ORIGINAL ARTICLE
DOI: 10.5433/1679-0375.2022v43n2p189

Short text classification applied to item description: Some methods evaluation

Classificação de texto curto aplicada à descrição de produto: Avaliação de alguns métodos

Gilsiley Henrique Darú[1]; Felipe Daltrozo da Motta[2];
Antonio Castelo[3]; Gustavo Valentim Loch[4]

Abstract

The increasing demand for information classification based on content in the age of social media and e-commerce has led to the need for automated product classification using their descriptions. This study aims to evaluate various techniques for this task, with a focus on descriptions written in Portuguese. A pipeline is implemented to preprocess the data, including lowercasing, accent removal, and unigram tokenization. The bag of words method is then used to convert text into numerical data, and five classification techniques are applied: argmaxtf, argmaxtfnorm, argmaxtfidf from information retrieval, and two machine learning methods logistic regression and support vector machines. The performance of each technique is evaluated using simple accuracy via thirty-fold cross validation. The results show that logistic regression achieves the highest mean accuracy among the evaluated techniques.

Keywords: text classification; product description; short text; logistic regression; bag of words.

Resumo

A crescente demanda por classificação de informações baseada em conteúdo na era das mídias sociais e do comércio eletrônico tem levado à necessidade de classificação automatizada de produtos com base nas suas descrições. Este estudo tem como objetivo avaliar várias técnicas para essa tarefa, com ênfase em descrições escritas em português. Uma pipeline é implementada para pré-processar os dados, incluindo conversão para minúsculas, remoção de acentos e separação por espaço de unigramas. Em seguida, o método sacola de palavras é usado para converter o texto em dados numéricos e cinco técnicas de classificação são aplicadas: argmaxtf, argmaxtfnorm, argmaxtfidf proveniente da recuperação de informação e duas técnicas de aprendizado de máquina: regressão logística e máquinas de vetores de suporte. O desempenho de cada técnica é avaliado usando a acurácia via validação cruzada com trinta conjuntos. Os resultados mostram que a regressão logística alcança a maior acurácia média entre as técnicas avaliadas.

Palavras-chave: classificação de texto; descrição do produto; texto curto; regressão logística; sacola de palavras.

[1] MSc., ICMC, USP, São Carlos, São Paulo, Brasil, E-mail: ghdaru@usp.br
[2] MSc., ICMC, USP, São Carlos, São Paulo, Brasil, E-mail: daltrozo@gmail.com
[3] Prof. Dr., ICMC, USP, São Carlos, São Paulo, Brasil, E-mail: castelo@icmc.usp.br
[4] Dr., PPGMNE, UFPR, Curitiba, PR, Brasil, E-mail: gustavo.valentim@gmail.com

Introduction

With the rapid growth of social media and e-commerce, the need to classify information based on its content has accelerated rapidly and many applications have appeared, such as spam filtering, sentiment analysis, advertising personalization, customer review, and labeling or text classification (ALSMADI; GAN, 2019).

In this study, text classification consists of an item or product description, obtaining its corresponding category. A short text definition is a text with up to 200 characters, (ALSMADI; GAN, 2019; SONG *et al.*, 2014). A product description is a short text because it has fewer than 200 characters. This task is an essential part of companies that need to catalog information, carry out their purchasing planning organization, physical organization by categories, use in category management, and promotion of the product on e-commerce sites.

Recently, a dataset called DARU was made available by Daru (2022) which contains classified product descriptions in Portuguese. This study aims to conduct the first analysis of the performance of text classification algorithms on the DARU dataset, using traditional information retrieval algorithms as well as support vector machines (SVMs) and logistic regression (LR). Previous research by Alsmadi and Gan (2019) has found that SVMs tend to have the best average performance among various supervised learning techniques for short text classification, with accuracy being the most commonly used metric. In contrast, (ALSMADI; GAN, 2019) found that logistic regression outperforms decision trees, random forests, SVMs, Naive Bayes, AdaBoost, and Bagging in terms of classification accuracy.

