Article

Not peer-reviewed version

A Comparative Study of Sentiment Analysis on Customer Reviews Using Machine Learning and Deep Learning

Logan Ashbaugh and Yan Zhang *

Posted Date: 11 November 2024

doi: 10.20944/preprints202411.0741.v1

Keywords: sentiment analysis; customer reviews; machine learning; deep learning; natural language processing

Article

A Comparative Study of Sentiment Analysis on Customer Reviews Using Machine Learning and Deep Learning

Logan Ashbaugh [1,†,‡] and Yan Zhang [2,†,‡*]

1 School of Computer Science and Engineering, California State University San Bernardino
2 School of Computer Science and Engineering, California State University San Bernardino
* Correspondence: Yan.Zhang@csusb.edu; Tel.: +1-909-537-5333
† Current address: 5500 University Parkway, San Bernardino, CA, 92407, USA.
‡ Two authors contributed equally to this work.

Abstract: Sentiment analysis is a key technique in natural language processing that enables computers to understand human emotions expressed in text. It is widely used in applications such as customer feedback analysis, social media monitoring, and product reviews. However, sentiment analysis of customer reviews presents unique challenges, including the need for large datasets and the difficulty of accurately capturing subtle emotional nuances in text. In this paper, we present a comparative study of sentiment analysis on customer reviews using both deep learning and traditional machine learning techniques. The deep learning models include Convolutional Neural Networks (CNN) and Recursive Neural Networks (RNN), while the machine learning methods consist of Logistic Regression, Random Forest, and Naive Bayes. Our dataset is composed of Amazon product reviews, where we utilize the star rating as a proxy for the sentiment expressed in each review. Through comprehensive experiments, we assess the performance of each model in terms of accuracy and effectiveness in detecting sentiment. This study provides valuable insights into the strengths and limitations of both deep learning and traditional machine learning approaches for sentiment analysis.

Keywords: sentiment analysis; customer reviews; machine learning; deep learning; natural language processing

---

1. Introduction

Sentiment analysis is a growing research area in natural language processing that enables computers to interpret and classify human emotions expressed in text. As the textual data generated and accumulated online continues to grow, sentiment analysis has become increasingly important for businesses, governments and researchers to gain insights into customer satisfaction, public opinion, and emotional trends [1,2]. This technology has widespread applications, from monitoring public opinions on social medias to analyzing customer feedback in online shopping. Within this field, customer review sentiment analysis is particularly valuable, as it allows businesses or companies to assess customer feedback and improve products and services based on the sentiment of reviews.

However, analyzing sentiment in customer reviews presents some challenges. The subjective nature of human emotions makes it difficult to accurately capture sentiment from plain textual data, as the same words can convey different meanings or emotions in varying contexts. Moreover, the large volume of data required to train effective models and the need to balance the dataset to avoid bias further complicate the sentiment analysis process. These challenges demand robust methodologies and algorithms to achieve reliable and effective sentiment analysis.

This paper presents a comparative study of various machine learning and deep learning models for sentiment analysis with the focus on Amazon product reviews. We use deep learning models such as Convolutional Neural Networks (CNN) and Recursive Neural Networks (RNN), as well as traditional machine learning algorithms such as Logistic Regression, Random Forest, and Naive Bayes.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 November 2024 doi:10.20944/preprints202411.0741.v1

The goal of this research is to perform a comparative analysis of these diverse techniques or algorithms, evaluating their performance in detecting sentiment, providing insights into their effectiveness in handling customer reviews.

The significance of this research lies in its potential to enhance sentiment analysis capabilities in online marketplaces, where understanding customer sentiment is crucial for product development, marketing strategies, and customer service. This study contributes valuable knowledge to identify the effective models for sentiment analysis, helping businesses make data-driven decisions based on customer feedback.

2. Related Work

Sentiment analysis has attracted significant attention in recent years, particularly with the rise of online shopping platforms where customers can leave reviews or comments about products. These reviews offer insights into customer satisfaction, and they are important source of feedback for businesses. Various approaches have been developed to improve the accuracy of sentiment analysis, using both traditional machine learning models and deep learning techniques. We review several works that have explored sentiment analysis on customer reviews, highlighting their methodologies, datasets, and results.

Haque et al. developed a sentiment analysis prototype to summarize Amazon product reviews, so that users don't have to go through hundreds of them manually [3]. They polarized each review as positive or negative and found that the support vector machine (SVM) achieved over 90% accuracy in determining overall sentiment [3]. Rashid and Huang gathered Amazon reviews to analyze the relationship between high-cost products and the number of helpful reviews [4]. They encountered an issue with biased ratings toward 4- and 5-star reviews, a challenge we also faced in our study. Despite this imbalance, they continued to explore other categories in their research [4].

AlQahtani studied sentiment analysis using machine learning, training models such as Logistic Regression, Random Forest, Naive Bayes, Bidirectional Long-Short Term Memory (Bi-LSTM), and Bidirectional Encoder Representations from Transformers (BERT) on Amazon reviews [5]. He found that BERT achieved the highest accuracy at 98%, while Bi-LSTM and Random Forest performed well, with accuracies of 94% each [5]. Kumar et al. also analyzed the sentiment of Amazon reviews using Naive Bayes, Logistic Regression, and SentiWordNet. They found out that Naive Bayes performed best among the algorithms tested [6].

