebal, mudah, cuci, mesin, pel, hasil, bersih, operasi, salmon, enak, potong, pas, matang}.

2) Term Frequency (TF) Calculation

The TF values were calculated using the formula presented in Equation (1). As an example, document D1 consists of four terms, and the term *bahan* appears once. Therefore, the TF value is calculated as follows:

$$TF(bahan, D1) = \frac{1}{4} = 0.25$$

The same calculation procedure was applied to all terms in each document. The resulting TF values are shown in Table 7.

Table 7. TF Calculation Results

Term	Term Frequency (TF)		
	D1	D2	D3
bahan	0.25	0	0
tebal	0.25	0	0
mudah	0.25	0.167	0.143
cuci	0.25	0	0
mesin	0	0.167	0
pel	0	0.167	0
hasil	0	0.167	0
bersih	0	0.167	0
operasi	0	0.167	0
salmon	0	0	0.143
enak	0	0	0.143
potong	0	0	0.143
pas	0	0	0.143
matang	0	0	0.143

3) Inverse Document Frequency (IDF) Calculation

The IDF values were calculated using the formula presented in Equation (2). As an example, the term *tebal* appears in two of the three documents, resulting in a DF value of 2. Therefore, the IDF value is calculated as follows:

$$IDF(tebal) = \log\left(\frac{1+3}{1+2}\right) + 1 \approx 1.693$$

The same calculation procedure was applied to all terms to obtain the DF and IDF values for each term. The resulting DF and IDF values are presented in Table 8.

Table 8. DF and IDF Calculation Results

Term (Indonesia)	DF	IDF
<i>bahan</i>	1	1.693
<i>tebal</i>	2	1.288
<i>mudah</i>	3	1.000
<i>cuci</i>	1	1.693
<i>mesin</i>	1	1.693
<i>pel</i>	1	1.693
<i>hasil</i>	1	1.693
<i>bersih</i>	1	1.693
<i>operasi</i>	1	1.693
<i>salmon</i>	1	1.693
<i>enak</i>	1	1.693
<i>potong</i>	1	1.693
<i>pas</i>	1	1.693
<i>matang</i>	1	1.693

4) Term Frequency Inverse Document Frequency (TF-IDF) Calculation

The TF-IDF weights were calculated using the formula presented in Equation (3). As an example, for the term *bahan* in document D1, the TF value is 0.25 and the IDF value is 1.693. Therefore, the resulting TF-IDF value is calculated as follows:

$$TF - IDF(bahan, D1) = 0.25 \times 1,693 = 0.423$$

The same calculation procedure was applied to all terms in each document to obtain TF-IDF-based numerical representations. The resulting TF-IDF weights are presented in Table 9.

Table 9. TF-IDF Calculation Results

Term (Indonesia)	TF-IDF		
	D1	D2	D3
<i>bahan</i>	0.423	0.000	0.000
<i>tebal</i>	0.322	0.000	0.184
<i>mudah</i>	0.250	0.167	0.143
<i>cuci</i>	0.423	0.000	0.000
<i>mesin</i>	0.000	0.282	0.000
<i>pel</i>	0.000	0.282	0.000
<i>hasil</i>	0.000	0.282	0.000
<i>bersih</i>	0.000	0.282	0.000
<i>operasi</i>	0.000	0.282	0.000
<i>salmon</i>	0.000	0.000	0.242
<i>enak</i>	0.000	0.000	0.242
<i>potong</i>	0.000	0.000	0.242
<i>pas</i>	0.000	0.000	0.242
<i>matang</i>	0.000	0.000	0.242

Based on the calculation results shown in Table 9, the TF-IDF weights indicate differences in the contribution of each term in representing the documents. Terms appearing in fewer documents tend to receive higher weights, whereas terms occurring more frequently across documents receive lower weights. These weighting results form TF-IDF-based numerical representations that are subsequently used as input features for the classification process.

3.2.2. Word2Vec

Based on the review examples presented in Table 6, a representation simulation was conducted using Word2Vec to illustrate the process of generating document vectors. In this method, each word is mapped to a numerical vector and subsequently combined to obtain a document representation. In this study, document vectors were generated using the mean vector approach. The stages of representation generation using Word2Vec are described as follows:

1) Word Embedding Generation

Each word in the documents was represented by a three-dimensional vector to illustrate the process of generating representations using Word2Vec. The use of illustrative vectors was intended to simplify the calculation process and improve interpretability without altering the fundamental concept of the Word2Vec method. The illustrative word vectors are presented in Table 10.