It is necessary to incorporate additional steps to perform text classification. Bhavani and Kumar (2021) introduces a typical text classification process, shown in Figure 1.

Figure 1 illustrates the various components involved in a text classification task, including product description, preprocessing, feature extraction, and algorithm. A product description is a short text containing information about a product, such as its name, brand, and features. For example, *"Kit Kat, Miniatures Assorted Chocolate and White Creme Wafer Bars, Christmas Candy, 19.2 oz, Bag"*. Preprocessing involves three main steps: tokenization, normalization, and noise removal. Tokenization involves dividing a string into smaller parts, while normalization involves removing irrelevant words and noise removal involves removing unwanted text and standardizing different word forms into a single appropriate form (BHAVANI; KUMAR, 2021). Feature extraction involves converting text into a set of numerical data. According to Silva et al. (2020), using a simple bag of words technique often produces better results for text classification tasks compared to other word transformation techniques.

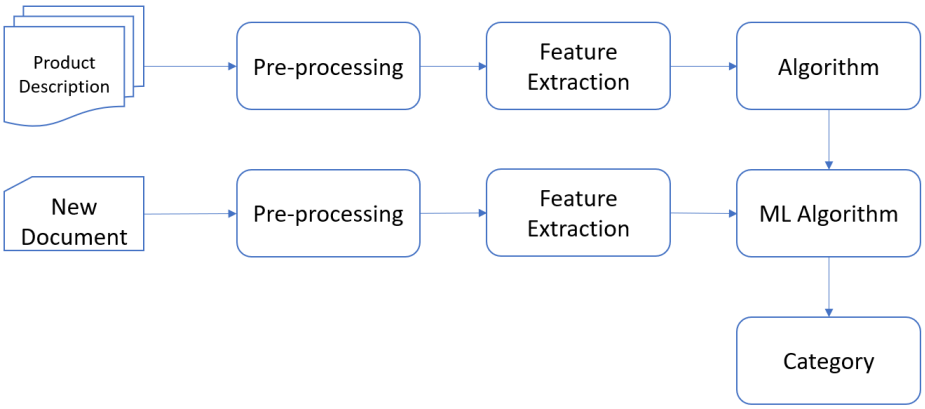
The cited works demonstrate that text classification is difficult. In the next section, this difficulty is demonstrated empirically through the classification of item descriptions.

A example of a text classification problem in Portuguese

A text classification problem involving products with the word "BISCOITO" in their name is presented. In Portuguese, "BISCOITO" means cracker. Searching for this word alone may not capture all relevant product descriptions, as some may abbreviate it as "BISC", such as "BISC LEITE MABEL 400G" and "BISC.PARATI MARIA PE 370GR". Therefore, the initial vocabulary must be expanded to include these abbreviations in order to classify these products correctly. Additionally, other words that are sometimes used to refer to crackers in Portuguese, such as "BOLACHA", "COOKIE", and "COOKIES", must also be added to the vocabulary list to ensure that all relevant products are captured. However, this solution is not foolproof, as some descriptions containing the word "BISCOITO" may not actually be crackers, such as "BISC PET CRACKER FORTAL 400G" and "BISC PET DOG CROCK 500G" which are pet food crackers, and "BISC LACTA BIS FLOWP 126G, LAKA BCO" and "CESTA SUPREME BISCOITO LIMAO 100G" which are chocolate and a gift basket, respectively. An example of this is shown in a t-shirt with the phrase "NÃO É BOLACHA, É BISCOITO!" in Figure 2.

The text aims to establish a benchmark for accuracy on a dataset containing Portuguese product descriptions. Comparisons between information retrieval techniques and machine learning techniques are carried out on this dataset. A pipeline is implemented and optimal hyperparameters for logistic regression and support vector machine are determined. The following section presents definitions and a practical example. The Related Works section discusses similar studies, while the Methodology and Results sections describe the procedures used and the results obtained.

Figure 1 – Text classifification process.



