PLOS ONE

RESEARCH ARTICLE

Enhancing machine learning-based sentiment analysis through feature extraction techniques

Noura A. Semary[1], Wesam Ahmed[1,2], Khalid Amin[1], Paweł Pławiak[3,4]*, Mohamed Hammad[1,5]*

1 Department of Information Technology, Faculty of Computers and Information, Menoufia University, Shibin El Kom, Egypt, **2** Department of Information Technology, Faculty of Computers and Artificial Intelligence, South Valley University, Hurghada, Egypt, **3** Department of Computer Science, Faculty of Computer Science and Telecommunications, Cracow University of Technology, Krakow, Poland, **4** Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Gliwice, Poland, **5** EIAS Data Science Lab, College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia

* plawiak.pawel@gmail.com (PP); mohammed.adel@ci.menofia.edu.eg (MH)

Citation: A. Semary N, Ahmed W, Amin K, Pławiak P, Hammad M (2024) Enhancing machine learning-based sentiment analysis through feature extraction techniques. PLoS ONE 19(2): e0294968. https://doi.org/10.1371/journal.pone.0294968

Editor: Nadeem Sarwar, Bahria University - Lahore Campus, PAKISTAN

Received: September 22, 2023

Accepted: November 12, 2023

Published: February 14, 2024

Data Availability Statement: All Data are available at: https://appen.com/pre-labeled-datasets/ and https://www.kaggle.com/datasets/eswarchandt/amazon-music-reviews.

Funding: The author(s) received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

Abstract

A crucial part of sentiment classification is featuring extraction because it involves extracting valuable information from text data, which affects the model's performance. The goal of this paper is to help in selecting a suitable feature extraction method to enhance the performance of sentiment analysis tasks. In order to provide directions for future machine learning and feature extraction research, it is important to analyze and summarize feature extraction techniques methodically from a machine learning standpoint. There are several methods under consideration, including Bag-of-words (BOW), Word2Vector, N-gram, Term Frequency- Inverse Document Frequency (TF-IDF), Hashing Vectorizer (HV), and Global vector for word representation (GloVe). To prove the ability of each feature extractor, we applied it to the Twitter US airlines and Amazon musical instrument reviews datasets. Finally, we trained a random forest classifier using 70% of the training data and 30% of the testing data, enabling us to evaluate and compare the performance using different metrics. Based on our results, we find that the TD-IDF technique demonstrates superior performance, with an accuracy of 99% in the Amazon reviews dataset and 96% in the Twitter US airlines dataset. This study underscores the paramount significance of feature extraction in sentiment analysis, endowing pragmatic insights to elevate model performance and steer future research pursuits.

1. Introduction

Public opinion plays a significant role in business operations and product perception. Additionally, since it explains human behaviour and how other people's opinions affect it, public opinion analysis is very helpful to governments. The application of sentiment analysis holds significant value in discerning the sentiment and perspective expressed in textual material

[1–3]. The problem can be framed as either a binary or multi-class classification task. Binary sentiment analysis separates texts into positive and negative classes, while multiclass sentiment analysis separates them into fine-grained categories [4, 5]. Sentiment analysis can be done on social media platforms like Twitter and websites, including comments, forums, blogs, and microblogs. The analysis of sentiment is usually performed by applying a rule-based system or a machine learning system. In recent years, machine learning systems have become increasingly popular because they are more versatile and easier to apply than traditional rule-based systems. Machine learning algorithms are trained to recognize underlying patterns in documents in order to classify them [6, 7]. Sentiment analysis based on machine learning involves three steps: feature extraction, feature selection, and machine learning classifier. The selection of feature extraction for better outcomes in many natural language processing (NLP) tasks, especially in sentiment analysis, is vital [8, 9]. Feature extraction is the methodological procedure of identifying and converting pertinent material from its original form into a more succinct and significant representation, with the purpose of facilitating analysis [10, 11].

One of the major challenges in sentiment classification tasks is the choice of feature extraction technique. In the analysis process, features are represented as a single unit and used to classify documents into the corresponding polarities [12, 13]. As a result of the large number of features, the overall system will be impacted by a heavy processing load, and the use of irrelevant features produces overfitting or underfitting models of classifiers. The system's performance is optimal when the feature set is considerably small but informative and accurate. Text embeddings or feature extraction techniques map text data into vectors, which can be a set of real numbers (a vector) that can be used as input to a machine learning model. There are numerous word representation models now in use. Based on the word distribution data, the models can be categorized as either traditional models or static models, according to [14, 15]. Depending on the specific feature extraction technique used, different types of information can be extracted from the text data. This study offers the following research points:

1. The use of different techniques for feature extraction is investigated for sentiment analysis tasks and provides a useful resource for assessing the strengths and limitations of different feature extraction approaches and making informed choices.
2. We explore the exact relationship between feature extraction, classification performance, and the training time of the methods.
3. Providing a discussion on how the accuracy of the machine learning algorithms changes with different feature extractions. It is still an unsolved problem and an unanswered question on how to select a suitable feature extraction technique to be used to obtain the best performance for capturing sentiment in different social media datasets.
4. In this study, we uncover which feature extraction technique is most effective for sentiment analysis tasks as well as the implications of our findings for practical applications such as monitoring social media sites, among other areas.

As for the rest of the study, it is arranged as follows: The background literature section presents relevant sentiment analysis work. The proposed system section explains the methodology and details of the study. The experimental results section displays the results. In discussion section, the outcomes are discussed. The conclusion section includes a summary of the paper's findings and future work.

2. Background literature

This section reviews relevant studies on feature extraction for sentiment analysis using machine learning models.

2.1 Previous studies

For sentiment analysis tasks, most of the feature extraction algorithms have been used with different machine learning models, but few studies have looked at their impact on their performance. A comparative study of sentiment analysis is shown in Table 1. Comparing and contrasting prior research will enable the present study to explore a different area that has not been discussed previously. The n-gram and term frequency-inverse document frequency (TF-IDF) are widely recognized as prominent feature extraction strategies in machine learning models, as seen by their prevalence in numerous prior studies.

Ahmed and Ahmed [16] applied TF-IDF, random forest (RF), Naïve Bayes (NB), and feature extraction to the collected fake news articles to classify them into positive and negative sentiments. Among the individual classifiers, the NB was the best and achieved the highest accuracy (89.30%).

Gaur *et al.* [17] used a machine learning algorithm based on the NB Classifier with TF-IDF feature extraction to classify the Twitter sentiment 140 dataset. Based on performance metrics including precision, recall, and accuracy, the suggested model's results showed improved accuracy (84.44%) and precision.

Qi and Shabrina [18] extracted data relating to COVID-19 from Twitter users in England's major cities. This study compares machine learning models as its main objective, such as multinominal Naïve Bayes (MNB), RF, and support vector classification (SVC), with lexicon-based approaches such as Vader and Textblob using two feature extraction methods (Word2-Vec embedding and TF-IDF). Overall, the SVC with TF-IDF had better accuracy than the other models.

Al sari *et al.* [19] created three different datasets from social media platforms to analyze the impressions about Saudi cruises. The methodology of the study is performed by applying

Table 1. Literature survey of sentiment analysis.

