Quantitative Analysis Powered by Naïve Bayes Classifier Algorithm to Data-Related Publications Social-Scientific Network

Tobias Ribeiro Sombra, Rose Marie Santini, Emerson Cordeiro Morais, Walmir Oliveira Couto, Alex de Jesus Zissou, Pedro Silvestre da Silva Campos, Paulo Cerqueira dos Santos Junior, Glauber Tadaiesky Marques, Otavio Andre Chase, SettingsJosé Felipe Souza de Almeida

Brazilian Institute of Information, Science and Technology (IBICT) - Federal University of Rio de Janeiro (UFRJ)

ABSTRACT

Quantitative evaluation of a dataset can play an important role in pattern recognition of technical-scientific research involving behavior and dynamics in social networks. As an example, are the adaptive feature weighting approaches by naive Bayes text algorithm. This work aims to present an exploratory data analysis with a quantitative approach that involves pattern recognition using the Mendeley research network; to identify logics given the popularity of document access. To better analyze the results, the work was divided into four categories, each with three subcategories, that is, five, three, and two output classes. The name for these categories came up due to data collection, which also presented documents with open access, dismembering proceedings, and journals for two more categories. As a result, the performance for the test examples showed a lower error rate related to the subcategory two output classes in the criterion of popularity by using the naive Bayes algorithm in Mendeley.

Keywords: Scientific Social Networks; Mendeley; Naïve Bayes; Machine Learning.

1. INTRODUCTION

Although an academic research article starts a study with a predetermined hypothesis, its research usually begins with the collection of data in which online social media tools can be some of the most rewarding and informative resources. Consequently, the use of social networks in scientific communication, whether for searching or sharing content in different areas of knowledge, has become a new reality in the last years (Nassi-Calò, 2017). Besides, this entire access procedure generates an adequate amount of digital information for data-based decision making, in which the methods of statistical pattern recognition are well suited to exploratory data classification methods (Jain et al., 2000).

Considering the advancement of social media in the popularity of scientific communication, a variety of platforms are turning attention to the academic community (Bik and Goldstein, 2013). At the same time, metrics such as webometry and altmetry are emerging in contrast to traditional impact factor measures based on bibliographic, which foster a culture of self-citation and citation cartels, neglecting their context, i.e., how and why certain articles are cited and, mainly, without observing their popularity (Nassi-Calò, 2017).

The work presented by Hoffmann et al. (2014), based on data from the academic research platform ResearchGate, a leading social network site (SNS) for scientists, presents results between relational metrics and others established impact measures, was tested in order to contribute for current debate on impact assessment based on online data altmetrics, focusing on a personal and relational network perspective. Although their measures of network centrality based on the analysis of social networks of online SNS were not considered in the context of impact assessment, their results on a small exploratory study suggest that such measures are related to established impact metrics and therefore can be useful, at least complementing existing forms of impact assessment.

Usually, the traditional evaluative bases of scientific articles use metrics to generate an impact factor, according to the number of citations that a journal receives. However, in this measurement of popularity, so to speak, the periodic is evaluated and not necessarily an individual survey. In order to propose a little of this but using pattern recognition techniques, the objective of this work is not exactly to measure the scientific impact but to contribute for the recognition of logics in the attributes of documents that help to identify patterns about accesses and their dynamics in the social networks. Moreover, this recognition will be done thinking about the popularity of most accessed documents (frequency), among which there may be cases about topics considered fashionable that are not accessed. Therefore, the survey will point to the popularity of access, regardless of the topic, and the SNS Mendeley will be used.

2. METHODOLOGY

The work of Wu et al. (2008) presents a ranking of some most influential algorithms in Data Mining identified and elected at the IEEE International Conference on Data Mining (ICDM), in December 2006. These ranking algorithms cover the tasks of classification, clustering, machine learning, association rules, link mining, which are among the essential topics in data mining research and development. Among these approaches to algorithms, we chose to apply the naive Bayes text classifier which has been widely used because of its simplicity in both the training and stage classifying (Remu *et al*. 2020; Pang and Bian, 2019; Mohammed *et al*. 2019; Sudha, 2019; Zang et al., 2016; Ting, 2011; Chakrabarti and Soundalgekar, 2003). In this sense, the pattern recognition method consists of three fundamental steps: data collection and selection, pre-processing, and data mining.

2.1. Mendeley Data Datasets

Mendeley's data repository is free-to-use and open access, which includes nearly 11 million indexed datasets. It enables us to deposit any research data, including raw and processed data, video, code, software, algorithms, protocols, and methods associated with the research manuscript (https://data.mendeley.com/datasets).

