# UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

## FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA


# DETECCIÓN DE NOTICIAS FALSAS, BOTS Y POTENCIAL VIRAL EN LAS REDES SOCIALES MEDIANTE APRENDIZAJE AUTOMÁTICO

**POR**

**CARLOS AUGUSTO JIMÉNEZ ZARATE**

**EN OPCIÓN AL GRADO DE**
**DOCTOR EN INGENIERÍA CON ORIENTACIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN**

**SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN. SEPTIEMBRE 2024.**

**UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN**
**FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA**
**SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO**

# DETECCIÓN DE NOTICIAS FALSAS, BOTS Y POTENCIAL VIRAL EN LAS REDES SOCIALES MEDIANTE APRENDIZAJE AUTOMÁTICO

**POR**

**CARLOS AUGUSTO JIMÉNEZ ZARATE**

**EN OPCIÓN AL GRADO DE:**
**DOCTOR EN INGENIERÍA CON ORIENTACIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN**

**SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN. SEPTIEMBRE 2024.**

# UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
## Facultad de Ingeniería Mecánica y Eléctrica
## Posgrado

Los miembros del Comité de Evaluación de Tesis recomendamos que la Tesis "Detección de noticias falsas, bots y potencial viral en las redes sociales mediante aprendizaje automático", realizada por el estudiante Carlos Augusto Jiménez Zarate, con número de matrícula 792875, sea aceptada para su defensa como requisito parcial para obtener el grado de Doctor en Ingeniería con orientación en Tecnologías de la Información.

**El Comité de Evaluación de Tesis**

Dra. Leticia Amalia Neira Tovar
Director

Dra. María Margarita Carrera Sánchez
Co-director

Dr. Jesús Adolfo Meléndez Guevara
Revisor

Dr. Carlos Esteban Chávez Pech
Revisor

Dr. Salvador Barrera Aldana
Revisor

Dra. Marta Silvia Del Río Guerra
Revisor

Dra. María Teresa Pérez Morales
Revisor

Vo.Bo.

Dr. Simón Martínez Martínez
Subdirector de Estudios de Posgrado

Institución 190001
Programa 551515
Acta Núm. 351

Ciudad Universitaria, a 11 de septiembre de 2024.

# AGRADECIMIENTOS

Agradezco a la Dra. Leticia Amalia Neira Tovar por su apoyo, revisión y contribuciones hechas a la presente investigación, asi mismo hago extensivo este agradecimiento a los revisores el Dr. Jesús Adolfo Meléndez Guevara, Dr. Carlos Esteban Chávez Pech, Dr. Salvador Barrera Aldana, Dra. Marta Silvia del Rio y la Dra. María Margarita Carrera Sánchez.

Agradezco a los Doctores: Dra. Aída Lucina González Lara, Dra. Leticia Amalia Neira Tovar, Dra. Sara Elena Garza Villarreal, Dra. María Isolde Hedlefs Aguilar, Dr. Carlos Esteban Chávez Pech, Dr. Giovanni Lizárraga Lizárraga, Dr. Luis Martín Torres Treviño y el Dr. Cesar Guerra Torres, integrantes del cuerpo docente de posgrado, al Dr. Arnulfo Treviño Cubero director de la Facultad de Ingeniería Mecánica y Eléctrica (FIME), al Dr. Simón Martínez Martínez subdirector de Posgrado de FIME, también agradezco a la Universidad Autónoma de Nuevo León (UANL) mi *Alma Máter*.

Gracias al maestro Jenaro Villamil Rodríguez presidente del Sistema de Público de Radiodifusión (SPR) del estado mexicano, al maestro Miguel Elorza-Vázquez director de la iniciativa Infodemia MX, quienes apoyaron esta investigación con la validación de un nuevo dataset de noticias falsas difundidas en Twitter en idioma español, en el contexto de la pandemia por COVID-19.

De igual manera agradezco a todas aquellas personas con las que coincidí durante el desarrollo de esta investigación, a mis alumnos, amigos y familiares que me alentaron en cada fase del doctorado.

Un agradecimiento especial a mis hermanos Tomás, Cynthia, Violeta y Stephanie por sus palabras de aliento, y a mis padres Teresa de Jesús Zarate Domínguez y Tomás Jiménez García por su amor infinito.

*Con amor para mis hijos*

*Javier, Oliver y Lorena,*

*y a mi esposa Lorena Cruz.*

*Especialmente dedico este esfuerzo*

*a Dios, la energía creadora.*

# ÍNDICE GENERAL

# RESUMEN

En la actualidad la mayoría de los modelos para la detección de noticias falsas en idioma español son entrenados con algoritmos de aprendizaje automático con datasets en idioma inglés o multilingües, para esta investigación se probaron tres datasets, dos ya existentes (Constraint e IberLef) y un tercer nuevo dataset que fue desarrollado mediante un proceso de par ciego, para lo cual se extrajeron tweets que interactuaron con el tema de la pandemia por COVID-19, se clasificaron de forma manual como noticia falsa o real en un primer momento por alumnos colaboradores de la Facultad de Ingeniería Mecánica y Eléctrica (FIME) de la Universidad Autónoma de Nuevo León (UANL), después se validaron de forma independiente por la iniciativa Infodemia.MX, perteneciente al Sistema Público de Radiodifusión (SPR) del estado mexicano. Para el desarrollo del modelo se consideró como mejor algoritmo el de regresión logística (RL) que obtuvo una exactitud con el dataset Constraint de 91.5% con un promedio del 81.1% entre los demás datasets. Además, la presente investigación desarrolló un segundo dataset “Bot COVID-19” que incluye una clasificación bot por un medio externo y diversas características de los usuarios de redes sociales. Para la detección de cuentas bot tomando en cuenta solo datos de su perfil público, se determinó que el algoritmo de extremo gradiente (XGBoost) fue el mejor, debido a que obtuvo una máxima exactitud (*Accuracy*) del 85.15%. Por último, se desarrolló un modelo capaz de determinar si un usuario es potencialmente viral o no, de acuerdo con el análisis de su red de amigos de amigos.

# CAPÍTULO 1

# INTRODUCCIÓN

## 1.1. Antecedentes

Las nuevas tecnologías de información y comunicación (TIC ´s) han avanzado vertiginosamente, permitiendo el procesamiento de grandes cantidades de datos y el desarrollo de plataformas digitales de comunicación social, esto ha ayudado a que exista hoy en día, una gran conexión social entre millones de personas, lo cual ha tenido un gran impacto en temas sociales, laborales y culturales. Pero en esta nueva dinámica de la comunicación social también han aumentado la difusión de contenidos de odio, clasismo o racismo, de las noticias catalogadas como falsas (*fake news*) y la polarización política y social [1] [2] [3] [4].

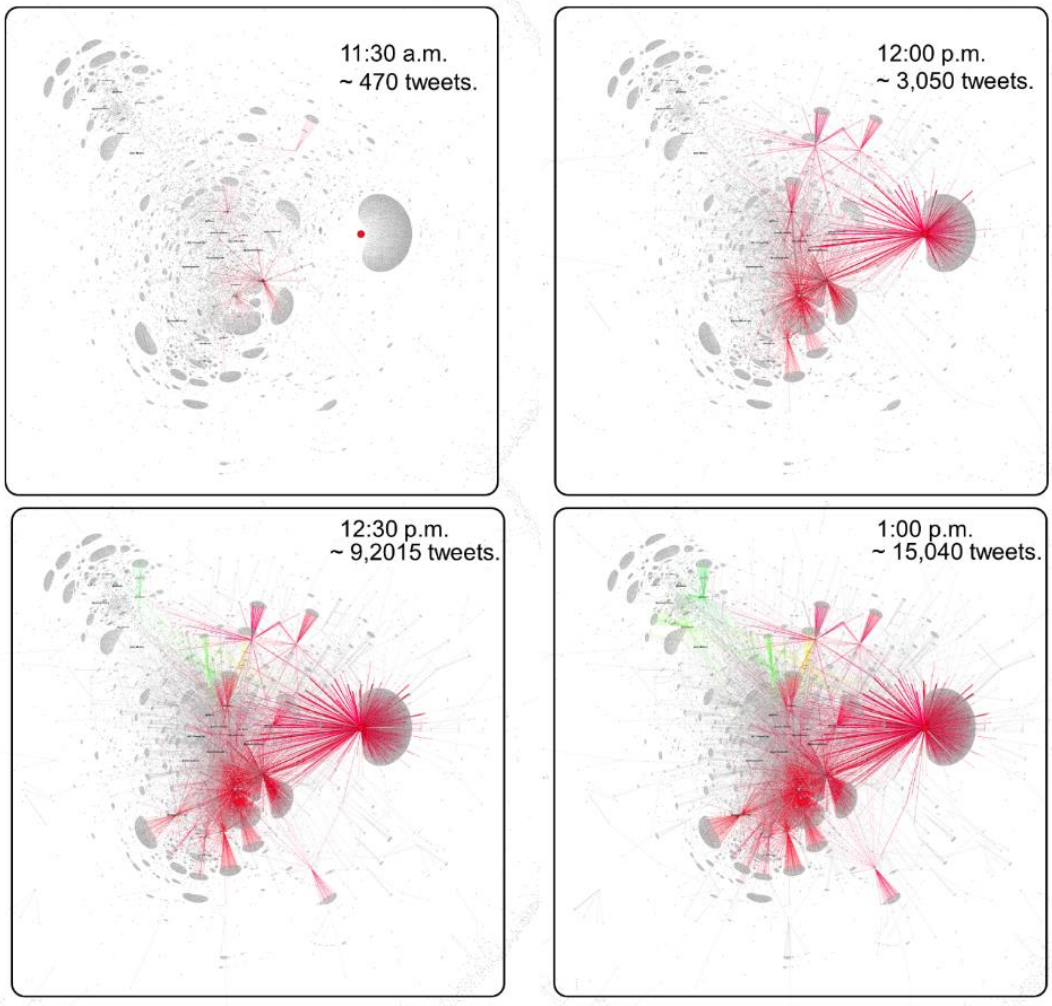
Knight Foundation reveló que más de un millón de tweets se transmiten con noticias falsas a nivel mundial; en México una investigación de la Comisión Nacional de Derechos Humanos de México (CNDH) alertó sobre los contenidos con información que no es exacta o no relata los hechos objetivamente, sino que tratan de manipular a los usuarios para obtener una mayor cantidad de interacciones de "*me gusta*" o también conocidos como "*likes*" en idioma inglés, o republicaciones (retweets), además el reporte de la CNDH mencionó que la mayoría de los usuarios no se toman el tiempo

suficiente, o algunos no poseen los recursos o instrumentos para verificar la información. La Universidad Nacional Autónoma de México (UNAM) en su reporte DGCS-318 determinó que, en las redes sociales más utilizadas por los mexicanos se difunden una gran cantidad de información falsa, también asegura que por lo menos un 89% de los usuarios de la plataforma Twitter han sido impactados por la desinformación [5] [6] [7].

Las noticias falsas se popularizan a una gran velocidad a través de las redes sociales, en la figura 1.1 podemos ver propagación de una noticia falsa de una supuesta balacera en el aeropuerto de Cancún el día 27 de marzo del año 2022. Todo inicio aproximadamente a las 11:30 a.m. en pocos minutos la noticia falsa había alcanzado una cantidad de 470 tweets, pero tan solo media hora después (12:00 p.m.) los tweets sobre esta supuesta balacera habían alcanzado más de 3,000 interacciones, los tweets aclaratorios de la noticia falsa se emitieron aproximadamente a las 12:30 p.m. Es decir, una hora después de haber iniciado la noticia falsa, sin embargo, la noticia falsa siguió propagándose, llegando a más de 15,000 interacciones a la 1:00 p.m.

**Figura 1.1.**

*Estructura y dinámica de una noticia falsa*




Nota: El color rojo indica interacciones con noticia falsas. Elaboración propia.

Las noticias falsas se han vuelto un gran problema de tal forma que, en la pandemia por COVID-19 la Organización de las Naciones Unidas (ONU) desarrolló una plataforma que permite verificar las noticias relacionadas con la pandemia en el portal "*shareverified.com*" (Figura 1.2).

**Figura 1.2.**

*Plataforma de la ONU para verificar información relacionada al COVID-19.*


*Fuente*: https://twitter.com/ONU_es/status/1263502059019489281.

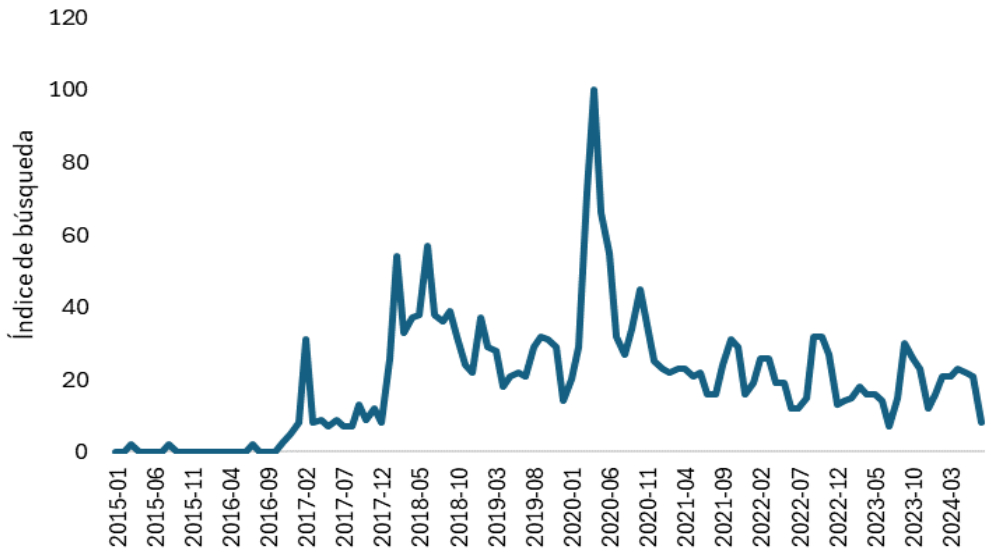
El diccionario de Cambridge asegura que las noticias falsas son historias que parecen noticias y que se propagan en la red de internet, con una finalidad política o de broma [8].

Las "*noticias falsas*" fueron catalogadas como las palabras más relevantes en el año 2017 por la editorial británica Collins [9], el significado de las noticias falsas o "*fake news*" ha sido constantemente buscado por usuarios de internet en Google, tal como se puede observar en la figura 1.3 donde se observa el índice en Google México de la búsqueda del término "***fake news"*** como son conocidas las "*noticias falsas*" en México, dentro *d*el periodo de análisis que comprende del año 2015 hasta el primer trimestre del año 2024, se puede ver un incremento a finales del año 2016, con un pico máximo en

marzo del 2020 que coincide con la declaración de pandemia del COVID-19 en México [10].

**Figura 1.3.**

*Búsqueda en Google en México de noticias falsas (fake news).*



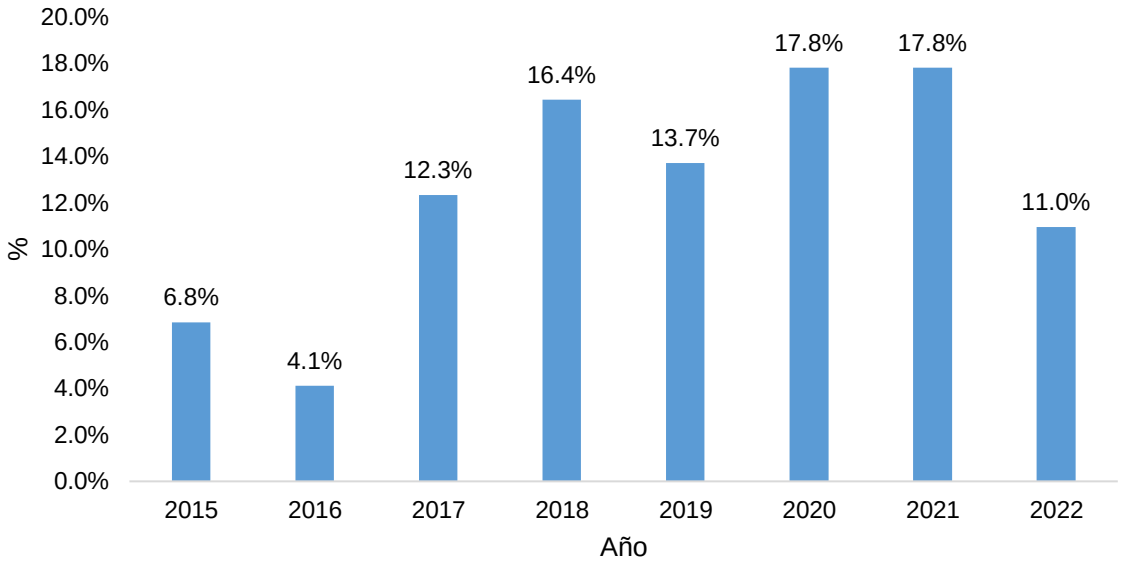

*Fuente*: Google Trends. Elaboración propia.

## 1.2. Estado del arte

Para obtener información relacionadas con el tema de esta investigación, se realizó una búsqueda en plataformas como "*Google Scholar*" y del Consorcio Nacional de Recursos de Información Científica y Tecnológica (Conricyt) perteneciente al Consejo Nacional de Ciencia y Tecnología (CONACYT), después se procedió a la lectura exhaustiva de los diversos artículos de investigación, en la figura 1.4 se puede observar la distribución por año de publicación de los artículos más recientes que se utilizaron para llevar a cabo la presente investigación.

**Figura 1.4.**

*Porcentaje de artículos más recientes utilizados para la presente investigación.*




Elaboración propia.

Twitter ahora conocido como "*X.com*", es una plataforma muy utilizada para difundir contenidos que promueven la desinformación o noticias falsas, una parte de estas campañas de difusión de la desinformación son apoyadas por cuentas automatizadas o "*bots*" que tienen la finalidad de impulsar como una tendencia (trending topic) a nivel local, regional o nacional, estas cuentas bots también se han detectado en campañas de odio [11].

Las noticias falsas se propagan con una mayor velocidad, además son capaces de crear cascadas de difusión de mayor profundidad comparadas con las noticias reales, dentro de las áreas de mayor impacto de las noticias falsas se encuentran la política, los deportes y los contenidos relacionados con los espectáculos [12].

El uso de las redes sociales genera estructuras que se analizan mediante el uso de modelos de sistemas complejos o probabilísticos, por ejemplo se ha visto que la probabilidad de que un usuario sea propenso a un contenido especifico, depende de la cantidad de fuentes originales de

información, en este contexto la propagación de contenidos mediante el uso de cuentas bot tienen una importancia para la propagación de las noticias falsas o con información errónea, en general de la desinformación [13].

La iniciativa “Hoaxy” de la Universidad de Indiana determinó mediante el análisis de 14 millones de tweets que se difundieron 400 mil noticias o información falsa durante 10 meses entre los años 2016 y 2017, y que gran parte la desinformación en esos tweets se debió a cuentas bot, los cuales son programas que pueden emitir contenido de forma automática, ellos desarrollaron una herramienta de análisis y verificación para las cuentas bot llamada “*BotoMeter*”, ambas herramientas usan los metadatos e información extraída de los usuarios, calculando además patrones de interacción y su contenido con otros usuarios [14] [15].

Gran parte de las investigaciones relacionadas con las noticias falsas, se han sustentado en el uso de algoritmos como el de Naive-Bayes (NB), Máquinas de Vectores de Soporte (MVS), también se han utilizado algoritmos de regresión logística y de árbol de decisión [16] [17] [18] [19] [20] [21] [22]. Otros estudios han utilizado técnicas de redes recurrentes, así como algoritmos de aprendizaje profundo (*depp learning*) para la clasificación automática de las noticias falsas [23] [24] [25] [26] [27].

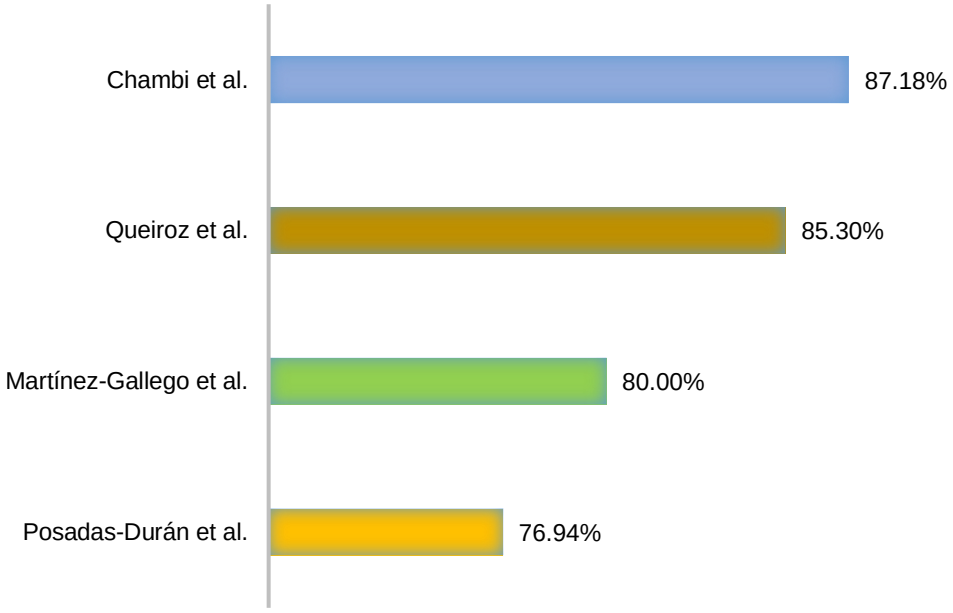
Como hemos visto, la mayoría de los modelos para la detección automática de noticias falsas se dividen en dos grupos, los algoritmos de aprendizaje automático y los algoritmos de aprendizaje profundo, ambos grupos se entrenan con datasets en idioma inglés y son muy pocos los modelos desarrollados para el idioma español, otras investigaciones han desarrollado modelos multi-lingüistas que son capaces de realizar traducciones de varios idiomas al inglés [28] [29] [30] [31] [32].

Para la detección de noticias falsas en idioma español se han desarrollado diversas investigaciones que han alcanzado una exactitud (*Accuracy*) del 87.18%, en la figura 1.5 se pueden observar la precisión alcanzada por otras investigaciones en la detección de noticias falsas en

español, el promedio de la precisión de estas investigaciones es del 82.36 % con una desviación estándar de 4.76% [33].

**Figura 1.5.**

*Exactitud (accuracy) de modelos de detección automática de noticias falsas en idioma español.*




Elaboración propia.

## 1.3. Justificación

Durante la pandemia por COVID-19 se generó otro tipo de epidemia, una relacionada a la gran cantidad de información real o falsa que se emite principalmente en las redes sociales, este fenómeno de información fue definido como Infodemia [34]; ante este escenario se hizo necesario nuevas plataformas de análisis de las noticias, para poder determinar si un contenido noticioso era real o falso, los contenidos *infodemicos* se han hecho presentes en los desastres naturales, temas políticos, movimientos sociales, marketing empresarial y en la salud pública, gran parte de esta avalancha desinformativa

se ha vuelto un problema de comunicación social que afecta la percepción de la realidad social [35] [36] [37] [38].

Tal como se ha visto, en el apartado 1.2 los estudios para detectar las noticias falsas, se sustentan en el uso de modelos diseñados primordialmente para noticias emitidas en idioma inglés, siendo muy pocos los que se han enfocado en el idioma español [39] [40] [41] [42], y no existe un sistema automático en México para la detección de noticias falsas en redes sociales en idioma español mexicano que supere el 90% de exactitud, esta investigación planteó una exploración exhaustiva de las investigaciones relacionadas con la detección automática de noticias falsas, para el desarrollo de un nuevo modelo que se sustente en el aprendizaje automático para detectar noticias falsas y cuentas bot, además que pueda determinar el potencial viral de un usuario de redes sociales.

## 1.4. Definición del problema

La mayoría de los modelos automáticos para la detección de noticias falsas fueron realizadas para el idioma inglés, y las pocas que existen en idioma español no superan el 90% de exactitud y no ofrecen la detección para el español mexicano; por lo que esta investigación se propuso desarrollar un modelo basado en algoritmos de aprendizaje automático que sea capaz de detectar una noticia falsa en idioma español mexicano por arriba del 90% de exactitud, además de la detección de usuarios tipo bot y el potencial viral de los usuarios de las redes sociales. Para la detección bot la mayoría de los sistemas implementados necesitan además de las características públicas de los usuarios ciertos metadatos no públicos y el análisis temporal de la actividad del usuario. Por último, el análisis viral se ha enfocado en los contenidos y pocos modelos han integrado el análisis del potencial viral de un usuario o cuenta de acuerdo con su red de amigos de amigos.

## 1.5. Objetivos

### 1.5.1. Objetivo general

Utilizar algoritmos de aprendizaje automático para el desarrollo de un modelo que sea capaz de detectar automáticamente una noticia falsa en idioma español que supere el 90% de exactitud, además que determine si un usuario de redes sociales es una cuenta bot y su potencial viral.

### 1.5.2 Objetivos específicos

- Analizar y categorizar tweets (X.com) de noticias reales y noticias falsas manualmente por el método de par ciego, para el desarrollo de un nuevo dataset de tweets en español mexicano.
- Analizar y categorizar tipo de usuarios de Twitter (X.com) para la creación de un nuevo dataset de usuarios tipo bot o humanos.
- Instrumentar métricas de evaluación para la comparación de datasets de noticias falsas en idioma español.
- Implementar un modelo de análisis para detectar noticias falsas en español mexicano, usuarios tipo bot y potencial viral mediante algoritmos de aprendizaje automático.

## 1.6. Hipótesis

Como se ha visto los modelos existentes para la detección de noticias falsas en español tienen un máximo de exactitud del 87.18%, para esta investigación se ha formulado una hipótesis general que asegura lo siguiente:

Es posible incrementar la exactitud de la detección de noticias falsas en idioma español mexicano, mediante el desarrollo de un modelo que utilice algoritmos de aprendizaje automático.

Otras hipótesis que se plantean dentro de esta investigación son:

- Es posible la detección de cuentas bot en redes sociales de México utilizando sus características de carácter público, mediante el uso de algoritmos de aprendizaje automático.
- Es posible calcular el potencial viral mediante el análisis de la red amigos de amigos de un usuario de redes sociales.

## 1.7. Preguntas de investigación

- ¿Es posible incrementar la exactitud de la detección de noticias falsas en español mexicano en las redes sociales mediante un modelo que utilice algoritmos de aprendizaje automático?
- ¿Es posible desarrollar un modelo para la detección de cuentas bot mediante las características públicas del perfil de usuario, utilizando algoritmos de aprendizaje automático en redes sociales?
- ¿Se puede determinar el potencial viral de un usuario de redes sociales por su red de amigos de amigos?

