electronics

Article

Analyzing Amazon Products Sentiment: A Comparative Study of Machine and Deep Learning, and Transformer-Based Techniques

Hashir Ali [1], **Ehtesham Hashmi** [2,*], **Sule Yildirim Yayilgan** [2] **and Sarang Shaikh** [2]

[1] Department of Computer Science, The University of Lahore, Lahore 54590, Punjab, Pakistan; hashirali129@gmail.com

[2] Department of Information Security and Communication Technology (IIK), Norwegian University of Science and Technology (NTNU), 2815 Gjøvik, Norway; sule.yildirim@ntnu.no (S.Y.Y.); sarang.shaikh@ntnu.no (S.S.)

[*] Correspondence: hashmi.ehtesham@ntnu.no

Abstract: In recent years, online shopping has surged in popularity, with customer reviews becoming a crucial aspect of the decision-making process. Reviews not only help potential customers make informed choices, but also provide businesses with valuable feedback and build trust. In this study, we conducted a thorough analysis of the Amazon reviews dataset, which includes several product categories. Our primary objective was to accurately classify sentiments using natural language processing, machine learning, ensemble learning, and deep learning techniques. Our research workflow encompassed several crucial steps. We explore data collection procedures; preprocessing steps, including normalization and tokenization; and feature extraction, utilizing the Bag-of-Words and TF–IDF methods. We conducted experiments employing a variety of machine learning algorithms, including Multinomial Naive Bayes, Random Forest, Decision Tree, and Logistic Regression. Additionally, we harnessed Bagging as an ensemble learning technique. Furthermore, we explored deep learning-based algorithms, such as CNNs, Bidirectional LSTM, and transformer-based models, like XLNet and BERT. Our comprehensive evaluations, utilizing metrics such as accuracy, precision, recall, and F1 score, revealed that the BERT algorithm outperformed others, achieving an impressive accuracy rate of 89%. This research provides valuable insights into the sentiment analysis of Amazon reviews, aiding both consumers and businesses in making informed decisions and enhancing product and service quality.

Keywords: data interpretation; deep learning; ensemble learning; machine learning; product sentiments; sentiment analysis; transformers

check for updates

Citation: 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. https://doi.org/10.3390/electronics13071305

Academic Editors: Dimitris Apostolou and Ruifeng Xu

Received: 10 March 2024
Revised: 27 March 2024
Accepted: 29 March 2024
Published: 31 March 2024

1. Introduction

The popularity of online commerce is on the rise, as evidenced by a recent study indicating that the count of digital shoppers has reached 2.64 billion in 2023 [1]. The rise of the internet has significantly contributed to the increase in user-generated content, as it allows individuals to express their opinions and engage in discussions across various platforms, such as blogs, online social networks, e-commerce websites, and forums. This trend has resulted in a vast amount of user-generated data [2,3]. Individuals, organizations, and governments must discern and utilize essential information from these data. The growing volume of these data highlights the challenge of efficiently and timely collecting relevant information, thereby emphasizing the need for computational linguistic approaches in data analysis [4].

When people shop online, they often rely on customer reviews to make decisions. Reviews provide valuable insights into product and service quality and customer experience. Positive reviews help businesses attract new customers and build trust. Negative reviews are also useful because they can provide feedback to companies on how to improve their products and services. Overall, customer reviews are a valuable tool for businesses and consumers [5]. They help businesses improve their products and services, as well as help consumers make informed purchasing decisions.

Sentiment Analysis (SA) is a process of determining the emotions or thoughts of a text. This can be conducted for customer reviews, social media posts, or other types of text [6,7]. SA allows you to gain insights from product reviews by identifying positive, negative, and neutral sentiments in comments. This information can be used to improve your product, identify areas for improvement, and understand what your customers are looking for. SA faces big challenges when it comes to understanding the different emotions in customer reviews. This is because people express themselves differently, and words like "great" or "bad" can mean different things depending on the situation. This can make it hard to obtain accurate insights into how satisfied customers are, which makes it tough to improve products. In today's digital world where user reviews are so important, it is crucial to be able to quickly analyze a lot of them. Current methods struggle to tell exactly how people feel, especially when the emotions are kind of in the middle, between positive and negative. That is why we need a system that can understand the emotions in context, so it can make accurate predictions and keep working well over time.

The primary objective of our project is to construct a robust SA model capable of categorizing product reviews into three predefined sentiment categories: positive, negative, or neutral. To achieve this, we have curated a dataset of 400,000 Amazon product reviews, meticulously performed data cleaning, and prepared the text data for analysis. Our approach involves harnessing a suite of natural language processing (NLP) techniques for feature extraction, encompassing tokenization; vectorization methods, such as Term Frequency-Inverse Document Frequency (TF–IDF) and Bag-of-Words (BOW); and even the potential integration of additional meta-information, such as product category or user ratings. Furthermore, our experimentation spans a diverse array of machine learning (ML), ensemble learning, and deep learning (DL) algorithms, ranging from traditional methods like Logistic Regression (LR), and Random Forest (RF) to advanced transformer-based models like BERT and XLNet. This comprehensive approach ensures the development of a robust and accurate sentiment analysis system that empowers businesses to make informed decisions, enhance their products, and, ultimately, elevate customer satisfaction.

In our research, our main contribution lies in improving and applying well-known sentiment detection methods by incorporating regularization techniques and hyperparameter tuning. We carefully applied this approach to a baseline dataset containing categories such as positive, neutral, and negative sentiments, with the goal of substantially improving sentiment detection accuracy (particularly for Amazon reviews) within these specific categories. We employ Bag-of-Words and TF–IDF vectorization techniques and utilize a range of machine learning (ML) and deep learning (DL) models, including Multinomial Naive Bayes (MNB), Random Forest (RF), Decision Tree (DT), Logistic Regression (LR), ensemble learning techniques like bagging, and deep learning methods such as Convolutional Neural Networks (CNNs) and Bidirectional Long Short-Term Memory (Bi-LSTM). Furthermore, we explore transformer-based models like BERT and XLNet, where BERT outperformed all other machine learning and deep learning models in this research. We calculated and evaluated all the results based on their accuracy, precision, recall, and F1 score.

This paper is structured as follows: Section 2 presents information on related work, Section 3 describes the methodology, Section 4 discusses the evaluation measurements employed for this project, Section 5 delves into the discussion of our research findings and results, and Section 6 outlines potential avenues for future research and development.

2. Related Work

Numerous research papers in the field of SA focus on various methods for extracting features and classifying review polarity within datasets. According to Scopus analysis, the quantity of documents related to SA is increasing annually. Details of the published documents by 2023 are provided in Table 1. This shows that more and more people are interested in understanding and analyzing emotions in written content.

Table 1. Published documents from 2010 to 2023.

Document Type	Documents
Article	54,839
Review	7789
Conference Paper	7727
Book Chapter	5388
Book	4391
Editorial	191
Note	162
Short Survey	92
Data Paper	60
Letter	53

Y. Abbas et al. [8] developed a supervised ML method to spot defective products in online reviews, using a dataset of Amazon reviews across four product categories. They applied statistical techniques like correlation [9] and information gain [10] for feature selection, focusing on features like emotional tone and emotions. Their best results came from the Random Forest classifier, with a 0.84 accuracy score. Meanwhile, C. Ahmed et al. [11] proposed a hybrid ML framework to analyze customer–service provider interactions, aiming to predict sentiment changes. They examined 5000 conversations on Twitter. In their study, D. Suhartono et al. [12] explored a three-category classification challenge in SA of pharmaceutical product reviews using Deep Neural Networks (DNNs) and weighted word embeddings. They incorporated two different word embedding methods, GLOVE [13] and Word2Vec [14], and processed these embeddings through multiple layers of Convolutional Neural Networks (CNNs).

K. S. Kumar et al. [15] mined Amazon reviews for three products, namely, the Apple iPhone 5S, Samsung J7, and Redmi Note 3, employing ML models and discovering that Naïve Bayes (NB) outperformed LR in categorizing reviews as positive or negative, evaluated through recall, precision, and F-measure. M. Qorich et al. [16] tackled text sentiment analysis on Amazon reviews by combining word embeddings with CNNs. Their study utilized two-word embedding techniques, namely, FastText [17] and Word2Vec, and applied these to three different datasets. The CNN model they developed showed enhanced performance over traditional ML- and DL-based methods, outperforming the established baselines in all datasets. Xu Yun et al. [18] from Stanford University utilized supervised learning algorithms, including the perceptron algorithm, NB, and Support Vector Machine (SVM), to predict Yelp review ratings, utilizing a 70–30% training–testing split for cross-validation and evaluating multiple classifiers to determine precision and recall values. A. S. Rathor et al. [19] randomly selected 3000 English Amazon reviews from a pool of 21,500. These reviews underwent preprocessing, which included the removal of repeated letters. Subsequently, the reviews were classified into positive, negative, or neutral categories using SVM, NB, and Maximum Entropy (ME) algorithms.