The first phase to do this was performed using an algorithm developed based on the Mendeley platform. This API allows the collection, as provided authentication between the user and the server. Subsequently, the algorithm performs authentication and starts the automatic collection. This collection is based on using the Mendeley API methods to perform queries, returning data regarding documents such as title, document type, year of publication, abstract, keywords, among others.

The entire collection is generated using links that are created according to the instruction available on the Mendeley website for developers. In the case of this study, we used the method search catalog, which allows a vast collection of documents, requiring the use of a specific parameter, which can be a title, author, source, or abstract. After this, it is necessary to separate the query words into four categories, as was observed to separate the documents which have open access concerning others.

Among other things, the research caused the breakdown of Proceedings and Journal creating two more categories, which were named Open_Proceedings and Open_Journal. This separation was possible due to a parameter in Mendeley called open access. If true for this rule is assigned, the return will be all documents that have open access in Mendeley. Table 1 shows the universe of 16,091,264 documents collected.

Table 1. Data categories with the total universe of documents found

Categories of data	Universe	Publication's last revision date
Open_Proceedings	3,416	26/07/2017
Proceedings	1,696,118	26/07/2017
Open_Journal	815,794	26/07/2017
Journal	13,575,936	26/07/2017

2.2. UCI Machine Learning Repository

UCI is a repository containing at least 100s of datasets from the University of California, School of Information and Computer Science. It classifies the datasets by the type of machine learning problem, allows us to find datasets for univariate and multivariate time-series datasets, classification, regression, or recommendation systems. Some of the datasets at UCI are already cleaned and ready to be used (https://archive.ics.uci.edu/ml/index.php).

The second phase of the method was based on the pre-processing in a series of actions to reduce noise in the data collected. In this context, pre-processing techniques support algorithmic research, improving efficiency, and facilitating the data mining process. At the same time, they allow the researcher to understand the nature of the data better to be mined.

All the procedures used were designed considering the criteria established by the naive Bayes algorithm, which presents a model based on the UCI machine learning repository database. This site is a repository of machine learning databases developed by the University of California Irvine that presents some standards for dataset composition. In this sense, a model for nominal data was used, since the naive Bayes algorithm has the supervised learning paradigm, and after all necessary treatments, such as selection and elimination of possible duplicate documents, the data collected was reduced.

Table 2 points to a significant reduction in data, which indicates that the collected universe contains noises that may impact other future processes. Therefore, it presents the subset that will be used when generating the files with the transformed data for classification by naive Bayes algorithm.

Table 2. Data categories with the universe, sample, and percentage of sample in relation to the universe after pre-processing

Categories of data	Universe	Sample	Sample (%)
Open_Proceedings	3,416	359	10.51%
Proceedings	1,696,118	70,615	4,16%
Open_Journal	815,794	166,450	20,40%
Journal	13,575,936	3,351,413	24,68%

2.3. Pre-processing and Data Mining

The third and final phase is responsible for handling the documents selected in the pre-processing to be adapted for the naive Bayes classifier. This adaptation basically consists of converting the attributes of the documents obtained to nominals, carrying out a process called discretization. Herein, such a process is based on establishing value ranges for numeric attributes, which allow us to try to adapt the best way possible to the Naive Bayes algorithm since it does not support the implementation of this type.

The naive Bayes classifier was implemented in Java code, and then experiments were carried out using the cross-validation method, in order to provide statistical support for the correct evaluation of the results, thus allowing the determination of a more suitable model for the proposed application (Dietterich, 1997). For this work, we developed a percentage discretization model, which basically consists of distributing each document to a respective output class based on the number of readers, in such a way that it can name the output classes.

Percentage values were obtained by searching the database for the document with the largest number of readers. After that, each document, the numerical value is converted into a percentage and then allocated to the corresponding value range. The calculation basically works as a rule of three, where a ratio is made between the value to be discretized and the highest value in the database. The results obtained in this process are allocated according to the appropriate range, which is divided into three subcategories, so-called: five output classes, three output classes, and two output classes. Table 3 shows the naming for the attributes of the classes in these subcategories.

Table 3. Subcategories with their respective possible classes

Subcategories	Possible classes
Five Output Classes	Not_Popular, Little_Popular, Popular, Very_Popular, Extremely_Popular
Three Output Classes	Not_Popular, Popular, Extramely_Popular
Two Output Classes	Not_Popular, Extremely_Popular

3. RESULTS AND DISCUSSION

In this work, a set of test examples was used considering the name of the subcategory. Thus, we chose to collect an example for each output class from the database at random. However, the result obtained after the classification, may not present the same class, since it will depend on the training of the classifier in the database. The classes in the examples follow the order to the naming of outputs, which are shown in Table 3, and their results will be presented in a specific graph for each subcategory.

