ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal
Regular Issue, Vol. 9 N. 2 (2020), 49-68
eISSN: 2255-2863
DOI: https://dx.doi.org/10.14201/ADCAIJ2020924968

Influence of Pre-processing Strategies on the Performance of ML Classifiers Exploiting TF-IDF and BOW Features

Amit Purushottam Pimpalkar and R. Jeberson Retna Raj

School of Computing, Sathyabama Institute of Science and Technology, Chennai, Tamilnadu, India
amit.pimpalkar@gmail.com

KEYWORD

Pre-Processing; Sentiment Analysis; BOW; TF-IDF; Evaluations Metrics; Twitter Classifier; Twitter

ABSTRACT

Data analytics and its associated applications have recently become important fields of study. The subject of concern for researchers now-a-days is a massive amount of data produced every minute and second as people constantly sharing thoughts, opinions about things that are associated with them. Social media info, however, is still unstructured, disseminated and hard to handle and need to be developed a strong foundation so that they can be utilized as valuable information on a particular topic. Processing such unstructured data in this area in terms of noise, co-relevance, emoticons, folksonomies and slangs is really quite challenging and therefore requires proper data pre-processing before getting the right sentiments. In this study we tried to study the impact of nine different pre-processing methods. Bag of Words (BOW), Term Frequency (TF) and Inverse Document Frequency (IDF) features were selected and influence of pre-processing strategies were measured on the performance of ML classifiers with two real life dataset; one with domain and other not.

We have evaluated five different Machine Learning (ML) algorithms viz Logistic Regression (LR), Decision Trees (DT), Multinomial Naive Bayes (MNB), Extreme Gradient Boosting (XGBoost), and Support Vector Machines (SVM). We have performed a comparative analysis of the success of these algorithms in order to decide which algorithm works best for the given dataset in terms of recall, accuracy, F1-score, precision, and Receiver Operator Characteristic (ROC). It is demonstrated that SVM classifier outperformed the other classifiers with superior evaluations of 73.12% and 94.91% for accuracy and precision respectively. It is highlighted in this research that the selection and representation of features along with various pre-processing techniques have a positive impact on the performance of the classification. The ultimate outcome indicates an improvement in sentiment classification and we noted that pre-processing approaches obviously suggest an improvement in the efficiency of the classifiers.

1. Introduction

Opinion Mining (OM) continues with the implementation of technology that can distinguish positive and negative expressions that are part of conversations and remarks. Because of the availability of massive volumes of data accessible on the internet, numerous companies have begun to take an interest in this because mining this knowledge may be of greater use to them. Twitter is a popular social networking platform where users exchange a few words known as "tweets". With only a 140 character-limit which ranges around 12 to 15 words on an average; around 3,60,000 tweets were tweeted per minute[1]. This acts as a way for people to share their opinions or emotions on various topics. Online sales have an enormous development, where consumers purchase a product and leave a message on their experience.

Such tweets include misspellings, slang's, and informal phrases along with language, an entity's viewpoint and symbolic words that contribute to the various data collection and interpretation problems. The vocabulary used by people of most social platforms is very informal. Consumers develop their own terms and spelling shortcuts, punctuation, misspellings, phrases, foreign words, URLs, gender-specific jargon and abbreviations. Therefore, this kind of text needs to be changed. The pre-processing approach supposed to be used to create the correct sensation for successful decision-making. This is applied in order to minimize the unstructured aspect of the data collected from social media. However, it can take a lot of time to adapt this approach to larger datasets. Pre-processing of data is a key phase in Sentiment Analysis (SA), because choosing the necessary pre-processing methods will courteously increase the accurately categorized instances in addition to the original feature space. SA plays a significant part here to help in seeking the secret emotion or perception within a message. Classifying a tweet into either a positive or a negative term is a common method for humans, but this manual practice is not adequate to manage massive volumes of data on the internet. "What other people think??" (Pang *et al.*, 2008) is often affected people's decisions on each and every subject. Since a massive volume of data is available on the internet, numerous companies have begun to take an interest in this, because mining this knowledge can be of tremendous use to them from a future perspective. This gives rise to a radically new and wide field of research known as SA. ML approaches are developed to overcome these issues. If the text contains more positive terms, a positive polarity is identified, and vice versa with the help of ML algorithms.

The major contributions and novelties to this article are (1) Conducting a detailed and systematic series of SA tests utilizing nine pre-processing methods utilizing five well known ML algorithms on training and testing sets; (2) Conduct a comprehensive analysis of the success between two datasets spread over 54,000 unstructured tweets. (3) Presenting a general considerable assumption on the consequence of conducting a broad variety of pre-processing methods, this leads to the enhancement of SA. (4) Presenting detailed observations that show pre-processing combinations substantially boost the classification outcomes for test datasets are used. (5) Compare two different types of features vectors (BOW, TF-IDF) and its effect on SA. (6) Compare the impact of the unclean text and cleaned text. (7) Assess the computational speed and accuracy of the MNB, LR, DT, XGBoost, and SVM classifiers to analyze the best performance.

The purpose of this research is to observe the effectiveness of pre-processing methodologies in classification problems. Assessing a classifier for its precision is really critical as it can't be used in real life activities without understanding the usefulness of a classifier in prediction. The research often presents the performance of one method but if that approach not contrasted with any other approach and tool, or if performance measurement approaches differ between studies ventures, it is extremely challenging to decide the strategies which are better in the given circumstances. The work contributes to the research by suggesting the best-fit algorithm among the algorithms selected to analyze the twitter data for SA.

This remaining paper is prepared as follows. The Section 2: Literature Review highlights few of the latest research concentrating in particular on pre-processing strategies for SA to find the research gaps of the existing systems. In Section 3: Design of the Proposed Work outlines the datasets used in our experiment, details about each pre-processing approach, features selection and extraction techniques used in research, and supervised machine learning methods include some basic examples of the relevant field. In Section 4: Experimental Evaluation and Discussion presents experimental outcomes and discusses a wider spectrum of results. In Section 5: Conclusion and Future Scope summarizes the findings and key assumptions of this research and proposes a variety of potential studies proposals.

2. Literature Review

Much research is currently underway in the fields of text analytics and pre-processing of text. During previous assessments, various common methods were used to clean the text only; thus, no effective approaches were employed. Several scholars focused on literary studies on the issue of classifying sentiments. Some databases are freely available online, and several researchers have already started using them to analyze the findings. As our research focuses on the general positive or negative opinion expressed in the investigation, we have focused our literature sample toward the classification of sentence level sentiment.

HaCohen-Kerner *et al.*, (2020) evaluate the effects of all possible permutations of six basic pre-processing techniques for spelling rectification, transferring capital letter into lower case, disposal of HTML tags, deletion of punctuation marks, abolishment of stop-words and curtailment of repeated characters on four benchmark text corporas using three NB, SMO (SVM variant) and RF ML methods using BOW unigrams and training and test sets. A study involving several basic text pre-processing methods on KNN, DT, RF, LR, Stochastic Gradients Descent (SGD), NB, SVM classifiers with NLTK to get classification accuracy for all types of pre-processing steps was discussed by Işik *et al.*, (2020). The efficiency of NB and SVM classifiers was evaluated to demonstrate their efficacy in Twitter data sentiment mining under different experimental setups. The Stanford Testing Sentiment data set (STS) was used by (Ismail *et al.*, 2016) for the purposes of study. (Hasan *et al.*, 2018) provided the evaluation of political views by applying supervised ML algorithms such as NB and SVM. Multiple traditional ML techniques were compared to identify documents level polarity and suggested LR has provided the best precision (Kamath *et al.*, 2018). NB, Maximum Entropy and SVM Classifier and performance efficiency were measured by (Pujari *et al.*, 2017) and revealed that SVM worked better on Product review dataset of Amazon.

