SSRG International Journal of Computer Science and Engineering
ISSN: 2348–8387 / https://doi.org/10.14445/23488387/IJCSE-V12I15P102

Volume 12 Issue 5, 10-18, May 2025
© 2025 Seventh Sense Research Group®

Original Article

Performance Analysis of Naïve Bayes and Stochastic Gradient Descent-based SVM for Sentiment Analysis

Shrushti Jeetendra Vadher

[1]*Department of Computer Engineering, Gandhinagar Institute of Technology, Gandhinagar, Gujarat, India.*

[1]*Corresponding Author : shrushtivadher30@gmail.com*

Received: 09 March 2025 Revised: 14 April 2025 Accepted: 02 May 2025 Published: 17 May 2025

Abstract - *This paper explores the effectiveness of sentiment classification on food review dataset, mainly focusing on Naïve bayes and Stochastic Gradient Descent-based Support Vector Machine (SGD-based SVM). The findings highlight the performance of both models and the impact of data preprocessing methods, as sentiment analysis is necessary for natural language processing and business customer services. The dataset obtained from Amazon on food review underwent preprocessing using Term Frequency-Inverse Document Frequency (TF-1DF) vectorization to transform textual data into numerical representations and Synthetic Minority Oversampling Technique to rectify class imbalances, ensuring fair and robust evaluation. The evaluation metrics demonstrate that SGD-based SVM has performed better than Naïve bayes with 84% and 79.9% accuracy, respectively. It can be observed that the SGD-based performs better as compared to the naïve bayes model.*

Keywords - *Class imbalance, Naïve bayes, Natural language processing, Sentiment analysis, Support Vector Machine.*

1. Introduction

Sentiment Analysis plays an important role in understanding customer opinions. The pervasive influence of online platforms on consumer behaviour has amplified the significance of sentiment analysis, a critical tool for deciphering customer opinions and gauging market trends [1]. E-commerce platforms, like blogs, reviews, and comments with vast quantities, necessitate automated sentiment analysis methods to extract actionable insights [2]. Sentiment analysis, also known as opinion mining, has become an indispensable technique for businesses and organizations seeking to understand public perception and make data-driven decisions [3]. Conventionally, sentiment analysis involves determining whether an expressed opinion is positive, negative, or neutral, providing a basis for understanding attitudes toward products, services, or topics [4]. The ability to automatically determine the sentiment of online reviews has numerous applications, including market research, brand monitoring, and customer service improvement [5]. By analysing the sentiments expressed in customer reviews, companies can gain a deeper understanding of their customers' needs and preferences, which can inform product development, marketing strategies, and customer support initiatives [6].

Roughly 97% of consumers consider online reviews when purchasing products [4]. However, the sheer volume of reviews can be overwhelming, making it difficult for consumers to sift through and extract relevant information [5] manually. Sentiment analysis offers a computational approach to automatically determine the polarity of opinions expressed in text, enabling efficient processing and summarization of vast review data [6]. This capability has proven invaluable in various domains, from market research and brand monitoring to customer service and political analysis [7]. Specifically, sentiment analysis helps understand customer preferences, identify areas for product improvement, and gauge a brand's or product's overall perception [8].

Sentiment analysis is pivotal in extracting valuable insights from extensive textual data, empowering businesses to understand customer sentiments, make informed decisions, and enhance their offerings [8]. Amazon's food review dataset categorises sentiments from customer feedback, pivotal for refining products, enhancing customer experiences, and gaining a competitive edge [7].

Customers and business need to understand the accurate product evaluation [8]. Due to the digital nature of online shopping, potential buyers often depend on previous customer reviews, making sentiment analysis a crucial tool for assessing the quality and desirability of products. The growing volume of user-generated content online necessitates the development of robust sentiment analysis techniques to extract meaningful insights.

Previous research has explored various machine-learning techniques for sentiment classification, and this paper adds to that body of knowledge. The sentiments ‘Positive’, ‘negative’, and ‘neutral’ imply that sentiment analysis delves into the emotional nuances behind opinions, recognizing that opinions are often emotional [9]. It is important to identify the target of any sentiment expressed clearly and to restrict the analysis to the immediate context of the target [10]. This study thoroughly compares two distinct machine learning methodologies, Naïve Bayes and Stochastic Gradient Descent-based Support Vector Machines, to classify the sentiments conveyed in Amazon food reviews, shedding light on their relative strengths and weaknesses in this specific application domain. Machine learning techniques have gained prominence in sentiment analysis because they can learn patterns from data and automatically make highly accurate predictions. Consumers consider online reviews for any product purchased, and any business owner also shows their customer reviews based on trust between the customer and the business owner. Genuine reviews with images also result in increased product sales.