Source: Adapted from Alsmadi and Gan (2019).

Figure 2 – T-shirt with word BISCOITO stamped.

Source: The authors.

Definitions

Text classification

According to Aggarwal and Zhai (2012), a classification problem involves a set of instances $D = \{X_1, ..., X_n\}$, where each instance belongs to one of k classes indexed by $\{1, 2, ..., k\}$, where k is the number of classes, X_i is an instance, for this work, a product description, n is the number of instances. A training set is used to build a classification model. A new unknown instance uses the built model to predict its class. The classification problem can be considered *hard* when a class is explicitly determined or *soft* when probabilities are assigned. Text classification requires converting text into numbers. It is necessary to use natural language processing techniques.

Natural Language Processing

Natural language processing is a field of study focused on the manipulation of texts. Table 1 in Baeza-Yates and Ribeiro-Neto (2013) provides definitions for some key concepts in this area.

Table 1 – Natural language processing definitions.

Term	Definition
Corpus	A collection of written texts or documents
Document	Any set of texts
Set	is a collection of objects, without order or repetition
Token	a part of a document
Type	a distinct token
Vocabulary (V)	A set of types
Vector	A mathematical representation of a document

Source: Based in Baeza-Yates and Ribeiro-Neto (2013).

Natural language processing (NLP) includes various techniques for representing words as numerical values, known as word vectors.

Word vectors

Word vectors are numerical representations of words in a dataset. These representations, also known as word embeddings, capture the relationships between words in the dataset and allow them to be analyzed and used in natural language processing tasks such as text classification, language translation, and sentiment analysis. Word vectors can be generated using various techniques, such as bag-of-words, term-frequency inverse document frequency (TF-IDF), and word2vec.

These techniques convert words into numerical vectors by considering the frequency and context of the words in the dataset. Word vectors are typically used as input to machine learning models, which can then learn from the relationships between the words and make predictions or perform other tasks. The next sections present Bag of words (Bow), term frequency (TF), and term frequency inverse document frequency denoted by TFIDF.

Bag of words and term frequency

Bag of words (BoW) is a popular representation method in object categorization, according to Zhang, Jin and Zhou (2010). It involves assigning a numerical value to each word in a description and representing the description as a vector indicating the presence or absence of each word, with a value of 1 indicating the word's presence and 0 indicating its absence. For example, the descriptions: 'Arroz tio joão 1 kg', 'ARROZ FUMACENCE PARB 1KG', 'FEIJAO CARIOCA 1KG AZULAO', 'Feij 1 Kg Preto Caldão'. would produce the following vocabulary: {arroz, tio, joao, 1, kg, fumacence, parb, 1kg, feijao, carioca, azulao, feij, preto, caldao}.

The representation of each word would be a vector with a 1 in the position corresponding to the word's position in the vocabulary and 0s in all other positions. For example, "arroz" would be represented by {1,0,0,0,0,0,0,0,0,0,0,0,0}. To represent a description, the description would first be preprocessed and tokenized, and then each token would be converted to its vector representation. Finally, these vectors would be added together, which is equivalent to a word count. For example, the description 'Feijao Preto Caldao Preto 1kg' would be converted to {0,0,0,0,0,0,0,1,1,0,0,0,2,1}.

Term frequency and inverse document frequency

Another method for vectorization is to multiply the term frequency (TF) by the inverse of the number of documents in which the term appears (BAEZA-YATES; RIBEIRO-NETO, 2013). In this case, a document represents a class and contains all of its descriptions. The weight of each term is calculated using the equation (1):

$$TFIDF_{ij} = TF_{ij} \cdot \log \frac{d_i}{|D|}, \qquad (1)$$

where TF_{ij} is the number of times that term i appears in document j, d_i is the number of documents that term i appears and $|D|$ is the number of documents in the corpus.

Related works

Alsmadi and Gan (2019) proposed a classification to supervised short text classification algorithms: linear classifier, probabilistic classifier, rule-base classifier, decision tree classifier, and example based classifier.

