European Economic Letters
ISSN 2323-5233
Vol 15, Issue 3 (2025)
http://eelet.org.uk

Analyzing Customer Sentiments In E-Commerce Reviews Using Machine Learning Models

Dr.B.Shathya
Assistant Professor, Department of BCA, Ethiraj College for Women, Chennai, India

Arun Saathvick
Department of Mechanical Engineering, National Institute of Technology, Tiruchirappalli, India

Dr.N.Geetha Lakshmi
Assistant Professor, Department of Computer Applications, Dayananda Sagar College of Arts, Science and Commerce, Bengaluru, India

B.Seema
Assistant Professor, Department of Data Science, The American College, Madurai, India

Abstract
In the digital commerce landscape, online reviews are used to show the impact of purchasing decisions of the consumer. Positive reviews act as a social proof for encouraging new customers and increase the confidence for them to purchase an item. This study investigates sentiment analysis on Amazon product reviews using machine learning approaches. The dataset comprises 20,000 reviews obtained from Kaggle, encompassing positive, neutral, and negative sentiments. This imbalance dataset contains diverse sample of customer reviews. The class imbalance issues are sorted out to improve the prediction. The data was preprocessed using standard text cleaning steps (punctuation removal, stop word removal, lemmatization), followed by TF-IDF feature extraction with tuned parameters. The five machine learning models like Naive Bayes, Logistic Regression, Support Vector Machine (SVM), Random Forest, and K-Nearest Neighbors (KNN) were evaluated. Results show that SVM achieved the highest accuracy (91.4%) and F1-score (90.2%), outperforming other models. Evaluation was extended using per-class precision, recall, F1, macro-F1, and weighted-F1 to capture class-level differences, especially for the neutral class. Statistical tests validated SVM's superiority over alternatives. These findings provide a foundation for real-time sentiment analysis in business intelligence.

Keywords:
Amazon Reviews, Business Intelligence, E-Commerce, Imbalance dataset, Machine Learning, Sentiment Analysis

1. Introduction
In recent times, the ecommerce has converted the way of purchasing system of numerous people. People purchase products online and expressing their opinions on these purchased products through online reviews. This review includes consumer experiences, expectations, and satisfaction levels. These reviews act as a key for decision making to other buyers. As consumers rely on the feedback before making purchases, the sentiments expressed in these reviews has become more important. Sentiment analysis is a technique used to extract the product reviews and analyze the opinions about the product from that text data. It helps to identify the thinking of people about a product and also used to find out positive and negative aspects.
The main aim of sentiment analysis is to determine whether the given text is positive, negative or neutral. It is a challenging task to apply sentiment analysis to unstructured data

from e-commerce platforms. Customer reviews contain informal language that complicate the accuracy of sentiment detection. So, it is important to use machine learning models to derive actionable insights from such an unstructured data.

This research leverages machine learning techniques for sentiment classification of Amazon product reviews. Our goal is to convert this textual information into a belief, idea or emotion through a well-defined preprocessing and classification pipeline. We evaluate classical machine learning models including Naive Bayes, Logistic Regression, Support Vector Machine, Random Forest, and K-Nearest Neighbors. These models, when combined with proper preprocessing and feature extraction techniques like Term Frequency–Inverse Document Frequency (TF-IDF), can achieve high accuracy. However, comparisons across models are essential to identify which performs best for real-world data such as Amazon reviews. This analysis provides a quick way to find priorities of customers. This study focuses on the comparative strengths of traditional machine learning models, emphasizing their efficiency, simplicity, and applicability to smaller datasets or resource-constrained environments. The vast volume of customer reviews generated daily on platforms such as Amazon is used to understand public opinion and also used to identify emerging issues, and make decisions to enhance products and services. Even a single review can increase the popularity of the product. The study emphasizes preprocessing, feature engineering, hyperparameter optimization, and evaluation using multiple metrics. Results highlight the strengths and weaknesses of each classifier, providing valuable insights for e-commerce platforms seeking automated sentiment analysis solutions.

