UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO
FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA, INFORMÁTICA Y MECÁNICA
ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA Y DE SISTEMAS

TESIS

IMPLEMENTACIÓN DE UN MODELO DETECTOR DE NOTICIAS FALSAS PARA REDUCIR LA DESINFORMACIÓN EN LA POBLACIÓN DEL PERÚ

Presentado por:
Br. Diego Yoshiro Dongo Esquivel

Para optar al título profesional de:
Ingeniero Informático y de Sistemas

Asesor:
Dr. Rony Villafuerte Serna

Cusco - Perú
2023

INFORME DE ORIGINALIDAD

(Aprobado por Resolución Nro.CU-303-2020-UNSAAC)

El que suscribe, **Asesor** del trabajo de investigación/tesis titulada: Implementación de un modelo detector de noticias falsas para reducir la desinformación en la población del Perú

presentado por: Diego Yoshiro Dongo Esquivel con DNI Nro.: 71589938

presentado por: ..... con DNI Nro.: .....

para optar el título profesional/grado académico de Ingeniero Informático y de Sistemas

Informo que el trabajo de investigación ha sido sometido a revisión por 3 veces, mediante el Software Antiplagio, conforme al Art. 6° del ***Reglamento para Uso de Sistema Antiplagio de la UNSAAC*** y de la evaluación de originalidad se tiene un porcentaje de 02 %.

Evaluación y acciones del reporte de coincidencia para trabajos de investigación conducentes a grado académico o título profesional, tesis

Porcentaje	Evaluación y Acciones	Marque con una (X)
Del 1 al 10%	No se considera plagio.	X
Del 11 al 30 %	Devolver al usuario para las correcciones.	
Mayor a 31%	El responsable de la revisión del documento emite un informe al inmediato jerárquico, quien a su vez eleva el informe a la autoridad académica para que tome las acciones correspondientes. Sin perjuicio de las sanciones administrativas que correspondan de acuerdo a Ley.	

Por tanto, en mi condición de asesor, firmo el presente informe en señal de conformidad y **adjunto** la primera página del reporte del Sistema Antiplagio.

Cusco, 07 de setiembre de 2023

Firma

Post firma Rony Ulloa Salas

Nro. de DNI 23937778

ORCID del Asesor 0000-0003-4607-522X

Se adjunta:

1. Reporte generado por el Sistema Antiplagio.
2. Enlace del Reporte Generado por el Sistema Antiplagio: 27239:261044086

turnitin

Identificación de reporte de similitud: oid:27259:261044086

NOMBRE DEL TRABAJO

noticiasFalsasV2

AUTOR

Diego Dongo

RECUENTO DE PALABRAS

24431 Words

RECUENTO DE CARACTERES

132414 Characters

RECUENTO DE PÁGINAS

81 Pages

TAMAÑO DEL ARCHIVO

2.4MB

FECHA DE ENTREGA

Sep 5, 2023 11:20 PM GMT-5

FECHA DEL INFORME

Sep 5, 2023 11:21 PM GMT-5

2% de similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base c

- 2% Base de datos de Internet
- 0% Base de datos de publicaciones
- Base de datos de Crossref
- Base de datos de contenido publicado de Crossr
- 2% Base de datos de trabajos entregados

Excluir del Reporte de Similitud

- Material bibliográfico
- Material citado
- Coincidencia baja (menos de 20 palabras)
- Fuentes excluidas manualmente
- Bloques de texto excluidos manualmente

Resumen

Dedicatoria

A mis padres y hermanas, por su inmenso amor.

Agradecimientos

Quiero expresar mi profundo agradecimiento a aquellas personas que han hecho posible la realización de este proyecto:

En primer lugar, a Adri, por su apoyo emocional y desinteresado en todo momento. Gracias por estar ahí y por creer en mí.

A Brandon, por su sincera amistad. Gracias por ser un compañero de confianza y por animarme a seguir adelante.

A mis amigos de toda la vida, Sebastián y Marcelo, por su apoyo incondicional y por sacarme siempre una sonrisa. Gracias por estar a mi lado.

A mi asesor, por su tiempo y esfuerzo en la corrección de este documento. Gracias por su paciencia y dedicación durante este proceso.

Y a todas las personas que han formado parte de mi camino, les estoy profundamente agradecido por su contribución y por hacer que esta aventura valga la pena.

Resumen

El avance tecnológico y la expansión de internet han intensificado la propagación de noticias falsas, generando así una gran preocupación en nuestra sociedad. Para abordar este problema en Perú, se desarrolló un modelo detector de fake news basado en aprendizaje automático. Se construyó un dataset de noticias locales a través de la API de Twitter, el cual fue analizado y procesado para entrenar los algoritmos Naive Bayes y Support Vector Machine. Además, se planteó un modelo híbrido que combinó los conocimientos de ambos algoritmos para un análisis más preciso, dicho modelo fue implementado en un bot de Telegram como prototipado de aplicación. Los resultados obtenidos indicaron una alta precisión en la detección de noticias falsas, con un 95,4 % para Naive Bayes, un 90,45 % para Support Vector Machine y un 95,35 % para el modelo híbrido. Se concluye que la construcción del dataset es un proceso complejo y la precisión de los modelos depende en gran medida de la calidad de los datos y características del mismo, por otro lado, los modelos desarrollados en la investigación muestran un alto grado de efectividad en la detección de noticias falsas en nuestros medios locales.

Palabras clave: Noticias falsas, aprendizaje automático, clasificador Bayesiano Ingenuo, Máquina de Vector de Soporte.

Abstract

The technological advancement and the expansion of the internet have intensified the spread of fake news, generating great concern in our society. To address this problem in Peru, a fake news detector model based on machine learning was developed. A dataset of local news was built through the Twitter API, which was analyzed and processed to train the Naive Bayes and Support Vector Machine algorithms. In addition, a hybrid model was proposed that combined the knowledge of both algorithms for a more accurate analysis, this model was implemented in a Telegram bot as an application prototype. The results indicated a high accuracy in detecting fake news, with 95.4 % for Naive Bayes, 90.45 % for Support Vector Machine, and 95.35 % for the hybrid model. It is concluded that building the dataset is a complex process and the accuracy of the models depends greatly on the quality of the data and its characteristics, on the other hand, the models developed in the research show a high degree of effectiveness in detecting false news in our local media.

Keywords: Fake news, machine learning, Naive Bayes, Support Vector Machine.

Introducción

La sociedad contemporánea se enfrenta a uno de los desafíos más graves de su historia: la propagación de noticias falsas. Estas *fake news* tienen un impacto negativo no solo en el ámbito político, sino también en el económico y social, y han sido intensificadas por el avance de la tecnología y el internet. Durante la pandemia de COVID-19, la desinformación ha afectado aún más la salud de las personas. En Perú y América Latina, existen algunas plataformas digitales que buscan detener la propagación de noticias falsas, pero el problema se ha agravado en la actualidad.

En este contexto, el presente trabajo tiene como objetivo implementar un modelo de detección de noticias falsas en los medios de comunicación que tienen una reputación cuestionable con el fin de reducir la desinformación en el Perú. Este modelo utiliza tanto el clasificador Naive Bayes (NB) como el optimizador Support Vector Machine (SVM) para realizar predicciones precisas sobre la veracidad de las noticias. La implementación de este modelo en los medios de comunicación será clave para alcanzar el objetivo general de reducir la desinformación en el país y mejorar la calidad de la información que reciben los ciudadanos.

En el Capítulo 1 se presenta detalladamente el planteamiento y formulación del problema, se establecen los objetivos a alcanzar, se brinda una justificación sólida y se delimita claramente el alcance del proyecto. Además, se describe con precisión la metodología empleada para llevar a cabo la investigación y se incluye un cronograma de actividades exhaustivo y actualizado.

