MANIPULACIÓN DE INFORMACIÓN Y DETECCIÓN DE BOTS EN REDES SOCIALES

Marcelo Patricio Rodríguez Fernández

LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN
DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN
UNIVERSIDAD NACIONAL DEL SUR

Tesis de Licenciatura en Ciencias de la Computación

Diciembre de 2020
Bahía Blanca

Director:
Diego C. Martínez

Índice

Índice de tablas

Índice de figuras

Capítulo 1. Introducción

En el mundo virtual, predominan las redes sociales. Plataformas como Facebook, Twitter, Instagram, Youtube son los principales medios para muchas actividades, como publicidad, comunicaciones personales, transmisiones de noticias, anuncios políticos, defensa de causas sociales y pueden tener una gran influencia en nuestra forma de pensar.

Un ejemplo destacable sobre la influencia de las redes sociales, es la elección de Trump en el año 2016. (Rodríguez-Andrés, 2018) analizó distintas investigaciones académicas e informaciones periodísticas y, profundizó en la estrategia y el uso de las redes sociales por parte del equipo de campaña de Trump. En dicha investigación, se destaca como una de las razones del éxito de Trump el papel importante que cumplieron las redes sociales, especialmente Twitter y Facebook. En el caso de Twitter, se observó que solo en Estados Unidos se escribieron un billón de tweets sobre las elecciones desde agosto de 2015 (cuando empezaron los primeros debates de las primarias) hasta el día de la votación. Trump convirtó a Twitter en una de sus principales herramientas de comunicación, y en el trabajo mencionado se señala que algunas investigaciones destacan que durante esas elecciones los mensajes de los candidatos en Twitter pudieron ser más influyentes ante los votantes que las noticias difundidas por los medios tradicionales. Con respecto a la red social Facebook, la campaña de Trump la utilizó con tres objetivos principales: recaudar fondos a través de pequeñas donaciones, difundir mensajes a públicos prioritarios y diseminar noticias. El aspecto más controvertido fue el uso de esta red para la difusión de noticias falsas a favor de Trump o en contra de Clinton, las cuales fueron muy difundidas por Facebook. Noticias falsas como el apoyo del papa Francisco a Donald Trump (ver Figura 1), que Hillary Clinton había mandado asesinar a un agente del FBI que la investigaba (ver Figura 2), o que había vendido armas al Estado Islámico (ver Figura 3) tuvieron mucha repercusión y lograron tener más impacto que las de los principales medios de comunicación y se señala que el 75 % de los estadounidenses expuestos a estas noticias falsas en Facebook las dio como verdaderas.

Figura 1: Noticia falsa sobre el Papa Francisco apoyando a Donald Trump publicada en WTOE 5 News.

Figura 2: Noticia falsa sobre el asesinato de un agente FBI publicada en Denver Guardian (sitio eliminado).

Figura 3: Noticia falsa sobre la venta de armas al Estado Islámico publicada en The Political Insider.

El manejo de las redes sociales puede ser impulsado por bots, un bot es una cuenta que produce contenido automatizado, se plantean algunas preguntas:

- ¿Somos víctimas de información errónea manipulada?
- ¿Qué pasa si las redes sociales se llenan de bots?
- ¿Se puede desarrollar una detección efectiva de bots antes de que sea demasiado tarde?

Este trabajo busca responder a estos interrogantes. El objetivo de esta tesis es detectar bots automáticamente y, para ese propósito, se implementa una aplicación web con el fin de detectar si una cuenta de Twitter es un bot o no, utilizando técnicas de Machine Learning.

Capítulo 2. Marco teórico

2.1. ¿Qué es un bot?

Es un algoritmo computacional que produce contenido automatizado e interactúa con humanos en las redes sociales, tratando de emular el comportamiento humano.

El foco más importante en el año 2020 radica en el comportamiento integral de una cuenta, no solo en si está automatizada o no. No se trata de una solución binaria de bot o no bot, hay factores intermedios que son importantes por lo que se debe centrar la atención en si el bot manipula la plataforma en donde interactúa.

2.2. Propósitos de los bots

Los bots pueden ser creandos con distintas intenciones:

- Intenciones benignas o útiles.

Ejemplo:

Bots de servicio al cliente, donde un bot puede ayudar a encontrar información sobre pedidos o reservas de viajes de manera automática. Esto es muy útil y eficiente para las empresas, más aún en tiempos de distanciamiento social.

- Intenciones maliciosas: bots que engañan, explotan y manipulan las redes sociales con rumores, spam, malware, información errónea o mentiras.

Ejemplos:

  - Bots que pueden inflar el apoyo a un candidato político durante elecciones.
  - Bots que pueden influir en la estabilidad de los mercados ya sea con un efecto positivo o negativo.
    - Para beneficiarse: bots que aumentan la “charla” en las redes sociales sobre alguna compañía.
    - Un caso para perjudicar: en 2013 un ejército de bots publicaron un rumor falso de un ataque terrorista en la Casa Blanca, lo que provocó una caída en sus bolsas.
  - Bots que explotan el delito cibernético logrando captar información privada en redes sociales.

2.3. Problemática y efecto bot

Los bots pueden dar la falsa impresión de que alguna información, independientemente su precisión, es muy popular y respaldada por muchos, ejerciendo una gran influencia en las redes sociales.

Hay múltiples casos en los que las acciones de un bot pueden influir sobre los humanos.

Pueden alterar la percepción de la influencia en redes sociales, ampliando artificialmente la audiencia de algunas personas o pueden arruinar la reputación de una empresa con fines comerciales o políticos. También se estudió que las emociones son contagiosas en las redes sociales, entonces un peligro potencial es que los bots manipulen la percepción de la realidad influyendo en el ánimo.

2.4. Evolución de los bots

Bots en su comienzo:

Solo publicaban contenido automáticamente. Fáciles de detectar, con estrategias simples como centrarse en un alto volumen de generación de contenido. Ejemplo: en 2011, un equipo de la Universidad de Texas, implementó una idea simple: crearon bots de Twitter que publicaban tweets sin sentido, los cuales a ningún humano podría llamarle la atención. Y observaron que esas cuentas captaron muchos seguidores, y que esos seguidores eran justamente bots buscando incrementar sus círculos sociales siguiendo cuentas al azar ciegamente.

Bots actualmente:

Ahora, el comportamiento de los humanos y el de los bots es más difuso, Por ejemplo, pueden sumarse a conversaciones, comentar publicaciones y responder preguntas. Para adquirir visibilidad, pueden infiltrarse en discusiones populares, generando contenido apropiado, mediante la identificación de palabras clave relevantes y la búsqueda online de información adecuada para esa conversación. Una vez que se identifica el contenido apropiado, los bots pueden producir automáticamente respuestas a través de algoritmos de lenguaje natural, posiblemente incluyendo referencias a medios o links a recursos externos.

2.5. Estado del arte: Sistemas de detección de bots

Tres clases:

- Sistemas de detección de bots basados en información de las redes sociales.

- Sistemas de detección de bots basados en crowdsourcing y aprovechamiento de la inteligencia humana.
- Métodos de machine learning basados en características que discriminan entre bots y humanos.
  - Algunos métodos son difíciles de categorizar en alguna de las tres clases mencionada ya que combinan ideas de los tres enfoques.

Ejemplos:

2.5.1. Detección de bots basada en grafos

Una entidad puede controlar múltiples bots para hacerse pasar por diferentes bots y lanzar un ataque (ataque sybil). Algunas estrategias para detectar cuentas sybil se basan en examinar la estructura de un grafo social. SibylRank (Ferrara, Varol, Davis, Menczer, & Flammini, 2016), por ejemplo, explota una característica para identificar grupos densamente interconectados, y emplea el paradigma de inocentes por asociación: una cuenta que interactúa con un usuario legítimo se considera legítima. La efectividad de esta estrategia de detección está limitada por la suposición que los usuarios legítimos se niegan a interactuar con usuarios desconocidos. El paradigma inocente por asociación produce altas tasas de falsos negativos, las redes sociales pueden tener usuarios legítimos que fueron víctimas de bots, y los bots sofisticados pueden infiltrarse de manera que sea imposible detectarlos únicamente con la información de la estructura de la red.

2.5.2. Detección de bots basados en crowdsourcing

Se exploró la posibilidad de detección humana, sugiriendo el crowdsourcing de detección de bots a grupos de trabajadores. Se probó la eficacia de los humanos para detectar cuentas de bots simplemente a partir de la información en sus perfiles, y se observó que la tasa de detección por humanos disminuye con el tiempo, pero es una muy buena técnica para un protocolo de votación por mayoría: se muestra un perfil a varios trabajadores y la opinión de la mayoría determina el veredicto de si se trata un bot o no. Esta estrategia muestra una tasa de falsos positivos cercana a cero, lo que es muy deseable.

Tres incovenientes de este enfoque:

- El crowdsourcing para la detección de bots podría funcionar si se implementa desde la etapa inicial de una red social, pero no es rentable en

redes con una gran base de datos preexistente de usuarios como Facebook y Twitter.

- El costo de mantener trabajadores "expertos" para detectar con precisión las cuentas falsas, dado que el trabajador "promedio" no se desempeña bien individualmente.
- Plantea problemas de privacidad si se expone información personal a trabajadores externos para su validación.

2.5.3. Detección de bots basada en características

La ventaja de centrarse en los patrones de comportamiento es que pueden codificarse en características y adaptarse a técnicas de machine learning. Esto permite clasificar las cuentas según su comportamiento observado, y comúnmente se utilizan diferentes clases de características para capturar el comportamiento de los usuarios:

Clase	Descripción
Red	Las características de red capturan varias dimensiones de los patrones de difusión de información. Características estadísticas se pueden extraer de redes basadas en retweets, menciones y de hashtags.
Usuario	Las características del usuario se basan en metadatos de Twitter relacionados con la cuenta, incluido el idioma, las ubicaciones geográficas y el tiempo de creación de la cuenta.
Amigos	Las características de amigos incluyen estadísticas descriptivas relacionadas a los contactos de una cuenta.
Timing	Las características del timing capturan patrones temporales de generación de contenido y consumo. Un ejemplos es el tiempo promedio entre dos publicaciones consecutivas.
Contenido	Las características del contenido se basan en claves lingüísticas calculadas a través del procesamiento del lenguaje natural. Los ejemplos incluyen la frecuencia de verbos, sustantivos y adverbios en tweets.
Sentimiento	Las características de sentimiento se crean utilizando algoritmos de análisis de sentimiento de propósito general, que incluyen felicidad, activación-dominancia-valencia y puntajes de emoción.

Tabla 1: Clases de características empleadas por sistemas de detección de bots basados en características.

Botometer (Davis, Varol, Ferrara, Flammini, & Menczer, 2016) entra en esta categoría de sistema de detección de bots, fue lanzado en 2014 y proporciona el servicio para la detección de bots en Twitter, emplea algoritmos de aprendizaje supervisados entrenados con ejemplos de comportamientos tanto humanos como de bots. El algoritmo de detección se basa en características altamente predictivas que capturan una variedad de comportamientos sospechosos y separan bien los bots de los humanos. Este sistema llega a una precisión de detección superior al 95%.

2.5.4. Combinando múltiples enfoques

Hay una necesidad de adoptar técnicas para detectar efectivamente los ataques sybil en las redes sociales.

Ejemplo: Renren Sybil (Ferrara, Varol, Davis, Menczer, & Flammini, 2016), un sistema que explora múltiples dimensiones de los comportamientos de los usuarios. Por ejemplo, examina los datos del flujo de clicks y muestra que los usuarios reales pasan más tiempo enviando mensajes y mirando contenido de otros usuarios, mientras que las cuentas sybil pasan tiempo recolectando perfiles y haciéndose amigos de otras cuentas. Al identificar también características altamente predictivas, como la frecuencia de invitaciones, las solicitudes salientes aceptadas y el coeficiente de agrupación de red, Renren puede clasificar las cuentas en perfiles prototípicos de bots o humanos.

Las cuentas Sybil en Renren tienden a trabajar juntas para difundir contenido similar. El enfoque Renren combina ideas de los tres enfoques mencionados, y se logran buenos resultados incluso teniendo en cuenta solo los últimos 100 clicks para cada usuario. Se ajustan los parámetros y se entrenan con un número fijo de cuentas reales conocidas para luego poder usar el clasificador.

2.6. ¿Podemos confiar en la información de las redes sociales?

Esto es lo que se cuestiona en (Paquet-Clouston, Bilodeau, & Décary-Hétu, 2017). Se cree que el tamaño de la audiencia de una cuenta de redes sociales, en términos de seguidores o amigos, es una buena medida de su influencia y popularidad.

