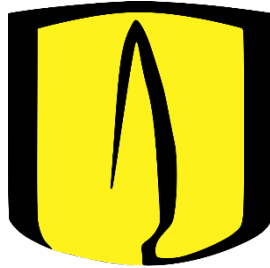


**Proyecto 1**  
**Herramientas de Analítica de Textos para la Identificación de Desinformación**  
**Política**



**Grupo 9**

**Carol Sofía Florido Castro - 202111430**

**Juan Martín Vásquez Cristancho - 202113314**

**Natalia Villegas Calderón – 202113370**

**Universidad de Los Andes**  
**Departamento de Ingeniería de Sistemas y Computación**  
**Inteligencia de Negocios - ISIS 3301**  
**Bogotá D.C., Colombia**

**2024**

## Tabla de contenido

<b>1. Introducción .....</b>	<b>3</b>
1.1. Definición del cliente .....	3
1.2. Documentación del proceso de aprendizaje automático.....	4
<b>2. Entendimiento y preparación de los datos .....</b>	<b>4</b>
2.1. Entendimiento de los datos.....	4
<b>3. Modelado y evaluación .....</b>	<b>7</b>
3.1. Modelo <i>Naive Bayes</i> (Natalia Villegas) .....	7
3.2. Modelo <i>Random Forest</i> (Carol Florido).....	8
3.3. Modelo KNN (Martin Vásquez) .....	10
3.4. Modelo Clasificador <i>Gradient Boosting</i> (Natalia Villegas) .....	12
3.5. Evaluación de métricas generalizadas.....	13
<b>4. Resultados .....</b>	<b>13</b>
4.1. Descripción de los resultados: Métricas y objetivos del negocio.....	13
4.2. Objetivo del negocio y selección del mejor modelo.....	14
4.3. Análisis de palabras .....	15
<b>5. Trabajo en equipo .....</b>	<b>16</b>
5.1. Roles y Tareas Realizadas por Cada Integrante .....	16
5.2. Reuniones de grupo .....	18
5.3. Uso de <i>ChatGPT</i> en el proyecto y otras NLP .....	19
5.4. Retos y soluciones .....	19

## 1. Introducción

Este proyecto tiene como objetivo desarrollar un sistema de clasificación de noticias falsas en el ámbito político mediante la implementación y comparación de diversos algoritmos de analítica de datos. Para lograrlo, se explorarán múltiples enfoques de aprendizaje automático, combinando modelos supervisados con técnicas avanzadas de procesamiento de lenguaje natural (NLP). El proceso iniciará con el entrenamiento de estos modelos utilizando un conjunto de datos previamente categorizado en noticias reales y falsas, lo que permitirá identificar patrones lingüísticos y características distintivas de la desinformación.





















Para evaluar el rendimiento de los modelos, se emplearán diversas métricas, como la precisión, el *recall* y la exactitud. No obstante, la métrica F1 será el criterio principal de comparación, ya que ofrece un equilibrio entre precisión (proporción de noticias correctamente clasificadas como falsas) y *recall* (proporción de noticias falsas detectadas con éxito). Este enfoque es crucial en problemas donde un sesgo en la clasificación podría comprometer la efectividad del sistema, garantizando que el modelo no solo sea preciso, sino también confiable en diferentes escenarios. Más allá de su relevancia técnica, este proyecto tiene un alto valor práctico, particularmente para plataformas digitales como Facebook, donde la difusión de noticias falsas puede alterar la percepción pública, influir en procesos electorales y fomentar la polarización social. La implementación de modelos robustos de detección permitiría a estas plataformas mejorar sus estrategias de moderación, reduciendo el impacto de la desinformación y fortaleciendo la confianza de los usuarios en la información que consumen.







### 1.1. Definición del cliente

El cliente de este proyecto son las plataformas digitales y redes sociales, especialmente aquellas que enfrentan el desafío de controlar la difusión de noticias falsas, como Facebook, Twitter y otras aplicaciones de contenido informativo. Estas plataformas tienen la responsabilidad de garantizar que la información presentada a sus usuarios sea veraz y confiable, evitando la propagación de desinformación que pueda influir en procesos electorales, generar desconfianza en las instituciones o intensificar la polarización social. Por lo cual, a través del desarrollo de modelos de analítica de datos para la clasificación de noticias, este proyecto busca proporcionar una solución que les permita mejorar sus estrategias de detección y mitigación de *fake news*, optimizando sus algoritmos de moderación de contenido. Al integrar modelos de aprendizaje automático que prioricen la precisión y el equilibrio en la detección de noticias falsas, se les ofrece una herramienta con la capacidad de adaptarse a nuevas tendencias en la desinformación y fortalecer sus sistemas de control sin comprometer la libertad de expresión ni generar sesgos en la moderación de contenido.

