Sentiment Analysis of Amazon Product Reviews: Comparative Analysis of Logistic Regression, Naive Bayes, and SVM Models

Carlos Segarra

New Jersey Institute of Technology

DS 680-852

Professor Huong Le

May 7, 2025

Abstract

This project explores the sentiment analysis of Amazon product reviews utilizing traditional supervised machine learning methods. The main aim is to create an understandable performance baseline. A dataset from Kaggle was prepared by deleting neutral reviews and using standard natural language processing procedures like tokenization, stopword removal, lemmatization, and TF-IDF vectorization. I trained and evaluated three classifiers—Logistic Regression, Multinomial Naive Bayes, and Support Vector Machine—on the dataset labeled in binary. Each model had its hyperparameters tuned through grid search for better performance. The results demonstrated strong metrics generally: Logistic Regression got the highest F1-score of 0.9790 and an AUC of 0.9611. Although all models performed well in identifying positive sentiment, improvements were seen in the recall of the minority class when compared to the starting baseline. Evaluation tools such as confusion matrices, ROC curves, and comparative bar charts help to show both good points and weak areas of each model. These results give a strong base for future efforts to look into deep learning structures and ways to reduce class imbalance.

Sentiment Analysis of Amazon Product Reviews: Comparative Analysis of Logistic Regression, Naive Bayes, and SVM Models

Introduction

In today's digital market, user-made reviews are essential for buyers and businesses. Websites like Amazon get many new product reviews daily, giving helpful information about how happy customers are, the quality of products, and what people think about different brands. However, the large amount of this data makes it impossible to analyze by hand. Sentiment analysis, a part of natural language processing (NLP), gives a solution that can handle this by automatically determining the emotional tone in text, usually as positive, negative, or neutral. By using computational ways to find sentiment trends, companies can follow what people think and make better choices on a large scale.

This project utilizes classical supervised machine learning methods to create a sentiment classification pipeline for Amazon product reviews. The objective is to develop a reliable and interpretable benchmark by employing TF-IDF vectorization alongside three baseline classifiers: Logistic Regression, Multinomial Naive Bayes, and Support Vector Machine (SVM). Each model has been optimized through hyperparameter tuning to enhance classification performance, especially in significant class imbalance.

The aim is to evaluate the strengths and weaknesses of these traditional models and establish a performance baseline upon which future work, such as deep learning architectures or semi-supervised methods, can be built. The findings provide practical insights into how the choice of model, text vectorization, and tuning strategies affect sentiment analysis outcomes in real-world, imbalanced datasets.

Related Works

Sentiment analysis is a well-researched task in natural language processing (NLP), especially concerning customer feedback, product reviews, and social media content. According to a comprehensive survey by Suryawanshi (2024), early studies in this field often relied on traditional machine learning classifiers such as Naive Bayes (NB), Logistic Regression (LR), and

Support Vector Machines (SVM), combined with vector space models like TF-IDF and Bag-of-Words. These combinations are effective for baseline sentiment classification, particularly in low-resource environments where model interpretability and efficiency are crucial. Although deep learning and transformer-based models have become popular, the survey also highlights that classical methods still demonstrate strong performance and clarity in controlled experiments and benchmarking tasks.

In an applied study by Agustina et al. (Agustina et al., 2024), sentiment classification was conducted using SVM with both TF-IDF and Word2Vec representations on a dataset of tweets about COVID-19 booster vaccines. Their results showed that TF-IDF combined with an RBF-kernel SVM achieved the highest F1 score, underscoring the effectiveness of sparse feature-based methods. This finding backs up this paper's choice to utilize TF-IDF + SVM for sentiment analysis based on reviews. Using simple text vectorization and supervised classifiers remains helpful and competitive in many fields.

