International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

# Sentiment Analysis of Indian Amazon Product Reviews: A Comparative Study of Machine Learning and Lexicon-Based Approaches

**Sachin Kalmani, Pratham Shinde**
MSC (Computer Science) Savitribai Phule Pune University Pune, India.

**Abstract- Sentiment analysis has become an essential technique for extracting actionable insights from user-generated content on e-commerce platforms. This study presents a comparative analysis of machine learning and lexicon-based approaches for sentiment classification of Amazon India product reviews, where sentiment labels are derived from user star ratings. Three machine learning models — Naive Bayes, Logistic Regression, and Random Forest — are evaluated alongside two lexicon-based methods, VADER and TextBlob. Text preprocessing and feature extraction are performed using standard natural language processing (NLP) techniques combined with TF-IDF vectorization. Models are tested under four train-test split configurations (80-20, 60-40, 40-60, and 20-80) to systematically assess the effect of training data size on performance. Results show that machine learning models consistently outperform lexicon-based approaches across all evaluation metrics. At the 80-20 split, Random Forest achieves the highest accuracy of 96.22%, followed by Logistic Regression at 85.97% and Naive Bayes at 80.58%. Lexicon-based methods plateau near 73-74% accuracy across all split configurations, confirming their insensitivity to training data volume. A notable finding is that at reduced training sizes, Naive Bayes (69.65% at 20-80) underperforms both VADER (74.11%) and TextBlob (72.90%), suggesting that lexicon-based methods are more reliable when labelled training data is scarce. These findings offer practical guidance for model selection in real-world sentiment analysis applications.**

**Keywords- Sentiment Analysis, Machine Learning, Amazon Product Reviews, Random Forest, Logistic Regression, Naive Bayes, VADER, TextBlob, TF-IDF Vectorization, Natural Language Processing, Indian E-Commerce.**

## I. INTRODUCTION

The rapid growth of e-commerce platforms has led to an unprecedented volume of user-generated content in the form of product reviews. These reviews are a critical resource for potential buyers, influencing purchasing decisions and shaping brand perception. Manually processing such large volumes of textual data is impractical, making automated sentiment analysis an important research and commercial priority.

Sentiment analysis, also referred to as opinion mining, is a computational technique used to identify the polarity of textual content — typically classified as positive, negative, or neutral. It has been widely applied across domains including product reviews, social media monitoring, and customer feedback systems. Two broad families of approaches dominate the field: lexicon-based methods that use predefined sentiment dictionaries, and machine learning models that learn classification patterns from labelled data.

Classical machine learning models such as Naive Bayes, Logistic Regression, and Random Forest have demonstrated strong performance on sentiment classification tasks by extracting statistical patterns from TF-IDF representations of text. In contrast, lexicon-based tools such as VADER and Text-Blob require no training data and are computationally lightweight, but are constrained by their inability to adapt to domain-specific language, sarcasm, or informal writing styles. Recent work on transformer-based models such as BERT has yielded further performance gains, though these approaches demand substantially greater computational resources and labelled data, limiting their practicality for lightweight deployments.

A recurring gap in existing literature is the limited focus on region-specific consumer behaviour. Most benchmark datasets for sentiment analysis are drawn from English-language global sources, and it is unclear whether findings generalize to

regional platforms where language style, product categories, and cultural context differ. Indian e-commerce in particular represents a distinctive context: consumer reviews frequently reflect local terminology, product-specific nuances, and a distinct purchasing culture that may not be well captured by general-purpose sentiment lexicons.

This study addresses this gap by conducting a comparative evaluation of machine learning and lexicon-based sentiment analysis techniques on Amazon India product review data. To rigorously assess the impact of training data availability, four different train-test split configurations are tested: 80-20, 60-40, 40-60, and 20-80. Performance is evaluated using accuracy, precision, recall, and F1-score. The findings contribute both methodological insights into model selection and practical guidance for deploying sentiment analysis systems in data-constrained environments.

## II. LITERATURE REVIEW

Sentiment analysis has been widely studied using machine learning, deep learning, transformer-based, and lexicon-based approaches. These methods have shown strong performance in classifying textual data across various domains. However, most existing studies rely on global datasets and overlook region-specific contexts. This review highlights key techniques, challenges, and the need for research focused on datasets such as Amazon India product reviews.

