Received: 21 January 2020 | Revised: 18 August 2020 | Accepted: 20 August 2020

DOI: 10.1002/eng2.12280

RESEARCH ARTICLE

Engineering Reports Open Access WILEY

Sentiment classification of skewed shoppers' reviews using machine learning techniques, examining the textual features

Mahdi Rezapour

Wyoming Technology Transfer Center, Laramie, Wyoming, USA

Correspondence
Mahdi Rezapour, Wyoming Technology Transfer Center, 1000 E. University Avenue, Dept. 3295, Laramie, WY 82071.
Email: rezapour2088@yahoo.com

Abstract

With the speedy growth of online shopping, it has become of crucial importance for product makers to analyze, and handle a wealth of products' reviews. However, such a high volume of reviews, along with a wide variety of opinions, makes it hard for manufacturers to know exactly how they can improve their products without having an efficient approach. For this purpose, the results of sentiment classification would help the customers to retrieve the necessary information to choose an appropriate product, and the sellers to effectively collect customer feedback in order to improve their products. Like most of the read-world problems, the shopping review data being used in this study were imbalanced, being predominately composed of positive with only a small percentage of negative reviews. Machine learning (ML) algorithms do not perform well when data are imbalanced, as they tend to get biased toward the overrepresented data category. The synthetic minority over-sampling technique (SMOTE) was used to address this class imbalance problem. In this study, three different ML-based algorithms, namely the Naïve Bayes (NB), Support Vector Machine, and decision tree (DT) were employed. An extensive preprocessing procedure was taken to prepare the text datasets, and details are discussed in the manuscript. The performance analysis indicated that the DT algorithm outperforms the other two methods. As positive reviews account for the majority of the reviews, sparse words removal for the data resulted in the removal of almost all negative reviews' sentiments. Hence, the model training process is here performed on positive and negative reviews separately. A combination of the review titles with their contents, separate tokenization process, applications of various N-gram, and maintaining stops words (e.g. "not" or "but") were some other steps considered to improve the performance of the model.

KEYWORDS

machine learning technique, opinion mining, polarity/opinion extraction, review classification, sentiment analysis, text classification|Natural language processing

1 | INTRODUCTION

Sentiment analysis is defined as analyzing people opinions about a product, or organization with the goal of products improvement. To ensure future products enhancement, the predictive performance of extracted sentiments needs to be evaluated. That is especially important for sentiments extraction as the reviewes need to be distinguished first based on the polarity of the sentiments, and then further analyses could be conducted to extract the content of those reviews for possible product enhancement. Sentiment classification aims at mining customers' reviews being written for certain products or services, and classifying those reviews into positive or negative.[1] Machine learning techniques, as means of sentiment classification, could be implemented to make a meaningful information/conclusion from shoppers' review.[2]

Sentiment analysis could be conducted in different levels including document level, sentence level, aspect-based, or comparative analyses.[3] Document level is a simplest form of sentiment, which assumes each document contains an opinion on one main object. A most common case of this approach is when there are only two classes for the sentiment category: positive and negative. The trained model, then, would be used to tag new documents into their new sentiment classes.

Sentence sentiment analysis also could be used when a more fine-grained view is expressed in the documents. The two mentioned approaches, document and sentence sentiment analyses, work when either the whole document or sentence talk about a single entity. When reviewers talk about different aspects or parts of a product, aspect-based method could be applied. Words such as "less," "more," "most," "least," or "-er" could help in analyzing the reviews.[4]

The sentiments analysis about a product review is unstructured in nature, filled with stop words, punctuations, and infrequent words so before conducting any machine learning analysis, the reviews need to be preprocessed. Often the negative review account for a small proportion of the whole datasets, which result in an imbalanced dataset. Beside the incapability of most machine learning techniques of identifying the under-sampled category, conducting sparsity analysis on the whole dataset at once might result in extraction of only keywords of overrepresented category, which are mostly related to positive reviews, while keywords of underrepresented category reviews would be discarded. In order to address these issues, in this study the analysis was conducted on the two sentiments, positive versus negative, separately, and then the two datasets were aggregated. After data preparation, synthetic minority over-sampling technique (SMOTE) was conducted on aggregating dataset to produce a balanced data. The SMOTE conducted as machine learning techniques works in favor of overrepresented category.

1.1 | Problem statement

It is important to discuss the challenges of working with the review datasets which require especial attention before reporting any result.