Ref.	Dataset	Feature Extraction	Model	Results
Ahmed and Ahmed [16]	Collected news articles	TF-IDF	NB	Accuracy = 89.30%
Gaur et al. [17]	Twitter sentiment 140	TF-IDF	NB	Accuracy = 84.44%
Qi and Shabrina [18]	Collected tweets about COVID-19	TF-IDF, and Word2Vec	MNB, SVC, RF, Vader, and Textblob	SVC with TF–IDF outperforms others with accuracy = 71%
Al sari et al. [19]	Instagram, Snapchat, and Twitter datasets	Unigrams	MLP, NB, RF, SVM, and voting	NB algorithm in Twitter with Over-sampling technique achieves accuracy = 85.26%
Mukherjee et al. [20]	Amazon reviews	TF-IDF	MNB, SVM, and ANN	ANN + Negation classifier performs the best with accuracy = 96.32%
Noori [21]	Customer reviews	TF-IDF	NB, SVM, DT, and KNN	Best accuracy reported for DT = 98.9%
Zahoor and Rohilla [22]	Collected tweets about different events	N-gram	NB, SVM, RF, and LSTM	NB outperforms others on most datasets with accuracy = 96.8%
Samuel et al. [23]	COVID-19 tweets	N-gram	NB and LR	NB outperforms LR with accuracy = 91.43%
Kumar et al. [24]	Book reviews	BOW, and Word2Vec	NB, ME, and SVM	SVM has the highest accuracy = 78%
Zarisfi et al. [25]	Twitter datasets	TF-IDF	SVM, MNB and hybrid algorithm	The hybrid method yields a better classification with accuracy = 85.92%

https://doi.org/10.1371/journal.pone.0294968.t001

machine learning algorithms such as multilayer perceptron (MLP), NB, voting, SVM, RF, and the n-grams feature extraction technique. The RF algorithm achieved 100% classification accuracy with oversampled Snapchat data.

Mukherjee *et al.* [20] presented a customized algorithm for detecting explicit negation. Different machine learning algorithms, such as NB, SVM, and Artificial Neural Networks (ANN), were performed on Amazon reviews to analyze the sentiments. The methodology of TF-IDF was employed to extract features. The ANN with a negative classifier achieved the best accuracy (96.32%).

Noori [21] proposed a new approach to classify customer sentiments. The paper collected customer reviews from an international hotel. The reviews are processed, and then the TF-IDF extractor is applied to build the document vectors and then trained into SVM, ANN, NB, k-nearest neighbor (K-NN), decision tree (DT), and C4.5 models. The result of the DT model is an accuracy of 98.9% with the number of features (1800), and this model performed better than others.

Zahoor and Rohilla [22] used NB, SVM, long short-term memory networks (LSTM), and RF classifiers and compared the findings. The N-gram extraction was used after preprocessing the datasets. The NB model has the highest accuracy on most datasets, such as the BJP and ML Khattar datasets. Samuel et al. [23] used NB and logistic regression (LR) models on tweets about COVID-19. The tweets are transformed into a text corpus, and then the most frequent words are identified using N-grams. Their results indicated a high accuracy of 91% with the NB method and an accuracy of 74% with LR for short tweets, and longer tweets performed relatively worse for both models.

Kumar *et al.* [24] examined how gender and age affected the customer reviews that had been gathered. Maximum entropy (ME), SVM, and LSTM models are applied. The NB, ME, and SVM algorithms all employ the Bag of Words (BOW) feature extraction, while word2vec is used in the LSTM model. The best accuracy for female data is with a group over 50.

Zarisfi *et al.* [25] used SVM and MNB with TF-IDF extraction on four Twitter datasets, namely the Strict Obama-McCain Debate dataset, the Obama-McCain Debate dataset, the STS-Gold dataset, and the Stanford testing dataset. Semantic scoring based on tweet class, semantic similarity, SWN scoring, and TF-IDF methods have been suggested for representing the features in the vector space. In three datasets, the proposed method outperformed the MNB algorithm. The MNB algorithm performs the best of all methods in the STS dataset.

2.2 Gap in literature

Previous studies indicate that there is a paucity of literature that needs to be discussed. There is a limited range of techniques used for feature extraction in previous works. The aim of this paper is to address this gap by evaluating different feature extraction techniques on the same dataset when using sentiment classification to choose the most suitable method. We want the best possible results when doing classification, so the method that we choose for feature extraction is important.

3. Proposed system

This section addresses the description of the datasets and preprocessing steps, as well as the feature extraction techniques, the SMOTE technique, and the classification model. Fig 1 illustrates the architectural design of our experiment, while Algorithm 1 provides a summary of our proposed system.

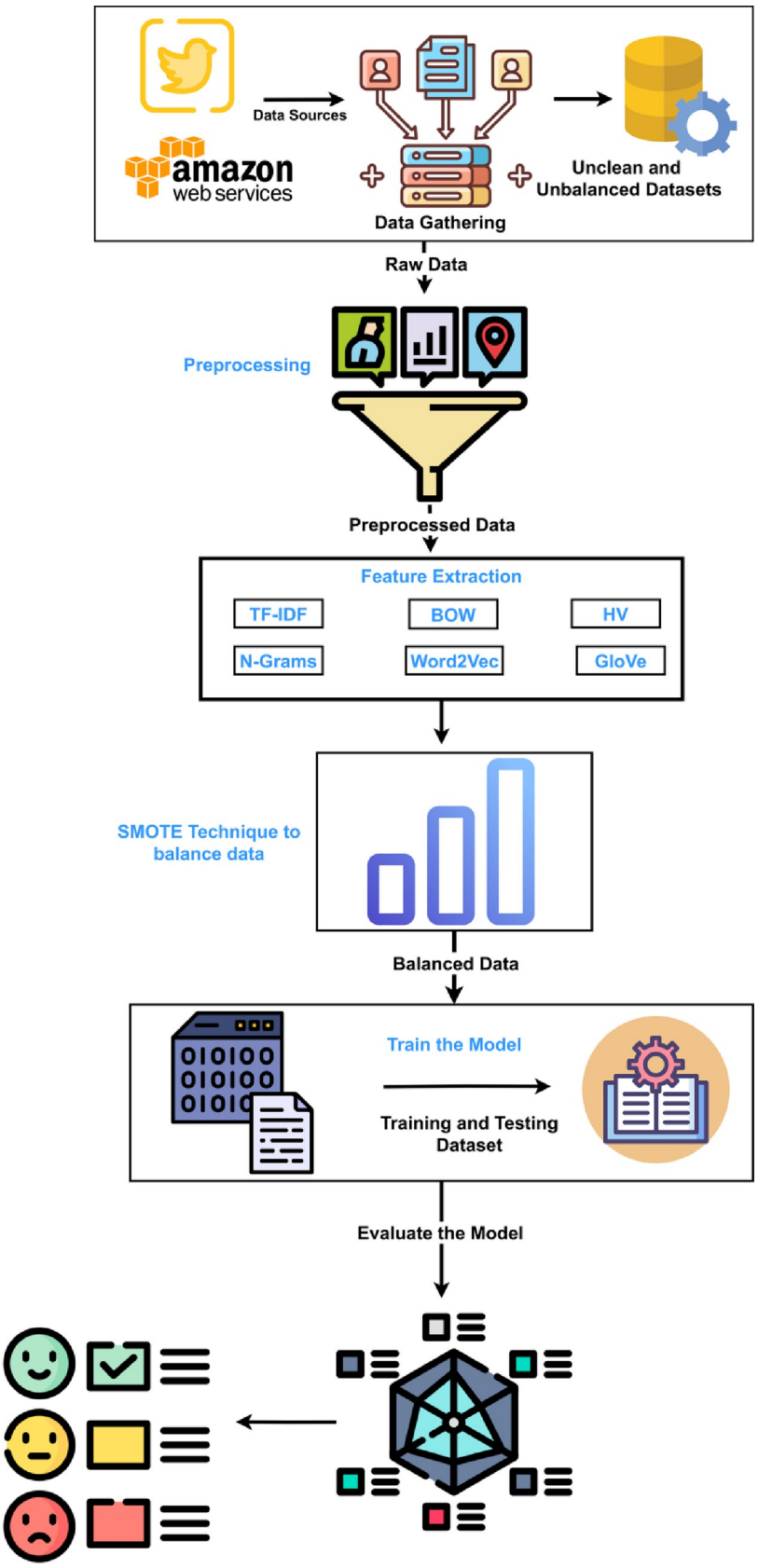

Fig 1. The architecture of the proposed model for sentiment classification.