3.1 Five Output Classes

Tables 4 to 7 show, as a measure of the random error, the confusion matrix, and the percentage of correctly classified examples (PCCE) for five output classes at Open_Proceedings, Proceedings, Open_Journal, and Journal, in that order.

Table 4. Confusion matrix and PCCE for the subcategory five output classes at Open_Proceedings

	Negative	Positive	Negative	Positive	Negative
Negative	**131**	21	10	12	4
Positive	46	**24**	6	0	1
Negative	27	6	**17**	6	3
Positive	7	2	1	**9**	2
Negative	6	3	3	0	**8**
		PCCE	53%		

Table 5. Confusion matrix and PCCE for the subcategory five output classes at Proceedings

	Negative	Positive	Negative	Positive	Negative
Negative	**50,009**	1,572	212	1	1
Positive	11,242	**2,192**	213	0	0
Negative	2,961	49	**216**	1	1
Positive	666	9	65	**9**	1
Negative	493	1	37	0	**4**
	PCCE		74%		

Table 6. Confusion matrix and PCCE for the subcategory five output classes at Open_Journal

	Negative	Positive	Negative	Positive	Negative
Negative	**46,353**	7,926	2,440	2,239	1.252
Positive	22,770	**9,780**	3,476	2,985	1,057
Negative	11,547	6,536	**4,379**	4,221	1,579
Positive	6,189	4.,015	3,458	**6,077**	3,293
Negative	2,323	1,268	1,322	3,821	**6,139**
		PCCE	44%		

Table 7. Confusion matrix and PCCE for the subcategory five output classes at Journal

	Negative	Positive	Negative	Positive	Negative
Negative	**683,043**	160,138	42,382	390,240	240,703
Positive	257,658	**334,004**	204,221	154,326	10.242
Negative	4,636	16,810	**405,076**	6,645	407
Positive	52,020	49,866	33,831	**111,331**	81,139
Negative	31,018	22,390	11,047	6,531	**41,709**
		PCCE	47%		

When looking at the tables, it is clear that the PCCE was between 40% and 60%, with the exception of Table 5 with 74%. All in all, however, everything points to this being the best solution, because it is a workable solution for this subcategory due to the lower margin of error. Figures 1 to 4 show the results found by the algorithm after the execution of the test examples for subcategory five output classes.

Figure 1. Test examples for the subcategory five output classes at Open_Proceedings.

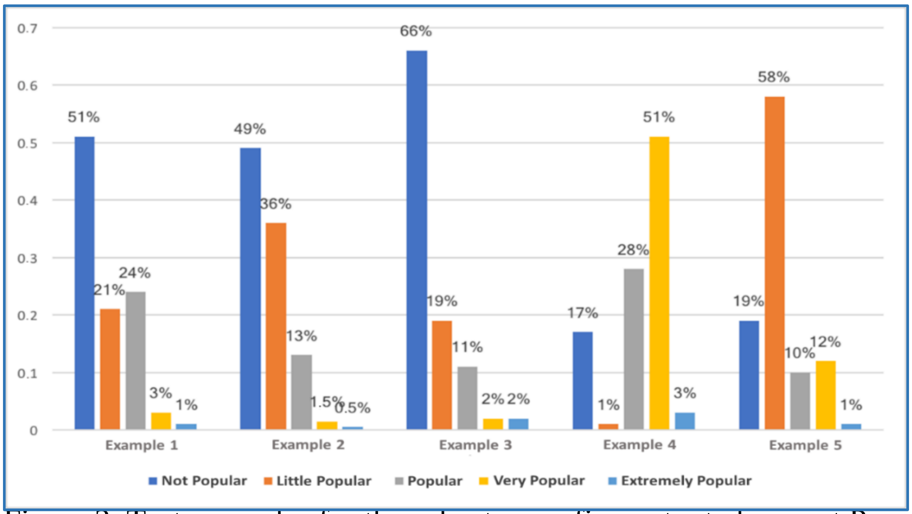


Figure 2. Test examples for the subcategory five output classes at Proceedings.

