

# ✱ Detección de noticias falsas con Machine Learning

Diego Rojas, Javier Mariño, Nubia Suárez

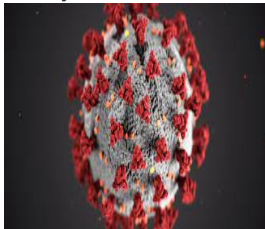
<https://github.com/fake-news-unal/fake-news>

---

August 3, 2021

# Introducción

**¿Qué son las noticias falsas?** Según el Ministerio de Tecnologías de la Información: "Se trata de historias o anuncios sensacionalistas, compartidos usualmente por redes sociales para involucrar emocionalmente a los lectores, generar atención y llevar a sus creadores a obtener rentabilidad por clics y viralización".



**¿Por qué son peligrosas?** Son utilizadas para desprestigiar a personajes públicos, robar y acceder a información personal, entorpecer el manejo de situaciones críticas (como el tratamiento de enfermedades) y obtener réditos políticos.

# Problema

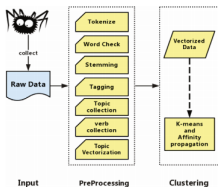
El problema a resolver por la estadística, y en específico el machine learning, es la detección de noticias falsas. Clasificar un texto como verdadero o falso de acuerdo a los procedimientos desarrollados dentro del Natural Language Processing.

El propósito de este proyecto fue comparar la efectividad, medida por la precisión, de distintos métodos del machine learning (Neural networks, LSTM, Naive Bayes, Regresión Logística,...) para la detección de noticias falsas.



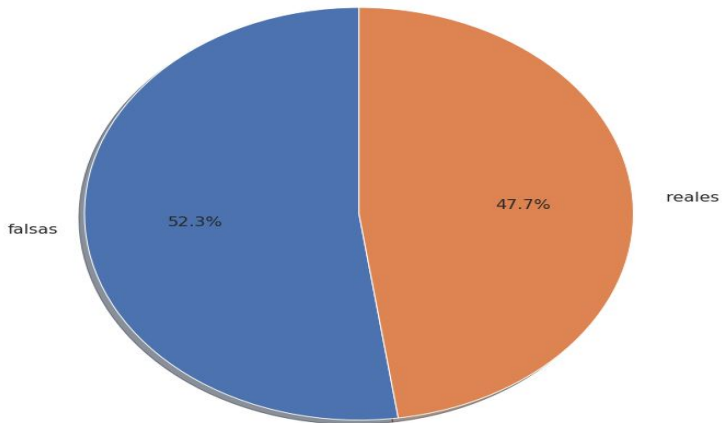
# Metodología

- ✦ La base de datos utilizadas para comparar los distintos métodos fue obtenida del portal Kaggle. Cuenta con un aproximado de 23.000 noticias falsas y 21.000 noticias verdaderas, con sus respectivos títulos, sobre política norteamericana.
- ✦ Los textos fueron tratados mediante las librerías Gensim y Stanza de Python. Con ellas se realizó el proceso de tokenización, lematización, eliminación de stop words y vectorización.
- ✦ Con los textos vectorizados se dividió la base tratada en un 80% de validación y 20% de testeo para la aplicación de los métodos.



