Working Documents

Automatic text classification using text-mining techniques: Application to the definitions used in the National Employment Survey of Chile

Authors:
Julio Guerrero
Julián Cabezas

No. 6, March 2019

Instituto Nacional de Estadísticas · Chile

NATIONAL STATISTICS INSTITUTE

Morandé 801, Santiago de Chile
Telephone: 56 232461000
E-mail: ine@ine.cl
Facebook: @ChileINE
Twitter: @INE_Chile

Julio Guerrero
Julián Cabezas
Technical Sub-directorate

The authors would like to thank the person in charge of this initiative, Jose Luis Aránguiz, the analysts of the Department of Household Statistics, Nicolás Von Hausen and Nicolás Maturana, and the Head of the Department of Research and Development, Denisse López. Their comments and their interest in the adoption of new technologies made this work possible. The authors also acknowledge the participation and support of colleagues from the Department of Statistical Infrastructure, the Department of Labour Studies, the Department of IT, the Department of Family Budget, and the Department of Research and Development, all of whom provided many hours of work and basic resources that contributed to the success of this study.

INE's Working Documents are addressed to researchers, academics, students, and specialists in economics. They are intended to provide a comprehensive analysis of key conceptual, analytical, and methodological aspects of INE's statistical products and to contribute to the exchange of ideas between the various components of the National Statistical System.

The interpretations and opinions expressed in Working Papers belong exclusively to the authors and their collaborators. They do not necessarily reflect the official position of INE or any other institution to which the collaborators of the documents belong.

Automatic text classification using text-mining techniques: Application to the definitions used in the National Employment Survey of Chile

Abstract

This document explores the methodological aspects of the automatic classification of texts, a task that consists of assigning free text documents to one or more predefined classes based on their content. To this end, the use of three machine-learning techniques is described: naïve Bayes (NB), support vector machine (SVM), and random forests (RF). The study analyzes the particular properties of learning with text data and identifies why these techniques are appropriate for this task. These techniques were evaluated empirically to support the theoretical findings, using the classification of the "profession, job, or occupation" and the "economic sector" of the employed population, which is based on data from the National Employment Survey (ENE) collected in 2017 by the National Statistics Institute (INE). All the evaluated techniques performed well in the classification. SVM performed best, its overall precision approximately 90%. SVM showed consistent precision in a wide variety of situations, and it is completely automatic, eliminating the need for manual adjustment of parameters.

Keywords: text mining, classification of texts, machine learning, national employment survey

1. Introduction

With the rapid growth of online information, text classification has become one of the key techniques for managing and organizing text data. Text categorization techniques are used to classify news, to find information of interest on the World Wide Web (WWW), and to guide user searches through hypertext (Joachims, 1998). Trained professionals are used to classify new items, but this process is costly and time consuming, limiting its utility. Consequently, there has been a growing interest in the development of technologies for automatic text classification (Aas & Eikvil, 1999).

For these purposes, a number of statistical-classification and machine-learning techniques have been applied to the classification of texts, including regression models (Aas & Eikvil, 1999), classifiers based on nearest neighbors (Aas & Eikvil, 1999; Pérez, 2017), decision trees (Pérez, 2017), Bayesian classifiers (Aas & Eikvil, 1999; Mitchell, 1997), rule learning algorithms (Aas & Eikvil, 1999), support vector machines (Joachims, 1998; Berry & Kogan, 2010), and neural networks (Aas & Eikvil), among others.

This same situation has been experienced at the National Statistics Institute (INE). INE, in the exercise of its public role, needs to develop and implement technological tools that address the tasks of coding large volumes of texts from open questions in the surveys that it conducts. This need has arisen within the processes required for the production of labor and sociodemographic statistics with strict publication deadlines. Currently, the coding process is performed manually and has a precision of approximately 84%. This process requires more than 3,600 working hours per month. For this reason, a solution is needed that outperforms manual sorting in efficiency and speed.

To assess which solution should be implemented at INE, it was decided to measure the quality and progress in the field of text classification, which requires a standardized collection of documents for analysis and testing (Aas & Eikvil, 1999). Within this framework, this study applies the automatic text classification procedure to a data set containing the definitions (documents) on the "trade, job, or occupation" and the "economic sector" of the employed population, based on data from the National Employment Survey (ENE) collected in 2017 by INE. Thus, this study seeks to 1) describe the steps inherent to an automatic text-classification process following a text-mining scheme; 2) analyze appropriate algorithms for text classification (i.e., machine learning algorithms such as naïve Bayes (NB), support vector machine (SVM) and random forests (RF); 3) build automatic classifiers for the definitions used in the ENE, based on the above techniques; and 4) evaluate the performance of the classifiers through the use of statistical metrics and select the most appropriate.

This study will analyze the methodological aspects related to an automatic text classification problem, ranging from the transformation of texts into an adequate representation for classification tasks—which has been traditionally addressed using a vector space model because of its simplicity and good performance (Aas & Eikvil, 1999; Alfaro & Allende, 2011; Welbers et al., 2017)—to the application of a classification technique and its subsequent evaluation.

This document is divided into six sections. The first section is the introduction to the study. Section 2 outlines the steps necessary for transforming raw texts into a representation suitable for classification tasks. Section 3 describes three successful techniques that have been chosen for the purposes of this study. Section 4 introduces performance metrics for the assessment of classification in a binary problem. Section 5 presents the experiments conducted using the ENE 2017 data set. Finally, section 6 provides conclusions and projections based on the results.

2. Data Preparation

Data preparation is the starting point of any statistical data analysis. Computational text analysis is no different in this respect, and it often presents some special challenges that can be daunting for both beginning and advanced analysts. Furthermore, preparing texts for analysis requires making decisions that can affect the precision, validity, and findings of a text-analysis study and can determine the techniques and methods used in the analysis (Welbers et al., 2017).

This section describes some preprocessing procedures and criteria for the selection of features and representation of texts. It then addresses the effectiveness of classification measures for class filtering.

2.1. Preprocessing

The initial step in text classification is to transform documents (typically, strings of characters) into a representation suitable for the learning algorithm and the classification task. The transformation of texts proceeds as follows (Aas & Eikvil, 1999; Welbers et al, 2017):

First, the process of **text normalization** is applied. In this process, words are transformed into a more uniform format so that a classifier can recognize when two words have (approximately) the same meaning, even if they are written in a slightly different form. Text normalization reduces the range of the vocabulary (i.e., the full range, or dimension, of the features used in the analysis). The process includes the following steps, whose order of execution is not arbitrary: (1) convert all text to lowercase, (2) remove html or other tags (special characters), (3) remove punctuation marks, (4) remove numbers, and (5) remove multiple blank spaces.

Next is **tokenization**, a process that consists of dividing a text or document into more specific features known as tokens, which are typically words or word combinations that constitute the most significant semantic components of texts. Because full texts are too specific to perform any meaningful calculations, tokenization is crucial to computational analysis.

2.2. Selection of features

In text classification, the selection of appropriate features (a word or token in a document) can be quite useful. Features are selected according to their contributions to class discrimination. If not selected, they are removed from the data in order to learn from and implement models. The two objectives in the selection of features are to reduce the dimensionality in the space of document features and to filter out irrelevant features. These objectives help to build an accurate and efficient model for document classification.

The reduction of dimensionality seeks to decrease the number of features to be modeled while preserving the content of individual documents. This generally helps to speed up the training process of a model. Filtering, meanwhile, is valuable for machine-learning algorithms such as RBF neural networks, which treat every data feature equally in their distance calculations and are therefore unable to distinguish between relevant and irrelevant features.

2.2.1. Procedure

The selection of features consists of two steps (Berry & Kogan, 2010):

1. For a given set of data, features are extracted and selected under an unsupervised scheme[1] by first removing common or high frequency words (called stop words), which are words that do not convey information about the content of the document (i.e., pronouns, articles, prepositions, conjunctions, etc.), and then applying a stemming procedure that removes suffixes to generate “stem or origin words”. In the latter process, words with the same conceptual meaning, such as “do” and “doing”, are grouped together.

2. In the documents, or corpus (i.e., the collection of documents), features whose frequency is below a defined threshold are removed from the data set. Such features do not assist in the differentiation of documents by class, and they can add noise to document classification. The selection process also eliminates features with very high frequency in the corpus of the data set because many of these features are distributed almost equally among the various classes and therefore are not valuable for characterizing the classes of the features. The features are then selected by their frequency distributions among the training documents of the different classes. Using the labeled training documents, this supervised feature-selection procedure seeks an

[1] An unsupervised scheme refers to automatic learning in which a model is adjusted to observations without *a priori* knowledge of the labels, or classes, to which they belong.

improved identification of the features with the greatest power to discriminate between classes.

2.2.2. Methods for the selection of features

Several supervised methods[2] for feature selection have been widely used in text classification (Sebastiani, 2002). These include the following metrics: information gain (IG), Chi-square statistic (CHI), and Odds Ratio (OR).

Information Gain (IG)

The IG criterion quantifies the amount of information obtained for class prediction by ascertaining the presence or absence of a feature in the document. The IG of a feature t in a class c can be expressed as:

$$IG(t,c) = \sum_{c\in\{c,\bar{c}\}} \sum_{t\in\{t,\bar{t}\}} P(t,c)\, log\left(\frac{P(t,c)}{P(t)P(c)}\right) \quad \textbf{(1)}$$

Where $P(c)$ and $P(t)$ denote the probability that a document belongs to class c and the probability that a feature t occurs in a document, respectively. $P(t,c)$ denotes the joint probability of t and c.

All probabilities can be estimated by frequency counts from the training data.

Chi-square statistic (CHI)

Another popular method of selecting features is the CHI statistic. This statistic measures the lack of independence between the occurrence of feature t and the occurrence of class c. The features are classified according to the following quantity:

[2] A supervised method is a technique that learns from a set of labeled examples (documents) whose class is known *a priori*.

$$CHI(t,c) = \frac{n[P(t,c)P(\bar{t},\bar{c}) - P(t,\bar{c})P(\bar{t},c)]^2}{P(t)P(\bar{t})P(c)P(\bar{c})} \quad \textbf{(2)}$$

Where n is the size of the training data, and the probability notations have the same interpretation as in equation (**1**). For example, $P(\bar{c})$ represents the probability that a document does not belong to class c.

Odds Ratio (OR)

The third feature-selection criterion, OR, has also been used in text classification. It measures the ratio between the probability of the occurrence of feature t in a document of class c and the probability that the feature does not occur in c. It can be defined as follows:

$$OR(t,c) = \frac{P(t \mid c)\left(1 - P(t \mid \bar{c})\right)}{\left(1 - P(t \mid c)\right)P(t \mid \bar{c})} \quad \textbf{(3)}$$

The effectiveness of feature-selection methods for text classification has been studied and compared, for example, by Yang & Pedersen (1997), who concluded that information gain (IG) produces the most stable results.

2.3. Representation of documents

Because of its simplicity and effectiveness, the vector space model is perhaps the most commonly used method of document representation (Aas & Eikvil, 1999). In this model, the documents are represented by word vectors. Normally, this is a collection of documents represented by a document—by—word matrix denoted by $\boldsymbol{A}$—where each entry represents the occurrences of a word in the document. It can be expressed as follows:

$$\boldsymbol{A} = (a_{dt}) \tag{4}$$

Where a_{dt} is the weight of word t in document d. Because each word does not normally appear in every document, matrix $\boldsymbol{A}$ is usually a sparse matrix (i.e., a large matrix in which most of its elements are zero). The number of columns (N) in the matrix corresponds to the number of words in the dictionary, so it can be a very large number.

Matrix $\boldsymbol{A}$ is one of the most common formats for representing the corpus of a text in a bag-of-words (BOW) format. The advantage of this representation is that it allows data to be analyzed with vector and matrix algebra, effectively moving from text to numbers. When using special matrix formats for sparse matrices, text data in $\boldsymbol{A}$ format are quite efficient in memory, and they can be analyzed with highly optimized operations (Welbers et al, 2017).

There are several ways to determine the weight a_{dt} of word t in document d (Aas & Eikvil, 1999), but most approaches are based on two empirical observations regarding the text:

1. The more times a word occurs in a document, the more relevant it is to the topic of the document.
2. The more times the word occurs throughout all the documents in the collection, the more poorly it discriminates between documents.

Let f_{dt} be the frequency of the word t in document d, M be the number of documents in the collection, N be the number of words in the collection after the stopwords have been removed and a stemming procedure has been performed (see section 2.2.), and n_t be the total number of times the word t occurs in the entire collection.

The simplest approach is to assign a weight of one if the word appears in the document, and a weight of zero otherwise. This can be expressed as follows:

$$a_{dt} = \begin{cases} 1, & si\, f_{dt} > 0 \\ 0, & e.o.c. \end{cases} \tag{5}$$

Another simple approach is to use the word frequency in the document,

$$a_{dt} = f_{dt} \tag{6}$$

However, these two schemes do not take into account the word frequency across all the documents in the collection. A well-known approach to calculating word weights is word weighting ‘$tf - idf$’, which assigns weights to the word t in document d in proportion to the number of occurrences of the word in the document, and in inverse proportion to the number of documents in the collection where the word occurs at least once. This scheme for weighting terms produces good classification results, and therefore it has been selected for this work. It can be calculated as follows:

$$a_{dt} = f_{dt} \cdot log\left(\frac{M}{n_t}\right) \tag{7}$$

3. Methods of text classification

Text classification is the problem of automatically assigning one or more predefined classes to free text documents (Aas & Eikvil, 1999). The more text information available, the more its effective retrieval becomes difficult without good indexing and a summary of the document's content. Document classification is one solution to this problem. A growing number of statistical methods and machine-learning techniques have been applied to text classification in recent years.

Most research in this area has been devoted to binary issues, where a document is classified as relevant or irrelevant with respect to a predefined topic of interest. However, many sources of textual data (such as internet news, e-mails, and digital libraries) consist of a range of subjects. They therefore pose the problem of classification of multiple classes.

A common approach to the problem of classification with multiple classes is to separate each problem into a binary classification, one for each class. For the methods naïve Bayes and support vector machine in particular, the classification of a new document requires the application of all binary classifiers and the consolidation of their predictions into a single decision. The result is a ranking of possible topics according to the probability of their belonging to each class.

