Scientific Research Publishing

Open Journal of Statistics, 2022, 12, 357-372
https://www.scirp.org/journal/ojs
ISSN Online: 2161-7198
ISSN Print: 2161-718X

A Statistical Analysis of Textual E-Commerce Reviews Using Tree-Based Methods

Jessica Kubrusly, Ana Luiza Neves, Thamires Louzada Marques

Department of Statistics, Universidade Federal Fluminense, Niterói, Brazil
Email: jessicakubrusly@id.uff.br, ana_neves@id.uff.br, thamireslouzada@id.uff.br

How to cite this paper: Kubrusly, J., Neves, A.L. and Marques, T.L. (2022) A Statistical Analysis of Textual E-Commerce Reviews Using Tree-Based Methods. *Open Journal of Statistics*, **12**, 357-372.
https://doi.org/10.4236/ojs.2022.123023

Received: February 23, 2022
Accepted: June 11, 2022
Published: June 14, 2022

Abstract

With the increasing interest in e-commerce shopping, customer reviews have become one of the most important elements that determine customer satisfaction regarding products. This demonstrates the importance of working with Text Mining. This study is based on The Women's Clothing E-Commerce Reviews database, which consists of reviews written by real customers. The aim of this paper is to conduct a Text Mining approach on a set of customer reviews. Each review was classified as either a positive or negative review by employing a classification method. Four tree-based methods were applied to solve the classification problem, namely Classification Tree, Random Forest, Gradient Boosting and XGBoost. The dataset was categorized into training and test sets. The results indicate that the Random Forest method displays an overfitting, XGBoost displays an overfitting if the number of trees is too high, Classification Tree is good at detecting negative reviews and bad at detecting positive reviews and the Gradient Boosting shows stable values and quality measures above 77% for the test dataset. A consensus between the applied methods is noted for important classification terms.

Keywords

Text Mining, Supervised Classification, Tree-Based Methods, Classification Trees, Random Forest, Gradient Boosting, XGBoost

1. Introduction

It is true that most Internet data is in unstructured form, primarily text. These originate from social networks, such as Facebook and Twitter, or even corporate data, such as complaints and opinion polls. A real example is The Women's Clothing E-Commerce Reviews[1], which consists of reviews written by real cus-

[1]Nick Brooks. (2018). The women's e-commerce clothing reviews. Retrieved 2019 from https://www.kaggle.com/nicapotato/womens-ecommerce-clothing-reviews.

tomers. Text Mining, a Data Mining branch, has arisen from the need to process and extract information from this large mass of textual data.

A Sentiment Analysis comprises the computational treatment of opinions, sentiments and text subjectivity [1]. Its techniques can be roughly categorized into two groups, the first consisting in the Lexicon-based Approach, which associates words or expressions with positive, negative or neutral feelings. In this regard, Aung and Myo [2] proposed to automatically analyze student text feedback by employing the Lexicon-based approach to predict teaching performance levels. Their system indicated the opinion results of teachers, represented as strongly positive, moderately positive, weakly positive, strongly negative, moderately negative, weakly negative or neutral. Palanisamy *et al.* [3] also applied this approach to classify tweets as positive or negative based on the contextual sentiment orientation of the employed words.

The other Sentiment Analysis group comprises the Machine Learning Approach. Within this scheme, the learning model can be obtained by both supervised and unsupervised training. This approach relies on Machine Learning Algorithms to solve a Sentiment Analysis as a regular classification problem, applying linguistic features as independent variables. In this regard, Onan [4] presented a text mining approach to analyze MOOC reviews through supervised learning methods. Following a MOOC review selection, the training dataset consisted in half negative and half positive reviews. The highest predictive performance comprised a classification accuracy of 95.80% for all compared configurations. Ko and Seo [5], on the other hand, proposed an unsupervised method that classifies documents into sentences, categorizing each sentence through keyword lists for each category and sentence similarity measures. This method can be applied in areas where low-cost text categorization is required, or in the development of training documents.

Concerning the Machine Learning Approach, documents can be pre-processed, describing the original unstructured database, in order to create a structured database. An alternative is to work with term (word) presence or frequency in each document [2]. In a second step, Machine Learning methods can then be applied to the structured data. The method choice depends on the final objective. Descriptive analyses can be applied, followed by more elaborate models, such as regression or classification models.

In this context, this study analyzed The Women's Clothing E-Commerce Reviews database, which consists of consumer comments regarding a particular clothing item and a label designating whether or not the consumer indicates said item. The aim was to correctly and automatically classify customer recommendations based on their textual reports, applying Text Mining techniques and tree-based methods.

The paper is organized as follows. Section 2 cites some related studies, Section 3 describes the main Text Mining process steps and Section 4 presents the applied tree-based methods, namely Classification Tree, Random Forest, Gradient

Boosting and XGBoost. Section 5 comprises quality measures for the classification methods, Section 6 described the database, Section 7 presents some interesting numerical results and analyses and, finally, the conclusions are reported in Section 8.

2. Related Work

Alrehili and Albalawi [6] conducted a study employing a sentimental analysis approach on a set of customer reviews collected from Amazon. Following a manual review classification and text processing, the dataset remained with a 1500 record (750 positive reviews and 750 negative reviews) and two attributes: ReviewsText and ReviewType. Six classifiers, namely Naive Bayes, SVM, Random Forest (RF), Bagging using RF, Boosting using RF and Voting, were applied and tested by unigram, bigram, and trigram with stop word removal and without it. The best performance was obtained by the Random Forest technique, with a 89.87% accuracy when using unigram and a stop word removal.