**1) Machine Learning-Based Sentiment Classification:**
A study by Dey L et al. (2016) explored the use of machine learning algorithms — specifically Naive Bayes and K-Nearest Neighbour (K-NN) — for sentiment classification of textual data [1]. The study demonstrated that Naive Bayes performs more effectively for movie review datasets due to its probabilistic nature, achieving above 80% accuracy and outperforming K-NN. For hotel reviews, both classifiers yielded lower and comparably similar accuracies. However, the study was conducted on general datasets (movie and hotel reviews) and did not focus on domain-specific data such as e-commerce reviews or regional consumer behaviour.

**2) Comparative Analysis of ML and Deep Learning Models on Amazon Reviews:**
A comparative study published in the MDPI journal Computers (2024) by Ashbaugh and Zhang evaluated multiple machine learning and deep learning models — including Logistic Regression, Random Forest, Naïve Bayes, Convolutional Neural Network (CNN), and Recursive Neural Network (RNN) — for sentiment analysis on Amazon product reviews [2]. The results indicated that deep learning models offered improved performance in capturing complex linguistic patterns, while traditional machine learning models remained computationally efficient. However, the study focused on global datasets and did not consider region-specific datasets such as Indian consumer reviews.

**3) Ensemble Learning for Sentiment Analysis:**
Research conducted by Breiman (2001) introduced the Random Forest algorithm, which has been widely applied in sentiment analysis tasks [3]. Ensemble models such as Random Forest have shown improved performance due to their ability to handle complex and non-linear relationships in textual data. However, many studies using these models do not compare them with lexicon-based approaches.

**4) Lexicon-Based Sentiment Analysis Approaches:**
Lexicon-based tools such as VADER proposed by Hutto and Gilbert (2014) and Text-Blob have been widely used for sentiment analysis [6]. These approaches rely on predefined sentiment dictionaries and are computationally efficient. However, they struggle with domain-specific language and contextual nuances.

**5) Challenges in Sentiment Analysis:**
A comprehensive survey by Medhat W et al. (2014) discussed challenges such as sarcasm detection, domain dependency, and ambiguity in sentiment analysis [7]. The study highlighted that no single model performs optimally across all domains.

**6) Sentiment Analysis on Product Review Data:**
The dataset introduced by McAuley J et al. (2015) has been widely used for sentiment analysis research on Amazon product reviews [8]. Many studies based on this dataset focus on general product categories rather than region-specific consumer behaviour.

**7) Machine Learning-Based Sentiment Classification on Amazon Reviews:**
A study by Kausar et al. (2023) applied machine learning techniques — specifically Decision Trees and Logistic Regression — for sentiment classification of Amazon product reviews using Bag-of-Words feature extraction [9]. The Decision Tree model achieved the highest accuracy of 99%,

while Logistic Regression achieved 94% accuracy. The study demonstrated that traditional machine learning models can effectively classify sentiments with strong accuracy. However, the study focused on general datasets and did not include comparative analysis with lexicon-based methods or region-specific datasets.

**8) Comparative Study of ML and Deep Learning Models:**
A study by Ali et al. (2024) conducted a comprehensive comparative analysis of machine learning and deep learning models — including Logistic Regression, Random Forest, Naive Bayes, CNN, Bidirectional LSTM, and transformer-based models such as BERT and XLNet — on Amazon review datasets [10]. The results showed that BERT outperformed all other models, achieving an accuracy of 89%, while traditional machine learning models remain computationally efficient. However, the study did not incorporate lexicon-based approaches or focus on specific regional datasets.

**9) Explainable Ensemble Learning for Amazon Sentiment Analysis:**
A 2025 study by Mokgwatjane and Paepae proposed an explainable ensemble learning framework for multiclass sentiment analysis (positive, neutral, negative) of Amazon product reviews across multiple domains such as appliances, groceries, and clothing [11]. The study demonstrated that ensemble models outperform individual classifiers and improve interpretability using SHAP-based explanations. However, the study focused on multi-domain datasets and did not consider region-specific sentiment analysis or Indian consumer behaviour.

