Model Proposal for the Detection of False Information About COVID-19 Using Machine Learning and Natural Language Processing Techniques

Yair Andrey Salinas Bolaños[1] , Wilfredo Ticona[1,2]
[1]Universidad ESAN, Lima, Perú, 17200648@esan.edu.pe
[2] Universidad Tecnológica del Perú, Lima, Perú, wmamani@esan.edu.pe

Abstract — One of the main problems that arose as a result of this health emergency was the circulation of false information on COVID-19. Therefore, the study carried out aimed to find the best classifier of false information on COVID-19 in the Peruvian context. For this, 2022 information records related to COVID-19 were collected through web scraping of websites, Facebook and Twitter, which were manually labeled as True or False and then validated. Natural Language Processing techniques such as Bag of Words, TF-IDF, Word2Vec and FastText were used for feature extraction. Finally, different Machine Learning model were developed using KNN, Decision Tree, Naive Bayes, SVM, Logistic Regression and MLP. The results were evaluated according to the Accuracy, Precision, Recall and F1-score metrics. The best model resulted from the combination of the SVM algorithm (C (0.5), gamma (1) and kernel (rbf)) with TF - IDF of dimension 300 and n-grams from 1 to 2, whose metrics were superior to the others with 87.41% Accuracy, 88.63% Precision, 87.39% Recall and 88% F1-score.

Keywords - False information, COVID-19, Machine Learning, Natural Language Processing, Accuracy.

Model Proposal for the Detection of False Information About COVID-19 Using Machine Learning and Natural Language Processing Techniques

Yair Andrey Salinas Bolaños[1] , Wilfredo Ticona[1,2]
[1]Universidad ESAN, Lima, Perú, 17200648@esan.edu.pe
[2] Universidad Tecnológica del Perú, Lima, Perú, wmamani@esan.edu.pe

Abstract — One of the main problems that arose as a result of this health emergency was the circulation of false information on COVID-19. Therefore, the study carried out aimed to find the best classifier of false information on COVID-19 in the Peruvian context. For this, 2022 information records related to COVID-19 were collected through web scraping of websites, Facebook and Twitter, which were manually labeled as True or False and then validated. Natural Language Processing techniques such as Bag of Words, TF-IDF, Word2Vec and FastText were used for feature extraction. Finally, different Machine Learning model were developed using KNN, Decision Tree, Naive Bayes, SVM, Logistic Regression and MLP. The results were evaluated according to the Accuracy, Precision, Recall and F1-score metrics. The best model resulted from the combination of the SVM algorithm (C (0.5), gamma (1) and kernel (rbf)) with TF - IDF of dimension 300 and n-grams from 1 to 2, whose metrics were superior to the others with 87.41% Accuracy, 88.63% Precision, 87.39% Recall and 88% F1-score.

Keywords - False information, COVID-19, Machine Learning, Natural Language Processing, Accuracy.

I. INTRODUCTION

In the context of the pandemic in Peru, misleading information shared about the closure of public places, vaccinations, and death statistics have fueled panic buying of groceries, disinfectants, masks, and paper products, which in turn it created shortages that interrupted the supply string. Such misinformation is commonly disseminated through the Facebook and Twitter platforms, generating fear, rejection, and distrust in public officials, which is why it is common for the Ministry of Health (MINSA) to publish publications to deny false information about COVID-19, circulating in the country. However, they cannot deny all of them as there are many posts and comments sharing fake content, making it difficult to manually detect each one. Likewise, the ease of access to large volumes of data online, whose veracity is often unknown, has raised doubts about the quality and reliability of the information available on health issues.

The study carried out by Nieves-Cuervo *et al.* [1] details the behavior of the propagation of false information in the context of mortality from COVID-19 in Peru, where it was found that said country presented the highest percentage regarding the inability to recognize false news (79.0%) and that it was the second most trusted in the content of social networks (46.0%) leading to a higher mortality rate from COVID-19 above countries such as Chile, Colombia, Brazil, Argentina and Mexico. Similarly, the survey carried out by researchers from the Pacific University [2] details that Peruvians are more likely to avoid COVID-19 vaccines because they do not trust them, since they believe that these vaccines are harmful to your health, modify your DNA or contain some kind of chip.