Shah *et al.* [7] performed research on the sentiment analysis of a BBC news data set, where each text was categorized into one of five categories: business, entertainment, politics, sport and tech. Stop words were removed, the text was converted to lowercase and the Porter Stemmer algorithm was used for stemming. The TF-IDF Vectorizer was implemented to transform the text into a numerical representation. It was not clear how many terms were used. The dataset was split into training (75%) and testing (25%) sets and, after splitting, the pipeline was used to implement the classifiers. The results indicate that the Logistic Regression classifier attained the highest accuracy, of 97% and the second best was the Random Forest classifier, with a 93% accuracy. The Logistic Regression algorithm emerged as the most stable classifier for a small data set.

Lin [8] also employed The Women's Clothing E-Commerce Reviews database to perform a sentiment analysis of customer recommendations, aiming at understanding the correlation between review features and product recommendations based on natural language processing (NLP), applying five machine learning algorithms, *i.e.*, Logistic Regression, Support Vector Machine (SVM), Random Forest, XGBoost and LightGBM. The best results were achieved by the LightGBM algorithm, obtaining the highest AUC value and accuracy. The Ridge Regression, Linear Kernel SVM and XGboost algorithms exhibited close performances, with a 94% accuracy.

Comparing our study with the one described above [8], despite the same database and similar classification methods, two important differences are highlighted:

1) The dataset was not categorized into training and test sets in the Lin study. For this reason, the authors could not evaluate the out-of-sample performance. Sometimes a high accuracy determined for the training set is the result of overfitting.

2) An in-depth processing of the raw review texts in the Lin study was not

conducted. Furthermore, no stop words were removed, nor were terms selected by their frequency. This is an important step in Natural Language Processing (NLP) and was carefully performed in the study presented in this paper.

3. Text Mining

Text Mining is the process that basically consists in the extraction of non-trivial patterns or knowledge from unstructured text documents. This process can be categorized into two main steps: refinement, which transforms the original textual database into a numerical database; and the information extraction process, which consists of detecting patterns from the refined database using conventional statistical tools [9].

The main refinement steps of a textual database, also called preprocessing techniques, are: Tokenization; Stop Word Removal; Normalization; Creation of the Document-term Matrix; Term Selection. A brief explanation of each is presented below.

Tokenization is the first preprocessing stage and aims to extract minimum text units from a free text. These units are called *tokens* and most often refer to a single word.

Stop Words are the most frequent terms in a language. They have no semantic value and only aid in the general understanding of the text. Stop words are usually characterized by articles, prepositions, punctuation, conjunctions and pronouns. A pre-established list is usually applied, called a *stoplist*. The removal of stop words considerably reduces the amount of tokens and improves the analysis to be performed.

Normalization is the process of grouping words that share the same pattern. The main normalization methods are stemming and lemmatization, and further explanations on these terms can be found in [10]. The lemmatization method will be applied herein, which, for example, replaces the tokens "calculate", "calculating" and "calculated" for the term "calculate".

Term Selection, proposed by [11], establishes a set of significant database terms. Non-significant terms, which have low semantic value, appear at very high or very low frequencies in document sets and are not considered in the analyses.

Following these steps, each document was then transformed into a bag of terms. Considering a textual database formed by n documents that together contain p terms, the $n \times p$ matrix A, where each element $a_{i,j}$ represents the frequency with which the term j occurs in document i, is called the **Document-term Matrix**. Each line of this matrix corresponds to a document and can be understood as an object. Each column corresponds to a term and can be understood as a document attribute.

4. Tree-Based Methods

Several tree-based methods are applied in classification problems. The first, Classi-

fication Trees, is simple and useful for data interpretation, although it is not competitive in terms of prediction accuracy. On the other hand, the Random Forest and Boosting methods grow multiple trees which are then combined to yield a single consensus prediction. Combining many trees can often result in prediction accuracy improvements, at the expense of some loss of interpretation.

4.1. Classification Trees

Considering a universe composed of n objects which are described by p attributes, each attribute is an independent variable X_j, $j = 1, \cdots, p$, and each object i belongs to a known class Y_i. A classification method aims to define a mathematical model capable of predicting the class of a new object when its p attributes are known. The quality of a classification method is determined by the proportion of correctly predicted classes (more details are given in Section 5).

The Classification Tree model is a type of classification method. It uses the tree structure to recursively partition the dataset. Once the input data has been split, the prediction is made from a simple classification method in each partition, such as the dominant class (see Figure 1). If the resulting tree has too many nodes, it is still possible to perform a pruning process. The pruning process eliminates some nodes in order to minimize estimation errors outside the sample.

In general, for each node k, starting at the root, a classification tree algorithm can be summarized by the following three steps:

1) Choose an X_j attribute from the available p and a constant a_k which best separate the objects arriving at node k according to the following partition: $X_j < a_k$ and $X_j \geq a_k$. This partition defines two new nodes, the node k children. For each of these two child nodes,

