Universidad Politécnica
de Madrid

Escuela Técnica Superior de Ingenieros Informáticos

Grado en Ciencia de Datos e Inteligencia Artificial

Trabajo Fin de Grado

Detección Automática de Noticias Falsas en Redes Sociales Usando Modelos de Procesamiento de Lenguaje Natural

Autor: Álvaro Hernández Rodríguez
Tutor: Antonio Jesús Díaz Honrubia

Madrid, junio 2025

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado
Grado en Ciencia de Datos e Inteligencia Artificial

Título: Detección Automática de Noticias Falsas en Redes Sociales Usando Modelos de Procesamiento de Lenguaje Natural

junio 2025

Autor: Álvaro Hernández Rodríguez
Tutor: Antonio Jesús Díaz Honrubia
Departamento de Lenguajes y Sistemas Informáticos e Ingeniería de Software
Escuela Técnica Superior de Ingenieros Informáticos
Universidad Politécnica de Madrid

Resumen

En la actualidad, las redes sociales y los entornos digitales se han convertido en las principales fuentes de información para millones de personas. Sin embargo, esta transformación ha traído consigo un fenómeno preocupante: la rápida difusión de noticias falsas o engañosas. La desinformación no solo afecta al ámbito político o mediático, sino que tiene consecuencias reales en la sociedad, desde el aumento de la polarización ideológica hasta la pérdida de confianza en instituciones públicas o la propagación de teorías peligrosas durante emergencias sanitarias. La facilidad con la que estos contenidos se comparten, la dificultad de verificación para el usuario medio y el uso creciente de tecnologías generativas hacen que la detección automática de estas noticias sea un reto urgente tanto para investigadores como para responsables de plataformas y gobiernos.

Ante esta problemática, el objetivo principal de este proyecto es diseñar, implementar y evaluar un sistema de detección automática de noticias falsas en español utilizando técnicas de procesamiento de lenguaje natural (PLN) e inteligencia artificial. El proyecto busca aportar una solución eficaz, escalable y adaptable a diferentes contextos de desinformación, contribuyendo a un entorno informativo más fiable. Para ello, se han utilizado distintos conjuntos de datos (noticias reales y falsas, tweets generados, y una combinación de ambos), y se han probado diversos métodos de representación del texto junto a modelos de clasificación de distintos niveles de complejidad, desde algoritmos clásicos hasta modelos avanzados basados en Transformers.

Los resultados obtenidos muestran que el enfoque basado en modelos de machine learning, deep learning y transformers presentan un gran potencial para afrontar esta tarea. Se ha comprobado también que el uso de embeddings como FastText mejora la precisión en modelos más sencillos cuando se trabaja con textos breves como los tweets. Además, se ha llevado a cabo un análisis de explicabilidad para interpretar mejor las predicciones de los modelos, así como una evaluación del impacto potencial del sistema desarrollado en términos sociales, económicos, medioambientales y culturales.

Finalmente, este proyecto pone de relieve la importancia de seguir investigando en este ámbito, proponiendo líneas de trabajo futuro como la integración de análisis multimodal (texto e imagen), el uso de técnicas de detección adaptativa frente a nuevos patrones de desinformación, y la incorporación de principios éticos que garanticen la equidad y la transparencia.

Abstract

Nowadays, social media and digital environments have become the main sources of information for millions of people. However, this shift has brought with it a troubling phenomenon: the rapid spread of false or misleading news. Disinformation not only affects the political or media landscape, but also has real consequences for society, ranging from increased ideological polarization to the erosion of trust in public institutions, or the spread of harmful theories during health emergencies. The ease with which such content is shared, the difficulty of verification for the average user, and the growing use of generative technologies make the automatic detection of fake news an urgent challenge for researchers, platform managers, and governments alike.

To address this issue, the main objective of this project is to design, implement, and evaluate an automatic fake news detection system in Spanish using Natural Language Processing (NLP) and artificial intelligence techniques. The project aims to provide an effective, scalable, and adaptable solution for different disinformation contexts, contributing to a more reliable information ecosystem. For this purpose, various datasets have been used (real and fake news, synthetically generated tweets, and a combination of both), and multiple text representation methods have been tested along with classification models of different complexity levels, from classical algorithms to advanced Transformer-based models.

The results obtained show that machine learning, deep learning, and Transformer-based approaches hold great potential for addressing this task. It has also been confirmed that using embeddings such as FastText improves accuracy in simpler models when dealing with short texts like tweets. In addition, an explainability analysis has been conducted to better understand the models' predictions, along with an evaluation of the system's potential impact across social, economic, environmental, and cultural dimensions.

Finally, this project highlights the importance of continued research in this area, proposing future lines of work such as the integration of multimodal analysis (text and image), the use of adaptive detection techniques against new misinformation patterns, and the incorporation of ethical principles that ensure fairness and transparency.

Tabla de contenidos

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

Vivimos en un mundo donde la información viaja a la velocidad de un simple clic. Plataformas como Twitter, Facebook o TikTok son mucho más que simples lugares donde poder compartir memes o videos virales: se han convertido en las principales fuentes de noticias para millones de personas gracias a su inmediatez y fácil acceso. Este acceso inmediato al conocimiento debería de ser una herramienta poderosa para construir una sociedad más informada, pero hay un problema que no podemos pasar por alto: las noticias falsas, o *fake news*, están por todas partes. Además, la facilidad con la que cualquier usuario puede publicar contenido, sin necesidad de cumplir con una verificación o filtros, ha democratizado la información, pero también la desinformación. Hoy en día, cualquier persona con conexión a internet puede compartir una noticia falsa, ya sea por error, por ideología o incluso con la idea de perjudicar a la sociedad. Más allá de esto, lo preocupante es que, una vez publicada, esa información puede recorrer el mundo en segundos, sin importar si es cierta o no [2].

Esto hace que las redes sociales se hayan convertido en un entorno perfecto para la desinformación: la combinación del alcance masivo de las publicaciones, su inmediatez y los algoritmos diseñados para maximizar la interacción hacen que las noticias falsas no desaparezcan con el tiempo [3]. Lo que importa no es la veracidad del contenido, sino su capacidad para generar clics, compartidos y reacciones. Esto nos obliga, como usuarios y como sociedad, a ser más críticos y a buscar soluciones que nos ayuden a frenar su impacto.

Los algoritmos de estas plataformas están diseñados para retener durante el mayor tiempo posible a sus usuarios. Se alimentan de su comportamiento y manera de interactuar con el resto de usuarios para ofrecer más contenido que lo mantenga dentro de la aplicación el mayor tiempo posible y dentro de esa lógica, lo sensacional, lo emocional o lo polémico siempre es lo que termina ganando. Las *fake news*, con sus titulares exagerados y afirmaciones sin matices sólidos, encajan perfectamente con esta lógica algorítmica [4][5]. El resultado es un círculo vicioso en el que los contenidos más virales, que no siempre son los reales, obtienen mayor visibilidad, favoreciendo su difusión y reforzando creencias erróneas. Por esto, detectar las noticias falsas no es solo un reto tecnológico, sino una necesidad urgente para proteger la calidad de la información [6].

A raíz de esto, nos encontramos con que las noticias falsas se difunden más rápido y más ampliamente que las verdaderas, especialmente cuando buscan la emoción del usuario [4]. Un simple vídeo viral o una historia compartida en un grupo de Facebook puede alcanzar a miles o incluso millones de personas antes de que nadie tenga tiempo de verificar si lo que se dice es cierto. En los tiempos que corren, estamos prácticamente acostumbrados a que cada cierto tiempo o después de algún acontecimiento relevante a nivel nacional o mundial, se extiendan infinidad de bulos por nuestras redes sociales. A esto se le suma otro fenómeno preocupante: muchas personas, aun teniendo acceso a fuentes fiables, tienden a creer y compartir información falsa si esta es acorde con sus creencias [5].

Además, en un entorno digital donde los contenidos compiten constantemente por captar la atención y retener el mayor tiempo posible a los usuarios, las noticias que generan más interacción son promocionadas automáticamente por los algoritmos. Esto significa que, aunque una información sea falsa, si logra ser suficientemente impactante o emocional, será amplificada, creando una errónea veracidad simplemente por su popularidad [6].

Las consecuencias de esta problemática ya han sido documentadas: desde la manipulación electoral hasta el aumento de la polarización política y la desconfianza social [7]. Incluso en contextos de emergencia sanitaria o crisis global, como la pandemia de la COVID-19, las noticias falsas han puesto en riesgo la salud pública al propagar teorías o desinformación sobre tratamientos y vacunas [8]. Por tanto, la lucha contra la desinformación no solo es una cuestión de ética informativa, sino también un desafío crucial para la democracia, la salud social y la toma de decisiones basada en evidencia. En este escenario, la inteligencia artificial y, en particular, las técnicas de procesamiento del lenguaje natural (PLN), ofrecen nuevas oportunidades para identificar, clasificar y mitigar el impacto de las noticias falsas. Sin embargo, aún quedan muchos retos por abordar.

1.1. Motivación y necesidad del proyecto

En el ecosistema digital en el que vivimos, las *fake news* no son simplemente un problema menor, sino que son una amenaza real y creciente que impacta a todos los niveles de la sociedad. No se trata sólo de titulares falsos o de bromas virales sin consecuencias. La desinformación tiene efectos notables en la forma en que nosotros, los usuarios de las redes sociales, entendemos el mundo, en cómo tomamos decisiones y en cómo funcionamos como sociedad. Tendemos a formar creencias a partir de lo que vemos en redes sociales, y si lo que vemos está distorsionado o directamente manipulado, esa distorsión se traslada a nuestra visión del mundo donde vemos cómo se alimenta el extremismo y se corroe la confianza de los medios tradicionales o cualquier otra fuente de información verificada [2].

Pero el daño no se limita únicamente a lo político. En el ámbito económico, por ejemplo, la difusión de información falsa puede influir en decisiones relacionadas con la inversión, acabar con la reputación de una empresa en cuestión de horas o manipular los precios de los mercados financieros. Un solo tweet con información falsa puede provocar la caída de una acción o sembrar el pánico en una industria entera y esto no es algo futuro: ha pasado, y seguirá pasando mientras no se tomen medidas efectivas para controlar la propagación de desinformación [3].

En el ámbito de la salud, las consecuencias son todavía más graves. Durante la pandemia de la COVID-19, se viralizaron millones de publicaciones con información falsa sobre vacunas, tratamientos milagrosos o teorías conspirativas sobre el origen del virus. Algunas de estas publicaciones fueron más compartidas, y por ende aceptadas, que las recomendaciones oficiales de organismos de salud. El resultado de esto fue que miles de personas rechazasen tratamientos, retrasasen su vacunación o incluso pusiesen en riesgo su vida y la de los demás [8].

También la cultura y la sociedad se ven afectadas. Las *fake news* contribuyen a crear divisiones donde antes no las había, a sembrar el odio hacia determinados grupos sociales o a construir enemigos ficticios. Se construyen narrativas que enfrentan a comunidades, que distorsionan la historia, que niegan la ciencia o que quitan la verdad a causas sociales. Todo ello, impulsado por la lógica de lo viral y reforzado por algoritmos que priorizan lo que más impacta y no lo que más aporta [5].

Además, estamos ante una problemática perfecta: un entorno digital que no filtra, una sociedad que no siempre está equipada para detectar lo falso, y unos incentivos económicos que premian el escándalo sobre la veracidad. Aqui es donde la lucha contra la desinformación ya no puede depender solo de la buena voluntad de los usuarios o de las correcciones posteriores. Se necesitan herramientas tecnológicas que puedan actuar de forma preventiva y escalable.

Es en este punto donde la inteligencia artificial, y especialmente el procesamiento del lenguaje natural (PLN), abren la puerta a soluciones innovadoras. Gracias a estas tecnologías, podemos analizar grandes volúmenes de texto en tiempos de ejecución ridículos, identificar patrones de lenguaje característicos de las noticias falsas y desarrollar sistemas capaces de detectar y clasificar automáticamente los contenidos problemáticos. Pero incluso estas herramientas deben construirse con responsabilidad, teniendo en cuenta su transparencia, su explicabilidad y su impacto ético.

1.2. Objetivos

Frente a la expansión de la desinformación en redes sociales, surge la necesidad de diseñar herramientas tecnológicas capaces de identificar, entender y eliminar su impacto. El objetivo principal del presente trabajo se sitúa dentro de esa preocupación, y se basará en desarrollar un sistema automatizado capaz de detectar noticias falsas en redes sociales mediante técnicas avanzadas de PLN, modelos

de Machine Learning y Deep Learning, y métodos de análisis de sentimientos, con el fin de contribuir a una información más fiable, comprensible y verificable en el entorno digital. Para alcanzar dicho objetivo principal, se establecen los siguientes objetivos específicos:

- **Revisión bibliográfica:** Realizar un estudio exhaustivo de la literatura científica existente sobre noticias falsas en redes sociales, su impacto en distintos ámbitos (político, social, económico, sanitario, etc.) y las principales metodologías empleadas hasta la fecha para su detección. Se pondrá especial énfasis en los avances recientes en el uso de inteligencia artificial y técnicas de PLN para abordar esta problemática.

- **Construcción y enriquecimiento del dataset:** Integrar un conjunto de datos compuesto por noticias reales y falsas, complementado con publicaciones estilo tweet generadas artificialmente a partir de un modelo de lenguaje (LLM) y un RAG (*Retrieval Augmented Generation*) basado en ejemplos de tweets reales. Se buscará, además, garantizar la diversidad temática, estilística y emocional del dataset generado.

- **Diseño e implementación del sistema de detección:** Desarrollar y entrenar modelos de clasificación utilizando técnicas de aprendizaje automático y Deep Learning. Se evaluarán distintas arquitecturas, incluyendo modelos tradicionales y modelos basados en transformadores. Asimismo, se incorporará un sistema de generación aumentada de ejemplos (enfoque RAG) para simular mejor el entorno real de las redes sociales.

- **Análisis de sentimientos y polaridad textual:** Investigar la carga emocional de las noticias falsas frente a las verdaderas, aplicando herramientas de análisis de sentimientos para estudiar la relación entre polaridad y veracidad. Se estudiará si el tono emocional (negativo, alarmista, polarizador) constituye un predictor útil para la clasificación automática.

- **Explicabilidad y transparencia del modelo:** Aplicar técnicas de Inteligencia Artificial Explicable (XAI) para interpretar las decisiones del sistema de detección y analizar qué elementos lingüísticos o estructurales son más relevantes para su funcionamiento. Este enfoque busca mejorar la confianza y comprensión del modelo por parte de usuarios y analistas.

- **Evaluación experimental y visualización de resultados:** Validar el rendimiento del sistema y distintos modelos mediante métricas como precisión, recall, F1-score y AUC. Se presentarán los resultados de forma visual e interpretativa, incluyendo gráficos de polaridad, distribución temática, y mapas de calor de explicabilidad.

Con estos objetivos, se aspira a construir un sistema no solo eficaz en la clasificación, sino también útil como herramienta de análisis y reflexión sobre la naturaleza de la desinformación. Además, se pretende que el resultado final sea reproducible, éticamente consciente y fácilmente escalable para su futura aplicación en contextos reales.

1.3. Planificación del proyecto

La planificación de este Trabajo de Fin de Grado representada a través de un diagrama de Gantt, presentado en la Figura 1.1, muestra la distribución de tareas a lo largo de las 14 semanas ideadas para la realización del mismo.

Como se puede observar, se comienza con la construcción del dataset, donde a partir de uno general con noticias falsas, se genera un dataset adicional de tweets creados a partir de un LLM (en este caso Gemini) y basándose en otro dataset con tweets reales que servirá al modelo como RAG y tenga como resultado una fiel representación de tweets etiquetados acerca de dichas noticias. Seguidamente, se realiza el preprocesamiento de dichos datos para prepararlos para la ejecución de los modelos y posterior comparación. A esto le sigue la explicabilidad de los resultados obtenidos por los modelos para comprender la decisión de etiquetar una noticia como falsa o verdadera. Finalmente, se realiza un estudio de polaridad para determinar si se sigue alguna tendencia en las noticias falsas frente a las verdaderas.

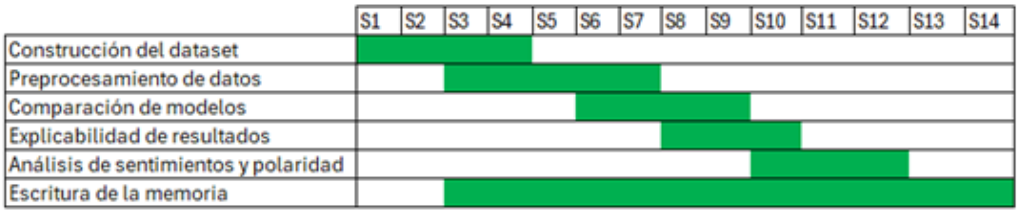

Figura 1.1: Plan de trabajo

Este diagrama de planificación se modificó frente al inicial debido a la incorporación de la arquitectura RAG en la generación de tweets a partir del LLM.

1.4. Estructura de la memoria

A lo largo de la memoria se sigue una estructura lógica que permite abordar de forma ordenada y completa los distintos componentes que conforman su desarrollo, desde la motivación inicial hasta las conclusiones finales y el análisis de impacto.

Tras la introducción presentada en este capítulo, en el capítulo 2, se incluye un estudio del estado del arte que recopila investigaciones previas en el ámbito de las *fake news*, especialmente en redes sociales. Se analizan enfoques actuales basados en modelos de lenguaje (LLM, del inglés *Large Language Models*), técnicas RAG y otros sistemas de clasificación de contenido, proporcionando una base sólida sobre la que se construye la propuesta del presente trabajo.

El siguiente bloque, el capítulo 3, se dedica a describir las tecnologías empleadas. Se detallan los lenguajes de programación, librerías y frameworks utilizados, así como el entorno de desarrollo. Este apartado permite comprender las decisiones técnicas adoptadas y su relación con los objetivos del proyecto. Pos-

teriormente, en el capítulo 4, se muestran los requisitos funcionales, no funcionales y técnicos del proyecto.

A nivel técnico, el verdadero núcleo del trabajo se encuentra en el desarrollo, redactado a lo largo del capítulo 5, que tras abordar los requisitos, se descompone en varias secciones clave: la preparación del conjunto de datos, la explicación del proceso de generación de dichos tweets mediante recuperación aumentada, y la implementación de distintos modelos de clasificación entrenados para distinguir entre noticias verdaderas y falsas.

Posteriormente, en el capítulo 6, se presentan los resultados obtenidos, donde se analizan tanto métricas de rendimiento como la explicabilidad de los textos clasificados permitiendo mostrar patrones distintos y en qué medida se han alcanzado los objetivos planteados.

La memoria también contempla una sección dedicada a las conclusiones y el trabajo futuro, en el capítulo 7, donde se resumen las aportaciones del proyecto, se reflexiona sobre sus limitaciones y se sugieren líneas de mejora para futuras investigaciones o desarrollos. Además, se incluye un análisis de impacto, en el capítulo 8, que examina cómo una herramienta como la abordada en el presente trabajo puede influir en diferentes ámbitos, como la sociedad, la economía o la cultura. Se considera no solo su utilidad técnica, sino también su potencial ético y social frente al fenómeno de la desinformación.

El trabajo finaliza con una recopilación bibliográfica, presente en el capítulo 8, que da soporte a cada uno de los bloques del proyecto. Se incluyen también anexos donde se mostrarán ejemplos de código, datos utilizados y resultados adicionales que complementan el cuerpo principal del documento.

Capítulo 2

Estado del arte

La detección de noticias falsas ha cobrado una gran importancia en los últimos años y con ello ha surgido un interés por la investigación dedicada a entender cómo funcionan y proponer métodos para combatirlas. En el caso particular de las redes sociales, el reto se complica aún más por la velocidad de propagación del contenido, la naturaleza de la información y la interacción directa entre usuarios.

Uno de los pilares teóricos más reconocidos propone la necesidad de abordar el fenómeno de las *fake news* desde una perspectiva científica interdisciplinar, es decir, combinando sociología, informática, ciencias cognitivas y comunicación. Esta visión ha guiado buena parte de los estudios posteriores, que han coincidido en señalar que el problema no solo reside en la información, sino también en los mecanismos sociales y tecnológicos que facilitan su difusión por diferentes vías [2].

Paralelamente, han aparecido contribuciones enfocadas en el análisis del contenido textual donde se realizó un análisis de la difusión de información en la red social Twitter y se confirmó que las noticias falsas no solo se propagan más rápido, sino que incluso tienen mayor alcance que las verdaderas. Estos hallazgos llevan a una nueva generación de herramientas basadas en Procesamiento de Lenguaje Natural (PLN) que buscan detectar patrones lingüísticos, sintácticos o semánticos característicos de las *fake news* [4].

