PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE HÁBITAT, INFRAESTRUCTURA Y CREATIVIDAD

CARRERA DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN

TRABAJO DE TITULACIÓN

TEMA:

# DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO BASADO EN APRENDIZAJE PROFUNDO PARA LA DETECCIÓN DE NOTICIAS FALSAS EN ESPAÑOL.

AUTOR:

ANTHONY ANDRÉS CORREA CASTRO

DIRECTOR:

HENRY NELSON ROA MARIN, PHD.

QUITO DM, OCTUBRE DE 2025

# Dedicatoria

Este trabajo de titulación está dedicado, con profundo agradecimiento, a mi familia, especialmente a mis padres, quienes han sido mi mayor apoyo y fortaleza durante todo mi proceso académico. Su sacrificio, consejos y confianza permanente me impulsaron a seguir adelante incluso en los momentos más difíciles.

Dedico también este logro a mi hermano, por su compañía, apoyo y motivación constante, y a mis tías, quienes con sus palabras de aliento y respaldo incondicional contribuyeron de manera significativa a mi crecimiento personal y profesional.

De igual manera, expreso mi sincero reconocimiento a mis profesores, por compartir sus conocimientos, orientar mi aprendizaje y aportar con su experiencia a mi formación académica.

# Agradecimiento

Expreso mi sincero agradecimiento a Dios por la vida, la fortaleza y las oportunidades que me ha concedido para alcanzar esta meta académica. A mis padres, por su amor incondicional, sacrificio y apoyo constante, quienes hicieron posible mi formación personal y profesional, brindándome siempre motivación y confianza para seguir adelante.

Asimismo, agradezco a todas las personas que, de una u otra manera, contribuyeron a lo largo de este proceso académico, acompañándome con su apoyo, consejos y enseñanzas.

# Resumen

El presente trabajo de titulación describe el desarrollo de un sistema para la detección de noticias falsas en español, basado en el uso de modelos de lenguaje preentrenados y técnicas de aprendizaje profundo. El proyecto se centra en la evaluación y comparación de distintos modelos con el fin de seleccionar aquel que presente el mejor desempeño.

Para el desarrollo del sistema, se utilizarán conjuntos de datos públicos que serán sometidos a un proceso de preprocesamiento, permitiendo el entrenamiento y evaluación de los modelos. Finalmente, el modelo seleccionado será implementado en un sistema web sencillo, el cual permitirá a los usuarios ingresar textos y obtener una predicción sobre la veracidad de la información de manera rápida y accesible.

## Tabla de contenidos

# ÍNDICE TABLAS



# ÍNDICE DE ILUSTRACIONES

# 1. Introducción

## 1.1 Justificación

En la era digital de hoy en día, la propagación de noticias falsas se ha convertido en un fenómeno mundial que afecta la credibilidad de muchos medios de comunicación, pues distorsiona la percepción pública y genera consecuencias sociales y políticas significativas, positivas o negativas. Este problema se agrava en los entornos digitales, donde la información se comparte de forma masiva y sin filtros, especialmente a través de redes sociales y plataformas de noticias.

Frente a esta situación, surge la necesidad de desarrollar herramientas tecnológicas capaces de identificar de manera automática y eficiente la veracidad de los contenidos informativos. En este contexto, los modelos de aprendizaje profundos han demostrado ser aliados notables en tareas de procesamiento del lenguaje natural y constituyen una alternativa prometedora para abordar problemas de desinformación.

El presente trabajo tiene como objetivo evaluar y comparar el desempeño de distintos modelos de lenguaje preentrenados, como DeBERTa, RoBERTa y DistilBERT , en la detección de noticias falsas en español. El estudio se desarrolla siguiendo la metodología CRISP-DM, lo que permite un enfoque estructurado que abarca desde la comprensión del problema y los datos hasta la evaluación de los modelos y su posterior implementación en un prototipo web desplegado en Hugging Face Spaces, que permite a los usuarios ingresar textos y obtener una predicción sobre la veracidad de la noticia.

## 1.2 Planteamiento del problema

La difusión de noticias falsas en medios digitales se ha incrementado considerablemente en los últimos años, afectando la confianza de los usuarios en las fuentes informativas que se encuentran a su alcance, producto de la facilidad con la que los contenidos falsos se crean y se viralizan; es por esto la importancia de este proyecto, que obedece a la necesidad urgente de contar con herramientas automatizadas, rápidas y confiables que puedan detectar información maliciosa o fraudulenta.

El problema central que aborda esta investigación es: *¿Qué tan efectivos son los modelos de lenguaje preentrenados basados en aprendizaje profundo para la detección de noticias falsas*

*en español, considerando criterios de precisión y eficiencia, y cuál de ellos presenta el mejor desempeño para su implementación en un prototipo web?*

Al abordar esta incógnita, surgen las siguientes inquietudes:

- ¿Qué modelos de lenguaje preentrenados ofrecen el mejor desempeño en la clasificación de noticias falsas en español?
- ¿Qué tipo de preprocesamiento y ajuste de datos se requieren para optimizar el rendimiento del modelo?
- ¿Qué métricas y técnicas de evaluación permiten medir de forma confiable la efectividad del sistema?
- ¿Cómo puede integrarse el modelo en una interfaz web que facilite su uso por usuarios no técnicos?

El estudio busca responder estas preguntas mediante el entrenamiento, la comparación y la evaluación de distintos modelos de aprendizaje profundo en un entorno controlado, para posteriormente implementar un prototipo funcional accesible desde la web.

## 1.3 Objetivos

### 1.3.1 General

- Comparar y evaluar tres modelos de lenguaje preentrenados para clasificar noticias en español como falsas o verdaderas, seleccionando el de mejor desempeño y desplegándolo en un espacio web de Hugging Face para que los usuarios puedan probarlo.

### 1.3.2 Específicos:

- Analizar el problema de la desinformación en los medios digitales y su impacto en el contexto hispanohablante.
- Seleccionar y preparar conjuntos de datos en español que incluyan noticias reales y falsas.
- Comparar los resultados obtenidos mediante métricas de desempeño como accuracy, precision, recall, F1-score y AUC-ROC.

- Implementar un prototipo web interactivo que permita a los usuarios ingresar textos y recibir una predicción sobre la veracidad de la noticia.

## 1.4 Alcance

El presente proyecto se enfoca en la evaluación y comparación de distintos modelos de lenguaje preentrenados basados en el aprendizaje profundo para la detección de noticias falsas en español, con el objetivo de identificar el modelo que obtenga el mejor desempeño y utilizarlo posteriormente en una aplicación web sencilla como apoyo para la verificación de información.

La información utilizada para el entrenamiento y validación del sistema provendrá de conjuntos de datos públicos disponibles en la web, los cuales serán sometidos a su respectivo proceso de preprocesamiento, que incluye la limpieza, normalización y preparación de los textos antes de su uso en los modelos.

Finalmente, se implementará un prototipo funcional en un entorno web que permitirá ingresar textos y obtener una predicción sobre su veracidad. El proyecto no contempla la creación de modelos desde cero ni el análisis de imágenes o videos, limitándose al procesamiento de texto en español.

# 2. Marco Teórico

## 2.1 Fundamentos conceptuales

### 2.1.1 Definición de desinformación y noticias falsas

En los últimos años, las llamadas fake news o noticias falsas se han vuelto mucho más comunes en internet, especialmente en las redes sociales. Estas noticias tienen como objetivo engañar a las personas, ya sea con fines políticos o económicos, o simplemente para obtener atención. Sin embargo, el término fake news no explica por completo todo lo que ocurre con la información engañosa en la red.

Según Rodríguez Pérez (2019), es más correcto hablar de desinformación, ya que este concepto incluye no solo las noticias falsas, sino también los rumores, la propaganda, la información manipulada o los contenidos creados para confundir al público. La desinformación consiste en difundir información falsa o distorsionada de manera intencional para engañar al lector. Frente a este problema, la inteligencia artificial se ha convertido en una herramienta ampliamente utilizada para identificar contenidos falsos y ayudar a las personas a acceder a información más confiable.

### 2.1.2 Procesamiento del lenguaje natural (PNL)

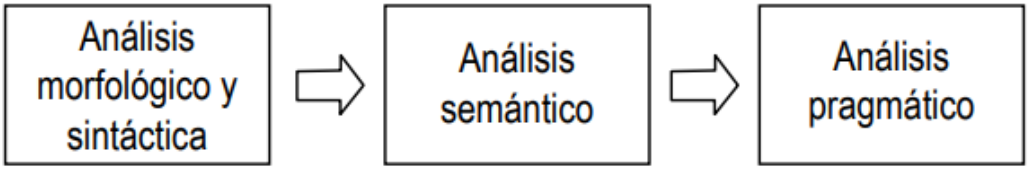
El Procesamiento del Lenguaje Natural (PLN) es una rama de la inteligencia artificial que busca que las computadoras puedan entender y comunicarse mediante el lenguaje humano. Es decir, que podamos hablarles o escribirles como lo hacemos con otras personas y que el sistema sea capaz de interpretar lo que decimos.

De acuerdo con Cortez Vásquez, Vega Huerta, Jaime y Quispe (2009), el PLN permite desarrollar programas que trabajan con el texto o el lenguaje hablado, e incluso crear modelos que imitan la forma en que los seres humanos comprenden y procesan las palabras. Para lograrlo, el PLN se analiza en distintos niveles:

- Nivel sintáctico: se trata de cómo se juntan las palabras para formar oraciones y qué papel cumple cada palabra en la oración.
- Nivel semántico: se centra en el significado de las palabras y cómo juntas crean sentido en una oración, sin fijarse en el contexto.

- Nivel pragmático: estudia cómo cambia el significado de una oración según la situación en la que se dice y lo que se dijo antes.

***Ilustración 1*** *Arquitectura de un Sistema de Procesamiento de Lenguaje Natural*




(Augusto Cortez Vásquez, 2009)

### 2.1.3 Aprendizaje profundo

El aprendizaje profundo (deep learning), según Pereyras (2015), es una forma de aprender que integra lo nuevo con lo que ya se sabe, lo que permite comprender mejor y recordar a largo plazo, en lugar de solo memorizar información.

Por ejemplo, usamos el aprendizaje profundo para que la computadora pueda identificar noticias falsas en español. A diferencia de métodos simples que solo buscan palabras clave, los modelos de deep learning entienden el contenido completo de la noticia y detectan patrones que indican si es verdadera o falsa, incluso en casos que nunca han visto antes.

Así, el aprendizaje profundo permite que el sistema "entienda" los textos y tome decisiones más precisas, lo que ayuda a detectar desinformación de manera eficiente.

### 2.1.4 Modelos Preentrenados

Un modelo preentrenado es un modelo de machine learning que ha sido entrenado previamente con grandes conjuntos de datos, con el objetivo de aprender patrones generales. Una vez finalizada esta etapa inicial, el modelo puede ser reutilizado y adaptado a nuevas tareas relacionadas, lo que permite disminuir significativamente el tiempo y los recursos necesarios frente a un entrenamiento desde cero (Stryker, 2025).

El desarrollo de este tipo de modelos desde cero suele implicar un alto costo computacional y un alto nivel de conocimiento. Por esta razón, normalmente son creados por grandes compañías tecnológicas, centros de investigación académica, organizaciones sin fines de lucro o comunidades de software libre. En áreas como el deep learning, donde los modelos manejan millones de parámetros, los modelos preentrenados representan una base sólida sobre la cual

se pueden construir nuevas soluciones sin necesidad de entrenar un modelo completamente desde 0.

Una de las principales ventajas de los modelos preentrenados es que ya cuentan con conocimientos previos, como la forma en que se estructura el lenguaje. Gracias a esto, pueden adaptarse fácilmente a conjuntos de datos más pequeños y a tareas específicas, lo que acelera y facilita el desarrollo de aplicaciones de machine learning.

