Computación blanda (2020) 24: 9049–9069 https://
doi.org/10.1007/s00500-019-04436-y

METODOLOGÍAS Y APLICACIÓN

Automatización del sistema de detección de noticias falsas utilizando un modelo de votación de varios niveles

Sawinder Kaur[1] · Parteek Kumar[2] · Ponnurangam Kumaraguru[3]

Publicado en línea: 2 de noviembre de 2019

Abstracto
Los problemas de las noticias falsas en línea han alcanzado una importancia cada vez mayor en la difusión de las noticias que dan forma a las noticias en línea. La información engañosa o poco confiable en forma de videos, publicaciones, artículos y URL se difunde ampliamente a través de plataformas de redes sociales populares como Facebook y Twitter. Como resultado, los editores y periodistas necesitan nuevas herramientas que les ayuden a acelerar el proceso de verificación del contenido que se ha originado en las redes sociales. Motivado por la necesidad de detección automatizada de noticias falsas, el objetivo es averiguar qué modelo de clasificación identifica características falsas con precisión utilizando tres técnicas de extracción de características, frecuencia de término-frecuencia de documento inverso (TF-IDF), vectorizador de conteo (CV) y hash. -Vectorizador (HV). Además, en este documento, se propone un modelo novedoso de conjunto de votaciones multinivel. El sistema propuesto se ha probado en tres conjuntos de datos utilizando doce clasificadores. Estos clasificadores de ML se combinan en función de su proporción de predicción falsa. Se ha observado que el clasificador pasivo agresivo, de regresión logística y de vector de soporte lineal (LinearSVC) se desempeña mejor individualmente utilizando enfoques de extracción de características TF-IDF, CV y HV, respectivamente, en función de sus métricas de rendimiento, mientras que el modelo propuesto supera al pasivo agresivo modelo por 0,8%, modelo de regresión logística por 1.3%, modelo LinearSVC en 0.4% usando TF-IDF, CV y HV, respectivamente. El sistema propuesto también se puede utilizar para predecir el contenido falso (forma textual) de los sitios web de redes sociales en línea.

Palabras clave Artículos de noticias falsos · Vectorizador de conteo · TF-IDF · Vectorizador de hash · Clasificadores · Contenido textual · Máquina modelos de aprendizaje

1. Introducción

Un creciente interés relacionado con la detección de noticias falsas ha atraído a muchos investigadores a medida que circula información falsa a través de plataformas de redes sociales en línea como Facebook y Twitter. El contenido falso se está extendiendo a un ritmo más rápido para ganar popularidad en las redes sociales, para distraer a las personas de los problemas críticos actuales. La mayoría de las personas creen que la información que reciben de varios sitios de redes sociales es confiable y verdadera, es decir, las personas están intrínsecamente sesgadas por la verdad. Además, las personas confían fácilmente y quieren creer en lo que realmente interpretan en sus mentes, es decir, con sesgos de confirmación. En general, se ha analizado que los humanos son incapaces de reconocer el engaño de forma eficaz. Debido a esto, se puede ver un impacto grave y negativo de los artículos falsos en la sociedad y las personas, lo que conduce a un desequilibrio del ecosistema de noticias. Se observó que durante la elección del presidente de Estados Unidos (Conroy et al.2015), la mayoría de los artículos ampliamente difundidos en las redes sociales eran falsos. Recientemente, un video falso relacionado con Kerela luchando contra las inundaciones se volvió viral en la plataforma de redes sociales (Facebook) afirmando que el Ministro Principal del estado está obligando al ejército indio a dejar de realizar las operaciones de rescate en las regiones inundadas de Kerela. Además, durante las elecciones nacionales de India (2019), se crearon varios grupos de WhatsApp (> 900.000) para difundir información falsa sobre el partido gobernante de India (http://bit.ly/2miuv9j). La mayoría de los artículos falsos

Comunicado por V. Loia.

B Sawinder Kaur
sawinderkaurvohra@gmail.com

Parteek Kumar
parteek.bhatia@thapar.edu

Ponnurangam Kumaraguru
pk@iiitd.ac.in

1 Doctoral Research Lab-II, Departamento de Ingeniería y Ciencias de la Computación, TIET, Patiala, India

2 Departamento de Ingeniería y Ciencias de la Computación, TIET, Patiala, India

3 Departamento de Ingeniería y Ciencias de la Computación, IIIT, Delhi, India

se crean para confundir a la gente y desencadenar su desconfianza. Tales problemas llevaron a los investigadores a buscar algunas formas automatizadas de acceder a los valores de verdad básicos del texto falso sobre la base del contenido textual publicado en artículos en plataformas sociales.

Las redes sociales permiten mantener y desarrollar relaciones con los demás. Ayuda a los usuarios a presentarse creando sus perfiles, compartiendo información a través de fotografías, imágenes y texto para vincular con otros miembros (Canini et al. 2011). Algunas de las redes sociales más populares (Ahmed et al. 2017) los sitios son Facebook, Twitter (Gupta et al. 2013a; Wang 2010; Benevenuto y col.2010), Instagram (Sen et al. 2018), WhatsApp (Garimella y Tyson 2018; Caetano y col.2018), LinkedIn, WeChat, Snapchat y Foursquare (Pontes et al. 2012b). Con la popularidad de los sitios de redes sociales (Dewan et al. 2013), el nivel de uso para compartir contenido en las redes sociales en línea ha aumentado. Varias son las razones del cambio de comportamiento de este tipo de consumos. El contenido compartido en las plataformas de redes sociales requiere menos tiempo y costo que en los periódicos o los medios de comunicación tradicionales. Es más fácil compartir contenido en forma de videos, blogs, publicaciones con amigos o usuarios. Esto da la facilidad de crecimiento a los autores y editores para publicar sus contenidos como artículos en entornos colaborativos. Hay un 13% del aumento global en el uso de redes sociales desde 2017 (Garimella y Tyson2018). Distribución y creación de contenido de noticias en forma de publicaciones (Weimer et al.2007), blogs, artículos, imágenes, videos, etc., se han difundido a través de sitios web de redes sociales (Dewan et al. 2013). Este aumento en los medios sociales también da lugar a la difusión de artículos falsos (Wei y Wan2017) a través de Internet.

1.1 Tipos de noticias falsas

Según la literatura, existen cinco tipos de noticias falsas. El primer tipo se puede ver en forma de*desinformación deliberada,* que es información engañosa que se difunde de forma calculada para engañar a los usuarios objetivo. Otras formas de noticias falsas pueden ser *clickbait*Chen y col. 2015; Shu y col.2017) que captan la atención del lector con el propósito de hacer que haga clic en las noticias falsas que se ven en Internet. Los usuarios que configuran sitios falsos generan enormes ingresos y al hacer clic en dichos sitios web se generan anuncios bombardeados. Artículos (Wei y Wan 2017) de fuentes satíricas como 'The Onion' a menudo repiten y comparten las noticias como si las historias impresas fueran verdaderas. *Parodia o satírico (*Rubin y col. 2016) los artículos utilizan la obscenidad, el absurdo y la exageración para comentar la actualidad y para incomodar a los lectores. *Titulares falsos* se exageran intencionadamente para llamar la atención del lector. En tales titulares, el título de los artículos puede no coincidir con el contexto de las historias, el titular puede leerse de una forma y declarar algo diferente como un hecho. Este tipo de noticias falsas es falso en el peor de los casos y engañoso en el mejor de los casos.*Engaños* es otro tipo de desinformación que engaña al lector deliberadamente al causar daños y pérdidas materiales a los usuarios.

1.2 Contribución

Los investigadores han analizado que un sistema automatizado a veces identifica los artículos de noticias falsos mejor que los humanos. Los sistemas automatizados pueden ser una herramienta importante para identificar historias, artículos, blogs y clics falsos que manipulan la opinión pública en los sitios de redes sociales (Dewan y Kumaraguru 2017; Jain y Kumaraguru2016). Teniendo en cuenta la necesidad del desarrollo de dicho sistema de detección de noticias falsas, en este documento hemos identificado artículos de noticias como falsos o reales mediante el uso de clasificadores de aprendizaje automático supervisados como Naïve Bayes (NB), Decision Tree (DT), Support Vector Machine (SVM).), Modelos lineales, redes neuronales (NN) y modelos de conjunto. Para obtener resultados efectivos, se han recopilado tres corpus diferentes (News Trends, Kaggle y Reuters) con características similares. Las técnicas de extracción de características, Frecuencia de términos – Frecuencia de documento inversa (TF – IDF), Count-Vectorizer (CV), Hashing-Vectorizer (HV), se utilizan para extraer vectores de características del contenido textual de los artículos. La efectividad de un conjunto de clasificadores se observa cuando predicen las etiquetas de las muestras de prueba después de aprender los datos de entrenamiento usando estas técnicas de extracción de características.

Varios modelos de ML supervisados se utilizan ampliamente para la categorización de datos textuales como falsos o reales, pero dichos modelos no pudieron obtener los resultados de un clasificador ideal. Por lo tanto, en este documento se ha propuesto un modelo de votación multinivel para construir un clasificador ideal con una mejora significativa que los modelos previamente existentes. Nuestra contribución a este documento es la siguiente:

- Se ha realizado un análisis estadístico de los conjuntos de datos recopilados (News Trends, Kaggle y Reuters) con instancias negativas y positivas.
- Se evalúan doce modelos ML utilizando técnicas de extracción de características TF-IDF, CV, HV para recuperar el mejor modelo basado en métricas de rendimiento.
- Se propone un método novedoso para fusionar los modelos ML basados en la tasa de predicción falsa.
- Propuso un clasificador de votación multinivel ideal (tres niveles) para verificar la efectividad del modelo de conjunto.
- Se realiza un estudio comparativo para demostrar la efectividad de nuestro modelo propuesto.

El desempeño del sistema de votación multinivel se analiza usando parámetros como precisión, recordatorio, especificidad, curva ROC, *F*1 puntuación.

El trabajo restante está organizado de la siguiente manera: Art. 2 ofrece una breve descripción del trabajo relacionado realizado en el campo de la clasificación de noticias falsas. El enunciado del problema se discute en la secc.3. Sección4 cubre la metodología del sistema propuesto. Sección5 presenta los clasificadores utilizados para detectar los artículos falsos y reales. La fase de evaluación junto con el análisis realizado por todos los clasificadores se presenta en la secc.6. los

El modelo propuesto se discute en la secc. 7, y su desempeño se evalúa en la Sect. 8. La comparación del modelo propuesto con el trabajo existente se discute en la secc.9. Sección10 concluye el artículo junto con sus trabajos futuros.

2. Trabajo relacionado

Recientemente se han desarrollado muchas técnicas para la detección de noticias falsas. En esta sección, se discute el trabajo de investigación estrechamente relacionado que se ha realizado para detectar noticias falsas en las redes sociales en línea.

La mayoría de los sitios sociales requieren energía y tiempo para eliminar o filtrar manualmente el spam. Markines y col. (2009) propuso seis aspectos destacados (TagSpam, TagBlur, DomFp, NumAds, Plagiarism, ValidLinks) de los sistemas de etiquetado que detectan diversas propiedades del spam social (Markines et al. 2009). Utilizando los seis aspectos destacados propuestos, los creadores evaluaron diferentes técnicas de aprendizaje automático administradas para identificar el spam con una precisión superior al 98% con una tasa de falsos positivos del 2%. La herramienta Weka utilizada para el experimento proporciona la mejor precisión cuando se evalúa sobre la base del clasificador AdaBoost. Para abordar el problema de la detección de promotores de video y spammers, Benevenuto et al. (2009) reunió manualmente la recopilación de pruebas de clientes genuinos de YouTube y los clasificó como legítimos, spammers y promotores. Los autores han investigado la viabilidad de detectar spammers y promotores aplicando un algoritmo de clasificación supervisado (Benevenuto et al.2009). El clasificador utilizado en este artículo identifica correctamente a la mayoría de los promotores.

Qazvinian y col. (2011) exploró tres características como basado en contenido memes basados en redes y específicos de microblogs para identificar correctamente los rumores (Qazvinian et al. 2011). Estas funciones también se utilizan para identificar desinformadores o usuarios que respaldan un rumor y tratan de difundirlo. Para el experimento, los autores recopilaron 10,000 tweets anotados manualmente de Twitter y lograron 0,95 en precisión promedio (MAP). Rubin y col. (2016) propusieron un modelo de detección de sátira con algoritmo basado en Support Vector Machine (SVM) en 4 dominios, como ciencia, negocios, noticias blandas y cívica (Rubin et al. 2016). Para verificar las fuentes de los artículos de noticias, los autores han discutido varios sitios web de noticias legítimos y satíricos. En este artículo, se eligen cinco funciones juntas para predecir la mejor combinación de funciones de predicción con un 90% de precisión y un 84% de recordación para identificar noticias satíricas que pueden ayudar a minimizar el impacto de la sátira en el engaño.

El análisis del evento real como BostonMarathonBlasts es realizado por Gupta et al. (2013a). Durante el evento, se observó que muchos perfiles falsos y maliciosos se originaban en Twitter. Los resultados mostraron que el 29% del contenido se originó durante los BostonBlasts (Gupta et al.2013a) fue viral, mientras que el 51% fueron opiniones y comentarios genéricos. Autores identi fi caron seis mil perfiles que se crearon inmediatamente después de que ocurrieron las explosiones y fueron suspendidos por Twitter. Los tabloides se utilizan a menudo para sensacionalizar, exagerar, producir contenido de noticias engañoso y de baja calidad. Ha surgido una nueva forma de tabloidización conocida como clickbaiting. Existe clickbaiting tanto textual como no textual, que es examinado por Chen et al. (2015), quien propuso un enfoque híbrido (Chen et al. 2015) para la detección automática de clickbait.

Rubin y col. (2015) utilizaron el modelado del espacio vectorial (VSM) y teoría de la estructura retórica (RST) para analizar noticias engañosas y veraces. RST capta la coherencia de la historia en términos de relaciones funcionales entre las unidades de texto útiles y también describe la estructura jerárquica de cada artículo / noticia. VSM se utiliza para identificar las relaciones entre la estructura retórica (Rubin et al. 2015), es decir, el contenido de cada artículo se puede representar como vectores en un espacio de alta dimensión.