The sections below describe some of the algorithms for text classification that have been proposed and evaluated in the past. The following algorithms have been considered as alternative classifiers in this work: naïve Bayes, support vector machines, and random forests. It should be noted that these algorithms operate under a supervised scheme (i.e., classifiers are trained with examples (documents) whose class has previously been labeled).

First, some general notation: Let $d = \{d_1, \ldots, d_m\}$ be a vector of documents to be classified and $c_1, \ldots, c_k$ be the possible classes. It is assumed that each document d_i can be expressed as a numerical vector representing the weights of terms or features $d_i = \{t_1, \ldots, t_m\} \in \Re^n$ (see section 2.3.)

3.1. Naïve Bayes

The naïve Bayes (NB) classifier is a probabilistic learning algorithm derived from Bayesian decision theory (Mitchell, 1997). The probability of a document d of class c denoted by $P(c|d)$ is calculated as follows:

$$\mathrm{P(c|d)} \propto P(c)\prod_{k=1}^{m} P(t_k|\mathrm{c}) \qquad \textbf{(8)}$$

Where $P(t_k|\ \mathrm{c})$ is the conditional probability that feature t_k occurs in a document of class c, and $P(c)$ is the *a priori* probability that a document occurs in class c. $P(t_k|\ \mathrm{c})$ can be used to measure how much evidence t_k provides that c is the correct class (Manning et al., 2008). In document classification, the class to which the feature belongs is determined by finding the most probable class, or maximum *a posteriori* (MAP), c_{MAP} , defined by:

$$c_{MAP} = \operatorname*{argmax}_{c\in c_k} P(c|d) = \operatorname*{argmax}_{c\in c_k} P(c)\prod_{k=1}^{m} P(t_k|\mathrm{c}) \qquad \textbf{(9)}$$

Equation (**9**) involves the multiplication of many conditional probabilities, one for each feature. In practice, the multiplication of probabilities often becomes a sum of logarithms of probabilities. Therefore, the maximization of the equation can also be determined by the following expression:

$$c_{MAP} = \operatorname*{argmax}_{c\in c_k}\left[\log P(c) + \sum_{k=1}^{m} \log P(t_k|\mathrm{c})\right] \qquad \textbf{(10)}$$

All model parameters (i.e., the *a priori* classes and probability distributions of the features) can be estimated with relative frequencies from training set d. Note that when a class and a document feature do not occur together in the training set, the probability estimate, based on the corresponding frequency, will be zero. This would leave the right side of equation (**10**) undefined. This problem can be mitigated by incorporating corrections, such as Laplace smoothing, into all probability estimates.

NB is a simple probability-learning model that can be implemented efficiently with linear complexity. A simplistic (naïve) assumption is that the presence or absence of a feature in a class is completely independent of any other feature. Despite the frequent imprecision of its oversimplified assumption (particularly for text domain problems), NB is one of the most widely used classifiers and has several properties that make it surprisingly useful and accurate (Berry & Kogan, 2010).

3.2. Support Vector Machine

Support vector machine (SVM) has been considered one of the most promising algorithms in text classification (Berry & Kogan, 2010). SVMs are linear classifiers that operate in a space of high-dimensional features. This space is a nonlinear mapping of the input space of the problem in question. In the transformed space, an SVM builds a hyperplane of separation that maximizes the distance between the training samples of two classes by selecting two parallel hyperplanes that are tangent to at least one sample of their kind. Such samples in the tangent hyperplanes are called support vectors. The distance between the two tangent planes is the margin classifier, which must be maximized. Thus, a linear SVM is also known as a maximum-margin classifier. An advantage of working in a high-dimensional feature space is that, in many problems, the nonlinear classification task in the original input space becomes a linear classification task in the high-dimensional feature space. SVM works in the high-dimensional feature space without incorporating any additional computational complexity.

SVM's strength comes from two key properties it possesses: kernel representation and margin optimization. In SVMs, a kernel function can assign a high-dimensional feature space and learn the classification task in that space without any additional computational complexity. A kernel function can also represent the dot product of two data-point projections in a high-dimensional feature space. Which high-dimensional space to use depends on the selection of a specific kernel function. The classification function used in SVMs can be written in terms of the dot products of the input data points. Therefore, using a kernel function, the classification function can be expressed in terms of dot products of the projections of input data points in a high-dimensional feature space. Kernel functions do not explicitly assign data points to higher-dimensional space. Instead, they provide SVMs with the advantage of learning the classification task in that higher-dimensional space.

The second key property of SVMs is the way in which they achieve the best classification function. SVMs minimize the risk of overfitting of training data by determining the classification function (a hyperplane) with a maximum margin of separation between the two classes. This property provides SVMs with a very powerful capacity for the generalization of classification.

This applies only to binary classification tasks. Therefore, the use of SVMs for text classification (a multi-class problem) must be approached as a series of binary classification problems.

In SVMs, the classification function is a hyperplane that separates the different classes of data.

$$\langle \boldsymbol{w}, x \rangle + b = 0 \qquad \textbf{(11)}$$

The notation $\langle \boldsymbol{w}, x \rangle$ represents the dot product between the coefficients of the normal vector $\boldsymbol{w}$, which is perpendicular to the hyperplane and the vector of variables x. The scaling (b) is a term that refers to the bias.

The solution to the classification problem is then specified by the normal vector $\boldsymbol{w}$. It is possible to demonstrate that vector $\boldsymbol{w}$ can be written as a linear combination of the data points x_i, where $i = 1, \dots, m$, (i.e., $\boldsymbol{w} = \sum_{i=1}^{m} \alpha_i x_i \, , \alpha_i \geq 0$). The data points x_i with non-zero α_i are called support vectors.

A kernel function k can be defined as $k(x_1, x_2) = \langle \Phi(x_1), \Phi(x_2) \rangle$, where $\Phi: X \rightarrow H$ is a mapping of points in input space X into hyper-dimensional space H. As can be seen, the kernel function implicitly maps the input data points into a higher-dimensional space and returns the dot product without mapping or calculating them. There are many suggested kernel functions for SVM. Some of the kernel functions widely used in the literature are linear function, $k(x_1, x_2) = \langle x_1, x_2 \rangle$; Gaussian radial base function (RBF), $k(x_1, x_2) = e^{-\sigma \|x_1 - x_2\|^2}$, and polynomial function, $k(x_1, x_2) = \langle x_1, x_2 \rangle^d$. The selection of a specific kernel function for an application depends on the nature of the classification task and the input data set. As can be inferred, the performance of SVMs depends largely upon which specific kernel function is used. According to Joachims (1998), the performance of the linear kernel function in text classification is comparable to the performance of nonlinear alternatives.

The classification function in (**11**), where y_i are the class labels of the entry points, has a dual representation as follows:

$$\sum_i \alpha_i \gamma_i \langle x_i, x \rangle + b = 0 \qquad \textbf{(12)}$$

Using a kernel function k, the dual classification function above can be written in high-dimensional space H as follows:

$$\sum_i \alpha_i \gamma_i k \langle x_i, x \rangle + b = 0 \qquad \textbf{(13)}$$

As mentioned above, the best classification function of SVM is the hyperplane that has the maximum margin separating classes. The problem of finding the hyperplane of the maximum margin can be formulated as a quadratic programming problem. With the dual representation of the above classification function in high-dimensional space H, the coefficients α_i of the best classification function are found in order to solve the following quadratic (dual) programming problem.

$$\max_{w} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j \, \gamma_i \gamma_j k\left(x_i, x_j\right) \tag{14}$$

$$subject\ to\ 0 \le \alpha_i \le \frac{C}{m}\ (i = 1, \ldots, m);\ \sum_{i=1}^{m} \alpha_i \gamma_i = 0$$

The parameter C in the above formulation is called the cost parameter of the classification problem. The cost parameter represents the penalty value used in SVMs to classify an input data point. A high value of C will result in a complex classification function with minimal misclassification of input data, while a low value of C will produce a simpler classification function. Therefore, setting an appropriate value for C is critical to the performance of SVMs.

The optimization problem described above is very challenging when the data set is very large because the computational complexity is equal to the square of the size of the data set. The computational and storage complexities can be reduced by dividing the training data set into a number of chunks and extracting support vectors from each of them. The support vectors can later be combined.

The same procedure can be used to incorporate new documents into the existing set of support vectors. It can be shown that the results of incorporating new documents are as good as the results of processing all documents together (Aas & Eikvil, 1999).

3.3. Random Forests

The random forests algorithm (Breiman, 2001) is an ensemble method for decision trees in which the bagging of each tree in a set of decision trees is constructed from a bootstrapped sample of feature vectors from the training data. Each bootstrap sample of feature vectors is obtained through repeated random sampling with replacement until the bootstrap sample size matches the size of the original training subset. This helps to reduce the variance of the classifier by lowering the possibility of overfitting with the training sample. When constructing each decision tree, only a subset of the randomly selected features n is considered for building each decision node, thus avoiding correlations between trees.

The random forests algorithm is used for document classification. It works as follows:

Step 1: train each tree with a random sample of "m" records, where $m < M$, and where M is the number for the entire sample. Approximately 63.2% of the training data is selected with the bootstrap method (Liu et al, 2015).

Step 2: build a decision tree with the extracted sample, which is not pruned. For the construction of the tree, $\sqrt{n}$ features are used by default from the n features available (classification problem).

Step 3: repeat steps 1 and 2. Construct a large number of decision trees and develop the sequence of decision-tree classification $\{h_1(X), h_2(X), \ldots, h_{ntree}(X)\}$.

During the modeling of random forest, the data comes from a bootstrap sample, so approximately 36.8% of the samples, which are called Out-Of-Bag (OOB), have not been extracted. The data from this sample are used as a test data set for measuring the performance of the model through its estimated OOB error rate. Breiman (2001) demonstrated that this is unbiased. Thus, the random forests model appears not to be overestimated.

Step 4: The final classification is determined for each recorded vote from the results of the decision-tree classification.

This can be expressed as follows: h_i is an individual decision tree model, Y represents the output variable (or target), and $I(\quad)$ is an indicator function.

$$H(x) = \underset{Y}{\operatorname{argmax}} \sum_{i=1}^{n} I(h_i(x) = Y) \qquad \textbf{(15)}$$

Each tree provides a classification of the remaining data (OOB), and the tree "votes" for that class. The forest chooses the classification that has the most votes over all the trees in the forest. This is the random forest's score, and the percentage of votes received by a class is the predicted probability. In a model with a binary response, if 200 of 500 trees are OOB, and 160 vote for class 1 and 40 vote for class 2, then the random forest model classifies it as class 1 with a probability of 0.80 (160/200).

From the n features randomly selected to construct each of the decision nodes, the condition is selected with respect to the class c that best reduces the Gini impurity metric g of the data, which indicates how often an element randomly selected from the set would be incorrectly labeled if it were randomly labeled according to the distribution of labels in the subset. The Gini impurity reaches its minimum (zero) when all cases of a node belong to a single target category. The higher its value, the more uncertain the classifier is about whether a feature vector belongs to one class or another. The calculation of Gini impurity for a tree node is obtained by the following expression:

$$\mathrm{g} = 1 - \sum_{c=1}^{m} P_c^2 \tag{16}$$

Where P_c is the probability that a document will be labeled as class c.

4. Performance metrics

A very important problem in text classification is how to evaluate the performance of classifiers (methods or models). Many measures have been used, each of which has been designed to evaluate some aspect of a system's classification performance. This section describes some of the metrics documented in the literature.

A common approach to the problem of classification with multiple classes is to separate each problem into a binary classification. For each class and each document, it is determined whether the document belongs to the class of interest (positive class) or not (negative class). When evaluating the performance of a classifier, four numbers derived from the confusion matrix are of interest to each class (Figure 1).

Figure 1. Confusion Matrix for a two-class problem

	Positive prediction	Negative prediction
Positive observed	TP	FN
Negative observed	FP	TN

Source: Own elaboration

Where,

TP: Number of documents correctly predicted for the class of interest.
FP: Number of documents incorrectly predicted for the class of interest.
FN: Number of documents incorrectly rejected for the class of interest.
TN: Number of documents correctly rejected for the class of interest.

From these quantities, the following performance metrics are determined:

- Recall: measures the precision of the classifier (model) for cases of the class of interest.

$$recall = \frac{\mathrm{TP}}{TP + FN} \tag{17}$$

- Precision: measures the precision of the classifier (model) for the predicted cases of the class of interest.

$$precision = \frac{TP}{TP + FP} \tag{18}$$

- Accuracy: measures the overall precision of the predictions made by the classifier (model).

$$accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{19}$$

Micro- and macro-averaging: there are two conventional methods of evaluating average performance across classes, namely macro-average and micro-average (Aas & Eikvil, 1999). The score of the macro-average is determined by calculating the performance metrics by class and then averaging these to calculate the overall averages. The score of the micro-average is determined by first calculating the totals of $TP, FP, FN,$ and TN for all classes and then using these totals to calculate the performance metrics. An important distinction between the two types of averages is that the micro-average gives equal weight to each *document*, while the macro-average gives equal weight to each *class*.

Break-even point: The above performance metrics can be misleading when examined on their own. A classifier usually exhibits a trade-off between recall and precision, where obtaining a high recall typically means sacrificing precision, and vice versa. If recall and precision are configured to have the same value, then this value is called the break-even point of the system. The break-even point has been commonly used in text classification evaluations (Aas & Eikvil, 1999).

F-measure: another evaluation criterion that combines recall and precision is the F-measure.

$$F_\beta = \frac{(1+\beta^2) \cdot \ recall \ \cdot precision}{\beta^2 \cdot precision + recall} \tag{20}$$

Where β is a parameter that permits different weightings of recall and precision. For this study, it has been assumed that $\beta = 1$. That is, the F1 score is the harmonic average between recall and precision.

5. Application to the definitions used in the ENE

As mentioned above, this document applies text-mining methods described in the previous sections to the definitions used in the National Employment Survey (ENE). INE must classify many texts every month, and thus the purpose of this study is to create an alternative to their manual classification.

5.1. The data set

The data set is the 183,784 documents from the 2017 database of the ENE. The definitions were derived from the responses of informants regarding their "profession, job, or occupation" and the "economic sector" of the company that employs them, which are taken from questions B1 and B14b of the questionnaire.