- Usually the review documents are skewed in favor of one type of review, positive vs negative. This would result in propensity of the machine learning algorithm to work in favor of an overrepresented category.
- Reviews have been characterized by emotional shoppers' comments. Although emotional characteristics would make the classification process more challenging, identification and extraction of the related emotions are necessary for a company/sellers to be able to monitor their customers' perceptions about a product. The customers' perception about a product would help to identify the main problem of a product.
- While leaving a review, the customers are required to assign a title to their reviews. The reviewers would leave the title due to a possible extreme emotion of excitement or frustration to have their voices to be heard. Those titles are not irrelevant and would present extreme amount of information just in few words.
- Identifying all the necessary elements of review is important to understand the complete context of the entire piece. Sometime the important words could be removed during preprocessing steps so especial attention is needed to keep those important terms.
- It is important to separate the reviews of an object, e.g. product or service, or attributes, e.g. component of a product, size, weight, to have a comprehensive presentation of the customers believes. It should be noted, while sentiment classification would be plausible for a combined product reviews of TV or jeans, for instance, the sentiment extraction would be very challenging. That is due to a use of different terminology for various products

1.2 | Study contribution

The main contributions of this paper are as follows:

1. Due to low number of negative reviews, tokenization was conducted on positive and negative reviews separately so important negative features would be extracted.
2. Due to having imbalanced dataset, the SMOTE technique was implemented after tokenization to create balanced response categories.
3. To improve the performance of prediction, titles of the reviews were also added to their contents.
4. To enhance the performance of the predicted model, different combinations of N-gram were used and evaluated, and it was found that a combination of unigram and bigram models would result in a better performance.
5. Ambiguity is another issue that degrades the accuracy. For instance, people have commonly used "I wanted to love but". So especial attention has been made to address this issue by keeping some stopwords in the dataset.

2 | RELATED WORK

Mainly two types of methods have been utilized for text mining purpose to identify whether a review is positive or negative. Those methods include semantic orientation and machine learning techniques. A phase has a negative semantic when it has a bad association such as "very awful," or a positive semantic when it has a positive semantic orientation such as "awesome product."

An unsupervised learning algorithm was presented for classifying a review into positive or negative.[5] Identification of positive or negative review would be based on semantic orientation such as "subtle," or "very." In this method, reviews would be classified based on average of semantic orientation of a given phrase.

Machine learning techniques have also been used extensively for evaluation of document level sentiment classification. Text classification (TC) is defined as using an algorithm to classify a set of text documents into different categories. The following studies would go over some of the studies used text document for sentiment analysis.

A study conducted to analyze the topics of online reviews for two competitive products using unsupervised method namely latent Dirichlet allocation (LDA) method.[6] The results of the LDA analysis highlighted the strength and weakness of the two products, providing valuable managerial implication.

In another study, the helpfulness and economic impact of product reviews was evaluated.[7] The results indicated, for instance, that negative reviews are negatively associated with sale product. Random forest machine learning technique was used to evaluate the impact for the reviews on sales. The study highlighted the importance of review evaluation on sell revenue.

The effectiveness of review sentiment on readership and helpfulness on online review was evaluated.[8] The results indicated that reviews with higher levels of positive sentiment in the title receive more readership while length and longevity of review positively influence the leadership and helpfulness.

Although text preprocessing steps are mainly common across the studies, a detailed step could vary based on the context of a study. For instance, in a past study, several steps were taken for text analysis: first documents were categorized into different categories using k-nearest neighbors algorithm (KNN), Naïve Bayes (NB), and term-graph, and then the most relevant documents were retrieved.[8,9] The results indicated that the KNN shows the maximum accuracy compared with the other methods.

Sentiment classification has been conducted often with the objective of extracting text customers/reviews, and classifying them into positive or negative opinions based on the polarity of those reviews.[10] A study conducted to evaluate whether a review is positive or negative using movie review as a dataset.[11] Three machine learning techniques were used in that study including NB, Maximum entropy classification, and Support vector machine (SVM). The total accuracy of unigrams and unigrams were presented and compared. The results indicated a combination of unigram and bigram outperformed the prediction.

In another study, methods such as Naive Bayes were used for analyzing the review dataset.[12] The main objective of the study was to determine the negative, positive, or neutral polarity distributions. In another study,[13] evaluation of a product reviews beyond like/dislike distinction was evaluated by examining its textual formatting features. For instance, the study considered a usage of capital letters, and repeated words.

The discussed review of the literature highlighted the research gap that could be bridged. Negative comments are of crucial importance for the product makers to improve the quality of the products. So beside using SMOT, and conducting

tokenization on positive and negative comments separately, negation words such as "not" in "will not," for instance, were excluded from stopwords to have a better estimation of prediction of negative comments. A combination of n-grams was taken advantages of, instead of using bi-gram or tri-grams alone to have a better prediction accuracy. Consider "this product is not good", for instance: ignoring "not" would end up in a positive review which would result in a possible worse performance of the model. In summary, while the above studies just identified positive or negative reviews, it is also important to extract the review textual content to help the product to be enhanced by their makers.

The SVM, and Naïve Bayes are among most widely used method of machine learning techniques in the natural language processing (NLP).[1,14,15] However due to simplicity and comparison, decision tree method has been added to the methods. A comparison across the methods would be made and presented in a future subsection.

3 | DATASET

In this study two product review from women shopping related to jeans and pants were used. The dataset was obtained from Kaggle, and the reviews belong to some products sold in Amazon, an E-commerce website. The data consist of 2167 (85%) positively labeled reviews, and 368 (15%) negative review. The polarity of these review is decided based on whether the shoppers indicated she would recommend the product to friends or not. If in a review's comment she indicated that she would recommend the product in the future to a friend, the review would be considered as positive, otherwise a review would be considered as negative.