Los investigadores han utilizado diferentes técnicas para identificar y revisar el contenido falso. Uno de los mejores y más comunes métodos de extracción de características es Bag of Words. Comprende un grupo de palabras recuperadas del contenido textual, de donde ngram (Ahmed et al.2017) se pueden extraer características. La segunda característica más importante que es similar al enfoque de la Bolsa de palabras es la frecuencia de término (TF), que está relacionada con la frecuencia de las palabras. Conroy y col. (2015) propuso un enfoque híbrido que combina tanto el aprendizaje automático como las señales lingüísticas con datos de comportamiento basados en la red (Conroy et al. 2015). El enfoque híbrido sigue las técnicas de ngram y Bag of Words para representar datos. Ahmed y col. (2017) propuso un sistema de detección de noticias falsas que utiliza n-gram (Magdy y Wanas 2010) análisis y frecuencia de términos-frecuencia de documentos inversa (TF-IDF) como una técnica de extracción de características (Ahmed et al. 2017). En este artículo, se utilizan seis clasificadores de aprendizaje automático y se utilizan dos técnicas de extracción de características diferentes para la comparación y la investigación. Volkova y col. (2017) construyó un modelo predictivo para gestionar las publicaciones de 130Knews como verificadas o maliciosas. Los autores han clasificado cuatro subtipos de noticias sospechosas como propaganda, clickbait, engaños y sátira (Volkova et al.2017). Chhabra y col. (2011) ha presentado una función aURLstatic-método de detección basado en la detección de sitios web maliciosos con resultados precisos. El autor se ha centrado en características externas como las direcciones IP. Además, una construcción de vector VSM (Chhabra et al.2011) se elige como modelo de vector de URL. El conjunto de datos tomado en este documento consiste en URL maliciosas que se descargaron de la plataforma de phishing denominada 'Phishtank' (Aggarwal et al.2012).

En nuestro mundo digital, las noticias falsas se difunden e impactan a millones de usuarios en las plataformas de redes sociales todos los días. Realmente se ha vuelto difícil separar la verdad de la ficción. Con la ayuda de modelos de aprendizaje automático, es posible detectar correos electrónicos no deseados en una etapa temprana con la ayuda de filtros de correo no deseado. Los clasificadores ML ayudan a resolver los problemas del mundo real.

tabla 1 Análisis comparativo de estudios de investigación.

Autores	Enfoque propuesto	Modelo	Conjunto de datos	Características
Markines et al. (2009)	Se analizaron seis funciones distintas para detectar spammers en redes sociales mediante el aprendizaje automático.	SVM, AdaBoost	Publicaciones de spam, etiquetas	TagSpam, TagBlur, DomFp, NumAds, Plagio, Vínculos válidos
Benevenuto et al. (2009)	Se propone un rastreador de respuestas en video para identificar spammers en la red social de video en línea	SVM	Usuario real de YouTube información	Atributos de video, características individuales del comportamiento del usuario, relación social entre usuarios a través de interacciones de respuesta de video
Qazvinian et al. (2011)	Tweets identificados en los que se respalda el rumor	Bayes ingenuo	Tweets	Memes específicos de Twitter basados en contenido, basados en la red
Chhabra y col. (2011)	Utilizando las características estáticas de las URL, se desarrolla un método para detectar sitios web maliciosos	Naïve Bayes, Logística Regresión, DT, SVM-RBF, SVM-Lineal, SVM-Sigmoide	Conjunto de datos de URL maliciosas de 'Phishtank'	Gramática, Léxica, Vectores y Estática
Gupta y col. (2013a)	Análisis del contenido de Twitter durante la maratón de Boston	Regresión logística	Tweets e información de usuario correspondiente	Compromiso con el tema, Compromiso global, Social reputación, simpatía, credibilidad
Chen y col. (2015)	Se analizaron las relaciones de coherencia entre engaños y noticias veraces	VSM	Muestras de noticias de 'Bluff the Listener' de NPR	Discurso
Rubin y col. (2015)	Se propone un enfoque híbrido que combina datos de comportamiento lingüísticos y basados en redes.	Lingüísticos, modelos de red	Oraciones de texto simples	Bolsa de palabras, n-gram
Conroy y col. (2015)	Se desarrolla un modelo de detección de sátiras	SVM	Periódicos nacionales de EE. UU. Y Canadá	Absurdo, Humor, Gramática, Afecto negativo, Puntuación
Ahmed y col. (2017)	Desarrollado basado en n-gram clasificador para diferenciar entre anuncios falsos artículos reales	LinearSVM	Articulos de noticias	TF-IDF
Caetano y col. (2018)	Se construyó un modelo predictivo para predecir 4 subtipos de noticias sospechosas: sátira, engaños, clickbait y propaganda	Modelos lingüísticos	Publicaciones de noticias	TF-IDF, Doc2Vec
Propuesto sistema	Utilizando datos textuales de los artículos, se desarrolla un modelo de votación eficiente en varios niveles para detectar artículos falsos.	SGD, PA, MultinomialNB, Aumento de gradiente, DT, AdaBoost	Articulos de noticias	TF-IDF, Count-Vectorizer, Hashing-Vectorizer

Además, ML ha facilitado a los usuarios del negocio de comercio electrónico, ya que ayuda a identificar el patrón oculto, agrupa los productos similares en un clúster y muestra el resultado al usuario final, lo que permite un sistema de recomendación basado en productos. También ayuda a resolver el problema de las recomendaciones injustas (D'Angelo et al.2019).

El análisis comparativo del trabajo relacionado realizado en el campo de la detección de noticias falsas y el sistema propuesto presentado en este documento se muestra en la Tabla 1.

Se ha analizado que el trabajo de investigación realizado en el campo de la detección de noticias falsas se restringe principalmente a clasificadores SVM e Naïve Bayes utilizando únicamente n-gram (Magdy y Wanas 2010) y TF-IDF presenta enfoques de extracción. No se ha realizado ningún trabajo sobre el perceptrón multicapa (MLP), la memoria a largo plazo a corto plazo (LSTM) (LeCun et al.2015) modelos y un enfoque de extracción basado en hash que también representa una mejor eficiencia. Además, la detección de noticias falsas existente (Ruchansky et al.2017) los modelos se construyen utilizando algoritmos de aprendizaje automático supervisados, mientras que la extracción manual de características es un método ineficaz y que consume más tiempo para lograr la mayor precisión. Las técnicas existentes estudiadas hasta ahora proporcionan una dirección a seguir en la investigación cuantitativa y cualitativa.

3 Enunciado del problema

Para abordar el problema de la generación y difusión de noticias falsas a través de varias plataformas sociales en línea, se elige una técnica de extracción de características adecuada para mejorar la eficiencia.

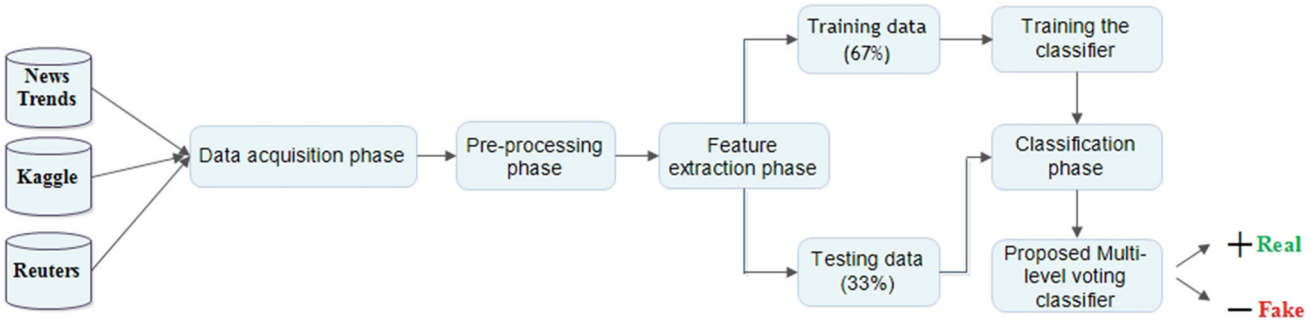

Figura 1 Arquitectura del sistema de detección automática de noticias falsas propuesto

eficiencia de los clasificadores de ML existentes. Se propondrá un modelo novedoso de conjunto de votaciones multinivel para desarrollar un sistema eficiente de detección de noticias falsas. Matemáticamente, el enunciado del problema se puede representar de la siguiente manera: para identificar $S = \{$falso, real$\}$ para un documento D dónde $D = \{t_1, t_2, ..., t_{norte}\}$ y t_I representa el texto en un artículo de noticias a_I elegido de un corpus con una serie de compromisos que se compone de título, cuerpo y etiqueta del artículo como $mi_{yo\,jk} = (t_I, B_j, l_k)$. La tarea es evaluar y analizar el mejor método de extracción de características. F_{metro} dónde $m = \{$TF-IDF, CV, HV$\}$ utilizando un clasificador de aprendizaje automático para calcular la alta eficiencia en nuestro sistema propuesto. El enfoque seguido en este documento se discutirá en la siguiente sección.

4 Metodología

La arquitectura del sistema de detección de artículos falsos propuesto (Ahmed y Abulaish 2012) se muestra en la Fig. 1. Para entrenar el sistema, se han recopilado tres corpus de tres fuentes diferentes descargando los conjuntos de datos de los sitios web de News Trends, Kaggle y Reuters. En la fase de preprocesamiento, se eliminan las palabras vacías y el texto duplicado de los artículos de noticias. Los valores faltantes, es decir, los valores no disponibles (NA), se recopilan y limpian en el siguiente paso. Los datos recuperados se dividen en dos partes, conjuntos de entrenamiento (0,67) y de prueba (0,33). A continuación, se lleva a cabo la fase de extracción de características para recuperar características significativas de los datos textuales. En esta fase, las características se extraen de los artículos. Tres técnicas de extracción de características, como Frecuencia de términos-Frecuencia de documento inversa (asigna ponderaciones de acuerdo con la importancia de los términos en el documento), Se han aplicado Count-Vectorizer (cuenta la frecuencia de los términos en un documento) y Hashing-Vectorizer (sigue el truco de hash). Las características recuperadas se envían luego al algoritmo de clasificación elegido en la siguiente fase. Los diversos modelos ML como MultinomialNB, Passive Aggressive, Stochastic Gradient Descent, Logistic Regression, Support Vector Classi fier, Nu-Support Vector Classi fier, Multi-Layer Perceptron, Linear SupportVectorClassi fier, AdaBoost, Gra-Impulso al cliente, árbol de decisión, clasificadores de votación (Sirajudeen et al. 2017) son elegidos para aprender e identificar los patrones y resultados de ellos. Luego, los modelos se evalúan en función de las métricas de rendimiento para lograr un clasificador eficiente. Con base en el análisis realizado, los modelos se integran para proponer un modelo de votación multinivel para lograr una alta eficiencia y luego se compara con el trabajo existente (Gao et al.2010) como se discute en la secc. 9. El trabajo detallado de cada fase que se ha implementado en el marco de Python se analiza a continuación.

4.1 Recolección de datos

Hay muchas fuentes de generación de artículos falsos como Facebook (Dewan y Kumaraguru 2015) y Twitter (Aggarwal et al. 2018; Gupta y Kumaraguru2012b, a; Gupta y col.2013b), que se utilizan como plataforma comercial para difundir noticias falsas. Hemos utilizado Tendencias de noticias (https://bit.ly/2zVRLxK), Kaggle (https://bit.ly/2Ex5VsX) y Reuters (https://bit.ly/2BmqBQE) conjunto de datos con atributos similares como titulares, cuerpo, nombre del editor del artículo, fecha de publicación, valores categóricos y faltantes. El corpus de News Trends, Kaggle y Reuters consta de 7.795; 4.048 y 21578 artículos de noticias etiquetados como noticias falsas y reales, respectivamente. Las estadísticas generales de nuestros tres conjuntos de datos recopilados se discuten en la Tabla2. Para analizar la distribución de la longitud de los títulos para artículos falsos y reales, la Fig.2 se visualiza, donde el *X*-El eje etiquetado como 'título' representa el número de términos utilizados en títulos o titulares de artículos de noticias, mientras que *Y*-El eje representa el número correspondiente de artículos que tienen la misma distribución de longitud. Se puede sacar una conclusión a partir de la distribución media visualizando la Fig.2 que la longitud de los títulos o titulares de los artículos de noticias falsos suele ser mayor que la de los artículos de noticias reales.

Para analizar más a fondo la distribución de la longitud del contenido corporal en los artículos, la Fig. 3 se observa, donde el *X*-El eje etiquetado como 'texto' representa el número de términos utilizados en los textos de los artículos de noticias, mientras que *Y*-El eje representa el número correspondiente de artículos que tienen la misma distribución de longitud.

Se ha analizado que la longitud del cuerpo / texto de los artículos de noticias reales suele ser mayor que la de los artículos de noticias falsos.

