Advances in Data Analysis and Classification
https://doi.org/10.1007/s11634-023-00536-8

REGULAR ARTICLE

Threshold-based Naïve Bayes classifier

Maurizio Romano[1] · **Giulia Contu**[1] · **Francesco Mola**[1] · **Claudio Conversano**[1]

Received: 30 September 2021 / Revised: 16 February 2023 / Accepted: 20 February 2023

Abstract
The Threshold-based Naïve Bayes (Tb-NB) classifier is introduced as a (simple) improved version of the original Naïve Bayes classifier. Tb-NB extracts the sentiment from a Natural Language text corpus and allows the user not only to predict how much a sentence is positive (negative) but also to quantify a sentiment with a numeric value. It is based on the estimation of a single threshold value that concurs to define a decision rule that classifies a text into a positive (negative) opinion based on its content. One of the main advantage deriving from Tb-NB is the possibility to utilize its results as the input of post-hoc analysis aimed at observing how the quality associated to the different dimensions of a product or a service or, in a mirrored fashion, the different dimensions of customer satisfaction evolve in time or change with respect to different locations. The effectiveness of Tb-NB is evaluated analyzing data concerning the tourism industry and, specifically, hotel guests' reviews from all hotels located in the Sardinian region and available on Booking.com. Moreover, Tb-NB is compared with other popular classifiers used in sentiment analysis in terms of model accuracy, resistance to noise and computational efficiency.

Keywords Naïve Bayes · Booking.com · Customer satisfaction · Sentiment analysis · Natural language processing · Word of mouth

Mathematics Subject Classification 62-08 · 62C10 · 62F15 · 62H30 · 62P20 · 68T50

✉ Claudio Conversano
conversa@unica.it

Maurizio Romano
romano.maurizio@unica.it

Giulia Contu
giulia.contu@unica.it

Francesco Mola
mola@unica.it

[1] Department of Economics and Business Science, University of Cagliari, Cagliari, Italy

Published online: 14 March 2023

1 Introduction

Big data is known as a new research paradigm that utilizes diverse sources of data and analytical tools to make inferences and predictions about reality (Boyd and Crawford 2012). With increasingly powerful natural language processing and machine learning capabilities, textual contents from the Web provide a huge shared cognitive and cultural context and have been analyzed in many application domains (Halevy et al. 2009). For example online reviews including their peripheral cues (user-supplied photos and the reviewer's personal information) are means of persuasive communication in order to build credibility and influence user behavior (Sparks et al. 2013). Operationally, many *Natural Language Processing* (NLP) challenges and techniques have been introduced, most of them addressed to gather the opinion of people. *Sentiment Analysis* (SA) has been used in frameworks such as subjectivity detection (Wiebe et al. 1999), polarity recognition (Schmunk et al. 2013) and rating inference (Esuli and Sebastiani 2006). Focusing on product review classification, various approaches only consider the polarity of the opinions (i.e., negative vs. positive) and rely on machine learning techniques trained over vectors of linguistic feature frequencies.

Machine learning has been applied in various domains of SA such as Twitter sentiment (Tavazoee et al. 2020), scientific citations, reputation evaluation and tourism (Jain et al. 2021). In this latter domain, the increasing amount of data available offers SA new possibilities to predict consumer sentiment and use it for business growth. Practitioners should be equipped with most updated consumers' feedback, particularly online reviews, to confirm that their conclusions are sound. In this paper, we consider the problem of measuring the customer satisfaction of clients' hosted in accommodations, hereafter hotels, based on their reviews. In particular, we focus on reviews obtained from Booking.com that are composed of two comments: a positive comment reporting what a client liked about the hotel service, and a negative comment reporting what she disliked. With such a text structure, main aim is to estimate the polarity, positive or negative, of the review as a whole. For this purpose, we compare the performance of different machine learning methods and highlight the effectiveness of the hereby proposed *Threshold-based Naïve Bayes classifier* (Tb-NB) introduced as a simple modification of the original Naïve Bayes classifier. After a formal description of Tb-NB features and its associated decision rule, we highlight that Tb-NB effectively discriminates positive comments from negative ones and, at the same time, allows us to quantify the (positive or negative) impact of a specific word within a review. At the same time, we show that the information deriving from Tb-NB can be used to support hotel management as the model output can be used further either cross-sectionally, geographically or longitudinally in a post-hoc analysis to evaluate different facets of customer satisfaction of hotel guests. At the same time, Tb-NB output can also be used as the basis of a prediction model for the score obtained by a hotel based on the reviews reported on Booking.com. In this view, predicted scores are considered as a benchmarking tool for a hotel to be evaluated. Last but not least, we consider accuracy, resistance to noise and computational efficiency of Tb-NB in comparison with other methods. The analyses are carried out on data about clients' reviews concerning hotels located in Sardinia retrieved from the Booking.com website.

The remaining of the paper is as follows. Before introducing formally Tb-NB (Sect. 3), we present the reference framework of this study in Sect. 2 together with the related literature. Next, we describe the data collection and cleaning process, and present the post-hoc analysis in Sect. 4. Section 5 focuses on the comparison of Tb-NB with alternative methods evaluating accuracy, resistance to noise, and computational efficiency of each classifier. Section 6 ends the paper with some concluding remarks.

2 Reference framework and related literature

This paper is framed within the "Natural Language Processing" (NLP) and its applications in "Sentiment Analysis" (SA).

"Natural Language Processing (NLP) is the field of designing methods and algorithms that take as input or produce as output unstructured, natural language data" (Goldberg 2017). Such kind of processing is considered to be highly challenging. Although humans are great users of language, "they are poor at formally understanding and describing the rules that govern language" (Brownlee 2017). However, rules are not the only problem: the textual nature of collected data, if treated in the usual way, is not sufficient. Using textual data by combining characters might lead to produce an immense amount of words, that are combinable in infinite ways. This "data sparseness" phenomenon makes it hard to work with the usual way of "learning from examples" from just raw data. Data preprocessing (filtration, lexical, grammatical, syntactic and semantic analyses) is then mandatory for reducing both data complexity and the intrinsic ambiguity of the human (natural) language.

NLP techniques are often used in Sentiment Analysis (SA) to extract the sentiment information from a text (e.g. Pang and Lee 2008). SA is aimed at categorizing people' reactions starting from a Natural Language text into positive, neutral, or negative responses. Organizations use SA to collect previous consumer experiences of their services or products and may use their findings for improving services as well as to collect valuable consumers' experiences about issues in newly released products (Chaturvedi et al. 2018). Usually, SA makes use of machine learning methods as it is usually applied by learning a statistical model, hereafter classifier, on labelled or unlabelled training data gathered from various sources. The predictive ability of the classifier is validated on more recent data in order to estimate the polarity of new text and enhance the decision-making ability of the classifier. Several classifiers have been used for this purpose and the literature about machine learning in SA is vast (see, for example Yang and Chen 2017).

Next, in an operational or marketing perspective SA of raw text based on NLP and machine learning is linked to the analysis of customers' experience regarding a service quality or product performance (Buttle 1998). This is traceable to the *Word Of Mouth* (WoM) and *electronic Word Of Mouth* (e-WoM) frameworks (Arndt 1967; Yuan et al. 2020; Hartline and Jones 1996; Harrison-Walker 2001; Mazzarol et al. 2007), whose marketing campaigns are perceived as more robust whilst trustworthy compared to traditional marketing channels. According to Sırma (2009), the proliferation of online customers' reviews (e-WOM) has been reported as one of the most important information sources in the industry and has gained considerable attention (Schuckert et al.

2015). Results reported in Nielsen (2007) and Rusticus (2007) show that most of consumers mainly rely on recommendations from other consumers, because consumers find it challenging to evaluate a product or a service before actually using it or trying it by themselves (O'Connor 2010; Yang et al. 2016).

Many studies in the field of SA are based on Natural Language text (see, for example, Ye et al. 2009 for an overview). Those who have mainly influenced our proposal are Santos et al. (2020), Meyer et al. (2019), and Weihs et al. (2005). Santos et al. (2020) compute a sentiment score defined in $[-4; +4]$ starting from NL text reviews using the "SentiStrength tool". Although they do not make use of a proper classifier, they carry out an experiment with volunteers whose results confirm the usefulness of the WOM/eWOM data for SA. Bachtiar et al. (2020) and Janowicz-Lomott et al. (2020) evaluate the performance of the Naïve Bayes classifier in comparison with that of SVM, CART and Random Forest using several performance metrics in a binary classification framework. Lastly, Narayanan et al. (2013) introduced an improved version of the Naïve Bayes classifier that they call Bernoulli NB. They consider a maximum likelihood version of NB in place of that based on "raw" frequencies. Their results confirm that the performance of the NB classifier is comparable (or even better) than that of other methods. As for the implementations of the Naïve Bayes classifier, Meyer et al. (2019) and Weihs et al. (2005) implement two classifiers that are similar to that introduced in this paper and can be used in SA. The e1071 R package (Meyer et al. 2019) implements the classical Naïve Bayes classifier, whilst the klaR (Weihs et al. 2005) R package can be considered as an extended version of e1071 as it utilizes kernel-density estimation within Naïve Bayes.

We build on the above-mentioned studies and consider two important aspects. The first one involves the data cleaning phase. Although we resort to the usual noise filtering, which is being considered mandatory when analyzing textual data, we reduce the dimensionality of the raw text data. We "merge the words by their meaning" reducing the amount of less frequent words whilst considering in exactly the same way words that might have been written in a different manner but contain the same information. A second important aspect relates to the core component of the proposed Tb-NB classifier. As better described in Sect. 3, we estimate the log-likelihood ratio of an event, the latter intended as "a word appears in a text corpora" and "the same word does not appear in a text corpora". The resulting sentiment score (value of the log-odds ratio) is defined in $(-\infty; +\infty)$. The most extreme values represent a negative sentiment or a positive one, respectively. Thus, the proposed classifier produces a continuous sentiment score that allows us not only to be consistent with the classification task (in other words, to classify a text into positive or negative) but also to use the obtained score in a post-hoc analysis that highlights how the different dimensions of service can influence the overall sentiment score (see Sect. 4.3).

3 Threshold-based Naïve Bayes classifier

Bayesian classifiers assign the most likely class to a given example described by its feature vector. The resulting classifier known as Naïve Bayes (NB) is remarkably successful in practice, often competing with much more sophisticated techniques (Huang

et al. 2003). It has proven effective in many practical applications, including text classification. The success of NB can be explained as follows: optimality in terms of classification error is obtained as long as, for each observed class, both the actual and estimated distributions agree on the most-probable class. For that, NB has been used extensively in SA to evaluate Customer Satisfaction. Some recent studies document about its effectiveness (see, for instance Noori 2021; Khan and Zubair 2020; Xu et al. 2020).

The original NB classifier is a probabilistic classifier based on Bayes' theorem and thus resulting in a conditional probability model. Its implementation in the e1071 R package (Meyer et al. 2019) assumes independence of the predictor variables, and Gaussian distribution (given the target class) of metric predictors, whilst the NB classifier implemented in the KlaR R package (Weihs et al. 2005) extends the original specification to kernel estimated densities and allows the user to specify prior probabilities. When dealing with textual data, based on the assumption of word independence the main goal of NB is estimating the probabilities of categories given a text document by using the joint probabilities of words and categories. In view of that, we assume NB is applied for textual data composed of a huge amount of instances or observations so that, asymptotically, the observed classes of the response variable are consistent with the true (unknown) ones. In this setting, no a-priori specification either of the distribution of the metric predictors or of the response class probabilities is required, thus the proposed approach is completely data-driven. It results in a new version of the NB classifier, called *Threshold-based Naïve Bayes* (Tb-NB), that utilizes a data-driven decision rule to assign a new case to the most likely between two alternative classes. This decision rule is based on a threshold whose value is estimated from the training data.

Tb-NB can be applied when dealing with a labeled context in which each text corpora is composed, by its nature, of two components: one positive and one negative. In the case of Booking.com data, observed (textual) data are organized in a collection of n reviews or opinions about a product or a service. Notationally, the set of reviews $\mathcal{R}$ is split into a training set of size n_r, and a test set of size $n_{\mathcal{R}} - n_r$.

$$\mathcal{R} = \{r_1, \ldots, r_j, \ldots, r_n\}, \qquad j = 1, \ldots, n_r, (n_{\mathcal{R}} - n_r + 1), \ldots, n_{\mathcal{R}}.$$

Following a data cleaning step described in detail in Sect. 4.2, all the n_w words included in the $n_{\mathcal{R}}$ reviews are collected in a Bag-of-Words (BoW)

$$\mathcal{W} = \{w_1, \ldots, w_i, \ldots w_{n_w}\}, \qquad i = 1, \ldots, n_w.$$

In the case of the Booking.com data, each review r_j is composed of both a positive comment c_j^+ and a negative comment c_j^-, which are two sets of words with positive or negative meaning. At least one between these two sets of words is included in r_j, thus one element between c_j^+ and c_j^- might be an empty set $\emptyset$. The content of a review r_j is linked to $\mathcal{W}$ as follows

$$r_j = (c_j^+ \cup c_j^-) = \{w_1, \ldots, w_k, \ldots w_K\} \qquad \text{with } (w_1, \ldots, w_k, \ldots w_K) \in \mathcal{W}$$

Considering a probability function $\pi(\cdot)$, Tb-NB builds on the Bayes' rule and computes a scoring function $\Lambda(\cdot)$ for all the n_r reviews included in the training set in order to predict, as accurately as possible, if a word w_k included in review r_j $(j = 1, \ldots, n_r)$ has a negative or positive sentiment.

