International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 16 (2024) pp. 547–565

Article

Enhancing Sentiment Analysis in Electronic Product Reviews Using Machine Learning Algorithms

Mohandas Archana [1,*] and Thambusamy Velmurugan [2,*]

1 PG Department of IT & BCA, Dwaraka Doss Goverdhan, Doss Vaishnav College, Chennai 600106, India
2 PG & Research Department of Computer Science Dwaraka, Doss Goverdhan Doss Vaishnav College, Chennai 600106, India
* Correspondence author: archanadgvc@gmail.com (M.A.); velmurugan_dgvc@yahoo.co.in (T.V.)

Received date: 1 March 2024; Accepted date: 16 April 2024; Published online: 10 July 2024

Abstract: In the era of electronic commerce, understanding customer sentiments through product reviews has become crucial for the smooth running of businesses. In this research work, thorough investigation of sentiment analysis in the electronic product reviews dataset, which is collected from the Flipkart and some other social media. The aim of this research work is to ascertain the polarity of consumer comments by doing sentiment analysis on text based electronic product reviews. A number of preprocessing methods were applied to the chosen dataset including stemming, tokenization, lemmatization, punctuation removal, and stop word removal. These actions were essential for improving the textual data and getting it ready for the further processing. The text-based data was transformed into a numeric format using vectorization techniques, and the resulting data was then fed into machine learning algorithms to identify sentiments. After that, the dataset was divided into training and testing portions in order to ensure a robust model evaluation. Using machine learning algorithms like Naive Bayes, Support Vector Machine (SVM), Random Forest, and Decision Tree, this work analyzes and classifies sentiment. The objective is to identify significant sentiments within textual reviews and examine the effectiveness and efficiency of the aforementioned machine learning techniques using preprocessed data. The best classification algorithm's performance is indicated by the outcomes it produces, out of all of them.

Keywords: naïve bayes algorithm; support vector machine; decision tree method; random forest algorithm; sentiment analysis

1. Introduction

Customer reviews are now a crucial component of product evaluation in the quickly changing world of e-commerce, as consumers increasingly rely on online platforms to make decisions about what to buy. Businesses now confront the difficult task of gleaning actionable insights from massive volumes of unstructured data, given the exponential growth of digital transactions. Sentiment analysis is a key instrument that provides a methodical way to comprehend and make use of the sentiments that are expressed in customer reviews. By looking at customer sentiment analysis, businesses can learn vital information about the benefits and drawbacks of their products. Positive sentiments emphasize features that appeal to customers, while negative sentiments highlight areas that require innovation and improvement.

Figure 1 describes a standard sentiment analysis pipeline, which begins with model training on annotated data and moves on to using the trained model to predict sentiment in fresh, unseen reviews. Understanding customer opinions and feedback is facilitated by the sentiment scores acquired, which offer insights into the sentiment conveyed in the text. Positive, negative, and neutral reviews were indicated by annotations on the training data's sentiments. The reviews went through a text cleansing procedure that probably included removing stop words, converting text to lowercase, and removing punctuation in order to improve the quality of the data.

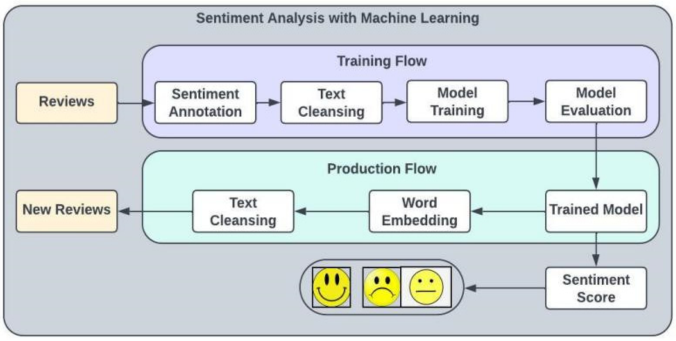

Figure 1. Sentiment Analysis with Machine Learning.

Word embedding techniques were utilized to capture the semantic relationships between the words in the reviews and represent them numerically. The preprocessed and embedded text data were used to train a sentiment analysis model. The model found patterns and connections between words in order to predict sentiment labels with accuracy. It is likely that the trained model's performance in predicting sentiments on the training dataset was evaluated using metrics such as accuracy, precision, recall, and F1 score. New, unseen reviews were used as the input for sentiment analysis. The new reviews underwent the same text cleansing process as the training reviews in order to guarantee consistency. Using word embedding techniques, the preprocessed text of the new reviews was converted into a format suitable for input into the trained model. The freshly trained model was applied to the new reviews using the patterns it had previously learned to predict the sentiment labels for the unseen data. The sentiment score, which indicates the anticipated sentiment polarity (positive, negative, or neutral) for each new review, was developed based on the output of the trained model.

The architecture for this research is depicted in Figure 2. The UCI repository provided the electronic product dataset, which served as the basis for sentiment analysis. To improve the dataset's quality, a number of preprocessing methods were used, such as stemming, tokenization, lemmatization, removing punctuation, and removing stop words from the text. In order to enable machine learning model compatibility, vectorization techniques were utilized to transform the processed text-based dataset into a numeric format. Two sections of the dataset were separated out: 80% for training and 20% for testing, in order to guarantee a comprehensive evaluation of the machine learning models. Several popular classification algorithms, such as Random Forest and Naive Bayes, were applied to the processed dataset. These algorithms were chosen to analyze the electronic product reviews' sentiment polarity. The effectiveness of each classification algorithm was assessed using metrics such as recall, accuracy, precision, and F1 score.

The remaining section of the research article is structured as, the reviews of the literature listed in Section 2. Section 3 examines the procedures and resources. Section 4 provides a description of the research work’s experimental outcomes. The research findings are concluded in Section 5.

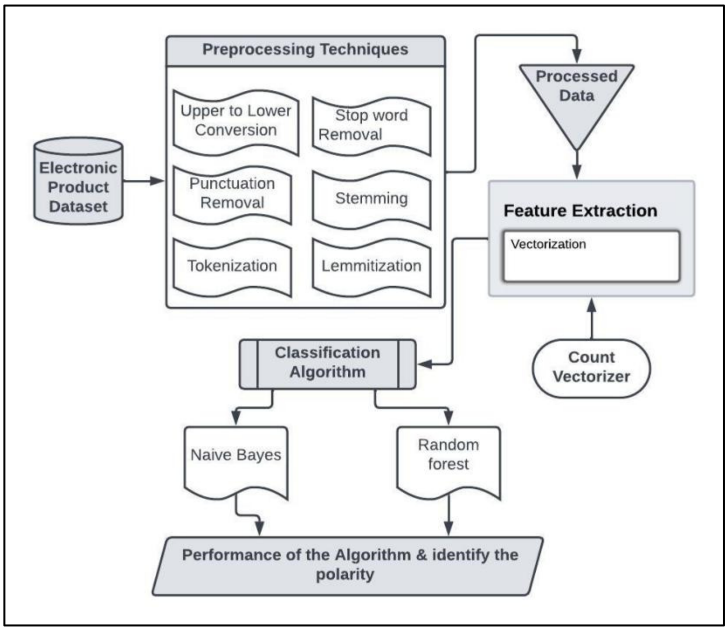

Figure 2. Architecture of the Proposed Research Work.