Bao, *et al.*, (2014) conducts a research study on the Stanford Twitter Sentiment Dataset to test the performance of a number of major features and emotions pre-processing methods. The author's result shows that the URL features, negation transition and repetitious text message standardization have a positive impact on the accuracy of classification, while stemming and lemmatization have a negative impact. Taking into consideration and to improve the impact of the SVM classifier on the sentiment classification task Singh, *et al.*, (2016) carried out experiments by applying the standardization scheme to the text of the tweets and ignoring their sentiment class. After the standardization process, they find the opinion of unspecified (slang) terms. The findings of their studies strongly show that the predicted scheme is not only resilient to data size, but also works well in terms of detection of sentiment accuracy.

A comparison of three separate forms of textual data pre-processing techniques; Stemming, Lemmatization and Spelling Correction and their impact on SA using SVM, NB and DL algorithms was observed by Pradha, *et al.*, (2019). Their review indicates that the SVM has done well. In addition, the evidence in their study suggests that the consequences of choosing the correct text data pre-processing on the opinion will facilitate rapid and accurate decision-making system. Different pre-processing techniques commonly employed on Twitter data sets were compared with (Effrosynidis *et al.*, 2017). For through pre-processing technique authors employ three separate ML algorithms and disclose the classification accuracy and the resulting number of features. They consider that strategies such as stemming, deleting numbers and eliminating elongated terms boost the precision, while such as deleting punctuation do not improve the accuracy of the system. Study of (Nivaashini *et al.*, 2018) presented the findings of the text pre-processing system for efficient extraction and selection of features, and then classifies tweets as positive, negative and neutral by means of ML techniques. The J48 algorithm works better in Twitter Sentimental Analysis (TSA) relative to other ML algorithms. An NLP based pre-processed data outline was developed by (Hasan *et al.*, 2019) to filter tweets where authors incorporated BoW model and TF-IDF model concept to SA using Twitter API. Using a massive dataset of a million tweets submitted to the StockTwits site to test the performance of a broad variety of pre-processing approaches and ML algorithms for financial SA suggested by (Renault *et al.*, 2019). The authors considered that adding bigrams and emojis significantly enhances efficiency in the classification of feelings. They showed that the pre-processing approach and sample scale have a significant effect on the connection between market confidence and returns on stocks.

Krouska, *et al.*, (2016) investigated with a sequence of pre-processing methods applied to three separate datasets, one without a particular domain and the other with similar subjects, and tested the output of four well-known classifiers. The findings demonstrate that with correct collection and description of characteristics, the accuracy of SA can be increased. In specific, unigram and 1-to-3-grams do more effectively than other depictions and the extraction feature increase the precision of the classification. The role of different pre-processing techniques, including the deletion of stop - words, stemming, and word vector on the accuracy of SA's on three ML algorithms via NB, ME and SVM was associated by Alam, *et al.*, (2018). Researchers determined the accuracy of the algorithms pre and post pre - processing stage. An outcome demonstrates that considerable improvement in the performance of the NB algorithm after the pre-processing steps. A minor difference in SVM algorithm accuracy was observed and odd, no change in accuracy was noted in ME after implementing the pre-processing steps. (Othman *et al.*, 2019) proposed learning sentiment related continuous representations of words as features for classification of Twitter feelings under a supervised learning approach. Compare the success of our method with the top three scoring teams who participated in the SemEval Sentiment Subtask-A classification challenge using RNN-LSTM on the same dataset. Three signal detection algorithms for volume tweets, sentiment tweets and top hashtags was advocated by (Nazir *et al.*, 2018). The algorithms used as the moving algorithm of the average threshold, the Gaussian algorithm and the hybrid algorithm. These algorithms were evaluated on data collected from Twitter in real time and demonstrate that the hybrid algorithm outperforms the other algorithms.

A method which combines TF-IDF and LDA schemes to measure the value of each research paper and combine K-means clustering algorithm of the same papers with related subjects was proposed by (Kim *et al.*, 2019). Experimental findings showed that the proposed method could identify research papers with related subjects by the key-terms which were extracted from paper abstracts. A two-stage approach to data analytics consisting of ML algorithms and combinatorial fusion was proposed by (Ho *et al.*, 2019). The first stage uses five ML algorithms for combinatorial fusion and then merges these algorithms with their subsets. Using a Kaggle data collection, the authors investigated to label each of the tweets as positive, negative or neutral sentiment. A classification algorithm focused on supervised ML techniques and word-based N-gram processing to automatically divide Twitter messages into credible and not credible ones introduced by (Hassan *et al.*, 2020). Five different supervised ML classification techniques were applied and the research examines two interpretations of features (TF and TF-IDF) and separate sets of N-gram terms. 10-fold cross validation performed on two datasets in English and Arabic languages for model training and testing. The best performance was achieved by combining both unigrams and bigrams, LSVM as a classifier and TF-IDF as a technique for extraction of features. Researchers explored the usefulness of deep CNN and RNN-based neural network model using embedded word vector in SA with different tuning of the hyper-parameters.

While considerable work has been conducted in this area and numerous pre-processing strategies exist in the literature, selecting the ideal pre-processing approach is still an open research problem. The issue of the short sentiment analysis of emotions continues a difficulty, with no systematic answer. Studies demonstrate that the right pre-processing approaches differ based on the task. Therefore it is requires to have a mechanism which will suggest the better pre-processing techniques exploiting different features on ML Classifiers with different and bigger dataset. Furthermore it is need of the hour which will suggest a procedure that requires fewer processing time with more precision, reliable, effective classifiers, and a quicker answer on bigger datasets.

3. Design of the Proposed Work

Figure 1 shows the data flow through different modules in the proposed work for Twitter Sentiment Analysis. Design of the proposed work is explained in following sub-sections.