## 1.2. Documentación del proceso de aprendizaje automático.

Tabla 1 OWNML Machine Learning Canvas para la detección de noticias falsas

OWNML MACHINE LEARNING CANVAS								
Designed for: BI - Proyecto 1		Designed by: Grupo 9						
Date: Feb/2025		Iteration: 4						
<div>PREDICTION TASK</div> <div></div> <div><p>Este proyecto aborda una tarea de clasificación binaria supervisada para diferenciar entre noticias falsas y verídicas en el ámbito político. Utiliza modelos de procesamiento de lenguaje natural (NLP) para analizar atributos como el título, contenido y fuente, identificando patrones que permitan clasificar cada noticia en una de las dos categorías.</p><p>Los resultados se generan en tiempo real o por lotes, facilitando la detección temprana de desinformación en plataformas digitales. Esto optimiza la moderación de contenido y minimiza la propagación de noticias falsas. La evaluación del modelo se basa en métricas como precisión, recall y F1-score, priorizando un equilibrio entre la detección efectiva y la reducción de falsos positivos.</p></div> <div><div>DECISIONS</div><div></div><div><p>Los resultados del modelo se convierten en recomendaciones o decisiones procesables para el usuario final mediante un sistema de moderación automatizado e interfaces de gestión de contenido. Cuando el modelo clasifica una noticia como falsa, la plataforma puede tomar diversas acciones, como marcar la publicación con una advertencia, reducir su alcance en los feeds de los usuarios o incluso bloquear su difusión si infringe las políticas de desinformación. Además, los resultados pueden enviarse a verificadores de datos (fact-checkers) para un análisis más detallado antes de aplicar sanciones definitivas. Estas decisiones permiten a la plataforma controlar la propagación de fake news sin comprometer la libertad de expresión, asegurando que los usuarios reciban información más confiable.</p></div></div> <div><div>VALUE PROPOSITION</div><div></div><div><p>La propuesta de valor de este modelo de detección de noticias falsas radica en su capacidad para analizar grandes volúmenes de información y detectar patrones que indican posibles intentos de desinformación. Utilizando modelos de clasificación avanzados, como Random Forest junto con representaciones TF-IDF, el sistema evalúa la credibilidad de las noticias en función de su contenido y su estructura lingüística. Para Facebook, esta solución ofrece un mecanismo automatizado y escalable que ayuda a mitigar la propagación de información engañosa dentro de la plataforma, protegiendo la integridad del ecosistema digital. Su implementación permite a medios de comunicación, plataformas digitales y organismos de verificación fortalecer la confianza del público en la información que consumen, reduciendo el impacto de la manipulación mediática. Además, el modelo prioriza la reducción de falsos negativos mediante métricas como F2-score, garantizando que los noticias falsas sean detectadas con alta sensibilidad sin comprometer la experiencia del usuario con restricciones innecesarias sobre contenido legítimo.</p></div></div> <div><div>DATA COLLECTION</div><div></div><div><p>No se debe diligenciar</p></div></div> <div><div>DATA SOURCES</div><div></div><div><p>En este proyecto, los datos provienen de fuentes internas y externas. Inicialmente, se utilizó una base de datos de Kaggle con noticias falsas y verídicas, pero en un entorno real, es necesario integrar fuentes en tiempo real para mejorar la detección del modelo.</p><p>Para obtener datos actualizados, se pueden utilizar APIs de redes sociales como Twitter y Facebook Graph, junto con APIs de noticias como Guardian Open Platform y NYTimes Article Search, que proporcionan acceso a artículos históricos y recientes. Además, las plataformas digitales pueden aprovechar sus bases internas, donde almacenan interacciones de usuarios, etiquetas de moderación y reportes de desinformación. La combinación de estas fuentes permite mejorar la capacidad del modelo para adaptarse a nuevas estrategias de difusión de fake news.</p></div></div> <tr><td><div>IMPACT SIMULATION</div><div></div><div><p>Para evaluar el impacto del modelo antes del despliegue, se consideran los costos y beneficios de las decisiones correctas e incorrectas. Un modelo preciso mejora la moderación en plataformas digitales, reduciendo la propagación de desinformación y fortaleciendo la confianza de los usuarios. Sin embargo, los falsos positivos pueden generar censura injustificada, afectando la libertad de expresión, mientras que los falsos negativos permiten la difusión de noticias falsas, afectando la percepción pública. Estos errores tienen costos reputacionales y pueden influir en la retención de usuarios y en la credibilidad del sistema de moderación.</p><p>Para simular el impacto antes del despliegue, se utilizan datos históricos etiquetados, junto con métricas de engagement como interacciones y compartidos. Se aplican técnicas como validación cruzada y pruebas A/B para medir la efectividad del modelo en un entorno controlado. Los criterios de despliegue priorizan un equilibrio entre precisión y recall, optimizando la métrica F1. Además, se establecen restricciones de equidad para evitar sesgos en la clasificación, asegurando que el modelo no favorezca ni penalice desproporcionadamente ciertos grupos o fuentes de información (por ejemplo, empleando técnicas de balanceo de datos como SMOTE).</p></div></td><td><div>MAKING PREDICTIONS</div><div></div><div><p>El modelo de detección de noticias falsas puede implementarse tanto en procesamiento por lotes como en tiempo real, dependiendo de los requerimientos operativos de Facebook. En un enfoque por lotes, el sistema analiza periódicamente grandes volúmenes de noticias, permitiendo una verificación programada y eficiente de contenido publicado en la plataforma. Esta modalidad es útil para revisar bases de datos históricas o realizar auditorías de contenido en intervalos predefinidos. Por otro lado, en una implementación en tiempo real, el modelo procesa las noticias al momento de su publicación o interacción, identificando posibles intentos de desinformación de manera inmediata y permitiendo alertas o restricciones dinámicas antes de su difusión masiva. La frecuencia de uso dependerá de la demanda y del nivel de control deseado, pudiendo ajustarse desde verificaciones en tiempo real para contenido viral hasta análisis por lotes en horarios específicos para optimizar el rendimiento y la eficiencia computacional del sistema.</p></div></td><td><div>BUILDING MODELS</div><div></div><div><p>Se entrenaron cuatro modelos de clasificación: Naive Bayes, Random Forest, KNN y Gradient Boosting, evaluando su desempeño con métricas como exactitud, precisión, recall y F1-score. Random Forest obtuvo el mejor rendimiento, equilibrando precisión y recall, mientras que Gradient Boosting mostró buenos resultados con mayor costo computacional. Naive Bayes fue eficiente pero limitado por suposiciones de independencia, y KNN tuvo el peor desempeño.</p><p>Para mantener la efectividad del modelo, es necesario actualizarlo periódicamente con nuevos datos y ajustes, asegurando su precisión en la detección de noticias falsas a lo largo del tiempo.</p></div></td><td><div>FEATURES</div><div></div><div><p>El modelo de detección de noticias falsas utiliza como principales variables el título y la descripción de la noticia, transformados en representaciones numéricas a través de TF-IDF, lo que permite capturar la relevancia de cada término dentro del corpus. Antes de esta transformación, el texto pasa por un proceso de preprocesamiento, que incluye la eliminación de caracteres especiales, conversión a minúsculas, eliminación de stopwords, lematización y tokenización para mejorar la calidad de los datos y reducir el ruido. Además, se pueden considerar variables adicionales, como la fecha de publicación para identificar patrones temporales en la desinformación. Estas características son utilizadas por el modelo Random Forest, el cual aplica múltiples árboles de decisión para detectar patrones en los datos y clasificar la noticia como verdadera o falsa. La combinación de estas transformaciones asegura que el modelo pueda aprender de estructuras lingüísticas y patrones textuales comunes en noticias falsas, mejorando su capacidad predictiva y minimizando falsos negativos.</p></div></td></tr> <tr><td colspan="2"><div>MONITORING</div><div></div><div><p>No se debe diligenciar</p></div></td><td colspan="2"></td></tr>	<div>IMPACT SIMULATION</div> <div></div> <div><p>Para evaluar el impacto del modelo antes del despliegue, se consideran los costos y beneficios de las decisiones correctas e incorrectas. Un modelo preciso mejora la moderación en plataformas digitales, reduciendo la propagación de desinformación y fortaleciendo la confianza de los usuarios. Sin embargo, los falsos positivos pueden generar censura injustificada, afectando la libertad de expresión, mientras que los falsos negativos permiten la difusión de noticias falsas, afectando la percepción pública. Estos errores tienen costos reputacionales y pueden influir en la retención de usuarios y en la credibilidad del sistema de moderación.</p><p>Para simular el impacto antes del despliegue, se utilizan datos históricos etiquetados, junto con métricas de engagement como interacciones y compartidos. Se aplican técnicas como validación cruzada y pruebas A/B para medir la efectividad del modelo en un entorno controlado. Los criterios de despliegue priorizan un equilibrio entre precisión y recall, optimizando la métrica F1. Además, se establecen restricciones de equidad para evitar sesgos en la clasificación, asegurando que el modelo no favorezca ni penalice desproporcionadamente ciertos grupos o fuentes de información (por ejemplo, empleando técnicas de balanceo de datos como SMOTE).</p></div>	<div>MAKING PREDICTIONS</div> <div></div> <div><p>El modelo de detección de noticias falsas puede implementarse tanto en procesamiento por lotes como en tiempo real, dependiendo de los requerimientos operativos de Facebook. En un enfoque por lotes, el sistema analiza periódicamente grandes volúmenes de noticias, permitiendo una verificación programada y eficiente de contenido publicado en la plataforma. Esta modalidad es útil para revisar bases de datos históricas o realizar auditorías de contenido en intervalos predefinidos. Por otro lado, en una implementación en tiempo real, el modelo procesa las noticias al momento de su publicación o interacción, identificando posibles intentos de desinformación de manera inmediata y permitiendo alertas o restricciones dinámicas antes de su difusión masiva. La frecuencia de uso dependerá de la demanda y del nivel de control deseado, pudiendo ajustarse desde verificaciones en tiempo real para contenido viral hasta análisis por lotes en horarios específicos para optimizar el rendimiento y la eficiencia computacional del sistema.</p></div>	<div>BUILDING MODELS</div> <div></div> <div><p>Se entrenaron cuatro modelos de clasificación: Naive Bayes, Random Forest, KNN y Gradient Boosting, evaluando su desempeño con métricas como exactitud, precisión, recall y F1-score. Random Forest obtuvo el mejor rendimiento, equilibrando precisión y recall, mientras que Gradient Boosting mostró buenos resultados con mayor costo computacional. Naive Bayes fue eficiente pero limitado por suposiciones de independencia, y KNN tuvo el peor desempeño.</p><p>Para mantener la efectividad del modelo, es necesario actualizarlo periódicamente con nuevos datos y ajustes, asegurando su precisión en la detección de noticias falsas a lo largo del tiempo.</p></div>	<div>FEATURES</div> <div></div> <div><p>El modelo de detección de noticias falsas utiliza como principales variables el título y la descripción de la noticia, transformados en representaciones numéricas a través de TF-IDF, lo que permite capturar la relevancia de cada término dentro del corpus. Antes de esta transformación, el texto pasa por un proceso de preprocesamiento, que incluye la eliminación de caracteres especiales, conversión a minúsculas, eliminación de stopwords, lematización y tokenización para mejorar la calidad de los datos y reducir el ruido. Además, se pueden considerar variables adicionales, como la fecha de publicación para identificar patrones temporales en la desinformación. Estas características son utilizadas por el modelo Random Forest, el cual aplica múltiples árboles de decisión para detectar patrones en los datos y clasificar la noticia como verdadera o falsa. La combinación de estas transformaciones asegura que el modelo pueda aprender de estructuras lingüísticas y patrones textuales comunes en noticias falsas, mejorando su capacidad predictiva y minimizando falsos negativos.</p></div>	<div>MONITORING</div> <div></div> <div><p>No se debe diligenciar</p></div>			
<div>IMPACT SIMULATION</div> <div></div> <div><p>Para evaluar el impacto del modelo antes del despliegue, se consideran los costos y beneficios de las decisiones correctas e incorrectas. Un modelo preciso mejora la moderación en plataformas digitales, reduciendo la propagación de desinformación y fortaleciendo la confianza de los usuarios. Sin embargo, los falsos positivos pueden generar censura injustificada, afectando la libertad de expresión, mientras que los falsos negativos permiten la difusión de noticias falsas, afectando la percepción pública. Estos errores tienen costos reputacionales y pueden influir en la retención de usuarios y en la credibilidad del sistema de moderación.</p><p>Para simular el impacto antes del despliegue, se utilizan datos históricos etiquetados, junto con métricas de engagement como interacciones y compartidos. Se aplican técnicas como validación cruzada y pruebas A/B para medir la efectividad del modelo en un entorno controlado. Los criterios de despliegue priorizan un equilibrio entre precisión y recall, optimizando la métrica F1. Además, se establecen restricciones de equidad para evitar sesgos en la clasificación, asegurando que el modelo no favorezca ni penalice desproporcionadamente ciertos grupos o fuentes de información (por ejemplo, empleando técnicas de balanceo de datos como SMOTE).</p></div>	<div>MAKING PREDICTIONS</div> <div></div> <div><p>El modelo de detección de noticias falsas puede implementarse tanto en procesamiento por lotes como en tiempo real, dependiendo de los requerimientos operativos de Facebook. En un enfoque por lotes, el sistema analiza periódicamente grandes volúmenes de noticias, permitiendo una verificación programada y eficiente de contenido publicado en la plataforma. Esta modalidad es útil para revisar bases de datos históricas o realizar auditorías de contenido en intervalos predefinidos. Por otro lado, en una implementación en tiempo real, el modelo procesa las noticias al momento de su publicación o interacción, identificando posibles intentos de desinformación de manera inmediata y permitiendo alertas o restricciones dinámicas antes de su difusión masiva. La frecuencia de uso dependerá de la demanda y del nivel de control deseado, pudiendo ajustarse desde verificaciones en tiempo real para contenido viral hasta análisis por lotes en horarios específicos para optimizar el rendimiento y la eficiencia computacional del sistema.</p></div>	<div>BUILDING MODELS</div> <div></div> <div><p>Se entrenaron cuatro modelos de clasificación: Naive Bayes, Random Forest, KNN y Gradient Boosting, evaluando su desempeño con métricas como exactitud, precisión, recall y F1-score. Random Forest obtuvo el mejor rendimiento, equilibrando precisión y recall, mientras que Gradient Boosting mostró buenos resultados con mayor costo computacional. Naive Bayes fue eficiente pero limitado por suposiciones de independencia, y KNN tuvo el peor desempeño.</p><p>Para mantener la efectividad del modelo, es necesario actualizarlo periódicamente con nuevos datos y ajustes, asegurando su precisión en la detección de noticias falsas a lo largo del tiempo.</p></div>	<div>FEATURES</div> <div></div> <div><p>El modelo de detección de noticias falsas utiliza como principales variables el título y la descripción de la noticia, transformados en representaciones numéricas a través de TF-IDF, lo que permite capturar la relevancia de cada término dentro del corpus. Antes de esta transformación, el texto pasa por un proceso de preprocesamiento, que incluye la eliminación de caracteres especiales, conversión a minúsculas, eliminación de stopwords, lematización y tokenización para mejorar la calidad de los datos y reducir el ruido. Además, se pueden considerar variables adicionales, como la fecha de publicación para identificar patrones temporales en la desinformación. Estas características son utilizadas por el modelo Random Forest, el cual aplica múltiples árboles de decisión para detectar patrones en los datos y clasificar la noticia como verdadera o falsa. La combinación de estas transformaciones asegura que el modelo pueda aprender de estructuras lingüísticas y patrones textuales comunes en noticias falsas, mejorando su capacidad predictiva y minimizando falsos negativos.</p></div>					
<div>MONITORING</div> <div></div> <div><p>No se debe diligenciar</p></div>								