Tabla 2 Estadísticas de corpus recopilados

Cuerpo	Artículo total	Artículos limpios	Articulos reales	Artículos falsos	Año de publicación
Tendencias de noticias	7795	6335	3171	3164	2017
Kaggle	4048	3983	1865	2118	2017
Reuters	21.578	19,969	9622	10,347	2004

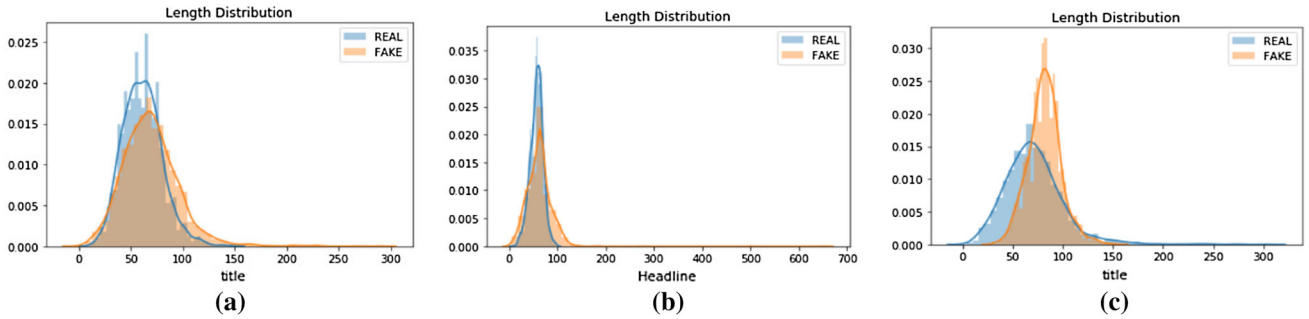

Figura 2 Distribución de la longitud del título para artículos falsos y reales sobre **a** Tendencias de noticias, **B** Kaggle y **C** Reuters corpora

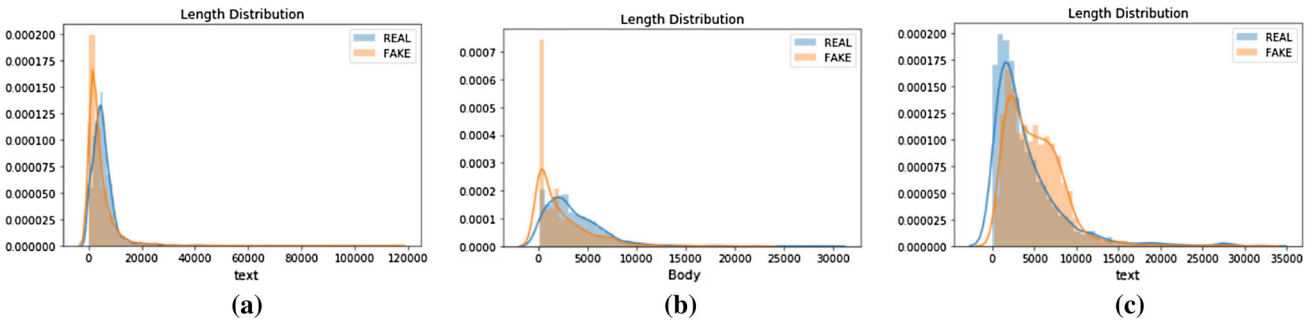

Fig. 3 Distribución de la longitud del texto para artículos falsos y reales sobre **a** Tendencias de noticias, **B** Kaggle y **C** Reuters corpora

como se muestra en la Fig. 3a, b, pero para el corpus de Reuters la longitud de los artículos de noticias reales es más que los artículos de noticias falsos como se ve en la Fig. 3C.

Las estadísticas de la distribución media de las Figs. 2 y 3 se comparan en la tabla 3. En general, después de analizar diferentes conjuntos de datos, se puede llegar a una conclusión de que los titulares de los artículos falsos son más largos y tienen un contenido corporal más corto que los artículos reales publicados en las redes sociales (Pontes et al. 2012a) sitios.

4.2 Preprocesamiento

En la fase de preprocesamiento, las palabras no semánticas, como preposiciones, conjunciones y pronombres, también conocidas como palabras vacías, se eliminan del documento textual, ya que proporcionan muy poca o ninguna información sobre el contenido falso de un artículo. Los datos redundantes en forma de cadenas textuales se eliminan del documento utilizando una expresión regular (*expresión regular)* en el siguiente paso como se muestra en la Fig. 4. los*regex* y *pandas* biblioteca se ha utilizado para realizar la tarea de preprocesamiento (Batchelor 2017). *Regex* la biblioteca se ha utilizado en Python para de fi nir un patrón de búsqueda que utiliza una secuencia de caracteres, mientras que *dropna* método de *pandas* se usa para limpiar los valores faltantes en pythonDataFrame. *estado_aleatorio* función para seleccionar las entradas del conjunto de datos que se utiliza además para dividir puntos de datos de entrenamiento y prueba, ya que se utiliza para dividir los datos de forma aleatoria.

Para evitar un ajuste excesivo, se utilizaron tres divisiones estándar (70:30, 67:33 y 60:40) para realizar el experimento. Cuando se realizó la primera división estándar (70:30), se observó que el punto de datos trataba un problema de ajuste insuficiente. Durante la segunda división (60:40), se analizó el sobreajuste de los datos, mientras que la tercera división (67:33) dio la mejor línea predicha que cubría la mayoría de los puntos de datos en el gráfico, por lo que se eligió una división estándar de 67:33. Luego, los datos de entrenamiento se envían a la fase de generación de características, como se explica en la siguiente sección.

4.3 Generación de funciones

Para extraer características numéricas de un documento textual, se realizan tokenización, recuento y normalización. Durante la tokenización, a cada palabra se le asigna un número entero único.

Tabla 3 Distribución media de artículos etiquetados para News Trends, Kaggle y Reuters corpora

Cuerpo	Distribución media del título etiquetado como falso	Distribución media de título etiquetado como real	Distribución media del texto etiquetado como falso	Distribución media del texto etiquetado como real
Tendencias de noticias	69,18	61,38	4121.04	5292.16
Kaggle	62,47	57,32	2380.82	3544.84
Reuters	81,79	71,29	5156.08	4540.26

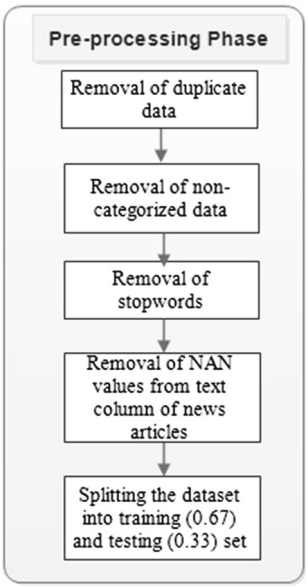

Figura 4 Pasos realizados durante el preprocesamiento fase

IDENTIFICACIÓN, después de lo cual se cuenta la aparición de tokens y luego tiene lugar la normalización de dichos tokens. Todo el proceso de convertir el documento textual en vector de características numéricas se denomina vectorización. En conjunto, esta estrategia (tokenización, recuento, normalización) se denomina 'Bolsa de n-gramas' o 'Bolsa de palabras' donde *norte* representa la secuencia continua de términos (Alahmadi et al. 2013). Las tres técnicas de extracción de características que se muestran en la Fig.5 para recuperar características del contenido textual de un artículo son Count-Vectorizer, TF-IDF yHashing-Vectorizer que utilizan *CountVectorizer, T fi dfVectorizer* y *HashingVectorizador* clases de *extracción de características* biblioteca de python, respectivamente.

4.3.1 Vectorizador de conteo (CV)

Count-Vectorizer se basa absolutamente en el recuento de apariciones de palabras en el documento. En la técnica Count-Vectorizer, se realizan tanto el recuento de ocurrencias de tokens como el proceso de tokenización. Count-Vectorizer tiene muchos parámetros para refinar el tipo de características. Uno puede construir características usando cualquiera de los tres parámetros, unigram (*min_d f* = 1), bigrama (*min_d f* = 2) y trigramas (*min_d f* = 3). Hemos usado *mente F* como 1 para nuestro experimento. Aquí, cada vector (término) en un documento representa el nombre de la característica individual y su ocurrencia se representa a través de una matriz para que sea más fácil de entender como se muestra en la Tabla 4. Se ha observado en la tabla anterior, después de la fase de preprocesamiento, los términos recuperados de los documentos se representan como vectores en la parte superior de la matriz dispersa y la frecuencia de los términos en un documento particular se representa mediante el recuento de ocurrencias. Las nubes de etiquetas de las 30 características principales recuperadas después de ejecutar el método CV se muestran en la Fig.6. Se observó que palabras como *corrupción, atacando, islámico, obama, perdiendo, com* se ven bajo nubes de etiquetas falsas.

CV también cuenta el número de palabras que aparecen con más frecuencia en el documento, lo que puede eclipsar las palabras que aparecen con menos frecuencia pero que pueden tener más importancia para la función del documento. Esta limitación de CV se puede manejar utilizando la técnica de extracción de características TF-IDF como se explica a continuación.

4.3.2 Término Frecuencia – Frecuencia inversa del documento (TF – IDF)

TF-IDF es una matriz de ponderación que se utiliza para medir la importancia de un término (recuento + peso) para un documento en un conjunto de datos. Los tokens recuperados de los datos textuales utilizando las técnicas TF-IDF y CV son los mismos, pero los pesos asignados a los tokens de ambas técnicas son diferentes. TF-IDF se compone de dos métricas, denominadas frecuencia de término (tf) y frecuencia de documento inversa (idf). TF-IDF está representado por Eq. (1).

$$t\,fi\,df = tf\,(t,\,d) \times idf t,\,d) \tag{1}$$

Aquí, la frecuencia de término se denota como tf y se calcula a partir del recuento (*C),* término (*t)* en el documento *D)* y representado como *tf (t, d) = ctd.* La frecuencia de aparición de palabras a una característica binaria se convierte utilizando 1 (presente en el documento) y 0 (no presente en el documento). Las frecuencias se pueden normalizar utilizando promedios y logaritmos. La frecuencia inversa del documento (idf) para una palabra *w* en el texto del documento *t)* calculado por la ecuación. (2).

$$idf t,\,d) = 1 + \text{registro}\ \frac{T}{(1 + gl\,(t))} \tag{2}$$

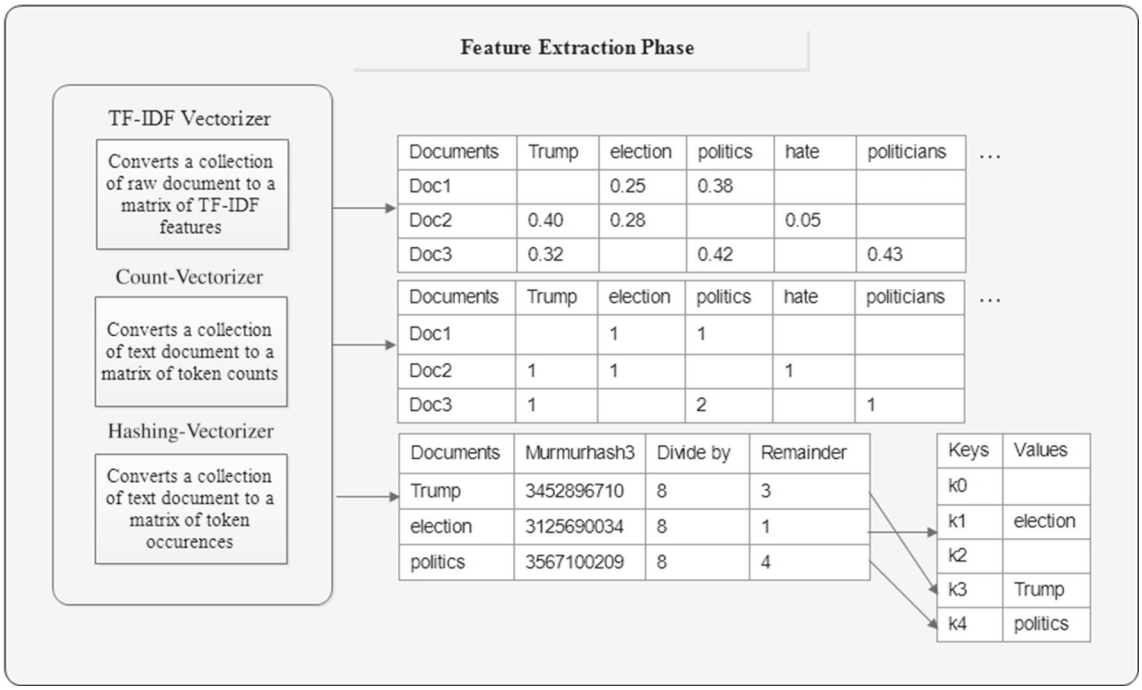

Figura 5 Conversión de contenido textual en vectores numéricos a través de las técnicas de extracción de características TF-IDF, Count-Vectorizer y Hashing-Vectorizer

Cuadro 4 Representación de matriz dispersa usando un Técnica de extracción de características Count-Vectorizer

Documento	Narendra	Elecciones	Votar	Política	Punjab	BJP	Candidato
Doc1	0	1	2	0	0	2	1
Doc2	4	0	1	0	0	1	0
Doc3	1	3	2	0	1	0	1

Aquí, T representa el total recuento de documentos en nuestro corpus y df (t) representa el recuento del número de documentos donde el término t está presente. El producto de dos medidas ayudará a calcular t fi df. La forma normalizada de Euclidean se usa para calcular la métrica final TF-IDF dada por la Ec. (3).

$$\text{t fi df} = \frac{\text{t fi df}}{||\text{t fi df}||} \tag{3}$$

Aquí, || t fi df || es la norma euclidiana. Las nubes de etiquetas de las 30 características principales para artículos falsos y reales que utilizan la técnica de extracción de características TF-IDF se muestran en la Fig.7. Se observó que palabras como *www, peligro, lo siento, guerras, islámico* son más comunes en artículos de noticias falsos. La diferencia entre el enfoque Count-Vectorizer y TF-IDF es que los tokens recuperados de los datos textuales son los mismos, pero ambos tienen pesos diferentes asignados a cada token que se extrae.

4.3.3 Hashing-Vectorizador (HV)

Es una técnica que ahorra memoria. A diferencia de las dos técnicas anteriores, los tokens se almacenan como una cadena y aquí el truco de hash se aplica para codificar las características como índices numéricos. Analicemos el concepto de HV utilizando un ejemplo que se muestra en la Fig.8. Cuando se ingresan datos, se recuperan los atributos hash de los datos. Los términos hash como Trump, elección y política se extraen del documento. En el siguiente paso, el truco hash se aplica a los atributos hash, donde un *Murmurhash3* La función se aplica a los términos hash para generar un número aleatorio. Además, los números aleatorios asignados se dividen entre 8 y se almacenan en diferentes claves, como *k*2, *k*3, *k*4 basado en los restos recuperados después de aplicar el *murmurhash3* función que se utiliza para búsquedas basadas en hash. Existe la posibilidad de colisión cuando los datos tienen los mismos atributos hash.