Research Gap: While both naïve bayes and support vector machines have been widely explored for sentiment analysis, there are limited studies on food product review textual data using data preprocessing techniques like TF-IDF and SMOTE for better model performance. Most existing research explores deep learning models, but basic machine learning models are implemented here with support vector machine optimised with stochastic gradient descent.

Problem Statement: This study uses advanced text preprocessing and class balancing techniques to compare classic machine learning models naïve bayes and support vector machine optimised with stochastic gradient descent on a textual dataset.

Objective and Novelty: The main goal of this study is to compare and evaluate both widely used machine learning models. The impact of SMOTE on the dataset and SGD on classification accuracy are key analysis points. Unlike previous studies, this study integrates TF-IDF and SMOTE for data preprocessing and balancing. The results are visualized to highlight the strengths and limitations of this approach.

Among the different machine learning techniques, Naïve Bayes and Support Vector Machines have emerged as popular choices for text classification tasks, including sentiment analysis. Here, the Synthetic Minority Oversampling Technique is applied to address the challenge of imbalanced class distributions in sentiment analysis datasets. The present study addresses these issues by applying SMOTE to generate synthetic samples, ensuring equal class representation. The method generates new instances from existing minority class samples, which effectively increases the diversity of the training data and prevents the classifier from being biased towards the majority class [11]. Here, Amazon food reviews were categorized into positive, negative, and neutral sentiments, providing a comprehensive dataset for sentiment analysis. The ratio of positive classes to negative, neutral sentiments almost doubled, so this was applied to ensure fair representation across classes. This ensures the classifier is not biased toward the majority class and can effectively generalize to unseen data. Text cleaning is a critical step in the preprocessing pipeline, involving removing special characters and stop words to reduce noise and improve the quality of the input data. The equation of TF-IDF is tf(t,d) * idf(t, D) where tf(t,d) is the number of times term t appears in a document d and idf(t, D) is the number of documents in the corpus D [12]. Optimized hyperparameters in Naïve Bayes and hinge loss with balanced class weights and optimized learning rate in SGD-based SVM enhance the classification models' performance.

Stochastic Gradient Descent is an iterative optimization algorithm used to find the minimum of a function [11]. In the context of machine learning, it is employed to update the weights of a model to minimize a loss function [13]. Unlike Batch Gradient Descent, which computes the gradient using the entire dataset, SGD updates each data point's weights [14]. This makes it computationally efficient, particularly for large datasets, but also introduces more noise in the weight updates [15]. The "stochastic" nature arises from the randomness in selecting individual data points for each iteration [11]. For high dimensional non-convex optimization problems, variants of SGD are commonly used and are still being improved [16]. One of the primary advantages of Stochastic Gradient Descent is its ability to escape local minima due to the noise introduced by the random sampling of data points [17]. This can be particularly useful when dealing with non-convex loss functions, where getting stuck in local minima can lead to suboptimal solutions. However, this noise can also lead to oscillations around the minimum, requiring careful tuning of the learning rate to ensure convergence. In practice, mini-batch gradient descent is preferred over full-batch gradient descent because mini-batch gradient descent generalizes faster [14].

Support Vector Machines are a class of supervised learning algorithms used for classification and regression. SVMs aim to find an optimal hyperplane that separates data points of different classes with the most significant possible margin [18]. The margin is the distance between the hyperplane and the closest data points from each class, known as support vectors. The objective is to maximize this margin, which leads to better generalization performance [11]. The optimization problem involves finding the weights and bias of the hyperplane that maximizes the margin while minimizing the classification error. SVMs are effective in high-dimensional spaces and are relatively robust to outliers due to the use of support vectors. The SVM algorithm effectively addresses many classification problems using supervised learning techniques [19]. SVC training constructs a model that allocates new instances to one or more categories after analysing data and recognizing patterns [20].

When aligning Stochastic Gradient Descent with Support Vector Machines, SGD is the optimization algorithm to train the SVM model. The loss function in SVM is typically the hinge loss, which penalizes misclassified data points and encourages a large margin [21]. The hinge loss function is non-differentiable at specific points, which can challenge traditional gradient descent methods. SGD, however, can handle non-differentiable loss functions by using subgradients. The combination of SGD and SVM benefits large-scale datasets where traditional SVM training algorithms become computationally infeasible [11]. The core idea is to iteratively update the SVM model's weights using the gradient of the hinge loss calculated on a single or a small batch of data points. The objective function of SVM with hinge loss can be written as:

L (w, b) = frac {1}{2} ||w||^2 + C sum_{i=1} ^{n} max (0, 1 - y_i (w^T x_i + b)) (1)

Where (w) is the weight vector, (b) is the bias, (x_i) is the input data point, is the corresponding label, and is the regularization parameter. This equation encapsulates the core of the SVM optimization problem [31] [32].