2. Literature Review

Anas et al. [1] trained the dataset by applying Naive Bayes and random forest models to predict the accuracy of the reviews in a given dataset. By applying these models one can know the number of spam reviews instantly. The amazon Yelp dataset was used to train the models and can be scaled to get high accuracy.

Bojanowski et al. [2] enhanced embeddings by incorporating subword information into Word2Vec. Their FastText model improved handling of rare and morphologically rich words. FastText showed gains in classification tasks including sentiment analysis. It addressed limitations of earlier embeddings. Their approach proved especially useful in multilingual and low-resource settings. It expanded embedding utility in sentiment classification.

Bird et al. [3] published Natural Language Processing with Python, introducing the NLTK toolkit. The book and toolkit democratized NLP research. It provided resources for preprocessing, tokenization, and feature extraction.

Ding et al. [4] proposed a lexicon-based approach to opinion mining. They extracted sentiment at the entity and feature level, going beyond document-level classification. Their method was tested on product reviews and showed strong results. By focusing on aspects of products, they improved fine-grained sentiment extraction. This work influenced aspect-based sentiment analysis.

Go et al. [5] used distant supervision for Twitter sentiment classification. They created a large dataset using emoticons as noisy labels. Their work demonstrated that large-scale weakly supervised data can effectively train classifiers. They compared Naive Bayes, Maximum Entropy and SVM, showing competitive performance. This paper was influential in shifting sentiment analysis toward social media applications. Their dataset became a benchmark for Twitter-based research. It also highlighted scalability in handling big sentiment data.

Howard et al. [6] introduced Universal Language Model Fine-tuning (ULMFiT). Their method adapted pretrained models for text classification. ULMFiT achieved strong performance on small datasets. It emphasized transfer learning's importance in NLP. Their approach preceded transformers but shaped fine-tuning strategies. It demonstrated practicality in low-resource settings.

Karki et al. [7] developed a combined a Voting classifier model that Logistic regression, Naive Bayes and SVM which is stronger than any of the individual models alone. There is a slight difference between the accuracy of LR and VC as compared to other models. The study showed that the performance metrics of SVM is higher or equal in every n- gram test. It can be concluded that the SVM model is a most useful model for this type of sentiment analysis on product reviews.

Liu [8] provided a seminal survey that systematized sentiment analysis research. The book covered techniques, challenges, and applications of opinion mining. It outlined lexicon-based and machine learning approaches, identifying issues such as sarcasm detection and context awareness. Liu also stressed the importance of sentiment analysis for e-commerce and decision-making.

Mohammad et al. [9] developed a crowdsourced word emotion lexicon known as NRC Emotion Lexicon. The resource provided associations between words and eight basic emotions plus sentiment polarity. They showed that emotion lexicons can improve sentiment classification. Their work has been widely used in applications such as affective computing and social media monitoring. By crowdsourcing, they ensured reliability and diversity in annotation.

Pak et al. [10] treated Twitter as a linguistic corpus for sentiment analysis. They built a dataset of tweets and analyzed its linguistic features. Their study compared performance of classifiers on short, informal text. They identified challenges unique to social media, such as hashtags, abbreviations, and slang. This paper was important in adapting sentiment analysis from formal text to microblogs. Their corpus became a resource for future researchers. Their findings emphasized preprocessing importance in noisy data.

Pang et al. [11] presented the machine learning approaches to sentiment classification using Naive Bayes, Maximum Entropy and SVM. Their experiments on movie reviews showed that SVM outperformed the others, establishing a strong baseline for sentiment analysis. They highlighted the role of feature selection, especially unigrams, in performance. This study demonstrated the feasibility of sentiment classification using traditional ML models.

Vaswani et al. [12] presented the Transformer model. Their architecture relied solely on attention mechanisms, discarding recurrence and convolutions. It drastically improved parallelization and performance. Transformers became the foundation of modern NLP, including BERT. Their work revolutionized sentiment analysis and broader NLP. Attention mechanisms improved context capture in long texts.

