







UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS



Research

Community-Based Early Warning System Model for Stream Overflow in Barranquilla

Modelo de sistema de alerta temprana para desbordamiento de arroyos en barranquilla basado en la comunidad

Iván Andrés Felipe Serna-Galeano¹, Ernesto Gómez-Vargas¹, and Julián Rolando Camargo-López¹

¹Universidad Distrital Francisco José de Caldas, Bogotá, Colombia 

Abstract

Context: This work aims to design and create an early warning model based on the community as an alternative for the mitigation of the disaster caused by the streams that overflow in Barranquilla (Colombia). This model is based on the contributions in social networks, which are consulted through the API of each social network and filtered according to their location.

Methods: With the information collected is performed cleaning and debugging, and then with natural language processing techniques tokenize vectorize the texts, seeking to operate mathematically to find the vector similarity between processed texts, thus generating a classification between texts associated with stream overflow and texts that are not associated with overflow.

Results: The texts classified as stream overflow are processed again to obtain a location or assign a default one, to consequently georeferenced these data in a map that allows to associate the risk zone and visualize it in a web application, monitoring and reducing the possible damage generated to the population.

Conclusions: To choose the best classifier, 3 classification algorithms were selected (random forest, extra tree, and k-neighbor), which presented better performance and R2 in reference to the data processed in the regressions performed. the three algorithms were trained, and the k-neighbor algorithm was found to be the best.

Keywords: Stream overflow, social network, Machine learning, Natural language processing

Article history

Received:
14th/Feb/2024

Modified:
17th/Mar/2024


Accepted:
18th/Apr/2024

Ing., vol. 29, no. 2,
2024, e21846

©The authors;
reproduction right
holder Universidad
Distrital Francisco
José de Caldas.

Open access



* **Correspondence:** jcamargo@udistrital.edu.co

Resumen

Contexto: Este artículo busca crear un modelo de alerta temprana como alternativa para mitigar el desastre provocado por los arroyos en Barranquilla Colombia, este modelo está basado en la comunidad por medio de redes sociales, consultando el api de cada red social, en donde se filtra por área de localización.

Métodos: Con la información recolectada y depurada por medio de técnicas de procesamiento de lenguaje natural se convierten los textos en vectores, con el fin de clasificar en base a la similitud vectorial entre los textos procesados, generando así una clasificación entre los textos asociados al flujo de desbordamiento y los textos que no lo son. asociado con el desbordamiento.

Resultados: Los textos clasificados como desbordamiento de arroyos son nuevamente procesados para obtener una ubicación o asignar una predeterminada, para posteriormente georreferenciar estos datos en un mapa que permite asociar la zona de riesgo y visualizarla en una aplicación web, monitoreando y reduciendo los posibles daños generados a la población.

Conclusiones: Para elegir el mejor clasificador se seleccionaron 3 algoritmos de clasificación random forest, extra tree y k-neighbor), los cuales presentaron mejor rendimiento y R2 en referencia a los datos procesados en las regresiones realizadas. Se entrenaron los tres algoritmos y se descubrió que el algoritmo k-neighbor era el mejor.

Palabras clave: arroyos, redes sociales, aprendizaje automático, procesamiento de lenguaje natural

Table of contents

		2.5.1. Exploratory data analysis and preprocessing	6
		2.5.2. Training	7
1. Introduction	2	3. Results	7
2. Methodology	3	4. Discussion	11
2.1. Model structure	3	5. Conclusions	14
2.2. Data collection	5	6. Acknowledgements	14
2.3. Data cleaning	6	7. CRediT author statement	14
2.4. Classifying the collected data	6	References	14
2.5. Information processing	6		

1. Introduction

Floods are natural risk events produced by excess water from rivers in areas that have been invaded in normally dry conditions (1). The city of Barranquilla (Colombia) presents a severe case of overflowing streams, rivers, and creeks that cross or border the city, causing flooding in the urban area, which brings material damage and even, in some situations, human losses; this problem is caused by various factors such as its location near the tributary of the Magdalena River and the sea (2), as well as

the low topographic slope that is around 5% (3). In addition, social and sanitary problems associated with garbage and poor planning cause the rainwater system to collapse quickly (4); the latter makes it a pivotal point to deal with emergencies during the rainy season (2). One way to mitigate the adverse effects of these climatic events is to design and create an early warning model that allows monitoring and alerting affected communities to reduce the impact of overflows, as detailed in (5). Currently, there are several methods of warning and tracking. For example, in Barranquilla, an early warning system was created using sensors and updating the information in a web application (4). This type of system has also been added to supply the system with solar energy (5). Currently, there are several warning and monitoring methods. In Barranquilla, an early warning system was created using sensors and updating the information in a web application (4); improvements have also been added to this system, seeking to supply the system with solar energy (5). Also, for the case study in Barranquilla, hydrological and hydraulic models have been developed to predict the areas where the flow will be high and fast to issue early warnings to the community (6).

On the other hand, in Barranquilla, monitoring of atmospheric phenomena has been implemented, making use of radars that generate alerts before an imminent storm or rain (7); as for data collected from social networks, in Japan, a system was created, which obtains information from Twitter and the processing of this information generates Tsunami alerts (8), likewise in the departmental risk plan of the region of Atlántico (Colombia), this risk scenario has been identified, as well as its causes and antecedents (9). Therefore, it is critical to find an additional solution to those proposed so far, where a model with input information from social networks will be advantageous and increasingly necessary, as this type of information stands out for being fast and updated. Additionally, it provides active monitoring that will generate alerts to the community. Given the above, this article will show in the first section the proposed flow for the model with the design of the system model structure, where the interaction between API, the database, and the processing of the information that generates the stream event prediction, in the second section the methodology used, which includes the data cleaning process, a previous analysis in which the models that best fit the behavior of the data will be reviewed. The third section describes the information processing and training of the selected algorithm. Finally, the results of this research process and the discussion and conclusions will be presented.