Naïve Bayes is a probabilistic classifier based on Bayes' theorem with a "naïve" assumption of independence between features. Despite its simplicity, it is remarkably effective in many real-world applications, particularly in text classification and natural language processing. The algorithm calculates the probability of a given data point belonging to a particular class based on the probabilities of its features occurring in that class. Bayes' theorem provides a way to calculate the posterior probability from the likelihood and prior probabilities. The conditional independence assumption simplifies the calculation but may not hold in practice. Nevertheless, the Naïve Bayes classifier often performs well, especially when the independence assumption is approximately satisfied.

In Natural Language Processing, Naïve Bayes is commonly used for text classification, sentiment analysis, and spam detection tasks. The features are typically the words present in the text, and the classes are the categories to which the text belongs. The algorithm learns the probability of each word occurring in each class from the training data. The prior probability of each class is estimated from the proportion of documents belonging to that class in the training data. Given a document, the Naïve Bayes classifier calculates the probability of the document belonging to each class and assigns the document to the class with the highest probability. The Naïve Bayes classifier determines which class each document belongs to by mapping a collection of documents Dn to a category Ck based on predetermined criteria or traits [11]. Naïve Bayes is suitable to be used in sentiment analysis [22]. The classifier functions using the Bayes theorem, considering the variables' independence [23]. Even though the assumption is not always accurate, it is frequently successful in text categorization, even rivalling more sophisticated approaches like support vector machines [24].

The Naïve Bayes classifier estimates the likelihood of particular characteristics in text classification. These traits usually relate to the event category. The probability of each word appearing in each class is learned from the training data [25]. The Bayes theorem is applied to determine the likelihood that a document belongs to each class, and the document is assigned to the class with the highest probability. The assumption is that each feature makes an equal and independent contribution to the outcome. Bernoulli Naive Bayes, Multinomial Naive Bayes, and Gaussian Naive Bayes are the three main types of Naive Bayes classifiers. Bernoulli Naïve Bayes is used when features are binary (e.g., presence or absence of a word), Multinomial Naïve Bayes is used when features represent counts or frequencies (e.g., word counts in a document), and Gaussian Naïve Bayes is used when features are continuous and follow a Gaussian distribution. The Naïve Bayes classifier is surprisingly effective despite the constraint of the interdependence among attributes, typically email words or phrases [24].

2. Method

There are different methods to perform sentiment analysis on different textual datasets; the execution performed here is machine learning models with a class balancing technique, which was discussed earlier. As shown in trend analysis figure (2), positive, negative and neutral sentiment classes are highly imbalanced as the positive class’s ratio is high, impacting the machine learning model, and the result would be biased towards the positive class as the data is then balanced it potentially improved model performance. After data preprocessing, which is crucial, the implementation was done on SGD-based SVM (Stochastic Gradient Descent-based Support Vector Machine) to train SVM, particularly for large datasets, by updating model parameters. This model performed better than naïve bayes in terms of prediction, as shown in Table (1).

2.1. Dataset Description

The dataset used in this study consists of Amazon food reviews; it has score and text reviews and is then categorized into three sentiment classes: Positive, Negative, and Neutral, with respective counts of 24,650, 15,384, and 8,181. The SMOTE algorithm was employed to generate synthetic samples to mitigate the effects of class imbalance, ensuring equal representation across all classes. This algorithm addresses class imbalance by generating synthetic samples for the minority classes, effectively increasing their representation in the dataset—text preprocessing steps encompassed removing special characters and stop words, followed by TF-IDF vectorization for feature extraction. TF-IDF, or Term Frequency-Inverse Document Frequency, is a widely used technique in natural language processing for converting text documents into numerical vectors that can be used as input for machine learning models.

To ensure a rigorous and unbiased evaluation, the dataset was partitioned into training and testing sets using stratified sampling, preserving the class distribution in both sets. Two classification models were implemented: Naïve Bayes and SGD-based SVM. The Naïve Bayes classifier was applied with optimized hyperparameters, while the SGD-based SVM utilized hinge loss with balanced class weights and optimized learning rate. Using hinge loss in the SGD-based SVM encourages the model to find a decision boundary that maximizes the margin between classes, potentially improving generalization performance.

2.2. Dataset Preprocessing

The methodology here is text processing, like removing stop words and irrelevant characters and showcasing only the important words for the model to train better. Subsequently, TF-IDF vectorization is applied to extract salient features, followed by class balancing to rectify disparities in class representation.

For any model training and testing, a proper pre-processed dataset is necessary for better evaluation. Here, the synthetic minority oversampling technique is applied to balance the dataset because, as in the original dataset, there are more positive reviews, so the result gets biased towards positive sentiment. A star rating greater than three is labelled as positive sentiment less than negative or neutral sentiment. Here, the figure given explains more about the distribution of star ratings.