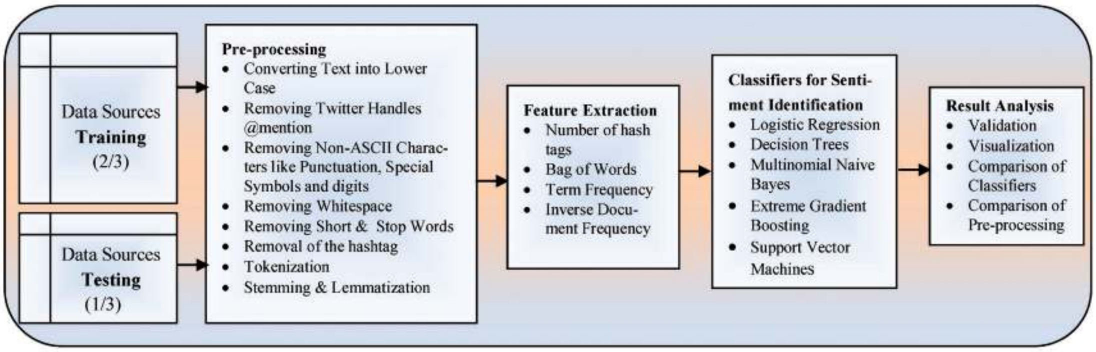
Data Sources
Training
(2/3)
Data Sources
Testing
(1/3)
Pre-processing
- Converting Text into Lower Case
- Removing Twitter Handles @mention
- Removing Non-ASCII Characters like Punctuation, Special Symbols and digits
- Removing Whitespace
- Removing Short & Stop Words
- Removal of the hashtag
- Tokenization
- Stemming & Lemmatization
Feature Extraction
- Number of hash tags
- Bag of Words
- Term Frequency
- Inverse Document Frequency
Classifiers for Sentiment Identification
- Logistic Regression
- Decision Trees
- Multinomial Naive Bayes
- Extreme Gradient Boosting
- Support Vector Machines
Result Analysis
- Validation
- Visualization
- Comparison of Classifiers
- Comparison of Pre-processing

Figure 1: Architectural Data Flow of Twitter Sentiment Analysis of Proposed Work

3.1. Dataset

We used the dataset that has been crawled and labeled positive and negative by Kaggle. Our data set is split into two sections-the datasets for testing (1/3) and training (2/3). This study uses two datasets which include reviews of Twitter generalized twits without any domain and Internet Movie Database (IMDb) with movie domain. The earlier dataset-1 has 49159 reviews (45467 positive and 3692 negative) and later dataset-2 contains 4845 reviews (3743 positive and 1102 negative). This research intends to generate a predictive model which can make the distinction between positive and negative reviews on various evaluation parameters which are widely used in predicting the efficiency of an algorithm wiz Accuracy, F1 Score, Recall, Precision and ROC.

3.2. Pre-Processing

This work focuses on discovering feelings for twitter data, because it is more difficult due to its unstructured composition, small scale, slangs use, misspells abbreviations, etc. Furthermore, as the tweets are very raw in nature, this work utilizes different pre-processing measures in order to gain valuable input data in the classification. The pre-processing step of text data is one of the influential methods for cleaning up and rendering using such unstructured data, mainly to keep revives noise-free such as unwanted characters, symbols, etc. at the classification phase. This is due to the reviews; expressions may identify two or three features that would be challenging to distinguish whether the sentence has related positivity or negativity. We have taken into account that, the reviews scraped from the twitter website contain some spelling errors and that will be a serious problem. We pre-processed the data with the help of a Regular Expression module that is built in feature of python.

Table 1: Narrative of the various Pre-processing techniques applied in the study

SNo	Technique Name	Narrative
1	Converting Text into Lower Case	The primary purpose of lowering the all the text to lowercases is to keep the words “Stunning Movie!” and “stunning movie!”from being interpreted as different phrases since they are the same. It reduces the amount of words that the dictionary needs to hold on a moment.
2	Removing Twitter Handles @mention	Users on Twitter used to mention usernames of other users in their tweets. We can clearly see in our literature study that the Twitter handles contribute nothing significant to solving our problem. So it’s better to take them out in our dataset.
3	Removing Non-ASCII Characters like Punctuation, Special Symbols and digits	Characters other than alphabets, numbers and special symbols have to be removed, because errors can be created during processing. Digits should even be omitted, as it has little to do with tweet sensation. We’ll replace everything here with spaces, except the characters and hashtags.
4	Removing Whitespace	Whitespace did not offer much significant importance to the text, so it is omitted for computational considerations.
5	Removing Short and Stop Words	Stop words in any language, e.g. refer to common words such as “are”, “am”, “is”, “we”, etc. and short words like “hmm”, “and”, “oh”, etc are of very little use. Since tweet polarity isn’t dependent on those words, so they can also be removed. There, we have to be a little cautious about selecting the length of the terms we choose to delete. Here we selected to delete all the words that are 3 or fewer in extent.

SNo	Technique Name	Narrative
6	Removal of the hashtag	To achieve so, the hashtag is omitted from the text and placed in a different section. We use the weight of the hashtag in a feature extraction process.
7	Tokenization	We'll now tokenize all of the filtered tweets in our dataset. Tokens are independent phrases or words and the process of splitting a string of text into tokens is tokenization or word segmentation or lexical analysis. Tokenization is a step dividing the longer text string into smaller pieces or tokens. Further processing is generally performed after appropriate tokenization of a piece of text.
8	Stemming	Stemming is the method of eliminating the affixes (suffixed, prefixes, infixes, circumcises) from a phrase to get a completely new word. For example, converting the word amazing/amazed to amaze its pure word. Both stemming and lemmatization is targeted at eliminating inflexive forms and often variant connected types of a term to a specific type of basis. This decreases overall words and promotes processing efficiency.
9	Lemmatization	Lemmatization originally referred towards doing things properly using a word vocabulary and morphological analysis, essentially meant merely to eliminate the extensions of inflection and restore the root or dictionary version of a word recognized as lemma.

3.3. Features Selection and Extraction

Extraction of features is to mine characteristics from datasets consisting of formats such as text and image, in a format supported by ML algorithms. There are various types of features that can be used for an analysis of sentiments. In building the model for this research following features were introduced and implemented for the study.

- Terms presence and frequency: It relates to the amount of occurrence the terms take place in the texts. It either gives a binary weighting to the words, or it uses term frequency weights so that a term often occurs in a document. Since each text is unique in length, a term is likely to appear even more frequently in lengthy documents than in smaller ones. Therefore, the word frequency was separated by the size of the document.

$$\mathrm{TF(t)} = \frac{\text{(The amount of occasions the term t is seen in the document)}}{\text{(Total number of terms in the document)}} \tag{1}$$

- Document Frequency and Term Frequency-Inverse Document Frequency (TF-IDF): This weight is a quantitative metric that is used to determine how important a term in a list or corpus is to a document. The importance gradually increases with the number of times that an increase proportionally to the number, but is offset by the word frequency in the corpus. Frequency of documents is the number of times features appear in all texts. The TF-IDF algorithm is used in any content to weigh a keyword and attribute importance to that keyword based on the number of times it appears in the document. Once the Document Frequency value of each feature is calculated, appropriate features are selected through the threshold. (Kshirsagar *et al.*, 2020; Mestry *et al.*, 2019; Gao *et al.*, 2019; Kermani *et al.*, 2019; Othman *et al.*, 2019; Sidorov *et al.*, 2019; Maryam *et al.*, 2018; Das *et al.*, 2018; Yamout *et al.*, 2018; White *et al.*, 2018; Alsmadi *et al.*, 2018; Gu *et al.*, 2018; Emelyanov *et al.*, 2017; Chen *et al.*, 2017; Krouska *et al.*, 2016).

$$\mathrm{IDF(t)} = \log \frac{(\text{Total number of documents})}{(\text{Number of documents with term t in it})} \tag{2}$$

$$\mathrm{TFIDF(t)} = \mathrm{TF(t)} * \mathrm{IDF\ (t)} \tag{3}$$

Throughout this research article, TF-IDF is retrieved by technique of vector space, and positive or negative text states are added with the longest and most common words in the dataset and the frequency of each term is suggested.