Ali et al. employed a variety of machine learning algorithms, including Multinomial Naive Bayes, Random Forest, Decision Tree, and Logistic Regression, as well as deep learning algorithms, including CNNs and Bidirectional LSTM, and transformer models, including XLNet and BERT to analyze the sentiment of Amazon product reviews. The experiments showed that the BERT algorithm outperformed others, achieving an accuracy rate of 89% [7]. Tan et al. analyzed the sentiment of Amazon produce reviews using algorithms such as Naive Bayes, Support Vector Machine (SVM), K-Nearest-Neighbors (KNN), and Long Short-Term Memory (LSTM)[8]. They ultimately ran into the problem of data imbalance (too many 5-star reviews) and the issue of not having enough data to properly train their models. Despite this, they found that LSTM performed the best with the highest accuracy [8]. In the future, they would like to gather data from other sources to attempt to balance the data.

Park et al. conducted sentiment analysis on 300,000 automobile reviews from 10 different internet communities, comparing the performances of Artificial Neural Network (ANN), Support Vector Machine (SVM), and Graph-based Semi-Supervised Learning (GSSL). Their results indicated that GSSL performed best with 98.1% accuracy, followed closely by SVM at 97.4% and ANN at 72.4% [9]. In a study of 1,200 tweets related to Travelloka, Diekson et al. compared Logistic Regression, SVM, and Naive Bayes, finding that SVM outperformed the others with an accuracy of 84.58%, followed by Naive Bayes with 82.91% and Logistic Regression with 82.50% [10] .

 doi:10.20944/preprints202411.0741.v1

Grljević et al. analyzed sentiment of review data collected from Amazon, IMDB, and Yelp, comparing the performance of Naive Bayes, SVM, and K-Nearest Neighbors (KNN). Their results showed that SVM was the most effective, achieving an F-measure of 79.70%, followed by Naive Bayes with 76.79% and KNN with 76.40% [11] Chinnalagu et al. compared three models — linear SVM (LSVM), fastText, and Bi-directional Long Short-Term Memory model (SA-BLSTM) — on sentiment analysis tasks. They found that fastText performed the best with an accuracy of 90.71%, followed by LSVM at 90.11%, and lastly SA-BLSTM at exactly 77% [12].

Obiedat et al. conducted an extensive comparison of several models for customer review sentiment analysis, including SVM particle swarm optimization + synthetic minority over-sampling technique (SVM-PSO+BSMOTE), SVM, extreme gradient boosting (XGBoost), Decision Tree (DT), Random Forest (RF), Naive Bayes (NB), K-Nearest Neighbers (KNN), and Logistic Regression (LR). Their results showed that SVM-PSO+BSMOTE achieved the highest accuracy at 80%, followed by Logistic Regression at 77%, SVM at 76%, RF at 75%, DT at 73%, XGBoost at 66%, KNN at 59%, and finally NB at 50% [13].

The existing research demonstrates that sentiment analysis of customer reviews has been widely explored using both traditional machine learning models and advanced deep learning techniques. SVM frequently emerges as a top performer across multiple studies, while newer models like BERT and Bi-LSTM show strong results in deep learning approaches. Despite variations in datasets, the comparative performance of different algorithms continues to provide insights for improving sentiment analysis. Our research builds on this foundation by further exploring how these models perform on Amazon customer reviews.

3. Dataset Collection and Preprocessing

3.1. Data Collection

We collected customer reviews from Amazon for a variety of products across different categories. To ensure a diverse dataset, reviews were gathered from multiple product types, with the earliest review dating back to May 2015 and the latest from February 2024. Each review included both the text of the review and the corresponding product star rating. In total, we collected approximately 30,000 reviews, with an average review length of 383 characters and contained 6,107 unique words. However, the distribution of ratings was skewed toward higher scores, with only around 3,500 1-star reviews compared to over 10,000 5-star reviews. This imbalance may be attributed to the fact that products with consistently poor ratings are likely removed from the platform, leaving more products with favorable ratings. As a result, most products had significantly more 4- or 5-star reviews than 1- or 2-star reviews. Table 1 outlines the number of customer reviews collected for each star rating (1 to 5 stars).

Table 1. The distribution of customer reviews by star rating.

Star rating	Amount of Reviews
5-star	10584
4-star	9026
3-star	6096
2-star	2877
1-star	3471

3.2. Data Preprocessing

The data preprocessing step involved several key operations to prepare the textual reviews for sentiment analysis.

- First, we removed punctuation to ensure the model wouldn't misinterpret symbols.

- Next, we normalized the case of all words, converting them to lowercase so that the model would not get different results from the same word with a different capital letter, for example, the words like "Good" and "good" would not be treated as distinct.
- Then, we filtered out stop words — common words such as "the", "are", "it" — that do not contribute meaningfully to the sentiment of a review.
- Finally, stemming was applied, reducing words to their base forms (e.g., "writing" becomes "write") to prevent the model from treating different word variations as distinct entities.