2. Review of Literature

A research work carried out by Sara Ashour Aljuhani and Norah Saleh Alghamdi in [1], In which it is stated how successful a number of machine learning algorithms are, such as logistic regression, convolutional neural networks, naive bayes, and stochastic gradient descent. Their experimental findings demonstrated that convolutional neural networks using word2vec as a feature extraction method yield the best results for both the balanced and imbalanced versions of the dataset. The research work titled as “Sentiment classification of online consumer reviews using word vector representations”, done by Bansal, Barkha, and Sangeet Srivastava in [2]. In this study, the researchers use the word2vec model to convert reviews into vector representations for classification. The dataset consists of over 400,000 customer reviews from Amazon's mobile phone category. Next, utilizing 10-folds cross-validation and CBOW (continuous bag of words) and skip-gram models, the researchers classify the customer reviews using a variety of machine learning algorithms, including SVM, Nave Bayes, Logistic Regression, and Random Forest. The results of the experiment show that Random Forest with CBOW achieves the best accuracy.

An additional study conducted by Lee et al. in [3] presents a method for developing a model to analyze user sentiment using word embedding space generated by learning review data of Amazon fashion products. Three SVM classifiers based on the quantity of positive and negative review data were learned for the experiments, and the word embedding space was generated by learning 5.7 million Amazon review data. The maximum accuracy of 88.0% was achieved when an SVM classifier was learned using 50,000 reviews, comprising 50,000 positive and 50,000 negative reviews, as per the experimental results.In a different study, “Sentiment analysis of product reviews using supervised learning”, conducted by Shah and Arkesha in [4], it is stated that machine learning techniques such as Naïve Bayes, Support Vector Machine, and FastText word embedded with CNN deep learning model were used to analyze the reviews of mobile phones. Out of all of these techniques, the CNN deep learning method integrated with FastText words yields superior results than the machine learning techniques.

Salmony et al.’s research paper from [5] used the Amazon product review dataset in conjunction with conventional machine learning techniques like NB, SVM, and RF. Ultimately, the researchers evaluate and compare the most effective approach. Of these techniques, Random Forest yields the most accurate results. In the study “Amazon Product Reviews: Sentiment Analysis Using Supervised Learning Algorithms”, conducted by Hawlader et al. in [6], the classification algorithms Naïve Bayes, Support Vector Machine, Random Forest, Decision Tree, Logistic Regression, and Multi-Layer Perceptron classifier were used to analyze the Amazon electronics product reviews. MLP produces the highest yield, 92%.

In this research, Table 1 describes the comparison of various research papers. The review of literature helps the researchers to understand the key concept, assessing methodologies, limitation and findings. This evaluation helps in refining the research work for implementing proposed methods and some existing methods in this research work.

Table 1. Comparison of various research work.

Reference No.	Title of the Paper	Author Name	Methods Used	Best Method
[1]	A comparison of sentiment analysis methods on Amazon reviews of Mobile Phones	Aljuhani, Sara Ashour, and Norah Saleh Alghamdi	convolutional neural networks, stochastic gradient descent, naive bayes, and logistic regression	convolutional neural networks provides the best result
[2]	Sentiment classification of online consumer reviews using word vector representations	Bansal, Barkha, and Sangeet Srivastava	SVM, Nave Bayes, Logistic Regression, and Random Forest	Random Forest scores highest F1 Score
[3]	User sentiment analysis on Amazon fashion product review using word embedding	Lee, Dong-yub, Jae-Choon Jo, and Heui-Seok Lim	Support Vector Machine	Support Vector Machine provides 88.0% of accuracy
[4]	Sentiment analysis of product reviews using supervised learning	Shah, Arkesha	Naïve Bayes, Support Vector Machine and also FastText word embedded with CNN deep learning model	FastText word embedded with CNN deep learning method achieves the best result
[5]	Supervised Sentiment Analysis on Amazon Product Reviews: A survey	Salmony, Monir Yahya Ali, and Arman Rasool Faridi	NB, SVM and RF	RF gives the best performance
[6]	Amazon product reviews: Sentiment analysis using supervised learning algorithms	Hawlader, Mohibullah, Arjan Ghosh, Zaoyad Khan Raad, Wali Ahad Chowdhury, Md Sazzad Hossain Shehan, and Faisal Bin Ashraf	Naïve Bayes, Support Vector Machine, Random Forest, Decision Tree, Logistic Regression and Multi-Layer perceptron classifier	MLP yield the best result of 92%.

3. Materials and Methods

Text pre-processing, or cleaning and modifying unstructured text material to prepare it for analysis, is a necessary step in natural language processing (NLP) research projects. The customer review data contains both processed and raw data [7,8]. Various machine learning algorithms are used to classify the provided text, taking into consideration both inputs. It is feasible to differentiate between favourable, unfavourable, and impartial customer feedback by scrutinizing the reviews posted on the Flipkart website.

3.1 Description of dataset

Flipkart_com_ecommerce_Aircooler_review is the name of the dataset relation, and it includes 1,192 occurrences with a total weight of 249. The dataset is divided into parts and categorised using 10 cross-validations based on detailed accuracy with class.

In the dataset, each attribute has two or more different values. The Table 2 below displays the same dataset with regard to Product, Review Month, and Review Text.

Table 2. Sample Dataset.

Rate	Review	Summary
5	super!	GREAT cooler excellent air flow!!!! and for this price its so amazing and unbelievablejust love it
5	awesome	best budget 2 fit cooler nice cooling$
3	fair	the quality is good but the power of air is decent
1	uselessproduct	very bad product itsa only a fan
3	fair	ok ok product*****

3.2. Preprocessing Methods

The unprocessed dataset of customer reviews for electronic products underwent tokenization, stop-word removal, lemmatization, upper-to-lower conversion, and feature extraction techniques [9][10]. The training and testing processes use different portions of the dataset. A total of 80% of the data was used for training, and 20% was used for testing. Among the pre-processing phases are as follows.

3.2.1. Stemming and Lemmatization

These methods are frequently employed in text preparation [11][12]. Lemmatization is slower than stemming because it understands the context of words before processing them, but stemming is faster since it cuts words without knowing the context.

stem(word) = Transform (word, Rule_1), Transform (word,Rule_2)…Transform(word, Rule_n)

Rule_1, Rule_2, ..., Rule_n are separate stemming rules or transformations that are applied to the input word in a sequential manner. The function called Transform (word, Rule). It modifies the input word according to a specified rule. The word's reduced or stemmed form is called stem (word).

3.2.2. Removal of Punctuation

The text is cleared of all punctuation in this stage. Python's string library has a pre-defined collection of punctuation, including '!"#$%&'()*+,-./:;?@[]_'|'.

RemovePunctuation(input_text)=replace(input_text,",.!?...","")

A straightforward mathematical abstraction of the punctuation removal procedure is represented by this formula [13–15]. The precise syntax in real programming or implementation will vary depending on the text processing tool or programming language you are using.

3.2.3. Lower the Text

One of the most common text preparation techniques for Python is to convert the text to the same case—ideally lower case. However, since lower casing can sometimes cause information loss, you do not have to finish this step every time you work on an NLP problem [16]. For example, sentences written in capital letters can convey excitement or dissatisfaction when addressing someone's feelings in any endeavour.

output_text = ToLowercase(input_text)

This demonstrates how to change the input text's case using the ToLowercase function. The text's uppercase letters would all need to be changed to their corresponding lowercase letters in order to put this into practice [17,18].

3.2.4. Tokenization

The text is divided into smaller units in this step. Depending on our issue statement, we can utilise either sentence tokenization or word tokenization[19,20].

tokens = Tokenize(input_text)

A function called tokenize takes in text input and outputs a series of tokens.

3.2.5. Stop Word Removal

Often used terms that are removed from the text because they don't contribute to the analysis are known as stopwords [21]. These words mean very little or nothing at all. The NLTK library contains a list of terms that are considered stopwords in the English language.