During review process, the reviewer were required to have a title for their reviews. Those titles include important information summarizing the content of the related review. Thus, that column in the dataset titled "title of review", were aggregated with the content of the related reviews to make most of the available information. Other variables/columns such as age of review and the numbers of star related to the product were excluded from the dataset.

The process of data preparation could be divided into two stages: first step involves classifying positive and negative review separately to identify the most dominant keywords. This step is primarily conducted because a proportion of negative comments is significantly lower than positive ones. Primarily, the data preparation was conducted on the dataset as a whole, but due to the majority of review being positive, it resulted in removal of all the negative important keywords in the sparsity screening.

Second, after identifying the keywords for positive and negative review separately, the n-grams words of both reviews were aggregated to prepare the full dataset in the second stage. Randomly 80% of the observation were extracted and used as training dataset, while the other 20% were used for a test dataset. After the model was trained, to evaluate the quality of a trained model, the trained model was applied on test dataset.

4 | DATA PREPROCESSING

Preprocessing method plays an important role in text mining due to the nature of language being used in the reviews. The preprocessing could leave out the irrelevant information from the review while keeping relevant contents. Few main preprocessing steps taken in this study could be categorized, for instance, into stopwords removal, stemming, and tokenization. The flow chart of the steps taken in this research is presented in Figure 1. The overviews of the steps taken in this study would be presented in the following paragraphs.

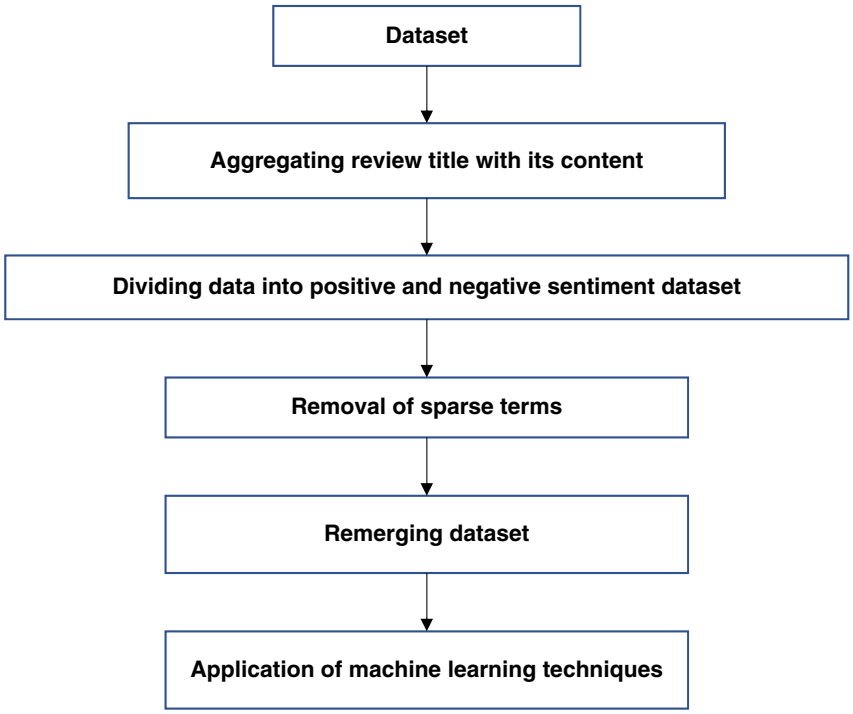
4.1 | Aggregating review title with review content

Due to the importance of review title, and the fact that it gives a summary of reviews sentiments, before starting any analysis, the content of a text of this cell is aggregated with contents of related review.

4.2 | Removal of stopwords

The reasons why stop words need to be removed from the review content is due the importance of removing less important words from the analysis as the irrelevant words significantly reduce the dimensionality of the term space. These words do not contain important significance to be used in text evaluation. These words often include more than 150 words including pronoun, and preposition such as "in," "an," "with," "i," "not," "but," or "we." However as "not" or "but"

FIGURE 1 Preprocessing steps



work as a negation in a sentence, these words are excluded from the list of the stopwords to have a better performance for negative reviews. This is because "no" for instance, could be used to negate important words such as "like" in "not like."

4.3 | Removing unhelpful common words

Besides removal of the stop words, some other words that were found not to add information to the final data were removed. These words are mostly related to the context of the considered review dataset. These words were identified at the final stage of data cleaning. These words included some words such as "dress," "xxs," "made," "hope," "pants," "bought," and "buy." For instance, both positive and negative reviews, almost in every review, use the words of "pants" or "jeans" so the inclusion of those words would not help in enhancing the model performance.

4.4 | Uppercase identification

This step is an important step of preprocessing as there are differences between lower and upper case in NLP. Moreover, sometimes the shoppers express their emotions by capital letter (eg, "LOVE", "HATE"), which would result in an identification of similar words as two words, which reduces the efficiency of the model. Thus, all the words in a sentence turn into a lower case to have entire words in consistent states.

4.5 | Punctuation identification

There are punctuations like commas, question marks, apostrophes, and more. The punctuation can also be used to express emotion or express highlighted feelings in few sentences such as "god!!!!". However, if these punctuations remain in a sentence, they would be treated as words and it would result in accuracy reduction of the analysis.