En este punto, es donde se representa una de las revisiones más completas sobre métodos de detección automática, clasificando los enfoques en cuatro bloques: basados en contenido, en comportamiento del usuario, en redes de difusión y en características visuales [9].

Por otro lado, la presencia de usuarios organizados con el objetivo de propagar *fake news* también ha captado la atención de la comunidad científica, la que ha descubierto la existencia de estrategias coordinadas de desinformación llevadas a cabo por cuentas automatizadas (bots), capaces de alterar el modo en el que se informa y amplificar contenidos falsos en cuestión de minutos, aprovechando el algoritmo de difusión de las redes sociales.

También han ganado protagonismo los estudios centrados en la psicología del usuario que determinan que la falta de razonamiento crítico es un predictor más potente en cuanto a las noticias falsas que la ideología o el nivel educativo. Este tipo de investigaciones han abierto la puerta a soluciones híbridas, donde la inteligencia artificial se combina con intervenciones conductuales, como sistemas de verificación asistida o indicadores de credibilidad.

A nivel técnico, los últimos avances han sido impulsados por el fuerte crecimiento de los modelos de lenguaje de gran escala (LLM, del inglés *Large Language Models*), como BERT o deBERTa, que han demostrado una capacidad superior para detectar matices lingüísticos complejos y entender contextos. Estos modelos han sido aplicados con éxito en tareas de clasificación binaria (verdadero/-falso), pero también en tareas más complicadas como la verificación de hechos (fact-checking) o la detección de sesgos ideológicos.

De forma paralela, han surgido enfoques más recientes como el uso de técnicas de RAG, que combinan la potencia de los modelos generativos con bases de conocimiento externas para mejorar la calidad y fiabilidad del análisis textual. Esta técnica resulta especialmente prometedora para entornos dinámicos como las redes sociales, donde el contexto cambia constantemente y los modelos deben adaptarse a nuevas narrativas o formas de manipulación.

2.1. LLM y técnicas RAG

La evolución de los LLM ha transformado significativamente el procesamiento del lenguaje natural. Estos modelos, entrenados con volúmenes masivos de datos, han demostrado una capacidad suficientemente buena como para comprender, generar y contextualizar texto en múltiples tareas, desde la traducción automática o generación de texto hasta la clasificación semántica.

En el contexto de la lucha contra la desinformación o *fake news* en redes sociales, su capacidad para detectar patrones lingüísticos complejos y manejar lenguaje ambiguo o emocional los convierte en herramientas realmente prometedoras. Sin embargo, no están exentos de limitaciones. A pesar de su enorme conocimiento, los LLM tienden a generar respuestas coherentes, pero en ocasiones incorrectas respecto a las instrucciones que le indicamos (un fenómeno conocido como alucinación), y además, su conocimiento está limitado por la fecha de corte del corpus con el que han sido entrenados [10].

Para tratar de abordar estas limitaciones y brindar a los modelos de un acceso mucho más actualizado y contrastado a la información, se han desarrollado técnicas de Retrieval-Augmented Generation (RAG). Esta estrategia combina la generación de texto de un LLM con un sistema externo de recuperación de información, es decir, cuando se le plantea una consulta al sistema, antes de generar la respuesta, este realiza una búsqueda en una base de datos, recupera los documentos o la información más relevantes y los introduce como contexto en el prompt o instrucciones del modelo. De esta forma, el LLM puede generar texto enriquecido con información contrastada, más precisa, personalizada y actual. En el campo de la detección de *fake news*, esta arquitectura permite

integrar evidencias procedentes de fuentes verificadas para respaldar o desmentir afirmaciones dudosas, algo esencial cuando se trata de publicaciones breves o sesgadas como las que circulan en redes sociales.

Recientemente, se ha demostrado que los sistemas basados en RAG no solo mejoran la precisión, sino que también aumentan la explicabilidad del proceso de verificación. El sistema VeraCT Scan, por ejemplo, aplica esta arquitectura para analizar contenido sospechoso, identificar afirmaciones clave, recuperar evidencias de fuentes fiables y generar una justificación sobre su veracidad [11]. Lo más sorprendente es que no solo decide si una noticia es falsa o no, sino que explica por qué, citando la fuente y el razonamiento detrás del juicio, lo cual es crucial para generar confianza en el usuario y aumentar la transparencia del sistema.

Los beneficios de este enfoque se amplifican cuando se aplica al contexto dinámico de las redes sociales. En estas plataformas, el contenido cambia constantemente y la información circula a una velocidad que los modelos estáticos no pueden seguir. Algunos estudios recientes han experimentado con combinaciones de LLM como Mixtral o GPT-4, junto con pipelines RAG conectados a buscadores en tiempo real, obteniendo sistemas capaces de verificar publicaciones prácticamente al instante, incluso cuando los hechos comentados acaban de suceder [12].

También se está evolucionando hacia entornos multimodales, donde no solo se integra texto, sino también imágenes, metadatos o incluso comportamiento de los usuarios. Modelos como CRAVE proponen estructurar la información recuperada en forma de narrativa, lo que permite al LLM comprender mejor el contexto emocional o ideológico de una publicación. Esto resulta muy útil para identificar patrones típicos de desinformación que no siempre son evidentes en el contenido literal del texto [13].

Otro enfoque prometedor consiste en aprovechar la capacidad de los LLM para aprender a partir de pocos ejemplos (few-shot learning). En lugar de entrenar desde cero un sistema con grandes volúmenes de datos etiquetados, se les proporciona un contexto reducido con ejemplos cuidadosamente seleccionados. Esta técnica ha sido aplicada con éxito en marcos como KFFD (Knowledge-Guided Few-shot *fake news* Detection), donde se guía al modelo a través de plantillas y hechos obtenidos por RAG, logrando de esta manera resultados competitivos con un esfuerzo mínimo de entrenamiento supervisado [14].

No obstante, el despliegue de estos sistemas también plantea desafíos. La calidad y fiabilidad de las fuentes utilizadas para la recuperación es un factor crítico: si el sistema accede a información sesgada o desactualizada, su salida puede ser incorrecta. También es necesario tener en cuenta el coste computacional de operar modelos tan pesados en tiempo real, así como los riesgos éticos asociados a su uso: sesgos heredados, falta de trazabilidad. Por ello, la integración de mecanismos de explicabilidad como SHAP o LIME sigue siendo esencial, no solo para evaluar el rendimiento técnico, sino para dar al sistema una capa de transparencia muy importante en entornos tan sensibles como la verificación informativa.

En resumen, los LLM, cuando se combinan con arquitecturas avanzadas como las RAG, ofrecen un enfoque híbrido que equilibra creatividad y precisión. Su aplicación en la detección de noticias falsas representa uno de los avances más significativos en el ámbito de la inteligencia artificial aplicada. Aunque aún quedan muchos retos por superar, todo apunta a que estas tecnologías dterminarán el futuro de la lucha automatizada contra la desinformación, tanto en redes sociales como en otros medios digitales.

2.2. Trabajos relacionados

La lucha contra la desinformación en redes sociales ha evolucionado de manera significativamente en los últimos años, en paralelo con el desarrollo de modelos de lenguaje de gran tamaño (LLM) y técnicas de aprendizaje profundo más avanzadas. Lejos de los primeros enfoques basados únicamente en reglas o en clasificadores tradicionales, los últimos avances se centran en construir sistemas que no solo detectan noticias falsas, sino que también entiendan el contexto y expliquen sus decisiones.

Uno de los más prometedores ha sido el diseño de arquitecturas que combinan fuentes internas (como el contenido del texto) con señales externas (como la fiabilidad de las fuentes o la interacción social). Se ha desarrollado recientemente un marco de detección que sigue el enfoque de "Detectar, Investigar, Juzgar y Determinar" en escenarios denominados de pocos disparos (few-shot), donde la disponibilidad de ejemplos etiquetados es limitada. Este sistema mejora el razonamiento semántico al combinar capacidades de lenguaje natural con bases de conocimiento y bases de hechos verificables, logrando identificar patrones de manipulación informativa incluso con escasos datos [15].

Siguiendo esta misma línea, se ha introducido una variante de la arquitectura de atención dentro de los LLM que prioriza la credibilidad de las fuentes en el proceso de inferencia. Esta técnica, conocida como "Credibility-Aware Attention Modification" (CrAM), ajusta dinámicamente los pesos de atención según la fiabilidad de los documentos recuperados, reduciendo el riesgo de incorporar información incorrecta durante la generación de respuestas [16]. Esta mejora ha sido especialmente útil en sistemas RAG (Retrieval-Augmented Generation), donde la precisión de los documentos externos influye directamente en la veracidad de la salida del modelo.

Más allá de esto, el uso de sistemas optimizados para la detección de *fake news* en tiempo real ha cobrado fuerza. Se ha implementado recientemente una arquitectura basada en BERT que opera sobre plataformas en la nube y que permite clasificar noticias falsas con una precisión cercana a la perfecta. Este modelo fue entrenado y evaluado en múltiples categorías temáticas (salud, política, entretenimiento, ciencia, entre otros), demostrando una mejora considerable con respecto a otros enfoques menos contextuales o menos robustos a cambios léxicos [17].

Otro enfoque emergente y prometedor es la combinación de inteligencia artificial con inteligencia colectiva. Esto se consigue a través de métodos híbridos que

fusionan la predicción automática con la participación de usuarios humanos, ponderando ambas fuentes de decisión según su fiabilidad histórica. Este tipo de enfoque se considera especialmente útil para abordar casos de ambigüedad, sarcasmo o noticias con múltiples interpretaciones posibles, y mejora la robustez del sistema ante campañas de desinformación [18].

En entornos de adversarial learning, también se han propuesto estrategias innovadoras para mejorar la generalización de los modelos frente a datos manipulados. Un caso reciente ha sido la implementación de un pipeline de generación de noticias falsas utilizando retroalimentación adversaria en tiempo real. En este sistema, las respuestas generadas por el modelo son reutilizadas como ejemplos para fortalecer su capacidad de detección, haciendo que aprenda de sus propios errores y refuerce su resistencia ante ataques de texto generados automáticamente [19]. Esta estrategia, que combina técnicas generativas con aprendizaje por refuerzo, se está consolidando como una vía prometedora para modelos cada vez más robustos y adaptables.

Por otro lado, se ha puesto especial atención en mejorar la explicabilidad de los sistemas de detección donde no solo buscan identificar si una noticia es falsa, sino también proporcionar una justificación clara y comprensible de por qué lo es. Esto ha motivado el desarrollo de frameworks que integran mecanismos de razonamiento justificable, permitiendo a los usuarios rastrear qué partes del texto, qué hechos externos o qué patrones lingüísticos han influido en la decisión del modelo. Esta capacidad de justificar decisiones es clave en aplicaciones sensibles, como medios de comunicación, procesos judiciales o auditorías algorítmicas. Asimismo, se han explorado estrategias más centradas en el comportamiento del usuario y la dinámica de difusión de la desinformación. Algunos estudios recientes han incorporado variables como la estructura temporal del tweet, la red de seguidores y la velocidad de compartición como señales adicionales para mejorar la detección. La integración de estas dimensiones sociales ha demostrado ser eficaz para anticipar si una publicación ganará tracción viral y, en caso afirmativo, si puede contener información engañosa.

En conjunto, el panorama actual muestra una clara evolución hacia sistemas más completos, multimodales y explicables. Las líneas de investigación más recientes coinciden en el uso de modelos generativos con retroalimentación, atención guiada por fiabilidad, integración de datos externos y adaptación en tiempo real. Esta tendencia no solo responde a la problemática de las *fake news*, sino también a la necesidad urgente de desarrollar soluciones escalables, transparentes y efectivas en un contexto digital que cambia constantemente.

2.3. Evolución técnica: de ML a Transformers en la detección de noticias falsas

Como se ha podido analizar a lo largo de este capítulo, la evolución tecnológica en la detección de noticias falsas ha seguido una trayectoria progresiva en la que cada etapa ha aportado capacidades nuevas y complementarias. Desde los primeros modelos de aprendizaje automático hasta los más recientes basados en transformers, el objetivo ha sido siempre el mismo: identificar patrones que permitan distinguir entre información real y la falsa, pero la manera de hacerlo ha cambiado radicalmente.

Los primeros enfoques en este campo se apoyaron en técnicas tradicionales de Machine Learning. Estos sistemas sirvieron para sentar las bases de lo que vendría después. Con ellos se logró demostrar que existen patrones medibles en los textos falsos, como el uso de determinados términos, estructuras sensacionalistas o indicios de polarización emocional [20]. En cambio, su rendimiento dependía en gran medida de la calidad de los hiperparámetros y de los datos, lo que limitaba su capacidad para adaptarse a nuevos estilos o formatos de desinformación.

Con el auge del Deep Learning, esta dependencia comenzó a desaparecer. Las redes neuronales profundas permitieron aprender directamente del texto sin necesidad de definir manualmente qué atributos eran relevantes. Esto significó un cambio donde los modelos comenzaron a captar por sí mismos relaciones semánticas más complejas, estilos discursivos y matices que antes quedaban fuera del alcance de los sistemas clásicos [21]. Además, la posibilidad de combinar diferentes tipos de datos, como texto e imágenes, abrió la puerta a modelos multimodales capaces de identificar *fake news* en otros formatos.

El siguiente salto significativo se produjo con la llegada de los modelos basados en transformers, cuya arquitectura, centrada en mecanismos de atención, revolucionó por completo el procesamiento del lenguaje natural. A diferencia de las redes anteriores, los transformers no solo entienden el significado de las palabras, sino también su relación con el contexto, incluso en fragmentos largos de texto, lo que resulta crucial en la detección de desinformación, donde una afirmación aislada puede pasar desapercibida, pero empieza a cobrar sentido cuando ampliamos la ventana [22].

Además, estos modelos han facilitado la transferencia de conocimiento entre dominios y lenguas, lo que los hace especialmente útiles en un entorno como las redes sociales. Gracias a ellos, se ha conseguido no solo mejorar la precisión de los sistemas de detección, sino también reducir el tiempo y los datos necesarios para entrenarlos [23].

En resumen, esta evolución progresiva en el tiempo ha incrementado la capacidad técnica de los sistemas y ha ido acercando la detección de *fake news* a una solución más adaptable a nuevos retos donde cada uno de estos avances ha sido construido sobre el anterior, conformando una base sólida para las soluciones actuales y las que seguro, están por venir.

Capítulo 3

Tecnologías empleadas

Durante el desarrollo de este trabajo se ha hecho uso de un conjunto variado de herramientas o tecnologías que han permitido abordar de forma eficiente las tareas de recopilación, procesamiento, modelado, análisis y visualización de los datos. En esta sección se describen las principales tecnologías utilizadas, sus ventajas y su papel en la implementación del sistema propuesto.

3.1. Lenguaje de programación y entornos de desarrollo

El lenguaje de programación principal utilizado en el desarrollo ha sido Python, por su versatilidad, simplicidad sintáctica y el amplio ecosistema de librerías disponibles. Es ampliamente adoptado en campos como el análisis de datos, el aprendizaje automático y el procesamiento de lenguaje natural, lo que lo convierte en una elección adecuada tanto para prototipado rápido como para desarrollo a escala [24].

Este desarrollo se ha realizado en dos entornos principales. Por un lado, Jupyter Notebook ha facilitado el análisis exploratorio de datos, la ejecución de los modelos de machine learning y deep learning y la visualización de resultados de forma interactiva [25]. Por otro lado, Google Colab ha permitido el uso de recursos de computación en la nube, especialmente unidades GPU, necesarias para entrenar modelos más pesados como los transformadores [26].

3.2. Procesamiento de lenguaje natural

Para la etapa de preprocesamiento de lenguaje natural se ha empleado la biblioteca NLTK, que proporciona utilidades para tokenizar, eliminar palabras vacías y realizar análisis morfosintáctico [27]. El análisis de polaridad se ha realizado con VADER, un modelo diseñado específicamente para el análisis de sentimientos en textos breves, como los provenientes de redes sociales [28].

3.3. Vectorización y modelado de texto

El texto ha sido transformado a vectores utilizando diversas técnicas. TF-IDF, disponible en scikit-learn, ha sido usada para los modelos tradicionales al capturar la relevancia relativa de los términos en el corpus [29]. Para una representación más semántica para el dataset de solo tweets, se han utilizado embeddings generados por FastText mediante la librería Gensim, lo que permite conservar similitud contextual entre palabras y obtener mejores resultados en las pruebas debido a que los textos cortos presentasn ciertas limitaciones [30].

En modelos de redes neuronales se ha aplicado el tokenizador de Keras junto con secuencias rellenadas para mantener uniformidad. Para los modelos basados en transformadores se han utilizado modelos preentrenados como DeBERTa y Bert, facilitando la transferencia de conocimiento desde grandes corpus [31].

3.4. Modelos de aprendizaje automático y profundo

Para abordar la clasificación binaria entre noticias falsas y verdaderas, se han utilizado distintos modelos de aprendizaje automático (ML) y aprendizaje profundo (DL), seleccionados en función de su capacidad para tratar texto y adaptarse a contextos con datos reales provenientes de redes sociales.

En primer lugar, se implementaron modelos clásicos mediante la biblioteca scikit-learn:

- Naïve-Bayes Multinomial
- Logistic Regression
- Random Forest
- Gradient Boosting

En cuanto al aprendizaje profundo, se han entrenado distintos modelos con TensorFlow y Keras, específicamente adaptados a la naturaleza secuencial del lenguaje y a la tarea de clasificación de texto:

- MLP (*Multilayer Perceptron*)
- RNN (*Recurrent Neuran Network*)
- LSTM (*Long Short-Term Memory*)
- BiLSTM (*Bidirectional LSTM*)

Por último, se utilizaron modelos preentrenados de tipo transformador, ejecutados en Google Colab para aprovechar aceleración con GPU y facilitar su entrenamiento con recursos computacionales adecuados:

- BERT (*Bidirectional Encoder Representations from Transformers*)
- DeBERTa (*Decoding-enhanced BERT with disentangled attention*)

Estos modelos permiten comparar enfoques de distinta complejidad y capacidad de generalización: desde modelos simples como MLP hasta modelos de última generación como DeBERTa. La elección de arquitecturas variadas ha sido clave para analizar el rendimiento del sistema en distintos escenarios, evaluar el impacto del contexto textual y estudiar el equilibrio entre precisión y explicabilidad.

3.5. Generación aumentada y uso de modelos generativos

Una parte fundamental del trabajo ha consistido en la generación de tweets artificiales que simulan el comportamiento de usuarios reales en redes sociales. Para ello, se ha utilizado Gemini 2.0 Flash a través de Vertex AI, la plataforma de inteligencia artificial de Google Cloud [32]. Gemini es un modelo de lenguaje de última generación capaz de generar texto coherente, contextualizado y realista. Gracias a su integración en Vertex AI, ha sido posible automatizar la creación de publicaciones en formato tweet a partir de noticias falsas y reales, ampliando así el dataset de entrenamiento y simulando un entorno más cercano al uso real de plataformas sociales.

Esta estrategia de generación aumentada ha seguido un enfoque basado en RAG (Retrieval-Augmented Generation), proporcionando al modelo ejemplos de tweets reales como contexto para guiar la generación de contenido más verosímil y relevante. Esta capacidad de contextualización y control de la generación textual ha sido clave para mejorar la calidad y representatividad del corpus de entrenamiento.

3.6. Explicabilidad de los modelos

Para garantizar la transparencia en las decisiones del sistema de detección y sus distintos modelos, se han utilizado técnicas de interpretabilidad como el análisis de coeficientes en modelos lineales o la visualización de probabilidades logarítmicas en clasificadores bayesianos. Además, se han generado visualizaciones de palabras relevantes en cada predicción, lo que ha permitido identificar patrones lingüísticos influyentes en la clasificación.

3.7. Visualización y análisis

La representación de resultados se ha realizado mediante bibliotecas como Matplotlib y Seaborn, generando gráficos de barras, diagramas de dispersión, histogramas y boxplots [33] [34]. Se ha utilizado WordCloud para mostrar los términos más relevantes en noticias verdaderas y falsas, mientras que Pandas ha sido esencial para el análisis exploratorio, la agregación de métricas y la manipulación de estructuras tabulares [35].

3.8. Gestión y almacenamiento de datos

El dataset principal y los tres datasets secundarios se han almacenado y procesado en formato CSV. Para preservar el estado de los modelos y tokenizadores tras el entrenamiento, se ha utilizado serialización mediante Joblib, facilitando su reutilización y validación en fases posteriores [36].

3.9. Evaluación de modelos

La evaluación de los modelos se ha realizado utilizando métricas clásicas en tareas de clasificación binaria: exactitud, precisión, recall, F1-score y área bajo la curva ROC (AUC), todas ellas implementadas a través de la biblioteca scikit-learn [29].