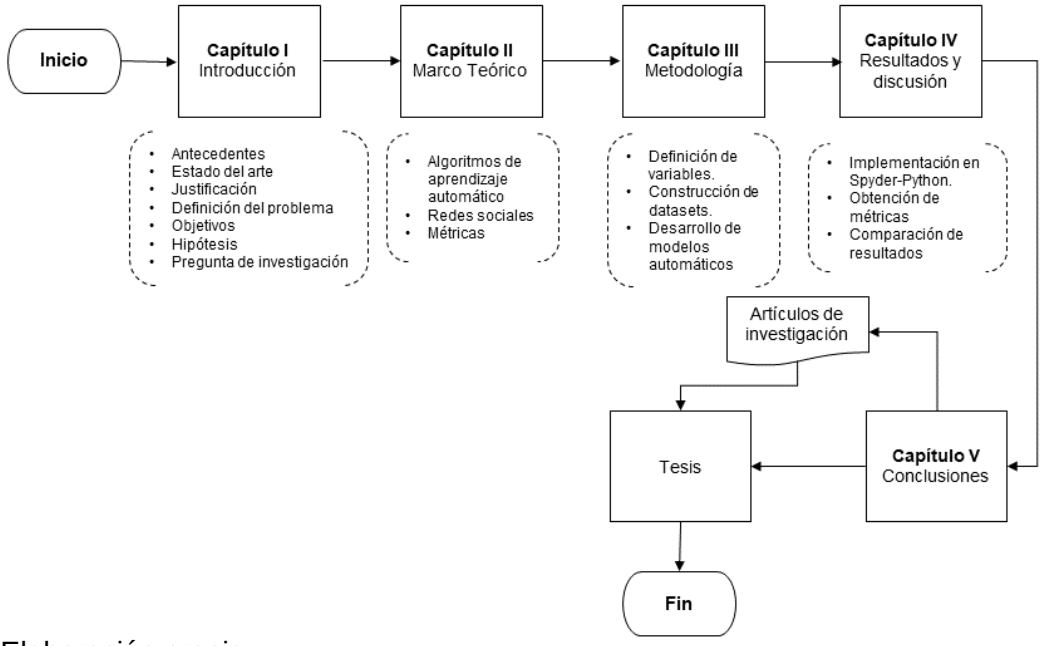
## 1.8. Organización de la tesis

Para llevar a cabo la siguientes investigación se presentó el siguiente flujo de trabajo (figura 1.6) en el capítulo 1, se expone los antecedentes y el estado del arte de las investigaciones realizadas anteriormente, además se tomó en cuenta la antigüedad de los artículos, la mayoría tiene una antigüedad máxima de cinco años, en este capítulo se encuentra definida la justificación para llevar a cabo la presente investigación, asi como la definición del problema, también se detalló el objetivo general y los objetivos específicos, el planteamiento de la hipótesis general y las hipótesis complementarias y por último se presentó las preguntas de investigación; en el capítulo 2 se aborda el marco teórico, donde se definen las nociones teóricas de los algoritmos de aprendizaje automático sus fórmulas y algunos ejemplos para su mayor

comprensión, también se presenta en este capítulo las teorías relacionadas con el análisis de redes sociales, para este caso se abordó las métricas relacionadas con la topología, influencia y la potencia viral de una red. El capítulo 3 se expone la metodología que se instrumentó para dar cumplimiento a los objetivos general y específicos, para lo cual se definieron las variables independientes tales como texto, perfil de usuario y red de amigos, además se estableció su relación con las variables dependientes, en este capítulo también se presentó el desarrollo de nuevos datasets y la adaptación de dos datasets para el entrenamiento para los algoritmos de aprendizaje automático, y por último se presentaron los modelos de detección automática de noticias falsas y cuentas bot; en el capítulo 4 se presenta los resultados obtenidos asi como la comparación de sus resultados. Por último, en el capítulo 5 se presenta las conclusiones y las líneas de trabajo que esta investigación puede abordar en el futuro.

**Figura 1.6.**

*Diagrama de flujo de la tesis.*




Elaboración propia.

# CAPÍTULO 2

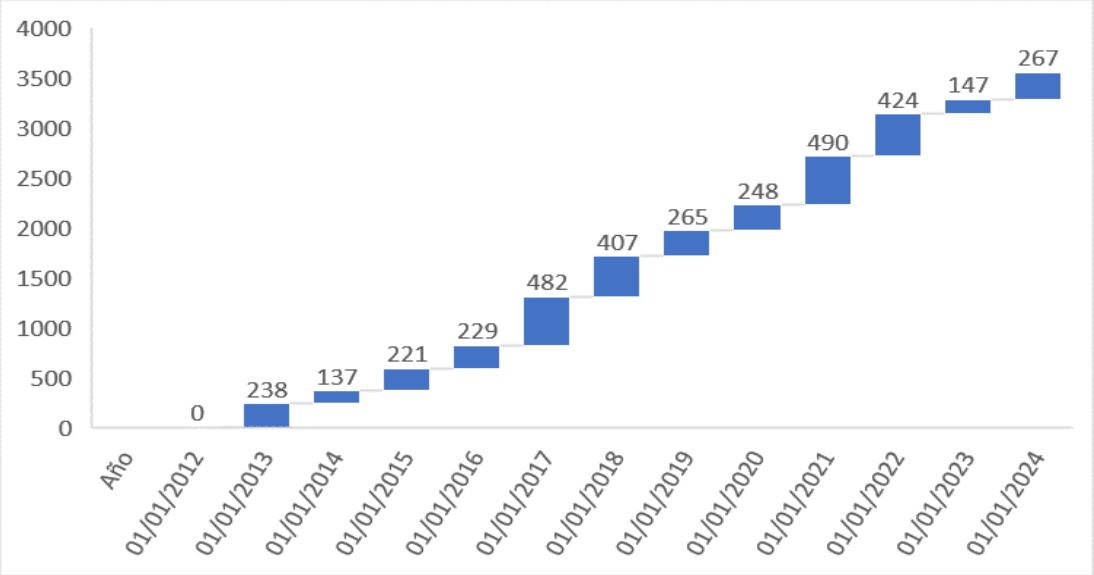
# MARCO TEÓRICO

## 2.1. Interacción digital

Los usuarios de las redes sociales o “*social media*” han aumentado en los últimos diez años, en la figura 2.1 se observa el incremento de usuarios por cada año desde el 2012 hasta el 2024, se puede ver que los años de mayor crecimiento fueron el 2017, 2021 y el 2022. Particularmente en el año 2021 se mostró el mayor incremento con 490 millones de nuevos usuarios, este comportamiento concuerda con el año en que se presentó la pandemia por COVID-19, la cual ocasionó que una gran parte de la población realizara actividades laborales o académicas desde sus hogares [43] .

**Figura 2.1.**

*Incremento de usuarios de redes sociales por año en todo el mundo.*



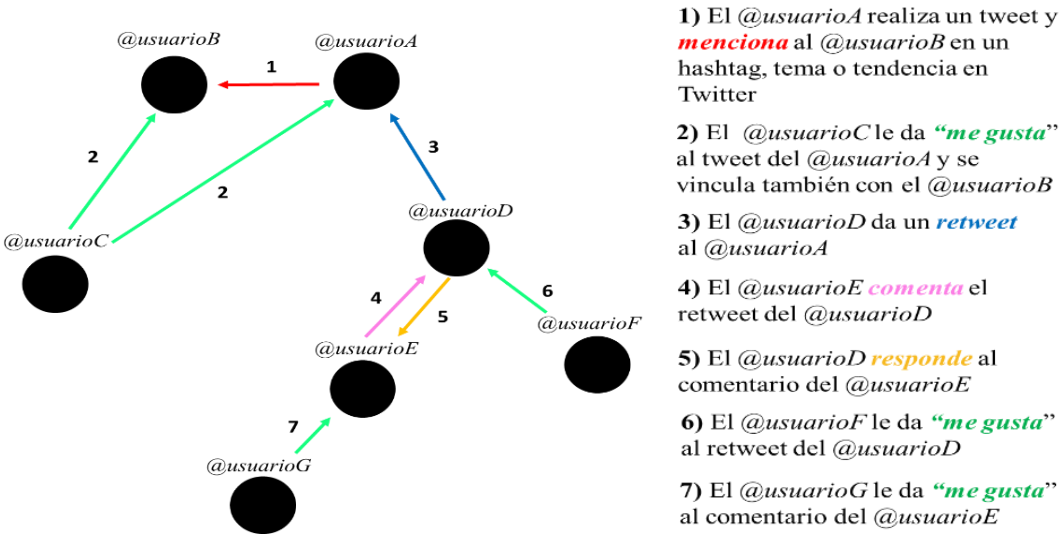

*Fuente*: *We Are Social & Hootsuite* (2024).

Esta nueva forma de interacción es un gran desafío para nuestra sociedad actual, las redes sociales están cambiando los esquemas de la actividad humana, incluso se ha concluido que la actividad derivada de la interacción digital es autónoma del horario solar [44].

Twitter (X.com) es una de las redes sociales más importante que se fundó en marzo del año 2006, tiene 35 oficinas en todo el mundo, cerca de 5,500 empleados [45], su API para el acceso de datos es una gran herramienta para la investigación de las interacciones digitales [46] [47]. Twitter permite generar cuatro tipos de interacción entre los usuarios ya sea en una sola dirección o en dos direcciones, si es de una sola dirección significa que el usuario interactuó mediante un mensaje, una reacción como *"me gusta"* o un retweet. Sin embargo, la interacción en dos direcciones se genera cuando hay una reciprocidad de mensajes, esto es un indicio de que hay una conversación digital, la figura 2.2 expone la estructura que pueden generarse entre los usuarios de Twitter [48].

**Figura 2.2.**

*Interacciones posibles en Twitter.*




Elaboración propia con datos de [47].

Una red compleja genera pocos nodos con muchas conexiones y muchos nodos con pocas conexiones, las redes complejas además cambian con el tiempo [49]. En Twitter se desarrollan tendencias continuamente, esto derivado de una gran interacción digital entre usuarios que conversan o interactúan con un cierto tema, tópico o hashtag (etiqueta).

## 2.2. Algoritmos de aprendizaje automático

Jhon McCarthy en 1956 definió la inteligencia artificial como: "*La ciencia e ingeniería para hacer maquinas inteligentes"*, además definió cuatro conceptos [50] :

- Sistema Experto: El problema será resuelto con un gran desempeño tal como lo haría un experto.
- Resolución de problemas: Evaluación de una pequeña gama de soluciones y decidir la más optima.
- Procesamiento natural del lenguaje: La máquina debe ser capaz de establecer una comunicación con los usuarios humanos de forma natural.
- Visión experta: Debe ser capaz de reconocer formas o características de cualquier objeto de manera automática.

Los algoritmos de aprendizaje automático también son designados como algoritmos "*aprendizaje artificial*", en términos comunes, se puede definir al aprendizaje como la forma en la cual un animal, maquina o un ser humano acrecienta el conocimiento y mejora sus acciones para realizar tareas o ciertas actividades; de tal manera que el aprendizaje automático se compone de dos procesos, primero selecciona las características más relevantes de un objeto o evento y después compara con datos conocidos si existieran, si encuentra diferencias significativas se adapta, un sistema de aprendizaje automático puede emplear algoritmos que requieren un gran poder de procesamiento computacional [51].

La clasificación es una de las tareas más significativas del aprendizaje automático que se dividen en dos tipos [52]:

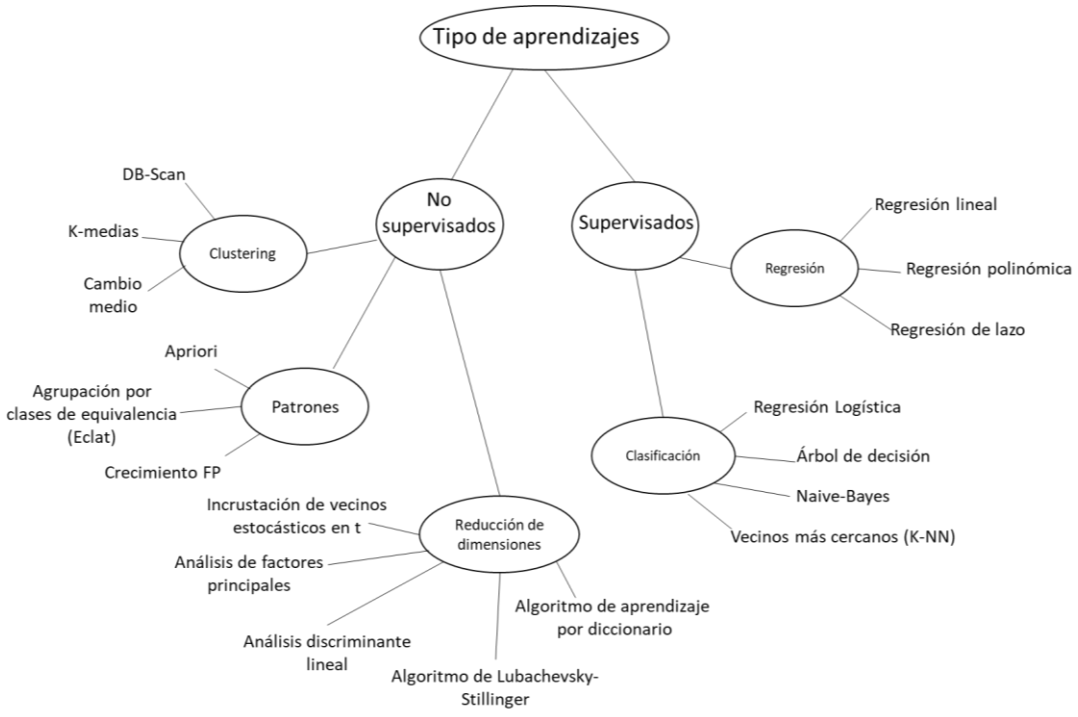
- Algoritmos de aprendizaje automático o "*machine learning"* que a su vez se subdividen en aprendizaje supervisado y el no supervisado.
- Algoritmos de lexicón, dentro de la cual se subdividen en algoritmos de clasificación por tipo de diccionario o de corpus.

La figura 2.3 nos revela los tipos y técnicas de los distintos algoritmos de aprendizaje automático, se puede ver que la técnica de aprendizaje supervisado se subdivide a su vez en algoritmos de clasificación y de regresión; existen otros tipos de algoritmos, como el algoritmo de vecinos más cercanos o "*K-NN*", el algoritmo de Naive-bayes o NB, o los algoritmos de

árbol de decisión y de regresión logística donde se encuentran los algoritmos de regresión lineal, polinómica y de lazo; en los algoritmos de aprendizaje no supervisado, se encuentran los algoritmos de "*clustering"* tales como "*DB-Scan*", "*K-medias*", "*Cambio medio*"; otro grupo son los algoritmos que buscan de patrones como los algoritmos "*A priori*", agrupación por clases (Eclat), de crecimiento o "*FP*", discriminante lineal, Lubachevsky-Stillinger, de incrustación de vecinos estocásticos, de factores principales, y el algoritmo por diccionario, son agrupados como algoritmos de reducción de dimensiones [53].

**Figura 2.3.**

*Mapa de algoritmos de aprendizaje automático.*




Elaboración propia.

Para llevar a cabo la presente investigación se consideró los siguientes algoritmos de aprendizaje automático.

### 2.2.1. Árbol de decisión (AD)

Este algoritmo utiliza representaciones visuales y se recomienda para clasificar, el método inicia con nodo semilla y se va descomponiendo en atributos, las bifurcaciones se realizan hasta las soluciones (hojas); esta técnica permite realizar clasificaciones y predicciones, además de la reducción de dimensiones para poder identificar las variables [54].

Para realizar la predicción de una nueva observación implica recorrer todo el árbol en función de los predictores hasta ubicar un nodo final, para el caso de regresión la predicción se determina por el promedio de las observaciones del conjunto de entrenamiento; la tabla 2.1 muestra un ejemplo donde se predicen el valor de “Y” para 10 observaciones (id), cada observación posee un valor de respuesta única para “Y” y sus posibles valores predictores para “X”.

**Tabla 2.1.**

*Asignación de observaciones para cada Id.*

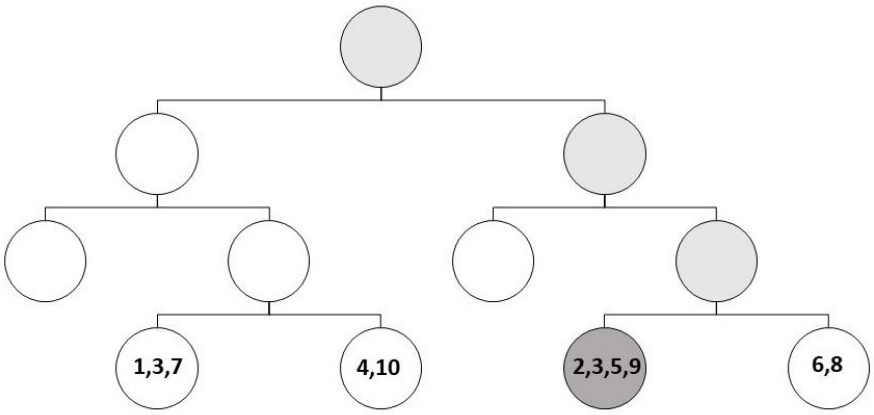
| id | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Y | 10 | 18 | 24 | 8 | 2 | 9 | 16 | 10 | 20 | 14 |
| X | … | … | … | … | … | … | … | … | … | … |

Elaboración propia.

Un ejemplo de la predicción para una observación se muestra en la figura 2.4, los nodos en gris nos muestran el camino de la predicción y gris oscuro el nodo final, de acuerdo con los índices de entrenamiento, el valor de la predicción es el promedio de “Y” de las observaciones con las id´s con valores de: 2,3,5 y 9.

**Figura 2.4.**

Ejemplo de *árbol de decisión.*




Elaboración propia.

La predicción de forma general para el algoritmo de "*árbol de decisión*" está definida por el promedio ponderado de las observaciones de los datos de entrenamiento, por el peso o ponderador "$w_i$" de cada observación depende, si forma parte del nodo final, el promedio ponderado queda de la siguiente forma:

$$\mu = \sum_{i=1}^{n} w_i Y_i. \quad (1)$$

El valor del vector de ponderación "w" es 1 para las observaciones del mismo nodo y un valor de 0 para los demás nodos, siguiendo el ejemplo anterior "w" quedaría definida como: w = (0,1,1,0,1,0,0,0,1,0).

Para que la suma de las observaciones sea igual a 1, se divide por el total de las observaciones que hay en el nodo final, en este caso son 4 observaciones que hay en el nodo final, por lo tanto, la ponderación seria:

$$w = (0, \frac{1}{4}, \frac{1}{4}, 0, \frac{1}{4}, 0,0,0, \frac{1}{4}, 0).$$

El valor final de la predicción quedaría de la siguiente forma:

$$\mu = \left[\begin{array}{c}(0 * 10) + \left(\frac{1}{4} * 18\right) + \left(\frac{1}{4} * 24\right) + (0 * 8) + \left(\frac{1}{4} * 2\right) + (0 * 9) + (0 * 16) + (0 * 10) + \\ \left(\frac{1}{4} * 20\right) + (0 * 14)\end{array}\right] = 16.$$

El resultado final se obtiene calculando el promedio simple y ponderado, sin embargo, el cálculo mediante la fórmula del promedio ponderado es de mayor utilidad que una presentación generalizada de la predicción [55].

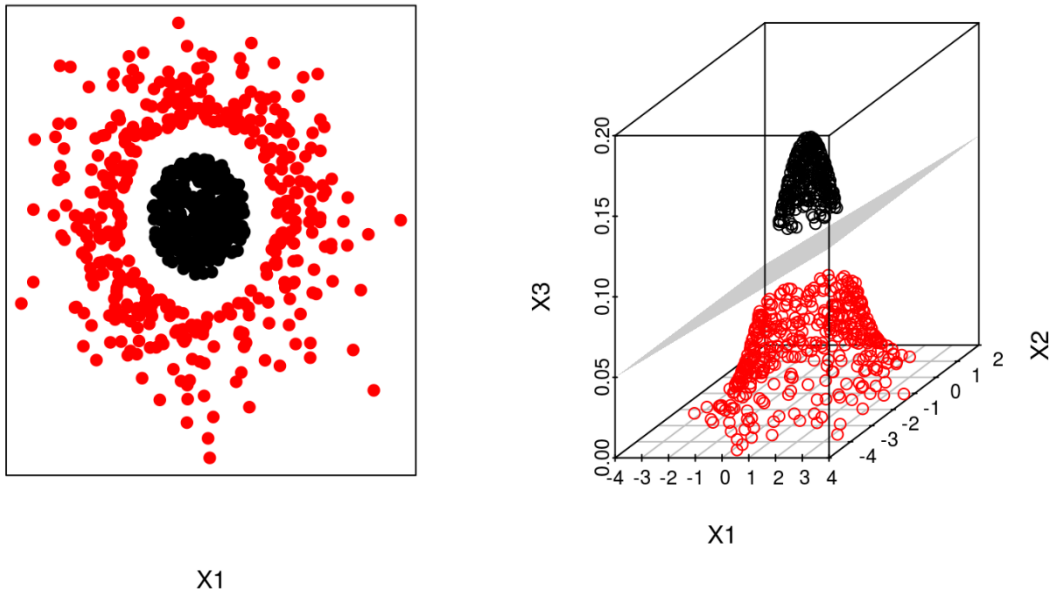
### 2.2.2. Algoritmo de Máquinas de Vectores de Soporte (MVS)

Este algoritmo se traduce como "*Support Vector Machine*" o SVM, es un algoritmo que pertenece al grupo de algoritmos de aprendizaje supervisado, también se denominan algoritmos no paramétricos, aunque si utilizan parámetros en su aprendizaje, se basa en la selección, tipificación y evaluación, a diferencia de la inferencia clásica, en los SVM los parámetros no se encuentran predeterminados y su cantidad está relacionada con los datos de entrenamiento [56].

Una variación de este tipo de algoritmos son los clasificadores marginales máximos o *"Maxim Margin Classifier"* (MMC) que tienen poca utilidad práctica, ya que solo se pueden utilizar cuando las categorías son linealmente divisibles y se utiliza más la clasificación por hiperplano, aunque esta variación no separe en su totalidad a las clases ofrece una mayor precisión de predicción cuando se aplica a nuevas observaciones, cuando no es posible separar dos clases, la alternativa es utilizar los SVM con una tercera clase o dimensión, en la figura 2.5 se puede observar cómo agregando una tercera dimensión se puede separar los datos, y llevar a cabo su clasificación de forma correcta.

**Figura 2.5.**

*Algoritmo de Maquina de vectores de soporte (MVS).*




*Fuente:* de [55].

Una alternativa para aumentar de dos dimensiones a tres dimensiones es aplicando la siguiente función:

$$f(x_1, x_2) = (x_1^2, \sqrt{2x_1x_2}, x_2^2) \quad \textbf{(2)}$$

La anterior fórmula es sola una alternativa, para una solución apropiada se debe utilizar la función de "*Kernel*" (*k*), que realiza el producto entre dos vectores, creando un nuevo espacio vectorial, hay una gran diversidad de funciones "*Kernel*", cada una tiene sus parámetros, cuyos valores óptimos se pueden establecer con la confirmación cruzada [57].

La función lineal de *Kernel* está dado por:

$$K = (x, x') = \ x * x'. \tag{3}$$

El Kernel polinómico esta dado por la siguiente función:

$$K = (x, x') = \ (x * x' + c \ )^{d}. \tag{4}$$

La forma Gaussiana del Kernel está dada por la siguiente función:

$$K = (x, x') = \exp(-\gamma \|x - x'\|^2). \tag{5}$$

.

### 2.2.3. Naive-Bayes (NB)

Este algoritmo es el más simple de una red bayesiana, donde todos los atributos son independientes del valor de clasificación, a esta propiedad se le conoce como independencia condicional [58].

Para utilizar el algoritmo de Naive-Bayes a la clasificación de textos, es indispensable conocer la posición de cada palabra dentro del texto, esto es de gran utilidad al momento de ir recorriendo el texto.

Para determinar la probabilidad de clase de un texto se emplea la siguiente fórmula:

$$C_{NB} = arg\max_{c \in C} P(c) \prod_{i \in pos} p(w_i|c). \tag{6}$$

donde $P(c)$ es la probabilidad "*a priori*" de la clase y $p(w_i|c)$ es la probabilidad del texto de pertenecer a una clase predeterminada.

Por ejemplo, si queremos realizar la detección o clasificación del sentimiento de un texto, primero se tiene que determinar la probabilidad de cada clase con la siguiente fórmula:

$$\hat{P}(c) = \frac{N_c}{N_{doc}} \tag{7}$$

donde: $N_c$ es la cantidad de documentos del conjunto de entrenamiento y $N_{doc}$ es la cantidad total de documentos.

La fórmula para determinar la probabilidad que tiene una palabra de pertenecer a un texto:

$$\hat{P}(w_i|c) = \frac{count(w_i,\ c) + 1}{\sum_{w \in V}\big(count(w, c)\big) + |V|} \tag{8}$$

donde $count(w_i,\ c)$ simboliza la frecuencia de cierta palabra en una clase, $\sum_{w \in V}\big(count(w, c)\big)$es la cantidad total de palabras en una clase y $|V|$ es la cantidad total de palabras de todas las clases predeterminadas.

En la tabla 2 podemos observar una muestra de textos del conjunto de entrenamiento, y en la fila final se ha denominado como "*prueba*", los textos de este ejemplo pertenecen a críticas realizadas a películas en idioma inglés, los textos de entrenamiento están catalogados como negativos (-) y positivos (+); la cantidad total de palabras de los textos clasificados como "*negativos*" es igual a 14, y la cantidad de las palabras dentro de la clase "*positivos*" es de 9. El conjunto total de palabras de entrenamiento es de 20, para este ejemplo solo se tomarán en cuenta el total de palabras del conjunto sin repetición, para el texto de prueba solo se contabilizaron 3 palabras diferentes y se retiró la palabra "*with*", ya que no está en los datos de entrenamiento [59].