https://doi.org/10.1371/journal.pone.0294968.g001

```
Algorithm 1. Framework of Our Proposed System.
Input: Raw dataset D
Output: Sentiment (Positive or Negative or Neutral)
Begin
  Clean input D (remove special symbols, stop-words, emoji,
URL, tokenization, etc)
  Assign sentiment labels to D
  Apply feature extraction to transform D into a feature vector
  Apply the SMOTE technique to balance the dataset
  Classify D using the random forest model
End
```

3.1 Dataset description

For the experiments, we picked two different datasets that consist of real-world user feedback, reflecting the opinions and sentiments of actual customers and users, making them a highly demanded source for researchers in the field. The first dataset is Twitter US Airlines, which CrowdFlower created in 2017. It offers a comprehensive collection of customer reviews of six significant American airlines and contains various features, as shown in Table 2. It has 14640 instances, out of which 2363 are positive tweets, 9178 are negative tweets, and the remaining 3099 are neutral tweets [26]. The second dataset is Amazon musical instrument reviews collected in 2020, which offer a rich collection of customer feedback and contain various features, as shown in Table 3. It has 10261 instances out of which 9022 are positive reviews, 467 are negative reviews, and the remaining 772 are neutral reviews [27]. The datasets are available at https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment and https://www.kaggle.com/datasets/eswarchandt/amazon-music-reviews. It can be observed that the two datasets have imbalanced data. Figs 2 and 3 illustrate the distribution of sentiment classes in the datasets.

3.2 Text preprocessing

The preprocessing step is essential in the sentiment analysis process. It transforms text into a format suitable for machine learning algorithms [28]. The preprocessing of text includes removing retweets because duplicate tweets might skew word frequency and increase the amount of space needed for running the experiment. In the next step, URLs should be removed since they have no meaning and won't affect sentiment. Removing punctuation, emojis, non-alphanumeric characters, and stop words is critical because they are not helpful for analysis, and in tokenization, the entire text or paragraph is divided into smaller units, known as tokens [29]. Finally, the lemmatization process removes inflectional endings and

Table 2. Feature description of the Twitter US airlines dataset.

Features	Description
Airline Sentiment Confidence	A numerical attribute that quantifies the amount of confidence in the classification of a tweet into one of three distinct classes.
Negative Reason	The rationale for deeming this tweet as having a negative connotation.
Negative Reason Confidence	The degree of certainty in establishing the underlying cause of a negative tweet.
Airline	The airline company's name.
Retweet Count	The quantification of retweets received by a specific tweet.
Text	Tweet initially published by the user.
Airline Sentiment	Labels for tweets (positive, negative, neutral).

https://doi.org/10.1371/journal.pone.0294968.t002

Table 3. Feature description of the Amazon dataset.

Features	Description
ReviewerID	ID of the reviewer
ASIN	ID of the product
Reviewer name	Name of the reviewer
Helpful	Helpfulness rating of the review
Review text	Text of the review
Overall	Rating of the product
Summary	Summary of the review.
UnixReviewTime	Time of the review (unix time).
ReviewTime	Time of the review (raw)

https://doi.org/10.1371/journal.pone.0294968.t003

returns the base or dictionary form of words, and the stemming process reduces words into word stems because some of the words might not be proper in the language. The WordNetLemmatizer lemmatization and PorterStemmer stemming were used for this study. Table 4 shows some examples before and after the preprocessing step from Amazon musical instrument reviews.

3.3 Feature extraction

In this study, the main contribution is the extraction of important features from datasets. The process of feature extraction holds significant importance in text processing as it effectively decreases the dimensionality of the feature space by selectively emphasizing the crucial aspects. Hence, in this work, we employed six different feature extraction methods, including Bow, TF-IDF, n-grams with a range of (1,2) which includes both unigrams (individual words) and

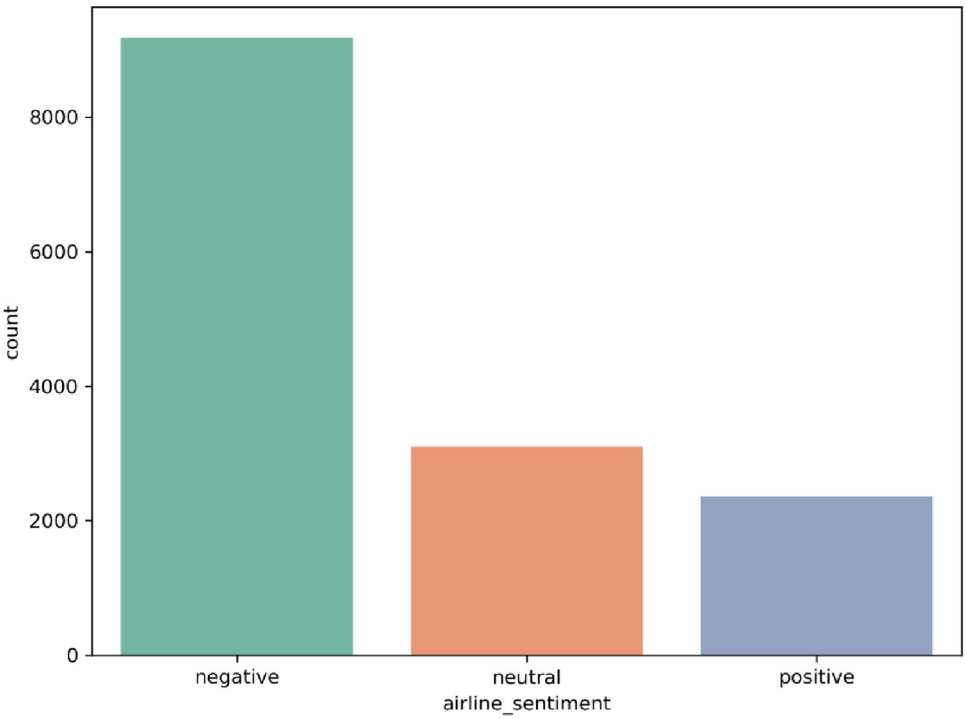

Fig 2. Sentiment data distribution of the Twitter dataset.

https://doi.org/10.1371/journal.pone.0294968.g002

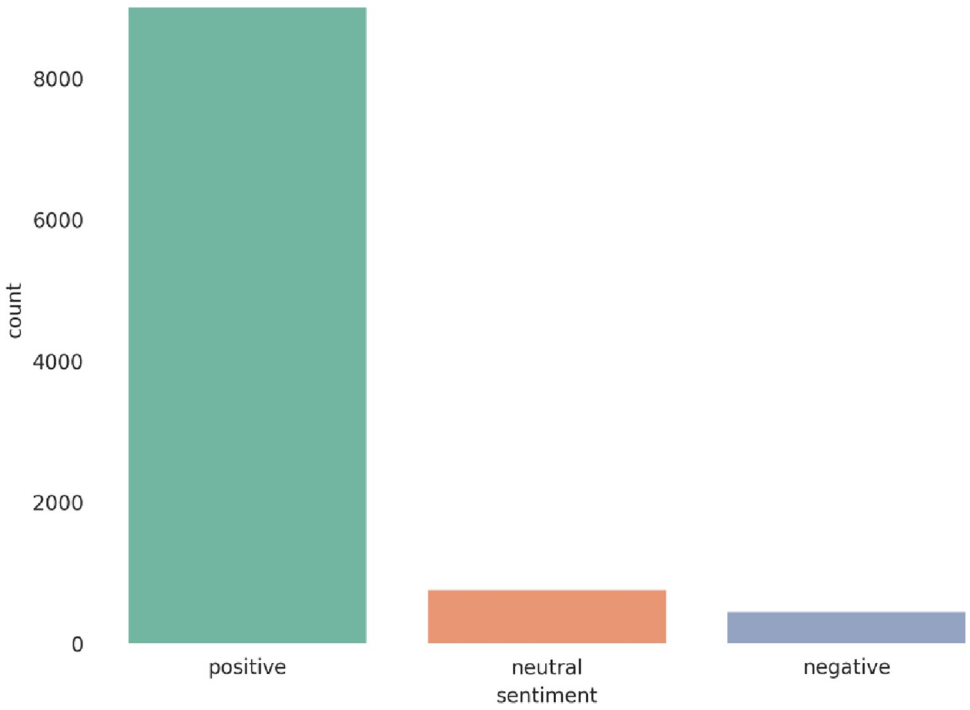

Fig 3. Sentiment data distribution of the Amazon dataset.

