Mundo Fesc E-ISSN: 2216-0388 P- ISSN: 2216-0353 Artículo Original

Detection of emergency situations using the Naive Bayes machine learning model.

Detección de situaciones de emergencias usando el modelo Naive- Bayes de machine learning.

Recibido: 26 de agosto de 2022
Aprobado: 4 de diciembre de 2022

Forma de citar: I. L. Vásquez-Rojas and M. J. Vívas-Cortéz, "Detection of emergency situations using the Naive Bayes machine learning model.", *Mundo Fesc*, vol. 13, no. 25, pp. 20–38, Jan. 2023, doi: 10.61799/2216-0388.1286.

Iván Leonel Vásquez Rojas
Decanato Ciencias Económicas y Contables
vasquezivan@ucla.edu.ve
Universidad Lisandro Alvarado
Barquisimeto, Venezuela.

Miguel José Vívas Cortez*
Facultad de ciencias exactas y naturales
mjvivas@puce.edu.ec
Pontificia Universidad Católica del Ecuador
Quito, Ecuador.

***Autor para correspondencia:**
vasquezivan@ucla.edu.ve

Detection of emergency situations using the Naive Bayes machine learning model

Abstract

Nowadays, social networks have gained ground in generating and obtaining instant information. This characteristic makes it very useful in the detection and warning of emergencies such as road accidents, fires, storms, floods, etc. This has led to the generation of a large number of works on the use of this information to address the problems generated by such an emergency, work such as that of A. Kansal, Y. Singh, N. Kumar "Detection of forest fire using Machine Learning technique" [1] or Chamorro Verónica "Classification of tweets using supervised learning models" [2], show the use of machine learning techniques for the detection of extraordinary situations. After these catastrophic or emergency situations, it is necessary to manage the services of care and protection of the population in the face of problems such as: information chaos and uncertainty in the needs and services that can find a solution in the timely detection of events that are truly emergencies. Thus, the purpose of this work is to use messages from X (Twitter) to classify emergencies and determine whether they really are emergencies or not. The machine learning algorithm, known as Naive-Bayes, is used to classify the X messages and determine the real emergencies. The result in the evaluation of the accuracy of the actual emergency classification was a ratio of 73.4% and an accuracy of 75.4% for emergencies classified as false. Overall, the model obtained has an accuracy of 74.6% in its classification predictions. It is considered that the use of a Naive-Bayes model for a prototype in classifying the emergency messages of the social network X could be of great use based on the results of the evaluation of its classification performance.

Keywords: machine learning, Bayes, classification, emergencies, models, twitter

Detección de situaciones de emergencias usando el modelo Naive- Bayes de machine learning

Palabras clave: aprendizaje automático, Bayes, clasificación, emergencias, modelos, twitter

Resumen

En la actualidad, las redes sociales han ganado terreno en la generación y obtención de información al instante. Esta característica la hace de gran utilidad en la detección y advertencias de emergencias tales como accidentes viales, incendios, tormentas, inundaciones, etc. Esto ha motivado la generación de una gran cantidad de trabajos acerca del aprovechamiento de esta información para enfrentar los problemas generados por tal emergencia, trabajo como el de A. Kansal, Y. Singh, N. Kumar "Detection of forest fire using Machine Learning technique" [1] o de Chamorro Verónica "Clasificación de tweets mediante modelos de aprendizaje supervisado" [2], muestran el uso de técnicas de machine learning para la detección de situaciones extraordinarias. Tras estas situaciones catastróficas o de emergencias, es necesario gestionar los servicios de atención y protección de la población ante problemas como: caos informativo e incertidumbre en las necesidades y servicios que pueden encontrar solución en la detección oportuna de eventos que son realmente emergencia. Así, el propósito de este trabajo utiliza mensajes de X (Twitter) para clasificar las emergencias y determinar si realmente lo son o no. Se utiliza para ello el algoritmo de machine Learning, conocido como Naive-Bayes, para la clasificación de los mensajes de X y determinar las emergencias reales. El resultado en la evaluación de la exactitud en la clasificación de emergencia real obtuvo una proporción de 73.4% y una precisión de 75.4% para las emergencias clasificadas como falsas. En general, el modelo obtenido tiene una exactitud del 74.6% en sus pronósticos de clasificación. Se considera que la utilización de un modelo Naive-Bayes para un prototipo en la clasificación de los mensajes de emergencia de la red social X podría ser de gran utilidad con base en los resultados de la evaluación de su performance de clasificación..

Introduction

Currently there are a large number of social networks to which users dedicate a high percentage of their time each day. These social networks have affected people's daily lives. One of the changes that is most evident today is how social networks are displacing traditional information media such as radio, television and others as information media. It is common for people to use Facebook, Twitter or Instagram applications to obtain information of interest or to provide information of interest at the moment.