Remove Stop Words (input_text, stop_words) = words where words $\in$ Tokenize (input_text) and words $\in$ stop_words

This communicates the idea that the output text is made up of all the words from the tokenized input text minus the words in the stop word set [22,23]. Among them are the following: he, most, other, some, such, no, nor, not, only, own, same, so, then, too, very, s, t, can, will, just, don, don't, should, should've, now. I, me, my, myself, we, our, ours, ourselves, you, you're, you've, you'll, you'd, your, yours, yourself, yourselves.

3.2.6. Vectorization

The term "vectorization" refers to a traditional technique for transforming input data from its original text-based format into real-number vectors, which is the format that is supported by ML models[24]. This method has been around since the invention of computers, it has proven extremely successful across numerous disciplines, and it is now utilised in NLP. Vectorization is a phase in the feature extraction process in machine learning [25]. By translating text to numerical vectors, the goal is to extract some distinguishing features from the text for the model to train on.

For count vectorization, the term-document matrix X can be expressed as

$$X_{ij=}\ Count\ (t_j, d_i), \tag{1}$$

This formulation generates a matrix X in which the values represent the counts of terms in the corresponding documents, the rows represent a document, and the columns represent a unique term in the vocabulary.

3.3. Machine Learning Algorithm

The textual review is classified using a Supervised Learning Technique called the Classification algorithms [26,27]. Training datasets are used to teach software how to classify different classes or groups. To find out which of these algorithms performs the best overall using the processed and unprocessed data, this study uses the classification methods Naive Bayes, Support Vector Machine, Random Forest, and Decision Tree.

3.3.1. Naïve Bayes

Naive Bayes is the name of a probabilistic algorithm used to solve classification problems. It makes predictions based on the probability that a given input belongs to a specific class [28].

The Naive Bayes mathematical expression is as follows:

$$\mathrm{P(C|X_1, X_2, \dots \dots \dots X_n)} = \frac{\mathrm{P(C). P(X_1|C). P(X_2|C) \dots . P(X_n|C)}}{\mathrm{P(X_1).\ P(X_2) \dots \dots . P(X_n)}} \tag{2}$$

The Bayes theorem serves as the foundation for the probabilistic machine learning algorithm Naive Bayes. The features will be denoted by 1, 2, X_1, X_2, X_n, and the class variable will be called C. The objective is to predict the likelihood of a particular class given the observed features.

where P(class|features) is the probability that the input is a member of a specific class based on the features. P(features|class) is the probability of observing the given features in a sample that is part of the class; P(features) is the probability of observing the given features in the dataset [29]. P(features|class) is the conditional probability of the features given the class. The prior probability of the class, or the likelihood that the class will appear in the dataset, is denoted by P(class).

3.3.2. Random Forest

For each decision tree in the method, a subset of features and a subset of data points are randomly chosen from the training set [30,31]. Then, using the chosen features and data points, it generates each decision tree independently using a similar splitting criterion to the decision tree algorithm.

$$RF(x) = \frac{1}{n}\ \sum\nolimits_{i=1}^{n} RF_i\ (x) \tag{3}$$

The mean prediction is frequently the final result of regression [32,33]. The mode function, denoted by mode mode, yields the most prevalent class in the classification scenario. The Random Forest has a total of n trees.

4. Results and Discussion

In order to predict sentiments from text-based reviews on the test set and extract the predicted sentiment labels, the trained models are applied to the dataset that was gathered from the Flipkart shopping website. The pertinent metrics to evaluate the models, including recall, accuracy, precision, F1-score, etc.

4.1. Results of Preprocessing

The text is now in a standardized format that can be analysed after the raw text data was preprocessed using the methods previously mentioned. To maintain consistency, all text has been converted to lowercase [34,35]. To cut down on noise, stopwords—common words that don't add much to the meaning—have been eliminated. To draw attention to the text's alphabetic content, punctuation has been removed. By eliminating suffixes, stemming has been used to reduce words to their most basic form and simplify the vocabulary [36,37]. The text has been tokenized, or broken up into individual words. Words have been further reduced to their dictionary or base form through lemmatization, which takes context into account for greater accuracy [38–40].

In order to prepare the text data for a range of natural language processing (NLP) applications and machine learning models, Figure 3 illustrates how the preprocessing steps aid in representing the text data in a more ordered and clean manner.

Summary	Summary_lower	Summary_wo_punct	Summary_wo_stop	Summary_token	Summary_stemmed	Summary_lemmitized
GREAT cooler excellent air flow!!!! and for this price its so amazing and unbelievablejust love it	great cooler excellent air flow!!!! and for this price its so amazing and unbelievablejust love it	great cooler excellent air flow and for this price its so amazing and unbelievablejust love it	great cooler excellent air flow price amazing unbelievablejust love	['great', 'cooler', 'excellent', 'air', 'flow', 'price', 'amazing', 'unbelievablejust', 'love']	['great', 'cooler', 'excel', 'air', 'flow', 'price', 'amaz', 'unbelievablejust', 'love']	['great', 'cooler', 'excel', 'air', 'flow', 'price', 'amaz', 'unbelievablejust', 'love']
best budget 2 fit cooler nice cooling$	best budget 2 fit cooler nice cooling$	best budget 2 fit cooler nice cooling	best budget 2 fit cooler nice cooling	['best', 'budget', '2', 'fit', 'cooler', 'nice', 'cooling']	['best', 'budget', '2', 'fit', 'cooler', 'nice', 'cool']	['best', 'budget', '2', 'fit', 'cooler', 'nice', 'cool']
the quality is good but the power of air is decent	the quality is good but the power of air is decent	the quality is good but the power of air is decent	quality good power air decent	['quality', 'good', 'power', 'air', 'decent']	['qualiti', 'good', 'power', 'air', 'decent']	['qualiti', 'good', 'power', 'air', 'decent']
very bad product its a only a fan	very bad product its a only a fan	very bad product its a only a fan	bad product fan	['bad', 'product', 'fan']	['bad', 'product', 'fan']	['bad', 'product', 'fan']
ok ok product*****	ok ok product*****	ok ok product	ok ok product	['ok', 'ok', 'product']	['ok', 'ok', 'product']	['ok', 'ok', 'product']

Figure 3. Results of preprocessing.

4.2. Result of Vectorization

It appears that the table is a numerical representation of customer reviews based on how frequently the terms “coolar”, “cooler” , “coolercons1” , and “cooleri” appear in each review. Each row in Table 3 represents a unique customer review, and each column displays the word's frequency in that review.

This column shows how many times the word "coolar" appears in each review. A zero count is nonexistent. cooler: This column displays the frequency with which the word “cooler” occurs in each review. Each review has a number between 1 and 3. coolercons1: This column displays the frequency with which the term “coolercons1” occurs in each review. A zero count is nonexistent. cooleri: This column displays the frequency with which the word "cooleri" occurs in each review. There are variations in the counts; a count is present in one review.

Table 3. Converted Text Data into numerical format.

Number of Times the Words Repeated in the Customer Reviews				
Text	**Coolar**	**Cooler**	**Coolercons1**	**Cooleri**
This cooler is good	0	1	0	0
Cooler performs well	0	1	0	0
Is cooler working good?	0	1	0	0
Good Product	0	0	0	0
This AC is best	0	0	0	0
Good Cooler	0	1	0	0
Cooler is bad	0	1	0	0
Is the Cooler Cooleris good?	0	2	0	0
Cooler is super. Cooler is working good	0	2	0	0
This is best cooler. Cooleri s best.	0	1	0	1

For instance, the word “cooler” appears twice in the eighth review, but the word “cooleri” appears once. The table gives a clear numerical picture of how frequently these terms appear in each customer review.