Table 10. Illustrative Word2Vec Vectors

Term (Indonesia)	Document Vectors		
	D1	D2	D3
bahan	[0.2, 0.4, 0.3]	0	0
tebal	[0.5, 0.2, 0.6]	0	[0.5, 0.2, 0.6]
mudah	[0.3, 0.5, 0.4]	[0.3, 0.5, 0.4]	[0.3, 0.5, 0.4]
cuci	[0.4, 0.3, 0.2]	0	0
mesin	0	[0.6, 0.4, 0.5]	0
pel	0	[0.2, 0.3, 0.4]	0
hasil	0	[0.5, 0.6, 0.3]	0
bersih	0	[0.4, 0.5, 0.2]	0
operasi	0	[0.3, 0.4, 0.6]	0
salmon	0	0	[0.7, 0.5, 0.4]
enak	0	0	[0.6, 0.3, 0.5]
potong	0	0	[0.4, 0.6, 0.3]
pas	0	0	[0.5, 0.4, 0.2]
matang	0	0	[0.3, 0.5, 0.6]

2) Mean Document Vector Calculation

Document representations were generated by computing the mean of the word vectors contained in each document. This process was performed by summing the values of each dimension across all word vectors and dividing them by the total number of words with corresponding Word2Vec vectors in the document. The mean document vector was calculated using the formula presented in Equation (4). For example, for Document 1, the word vectors were summed in each dimension as follows:

$$\sum_{i=1}^n \vec{w}_i = [1.4, 1.4, 1.5]$$

Since Document 1 consists of four words, the mean document vector was calculated as follows:

$$\vec{d} = \frac{1}{4} [1.4, 1.4, 1.5] = [0.350, 0.350, 0.375]$$

The same calculation procedure was applied to all documents to obtain the mean document vector values. The resulting mean document vectors are presented in Table 11.

Table 11. Mean Document Vector Calculation Results

Document	Sum of Vectors	Number of Words	Mean Vector
D1	[1.4; 1.4; 1.5]	4	[0.350; 0.350; 0.375]
D2	[2.3; 2.7; 2.4]	6	[0.383; 0.450; 0.400]
D3	[3.3; 3.0; 3.0]	7	[0.471; 0.429; 0.429]

3) Document Vector Generation

The resulting mean vector for each document forms the final document representation in numerical vector form. This representation captures the combined information from all words within a document and is subsequently used as input features for classification using the Support Vector Machine (SVM) algorithm.

3.2.3. Text Representation

Based on the representation process, text representations were generated for the training data using the TF-IDF and Word2Vec methods. The resulting representations are presented in Table 12.

Table 12. Text Representation Results

Text Representation	Training Matrix Size	Testing Matrix Size
TF-IDF	8,991 x 5,000	2,248 x 5,000
Word2Vec	8,991 x 100	2,248 x 100

The TF-IDF method produced a feature matrix of size 8,991 × 5,000, indicating that 8,991 training instances were represented using 5,000 term-weighted features. This representation generated a sparse data structure in which the feature values reflect the relative importance of terms in each document.

In contrast, the Word2Vec method produced a feature matrix of size $8,991 \times 100$, indicating that 8,991 training instances were represented as 100-dimensional document vectors. This representation generated a dense data structure designed to capture semantic relationships among words. The differences in representation characteristics between the two methods were subsequently used as the basis for the classification process and model performance evaluation.

3.3. Classification Results

The classification performance of the TF-IDF and Word2Vec text representations was assessed using five-fold stratified cross-validation and a hold-out test set. The cross-validation results were used to examine the consistency and robustness of model performance during model development, while the hold-out test set was used for final evaluation on previously unseen data. Model performance was evaluated based on accuracy, macro precision, macro recall, and macro F1-score. In addition, class-wise evaluation was conducted using precision, recall, F1-score, support, and confusion matrices to provide a more comprehensive assessment of classification performance.

3.3.1. Overall Classification Performance

The overall classification performance of the TF-IDF and Word2Vec text representations was evaluated using the hold-out test set, which consisted of 20% of the dataset and was not involved in model training, cross-validation, or SMOTE. Performance was assessed using accuracy, macro precision, macro recall, and macro F1-score. The evaluation results are presented in Table 13.