Thus, among the research on the use of social networks and the data generated by users, the research on obtaining information in emergency or risk situations stands out. In this sense, the development of this work is fair by obtaining a model that allows the use of social networks to alert situations of real risk and consequently palliative measures. The key word here is modeling and as defined in [3] it is "the ability to construct a probabilistic interpretation of an observed phenomenon and the story that goes with it."

One of the most important social networks today is "X" or also known as Twitter, so based on the data provided by Kaggle [4], our objective is to build a Machine Learning model that allows classifying messages on Twitter about emergencies, this It is a model that predicts which messages refer to real disasters or emergencies and which do not.

This type of problem finds a solution in the algorithms obtained from the Naive Bayes method.in the area of machine learning, basically, the Naive-Bayes algorithm works using probability theory and more specifically Bayesian statistics for classification problems, which is summarized in solving problems where the response variables are qualitative as seen in [5], [6] and [7]. Within classification methods, it is common to calculate the probability that a certain observation belongs to the categories of the qualitative variable as the basis of the classification process. So, predicting a qualitative response for a particular observation (this process is known as classification) involves assigning the observation to a certain category.

Among the classification techniques useful for predicting qualitative responses or classification we have logistic regression models, K-close neighborhood, linear discriminant analysis, quadratic discriminant analysis, which we can see in the works of [8] and [9] finally the Naive-Bayes algorithm, very useful for text classification work as indicated in the studies [10], [11], [12] and [13], this last method being the one that we will use in the development of this work for the classification of Twitter messages.

Naive-Bayes model

The Naive-Bayes classification model is frequently used for text classification, it works very well in these cases because this type of problem in which there is information on numerous attributes that must be considered at the same time for the purpose to estimate, based on these attributes, the probability of a result.

Before delving into more details of the Naive-Bayes model, let us consider some important aspects according to [14].

Probability concerns the study of uncertainty; we could see it as the portion of times that an event occurs or how often a certain event occurs. So, in the end we want to measure or quantify uncertainty values, what happens is that we do not determine the probabilities directly from the results of an experiment, instead we work with more convenient numerical variables to determine the probabilities, here the idea of the random variables.

Theorem (Bayes) The conditional probability of an event A, given that another event B occurred, is given by:

$$P(A/B)=\frac{P(A\ y\ B)}{P(B)} \quad (1)$$

Where $P(B)>0$.

The joint probability also denoted by, is the probability that evaluates the possibility of two events occurring simultaneousl y. *P(AyB) P(A∩B)*

Definition: Given two events A and B, they are said to be independent if and only if the occurrence of one of these events does not affect the probability of occurrence of the other event.

Conditional formula for independent events: Two events A and B, are independent if any of these conditions are true:

a.- $P(A/B)=P(A)$ (2)
b.- $P(A/B)=P(B)$ (3)

Multiplication formula

a.- Given any two events A and B, then it is true that:

$$P(A\ y\ B)=P(A).P(A/B) \quad (4)$$

b.- If events A and B are independent, formulas (2) and (3) are reduced (4) to the following:

$$P(A\ y\ B)=P(A).P(B) \qquad (5)$$

Observation: since you have to: $P(A\ y\ B)=P(B\ y\ A)$

$$P(A/B)=\frac{P(A\ y\ B)}{P(B)}=\frac{P(B/A).P(A)}{P(B)} \qquad (6)$$

The final expression (6) is useful to classify (or filter) observations in a problem under established conditions or where there is evidence or knowledge of conditional probabilities, one case for its use is when you want to filter emails that are not desired under certain characteristics. For example, consider emails with the word "offer" as unwanted (spam), then the logical thing would be to classify this email according to the value of the following probability

$$P(span/oferta)=\frac{P(oferta/span).P(spam)}{P(oferta)} \qquad (7)$$

That is, if this probability in (7) is greater than a certain number, classify it as spam. When you know how many spam emails have the word "offer", how many emails are spam and how many have the word "offer".

Total Probability Theorem.

Definition: A set of events {E1, E2,…, En}, mutually exclusive and collectively exhaustive of a sample space S, are known as a partition of Ω. Or equivalently

- $\Omega = E1 \cup E2 \cup \ldots, \cup En$ (colativamente exhaustivo)
- $Ei \cap Ej = \emptyset$ (son disjuntos)

Theorem (total probability). Let {E1,E2,…,En} be a partition of a sample space Ω, such that P(Ei)>0,i=1,…n. Then for any event A, it holds that:

$$P(A)=P(E1).P(A/E1)+P(E2).P(A/E2)+\cdots+P(En).P(A/En) \qquad (8)$$

Bayes' rule in Bayes' theorem in (1), together with the total probability theorem, allows us to obtain the following result, also known as Bayes' rule.

