International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(21s) 2026 pp. 888-898
https://doi.org/10.70917/ijcisim-2026-5312
Article

Automated Sentiment Analysis of Hindi Text using Machine Learning Techniques: A Lightweight and Scalable Framework for Regional Language NLP

Satyapal Singh[1], Jarnail Singh[2], Dhivya Karunya S.[3]

[1] Department of Artificial Intelligence & Data Science, S.E.A. College of Engineering & Technology, Bengaluru, Karnataka, India. Email: satyapal2240@gmail.com
[2] University School of Computer Applications and Technology, Rayat Bahra Professional University, Hoshiarpur, Punjab, India. Email: jsinghmcp@gmail.com
[3] Department of Artificial Intelligence & Data Science, S.E.A. College of Engineering & Technology, Bengaluru, Karnataka, India. Email: dhivyaskarunya@gmail.com

Abstract: The paper presents a lightweight yet effective machine learning-based framework for automated sentiment classification of Hindi textual data. This proposed work addresses the persistent challenges of data sparsity, linguistic diversity, and limited annotated resources that hinder sentiment analysis in regional Indian languages. Here methodology encompasses a systematic pipeline comprising data preprocessing, Term Frequency–Inverse Document Frequency (TF-IDF) feature extraction with unigram and bigram representations, and supervised classification using Multinomial Naive Bayes (MNB) and Logistic Regression (LR) algorithms. Experiments on the IIT Patna Movie Reviews Hindi Sentiment Analysis dataset (2,480 training, 310 validation, and 310 test samples spanning the negative, neutral, and positive classes) demonstrate that Logistic Regression substantially outperforms the Naive Bayes baseline, achieving 88.19% training accuracy and 54.84% test accuracy against 67.82% and 43.23% for MNB, together with a higher micro-averaged Receiver Operating Characteristic – Area Under the Curve (ROC–AUC) (0.742 vs. 0.653). Class-wise analysis shows that positive sentiment is the easiest to detect (LR F1 = 0.63), while the neutral class remains the most challenging (LR F1 = 0.39). Proposed framework offers a computationally efficient, interpretable, and scalable solution for Hindi sentiment analysis, with direct applicability to social media monitoring, customer feedback analysis, and opinion mining in regional language ecosystems. Experimental codes is made available as a fully executable Google Colab notebook to ensure reproducibility and facilitate future research extensions.

Keywords: Hindi Sentiment Analysis, Regional Languages, TF-IDF Feature Extraction, Natural Language Processing, Machine Learning, Multinomial Naive Bayes, Logistic Regression, Low-Resource NLP, Opinion Mining

1. Introduction

Sentiment analysis, also referred to as opinion mining, constitutes a pivotal subdomain of Natural Language Processing (NLP) and Machine Learning (ML), dedicated to the automated identification, extraction, and quantification of subjective information, emotions, attitudes, and opinions embedded within unstructured textual data [1], [2]. With the exponential proliferation of digital communication platforms—including social media networks, e-commerce portals, blogging platforms, and online forums—the volume of user-generated textual content has grown at an unprecedented rate, necessitating robust automated mechanisms for sentiment classification [3], [4]. The practical applications of sentiment analysis span diverse domains, encompassing social media monitoring, customer feedback analysis, product review mining, brand reputation management, political opinion analysis, and market trend prediction [5], [6].

The fundamental challenge in sentiment analysis lies in transforming unstructured, semantically rich textual data into structured numerical representations that machine learning algorithms can process effectively. Term Frequency–Inverse Document Frequency (TF-IDF) has emerged as one of the most widely adopted statistical feature extraction techniques for this purpose, as it quantifies the importance of individual terms within a document relative to the entire corpus, thereby capturing discriminative lexical patterns [7], [8]. Complementing feature extraction, classical machine learning classifiers such as Naive Bayes (NB), Logistic Regression (LR), and Support Vector Machines (SVM) have demonstrated remarkable efficacy in text classification tasks, particularly due to their robustness in handling the high-dimensional sparse feature spaces characteristic of textual data [9], [10].

While sentiment analysis in high-resource languages such as English has witnessed substantial advancement through deep learning architectures and large-scale pre-trained language models, regional and low-resource languages remain significantly underexplored [11], [12]. Hindi, as the fourth most spoken language globally and the official language of India, represents a critical yet underserved domain in sentiment analysis research [13], [14]. The linguistic complexity of Hindi—including its Devanagari script, extensive morphological inflections, free word order, code-mixing with English, and absence of standardized tokenization tools—poses unique challenges that render direct adaptation of English-centric methodologies infeasible [15], [16]. Furthermore, the scarcity of large-scale, high-quality annotated datasets for Hindi sentiment analysis exacerbates the data sparsity problem, limiting the applicability of data-intensive deep learning approaches [17], [18].