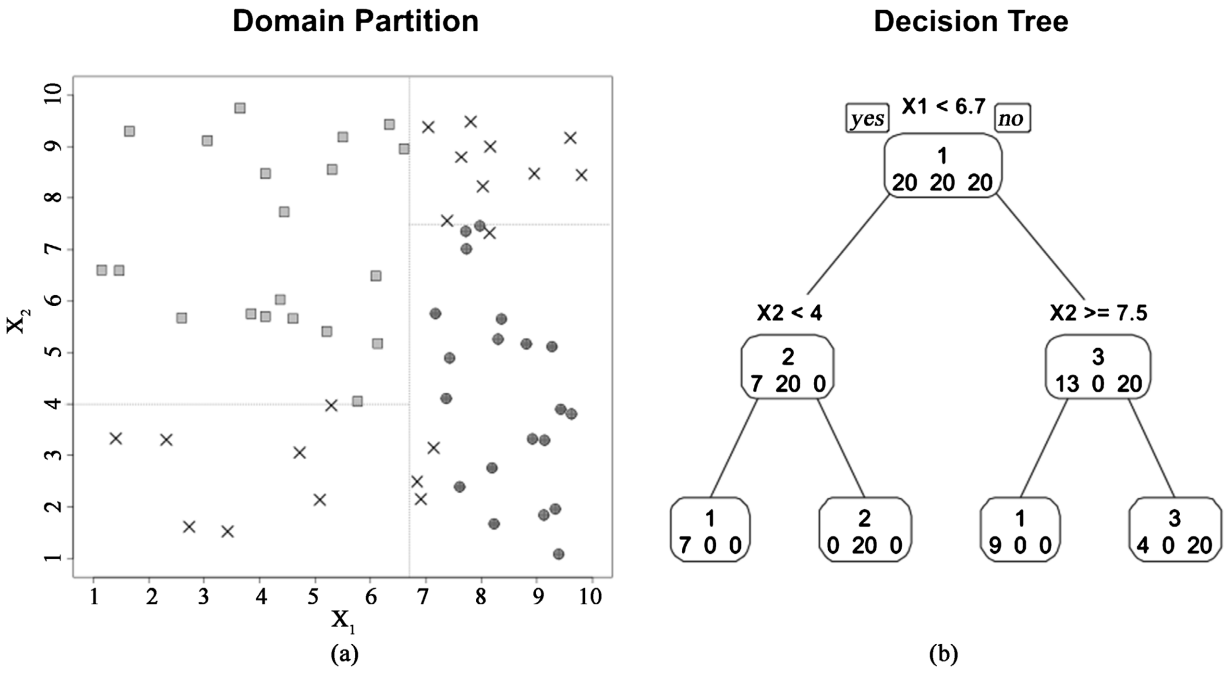

Figure 1. A classification tree example.

2) If this node comprises the prevalence of any class, *i.e.* a prevalence greater than a pre-defined value, this node becomes a leaf with this prevalent class and END.

3) Otherwise, go back to step 1 considering only the objects that arrived at this child node.

The choice of the X_j attribute and the constant a_k mentioned in Step 1 is performed in order to optimize the class division. The values of these two variables originate from the result of a complex optimization problem, which seeks to minimize impurities or maximize prevalence in the two new child nodes. Some different classification tree algorithms exist, and the main difference between them are the way the dataset is partitioned, *i.e.*, the choice of the X_j and a_k; the partition stopping criterion; and the pruning process [12].

For example, the THAID [13] [14], C4.5 [15] and CART [16] algorithms use node impurity measurements and divide a node by searching exhaustively over all X that minimize the total impurity of their two child nodes. Since the impurity measurements are different for each algorithm, the result is a difference in the way the dataset is partitioned. In addition, while a THAID division stops if the relative decrease in impurity is below a pre-specified threshold, C4.5 and CART first grow a tree too large and then prune it to a smaller size.

4.2. Random Forest (Bagging)

The Random Forest is a classification model created by Breiman [17] in order to improve the prediction of classification tree models. According to Breiman, the Random Forest is a classifier consisting of a collection of M tree-structured classifiers where each tree is constructed from a smaller dataset composed of n objects and p attributes.

The n objects are selected from a Bagging strategy, such as Boostrap Sampling [18]. The p attributes are selected randomly and without replacement. Each combination of these two draws results in a decision tree. After many trees are generated, these results are combined to provide a final prediction: the most popular class among each of the M predictions.

Just like classification trees, several Random Forest algorithms are available. These differ in the way the sample is selected and also in the adopted classification tree algorithm. In this study, we apply the Liaw and Wiener method [19] through the use of the randomForest package available in the R Program [20], described in the following steps.

1) Draw M bootstrap samples from the original data. Each sample is a $\tilde{n}$ size, $\tilde{n} \le n$.

2) For each of the bootstrap samples, grow an unpruned classification tree, with the following modification: at each node, rather than choosing the best split among all p attributes, randomly sample $\tilde{p}$ of the attributes, $\tilde{p} \le p$, and choose the best split from among those variables.

3) Predict new data by aggregating the M trees predictions (*i.e.*, majority votes for classification).

The values of the M, $\tilde{n}$ and $\tilde{p}$ constants are previously defined. For example, in the randomForest package, the default values for classification are: $M = 500$, $\tilde{n} = n$ and $\tilde{p} = \lfloor \sqrt{p} \rfloor$ (integer part of the square root of p).

4.3. Gradient Boosting Tree (Boosting)

The Random Forest method is a bagging algorithm. In these algorithms, trees are grown in parallel to obtain the average prediction across all trees, where each tree is built on an original data sample. Gradient boosting, on the other hand, employs a sequential approach in obtaining predictions. In the Gradient Boosting method, each decision tree predicts the error of the previous one [21].