The target variables for training the classifiers are the variables of economic activity from the CIUO (the Spanish version of the International Standard Classification of Occupations (ISCO)) and CAENES (Classification of National Economic Activity for Socio-demographic Surveys) at one and two digits. CIUO classification (DANE, 2005) is one of the main classifications for which the International Labour Organization (ILO) is responsible. CIUO organizes jobs into a series of clearly defined groups according to characteristic tasks. CAENES (INE, 2016) is based on the Chilean Classification of Economic Activities CIIU4.CL 2012, whose structure facilitates the classification of economic activities. CAENES is used because its most disaggregated category (class) can include several subclasses, classes, or groups of CIIU4.CL 2012 and other categories can be added to it as needed, allowing the level of detail required for household surveys.

In short, four target variables are used in the training of classifiers, namely CIUO-1, CIUO-2, CAENES-1 and CAENES-2. These variables account for the economic-activity classifiers CIUO at one and two digits, and CAENES at one and two digits, respectively. CIUO-1 consists of 10 classes, CIUO-2 consists of 27 classes, CAENES-1 consists of 21 classes, and CAENES-2 consists of 83 classes.

It should be noted that the documents used in the classification process of CIUO-1 and CIUO-2 are the result of the concatenation of the "occupation, description of tasks" and the "economic sector" to which the company belongs, as answered by the informants. In CAENES-1 and CAENES-2, the texts are associated with the "economic sector" to which the company belongs, as declared by the informant.

In the data used for CAENES-1 and CAENES-2 classifications, the presence of null and missing values in the documents was detected. Therefore, they were removed, leaving 177,176 documents in the data set. To construct the classifiers, the input data sets were divided into a training set of 80% of the cases, and a test set of 20%. Table 1 summarizes the composition of the training and test sets for the construction of each classifier.

Table 1. Training and test data sets for CIUO and CAENES

Classifier	No. training sets	No. test sets
CIUO-1	147,027	36,757
CIUO-2	147,027	36,757
CAENES-1	141,740	35,436
CAENES-2	141,740	35,436

Source: Own elaboration

5.2. Data preparation

The bodies of all documents were converted from the original format (i.e., character strings) to word vectors. Below, the steps of this procedure are described.

1. Individual words were extracted through standardization and tokenization (see section 2.1). Then, using a list of 344 high-frequency words in Spanish (e.g., *ante*, *de*, *cual*, *entonces*), the stopwords were removed. Subsequently, a stemming process was completed with the R hunspell library (Ooms, 2017). This procedure resulted in 37,685 unique words.

2. The Information Gain (IG) was used as a metric for the selection of features in order to distinguish words that provide more information for classification. The results of this metric can be seen in **annex 1**. In addition, words appearing below a specified frequency (in this case 5) were removed from all documents in the collection (see section 2.2.). Thus, 8,448 words remained in the CIUO classification. The reduction in dimensionality resulted in a document-word matrix of 147.027 *x* 8.448 for the training data set. For CAENES classification, 5,346 words remained, resulting in a document-word matrix of 141,740 *x* 5,346 for the training data set.

3. The weight measurement "*tf - idf*" was used for indexing words in the document-word matrix (see section 2.3.).

5.3. Use of machine-learning methods

This study sought to apply and evaluate three machine-learning techniques for the classification of documents (definitions) in some of the groups of economic activity defined by CIUO (one and two digits) and CAENES (one and two digits). Texts were classified in supervised mode, that is, classification included training with examples of previously labeled classes (groups of activity).

The parameters for the techniques used in the comparative evaluation were established from the test results in the data set. No additional parameter adjustments were made. Although tuning parameters to specific data sets may be beneficial, the use of generally accepted configurations is more typical in practice. The need for significant time and effort for fine-tuning the parameters can often be an impediment to their practical application, and it can lead to specific data overfitting problems.

For naïve Bayes (NB), the *a priori* classes and probability distributions of features were estimated from the frequencies of the training data set.

For support vector machine (SVM), a linear kernel was used. The optimization of the cost C parameter through a 10-fold cross validation resulted in an optimal value of 0.75 for CIUO and CAENES to one digit and an optimal value of 1.0 for CIUO and CAENES to two digits.

For random forest (RF), the number of features was set at $\sqrt{n}$, and the number of trees was set at $T = 1000$.

These three methods were implemented with the statistical software R (R Core Team, 2018).

5.4. Results

The results obtained from the application of the previously described methods to ENE definitions are shown below. Precision, recall, and F1 score were used to evaluate the performance of the different methods (see section 4).

5.4.1. CIUO-1

Table 2 displays in descending order the numbers of documents in the collection used for training and testing the models for each of the classes of CIUO-1. The frequency of class 9 (elementary occupations) stands out.

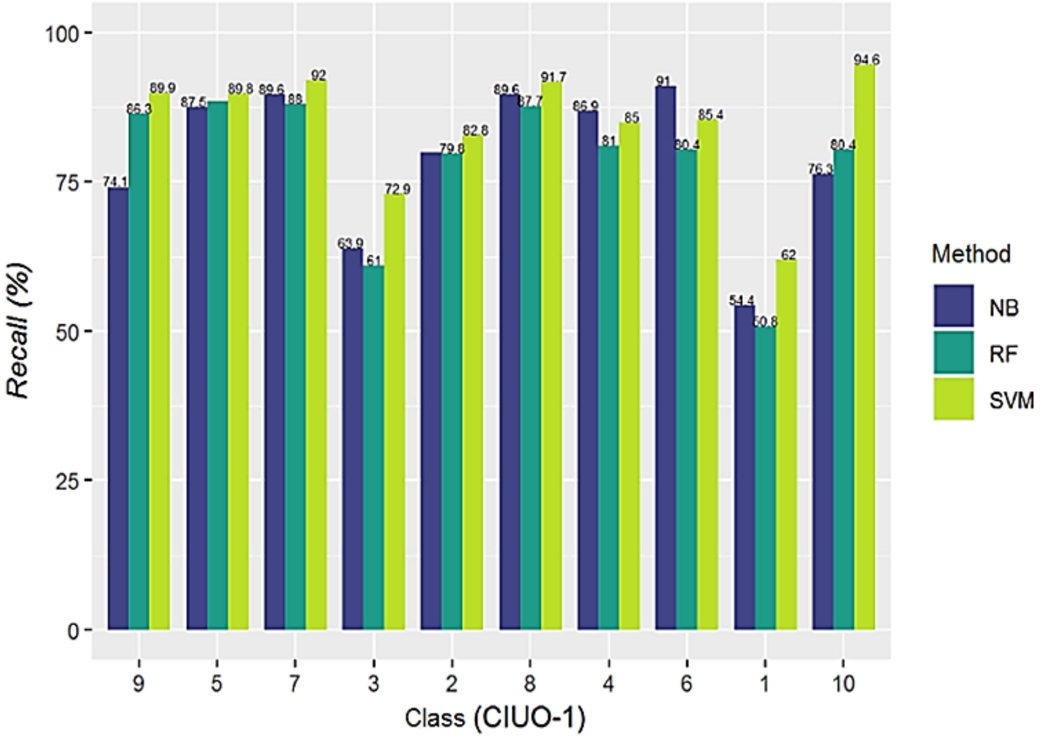
Table 2. Training and test data sets for CIUO-1

Class	No. training sets	No. test sets
9. Unskilled workers	35,469	8,788
5. Service and sales workers	22,337	5,606
Officers, operators, artisans, and workers in the manufacturing, construction, and mining industries	20,378	4,984
3. Technicians with non-university post-secondary training and assistants	16,309	4,005
2. Science professionals and intellectuals	15,900	4,050
8. Plant and machine operators and assemblers	12,942	3,268
4. Clerical support workers	12,667	3,228
6. Agricultural, forestry, and fishery workers	7,223	1,816
1. Members of the executive branch, legislative bodies, and management staff of public administration and of companies	2,616	688
10. Armed forces	1,186	324

Source: Own elaboration

Chart 1 shows that all three methods have a relatively stable performance across classes. SVM performed best in all classes, especially in classes 9, 5, and 7, which have the largest number of documents in the collection. The recall of SVM for these classes was approximately 90%. In contrast, classes 3 and 1 performed poorly for all methods, its level of recall below 70%.

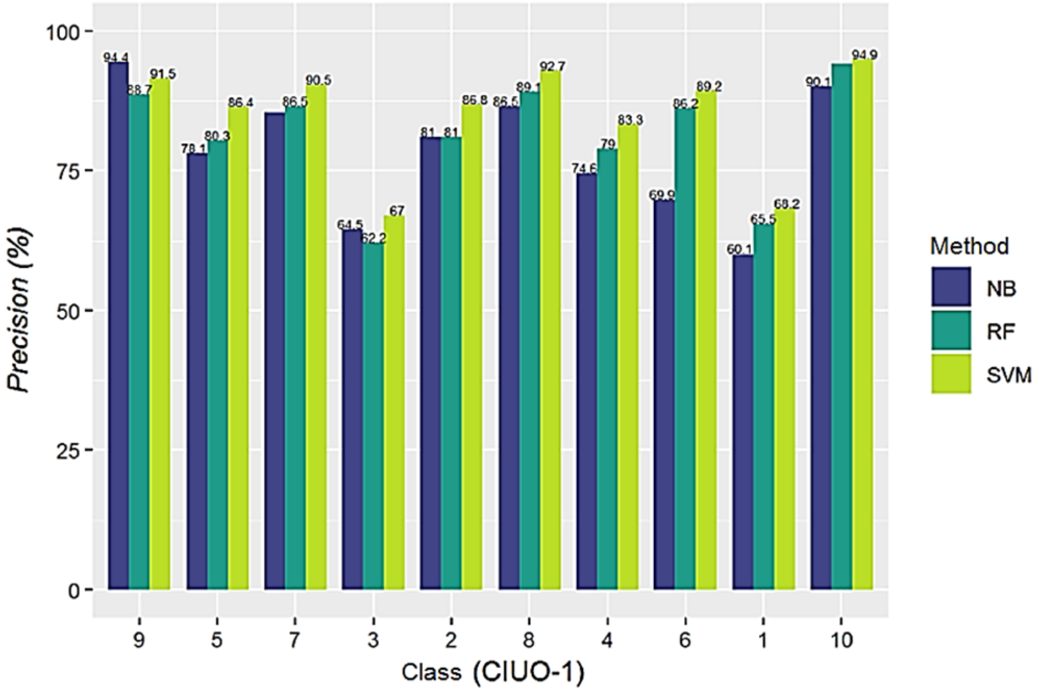
Chart 1. Recall performance, according to method for CIUO-1



Source: Own elaboration

As can be seen chart 2, NB works very well in predicting class 9 (the most populated class in the data set), achieving a precision of 94%. All three methods show a relatively stable performance of over 80%, mainly in the classes with the largest number of documents in the collection (i.e., 9, 5, and 7). As in recall, classes 3 and 1 performed poorly, their precision approximately 68%.

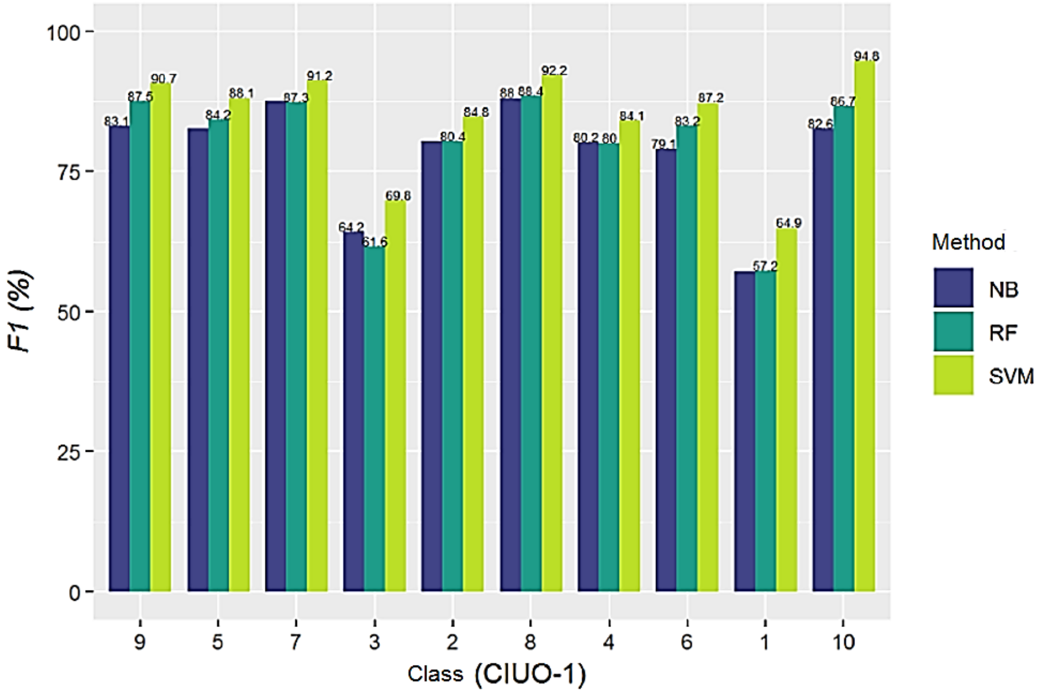
Chart 2. Precision performance, according to method for CIUO-1



Source: Own elaboration

In chart 3, the F1 score shows results similar to the recall method, but SVM performed best among the three methods.

Chart 3. F1-score performance, according to method for CIUO-1



Source: Own elaboration

5.4.2. CIUO-2

Table 3 displays in descending order the numbers of documents in the collection used for training and testing the models for each of the classes of CIUO-2. The most notable frequencies were in classes 91 and 52, which refer to sales and service activities.