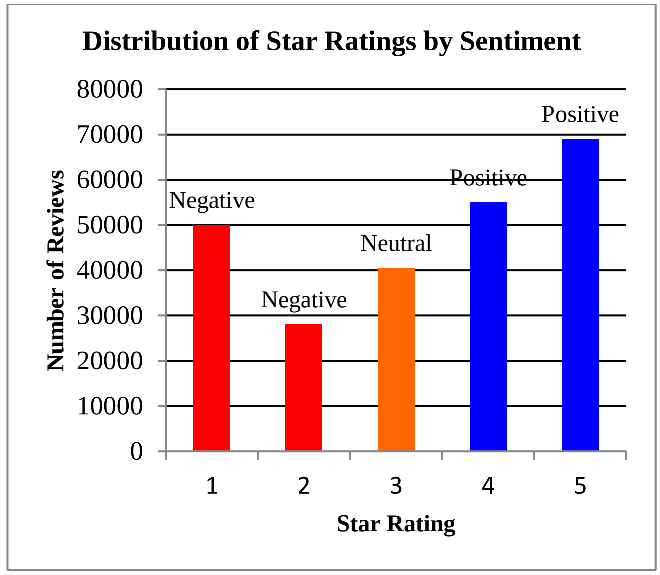


Fig. 1 Distribution of star ratings by sentiment labels

2.3. Dataset Balancing

The dataset is completely balanced with the SMOTE algorithm, with 9339 data for each star evaluation. As discussed earlier, dataset balancing is performed here so that the model is not biased towards positive reviews and fair evaluation is performed.

2.4. Sentiment Review

The dataset used here for the observation has the most positive reviews class compared to the other two. The reviews are classified based on star ratings in the dataset. Figure (2) suggests that many users have expressed favourable opinions, contributing to a positive overall sentiment. Conversely, the lower count of Negative sentiments indicates that unfavourable opinions are less prevalent, further reinforcing the dominance of positive sentiment.

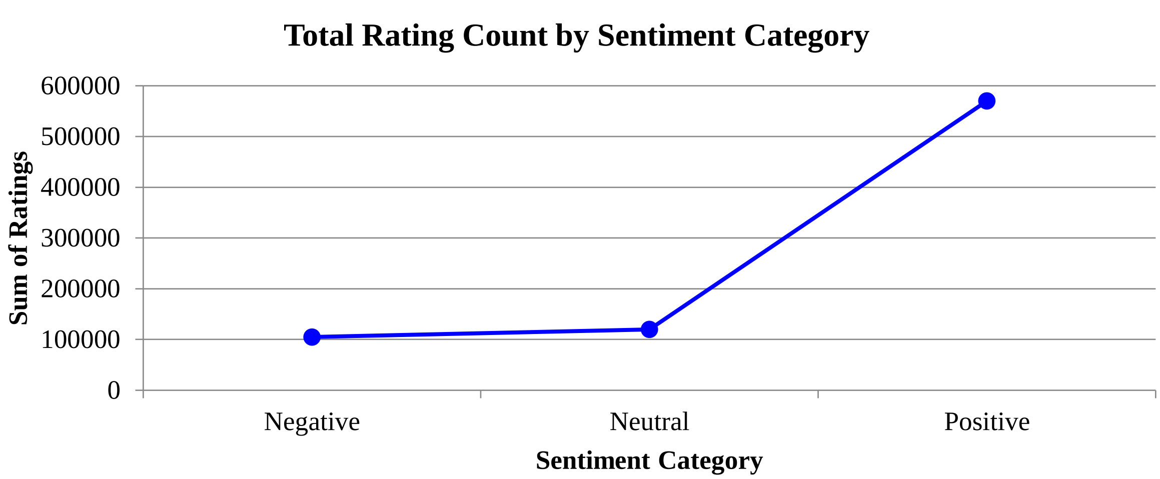


Fig. 2 Total rating count by sentiment category

The visual representation of sentiment distribution showcases the trend analysis of sentiment category vs sum of ratings, plotting sentiment classes against the cumulative count of ratings associated with each class. This graphical depiction provides a clear and concise overview of the prevailing sentiments within a given dataset. It can be easily observed that positive sentiments have a higher ratio than the other two sentiments, which could affect the model results. A series of methodological steps were meticulously executed to effectively discern subjective sentiments from textual data, encompassing dataset preparation, model training, and performance evaluation. These steps ensured the robustness and reliability of the sentiment analysis conducted on Amazon food reviews [35]

