

# Verifake

## Predicción de fake news

Beatriz Tranche Robles

VERIFAKE

---

**FAKE  
NEWS!**

# Introducción

## **Contexto:**

En un mundo saturado de información, la desinformación se ha convertido en un problema crítico

## **Objetivo del Proyecto:**

Desarrollar un sistema que detecte fake news  
Añadir análisis de datos, programa de webscraping para facilitar la obtención de datos y un generador de clickbaits

## **Estrategia:**

A través de modelos de machine learning , busco identificar y clasificar contenido engañoso.  
Facilitar la obtención de datos  
Generar engagement

# Contexto y motivación

## Importancia de la Detección de Fake News

1.

La proliferación de fake news amenaza la de la integridad de la información.

Los estudios muestran que el 70% de los españoles creen haber visto noticias falsas en las redes sociales.

### Generación de clickbaits:

2.

Maximizar el engagement y el tráfico

Aprovechar la tecnología, para ofrecer un enfoque innovador que beneficia tanto a los productores de contenido como a la audiencia.

### Webscraping:

3.

Permite la recolección automatizada de noticias falsas de diversas fuentes en línea.

Esto soluciona la dificultad de encontrar bases de datos adecuadas.

# Objetivos iniciales

accuracy			0.20	506
macro avg	0.04	0.06	0.04	506
weighted avg	0.14	0.20	0.15	506

**Desarrollar un modelo de deep learning para predecir las fuentes de noticias falsas**

**Responder a hipótesis como:**

**¿ Qué beneficios económicos son derivados de la generación de fake news?**

**¿Qué medios son mas reincidentes en la publicación de fake news?**

**¿Cuál es el sesgo característico de estos medios?**

Train  
accuracy  
is 98%

Test accuracy  
is 50%



La complejidad del modelo de deep learning requería más datos de los que tenía disponibles.

Las bases de datos existentes estaban desactualizadas o eran insuficientes.

Mis modelos eran más como 'noticias falsas' en sí mismos: confusos y poco confiables

**Dificultades en la Implementación**

**Falta de Datos Adecuados**

**Resultados Decepcionantes**

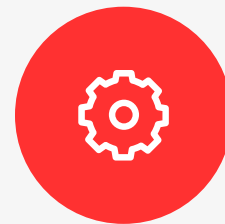
# Desafíos encontrados

# Cambio de Dirección



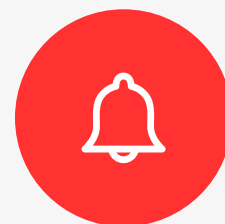
## Decisión de Cambio

Después de ver los resultados decepcionantes, comprendí que necesitaba ajustar mi enfoque. En lugar de centrarme en predecir las fuentes de noticias falsas, decidí abordar el problema más directamente: la detección de las propias fake news.



## Búsqueda de Bases de Datos:

Sin embargo, la búsqueda de bases de datos adecuadas se convirtió en otro desafío monumental. La frustración se apoderó de mí al descubrir que la mayoría de los conjuntos de datos estaban desactualizados o eran de acceso limitado.



## La Luz al Final del Túnel:

Fue un momento de 'Eureka' en el que comprendí que, si quería avanzar, tendría que crear mis propios conjuntos de datos.

# Desarrollo de Web Scraping

```
Codeium: Refactor | Explain | X
def scrape_page(page_number):
    """
    Scrapes data from a single page of the website.

    :param page_number: The number of the page to scrape.
    :return: A list of dictionaries with the scraped data.
    """
    driver = webdriver.Chrome()
    data = [] # Store the data for a single page
    cont = 0

    try:
        # Open the page
        driver.get(f'https://dbkf.ontotext.com/#!/searchViewResults?orderBy=score&lang=es&page={page_number}')

        # Wait until the h5 elements are present
        h5_elements = WebDriverWait(driver, 10).until(
            EC.presence_of_all_elements_located((By.CSS_SELECTOR, 'h5.claim-text'))
        )

        # Iterate over all h5 elements
        for index in range(len(h5_elements)):
            try:
                # Re-obtain the h5 elements to avoid stale element references
                h5_elements = WebDriverWait(driver, 10).until(
                    EC.presence_of_all_elements_located((By.CSS_SELECTOR, 'h5.claim-text'))
                )
            except:
                pass
            # Extract data from the h5 element
            data.append({
                'index': index,
                'text': h5_elements[index].text
            })

    except Exception as e:
        print(f"Error scraping page {page_number}: {e}")

    return data
```