Version 12. Created by Louis Dourad, Ph.D. Licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#). Please keep this mention and the link to [OWNML](#) when sharing.

OWNML.CO



Version 1.2. Created by Louis Dorard, Ph.D. Licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/). Please keep this mention and the link to [ownml.co](https://ownml.co) when sharing.

**OWNML.CO**

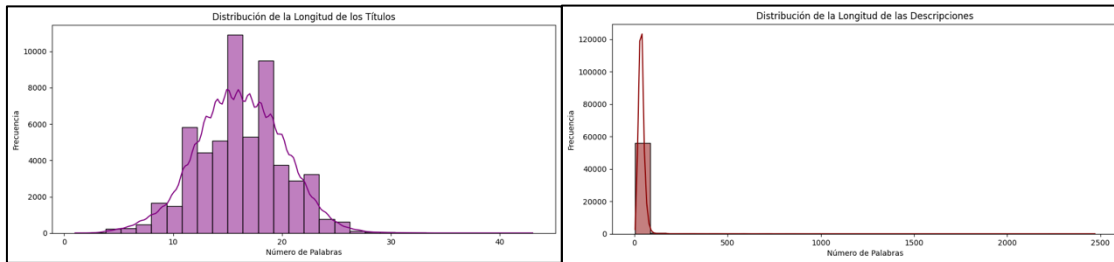
## 2. Entendimiento y preparación de los datos

### 2.1. Entendimiento de los datos

En el proceso de entendimiento de los datos, se realizó una exploración inicial del *dataset* de noticias falsas y verídicas, compuesto por 57,063 registros y 5 columnas: ID, *Label*, Título, Descripción y Fecha. La columna *Label* indica si una noticia es falsa (0) o verídica (1). Se verificaron los tipos de datos, identificando que la variable Fecha estaba en formato de objeto, por lo que se transformó a *datetime* para su correcto análisis temporal.

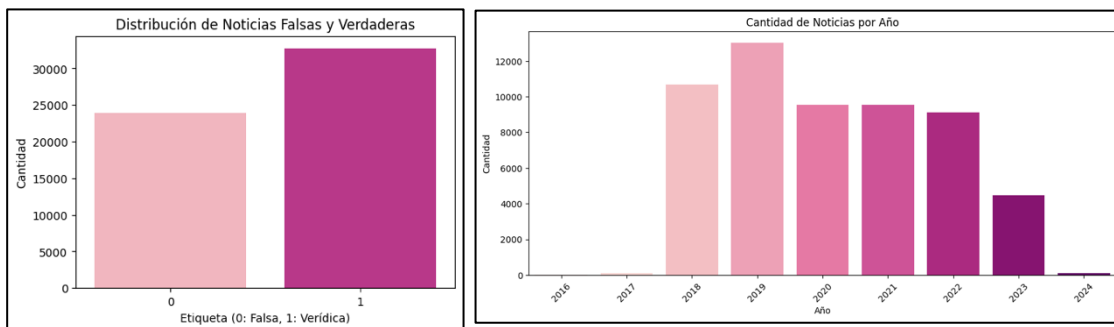
Se realizó un análisis exploratorio de las variables Título y Descripción, examinando la cantidad de valores únicos, la distribución de caracteres y la longitud promedio en los textos. Los títulos varían entre 12 y 255 caracteres, con una media y mediana de 95 caracteres, reflejando una distribución simétrica, donde el 75% no supera los 111 caracteres (ver imagen 1), lo que sugiere una tendencia hacia titulares cortos y directos, un factor clave en la difusión de noticias. En cuanto a las descripciones, se identificaron 49,638 valores únicos, con una longitud que oscila entre 33 y 14,019 caracteres y una media de 243 caracteres (ver imagen 1). Se encontró contenido repetido, siendo la noticia más recurrente mencionada siete veces, lo que sugiere posibles duplicaciones o

reiteración de narrativas, aspecto a considerar en la fase de preprocesamiento para evitar sesgos.



*Ilustración 1 Distribución de la longitud de los títulos (izquierda),  
distribución de la longitud de las descripciones (derecha)*

El análisis de la variable *Fecha* revela que las noticias abarcan el período 2016-2024, sin valores atípicos, lo que facilita el estudio de la evolución de la desinformación en relación con eventos políticos clave. Como se observa en la Imagen 1, la cantidad de noticias alcanza su punto máximo en 2019, seguido de un descenso progresivo. Entre 2020 y 2022, el volumen de noticias se mantiene relativamente estable, pero en 2023 experimenta una caída pronunciada, tendencia que se repite en 2024. Adicionalmente, se evaluó la completitud y unicidad de los datos, encontrando un pequeño porcentaje de valores nulos en la columna Título (0.028%) y la presencia de 445 registros duplicados (0.78%), que podrán ser eliminados en la limpieza. Finalmente, se analizó la distribución de la variable *Label*, confirmando un leve desbalance en las clases (58.1% noticias verdícas y 41.9% falsas) (ver imagen 2), lo que debe considerarse en el entrenamiento del modelo. En general, no se encontraron problemas de validez ni consistencia, y los datos presentan una estructura adecuada para su procesamiento en etapas posteriores.