- Bag of Words: Feature Selection is either categorized as lexicon or statistical. The feature selection techniques treat the documents either as a group of words or BOW, or as a string that holds word sequence in the document is considered as statistical strategies. Due to its simplicity BOW is used for the classification process. Stop-word elimination and stemming are the most basic selection measures for the application. The BOW model is associated only with the issue of whether recognized words occur in the document, or not. The intuition is that they'll have relevant ones, and then documents are similar. Further, we can learn much about the document's significance from the information by itself. The aim is to turn every free text document into a vector that we can use as input or output for a model of ML (HaCohen-Kerner *et al.*, 2020; Kshirsagar *et al.*, 2020; Dhanjal *et al.*, 2019; Bilgin *et al.*, 2019; Gao *et al.*, 2019; Othman *et al.*, 2019; Das *et al.*, 2018; White *et al.*, 2018; Alsmadi *et al.*, 2018).

  Since we understand that, if there are 26 words in the vocabulary, we might use 26 fixed-length document representations, with a single vector position to score every other word. The simplest coding scheme is to mark a Boolean value for the presence of words, present =1, and absent = 0. This is illustrated as follows with an example, considering we have 4 documents as;

  D1: I am so happy and surprised by this movie!

  D2: This movie was so frustrating.

  D3: What an absolutely stunning movie!

  D4: This movie took me by surprise.

  Here we need to develop a vocabulary utilizing identical terms from all the texts by ignoring case and punctuation, since they don't produce the necessary knowledge for the model as ['happy', 'surprise', 'movie', 'frustrate', 'absolute', 'stun', 'took']

  Here, D=4, N=7

  The matrix M of size 4 X 7 will be represented as:

Table 2: Example of document containing term frequencies of word

	happy	surprise	movie	frustrate	absolute	stun	took
D1	1	1	1	0	0	0	0
D2	0	0	1	1	0	0	0
D3	0	0	1	0	1	1	0
D4	0	1	1	0	0	0	1

Table 2 above shows the training features in each document which contain term frequencies of each word as we used for our illustration.

3.4. Classifiers for Sentiment Identification

This section focuses on the theoretical context used in this research by the conventional ML algorithms.

Logistic Regression: It is a supervised classifier of machine learning, which extracts real-evaluated input attributes, multiplies by weight, adds sums and passes the sum to generate a probability via the *sigmoid* function. Thresholds are used for decision making. LR is one of the most useful analytical algorithms, which has the ability to evaluate the importance of individual features in a transparent way. The LR hypothesis tends to restrict the cost function between 0 and 1. Thus, linear functions fail to support it as it can have a value greater than 1 or less than 0 which is not possible according to the LR hypothesis. The true method of LR is binary or binomial LR in which the target or dependent variable can only have 2 possible types, either 1 or 0. This helps one to model a relationship between several response variable and a target variable binary / binomial. In the case of LR, the linear method is essentially used in the following relationship as a reference to another method, such as *g*

$$h_{\emptyset}(x) = g(\emptyset^{T}\, x) \tag{4}$$

where $0 \leq h_{\emptyset} \leq 1$

In this case *g* is the logistic or sigmoid function that can be given as follows

$$g(z) = \frac{1}{1 + e^{-z}} \tag{5}$$

where $z = \emptyset^{T}x$

- Decision Trees: It is a supervised classifier model that uses data to form the DT with known labels, and then the model is applied to the test data. Where each tree node represents a test on a data set attribute, and its children represent the results. The leaf nodes represent the end data point groups. The best testing condition or decision has to be made for every node in the tree. For a given node t, where p(j|t) is the relative frequency of class j at node t.

$$GINI(t) = 1 - \sum_{j} [p(j|t)]^{2} \tag{6}$$

- Multinomial Naive Bayes: MNB is a simplified variant of the Naive Bayes algorithm and is ideally adapted to the classification of text documents. While simple NB model text as the addition and exclusion of different words, MNB explicitly model the word counts and adjust the fundamental formulas to be used. It can be described and computed as

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c) \tag{7}$$

where

$P(t_k|c)$ is the conditional probability of the term t_k occurring in a document of the class c.

P(c) is the prior probability of a document occurring in class c.

- XGBoost: XGBoost is an optimized library to improve the distributed gradient algorithm that generates a predictive model that would be a series of low predictive decision trees. Compared to other ML algorithm, modeling is very simple, and training is fairly easy. In specific, it can be parallelized to the degree that it enables the training of larger models through several processors. This can accommodate fragmented data that fails with other algorithms and, in a wide variety of situations, provides well-proven predictions. We use the range of K models by incorporating their outputs in the following way.

  $$\hat{y}_i = \sum_{K=1}^{K} f_k(x_i), \; f_k \in F \qquad (8)$$

  where

  F is the space of trees

  x_i is the input

  ŷi is the final output.

  We attempt to minimize the following loss function

  $$L(\Phi) = \sum_{i} l(\hat{y}_i, y_i) + \sum_i \Omega\,(f_k) \qquad (9)$$

  where $\Omega\,(f) = \Upsilon T + \frac{1}{2}\lambda\,\|w\|^2$

  Υ and λ are the hyper parameter

  T is number of tree node

  W is the vector of nodes

- Support Vector Machines**:** The SVM approach is basically an analysis of the multidimensional space of various groups in a hyperplane. It's a non-probabilistic binary linear classifier. SVM can produce the hyperplane iteratively to reduce the error. The goal of SVM is to segment the datasets into groups in order to reach the Maximum Marginal Hyperplane (MMH). SVM utilizes a kernel trick strategy in which the kernel takes a low-dimensional input space and converts it into a larger-dimensional space. In simple words, by adding more dimensions to it the kernel converts non-separable problems into separable problems. Support vectors are predictor variables relatively close to the hyper-plane, and affect the hyper-plane's position and orientation. Using such vectors for help we optimize the classifier's range. Deleting the support vectors will alter the hyper-plane position. These are the points which will help us to build our SVM. SVM classifiers have excellent precision, and function well with high dimensional space. Basically, SVM classifiers use a subset of training points, and thus consume far less memory in the end.

  The hyper-plane equation is as follow

  $$w^T x + b = 0 \qquad (10)$$

  Where

  w is the weight vector

  b is the bias

Maximizing-Margin represents Minimizing Loss so we are trying to optimize the distance between the data points and the hyper-plane. The loss function which helps to optimize the margin is loss of the hinge.

$$L(w) = \sum_{i=1} \max(0, 1 - y_i[w^T x_i + b]) + \lambda\|w\|_2^2 \qquad (11)$$

where

$\max(0, 1 - y_i[w^T x_i + b])$is a Loss Function
$\max + \lambda\|w\|_2^2$ is a Regularization Factor

4. Experimental Evaluation and Discussion

In this section, evaluations obtained on various parameters by all the algorithms for both datasets are discussed. We converse the execution of the presented algorithms in previous section. As explained earlier all processing steps, data is separated to raw and processed data. This segment addresses the output improvement and the comparative study of the five ML algorithms. A Word cloud (or Tag cloud) is a snapshot of text corpus and that can be seen for positive and negative words of dataset-1 and dataset-2 in Figure 2 and Figure 3 respectively. It displays a list of words with varying font size or color. This format is useful to get the most prominent terms perceived quickly. These words are displayed in a chart from a given corpus, with the most important words being written with larger, bold fonts, while less important words are displayed with smaller, thinner fonts or not displayed at all.