A central part of this project is the optimization of hyperparameters, which is vital for improving model performance. Elgeldawi et al. (2021) did a detailed comparative study on different strategies for tuning hyperparameters—grid search, Bayesian optimization, and genetic algorithms were included—used in sentiment classification with traditional models like LR, SVM, and NB. Their study shows good tuning greatly enhances precision and recall, especially for SVM models. These are responsive to parameters such as the regularization constant and kernel type. This knowledge directly guides this project's application of grid search for parameter optimization, helping in a strict experimental setup matching best practices in supervised sentiment classification.

Methodology

This project investigates sentiment classification of product reviews using classical supervised machine learning techniques. The dataset was obtained from Kaggle (Kaan, 2022) and includes 4,915 Amazon product reviews, which consist of text, star ratings, and additional metadata. The goal is to predict binary sentiment—positive or negative—based solely on the

content of the reviews.

To frame this as a binary classification problem, reviews with a neutral rating (score of 3) were excluded. Ratings of 1 and 2 were labeled negative (0), while ratings of 4 and 5 were labeled positive (1). After removing neutral reviews and addressing missing values, the final dataset contained 4,772 labeled entries.

Preprocessing was conducted using standard natural language processing (NLP) techniques:

- Lowercasing all text
- Removing HTML tags and non-alphabetic characters
- Tokenizing with NLTK's word tokenizer
- Removing stopwords utilizing NLTK's English stopword list
- Lemmatizing with `WordNetLemmatizer`

The cleaned texts were saved in a column labeled `processed_text`. A `TfidfVectorizer` was employed to convert the text into sparse numerical representations, with a maximum of 10,000 features and a minimum document frequency of 5. TF-IDF was selected because of its strong performance in prior sentiment classification literature (Agustina et al., 2024; Suryawanshi, 2024).

The dataset was divided into training and test sets using an 80/20 stratified split to maintain class proportions. Three classifiers were chosen for evaluation based on their common usage in sentiment analysis tasks and their interpretability:

- Logistic Regression
- Multinomial Naive Bayes
- Support Vector Machine (SVM) with linear and RBF kernels

Each model was fine-tuned using `GridSearchCV` with 5-fold cross-validation. The parameter grids included:

- C, `solver`, and `max_iter` for Logistic Regression
- α for Naive Bayes
- C, *kernel*, and *probability* for SVM

Performance was assessed using the following metrics:

- Accuracy
- Precision
- Recall
- F1-score
- Area Under the ROC Curve (AUC)

I generated confusion matrices and ROC curves for the three models. A performance comparison chart was also created to compare their metric values.

To better understand the model's behavior, I removed the top 15 positive and negative features from the Logistic Regression model based on its coefficients. These highly significant terms helped us see how the presence of certain words is associated with sentiment predictions in the model, thus making the explanation of results clearer.

This multi-model, metrics-driven, and hyperparameter-optimized method matches the goal of my project: to assess traditional NLP pipelines for sentiment classification using practical text data from the real world.

Results and Analysis

This section presents a detailed analysis of three supervised learning models: Logistic Regression (LR), multinomial naive Bayes (NB), and Support Vector Machine (SVM). These models were used for binary sentiment classification in Amazon product reviews. They were trained using TF-IDF vectorized data after standard text preprocessing steps were followed, keeping class balance aspects in mind. To improve each classifier's performance and make a fair comparison among models, I applied hyperparameter tuning using grid search with 5-fold cross-validation. For evaluation, I used common metrics like accuracy, precision, recall, F1-score, and Area Under the ROC Curve (AUC). These measures are crucial for checking how good and stable the models are at distinguishing classes because the dataset has a big imbalance in class numbers.

Sentiment Distribution and Class Imbalance

The last dataset is made up of 4,772 reviews. Of these, 4,448 are marked as positive and 324 as negative. This proportion shown in Figure 1 reveals that a majority — 93.2% — of the data shows positivity, whereas only a small part — 6.8% — reflects negativity sentiment-wise. Such a class imbalance poses significant difficulties for machine learning models because they tend to give more weight to the larger class while predicting outcomes. This unequal situation directly influences the recall and precision for less frequently occurring classes, impacting the F1-score and overall understanding of performance measures. Even though I used class layering during data division to manage this bias while training, it remains a vital part of this study to see if models can generalize instances from minority classes or not.