II. RELATED WORKS

In the work of Bojjireddy *et al.* [3], the authors used 3 fake news datasets, the first of which was created through web scraping and the other two were combinations of Kaggle and Fakeorreal sources. Count Vectorizer and Tf-Idf techniques were applied. for the vectorization of the data and the false information classification model was built with the Multinomial Naïve Bayes, Support Vector Machine, Multilayer Perceptron, Decision Tree, Random Forest and Boosting Gradient techniques. The best model resulted from the combination of the Multilayer Perceptron and the Count Vectorizer, obtaining an accuracy of 0.96.

Similarly, Hayawi *et al.* [4] applied a novel framework for the detection of COVID-19 vaccine misinformation. 15073 COVID-19 vaccine tweets were collected, which, through reliable sources, were annotated as misinformation tweets or general tweets. This annotation was later validated with medical experts. For the development of the model, the XGBoost, LSTM, and BERT techniques were used, in combination with the Tf-Idf and GloVe vectorization techniques. The results showed that the best model was obtained with BERT, since it provided an accuracy of 0.98, F1-score of 0.98, precision of 0.97 and recall of 0.98.

In the work carried out by Elhadad *et al.* [5] a model was developed to detect misleading information related to COVID-19, based on reliable sources such as WHO, UNICEF and ONU. Data cleaning, tag standardization and binarization, data integration, and feature engineering were performed. For the development the algorithms Decision Tree, kNN, Logistic Regression, LSVM, Multinomial Naïve Bayes, Bernoulli Naïve Bayes, Perceptron, Neural Network, Ensemble Random Forest and XGBoost were considered; together with the TF, Tf-Idf and Word embeddings feature extraction techniques. The best model resulted from the combination between Neural Network and TF, since it obtained an accuracy of 0.99.

Finally, in the work of Mahlous and Al-Laith. [6], 37000 Arabic tweets were worked with, which were manually and automatically annotated as disinformation or not. They were cleaned (elimination of mentions, links, hashtags, punctuation, stop words and non-Arabic words). For the

development of the model, the Naive Bayes, Logistic Regression, SVM, Multilayer Perceptron, Random Forest and XGBoost techniques were used; and, for the extraction of characteristics, the Count Vetorizer and Tf-Idf techniques were used. The best model was obtained with Logistic Regression and Count Vectorizer, since its F1-score was 0.93.

As evidenced, there are various related works that seek to deal with this problem, for which Machine Learning and Natural Language Processing techniques are of great help, since they facilitate the automation of the recognition of true or false information about COVID-19, analyzing patterns and identifying characteristics of each record. This article seeks to develop a robust classifier model applied to a new own dataset prepared for the Peruvian context, considering the techniques mentioned above in combination with the methodologies used.

III. METHODOLOGY

The proposed methodology was based on [3], [4] and [7]. It consists of five phases: Dataset construction, Preprocessing, Feature Extraction, Model Implementation and Evaluation. Then, each stage has general activities that are of paramount importance for the development of the research. The methodology used is shown graphically below (Fig. 1):
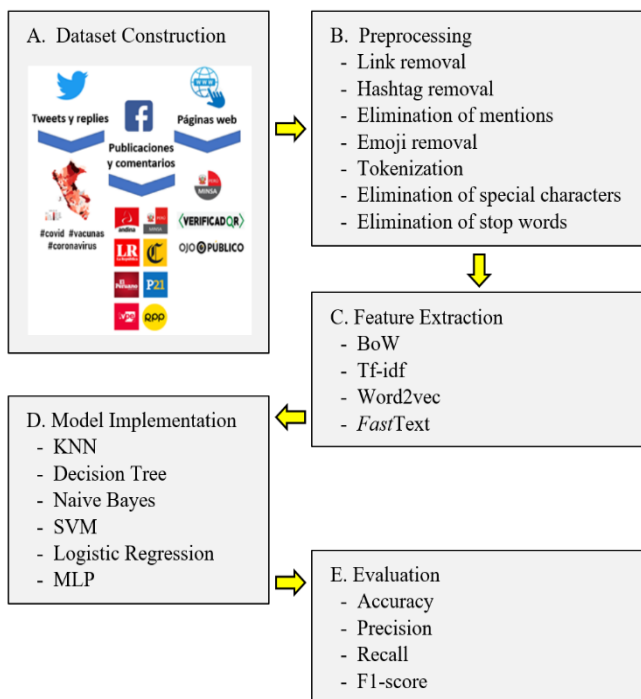

Fig. 1 Proposed Methodology.