With $T(\mathbf{X})$ as a decision tree using $\mathbf{X}$ independent variables with $\gamma\%$ of accuracy, ε the error and Y the dependent variable;

$$Y = T(X) + \varepsilon. \tag{1}$$

Next, the residual $Y - \hat{Y} = Y - T(X)$ can be predicted by another decision tree T_1, $\varepsilon = T_1(X) + \varepsilon_1$. When combining these two steps a new model for Y is obtained:

$$Y = T(X) + T_1(X) + \varepsilon_1 \tag{2}$$

In the next step the new residual can be predicted by another decision tree T_2, $\varepsilon_1 = T_2(X) + \varepsilon_2$, and then

$$Y = T(X) + T_1(X) + T_2(X) + \varepsilon_2 \tag{3}$$

The preceding equation is likely to present a greater accuracy than $\gamma\%$, while in the equation above three decision trees are considered [21].

The Gradient Boosting algorithm can be summarized by the following steps, where t is the maximum number of iterations, λ is the learning rate and $\delta\%$ is the minimum AUC (Area Under the Curve) required. These three parameters must be defined in advance.

1) Build a Classification Tree T that describe predicted class Y as a function of the independent variables $\mathbf{X}$ and set $k = 1$.

$$Y = T(\mathbf{X}) + \varepsilon \quad \Rightarrow \quad \hat{Y} = T(X)$$

2) Define the k-th iteration residuals, $r_k = Y - \hat{Y}$, and fit a Regression Tree with response variable r_k and predictor variables $\mathbf{X}$, terming this tree, T_k.

$$Y = T(\mathbf{X}) + \sum_{j=1}^{k} T_j(X)\lambda + \varepsilon_k \quad \Rightarrow \quad \hat{Y} = T(\mathbf{X}) + \sum_{j=1}^{k} T_j(X)\lambda$$

3) Calculate the model error given by the AUC (Area Under the Curve) metric, called AUC_k. If $k < t$ and $AUC_k < \delta\%$, set $k = k + 1$ and go back to step 3. Otherwise, end the algorithm.

4.4. XGBoost (Boosting)

The XGBoost method is an upgraded Gradient Boosting Tree algorithm that can flexibly process sparse data and missing values [8]. The system runs more than

ten times faster than existing popular solutions on a single machine and scales to billions of examples in distributed or memory-limited settings. It also incorporates a regularized model to prevent overfitting [22].

5. Quality Measures

Consider that a database with n objects is tested by a classifier, which will predict one between two possible classes for each object. After the test it is possible to build a confusion matrix like the one presented in **Table 1**.

Table 1. Confusion matrix.

Predicted class	Actual class	
	Negative	Positive
Negative	True Negative (TN)	False Negative (FN)
Positive	False Positive (FP)	True Positive (TP)

From the confusion matrix, some quality classification measures can be defined: True Positive Rate, True Negative Rate, Accuracy, Precision and F1 score. Each one is defined below.

Name	Expression	Description
True Positive Rate	$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{FN} + \mathrm{TP}}$	Also called sensitivity or recall: is the proportion of actual positives that are correctly identified as a positive class.
True Negative Rate	$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$	Also called specificity: is the proportion of actual negatives that are correctly identified as a negative class.
Accuracy	$\mathrm{A} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TN} + \mathrm{FP} + \mathrm{FN} + \mathrm{TP}}$	Is the proportion of correct predictions in the total observations.
Precision	$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{FP} + \mathrm{TP}}$	is the proportion of predicted positives that are actually a positive class.
F1 score	$\mathrm{F1} = 2\frac{\mathrm{P} \times \mathrm{TPR}}{\mathrm{P} + \mathrm{TPR}}$	Is the harmonic means of the precision and recall (TPR) and can be used as a single measure of the test performance for the positive class.

6. The Dataset

The Women's Clothing E-Commerce Reviews was used as the dataset for this study and revolves around reviews written by customers. This dataset includes 23,486 rows and 10 feature variables. Each row corresponds to a customer review, and includes the following variables: Clothing ID; Age; Title; Review Text; Rating; Recommended IND; Positive Feedback Count; Division Name; Department Name; and Class Name. Of the 23,486 rows in the database, 19,314 refer to recommended items while the other 4,172 refer to non-recommended items.

Only three variables were considered herein: **Review Text**, string variable for the review body; **Title**, string variable for the title of the review; and **Recommended IND**, binary variable stating whether the customer recommends the product, where 1 is recommended and 0 is not recommended. Variables Title and Review Text were concatenated in order to add more wealth of information to the analysis. Then, only the text was used as a classification method attributes.

First, the dataset was randomly split into 70% as a training set and 30% as a testing set. Since the original database contains many more recommended objects compared to non-recommended, not all of the documents should be used. To select balanced sets and respect the 70/30 ratio, the number of documents in the training set was 5840 and the number of documents in testing was 2504.

Figure 2 presents a diagram with the main stages of this research.

7. Results

This study was performed using the R Program [20]. The tidytext [23], tm [24] and textstem [25] packages were used for textual pre-processing. The rpart [26], randomForest [19], gbm [27] and xgboost [22] packages were used for the Classification Tree, Random Forest, Gradient Boosting and XGBoost analyses, respectively.