La elección de estas métricas responde a la necesidad de evaluar correctamente el desempeño del sistema más allá de un simple acierto general. En el contexto de detección de noticias falsas, los errores de clasificación pueden tener implicaciones muy diferentes. Por ejemplo, etiquetar incorrectamente una noticia falsa como verdadera (falso negativo) puede ser más dañino que el error inverso, ya que contribuye a la propagación de desinformación.

La métrica de precisión indica qué proporción de los elementos clasificados como positivos (noticias falsas) lo son realmente, lo cual es clave para evitar falsos positivos. Por su parte, el recall mide qué proporción de todas las noticias falsas existentes ha sido correctamente identificada, lo que resulta especialmente útil en contextos donde se quiere minimizar el riesgo de que pase desapercibida una *fake news*.

El F1-score combina ambas métricas en una media armónica, equilibrando precisión y recall, lo que es especialmente útil cuando existe un desbalance entre clases o una necesidad de equilibrar ambos tipos de error. Finalmente, el AUC-ROC permite evaluar la capacidad del modelo para discriminar entre clases a diferentes umbrales de decisión, aportando una visión global del comportamiento del sistema en escenarios variados.

Además, se ha aplicado validación cruzada para obtener una estimación más robusta del rendimiento del modelo y evitar sobreajuste. Asimismo, se ha utilizado GridSearch para optimizar los hiperparámetros de los algoritmos, mejorando la capacidad predictiva de los clasificadores y asegurando una comparación justa entre ellos.

Capítulo 4

Requisitos del sistema

Este capítulo recoge los requisitos principales que guían el desarrollo del sistema de detección automática de noticias falsas en redes sociales. Se establecen tanto los requisitos funcionales y no funcionales, como los requisitos técnicos y las limitaciones asumidas a lo largo del proyecto.

4.1. Requisitos funcionales

Los requisitos funcionales describen las funcionalidades esenciales que el sistema debe cumplir:

- El sistema debe permitir la clasificación binaria de entradas textuales como noticias verdaderas o falsas.
- Debe aceptar tres tipos distintos de entrada:
  - Solo texto de la noticia original.
  - Solo texto del tweet generado sintéticamente.
  - Combinación del texto de la noticia y del tweet asociado.
- Debe ser capaz de generar contenido sintético tipo tweet a partir de noticias mediante un modelo generativo (Gemini 2.0 Flash).
- El sistema debe almacenar los modelos entrenados y los vectorizadores/-tokenizadores para su reutilización.
- Debe proporcionar medidas de rendimiento de los modelos mediante métricas estándar (accuracy, precisión, recall, F1-score, AUC).
- En los modelos que lo permitan, se debe ofrecer una explicación interpretable del resultado de clasificación.

4.2. Requisitos no funcionales

Los requisitos no funcionales definen condiciones que afectan al funcionamiento general del sistema, pero no están relacionadas con funciones específicas:

- El sistema debe ejecutarse de forma reproducible en entornos Python (Jupyter o Colab).
- El código debe estar organizado por bloques funcionales (preprocesamiento, entrenamiento, evaluación).
- La generación de contenido debe realizarse de manera controlada, evitando ambigüedades o contradicciones evidentes.
- El sistema debe priorizar la transparencia y trazabilidad de resultados para facilitar su interpretación por parte de analistas o usuarios finales.
- Debe poder entrenar modelos en tiempos razonables (menos de una hora por modelo, en la mayoría de los casos).

4.3. Requisitos técnicos

Los requisitos técnicos describen las herramientas, entornos y librerías utilizados en el desarrollo del sistema:

- Lenguaje de programación: Python 3.12.
- Entornos de ejecución:
  - Jupyter Notebook para modelos de machine learning y deep learning.
  - Google Colab con GPU para modelos tipo Transformer.
- Librerías principales:
  - Procesamiento del lenguaje: NLTK, VADER, re, wordcloud.
  - Vectorización: scikit-learn (TF-IDF), Gensim (FastText), Keras Tokenizer.
  - Modelado: scikit-learn, TensorFlow, Keras, HuggingFace Transformers.
  - Visualización: matplotlib, seaborn, pandas.
  - Almacenamiento: pickle, joblib.
- Infraestructura de generación de texto: Gemini 2.0 Flash mediante Vertex AI.
- Datos: Dataset inicial de noticias verdaderas y falsas, dataset externo de tweets reales, y dataset mixto con tweets generados a partir de noticias.

Capítulo 5

Desarrollo del proyecto

En este capítulo se describe el proceso completo llevado a cabo para el desarrollo del sistema de detección automática de noticias falsas en redes sociales. Desde la recopilación y generación del conjunto de datos, pasando por el preprocesamiento de los datos y la representación de los contenidos, hasta el entrenamiento, evaluación e interpretación de modelos de clasificación.

El objetivo es diseñar un flujo de trabajo por módulos, reproducible y fácilmente extensible, que permita comparar distintos enfoques de aprendizaje automático, aprendizaje profundo y modelos basados en transformadores. Para ello, se han construido tres versiones diferenciadas del dataset (noticias, tweets generados y combinación de ambos), y se han entrenado y validado diversos modelos sobre cada una de ellas.

A lo largo del capítulo también se detallan los métodos utilizados para visualizar y analizar los resultados, así como las herramientas empleadas para garantizar la transparencia y explicabilidad de las decisiones tomadas por los modelos. Finalmente, se abordan algunas consideraciones éticas derivadas del uso de inteligencia artificial para la generación y clasificación de contenido textual.

Además, en la Figura 5.1 se presenta un diagrama de la arquitectura del proyecto. En ella podemos ver tres partes diferenciadas, la primera hace referencia a la elección y construcción del dataset, tanto el base como el generado por LLM. La segunda parte incluye los modelos elegidos para la evaluación del sistema y por último se contemplan las métricas de evaluación y utilización de otros métodos para evaluar los resultados de cada modelo.

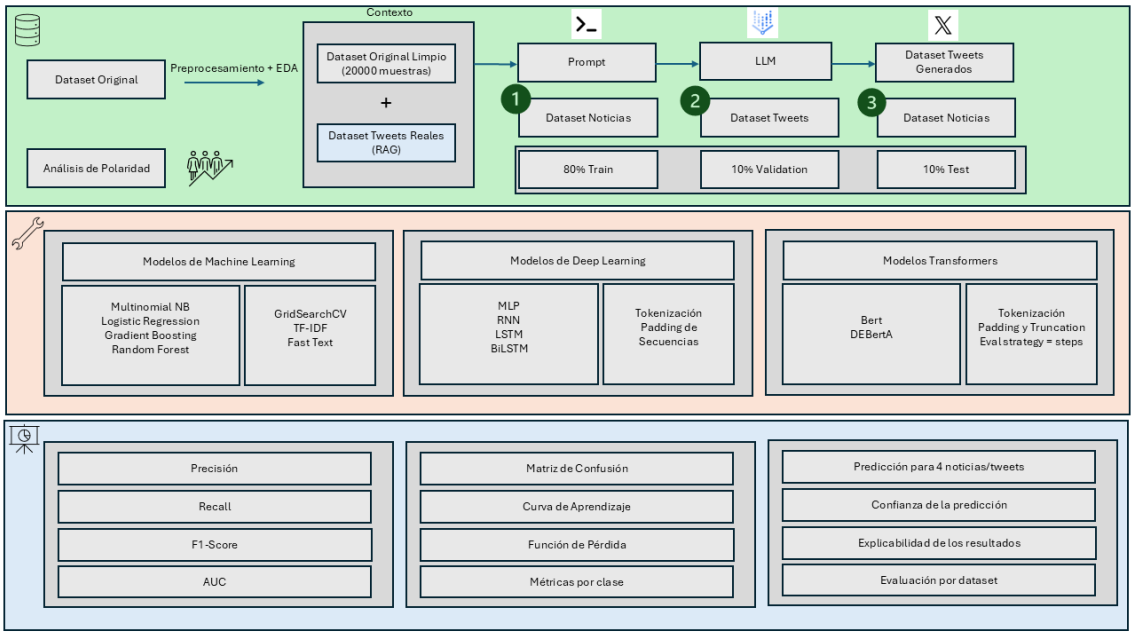

Figura 5.1: Diagrama general de la arquitectura del sistema desarrollado.

5.1. Obtención y preparación del conjunto de datos

Esta sección describe el proceso completo de construcción del conjunto de datos utilizado para entrenar y evaluar el sistema de detección de noticias falsas. El objetivo es contar con una base realista y representativa que combine contenido auténtico y generado artificialmente, replicando así las dinámicas propias de las redes sociales.

5.1.1. Selección del dataset base

El trabajo comienza con la búsqueda de un conjunto de noticias que estén claramente etiquetadas como verdaderas o falsas, y que estén distribuidas de forma equilibrada. Para ello se selecciona el dataset *Fake news* Detection Datasets publicado en la plataforma Kaggle [37], que contiene noticias procedentes de diversos medios, separadas ya en archivos distintos según su veracidad. Esto podemos verlo en la Tabla 5.1 [37]. El idioma de este dataset es el ingés debido a que presenta mejores resultados que otro idiomas como el español. Más allá, la Tabla 5.1 muestra la distribución temática de los artículos incluidos en el dataset original, diferenciando entre noticias reales y falsas. En el caso de las noticias verdaderas, se observa una concentración en dos grandes categorías: politics-news (11.272 artículos) y world-news (10.145 artículos), lo que refleja una cobertura informativa centrada en asuntos internacionales y de actualidad política global, característica habitual de medios tradicionales.

Por otro lado, las noticias falsas presentan una mayor fragmentación temática. Aunque la categoría genérica news agrupa la mayor parte de los artículos (9.050), también destacan otras como politics (6.841) y left-news (4.459), que

pueden estar más asociadas a contenidos polarizantes o sesgados. Otras categorías menos representadas como government-news, middle-east o US news sugieren una diversidad artificial de temas, probablemente con el objetivo de simular variedad informativa sin una base periodística sólida.

Esta diferencia en la estructura temática entre ambas clases puede ser indicativa del enfoque editorial: mientras las noticias verdaderas responden a secciones informativas reconocibles, las falsas tienden a reproducir titulares impactantes sin seguir una lógica editorial clara. Este patrón puede ser aprovechado por los modelos de clasificación al detectar la inconsistencia semántica y contextual propia de los contenidos engañosos.

Tabla 5.1: Resumen del dataset utilizado

News	**Size**	**Subjects**	
	(Number of articles)	**Type**	**Articles size**
Real-News	21417	*World-News*	*10145*
		Politics-News	*11272*
Fake-News	23481	*Government-News*	*1570*
		Middle-east	*778*
		US News	*783*
		left-news	*4459*
		politics	*6841*
		News	*9050*

Se combinan ambas fuentes, tanto las noticias verdaderas como las falsas, y se eliminan las columnas irrelevantes como el título, la temática o la fecha de publicación. Posteriormente, se añade una nueva columna llamada label, asignando el valor 0 a las noticias falsas y 1 a las verdaderas. Se seleccionan aleatoriamente 10.000 entradas de cada tipo, lo que permite obtener un conjunto final balanceado con 20.000 ejemplos.

Como parte del análisis exploratorio, se generaron nubes de palabras para observar los términos más frecuentes en cada clase, lo que permite identificar diferencias léxicas o temáticas entre las noticias verdaderas y falsas (Figura 5.2 y Figura 5.3). Como podemos observar, las nubes de palabras permiten identificar diferencias léxicas relevantes entre las noticias verdaderas y las falsas. En el caso de las noticias reales (izquierda), destacan términos como said, us, Trump, state o Wednesday, que remiten a declaraciones institucionales o coberturas informativas estructuradas. Por el contrario, en las noticias falsas (derecha), aunque se repiten algunas palabras como said o Trump, aparecen otras como one, people o know, más genéricas y emocionales.

Este contraste sugiere que, mientras las noticias verdaderas tienden a emplear un lenguaje más concreto y referencial, las noticias falsas recurren con mayor frecuencia a términos vagos, impersonales o apelativos, lo que puede reflejar una estrategia narrativa centrada en generar impacto emocional más que en informar con precisión. Esta diferencia es especialmente relevante para los modelos de clasificación, que pueden aprender a reconocer patrones léxicos y contextuales en función del tipo de fuente.

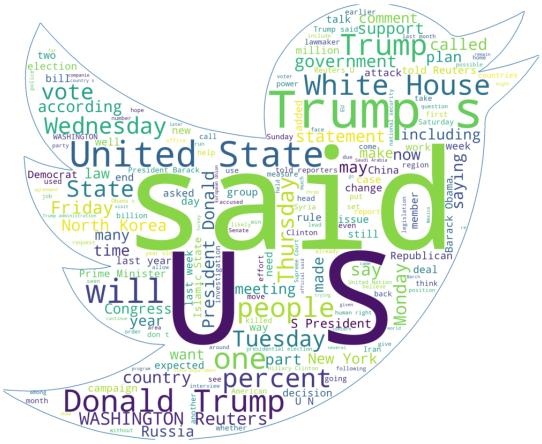

Figura 5.2: Wordcloud: Noticias reales

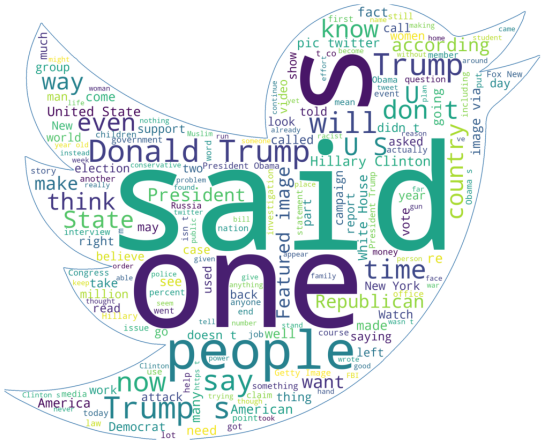

Figura 5.3: Wordcloud: Noticias falsas

5.1.2. Análisis de polaridad de las noticias

Además del análisis semántico y estructural, se decide estudiar la polaridad emocional de las noticias para comprobar si existe una diferencia significativa entre noticias reales y falsas en cuanto a su tono afectivo. Para ello, se utiliza la herramienta VADER (Valence Aware Dictionary for Sentiment Reasoning), integrada en la librería NLTK. Esta herramienta ofrece una métrica denominada compound score, cuyo valor oscila entre -1 (sentimiento negativo) y +1 (sentimiento positivo).

Se aplica este análisis a las 20.000 noticias del conjunto de datos original, generando un valor de polaridad por texto. A partir de estos resultados, se calculan estadísticas descriptivas por clase. Como se muestra en la Tabla 5.2, la media de polaridad de las noticias reales es positiva (0.057), mientras que la de las falsas es negativa (-0.084), lo que sugiere que las noticias verdaderas tienden a expresar mensajes más neutrales o positivos, frente a un tono más negativo o alarmista en las falsas. Esta diferencia se mantiene a nivel de la mediana y se confirma por la desviación estándar.

Tabla 5.2: Estadísticas de polaridad por clase

Clase	Media	Desv. Estándar	Mínimo	Máximo	Mediana	Tamaño
FALSA	-0.084	0.842	-1.0000	0.9999	-0.2206	10000
REAL	0.058	0.826	-0.9998	0.9999	0.2493	10000

La Figura 5.4 muestra la distribución de polaridad en forma de boxplot. En ella se observa claramente que las noticias falsas presentan un rango más centrado en valores negativos, mientras que las reales se distribuyen más hacia valores positivos.

La Figura 5.5 refuerza esta conclusión, mostrando la densidad de polaridad para ambas clases. Se aprecia un mayor número de noticias falsas en los extremos negativos (especialmente en el valor -1), mientras que las reales tienden a concentrarse más en el rango positivo. Este patrón sugiere que la desinformación

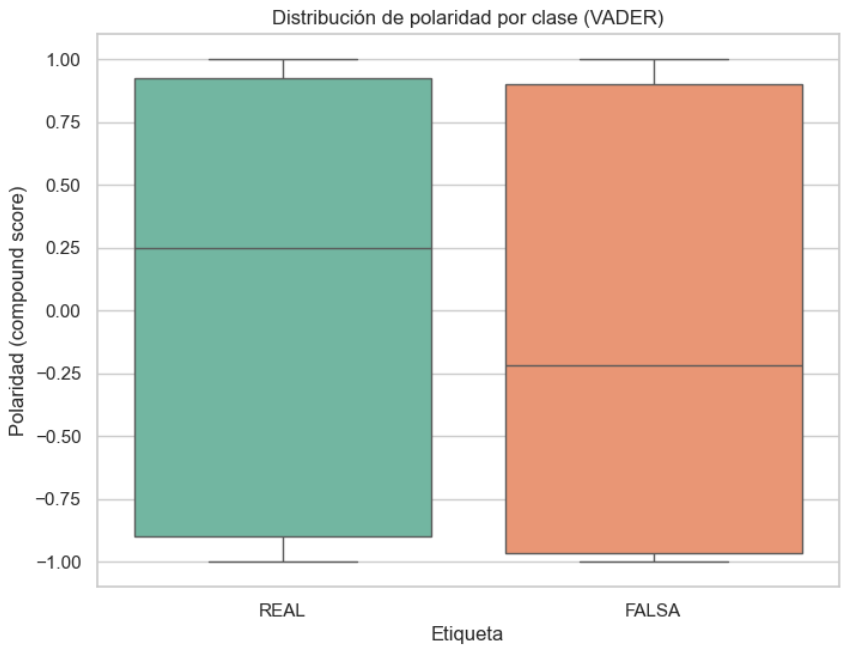

Figura 5.4: Distribución de polaridad por clase (VADER)

tiende a explotar emociones fuertes, como la indignación o el miedo, mientras que las noticias reales mantienen un estilo más moderado o neutro.

Para confirmar si esta diferencia es significativa desde el punto de vista estadístico, se realiza un test no paramétrico de Mann-Whitney U, adecuado para comparar distribuciones sin asumir normalidad. El valor obtenido (U = 55.287.170) y el correspondiente p-valor extremadamente bajo ($2{,}33 \cdot 10^{-38}$) permiten rechazar la hipótesis nula y afirmar que existe una diferencia estadísticamente significativa entre la polaridad de las noticias reales y falsas.

Esto es especialmente relevante en el contexto de la clasificación automática, ya que el tono emocional puede actuar como un indicador complementario útil para mejorar el rendimiento de los modelos. Además, refuerza la hipótesis de que las noticias falsas tienden a emplear un lenguaje más polarizado, sensacionalista o extremo, con el objetivo de captar más fácilmente la atención del lector.

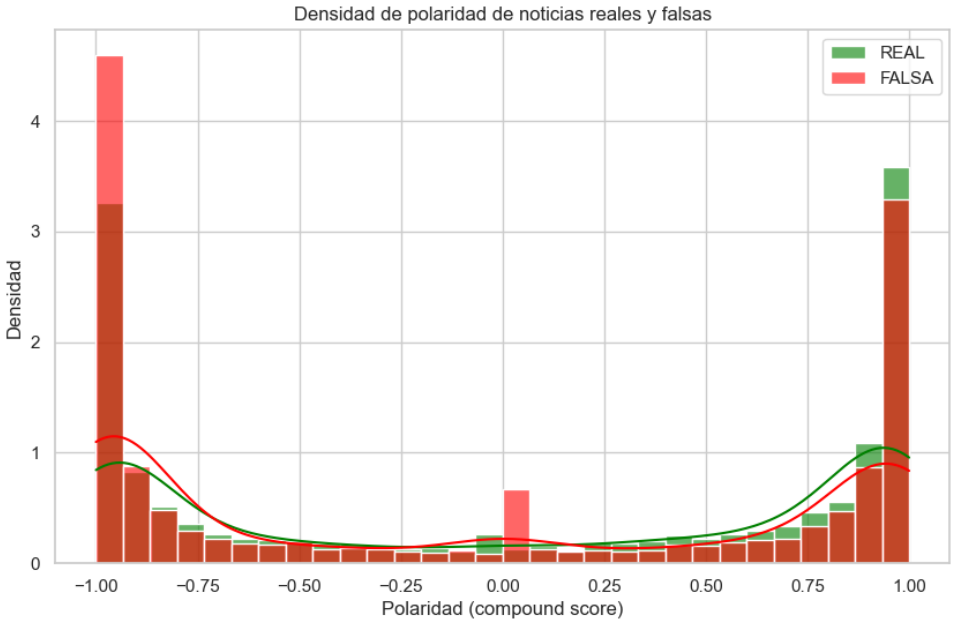

Figura 5.5: Densidad de polaridad de noticias reales y falsas

5.1.3. Generación de tweets mediante LLM y RAG

Para reflejar de forma realista cómo circulan las noticias en redes sociales, se decide generar tweets a partir de las noticias originales. Estos tweets permiten transformar el contenido largo y estructurado de una noticia en una forma breve, informal y emocionalmente cargada, similar al tipo de publicaciones que se viralizan en plataformas como Twitter. Con ello, se persigue no solo enriquecer el conjunto de datos, sino también ofrecer un punto de análisis más adaptado al consumo digital actual.