Bhavani and Kumar (2021) cited K-Nearest Neighbors, Naïve Bayes, Support Vector Machine, Decision Tree, Neural Networks, Convolutional Neural Network (CNN), Linear Regression, and Recurrent Neural Network (RNN).

Pranckevicius and Marcinkevicius (2017) used Naïve Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression. Aggarwal and Zhai (2012) divide similarly to Alsmadi and Gan (2019) but include other classifiers based on genetic algorithms.

To the text classification task Shah *et al.* (2020) compared logistic regression, Random Forest and KNN models. They concluded that logistic regression presented the best results when using the accuracy metric. In his work, the metrics of precision, accuracy, F1-score, support matrix, and confusion were used.

In Pranckevicius and Marcinkevicius (2017), the authors compared the use of Naïve Bayes, Random Forest, decision tree, Support Vector Machines (SVM), and logistic regression classifiers for text classification. The aim of this article was to compare these classifiers by evaluating their classification accuracy, based on the size of the training datasets. They concluded that logistic regression presents the best average accuracy for the short text classification task.

Alsmadi and Gan (2019) evaluated twenty three articles. The SVM algorithm was the best in more than fifty percent of the articles. The author showed in his study that ensembled methods presented the best result in most than thirty percent. The most used word embedding was TFIDF.

Silva *et al.* (2020) studied many word embedding techniques combined with supervised algorithms to detect fake news in Portuguese. The author showed that the bag of words combined with LR, SVM, NB, DT, RF, Adaboost, and bagging outperformed Bow with other preprocessing techniques or more modern techniques like Word2vec or Fast text. The best accuracy was obtained by Logistic Regression when compared with the methods cited above.

Materials and methods

Dataset

The dataset for this study was obtained from Daru (2022). The first five rows are shown in Table 2.

Table 2 – Natural language processing definitions.

Product description	Category
Apresuntado rezende pec kg	apresuntado
carne suin espinhaco kg	carne suína
whisky white 1L horse trad.	whisky
whisky johnn walker 1L un	whisky
lte cond moca desn tp 395G	leite condensado

Source: The authors.

The datase, Table 2, includes product descriptions in Portuguese in the first column and their corresponding categories in the second column, and contains 158113 product descriptions, distributed over 727 categories.

However, the dataset is not balanced, with the largest category being "BISCOITO" (cracker) containing over 8,000 items and the smallest categories being "CHAPINHA" (hair crimper), "ESTANTE" (shelf), and "CAFETEIRA" (coffee machine) each having only one row.

The dataset includes product descriptions from the 18 largest retailers in Brazil, based on the ABRAS ranking, and only those products that represent 95% of sales are classified.

Language programming and machine learning library

For this study, the Python programming language (ROSSUM; DRAKE, 2009) and scikitlearn library (PEDREGOSA *et al.*, 2001) were used.

Pipeline

A pipeline is a sequence of tasks executed to reach an objective. The objective of this work is to classify a description. It is used two pipelines, Figure 3, one for argmaxs methods and another for machine learning methods. In the next sections, each task is explained.

Pre-processing

For this work, the following pre-processing techniques were used:

- Conversion to Uppercase or Lowercase.
  Example:{ARROZ, Arroz, arroz}→arroz
- Accent extraction.
  Example: {Feijão, Açúcar}→{Feijao, acucar}

Tokenization is realized separating description by space. For example, the description "Arroz TIO JOÃO 1kg" is broken in {arroz, tio, joao, 1kg}.

Feature extraction

Feature extraction is the process of identifying and extracting important characteristics or patterns from a dataset in order to use them for further analysis or to build predictive models. It is commonly used in machine learning and data analysis to simplify and prepare data for modeling.

There are various methods for feature extraction, including selecting a subset of the features, creating new features from existing ones, projecting the data onto a lower-dimensional space, and encoding categorical variables.