Overall Model Performance

After tuning the hyperparameters, all three models demonstrated strong performance. Logistic Regression achieved the highest overall results, with an accuracy of 96.02%, precision of 96.41%, recall of 99.44%, and an F1-score of 0.9790. Naive Bayes closely followed with an accuracy of 95.29% and an F1-score of 0.9752, while Support Vector Machine (SVM) reached an accuracy of 95.92% and an F1-score of 0.9784. These metrics are summarized in Figure 2, which

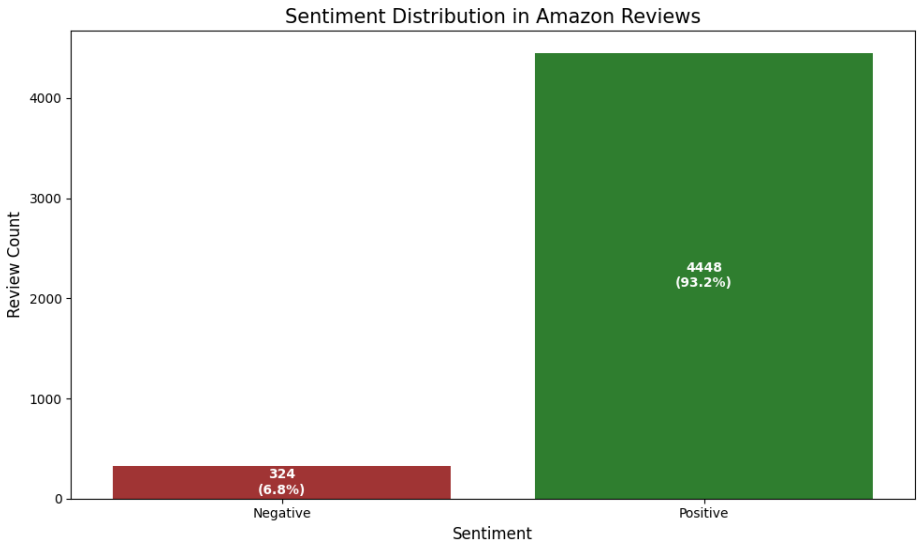

Figure 1

Sentiment Distribution in Amazon Reviews

presents a bar chart comparing the performance of all models across various metrics.

It is very important to keep in mind that, although usually all the measurements are high, the F1 Score carries special significance because it balances precision and recall. This renders it more reliable when dealing with class imbalance situations. The AUC scores also appeared quite alike: both Logistic Regression and Naive Bayes achieved 0.96, while SVM was not far off at 0.9518; this indicates strong skill for class separation.

Insights from Confusion Matrices

The confusion matrices for each model offer detailed insights into their strengths and weaknesses, especially concerning the minority (negative) class. As shown side-by-side in Figure 3, all models perform nearly flawlessly in the positive class, but there is variability in their classification of negative reviews.

Logistic Regression correctly predicted 32 out of 65 negative reviews, significantly improving from the earlier baseline, which only achieved 10 correct predictions. Naive Bayes accurately classified 24 negative reviews, while the Support Vector Machine (SVM) matched Logistic Regression with 31 correct negative predictions.