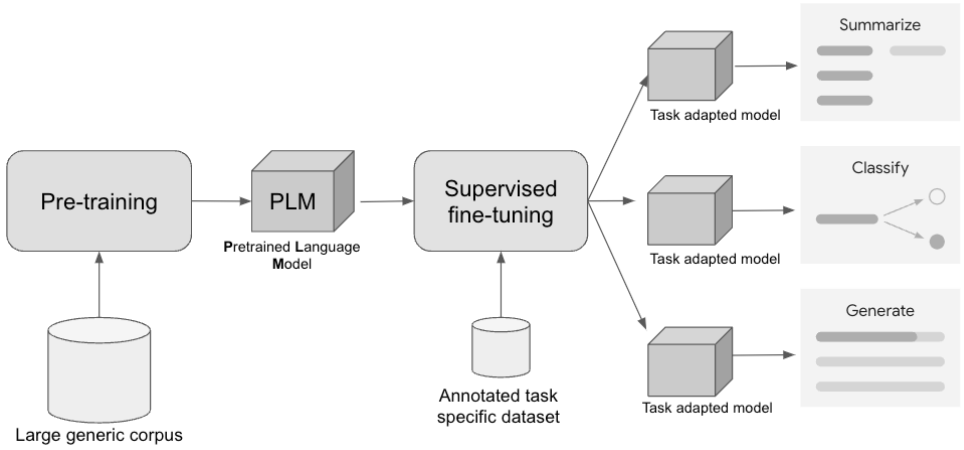
### 2.1.5 Fine-tuning

Según Bergmann (2024), el fine-tuning es una técnica de machine learning que consiste en adaptar un modelo previamente entrenado para resolver una tarea específica. Tal como se muestra en la ilustración 2, el proceso inicia con una etapa de preentrenamiento, en la que el modelo aprende información general a partir de grandes volúmenes de datos. Como resultado de esta etapa, se obtiene un modelo de lenguaje preentrenado, capaz de comprender patrones generales del lenguaje.

Posteriormente, este modelo preentrenado pasa por una fase de fine-tuning supervisado, en la que se utiliza un conjunto de datos más pequeño y específico, previamente etiquetado según la tarea que se desea resolver. En esta etapa, el modelo no aprende desde cero, sino que ajusta sus parámetros internos para especializarse en un objetivo concreto, como la clasificación de textos, la generación de contenido o el resumen de información.

El fine-tuning permite que un mismo modelo base se adapte a diferentes tareas sin necesidad de desarrollar un modelo nuevo para cada caso. Por ejemplo, a partir del mismo modelo preentrenado, es posible obtener modelos ajustados para clasificar información, generar texto o resumir documentos, según los datos utilizados durante el proceso de ajuste.

***Ilustración 2*** *Flujo de fine-tuning*




(Huizenga, 2024)

## 2.2 Modelos utilizados

### 2.2.1 Modelos basados en Transformers

#### 2.2.1.1 Arquitectura Transformer

La arquitectura **Transformer** es una forma moderna en la que los modelos de inteligencia artificial trabajan con texto para entender mejor lo que se dice. A diferencia de otros modelos más antiguos, el Transformer no analiza las palabras una por una en orden, sino que **procesa toda la oración al mismo tiempo**, lo que le permite captar mejor el contexto general.

La idea clave del Transformer es la **atención**. Esto significa que el modelo aprende a fijarse más en las palabras que son importantes y a identificar cómo se relacionan entre sí. Por ejemplo, puede entender que dos palabras tienen una relación fuerte aunque estén lejos dentro de la oración, y que esa relación es necesaria para comprender bien el mensaje completo.

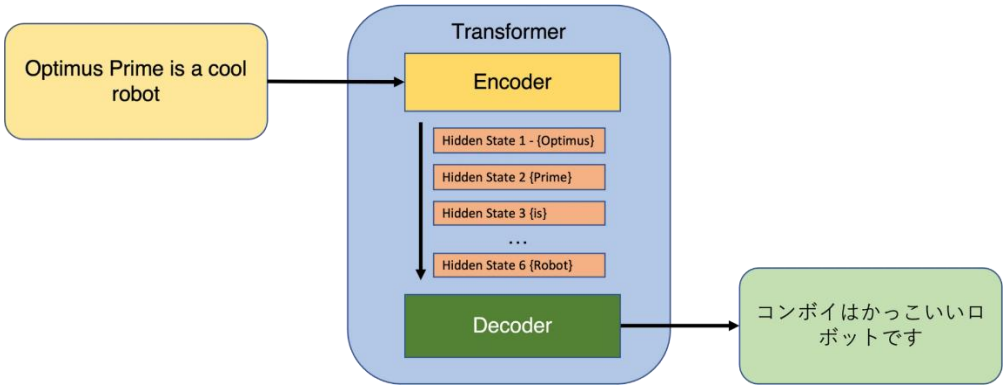
Esta arquitectura se divide en dos partes principales: el Encoder y el Decoder. El Encoder recibe la oración de entrada y se encarga de analizar cada palabra, generando representaciones internas que resumen su significado según el contexto. En otras palabras, no solo entiende la palabra como tal, sino también cómo se usa dentro de la frase.

Luego, esa información pasa al Decoder, que utiliza lo aprendido por el Encoder para generar una salida. Esta salida puede ser una traducción, un resumen o cualquier otro tipo de texto. El

Decoder se apoya tanto en la información original como en las palabras que ya ha generado, lo que ayuda a que el resultado final sea coherente.

Como se muestra en la figura, el proceso comienza cuando una frase como *"Optimus Prime is a cool robot"* entra al Encoder. Ahí se analizan las relaciones entre las palabras y se construye una representación del significado de la oración. Después, el Decoder usa esa información para producir una salida en otro formato. Gracias a esta forma de trabajo, el Transformer es muy eficiente y se ha convertido en la base de muchos modelos actuales de inteligencia artificial.

***Ilustración 3*** *Representación visual del modelo Transformer en NLP*




(Rogel-Salazar, 2023)

### 2.2.1.2 Modelo BERT (modelo base)

Según Luna (2024), BERT es un modelo de lenguaje que se utiliza en inteligencia artificial para entender textos de forma más natural. Su nombre proviene de Bidirectional Encoder for Transformers, lo que significa que está construido sobre la arquitectura Transformer y analiza el texto en ambas direcciones, es decir, de izquierda a derecha y de derecha a izquierda simultáneamente.

Al estar basado en Transformers, BERT aprovecha su capacidad para analizar simultáneamente todas las palabras de una oración y entender cómo se relacionan entre sí en el contexto. Esto le permite captar mejor el significado real de una frase, ya que una palabra puede cambiar de sentido según las que la rodean.

BERT fue propuesto por Devlin y su equipo Google AI en 2018 como una mejora frente a modelos anteriores que solo analizaban el texto en una sola dirección.

### 2.2.1.3 Modelo DeBERTa

DeBERTa es un modelo de lenguaje desarrollado por Microsoft como una mejora directa sobre BERT. Su nombre proviene de Decoding-enhanced BERT with Disentangled Attention, y su principal diferencia con BERT es que separa la información del contenido y la posición de las palabras, lo que le permite entender mejor el contexto de una oración (Hugging Face, 2021).

Al igual que BERT, DeBERTa está basado en la arquitectura Transformer y utiliza un enfoque bidireccional para analizar el texto, es decir, considera tanto las palabras anteriores como las posteriores dentro de una frase. Esto hace que el modelo tenga un mejor desempeño en tareas de comprensión del lenguaje.

La versión utilizada en este trabajo es DeBERTa, una variante multilingüe entrenada con grandes volúmenes de texto en distintos idiomas, lo que la hace adecuada para tareas en español.

### 2.2.1.4 Modelo RoBERTa

RoBERTa es un modelo desarrollado por Facebook AI (Meta) que se basa directamente en BERT, pero con un proceso de entrenamiento optimizado. Los investigadores demostraron que BERT no estaba aprovechado al máximo, por lo que RoBERTa mejora su rendimiento al ajustar la forma en que el modelo se entrena. (Hugging Face, 2019).

Este modelo mantiene la arquitectura Transformer y el enfoque bidireccional de BERT, pero introduce mejoras como el uso de más datos, un mayor tiempo de entrenamiento y un método de enmascaramiento dinámico de palabras. Gracias a estos cambios, RoBERTa logra una mejor comprensión del texto sin modificar la estructura base de BERT.

RoBERTa está disponible en Hugging Face, lo que permite su uso en tareas de clasificación.

### 2.2.1.5 Modelo DistilBERT

DistilBERT es un modelo desarrollado por Hugging Face como una versión más ligera de BERT. Su objetivo principal es reducir el tamaño del modelo y acelerar el tiempo de respuesta, manteniendo un rendimiento similar al modelo original.

Este modelo se basa en BERT y en la arquitectura Transformer, pero utiliza una técnica llamada distillation, donde un modelo pequeño aprende a imitar el comportamiento de uno más grande. Gracias a esto, DistilBERT conserva gran parte del conocimiento de BERT, pero con menos parámetros y menor costo computacional (Hugging Face, 2019).

Por estas características, DistilBERT es especialmente útil cuando se requiere eficiencia, como en aplicaciones web o sistemas con recursos limitados.

## 2.3 Herramientas

### 2.3.1 Hugging Face

Hugging Face es una plataforma y biblioteca de machine learning especializada en procesamiento de lenguaje natural (NLP). Permite descargar, entrenar y probar modelos de lenguaje preentrenados así como crear notebooks para tareas como clasificación de texto, traducción, resumen y generación de contenido.
Se decidió usar Hugging Face porque ofrece acceso rápido a gran variedad de modelos preentrenados, reduce el tiempo de desarrollo y permite experimentar con distintas arquitecturas sin complicaciones. Además, cuenta con una comunidad muy activa que comparte ejemplos y tutoriales, lo que facilita aprender y resolver problemas durante el proyecto (Hugging Face, 2023).

### 2.3.3 Google Colab

Google Colab es un entorno de programación en la nube que permite ejecutar código en Python directamente desde el navegador. Ofrece acceso gratuito a GPUs y TPUs, lo que permite entrenar grandes modelos de IA sin depender del hardware local. También permite guardar los notebooks en Google Drive y colaborar en tiempo real con otras personas.

Se eligió Google Colab porque es práctico, gratuito y facilita entrenar grandes modelos de NLP sin problemas de hardware, además de permitir trabajar de manera colaborativa y organizada (Google, 2023).

### 2.3.4 GitHub

GitHub es una plataforma para almacenar, gestionar y compartir proyectos de programación, muy útil para el control de versiones. Permite crear repositorios, trabajar en ramas diferentes y mantener un historial completo de los cambios realizados.

Se decidió usar GitHub para acceder a datasets creados y compartidos por la comunidad, lo que permitió acceder a recursos ya disponibles para el desarrollo del proyecto (GitHub, 2023).

### 2.3.4 Kaggle

Kaggle es una plataforma en línea utilizada en el ámbito del data science y machine learning que ofrece una gran cantidad de datasets públicos listos para su uso. La plataforma permite a investigadores, estudiantes y desarrolladores acceder fácilmente a datos reales para el desarrollo y evaluación de modelos.

En este proyecto, se utilizó Kaggle como una de las fuentes de datos, obteniéndose información que sirvió para el entrenamiento y validación de los modelos de lenguaje implementados.

## 2.4 Metodologia

### 2.4.1 CRISP-DM

CRISP-DM es una metodología estándar utilizada para organizar y guiar los proyectos de minería de datos de manera estructurada y práctica. Su nombre significa Cross-Industry Standard Process for Data Mining y fue diseñada para aplicarse a distintos tipos de industria, sin depender de una herramienta o sector específico. Esta metodología propone un proceso cíclico que permite mejorar continuamente el análisis de datos, y identifica seis fases principales: comprensión del negocio, comprensión de los datos, preparación de los datos, modelado, evaluación y despliegue. Una de sus principales ventajas es que no sigue un orden rígido, ya que el proyecto puede avanzar y retroceder entre las fases según los resultados obtenidos, facilitando así una mejor adaptación a los objetivos del negocio y a la realidad de los datos (de Ville, 2001, p. 37)

### 2.4.2 Fases de CRISP-DM

#### 2.4.2.1 Comprensión del negocio

Esta fase es el punto de partida del proyecto. Aquí se define qué problema se quiere resolver usando los datos y qué se espera lograr. También se aclaran los objetivos, se revisan los recursos disponibles y se organiza un plan de trabajo. Si esta parte no queda clara desde el inicio, el proyecto puede desviarse, aunque el modelo funcione bien *(*Hotz, 2023*)*.