En el Capítulo 2 se examinan cinco estudios internacionales relevantes sobre la detección de noticias falsas. Además, se presenta una revisión de las teorías clave que sustentan el proyecto, incluyendo las *fake news*, el aprendizaje automático y los modelos NB y SVM que se utilizaron en el mismo. Finalmente, se ofrece una definición concisa de las librerías de software utilizadas en el proyecto.

El Capítulo 3 se enfoca en una descripción detallada y exhaustiva del desarrollo de los modelos de aprendizaje automático. Se incluye información sobre la creación y adquisición del conjunto de datos que se llevó a cabo mediante la utilización de la Application Programming Interface (API) de Twitter, seguida de un proceso de limpieza y filtrado de ruido. Además, se realiza un análisis riguroso de los datos, se construyen histogramas de palabras y se extraen características importantes para el desarrollo de los algoritmos NB y SVM mencionados previamente.

El Capítulo 4 se centra en la evaluación de los modelos de aprendizaje automático desarrollados en el Capítulo 3. Se llevan a cabo pruebas exhaustivas de rendimiento utilizando matrices de confusión, y se ajustan los parámetros de entrada para determinar las configuraciones óptimas para cada modelo. De esta manera, se logra obtener un mejor ren-

dimiento y una mayor precisión en la detección de noticias falsas. Además, se proporciona una representación gráfica de las características distintivas entre ambas clases de noticias, lo que facilita una comprensión clara y concisa de los resultados obtenidos.

En el Capítulo 5 se aborda una mejora en los modelos desarrollados anteriormente. Se propone un algoritmo híbrido que combina las habilidades de dos técnicas, SVM y NB, para obtener una detección más precisa de noticias falsas. El algoritmo híbrido aprovecha la capacidad semántica y sintáctica del modelo SVM con la información léxica obtenida por el algoritmo NB, lo que permite mejorar la precisión en la detección de noticias falsas.

En el Capítulo 6 se presenta el prototipo de una aplicación de detección de noticias falsas basado en el modelo previamente entrenado. Se describe el proceso de desarrollo de un bot de Telegram. Además, se explica cómo se importó y utilizó el conocimiento obtenido en el proceso de desarrollo del prototipo. Finalmente, se incluyen imágenes para ilustrar el funcionamiento y uso del bot de detección de noticias falsas.

En el Capítulo 7, se lleva a cabo una evaluación detallada y discusión de los resultados logrados en el proyecto en comparación con los objetivos establecidos y los antecedentes revisados. Se incluyen también una descripción de los desafíos y obstáculos que se presentaron durante el proceso de investigación y desarrollo del proyecto.

Finalmente, se presentan conclusiones en relación a los objetivos establecidos. Además, se ofrecen recomendaciones sobre cómo se puede mejorar el proyecto en el futuro, optimizando la adquisición y procesamiento de datos para aumentar la precisión y lograr resultados más fiables. Por último, se adjunta como anexo una tabla con noticias de prueba externas al *dataset*.

Listado de abreviaturas

API Interfaz de Programación de Aplicaciones.

BERT Bidirectional Encoder Representations from Transformers.

BM Benchmarking Method.

BoW Bag of Words.

CVXOPT Python Software for Convex Optimization.

DRAE Diccionario de la Real Academia Española.

DS Dataset.

KDD Knowledge Discovery in Databases.

LR Regresión Logística.

ML Aprendizaje Automático.

NASA Administración Nacional de Aeronáutica y el Espacio.

NB Clasificar Bayesiano Ingenuo.

NDW Non Dictionary Words.

NLP Procesamiento de Lenguaje Natural.

NLTK Natural Language Toolkit.

OMS Organización Mundial de Salud.

OPS Organización Panamericana de Salud.

POS Part of Speech Tagging.

RF Random Forest.

SAS Sentiment Analysis Spanish.

SVM Máquina de Vector de Soporte.

SVM-NB Modelo Híbrido.

TDLib Telegram Database Library.

TF-IDF Term Frecuency – Inverse Document Frequency.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Aspectos generales

1.1. Planteamiento del problema

1.1.1. Descripción del problema

Las noticias falsas han sido un problema en la humanidad durante miles de años, actualmente, con el avance de la tecnología y el internet, se ha vuelto aún más fácil difundirlas. Estas noticias falsas, también conocidas como *fake news*, afectan no solo el ámbito político, sino también el económico y social. Durante la pandemia de COVID-19, estas noticias tuvieron un impacto significativo en la salud de las personas en todo el mundo. Afortunadamente, existen sitios web sin fines de lucro que se dedican a combatir esta desinformación mediante la metodología de *fact-checking*, que consiste en evaluar la veracidad de una noticia en base a fuentes oficiales, autores y su impacto social.

En Perú y América Latina, existen varias plataformas digitales que buscan detener la propagación de noticias falsas mediante proyectos colaborativos, como *Verificador*, *Comproba*, *RedCheq*, entre otros. Estas plataformas realizan búsquedas exhaustivas en internet para verificar la veracidad de una afirmación. Sin embargo, debido a la situación actual, este problema se ha agravado aún más, ya que la desinformación no solo se genera por los usuarios sino también por algunos noticieros o periódicos de poco prestigio.

La propuesta que se lleva a cabo con el presente trabajo se basa en un modelo de Machine Learning (ML) que utiliza el clasificador NB y el optimizador SVM, con el fin de obtener resultados precisos sobre la veracidad de estas noticias en nuestro país.

1.1.2. Identificación del problema

Desinformación de la población peruana por la publicación de noticias falsas en los medios de comunicación del país.

1.2. Formulación del problema

1.2.1. Problema general

Existencia masiva de noticias falsas publicadas en los noticieros peruanos, desinformando a la población por la falta de filtros reguladores de las mismas.

1.2.2. Problemas específicos

- No se cuenta con modelos predictivos que identifiquen noticias falsas basados en el contexto nacional.
- No se cuenta con un *dataset* de noticias peruanas con las características necesarias como para entrenar el modelo desarrollado en el proyecto.

1.3. Objetivos

1.3.1. Objetivo general

Implementar un modelo de detección de noticias falsas en los medios de comunicación que tienen una reputación cuestionable con el fin de reducir la desinformación en el Perú.

1.3.2. Objetivos específicos

- Construir un *dataset* con noticias verdaderas y falsas de las páginas web de los medios informativos en Perú.
- Limpiar y procesar los elementos del *dataset*.
- Desarrollar un modelo predictivo basado en el clasificador *Naive Bayes* y el optimizador *Support Vector Machine*.
- Entrenar el modelo y verificar la precisión de los resultados obtenidos mediante pruebas de *benchmarking* y matrices de confusión.

1.4. Justificación

1.4.1. Conveniencia

Si bien, estos últimos años la sociedad se ha vuelto consciente de que no toda la información en las redes sociales o medios digitales es útil o veraz, la gravedad del problema no disminuye, este tipo de noticias se difunden con mayor facilidad, manipulan la opinión pública y son una amenaza.

Por ello, es de suma importancia el planeamiento y desarrollo de nuevas técnicas para contrarrestar la propagación de estas noticias y, de esta manera, garantizar la transparencia de nuestros medios informativos nacionales. Entre éstas, se incluyen técnicas de detección manual como el *fact-checking* y autónoma como los modelos predictivos de ML, este último con capacidades superiores para realizar dicha tarea, y con un amplio camino por delante para conseguir predicciones más eficientes durante los siguientes años.

1.4.2. Relevancia

La importancia en el desarrollo del presente proyecto radica principalmente en combatir el problema actual de desinformación en nuestra sociedad, utilizando como alternativa un modelo de ML, el cual es capaz de detectar las noticias con poco o nulo respaldo en fuentes confiables. El auge en tecnología nos obliga a estar preparados para los problemas futuros, las difusión de *fake news* apenas está comenzando, éste es el momento de dar un paso firme en contra de la desinformación mediática, protegiendo la información realmente importante y descartando las publicaciones sin sustento alguno.