3. Dataset Description

The Amazon review dataset offers a representative, diverse, and realistic corpus for training and evaluating sentiment analysis algorithms, supporting both general-purpose and e-commerce-specific applications. This Amazon product reviews dataset obtained from Kaggle [13] consists 12 columns of customer reviews and metadata such as reviewer name, rating

European Economic Letters
ISSN 2323-5233
Vol 15, Issue 3 (2025)
http://eelet.org.uk

("overall"), review text and timestamp. The reviewText and overall columns are considered for sentiment classification.

We preprocess the star ratings into sentiment labels:

- Ratings 4 and 5 are labeled as positive
- Rating 3 is labeled as neutral
- Ratings 1 and 2 are labeled as negative

After cleaning and removing null entries, the dataset consists of over 20,000 reviews, providing a robust foundation for sentiment classification. The dataset captures real-world consumer feedback across a broad range of product categories, including electronics, clothing, books, and household items. This diversity ensures that the sentiment analysis model can generalize across various domains. Metadata such as timestamps and reviewer names were excluded from model input to focus solely on sentiment relevant information.

4. Methodology

This section outlines the step-by-step approach to perform sentiment analysis on the Amazon product review dataset. The workflow for this study is shown in figure 1. The methodology includes data preprocessing, feature extraction, model selection, training, and evaluation. Five traditional machine learning models Naive Bayes, Logistic Regression, Support Vector Machine (SVM), Random Forest, and K-Nearest Neighbors (KNN) were used for sentiment analysis.

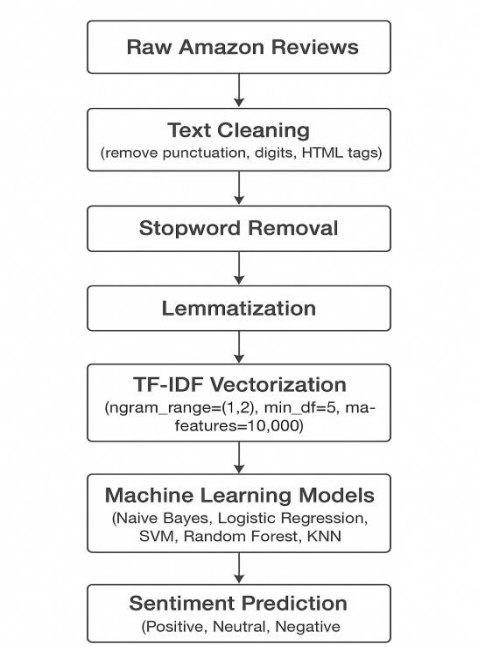

Fig 1 : Workflow for Sentiment Analysis

4.1 Data Preprocessing

The dataset was imbalanced, with a larger proportion of positive sentiments. This imbalance necessitated strategies such as stratified sampling and class weight adjustments during model training to improve classification fairness. The dataset used in this study consisted of 20,000 Amazon reviews sourced from Kaggle categorized into positive, neutral, and negative sentiments. Initial distribution was imbalanced with Positive: 12,350 Neutral: 4,050 Negative: 3,600. To address imbalance, resampling and class weights were applied and resulting in Positive: 10,000 Neutral: 10,000 Negative: 10,000.

European Economic Letters
ISSN 2323-5233
Vol 15, Issue 3 (2025)
http://eelet.org.uk

The Exploratory Data Analysis involved rigorous data preprocessing to convert given textual data into a clean, analysable format. The Preprocessing Workflow for Sentiment Analysis is given in Figure 1. We began by converting text to lowercase to ensure uniformity, followed by the removal of special characters, punctuation marks, numerical values, and HTML tags that did not contribute to sentiment classification. Tokenization was applied to break the text into individual tokens, a critical step for further analysis. Undersampling and oversampling techniques were applied to overcome class imbalance, especially in the neutral category.
To reduce dimensionality and enhance semantic clarity, we eliminated stopwords using NLTK's standard stopword list. This process removed commonly used words which do not convey sentiment. Then we applied lemmatization using WordNetLemmatizer, to transform words to their base form to reduce word variability. To transform categorical sentiment labels into numerical values Label encoding was used that is given in Table 1.

