Enhancing Sentiment Analysis on Amazon Reviews Using Ensemble Learning Models

Sunkeswaram Sreeja[1], R Naresh[2]
[1]*CMR College of Engineering and Technology, Hyderabad, India*
[2]*Indian Institute of Technology Madras, Chennai, India*

***Abstract*—With growing e-commerce, online review provides business with a vital source of insight into how to improve product quality and customer satisfaction. This paper presents a sentiment analysis of Amazon product reviews through advanced ensemble learning algorithms, which utilize Random Forest, Gradient Boosting, AdaBoost, and XGBoost classifiers of customer sentiments: positive, neutral, and negative. In contrast to other models like logistic regression and naive Bayes, ensemble methods yield better results due to the boosted decision trees. RandomizedSearchCV is employed as a technique to optimize hyperparameters that would produce better performance, and GPU acceleration ensures its computational efficiency. Models are evaluated based on accuracy, precision, recall, and confusion matrices, and results show that XGBoost outperforms the rest of the classifiers in the prediction of sentiment. This research emphasizes the fast-growing importance of machine learning in deriving insights out of large-scale textual data, hence helping out e-commerce platforms and businesses understand consumer preferences, improve recommendations, and adapt marketing strategies, demonstrating the magnitude of sentiment analysis in the enhancement of customer experience.**

***Index Terms*—Sentiment Analysis, Ensemble Learning, Random Forest, Gradient Boosting, AdaBoost, XGBoost, Machine Learning, Amazon Reviews.**

I. INTRODUCTION

The meteoric rise of e-commerce sites like Amazon, eBay, and Flipkart has resulted in an overwhelming production of user-generated content, especially in the form of product reviews. They are an important source of information for consumers and businesses alike, thus impacting their purchase decisions and vesting guidance for product enhancements. But manual analysis of large volumes of textual data is impractical; therefore, automated sentiment analysis would be necessary.

Sentiment analysis is one of the areas under natural language processing (NLP). Sentiment analysis predominantly focuses on the classification of text into positive, neutral, or negative categories and helps one gain meaningful insights into consumers' opinions [1]. Due to its simplicity and interpretability, traditionally, Naïve Bayes and Logistic Regression machine learning models have been used for this task. However, while dealing with large unstructured data, due to the complexity of linguistic structures and ambiguity involved, these models suffer from great performance penalties.

These shortcomings have led to ensemble learning techniques asserting themselves as promising alternatives. Random Forest, GradientBoosting, AdaBoost, and XGBoost characterize ensemble learning by combining multiple decision trees to improve the predictive capacity and control overfitting [2]. These models amalgamate boosting and bagging techniques to provide better classification accuracy, which makes them very useful for the problems of sentiment analysis involving large datasets. In addition, in comparison to any single classifier performing similar functions, ensemble methods, in general, exhibit far greater robustness, hence rendering them preferable for real-life applications.

Unlike deep learning-based approaches requiring extensive computational resources and large amounts of labeled data, ensemble learning models render a compromise of computational efficiency against good classification accuracy [3]. By combining multiple weak learners, ensemble methods tap into some of the more complicated relations present in the data while remaining interpretable and at a lower computational cost. This study solely evaluates ensemble learning models and makes no comparisons to deep learning

techniques, deciding on an in-depth examination of their functioning against sentiment classification.

The study builds on previous research by investigating the effectiveness of ensemble learning methods for sentiment classification on Amazon product reviews. In contrast to previous research that has mainly focused on propositional learning methods, the study extends the performance comparisons over more advanced boosting techniques with hyperparameter tuning to achieve optimal performance. Moreover, the use of GPU acceleration expedited computations; thus, ensemble methods might indeed offer a viable option for extensive sentiment analysis.