En la mayoría de los casos, la atracción de nuevos seguidores y amigos se realiza mediante la publicación de contenido interesante. Sin embargo, en algunos casos, los usuarios eligen comprar su base de fans, una estrategia que forma parte de lo que se llama fraude en las redes sociales.

El fraude en las redes sociales es el proceso de crear likes, seguidores, vistas o cualquier otra acción en redes como Facebook, Twitter, YouTube e Instagram para aumentar artificialmente la base de amigos/seguidores de una cuenta. Por supuesto, tener miles de seguidores en una red social no se traduce directamente en influencia sobre los seguidores, pero crea una ilusión de popularidad que el usuario puede aprovechar. Este método falsifica los datos de las redes sociales y crea desinformación que podría conducir a una disminución de la confianza de los usuarios en las redes.

Los servicios de fraude se basan en una de dos estrategias para obtener el control sobre las cuentas que utilizan para aumentar artificialmente el número de seguidores de sus clientes:

1) Comprometer cuentas reales existentes: se logra al atraer a los usuarios para que hagan click en enlaces con la promesa de seguidores gratis, pero en cambio comprometen las credenciales de sus cuentas y las usan para dar me gusta, seguir o ver otras cuentas.
2) Crear cuentas nuevas y falsas: consiste en crear nuevas cuentas falsas y usarlas para generar seguimientos, me gusta o vistas.
   Se pueden crear grandes cantidades de cuentas falsas para el fraude en redes sociales utilizando granjas de clics o botnets.
   Las granjas de clicks consisten en grandes grupos de trabajadores con salarios bajos, habitualmente establecidos en países en desarrollo, contratados para realizar el fraude a pedido.
   Las botnets, por otro lado, son grupos de ordenadores controlados remotamente por un tercero denominado “maestro de bot”. Usando este sistema, las cuentas falsas se crean automáticamente y se utilizan para realizar likes, seguimientos y vistas.

Sin embargo, para evitar la creación de miles de cuentas falsas, las redes sociales han desarrollado varias técnicas para evitar grandes creaciones de cuentas automatizadas por botnets:

- uso de CAPTCHAS
- verificación de números de teléfono
- blacklist de IP.

Si bien las barreras de registro planteadas por las redes sociales son evadidas rutinariamente por los servicios de fraude de manera automatizada, por ejemplo, simplemente generando direcciones de correo electrónico aleatorias sobre la marcha, se cree que representan uno de los mayores desafíos que enfrentan los transgresores cuando realizan el fraude.

2.6.1. El precio del fraude en las redes sociales

Se puede acceder a cientos de sitios web especializados en ofrecer estos servicios fraudulentos mediante la búsqueda de "comprar likes" o "comprar seguidores" en los motores de búsqueda populares.

La mayoría de los sitios web que anuncian servicios de fraude en las redes sociales ofrecen conjuntos diversificados de servicios para varias redes sociales, incluidos tweets en Twitter o me gusta en Instagram. Algunos servicios son más caros que otros, por ejemplo, comprar un comentario es más costoso que comprar un like. Algunas redes sociales y el tamaño del servicio (número de me gusta) también afectan los precios.

En la siguiente tabla se presenta el precio medio de comprar 1,000 seguidores y comprar 1000 likes para cuatro redes sociales populares: Facebook, Instagram, Twitter y YouTube.

	$USD / 1,000 seguidores	$ USD / 1,000 likes
Facebook	$29	$20
Instagram	$13	$14
Twitter	$12	$15
YouTube	$51	$50

Tabla 2: Precio de seguidores y likes en el mercado ilícito de los servicios de fraude en las redes sociales.

La diferencia de precios puede estar relacionada con los desafíos que enfrentan los proveedores cuando intentan realizar el fraude sin ser detectados.

2.6.2. Perfil de la demanda de servicios de fraude

Los clientes potenciales del servicio de fraude en redes sociales parecen ser entidades, individuos, empresas y empresarios, que pueden no tener el tiempo y los recursos necesarios para ganar una gran base de seguidores a través de sofisticadas estrategias de marketing. Sin embargo, dado que valoran la popularidad en redes sociales, recurren al mercado ilícito para obtener una gran base de seguidores a bajo costo.

Los servicios de fraude merecen atención por parte de la comunidad investigadora por tres razones:

- Representa costos importantes para las empresas de redes sociales, que necesitan desarrollar algoritmos y herramientas para detectar actividades fraudulentas.

- Crea datos sesgados y aumenta artificialmente la popularidad de ciertas cuentas. Esto impulsa la desinformación. Cuando las personas son engañadas y mal informadas, su confianza establecida con las redes disminuye.
- El suministro del fraude está impulsado en parte por botnets que contaminan la infraestructura de Internet. No hay que pasar por alto que una botnet podría realizar actividades ilícitas como robar credenciales.

2.7. Manipulación y personalización maliciosa

La detección de cuentas y mensajes engañosos es un primer intento de proteger a los usuarios de las experiencias online perjudiciales. Además, es un paso importante para garantizar interacciones más seguras a través de las redes sociales. Sin embargo, los ataques se están volviendo más sofisticados y, las personas pueden ser engañadas para revelar información personal cuando los incentivos superan sus preocupaciones de privacidad. Esto exige herramientas de sensibilización más efectivas, ya que estos instrumentos juegan un papel clave en el apoyo a los usuarios al tomar decisiones de privacidad online. Por ejemplo, investigadores proponen el uso de patrones de riesgo para alertar a los usuarios cuando están a punto de revelar información privada dentro de publicaciones en redes sociales. Sin embargo, hasta donde sabemos, no se hizo mucho esfuerzo para informar a los usuarios sobre los riesgos de revelar información personal.

El uso de intervenciones (es decir, mensajes de advertencia o sugerencias) es un enfoque prometedor para incentivar a los usuarios a proteger su privacidad. Sin embargo, también se ha demostrado que tales intervenciones pueden resultar molestas para los usuarios con pocas preocupaciones de privacidad. Por lo tanto, las advertencias deben estar alineadas de alguna manera con los objetivos y expectativas de privacidad de cada usuario individual.

2.8. Manipulación y fake news

La manipulación en las redes sociales toma muchas formas, una de ellas son las *fake news*.

El término "fake news" se refiere tanto al globalmente hablado espacio informativo post-verdad, como a piezas de información que se difunden intencionalmente, mientras se sabe que son falsas.

Más allá de este término, en (Gadek, Justine, & Everwyn, 2019) se remarca que el problema real se refiere a la manipulación de la información y a las muchas formas de mezclar intención, información (por ejemplo, mensajes) y conocimiento (por ejemplo, hechos).

Las redes sociales son un entorno primario para la propagación de noticias falsas

Siempre existe cierta incertidumbre sobre la legitimidad de las opiniones, ya que pueden constituir una intención de desestabilización durante una campaña de manipulación.

Varios enfoques intentan identificar manipulaciones, ya sea a través del análisis de contenido o estructura (por ejemplo, clickbait), mientras que otros se centran en la topología social de la propagación de opinión. Todas estas técnicas permiten a los analistas calificar a los emisores y propagadores de información y sus patrones, lo que finalmente da como resultado una puntuación de credibilidad.

Un segundo enfoque utilizado para activar la campana de "manipulación" se basa en verificar la probabilidad de cada pieza de información. Si bien las noticias falsas parecen reales a primera vista, a menudo no resisten contra una mente bien informada.

2.8.1. Ejemplos

Las manipulaciones de las redes sociales atribuidas a Rusia

Las sospechas muy fuertes sobre la participación de los servicios rusos en la conducción de la campaña presidencial de Estados Unidos de 2016, que resultó en la elección de Donald Trump.

Se han detectado al menos dos operaciones concretas: el hackeo de la base de datos del servidor de correo del Partido Demócrata, revelando el modo de funcionamiento interno del partido en medio de la campaña, y el uso masivo de cuentas falsas en las redes sociales Facebook, Twitter y Reddit. Estas cuentas falsas propagaron contenido emitido por medios de comunicación dudosos: incluso si algunos de estos sitios estaban motivados únicamente por dinero, como estos empresarios macedonios que diseñan sitios de clickbait, otros parecen estar más estrechamente conectados con el Kremlin, que tiene una larga historia de estrategia ciberespacial.

Otro tipo de acción sospechosa pasa por la distribución de anuncios en las redes sociales, y especialmente en Facebook a través de memes. El Partido Demócrata hace pública una breve lista de tales anuncios, y sugiere que los anuncios fueron pagados por “empresas cercanas al Kremlin”.

También aparecieron rastros menos convencionales, como una lista de cuentas de Twitter, banneadas pero etiquetadas como cuentas falsas manejadas desde Rusia. Esta atribución sigue siendo negada por el Kremlin y, a nivel mundial, no se puede verificar. La propia existencia de una agencia de desinformación rusa parece formar parte de la estrategia de poder.

Sin embargo, el impacto de estas cuentas "rusas" es real, ya que han podido aparecer constantemente en los principales medios de comunicación: a menudo en sitios de noticias cuestionables (Telegraph, Buzzfeed), pero también en referencias (BBC, The Guardian). Cabe aclarar que, hasta ahora, no se pudo establecer ninguna prueba "real" de la participación del Estado ruso.

Cambridge Analytica

La compañía Cambridge Analytica afirmó (sin pruebas) haber inclinado la votación para "Leave" en el referéndum para Brexit en junio de 2016. Entre sus herramientas aparece la distribución masiva de contenido en las redes sociales para imponer su narrativa al público.

2.9. Introducción a Machine Learning

2.9.1. Definición

[Machine Learning es el] campo de estudio que brinda a las computadoras la capacidad de aprender sin ser programadas explícitamente.

—Arthur Samuel, *1959*

Y una definición más orientada a la ingeniería:

Se dice que un programa de computadora aprende de la experiencia E con respecto a alguna tarea T y alguna medida de performance P, si su desempeño en T, medido por P, mejora con la experiencia E.

—Tom Mitchell, *1997*

Ejemplo:

Por ejemplo, el filtro de correo no deseado es un programa de machine learning o aprendizaje automático que puede aprender a marcar el correo no deseado dados

ejemplos de correos electrónicos no deseados (por ejemplo, reportados por los usuarios) y ejemplos de correos electrónicos regulares (no spam). Los ejemplos que utiliza el sistema para aprender se denominan conjunto de entrenamiento. Cada ejemplo de entrenamiento se denomina instancia de entrenamiento (o muestra). En este caso, la tarea T es marcar spam para nuevos correos electrónicos, la experiencia E son los datos de entrenamiento y la medida de performance P debe definirse; por ejemplo, puede utilizar la proporción de correos electrónicos clasificados correctamente. Esta medida de performance en particular se llama exactitud y se usa a menudo en tareas de clasificación.

2.9.2. ¿Por qué utilizar machine learning?

El aprendizaje automático es ideal para:

- Problemas para los cuales las soluciones existentes requieren muchos ajustes manuales o largas listas de reglas: un algoritmo de aprendizaje automático a menudo puede simplificar el código y funcionar mejor.

  Ejemplo:

  Un filtro de spam basado en técnicas de aprendizaje automático aprende automáticamente qué palabras y frases son buenos predictores de spam detectando patrones de palabras inusualmente frecuentes en los ejemplos de spam en comparación con los ejemplos de correos regulares. Un programa con este enfoque, es mucho más corto, más fácil de mantener y probablemente más preciso que un programa que no haga uso del aprendizaje automático.
- Problemas complejos para los que no existe una buena solución utilizando un enfoque tradicional: las mejores técnicas de Machine Learning pueden encontrar una solución.

  Ejemplo:

  Reconocimiento de voz, la mejor solución (al menos hoy) es escribir un algoritmo que aprenda por sí mismo, dados muchos ejemplos de grabaciones para cada palabra.
- Entornos fluctuantes: un sistema de aprendizaje automático puede adaptarse a nuevos datos.
- Obtener conocimientos sobre problemas complejos y grandes cantidades de datos.

2.9.3. Tipos de sistemas de aprendizaje automático

Teniendo en cuenta (Géron, 2019), hay diferentes tipos de sistemas de aprendizaje automático que se pueden clasificar en categorías según:

- Si están entrenados o no con supervisión humana (supervisado, no supervisado, semisupervisado y aprendizaje reforzado)
- Si pueden aprender o no de forma incremental sobre la marcha (aprendizaje en línea o por lotes)
- Ya sea que funcionen simplemente comparando nuevos puntos de datos con puntos de datos conocidos o, en su lugar, detecten patrones en los datos de entrenamiento y creen un modelo predictivo (aprendizaje basado en instancias o aprendizaje basado en modelos).