M. S. Elli et al. [20] achieved high accuracy by extracting sentiment from reviews to develop a business model, primarily utilizing MNB and SVM as classifiers. A. Cernian et al. [21] analyzed 300 Amazon electronic device reviews, applying SentiWordNet for phrase-to-word vector conversion and assessing LR, SGD, NB, and CNNs with various feature extraction methods, with CNNs and Word2Vec achieving 91% accuracy; Lime was used to explain review classifications. M. Nasret al. [22] opted for simpler algorithms to enhance comprehensibility, although they excelled in accuracy when combined with SVM, potentially facing challenges on larger datasets. W. Tan et al. [23] explored the correlation between Amazon product reviews and customer ratings, leveraging diverse ML algorithms, including NB analysis, SVM, K-Nearest Neighbor (KNN) methods, and DNNs like RNN. Finally, in [24], over 100,000 Chinese clothing product reviews from Amazon underwent processing, including text segmentation, POS tagging, and stop word and punctuation removal using the ICTCLAS 4 system. The approach involved sentiment classification using Word2Vec for semantic feature extraction and SVMperf for comment text classification.

Huang et al. [25] achieved a sentiment classification accuracy of 64.1%, focusing specifically on long-range contexts in the Weibo tweet texts dataset. They accomplished this by utilizing a Hierarchical LSTM network. Hasan et al. [26] utilized unigram features in conjunction with the Naive Bayes algorithm. They applied this approach to translated Urdu tweet data and employed various sentiment analyzers, such as SentiWordNet, Word Sense Disambiguation, and TextBlob. As a result of their efforts, they achieved a binary sentiment classification accuracy of 79%. Deriu et al. [27] incorporated a CNN model that consists of two convolutional layers followed by two consecutive pairs of pooling layers. This CNN model was applied to classify multilingual sentiment datasets composed of tweet data, and it achieved an F1-score of 67.79%. Jin et al. [28] enhanced the BERT-based model in a multi-label classification task by integrating BERT embeddings with a modified TF–IDF model. They conducted an evaluation using a dataset of customer reviews for restaurants, achieving an accuracy rate of 64%. Ouyang et al. [29] introduced a framework that combines Word2Vec with a CNN model, featuring three sets of convolutional layers and pooling layers. They applied this framework to classify five labels within the MR dataset corpus, achieving a fine-grained sentiment accuracy of 45.4%.

In this study, we used a large database containing products from various categories and applied several ML, ensemble learning, and DL-based techniques. Our main goal is to create a model that can achieve higher accuracy.

3. Methodology

This section offers a summary of the intended step-by-step approach for conducting SA on Amazon reviews. The process, as illustrated in Figure 1, consists of several stages. It initiates with data acquisition and subsequent preprocessing of the collected data. Following this, feature extraction techniques are applied, followed by the training of ML, ensemble learning, and DL models. Finally, the results of these models are evaluated. We enhanced the schema outlined in [30] and adapted it to create a new schema tailored to our study.

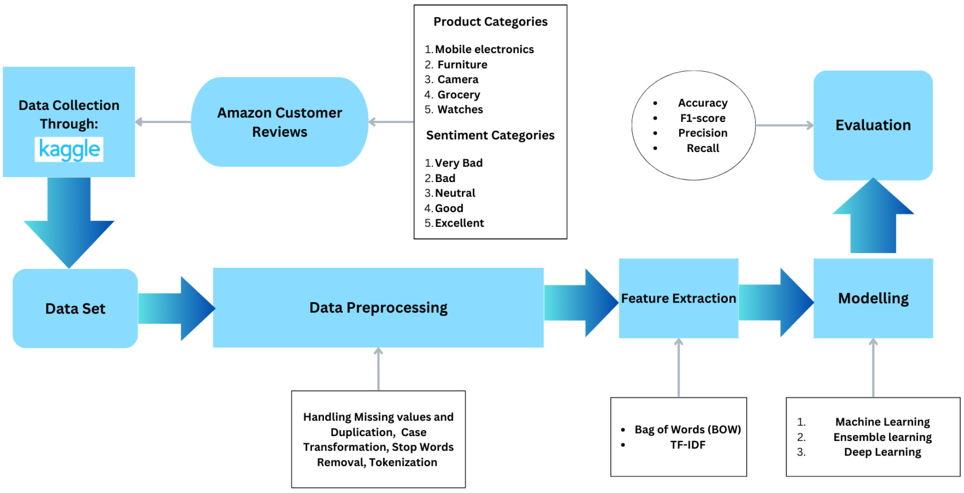


Figure 1. Proposed sentiment architecture for Amazon reviews.

3.1. Data Collection

We employed a publicly available dataset **sourced from Amazon**, accessible through the Kaggle platform, to conduct sentiment analysis (SA). The dataset contains reviews spanning five diverse product categories: mobile electronics, furniture, camera, grocery, and watches. These reviews, originating from Amazon, were publicly accessible on Kaggle [31]. The dataset is quite large, with more than 100 million reviews in total. However, we have chosen to focus on a subset of 80,000 reviews from each of these categories, resulting in a dataset of 400,000 records in total. Each record within the extracted Amazon reviews dataset is associated with a set of attributes, including marketplace information, customer-ID, review-ID, product-ID, product parent, product title, product category, star rating, helpful votes, total votes, verification status of the purchase, review headline, and review body. These attributes collectively provide a comprehensive foundation for conducting an in-depth analysis as part of this research endeavor.

3.2. Exploratory Data Analysis

Exploratory data analysis (EDA), a term coined by John W. Tukey, is the practice of thoroughly examining data to uncover patterns and insights without making assumptions [32]. It is an essential first step in data analysis. In our thorough look at the data, we carefully studied how people rated something using stars. Here is what we found: Based on the data presented in Figure 2, we can observe a clear distribution of star ratings in the reviews. The majority, approximately 61.2%, received a 5-star rating, indicating a high level of satisfaction among reviewers. Another substantial portion, about 15.4%, was rated with 4 stars, suggesting that a significant number of people were quite pleased. Approximately 7.8% of the reviews received a 3-star rating, indicating a moderate level of satisfaction. A smaller segment, around 5.2%, received 2 stars, implying a lower level of satisfaction. Finally, 10.4% of the reviews were awarded just 1 star, signaling a substantial degree of dissatisfaction.

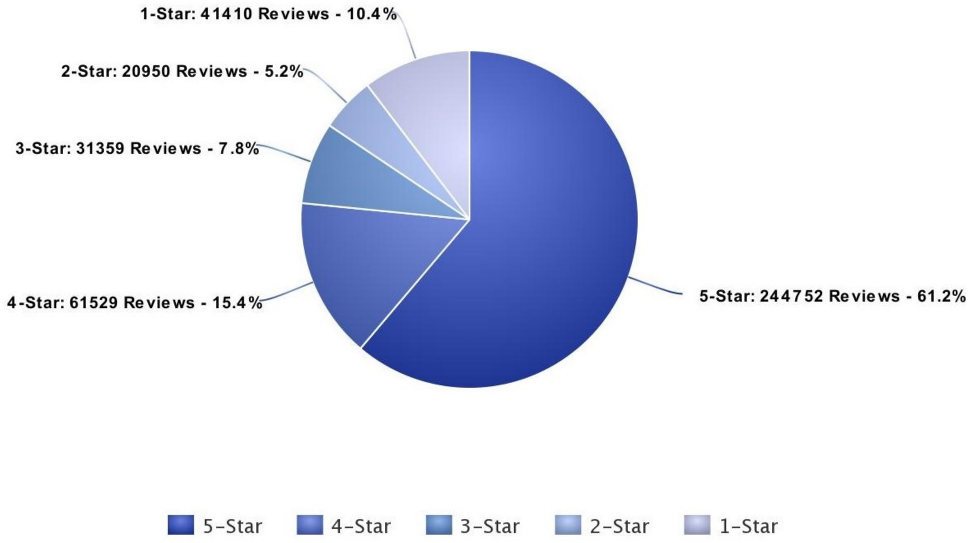


Figure 2. Star rating distribution of reviews.

Upon scaling the dataset to categorize reviews by polarity, which involves merging 4- and 5-star ratings into the positive category, 2- and 1-star ratings into the negative category, and assigning 3-star ratings to the neutral category, we discern a distinct distribution. Positive reviews constitute 76.6% of the dataset, reflecting a substantial prevalence of highly favorable sentiments among reviewers. Simultaneously, neutral reviews make up 7.8% of the dataset, indicating a relatively balanced sentiment, where reviewers express neither overwhelmingly positive nor negative views. Conversely, negative reviews account for 15.6% of the dataset, revealing a substantial presence of notably unfavorable sentiments. This polarity analysis is further substantiated by the visual representation in Figure 3, which serves to reinforce our assessment of the dataset's sentiment composition.