Table 3. Training and test data sets for CIUO-2

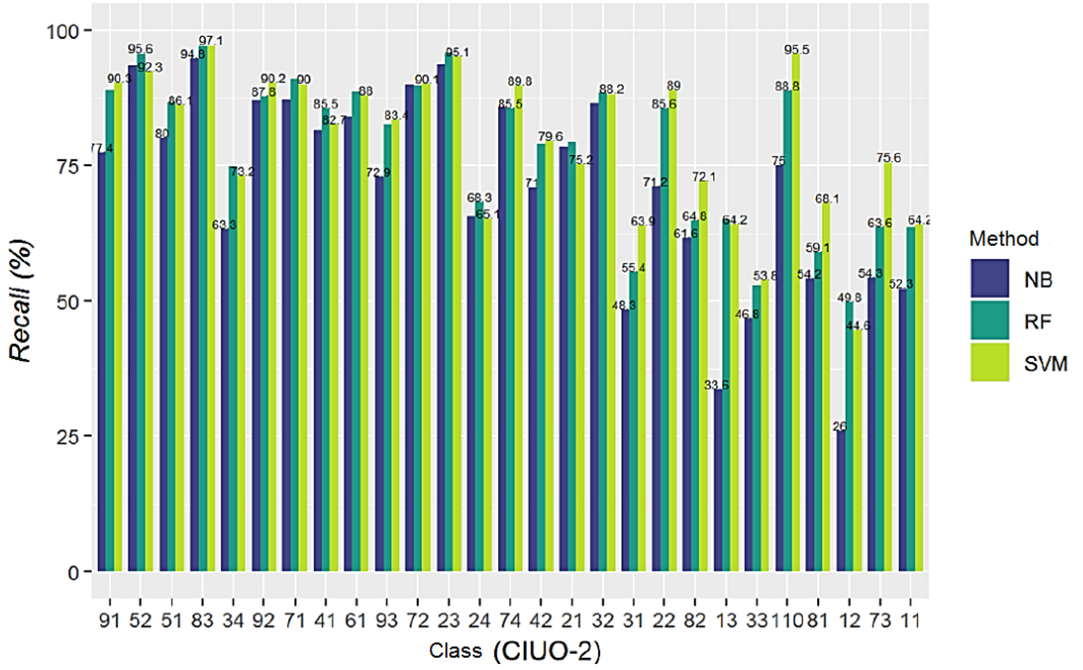
Class	No. training sets	No. test sets
91. Unskilled service workers (except domestic service workers and similar)	19,503	4,794
52. Protection and security services personnel	12,469	3,156
51. Personal service workers	9,868	2,450
83. Vehicle drivers and mobile heavy equipment operators	9,840	2,549

Class	No. training sets	No. test sets
34. Other associate professionals and assistants	9,756	2,354
92. Domestic service workers, cleaners, launderers, ironers, and the like	9,128	2,285
71. Officers and workers in extractive industries	8,544	2,077
41. General and keyboard clerks	8,501	2,140
61. Farmers and forestry, livestock, and fisheries workers	7,223	1,816
93. Mining, construction, manufacturing, and transport workers	6,838	1,709
72. Construction officers and workers	6,024	1,442
23. Education Professionals	5,588	1,378
24. Other scientific and intellectual professionals	5,398	1,393
74. Machine and equipment mechanics and installers	4,992	1,267
42. Customer service workers	4,166	1,088
21. Professionals in the physical, chemical, mathematical, and engineering sciences	2,899	793
32. Associate professionals in the biological, medical, and health sciences	2,683	658
31. Associate professionals in the physical, chemical, engineering, and related sciences	2,231	583
22. Professionals in the biological sciences, medicine, and health	2,015	486
82. Machine operators and assemblers	1,978	462
13. Public and private production and specialized services managers	1,657	446
33. Teaching assistants and instructors in formal, special, and vocational education	1,639	410
110. Officers of the armed forces	1,186	324
81. Stationary Plant and Machine Operators	1,124	257
12. General managers of private companies	837	213
73. Metalworkers and similar	818	198
11. Chief executives, senior officials, and legislators	122	29

Source: Own elaboration

In chart 4, all three methods are relatively stable across classes. SVM performed best in almost all classes, achieving a recall of about 90% in classes 91 and 52, which contain the most documents in the collection. RF performed similarly to SVM, particularly in the classes with the highest number of documents. Finally, NB is at a distance from SVM and RF in almost all classes. In addition, the results are consistent with those observed in CIUO-1. The results of classes such as 11, 12, 13, 31, 32, and 33 of CIUO-2 are similar to the results of classes 1 and 3 of CIUO-1, whose recall was approximately 60%.

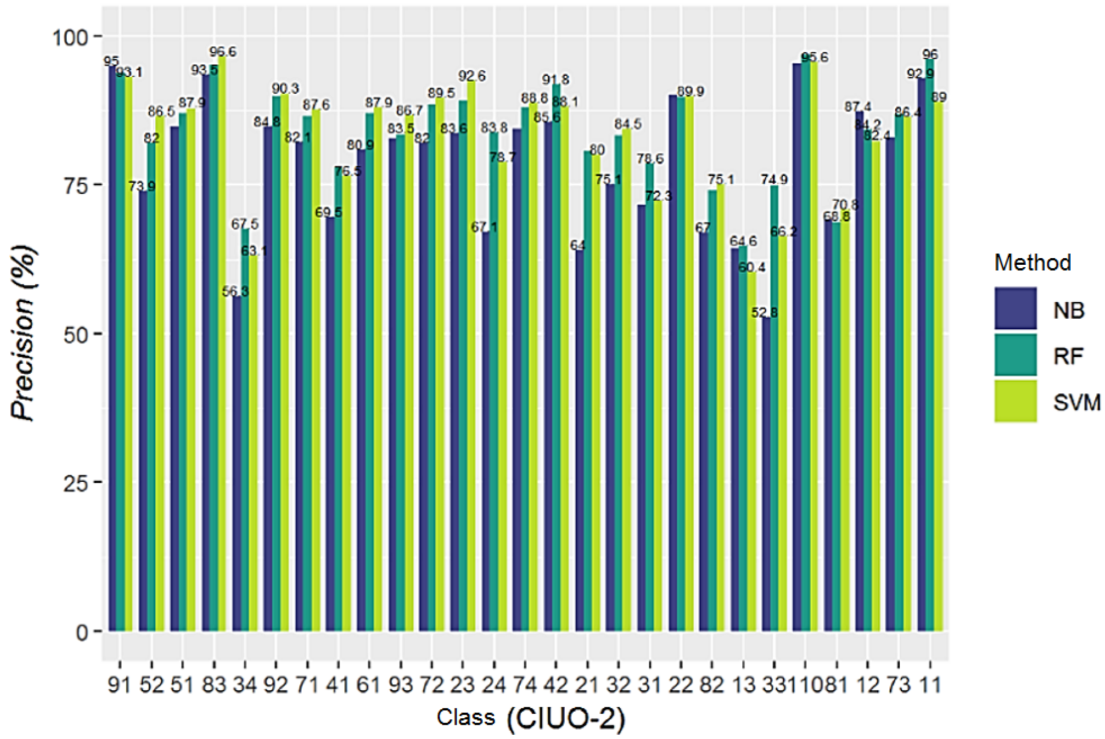
Chart 4. Recall performance, according to method for CIUO-2



Source: Own elaboration

In chart 5, all three methods show relatively stable behavior across classes. For class 91, NB exhibits a remarkable precision of 95%. SVM and RF stand out for their parity in high performance across classes, mainly in those with a higher number of documents, achieving a precision of approximately 85%.

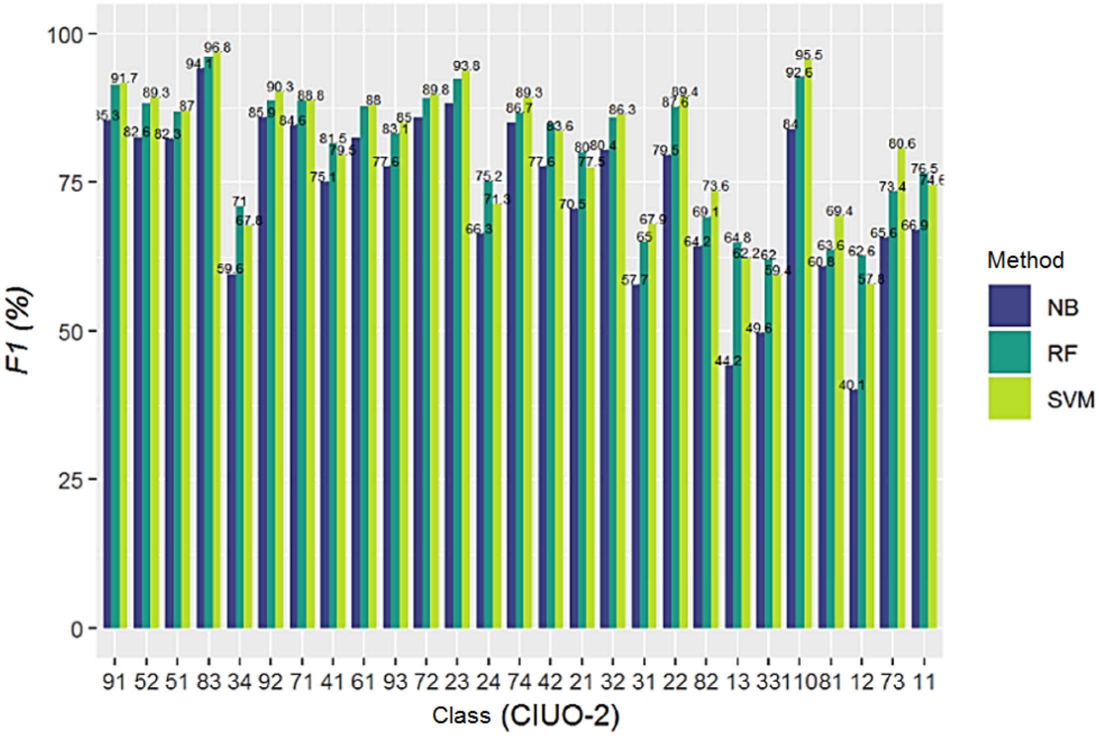
Chart 5. Precision performance, according to method for CIUO-2



Source: Own elaboration

In chart 6, F_1 shows a similar pattern to the results of recall, where SVM performed best in almost all classes.

Chart 6. F1-score performance, according to method for CIUO-2



Source: Own elaboration

5.4.3. CAENES-1

Table 4 displays in descending order the numbers of documents in the collection used for training and testing the models for each of the classes of CAENES-1.

Table 4. Training and test data sets for CAENES-1

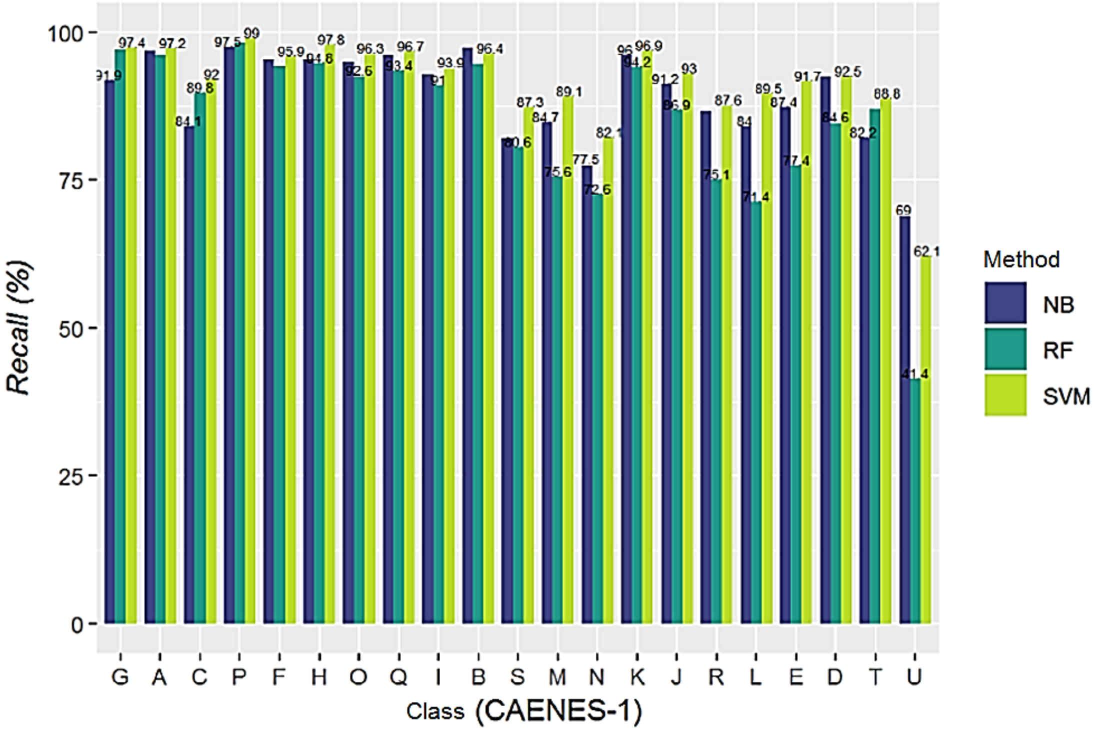
Class	No. training sets	No. test sets
G. Wholesale and retail trade; repair of motor vehicles and motorcycles	26,718	6,556
A. Agriculture, forestry, and fishing	17,569	4,439
C. Manufacturing	14,688	3.688
P. Education	13,027	3,201
F. Construction	11,585	2,840

Class	No. training sets	No. test sets
H. Transportation and storage	9,179	2,388
O. Public administration and defense; compulsory social security	9,064	2,343
Q. Human health and social work activities	8,030	2,048
I. Accommodation and food service activities	6,398	1,575
B. Mining and quarrying	4,707	1,193
S. Other service activities	4,258	997
M. Professional, scientific, and technical activities	3,665	885
N. Administrative and support activities	3,236	863
K. Financial and insurance activities	2,294	575
J. Information and communication	1,960	512
R. Arts, entertainment, and recreation	1,647	420
L. Real estate activities	1,127	288
E. Water supply; sewerage, waste management, and remediation activities	975	230
D. Electricity, gas, steam, and air conditioning supply	806	197
T. Activities of households as employers; undifferentiated goods- and services-producing activities of households for own use	769	187
U. Activities of extraterritorial organizations and bodies	24	5

Source: Own elaboration

In Chart 7, all three methods show very stable performances, achieving a recall over 90% in almost all classes, a very satisfactory result. In contrast, the recall of class U (activities of extraterritorial organizations and bodies) was below 70%. However, because of its low number of training (and test) documents, this result should be taken only as a descriptive reference. Overall, SVM shows a slight performance advantage over NB and RF.