Estos criterios no son exclusivos; pueden combinarse de la manera que el problema a resolver lo requiera.

2.9.3.1. Aprendizaje supervisado / no supervisado

Los sistemas de aprendizaje automático se pueden clasificar según la cantidad y el tipo de supervisión que reciben durante el entrenamiento. Hay cuatro categorías principales: aprendizaje supervisado, aprendizaje no supervisado, aprendizaje semisupervisado y aprendizaje reforzado.

Aprendizaje supervisado

En el aprendizaje supervisado, los datos de entrenamiento que alimentan al algoritmo incluyen las soluciones deseadas, llamadas etiquetas.

En la Figura 4 se muestra un conjunto de entrenamiento etiquetado para el aprendizaje supervisado (por ejemplo, clasificación de spam).

Una tarea típica de aprendizaje supervisado es la *clasificación*. El filtro de spam es un buen ejemplo de esto: está entrenado con muchos correos electrónicos de ejemplo junto con su clase (spam o regular), y debe aprender a clasificar nuevos correos electrónicos.

Otra tarea típica es predecir un valor numérico como objetivo, como el precio de un automóvil, dado un conjunto de características (kilometraje, antigüedad, marca, etc.) llamadas predictores. Este tipo de tarea se llama *regresión*. Para entrenar el sistema, debe darle muchos ejemplos de automóviles, incluyendo tanto sus predictores como sus etiquetas (es decir, sus precios).

Algunos de los algoritmos de aprendizaje supervisado son:

- K-vecinos más cercanos
- Árboles de decisión
- Redes neuronales

Figura 4: Conjunto de entrenamiento para aprendizaje supervisado (ejemplo: correo electrónico no deseado, spam).

Aprendizaje no supervisado

En el aprendizaje no supervisado, los datos de entrenamiento no están etiquetados.

Los algoritmos de aprendizaje no supervisados más importantes entran en alguna de estas categorías:

- Agrupamiento (clustering):

  Consiste en la agrupación automática de datos.

  Por ejemplo, supongamos que se tienen muchos datos sobre los visitantes de un blog y se desea ejecutar un algoritmo de agrupación para intentar detectar grupos de visitantes similares como se muestra en la Figura 5. En ningún momento se le dice al algoritmo a qué grupo pertenece un visitante: encuentra esas conexiones sin esa ayuda. Por ejemplo, puede notar que el 40% de sus visitantes son hombres que aman las historietas y generalmente leen el blog por la noche, mientras que el 20% son jóvenes amantes de la ciencia ficción que visitan el blog los fines de semana, y así sucesivamente. Si usa un algoritmo de agrupamiento jerárquico, también se puede subdividir cada grupo en grupos más pequeños. Esto puede ayudarlo a orientar las publicaciones para cada grupo.

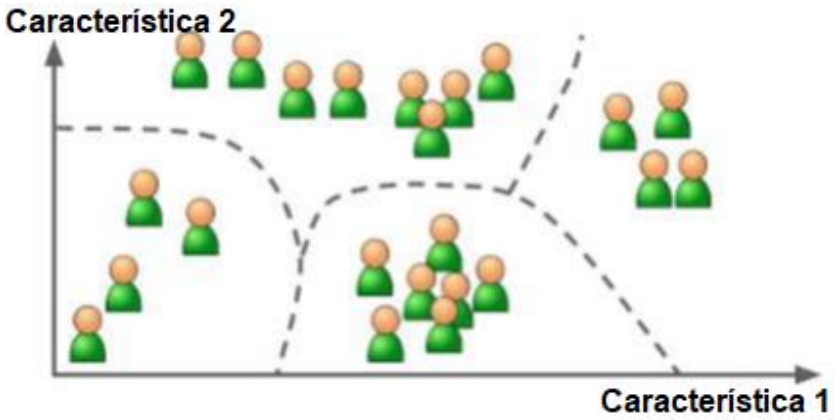

Figura 5: Agrupamiento por características.

- Detección de anomalías

  Los algoritmos de detección de anomalías buscan detectar valores anormales, por ejemplo, detectar transacciones inusuales de tarjetas de crédito para evitar fraudes, detectar defectos de fabricación o eliminar automáticamente los valores atípicos de un conjunto de datos antes de enviarlos a otro algoritmo de aprendizaje.

  Como se muestra en la Figura 6, Al sistema se le proporciona en su mayoría instancias normales durante el entrenamiento, por lo que aprende a reconocerlas y cuando ve una nueva instancia puede saber si se ve como una normal o si es probable que sea una anomalía.

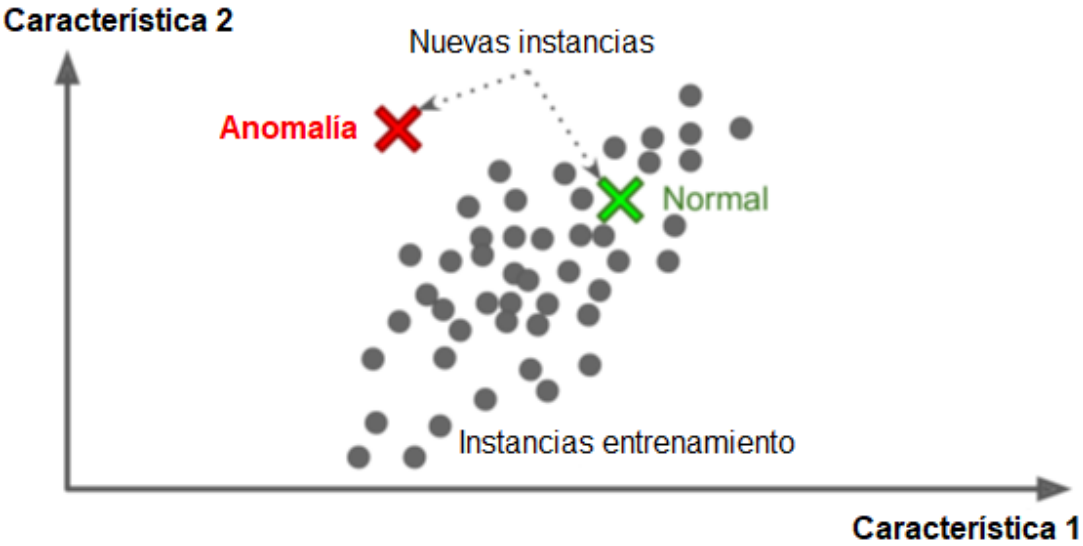

Figura 6: Detección de anomalías.

- Visualización y reducción de dimensionalidad:

  Los algoritmos de visualización también son buenos ejemplos de algoritmos de aprendizaje no supervisados: se alimenta con una gran cantidad de datos

complejos y sin etiquetar, y generan una representación 2D o 3D de sus datos q. Estos algoritmos intentan preservar tanta estructura como pueden , para que pueda comprender cómo se organizan los datos y tal vez identificar patrones inesperados.

La reducción de dimensionalidad, en la que el objetivo es simplificar los datos sin perder demasiada información. Una forma de hacer esto es fusionar varias características relacionadas en una. Por ejemplo, el kilometraje de un automóvil puede estar muy relacionado con su antigüedad, por lo que el algoritmo de reducción de dimensionalidad los fusionará en una característica que representa el desgaste del automóvil. A esto se le llama extracción de características.

- Aprendizaje de reglas de asociación:

  El objetivo del aprendizaje de reglas de asociación es profundizar en grandes cantidades de datos y descubrir relaciones interesantes entre los atributos o características. Por ejemplo, en un supermercado la ejecución de una regla de asociación en sus registros de ventas puede revelar que las personas que compran milanesas y papas fritas también tienden a comprar mayonesa, por lo tanto, es posible que desee colocar esos productos cerca unos de otros.

2.9.3.2. Aprendizaje semisupervisado

Algunos algoritmos pueden tratar con datos de entrenamiento parcialmente etiquetados, generalmente muchos datos sin etiquetar y un poco de datos etiquetados. Esto se llama aprendizaje semisupervisado.

En la Figura 7, se muestra un ejemplo de aprendizaje semisupervisado, donde se presentan una gran cantidad de datos sin etiquetar, algunos datos clasificados como “Triángulo” y otros pocos datos clasificados como “Cuadrado”, una nueva instancia deberá clasificarse en base a esos datos de entrenamiento.

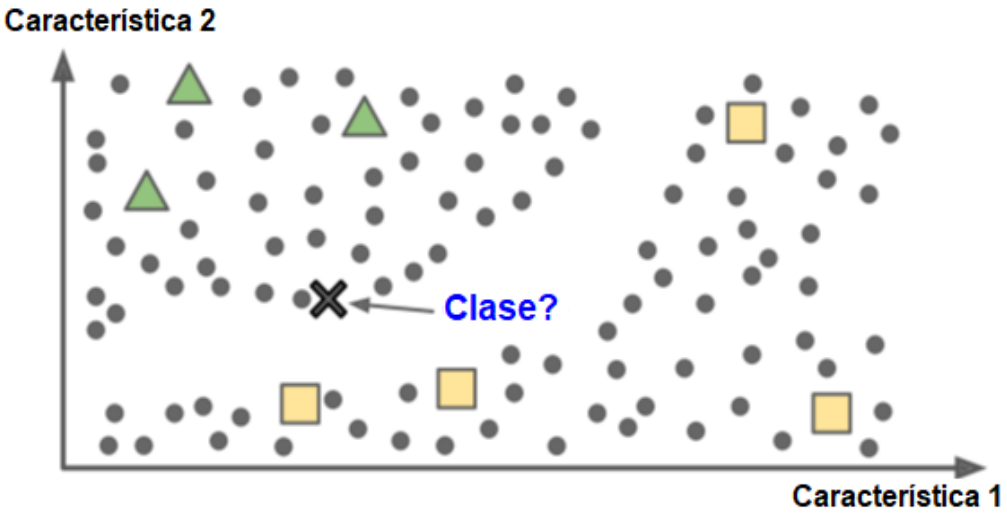

Figura 7: Aprendizaje semisupervisado.

Ejemplo: Un servicio de alojamiento de fotos, como Google Photos. Una vez que suben todas las fotos al servicio, reconoce automáticamente que la misma persona A aparece en las fotos 1, 5 y 11, mientras que otra persona B aparece en las fotos 2, 5 y 7. Esta es la parte no supervisada del algoritmo (agrupamiento). Ahora todo lo que necesita el sistema es que le digas quiénes son estas personas, solo una etiqueta por persona, y puede nombrar a todos en cada foto, lo cual es útil para buscar fotos.

2.9.3.3. Aprendizaje reforzado

En el aprendizaje reforzado, el sistema de aprendizaje, llamado agente en este contexto, puede observar el entorno, seleccionar y realizar acciones y obtener recompensas a cambio (o penalizaciones en forma de recompensas negativas). Luego, debe aprender por sí mismo cuál es la mejor estrategia, llamada política, para obtener la mayor recompensa a lo largo del tiempo. Una política define qué acción debe elegir el agente cuando se encuentra en una situación determinada.

Por ejemplo, muchos robots implementan algoritmos de aprendizaje reforzado para aprender a caminar.

En la Figura 8, se muestra el proceso en el cual un robot hace uso de aprendizaje reforzado para aprender a caminar en un entorno donde hay fuego y agua, teniendo una penalidad si se acerca al fuego, luego de caminar hacia el fuego o agua, se obtiene una recompensa y se actualiza la política.