**10) Domain-Specific Sentiment Analysis Using Transformer Models:**
A recent study by Al Hafidh and Al-Karawi (2025) applied the BERT transformer model for sentiment classification of Amazon electronics reviews using a large-scale dataset of over 6.7 million reviews [12]. The model achieved a training accuracy of 93.5% with an F1 score of 93.50, demonstrating BERT's effectiveness in detecting sentiment polarity through contextual understanding. However, the approach requires high computational resources and does not emphasize lightweight machine learning alternatives or regional datasets.

## III. MATERIAL & METHODS

This study compares machine learning and lexicon-based methods for sentiment analysis of Amazon product reviews, focusing on Indian consumer feedback. The methodology is divided into six stages: dataset collection, text preprocessing, feature extraction, model implementation, evaluation strategy, and experimental setup.

**Dataset Description:**
The dataset used in this study is the Indian Products on Amazon dataset obtained from Kaggle. It contains customer reviews from Amazon India, where each entry includes the review text and a corresponding star rating ranging from 1 to 5. Reviews with missing or empty text were removed during preprocessing.

For analysis, the ratings are converted into three sentiment categories: Negative (1–2 stars), Neutral (3 stars), and Positive (4–5 stars).

Table I: Sentiment Class Distribution

| Sentiment Class | Rating Range | No. of Samples |
|---|---|---|
| Negative | 1 - 2 stars | 676 |
| Neutral | 3 stars | 198 |
| Positive | 4 – 5 stars | 1902 |
| Total | | 2776 |

**Text Preprocessing Pipeline:**
Before feeding text into any model, the raw review data is cleaned and standardized through the following steps applied in order:

- **Case normalization**: All text is converted to lowercase to avoid treating the same word differently based on capitalization.
- **Noise removal:** URLs, HTML tags, punctuation, and special characters are removed using regular expressions.
- **Tokenization:** The cleaned text is split into individual words using NLTK's word tokenizer.
- **Stopword removal:** Common words that carry little meaning (e.g., the, is, and) are removed using NLTK's English stopword list.
- **Lemmatization:** Words are reduced to their base forms using WordNet Lemmatizer, which helps reduce vocabulary size while keeping the meaning intact.

**Feature Extraction**

Since machine learning models require numerical input, the pre-processed text is converted into numbers using Term Frequency–Inverse Document Frequency (TF-IDF) vectorization. TF-IDF gives higher weight to words that appear frequently in a specific review but rarely across the whole dataset, helping the model focus on meaningful words.

TF-IDF is chosen over simple Bag-of-Words because it better handles common but uninformative words. Advanced embedding methods like Word2Vec or BERT are not used in this study, as the focus is on comparing traditional approaches efficiently. A maximum vocabulary size of 5000 features is used, with sub-linear TF scaling applied to reduce the effect of very frequent terms.

**Model Implementation**

**1) Machine Learning Classifiers**

Three supervised machine learning models are trained and tested in this study:

- **Naive Bayes (Multinomial):** A probability-based classifier that works well with text data despite assuming all features are independent of each other. It is fast, simple, and serves as a strong baseline.
- **Logistic Regression:** A linear model that estimates the probability of each class. It is known for being reliable and easy to interpret in text classification tasks.
- **Random Forest:** An ensemble method that builds multiple decision trees and combines their results. It can capture complex patterns in data and tends to perform well even without much tuning.

All three models are trained on TF-IDF features, and hyperparameters are tuned using grid search with five-fold cross-validation on the training data.

**2) Lexicon-Based Approaches**

Two rule-based sentiment tools are also evaluated as baselines that do not require any training:

- **VADER:** A sentiment tool designed for informal and social media text. It uses a built-in word list along with rules for handling negation, capitalization, and punctuation to produce a sentiment score.
- **TextBlob:** A simpler tool that assigns polarity scores to text based on a predefined sentiment dictionary.

Since these tools do not need training data, they are directly applied to the test set in each split configuration, allowing a fair comparison with the trained models.

Experimental Design: Train-Test Split Strategy

To understand how the amount of training data affects model performance, four different train-test splits are used:

Table II: Training and Testing Percentage

| Configuration | Training Set | Test Set |
|---|---|---|
| Split 1 | 80% | 20% |
| Split 2 | 60% | 40% |
| Split 3 | 40% | 60% |
| Split 4 | 20% | 80% |

Stratified splitting is used in all cases to make sure each class is proportionally represented in both the training and test sets. This setup helps in understanding how well each model performs when less training data is available.