La generación de estos tweets se lleva a cabo con la ayuda de Gemini 2.0 Flash, un modelo de lenguaje desarrollado por Google y disponible en Vertex AI. La elección de este modelo frente a otras alternativas (como GPT-3.5 o PaLM) responde a varios motivos. En primer lugar, Gemini 2.0 Flash está optimizado para tareas de generación rápida y eficiente, lo cual permite trabajar con grandes volúmenes de datos sin comprometer el rendimiento ni el coste computacional. En segundo lugar, mantiene una gran coherencia semántica y emocional en contextos breves, lo que lo convierte en una herramienta ideal para la redacción de tweets.

El proceso completo de generación se organiza a través de un pipeline implementado en Python. Para cada noticia del dataset, se generan dos tweets únicos que simulan reacciones espontáneas de usuarios. Un componente clave de esta generación es el uso de un enfoque basado en RAG (Retrieval-Augmented Generation), en el cual se recupera un conjunto amplio de ejemplos reales de tweets a partir de un segundo dataset de Kaggle [38]. Estos ejemplos no se emplean como datos de entrenamiento, sino como contexto para el prompt de entrada, actuando como guía de estilo, tono y estructura.

El prompt se elaboró cuidadosamente la instrucción enviada a Gemini 2.0 Flash para cada artículo. En términos generales, el prompt incluía un bloque de contexto más de mil tweets reales representativos, seguido de la noticia original etiquetada, y finalmente la tarea solicitada al modelo. Se enfatiza en que los tweets deben escribirse en estilo informal, con tono auténtico y humano de redes sociales (incluyendo hashtags relevantes y expresiones vulgares), y capturar la esencia de la noticia. El LLM, al procesar este prompt, genera dos mensajes simulando tweets.

La Figura 5.6 ilustra gráficamente cómo funciona el enfoque RAG. El modelo no genera texto desde cero, sino que primero accede a un conjunto de información contextual que refuerza su respuesta. En este caso, la recuperación de ejemplos reales permite que el modelo ajuste su tono, vocabulario y estilo a un registro más verosímil y cercano al lenguaje social real.

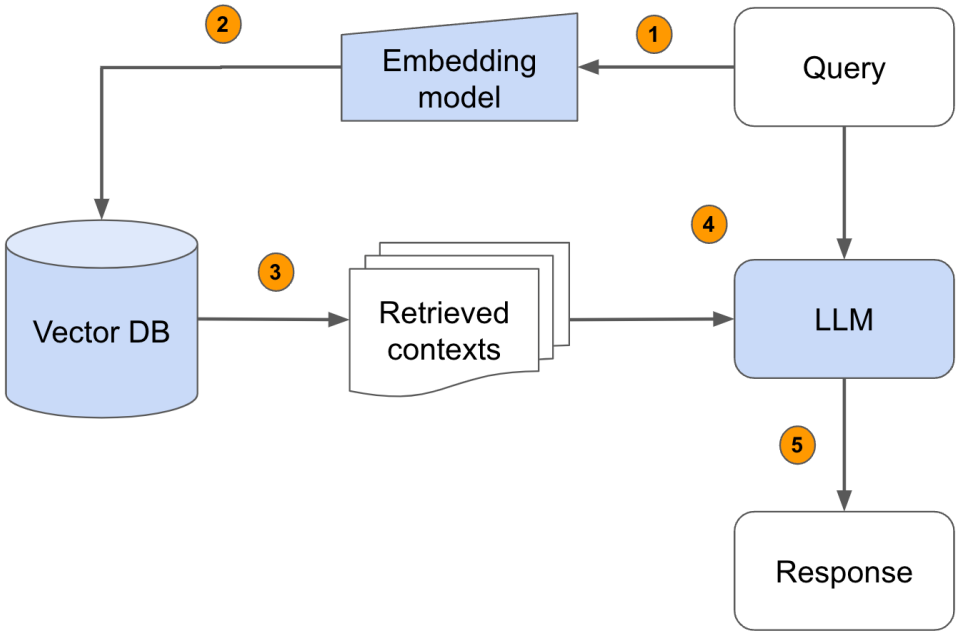

Figura 5.6: Funcionamiento general del sistema RAG. Fuente: [1]

Este procedimiento permite generar un segundo conjunto de datos compuesto por 40.000 tweets (dos por cada noticia original), todos etiquetados con la misma clase que la noticia de la que proceden. El motivo de generar dos tweets por noticia fue doble: por un lado, agregar más variedad lingüística por cada pieza informativa (un solo tweet podría no reflejar todas las facetas de una noticia, mientras que dos tweets permiten perspectivas ligeramente diferentes, por ejemplo uno podría ser más neutral y factual y el otro más emocional u opinativo). Por otro lado, duplicar el número de muestras de texto corto disponibles para analizar, lo cual resulta útil al entrenar modelos en el conjunto de tweets exclusivamente. Este nuevo corpus se convierte en un recurso especialmente valioso para comparar estilos y evaluar modelos en texto breve.

En conjunto, la combinación de Gemini 2.0 Flash y el enfoque RAG permite

una generación de texto contextualizada, rápida y realista. Este proceso no solo amplía el conjunto de datos, sino que simula las dinámicas de comunicación en redes sociales, algo crucial para luchar contra la desinformación.

5.1.4. Preprocesamiento y construcción de las tres versiones del dataset

Con el texto de las noticias preprocesado y los tweets generados, se procedió a construir tres versiones del conjunto de datos, pensadas para evaluar diferentes escenarios de entrenamiento. Todas las versiones comparten las mismas instancias base (es decir, corresponden a las mismas noticias originales y conservan las etiquetas de verdad/falsedad), pero difieren en la fuente textual que proporcionan al modelo como entrada:

1. **Dataset de noticias (solo texto de noticias)**: en esta versión, cada ejemplo de entrenamiento consiste únicamente en el texto completo de la noticia original, junto con su etiqueta (0 o 1). Este conjunto permite entrenar modelos que intentan detectar si una noticia es falsa o verdadera analizando únicamente el contenido de la noticia en sí. Es el enfoque tradicional de detección de *fake news* mediante clasificación de artículos periodísticos.
2. **Dataset de tweets generados (solo tweets):** en esta variante, cada ejemplo consiste en los dos tweets sintéticos generados a partir de una noticia, acompañado de la misma etiqueta que tenía la noticia original. Este escenario evalúa si las señales presentes en textos breves (como el lenguaje empleado, el tono o ciertas palabras clave) son suficientes para determinar la veracidad de un texto.
3. **Dataset combinado (noticia + tweet):** esta versión fusiona ambos tipos de información. Para cada instancia, se proporciona al modelo tanto el texto de la noticia original como la combinación de sus tweets asociados como una sola secuencia de entrada, junto con la etiqueta correspondiente. El propósito de este conjunto combinado es permitir que el modelo aproveche dos niveles de contexto, pero también identificar si esta unión de ambos textos puediera generar ruido en la detección.

Antes de usar estos datos en los modelos, se realiza un preprocesamiento común. Todos los textos se convierten a minúsculas, se eliminan las stopwords y caracteres especiales como emoticonos, puntuación, expresiones entre corchetes, y se normalizan los espacios. También se eliminan duplicados y se revisan los tweets generados para filtrar aquellos que estén vacíos o sean incoherentes. Con estas tres variantes, se consigue observar cómo influye el formato textual, la longitud y el tono emocional en la eficacia del sistema de detección, lo que enriquece el análisis posterior.

5.2. Representación numérica del texto

Para entrenar modelos capaces de clasificar noticias como verdaderas o falsas, es necesario transformar el texto en una representación numérica que pueda ser procesada por los algoritmos. Esta sección detalla los distintos métodos de representación textual empleados en el proyecto, adaptados a cada tipo de modelo.

5.2.1. Vectorización clásica: TF-IDF

En los modelos de aprendizaje automático tradicionales se utiliza la técnica TF-IDF (Term Frequency-Inverse Document Frequency) como método de vectorización. Esta técnica pondera las palabras según su frecuencia en un documento y su rareza en el conjunto total de textos, permitiendo identificar términos relevantes sin dar demasiada importancia a palabras comunes. Su principal ventaja está en su simplicidad, eficiencia computacional y buena capacidad de generalización en textos más largos y estructurados, como es el caso de las noticias. La implementación se realiza mediante la clase TfidfVectorizer de scikit-learn, integrada en un pipeline de evaluación junto con los clasificadores, y optimizada mediante GridSearchCV.

5.2.2. Embeddings con FastText

Para textos más breves y menos estructurados como los tweets generados a partir de las noticias, se recurre al uso de FastText, un modelo de representación vectorial que tiene en cuenta no solo palabras completas sino también subpalabras. Esto lo hace especialmente eficaz en textos informales o ruidosos, donde pueden aparecer abreviaciones, errores ortográficos o términos nuevos. A diferencia de modelos como Word2Vec, FastText permite calcular representaciones para palabras no vistas en el entrenamiento, lo que mejora la robustez de los clasificadores. En este proyecto, se entrena un modelo FastText sobre los datos de entrenamiento y se utiliza la media de los vectores de cada palabra para representar cada documento antes de entrenar el clasificador correspondiente.

5.2.3. Tokenización y padding para modelos de secuencia

En el caso de los modelos de deep learning, se utiliza un enfoque basado en secuencias. Para ello, se aplica un proceso de tokenización con la herramienta Tokenizer de Keras, que convierte cada texto en una secuencia de enteros basada en un vocabulario limitado. Posteriormente, se aplica padding para asegurar que todas las secuencias tengan la misma longitud, permitiendo su procesamiento en redes neuronales como RNN, LSTM o BiLSTM. Este enfoque es especialmente útil para capturar patrones sintácticos y semánticos a lo largo de la secuencia, y se adapta bien a tareas de clasificación donde el orden y contexto de las palabras es importante.

5.3. Modelado y entrenamiento

Una vez preprocesado y preparado el conjunto de datos en sus tres versiones noticias puras, tweets generados y combinación de ambos, se procede al modelado y entrenamiento de distintos algoritmos de aprendizaje automático. El objetivo es analizar su rendimiento frente a la tarea de clasificación binaria (noticia real o falsa), comparando cómo afectan los distintos tipos de entrada textual al comportamiento de los modelos.

Para este propósito, se han entrenado cuatro modelos clásicos de machine learning: Multinomial Naïve-Bayes, Logistic Regression, Random Forest y Gradient Boosting. Cada uno de estos modelos ha sido entrenado tres veces, una por cada una de las variantes del dataset:

- Dataset con noticias exclusivamente.
- Dataset con tweets generados mediante LLM y RAG.
- Dataset con la combinación de ambos.

Para garantizar una evaluación robusta y coherente, el conjunto de datos se divide con una proporción del 80 % para entrenamiento, 10 % para validación y 10 % para test, asegurando la estratificación de clases. Esta división se mantiene constante a través de todos los modelos y variantes del dataset.

5.3.1. Preprocesamiento y evaluación

En todos los casos, se emplea una pipeline basada en TF-IDF para vectorizar el texto, seguido del modelo correspondiente. El texto se preprocesa utilizando las librerías nltk para eliminar signos de puntuación, tokenizar y filtrar stopwords del idioma inglés.

Para ajustar los hiperparámetros, se utiliza GridSearchCV con validación cruzada de 5 particiones (cv=5), maximizando la métrica F1-score macro, lo que permite ponderar equitativamente el rendimiento en ambas clases (real y falsa).

La evaluación se realiza sobre los tres conjuntos (entrenamiento, validación y test) y contempla las siguientes métricas: precisión, recall, F1-score, AUC-ROC y log-loss. Además, se generan automáticamente matrices de confusión y curvas de aprendizaje, lo que facilita un análisis visual del comportamiento de cada modelo.

El procedimiento se encuentra encapsulado en funciones reutilizables dentro del archivo eval_model.py, lo que favorece la reproducibilidad y consistencia de los experimentos.

5.3.2. Modelos de Machine Learning

Aquí, se describen los modelos clásicos de aprendizaje automático empleados para la clasificación de noticias falsas. Estos modelos han sido seleccionados por su equilibrio entre simplicidad, interpretabilidad y rendimiento. A continuación, se resumen brevemente sus principales características:

- **Multinomial Naïve-Bayes**: este modelo se basa en la aplicación del teorema de Bayes suponiendo independencia condicional entre las características. En el contexto de procesamiento de texto, se adapta especialmente bien a datos representados como conteos o frecuencias, como es el caso del TF-IDF. Su ventaja principal radica en su simplicidad y eficiencia computacional.
- **Logistic Regression**: la regresión logística aplica una función sigmoide para modelar la probabilidad de que una observación pertenezca a una clase. A pesar de su nombre, se trata de un modelo de clasificación binaria muy robusto y ampliamente utilizado como línea base en tareas de NLP por su interpretabilidad y buen rendimiento con datos linealmente separables.
- **Random Forest**: Random Forest es un modelo de ensamblado basado en múltiples árboles de decisión entrenados sobre subconjuntos aleatorios del conjunto de datos. Su capacidad para manejar datos no lineales y reducir el sobreajuste mediante bagging lo convierte en una opción sólida en contextos con ruido o alta dimensionalidad.
- **Gradient Boosting**: este modelo también se basa en árboles de decisión, pero los entrena secuencialmente, cada uno corrigiendo los errores del anterior. Su enfoque en la optimización por gradiente lo hace especialmente efectivo, aunque más sensible al sobreajuste si no se regula adecuadamente. Es ideal para capturar relaciones complejas en los datos.

5.3.3. Modelos de Deep Learning

El uso de redes neuronales profundas permite capturar patrones complejos y no lineales en los datos textuales. Estas arquitecturas están diseñadas para procesar secuencias y relaciones contextuales que escapan a los métodos tradicionales, resultando particularmente útiles en tareas como la detección de noticias falsas. A continuación, se describen los modelos evaluados en este trabajo:

- **MLP**: el Perceptrón Multicapa (MLP) representa una de las arquitecturas neuronales más simples, pero eficaces en problemas de clasificación binaria. Se trata de una red feedforward en la que la información fluye en una única dirección, desde la capa de entrada hasta la de salida, pasando por una o más capas ocultas completamente conectadas. En esta implementación, se emplea una capa de embedding seguida de una operación de max pooling global, una capa densa intermedia con activación ReLU y una capa de salida sigmoide.

- **RNN**: las Redes Neuronales Recurrentes introducen una capacidad de memoria que les permite procesar secuencias de texto considerando el orden de las palabras. A diferencia del MLP, las RNN procesan una palabra a la vez y actualizan un estado oculto que se transmite en cada paso. No obstante, sufren del problema del desvanecimiento del gradiente, que limita su capacidad para aprender dependencias a largo plazo.
- **LSTM**: las redes LSTM (Long Short-Term Memory) son una evolución de las RNN diseñadas para superar sus limitaciones. Gracias a su arquitectura con compuertas (de entrada, olvido y salida), las LSTM pueden retener información relevante durante más tiempo, lo cual resulta crucial en análisis de texto.
- **BiLSTM**: el modelo BiLSTM extiende las LSTM tradicionales al incorporar procesamiento bidireccional: una lectura del texto hacia adelante y otra hacia atrás. Esto mejora la capacidad de comprensión contextual, especialmente útil en tareas donde el significado depende de palabras futuras y pasadas.

5.3.4. Modelos basados en Transformers

Los modelos basados en Transformers han supuesto un avance significativo en el campo del procesamiento del lenguaje natural, al permitir una representación contextual rica de cada palabra dentro de un texto. A diferencia de las arquitecturas tradicionales, los Transformers procesan toda la secuencia simultáneamente y aplican mecanismos de atención que ponderan la importancia de cada token. En este trabajo se han evaluado las siguientes variantes:

- **BERT base uncased**: se entrena inicialmente en tareas genéricas y luego se ajusta mediante fine-tuning. Esta variante ignora las mayúsculas y permite un modelado contextual bidireccional muy eficaz en textos ambiguos o con matices.
- **DeBERTa v3 base**: introduce mejoras como la atención desacoplada y embeddings enriquecidos con contexto. Esto le permite capturar relaciones semánticas más profundas y adaptarse mejor a textos reales de redes sociales o noticias con lenguaje informal.

5.3.5. Reporte estadístico

Para evaluar el rendimiento de los modelos entrenados, se emplea un conjunto de métricas que permiten obtener una visión detallada de su capacidad para distinguir entre noticias falsas y reales, optando por un enfoque que analiza diversos aspectos del comportamiento del modelo en tareas de clasificación binaria.

En primer lugar, se calcula la precisión (precision), que mide qué proporción de los elementos clasificados como positivos son realmente positivos, resultando relevante en contextos donde los falsos positivos pueden tener consecuencias

indeseadas, como cuando una noticia verdadera se clasifica erróneamente como falsa, generando desconfianza injustificada.

La exhaustividad o recuperación (recall) evalúa cuántos de los verdaderos positivos han sido identificados correctamente por el modelo. En este trabajo, su valor resulta clave para entender cuántas noticias falsas logra detectar el sistema entre todas las que realmente lo son, lo cual es fundamental en tareas de detección automatizada de desinformación.

El F1-score representa una medida armónica entre precisión y recall, y resulta útil cuando se desea encontrar un equilibrio entre ambos. Dado que en la práctica puede haber ligeras descompensaciones entre clases o diferentes costes asociados a errores de tipo I y II, el F1-score ofrece una síntesis robusta para comparar modelos de forma objetiva.

Además, se incluye la métrica AUC (Área bajo la curva ROC), que analiza la capacidad del modelo para discriminar entre las dos clases a distintos umbrales de decisión. Esta métrica es especialmente valiosa cuando se quiere evaluar el comportamiento del modelo independientemente del umbral elegido, proporcionando una medida global de su capacidad predictiva.

Otra métrica incorporada es el log loss (o pérdida logarítmica), que mide el grado de error en las predicciones probabilísticas. Un log loss bajo indica que el modelo no solo acierta en sus predicciones, sino que lo hace con un nivel de confianza razonable. Esta métrica penaliza fuertemente las predicciones erróneas con alta probabilidad, por lo que es una herramienta útil para evaluar modelos en tareas sensibles al grado de certeza.

Para complementar estas métricas globales, se genera un reporte de clasificación detallado, que incluye los valores de precisión, recall y F1-score por separado para cada clase (noticias falsas y verdaderas). Este análisis por clase permite detectar posibles sesgos o desigualdades en el rendimiento, como una tendencia a clasificar mejor una clase en detrimento de la otra.

Asimismo, se presenta la matriz de confusión, que permite visualizar directamente el número de verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos. Esta matriz resulta muy ilustrativa para entender el tipo de errores que comete el modelo y orientar acciones correctivas si fuera necesario.

Más allá de esto, se incluye una curva de aprendizaje, que muestra la evolución del rendimiento del modelo en función del tamaño del conjunto de entrenamiento. Esta gráfica permite evaluar si el modelo se encuentra infraentrenado o sobreentrenado, y si podría beneficiarse de más datos o ajustes en su configuración. Además, permite analizar la estabilidad del aprendizaje y la capacidad de generalización del modelo en distintos escenarios. De esta manera, se proporciona una evaluación completa y rigurosa de los modelos desarrollados, permitiendo no solo compararlos entre sí, sino también identificar sus fortalezas y limitaciones.