https://doi.org/10.1371/journal.pone.0294968.g003

bigrams (pairs of consecutive words), global vector for word representation (GloVe), hashing vectorizer (HV), and word2vec, to extract features from the datasets, as shown in Fig 4. The selection of these specific feature extraction techniques is based on their established effectiveness in sentiment analysis tasks and their ability to capture different aspects of text data [29]. The chosen feature extraction techniques can improve classification or prediction accuracy and maximize the utility and relevance of the feature extraction process, leading to more meaningful and impactful outcomes.

3.3.1 TF-IDF. This method is derived from language modeling theory. According to the theory, words in a text can be divided into two categories based on their eliteness: those with eliteness and those without. Its calculation is based on a combination of two metrics, one of which measures how many times a word appears in a collection of documents, and the other measures the word's inverse document frequency. In a document, term frequency (TF) counts the number of times words appear, and inverse document frequency (IDF) is a method that helps distinguish and classify documents easily by giving importance or weightage to words that are unique to a certain set of documents [30]. Words in the document with high or low-frequency terms are given more weight by the IDF. Combining TF and IDF is known as TF-IDF. According to Eq (1), the mathematical representation of the weight of a term in a

Table 4. Some examples of Amazon reviews dataset.

Reviews	Reviews after preprocessing
Not much to write about here, but it does exac. . .	much write exactli suppos filter pop sound rec. . .
The product does exactly as it should and is q. . .	product exactli quit affordablei realiz doubl. . .
The primary job of this device is to block the. . .	primari job devic block breath would otherwis. . .
Nice windscreen protects my MXL mic and preven. . .	nice windscreen protect mxl mic prevent pop th. . .

https://doi.org/10.1371/journal.pone.0294968.t004

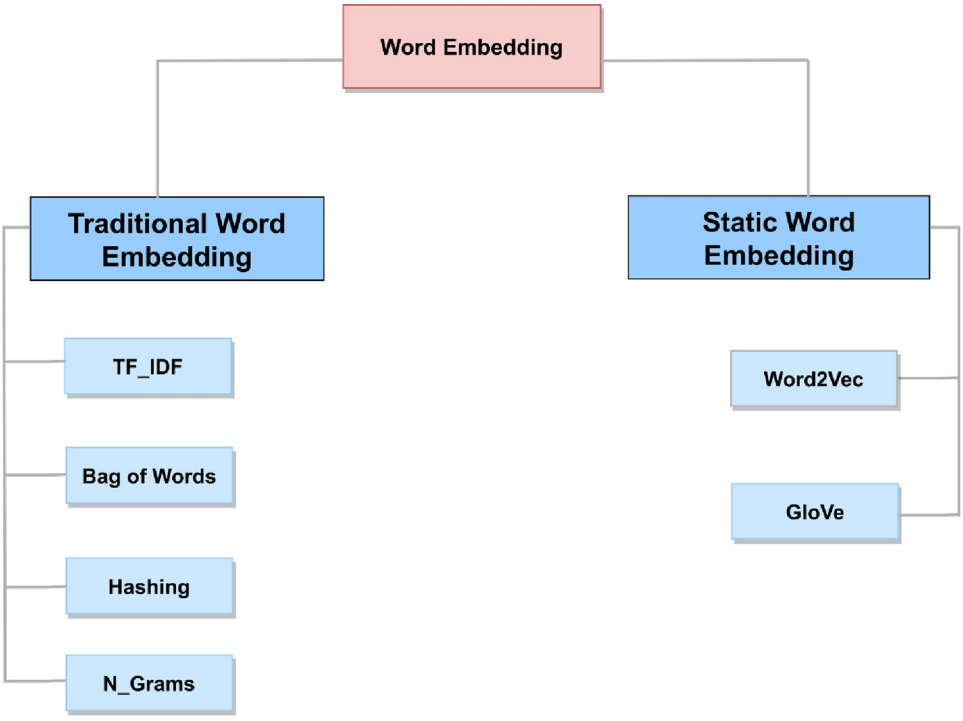

Fig 4. The general structure of the word representation models.

https://doi.org/10.1371/journal.pone.0294968.g004

document by the TF-IDF method.

$$W(d,t) = TF(d,t) * \log\left(\frac{N}{df(t)}\right) \quad (1)$$

In this equation, N denotes the number of documents, and df (t) indicates how many documents contain the term t in the corpus. The initial term introduces an enhancement to recall, while the subsequent term contributes to precision. Although TF-IDF endeavors to address the issue of frequently occurring terms in a document, it is not without its constraints; for instance, when each word is displayed independently as an index, it is incapable of considering word similarity. However, TF-IDF vectors exhibit superior accuracy compared to alternative methodologies.

3.3.2 BOW. It is one of the simplest feature extraction model categories, and it does not take the order of the words into account. This model is a method of encoding text data. It is simple to use and learn, and it has proven to work effectively for document classification and language modeling. There are some limitations of BOW models like sparsity, and the complete disregard for word order ignores the context, which in turn ignores the semantics of the words used in the document [31]. Three steps describe how this model works: the first step is text tokenization, then tokenizing each sentence into words and counting how many times each word appears in each sentence, and finally, constructing the model by creating a vector to identify If the word is frequently used, it might be set to 1, otherwise 0 and generate the output.

3.3.3 N-Grams. The n-gram technique consists of any sequence of n-words that occur "in that order" in a text set. This technique was the first to attempt to impose a window to capture the ordering among words. The n-gram method ignores individual words and instead focuses

on multiword tokens and their ordering within the context window. The N-gram does not necessarily capture contextual information, but it is effective at capturing word ordering among words. When these words appear together, they may have an entirely different meaning than when they appear separately. This model is relatively easy to obtain, and a manageable-sized vector can be used to represent the text [32]. In this study, the n-gram with a range of (1, 2) is used, and this range refers to a combination of unigrams (single words) and bigrams (pairs of consecutive words) in a given text.

3.3.4 Hashing vectorizer. In the hashing vectorization (HV) method, collections of review text are transformed into a matrix of token occurrences. A hashing vectorizer returns the account for every token in the document, so it is no different from a regular BOW model in terms of how text features are turned into a numeric representation. However, hashing vectorizers have the following advantages: they scale better with large document sets and work well with batch processing [33]. The limitation is the potential for hash collisions, and the larger feature space can introduce additional computational overhead, leading to longer training times.

3.3.5 Static word embedding. Word embeddings are numerical representations of words or phrases that depict the relationships between them in a multidimensional space as well as their semantic meaning. These representations are typically learned from large amounts of text data using neural networks. There are some key characteristics of word embedding such as similar words having the same embedding, values and each word having a distinct word embedding or vector, which is only a list of numbers for each word. This study uses the word2vec model and the glove model, two of the most popular algorithms for word embeddings. The first model is the Word2Vec Model was first introduced by [34], is popular and widely used in learning word embeddings from raw text. Based on the idea of distributed representation of words, word2vec (word embeddings) uses a shallow neural network to learn word embeddings and predict the relation between every word and its context words. With this method, relevant information from the texts is captured, resulting in good results.