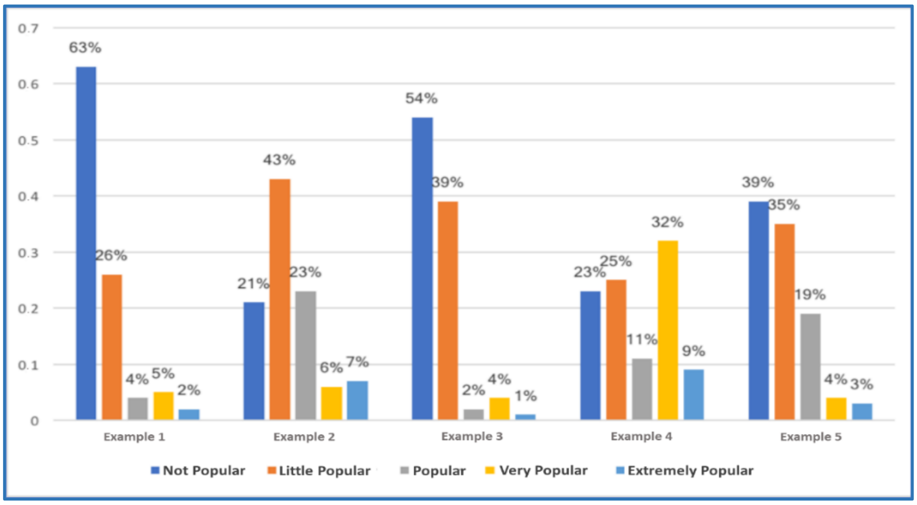


Figure 3. Test examples for the subcategory five output classes at Open_Journal.

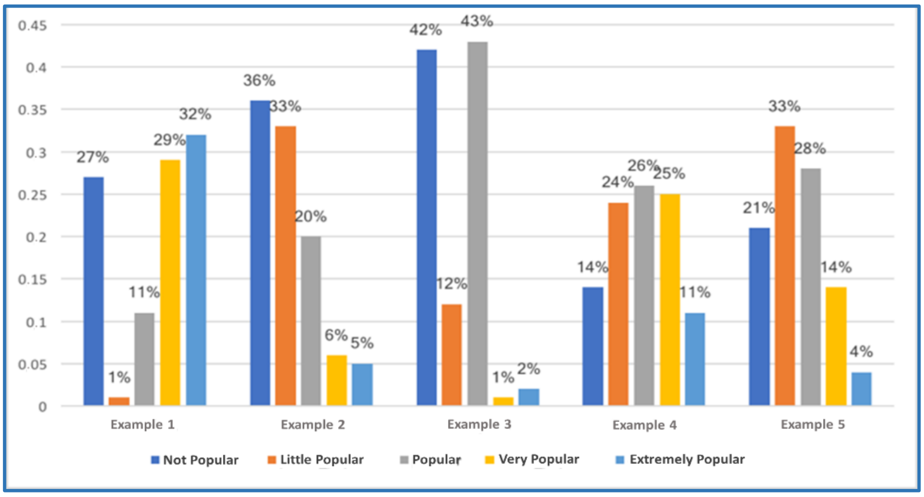


Figure 4. Test examples for the subcategory five output classes at Journal.

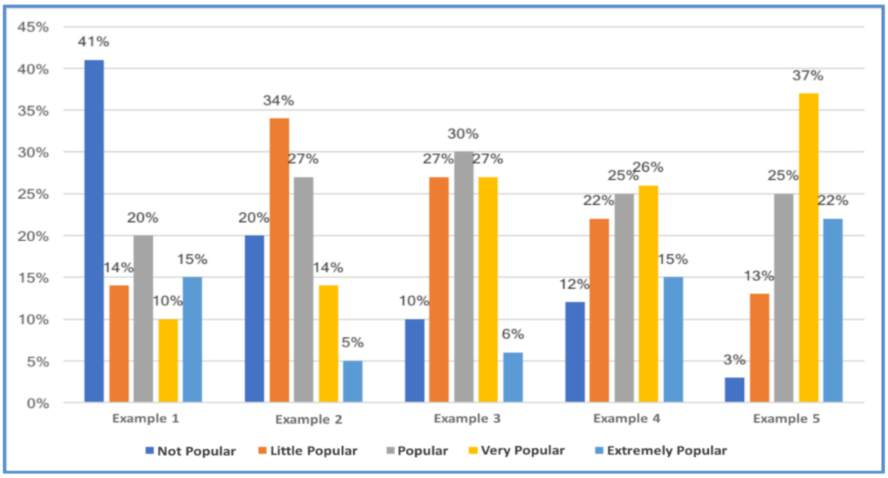


Figure 1 shows the algorithm performance for examples 1, 2, and 3 as not popular, while example 4 presented the class very popular, and example 5 as little popular. In Figure 2, it can be seen that the naive Bayes algorithm foresaw examples 1, 3, and 5 as not popular, for example 2 as little popular, and example 4 as very popular. In Figure 3, it can be seen that the algorithm classified example 1 as extremely popular, example 2 as not popular, examples 3 and 4 as popular, and example 5 was classified as a little popular. Finally, in Figure 5, it is noted that the algorithm classified example 1 as not popular, example 2 as a little popular, example 3 as popular, and examples 4 and 5 as very popular.