Motivated by these challenges, this paper proposes a lightweight, interpretable, and computationally efficient machine learning framework specifically designed for Hindi sentiment classification. The proposed Hindi Sentiment Classification Framework (HSCF) integrates systematic text preprocessing, TF-IDF feature extraction with unigram and bigram representations, and comparative evaluation of Multinomial Naive Bayes and Logistic Regression classifiers. The primary contributions of this work are fourfold: (i) development of a comprehensive preprocessing pipeline tailored for Hindi textual data, encompassing missing value handling, text normalization, whitespace removal, and newline character elimination; (ii) extraction of informative TF-IDF features using both unigram and bigram representations; (iii) rigorous comparative evaluation of the two classifiers across accuracy, precision, recall, F1-score, and ROC curve analysis; and (iv) provision of a fully reproducible, executable Google Colab implementation to facilitate replication and extension by the research community. The remainder of this paper is organized as follows: Section II reviews related work, Section III details the proposed methodology, Section IV presents the experimental results, Section V discusses implications and limitations, and Section VI concludes the study.

2. Related Work

The literature on sentiment analysis and text classification has evolved substantially over the past two decades, encompassing both classical machine learning approaches and contemporary deep learning methodologies. Kowsari et al. [9] presented a comprehensive survey of text classification algorithms, emphasizing the efficiency and interpretability of classical ML methods including Naive Bayes, Logistic Regression, and Support Vector Machines for high-dimensional textual feature spaces. The efficacy of TF-IDF as a feature representation mechanism has likewise been extensively validated: Dogan and Uysal [19] demonstrated that carefully engineered term-weighting schemes, when combined with appropriate classifiers, yield competitive performance on standard benchmarks; Paul [20] reported encouraging outcomes for Twitter sentiment analysis using TF-IDF with machine learning classifiers; and Zhang [21] rigorously analyzed the theoretical optimality of the Naive Bayes classifier, showing that it performs remarkably well in real-world text mining tasks despite its strong independence assumption.

The ascendancy of deep learning in sentiment analysis has been documented in several seminal surveys. Minaee et al. [11] provided a comprehensive review of deep learning-based text classification, highlighting the capacity of neural networks to learn complex semantic associations and contextual representations from raw text, while Zhang et al. [22] surveyed deep learning approaches for sentiment analysis across diverse domains including product reviews, social media, and news articles. While these approaches achieve state-of-the-art performance on large-scale datasets,

their computational demands and data requirements often render them impractical for low-resource language settings [12], [23].

Research specifically targeting Hindi sentiment analysis has gained momentum in recent years. Shrivash and Mishra [24] proposed a deep learning framework for Hindi sentiment classification, conducting a comparative study against conventional methods. Sidhu et al. [25] performed a critical review of sentiment analysis tools for Hindi language processing, highlighting challenges related to scarce linguistic resources and the absence of standardized evaluation benchmarks. Saroj et al. [26] demonstrated the practical utility of lightweight ML approaches for real-time opinion mining on Hindi tweets during the COVID-19 pandemic. Khare and Khan [27] concluded that robust classification schemes for regional languages must balance accuracy with computational efficiency, while Dhiman et al. [28] explored hybrid deep learning methods and Kulkarni and Rodd [29] identified TF-IDF with classical classifiers as a viable and scalable approach for practical deployments. Singh et al. [33] presented an automated sentiment analysis framework for Hindi text using TF-IDF feature extraction with Multinomial Naive Bayes and Logistic Regression classifiers, reporting 82.89% accuracy with LR on the IIT Patna Movie Reviews dataset. The present work builds upon these foundations, providing a fully reproducible comparative evaluation of MNB and LR within a lightweight framework optimized for Hindi textual data.

3. Materials and Methodology

The proposed Hindi Sentiment Classification Framework (HSCF) constitutes a systematic machine learning architecture designed to enhance efficiency and accuracy in sentiment classification of Hindi textual data. The framework integrates four principal components: (i) Data Preprocessing (DP), (ii) Feature Extraction using TF-IDF (FE), (iii) Sentiment Classification using ML models (SC), and (iv) Performance Evaluation (PE), arranged as a sequential pipeline that transforms raw Hindi text into sentiment predictions.