In word2vec, SG (skip-gram) and CBOW (Continuous Bag-of-Words) algorithms are used to produce word vectors [34]. The SG model is used to store semantic and syntactic information about sentences. In this study, the SG model is implemented with a vector size of 100, which means that each word will be represented by a vector of length 100, and a window size of 5, which shows the maximum distance between the current and predicted word within a sentence. The choice of the value of the window parameter balances between capturing local context and capturing broader semantic relationships, and the vector provides a good balance between capturing semantic information and computational efficiency. The aim of this model is to maximize the classification of words based on other words in the same sentence.

The GLOVE (Global Vectors for Word Representation) method has been developed by [35]. This method is used for producing word embeddings and is an unsupervised procedure. A meaningful space is constructed for the words, in which the distance between words correlates with semantic similarity. The global word cooccurrence matrix is aggregated from a corpus for training purposes. As a result, the resulting representations of the word exhibit interesting linear substructures in vector space. In this model, a large corpus of data has been used to train it [36]. The model is not able to capture out-of-vocabulary words from the corpus and consumes a great deal of memory during storage. It is effective and scalable for huge corpora because it combines latent semantic analysis and CBOW. We perform experiments using a vector embedding dimension of 300.

3.4 Synthetic minority over-sampling technique

The distribution of positive, neutral, and negative polarities in the datasets in this study is unbalanced. This imbalanced data may have a significant negative impact on the machine learning models' performance because it may tilt the decision surface in favor of the majority class. The oversampling approach is used to solve the issue of class imbalance. This approach works by increasing the size of the data, which creates more features for model training and could be helpful to enhance the model's accuracy. In this study, we use the synthetic minority oversampling (SMOTE) method for oversampling. The SMOTE is a state-of-the-art method proposed by [37, 38]. This method was selected because it avoids information loss, is simple to interpret and implement, and helps to solve the overfitting issue for unbalanced datasets. Randomly, SMOTE selects the smaller classes and finds their K-nearest neighbors. Based on the K-nearest neighbor for each selected sample, a new minority class is constructed [39]. With 70:30 ratios, the data is divided into training and testing sets after the oversampling process.

3.5 Machine learning model

The proposed ensemble classifier was trained on the training set for classifying the sentiments in the datasets and evaluated on the test data. The ML algorithm used in this work is a random forest classifier. This model was chosen for this study because it helps to avoid overfitting, provides a measure of feature importance, and produces a reasonable prediction without adjusting hyperparameters. This is a supervised ML algorithm that is used for regression and classification purposes and belongs to the ensemble learning family [40]. In the random forest model, decision trees are constructed from datasets and create a forest made of trees. A random forest classifier consists of the following steps: The first step is to select random data samples from the available dataset. For each selected data sample, a decision tree is constructed, and a prediction value is extracted from each decision tree. For node splitting, the Gini coefficient method is applied as follows:

$$Gini(D) = 1 - \sum_{i=1}^{n} p_i^2 \tag{2}$$

Where D represents the dataset and Pi represents the probability of decision classes appearing in D. After obtaining prediction values from each decision tree, a voting method is applied. The final prediction result is selected based on the prediction value with the most votes [41]. In order to increase accuracy, RF was implemented with n-estimators equal to 100, which indicates how many trees contributed to the prediction. To decrease the probability of the decision tree overfitting, the 'max_depth' setting is set to 5, which shows that every decision tree can go to a maximum of five levels.

3.6 Performance measures

To examine the performance of the suggested model using different feature extraction techniques, we used several standard performance measures. Specifically, we used recall, accuracy, precision, and F1-measure. To calculate all four metrics, machine learning models can be visualized by using a confusion matrix [42, 43]. The elements of this matrix are False Negative (FN), True Positive (TP), False Positive (FP), and True Negative (TN). The performance evaluation of classifiers is made according to the following formulas:

$$\text{Accuracy}: \quad \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$\text{Precision}: \quad \frac{TP}{TP + FP} \tag{4}$$

$$\text{Recall}: \quad \frac{TP}{TP + FN} \tag{5}$$

$$\text{F1}-\text{Measure}: \quad 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

4. Experimental results

We conducted the experiments on Google Colab, a cloud-based graphical processing unit (GPU)-based platform offered by Google Inc. The classification algorithm was implemented using the Scikit-learn library. Due to the moderate size of the dataset, ML algorithms are used rather than deep learning algorithms for classification. We conducted experiments using a set of datasets that are commonly used in sentiment analysis by applying a random forest classifier using different word representation models and based on the parameters shown in Table 5.

It is observed that the sentiment classes in the datasets are imbalanced, so the SMOTE technique is applied. For the two datasets, a total of 70% is used for the training process, and the other 30% is used for testing using the random forest model as shown in Table 6. The performance of the random forest algorithm is evaluated on different metrics such as recall, precision, accuracy, and F1-measure.

Computational efficiency is calculated by using the training time which is the time it takes to train the model, and the prediction time which is the time it takes to predict the labels for a new set of instances after each feature extraction. A comparison of all the feature extraction methods on the Twitter dataset is shown in Table 7, where the TF-IDF and HV methods achieve the highest accuracy, but the TF-IDF is much faster than others. The n-gram has the lowest accuracy, but it also has a low training time.

A comparison of all the feature extractions on the Amazon reviews dataset is shown in Table 8. The TF-IDF achieves the highest precision, accuracy, recall, and F1-measure. It also has the lowest training time.

Table 5. The parameters tuned with respect to the random forest model.

Parameters	Values
n_estimators	100
Criterion	Gini
max_depth	5
max_features	sqrt
random_state	42

https://doi.org/10.1371/journal.pone.0294968.t005

Table 6. The total size and size of (train/test) of the datasets.

Dataset	Total size	Training Set size	Testing Set size
Twitter US airlines	14640	10248	4392
Amazon musical instrument reviews	10261	7183	3078

https://doi.org/10.1371/journal.pone.0294968.t006

Table 7. Performance and time of the random forest classifier on a Twitter dataset.

Feature extraction	accuracy	Precision	Recall	F1-measure	Training time	Prediction time
TF_IDF	96	95	96	95	11.285836	0.497233
N_Gram	86	87	86	86	13.926802	0.541020
BOW	87	87	87	87	16.031671	0.535141
Hashing Vectorizer	96	96	96	96	79.441338	0.809710
Word2Vec	93	93	93	93	19.753669	0.214723
Glove	92	92	92	92	35.825151	0.180461

https://doi.org/10.1371/journal.pone.0294968.t007

Fig 5 displays a comparison of the training time for the proposed model following the dataset's feature extraction. The HV method requires significantly more training time compared to other methods.

Fig 6 displays the testing time for the proposed model following the dataset's feature extraction. The HV method takes a longer prediction time, followed by the n-gram and Bow methods. Fig 7 displays the proposed model's accuracy. The highest accuracy values of the proposed model on the datasets for TF-IDF and HV methods.

5. Discussion

In this section, we will have an overall discussion of the experimental results from the previous section. It has been noted that all the feature extraction methods performed well, with high accuracy and balanced precision, recall, and F1-measure, so the model's performance is not skewed by the majority class and the model can generalize well to all classes. This suggests that the other methods are also capable of extracting important features from the text data.

From Fig 6, the comparison between the outcomes proves that the performance of the model is the highest after the TF_IDF and HV methods for both datasets. The TF-IDF achieves an accuracy of 99% with the Amazon dataset and 96% with the Twitter dataset. The performance of the model is improved, especially on TF-IDF vectorization because the model can benefit from the ability of this extractor to focus on important and discriminative terms while down-weighting common and less informative terms.