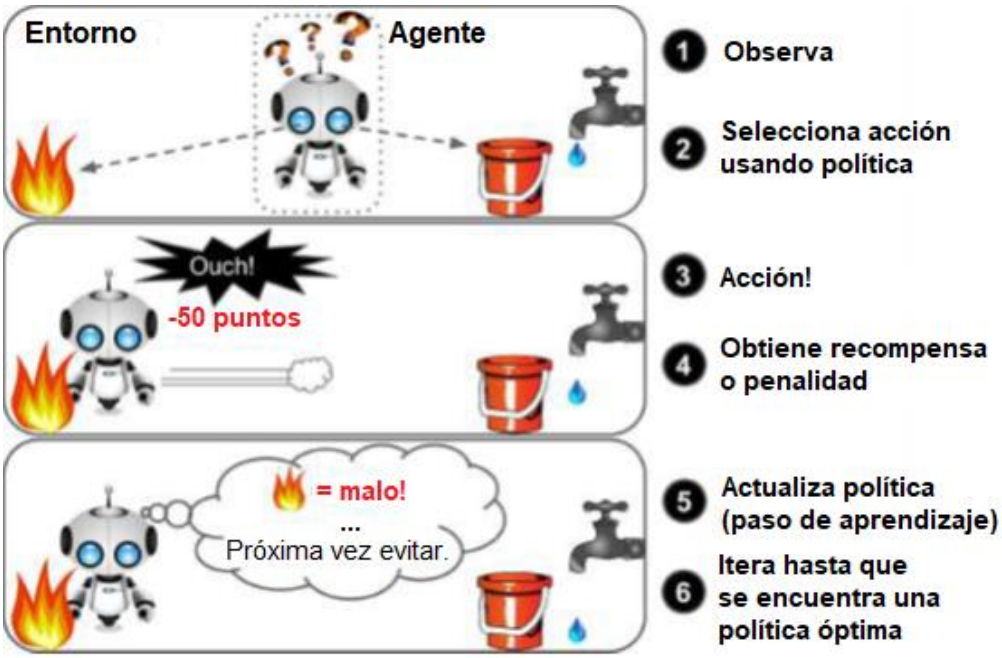

Figura 8: Aprendizaje reforzado.

2.9.3.4. Aprendizaje por lotes y en línea

Otro criterio utilizado para clasificar los sistemas de aprendizaje automático es si el sistema puede aprender de forma incremental a partir de un flujo de datos entrantes.

Aprendizaje por lotes

En el aprendizaje por lotes, el sistema es incapaz de aprender de forma incremental: debe entrenarse utilizando todos los datos disponibles. Por lo general, esto requerirá mucho tiempo y recursos informáticos, por lo que generalmente se realiza sin conexión. Primero se entrena el sistema, luego se lanza a producción y se ejecuta sin más aprendizaje; simplemente aplica lo que ha aprendido. A esto se le llama aprendizaje fuera de línea.

Aprendizaje en línea

En el aprendizaje en línea, se entrena al sistema de manera incremental al alimentarlo con instancias de datos de manera secuencial, ya sea individualmente o por grupos pequeños llamados mini lotes. Cada paso de aprendizaje es rápido y económico, por lo que el sistema puede aprender sobre nuevos datos sobre la marcha, a medida que llegan.

El aprendizaje en línea es excelente para los sistemas que reciben datos como un flujo continuo (por ejemplo, precios de las acciones) y necesitan adaptarse a los cambios de

manera rápida o autónoma. También es una buena opción si tiene recursos de hardware limitados.

2.9.3.5. Aprendizaje basado en instancias vs. aprendizaje basado en modelos

Una forma más de categorizar los sistemas de aprendizaje automático es cómo se generalizan.

Hay dos enfoques principales para la generalización: aprendizaje basado en instancias y aprendizaje basado en modelos.

Aprendizaje basado en instancias

En el aprendizaje basado en instancias: el sistema "memoriza" los ejemplos del conjunto de entrenamiento y su clase, luego para generalizar, lo realiza en base a una medida de similitud de la instancia a clasificar con los ejemplos memorizados.

Por ejemplo, en la Figura 9, la nueva instancia se clasificaría como un triángulo porque la mayoría de las instancias más similares pertenecen a esa clase.

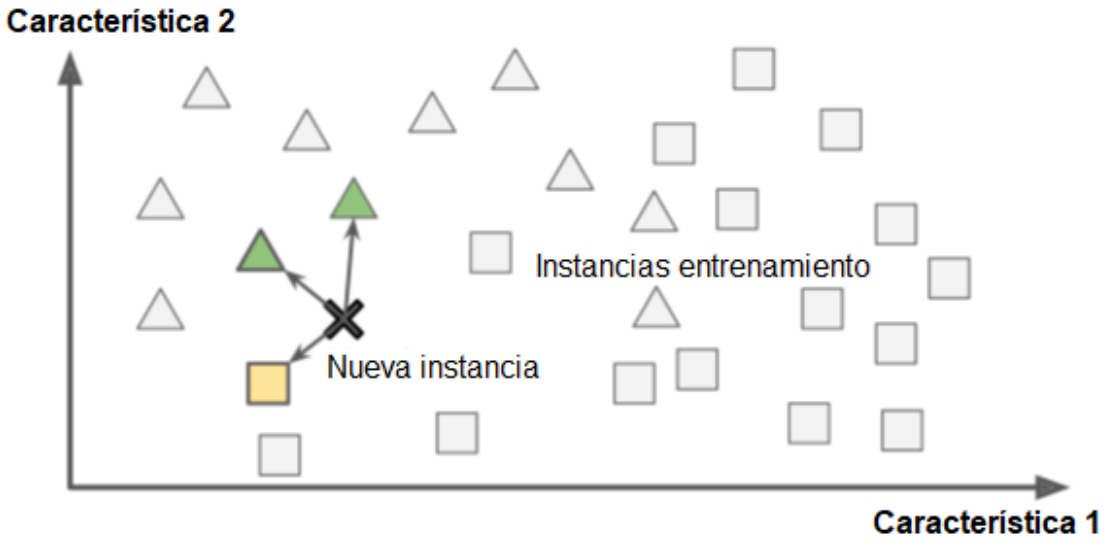

Figura 9: Nueva instancia en aprendizaje basado en instancia

Aprendizaje basado en modelos

Otra forma de generalizar a partir de un conjunto de ejemplos es construir un modelo de estos ejemplos y luego usar ese modelo para hacer predicciones, esto se llama aprendizaje basado en modelos.

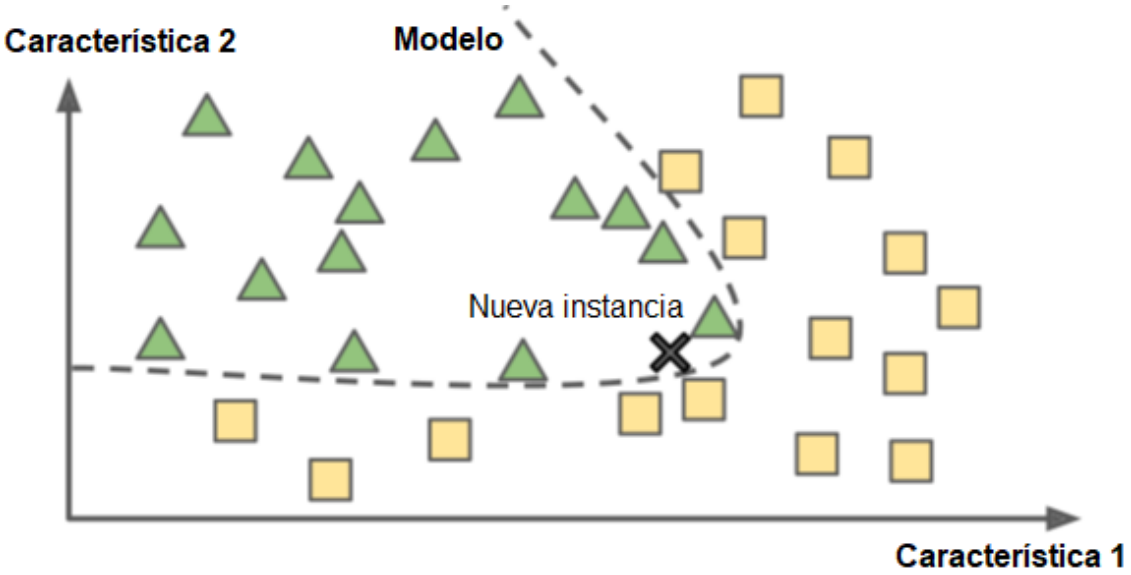

Figura 10: Nueva instancia en aprendizaje basado en modelos

2.9.4. Proceso del aprendizaje automático

Como indica (Chamorro Alvarado, 2018), construir un modelo de aprendizaje automático, no se reduce solo a utilizar un algoritmo de aprendizaje automático; sino que es todo un proceso que suele involucrar los pasos que se muestran en Figura 11:

Figura 11: Proceso de aprendizaje automático.

2.9.1. Recolectar los datos

Recolectar los datos consiste en juntar, agrupar o conseguir los diferentes datos a utilizar, podemos recolectar los datos desde diversas fuentes, por ejemplo, datos de un sitio web, utilizando una API o desde una base de datos. Podemos también utilizar otros dispositivos que recolectan los datos por nosotros; o utilizar datos que son de dominio público. El número de opciones que tenemos para recolectar datos es sumamente diverso. Esta parte del proceso parece obvia, pero es uno de los más complicados y conlleva mucho tiempo.

2.9.2. Pre-procesar los datos

Pre-procesar los datos, consiste en que una vez que tenemos los datos, los datos obtenidos tengan el formato correcto para alimentar el algoritmo de aprendizaje. Es prácticamente inevitable tener que realizar varias tareas de preprocesamiento antes de utilizar los datos.

Igualmente, este punto suele ser mucho más sencillo que el paso anterior.

2.9.3. Explorar los datos

Finalizados los pasos anteriores, podemos realizar un preanálisis para corregir los casos de valores faltantes o intentar encontrar a simple vista algún patrón en los mismos que nos facilite la construcción del modelo. En esta etapa suelen ser de mucha utilidad las medidas estadísticas y los gráficos en 2 y 3 dimensiones para tener una idea visual de cómo se comportan nuestros datos. En este punto podemos detectar valores atípicos que debamos descartar; o encontrar las características que poseen mayor influencia para realizar una predicción.

2.9.4. Entrenar el algoritmo

Entrenar el algoritmo requiere utilizar las técnicas de aprendizaje automático, en esta etapa alimentamos o introducimos al o los algoritmos de aprendizaje los datos que hemos procesado en las etapas anteriores. Los algoritmos deben ser capaces de extraer información útil de los datos preprocesados, para luego realizar predicciones de forma eficiente.

2.9.5. Evaluar el algoritmo

Evaluar el algoritmo consiste en poner a prueba la información o conocimiento que el algoritmo obtuvo del entrenamiento del paso anterior. Evaluamos que tan preciso es el algoritmo en sus predicciones y si no se obtiene el rendimiento esperado, podemos volver a la etapa anterior y continuar entrenando el algoritmo cambiando algunos parámetros hasta lograr un rendimiento aceptable.

2.9.6. Utilizar el modelo

Utilizar el modelo consiste en poner a prueba el modelo seleccionado, utilizando los nuevos datos, con el fin de etiquetarlos de forma correcta. Se evalúa el rendimiento del modelo y en caso de no obtener el resultado esperado se regresa a revisar todos los pasos anteriores, hasta obtener buenos resultados.

2.10 .Trabajo relacionado

Habiendo introducido conceptos básicos de Machine Learning y la problemática de la manipulación de información en redes sociales, en esta sección se presenta un estudio (Sen, y otros, 2018) realizado en la Universidad Tecnológica de Nanyang (Singapur) para la detección de likes falsos en Instagram.

2.10.1. Detección automática de likes falsos en Instagram

Además de utilizarse como medio de comunicación, las redes sociales también se utilizan para ganar popularidad, aumentar la autoestima social y promover las empresas. Incluso las marcas, los anunciantes y los algoritmos de recomendación de redes sociales confían en las métricas de popularidad de los usuarios y del contenido compartido en estos servicios. Para obtener más beneficios, los usuarios a menudo aumentan artificialmente la popularidad. Tal refuerzo artificial de popularidad puede hacer que las marcas pierdan dinero, los anunciantes no lleguen a la audiencia relevante y los algoritmos de recomendación dar malas sugerencias.

En el estudio, se centran en el compromiso inorgánico recibido por un usuario. Los estudios previos con el objetivo de detectar comportamientos de simulación falsa, suponen que si un usuario le ha dado uno o dos Me gusta falsos, todos sus Me gusta son falsos. Sin embargo, creen que es una suposición errónea ya que un solo usuario puede generar un compromiso orgánico, así como inorgánico. Por ejemplo, a un usuario de Instagram le puede gustar el contenido en el que está realmente interesado, y además, el mismo usuario también puede ser parte de una red de 'me gusta', donde le gusta un contenido aleatorio solo para recibir likes y aumentar su propia popularidad. Por lo tanto, ellos proponen que el verdadero alcance/valor social del usuario se determine cancelando el efecto del compromiso falso que recibe, y que dependa en gran medida solo del compromiso orgánico. Definimos el compromiso de gusto orgánico en Instagram como una acción de un ‘me gusta’ en la que un usuario da a una publicación cuando tiene un interés genuino en el contenido, o en el usuario que publica el contenido (posteador). En dicho estudio, el objetivo es identificar el ingenio de los me gusta determinando la intención del usuario de que le guste una publicación. En particular, definimos *el objetivo* como: *dado un me gusta L, a quien le gusta una publicación específica p de un posteador S. Encontrar las características de L, p y S, para determinar la probabilidad de que le guste realmente una publicación p.*

Para culminar, crean un modelo basado en aprendizaje automático para distinguir automáticamente un me gusta falso de un me gusta orgánico.