2. Methodology

2.1. Model structure

For the model design and structure, a flow verification was initially established, where the precipitation percentage was validated through a meteorological API. If the required percentage was met, the proposed model began by collecting information from the API of the social network X (formerly known as *Twitter*), as described in Figure. 1. The collected data were stored in a specific database and then processed using the trained algorithm selected for the classification. The database was then updated with each classification result (Figure. 2). In cases where the text contained a location (address), this information was extracted to update the database; otherwise, a default address was assigned. Afterwards, the geocoding process began, yielding coordinates (latitude and longitude) and that were

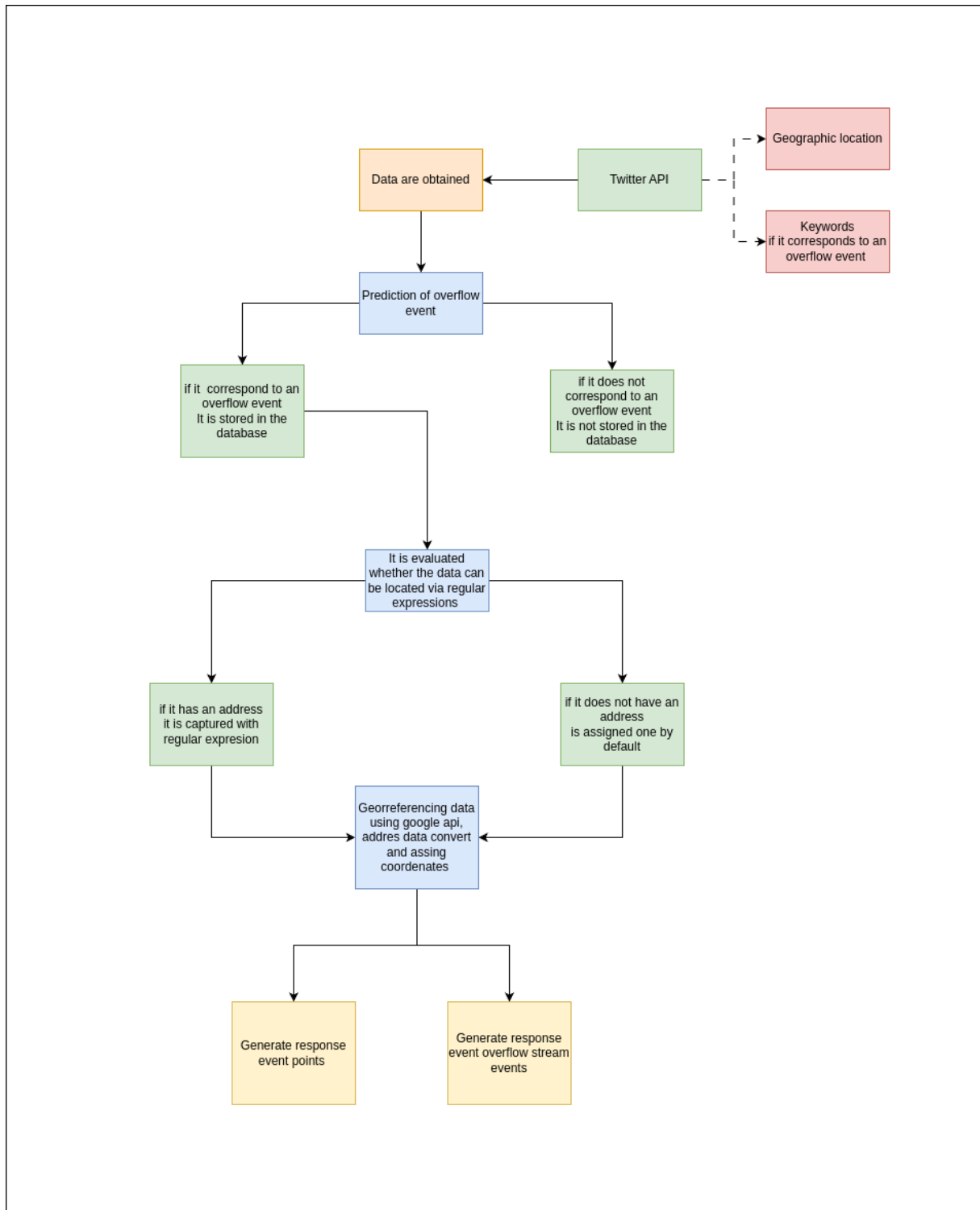


Figure 1. Model flow

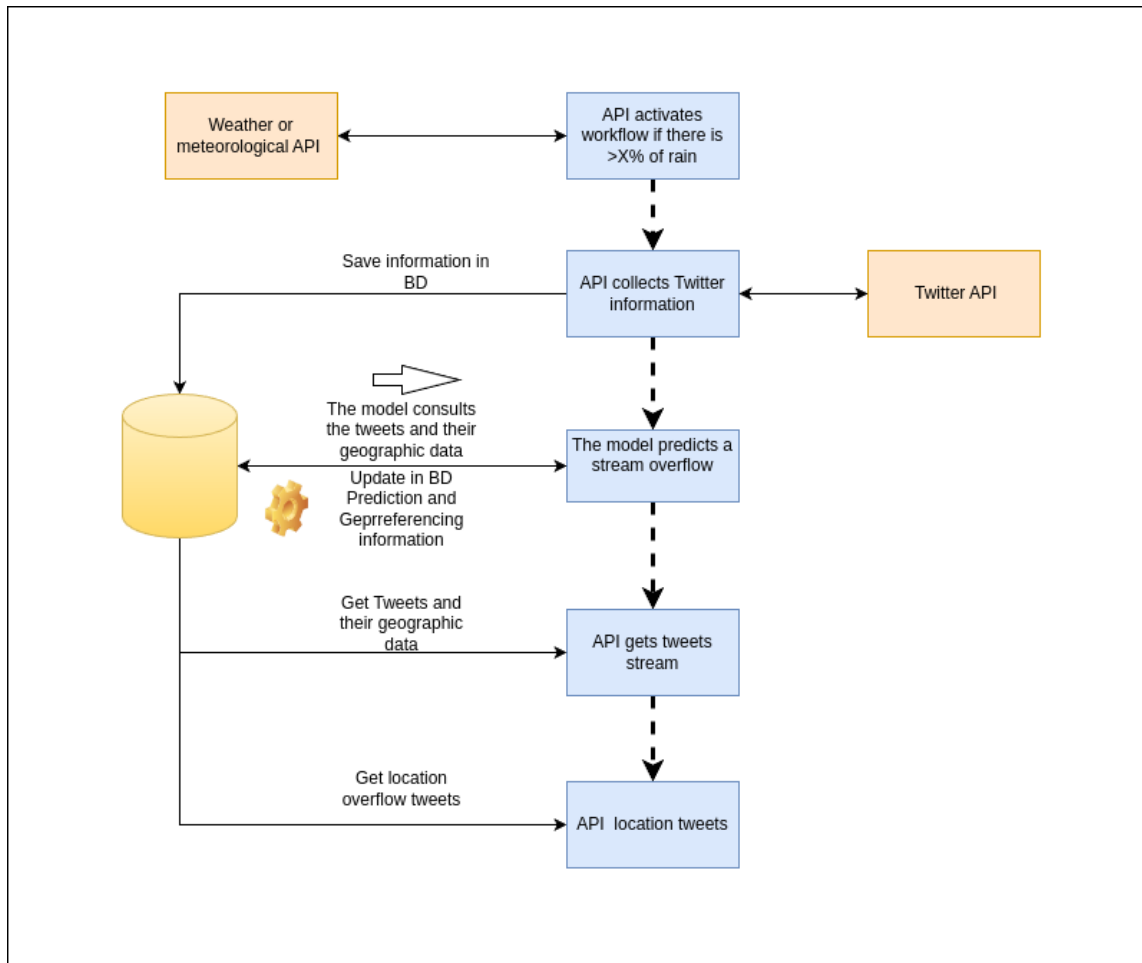


Figure 2. Processing flow

assigned to the data. The next step in the proposed model was to identify whether the recorded events occurred in stream areas. To this effect, it was necessary to make a spatial crossing using a geoprocess, which sought to intercept the location of the events detected as streams vs. the polygons where they circulate. A record of streams was established as roads in Barranquilla (Colombia). The result of the geoprocess was the frequency of events in each stream polygon and the location of each river stream event. This information was displayed on a map (web map) along with the record of the events found in order to generate alerts to the population in risk areas.

2.2. Data collection

The information needed to feed the database was obtained by creating a research account in the X social network (formerly known as *Twitter*) (10). Initially, a search was performed using the X API. The search was filtered by keywords, which were selected based on information from publications such as news and informative posts on the Internet. The words used to perform the search were the following: (Arroyo, Emergencias, Arroyos Barranquilla, Arroyos en la calle, Arroyo en la carrera, creciente, reporte lluvias, calle y carrera). in a time window from 2006 to December 2022. Once the best keywords were

selected, a filter by location was added, taking the city of Barranquilla as the center and adding a 10 km radius to generate the coverage area.

2.3. Data cleaning

Data cleaning is a process that consists of correcting and removing incorrect or duplicated information through computerized methods, as indicated in (11). In this case, the collected data were stored in a SQL database, for which the database engine "Postgresql" was used, and the collected information was structured in database tables. For the data cleaning process, the Python programming language was used, as well as the "Spacy," "Numpy," and "Skylearn" libraries, which contain functionalities that allow connecting the programming language with the database, obtaining each data stored in the database and performing the data cleaning. In the data cleaning process, the methodology mentioned in (12) and (13) was followed where the quality of the data was evaluated. Then the methodology was replicated, where a grammatical analysis of the texts contained in each data was made, finding data with special characters, links, and emojis, after which the data transformation continued, where special characters, emojis, and links were removed using the libraries, leaving the data normalized. Finally, duplicated data found in the database was removed. The debugging performed on the database made it possible to give consistency to the information, thus reducing possible errors generated at the training time.