BOW performs similarly to the n-gram method, with slightly lower accuracy and F1-measure in which the accuracy of BOW is 90% on the Amazon dataset and 87% on another dataset. The BOW method shows consistent precision and recall across both datasets, indicating that it maintains a good balance between correctly identifying sentiments. From Fig 4 the training time was relatively fast in the TF-IDF but the training time of HV is the longest in both datasets due to the hashing process and the potential for hash collisions. The Word2Vec and GloVe models have slightly lower accuracy than TF-IDF, but Word2Vec is much faster to train than the GloVe model, especially for the Amazon reviews dataset. From Fig 5 the prediction time of

Table 8. Performance and time of the random forest classifier on the Amazon dataset.

Feature extraction	Accuracy	Precision	Recall	F1-measure	Training time	Prediction time
TF_IDF	99	99	99	99	8.738268	0.199666
N-gram	89	93	92	92	10.649860	0.217366
BOW	90	91	90	90	10.967826	0.209918
Hashing Vectorizer	98	98	99	98	62.155688	1.326388
Word2Vec	96	96	96	96	8.990855	0.101492
Glove	97	97	97	97	48.304039	0.231222

https://doi.org/10.1371/journal.pone.0294968.t008

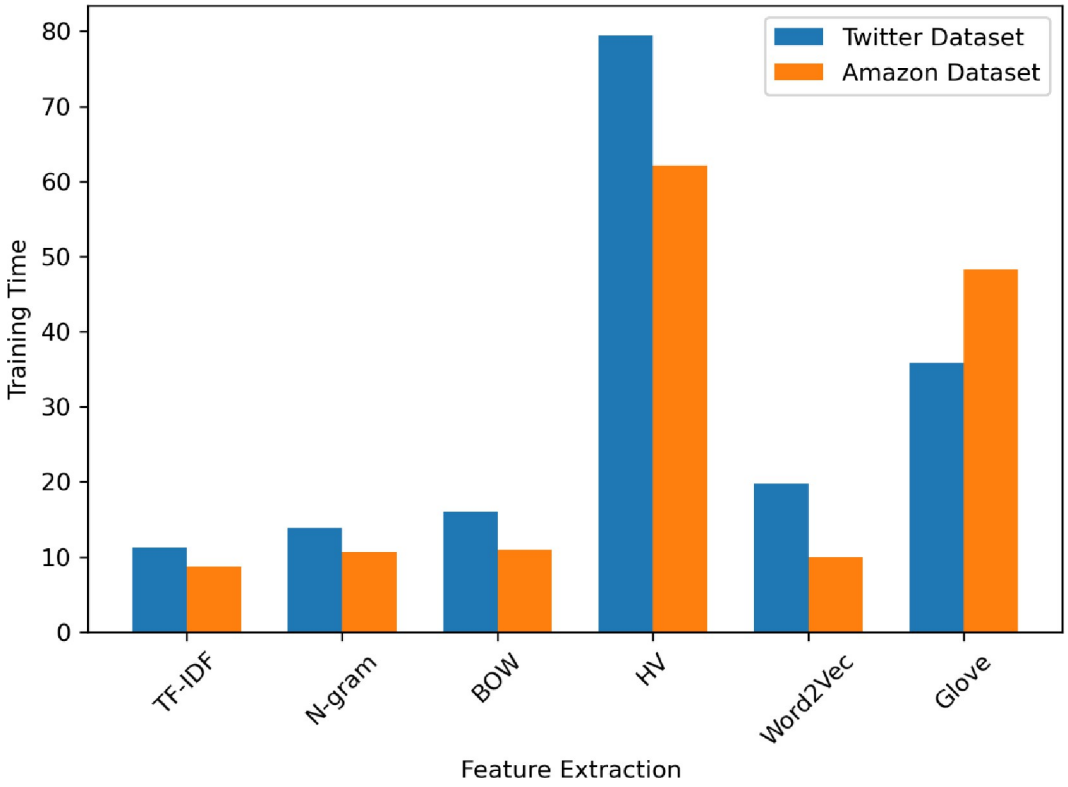

Fig 5. The training time of the datasets.

https://doi.org/10.1371/journal.pone.0294968.g005

the HV method is the longest on both datasets but the prediction time of TF-IDF is relatively the same on both datasets.

Overall, the TF-IDF extractor provides a good balance between performance across all evaluation metrics and computational efficiency. Thus, it is important to consider the trade-off between training time and the specific requirements of the sentiment analysis task when choosing a feature extraction method because some techniques may yield higher accuracy, but the training time becomes too long, and this may not be practical for real-time sentiment analysis applications.

Our experiment has shown that selecting the right feature extraction method has a significant impact on the performance of an ML algorithm, which means that rather than spending a lot of time optimizing a specific classifier, it might be worthwhile to spend more time choosing the right feature extraction method. Also, the impact is on business organizations that may be able to detect negative reviews more efficiently. In a short period of time, business organizations can learn about customer demand after inspecting negative reviews, and they can reshape their products and policies accordingly.

Although we have shown successful feature extraction-based sentiment analysis and ML, there are several limitations to this work that could be explored in the future: this study is based on only English-language reviews that were analyzed and another limitation is that we have only tested the random forest model in our experiments.

6. Conclusion

In the last few years, feature extraction and machine learning have become more popular for analysis and prediction. The effectiveness of sentiment analysis on social media is studied in

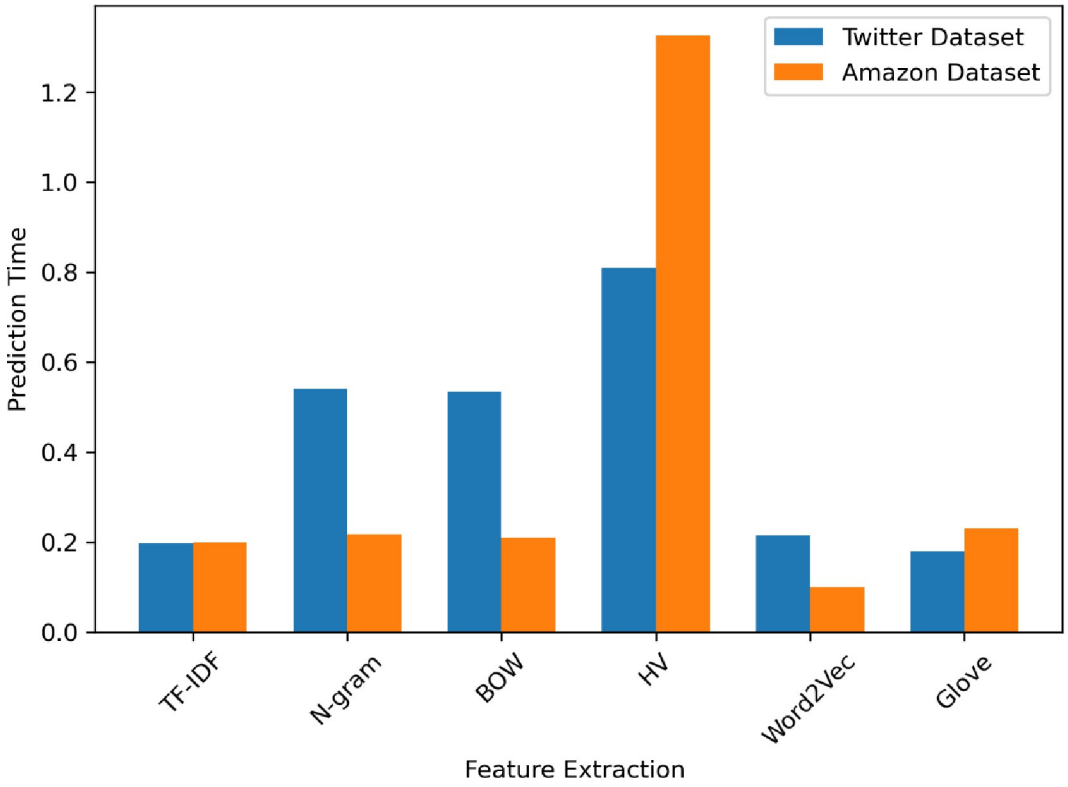

Fig 6. The prediction time of the datasets.