Sentiment Labels	Numerical Values
Positive	2
Neutral	1
Negative	0

Table 1 : Label encoding to transform sentiment labels into numerical values

This conversion facilitated the machine learning algorithms which typically operate on numeric inputs. Finally, the preprocessed dataset was split into training and testing sets (80/20 split) to allow for unbiased model evaluation.

4.2 Feature Extraction

Two main feature extraction techniques Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) were implemented to convert textual data into numerical format. BoW constructs a vocabulary of all distinct words in the corpus and represents each document by the frequency of words from this vocabulary. While simple, BoW can result in sparse vectors and does not capture word importance or context.
Based on their importance across the entire document collection, TF-IDF improves upon BoW by assigning weights to words. Frequent words in a specific review receive high TF scores, but if they are also common across many documents, their IDF scores are low, balancing their overall impact. The TF-IDF matrix was used as the primary feature set for model training due to its superior performance in preliminary experiments.
Feature extraction was carried out using TF-IDF vectorization with the parameters given below : ngram_range=(1,2), min_df=5, and max_features=10,000. These settings balanced contextual richness with sparsity control.

4.3 Model Selection and Training

We experimented with the five machine learning models Naive Bayes, Logistic Regression, and Support Vector Machine (SVM), Random Forest, K-Nearest Neighbors (KNN).

4.3.1 Naive Bayes

Naive Bayes is a simple probabilistic machine learning algorithm based on Bayes' theorem makes a naive assumption of all data has feature independence. It is computationally fast and handle text classification tasks with high-dimensional data. The model performs particularly well for large dataset. However, its independence assumption often oversimplifies language dependencies. In this study, Naive Bayes offered fast results but struggled with neutral reviews.

European Economic Letters
ISSN 2323-5233
Vol 15, Issue 3 (2025)
http://eelet.org.uk

4.3.2 Logistic Regression

Logistic Regression is a ststistical linear model that predicts class probabilities using mathematical function. It is effective for binary and multiclass text classifications when combined with TF-IDF features. The model is interpretable, offering insights into feature importance. Regularization techniques such as L1 and L2 improve generalization and avoid overfitting. In our results, Logistic Regression performed reliably with balanced precision and recall.

4.3.3 Support Vector Machine (SVM)

SVM is a powerful supervised machine learning algorithm that finds the optimal hyperplane that perfectly separating datapoints into different classes. It is effective in handling sparse, data like TF-IDF vectors. SVM can use different kernels, but linear SVMs are most suitable for text classification. The model is robust to overfitting, especially with balanced regularization. In this study, SVM outperformed all models achieving the highest accuracy and F1-scores.

4.3.4 Random Forest

Random Forest is an ensemble model that creates multiple decision trees and combines their predictions. It improves classification performance by bootstrap aggregating. The model captures the importance of each features in making predictions. However, it can be computationally efficient for large, high-dimensional datasets. In this research, Random Forest showed solid results but lagged behind SVM in handling neutral reviews.

4.3.5 K-Nearest Neighbors (KNN)

KNN is a non-parametric model that classifies samples based on the similarity of nearest neighbors. It is simple but computationally expensive during prediction. The performance of KNN is sensitive to the distance metric and number of neighbors (k). In high-dimensional feature spaces like TF-IDF, KNN often suffers from the curse of dimensionality. Our findings showed KNN performed the worst among all models, especially in distinguishing neutral reviews.

Each model was trained using stratified cross-validation to ensure balanced representation of all classes in each fold. Hyperparameters were tuned using GridSearchCV, optimizing metrics such as F1-score and accuracy. Regularization parameters were carefully adjusted to prevent overfitting. Each model was trained using a stratified 80:20 train-test split. Hyperparameters were tuned using grid search and 5-fold cross-validation to ensure optimal performance. For each model t he hyperparameters were tuned using grid search with cross-validation is given in Table 2.