Specifically, for a given review r_j we define $\Lambda(w_k|(c_j^+, c_j^-) \in r_j)$ as the log-odds ratio of the probability that a comment c_j is positive given that it includes a certain word w_k, that is $\pi(c_j^+|w_k)$ over the probability that c_j is negative given that it includes w_k, denoted as $\pi(c_j^-|w_k)$. Thus, the log-odds ratio Λ for a word w_k in a review r_j $(j = 1, \ldots, n_r)$ is

$$
\begin{aligned}
&\Lambda\left(w_k|(c_j^+, c_j^-) \in r_j\right) \\
&= \log\left[\frac{\pi(c_j^+|w_k)}{\pi(c_j^-|w_k)}\right] \\
&= \log\left[\frac{\pi(w_k|c_j^+)}{\pi(w_k|c_j^-)} \cdot \frac{\pi(\bar{w}_k|c_j^+)}{\pi(\bar{w}_k|c_j^-)} \cdot \frac{\pi(c_j^+)}{\pi(c_j^-)}\right] \\
&= \underbrace{\left[\log\pi(w_k|c_j^+) - \log\pi(w_k|c_j^-)\right]}_{\mathcal{L}(w_k)} + \underbrace{\left[\log\pi(\bar{w}_k|c_j^+) - \log\pi(\bar{w}_k|c_j^-)\right]}_{\mathcal{L}(\bar{w}_k)} \\
&\quad + \left[\log\pi(c_j^+) - \log\pi(c_j^-)\right] \\
&\approx \mathcal{L}(w_k) + \mathcal{L}(\bar{w}_k)
\end{aligned} \tag{1}
$$

$\Lambda(w_k)$ derives from the sum of two components: a function $\mathcal{L}(w_k)$ that measures how likely a specific word w_k is present in a text corpora, and a function $\mathcal{L}(\bar{w}_k)$ that measures how likely w_k is not present in the same text. These two functions derive from the log-likelihood ratio of the event $(w_k \in r_j)$ and $(w_k \notin r_j)$, respectively. The term $\left[\log\pi(c_j^+) - \log\pi(c_j^-)\right]$ in Eq. 1 is discarded as it is constant for all the words included in $\mathcal{W}$. It corresponds to the proportions of observed positive (negative) comments in the set of reviews $\mathcal{R}$.

The log-likelihood scores can be implemented for a whole review, as well as by limiting the analysis to the single (positive or negative) comment in order to predict its polarity.

In the first case, Eq. (1) allows us to understand if a whole review r_j $(j = 1, \ldots, n_r)$ has a negative or a positive sentiment by computing the scoring function $\Lambda(w_k)$ for all the K words included in its content:

$$
\begin{aligned}
\Lambda\left(r_j\right) &= \Lambda\left(w_1, \ldots, w_k, \ldots w_K\right) = \sum_{k=1}^{K} \Lambda\left(w_k|(c_j^+, c_j^-) \in r_j\right) \\
&= \sum_{k=1}^{K} \mathcal{L}(w_k) + \mathcal{L}(\bar{w}_k)
\end{aligned} \tag{2}
$$

In the second case, it is possible to to assess if a comment $c_j = c_j^+ \cup c_j^-$ $(j = 1, \ldots, n_c)$ has a negative or a positive sentiment by computing the scoring function $\Lambda(w_m)$ for all the M words included in its content:

$$\begin{aligned}\Lambda\left(c_j\right) &= \Lambda\left(w_1, \ldots, w_m, \ldots w_M\right) = \sum_{m=1}^{M} \Lambda\left(w_m | (c_j^+, c_j^-) \in c_j\right) \\ &= \sum_{m=1}^{M} \mathcal{L}(w_m) + \mathcal{L}(\bar{w}_m) \end{aligned} \tag{3}$$

with $(w_1, \ldots, w_m, \ldots w_M) \in c_j \in \mathcal{W}$. In this case, the training set is composed of n_c comments (being part of the entire set of $\mathcal{C}$ comments) for which the supervised learning task is predicting their polarity. Thus, Tb-NB proceeds by computing the log-odds ratio $\Lambda(w_k)$ (Eq. 1) for all the words $w_k \in \mathcal{W}$ and next it aggregates the quantities $\Lambda(w_k)$ computed with respect to the set of words included in each review r_j $(j = 1, \ldots, n_r)$ based on Eq. 2, or the set of words included in each comment c_j $(j = 1, \ldots, n_c)$ based on Eq. 3.

Once the set of scores $\Lambda\left(r_j\right)$, or $\Lambda\left(c_j\right)$, is computed, a decision rule $\mathcal{D}$ has to be defined in order to classify the review r_j^* included in the test set $(j^* = n_\mathcal{R} - n_r + 1, \ldots, n_\mathcal{R})$, or the comment c_j^* included in the test set $(j^* = n_\mathcal{C} - n_c + 1, \ldots, n_\mathcal{C})$, as positive or negative. The decision rule $\mathcal{D}$ is defined based on the estimated value of the threshold parameter τ corresponding to a specific value of the log-odds ratio $\Lambda(\cdot)$ that allows us to classify a review r_j^*, or a comment c_j^*, as positive (+1) or negative (-1). For example, the decision rule $\mathcal{D}$ for r_j^* is defined as

$$\mathcal{D}_{r_j^*}: \begin{cases} \Lambda\left(r_j^*\right) > \hat{\tau} & \rightarrow r_j^* = -1 \\ \Lambda\left(r_j^*\right) \leq \hat{\tau} & \rightarrow r_j^* = +1 \end{cases} \qquad (j^* = n_\mathcal{R} - n_r + 1, \ldots, n_\mathcal{R}) \tag{4}$$

whilst it is defined in the same way for a comment by replacing r_j^* with c_j^*, and n_r with n_c, in Eq. 4.

The threshold τ is the unique parameter of the Tb-NB classifier, which is estimated from the training data. Dealing with a binary classification problem, τ is specified alternatively as the value minimizing the Type I error, the Type II error or both errors depending on the peculiarities of the specific classification problem and the objectives of the analysis (e.g. in Sect. 4.3 we minimize them both). In any case, it is estimated by applying k-fold cross-validation on the n_r (n_c) reviews (comments) included in the learning set.

If the reference text corpora is the set of observed comments, the polarity of each comment is known a-priori, thus the distributions of the Type I and Type II errors are defined based on the different values of the log-odds ratio $\Lambda(c_j)$ obtained applying the Tb-NB classifier with prior probabilities corresponding to the observed classes (proportions of positive or negative comments).

Whereas, when the reference text corpora is the set of reviews the ground truth is unknown. In this case, the unsupervised experiment is turned into a supervised

learning experiment and some a-priori information about the polarity of a text corpora is retrieved from external sources or from data pre-processing. The latter includes, for example, context-based word embeddings that incorporates the sentiment polarity information from (external) labeled corpora (Yu et al. 2018).

4 Analyzing reviews on Booking.com with Tb-NB

The various steps of the analysis of the Booking.com data through Tb-NB are represented with a flowchart in Fig. 1. The original reviews are scraped and collected in a specific dataset to which a data cleaning process is applied in order to train the Tb-NB classifier on cleaned data. Tb-NB produces sentiment scores for each review that are next used in a post-hoc analysis to show how these scores vary with respect to specific hotel category, location of the hotel and/or different periods. In this paper, we apply the Tb-NB classifier to predict the sentiment associated with the reviews containing opinions about all the Sardinian hotels offering accommodations on Booking.com. We hereby describe the main feature of collected data, the data cleaning process, and the performance of the proposed classifier in comparison with that of other possible competitors.

4.1 Data collection

Hotel guests who reserved through Booking.com are asked to leave a review about their accommodation experience. The review is structured in two parts: a positive comment and/or a negative one. Two examples of these kind of reviews on Booking.com are shown in Fig. 2.

We have collected data about Booking.com's reviews through web-scraping. A Python extractor has been implemented to retrieve all the valuable information that is publicly available on the platform once the user has specified a specific destination

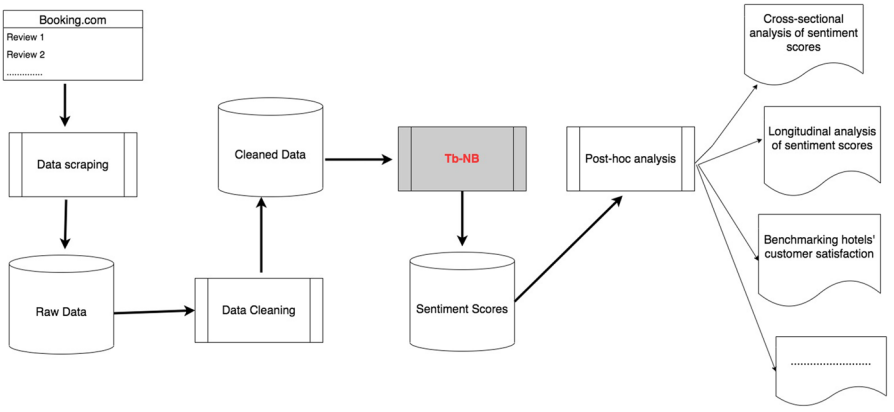

Fig. 1 Flowchart summarizing the analysis of the Booking.com data with Tb-NB

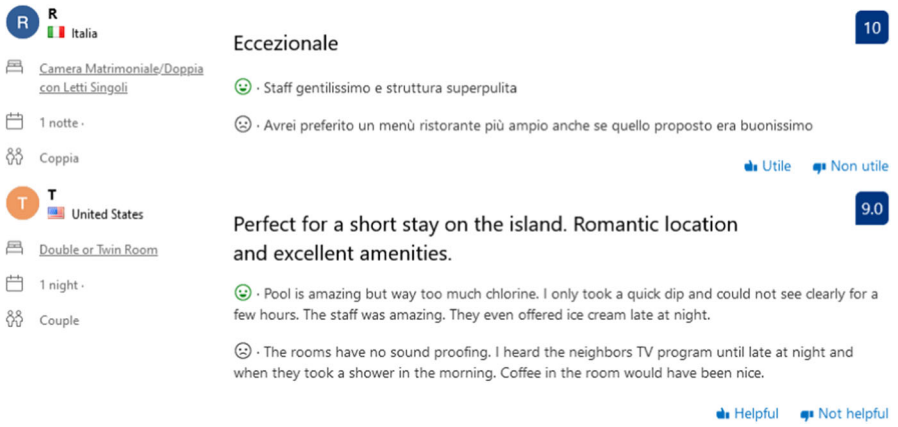

Fig. 2 Two examples of reviews on Booking.com, one in Italian and one in English. For each review, the first comment (☺) is positive whilst the second one (☹) is negative. A reviewer might decide to leave just one of them. A review is the union of the two comments provided by a reviewer

and a range of dates. Such an extractor rely on three main libraries: Requests, that allows us to send HTTP/1.1 requests extremely easily;[1] BeautifulSoup, that allows us to scrape information from web pages;[2] and Parallel, that allows the two previous libraries to work in parallel.[3]

Retrieved data have been next organized into flat tables. They concern 619 hotels operating in Sardinia. For them, it was possible to scrape 66,237 reviews consisting of 106,800 comments in Italian (86.14%) or English (13.86%) collected from January 3, 2015 to May 27, 2018 (1,240 days). Of these, 62,291 are positive comments and 44,509 are negative comments.

Data have been next organized into two datasets including 127 features in total. The first dataset (Hotels dataset) includes information about the hotel. Specifically, it concerns some peculiarities of the hotel (3 features), information about the reviews (8 features) and the reviewer (2 features), the scores assigned by Booking.com (11 features) and their components (12 features), guests (8 features), characteristics of the accommodation (32 features), length of stay (6 features) and other information (4 features). An example of data contained in this dataset is shown in Table 1.

The second dataset (Reviews dataset) includes information about the specific review. In particular, it includes information about the hotel (2 features), the content of the review and the comments (positive and/or negative, 6 features), the reviewer (2 features), the Booking.com's score components associated with the specific review (6 features), the guest (4 features), the type of accommodation (16 features), the length of stay (3 features) and other information (2 features). An example of data contained in this dataset is shown in Table 2.

[1] https://pypi.org/project/requests/.

[2] https://pypi.org/project/beautifulsoup4/.

[3] https://joblib.readthedocs.io.