Datos

Instancias de Likes falsos (FakeLike_data): existen múltiples fuentes de “Me gusta” falsos, como aplicaciones o servicios web pagos, plataformas de negociación en las que

un usuario participa en dar “Me gusta” a cambio de “Me gusta” y bots que se activan en función de hashtags. Instagram también permite a los usuarios publicar videos y mantiene su recuento de visitas y recuento de me gusta.

En el estudio, se supone que si un video ha recibido “Me gusta”, pero tiene cero visitas, entonces las instancias de likes son falsas, porque se generaron sin ver el contenido correctamente. Así, capturan 16,448 instancias de likes (información sobre el usuario que realiza el “Me gusta”, publicación y el usuario fuente) y lo agregan a FakeLike_data.

Instancias de Likes aleatorias (RandLike_data): es difícil obtener un verdadero conjunto de datos positivo de “Me gusta” genuinos. Por lo tanto, en su lugar, recopilan un conjunto aleatorio mucho más grande de instancias de likes para hacer una comparación con los “Me gusta” falsos y para usar como clase negativa para construir un modelo de aprendizaje automático para identificar los “Me gusta” falsos.

Dado que Instagram no proporciona una forma directa de muestrear usuarios/publicaciones aleatorias, obtienen un conjunto inicial de usuarios de Instagram, y extraen sus seguidores y sus conexiones de seguidores con un algoritmo BFS. Esto les da una muestra de 1 millón de usuarios de Instagram, de los cuales toman un subconjunto más pequeño de usuarios y extraen sus publicaciones y los “Me gusta” en cada una de esas publicaciones. De esta manera, obtienen un conjunto de datos de 134,669 instancias de likes en RandLike_data. Se supone que predominantemente, las instancias de likes en RandLike_data serían genuinas.

Una de las limitaciones del estudio es este conjunto de datos ruidoso, pero con un conjunto de datos negativo limpio, la identificación supervisada basada en el aprendizaje de los likes falsos solo mejoraría.

Los datos recopilados se resumen en la Tabla 3:

	#likes	#publicaciones (p)	#Likers (L)	#Posteadores (S)
FakeLike_data	16.448	9.932	9.301	7.822
RandLike_data	134.669	1.717	47.233	738

Tabla 3: Conjuntos de datos de instancias de "Me gusta" en Instagram

Análisis

Si bien es prácticamente imposible saber por qué a un usuario le puede gustar una publicación, es posible comprender cómo el usuario podría haber encontrado la publicación, lo cual es un requisito previo no trivial para que le guste. En base a esta intuición, en el estudio se enumeran las razones plausibles para que un usuario realmente le guste la publicación de otro usuario.

Introduciendo su definición de *una instancia de un like: dado un posteador S cuya publicación p ha sido likeada por L, definimos una instancia de like como la tupla (L, p, S).* Una instancia de like está diseñada para contener propiedades de publicación para garantizar que un liker sea evaluado en función de las publicaciones individuales que le gustan.

No suponen que si un solo "Me gusta" generado por un "Me gusta" falso, entonces todos sus otros "Me gusta" también son falsos. Luego, en el estudio se define un conjunto de hipótesis que arrojan luz sobre la obtención genuina de un "Me gusta" en Instagram, estas hipótesis se enumeran a continuación

Hipótesis

Efectos de red

Hipótesis 1: Es más probable que a un liker L le guste genuinamente la publicación de S si L es un seguidor de S.

Hipótesis 2: Es más probable que a un liker L le guste genuinamente la publicación de S si L es un seguidor de los seguidores de S.

Para probar la hipótesis 1 e hipótesis 2, estudiaron las conexiones seguidor y seguidor-seguidor de los posteadores en los conjuntos de datos de likes falsos y aleatorios, y descubrieron que a los verdaderos seguidores les gusta más sus publicaciones de seguidores que los seguidores falsos.

Superposición de intereses

Hipótesis 3: un usuario L tendrá más posibilidades de que le guste genuinamente la publicación de S si L y S comparten intereses.

Para capturar la superposición de intereses entre dos usuarios de Instagram, primero definen su *perfil de interés* y el alcance de la superposición como *afinidad*.

Perfil de interés: *Dado un usuario u, se define el perfil de interés de u como un conjunto de temas* $(t_u^1, t_u^2, \dots, t_u^n)$ *donde estos temas se infieren de la biografía y publicaciones de u* $(p_u^1, p_u^2, \dots, p_u^n)$.

Extracción del tema. Infirieren temas de fuentes textuales como la biografía y los pie de video usando Wikification.

Coincidencia de temas. Para hacer coincidir los temas, utilizan las similitudes de word2vec entre dos tuplas de intereses. Se definen los atributos de una publicación como los temas wikificados de la publicación y la similitud temática entre los usuarios de la siguiente manera:

Afinidad. Dado el perfil de interés de los usuarios a y b como $I_a = (t_a^1, \dots, t_a^n)$ y $I_b = (t_b^1, \dots, t_b^m)$, se define a la afinidad como:

$Affinity(T_a, T_b) = \frac{1}{n}\sum_{i=1}^{n} \max_{1 \le j \le m} w2vec(t_a^i, t_b^j)$.

El valor de afinidad está entre [0, 1], un valor de afinidad más alto indica una alta coincidencia tópica.

Así, descubrieron que el 60% de los likers falsos tienen un valor de afinidad de 0.475, en comparación con la afinidad de 0.58 por la misma proporción de conjuntos aleatorios de likers.

Frecuencia de "Me gusta"

Hipótesis 4: A un liker L genuinamente le gustará más de una publicación del posteador S.

Posteador influencer

Hipótesis 5: Un usuario L tendrá más posibilidades de que genuinamente le guste la foto de S, si S es un usuario "influyente" o una celebridad.

Hipótesis 6: Es más probable que un usuario S atraiga "Me gusta" falsos si usa hashtags de granja de enlaces en sus publicaciones.

Se ha demostrado que los hashtags desempeñan un papel importante en Instagram al expandir el alcance de las publicaciones y atraer más "Me gusta".

Hashtags tópicos

Hipótesis 7: Un usuario S con likes genuinos tendrá hashtags tópicos en sus publicaciones.

Para detectar hashtags tópicos siguen dos pasos. Primero filtran todos los hashtags de granjas de enlace, así como los hashtags no tópicos populares. Luego, segmentan esos hashtags y usan Wikifier para ver qué proporción de hashtags pertenece a un tema. Las instancias de “Me gusta” falsos tienden a tener una menor proporción de hashtags tópicos.

Construcción de modelo de clasificación y resultados

Además de las características basadas en las hipótesis descritas anteriormente, también consideran los atributos basados en el usuario, como el volumen de publicaciones generadas, el número promedio de publicaciones por día y la completitud del perfil (presencia de foto de perfil, nombre y biografía).

Así, logran poder detectar instancias de likes falsos con una precisión del 83,5% utilizando un modelo de red neuronal. Probaron con diferentes algoritmos de clasificación y en todos los experimentos realizaron una validación cruzada de 10 veces, utilizando el 80% del conjunto de datos como datos de entrenamiento y el 20% para la validación. Para el modelo basado en perceptrón multicapa, se utilizó 2 capas ocultas con 200 neuronas cada una. Ambas capas con la función de activación sigmoidea, y la capa de salida tiene una caída de 0.2 para evitar el sobreajuste.

Capítulo 3. Marco metodológico

Ya se introdujo el concepto de bot y una introducción al Machine Learning, decidí por desarrollar una aplicación web que sea capaz de detectar si una cuenta pública de Twitter es un bot o no, utilizando técnicas de Machine Learning. El propósito del detector bot es ayudar a evitar a que bots contaminen las redes sociales, en este caso, Twitter. Sirvió como guía el trabajo realizado en (Saharan, 2019).

Como se menciona en (Battur & Yaligar, 2019), los usuarios reciben muchos tweets en los que algunos de ellos son de bots. La detección de bots es necesaria para identificar a los usuarios falsos y proteger a los usuarios genuinos de información errónea y de intenciones maliciosas.

El bot de Twitter es un software que envía tweets automáticamente a los usuarios. Los bots están diseñados para realizar actividades como enviar spam. Las intenciones maliciosas de los bots de Twitter son:

1) Difundir rumores y noticias falsas.
2) Difamar a alguna persona o institución.
3) Se crean comunicaciones falsas para robar credenciales.
4) Dirigir a usuarios genuinos hacia sitios web falsos.
5) Cambiar pensamientos sobre un individuo o grupo, influyendo en la popularidad.

En resumen, un bot es una cuenta automatizada pero, como se había mencionado anteriormente, el foco más importante en el 2020 está en el comportamiento integral de la cuenta y centrarse en si manipula la plataforma.

Según Twitter (Roth & Pickles, 2020), la manipulación de su plataforma ocurre cuando visualizan alguna de estas conductas:

- El uso malicioso de la automatización para perjudicar e interrumpir la conversación pública, como tratar de conseguir que algo se convierta en tendencia.
- La amplificación artificial de las conversaciones en Twitter, incluso mediante la creación de cuentas múltiples o superpuestas
- Generar, solicitar o comprar falso compromiso.
- Tuitear, participar en la conversación o “seguir” de manera masiva o agresiva.

- El uso de hashtags en forma de spam, incluyendo el uso de hashtags no relacionados en un tweet.

El detector bot implementado es un sistema de aprendizaje automático supervisado, basado en modelos y en línea. Se exploran distintos algoritmos de Machine Learning. A continuación se detalla cómo fue realizado este trabajo.

3.1. Política de Twitter relativa a la manipulación de la plataforma

Como se detalla en (Twitter, Inc., 2020), hay muchas formas de manipular la plataforma, las reglas de Twitter están destinadas a contrarrestar una gran variedad de comportamientos prohibidos. De acuerdo con esta política, se prohiben una variedad de comportamientos en las siguientes áreas:

3.1.1. Cuentas e identidad

No puedes engañar a otros en Twitter mediante la administración de cuentas falsas.

Esto incluye el uso de información de cuenta engañosa para llevar a cabo acciones de spam, obstaculización o acoso. Algunos ejemplos de factores que se tienen en cuenta:

- usar fotos de perfil robadas o de archivo, en especial las que representan a otras personas.
- usar biografías de perfil robadas o copiadas.
- usar información de perfil intencionalmente engañosa, como la ubicación del perfil.

No puedes amplificar ni obstaculizar de forma artificial las conversaciones mediante el uso de varias cuentas o coordinando con otros para incumplir las Reglas de Twitter.

Esto incluye:

- cuentas superpuestas: administrar varias cuentas con casos de uso superpuestos, como personas idénticas o similares o contenido sustancialmente similar.
- cuentas que interactúan entre sí: administrar varias cuentas que interactúan entre sí para aumentar o manipular la importancia de determinados Tweets o cuentas.
- coordinación: crear varias cuentas para publicar contenido duplicado o crear interacciones falsas, por ejemplo:

- publicar Tweets o hashtags idénticos o sustancialmente similares desde varias cuentas que administras.
- interactuar (Retweets, Me gusta, menciones, votaciones de encuestas de Twitter) reiteradamente con los mismos Tweets o las mismas cuentas desde varias cuentas que administras.
- coordinar con otras personas o recompensarlas para que generen interacciones o amplificaciones artificiales, incluso si las personas involucradas usan una sola cuenta.
- coordinar con otras personas para participar o promover incumplimientos de las Reglas de Twitter, incluidos incumplimientos de la política relativa a los comportamientos abusivos.

3.1.2. Interacciones y métricas

No puedes aumentar de forma artificial tu propia base de seguidores o la de otros.

Por ejemplo:

- compra/venta de amplificación de métricas de la cuenta o del Tweet: vender o comprar seguidores o interacciones (Retweets, Me gusta, menciones, votaciones de encuestas de Twitter).
- aplicaciones: usar o promocionar servicios o aplicaciones de terceros que ofrecen agregar seguidores o interacciones a los Tweets.
- aumento recíproco: intercambiar o coordinar el intercambio de seguidores o interacciones de Tweets (por ejemplo, participar en acciones conocidas como "trenes de seguidores", "decks" y "Retweet por Retweet").
- transferencia o venta de cuentas: vender, comprar o intercambiar, u ofrecer la venta, compra o intercambio de cuentas de Twitter, nombres de usuario o acceso temporal a cuentas de Twitter.