Figures 4 and 5 shows the distinct instance of the data is represented by each row, and a unique word (coolar, cooler, coolercons1, cooler) is represented by each column. The frequency or presence of each word in the corresponding instance is indicated by the numerical values in the cells. The code looks to be vectorizing using a technique similar to bag-of-words. This method treats every distinct word as a separate feature, and each word’s presence or absence is denoted by a numerical value (0 or 1, in this example). This is a commonly used technique in natural language processing to convert textual data into a format that is compatible with machine learning algorithms[41][42]. The generated numeric vectors can be fed into a variety of machine learning models.

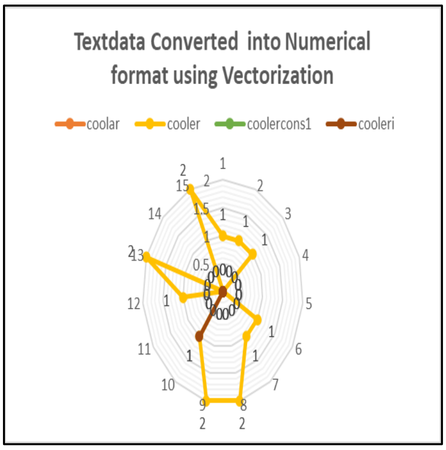

Figure 4. Results of Vectorization.

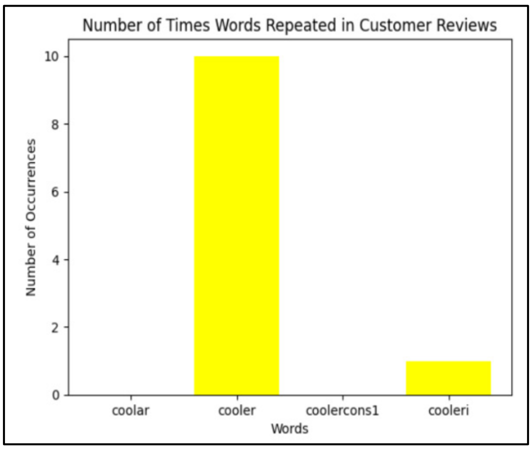

Figure 5. Number of times words repeated.

4.3. Results of Naïve Bayes Algorithm

When assessing the efficacy and performance of machine learning algorithms, such as the Naïve Bayes algorithm used in sentiment analysis, experimental results are essential [43,44]. The Table 4 displays the actual sentiments of different texts as well as the sentiments that the Naïve Bayes algorithm predicted. These experimental results are significant in a number of ways.

The usefulness of the trained model is confirmed by the outcomes of the experiments. Here, the Naïve Bayes model’s accuracy in predicting sentiments serves as validation [45,46]. A well-performing model has high accuracy and good agreement between the sentiments predicted and observed. The text attribute in the table shows that the statement for which a sentiment prediction using the Naïve Bayes algorithm is made is represented by this column. Actual_Sentiment attribute indicates that the true sentiment label of the text as supplied by the dataset. Three classes of emotions exist: positive, negative, and neutral. Predicted_Sentiment attribute displays the sentiment labels predicted by the Naïve Bayes algorithm for every corresponding text.

Table 4. Sentiment Prediction using Naïve Bayes Algorithm.

Text	Actual_ Sentiment	Predicted_ Sentiment
This cooler is good	Neutral	positive
Cooler performs well	Positive	positive
Is cooler working good?	Positive	positive
Good Product	Positive	positive
This AC is best	Positive	positive
Good Cooler	Positive	positive
Excellent	Positive	positive
Very bad product	Negative	negative
Cooler is bad	Negative	negative
Is the Cooler Cooler is good?	Positive	positive
Cooler is super. Cooler is working good	Positive	positive
This is best cooler. Cooleri s best.	Positive	positive

Table 5 shows that the Naïve Bayes algorithm for sentiment analysis, the "Actual Count" table presents an analysis of the actual sentiments within a dataset. For every sentiment class, the number of instances is displayed. The number of times the Naïve Bayes algorithm classified the text's true sentiment as positive is represented in this category. There are 181 instances in this dataset where the algorithm accurately predicted a positive sentiment. The number of times the Naïve Bayes algorithm determined that the text's true sentiment was negative is represented in this category. There are sixteen cases in this dataset where the algorithm accurately predicted a negative sentiment. The number of cases in which the Naïve Bayes algorithm determined the text's true sentiment to be neutral is represented in this category. 42 times in this dataset show that the algorithm accurately predicted a neutral sentiment.

Table 5. Actual Count using Naïve Bayes Algorithm.

Actual Count	
Positive	181
Negative	16
Neutral	42

Based on actual Naïve Bayes algorithm classifications, the “Actual Count” table presents a quantitative overview of the sentiment distribution in the dataset. This data is important for evaluating the dataset's sentiment balance and comprehending how well the algorithm predicts each sentiment category shows in Figures 6 and 7.

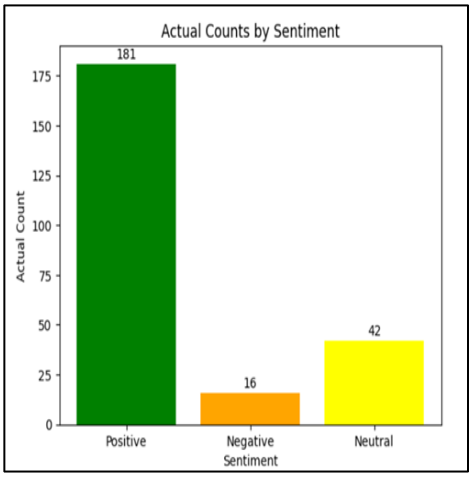

Figure 6. Actual Count using Naïve Bayes.

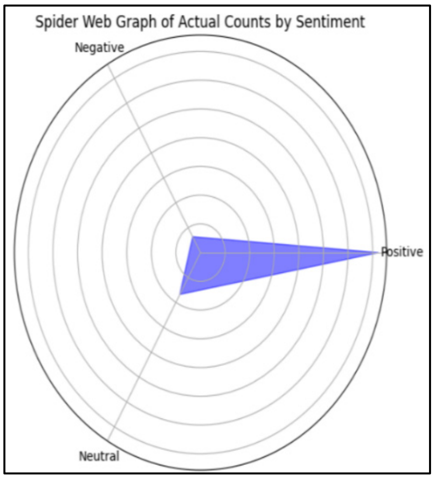

Figure 7. Actual Count by Sentiment.

It is especially helpful for assessing the algorithm's performance and pinpointing possible areas for development, like resolving dataset imbalances or enhancing the model to handle particular sentiment classes more skilfully.

The "Predicted Count" in Table 6 represents a detailed analysis of the sentiments that the Naïve Bayes algorithm predicted for the specified dataset. The number of occurrences for each anticipated sentiment class is displayed.

Table 6. Predicted Count using Naïve Bayes Algorithm.

Predicted Count	
Positive	225
Negative	4
Neutral	10

The number of occurrences in which the Naïve Bayes algorithm predicted a positive sentiment is represented by the positive category shows in Figures 8 and 9.