Let {E1,E2} be a partition of a sample space Ω, such that P(Ei)>0,i=1,2. Then for any event B, it holds that:

$$P(E_1/B)=\frac{P(E_1).P(B/E_1)}{\sum_i P(E_i).P(B/E_i)} \qquad (9)$$

Random variable X is a numerical description of the possible results of an experiment. A random variable is viewed as a function that takes elements in the sample space and assigns them a numerical value.

Frequent concepts in probability are:

- Sample space Ω, which is the set of all possible outcomes of an experiment.
- Event space A is the space whose elements are events A. An event A is a subset of the sample space Ω. We define an event as the set formed by none, one or more results of an experiment.
- The probability P, a number P(A) is associated with each event A that measures the possibility or belief value that the event may occur.
- Thus a random variable

Discrete and continuous random variables: depending on whether the objective space Γ is discrete or continuous, in the same way we will refer to the type of random variable. We can then see that for a random variable Y, the set, where they are sample points in Ω to which X assigns 0. Then it is the sum of the probabilities of the sample points to which the value of x is assigned.{Y=0}={E_1,E_2,…,E_r } E_1 P(X=x)

Definition: The probability distribution for a random variable X can be represented by a formula, a table or a graph, which provides $P(X=x)\ \forall x$

Theorem: Any distribution of discrete random variable it is true that

- $0 \leq P(x_i) \leq 1$
- $\Sigma P(x_i)=1$

When the objective space is discrete we can determine the probability that a random variable takes a particular value and we denote it with. When the objective space is continuous, the natural thing is to determine the probability that a random variable is in an interval. $x \in \Gamma P(X=x)$

Definición: If X is a continuous random variable, then the probability distribution function of $F(x)=P(X<x)$ para todo $-\infty \leq x \leq \infty$

Definición: Definition: If is a distribution function of X, then

$$f(x)=\frac{dF(X)}{dx}=F'(Y) \quad (10)$$

It is defined as the density function of

$$F(x)=P(X<x)=\int_{-\infty}^{x} f(t)dt \quad (11)$$

Theorem: Yes *f(x)* is a density function of a random variable

1.- $f(x)\geq 0, \forall x$
2.- $F(\infty)=P(X<\infty)=\int f(t)dt$

Multivariate probability.

In order to extend the concepts and properties in probability distributions with more than one random variable below they are shown with only two variables, and the general case is similar.

Definition: If they are two discrete random variables, then the joint probability of this is given by X_1 y X_2 X_1 y X_2

$$F(x_1,x_2)=P(x_1,x_2)=P(X_1=x_1,X_2=x_2)$$

Theorem: If there are two discrete random variables with a joint probability function, then the following holds: X_1 y X_2 $F(x_1,x_2)=P(x_1,x_2)$

- $0\leq P(x_1,x_2)\leq 1 \forall x_1,x_2$
- $\sum_{x_1, x_2} P(x_1,x_2)=1$

Definition: If they are two continuous random variables, the joint probability distribution function is given by X_1 y X_2

$$F(x_1,x_2)=P(x_1,x_2)=P(X_1<x_1,X_2<x_2)$$

Definition: If they are two continuous random variables, with a joint probability distribution function given by . If there exists a non-negative function such that X_1 y X_2 $F(x_1,x_2)F(x_1,x_2)$

$$F(x_1,x_2)=\int_{-\infty}^{x_1} \int_{-\infty}^{x_2} f(t_1,t_2)dt_1\, dt_2 \quad \text{para } -\infty<x_1,x_2<\infty$$

Then we say that they are joint continuous random variables and is the joint density function of X_1,X_2, $f(x_1,x_2)$

Theorem: If $f(x_1,x_2)$ is a joint continuous density function of X_1,X_2.

$$1.- f(x_1,x_2)\geq 0, \forall x_1,x_2$$
$$2.- \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(t_1,t_2)dt_1\, dt_2 =1$$

Definition: be $F(x_1)$and distribution functions of respectively and the joint distribution function of. Then they are said to be independent if and only if $F(x_2)$ X_1,X_2 $F(x_1,x_2)$ X_1,X_2 X_1 y X_2

$$F(x_1,x_2)=F(x_1).F(x_2)\ \forall x_1,x_2 \quad (12)$$

If they don't call themselves dependent.

Let us note that the above definition is consistent with the definition of event independence. This property of independence underlies the Naive-Bayes model, since it is based on assuming that the variables have this characteristic.