For instance, the raw review input "Comes with a lot, the dogs seem to like them. Smell pretty bad." would be transformed into ['come', 'lot', 'dog', 'seem', 'like', 'smell', 'pretti', 'bad'] after preprocessing.

This process of removing non-essential words and standardizing word forms helps reduce ambiguity and enhances the model's performance. After preprocessing, the review dataset contained approximately 6,079 unique words, with an average review length of around 130 characters.

4. Methodologies

In this comparative study, we employed two deep learning models and three traditional machine learning algorithms to analyze customer reviews for sentiment classification. The deep learning models used were the Recursive Neural Network (RNN) and the Convolutional Neural Network (CNN), while the machine learning algorithms included Logistic Regression, Naive Bayes, and Random Forest. Each of these models brings unique strengths to sentiment analysis classification, and the following subsections provide an overview of how each algorithm operates and its specific advantages for this task.

4.1. Logistic Regression

Logistic Regression is a widely used machine learning algorithm for classification tasks, making it suitable for customer review sentiment analysis where reviews can be classified into multiple sentiment classes. Logistic regression model estimates the probability that a given input, such as a customer review, belongs to a particular sentiment class by using a logistic/sigmoid function [14]. For multi-class classification, Logistic Regression is often extended through techniques like one-vs-rest (OvR) or softmax regression, allowing it to handle more than two sentiment classes [15]. While Logistic Regression is relatively simple compared to deep learning models, it is highly interpretable and computationally efficient, making it a reliable choice for sentiment analysis when working with large datasets [15]. However, its ability to capture complex patterns or contextual information may be more limited compared to deep learning models like RNNs or CNNs.

4.2. Naive Bayes Classification

Naive Bayes is a probabilistic machine learning algorithm based on Bayes' Theorem, often used for text classification tasks such as sentiment analysis. It assumes that the features (in this case, words in a review) are conditionally independent of each other given the class label, which is a "naive" assumption in practice but works surprisingly well in many scenarios [16]. For sentiment analysis, Naive Bayes calculates the probability of a review belonging to a specific sentiment class (e.g., positive, negative, or neutral) by considering the likelihood of the individual words in the review [15]. Despite its simplicity, Naive Bayes is fast, easy to implement, and performs well with smaller datasets, especially when the independence assumption holds [17]. One of its strengths is its ability to handle noisy data effectively, but its performance can degrade when the assumption of feature independence is violated.

4.3. Random Forest Classification

Random Forest classification is an ensemble learning algorithm that combines the predictions of multiple decision trees to improve classification performance and avoid overfitting [18]. In the context of sentiment analysis, Random Forest works by constructing multiple decision trees from various

 doi:10.20944/preprints202411.0741.v1

subsets of the training data, and then taking a majority vote from the individual trees to classify a review. This approach helps reduce overfitting, which is a common issue in single decision tree models, and increases the robustness of the model [19]. Random Forest is particularly effective in handling large datasets with a variety of features, such as customer reviews, by leveraging its ability to capture complex patterns through feature randomness. Its strengths include high accuracy, scalability, and resistance to noise in the data [19]. Random Forest can handle lots of variation in the input data, which is very helpful with our use case since there is a lot of variation in the emotion of sentences. However, compared to deep learning models like CNNs or RNNs, Random Forest may struggle with understanding nuanced contextual relationships between words in a sentence.

4.4. Recursive Neural Network (RNN)

A Recursive Neural Network (RNN) operates similarly to other neural networks, with an input layer, hidden layers, and an output layer. However, what distinguishes an RNN is the recurrent connections within the hidden layers which make it able to loop back, or "recur," over certain layers within the network [20]. This recurrent nature allows the model to retain and use information from previous inputs, which is particularly useful for sentiment analysis [21]. By remembering past inputs, the RNN can provide contextual understanding to the current input, enabling it to interpret how earlier words in a sentence influence the sentiment of later ones. This characteristic is especially useful in the natural language processing (NLP) area due to this ability to maintain context, making them effective for tasks like sentiment analysis, where the meaning of a word is often shaped by the surrounding text [20,22].

4.5. Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a type of deep learning model traditionally used in image processing, but it has proven to be useful and effective for natural language processing tasks such as sentiment analysis [23]. In a CNN, the network uses convolutional layers to scan and extract features from input data, which in this case are sequences of words in a customer review [24]. By applying filters across different parts of the text, CNNs can capture local dependencies and patterns, such as common word combinations that signal sentiment. Unlike models that process input sequentially, CNNs focus on identifying important features within a fixed window of words, making them effective for identifying key phrases or expressions that indicate sentiment [24]. CNNs are also efficient and can handle large-scale datasets, making them a robust option for analyzing customer reviews [25].

5. Experimental Results and Analysis

5.1. Training Classification Models

After collecting and preprocessing the dataset of approximately 30,000 customer reviews, the next step was to prepare the data for training. For traditional machine learning models such as Logistic Regression, Naive Bayes, and Random Forest, we converted the text reviews into a matrix of token counts. This process transforms each text review into numerical features, with the number of features corresponding to the vocabulary size extracted from the dataset and the values of features as the frequency count of the word in the review. Scikit-learn machine learning library is employed to train traditional machine learning models.