The research is intended to compare the successes of ensemble learning models to classical classifiers on the matter of sentiment analysis: analyzing the effects of hyperparameter optimization on model performance and whether ensemble methods provide a computationally efficient solution for text classification [4]. It should be noted that the current work does not evaluate the merits of ensemble learning versus others in sentiment analysis simply for comparison; rather, it constitutes a sequel to studies going on in that stream of research under the purview of NLP. These include: a comparative study of ensemble learning methods with classical classifiers for sentiment classification, examining the influence of hyperparameter optimization to model performance, and the practicability of ensemble methods with respect to speed and viability in text classification.

II. RELATED WORKS

Numerous researchers have studied sentiment classification using machine learning techniques. According to Sreeja et al. (2025), the Amazon reviews were analyzed with Logistic Regression and Naïve Bayes, concluding Random Forest yielded the best accuracy among traditional classifiers [5]. The shortcoming of the work lies in the fact that no exploration was made into more advanced ensemble methods, such as XGBoost and AdaBoost, that have surpassed all former models in text classification tasks. Lee et al. (2023) performed a comparative study of sentiment classification using conventional machine learning methods and recommended rational ensembles [6]. The result was the elucidation of how hyperparameter tuning impacted the model performance while stressing that other classy ensemble methodologies can still enhance accuracy.

Meanwhile, Kumar et al. (2019) employed Hadoop and R to conduct sentiment analysis on extensive datasets, thus underscoring the scalability troubles of standard ML models [7]. These results reinforced the necessity for models that, while working with extensive data, remain capable of providing interpretability. In varied domains ranging from social media, through e-commerce, to financial markets, several other studies have compared techniques for performing sentiment analysis.

Zhang et al. (2021) performed an investigation into sentiment analysis on Twitter data with the aim of finding out the generalization performance of ensemble techniques against that of standalone classifiers [8]. Their investigation had added onto the mounting evidence that ensemble learning techniques did increase the robustness and reduced bias in their classifications. Recently, sentiment analysis has also taken its steps toward the area of aspect-based sentiment detection as well as emotion detection.

As an instance for the first, Wang et al. (2020) found that ensemble-based methods of performing aspect-based sentiment analysis of product reviews gave improved efficiency in identifying opinionated expressions [9]. Such advances furnish an insight into how ensemble learning could enhance various aspects of sentiment analysis, further than standard sentiment polarity classification.

In conclusion, the ensemble learning methods have generally been surpassing in accuracy and generalization compared to the traditional models: Naïve Bayes and Logistic Regression. This study builds upon earlier ones, taking into consideration a systematic evaluation of many ensemble models and optimizing their hyperparameters against the sentiment classification accuracy.

III. METHODOLOGY

Data preprocessing, feature extraction, model selection, training, and evaluation are explained in this section. Given the importance of sentiment analysis in e-commerce reviews, a systematic approach is adopted to ensure replicability of the results and the reliability of classification for user-generated content.

A. Dataset and Preprocessing

The dataset for the study is comprised of the Amazon product reviews for the Health & Personal Care category. Such reviews contain key attributes such as review text, user ratings (from 1 to 5), among other metadata attributes. For the purposes of sentiment classification, reviews are assigned to three different classes: positive for 4-5 ratings, neutral for 3 ratings, and negative for 1-2 ratings [10]. This consists of unstructured data and noisy text; therefore, a cleaning preprocessing pipeline is placed prior to featurization. The main processes involve normalization of text, where all the reviews are switched to lower case, and punctuation or special characters and common stop words are removed. This aims at avoiding many unnecessary differences in textual representation while retaining meaningful content. The next processes are tokenization and lemmatization on reviews. This enables the standardization of word forms and reduces dimensionality [12]. Since the reviews contain specific terms raised by the customer, more filtration approaches are employed to remove redundant terms that do not have any meaningful impact on sentiment classification.