1.4.3. Implicancias prácticas

- Modelos de ML más complejos, con la capacidad de hacer predicciones precisas de noticias publicadas en formatos de audio, imagen o video, siendo éstos, tipos de archivos más comunes y difundidos actualmente.
- Páginas web y aplicaciones móviles capaces de detectar en tiempo real si la información proporcionada en sus plataformas es veraz.
- Modelos predictivos aplicando novedosas técnicas de ML no solo para estudios sobre *fake news*, sino también para análisis sentimental, procesamiento de lenguaje natural u otras investigaciones sociales basadas en el Perú y localidades de habla hispana, ya que el *dataset* del proyecto podrá ser reutilizado para este tipo de proyectos.

1.4.4. Valor teórico

- Un modelo de ML detector de noticias falsas basado plenamente en el contexto peruano, entrenado con un *dataset* de noticias locales recuperadas de los medios informativos más populares en las redes sociales.

1.4.5. Utilidad metodológica

El *dataset* del proyecto podrá ser reutilizado en futuros trabajos de investigación basados en análisis sentimental o procesamiento de lenguaje natural en el contexto peruano. Mientras que el modelo predictivo podrá ser un referente de múltiples aplicaciones futuras para la detección de *fake news* en el país, aplicando novedosas técnicas de ML.

1.5. Delimitación de estudio

1.5.1. Delimitación espacial

El área de investigación del proyecto son las publicaciones digitales, en formato de texto, de noticieros y medios informativos en el territorio peruano, las cuales son apropiadas para el entrenamiento del modelo desarrollado.

1.5.2. Delimitación temporal

El desarrollo de esta propuesta de investigación se llevó a cabo durante los meses de noviembre del año 2021, donde se inició la construcción gradual del *dataset*, hasta enero de 2023, periodo dedicado para el desarrollo del modelo predictivo.

1.6. Método

1.6.1. Alcance

- El modelo de *machine learning* tiene la capacidad de detectar la veracidad de noticias en formato de texto, las cuales deben tener una longitud mínima de cuarenta palabras, por motivos de precisión.
- Se utiliza el clasificador NB y el optimizador SVM para la detección de noticias falsas, siendo ambos algoritmos altamente eficientes para este propósito.

1.6.2. Diseño

En este estudio se utiliza un enfoque de investigación descriptivo, de acuerdo a Hernández Sampieri et al. (2014). El diseño metodológico se basa en la recolección de noticias de cada clase mediante la API de Twitter. Los datos recolectados son analizados mediante técnicas y librerías específicas para el análisis sentimental y emocional de los textos. Posteriormente, se realizan comparaciones y análisis a través de gráficos y tablas para obtener patrones y tendencias en los datos. Estos resultados son utilizados para entrenar los modelos predictivos descritos en el proyecto, finalmente, se interpretan dichos resultados y se obtienen conclusiones.

1.6.3. Para el desarrollo de la parte informática

Para cumplir con los objetivos planteados, se considera la metodología Knowledge Discovery in Databases (KDD):

1. Comprensión del dominio de estudio y establecimiento de objetivos: Se reconocen las fuentes de información más importantes y se establecen los límites y objetivos de lo que se pretende.

2. Creación de un *dataset*: Se recopilan noticias verdaderas y falsas de los medios de información más utilizados en internet, tanto como redes sociales y páginas web informativas de Perú.

3. Limpieza y procesamiento de datos: Se elimina el ruido, las inconsistencias y los datos duplicados del *dataset* para evitar resultados inválidos o poco confiables.

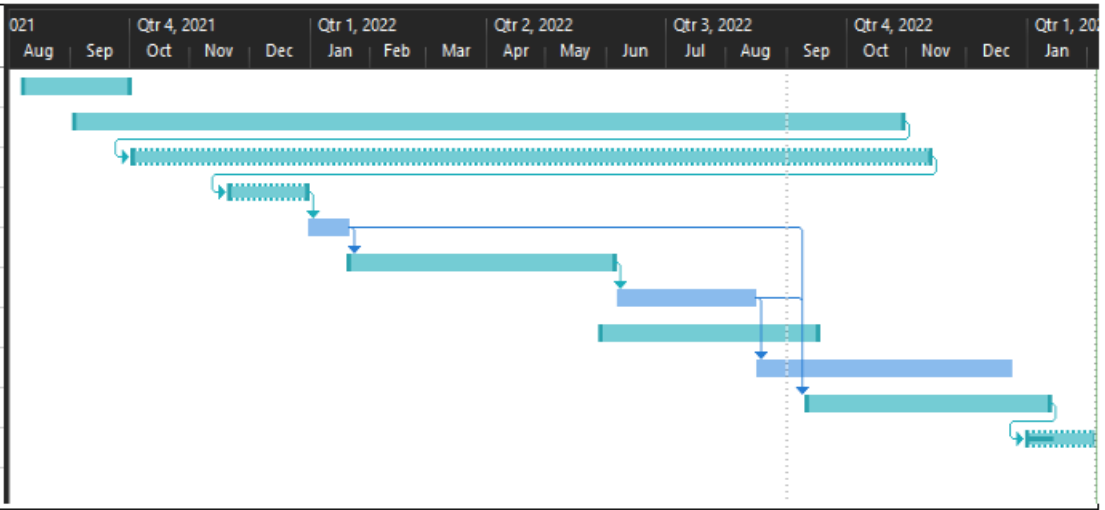
4. Minería de datos: Se descubren patrones y relaciones presentes en los datos mediante los algoritmos y técnicas seleccionados.

5. Interpretación de los patrones minados: Los resultados se presentan en un formato entendible por medio de técnicas de visualización.

6. Utilización del conocimiento descubierto: Se documenta y presenta un informe del conocimiento descubierto para incorporarlo en futuros sistemas.

1.6.4. Cronograma de actividades

ID	Actividad
1	Revisión de antecedentes y establecimiento de objetivos
2	Creación gradual del dataset de noticias
3	Limpieza y preprocesamiento de datos
4	Desarrollo de algoritmo Naive Bayes
5	Verificación e interpretación de resultados Naive Bayes
6	Desarrollo de algoritmo Support Vector Machine
7	Verificación e interpretación de resultados Support Vector Machine
8	Planteamiento y desarrollo de Modelo Híbrido
9	Comparación de resultados y optimización de código
10	Documentación del desarrollo del proyecto
11	Retroalimentación general



Project: CronogramaActividade
Date: Mon 2/6/23

Task	Inactive Summary	External Tasks
Split	Manual Task	External Milestone
Milestone	Duration-only	Deadline
Summary	Manual Summary Rollup	Progress
Project Summary	Manual Summary	Manual Progress
Inactive Task	Start-only	
Inactive Milestone	Finish-only	

Figura 1.1: Cronograma de actividades para el proyecto

Capítulo 2

Marco teórico

2.1. Antecedentes

2.1.1. Antecedentes internacionales

Ahmad et al. (2020). *Fake News Detection Using Machine Learning Ensemble Methods*, University of Engineering and Technology, Peshawar, Pakistan.

Conclusiones:

- La tarea de clasificar noticias manualmente requiere dominio y experiencia para identificar anomalías en el texto. Un paso importante para reducir su difusión es identificar los elementos clave que intervienen en ellas.
- La teoría de grafos y técnicas de ML pueden emplearse para identificar las fuentes clave implicadas en la difusión de las noticias falsas. Las pruebas realizadas con cuatro *datasets* diferentes, divididos en un 70/30, obtuvo resultados del 98 %, 31 %, 54 % y 88 % respectivamente, utilizando un SVM lineal.

Comentario: En base a este trabajo, podemos considerar importante la extracción de características, las cuales incluyen porcentajes de palabras con implicancias positivas o negativas, *stop words* y elementos gramaticales como adjetivos, verbos o preposiciones. Adicionalmente a ello se complementan estas propiedades con un grupo de 93 características adicionales utilizando el software externo LIWC2015. También se mencionan los kernels utilizados para separar el conjunto de datos al utilizar una SVM, los cuales son sigmoide, polinomial, gaussiano y lineal. Según las estadísticas mostradas, este trabajo obtuvo altos puntajes de precisión, llegando a valores de 99 % para la técnica de *Random Forest* y 98 % para el SVM.