Table 13. SVM Performance Evaluation Results

Text Representation	Accuracy	Macro Precision	Macro Recall	Macro F1-Score
TF-IDF + SVM	0.9542	0.9168	0.9222	0.9188
Word2Vec + SVM	0.9101	0.8455	0.8538	0.8491

Table 13 reports the performance of the two approaches across the evaluation metrics. The TF-IDF-based SVM achieved 95.42% accuracy, with macro precision, macro recall, and macro F1-score of 91.68%, 92.22%, and 91.88%, respectively. Meanwhile, the Word2Vec-based SVM obtained 91.01% accuracy, 84.55% macro precision, 85.38% macro recall, and 84.91% macro F1-score. Based on these descriptive results, TF-IDF produced higher

classification performance than Word2Vec under the experimental conditions of this study.

3.3.2. Class-wise Performance

A category-level evaluation was conducted to examine the classification performance in greater detail. Precision, recall, F1-score, and support were calculated for each product category based on the hold-out test data. The resulting performance of the TF-IDF-SVM and Word2Vec-SVM models is reported in Table 14.

Table 14. Classification Results for the TF-IDF and Word2Vec

Product Category	TF-IDF			Word2Vec			Support
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	
Electronics	0.87	0.86	0.87	0.80	0.76	0.78	146
Mobile Phones & Tablets	0.94	0.90	0.92	0.83	0.81	0.82	160
Health	0.93	0.91	0.92	0.81	0.87	0.84	275
Food & Beverages	0.99	0.98	0.99	0.98	0.96	0.97	1147
Sports	0.96	0.97	0.96	0.94	0.93	0.93	381
Hardware & Tools	0.81	0.91	0.86	0.73	0.80	0.76	139

Table 14 show that the TF-IDF achieved higher precision, recall, and F1-score than the Word2Vec model across all six product categories. The highest F1-score was obtained for the Food & Beverages category, with 0.99 for TF-IDF and 0.98 for Word2Vec. In contrast, the Hardware & Tools category obtained the lowest F1-score for both methods, with 0.86 for TF-IDF and 0.76 for Word2Vec. Overall, TF-IDF demonstrated better class-wise classification performance across all six categories than Word2Vec.

3.3.3. Confusion Matrix Analysis

To further examine the classification performance of both text representation methods, confusion matrices were analyzed to identify correctly classified instances and common misclassification patterns across product categories. The confusion matrices of the TF-IDF + SVM and Word2Vec + SVM models are presented in Figures 5 and 6.