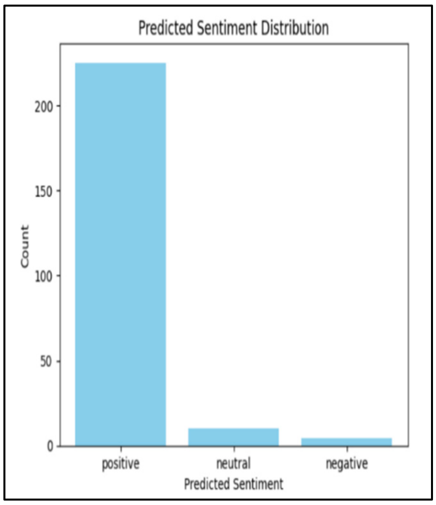

Figure 8. Predicted Count.

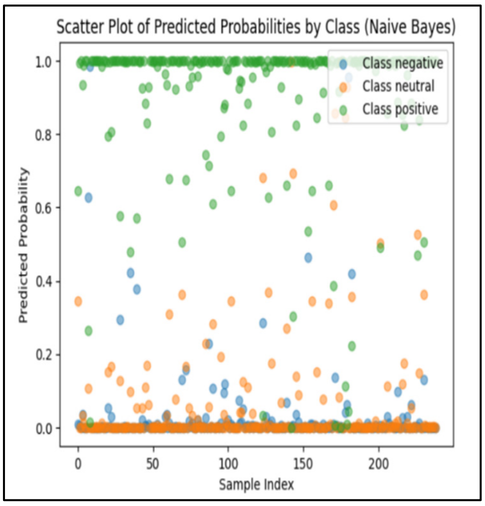

Figure 9. Predicted Probabilities.

There are 225 cases in this dataset where the algorithm correctly predicted a favourable sentiment shows in Table 7.

Table 7. Comparison of Predicted and Actual Count.

Predicted Count		**Actual Count**	
Positive	225	Positive	181
Negative	4	Negative	16
Neutral	10	Neutral	42

The number of times the Naïve Bayes algorithm predicted a negative sentiment is represented by the negative category. Four times in this dataset did the algorithm predict a negative sentiment shows in Figures 10 and 11.

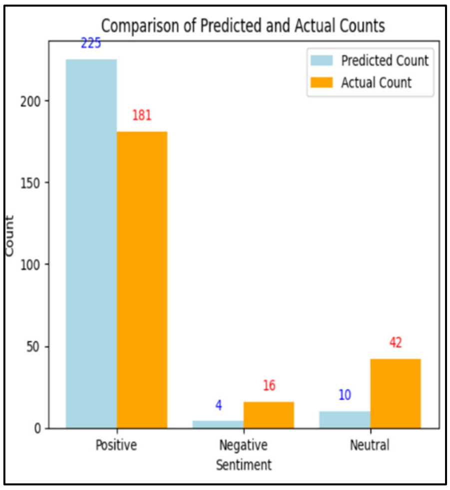

Figure 10. Sentiment Counts.

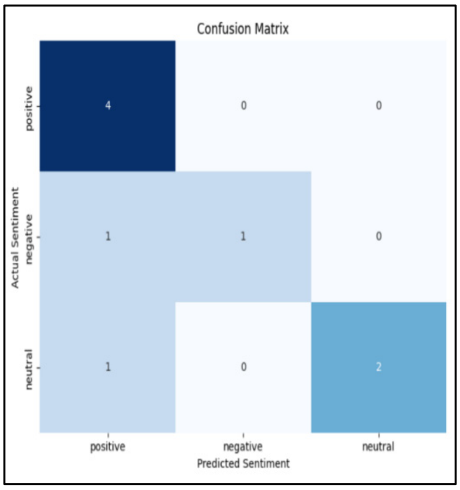

Figure 11. Confusion matrix.

Table 8 shows with great precision and recall, the model did a good job at predicting the Negative and Neutral classes. The precision for the Positive class is good, meaning that the model is frequently right when it predicts a positive outcome. Nonetheless, there may be chances to enhance memory in order to record more true good examples. The particular objectives and task specifications determine the overall performance. Modifications can be made to optimise the model for recall, precision, or a combination of the two depending on the situation.

Table 8. Performance Metrics of Naïve Bayes.

Naïve Bayes	**Precision**	**Recall**	**F1-Score**	**Support**
Negative	1	0.25	0.4	16
Neutral	0.8	0.19	0.31	42
Positive	0.8	0.99	0.88	181
Macro	0.87	0.48	0.53	239
Weighted	0.81	0.8	0.75	239

Figure 12 represents the different performance metrics for a Naïve Bayes algorithm used in sentiment analysis. The number of correctly predicted instances of a class among all instances predicted to belong to that class is known as precision. The number of correctly predicted instances of a class out of all actual instances of that class is called recall, which is also referred to as sensitivity or true positive rate.

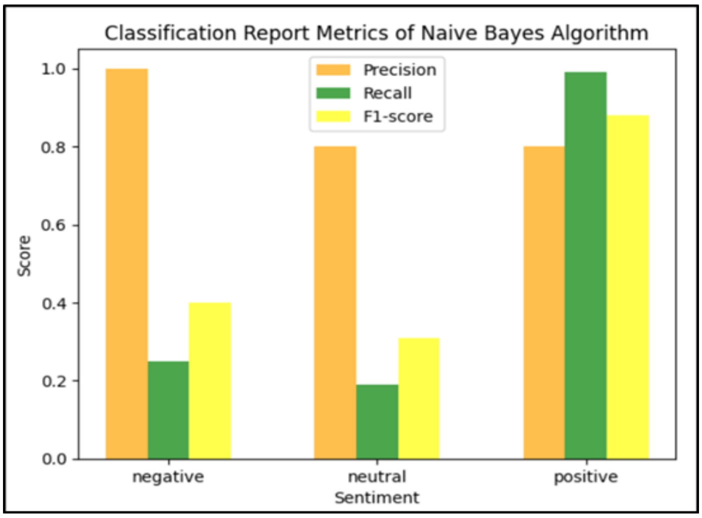

Figure 12. Performance Metrics of Naïve Bayes Algorithm.

The F1-score offers a fair assessment of a model's performance since it is the harmonic mean of precision and recall. The number of real instances of each class in the dataset is represented by the term “support”. Without taking into account class imbalances, macro-averaging determines the average performance across all classes. By assigning greater weight to classes with more instances, weighted averaging takes into account class imbalances.

4.4. Results of Random Forest Algorithm

The Random Forest algorithm’s sentiment prediction results are shown in Table 9. The true sentiment labels of the texts in the dataset are represented by the Actual_Sentiment column.

Table 9. Sentiment Prediction using Random Forest Algorithm.

Actual_Sentiment	Predicted_Sentiment
neutral	neutral
positive	negative
positive	positive
positive	positive
positive	neutral
positive	positive
positive	positive
negative	negative
negative	negative
positive	positive
positive	positive
positive	positive
positive	positive
positive	positive
positive	positive

The Random Forest algorithm predicts the sentiment labels for each corresponding text, which are shown in the Predicted_Sentiment column. With 11 out of 14 cases where the algorithm accurately predicted the sentiment, the accuracy rate was about 78.6%.

Misclassifications occur when a genuine "positive" sentiment is mistakenly classified as "negative." Two examples that truly expressed the sentiment "positive" were mistakenly labeled as "neutral", given that the majority of the instances fall into the "positive" sentiment class, the class distribution in the dataset seems to be skewed in that direction. The Random Forest algorithm appears to be consistent in predicting positive sentiments, as evidenced by the majority of correct predictions for instances where the actual sentiment was “positive”.