Naive-Bayes algorithm

According to [15] and [16], basically the Naive-Bayes algorithm works using Bayes theorem for classification problems. A classic example of classification consists of determining spam emails, as shown in [17]. This assumes that all characteristics are equally important and independent. This condition is rarely true in real life, but despite this assumption Naive Berries works well. One explanation is that the precise value of the estimated probability does not matter as long as the prediction is accurate.

So, if we want to classify an observation X into one of the k-classes, with k>=2. Let E1, E2 ,…, Ek be the k-classes of the response variable, such that P(Ei)>0,i=1,…n. Then by Bayes' theorem in (10), we have that:

$$P(Y=Ei/X=x)= \frac{P(E).\ P(X=x/Y=E_i)}{P(E1).\ P(X=x/Y=E_i)+...+P(E_n).P\ (X=x/Y=E_n)} \quad (14)$$

To simplify the notation let's do $p_i(x)=P(Y=E_i/X=x)$ which is the probability that an observation belongs to class i.

In the previous equation (14), instead of calculating directly as we did in the logistic model, we are going to estimate y. Generally, the estimation of is easy if we have a random sample of the population, we simply calculate the fraction of training observations that belong to the ith class. However, estimating is a little more complex and as we will see, it will be necessary to make some assumptions. Let us remember that the Bayes classifier of formula (12) classifies an observation x to the class for which it is the largest. So if we can find a way to estimate , we can use equation (12) in order to approximate the Bayes estimator $p_i(x)P(E_i)P(X=x/Y=E_i)P(E_i)P(X=x/Y=E_i)\ p_i(x)P(X=x/Y=E_i)$

Before developing the Naive Bayes method, let us consider the p-dimensional density function for an observation x in the -ith class. The method is based on assuming the independence of the predictors for each class, so from (11) we have: $f_i(x)$

$$f_i(x)=f_{i1}(x1) \times f_{i2}(x_2)\times...\times f_{iP}(x_P) \quad (15)$$

For all y is the density function of the nth predictor among the observations of the class (if the variable is discrete then it is the probability of the nth predictor among the observations of the class). Thus, there is no association of the predictors in each class. This assumption most of the time does not occur but even then, this assumption is made obtaining good results. $i=1,...,nf_i\, j\; E_i f_i\, j= p\ (x_j/E_i)E_i$

Thus, assuming this independence as in (13) we have that:

$$P(Y=Ei/X=x)=\frac{P(E_i)\times P(X=x/Y=E_i)}{P(E_1)\times P(X=x/Y=E_1)+...+P(E_n).P\ (X=x/Y=E_n)}$$

$$=\frac{P(E_i)\times f_{i1}(x_1 f_{i2} x_2 \times ... \times f_{ip}(x_p)}{\sum_{l=1}^{n} P(E_l)\times f_{l1}\ (x_1)\times f_{i2}\ (x_2)\times ...\times f_{lp}(x_p)}$$

To estimate the one-dimensional density function using the training data we have several options: f_{ij}

1. With quantitative, then we can assume that it is approximately normal. That is, we assume that for each class the nth predictor is described by a normal distribution. $X_j\ X_j/Y=E_n$

2. With quantitative, then another option is to use a non-parametric estimate. A simple way to do this is to make a histogram for the predictor observations with each class. SoX_j $f_{kj}\ f_{kj}$

3. With quantitative, then. We can simply count the proportion of observations for the nth predictor corresponding to each class. For example, suppose that y we have 100 observations in the kth class. Suppose that the nth predictor takes about the values 1,2 and 3 in 32, 55 and 13 of those observations respectively. Then we estimate how X_j $X_j \epsilon \{1,2,3\}\ f_{kj}$

PTo illustrate how Naive-Bayes works using the third method described above, suppose that we want to classify objects into two different types of classes A or B, these objects meet the characteristics w1, w2, w3 and w4, which we assume are independent. then our method is based on determining the probability ~~

$$P(A/w1\cap \sim w2\cap \sim w3\cap w4)=\frac{P(w1\cap \sim w2\cap \sim w3\cap w4/A).P(A)}{P(w1\cap \sim w2\cap \sim w3\cap w4)}$$

The development of the formula will be computationally difficult to calculate each time we add more features to the problem which is what normally happens, given this situation a large amount of memory will be needed to store the probabilities of all the possible intersections. So if we assume what the Naive-Bayes algorithm does about independence, the previous expression would be transformed into:

$$P(A/w1 \cap \sim w2 \cap \sim w3 \cap w4) = \frac{P(w1 \cap \sim w2 \cap \sim w3 \cap w4/A).P(A)}{P(w1 \cap \sim w2 \cap \sim w3 \cap w4)}$$

$$= \frac{P(w1/A).P(\sim w2/A).P(\sim w3/A).P(w4/A).P(A)}{P(w1).P(\sim w2).P(\sim w3).P(w4)}$$