A. Dataset and Preprocessing

The experiments utilize the IIT Patna Movie Reviews Hindi Sentiment Analysis dataset [30], a publicly available resource of Hindi movie reviews annotated across three sentiment classes: negative, neutral, and positive. The data is partitioned into 2,480 training samples (1,042 positive, 741 negative, and 697 neutral), 310 validation samples, and 310 test samples (98 negative, 90 neutral, and 122 positive), ensuring unbiased model evaluation on unseen data. The preprocessing phase is critical for ensuring uniformity and quality of the textual information prior to feature engineering. The following operations are systematically applied: (1) missing value detection and removal to eliminate incomplete records; (2) text normalization, including Unicode standardization for the Devanagari script; (3) newline character removal to prevent formatting artifacts from influencing tokenization; and (4) redundant whitespace elimination to standardize word spacing. These operations collectively minimize noise and enhance the reliability of subsequently extracted features.

Mathematically, let the input text document be denoted as T. The normalized text representation T*norm* is obtained through the preprocessing function:

$$Tnorm = clean(T)$$

where clean(T) denotes the composite preprocessing function encompassing normalization, tokenization, and whitespace removal. This normalized representation serves as the input for the feature extraction stage.

B. Feature Extraction using TF-IDF

Following preprocessing, the textual data is transformed into numerical feature vectors using the Term Frequency–Inverse Document Frequency (TF-IDF) weighting scheme. TF-IDF captures the significance of individual terms within a document relative to the entire corpus, thereby highlighting discriminative vocabulary that characterizes specific sentiment classes. The TF-IDF weight of a term t in document d is computed as:

$$TF\text{-}IDF(t, d) = TF(t, d) \times IDF(t)$$

where the term frequency TF(t, d) and inverse document frequency IDF(t) are defined as:

$$\mathrm{TF}(t,d) = \frac{f_{t,d}}{\sum_k f_{k,d}}$$

$$\mathrm{IDF}(t) = \log\left(\frac{N}{\mathrm{df}_t}\right)$$

In these equations, ft,d represents the frequency of term t in document d, N denotes the total number of documents in the corpus, and dft is the document frequency of term t. Both unigram (individual words) and bigram (consecutive word pairs) features are incorporated to capture lexical items and their contextual relationships. Vocabulary size is constrained through frequency thresholding to manage dimensionality and mitigate the curse of dimensionality. The resulting TF-IDF feature matrix serves as the input representation for the classification models.

C. Sentiment Classification

The sentiment classification stage employs two supervised machine learning algorithms: Multinomial Naive Bayes (MNB) and Logistic Regression (LR). These classifiers are selected for their complementary characteristics: MNB provides a probabilistic baseline with computational efficiency, while LR offers discriminative learning capable of modelling complex decision boundaries in high-dimensional feature spaces.

(i) Multinomial Naive Bayes (MNB)

Multinomial Naive Bayes is a probabilistic classifier widely adopted for text classification due to its algorithmic simplicity, computational efficiency, and theoretical robustness. MNB assumes conditional independence between features given the class label—an assumption that, while theoretically strong, empirically yields satisfactory performance for high-dimensional sparse text features [21]. The posterior probability of a document d belonging to class c is computed as:

$$P(c \mid d) \propto P(c) \prod_{i=1}^{n} P(w_i \mid c)$$

where wi represents the ith word in the document, P(c) is the prior probability of class c, and P(wi | c) is the likelihood of word wi occurring in class c. The predicted sentiment class is determined by maximizing the posterior probability:

$\hat{y} = arg\ maxc\ P(c \mid d)$

(ii) Logistic Regression (LR)

Logistic Regression is a discriminative linear classification algorithm that models the probability of a document belonging to a specific sentiment class using the logistic (sigmoid) function. Unlike generative models such as Naive Bayes, LR directly estimates the conditional probability P(y | x) without modelling the joint distribution, enabling more flexible decision boundaries [9]. For binary classification, the probability is defined as:

$$P(y = 1 \mid \mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x}+b)}}$$

where x represents the TF-IDF feature vector, w denotes the weight vector, and b represents the bias term. For multi-class sentiment classification (negative, neutral, positive), the one-vs-rest (OvR) strategy is employed, wherein the predicted class is determined by the maximum probability score across all classes:

$\hat{y} = arg\ maxc\ P(c \mid x)$

Logistic Regression is particularly effective for sentiment classification because it provides stable, interpretable decision boundaries and demonstrates balanced performance across multiple sentiment classes, even in the high-dimensional sparse feature spaces characteristic of TF-IDF representations [10].