*Ilustración 2 Distribución de noticias falsas y verdaderas (izquierda),  
distribución cantidad de noticias por año (derecha)*

## Preparación y limpieza de los datos

### 2.1.1. Análisis de calidad de los datos

Como se mencionó anteriormente, el conjunto de datos presenta un alto nivel de completitud, con valores nulos únicamente en la columna *Título* (0.028% de los registros), lo que indica que la información está mayormente disponible para su análisis. En términos de unicidad, se identificaron 445 registros duplicados, representando el 0.78% del total de datos, por lo que su eliminación en el proceso de limpieza no afectará

significativamente la cantidad de información. Por otro lado, la variable *Label*, que clasifica las noticias como falsas (0) o verídicas (1), solo contiene estos dos valores esperados, asegurando su integridad y evitando la necesidad de correcciones adicionales en este aspecto. Además, el análisis de validez muestra que las fechas de publicación de las noticias oscilan entre 2016 y 2024, lo cual es coherente con la naturaleza del problema estudiado y no presenta valores atípicos. En cuanto a la consistencia, los datos textuales no muestran problemas estructurales, y la categorización de las noticias ha sido verificada, garantizando que no existan etiquetas erróneas. En conclusión, el *dataset* se encuentra en buenas condiciones, con una mínima cantidad de valores nulos y duplicados, permitiendo una limpieza sencilla que mejorará la calidad de los datos sin comprometer su representatividad ni su validez para el análisis.

### 2.1.2. Limpieza de datos

El proceso de limpieza de datos comenzó con la identificación y eliminación de registros duplicados, dado que representaban un 0.78% del total del conjunto de datos. Dado que la cantidad de registros afectados era mínima, se optó por eliminarlos para prevenir posibles problemas en el análisis y sesgos en la clasificación de noticias. Luego, se abordó la estandarización y normalización del texto mediante la implementación de diversas funciones de preprocesamiento. Inicialmente, los textos fueron convertidos a minúsculas para evitar discrepancias entre palabras idénticas con distinta capitalización. Posteriormente, se eliminaron caracteres especiales y símbolos extraños, dejando únicamente letras, números y signos de puntuación relevantes. También se aplicó una limpieza adicional para reducir espacios en blanco innecesarios, garantizando una estructura más uniforme en los datos.

Para mejorar la calidad del análisis, se eliminaron los números del texto, ya que no aportaban información significativa en el contexto de clasificación de noticias. Asimismo, se removieron signos de puntuación y caracteres no ASCII, evitando posibles problemas en el procesamiento. Por otro lado, un paso fundamental fue la eliminación de *stopwords* en español, reduciendo términos que no aportaban valor semántico al análisis. Además, se aplicaron técnicas de *stemming*<sup>1</sup> y lematización<sup>2</sup> para reducir las palabras a sus raíces, mejorando la representación semántica y reduciendo la dimensionalidad del texto.

Dado que tanto el título como la descripción de cada noticia contienen información clave para la clasificación, se decidió combinarlos en una nueva variable denominada "Texto", concentrando toda la información en una única columna. Esta transformación permitió un tratamiento unificado de los datos y facilitó su posterior procesamiento en los modelos de clasificación. Para convertir los textos en una representación numérica, se aplicaron dos técnicas de vectorización. Primero, se utilizó *Count Vectorization*<sup>3</sup>, que convierte los

---

<sup>1</sup> Proceso de reducción de palabras a su raíz eliminando afijos, sin garantizar que la forma resultante sea una palabra válida en el diccionario (Murel & Kavlakoglu, s.f.)

<sup>2</sup> Proceso de convertir una palabra a su forma base o "lema", asegurando que la versión resultante sea gramaticalmente correcta y reconocida en el diccionario. (Murel & Kavlakoglu, s.f.)

<sup>3</sup> Técnica de procesamiento de texto que transforma documentos en una matriz numérica basada en la frecuencia de aparición de cada palabra, sin considerar su contexto ni orden. (Navarro, 2024)

textos en vectores de frecuencia de palabras mediante una transformación binaria 1 a 1. Sin embargo, este método no distingue entre términos relevantes y palabras comunes, lo que puede afectar el desempeño del modelo. Posteriormente, se aplicó TF-IDF (*Term Frequency-Inverse Document Frequency*<sup>4</sup>), que ajusta la importancia de cada palabra en función de su frecuencia en el documento y en el *corpus* completo, penalizando los términos más repetidos y resaltando aquellos más distintivos. Tras la vectorización, el conjunto de datos se dividió en 60% para entrenamiento y 40% para validación, asegurando una distribución adecuada para evaluar el modelo. Además, se implementó SMOTE (*Synthetic Minority Over-sampling Technique*<sup>5</sup>) con el objetivo de balancear las clases y mejorar la capacidad del modelo para detectar tanto noticias falsas como verídicas, evitando sesgos hacia la clase mayoritaria.

Finalmente, se optó por trabajar exclusivamente con TF-IDF, ya que su capacidad para reducir la influencia de términos comunes permitió mejorar la precisión y el rendimiento del modelo. Al penalizar las palabras más repetidas, se logró una representación más equilibrada del contenido de las noticias, optimizando la clasificación de textos en el modelo de aprendizaje automático.

### 3. Modelado y evaluación

#### 3.1. Modelo *Naive Bayes* (Natalia Villegas)

Para la clasificación de noticias falsas, se empleó el modelo *Naive Bayes Multinomial*, un algoritmo basado en el Teorema de Bayes, especialmente diseñado para el análisis de texto (*Scikit-learn*, s.f.). Este modelo es ideal para datos discretos y se utiliza comúnmente en tareas como detección de *spam* y clasificación de documentos. Su funcionamiento se basa en modelar la frecuencia de palabras en cada categoría y calcular la probabilidad de que un texto pertenezca a una clase determinada en función de la aparición de sus términos (Jurafsky & Martin, 2021).

El *Naive Bayes Multinomial* asume que las palabras dentro de un documento son independientes entre sí, lo que simplifica su aplicación (Geeks for Geeks, 2025). En este caso, el modelo trabajó con la representación TF-IDF de los textos, la cual ajusta la importancia de cada palabra según su frecuencia en el documento y su relevancia. Aunque este modelo fue diseñado para trabajar con recuentos enteros de palabras, en la práctica también puede manejar valores fraccionarios como los generados por TF-IDF (*Scikit-learn*, s.f.), lo que lo hace altamente adaptable.

En el primer modelo, al aplicar SMOTE para equilibrar el conjunto de entrenamiento, se obtuvo una exactitud del 87.91%, con un F1-score de 0.88. Sin embargo, al evaluar en el conjunto de validación, la exactitud disminuyó a 82.28%, lo que sugiere que el modelo perdió capacidad de generalización. Como se observa en la matriz de confusión, SMOTE ayudó a mejorar la clasificación de noticias falsas, reduciendo los falsos negativos, pero

---

<sup>4</sup> Técnica que asigna peso a las palabras en un documento, aumentando la importancia de las frecuentes en ese texto y reduciendo la de las comunes en todo el corpus. (Jain, 2024)

<sup>5</sup> Técnica de “sobremuestreo” que genera datos sintéticos de la clase minoritaria para equilibrar el *dataset* y mejorar el rendimiento del modelo sin causar sobreajuste. (Maklin, 2022)

a costa de un aumento significativo en los falsos positivos. Esto indica que el modelo “sobreaprendió” los patrones de la clase minoritaria artificialmente generada, lo que llevó a una mayor propensión a clasificar noticias verdícas como falsas.

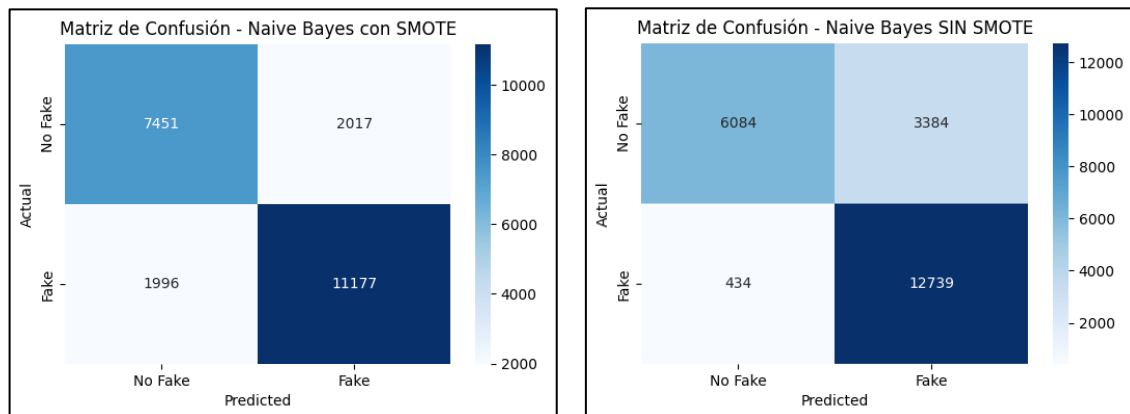


Ilustración 3 Matriz de confusión para los modelos de Naive Bayes

En el segundo modelo, se eliminó SMOTE para permitir que el modelo aprenda directamente de la distribución original de los datos. Esto resultó en una mejora en la exactitud de validación, alcanzando un 83.14%, con un F1-score de 0.87, mostrando un mejor balance entre precisión y *recall*. No obstante, la matriz de confusión evidencia un incremento en los falsos negativos, lo que significa que algunas noticias falsas no fueron detectadas correctamente. A pesar de esta desventaja, el modelo sin SMOTE presenta un rendimiento general más estable y evita el sesgo introducido por el sobremuestreo sintético, lo que lo hace más confiable para la clasificación de noticias en un entorno real.