Table 1 Hotels data: $n = 619$

Name	Type	Postal code	City	Reviews	Positive comments	Negative comments	⋯
Hotel 1	Other Facilities	09044	Sant'Isidoro	35	35	19	⋯
Hotel 2	3–Star	09049	Villasimius	289	286	104	⋯
Hotel 3	3–Star	07013	Mores	125	123	42	⋯
Hotel 4	4–Star	09123	Cagliari	725	678	492	⋯
⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯
⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯
Hotel 619	4–Star	07026	Olbia	2147	1975	1545	⋯

Table 2 Reviews data: 106,800 comments from 66,237 reviews

Name	id. comment	id. review	Text	Neg or Pos	Booking score	Business	Length of stay	⋯
Hotel 1	1	1	christina was the best...	Pos	10.0	Yes	1–3	⋯
Hotel 1	2	2	we travelled into cagliari...	Pos	9.2	No	4–7	⋯
Hotel 1	3	3	it was fantastic...	Pos	10.0	No	> 7	⋯
⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯
⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯
Hotel 619	106,800	66,237	il wifi e le zanzariere non erano presenti...	Neg	5.0	Yes	4–7	⋯

The complete list of all variables included in the two datasets is reported in the "Appendix". An anonymized random sample of the full dataset is available on the Github repository.[4]

4.2 Data cleaning

The just-downloaded raw data is usually not suitable for the analysis. It has many unnecessary words like stop words that do not explain the meaning of the sentence as well as acronyms whose meaning is difficult to decipher and hence tend to confuse the algorithm. Moreover, it contains emojis which have helpful information, so they have to be converted into meaningful text. Data cleaning is a basic step for preprocessing the data and make it usable for the analysis. Below mentioned are the details of all the subsequent steps used in data cleaning for every single observation in the dataset.

1. *Preprocessing*: a basic—but necessary—filtration is done before moving to the next step. It consists in removing links, especially partial ones, acronyms since their meaning is difficult to decipher, and recurrent and meaningless keywords like RT (re-tweets), @username, uninterested #hashtags, etc.;
2. *Emoticons conversion*: such valuable information, especially regarding the sentence sentiment polarity, is contained inside emoticons—like:-) or:-( )—and in emojis—like ☺ or ☹. In order to consider them in the same way, emoticons are converted into emojis;
3. *Emoji replacement*: Once all emoticons have become emojis, the next step is to replace the emojis with their corresponding meaning so that they can be further treated and analyzed together within the normal text. In that way, all the meaningful symbols are now converted into the text;
4. *Stop words & alphanumeric characters processing*: the incoming text is first tokenized into separate words, and any punctuation adjacent to the words is also separated. Thereafter, these punctuation symbols, along with some alphanumeric characters that might be present, are detected and removed. Cases of all alpha-

[4] https://shorturl.at/fix58.

Table 3 Example of a threshold-based Naïve Bayes output

	w_1	w_2	w_3	w_4	w_5	$\cdots$
$\pi(w_k\|c_j^-)$	0.011	0.026	0.002	0.003	0.003	$\cdots$
$\pi(w_k\|c_j^+)$	0.007	0.075	0.005	0.012	0.001	$\cdots$
$\mathcal{L}(w_k)$	0.411	−1.077	−1.006	−1.272	1.423	$\cdots$
$\mathcal{L}(\bar{w}_k)$	−0.004	0.052	0.003	0.008	−0.002	$\cdots$

bets are normalized to lowercase. Next, stop words like "a", "the", "do", "to" are removed from data as they do not provide any valuable information about the sentiment deriving from a specific text. However, consistent with Chai (2019) and Morante and Blanco (2021), negative words like "not" are kept as they completely alter the meaning of a sentence and profoundly impact its sentiment;

5. *stemming*: in the last data cleaning phase, the tokens are stemmed; in other words, they are reduced to their root or base form. For example, "fishing," "fished," "fisher" are all reduced to the stem "fish". In that way, words that are related to the same topic by their root or base form are merged.

4.3 Tb-NB results and post-hoc analysis

We apply the Tb-NB classifier to the cleaned Booking.com data to evaluate its accuracy and to show its usefulness in a post-hoc analysis of the results arising from it.

Following the basic steps of Tb-NB, described in Sect. 3, we compute the log odds ratio (Eq. 1) for each word w_k included in the Bag-of-Words $\mathcal{W}$ obtained after data cleaning as well as for each comment c_j (Eq. 3). An example of the values of some of the components of the score function $\Lambda(c_j)$ specified in Eq. 3 is shown in Table 3.

We apply fivefold cross-validation to estimate the threshold parameter τ on the entire set of 106,800 comments through the decision rule specified in Eq. 4. We estimate $\Lambda(c_j)$ for the observations included in the original data but not in the considered kth fold ($k = 1, \ldots, 5$) and compute the Misclassification Error (ME) for observations included in the kth fold. The estimated τ is chosen as the one minimizing at the same time both the Type I and Type II errors, as both errors (classifying a comment as positive when it is negative, or vice versa) are considered as equally important in this particular type of analysis. As shown in Fig. 3, the estimated τ is $\hat{\tau} = 1.138$, as this value is that minimizing simultaneously both the Type I and the Type II errors.

Setting the classification rule introduced in Eq. 4 as $\mathcal{D}_{c_j}$, with $\hat{\tau} = 1.138$ Tb-NB is able to classify correctly 91.1% of the out-of-fold instances.

Thus, the Tb-NB classifier performs well in classifying a comment as positive or negative. Moreover, the versatile nature of the values produced by the scoring function $\Lambda(w_k)$ helps in the interpretation of the results of the analysis. To this purpose, values of $\Lambda(w_k)$ can be aggregated together based on some specific criteria. The criterion that Tb-NB utilizes to merge values of $\Lambda(w_k)$ when classifying the out-of-fold observations is, for a given set of words included in a comment c_j ($c_j \in \mathcal{W}$), the aggregation of the values of $\Lambda(w_k)$ checking if w_k belongs (or not) to a positive comment (c_j^+) and/or

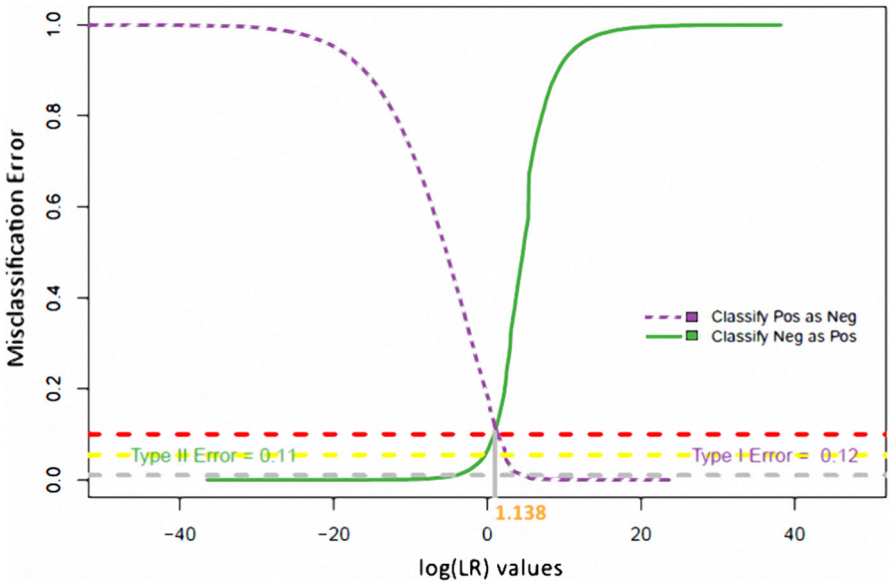

Fig. 3 Estimation of the threshold parameter τ for the Booking.com data ($\log(LR)$ is the value of the log-odds ratio $\Lambda(c_j)$ specified in Eq. 3)

to a negative comment (c_j^-) included in a review r_j. Thus, the main driver of this aggregation criterion is the out-of-fold prediction accuracy.

Alternatively, different aggregation criteria can be considered. One interesting possibility is the definition of some reference categories, or macro-words, and the computation of the scoring function Λ for each category. With $\mathcal{H}$ categories available, the scoring function $\Lambda(h)$ ($h = 1, \ldots, \mathcal{H}$) is computed aggregating the values of $\Lambda(w_j)$ for all the words w_j belonging to the hth category. There are many ways to identify the most informative categories. For example, it is possible to consider user-defined categories, categories identified by some context-domain knowledge or retrieved in literature. In any case, categories are associated with subsets of words having the same meaning. For instance, with the Booking.com data, the words "breakfast", "restaurant", "lunch", etc. all belong to the "food" category. Proceeding in this way, we consider words with similar meanings and manually assign them to macro-words, each one corresponding to a category. These macro-words include all the words composing the Bag-of-Words ($\mathcal{W}$). The set of reference categories identified for the Booking.com data is: "cleaning", "comfort", "position", "price-quality-rate", "services", "staff", "wifi", "bar", "food", "hotel", "room", "sleep-quality", and "other". These categories have been identified starting from the set of original categories available on the Booking.com website and adding new ones ("bar", "food", "hotel", "room", "sleep-quality", and "other") observing that many comments report some characteristics of the service not considered in the original categories. More specifically, we consider separately the bar and restaurant (food) service, the other room services besides cleaning and comfort (i.e., for example, the toilet service) that are assigned to the "room" category,

and the other features of a hotel beside those specifically mentioned (assigned to the "hotel" category), the quality of sleeping as it is one of the most discussed topics and, eventually, the category "other" as the residual one.

Once categories are identified, it is possible to compute the value of the scoring function Λ for each category focusing the analysis on a specific hotel, as well as on a specific hotel category or a specific destination that includes all the hotels located in a certain area. These disaggregated analyses allow us to understand the strengths and weaknesses of a specific hotel, a hotel category, or of a destination, respectively. These kinds of assessments can be done both cross-sectionally or longitudinally.

As an example, to better highlight the information content of these categories, Fig. 4 shows where the hotels are located in the Sardinian region (Fig. 4, top panel) together with the average overall score Λ obtained from the Tb-NB classifier in the period January 3, 2015, to May 27, 2018 (Fig. 4, bottom panel). Data are colored with different intensities of red or blue based on the 8 districts that comprise the Sardinian region. Recalling that the more positive the score obtained for the overall sentiment the more positive the clients' satisfaction, Fig. 4 (bottom panel) shows the values of the scoring function Λ corresponding to different satisfaction levels. This kind of representation immediately gives an idea of the geographical distribution of the average overall satisfaction of hotels' guests: it is possible to notice that the districts of Cagliari, Olbia Tempio, and Sassari are those having the highest number of hotels but the highest average clients' satisfaction is observed in the districts of Carbonia-Iglesias, Nuoro, and Ogliastra.

Moreover, since reviews occur in different time occasions it is possible to consider time series data obtained by computing longitudinally the scoring function $\Lambda(\cdot)$ for each category h in order to assess how the quality of a specific service offered by hotel(s) changes in time. To demonstrate the information content obtainable from this kind of aggregation criterion, the values of the scoring function $\Lambda(\cdot)$ computed in the period February 1, 2018 to May 27, 2018 are plotted in Fig. 5 for all the above-mentioned categories with regard to a Sardinian hotel that has obtained the maximum score (10 points) on the Booking.com website.

For each line represented in the plot, the higher the score the more positive is the sentiment and vice versa. The line in black is the overall sentiment score. Despite the maximum score assigned by Booking.com, observing the plotted values of the scoring function $\Lambda(\cdot)$ for the different categories it is possible to notice that the quality of the food services offered by this hotel is decreasing in time. Likewise, the same kind of plot is shown for all the hotels located in the district of Cagliari (Fig. 6, top panel) and in that of Sassari (Fig. 6, bottom panel) with respect to the whole set of reviews collected in the period January 3, 2015 to May 27, 2018.

Comparing trends observed for the different service categories in the two districts, as well as in the overall score, it is possible to notice how the quality of each service offered by hotels in the two districts evolves in time.