YAnd consequently, the probability that it is B, given that it has the characteristics w1, ~w2, ~w3 y w4, es:

$$P(A/w1 \cap \sim w2 \cap \sim w3 \cap w4) = \frac{P(w1 \cap \sim w2 \cap \sim w3 \cap w4/A).P(A)}{P(w1 \cap \sim w2 \cap \sim w3 \cap w4)}$$

The resulting expressions will be those that participate in the algorithm since they are less difficult to calculate to determine the established conditional probability.

Types of Naive Bayes classifiers:

- **Multinomial Naive Bayes:** used mostly for document classification, this algorithm uses the frequency of the words present in the text to determine which class it belongs to.
- **Bernoulli Naive Bayes:** similar to multinomial with the difference that instead of considering the frequency of the words, it establishes the variable as Boolean, that is, whether or not it is in the text.
- **Gaussian Naive Bayes:** here we consider the case that the predictor variables are continuous.

Possibly one of the never-happening features adds an additional drawback, resulting in a conditional probability that it is spam (or not) equal to zero (or 100% probability). But the remaining characteristics could usually be associated with spam messages, so an alternative to avoid this drawback is to use the Laplace estimator.

Evaluation of the performance of the selected model

Evaluating the performance of machine learning algorithms is one of the fundamental tasks in their construction. Questions such as "How to measure the success of a model" or "how would I know if I have been successful" are answered by evaluating the machine models. Learning.

In order to measure the performance of a machine learning model, the best way is to capture whether the classifier is successful in its purpose. When we develop a classification model, we generally have the following elements: current values of the class, predicted values of the class and estimated probability of the prediction. Then the current class values and the predicted values will be a key part of the evaluation.

If we consider a binary classification then there are only two possible classes; in the case of having more than two classes the methods are easily extensible to these cases.

There are several ways to measure performance in classification models see [18]: Accuracy, confusion matrix, log-loss, AUC and precision-recall.

The accuracy, is the most common measure of the performance of a classification model, it is a metric that measures the frequency with which the classifier correctly makes the prediction. It is defined as the ratio of the correct number of predictions to the total number of points in the test set.

$$\text{Exactitud} = \frac{\text{número de predicciones correctas}}{\text{número total de puntos}} \quad (16)$$

A confusion matrix or table, shows in detail the correct and incorrect classifications in each class. The classes of interest are known as positive class, while the others are known as negative classes. On the other hand, where the model makes a correct prediction, we know it as true and where the model is wrong, we know it as false, then in a two-class classification the following will be observed:

Table I. Model prediction

	Model prediction		
Real value		Yeah	No
	Yes (positive)	True positive TP	False positive FP
	No (negative)	False negative FN	True negative TN

The sensibility of the model, also called positive class, measures the proportion of positive elements correctly classified among those that are truly positive.

$$\text{sensibilidad} = \frac{\text{TP}}{\text{TP+FN}} \quad (17)$$

The specificity of the model, also called negative class, measures the proportion of negative observations correctly classified among those that are truly negative.

$$\text{especificidad} = \frac{\text{TP}}{\text{TP+FP}} \quad (18)$$

AUC, visualization are useful to understand the performance of the machine learning algorithm in general. The visualization represents how a model works across a wide range of conditions. Since two algorithms have different biases, it is possible that two models with similar accuracy in their performance will have large differences in how they achieve their accuracy. The receiver operating characteristic (ROC) curve is commonly used to examine the contrast between the class of true positives while avoiding false positives.

The classification characteristics on the ROC curve can be measured by the area under the curve (AUC) ranging from 0.5 for no predictive value to 1 for perfect classifiers. Values close to 1 do not indicate a model with high performance.

Classification of X's messages.

Twitter is one of the most popular social networks in the world with more than 300 million users, generating around 500 million tweets (posts) daily and more than 350 thousand per minute, which mainly uses text messages to share information from various sources. natures, sports, music, politics, advertising, events or events and many more. Thus, it is natural that it is used as a means of instant information for emergencies.