A. Dataset Construction

First, the data sources with which to work were chosen, the social networks Facebook and Twitter were chosen, because, according to IPSOS (2021), Facebook was one of the networks that led the frequency of use on social networks in Peru; while Twitter was also taken into account because, although the use of Twitter is not so frequent in the country, it was possible to show that certain public officials used this medium to make their opinions known, regardless of whether they were wrong. Likewise, the website of MINSA and the factcheckers VerificadorLR and OjoPúblico were considered.

Second, scraping was carried out for each of these sources. For Facebook, the facebook-scraper library was used, with which 21510 posts and, in turn, 1282100 comments were collected from January 1st to April 30th, 2022, which were stored in csv files. These data were obtained from the accounts of the most representative broadcast media in the country, such as Perú21, La República, RPP, El Comercio, El Peruano, Andina, TV Perú and MINSA. On the other hand, for the extraction of Twitter, the platform's api (Tweepy) was used, with which 53878 tweets were extracted, for this it was necessary to define keywords such as "covid", "coronavirus" and "vaccines"; the extraction was given for each department of Peru (24) and the Constitutional Province of Callao, so it was necessary to obtain the coordinates of each department through the Geocode.xyz web page; This procedure was carried out on 2 dates: April 25th and 30th, in order to cover various content and avoid repeated content, these were also stored in csv files. Finally, for the scraping of the web pages, the algorithms were developed with the Selenium and BeautifulSoup libraries; since each web page had a unique structure, an analysis of its own was necessary to create the algorithm according to its requirements: 912 news items were obtained from the MINSA website (title, date and link), 133 from Verificador LR (title and link) and 100 from OjoPúblico (title, date and link); which were saved in csv files.

Third, the datasets (csv) were equalized, renaming the names of the columns, and eliminating those that were useless. Then, repeated records and records that were not related to covid-19 were eliminated, for which some keywords were defined (Table I). Likewise, the "Class" column was created with an initial value of None, which would represent that the record is neither True nor False. The result of this was a dataset with 4 columns ("Class", "Content", "Date" and "Link") and 80515 rows.

TABLE I
LIST OF CORONAVIRUS KEYWORDS

#	Keyword	#	Keyword
1	covid	10	oms
2	coronavirus	11	presencial
3	pandemia	12	mascarilla
4	vacuna	13	camas uci
5	sars-cov-2	14	variante
6	ómicron	15	dosis
7	deltacron	16	contagio
8	molecular	17	contagiado
9	pcr	18	cuarentena

Finally, the manual labeling of the records was carried out, for this, each content was compared with each of the reliable sources: WHO, PAHO and MINSA; if the record contradicted any publication from these sources, it was labeled False, otherwise, if it was similar, it would be labeled True, on the other hand, if the information was not published, the original label (*None*) was kept. Sarcastic records and religious tendencies were put aside. The result was a dataset with 2022 labeled records, which were validated with expert personnel from MINSA.

B. Preprocessing

In this phase, the removal of links, hashtags and mentions was carried out using regular expressions, then the emojis were removed with the emoji library. Subsequently, Spanish characters such as the diaeresis and the tilde were eliminated through the unicode library. Next, the content was

normalized (convert to lowercase) and tokenized using nltk. Table II shows the activities of the preprocessing phase.

TABLE II
PREPROCESSING ACTIVITIES

Activities	Description	Tasks
Remove links, hashtags, mentions and emojis.	Elimination of irrelevant content such as links, hashtags, mentions and emojis.	Create regular expression to remove links, hashtags and mentions.
		Identify library to remove emojis.
Clean up the content.	Cleaning of the content and standardizing in the same format in order to better vectorize features.	Eliminate special characters typical of Spanish.
		Convert to lowercase.
		Tokenize each element.
		Remove punctuation marks and numbers.
		Delete stop words.
		Create the corpus.