3.2. Three Output Classes

Table 8. Confusion matrix and PCCE for the subcategory three output classes at Open_Proceedings.

	Negative	Positive	Negative
Negative	**124**	20	34
Positive	63	**32**	19
Negative	20	7	**37**
	PCCE	54%	

Table 9. Confusion matrix and PCCE for the subcategory three output classes at Proceedings.

	Negative	Positive	Negative
Negative	**50,217**	1,018	560
Positive	11,296	**1,767**	584
Negative	3,978	558	**637**
	PCCE	75%	

Table 10. Confusion matrix and PCCE for the subcategory three output classes at Open_Journal.

	Negative	Positive	Negative
Negative	**59,313**	23,540	594
Positive	22,859	**43,238**	2,030
Negative	1,528	9,802	**3,543**
	PCCE	64%	

Table 11. Confusion matrix and PCCE for the subcategory three output classes at Journal.

	Negative	Positive	Negative
Negative	**1,060,913**	420,061	221,919
Positive	154,063	**739,921**	121,656
Negative	189,649	416,461	**226,770**
	PCCE	60,5%	

The PCCE values presented in Tables 9, 10, and 11 show a smaller error result for the subcategory three output classes when compared to five output classes. Figures 5 to 8 present the results found by the algorithm after the execution of the test examples for three output classes.

Figure 5. Test examples for the subcategory three output classes at Open_Proceedings.

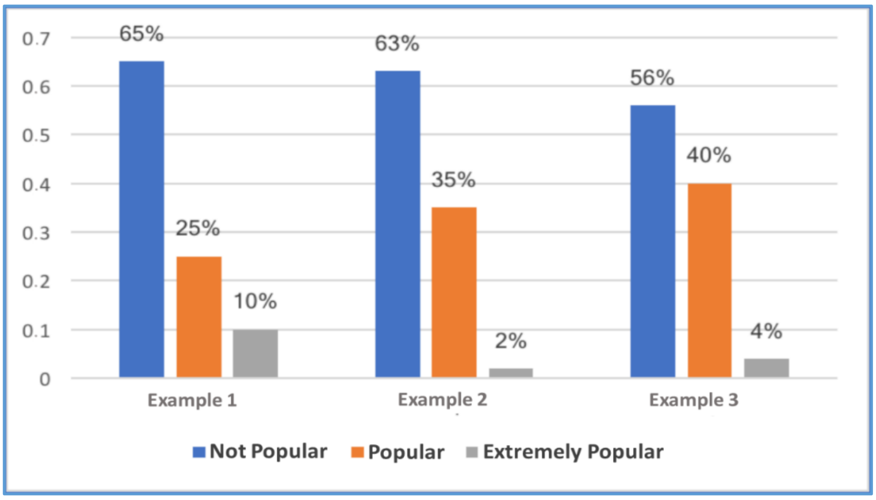


Figure 6. Test examples for the subcategory three output classes at Proceedings.

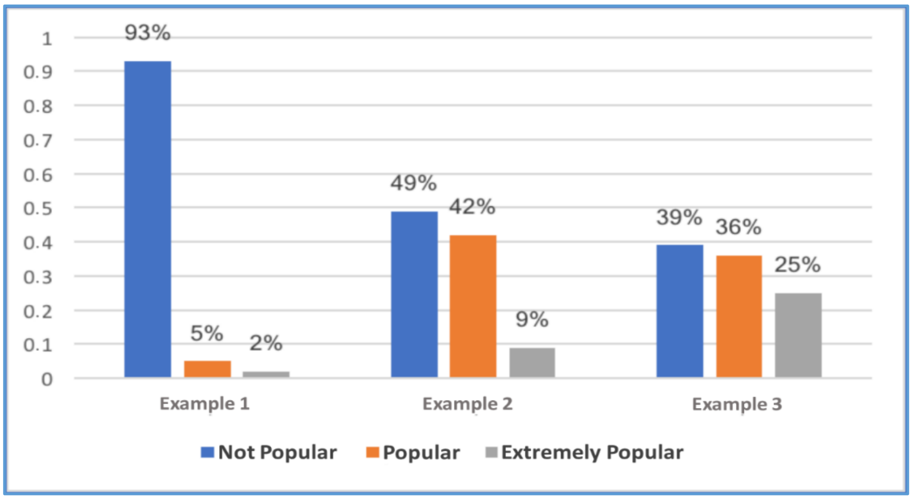


Figure 7. Test examples for the subcategory three output classes at Open_Journal.