Based on the Random Forest algorithm’s actual classifications, the “Sentiment Actual Count” in Table 10 offers a quantitative summary of the dataset's sentiment distribution. This data is important for evaluating the dataset’s sentiment balance and comprehending how well the algorithm predicts each sentiment category. It is especially helpful for assessing the algorithm's performance and pinpointing possible areas for development, like resolving dataset imbalances or enhancing the model to handle particular sentiment classes more skillfully.

Table 10. Sentiment Actual Count using Random Forest Algorithm.

Sentiment	Actual_Count
positive	181
neutral	42
negative	16

Figures 13 and 14 shows the "Sentiment Actual Count using Random Forest (RF) Algorithm" presents a sentiment analysis using the Random Forest algorithm, allowing for a detailed analysis of the actual sentiments within the dataset. For every sentiment class, the number of instances is displayed. The dataset's sentiment classes—positive, neutral, and negative—are represented by the sentiment column. Based on actual classifications by the Random Forest algorithm, the Actual_Count column shows the number of instances for each sentiment class.

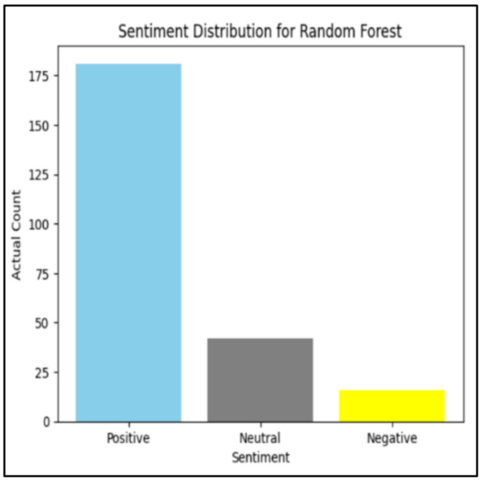

Figure 13. Sentiment Actual Count.

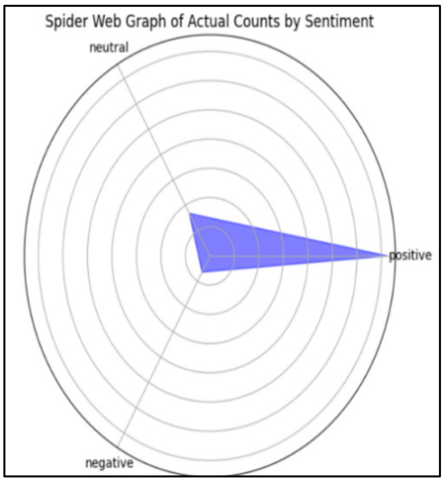

Figure 14. Actual Counts of Sentiment.

The sentiment predictions generated by a model on a particular dataset or collection of text samples are broken out in this Table 11. It is helpful in comprehending the anticipated sentiment distribution and evaluating how well the model captures positive, neutral, and negative attitudes in the data. To assess the accuracy, precision, recall, and other performance metrics of the model, additional analysis may entail contrasting these predictions with the actual feelings, contingent upon the particular task and environment for which the model is intended.

Table 11. Sentiment Predicted Count using Random Forest Algorithm.

Sentiment	**Predicted_Count**
positive	182
neutral	46
negative	11

The anticipated counts for each sentiment group are clearly compared in the bar chart. The same data is shown in a different way in a scatter plot chart, where each point represents a sentiment category and its associated expected count shown in Figures 15 and 16.

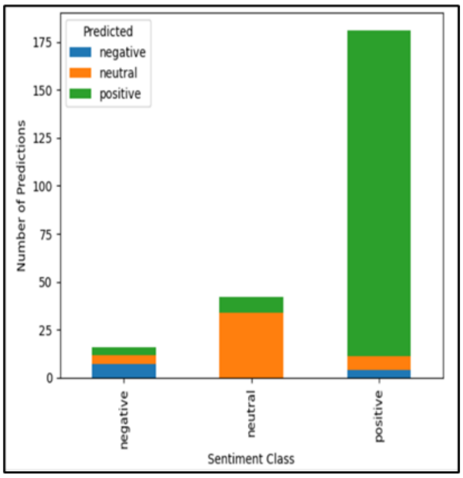

Figure 15. Sentiment Predicted Count.

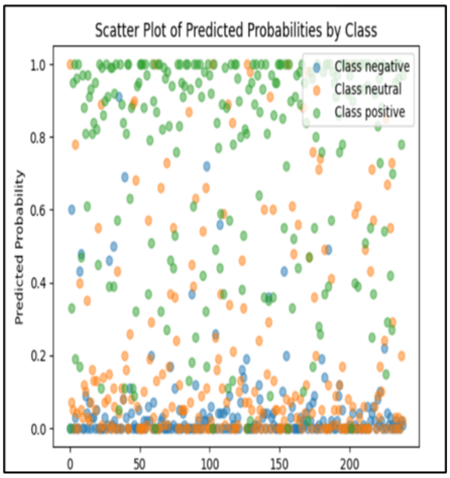

Figure 16. Predicted Probabilities by class.

This Table 12 is helpful in determining how well the model matches the sentiment distribution in reality. Variations between actual and expected counts can reveal areas in which the model is working well or where it may need to be improved. A more thorough assessment of the model's performance may be obtained through additional analysis, such as the computation of performance measures (such as precision, recall, and accuracy).

Table 12. Comparison of Actual and Predicted Count.

Sentiment	Actual Count	Predicted Count
positive	181	182
neutral	42	46
negative	16	11

When it comes to sentiment analysis, a waterfall chart is a useful visual aid that shows how actual and expected sentiments change over time. Knowing how well a sentiment analysis model performs across several sentiment categories can be helpful shown in Figures 17 and 18.

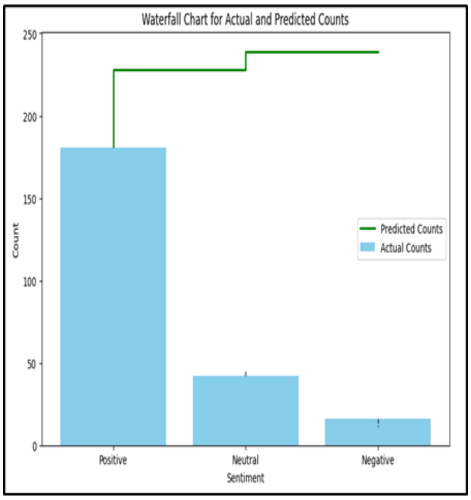

Figure 17. Actual & Predicted Counts.

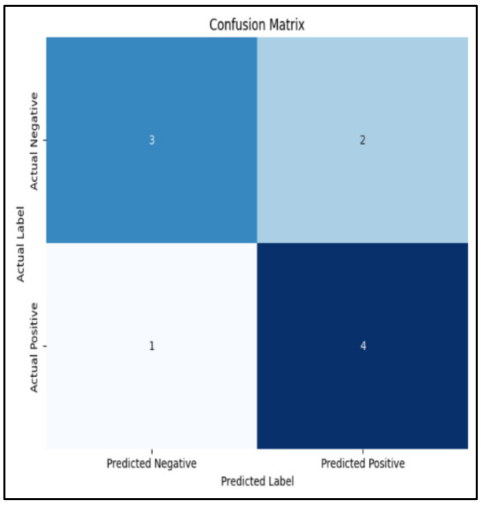

Figure 18. Confusion matrix of counts.

The Table 13 elaborates to assess a classification model's performance is called a confusion matrix. It offers a synopsis of the actual and anticipated classifications, enabling a thorough examination of the model's effectiveness across several classes. The matrix can be expanded to multi-class settings; however, it is especially helpful in binary classification issues.