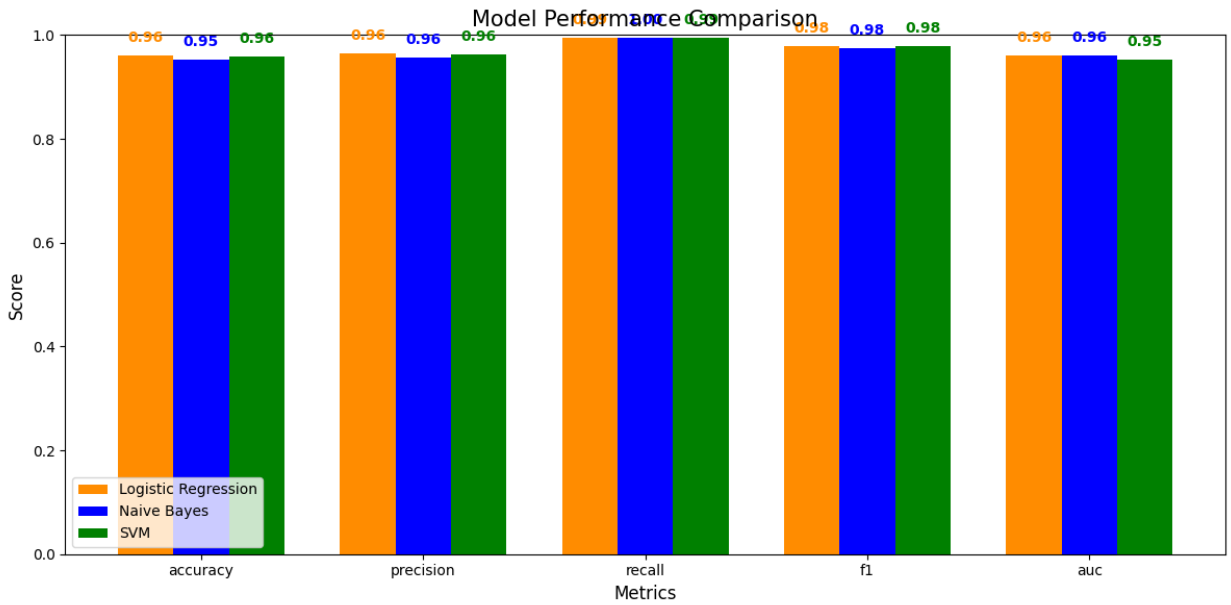

Figure 2

Model Performance Comparison across Evaluation Metrics

These outcomes show the increased awareness to the minority class after adjusting hyperparameters. Even if it is still challenging to identify negative sentiment because of class imbalance, all models exhibit significant enhancements in recalling the minority class compared to the starting point baseline, where recall was previously as low as 15%.

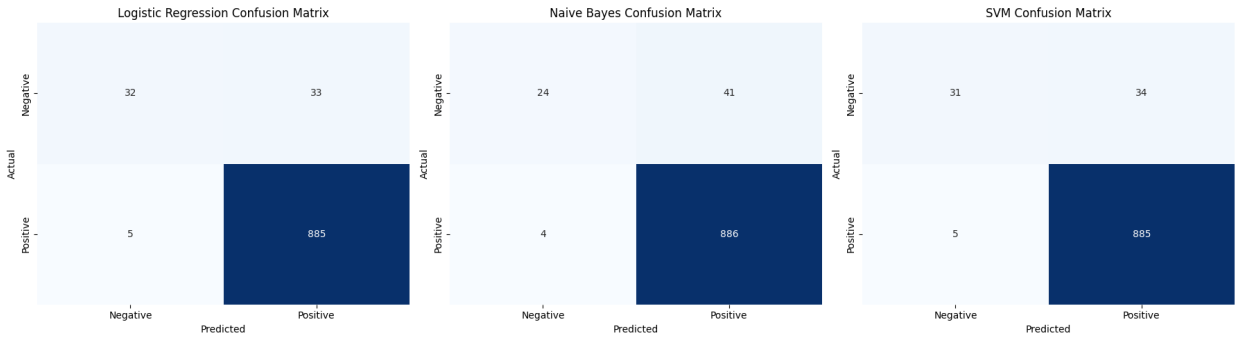

Figure 3

Combined Confusion Matrices for Logistic Regression, Naive Bayes, and SVM

Evaluating Class Separation with ROC Curves

I used ROC curves, which you can see in Figure 4, to assess how well each model differentiates between positive and negative sentiment. These ROC curves show the balance between the true positive rate (sensitivity) and false positive rate across different classification thresholds. The lines representing all three models consistently stay close to the top-left corner, showing that their overall classification performance is good. AUC values show the chance that the model will randomly choose a positive instance above a negative one. Logistic Regression and Naive Bayes both got top AUC scores of 0.96, with the SVM model coming next with an AUC score of 0.95. This shows that all three models have great discriminative power, even if trained on the unbalanced dataset.