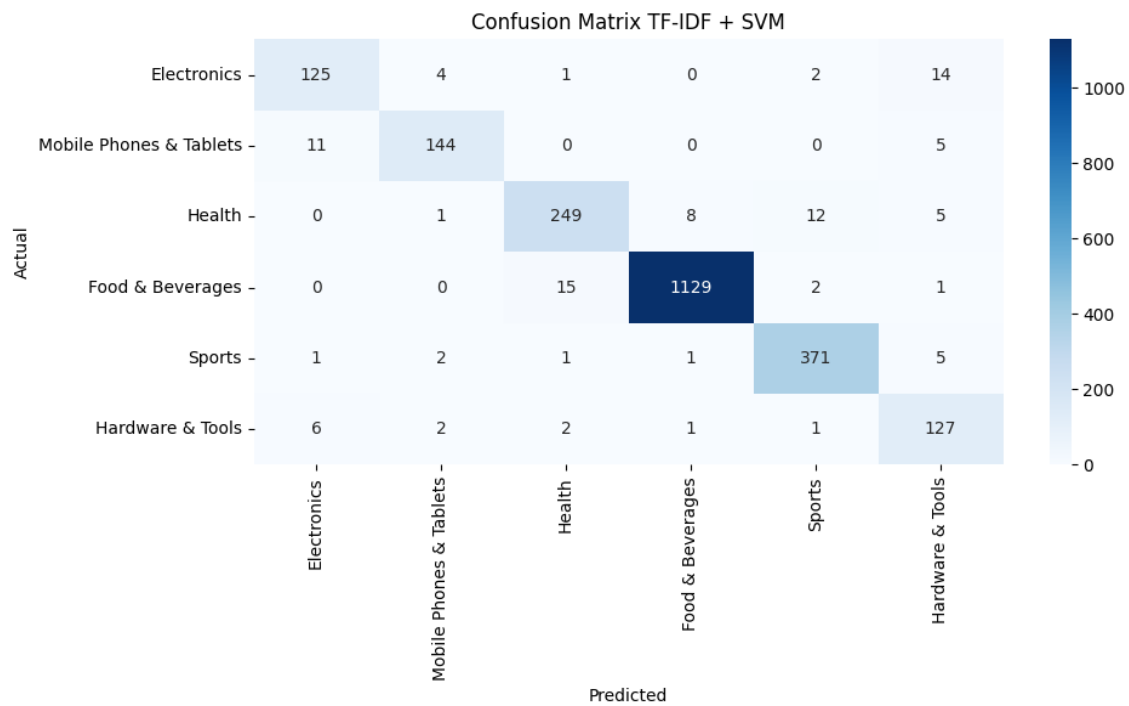


Figure 5. Confusion Matrices of the TF-IDF + SVM

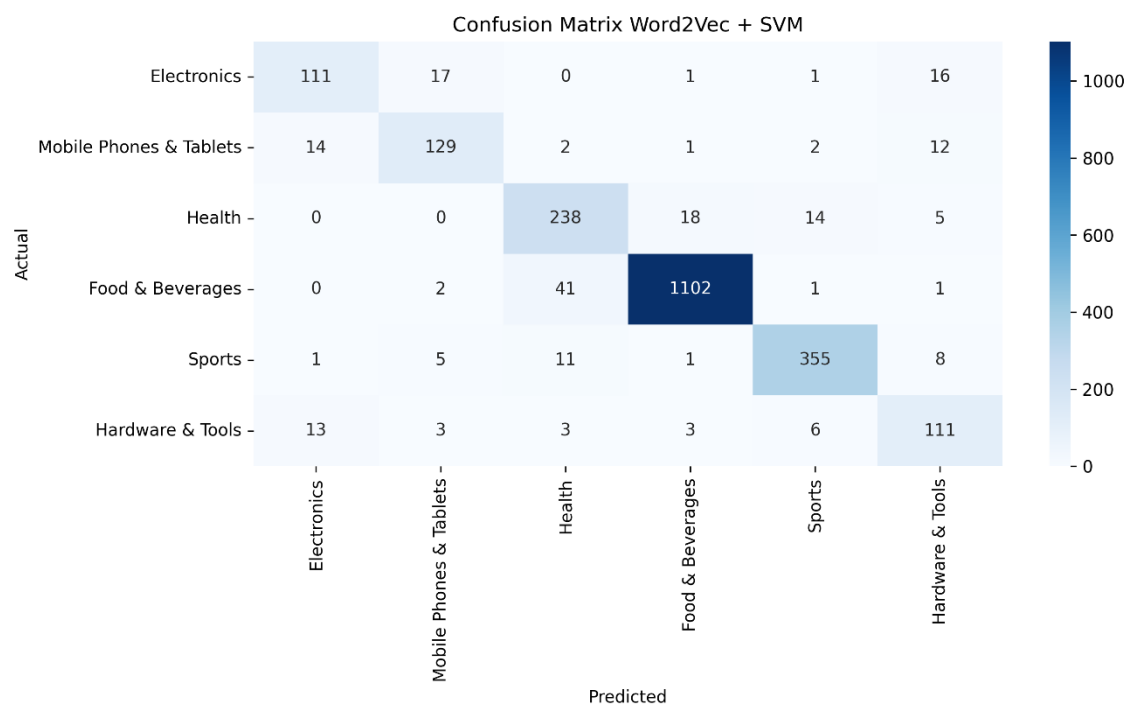


Figure 6. Confusion Matrices of the Word2Vec + SVM

Figures 5 and 6 show that the TF-IDF + SVM model produced higher diagonal values than the Word2Vec + SVM model across all six product categories, indicating a higher number

of correctly classified instances. Both models achieved the highest number of correct predictions for the Food & Beverages category, with 1,129 instances for TF-IDF + SVM and 1,102 instances for Word2Vec + SVM. The most notable misclassification for the TF-IDF + SVM model occurred between Food & Beverages and Health, with 15 Food & Beverages instances classified as Health. In comparison, the Word2Vec + SVM model misclassified 41 Food & Beverages instances as Health. Overall, the confusion matrices indicate that TF-IDF + SVM produced fewer misclassifications and more correct predictions across the six product categories than Word2Vec + SVM.

3.4. Performance Comparison

This section presents a comparative evaluation of the TF-IDF and Word2Vec text representation methods. The comparison includes five-fold cross-validation to assess model robustness and generalization, with statistical significance subsequently assessed using the Wilcoxon signed-rank test to determine whether the observed performance differences are statistically significant.