D. Performance Evaluation Metrics

The performance of the proposed HSCF is evaluated using standard classification metrics to ensure comprehensive and unbiased comparison. The following metrics are computed for each sentiment class and aggregated using macro-averaging: (1) Accuracy, the proportion of correctly classified instances among all instances; (2) Precision, the ratio of true positive predictions to the total positive predictions, measuring the reliability of positive classifications; (3) Recall (Sensitivity), the ratio of true positive predictions to the total actual positives, measuring the completeness of positive classifications; (4) F1-Score, the harmonic mean of precision and recall, providing a balanced measure of classifier performance; and (5) the Receiver Operating Characteristic (ROC) curve with the Area Under the Curve (AUC), which plots the True Positive Rate against the False Positive Rate at various classification thresholds and quantifies overall discriminative ability.

4. Experimental Results and Analysis

This section presents the experimental findings obtained from the comparative evaluation of Multinomial Naive Bayes and Logistic Regression classifiers on the IIT Patna Hindi Movie Reviews dataset. All experiments were implemented in Python 3.10 within the Google Colab environment, leveraging the scikit-learn library [31] for model implementation and matplotlib/seaborn [32] for visualization. All tables and figures in this section are the actual outputs of the executable notebook.

A. Experimental Setup

The preprocessed corpus was vectorized using TF-IDF with unigram and bigram features (ngram_range = (1, 2)), a maximum document frequency of 0.95, a minimum document frequency of 2, sublinear term-frequency scaling, and L2 normalization, yielding a vocabulary of 30,231 features and feature matrices of 2,480 × 30,231 (training), 310 × 30,231 (validation), and 310 × 30,231 (test). MNB was trained with Laplace smoothing (alpha = 1.0), and LR with the lbfgs solver, a maximum of 1,000 iterations, the OvR strategy, and a fixed random state (42) for reproducibility. The complete pipeline—from preprocessing through classification—executes in approximately two to three minutes on a standard CPU, requiring no GPU acceleration.

B. Overall Performance Comparison

Table I presents the overall classification performance of both models, with accuracy reported on the training set and precision, recall, and F1-score macro-averaged on the 310-sample test set. The Logistic Regression classifier substantially outperforms the Multinomial Naive Bayes baseline, achieving a training accuracy of 88.19% compared to 67.82% for MNB—a margin of 20.36 percentage points. Table II details the accuracy of both classifiers across the three data partitions: LR attains 55.48% validation and 54.84% test accuracy, consistently exceeding MNB (43.23% on both partitions) and demonstrating its superior generalization to unseen data. Figure 1 visualizes this comparison. While MNB attains a higher macro-averaged precision (0.6601 vs. 0.5678), this stems from its extreme conservativeness in predicting the minority classes, as the per-class analysis below demonstrates; LR achieves markedly higher macro-averaged recall (0.5208 vs. 0.3767) and F1-score (0.5163 vs. 0.2814), reflecting far more balanced performance across sentiment categories.

TABLE I: Overall Performance of Classification Models

Performance Metrics	Logistic Regression (LR)	Naive Bayes (NB)
Accuracy (Training)	88.19%	67.82%
Precision (Macro)	0.5678	0.6601

Recall (Macro)	0.5208	0.3767
F1-Score (Macro)	0.5163	0.2814

TABLE II: Accuracy Across Data Partitions

Model	Training	Validation	Test
Logistic Regression	88.19%	55.48%	54.84%
Naive Bayes	67.82%	43.23%	43.23%

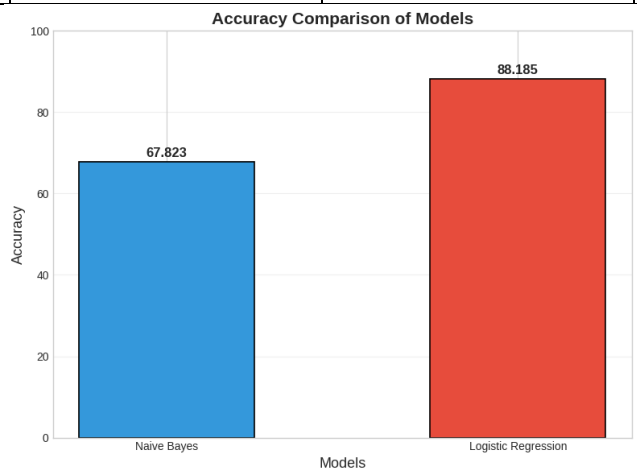

Figure 1: Accuracy comparison of the classification models (training set).