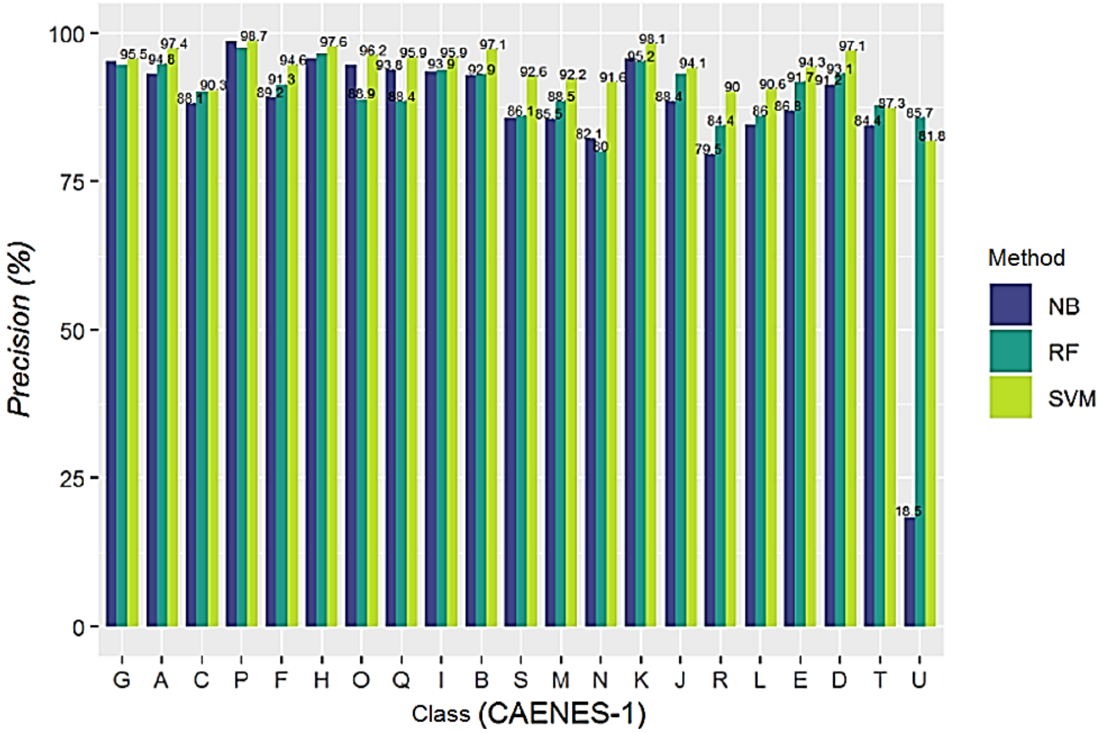
Chart 7. Recall performance according to method for CAENES-1



Source: Own elaboration

In chart 8, all three methods show a high level of precision. SVM performs best, reaching over 90% in almost all classes. As in recall, the performances observed in class U should be considered as descriptive only.

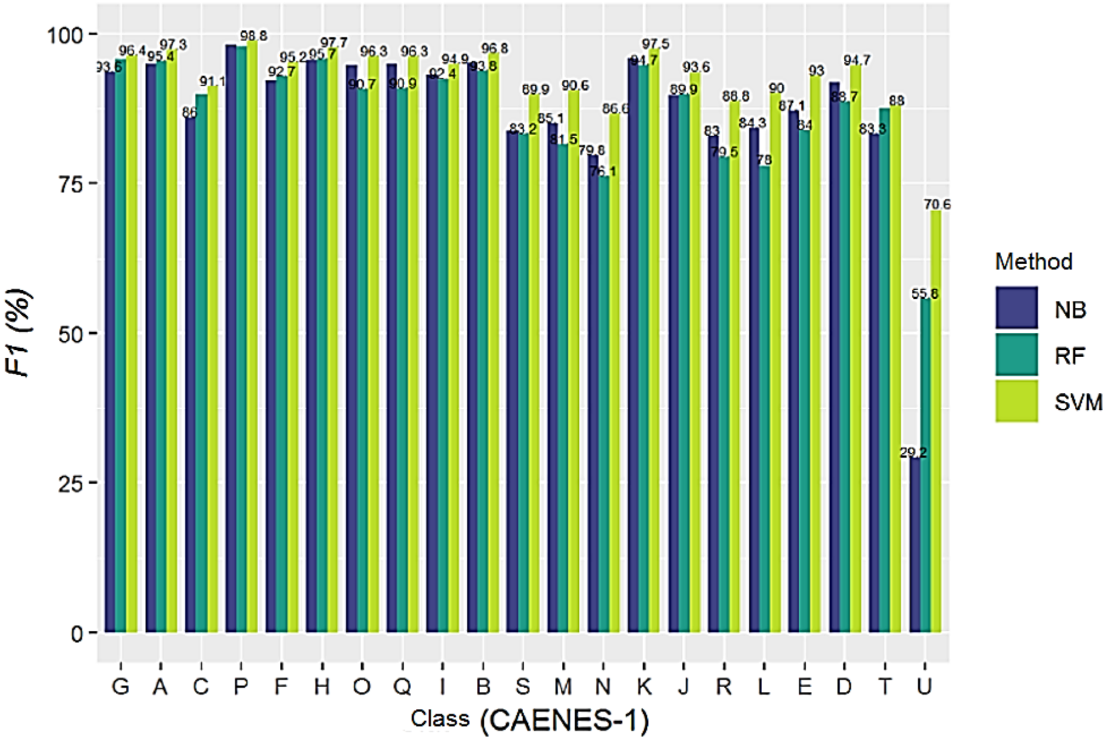
Chart 8. Precision performance, according to method for CAENES-1



Source: Own elaboration

In chart 9, the F1 score shows a pattern similar to the results observed in precision. SVM performed best in all classes, achieving a performance above 90% in almost all classes.

Chart 9. F1-score performance, according to method for CAENES-1



Source: Own elaboration

5.4.4. CAENES-2

Table 5 shows in descending order the numbers of documents in the collection used for training and testing the models for each of the classes that make up the top 27 of the number of documents for CAENES-2. These classes represent approximately 86% of the documents in the collection. Table 5 also shows a notable frequency of documents belonging to Class 48 (wholesale and retail trade, except for motor vehicles and bicycles). Charts 10, 11, and 12 show the performances for these classes, according to recall, precision, and F1 score, respectively. For the performance of the remaining classes, see **annex 2**.

Table 5. Training and test data sets for the Top 27 of CAENES-2

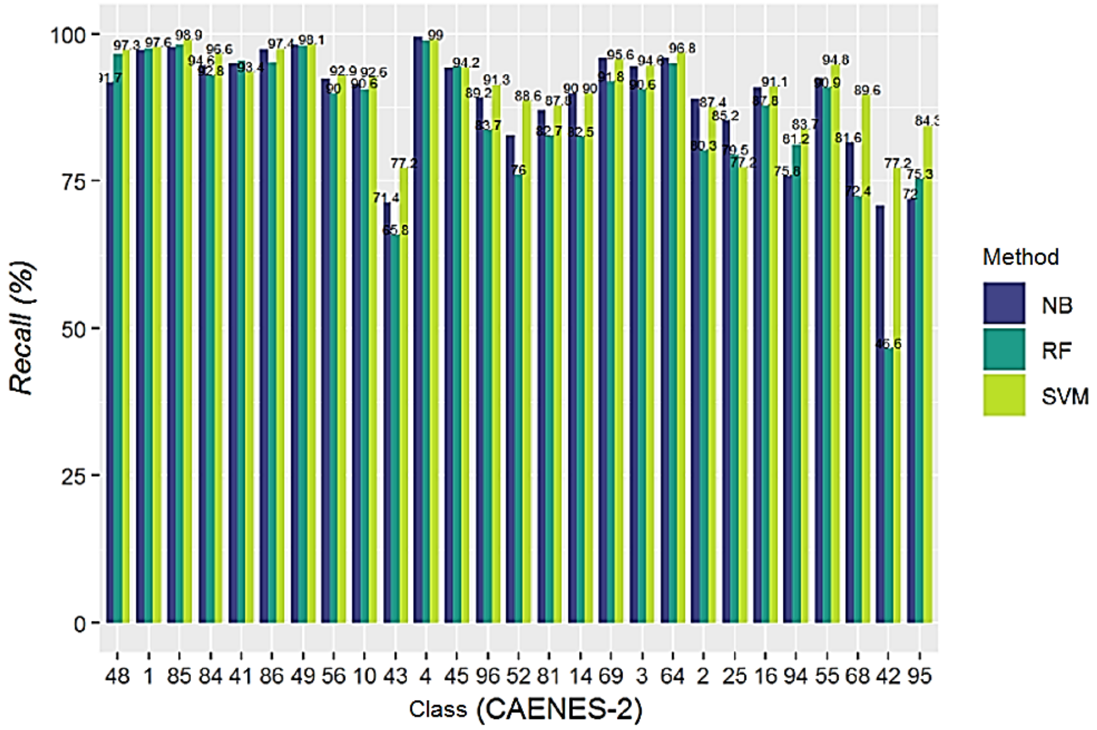
Class	No. training sets	No. test sets
48. Wholesale and retail trade; repair of motor vehicles and motorcycles	23,908	5,848
1. Crop and animal production, hunting, and related service activities	14,653	3,684
85. Education	13,027	3,201
84. Public administration and defense; compulsory social security	9,064	2,343
41. Construction of buildings	6,775	1,727
86. Human health activities	6,657	1,715
49. Land transport and transport via pipelines	6,652	1,703
56. Food and beverage service activities	5,221	1,280
10. Manufacture of food products	5,071	1,202
43. Specialized construction activities	3,796	874
4. Mining and processing of copper	3,743	977
45. Wholesale and retail trade and repair of motor vehicles and motorcycles	2,810	708
96. Other personal service activities	2,077	468
52. Warehousing and support activities for transportation	1,740	466
81. Services to buildings and landscape activities	1,663	411
14. Manufacture of wearing apparel	1,624	426
69. Legal and accounting activities	1,610	384
3. Fishing and aquaculture	1,528	399
64. Financial service activities, except insurance and pension funding	1,486	373
2. Forestry, logging, and related activities	1,388	356
25. Manufacture of fabricated metal products, except machinery and equipment, and metal working services	1,301	316
16. Manufacture of wood and of products of wood and cork, except furniture; manufacture of articles of straw and plaiting materials	1,256	342
94. Other personal service activities	1,216	288

Class	No. training sets	No. test sets
55. Food and beverage service activities	1,177	295
68. Real estate activities	1,127	288
42. Wholesale and retail trade and repair of motor vehicles and motorcycles	1,014	239
95. Repair of computers and personal and household goods	965	241

Source: Own elaboration

In chart 10, all three methods show high performance in almost all classes, achieving a recall of approximately 90%, a quite satisfactory result. This is observed primarily in the classes with the largest number of training (and test) documents.

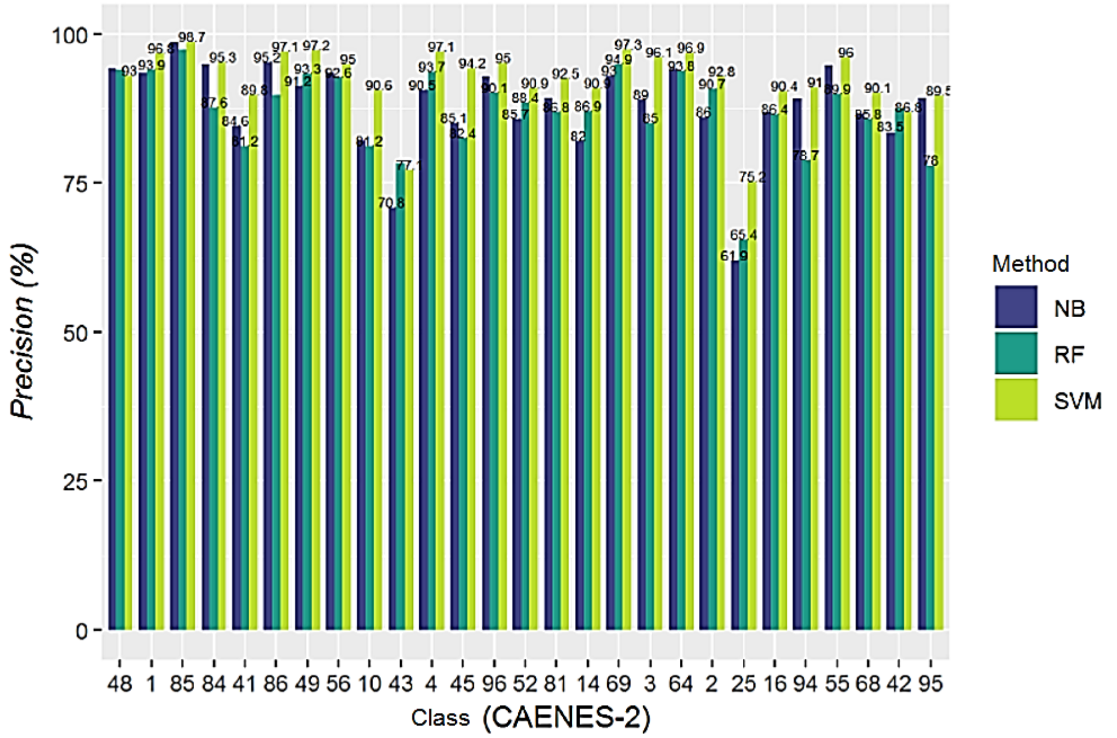
Chart 10. Recall performance, according to method for CAENES-2. Classes are of the first twenty-seven positions, according to the number of documents in the collection.



Source: Own elaboration

In chart 11, all three methods exhibit relatively stable performances, achieving a precision of approximately 90%. SVM performed best in almost all classes.