In order to classify messages on Twitter to know which refer to real emergencies and which do not, we will use the tm, Snowball, wordcloud, e1071, tydiverse packages in the R programming language and the data set obtained from Kaggle [3]. These data are provided in a file which contains the variables id, keywords, text (where the messages are) and the variable target (label) that shows their classification, in real emergencies or not with the numbers 1 and 0 respectively. It also contains 7613 observations or messages as in Figure 1.

id	keyword	location	text	target
1			Our Deeds are the Reason of this #earthquake May ALLAH Forgive us all	1
4			Forest fire near La Ronge Sask. Canada	1
5			All residents asked to 'shelter in place' are being notified by officers. No other evacuation or shelt	1
6			13,000 people receive #wildfires evacuation orders in California	1
7			Just got sent this photo from Ruby #Alaska as smoke from #wildfires pours into a school	1
8			#RockyFire Update => California Hwy. 20 closed in both directions due to Lake County fire - #CAfire	1
10			#flood #disaster Heavy rain causes flash flooding of streets in Manitou, Colorado Springs areas	1
13			I'm on top of the hill and I can see a fire in the woods...	1
14			There's an emergency evacuation happening now in the building across the street	1
15			I'm afraid that the tornado is coming to our area...	1
16			Three people died from the heat wave so far	1
17			Haha South Tampa is getting flooded hah- WAIT A SECOND I LIVE IN SOUTH TAMPA WHAT AM I GOI	1
18			#raining #flooding #Florida #TampaBay #Tampa 18 or 19 days. I've lost count	1
19			#Flood in Bago Myanmar #We arrived Bago	1
20			Damage to school bus on 80 in multi car crash #BREAKING	1
23			What's up man?	0

Figure 1. Clasification of the messages

We divided this data set into two, a first set with 75% of the data (5710 messages) to train the model and the second set with 25% of the data (1903 messages) to evaluate the model obtained.
Some words may appear in both messages classified as real emergencies and in those classified as non-emergencies, but the Naive Bayes algorithm takes into account the frequencies of these words in the messages to establish a probability that determines how a message will be classified. For example, Figure 2 represents the most frequent

words in emergency messages in a words cloud.

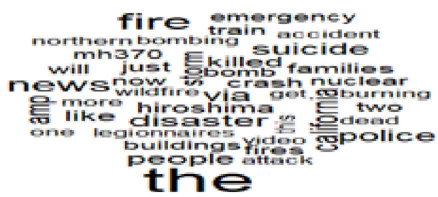

Figura 2. Words cloud the words more frequents in the messges of emergency.

Data processing

In this first part we try to give structure to the data as in [19], we carry out the work of text mining with the use of R and using tools provided by the tm package in R. Corpus creation and manipulation methods are shown, text processing and creation of document matrices.

The fundamental structure called corpus is seen as objects that store collections of documents, the definition of corpora allows operations to be carried out with them, along with their documents, and also provides accessibility to documents and metadata.

```
tw_corpus<-VCorpus(VectorSource(tw_datos$text)
```

Data transformation.

Once we have the corpus defined, and it is necessary to have uniformity in the text, this can be seen as a text analysis, here we discover relevant information that may not be visible, with the text processing, we extract related words, we remove ambiguities such as frequently used words, signs or symbols, we uniformize the text messages, we clean the data, in short, we transform it into a suitable format for the model training. To do this we use the functions within tm_map (), which apply to all elements of the corpus. In order to finish the preparation phase, we are going to divide the message into individual terms using tokenization, a token is a simple element of text or simply a word, from this a frequency matrix is generated where the frequency of each word is indicated. In this stage we will use a function called DocumentTermMatrix () which takes a corpus and creates a structure called DTM in which its rows are the SMS (documents or message) and in the columns are the words (or terms) and the frequencies of each word per message (or document).

```
tw_dtm2<-DocumentTermMatrix(tw_corpus,control = list(
 tolower=TRUE,
 removeNumbers=TRUE,
 stopwords=TRUE,
```

```
 removePunctuation=TRUE,
 stemming=TRUE,
 stripWhitespace= TRUE
))
```

A look at a part of the DTM matrix obtained shows us its structure, the 7613 documents (messages) and the 18394 total terms (words) of the messages and their frequencies in each message.

```
<<DocumentTermMatrix (documents: 7613, terms: 18394)>>
Non-/sparse entries: 70234/139963288
Sparsity           : 100%
Maximal term length: 51
Weighting          : term frequency (tf)
Sample             :

      Terms
Docs   amp bomb fire get just like new now via will
1541   0   2    0    0   0    0    0   0   0   0
1832   1   0    0    0   0    0    0   0   0   0
1910   0   0    0    0   0    0    0   0   0   0
1943   0   0    0    0   0    0    0   1   0   0
2149   0   0    0    1   1    0    0   0   0   0
4189   0   0    1    0   0    0    0   0   0   0
4199   0   0    0    0   0    0    0   0   0   0
4458   0   0    0    0   0    0    0   0   0   0
5981   2   0    0    0   0    2    0   0   0   0
6129   0   0    0    0   0    0    0   0   1   0
```

Then we separate the data into two sets (tw_dtm_train) with 75% to train the model and another (tw_dtm_test) with the remaining 25% of the data to evaluate the accuracy of the obtained model.