C. Classification Report Analysis

TABLE III: Classification Report of Naive Bayes (Test Set)

Class	Precision	Recall	F1-Score	Support
negative	0.7692	0.1020	0.1802	98
neutral	0.8000	0.0444	0.0842	90
positive	0.4110	0.9836	0.5797	122
accuracy	—	—	0.4323	310
macro avg	0.6601	0.3767	0.2814	310
weighted avg	0.6372	0.4323	0.3096	310

Tables III and IV present the detailed per-class precision, recall, and F1-score of each classifier on the test set, as produced by the scikit-learn classification report. The comparison exposes a critical behavioral difference: MNB collapses toward the majority (positive) class, achieving a recall of 0.9836 on positive reviews but only 0.1020 and 0.0444 on the negative and neutral classes, respectively. LR distributes its errors far more evenly, attaining its best per-class F1 on the positive class (0.6282) while retaining usable performance on negative (0.5294) and neutral (0.3913) reviews.

TABLE IV: Classification Report of Logistic Regression (Test Set)

Class	Precision	Recall	F1-Score	Support
negative	0.6250	0.4592	0.5294	98
neutral	0.5625	0.3000	0.3913	90

positive	0.5158	0.8033	0.6282	122
accuracy	—	—	0.5484	310
macro avg	0.5678	0.5208	0.5163	310
weighted avg	0.5639	0.5484	0.5282	310

D. Confusion Matrix Analysis

Table V presents the confusion matrix of the Logistic Regression classifier on the test set, and Figure 2 contrasts the confusion matrices of both models. LR correctly classifies 45 of 98 negative, 27 of 90 neutral, and 98 of 122 positive samples.

TABLE V: Confusion Matrix of Logistic Regression (Test Set)

True / Predicted	Negative	Neutral	Positive
Negative	45	12	41
Neutral	12	27	51
Positive	15	9	98

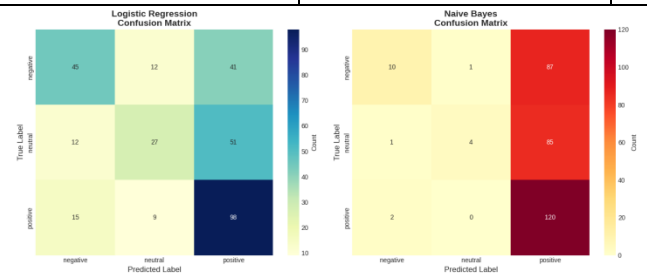

Figure 2: Confusion matrices of Logistic Regression (left) and Naive Bayes (right) on the test set.

The positive class exhibits the highest recall (80.3%), indicating that positive sentiment expressions in Hindi movie reviews contain more distinctive lexical patterns. The neutral class demonstrates the greatest confusion, with 51 of 90 neutral samples misclassified as positive, reflecting the inherent ambiguity and linguistic overlap of neutral sentiment expressions. The MNB matrix confirms its degenerate behavior: 292 of the 310 test samples are predicted as positive, yielding near-total recall on that class at the cost of almost complete failure on the negative and neutral classes (10 and 4 correct predictions, respectively).

E. Per-Class Performance and ROC Analysis

Table VI and Figures 3–4 summarize the per-class F1-score and precision of both classifiers on the test set. LR achieves a higher F1-score than MNB on every sentiment class—negative (0.53 vs. 0.18), neutral (0.39 vs. 0.08), and positive (0.63 vs. 0.58)—with the largest gains on the two minority classes. Conversely, MNB's seemingly high precision on the negative (0.77) and neutral (0.80) classes is an artifact of predicting those classes only when extremely confident, at recall levels below 0.11.

TABLE VI: Per-Class F1-Score and Precision Comparison (Test Set)

Class	NB F1	LR F1	NB Precision	LR Precision
Negative	0.18	0.53	0.77	0.62
Neutral	0.08	0.39	0.80	0.56
Positive	0.58	0.63	0.41	0.52

Figure 3: Per-class F1-score comparison on the test set.