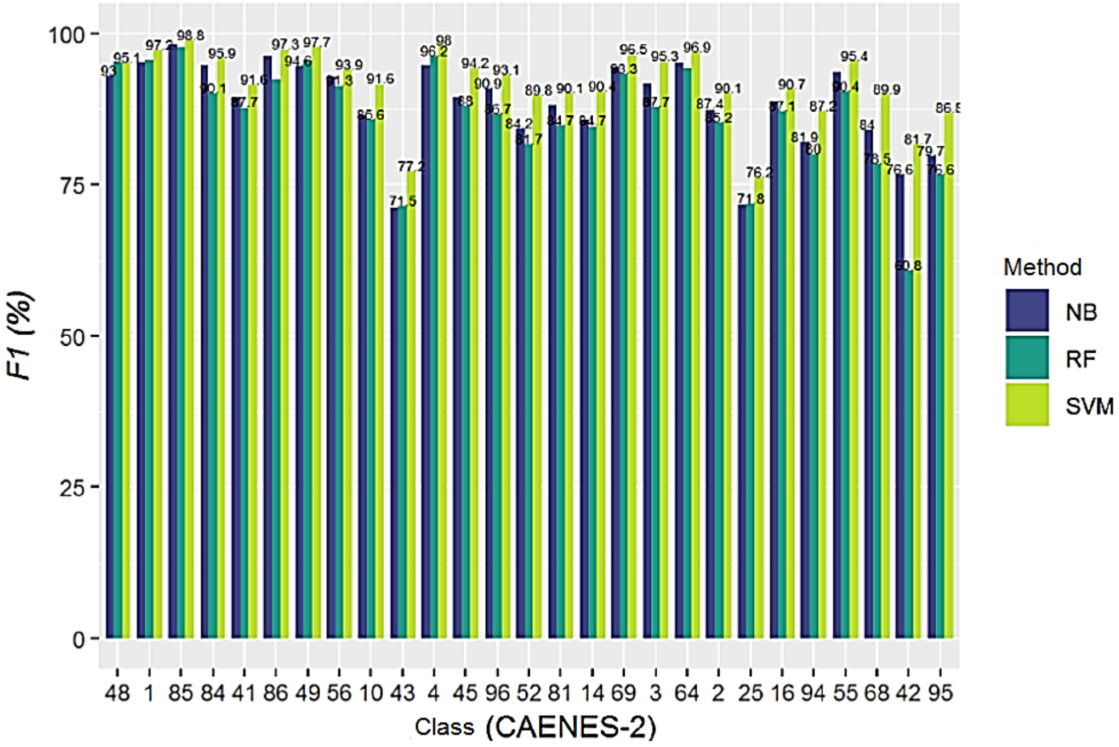
Chart 11. Precision performance, according to method for CAENES-2. Classes are of the first twenty-seven positions, according to the number of documents in the collection.



Source: Own elaboration

In chart 12, the F1 score shows a similar pattern to the results of precision. SVM performed best among the three methods.

Chart 12. F1-score performance, according to method for CAENES-2. Classes are of the first twenty-seven positions, according to the number of documents in the collection.

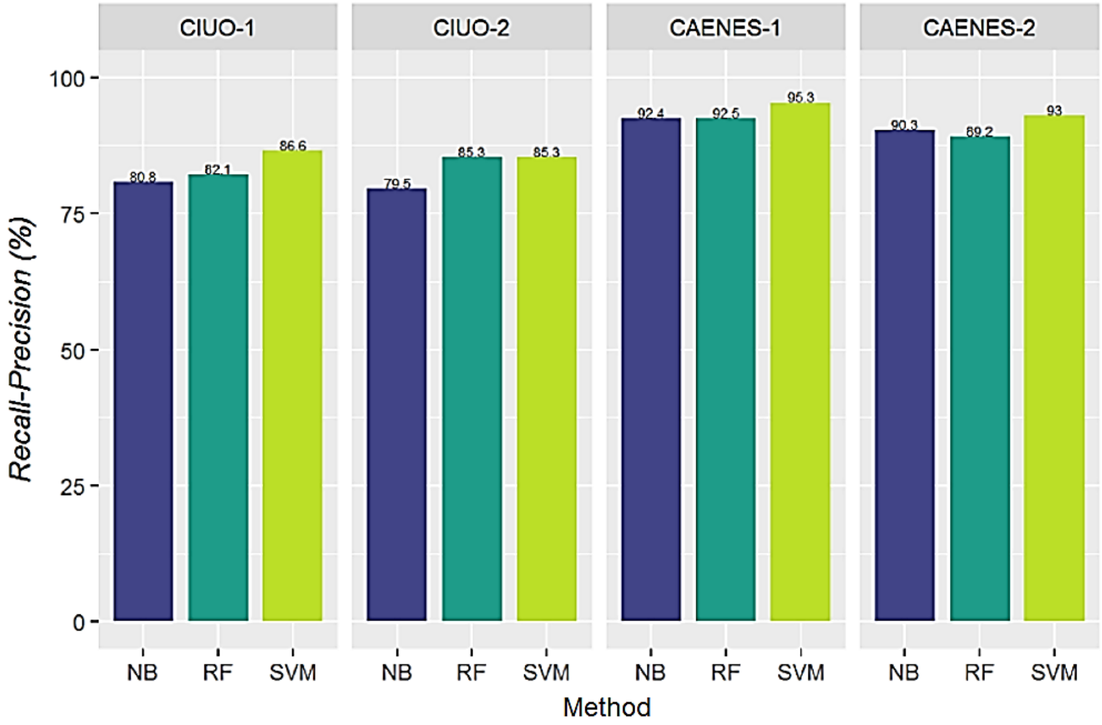


Source: Own elaboration

5.4.5. Overall performance

To evaluate the overall effectiveness of the methods, we followed other studies reported in the literature by using a recall/precision break-even point that accounts for the trade-off between recall and precision. Average class performance was calculated using micro-averages, a measure that gives equal weight to each document in the collection (see section 4). Chart 13 shows the recall/precision obtained for both the CIUO and CAENES classifications, including all classes in each case. It should be noted that all three methods are very competitive and work well in the task of classification, achieving a recall/precision of over 80% for CIUO and over 90% for CAENES, which is quite satisfactory. SVM achieved the highest performances for recall/precision in CIUO-1 (86.6%), CAENES-1 (95.3%), and CAENES-2 (93.0%). For CIUO-2, SVM and RF preformed best, both achieving a recall/precision of 85.3%.

Chart 13. Recall-Precision performance, according to method for CIUO and CAENES classifications



Source: Own elaboration

5.5. Discussion and comparison of methods

Compared to the classification of texts in general, the classification of definitions in the ENE is a special task, and it is quite challenging. The target variables have a range of possible classes: CIUO-1 has 10 classes, CIUO-2 has 27, CAENES-1 has 21, and CAENES-2 has 83 classes. Although some classes appear similar, important differences in the features of classes of economic activity should not be overlooked. In CIUO-1, for example, the vocabulary is far more extensive in class 9 than in class 10. Ideally, a successful machine-learning algorithm used in this particular classification domain should make full use of the possible differences among classes of economic activity. More importantly, it should profile classes precisely with only a small number of false positives. In order to characterize economic activities in a way that distinguishes related classes, it is important to have correctly standardized texts and, when collecting information, to capture specific words. For example, CIUO-1 performed rather poorly in classes 1 and 3 because these classes include technical and professional activities of the informant that are captured in class 2.

As in many other machine-learning applications, declaring an algorithm as the best for the classification of definitions is a very difficult task, perhaps an impossible one. The experiments and analyses conducted in this study, however, have revealed some interesting characteristics of the three methods investigated. They are summarized below:

1. Naïve Bayes (NB) is simple, and it is the fastest in model learning among the three methods. It works well for the classification of texts. Because the algorithm assumes that the individual features are completely independent of each other, the model can benefit from an effective selection of features, which has been demonstrated in the experiments. In the same vein, NB may perform poorly if applied to a data set with some observable dependencies among features. In the experiments, NB performed remarkably well in its precision over the most populated class of each of the classifications. A possible explanation is that the probabilities were established *a priori* through the frequencies observed in the documents.

2. Support vector machine (SVM), as reported in many previous studies, is a very stable classifier, and it can be scaled to the dimensions of features. In this study, SVM was the best classifier, as reflected through the measurements of recall, precision, F1 score, and recall/precision. The recall/precision break-even point of SVM was 86.6% for CIUO-1, 85.3% for CIUO-2, 95.3% for CAENES-1, and 93.0% for CAENES-2. The kernels function and cost parameter *C* selected for SVM dramatically influence the outcome. In this study, a linear kernel function was chosen, which turned out to be relatively fast in model training (approximately 2 hours in processing). The cost parameter was determined through 10-fold cross validation. An important property of SVM is the way in which it reaches the best classification function by establishing the maximum margin of separation between two classes. This gives SVM a powerful capacity for generalization of classification.

3. Random forest (RF) works best in the CIUO-2 classification and is very competitive, its performances similar to those of SVM. Although hardware

requirements for calculations make the training of the models relatively slow (approximately 14 hours in processing), the application of bagging, which is a feature of RF, gives it a powerful capacity for the generalization of classification. Because the number of features to be used in RF is crucial, the number chosen for this study is the same as the number recommended in the literature and as the number set by default in the R implementation (i.e., the square root of the number of features). This parameter can be calibrated in the future for better results.

The three methods work very well in the task of document classification, most notably in their performances using CAENES, where they reach 90% in all the observed metrics. One possible explanation for this remarkable result is that the definition used in training (and testing) the models is derived exclusively from the description of the "economic sector" of the company where the informant works. This provides shorter texts with specific words that characterize an economic activity. The methods have reasonable processing times, which can however be decreased by improving available hardware and software.

It is clear that the three algorithms are viable alternatives for obtaining precise classifications and obtaining them in much less time than is possible with manual classification. This reduces operational costs and increases efficiency.

SVM performed best in statistical terms, although RF is a good alternative to CIUO-2 classification because it is similar in capacity to SVM. However, in computational terms, SVM offers a shorter processing time. Consequently, based on the statistical and computational criteria analyzed, the use of the SVM model is recommended for the classification of the definitions of the National Employment Survey (ENE).

6. Conclusions and projections

Typical text classification consists of the following steps: preprocessing, reduction of dimensionality, proper representation of documents, and classification. In this study, different methods have been described for all these steps, including alternatives that can be tested in greater depth in new studies. In addition to the theoretical introduction of these methods, this study evaluated the application of NB, SVM, and RF methods to the classification of ENE definitions. These algorithms were evaluated using the ENE 2017 data set. The best results were obtained with the SVM method, which was also easy to apply. In the future, these methods can be improved through a finer calibration of model parameters and through a higher quality of the input data, which would generate even better results in terms of precision.

This document has responded to the need for automated systems to classify the large volumes of information handled by INE, most of which is classified manually. Thus, the text-mining and machine-learning algorithms introduced in this study have been proposed within INE as a viable alternative that improves precision and decreases costs.

With the dramatic increase in the use of the internet and other sources of information, the volume of documents that institutions need to process has exploded. INE must play a fundamental role in this task by adequately adopting the latest technologies available for the development of its public role. In the future, INE must study the application of these or other more complex algorithms (such as deep learning algorithms) to much larger volumes of data.

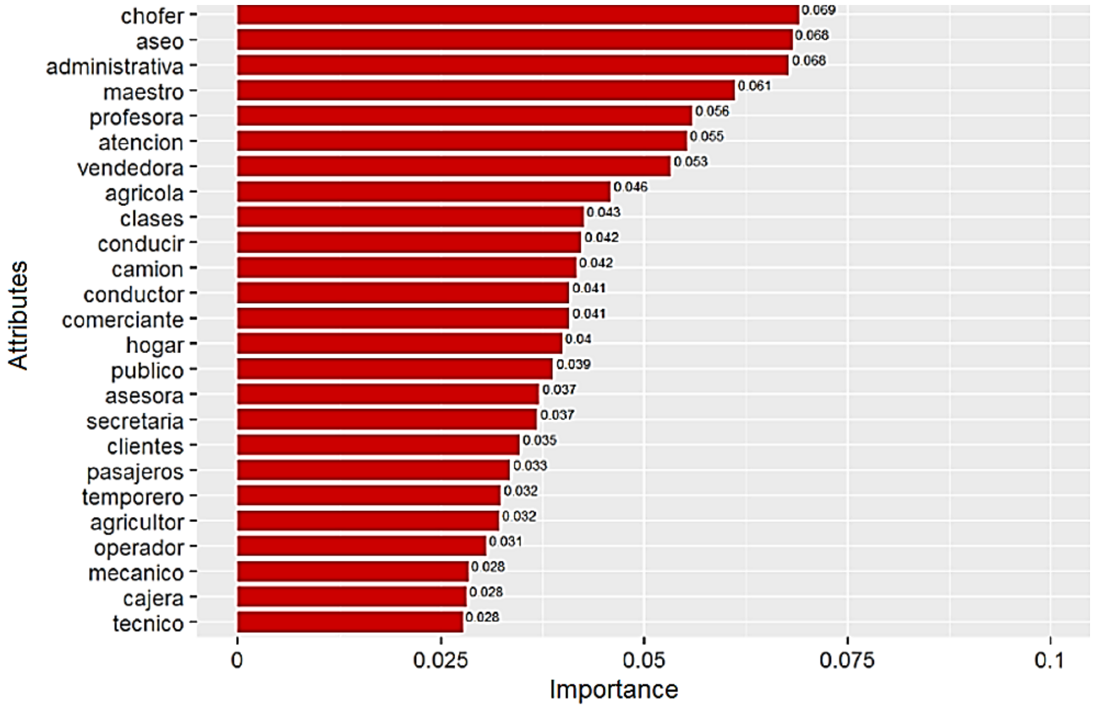
Finally, this study shows that it is possible, in the context of public service, to make improvements based on the use of free and open source software, such as the R platform. This could free up resources that are currently being spent on paid products, and it could take advantage of the opportunities for collaboration between institutions that this type of software facilitates.

Annexes

Annex 1: Information Gain

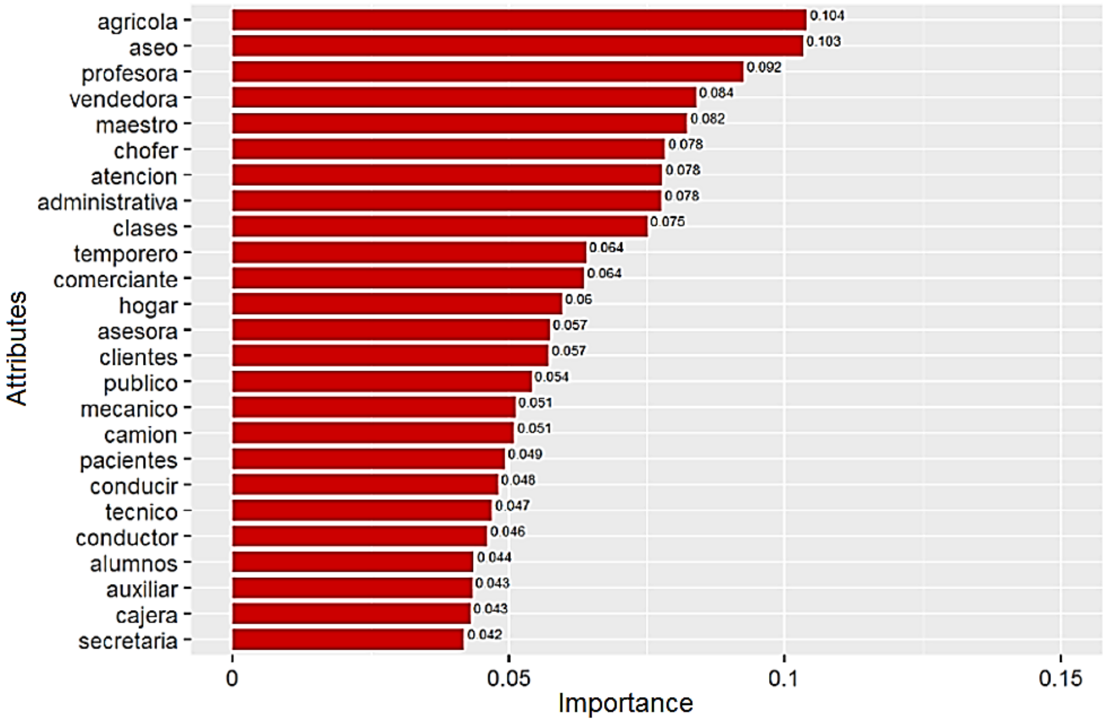
Charts A1 to A4 show the Information Gain (IG) of the top 25 attributes (features) for the classifications CIUO-1, CIUO-2, CAENES-1 and CAENES-2, respectively.