Once this was done, the non-alphabetic tokens (punctuation marks, numbers and special characters) were removed, in order to work only with words; and, finally, the stop words were removed with the nltk library; Although this bookstore already had some stop words in Spanish, it was necessary to add more, since these were not enough, they went to the COUNTWORDSFREE website to obtain a more complete list of stop words, likewise, they were added manually to the list certain Peruvian jargons and misspelled words typical of the context, such as 'pues', 'pes', 'pe', 'xq', 'xd', 'yara', etc.; the result of this was a list of 626 stop words.

After carrying out everything mentioned, 2 corpus were obtained: the first in form of a string for the BoW and Tf-Idf techniques, and the second, in form of a list of strings for w2v and fastText. Fig. 2 shows the original content and the resulting corpus:

Content
Hospital Villa El Salvador refuerza vacunación…
Hay q ser muy ignorante después de toda la inf…
Toxicidad del grafeno: El arma perfecta para e…
Desde el Cusco, durante el VI Consejo de Minis…
La vacunación contra el #COVID19 en estudiantes…
Content_Preprocessed
hospital villa salvador refuerza vacunacion di…
ignorante informacion ponerse dosis
toxicidad grafeno arma perfecta enfermar matar…
cusco consejo ministros descentralizado minist…
vacunacion estudiantes obligatoria retornar cl…
Content_Preprocessed_Embedding
[hospital, villa, salvador, refuerza, vacunacion…
[ignorante, informacion, ponerse, dosis]
[toxicidad, grafeno, arma, perfecta, enfermar,…
[cusco, consejo, ministros, descentralizado, m…
[vacunacion, estudiantes, obligatoria, retorna…

Fig. 2 Original content and the resulting corpus.

C. Feature Extraction

In this phase, four Natural Language Processing techniques were applied, we worked with the sklearn library for the vectorization of characteristics for the first 2 techniques: CountVectorizer (BoW) and Tf-Idf. 3 vectorizers were created for each technique, each with size 100, 200 and 300; however, for Tf-Idf, a range of n-grams from 1 to 2 was also considered. Table III shows the activities of the feature extraction phase.

TABLE III
FEATURE SELECTION ACTIVITIES

Activities	Description	Tasks
Perform feature extraction with BoW.	Extraction of the characteristics of each record with the BoW technique.	Identify the relevant parameters.
		Create 3 different vectors and apply to the corpus.
Perform feature extraction with Tf-Idf.	Extraction of the characteristics of each record with the Tf-Idf technique.	Identify the relevant parameters.
		Create 3 different vectors and apply to the corpus.
Perform feature extraction with Word2Vec.	Extraction of the characteristics of each record with the Word2Vec technique.	Train the embedding with the general corpus that includes all the records.
		Identify the relevant parameters.
		Create 3 different vectors and apply to the corpus.
Perform feature extraction with FastText.	Extraction of the characteristics of each record with the FastText technique.	Train the embedding with the general corpus that includes all the records.
		Identify the relevant parameters.
		Create 3 different vectors and apply to the corpus.

Tables IV and V show the parameters used in each vectorizer for BoW and Tf-Idf, respectively. The records of the content preprocessed with both techniques were vectorized.

TABLE IV
BAG OF WORDS PARAMETERS

Vectorizers	Parameters
	max_features
cv_1	100
cv_2	200
cv_3	300

TABLE V
TF-IDF PARAMETERS

Vectorizers	Parameters	
	max_features	*ngram_range*
tf_idf_1	100	1,2
tf_idf_2	200	1,2
tf_idf_3	300	1,2

Regarding the following 2 techniques related to embedding models (*w2v and fastText*), as they are pre-trained models, a new training was necessary to give context to the words. To carry out this process, the links of the Facebook posts, the news of the web pages and the publications on Twitter of the original dataset (80515) were taken, the Facebook comments were not considered, since the bad writing in them would complicate the training of the models. Subsequently, the content of each link was extracted using the newspaper3k library; It is worth mentioning that this library was not able to extract the content of some links, so it was necessary to eliminate those empty records; then, records in a language other than Spanish were eliminated, for which the langdetect library was used. Finally, duplicate records were eliminated, obtaining a dataset of 11631 rows with only one column referring to the content.