**Tabla 2.2.**

*Ejercicio de clasificación de sentimiento* [60]

| Fase | Clase | Documento (texto) | Número de palabras |
|---|---|---|---|
| Entrenamiento | (-) | *just plain boring* | 3 |
| | (-) | *entirely predictable and lacks energy* | 5 |
| | (-) | *no surprises and very few laughts* | 6 |
| | (+) | *very powerful* | 2 |
| | (+) | *the most fun film of the summer* | 7 |
| Prueba | (¿?) | *predictable with no fun* | 3 |

Elaboración propia con datos de Jurafsky y J. H. Martin, “Speech and Language Processing,” 2019.

Para calcular la probabilidad de sentimiento negativo, primero se determinaron las probabilidades de cada una de las clases, para la clase con sentimiento negativo seria de la siguiente forma:

$$P(-) = \frac{N_{c(-)}}{N_{doc}} = \frac{3}{5}$$

para determinar la probabilidad del sentimiento positivo se realizó lo siguiente:

$$P(+) = \frac{N_{c(+)}}{N_{doc}} = \frac{2}{5}$$

después se calcularon las probabilidades por cada clase de la manera siguiente:

para la palabra “*predictable”:*

$$P("predictable"|(-)) = \frac{(1+1)}{(14+20)} = \frac{2}{34}$$

$$P("predictable"|(+)) = \frac{(0+1)}{(9+20)} = \frac{1}{29}$$

para la palabra “*no*”:

$$P("no"|(-)) = \frac{(1+1)}{(14+20)} = \frac{2}{34}$$

$$P("no"|(+)) = \frac{(0+1)}{(9+20)} = \frac{1}{29}$$

y para la palabra "*fun*":

$$P("fun"|(-)) = \frac{(0+1)}{(14+20)} = \frac{1}{34}$$

$$P("fun"|(+)) = \frac{(1+1)}{(9+20)} = \frac{2}{29}$$

finalmente se aplica el algoritmo de naive-bayes a todo el texto de prueba.

Sin la palabra "*with*" la probabilidad de que el texto para la clase de sentimiento negativo es:

$$P(-)P(S|(-)) = \left(\frac{3}{5}\right)\left[\left(\frac{2}{34}\right)\left(\frac{2}{34}\right)\left(\frac{1}{34}\right)\right] = 0.00610\%$$

y la probabilidad para la clase de sentimiento positivo es:

$$P(+)P(S|(+)) = \left(\frac{2}{5}\right)\left[\left(\frac{1}{29}\right)\left(\frac{1}{29}\right)\left(\frac{2}{29}\right)\right] = 0.00328\%$$

Dado que la probabilidad de pertenencia a la clase de sentimiento negativo es mayor que la probabilidad de pertenencia a la clase de sentimiento positivo, el texto "*predictable with no fun"* pertenece a la clase de sentimiento negativo.

Para una clasificación de mayor tamaño, se tiene que representar el texto con la técnica de bolsa de palabras (Bags of words), donde se asume que la disposición del orden de las palabras no es relevante y las clases solo se agrupan tomando en cuenta la identificación de cada palabra [60].

### 2.2.4. Regresión Logística (RL)

Este algoritmo tiene su fundamento en las técnicas de regresión logística de la ciencia estadística, y trabaja con clases binarias. Pero se puede utilizarse en la clasificación multiclase, mediante la técnica de "*uno contra*

*todos*" se entrena el mismo número de clasificadores binarios de acuerdo con las clases del conjunto, otra técnica de clasificación multiclase mediante regresión logística es conocida como "*uno por uno*".

El algoritmo de regresión logística es una representación alternativa con la diferencia que el modelo de regresión logística usa la función sigmoidea en lugar de una función lineal [61].

El algoritmo de regresión logística es capaz de resolver la probabilidad de solución para $Y = 1$ en función de "n" variables tales como $X_1$, $X_1$ …$X_n$ , que pueden ser variables continuas, discretas, ordinales, nominales o dicotómicas, además determina los coeficientes $\beta_0$, $\beta_1$ … $\beta_n$ que mejor convengan a la fórmula [62]:

$$P(Y = 1) = \frac{1}{1 + \exp(-\beta_0 - \beta_1 X_1 - \ \ldots - \beta_n X_n)}. \tag{9}$$

La fórmula anterior describe la regresión logística, pero también se pueden utilizar las siguientes fórmulas:

$$P(Y = 1|x) = \frac{e^{wx+b}}{1 + e^{wx+b}} \tag{10}$$

$$P(Y = 0|x) = \frac{1}{1 + e^{wx+b}} \tag{11}$$

donde "*w*" nos indica el peso o ponderación de la variable, "b" el sesgo de la variable, ambos modelos el de regresión logística y lineal son modelos estándar, la diferencia es que el algoritmo de regresión logística introduce la función sigmoidea (no lineal) [63].

### 2.2.5. Pasivo-Agresivo (PA)

El algoritmo pasivo-agresivo posee una gran capacidad de margen para el ranking "*en línea"*, la denominación de este algoritmo se debe a que si la predicción es apropiada la mantiene y no realiza ningún cambio, y es agresivo porque si la predicción es equivocada realiza ajustes en la representación, este algoritmo se fundamenta en que, algún cambio en la representación podrá realizar la predicción correcta [64].

El algoritmo Pasivo-Agresivo determina los pesos de las variables de la función de regresión con una perdida cercana a cero, con la finalidad de que el vector de ponderación (pesos) sea similar al anterior, el algoritmo adecua cada factor de "λ", por lo que $\lambda_t$ esta dado por:

$$\lambda_t = \phi(\hat{y}) \frac{\sqrt{l(\hat{y})} - \lambda_{t-1}\ \phi(\hat{y})}{\|\phi(\hat{y})\|^2 + \frac{1}{C}} \tag{12}$$

donde *C* representa el factor de agresividad; $l(\hat{y})$ representa el diferencial entre la calidad de la traducción y la hipótesis del sistema.

$\phi(\hat{y})$ es la diferencia de las características de $y^*$ y la traducción de $\hat{y}$ definida por:

$$\phi(\hat{y}) = h(\ x\ , y^*) - h(\ x\ , \hat{y}\ ) \tag{13}$$

Cuando no se cumpla las condiciones anteriores se realiza la actualización con lo siguiente fórmula:

$$\lambda_{t-1}\ \phi(\hat{y}) \geq \sqrt{l(\hat{y})} \tag{14}$$

### 2.2.6. Multicapa Perceptrón

En las redes neuronales artificiales, los nodos (neuronas) se interconectan mediante la sinapsis que son conexiones direccionales, los nodos suelen agruparse en unidades llamadas capas, el conjunto de estas capas es lo que se denomina red neuronal.

El concepto perceptrón fue introducido por Rosenblatt [65], es un modelo compuesto por dos capas de redes neuronales, su función es la siguiente:

$$y = f(\sum_{j=1}^{n} w_{ij}x_j - \theta_i) \tag{15}$$

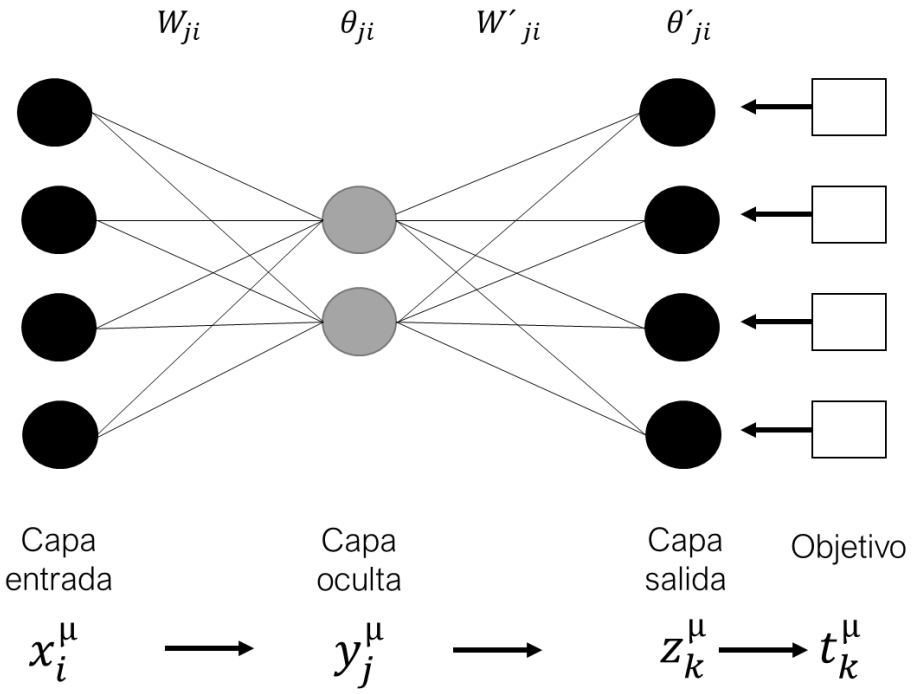
donde $x_j$ es un conjunto de entradas, $w_{ij}$ son los pesos sinápticos con $j = 1, \dots, n$ y $\theta_i$ es un parámetro adicional denominado umbral.

Las neuronas o nodos de la capa de entrada son variables discretas, mientras que la capa final es una función escalón, el algoritmo básico del perceptrón se puede emplear para categorizar y determinar los pesos sinápticos para clasificar los patrones del conjunto cuyos datos hayan sido previamente clasificados. El perceptrón básico tiene limitaciones por la diferencia de patrones que pueden ser separados por el hiperplano, y es determinado por las neuronas de la capa de entrada, una forma de resolver esta limitación es utilizando capas ocultas (figura 2.6), logrando una red neuronal denominada “perceptrón multicapa” (fórmula 16) [66].

$$Z_k = \sum_{j=1}^{O} w'_{kj}y_j - \theta'_k = \sum_{j=1}^{O} w'_{kj}f - (\sum_{j=1}^{O} w'_{ji}x_i - \theta_j) - \theta'_k \tag{16}$$

**Figura 2. 6.**

*Algoritmo perceptrón multicapa (MLP).*




Elaboración propia a partir de [65].

## 2.3. Análisis de redes sociales

El estudio de las redes se ha vuelto un indispensable en varios campos tales como la biología, telecomunicaciones, sociología o la informática, recientemente se ha utilizado en los campos del análisis del lenguaje. Las redes sociales se fundamentan en conceptos de los sistemas complejos. Barabasi expuso de forma integral diversos fundamentos, teorías y aplicaciones de las redes y estableció los pilares de la nueva ciencia de redes [67].

El análisis de redes ha sido útil para estudiar la operación de grupos de terroristas o criminales, los cuales pueden formar grupos autónomos, pero que actúan con una distribución de red compleja, este tipo de organización no posee una autoridad central y utilizan la red de internet como herramienta difusora de contenidos de su ideología [68]. El estudio de situaciones complejas mediante el análisis de redes es tan relevante que, incluso existe un comité para futuras aplicaciones del ejército de los Estados Unidos, donde se aborda su aplicación en diversos casos de estudio, como la respuesta del gobierno a un ataque biológico, este comité clasificó en tres tipos las áreas de su interés:

- En la biología se utiliza para analizar la transmisión de virus o enfermedades en las regiones, pueblos o ciudades.
- Las redes físicas se enfocan es analizar las topologías y dinámicas de las redes de logística, transporte y telecomunicaciones.
- Las redes sociales, en esta área se estudia el comportamiento de las personas, y la forma de interacción de los grupos sociales en línea, o el intercambio de información.

El comité militar también definió diversos aspectos de aplicaciones de las redes en diversos contextos, por ejemplo, es muy probable que un ingeniero al mencionarle la palabra "*redes*" piense en un circuito eléctrico o en un sistema de radiotelecomunicación, un sociólogo en las redes de influencia social, y un empresario en las redes de comercio o en las redes que se crean por las interacciones humanas que posibilitan el funcionamiento de las empresas [69].

En idioma inglés las redes sociales se denominan como "*social media",* este concepto ha ido evolucionando a la par de la propia red de internet, una de las primeras aplicaciones fue la plataforma "*UseNet*" que se implementó en 1979 por Tom Trucott y Jim Ellis, este sistema era capaz de realizar intercambios de mensajes de texto y algunos tipos de archivos entre dos computadoras mediante el protocolo de comunicación de UNIX-to-UNIX (UUCP), este sistema progresó hasta crear pequeñas comunidades difusoras

de noticias, años después se desarrolló "*open diary*" que se fundó en el 1998 por Bruce Ableson y Susan Ableson, fue el sistema precursor de las redes sociales 2.0, esta plataforma permitía el trabajo colaborativo de los usuarios en línea, donde podían escribir y crear contenido propio y subirlo a internet con gran facilidad, este fue el inicio de los "*weblogs*" o blogs [70].

En el año 2004, se fundó Facebook y en el 2006 Twitter (X.com), ambas plataformas siguen siendo de las más populares a nivel global. Aunque popularmente son conocidas como redes sociales, tienen características diferentes, por un lado Facebook cumple con las características de una red social, ya que permite a los usuarios tener a sus amigos y crear una comunidad o grupos sociales, por otro lado, en Twitter es más un *microblog* de contenido abierto a casi cualquier usuario, la información o contenido que se difunde puede ser leído por usuarios conocidos o desconocidos, esto hace que la información alcance a más personas y ciertos casos a distintos grupos sociales [71].

En años recientes la interconexión digital ha crecido de forma acelerada, ya sea entre personas, procesos o sistemas mediante la red de internet y apoyados por microprocesadores muy potentes, capaces de ejecutar diversos programas de inteligencia artificial; esto ha originado el concepto de la industria 4.0 [72], donde las plataformas de interacción digital como Facebook o Twitter son capaces de estructurar sistemas complejos, creando vínculos de interacción entre los usuarios (nodos), estas estructuras complejas pueden generar redes de varios grados de interacción [73].

Para analizar las redes sociales hay un conjunto de métricas que pueden subdividirse de la manera siguiente:

- Topología. Dentro de este grupo de métricas se encuentran el grado de la red, el "*clustering"* y el coeficiente de modularidad.
- Influencia. Aquí se encuentran las métricas que miden la cercanía entre los nodos de una red, la transitividad, el coeficiente de intermediación, asi como los índices de "*eigenvector"* y de "*pagerank".*

- Potencia viral. Las métricas de este grupo miden la interconexión entre los usuarios (nodos) esto se logra calculando la distancia, camino o ruta promedio entre dos nodos de una red, asi como el diámetro de la red.

A continuación, se presentan las métricas que esta investigación utilizó para el desarrollo del análisis de redes.

*Grado (Degree):* esta métrica determina los enlaces o vínculos que tiene un nodo dentro de la red, los enlaces pueden ser pueden entrada, esto se origina cuando le envían un mensaje o comentario a un usuario o de salida que se origina cuando el usuario emite un mensaje o comentario hacia otro usuario; si la red que se analiza contiene nodos con un alto grado ya sea de entrada o de salida significa que pertenece a una red compleja con una distribución de libre escala, este tipo de distribución tiene diferencias significativas con una distribución normal o aleatoria. En las redes sociales se pueden desarrollar enlaces o vínculos bidireccionales o también llamados simétricos, esto puede suceder cuando un usuario realiza menciones o comentarios hacia otro usuario, y a su vez recibe respuestas, comentarios u otras interacciones como los "*me gusta*" o *retweets*; tal como se ha mencionado, estas redes generan sistemas o redes complejas que pueden representarse con la siguiente función de libre escala:

$$\mathcal{P}_k \sim k^{-\gamma} \tag{17}$$

donde $\mathcal{P}_k$ es la probabilidad de grado del nodo en una red y *k* representa el grado o los vínculos de los nodos, el coeficiente de la ecuación esta dado por "$\gamma$", una red de libre escala tendrá un coeficiente "$\gamma$" entre un valor de 2 y 3 [74].

*Clustering*: es un coeficiente que evalúa los enlaces triangulares y la conexión de un nodo con otros nodos vecinos, para las redes sociales este concepto se utiliza para cuantificar las conexiones entre los amigos de un

usuario. El *clustering* para un nodo dentro de una red está dado por la siguiente fórmula:

$$C_i = \frac{\lambda_{G(v)}}{\tau_{G(v)}} \tag{18}$$

donde $\lambda_{G(v)}$ simboliza la cantidad de subgrafos con tres vínculos o enlaces y tres vértices y la variable $\tau_{G(v)}$ es la cantidad de conexiones llamadas tripletes sobre el vértice del nodo.

Si deseamos evaluar las posibles conexiones entre todos los nodos de una red, se puede determinar el promedio de "*clustering"* con la siguiente fórmula:

$$\bar{C} = \frac{1}{n}\sum_{i=1}^{n} C_i \tag{19}$$

donde la variable *n* es la cantidad de vértices en una red, y el termino $C_i$ es el coeficiente de *clustering* del nodo "*i"* [75].

*Modularidad*: este coeficiente evalúa la potencia de separación de los subgrupos dentro de una red, su utilidad principal es para detectar las comunidades, su operación está basada un arreglo de tipo matricial que se puede determinar con la siguiente fórmula:

$$Q = \frac{1}{4m} S^T B_S \tag{20}$$

donde S es la variable que representa el vector de la columna cuyos elementos son $S_i$ y la matriz simétrica real "$B_S$" que a su vez se determina por $B_S = A_{ij} - \frac{k_i k_j}{2m}$, la cual también se denomina matriz de modularidad [76].

Las métricas de red nos ofrecen en conjunto una evaluación de la estructura de la red, además se puede determinar la influencia de los usuarios o nodos relevantes dentro de una red [77].

*Cercanía (closeness):* esta métrica calcula la distancia más corta de un nodo a otro nodo En las redes sociales la cercanía es una métrica que evalúa la accesibilidad sobre un usuario o nodo en la red, en las redes de información, esta métrica representa la rapidez de difusión y puede determinarse mediante la siguiente fórmula.

$$C_{clo} = \frac{1}{\sum_{j=1}^{n}(S)_{ij}} \tag{21}$$

donde *S* representa la matriz de distancias o longitudes entre los usuarios o nodos de una red.

*Transitividad*: está definida por el número de triángulos posibles de una red dividido entre la cantidad de tripletas conectados, en el análisis de redes, esta métrica evalúa la existencia de comunidades con una fuerte conexión, se ha determinado que las redes complejas tienen una alta transitividad [78], esta métrica se puede calcular con la siguiente fórmula.

$$\tau = \frac{3 * Número\ de\ triangulos}{Número\ de\ tripletas}\ . \tag{22}$$

*Intermediación (betweenness):* este concepto fue desarrollado por Freeman [79] para evaluar el control que puede tener una persona en una red social, la idea principal de este algoritmo es elegir aleatoriamente dos nodos, y luego una de las posibles rutas más cortos entre los nodos, de tal manera que los nodos con mayor valor serán los que tengan una mayor probabilidad dentro de la ruta, para determinar la intermediación de los nodos se utiliza la siguiente fórmula.

$$C_B = \sum_{j,k} \frac{b_{jik}}{bjk} \tag{23}$$

donde $bjk$ es la cantidad de rutas más cortas desde el nodo "*j*" hasta el nodo "*k*" y $b_{jik}$ es la cantidad de rutas más cortas desde "*j*" hasta "*k*" que pasan a través del nodo "*i*".

Para calcular el alcance viral de una red se utilizó para esta investigación las siguientes métricas.

- Diámetro de la red, representa el recorrido máximo promedio entre dos nodos de una red.
- Longitud promedio de la ruta más corta, esta métrica evalúa la distancia o grados promedio de separación entre dos nodos en una red y se pude determinar con la siguiente fórmula:

$$l_{PR} = \frac{2}{n(n-1)} \sum d_{ij} \tag{26}$$

donde "*n*" es la cantidad total de nodos en una red y $d_{ij}$ representa las trayectorias entre los nodos que componen la red [80].

## 2.4. Potencia Viral

El modelo SIR ((Susceptible-Infectado-Recuperado) fue desarrollado para estudiar la propagación de elementos virales biólogos fue desarrollado hace más de un siglo, para este modelo, Weiss [81] definió cinco mecanismos a considerar en el modelo SIR, el primero es la transmisibilidad entre los miembros de una población, el segundo está relacionado con los cambios en el comportamiento de los virus, el tercero es sobre la heterogeneidad de la población, el cuarto es la capacidad de mutación y el quinto es sobre la disminución de la capacidad inmunológica de la población a la infección.

Ciertos contenidos en las redes sociales pueden generar tendencias o "*trendings topics"* que son producto de la viralización, Nazim [82] determinó que la difusión viral de algún contenido se origina desde una fuente hacia otras fuentes mediante internet. y que esta propagación puede ser *on-line* u *off-line*, además puede ser positiva o negativa; un tipo de contenidos muy popular en las redes sociales son los memes, este concepto fue desarrollado por Dawkins [83], quien abordó la evolución y adaptación de la cultura humana a través del tiempo; otros como Wang et al, [84] determinaron mediante el modelo SIR que los memes son altamente virales.

Goel et al, [85] desarrollaron un método para el análisis de la estructura de redes, para explicar la propagación viral a treves de las cascadas de difusión, y determinaron que las imágenes y los videos en Twitter son más populares que los contenidos de noticias, mientras que Weng et al, [73] estudiaron la propagación de los memes y determinaron que la viralización de un meme depende más de la estructura de las comunidades que van adquiriendo el meme, en el área del marketing viral también se ha desarrollado diversas investigaciones enfocadas a la calidad del mensaje, tal como Yuping [86], quien determinó que la calidad del mensaje influye más que tener una red de semillas (*Influencers*) considerables.

Las redes del mundo pequeño son un intermedio entre las redes regulares y aleatorias, tal como lo señala Ch'ng [87]. quien ha determinado diversas características de las redes del mundo pequeño, entre ellas que tienen una longitud de ruta más corta y un alto coeficiente de agrupamiento, además presentan una dinámica de autoorganizadas, a este tipo de redes también se les conoce como redes del mundo real.

Dentro de las redes del mundo pequeño, el coeficiente sigma nos indica que, si es mayor a uno pertenece a una red del mundo pequeño [88], las cuales son propicias para la infección viral, en este caso de contenidos o información en redes como Twitter [89].

## 2.5 Evaluación de datasets

Esta investigación evaluará los datasets de noticias falsas con los siguientes algoritmos de aprendizaje automático con las siguientes métricas de evaluación [90]: precisión (precisión), recuperación (recall), F1-score, y exactitud (accuracy) en los datasets de noticias falsas, para determinar que algoritmo y que dataset es el más adecuado para desarrollar el modelo para detectar en línea las noticias falsas y también para detectar cuentas bot en Twitter (X.com).

### 2.5.1 Precisión

Esta métrica calcula la cantidad de solicitudes correctas recuperadas dividida por todas solicitudes recuperadas.

$$Precision: P = \frac{TP}{TP + FP} \tag{27}$$

donde TP se refiere a los positivos verdaderos (True Positive) y FP (False Positive) a los falsos positivos.

### 2.5.2 Recall (Recuperación)

Esta métrica mide la cantidad de solicitudes correctas recuperadas divididas por todas las solicitudes correctas.

$$Recall: R = \frac{TP}{TP + FN} \tag{28}$$

donde TP (True Positive) se refiere a los positivos verdaderos y FN (False Negative) a los falsos negativos.

### 2.5.3 F1-score

Esta métrica determina el promedio ponderado de la precisión y la recuperación, según la función de ponderación, y puede tener varios índices que otorgan diferentes pesos a la precisión y recuperación.

$$F - score = F_\beta = (1 + \beta^2) * \frac{P * R}{\beta^2 * P + R} \quad (29)$$

Con β=1 se obtiene la puntuación estándar dando como resultado:

$$F - score = F_1 = F = 2 * \frac{P * R}{P + R} \quad (30)$$

### 2.5.4. Exactitud (Accuracy)

Esta métrica determina la proporción de las solicitudes verdaderas recuperadas, tanto positivas como negativas, entre todas las solicitudes recuperadas, y es una media aritmética que pondera la precisión y la precisión inversa tal como se muestra en la fórmula 31.

$$Accuracy: A = \frac{TP + TN}{TP + TN + FP + FN} \quad (31)$$

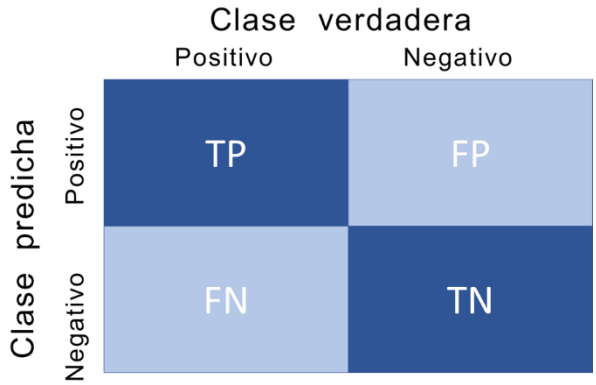
donde TP se refiere a los positivos verdaderos, TN a los verdaderos negativos, FP a los falsos positivos y FN a los falsos negativos.

### 2.5.5. Matriz de confusión

Esta es una herramienta muy útil para calcular las predicciones de los algoritmos de aprendizaje automático, además de las predicciones correctas e incorrectas generadas por un clasificador [91]. Su estructura visual es de muy fácil lectura, nos ofrece las medidas de los positivos verdaderos (TP), los verdaderos negativos (TN), asi como los falsos positivos (FP) y los falsos negativos (FN) de una manera integral, Donde el valor de TP es la probabilidad de que el valor positivo haya sido predicho como positivo, mientras que el valor de TN es la probabilidad de que el valor negativo haya sido predicho como valor negativo, y el valor de FP es la probabilidad de que el valor haya sido predicho como positivo cuando es negativo, también se le conoce como error tipo I, por último, el valor de FN se refiere a la al valor de la predicción como negativa cuando es falso, tal como se observa en la figura 2.7.

**Figura 2.7.**

*Organización de una matriz de confusión.*




Elaboración propia.