Por último, se muestra en la Figura 5.7 un diagrama de flujo del desarrollo del proyecto

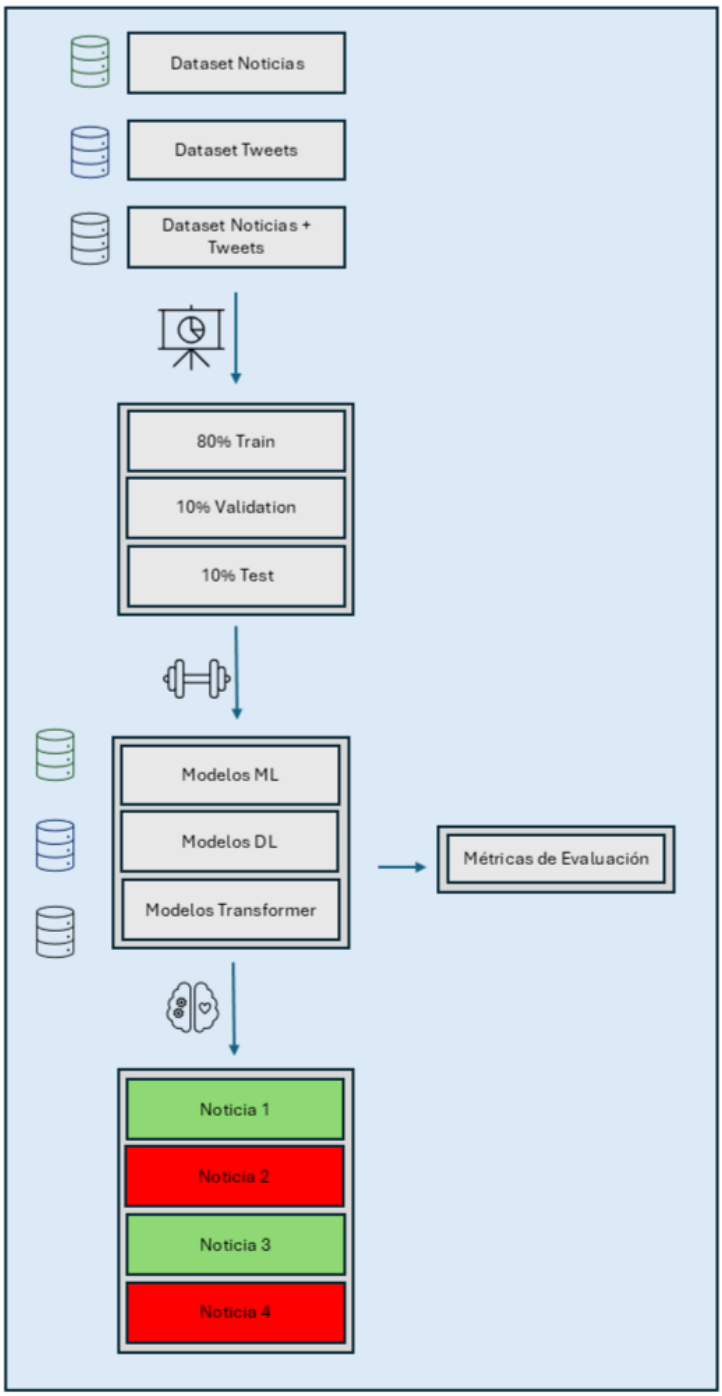

Figura 5.7: Diagrama de flujo del sistema

Capítulo 6

Evaluación

En este capítulo se realizará la presentación de resultados y métricas obtenidas por cada uno de los modelos con cada dataset creado. Las representaciones de la matriz de confusión y curva de aprendizaje de los modelos estarán accesibles en el anexo del trabajo.

6.1. Evaluación del Dataset con Texto

A lo largo de esta sección se presenta una evaluación profunda del rendimiento de los modelos de aprendizaje automático entrenados utilizando únicamente el texto original de las noticias.

6.1.1. Modelos de Machine Learning

Los modelos evaluados incluyen Multinomial Naïve-Bayes, Logistic Regression, Gradient Boosting y Random Forest. Para cada modelo se detallan las métricas de rendimiento sobre los diferentes subconjuntos.

Como se puede observar, las Tablas 6.1 y 6.2 muestran que los modelos de ensamblado (Gradient Boosting y Random Forest) alcanzan el mejor rendimiento general en test, con F1-score y AUC cercanos al 0.99 y una gran precisión por clase. Gradient Boosting, en particular, destaca por su menor log-loss, indicando una mejor calibración de las probabilidades. Logistic Regression también ofrece un rendimiento notable con métricas muy equilibradas entre clases y gran capacidad de generalización.

Por el contrario, Multinomial Naïve-Bayes, aunque computacionalmente eficiente, obtiene resultados claramente inferiores, especialmente en recall para la clase 0, lo que sugiere una mayor tasa de falsos negativos. En conjunto, los modelos más complejos superan a los más simples tanto en métricas globales como por clase, consolidando su idoneidad para tareas de clasificación de noticias falsas.

Tabla 6.1: Métricas globales en test para modelos de Machine Learning

Modelo	Precisión	Recall	F1-score	AUC	Log-loss
Multinomial Naïve-Bayes	0.8839	0.8642	0.8672	0.9604	0.4222
Logistic Regression	0.9831	0.9833	0.9832	0.9918	0.1212
Gradient Boosting	0.9903	0.9893	0.9897	0.9945	0.0773
Random Forest	0.9903	0.9893	0.9897	0.9953	0.2447

Tabla 6.2: Métricas por clase en test para modelos de Machine Learning

Modelo	Clase	Precisión	Recall	F1-score
Multinomial Naïve-Bayes	0	0.9423	0.7699	0.8474
	1	0.8256	0.9585	0.8871
Logistic Regression	0	0.9805	0.9839	0.9822
	1	0.9858	0.9828	0.9843
Gradient Boosting	0	0.9965	0.9816	0.9890
	1	0.9840	0.9970	0.9905
Random Forest	0	0.9965	0.9816	0.9890
	1	0.9840	0.9970	0.9905

6.1.2. Modelos de Deep Learning

Ahora, se presentan los resultados obtenidos con modelos de aprendizaje profundo aplicados al texto de las noticias. Se han evaluado cuatro arquitecturas: una red neuronal multicapa (MLP), una red neuronal recurrente (RNN), una red LSTM y su variante bidireccional (BiLSTM). Todas las redes fueron entrenadas bajo condiciones homogéneas para garantizar una comparación justa.

Se observa que los modelos de Deep Learning han ofrecido resultados verdaderamente prometedores. El MLP logró el mayor AUC, mientras que BiLSTM presentó el mejor balance entre precisión y recall por clase. Aunque la RNN es la más simple en cuanto a estructura, su rendimiento fue a la par del resto de modelos. Además, al igual que en los modelos de Machine Learning, los resultados son casi perfectos.

Tabla 6.3: Métricas globales en test para modelos de Deep Learning

Modelo	Precisión	Recall	F1-score	AUC
MLP	0.9812	0.9812	0.9812	0.9977
RNN	0.9704	0.9704	0.9704	0.9947
LSTM	0.9747	0.9747	0.9747	0.9965
BiLSTM	0.9779	0.9779	0.9779	0.9960

Tabla 6.4: Métricas por clase en test para modelos de Deep Learning

Modelo	Clase	Precisión	Recall	F1-score
MLP	0	0.9815	0.9781	0.9798
	1	0.9808	0.9838	0.9823
RNN	0	0.9711	0.9655	0.9683
	1	0.9698	0.9747	0.9722
LSTM	0	0.9724	0.9735	0.9730
	1	0.9767	0.9757	0.9762
BiLSTM	0	0.9770	0.9758	0.9764
	1	0.9788	0.9798	0.9793

Tabla 6.5: Tiempos de entrenamiento y rendimiento para modelos ML y DL

Modelo	Runtime (s)	Samples/s	Epochs
Machine Learning			
Multinomial Naïve-Bayes	51.94	285.87	-
Logistic Regression	72.72	204.18	-
Gradient Boosting	113.4	130.93	-
Random Forest	91.64	162.03	-
Deep Learning			
MLP	24.30	611.03	10
RNN	34.57	429.51	7
LSTM	61.2	242.61	5
BiLSTM	108.52	136.82	6

6.1.3. Modelos Transformers

Por último, se presentan los resultados obtenidos utilizando modelos basados en Transformers, concretamente BERT y DeBERTa. Ambos fueron ajustados con el mismo número de épocas y evaluados sobre test en un entorno con utilización de GPU por los altos tiempos de entrenamiento y ejecución en local.

Como se muestra en las Tablas 6.6 y 6.7, ambos modelos alcanzan un rendimiento prácticamente perfecto. DeBERTa obtiene un log-loss menor y una ligera ventaja en F1-score, mientras que BERT logra un mayor AUC (0.9998). Ambos presentan precisión y recall muy equilibrados y superiores al 99 %. Estas métricas reafirman que los Transformers son altamente eficaces para la detección de noticias falsas, superando incluso a modelos de aprendizaje profundo tradicionales.

Tabla 6.6: Métricas globales en test para modelos Transformers

Modelo	Precisión	Recall	F1-score	AUC	Log-loss	Accuracy
BERT	0.9980	0.9960	0.9970	0.9998	0.0281	0.9968
DeBERTa	0.9990	0.9960	0.9975	0.9997	0.0160	0.9973

Con estos resultados para el primer dataset vemos que los resultados evidencian una clara progresión en el rendimiento a medida que se incrementa la comple-

Tabla 6.7: Tiempos de entrenamiento y rendimiento para modelos Transformers

Modelo	Runtime (s)	Samples/s	Epochs
BERT	3345.868	22.189	5
DeBERTa	2754.2664	26.95	5

jidad y capacidad de los modelos. Los modelos de Machine Learning ofrecen resultados aceptables, especialmente los algoritmos de ensamblado como Gradient Boosting y Random Forest. Sin embargo, los modelos de Deep Learning superan sus métricas al captar mejor las dependencias en el texto. Finalmente, los Transformers (BERT y DeBERTa) destacan por su rendimiento casi perfecto, logrando los valores más altos de AUC, precisión y F1-score. Esta superioridad se debe a su arquitectura basada en atención, que permite capturar relaciones contextuales complejas en el lenguaje natural con mayor profundidad que los modelos anteriores.

6.2. Evaluación del Dataset con solo Tweets

Seguidamente, se analiza el rendimiento de los distintos modelos de clasificación entrenados exclusivamente con el dataset de tweets. Dada la brevedad y naturaleza del texto, se opta por aplicar FastText como técnica de vectorización para algunos modelos de machine learning aparte de utilizar el vectorizador tradicional, lo que ha contribuido a mejorar significativamente los resultados en comparación con técnicas como TF-IDF.

6.2.1. Modelos de Machine Learning

De nuevo, los modelos evaluados incluyen Multinomial Naïve-Bayes, Logistic Regression, Gradient Boosting y Random Forest. Todos han sido entrenados con validación cruzada (5 folds) y han utilizado representaciones TF-IDF. Además, se incluyen versiones de estos modelos que emplean FastText, mostrando una mejora notable en todos los indicadores.

Como se observa en las Tablas 6.8 y 6.9, los modelos que incorporan FastText superan ampliamente a sus equivalentes con TF-IDF. En particular, Gradient Boosting y Logistic Regression logran los mejores resultados, con un F1-score cercano a 0.85 y un AUC superior a 0.91, se vuelve a repetir que estos dos modelos ensamblados superan al resto de modelos de machine learning. Por otro lado, Multinomial Naïve-Bayes ofrece un rendimiento inferior, especialmente en el recall de la clase 0.

Tabla 6.8: Métricas globales en test para modelos ML con y sin FastText

Modelo	Precisión	Recall	F1-score	AUC	Log-loss
Multinomial Naïve-Bayes	0.6408	0.6341	0.6331	0.6998	0.6315
Logistic Regression	0.6433	0.6434	0.6433	0.7041	0.6250
Gradient Boosting	0.6543	0.6507	0.6508	0.7163	0.6153
Random Forest	0.6540	0.6410	0.6382	0.7075	0.6352
Logistic Regression + FastText	0.8325	0.8321	0.8323	0.9151	0.3736
Gradient Boosting + FastText	0.8479	0.8485	0.8482	0.9238	0.3521
Random Forest + FastText	0.8453	0.8464	0.8456	0.9181	0.4026

Tabla 6.9: Métricas por clase en test para modelos ML con y sin FastText

Modelo	Clase	Precisión	Recall	F1-score
Multinomial Naïve-Bayes	0	0.6407	0.5293	0.5797
	1	0.6409	0.7389	0.6864
Logistic Regression	0	0.6192	0.6249	0.6220
	1	0.6674	0.6619	0.6646
Gradient Boosting	0	0.6477	0.5777	0.6107
	1	0.6608	0.7237	0.6908
Random Forest	0	0.6677	0.5017	0.5729
	1	0.6404	0.7804	0.7035
Logistic Regression + FastText	0	0.8246	0.8170	0.8208
	1	0.8404	0.8472	0.8438
Gradient Boosting + FastText	0	0.8333	0.8458	0.8395
	1	0.8626	0.8512	0.8569
Random Forest + FastText	0	0.8242	0.8527	0.8382
	1	0.8664	0.8401	0.8530

6.2.2. Modelos de Deep Learning

Los modelos de Deep Learning aplicados sobre tweets incluyen una red neuronal multicapa (MLP), una red RNN, una red LSTM y una variante bidireccional (BiLSTM). Como se observa en las Tablas 6.10 y 6.11, todos ellos superan el 87 % de precisión, con AUCs en torno al 0.94. El modelo LSTM obtiene el mejor equilibrio entre todas las métricas.

Tabla 6.10: Métricas globales en test para modelos DL

Modelo	Precisión	Recall	F1-score	AUC
MLP	0.8756	0.8756	0.8757	0.9436
RNN	0.8745	0.8745	0.8747	0.9345
LSTM	0.8810	0.8810	0.8810	0.9451
BiLSTM	0.8724	0.8724	0.8725	0.9388

Tabla 6.11: Métricas por clase en test para modelos DL

Modelo	Clase	Precisión	Recall	F1-score
MLP	0	0.8609	0.8757	0.8682
	1	0.8890	0.8755	0.8822
RNN	0	0.8479	0.8918	0.8693
	1	0.9003	0.8593	0.8793
LSTM	0	0.8707	0.8757	0.8732
	1	0.8901	0.8856	0.8879
BiLSTM	0	0.8496	0.8838	0.8663
	1	0.8940	0.8623	0.8779

Tabla 6.12: Tiempos de entrenamiento y rendimiento para modelos ML y DL

Modelo	Runtime (s)	Samples/s	Epochs
Machine Learning			
Multinomial Naïve-Bayes	10.29	1442.95	-
Logistic Regression	9.35	1588.02	-
Logistic Regression - FT	20.62	90.08	-
Gradient Boosting	19.33	768.13	-
Gradient Boosting - FT	66.4	27.97	-
Random Forest	16.6	894.46	-
Random Forest - FT	21.21	87.55	-
Deep Learning			
MLP	15.4	964.16	6
RNN	25.35	585.72	5
LSTM	73.5	202.01	6
BiLSTM	112.18	132.36	6

6.2.3. Modelos Transformers

Finalmente, se presentan los resultados de los modelos Transformers ajustados con el dataset de tweets. Se utilizaron BERT y DeBERTa durante 5 épocas. Los resultados, detallados en las Tablas 6.13 y 6.14, muestran que DeBERTa supera ligeramente a BERT en todas las métricas salvo en precisión.

Tabla 6.13: Métricas globales en test para modelos Transformers

Modelo	Precisión	Recall	F1-score	AUC	Log-loss	Accuracy
BERT	0.8858	0.8723	0.8790	0.9401	0.3197	0.8722
DeBERTa	0.8700	0.9422	0.9047	0.9610	0.3533	0.8943

En conclusión, los resultados en el dataset de tweets refuerzan las ventajas del uso de técnicas de representación enriquecidas como FastText, especialmente en textos cortos. Asimismo, los modelos Transformers consolidan su superioridad en clasificación de lenguaje natural, ofreciendo el mejor rendimiento general.

Tabla 6.14: Tiempos de entrenamiento y rendimiento para modelos Transformers

Modelo	Runtime (s)	Samples/s	Epochs
BERT	2715.7421	27.315	5
DeBERTa	1925.9645	380516	5

6.3. Evaluación del Dataset Combinado (Texto + Tweets)

Por último, se evalúa el rendimiento de los modelos de clasificación entrenados utilizando tanto el texto original de las noticias como los tweets relacionados. La combinación de ambas fuentes aporta riqueza contextual y mejora la capacidad de los modelos para identificar patrones relacionados con la veracidad de la información.

6.3.1. Modelos de Machine Learning

Como se observa en las Tablas 6.15 y 6.16, el rendimiento general es verdaderamente bueno, especialmente para Gradient Boosting y Random Forest, con F1-score superiores a 0.98 y AUC cercanos al 0.995. Logistic Regression también ofrece un excelente desempeño, mientras que Multinomial Naïve-Bayes, aunque con resultados aceptables, queda ligeramente por debajo.

Tabla 6.15: Métricas globales en test para modelos ML

Modelo	Precisión	Recall	F1-score	AUC	Log-loss
Multinomial Naïve-Bayes	0.8757	0.8648	0.8671	0.9506	0.4274
Logistic Regression	0.9828	0.9826	0.9827	0.9909	0.1256
Gradient Boosting	0.9903	0.9893	0.9897	0.9944	0.0768
Random Forest	0.9903	0.9893	0.9897	0.9953	0.2678

Tabla 6.16: Métricas por clase en test para modelos ML

Modelo	Clase	Precisión	Recall	F1-score
Multinomial Naïve-Bayes	0	0.9118	0.7975	0.8508
	1	0.8396	0.9322	0.8834
Logistic Regression	0	0.9827	0.9804	0.9816
	1	0.9828	0.9848	0.9838
Gradient Boosting	0	0.9965	0.9816	0.9890
	1	0.9840	0.9970	0.9905
Random Forest	0	0.9965	0.9816	0.9890
	1	0.9840	0.9970	0.9905

6.3.2. Modelos de Deep Learning

En cuanto a los modelos de aprendizaje profundo, se muestra en las tablas 6.17 y 6.18 que el modelo MLP es el más preciso, con AUC de 0.9977 y F1-score promedio de 0.9806, seguido muy de cerca por LSTM y BiLSTM. La red RNN, aunque estructuralmente más simple, mantiene métricas aceptables por encima del 86 %.

Tabla 6.17: Métricas globales en test para modelos DL

Modelo	Precisión	Recall	F1-score	AUC
MLP	0.9806	0.9806	0.9806	0.9977
RNN	0.8638	0.8638	0.8638	0.9334
LSTM	0.9677	0.9677	0.9677	0.9867
BiLSTM	0.9618	0.9618	0.9618	0.9883

Tabla 6.18: Métricas por clase en test para modelos DL

Modelo	Clase	Precisión	Recall	F1-score
MLP	0	0.9760	0.9827	0.9794
	1	0.9847	0.9787	0.9817
RNN	0	0.8508	0.8596	0.8552
	1	0.8754	0.8674	0.8714
LSTM	0	0.9623	0.9689	0.9656
	1	0.9725	0.9666	0.9695
BiLSTM	0	0.9404	0.9804	0.9600
	1	0.9821	0.9453	0.9634

Tabla 6.19: Tiempos de entrenamiento y rendimiento para modelos ML y DL

Modelo	Runtime (s)	Samples/s	Epochs
Machine Learning			
Multinomial Naïve-Bayes	51.72	287.08	-
Logistic Regression	67.38	220.36	-
Gradient Boosting	108.68	136.52	-
Random Forest	104.52	142.06	-
Deep Learning			
MLP	22.05	673.38	10
RNN	24.93	595.59	5
LSTM	94.07	157.84	8
BiLSTM	151.14	98.24	8

6.3.3. Modelos Transformers

Finalmente, se han ajustado BERT y DeBERTa. Ambos ofrecen un rendimiento prácticamente perfecto, con F1-score superiores al 99 % y AUC cercanos a 1.0. En este caso, DeBERTa vuelve a destacar ligeramente sobre el otro modelo de transformers por su menor log-loss y mayor precisión.

Tabla 6.20: Métricas globales en test para modelos Transformers

Modelo	Precisión	Recall	F1-score	AUC	Log-loss	Accuracy
BERT	0.9949	0.9960	0.9954	0.9997	0.0370	0.9952
DeBERTa	0.9980	0.9949	0.9965	0.9994	0.0266	0.9962

Tabla 6.21: Tiempos de entrenamiento y rendimiento para modelos Transformers

Modelo	Runtime (s)	Samples/s	Epochs
BERT	3209.8265	23.129	5
DeBERTa	2581.4584	28.759	5

Como conclusión, la fusión de las fuentes textuales ha permitido a los modelos alcanzar un nivel de rendimiento algo superior a cuando simplemente utilizamos el texto de la noticia y más notable cuando lo hacemos sobre el texto breve de los tweets. Los modelos de aprendizaje profundo y, especialmente, los Transformers, se benefician claramente de la riqueza semántica combinada, logrando métricas casi perfectas en todas las evaluaciones.

6.4. Resultados de Clasificación

Una vez comprobada la evaluación de los modelos, se presentan ejemplos concretos de predicciones realizadas por los modelos de Machine Learning, Deep Learning y Transformers sobre cuatro noticias, dos verdaderas y dos falsas. La estructura se organiza por tipo de dataset y cada noticia se acompaña de su etiqueta real, el texto original en inglés y una tabla comparativa de predicciones y niveles de confianza para los distintos modelos.