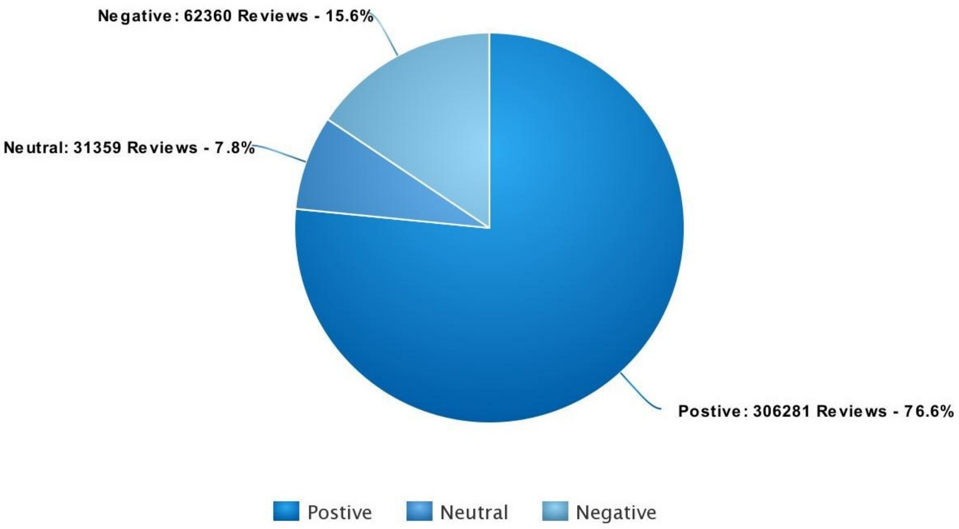


Figure 3. Sentiment classification of star ratings.

3.3. Data Preprocessing

Data preprocessing is a vital step in NLP tasks, optimizing the efficiency of knowledge discovery. It involves techniques like data cleaning, integration, transformation, and reduction, all aimed at preparing and improving the dataset for more effective analysis [33]. Figure 4 represents all preprocessing steps involved in this research.

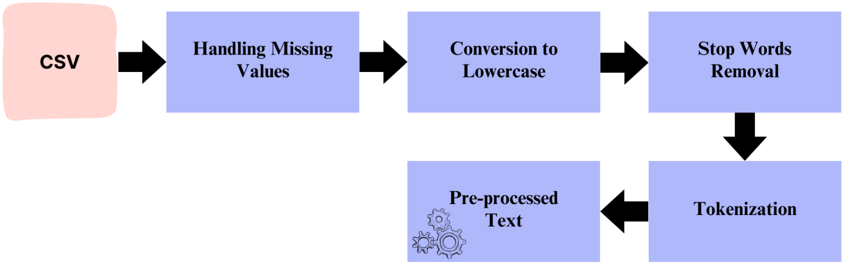


Figure 4. Data pre-processing steps.

3.3.1. Handling Missing Values

When addressing the issue of missing values within the dataset, our primary focus lies on managing the missing entries within the “review body” and “star rating” features. This emphasis is placed due to the significance of these features in relation to sentiments and their corresponding outputs. The specifics regarding the count of missing values for each feature in our dataset is detailed in Table 2.

For features with an object data type, we have employed the **fillna()** method available in Python to populate the null values. Additionally, for the “star rating” feature, we have opted to utilize the **Interpolate** method. This method calculates an average based on the values that surround the empty cell, both above and below it, thereby aiding in the imputation of missing data.

Table 2. Missing values.

Features	Data Type	Missing Values
Marketplace information	Object	0
Customer-ID	Object	0
Review-ID	Object	4
Product-ID	Object	6
Parent product details	Integer	6
Product title	Object	6
Product category	Object	6
Star rating	Integer	8
Helpful votes	Object	8
Total votes	Object	8
Verification status of the purchase	Object	8
Review headline	Object	13
Review body	Object	136

3.3.2. Lowercase Conversion

In this step, we convert all review words into lowercase. For example, “Great” and “aMazIng” are transformed into “great” and “amazing”. Lowercasing helps standardize the text and reduces the dimensionality of the data by treating words in a case-insensitive manner.

3.3.3. Removal of Stop Words

Stop words are elements in a sentence that hold no significance across all sectors in the field of text mining. We have eliminated all stop words, punctuation marks, and HTML tags from the reviews within our corpus. This preprocessing step helps to reduce noise in the data and improve computational efficiency.

3.3.4. Tokenization

In our research, we applied both sentence tokenization and word tokenization. Tokenization is a process where a sequence of text is broken down into individual components known as tokens. These tokens can encompass single words, phrases, or even entire sentences [34,35]. These tokens then serve as inputs for various processes such as parsing and text mining. It helps models focus on the meaning of individual units rather than processing the entire text as a single sequence.

In Figure 5, we have depicted the appearance of the text both before and after preprocessing.

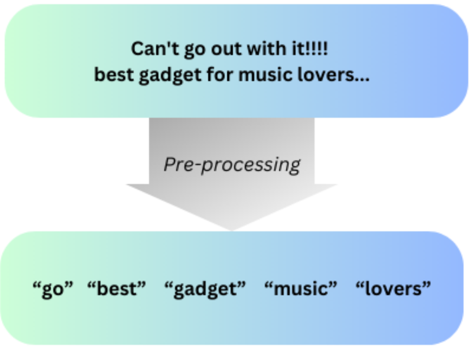

Can't go out with it!!!!
best gadget for music lovers...
Pre-processing
“go” “best” “gadget” “music” “lovers”

Figure 5. Reviews before and after preprocessing.

3.4. Feature Extraction

In the realm of NLP, feature extraction is a vital process where we transform unprocessed text into a numerical format that can be readily handled by ML algorithms. In this project, we have implemented **BOW** and **TF–IDF** extraction techniques.

In the context of sentiment analysis, the Bag-of-Words (BoW) approach plays a pivotal role in transforming textual data into a format that can be understood by machine learning algorithms. By representing each review as a "bag" containing the count of words without considering their order, we simplify the complexity of language for computational analysis. This method allows us to convert diverse and unstructured customer reviews into numerical vectors, forming the foundation for our sentiment analysis model. The BoW approach serves as a powerful tool in this project, enabling us to quantify the occurrence of specific words across reviews and extract meaningful patterns. While it may not capture the nuanced relationships between words, it provides a straightforward and efficient means to process large volumes of text, facilitating the extraction of valuable insights from the multitude of Amazon product reviews we aim to analyze. The formula for CBOW is given by:

$$P(w_t|\text{context}) = \text{softmax}\left(W_{\text{out}} \cdot \frac{1}{n}\sum_{i=1}^{n} W_{\text{in}}[w_{t-i}] + b_{\text{in}}\right) \tag{1}$$

where:

- $P(w_t|\text{context})$ is the conditional probability of the target word w_t, given its context.
- W_{in} is the input word embedding matrix.
- W_{out} is the output word embedding matrix.
- b_{in} is the bias term.
- w_{t-i} represents the context words for $i = 1, 2, \ldots, n$.
- The sum term in the formula computes the average of the word embeddings of the context words.

In this research, we also use TF–IDF to figure out which words are really important. TF–IDF looks at how often a word shows up in one review compared to how rare it is across all reviews. This helps us focus on words that truly matter for understanding feelings in each review. So, TF–IDF is like a smart way of picking out the words that tell us the most about what people think in their reviews.

$$\text{TF–IDF}(t, d, D) = \text{tf}(t, d) \cdot \text{idf}(t, D) \tag{2}$$

where:

- $\text{TF–IDF}(t, d, D)$ is the TF–IDF score for term t in document d with respect to the document set D.
- $\text{tf}(t, d)$ is the term frequency of term t in document d.
- $\text{idf}(t, D)$ is the inverse document frequency of term t in the document set D.

The term frequency $\text{tf}(t, d)$ is often computed as the ratio of the number of occurrences of term t in document d to the total number of terms in d. The inverse document frequency $\text{idf}(t, D)$ is computed as the logarithm of the ratio of the total number of documents in D to the number of documents containing term t.

4. Evaluating Measurements

The evaluation of classification methods can be determined through metrics such as **accuracy, F-score, recall, and precision**. These parameters are valuable for assessing the effectiveness of supervised ML algorithms. They rely on information from a matrix called the confusion matrix or contingency table [36], which provides a detailed breakdown of true positives, true negatives, false positives, and false negatives, enabling a comprehensive evaluation of the model's performance.

$$\text{Accuracy} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{3}$$

$$\text{Precision} = \frac{T_P}{T_P + F_P} \tag{4}$$

$$\text{Recall} = \frac{T_P}{T_P + F_N} \tag{5}$$

$$\text{F1-Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

5. Results and Discussion

In this section, we will discuss all the results obtained throughout our study. Firstly, we will delve into the outcomes of our ML models. Following that, we will explore the results achieved using DL techniques. Lastly, we will examine the performance of our transformer-based models.