3.4.1. Five-Fold Cross-Validation Comparison

The performance of the TF-IDF and Word2Vec classification models was first examined using five-fold stratified cross-validation. The mean and standard deviation of accuracy, macro precision, macro recall, and macro F1-score across the five folds are presented in Table 15.

Table 15. Five-Fold Cross-Validation Results

Text Representation	Accuracy	Macro Precision	Macro Recall	Macro F1-Score
TF-IDF	0.9595 ± 0.0045	0.9313 ± 0.0096	0.9278 ± 0.0084	0.9293 ± 0.0090
Word2Vec	0.9080 ± 0.0090	0.8458 ± 0.0169	0.8602 ± 0.0168	0.8519 ± 0.0165

Based on the results shown in Table 15, TF-IDF achieved higher performance than Word2Vec across the evaluated metrics, with an average accuracy of 95.95% and macro F1-score of 92.93%, compared with 90.80% and 85.19% for Word2Vec, respectively. The higher macro F1-score indicates that TF-IDF provided better overall classification performance across the product categories. Furthermore, the relatively small standard

deviations indicate that the performance of both text representation methods remained consistent across the five cross-validation folds. To provide a visual comparison of the performance differences across evaluation metrics, the average cross-validation scores of TF-IDF and Word2Vec are presented in Figure 7.

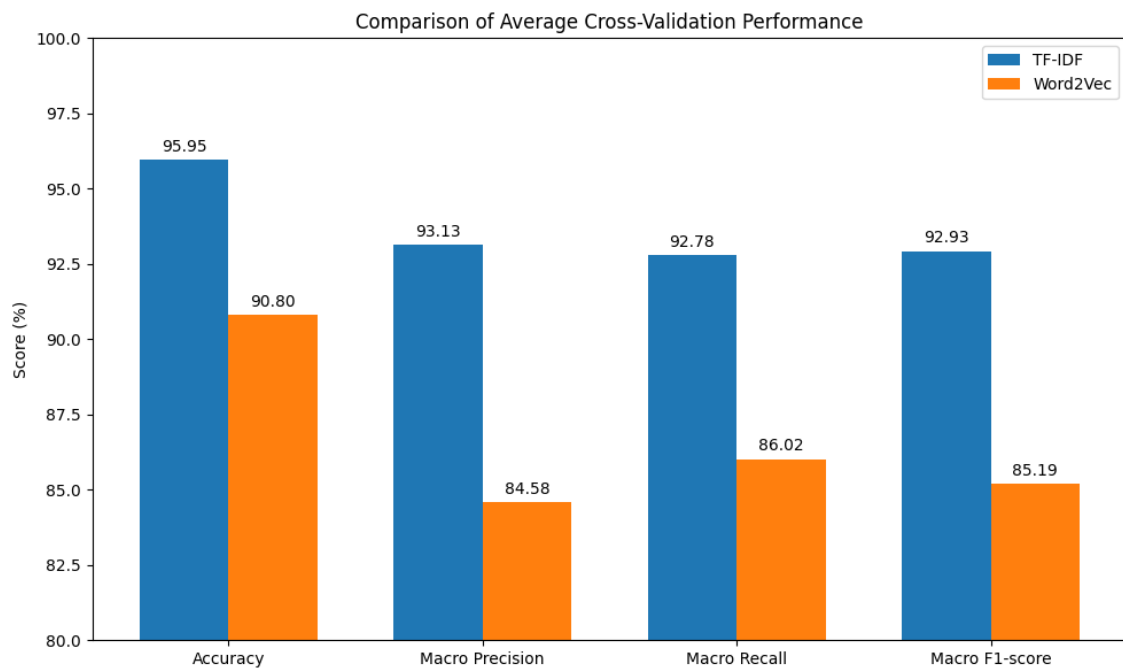


Figure 7. Comparison of Average Cross-Validation Performance of TF-IDF and Word2Vec

Figure 7 further illustrates the comparison of the average cross-validation scores between TF-IDF and Word2Vec. As shown in the figure, TF-IDF consistently outperformed Word2Vec across all evaluation metrics. Although both models exhibited relatively low standard deviation values, indicating stable performance across the five cross-validation folds, TF-IDF achieved higher mean accuracy, macro precision, macro recall, and macro F1-score. These findings show that the two text representation methods produced different levels of classification performance under the experimental settings adopted in this study.

3.4.2. Statistical Significance Analysis

To further assess whether the performance differences between the TF-IDF and Word2Vec representations were statistically significant, the Wilcoxon signed-rank test

was applied to the five-fold cross-validation scores. The test was conducted separately for accuracy, macro precision, macro recall, and macro F1-score, using a significance level of 0.05.