6.4.1. Dataset: Texto

Noticia 1 — Real (1)

Text: *London (Reuters) — Donald Trump told Britons on Sunday he supported Brexit, repeating just days before the vote that he thinks the UK would be better off outside the European Union. The campaign resumed after a three-day pause following the killing of MP Jo Cox. Trump, the presumptive Republican U.S. presidential nominee, said in a newspaper interview he was backing an out vote.*

Tabla 6.22: Predicciones del modelo para Noticia 1

Model	Prediction	Confidence (FAKE / REAL)
Multinomial Naïve-Bayes	REAL (1)	0.2264 / 0.7736
Logistic Regression	REAL (1)	0.0000 / 1.0000
Random Forest	REAL (1)	0.2093 / 0.7907
Gradient Boosting	REAL (1)	0.0442 / 0.9558
MLP	REAL (1)	0.0005 / 0.9995
RNN	REAL (1)	0.0549 / 0.9451
LSTM	REAL (1)	0.1925 / 0.8075
BiLSTM	REAL (1)	0.1570 / 0.8430
BERT	REAL (1)	0.0000 / 1.0000
DeBERTa	REAL (1)	0.0000 / 1.0000

Noticia 2 — Fake (0)

Text: *All we can say about this development is that it's about time the GOP congressman did something about the wild goose chase Mueller and his team have been going on. Nineteen Republican congressmen signed a letter on Friday requesting congressional hearings to hold Mueller's team accountable.*

Tabla 6.23: Predicciones del modelo para Noticia 2

Model	Prediction	Confidence (FAKE / REAL)
Multinomial Naïve-Bayes	FAKE (0)	0.5049 / 0.4951
Logistic Regression	FAKE (0)	0.9944 / 0.0056
Random Forest	FAKE (0)	0.7783 / 0.2217
Gradient Boosting	FAKE (0)	0.9564 / 0.0436
MLP	FAKE (0)	0.9992 / 0.0008
RNN	FAKE (0)	0.9495 / 0.0505
LSTM	FAKE (0)	0.9788 / 0.0212
BiLSTM	FAKE (0)	0.9871 / 0.0129
BERT	FAKE (0)	0.9998 / 0.0002
DeBERTa	FAKE (0)	0.9924 / 0.0076

Noticia 3 — Real (1)

Text: *Washington (Reuters) — U.S. President Donald Trump and British Prime Minister Theresa May will speak in a call scheduled for Tuesday morning, the White House said. The White House gave no further details about the call, which comes after the two leaders met in Washington last month.*

Tabla 6.24: Predicciones del modelo para Noticia 3

Model	Prediction	Confidence (FAKE / REAL)
Multinomial Naïve-Bayes	REAL (1)	0.0922 / 0.9078
Logistic Regression	REAL (1)	0.0000 / 1.0000
Random Forest	REAL (1)	0.1968 / 0.8032
Gradient Boosting	REAL (1)	0.0442 / 0.9558
MLP	REAL (1)	0.0004 / 0.9996
RNN	REAL (1)	0.0087 / 0.9913
LSTM	REAL (1)	0.0324 / 0.9676
BiLSTM	REAL (1)	0.0090 / 0.9910
BERT	REAL (1)	0.0000 / 1.0000
DeBERTa	REAL (1)	0.0000 / 1.0000

Noticia 4 — Fake (0)

Text: *The NY Daily News is never shy when it comes to their covers. This time, they condemned the act of terror against the LGBT community at Pulse nightclub in Orlando. The newspaper harshly criticized the NRA, blaming them for resisting tighter gun control laws.*

Tabla 6.25: Predicciones del modelo para Noticia 4

Model	Prediction	Confidence (FAKE / REAL)
Multinomial Naïve-Bayes	REAL (1)	0.4680 / 0.5320
Logistic Regression	FAKE (0)	0.9996 / 0.0004
Random Forest	FAKE (0)	0.7794 / 0.2206
Gradient Boosting	FAKE (0)	0.9564 / 0.0436
MLP	FAKE (0)	0.9966 / 0.0034
RNN	FAKE (0)	0.9844 / 0.0156
LSTM	FAKE (0)	0.9613 / 0.0387
BiLSTM	FAKE (0)	0.9624 / 0.0376
BERT	FAKE (0)	1.0000 / 0.0000
DeBERTa	FAKE (0)	0.9950 / 0.0050

6.4.2. Dataset: Solo Tweets

Tweet 1 — Real (1)

Text: *Trump joins Brexit discussion, saying the UK would be better off outside the EU... right after Jo Cox was killed. Zero awareness. #brexit #trump*

Tabla 6.26: Predicciones del modelo para Tweet 1

Model	Prediction	Confidence (FAKE / REAL)
Multinomial Naïve-Bayes	REAL (1)	0.4247 / 0.5753
Logistic Regression	REAL (1)	0.4023 / 0.5977
Random Forest	REAL (1)	0.3722 / 0.6278
Gradient Boosting	REAL (1)	0.3859 / 0.6141
Logistic Regression + FastText	REAL (1)	0.0317 / 0.9683
Random Forest + FastText	REAL (1)	0.1930 / 0.8070
Gradient Boosting + FastText	REAL (1)	0.0841 / 0.9159
MLP	REAL (1)	0.1104 / 0.8896
RNN	REAL (1)	0.0877 / 0.9123
LSTM	REAL (1)	0.2734 / 0.7266
BiLSTM	REAL (1)	0.2734 / 0.7266
BERT	REAL (1)	0.1798 / 0.8202
DeBERTa	REAL (1)	0.0054 / 0.9946

Tweet 2 — Fake (0)

Text: *19 GOP congressmen want to shut down Mueller, hiding behind unlimited power. Scared they might get found out. #mueller #gopdrama*

Tabla 6.27: Predicciones del modelo para Tweet 2

Model	Prediction	Confidence (FAKE / REAL)
Multinomial Naïve-Bayes	FAKE (0)	0.5278 / 0.4722
Logistic Regression	FAKE (0)	0.6043 / 0.3957
Random Forest	REAL (1)	0.4801 / 0.5199
Gradient Boosting	FAKE (0)	0.5637 / 0.4363
Logistic Regression + FastText	FAKE (0)	0.6500 / 0.3500
Random Forest + FastText	REAL (1)	0.4152 / 0.5848
Gradient Boosting + FastText	FAKE (0)	0.5037 / 0.4963
MLP	REAL (1)	0.2919 / 0.7081
RNN	REAL (1)	0.3618 / 0.6382
LSTM	FAKE (0)	0.6586 / 0.3414
BiLSTM	FAKE (0)	0.8163 / 0.1837
BERT	REAL (1)	0.4067 / 0.5933
DeBERTa	REAL (1)	0.0230 / 0.9770

Tweet 3 — Real (1)

Text: *Trump and Theresa May are about to have a private call. Nothing good ever comes from those. Calls, chaos and diplomacy don't mix. #trump #ukpolitics*

Tabla 6.28: Predicciones del modelo para Tweet 3

Model	Prediction	Confidence (FAKE / REAL)
Multinomial Naïve-Bayes	REAL (1)	0.4490 / 0.5510
Logistic Regression	FAKE (0)	0.5142 / 0.4858
Random Forest	REAL (1)	0.4915 / 0.5085
Gradient Boosting	FAKE (0)	0.5162 / 0.4838
Logistic Regression + FastText	REAL (1)	0.0094 / 0.9906
Random Forest + FastText	REAL (1)	0.1280 / 0.8720
Gradient Boosting + FastText	REAL (1)	0.0567 / 0.9433
MLP	REAL (1)	0.0464 / 0.9536
RNN	REAL (1)	0.0410 / 0.9590
LSTM	REAL (1)	0.0929 / 0.9071
BiLSTM	REAL (1)	0.4592 / 0.5408
BERT	REAL (1)	0.0676 / 0.9324
DeBERTa	REAL (1)	0.0031 / 0.9969

Tweet 4 — Fake (0)

Text: *NY Daily News slams the NRA after the Pulse nightclub attack. Defending the NRA is disgusting. #guncontrolnow #orlando*

Tabla 6.29: Predicciones del modelo para Tweet 4

Model	Prediction	Confidence (FAKE / REAL)
Multinomial Naïve-Bayes	REAL (1)	0.3420 / 0.6580
Logistic Regression	REAL (1)	0.3250 / 0.6750
Random Forest	REAL (1)	0.4476 / 0.5524
Gradient Boosting	REAL (1)	0.3941 / 0.6059
Logistic Regression + FastText	FAKE (0)	0.9399 / 0.0601
Random Forest + FastText	FAKE (0)	0.7959 / 0.2041
Gradient Boosting + FastText	FAKE (0)	0.9205 / 0.0795
MLP	FAKE (0)	0.8992 / 0.1008
RNN	FAKE (0)	0.8389 / 0.1611
LSTM	FAKE (0)	0.9535 / 0.0465
BiLSTM	FAKE (0)	0.9159 / 0.0841
BERT	FAKE (0)	0.9153 / 0.0847
DeBERTa	FAKE (0)	0.9967 / 0.0033

6.4.3. Dataset: Texto + Tweet

Noticia 1 — Real (1)

Text: *London (Reuters) — Donald Trump told Britons on Sunday he supported Brexit, repeating just days before the vote that he thinks the UK would be better off outside the European Union. The campaign resumed after a three-day pause following the killing of MP Jo Cox. Trump, the presumptive Republican U.S. presidential nominee, said in a newspaper interview he was backing an out vote.*

Tabla 6.30: Predicciones del modelo para Noticia 1

Model	Prediction	Confidence (FAKE / REAL)
Multinomial Naïve-Bayes	REAL (1)	0.3763 / 0.6237
Logistic Regression	REAL (1)	0.0004 / 0.9996
Random Forest	REAL (1)	0.2991 / 0.7009
Gradient Boosting	REAL (1)	0.0437 / 0.9563
MLP	REAL (1)	0.0005 / 0.9995
RNN	REAL (1)	0.0445 / 0.9555
LSTM	REAL (1)	0.0292 / 0.9708
BiLSTM	REAL (1)	0.0112 / 0.9888
BERT	REAL (1)	0.0000 / 1.0000
DeBERTa	REAL (1)	0.0000 / 1.0000

Noticia 2 — Fake (0)

Text: *All we can say about this development is that it's about time the GOP congressman did something about the wild goose chase Mueller and his team have been going on. Nineteen Republican congressmen signed a letter on Friday requesting congressional hearings to hold Mueller's team accountable.*

Tabla 6.31: Predicciones del modelo para Noticia 2

Model	Prediction	Confidence (FAKE / REAL)
Multinomial Naïve-Bayes	FAKE (0)	0.5038 / 0.4962
Logistic Regression	FAKE (0)	0.9550 / 0.0450
Random Forest	FAKE (0)	0.7206 / 0.2794
Gradient Boosting	FAKE (0)	0.9550 / 0.0450
MLP	FAKE (0)	0.9924 / 0.0076
RNN	REAL (1)	0.4254 / 0.5746
LSTM	REAL (1)	0.0418 / 0.9582
BiLSTM	FAKE (0)	0.9962 / 0.0038
BERT	FAKE (0)	1.0000 / 0.0000
DeBERTa	FAKE (0)	0.5741 / 0.4259

Noticia 3 — Real (1)

Text: *Washington (Reuters) — U.S. President Donald Trump and British Prime Minister Theresa May will speak in a call scheduled for Tuesday morning, the White House said. The White House gave no further details about the call, which comes after the two leaders met in Washington last month.*

Tabla 6.32: Predicciones del modelo para Noticia 3

Model	Prediction	Confidence (FAKE / REAL)
Multinomial Naïve-Bayes	REAL (1)	0.2651 / 0.7349
Logistic Regression	REAL (1)	0.0001 / 0.9999
Random Forest	REAL (1)	0.1875 / 0.8125
Gradient Boosting	REAL (1)	0.0437 / 0.9563
MLP	REAL (1)	0.0001 / 0.9999
RNN	REAL (1)	0.0891 / 0.9109
LSTM	REAL (1)	0.0200 / 0.9800
BiLSTM	REAL (1)	0.0057 / 0.9943
BERT	REAL (1)	0.0000 / 1.0000
DeBERTa	REAL (1)	0.0000 / 1.0000

Noticia 4 — Fake (0)

Text: *The NY Daily News is never shy when it comes to their covers. This time, they condemned the act of terror against the LGBT community at Pulse nightclub in Orlando. The newspaper harshly criticized the NRA, blaming them for resisting tighter gun control laws.*

Tabla 6.33: Predicciones del modelo para Noticia 4

Model	Prediction	Confidence (FAKE / REAL)
Multinomial Naïve-Bayes	FAKE (0)	0.5790 / 0.4210
Logistic Regression	FAKE (0)	0.9989 / 0.0011
Random Forest	FAKE (0)	0.7403 / 0.2597
Gradient Boosting	FAKE (0)	0.9564 / 0.0436
MLP	FAKE (0)	0.9730 / 0.0270
RNN	REAL (1)	0.3848 / 0.6152
LSTM	FAKE (0)	0.9942 / 0.0058
BiLSTM	FAKE (0)	0.9969 / 0.0031
BERT	FAKE (0)	1.0000 / 0.0000
DeBERTa	FAKE (0)	0.9843 / 0.0157

6.5. Explicabilidad de los Resultados

Para comprender mejor el funcionamiento interno de los modelos de clasificación y aumentar su transparencia, se ha realizado un análisis de explicabilidad sobre las cuatro noticias, base de la predicción de los modelos. Este análisis se ha centrado en identificar las palabras más influyentes en cada predicción, utilizando para ello diferentes técnicas en función del tipo de modelo.

- Para los modelos de **Machine Learning clásico** (Multinomial Naïve-Bayes y Regresión Logística), se han utilizado los coeficientes del modelo entrenado para determinar qué tokens contribuyen más positivamente o negativamente a la predicción.
- Para los modelos de **Deep Learning** (MLP, RNN, LSTM y BiLSTM), se ha aplicado una técnica de atribución de relevancia basada en la propagación de gradientes, permitiendo estimar la influencia relativa de cada palabra.
- En el caso de **Transformers** (BERT y DeBERTa), que no permiten extraer fácilmente palabras clave con la arquitectura utilizada, se ha evaluado la confianza del modelo en cada predicción como aproximación interpretativa.

Dado que el comportamiento de los modelos puede variar significativamente según el tipo de entrada, el análisis se ha dividido por dataset.

Dataset: Solo Texto

En las noticias completas, los modelos clásicos tienden a asignar pesos positivos a términos periodísticos o institucionales como *"reuters"* (coeficiente de 4.50 en Naïve-Bayes y 61.71 en LogReg) y *"said"*, y pesos negativos a términos más ambiguos como *"like"*, *"trump"* o *"donald"*. Por ejemplo, en una noticia real, el modelo Naïve-Bayes asoció *"reuters"* y *"said"* con la clase REAL, mientras penalizaba *"trump"* y *"donald"* como indicadores de contenido falso.

Dataset: Solo tweets

En este conjunto, el lenguaje informal y breve genera un patrón diferente. Se observa que palabras como *"cox"*, *"brexit"*, *"says"* o *"good"* se asocian con contenido real, mientras que términos como *"zero"*, *"scared"*, *"shut"* o *"mueller"* aparecen frecuentemente en predicciones falsas. Los modelos de Deep Learning también destacan términos emocionales o polémicos como *"jumping"* o *"lot"*. Esta sensibilidad al tono o contexto emocional refuerza la necesidad de considerar la naturaleza del lenguaje usado en redes sociales.

Dataset: Texto + Tweets

En el dataset combinado, la explicabilidad revela una mayor riqueza semántica. En las noticias reales, palabras como *"minister"*, *"theresa"*, *"diplomacy"* o *"british"* refuerzan la predicción positiva, mientras que palabras como *"zero"*, *"cox"* o *"awareness"* se relacionan con predicciones falsas. Los modelos MLP y BiLSTM muestran coherencia con los clásicos al destacar términos como *"brexit"* y *"eu"*

como indicadores positivos. Además, en el caso de noticias falsas, las palabras clave destacadas suelen tener connotaciones sensacionalistas o polémicas (*"despicable"*, *"hiding"*, *"guncontrolnow"*).

Comparación general

Se demuestar que la explicabilidad varía no solo por arquitectura del modelo, sino también por el tipo de entrada. Los modelos clásicos permiten una trazabilidad clara a través de coeficientes de tokens, mientras que los modelos neuronales ofrecen una visión más contextual e implícita. No obstante, existe una tendencia común: los modelos tienden a identificar como indicios de veracidad aquellos términos asociados a fuentes formales, figuras institucionales o contenido neutral, mientras que los términos con carga emocional o polémica se vinculan con mayor frecuencia a la desinformación.

6.6. Discusión de los Resultados

A partir de los resultados obtenidos en las evaluaciones realizadas, se establece claramente la influencia del tipo de modelo y ddatos en el rendimiento para la detección automática de noticias falsas. A continuación, se presenta una discusión estructurada y comparativa que analiza tanto el comportamiento interno de cada grupo de modelos (Machine Learning, Deep Learning y Transformers) como las diferencias observadas entre los datasets empleados.

Inicialmente, al analizar los modelos clásicos de Machine Learning entrenados con el dataset de texto completo, se destaca la notable superioridad de los métodos de ensamblado (Gradient Boosting y Random Forest), ambos alcanzando métricas excepcionalmente altas, con valores de F1-score y AUC cercanos al 0.99. La Logistic Regression, aunque ligeramente inferior, también muestra una excelente generalización, evidenciada por un equilibrio notable entre precisión y recall para ambas clases. Por el contrario, Multinomial Naïve-Bayes presenta limitaciones considerables, especialmente en recall para la clase negativa (falsas noticias), indicando una vulnerabilidad a falsos negativos, probablemente derivada de la simplicidad del modelo y la independencia asumida entre características.

Al contrastar estos resultados con los obtenidos con Deep Learning sobre el mismo dataset, se observa una leve mejora en la capacidad predictiva global. En particular, el modelo MLP, aunque estructuralmente más sencillo que los recurrentes, logra métricas muy próximas a la perfección (AUC de 0.9977), lo que sugiere que, en textos completos, las dependencias contextuales inmediatas son adecuadamente capturadas por redes menos complejas. Los modelos recurrentes (RNN, LSTM, BiLSTM), aunque también presentan resultados excelentes, no muestran mejoras sustanciales respecto al MLP, lo que podría indicar que la longitud y estructura relativamente homogénea de las noticias favorecen modelos con arquitecturas más simples.

La evaluación con modelos basados en Transformers (BERT y DeBERTa) revela un desempeño aún más superior, consolidando la clara ventaja de estas arquitecturas avanzadas basadas en atención. Ambos modelos alcanzan resultados prácticamente perfectos, con DeBERTa ligeramente superior en términos de F1-score (0.9975 frente a 0.9970 de BERT), aunque BERT obtiene un valor marginalmente superior en AUC. La precisión casi absoluta de los Transformers en textos completos confirma su idoneidad para tareas de clasificación de lenguaje natural con alta complejidad contextual.

Al considerar exclusivamente el dataset compuesto por tweets, se produce una notable caída en el rendimiento de los modelos, especialmente en aquellos basados únicamente en técnicas tradicionales como TF-IDF. Esto se explica por la naturaleza breve, informal y ambigua del lenguaje empleado en redes sociales. La incorporación de la técnica de representación vectorial enriquecida (FastText) provoca una gran mejora en todos los modelos clásicos, destacando nuevamente Gradient Boosting y Logistic Regression por su robustez. Por otro lado, Multinomial Naïve-Bayes sigue mostrando una insuficiencia marcada en este tipo de contextos más informales.

Ante esto, los modelos de Deep Learning mantienen un rendimiento elevado sobre tweets, con el modelo LSTM mostrando el mejor equilibrio general (AUC de 0.9451). Este resultado refleja la capacidad de los modelos recurrentes para capturar dependencias semánticas y sintácticas sutiles en textos cortos. Transformers, aunque ligeramente inferiores en comparación con los textos completos, siguen mostrando una notable eficacia, especialmente DeBERTa, que logra el mejor rendimiento general entre todos los modelos aplicados a tweets, con un F1-score de 0.9047 y AUC de 0.9610.

Finalmente, en el dataset combinado (texto completo y tweets), se observa que la unión de ambos formatos textuales permite recuperar e incluso mejorar el desempeño global observado en el dataset original. Aquí, nuevamente, los modelos clásicos de ensamblado y Logistic Regression alcanzan excelentes métricas, siendo Gradient Boosting y Random Forest casi indistinguibles en rendimiento, con un F1-score y AUC próximos al 0.995. La evidente mejora en los modelos clásicos sugiere que la combinación de fuentes textuales diversificadas favorece notablemente la detección de patrones complejos en los datos, es decir, cuanto mas datos tengan los modelos y se preserve la calidad de los mismos, se desenvolverán mejor y presentarán resultados superiores.

La evaluación de los modelos de Deep Learning con el dataset combinado muestra una mayor dispersión entre resultados. El modelo MLP obtiene el mejor rendimiento global, superando notablemente a modelos más complejos como RNN y BiLSTM, lo que podría sugerir que la combinación de textos largos y breves favorece a redes con estructuras más simples capaces de generalizar mejor.