#### 2.4.2.2 Comprensión de los datos

En esta etapa se empieza a trabajar directamente con los datos. Se recopila la información disponible, se revisa cómo están estructurados los datos y se exploran para entenderlos mejor. El objetivo es detectar errores, valores faltantes o cualquier problema que pueda afectar el proyecto más adelante *(*Hotz, 2023*)*.

#### 2.4.2.3 Preparación de los datos

Aquí es donde se invierte más tiempo. Los datos se limpian, se ordenan y se transforman para que puedan usarse sin problemas en los modelos. También se pueden crear nuevas variables a partir de la información existente. Una buena preparación de los datos hace que los resultados finales sean mucho más confiables *(*Hotz, 2023*)*.

#### 2.4.2.4 Modelado

En esta fase se prueban diferentes técnicas de minería de datos para identificar la que mejor se ajuste al problema del proyecto. Se construyen uno o varios modelos y se comparan sus resultados. Aunque es la parte más llamativa, suele ser más breve que la preparación de los datos *(*Hotz, 2023*)*.

#### 2.4.2.5 Evaluación

Aquí se revisa si el modelo realmente cumple con lo que se quería desde el inicio del proyecto. No solo importa que funcione técnicamente bien, sino que los resultados tengan sentido y ayuden a resolver el problema planteado *(*Hotz, 2023*)*.

#### 2.4.2.6 Implementación

Es la última fase del proyecto. Consiste en usar los resultados obtenidos, ya sea mediante reportes o gráficos, o integrando el modelo en una aplicación *(*Hotz, 2023*)*.

# 3. Desarrollo de la metodología

## 3.1 Comprensión del negocio

### 3.1.1 Objetivo del negocio

El objetivo del negocio de este proyecto es apoyar la identificación de noticias falsas en español mediante una herramienta tecnológica, que permita a los usuarios evaluar la veracidad de contenidos informativos de forma rápida y accesible. Para ello, se busca comparar distintos modelos de lenguaje preentrenados y seleccionar el que presente el mejor desempeño, con el fin de integrarlo en la web para facilitar su uso por personas sin conocimientos técnicos.

### 3.1.2 Objetivo de minería

El objetivo de minería es entrenar, evaluar y comparar modelos de lenguaje preentrenados (DeBERTa, RoBERTa y DistilBERT) para clasificar textos de noticias en español como falsas o verdaderas, utilizando técnicas de aprendizaje profundo y métricas de evaluación que permitan determinar cuál modelo ofrece mejores resultados para su posterior implementación.

## 3.2 Compresión de los datos

Para este proyecto se utilizaron varios datasets públicos, obtenidos de plataformas como Kaggle, GitHub, Hugging Face y Zenodo, que contienen noticias clasificadas como reales o falsas. La mayor parte de los datos se encuentra en español; sin embargo, también se incluyó un conjunto de noticias reales en inglés, que será tratado y ajustado durante la fase de preparación de los datos.

El uso de diferentes fuentes permite trabajar con textos de diversos temas, estilos de redacción y longitudes, lo que contribuye a que los modelos de lenguaje aprendan patrones más variados y no se enfoquen únicamente en un solo tipo de noticia.

### 3.2.1 Datasets Utilizados

#### 3.2.1.1 Fake News Corpus Español

Corpus en español con noticias etiquetadas como reales o falsas, cubriendo categorías como ciencia, deportes, economía, educación, entretenimiento, política, salud, seguridad y sociedad. Es útil porque ofrece variedad temática y ya viene con etiquetas claras.

El dataset cuenta con 572 registros (filas) y 7 variables (columnas), en la siguiente tabla se muestra el nombre de variable, la descripción, el tipo de dato.

***Tabla 1*** *Descripción de variables del corpus Fake News Corpus Español*

| Variable | Descripción | Tipo de dato |
|---|---|---|
| ID | Identificador único de cada noticia. Sirve solo para referencia o indexación. | int |
| CATEGORY | Indica si la noticia es verdadera o falsa. | bool |
| TOPICS | Tema principal de la noticia (por ejemplo: política, sociedad, salud, etc.). | string |
| SOURCE | Medio o portal de donde proviene la noticia. | string |
| HEADLINE | Título o encabezado de la noticia. Puede combinarse con el texto principal. | string |
| TEXT | Cuerpo o contenido principal de la noticia. | string |
| LINK | Enlace web original de la noticia. | string |

#### 3.2.1.2 Spanish Fake News Dataset

Este conjunto de datos contiene noticias falsas en español, recopiladas y organizadas con fines de investigación académica. Los textos se encuentran estructurados y anotados, lo que facilita su análisis y reduce problemas durante el preprocesamiento de los datos.

El dataset cuenta con 2552 registros (filas) y 7 variables (columnas), en la siguiente tabla se muestra el nombre de variable, la descripción, el tipo de dato.

***Tabla 2*** *Descripción de variables del Spanish Fake News Dataset*

| Variable | Descripción | Tipo de dato |
|---|---|---|
| Topic | Categoría temática de la noticia (por ejemplo: política, salud, COVID-19, sociedad, etc.). | string |
| Link source | Enlace a la noticia original, informe de verificación o fuente de la afirmación. Los enlaces inválidos fueron eliminados. | string |
| Media | Medio o plataforma donde apareció la afirmación falsa (por ejemplo: Facebook, YouTube, sitio web, etc.). | string |
| Date | Fecha de publicación o verificación de la noticia, en formato YYYY-MM-DD. | string |
| Author | Autor de la noticia o del contenido, si está disponible. Puede estar vacío. | string |
| Headlines | Título o resumen de la noticia o artículo que contiene la información falsa. | string |

| | | |
|---|---|---|
| Fake statement | Afirmación falsa o desinformación citada en el artículo de verificación. Es el texto principal para entrenar el modelo. | string |

### 3.2.1.3 Fakes Storage Dataset

Este dataset, alojado en GitHub, reúne noticias falsas en español e inglés obtenidas de distintas fuentes de Internet. Su principal aporte es la diversidad en la redacción de los textos, lo que ayuda a que los modelos no se adapten únicamente a un solo estilo de noticia.

El dataset cuenta con 10554 registros (filas) y 4 variables (columnas). En la siguiente tabla se muestran el nombre de la variable, la descripción y el tipo de dato.

***Tabla 3*** *Descripción de variables de Fakes Storage Dataset*

| Variable | Descripción | Tipo de dato |
|---|---|---|
| Id | Identificador numérico único de cada noticia dentro del dataset. | int |
| Titulo | Título o encabezado de la noticia o artículo verificado. Puede contener el texto principal de análisis. | string |
| Link | Enlace a la fuente original o al artículo de verificación | string |
| source | Archivo o fuente del dataset de donde proviene la noticia util para identificar el origen del dato. | string |

### 3.2.1.4 Spanish Political Fake News Dataset

Este conjunto de datos está enfocado en noticias políticas en español, etiquetadas como falsas (1) o reales (0). Es relevante para el proyecto porque la desinformación política es uno de los ámbitos donde más circulan noticias falsas, especialmente en redes sociales.

El dataset cuenta con 57231 registros (filas) y 5 variables (columnas). En la siguiente tabla se muestran el nombre de la variable, la descripción y el tipo de dato.

***Tabla 4*** *Descripción de variables de Spanish Political Fake News Dataset*

| Variable | Descripción | Tipo de dato |
|---|---|---|
| Id | Identificador único de cada noticia dentro del dataset. | string |
| news_url | URL del artículo de la noticia o del sitio web que publicó la noticia. | string |
| Title | Título del artículo de noticias. Contiene el encabezado principal de la noticia. | string |
| tweet_ids | Lista de identificadores de tweets que comparten la noticia. Los IDs están separados por tabulaciones dentro del campo. | string |

### 3.2.1.5 News Data Dataset

Este dataset contiene noticias reales en inglés recopiladas del canal de noticias AriseTV. Aunque los textos están en otro idioma y solo incluyen noticias reales, se utilizan como apoyo para reforzar la clase de noticias reales en el proceso de entrenamiento.

El dataset cuenta con 5514 registros (filas) y 3 variables (columnas). En la siguiente tabla se muestran el nombre de la variable, la descripción y el tipo de dato.

***Tabla 5*** *Descripción de variables de News Data Dataset*

| Variable | Descripción | Tipo de dato |
|---|---|---|
| Title | Título de la noticia. Resume el contenido principal del artículo. | string |
| Excerpt | Extracto o resumen corto del contenido de la noticia. | string |
| Category | Categoría temática de la noticia, como sports, business, politics, health. | string |

## 3.3 Preparación de los datos

Se realiza la exploración de los datos utilizando el lenguaje de programación Python y sus librerías correspondientes, con el fin de realizar un análisis detallado de las variables más relevantes que se emplearán en los modelos.

**Fake News Corpus Español**

***Ilustración 4*** *Primeras filas del dataset Fake News Corpus Español*

```
import pandas as pd

df = pd.read_csv("1. Fake News Corpus Español.csv")
df.head()
```

| | ID | CATEGORY | TOPICS | SOURCE | HEADLINE | TEXT | LINK |
|---|---|---|---|---|---|---|---|
| 0 | 1 | True | Covid-19 | El Economista | Covid-19: mentiras que matan | El control de la Covid-19 no es sólo un tema d... | https://www.eleconomista.com.mx/opinion/Covid-... |
| 1 | 2 | False | Política | El matinal | El Gobierno podrá acceder a las IPs de los móv... | El Gobierno de Pedro Sánchez y Pablo Iglesias ... | https://www.elmatinal.com/espana-ultima-hora/e... |
| 2 | 3 | True | Política | El País | La comunidad musulmana catalana denuncia a Vox... | Las tres federaciones que agrupan al 90% de la... | https://elpais.com/espana/elecciones-catalanas... |
| 3 | 4 | False | Política | AFPFactual | NaN | Se han dado a conocer los datos electorales pr... | https://perma.cc/GYE6-SPMB |
| 4 | 5 | True | Sociedad | La Republica | El censo poblacional 2018 tendrá un costo de $... | La primera fase del censo será virtual y solo ... | https://www.larepublica.co/economia/el-censo-p... |

***Ilustración 5*** *Valores nulos del dataset Fake News Corpus Español*

```
Valores nulos por columna:
ID           0
CATEGORY     0
TOPICS       0
SOURCE       7
HEADLINE    72
TEXT         0
LINK         3
dtype: int64
```

El dataset presenta valores nulos en SOURCE, HEADLINE y LINK. Los más relevantes son los 72 vacíos en HEADLINE y los 7 en SOURCE, ya que pueden reducir la calidad del texto que el modelo analiza. Para solucionarlo, se reemplazarán los titulares faltantes por las 20 primeras palabras del texto.

***Ilustración 6*** *Proceso de reemplazo de headlines nulos con texto inicial*

**Rellenar campo Headlines nulos con las 20 primeras palabras del texto.**

```
import pandas as pd

# Reemplazar los titulares vacíos con las primeras 20 palabras del texto
df['HEADLINE'] = df.apply(
    lambda x: ' '.join(x['TEXT'].split()[:20]) if pd.isna(x['HEADLINE']) else x['HEADLINE'],
    axis=1
)
```

```
df.head()
```

| | ID | CATEGORY | TOPICS | SOURCE | HEADLINE | TEXT | LINK |
|---|---|---|---|---|---|---|---|
| 0 | 1 | True | Covid-19 | El Economista | Covid-19: mentiras que matan | El control de la Covid-19 no es sólo un tema d... | https://www.eleconomista.com.mx/opinion/Covid-... |
| 1 | 2 | False | Política | El matinal | El Gobierno podrá acceder a las IPs de los móv... | El Gobierno de Pedro Sánchez y Pablo Iglesias ... | https://www.elmatinal.com/espana-ultima-hora/e... |
| 2 | 3 | True | Política | El País | La comunidad musulmana catalana denuncia a Vox... | Las tres federaciones que agrupan al 90% de la... | https://elpais.com/espana/elecciones-catalanas... |
| 3 | 4 | False | Política | AFPFactual | Se han dado a conocer los datos electorales pr... | Se han dado a conocer los datos electorales pr... | https://perma.cc/GYE6-SPMB |
| 4 | 5 | True | Sociedad | La Republica | El censo poblacional 2018 tendrá un costo de $... | La primera fase del censo será virtual y solo ... | https://www.larepublica.co/economia/el-censo-p... |

En el caso de SOURCE, se completaron los valores vacíos utilizando la información del enlace asociado (LINK), extrayendo el nombre del sitio web correspondiente, como Libertad Digital, Facebook, BBC, Perma.cc o Archive.is. Los 3 enlaces vacíos no afectan el entrenamiento, por lo que el campo LINK puede omitirse en el preprocesamiento.