4.6 | Word stemming

Word stemming is an important step in text mining preprocessing. This step is conducted without compromising on the document's precision. The main objective of this step is to retrieve a root word by stemming from different words forms such as adjective, adverb, or verb.

In this study, Porter's stemming algorithm was used for stemming the documents.[16] For instance, words like "introduction," "introducing," "introduces" would map back into "introduce." This is because mostly those words would bring the similar contextual meaning to the review text and having all of these words would result in a possible identification of all these words, or removal of them due to their sparsity. However, it should be noted that there are some associated errors with the stemming step. For instance, over-stemming and under stemming could happen. The example for over-stemming would be "university," "universal," "universe" which would stem into "universe." Having said that, as this issue would mostly occur for a non-review document, the over/under-stemming is not applicable for this study.

4.7 | Removing sparse terms

The words that were found to be merely present in the whole documents due to various reasons such as some unique emotions or set of personal words need to be removed to achieve a higher accuracy. In other words, while using a bag of words, ignoring the terms that occur less frequent could help to prevent the model from overfitting and higher error rate. Sparsity was set only in the first step of data preparation over positive and negative review to have about 20 n-gram words. Then these words were aggregated in the next step. In the analysis conducted in this study, sparse equal to -min of document frequency, or minimum values of a feature documents frequency were set as 95%.

4.8 | N-gram

N-gram is a sequence of *n* items from a given sample of text document. N-gram of size 1, or unigram, refers to a single word such as "good" or "bad." At the final stage it was found that many of these words were negated by "not," which would not be captured by unigram. Thus, only bigram of size 2, and trigram or size 3 were considered in this study. It should be noted while trigram phrases were considered in the model, only one phrase, "size_fit_perfect" was identified and included in the dataset. After conducting the above process, the datasets would be ready for a further processing. Some of the identified sentiments are included in Table 1.

4.9 | SMOTE

A data are called to be skewed if sample from one category is significantly higher in number than the other class.[17] Learning from skewed dataset would result in produced bias for classifiers: having a much higher predictive accuracy over the majority classes with a poor prediction on the minority class. These types of datasets have been often called imbalanced data. For this scenario, most of the available algorithm focus on classification of major sample while ignoring/misclassifying minority sample.[18] Imbalanced data have been reported for text classification extensively.

TABLE 1 Top n-gram features

Negative		Positive	
Sentence feeding into machine learning algorithm	**Complete sentence (example)**	**Sentence feeding into machine learning algorithm**	**Complete sentence (example)**
Want-love	I wanted to love these pants, but they were larger than i expected	Love_love	Love love love!
Run-small	for starters, these pants run small	Fit_true	they fit true to size
Not-worth	but these are not worth even the sale price	Perfect_fit	they were a perfect fit for me
Not-flatter	Pants are not flattering	High_recommend	i highly recommend!

Data-preprocessing sampling would be applied on dataset by either adding new samples to existing samples (over-sampling), or removing existing sample (under-sampling). It is known that under-sampling would discard very useful information, while over sampling would result in overfitting problems.

SMOTE can be used for the construction of classifiers from imbalanced database.[19] This method generates synthesis minority samples to oversample the minority class. SMOTE is an over-sampling approach in which the minority class is over-sampled by creating "synthetic" examples rather than by over-sampling with replacement. This method is useful as it prevents overfitting and it forces the region of minority to be more general. The way this method works is by taking each minority class sample, and introducing synthetic examples along the line segments joining any of the k minority class nearest neighbors. In simple words, SMOTE draws lines between existing minority instances, and then would create synthetic minority instances somewhere on those lines.

However, it should be noted that there are advantages and disadvantages to SMOTE. The advantage is that SMOTE creates synthetic rather than oversampling replacements, which would prevent overfitting problem. A synthetic example would be created along the line segments joining the k minority class nearest neighbors. As the created samples are not exact copies of original data, this technique could be used in machine learning techniques without having a bias. However, it should be noted as SMOTE does not take into consideration that examples could be from other classes, it could result in the overlapping the classes and consequently additional noise and increasing confusion matrix error rate.

4.10 | Feature selection: random forest recursive feature elimination (RF-RFE)

This method was conducted before applying a main algorithm, and after the SMOTE application on the dataset. The objective of this method is helping classifier to reach optimal performance by selecting most relevant/helpful features. Feature reduction is necessary for any analysis especially before conducting a machine learning technique.

Random forest is made of several decision trees. The trees would be grown based on answering specific questions. Gini loss function could be used to measure the model impurity. Gini index is a measure of how often a randomly chosen element from the dataset would be incorrectly labeled. Feature selection can be described as a number of iterations, in which feature importance would be measured and the less relevant ones would be removed. The recursion is needed as the importance of each feature would be changed dramatically over different subset of features during stepwise feature elimination.[20]

Different strings could be used to clarify how a model should be selected based on various summary metrics. Common metrics are root mean square error for regression problem, and the Kappa (accuracy) for classification problem. As this study is dealing with classification, the Kappa was used for selection of an optimal model.