5.1. Experimental Setup

Experiments were conducted using Google Colab with GPU support. Various models, including CNN, BI-LSTM, BERT, and XLNet, were implemented using Keras and PyTorch. Additionally, Scikit-learn was utilized to create machine learning models, which underwent hyperparameter tuning through grid search. Details of these models are outlined in Table 3, while Table 9 provides the configuration settings for transformer-based models.

Table 3. Configuration details for ML models.

Model	Regularization	Hyperparameter
DT	Split_min: [2, 5, 10]	GridSearchCV
RF	N-Estimators: [50, 100, 200]	GridSearchCV
LR	C: [1, 10, 100]	GridSearchCV

In Table 3, the "Split_min" values of 2, 5, and 10 for DT set the minimum number of samples required for a node split, impacting the tree's complexity and potential for overfitting. In RF, the "N-Estimators" parameter, with values 50, 100, and 200, determines the number of trees in the forest, balancing computational efficiency and model accuracy. For LR, the regularization parameter "C" is tested at values 1, 10, and 100. This parameter is crucial in controlling the strength of regularization, preventing overfitting by penalizing the magnitude of coefficients. Lower C values imply more regularization, constraining the model to simpler decision boundaries. The various parameters for different models were carefully fine-tuned using GridSearchCV, which involves an exhaustive search across specified parameter values. GridSearchCV systematically explores combinations of parameters, selecting the ones that result in the best performance metrics. This ensures that each model is precisely adjusted for optimal accuracy and generalization.

The dataset was split into an **80% training set** and a **20% testing set**. Models were optimized and trained on the training set before being evaluated on the testing set. Results for each model on the testing set were recorded.

5.2. Machine Learning

In our study, we used ML methods and popular algorithms to analyze sentiment in Amazon product reviews. We treated SA as a standard text categorization task, considering language and structure [37–43]. We tested various classifiers like MNB, RF, LR, and DT to assess their effectiveness in this context.

Originally, our reviews were categorized into five classifications, ranging from 1 (very bad) to 5 (excellent). For this research, we trained ML models based on this **five-category**

system. The results, obtained by applying vectorization techniques BOW and TF–IDF to ML algorithms, are detailed in Table 4.

Table 4. The results of the ML model on 5 classifications.

	Accuracy	F1-Score	Precision	Recall
MNB (BOW)	0.682	0.634	0.620	0.682
RF (BOW)	0.680	0.604	0.595	0.680
LR (BOW)	0.696	0.655	0.642	0.696
DT (BOW)	0.602	0.592	0.583	0.602
MNB (TF–IDF)	0.668	0.576	0.593	0.668
RF (TF–IDF)	0.679	0.596	0.593	0.679
LR (TF–IDF)	**0.702**	0.653	0.648	0.702
DT (TF–IDF)	0.607	0.595	0.584	0.607

As observed in the preceding table, the highest achievable accuracy stands at 70%, achieved through the LR classifier. However, it is essential to highlight that achieving a 70% accuracy rate may not be considered strong. This is mainly because the models struggle to tell apart 1-star and 2-star ratings, as they use similar negative words. The same issue arises with 4-star and 5-star ratings, as words like "good" and "great" are common to both, making it challenging for the model to provide accurate predictions. We tackled this problem by **merging the 4- and 5-star** ratings into a single category and the 1- and 2-star ratings into another. As a result, our dataset now consists of three groups: **positive, neutral,** and **negative**.

In this iteration, we commenced our analysis by categorizing the dataset into three distinct classes: positive, neutral, and negative. The outcomes of our investigation, achieved through the utilization of BOW vectorization representation and employing the same ML models, are meticulously documented in Table 5. This refinement in classification yielded noteworthy improvements in accuracy. Notably, the **LR** model stood out with a remarkable accuracy rate of **86%**.

Table 5. The results of the ML model on 3 classifications.

	Accuracy	F1-Score	Precision	Recall
MNB (BOW)	0.838	0.817	0.808	0.838
RF (BOW)	0.842	0.827	0.812	0.842
LR (BOW)	0.854	0.837	0.829	0.854
DT (BOW)	0.789	0.784	0.780	0.789
MNB (TF–IDF)	0.822	0.774	0.791	0.822
RF (TF–IDF)	0.846	0.828	0.811	0.846
LR (TF–IDF)	**0.861**	0.839	0.826	0.861
DT (TF–IDF)	0.787	0.782	0.777	0.787

5.3. K-Fold Cross-Validation

K-fold cross-validation is a technique in ML that helps us understand how well a model will work on new, unseen data. Instead of just using one set of data to train and test a model, we split our data into K equal parts (usually 5 or 10), and we train and test the model K times. Each time, a different part of the data is used for testing and the rest for training.

After employing a variety of ML models and methodologies, we proceeded to implement the k-fold cross-validation technique. The outcomes of all the models have been summarized in Table 6, where we employed a 5-fold cross-validation approach. In particular, the **LR** model performed better than all other models, earning a remarkable accuracy score of **85.9%**.

Table 6. The results of the ML model using k-fold on 3 classifications.

	Accuracy	**F1-Score**	**Precision**	**Recall**
MNB	0.823	0.776	0.790	0.823
RF	0.843	0.829	0.811	0.843
LR	**0.859**	0.842	0.828	0.859
DT	0.785	0.780	0.777	0.785

5.4. Ensemble Learning

In our study after applying ML techniques, we transitioned to ensemble learning and employed the same ML models using a **bagging technique**. When setting the number of estimators to 5, we achieved an impressive accuracy score of 85.4% with the LR classifier. The results of these experiments are listed in Table 7.

Table 7. The results of the ML model using bagging on 3 classifications.

	Accuracy	**F1-Score**	**Precision**	**Recall**
MNB	0.822	0.774	0.793	0.822
RF	0.851	0.832	0.829	0.851
LR	**0.854**	0.830	0.824	0.854
DT	0.817	0.802	0.792	0.817

In our research, we observed that a logistic regression classifier outperformed RF, MNB, and DT classifiers on our dataset for several reasons. Firstly, the dataset exhibited linear separability, making logistic regression, a linear classifier, highly effective. Additionally, the dataset was relatively large, and logistic regression's lower model complexity helped prevent overfitting, unlike RF, MNB, and DT. The linearity of the features and the simplicity of logistic regression made it robust against noisy or irrelevant data, providing a clear and interpretable model. Moreover, logistic regression could handle imbalanced data well, assign probabilities to classes, and was less sensitive to hyperparameters. These factors collectively contributed to the superior performance of logistic regression in our experiments.

5.5. Deep Learning

We transitioned from conventional ML methods, including ensemble learning, to embrace deep learning techniques, such as **CNNs** and **Bi-LSTM**. This shift was driven by the inherent strengths of deep learning in automatically extracting hierarchical features and representations from text data. These models demonstrate an adeptness at capturing intricate patterns and contextual nuances crucial for understanding sentiment in product reviews. These models, particularly the Bi-LSTM networks, are integral to sentiment analysis, excelling in understanding the sequential nature of language. By processing input data both forward and backward, Bi-LSTM captures long-range dependencies and contextual nuances within sentences. The results of our DL models, detailed in Table 8, underscore the superiority of DL in SA, showcasing higher accuracy and a better ability to discern the distinction of customer sentiments compared to conventional ML methods.

Table 8. The results of DL models on 3 classifications.

	Accuracy	**F1-Score**	**Precision**	**Recall**
CNN	0.849	0.835	0.821	0.849
BI-LSTM	**0.871**	0.861	0.855	0.871

The **CNN** shows the lowest accuracy at **85%**, which is interesting considering its primary design is for visual data. However, it surprisingly performs well in textual data analysis. Following closely is our **Bi-LSTM model**, achieving an impressive **87%** accuracy, surpassing our traditional ML models. In the upcoming section, we will discuss in detail our Bi-LSTM models to better understand its strengths and contributions to our research.

Bi-LSTM

Our sentiment analysis model, a Bi-LSTM setup, delivered an impressive 87% accuracy, and it did so after just **three rounds of training**. Let us break down the model's key elements and the layers it employs. In our SA model, we used different layers, each with its job to help the model understand and predict sentiments in text. First, we had the Embedding Layer, which changed the words in the reviews into numbers so the computer could understand them. Then, there was the Spatial Dropout Layer, which made sure the model did not get too good at one thing and helped it stay flexible. Next, we used a Bi-LSTM layer with 128 units to look at the words in the reviews from both sides, like reading a book forwards and backward, to catch all the details. We also added another Bi-LSTM layer with 64 units to dive even deeper into the reviews. After that, we had a layer with ReLU activation to make the model smarter and an Output Layer to predict whether the sentiment is positive, negative, or neutral. Altogether, these layers helped our model understand reviews, find patterns, and make accurate predictions.