2.4. Classifying the collected data

Each tweet captured in the search was labeled in three categories, set as a classification criteria variable as follows: event will contain Y (if stream) or N (not stream) data and refers to whether the tweet indicates risk of a stream event, the sarcasm column which will contain Y values (if sarcasm) or N values (not sarcasm) and refers to phrases with a mocking tone that seek to say the opposite, the location column which will contain Y values (if it has an address) or N values (no address) and refers to the geographic location detected in the tweet. For manual classification, each tweet is discriminated against by a user in charge of labeling each data in its corresponding categories. For this classification, the tweet's semantics and context were taken as criteria. This was done manually, considering that it is a supervised process and given that the initial training depends on human interaction, it may contain bias according to the criteria of the person performing the initial classification.

2.5. Information processing

2.5.1. Exploratory data analysis and preprocessing

The exploratory analysis consisted of conducting a series of initial studies and tests necessary to obtain basic approximations to data processing (14) using the information previously stored in the database, which had been previously cleaned and normalized. As in data cleaning, the exploratory analysis was carried out using the Python programming language and the Numpy (15), Scikit learn (16), Spacy (17), and Nltk (18) libraries. Then, we proceeded to establish a count of the number of times that words are repeated (frequency) in the database; this count is essential to determine the words with the highest frequency in the information collected. Another critical step is to filter the collected data by the

one containing the event label equivalent to ("Y"); in this way, the words with the highest frequency during a stream event were found. Additionally, the number per word found helps make histograms and helps in classification to account for data labeled as events, sarcasm, and location.

Additionally, in the exploratory analysis process, the aim was to generate graphs and statistics that would allow the behavior of each variable to be elucidated (14), thus obtaining the regression models with the best behavior in the trend and behavior of the data (19). Given the above, the pre-trained algorithm "es_core_news_lg" from the Spacy library (17) was used to deliver the new algorithm. It should be noted that the pre-trained algorithm has an accuracy of 100% in tokenization, 99% in part of speech, and 98% in morphological analysis (17). Consequently, we proceeded to vectorize the data from the tokenized data to process vectors and not words. We performed mathematical operations on the vectors and regression, comparing the different statistical models and their behavior. We verified and chose the statistical models with an R2 closer to 1 and a lower Root Mean Squared Error (RMSE).