# CAPÍTULO 3

# METODOLOGÍA

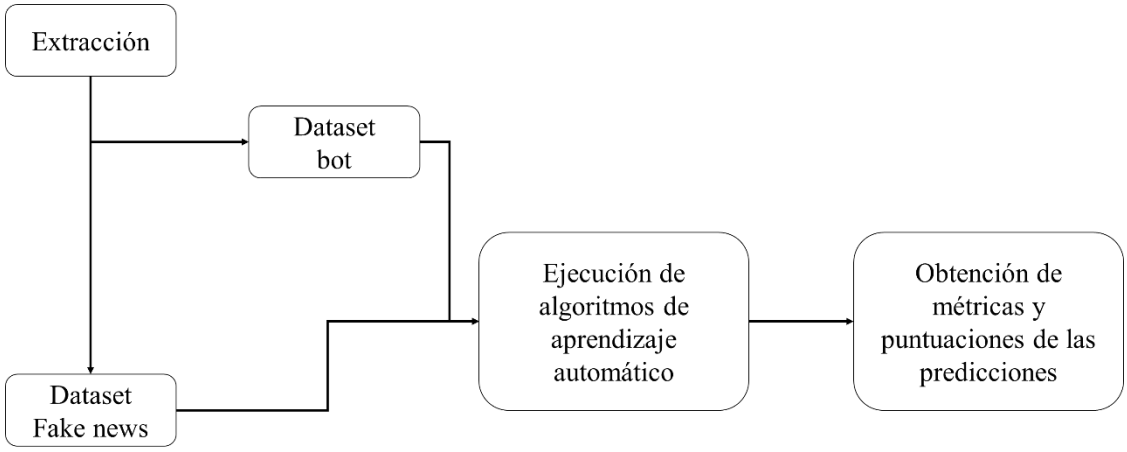
## 3.1. Descripción

En la sección 1.3 se mostraron los conceptos teóricos de diversos algoritmos de aprendizaje automático, tales como el de árbol de decisión (AD), máquinas de soporte de vectores (MSV), Naive-bayes (NB) y regresión logística (RL), además se mostró el concepto teórico de un algoritmo de redes neuronales denominado "*multicapa-perceptrón*" para la detección automática de noticias falsas y cuentas bot.

Para lograr el objetivo planteado en la sección 1.5.1 se diseñó la siguiente metodología: primero la extracción de datos mediante la API de Twitter, en segundo lugar el desarrollo de dos datasets, uno para detectar noticias falsas o *fake news* y otro para detectar cuentas bot, y en tercer lugar la ejecución de diversos algoritmos de aprendizaje automático, mediante la plataforma Spyder-Python para poder obtener las métricas y puntuaciones de las predicciones de los distintos algoritmos en los distintos datasets propuestos (Figura 3.1).

**Figura 3.1.**

*Modelo para la detección automática de noticias falsas en idioma español.*




Elaboración propia.

La presente investigación se propuso crear un modelo para detectar las noticias falsas que se difunden en las redes sociales, para conseguir esto primero se experimentara un dataset de contenidos elaborado por la iniciativa constraint@2021, después se probara con un segundo dataset con tweets relacionados con el COVID-19, y por último con un tercer dataset desarrollado por la iniciativa IberLef 2021, el cual contiene noticias o contenido falso divulgado en medios digitales.

El modelo plantea el análisis de los tweets y las características del usuario.

- Tweet: Los tweets pueden estar compuestos, por imágenes, videos, gifs, enlaces, hashtags y el más común es el texto, para esta investigación se plantea utilizar el texto del tweet.
- Usuario: Los usuarios de Twitter tienen características que pueden ser públicas como el número de seguidores, amigos o usuarios a los que están siguiendo, la fecha de creación del usuario, además de la cantidad de tweets y “me gustas” realizados y el número de listas a las

que pertenecen, además se determinara si la red de amigos del usuario conforma una red con potencial viral.

## 3.2. Variables

Esta investigación ha definido dentro de los objetivos específicos (sección 1.5.2) la creación de un nuevo dataset de noticias falsas, para lo cual se desarrolló una aplicación para extraer los tweets emitidos sobre el tema de las vacunas para el COVID-19; los tweets extraídos poseen tres tipos de variables que se tomaron como independientes, y cada una se relacionaron con variables dependientes.

Las variables independientes derivadas de los tweets o contenidos noticiosos que planteo esta investigación son:

a) Texto.

b) @usuario.

c) Red de amigos de amigos.

Las variables dependientes para el texto son:

a) Noticia real (Real).

b) Noticia falsa (Falso).

Las variables dependientes para el @usuario serán:

a) Bot.

b) Humana.

Las variables dependientes de la red de amigos serán:

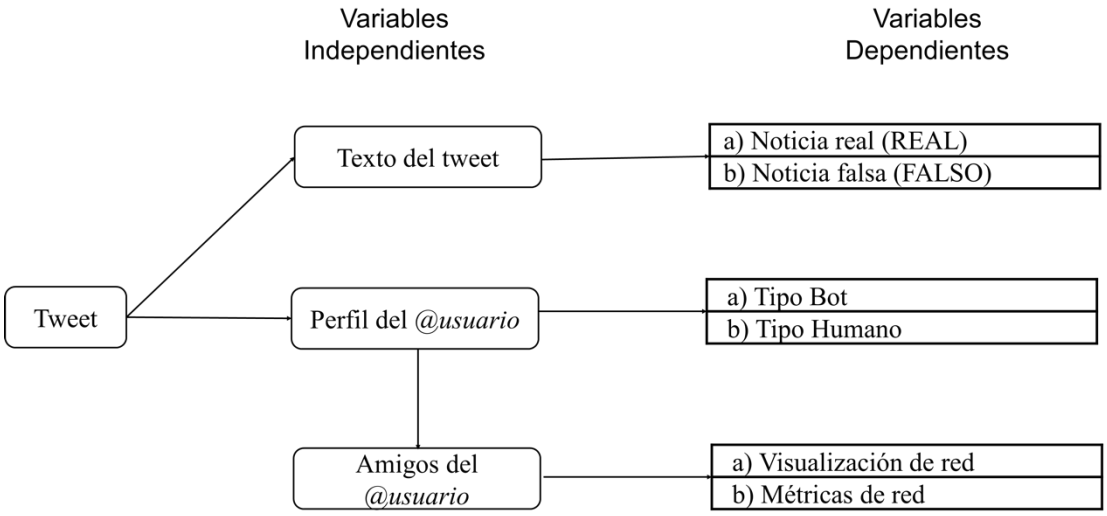
a) Visualización.

b) Métricas de red.

Las variables independientes y las dependientes se presentan en la figura 3.2, donde podemos observar que el análisis del texto dentro del modelo de detección de aprendizaje automático. Además, el modelo propuesto determina por las características de carácter público del usuario, si es un usuario o cuenta es tipo bot o humana, y por último evalúa la potencia viral con el análisis de la red de amigos de amigos del usuario.

**Figura 3.2.**

*Variables para el modelo detección de noticias falsas, cuentas bot y potencial viral.*




Elaboración propia.

## 3.3. Datasets de noticias falsas

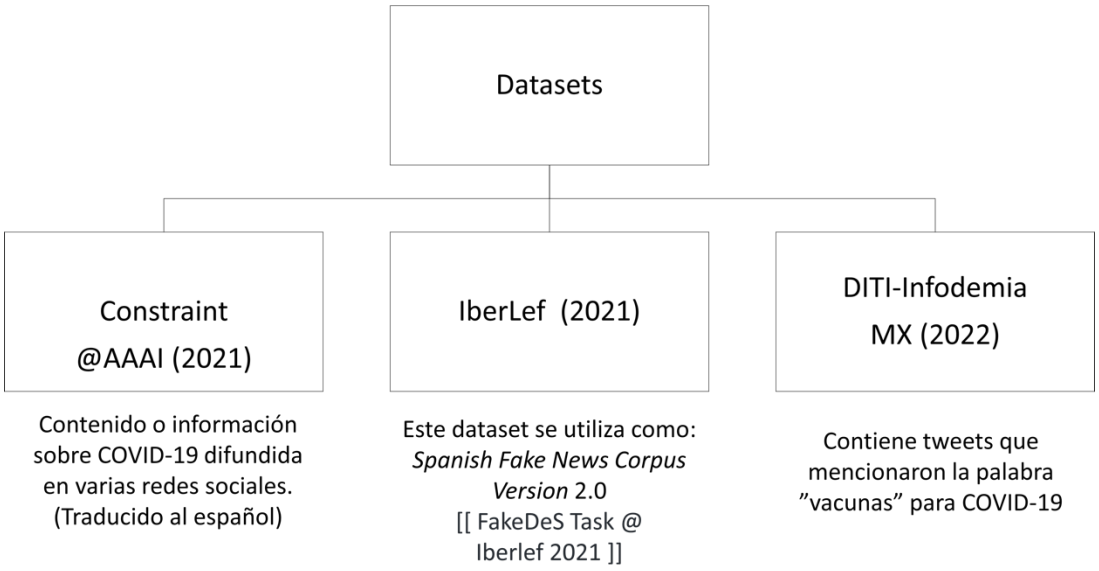
Para la prueba de algoritmos se propuso utilizar tres datasets; el primer dataset es el desarrollado por la iniciativa Constraint@AAAI2021, el cual contiene 6,420 publicaciones con información falsa o real sobre el COVID-19 que fueron expuestos en diversas redes sociales como Twitter, Facebook e Instagram [92].

El segundo dataset contiene en total 1,248 publicaciones y noticias en varias áreas como política, deportes o espectáculos, fue desarrollado por Juan P. Posadas Durán y presentado en el foro de evaluación para lenguajes ibéricos conocido como "*IberLef*" (Iberian Languages Evaluation Forum), este foro es una iniciativa para fomentar el estudio sobre el procesamiento del lenguaje natural, y cuenta con la asistencia de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN) [93].

El tercer dataset contiene 638 tweets clasificados manualmente, el cual fue desarrollado dentro de esta investigación en conjunto con la iniciativa Infodemia MX que es un programa impulsado por el Sistema de Radiodifusión Pública (SPR) del estado mexicano, los tweets fueron extraídos entre el 25 y 27 de enero del año 2022, en la figura 3.3 se observan los tres datasets: Constraint, IberLef e Infodemia Mx que integraron la fuente de entrenamiento y prueba, de los algoritmos de aprendizaje automático para el desarrollo de esta investigación.

**Figura 3.3.**

*Datasets de noticias falsas en español.*

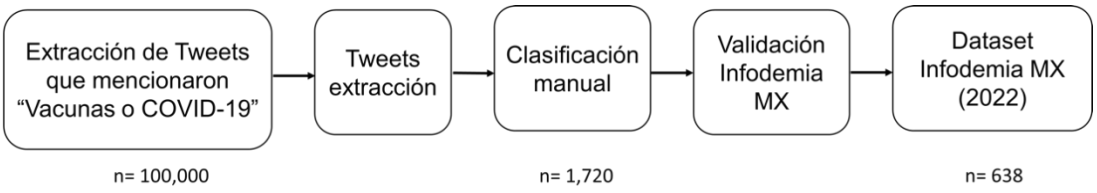



Elaboración propia.

El dataset desarrollado por la presente investigación fue denominado DITI-Infodemia MX, cada uno de los tweets seleccionados fue leído dentro del contexto del COVID-19 y se verifico su contenido; en la figura 3.4 se observa el proceso que se llevó a cabo para la clasificación y validación del dataset, primero se realizó una pre-clasificación por alumnos pertenecientes a la Facultad de Ingeniería Mecánica y Eléctrica (FIME), de la Universidad Autónoma de Nuevo León (UANL). Para llevar a cabo dicha tarea se les dio una capacitación a los alumnos colaboradores sobre la clasificación de noticias falsas, después se enviaron los datos a la iniciativa Infodemia.MX para una validación independiente quedando 638 tweets clasificados en noticia falsa o noticia real.

**Figura 3.4.**

*Proceso de validación del dataset DITI-Infodemia MX.*




Elaboración propia

La tabla 3 muestra algunos ejemplos de las clasificaciones para el dataset de DITI-Infodemia Mx, la clasificación tiene solo dos categorías: noticia real (Real) o noticia falsa (Falso), se puede ver una noticia falsa como la siguiente: "*Tiene razón,conozco a muchos vacunados que se han comprado la freidora esa sin aceite!...y otros la termomix…hipnotizan a la peña con las vacunas para q compren electrodomésticos…???? Eric Clápton machote,psicólogo?"*

**Tabla 3.1.**

*Muestra de clasificación manual.*

| Texto | Clasificación |
|---|---|
| Tiene razón,conozco a muchos vacunados que se han comprado la freidora esa sin aceite!...y otros la termomix…hipnotizan a la peña con las vacunas para q compren electrodomésticos…???? Eric Clápton machote,psicólogo? | Falso |
| ¿En que cueva habitas tu como ermitaño? ¿No te has enterado que en #Cuba se ha vacunado 93% d la población mayor de 2 años y ahora recibimos la 4ta dosis de refuerzo con vacunas cubanas? Despierta amigo ¿o es que eres un bots? #CubaVive #CubaPorLaVida #CubaEsJusta #DeZurdaTeam https://t.co/PqH198EsfA | Real |

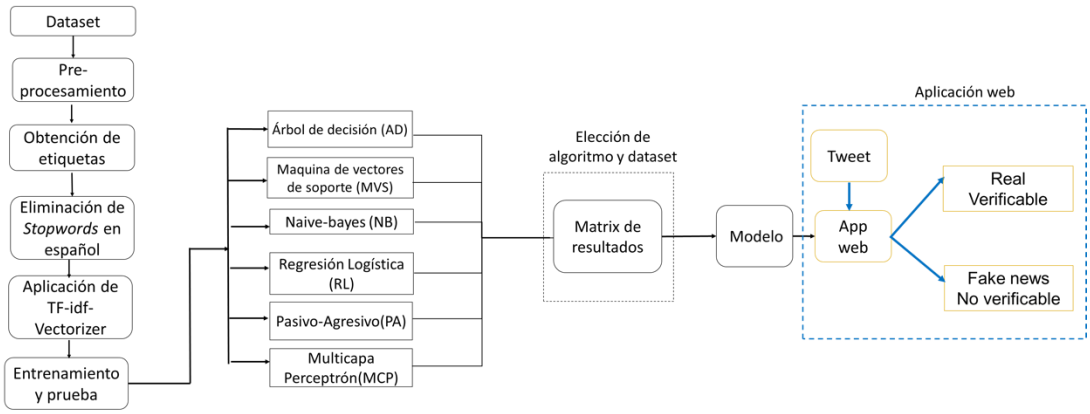
Elaboración propia.

## 3.4. Modelo de aprendizaje automático para la detección de noticias falsas

Para el sistema de detección de noticias falsas en línea se propuso el siguiente modelo que se puede observar en la figura 3.5, como primer paso se da entrada a los textos del tweets, después se filtraron las letras y símbolos o caracteres especiales, después se obtuvieron las etiquetas de la clasificación manual (Real o Falso), también se eliminaron las palabras de parada o "*stopwords*", a continuación se empleó la técnica de vectorización de texto para constituir dos subconjuntos, el de entrenamiento y el de prueba, para cada uno de los algoritmos de aprendizaje automático propuestos, se evaluaron las métricas de rendimiento y su matriz de confusión, para después comparar los resultados, el modelo completo se desarrollará en el capítulo 4.

**Figura 3.5.**

*Modelo para el desarrollo de la aplicación para la detección de noticias falsas en idioma español.*




Elaboración propia.

## 3.5. Dataset bot COVID-19

En la sección 3.2 se precisan las variables de la presente investigación, una de las variables independientes es el perfil del usuario, para lo cual fue necesario desarrollar un nuevo dataset de usuarios que fueron clasificadas como bot, esto con la finalidad de entrenar un modelo de detección automática de usuarios o cuentas bot en Twitter (X.com) [94] que solo utilice la información o características de carácter público de un usuario tales como:

- Seguidores (Son usuarios que siguen a otro usuario, también se le conocen como "*fans"*).
- Siguiendo (También se conocen como los amigos del usuario).
- Favoritos (Son la cantidad de reacciones conocidas también como "me gusta" y tienen el icono de un corazón).
- Tweets (Son los tweets emitidos por el usuario).
- Listas (Es una herramienta para organizar los contenidos del usuario).

También se pueden determinar otras características complementarias que se enuncian a continuación:

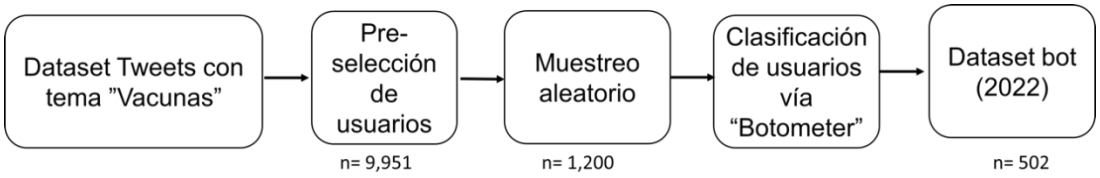
- Tiempo o antigüedad de cuenta (Esta métrica se calcula restando de la fecha actual a la fecha de creación de la cuenta).
- Calidad de cuenta (Esta métrica nos indica la relación entre seguidores y seguidos, algunos investigadores han determinado que el 60% de los bots tienen una puntuación por debajo de 0.5 en esta métrica [95].
- Tweets por día (Se divide la cantidad de tweets entre el tiempo de la cuenta).
- Favoritos por día (Se divide la cantidad de favoritos o me gusta entre el tiempo de la cuenta).

Para el desarrollo del nuevo dataset se consultó el nivel bot de cada usuario con la herramienta en línea "*https://botometer.osome.iu.edu/"*.

El proceso de desarrollo del dataset, consistió primero en pre-seleccionar 9,951 usuarios, después se tomó una muestra aleatoria de 1,200, por último, se clasificaron manualmente a 502 usuarios, resultando 252 usuarios como cuentas bot y 250 usuarios como cuentas con perfil humano (Figura 3.6).

**Figura 3.6.**

*Proceso para el desarrollo del dataset bot.*




Elaboración propia.

La tabla 3.2 nos muestra las características de algunos usuarios y su clasificación obtenida mediante la plataforma "*botometer*", los usuarios bots en la columna "*Bot*" se identificaron con el número 1 y los usuarios catalogados como humanos se identificaron con el número 0.

**Tabla 3.2.**

*Muestra del dataset bot COVID-19.*

| Usuario | Bot | Seguidores | Siguiendo | Favoritos | Tweets | Listas | Tiempo de cuenta | Tweets por día | Calidad |
|---|---|---|---|---|---|---|---|---|---|
| @usuario1 | 0 | 62 | 685182 | 2058 | 633 | 0 | 41.83 | 64.33 | 0.31 |
| @usuario2 | 0 | 453 | 173 | 80713 | 81820 | 3 | 1643.01 | 98.92 | 0.75 |
| @usuario3 | 1 | 1 | 26 | 86 | 57 | 0 | 178.36 | 0.80 | 0.01 |
| @usuario4 | 0 | 3330 | 323 | 13946 | 12168 | 6 | 567.85 | 45.99 | 0.48 |
| @usuario5 | 1 | 3 | 107 | 588 | 284 | 0 | 541.94 | 1.61 | 0.6 |
| @usuario6 | 1 | 0 | 56 | 11 | 6 | 0 | 509.06 | 0.03 | 0 |

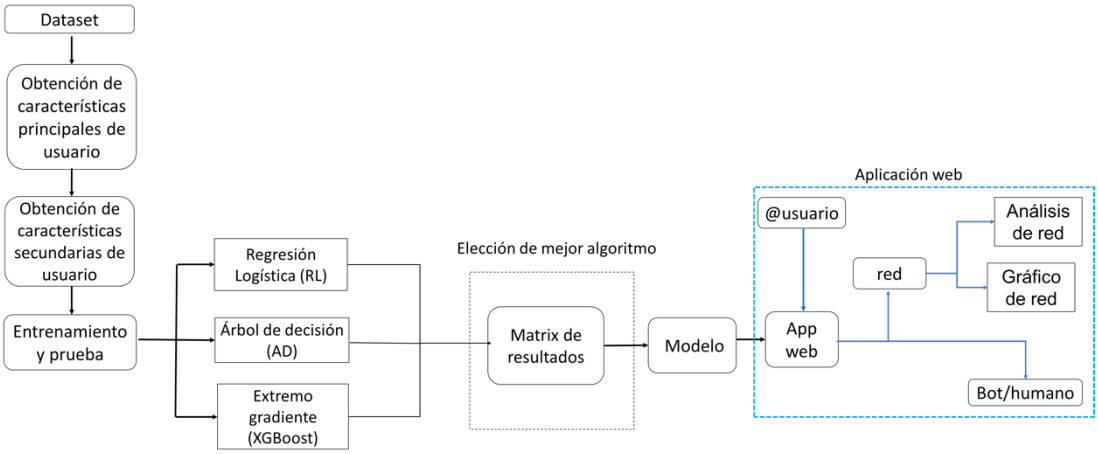
Elaboración propia.

## 3.6. Modelo de aprendizaje automático para la detección de cuentas bot

El modelo para el desarrollo de la detección en línea del dataset bot COVID-19 se muestra en la figura 3.7, donde se puede ver el proceso para la obtención de características principales y secundarias de las cuentas, para después particionar el dataset en entrenamiento y prueba, después se ejecutan los algoritmos de aprendizaje automático y se elige el mejor algoritmo para implamentar una aplicación web para la detección en línea de cuentas bot.

**Figura 3.7.**

*Modelo para la detección en línea de cuentas bot.*




Elaboración propia.

## 3.7. Determinación de red de amigos de amigos

De acuerdo al apartado 2.4, esta investigación se enfocó en determinar la red viral de amigos de un usuario de redes sociales, para este caso se desarrolló un modelo para la extracción de una porción de la red de amigos-amigos de un usuario en la red social de Twitter que determina el grado, la modularidad, la transitividad, y el promedio de ruta o camino entre dos nodos (ver apartado 2.3), también es capaz de generar una visualización de la estructura de la red, para poder determinar si la red de amigos pertenece a una cámara de eco o tiene posibilidades de viralizar algún contenido se debe determinar el valor de dos coeficientes característicos de las redes del mundo pequeño, estos coeficientes son sigma y omega, si el coeficiente de omega es mayor a 1 se dice que la red pertenece al mundo pequeño y por lo tanto es altamente viral entre la red de amigos de amigos de un usuario, el coeficiente de sigma debe estar cercano a cero para que la red se considere como una red del mundo pequeño, a continuación se presenta el pseudo-código para determinar la red viral de amigos-amigos de un usuario en Twitter.

**Proceso** Red_de_amigos_de_amigos
**Entradas:**
API_keys ← Escribe las API keys para Twitter
usuario ← Escribe el nombre del usuario(@screen_name)
**Para** cada usuario
Obtener los amigos del usuario (nodos)
**Para** cada amigo de los amigos del usuario
Obtener los amigos (nodos)
Cruzar los que sean amigos (vínculos)

**Finalizar**
**Finalizar**
Crear red
**Fin**

# CAPÍTULO 4

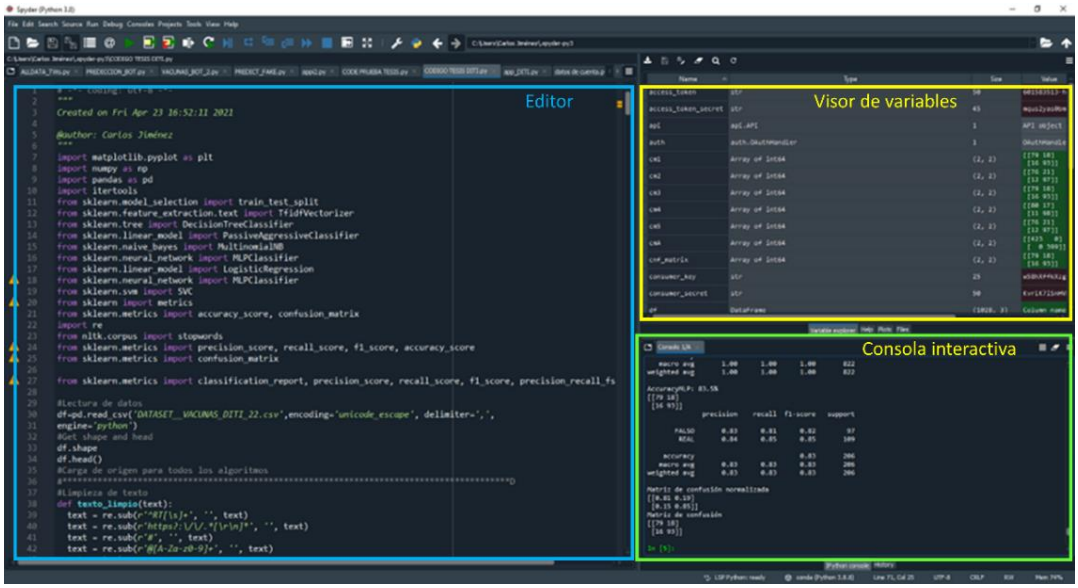
# DESARROLLO Y RESULTADOS

## 4.1. Entorno de desarrollo (Spyder)

Para el desarrollo de los modelos se utilizó un entorno de desarrollo integrado o IDE por sus siglas en inglés (Integrated develepment Envorinment), mediante el uso de la plataforma conocido como “Spyder” por sus siglas en inglés (*Scientific Python Develoment EnviRonment),* es una plataforma de código abierto y está escrito en programa Python, posee un editor, un visor de variables y una consola interactiva que se puede personalizar (Figura 4.1) [96].

**Figura 4.1.**

*Entorno de desarrollo integrado de Spyder 4.2.5.*




Fuente: Captura de pantalla a programa al entorno de desarrollo.

## 4.2. Resultado de análisis de texto en datasets de noticias falsas

Para una mejor comprensión de los tres datasets propuestos para la detección de noticias falsas en español, se realizó un análisis de texto donde se determinó el promedio y la cantidad de palabras por tipo de clasificación y de sentimientos mediante el método desarrollado por Nielsen [94], en la tabla 4.1 se puede observar el resultado comparativo entre los tres datasets, en promedio la cantidad de palabras en la noticia real fue mayor que la cantidad de palabras en las noticias falsas, de igual forma en la cantidad de palabras distintas, la noticia o contenido real obtuvo la mayor cantidad, lo cual nos indica que los textos clasificados como noticia real poseen un mayor vocabulario frente a los textos clasificados como noticia falsa; además se observa que en los textos clasificados como noticias falsas el sentimiento

negativo fue mayoría con un valor de 60.1%, mientras que el sentimiento positivo alcanzo un valor del 35.0%, y el sentimiento neutro un valor del 4.9%; mientras que, en los textos clasificados como noticia real el promedio de los sentimientos negativos y positivos fueron equilibrados, con valores del 48.1% y 48.2% respectivamente, y el sentimiento neutro representó un 3.7%.