To assess our model's ability to distinguish between different sentiments, we utilized a confusion matrix. This matrix, detailed in Figure 6, offers a comprehensive breakdown of how well our model classified sentiments. Specifically, it provides insights into the number of positive reviews correctly identified as positive, neutral reviews correctly classified as neutral, and negative reviews accurately categorized as negative. Essentially, this matrix serves as a visual representation of our model's performance in distinguishing between various sentiment categories, offering a concise summary of its classification accuracy.

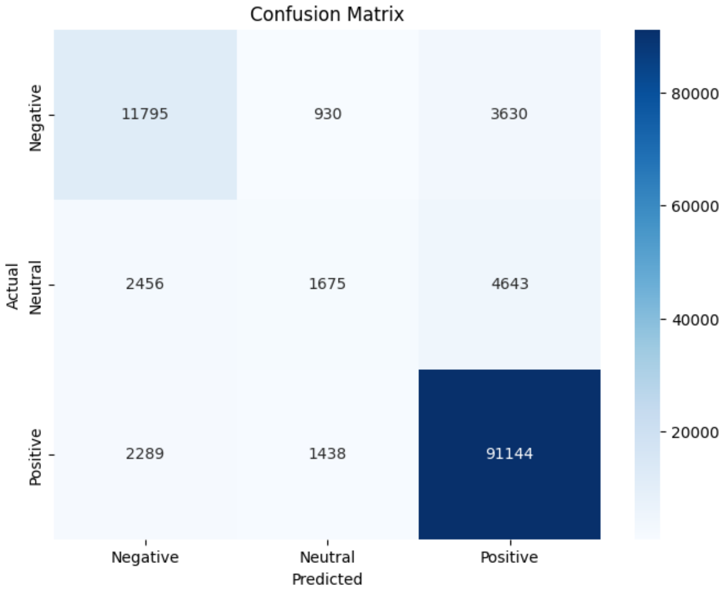


Figure 6. Bi-LSTM confusion matrix.

5.6. Transformer-Based Models

In this research, following the application of machine learning and deep learning models, we advanced to the final implementation stage, incorporating transformer-based models. Specifically, we utilized pre-trained BERT and XLNet models, which had undergone training on extensive datasets, enhancing their contextual understanding. This pre-training facilitated a performance surpassing that of both traditional machine learning and deep learning techniques. Table 9 details the configuration of both the BERT and XLNet models.

Table 9. Configuration details of transformer-based models.

	Epochs	Batch Size	Learning Rate	Loss Optimizer
BERT	5	32	2×10^{-5}	Adam W
XLNet	5	32	2×10^{-5}	Adam W

5.6.1. BERT

In our experiment, we use the BERT variant known as **BERT-base and BERT-large**. This model is pre-trained on the English language using a masked language modeling (MLM) objective, as initially introduced in [44]. Our methodology involved several key steps. Firstly, we loaded the BERT model and utilized its tokenizer to break down the reviews into manageable tokens. We then established a PyTorch DataLoader configured for batch processing, with a batch size of 32, to efficiently handle the data. To optimize our model, we employed the AdamW optimizer, specifically designed to address weight decay issues and enhance the training of neural networks. The learning rate was set at 2×10^{-5} to control the size of parameter updates during training. Subsequently, we trained the model on our training dataset for three epochs and achieved an accuracy of **89%**. The results, including a confusion matrix depicting our model's performance on the testing dataset, are illustrated in Figure 7.

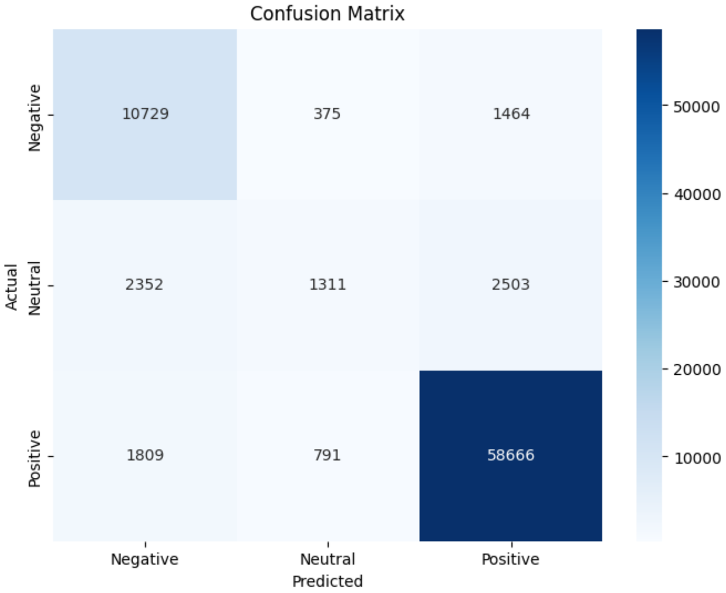


Figure 7. BERT confusion matrix.

Figures 8 and 9 show that the training and validation scores stay close together as the model learns. This means that the model is not just memorizing the training data, but is also doing well on new data. The scores for both training and validation keep getting better over time, which shows that the model is improving. This is important because it means the model can work well. Overall, these results suggest that the model is doing a good job of learning and can handle new data well.

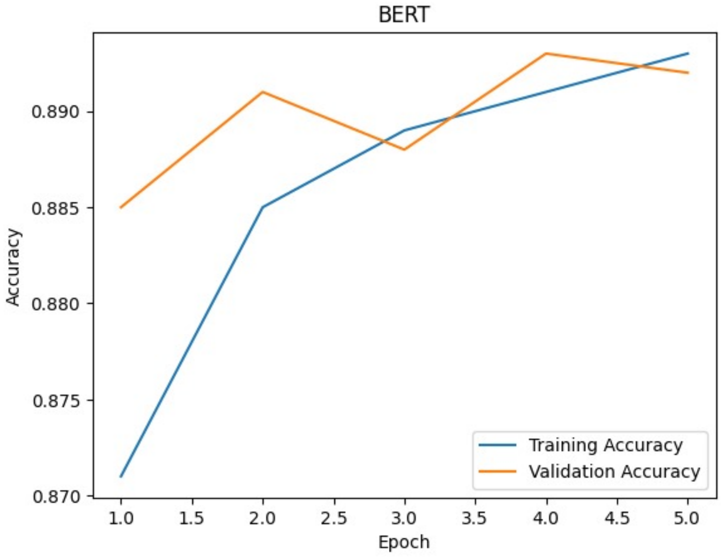


Figure 8. BERT Accuracy Curve.

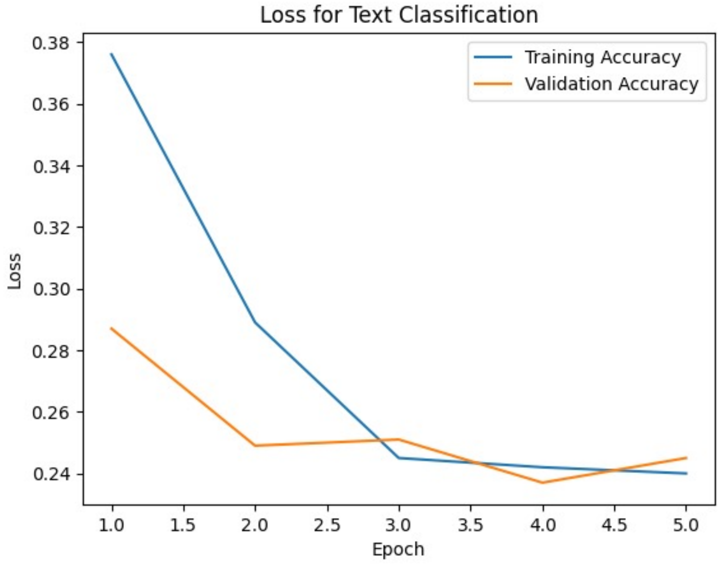


Figure 9. BERT loss curve.

5.6.2. XLNet

XLNet is a transformer model that learns bidirectional contexts by maximizing the expected likelihood across all possible permutations of the input sequence factorization order (https://huggingface.co/docs/transformers/model_doc/xlnet). First, we loaded the XLNet model, and then we used its tokenizer to break down the reviews into smaller parts called tokens. After that, we organized the data in a batch size of 16 to help the computer handle them better. To improve our program, we used the AdamW loss optimizer, which is good for training these types of programs. We set a learning rate at 2×10^{-5} to control how much the program learns in each step. Finally, we taught our program using our data for five epochs. In our test, we reached an **88%** accuracy using XLNet.

From the above description, it is evident that both BERT and XLNet exhibited similar results in terms of accuracy. However, the BERT model slightly outperforms XLNet in terms of F1 score. The detailed results of both models are provided in Table 10.

Table 10. The results of transformer-based models on 3 classifications.