Table 16. Wilcoxon Signed-Rank Test Results

Evaluation Metric	Wilcoxon Statistic	p-value	Decision
Accuracy	0.0000	0.0625	Not significant
Macro Precision	0.0000	0.0625	Not significant
Macro Recall	0.0000	0.0625	Not significant
Macro F1-score	0.0000	0.0625	Not significant

The results showed a Wilcoxon statistic of 0.0000 and a p-value of 0.0625 for all four-evaluation metrics. Since the p-value was greater than the significance level of 0.05, the null hypothesis was not rejected, indicating that the observed differences between TF-IDF and Word2Vec were not statistically significant at the 5% significance level. Nevertheless, TF-IDF consistently achieved higher scores than Word2Vec across all five cross-validation folds and all evaluation metrics. Therefore, although TF-IDF demonstrated better performance descriptively, the statistical test did not provide sufficient evidence to confirm a significant difference between the two text representation methods.

3.5. Discussion

The experimental results indicate that the TF-IDF text representation obtained better classification performance than Word2Vec when combined with the Support Vector Machine (SVM) classifier. TF-IDF obtained higher scores in both the hold-out test and five-fold cross-validation, while also maintaining relatively low standard deviations across the cross-validation folds. However, the Wilcoxon signed-rank test produced a p-value of 0.0625, which was greater than the significance level of 0.05. Therefore, although TF-IDF consistently achieved higher performance than Word2Vec, the observed differences were not statistically significant at the 5% significance level. These findings indicate that TF-IDF provided better descriptive performance under the experimental settings adopted in this study, but the difference should not be interpreted as statistically significant.

The higher performance of TF-IDF may be related to the characteristics of the review dataset, which predominantly contains short texts with an average length of 8.30 tokens and a median of 6 tokens. For short-text classification, TF-IDF can emphasize terms that are highly informative for distinguishing product categories based on their frequency and distribution across documents. In contrast, the Word2Vec representation used in this study was constructed by averaging individual word vectors to obtain a single document vector. This averaging process may reduce the contribution of highly informative terms and does not preserve word-order information. As a result, the resulting document representation may be less discriminative for distinguishing between categories with overlapping vocabulary.

The confusion matrix analysis also revealed differences in classification difficulty among the product categories. The Food & Beverage category achieved the highest classification performance for both text representation methods, suggesting that reviews in this category contained relatively distinctive vocabulary. In contrast, more misclassifications occurred among the Electronics, Mobile Phones & Tablets, and Hardware & Tools categories. These categories may share product-related terminology, including references to devices, components, accessories, and technical specifications, which can create vocabulary overlap between categories. This overlap may make the categories more difficult to distinguish, particularly when the document representation is based on averaged word vectors.

This study has several limitations. First, the dataset was obtained from a single e-commerce platform, which may limit the generalizability of the findings to other platforms or domains. Second, a rule-based text refinement strategy using category-specific keywords was applied during preprocessing to remove generic and less informative reviews while retaining their original category labels. Although this strategy was intended to improve the quality of the textual data, the filtering process may have reduced the diversity of the dataset and potentially made the classification task easier by retaining reviews containing category-related terms. Therefore, the resulting performance may not fully represent classification performance on unfiltered review data. Future studies should evaluate the methods using more diverse and complete datasets and investigate alternative preprocessing strategies that reduce dependence on manually defined category-specific keywords. In addition, advanced text representation

methods such as FastText, BERT, and IndoBERT could be investigated to capture contextual and semantic information more effectively.

4. CONCLUSION

This study demonstrates that TF-IDF achieved higher classification performance than Word2Vec for multiclass product category classification using the Support Vector Machine (SVM) algorithm under the experimental settings adopted in this study. Evaluation through five-fold cross-validation showed that TF-IDF achieved higher mean accuracy, macro precision, macro recall, and macro F1-score than Word2Vec. The hold-out test results also showed that TF-IDF achieved higher performance across all evaluation metrics. Although TF-IDF consistently outperformed Word2Vec across the five cross-validation folds, the Wilcoxon signed-rank analysis produced a p-value of 0.0625, indicating that the observed differences did not reach statistical significance at the 5% level. These findings suggest that TF-IDF provided better descriptive performance for the short review texts used in this study. This study is limited to a filtered review dataset collected from a single e-commerce platform and employed a simple averaged Word2Vec representation. In addition, the rule-based text refinement strategy using category-specific keywords may have reduced the diversity of the dataset and potentially influenced the classification performance. Therefore, future research should evaluate the methods using more diverse and complete datasets, investigate alternative preprocessing strategies, and explore more diverse text representation methods, such as FastText, Doc2Vec, IndoBERT, multilingual BERT, and sentence embeddings. Future studies should also validate the proposed approach using reviews from multiple e-commerce platforms and external datasets and compare classification performance with and without keyword-based text refinement to investigate its potential influence on model performance.