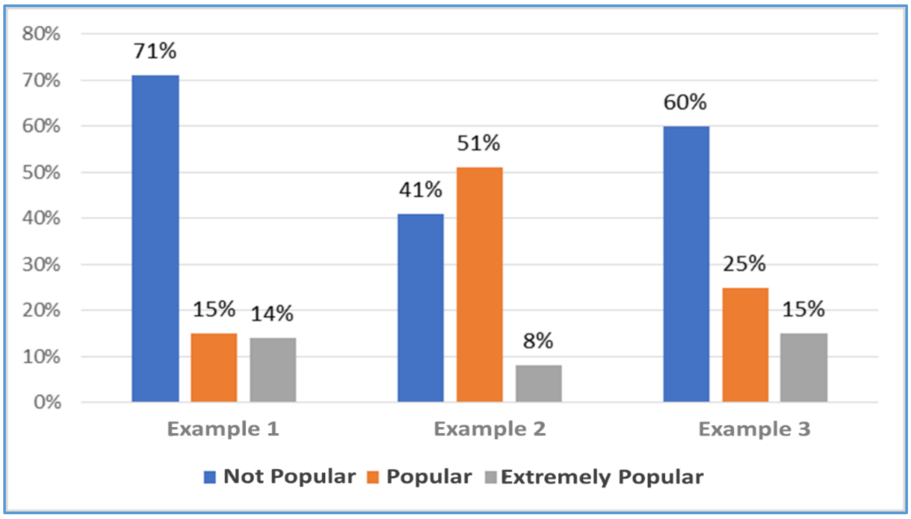
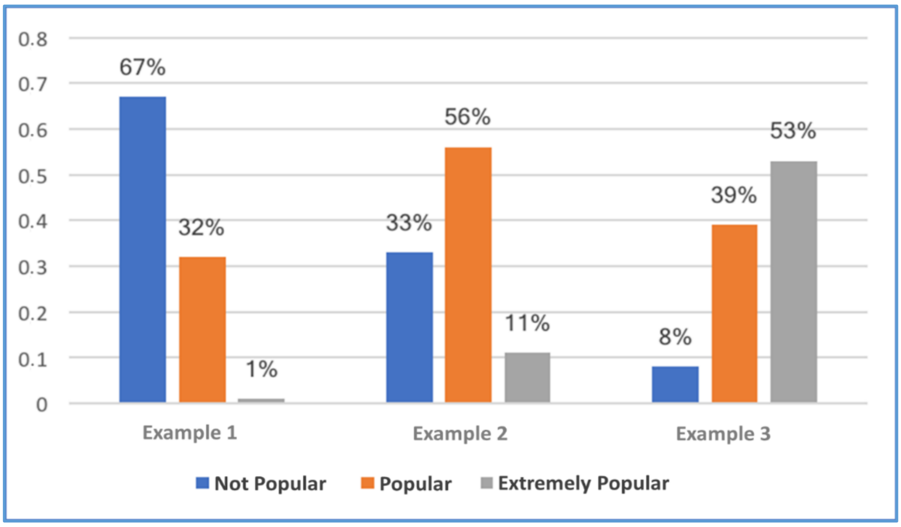


Figure 8. Test examples for the subcategory three output classes at Journal.



Figures 5 and 6 show the best algorithm performance for the test examples as not popular. While example 4 presented the class very popular, and example 5 as a little popular. In Figure 7, as well as in Figure 8, it is noted that the algorithm classified example 1 as not popular, example 2 as popular and 3 as extremely popular.

3.3. Two Output Classe

Table 12. Confusion matrix and PCCE for the subcategory two output classes at Open_Proceedings.