### 3.2. Modelo *Random Forest* (Carol Florido)

El modelo de clasificación *Random Forest* fue optimizado mediante la técnica de *RandomizedSearchCV* con el objetivo de encontrar la mejor combinación de hiperparámetros que maximizara su rendimiento. Se exploraron distintas configuraciones en términos del número de árboles en el bosque, la profundidad máxima de los árboles, la cantidad mínima de muestras necesarias para dividir un nodo, el número mínimo de muestras por hoja y el uso de muestreo con reemplazo. Tras realizar 15 iteraciones con validación cruzada de tres pliegues, se identificó que la mejor configuración incluía 400 árboles, sin restricción en la profundidad, con un mínimo de dos muestras para la división de un nodo, dos muestras por hoja y sin el uso de muestreo con reemplazo.

El desempeño del modelo fue evaluado tanto en el conjunto de entrenamiento como en el conjunto de validación, reportando métricas clave para medir su capacidad predictiva. En la fase de entrenamiento, el modelo obtuvo un alto nivel de precisión y *recall*, lo que indica que logró ajustar correctamente los datos y detectar la mayoría de los casos positivos. Sin embargo, el desempeño en el conjunto de validación es un mejor indicador de su capacidad de generalización. En esta etapa, se obtuvo una precisión del 89.66 %, una precisión positiva del 87.76 %, un *recall* del 95.54 % y un puntaje F1 de 91.49 %. Adicionalmente, el puntaje F2, que otorga un mayor peso a la capacidad del modelo para identificar los casos positivos, alcanzó un valor de 93.88 %, mientras que el área bajo la



curva ROC fue de 88.50 %. Sin embargo, también es fundamental analizar los resultados por clase para comprender mejor el comportamiento del modelo. Para la clase negativa (etiqueta 0), que representa los casos no fraudulentos o la ausencia de la característica de interés, se obtuvo una precisión del 93 %, lo que significa que, cuando el modelo predice que un caso es negativo, en el 93 % de las ocasiones está en lo correcto. Sin embargo, el *recall* para esta clase es del 81 %, lo que implica que el modelo no identifica el 19 % de los casos negativos correctamente, clasificándolos erróneamente como positivos. Esto se traduce en una mayor cantidad de falsos positivos, es decir, casos que fueron predichos incorrectamente como positivos cuando en realidad no lo eran. Por otro lado, la clase positiva (etiqueta 1), que representa los casos fraudulentos o la presencia de la característica de interés, tiene una precisión del 88 %, lo que indica que, de todas las instancias que el modelo predijo como positivas, el 88 % son realmente positivas. Además, su *recall* es del 96 %, lo que significa que el modelo es capaz de detectar la mayoría de los casos positivos, con solo un 4 % de falsos negativos. Este alto *recall* es especialmente relevante en problemas donde la correcta identificación de los casos positivos es prioritaria, ya que minimiza el riesgo de no detectar instancias importantes.

Estos resultados indican que el modelo presenta un buen equilibrio entre precisión y *recall*, favoreciendo la detección de la clase positiva sin comprometer excesivamente la precisión general. La matriz de confusión muestra que la mayoría de los casos han sido clasificados correctamente, con una baja tasa de falsos negativos, lo que es relevante en problemas donde la correcta identificación de los casos positivos es prioritaria. En general, el modelo *Random Forest* seleccionado demuestra ser una opción robusta para la clasificación dentro del contexto del problema planteado.

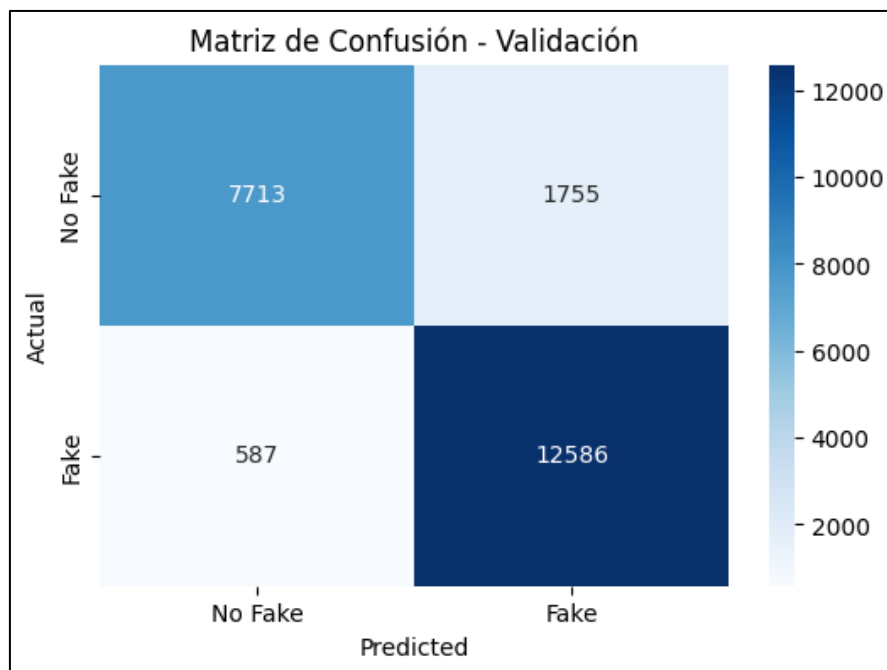


Ilustración 4 Matriz de confusión modelo de Random Forest

### 3.3. Modelo KNN (Martin Vásquez)

El modelo *K-Nearest Neighbors* (KNN) fue empleado en la tarea de clasificación debido a su enfoque basado en la similitud entre los datos. KNN es un algoritmo de aprendizaje supervisado que asigna una nueva observación a la clase mayoritaria entre sus vecinos más cercanos. Es un modelo no paramétrico, lo que significa que no hace suposiciones sobre la distribución de los datos y, en su lugar, toma decisiones basadas en la cercanía entre puntos en el espacio de características (Scikit-learn, s.f.). Este método es ampliamente utilizado en clasificación y regresión, ya que su simplicidad y eficacia permiten obtener buenos resultados sin necesidad de una etapa de entrenamiento prolongada.

El proceso inició con la representación de los textos mediante la matriz TF-IDF, que transforma los documentos en vectores de características ponderadas. el conjunto de datos se dividió en 80% para entrenamiento y 20% para validación, asegurando la reproducibilidad mediante la fijación de un *random\_state*. Dado que los datos estaban desbalanceados, se aplicó SMOTE, una técnica que genera ejemplos sintéticos de la clase minoritaria para evitar que el modelo tenga un sesgo hacia la clase predominante.

Inicialmente, se implementó KNN con tres vecinos entrenado sobre los datos con SMOTE. Sin embargo, los resultados mostraron que la estandarización de los datos afectaba negativamente el rendimiento del modelo. La normalización con *MaxAbsScaler*, una técnica que ajusta los valores a una escala entre -1 y 1 sin alterar la estructura dispersa de la matriz, no logró mejorar el desempeño. De hecho, al aplicar esta transformación, los resultados en validación se deterioraron significativamente, lo que sugiere que la distancia euclidiana, utilizada por defecto en KNN, no se benefició de la estandarización en este caso. A continuación

Los resultados al entrenar al modelo únicamente con SMOTE se obtuvo una exactitud del 77.08%, con un F1-score de 0.7760. Sin embargo, al evaluar en el conjunto de validación, la exactitud disminuyó a 55%, lo que sugiere que el modelo perdió capacidad de generalización. Esto indica que el modelo “sobreaprendió” los patrones de la clase minoritaria artificialmente generada, lo que llevó a una mayor posibilidad de clasificar noticias verídicas como falsas.