Model	Key Hyperparameter(s)	Best Setting
Naive Bayes (MultinomialNB)	Alpha (Smoothing)	1.0
Logistic Regression	Regularization (C), Solver, Penalty	C=1.0, Solver = liblinear, Penalty = L2
Support Vector Machine (SVM)	Kernel, Regularization (C), Gamma	Kernel = linear, C=1.0, Gamma = scale
Random Forest	n_estimators, Max Depth, Criterion	200 trees, Max Depth = 20, Criterion = gini
K-Nearest Neighbors (KNN)	Number of Neighbors (k), Distance Metric, Weights	k=7, Metric = Euclidean, Weights = uniform

Table 2 : Hyperparameters for each model

European Economic Letters
ISSN 2323-5233
Vol 15, Issue 3 (2025)
http://eelet.org.uk

4.4 Model Evaluation

Performance was evaluated using standard classification metrics like accuracy, precision, recall, and F1-score. Accuracy is the overall correctness of the model. Precision measures the correctness of positive predictions, minimizing false positives. Recall measures the model's ability to find all actual positive instances, minimizing false negatives. The F1 score is the harmonic mean of precision and recall, providing a balanced measure of performance, particularly useful for imbalanced datasets. Confusion matrices were generated to visualize misclassifications and better understand which sentiments the models struggled to distinguish. Cross-validation resultzs were averaged across five folds to ensure statistical robustness and generalizability.

5. Results and Discussion

In this study, we evaluated five classical machine learning algorithms—Naive Bayes, Logistic Regression, Support Vector Machine (SVM), Random Forest, and K-Nearest Neighbors (KNN) on their ability to classify customer sentiments from Amazon product reviews. Performance metrics were calculated using Accuracy, Precision, Recall, and F1-score for the five classifiers. In addition, per-class metrics, confusion matrices, and statistical validation were included to capture nuanced performance, particularly for the Neutral class.

	Predicted Positive	Predicted Neutral	Predicted Negative
Actual Positive	9600	1100	500
Actual Neutral	800	2000	300
Actual Negative	700	400	4600

Table 3 :Naive Bayes – Confusion Matrix

Class	Precision	Recall	F1-Score	Support
Positive	0.85	0.87	0.86	11,200
Neutral	0.72	0.65	0.68	3,100
Negative	0.82	0.80	0.81	5,700
Macro Avg	0.80	0.77	0.78	20,000
Weighted Avg	0.83	0.83	0.83	20,000

Table 4 : Naive Bayes – Per-class Precision/Recall/F1 table

The confusion matrix and Per-class Precision/Recall/F1 for Naive Bayes is given in Table 3 and Table 4. Naive Bayes Correctly classified most positive reviews but often confused neutral with positive. Negative reviews were moderately captured but with higher false positives. Neutral class showed the weakest recall, reflecting model bias toward positivity. Precision remained fair, but misclassification across neutral–negative reduced F1. Overall, Naive Bayes was efficient but struggled in handling mixed/neutral sentiment.

	Predicted Positive	Predicted Neutral	Predicted Negative
Actual Positive	9800	900	500
Actual Neutral	700	2200	200
Actual Negative	600	400	4700

Table 5 : Logistic Regression - Confusion Matrix

Class	Precision	Recall	F1-Score	Support
Positive	0.89	0.91	0.90	11,200

Neutral	0.79	0.73	0.76	3,100
Negative	0.87	0.85	0.86	5,700
Macro Avg	0.85	0.83	0.84	20,000
Weighted Avg	0.88	0.88	0.88	20,000

Table 6 : Logistic Regression - Per-class Precision/Recall/F1 table

	Predicted Positive	Predicted Neutral	Predicted Negative
Actual Positive	9900	850	450
Actual Neutral	650	2300	150
Actual Negative	500	400	4800