Conjunto de entrenamisento:

```
tw_dtm_train<-tw_dtm2[1:5710,]
tw_dtm_test<-tw_dtm2[5711:7613,]
```

Reduce the dimension.

We then remove words with low frequency in each of the training sets and the evaluation set. We then eliminate any words that appear in less than 5 messages, thus significantly reducing the matrices and not affecting the matrix's inherent relationships.

The classifier we use in R is based on the Bernoulli Naive-Bayes classifier, which works with categorical variables, so we generate a matrix that assigns "no" to the boxes with zero frequency and "yes" to the boxes with a frequency other than zero in the frequency matrix obtained previously. For both the training and evaluation set.

```
conver_counts<-function(x){x<-ifelse(x>0,"Yes","No")}
tw_train1<-apply(tw_dtm_freq_train,MARGIN = 2,conver_counts)
```

Naive Bayes with R. Thus, having the sets of message rows converted into a format that can be represented by a statistical model, let's apply the Naive Bayes algorithm from the e10741 package to the training set to obtain the prediction model. One way to do this is illustrated in [20].

This will use the presence or absence of words to estimate the probability that a given message is a true emergency or not.

```
>tw_classifier<-naiveBayes(twF_train,tw_train_label)
  0    1
3296 2414
```

Model evaluation. To evaluate the classifier we obtained, we will use the messages in the test set along with those with the labels stored in the test vector. We thus generate a prediction, then compare the classes predicted by the model with the actual classification using a cross-score table and determine the metrics of accuracy, sensitivity and specificity.

```
tw_test_pred<-predict(tw_classifier,twF_test)
str(tw_test_pred)
CrossTable(tw_test_pred,tw_test_label,prop.chisq = FALSE,prop.c = FALSE,
        prop.r = FALSE,
        dnn = c("predic","actual") )
```

```
             | actual
      predic |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
           0 |       834 |       271 |      1105 |
             |     0.438 |     0.142 |           |
-------------|-----------|-----------|-----------|
           1 |       212 |       586 |       798 |
             |     0.111 |     0.308 |           |
-------------|-----------|-----------|-----------|
Column Total |      1046 |       857 |      1903 |
-------------|-----------|-----------|-----------|
```

Thus, we define the positive class as (857/1903) and the negative class (1046/1903) as

the messages
- True positives: correctly classified in the class of interest 586
- True negatives: correctly classified in the class of no interest 834
- False positives: incorrectly classified in interest class 271
- False negatives: incorrectly classified in class of non-interest 212

Using equations (14), (15) and (16) we have the following results:
Accuracy=0.746
Sensitivity=0.734
Specificity=0.754
AUC: Area under the curve= 0.7406, this value represents the area under the curve in figure 3.

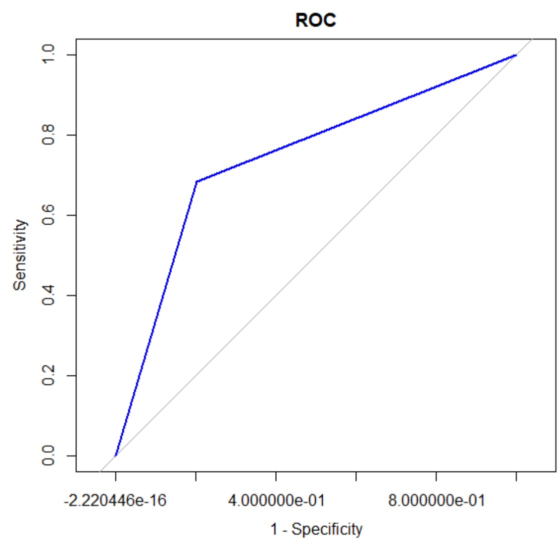

Figure 3. Area under the curve

Conclusion

We note how the use of the Naive-Bayes method for the detection of real emergencies or catastrophes based on messages from X (Twitter) generates a model according to the evaluation capable of classifying a real emergency with a proportion of 73.4% among those classified as emergencies. (sensitivity) and classifies false emergencies with an accuracy of 75.4% (specificity) among those classified as false. In general, the model obtained has an accuracy of 74.6% in its forecasts.

We also observed that the model is wrong 11.1% of the time when classifying messages as emergencies that really are not. It also does it 14.1% of the time with those that are truly emergencies. The AUC or area under the curve with a value of approximately 0.74 confirms an acceptable level of model performance to predict real and non-real emergencies.