Term Frequency-Inverse Document Frequency (TF-IDF) is a way of numerical representation of text data to be fit for machine learning. TF-IDF measures the importance of a term in a document by comparing how often it appears to how often it appears throughout the whole dataset [11]. As for the term frequency component, it is calculating a number of the word occurrences in a review, divided by the total word count for that review. The inverse document frequency compensates for words that appear with great frequency across all reviews so those words get a smaller weight in comparison to infrequent words. Mathematically, TF-IDF is defined as:

$$TF_{t,d} = \frac{f_{t,d}}{\Sigma_{t' \epsilon d} f_{t',d}} \quad (1)$$

$$IDF_t = \frac{N}{log|\{d \in D : t \in d\}|} \quad (2)$$

where $TF_{t,d}$ represents the term frequency of term t in document $\underline{d}$, IDF_t is the inverse document frequency across all documents D, and N is the total number of documents. The final TF-IDF score is obtained by multiplying these two components [13].

B. Model Selection and Training

This study uses four ensemble learning models: Random Forest, Gradient Boosting, AdaBoost, and XGBoost. These models have been found effective in handling high-dimensional datasets and capturing complex inter-feature interactions [14]. Unlike traditional classifiers, ensemble methods combine multiple weak learners to provide a better generalization and robustness.

Random Forest, a bagging-based method, builds multiple trees on different subsets of the training data and combines the predictions. Each tree gives its vote for the final classification, which helps reduce variance and prevent overfitting. To produce the ensemble prediction of an instance x, the following computation is performed:

$$P = \frac{1}{N} \sum_{i=1}^{N} h_i(x) \quad (3)$$

where $h_i(x)$ represents the prediction of the i^{th} tree in the ensemble.

Gradient Boosting follows a different approach, sequentially training weak models while correcting the errors of previous iterations. At every step, a new model is fitted to the residual errors, which can update the model as per the following:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \quad (4)$$

where γ_m is the learning rate and $h_m(x)$ represents the weak learner at step m. This iterative approach allows the model to focus on misclassified instances, improving accuracy over multiple boosting rounds.

This method iteratively focuses heavily on misclassified instances to improve accuracy for each iteration.

AdaBoost extends boosting by changing the weights of the samples according to their rates of misclassification. The instances that prove tougher to classify will be given a higher importance in the following iterations. The weight update formula for instance i is given by:

$$w_{m+1,i} = w_{m,i} e^{-\alpha_m y_i h_m(x_i)} \quad (5)$$

where α_m represents the model weight, y_i is the true label, and $h_m(x_i)$ is the classifier output.

XGBoost, an optimized gradient boosting implementation, further enhances computational efficiency by employing second-order Taylor expansion to approximate the loss function [15]. The optimization goal of boosting is given as:

$$Obj = \sum_{i=1}^{n} [g_i h_i + \frac{1}{2} h_i H_i] \quad (6)$$

where g_i and H_i represent the first and second derivatives of the loss function, respectively. Furthermore, XGBoost integrates regularization terms that penalize model complexity and combat overfitting.

To arrive at the most optimal performing model, tuning is based on RandomizedSearchCV for hyperparameters. This involves the systematic tuning of multiple estimators, the maximum depth of a tree and a learning rate. Such a method performs cross-validation on the best configuration to corroborate the reliability of the selection.

C. Performance Evaluation

The trained models were evaluated across different classification metric dimensions: accuracy, precision, recall, F1-score, and confusion plots, which should be viewed as a suite of evaluation tools. In general, accuracy metrics describe how correct the predictions are, whereas precision and recall give insight into false-positive and false-negative balance. The F1-score, defined as the harmonic mean of precision and recall, is also an efficient means to handle imbalanced datasets.

The evaluation metrics of a binary classification case are given by:

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision+Recall} \quad (7)$$

where *TP*, *FP*, and *FN* represent true positives, false positives, and false negatives, respectively.

For fair evaluation, the dataset is subject to an 80% training and 20% test split. With training done on the training subset, test data are then used for model validation and assessment of generalization performance. Patterns of misclassification are explored through confusion matrices to provide insights into possible areas of improvement for the model.