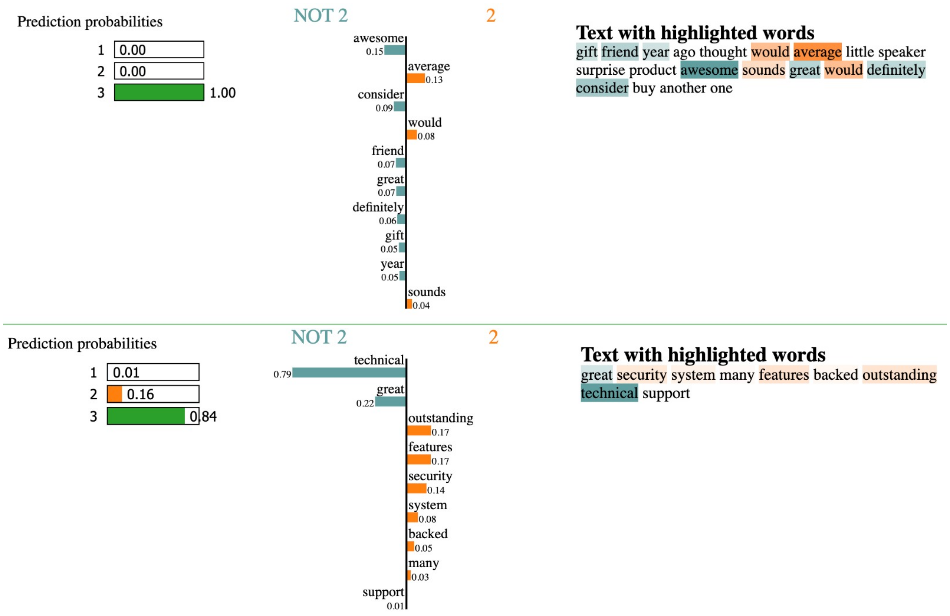
	Accuracy	F1-Score	Precision	Recall
BERT$_{base}$	**0.89**	**0.88**	0.88	0.89
XLNet$_{large}$	0.88	0.87	0.87	0.88
BERT$_{large}$	0.87	0.87	0.88	0.87

5.7. Interpretability Modeling of BERT

Local Interpretable Model-Agnostic Explanations (LIME) is a method created to understand and assess predictions made by various learning algorithms. Its purpose is to shed light on how well a model's predictions match the specific needs of a given task [45]. In this study, we investigate the reasoning behind the predictions generated by our proposed BERT model using LIME.

In Figure 10, our model detected positive sentiments. In the first sentence, it predicted positivity based on words such as "awesome", "great", "sounds", and "definitely". These words typically convey positive feelings, and the model emphasizes their importance by giving them high weights. Conversely, words like "average" and "little" usually suggest a negative tone, but the model played down their impact with low weights, minimizing their effect on the overall positive prediction.

Moving to the second sentence, the model identified positive sentiments due to the presence of words like "great", "outstanding", "many", "features", "backed", and "support". These words are commonly associated with positive feelings, and the model assigned them high weights. While the word "security" is not inherently positive or negative, it often appears with positive words like "system" and "features", so the model assigned it a positive weight as well.



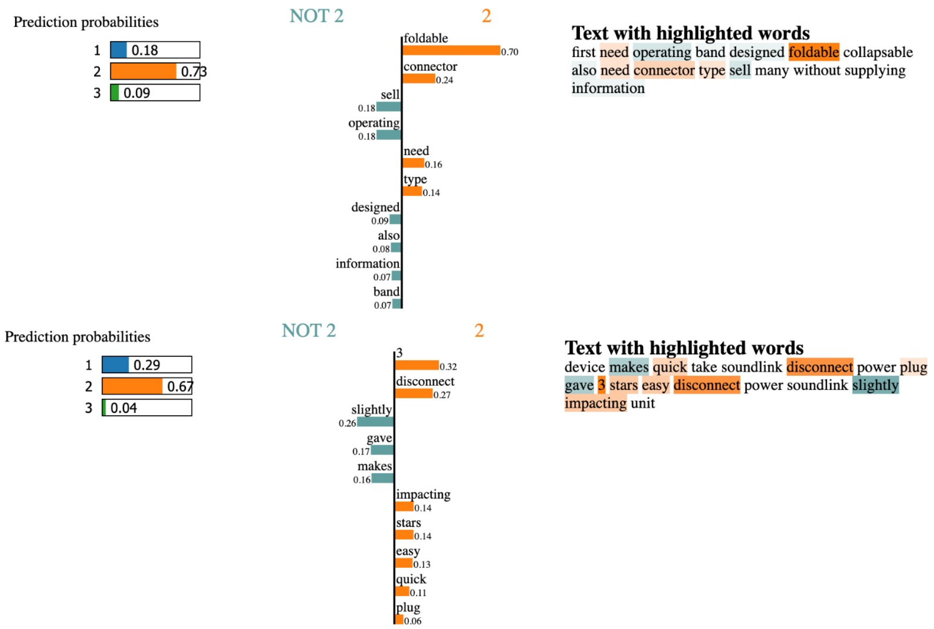
gift friend year ago thought would average little speaker surprise product awesome sounds great would definitely consider buy another one

great security system many features backed outstanding technical support

Figure 10. LIME visualization for positive reviews.

The LIME diagram in Figure 11 illustrates the neutral sentiments predicted by our model. In the first sentence, the model assigned a 70% probability to neutrality, followed by 18% for positivity and 12% for negativity. This prediction was influenced by words like "foldable", "collapsible", and "designed", which did not strongly lean towards either positive or negative sentiment. The neutral words "many" and "information" were given low weights. Although the phrase "without supplying information" slightly suggested negativity, it was not potent enough to outweigh the overall neutral tone.

In the second sentence, the model expressed a high confidence of 67% in the sentence being neutral, with a possibility of 29% for positivity and a minimal 4% for negativity. The frequent mentions of "disconnect" and "power" may have been considered negative, but the inclusion of "easy" and "quick", along with other neutral words, balanced the sentiment towards neutrality. It is worth noting that the model's sentiment analysis may also be influenced by its training data and specific nuances it has learned.



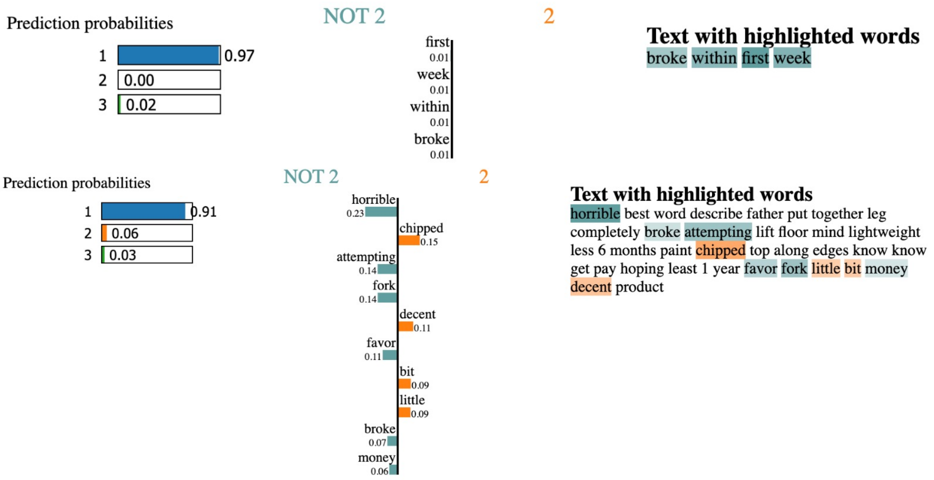
first need operating band designed foldable collapsable also need connector type sell many without supplying information

device makes quick take soundlink disconnect power plug gave 3 stars easy disconnect power soundlink slightly impacting unit

Figure 11. LIME visualization for neutral reviews.

In Figure 12, the LIME diagram illustrates the negative sentiments predicted by our model. In the first sentence, the BERT model pinpointed "slow" and "very slow" as the key factors contributing to its negative sentiment prediction. Despite positive descriptions of the display and battery life, these aspects were not enough to counterbalance the strong negativity linked to the performance issues mentioned in the sentence.

Moving to the second sentence, the model identified "slow" as the most critical factor for its negative sentiment prediction. Even though the food and portions were positively described, this positive aspect could not outweigh the pronounced negativity associated with the slow service. Therefore, the model classified it as having a negative sentiment.



broke within first week

horrible best word describe father put together leg completely broke attempting lift floor mind lightweight less 6 months paint chipped top along edges know know get pay hoping least 1 year favor fork little bit money decent product

Figure 12. LIME visualization for negative reviews.