2.5. Model Training

After the dataset was processed and the labels were given to each review according to the ratings in the dataset, it was observed that the positive class sentiment's ratio was more than the other two sentiments, which could make the models biased, so SMOTE was applied on the dataset to make it balanced for each sentiment classes and fair evaluation. Then, the Naïve bayes and SGD-based SVM were implemented on the balanced dataset and obtained 79.9% and 84% accuracy, respectively. Implementing the sentiment analysis pipeline leveraged several key tools and libraries within the Python ecosystem. Specifically, scikit-learn was used for machine learning tasks [34]. The TF-IDF vectorization was also performed using scikit-learn, employing the TfidfVectorizer class. The Term Frequency-Inverse Document Frequency, also implemented in scikit-learn, is a feature extraction technique widely used in natural language processing that converts text documents into numerical vectors [11]. The selection of hyperparameters for the machine learning models was guided by established best practices and empirical experimentation. To optimize the performance of the machine learning models, a grid-search method was implemented to compute and compare the performance metrics for the same classifiers by using different parameters with different values to get the best combination of parameters [11].

3. Results and Discussion

3.1. Result

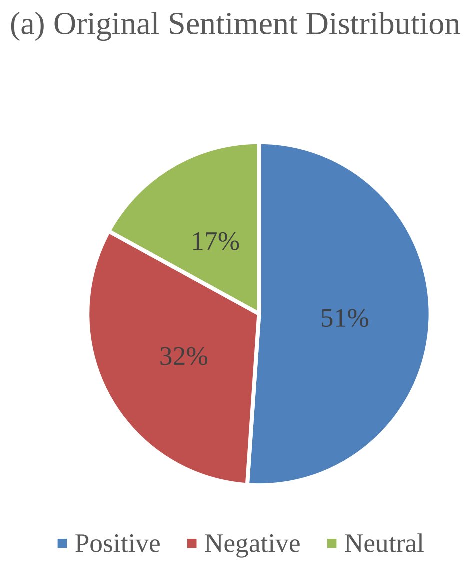
The models were evaluated using accuracy, Precision, Recall, and F1-score metrics, focusing on determining the best-performing classifiers [33].

3.1.1. Naïve Bayes

The naïve bayes model showcases that 2,201 negative instances were incorrectly classified as neutral, and 875 were misclassified as positive. Neutral sentiments were also classified with reasonable accuracy (5,762 instances), but a considerable number were misidentified as negative (1,176 instances) or positive (1,243 instances). The model correctly classified 5,762 neutral instances, but 1,176 were misclassified as negative, and 1,243 were misclassified as positive.

The Naïve Bayes classifier exhibited strong performance regarding positive sentiments, correctly classifying 20,459 instances. Nevertheless, 1,192 positive instances were misclassified as negative and 2,999 as neutral. The model predicts accurately neutral class but struggles with positive and negative classes, and the overall observations show that the model is balanced but noisy.

The figure shows all three sentiment labels' original and predicted proportional distribution.



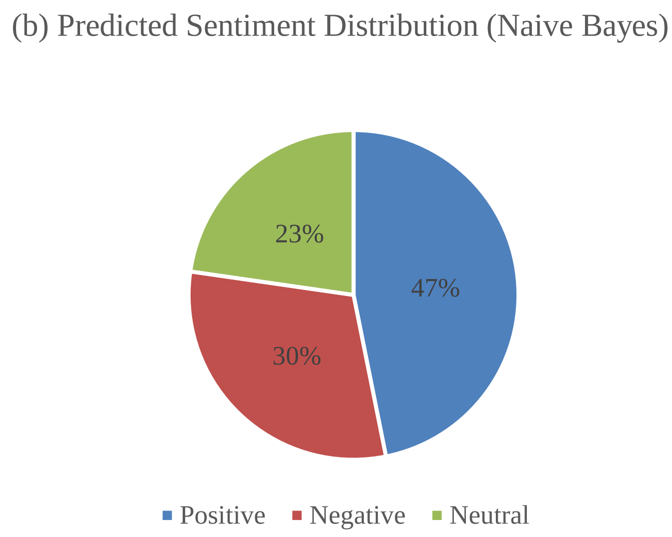
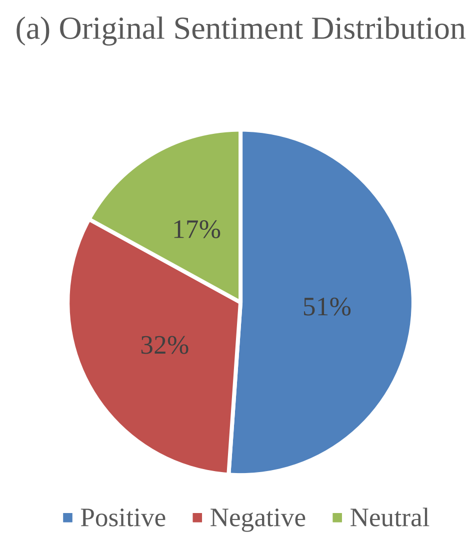