Supongamos en nuestro documento de ejemplo, tenemos a Trump y las palabras de política como claves importantes que se ven más de una vez, provocando así colisiones en *k*3, *k*4 posiciones. Los valores colisionados son ocupados por otras posiciones vacantes en un conjunto de documentos. Este procesamiento de colisiones se trata con procesamiento paralelo. El proceso se explica conceptualmente en la Fig. 9. A estos seis términos se les asignan seis claves como se muestra en la Fig.9 y se ingresan en la tabla hash. Los valores hash de las claves *k*1, *k*3, *k*6 son iguales, es decir, *Triunfo; k*4 y *k*5 son iguales,

Figura 6 Top 30 **a** Tendencias de noticias falsas, **B** Tendencias de noticias reales, **C** Kaggle falso **D** Kaggle real **mi** Reuters falso y **F** Nubes de palabras reales de Reuters que utilizan la técnica de extracción de características Count-Vectorizer

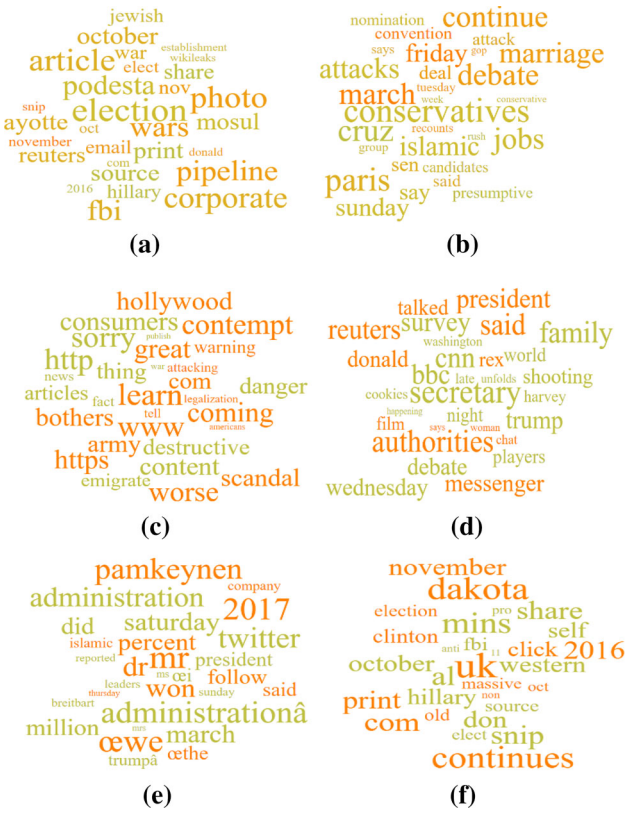

Figura 7 Top 30 **a** Tendencias de noticias falsas, **B** Tendencias de noticias reales, **C** Kaggle falso **D** Kaggle real **mi** Reuters falso y **F** Nubes de palabras reales de Reuters que utilizan la técnica de extracción de características TF-IDF

es decir, la política, pero el resto tiene valores diferentes. Debido a la colisión, *k*1, *k*3 y *k*6 no se pueden colocar en el mismo conjunto (*S*1). Para habilitar el procesamiento paralelo,*k*1, *k*3 y *k*6 se colocan en diferentes conjuntos. No se pueden colocar dos valores hash iguales en un solo conjunto. Diferentes claves como*k*1, *k*2 y *k*4 se pueden colocar en el mismo conjunto *(S*1) ya que estas claves tienen valores diferentes. Valores en llaves,*k*3 y *k* 6, son iguales, por lo que no se pueden procesar en paralelo; por lo tanto, se procesan en diferentes conjuntos. Los valores en*S*1, *S*2, *S*3 están organizados en vectores (características numéricas) y se pueden procesar mediante operaciones vectoriales.

El inconveniente de HV es que no hay forma de obtener los nombres de las características de los índices de características, es decir, la transformación inversa no se puede usar para calcular los nombres de las características más importantes a través de Hashing-Vectorizer, a diferencia de los otros dos métodos.

5 algoritmos de clasi fi cación

El conjunto de datos procesados recuperado después de la fase de preprocesamiento y extracción de características se envía a la fase de clasificación para la identificación de artículos de noticias falsos. En este artículo, seis técnicas de aprendizaje automático, es decir, Naïve Bayes (MultinomialNB), Support Vector Machine [Support Vector Classi fi er

Documents	Murmurhash3	Divide by	Remainder
Triump	3452896710	8	3
election	3125690034	8	2
politics	3567100209	8	4

Keys	Values
k0	-
k1	-
k2	election
k3	Triump
k4	politics

Figura 8 Truco de hash en datos textuales usando la función Murmurhash3 para obtener valores en un rango específico

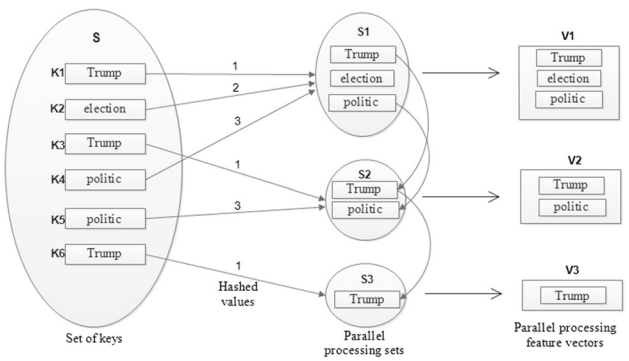

Figura 9 Entrada de datos redundantes en una tabla hash durante el procesamiento en paralelo

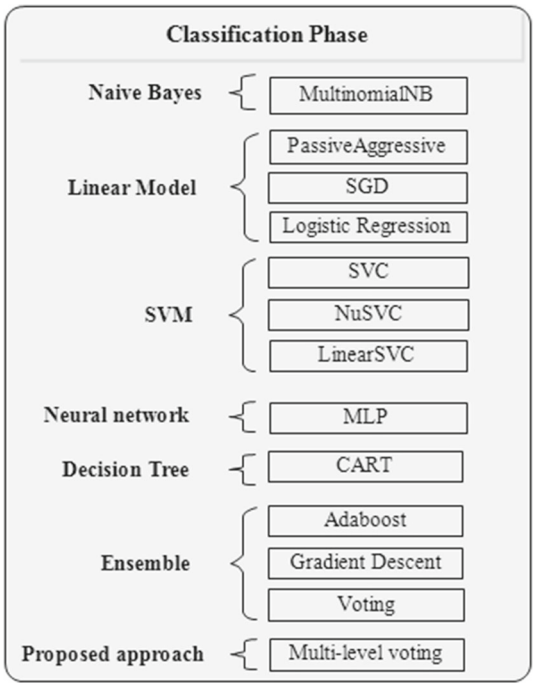

Figura 10 Fase de clasi fi cación

(SVC), NuSVC, LinearSVC], árbol de decisión (CART), lineal [pasivo agresivo (PA), descenso de gradiente estocástico (SGD), regresión logística (LR)], red neuronal [perceptrón multicapa (MLP)] y modelos de conjunto (AdaBoost, Gradient Boosting, Voting) se han aplicado como se muestra en la Fig. 10.

Las funciones del clasificador ayudan a mapear los vectores de características de entrada $F \varepsilon F$ para imprimir etiquetas $l \varepsilon \{1, 2, 3, ..., norte\}$, dónde F es el espacio de características. El espacio de características se representa como $F = \{\text{Falso, real}\}^R$, dónde R es el número real. Nuestro objetivo es aprender la función del clasificador a partir de datos de entrenamiento etiquetados.

5.1 Bayes ingenuo (NB)

Es un tipo de clasificador de probabilidad. Funciona con el teorema de Bayes y maneja variables categóricas y continuas. NB asume que cada par de características con valor etiquetado es independiente entre sí. Dada una colección de D documenciones de artículos de noticias, $Di = \{D1, D2, ..., Dnorte\}$, donde cada documento consta de T términos como $Di = \{t1, t2, ..., tmetro\}$. Entonces, la probabilidad de DI ocurrencia en la etiqueta de clase CI es dado por la Ec. (4).

$$\text{ORDENADOR PERSONAL}_{i\,|Di)} = \text{ORDENADOR PERSONAL}_{I)} \prod_{n=1}^{tmetro} P\,(dn\,|\,CI) \tag{4}$$

Aquí, la probabilidad condicional del término $tmetro$ presente en un Documento de etiqueta de clase CI y la probabilidad previa de que el documento ocurra en la etiqueta de clase CI se denota por $\text{ORDENADOR PERSONAL}_{I)}$.

Multinomial Naïve Bayes (MultinomialNB) es un tipo de algoritmo Naïve Bayes utilizado para la clasificación de texto. Los datos utilizados en la clasificación de texto para aplicar MultinomialNB se pueden representar como *TF-IDF* vectores, hash de vectores y conteo de vectores.
Los vectores de características $V\,f = (V\,F1, V\,F2, ..., V\,fn)$ están parametrizados para cada clase $Cnorte$ en la distribución, donde *norte* representa los números de función. La probabilidad de observar $V\,F$ es dado por Eq. (5).

$$\text{ORDENADOR PERSONAL}_{n\,|pag)} \prod_{n=1}^{PAG} F_{jnorte!} \prod_{n=1}^{} {}^{V_{norte}\,F_{norte}} \tag{5}$$

Aquí, $Fnorte$ es el número de veces que *norte* ha ocurrido la característica en el documento, XI es el número de extracciones extraídas del bolsa de características. $V\,{}^{F_{norte}}{}_{norte}$ y $F_{jnorte!}$ se calculan a partir de datos de entrenamiento.

5.2 Modelo lineal

El modelo lineal ayuda a clasificar el grupo haciendo combinaciones lineales de vectores de características. Los clasificadores lineales funcionan bien con muchas características, pero funcionan mejor para la clasificación de documentos (las características se extraen del texto). Si **v** es el vector de características de entrada al clasificador, entonces la puntuación resultante viene dada por la ecuación. (6).

$$s = f\,(\mathbf{vw}) = F\left(\sum_{I} wI\,VI\right) \tag{6}$$

aquí **w** es el peso de un vector de características y la función F da la salida de deseos de dos vectores. Los tres modelos lineales utilizados en este artículo son clasificadores pasivo agresivo (PA), descenso de gradiente estocástico (SGD) y regresión logística (LR). El algoritmo de PA tiene un comportamiento similar con el clasificador de perceptrón en términos de tasa de aprendizaje, pero tiene un comportamiento diferente en términos de parámetro de regularización. El clasificador PA es equivalente a PA-1 (Dewan y Kumaraguru 2015) cuando el parámetro de pérdida es *bisagra* y PA-ll (Dewan y Kumaraguru 2015) cuando el parámetro de pérdida *bisagra_cuadrada.* El segundo modelo lineal utilizado en este artículo es SGD. El modelo SGD se actualiza con la tasa de aprendizaje decreciente después de cada intervalo de muestra y, de forma predeterminada, el parámetro de pérdida utilizado en este documento es *bisagra.* El clasi fi cador también permite el aprendizaje por minibatch. El modelo LR se puede utilizar para la clasificación de problemas tanto binarios como múltiples. Cualquier otro formato de entrada aparte de *fl oat64* se convierte en este clasificador. Los tres modelos lineales discutidos anteriormente pueden tomar tanto la matriz dispersa como la densa como entrada.

5.3 Máquina de vectores de soporte (SVM)

SVM trabaja con el principio de minimización de riesgos estructurales. Se define por un 'mejor' hiperplano separador y también se denomina como

un clasificador discriminativo. A través del modelo SVM, los vectores de características recuperados de documentos de texto de artículos de noticias se representan como puntos en el espacio de características. Luego, los vectores de características se mapean de tal manera que se ve un espacio amplio para realizar una clasificación lineal. En nuestro conjunto de datos, los vectores de características están marcados haciendo dos categorías, $C = \{$ Falso, Real$\}$, y luego el clasificador de entrenamiento construye un modelo que asigna nuevos vectores de características a ambas categorías definidas.

Las clases de SVM como el Clasificador de vectores de soporte lineal (LinearSVC), el Clasificador de vectores de Nu-Support (NuSVC), el Clasificador de vectores de soporte (SVC) se utilizan para realizar la clasificación en el conjunto de datos. NuSVC y SVC son casi similares pero usan conjuntos de parámetros ligeramente diferentes y sus formulaciones matemáticas también varían, mientras que LinearSVC es otro tipo de clasificación de vectores de soporte (SVC) y usa el caso del kernel lineal. Las tres clases toman entrada en forma de dos arreglos con arreglo X tener un tamaño bidimensional [sample_number, feature_vectors] para manejar los datos de entrenamiento y la matriz Y tener un tamaño bidimensional [category_label, numero de muestra]. La función de decisión que es la misma ambos para SVC y NuSVC viene dada por Eq. (7).

$$sgn\left(\sum_{f=1}^{norte} y_F \alpha_F K(V_f, V) + \mu\right) \quad (7)$$

dónde V_F son los vectores de características de entrenamiento, $f = 1, 2, \ldots,$ *norte* en dos categorías. $K(V_f, V)$ es el kernel y $y_F \alpha_F$ es el parámetro de coeficiente dual que contiene vectores de soporte y un término de intersección independiente μ. La única diferencia entre SVC ($C = [0,\infty]$) y NuSVC ($C = [0, 1]$) se ve desde el parámetro C que es el parámetro de penalización del término de error. La clase LinearSVC admite entradas tanto dispersas como densas y se implementa en términos de liblinear, por lo que es más flexible en términos de función de pérdida y penalizaciones, y se adapta mejor a muestras de prueba grandes.

5.4 Red neuronal (NN)

Las redes neuronales están compuestas por neuronas altamente interconectadas para resolver problemas específicos de manera paralela. En este documento, el perceptrón multicapa (MLP) se implementa en nuestro conjunto de datos recopilados. El clasificador se puede entrenar en cualquiera de las dos regresiones conjunto de datos de clasificación o clasificación. Los vectores de características $V_f = \{v_1, v_2, \ldots, v_{norte}\}$ se recuperan después de la fase de extracción de características y, a través del conjunto de datos de entrenamiento, el clasificador aprende una función determinada ción en Eq. (8).

$$f(n): Ri \rightarrow Ro \quad (8)$$

dónde I son las dimensiones de entrada y o son las dimensiones de una salida. En MLP, puede haber una o más capas no lineales (capa oculta) entre la capa de entrada y la de salida. La entrada La capa está formada por neuronas, donde cada neurona representa la característica de entrada que se alimenta a la capa oculta. Luego, la capa oculta calcula la suma ponderada $w_1 v_1 + w_2 v_2 + \cdots + w_I v_I$, seguido de función $f(n)$. El valor de salida lo da la última capa oculta y lo recibe el capa de salida.