Kappa is specifically helpful for imbalanced dataset. This value would be calculated as follows:

$$k = \frac{p_0 + p_e}{1 - p_e} \tag{1}$$

where p_0 is accuracy, and p_e is the hypothetical probability of chance agreement, and p_e could be written as:

$$p_e = \frac{1}{N^2} \sum_k n_{k1} n_{k2} \tag{2}$$

where N is the number of observations and n_{ki} is the number of times a classifier i predicted category k.

5 | MACHINE LEARNING ALGORITHMS

The input for the analysis would be corpus, which could be converted from csv format. As discussed above, the documents need to be preprocessed by methods such as stemming, tokenization, and entity extraction. After conducting n-grams, the bag of words was used to break words into individual word counts variables. The machine learning algorithms have been used in all areas of day-to-day life, from anomaly detection,[21] and crash prediction,[22] to sentiment analysis of the investors.[23] The below sections discuss the machine learning techniques have been used for training and evaluation of model accuracy. It should be noted that all the algorithms were conducted in R.

5.1 | Naïve Bayes

NB classifier is a family of probabilistic classifier which is based on application of Bayes theorem with independence assumption between feature being referred to naïve. Based on probability, the independence of features means that the impact of an individual feature on response is working independently from other features on selection of response. Naive Bays follows Bayes' theorem as follows:

$$P(y \mid X) = \frac{P(X \mid y)P(y)}{P(X)} \tag{3}$$

where y is response and X are independent features as $X = (x_1, x_2, x_3, x_4, \ldots, x_n)$.

$P(X \mid y)$ indicates a probability of having, for instance, a positive review given a review contain "fit-well."

Now implementing independence of features, we will have:

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1 \mid y)(x_2 \mid y). \ldots (x_n \mid y)}{P(x_1)P(x_2) \ldots P(x_n)} \tag{4}$$

The above could be written as:

$$p(y \mid x_1 + \ldots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i \mid y)}{P(x_1)P(x_1) \ldots P(x_n)} \tag{5}$$

For the above equation, as the denominator is constant for a given input, it would be removed so we would have:

$$p(y \mid x_1 + \ldots, x_n) \rightarrow P(y)\prod_{i=1}^{n} P(x_i \mid y.) \tag{6}$$

Now classifier needs to be created to find the probability of given set of input with the objective of maximizing probability for output as follows:

$$\hat{y} = \text{argmax}_y P(y)\prod_{i=1}^{n}(x_i \mid y) \tag{7}$$

Combining Naïve Bayes classifier with a decision rule, the maximum posterior would pick the hypothesis that is most probable; the Bayes classifier can be written as follows:

$$\hat{y} = \underset{k\epsilon\{1,000,K\}}{\text{argmax}}\, P(C_k)\prod_{i=1}^{n} p(x_i \mid C_k) \tag{8}$$

where class label $y = C_k$, where k is a binary classification, and argmax is arguments of maxima, in which the $\hat{y}$ would be maximized based on specific i.

Beside the assumption of independence of different predictors' effects on response, there are other assumptions that need to be considered. Other assumptions are the order of words being not important and the weight/importance of each word is equal to all other words. Although those assumption do not hold true for most real world problems, this method perform on text classification very well.[24] Specifically, this is a popular method for text categorization, which is based on word frequencies as features. For instance, each of the terms "Run-small" and "Not-worth" are associated with a probability, and a decision would be computed by the sum of the total of the probability of different words for a review.

5.2 | Decision tree

In a simple word, questions would be asked in a most efficient ways partitioning the dataset, leading to an answer. The questions would be chosen to provide a best split: the questions would be stopped when there are less observations at a node than a designed value, or when all the observations belonging to the same class. There are different parts to decision

[Correction added on 18 August 2021, after first online publication: Equations 3, 4 and 5 have been corrected in this version.]

tree including root node (the population/sample), and splitting (dividing a node into two or more sub-nodes). When a node cannot be divided into a further node it would be called terminal or leaf node.

When an optimization conducts on a tree by removing sub-nodes, the process is called pruning. As discussed, a most common cost function that can be used for pruning or growing a tree is Gini index. It shows how pure a region is by evaluating how much of the training data in a region belongs to a single class. This function gives an idea how mix the classes are for a binary group. A perfect classification would result in a Gini score of 0, while a worst-case scenario result in split of 50/50.

Gini index value (G) would be written as below:

$$G = \sum_{c=1}^{C} \hat{\pi}\,(1 - \hat{\pi}\,) \tag{9}$$

where $\hat{\pi}$ defines how much of the training data in a region belongs to a unique class

5.3 | SVM

SVM has been used as an algorithm which can use data for classification or regression analysis. For classification, the objective of SVM is to decide which class belongs to which data point. The dataset can be viewed as p-dimensional vector, which could be separated by $p-1$ hyperplanes. The vectors from the same distance from the hyperplane is named support vectors. Like any machine learning techniques, main parts of SVM consists of constrains, cost, and optimization functions. The objective function is to find hyperplanes that could create a largest separation between the data. In other words, the hyperplane would be selected to maximize the distance from hyperplane to the nearest datapoints.