Chart A1. Information gain of top 25 attributes of CIUO-1



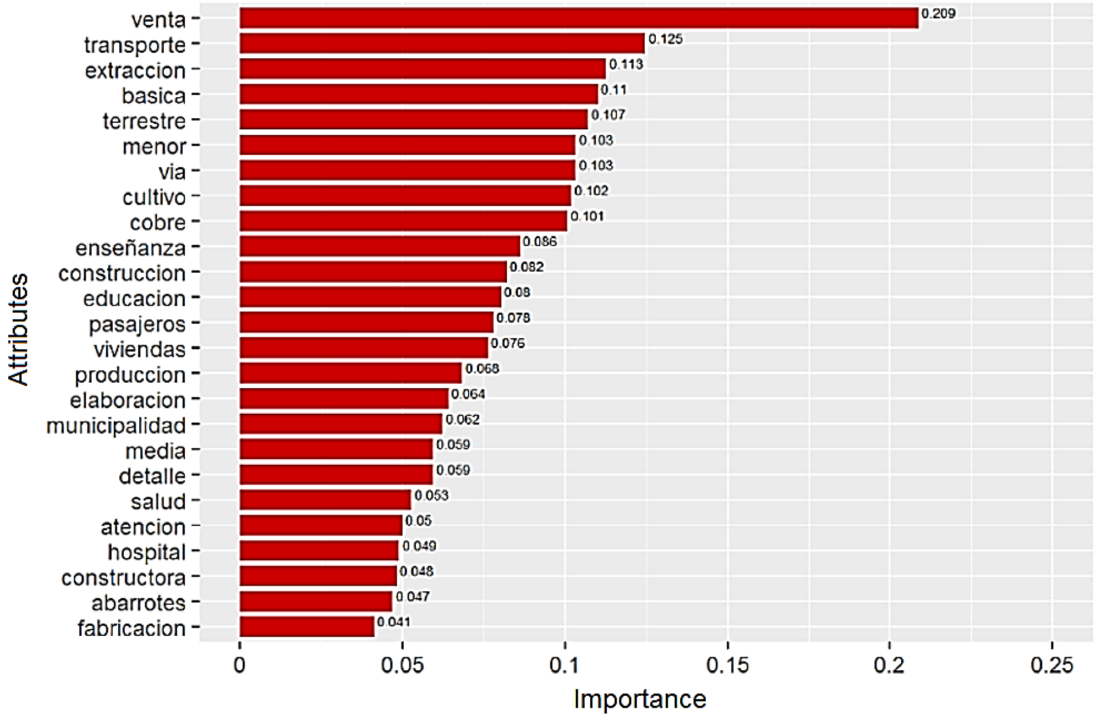
Source: Own elaboration

Chart A2. Information gain of top 25 attributes of CIUO-2



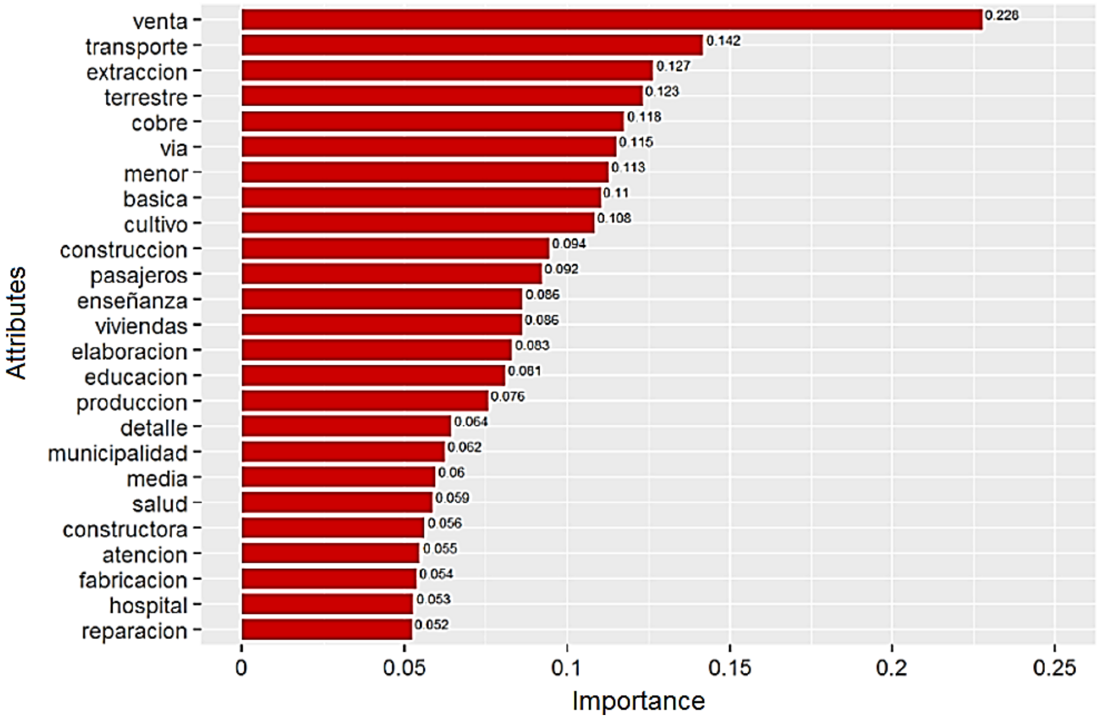
Source: Own elaboration

Chart A3. Information gain of top 25 attributes of CAENES-1



Source: Own elaboration

Chart A4. Information gain of top 25 attributes of CAENES-2



Source: Own elaboration

Annex 2: Results of CAENES-2

Table A1 displays in descending order the numbers of documents in the collection used for training and testing the models for each of the classes of CAENES-2.

Table A1. Training and test data sets for CAENES-2

Class	No. training sets	No. test sets
48. Wholesale and retail trade; repair of motor vehicles and motorcycles	23,908	5,848
1. Crop and animal production, hunting, and related service activities	14,653	3,684
85. Education	13,027	3,201
84. Public administration and defense; compulsory social security	9,064	2,343
41. Construction of buildings	6,775	1,727
86. Human health activities	6,657	1,715
49. Land transport and transport via pipelines	6,652	1,703
56. Food and beverage service activities	5,221	1,280
10. Manufacture of food products	5,071	1,202
43. Specialized construction activities	3,796	874
4. Mining and processing of copper	3,743	977
45. Wholesale and retail trade and repair of motor vehicles and motorcycles	2,810	708
96. Other personal service activities	2,077	468
52. Warehousing and support activities for transportation	1,740	466
81. Services to buildings and landscape activities	1,663	411
14. Manufacture of wearing apparel	1,624	426
69. Legal and accounting activities	1,610	384
3. Fishing and aquaculture	1,528	399
64. Financial service activities, except insurance and pension funding	1,486	373
2. Forestry, logging, and related activities	1,388	356
25. Manufacture of fabricated metal products, except machinery and equipment, and metal working services	1,301	316

Class	No. training sets	No. test sets
16. Manufacture of wood and of products of wood and cork, except furniture; manufacture of articles of straw and plaiting materials	1,256	342
94. Other personal service activities	1,216	288
55. Food and beverage service activities	1,177	295
68. Real estate activities	1,127	288
42. Wholesale and retail trade and repair of motor vehicles and motorcycles	1,014	239
95. Repair of computers and personal and household goods	965	241
61. Telecommunications	935	249
93. Sports activities and amusement and recreation activities	830	222
88. Social work activities without accommodation	817	198
71. Architectural and engineering activities; technical testing and analysis	816	211
35. Electricity, gas, steam, and air conditioning supply	806	197
97. Activities of households as employers of domestic personnel	769	187
33. Repair and installation of machinery and equipment	679	167
11. Manufacture of alcoholic and non-alcoholic beverages	642	180
36. Water collection, treatment, and supply	618	131
31. Manufacture of furniture	587	179
87. Residential care activities	556	135
82. Office administrative, office support, and other business support activities	531	176
62. Computer programming, consultancy, and related activities	519	136
65. Insurance, reinsurance, and pension funding, except compulsory social security	513	128
23. Manufacture of other nonmetallic mineral products	503	103
17. Manufacture of paper and paper products	494	129
22. Manufacture of rubber and plastics products	425	114

Class	No. training sets	No. test sets
20. Manufacture of chemicals and chemical products	419	97
80. Security and investigation activities	410	114
74. Other professional, scientific, and technical activities	400	97
32. Other manufacturing	385	113
90. Creative, arts, and entertainment activities	383	100
8. Other mining and quarrying	367	88
18. Printing and reproduction of recorded media	361	93
73. Advertising and market research	349	74
92. Gambling and betting activities	324	71
38. Waste collection, treatment, and disposal activities; materials recovery	312	85
50. Water transport	302	101
66. Activities auxiliary to financial service and insurance activities	295	74
53. Postal and courier activities	283	78
7. Mining of metal ores, except copper	282	65
77. Rental and leasing activities, except real estate	273	63
24. Manufacture of basic metals	268	65
72. Scientific research and development	214	52
21. Manufacture of pharmaceuticals, medicinal chemicals, and botanical products	203	34
51. Air transport	202	40
30. Manufacture of motor vehicles, trailers, semi-trailers, and other transport equipment	189	58
60. Radio and television programming and broadcasting activities	184	41
78. Activities related to the provision of employment	182	44
27. Manufacture of electrical equipment	181	43
79. Travel agency, tour operator, reservation service, and related activities	177	55

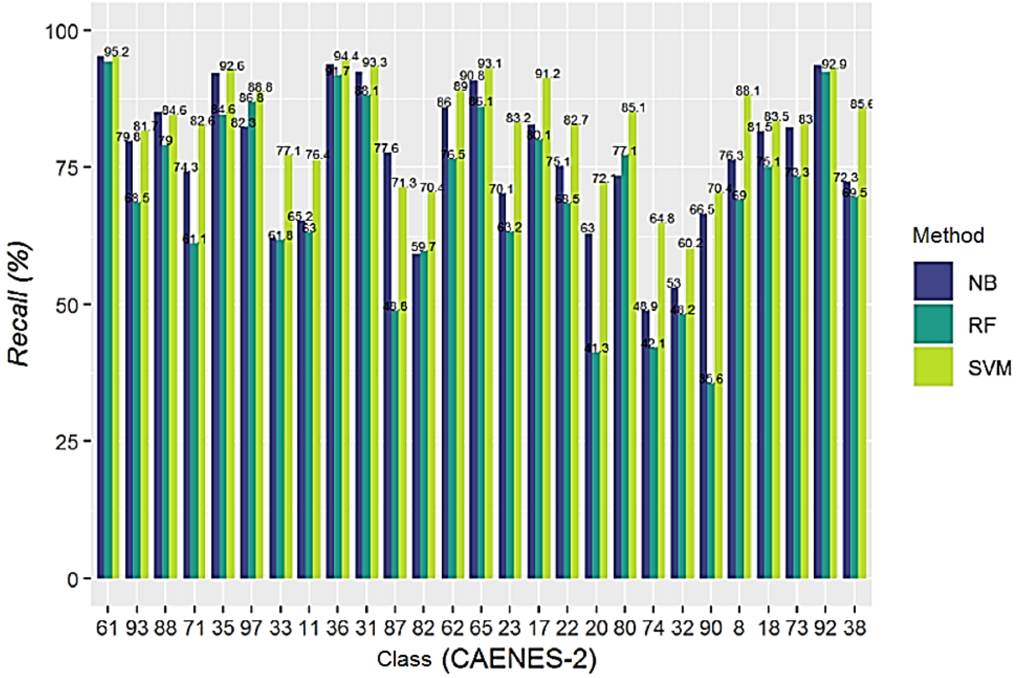
Class	No. training sets	No. test sets
9. Mining support service activities	169	29
59. Motion picture, video, and television program production, sound recording, and music publishing activities	154	44
58. Publishing activities	150	36
75. Veterinary activities	142	29
70. Activities of head offices; management consultancy activities	134	38
91. Libraries, archives, museums, and other cultural activities	110	27
6. Extraction of crude petroleum and natural gas	96	22
19. Manufacture of coke and refined petroleum products	77	21
5. Mining of coal and lignite	50	12
37. Sewerage	43	14
99. Activities of extraterritorial organizations and bodies	24	5
12. Manufacture of tobacco products	23	6
63. Information service activities	18	6

Source: Own elaboration

Charts A5 to A10 show the performances of the three methods in terms of recall, precision, and F1 score for CAENES-2 classes located between position 28 to 81[3], according to the number of training (and test) documents in the collection.