3.1.3. Uso indebido de las funciones de los productos de Twitter

No puedes hacer un uso indebido de las funciones de los productos de Twitter para obstaculizar la experiencia de otros. Por ejemplo:

3.1.3.1. Tweets y Mensajes Directos

- enviar respuestas, menciones o Mensajes Directos no solicitados de forma masiva, intensa o en grandes volúmenes.

- publicar y eliminar el mismo contenido reiteradamente.
- publicar reiteradamente Tweets idénticos o casi idénticos, o enviar reiteradamente Mensajes Directos idénticos.
- publicar Tweets o enviar Mensajes Directos reiteradamente que solo contengan enlaces compartidos sin comentarios, de modo que esto comprenda la mayor parte de tus acciones de Tweets/Mensajes Directos.

3.1.3.2. Seguimiento

- "seguimiento intermitente": seguir y luego dejar de seguir un gran número de cuentas en un intento de aumentar el propio número de seguidores.
- seguimiento indiscriminado: seguir o dejar de seguir un gran número de cuentas no relacionadas en un período de tiempo reducido, especialmente por medios automáticos.
- duplicar los seguidores de otra cuenta, especialmente por medios automáticos.

3.1.3.3. Interacción

- interactuar de forma intensa o automática con los Tweets para dirigir el tráfico o la atención hacia ciertas cuentas, sitios web, productos, servicios o iniciativas.
- agregar usuarios a Listas o Momentos de forma masiva.

3.1.3.4. Hashtags

- usar un hashtag popular o que es tendencia con la intención de trastornar o manipular una conversación o de dirigir el tráfico o la atención hacia ciertas cuentas, sitios web, productos, servicios o iniciativas.
- twittear con una cantidad excesiva de hashtags no relacionados en un solo Tweet o en varios Tweets.

3.1.3.5. URL

- publicar o establecer enlaces a contenido fraudulento destinado a dañar u obstaculizar el navegador o el equipo de otro usuario (software malicioso), o a vulnerar su privacidad (phishing).
- publicar enlaces engañosos o que den lugar a confusión; por ejemplo, enlaces de afiliación.

3.2. Descripción del modelo propuesto

El modelo propuesto se muestra en la Figura 12, la cual retrata la secuencia de pasos que deben seguir para la clasificación de los usuarios. Este modelo puede ser implementado fácilmente por empresas de redes sociales.

Figura 12: Flujo para predicción en el detector bot.

Este proceso se repite y a medida que avanza el tiempo, el número de datos de entrenamiento aumenta y el clasificador se vuelve cada vez más preciso en sus predicciones.

3.3. Datos

Los algoritmos de aprendizaje automático supervisados requieren un dataset de características con una etiqueta que clasifique cada fila o resultado. Las características son, por lo tanto, la entrada que utilizan los modelos de aprendizaje automático supervisados para predecir un resultado. Estas características pueden ser los atributos que se obtienen a través de APIs que describen una pieza de información sobre una cuenta de una plataforma de red social, como el número de amigos.

Construí un dataset llamado *Humanos*, obtienendo 75.000 tweets provenientes de 2.500 usuarios verificados que cuentan con ciertas características. Para obtener 2.500 usuarios se armó una lista con nombres de usuarios seguidos por la cuenta *@verified*,

esos usuarios están verificados, de cada usuario verificado se obtienen 30 tweets, y de cada tweet se toman distintas atributos como el ID del tweet, cuándo fue creado, cantidad de retweets y favoritos, entre otras características que serán presentadas más adelante.

También se arma un dataset *Bots*, en este caso, se extraen nombres de usuario de cuentas bots que se encuentran en el dataset *botwiki-2019*[1]. Se extraen los nombres de usuario de los bots, de los cuales se toman 670 y se obtienen 40 tweets de cada uno, llegando así a 26.800 tweets de los cuales se extraen las mismas características que para el dataset *Humanos*.

Para el dataset *Total* se toman los dos datasets anteriores y se extraen algunas características y se agregan otras que servirán para la predicción, estas características serán explicadas más adelante. Este dataset servirá como base para tomar algunas características con las que se entrenarán los clasificadores.

En resumen, los conjuntos de datos *Humanos* y *Bots* sirvieron para formar el dataset *Total*, *Total* con algunas características adicionales es el dataset que alimenta el entrenamiento inicial de los clasificadores. En la Tabla 4 se presentan los conjuntos de datos creados.

Dataset	Fuente	Descripción
Humanos	Twitter API (Tweepy): recolectados 2.500 usuarios verificados de los cuales se obtuvieron 75.000 tweets.	Dimensión: (75000, 19) Variables: 19
Bots	Twitter API (Tweepy): Para usuarios bots, se toman 670 usuarios del dataset *botwiki-2019* obtenido en botometer.iuni.iu.edu/bot-repository/datasets.html, de los cuales se obtuvieron 26.800 tweets.	Dimensión: (26800, 19) Variables: 19
Total	Se toman los datasets *Humanos* y *Bots*, se extraen algunas características y se agregan otras para la predicción.	Dimensión: (3188, 17) Características: 16 Target: 1 (bot)

Tabla 4: Conjuntos de datos creados.

3.4. Extracción y análisis de características

Para el entrenamiento inicial de los clasificadores, los conjuntos de datos *Humanos*, *Bots* y *Total* fueron creados con Python utilizando las librerías Pandas (útil para la ciencia de datos y Machine Learning, las razones son muchas y es que ofrece estructuras de datos poderosas, expresivas y flexibles que facilitan la manipulación y

[1] El dataset botwiki-2019 fue obtenido en https://botometer.osome.iu.edu/bot-repository/datasets.html.

análisis de datos) y Tweepy (útil para acceder a la API de Twitter). Luego, en la implementación de la aplicación web para manejar las operaciones referidas a los clasificadores se utilizó Rubix ML que es una librería machine learning de alto nivel para el lenguaje de programación PHP.

Se extraen muchas características de Twitter, a continuación presento una lista de la información extraída de cada tweet junto a datos del usuario autor que lo publicó:

- ID del tweet
- Texto del tweet
- Fecha de creación del tweet
- Cantidad de favoritos del tweet
- Cantidad de RTs del tweet
- Cantidad de menciones en el tweet
- Cantidad de links en el tweet
- Cantidad de hashtags en el tweet
- Fuente del tweet
- ID del usuario
- Nombre de pantalla del usuario
- Nombre del usuario
- Fecha de creación del usuario
- Biografía del usuario
- Longitud de la biografía del usuario
- Ubicación del usuario
- Cantidad de seguidos por el autor del tweet
- Cantidad de seguidores del autor del tweet
- Cantidad de tweets del usuario autor del tweet

Algunas características más se generan y agregan utilizando estas características. Se agrega una columna fechaActual que será de utilidad.

Las características que se añaden por usuario al evaluar las características existentes, están dadas por:

- $Promedio\ de\ favoritos = \frac{\text{Sumatoria de la cantidad de favoritos}}{30}$
- $Promedio\ de\ retweets = \frac{\text{Sumatoria de la cantidad de retweets}}{30}$

- $Promedio\ de\ menciones = \frac{\text{Sumatoria de la cantidad de menciones}}{30}$
- $Promedio\ de\ links = \frac{\text{Sumatoria de la cantidad de links}}{30}$
- $Promedio\ de\ hashtags = \frac{\text{Sumatoria de la cantidad de hashtagss}}{30}$
- $Reputación\ de\ usuario = \frac{\text{Promedio de retweets}}{\text{Cantidad de seguidores}}$
- $Antiguedad =$ fechaActual $-$ Fecha de creación del usuario
- $Tweet\ por\ día = \frac{\text{Cantidad de tweets}}{\text{Antiguedad}}$
- $Tweet\ por\ seguidor = \frac{\text{Cantidad de tweets}}{\text{Cantidad de seguidores}}$
- $Antiguedad\ por\ seguidos = \frac{\text{Antiguedad}}{\text{Cantidad de seguidos}}$
- $name_{binary} = \begin{cases} 1 \\ 0 \end{cases}$ dependiendo si el nombre contiene alguna palabra relacionada a bot.
- $user\_name_{binary} = \begin{cases} 1 \\ 0 \end{cases}$ dependiendo si el nombre de pantalla contiene alguna palabra relacionada a bot.
- $description_{binary} = \begin{cases} 1 \\ 0 \end{cases}$ dependiendo si la biografía contiene alguna palabra relacionada a bot.

En los promedios el cociente es por 30 porque se recopilaron 30 tweets por usuario, salvo en el caso del dataset *Bots* que la división es por 40 dado que se recopilan esa cantidad de tweets por usuario bot.

3.5. Selección de características para la predicción

Las características que se tienen en cuenta para entrenar los clasificadores son las siguientes:

- Reputación de usuario.
- Promedios de hashtags, links, retweets, favoritos y menciones en los tweets.
- Antigüedad de la cuenta.
- Cantidad de usuarios seguidos.
- Cantidad de usuarios seguidores.
- Cantidad de tweets.
- Cantidad de tweets por día.
- Cantidad de tweets por seguidor.

- Cantidad de días de la cuenta por usuarios seguidos.
- Valores binarios que indican si el nombre de usuario, el nombre o la descripción contienen alguna palabra relacionada a bot.

3.5.1. Clase objetivo

bot: toma el valor “bot” para el perfil de usuario de una cuenta bot, y el valor “noBot” para el perfil de usuario de una cuenta manejada por un humano.

3.6. Métricas de evaluación

A continuación, se presenta la terminología necesaria para comprender las métricas que se utilizaron, estos términos fueron tomados de (Google Developers, 2020):

- Ejemplo

  Fila de un conjunto de datos. Un ejemplo contiene una o más características o atributos y, posiblemente, una etiqueta.

- Clase positiva:

  En la clasificación binaria, las dos clases posibles se etiquetan como positiva y negativa. El resultado positivo es aquello que estamos probando. En el detector bot, la clase positiva es “bot”.

- Clase negativa

  En la clasificación binaria, una clase se expresa como positiva y la otra como negativa. La clase positiva es lo que estamos buscando y la clase negativa es la otra posibilidad. En el detector bot, la clase positiva es “noBot”.

- Verdadero positivo (VP)

  Ejemplo en el que el modelo predijo correctamente la clase positiva. Por ejemplo, el modelo infirió que un mensaje de correo electrónico era spam y realmente lo era.

- Verdadero negativo (VN)

  Un ejemplo en el que el modelo predijo correctamente la clase negativa. Por ejemplo, el modelo infirió que un mensaje de correo electrónico no era spam y realmente no lo era.

- Falso positivo (FP)

  Ejemplo en el que el modelo predijo de manera incorrecta la clase positiva. Por ejemplo, el modelo infirió que un mensaje de correo electrónico en particular

era spam (la clase positiva), pero ese mensaje de correo electrónico en realidad no era spam.

- Falso negativo (FN)

  Ejemplo en el que el modelo predijo de manera incorrecta la clase negativa. Por ejemplo, el modelo infirió que un mensaje de correo electrónico en particular no era spam (la clase negativa), pero ese mensaje de correo electrónico en realidad era spam.

Para evaluar el rendimiento en la clasificación, usé las métricas de exactitud y precisión.

- Exactitud

  Fracción de predicciones que se realizaron correctamente en un modelo de clasificación.

  En la clasificación de clases múltiples, la exactitud se define de la siguiente manera:

$$Exactitud = \frac{Predicciones\ correctas}{Número\ total\ de\ ejemplos}$$

  En la clasificación binaria, la exactitud se calcula con la siguiente fórmula:

$$Exactitud = \frac{VP + VN}{VP + VN + FP + FN}$$

- Precisión

  La precisión identifica la frecuencia con la que un modelo predijo correctamente la clase positiva, mide la calidad de la predicción. Para calcular la precisión se hace uso de la siguiente fórmula:

$$Precisión = \frac{VP}{VP + FP}$$

3.7. Clasificación

Una tarea típica de aprendizaje supervisado es la *clasificación*.

La clasificación es una técnica de categorizar un objeto en una clase particular basada en el conjunto de datos de entrenamiento que se utilizó para entrenar al clasificador, con la particularidad que las muestras del conjunto de datos están etiquetadas, para tal fin, alimentamos los clasificadores con dataset *Total*. El clasificador es un algoritmo utilizado para la clasificación.