**Evaluation Metrics**

Each model is evaluated using four standard metrics:

- **Accuracy:** The percentage of total predictions that are correct.
- **Precision:** Out of all predicted positives, how many were actually positive.
- **Recall:** Out of all actual positives, how many were correctly predicted.
- **F1-Score:** The balance between precision and recall, especially useful when classes are not evenly distributed.

Since the dataset is imbalanced, macro-averaged F1-score is used as the main metric for comparing models.

## IV. RESULTS AND DISCUSSION

**Performance Evaluation Across Models**

The performance of all models is evaluated using accuracy, precision, recall, and F1- score across four train-test split configurations. Tables III-VI present the complete comparative results.

Table III: Performance Comparison – 80-20 Train-Test Split

| Model | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| Naive Bayes | 0.805755 | 0.774237 | 0.805755 | 0.761152 |

| | | | | |
|---|---|---|---|---|
| Logistic Regression | 0.859712 | 0.875294 | 0.859712 | 0.832453 |
| Random Forest | 0.96223 | 0.963686 | 0.96223 | 0.960173 |
| VADER | 0.73741 | 0.746223 | 0.73741 | 0.72076 |
| TextBlob | 0.732014 | 0.736448 | 0.732014 | 0.715442 |

Table IV: Performance Comparison – 60-40 Train-Test Split

| Model | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| Naive Bayes | 0.765077 | 0.747512 | 0.765077 | 0.705297 |
| Logistic Regression | 0.827183 | 0.851057 | 0.827183 | 0.793546 |
| Random Forest | 0.89829 | 0.90671 | 0.89829 | 0.892596 |
| VADER | 0.744374 | 0.744 | 0.744374 | 0.728893 |
| TextBlob | 0.736274 | 0.736663 | 0.736274 | 0.723707 |

Table V: Performance Comparison – 40-60 Train-Test Split

| Model | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| Naive Bayes | 0.731092 | 0.732665 | 0.731092 | 0.650234 |
| Logistic Regression | 0.807323 | 0.83746 | 0.807323 | 0.766283 |
| Random Forest | 0.860744 | 0.873164 | 0.860744 | 0.847946 |
| VADER | 0.741897 | 0.740444 | 0.741897 | 0.727961 |
| TextBlob | 0.732893 | 0.73739 | 0.732893 | 0.721233 |

Table VI: Performance Comparison – 20-80 Train-Test Split

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1 Score** |
|---|---|---|---|---|
| Naive Bayes | 0.696533 | 0.709382 | 0.696533 | 0.582681 |
| Logistic Regression | 0.751463 | 0.805662 | 0.751463 | 0.686596 |
| Random Forest | 0.805043 | 0.841288 | 0.805043 | 0.775105 |
| VADER | 0.741108 | 0.740866 | 0.741108 | 0.726214 |
| TextBlob | 0.728951 | 0.734749 | 0.728951 | 0.716105 |

Figure 1 illustrates the accuracy of all five models across these splits. Random Forest consistently achieved the highest accuracy at every training size, showing a steady upward trend and reaching approximately 96% at the 80% training split. Logistic Regression followed a similar improving trend, while VADER and TextBlob remained relatively stable across all splits due to their lexicon-based, training-independent nature. Naïve Bayes showed modest improvement with more data but remained below the supervised learning models.

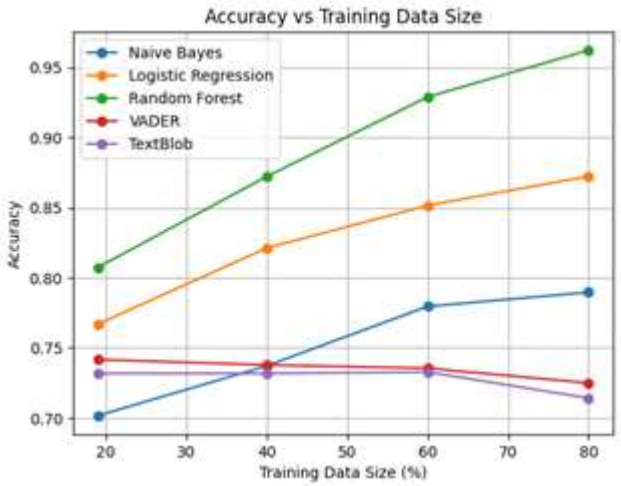


Fig.1. Accuracy vs Training Data Size

To provide a consolidated view of the performance of all models, Figure 2 presents a bar chart comparing the overall accuracy of each model under the 80-20 train-test split. Random Forest achieved the highest accuracy of approximately 96%, followed by Logistic Regression at around 87% and Naïve Bayes at approximately 79%. The lexicon-based approaches, VADER and TextBlob, recorded the lowest accuracies of roughly 72% and 71% respectively, highlighting the limitation of rule-based methods when applied to domain-specific or nuanced sentiment data. These results clearly demonstrate that ensemble-based supervised learning models, particularly Random Forest, are better suited for sentiment