2.5.2. Training

For training, the information was standardized by changing the values of the labels for each data by "1" and "0", where "1" corresponds to "Y" (Yes) and "0" to "N" (No). The information obtained was exported in CSV format to train the cleaned data. This copy was made so as not to manipulate the information initially collected and to make it easier to read the corpus file (input data for processing). Taking into account the data structure and its pre-processing shown above, we proceeded to generate a matrix from the corpus; we used the embedding method (vectorization method), which consists of converting the words or sentences (linguistic units) into vectors, for this we used the pre-trained algorithm es_core_news_lg from the Spacy library (18), which would generate for the entire corpus an array of vectors equivalent to a matrix. Given the vectorized data set, this array was split using the Split method, leaving 70% of the total vectorized data for training and 30% for testing.

After training the algorithms that showed the best behavior in the regressions performed for 1, 2, and 3 variables (event, sarcasm, and location), the algorithm was trained with the information previously divided, classified, and vectorized, obtaining as a result a trained classification machine learning algorithm, which was saved in PKL format and loaded in Python. The different tests were conducted using 30% of the divided data, while the training was tested using the Scikit learn library (16).

3. Results

According to the information collected, it was possible to obtain 63259 data. In Table I it is possible to see the data found per year and per keyword, where an exponential increase in the amount of information related to streams is observed, it is also possible to find that words such as "stream" or "rain" have a greater number of coincidences with respect to the other keywords.

For data cleaning, it was found that the initial search criteria (streams, flood, and overflow) obtained similar results to other search criteria previously applied, causing duplication of information in the

Table I. Tweets by year

Year	Stream on the avenue	Stream on the street	Streams	Flood	Stream emergency	Rain	La Felicidad Stream	Stream	Barranquilla Streams	Country Stream	Streams in Barranquilla	40th Avenue
2006	0	0	0	0	0	0	0	0	0	0	0	0
2007	0	2	6	36	0	209	0	999	0	0	0	5
2008	0	1	33	130	0	499	0	500	0	0	0	21
2009	0	43	486	494	0	497	0	492	13	0	7	207
2010	16	73	489	488	7	490	19	496	492	3	364	496
2011	36	493	495	494	14	496	52	495	494	69	492	494
2012	112	489	480	471	26	494	124	484	494	28	496	497
2013	92	496	484	477	55	487	134	490	495	39	495	498
2014	122	484	491	479	35	491	56	478	478	18	494	497
2015	187	480	493	488	62	494	174	450	474	65	475	494
2016	459	496	481	493	117	490	492	491	495	127	491	488
2017	294	475	494	481	68	479	364	477	490	43	490	496
2018	488	457	481	495	208	483	404	435	493	71	495	475
2019	440	480	462	477	493	489	64	488	485	24	486	496
2020	306	493	496	490	252	442	297	496	465	44	465	470
2021	400	500	499	495	101	495	352	496	496	22	496	499
2022	498	497	500	499	45	500	186	499	493	29	494	497
Total	3450	5959	6870	6987	1483	7535	2718	7766	6357	582	6240	6630

Note: The data were retrieved between 2006 and 2022

database, which generated a considerable increase in information cleaning times. Another critical factor in this section was the purification of special characters and stopwords to ensure a text that is possible to analyze and train (corpus). This reduced by about 40 % the volume of information initially obtained, as it went from 63259 tweets collected to 36720 purified. Additionally, it was found that 8600 were related to a stream event in the city of Barranquilla after manual classification. Of these, 1600 had a text that could be associated with an address and used for geo-coding.

After comparing the different models tested, it can be noted that the random forest model has the best data adjustment; after being adjusted, this regression model presents an R-squared of 7899, as shown in Table II. Where the statistics mean absolute error (MAE), mean square error (MSE), root mean square error (RMSE), coefficient of determination or R-squared (R²), root mean log error (RMSLE), and mean absolute percentage error (MAPE) are found. Additionally, it is possible to note that this adjusted model presents an RMSE of 0.1795, as shown in Table II, corroborating that it has the best statistics for the required adjustment.

To confirm the fit, the same analysis was validated, considering the classification variables such as location and sarcasm, finding similarity of results and the behavior of the data; for this purpose, error prediction graphs were also created, as shown in Figure 3, where the adjusted error and the predicted error are denoted, which shows no significant variation.

Table II. Data regression and results by statistical model

Algorithm	MAE	MSE	RMSE	R2	RMSLE	MAPE
Random Forest	0,0489	0.0323	0.1795	0.7899	0.1257	0.1306
Extreme Gradient	0.0706	0.0328	0.1807	0.7867	0.1251	0.1951
Extra Trees	0.0476	0.0363	0.1903	0.7639	0.1330	0.1255
Light Gradient	0.0816	0.0384	0.1959	0.7498	0.1363	0.2221
K Neighbors	0.0693	0.0406	0.2015	0.7356	0.1425	0.1743
Decision Tree	0.0450	0.0450	0.2119	0.7070	0.1469	0.1213
Gradient Boosting	0.1197	0.0543	0.2327	0.6474	0.1581	0.3455
Bayesian Ridge	0.1821	0.0793	0.2815	0.4844	0.1955	0.4836
Least Angle	0.1812	0.0794	0.2817	0.4835	0.1953	0.4878
Rige	0.1812	0.0795	0.2817	0.4834	0.1953	0.4979
AdaBoost	0.2378	0.0996	0.3155	0.3516	0.2385	0.3698
Lasso	0.2998	0.1499	0.3871	0.0247	0.2717	0.7895
Elastic Net	0.2998	0.1499	0.3871	0.0247	0.2717	0.7894
Lasso Least Angle	0.2998	0.1499	0.3871	0.0247	0.2717	0.7895
Linear	0.2994	0.1499	0.3871	0.0246	0.2717	0.7894

Note, after performing the corresponding data regressions, the Random Forest statistical model showed a better performance; its R2 was the closest to 1, implying a better fit.

The most frequently found words coincided in most cases with the initial search criteria, which improved the rate of search and information acquisition during the training of the neural processing algorithm (NPL). Figure 4 presents a Word-Cloud showing the words with the most repetitions in the information collected.