Table 13. Performance Metrics using Random Forest Algorithm.

Naïve Bayes	Precision	Recall	F1-Score	Support
Negative	1	0.25	0.4	16
Neutral	0.8	0.19	0.31	42
Positive	0.8	0.99	0.88	181
Macro	0.87	0.48	0.53	239
Weighted	0.81	0.8	0.75	239

The model performs well in predicting positive attitudes, as evidenced by its excellent recall, precision, and F1-Score for the Positive class. With somewhat balanced recall and precision, the Neutral class performs well as well. The lower precision and recall of the Negative class indicate some difficulties in correctly predicting negative attitudes.

When taking class imbalances into account, macro and weighted averages offer an overview of the overall performance of the model shown in Figure 19. When taken as a whole, these metrics provide a thorough assessment of the model's ability to predict attitudes in various classifications.

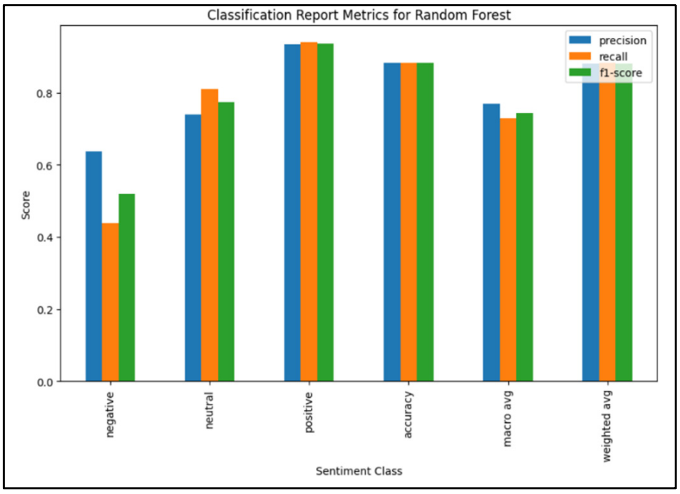

Figure 19. Performance Metrics using Random Forest Algorithm.

5. Conclusions

Based on the performance of the four classification methods—Naive Bayes, Support Vector Machine (SVM), Random Forest (RF), and Decision Tree (DT)—were used after analyzing the electronic product dataset using both processed and unprocessed data. With the highest accuracy of 87% for the raw data and 89% for the processed data, Random Forest performed better than the other techniques in both scenarios. This suggests that for this specific dataset, RF is the optimal algorithm to apply. Furthermore, compared to the raw data, all algorithms performed more accurately after the data was processed. The accuracy of both SVM and Naive Bayes models was 86% for both processed and raw data. However, when utilizing processed data as opposed to unprocessed data, DT and RF demonstrated increased accuracy. Therefore, it can be said that the accuracy of the algorithms used is positively impacted by the data processing. The best algorithm for this dataset is RF, which can also be used for future analysis and forecasting.

Author Contributions

M.A. designing and writing the source code and some parts of the paper prepared. The results shown in figures and tables are obtained by running the source code. T.V. formulating the entire paper via its structure and suggested the research contribution which is obtained by means of results. Also, T.V. corrected the grammatical mistakes and written some parts of the paper. All authors have read and agreed to the published version of the manuscript.

Funding

There is no research funding for this research work.

Conflict of Interest Statement

There is no conflict of Interest to publish this paper.

Data Availability Statement

The data used in this research work are obtained and taken from the Kaggle open dataset.

References

1. Aljuhani, Sara Ashour, and Norah Saleh Alghamdi, "A comparison of sentiment analysis methods on Amazon reviews of Mobile Phones", *International Journal of Advanced Computer Science and Applications* 10, no. 6, 2019.
2. Bansal, Barkha, and Sangeet Srivastava, "Sentiment classification of online consumer reviews using word vector representations," *Procedia computer science,* no. 132, pp.1147-1153,2018.
3. Lee, Dong-yub, Jae-Choon Jo, and Heui-Seok Lim, "User sentiment analysis on Amazon fashion product review using word embedding", *Journal of the Korea Convergence Society* 8, no. 4, pp.1-8, 2017.
4. Shah, Arkesha, "Sentiment analysis of product reviews using supervised learning", *Reliability: Theory & Applications* 16, no. SI 1 (60), pp. 243-253, 2021.
5. Salmony, Monir Yahya Ali, and Arman Rasool Faridi, "Supervised Sentiment Analysis on Amazon Product Reviews: A survey", IEEE, *2nd International Conference on Intelligent Engineering and Management (ICIEM)*, pp. 132-138, 2021.
6. Hawlader, Mohibullah, Arjan Ghosh, Zaoyad Khan Raad, Wali Ahad Chowdhury, Md Sazzad Hossain Shehan, and Faisal Bin Ashraf ,"Amazon product reviews: Sentiment analysis using supervised learning