classification tasks compared to traditional and lexicon-based methods.

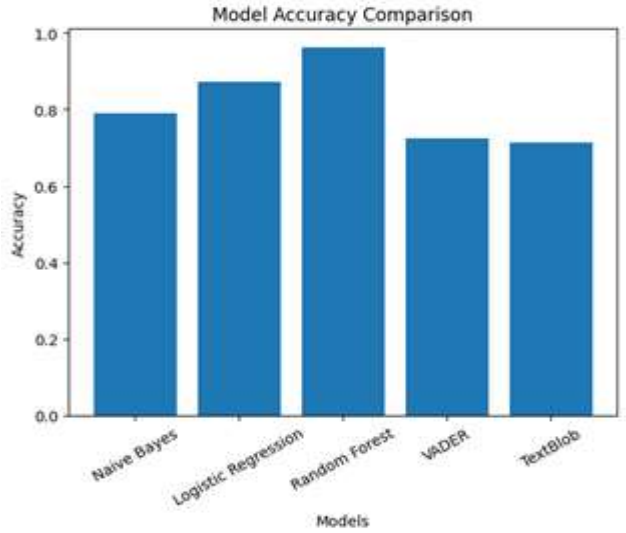


Fig.2. Model Accuracy Comparison (80-20 split)

To gain deeper insight into the classification behaviour of the best-performing model, the confusion matrix for Random Forest under the 80-20 train-test split is presented in Figure 3. Since generating confusion matrices for all models across all splits would be exhaustive, only the top-performing configuration is analysed here. As observed, the model correctly classified 244 negative, 48 neutral, and 1501 positive instances. The primary source of misclassification was the neutral class, where 106 neutral samples were predicted as positive and 4 as negative, reflecting the inherent ambiguity of neutral sentiment. The strong diagonal dominance confirms the model's robust classification capability, particularly for positive and negative sentiments.

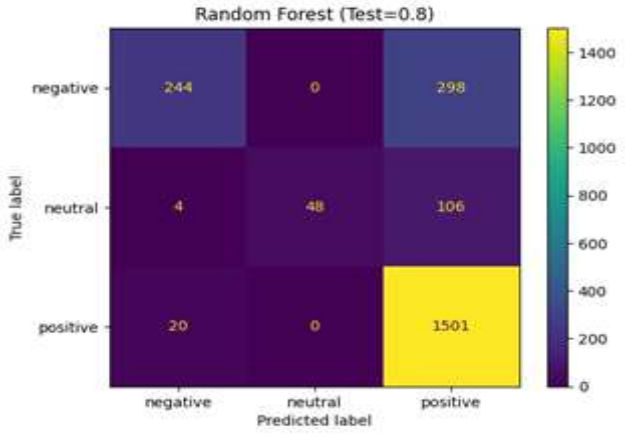


Fig.3. Confusion matrix of Random Forest

**Discussion**

The results consistently highlight the superiority of machine learning models over lexicon-based approaches across all split configurations, with several nuanced patterns worth noting.

Random Forest achieves the highest accuracy in every configuration, peaking at 96.22% under the 80-20 split and declining to 80.50% at the 20-80 split. Its strong performance stems from its ensemble architecture, which aggregates multiple decision trees to reduce variance and capture complex non-linear relationships in TF-IDF feature space. However, this exceptionally high accuracy should be validated using cross-validation in future work to rule out potential overfitting.

Logistic Regression offers a stable and interpretable alternative, achieving 85.97% at the 80-20 split and maintaining reasonable performance even under data-constrained conditions (75.15% at 20-80), making it a practical choice when model transparency is a priority.