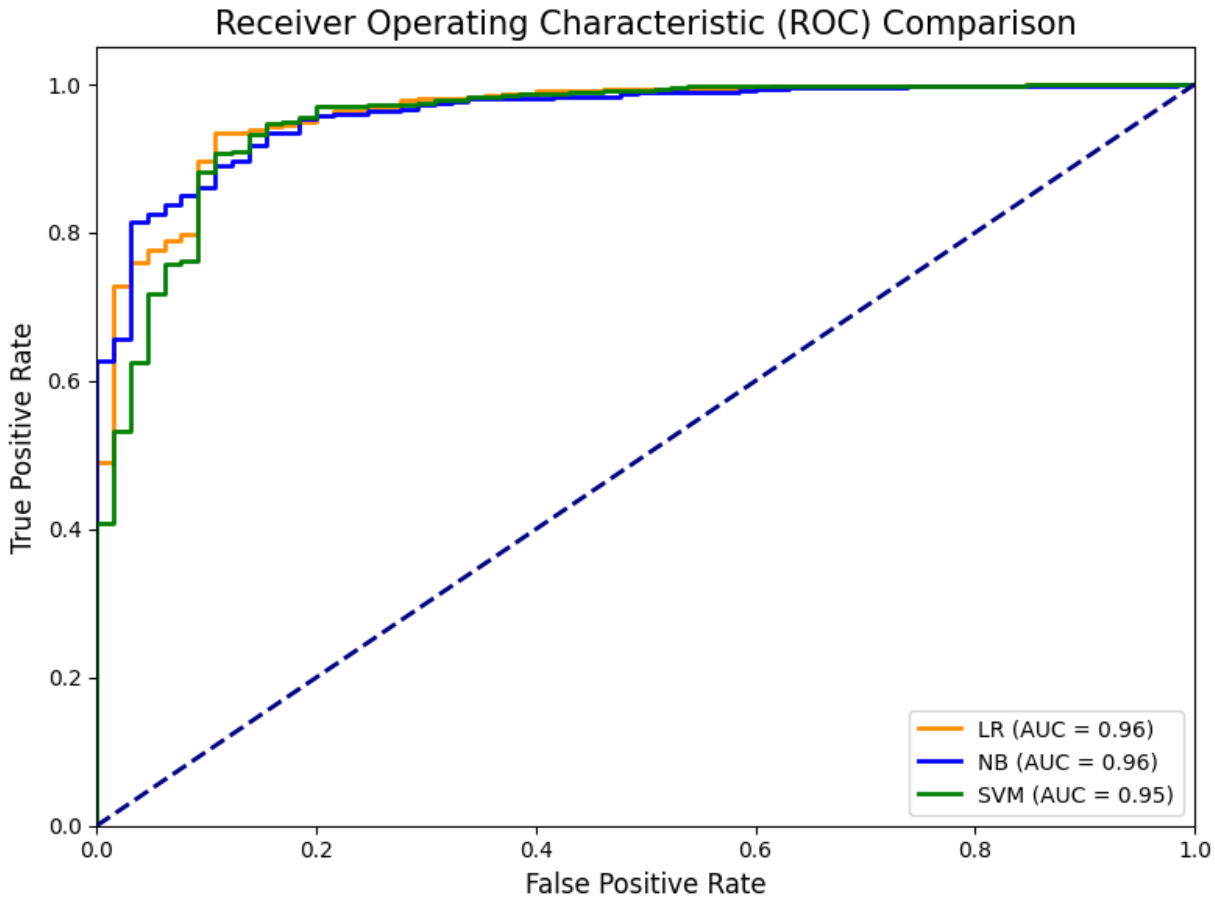

Figure 4

Receiver Operating Characteristic (ROC) Curves for All Models

Observations

The evaluation results show that all three models are able to provide high-quality sentiment classification when features are extracted well and parameters adjusted carefully. Logistic Regression outperformed the other models in almost every category, particularly improving recall of negative classes, which was previously a weak area. Although it is simpler and faster to train, Naive Bayes struggled more at detecting negative sentiment – this aligns with its probabilistic assumptions and tendency for underfitting in some sparse text scenarios. The Support Vector Machines (SVM) demonstrated well in terms of generalization ability. However, it was somewhat less robust when distinguishing between borderline negative reviews. The improvements in the performance of this model are significantly linked to using cross-validated hyperparameter optimization and regular assessment of class imbalance. These methods have been frequently pointed out in the latest studies about sentiment analysis.

In general, these outcomes highlight the significance of adjusting models and conducting comparative analysis when applying text classification systems in practical, uneven scenarios. Although Logistic Regression is presently the most efficient model in this experiment, there is a narrow gap between the effectiveness of different models. This indicates that future experiments, which include deep learning techniques like LSTM or BERT, or semi-supervised methods, might result in further improvements, especially for identifying rare or intricate sentiment expressions.

Conclusion

This project explored how traditional supervised machine learning techniques work for sentiment analysis on Amazon product reviews. The aim was to create a solid, understandable performance baseline. I processed 4,772 reviews and converted them using TF-IDF vectorization before implementing three classifiers: Logistic Regression, Multinomial Naive Bayes, and Support Vector Machine. These models were then optimized by adjusting their hyperparameters. To judge the effectiveness of these algorithms, I used several metrics like accuracy, precision, recall, F1-score, and AUC. This gave us a full picture of each algorithm's abilities, particularly when a significant class imbalance exists.