The use of machine learning models for security, risk, management of emergency and disaster situations has been increasing, currently the use of data analysis and forecasting for decision making is becoming more frequent, due to the implications clear that it entails good or bad decisions, being a decisive factor in the management of emergency situations. With this work we show the use of one of the algorithms in such situations, where the information from the networks serves as a data source. This demonstrates a naturally growing field where the application of models and machine learning techniques can contribute to problem solutions in these cases. The use of different models in order to determine which one has the best performance in classification could result in a useful prototype in solving problems of this class.

It is clear that by taking advantage of the development of machine learning and data science, opportunities and challenges are opening up in these works and consequently there will be new information that leads to predictions with better accuracy. New, more complex methods are built, more scalable for large amounts of data, surpassing conventional models. In short, the use of machine learning methods can contribute to improving the management of emergency situations, supporting analysis and decision making. Also, as a tool that allows resources to be optimized in such situations.

References

[1] A. Kansal, Y. Singh, N. Kumar y V. Mohindru "Detection of forest fire using Machine Learning technique" Journal of an university of Architecture & Technology. Oct 2020. https://www.researchgate.net/publication/344462171_Forest_Fires_Detection_Using_Machine_Learning_Techniques

[2] V. Chamorro, "Clasificación de tweets mediante modelos de aprendizaje supervisado". Universidad complutense 2018 https://hdl.handle.net/20.500.14352/14246

[3] Kaggle, Machine Learning and Data Science Community, "Predicting tweeter sentiment", 2021. https://www.kaggle.com/code/yasserh/predicting-twitter-sentiments-top-ml-models

[4] F. Harrell. Regression Modeling Strategies, Springer. Nashville, USA, 2015.

[5] O. Basheer "Application of Naïve Bayes to students ´performance classification". Asian Journal probability and statistics. Nigeria, September 2023. https://www.researchgate.net/publication/374113429_Application_of_Naive_Bayes_to_Students%27_Performance_Classification

[6] Z. Zhang "Sentiment Analysis of twitter comments using Naïve Bayes Classifier". Communications in humanities research, October 2023. https://www.researchgate.net/publication/375114697_Sentiment_Analysis_of_Twitter_Comments_Using_Naive_

Bayes_Classifier

[7] K. Chitra. “Netflix ranking by combination of k-nearest neighbor and singular value decomposition” International Journal of computational Science and engineering, India 2020. https://www.ripublication.com/ijcse20/ijcsev10n1_01.pdf

[8] T. Fontalvo “Aplicación del análisis discriminante para evaluar el mejoramiento de los indicadores financieros en las empresas del sector de extracción de petróleo y gas natural en Colombia” Soluciones del posgrado EIA, 2011. https://revistas.eia.edu.co/index.php/SDP/article/view/340

[9] G. Castillo. “Técnica de clasificación bayesiana para identificar posible plagio en información textual”. Revista cubana de ciencias informáticas, La Habana 2014. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992014000400008

[10] L. Dubiau. “Análisis de sentimientos sobre un Corpus en español”. Facultad de ingeniería, Universidad Buenos Aires. 2013. https://42jaiio.sadio.org.ar/proceedings/simposios/Trabajos/ASAI/04.pdf

[11] D. Buzic. “Lyrics Classification using Naive Bayes”. International convention on information and communication Technology. Croatia 2018. https://www.researchgate.net/publication/325645213_Lyrics_Classification_Using_Naive_Bayes

[12] H. Ramadhan. “Sentiment analysis on Indonesia-English code-mixed data”. International conference for convergence in technology. April 2023.

[13] D. Wackerly. Estadística matemática con aplicaciones. Miami Florida, Thomson, 2002.

[14] G. James, D. Witten, T. Hastie, R. Tibshirani. An introduction to Statistical Learning. Springer. August 2021. https://www.statlearning.com/

[15] B. Lantz. Machine Learning with R. Pack Publishing LTD. Apr 2019. https://www.packtpub.com/product/machine-learning-with-r-third-edition/9781788295864

[16] T. Hastie. The elements of statistical learning, Springer. California, USA, 2008.

[17] A. Zheng. Evaluating Machine Learning Models, O´Really, USA, 2015.

[18] I. Feinerer. “Text mining Infrastructure in R”. Journal of statistical software, March 2008. https://www.jstatsoft.org/article/view/v025i05

[19] J. Bosco Naive Bayes con R para clasificación de textos. R Pubs by Rstudio, Abril 2018. https://rpubs.com/jboscomendoza/naive_nayes_con_r_clasificacion_texto