Zhang et al. (2019). *FakeDetector: Effective Fake News Detection with Deep Diffusive Neural Network*, IFM Lab, Department of Computer Science, Florida State University, FL, USA.

Conclusiones:

- Basado en las conexiones entre artículos de noticias, creadores y sujetos de noticias, se ha propuesto un modelo *deep diffusive network* para incorporar la información de la

estructura de la red en el aprendizaje del modelo. Además de un nuevo modelo llamado GDU, el cual acepta múltiples entradas de diferentes fuentes simultáneamente y puede fusionar de manera efectiva estas entradas para generar salidas con puertas de *olvido* y *ajuste* de contenido.

- Los extensos experimentos realizados en un conjunto de datos de noticias falsas del mundo real, es decir, PolitiFact, han demostrado el rendimiento sobresaliente del modelo propuesto en la identificación de artículos, creadores y sujetos de noticias falsas en la red.

Comentario: En este trabajo se considera de gran utilidad la técnica de recopilación de textos para su *dataset*, utilizando la Twitter API se extraen 14,055 publicaciones de la cuenta oficial de PolitiFact, proyecto cuyo objetivo es recolectar conversaciones, artículos y publicaciones en redes sociales sobre temas políticos, para realizar un fact-checking y evitar su propagación. En este trabajo, se realiza también un análisis de la colección de los datos, para determinar las diferencias entre ambas clases de noticias.

Aldwairi and Alwahedi (2018). *Detecting Fake News in Social Media Networks*, College of Technological Innovation, Zayed University, Abu Dhabi, Emiratos Árabes Unidos.

Conclusiones:

- Se utilizó el software *WEKA*, específicamente los clasificadores *BayesNet*, *Logistic*, *Random Tree* y *Naive Bayes* para el análisis y minería de datos basados en métricas como: *Precision* y *Recall*.
- Los resultados al evaluar la capacidad de este método, mostraron un buen rendimiento a la hora de identificar posibles fuentes de noticias falsas. Para el algoritmo de NB se obtuvieron resultados de precisión de 98.7 %.

Comentario: De la misma manera que Ahmad et al. (2020) se utiliza un software externo para en este trabajo, los autores construyen el *dataset* a partir de artículos y publicidades de páginas como Facebook, Forex y Reddit en árabe e inglés, seleccionan los atributos necesarios y emplean los clasificadores propios de WEKA para realizar sus pruebas de *benchmarking*, los cuales muestran resultados de precisión mayores al 94 %. Este proyecto se enfoca principalmente en la detección de sitios web como fuentes de noticias falsas.

Younus Khan et al. (2021). *A benchmark study of machine learning models for online fake news detection*, Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh.

Conclusiones:

- Se encontró que los modelos basados en Bidirectional Encoder Representations from Transformers (BERT) han logrado un mejor rendimiento que todos los demás modelos en todos los conjuntos de datos, llegando a puntajes de precisión de hasta 98 %. Más importante aún, se encontró que los modelos pre-entrenados basados en BERT son robustos al tamaño del conjunto de datos y pueden funcionar significativamente mejor en un tamaño de muestra muy pequeño.

- Al realizar pruebas sobre los *datasets*: *Liar*, *Fake or real news* y *Combined Corpus*, se obtuvieron resultados del 56 %, 67 % y 71 % para el SVM, y 60 %, 86 % y 93 % para NB, respectivamente. Demostrando así que NB con n-gram puede lograr resultados similares a los modelos basados en redes neuronales en un conjunto de datos cuando el tamaño es suficiente.
- Los resultados y hallazgos basados en este análisis comparativo pueden facilitar futuras investigaciones y también ayudar a las organizaciones (por ejemplo, portales de noticias en línea y redes sociales) a elegir el modelo más adecuado que esté interesado en detectar noticias falsas.

Comentario: En este trabajo se explora los métodos más utilizados actualmente en la detección de *fake news* y se realiza un amplio y completo *benchmarking* de los mismos, para determinar cuales son los más eficientes. Los resultados obtenidos indican que los métodos con mayor precisión son los pre-entrenados BERT y los modelos NB y SVM, según el tamaño de los dataset con los que se experimente.

Espejel et al. (2022). *Detección automática de noticias falsas usando representaciones textuales tradicionales y soluciones basadas en aprendizaje profundo*, División de Investigación y Posgrado, Universidad Politécnica de Tulancingo y Departamento de Mecatrónica, Universidad Politécnica de Pachuca, Hidalgo, México.

Conclusiones:

- El desempeño de los modelos depende en gran medida de las características usadas en la representación del texto, así como en los algoritmos de aprendizaje automático y la configuración de la arquitectura aplicada para la tarea.
- En los experimentos con la colección 2021 se observa un desempeño menor al que se obtiene con el corpus 2020, obteniendo una precisión de 77.97 % para el SVM, lo que sugiere que los modelos evaluados tienen una dificultad para detectar las variaciones temáticas y lingüísticas que se presentan en este conjunto de noticias.

Comentario: En este trabajo se utiliza un *dataset* conformado por 2 colecciones de noticias presentadas en las competencias *FakeDeS 2020* y *FakeDeS 2021* en México, con más de 2500 instancias, de las cuales se utiliza el 70 % para entrenamiento y 30 % prueba. Se menciona también que los datos de entrenamiento fueron documentos publicados solo en México y los de prueba en múltiples países de Latinoamérica. Se realiza un análisis de las colecciones de datos como frecuencia de elementos gramaticales, longitud de los textos y cantidad de palabras vacías. Para finalmente comparar los resultados obtenidos con un modelo de referencia que utiliza los clasificadores SVM, Regresión Logística (LR) y Random Forest (RF). Los resultados obtenidos en este trabajo no superan el desempeño de su modelo de referencia.

2.2. Bases teóricas

2.2.1. Fake news

Las noticias son un compromiso con la verdad, nos referimos a hechos reales en el mundo. Las *fake news* son, en ese sentido, un oxímoron, noticias falsificadas intencionalmente, un contenido pseudoperiodístico, información engañosa presentada como noticia y difundida a través de diversos medios con el fin de generar desinformación en los ciudadanos. El término se volvió popular durante la campaña de Donald Trump en 2016 para desafiar a su contendiente Hillary Clinton y a los medios de comunicación estadounidenses New York Times y CNN. Como alude Roberto Rodríguez Andrés en su artículo "Trump 2016: ¿Presidente gracias a las redes sociales?":

> El manejo de internet y las redes sociales, especialmente Twitter y Facebook, ha sido señalado como uno de los factores que contribuyó al triunfo de Donald Trump en las elecciones de 2016 en los Estados Unidos. Rodríguez-Andrés (2018).

Figura 2.1: Mark Zuckerberg, cofundador de Facebook, haciendo mea culpa frente al comité del Senado

Fuente: Kurtzleben (2018). CEO de Facebook, Mark Zuckerberg testificando ante el Senado

Después de las elecciones de 2016, muchos temían que los artículos de noticias falsas difundidos en Facebook influyeran en los resultados de las elecciones. Kurtzleben (2018). En la Figura 2.1 se puede observar al cofundador de dicha red social, Mark Zuckerberg, testificando en la popular audiencia ante el comité del Senado, donde asumió la responsabilidad personal por la desinformación:

> "Es claro, ahora, que no hicimos lo suficiente para evitar que estas herramientas también se usaran para hacer daño. Eso se aplica a las noticias falsas, la interferencia extranjera en las elecciones y el discurso de odio, así como a los desarrolladores y la privacidad de los datos. No tuvimos una visión lo suficientemente amplia de nuestra responsabilidad, y eso fue un gran error. Fue mi error, y lo siento. Empecé Facebook, lo administro y soy responsable de lo que sucede aquí."