The pre-processing described in Section 3 was performed for the training dataset. An extra step was performed after the word tokenization, were we concatenate the term "not' with the next word. Subsequently, stop words were removed and the normalization process was conducted based on Mechura's English

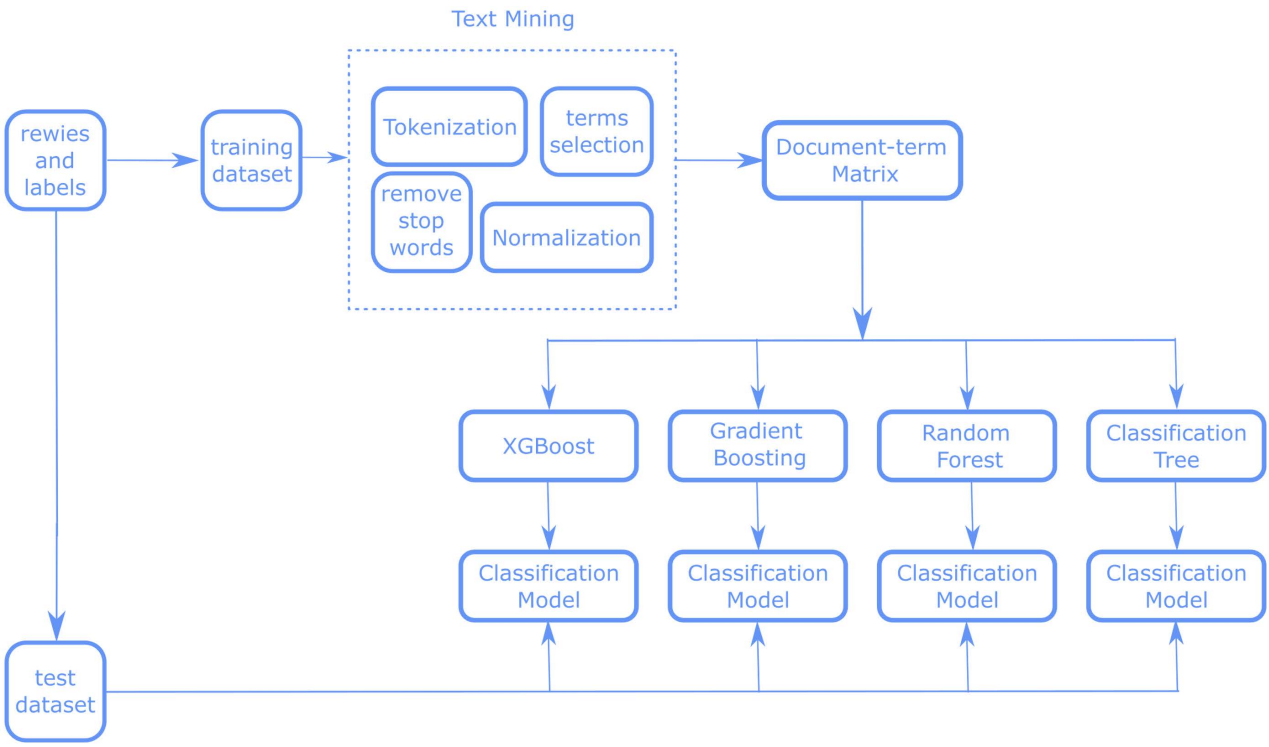

Figure 2. Research content diagram.

lemmatization list[2]. Following this process, the textual database contained 123,769 terms.

A term selection was carried out to select 100 of these terms. The 10 most frequent terms in all documents and the 10 most frequent terms per category, positive and negative, were analyzed. Some frequent terms were noted in all documents as among the top 10 by category, and because of that, comprised non-significant terms and were dismissed. These terms were removed in an iterative process until no more terms in common between top 10 frequent terms in positive reviews and top 10 frequent terms in negative reviews remained. The terms removed in this process were as follows: "dress", "fit", "love", "size", "top", "wear", "color", "fabric", "cute", "shirt", "run", "pretty", "beautiful", "short", "sweater", "material", "nice" and "buy".

The 10 most frequent terms in all documents and per category, positive and negative, are presented in **Figure 3**. A naive visual inspection would indicate that the most frequent terms in the positive reviews are more positive than those in the negative reviews, confirmed by the count of positive and negative terms in each of these groups. The classification of terms as positive or negative can be assessed by a sentiment lexicon dictionary, like the Bing sentiment lexicon available by the package textdata [28] in R Program [20]. The positive terms presented in **Figure 3** are "perfect", "flatter", "soft" and "comfortable", all more frequent in the positive reviews. Only "disappoint" comprises a negative term by Bing dictionary, while the other terms were considered neutral. We believe that in the specific e-commerce context, the term "return" can be considered a negative term.

The 100 most frequent terms were selected from the remaining terms. The

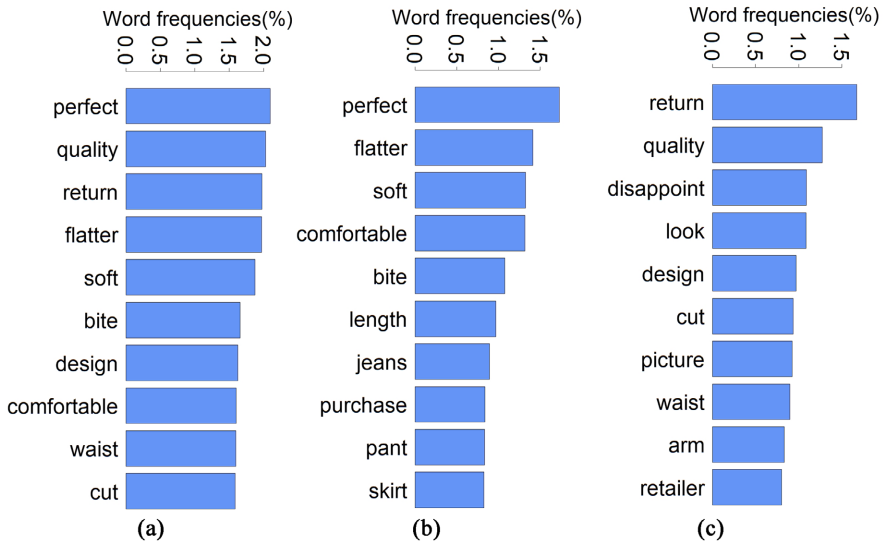

Figure 3. Most Frequent Words within the Training Dataset. (a) Complete Data; (b) Positive Label; (c) Negative Label.