For the deep learning models — Convolutional Neural Networks (CNN) and Recursive Neural Networks (RNN) — we processed the text reviews by building a token dictionary. Each unique token or word in the review dataset was assigned an integer value. We then converted the text reviews into sequences of integers, where each integer represents the index of a token. This sequence-based representation allows the deep learning models to process and learn from the textual data. The deep learning library Keras is utilized to build CNN and RNN models.

Once the data was appropriately transformed, we split the dataset into training and testing sets, with 80% of the data (around 24,000 reviews) used for training and 20% (approximately 6,000 reviews) reserved for testing. This split ensured that the models were trained on a substantial portion of the data while leaving enough for performance evaluation on unseen data. The models were trained using the preprocessed textual reviews along with the star ratings as output labels, to predict the star ratings of customer reviews.

5.1.1. Training Logistic Regression Model

Given that the task involves predicting 5 distinct star ratings (class labels) based on customer reviews, we employed multinomial logistic regression. This model predicts a multinomial probability distribution for each input review, fitting a matrix of coefficients where each row vector corresponds to one of the five star ratings. We used the "lbfgs" solver, which is an optimization algorithm that approximates the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm in Scikit-learn. The solver supports both L2 regularization and multiclass classification, making it well-suited for our task, especially since it is known to converge faster for high-dimensional data. L2 regularization was applied to handle sparse input, improve numerical stability, and introduce a penalty term to reduce overfitting. The model was trained over 100 iterations, which was sufficient for the solver to converge and achieve stable results.

5.1.2. Training Naive Bayes Classification Model

We used the Multinomial Naive Bayes implementation, which is well-suited for multinomially distributed data, such as word vector counts derived from customer reviews. This algorithm assumes that each feature (or word) contributes independently to the likelihood of a class, making it effective for text classification tasks. To improve the model's robustness, we applied Laplace smoothing, with a smoothing prior $\alpha = 1$. This technique ensures that features not present in the training data are accounted for and prevents the calculation of zero probabilities in future computations, which could otherwise negatively impact the model's predictions.

5.1.3. Training Random Forest Classification Model

In our setup, the model builds 100 decision trees, each tree trained using a random subset of features from the full set of customer reviews. Each decision tree is trained on all samples from the training set, allowing the model to learn diverse patterns across the data. Gini impurity was used as the criterion to measure the quality of splits at each node, ensuring the best possible split based on feature values. Nodes were expanded until either all the leaves became pure, meaning all the samples within them belong to the same class, or until the leaves contained less than two samples. There was no limit on the number of leaf nodes in the trees, allowing for more complex decision boundaries to be captured. This ensemble method of averaging the output from 100 different trees helps improve the model's generalization capability and performance on the test set.

5.1.4. Training RNN Model

For the Recursive Neural Network (RNN) model, we designed the architecture to handle the sequential nature of text data. The model begins with an embedding layer, which transforms each word in the input sequence into a dense vector representation. This is followed by a Long Short-Term Memory (LSTM) layer, which helps capture long-term dependencies in the text, making it particularly suited for sentiment analysis where context and word order are important. After the LSTM layer, we added a pooling layer to reduce the dimensionality while retaining essential information. The final network structure consists of three fully connected layers, each with 50 neurons, and the output layer, which contains 5 neurons — one for each sentiment class corresponding to the star ratings (1 to 5). This configuration allows the RNN to process customer reviews and predict sentiment effectively.

5.1.5. Training CNN Model

For the Convolutional Neural Network (CNN) model, we structured it to capture the spatial hierarchies in the text data through convolution operations. Similar to the RNN, the model begins with an embedding layer that converts the words into vector representations. Following this, we applied a convolutional layer with a kernel size of 8, allowing the model to detect local patterns in the text, such as common word combinations or phrases associated with specific sentiments. After the convolutional layer, a pooling layer was added to reduce the feature space, followed by a flattening layer to prepare the data for the fully connected layers. The CNN's fully connected part consists of three dense layers, with widths of 50, 10, and 5 neurons, respectively, with the final layer producing the output for the 5 sentiment classes (1 to 5 stars). This architecture is well-suited for recognizing patterns in the customer reviews and predicting their sentiment classification.

5.2. Models Evaluation

We evaluate and compare the performance of five sentiment analysis models: Logistic Regression, Naive Bayes, Random Forest, RNN, and CNN, using a test set of customer reviews. For each model, we generated a confusion matrix and a classification report to assess key performance metrics such as accuracy, precision, recall, and F1-score. This allows us to understand how well each model performs in predicting sentiment classification across the 1- to 5-star rating scale. In addition to evaluating these models, we also evaluate the sentiment polarity outputs from NLTK and TextBlob, two widely-used text analysis tools. Finally, we provide a detailed comparison of all models to identify which approach offers the best results for sentiment analysis in this context.