Estos últimos años las *fake news* se han convertido en una amenaza para la sociedad ya que inducen al error y manipulan decisiones. Como se menciona en United Nations News:

> El auge de las redes sociales ha generado mayor dificultad para una persona promedio en diferenciar hechos reales y desinformación. Fassihi (2018).

Durante la pandemia de COVID-19, la desinformación por *fake news* se ha incrementado y ha traído consigo catástrofes, según dijo el Dr. Tedros Adhanom Ghebreyesus, Director General de la OMS:

> La confianza pública en la ciencia y la evidencia es esencial para superar el COVID-19, por lo tanto, encontrar soluciones a la infodemia es tan vital para salvar vidas de la COVID-19 como las medidas de salud pública, como el uso de mascarillas y la higiene de las manos, para el acceso equitativo a vacunas, tratamientos y diagnósticos. Adhanom Ghebreyesus (2021).

Infodemia

Este nuevo término se refiere a la gran cantidad de información en períodos cortos de tiempo relacionada a temas específicos, principalmente en circunstancias concretas como la pandemia actual, esta información puede ser verídica o no y se propaga por internet tan rápidamente como un virus. La Organización Panamericana de Salud (OPS) junto a la Organización Mundial de Salud (OMS) se han pronunciado al respecto en su hoja informativa titulada "Entender la infodemia y la desinformación en la lucha contra la COVID-19", donde se narra:

> El brote de COVID-19 y la respuesta correspondiente han estado acompañados de una infodemia masiva, es decir, de una cantidad excesiva de información –en algunos casos correcta, en otros no– que dificulta que las personas encuentren fuentes confiables y orientación fidedigna cuando las necesitan. OPS (2020).

En el anterior documento también se describe de qué maneras la infodemia puede agravar la pandemia generando ansiedad, depresión, agobio, agotamiento emocional y empeorando la toma de decisiones de las personas.

2.2.2. Machine learning

El aprendizaje automático es un subcampo de las ciencias de la computación que tiene como objetivo principal desarrollar métodos que permitan a una computadora mejorar con la experiencia y en función de los datos con los que entrena. Como describe Ethem Alpaydin:

> El aprendizaje automático es programar computadoras para optimizar un criterio de rendimiento utilizando datos de ejemplo o experiencias pasadas. Tenemos un modelo definido hasta unos parámetros, y el aprendizaje es la ejecución de un programa informático para optimizar los parámetros del modelo utilizando los datos de entrenamiento o experiencia pasada. Alpaydın (2010).

Entre otras definiciones de *machine learning* se encuentra la de Jeremy Howard y Sylvain Gugger, en su libro de "Deep Learning for Coders with fastai & PyTorch", donde definen *machine learning* como:

> El entrenamiento de programas desarrollados al permitir que una computadora aprenda de su experiencia, en lugar de codificar manualmente los pasos individuales. Howard and Gugger (2020).

Como se mencionó, el ML es una de las ramas más amplias de las ciencias de la computación. Es por ello que existen numerosos términos utilizados en la gran mayoría de las técnicas de aprendizaje automático. A continuación se da una breve descripción de las expresiones más utilizadas:

Dataset

El Dataset (DS) es un conjunto de datos dividido en filas y columnas, donde cada una de ellas corresponde a una variable o característica de cada elemento. Existen grandes repositorios que almacenan *datasets* para uso público, sin embargo, al momento de redactar este documento, no se encontró uno que contenga noticias falsas con las características necesarias para los objetivos del proyecto, es por ello que se construirá uno propio.

El modelo de ML utilizará el DS para el *training* y *testing*, los porcentajes y parámetros serán ajustados manualmente según los resultados obtenidos.

Training dataset Conjunto de elementos utilizados durante el proceso de aprendizaje o entrenamiento, donde el modelo podrá adquirir conocimiento para futuras predicciones.

Validation dataset También conocido como *development set* es un conjunto de elementos utilizados para el *testing* o validación, donde se podrá comprobar si el conocimiento adquirido por el modelo es correcto.

Para la construcción del DS para el proyecto son necesarias técnicas como *web scrapping* o el uso de la API de Twitter para la extracción de textos desde páginas web o redes sociales:

API de Twitter Ésta es una herramienta muy útil, ya que puede ser utilizada para recuperar y analizar datos de la red social. En el proyecto se recuperan tweets desde cuentas de noticieros formales e informales para la creación del DS, proceso que se describe más detalladamente en el Capítulo 3.

Tabla 2.1: Niveles de accesso de la API de Twitter

Esencial	**Elevado**	**Investigación Académica**
500.000 tweets/mes	2 millones tweets/mes	10 millones tweets/mes
1 Proyecto por cuenta	1 Proyecto por cuenta	Archivo completo
1 entorno/proyecto	3 entornos/proyecto	Operadores avanzados
Limitado v1.1 estándar	Acceso v1.1 estándar	
Sin acceso v1.1 premium	Acceso v1.1 premium	

Fuente: twi (2021b). Niveles de acceso *Twitter API*

La Tabla 2.1 muestra los niveles de acceso que se describen en la documentación oficial de *Twitter Develop*, en el proyecto se trabaja con la versión *Elevated*, que otorga suficiente acceso para la construcción del DS. Como se describe también en la documentación, el *tweet object* que retorna la API tiene múltiples atributos como: *id*, *text*, *user*, *place*, *reply_count*, *entities*, *created_at*, entre otros. twi (2021a). Para el proyecto se consideran los atributos *id*, *text* y *created_at*, los cuales, nuevamente, se describen de forma detallada en el Capítulo 3.

Web scrapping Si bien la Twitter API otorga gran cantidad de datos útiles, existe información valiosa para el *dataset* en páginas web fuera de Twitter u otras redes sociales, es por ello que son necesarias técnicas como el *web scrapping*, el cual es un proceso para extraer datos desde el propio código HTML de las páginas web, esta técnica es muy usada en diversos negocios digitales que dependen del análisis y recolección de datos.

Extracción de características

La extracción de características es un proceso por el cual se obtiene un conjunto de variables más pequeño que el original, estas características deben ser diferenciadoras para generar resultados de predicción óptimos. Dishaa Agarwal, en su artículo web *"Guide For Feature Extraction Techniques"*, menciona:

> La técnica de extracción de características nos brinda nuevas características que son una combinación lineal de las características existentes. El nuevo conjunto de funciones tendrá valores diferentes en comparación con los valores de funciones originales. El objetivo principal es que se requieran menos características para capturar la misma información. Agarwal (2021a).

Para dar solución al problema principal de este trabajo de investigación, se considera una selección amplia de características diferenciadoras entre las noticias, las cuales se agrupan en:

- Análisis sentimental: Se refiere al sentimiento positivo, negativo o neutral que expresa la noticia. Para el proyecto se utilizan las librerías *PySentimiento* y Sentiment Analysis Spanish (SAS), la primera genera 3 características, una para cada propiedad, con puntajes de 0 a 1, mientras que la segunda produce únicamente una característica que toma valores cercanos a 0 para indicar sentimiento negativo, a 1 para positivo y a 0.5 para neutral.
- Análisis emocional: Se refiere al puntaje de felicidad, odio, ira o tristeza que expresa un texto. Gracias a ello se podrá determinar el nivel de imparcialidad dentro de una noticia. Para este trabajo de investigación se utiliza también la librería *PySentimiento*, la cual determina los puntajes basándose en *EmoEvent*, un *dataset* muy bien desarrollado para este análisis.
- Análisis de odio: Se refiere al puntaje de agresividad u odio expresado en un texto. Estas características se consideran muy importantes para la investigación, ya que se demuestra, en capítulos posteriores, que son muy diferenciadoras para ambas clases de noticias. Nuevamente se utiliza la librería *PySentimiento* para este propósito.
- *Non Dictionary Words*: Muestra la proporción de uso de palabras erróneas e inexistentes en el diccionario de la lengua española. Esta característica determinará el nivel de formalidad utilizado en una noticia. En el proyecto se utiliza la librería *autocorrect*, la cual determina si una palabra existe o no, luego se calcula una proporción de palabras inexistentes por cada noticia con valores entre 0 y 1.