[2]Michal Měchura (2018). Lemmatization-lists. Retrieved 2019 from https://github.com/michmech/lemmatization-lists.

final Document-term matrix was a 5618 × 100 matrix, applied as the input data for the classification methods presented below.

A simple Classification Tree (CT) was adjusted according to the CART algorithm [16]. A Random Forest method was run with the following parameter values: $\tilde{n} = 5618$ and $\tilde{p} = \lfloor \sqrt{p} \rfloor = \lfloor \sqrt{100} \rfloor = 10$. In order to test the method performance for different M parameter values, $M = 100$, 300, and 500 were adopted (RF100, RF300 and RF500). $M = 500$ is the default value for the Random Forest package [19].

For the Gradient Boosting algorithm $\lambda = 0.1$ and $t = 100$, 300 and 500, (GB100, GB300 and GB500) where t is the number of performed tree iterations and λ is the learning rate. $t = 100$ is the default value for gbm package [27], but for comparison purposes the same number of trees as in the Random Forest method was applied. The same was done for XGboost algorithm, $t = 100$, 300 and 500 (XGB100, XGB300 and XGB500) and $\lambda = 0.3$. For Gradient Boosting and XGboost algorithms λ was set as the default R Program [20] packages values.

A total of 10 methods were run, as follows: CT, RF100, RF300, RF500, GB100, GB300, GB500, XGB100, XGB300 and XGB500. The results of quality measures for the training dataset are presented in **Table 2** and in **Figure 4**. In the latter, the Classification Tree (CT) is indicates in blue, the Random Forest methods (RF100, RF300 and RF500), in red, Gradient Boosting methods (GB100, GB300 and GB500), in yellow and XGBoost methods (XGB100, XGB300 and XGB500), in green.

The three Random Forest methods exhibited the best results for all quality measures. The number of trees did not influence the performance of this method. The XGBoost method ranked second, also with good results. However, unlike the Random Forest method, the higher the number of trees, the better the quality measures of the training data.

Table 2. Evaluation of the different models applied to the training dataset.

Method	Accuracy	Precision	Recall	Specificity	F1
CT	0.7122	0.8213	0.5344	0.8862	0.6475
RF100	0.9706	0.9729	0.9676	0.9736	0.9702
RF300	0.9713	0.9688	0.9734	0.9694	0.9711
RF500	0.9710	0.9661	0.9755	0.9665	0.9708
GB100	0.7729	0.7939	0.7305	0.8144	0.7609
GB300	0.7933	0.7827	0.8060	0.7809	0.7942
GB500	0.8003	0.8081	0.7819	0.8182	0.7948
XGB100	0.8740	0.8789	0.8643	0.8834	0.8716
XGB300	0.9179	0.9264	0.9061	0.9296	0.9161
XGB500	0.9422	0.9572	0.9244	0.9595	0.9405

The Gradient Boosting algorithm presented quality measures above 73% for all metrics, with the behavior not very sensitive to variations in the number of trees. The Classification Tree method presented the worst results and a high Specificity combined with low Recall value. This indicates that the Classification Tree method can detect negative classes but not positive ones.

Table 3 displays the 10 most relevant terms for each method. The first four most important terms were the same for all methods: "return", "disappoint", "perfect" and "comfortable". In addition, three other terms appear in common in the list of the top 10 most important terms for all methods, *i.e.*, "jeans", "unflattering" and "look". Therefore, a consensus is noted between the applied methods concerning important classification terms.

The out-of-sample results are presented in Table 4 and in Figure 5. The color criteria in Figure 5 are the same as that presented in Figure 4.

All methods showed similar results, except for the Classification Tree method. Even the Random Forest and XGBoost, which performed better on the training

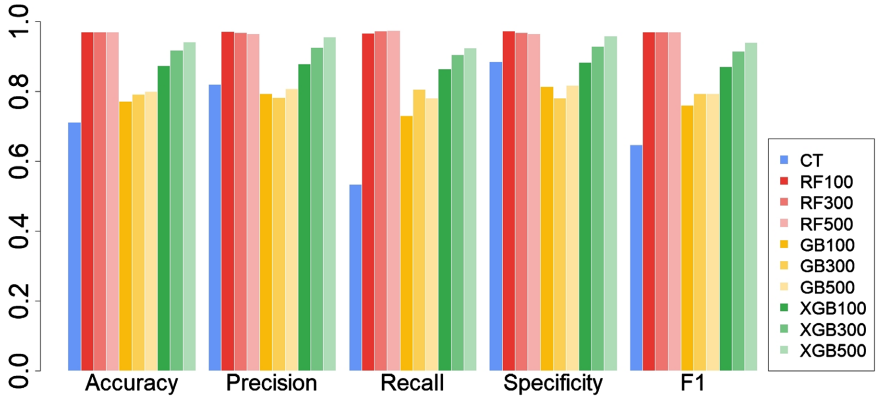

Figure 4. Evaluation of the different models applied to the training dataset.

Table 3. The ten most relevant terms per method.

	CT	RF			GB			XGB		
		100	300	500	100	300	500	100	300	500
1st	"return"									
2nd	"disappoint"									
3rd	"perfect"									
4th	"comfortable"									
5th	"unflattering"	"look"						"unflattering"		
6th	"jeans"	"unflattering"						"look"		
7th	"perfectly"	"huge"		"jeans"			"flatter"		"jeans"	
8th	"look"	"jeans"		"flatter"			"jeans"		"huge"	
9th	"flatter"	"bad"		"jeans"			"huge"		"flatter"	
10th	"comfy"	"perfectly"		"jeans"			"bad"			

Table 4. Evaluation of the different models applied to the testing dataset.