Figure 5 presents a word frequency plot of the social network "X" during a stream event. Ten new algorithms were trained using classification variables set for this model to select the classification algorithm. From the ten tests, it was possible to choose the three best-performing algorithms during the exploratory analysis (KNC, RF, EXTRA). These three algorithms were tested with the classification variables as follows: "event," "event, sarcasm," "event, sarcasm, location." According to the results (Table III), it was found that the classification variable sarcasm has great importance in the training of the model since it allows discriminating ambiguities from two-way phrases in several Latin American areas; the use of sarcasm as a communication tool is used every day.

The "Extra Tree" of the trained algorithms showed a statistically significant performance that performed best in the statistics. The "Extra Tree" obtained up to 93.66% accuracy on one classification variable (event). However, the same algorithm obtained 78.99% of R2 in the regression. When two variables (event, sarcasm) were analyzed, the algorithm obtained an accuracy of 98.66%, and with three variables (event, sarcasm, location) obtained 98.24%. The second-best performing algorithm was the K-Neighbors Classifier, which obtained 94% accuracy with one variable (event), while the regression

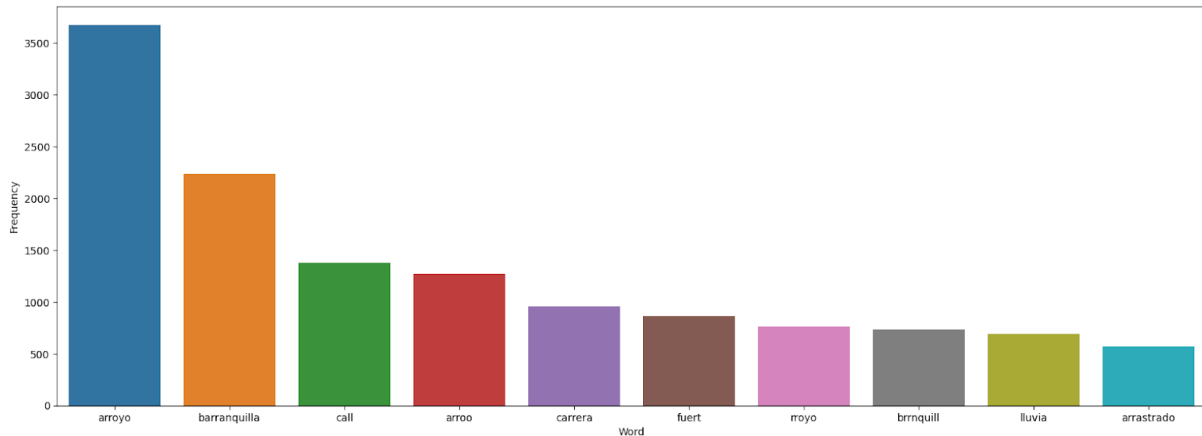


Figure 5. Frequency and intensity of word use during a stream (note: there are words that predominantly appear during a stream event, such as stream, Barranquilla, and rainfall report). The units shown above quantity *vs.* item

Table III. Results of testing in the training algorithms

Algorithm	Fitted precision	Stream	Sarcasm	Location	Fails
KNC	0.9479	X	X	–	88
KNC	0.9391	X	X	X	92
KNC	0.9400	X	–	–	94
RF	0.9822	X	X	–	97
RF	0.9763	X	X	X	104
RF	0.9771	X	–	–	102
EXTRA	0.9366	X	–	–	125
EXTRA	0.9866	X	X	–	105
EXTRA	0.9400	X	X	X	125

stream and the location of each tweet issued that is related to stream overflows, finding that the proposed model works in the best case with an error rate of 0.22, generating an alert to the population according to the location and proximity of these events as shown in Figure 6.

4. Discussion

An early warning model was built from information on stream overflow events in the city of Barranquilla, Colombia, obtained from the social network "X." To improve information filtering, a filter by coverage area was added, which allowed obtaining information only from the required area. This search method immensely helped to delimit the collection of required information and

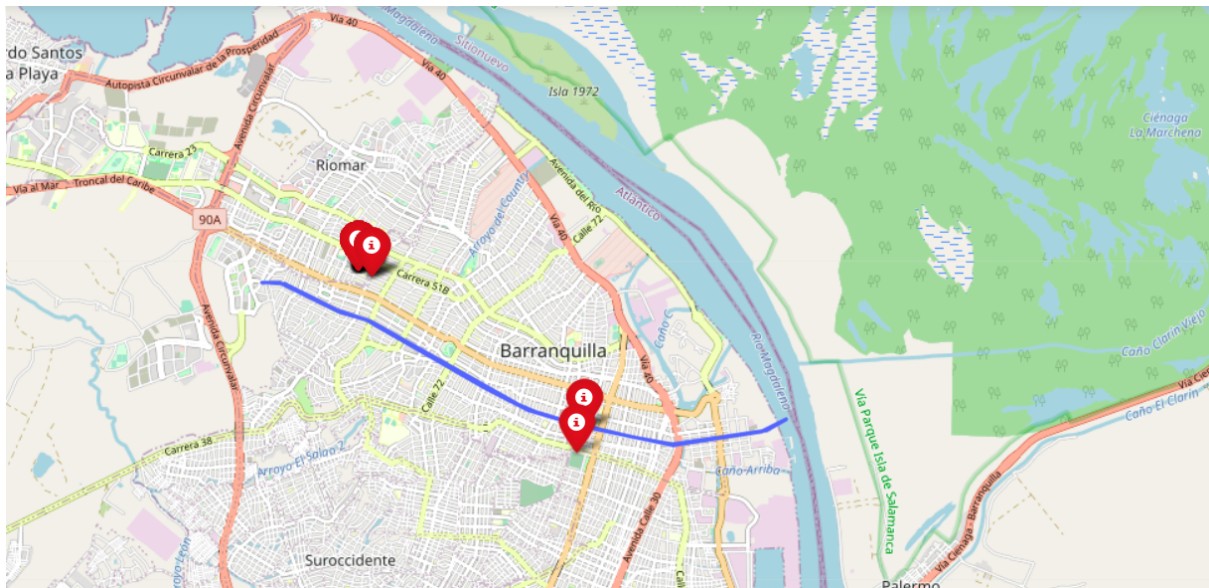


Figure 6. Web map generated by the early warning model for stream overflow events in Barranquilla based on community data. The map shows an example of identified overflow events, as they match the location associated with the analyzed Twitter posts

reduce the data that generated noise in the data cleaning and subsequent training. Therefore, it is possible to consider data cleaning as the totality of operations performed on the data to eliminate anomalies and obtain an accurate and unique representation (20). One of the major drawbacks was the high degree of duplicity in the information, since when searching by area of coverage and keywords, the information contained repeated results; this significantly increased the amount of information in the database. In the same sense, before training the algorithm, it was necessary to label the data obtained; this classification was done manually since the classification criteria can be subjective.