5.2.1. Logistic Regression Classification Model Evaluation

The Logistic Regression model was evaluated using the test dataset, and its performance was measured using a confusion matrix and classification report. The confusion matrix is shown in Table 2.

Table 2. The confusion matrix of Logistic Regression sentiment classification model.

True/Predicted	1	2	3	4	5
1	665	0	1	0	0
2	0	566	8	2	2
3	0	0	1224	4	12
4	0	0	14	1754	13
5	0	0	3	34	2109

This matrix shows that the model performs well across all classes, particularly for class 1 (1-star reviews), where it has perfect precision and recall. The classification report, summarizing the model's precision, recall, F1-score, and support for each star rating, is provided in Table 3.

Table 3. The classification report of Logistic Regression sentiment classification model.

Class	Precision	Recall	F1	Support
1	1.00	1.00	1.00	666
2	1.00	0.98	0.99	578
3	0.98	0.99	0.98	1240
4	0.98	0.98	0.98	1781
5	0.99	0.98	0.99	2146
Accuracy		0.99		6411
Macro Avg	0.99	0.99	0.99	6411
Weighted Avg	0.99	0.99	0.99	6411

The Logistic Regression model achieves an overall accuracy of 99%, with a high level of consistency across all classes. The macro average for precision, recall, and F1-score is 0.99, indicating balanced performance. The weighted average, accounting for class support, also shows 99% across these metrics, demonstrating that Logistic Regression is an effective model for customer review sentiment analysis.

5.2.2. Naive Bayes Classification Model Evaluation

The Naive Bayes classification model showed a relatively lower overall accuracy compared to Logistic Regression model, achieving 84% accuracy on the test set. The confusion matrix in Table 4 shows that the Naive Bayes model struggles more with correctly classifying reviews into the proper star categories, particularly for classes 4 and 5.

Table 4. The confusion matrix of Naive Bayes sentiment classification model

True/Predicted	1	2	3	4	5
1	587	9	24	9	37
2	1	440	64	11	62
3	7	3	1047	31	152
4	17	11	70	1260	423
5	16	0	49	42	2039

The confusion matrix indicates that the model misclassified several instances across different ratings. For example, in the 4-star class, 423 reviews were misclassified as 5-star reviews, while 70 reviews were misclassified as 3-star reviews. Similarly, for the 3-star rating, 152 reviews were misclassified as 5-star reviews, showing that the model has difficulty distinguishing between these ratings.

The classification report in Table 5 further shows that while precision and recall values for the 1-star ratings are high, the 2-star and 4-star categories exhibit lower recall values.

Table 5. The classification report of Naive Bayes sentiment classification model.

Class	Precision	Recall	F1	Support
1	0.93	0.88	0.91	666
2	0.95	0.76	0.85	578
3	0.83	0.84	0.84	1240
4	0.93	0.71	0.80	1781
5	0.75	0.95	0.84	2146
Accuracy		0.84		6411
Macro Avg	0.88	0.83	0.85	6411
Weighted Avg	0.85	0.84	0.84	6411

The model achieves its best performance with the 1-star and 2-star ratings, with an F1-score of 0.91 and 0.85, respectively, indicating good precision in these cases. However, the F1-scores for the 4-star (0.80) and 5-star (0.84) ratings reveal the model's challenges in classifying higher-star reviews accurately.

5.2.3. Random Forest Classification Model Evaluation

The Random Forest model performed exceptionally well on the test set, with nearly perfect classification across all star ratings. The confusion matrix in Table 6 shows that the majority of predictions align exactly with the true labels, indicating high accuracy. For example, for 1-star reviews, the model correctly classified 665 out of 666 test reviews, with no misclassifications in any other class. Similarly, for 5-star reviews, 2135 out of 2146 were correctly predicted, with only a few misclassifications into 3- and 4-star classes.

Table 6. The confusion matrix of Random Forest sentiment classification model.

True/Predicted	1	2	3	4	5
1	665	0	1	0	0
2	0	570	6	1	1
3	0	0	1224	4	12
4	0	0	0	1774	7
5	0	0	2	9	2135

The classification report in Table 7 further confirms this strong performance, with precision, recall, and F1-scores close to or at 1.00 across all classes.

Table 7. The classification report of Random Forest sentiment classification model.

Class	Precision	Recall	F1	Support
1	1.00	1.00	1.00	666
2	1.00	0.99	0.99	578
3	0.99	0.99	0.99	1240
4	0.99	1.00	0.99	1781
5	0.99	0.99	0.99	2146
Accuracy		0.99		6411
Macro Avg	1.00	0.99	0.99	6411
Weighted Avg	0.99	0.99	0.99	6411

The overall accuracy of the model is 99%, and the macro and weighted averages also indicate strong performance across the board. The precision and recall for each star rating are exceptionally high, demonstrating that the Random Forest classification model is effective in both minimizing false positives and capturing true positives. This level of performance suggests that the Random Forest model is highly reliable for customer review sentiment analysis in this dataset.

5.2.4. RNN Classification Model Evaluation

The RNN model was evaluated on the test set using a confusion matrix and a classification report. The confusion matrix for the RNN model is shown in Table 8.