**Tabla 4.1.**

*Análisis de texto y sentimiento a los datasets de noticias falsas en español.*

| | Constraint | Iberlef | DITI-Infodemia MX | **Promedio** |
|---|---|---|---|---|
| Promedio de palabras "Falso" | 162.1 | 2,090.9 | 124.0 | 792.3 |
| Promedio de palabras "Real" | 244.4 | 3,303.3 | 144.0 | 1,230.6 |
| Cantidad de palabras totales en "Falso" | 69,017 | 203,821 | 5,298 | 92,712 |
| Cantidad de palabras totales en "Real" | 103,272 | 318,350 | 5,401 | 142,341 |
| Cantidad de palabras distintas "Falso" | 10,090 | 24,600 | 1,525 | 12,071.7 |
| Cantidad de palabras distintas "Real" | 8,093 | 29,280 | 1,601 | 12,991.3 |
| % Sentimiento positivo en "Falso" | 34.1% | 45.7% | 25.2% | 35.0% |
| % Sentimiento neutro en "Falso" | 4.9% | 2.1% | 7.7% | 4.9% |
| % Sentimiento negativo en "Falso" | 61.0% | 52.2% | 67.1% | 60.1% |
| % Sentimiento positivo en "Real" | 44.3% | 51.9% | 48.0% | 48.1% |
| % Sentimiento neutro en "Real" | 4.8% | 2.3% | 4.1% | 3.7% |
| % Sentimiento negativo en "Real" | 50.9% | 45.8% | 48.0% | 48.2% |

Elaboración propia.

Para el análisis de rendimiento de los algoritmos de aprendizaje automático, se ejecutaron varias librerías del programa “*Sklearn*”, el cual es un programa ampliamente utilizado para realizar análisis de regresión o clasificación mediante la aplicación de algoritmos de aprendizaje automático, además posee módulos para la extracción de características, procesamiento de datos y evaluación de modelos, este programa se basa en bibliotecas de programas como Python, Numpy y Matplolib; tiene una licencia permisiva, por

lo que se puede utilizar en aplicaciones comerciales o académicas sin restricciones [98]. Con la siguiente línea de código en la consola del IDE de Spyder se puede instalar la librería completa de sklearn:

```
pip install scikit-learn.
```

Otras librerías de Sklearn que son necesarias instalar para ejecutar el modelo propuesto son:

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.neural_network import MLPClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neural_network import MLPClassifier
from sklearn.svm import SVC
from xgboost import XGBClassifier
from sklearn import metrics
from sklearn.metrics import accuracy_score, confusion_matrix
from nltk.corpus import stopwords
from sklearn.metrics import precision_score, recall_score, f1_score,
accuracy_score
from sklearn.metrics import confusion_matrix.
```

### 4.3. Resultado de procesamiento de datasets de noticias falsas

El modelo propuesto en esta investigación inicia con la entrada de los datos que componen el dataset, después se realizó la limpieza de los datos, donde cada tweet del conjunto se procesó para eliminar los caracteres especiales o palabras que no abonan como los retweets o "*RT*", también se eliminaron los números y otros símbolos como el "*hashtag"* (#) el cual es utilizado para etiquetar alguna frase, nombre o tema para distinguir un tema específico en Twitter.

A continuación, se muestra la función que se desarrolló para el preprocesamiento de texto:

```
#Limpieza de texto
def ntexto(texto):
  texto =  text.lower()
  texto = re.sub(r'^RT[\s]+', '', texto)
  texto = re.sub(r'https?:\/\/.*[\r\n]*', '', texto)
  texto = re.sub(r'#', '', texto)
  texto = re.sub(r'@[A-Za-z0-9]\S+', '', text0)
  # Eliminación de signos de puntuación
  regex = '[\\!\\"\\#\\$\\%\\&\\\'\\(\\)\\*\\+\\,\\-
\\.\\/\\:\\;\\<\\=\\>\\?\\@\\[\\\\\\]\\^_\\`\\{\\|\\}\\~]'
  text = re.sub(regex , ' ', texto)
  text = re.sub("\\s+", ' ', texto)
  return texto
df['ntexto'] = df['texto'].apply(ntexto)
df['ntexto'].
```

En la tabla 4.2 se muestra una parte del resultado de este proceso de "*limpiar texto*" con el dataset de DITI-Infodemia.MX, se puede observar como todas las palabras que iniciaron con el símbolo "@" fueron eliminadas, enTwitter se identifica con el símbolo de arroba (@) a los usuarios o cuentas, también se eliminaron las palabras que inician con "http://", ya que representan direcciones de sitios web y otros tipos de caracteres especiales.

**Tabla 4.2.**

*Limpieza de texto.*

| Texto original | Texto limpio |
|---|---|
| @Pablo76Martin @LuisGasulla Lo que se sabe es que las vacunas no contienen grafeno | Lo que se sabe es que las vacunas no contienen grafeno |
| @jedomm @poirotes @24HorasTVN No son vacunas, entiende de una vez! | No son vacunas, entiende de una vez |
| @Pildora_II Que las vacunas son peor que el virus. https://t.co/BzOudK5cRa | Que las vacunas son peor que el virus. |

Elaboración propia.

Después de limpiar el texto de cada tweet o contenido, se pasó a la obtención de etiquetas de las columnas del dataset, este proceso dio lectura a la clasificación de cada contenido, el cual es "*Real*" para los tweets que tuvieron contenidos o noticias reales, y "*Falso*" para los tweets con contenidos o noticias falsas, con las siguientes instrucciones se dio lectura a las etiquetas o calcificaciones de los textos:

```
#Obtención de etiquetas 1
labels=df.label
labels.head()
print(df.head()).
```

La tabla 4.3 nos muestra una parte de este proceso en el dataset de DITI-Infodemia.MX, donde se puede observar que solo fue tomado su número índice y su etiqueta respectiva.

**Tabla 4.3.**

*Obtención de etiquetas.*

| Índice | Etiqueta |
|---|---|
| 515 | Falso |
| 516 | Falso |
| 517 | Falso |
| 518 | Falso |
| 519 | Falso |
| 520 | Real |
| 521 | Real |
| 522 | Real |

Elaboración propia.

La tabla 4.4 nos muestra los porcentajes por tipo de noticias o contenido en cada uno de los datasets propuestos para esta investigación, se puede observar que en los tres datasets hay un equilibrio entre las clases falso y real.

**Tabla 4.4.**

*Tipo de clasificación de contenidos en datasets.*

| Tipo de contenido | Constraint | Iberlef | Infodemia |
|---|---|---|---|
| Falso | 47.3% | 50.0% | 52.5% |
| Real | 52.7% | 50.0% | 47.5% |

Elaboración propia.

Para el ajuste y transformación del dataset, se utilizó el algoritmo de frecuencia de termino-frecuencia de documento inversa, el cual se le conoce como "*Tf-Idf Vectorizer*", para inicializar este algoritmo y adecuarlo al idioma español es necesario establecer los "*stopwords*" o palabras vacías, tales como "*de*", "*la*", "*que*", "*el*" y también en español castellano como "*vuestros*", "*vuestras*" o "*sois*", en total son 312 palabras, después de este proceso solo quedaron las palabras importantes de un texto [99].

Se aplicó la división del dataset en dos partes, una para el entrenamiento y otra para las pruebas de clasificación, además se instrumentó el algoritmo de TF-IDF para cada una de las palabras que componen el daaset.

El algoritmo TF-IDF realizó observaciones sobre cuantas veces aparece una palabra en el texto y al mismo tiempo en otros textos, esto significo que determino una puntuación para cada palabra, en este caso por tweet del dataset, este valor aumento con cada vez que la palabra apareció en el conjunto de datos, pero disminuyó con cada vez que sale en otro texto [100], la fórmula de este algoritmo es para TF la siguiente:

$$Tf(w,d) = \log\left(1 + f(w,d)\right) \tag{32}$$

donde $f(w,d)$ es la frecuencia de la palabra "$w$" en el documento "$d$".

Para calcular la frecuencia inversa (Idf) es:

$$idf(w, D) = \log\left(\frac{N}{f(w, D)}\right) \tag{33}$$

donde $N$ es el número de documentos y $D$ es conjunto total y $f(w, D)$ es la frecuencia de la palabra $w$ dentro del conjunto de datos.

Una parte de la ejecución del algoritmo “TF-IDF” en el dataset de Infodemia se puede observar en la tabla 4.5, donde se muestran algunas palabras y el cálculo de frecuencia de documento inversa o IDF.

**Tabla 4.5.**

*Coeficientes del algoritmo de Frecuencia Inversa (FI).*

| Palabra | IDF |
|---|---|
| chips | 7.01981 |
| grafeno | 2.41464 |
| grafetonita | 7.01981 |
| política | 6.10352 |
| antivacuna | 6.32666 |

Elaboración propia.

Siguiendo el proceso del modelo, se efectuó la división del dataset de DITI-InfodemiaMx, una parte para el entrenamiento de los modelos algorítmicos y otra parte para la prueba, la tabla 4.6 nos muestra el resultado de la partición de los datasets de noticias falsas, en una proporción para los tres datasets de 80% de los datos para entrenamiento y un 20% para pruebas.

**Tabla 4.6.**

*Partición de los datos en entrenamiento y prueba.*

| Partición | Constraint | Iberlef | DITI-Infodemia MX |
|---|---|---|---|
| Entrenamiento | 5136 | 998 | 510 |
| Prueba | 1284 | 250 | 128 |
| Total de tweets / contenido | 6420 | 1248 | 638 |

Elaboración propia.

## 4.4. Resultado de evaluación de dataset Constraint

El dataset de Constraint se compone de 6,420 noticias relacionadas con el COVID-19, las cuales fueron clasificadas en noticia real (Real) o noticia falsa (Falso), el dataset originalmente está escrito en idioma inglés y se tradujo al español; de acuerdo al modelo propuesto en 3.4, se desarrolló un código en Spyder-Python para llevar a cabo el entrenamiento, prueba y medición de los algoritmos propuestos, la tabla 4.7 nos muestra la matriz de evaluación, se puede observar que el algoritmo de Maquina de Vectores de Soporte (MVS) obtuvo la mejor exactitud (*Accuracy)* con un valor del 92.76%, seguido del algoritmo Pasivo-Agresivo (PA) que obtuvo un "*Accuracy"* de 92.29% y en tercer lugar se situó el algoritmo Multica Perceptrón (MP) con un valor de exactitud del 91.90%.

**Tabla 4.7.**

*Matriz de evaluación para el dataset de Constraint.*

| Algoritmos | Clasificación | Métricas | | | |
|---|---|---|---|---|---|
| | | **Precisión** | ***Recuperación*** | ***f1-score*** | ***Exactitud*** |
| AD | Falso | 0.85 | 0.82 | 0.83 | 84.03% |
| | Real | 0.83 | 0.86 | 0.85 | |
| MVS | Falso | 0.92 | 0.93 | 0.93 | 92.76% |
| | Real | 0.93 | 0.92 | 0.93 | |
| NB | Falso | 0.95 | 0.83 | 0.89 | 89.49% |
| | Real | 0.85 | 0.96 | 0.90 | |
| RL | Falso | 0.91 | 0.92 | 0.91 | 91.51% |
| | Real | 0.92 | 0.91 | 0.92 | |
| PA | Falso | 0.93 | 0.91 | 0.92 | 92.29% |
| | Real | 0.92 | 0.93 | 0.92 | |
| MP | Falso | 0.92 | 0.91 | 0.92 | 91.90% |
| | Real | 0.92 | 0.93 | 0.92 | |
| | | | | Promedio | 90.33% |
| | | | | Desviación estándar | 3.30% |

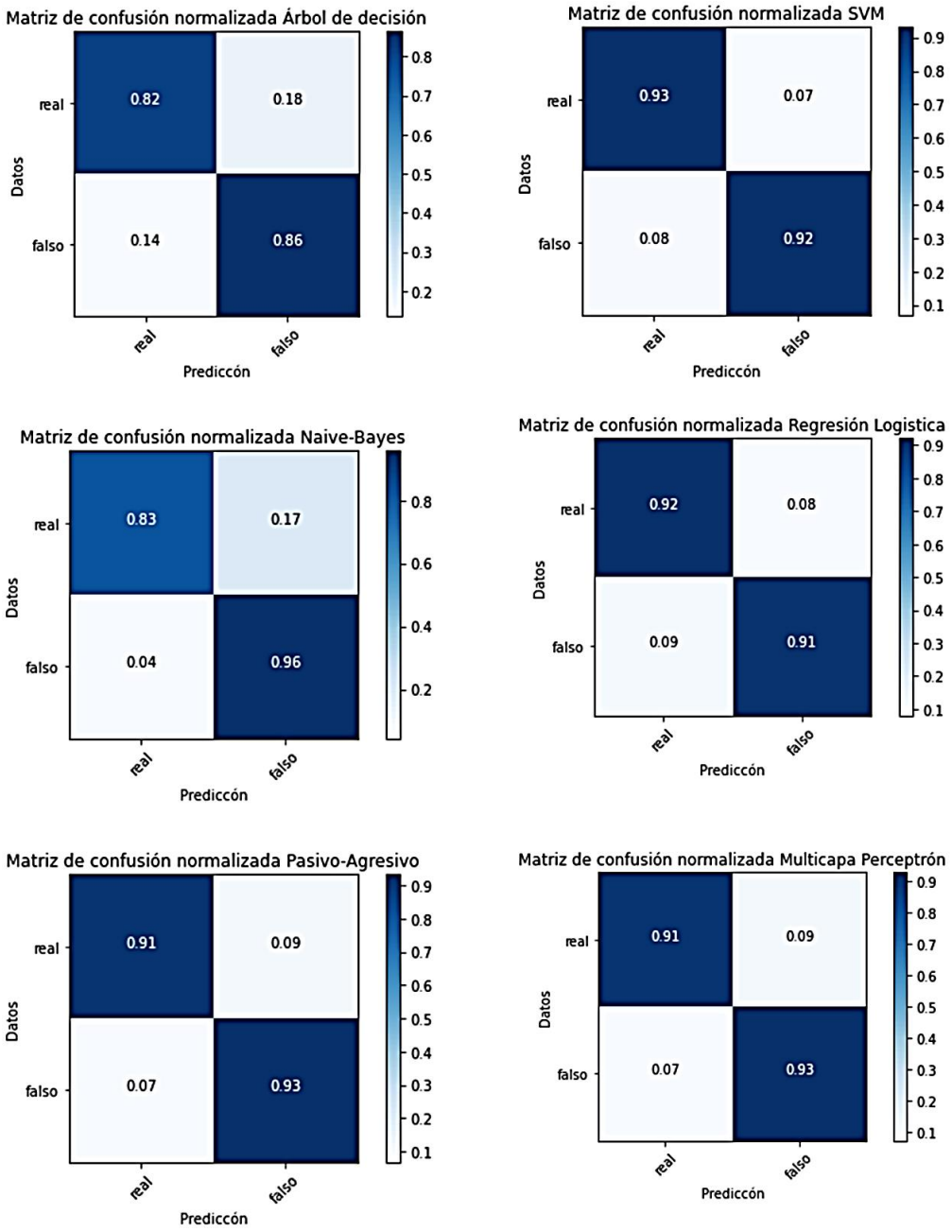
Elaboración propia.

### 4.4.1. Matrices de confusión de dataset Constraint

Para el dataset Constraint, el algoritmo de Naive -Bayes fue el que obtuvo el valor más alto con un 96% para la noticia falsa, y en noticia real fue el algoritmo de Maquina de vectores de soporte, con un valor del 93% (Figura 4.2).

**Figura 4.2.**

*Matrices de confusión para el dataset Constraint.*




Elaboración propia.

## 4.5. Resultado de evaluación de dataset IberLef

El dataset de Iberlef está compuesto por 1,248 noticias o contenidos que fueron catalogados como noticia real (Real) o como noticia falsa (Falso), el dataset está en idioma español, en la tabla 4.8 podemos ver la matriz de evaluación, donde se puede ver que el algoritmo de Regresión Logística (LR) fue el obtuvo la mejor exactitud (*Accuracy*) con un valor del 72.80%, seguido del algoritmo de Maquina de Vectores de Soporte (MVS) que obtuvo un valor de 72.00% y en tercer lugar se ubicó el algoritmo Pasivo-Agresivo (PA) con un valor de 71.60%.

**Tabla 4.8.**

*Matriz de evaluación para el dataset de IberLef.*

| **Algoritmos** | **Clasificación** | **Métricas** | | | |
|---|---|---|---|---|---|
| | | **Precisión** | ***Recuperación*** | ***f1-score*** | ***Exactitud*** |
| AD | Falso | 0.62 | 0.57 | 0.6 | 60.80% |
| | Real | 0.59 | 0.64 | 0.62 | |
| MVS | Falso | 0.72 | 0.74 | 0.73 | 72.00% |
| | Real | 0.72 | 0.7 | 0.71 | |
| NB | Falso | 0.83 | 0.28 | 0.41 | 60.40% |
| | Real | 0.56 | 0.94 | 0.7 | |
| RL | Falso | 0.73 | 0.75 | 0.74 | 72.80% |
| | Real | 0.73 | 0.71 | 0.72 | |
| PA | Falso | 0.73 | 0.69 | 0.71 | 71.60% |
| | Real | 0.7 | 0.74 | 0.72 | |
| MP | Falso | 0.72 | 0.66 | 0.69 | 70.00% |
| | Real | 0.68 | 0.74 | 0.71 | |
| | | | | Promedio | 67.93% |
| | | | | Desviación estándar | 5.80% |

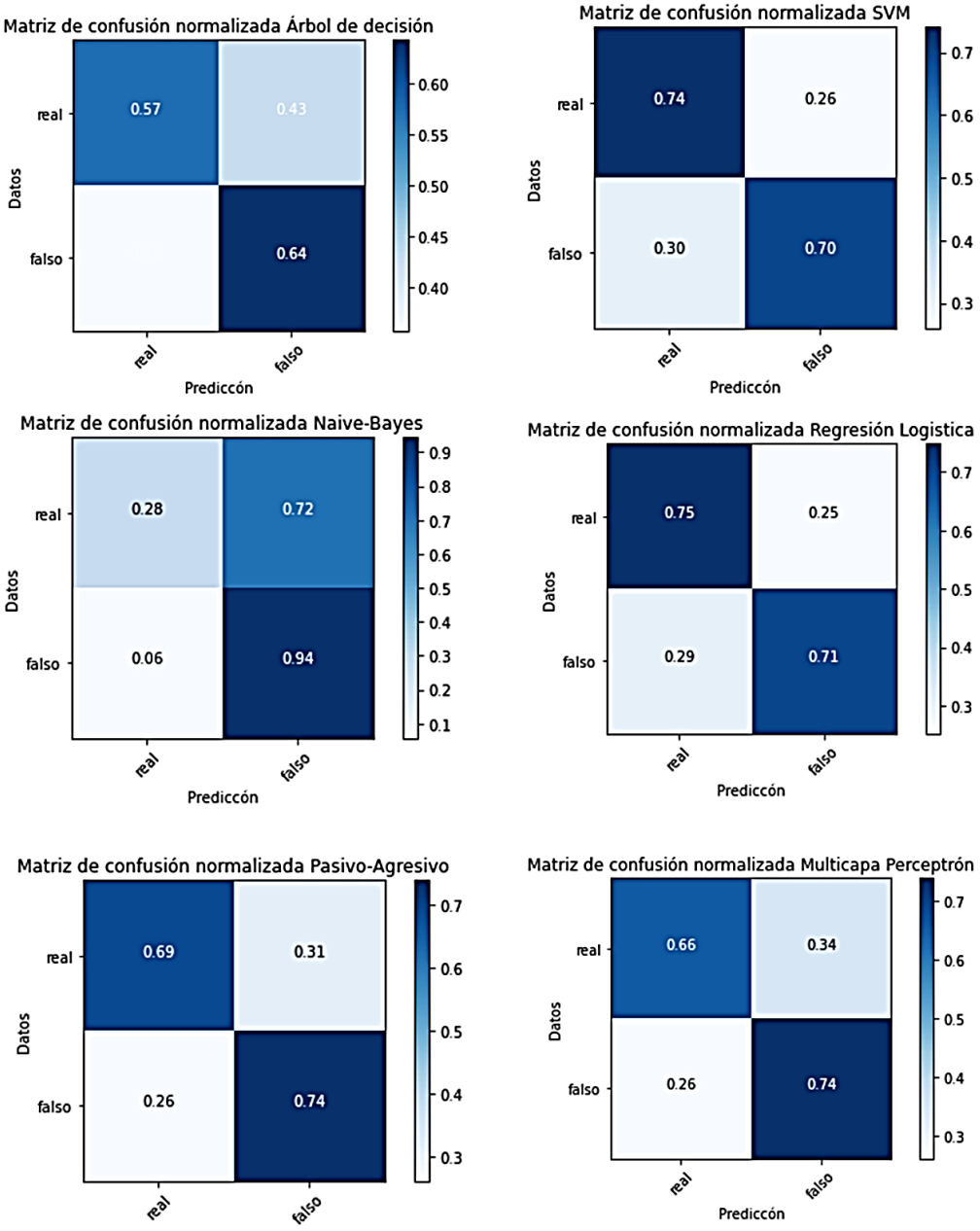
Elaboración propia.

### 4.5.1. Matrices de confusión de dataset IberLef

Para el dataset de IberLef, se puede ver que el algoritmo de Naive-Bayes (NB) fue el de mayor valor con un 94% de predicción en noticia falsa, mientras que, el algoritmo de regresión logística obtuvo el valor mayor para noticia real con 75% (Figura 4.3).

**Figura 4.3.**

*Matrices de confusión para el dataset IberLef.*




Elaboración propia.

## 4.6. Resultado de evaluación de dataset DITI-Infodemia MX

El dataset de Infodemia MX se compone 638 tweets en idioma español que están relacionados con el tema de las vacunas del COVID-19, en la tabla 4.9 se puede ver la matriz de evaluación del dataset con los algoritmos propuestos para esta investigación, donde se puede observar que el algoritmo de Regresión Logística (RL) fue el que obtuvo la mejor exactitud *(Accuracy)* con un valor de 78.91%, seguido de los algoritmos de Maquina de Vectores de Soporte (MVS), Pasivo-Agresivo (PA) y Multicapa Perceptrón (MP) los tres algoritmos con un valor de 78.12%.

**Tabla 4.9.**

*Matriz de evaluación para el dataset de DITI-Infodemia MX.*

| **Algoritmos** | **Clasificación** | **Métricas** | | | |
|---|---|---|---|---|---|
| | | **Precisión** | ***Recuperación*** | ***f1-score*** | ***Exactitud*** |
| AD | Falso | 0.78 | 0.81 | 0.79 | 76.56% |
| | Real | 0.74 | 0.71 | 0.73 | |
| MVS | Falso | 0.82 | 0.78 | 0.8 | 78.12% |
| | Real | 0.73 | 0.79 | 0.76 | |
| NB | Falso | 0.81 | 0.78 | 0.79 | 77.34% |
| | Real | 0.73 | 0.77 | 0.75 | |
| RL | Falso | 0.85 | 0.76 | 0.8 | 78.91% |
| | Real | 0.73 | 0.82 | 0.77 | |
| PA | Falso | 0.81 | 0.79 | 0.8 | 78.12% |
| | Real | 0.74 | 0.77 | 0.75 | |
| MP | Falso | 0.83 | 0.76 | 0.8 | 78.12% |
| | Real | 0.73 | 0.8 | 0.76 | |
| | | | | Promedio | 77.86% |
| | | | | Desviación estándar | 0.81% |

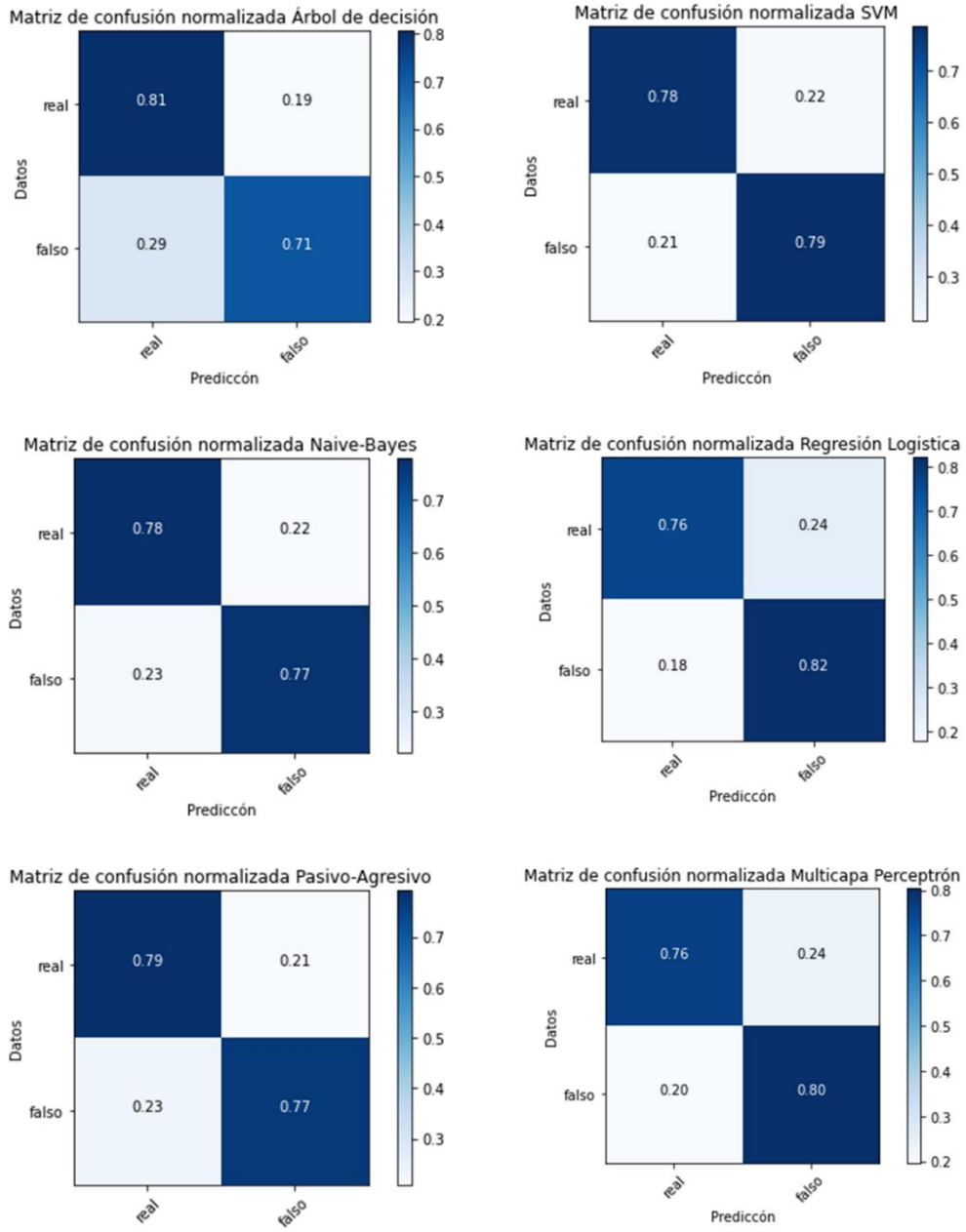
Elaboración propia.

### 4.6.1. Matrices de confusión de DITI-Infodemia MX

Para el dataset de DITI-Infodemia.MX el algoritmo de regresión logística fue el que obtuvo el mayor valor de predicción con un 82% para noticia falsa, mientras que, para la noticia real fue el árbol de decisión con un valor de predicción del 81% (Figura 4.4).

**Figura 4.4.**

*Matrices de confusión para el dataset DITI-Infodemia MX.*




Elaboración propia.

## 4.7. Resultado de elección del mejor algoritmo para noticias falsas

El concentrado de la métrica de "*Accuacy"* de los distintos algoritmos en los tres datasets de prueba se pueden observar en la tabla 4.10, donde se puede ver que el algoritmo de Maquinas de Vectores de Soporte (MVS) obtuvo la mejor exactitud (*Accuracy*) en el dataset de Constraint con un valor del 92.8%, también se observó que al algoritmo de Regresión Logística (RL) fue el que obtuvo el mejor promedio de exactitud en los tres datasets con un valor del 81.1% y también la mejor desviación estándar con un valor del 9.5%, por lo que se decidió que, el algoritmo de regresión logística (RL) es el mejor algoritmo para el desarrollo del modelo para la detección de las noticias falsas.

**Tabla 4.10.**

*Comparativo de promedios y desviación estándar de exactitud (Accuracy).*

| Algoritmos | **Constraint** | **Iberlef** | **DITI-Infodemia MX** | **Promedio** | **Desviación estándar** |
|---|---|---|---|---|---|
| AD | 84.0% | 60.8% | 76.6% | 73.8% | 11.9% |
| MVS | 92.8% | 72.0% | 78.1% | 81.0% | 10.7% |
| NB | 89.5% | 60.4% | 77.3% | 75.7% | 14.6% |
| RL | 91.5% | 72.8% | 78.9% | 81.1% | 9.5% |
| PA | 92.3% | 71.6% | 78.1% | 80.7% | 10.6% |
| MP | 91.9% | 70.0% | 78.1% | 80.0% | 11.1% |
| Promedio | 90.3% | 67.9% | 77.8% | | |
| Desv. Est. | 3.3% | 5.7% | 0.79% | | |

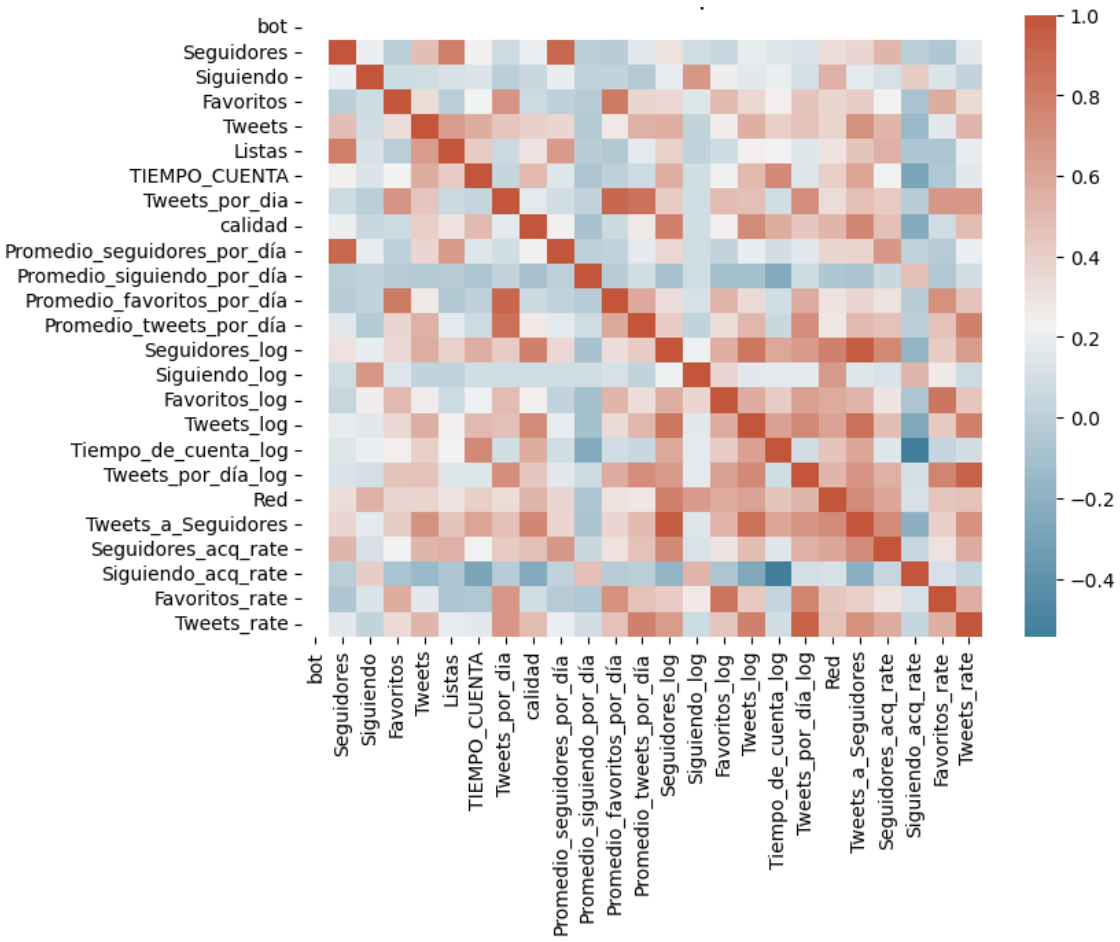
Elaboración propia.

## 4.8. Características de usuario en dataset “Bot COVID-19”

Tal como se mostró en el apartado 3.5 se desarrolló un dataset bot para la detección de cuentas o usuarios de tipo bot en Twitter se toma en cuenta ciertas características, en la figura 4.5 se muestra las correlaciones entre diversas características para los usuarios clasificados como bots, donde se puede observar que varias características tienen un nivel aceptable de correlación.

**Figura 4.5**.

*Correlación de características de usuarios tipo bot.*

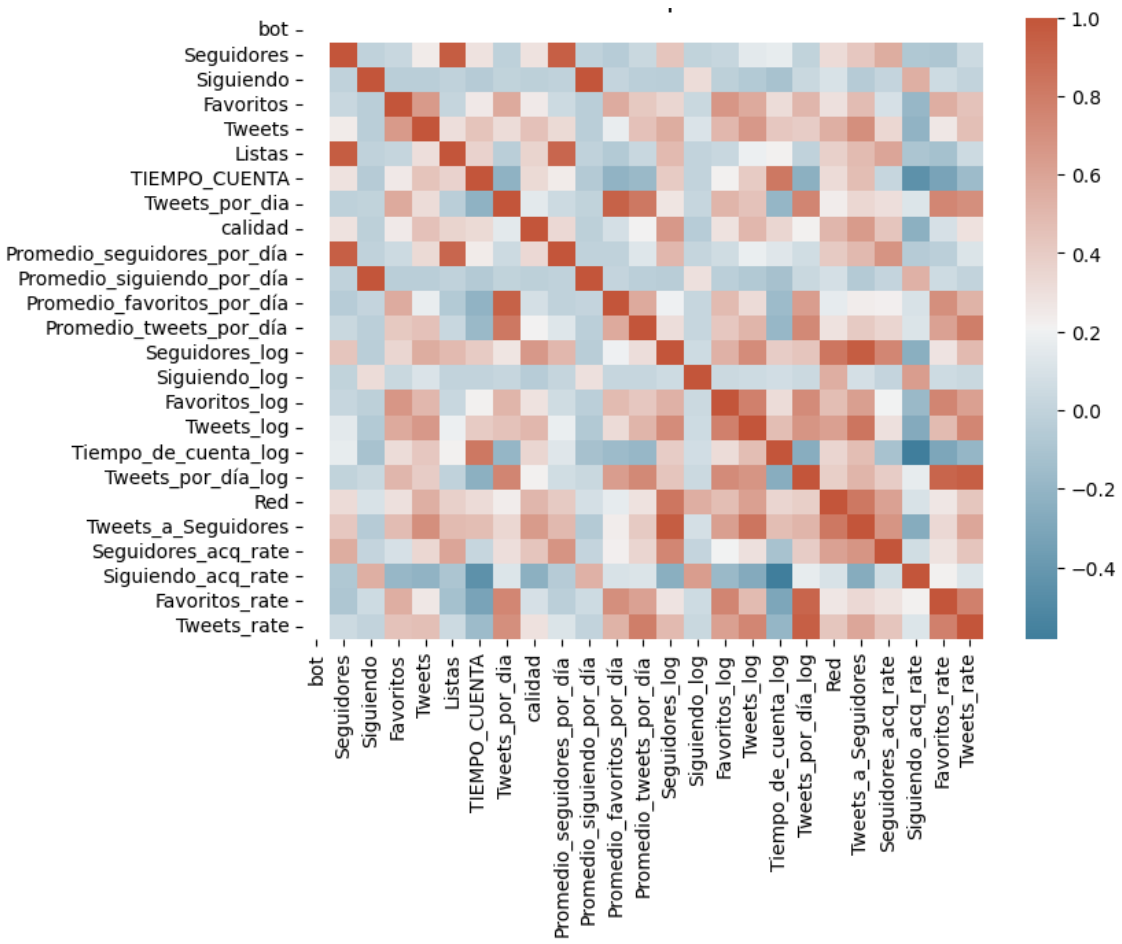



Elaboración propia.

En la figura 4.6 se muestran las correlaciones de diversas características para los usuarios clasificados como humanos, donde se puede ver que varias de las características tienen un nivel aceptable de correlación.

**Figura 4.6.**

*Correlación de características de usuarios tipo humano.*

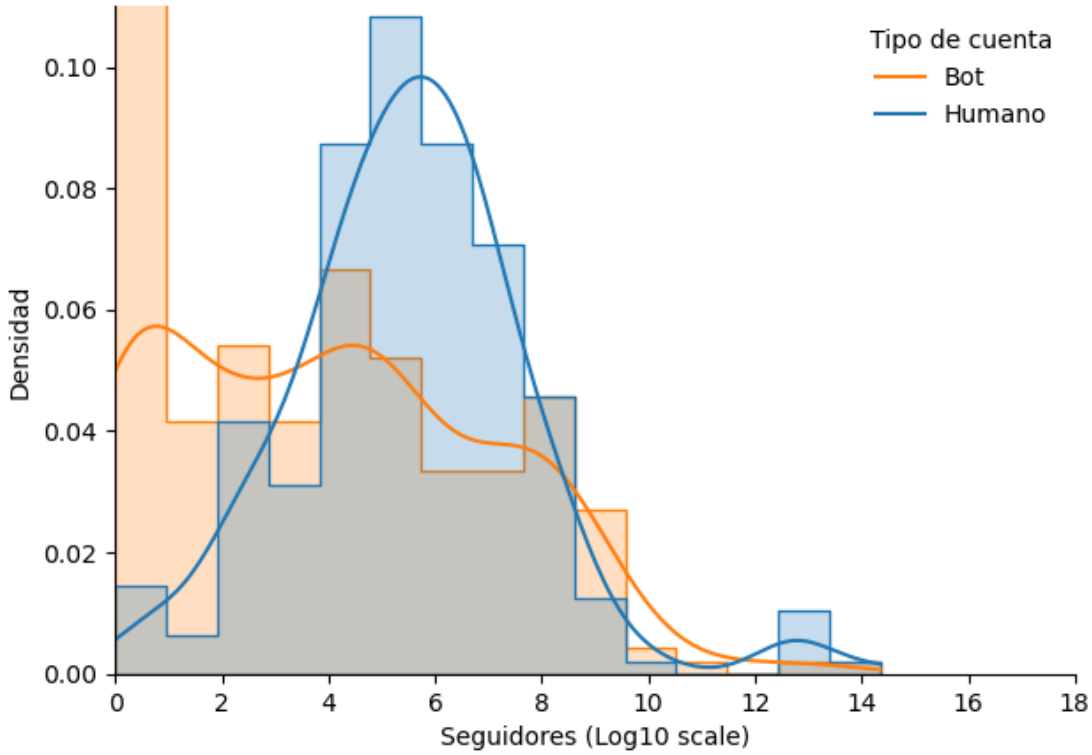



Elaboración propia.

Como se ha visto, hay una correlación entre las características y la clasificación bot en ambos tipos de cuentas, pero también nos es útil conocer las diferencias entre los tipos de cuenta, para esto se realizó un análisis de las distribuciones, tal como se observa en la figura 4.7 se puede ver que, la distribución de las cuentas clasificadas como humanas siguen una distribución de tipo normal, mientras que, las cuentas tipo bot tienen una distribución sesgada hacia una cantidad menor de seguidores.

**Figura 4.7.**

*Distribución por densidad en escala logarítmica del número de seguidores por tipo de cuenta en dataset "Bot COVID-19".*

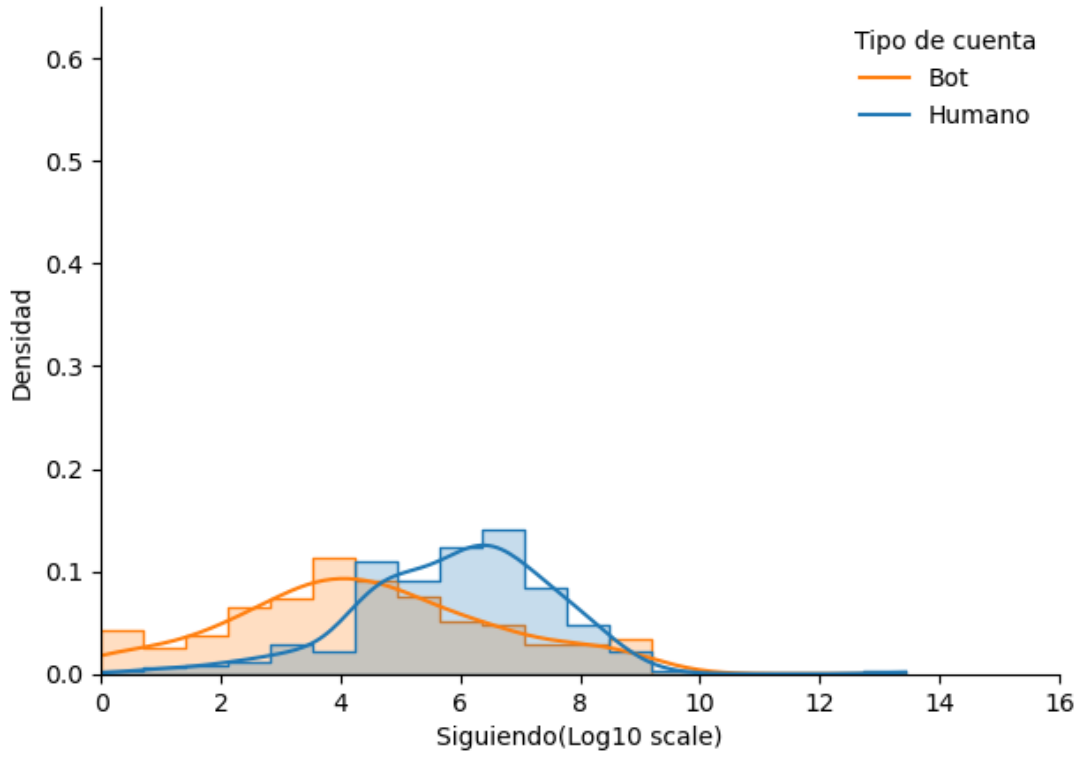



Elaboración propia.

En la cantidad de cuentas a las que sigue los usuarios tipo bot/humano, también se encontró diferencias en sus distribuciones, tal como se muestra en la figura 4.8, ambos tipos de cuentas presentan una distribución de tipo normal, pero la distribución de cuentas tipo humano posee visiblemente una media mayor que la distribución de tipo bot, a la característica de siguiendo también se le denomina "*amigos*", porque se presume que la acción de seguir a alguien es porque lo considera cercano y quiere recibir contenido directamente, de acuerdo a este resultado se puede inferir que las cuentas humanas tienen más amigos que las cuentas bot.

**Figura 4.8.**

*Distribución por densidad en escala logarítmica del número de cuentas siguiendo (amigos) por tipo de cuenta en dataset "Bot COVID-19".*

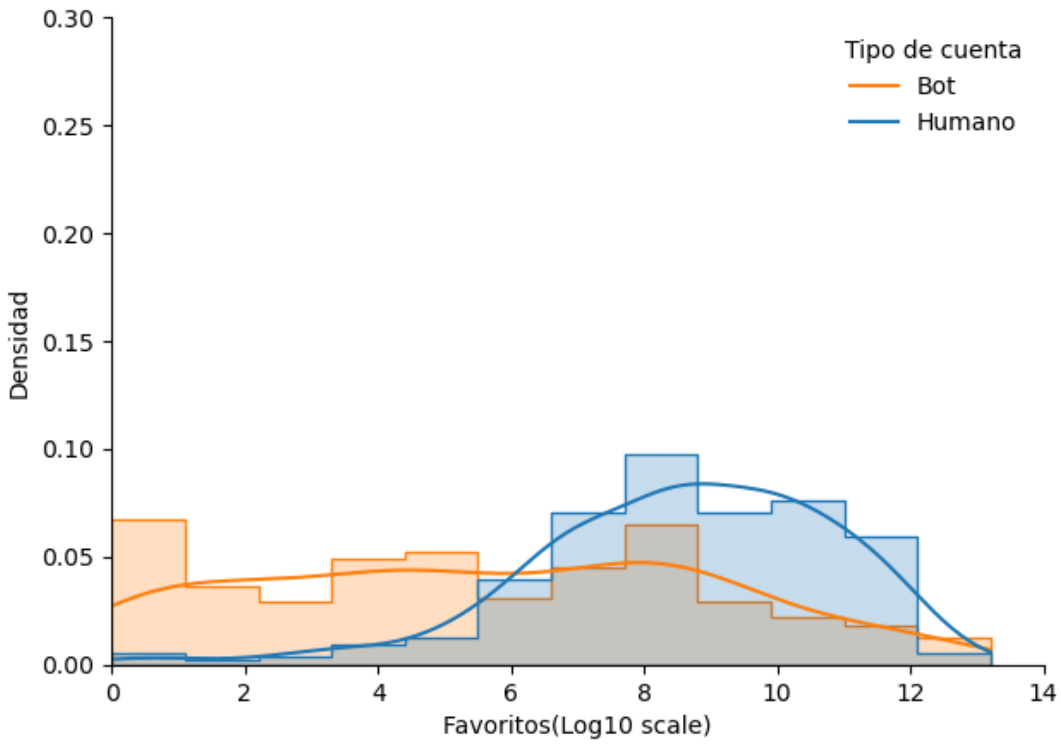



Elaboración propia.

Los usuarios o cuentas en Twitter pueden emitir “*favoritos*”, esta acción se utiliza para mostrar que algún contenido les gusta, la figura 4.9 nos muestra las distribuciones de favoritos de las cuentas tipo bot/humano, se observa que la distribución de cuentas tipo humano tienen una mayor distribución más normal y con una media visiblemente mayor de favoritos que las cuentas tipo bot.

**Figura 4.9.**

*Distribución por densidad en escala logarítmica del número de favoritos (me gustas) emitidos por tipo de cuenta en dataset ”Bot COVID-19”.*

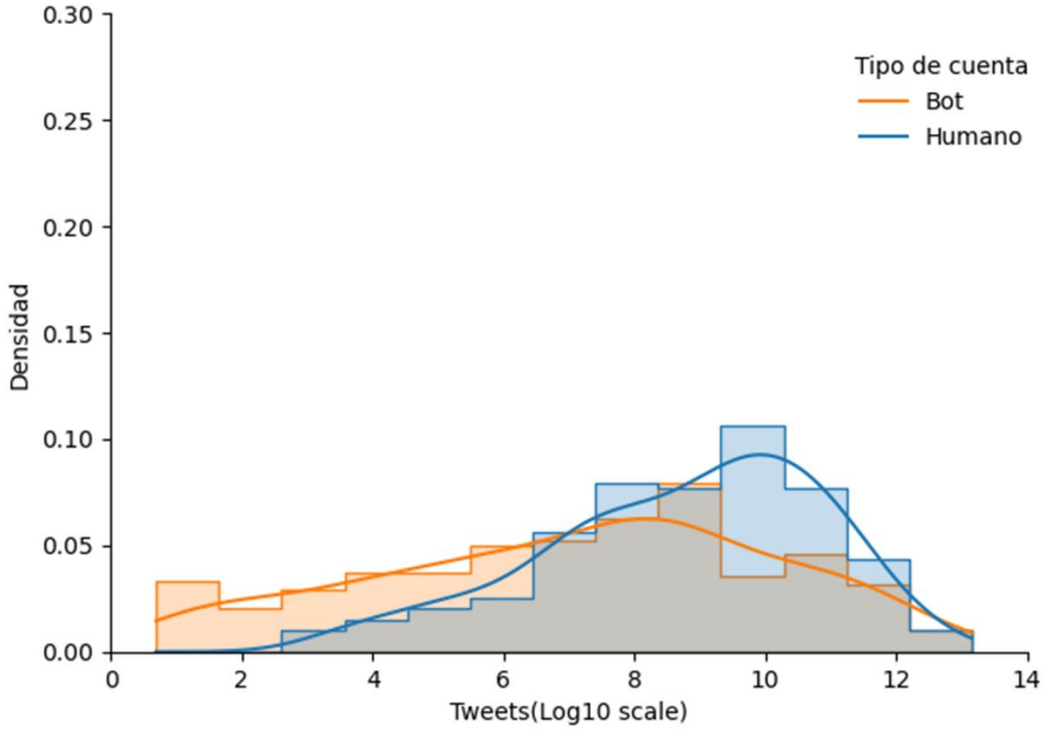



Elaboración propia.

La función principal que poseen los usuarios o cuentas en Twitter es emitir tweets que pueden ser textos, imágenes, gifs, videos cortos o links, toda esta actividad se denomina simplemente como la cantidad de tweets, en la figura 4.10 se puede observar la distribución para ambos tipos de cuentas de los tweets emitidos, se puede observar que, los usuarios o cuentas tipo humano, tienen una aproximación a la normalidad y visiblemente una mayor media de tweets que las cuentas tipo bot.

**Figura 4.10.**

*Distribución por densidad en escala logarítmica del número de tweets emitidos por tipo de cuenta en dataset "Bot COVID-19".*

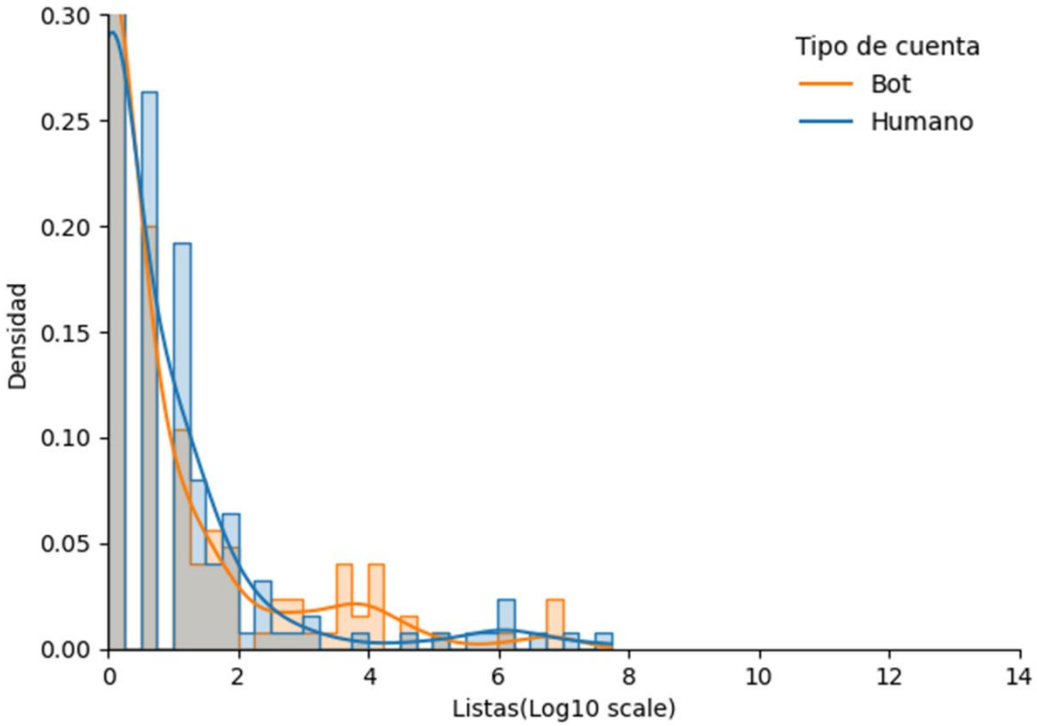



Elaboración propia.

En Twitter no existen grupos como en otras redes sociales, pero la opción de listas es utilizada para priorizar u organizar contenidos emitidos por los usuarios seleccionados, en la figura 4.11 se muestra la distribución de ambos tipos de usuarios, se puede ver que en esta característica siguen una distribución similar sesgada hacia un menor número de listas.

**Figura 4.11.**

*Distribución por densidad en escala logarítmica del número de listas por tipo de cuenta en dataset "Bot COVID-19".*

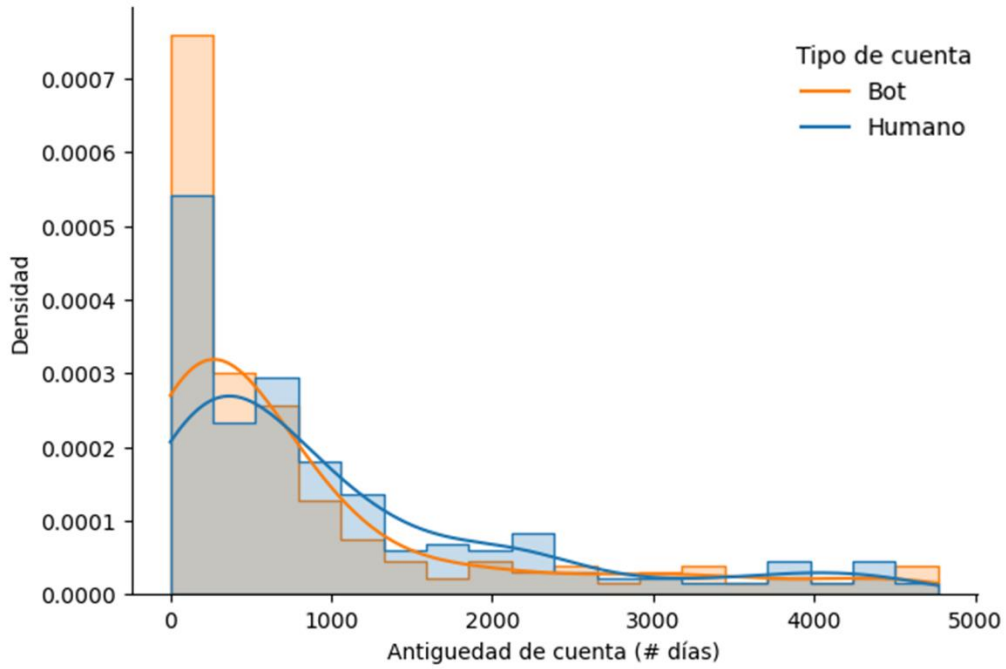



Elaboración propia.

Otra de las características es la antigüedad de la cuenta, esta métrica se calculó en número de días al momento de la extracción de los datos para el dataset bot, en la figura 4.12 se observa la densidad de la distribución, en este caso se omitió la escala logarítmica, se puede ver que ambos tipos de cuentas presentan una distribución similar, pero la cuentas tipo bot tienen un sesgo hacia un menor número de días de antigüedad.

**Figura 4.12.**

*Distribución de antigüedad de cuentas en días en dataset "Bot COVID-19".*



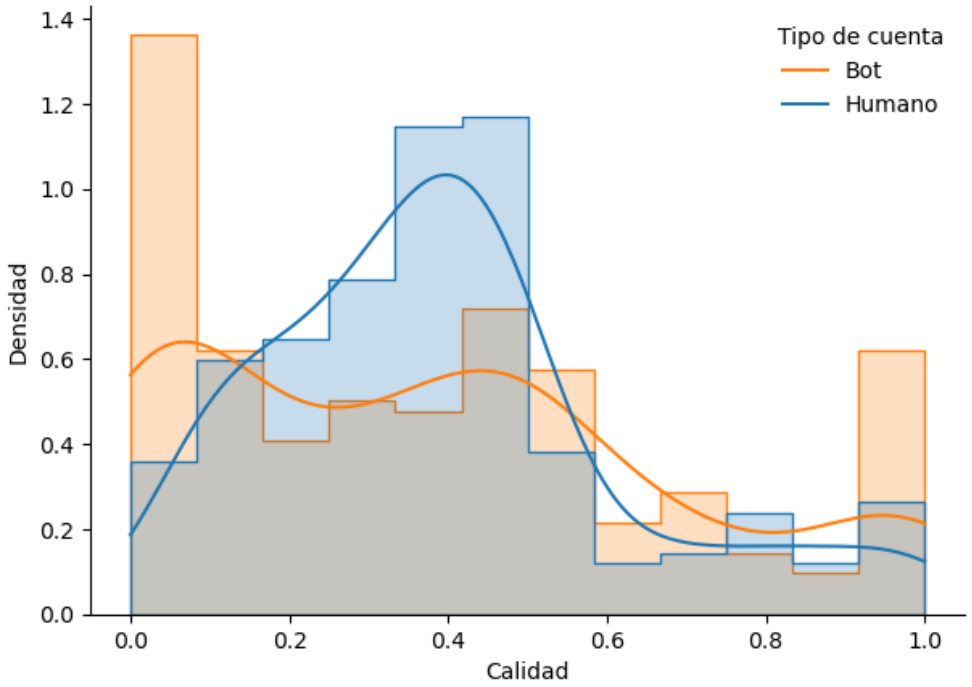

Elaboración propia.

Como hemos visto, los usuarios o cuentas en Twitter tienen la posibilidad de generar seguidores o de seguir a otras cuentas, la relación entre ambas acciones se ha denominado como reputación o índice de calidad de la cuenta, la cual ha sido utilizada como una característica importante en la detección bot, la figura 4.13 nos muestra la distribución de ambos tipos de cuentas, se puede observar que, los usuarios tipo humano tienen una

distribución más normal mientras que, los bots tienen un sesgo hacia un menor índice de calidad de cuenta.

**Figura 4.13.**

*Distribución de calidad de cuenta en dataset "Bot COVID-19".*

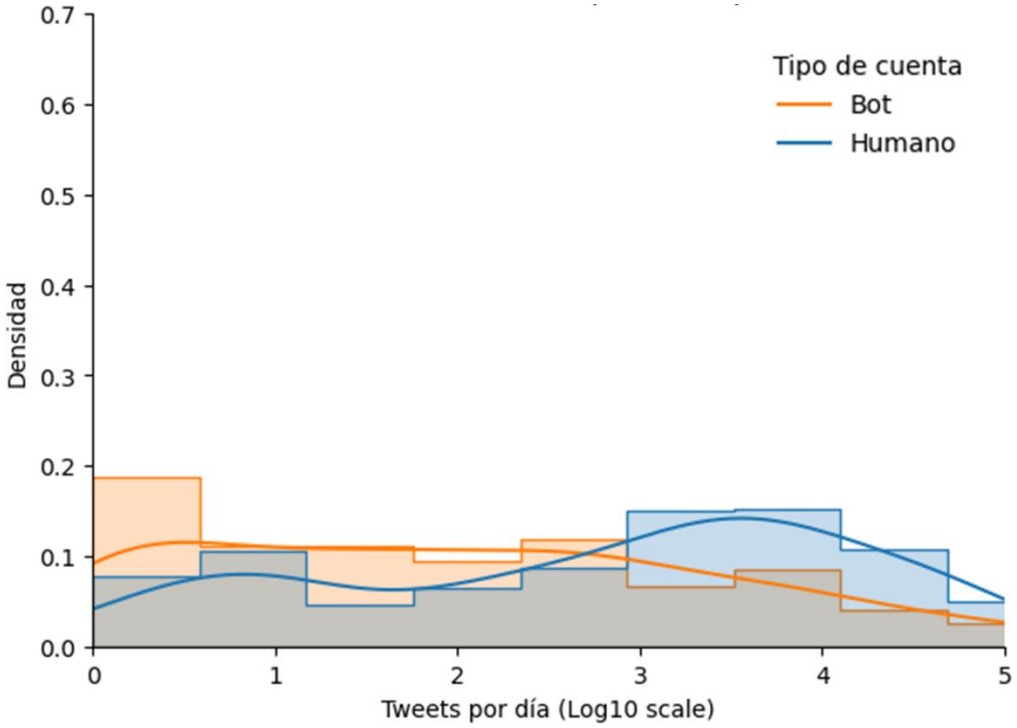



Elaboración propia

Con los datos desarrollados por el dataset "*Bot COVID-19*", se puede determinar el promedio de tweets que emiten las cuentas por día, esta métrica nos indica que tan activo es una cuenta, en la figura 4.14 podemos observar que, la distribución de las cuentas tipo humano tienen un ligero sesgo hacia un mayor número de tweets por día, mientras que, las cuentas tipo bot tienen un sesgo hacia un menor número de tweets por día.