Some of the most representative diagrams obtained during the exploratory analysis were the bar diagram, scatter, and intensity plots. These allowed during the exploratory study to determine the behavior of each variable (14) and from the graphs and statistics obtained to generate initial hypotheses of the behavior of each of the tested algorithms, unlike statistical inference where the hypotheses are born preliminarily and are subsequently challenged and tested from the confirmatory tests (21). Other exploratory analysis techniques also allow for comparing the data distribution by applying a statistical model. Thus, it is possible to determine which statistical model best fits the trend and behavior of the data (19). In this work, it was possible to obtain different regression graphs and models that allowed for the analysis of the behavior of the data and its trend; likewise, these results allowed for the identification of the models with the best behavior. However, this preprocessing required a pre-trained algorithm, which had to be focused on the Spanish language; for this study, an algorithm trained with news in Spanish was used (17).

During the processing and training of the algorithm using tweets, it is critical to highlight the importance of natural language processing, whose objective is the interaction between the computer

and human language (22). It can also be indicated that natural language processing is the ability of machines to process information communicated in human language (23), i.e., natural language processing aims to analyze, understand, and generate language that humans use naturally (24). Thus, text processing seeks to detect and write rules that form structural patterns and then find those patterns in linguistic units such as letters, words, and sentences (22) to be embedded in vectors (24). Additionally, natural language processing has different levels of complexity; each of these represents a type of analysis to be performed to extract specific information; among these levels, it is possible to find morphological, lexical, syntactic, semantic, and pragmatic (25–27).

The morphological level is responsible for analyzing the composition of words, the linguistic level is responsible for establishing the individual meaning of each word, the syntactic level is responsible for investigating the function of each word within the sentence, the semantic level is responsible for establishing the meaning of the sentence from the interaction of words, and the pragmatic level is responsible for analyzing the comprehension of a text (25). The techniques immersed in natural language processing are sentence detection, word segmentation, and discrimination, grammatical tagging, also known as Part of Speech (POS), morphological segmentation, and stopword elimination (28).

Among the best-known applications in the field of natural language processing are content classification and summarization, automatic contextual extraction, sentiment analysis, speech-to-text conversation, and, finally, machine translation (24, 29, 30). The advances in data augmentation and its usefulness in applying different problems (31). This system is also used as a text classifier using various algorithms (32), where the main challenge is to discriminate from ambiguities (33), obtaining. As a result, there is a binary answer in the classification.

Natural language processing methods are usually used to classify texts. These methods allow for identifying the data's tendencies, so they are widely used for classification depending on the probability of similarity to a target variable. Among the most used methods to classify texts are K-Neighbor, Logistic Regression, Naive Bayes, SVM, Random Forest (34), and, in some cases, regressions such as Linear regression (35). Algorithms such as Extra Tree and random forest are among the family of random forest methods. This type of algorithm combines the randomness of the subspace and bagging, which trains multiple decision trees slightly different from the data set (36). On the other hand, the K-Neighbor Algorithm is a statistical model that uses the Euclidean distance to determine which data is closer to the data to be classified. Depending on the count found, the target data will be classified accordingly (37). For the case of this study, the K-Neighbor classification algorithm (KNC) was chosen since it presented the best behavior when analyzing and sorting actual data. During the last years, several research has focused on this field, which has allowed rapid progress in the subject to the point that nowadays, it is possible to find natural language applications in cellphone and virtual assistants or different types of call centers (33, 38, 39).

Finally, this study applied training techniques such as tokenization, which seeks to divide sentences into semantic units; vectorization, which aims to convert the union of semantic units into vectors and, in

some cases, the identification of the part of speech (POS) (40), which seeks to identify the function of the previously tokenized word in the context, i.e., whether it is a verb, adverb, adjective, connector and thus establish the weight within the sentence. This type of preprocessing helps to condition the algorithm's training so that the necessary coincidences and structures are found to predict the classification variables later. Having the information vectorized, mathematical operations are applied to the vectors, seeking as objective the vector similarity between the target variable and the text to be classified. Likewise, although the algorithms with the best statistics were chosen, it was not the ones that showed the best adjustment when evaluating the selected one since overfitting in the models is denoted, which caused unreliable classifications. Given the above, the k-neighbor algorithm was the one that presented the best classification results.

5. Conclusions

The proposed model allowed the collection of information relevant to the case study, which allowed the detection of possible events from the collected data and georeferencing them, whereby by changing the different prioritized filters (location and keywords), it is possible to adapt to a problem unique to the region. On the other hand, the exploratory analysis from the regression allowed us to correctly determine a group of algorithms with better behavior, as well as the words with higher frequency in a stream event. An essential factor to mention was the impact of the column or variable "sarcasm," as it obtained a significant weight in the exploratory analysis and training since, at the time of vectorizing the data, the selected text favors the resolution of ambiguities in the tweets collected and filtered due to the semantics of each text analyzed, which is strongly altered by the popular vocabulary of the study area.

6. Acknowledgements

The authors would like to thank the Twitter team for providing a Twitter academic license, which was useful for data search and algorithm training. These processes involved long time intervals, but the collected information constituted a sufficient and necessary input for executing this research project.

7. CRediT author statement

All authors contributed equally to the research.