***Ilustración 7*** *Función de detección y relleno automático de fuentes basado en enlaces*

**Funcion para rellenar Campo Source con su Link.**

```
def detectar_fuente(link):
    if pd.isna(link):
        return None
    link = link.lower()
    if "libertaddigital" in link:
        return "Libertad Digital"
    elif "facebook.com" in link:
        return "Facebook"
    elif "bbc.com" in link:
        return "BBC"
    elif "perma.cc" in link:
        return "Perma.cc"
    elif "archive.is" in link:
        return "Archive.is"
    else:
        return "Fuente desconocida"

df.loc[df['SOURCE'].isnull(), 'SOURCE'] = df[df['SOURCE'].isnull()]['LINK'].apply(detectar_fuente)
```

Verificamos si el dataset tiene valores nulos.

***Ilustración 8*** *Verificación de valores nulos*

```
print("\nValores nulos por columna:")
print(df.isnull().sum())
```

```
Valores nulos por columna:
ID          0
CATEGORY    0
TOPICS      0
SOURCE      0
HEADLINE    0
TEXT        0
LINK        3
```

## Spanish Fake News Dataset

***Ilustración 9*** *Primeras filas del dataset Spanish Fake News Dataset*

```
import pandas as pd

df = pd.read_csv("2. Spanish Fake News Dataset.csv")
df.head()
```

| | Topic | Link source | Media | Date | author | Headlines | Fake statement |
|---|---|---|---|---|---|---|---|
| 0 | Society | NaN | 12 Minutos | 2021-01-12 | NaN | NaN | Ingresada una influencer madrileña por hipoter... |
| 1 | Politics | https://123ru.net/foreign/193374933/ | 123ru.net | 2019-03-27 | 123ru.net | López Obrador insiste con España y pide crear ... | El objetivo, afirma, sería "hacer una relatorí... |
| 2 | COVID-19 | NaN | 12minutos | 2020-12-18 | NaN | NaN | Inglaterra se sume en el caos" después de que ... |
| 3 | Celebrities | http://www.12minutos.com/58054f67915c9/dylan-r... | 12minutos | NaN | NaN | Dylan rechaza el Nobel de Literatura | Luego de varios días de infructuosos intentos ... |
| 4 | Legislation | http://www.12minutos.com/58273561a837e/los-her... | 12minutos | NaN | NaN | NaN | Los hermanos mayores pagaran 75€ al mes a sus ... |

***Ilustración 10*** *Valores nulos del dataset Spanish Fake News Dataset*

```
Valores nulos por columna:
Topic                0
Link source       1345
Media              598
Date              1175
author            1640
Headlines         1856
Fake statement       0
```

En este proceso se realizará una limpieza y mejora del dataset. Primero, los valores faltantes en la columna Headlines se completarán con las primeras 20 palabras del texto en Fake statement, de modo que todas las noticias tengan un titular representativo. Finalmente, se conservarán solo las columnas más relevantes, Headlines y Fake statement, generando un dataset más limpio y listo para el entrenamiento del modelo.

***Ilustración 11*** *Relleno de headlines nulos con primeras 20 palabras del Fake statement*

```
#1. Rellenar Headlines con las primeras 20 palabras de Fake statement si está vacío
df['Headlines'] = df.apply(
    lambda x: ' '.join(str(x['Fake statement']).split()[:20]) if pd.isna(x['Headlines']) else x['Headlines'],
    axis=1
)
```

Finalmente, se conservarán solo las columnas más relevantes, Headlines y Fake statement, generando un dataset más limpio y listo para el entrenamiento del modelo.

***Ilustración 12*** *Dataset final con columnas Headlines y Fake statement para entrenamiento*

```
df.head()
```

| | Headlines | Fake statement |
|---|---|---|
| 0 | Ingresada una influencer madrileña por hipoter... | Ingresada una influencer madrileña por hipoter... |
| 1 | López Obrador insiste con España y pide crear ... | El objetivo, afirma, sería “hacer una relatorí... |
| 2 | Inglaterra se sume en el caos” después de que ... | Inglaterra se sume en el caos” después de que ... |
| 3 | Dylan rechaza el Nobel de Literatura | Luego de varios días de infructuosos intentos ... |
| 4 | Los hermanos mayores pagaran 75€ al mes a sus ... | Los hermanos mayores pagaran 75€ al mes a sus ... |

***Ilustración 13*** *Verificación final de valores nulos por columna tras el preprocesamiento*

```
Valores nulos por columna:
Topic                0
Link source       1345
Media              598
Date              1175
author            1640
Headlines            0
Fake statement       0
```

**Fakes Storage Dataset**

***Ilustración 14*** *Primeras y últimas de las filas del dataset Fakes Storage Dataset*

```
import pandas as pd

df = pd.read_csv("3. Fakes Storage Dataset.csv")

print("🔹 Primeras filas del dataset:")
display(df.head())

print("\n🔹 Últimas filas del dataset:")
display(df.tail())
```

🔹 Primeras filas del dataset:

| | id | titulo | link | source |
|---|---|---|---|---|
| 0 | 0 | No, los datos de positivos de un estudio de la... | https://maldita.es/malditobulo/20211220/complu... | fakenewsMaldita.json |
| 1 | 1 | La imagen de la cafeteria del Congreso de los ... | https://maldita.es/malditobulo/20211220/cafete... | fakenewsMaldita.json |
| 2 | 2 | Como los antivacunas han utilizado el problema... | https://maldita.es/malditobulo/20211220/kun-ag... | fakenewsMaldita.json |
| 3 | 3 | No, Mexico no permite modificar el acta de nac... | https://maldita.es/malditobulo/20211220/mexico... | fakenewsMaldita.json |
| 4 | 4 | No, esta ilustracion del artista Walter Molino... | https://maldita.es/malditobulo/20211220/pintur... | fakenewsMaldita.json |

🔹 Últimas filas del dataset:

| | id | titulo | link | source |
|---|---|---|---|---|
| 10549 | 3269 | Republican Economist Asks 'Retraction' – But ... | https://www.factcheck.org/2003/12/republican-e... | fakenewsFactCheck.json |
| 10550 | 3270 | Puncturing a Republican Tax Fable | https://www.factcheck.org/2003/12/puncturing-a... | fakenewsFactCheck.json |
| 10551 | 3271 | Facts Take a Bath at Democratic Debate | https://www.factcheck.org/2003/12/facts-take-a... | fakenewsFactCheck.json |
| 10552 | 3272 | Liberal Group Attacks Dean on Gun Control | https://www.factcheck.org/2003/12/liberal-grou... | fakenewsFactCheck.json |
| 10553 | 3273 | Attack Ad by Anti-tax Group Too Close for Dean... | https://www.factcheck.org/2003/12/attack-ad-by... | fakenewsFactCheck.json |

***lustración 15*** *Valores nulos del dataset Fakes Storage Dataset*

```
Valores nulos por columna:
id        0
titulo    0
link      0
source    0
dtype: int64
```

En este dataset se identificó la presencia de registros con información en inglés, que fueron traducidos al español. No se encontraron datos nulos durante el proceso de exploración; por lo tanto, no fue necesario aplicar técnicas de imputación o de eliminación de registros.

***Ilustración 16*** *Código de traducción automática de registros en inglés dentro del dataset*

```python
import pandas as pd
from deep_translator import GoogleTranslator

# Cargar dataset
df = pd.read_csv("/content/3. Fake News Dataset Español.csv")

# Mostrar primeras filas
print("🔹 Primeras filas del dataset:")
display(df.head())

# Inicializar traductor
translator = GoogleTranslator(source="en", target="es")


# Traducir SOLO la columna 'titulo'
df_tail["titulo"] = df_tail["titulo"].apply(
    lambda x: translator.translate(str(x))
)

# Mostrar últimas filas
print("🔹 Últimas filas del dataset ):")
display(df_tail)
```

***Ilustración 17*** *Muestra del dataset con títulos traducidos al español (primera y última filas)*

🔹 Primeras filas del dataset:

| | id | titulo | link | source |
|---|---|---|---|---|
| 0 | 0 | No, los datos de positivos de un estudio de la... | https://maldita.es/malditobulo/20211220/complu... | fakenewsMaldita.json |
| 1 | 1 | La imagen de la cafeteria del Congreso de los ... | https://maldita.es/malditobulo/20211220/cafete... | fakenewsMaldita.json |
| 2 | 2 | Como los antivacunas han utilizado el problema... | https://maldita.es/malditobulo/20211220/kun-ag... | fakenewsMaldita.json |
| 3 | 3 | No, Mexico no permite modificar el acta de nac... | https://maldita.es/malditobulo/20211220/mexico... | fakenewsMaldita.json |
| 4 | 4 | No, esta ilustracion del artista Walter Molino... | https://maldita.es/malditobulo/20211220/pintur... | fakenewsMaldita.json |

🔹 Últimas filas del dataset (títulos traducidos al español):

| | id | titulo | link | source |
|---|---|---|---|---|
| 10544 | 3264 | ¿Era Wesley Clark republicano? | https://www.factcheck.org/2004/01/was-wesley-c... | fakenewsFactCheck.json |
| 10545 | 3265 | Incluso sus opositores consideran que el anunc... | https://www.factcheck.org/2004/01/even-opponen... | fakenewsFactCheck.json |
| 10546 | 3266 | ¿Bush está abusando de personas mayores con be... | https://www.factcheck.org/2004/01/is-bush-abus... | fakenewsFactCheck.json |
| 10547 | 3267 | Mentiras y errores en el debate demócrata | https://www.factcheck.org/2004/01/fibs-and-flu... | fakenewsFactCheck.json |
| 10548 | 3268 | El ataque de Gephardt a "Enron" no da en el bl... | https://www.factcheck.org/2003/12/gephardt-enr... | fakenewsFactCheck.json |
| 10549 | 3269 | Un economista republicano pide una "retracción... | https://www.factcheck.org/2003/12/republican-e... | fakenewsFactCheck.json |
| 10550 | 3270 | Perforando una fábula fiscal republicana | https://www.factcheck.org/2003/12/puncturing-a... | fakenewsFactCheck.json |
| 10551 | 3271 | Los hechos se bañan en el debate demócrata | https://www.factcheck.org/2003/12/facts-take-a... | fakenewsFactCheck.json |
| 10552 | 3272 | Grupo liberal ataca a Dean por control de armas | https://www.factcheck.org/2003/12/liberal-grou... | fakenewsFactCheck.json |
| 10553 | 3273 | El anuncio de ataque del grupo anti-impuestos ... | https://www.factcheck.org/2003/12/attack-ad-by... | fakenewsFactCheck.json |

La traducción de los textos permitió unificar el idioma de los datos, generando así un dataset más limpio, consistente y adecuado para el entrenamiento del modelo.