Another useful option is using the output obtained from Tb-NB as input for another auxiliary model. For this purpose, we have used the polarity (positive or negative) estimated with Tb-NB for each review r_j, together with the values of the components $\mathcal{L}(w_i \in h)$ and $\mathcal{L}(\bar{w}_i \in h)$ of the scoring function $\Lambda(\cdot)$ computed for all the words belonging to the previously discussed macro-words or aggregated categories ($h =$

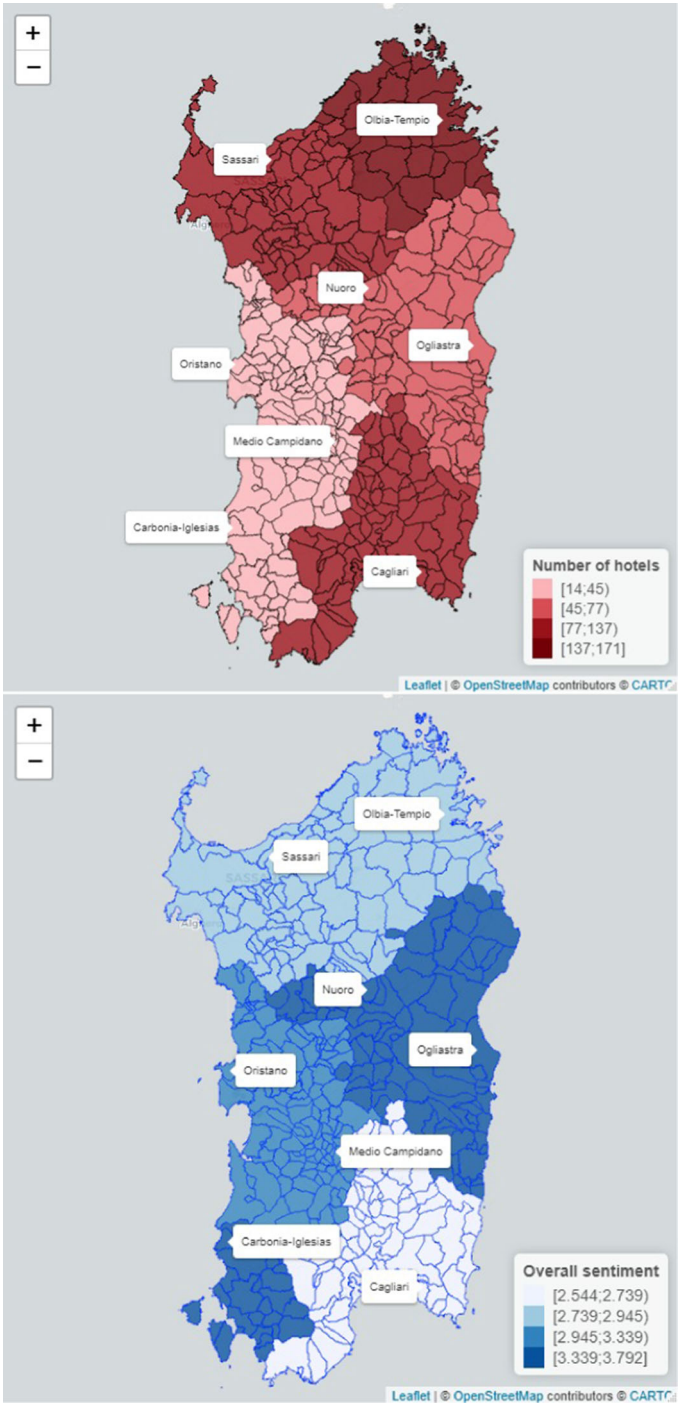

Fig. 4 The number of hotels in the 8 districts of the Sardinian region (Top panel) and the average clients' satisfaction observed in the period January 3, 2015, to May 27, 2018 (bottom panel)

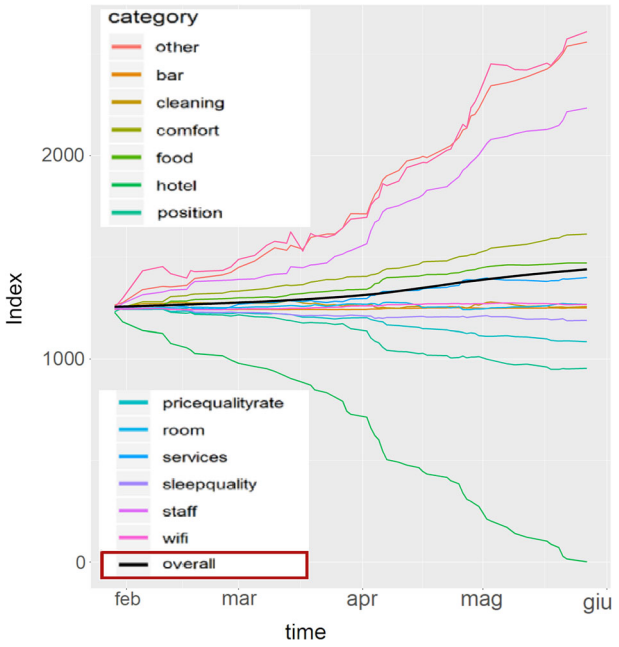

Fig. 5 Time-series of the scoring function $\Lambda(\cdot)$ for each category of services offered by a hotel scored 10/10 on Booking.com

$1, \ldots, \mathcal{H}$) as predictors in a regression model aimed at predicting the official score assigned by Booking.com to hotels, namely a response variable defined in [0, 10]. We compare the MSE obtained by the model using this set of predictors to that of the model including as predictors all the words included in the Bag-of-Words $\mathcal{W}$. In this case, the best performing model is Random Forest but, importantly, MSE is reduced of 13.60% when using the model including only the polarity and the scores of the categories as predictors compared to the case of the most extended model. Of course, the benefits in terms of computing time are also relevant, as the reduced model runs in about 9 min with a 4.5 Ghz Exacore Processor with 16 Gb of RAM, whilst the complete models runs in more than 3 h with the same machine. Since Booking.com provides scores for each service offered by a hotel, the same prediction model can be applied for each service, thus obtaining predicted scores arising from reviews' content for each hotel service. Anyway, considering the log-likelihood ratios of categories (originated by merging similar words included in the BoW), this simple experiment demonstrates that using them as inputs for an auxiliary model, in place of the single words ($w_k \in \mathcal{W}$), considerably improves the prediction of such an auxiliary model.

5 Benchmarking Tb-NB

The performance of Tb-NB is compared with that of other well-known classifiers, in particular: Logistic Regression (LOG), Random Forest (RF), standard Naïve Bayes

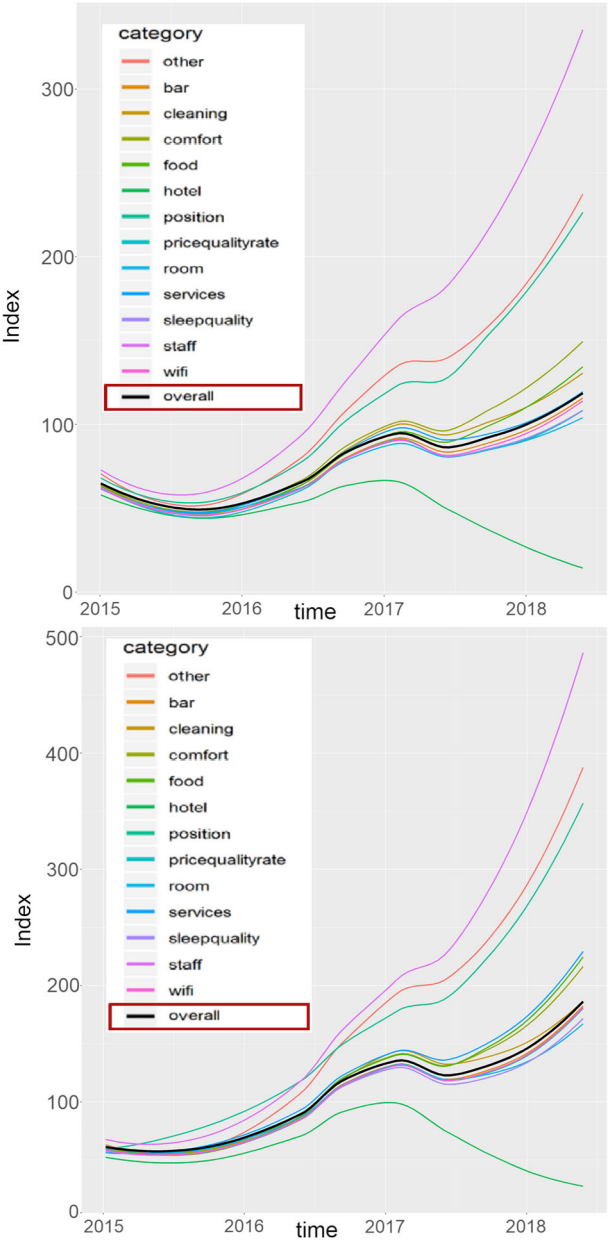

Fig. 6 Time-series of the scoring function Λ for each category of services offered by hotels located in the district of Cagliari (top panel) and in that of Sassari (bottom panel)

(NB E1071), Naïve Bayes using kernel estimated densities (NB KlaR), Decision Trees (CART), Linear Discriminant Analysis (LDA), and Support Vector Machine (SVM). Comparisons are based on three main factors: prediction accuracy, noise resistance,

and computational efficiency. All of these aspects are analyzed more in detail in what follows.

5.1 Accuracy

As anticipated in Sect. 4.3, Tb-NB is able to correctly estimate 91.1% of observed comments. For this data, we compare the performance of Tb-NB with that of competitors using diverse classification performance metrics: accuracy, sensitivity, fall-out, F1 score, and Matthews' correlation coefficient (see Chicco and Jurman 2020 for a comparison). Results are summarized in Table 4. The Threshold-based Naïve Bayes classifier performs considerably better than competitors as it provided a Matthews correlation coefficient (Accuracy) of 0.813 (0.9111) versus an average value of 0.508 (0.805) obtained from the alternatives. The superiority of Tb-NB over competitors is also evident if we compare graphically the accuracy of different classifiers through the ROC curves (Fig. 7).

To further enforce the validity of the results obtained on the whole dataset we consider how classifiers accuracy varies when changing the dataset size as well as the size of both training and test sets. To this purpose, we re-estimate the classifiers repeating the analysis several times based on the following factors:

(a) The total sample size n: 20,000; 50,000 and 100,000;
(b) The training-test set proportions: 50–50, 67–33, 80–20;
(c) The selection of comments with more than three words only.

The last factor is considered because preliminary trials seem to indicate that removing short comments (up to three words) improves the classification accuracy of Tb-NB. For each combination of total sample size × training-test set proportions × elimination of short comments we estimate the different classifiers 100 times on resampled versions of the original data and compute the performance metrics reported in Table 5. For the sake of brevity, Table 5 reports results for the 80–20 cases only. Results obtained for the other training-test set proportions are very similar to those of Table 5 and are reported in the "Appendix". Results reported in Table 5 indicate that, although

Table 4 Performance metrics on raw data using fivefold cross validation

Classifier	ACC	Sensitivity	Fall-out	F1	MCC
Tb-NB	**0.911**	0.929	**0.117**	**0.926**	**0.813**
LOG	0.850	0.884	0.532	0.877	0.361
RF	0.811	0.873	0.591	0.849	0.303
NB(E1071)	0.806	0.804	0.389	0.834	0.390
NB(KLAR)	0.806	0.804	0.389	0.834	0.390
CART	0.768	0.842	0.587	0.815	0.272
LDA	0.764	0.860	0.641	0.816	0.246
SVM	0.793	**0.930**	0.290	0.771	0.621
Average	0.805	0.893	0.377	0.810	0.508

Best values of performance metrics reported in bold
ACC accuracy, *F1* F1-score, *MCC* Matthews correlation coefficient

Table 5 Benchmarking Tb-NB: performance metrics with the training-test set proportion 80–20

n		20.000					50.000					100.000				
	Classifier	ACC	TPR	TNR	F1	$\overline{rk}$	ACC	TPR	TNR	F1	$\overline{rk}$	ACC	TPR	TNR	F1	$\overline{rk}$
LEARNING SET (80%)	Tb-NB	0.878	0.910	0.839	0.894	4.75	0.879	0.915	0.833	0.894	4.00	0.878	0.917	0.830	0.893	4.75
	Tb-NB*	(0.910)	(0.929)	(0.881)	(0.925)	(4.50)	(0.911)	(0.930)	(0.882)	(0.926)	(4.25)	(0.911)	(0.930)	(0.882)	(0.926)	(4.50)
	NB(KLAR)	0.866	0.938	0.790	0.878	5.00	0.867	0.937	0.793	0.879	5.50	0.866	0.936	0.793	0.878	5.50
	NB(KLAR)*	(0.901)	(0.946)	(0.844)	(0.915)	(5.00)	(0.899)	(0.945)	(0.840)	(0.913)	(5.00)	(0.900)	(0.945)	(0.841)	(0.914)	(5.00)
	NB(E1071)	0.866	0.938	0.790	0.878	5.00	0.867	0.937	0.793	0.879	5.50	0.867	0.936	0.793	0.879	5.25
	NB(E1071)*	(0.901)	(0.946)	(0.844)	(0.915)	(5.00)	(0.899)	(0.945)	(0.840)	(0.913)	(5.00)	(0.900)	(0.945)	(0.841)	(0.914)	(5.00)
	RF	**0.939**	0.945	0.930	**0.948**	**1.50**	**0.940**	**0.948**	**0.928**	**0.948**	**1.00**	**0.939**	**0.949**	**0.925**	**0.947**	**1.00**
	RF*	(**0.958**)	(**0.963**)	(**0.950**)	(**0.965**)	(**1.00**)	(**0.946**)	(0.953)	(**0.937**)	(**0.954**)	(**1.25**)	(**0.946**)	(**0.955**)	(**0.932**)	(**0.953**)	(**1.00**)
	SVM	0.858	0.929	0.784	0.871	6.50	0.874	0.923	0.816	0.888	4.50	0.885	0.920	0.841	0.899	3.50
	SVM*	(0.878)	(0.940)	(0.803)	(0.893)	(6.50)	(0.891)	(0.940)	(0.829)	(0.906)	(6.50)	(0.899)	(0.939)	(0.846)	(0.913)	(6.00)
	CART	0.780	0.890	0.684	0.790	7.75	0.777	0.891	0.680	0.786	8.00	0.776	0.893	0.678	0.785	8.00
	CART*	(0.808)	(0.869)	(0.734)	(0.834)	(8.00)	(0.809)	(0.869)	(0.734)	(0.835)	(8.00)	(0.810)	(0.875)	(0.731)	(0.835)	(8.00)
	LDA	0.903	0.920	0.881	0.917	3.00	0.902	0.919	0.878	0.915	3.00	0.901	0.919	0.877	0.915	3.00
	LDA*	(0.916)	(0.939)	(0.883)	(0.929)	(3.50)	(0.914)	(0.939)	(0.879)	(0.928)	(3.75)	(0.914)	(0.939)	(0.878)	(0.927)	(3.50)
	LOG	0.922	**0.960**	**0.960**	0.933	**1.50**	0.875	0.926	0.815	0.886	4.50	0.873	0.925	0.814	0.885	4.75
	LOG*	(0.922)	(**0.960**)	(0.872)	(0.933)	(2.50)	(0.916)	(**0.957**)	(0.862)	(0.928)	(2.25)	(0.918)	(0.937)	(0.891)	(0.932)	(3.00)