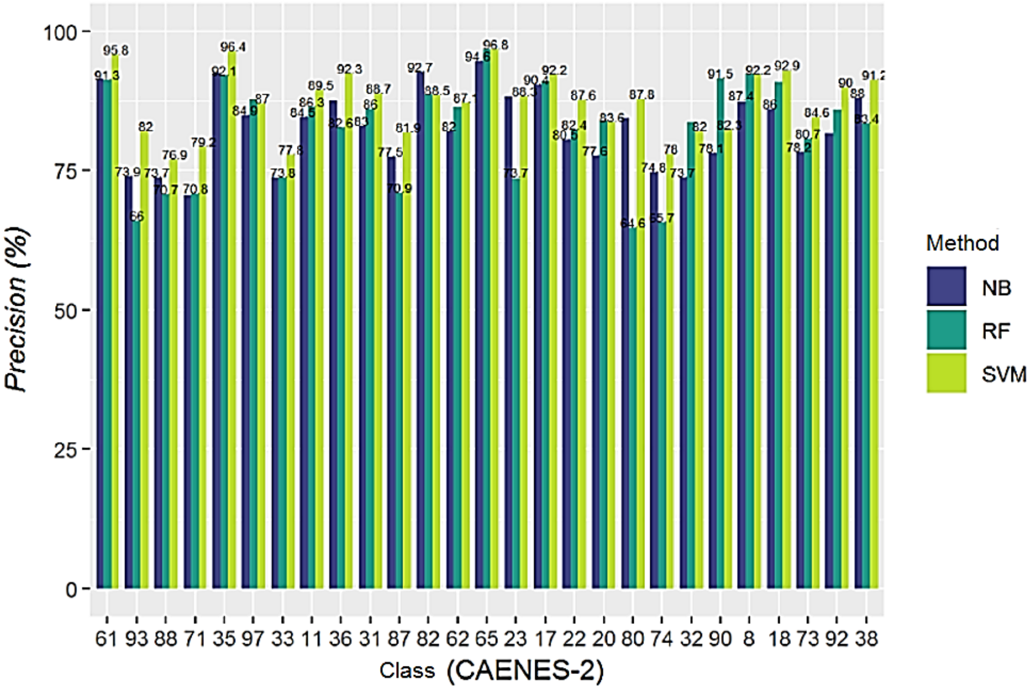
[3] CAENES-2 classification contains 83 classes. However, indicators could be reported for only 81 classes of the data set.

Chart A5. Recall performance, according to method for CAENES-2. Classes in positions 28 to 54, according to the number of documents in the collection.



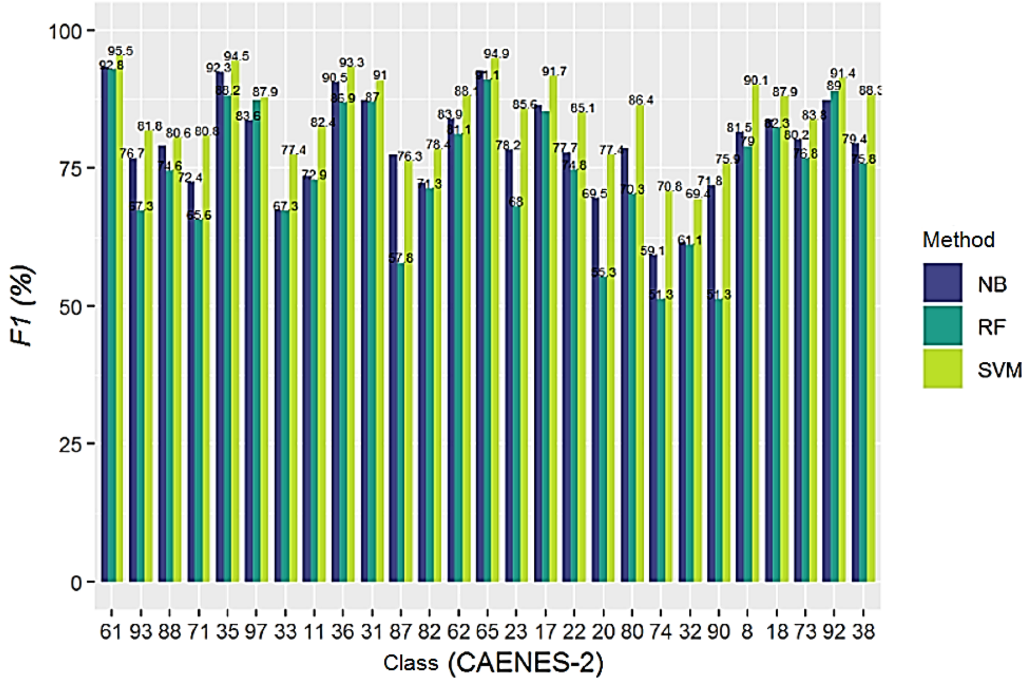
Source: Own elaboration

Chart A6. Precision performance, according to method for CAENES-2. Classes in positions 28 to 54, according to the number of documents in the collection.



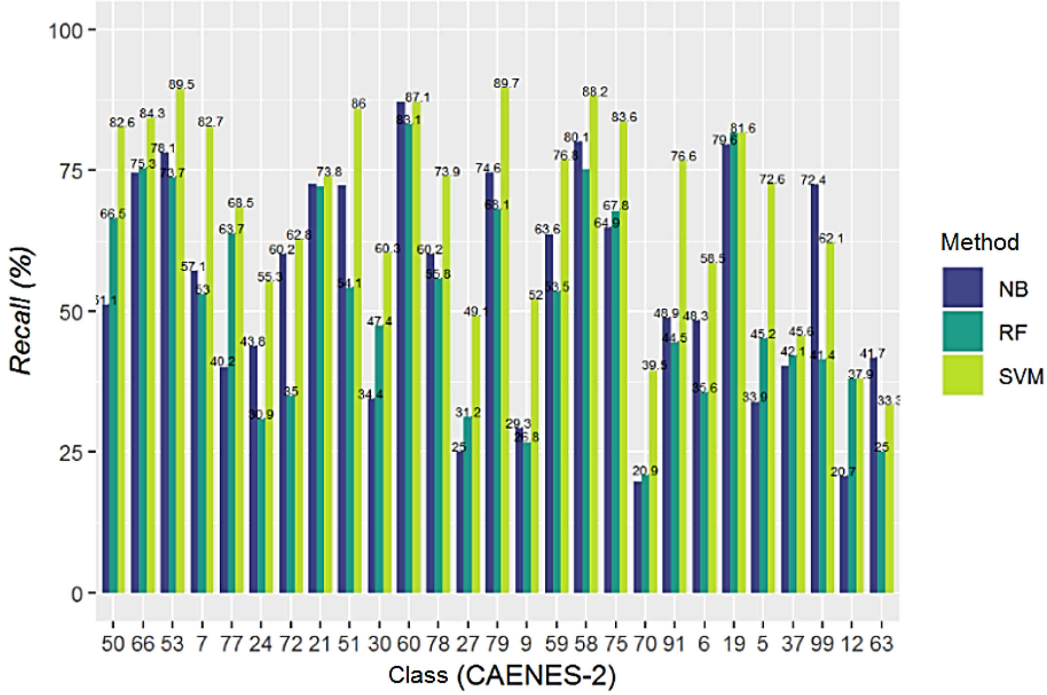
Source: Own elaboration

Chart A7. F1-score performance, according to method for CAENES-2. Classes in positions 28 to 54, according to the number of documents in the collection.



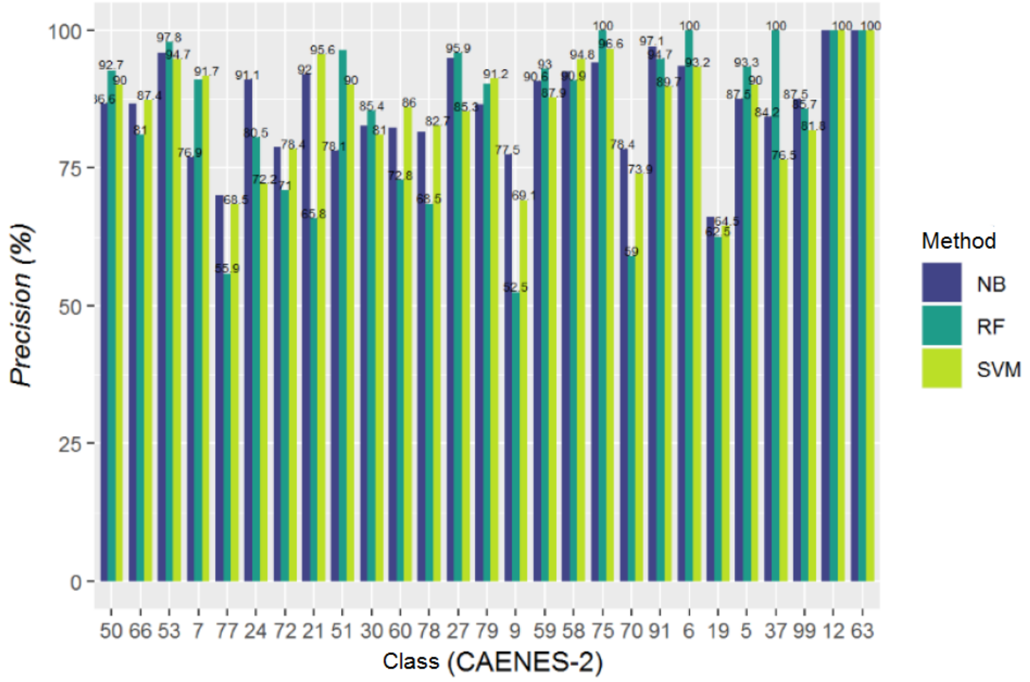
Source: Own elaboration

Chart A8. Recall performance, according to method for CAENES-2. Classes in positions 55 to 81, according to the number of documents in the collection.



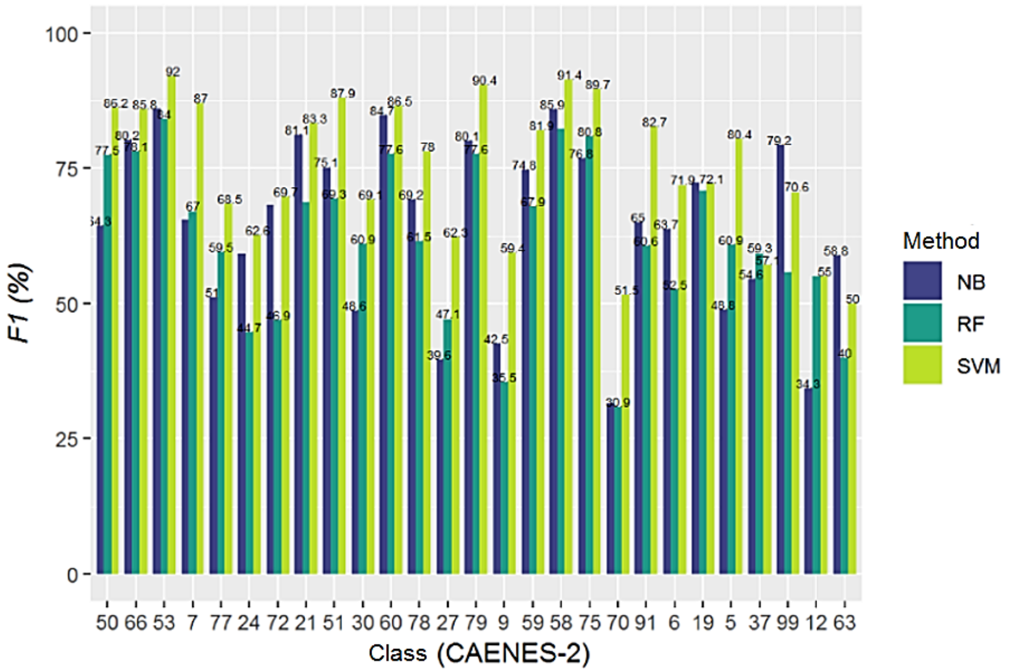
Source: Own elaboration

Chart A9. Precision performance, according to method for CAENES-2. Classes in positions 55 to 81, according to the number of documents in the collection.



Source: Own elaboration

Chart A10. F1-score performance, according to method for CAENES-2. Classes in positions 55 to 81, according to the number of documents in the collection.



Source: Own elaboration

Bibliography

Aas, K. & Eikvil, L. (1999). *Text Categorisation: A Survey*, Norwegian Computer Center, 3 – 37.

Alfaro, R. & Allende, H. (2011). *Text Representation in Multi-label Classification: Two New Input Representations*, International Conference on Adaptive and Natural Computing Algorithms 2011, 1-10.

Berry, M. W. & Kogan, J. (2010). Text Mining: Applications and Theory. United Kingdom: John Wiley & Sons, Ltd.

Breiman, L. (2001). *Random Forests*, Machine Learning, 45 (1), 5 – 32.

Departamento Administrativo Nacional de Estadística (DANE), Colombia (Diciembre de 2005). *Clasificación Internacional Uniforme de Ocupaciones Adaptada para Colombia*. Retrieved from https://www.dane.gov.co/files/sen/nomenclatura/ciuo/CIUO_88A_C_2006.pdf

Instituto Nacional de Estadísticas (INE), Chile (Abril de 2016). *Clasificador de Actividades Económicas Nacional para Encuestas Sociodemográficas*. Retrieved from http://historico.ine.cl/canales/chile_estadistico/mercado_del_trabajo/empleo/metodologia/pdf/caenes.pdf

Joachims, T. (1998). *Text Categorization with support vector machines: Learning with many relevant features*, In Proc. 10th European Conference on Machine Learning (ECML), Springer Verlag.

Liu, Ch., Chan, Y., Alam, S.H. & Fu, H. (2015). *Financial Fraud Detection Model: Based on Random Forest*, International Journal of Economics and Finance 7 (7), 178 – 188.

Manning, CH., Raghavan, P. & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.

Mitchell, T.M. (1997). Machine Learning. McGraw-Hill Science.

Ooms, J. (2017). *High Performance Stemmer, Tokenizer, and Spell Checker*. Retrieved from https://cran.r-project.org/web/packages/hunspell/hunspell.pdf

Pérez, S. (2017). *Análisis y Clasificación de Textos con Técnicas Semi Supervisadas Aplicado a Área Atención al Cliente* (tesis de pregrado). Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile.

R Core Team (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.

Sebastiani, F. (2002). *Machine Learning in Automated Text Categorization*, ACM Computing Surveys, 34(1), 1 – 47.

Welbers, K., Van Atteveldt, W. & Benoit, K. (2017). *Text Analysis in R*, Communication Methods and Measures, 11(4), 245 – 265.

Yang, Y. & Pedersen, J. (1997). *A Comparative Study on Feature Selection in Text Categorization*, Proceedings of the Fourteenth International Conference on Machine Learning, 412 – 420.