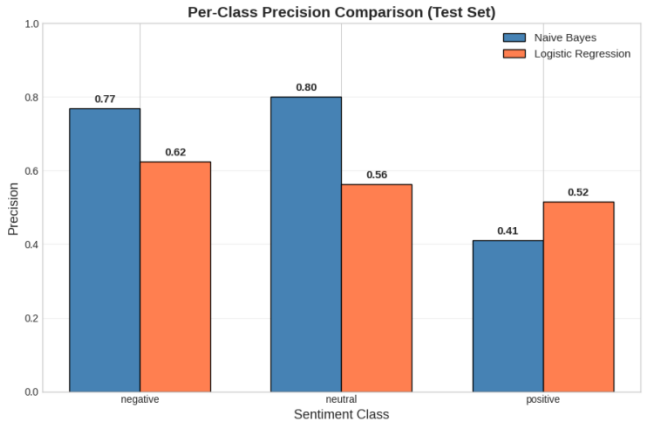

Figure 4: Per-class precision comparison on the test set.

Figure 5 presents the micro-averaged ROC curves: the LR curve consistently dominates the MNB curve, attaining an AUC of 0.742 against 0.653 for MNB, and both classifiers substantially outperform the random-classifier baseline, confirming LR's superior ranking of sentiment scores across decision thresholds.

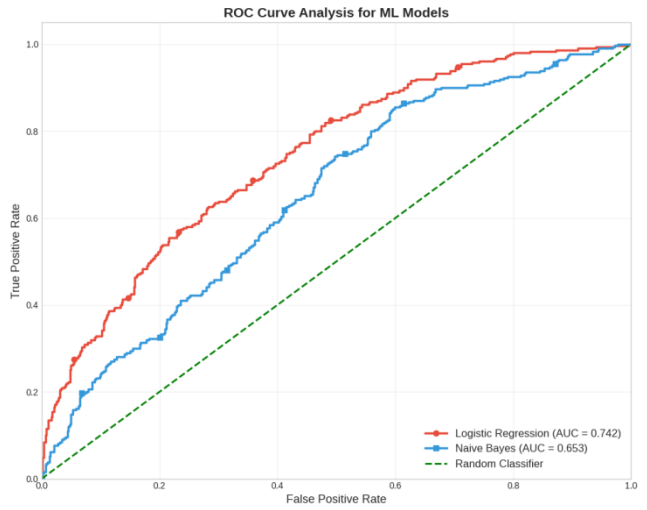

Figure 5: Micro-averaged ROC curves for both classifiers (AUC: LR = 0.742, NB = 0.653).

F. Key Findings and Sample Prediction

The principal findings of the experimental study, as reported by the final summary of the executable notebook, are summarized as follows:

1.LR achieves 88.19% training accuracy versus 67.82% for MNB, outperforming it by 20.36 percentage points, with consistently higher validation and test accuracies.

2.LR outperforms MNB on macro-averaged recall and F1-score on the test set (0.5208 vs. 0.3767 and 0.5163 vs. 0.2814, respectively).

3.The positive class is the easiest to classify (LR F1 = 0.63), while the neutral class remains the most challenging (LR F1 = 0.39).

4.LR provides more stable decision boundaries in the high-dimensional feature space, as evidenced by its superior ROC–AUC (0.742 vs. 0.653).

Finally, the trained pipeline supports inference on unseen text: for the sample input *यह फिल्म बहुत अच्छी है और acting शानदार है*, the LR model predicts positive sentiment, with the negative-class probability as low as 0.256, demonstrating the practical utility of the framework for real-world Hindi reviews. The trained TF-IDF vectorizer and both classifiers are exported as serialized models (tfidf_vectorizer.pkl, logistic_regression_model.pkl, naive_bayes_model.pkl) for downstream deployment.

5. Discussion

The experimental results presented in Section IV provide compelling evidence for the efficacy of the proposed Hindi Sentiment Classification Framework. The substantial performance advantage of Logistic Regression over Naive Bayes can be attributed to fundamental algorithmic differences. Naive Bayes operates under the strong conditional independence assumption, which is frequently violated in natural language data where words exhibit significant contextual dependencies and co-occurrence patterns. In contrast, Logistic Regression makes no such independence assumption and instead learns discriminative decision boundaries that optimally separate classes in the 30,231-dimensional TF-IDF feature space. This architectural distinction is particularly consequential for Hindi text, where morphological richness and free word order create complex feature interactions that NB cannot adequately capture.