https://doi.org/10.1371/journal.pone.0294968.g006

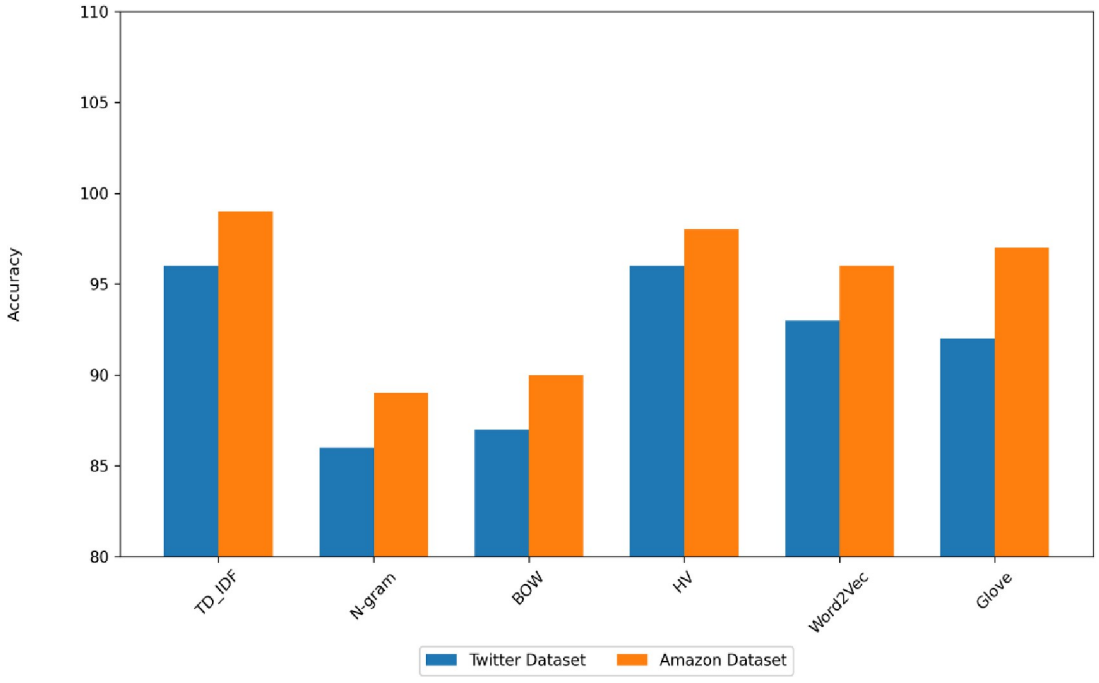

Fig 7. The accuracy of the datasets.

https://doi.org/10.1371/journal.pone.0294968.g007

this research using six distinct feature extraction techniques, and the key findings are discussed. So, in this work, there are two different datasets from diversified social media platforms to evaluate the performance of the suggested model. A data preprocessing stage is executed on the dataset to remove several superfluous symbols and then employ feature extraction with the SMOTE technique. A state-of-the-art ML algorithm is used to train the extracted features, namely the random forest algorithm. After each feature extraction, the ML algorithm's performance is evaluated.

On both datasets, the random forest offers the highest accuracy with TF-IDF and fewer training and prediction times than others. The results indicate that the choice of suitable methods for feature extraction plays a crucial role in determining the effectiveness of sentiment analysis tasks, with some techniques performing better than others. These findings have important implications for practitioners and researchers working in the field of sentiment analysis. They suggest that careful consideration should be given to the choice of feature extraction techniques when developing sentiment analysis models for social media.

In future studies, the analysis can be expanded to include other languages, such as Arabic, and can explore other machine learning models, such as deep learning models or transformers, to see if they can improve the accuracy of sentiment analysis on imbalanced datasets with different feature extraction techniques. Additionally, we can use hybrid feature extraction techniques to explore the impact of this improvement on the performance of the sentiment analysis classification. Finally, we intend to apply our method to more recent datasets.

Supporting information

S1 File. The file contains the data and supporting tables.
(DOCX)

Acknowledgments

The authors would like to acknowledge Prince Sultan University for their valuable support.

Author Contributions

Conceptualization: Wesam Ahmed, Mohamed Hammad.

Data curation: Wesam Ahmed.

Funding acquisition: Paweł Pławiak.

Investigation: Noura A. Semary, Paweł Pławiak, Mohamed Hammad.

Methodology: Wesam Ahmed.

Project administration: Paweł Pławiak.

Software: Wesam Ahmed.

Supervision: Noura A. Semary, Khalid Amin, Mohamed Hammad.

Validation: Khalid Amin, Mohamed Hammad.

Visualization: Noura A. Semary.

Writing – original draft: Wesam Ahmed, Mohamed Hammad.

Writing – review & editing: Wesam Ahmed, Paweł Pławiak, Mohamed Hammad.

References

1. Birjali M, Kasri M, Beni-Hssane A. A comprehensive survey on sentiment analysis: approaches, challenges and trends. Knowl-Based Syst 226:107134. (2021); 226. https://doi.org/10.1016/j.knosys.2021.107134.
2. Omar A, Abd El-Hafeez T. Quantum computing and machine learning for Arabic language sentiment classification in social media. Scientific Reports.2023. https://doi.org/10.1038/s41598-023-44113-7 PMID: 37828056
3. Khairy M, Mahmoud TM, Omar A, Abd El-Hafeez T. Comparative performance of ensemble machine learning for Arabic cyberbullying and offensive language detection. Language Resources and Evaluation. 2023. https://doi.org/10.1007/s10579-023-09683-y.
4. Mamdouh F H, Abd El-Hafeez T. A new feature selection method based on frequent and associated itemsets for text classification. Concurrency and Computation: Practice and Experience. 2022. https://doi.org/10.1002/cpe.7258.
5. Omar A, Mahmoud TM, Abd-El-Hafeez T, Mahfouz A. Multi-label arabic text classification in online social networks. Information Systems. 2021. https://doi.org/10.1016/j.is.2021.101785.
6. Khairy M, Mahmoud TM, Abd-El-Hafeez T. Automatic detection of cyberbullying and abusive language in Arabic content on social networks: a survey. Procedia Computer Science. 2021 Jan 1; 189:156–66. https://doi.org/10.1016/j.procs.2021.05.080.
7. Farghaly HM, Ali AA, El-Hafeez TA. Developing an Efficient Method for Automatic Threshold Detection Based on Hybrid Feature Selection Approach. InArtificial Intelligence and Bioinspired Computational Methods: Proceedings of the 9th Computer Science On-line Conference Springer International Publishing.2020. https://doi.org/10.1007/978-3-030-51971-1_5.
8. Farghaly HM, Ali AA, Abd El-Hafeez T. Building an Effective and Accurate Associative Classifier Based on Support Vector Machine. SYLWAN. 2020.
9. Mamdouh F H, Abd El-Hafeez T. A high-quality feature selection method based on frequent and correlated items for text classification. Soft Computing. 2023. https://doi.org/10.1007/s00500-023-08587-x.
10. Goodrum H, Roberts K, Bernstam EV. Automatic classification of scanned electronic health record documents. Int J Med Inform 144:104302, 144. 2020; 144. https://doi.org/10.1016/j.ijmedinf.2020.104302 PMID: 33091829
11. Blanco A, Perez-de-Vinaspre O, Perez A, Casillas A. Boosting ICD multi-label classification of health records with contextual embeddings and label-granularity. Comput Methods Programs Biomed. 2020; 188. https://doi.org/10.1016/j.cmpb.2019.105264 PMID: 31851906
12. Alqaisi T, O'Keefe S. En-Ar bilingual word embeddings without word alignment: Factors Effects. In: Proc Fourth Arab Nat Lang Process Work— Assoc Comput Linguist ANLPW-ACL-2019, 97–107. 2019. https://doi.org/10.18653/v1/w19-4611.
13. Li Y, Yang T. Word embedding for understanding natural language: a survey. Big Data Appl. 2018; 26. https://doi.org/10.1007/978-3-319-53817-4_4.
14. Sun F, Guo J, Lan Y, Xu J, & Cheng X. Learning word representations by jointly modeling syntagmatic and paradigmatic relations. n Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing.2015; 1: 136–145.
15. Lai S, Liu K, Xu L, & Zhao J. How to generate a good word embedding?. IEEE Intelligent Systems. 2016; 31: 5–14. https://doi.org/10.1109/MIS.2016.45.
16. Ahmed J, Ahmed M. Classification, detection, and sentiment analysis using machine learning over next-generation communication platforms. Microprocessors and Microsystems. 2023;98. https://doi.org/10.1016/j.micpro.2023.104795.
17. Gaur P, Vashistha S, Jha P. Twitter Sentiment Analysis Using Naive Bayes-Based Machine Learning Technique. In: Shakya S., Du KL., Ntalianis K. (eds) Sentiment Analysis and Deep Learning. Advances in Intelligent Systems and Computing, Springer, Singapore. 2023;1432. https://doi.org/10.1007/978-981-19-5443-6_27.
18. Qi Y, Shabrina Z. Qi Y, Shabrina Z. Sentiment analysis using Twitter data: a comparative application of lexicon-and machine-learning-based approach. Social Network Analysis and Mining. 2023;13. https://doi.org/10.1007/s13278-023-01030-x.
19. Al sari B., Alkhaldi R., Alsaffar D. et al. Sentiment analysis for cruises in Saudi Arabia on social media platforms using machine learning algorithms. *Journal of Big Data*.2022; 9,1–28. https://doi.org/10.1186/s40537-022-00568-5 PMID: 35223367
20. Mukherjee P, Badr Y, Doppalapudi S, Srinivasan SM, Sangwan RS, Sharma R. Effect of negation in sentences on sentiment analysis and polarity detection. Procedia Computer Science. 2021; 1:185:370–9. https://doi.org/10.1016/j.procs.2021.05.038.