El detector bot implementa tres clasificadores: K vecinos más cercanos, Naïve Bayes Gaussiano y Bosque aleatorio. Los detalles sobre el funcionamiento de los clasificadores se dan a continuación.

3.7.1. K vecinos más cercanos

K vecinos más cercanos (k-nearest neighbors, en inglés) es un algoritmo basado en instancia de tipo supervisado de Machine Learning.

En la clasificación por K vecinos más cercanos, la salida es la pertenencia a una clase. Un objeto es clasificado como perteneciente a una clase si la mayoría de sus k vecinos pertenecen a esa clase.

Funcionamiento:

1. Calcular la distancia entre el objeto a clasificar y el resto de los ítems del dataset de entrenamiento.
2. Seleccionar los “k” elementos más cercanos (con menor distancia, según alguna métrica de distancia como la distancia euclidiana).
3. Realizar una “votación de mayoría” entre los k puntos: los puntos de una clase que “dominen” decidirán la clasificación final.

   Para evitar empates, en la clasificación binaria se prefiere que el número de vecinos k seleccionado sea un número impar.

Ejemplo:

En el siguiente ejemplo que se muestra en (Hernandez, 2020), se busca clasificar el punto (4, 3) para saber si es de color azul o rojo, con k=7 vecinos y como métrica la distancia euclidiana.

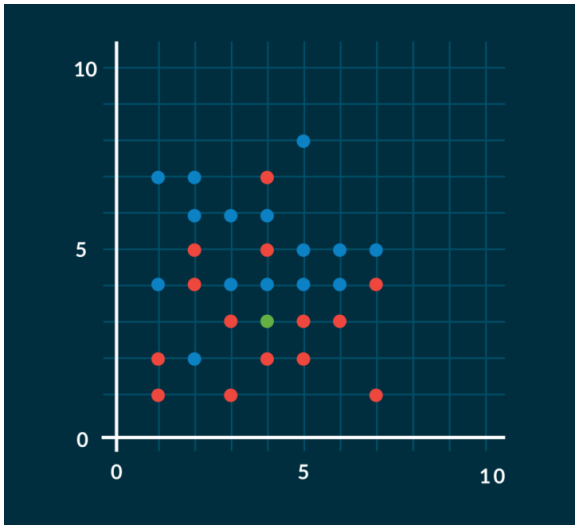

Figura 13: Ejemplo para K-nn.

En las Tablas 5 y 6 se muestran las distancias euclidiana de cada punto al punto (4, 3):

Rojo	Distancia al (4,3)
(1,1)	3.605551275
(1,2)	3.16227766
(2,4)	2.236067977
(2,5)	2.828427125
(3,1)	2.236067977
(3,3)	1
(4,2)	1
(4,5)	2
(4,7)	4
(5,2)	1.414213562
(5,3)	1
(6,3)	2
(7,1)	3.605551275
(7,4)	3.16227766

Tabla 5: Detalle puntos rojos para ejemplo K-nn.

Azul	Distancia al (4,3)
(1,4)	3.16227766
(1,7)	5
(2,2)	2.236067977
(2,7)	4.472135955
(2,8)	5.385164807
(3,4)	1.414213562
(3,6)	3.16227766
(4,4)	1
(4,6)	3
(5,4)	1.414213562
(5,5)	2.236067977
(5,8)	5.099019514
(6,4)	2.236067977
(6,5)	2.828427125
(7,5)	3.605551275

Tabla 6: Detalle puntos azules para ejemplo K-nn.

Se puede ver que los puntos (3, 3), (3, 4), (4, 2), (4, 4), (5, 2), (5, 3) y (5, 4) son los puntos con menor distancia al punto (4, 3), de los cuales 4 son rojos y 3 son azules. Por lo tanto, se clasifica el punto (4, 3) como rojo.

3.7.1.1. Resultado de K vecinos más cercanos

Para el clasificador de K vecinos más cercanos se optó por k=3 vecinos y como medida de distancia se tomó la distancia Manhattan, al entrenarse este clasificador se obtuvieron las siguientes métricas:

- Exactitud: 0.95454545454545
- Precisión: 0.94034416292112

3.7.2. Naïve Bayes Gaussiano

Naïve Bayes es un método de clasificación estadística basada en el Teorema de Bayes. Es uno de los algoritmos de aprendizaje supervisado más simples. Los clasificadores Naïve Bayes tienen alta precisión y velocidad en grandes conjuntos de datos.

El clasificador Naïve Bayes asume como hipótesis la independencia condicional de clase: el efecto de una característica particular en una clase es independiente de otras

características. Por ejemplo, un solicitante de préstamo es deseable o no dependiendo de sus ingresos, historial de préstamos y transacciones anteriores, edad y ubicación. Incluso si estas características son interdependientes, estas características se consideran de forma independiente. Esta suposición simplifica el cálculo y por eso se considera ingenuo (naïve).

El Teorema de Bayes está expresado por la siguiente ecuación:

$$P(H|D) = \frac{P(D|H)P(H)}{P(D)}$$

$P(H)$ es el conocimiento inicial que tenemos sobre que la hipótesis H sea la correcta, se le suele denominar la probabilidad a priori de H.

$P(D)$ se define de forma similar, pero esta vez sobre los datos D.

$P(D|H)$ denota la probabilidad de observar los datos D dado que tenemos la hipótesis H, se le suele denominar verosimilitud.

$P(H|D)$ es la probabilidad a posteriori que la hipótesis H tiene, dados los datos observados D.

La fórmula nos indica la probabilidad de que una hipótesis H sea verdadera si algún evento D ha sucedido. Esto es importante dado que, normalmente obtenemos la probabilidad de los efectos dadas las causas, pero el Teorema de Bayes nos indica la probabilidad de las causas dados los efectos.

Ejemplo:

Clima	Temperatura	Humedad	Ventoso	Tenis (Clase objetivo)
Lluvioso	Calor	Alta	Falso	No
Lluvioso	Calor	Alta	Verdadero	No
Nublado	Calor	Alta	Falso	Sí
Soleado	Templado	Alta	Falso	Sí
Soleado	Frío	Normal	Falso	Sí
Soleado	Frío	Normal	Verdadero	No
Nublado	Frío	Normal	Verdadero	Sí
Lluvioso	Templado	Alta	Falso	No
Lluvioso	Frío	Normal	Falso	Sí
Soleado	Templado	Normal	Falso	Sí
Lluvioso	Templado	Normal	Verdadero	Sí
Nublado	Templado	Alta	Verdadero	Sí
Nublado	Calor	Normal	Falso	Sí
Soleado	Templado	Alta	Verdadero	No

Tabla 7: Dataset para predecir si se juega o no un partido de tenis.

Considerando el conjunto de datos de la Tabla 7, suponemos que no existe ningún par con características dependientes. Por ejemplo, la temperatura categorizada como "Calor" no depende de la humedad o el clima como "Lluvioso" no tiene ningún efecto sobre los vientos.

Y a cada característica se le da el mismo peso, ninguna de las características es irrelevante y se supone que contribuye por igual al resultado.

Sea el vector dependiente (clase objetivo) y, y el conjunto de características X donde contiene características como $X = (x_1, x_2, \dots, x_n)$, se puede expresar el Teorema de Bayes como:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

Suponiendo el caso $X = (Soleado,\ Calor,\ Normal,\ Falso)$ y se busca predecir y.

En caso de probabilidad que se juegue tenis:

$$P(Sí|hoy) = \frac{P(Clima\ Soleado|Sí)P(Temperatura\ Calor|Sí)P(Humedad\ Normal|Sí)P(Ventoso\ Falso|Sí)P(Sí)}{P(hoy)}$$

Y en caso de probabilidad de no jugar al tenis, se puede escribir la ecuación como:

$$P(No|hoy) = \frac{P(Clima\ Soleado|No)P(Temperatura\ Calor|No)P(Humedad\ Normal|No)P(Ventoso\ Falso|No)P(No)}{P(hoy)}$$

Después de un cálculo matemático, podemos calcular la probabilidad de cada característica con respecto a las condiciones de la variable dependiente "Sí" y "No".

$P(Sí|hoy) = 0.0141$

$P(No|hoy) = 0.0068$

Entonces, para el vector $X = (Soleado,\ Calor,\ Normal,\ Falso)$, como $P(Sí|hoy) > P(No|hoy)$ se obtiene la predicción para y como $y = Sí$.

Como se explica en (Kumar, 2020), en este ejemplo, los datos proporcionados fueron de tipo discreto. Si los datos son variables continuas, se puede aplicar Naïve Bayes Gaussiano. Los diferentes clasificadores ingenuos de Bayes se diferencian principalmente por las suposiciones que hacen con respecto a la distribución de $P(x_i|y)$.

En el algoritmo Naïve Bayes Gaussiano, se supone que los valores continuos asociados con cada característica se distribuyen de acuerdo con una distribución Gaussiana. Una distribución Gaussiana también se denomina distribución normal. Cuando se traza, da

una curva en forma de campana que es simétrica con respecto a la media de los valores de las características, como se muestra en la Figura 14.

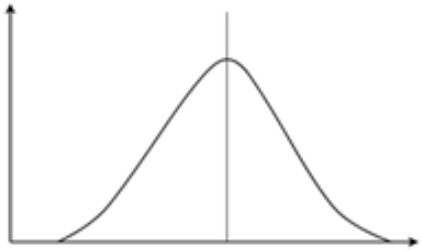

Figura 14: Distribución Gaussiana o Normal.

La probabilidad de las características se considera Gaussiana, entonces, la probabilidad condicional viene dada por la siguiente fórmula:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

3.7.2.1. Resultado de Naïve Bayes Gaussiano

Al entrenarse el clasificador Naïve Bayes Gaussiano se obtuvieron las siguientes métricas:

- Exactitud: 0.97648902821317
- Precisión: 0.96945240485273

3.7.3. Bosque aleatorio

Los bosques aleatorios es un algoritmo de aprendizaje supervisado. Se puede utilizar tanto para clasificación como para regresión. También es el algoritmo más flexible y fácil de usar. Un bosque está compuesto por árboles. Se dice que cuantos más árboles tiene, más robusto es un bosque. Los bosques aleatorios crean árboles de decisión en ejemplos seleccionados al azar, obtienen una predicción de cada árbol y seleccionan la mejor solución mediante votación.

Funcionamiento:

1. Selecciona muestras aleatorias de un dataset determinado.
2. Construye un árbol de decisión para cada muestra y obtiene un resultado de predicción de cada árbol de decisión.
3. Realiza una votación por cada resultado previsto.
4. Selecciona el resultado de la predicción con más votos como predicción final.

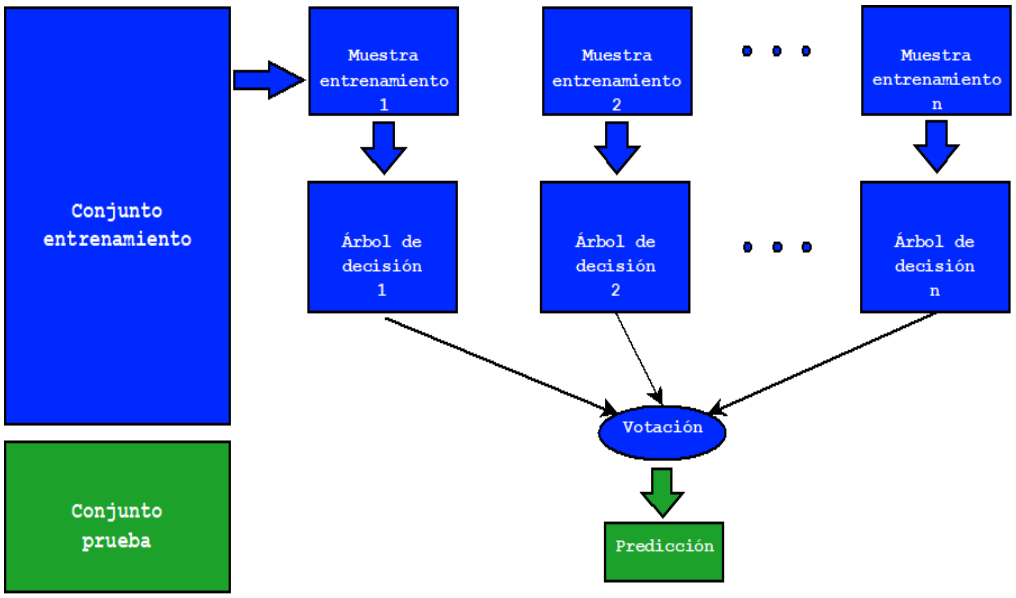

Figura 15: Predicción en bosques aleatorios.