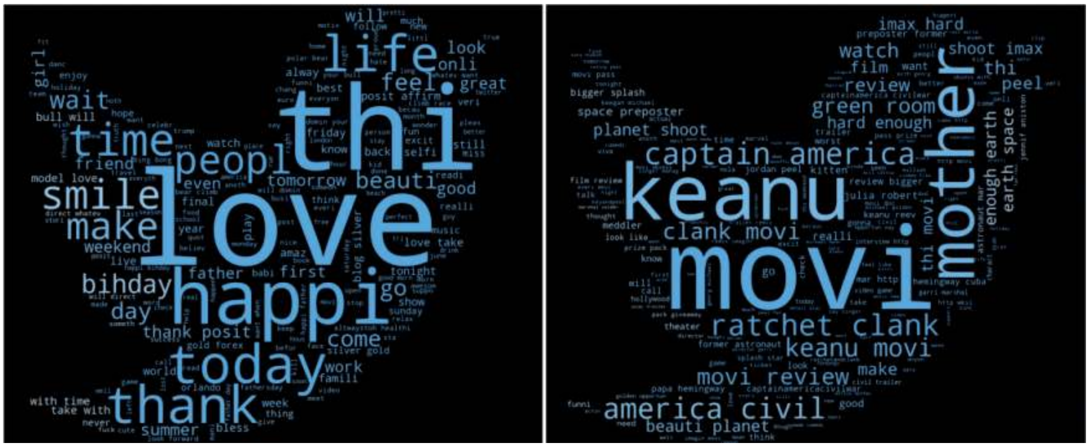

Figure 2: Resulting word cloud for the Positive words for dataset 1 and dataset 2.

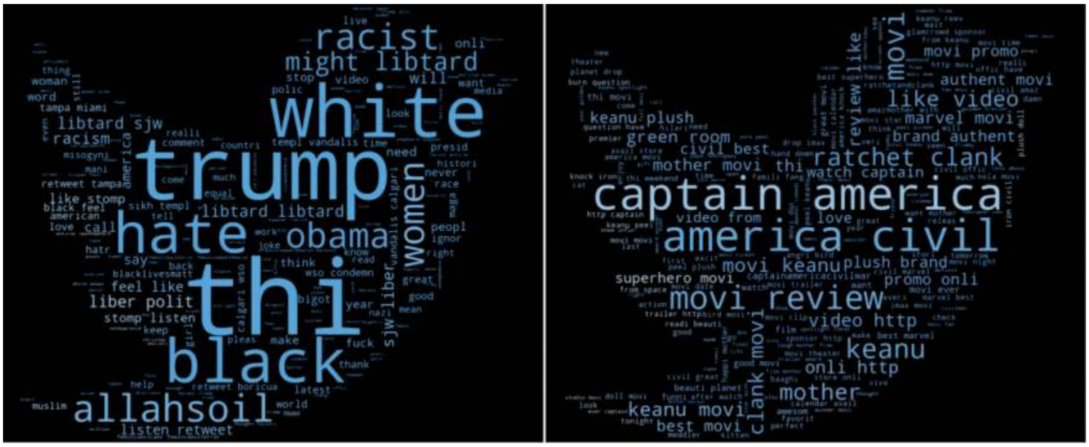

Figure 3: Resulting word cloud for the Negative words for dataset 1 and dataset 2

The best evaluation classifier result has been displayed in Table 5 and Table 6 with bold typeset. From the empirical evidences as seen from Tables 5 and 6, it has been found SVM classifier is most suitable classifier for the twitter SA. From Table 5, we conclude that for generalized Twitter Dataset-1 we have obtained the SVM classifier highest accuracy with 94.91% whereas the same classifier resulted 73.12% if evaluation criteria selected as precision for BOW feature. It has been found that for Dataset-2 of IMDb movies again SVM classifier gives the better evaluations with 99.33% and 78.23% for Accuracy and precision respectively despite the fact that accuracy of DT classifier produces slight better results with 83.87% for precision. With table 6 and TF-IDF as feature then linear regression classifier attained the highest accuracy with 94.99% and other classifiers gives the accuracy more than 94.02%. At the same time by considering the precision as an evaluation criterion SVM predicts the best grades with 67.46%. It was noted that for Dataset-2 of IMDb movies SVM classifier outperformed the other classifiers with better evaluations of 81.94% and 82.02% for Accuracy and precision respectively. The graphical representation for the Table 5 and 6 can be seen in Figure 4.

Table 3: Performance comparison with different classification methods on the datasets for BOW feature Without Preprocessing

Parameter	**LR**		**DT**		**MNB**		**XGBoost**		**SVM**	
BOW	Dataset-1	Dataset-2	Dataset-1	Dataset-2	Dataset-1	Dataset-2	Dataset-1	Dataset-2	Dataset-1	Dataset-2
Recall	49.04	97.59	50.81	85.31	**59.49**	91.32	51.25	95.06	43.59	**99.46**
Accuracy	94.46	79.66	89.38	76.16	91.77	78.84	94.38	**80.28**	**94.57**	79.05
F1	55.63	88.12	40.39	84.69	50.59	86.96	**56.40**	**88.17**	53.23	88.00
Precision	64.28	80.32	33.52	**84.07**	44.00	83.00	62.70	82.21	**68.36**	78.91

Table 4: Performance comparison with different classification methods on the datasets for TF-IDF feature Without Preprocessing

Parameter	LR		DT		MNB		XGBoost		SVM	
BOW	Dataset-1	Dataset-2	Dataset-1	Dataset-2	Dataset-1	Dataset-2	Dataset-1	Dataset-2	Dataset-1	Dataset-2
Recall	51.69	98.90	48.76	88.61	46.29	**99.35**	**54.16**	95.19	47.37	98.83
Accuracy	**95.08**	80.08	93.35	77.70	94.85	80.18	94.57	80.09	95.00	**81.21**
F1	58.72	88.86	49.80	86.38	**59.79**	88.85	57.44	88.36	56.17	**89.31**
Precision	67.95	79.95	50.88	**83.94**	67.41	80.35	61.14	82.45	**68.98**	81.47

Table 5: Performance comparison with different classification methods on the datasets for BOW feature With Preprocessing

Parameter	LR		DT		MNB		XGBoost		SVM	
BOW	Dataset-1	Dataset-2	Dataset-1	Dataset-2	Dataset-1	Dataset-2	Dataset-1	Dataset-2	Dataset-1	Dataset-2
Recall	51.10	98.39	60.08	90.25	**61.56**	92.52	54.34	96.26	44.47	**99.33**
Accuracy	94.58	78.94	91.95	79.05	92.75	77.50	94.45	**79.56**	**94.91**	78.12
F1	57.21	87.84	51.41	86.94	54.6	86.40	**58.11**	**87.92**	55.31	87.53
Precision	64.98	79.33	44.93	**83.87**	49.06	81.05	62.43	80.92	**73.12**	78.23