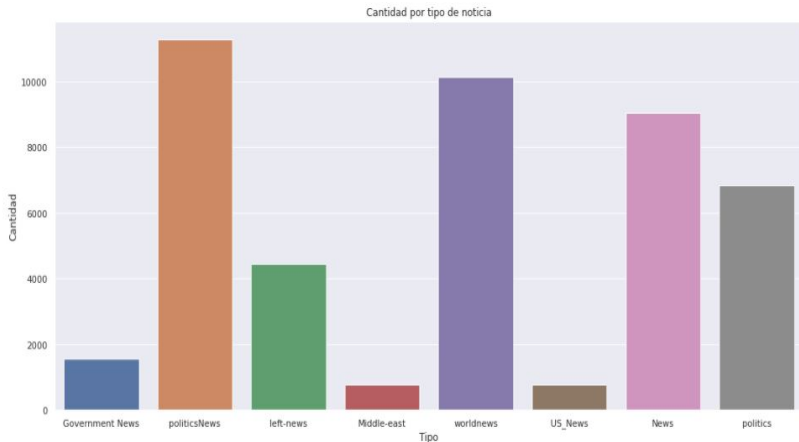
# Visualización de los datos

## Porcentaje de noticias



# Visualización de los datos

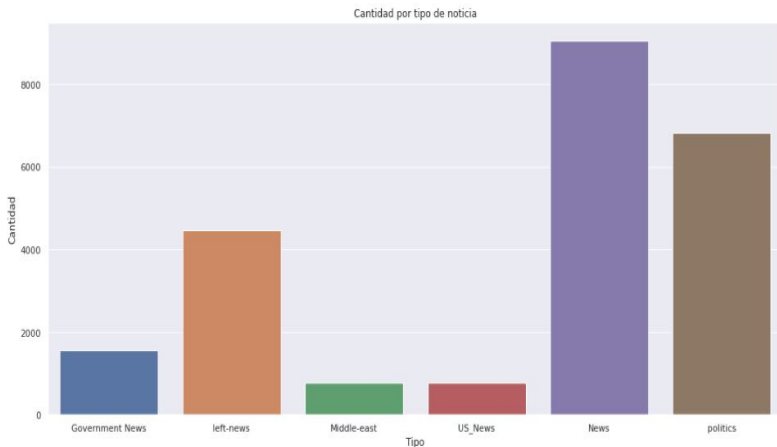
## TIPO: Noticias Reales





# Visualización de los datos

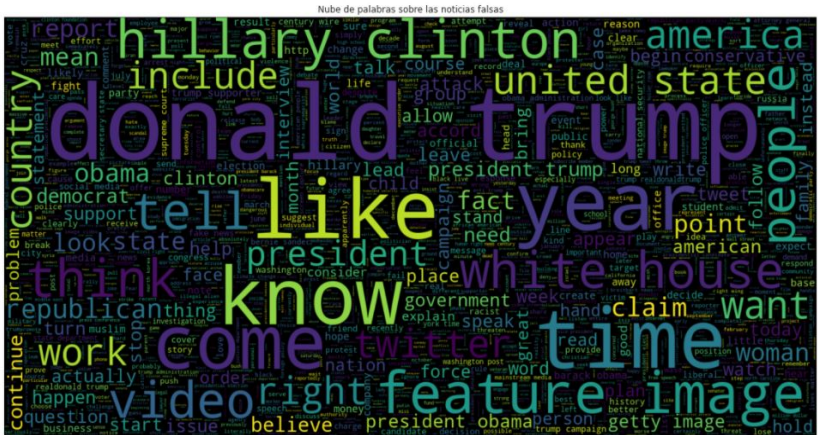
## TIPO: Noticias Falsas



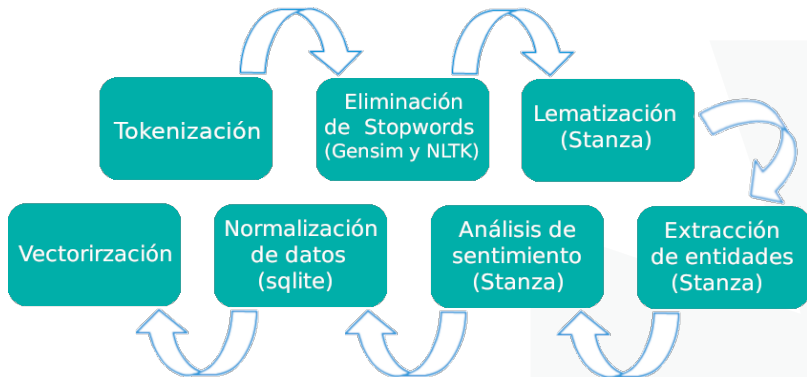


# Visualización de los datos

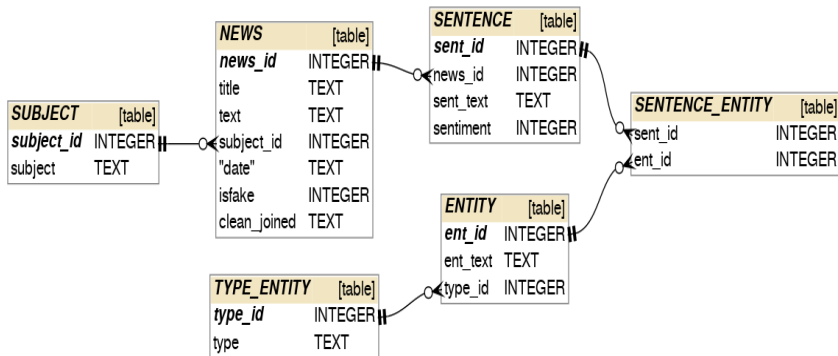
## TIPO: Noticias Falsas



# Preprocesamiento



# Preprocesamiento (Modelo de datos)



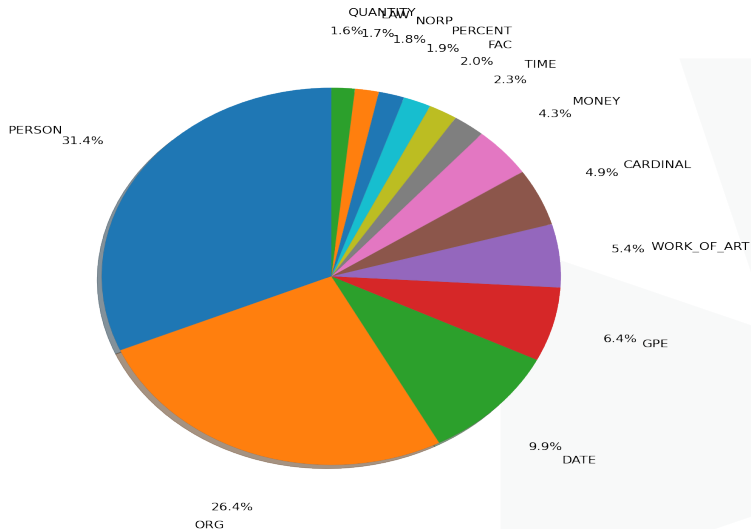
# Exploración de entidades

## Tabla de tipos de entidades

Tipo	Descripción
GPE	Países, ciudades o estados
NORP	Nacionalidades, grupos políticos o religiosos
ORG	Compañías, agencias o instituciones
DATE	Periodos y fechas
PERSON	Personas incluyendo ficción
PERCENT	Porcentajes
MONEY	Valores monetarios
CARDINAL	Números
ORDINAL	Números en letras "first", "second", etc
LAW	Documentos nombrados que representan leyes
WORK_OF_ART	Títulos de libros, canciones, etc
TIME	Tiempos que son inferiores a un día
FAC	Construcciones, edificios, avenidas, etc.
LOC	Ubicaciones diferentes a GPE como ríos, montañas, etc
EVENT	Huracanes, batallas, guerras, eventos deportivos, etc
QUANTITY	Medidas como pesos o distancias
PRODUCT	Objetos, carros, comida, etc (servicios no)
LANGUAGE	Cualquier idioma mencionado