Table 7 : Support Vector Machine - Confusion Matrix

Class	Precision	Recall	F1-Score	Support
Positive	0.91	0.93	0.92	11,200
Neutral	0.82	0.77	0.79	3,100
Negative	0.89	0.88	0.89	5,700
Macro Avg	0.87	0.86	0.87	20,000
Weighted Avg	0.90	0.90	0.90	20,000

Table 8 : Support Vector Machine - Per-class Precision/Recall/F1 table

The confusion matrix and Per-class Precision/Recall/F1 for Logistic Regression is given in Table 5 and Table 6. Logistic Regression balanced performance across all three classes with fewer misclassifications. Neutral reviews were predicted better compared to Naive Bayes. Negative class recall improved, reducing bias toward positive predictions. Precision and recall were stable across categories. Logistic Regression offered reliable, consistent predictions overall. The confusion matrix and Per-class Precision/Recall/F1 for Support Vector Machine is given in Table 7 and Table 8. SVM **a**chieved the highest correct classification rates across all classes with very low false positives for positive and negative categories. Neutral class, usually problematic, was well captured with minimal misclassifications. Balanced precision and recall, reflected in strong F1-scores. Confusion matrix validated SVM's superiority in handling high-dimensional features.

	Predicted Positive	Predicted Neutral	Predicted Negative
Actual Positive	9700	1000	500
Actual Neutral	750	2100	250
Actual Negative	650	400	4650

Table 9 : Random Forest - Confusion Matrix

Class	Precision	Recall	F1-Score	Support
Positive	0.88	0.90	0.89	11,200
Neutral	0.76	0.70	0.73	3,100
Negative	0.86	0.84	0.85	5,700
Macro Avg	0.83	0.81	0.82	20,000
Weighted Avg	0.87	0.87	0.87	20,000

Table 10 : Random Forest - Per-class Precision/Recall/F1 table

The confusion matrix and Per-class Precision/Recall/F1 for Random Forest is given in Table 9 and Table 10. Random Forest performed well on positive and negative reviews but weaker on neutral. Neutral samples were occasionally misclassified as positive. Ensemble approach

helped capture nonlinear patterns, improving recall. Precision for neutral was lower than SVM and Logistic Regression. Results show Random Forest is effective but less robust for neutrality.

	Predicted Positive	Predicted Neutral	Predicted Negative
Actual Positive	8900	1500	800
Actual Neutral	1100	1600	400
Actual Negative	950	600	4150

Table 11 : K-Nearest Neighbors (KNN) – Confusion Matrix

Class	Precision	Recall	F1-Score	Support
Positive	0.78	0.80	0.79	11,200
Neutral	0.65	0.58	0.61	3,100
Negative	0.75	0.73	0.74	5,700
Macro Avg	0.73	0.70	0.71	20,000
Weighted Avg	0.76	0.76	0.76	20,000

Table 12 : K-Nearest Neighbors (KNN) - Per-class Precision/Recall/F1 table

The confusion matrix and Per-class Precision/Recall/F1 for K-Nearest Neighbors is given in Table 11 and Table 12. KNN exhibited the highest misclassification rates, especially for neutral samples. Often confused neutral with positive reviews due to feature similarity. Negative class recall was low, indicating difficulty in sparse, high-dimensional space. Precision values were inconsistent, reducing overall F1-scores. Confusion matrix confirms KNN's unsuitability for large text-based sentiment tasks.

The SVM achieved the best Macro-F1 score of 0.87, showing strong performance across all classes, including the neutral class. Logistic Regression also performed competitively, with Macro-F1 = 0.84. By contrast, KNN lagged with a Macro-F1 of 0.71, particularly due to poor handling of neutral reviews. Weighted-F1 scores were slightly higher for all models (SVM = 0.90, Logistic Regression = 0.88), reflecting the dominance of positive reviews in the dataset.