3

This new dataset went through the same preprocessing as previously explained, where the result was expressed in a training corpus in the form of a list of strings. The Gensim library was used to train both models. Table VI shows the best parameters found after w2v training; while the VII table shows the parameters of the fastText. With these models, the vectorization of the records of the preprocessed corpus was carried out.

TABLE VI
WORD2VEC PARAMETERS

Vectorizers	Parameters			
	size	window	min_count	sg
w2v_1	100	5	5	0
w2v_2	200	5	5	0
w2v_3	300	2	5	1

TABLE VII
FASTTEXT PARAMETERS

Vectorizers	Parameters					
	size	window	min_count	sg	min_n	max_n
fastText_1	100	4	5	0	2	5
fastText_2	200	4	3	0	2	4
fastText_3	300	4	4	0	2	6

D. Model Implementation

For the construction of the model, the coding of the "Class" column was carried out, whose textual values became numerical: *True* (1) and *False* (0), this new coding was joined with the previously vectorized records with the 4 techniques. Next, the data was suffled in order to avoid bias. Then, the partition for training and testing was carried out, considering 80% and 20% of the data, respectively. Finally, the GridSearchCV algorithm was used with cross-validation of 5 for each Machine Learning technique. Table VIII shows the activities of the model building phase.

TABLE VIII
MODEL BUILDING ACTIVITIES

Activities	Description	Tasks
Class coding	Labeling of each vector numerically according to its class	Label each vector with 1 (true) and 0 (false) values
Dataset partition	Partition of the dataset in training (80%) and test (20%); followed by 5-fold cross-validation in training	Partition of the dataset in training (80%) and test (20%)
Use of Machine Learning techniques	Training the models with the training and test data	Create one grid to identify the best parameters of each model
		Build a classification model by machine learning technique
		Obtain the results of the performance of each model in a new data (test)

The following shows the parameters with their values tuned in each algorithm.

1) KNN
    n_neighbors: 3,5,7,9,11,13,15,17,19,21,23,25,27,29
    weights: uniform, distance
    metric: minkowski, euclidean, manhattan

2) Decision Tree
    criterion: gini, entropy
    max_depth: 3,4,5,6,7,8,9
    min_samples_leaf: 5,6,7,8,9,10,11,12,13,14,15,16, 17,18,19
    min_samples_split: 2,3,4,5,6,7,8,9
    max_features: auto, sqrt, log2, None

3) Naive Bayes
    alpha: 1, 0.1, 0.01, 0.001, 0.0001, 0.00001

4) SVM
    C: 0.1, 0.5, 1.0, 10.0, 100.0
    gamma: 1, 0.1, 0.01, 0.001, 0.0001
    kernel: rbf, linear, poly, sigmoid

5) Logistic Regression
    solver: newton-cg, lbfgs, sag, saga
    penalty: none, l2
    C: 0.001, 0.01, 0.1, 1, 10, 100

6) MLP
    Input_layer: 100, 200, 300
    Hidden_layers: (10), (50), (100), (10,10), (50,50), (100,10), (200,20), (300,30)
    Activation_function: logistic, tanh, relu
    Optimizer: sgd, adam
    Epochs: 100

E. Evaluation

Accuracy: The overall accuracy of a model is simply the number of correct predictions divided by the total number of predictions [8].

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

Precision: It is the number of correct positive results divided by the number of positive results predicted by the classifier [8].

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall: It is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive) [9].

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Where:

    TP: $True Positives$
    FP: $False Positives$
    TN: $True Negatives$
    FN: $False Negatives$

F1Score: Allow the balance between precision and recall [9].

$$F1Score = \frac{Precision * Recall}{Precision + Recall} \qquad (4)$$

The following table shows the best metrics of each Machine Learning model with its respective vectorizer after parameter tuning.