**Figura 4.14.**

*Distribución por densidad en escala logarítmica de tweets por día por tipo de cuenta en dataset "Bot COVID-19".*

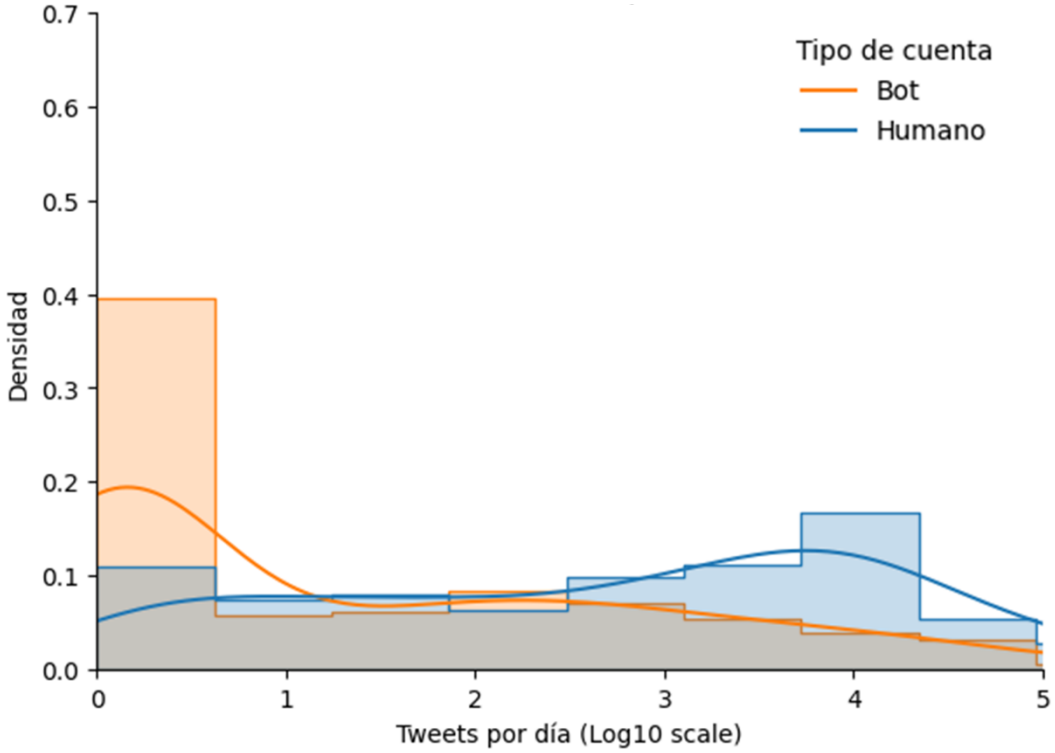



Elaboración propia.

La última característica del modelo es el número de favoritos realizaos por día, en la figura 4.15 podemos observar la distribución por tipo de cuenta, donde se puede ver que las cuentas o usuarios tipo humano tienen un sesgo hacia un mayor número de favoritos por día, mientras que las cuentas o usuarios tipo bot tienen un sesgo hacia un menor número de favoritos por día.

**Figura 4.15.**

*Distribución por densidad en escala logarítmica de favoritos por día por tipo de cuenta en dataset "Bot COVID-19".*




Elaboración propia.

### 4.8.1. Resultado de evaluación de dataset Bot COVID-19

De acuerdo con el modelo planteado en el apartado 3.5 para la detección de cuentas bot, se realizó un procesamiento del dataset "*Bot COVID-19*", para lo cual se necesita ingresar las características principales y procesar otras características de las cuentas del dataset bot, la tabla 4.11 nos muestra una parte de la salida del procesamiento de entrada de las características de los usuarios del dataset "*Bot COVID-19*", se puede observar el procesamiento de las características del usuario, que contiene la cantidad de seguidores, la cantidad de usuarios que el usuario sigue, los favoritos y

tweets emitidos por el usuario, las listas en las que esta el usuario, asi como el tiempo de la cuenta o antigüedad, además de los twees promedio por día, la calidad de la cuenta, y los seguidores, siguiendo y favoritos promedio de cada usuario.

**Tabla 4.11.**

*Muestra del resultado del procesamiento de entrada de las características de usuario.*

| Usuario | Seguidores | Siguiendo | Favoritos | Tweets | Listas | Tiempo cuenta | Tweets por día | calidad | Seguidores acq rate | Siguiendo acq rate | Favoritos rate |
|---|---|---|---|---|---|---|---|---|---|---|---|
| @usuario1 | 113 | 1850 | 150 | 8629 | 0 | 2916.7 | 3.0 | 0.514 | 0.0 | 0.491 | 0.1 |
| @usuario2 | 4130 | 549 | 101865 | 163807 | 4 | 1655.9 | 160.4 | 0.465 | 1.3 | 0.286 | 4.1 |
| @usuario3 | 838 | 1151 | 70193 | 79635 | 10 | 4346.3 | 34.5 | 0.334 | 0.2 | 0.235 | 2.8 |
| @usuario4 | 1468 | 1062 | 14674 | 97420 | 6 | 2208.1 | 50.8 | 0.417 | 0.5 | 0.393 | 2.0 |
| @usuario5 | 1468 | 182 | 14674 | 97420 | 6 | 2208.1 | 50.8 | 0.417 | 0.5 | 0.079 | 2.0 |

Elaboración propia.

Después de la entrada de los datos de las características se procede a dar entrada en las variables independientes “x” y la variable dependiente “y” para el proceso de entrenamiento de los algoritmos de regresión logística, árbol de decisión y extremo gradiente, con el siguiente pseudo-código:

**Proceso** Detección_bot
Entrada:
  Datos ← Lectura de datos dataset
  X ← Obtención de características de cada usuario
  Y ← Obtención de clasificación (Bot/humano)
Partición y entrenamiento de dataset [X,Y] en proporción 80-20
Predicción con modelos:
  Regresión logística (RL)
  Árbol de decisión (AD)
  Extremo Gradiente (XGBoost)
  Obtención de matriz de evaluación
**Fin**

En la tabla 4.12 se muestra la matriz de evaluación para el dataset bot, donde se puede observar que el algoritmo de extremo gradiente (XGBoost) obtuvo la mejor exactitud (*Accuracy)* con un valor de 85.15%, seguido del algoritmo de árbol de decisión que obtuvo un valor de 79.21%, el tercer lugar lo obtuvo el algoritmo de regresión logística con un valor del 42.57%.

**Tabla 4.12.**

*Matriz de evaluación para el dataset bot.*

| Algoritmos | Clasificación | Métricas | | | |
|---|---|---|---|---|---|
| | | Precisión | Recall | f1-score | Accuracy |
| RL | Humano | 0.42 | 1 | 0.59 | 42.57% |
| | Bot | 1 | 0.02 | 0.03 | |
| AD | Humano | 0.72 | 0.81 | 0.76 | 79.21% |
| | Bot | 0.85 | 0.78 | 0.81 | |
| XGBOOST | Humano | 0.75 | 0.95 | 0.84 | 85.15% |
| | Bot | 0.96 | 0.78 | 0.86 | |

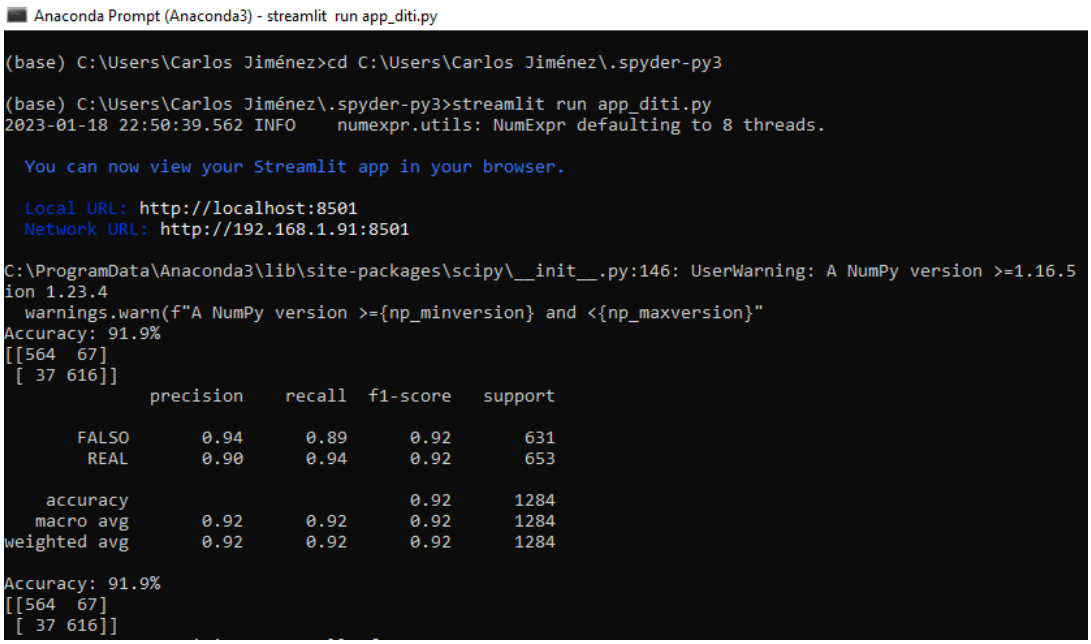
Elaboración propia.

## 4.9. Resultado de detección en línea de noticias falsas, bot y viralización

Para la implementación de la detección en línea de noticias falsas en idioma español y cuentas bot, se realizó una aplicación web en su versión de prueba, que se sustentó en los modelos descritos en los apartados 3.4 y 3.5. Para la detección de noticias falsas el mejor algoritmo fue el de regresión logística (ver apartado 4.3.13), mientras que, para la detección de cuentas bot, el mejor algoritmo fue el de extremo gradiente (XGBoost), para realizar una aplicación se ejecutó la librería de Streamlit mediante el "*Anaconda prompt*", el cual asignara una "*URL local*" y una "*Network URL*" para poder abrir el navegador web predeterminado (Figura 4.16).

**Figura 4.16.**

*Consola de Anaconda Prompt para ejecutar el código de aplicación web mediante Streamlit en Python 3.9.*

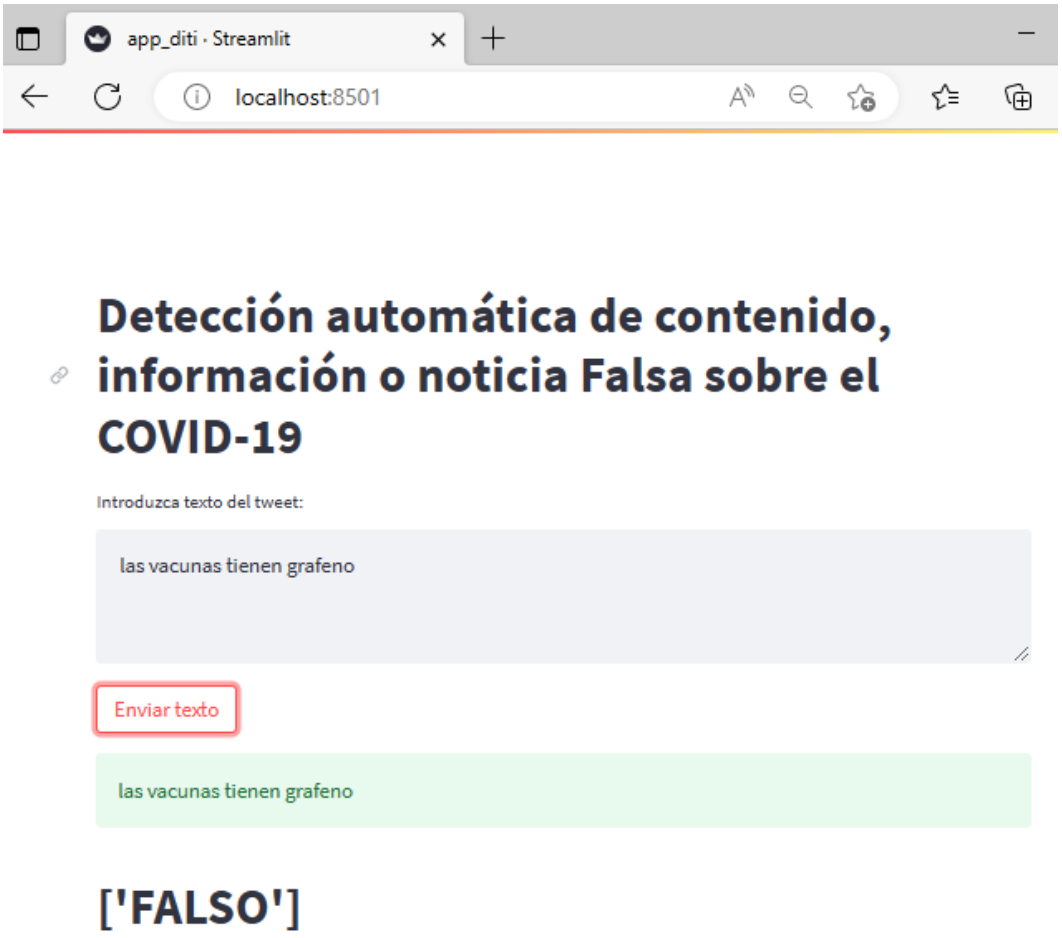



Captura de pantalla.

Tras ejecutar el archivo que contiene la aplicación para la detección en línea de noticias falsas relacionadas con el COVID-19 se abre una ventana del navegador predeterminado, en este caso en el equipo de cómputo utilizado para ejecutar los códigos se tiene como navegador predeterminado a Microsoft Edge, en la figura 4.17 se muestra una detección realizada para una noticia falsa muy común "*las vacunas tienen grafeno*" y se observa como resultado que la aplicación la detecto correctamente como noticia falsa.

**Figura 4.17.**

*Captura de pantalla de la detección de noticia falsa en la aplicación desarrollada en Streamlit.*




Captura de pantalla.

El modelo se complementó con una sección para la detección de cuentas bot y su potencial viral de acuerdo con su red de amigos de amigos del usuario en Twitter, la figura 4.18 nos muestra la captura de pantalla de la salida del procesamiento de una cuenta, en este caso la cuenta analizada resulto con clasificación de ser cuenta tipo bot con una probabilidad del 65.54%, en la figura 4.19 se puede observar la captura de pantalla de su red de amigos-amigos, además de la distribución de grado de la red y las métricas de red, como el número de nodos y vínculos, promedio de grado, *clustering*, modularidad, transitividad, distancia entre nodos y el coeficiente sigma y su clasificación de red viral o no.

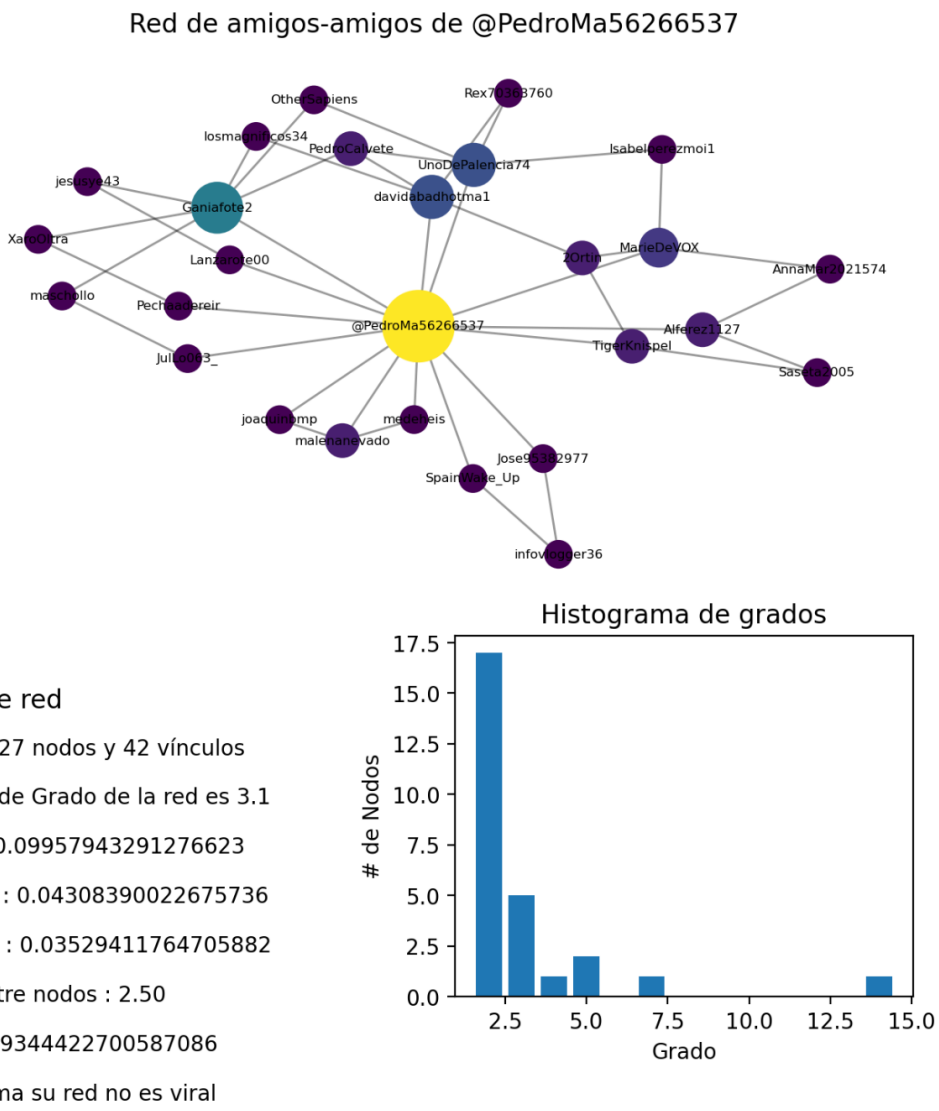
**Figura 4.18.**

Captura *de pantalla de la aplicación desarrollada en Streamlit para la detección bot en Twitter.*


Captura de pantalla.

**Figura 4.19.**

*Captura de pantalla de aplicación desarrollada en Streamlit para la detección de la red de amigos de amigos de un usuario en Twitter.*




Captura de pantalla.

# CAPÍTULO 5

# CONCLUSIONES Y TRABAJO FUTURO

Esta investigación se planteó utilizar el aprendizaje automático (*Machine learning*) para la generación de un modelo más eficiente para la detección de noticias falsas, obteniendo como producto final un modelo que puede implementarse en un sitio web para la detección automática de noticias falsas en idioma español en las redes sociales como Twitter (X.com), asi como la detección de cuentas o usuarios tipo bot, además del desarrollo de un modelo que permita determinar si es potencialmente viral.

Con el propósito de conseguir cumplir con lo planteando en el objetivo general y específicos, se desarrolló dos modelos de análisis mediante diversos algoritmos de aprendizaje automático, el primero se enfoca en las noticias falsas relacionadas con el tema de la pandemia por COVID-19, lo que implicó desarrollar un nuevo dataset de contenidos falsos y verdaderos en idioma español, además se adecuó al idioma español un dataset de noticias falsas de la iniciativa Constraint relacionados también con el tema de la pandemia y que fueron emitidos en varias redes sociales, asimismo se incluyó en la etapa de análisis un dataset de noticias falsas en español con temas diversos proveniente de la iniciativa IberLef. El segundo modelo se desarrolló

para determinar si una cuenta o usuario en Twitter tiene un perfil bot o humano, lo cual involucro el desarrollo de un nuevo dataset bot relacionado con el tema del COVID-19, paralelamente se desarrolló un sistema para determinar si su red de amigos de amigos es potencialmente viral.

El desarrollo de ambos modelos permitió la implementación de un sistema en línea (web) en una versión de prueba, capaz de detectar noticias falsas, cuentas bot y el potencial viral de un usuario en idioma español para Twitter.

## 5.1. Conclusión del modelo de detección de noticias falsas

De acuerdo con el análisis de texto de los datasets de noticias falsas, se pudo determinar que el promedio de palabras en las noticias falsas es menor al promedio de las palabras en las noticias verdaderas o reales, además, en el conteo de palabras distintas, las noticias falsas tienen un menor número que las noticias verdaderas o reales en los tres datasets analizados. El análisis de sentimiento arrojó que, las noticias falsas son más negativas llegando a un promedio entre los tres datasets del 60,1% de sentimiento negativo, un 35.0% de sentimiento positivo y un 4.9% de sentimiento neutro, mientras que, las noticias verdaderas o reales, se observó un equilibrio entre los sentimientos, dado que el sentimiento negativo promedio entre los tres datasets fue de 48.2%, el sentimiento positivo fue de 48.1% y el sentimiento neutro de 3.7%.

Para el entrenamiento de los algoritmos de aprendizaje automático se realizó una partición de 80% de los datos para entrenamiento y un 20% para prueba en los tres datasets.

La matriz de evaluación para el dataset de la iniciativa Constraint nos mostró que el algoritmo de mayor exactitud *(Accuracy)* fue el de Máquina de Vectores de Soporte (MVS) con un valor de 92.76%, en promedio los algoritmos evaluados para el dataset Constraint tuvieron un valor de 90.33% de exactitud (*Accuracy)* con una desviación estándar de 3.30%.

La matriz de evaluación para el dataset de la iniciativa IberLef mostró que, el algoritmo de Regresión Logística (RL) obtuvo la mejor exactitud (*Accuracy*) un valor de 72.80%, y en promedio los algoritmos propuestos obtuvieron un valor de 67.93% con una desviación estándar de 5.80%.

Los resultados para el dataset de DITI-Infodemia-MX muestra que la mayor exactitud fue para el algoritmo de Regresión Logística (RL) que obtuvo el valor más alto con un 78.91%, en promedio la exactitud de los algoritmos fue de 77.89% con una desviación estándar de 0.81%.

Los datos de evaluación de los dataset nos muestran que, los valores más altos de precisión fueron de la iniciativa Constraint, por lo que fue elegido como el dataset de entrenamiento para la aplicación web.

En la evaluación del mejor algoritmo y mejor dataset de noticias falsas en español se determinó que seleccionar el algoritmo y dataset con mayor promedio, en este caso el algoritmo fue el de Regresión Logarítmica (RL) y el dataset de mejor promedio fue Constraint, la exactitud del algoritmo y dataset seleccionados fue de 91.5%, superando al mejor modelo que había para la detección de noticias falsas en español que tiene un 87.18% de exactitud [33]. Por lo que se concluye que, si es posible el desarrollo de un modelo para la detección de noticias falsas en español mexicano mediante aprendizaje automático que supere en exactitud a los modelos anteriores.

## 5.2. Conclusión del modelo de detección de usuarios bot

La matriz de correlación entre las características de las cuentas tipo bot y tipo humano del dataset bot COVID-19 mostró que, si hay características con una alta correlación; en el análisis de la distribución logarítmica de seguidores se observó que las cuentas o usuarios tipo humano siguen una distribución normal, mientras que los usuarios tipo bot tienen una cantidad menor de seguidores, en la distribución logarítmica de siguiendo, las cuentas o usuarios tipo humano siguen a un mayor número de cuentas que las cuentas o usuarios tipo bot, para la distribución logarítmica de

favoritos las cuentas o usuarios tipo humano tienen un mayor número de favoritos realizados que los usuarios tipo bot, en la distribución logarítmica de tweets realizados los usuarios tipo humano presentan una ligera cantidad mayor que los usuarios tipo bot, en la distribución logarítmica de la cantidad de listas a las que pertenece una cuenta o usuario, ambos tipos bot y humanos tienen una similitud, también en la distribución de la antigüedad de las cuentas o usuarios tienen una similitud, en la distribución de la calidad de la cuenta o usuario, se observó que las cuentas o usuarios tipo humano siguen una distribución más normal que las cuentas tipo bot, mientras que en la distribución logarítmica de tweets y favoritos promedio por día, las cuentas tipo humano realizan ligeramente más tweets que las cuentas tipo bot.

Los resultados de la matriz de evaluación del dataset bot, se observó que el algoritmo de extremo gradiente (XGBoost) resulto con la mayor exactitud con un valor del 85.15%, por lo que se eligió el mejor algoritmo para el sistema en línea para la detección de cuentas tipo bot. Por lo que se concluye que, si es posible desarrollar un modelo para la detección de usuarios tipo bot mediante las características de carácter público del perfil de un usuario, mediante aprendizaje automático.

## 5.3. Conclusión de la aplicación web para la detección en línea de noticias falsas, usuarios bot y potencial viral

La aplicación de prueba para el sistema en línea se compone de dos partes, la primera permite la detección en línea de noticias falsas en español en el contexto de la pandemia por COVID-19, el texto se introduce mediante la escritura manual o mediante copiar y pegar el texto del tweet, y la salida se compone de una clasificación dicotómica, “*Falso*” para las noticias o contenido falso y “*Real*” para las noticias o contenido real o verdadero, la segunda parte de la aplicación permite la detección de cuentas tipo bot, mediante la escritura manual o copiar y pegar el usuario de Twitter, la salida

es una clasificación dicotómica que puede ser bot o humano, pero además provee la probabilidad de que una cuenta sea bot; en paralelo esta segunda parte de la aplicación precisa la red de los últimos 20 amigos de los últimos 20 amigos del usuario, además de calcular las métricas de red como el número de nodos y vínculos, también calcula el grado promedio de la red, el promedio del coeficiente de *clustering*, el coeficiente de transitividad, el índice de modularidad, el diámetro de la red, el promedio de la distancia o longitud entre los nodos de la red, y por ultimo determina el coeficiente de sigma y omega que permite establecer si una red pertenece a las redes del mundo pequeño que son altamente virales en las redes de amigos de amigos, la salida de esta sección es una leyenda que expresa si la red de amigos es potencialmente viral o no. Por los resultados obtenidos se concluye que, si es posible determinar el potencial viral de un usuario por su red de amigos de amigos.

## 5.4. Comentarios finales

Las noticias falsas han crecido de forma exponencial y se han convertido en un problema en diversas áreas de nuestra sociedad actual; en el contexto de la pandemia por COVID-19 hubo un gran flujo de información en su mayoría falsa o desinformativa la cual fue denominada infodemia por la Organización Mundial de Salud, desarrollar un modelo más eficiente para la detección de noticias falsas que pueda ayudar a contrarrestar la creciente desinformación, fue una de las motivaciones principales de la presente investigación, dentro del desarrollo de la presente investigación de crearon dos nuevos datasets uno de noticias falsas y otro de cuentas bot relacionados con el tema de la pandemia por CODI-19, de acuerdo a los resultados obtenidos de los modelos propuestos, se pudo determinar que si es posible desarrollar un modelo más eficiente para la detección automática de noticias falsas en idioma en español mexicano en las redes sociales.

La presente investigación también demostró que si es posible la detección de cuentas tipo bot en cuentas que interactuaron con el tema COVID-19 en idioma español mexicano, además de la determinación de diversas métricas que puedan pronosticar si un usuario o una cuenta es potencialmente viral.

Como parte de los productos realizados dentro de la presente investigación se enumeran dos artículos publicados:

- Towards the Comprehensive Detection of Fake News in Socio-digital Media in Mexico with Machine Learning.
  https://link.springer.com/chapter/10.1007/978-3-031-07670-1_5

- Proposal for a model for detecting fake news on social media in México.
  https://www.esociety-conf.org/wp-content/uploads/2022/03/13.2-4.pdf

Y un tercer artículo que se encuentra en su fase de revisión por la revista científica "*Data & Knowledge Engineering*" titulado:

- Detection of fake news in Spanish, bots, and viral power in social networks, in the context of the COVID-19. (En revision)