5.5 Árbol de decisión (DT)

Los clasi fi cadores DT se pueden utilizar tanto para regresión como para clasi fi cación. El clasificador predice la variable objetivo aprendiendo los datos de la característica y dividiendo el área en subregiones. En base a dos criterios, se dividen múltiples características: una es una medida de impureza y otra es la ganancia de información. En nuestro conjunto de datos, *'gini'* es la medida de impureza elegida para calcular la mayor ganancia de información en cada nodo para dividir el DT. En el caso de los datos del documento, las condiciones dependen del término particular en un documento de texto del artículo de noticias. Los datos se dividen repetidamente hasta que el nodo hoja no se puede dividir más adquiriendo la menor información sobre ellos. La mayoría de las etiquetas en el nodo hoja se utilizan para clasificar los datos textuales.

5.6 Métodos de conjunto

Tales métodos ayudan a construir un algoritmo de aprendizaje combinando los estimadores para obtener un clasificador robusto sobre un clasificador único. Los métodos de refuerzo implementados en nuestro conjunto de datos son Gradient Descent Boosting (GDB) y el clasificador AdaBoost para la clasificación binaria. El clasi fi cador AdaBoost asigna más peso a los vectores de características que son difíciles de manejar y menos peso a las características que pueden manejarse fácilmente. Este proceso se repite hasta que el clasificador clasifica correctamente los datos de entrenamiento. El modelo GDB funciona con tres elementos, como el alumno débil, la función de pérdida y el modelo aditivo, como se explica en Shu et al. (2017).

El otro tipo de clasificador que puede ser útil para equilibrar las debilidades individuales de los modelos ML es VotingClassi fier. En este artículo, el clasificador de votaciones ha predicho las categorías basadas en la votación 'dura' que clasifica la muestra según la etiqueta de la clase mayoritaria. Para evaluar los clasificadores discutidos anteriormente, se definen las métricas de desempeño y los resultados experimentales correspondientes se discuten en la siguiente sección.

6 Fase de evaluación

Las medidas de rendimiento para clasificadores binarios aplicadas en este documento se evaluaron con la ayuda de una matriz de confusión definida por cuatro celdas, como se muestra en la Tabla 5, dónde

Cuadro 5 Una representación de matriz de confusión

Real↓ Predicho→	Falso	Verdadero
Falso	TP (a)	FP (b)
Verdadero	FN (c)	TN (d)

- la celda 'a' cuenta el documento predicho como 'Falso' cuando en realidad es 'Falso', lo que se conoce como tasa de verdaderos positivos (TP).
- la celda 'b' cuenta el documento predicho como 'Real' cuando en realidad es 'Falso', lo que se conoce como tasa de falsos positivos (FP).
- la celda 'c' cuenta el documento predicho como 'Falso' cuando en realidad es 'Real', lo que se conoce como tasa de falsos negativos (FN).
- la celda 'd' cuenta el documento predicho como 'Real' cuando en realidad es 'Real', lo que se conoce como tasa de verdadero negativo (TN).

La medida de rendimiento convencional se ha evaluado a partir de las celdas de matriz de confusión anteriores. Las medidas calculadas a partir de la matriz son precisión representada por la ecuación. (9), recuerda por Eq. (10), especificación por la ecuación. (11), precisión por Eq. (Eq. (12), error por 13), *F*1 puntuación de la Ec. (14) como se muestra debajo.

$$\text{Precisión (Pr)} = \frac{a}{a+b} \tag{9}$$

$$\text{Recordar (Re)} = \frac{a}{a+c} \tag{10}$$

$$\text{Especificidad (Sp)} = \frac{D}{d+b} \tag{11}$$

$$\text{Precisión (Acc)} = \frac{a+d}{n}, \text{ dónde } n = a+b+c+d > 0 \tag{12}$$

$$\text{Error (Err)} = \frac{b+c}{n}, \text{ dónde } n = a+b+c+d > 0 \tag{13}$$

$$F1 \text{ puntuación } (F1) = \frac{2 \times Pr \times Re}{Re + Pr} \tag{14}$$

Las predicciones realizadas por los modelos de clasificación se evalúan en esta fase en función de sus métricas de rendimiento. La medida de rendimiento más intuitiva es la precisión, que ayuda para predecir el mejor modelo. Varios modelos de aprendizaje automático utilizados en el experimento son MultinomialNB (*C*1), pasivo agresivo (*C*2), Descenso de gradiente estocástico (*C*3), regresión logística (*C*4), SVC (*C*5), NuSVC (*C*6), LinearSVC (*C*7), perceptrón multicapa (*C*8), árbol de decisión (*C*9), AdaBoost (*C*10), descenso en gradiente (*C*11), Votación (*C*12) y votación multinivel (*C*13) clasificadores. Para evaluar estos modelos, se muestra un análisis comparativo en la Fig.11. El experimento se realiza en tres corpus diferentes (News Trends, Kaggle y Reuters). En este artículo, se utilizan técnicas de extracción de características TF-IDF, CV y HV para extraer los vectores de características de los documentos del corpus elegido. En mesa6, se compara la medida de precisión de varios clasificadores ML.

La mejor precisión obtenida por los 3 modelos principales en los tres corpus es el clasificador de vector de soporte lineal (LSVC), los clasificadores pasivo agresivo (PA) y de regresión logística (LR).

Otros parámetros usados para evaluar la medida de desempeño de los clasificadores usados en este documento son precisión, recuperación y *F*1 puntuación. La métrica de precisión ayuda a calcular la proporción de artículos de noticias que se predice que son falsos y, en realidad, también pertenece a la categoría de artículos de noticias falsos. El análisis comparativo de la métrica de precisión se muestra en la Tabla7.

La métrica de recuperación ayuda a calcular la proporción de artículos noticiosos que se predice que son falsos, pero que en realidad pertenecen tanto a artículos falsos como reales. El análisis comparativo de la métrica de recuperación se muestra en la Tabla8. La métrica específica ayuda a calcular la proporción de artículos de noticias que se predice correctamente como un artículo de noticias real que se sabe que no es falso. La especificidad se mide como inversa a la métrica de recuerdo. El análisis comparativo de la métrica específica se muestra en la Tabla9. La precisión solo se mide como una métrica sólida cuando los valores de falso negativo y falso positivo están más cerca entre sí; de lo contrario, la métrica no se considera una buena medida de rendimiento. Para transmitir un equilibrio entre recuerdo y precisión, *F*La métrica de rendimiento de 1 puntuación se selecciona para recuperar el mejor modelo. *F*La métrica de 1 puntuación ayuda a tener en cuenta tanto los falsos positivos como los falsos negativos. El análisis comparativo de*F*La métrica de 1 puntuación se muestra en la tabla 10.

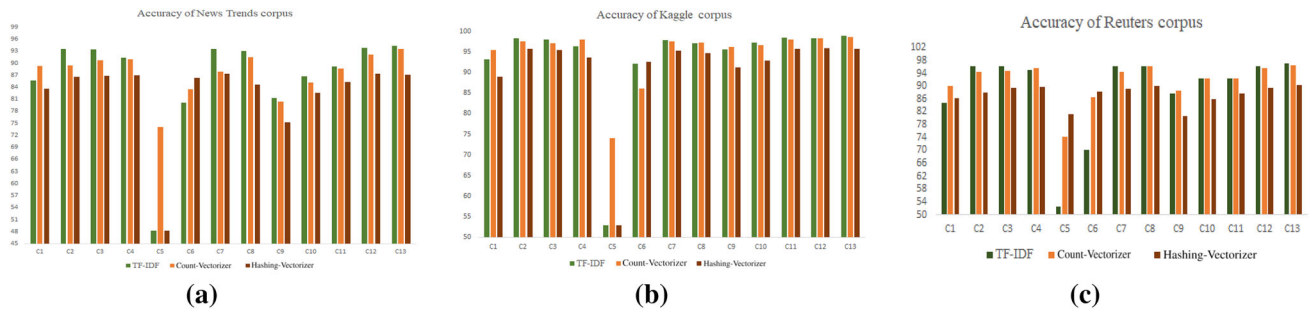

Figura 11 Análisis de rendimiento utilizando métricas de precisión para **a** Tendencias de noticias, **B** Kaggle y **C** Conjunto de datos de Reuters sobre la base de TF-IDF, Hashing-Vectorizer y técnicas de extracción de características Count-Vectorizer

Tabla 6 Análisis comparativo de la medida de precisión utilizando clasificadores de aprendizaje automático

Modelos	Tendencias de noticias			Kaggle			Reuters		
	TF-IDF	CV	HV	TF-IDF	CV	HV	TF-IDF	CV	HV
Multinomial Naïve Bayes	85,7	89,3	83,6	93,2	95,4	89	84,8	89,9	86,2
Pasivo Agresivo	**93,5**	89,4	86,6	**98,3**	**97,6**	**95,7**	**96,2**	94,3	88
Regresión logística de descenso	93,4	**90,7**	86,8	98	97,1	95,5	**96,2**	94,8	89,4
de gradiente estocástico	91,4	**91,0**	**87,0**	96,4	98	93,6	94,9	**95,7**	**89,6**
Admite clasificador de	48,2	74,1	48,2	52,9	74	52,9	52,5	74,3	81,3
vectores NuSVC	80,1	83,5	**86,3**	92,2	86,1	92,6	70	86,6	88,3
LinearSVC	**93,6**	87,9	**87,3**	**97,9**	**97,6**	95,3	**96,3**	94,4	89,3
Perceptrón multicapa	93	91,5	84,7	97,1	97,3	94,7	96,2	96,2	90,1
Árbol de decisión	81,3	80,4	75,2	95,6	96,2	91,3	87,7	88,5	80,6
AdaBoost	86,7	85,1	82,6	97,3	96,6	92,9	92,5	92,4	85,9
Aumento de gradiente	89,2	88,6	85,3	98,5	**98,0**	**95,7**	92,3	92,5	87,6
Clasificador de votación	93,8	92,1	87,3	98,3	98,3	95,9	96,1	96,4	90,3

Negrita indica los mejores valores de métrica de rendimiento entre los otros modelos

Tabla 7 Análisis comparativo de métricas de precisión utilizando clasificadores de aprendizaje automático

Modelos	Tendencias de noticias			Kaggle			Reuters		
	TF-IDF	CV	HV	TF-IDF	CV	HV	TF-IDF	CV	HV
Multinomial Naïve Bayes	73,3	85,8	88,4	92,3	95,8	84,4	99,1	97,3	89,3
Pasivo Agresivo	**94,5**	90,0	87,6	**98,7**	**96,9**	**97,1**	**96,6**	93,6	**91,1**
Regresión logística de descenso	94,8	**89,4**	88,1	**98,7**	97,8	**95,8**	**96,4**	94,2	90,0
de gradiente estocástico	95,4	**94,0**	**89,8**	95,4	97,9	94,2	94,7	**95,2**	89,1
Admite clasificador de	100	96,6	100	100	97,2	100	100	53,1	82,0
vectores NuSVC	94,5	96,3	90,4	97,5	97,9	91,6	41,9	77,8	85,7
LinearSVC	**96,1**	88,8	**87,8**	**98,1**	**96,5**	96,2	**93,6**	94,1	89,4
Perceptrón multicapa	93,6	93,1	84,0	97,2	96,5	96,2	92,1	94,3	89,9
Árbol de decisión	80,2	81,9	75,0	97,2	96,6	92,8	87,1	88,3	80,1
AdaBoost	89,9	88,7	83,6	97,2	96,6	94,1	92	90,6	84,9
Aumento de gradiente	91,9	92,6	89,0	98,4	**97,8**	**95,4**	90,7	90,4	89,9
Clasificador de votación	95,8	94,2	89,2	98,8	97,8	96,8	96,8	95,0	89,9

Negrita indica los mejores valores de métrica de rendimiento entre los otros modelos

Tabla 8 Análisis comparativo de la métrica de recuerdo utilizando clasificadores de aprendizaje automático

Modelos	Tendencias de noticias			Kaggle			Reuters		
	TF-IDF	CV	HV	TF-IDF	CV	HV	TF-IDF	CV	HV
Multinomial Naïve Bayes	95,9	91,5	79,7	94,6	95,4	94	77,9	85,4	85,0
Pasivo Agresivo	**92,2**	88,1	85,0	**98,0**	**98,3**	**94,8**	**96,0**	95,3	86,7
Regresión logística de descenso	91,7	**90,9**	84,9	**97,5**	96,7	**95,6**	**96,2**	95,7	89,7
de gradiente estocástico	87,7	**88**	**84,2**	97,7	98,5	93,7	95,5	**96,5**	**90,7**
Admite clasificador de	48,2	65,7	48,2	52,9	67,4	52,9	52,5	96,3	82,3
vectores NuSVC	72,4	75,9	**82,6**	88,7	80,2	94,2	99,4	95,8	91,4
LinearSVC	**91,1**	86,4	**86**	**97,8**	**98,9**	94,9	**96,6**	95,1	90,1
Perceptrón multicapa	91,9	89,9	84,1	97,2	98,2	93,8	96,4	95,8	89,8
Árbol de decisión	80,7	78,3	73,9	94,5	96,1	90,9	89,2	89,6	82,3
AdaBoost	83,5	81,8	80,9	97,5	96,8	92,6	93,6	94,5	87,8
Aumento de gradiente	86,4	85,0	82,0	98,7	98,4	96,3	94,2	95,1	89,9
Clasificador de votación	91,6	89,7	85,1	98,0	98,9	95,4	95,0	96,6	89,8

Negrita indica los mejores valores de métrica de rendimiento entre los otros modelos

Cuadro 9 Análisis comparativo de métricas específicas utilizando clasificadores de aprendizaje automático