REFERENCES

- [1] F. Tonda and M. Asif Khan, "Pengaruh Promosi dan Online Customer Reviews Terhadap Keputusan Pembelian Melalui Minat Beli (Literature Review Manajemen Pemasaran)," *J. Manaj. dan Pemasar. Digit.*, vol. 2, no. 3, pp. 305–318, 2024, doi: 10.38035/jmpd.v2i3.213.

- [2] T. Chen, P. Samaranayake, X. Y. Cen, M. Qi, and Y. C. Lan, "The Impact of Online Reviews on Consumers' Purchasing Decisions: Evidence From an Eye-Tracking Study," *Front. Psychol.*, vol. 13, no. June, 2022, doi: 10.3389/fpsyg.2022.865702.
- [3] Q. Li *et al.*, "A Survey on Text Classification: From Traditional to Deep Learning," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 2, 2022, doi: 10.1145/3495162.
- [4] F. S. Aditama, D. Krismawati, and S. Pramana, "Multiclass Classification Of Marketplace Products With Machine Learning," *MEDIA Stat.*, vol. 17, no. 1, pp. 25–35, Oct. 2024, doi: 10.14710/medstat.17.1.25-35.
- [5] M. F. Naufal and S. F. Kusuma, "Natural Language Processing Untuk Otomatisasi Pengenalan Pronomina Dalam Kalimat Bahasa Indonesia," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 9, no. 5, pp. 1011–1018, 2022.
- [6] A. R. Hanum *et al.*, "ANALISIS KINERJA ALGORITMA KLASIFIKASI TEKS BERT DALAM MENDETEKSI BERITA HOAKS Assyfa," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 11, no. 3, pp. 537–546, 2024, doi: 10.25126/jtiik2024118093.
- [7] S. S. Shaker, D. Alhajim, A. Ali, T. Al-Khazaali, H. Aqeel Hussein, and A. F. Athab, "Feature Extraction based Text Classification: A review," *J. Algebr. Stat.*, vol. 13, no. 1, pp. 646–653, 2022.
- [8] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manag.*, vol. 24, no. 5, pp. 513–523, Jan. 1988, doi: 10.1016/0306-4573(88)90021-0.
- [9] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *Advances in Neural Information Processing Systems*, C. J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., Curran Associates, Inc., 2013. [Online].
- [10] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.
- [11] D. R. V. Saputra, M. Maimunah, and E. R. Arumi, "Optimizing Aspect-Based Sentiment Analysis for Kyai Langgeng Park Using PSO and SVM," *J. Inf. Syst. Informatics*, vol. 6, no. 4, pp. 2856–2867, 2024, doi: 10.51519/journalisi.v6i4.930.
- [12] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.