Hinge function or a loss function is a function that could be used to maximize the objective function. Data are not linearly separable in a soft-margin hyperplane compared with hard-margin. When the two classes of the response are not linearly separable due to some outliers, the hyperplane condition would be relaxed by adding some slack ξ

$$y_i(x_i^T w + b) \geq 1 + \xi_i{}^k \;\; y_i \in \{-1, 1\} \text{ and } i = 1, \ldots, m \tag{10}$$

where w is a weight of a vector (weight vector), which initialize at 0, and b is a bias.

The distance between the two vectors (margin) is $2/w$, thus this value needs to be maximized with some constraints as follows:

$$\text{Objective function} = \text{maximizing } \frac{2}{w}(\text{distance between two hyperplanes}) \tag{11}$$

$$\text{Constrains}: \begin{cases} f(x) = wx_i + b \geq 1 \text{if } y_i = +1 \\ f(x) = wx_i + b \leq 1 \text{ if } y_i = -1 \end{cases} \tag{12}$$

where $2/w$ is a margin, distance between the two hyperplane is w or whole margin, and $wx_i + b$ is the equation for hyperplane, there are two hyperplane in Equation (12), and b is a bias. Anything above or on the boundary line of the $wx_i + b \geq 1$ would be considered as one class, and anything on or below the line of $wx_i + b \leq 1$ would be considered as another class. The Both lines in Equation (12) would be drawn based on a closest point in the negative and positive areas. It should be noted that values of $+1$ and -1 are chosen for maximizing the separation. If x_i is misclassified then $w \to w + \alpha \text{sign}(f(x_i))x_i$ until all data are correctly classified, or $2/w$ would be maximized. As this is a constrained optimization problem, it would be solved by the Lagrangian multipler method. It should be noted that Kernel trick would be used to increase the dimensionality of the data by transforming it (eg, 2-D into 3-D), and then creating the hyperplanes.

5.4 | Comparison across implemented algorithms

There are some intuitive differences across the implemented method worth discussion. All the methods are popular tools for building a prediction model. The methods are considered as non-parametric as they make no assumption on the data distribution, and the data structure. The quality of the decision tree prediction significantly depends on the tree size and the designated error threshold. The performance of SVM significantly depends on the types of kernel being

used for classification, for example, linear vs nonlinear. Thus, while SVM uses the kernel of turning a non-separable data into a linear or nonlinear separable problems based on maximizing the distance, the decision tree split the dataset based on answering specific questions. It should be noted that while decision tree and SVM involve with cost functions, Naïve Bayes method does not involved with any cost function: a simple classifier being based on the events probabilities.

While one method might work better on a specific problem, it might not work as well on another problem. All depend on the problem statement, quality of data, and level of sparsity of the dataset categories. Based on the discussed mathematical equations, while Naïve Bayes treat features as independent, SVM and decision trees look at the features as a whole, considering their interactions.

5.5 | Performance evaluation

Different metrics could be used to evaluate the performance of machine learning algorithm which is based on confusion or contingency matrix. True positive (TP) indicates the number of reviews which were positive and were classified as positive. False positive (FP) are the number of reviews that were positive but were incorrectly reviewed as negative. Similar explanations would be applied for true negative (TN), and false negative (FN).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{13}$$

Although the data are transformed to balanced, it would be helpful, especially for imbalanced dataset, to consider the Specificity metric for a comparison. The method could be written as:

$$\text{Specifity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{14}$$

6 | RESULTS

After conducting data preprocessing, and SMOTE to address imbalanced dataset, feature reduction was conducted to enhance the quality of the model by including important features only. For feature selection, the external resampling method was set as cross-validation. Number of resampling iterations was set as 10, 10-fold cross-validation. Cohen's kappa was set as a summary metric for evaluating the prediction performance of classifiers.

Different values were set for the number of predictors in the model as 10, 20, 30, 40, and 52. As can be seen from Figure 2, the model performs the best when all the predictors are included in the model Thus, for all the machine learning