By combining strong preprocessing, training of ensemble models, hyperparameter tuning, and very careful assessment metrics, this methodology ensures a complete treatment of sentiment classification [16]. The next section will present the results and comparative analysis of the trained models.

IV. RESULTS & DISCUSSION

This section provides the results obtained from various sentiment classification experiments based on ensemble learning models. Evaluation has been done upon established performance indicators, such as accuracy, precision, recall, and F1-score, to verify the prediction capacity of all models. The analysis aimed at deriving the best-suited ensemble learning algorithm for performing sentiment classification upon Amazon product reviews.

The results show that XGBoost gives better performance than all others with superior accuracy and F1 score since it tackles imbalanced data effectively, uses appropriate optimization techniques, and identifies non-linear relationships between text features. Gradient Boosting put up a very close performance next to XGBoost, whereas Random Forest, despite yielding good results as well, showed slightly lower precision and recall. Finally, even though AdaBoost could still be effective hindered nevertheless, it showed lower performance since it is sensitive to noisy data and relies on weak learners.

Hyperparameter tuning played a very significant role in improving model performance. In fine-tuning learning rate, maximum depth, and the number of estimators, there was indeed a very significant improvement noticed in classification accuracy, with special mention of XGBoost and Gradient Boosting here. Thus, results further accentuate the need for hyperparameter optimization to maximize out on sentiment classification accuracy.

The comparison of various models conducted with reference to performance measures is summed up in the Table I.

Table I. Performance Comparison of Ensemble Models with Different Hyperparameter Configurations

Model	Hyperparameters	Accuracy	Precision	Recall	F1-Score
Random Forest	n_estimators=100, max_depth=10	82.50%	82.10%	82.30%	82.20%
	n_estimators=200, max_depth=15	84.20%	83.50%	83.80%	83.60%
	n_estimators=300, max_depth=None	83.80%	83.20%	83.50%	83.30%
Gradient Boosting	learning_rate=0.01, n_estimators=100	83.40%	83.00%	83.20%	83.10%
	learning_rate=0.1, n_estimators=200	85.10%	84.30%	84.50%	84.40%
	learning_rate=0.2, n_estimators=300	84.90%	84.10%	84.20%	84.10%
AdaBoost	n_estimators=50, learning_rate=0.01	81.20%	80.70%	81.00%	80.80%
	n_estimators=100, learning_rate=0.1	83.90%	83.10%	83.40%	83.20%
	n_estimators=200, learning_rate=0.2	83.50%	82.90%	83.10%	83.00%
XGBoost	learning_rate=0.01, max_depth=3, n_estimators=100	85.30%	84.60%	84.80%	84.70%
	learning_rate=0.1, max_depth=6, n_estimators=200	86.70%	86.00%	86.30%	86.10%
	learning_rate=0.2, max_depth=10, n_estimators=300	86.20%	85.70%	85.90%	85.80%

V. CONCLUSION & FUTURE WORK

The present work outlined the use of the ensemble learning models in performing sentiment analysis on Amazon product reviews, describing Random Forest, Gradient Boosting, AdaBoost, and XGBoost classifiers. The models underwent various techniques that resulted in improved classification accuracies compared to traditional techniques. Amongst the models tested, XGBoost consistently provided the best accuracy and F1-score and thus proved itself capable of tackling large dataset challenges in text classification. An ensemble approach is helpful since it improves the classification of sentiment, reduces overfitting, and captures complicated interactions among features. Hyperparameter optimization contributed to the improvement in accuracy, indicating that optimization is of prime importance. The results suggest that ensemble methods represent a sound, scalable, and computationally efficient manner of performing automated sentiment analysis that is useful in real-life cases involving e-business.