Table 5 continued

n		20.000					50.000					100.000				
	Classifier	ACC	TPR	TNR	F1	$\overline{rk}$	ACC	TPR	TNR	F1	$\overline{rk}$	ACC	TPR	TNR	F1	$\overline{rk}$
TEST SET (20%)	Tb-NB	0.867	0.926	0.804	0.880	4.00	0.869	0.929	0.804	0.882	4.50	0.867	0.931	0.800	0.880	4.50
	Tb-NB*	(0.859)	(0.941)	(0.780)	(0.868)	(3.75)	(0.840)	(**0.945**)	(0.751)	(0.845)	(4.00)	(0.755)	(0.948)	(0.667)	(0.709)	(4.50)
	NB(KLAR)	0.862	**0.934**	0.787	0.874	4.50	0.865	**0.936**	0.790	0.878	5.25	0.867	**0.936**	0.793	0.879	5.75
	NB(KLAR)*	(0.853)	(**0.943**)	(0.767)	(0.863)	(4.50)	(0.837)	(0.941)	(0.747)	(0.841)	(5.50)	(0.741)	(**0.951**)	(0.652)	(0.686)	(5.75)
	NB(E1071)	0.862	**0.934**	0.787	0.874	4.50	0.865	**0.936**	0.790	0.878	5.50	0.866	**0.936**	0.793	0.878	5.25
	NB(E1071)*	(0.853)	(**0.943**)	(0.767)	(0.863)	(4.50)	(0.837)	(0.941)	(0.747)	(0.841)	(5.50)	(0.741)	(**0.951**)	(0.652)	(0.686)	(5.75)
	RF	0.878	0.876	**0.882**	0.898	3.75	0.891	0.895	**0.886**	0.908	3.50	0.896	0.902	**0.887**	0.912	3.50
	RF*	(0.878)	(0.891)	(**0.861**)	(0.895)	(**3.00**)	(0.877)	(0.892)	(**0.857**)	(**0.890**)	(**2.75**)	(**0.846**)	(0.884)	(**0.811**)	(**0.848**)	(**2.75**)
	SVM	0.855	0.926	0.780	0.868	6.00	0.872	0.922	0.814	0.887	4.00	0.884	0.919	0.839	0.898	4.00
	SVM*	(0.842)	(0.940)	(0.752)	(0.850)	(6.25)	(0.845)	(0.940)	(0.759)	(0.852)	(4.00)	(0.793)	(0.930)	(0.710)	(0.771)	(4.00)
	CART	0.776	0.887	0.680	0.786	7.75	0.776	0.890	0.679	0.786	8.00	0.776	0.894	0.678	0.785	8.00
	CART*	(0.786)	(0.873)	(0.703)	(0.800)	(8.00)	(0.779)	(0.876)	(0.694)	(0.788)	(8.00)	(0.736)	(0.903)	(0.654)	(0.694)	(6.75)
	LDA	0.890	0.908	0.865	0.905	3.00	0.897	0.914	0.872	0.911	3.25	0.898	0.917	0.874	0.912	3.25
	LDA*	(**0.881**)	(0.926)	(0.828)	(0.893)	(**3.25**)	(0.875)	(0.930)	(0.816)	(0.886)	(3.50)	(0.824)	(0.919)	(0.754)	(0.815)	(2.75)
	LOG	**0.891**	0.908	0.869	**0.907**	**2.50**	**0.901**	0.916	0.879	**0.915**	**2.25**	**0.903**	0.919	0.881	**0.916**	**2.25**
	LOG*	(**0.881**)	(0.921)	(0.833)	(**0.894**)	(2.75)	(**0.878**)	(0.926)	(0.825)	(**0.890**)	(2.75)	(0.827)	(0.916)	(0.760)	(0.820)	(3.75)

ACC accuracy, *TPR* true positive rate, *TNR* true negative rate, *F1* F1-score, $\overline{rk}$ average rank of the performance metrics ACC, TPR, TNR, and F1
Values in parenthesis are the performance metrics obtained considering reviews including comments with more than 3 words only. The corresponding classifier is marked with a "*". Best values of performance metrics are reported in bold

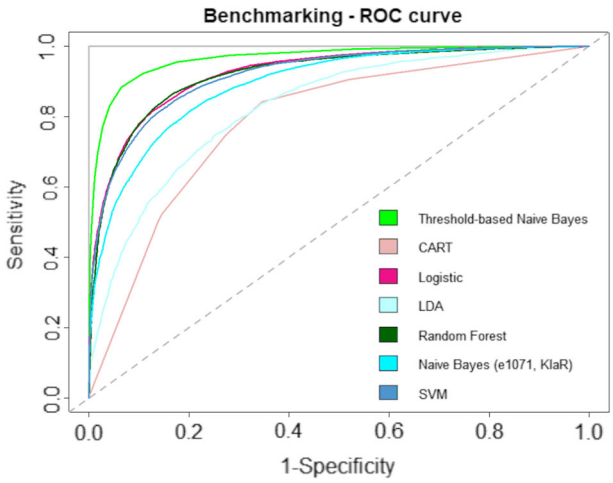

Fig. 7 ROC curve for benchmarking of the Threshold-based Naïve Bayes Classifier

Tb-NB is never the best nor the worst performing classifier, it provides values of classification accuracy metrics that are in line with those obtained by other classifiers. This finding enforces the strength of Tb-NB as it probably provides the most easily interpretable and usable output, as demonstrated in Sect. 4.3.

5.2 Noise resistance

In sentiment analysis Natural Language text comments are usually classified into *positive* or *negative*. Thus, the response variable is usually a binary response $y = \{-1, +1\}$. A classifier is more and more accurate as long as the two conditional distributions $\mathbf{X}|(y = -1)$ and $\mathbf{X}|(y = +1)$ are well separated. If this separation is not so evident even a complex classifier cannot be very accurate in estimating the polarity of Natural Language text comments. In this framework, we evaluate the noise resistance of compared classifiers as their ability to be as much as possible accurate when the two above-mentioned conditional distributions have some degree of overlapping. To make the two distributions overlap we artificially inject some noise into the original data. Noise injection is obtained by scrambling the label of the response variable, from positive to negative or vice versa, while keeping the content of the words unchanged in a subset of observed comments. Next, the resistance of a classifier is evaluated as its ability to provide good classification performance metrics in the perturbed dataset, that is a dataset composed of a proportion of original unperturbed data and a proportion of perturbed data. In our experiments, we compare the performance of the different classifiers while varying the proportion of perturbed data from 1 to 50% of the sample size.

Results of these resistance tests are reported in Table 6 and Figs. 8 and 9. Table 6 reports the values of the classification performance metrics in the case the percentages of perturbed data are equal to 0%, 25%, 33%, and 50%, respectively.

Table 6 Classifiers' resistance—performance metrics obtained by cross-validation for each percentage of perturbed data

Classifier	Perturbed data (%)	ACC	TPR	TNR	F1	MCC	$\overline{rk}$
Tb-NB	0	0.911	0.930	0.882	0.926	0.814	5.2
NB(KLAR)		0.899	**0.947**	0.838	0.913	0.796	5.6
NB(E1071)		0.899	**0.947**	0.838	0.913	0.796	5.6
RF		**0.953**	**0.963**	**0.939**	**0.961**	**0.903**	1.0
SVM		0.899	0.939	0.846	0.913	0.793	6.2
CART		0.810	0.875	0.731	0.835	0.616	7.8
LDA		0.914	0.939	0.878	0.927	0.821	4.4
LOG		0.918	0.937	0.891	0.932	**0.830**	3.6
TB-NB	25	**0.909**	0.912	**0.903**	**0.926**	**0.808**	3.0
NB(KLAR)		0.878	**0.964**	0.786	0.891	0.766	5.2
NB(E1071)		0.878	**0.964**	0.786	0.891	0.766	5.2
RF		0.832	0.880	0.769	0.857	0.655	7.6
SVM		0.886	**0.957**	0.806	0.900	0.777	4.2
CART		0.685	0.915	0.565	0.665	0.465	8.6
LDA		0.902	0.928	0.865	0.918	0.798	3.4
LOG		0.902	0.927	0.866	0.918	0.797	3.4
TB-NB	33	0.883	0.858	**0.937**	**0.909**	0.757	4.4
NB(KLAR)		0.881	**0.964**	0.791	0.893	**0.771**	4.2
NB(E1071)		0.879	**0.965**	0.787	0.892	0.768	5.0
RF		0.768	0.836	0.685	0.798	0.529	8.0
SVM		0.885	0.955	0.805	0.899	0.774	4.0
CART		0.601	0.601	NaN	0.751	−1.000	9
LDA		**0.887**	0.915	0.848	0.905	0.767	3.4
LOG		**0.887**	0.915	0.848	0.905	0.767	3.4
TB-NB	50	**0.606**	**0.606**	0.313	**0.755**	0.005	3.0
NB(KLAR)		0.417	0.562	0.343	0.555	−0.096	6.8
NB(E1071)		0.477	0.567	0.350	0.560	−0.083	5.8
RF		0.502	0.603	0.401	0.501	0.004	5.0
SVM		0.526	0.517	0.122	**0.684**	−0.188	5.8
CART		0.502	NaN	NaN	0.386	−1.000	8.0
LDA		0.497	0.610	**0.408**	0.518	0.018	3.8
LOG		0.497	0.610	**0.408**	0.518	0.018	3.8

ACC accuracy, *TPR* true positive rate, *TNR* true negative rate, *F1* F1-score, *MCC* Matthews' correlation coefficient, $\overline{rk}$ average rank of the performance metrics ACC, TPR, TNR, F1 and MCC
Best values of performance metrics are reported in bold

It is worth noticing that Tb-NB is always among the top-ranked classifiers when increasing the percentage of perturbed data. When it is not ranked first, it provides values of the performance metrics that are always not far from those of the best-performing classifier.

The high resistance of Tb-NB is even more evident if the classifiers are compared in terms of Matthews' correlation coefficient (MCC). As is well known, MCC varies between -1 (worst classifier) and 1 (best classifier), whilst MCC$=0$ indicates that the classifier is performing like a "toss-a-coin" model. Thus, if the MCC value is between 0 and 1 a classifier is "usable", otherwise, it is "useless". Figure 8 compares the different classifiers in terms of MCC by varying the percentage of perturbed data from of one unit at a time, from 1 to 50%.

Results reported in Fig. 8 show that MCC of all the classifiers but not Tb-NB decreases rapidly to zero, or even -1, as long as the percentage of perturbed data increases from 33 to 50%. Tb-NB, instead, presents good values of MCC up to a percentage of perturbed data higher than 45%.

To further investigate the good performance of Tb-NB, we compute the other performance metrics for Tb-NB only still varying the percentage of perturbed data from one unit at a time, from 1 to 50%. In this case, besides the previously computed performance metrics, we also consider the BookMaker informedness (BM) and MarKedness (MK) (Chicco et al. 2021). Results reported in Fig. 9 show that Tb-NB is resistant to noise injection with respect to all the considered classification performance metrics.

5.3 Computational efficiency

Finally, after assessing how accurate and resistant the proposed classifier is, we consider the computational efficiency. For this purpose, to assess how much time is required to train a classifier and predict new instances, we consider a dataset of 100 random observations drawn from the entire set of Booking.com reviews. We randomly split the set of 100 cases into a learning set and a test set of equal size. For the set of classifiers already considered to evaluate accuracy and resistance, we compute the time required to train the classifier and that required to predict test set cases and compare the performance obtained for the different classifiers with that of Tb-NB. The whole experiment (sampling, estimation, and prediction) is repeated 100 times.