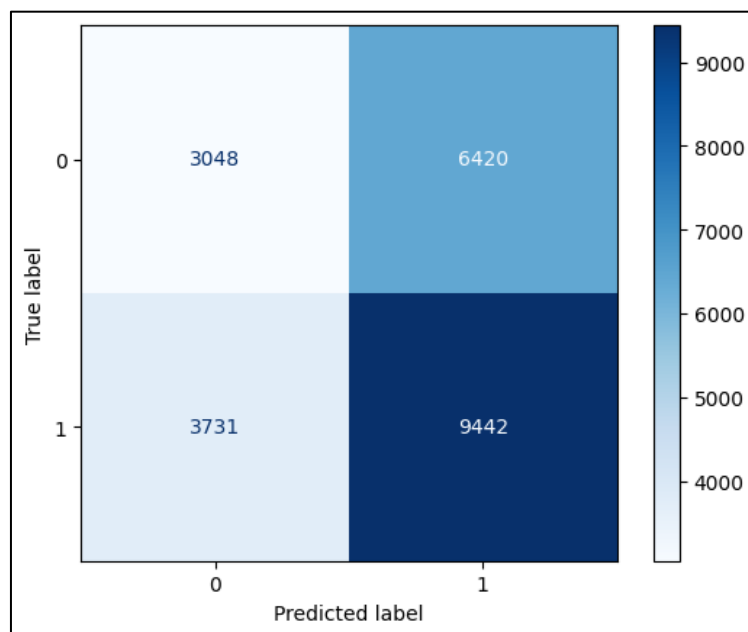


Ilustración 5 Matriz de confusión KNN – Hiperparámetros base

Se realizó una búsqueda de hiperparámetros con *GridSearchCV*, evaluando distintos valores de vecinos y métricas de distancia. El mejor modelo utilizó 9 vecinos y distancia euclidiana, logrando una precisión del 78.61% en entrenamiento y 64% en validación. Sin embargo, la tasa de falsos verdaderos seguía siendo alta, lo que indica dificultades en la clasificación de noticias falsas, igualmente, pero en menor proporción tuvo errores al clasificar noticias verdaderas como falsas. El análisis mostró que la elección del número de vecinos y la métrica de distancia afecta significativamente el desempeño del modelo. Aunque SMOTE ayudó a equilibrar los datos, introdujo sesgos que impactaron la generalización. Se sugiere explorar técnicas adicionales de balanceo o combinar KNN con otros enfoques más robustos, como o redes neuronales, para mejorar la clasificación en conjuntos de datos desbalanceados.

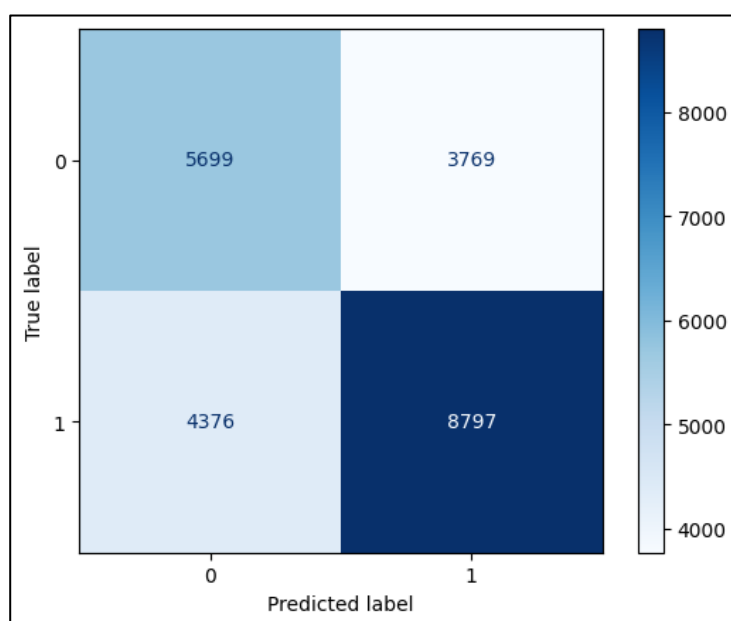


Ilustración 6 Matriz de confusión KNN hiperparámetros mejorados

### 3.4. Modelo Clasificador *Gradient Boosting* (Natalia Villegas)

Para la clasificación de noticias falsas, se implementó el algoritmo *Gradient Boosting*, un método de ensamble basado en árboles de decisión, que optimiza una función de pérdida diferenciable en cada etapa. Este modelo se entrena secuencialmente, donde cada nuevo árbol aprende corrigiendo los errores cometidos por los modelos previos, lo que permite capturar relaciones complejas en los datos y mejorar la precisión de la clasificación.

Inicialmente, se utilizó un *Gradient Boosting Classifier* con 100 estimadores y una profundidad de 3, entrenado sobre los datos balanceados con SMOTE. El desempeño del modelo en el conjunto de entrenamiento mostró una exactitud del 86.44%, con una precisión del 79.74% y un *recall* del 97.71%, lo que indica una alta capacidad para identificar correctamente las noticias falsas. En la validación, el modelo alcanzó un 87% de exactitud, con una precisión de 89.18% y un *recall* de 84.73%, manteniendo un buen equilibrio entre precisión y sensibilidad.

En busca de generar una mejora, se realizó un ajuste de hiperparámetros mediante GridSearchCV, optimizando la cantidad de estimadores y la profundidad de los árboles. La mejor configuración encontrada fue 500 estimadores y una profundidad de 5, con el criterio de pérdida *friedman\_mse*, lo que permitió un aumento significativo en el rendimiento. Con estos ajustes, el modelo alcanzó en entrenamiento una exactitud del 94.17%, con una precisión de 90.49% y un *recall* del 98.73%. En la validación, la exactitud mejoró al 91%, con una precisión del 92.24% y un *recall* del 90.08%, evidenciando una mayor generalización del modelo sin caer en sobreajuste.

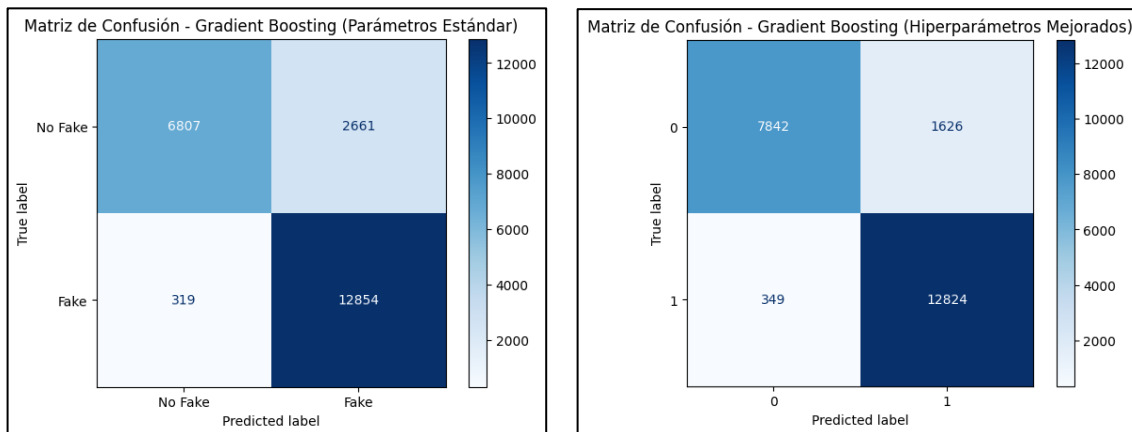


Ilustración 7 Matrices de confusión Gradient Boosting con hiperparámetros estándar y mejorados

En la matriz de confusión, se evidencia una reducción significativa en la cantidad de falsos positivos y falsos negativos, lo que indica que el modelo logra una mejor discriminación entre noticias falsas y verídicas. No obstante, *Gradient Boosting* presenta como principal desventaja su alto tiempo de entrenamiento, que aumenta considerablemente a medida que se incrementa el número de estimadores y la profundidad de los árboles. Por ejemplo, al realizar una búsqueda en grilla con solo dos valores de estimadores y dos de profundidad máxima, junto con tres particiones de validación cruzada (CV=3), el proceso tomó aproximadamente 20 minutos, incluso utilizando paralelización con *n\_jobs*. A pesar de esta complejidad computacional, los

resultados obtenidos posicionan a este modelo como una de las mejores opciones para la clasificación de noticias falsas, gracias a su capacidad de generalización y su precisión en la identificación de patrones.