5.8. Classification Errors

To gain a deeper understanding of the comments that the model inaccurately classified, we opted to conduct a more in-depth analysis of the predictions. Additionally, we aimed to closely examine the comments that the model predicted incorrectly, with the goal of uncovering potential issues related to the dataset and the process of annotation. Based on our analysis of all the models' performance, we have chosen to focus our investigation on the predictions made by our two top-performing models, namely, the Bi-LSTM and BERT models. These models exhibited the most favorable metrics among all the models evaluated. Table 11 displays the classification errors made by the Bi-LSTM model in its predictions. The first two rows represent instances where neutral sentiments were incorrectly classified as positive and negative, respectively. In the first case, words like "nice" and "well" were present, leading the model to classify it as positive. Similarly, in the second case, the presence of both "good" and "bad" words may have created ambiguity, making it challenging for the model to accurately discern the context as neutral.

In the next two instances, the model's misclassification of negative sentiments as neutral and positive can be attributed to its reliance on specific keywords and phrases for sentiment analysis. In record 4, where the sentiment is again negative but the model predicted it as positive, the model appears to have focused on the phrase "user-friendly". This phrase is often associated with positive feedback, and the model might have interpreted it as an indicator of positivity, even though the overall context was negative.

Table 11. Classification errors by Bi-LSTM.

Actual	Predict	Sentence
3	5	looks nice fits well apparently good material satisfy
3	1	delivered time looks great side band flimsy fell apart adjusted
1	3	ok get pay got good quality would buy
1	5	one gadget probably used years dust collecting user friendly
5	1	feel like plastic job want correct
5	3	bit smaller expected still ok

After analyzing the misclassifications made by our BERT model, let us now delve into the details of the misclassified reviews. These details are provided in Table 12. In the case of the two misclassified neutral sentiments, it is noteworthy that they contain words from both the positive and negative categories. In some sentences, the model struggled to grasp the contextual meaning accurately.

Table 12. Classification errors by BERT.

Actual	Predict	Sentence
3	5	medium quality order one little heavier
3	1	stereo outlets added speakers low quality get pay guess
1	3	cheaply made ok daughter needs
1	5	hot red pepper flakes wal mart carry bigger punch think
5	1	case cute breaks really easily got today soon put one little piece broke durable
5	3	somewhat disappointed back wood assembly seemed easy son law

Mistakes in labeling reviews as either positive or negative can lead to the misclassification of reviews. For instance, in the fifth record, the review discusses a product's durability issue, typically conveying a negative sentiment. However, it is mistakenly classified as positive, possibly due to subjective interpretation or annotation errors. This highlights the importance of precise and consistent annotation practices, as the BERT model relies on these annotations for training and can replicate the inaccuracies they inherit from the labeled data. Periodic review and refinement of annotations are crucial for enhancing model accuracy and mitigating such misclassification.

5.9. Comparison of the Proposed Approach with the State-of-the-Art

In this section, we examine the performance of our trained model across datasets used in various studies, and the details are listed in Table 13. Tan et al. [23] utilized the Consumer Reviews of the Amazon Products dataset, consisting of 34,660 data records for Amazon products like the Kindle and Fire TV Stick. They initially employed an LSTM model, achieving an accuracy rate of 71.5% on their testing dataset. However, upon implementing our proposed BERT model on their entire dataset, we realized a substantial improvement, with an impressive accuracy score of 93.7% and an F1 score of 93%, significantly surpassing the baseline. Qorich et al. [16] focused on a dataset centered around Amazon reviews, encompassing a substantial 400,00 records. Their rigorous testing leads up to a commendable peak accuracy of 90%. Upon deploying our proposed model across their testing dataset, we achieved an exceptional accuracy of 91%, further supported by an F1 score of 91.4% in the context of binary-class classification. AlQahtani et al. [46] focused on a dataset centered around Amazon mobile phone reviews, encompassing a substantial 413,840 records. Their rigorous testing leads up to an exceptional peak accuracy of 94%. Upon deploying our proposed model across this extensive dataset, we achieved a commendable accuracy of 90% in the context of multi-class classification. The reason our model did not outperform theirs is that our model was trained on a broader dataset with five different product categories, and the specific category of mobile electronics (which includes mobile accessories) constituted only 20% of our entire dataset. In contrast, the author's paper focused exclusively on mobile phone reviews. Despite this, our model achieved a commendable 90% accuracy, demonstrating its strong performance even on data it was not specifically trained on. This underscores the adaptability and effectiveness of our model across diverse and unseen data. Ahmed et al. [47] used the Amazon Fine Food dataset, which contains reviews of fine foods from Amazon spanning more than 10 years. The dataset includes approximately 500,000 reviews up to October 2012, providing information on products, users, ratings, and plain text reviews. From this extensive dataset, the users selected a subset comprising an equal number (82,007) of reviews from both positive and negative categories. The author achieved the highest accuracy of 88% using Linear SVC on the testing dataset. When applying our proposed model, we achieved an accuracy of 86%. The lower accuracy of our model can be attributed to the absence of product categories related to food reviews in our dataset. This category was entirely new to our model, leading to a slight decrease in performance. However, our results are still respectable when compared to the author's proposed model. This indicates the generalization capability of our model across different categories.

Table 13. The results the proposed model on other datasets.

Ref	Dataset Name	Baseline Accuracy	Proposed Work Accuracy
[23]	Consumer Reviews of Amazon Products	71.5%	93.7%
[16]	Amazon Reviews for Sentiment Analysis	90%	91%
[46]	Amazon Mobile Phone Reviews	94%	90%
[47]	Amazon Fine Food Reviews	88%	86%

6. Conclusions and Future Work

In the realm of eCommerce websites, sentiment analysis stands as an essential and widely adopted approach for extracting valuable insights from textual data. These sites generate a great amount of text every day, like suggestions, feedback, tweets, and comments. Additionally, customers share their opinions through reviews, ratings, and emoticons. Analyzing these reviews helps customers learn more about products and make better decisions. To achieve this goal, we applied a range of NLP, ML, EL, DL and transformer-based techniques to accurately classify sentiments. Among these techniques, the BERT model emerged as the top performer, achieving an accuracy rate of 89%. There were some limitations encountered in our study. To distinguish between similar sentiments such as ratings of 4 and 5, and 1 and 2, we suggest exploring more advanced sentiment analysis techniques that incorporate context and semantics, along with leveraging sophisticated models.

In future research endeavors, we recommend exploring several potential areas. Researchers might want to expand the analysis to include reviews written in different languages, like Roman Urdu and Arabic, and from various cultural contexts. This broader scope could offer a more complete understanding of how well the models work across different types of data. It would also be valuable to test the models on datasets from sources other than Amazon to ensure they are robust and applicable in diverse domains. Adding user metadata, such as the reviewer's purchase history and rating patterns, to the analysis could improve the accuracy of sentiment classification. Future work could also focus on aspect-based sentiment analysis, assessing sentiments related to specific aspects of a product, such as price or quality. Lastly, discussing potential real-world applications of the models and creating a feedback loop where model predictions are assessed and refined based on actual business outcomes could showcase their practical usefulness. Pursuing these recommendations has the potential to make significant contributions to the field of sentiment analysis.

Author Contributions: Conceptualization, H.A.; Methodology, H.A.; Validation, H.A., E.H. and S.S.; Formal analysis, S.Y.Y.; Writing—original draft, H.A.; Visualization, E.H. and S.Y.Y.; Supervision, E.H., S.Y.Y. and S.S.; Project administration, E.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

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

References

1. Statistics Library. Available online: https://www.oberlo.com/statistics (accessed on 9 March 2024).
2. Do, H.H.; Prasad, P.W.; Maag, A.; Alsadoon, A. Deep learning for aspect-based sentiment analysis: A comparative review. *Expert Syst. Appl.* **2019**, *118*, 272–299. [CrossRef]
3. Wang, J.; Xu, B.; Zu, Y. Deep learning for aspect-based sentiment analysis. In Proceedings of the 2021 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE), Chongqing, China, 9–12 July 2021; IEEE: New York, NY, USA, 2021; pp. 267–271.
4. Rahman, M.M.; Islam, M.N. Exploring the performance of ensemble machine learning classifiers for sentiment analysis of COVID-19 tweets. In *Sentimental Analysis and Deep Learning: Proceedings of ICSADL 2021*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 383–396.
5. Wankhade, M.; Rao, A.C.S.; Kulkarni, C. A survey on sentiment analysis methods, applications, and challenges. *Artif. Intell. Rev.* **2022**, *55*, 5731–5780. [CrossRef]
6. Xu, Q.A.; Chang, V.; Jayne, C. A systematic review of social media-based sentiment analysis: Emerging trends and challenges. *Decis. Anal. J.* **2022**, *3*, 100073. [CrossRef]
7. Medhat, W.; Hassan, A.; Korashy, H. Sentiment analysis algorithms and applications: A survey. *Ain Shams Eng. J.* **2014**, *5*, 1093–1113. [CrossRef]
8. Abbas, Y.; Malik, M. Defective products identification framework using online reviews. *Electron. Commer. Res.* **2023**, *23*, 899–920. [CrossRef]
9. Crnovrsanin, T.; Di Bartolomeo, S.; Wilson, C.; Dunne, C. Indy Survey Tool: A framework to unearth correlations in survey data. In Proceedings of the 2023 IEEE Visualization and Visual Analytics (VIS), Melbourne, Australia, 21–27 October 2023; Volume 2023.
10. Dogra, V.; Verma, S.; Chatterjee, P.; Shafi, J.; Choi, J.; Ijaz, M.F. A complete process of text classification system using state-of-the-art NLP models. *Comput. Intell. Neurosci.* **2022**, *2022*, 1883698. [CrossRef]
11. Ahmed, C.; ElKorany, A.; ElSayed, E. Prediction of customer's perception in social networks by integrating sentiment analysis and machine learning. *J. Intell. Inf. Syst.* **2023**, *60*, 829–851. [CrossRef]
12. Suhartono, D.; Purwandari, K.; Jeremy, N.H.; Philip, S.; Arisaputra, P.; Parmonangan, I.H. Deep neural networks and weighted word embeddings for sentiment analysis of drug product reviews. *Procedia Comput. Sci.* **2023**, *216*, 664–671. [CrossRef]