TABLE IX
METRICS OF MACHINE LEARNING MODELS

		Metrics			
		Accuracy	Precision	Recall	F1-Score
KNN	cv_1	0.80	0.77	0.88	0.83
	cv_2	0.80	0.76	0.90	0.82
	cv_3	0.78	0.74	0.91	0.81
	tf_idf_1	0.83	0.83	0.87	0.85
	tf_idf_2	0.83	0.82	0.87	0.84
	tf_idf_3	**0.86**	0.85	0.90	**0.87**
	w2v_1	0.69	0.70	0.73	0.71
	w2v_2	0.74	0.72	0.81	0.76
	w2v_3	0.69	0.67	0.79	0.73
	fastText_1	0.71	0.78	0.63	0.70
	fastText_2	0.72	0.75	0.70	0.72
	fastText_3	0.70	0.80	0.58	0.67
Decision Tree	cv_1	0.76	0.74	0.85	0.79
	cv_2	0.76	0.74	0.85	0.79
	cv_3	0.76	0.74	0.85	0.79
	tf_idf_1	0.76	0.75	0.83	0.78
	tf_idf_2	0.76	0.74	0.84	0.79
	tf_idf_3	**0.77**	0.76	0.81	**0.79**
	w2v_1	0.60	0.63	0.60	0.62
	w2v_2	0.63	0.67	0.59	0.63
	w2v_3	0.66	0.74	0.56	0.64
	fastText_1	0.69	0.70	0.71	0.71
	fastText_2	0.70	0.71	0.73	0.72
	fastText_3	0.62	0.63	0.70	0.66
Naive Bayes	cv_1	0.82	0.86	0.80	0.83
	cv_2	0.84	0.87	0.82	0.84
	cv_3	0.85	0.88	0.83	0.85
	tf_idf_1	0.81	0.82	0.83	0.82
	tf_idf_2	0.83	0.84	0.84	0.84
	tf_idf_3	**0.86**	0.87	0.87	**0.87**
	w2v_1	0.68	0.85	0.49	0.62
	w2v_2	0.68	0.88	0.47	0.61
	w2v_3	0.66	0.88	0.43	0.57
	fastText_1	0.73	0.80	0.65	0.72
	fastText_2	0.72	0.81	0.61	0.70
	fastText_3	0.70	0.81	0.58	0.67
SVM	cv_1	0.84	0.83	0.88	0.85
	cv_2	0.82	0.80	0.86	0.83
	cv_3	0.85	0.85	0.86	0.86
	tf_idf_1	0.86	0.88	0.84	0.86
	tf_idf_2	0.86	0.88	0.86	0.87
	tf_idf_3	**0.87**	0.88	0.87	**0.88**
	w2v_1	0.73	0.68	0.91	0.78
	w2v_2	0.80	0.83	0.78	0.80
	w2v_3	0.79	0.91	0.68	0.78
	fastText_1	0.62	0.58	1	0.74
	fastText_2	0.84	0.82	0.89	0.85
	fastText_3	0.84	0.92	0.77	0.84
Logistic Regression	cv_1	0.84	0.84	0.86	0.85
	cv_2	0.83	0.83	0.85	0.84
	cv_3	0.86	0.86	0.88	0.86
	tf_idf_1	0.85	0.87	0.84	0.86
	tf_idf_2	0.85	0.86	0.85	0.85
	tf_idf_3	**0.86**	0.88	0.87	**0.87**
	w2v_1	0.69	0.72	0.67	0.69
	w2v_2	0.74	0.77	0.72	0.74
	w2v_3	0.77	0.81	0.74	0.77
	fastText_1	0.76	0.78	0.74	0.76
	fastText_2	0.82	0.85	0.80	0.82
	fastText_3	0.80	0.83	0.78	0.80