	Negative	Positive
Negative	**214**	41
Positive	56	**43**
PCCE	73%	

Table 13. Confusion matrix and PCCE for the subcategory two output classes at Proceedings

	Negative	Positive
Negative	**42,404**	9,391
Positive	7,196	**11,624**
PCCE	77%	

Table 14. Confusion matrix and PCCE for the subcategory two output classes at Open_Journal

	Negative	Positive
Negative	**98,316**	13,989
Positive	25,643	**28,490**
PCCE	76%	

Table 15. Confusion matrix and PCCE for the subcategory two output classes at Journal

	Negative	Positive
Negative	**2,040,763**	245,312
Positive	525,513	**539,825**
PCCE	77%	

This subcategory showed a significant increase in PCCE when compared to the subcategories, three classes of output, and five classes of output. The algorithm still has a margin of error, but much smaller compared to the others. Figures 9 to 12 show the results found by the algorithm after running the test examples for the subcategory two output classes.

Figure 9. Test examples for the subcategory two output classes at Open_Proceedings.

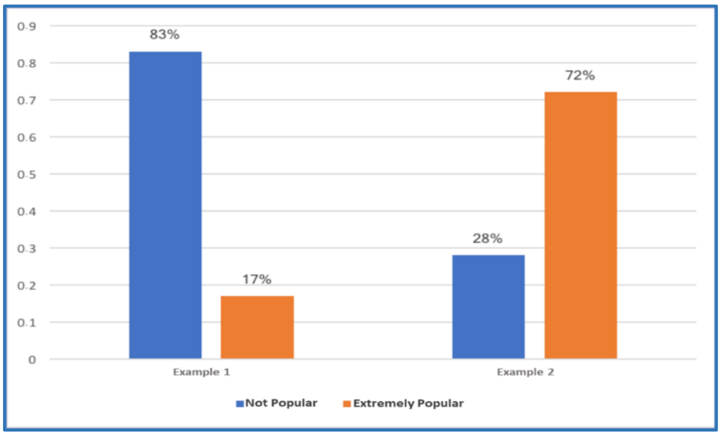


Figure 10. Test examples for the subcategory two output classes at Proceedings.

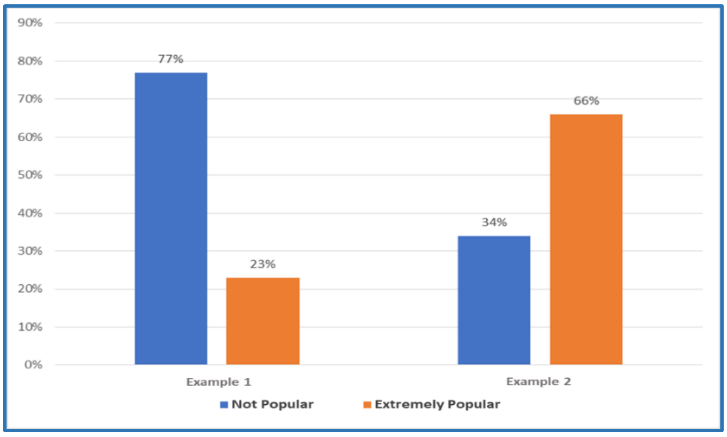


Figure 11. Test examples for the subcategory two output classes at Open_Journal.

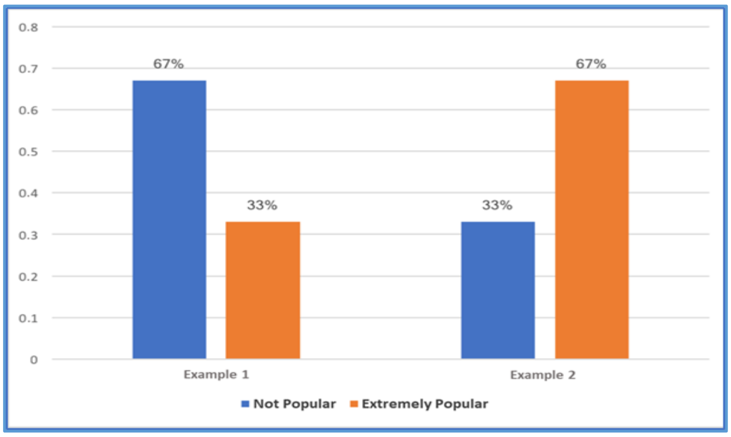
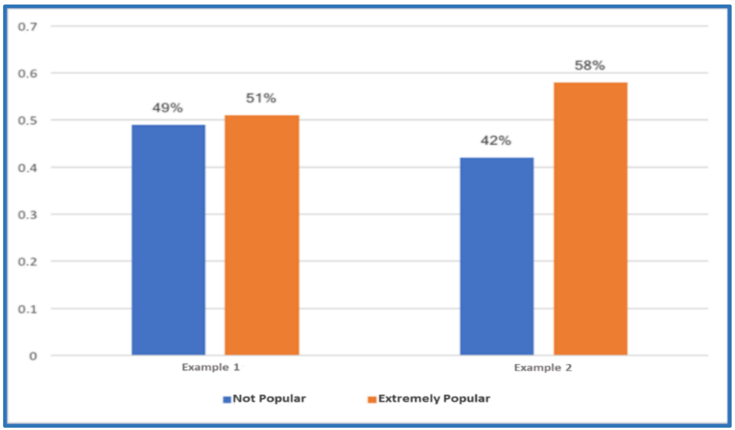