Los Bosques Aleatorios tienen una variedad de aplicaciones, por ejemplo, motores de recomendación y clasificación de imágenes, también se pueden utilizar para predecir enfermedades.

3.7.3.1. Resultado de Bosque Aleatorio

Para el clasificador de Bosque Aleatorio se decidió que las muestras aleatorias se dividan en 100 árboles de decisión, al entrenarse este clasificador se obtuvieron las siguientes métricas:

- Exactitud: 0.83385579937304
- Precisión: 0.82605316261463

3.8. Funcionalidades adicionales

Como funcionalidades adicionales, está la posibilidad de visualizar los tweets que sirvieron para extraer las características para la clasificación de una cuenta en bot o humano.

También, se pueden obtener los puntajes de Botometer (mencionado en la sección 2.5.3.). Botometer toma un nombre de usuario de Twitter y da ciertas puntuaciones, el rango de esas puntaciones va desde 0 hasta 5, mientras esa puntuación es más cercana a 0 esa cuenta de Twitter tiene una activad similar a un humano, y si esa puntuación es cercana a 5 dicha cuenta de Twitter tiene una actividad más similar a un bot. A continuación, se detallan los puntajes mostrados por el puntaje Botometer:

- General: Basado en una comparación de varios modelos entrenados en diferentes tipos de bots y en cuentas humanas, corresponde al modelo con mayor confianza.
- Astroturf: Bots políticos etiquetados manualmente y cuentas involucradas en trenes de seguimiento que eliminan contenido sistemáticamente.
- Fake follower: Bots comprados para aumentar el número de seguidores.
- Financial: Bots que publican utilizando cashtags.
- Other: Varios otros bots obtenidos a partir de anotaciones manuales, feedback de los usuarios, etc.
- Self declared: Bots de botwiki.org.
- Spammer: Cuentas etiquetadas como spambots de varios datasets.

3.9. Detector bot

La aplicación web implementada se hizo pública utilizando la plataforma Heroku, optándose por el plan gratuito. Heroku provee el servicio de base de datos PostgreSQL gratuito, se hace uso del mismo.

El link a la aplicación es el siguiente:

https://tesis-uns.herokuapp.com/

Capítulo 4. Conclusiones y trabajo futuro

El futuro de los ecosistemas de redes sociales podría ya apuntar en la dirección donde la interacción máquina-máquina sea la norma, y los humanos navegan en un mundo poblado principalmente por bots. Por eso, es necesario que los bots y los humanos puedan reconocerse entre sí, para evitar situaciones extrañas, o incluso peligrosas basadas en suposiciones falsas.

Los comportamientos de los bots ya son bastante sofisticados: pueden construir redes sociales realistas y producir contenido creíble con patrones similares a los humanos. La necesidad de instancias de entrenamiento es una limitación del aprendizaje supervisado en tal escenario. Nadie sabe exactamente cuántos bots pueblan las redes sociales, o qué parte del contenido puede atribuirse a los bots.

Los servicios de fraude de las redes sociales son fácilmente accesibles a través de búsquedas online. De este modo, los clientes pueden encontrar fácilmente el servicio que están buscando. La gran variación de precio en la venta muestra que el mismo servicio puede tener un precio muy bajo o muy caro, lo que lo hace accesible a cualquier tipo de consumidor.

Estos contribuidores contribuyen a la desinformación en las redes sociales.

La manipulación de datos de redes sociales crea desinformación, que a su vez puede ser perjudicial para el ecosistema de redes sociales. Es necesario tomar medidas para disminuir este fenómeno desde todos los ángulos: eliminar sitios web fraudulentos, desarrollar técnicas de detección más sofisticadas, establecer fuertes barreras de registro y contribuir a la conciencia de los potenciales compradores potenciales de fraude en redes sociales, tratando de brindarles las herramientas que necesitan para ganar credibilidad y popularidad.

Con el aumento de la manipulación de las redes sociales, la fiabilidad de la información proporcionada puede ser cuestionada. Como las redes sociales son los principales medios para muchas actividades sociales, económicas y políticas y comprometen a una gran parte de la población mundial, parte de la responsabilidad de los desarrolladores de redes sociales radica en evitar la manipulación de datos en sus plataformas.

Las personas seguirán siendo susceptibles a los riesgos de manipulación y privacidad a menos que se tomen acciones coordinadas entre los desarrolladores de tecnologías de medios y los usuarios. La premisa de tales contramedidas no es prohibir que las

personas compartan actualizaciones de estado, fotos y redes, sino apoyarlas en sus decisiones de privacidad individuales. Esto no solo aumentaría sus niveles de conocimiento del riesgo, sino que también les permitiría revelar información privada bajo su propia responsabilidad.

La detección de manipulación de información ha sido durante mucho tiempo un problema, siempre tuvo el poder de confundir a las personas. Además, este problema ha ganado complejidad durante la actual revolución tecnológica. La automatización y la ampliación de la producción de medios a través de las redes sociales son, al mismo tiempo, un gran desafío y una oportunidad.

Un gran desafío, debido a la gran cantidad de partes interesadas y emisores activos de contenido. Incluso para una mente bien formada, ya no es posible comprender las complejas relaciones entre emisores y lectores en las redes sociales.

Una oportunidad, porque las máquinas y los informáticos proporcionan herramientas y métodos de alta eficiencia para manejar los datos y convertirlos en un conocimiento constante y actualizado. Sin embargo, necesitan trabajar con los usuarios finales, para finalmente permitir una verdadera detección de manipulación de la información, capitalizando los conocimientos expertos y automáticos.

Con respecto al detector bot implementado, se investigaron distintos algoritmos de Machine Learning y elegí tres de ellos: *K vecinos más cercanos*, *Naïve Bayes Gaussiano* y *Bosque aleatorio,* La detección de bots no es una tarea fácil, se utilizan muchas características para tal fin, por lo que el resultado obtenido puede no ser certero. Los modelos entrenados con esos algoritmos dieron buenos resultados, siendo el modelo de Bosques Aleatorios el modelo menos confiable con una exactitud y precisión del 83%, el modelo basado en el algoritmo Naïve Bayes Gaussiano resultó ser el más confiable con una exactitud del 98% y una precisión del 97%. Se probaron distintas cuentas de Twitter, tomando como casos de prueba cuentas de Twitter que caen en alguno de estos casos: cuenta con nombre de usuario inválido, cuenta con nombre de usuario inexistente, cuenta sin tweets, cuenta con los tweets protegidos o cuenta válida con tweets públicos. Para los primeros 4 casos mencionados, se muestran adecuadamente sus respectivos mensajes de error. Para el caso de la cuenta válida, se muestra el resultado el cual es almacenado en la base de datos junto a los datos que se

extraen con la base del conjunto de datos inicial, así, los modelos se van entrenando incrementalmente con cada predicción.

La aplicación web cuenta con limitaciones para el tamaño de la base de datos PostgreSQL por el plan gratuito de Heroku, se tiene un límite para 10.000 filas y una capacidad de almacenamiento de 1 GB.

La aplicación web implementada parte de información de la cuenta pública que se puede ver en Twitter e identifica las características que la convierten en un bot; por ejemplo, el nombre de la cuenta, el número de tweets, la ubicación en la biografía, los hashtags utilizados, etc.

Este enfoque es limitado. Por ejemplo, una cuenta con un nombre extraño puede pertenecer a alguien a quien se le recomendó automáticamente ese nombre de usuario debido a que su nombre real ya estaba tomado cuando se registró. Una cuenta sin foto o ubicación podría ser alguien que quiere proteger su privacidad o cuyo uso de Twitter pueda exponerlos a un riesgo, como un activista. A muchos en Twitter tampoco les gusta añadir mucha biografía o ubicación a su perfil. Incluso si todos esos detalles públicos se colocan en un modelo de machine learning para tratar de predecir de forma probable si una cuenta es un bot, cuando se apoyan en el análisis humano de la información de la cuenta pública, ese proceso contiene prejuicios desde un principio. Otro ejemplo de la vida cotidiana, alguien preocupado por el medio ambiente que twittea 100 veces al día con un determinado hashtag no significa que sea un “bot político”, sino que es un ciudadano activo que organiza una campaña online para impulsar un cambio en su comunidad.

Un trabajo futuro, más allá de las características públicas, puede enfocarse en patrones de interacción y en el contenido de los mensajes, siempre teniendo en cuenta si se manipula la plataforma o no.

Cabe destacar que marcar cuentas falsas no evita el fraude o la manipulación y solo sirve para limpiar la red social a futuro.

La lucha contra los bots terminará solo cuando la efectividad de la detección temprana aumente suficientemente el costo del engaño.

Bibliografía

Aïmeur, E., Díaz Ferreyra, N., & Hage, H. (Noviembre de 2019). Manipulation and Malicious Personalization: Exploring the Self-Disclosure Biases Exploited by Deceptive Attackers onSocial Media. *Frontiers in Artificial Intelligence, 2*(26), 1-12. doi:10.3389/frai.2019.00026

Battur, R., & Yaligar, N. (Julio de 2019). Twitter Bot Detection using Machine Learning Algorithms. *International Journal of Science and Research (IJSR), 8*(7), 304-307. Obtenido de https://www.ijsr.net/search_index_results_paperid.php?id=ART20199245

Chamorro Alvarado, V. L. (2018). *Clasificación de Tweets mediante modelos de aprendizaje supervisado.* (Trabajo Fin Máster en Ingeniería Informática, Universidad Complutense de Madrid, Madrid, España). Obtenido de https://eprints.ucm.es/49774/1/TFM%20Veronica%20Chamorro%20Alvarado.pdf.

Davis, C. A., Varol, O., Ferrara, E., Flammini, A., & Menczer, F. (2016). BotOrNot: A System to Evaluate Social Bots. *25th International World Wide Web Conference Companion*, (pp. 273-274). doi:http://dx.doi.org/10.1145/2872518.2889302

Ferrara, E., Varol, O., Davis, C., Menczer, F., & Flammini, A. (Julio de 2016). The Rise of Social Bots. *Communications of the ACM*, 96-104. doi:10.1145/2818717

Gadek, G., Justine, V., & Everwyn, J. (Noviembre de 2019). Manipulation and fake news detection on social media: a two domain survey, combining social network analysis and knowledge bases exploitation. *C&ESAR Conference*, 1-14. Obtenido de https://www.cesar-conference.org/wp-content/uploads/2019/11/20191120_J2_250_G-GADEK_Manipulation_social_media.pdf

Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2da ed.). O'Reilly Media, Inc.

Google Developers. (11 de Agosto de 2020). Machine Learning Glossary. Obtenido de https://developers.google.com/machine-learning/glossary

Hernandez, R. (2020, Junio 15). *Método de los K vecinos más cercanos*. Retrieved from Medium: https://medium.com/soldai/m%C3%A9todo-de-los-k-vecinos-m%C3%A1s-cercanos-f8231c28f7c7

Kumar, N. (2020, Mayo 15). *Naive Bayes Classifiers*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/naive-bayes-classifiers/

Paquet-Clouston, M., Bilodeau, O., & Décary-Hétu, D. (2017). Can We Trust Social Media Data? Social Network Manipulation by an IoT Botnet. *#SMSociety17: Proceedings of the 8th International Conference on Social Media & Society* (págs. 1-9). Toronto: Association for Computing Machinery. doi:10.1145/3097286.3097301

Rodríguez-Andrés, R. (2018). Trump 2016: ¿presidente gracias a las redes sociales? *Palabra Clave, 21*(3), 831-859. doi:10.5294/pacla.2018.21.3.8

Roth, Y., & Pickles, N. (18 de Mayo de 2020). *Bot or not? The facts about platform manipulation on Twitter*. Obtenido de [Entrada de blog]: https://blog.twitter.com/en_us/topics/company/2020/bot-or-not.html

Saharan, R. (2019). *Spammer detection from twitter*. Obtenido de [Repositorio GitHub]: https://github.com/radheysm/Spammer-detection-from-twitter

Sen, I., Aggarwal, A., Mian, S., Singh, S., Kumaraguru, P., & Datta, A. (2018). Worth its Weight in Likes: Towards Detecting Fake Likes on Instagram. *WebSci '18: 10th ACM Conference on Web Science* (pp. 1-5). Amsterdam: ACM. doi:10.1145/3201064.3201105

Twitter, Inc. (Septiembre de 2020). Política relativa al spam y la manipulación de la plataforma. Obtenido de https://help.twitter.com/es/rules-and-policies/platform-manipulation