Naïve Bayes exhibits the greatest sensitivity to training data size, dropping from 80.58% at the 80-20 split to 69.65% at the 20-80 split — falling below both VADER (74.11%) and TextBlob (72.90%). This crossover suggests that lexicon-based methods may be preferable to Naïve Bayes in low-resource scenarios.

VADER and TextBlob remain stable at approximately 73–74% across all splits, confirming their independence from training data. While this consistency makes them reliable baselines, their inability to adapt to domain-specific language limits their overall performance on Amazon India reviews.

This study has a few limitations. The dataset is restricted to English Amazon India reviews, which may limit generalizability. Sentiment labels derived from star ratings may not always reflect actual textual sentiment, and class imbalance may inflate accuracy metrics. These aspects should be addressed in future work.

## V. CONCLUSION

This study presents a systematic comparison of five sentiment analysis approaches — Naive Bayes, Logistic Regression, Random Forest, VADER, and TextBlob — evaluated on Amazon India product review data across four training data configurations. The results yield several actionable conclusions for practitioners choosing between sentiment analysis approaches.

When sufficient labeled data is available (80-20 or 60-40 splits), Random Forest is the recommended model, achieving up to 96.22% accuracy through its ensemble-based generalization capability. Logistic Regression is a strong alternative that balances accuracy with interpretability and computational efficiency, making it well-suited for production systems where model transparency is valued. Naive Bayes is best avoided when training data is limited, as it degrades significantly at lower data volumes and can be outperformed by simpler lexicon-based methods.

For scenarios where labeled training data is scarce or unavailable, VADER is the preferred lexicon-based tool, offering consistent performance near 74% accuracy with no training overhead. This makes it appropriate as a deployment baseline or for rapid prototyping. The stability of lexicon-based methods across all split conditions also makes them useful as sanity-check benchmarks in any sentiment analysis evaluation.

A key insight from this study is the data sensitivity crossover: when training data falls below approximately 20-40% of the dataset, Naive Bayes underperforms both VADER and TextBlob, inverting the typical hierarchy between ML and lexicon-based methods. Practitioners working in low-resource settings should take this into account when selecting models.
Future work should extend this study in several directions. Applying multilingual and code-mixed models such as IndicBERT or mBERT would better capture the linguistic diversity of Indian consumer reviews. Aspect-based sentiment analysis could provide finer-grained insights by linking sentiment to specific product attributes (e.g., price, delivery, quality). Additionally, addressing class imbalance through oversampling techniques and validating results with k-fold cross-validation would strengthen the reliability of the reported performance figures.

## REFERENCE

1. L. Dey et al., "Sentiment Analysis of Review Datasets Using Naive Bayes and K-NN Classifier," IJIEEB, vol. 8, no. 4, pp. 54–62, 2016.
2. L. Ashbaugh and Y. Zhang, "A Comparative Study of Sentiment Analysis on Customer Reviews Using Machine Learning and Deep Learning," Computers, vol. 13, no. 12, article 340, 2024.
3. L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
4. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
5. J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv:1810.04805, 2019.
6. C. J. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text," Proc. ICWSM, 2014.
7. W. Medhat, A. Hassan, and H. Korashy, "Sentiment Analysis Algorithms and Applications: A Survey," Ain Shams Engineering Journal, vol. 5, no. 4, pp. 1093–1113, 2014.
8. J. McAuley, R. Pandey, and J. Leskovec, "Inferring Networks of Substitutable and Complementary Products," Proc. ACM SIGKDD, 2015.
9. M. A. Kausar et al., "Sentiment Classification based on Machine Learning Approaches in Amazon Product Reviews," ETASR, vol. 13, no. 3, pp. 10849–10855, 2023.
10. H. Ali et al., "Analyzing Amazon Products Sentiment: A Comparative Study of Machine and Deep Learning, and Transformer-Based Techniques," Electronics, vol. 13, no. 7, article 1305, 2024.
11. K. Mokgwatjane and T. Paepae, "An Explainable Ensemble Machine Learning Approach for Multi-Domain, Multiclass Sentiment Analysis in Amazon Product Reviews," Machine Learning with Applications, 2025.
12. N. Al Hafidh and A. Al-Karawi, "Advanced Sentiment Analysis of Amazon Electronics Reviews Leveraging BERT," Procedia Computer Science, vol. 258, pp. 3608–3618, 2025.