En la Subsección 3.3.2 se describe de manera más detallada estas características y sus aplicaciones en el modelo SVM. Adicionalmente, en la Subsección 4.2.1 se realiza un *benchmarking* con estas propiedades para determinar las duplas más diferenciadoras.

Matriz de confusión

La matriz de confusión es una herramienta empleada en el ML para cuantificar el desempeño de un modelo. Las columnas representan los valores predichos por el modelo, mientras que las filas, los valores reales.

De esta manera, la matriz de confusión del proyecto tendrá 4 campos: cantidad de elementos verdaderos que son etiquetados como verdaderos (VV), verdaderos que son etiquetados como falsos (VF), falsos que son etiquetados como verdaderos (FV) y falsos que son etiquetados como falsos (FF). Como se muestra en la Tabla 2.2:

Tabla 2.2: Matriz de confusión para detección de noticias falsas

		Predicción	
		Verdadero	Falso
Etiqueta real	Verdadero	VV	VF
	Falso	FV	FF

Tomando en cuenta los elementos de la matriz de confusión, la precisión o *accuracy* será calculada de la siguiente manera:

$$Accuracy = \frac{VV + FF}{VV + FF + VF + FV} \tag{2.1}$$

donde:

- VV es la cantidad de elementos etiquetados correctamente como verdaderos.
- FF es la cantidad de elementos etiquetados correctamente como falsos.
- VF es la cantidad de elementos etiquetados incorrectamente como falsos.
- FV es la cantidad de elementos etiquetados incorrectamente como verdaderos.

Aplicando la fórmula anterior, a los datos obtenidos por medio del modelo, se espera obtener un umbral de *accuracy* mayor o igual al 80 % para dar por concluido este trabajo de investigación.

Modelo híbrido

Muchos de los algoritmos de aprendizaje automático son buenos realizando tareas específicas, pero estos algoritmos pueden no aprovechar todo el potencial de los datos con los que se trabaja. En *machine learning* se conoce como modelo o algoritmo híbrido a uno o varios algoritmos simples que trabajan juntos para complementarse entre ellos. Es así como pueden mejorar su capacidad de predicción y resolver problemas con mayor complejidad. Domo (2022)

Existen diversas técnicas para interactuar con los datos, las cuales dependen del problema que se intente resolver. En este proyecto de investigación se implementa un modelo híbrido que mejora la precisión del optimizador SVM aprovechando el conocimiento del clasificador NB, lo cual se describe con mayor detalle en la Sección 5.1.

2.2.3. Clasificador Naive Bayes

El clasificador bayesiano ingenuo es una técnica probabilística que asume que la presencia o ausencia de una característica no está relacionada con otra. JavaTPoint (sf).

Los clasificadores bayesianos ingenuos utilizan histogramas y pueden lograr resultados muy precisos en la detección de noticias falsas, es por eso que se optó por su uso para el proyecto.

Teorema de Bayes

El Teorema de Bayes es una proposición que fue planteada y publicada por Thomas Bayes en 1763, expresa la probabilidad condicional de un evento aleatorio A dado B en términos de la distribución de probabilidad condicional del evento B dado A y la distribución de probabilidad marginal de solo A. Parzen (2021). El clasificador NB es llamado *ingenuo* ya que utiliza de manera simplificada este teorema.

Definición. Sea $\{A_1,A_2,...,A_i,...,A_n\}$ un conjunto de sucesos mutuamente excluyentes y exhaustivos tales que la probabilidad de cada uno de ellos es distinta de cero (P$[A_i] \neq 0$ para i = 1,2,...,n). Si B es un suceso cualquiera del que se conocen las probabilidades condicionales P(B$|A_i$) entonces la probabilidad P(A_i|B) viene dada por la expresión:

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)} \tag{2.2}$$

donde:

- *P(A_i)* son las probabilidades a priori.
- *P(B$|A_i$)* es la probabilidad de B en la hipótesis A_i.
- *P($A_i|B$)* son las probabilidades a posteriori.
- *P(B)* es la probabilidad de que ocurra el evento B.

Histograma de palabras

El clasificador NB utiliza dos histogramas de palabras para determinar la probabilidad general de un texto por medio de probabilidades individuales de las palabras que lo conforman. Según la probabilidad mayor obtenida, el texto es etiquetado como una noticia verdadera o falsa. La Figura 2.2 muestra los histogramas generados en el *paper FakeDetector: Effective Fake News Detection with Deep Diffusive Neural Network* considerado como antecedente importante para este proyecto.

Token Este término tiene diversas aplicaciones en la informática tales como: seguridad, reducción de riesgos o mapeo de datos confidenciales, sin embargo, para el presente trabajo de investigación se considerarán *tokens* a los componentes léxicos como palabras. De esta manera, cada noticia dentro del DS está conformada por varios *tokens* o palabras.

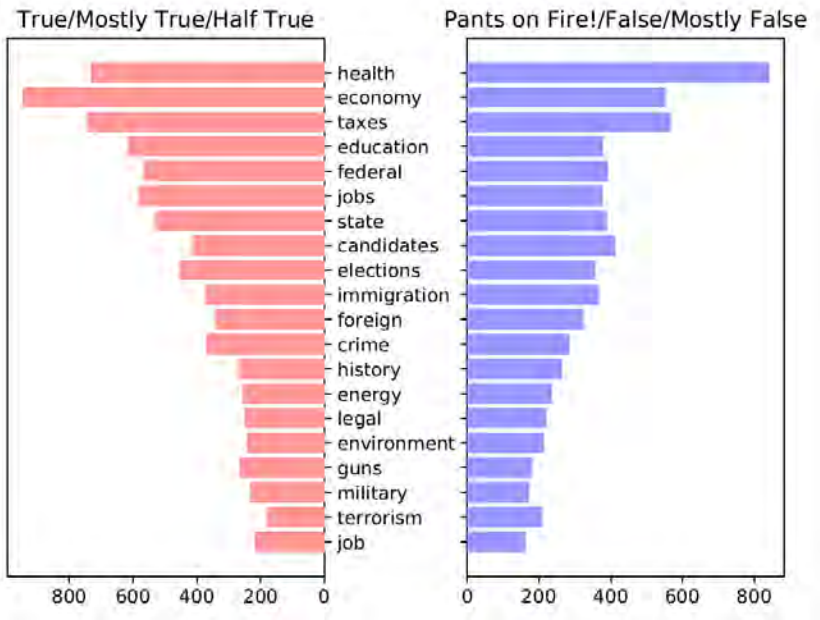


Figura 2.2: Histogramas de noticias verdaderas y falsas generadas por *FakeDetector*

Fuente: Zhang et al. (2019). Top 20 temas por cantidad de artículos

Stop words También conocidas como *negative dictionary* son todas aquellas palabras que no aportan significado a un texto, tal y como hacen los sustantivos, adjetivos, verbos o adverbios. Es por ello que se consideran *stop words* a las preposiciones, cojunciones y artículos. En el proyecto se utilizará la librería Natural Language Toolkit (NLTK) para obtener dichas palabras en español y poder validarlas dentro del trabajo y de esta manera reducir el ruido en las predicciones.

Técnica de alisamiento En algunas casos sucede que el *testing dataset* presenta tokens que no existen ni se observan en el *training dataset*, por lo tanto la probabilidad del token sería de 0, ocasionando, a su vez, que la probabilidad de todo el texto también obtenga el mismo valor erróneo. Ésta suele considerarse una desventaja del algoritmo NB, es por ello que se utiliza la técnica de alisamiento, la cual consiste en inicializar la cantidad de cada palabra dentro de los histogramas con un valor de 1.