In this specific work, the authors prepared the dataset by applying preprocessing and tokenization techniques, represented each token in a vector space using bag of words (BoW), term frequency (TF), and term frequency-inverse document frequency (TF-IDF), and then evaluated the performance of five different techniques on the dataset, including three based on information retrieval and two using machine learning.

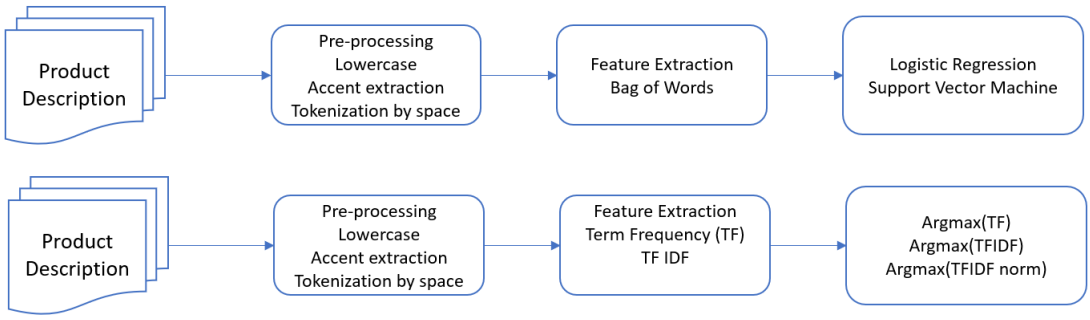
Evaluated models

For evaluation, the authors selected five techniques: three based on information retrieval (ArgmaxTF, ArgmaxTFnorm, and ArgmaxTFIDF) and two using machine learning (Logistic Regression with Bag of Words and Support Vector Machines with Bag of Words). The reasoning for choosing these techniques is explained in the introduction section of the work.

Argmax term frequency

This model groups descriptions of a category and vectors them by the frequency of terms, TF. This generates the array A with dimension $|V| \times |D|$, where $|V|$ is the vocabulary size and $|D|$ is the number of documents.

Figure 3 – The Pipeline used by authors, above to machine learning algorithms and under to argmaxs algorithms.



Source: The authors.

The description is converted into its representation by the frequency of terms and is verified from the similarity of the words. The vector **x** is the bag of words of description i and has dimension $(V,1)$. Thus, the multiplication in equation (2) has the dimension $(D,V)\times(V,1)$, which provides a vector of dimension D, that is, the number of similar terms in the description with the category given by

$$Category = argmax(A^T x_i). \tag{2}$$

Argmax Normalized Term-Frequency

The same comments are valid for this model with the difference that vector "A" is column normalized, where the element $A_{\cdot}$ represents a row vector with the column norms of "A" being the divisor of each column of "A", and the category given by equation (3)

$$Category = argmax\left(\frac{A}{A_{\cdot}}^T x_i\right). \tag{3}$$

Argmax term frequency inverse document frequency

This model groups the descriptions of a category and vectorizes them by the frequency of terms, TF and multiplies by the inverse of the number of documents, as described in the section on *Term frequency and inverse document frequency*. This generates the array B with dimension $|V|\times|D|$, where V is the vocabulary size and D is the number of documents. The description is converted to its representation by the frequency of terms and it is verified from the similarity of words. The vector x is the bag of words of the vector **x** and has dimension $(V,1)$. Thus, the multiplication below has the dimension $(D,V)\times(V,1)$, which provides a vector of dimension D, that is, the number of similar terms in the description with the category.

Support vector machine

A support vector machine, SVM, is an optimization algorithm that aims to find a hyperplane that minimizes its distance from the support vectors. It is given by the equation (4)

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C\sum_{i=1}^{n} \zeta_i \tag{4}$$

subject to

$$y_i(w^T\phi(x_i)+b) \geq 1-\zeta_i,$$
$$\zeta_i \geq 0, i = 1,...,n,$$

obtained from the library documentation (PEDREGOSA *et al.*, 2001).