Table 6: Performance comparison with different classification methods on the datasets for TF-IDF feature With Preprocessing

Parameter	LR		DT		MNB		XGBoost		SVM	
BOW	Dataset-1	Dataset-2	Dataset-1	Dataset-2	Dataset-1	Dataset-2	Dataset-1	Dataset-2	Dataset-1	Dataset-2
Recall	52.46	**100**	54.01	89.61	47.83	99.61	**55.86**	94.28	47.99	98.96
Accuracy	**94.99**	79.97	94.02	78.53	94.73	80.28	94.51	79.97	94.92	**81.94**
F1	**58.62**	88.81	54.98	86.90	55.11	88.92	57.92	88.21	56.08	**89.69**
Precision	66.40	79.87	56.00	**84.35**	64.98	80.31	60.13	82.87	**67.46**	82.02

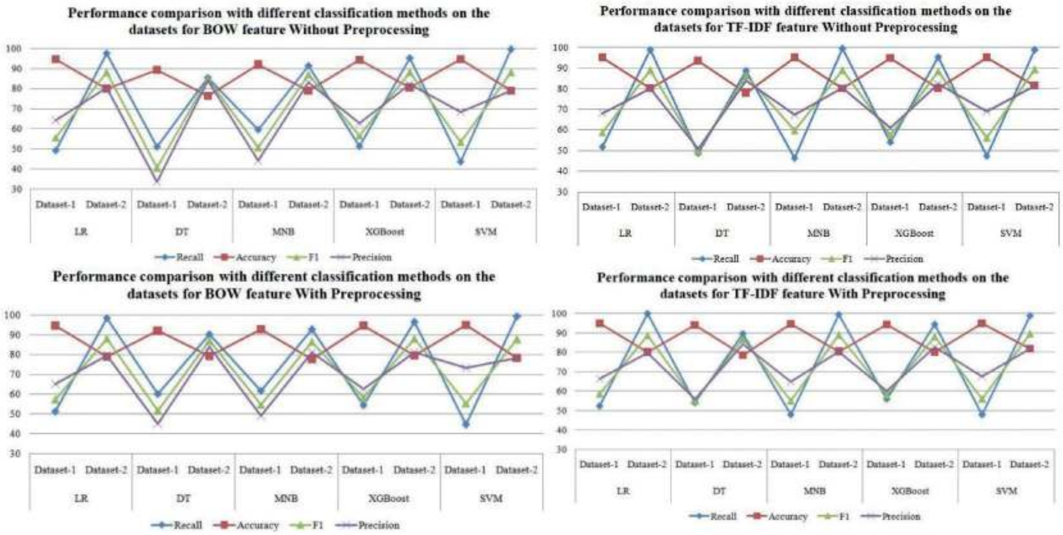

Figure 4: Performance of different models with respect to feature selected and dataset

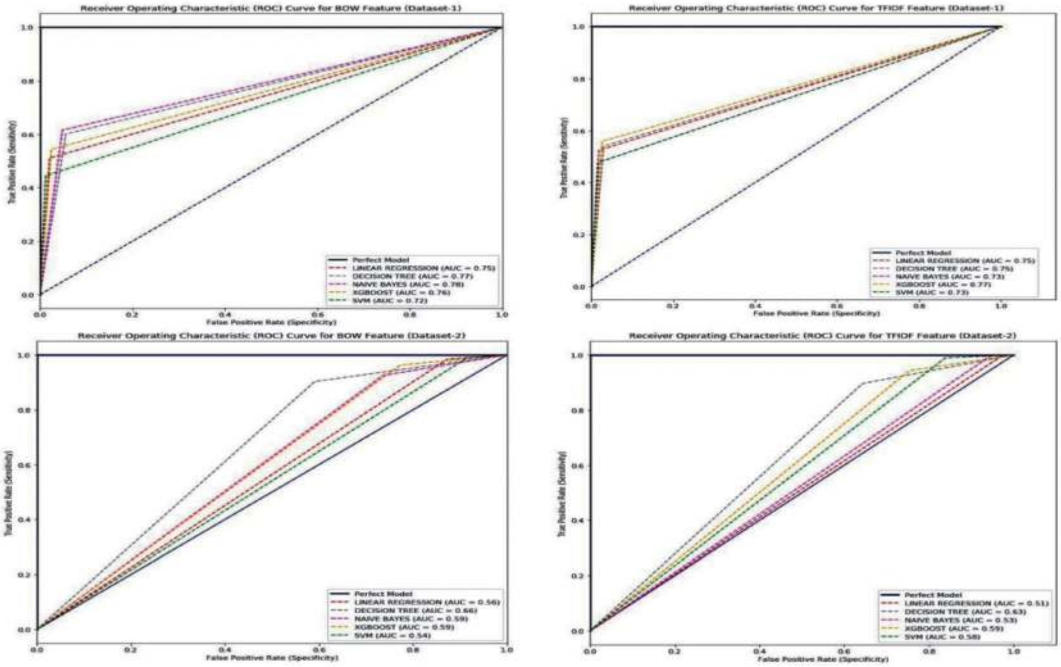

Figure 5: ROC Performance of different models with respect to feature selected and dataset.

One of the evaluation metric for problems of binary classification is the ROC curve which maps the True Positive Rate (TPR) against the False Positive Rate (FPR) at specific threshold values and effectively distinguishes the ‘signal’ from the ‘noise’. The Area Under the Curve (AUC) is an indicator of the classifier’s capacity to differentiate between two classes and is seen as a representation of the ROC curve. The higher the AUC, the stronger the overall accuracy of the diagnostic test is to accurately distinguish data. As seen in figure 5, we conclude that the dataset 1 is much better as compared to dataset 2 and gives the accuracy in high 80’s, whereas dataset 2 resulted into high 60’s only, for the binary classification by setting optimal threshold value. Figure 6 represents the time performance of the each algorithm on pre-processed dataset with both features utilized in this work.