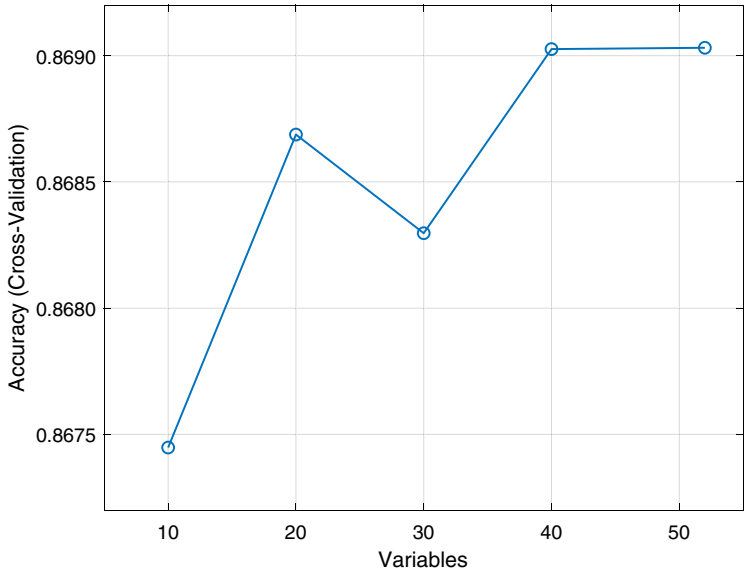


FIGURE 2 Feature elimination technique based on random forest

TABLE 2 Performance evaluation of algorithms for text classification

Method	Confusion matrix	Positive (0)P	Negative (1)P	Specificity	Accuracy
Naïve Bayes	Positive (0)R	238	219	67%	72%
	Negative (1)R	47	452		
SVM	Positive (0)	423	31	92%	82%
	Negative (1)	136	366		
Decision tree	Positive (0)	448	9	98%	89%
	Negative (1)	100	399		

techniques, we included all predictors chosen by n-grams method. Also, after conducting few experimental analyses, it was found that a combination of bigram and trigram would result in a better accuracy. As expected, although the machine learning techniques on original dataset resulted in low accuracy performances for negative review, after implementing SMOTE, the negative review results improved significantly (see Table 2).

6.1 | Application of SVM

The error rate based on confusion matrix is presented in Table 2. C-classification, classification machine, was used for type of SV. The 5-fold cross-validation was used for training the model to evaluate how well model can predict in general. A linear kernel was used for prediction as this type resulted in a better performance. Kernel method has been used for pattern analysis. This model was able to predict the polarity of review, positive vs negative, with accuracy of 82%.

6.2 | Application of Naïve Bayes

This classifier is a probabilistic model which is based on Bays theorem. In addition for this method reaching a lowest accuracy for negative review, the total review accuracy indicates that this model performs with a lowest accuracy in differentiating between positive and negative review.

6.3 | Application of decision tree

For this method, different hyperparameters could be set for the syntax to obtain a better accuracy. The minimum number of observations that must exist in a node for a split to be attempted is one of the important hyperparameters which if it is set wrongly, a model would be overfitted. For this model, a minimum number of 5 was set for this hyperparameter, which produced a higher accuracy. Another hyperparameter is a minimum number of observations in any terminal, which somehow depends on a setting of minimum number of observations in each split. It is intuitive that this value must be less than minimum number of observations in each split. By default, this value would be set as minimum number of $\frac{\text{observation}}{3}$. However, it was found that the value of 2 would result in a best performance.

The last hyperparameter was complexity parameter (cp). This hyperparameter could be used to control the size of the decision tree to prevent overfitting by preventing tree from growing if it does decrease error by a specific value. Any split which does not decrease the overall lack of fit by a factor of cp would not be attempted. The main role of this parameter is to save computing time by pruning off splits that are obviously not worthwhile. This value is set close to zero for the model to try all the options with no concern about overfitting. This method produces a highest accurate result compared with the other techniques.

7 | CONCLUSION AND DISCUSSION

Human language has been used not only for exchanging information but also for conveying positive or negative opinions about a product. The products reviews contain a wealth of information which could help shoppers and producers at the

same time. However, an essential issue is how to find a way to identify various sentiments being expressed in a text, and whether an expression indicates positive or negative opinions about a product. After identifying the knowledge about a polarity of a review, it is important to identify the factors that impact the sentiments of the shoppers so the products would be improved in the future. Identification of the keywords and how these words could predict future sentiment are one of the first steps to achieve the aforementioned goals.

Thus, this study was conducted by identifying an associated sentiment with each review content. A considerable effort has been made to preprocess the dataset before conducting any machine learning technique on the dataset. As negative reviews in this study accounted for less than 15% of all the review, if sparse term removal was conducted on this dataset, the preprocessing would result in a removal of almost all negative words. Thus, preprocessing steps were conducted separately on positive and negative reviews. Besides, words such as "but" or "not" were excluded from the list of stopwords due to the importance of these words for prediction of negative reviews. That is especially important as most of the reviewers negate their reviews by "not" or "but" like "I do not like" or "I wanted to like but." Also, some words like "pants," or "buy" were removed from text data as they did not add to the polarity of the prediction. Inclusion of those words resulted in removal of much of the important features.

As unigram resulted in keeping a single word with multiple meaning such as "like" in "do like" vs "not like," the sequence of one item is not included in the text preprocessing steps. As a combination of bigram and trigram resulted in a better performer, a combination of these two were considered for the preprocessing steps. In the next steps, and since Classification for imbalanced data is biased in favor of the majority class, SMOTE technique was used to produce a balanced dataset.

Various machine learning techniques, such as NB, SVM, and decision tree were conducted, and the results were compared. The results indicated that decision tree perform better in prediction of the review sentiments compared with the other two methods. The findings highlight an acceptable accuracy rate despite applications of machine learning techniques on imbalanced dataset with low observations frequencies.