**Spanish Political Fake News Dataset**

***Ilustración 18*** *Primeras filas del dataset Spanish Political Fake News Dataset*

```
import pandas as pd

df = pd.read_csv("4. Spanish Political Fake News Dataset .csv")
# Mostramos las primeras filas
display(df.head())
```

| | ID | Label | Titulo | Descripcion | Fecha |
|---|---|---|---|---|---|
| 0 | ID | 1 | Moreno intenta apaciguar el flanco sanitario m... | El presidente abre la puerta a unos comicios e... | 19/04/2022 |
| 1 | ID | 1 | La Abogacía del Estado se retira como acusació... | En un escrito, la abogada del Estado Rosa Marí... | 17/09/2021 |
| 2 | ID | 0 | Las promesas incumplidas de Pablo Echenique en... | Este lunes y martes la Asamblea de Madrid acog... | 12/09/2022 |
| 3 | ID | 1 | Sánchez defiende 'resolver el problema' de la ... | Resulta evidente que la ley ha tenido algunos ... | 07/02/2023 |
| 4 | ID | 1 | Ian Gibson cierra la lista electoral de la con... | El hispanista, que ya ocupó un puesto simbólic... | 12/04/2023 |

***Ilustración 19*** *Valores nulos del dataset Spanish Political Fake News Dataset*

```
Valores nulos por columna:
ID             0
Label          0
Titulo         0
Descripcion    0
Fecha          0
```

Durante el proceso de análisis, se decidió no usar la variable Fecha, ya que no aporta información relevante para el objetivo del modelo. Al evaluar la estructura del dataset, se constató que no existen valores nulos ni inconsistencias en los tipos de datos, por lo que, tras esta depuración, el dataset se encuentra limpio y listo para el análisis y el entrenamiento del modelo.

**News Data Dataset**

***Ilustración* 20** Primeras filas del dataset News Data Dataset

```
import pandas as pd

df = pd.read_csv("news_data_dataset.csv")
df.head()
```

| | Title | Excerpt | Category |
|---|---|---|---|
| 0 | Uefa Opens Proceedings against Barcelona, Juve... | Uefa has opened disciplinary proceedings again... | sports |
| 1 | Amazon Blames Inflation as It Increases Cost o... | The increases are steeper than the 17 percent ... | business |
| 2 | Nigeria's Parliament Passes Amended Electoral ... | Nigeria's Senate on Tuesday passed the harmoni... | politics |
| 3 | Nigeria: Lagos Governor Tests Positive for Cov... | The Lagos State Governor, Mr. Babajide Sanwo-O... | health |
| 4 | South Africa Calls For Calm as Electoral Refor... | South Africa has raised concerns about the det... | politics |

***Ilustración 21*** *Valores nulos del dataset News Data Dataset*

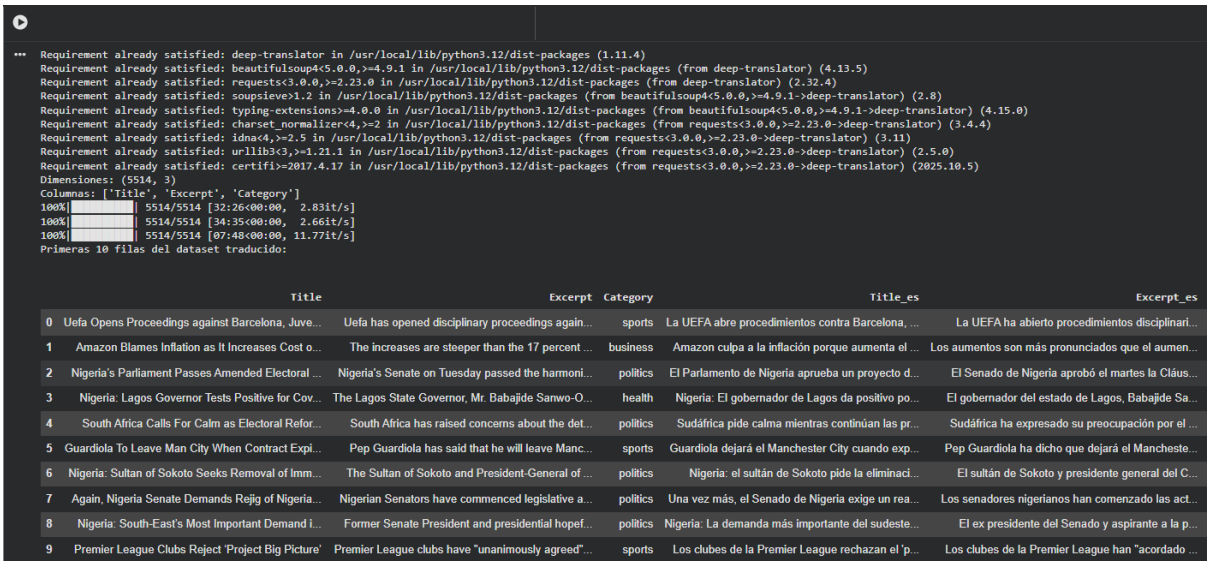
```
Valores nulos por columna:
Title       0
Excerpt     0
Category    0
```

Toda la información de este dataset se encontraba originalmente en idioma inglés y, durante la exploración de los datos, se verificó que no existen valores nulos en ninguna de las columnas (Title, Excerpt y Category).

***Ilustración 22*** *Proceso de traducción automática de títulos, extractos y categorías al español*

```
!pip install deep-translator
from deep_translator import GoogleTranslator
import pandas as pd
from tqdm import tqdm

# Cargar el dataset
df = pd.read_csv("news_data_complete.csv")

# Mostrar información básica
print("Dimensiones:", df.shape)
print("Columnas:", df.columns.tolist())

# Activar barra de progreso
tqdm.pandas()

# --- Traducir columnas al español ---
df['Title_es'] = df['Title'].progress_apply(
    lambda x: GoogleTranslator(source='auto', target='es').translate(x) if pd.notnull(x) else x
)

df['Excerpt_es'] = df['Excerpt'].progress_apply(
    lambda x: GoogleTranslator(source='auto', target='es').translate(x) if pd.notnull(x) else x
)

df['Category_es'] = df['Category'].progress_apply(
    lambda x: GoogleTranslator(source='auto', target='es').translate(x) if pd.notnull(x) else x
)

# --- Mostrar las primeras 10 filas traducidas ---
print("\nPrimeras 10 filas del dataset traducido:\n")
display(df.head(10))
```

***Ilustración 23*** *Resultado de la traducción automática de Title, Excerpt y Category al español*



```
Requirement already satisfied: deep-translator in /usr/local/lib/python3.12/dist-packages (1.11.4)
Requirement already satisfied: beautifulsoup4<5.0.0,>=4.9.1 in /usr/local/lib/python3.12/dist-packages (from deep-translator) (4.13.5)
Requirement already satisfied: requests<3.0.0,>=2.23.0 in /usr/local/lib/python3.12/dist-packages (from deep-translator) (2.32.4)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.12/dist-packages (from beautifulsoup4<5.0.0,>=4.9.1->deep-translator) (2.8)
Requirement already satisfied: typing-extensions>=4.0.0 in /usr/local/lib/python3.12/dist-packages (from beautifulsoup4<5.0.0,>=4.9.1->deep-translator) (4.15.0)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.23.0->deep-translator) (3.4.4)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.23.0->deep-translator) (3.11)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.23.0->deep-translator) (2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.23.0->deep-translator) (2025.10.5)
Dimensiones: (5514, 3)
Columnas: ['Title', 'Excerpt', 'Category']
100%|██████████| 5514/5514 [32:26<00:00,  2.83it/s]
100%|██████████| 5514/5514 [34:35<00:00,  2.66it/s]
100%|██████████| 5514/5514 [07:48<00:00, 11.77it/s]
Primeras 10 filas del dataset traducido:
```

| | Title | Excerpt | Category | Title_es | Excerpt_es |
|---|---|---|---|---|---|
| 0 | Uefa Opens Proceedings against Barcelona, Juve... | Uefa has opened disciplinary proceedings again... | sports | La UEFA abre procedimientos contra Barcelona, ... | La UEFA ha abierto procedimientos disciplinari... |
| 1 | Amazon Blames Inflation as It Increases Cost o... | The increases are steeper than the 17 percent ... | business | Amazon culpa a la inflación porque aumenta el ... | Los aumentos son más pronunciados que el aumen... |
| 2 | Nigeria's Parliament Passes Amended Electoral ... | Nigeria's Senate on Tuesday passed the harmoni... | politics | El Parlamento de Nigeria aprueba un proyecto d... | El Senado de Nigeria aprobó el martes la Cláus... |
| 3 | Nigeria: Lagos Governor Tests Positive for Cov... | The Lagos State Governor, Mr. Babajide Sanwo-O... | health | Nigeria: El gobernador de Lagos da positivo po... | El gobernador del estado de Lagos, Babajide Sa... |
| 4 | South Africa Calls For Calm as Electoral Refor... | South Africa has raised concerns about the det... | politics | Sudáfrica pide calma mientras continúan las pr... | Sudáfrica ha expresado su preocupación por el ... |
| 5 | Guardiola To Leave Man City When Contract Expi... | Pep Guardiola has said that he will leave Manc... | sports | Guardiola dejará el Manchester City cuando exp... | Pep Guardiola ha dicho que dejará el Mancheste... |
| 6 | Nigeria: Sultan of Sokoto Seeks Removal of Imm... | The Sultan of Sokoto and President-General of ... | politics | Nigeria: el sultán de Sokoto pide la eliminaci... | El sultán de Sokoto y presidente general del C... |
| 7 | Again, Nigeria Senate Demands Rejig of Nigeria... | Nigerian Senators have commenced legislative a... | politics | Una vez más, el Senado de Nigeria exige un rea... | Los senadores nigerianos han comenzado las act... |
| 8 | Nigeria: South-East's Most Important Demand i... | Former Senate President and presidential hopef... | politics | Nigeria: La demanda más importante del sudeste... | El ex presidente del Senado y aspirante a la p... |
| 9 | Premier League Clubs Reject 'Project Big Picture' | Premier League clubs have "unanimously agreed"... | sports | Los clubes de la Premier League rechazan el 'p... | Los clubes de la Premier League han "acordado ... |

Para facilitar su uso en el modelo, los textos se tradujeron al español. Además, se eliminó la columna de categoría, ya que no resultó relevante para el objetivo del modelo. De esta forma, se obtuvo un conjunto de datos más homogéneo, limpio y preparado para el entrenamiento del modelo.

**Unión y Creación de un Dataset Global Etiquetado Listo para el Entrenamiento**

En primer lugar, se cargaron todos los datasets ya preprocesados y se procedió a la estandarización de los nombres de las columnas, unificando las variables que representan el título, la descripción y la etiqueta de clasificación.

***Ilustración 24*** *Carga de múltiples datasets de noticias falsas en español para integración*

```python
import pandas as pd

# === 1. Cargar todos los datasets ===
df1 = pd.read_csv("1. Fake News Corpus Español_limpio.csv")
df2 = pd.read_csv("2. Spanish Fake News Dataset_limpio.csv")
df4 = pd.read_csv("4. Spanish Political Fake News Dataset .csv")
df5 = pd.read_csv("5.FakesStorage_Limpio_Traducido.csv")
df6 = pd.read_csv("news_data_dataset_traducido_Limpio.csv")
```

En algunos datasets, la variable de clasificación se encontraba representada como valores booleanos en formato texto, por lo que se realizó una limpieza de estos valores y se

transformaron a una codificación numérica, donde el valor 1 representa noticias verdaderas y 0 noticias falsas

***Ilustración 25*** *Normalización y mapeo de columnas en el primer dataset*

```
# === 2. Normalizar columnas según su estructura ===

# --- df1: contiene "CATEGORY" con 'True'/'False'  ---
df1 = df1.rename(columns={
    "CATEGORY": "Label",
    "HEADLINE": "Titulo",
    "TEXT": "Descripcion"
})

# Limpiar valores de Label y mapear correctamente
df1["Label"] = df1["Label"].astype(str).str.strip().str.lower()
df1["Label"] = df1["Label"].map({
    "true": 1,
    "false": 0
})
```

Posteriormente, se trataron de forma individual aquellos datasets que contenían únicamente noticias falsas o únicamente noticias verdaderas, asignando la etiqueta correspondiente a todos sus registros. En otros casos, fue necesario corregir la interpretación original de las etiquetas, ya que algunos conjuntos de datos utilizaban una codificación inversa. Este ajuste permitió mantener la coherencia en la definición de las clases a lo largo de todo el conjunto de datos integrado.