Method	Accuracy	Precision	Recall	Specificity	F1
CT	0.7092	0.8015	0.5476	0.8673	0.6507
RF100	0.7717	0.7669	0.7734	0.7700	0.7701
RF300	0.7729	0.7640	0.7826	0.7634	0.7732
RF500	0.7700	0.7563	0.7894	0.7510	0.7725
GB100	0.7608	0.7724	0.7321	0.7890	0.7517
GB300	0.7783	0.7631	0.8003	0.7568	0.7812
GB500	0.7883	0.7909	0.7776	0.7988	0.7842
XGB100	0.7779	0.7753	0.7759	0.7799	0.7756
XGB300	0.7696	0.7664	0.7683	0.7708	0.7674
XGB500	0.7588	0.7648	0.7397	0.7774	0.7520

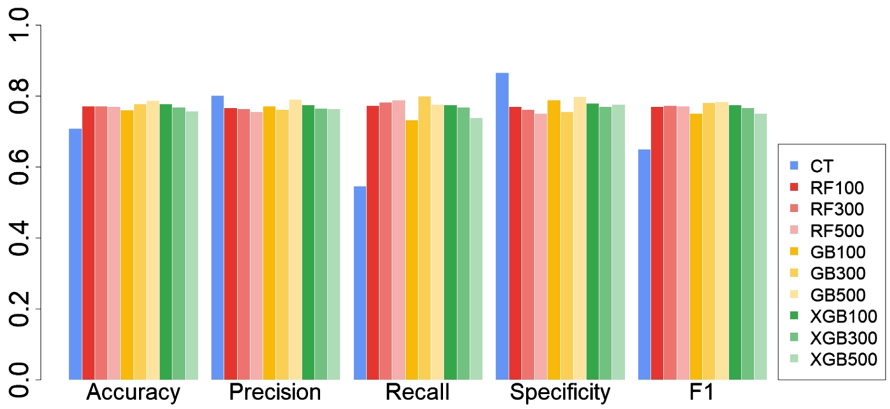

Figure 5. Evaluation of the different models applied to the test dataset.

dataset, now no longer exhibit advantages over the other methods. This may indicate that these models are overfitting the training data, especially the Random Forest and XGBoost models with many trees.

The Classification Tree results are exhibited in Table 4 and Figure 5 for. This model exhibited the best Specificity value and the worst Recall value. This method again presents a high Specificity value and a low Recall value, indicating that it is not able to identify positive reviews.

It is important to note the good performance of the Gradient Boosting method. Its in-sample quality measures are close to the out-of-sample quality measures, with values above 77% for all metrics, indicating its real classification capability.

8. Conclusions

A Text Mining analysis via consumer reviews, *i.e.*, free text, referring to recommendable or not recommendable products, was performed. The goal was to predict whether a product is recommended by the consumer based on their review alone. Tree-based classification methods were applied in a balanced training da-

taset containing 5840 review, with the test dataset containing 2504 reviews. The training and test datasets were randomly selected.

The Text Mining analysis indicated that some very frequent terms were non-significant, as they appeared very frequently in both positive and negative reviews, as follows: "dress", "fit", "love", "size", "top", "wear", "color", "fabric", "cute", "shirt", "run", "pretty", "beautiful", "short", "sweater", "material", "nice" and "buy", eliminated before applying the classification methods.

The 10 most frequent words in the positive reviews were "perfect", "flatter", "soft", "comfortable", "bite", "length", "jeans", "purchase", "pant" and "skirt". On the other hand, the 10 most frequent words in the negative reviews were "return", "quality", "disappoint", "cut", "picture", "waist", "arm" and "retailer". A naive visual inspection would seem to indicate that the most frequent terms in the positive reviews are more positive than the most frequent in the negative reviews. A simple sentimental analysis demonstrated that, considering the Bing sentiment lexicon, the positive terms among these words were among the most frequent in the positive reviews, and the only negative one was noted among the most frequent in negative reviews.

The Classification Tree, Random Forest, Gradient Boosting and XGBoost methods were applied to classify documents from a supervised database. The Random Forest and Gradient Boosting methods are not very sensitive to a varying number of trees, while the XGBoost method is. The higher number of trees, the more the XGBoost overfitted the training data. The results suggest that the Random Forest overfitted the training data independent of the number of trees, as $M < 100$ would not be appropriate for this parameter.

The Classification Tree presented high Specificity and low Recall in the test dataset. This means this method is adequate for the detection of negative reviews, but cannot detect positives reviews. The Gradient Boosting was the most robust method and the number of trees did not make much difference regarding the results. The Gradient Boosting exhibited similar metrics when comparing the training and test datasets. Considering the 500-tree model, all quality measures were above 77% in the test dataset.

Finally, the first four most important terms were the same for all methods, and three other terms appear in common in the list of the top 10 most important classification terms for all methods. This suggests a consensus between the applied methods regarding important classification terms. The seven common terms in the top 10 most important terms for the classifying methods were "return", "disappoint", "perfect", "comfortable", "jeans", "unflattering" and "look".