Los modelos Transformers, sin embargo, vuelven a demostrar una superioridad clara en términos absolutos, acercándose a valores prácticamente perfectos tanto en F1-score como en AUC, especialmente DeBERTa (F1-score de 0.9965 y log-loss de 0.0266), que destaca por su mejor calibración probabilística.

En pocas palabras, se observa claramente la superioridad de los modelos basados en Transformers en tareas de detección de noticias falsas, especialmente cuando se enfrentan a la combinación de formatos textuales. Asimismo, se evidencia que modelos clásicos robustos, apoyados por representaciones enriquecidas como FastText, son altamente efectivos en contextos específicos como tweets y textos similares.

Capítulo 7

Conclusiones y Trabajo Futuro

7.1. Conclusiones

Este proyecto ha abordado el desafío de la detección automática de noticias falsas utilizando técnicas avanzadas de Procesamiento de Lenguaje Natural (PLN). A través de la implementación y comparación de distintos modelos de Machine Learning, Deep Learning y Transformers, se obtuvieron resultados significativamente efectivos en cada uno de los datasets evaluados: noticias completas, tweets aislados y la combinación de ambos.

Entre los hallazgos más relevantes destaca la superioridad clara de los modelos basados en Transformers, específicamente BERT y DeBERTa, que alcanzaron métricas casi perfectas en precisión, recall y F1-score, cercanas o superiores a la perfección. También se confirmó que los modelos de aprendizaje profundo, como las arquitecturas LSTM y BiLSTM, ofrecen resultados robustos, aunque ligeramente inferiores a los Transformers en términos generales.

La incorporación de FastText como técnica de vectorización especialmente adaptada para textos breves, como tweets, demostró una mejora notable en el rendimiento de los modelos tradicionales de Machine Learning, subrayando la importancia de adaptar las técnicas de representación textual al contexto específico de cada dataset.

Además, se realizó un análisis detallado sobre la explicabilidad de los resultados, lo que permitió identificar claramente las palabras y términos clave que los distintos modelos emplean para realizar sus predicciones. Este análisis ofreció una comprensión más profunda del comportamiento interno de cada modelo.

7.2. Trabajo Futuro

Aunque los resultados obtenidos en este trabajo han sido prometedores, quedan múltiples áreas en las que es posible avanzar y profundizar. A continuación se detallan diversas líneas de investigación y desarrollo futuro:

- **Optimización de Modelos y Eficiencia Computacional:** Explorar técnicas avanzadas de compresión y optimización de modelos para reducir los costes computacionales y medioambientales derivados de entrenamientos y despliegues a gran escala.
- **Análisis Multimodal Avanzado:** Incorporar otras modalidades de datos, como imágenes, vídeos o audio, para desarrollar un enfoque multimodal más completo y robusto en la detección de noticias falsas, permitiendo así captar otros patrones.
- **Desarrollo Continuo y Adaptativo:** Implementar estrategias de aprendizaje continuo que permitan actualizar y mejorar los modelos frente a nuevas técnicas emergentes en la creación y propagación de *fake news*.
- **Aplicación en Tiempo Real y Escalabilidad:** Diseñar un sistema escalable y capaz de operar en tiempo real en grandes plataformas como redes sociales, asegurando una detección inmediata que limite la difusión inicial de noticias falsas.
- **Explicabilidad y Transparencia Mejoradas:** Continuar explorando técnicas avanzadas de explicabilidad para hacer que las decisiones de los modelos sean más transparentes y comprensibles para usuarios no técnicos, facilitando así la aceptación social y el uso ético de esta tecnología.

Los trabajos futuros podrán apoyarse en los resultados obtenidos en este trabajo para continuar perfeccionando las capacidades de análisis automático, ampliando su alcance y aplicabilidad en entornos reales. De este modo, será posible no solo incrementar la eficiencia de los sistemas de detección, sino también promover una adopción tecnológica orientada al bienestar del usuario.

Capítulo 8

Análisis de impacto

8.1. Análisis general

A lo largo de este capítulo, se examina el impacto potencial del proyecto en diversos contextos: personal, empresarial, social, económico, medioambiental y cultural. Se destacan además los beneficios esperados en cada ámbito, así como los posibles efectos adversos o riesgos asociados. Más allá de esto, se analiza la contribución del trabajo a los Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030 que resultan relevantes, y se incluyen reflexiones sobre decisiones de desarrollo del proyecto que fueron influenciadas por consideraciones de impacto.

8.1.1. Impacto personal

Tal como se expuso en la Introducción del presente trabajo, la propagación de noticias falsas en redes sociales tiene consecuencias directas en los individuos, afectando su percepción de la realidad y sus decisiones cotidianas. En el ámbito personal, la herramienta desarrollada puede aportar beneficios significativos. Por un lado, brinda a los usuarios indicios de que un contenido puede ser falso, ayudándoles a estar mejor informados lo que contribuye a mejorar la seguridad y confianza con que cada persona navega por las redes sociales, al reducir la probabilidad de que sean engañados por desinformación. Además, al explicar por qué una noticia podría ser falsa, el sistema puede fomentar el pensamiento crítico y la alfabetización mediática del individuo, enseñándole a reconocer patrones de noticias dudosas por sí mismo.

No obstante, también existen posibles efectos adversos a nivel personal. Un riesgo es que el usuario desarrolle excesiva dependencia de la herramienta, confiando ciegamente en el veredicto automático y descuidando su propio juicio crítico. Si el sistema cometiese errores (por ejemplo, marcando erróneamente una noticia verdadera como falsa o viceversa), podría desinformar involuntariamente al usuario o generar confusión y desconfianza. Aun así, este proyecto ha prestado especial atención a minimizar falsos positivos y negativos (véase el capítulo de Resultados, donde se reporta una alta precisión del modelo).

8.1.2. Impacto empresarial

En el ámbito empresarial, los resultados de este proyecto pueden traducirse en beneficios para distintas organizaciones e industrias. Las empresas tecnológicas y plataformas de redes sociales podrían integrar el modelo de detección automática para mejorar la moderación de contenido, identificando rápidamente bulos virales y limitando su difusión. Esto fortalece la confianza de los usuarios en la plataforma y protege la imagen de la empresa, que demuestra responsabilidad al combatir la desinformación. Igualmente, medios de comunicación y agencias de noticias pueden utilizar herramientas de este tipo como apoyo en sus procesos de verificación de datos, agilizando la detección de noticias potencialmente falsas antes de su publicación o difusión. En un contexto más amplio, la tecnología desarrollada abre oportunidades de negocio para empresas especializadas en ciberseguridad, marketing digital o relaciones públicas podrían emplearla para monitorear información falsa relacionada con sus marcas o sectores, protegiendo así su reputación y la de sus productos.

8.1.3. Impacto social

El sistema desarrollado ofrece beneficios claros al reducir la circulación de información falsa, fortaleciendo la confianza pública en las instituciones y promoviendo una ciudadanía más crítica e informada. Sin embargo, existe el riesgo de censura inadvertida de opiniones minoritarias y una posible reducción en la responsabilidad individual hacia la verificación de información. Por ello, se ha procurado asegurar la imparcialidad del algoritmo y se enfatiza en la importancia de complementar la herramienta con programas de educación mediática para evitar una dependencia excesiva en la tecnología.

8.1.4. Impacto económico

En cuanto al ámbito económico, se protege su estabilidad al prevenir pérdidas económicas asociadas a desinformación, generando además oportunidades laborales y comerciales en el sector tecnológico. No obstante, los costos elevados de implementación y mantenimiento podrían limitar la adopción generalizada, especialmente para organismos con menos recursos. Además, errores del sistema (falsos negativos o positivos) podrían generar pérdidas económicas indirectas, destacando la necesidad de gestionar adecuadamente los riesgos asociados.

8.1.5. Impacto medioambiental

Aunque de manera indirecta, el proyecto contribuye a combatir la desinformación climática, favoreciendo discusiones públicas basadas en hechos científicos y mejorando la concienciación ambiental. Aún así, el alto consumo energético asociado a la escalabilidad global del sistema podría generar emisiones significativas, por lo que es vital continuar buscando soluciones eficientes y sostenibles.

8.1.6. Impacto cultural

El sistema ayuda a mantener la integridad cultural e histórica, fomentando un entorno informativo saludable y fortaleciendo valores como el pensamiento crítico y la veracidad. Ahora se centra en contenidos en inglés, pero con la traducción de la noticia, se podría frenar la brecha lingüística. No obstante, persisten riesgos de sesgos culturales que podrían malinterpretar contenido legítimo (satírico o local), afectando negativamente la diversidad y expresión cultural.

8.2. Contribución a los Objetivos de Desarrollo Sostenible (ODS)

La Agenda 2030 establece Objetivos de Desarrollo Sostenible (ODS) que abordan desafíos globales [39]. Este TFG contribuye directamente a los siguientes:

- **ODS 3: Salud y Bienestar.** Reduce la difusión de bulos sanitarios, facilitando decisiones informadas y mejorando respuestas a riesgos globales.
- **ODS 4: Educación de Calidad.** Promueve la alfabetización digital y mediática, fortaleciendo el pensamiento crítico frente a la desinformación en contextos educativos.
- **ODS 5: Igualdad de Género.** Protege contra contenidos discriminatorios hacia mujeres, creando entornos digitales más seguros e igualitarios.
- **ODS 9: Industria, Innovación e Infraestructura.** Impulsa la innovación tecnológica mediante IA aplicada a contenidos digitales, fortaleciendo infraestructuras digitales y cooperación académico-empresarial.
- **ODS 10: Reducción de las Desigualdades.** Protege grupos vulnerables frente a noticias falsas dirigidas, disminuyendo desigualdades digitales y lingüísticas mediante adaptación a contextos hispanohablantes.
- **ODS 13: Acción por el Clima.** Combate la desinformación climática, apoyando debates ambientales basados en evidencia científica esenciales para implementar políticas efectivas.
- **ODS 16: Paz, Justicia e Instituciones Sólidas.** Previene la propagación de *fake news* que podrían incitar a la violencia o debilitar la confianza institucional, favoreciendo sociedades más pacíficas y democráticas.

Bibliografía

[1] S. Sahin. (2024) What is retrieval-augmented generation (rag) in llm and how it works. Accessed May 2025. [Online]. Available: https://medium.com/@sahin.samia/what-is-retrieval-augmented-generation-rag-in-llm-and-how-it-works-a8c79e35a172

[2] D. M. Lazer, M. A. Baum, Y. Benkler, A. J. Berinsky, K. M. Greenhill, F. Menczer, M. J. Metzger, B. Nyhan, G. Pennycook, D. Rothschild, M. Schudson, S. A. Sloman, C. R. Sunstein, E. A. Thorson, D. J. Watts, and J. L. Zittrain, "The science of fake news," *Science*, vol. 359, no. 6380, pp. 1094–1096, 2018.

[3] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 211–236, 2017.

[4] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018.

[5] M. Del Vicario, A. Bessi, F. Zollo, F. Petroni, A. Scala, G. Caldarelli, H. E. Stanley, and W. Quattrociocchi, "The spreading of misinformation online," *PNAS*, vol. 113, no. 3, pp. 554–559, 2016.

[6] G. Pennycook and D. G. Rand, "Lazy, not biased: Susceptibility to partisan fake news is better explained by lack of reasoning than by motivated reasoning," *Cognition*, vol. 188, pp. 39–50, 2019.

[7] E. Pariser, *The Filter Bubble: What the Internet is Hiding from You*. Penguin UK, 2011.

[8] S. Zannettou, T. Caulfield, J. Blackburn, E. D. Cristofaro, M. Sirivianos, G. Stringhini, and G. Suarez-Tangil, "Disinformation warfare: Understanding state-sponsored trolls on twitter and their influence on the web," in *Companion Proceedings of the 2019 World Wide Web Conference*, 2019, pp. 281–289.

[9] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Understanding and detecting fake news on social media: A survey," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 15, no. 4, pp. 1–42, 2019.

[10] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu *et al.*, "Survey of hallucination in natural language generation," *ACM Computing Surveys (CSUR)*, vol. 55, no. 12, pp. 1–38, 2023.

[11] VeraCT Team, "Veract scan: Retrieval-augmented fake news detection with justifiable reasoning," https://arxiv.org/abs/2406.10289, 2024, arXiv:2406.10289.

[12] A. Unknown, "Fake news detection with retrieval augmented generative artificial intelligence," KDD 2024 Workshop KiL Submission, 2024.

[13] ——, "Crave: Leveraging retrieval-augmented llm for social media analysis," https://arxiv.org/abs/2504.10166, 2025, arXiv:2504.10166.

[14] Q. Wang, Y. Yang, J. Ma, Z. Wang, and X. Wang, "A knowledge-guided framework for few-shot fake news detection," https://arxiv.org/abs/2407.08952, 2024, arXiv:2407.08952.

[15] Y. Liu, J. Zhu, X. Liu, H. Tang, Y. Zhang, K. Zhang, X. Zhou, and E. Chen, "Detect, investigate, judge and determine: A knowledge-guided framework for few-shot fake news detection," https://arxiv.org/abs/2407.08952, 2024.

[16] B. Deng, W. Wang, F. Zhu, Q. Wang, and F. Feng, "Cram: Credibility-aware attention modification in llm for combating misinformation in rag," https://arxiv.org/abs/2406.11497, 2024.

[17] E. Yilmaz, R. M. Yilmaz, and R. Yilmaz, "Real-time fake news detection in online social networks," *Scientific Reports*, vol. 14, p. 76102, 2024.

[18] Y. Chai, K. Shi, J. Xie, C. Liu, Y. Jiang, and Y. Liu, "Detecting fake news on social media: A novel reliability aware machine-crowd hybrid intelligence-based method," https://arxiv.org/abs/2412.06833, 2024.

[19] C. Chen and K. Shu, "Real-time fake news from adversarial feedback," https://arxiv.org/abs/2410.14651, 2024.

[20] N. Ahmed, I. Traore, and S. Saeed, "Detecting fake news using machine learning: A literature review," *SmartCR*, vol. 9, no. 4, pp. 239–255, 2019.

[21] A. Pathak and M. Sharma, "Combining lstm and user metadata for fake news detection in twitter," in *Proc. of the 13th Intl. Conf. on Social Computing*, 2020, pp. 157–162.

[22] D. Qian, Z. Huang, and X. Jin, "Exploring bert for fake news detection: The importance of attention and context," in *2021 IEEE Intl. Conf. on Data Mining (ICDM)*, 2021, pp. 1225–1230.

[23] K. Praseed, A. Ranjan, and A. Mehta, "Multilingual transformer models for fake news detection in hindi," *Procedia Computer Science*, vol. 199, pp. 563–570, 2022.

[24] "Python programming language," https://www.python.org, 2024.

[25] "Project jupyter," https://jupyter.org, 2024.

[26] "Google colaboratory," https://colab.research.google.com, 2024.

[27] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly Media, 2009.

[28] C. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," https://github.com/cjhutto/vaderSentiment, 2014.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort *et al.*, "Scikit-learn: Machine learning in python," https://scikit-learn.org, 2011.

[30] T. Mikolov, E. Grave, P. Bojanowski, and A. Joulin, "Fasttext: Efficient text classification and representation," https://fasttext.cc, 2024.

[31] T. Wolf, L. Debut, V. Sanh *et al.*, "Transformers: State-of-the-art natural language processing," https://huggingface.co/transformers, 2020.

[32] "Gemini 2.0 flash - google deepmind," https://cloud.google.com/vertex-ai/docs/generative-ai/model-reference/gemini, 2024, accedido en mayo de 2025.

[33] "Matplotlib: Visualization with python," https://matplotlib.org, 2024.

[34] "Seaborn: Statistical data visualization," https://seaborn.pydata.org, 2024.

[35] "Pandas: Python data analysis library," https://pandas.pydata.org, 2024.

[36] "Joblib: Lightweight pipelining in python," https://joblib.readthedocs.io, 2024.

[37] E. Yetim, "Fake news detection datasets," 2023, accessed May 2025. [Online]. Available: https://www.kaggle.com/datasets/emineyetm/fake-news-detection-datasets

[38] M. Marchetti, "Tweets dataset," 2021, accessed May 2025. [Online]. Available: https://www.kaggle.com/datasets/mmmarchetti/tweets-dataset

[39] Naciones Unidas, "Objetivos de desarrollo sostenible (ods)," 2015, accedido: 29-May-2025. [Online]. Available: https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/

Anexos

Apéndice A

Primer anexo

En este apéndice se incluyen las visualizaciones complementarias de los modelos de aprendizaje automático evaluados. Para cada modelo se presenta su correspondiente matriz de confusión y curva de aprendizaje sobre el conjunto de test. Estas imágenes permiten observar gráficamente el comportamiento del modelo, detectar posibles sesgos y evaluar su capacidad de generalización.