2.2.4. Optimizador Support Vector Machine

Una SVM es un método complejo que separa matemáticamente grupos de elementos con características distintas. Esta separación puede darse como una línea recta, un plano recto o un hiperplano para N dimensiones. Como describen Nello Cristianini y John Shawe-Taylor:

> Las máquinas de vectores de soporte (SVM) son sistemas de aprendizaje que utilizan un espacio de hipótesis de funciones lineales en un espacio de características de alta dimensión, entrenado con un algoritmo de aprendizaje de la teoría de optimización que implementa un sesgo de aprendizaje derivado de la teoría de aprendizaje estadístico. Cristianini and Shawe-Taylor (2000).

Para poder entender mejor el funcionamiento de una SVM se deben tener en consi-

deración otros términos utilizados en el algoritmo, tales como *clusters*, hiperplanos y *kernel trick*:

Cluster

Los *clusters* son conjuntos o agrupaciones de elementos con características similares, tal y como se puede apreciar en la Figura 2.3. La máquina de vector de soporte traza un hiperplano entre los clústeres, para, de esta manera, poder clasificar los nuevos elementos.

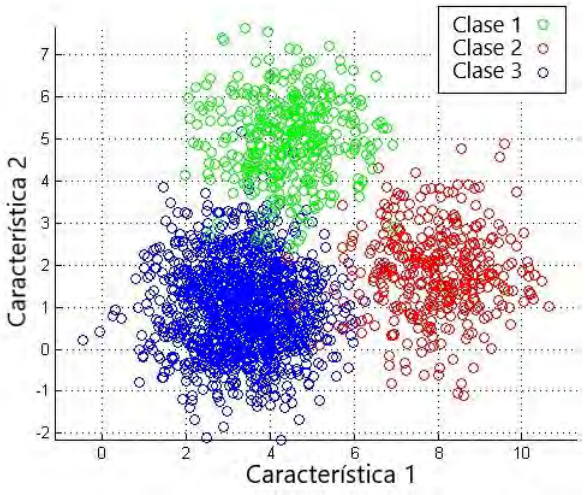


Figura 2.3: *Clustering* para problema *K-Means*

Fuente: Joshi (2013). *Clustering* para problema no supervisado utilizando *K-Means*

Producto escalar

También llamado producto punto o interno, es una operación algebraica de vectores que da como resultado un escalar.

Definición. Dado dos vectores a = (a_1, a_2, ..., a_n) y b = (b_1, b_2, ..., b_n), su producto escalar es la suma de los productos componente por componente, entonces se define como:

$$\mathrm{a} \cdot \mathrm{b} = (a_1 \cdot b_1 + a_2 \cdot b_2 + ... + a_n \cdot b_n)$$

Este producto de vectores es muy utilizado en el algoritmo SVM para determinar si la proyección de un elemento sobre otro punto perpendicular al hiperplano pertenece a la zona positiva o negativa del mismo. La Figura 2.4 muestra la proyección del vector a sobre el vector b.

Hiperplano

Un hiperplano es un subespacio menor en una dimensión al espacio ambiental, es decir, si el espacio es unidimensional (como una recta), el hiperplano es un punto, para un espacio bidimensional (como un plano), el hiperplano es una recta. Sin embargo, el término se utiliza generalmente para mayores dimensiones, que no pueden ser graficadas. Para la

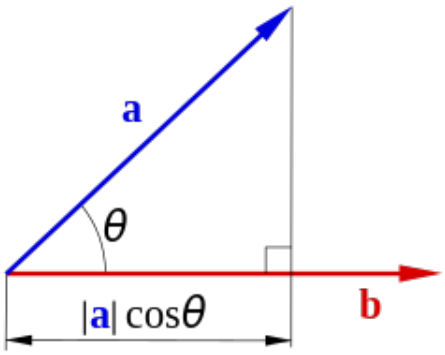


Figura 2.4: Proyección escalar de vector a sobre b

Fuente: Meyer (2000). Producto escalar entre vectores

máquina de vector de soporte, un buen hiperplano es aquel que maximiza la distancia entre los *clusters*, éste es el principal objetivo de la SVM. Es por eso que se encuentran varios hiperplanos que clasifican los datos pero se selecciona el que se encuentre más alejado de los vectores de soporte, o el que tenga el mayor *margin*.

Definición. Un hiperplano, generalizado para p-dimensiones se describe como:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p = 0 \tag{2.3}$$

Dados los parámetros β_0, β_1, β_2, ... y β_p, todos los pares de valores $\mathbf{x} = (x_1, x_2, ..., x_p)$ para los que cumple la igualdad son puntos del hiperplano.

Cuando $\mathbf{x}$ no satisface la ecuación:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p < 0 \tag{2.4}$$

Caso contrario:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p > 0 \tag{2.5}$$

De esta manera, si una nueva noticia es ingresada, y sus características $\mathbf{x}$ cumplen con la ecuación 2.4, entonces dicha noticia será etiquetada como falsa, por otro lado, si sus características cumplen con la ecuación 2.5 será etiquetada como noticia verdadera.

Margin

Se considera margen a la distancia entre el hiperplano y los elementos más cercanos, llamados vectores de soporte. Como se mencionó, el objetivo del optimizador SVM es maximizar esta distancia, un gran margen es considerado bueno para el modelo. Existen dos tipos de márgenes en este algoritmo, *hard margin* y *soft margin*.

Hard margin Este margen no permite ninguna clasificación errónea. En caso de que nuestros datos no sean separables de manera lineal, la SVM de margen duro no devolverá

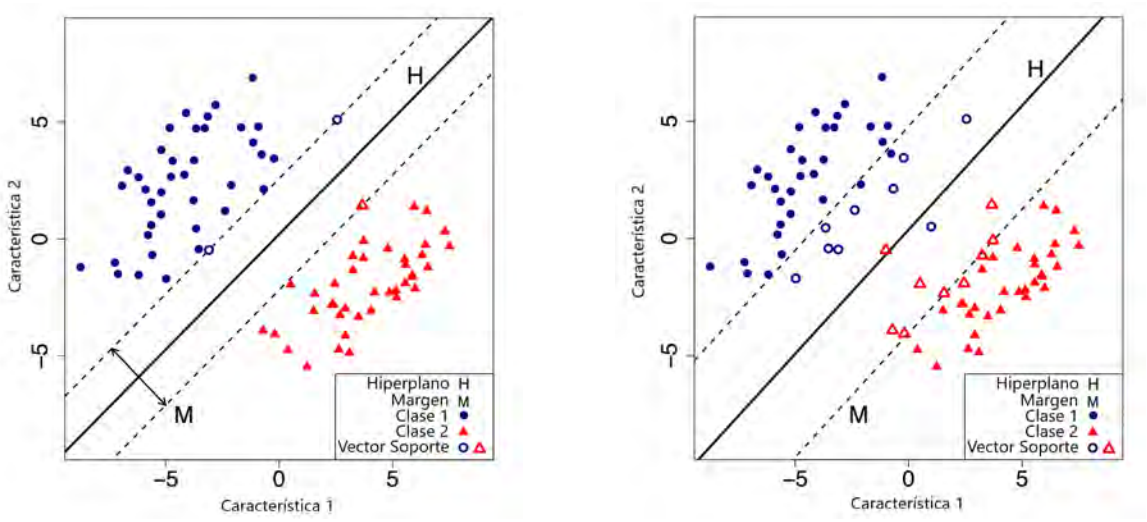
ningún hiperplano, ya que no podrá separar los datos. Agarwal (2021b)

El *hard margin* tiene poca aplicación práctica. Cuando se trabaja con datos de la vida real, como en el proyecto, los *datasets* no suelen ser perfectamente separables y este margen puede tener problemas de *overfitting* o sobreajuste, lo cual se refiere a sobreentender demasiado las muestras de entrenamiento disponible. Como describe Joaquín Amat Rodrigo sobre el *hard margin*, en su artículo titulado *"Máquinas de Vector de Soporte (SVM) con Python"*:

> Incluso cumpliéndose estas condiciones ideales, en las que exista un hiperplano capaz de separar perfectamente las observaciones en dos clases, esta aproximación sigue presentando dos inconvenientes: es muy sensible a la variación de los datos (poca robustez) y suele conllevar problemas de *overfitting*. Amat Rodrigo (2020).