Modelos	Tendencias de noticias			Kaggle			Reuters		
	TF-IDF	CV	HV	TF-IDF	CV	HV	TF-IDF	CV	HV
Multinomial Naïve Bayes	79,6	87,5	88	91,6	95,2	84,3	98,7	96,4	87,4
Pasivo Agresivo	**94,8**	90,5	88,2	**98,5**	**96,6**	**96,6**	**96,2**	93,1	89,6
Regresión logística de descenso	95	**90,4**	88,6	**98,5**	97,5	**95,3**	**96**	93,7	88,9
de gradiente estocástico	95,3	**94**	**89,9**	94,9	**97,4**	93,4	94,2	**94,8**	88,2
Admite clasificador de	0	94,4	0	0	95,1	0	0	65,3	80,2
vectores NuSVC	92,9	95,4	**90,2**	96,9	96,5	90,9	61,3	79,6	85,2
LinearSVC	**96,2**	89,3	**88,5**	**97,8**	**96,2**	95,7	**95,9**	93,5	88,4
Perceptrón multicapa	93,9	93,4	85,1	96,9	96,1	95,6	94,3	93,2	86,1
Árbol de decisión	81,7	82,4	76,4	96,8	96,2	91,7	86,1	87,3	78,6
AdaBoost	89,9	88,6	84,2	96,9	96,2	93,2	91,3	90,1	83,9
Aumento de gradiente	92	92,5	88,9	98,2	**97,5**	**94,8**	90,1	89,9	85,2
Clasificador de votación	95,9	94,3	89,5	98,6	97,6	96,3	96,4	94,6	88,8

Negrita indica los mejores valores de métrica de rendimiento entre los otros modelos

Tabla 10 Análisis comparativo de *F*Métrica de 1 puntuación utilizando clasificadores de aprendizaje automático

Modelos	Tendencias de noticias			Kaggle			Reuters		
	TF-IDF	CV	HV	TF-IDF	CV	HV	TF-IDF	CV	HV
Multinomial Naïve Bayes	86,9	89,4	83,6	93,4	95,5	88,9	87,2	90,9	87,0
Pasivo Agresivo	**93,4**	89,2	86,5	**98,3**	**97,5**	**95,9**	**96,2**	94,4	**88,8**
Regresión logística de descenso	93,3	**90,6**	86,7	**98,0**	97,2	95,6	**96,2**	94,9	**89,8**
de gradiente estocástico	91,3	**90,9**	**86,8**	96,5	**98,1**	93,9	95,0	95,8	**89,8**
Admite clasificador de	sesenta y cinco	77,4	sesenta y cinco	69,1	79,6	69,1	68,8	68,4	82,1
vectores NuSVC	81,3	84,5	86,2	92,8	88,1	92,8	58,9	85,8	88,4
LinearSVC	**93,5**	87,8	**87,2**	97,9	**97,3**	94,9	**95**	**94,5**	89,7
Perceptrón multicapa	92,8	91,6	84,5	97,2	97,3	94,9	93,2	94,1	88,7
Árbol de decisión	81,1	80,2	75,1	95,8	96,3	91,8	88,1	88,9	81,1
AdaBoost	86,5	85,0	82,5	97,3	96,6	93,3	92,7	92,5	86,3
Aumento de gradiente	89,1	88,5	85,3	98,5	**98,0**	**95,8**	92,4	92,6	89,9
Clasificador de votación	93,7	91,8	87,1	98,3	98,3	96	95,8	95,7	89,8

Negrita indica los mejores valores de métrica de rendimiento entre los otros modelos

El objetivo de la República de China para notar el aumento en falso aumento son las tasas positivas (FPR) con un en las tasas de verdaderos positivos (TPR) con un umbral variable de los clasificadores utilizados en este documento. El rendimiento de los modelos de clase en varios umbrales se muestra a través de gráficos en la Fig.12. La curva dibujada en el gráfico se conoce como curva de característica operativa del receptor (ROC). Las curvas de la República de China para las tendencias de noticias, Kaggle y Reuters se trazan utilizando dos parámetros como FPR y TPR como dada por Ecs. (15) y (dieciséis).

$$\text{TPR} = \frac{a}{a + C} \tag{15}$$

$$\text{FPR} = \frac{B}{b + d} \quad \text{(dieciséis)}$$

Aquí, *a B C,* y *D* representan las tasas TP, FP, FN y TN, respectivamente. predice el documento como 'Real' cuando en realidad es 'falso', conocido como tasa de falsos positivos (FP), *C* cuenta el documento predicho como "falso" cuando en realidad es "real", lo que se conoce como tasa de falsos negativos (FN), *D* cuenta el documento predicho como "Real" cuando en realidad es "Real", lo que se conoce como tasa de verdadero negativo (TN).

6.1 Principales hallazgos

Los principales hallazgos se refieren a la cuestión de si un clasificador capacitado que utilice artículos de noticias antiguos puede dar resultados precisos y eficientes para categorizar las diferencias entre contenidos falsos y reales. Se ha observado que el desempeño de la clasificación de artículos periodísticos depende del corpus y del tipo de modelo de clasificación. En nuestro experimento, se han recopilado tres corpus de tres fuentes diferentes (https://bit.ly/2zVRLxK, https://bit.ly/2Ex5VsX, https: // bit. ly / 2BmqBQE). Cada corpus se divide en entrenamiento (0,67)

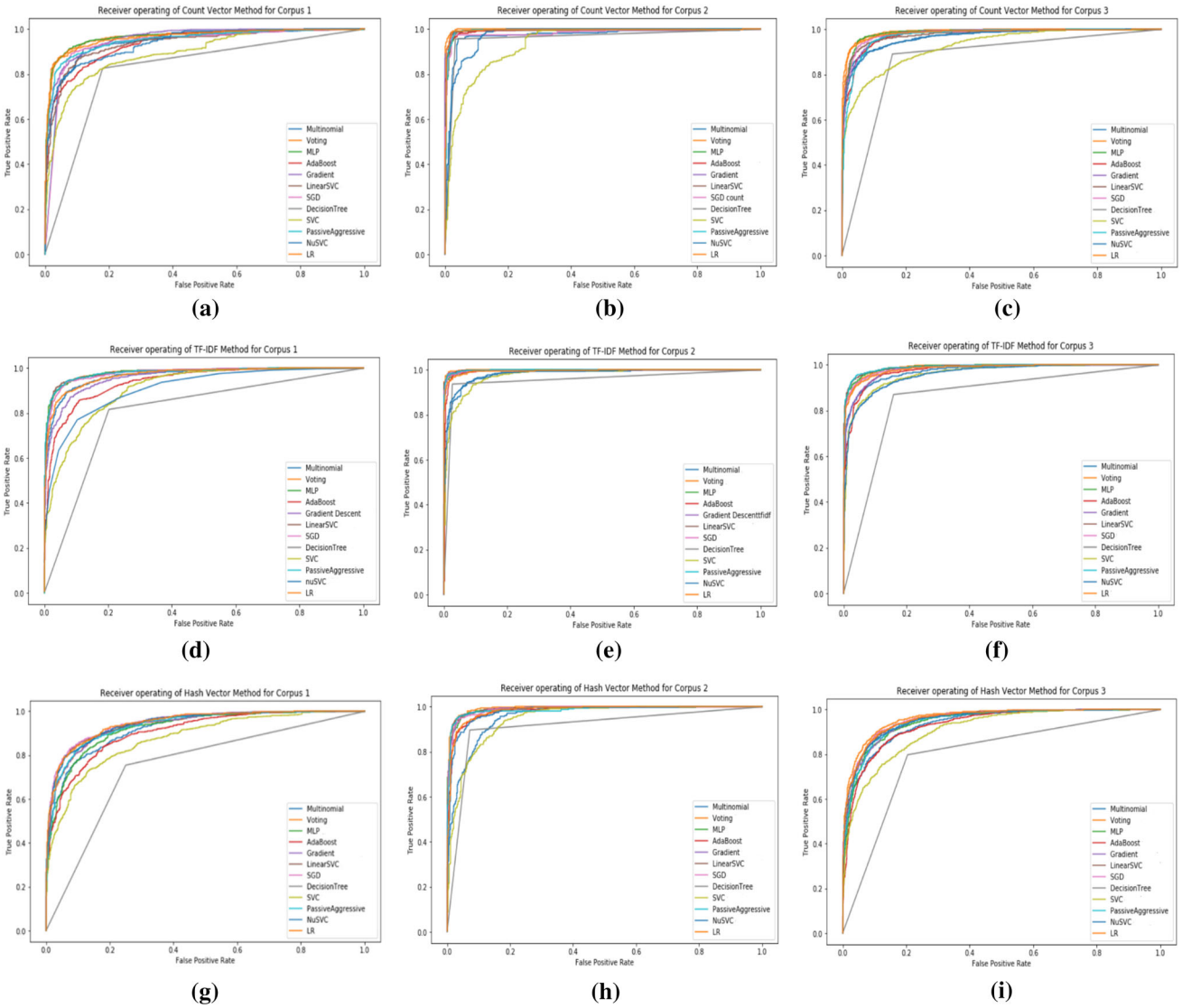

Figura 12 TPR versus FPR en diferentes umbrales de clasificación para **a** Count-Vectorizer en tendencias de noticias, **B** Count-Vectorizer en Kaggle, **C** Count-Vectorizer en Reuters, **D** TF-IDF sobre tendencias de noticias, **mi** TF-IDF activado Kaggle, **F** TF-IDF en Reuters, **gramo** Hashing-Vectorizer en tendencias de noticias, **h** Hashing-Vectorizer en Kaggle y **I** Hashing-Vectorizer en conjuntos de datos de Reuters

y conjuntos de prueba (0,33). El experimento se realizó en estos conjuntos de datos elegidos utilizando técnicas de extracción de características de frecuencia de documento inversa de término-frecuencia (TF-IDF), Count-Vectorizer (CV) y Hashing-Vectorizer (HV). Desde la perspectiva de la precisión, Passive Aggressive (93,2%) y LinearSVC (93,2%) superan a otros modelos en los tres corpus (News Trends, Kaggle, Reuters), mientras que Passive Aggressive (96%) y LinearSVC (95,9%) funcionan mejor con TF-IDF, regresión logística (94,9%) y descenso de gradiente estocástico (94,2%) funcionan mejor con CV, y SVC lineal (90,6%) y descenso de gradiente estocástico (90,5%) funcionan mejor utilizando HV individualmente para los tres cuerpos.

Un clasi fi cador se considera utilizable solo si logra tanto alta precisión como recuperación. Para promediar los resultados tanto de precisión como de recuperación, *F*Se tiene en cuenta 1 puntuación. ación. Al evaluar *F*Métrica de 1 puntaje, se observó que Pasivo Agresivo (93,3%), Descenso de gradiente estocástico (93,5%) y LinearSVC (93%) superan a otros modelos en los tres corpus de artículos de noticias utilizando técnicas de extracción de características TF-IDF, CV y HV. El pasivo agresivo (95,9%), el descenso de gradiente estocástico (95,8%) y el SVC lineal (95,4%) funcionan mejor con TF-IDF, regresión logística (94,9%), descenso de gradiente estocástico (94,2%), pasivo agresivo (93,7%), LinearSVC (93.2) se desempeña mejor con CV, y Pasivo Agresivo (90.4%), Descenso de Gradiente Estocástico (90.7%) y LinearSVC (90.6%) se desempeñan mejor usando HV. El modelo de votación multinivel propuesto supera al modelo pasivo agresivo en un 0,8%, un 0,6% y un 1,0% utilizando el enfoque TF-IDF; supera la regresión logística en un 2,6%, 0,7%, 0,8% utilizando el enfoque de CV;

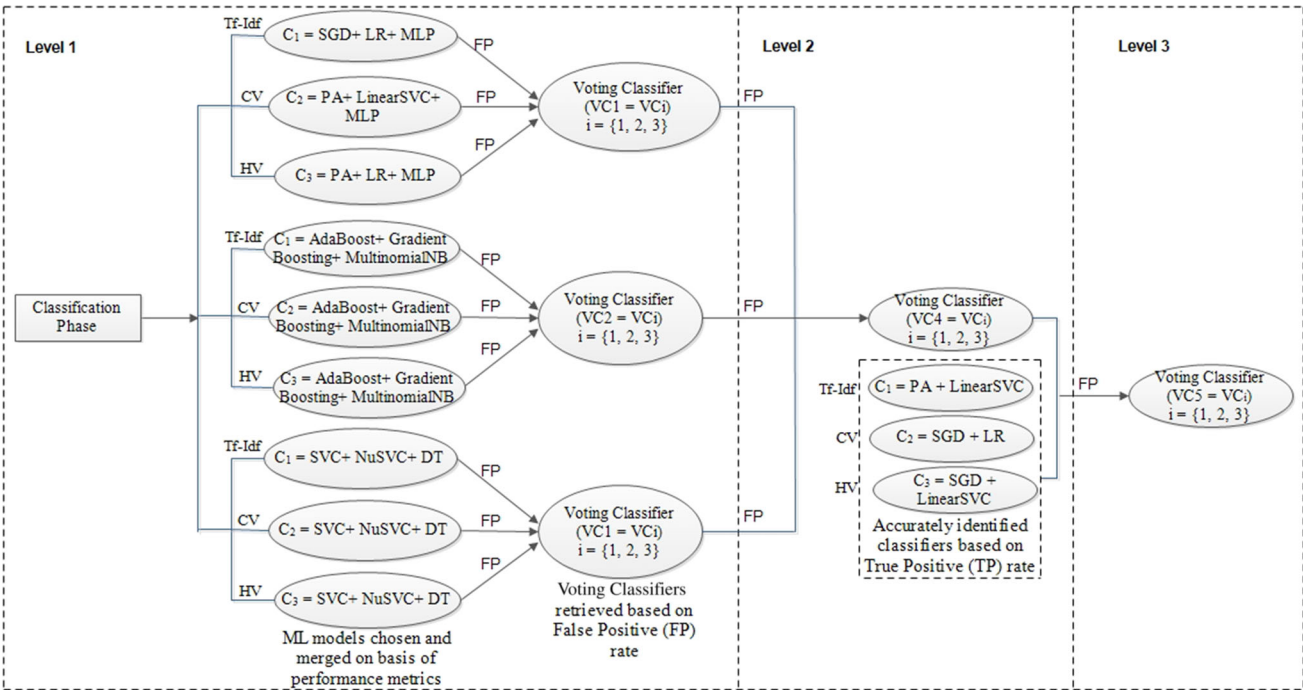

Figura 13 Arquitectura del modelo de votación multinivel propuesto

supera al LinearSVC en 0.0%, 0.5%, 0.9% usando el enfoque HV en News Trends, Kaggle y Reuters corpus, respectivamente.