## 5.5. Trabajo futuro

La presenta investigación se fundamentó en el uso de diversos algoritmos de aprendizaje automático, como trabajo futuro se propone incrementar el número de datos que componen los datasets de noticias falsas en español y de cuentas bot; además de aplicar algoritmos de redes recurrentes y de algoritmos de aprendizaje profundo, para comparar la exactitud en la detección de noticias falsas y cuentas bot en las redes sociales.

Para la determinación de la potencia viral de un usuario en Twitter, se propone incrementar el grado de la red de amigos a un tercer grado, esto es generar, visualizar y analizar la red de amigos de amigos de los amigos del usuario y poder observar con más detalle la red de interacción de un usuario de las redes sociales.

# BIBLIOGRAFÍA

[1] Gladden, D. J. (2018). The effects of smartphones on social lives: how they affect our social interactions and attitudes.

[2] Del Vicario, M., Bessi, A., Zollo, F., Petroni, F., Scala, A., Caldarelli, G., & Quattrociocchi, W. (2016). The spreading of misinformation online. Proceedings of the national academy of Sciences, 113(3), 554-559.

[3] Bondielli, A., & Marcelloni, F. (2019). A survey on fake news and rumour detection techniques. Information sciences, 497, 38-55.

[4] Erboz, G. (2017). How to define industry 4.0: Main pillars of industry 4.0. Managerial trends in the development of enterprises in globalization era, 761, 761-767.

[5] CNDH. (2019). Reporte sobre las campañas de desinformación, noticias falsas (fake news) y su impacto en el derecho a la libertad de expresión.

[6] Hindman, M., & Barash, V. (2018). Disinformation, and influence campaigns on twitter. Knight Foundation: George Washington University.

[7] UNAM. (2020). Boletín UNAM-DGCS-318. Accesado: 23, abril,2022. Disponible en https://www.dgcs.unam.mx/boletin/bdboletin/2020_318.html.

[8] Patel, R. H., Patel, R., Patel, S., & Patel, N. (2022). Detecting fake news using machine learning. In Intelligent Data Communication Technologies and Internet of Things: Proceedings of ICICI 2021 (pp. 613-625). Singapore: Springer Nature Singapore.

[9] Collins. (2017). Collins 2017 Word of the Year Shortlist. Accesado: 2, febrero,2022. Disponible en https://blog.collinsdictionary.com/language-lovers/collins-2017-word-of-the-year-shortlist/.

[10] Google Trends. (2024). Interés a lo largo del tiempo de ‘Fake News’ en México.

[11] Stella, M., Ferrara, E., & De Domenico, M. (2018). Bots increase exposure to negative and inflammatory content in online social systems. Proceedings of the National Academy of Sciences, 115(49), 12435-12440.

[12] Vosoughi, S., Roy, D., & Aral, S. (2018). The spread of true and false news online. science, 359(6380), 1146-1151.

[13] Mønsted, B., Sapieżyński, P., Ferrara, E., & Lehmann, S. (2017). Evidence of complex contagion of information in social media: An experiment using Twitter bots. PloS one, 12(9), e0184148.

[14] Shao, C., Ciampaglia, G. L., Flammini, A., & Menczer, F. (2016). Hoaxy: A platform for tracking online misinformation. In Proceedings of the 25th international conference companion on world wide web (pp. 745-750).

[15] Davis, C. A., Varol, O., Ferrara, E., Flammini, A., & Menczer, F. (2016). Botornot: A system to evaluate social bots. In Proceedings of the 25th international conference companion on world wide web (pp. 273-274).

[16] Ahmed, S., Hinkelmann, K., & Corradini, F. (2022). Combining machine learning with knowledge engineering to detect fake news in social networks-a survey. arXiv preprint arXiv:2201.08032.

[17] Ozbay, F. A., & Alatas, B. (2020). Fake news detection within online social media using supervised artificial intelligence algorithms. Physica A: statistical mechanics and its applications, 540, 123174.

[18] Khan, J. Y., Khondaker, M. T. I., Afroz, S., Uddin, G., & Iqbal, A. (2021). A benchmark study of machine learning models for online fake news detection. Machine Learning with Applications, 4, 100032.

[19] Choraś, M., Giełczyk, A., Demestichas, K., Puchalski, D., & Kozik, R. (2018). Pattern recognition solutions for fake news detection. In Computer Information Systems and Industrial Management: 17th International Conference, CISIM 2018, Olomouc, Czech Republic, September 27-29, 2018, Proceedings 17 (pp. 130-139). Springer International Publishing.

[20] Khan, J. Y., Khondaker, M. T. I., Afroz, S., Uddin, G., & Iqbal, A. (2021). A benchmark study of machine learning models for online fake news detection. Machine Learning with Applications, 4, 100032.

[21] O'Brien, N. (2018). Machine learning for detection of fake news (Doctoral dissertation, Massachusetts Institute of Technology).

[22] Sharma, U., Saran, S., & Patil, S. M. (2020). Fake news detection using machine learning algorithms. International Journal of creative research thoughts (IJCRT), 8(6), 509-518.

[23] Zhang, J., Dong, B., & Philip, S. Y. (2020). Fakedetector: Effective fake news detection with deep diffusive neural network. In 2020 IEEE 36th international conference on data engineering (ICDE) (pp. 1826-1829). IEEE.

[24] Thota. (2018). Fake News Detection : A Deep Learning Approach. vol. 1, no. 3.

[25] Liu and Y.-F. B. Wu. (2018).Early Detection of Fake News on Social Media Through Propagation Path Classification with Recurrent and Convolutional Networks. [Online]. Available: www.aaai.org.

[26] Dabholkar, R. Kalapurackal, S. Timapur, and A. Lopes. (2024).Fake News Detection Using Machine Learning [Online]. Disponible en: https://www.propulsiontechjournal.com/index.php/journal/article/view/6026/3987.

[27] Monti, F. Frasca, D. Eynard, D. Mannion, and M. M. Bronstein. (2019). Fake News Detection on Social Media using Geometric Deep Learning. [Online]. Disponible en: http://arxiv.org/abs/1902.06673.

[28] Posadas-Durán, H. Gomez-Adorno, G. Sidorov, and J. J. M. Escoba.(2019). Detection of fake news in a new corpus for the Spanish language. *Journal of Intelligent and Fuzzy Systems*, vol. 36, no. 5, pp. 4868–4876. doi: 10.3233/JIFS-179034.

[29] Arce-Cardenas, D. Fajardo-Delgado, and M. Á. Álvarez-Carmona.(2019).TecNM at MEX-A3T 2020: Fake News and Aggressiveness Analysis in Mexican Spanish. [Online]. Disponible: http://ceur-ws.org

[30] Martínez-Gallego, A. M. Álvarez-Ortiz, and J. D. Arias-Londoño. (2021). Fake News Detection in Spanish Using Deep Learning Techniques. [Online]. Disponible: http://arxiv.org/abs/2110.06461

[31] Reyes-Magaña, “ForceNLP at FakeDeS. (2021). Analysis of Text Features Applied to Fake News Detection in Spanish.

[32] Shrestha, F. Spezzano, and A. Joy. (2020) . Detecting Fake News Spreaders in Social Networks via Linguistic and Personality Features Notebook for PAN at CLEF. [Online]. Disponible: https://docs.python.org/3/library/xml.etree.elementtree.html.

[33] Chambi-Apaza and R. Flores-Quispe. (2022). DETECTION OF FAKE NEWS IN THE SPANISH LANGUAGE USING MACHINE LEARNING TECHNIQUES. *J Theor Appl Inf Technol*, vol. 31, p. 20, [Online]. Disponible: www.jatit.org.

[34] OPS. (2020). Entender la infodemia y la desinformación en la lucha la COVID-19. Disponible: https://iris.paho.org/bitstream/handle/10665.2/52053/Factsheet-Infodemic_spa.pdf.

[35] Tashtoush, B. Alrababah, O. Darwish, M. Maabreh, and N. Alsaedi. (2022). A Deep Learning Framework for Detection of COVID-19 Fake News on Social Media Platforms. *Data (Basel)*, vol. 7. no. 5. p. 65. doi: 10.3390/data7050065.

[36] DigitalCommons, U. All Graduate Theses, and M. Rajdev. (2015). Fake and Spam Messages: Detecting Misinformation During Natural Disasters on Social Media. [Online]. Disponible: https://ieeexplore.ieee.org/document/7396773.

[37] Wang, Z. X. Tan, Y. Ye, L. Wang, K. H. Cheong, and N. Xie.(2017). OPEN A rumor spreading model based on information entropy. pp. 1–14, doi: 10.1038/s41598-017-09171-8.

[38] Visentin, G. Pizzi, and M. Pichierri.(2019). Fake News, Real Problems for Brands: The Impact of Content Truthfulness and Source Credibility on consumers’ Behavioral Intentions toward the Advertised Brands. *Journal of Interactive Marketing*, vol. 45, pp. 99–112, doi: 10.1016/j.intmar.2018.09.001.

[39] Beatriz. (2020). Procesamiento de Lenguaje Natural: una solución para detectar noticias falsas sobre la 4T en México Natural Language Processing: A solution to Detect Fake News about 4T in Mexico. ISSN: 1870-4069

[40] Martínez-Gallego, A. M. Álvarez-Ortiz, and J. D. Arias-Londoño.(2021). Fake News Detection in Spanish Using Deep Learning Techniques. [Online]. Disponible: http://arxiv.org/abs/2110.06461

[41] Villatoro-Tello, G. Ramírez-De-La-Rosa, S. Kumar, S. Parida, and P. Motlicek.(2020). Idiap and UAM Participation at MEX-A3T Evaluation Campaign. [Online]. Disponible: http://ceur-ws.org

[42] Abonizio, J. I. de Morais, G. M. Tavares, and S. B. Junior.(2020). Language-independent fake news detection: English, Portuguese, and Spanish mutual features. *Future Internet*, vol. 12, no. 5, doi: 10.3390/FI12050087.

[43] W. A. Social.(2020). DIGITAL 2020 : GLOBAL DIGITAL. pp. 1–60. Disponible en: https://datareportal.com/reports/digital-2020-global-digital-overview

[44] Morales, A. Vavilala,Benito,M y Bar-yam,Y (2017). Global patterns of synchronization in human communications.

[45] Dorsey,J. (2020). A Note from Jack Welcome to Twitter's first Global Impact Report. Disponible en: https://about.twitter.com/content/dam/about-twitter/en/company/global-impact-2020.pdf.

[46] Nbviewer.(2020). Usando la API de STREAMING de Twitter." [Online]. Disponible: https://nbviewer.org/github/GeoScripting-WUR/PythonWeek/blob/gh-pages/HarvestingRealTimeTweets.ipynb.

[47] Roesslein.J. (2020). Referencia de la API_ documentación de tweepy 3.5.0. Disponible en: https://docs.tweepy.org/en/v3.5.0/.

[48] Val,E. M. Rebollo, V. Botti.(2015). Does the Type of Event Influence How User Interactions Evolve on Twitter. doi: 10.1371/journal.pone.0124049.

[49] Foroozani.A, M. Ebrahimi.(2019). Anomalous information diffusion in social networks: Twitter and Digg. *Expert Syst Appl*, vol. 134, pp. 249–266. doi: 10.1016/j.eswa.2019.05.047.

[50] Kumar, S. Singh, G. Kaur.(2019).Fake News Detection of Indian and United States Election Data using Machine Learning Algorithm. no. 11, pp. 1559–1563, 2019, doi: 10.35940/ijitee.K1829.0981119.

[51] Moreno Ribas. (1994). Aprendizaje automático. UPC. Disponible en: https://upcommons.upc.edu/handle/2117/370324?show=full.

[52] Medhat, A. Hassan, H. Korashy.(2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113. doi: 10.1016/j.asej.2014.04.011.

[53] Barrios,J.(2020). Inteligencia Artificial y Machine Learning para todos. Disponible en: https://www.juanbarrios.com/inteligencia-artificial-y-machine-learning-para-todos/.

[54] Bouza.C.(2012). LA MINERÍA DE DATOS : ARBOLES DE DECISIÓN Y SU APLICACIÓN EN ESTUDIOS. Disponible en: https://rideca.cs.buap.mx/web/files/articulo_itBUo0uWlAaJENf.pdf.

[55] Rodrigo,J.(2020). Árboles de decisión , Random Forest , Gradient Introducción. Disponible en: https://cienciadedatos.net/documentos/33_arboles_decision_random_forest_gradient_boosting_c50.

[56] Kecman.(2014). Support Vector Machines – An Introduction Support Vector Machines – An Introduction. doi: 10.1007/10984697.

[57] Rodrigo,J.(2020). Máquinas de Vector Soporte ( Support Vector Machines , SVMs ) Introducción Hiperplano y Maximal Margin Classifier. pp. 1–42, 2020.

[58] Zhang,H.(2004). The Optimality of Naive Bayes.

[59] Jurafsky,D. y J. H. Martin.(2019). Speech and Language Processing.

[60] Jurafsky,D. y J. H. Martin.(2019). Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition Third Edition draft.

[61] Daniel,J. y Martin,J. (2021). Speech and Language Processing.

[62] Rodriguez,D. (2018)“La regresión logística - Analytics Lane”, Accesado: marzo,12,2023.[Online].Disponible en https://www.analyticslane.com/2018/07/23/la-regresion-logistica/#google_vignette.

[63] Universidad Carlos III. Introducción a la regresión logística. Disponible en: https://halweb.uc3m.es/esp/Personal/personas/amalonso/esp/bstat-tema9.pdf.

[64] Shabani, V., Havolli, A., Maraj, A., & Fetahu, L. (2023). Fake news detection using naive Bayes classifier and passive aggressive Classifier. In 2023 12th Mediterranean Conference on Embedded Computing (MECO) (pp. 1-6). IEEE.

[65] Olabe, X. B. (1998). Redes neuronales artificiales y sus aplicaciones. Publicaciones de la Escuela de Ingenieros.

[66] Larrañaga,P. (2023).Tema 8. Redes Neuronales. 930. [Online].Disponible en https://www.researchgate.net/publication/268291232_Tema_8_Redes_Neuronales.

[67] BARABASI, Albert (2002) *Linked: "The New Science of Networks*.

[68] Arquilla, J., & Ronfeldt, D. (2001). The advent of netwar (revisited). Networks and netwars: The future of terror, crime, and militancy, 1-25.

[69] *Committee on Network Science for Future Army Applications*. (2005). doi: 10.17226/11516.

[70] Kaplan and M. Haenlein. (2009).Users of the world, unite! The challenges and opportunities of Social Media. *Bus Horiz*, vol. 53, no. 1, pp. 59–68, Jan.doi: 10.1016/j.bushor.2009.09.003.

[71] Candale, C. V. (2020). PARTICULARIDADES SINTÁCTICAS DEL LENGUAJE DE LOS JÓVENES ESPAÑOLES Y RUMANOS EN LAS REDES SOCIALES. Studia Universitatis Babes-Bolyai-Philologia, 65(2), 15-28.

[72] Erboz,G.(2018). HOW TO DEFINE INDUSTRY 4 . 0 : The Main Pillars of Industry 4 . 0.

[73] Weng, L. (2014). *Information diffusion on online social networks.* (Doctoral dissertation, Indiana University).

[74] Barabási.(2007). From Network Structure. *IEEE Control Systems Magazine*, no. August, pp. 33–42, 2007, doi: 10.1109/MCS.2007.384127.

[75] Shang, Y., y Yang, Y. (2017). Clustering coefficients of large networks. Information Sciences, 382, 350-358.

[76] Newman, M. (2006).Modularity and community structure in networks. Disponible en: www.pnas.orgcgidoi10.1073pnas.0601602103.

[77] Freeman, L. C. (2002). Centrality in social networks: Conceptual clarification. Social network: critical concepts in sociology. Londres: Routledge, 1, 238-263.

[78] Aghagolzadeh, M., Barjasteh, I., & Radha, H. (2012). Transitivity matrix of social network graphs. In 2012 IEEE Statistical Signal Processing Workshop (SSP) (pp. 145-148). IEEE.

[79] Freeman, L. C. (1977). A set of measures of centrality based on betweenness. Sociometry.

[80] Goel, S., Watts, D. J., & Goldstein, D. G. (2012). The structure of online diffusion networks. In Proceedings of the 13th ACM conference on electronic commerce (pp. 623-638).

[81] Weiss, H. H. (2013). The SIR model and the foundations of public health. Materials matematics, 0001-17.

[82] Abdullah, R. N. (2021). Viral in social media: the viralor and the viralee.

[83] Dawkins, R., & Suárez, J. R. (1979). El gen egoísta (p. 77). Barcelona: Labor.

[84] Wang, L., & Wood, B. C. (2011). An epidemiological approach to model the viral propagation of memes. Applied Mathematical Modelling, 35(11), 5442-5447.

[85] Goel, S., Anderson, A., Hofman, J., & Watts, D. J. (2016). The structural virality of online diffusion. Management science, 62(1), 180-196.

[86] Liu-Thompkins, Y. (2012). Seeding viral content: The role of message and network factors. Journal of advertising research, 52(4), 465-478.

[87] Ch'ng, E. (2015). Local interactions and the emergence of a Twitter small-world network. arXiv preprint arXiv:1508.03594.

[88] Humphries, M. D., & Gurney, K. (2008). Network 'small-world-ness': a quantitative method for determining canonical network equivalence. PloS one, 3(4), e0002051.

[89] Ch'ng, E. (2015). Local interactions and the emergence of a Twitter small-world network. arXiv preprint arXiv:1508.03594.

[90] Corso, C. L. (2009). Aplicación de algoritmos de clasificación supervisada usando Weka. Córdoba: Universidad Tecnológica Nacional, Facultad Regional Córdoba.

[91] Karimi. (2024)."Confusion Matrix Some of the authors of this publication are also working on these related projects: Data Cleaning Process View project." [Online]. Available: https://www.researchgate.net/publication/355096788.

[92] "Dataset Constraint," Web. First Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situation.

[93] "Dataset IberLef."2022. Iberian Languages Evaluation Forum.

[94] “Centro de ayuda Twitter,” https://help.twitter.com/es/using-twitter/liking-tweets-and-moments.

[95] Chu, Z., Gianvecchio, S., Wang, H., & Jajodia, S. (2012). Detecting automation of twitter accounts: Are you a human, bot, or cyborg?. IEEE Transactions on dependable and secure computing, 9(6), 811-824.

[96] AprendeIA, “IDE Spyder para Python.”2020.

[97] Nielsen, F. Å. (2011). A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903.*

[98] Hackeling, G. (2017). Mastering Machine Learning with Scikit-learn. Packt Publishing, Birmingham.

[99] Ganesan, K. (2014). A Note On Constructing Domain Specific Stop Word List.

[100] Borcan, M. (2020). Tf-idf explained and python sklearn implementation. Medium.

# ÍNDICE DE FIGURAS

# ÍNDICE DE TABLAS