Results are summarized in Table 7 in terms of average computing time. It is demonstrated that Tb-NB is considerably quicker both in training and predicting time compared to the others. In particular:

(a) Tb-NB is $\sim$288 times quicker in prediction, and $\sim$5 times quicker in training, than the standard Naïve Bayes;
(b) Tb-NB is $\sim$19 times quicker in training and $\sim$82 times quicker in prediction than logistic regression (LOG);
(c) Tb-NB is $\sim$12 times quicker in training and $\sim$26 times quicker in prediction than random forest (RF);
(d) Besides Tb-NB, LDA and CART are the less computationally demanding classifiers. However, it is worth recalling that CART is very sensitive to outliers, as well as it is most of the time the less accurate and resistant classifiers among the

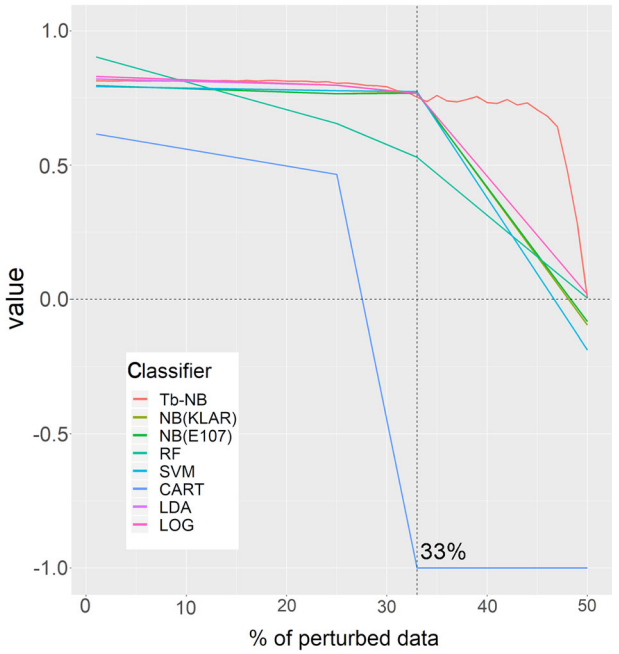

Fig. 8 Classifiers' resistance—MCC values over perturbed data percentage variation

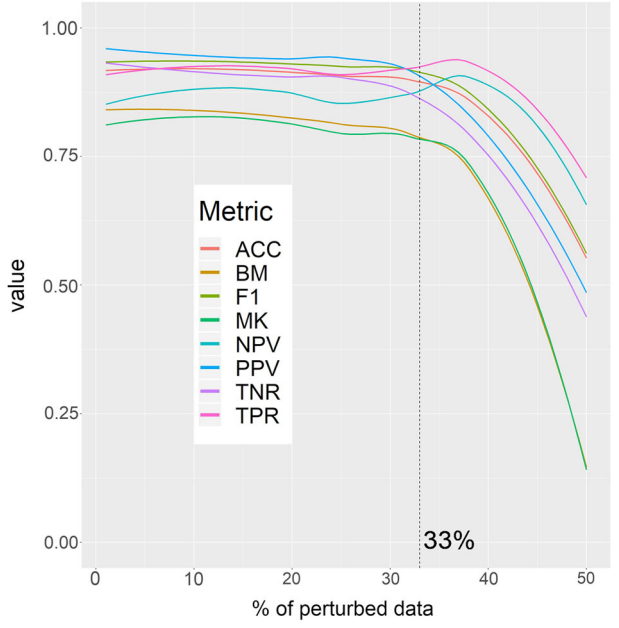

Fig. 9 Threshold-based NB performance indicators

Table 7 Computational efficiency

MODEL	TRAINING TIME	PREDICTING TIME	TRAINING TIME/Tb-NB	PREDICTING TIME/Tb-NB
Tb-NB	5.273	0.371	1.000	1.000
NB(KLAR)	26.999	106.608	5.120	287.654
NB(E1071)	26.999	106.608	5.120	287.654
RF	64.876	9.718	12.303	26.222
SVM	35.019	24.329	6.641	65.645
CART	11.177	4.162	2.120	11.229
LDA	8.418	0.779	1.549	2.101
LOG	102.026	30.260	19.348	81.650

Average training and predicting time (in seconds) of the considered classifiers for a training set of 50 observations and a test set of the same size (100 experiments)

set of the compared classifiers. As for LDA, it works well only when the initial assumptions (Gaussian distribution for each class, linear boundary, classes with the same variance, etc.) are met.

6 Concluding remarks

Nowadays, online word-of-mouth is a very important resource for electronic businesses because people pay close attention to user-generated reviews to decide on a specific product or service or to have an idea about the reputation of the product or service supplier. It happens that more and more consumers trust other consumers' reviews posted online. This phenomenon is characterizing the tourism industry also, thanks to the unprecedentedly growing of Internet applications for travel and tourism. Travel-related information, including hotel reviews, which are becoming one of the most popular online word-of-mouth. Consumer reviews can help travelers filter an enormous amount of information about their possible options. For example, consumers' reviews can be effective for the performance of small hospitality businesses that cannot access big advertising campaigns. More generally, word-of-mouth can affect managers to consider their brand building, product development, and quality assurance. However, when reading online reviews people might get confused because the available information is too vast. The main reason is that people are unable to read all the available reviews one by one. Thereby, sentiment analysis and classification of reviews into positive or negative opinions, has aroused researchers' great interest in recent years.

The above-mentioned considerations motivate the Threshold-based Naïve Bayes (Tb-NB) classifier introduced in this paper. It is a flexible, completely data-driven classifier, able to classify textual content into a positive or negative opinion based on a decision rule deriving from a single threshold value to be estimated from data. It is worth noticing that the model is completely nonparametric as no distribution of variables among the different classes is assumed. Obtained results depend on the

words included in the text only, thus paraphrasing a well-known motto popular in the context of nonparametric statistics it is possible to state that the inspiring principle of Tb-NB is “Let the words speak for themselves!".

We have shown that Tb-NB output allows the user to evaluate different dimensions of the service offered and thus understand the strengths and weaknesses of the supplier. The output of Tb-NB can be fruitfully utilized in a post-hoc analysis to understand how the quality of service, and the customer satisfaction levels, vary in time or in different areas. Moreover, we have shown that Tb-NB is accurate, resistant, and efficient from a computational viewpoint compared to other popular classifiers used for Sentiment Analysis.

The performance of Tb-NB has been evaluated on Booking.com data characterized by the presence of two sections in a review, namely a positive comment and a negative one but, in principle, Tb-NB can be used for any type of textual data. The analysis has been focused on reviews obtained from hotels guests of Sardinian hotels as a case study but no limitation about the number and the type of reviews to be analyzed exists. We have chosen Booking.com as a reference platform as the reviews there available come from customers who effectively stayed in a hotel.

Future research is mainly addressed to the generalization of the philosophy supporting Tb-NB into a framework that will allow us to use Tb-NB even for unlabeled textual data, as well as to the specification of an improved (iterative) version of Tb-NB which focuses on misclassified reviews in order to derive a more accurate decision rule for them.

Funding Open access funding provided by Universitá degli Studi di Cagliari within the CRUI-CARE Agreement. Funding was provided by Ministry of University (IT) - Prog. Dipartimenti di Eccellenza (Grant No. 1.005.14/2019).

Appendix

List of variables included in the Hotels data

1. Name: hotel’ name
2. Type: hotel type {1–Star, 2–Star, 3–Star, 4–Star, 5–Star, Other Facilities}
3. Postal Code: address related infos
4. City: address related infos
5. Oldest review: the date of the oldest review
6. Newest review: the date of the newest review
7. Analyzed days: total count of the considered days
8. Analyzed reviews/Analyzed days: proportion of analyzed reviews per day

9. Reviews/Analyzed days: proportion of total reviews count per day
10. Analyzed reviews: total count of the gathered reviews
11. Reviews: total count of the hotel' reviews
12. Analyzed reviews/Reviews: proportion of gathered reviews over the total amount
13. Positive comments: total count of the positive comments
14. Negative comments: total count of the negative comments
15. Booking overall score
16. Booking cleaning score
17. Booking comfort score
18. Booking position score
19. Booking services score
20. Booking staff score
21. Booking price-quality-rate score
22. Booking wifi score
23. Positive scores: mean of the positive comments scores
24. Negative scores: mean of the negative comments scores
25. Reviewer' reviews count: mean of the reviews count per reviewer
26. Apartment Standard: proportion of reviews with this room type
27. Apartment Superior: proportion of reviews with this room type
28. Bungalow Standard: proportion of reviews with this room type
29. Bungalow Superior: proportion of reviews with this room type
30. Double Standard (same-bed): proportion of reviews with this room type
31. Double Superior (same-bed): proportion of reviews with this room type
32. More Double Rooms: proportion of reviews with this room type
33. More Rooms: proportion of reviews with this room type
34. Single room Standard: proportion of reviews with this room type
35. Single room Superior: proportion of reviews with this room type
36. Double Standard: proportion of reviews with this room type
37. Double Superior: proportion of reviews with this room type
38. Triple Standard: proportion of reviews with this room type
39. Triple Superior: proportion of reviews with this room type
40. Family Standard: proportion of 4 beds standard rooms reviews
41. Family Superior: proportion of 4 beds superior rooms reviews
42. Group: proportion of group of people (2 or more) reviews
43. Single traveller: proportion of single travellers reviews
44. Business trip: proportion of business trip travellers reviews
45. Pleasure trip: proportion of business trip travellers reviews
46. Length of stay 1–3: proportion of reviews with a 1–3 nights stay
47. Length of stay 4–7: proportion of reviews with a 4–7 nights stay
48. Length of stay > 7: proportion of reviews with a > 7 nights stay
49. With a pet: proportion of travellers with a pet reviews
50. Other: proportion of reviews with none of the previous type of rooms
51. Apartment Standard (only positive): proportion of reviews with this room type
52. Apartment Superior (only positive): proportion of reviews with this room type
53. Bungalow Standard (only positive): proportion of reviews with this room type
54. Bungalow Superior (only positive): proportion of reviews with this room type

55. Double Standard (same-bed) (only positive): proportion of reviews with this room type
56. Double Superior (same-bed) (only positive): proportion of reviews with this room type
57. More Double Rooms (only positive): proportion of reviews with this room type
58. More Rooms (only positive): proportion of reviews with this room type
59. Single room Standard (only positive): proportion of reviews with this room type
60. Single room Superior (only positive): proportion of reviews with this room type
61. Double Standard (only positive): proportion of reviews with this room type
62. Double Superior (only positive): proportion of reviews with this room type
63. Triple Standard (only positive): proportion of reviews with this room type
64. Triple Superior (only positive): proportion of reviews with this room type
65. Family Standard (only positive): proportion of 4 beds standard rooms reviews
66. Family Superior (only positive): proportion of 4 beds superior rooms reviews
67. Group (only positive): proportion of group of people (2 or more) reviews
68. Single traveller (only positive): proportion of single travellers reviews
69. Business trip (only positive): proportion of business trip travellers reviews
70. Pleasure trip (only positive): proportion of business trip travellers reviews
71. Length of stay 1–3 (only positive): proportion of reviews with a 1–3 nights stay
72. Length of stay 4–7 (only positive): proportion of reviews with a 4–7 nights stay
73. Length of stay > 7 (only positive): proportion of reviews with a > 7 nights stay
74. With a pet (only positive): proportion of travellers with a pet reviews
75. Other (only positive): proportion of reviews with none of the previous type of rooms
76. Apartment Standard (only negative): proportion of reviews with this room type
77. Apartment Superior (only negative): proportion of reviews with this room type
78. Bungalow Standard (only negative): proportion of reviews with this room type
79. Bungalow Superior (only negative): proportion of reviews with this room type
80. Double Standard (same-bed) (only negative): proportion of reviews with this room type
81. Double Superior (same-bed) (only negative): proportion of reviews with this room type
82. More Double Rooms (only negative): proportion of reviews with this room type
83. More Rooms (only negative): proportion of reviews with this room type
84. Single room Standard (only negative): proportion of reviews with this room type
85. Single room Superior (only negative): proportion of reviews with this room type
86. Double Standard (only negative): proportion of reviews with this room type
87. Double Superior (only negative): proportion of reviews with this room type
88. Triple Standard (only negative): proportion of reviews with this room type
89. Triple Superior (only negative): proportion of reviews with this room type
90. Family Standard (only negative): proportion of 4 beds standard rooms reviews
91. Family Superior (only negative): proportion of 4 beds superior rooms reviews
92. Group (only negative): proportion of group of people (2 or more) reviews
93. Single traveller (only negative): proportion of single travellers reviews
94. Business trip (only negative): proportion of business trip travellers reviews
95. Pleasure trip (only negative): proportion of business trip travellers reviews

96. Length of stay 1–3 (only negative): proportion of reviews with a 1–3 nights stay
97. Length of stay 4–7 (only negative): proportion of reviews with a 4–7 nights stay
98. Length of stay > 7 (only negative): proportion of reviews with a > 7 nights stay
99. With a pet (only negative): proportion of travellers with a pet reviews
100. Other (only negative): proportion of reviews with none of the previous type of rooms
101. SleepQuality: proportion of reviews with a SleepQuality topic
102. Room: proportion of reviews with a Room topic
103. Services: proportion of reviews with a Services topic
104. PriceQualityRate: proportion of reviews with a PriceQualityRate topic
105. Cleaning: proportion of reviews with a Cleaning topic
106. Food: proportion of reviews with a Food topic
107. SleepQuality (only positive): proportion of reviews with a SleepQuality topic
108. Room (only positive): proportion of reviews with a Room topic
109. Services (only positive): proportion of reviews with a Services topic
110. PriceQualityRate (only positive): proportion of reviews with a PriceQualityRate topic
111. Cleaning (only positive): proportion of reviews with a Cleaning topic
112. Food (only positive): proportion of reviews with a Food topic
113. SleepQuality (only negative): proportion of reviews with a SleepQuality topic
114. Room (only negative): proportion of reviews with a Room topic
115. Services (only negative): proportion of reviews with a Services topic
116. PriceQualityRate (only negative): proportion of reviews with a PriceQualityRate topic
117. Cleaning (only negative): proportion of reviews with a Cleaning topic
118. Food (only negative): proportion of reviews with a Food topic

List of variables included in the Reviews data

1. Name: associated hotel' name
2. id. comment
3. id. review
4. Text: text corpora of the review
5. Neg or Pos: text classified by the reviewer as Negative or Positive {Neg, Pos}
6. Booking Score: Booking overall score
7. Booking sort position: sorted position of the review from Booking
8. Date: date of the review
9. Reviewer' reviews count: reviews count of the reviewer
10. Apartment Standard: review with this type of room (yes/no)
11. Apartment Superior: review with this type of room (yes/no)
12. Bungalow Standard: review with this type of room (yes/no)
13. Bungalow Superior: review with this type of room (yes/no)
14. Double Standard (same-bed): review with this type of room (yes/no)
15. Double Superior (same-bed): review with this type of room (yes/no)
16. More Double Rooms: review with this type of room (yes/no)
17. More Rooms: review with this type of room (yes/no)

18. Single room Standard: review with this type of room (yes/no)
19. Single room Superior: review with this type of room (yes/no)
20. Double Standard: review with this type of room (yes/no)
21. Double Superior: review with this type of room (yes/no)
22. Triple Standard: review with this type of room (yes/no)
23. Triple Superior: review with this type of room (yes/no)
24. Family Standard: review with this type of room (yes/no)
25. Family Superior: review with this type of room (yes/no)
26. Group: review from a group of 2 or more people (yes/no)
27. Single traveller: review from a single trip traveller (yes/no)
28. Business trip: review from a business trip traveller (yes/no)
29. Pleasure trip: review from a business trip traveller (yes/no)
30. Length of stay: {1–3, 4–7, $>$ 7} nights
31. With a pet: review from a traveller with a pet (yes/no)
32. Other: review with none of the previous type of rooms (yes/no)
33. SleepQuality: review with a SleepQuality topic (yes/no)
34. Room: review with a Room topic (yes/no)
35. Services: review with a Services topic (yes/no)
36. PriceQualityRate: review with a PriceQualityRate topic (yes/no)
37. Cleaning: review with a Cleaning topic (yes/no)
38. Food: review with a Food topic (yes/no)

See Tables 8 and 9.