Fig. 3 Proportional distribution of sentiment classes and predicted sentiment classes (a) Original Sentiment Distribution, (b) Predicted Sentiment Distribution by Naïve Bayes

3.1.2. SGD-based SVM

In contrast to naïve bayes, SGD-based SVM shows high accuracy in positive and negative classes but struggles with neutral classes. It has better Precision and is more accurate for positive and negative classes. The model accurately classified negative sentiments, with 13,192 instances correctly identified as negative out of 15,384 negative reviews. However, fewer negative instances were misclassified as neutral or positive. Specifically, 286 negative instances were incorrectly classified as neutral, and 1,906 were misclassified as positive. The SGD SVM struggled with neutral sentiments, correctly classifying only 3,815 instances. Many neutral instances were misclassified as negative or positive, with 1,551 instances incorrectly classified as negative and 2,815 as positive. The SGD SVM showcased exceptional performance for positive sentiments, correctly classifying 23,672 instances out of total positive reviews 24650. The SVM model with linear kernel showcased the same accuracy, 80%, like naïve bayes, so the implementation was performed on SGD-based SVM with an accuracy of 84% for better performance and fast compared to the simple SVM model.

Here, the figure shows a proportional distribution of labels showcasing original and predicted labels. It can be observed that this model performs slightly better than naïve bayes for predicting positive and negative class sentiments.



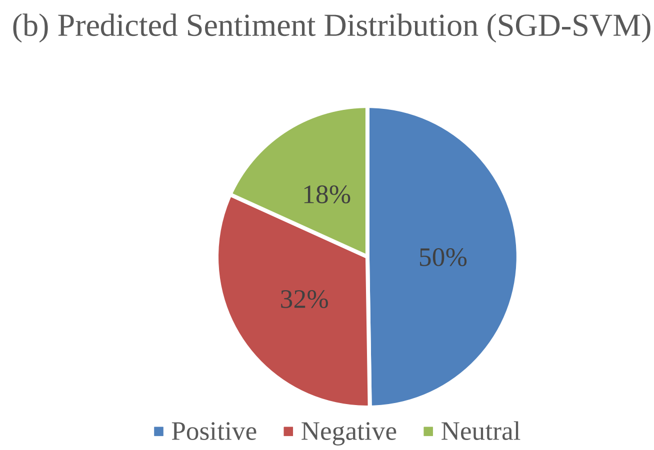


Fig. 4 Proportional distribution of sentiment classes and predicted sentiment classes (a) Original Sentiment Distribution, (b) Predicted Sentiment Distribution by SGD-based SVM

3.2. Discussion

The classification models naïve bayes and SGD-based SVM were implemented on the textual dataset with accuracy of 79.9% and 84.4%, respectively. The evaluation was based on accuracy, Precision, recall and F1 score. Furthermore, macro and weighted averaging techniques aggregate these per-class metrics into a single representative score, providing a more holistic evaluation of the model's overall performance. The confusion matrix is a foundation for computing many crucial evaluation metrics, allowing a detailed breakdown of true positives, false positives, true negatives, and false negatives for each class [27].

When evaluating models designed for sentiment analysis, particularly those dealing with negative, neutral, and positive sentiments, it is crucial to consider metrics tailored to each sentiment class. Negative Precision quantifies the proportion of instances predicted as negative, offering insight into the model's reliability in identifying negative sentiments. Negative Recall, however, measures the proportion of actual negative instances correctly identified as negative by the model, reflecting its ability to capture the entirety of negative sentiments [28]. The Negative F1-score is the harmonic mean of negative Precision and negative Recall, providing a balanced measure of the model's performance in identifying negative sentiments, especially useful when there is an uneven class distribution [29]. Similarly, Neutral Precision indicates the model's accuracy in predicting neutral instances for neutral sentiments, while Neutral Recall measures its completeness in capturing all actual neutral instances. The Neutral F1-score combines these two aspects, providing a metric that balances Precision and recall for the neutral class. For positive sentiments, Positive Precision reflects the accuracy of optimistic sentiment predictions. At the same time, Positive Recall indicates the model's ability to identify all actual positive instances, with the Positive F1 score providing a balanced assessment of positive sentiment classification [30]. The higher Recall of SGD-based SVM indicates its ability to capture more relevant positive instances.

The observation of these models is given in Table (1) for a better understanding and comparison of both models. It can be seen that SGD-based SVM struggles with ‘neutral’ classes in comparison to naïve bayes, whereas it performs better in positive and negative sentiment prediction.