MLP					
	cv_1	0.84	0.85	0.85	0.85
	cv_2	0.82	0.83	0.84	0.83
	cv_3	**0.85**	0.86	0.85	**0.85**
	tf_idf_1	0.84	0.87	0.83	0.85
	tf_idf_2	0.83	0.85	0.81	0.83
	tf_idf_3	0.84	0.85	0.85	0.85
	w2v_1	0.69	0.76	0.59	0.67
	w2v_2	0.74	0.77	0.72	0.74
	w2v_3	0.76	0.78	0.77	0.77
	fastText_1	0.74	0.72	0.82	0.77
	fastText_2	0.79	0.81	0.78	0.80
	fastText_3	0.74	0.70	0.87	0.78

IV. RESULTS

Below is a summary table of the models with the best combination of techniques, with their parameters and the results of the metrics:

TABLE X
COMPARISON OF THE BEST METRICS OF EACH MODEL

Model	Parameters	Accuracy	Precision	Recall	F1-Score
KNN (*tf_idf_3*)	n_neighbors: 25 weights: *distance* metric: *minkowski*	86.17%	84.65%	**90.19%**	87.33%
Decision Tree (*tf_idf_3*)	criterion: *gini*, max_depth: 9 max_features: *None* min_samples_leaf: 11 min_samples_split: 2	76.53%	75.98%	81.31%	78.56%
Naive Bayes (*tf_idf_3*)	alpha: 0.1	86.17%	86.92%	86.92%	86.92%
SVM (*tf_idf_3*)	**C: 0.5 gamma: 1 kernel:** *rbf*	**87.41%**	**88.63%**	87.38%	**88.00%**
Logistic Regression (*tf_idf_3*)	C: 10, penalty: *l2* solver: *newton-cg*	86.67%	87.74	86.92%	87.32%
MLP (*cv_3*)	1° capa: 100 neuronas 2° capa: 10 neuronas (*relu*) 3° capa: 1 neurona (*softmax*) Optimizer: adam	85.19%	86.32%	85.51%	85.92%

Although it is observed that the recall of SVM is lower than that of KNN, the other metrics manage to exceed them, which is why SVM is considered to be the best model for classifying false information about covid-19. This is reflected in its performance, since the model manages to correctly classify each covid record as *True* or *False* with 87.41% certainty, likewise, it classifies false positives to a lesser extent with an accuracy of 88.63% and, of the total information records labeled as True, manages to correctly classify 87.38% of the total, these last 2 metrics are reflected

5

in the value of the F1-score with 88%, being higher than the rest of the values.

V. CONCLUSIONS

The present work combines various methodologies and techniques from previous works to provide a solution to one of the problems that continues to have an impact today in Peru: the disinformation of COVID-19. Although the highest concentration of misinformation occurred during the pandemic (2020), over the years, a large part of Peruvian citizens have been influenced by misinformation about COVID-19, which has resulted today, many of them refuse to get vaccinated for fear that it could be harmful to their health.

The creation of the dataset was one of the most complicated processes, since not only was it enough to collect data from the web, but manual labeling and subsequent validation were also necessary. In the investigation, the web scraping technique was carried out to extract the data from the web automatically, greatly facilitating the collection of the same from sources such as Twitter, Facebook and web pages; however, not all the records collected were suitable for the development of the classification model since most of them dealt with COVID-19 topics in a sarcastic manner and with religious tendencies, while others did not talk about COVID-19 as such but rather of general topics for which they were discarded; with the remaining records, it was necessary to clean up for better labeling and validation with MINSA workers who are health experts.

Likewise, a good preprocessing was carried out, which was reflected in the results of the model. This preprocessing was carried out based on the information explained in the state of the art but with some modifications, such as the case of including some type of stemmer or not, at the beginning it was decided to do so but good results were not obtained, and the case of add Peruvian slang as part of the stop words, it was decided to do so since Peru is a country where this type of informal language is commonly used. With the preprocessing exposed in the work, it was enough to carry out a good training of the model.

The diversity of existing techniques in the field of Natural Language Processing is a great advantage when you want to vectorize textual records, since they facilitate the conversion of textual data to numerical data. For the investigation, 4 techniques were decided: Bag of Word, Tf-Idf, Word2Vec and fastText, each with 3 variants regarding its size (100, 200 and 300), these values were chosen because when testing with sizes of 500, 1000 and 1500, the results were lower than those finally chosen. While the best technique turned out to be Tf-Idf, this doesn't mean that embedding models are bad for vectorizing text; That is why, although these models have not achieved great similarities in their training due to the small corpus with which they were carried out (11631), they managed to provide good vectorizations of the records, this means that if they were trained with a greater amount of data, the context that would be provided to the words would be more precise, so its vectorization would probably be superior to the Tf-Idf.