Una vez normalizadas las etiquetas y los nombres de las variables, se seleccionaron únicamente las columnas relevantes para el estudio: la etiqueta de clasificación, el título y la descripción de la noticia. A continuación, todos los datasets fueron unificados en un solo conjunto de datos, eliminando valores nulos

***Ilustración 26*** *Proceso de estandarización, fusión y limpieza de múltiples datasets*

```python
# Eliminar filas sin Label (por si hay valores vacíos o extraños)
df1 = df1.dropna(subset=["Label"])

# --- df2: todas falsas ---
df2 = df2.rename(columns={"Headlines": "Titulo", "Fake statement": "Descripcion"})
df2["Label"] = 0

# --- df4: ya tiene Label correcto ---
df4 = df4.rename(columns={"Titulo": "Titulo", "Descripcion": "Descripcion", "Label": "Label"})

# --- df5: invertir etiquetas (originalmente 0=real, 1=fake) ---
df5 = df5.rename(columns={"title_es": "Titulo", "label": "Label"})
df5["Descripcion"] = df5["Titulo"]
df5["Label"] = df5["Label"].apply(lambda x: 0 if x == 1 else 1)

# --- df6: todas verdaderas ---
df6 = df6.rename(columns={"Title_es": "Titulo", "Excerpt_es": "Descripcion"})
df6["Label"] = 1

# === 3. Seleccionar columnas consistentes ===
dfs = [
    df1[["Label", "Titulo", "Descripcion"]],
    df2[["Label", "Titulo", "Descripcion"]],
    df4[["Label", "Titulo", "Descripcion"]],
    df5[["Label", "Titulo", "Descripcion"]],
    df6[["Label", "Titulo", "Descripcion"]]
]

# === 4. Unir ===
df_total = pd.concat(dfs, ignore_index=True)

# === 5. Limpiar ===
df_total.dropna(subset=["Label", "Titulo", "Descripcion"], inplace=True)
df_total.drop_duplicates(subset=["Titulo", "Descripcion"], inplace=True)
```

Posteriormente, se incorporó una columna de identificación única para cada registro, lo que facilita la gestión y la referencia de las noticias dentro del dataset. Con el conjunto de datos completo ya estructurado, se aplicó un proceso de balanceo de clases para evitar sesgos durante el entrenamiento del modelo. Para ello, se seleccionó el mismo número de noticias verdaderas y falsas mediante un muestreo aleatorio controlado.

***Ilustración 27*** *Asignación de ID y equilibrio de clases*

```python
# === 6. Crear ID incremental ===
df_total.insert(0, "ID", range(1, len(df_total) + 1))

# === 7. Balancear ===
fake = df_total[df_total["Label"] == 0]
true = df_total[df_total["Label"] == 1]
min_len = min(len(fake), len(true))
df_balanced = pd.concat([
    fake.sample(min_len, random_state=42),
    true.sample(min_len, random_state=42)
]).sample(frac=1, random_state=42).reset_index(drop=True)
```

Finalmente, se generaron dos versiones del dataset: una versión completa, que contiene todos los registros disponibles tras la integración, y una versión balanceada, diseñada específicamente para el entrenamiento de modelos de clasificación.

***Ilustración 28*** *Guardado y verificación final del dataset balanceado*

```
# === 8. Guardar ===
df_total.to_csv("dataset_noticias_completo.csv", index=False, encoding="utf-8")
df_balanced.to_csv("dataset_noticias_balanceado.csv", index=False, encoding="utf-8")

# === 9. Verificación ===
print("✅ Dataset completo y balanceado listos\n")

print("📊 Conteo de etiquetas en dataset completo:")
print(df_total["Label"].value_counts(), "\n")

print("📊 Conteo de etiquetas en dataset balanceado:")
print(df_balanced["Label"].value_counts(), "\n")

print("📄 Total de registros:")
print("Dataset completo:", len(df_total))
print("Dataset balanceado:", len(df_balanced), "\n")

print("🔎 Valores nulos por columna:")
print(df_total.isnull().sum())
```

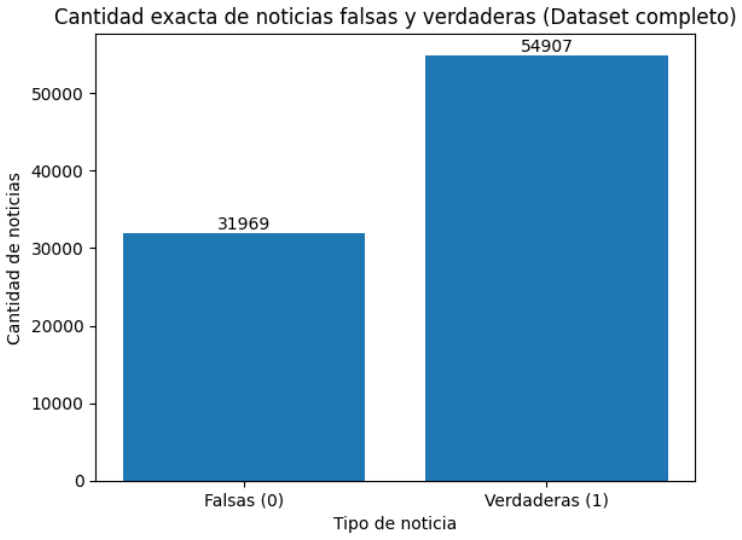
Ambas versiones fueron almacenadas en formato CSV y se verificó que no existían valores nulos ni inconsistencias en los datos. Este proceso permitió obtener un conjunto de datos limpio, coherente y adecuado para el desarrollo del modelo.

***Ilustración 29*** *Estadísticas descriptivas finales del dataset balanceado*

```
✅ Dataset completo y balanceado listos

📊 Conteo de etiquetas en dataset completo:
Label
1    54907
0    31969
Name: count, dtype: int64

📊 Conteo de etiquetas en dataset balanceado:
Label
1    31969
0    31969
Name: count, dtype: int64

📄 Total de registros:
Dataset completo: 86876
Dataset balanceado: 63938

🔎 Valores nulos por columna:
ID             0
Label          0
Titulo         0
Descripcion    0
dtype: int64
```

El gráfico muestra la cantidad exacta de noticias falsas y verdaderas presentes en el dataset completo. Se observa que existen 31,969 noticias falsas y 54,907 noticias verdaderas, lo que evidencia una distribución desigual entre las clases. Esta diferencia indica un claro desbalance en los datos, en el que las noticias verdaderas están significativamente más representadas que las falsas.

***Ilustración 30*** *Distribución de noticias falsas vs. verdaderas en el dataset completo*




Debido a este desbalance, se decidió aplicar un proceso de balanceo del dataset para evitar que el modelo aprenda de forma sesgada hacia la clase mayoritaria. Al equilibrar la cantidad de noticias falsas y verdaderas, se busca mejorar el desempeño del modelo y garantizar una clasificación más justa y precisa durante el entrenamiento.

## 3.4 Modelado

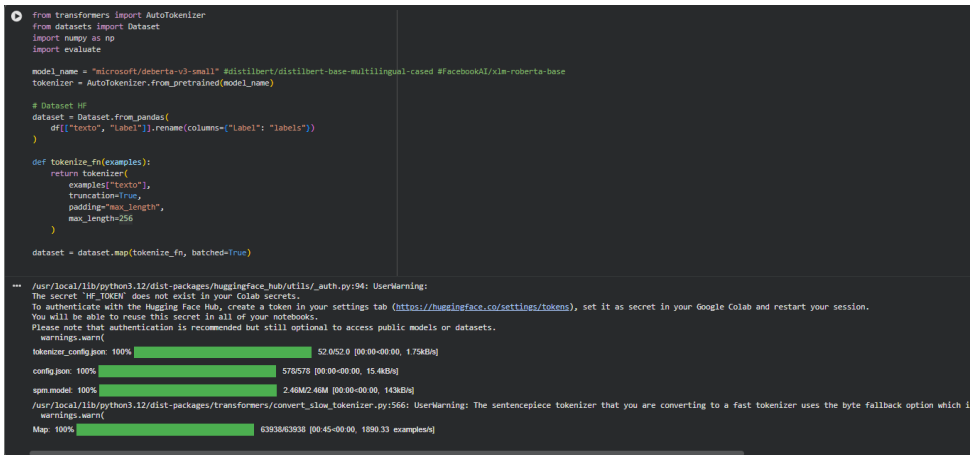
En esta fase se evaluaron los tres modelos preentrenados en Google Colab como entorno de trabajo. Para el entrenamiento y la evaluación se utilizó una GPU T4, lo que permitió que los modelos se entrenaran más rápido y que el proceso de validación cruzada se realizara de forma más eficiente.

Antes de comenzar con el entrenamiento de los modelos, es necesario iniciar sesión en Hugging Face, ya que la mayoría de los modelos preentrenados se descargan directamente de su repositorio oficial. Al autenticarse con una cuenta, se obtiene acceso completo a los modelos públicos y se evitan restricciones al descargarlos varias veces durante el entrenamiento.

***Ilustración 31*** *Proceso de login y tokenización en Hugging Face Hub*


En este bloque se prepara el texto para que el modelo pueda entenderlo. Primero se elige el modelo preentrenado DeBERTa, DistilBERT o RoBERTa y se carga su tokenizer. El tokenizer convierte el texto en tokens, que básicamente son números que representan palabras o partes de palabras. Luego se crea el dataset usando solo el texto y la etiqueta, y se renombra la columna a labels porque así lo necesita Hugging Face., al final, el tokenizer se aplica a todo el dataset, dejando los datos listos para entrenar y evaluar los modelos.

***Ilustración 32*** *Tokenizer y conversión de texto a tokens para DeBERTa, DistilBERT y RoBERTa*




En este bloque se definen las métricas más importantes que se usarán para evaluar y comparar los modelos. Estas métricas permiten ver qué tan bien clasifica cada modelo, no solo en términos de aciertos generales, sino también en su capacidad de distinguir entre noticias reales y falsas.

***Ilustración 33*** *Definición de métricas de evaluación accuracy, precisión, recall, F1, ROC-AUC*

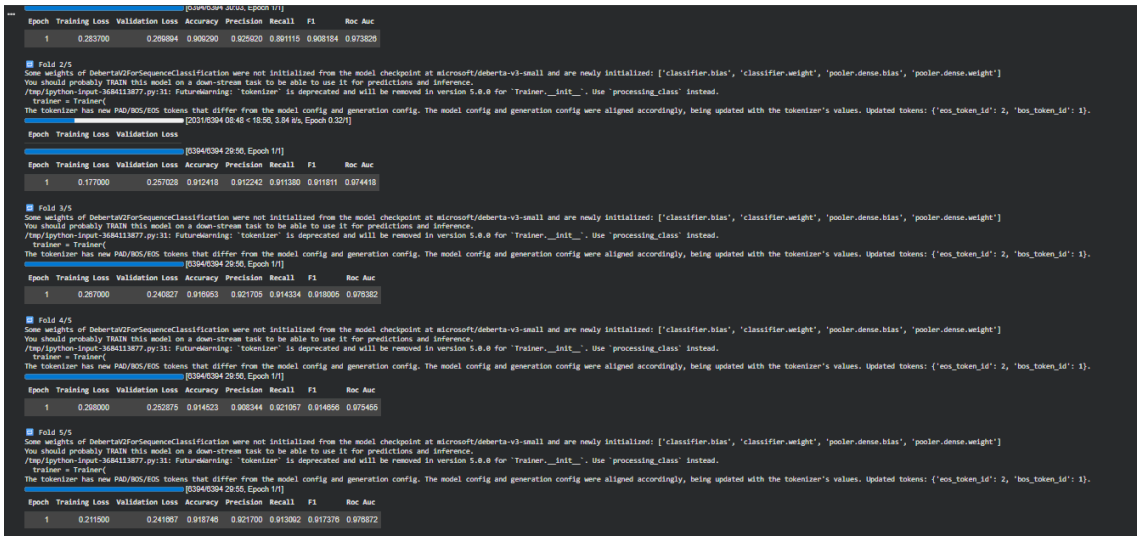

Aquí se utiliza validación cruzada con 5 partes para evaluar el modelo de forma más objetiva. El dataset se divide en 5 grupos y el modelo se entrena varias veces, usando un grupo distinto para validar en cada vuelta.