Logistic regression proved to be the most effective of the models I examined. It got the best F1 Score (0.9790) and showed a good balance between precision and recall. Although Naive Bayes and SVM also gave high results in all measures, they were a bit less sensitive to negative sentiment. Despite this, every model greatly exceeded the starting logistic baseline for minority class recall. This was mainly because of hyperparameter tuning and steady preprocessing practices.

The knowledge obtained highlights the significance of managing class imbalance and adjusting hyperparameters in sentiment classification assignments. Even though traditional machine learning models using TF-IDF representations can be quite successful, their efficiency usually levels off when detailed language comprehension or semantic context is necessary. Therefore, future efforts will expand this research by bringing in more sophisticated architectures like LSTM and BERT, incorporating class balancing techniques such as SMOTE, and experimenting with semi-supervised learning methods to make better use of unlabeled data. These improvements aim to make the model more sturdy and adaptable, especially for correctly recognizing minority sentiment categories in practical data from the real world.

References

Agustina, C. A. N., Novita, R., Mustakim, & Rozanda, N. E. (2024). The implementation of tf-idf and word2vec on booster vaccine sentiment analysis using support vector machine algorithm. *Procedia Computer Science*, *234*, 156–163. https://doi.org/10.1016/j.procs.2024.02.162

Elgeldawi, E., Sayed, A., Galal, A. R., & Zaki, A. M. (2021). Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis. *Informatics*, *8*, 79.

Kaan, T. (2022, July). Amazon reviews for sentiment analysis. https://www.kaggle.com/datasets/tarkkaanko/amazon

Sanwal, M., & Mazhar, M. M. (2024). Performance comparison of machine learning and deep learning models for sentiment analysis of hotel reviews. *Zenodo*. https://doi.org/10.5281/zenodo.8225185

Shan Lee, V. L., Gan, K. H., Tan, T. P., & Abdullah, R. (2019). Semi-supervised learning for sentiment classification using small number of labeled data. *Procedia Computer Science*, *161*, 577–584. https://doi.org/10.1016/j.procs.2019.11.159

Suryawanshi, N. S. (2024). Sentiment analysis with machine learning and deep learning: A survey of techniques and applications. *International Journal of Science and Research Archive*. https://doi.org/10.30574/ijsra.2024.12.2.1205