In the same way, the different experiments carried out with the Machine Learning algorithms allowed us to find a robust classification model with an Accuracy of 87.41%, this model used the SVM technique with the parameters of C (0.5), gamma (1) and kernel. (rbf). Obtaining these results was more complex than initially believed, since, to tune the hyperparameters, it was not enough just to place different values, but it was also necessary to make sure that these values make sense and do not affect the model because otherwise the training it would not be the correct one, since it would fall in overfitting or underfitting. To reduce the risk of these problems, it was important to do cross-validation in each training, so that, in some cases, the models took a long time to run. In addition, the more parameters were added or the more values were added for each parameter, this time of training increased even more; however, because the data was small, the time did not exceed the execution hour, even so, it is important to take into account that if you decide to increase the size of the dataset, performing all these experiments can become exhausting.

Although in this research the url, authors and date of each record were not used as part of the modeling, it is planned that for future work they will be taken into consideration, since this metadata can provide relevant information about reliable or misleading sources. This is because the link and author of a record belonging to a reliable website such as the WHO will always have a characteristic structure that will differ from other unreliable websites that only seek to place links that attract the attention of readers. In this way, the metadata could be used together with the content of the same news in order to develop more robust classifiers of false information about COVID-19.

Finally, it can be concluded that a greater number of records and a greater diversity of them could improve the results evidenced with the 2022 records of the classification model, since, although the model can learn to differentiate false information from COVID-19 in the Peruvian context, this would be more robust if it were trained with more data, even so, this research provides a first scope for future works that want to take it as a reference and use a similar methodology for the detection of false information.

REFERENCES

[1] G. Nieves-Cuervo, E. Manrique-Hernández, A. Robledo-Colonia, and A. Grillo, "Infodemic: fake news and trends mortality from COVID-19 in six Latin American countries," Pan American Journal of Public Health., vol. 45, p. 44, June 2021.

[2] M. Bird, P. Muñoz, F. Freier, and S. Arispe, "48% of Peruvians who would not get vaccinated against COVID-19 believe that more tests are needed for vaccines," Research Center University of the Pacific.

[3] S. Bojjireddy, S. A. Chun, and J. Geller, "Machine learning approach to detect fake news, misinformation in covid-19 pandemic". DG.O2021: The 22nd Annual International Conference on Digital Government Research, June 2021.

[4] K. Hayawi, S. Shahriar, M. A. Serhani, I. Taleb, and S. S. Mathew, "ANTi-Vax: a novel Twitter dataset for COVID-19 vaccine misinformation detection", Public health, vol. 203, p. 23-30, December 2022.

[5] M. K. Elhadad, K. F. Li, and F. Gebali, "Detecting misleading information on COVID-19", Ieee Access, vol 8, p. 165201-165215, September 2020.

[6] A. R. Mahlous, and A. Al-Laith, "Fake news detection in Arabic tweets during the COVID-19 pandemic", International Journal of Advanced Computer Science and Applications, vol 12, p. 778-788, July 2021.

[7] Y. Tashtoush, B. Alrababah, O. Darwish, M. Maabreh and N. Alsaedi, "A Deep Learning Framework for Detection of COVID-19 Fake News on Social Media Platforms", Information Systems and Data Management, 7(5), 65, https://doi.org/10.3390/data7050065, may 2022.

[8] M. K. Dahouda and I. Joe, "A Deep-Learned Embedding Technique for Categorical Feature Encoding", in IEEE Access, vol. 9, pp. 114381-114391, 2021, doi: 10.1109/ACCESS.2021.3104357.

[9] M. K. Dahouda and I. Joe, "A Deep-Learned Embedding Technique for Categorical Features Encoding," in IEEE Access, vol. 9, pp. 114381- 114391, 2021, doi: 10.1109/ACCESS.2021.3104357.