21. Noori B. Classification of Customer Reviews Using Machine Learning Algorithms. Applied Artificial Intelligence.2021:567–588. https://doi.org/10.1080/08839514.2021.1922843.

22. Zahoor S, and Rohilla R. Twitter Sentiment Analysis Using Machine Learning Algorithms: A Case Study. 2020 International Conference on Advances in Computing, Communication & Materials. 2020:194–199. https://doi.org/10.1109/ICACCM50413.2020.9213011.

23. Samuel J, Ali GM, Rahman MM, Esawi E, Samuel Y. Covid-19 public sentiment insights and machine learning for tweets classification. Information.2020;11 https://doi.org/10.3390/info11060314.

24. Kumar S, Gahalawat M, Roy PP, Dogra DP, Kim BG. Exploring Impact of Age and Gender on Sentiment Analysis Using Machine Learning. Electronics.2020;9. https://doi.org/10.3390/electronics9020374.

25. Zarisfi K F, Sadeghi F, Eslami E. Solving the twitter sentiment analysis problem based on a machine learning-based approach. Evolutionary Intelligence. 2020; 13:381–98. ttps://doi.org/10.1007/s12065-019-00301-x.

26. Tan KL, Lee CP, Anbananthen K SM, Lim K M. RoBERTa-LSTM: a hybrid model for sentiment analysis with transformer and recurrent neural network. *IEEE Access*, 2022; 10: 21517–21525. https://doi.org/10.1109/ACCESS.2022.3152828.

27. Campos P, Pinto E, Torres A. Rating and perceived helpfulness in a bipartite network of online product reviews. Electronic Commerce Research. 2023;1–33. https://doi.org/10.1007/s10660-023-09725-1.

28. Chopra M, Singh SK, Aggarwal K, Gupta A. Predicting catastrophic events using machine learning models for natural language processing. In: Data mining approaches for big data and sentiment analysis in social media. IGI Global. 2022:223–243. https://doi.org/10.4018/978-1-7998-8413-2.ch010.

29. Chong WY, Selvaretnam B, Soon LK. Natural language processing for sentiment analysis: an exploratory analysis on tweets. In: 2014 4th international conference on artificial intelligence with applications in engineering and technology. IEEE. 2014:212–217. https://doi.org/10.1109/ICAIET.2014.43.

30. Bordoloi M, Biswas SK. Sentiment analysis: A survey on design framework, applications and future scopes. Artificial Intelligence Review. 2023; 20:1–56. https://doi.org/10.1007/s10462-023-10442-2 PMID: 37362892

31. Liaqat MI, Hassan MA, Shoaib M, Khurshid SK, Shamseldin MA. Sentiment analysis techniques, challenges, and opportunities: Urdu language-based analytical study. PeerJ Computer Science. 2022;8. https://doi.org/10.7717/peerj-cs.1032 PMID: 36091980

32. Gohil S, Vuik S, Darzi A. Sentiment analysis of health care tweets: review of the methods used. JMIR Public Health Surveill. 2018; 4. https://doi.org/10.2196/publichealth.5789 PMID: 29685871

33. Barbounaki SG, Gourounti K, Sarantaki A. Advances of Sentiment Analysis Applications in Obstetrics/Gynecology and Midwifery. Mater Sociomed.2021:225–230. https://doi.org/10.5455/msm.2021.33.225-230 PMID: 34759782

34. Chen Q, Sokolova M. Specialists, scientists, and sentiments: Word2Vec and Doc2Vec in analysis of scientific and medical texts. SN Computer Science. 2021; 2:1–11. https://doi.org/10.1007/s42979-021-00807-1 PMID: 34414378

35. Mikolov T, Yih WT, Zweig G. Linguistic regularities in continuous space word representations. In Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies. 2013: 746–751.

36. Pennington J., Socher R. and Manning C.D., (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 1532–1543.

37. Sunitha D, Patra RK, Babu NV, Suresh A, Gupta SC. Twitter sentiment analysis using ensemble based deep learning model towards COVID-19 in India and European countries. Pattern Recognition Letters. 2022; 158:164–70. https://doi.org/10.1016/j.patrec.2022.04.027 PMID: 35464347

38. Maciejewski T, Stefanowski J. Local neighbourhood extension of SMOTE for mining imbalanced data. In2011 IEEE symposium on computational intelligence and data mining (CIDM) 2011 Apr 11 (pp. 104–111). IEEE. https://doi.org/10.1109/CIDM.2011.5949434.

39. Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C. Safe-Level-Synthetic Minority Over-Sampling Technique for Handling the Class Imbalanced Problem. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer.2009:475–482. https://doi.org/10.1007/978-3-642-01307-2_43.

40. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research. 2002; 16:321–57. https://doi.org/10.1613/jair.953.

41. Reddy Maddikunta PK, Srivastava G, Reddy Gadekallu T, Deepa N, Boopathy P. Predictive model for battery life in IoT networks. IET Intelligent Transport Systems. 2020; 14:1388–95. https://doi.org/10.1049/iet-its.2020.0009

42. Yan X, Jin Y, Xu Y, Li R. Wind turbine generator fault detection based on multi-layer neural network and random forest algorithm. In 2019 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia). 2019: 4132–4136. https://doi.org/10.1109/ISGT-Asia.2019.8881778.

43. Al Amrani Y, Lazaar M, El Kadiri KE. Random forest and support vector machine based hybrid approach to sentiment analysis. Procedia Computer Science. 2018; 127:511–20. https://doi.org/10.1016/j.procs.2018.01.150.