Se usó solo un epoch porque el modelo ya viene entrenado. Con una sola pasada por los datos basta para ajustarlo al problema y comparar los modelos.

***Ilustración 34*** *Entrenamiento con validación cruzada de 5 folds usando Trainer de Hugging Face*

```python
from sklearn.model_selection import KFold
from transformers import AutoModelForSequenceClassification, Trainer, TrainingArguments
from scipy.special import softmax

kf = KFold(n_splits=5, shuffle=True, random_state=42)

for fold, (train_idx, val_idx) in enumerate(kf.split(dataset)):
    print(f"\n Fold {fold+1}/5")

    train_dataset = dataset.select(train_idx.tolist())
    val_dataset = dataset.select(val_idx.tolist())

    model = AutoModelForSequenceClassification.from_pretrained(
        model_name,
        num_labels=2
    )

    args = TrainingArguments(
        output_dir=f"./mdeberta_fold_{fold}",
        eval_strategy="epoch",
        save_strategy="no",
        learning_rate=2e-5,
        per_device_train_batch_size=8,
        per_device_eval_batch_size=8,
        num_train_epochs=1,
        weight_decay=0.01,
        logging_steps=50,
        report_to=[]
    )

    trainer = Trainer(
        model=model,
        args=args,
        train_dataset=train_dataset,
        eval_dataset=val_dataset,
        tokenizer=tokenizer,
        compute_metrics=compute_metrics
    )

    # Entrenar
    trainer.train()

    # Métricas del fold
    metrics = trainer.evaluate()
    cv_results.append(metrics)

    # Predicciones para matriz global
       # Predicciones para métricas globales
    preds = trainer.predict(val_dataset)

    probs = softmax(preds.predictions, axis=1)

    y_true_all.extend(preds.label_ids)
    y_pred_all.extend(np.argmax(probs, axis=1))
    y_prob_all.extend(probs[:, 1])
```

***Ilustración*** 35 Log de entrenamiento mostrando métricas en cada iteración del fold




## 3.5 Evaluación

### 3.5.1 Análisis de resultados

***Tabla 6*** *Métricas de evaluación de los modelos de clasificación*

| Modelo | Accuracy | Precision | Recall | F1-score | ROC-AUC |
|---|---|---|---|---|---|
| **DeBERTa** | 0.91 | **0.92** | 0.91 | **0.91** | **0.9751** |
| **RoBERTa** | **0.92** | 0.91 | **0.92** | 0.91 | 0.9738 |
| **DistilBERT** | 0.91 | 0.91 | 0.91 | 0.91 | 0.9738 |

La tabla 6 presenta los resultados de evaluación de los tres modelos.

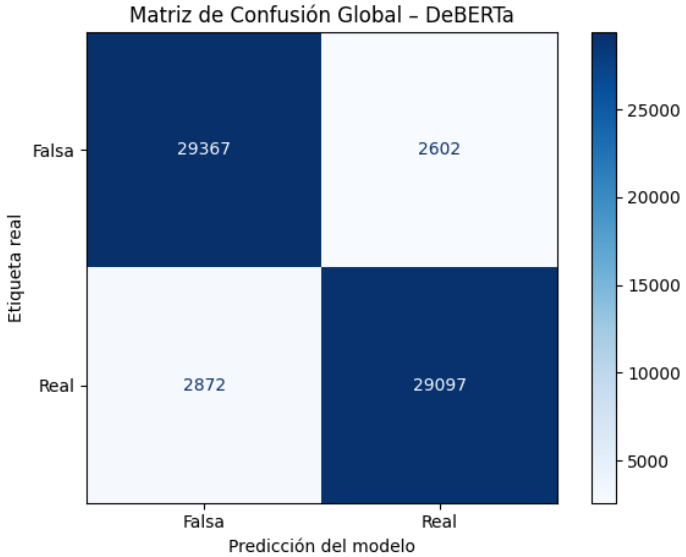
### 3.5.2 Resultados DeBERTa

***Ilustración 36*** *Matriz de confusión, reporte de clasificación y ROC-AUC global para DeBERTa*

```
Matriz de Confusión Global (Validación Cruzada)
[[29367  2602]
 [ 2872 29097]]

Reporte de Clasificación Global
              precision    recall  f1-score   support

       Falsa       0.91      0.92      0.91     31969
        Real       0.92      0.91      0.91     31969

    accuracy                           0.91     63938
   macro avg       0.91      0.91      0.91     63938
weighted avg       0.91      0.91      0.91     63938


ROC-AUC Global (Validación Cruzada)
0.9751208649461933
```

La matriz de confusión de DeBERTa muestra que el modelo clasificó correctamente 29,367 noticias falsas, que son las que realmente eran falsas y el modelo acertó al identificarlas, y 29,097 noticias reales, que eran verdaderas y también fueron clasificadas correctamente. Hubo algunos errores: 2,602 noticias reales fueron clasificadas como falsas, lo que indica pocas alarmas falsas, y 2,872 noticias falsas se marcaron como reales, lo que significa que algunas noticias falsas pasaron desapercibidas.

***Ilustración 37*** *Gráfico de matriz de confusión de DeBERTa*



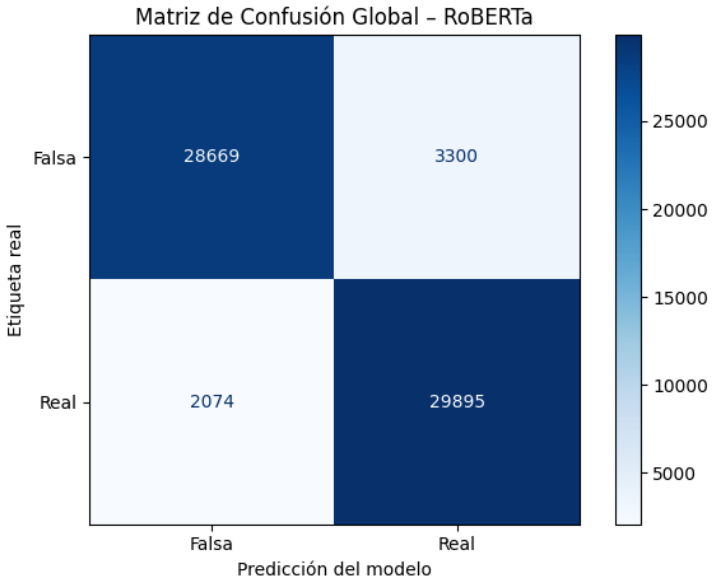
### 3.5.3 Resultados RoBERTa

***Ilustración* 38** Matriz de confusión, reporte de clasificación y ROC-AUC global para RoBERTa

```
Matriz de Confusión Global (Validación Cruzada)
[[28669  3300]
 [ 2074 29895]]

Reporte de Clasificación Global
              precision    recall  f1-score   support

       Falsa       0.93      0.90      0.91     31969
        Real       0.90      0.94      0.92     31969

    accuracy                           0.92     63938
   macro avg       0.92      0.92      0.92     63938
weighted avg       0.92      0.92      0.92     63938


ROC-AUC Global (Validación Cruzada)
0.9737648463546389
```

La matriz de confusión de RoBERTa muestra que el modelo clasificó correctamente 28,669 noticias falsas, que eran falsas en la realidad, y 29,895 noticias reales, que eran verdaderas y también fueron identificadas correctamente. Cometió algunos errores: 3,300 noticias reales se clasificaron como falsas, generando algunas falsas alarmas, y 2,074 noticias falsas se marcaron como reales, lo que significa que algunas noticias falsas pasaron desapercibidas.

***Ilustración 39*** *Gráfico de matriz de confusión de RoBERTa*



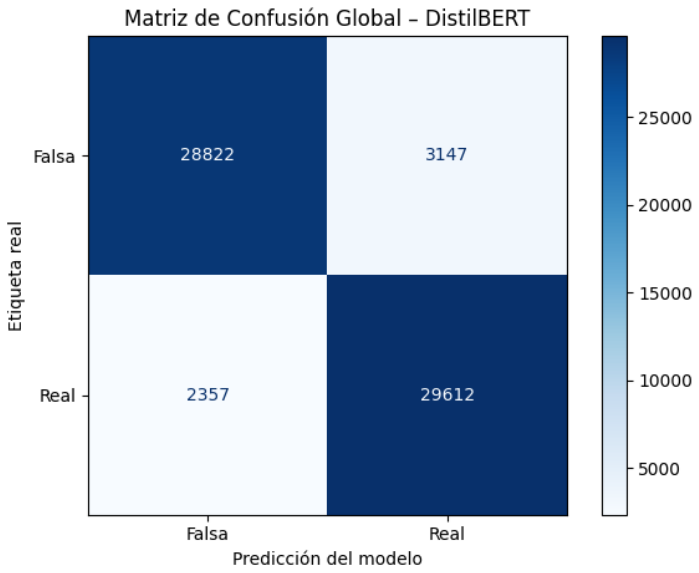
### 3.5.4 Resultados DistilBERT

***Ilustración 40*** *Matriz de confusión, reporte de clasificación y ROC-AUC global para DistilBERT*

```
Matriz de Confusión Global (Validación Cruzada)
[[28822  3147]
 [ 2357 29612]]

Reporte de Clasificación Global
              precision    recall  f1-score   support

       Falsa       0.92      0.90      0.91     31969
        Real       0.90      0.93      0.91     31969

    accuracy                           0.91     63938
   macro avg       0.91      0.91      0.91     63938
weighted avg       0.91      0.91      0.91     63938


ROC-AUC Global (Validación Cruzada)
0.9738268203750484
```

La matriz de confusión de DistilBERT muestra que el modelo clasificó correctamente 28,822 noticias falsas, que realmente eran falsas, y 29,612 noticias reales, que también fueron identificadas correctamente. Cometió algunos errores: 3,147 noticias reales se marcaron como falsas, generando falsas alarmas, y 2,357 noticias falsas se clasificaron como reales, dejando pasar algunas noticias falsas.

***Ilustración 41*** *Matriz de confusión, reporte de clasificación y ROC-AUC global para DistilBERT*



Aunque los tres modelos funcionan bastante bien y tienen métricas muy similares, DeBERTa se destaca ligeramente en precisión y ROC-AUC, lo que indica que detecta un poco mejor tanto noticias falsas como reales. Además, su arquitectura más moderna le permite capturar mejor el contexto del lenguaje en español, lo que hace que los errores sean un poco menores que los de RoBERTa y DistilBERT. Por eso, para la implementación de la aplicación final se decidió usar DeBERTa, ya que ofrece un buen equilibrio y asegura que el sistema de verificación de noticias funcione de manera confiable.

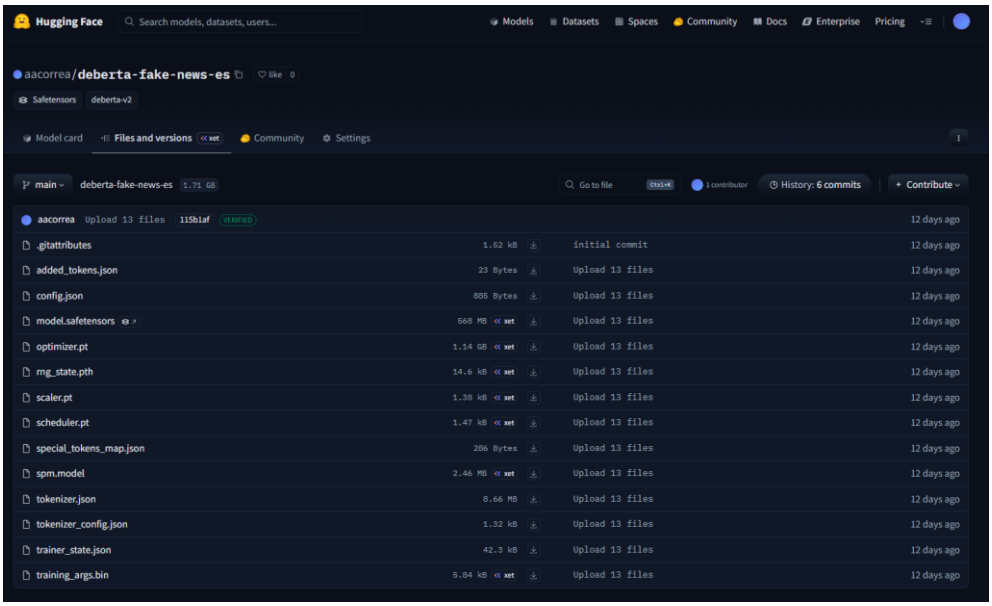
## 3.6 Implementación

En esta sección se explica cómo se implementó el modelo entrenado y cómo se hizo mediante la plataforma Hugging Face, mediante una aplicación web sencilla.