Table 8. The confusion matrix for RNN sentiment classification model.

True/Predicted	1	2	3	4	5
1	684	4	1	0	0
2	5	554	3	1	1
3	0	8	1154	12	11
4	0	1	4	1807	8
5	0	1	2	37	2113

From the confustion matrix, we observe that the RNN model performs quite well across all classes, with very few misclassifications. Most predictions align with the true labels, with high precision and recall across all star ratings. The classification report in Table 9 further details the performance metrics for each class.

Table 9. The classification report for RNN sentiment classification model.

Class	Precision	Recall	F1	Support
1	0.99	0.99	0.99	689
2	0.98	0.98	0.98	564
3	0.99	0.97	0.98	1185
4	0.97	0.99	0.98	1820
5	0.99	0.98	0.99	2153
Accuracy		0.98		6411
Macro Avg	0.98	0.98	0.98	6411
Weighted Avg	0.98	0.98	0.98	6411

The RNN model achieves an accuracy of 98%, with balanced precision and recall across all classes. The macro and weighted averages are also around 98%, showcasing the RNN model's robustness in handling a diverse set of customer reviews. The model's ability to capture sequential information contributes to its success in sentiment analysis, effectively recognizing the context and structure of customer reviews.

5.2.5. CNN Classification Model Evaluation

The confusion matrix in Table 10 provides insights into CNN model's performance across the five-star rating categories.

Table 10. The confusion matrix for CNN sentiment classification model.

True/Predicted	1	2	3	4	5
1	688	0	1	0	0
2	10	551	0	0	3
3	60	7	1112	0	6
4	0	1	0	1817	2
5	0	0	346	6	1801

The matrix shows that the CNN model performed well in predicting 2-star and 4-star ratings, with minimal misclassification. However, there was some misclassification in the 3-star and 5-star ratings, with 60 reviews from class 3 being misclassified into class 1, and 346 reviews from class 5 being misclassified into class 3.

The classification report for the CNN model, as shown in Table 11, indicates strong performance across all classes, with an overall accuracy of 93%.

Table 11. The classification report for CNN sentiment classification model.

Class	Precision	Recall	F1	Support
1	0.91	1.00	0.95	689
2	0.99	0.98	0.98	564
3	0.76	0.94	0.84	1185
4	1.00	1.00	1.00	1820
5	0.99	0.84	0.91	2153
Accuracy		0.93		6411
Macro avg	0.93	0.95	0.94	6411
Weighted avg	0.94	0.93	0.93	6411

The model achieved a perfect precision and recall score of 1.00 for class 4, indicating excellent prediction capability in this category. Classes 2 and 5 showed high precision, but the F1-score for class 3 was lower at 0.84, reflecting the model's challenges in correctly classifying 3-star reviews. Despite these challenges, the weighted average precision, recall, and F1-score were all above 0.93, highlighting the CNN model's overall performance in the sentiment analysis task.

5.3. Model Comparison and Discussion

In this subsection, we compare the performance of the five classification models we trained: Logistic Regression, Naive Bayes, Random Forest, RNN, and CNN. Additionally, we include sentiment polarity scores from NLTK and TextBlob for further comparison.

NLTK produces compound polarity scores ranging from -1 to 1, where scores below 0 indicate negative sentiment, a score of 0 indicates neutral sentiment, and scores above 0 indicate positive sentiment [26,27]. Similarly, TextBlob sentiment polarity scores categorize sentiment as negative (score < 0), neutral (score = 0), or positive (score > 0) [28,29]. Table 12 and Table 13 show the confusion matrix and classification report of NLTK sentiment classification models on the entire dataset.

Table 12. The confusion matrix of NLTK polarity.

True/Predicted	Negative	Neural	Positive
Negative	1632	663	4053
Neural	1007	924	4165
Positive	1586	1477	16547

Table 13. The classification report of NLTK sentiment classification model.

	Precision	Recall	F1-Score	Support
Negative	0.39	0.26	0.31	6348
Neural	0.30	0.15	0.20	6096
Positive	0.67	0.84	0.75	19610
Accuracy		0.60		32054
Macro Avg	0.45	0.42	0.42	32054
Weighted Avg	0.54	0.60	0.56	32054

Tables 14 and 15 show the confusion matrix and classification report of TextBlob sentiment classification models on the entire dataset.

Table 14. The confusion matrix of TextBlob polarity.

True/Predicted	Negative	Neural	Positive
Negative	2383	832	3133
Neural	1303	1613	3180
Positive	2918	2589	14103

Table 15. The classification report of TextBlob sentiment classification model.

	Precision	Recall	F1-Score	Support
Negative	0.36	0.38	0.37	6348
Neural	0.32	0.26	0.29	6096
Positive	0.69	0.72	0.70	19610
Accuracy		0.56		32054
Macro Avg	0.46	0.45	0.45	32054
Weighted Avg	0.56	0.56	0.56	32054

Table 16 summarizes the accuracy of all seven sentiment classification models.

Table 16. The comparison of accuracy of 7 sentiment classification models.