The method requires some hyperparameters: Kernel functions, C, class weight, and degree. The kernel function is a mathematical trick that creates auxiliary dimensions that allow generating the hyperplane in a more efficient way. The main ones are linear, polynomial, radial, and sigmoid basis functions. These functions are given in the equation by the term $\phi(x)$. The C hyperparameter is related to the tolerance in relation to the vectors being outside the support region. The higher this hyperparameter, the higher the cost and therefore the lower the tolerance. Again, the class weight hyperparameter applies a weight to classes with more records. The degree hyperparameter is only applicable to the polynomial model.

Logistic regression

Logistic regression is a supervised machine learning algorithm used for classification. Here the technique will not be presented, only the definition of the model with the hyperparameters that were used.

Its general equation is:

$$\min_{w,c} \frac{1-\rho}{2} w^T w + \rho \|w\|_1 + C \sum_{i=1}^{n} \log\left(e^{-y_i(X_i^T w + c)} + 1\right). \quad (5)$$

The LR hyperparameters obtained from Pedregosa *et al.* (2001) are solver, penalty, and C. The solver used was the saga as informed in the reference being the preferred one for the base with tens of thousands of samples. The penalty represents detractors to be applied to each element of the attributes, where l1 is the Euclidean norm, l2 is the modulus and elasticnet is a combination of weights between l1 and l2, where the hyperparameter that determines the importance is l1_ratio, in the formula denoted by ρ. So making sense only when the penalty is elasticnet. The hyperparameter C is the inverse of the penalty, as seen in the formula. The class_weight hyperparameter refers to the weight given to the samples. If its value is balanced, a weight of $1/n_c$ is used, where n_c is the number of samples of the analyzed class. Otherwise, all samples are considered to be of the same weight.

Evaluation

The accuracy was selected and justified in the introduction. To find the best hyperparameters the mean accuracy was calculated for a four fold cross validation to each combination. The decision is justified from the work (BENGIO; GRANDVALET, 2003), where above four folds, the accuracy is stable for no outliers dataset. As the growth of the combination is exponential, a sample of the original dataset was selected, in this case, a thousand samples, just by convenience.

The second phase consists in to evaluate the model performance over accuracy to the selected models. For this phase is utilized 30 folds for statistical purposes.

Results and discussions

Logistic regression: hyperparameter selection

The total combinations tested was 80. These were generated from all viable combinations of hyperparameters *penalty*, with values {"none", "l1", "l2", "elasticnet"}, *C* with values {0.1,1,10,100,1000}, *class weight* with values {None, 'balanced'} and $l1_{ratio}$ with values {0.1, 0.3, 0.5, 0.7, 0.9}, only when the *penalty* is elasticnet.

After executing the models, the data were grouped by hyperparameter and the results obtained presented in Tables 3 to 5. It can be seen that not applying a penalty has the best mean accuracy and the lower standard deviation accuracy. Because of this, the evaluation of the $l1_{ratio}$ hyperparameter is not relevant. Similarly, the non-application of weights has the best mean and the lower standard deviation.

Finally, the value 100 to C hyperparameter presents the best accuracy and the lowest standard deviation. The selected hyperparameters were C equal to 100, solver equal to saga, no penalty, and no class weight.

Table 3 – Penalty hyperparameter mean and standard deviation accuracies results.

value	mean %	deviation %
none	87.69	2.31
l2	79.15	22.12
elasticnet	76.59	22.23
l1	73.16	28.56

Source: The authors.

Table 4 – Class weight hyperparameter mean and standard deviation accuracies results.

value	mean %	deviation %
none	81.06	15.59
balanced	74.62	26.15

Source: The authors.

Table 5 – C hyperparameter mean and standard deviation accuracies results.

value	mean %	deviation %
100.0	88.31	0.89
1.0	86.11	2.70
1000.0	83.62	1.02
10.0	83.09	18.46
0.1	53.57	29.71

Source: The authors.