Table 8 Table 5 bis—Benchmarking Tb-NB: performance metrics with the training-test set proportion 50–50

n		20.000					50.000					100.000				
	Classifier	ACC	TPR	TNR	F1	$\overline{rk}$	ACC	TPR	TNR	F1	$\overline{rk}$	ACC	TPR	TNR	F1	$\overline{rk}$
LEARNING SET (50%)	Tb-NB	0.878	0.907	0.841	0.894	4.75	0.878	0.912	0.836	0.894	4.00	0.878	0.915	0.833	0.893	4.00
	Tb-NB*	(0.909)	(0.922)	(0.881)	(0.925)	(4.50)	(0.910)	(0.929)	(0.882)	(0.926)	(4.00)	(0.910)	(0.929)	(0.881)	(0.926)	(4.50)
	NB(KLAR)	0.868	0.938	0.793	0.880	5.00	0.865	0.937	0.788	0.877	5.50	0.864	0.937	0.788	0.876	5.25
	NB(KLAR)*	(0.898)	(0.946)	(0.837)	(0.911)	(5.50)	(0.896)	(0.945)	(0.833)	(0.909)	(5.50)	(0.896)	(0.945)	(0.834)	(0.910)	(5.25)
	NB(E1071)	0.869	0.938	0.794	0.881	5.00	0.865	0.937	0.788	0.877	5.50	0.865	0.934	0.792	0.876	5.25
	NB(E1071)*	(0.902)	(0.947)	(0.844)	(0.916)	(4.50)	(0.900)	(0.946)	(0.840)	(0.914)	(4.50)	(0.900)	(0.946)	(0.841)	(0.915)	(4.25)
	RF	**0.939**	0.943	0.933	**0.948**	**1.50**	**0.939**	**0.946**	**0.930**	**0.948**	**1.00**	**0.939**	**0.946**	**0.927**	**0.947**	**1.00**
	RF*	(**0.958**)	(**0.963**)	(**0.950**)	(**0.965**)	(**1.00**)	(**0.947**)	(0.953)	(**0.937**)	(**0.954**)	(**1.00**)	(**0.946**)	(**0.955**)	(**0.933**)	(**0.954**)	(**1.00**)
	SVM	0.851	0.930	0.771	0.863	6.50	0.865	0.926	0.896	0.878	5.25	0.876	0.924	0.821	0.889	4.00
	SVM*	(0.870)	(0.940)	(0.789)	(0.885)	(6.50)	(0.883)	(0.940)	(0.815)	(0.898)	(6.50)	(0.891)	(0.939)	(0.832)	(0.906)	(6.25)
	CART	0.781	0.890	0.687	0.792	8.00	0.777	0.891	0.681	0.787	8.00	0.777	0.893	0.679	0.785	8.00
	CART*	(0.808)	(0.870)	(0.733)	(0.834)	(8.00)	(0.809)	(0.870)	(0.733)	(0.835)	(8.00)	(0.810)	(0.876)	(0.731)	(0.834)	(8.00)
	LDA	0.906	0.922	0.883	0.919	3.75	0.902	0.919	0.879	0.916	3.00	0.902	0.919	0.878	0.915	3.00
	LDA*	(0.916)	(0.936)	(0.887)	(0.929)	(3.25)	(0.914)	(0.936)	(0.883)	(0.928)	(3.00)	(0.913)	(0.936)	(0.881)	(0.927)	(3.50)
	LOG	0.914	**0.952**	**0.951**	0.926	**1.50**	0.871	0.923	0.813	0.884	4.25	0.869	0.922	0.811	0.882	5.00
	LOG*	(0.916)	(**0.953**)	(0.865)	(0.928)	(2.75)	(0.910)	(**0.950**)	(0.856)	(0.923)	(3.50)	(0.912)	(0.930)	(0.884)	(0.927)	(3.25)

Table 8 continued

n		20.000					50.000					100.000				
	Classifier	ACC	TPR	TNR	F1	$\overline{rk}$	ACC	TPR	TNR	F1	$\overline{rk}$	ACC	TPR	TNR	F1	$\overline{rk}$
TEST SET (50%)	Tb-NB	0.870	0.922	0.813	0.883	4.00	0.870	0.924	0.812	0.883	4.00	0.869	0.926	0.808	0.882	4.50
	Tb-NB*	(0.861)	(0.939)	(0.784)	(0.872)	(4.00)	(0.842)	(**0.943**)	(0.755)	(0.848)	(3.25)	(0.757)	(0.946)	(0.671)	(0.713)	(4.50)
	NB(KLAR)	0.863	**0.932**	0.791	0.876	4.75	0.864	**0.934**	0.790	0.876	5.00	0.865	**0.935**	0.792	0.877	5.00
	NB(KLAR)*	(0.850)	(**0.942**)	(0.761)	(0.861)	(4.75)	(0.834)	(0.940)	(0.742)	(0.840)	(5.75)	(0.738)	(**0.950**)	(0.647)	(0.684)	(6.00)
	NB(E1071)	0.864	**0.932**	0.792	0.876	4.25	0.864	**0.934**	0.790	0.876	5.00	0.865	**0.934**	0.792	0.876	5.50
	NB(E1071)*	(0.854)	(**0.941**)	(0.769)	(0.864)	(4.25)	(0.838)	(0.939)	(0.750)	(0.843)	(5.00)	(0.742)	(**0.949**)	(0.655)	(0.688)	(5.25)
	RF	0.875	0.868	**0.886**	0.896	3.75	0.885	0.884	**0.887**	0.903	3.75	0.890	0.891	**0.888**	0.907	3.75
	RF*	(0.869)	(0.880)	(**0.853**)	(0.888)	(**3.50**)	(0.867)	(0.881)	(**0.849**)	(**0.883**)	(**3.50**)	(**0.837**)	(0.873)	(**0.803**)	(**0.841**)	(**2.75**)
	SVM	0.847	0.927	0.767	0.870	7.75	0.863	0.925	0.794	0.876	5.00	0.874	0.922	0.819	0.888	4.00
	SVM*	(0.834)	(0.940)	(0.739)	(0.842)	(6.00)	(0.837)	(0.940)	(0.746)	(0.843)	(5.00)	(0.785)	(0.930)	(0.697)	(0.763)	(4.00)
	CART	0.777	0.886	0.683	0.788	7.75	0.776	0.889	0.679	0.786	7.75	0.776	0.892	0.679	0.784	7.75
	CART*	(0.787)	(0.874)	(0.703)	(0.802)	(8.00)	(0.780)	(0.877)	(0.694)	(0.790)	(8.00)	(0.737)	(0.904)	(0.653)	(0.696)	(6.75)
	LDA	0.882	0.898	0.859	0.899	3.25	0.894	0.911	0.869	0.908	3.25	0.895	0.913	0.871	0.910	3.25
	LDA*	(**0.880**)	(0.918)	(0.833)	(0.895)	(**2.25**)	(0.875)	(0.922)	(0.821)	(0.888)	(2.25)	(0.823)	(0.912)	(0.759)	(0.817)	(2.75)
	LOG	**0.883**	0.900	0.860	**0.900**	**2.25**	**0.897**	0.913	0.876	**0.912**	**2.25**	**0.899**	0.915	0.878	**0.914**	**2.25**
	LOG*	(**0.875**)	(0.914)	(0.827)	(**0.889**)	(3.25)	(**0.872**)	(0.919)	(0.818)	(**0.885**)	(3.25)	(0.802)	(0.903)	(0.737)	(0.782)	(4.00)

ACC accuracy, *TPR* true positive rate, *TNR* true negative rate, *F1* F1-score, $\overline{rk}$ average rank of the performance metrics ACC, TPR, TNR, and F1
Values in parenthesis are the performance metrics obtained considering reviews including comments with more than 3 words only. The corresponding classifier is marked with a "*". Best values of performance metrics are reported in bold

Table 9 Table 5 ter—Benchmarking Tb-NB: performance metrics with the training-test set proportion 67–33

n		20.000					50.000					100.000				
	Classifier	ACC	TPR	TNR	F1	$\overline{rk}$	ACC	TPR	TNR	F1	$\overline{rk}$	ACC	TPR	TNR	F1	$\overline{rk}$
LEARNING SET (67%)	Tb-NB	0.878	0.908	0.840	0.894	4.75	0.878	0.914	0.834	0.894	4.00	0.878	0.914	0.834	0.894	4.00
	Tb-NB*	(0.910)	(0.929)	(0.881)	(0.925)	(4.50)	(0.911)	(0.930)	(0.882)	(0.926)	(4.00)	(0.911)	(0.930)	(0.882)	(0.926)	(4.50)
	NB(KLAR)	0.867	0.938	0.792	0.879	5.50	0.866	0.937	0.791	0.878	5.50	0.866	0.937	0.791	0.878	5.50
	NB(KLAR)*	(0.899)	(0.946)	(0.840)	(0.913)	(5.50)	(0.897)	(0.945)	(0.837)	(0.911)	(5.50)	(0.898)	(0.945)	(0.838)	(0.912)	(5.50)
	NB(E1071)	0.867	0.938	0.792	0.879	5.00	0.866	0.937	0.791	0.878	5.50	0.866	0.937	0.791	0.878	5.50
	NB(E1071)*	(0.902)	(0.947)	(0.840)	(0.915)	(4.50)	(0.899)	(0.946)	(0.840)	(0.913)	(4.50)	(0.900)	(0.945)	(0.841)	(0.914)	(4.25)
	RF	**0.938**	0.943	0.931	**0.947**	**1.50**	**0.939**	**0.947**	**0.929**	**0.948**	**1.00**	**0.939**	**0.947**	**0.929**	**0.948**	**1.00**
	RF*	(**0.958**)	(**0.963**)	(**0.950**)	(**0.965**)	(**1.00**)	(**0.947**)	(0.953)	(**0.937**)	(**0.954**)	(**1.25**)	(**0.946**)	(**0.955**)	(**0.932**)	(**0.955**)	(**1.00**)
	SVM	0.856	0.929	0.779	0.868	6.50	0.869	0.925	0.806	0.883	5.00	0.869	0.925	0.806	0.883	5.00
	SVM*	(0.874)	(0.940)	(0.796)	(0.889)	(6.50)	(0.887)	(0.940)	(0.822)	(0.902)	(6.50)	(0.895)	(0.939)	(0.839)	(0.910)	(6.00)
	CART	0.780	0.889	0.685	0.791	8.00	0.777	0.891	0.680	0.787	8.00	0.776	0.890	0.679	0.786	8.00
	CART*	(0.808)	(0.869)	(0.733)	(0.834)	(8.00)	(0.809)	(0.870)	(0.734)	(0.835)	(8.00)	(0.810)	(0.876)	(0.731)	(0.835)	(8.00)
	LDA	0.904	0.920	0.882	0.918	3.75	0.902	0.919	0.879	0.916	3.00	0.902	0.919	0.879	0.916	3.00
	LDA*	(0.916)	(0.938)	(0.885)	(0.929)	(3.50)	(0.914)	(0.938)	(0.881)	(0.928)	(3.25)	(0.914)	(0.938)	(0.879)	(0.927)	(3.75)
	LOG	0.920	**0.958**	**0.957**	0.931	**1.50**	0.873	0.925	0.814	0.885	4.00	0.873	0.925	0.814	0.885	4.00
	LOG*	(0.919)	(**0.957**)	(0.868)	(0.931)	(2.50)	(0.913)	(**0.954**)	(0.859)	(0.926)	(3.00)	(0.915)	(0.934)	(0.888)	(0.930)	(3.00)