Model	Accuracy
Logistic Regression	0.99
Naive Bayes	0.84
Random Forest	0.99
RNN	0.98
CNN	0.93
NLTK	0.60
TextBlob	0.56

In comparing the five machine learning and deep learning models, we observe variations in accuracy, precision, recall, and F1-scores that highlight each model's strengths in sentiment analysis. Among the machine learning models, both Random Forest and Logistic Regression achieved a high accuracy of 0.99, indicating their capability to capture and differentiate sentiment patterns in the dataset. Random Forest performs slightly better in overall accuracy than Logistic Regression and shows strong precision and F1-scores due to its ensemble approach.

The experimental result also reveals that deep learning models (RNN and CNN) do not outperform traditional machine learning models (Random Forest and Logistic Regression) in this sentiment analysis task. RNN, with an accuracy of 0.98, closely follows these machine learning models, demonstrating its ability to handle sequential data with a high degree of accuracy, although it did not surpass them in this case. CNN, at 0.93 accuracy, performs well but slightly lower than RNN and the top-performing machine learning models, which could be due to its focus on local feature detection rather than the sequential patterns RNN captures.

The evaluation results demonstrate that our machine learning and deep learning models outperform both NLTK and TextBlob in terms of accuracy and other performance metrics. The primary reason for this performance is that our models are trained specifically on the review datasets, allowing them to develop a comprehensive feature dictionary tailored to the sentiment expressed within those reviews. This training process enables the models to learn the nuances and context of the data, capturing relationships between words and their contributions to sentiment. In contrast, NLTK and TextBlob rely on a predefined dictionary that do not adapt to the specific characteristics of the review data. As a result, they lack the depth of understanding that comes from training on a specialized dataset, leading to less accurate predictions. Thus, the models we implemented not only provide better accuracy but also demonstrate a greater ability to generalize sentiment based on the specific language and context of the customer reviews.

6. Conclusions and Future Work

In this study, we conducted a comparative analysis of five classification models, including three traditional machine learning algorithms (Logistic Regression, Naive Bayes, and Random Forest) and two deep learning models (RNN and CNN), to assess their effectiveness in customer review sentiment analysis. Our findings show that Random Forest and Logistic Regression achieved the highest accuracy (0.99), closely followed by RNN (0.98) and CNN (0.93), while Naive Bayes scored a lower accuracy of 0.84. Additionally, we evaluated NLTK and TextBlob sentiment analysis tools, which provided insights but achieved lower accuracy scores of 0.60 and 0.56, respectively, due to their reliance on general sentiment lexicons rather than specific model training.

The high performance of Random Forest and Logistic Regression demonstrates that traditional machine learning models can effectively classify sentiment in customer reviews when properly trained on specific domain data. While deep learning models offer strengths in handling sequential data and complex language structures, machine learning models outperform in this case, potentially due to their efficient feature extraction and classification mechanisms. This study emphasizes the value of selecting models based on dataset characteristics and computational efficiency, as well as the benefit of model-specific training over generic sentiment analysis tools.

Future work may explore more complex architectures, larger datasets, or hybrid models that integrate the sequential capabilities of deep learning with the robust feature extraction of traditional machine learning. This approach could further enhance performance and adaptability across different sentiment analysis tasks.

Author Contributions: Conceptualization, Y.Z.; methodology, Y.Z.; investigation, L.A.; data curation, L.A.; experiments, L.A.; writing—original draft preparation, L.A.; writing—review and editing, Y.Z.; supervision, Y.Z; All authors have read and agreed to the published version of the manuscript.

Funding: The research did not receive external funding.

Data Availability Statement: Data available upon request from the authors.