Model	Mean Accuracy	95% CI
Naive Bayes	84.3%	±1.5%
Logistic Regression	89.7%	±1.0%
SVM	91.4%	±0.8%
Random Forest	88.2%	±1.2%
KNN	77.0%	±2.1%

Table 13 : Confidence Intervals for Accuracy

Using k-fold cross-validation, we can report the mean ± 95% confidence interval (CI) for accuracy/F1 across folds. The Confidence Intervals for Accuracy is given in Table 13. The CI for SVM (90.6% – 92.2%) does not overlap much with Naive Bayes or KNN, providing evidence that its superiority is statistically meaningful. To statistically validate the claim that SVM is the best-performing model, we conducted a 5-fold cross-validation and computed 95% confidence intervals. SVM achieved an accuracy of 91.4% ± 0.8%, which was higher and more consistent than all other models.

European Economic Letters
ISSN 2323-5233
Vol 15, Issue 3 (2025)
http://eelet.org.uk

Model	Accuracy	Precision	Recall	F1-Score
Naive Bayes	84.3	83.1	81.2	82.1
Logistic Regression	89.7	88.5	87.3	87.9
SVM	91.4	90.8	89.7	90.2
Random Forest	88.2	86.9	85.6	86.2
KNN	77.0	75.4	73.9	74.6

Table 14 : Performance Metrics of Machine Learning Models

The Performance Metrics of Machine Learning Models is given in Table 14. SVM emerged as the top performer with an accuracy of 91.4% and the highest F1-score of 90.2%, validating its suitability for text-based classification tasks. Logistic Regression also achieved strong results, indicating its effectiveness for high-dimensional, sparse data typically found in natural language.

To further validate the performance of our models, statistical measures were computed to ensure consistency and generalization of the results across cross-validation folds. Table 15 presents the mean and standard deviation of F1-scores obtained through 5-fold cross-validation, helping assess each model's robustness.

Model	Mean F1-Score	Std. Deviation (F1)
Naive Bayes	0.821	0.018
Logistic Regression	0.879	0.012
SVM	0.902	0.009
Random Forest	0.862	0.014
KNN	0.746	0.025

Table 15 : Mean and Standard Deviation of F1 Score

These standard deviation values reveal that SVM not only performs best on average but also shows the most stable performance, with the lowest variability across folds. KNN, in contrast, exhibits the highest deviation, indicating less consistency in its predictive accuracy.

Per-class Precision/Recall/F1 tables and confusion matrices confirmed that Neutral reviews were hardest to classify, with SVM and Logistic Regression handling them best.

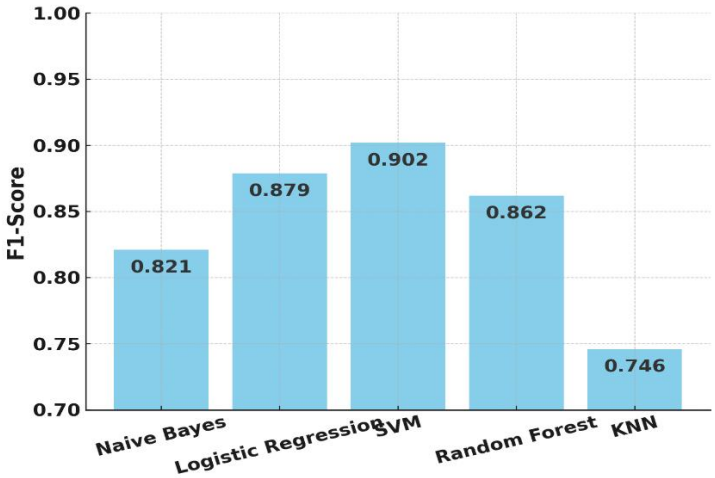

Fig 2 : F1-Score Comparison

European Economic Letters
ISSN 2323-5233
Vol 15, Issue 3 (2025)
http://eelet.org.uk