Table 9 continued

n		20.000					50.000					100.000				
	Classifier	ACC	TPR	TNR	F1	$\overline{rk}$	ACC	TPR	TNR	F1	$\overline{rk}$	ACC	TPR	TNR	F1	$\overline{rk}$
TEST SET (33%)	Tb-NB	0.870	0.923	0.811	0.884	4.00	0.869	0.926	0.808	0.883	3.75	0.869	0.926	0.808	0.883	3.75
	Tb-NB*	(0.860)	(0.940)	(0.782)	(0.870)	(4.00)	(0.841)	(**0.945**)	(0.753)	(0.846)	(3.5)	(0.756)	(0.947)	(0.669)	(0.711)	(4.50)
	NB(KLAR)	0.865	**0.935**	0.791	0.878	4.50	0.865	**0.935**	0.790	0.877	5.25	0.865	**0.935**	0.790	0.877	5.25
	NB(KLAR)*	(0.852)	(**0.943**)	(0.764)	(0.862)	(4.75)	(0.835)	(0.940)	(0.744)	(0.841)	(5.75)	(0.740)	(**0.950**)	(0.650)	(0.685)	(6.00)
	NB(E1071)	0.865	**0.935**	0.791	0.878	4.50	0.865	**0.935**	0.790	0.877	5.25	0.865	**0.935**	0.790	0.877	5.25
	NB(E1071)*	(0.854)	(**0.942**)	(0.768)	(0.863)	(4.25)	(0.837)	(0.940)	(0.748)	(0.842)	(5.25)	(0.742)	(**0.950**)	(0.654)	(0.687)	(5.25)
	RF	0.882	0.878	**0.888**	0.902	3.75	0.888	0.889	**0.886**	0.906	3.75	0.888	0.889	**0.886**	0.906	3.75
	RF*	(0.873)	(0.886)	(**0.857**)	(0.892)	(**3.50**)	(0.872)	(0.886)	(**0.853**)	(**0.887**)	(**3.50**)	(**0.842**)	(0.879)	(**0.807**)	(**0.844**)	(**2.75**)
	SVM	0.852	0.926	0.774	0.865	7.75	0.868	0.923	0.804	0.881	4.75	0.868	0.923	0.804	0.881	4.75
	SVM*	(0.838)	(0.940)	(0.745)	(0.846)	(6.00)	(0.841)	(0.940)	(0.753)	(0.848)	(4.50)	(0.789)	(0.930)	(0.703)	(0.767)	(4.00)
	CART	0.777	0.887	0.681	0.788	7.75	0.776	0.890	0.679	0.786	7.75	0.776	0.890	0.679	0.786	7.75
	CART*	(0.787)	(0.873)	(0.703)	(0.801)	(8.00)	(0.779)	(0.877)	(0.694)	(0.789)	(8.00)	(0.737)	(0.904)	(0.654)	(0.695)	(7.00)
	LDA	0.888	0.906	0.862	0.904	3.00	0.895	0.913	0.871	0.910	3.25	0.895	0.913	0.871	0.910	3.25
	LDA*	(**0.881**)	(0.922)	(0.831)	(0.894)	(**2.25**)	(0.875)	(0.926)	(0.819)	(0.887)	(3.00)	(0.824)	(0.916)	(0.756)	(0.816)	(2.75)
	LOG	**0.889**	0.906	0.866	**0.905**	**2.50**	**0.899**	0.914	0.877	**0.913**	**2.25**	**0.899**	0.914	0.877	**0.913**	**2.25**
	LOG*	(**0.878**)	(0.918)	(0.830)	(**0.892**)	(3.25)	(**0.875**)	(0.923)	(0.821)	(**0.887**)	(2.50)	(0.805)	(0.906)	(0.740)	(0.785)	(3.75)

ACC accuracy, *TPR* true positive rate, *TNR* true negative rate, *F1* F1-score, $\overline{rk}$ average rank of the performance metrics ACC, TPR, TNR, and F1
Values in parenthesis are the performance metrics obtained considering reviews including comments with more than 3 words only. The corresponding classifier is marked with a "*". Best values of performance metrics are reported in bold

References

Arndt J (1967) Role of product-related conversations in the diffusion of a new product. J Market Res 4(3):291–295. https://doi.org/10.2307/3149462

Bachtiar FA, Paulina W, Rusydi AN (2020) Text mining for aspect based sentiment analysis on customer review: a case study in the hotel industry. In: Serdült U, Loshchilov A, Mahmudy WF, Nurwasito H (eds) Proceedings of the 5th international workshop on innovations in information and communication science and technology (canceled by authorities due to SARS-CoV-2), CEUR workshop proceedings, vol 2627, pp 105–112, Malang, Indonesia, CEUR-WS.org

Boyd D, Crawford K (2012) Critical questions for big data: provocations for a cultural, technological, and scholarly phenomenon. Inf Commun Soc 15(5):662–679. https://doi.org/10.1080/1369118X.2012.678878

Brownlee J (2017) Deep learning for natural language processing: develop deep learning models for your natural language problems. In: Machine learning mastery, 1.7 edition

Buttle FA (1998) Word of mouth: understanding and managing referral marketing. J Strateg Market 6(3):241–254. https://doi.org/10.1080/096525498346658

Chai C (2019) Text mining in survey data. Surv Pract 12:1–13. https://doi.org/10.1017/S1351324920000534

Chaturvedi I, Cambria E, Welsch RE, Herrera F (2018) Distinguishing between facts and opinions for sentiment analysis: survey and challenges. Inf Fusion 44:65–77. https://doi.org/10.1016/j.inffus.2017.12.006

Chicco D, Jurman G (2020) The advantages of the Matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. BMC Genom. https://doi.org/10.1186/s12864-019-6413-7

Chicco D, Tötsch N, Jurman G (2021) The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. BioData Min 14(13):1–22. https://doi.org/10.1186/s13040-021-00244-z

Esuli A, Sebastiani F (2006) Determining term subjectivity and term orientation for opinion mining. In: 11th conference of the European chapter of the association for computational linguistics, pp 193–200, Trento, Italy, Association for Computational Linguistics. ISBN 1-932432-59-0

Goldberg Y (2017) Neural network methods in natural language processing. Synth Lect Hum Lang Technol 10(1):1–309. https://doi.org/10.2200/S00762ED1V01Y201703HLT037

Halevy A, Norvig P, Pereira F (2009) The unreasonable effectiveness of data. IEEE Intell Syst 24(2):8–12. https://doi.org/10.1109/MIS.2009.36

Harrison-Walker LJ (2001) The measurement of word-of-mouth communication and an investigation of service quality and customer commitment as potential antecedents. J Serv Res 4(1):60–75. https://doi.org/10.1177/109467050141006

Hartline MD, Jones KC (1996) Employee performance cues in a hotel service environment: influence on perceived service quality, value, and word-of-mouth intentions. J Bus Res 35(3):207–215. https://doi.org/10.1016/0148-2963(95)00126-3

Huang J, Lu J, Ling C (2003) Comparing Naive Bayes, decision trees, and svm with auc and accuracy. In: Third IEEE international conference on data mining, pp 553–556. https://doi.org/10.1109/ICDM.2003.1250975

Jain PK, Pamula R, Srivastava G (2021) A systematic literature review on machine learning applications for consumer sentiment analysis using online reviews. Comput Sci Rev 41:100413. https://doi.org/10.1016/j.cosrev.2021.100413

Janowicz-Lomott M, Łyskawa K, Polychronidou P, Karasavvoglou A (eds) (2018) Economic and financial challenges for Balkan and eastern European countries. In: Proceedings of the 10th international conference on the economies of the Balkan and Eastern European Countries in the Changing World (EBEEC) in Warsaw, Poland Springer proceedings in business and economics. Springer, Cham, 2020. ISBN 978-3-030-39926-9 978-3-030-39927-6. https://doi.org/10.1007/978-3-030-39927-6

Khan AH, Zubair M (2020) Classification of multi-lingual tweets, into multi-class model using Naïve Bayes and semi-supervised learning. Multimed Tools Appl 79(43–44):32749–32767. https://doi.org/10.1007/s11042-020-09512-2

Mazzarol T, Sweeney JC, Soutar GN (2007) Conceptualizing word-of-mouth activity, triggers and conditions: an exploratory study. Eur J Market 41(11/12):1475–1494. https://doi.org/10.1108/03090560710821260

Meyer D, Dimitriadou E, Hornik K, Weingessel A, Leisch F (2019) E1071: misc functions of the department of statistics, probability theory group (Formerly: E1071), TU Wien

Morante R, Blanco E (2021) Recent advances in processing negation. Nat Lang Eng 27:121–130. https://doi.org/10.1007/s10115-019-01410-w

Narayanan V, Arora I, Bhatia A (2013) Fast and accurate sentiment classification using an enhanced Naive Bayes model. In: Hutchison D, Kanade T, Kittler J et al (eds) Intelligent data engineering and automated learning—IDEAL 2013, vol 8206, pp 194–201. Springer, Berlin. ISBN 978-3-642-41277-6 978-3-642-41278-3. https://doi.org/10.1007/978-3-642-41278-3_24

Nielsen (2007) Trust in advertising. A global Nielsen consumer report

Noori B (2021) Classification of customer reviews using machine learning algorithms. Appl Artif Intell 35(8):567–588. https://doi.org/10.1080/08839514.2021.1922843

O'Connor P (2010) Managing a hotel's image on TripAdvisor. J Hosp Market Manag 19(7):754–772. https://doi.org/10.1080/19368623.2010.508007

Pang B, Lee L (2008) Opinion mining and sentiment analysis. Found Trends Inf Retr 2(1–2):1–135. https://doi.org/10.1561/1500000011

Rusticus S (2007) Creating brand advocates. Justin Kirby and Paul Marsden, Oxford

Santos G, Mota VFS, Benevenuto F, Silva TH (2020) Neutrality may matter: sentiment analysis in reviews of Airbnb, Booking, and Couchsurfing in Brazil and USA. Soc Netw Anal Min 10(1):45. https://doi.org/10.1007/s13278-020-00656-5

Schmunk S, Höpken W, Fuchs M, Lexhagen M (2013) Sentiment analysis: extracting decision-relevant knowledge from UGC. In: Xiang Z, Tussyadiah I (eds) Information and communication technologies in tourism 2014. Springer, Cham, pp 253–265. ISBN 978-3-319-03972-5 978-3-319-03973-2. https://doi.org/10.1007/978-3-319-03973-2_19

Schuckert M, Liu X, Law R (2015) A segmentation of online reviews by language groups: how English and Non-English speakers rate hotels differently. Int J Hosp Manag 48:143–149. https://doi.org/10.1016/j.ijhm.2014.12.007

Sırma E (2009) Word-of-mouth marketing from a global perspective. Ph.D. thesis, Instituto Universitàrio de Lisboa,

Sparks BA, Perkins HE, Buckley R (2013) Online travel reviews as persuasive communication: the effects of content type, source, and certification logos on consumer behavior. Tour Manag 39:1–9. https://doi.org/10.1016/j.tourman.2013.03.007

Tavazoee F, Conversano C, Mola F (2020) Recurrent random forest for the assessment of popularity in social media. Knowl Inf Syst 62:1847–1879. https://doi.org/10.1007/s10115-019-01410-w

Weihs C, Ligges U, Luebke K, Raabe N (2005) klaR analyzing German business cycles. In: Baier D, Decker R, Schmidt-Thieme L (eds) Data analysis and decision support. Springer, Berlin, pp 335–343. ISBN 978-3-540-26007-3. https://doi.org/10.1007/3-540-28397-8_36

Wiebe JM, Bruce RF, O'Hara TP (1999) Development and use of a gold-standard data set for subjectivity classifications. In: Proceedings of the 37th annual meeting of the association for computational linguistics, College Park, Maryland, USA. Association for Computational Linguistics, pp 246–253. https://doi.org/10.3115/1034678.1034721

Xu F, Pan Z, Xia R (2020) E-commerce product review sentiment classification based on a Naïve Bayes continuous learning framework. Inf Process Manag 57(5):102221. https://doi.org/10.1016/j.ipm.2020.102221

Yang P, Chen Y (2017) A survey on sentiment analysis by using machine learning methods. In: 2017 IEEE 2nd information technology, networking, electronic and automation control conference (ITNEC), pp 117–121. https://doi.org/10.1109/ITNEC.2017.8284920

Yang Y, Mueller NJ, Croes RR (2016) Market accessibility and hotel prices in the Caribbean: the moderating effect of quality-signaling factors. Tour Manag 56(C):40–51

Ye Q, Zhang Z, Law R (2009) Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. Expert Syst Appl 36(3):6527–6535. https://doi.org/10.1016/j.eswa.2008.07.035

Yu L-C, Wang J, Lai KR, Zhang X (2018) Refining word embeddings using intensity scores for sentiment analysis. IEEE/ACM Trans Audio Speech Lang Process 26(3):671–681. https://doi.org/10.1109/TASLP.2017.2788182

Yuan Y-H, Tsao S-H, Chyou J-T, Tsai S-B (2020) An empirical study on effects of electronic word-of-mouth and Internet risk avoidance on purchase intention: from the perspective of big data. Soft Comput 24(8):5713–5728. https://doi.org/10.1007/s00500-019-04300-z