References

- [1] Riesgo por Inundación - IDIGER». Accedido: 25 de febrero de 2024. [Online]. Available: <https://www.idiger.gov.co/rinundacion> ↑2
- [2] H. D. Van Strahlen Bartel, "Estudio de la problemática de los arroyos urbanos de la cuenca El Rebolo (Barranquilla, Colombia) y propuesta de soluciones," Master's thesis, Universitat Politècnica de València, 2017. [Online]. Available: <http://hdl.handle.net/10251/90068> ↑2,3

- [3] H. Ávila, "Perspectiva del manejo del drenaje pluvial frente al cambio climático-caso de estudio: ciudad de Barranquilla, Colombia," *Rev. Ing.*, vol. 36, pp. 54-59, 2012. <http://www.scielo.org.co/pdf/ring/n36/n36a11.pdf> ↑3
- [4] J. A. Sepúlveda Ojeda, "Aplicación web para la visualización de sensores del sistema de alertas tempranas de los arroyos de Barranquilla-Colombia," *Rev. Espacios*, vol. 38, no. 47, p. 17, 2017. <http://hdl.handle.net/11323/2024> ↑3
- [5] L. J. Pérez Flórez and J. S. Hernández Miranda, "Diseño del modelo económico energético para un sistema de alerta temprana (MEESAT) para los arroyos de Barranquilla," undergraduate thesis, Universidad de la Costa, 2015. <http://hdl.handle.net/11323/4899> ↑3
- [6] M. Acosta-Coll, F. Ballester-Merelo, and M. Martínez-Peiró, "Early warning system for detection of urban pluvial flooding hazard levels in an ungauged basin," *Nat. Hazards*, vol. 92, pp. 1237-1265, 2018. <https://doi.org/10.1007/s11069-018-3249-4> ↑3
- [7] M. A. Coll, "Sistemas de alerta temprana (SAT) para la reducción del riesgo de inundaciones súbitas y fenómenos atmosféricos en el área metropolitana de Barranquilla," *Sci. Tech.*, vol. 18, no. 2, pp. 303-308, 2013. <https://revistas.utp.edu.co/index.php/revistaciencia/article/view/8661/5411> ↑3
- [8] A. Chatfield and U. Brajawidagda, "Twitter tsunami early warning network: A social network analysis of Twitter information flows," in *23rd Australasian Conf. Info. Syst.*, 2012, pp. 1-10. <https://core.ac.uk/download/pdf/301388984.pdf> ↑3
- [9] Gobernacion del Atlántico, "Plan departamental de gestión del riesgo Atlántico (Colombia)," 2021. [Online]. Available: <http://repositorio.gestiondelriesgo.gov.co/handle/20.500.11762/392?locale-attribute=es> ↑3
- [10] X, "Use Cases, Tutorials, & Documentation," X Developer Platform. <https://developer.twitter.com/en> (accessed July 8, 2023). ↑5
- [11] Y. Valdés Hernández and D. Marmol Lacal, "DBAnalyzer 2.0, sistema para analizar bases de datos libre," undergraduate thesis, Universidad de las Ciencias Informáticas, 2008. https://repositorio.uci.cu/jspui/bitstream/ident/TD_1287_08/1/TD_1287_08.pdf ↑6
- [12] E. Estoque Cabrera, L. Baró Galán, and M. E. Escobar Pompa, "Implementación de algoritmos para la limpieza de datos," undergraduate thesis, Universidad de las Ciencias Informáticas, 2015. <https://repositorio.uci.cu/jspui/handle/ident/8774> ↑6
- [13] I. Zeroual and A. Lakhouaja, "Data science in light of natural language processing: An overview," *Procedia Comput. Sci.*, vol. 127, pp. 82-91, 2018. <https://doi.org/10.1016/j.procs.2018.01.101> ↑6
- [14] G. D. Buzai and C. A. Baxendale, "Análisis exploratorio de datos espaciales," *Geogr. Sist. Inf. Geográfica*, no. 1, pp. 1-11, 2009. https://ri.unlu.edu.ar/xmlui/bitstream/handle/rediunlu/702/Buzai_An%C3%A1lisis%20Exploratorio%20de%20Datos%20Espaciales.pdf?sequence=1&isAllowed=y ↑6, 7, 12
- [15] «NumPy - documentation». [Online]. Available: <https://numpy.org/> ↑6
- [16] «scikit-learn: machine learning in Python — scikit-learn 1.4.1 documentation». [Online]. Available: <https://scikit-learn.org/stable/> ↑6, 7

- [17] «Spanish · spaCy Models Documentation», Spanish. [Online]. Available: <https://spacy.io/models/es> ↑6,7,12
- [18] «NLTK :: Natural Language Toolkit». Accedido: 25 de febrero de 2024. [En línea]. [Online]. Available: <https://www.nltk.org/> ↑6,7
- [19] Z. Jianqiang and G. Xiaolin, "Comparison research on text pre-processing methods on twitter sentiment analysis," *IEEE Access*, vol. 5, pp. 2870-2879, 2017. <https://doi.org/10.1109/ACCESS.2017.2672677> ↑7,12
- [20] H. Müller and J. C. Freytag, "Problems, methods, and challenges in comprehensive data cleansing", 2005. [Online]. Available: <https://tarjomefa.com/wp-content/uploads/2015/06/3229-English.pdf> ↑12
- [21] P. Carranza and J. Fuentealba, "Una introducción al análisis exploratorio de datos por medio de Google Analytics," *Yupana Rev. Educ. Matemática UNL*, vol. 7, pp. 53-65, 2013. <https://doi.org/10.14409/yu.v1i7.4262> ↑12
- [22] M. M. E. Torres and R. Manjarrés-Betancur, "Asistente virtual académico utilizando tecnologías cognitivas de procesamiento de lenguaje natural," *Rev. Politécnica*, vol. 16, no. 31, pp. 85-96, 2020. <https://doi.org/10.33571/rpolitec.v16n31a7> ↑13
- [23] A. Gelbukh, "Procesamiento de lenguaje natural y sus aplicaciones," *Komputer Sapiens*, vol. 1, pp. 6-11, 2010. <https://www.gelbukh.com/CV/Publications/2010/Procesamiento%20de%20lenguaje%20natural%20y%20sus%20aplicaciones.pdf> ↑13
- [24] L. Deng, "Deep learning: from speech recognition to language and multimodal processing," *APSIPA Trans. Signal Inf. Process.*, vol. 5, no. 1, Jan. 2016. <https://doi.org/10.1017/ATSIP.2015.22> ↑13
- [25] F. Ramos and J. Vélez, "Integración de técnicas de procesamiento de lenguaje natural a través de servicios web," undergraduate thesis, Universidad Nacional del Centro de la Provincia de Buenos Aires, 2016. [Online]. Available: <https://www.ridaa.unicen.edu.ar/handle/123456789/644> ↑13
- [26] P. Johri, S. K. Khatri, A. T. Al-Taani, M. Sabharwal, S. Suvanov, and A. Kumar, "Natural language processing: History, evolution, application, and future work," in *Proc. 3rd Int. Conf. Computing Informatics Networks: ICCIN 2020*, 2021, pp. 365-375. http://dx.doi.org/10.1007/978-981-15-9712-1_31 ↑13
- [27] M. Maldonado, D. Alulema, D. Morocho, and M. Proano, "System for monitoring natural disasters using natural language processing in the social network Twitter," in *2016 IEEE Int. Carnahan Conf. Sec. Tech. (ICCST)*, 2016, pp. 1-6. <https://doi.org/10.1109/CCST.2016.7815686> ↑13
- [28] D. Moreira et al., "Análisis del estado actual de procesamiento de lenguaje natural," *Rev. Ibérica Sist. Tecnol. Informação*, no. E42, pp. 126-136, 2021. <https://dialnet.unirioja.es/servlet/articulo?codigo=8624557> ↑13
- [29] A. Gutiérrez Domínguez, "Aplicación de técnicas de procesamiento de lenguaje natural (NLP) en Twitter para la evaluación de políticas agrarias y del medio rural," Master's thesis, 2022. [Online]. Available: <http://hdl.handle.net/10251/186767> ↑13