Figure 12. Test examples for the subcategory three output classes at Journal.



Figures 9, 10, and 11 show the performance by classifier algorithm for the test examples as not popular and extremely popular, respectively. In Figure 10, both tests were rated as extremely popular. All data categories showed better results in this situation.

4. FINAL CONSIDERATIONS

While citations appear as the relational metric of influence in the impact factor of the evaluative bases of scientific articles, the popularity of scientific themes has a strong relationship with the statistical analysis of scientific social networks. Thus, this work presented an exploratory research that involved the classification of scientific documents using the Mendeley database and the Naive Bayes algorithm. Based on the results, the percentage discretization model reiterates that the subcategory "two output classes" showed better results, taking into account that the PCCE was higher compared to the other subcategories. The training examples in this case showed better distribution, leaving the subcategory more balanced. It is worth mentioning that this proposal does not intend to enter into the merits of the discussions about impact factors, although it presents a contribution within the scope of the SNS and its importance by using naive Bayes classifier.

REFERENCES

CHAKRABARTI, S. R. S., and SOUNDALGEKAR, M.V. "Fast and Accurate Text Classification Via Multiple Linear Discriminant Projection," **International Journal on Very Large Data Bases**, pp. 170-185, 2003.

DIETTERICH, T.G. "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms," **Neural Computation**, v. 10, pp. 1895-1924, 1997.

HOFFMANN, C. P., LUTZ, C., and MECKEL, M. "Impact Factor 2.0: Applying Social Network Analysis to Scientific Impact Assessment," In: **47th Hawaii International Conference on System Sciences**, Waikoloa/ HI, pp. 1576-1585, 2014.

JAIN, A. K., DUIN, R., and MAO, J. "Statistical Pattern Recognition: A Review," **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 22, pp. 4-37, 2000.

NASSI-CALÒ, L. "Evaluation metrics in science: current status and prospects." **Latin American Journal of Nursing**, v. 25, 2017.

MOHAMMED Z., FARHAZ, M., IRSHD, M., BASTHIKODI, M., and FAIZABADI, A. R. A. "Comparative Study for Spam Classifications in E-mail Using Naïve Bayes and SVM Algorithm," **Journal of Emerging Technologies and Innovative Research (JETIR).** v. 6, pp. 391-393, 2019.

PANG, J. and BIAN, J. Android Malware Detection Based on Naïve Bayes**.** In: **IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)**, Beijing, China, p. 483-486, 2019.

REMU, R. H., OMAR, F., FERDOUS, R., ARIFEEN, M., SAKIB, S., and Reza, S. M. S. "Naive Bayes based Trust Management Model for Wireless Body Area Networks," **International Conference on Computing Advancements (ICCA)**, New York, USA, pp. 1-4, 2020.

SUDHA, M. **Applied Computational Intelligence**. Dwarka/ND, India, Educreation Publishing, 2019.

TING, S.L. IP, W.H. and TSANG, A. "Is Naïve Bayes a Good Classifier for Document Classification?" **International Journal of Software Engineering and its Applications**, v. 5, 2011.

ZHANG, L., JIANG L., LI, C. and KONG, G. "Two feature weighting approaches for naive Bayes text classifiers," **Knowledge-Based Systems**, pp. 137–144, 2016.

BIK, H.M. and GOLDSTEIN, M.C. "An Introduction to Social Media for Scientists," **PLoS Biology**, v. 11, n. 4, 2013.

WU, X., KUMAR, V., ROSS, Q. J., GHOSH, J., YANG, Q., MOTODA, H., MCLACHLAN, G. J., ANGUS N., LIU, B., YU, P. S., ZHOU, Z., STEINBACH, M., HAND, D. J., and STEINBERG, D. "Top 10 Algorithms in Data Mining. Knowledge and Information Systems," v. 14, pp. 1-37, 2008.