A.1. Dataset: Solo Texto

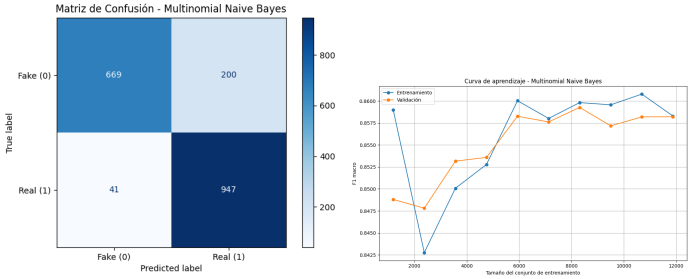

Figura A.1: Matriz de Confusión y Curva de Aprendizaje - Multinomial Naïve-Bayes

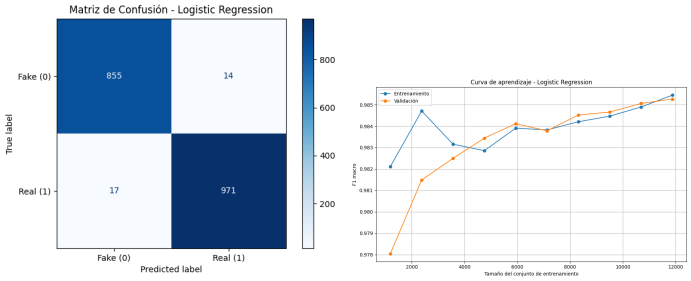

Figura A.2: Matriz de Confusión y Curva de Aprendizaje - Logistic Regression

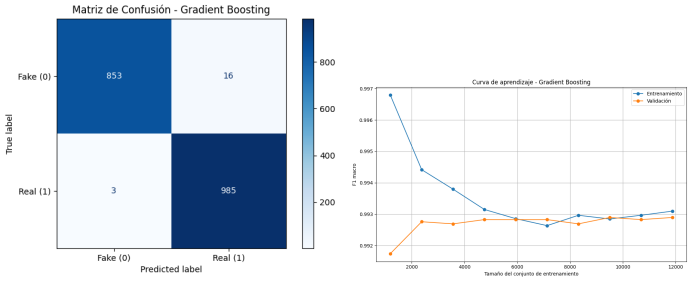

Figura A.3: Matriz de Confusión y Curva de Aprendizaje - Gradient Boosting

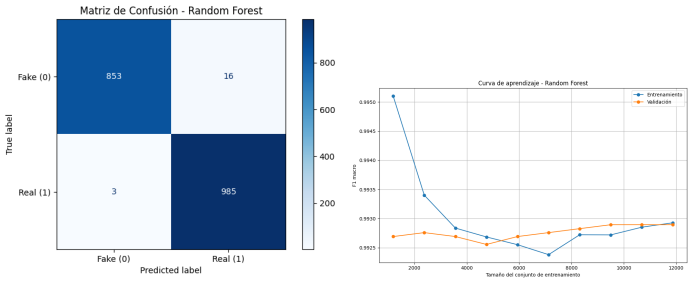

Figura A.4: Matriz de Confusión y Curva de Aprendizaje - Random Forest

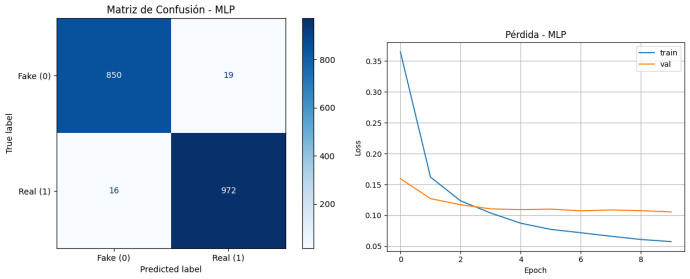

Figura A.5: Matriz de Confusión y Función de pérdida - MLP

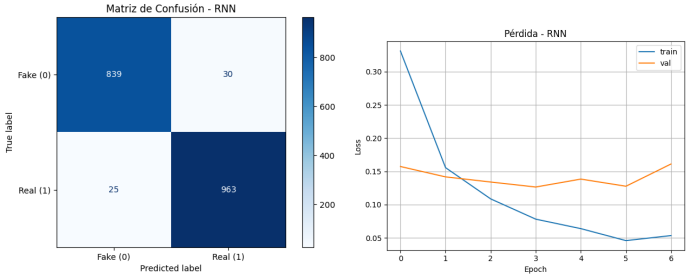

Figura A.6: Matriz de Confusión y Función de pérdida - RNN

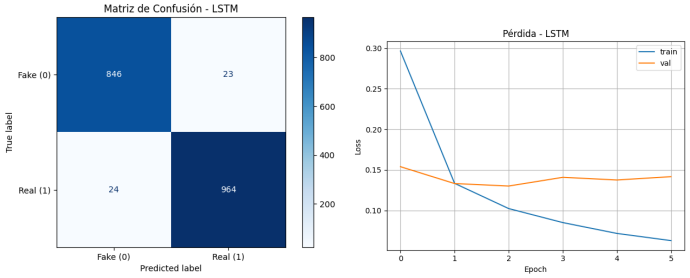

Figura A.7: Matriz de Confusión y Función de pérdida - LSTM

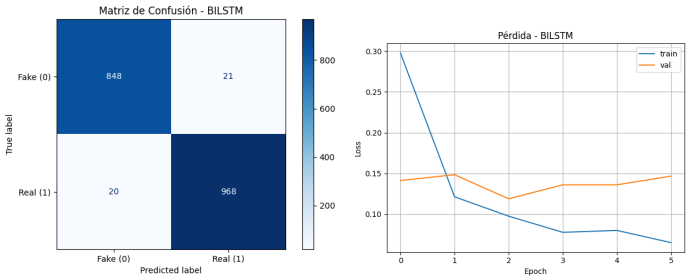

Figura A.8: Matriz de Confusión y Función de pérdida - BILSTM

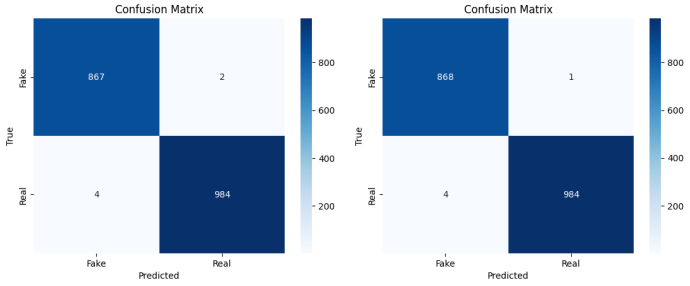

Figura A.9: Matriz de Confusión - Bert y DeBertA

A.2. Dataset: Solo Tweets

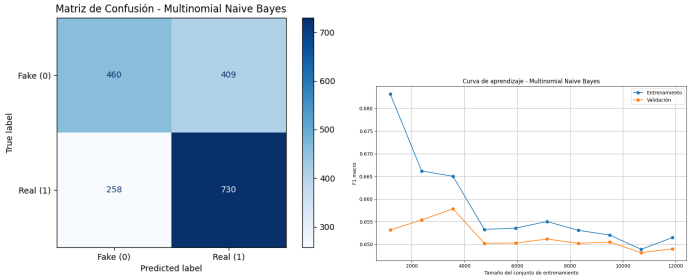

Figura A.10: Matriz de Confusión y Curva de Aprendizaje - Multinomial Naïve-Bayes

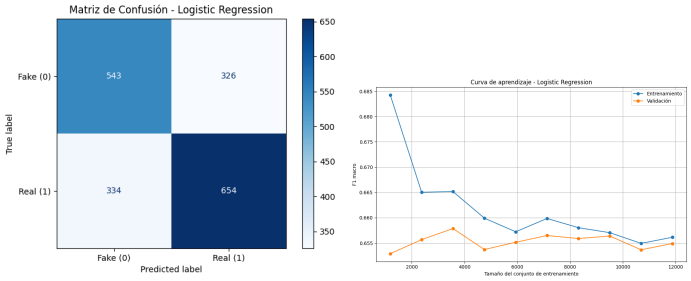

Figura A.11: Matriz de Confusión y Curva de Aprendizaje - Logistic Regression

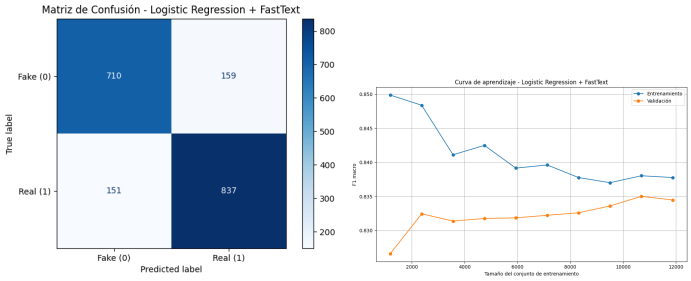

Figura A.12: Matriz de Confusión y Curva de Aprendizaje - Logistic Regression

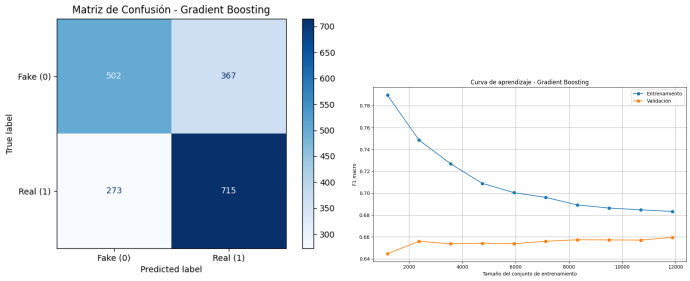

Figura A.13: Matriz de Confusión y Curva de Aprendizaje - Gradient Boosting

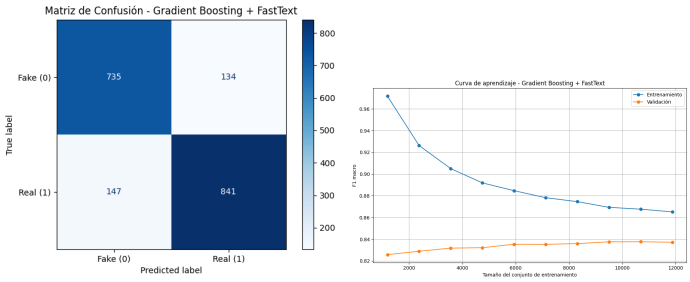

Figura A.14: Matriz de Confusión y Curva de Aprendizaje - Gradient Boosting

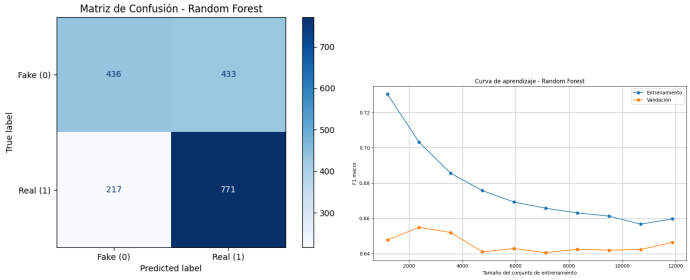

Figura A.15: Matriz de Confusión y Curva de Aprendizaje - Random Forest

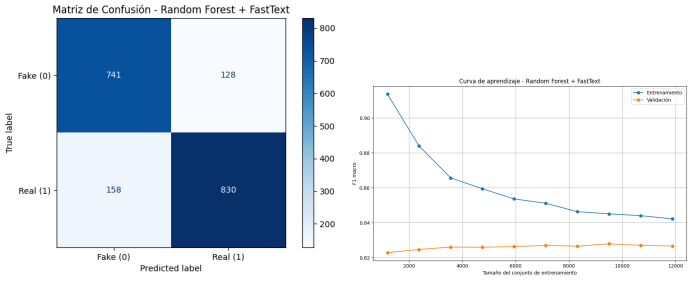

Figura A.16: Matriz de Confusión y Curva de Aprendizaje - Random Forest

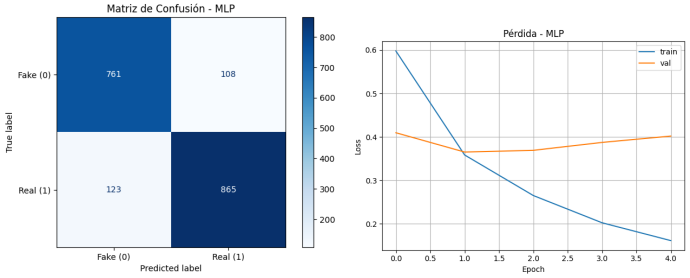

Figura A.17: Matriz de Confusión y Función de pérdida - MLP

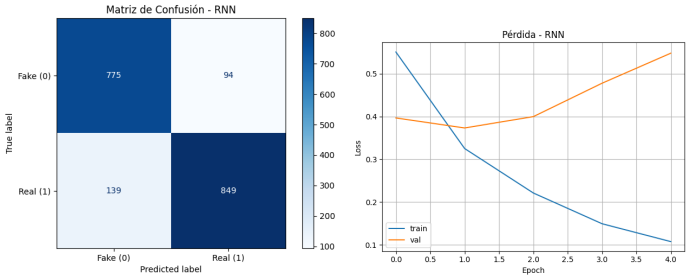

Figura A.18: Matriz de Confusión y Función de pérdida - RNN

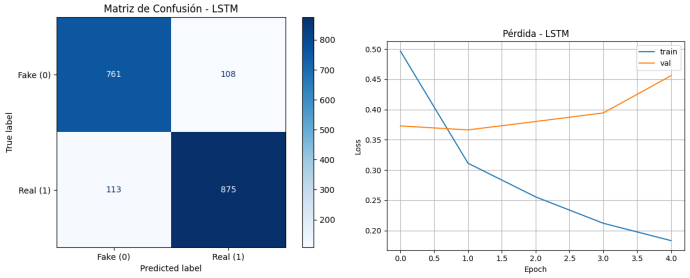

Figura A.19: Matriz de Confusión y Función de pérdida - LSTM

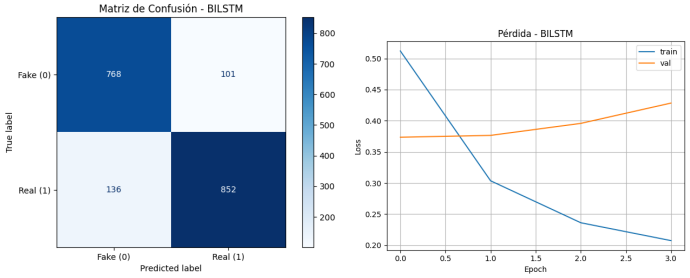

Figura A.20: Matriz de Confusión y Función de pérdida - BILSTM

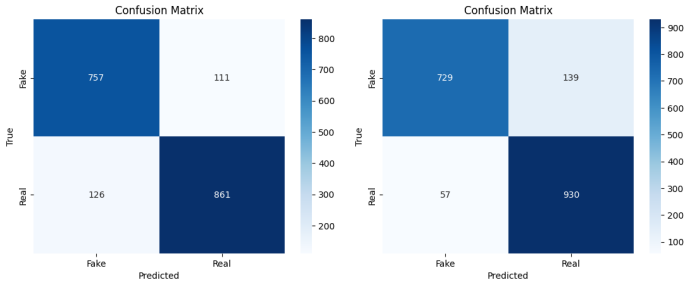

Figura A.21: Matriz de Confusión - Bert y DeBertA

A.3. Dataset: Texto + Tweets

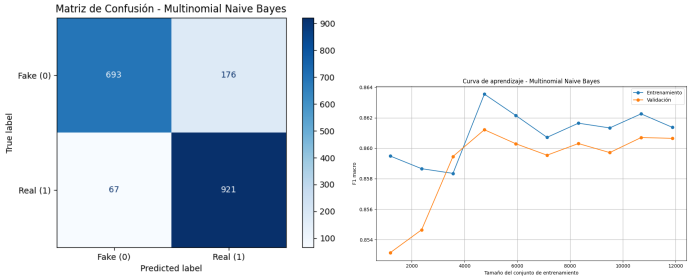

Figura A.22: Matriz de Confusión y Curva de Aprendizaje - Multinomial Naïve-Bayes

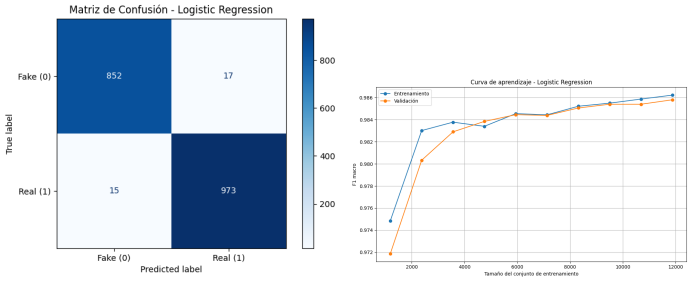

Figura A.23: Matriz de Confusión y Curva de Aprendizaje - Logistic Regression

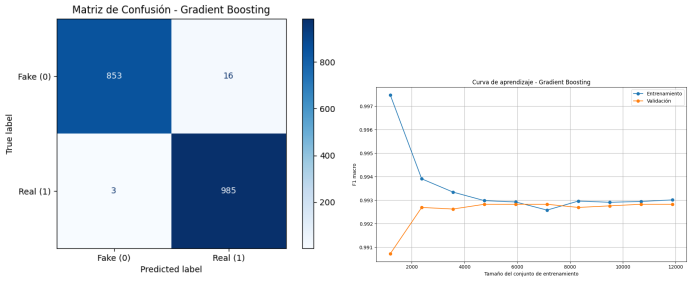

Figura A.24: Matriz de Confusión y Curva de Aprendizaje - Gradient Boosting

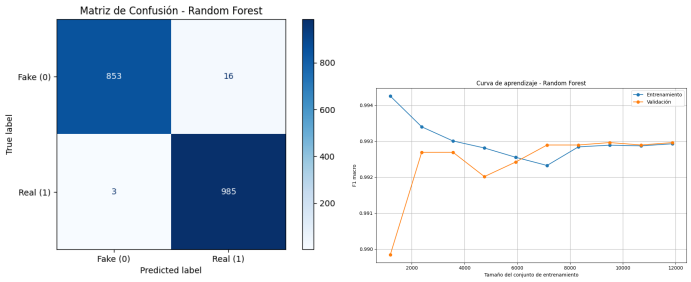

Figura A.25: Matriz de Confusión y Curva de Aprendizaje - Random Forest

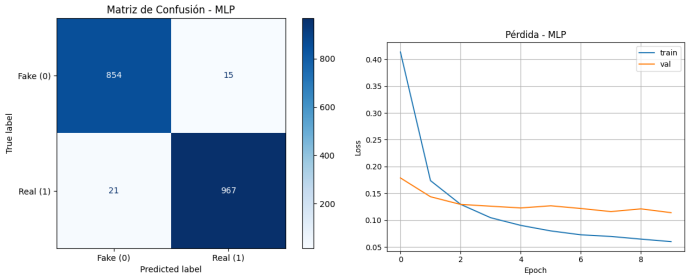

Figura A.26: Matriz de Confusión y Función de pérdida - MLP

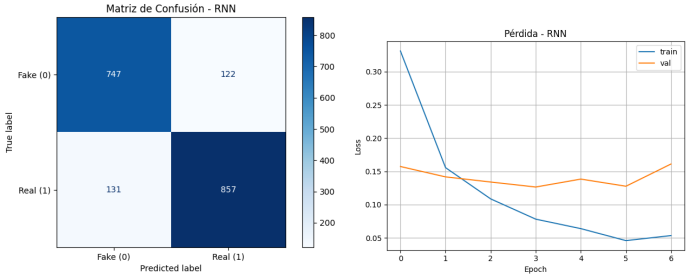

Figura A.27: Matriz de Confusión y Función de pérdida - RNN

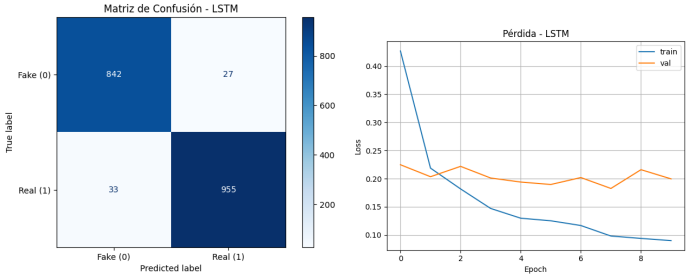

Figura A.28: Matriz de Confusión y Función de pérdida - LSTM

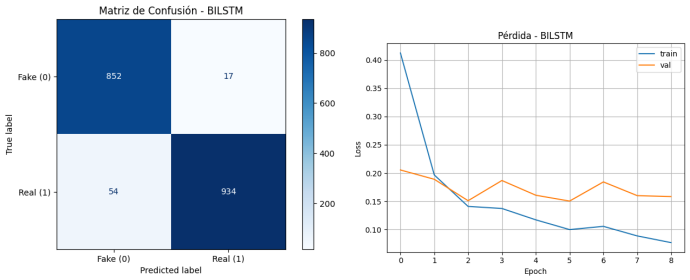

Figura A.29: Matriz de Confusión y Función de pérdida - BILSTM

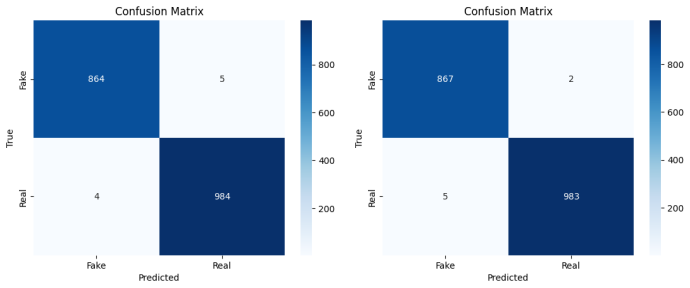

Figura A.30: Matriz de Confusión - Bert y DeBertA

Apéndice B

Documento de Turnitin

ALVARO HERNANDEZ RODRIGUEZ

TFG_ALVARO_HERNANDEZ_RODRIGUEZ.pdf

Turnitin Memoria Final

TFG ETSIINF (Moodle PP)

Universidad Politecnica de Madrid

Detalles del documento

Identificador de la entrega

trn:oid:::1:3266936450

Fecha de entrega

2 jun 2025, 5:05 p.m. GMT+2

Fecha de descarga

3 jun 2025, 10:24 a.m. GMT+2

Nombre de archivo

27789_ALVARO_HERNANDEZ_RODRIGUEZ_TFG_ALVARO_HERNANDEZ_RODRIGUEZ_83714_7797115....pdf

Tamaño de archivo

2.9 MB

91 Páginas

22.308 Palabras

121.804 Caracteres

4% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Exclusions

- 2 Excluded Sources

Top Sources

0% Internet sources
0% Publications
4% Submitted works (Student Papers)

Top Sources

0% Internet sources
0% Publications
4% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1 Student papers
Universidad Politécnica de Madrid 1%

2 Student papers
Universidad San Francisco de Quito <1%

3 Student papers
Universitat Politècnica de València <1%

4 Student papers
Universidad Europea de Madrid <1%

5 Student papers
Universidad de Chile <1%

6 Student papers
ucol <1%

7 Student papers
Chester College of Higher Education <1%

8 Student papers
Consorcio CIXUG <1%

9 Student papers
Universidad Internacional de la Rioja <1%

10 Student papers
Universidad Carlos III de Madrid <1%

11 Student papers
Escuela Politecnica Nacional <1%

12 Student papers

ipn <1%

13 Student papers

National Economics University <1%

14 Student papers

Universidad de Deusto <1%

15 Student papers

University of Birmingham <1%

16 Student papers

Corporación Universitaria Minuto de Dios,UNIMINUTO <1%

17 Student papers

Universidad Rey Juan Carlos <1%

18 Student papers

Universidad TecMilenio <1%

19 Student papers

Universidad a Distancia de Madrid <1%

20 Student papers

uaq <1%

21 Student papers

CORPORACIÓN UNIVERSITARIA IBEROAMERICANA <1%

22 Student papers

UNIBA <1%

23 Student papers

Universidad de Cantabria <1%

24 Student papers

Universidad de León <1%

25 Student papers

Universidad de Málaga <1%

26 Student papers

ebsu <1%

Este documento esta firmado por

	**Firmante**	CN=tfgm.fi.upm.es, OU=CCFI, O=ETS Ingenieros Informaticos - UPM, C=ES
	Fecha/Hora	Tue Jun 03 10:37:55 CEST 2025
	Emisor del Certificado	EMAILADDRESS=camanager@etsiinf.upm.es, CN=CA ETS Ingenieros Informaticos, O=ETS Ingenieros Informaticos - UPM, C=ES
	Numero de Serie	561
	Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)