Acknowledgments: The research project was supported by CSUSB Undergraduate Success Research Assistantship Program (USRA), funded by the Learning Aligned Employment Program (LAEP) in the fall of 2023 and the spring of 2024. We thank the High Performance Computing Program team at CSUSB, especially Dr. Dung Vu for his support.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Al-Otaibi, S.; Alnassar, A.; Alshahrani, A.; Al-Mubarak, A.; Albugami, S.; Almutiri, N.; Albugami, A. Customer satisfaction measurement using sentiment analysis. *International Journal of Advanced Computer Science and Applications* **2018**, *9*.
2. Verma, S. Sentiment analysis of public services for smart society: Literature review and future research directions. *Government Information Quarterly* **2022**, *39*, 101708.
3. Haque, T.U.; Saber, N.N.; Shah, F.M. Sentiment analysis on large scale Amazon product reviews. 2018 IEEE international conference on innovative research and development (ICIRD). IEEE, 2018, pp. 1–6.
4. Rashid, A.; Huang, C.Y. Sentiment Analysis on Consumer Reviews of Amazon Products. *International Journal of Computer Theory and Engineering* **2021**, *13*, 35–41.
5. AlQahtani, A.S. Product sentiment analysis for amazon reviews. *International Journal of Computer Science & Information Technology (IJCSIT)* **2021**, *13*, 1–16.
6. Kumar, K.S.; Desai, J.; Majumdar, J. Opinion mining and sentiment analysis on online customer review. 2016 IEEE international conference on computational intelligence and computing research (ICCIC). IEEE, 2016, pp. 1–4.
7. Ali, H.; Hashmi, E.; Yayilgan Yildirim, S.; Shaikh, S. Analyzing amazon products sentiment: a comparative study of machine and deep learning, and transformer-based techniques. *Electronics* **2024**, *13*, 1305.
8. Tan, W.; Wang, X.; Xu, X. Sentiment Analysis for Amazon Reviews. https://cs229.stanford.edu/proj2018/, 2018. Online; accessed Aug 28, 2024.
9. Park, S.; Cho, J.; Park, K.; Shin, H. Customer sentiment analysis with more sensibility. *Engineering Applications of Artificial Intelligence* **2021**, *104*, 104356.
10. Dieksona, Z.A.; Prakosoa, M.R.B.; Qalby, M.S.; Putraa, M.; Achmada, S.; Sutoyoa, R. Sentiment analysis for customer review: Case study of Traveloka. *Procedia Computer Science* **2023**, *216*, 682–690.
11. Grljević, O.; Bošnjak, Z. Sentiment analysis of customer data. *Strategic Management - International Journal of Strategic Management and Decision Support Systems in Strategic Management* **2018**, *23*.
12. Chinnalagu, A.; Durairaj, A.K. Context-based sentiment analysis on customer reviews using machine learning linear models. *PeerJ Computer Science* **2021**, *7*, e813.
13. Obiedat, R.; Qaddoura, R.; Ala'M, A.Z.; Al-Qaisi, L.; Harfoushi, O.; Alrefai, M.; Faris, H. Sentiment analysis of customers' reviews using a hybrid evolutionary SVM-based approach in an imbalanced data distribution. *IEEE Access* **2022**, *10*, 22260–22273.
14. Pampel, F.C. *Logistic regression: A primer*; Number 132, Sage publications, 2020.
15. Bahtiar, S.A.H.; Dewa, C.K.; Luthfi, A. Comparison of Naïve Bayes and Logistic Regression in Sentiment Analysis on Marketplace Reviews Using Rating-Based Labeling. *Journal of Information Systems and Informatics* **2023**, *5*, 915–927.
16. Murphy, K.P.; others. Naive bayes classifiers. *University of British Columbia* **2006**, *18*, 1–8.
17. Webb, G.I.; Keogh, E.; Miikkulainen, R. Naïve Bayes. *Encyclopedia of machine learning* **2010**, *15*, 713–714.
18. Belgiu, M.; Drăguţ, L. Random forest in remote sensing: A review of applications and future directions. *ISPRS journal of photogrammetry and remote sensing* **2016**, *114*, 24–31.
19. Rigatti, S.J. Random forest. *Journal of Insurance Medicine* **2017**, *47*, 31–39.
20. Paulus, R.; Socher, R.; Manning, C.D. Global belief recursive neural networks. *Advances in Neural Information Processing Systems* **2014**, *27*.
21. Kurniasari, L.; Setyanto, A. Sentiment analysis using recurrent neural network. *Journal of Physics: Conference Series* **2020**, *1471*, 012018.
22. Bowman, S.; Potts, C.; Manning, C.D. Recursive neural networks can learn logical semantics. Proceedings of the 3rd workshop on continuous vector space models and their compositionality, 2015, pp. 12–21.
23. Krichen, M. Convolutional neural networks: A survey. *Computers* **2023**, *12*, 151.
24. Liao, S.; Wang, J.; Yu, R.; Sato, K.; Cheng, Z. CNN for situations understanding based on sentiment analysis of twitter data. *Procedia computer science* **2017**, *111*, 376–381.
25. Saxena, A. An introduction to convolutional neural networks. *Int. J. Res. Appl. Sci. Eng. Technol* **2022**, *10*, 943–947.
26. Yao, J. Automated Sentiment Analysis of Text Data with NLTK. *Journal of Physics: Conference Series* **2019**, *1187*, 052020. doi:10.1088/1742-6596/1187/5/052020.

 doi:10.20944/preprints202411.0741.v1

27. Vencer, L.V.T.; Bansa, H.; Caballero, A.R. Data and Sentiment Analysis of Monkeypox Tweets using Natural Language Toolkit (NLTK). The 8th International Conference on Business and Industrial Research (ICBIR 2023). IEEE, 2023, pp. 392–396.
28. Nehal, N.; Jeet, D.; Sharma, V.; Mishra, S.; Iwendi, C.; Osamor, J. Twitter sentiment analysis and emotion detection using NLTK and TextBlob. The 4th International Conference on Computation, Automation and Knowledge Management (ICCAKM 2023). IEEE, 2023. doi:10.1109/ICCAKM58659.2023.10449540.
29. Mahgoub, A.; Atef, H.; Nasser, A.; Yasser, M.; Medhat, W.M.; Darweesh, M.S.; El-Kafrawy, P.M. Sentiment analysis: Amazon electronics reviews using bert and textblob. The 20th International Conference on Language Engineering (ESOLEC 2022). IEEE, 2022, Vol. 20, pp. 6–10.