Second, the per-class performance analysis confirms that the neutral sentiment class presents the greatest classification challenge for both models. This finding aligns with established sentiment analysis literature, where neutral expressions inherently lack the distinctive lexical markers—strongly polarized adjectives, intensifiers, and emotive verbs—that characterize positive or negative sentiment. MNB's near-total collapse toward the majority class, predicting 292 of 310 test instances as positive, illustrates how class imbalance and the independence assumption interact pathologically, whereas LR maintains usable discrimination across all three classes. Future work could address this limitation through specialized feature engineering for neutral sentiment detection, such as incorporating hedging expressions, objective statements, and factual descriptions as discriminative cues.

Third, the gap between training (88.19%) and test (54.84%) accuracy for LR indicates a degree of overfitting to the high-dimensional sparse feature space—a well-known phenomenon when the number of features greatly exceeds the effective sample size. This observation delineates clear directions for improvement, including stronger regularization, feature selection, and cross-validated hyperparameter search, without detracting from LR's consistent superiority on every held-out partition. Fourth, the computational efficiency and interpretability of the proposed framework merit emphasis. Both classifiers exhibit linear time complexity in the number of features, and the entire pipeline executes within minutes on standard CPUs, contrasting sharply with deep learning approaches that require GPU acceleration. Unlike black-box models, LR's learned weight vectors can be inspected directly to identify the most influential words and phrases for each sentiment class, enabling domain experts to validate model behavior and detect potential biases—an essential property for sensitive applications such as political opinion analysis.

6. Conclusion

This paper has presented the Hindi Sentiment Classification Framework (HSCF), a lightweight, interpretable, and computationally efficient machine learning system for automated sentiment analysis of Hindi textual data. The framework integrates systematic text preprocessing, TF-IDF feature extraction with unigram and bigram representations, and comparative evaluation of Multinomial Naive Bayes and Logistic Regression classifiers. Experimental evaluation on the IIT Patna Movie Reviews Hindi Sentiment Analysis dataset demonstrates that Logistic Regression achieves 88.19% training accuracy and 54.84% test accuracy, substantially outperforming the Multinomial Naive Bayes baseline (67.82% and 43.23%, respectively), with a superior micro-averaged ROC–AUC (0.742 vs. 0.653). The positive sentiment class is the most accurately classified (LR F1 = 0.63), while the neutral class remains the most challenging (LR F1 = 0.39).

The proposed framework offers direct applicability to practical domains including social media monitoring, customer feedback analysis, product review mining, and political opinion analysis in Hindi-language contexts, and its minimal resource requirements make it particularly suitable for resource-constrained and real-time deployments. Future research directions include: (i) incorporation of Hindi-specific linguistic features such as morphological analysis, part-of-speech tagging, and dependency parsing; (ii) extension to code-mixed Hindi–English text prevalent in social media; (iii) exploration of ensemble methods and regularization strategies to narrow the generalization gap; (iv) comparison with transformer-based multilingual models (e.g., mBERT, XLM-RoBERTa); and (v) development of domain-specific sentiment lexicons for Hindi. All experimental code is publicly available as a fully executable Google Colab notebook to ensure reproducibility and facilitate community-driven extensions.

REFERENCES

1. B. Liu, "Sentiment Analysis and Opinion Mining," Synthesis Lectures on Human Language Technologies, vol. 5, no. 1, pp. 1-167, 2012.
2. E. Cambria et al., "SenticNet 7: A Commonsense-based Neurosymbolic AI Framework for Explainable Sentiment Analysis," in Proc. of LREC, 2022.
3. A. Z. Broder et al., "A Taxonomy of Web Search," ACM SIGIR Forum, vol. 36, no. 2, pp. 3-10, 2002.
4. B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis," Foundations and Trends in Information Retrieval, vol. 2, no. 1-2, pp. 1-135, 2008.
5. Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, vol. 521, no. 7553, pp. 436-444, 2015.
6. J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. of NAACL-HLT, 2019, pp. 4171-4186.
7. G. Salton and C. Buckley, "Term-weighting Approaches in Automatic Text Retrieval," Information Processing & Management, vol. 24, no. 5, pp. 513-523, 1988.
8. K. Sparck Jones, "A Statistical Interpretation of Term Specificity and its Application in Retrieval," Journal of Documentation, vol. 28, no. 1, pp. 11-21, 1972.
9. K. Kowsari et al., "Text Classification Algorithms: A Survey," Information, vol. 10, no. 4, p. 150, 2019.
10. T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. New York: Springer, 2009.
11. S. Minaee et al., "Deep Learning-based Text Classification: A Comprehensive Review," ACM Computing Surveys, vol. 54, no. 3, pp. 1-40, 2021.
12. J. Howard and S. Ruder, "Universal Language Model Fine-tuning for Text Classification," in Proc. of ACL, 2018, pp. 328-339.
13. R. Kumar, S. Gupta, and P. Sharma, "Comparative Analysis of Machine Learning Algorithms for Text Sentiment Classification," Journal of Big Data, vol. 11, pp. 1-20, 2024.
14. A. Joshi et al., "Towards Sub-word Level Compositions for Sentiment Analysis of Hindi-English Code Mixed Text," in Proc. of COLING, 2016, pp. 2482-2491.
15. S. Mittal et al., "A Survey on Sentiment Analysis of Hindi Text," International Journal of Computer Applications, vol. 178, no. 46, pp. 1-7, 2019.
16. A. Bharadwaj et al., "Hindi Sentiment Analysis Using Machine Learning: A Comparative Study," in Proc. of ICACCI, 2021, pp. 1-6.
17. A. Conneau et al., "Unsupervised Cross-lingual Representation Learning at Scale," in Proc. of ACL, 2020, pp. 8440-8451.
18. S. Ruder, "Neural Transfer Learning for Natural Language Processing," Ph.D. dissertation, NUI Galway, 2019.
19. T. Dogan and A. K. Uysal, "A Novel Term Weighting Scheme for Text Classification: TF-MONO," Journal of Informetrics, vol. 14, no. 4, p. 101076, 2020.