13. Mohammed, S.M.; Jacksi, K.; Zeebaree, S. A state-of-the-art survey on semantic similarity for document clustering using GloVe and density-based algorithms. *Indones. J. Electr. Eng. Comput. Sci.* **2021**, *22*, 552–562. [CrossRef]
14. Johnson, S.J.; Murty, M.R.; Navakanth, I. A detailed review on word embedding techniques with emphasis on word2vec. In *Multimedia Tools and Applications*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 1–29.
15. Kumar, K.S.; Desai, J.; Majumdar, J. Opinion mining and sentiment analysis on online customer review. In Proceedings of the 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Chennai, India, 15–17 December 2016; IEEE: New York, NY, USA, 2016; pp. 1–4.
16. Qorich, M.; El Ouazzani, R. Text sentiment classification of Amazon reviews using word embeddings and convolutional neural networks. *J. Supercomput.* **2023**, *79*, 11029-54. [CrossRef]
17. Hashmi, E.; Yayilgan, S.Y.; Yamin, M.M.; Ali, S.; Abomhara, M. Advancing Fake News Detection: Hybrid Deep Learning with FastText and Explainable AI. *IEEE Access* **2024**, *12*, 44462–44480. [CrossRef]
18. Xu, Y.; Wu, X.; Wang, Q. Sentiment analysis of yelp's ratings based on text reviews. In Proceedings of the 2015 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), Timisoara, Romania, 21–24 September 2015; Volume 17, pp. 117–120.
19. Rathor, A.S.; Agarwal, A.; Dimri, P. Comparative study of machine learning approaches for Amazon reviews. *Procedia Comput. Sci.* **2018**, *132*, 1552–1561. [CrossRef]
20. Elli, M.S.; Wang, Y.F. Amazon reviews, business analytics with sentiment analysis. In *Perceived Derived Attributes of Online Customer Reviews*; Elwalda, A., Lü, K., Ali, M., Eds.; Elsevier: Amsterdam, The Netherlands, 2016.
21. Cernian, A.; Sgarciu, V.; Martin, B. Sentiment analysis from product reviews using SentiWordNet as lexical resource. In Proceedings of the 2015 7th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Bucharest, Romania, 25–27 June 2015; IEEE: New York, NY, USA, 2015; p. WE-15.
22. Nasr, M.M.; Shaaban, E.M.; Hafez, A.M. Building sentiment analysis model using Graphlab. *Int. J. Sci. Eng. Res.* **2017**, *8*, 11551160.
23. Tan, W.; Wang, X.; Xu, X. Sentiment Analysis for Amazon Reviews. 2018; pp. 1–5. Available online: https://cs229.stanford.edu/proj2018/report/122.pdf (accessed on 9 March 2024).
24. Zhang, D.; Xu, H.; Su, Z.; Xu, Y. Chinese comments sentiment classification based on word2vec and SVMperf. *Expert Syst. Appl.* **2015**, *42*, 1857–1863. [CrossRef]
25. Huang, M.; Cao, Y.; Dong, C. Modeling rich contexts for sentiment classification with lstm. *arXiv* **2016**, arXiv:1605.01478.
26. Hasan, A.; Moin, S.; Karim, A.; Shamshirband, S. Machine learning-based sentiment analysis for twitter accounts. *Math. Comput. Appl.* **2018**, *23*, 11. [CrossRef]
27. Deriu, J.; Lucchi, A.; De Luca, V.; Severyn, A.; Müller, S.; Cieliebak, M.; Hofmann, T.; Jaggi, M. Leveraging large amounts of weakly supervised data for multi-language sentiment classification. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 1045–1052.
28. Jin, Z.; Lai, X.; Cao, J. Multi-label sentiment analysis base on BERT with modified TF-IDF. In Proceedings of the 2020 IEEE International Symposium on Product Compliance Engineering-Asia (ISPCE-CN), Chongqing China, 6–8 November 2020; IEEE: New York, NY, USA, 2020; pp. 1–6.
29. Ouyang, X.; Zhou, P.; Li, C.H.; Liu, L. Sentiment analysis using convolutional neural network. In Proceedings of the 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, Liverpool, UK, 26–28 October 2015, IEEE: New York, NY, USA, 2015; pp. 2359–2364.
30. Păvăloaia, V.D.; Teodor, E.M.; Fotache, D.; Danileţ, M. Opinion mining on social media data: sentiment analysis of user preferences. *Sustainability* **2019**, *11*, 4459. [CrossRef]
31. Amazon US Customer Reviews Dataset. Available online: https://www.kaggle.com/datasets/cynthiarempel/amazon-us-customer-reviews-dataset (accessed on 9 March 2024).
32. Morgenthaler, S. Exploratory data analysis. *Wiley Interdiscip. Rev. Comput. Stat.* **2009**, *1*, 33–44. [CrossRef]
33. Alasadi, S.A.; Bhaya, W.S. Review of data preprocessing techniques in data mining. *J. Eng. Appl. Sci.* **2017**, *12*, 4102–4107.
34. Bahrawi, B. Sentiment analysis using random forest algorithm-online social media based. *J. Inf. Technol. Its Util.* **2019**, *2*, 29–33. [CrossRef]
35. Fikri, M.; Sarno, R. A comparative study of sentiment analysis using SVM and Senti Word Net. *Indones. J. Electr. Eng. Comput. Sci.* **2019**, *13*, 902–909.
36. Tripathy, A.; Agrawal, A.; Rath, S.K. Classification of sentiment reviews using n-gram machine learning approach. *Expert Syst. Appl.* **2016**, *57*, 117–126. [CrossRef]
37. El-Halees, A.M. Arabic text classification using maximum entropy. *IUG J. Nat. Stud.* **2015**, *15*, 157–167.
38. Yu, F.; Moh, M.; Moh, T.S. Towards extracting drug-effect relation from Twitter: a supervised learning approach. In Proceedings of the 2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), New York, NY, USA, 9–10 April 2016; IEEE: New York, NY, USA, 2016; pp. 339–344.
39. Sneka, G.; Vidhya, C. Algorithms for Opinion Mining and Sentiment Analysis: An Overview. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **2016**, *6*, 1–5.

40. Kashyap, H.; Buksh, B. Combining Naïve Bayes and modified maximum entropy classifiers for text classification. *IJ Inf. Technol. Comput. Sci.* **2016**, *9*, 32–38. [CrossRef]
41. Kaufmann, M. JMaxAlign: A maximum entropy parallel sentence alignment tool. In Proceedings of the COLING 2012: Demonstration Papers, Mumbai, India, 8–15 December 2012; pp. 277–288.
42. Deshmukh, J.S.; Tripathy, A.K. Entropy based classifier for cross-domain opinion mining. *Appl. Comput. Inform.* **2018**, *14*, 55–64. [CrossRef]
43. Nigam, K.; Lafferty, J.; McCallum, A. Using maximum entropy for text classification. In Proceedings of the IJCAI-99 Workshop on Machine Learning for Information Filtering, Stockholom, Sweden, 31 July–6 August 1999; Volume 1, pp. 61–67.
44. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
45. Hashmi, E.; Yayilgan, S.Y. Multi-class hate speech detection in the Norwegian language using FAST-RNN and multilingual fine-tuned transformers. *Complex Intell. Syst.* **2024**, 1–22. [CrossRef]
46. AlQahtani, A.S. Product sentiment analysis for amazon reviews. *Int. J. Comput. Sci. Inf. Technol. (IJCSIT)* **2021**, *13*. [CrossRef]
47. Ahmed, H.M.; Javed Awan, M.; Khan, N.S.; Yasin, A.; Faisal Shehzad, H.M. Sentiment analysis of online food reviews using big data analytics. *Elem. Educ. Online* **2021**, *20*, 827–836.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.