### 3.5. Evaluación de métricas generalizadas

Al evaluar los diferentes modelos utilizados en la clasificación de noticias falsas, se observa que cada uno presenta ventajas y limitaciones en términos de precisión, *recall* y capacidad de generalización. Naive Bayes Multinomial, si bien es un modelo ampliamente utilizado en procesamiento de texto, mostró un comportamiento variable dependiendo de la estrategia de balanceo de datos aplicada. Con SMOTE, logró mejorar la detección de la clase minoritaria, pero esto tuvo un costo en términos de falsos positivos. Sin SMOTE, su desempeño fue más estable y menos sesgado, aunque con una ligera disminución en el *recall* de la clase positiva. Por otro lado, K-Nearest Neighbors (KNN) enfrentó dificultades significativas en la clasificación, especialmente en la generalización a nuevos datos. A pesar de los esfuerzos por optimizar sus hiperparámetros y aplicar balanceo de clases, su precisión en validación fue la más baja entre los modelos analizados. La sensibilidad de KNN a la distancia entre puntos hizo que la transformación de datos afectara su rendimiento de manera negativa, lo que lo convierte en una opción menos confiable en este contexto. *Gradient Boosting*, en comparación, mostró un mejor equilibrio entre precisión y *recall*, con un rendimiento destacado en la identificación de noticias falsas. Su capacidad para aprender de errores previos y ajustar sus predicciones permitió que el modelo alcanzara una buena exactitud en validación. Sin embargo, su desempeño no superó al modelo de Random Forest, que logró una mayor estabilidad y generalización gracias a su enfoque de ensamblado. Dentro de este análisis, Random Forest demostró ser la opción más robusta, con el puntaje F1 más alto entre los modelos evaluados. Su capacidad para manejar datos desbalanceados sin necesidad de técnicas de sobremuestreo le permitió mantener una alta precisión sin comprometer el *recall*. Además, su flexibilidad para capturar relaciones complejas en los datos sin caer en sobreajuste lo convierte en la mejor alternativa para la clasificación de noticias falsas en este estudio.

## 4. Resultados

### 4.1.Descripción de los resultados: Métricas y objetivos del negocio

Tabla 2 Comparación de las métricas de los modelos

	Naive Bayes con Smote	Naive Bayes Sin Smote	Random Forest	Random Forest con Ajuste de Hiperparámetros	KNN	KNN con Ajuste de Hiperparámetros	Clasificador Gradient Boosting	Clasificador Gradient Boosting con Ajuste de Hiperparámetros
Exactitud	0,823	0,831	0,882	0,896	0,55	0,67	0,87	0,91
Precisión	0,846	0,790	0,872	0,877	0,520	0,661	0,890	0,920
Recall	0,850	0,967	0,934	0,955	0,518	0,649	0,848	0,899
Puntuación F1	0,848	0,870	0,902	0,914	0,512	0,650	0,859	0,907
Puntuación F2	0,849	0,926	0,921	0,938	0,7856	0,8273	0,9329	0,9686
ROC AUC Score	0,817	0,805	0,871	0,885	0,7718	0,7723	0,8636	0,941

Para evaluar el desempeño de los modelos entrenados en la detección de noticias falsas en el ámbito político, se analizaron varias métricas clave: exactitud, precisión, recall, F1-score, F2-score y ROC AUC. Dado que el objetivo del negocio es minimizar la propagación de desinformación sin generar demasiados falsos positivos en noticias verídicas, se priorizó la métrica F1-score, ya que equilibra la precisión y el recall, evitando sesgos en la clasificación.

Entre los modelos evaluados, Random Forest con ajuste de hiperparámetros mostró el mejor rendimiento global, alcanzando un F1-score de 0.914, superando a los demás modelos en términos de equilibrio entre precisión (0.877) y recall (0.955). Aunque Gradient Boosting con ajuste de hiperparámetros obtuvo un rendimiento competitivo con un F1-score de 0.907, este modelo requiere un mayor poder computacional y tiempos de entrenamiento más prolongados, lo que lo hace menos viable para una implementación inmediata en entornos productivos con grandes volúmenes de datos. Por otro lado, Naive Bayes sin SMOTE mostró un mejor desempeño que su versión con SMOTE, ya que aunque la versión balanceada mejoraba el recall, generaba un aumento significativo en los falsos positivos, afectando la precisión del modelo.

Modelos como KNN, incluso con ajuste de hiperparámetros, presentaron un desempeño considerablemente inferior, con un F1-score de 0.65 y una exactitud del 67%, lo que indica que no logró generalizar adecuadamente en la tarea de clasificación de noticias falsas y verídicas. Este resultado confirma que KNN no es la mejor opción para este tipo de problemas, ya que su desempeño es altamente sensible a la cantidad de vecinos seleccionados y al tamaño del conjunto de datos.

#### **4.2. Objetivo del negocio y selección del mejor modelo**

El objetivo del negocio es proporcionar a plataformas digitales y redes sociales un modelo confiable para la detección de noticias falsas, asegurando un equilibrio entre precisión y recall para minimizar tanto la propagación de desinformación como la eliminación errónea de noticias verídicas. En este contexto, Random Forest con ajuste de hiperparámetros es la mejor opción, ya que ofrece el mayor F1-score y un buen equilibrio entre métricas sin requerir un alto costo computacional. Si bien Gradient Boosting es un modelo competitivo en términos de precisión, su alto costo en tiempo de entrenamiento lo hace menos viable para una implementación inmediata en entornos de producción. No obstante, podría considerarse como una alternativa para escenarios donde se disponga de recursos computacionales más avanzados y se busque optimizar aún más la detección de fake news.

En conclusión, Random Forest con ajuste de hiperparámetros es la mejor opción para este caso de uso, ya que proporciona una solución efectiva, escalable y de alto rendimiento para mejorar las estrategias de moderación de contenido en plataformas digitales.

Para comprender las diferencias en el lenguaje utilizado en los títulos de noticias falsas y verídicas en el ámbito de la política española, se generaron nubes de palabras con los términos más frecuentes en cada categoría. En la primera imagen, que representa los títulos de noticias falsas, se destacan términos como "gobierno", "catalunya", "iniciativa", "coalición", "congreso", "canarias" y "presupuesto". La presencia de estas palabras sugiere un enfoque en temas de autonomía regional, pactos políticos y decisiones gubernamentales, posiblemente presentados de manera exagerada o tergiversada para generar desinformación. También se observa un alto uso de nombres propios y partidos políticos, lo que indica una tendencia a personalizar los titulares con el fin de captar la atención de los lectores.

[illegible]

*Ilustración 8 Nube de palabras para los títulos de las noticias (a la izquierda son las noticias falsas y a la derecha las verídicas)*

15





Ilustración 9 Nube de palabras para las descripciones de las noticias (a la izquierda son las noticias falsas y a la derecha las verdaderas)

Las noticias falsas parecen enfocarse en narrativas más especulativas y en el uso de términos que pueden generar polémica o llamar la atención del lector. Utilizan con frecuencia palabras que sugieren acuerdos políticos o alianzas estratégicas, lo que puede contribuir a la desinformación. En cambio, las noticias verdaderas presentan un lenguaje más institucional y técnico, con énfasis en partidos, leyes y actores políticos clave, lo que sugiere una intención más informativa y objetiva. Estas diferencias reflejan cómo el contenido y la forma de estructurar los textos pueden influir en la percepción de credibilidad de una noticia.

Una estrategia clave para abordar la desinformación identificada en los títulos y descripciones de las noticias es desarrollar campañas de alfabetización mediática dirigidas a los lectores. Estas campañas deben centrarse en enseñar a los usuarios a identificar señales de noticias falsas, como el uso de lenguaje especulativo, la personalización excesiva de los titulares y la falta de fuentes verificables. Además, la implementación de herramientas de verificación automática dentro de las plataformas digitales permitiría alertar a los usuarios sobre posibles sesgos o manipulaciones en los contenidos que consumen, fomentando un análisis crítico y una lectura más informada.

## 5. Trabajo en equipo

Para la implementación del proyecto, el equipo se organizó en tres roles principales: líder de proyecto, líder de analítica y líder de datos. La distribución de tareas y responsabilidades se realizó equitativamente, asegurando que cada integrante contribuyera de manera efectiva al desarrollo del modelo de clasificación de noticias falsas. Se llevaron a cabo reuniones estratégicas para definir los objetivos, asignar actividades y dar seguimiento al avance del trabajo.