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Medhat, W., Hassan, A. and Korashy, H. (2014) Sentiment Analysis Algorithms and

Applications: A Survey. *Ain Shams Engineering Journal*, **5**, 1093-1113. https://doi.org/10.1016/j.asej.2014.04.011

[2] Aung, K.Z. and Myo, N.N. (2017) Sentiment Analysis of Students' Comment Using Lexicon Based Approach. 2017 *IEEE/ACIS* 16*th International Conference on Computer and Information Science* (*ICIS*), Wuhan, 24-26 May 2017, 149-154.

[3] Palanisamy, P., Yadav, V. and Elchuri, H. (2013) Serendio: Simple and Practical Lexicon Based Approach to Sentiment Analysis. *Second Joint Conference on Lexical and Computational Semantics* (**SEM*), Volume 2, 543-548.

[4] Onan, A. (2021) Sentiment Analysis on Massive Open Online Course Evaluations: A Text Mining and Deep Learning Approach. *Computer Applications in Engineering Education*, **29**, 572-589. https://doi.org/10.1002/cae.22253

[5] Ko, Y. and Seo, J. (2000) Automatic Text Categorization by Unsupervised Learning. *COLING* 2000: *The* 18*th International Conference on Computational Linguistics*, Volume 1, 453-459.

[6] Alrehili, A. and Albalawi, K. (2019) Sentiment Analysis of Customer Reviews Using Ensemble Method. 2019 *International Conference on Computer and Information Sciences* (*ICCIS*), Sakaka, 3-4 April 2019, 1-6. https://doi.org/10.1109/ICCISci.2019.8716454

[7] Shah, K., Patel, H., Sanghvi, D. and Shah, M. (2020) A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification. *Augmented Human Research*, **5**, 1-16. https://doi.org/10.1007/s41133-020-00032-0

[8] Lin, X.X. (2020) Sentiment Analysis of e-Commerce Customer Reviews Based on Natural Language Processing. *Proceedings of the* 2020 2*nd International Conference on Big Data and Artificial Intelligence*, Johannesburg, 28-30 April 2020, 32-36.

[9] Tan, A.-H., *et al.* (1999) Text Mining: The State of the Art and the Challenges. *Proceedings of the PAKDD* 1999 *Workshop on Knowledge Discovery from Advanced Databases*, Volume 8, 65-70.

[10] Toman, M., Tesar, R. and Jezek, K. (2006) Influence of Word Normalization on Text Classification. *Proceedings of InSciT*, Vol. 4, 354-358.

[11] Luhn, H.P. (1958) The Automatic Creation of Literature Abstracts. *IMB Journal*, **2**, 159-165. https://doi.org/10.1147/rd.22.0159

[12] Loh, W.-Y. (2011) Classification and Regression Trees. *Wiley Interdisciplinary Reviews*: *Data Mining and Knowledge Discovery*, **1**, 14-23. https://doi.org/10.1002/widm.8

[13] Fielding, A. and O'Muircheartaigh, C.A. (1977) Binary Segmentation in Survey Analysis with Particular Reference to Aid. *Journal of the Royal Statistical Society*: *Series D* (*The Statistician*), **26**, 17-28. https://doi.org/10.2307/2988216

[14] Messenger, R. and Mandell, L. (1972) A Modal Search Technique for Predictive Nominal Scale Multivariate Analysis. *Journal of the American Statistical Association*, **67**, 768-772. https://doi.org/10.1080/01621459.1972.10481290

[15] Ross Quinlan, J. (2014) C4.5: Programs for Machine Learning. Elsevier, Amsterdam.

[16] Breiman, L., Friedman, J., Stone, C.J. and Olshen, R.A. (1984) Classification and Regression Trees. CRC Press, Boca Raton.

[17] Breiman, L. (2001) Random Forests. *Machine Learning*, **45**, 5-32. https://doi.org/10.1023/A:1010933404324

[18] Zhan, C.J., Zheng, Y.F., Zhang, H.J. and Wen, Q.S. (2021) Random-Forest-Bagging Broad Learning System with Applications for Covid-19 Pandemic. *IEEE Internet of Things Journal*, **8**, 15906-15918. https://doi.org/10.1109/JIOT.2021.3066575

[19] Liaw, A. and Wiener, M. (2002) Classification and Regression by Random Forest. *R News*, **2**, 18-22.

[20] R Core Team (2020) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna.

[21] Kishore Ayyadevara, V. (2018) Pro Machine Learning Algorithms. Apress, Berkeley. https://doi.org/10.1007/978-1-4842-3564-5

[22] Chen, T.Q. and Guestrin, C. (2016) Xgboost: A Scalable Tree Boosting System. *Proceedings of the* 22*nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, 13-17 August 2016, 785-794. https://doi.org/10.1145/2939672.2939785

[23] Silge, J. and Robinson, D. (2016) Tidytext: Text Mining and Analysis Using Tidy Data Principles in R. *The Journal of Open Source Software*, **1**, 37. https://doi.org/10.21105/joss.00037

[24] Feinerer, I., Hornik, K. and Meyer, D. (2008) Text Mining Infrastructure in R. *Journal of Statistical Software*, **25**, 1-54. https://doi.org/10.18637/jss.v025.i05

[25] Rinker, T.W. (2018) Textstem: Tools for Stemming and Lemmatizing Text. Version 0.1.4. Buffalo, New York.

[26] Therneau, T. and Atkinson, B. (2018) Rpart: Recursive Partitioning and Regression Trees. R Package Version 4.1-13.

[27] Greenwell, B., Boehmke, B., Cunningham, J. and GBM Developers (2020) Gbm: Generalized Boosted Regression Models. R Package Version 2.1.8.

[28] Hvitfeldt, E. (2020) Textdata: Download and Load Various Text Datasets. R Package Version 0.4.1.