Para evaluar el rendimiento predictivo de nuestro enfoque, se traza ROC_AUC. Pasivo Agresivo, Descenso de gradiente estocástico y LinearSVC, Impulso de gradiente, Regresión logística superan a otros modelos ML sobre la base de la métrica ROC_AUC con TPR> 0.97. Pasivo Agresivo, Descenso de gradiente estocástico y LinearSVC da TPR> 0.97 usando TF-IDF, Regresión logística da TPR> 0.98 usando CV, y Descenso de gradiente estocástico, LinearSVC, Regresión logística da TPR> 0.95 para tendencias de noticias, conjuntos de datos de Kaggle y Reuters, respectivamente. Con base en el tiempo de entrenamiento requerido por varios clasificadores de ML, se ha observado que existe una compensación entre eficiencia y precisión. El tiempo de entrenamiento requerido por HV es menor que el TF-IDF y la técnica CV, pero compromete la precisión métrica. Se ha analizado que la técnica de hash es útil cuando el objetivo es lograr una alta eficiencia en un gran conjunto de datos. Los dos clasificadores de ML, como Regresión logística y LinearSVC, se eligen entre otros modelos que dan como resultado una alta precisión y eficiencia para superar el problema de la compensación.

7 Modelo de votación propuesto a varios niveles

El modelo de votación multinivel propuesto no solo ayuda a mejorar la precisión, sino que también ayuda a reducir el tiempo de formación de los clasificadores. La reducción del tiempo de formación ayuda para aumentar la eficiencia de nuestro modelo mediante la introducción de métodos paralelos donde los aprendices básicos se generan en paralelo. La motivación detrás del modelo propuesto es analizar la independencia entre los alumnos de base. Se proponen tres niveles para realizar el experimento como se comenta a continuación.

Nivel 1 Se combinan conjuntos de clasificadores de tres ML en función de su métrica de rendimiento (tasa de FP) para aplicar el Clasificador de votación. Se recuperan los modelos de votación (VC1, VC2, VC3).

Nivel 2 Se recupera un clasificador de votación (VC4) después de fusionar los tres modelos (VC1, VC2, VC3) sobre la base de su tasa de falsos positivos.

Nivel 3 Las predicciones falsas de Voting Classi fier (VC4) se fusionan con PA y LinearSVC para TF-IDF, LR y SGD para CV y SGD, LinearSVC para HV para obtener la predicción final.

Sobre la base de la tasa mínima de falsos positivos (FP), los modelos ML se fusionan para superar la debilidad de los modelos individuales existentes. Cuanto más mínimo sea el ratio FP, más preciso será el modelo para predecir el contenido como falso. Se utilizan tres técnicas de extracción de características (TF-IDF, CV y HV) para extraer las características de un conjunto de datos recopilados. Sobre la base de la relación FP, los modelos se seleccionan y combinan para dar una predicción adecuada. Primer grupo en el nivel 1, SGD (Tendencias de noticias), LR (Kaggle) y MLP (Reuters) utilizando TFIDF; SGD (Tendencias de noticias), LinearSVC (Kaggle) y MLP (Reuters) utilizando CV; PA (News Trends), LR (Kaggle) y MLP (Reuters) que utilizan HV se fusionan para construir el Voting Classi fier (VC1). Segundo clúster en el nivel 1, AdaBoost, Gradient Boosting y MultinomialNB (News Trends, Kaggle

Cuadro 11 Análisis comparativo del modelo de votación multinivel propuesto con el clasificador de votación

Modelos	Tendencias de noticias			Kaggle			Reuters		
	TF-IDF	CV	HV	TF-IDF	CV	HV	TF-IDF	CV	HV
Precisión									
Clasificador de votaciones	93,8	92,1	87,3	98,3	98,3	95,9	96,1	96,4	90,3
Clasificador de votación múltiple	94,3	93,6	87,1	98,9	98,7	95,8	97,2	96,5	90,2
Precisión									
Clasificador de votación	95,8	94,2	89,2	98,8	97,8	96,8	96,8	95	89,9
Clasificador de votación múltiple	96,4	94,3	89,6	99,1	98,3	96,8	98,4	97,6	90,4
Recordar									
Clasificador de votación	91,6	89,7	85,1	98	98,9	95,4	95	96,6	89,8
Clasificador de votación múltiple	93,1	91,4	85,2	98,7	98,8	94,4	96,8	95,2	90,8
Ciudad específica									
Clasificador de votación	95,9	94,3	89,5	98,6	97,6	96,3	96,4	94,6	88,8
Clasificador de votación múltiple	96	92,4	86,1	98,2	97,1	93,6	95,7	94,6	87,9
*F*1 puntuación									
Clasificador de votación	93,7	91,8	87,1	98,3	98,3	96	95,8	95,7	89,8
Clasificador de votación múltiple	94,7	92,8	87,3	98,8	98,7	95,5	97,7	97	90,5

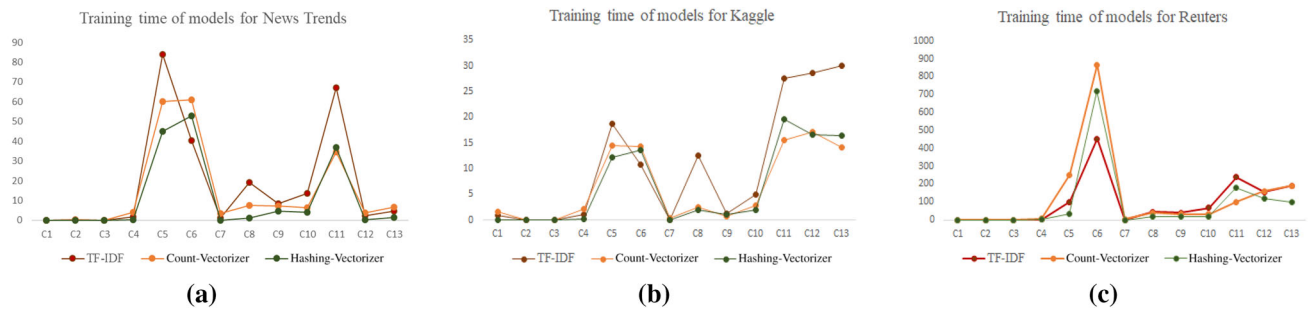

Figura 14 Comparación de tiempo de formación para **a** Tendencias de noticias, **B** Kaggle y **C** Conjunto de datos de Reuters sobre la base de TF-IDF, Hashing-Vectorizer y Contar-Técnicas de extracción de características del vectorizador

y Reuters) que utilizan TF-IDF, CV y HV se fusionan para construir VC2. El tercer grupo en el nivel 1, SVC, NuSVC y DT (News Trends, Kaggle y Reuters) que utilizan TF-IDF, CV y HV se fusionan para construir VC3 como se muestra en la Fig.13.

Basado en la proporción de FP de tres clasificadores de votación propuestos (VC1, VC2 y VC3), se recupera un cuarto clasificador de votación (VC4) en el nivel 2 basado en las tasas de FP de los clasificadores VC1, VC2 y VC3. En el nivel 3, PA, LinearSVC usando TF-IDF; SGD y LR usando CV; SGD y LinearSVC que usan HV se recuperan en función de la tasa de TP y se agrupan aún más con VC4. Sobre la base de la tasa de FP, se recupera VC5 para dar la predicción final del modelo propuesto.

8 Evaluación del desempeño del modelo de votación multinivel

En el clasificador de votación multinivel propuesto, los tres mejores modelos de ML se combinan de cada técnica de extracción de características. como se discutió en la sección anterior. Se puede observar en la tabla11 que el modelo propuesto supera al clasificador de votación en un 0,73%, 0,66% y 0,13% utilizando TF-IDF, CV y HV, respectivamente, en términos de métrica de precisión. De manera similar, el modelo propuesto también ofrece una mejora significativa en la precisión, recuperación, especificidad y *F*Medidas de desempeño de 1 puntaje.

Para comparar los métodos de vectorización utilizados en términos de tiempo de entrenamiento requerido para entrenar los datos, la Fig. 14 describe que la técnica HV es más eficiente que otros métodos de extracción de características, ya que el tiempo requerido para entrenar el modelo de votación multinivel propuesto es mínimo para los conjuntos de datos de News Trends y Reuters. Para lograr una mejor eficiencia (menos tiempo de entrenamiento), el método TF-IDF solo es adecuado cuando los datos no son demasiado grandes, mientras que el hash se puede usar para grandes conjuntos de datos cuando es necesario hacer un compromiso entre eficiencia y precisión. Después de realizar nuestro experimento, en la siguiente sección se extraen algunas conclusiones que ayudan a analizar el mejor clasificador a elegir por su alta eficiencia y precisión.

Cuadro 12 Comparación de los modelos existentes con un sistema propuesto para el corpus de Reuters

Autores	Enfoque propuesto	Características	Precisión del modelo
Stein y Zu Eissen (2008)	Presentó un enfoque de clasificación bayesiano que utiliza características específicas de clase para la clasificación automática de texto.	Basado en temas y basado en documentos modelado	LapPLSI: 74,6%
Mishu y Ra fi uddin (2016)	Documento de texto clasificado usando varios clasificadores	Frecuencia de documentos	MultinomialNB: 72% LR: 73,5% SGD: 76% SVC: 78% SVC lineal: 83,3% Votación: 89%
Análisis basado en modelos de AA individuales	Artículos clasificados como falsos o reales usando varios clasificadores	TF-IDF	SGD: 96,2% AP: 96,2% LinearSVC: 96,3% MLP: 96,2% AdaBoost: 92,5% Votación: 96,4%
		Vectorizador de conteo	MLP: 96,2% DT: 88,5% LR: 95,7% MultinomialNB: 89,9% Aumento de gradiente: 92,5% de SVC: 81,3%
		Vectorizador de hash	NuSVC: 88,3%
Propuesta de varios niveles modelo de votacion	Se propone un sistema de detección de noticias falsas para lograr alta precisión y alta eficiencia	TF-IDF	Votación en varios niveles: 97,2%
		Vectorizador de conteo	Votación multinivel: 96,5%
		Vectorizador de hash	Voto multinivel: 90,2%

9 Comparación con las obras existentes

Los resultados dados por nuestro sistema también se comparan con otros trabajos existentes sobre detección de noticias falsas como se muestra en las Tablas. 12, 13 y 14 utilizando corpus de Reuters, Kaggle y News Trends, respectivamente. Los investigadores han utilizado enfoques de aprendizaje automático para realizar la detección de noticias falsas en varias plataformas de redes sociales. Se ha analizado desde la tabla12 que la técnica de extracción de características basada en la frecuencia de documentos utilizada por Mishu y Ra fi uddin (2016) para MultinomialNB (72%), SVC (78%) y VotingClassi fier (89%) (Mishu y Ra fi uddin 2016) no ofrece una mejor precisión que nuestro sistema propuesto para los tres clasificadores. En nuestro sistema propuesto, MultinomialNB (89,9%), SVC (81,3%) y Voting (96,4%) superan a Mishu y Ra fi uddin (2016) modelos.

Se ha observado en la tabla 13 que la técnica de extracción de características basada en TF-IDF utilizada por Ahmed et al. (2017) para clasificadores de ML no ofrece una mayor precisión que nuestro sistema propuesto. La observación registrada muestra que nuestro

El modelo de votación propuesto en varios niveles (97,2%) supera al aumento de gradiente en un 0,4% con TF-IDF, el LR en un 2,3% con CV y NuSVC en un 3,6% con técnicas de extracción de características de alto voltaje, respectivamente, para el corpus de Kaggle. De la tabla14, se puede analizar que nuestro modelo de votación multinivel propuesto supera a PA en 0.8% usando TF-IDF, MultinomialNB en 4.3% usando CV y NuSVC en 0.8% usando técnicas de extracción de características de HV, respectivamente, para el corpus de Tendencias de Noticias. A partir del análisis comparativo, se ha analizado que nuestro modelo de votación multinivel propuesto que utiliza las tres técnicas de extracción de características supera el rendimiento en comparación con las métricas de rendimiento individuales de los modelos ML utilizados por Ahmed et al. (2017) y Mishu y Ra fi uddin (2016) para desarrollar un sistema de clasificación automática de las características de los artículos de noticias para etiquetarlos como artículos de noticias falsos o reales. Se ha observado en la tabla12
que el modelo propuesto supera al modelo PA en un 0,9% utilizando TF-IDF, el modelo LR en un 0,8% utilizando CV y el modelo NuSVC por 1,9% utilizando técnicas de extracción de características HV, respectivamente, en comparación con sus métricas de rendimiento.

Cuadro 13 Comparación de los modelos existentes con un sistema propuesto para el corpus de Kaggle

Autores	Enfoque propuesto	Características	Precisión del modelo
Ahmed y col. (2017)	Propuso un modelo de detección de noticias falsas utilizando análisis de n-gramas y técnicas de aprendizaje automático	TF y TF-IDF basados en n-gram	LSVM: 92%
			KNN: 83,1%
			SVM: 86%
			DT: 89%
			SGD: 89%
			LR: 89%
Análisis basado en modelos de AA individuales	Artículos clasificados como falsos o reales usando varios clasificadores	Vectorizador de conteo	MultinomialNB—93,2%
			SVC: 52,9%
			LR: 96,4%
			MLP: 97,1%
			DT: 95,6%
			Votación: 98,3%
		TF-IDF	LinearSVC: 97,9%
			SGD: 98%
			AP: 98,3%
			AdaBoost: 97,3%
			Degradado Impulso: 98,5%
			Votación: 98,3%
		Vectorizador de hash	NuSVC: 92,2%
Propuesta de varios niveles modelo de votacion	Se propone un sistema de detección de noticias falsas para lograr una alta precisión y alta eficiencia.	TF-IDF	Multi nivel votación: 98,9%
		Vectorizador de conteo	Multi nivel votación: 98,7%
		Vectorizador de hash	Multi nivel votación: 95,8%

10 Conclusión y alcance futuro

El objetivo de este trabajo es analizar la técnica supervisada más conocida para la detección de noticias falsas. Cuando se trata de clasificadores de aprendizaje automático, la cuestión clave no es encontrar un clasificador de aprendizaje superior a otros, sino más bien encontrar las condiciones bajo las cuales un modelo particular supera a otros para un problema dado. El conjunto de atributos extraídos del corpus tomado utiliza tres técnicas de extracción de características (TF-IDF, CV y HV) para alimentar los vectores de características extraídos en los modelos de aprendizaje automático seleccionados. Algunas características tomadas de los conjuntos de datos para la tarea de aprendizaje son los atributos categóricos, los valores perdidos, los titulares del artículo, el cuerpo del artículo y el nombre del editor. Varios clasificadores, Multinomial Naïve Bayes (MultinomialNB), Pasivo Agresivo (PA), Descenso de gradiente estocástico (SGD), Regresión logística (LR), sion Tree (DT), AdaBoost, Gradient Boosting y Voting Classi fier se analizaron sobre la base de medidas de rendimiento. Después de analizar los clasificadores, la atención se centró en utilizar las fortalezas de un modelo para complementar las debilidades de otro. Por lo tanto, se propuso el modelo de votación multinivel, que integra varios modelos ML basados en sus tasas de FP para recuperar un clasificador de votación de noticias para obtener un mejor análisis de predicción. El modelo desarrollado ayuda a resolver el problema del equilibrio entre precisión y eficiencia.