Table 1. Performance metrics comparison of classification models

Metric	Naïve Bayes	SGD-based SVM
Accuracy	79.91%	84.37%
Negative Precision	83.86%	85.19%
Negative Recall	80.00%	85.75%
Negative F1-score	81.88%	85.47%
Neutral Precision	52.56%	87.96%
Neutral Recall	70.43%	46.63%
Neutral F1-score	60.19%	60.95%
Positive Precision	90.62%	83.37%
Positive Recall	82.99%	96.03%
Positive F1 score	86.64%	89.25%
Macro avg F1 score	76.24%	78.56%
Weighted avg F1 score	80.63%	83.25%

The Table shows that SGD-based SVM outperforms naïve bayes in accuracy and F1 scores.

4. Conclusion

This study used the food review dataset to compare machine learning models on sentiment analysis. First, the data was preprocessed as a clean and processed dataset, which is necessary for model evaluation, so only important words that affect the sentiment are considered. Then, the labels 'positive', 'negative' and 'neutral' were applied based on the star ratings given in the dataset. Later, there was the observation that the ratio of positive sentiment class is higher than that of other sentiment classes, and this could affect the result of the prediction as models would be biased towards the positive class. So, to solve this synthetic minority oversampling technique, the sentiment classes were balanced equally for better and more accurate results. Later, the naïve bayes model was implemented on the balanced dataset, and the accuracy obtained was 79.9%. Then, SGD-based SVM was implemented because simple SVM gave the same result as other models, so it was tuned, and the accuracy obtained was 84%.

It was observed that models struggle to understand sarcastic and negation words, which can be improved further by implementing ensemble or pre-trained transformer models.

References

[1] Honglie Zhang, "Model Comparison in Sentiment Analysis: A Case Study of Amazon Product Reviews," *Highlights in Science Engineering and Technology*, vol. 16, pp. 23-31, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[2] Mohammad Mohaiminul Islam, and Naznin Sultana, "Comparative Study on Machine Learning Algorithms for Sentiment Classification," *International Journal of Computer Applications*, vol. 182, no. 21, pp. 1-7, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[3] Stuti M. Meshram, and Neeraj Sahu, "Sentiment Analysis of E-commerce Product Review through Machine Learning," *International Journal for Multidisciplinary Research*, vol. 5, no. 3, pp. 1-5, 2023. [Publisher Link] [CrossRef]

[4] Karthick Prasad Gunasekaran, "Exploring Sentiment Analysis Techniques in Natural Language Processing: A Comprehensive Review," *arXiv*, pp. 1-6, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] Antony Samuels, and John Mcgonical, "Sentiment Analysis on Customer Responses," *arXiv*, pp. 1-3, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[6] Maite Taboada, "Sentiment Analysis: An Overview from Linguistics," *Annual Review of Linguistics*, vol. 2, pp. 325-347, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[7] Jamin Rahman Jim et al., "Recent Advancements and Challenges of NLP-based Sentiment Analysis: A State-of-the-art Review," *Natural Language Processing Journal*, vol. 6, pp. 1-30, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] Nishit Shrestha, and Fatma Nasoz, "Deep Learning Sentiment Analysis of Amazon.com Reviews and Ratings," *arXiv*, pp. 1-15, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[9] Kuat Yessenov, and Sasa Misailovic, "Sentiment Analysis of Movie Review Comments," *Methodology*, vol. 17, no. 17, pp. 1-7, 2009. [Google Scholar] [Publisher Link]

[10] Mika V. Mäntylä, Daniel Graziotin, and Miikka Kuutila, "The Evolution of Sentiment Analysis - A Review of Research Topics, Venues, and Top Cited Papers," *Computer Science Review*, vol. 27, pp. 16-32, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[11] Shadi Diab, "Optimizing Stochastic Gradient Descent in Text Classification Based on Fine-Tuning Hyper-Parameters Approach. A Case Study on Automatic Classification of Global Terrorist Attacks," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 16, no. 12, pp. 155-160, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[12] Earl E. Reber, Richard L. Michell, and Clarence J. Carter, "Oxygen Absorption in the Earth's Atmosphere," *Aerospace Corp El Segundo Calif Los Operatios*, pp. 1-102, 1968. [Publisher Link]

[13] Kaustubh Yadav, "A Comprehensive Study on Optimization Strategies for Gradient Descent in Deep Learning," *arXiv*, pp. 1-12, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[14] Subin Sahayam, John Zakkam, and Umarani Jayaraman, "Can we Learn Better with Hard Samples?," *arXiv*, pp. 1-10, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] Diederik P. Kingma, and Jimmy Ba, "Adam: A Method for Stochastic Optimization," *arXiv*, pp. 1-15, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[16] Jayanth Koushik, and Hiroaki Hayashi, "Improving Stochastic Gradient Descent with Feedback," *Open Review.net*, pp. 1-9, 2016. [Google Scholar] [Publisher Link]