Decisión de Crear un Scraper:

Al darme cuenta de que necesitaba un conjunto de datos, decidí crear un programa de web scraping.

Era mi única opción si quería acceder a un flujo constante de datos frescos y relevantes sobre noticias.

# Generador de Clickbait

## Implementación del Generador de Clickbait



## Creación del Generador

Este generador utiliza la API de OpenAI para crear titulares que generan engagement y atraen la atención de los lectores.



# Resultados Finales

Menú de Navegación

Predicción de Fake News

Scraping de Noticias

Análisis de Datos

Generador de Titulares

## Generador de Titulares Clickbait

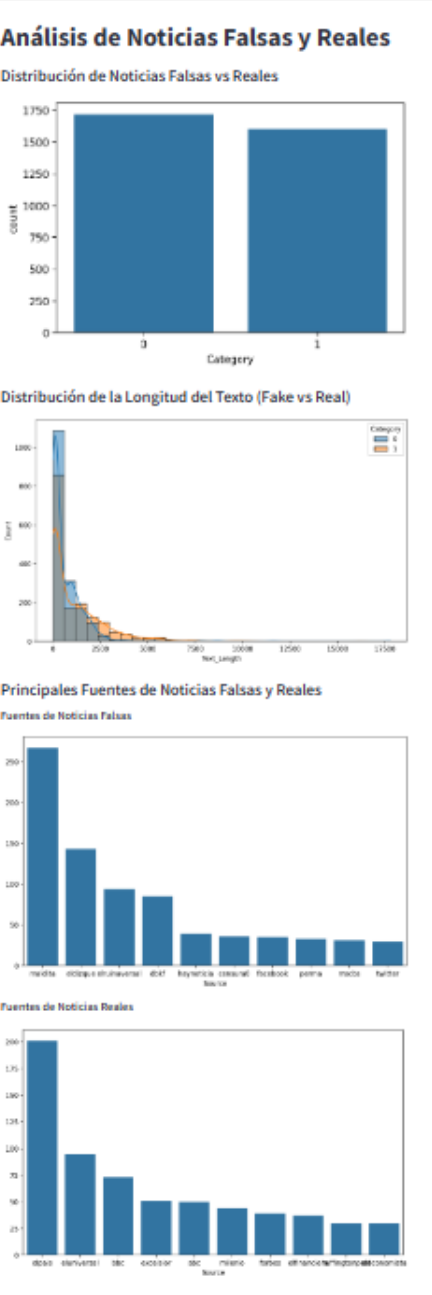
Introduce la noticia para el titular clickbait:

Íñigo Errejón ha anunciado este jueves que dimite como portavoz del Grupo Parlamentario de Sumar, q

Generar titular clickbait

Titular generado:

¡Escándalo en la política! Íñigo Errejón dimite tras acusaciones de violencia machista ¡No te pierdas los detalles!



Menú de Navegación

Predicción de Fake News

Scraping de Noticias

Análisis de Datos

Generador de Titulares

## Detección de Noticias Falsas

Introduce el texto de la noticia que deseas analizar:

Longitud del texto actual: 0 caracteres (máximo permitido: 1000).

Elige el modelo que deseas utilizar para la predicción:

Logistic Regression

Analizar Noticia

## Scraping de Noticias Falsas

Número de páginas

3

Número de workers (threads)

3

-

+

☐ ¿Deseas guardar los resultados en un archivo CSV?

Ejecutar Scraping

# Muchas gracias



[Enlace a proyecto en github](#)