The methodological steps taken in this research could be used in the future study to address the associated concerns with online shoppers' reviews, especially the negative aspects of products. The methodological approach highlight that the method could be used not only for identifying positive or negative reviews but also highlight the problems or advantages of a product. For instance, it was found that the main aspects of the products were related to their sizes: either reviewers were happy with the pants/jeans products' size or they believed they run small or large. More emphasize should be given on the design of the products' size to enhance the shoppers' satisfaction. For instance, specific products related to complaints about size should be pulled out and more work should be assigned to fix the problem related to that specific product. Future work would take advantages of words repetitions, for example, . "love love love," and use of capitalized words, "HATE IT," to show the degree of emotions.

7.1 | Concluding remarks

It has been argued that negative reviews could hurt the credibility or devalue a business. However, there is more that can be learnt from customers' reviews. What customers hate about a product would help to address the products' issues before tarnishing the credibility of a service. Addressing this issue could help to connect customers with the service providers in order to build credibility and trust. Businesses would benefit from constructive negative reviews as those reviews are most likely to help them to have a more realistic vision about their businesses and take actions to enhance trust across their customers. Furthermore, negative reviews also give the companies the opportunity to challenge themselves and make an improvement in their products.

PEER REVIEW INFORMATION

Engineering Reports thanks Andry Alamsyah and other anonymous reviewers for their contribution to the peer review of this work.

CONFLICT OF INTEREST

The authors declare no potential conflict of interest.

ORCID

Mahdi Rezapour https://orcid.org/0000-0003-0774-737X

REFERENCES

1. Ye Q. Zhang Z. and Law R., Sentiment classification of online reviews to travel destinations by supervised machine learning approaches, *Expert Syst Appl.*, 2009;36(3)6527–6535.
2. Liu B. Sentiment analysis and opinion mining. *Synth Lect Human Lang Technol*. 2012;5(1):1-167.
3. Feldman R. Techniques and applications for sentiment analysis. *Commun ACM*. 2013;56(4):82-89.
4. Jindal N, Liu B. Identifying comparative sentences in text documents. Paper presented at: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; 2006:244-251.
5. Turney PD. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. Paper presented at: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics; 2002:417-424.
6. Wang W, Feng Y, Dai W. Topic analysis of online reviews for two competitive products using latent Dirichlet allocation. *Electron Commerce Res Appl*. 2018;29:142-156.
7. Ghose A, Ipeirotis PG. Estimating the helpfulness and economic impact of product reviews: mining text and reviewer characteristics. *IEEE Trans Knowl Data Eng*. 2010;23(10):1498-1512.
8. Salehan M, Kim DJ. Predicting the performance of online consumer reviews: a sentiment mining approach to big data analytics. *Decis Support Syst*. 2016;81:30-40.
9. Bijalwan V, Kumar V, Kumari P, Pascual J. KNN based machine learning approach for text and document mining. *Int J Database Theory Appl*. 2014;7(1):61-70.
10. Dave K, Lawrence S, Pennock DM. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. Paper presented at: Proceedings of the 12th International Conference on World Wide Web; 2003:519–528.
11. Pang B, Lee L, Vaithyanathan S. Thumbs up?: Sentiment classification using machine learning techniques. Paper presented at: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing; 2002; vol. 10:79-86.
12. Nithya R, Maheswari D. Sentiment analysis on unstructured review. Paper presented at: 2014 International Conference on Intelligent Computing Applications; 2014:367–371.
13. Teh PL, Pak I, Rayson P, Piao S. Exploring fine-grained sentiment values in online product reviews. Paper presented at: 2015 IEEE Conference on Open Systems (ICOS); 2015:114–118.
14. Joachims T. Text categorization with support vector machines: Learning with many relevant features. Paper presented at: European Conference on Machine Learning; 1998:137–142.
15. Yang Y, Liu X. A re-examination of text categorization methods. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, 1999:42-49.
16. Porter MF. An algorithm for suffix stripping. *Dent Prog*. 1980;14(3):130-137.
17. Wang S, Yao X. Multiclass imbalance problems: analysis and potential solutions. *IEEE Trans Syst Man Cybern B Cybern*. 2012;42(4):1119-1130.
18. Longadge R, Dongre S. Class imbalance problem in data mining review; 2013. arXiv Preprint arXiv:1305.1707.
19. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. *J Artif Intell Res*. 2002;16:321-357.
20. Granitto PM, Furlanello C, Biasioli F, Gasperi F. Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products. *Chemom Intel Lab Syst*. 2006;83(2):83-90.
21. Rezapour M. Anomaly detection using unsupervised methods: credit card fraud case study. *Int J Adv Comput Sci Appl*. 2019;10(11).
22. Rezapour M, Ksaibati K. Application of various machine learning architectures for crash prediction, considering different depths and processing layers. *Eng Reports*. 2020;e12215. https://doi.org/10.1002/eng2.12215.
23. Guijarro F, Moya-Clemente I, Saleemi J. Liquidity risk and investors' mood: linking the financial market liquidity to sentiment analysis through twitter in the S&P500 index. *Sustainability*. 2019;11(24):7048.
24. El-Halees AM. A comparative study on Arabic text classification. Vol 30 (2); 2008.

How to cite this article: Rezapour M. Sentiment classification of skewed shoppers' reviews using machine learning techniques, examining the textual features. *Engineering Reports*. 2021;3:e12280. https://doi.org/10.1002/eng2.12280