[17] Kai Zhang et al., "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142-3155, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[18] Thorsten Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," *Machine Learning: ECML-98*, pp. 137-142, 1998. [CrossRef] [Google Scholar] [Publisher Link]

[19] Liliya A. Demidova, "Two-Stage Hybrid Data Classifiers Based on SVM and kNN Algorithms," *Symmetry*, vol. 13, no. 4, pp. 1-32, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[20] David L. de Souza et al., "Fault Detection and Diagnosis Using Support Vector Machines - A SVC and SVR Comparison," *Journal of Safety Engineering*, vol. 3, no. 1, pp. 18-29, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[21] Hwanjo Yu, and Sungchul Kim, “SVM Tutorial - Classification, Regression and Ranking,” *Handbook of Natural Computing*, pp. 479-506, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[22] Hilman Wisnu, Muhammad Afif, and Yova Ruldevyani, “Sentiment Analysis on Customer Satisfaction of Digital Payment in Indonesia: A Comparative Study using KNN and Naïve Bayes,” *Journal of Physics Conference Series*, vol. 1444, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[23] S.G. Fitri et al., “Naïve Bayes Classifier Models for Cerebral Infarction Classification,” *Journal of Physics Conference Series*, vol. 1490, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[24] Kiran Bolaj, “Text Categorization System for English Text Documents using Naïve Bayes Classifier,” *International Journal of Computer Applications*, vol. 177, no. 48, pp. 1-4, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[25] Sebastian Raschka, “Naive Bayes and Text Classification I - Introduction and Theory,” *arXiv*, pp. 1-20, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[26] Qijia Wei, “Understanding of the Naive Bayes Classifier in Spam Filtering,” *AIP Conference Proceedings*, vol. 1967, no. 1, pp. 1-8, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[27] Rimsha Rafique et al., “Deep Fake Detection and Classification using Error-level Analysis and Deep Learning,” *Scientific Reports*, vol. 13, pp. 1-13, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[28] Marina Sokolova, and Guy Lapalme, “A Systematic Analysis of Performance Measures for Classification Tasks,” *Information Processing & Management*, vol. 45, no. 4, pp. 427-437, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[29] Kai Ming Ting, “Confusion Matrix,” *Encyclopedia of Machine Learning and Data Mining*, 2017. [CrossRef] [Publisher Link]

[30] David M.W. Powers, “What the F-measure doesn’t Measure: Features, Flaws, Fallacies and Fixes,” *arXiv*, pp. 1-19, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[31] Hery Iswanto et al., “Comparison of Algorithms on Machine Learning for Spam Email Classification,” *International Journal of Information System & Technology*, vol. 5, no. 4, pp. 1-10, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[32] Robert E. Karlsen, David J. Gorsich, and Grant R. Gerhart, “Target Classification via Support Vector Machines,” *Optical Engineering*, vol. 39, no. 3, 2000. [CrossRef] [Google Scholar] [Publisher Link]

[33] Muhammad Haroon et al., “Sentiment Analysis of Customer Reviews on E-commerce Platforms: A Machine Learning Approach,” *Bulletin of Business and Economics (BBE)*, vol. 13, no. 3, pp. 230-238, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[34] Emma Haddi, Xiaohui Liu, and Yong Shi, “The Role of Text Pre-processing in Sentiment Analysis,” *Procedia Computer Science*, vol. 17, pp. 26-32, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[35] Ayal Taitler et al., “The 2023 International Planning Competition,” *AI Magazine*, vol. 45, no. 2, pp. 280-296, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[36] Alexandra Balahur et al., “Sentiment Analysis in the News,” *arXiv*, pp. 1-5, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[37] Logan Ashbaugh, and Yan Zhang, “A Comparative Study of Sentiment Analysis on Customer Reviews Using Machine Learning and Deep Learning,” *Computers*, vol. 13, no. 12, pp. 1-16, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[38] Gilad Mishne, and Natalie Glance, “Predicting Movie Sales from Blogger Sentiment,” *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pp. 1-4, 2006. [Google Scholar] [Publisher Link]

[39] Md Mahbubur Rahman, and Shaila Shova, “Emotion Detection from Social Media Posts” *arXiv*, pp. 1-9, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[40] Vivek Narayanan, Ishan Arora, and Arjun Bhatia, “Fast and Accurate Sentiment Classification Using an Enhanced Naive Bayes Model,” *Intelligent Data Engineering and Automated Learning-IDEAL 2013*, pp. 194-201, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[41] Aman Sawarn, Ankit, and Monika Gupta, “Comparative Analysis of Bagging and Boosting Algorithms for Sentiment Analysis,” *Procedia Computer Science*, vol. 173, pp. 210-215, 2020. [CrossRef] [Google Scholar] [Publisher Link]