### 5.1. Roles y Tareas Realizadas por Cada Integrante

**Natalia Villegas Calderón (Líder de Proyecto)**



- **Horas dedicadas:** 20
- **Contribución:** 33.3/10
- **Algoritmo trabajado:** Naive Bayes y *Gradient Boosting*
- **Tareas:**
  - Gestionar el desarrollo del proyecto y asegurar la equidad en la distribución de tareas.
  - Definir fechas de reuniones y coordinar la entrega final.
  - Verificar que los entregables cumplieran con los requisitos y estándares del curso.
  - Documentar el entendimiento del problema y la estructura del análisis.
  - Apoyar en la integración final del documento y la presentación del proyecto.
  - Subir la entrega final del grupo.
  - Implementa los dos algoritmos escogidos *Naive Bayes* y *Gradient Boosting*.
  - Elabora 3 casillas del OWNML ML Canva.
  - Garantizar la consistencia del *Notebook*.
  -

#### **Carol Sofía Florido Castro (Líder de Analítica)**

- **Horas dedicadas:** 20
- **Contribución:** 33.3/10
- **Algoritmo trabajado:** *Random Forest*
- **Tareas:**
  - Implementar y evaluar el modelo de *Random Forest*, asegurando la optimización de sus parámetros.
  - Comparar el desempeño del modelo con otras opciones para determinar su viabilidad.
  - Generar análisis de métricas y visualizaciones de resultados.
  - Documentar los hallazgos y mejoras en el rendimiento del modelo.
  - Diseño de las diapositivas y guion del video.
  - Almacenar, configurar y mantener el repositorio en las mejores condiciones.

#### **Juan Martin Vásquez Cristancho (Líder de Datos)**

- **Horas dedicadas:** 20
- **Contribución:** 33.3/10
- **Algoritmo trabajado:** Algoritmo trabajado: K-Vecinos Más Cercanos (KNN)
- **Tareas:**
  - Preparar los datos para su uso en los modelos, asegurando su limpieza y estructuración.
  - Implementar el modelo de : K-Vecinos Más Cercanos y analizar su desempeño.
  - Evaluar el impacto del uso de técnicas como SMOTE para balancear los datos.
  - Mantener el repositorio de trabajo y garantizar la disponibilidad de los datos para el equipo.
  - Grabar, editar y subir el video a YouTube para su visualización.
  - Colocar la información en las diapositivas y ayudar en el guion.

- Documentar las reuniones realizadas.
- Documentar el trabajo del grupo.

## 5.2. Reuniones de grupo

### Reunión de lanzamiento y planificación:

- **Fecha:** 7 de febrero de 2025
- **Integrantes:** Todos
- **Resumen:** Durante esta reunión se establecieron las reglas de trabajo en equipo, los roles y las tareas asignadas a cada integrante. Se definieron las expectativas para el desarrollo del proyecto y se inició la construcción del Canvas de *Machine Learning*. También se establecieron los medios de comunicación y fechas tentativas de seguimiento.

### Reunión de ideación y control de avances

- **Fecha:** 14 de febrero de 2025
- **Integrantes:** Todos
- **Resumen:** Se revisó el progreso en el Canvas de *Machine Learning* y se discutieron posibles enfoques para los modelos de clasificación. En este punto, aún no se había abordado el tema en clase, por lo que se plantearon ideas preliminares sobre técnicas de análisis de datos y modelos supervisados. También se profundizó en la comprensión del problema para garantizar que los modelos propuestos estuvieran alineados con los objetivos del proyecto.

### Reunión de división de modelos y avances

- **Fecha:** 18 de febrero de 2025
- **Integrantes:** Todos
- **Resumen:** Tras revisar en clase los diferentes modelos de clasificación, se asignaron responsabilidades específicas a cada integrante para la implementación de los algoritmos. Una de las integrantes logró avanzar rápidamente en la implementación de un modelo y compartió sus hallazgos con el equipo, facilitando la integración.

### Reunión de seguimiento y consolidación final

- **Fecha:** 21 de febrero de 2025
- **Integrantes:** Todos
- **Resumen:** Reunión presencial en la que se discutieron los avances de cada integrante y se evaluó el estado general del proyecto. Se identificaron las tareas pendientes y se establecieron prioridades para la entrega final del día siguiente. Se verificó la calidad de los modelos implementados y la documentación del proyecto para garantizar su completitud y coherencia.

### Reunión de entrega y ajustes finales

- **Fecha:** 22 de febrero de 2025

- **Integrantes:** Todos
- **Resumen:** Durante el día de la entrega, los integrantes del equipo estuvieron en constante comunicación y se reunieron en varias ocasiones para revisar y mejorar los diferentes entregables. Se realizaron ajustes finales en la documentación, métricas de los modelos y presentación, asegurando que toda la información estuviera clara y alineada con los objetivos del proyecto.

### 5.3. Uso de *ChatGPT* en el proyecto y otras NLP

*ChatGPT* fue una herramienta clave en el desarrollo del proyecto, principalmente para la corrección y optimización del código. Se utilizó para solucionar problemas en la ejecución de los modelos, especialmente en la optimización del recorrido de los datos, ya que inicialmente el procesamiento era demasiado complejo y hacía que los modelos tardaran en responder. Además, fue útil en la depuración de errores, ayudando a interpretar mensajes de fallo y detectar problemas con versiones de *Git*, lo que facilitó la gestión del código en el repositorio. También se empleó para generar y mejorar gráficos, agregando títulos, optimizando la presentación y asegurando una visualización más clara de los resultados.

Otro uso relevante fue en la comprensión de conceptos técnicos, ya que se le consultaron explicaciones sobre los diferentes algoritmos empleados en el proyecto y su funcionamiento detallado. Esto permitió una mejor comprensión de sus ventajas y limitaciones, lo que facilitó la toma de decisiones sobre qué modelos aplicar.

### 5.4. Retos y soluciones

Uno de los mayores retos del proyecto fue la gestión del tiempo, sobre todo porque coincidió con otras entregas académicas, lo que nos dejó poco margen para avanzar con calma. Esto hizo que muchas decisiones sobre los modelos y el procesamiento de datos tuvieran que tomarse en la última semana, aumentando la carga de trabajo y la presión. Para intentar manejarlo mejor, organizamos reuniones semanales para hacer seguimiento y repartir las tareas de manera equitativa. Aun así, al acercarse la fecha de entrega, tuvimos que trabajar a un ritmo mucho más acelerado para completar todo a tiempo, lo que generó bastante estrés en el equipo.

Otro problema fue el manejo del repositorio en *GitHub*, especialmente al trabajar con *Jupyter Notebook*, ya que generaba conflictos de versiones y errores de concurrencia que eran complicados de resolver. También hubo dificultades con el tamaño de los archivos, en especial los CSV del procesamiento de datos y el .pkl de los modelos, lo que complicó su almacenamiento y actualización en el repositorio. Además, interpretar los resultados con SMOTE fue un reto, porque, aunque ayudaba a balancear las clases, también aumentaba la cantidad de falsos positivos. Después de analizar las métricas, decidimos que el modelo sin SMOTE era una mejor opción, ya que lograba un mejor equilibrio entre precisión y *recall*.

## Referencias

- Jain, A. (2024, 4 de febrero). TF-IDF in NLP (Term Frequency Inverse Document Frequency). Medium. <https://medium.com/@abhishekjainindore24/tf-idf-in-nlp-term-frequency-inverse-document-frequency-e05b65932f1d>
- Jurafsky, D., & Martin, J. H. (2021). *Speech and Language Processing (3rd ed.)*. Stanford University. Recuperado de <https://web.stanford.edu/~jurafsky/slp3/>
- Maklin, C. (2022, 14 de mayo). Synthetic Minority Over-sampling Technique (SMOTE). Medium. <https://medium.com/@corymaklin/synthetic-minority-over-sampling-technique-smote-7d419696b88c>
- Murel, J. & Kavlakoglu, E. (s.f.). *¿Qué son el stemming y la lematización?*. IBM. Recuperado de <https://www.ibm.com/es-es/think/topics/stemming-lemmatization>
- Navarro, S. (2024, 8 de noviembre). ¿Qué es el count vectorizer? KeepCoding. Recuperado de <https://keepcoding.io/blog/que-es-el-count-vectorizer/>
- Scikit-learn. (s.f.). *KNeighborsClassifier*. Scikit-learn *Documentation*. Recuperado de <https://scikit-learn.org/stable/modules/neighbors.html>
- Scikit-learn. (s.f.). *Naive Bayes - MultinomialNB*. Scikit-learn *Documentation*. Recuperado de [https://scikit-learn.org/stable/modules/naive\\_bayes.html#multinomial-naive-bayes](https://scikit-learn.org/stable/modules/naive_bayes.html#multinomial-naive-bayes)
- Scikit-learn. (s.f.). *KNeighborsClassifier*. Scikit-learn *Documentation*. Recuperado de <https://scikit-learn.org/stable/modules/neighbors.html>