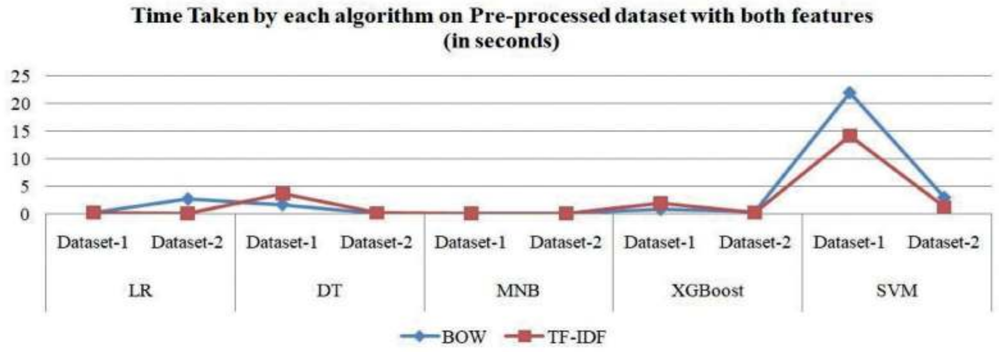

Figure 6: Time Taken by each algorithm on Pre-processed dataset with both features (in seconds)

Table 7 Classifiers’ F1-Score before and after preprocessing for BOW feature with Dataset-1

Algorithm	Accuracy before Pre-processing (%)	Accuracy after Pre-processing (%)	Net Improvement (%)
LR	55.63	57.21	1.58
DT	40.39	51.41	11.02
MNB	50.59	54.6	4.01
XGBoost	56.40	58.11	1.71
SVM	53.23	55.31	2.08

Table 8 Classifiers’ F1-Score before and after preprocessing for TF-IDF feature with Dataset-1

Algorithm	Accuracy before Pre-processing (%)	Accuracy after Pre-processing (%)	Net Improvement (%)
LR	58.72	58.62	-0.10
DT	49.80	54.98	5.18
MNB	59.79	55.11	-4.68
XGBoost	57.44	57.92	0.48
SVM	56.17	56.08	-0.09

Table 9 Classifiers' F1-Score before and after preprocessing for BOW feature with Dataset-2

Algorithm	Accuracy before Pre-processing (%)	Accuracy after Pre-processing (%)	Net Improvement (%)
LR	87.84	88.12	0.28
DT	86.94	84.69	-2.25
MNB	86.40	86.96	0.56
XGBoost	87.92	88.17	0.25
SVM	87.53	88.00	0.47

Table 10 Classifiers' F1-Score before and after preprocessing for TF-IDF feature with Dataset-2

Algorithm	Accuracy before Pre-processing (%)	Accuracy after Pre-processing (%)	Net Improvement (%)
LR	88.81	88.86	0.05
DT	86.90	86.38	-0.52
MNB	88.92	88.85	-0.07
XGBoost	88.21	88.36	0.15
SVM	89.69	89.31	-0.38

From Table 3, 4, 7, 8, 9 and 10 our experimental research clearly demonstrates that text pre-processing significantly impacts the accuracy of ML algorithms. Furthermore, it concludes that, in the some particular situation of LR, DT MNB, and SVM algorithms, accuracy has considerably boosted following the formulation of the text pre-processing steps. In certain instances, it is frequently observed that the pre-processing of the dataset increases the net improvement of more than 11%. We infer that in some situations, where such pre-processing approach has not been put into operation the efficiency of the classifiers declines with a very limited margin. This result expects a slight spike as well as a decrease in the accuracy of classification on the classifiers. Interestingly, no major improvement in F1-Score was seen in the XGBoost algorithm. Tests have shown that the performance of the SVM algorithm was greatly increased after pre-processing. The average time taken by LR, DT MNB, and XGBoost algorithms on the pre-processed data around 2 seconds only on an average and at the same time SVM zooms around 17 seconds on both the features selected for evaluation. We genuinely think that each pre-processing step; impacts the accuracy of a ML algorithm in its own way on assorted features. Proper data pre-processing technique plays a crucial role in the classifier's predictive performance when using unstructured data. Finally, it is demonstrated that the choosing and representation of features can have a positive impact on the performance of the classification.

5. Conclusion and Future Scope

By carrying out this research, we analyze the productivity of each algorithm, LR, DT, MNB, XG-Boost and SVM in conjunction with the assessment metrics selected as, recall, accuracy, precision, F1 score and ROC; and decide the impact on classification efficiency on two data sets with nine pre-processing techniques to the data obtained from Twitter. In view of the above, pre-processing the data is a significant step in SA, which maximises the amount of correctly defined instances. A synthesis of TF-IDF and BOW features are included for implementation. Research implications indicate that after pre-processing of the dataset the F1-score of the classifier is substantially improved on BOW feature, whereas the effect is negligible for the TF-IDF.

A distinction is drawn from the results obtained between the models and the SVM classification model has shown a higher accuracy and precision than the other classifiers, with superior evaluations of 73.12% and 94.91% respectively to classify the Twitter content. The work underlines the positive impact on the efficiency of the classification by means of collection and depiction of features along with different pre-processing techniques. The ultimate findings suggest a rise in sentiment classification and we have found that pre-processing methods naturally cause an increase in classifiers' performance.

In comparison, the case where the tweet includes just an image, links or references are omitted. In order to determine the optimum setting, it is also worth investigating further the possible exploratory solutions not contained in this analysis. With respect to attribute methodology, a more in-depth analysis that concentrate on choosing the best classification model to evaluate important features or to evaluate ranking methods such as Information-gain, Chi-square, etc. POS tags, topic-based SA, as well as a variety of techniques to choose the features to boost classifications. Sarcastic messages contain optimistic words or even enhanced positive words to express a critical view or the other way around. In addition, potential work will include integrated approaches that conduct both feature acquisition and model refinement at around the same stage. Additionally, as the computational power is increasing day by day as the growth of data on the web, we may employ the deep learning methods for better results. We think that creating further use of the larger dataset of customer reviews available through the internet will increase this application's scope and usability. In the future, the program will suggest could pave the way for users to suggest a suitable service or product based on the research that we do in this method.

6. References

Alam, S., and Yao, N. (2018). The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis, Computational and Mathematical Organization Theory. doi:10.1007/s10588-018-9266-8.

Alsmadi, I. and Hoon, GK., (2018). Term weighting scheme for short-text classification: Twitter corpuses. Neural Computing and Applications. doi:10.1007/s00521-017-3298-8.

Bao, Y., Quan, C., Wang, L., and Ren, F. (2014). The Role of Pre-processing in Twitter Sentiment Analysis, Lecture Notes in Computer Science, 615–624. doi:10.1007/978-3-319-09339-0_62.

Bilgin, M., and Köktaş, H., (2019). Sentiment Analysis with Term Weighting and Word Vectors, The International Arab Journal of Information Technology, Vol. 16, No. 5, pp 953-959.

Chatzakou, D., and Vakali, A., (2015). Harvesting Opinions and Emotions from Social Media Textual Resources, IEEE Internet Computing, pp 46-50.

Chen, J., Chen C., and Liang, Y., (2016). Optimized TF-IDF Algorithm with the Adaptive Weight of Position of Word, Advances in Intelligent Systems Research, volume 13, pp 114-117. doi: 10.2991/aiie-16.2016.28.

Das, B., and Chakraborty, S., (2018). An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation. arXiv: 1806.06407.

Dhanjal, K., and Sangeeta, (2019). Applying Machine Learning Algorithms for News Articles Categorization: Using SVM and kNN with TF-IDF Approach, Smart Computational Strategies: Theoretical and Practical Aspects, pp 95–105. doi:10.1007/978-981-13-6295-8_9.

Effrosynidis, D., Symeonidis, S., and Arampatzis, A., (2017). A Comparison of Pre-processing Techniques for Twitter Sentiment Analysis, Lecture Notes in Computer Science, 394–406. doi:10.1007/978-3-319-67008-9_31.

Emelyanov, GM., Mikhailov, DV., and Kozlov, AP., (2017). The TF-IDF measure and analysis of links between words within N-grams in the formation of knowledge units for open tests, Pattern Recognition and Image Analysis. 27, 825–831. doi:10.1134/S1054661817040058.