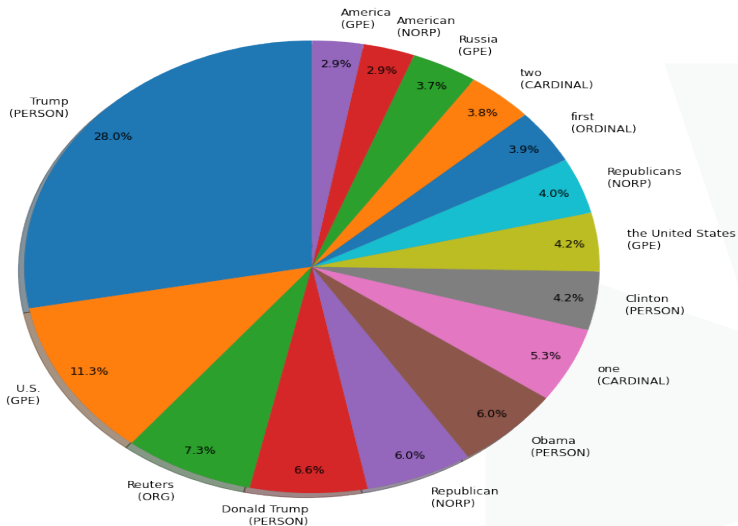
# Exploración de entidades

## Tipos de entidades



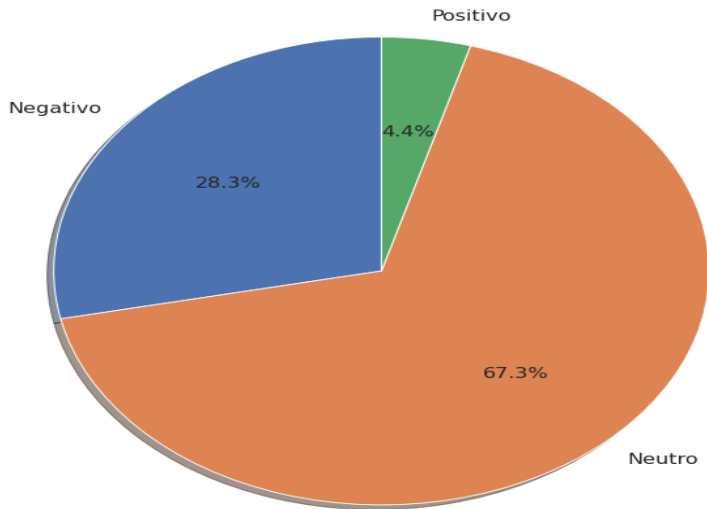
# Exploración de entidades

## Entidades



# Exploración de sentimiento

## Sentimiento

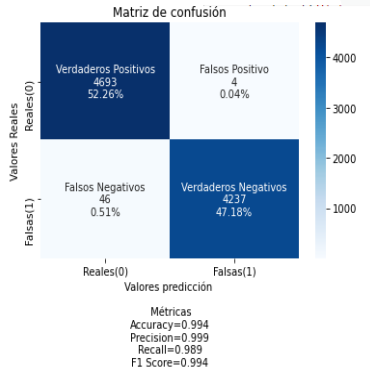


# Red Neuronal

Para la construcción de la red neuronal se utilizaron como inputs los vectores generados por Gensim y Stanza. El optimizador utilizado fue "Adam" y la función de perdida la "binary-crossentropy". El modelo alcanzó, después de 2 épocas, una precisión de 0,9997 en entrenamiento y 0,9985 en validación.

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 40, 128)	1132157568
flatten (Flatten)	(None, 5120)	0
dense (Dense)	(None, 64)	327744
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2080
dropout_1 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 16)	528
dropout_2 (Dropout)	(None, 16)	0
dense_3 (Dense)	(None, 1)	17
Total params: 1,132,487,937		
Trainable params: 1,132,487,937		
Non-trainable params: 0		





# Naive Bayes

El clasificador bayesiano se construyó con base en los inputs generados por Gensim y Stanza. Se limitaron las secuencias a vectores de máximo 40 componentes con la función `pad_sequences` de TensorFlow. Además, se ajustó mediante la librería ScikitLearn de Python el algoritmo Gaussiano Naive Bayes de Clasificación.

Con un 80% de los datos dirigidos al entrenamiento y un 20% al testeo, se alcanzó la precisión más baja entre todos los modelos con tan sólo un 59% de noticias clasificadas correctamente.

# LSTM (Long short-term memory)

Model: "sequential"

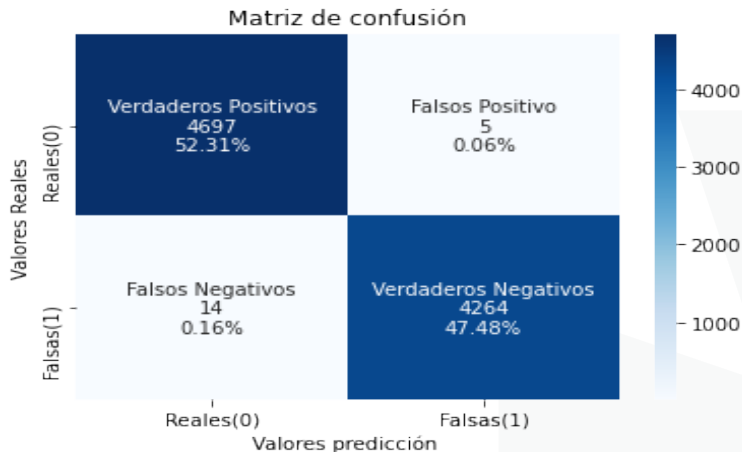
Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, None, 128)	1132157568
bidirectional (Bidirectional)	(None, 256)	263168
dense (Dense)	(None, 128)	32896
dense_1 (Dense)	(None, 1)	129