Support vector machine SVM: hyperparameter selection

A total of 80 combinations were evaluated. These were generated from all viable combinations of hyperparameters *kernel*, with values {"linear", "poly", "rbf", "sigmoid"}, *C* with values 0.1,1,10,100,1000}, *class weight* with values {None, 'balanced'} and *degree* with values {2,3,5,8,13}, only for the *kernel* poly.

After the evaluation of all combinations, Tables 6 to 8 were generated. It can be observed that the kernel with the best result is sigmoid, with higher mean accuracy and lower standard deviation. The best hyperparameters C is the value 100. As the polynomial model was not chosen, there is no need to choose the degree hyperparameter. Finally, the model did not distinguish between balancing or not of classes.

The selected hyperparameters were C equal to 100, kernel equal to a sigmoid, and class weight equal to balanced.

Table 6 – Kernel hyperparameter mean and standard deviation accuracies results.

value	mean %	deviation %
sigmoid	87.35	1.32
rbf	85.32	3.20
linear	84.87	7.98
poly	45.18	30.78

Source: The authors.

Table 7 – C hyperparameter mean and standard deviation accuracies results.

value	mean %	deviation %
100	66.45	29.18
1000	64.44	30.77
10	63.16	31.14
1	60.88	32.23
0.1	31.30	34.65

Source: The authors.

Table 8 – Class weight hyperparameter mean and standard deviation accuracies results.

value	mean %	deviation %
balanced	56.87	35.58
none	56.86	33.09

Source: The authors.

Final results

For the final model, all samples were tested using cross-validation with 30 folds for each of the 5 models described. Each fold consists in 96.67% or 152842 instances to train and 3.33% or 5271 instances to test. The result after the executions are shown in Table 9.

All models have a low standard deviation, less than 1%. This is justified by the big training set when compared to the test set.

The most predictive model is logistic regression with 93.57% of accuracy and 0.25% deviation, corroborating with the articles analyzed. Following, support vector machines with 89.88% of accuracy and 0.40% deviation. This result is similar to the literature where LR and SVM present generally the best accuracy results.

On the other hand, the lowest accuracy is obtained with the simple term frequency or argmaxtf. This is the most simple model and here it showed that is the least predictive model with 59.77% accuracy.

Subsequently, the argmaxtfnorm model, which normalizes each document vector, significantly increased accuracy performance from 59.77% to 74.44%, and presents a result similar to argmaxtfidf (with 76.38% accuracy). This improvement when normalizing the vector is due to the effect of the amount being removed in each category, making them comparable. The best accuracy for information retrieval models was obtained with argmaxtfidf with 76.38% accuracy.

Table 9 – Mean accuracy and standard deviation obtained from a thirty folds evaluation by model.

method	mean %	deviation %
regression	93.57	0.25
svc	89.88	0.40
argmaxtfidf	76.38	0.71
argmaxtfnorm	74.44	0.62
argmaxtf	59.77	0.75

Source: The authors.

Conclusion

The aim of this text was to analyze the performance of machine learning algorithms on a dataset containing product descriptions in Portuguese. The results showed that machine learning algorithms outperformed basic word counting or weighting techniques in short text classification tasks with product descriptions in Portuguese.

Simple preprocessing techniques, such as converting the text to lowercase, removing noise, and tokenizing by space, combined with a bag of words representation and logistic regression, proved to be an effective and efficient method for classifying product descriptions.

The best results were obtained using logistic regression with the following hyperparameters: C equal to 100, solver equal to sage, without class weight, and without penalty. Support vector machines outperformed information retrieval algorithms but do not logistic regression. For this technique, SVM, the best results were obtained with the following combination of hyperparameters: C equal to 100, kernel equal to a sigmoid, and weight of class equal to balanced.

Among information retrieval techniques, argmaxtfnorm and argmaxtfidf obtained similar results and the worst result was obtained with a simple word count – argmaxtf. Cross-validation with four sets and 1,000 samples demonstrated viability to find the best set of hyperparameters and cross-validation with 30 sets allowed to generate statistics to evaluate the accuracy mean.