En el futuro, se creará una GUI basada en la web para el sistema de detección de noticias falsas propuesto para clasificar las noticias como falsas o reales en plataformas de redes sociales en tiempo real como Facebook, Instagram, Twitter y WhatsApp. Además, el conjunto de datos anotado en forma de imágenes (con contenido textual escrito en ellas) se recopilará y mantendrá desde las plataformas de Facebook y Reddit. El conjunto de datos anotado se puede utilizar para detectar imágenes falsas en el futuro, ya que no hay ningún conjunto de datos disponible en la actualidad. El sistema propuesto tiene el potencial

Cuadro 14 Comparación de los modelos existentes con un sistema propuesto para el corpus de Tendencias de Noticias

autores	Enfoque propuesto	Características	Precisión del modelo
Kuleshov y col. (2018)	El autor muestra la existencia de ejemplos contradictorios en la clasificación del lenguaje natural.	n-gramo	NB: 93%
Análisis basado en modelos de AA individuales	Artículos clasificados como falsos o reales usando varios clasificadores	TF-IDF	PA: 93,5%
			SGD: 93,4%
			SVC lineal: 93,6%
			MLP: 93%
			DT: 81,3%
			AdaBoost-86,7%
			Degradado Impulso: 89,2%
			Votación: 93,8%
			LR: 91,4%
		Vectorizador de conteo	MultinomialNB—89,3%
			SVC: 74,1%
		Hashing – Vectorizador	NuSVC: 86,3%
Propuesta de varios niveles modelo de votacion	Se propone un sistema de detección de noticias falsas para lograr una alta precisión y alta eficiencia.	TF-IDF	Multi nivel votación: 94,3%
		Vectorizador de conteo	Multi nivel votación: 93,6%
		Hashing – Vectorizador	Multi nivel votación: 87,1%

para dar impulso a diversas aplicaciones emergentes, como el control de la difusión de noticias falsas durante las elecciones, el terrorismo, las calamidades naturales, los delitos para el mejoramiento de la sociedad.

Agradecimientos Esta publicación es el resultado del trabajo de I + D realizado en el proyecto en el marco del plan de doctorado Visvesvaraya del Ministerio de Electrónica y Tecnología de la Información del Gobierno de la India, que está siendo implementado por Digital India Corporation (anteriormente Media Lab Asia).

Fondos La financiación fue proporcionada por Digital India Corporation (anteriormente Media Lab Asia) (Grant No. U72900MH2001NPL133410).

Cumplimiento de estándares éticos

Conflicto de interés Los autores declaran que no tienen intereses financieros en competencia o relaciones personales conocidas que pudieran haber influido en el trabajo informado en este artículo.

Referencias

Aggarwal A, Rajadesingan A, Kumaraguru P (2012) PhishAri: autodetección de phishing en tiempo real matic en twitter. En: Cumbre de investigadores de eCrime (eCrime). IEEE, págs. 1–12

Aggarwal A, Kumar S, BhargavaK, Kumaraguru P (2018) El seguidor falacia de conteo: detectar usuarios de Twitter con conteo de seguidores manipulado

Ahmed F, Abulaish M (2012) Un enfoque basado en MCL para la prodetección de archivos en redes sociales online. En: IEEE 11th conferencia internacional sobre confianza, seguridad y privacidad en la informática y las comunicaciones (TrustCom). IEEE, págs. 602–608

Ahmed H, Traore I, Saad S (2017) Detección de noticias falsas en línea mediante n-gramanálisis y técnicas de aprendizaje automático. En: Conferencia internacional sobre sistemas inteligentes, seguros y confiables en entornos distribuidos y en la nube. Springer, pp 127-138 Alahmadi A, Joorabchi A, Mahdi AE (2013) Una nueva representación de texto esquema que combina enfoques de bolsa de palabras y bolsa de conceptos para la clasificación automática de texto. En: 2013 7ma conferencia y exposición IEEE GCC (GCC). IEEE, págs. 108–113

Batchelor O (2017) Sacar la verdad: el papel de las bibliotecas en la lucha contra las noticias falsas. Ref Serv Rev. 45 (2): 143

Benevenuto F, Rodrigues T, AlmeidaV, Almeida J, GonçalvesM (2009) Detección de spammers y promotores de contenido en redes sociales de video online. En: Actas de la 32ª conferencia internacional ACM SIGIR sobre investigación y desarrollo en la recuperación de información. ACM, págs. 620–627

Benevenuto F, Magno G, Rodrigues T, Almeida V (2010) Detectando spammers en twitter. En: Conferencia de colaboración, mensajería electrónica, antiabuso y spam (CEAS), vol 6, p 12

Caetano JA, de Oliveira JF, Lima HS, Marques-Neto HT, Magno G, Meira W Jr, Almeida VA (2018) Analizando y caracterizando discusiones políticas en grupos públicos de WhatsApp. preimpresión arXiv arXiv: 1804.00397

Canini KR, Suh B, Pirolli PL (2011) Encontrar información creíble fuentes en redes sociales basadas en contenido y estructura social. En: Tercera conferencia internacional de IEEE sobre informática social (SocialCom). Tercera conferencia internacional IEEE sobre privacidad, seguridad, riesgo y confianza (PASSAT). IEEE, págs. 1–8

Chen Y, Conroy NJ, Rubin VL (2015) Contenido en línea engañoso: rec-
Reconocer clickbait como noticias falsas. En: Actas de la ACM de 2015 sobre el taller sobre detección de engaños multimodal. ACM, págs. 15–19 Chhabra S, Aggarwal A, Benevenuto F, Kumaraguru P (2011) Phi.sh $ ocial: el panorama del phishing a través de URL cortas. En: Actas de la 8ª conferencia anual de colaboración, mensajería electrónica, anti-abuso y spam. ACM, págs. 92–101
Conroy NJ, Rubin VL, Chen Y (2015) Detección automática de engaños:
métodos para encontrar noticias falsas. Proc Assoc Inf Sci Technol 52 (1): 1
D'Angelo G, Palmieri F, Rampone S (2019) Detectando recomendaciones desleales
mendaciones en entornos generalizados basados en la confianza. Inf Sci 486: 31 Dewan P, Kumaraguru P (2015) Towards automatic real time identi fi
cación de publicaciones maliciosas en Facebook. En: 13º Congreso anual sobre privacidad, seguridad y confianza (PST). IEEE, págs. 85–92
DewanP, KumaraguruP (2017) Inspector de Facebook (FbI): hacia la
matic detección en tiempo real de contenido malicioso en Facebook. Soc Netw Anal Min 7 (1): 15
Dewan P, Gupta M, Goyal K, Kumaraguru P (2013) Multiosn: tiempo real
Monitoreo de eventos del mundo real en múltiples redes sociales en línea. En: Actas del quinto taller de intercambio de investigación académica colaborativa de IBM. ACM, pág.6
Noticias falsas en whatsapp. http://bit.ly/2miuv9j. Consultado por última vez el 27 de agosto de 2019.

Gao H, Hu J, Wilson C, Li Z, Chen Y, Zhao BY (2010) Detectando y
caracterizando las campañas de spam social. En: Actas de la décima conferencia ACM SIGCOMM sobre medición de Internet. ACM, págs. 35–47
GarimellaK, TysonG (2018) WhatsApp, doc? Un primer vistazo aWhatsApp
datos de grupos públicos. preimpresión arXivarXiv: 1804.01473
Gupta A, Kumaraguru P (2012a) Ranking de credibilidad de los tweets durante
eventos de alto impacto. En: Actas del 1er taller sobre privacidad y seguridad en las redes sociales en línea. ACM, pág. 2
GuptaA, KumaraguruP (2012b) Twitter explota con actividad en Mom-
bai explosiones! ¿Un salvavidas o un demonio no supervisado al acecho? Reporte técnico
Gupta A, Lamba H, Kumaraguru P (2013a) $ 1.00 por rt # Boston-
Maratón #PrayForBoston: análisis de contenido falso en Twitter. En: Cumbre de investigadores de eCrime (eCRS). IEEE, págs. 1–12
Gupta A, Lamba H, Kumaraguru P, Joshi A (2013b) Arena falsa:
caracterizar e identificar imágenes falsas en twitter durante el huracán sandy. En: Actas de la 22ª conferencia internacional sobre world wide web. ACM, págs. 729–736
Jain P, Kumaraguru P (2016) Sobre la dinámica del cambio de nombre de usuario
comportamiento en twitter. En: Actas de la 3ª conferencia IKDD sobre ciencia de datos. ACM, pág.6
Base de datos de Kaggle. https://bit.ly/2BmqBQE. Consultado por última vez el 22 de octubre de 2017 en la base de datos de Kaggle.https://bit.ly/2Ex5VsX. Último acceso: 24 de octubre de 2017 LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Naturaleza
521 (7553): 436
Kuleshov V, Thakoor S, Lau T, Ermon S (2018) Ejemplos de adversarios
para problemas de clasi fi cación del lenguaje natural MagdyA, WanasN (2010)
documentos. En: Actas del 2º taller internacional sobre búsqueda y minería de contenidos generados por los usuarios. ACM, págs. 103–110
Markines B, Cattuto C, Menczer F (2009) Detección de spam social. En
Actas del 5º taller internacional sobre recuperación de información contradictoria en la web. ACM, págs. 41–48
Mishu SZ, Rafi uddin S (2016) Análisis de desempeño de supervisados
algoritmos de aprendizaje automático para la clasificación de texto. En: XIX Congreso internacional de informática y tecnologías de la información (ICCIT). IEEE, págs. 409–413
Base de datos de tendencias de noticias. https://bit.ly/2zVRLxK. Último acceso: 18 de octubre de 2017.

Pontes T, Magno T, Vasconcelos M, Gupta A, Almeida J, Kumaraguru
P, Almeida V (2012a) Cuidado con lo que compartes: inferir la ubicación de la casa en las redes sociales. En: IEEE 12th international conference on data mining workshops (ICDMW). IEEE, págs. 571–578 PontesT,
VasconcelosM, Almeida J, KumaraguruP, AlmeidaV (2012b)
Sabemos dónde vives: caracterización de la privacidad del comportamiento cuadrangular. En: Actas de la conferencia ACM de 2012 sobre computación ubicua. ACM, págs. 898–905
Qazvinian V, Rosengren E, Radev DR, Mei Q (2011) Se rumorea que:
Identificar información errónea en microblogs. En: Actas de la conferencia sobre métodos empíricos en el procesamiento del lenguaje natural. Association for Computational Linguistics, págs. 1589-1599
Rubin VL, Conroy NJ, Chen Y (2015) Towards news veri fi cation:
métodos de detección de engaños para el discurso periodístico. En: conferencia internacional de Hawái sobre ciencias de sistemas
Rubin V, Conroy N, Chen Y, Cornwell S (2016) ¿Noticias falsas o verdad?
Usar señales satíricas para detectar noticias potencialmente engañosas. En: Actas del segundo taller sobre enfoques computacionales para la detección de engaños, págs. 7–17
Ruchansky N, Seo S, Liu Y (2017) CSI: un modelo profundo híbrido para falsificaciones
detección de noticias. En: Actas de la ACM 2017 sobre la conferencia sobre gestión de la información y el conocimiento. ACM, págs. 797–806
Sen I, Aggarwal A, Mian S, Singh S, Kumaraguru P, Datta A (2018)
Vale su peso en Me gusta: para detectar Me gusta falsos en Instagram. En: Actas de la 10thACMconference onweb science. ACM, págs. 205–209
Shu K, Sliva A, Wang S, Tang J, Liu H (2017) Detección de noticias falsas en
redes sociales: una perspectiva de minería de datos. ACM SIGKDD Explor Newsl 19 (1): 22
Sirajudeen SM, Azmi NFA, Abubakar AI (2017) Noticias falsas en línea
algoritmo de detección. J Theor Appl Inf Technol 95 (17): 4114
Stein B, Zu Eissen SM (2008) Modelos de recuperación para la clasificación de género.
Scand J Inf Syst 20 (1): 3
Volkova S, Shaffer K, Jang JY, Hodas N (2017) Separando hechos de
ficción: modelos lingüísticos para clasificar publicaciones de noticias sospechosas y confiables en twitter. En Actas de la 55ª reunión anual de la asociación de lingüística computacional (volumen 2, artículos breves), vol 2, págs. 647–653
Wang AH (2010) No me sigas: detección de spam en twitter. En: Pro-
cedidos de la conferencia internacional de 2010 sobre seguridad y criptografía (SECRYPT). IEEE, págs. 1–10
Wei W, Wan X (2017) Aprendiendo a identificar ambiguos y engañosos
titulares de las noticias. preimpresión arXiv arXiv1705.06031
Weimer M, Gurevych I, Mühlhäuser M (2007) Evaluación automática
la calidad de las publicaciones en los debates en línea sobre software. En: Actas de la 45ª reunión anual de la ACL sobre carteles interactivos y sesiones de demostración. Association for Computational Linguistics, págs. 125-128