=====  
Total params: 1,132,453,761

Trainable params: 1,132,453,761

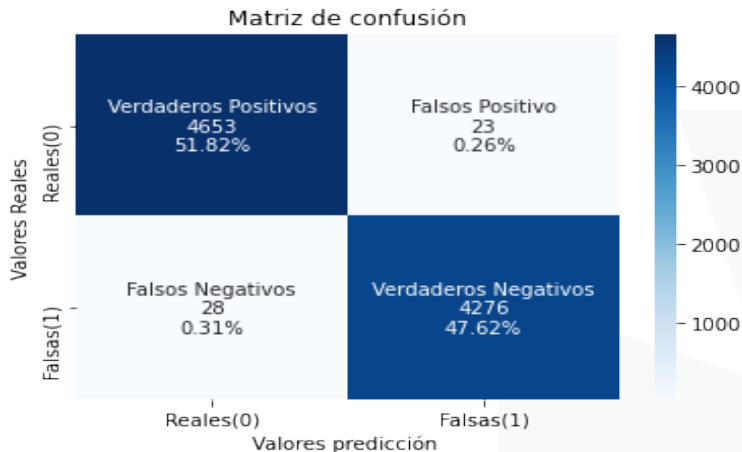
Non-trainable params: 0

# LSTM (Long short-term memory)



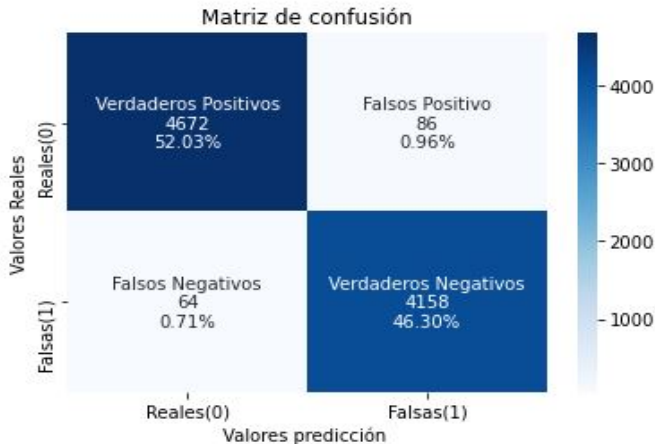
Métricas  
Accuracy=0.998  
Precision=0.999  
Recall=0.997  
F1 Score=0.998

# SVM (Support vector machine)



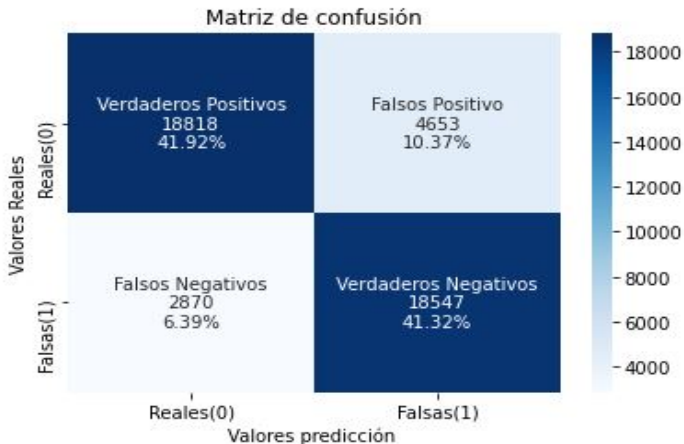
Métricas  
Accuracy=0.994  
Precision=0.995  
Recall=0.993  
F1 Score=0.994

# Regresión Logística



Métricas  
Accuracy=0.983  
Precision=0.980  
Recall=0.985  
F1 Score=0.982

# K-means (K-medias)



Métricas  
Accuracy=0.832  
Precision=0.799  
Recall=0.866  
F1 Score=0.831

# Comparación de resultados




	<b>LSTM</b>	<b>SVM</b>	<b>Neural Network</b>	<b>Logistic</b>	<b>K Means</b>	<b>Naive Bayes</b>
<b>Accuracy</b>	0,998	0,994	0,994	0,983	0,832	0,589
<b>Precision</b>	0,999	0,995	0,999	0,980	0,799	0,542
<b>Recall</b>	0,997	0,993	0,989	0,985	0,866	0,852
<b>F1 Score</b>	0,998	0,994	0,994	0,982	0,831	0,663

# Conclusiones

- ✚ El modelo que mejor resultados obtuvo fue el LSTM con resultados muy cercanos a la red neuronal y el modelo con los resultados más bajos fue Naive Bayes.
- ✚ Aunque con precisiones distintas, los modelos machine learning son valiosos para detectar noticias falsas. Constituyen una solución al problema presentado.
- ✚ Ahora el verdadero problema a solucionar es como implementar estos identificadores de noticias falsas, basados en machine learning, en la detección en tiempo real.



# Referencias

-  Stanford NLP Group. (2014). Stanza v1.2.0 [software library]. Stanford, California. Disponible en <https://stanfordnlp.github.io>
-  Ahmed H, Traore I, Saad S. “Detecting opinion spams and fake news using text classification”, Journal of Security and Privacy, Volume 1, Issue 1, Wiley, January/February 2018.
-  Ahmed H, Traore I, Saad S. (2017) “Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. In: Traore I., Woungang I., Awad A. (eds) Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. ISDDC 2017. Lecture Notes in Computer Science, vol 10618. Springer, Cham (pp. 127-138).