As suggestions for future work, the authors recommend expanding preprocessing techniques to include adding tags, clearing and naming entities, and using other word embedding techniques such as word2vec, LDAP, and FastText. They also suggest using bigram or skip-gram tokenization, incorporating out-of-vocabulary words, and applying dimensionality reduction techniques such as PCA, as well as exploring other machine learning techniques not covered in this work.

Acknowledgments

A. Castelo thanks the financial support from the São Paulo Research Foundation (FAPESP) grants 2013/07375-0 and 2019/07316-0. A. Castelo and G. H. Darú acknowledges the support by ICMC - Institute of Mathematics and Computational Sciences - University of Sao Paulo (USP). G. H. Darú thanks for the support by Neogrid.

References

AGGARWAL, C. C.; ZHAI, C. A survey of text classification algorithms. *In*: AGGARWAL, C. C.; ZHAI, C. (ed.). *Mining text data*. New York: Springer, 2012. p. 163-222. DOI: https://doi.org/10.1007/978-1-4614-3223-4_6.

ALSMADI, I.; GAN, K. H. Review of short-text classification. *International Journal of Web Information Systems*, Bingley, v. 15, n. 2, p. 155-182, 2019. DOI: https://doi.org/10.1108/IJWIS-12-2017-0083.

BAEZA-YATES, R.; RIBEIRO-NETO, B. *Recuperação de Informação*: conceitos e tecnologia das máquinas de busca. 2. ed. Porto Alegre: Bookman Editora, 2013.

BENGIO, Y.; GRANDVALET, Y. No unbiased estimator of the variance of k-fold cross-validation. *Advances in Neural Information Processing Systems*, San Mateo, v. 16, p. 1-8, 2003.

BHAVANI, A.; KUMAR, B. S. A review of state art of text classification algorithms. *In*: INTERNATIONAL CONFERENCE ON COMPUTING METHODOLOGIES AND COMMUNICATION, 5., 2021, Erode. *Proceedings* [...]. [Piscataway]: IEEE, 2021. p. 1484-1490.

DARU, G. H. *Classificação produtos varejo CPG PTBR*. [*S. l.*]: Kaggle, 2022. Available from: https://www.kaggle.com/dsv/4265348. Access in: Dec. 28, 2022

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikitlearn: Machine learning in Python. *Journal of Machine Learning Research*, Cambridge, v. 12, p. 2825-2830, 2011.

PRANCKEVICIUS, T.; MARCINKEVICIUS, V. Comparison of naive bayes, random forest, decision tree, sup- port vector machines, and logistic regression classifiers for text reviews classification. *Baltic Journal of Modern Computing*, Latvia, v. 5, n. 2, p. 221, 2017. DOI: https://doi.org/10.22364/bjmc.2017.5.2.05.

ROSSUM, G. V.; DRAKE, F. L. Python 3 reference manual. Scotts Valley: CreateSpace, 2009.

SHAH, K.; PATEL, H.; SANGHVI, D.; SHAH, M. A comparative analysis of logistic regression, random forest and KNN models for the text classification. *Augmented Human Research*, [London], v. 5, n. 1, p. 1-16, 2020. DOI: https://doi.org/10.1007/s41133-020-00032-0.

SILVA, R. M.; SANTOS, R. L.; ALMEIDA, T. A.; PARDO, T. A. Towards automatically filtering fake news in portuguese. Expert Systems with Applications, Elmsford, v. 146, p. 113-199, 2020. DOI: https://doi.org/10.1016/j.eswa.2020.113199.

SONG, G.; YE, Y.; DU, X.; HUANG, X.; BIE S. Short text classification: a survey. *Journal of multimedia*, Oulu, v. 9, n. 5, p. 634-643, 2014.

ZHANG, Y.; JIN, R.; ZHOU, Z.-H. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, Berlin, v. 1, n. 1, p. 43-52, 2010.

Received: Nov. 8, 2022
Accepted: Dec. 26, 2022
Published: Dec. 27, 2022