- [13] F. Rifaldy, Y. Sibaroni, and S. S. Prasetyowati, "Effectiveness of Word2Vec and TF-IDF in Sentiment Classification on Online Investment Platforms using Support Vector Machine," *JIPi (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 10, no. 2, pp. 863–874, 2025.
- [14] J. Zhou, Z. Ye, S. Zhang, Z. Geng, N. Han, and T. Yang, "Investigating response behavior through TF-IDF and Word2vec text analysis: A case study of PISA 2012 problem-solving process data," *Heliyon*, vol. 10, no. 16, p. e35945, 2024, doi: 10.1016/j.heliyon.2024.e35945.
- [15] A. Al Tawil, L. Almazaydeh, D. Qawasmeh, B. Qawasmeh, M. Alshinwan, and K. Elleithy, "Comparative Analysis of Machine Learning Algorithms for Email Phishing Detection Using TF-IDF, Word2Vec, and BERT," *Comput. Mater. Contin.*, vol. 81, no. 2, pp. 3395–3412, 2024, doi: 10.32604/cmc.2024.057279.
- [16] D. I. Sumantiawan, J. E. Suseno, and W. A. Syafei, "Sentiment Analysis of Customer Reviews Using Support Vector Machine and Smote-Tomek Links For Identify Customer Satisfaction," *J. Sist. Info. Bisnis*, vol. 13, no. 1, pp. 1–9, 2023, doi: 10.21456/vol13iss1pp1-9.
- [17] S. Abdurrahman, "Tokopedia Product Reviews 2025," Kaggle. Accessed: Dec. 15, 2025. [Online]. Available: <https://www.kaggle.com/ds/8992580>
- [18] D. Rifaldi and A. Fadlil, "Teknik Preprocessing Pada Text Mining Menggunakan Data Tweet 'Mental Health,'" *Decod. J. Pendidik. Teknol. Inf.*, vol. 3, no. 2, pp. 161–171, 2023.
- [19] Ryzal Nur Alvandy and Arita Witianti, "Analisis Sentimen Ulasan E-Commerce Menggunakan Metode SVM," *Elkom J. Elektron. dan Komput.*, vol. 18, no. 2, pp. 164–175, 2025, doi: 10.51903/elkom.v18i2.3253.
- [20] A. B. Mutiara, E. P. Wibowo, and P. I. Santosa, "Improving the accuracy of text classification using stemming method , a case of non - formal Indonesian conversation," *J. Big Data*, pp. 1–16, 2021, doi: 10.1186/s40537-021-00413-1.
- [21] A. Alaiya, N. Nurdin, and C. Agusniar, "Sentiment Analysis of E-Commerce Product Reviews on Tokopedia Using Support Vector Machine," *J. Appl. Informatics Comput.*, vol. 9, no. 5, pp. 2869–2878, 2025, doi: 10.30871/jaic.v9i5.10977.
- [22] N. P. Yunita, "Aplikasi Pencarian Hadis Menggunakan Vector Space Model Dengan Pembobotan TF-IDF Dan Confix-Stripping Stemmer," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 3, pp. 665–676, 2023, doi: 10.25126/jtiik.2023106736.

- [23] Y. Wang, "Research on the TF-IDF algorithm combined with semantics for automatic extraction of keywords from network news texts," *J. Intell. Syst.*, vol. 33, no. 1, 2024, doi: 10.1515/jisys-2023-0300.
- [24] Z. Jiang, B. Gao, Y. He, Y. Han, P. Doyle, and Q. Zhu, "Text Classification Using Novel Term Weighting Scheme-Based Improved TF-IDF for Internet Media Reports," *Math. Probl. Eng.*, vol. 2021, pp. 1–30, Mar. 2021, doi: 10.1155/2021/6619088.
- [25] Yuliska, D. H. Qudsi, J. H. Lubis, K. U. Syaliman, and N. F. Najwa, "Analisis Sentimen Pada Data Saran Mahasiswa Terhadap Kinerja Departemen Di Perguruan Tinggi Menggunakan Sentiment Analysis in the Student ' S Reviews of College Department Performance Using," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 5, pp. 1067–1076, 2021, doi: 10.25126/jtiik.202184842.
- [26] Z. H. Kilimci and S. Akyokus, "Deep learning- and word embedding-based heterogeneous classifier ensembles for text classification," *Complexity*, vol. 2018, 2018, doi: 10.1155/2018/7130146.
- [27] G. B. R. Wasiat, P. S. Sasongko, E. Suharto, S. Kurniawan, I. Waspada, and Rismiyati, "Classification of Generative AI Answers Versus Human Answers on Indonesian Texts Using the TF-IDF and Support Vector Machine Methods," in *2025 8th International Conference on Informatics and Computational Sciences (ICICoS)*, 2025, pp. 247–252. doi: 10.1109/ICICoS68590.2025.11329997.
- [28] A. Nabilah, N. I. Sari, M. Afrina, and A. Ibrahim, "The Sentiment Analysis Of Indonesian Startup Application Reviews Using TF-IDF + SVM and FastText: A Comparative Study," vol. 10, no. 3, pp. 125–137, 2026.
- [29] M. Carpita and S. Golia, "Categorical classifiers in multiclass classification with imbalanced datasets," *Stat. Anal. Data Min.*, vol. 16, no. 4, pp. 391–405, 2023, doi: 10.1002/sam.11624.
- [30] Wijiyanto, A. I. Pradana, and V. Atina, "Teknik K-Fold Cross Validation untuk Mengevaluasi Kinerja Mahasiswa," *J. Algoritma*, vol. 21, no. 1, pp. 239–248, 2024, doi: 10.33364/algoritma/v.21-1.1618.