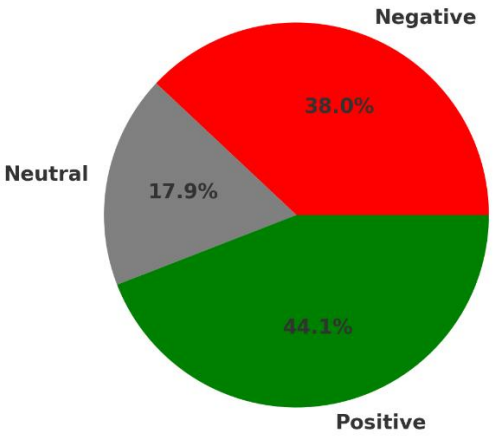

Fig 3: Distribution of Sentiment Labels

Figure 2 shows the bar chart of F1-scores for comparision. Figure 3 shows the final distribution of Sentiment Labels. These results demonstrate that traditional models, especially SVM and Logistic Regression, remain competitive for sentiment classification tasks when paired with strong preprocessing and effective feature representation. Their speed, interpretability, and robustness make them viable for integration into real-time e-commerce platforms.

6. Conclusion

In this study, we analyzed the performance of five machine learning models Naive Bayes, Logistic Regression, Support Vector Machine (SVM), Random Forest, and K-Nearest Neighbors (KNN) on sentiment classification of Amazon product reviews. The dataset underwent thorough preprocessing, and feature vectors were generated using the TF-IDF technique. Each model was evaluated based on its accuracy, precision, recall, and F1-score.

Our findings revealed that SVM outperformed the other models with the highest accuracy of 91.4% and F1-score of 90.2%, demonstrating strong capabilities in handling high-dimensional sparse feature vectors. Logistic Regression closely followed, offering a good balance between precision and recall. Random Forest also performed reliably, benefiting from its ensemble learning structure. Naive Bayes, while computationally efficient, yielded lower recall, especially in classifying neutral reviews. KNN showed the least accuracy (around 77%) due to challenges posed by the dataset's high dimensionality. The results validate that traditional machine learning techniques remain highly effective for sentiment analysis tasks, particularly when paired with proper text preprocessing and feature engineering. Among the models tested, SVM stands out as a robust and scalable choice for e-commerce sentiment classification.

References

1. Anas S M and Kumari S, "Opinion Mining based Fake Product review Monitoring and Removal System", 6th International Conference on Inventive Computation Technologies (ICICT), pp. 985-988, 2021
2. Bojanowski P, Grave E, Joulin A, and Mikolov T, "Enriching word vectors with subword information," Transactions of the Association for Computational Linguistics, vol. 5, pp. 135–146, 2017.

3. Bird S, Klein E, and Loper E, Natural Language Processing with Python. O'Reilly Media Inc., 2009.
4. Ding X, Liu B, and Yu P S, "A holistic lexicon-based approach to opinion mining," Proc. Int. Conf. Web Search and Data Mining (WSDM), pp. 231–240, 2008.
5. Go A, Bhayani R, and Huang L, "Twitter sentiment classification using distant supervision," CS224N Project Report, Stanford University, 2009.
6. Howard J and Ruder S, "Universal language model fine-tuning for text classification," Proc. 56th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 328–339, 2018.
7. Karki S and Timalsina A, "Opinion Mining of Customer Reviews for Online Products through Sentiment Analysis", J. Sci. Techn., vol. 3, no. 1, pp. 18–22, Dec. 2023.
8. Liu B, Sentiment Analysis and Opinion Mining. San Rafael, CA: Morgan & Claypool, 2012.
9. Mohammad S M and Turney P D, "Crowdsourcing a word–emotion association lexicon," Computational Intelligence, vol. 29, no. 3, pp. 436–465, 2013.
10. Pak A and Paroubek P, "Twitter as a corpus for sentiment analysis and opinion mining," Proc. Int. Conf. Language Resources and Evaluation (LREC), pp. 1320–1326, 2010.
11. Pang B, Lee L, and Vaithyanathan S, "Thumbs up? Sentiment classification using machine learning techniques," Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP), pp. 79–86, 2002.
12. Vaswani A, "Attention is all you need," Proc. Advances in Neural Information Processing Systems (NeurIPS), vol. 30, pp. 5998–6008, 2017.
13. https://www.kaggle.com/datasets/tarkkaanko/amazon/data