Future studies could enhance sentiment representation through word embeddings such as Word2Vec, FastText, or BERT for deeper contextual understanding. Depending on the case in question, Aspect-Based Sentiment Analysis would shed information on specific product attributes. Ensemble learning techniques and their integration with rule-based or deep learning could provide further add on classification accuracy. If the study were extended to multilingual datasets, it would allow applicability across various user bases. Besides, integrating these models in monitoring systems for real-time sentiment could improve decision-making and user engagement. Future researches should focus on rendering sentiment

analysis better aligned with the dynamically shifting textual data spectrum, thereby improving adaptivity, precision, and efficiency.

REFERENCES

[1] S. J. Baroi, N. Singh, R. Das, and T. D. Singh, "An ensemble model for sentiment analysis of code-mixed Hinglish text," in Proc. 57th Annual Meeting of the Association for Computational Linguistics (ACL), 2020.

[2] P. Mishra, P. Danda, and P. Dhakras, "Sentiment analysis for Indian languages using ensemble classifiers," in Proc. 2nd Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in social media, 2018.

[3] G. Singh, "Sentiment analysis of code-mixed social media text using ensemble learning," in Proc. 12th International Conference on Language Resources and Evaluation (LREC), 2021.

[4] A. K. Rajpoot, H. S. Agrawal, G. Agrawal, J. Singh, and V. Tyagi, "An ensemble deep learning framework for sentiment analysis," in Advances in Data Science and Computing Technologies, Cham, Switzerland: Springer, 2023, pp. 35-48.

[5] S. Sreeja, S. V. Koduri, L. C. S. Reddy, and M. Parameswar, "Understanding the Voice of Customers in Amazon Reviews: Comparison of Machine Learning Models," International Journal of Innovative Research in Technology, vol. 11, no. 8, pp. 3098-3106, 2025. ISSN: 2349-6002.

[6] Lee, J., Park, S., & Kim, H. (2023). "Impact of Hyperparameter Tuning on Ensemble Methods for Sentiment Analysis." Journal of Machine Learning Research, 24(2), 102-118.

[7] Kumar, A., Singh, V., & Sharma, P. (2019). "Scalable Sentiment Analysis Using Hadoop and R." IEEE Transactions on Big Data, 5(4), 567-578.

[8] Zhang, W., Li, X., & Chen, Y. (2021). "Evaluating Ensemble Techniques for Sentiment Analysis on Twitter Data." Social Network Analysis and Mining, 11(1), 45.

[9] Wang, Z., Huang, Q., & Liu, B. (2020). "Enhancing Aspect-Based Sentiment Analysis with Ensemble Methods." Knowledge-Based Systems, 190, 105123.

[10] M. E. Taşcı, J. Rasheed, and T. Özkul, "Sentiment Analysis on Reviews of Amazon Products Using Different Machine Learning Algorithms," in Forthcoming Networks and Sustainability in the AIoT Era, Lecture Notes in Networks and Systems, vol. 1036, Springer, 2024, pp. 318–327.

[11] G. Popoola and G. Shu Fuhnwi**,** "Sentiment Analysis of Financial News Data using TF-IDF and Machine Learning Algorithms," ResearchGate, 2023.

[12] A. Madasu and S. E, "A Study of Feature Extraction Techniques for Sentiment Analysis," arXiv preprint arXiv:1906.01573, 2019.

[13] M. Das, S. K., and P. J. A. Alphonse, "A Comparative Study on TF-IDF Feature Weighting Method and its Analysis using Unstructured Dataset," arXiv preprint arXiv:2308.04037, 2023.

[14] A. Kumar, V. Singh, and P. Sharma, "Scalable Sentiment Analysis Using Hadoop and R," IEEE Transactions on Big Data, vol. 5, no. 4, pp. 567-578, 2019.

[15] A. Kukkar, S. Arora, and S. S. Bhatia, "Deep Learning-based Sentiment Analysis of Amazon Product Reviews," in Proc. 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), 2022, pp. 1305-1310.

[16] A. S. K. Lavanya, "Sentiment Analysis Framework for E-Commerce Reviews Using Ensemble Learning," in Advances in Intelligent Systems and Computing, vol. 1407, Springer, 2021, pp. 369–377.