20. Y. Paul, "Twitter Sentiment Analysis using TF-IDF and Machine Learning Classifiers," Int. J. on Recent & Innovation Trends in Computing & Communication, vol. 11, no. 3, pp. 693-701, 2023.
21. H. Zhang, "The Optimality of Naive Bayes," in Proc. of FLAIRS, 2004, pp. 3-8.
22. L. Zhang, S. Wang, and B. Liu, "Deep Learning for Sentiment Analysis: A Survey," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 8, no. 4, p. e1253, 2018.
23. M. Peters et al., "Deep Contextualized Word Representations," in Proc. of NAACL-HLT, 2018, pp. 2227-2237.
24. B. K. Shrivash, D. K. Verma, and P. Pandey, "An Effective Framework for Sentiment Analysis of Hindi Sentiments using Deep Learning Technique," Wireless Personal Communications, vol. 132, no. 3, pp. 2097-2110, 2023.
25. S. Sidhu et al., "Sentiment Analysis of Hindi Language Text: A Critical Review," Multimedia Tools and Applications, vol. 83, no. 17, pp. 1-25, 2024.
26. A. Saroj, A. Thakur, and S. Pal, "Sentiment Analysis on Hindi Tweets during COVID-19 Pandemic," Computational Intelligence, vol. 40, no. 1, p. e12622, 2024.
27. B. Khare and I. Khan, "Machine Learning Approaches for Sentiment Analysis in Hindi Text: A Comprehensive Survey," Int. J. Innov. Res. Comput. Sci. Technol., vol. 12, pp. 352-357, 2024.
28. A. Dhiman et al., "Sentiment Classification in Hindi Text using Hybrid Deep Learning Method," Int. J. of Information Technology, pp. 1-17, 2024.
29. D. S. Kulkarni and S. S. Rodd, "Sentiment Analysis in Hindi—A Survey on the State-of-the-art Techniques," ACM Trans. Asian Low-Resour. Lang. Inf. Process., vol. 21, no. 1, pp. 1-46, 2021.
30. IIT Patna Movie Reviews Hindi Sentiment Analysis Dataset, Kaggle, 2023. [Online]. Available: https://www.kaggle.com/code/khushipitroda/iit-patna-movie-reviews-hindi-sentiment-analysis
31. F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," J. Machine Learning Research, vol. 12, pp. 2825-2830, 2011.
32. J. D. Hunter, "Matplotlib: A 2D Graphics Environment," Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.
33. S. Singh, G. Divya, R. K, D. K. S, L. Vijayan, and U. Desai, "Automated Sentiment Analysis of Hindi Text using Machine Learning Techniques," in Proc. of the 9th International Conference on Inventive Computation Technologies (ICICT-2026), IEEE, 2026, pp. 2181-2186, doi: 979-8-3315-6917-4/26/$31.00. [Online]. Available: https://ieeexplore.ieee.org
34. https://github.com/satyapal2240/paper1_rbpu-Proposed-automated-sentiment-analysis-of-hindi-text-using-ML/blob/main/Paper1_rbpu-proposed-automated-Sentiment_Analysis-of-hindi-text-using-ML.ipynb