algorithms", IEEE, *International Conference on Electronics, Communications and Information Technology (ICECIT)*, pp. 1-6, 2021.
7. Haseeb, Abdul, Rabia Taseen, Marryam Sani, and Qamas Gul Khan, "Sentiment Analysis on Amazon Product Reviews using Text Analysis and Natural Language Processing Methods", *International Conference on Engineering, Natural and Social Sciences*, vol. 1, pp. 446-452, 2023.
8. Martis, Eesha, Rutuja Deo, Sejal Rastogi, Keshav Chhaparia, and Ameyaa Biwalkar, "A Proposed System for Understanding the Consumer Opinion of a Product Using Sentiment Analysis", *Sentiment Analysis and Deep Learning: Proceedings of ICSADL 2022*, pp. 555-568, 2023.
9. Mutinda, James, Waweru Mwangi, and George Okeyo, "Sentiment analysis of text reviews using lexicon-enhanced bert embedding (LeBERT) model with convolutional neural network", *Applied Sciences* 13, no. 3, pp.1445, 2023.
10. Solairaj, A., G. Sugitha, and G. Kavitha. "Enhanced Elman spike neural network-based sentiment analysis of online product recommendation", *Applied Soft Computing 132*, pp. 109789, 2023.
11. Diekson, Ziedhan Alifio, Muhammad Rivyan Bagas Prakoso, Muhammad Savio Qalby Putra, Muhammad Shaden Al Fadel Syaputra, Said Achmad, and Rhio Sutoyo, "Sentiment analysis for customer review: Case study of Traveloka", *Procedia Computer Science* 216, pp. 682-690, 2023.
12. Li, Xianyong, Jiabo Zhang, Yajun Du, Jian Zhu, Yongquan Fan, and Xiaoliang Chen, "A novel deep learning-based sentiment analysis method enhanced with Emojis in microblog social networks", *Enterprise Information Systems* 17, no. 5, pp. 2037160, 2023.
13. Jena, Om Prakash, Alok Ranjan Tripathy, Manas Prasad Rout, and Partha Sarathi Pattnayak. "Sentiment Analysis on Movie Review Data Using Ensemble Machine Learning Approaches." In *Innovations in Intelligent Computing and Communication: First International Conference, ICIICC 2022, Bhubaneswar, Odisha, India, December 16-17, 2022, Proceedings*, pp. 389-400, 2023.
14. Başarslan, Muhammet Sinan, and Fatih Kayaalp, "Sentiment analysis with ensemble and machine learning methods in multi-domain datasets", *Turkish Journal of Engineering* 7, no. 2 pp. 141-148, 2023.
15. Mujawar, Sofiya S., and Pawan R. Bhaladhare. "An Aspect based Multi-label Sentiment Analysis using Improved BERT System." *International Journal of Intelligent Systems and Applications in Engineering* 11, no. 1, pp. 228-235, 2023.
16. Dey, Sanjay, Sarhan Wasif, Dhiman Sikder Tonmoy, Subrina Sultana, Jayjeet Sarkar, and Monisha Dey, "A comparative study of support vector machine and Naive Bayes classifier for sentiment analysis on Amazon product reviews", IEEE,*International Conference on Contemporary Computing and Applications (IC3A)*, pp. 217-220, 2020.
17. Mubarok, Mohamad Syahrul, Adiwijaya, and Muhammad Dwi Aldhi, "Aspect-based sentiment analysis to review products using Naïve Bayes," In *AIP Conference Proceedings*, AIP Publishing LLC, vol. 1867, Issue no. 1, pp. 020060, 2017.
18. Singla, Zeenia, Sukhchandan Randhawa, and Sushma Jain, "Sentiment analysis of customer product reviews using machine learning", IEEE, *international conference on intelligent computing and control (I2C2)*, pp. 1-5, 2017.
19. Brownfield, Steven, and Junxiu Zhou, "Sentiment analysis of Amazon product reviews", *Proceedings of the Computational Methods in Systems and Software*, Springer, Cham, pp. 739-750, 2020.
20. Bhatt, Aashutosh, Ankit Patel, Harsh Chheda, and Kiran Gawande, "Amazon review classification and sentiment analysis", *International Journal of Computer Science and Information Technologies* 6, Issue no. 6, pp. 5107-5110, 2015.
21. Rathor, Abhilasha Singh, Amit Agarwal, and Preeti Dimri, "Comparative study of machine learning approaches for Amazon review", *Procedia computer science 132*, pp.1552-1561, 2018.
22. Abraham, Minu P., and Udaya Kumar Reddy KR, "Feature based sentiment analysis of mobile product reviews using machine learning techniques", *International Journal* 9, pp. 2, 2020.
23. Shaheen, Momina, Shahid M. Awan, Nisar Hussain, and Zaheer A. Gondal, "Sentiment analysis on mobile phone reviews using supervised learning techniques", *International Journal of Modern Education & Computer Science* 11, pp. 7, 2019.
24. Salem, Mohammed AM, and Ashraf YA Maghari, "Sentiment Analysis of Mobile Phone Products Reviews Using Classification Algorithms", IEEE, *International Conference on Promising Electronic Technologies (ICPET)*, pp. 84-88, 2020.
25. Yiran, Ye, and Sangeet Srivastava, "Aspect-based Sentiment Analysis on mobile phone reviews with LDA", In *Proceedings of the 2019 4th International Conference on Machine Learning Technologies*, pp. 101-105, 2019.
26. Kumari, Upma, Arvind K. Sharma, and Dinesh Soni, "Sentiment analysis of smart phone product review using SVM classification technique",IEEE, *International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, pp. 1469-1474, 2017.
27. Dubey, Tulika, and Aaditya Jain, "Sentiment analysis of keenly intellective smart phone product review utilizing SVM classification technique",IEEE, *10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1-8, 2019.
28. Wahyudi, Mochamad, and Dinar Ajeng Kristiyanti, "Sentiment Analysis of Smartphone Product Review Using Support Vector Machine Algorithm-Based Particle Swarm Optimization", *Journal of Theoretical & Applied Information Technology* 91, pp. 1 ,2016.
29. D'souza, Stephina Rodney, and Kavita Sonawane, "Sentiment analysis based on multiple reviews by using machine learning approaches", IEEE, *International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 188-193, 2019.

30. Wahyudi, Mochamad, and Dinar Ajeng Kristiyanti, "Sentiment Analysis of Smartphone Product Review Using Support Vector Machine Algorithm-Based Particle Swarm Optimization", *Journal of Theoretical & Applied Information Technology* 91, pp. 1 ,2016.
31. Pujari, Chetana, and Nisha P. Shetty, "Comparison of classification techniques for feature-oriented sentiment analysis of product review data", *Data Engineering and Intelligent Computing*, Springer, pp. 149-158, 2018.
32. Fang, Xing, and Justin Zhan, "Sentiment analysis using product review data", *Journal of Big Data 2*, Vol. 1, pp. 1-14, 2015.
33. Al Amrani, Yassine, Mohamed Lazaar, and Kamal Eddine El Kadiri, "Random forest and support vector machine based hybrid approach to sentiment analysis", *Procedia Computer Science*, Vol. 127, pp. 511-520, 2018.
34. Başarslana, Muhammet Sinan, and Fatih Kayaalpb, "Sentiment Analysis with Machine Learning Methods on Social Media", Vol. 5, 2015.
35. Wang, Hanshi, Lizhen Liu, Wei Song, and Jingli Lu, "Feature-based Sentiment Analysis Approach for Product Reviews", *Journal of Softw. 9*, Vol. 2, pp. 274-279, 2014.
36. Latha.U, and T. Vemurugan, " Analyzing agricultural text data using classification algorithms.", *International conference on small and medium business,* pp. 339-344, 2016.
37. Alsolamy, Afnan Atiah, Muazzam Ahmed Siddiqui, and Imtiaz Hussain Khan, "A Corpus Based Approach to Build Arabic Sentiment Lexicon", *International Journal of Information Engineering & Electronic Business*, Vol. 11, No. 6, 2019.
38. Kumar, Akshi, and Teeja Mary Sebastian, "Sentiment analysis on twitter", *International Journal of Computer Science Issues (IJCSI)*, Vol. 9, No. 4, pp. 372, 2012.
39. Darwich, Mohammad, Shahrul Azman Mohd, Nazlia Omar, and Nurul Aida Osman, "Corpus-Based Techniques for Sentiment Lexicon Generation: A Review", *Journal of Digital Information Management*, Vol.17, No. 5, pp. 296, 2019.
40. Chathuranga, P. D. T., S. A. S. Lorensuhewa, and M. A. L. Kalyani, "Sinhala sentiment analysis using corpus-based sentiment lexicon", IEEE, *19th international conference on advances in ICT for emerging regions (ICTer)*, Vol. 250, pp. 1-7, 2019.
41. Abdulla, Nawaf A., Nizar A. Ahmed, Mohammed A. Shehab, and Mahmoud Al-Ayyoub, "Arabic sentiment analysis: Lexicon-based and corpus-based", IEEE, *Jordan conference on applied electrical engineering and computing technologies (AEECT)*, pp. 1-6, 2013.
42. Cordeiro, Cheryl Marie, "A corpus-based approach to understanding market access in fisheries and aquaculture: a systematic literature review", *International Journal of Economics and Management Engineering,* Vol. 13, no. 10, pp. 1324-1333, 2019.
43. Singh, Jaspreet, Gurvinder Singh, and Rajinder Singh, "Optimization of sentiment analysis using machine learning classifiers", *Human-centric Computing and information Sciences,* Vol. 7, no.1, pp.1-12, 2017.
44. Neethu, M. S., and R. Rajasree, "Sentiment analysis in twitter using machine learning techniques*",* IEEE, *Fourth international conference on computing, communications and networking technologies (ICCCNT)*, pp. 1-5, 2013.
45. Le, Bac, and Huy Nguyen, "Twitter sentiment analysis using machine learning techniques", *In Advanced Computational Methods for Knowledge Engineering,* Springer, pp. 279-289, 2015.
46. Hemalatha, I., GP Saradhi Varma, and A. Govardhan, "Sentiment analysis tool using machine learning algorithms", *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS),* Vol. *2*, no. 2, pp. 105-109, 2013.