Gao, W., Peng, M., Wang, H., Zhang, Y., Xie Q., and Tian, G., (2019). Incorporating word embeddings into topic modeling of short text, Knowledge and Information Systems 61, 1123–1145. doi:10.1007/s10115-018-1314-7.

Gu, Y., Wang, Y., Huan, J., Sun, Y., and Jia, W., (2018). An Improved TFIDF Algorithm Based on Dual Parallel Adaptive Computing Model, In I*EEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. doi:10.1109/cybermatics_2018.2018.00133.

HaCohen-Kerner, Y., Miller, D., and Yigal, Y. (2020). The influence of preprocessing on text classification using a bag-of-words representation, PLOS ONE, 15(5), e0232525. doi:10.1371/journal.pone.0232525.

Hasan, A., Moin, S., Karim, A. and Shamshirband, S. (2018). Machine Learning-Based Sentiment Analysis for Twitter Accounts Mathematical and Computational Applications, 23(1), 11. doi:10.3390/mca23010011.

Hasan, MR., Maliha, M. and Arifuzzaman, M., (2019). Sentiment Analysis with NLP on Twitter Data, International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2). doi:10.1109/ic4me247184.2019.9036670.

Hassan, N., Gomaa, W., Khoriba, G. and Haggag, M., (2020). Credibility Detection in Twitter Using Word N-gram Analysis and Supervised Machine Learning Techniques, International Journal of Intelligent Engineering and Systems, Vol.13, No.1. doi: 10.22266/ijies2020.0229.27.

Ho, J., Ondusko, D., Roy, B. and Hsu, DF., (2019). Sentiment Analysis on Tweets Using Machine Learning and Combinatorial Fusion, IEEE International Conference on Dependable, Autonomic and Secure Computing, In *International Conference on Pervasive Intelligence and Computing, International Conference on Cloud and Big Data Computing*. doi:10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00191.

Işik, M, Dağ, H. (2020). The impact of text preprocessing on the prediction of review ratings, Turkish Journal of Electrical Engineering and Computer Science, 28 (3), 1405-1421. doi: 10.3906/elk-1907-46.

Ismail, H., Harous, S. and Belkhouche, B., (2016). A Comparative Analysis of Machine Learning Classifiers for Twitter Sentiment Analysis, In *International Conference on Intelligent Text Processing and Computational Linguistics – CICLing*.

Kamath, CN., Bukhari, SS., and Dengel, A., (2018). Comparative Study between Traditional Machine Learning and Deep Learning Approaches for Text Classification, Proceedings of the ACM Symposium on Document Engineering- DocEng. doi:10.1145/3209280.3209526.

Kermani, ZF., Sadeghi, F., and Eslami, E., (2019). Solving the twitter sentiment analysis problem based on a machine learning-based approach, Evolutionary Intelligence. doi:10.1007/s12065-019-00301-x.

Kim, SW., and Gil, JM., (2019). Research paper classification systems based on TF-IDF and LDA schemes, Human-Centric Computing and Information Sciences, 9(1). doi:10.1186/s13673-019-0192-7.

Krouska, A., Troussas, C., and Virvou, M. (2016). The effect of preprocessing techniques on Twitter sentiment analysis, In *7th International Conference on Information, Intelligence, Systems & Applications (IISA)*. doi:10.1109/iisa.2016.7785373.

Kshirsagar, V., (2020). Detecting Hate tweets — Twitter Sentiment Analysis, https://towardsdatascience.com/detecting-hate-tweets-twitter-sentiment-analysis-780d8a82d4f6, (ONLINE last accessed on 06/06/2020).

Maryam, A., and Ali, R. (2018). Temporal TF-IDF-Based Twitter Event Summarization Incorporating Keyword Importance, Smart Innovation, Systems and Technologies, pp 559–566. doi:10.1007/978-981-13-1747-7_54.

Mestry, S., Singh, H., Chauhan, R., Bisht, V., and Tiwari, K., (2019). Automation in Social Networking Comments With the Help of Robust fastText and CNN, In *1st International Conference on Innovations in Information and Communication Technology (ICIICT)*. doi:10.1109/iciict1.2019.8741503.

Mrabti, S. El., Achhab, M. Al., and Lazaar, M., (2018). Comparison of Feature Selection Methods for Sentiment Analysis, Big Data, Cloud and Applications, pp 261–272. doi:10.1007/978-3-319-96292-4_21.

Nazir, F., Ghazanfar, MA., Maqsood, M., Aadil, F., Rho, S. and Mehmood, I., (2018). Social media signal detection using tweets volume, hashtag, and sentiment analysis, Multimedia Tools and Applications. doi:10.1007/s11042-018-6437-z.

Nivaashini, M., Soundariya, RS. and Thangaraj, P., (2018). Comparative Analysis of Machine Learning Approaches for Twitter Sentiment Analysis, Journal of Computational and Theoretical Nanoscience, 15(5), pp 1743–1749. doi:10.1166/jctn.2018.7371.

Othman, R., Abdelsadek, Y., Chelghoum, K., Kacem, I. and Faiz, R., (2019). Improving Sentiment Analysis in Twitter Using Sentiment Specific Word Embeddings, In *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*. doi:10.1109/idaacs.2019.8924403.

Pang, B., Lee, L., (2008). Opinion mining and sentiment analysis. Foundation Trends Information Retrieval 2(1–2), pp 1–135.

Pradha, S., Halgamuge, M. N., and Tran Quoc Vinh, N. (2019). Effective Text Data Preprocessing Technique for Sentiment Analysis in Social Media Data. In *11th International Conference on Knowledge and Systems Engineering (KSE)*. doi:10.1109/kse.2019.8919368

Pujari, C., Aiswarya, and Shetty, NP., (2017). Comparison of Classification Techniques for Feature Oriented Sentiment Analysis of Product Review Data, Data Engineering and Intelligent Computing, pp. 149–158. doi:10.1007/978-981-10-3223-3_14.

Renault, T., (2019). Sentiment analysis and machine learning in finance: a comparison of methods and models on one million messages. Digital Finance. doi:10.1007/s42521-019-00014-x.

Sidorov, G., (2019). Vector Space Model for Texts and the tf-idf Measure, In *Syntactic n-grams in Computational Linguistics. Springer Briefs in Computer Science*, pp 11–15. doi:10.1007/978-3-030-14771-6_3.

Singh, T., and Kumari, M. (2016). Role of Text Pre-processing in Twitter Sentiment Analysis, Procedia Computer Science, 89, 549–554. doi:10.1016/j.procs.2016.06.095

White, HD., (2018). Bag of works retrieval: TF*IDF weighting of works co-cited with a seed, International Journal of Digital Library 19, pp 139–149, 2018. doi: 10.1007/s00799-017-0217-7.

Yamout, F. and Lakkis, R., (2018). Improved TFIDF weighting techniques in document Retrieval, In *Thirteenth International Conference on Digital Information Management (ICDIM)*. doi:10.1109/icdim.2018.8847156.

7. Conflict of Interest

Authors declare no conflicts of interest.