- [30] Z. Zong and C. Hong, "On application of natural language processing in machine translation," in *2018 3rd Int. Conf. Mech. Control Comp. Eng. (ICMCCE)*, Sep. 2018, pp. 506-510. <https://doi.org/10.1109/ICMCCE.2018.00112> ↑13
- [31] B. Li, Y. Hou, and W. Che, "Data augmentation approaches in natural language processing: A survey," *AI Open*, vol. 3, pp. 71-90, Jan. 2022, <https://doi.org/10.1016/j.aiopen.2022.03.001> ↑13
- [32] M. B. Hernández and J. M. Gómez, "Aplicaciones de procesamiento de lenguaje natural," *Rev. Politécnica*, vol. 32, 2013. https://revistapolitecnica.epn.edu.ec/ojs2/index.php/revista_politecnica2/article/view/32 ↑13
- [33] A. Yadav, A. Patel, and M. Shah, "A comprehensive review on resolving ambiguities in natural language processing," *AI Open*, vol. 2, pp. 85-92, Jan. 2021. <https://doi.org/10.1016/j.aiopen.2021.05.001> ↑13
- [34] A. A. Turdjai y K. Mutijarsa, «Simulation of marketplace customer satisfaction analysis based on machine learning algorithms», en *2016 International Seminar on Application for Technology of Information and Communication (ISEMANTIC)*, ago. 2016, pp. 157-162. <https://doi.org/10.1109/ISEMANTIC.2016.7873830> ↑13
- [35] Y. Takefuji and K. Shoji, "Effectiveness of ensemble machine learning over the conventional multivariable linear regression models". [Online]. Available: http://202.240.109.17/publications/pdf/reg_vs_ml.pdf ↑13
- [36] Y. Al Amrani, M. Lazaar, and K. E. El Kadiri, "Random forest and support vector machine based hybrid approach to sentiment analysis," *Procedia Comput. Sci.*, vol. 127, pp. 511-520, 2018. <https://doi.org/10.1016/j.procs.2018.01.150> ↑13
- [37] D. S. Osorio, J. J. M. Escobar, L. Chanona-Hernández, G. Sidorov, and C. J. Núñez-Prado, "Clasificación bi-clase de canciones infantiles aplicando inteligencia artificial y procesamiento de lenguaje natural," *Res. Comput. Sci.*, vol. 151, no. 5, pp. 31-38, 2022. ISSN 1870-4069 ↑13
- [38] S. Quarteroni, «Natural language processing for industry: ELCA's experience», *Inform.-Spektrum*, vol. 41, n.o 2, pp. 105-112, 2018. GF <https://doi.org/10.1007/s00287-018-1094-1> ↑13
- [39] N. Kaur, V. Pushe, and R. Kaur, "Natural language processing interface for synonym," *Int. J. Comput. Sci. Mob. Comput.*, vol. 3, n.o 7, pp. 638-642, 2014. ISSN 2320-088X ↑13
- [40] E. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: The good the bad and the omg!," in *Proc. Int. AAAI Conf. Web Soc. Media*, 2011, pp. 538-541. <https://doi.org/10.1609/icwsm.v5i1.14185> ↑14

Iván Andrés Felipe Serna-Galeano

Cadastral and geodetic engineer from Universidad Distrital Francisco José de Caldas and Master's student in Information and Communications Sciences at the Department of Engineering of Universidad Distrital Francisco José de Caldas in Bogotá, Colombia.

Email: iasernag@udistrital.edu.co

Ernesto Gómez-Vargas

Full professor at the Department of Engineering of Universidad Distrital Francisco José de Caldas in Bogotá, Colombia. Electronics engineer, specialist in Mobile Telecommunications, and master in Teleinformatics from Universidad Distrital Francisco José de Caldas; PhD in Engineering from Pontificia Universidad Javeriana, Bogotá.

Email: egomez@udistrital.edu.co

Julián Rolando Camargo-López

Full professor at the Department of Engineering of Universidad Distrital Francisco José de Caldas in Bogotá, Colombia. Electronics engineer from Universidad Distrital Francisco José de Caldas, specialist in Design and Construction of Telematic Solutions from Universidad Autónoma de Colombia, and master in Information and Communications Sciences from Universidad Distrital Francisco José de Caldas.

Email: jcamargo@udistrital.edu.co