### 3.6.1 Publicación del modelo en Hugging Face

Luego de finalizar el entrenamiento y la evaluación del modelo DeBERTa, el modelo resultante fue subido a la plataforma Hugging Face para reutilizarlo y probarlo de manera práctica. El repositorio del modelo sigue la estructura estándar de Hugging Face, lo que garantiza que el modelo funcione correctamente al cargar. Entre los archivos más importantes se encuentran el archivo model.safetensors, que contiene los pesos entrenados, y los archivos del tokenizador, que permiten procesar correctamente los textos en español.

***Ilustración 42*** *Repositorio del modelo DeBERTa publicado en Hugging Face Hub*



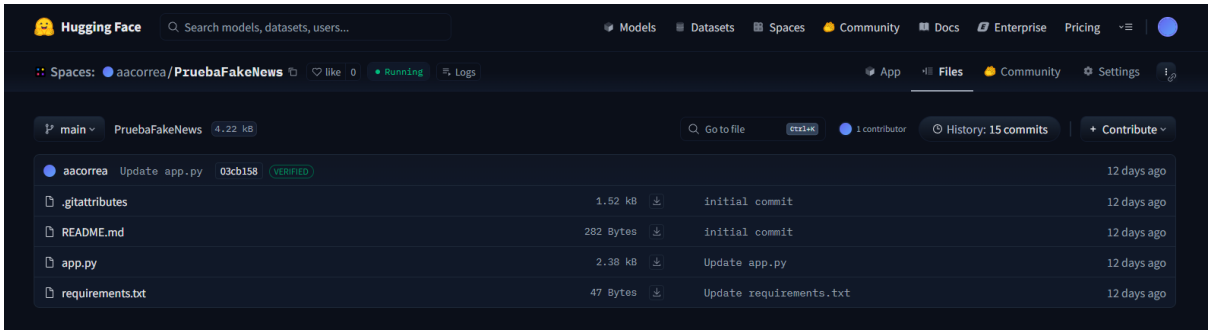
### 3.6.3 Implementación de la aplicación web

Para demostrar el funcionamiento del modelo de forma más visual y fácil de usar, se creó una aplicación web sencilla con Hugging Face Spaces. Esta aplicación permite al usuario ingresar un texto y obtener como resultado si es una noticia falsa o verdadera.

El Space cuenta con una estructura simple, compuesta principalmente por los siguientes archivos:

- app.py: contiene la lógica principal de la aplicación.
- requirements.txt: define las librerías necesarias para que la aplicación funcione.README.
- md: incluye una breve descripción del Space.

***Ilustración 43*** *Espacio en Hugging Face con la aplicación web de detección de noticias falsas*



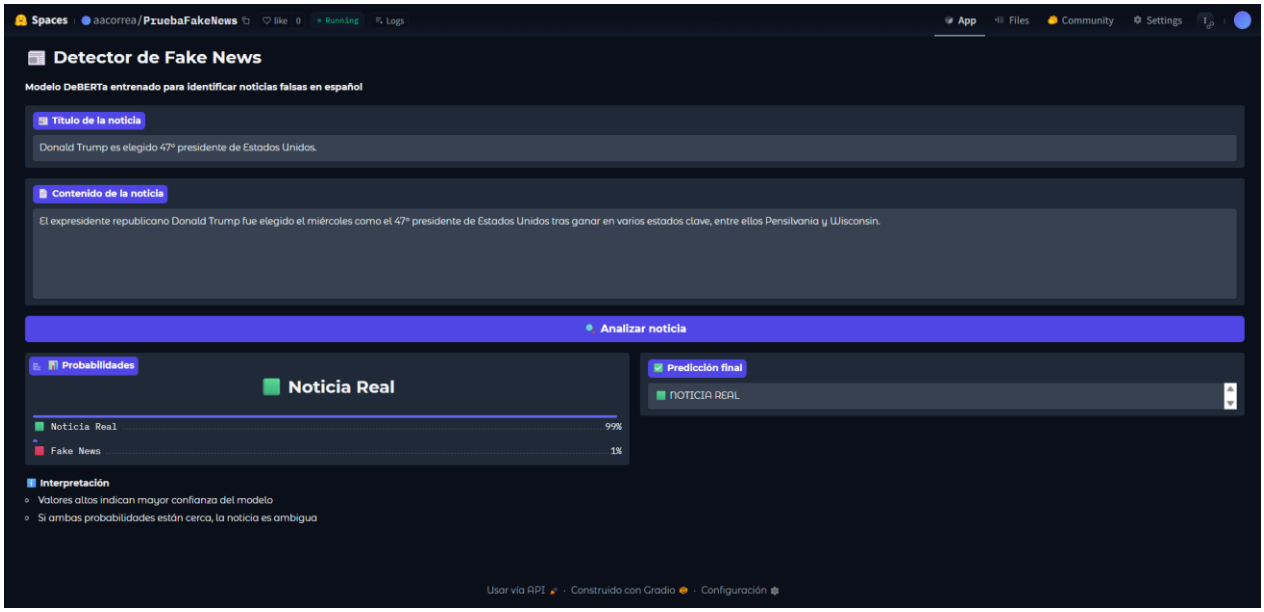

### 3.6.5 Uso de la aplicación

Al estar desplegada en Hugging Face Spaces, la aplicación puede utilizarse directamente en un navegador web, sin necesidad de instalar programas adicionales. Esto permite comprobar de forma práctica el desempeño del modelo entrenado.

Para utilizarla, el usuario debe ingresar el título de la noticia y el contenido del texto en los campos correspondientes.

Una vez ingresada la información, se presiona el botón Analizar noticia, y el sistema procesa el texto utilizando el modelo DeBERTa entrenado.

***Ilustración 44*** *Interfaz de la aplicación web en Hugging Face Spaces para detección de fake news*



# 4. Conclusiones y Recomendaciones

## 4.1 Conclusiones

- El uso de la metodología CRISP-DM ayudó a organizar todo el desarrollo del proyecto, desde la definición del problema hasta la implementación final. Seguir estas fases permitió tener claro qué se quería lograr en cada etapa y facilitó la toma de decisiones durante el entrenamiento y la evaluación de los modelos.
- Durante la preparación de los datos se pudo comprobar que la calidad de los datasets es clave para obtener buenos resultados. La limpieza, unificación, traducción y balanceo de los datos fueron pasos necesarios para evitar sesgos y asegurar que los modelos aprendieran a partir de información consistente.
- Aun cuando las métricas fueron cercanas, DeBERTa destacó ligeramente en precisión y ROC-AUC, lo que indica una mejor capacidad para clasificar correctamente las noticias. Por esta razón, fue seleccionado como el modelo final para la implementación del prototipo.
- La implementación del modelo en una aplicación web mediante Hugging Face Spaces permitió comprobar de forma práctica el funcionamiento del sistema. El prototipo desarrollado demuestra que es posible llevar modelos de aprendizaje profundo a aplicaciones accesibles
- En general, este proyecto demuestra que el uso de modelos de lenguaje preentrenados es una solución viable y efectiva para la detección automática de noticias falsas en español, y deja una base sólida para futuros trabajos que busquen mejorar el modelo o ampliar el prototipo.

## 4.2 Recomendaciones

- Se recomienda contar con adecuados recursos de cómputo, ya que los modelos de lenguaje preentrenados utilizados requieren una buena capacidad de procesamiento, especialmente durante la fase de entrenamiento y ajuste. El uso de GPU puede facilitar y acelerar este proceso.
- Ampliar y actualizar los datasets utilizados, incorporando datos más recientes y de diferentes fuentes, ya que la forma en la que se generan las noticias falsas cambia constantemente, sobre todo en redes sociales.

- Usar el prototipo como una herramienta de apoyo y no como la única forma de verificación, promoviendo siempre el análisis crítico de la información por parte de los usuarios.
- Es recomendable realizar un mejor balance de los datos entre noticias reales y falsas, ya que una distribución desigual puede afectar el desempeño del modelo y generar sesgos en las predicciones.
- Optimizar el prototipo web para mejorar la experiencia del usuario, haciendo la interfaz más clara y agregando mensajes sencillos que expliquen el resultado de la predicción

# 5. Bibliografía


alcorpas10. (2025). *GitHub - alcorpas10/FakesStorage: This is a spanish dataset for fake news detection*. GitHub. https://github.com/alcorpas10/FakesStorage

Cortez Vásquez, M. A., Vega Huerta, M. H., Jaime, L., & Quispe, P. (2009). Procesamiento de lenguaje natural. *Revista de Ingeniería de Sistemas e Informática, 6*(2), 47–52. https://d1wqtxts1xzle7.cloudfront.net/77493941/5121-libre.pdf

Daniel. (2021, diciembre 14). *Kaggle: todo lo que hay que saber sobre esta plataforma*. DataScientest.

de Ville, B. (2001). *Microsoft Data Mining: Integrated Business Intelligence for E-Commerce and Knowledge Management*. Reino Unido: Elsevier Science.

DistilBERT. (2019). Huggingface.co. https://huggingface.co/docs/transformers/model_doc/distilbert

GitHub. (2023). *About GitHub*. https://github.com/about

Google. (2023). *Google Colaboratory*. https://colab.research.google.com/

Hotz, N. (2023, enero 19). *What is CRISP-DM?* Data Science Process Alliance. https://www.datascience-pm.com/crisp-dm-2/

Hugging Face. (2023). *Transformers documentation*. https://huggingface.co/docs/transformers

Luna, J. C. (2024, septiembre 11). *¿Qué es el BERT? Introducción a los modelos BERT*. DataCamp. https://www.datacamp.com/es/blog/what-is-bert-an-intro-to-bert-models

María Grandury. (2021). *fake_news_corpus_spanish*. Huggingface.co. https://huggingface.co/datasets/mariagrandury/fake_news_corpus_spanish

microsoft/mdeberta-v3-base · Hugging Face. (2021). Huggingface.co. https://huggingface.co/microsoft/mdeberta-v3-base

Pereyras, A. (2015). *¿Qué es el Aprendizaje Profundo?* Nuevas pedagogías para el cambio educativo. https://d1wqtxts1xzle7.cloudfront.net/87348741/AP_ale-pereyras-libre.pdf

Raheja, S. (2023, marzo 20). *DeBERTa V3: The Most Recent Member of DeBERTa Family of Generative AI Models*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2023/03/deberta-v3-the-most-recent-member-of-deberta-family-of-generative-ai-models/

Rodríguez Pérez, C. (2019). No diga fake news, di desinformación: una revisión sobre el fenómeno de las noticias falsas y sus implicaciones. *Comunicación, 40*, 65–74. https://doi.org/10.18566/comunica.n40.a05

RoBERTa. (2019). Huggingface.co. https://huggingface.co/docs/transformers/model_doc/roberta

Samuel, O. C. (2025, noviembre 24). *news-data*. Huggingface.co. https://huggingface.co/datasets/okite97/news-data

Stryker, C. (2025, julio 21). *Modelo preentrenado*. IBM. https://www.ibm.com/es-es/think/topics/pretrained-model

Tretiakov, A., Sergio, D. M., & Martín, A. (2021). *Spanish Fake News Dataset*. Zenodo (CERN European Organization for Nuclear Research). https://doi.org/10.5281/zenodo.15592391

Vizoso, J. O. (2017). *Spanish Political Fake News*. Kaggle.com. https://www.kaggle.com/datasets/javieroterovizoso/spanish-political-fake-news