Soft margin Por otro lado el *soft margin* o margen suave permite que algunos puntos estén dentro del margen y otros se encuentren en el lado equivocado del hiperplano. Es así como el margen suave evita sobreajustes en el modelo y puede aplicarse a casos reales donde no sea posible encontrar un hiperplano que separe perfectamente las dos clases debido a la presencia de ruido en los datos o a la superposición de clases.

El margen suave es controlado por un parámetro de regularización C, que determina la importancia de los puntos de error. Un valor grande de C permite un margen estrecho y menos errores, mientras que un valor pequeño de C permite un margen más amplio y más errores. Los resultados de las pruebas de rendimiento que se realizarán en los capítulos siguientes demostrarán las configuraciones óptimas al variar dicho parámetro C.



(a) Ejemplo *Hard Margin* (b) Ejemplo *Soft Margin*

Figura 2.5: Diferencia entre márgenes utilizados en la SVM

Fuente: Agarwal (2022). *Hard* y *Soft* SVM

En la Figura 2.5 se muestra la diferencia entre ambos tipos de márgenes, el de la izquierda muestra que los datos son perfectamente separables, este caso es poco probable al trabajar con datos de la vida real, por otro lado, la figura de la derecha muestra clústeres con datos superpuestos, este caso es más común, y solo puede ser resuelto utilizando un margen suave, que admitirá algunos errores dentro del mismo y podrá determinar así un hiperplano H óptimo para resolver el problema.

Kernel trick

El éxito en la precisión de muchos algoritmos de ML se basa en la adecuada elección del espacio de características para nuestro problema. El *kernel trick* es un método por el cual se transforma el espacio del conjunto de datos a uno más sencillo de clasificar, como se puede ver en la Figura 2.6. La mayoría de los *datasets* de la vida real generan *clusters* que no pueden ser separados como ocurre con los generados en el proyecto, es por ello que es necesaria la aplicación del *kernel trick*.

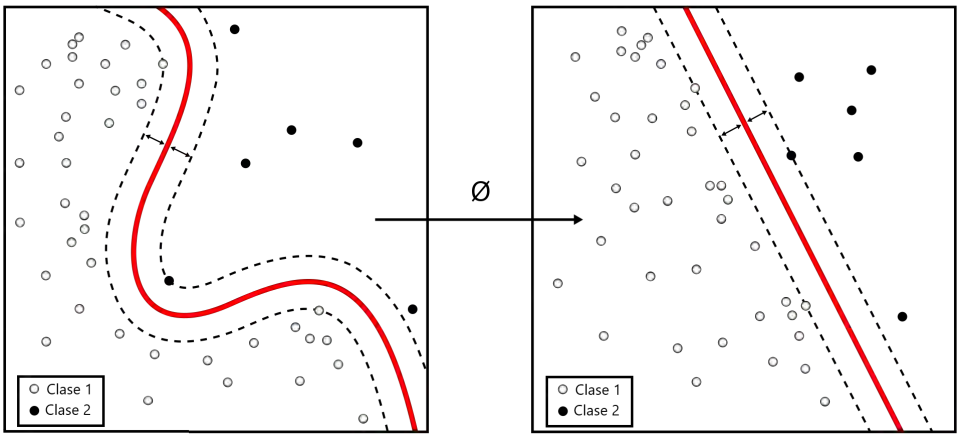


Figura 2.6: *Kernel machine* utilizada para convertir una función no lineal a lineal

Fuente: Jin and Wang (2012). Kernel relacionado a la transformación ϕ

Considerando la matriz de kernel K definida como:

$$K = \begin{vmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \cdots & k(x_n, x_n) \end{vmatrix} \tag{2.6}$$

Para transformar el espacio del conjunto de datos del proyecto se utilizan los kernel típicos: lineal, polinomial y gaussiano:

- **Lineal:**

$$k(x_i, x_j) = x_i^T x_j \tag{2.7}$$

  Donde x_i y x_j son dos vectores de entrada, y T representa la transposición o el proceso de intercambiar las filas y columnas del vector x_i.

- **Polinomial:**

$$k(x_i, x_j) = (x_i^T x_j + 1)^p \tag{2.8}$$

  Donde x_i y x_j son dos vectores de entrada, T representa la transposición del vector x_i y p es la potencia especificada.

- **Gaussiano:**

$$k(x_i, x_j) = exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2}) \tag{2.9}$$

  Donde x_i y x_j son dos vectores de entrada, *sigma* es el parámetro especificado para ajustar la forma del kernel y $||x_i - x_j||$ representa la norma euclidiana o distancia entre los vectores x_i y x_j.

2.2.5. Naive Bayes vs Support Vector Machine

Al tener un panorama más amplio respecto a las técnicas utilizadas, se planea obtener un único modelo que aproveche el conocimiento de ambos algoritmos. Por ello es necesario comprender sus ventajas y desventajas.

Tabla 2.3: Diferencias entre Naive Bayes y Support Vector Machine

Comparación entre los modelos utilizados	
NB	**SVM**
1. Implementación sencilla.	**1.** Implementación compleja.
2. Alta precisión en la clasificación de textos.	**2.** Su precisión depende de las características extraídas de los datos.
3. Requiere menor potencia de cómputo.	**3.** Requiere mayor potencia de cómputo.
4. Las predicciones pueden fallar al no considerar la sintaxis de los textos.	**4.** Es un optimizador, busca el mejor hiperplano para sus predicciones.
5. La persistencia de su conocimiento tiene tamaño reducido, es portable.	**5.** La persistencia de sus datos es extremadamente pesada, requiere sus analizadores sentimentales y emocionales.

De acuerdo a la Tabla 2.3, la técnica NB ofrece varias ventajas en comparación con el algoritmo SVM. Entre ellas se destacan su portabilidad, sencillez de implementación, la baja potencia de cómputo necesaria para su entrenamiento y una alta precisión en la detección de noticias falsas. Por otro lado, la implementación del SVM es más compleja, requiere mayor potencia de cómputo para su entrenamiento y su precisión depende en gran medida de las características consideradas y la representación de los datos. Sin embargo, el SVM utilizado en este proyecto tiene la capacidad de entender el contexto sentimental y emocional de las noticias, mientras que el NB no considera la sintaxis de los textos.

2.2.6. Librerías utilizadas

- **NumPy:** Librería *open source*, cuyo objetivo es la computación numérica en Python. Fue creado en 2005, basándose en el trabajo inicial de las bibliotecas Numeric y Numarray. NumPy siempre será un software 100 % de código abierto y de uso gratuito para todos. Berg (2022).
- **Pandas:** Proyecto *open source* para la manipulación de datos, lectura y escritura en estructuras de diferentes formatos como CSV, bases de datos SQL y archivos de texto.
- **ScikitLearn:** Librería *open source* de *machine learning* para el lenguaje de programación Python diseñada para interactuar con otras bibliotecas numéricas como NumPy. Cuenta con varios algoritmos de clasificación, regresión y agrupamiento, que incluyen máquinas de vectores de soporte, bosques aleatorios, aumento de gradiente y k-means. En el proyecto, ScikitLearn es utilizada para comparar los resultados obtenidos de sus algoritmos propios frente a los desarrollados en el proyecto. Fabian et al. (2011).
- **Matplotlib:** Librería de Python especializada en la creación de gráficos en dos dimensiones, los cuales incluyen histogramas, diagramas de barras, líneas, áreas, entre otros. Los gráficos propios del proyecto son generados utilizando esta herramienta.

- **NLTK:** *Natural Language Toolkit* se creó originalmente en 2001 como parte de un curso de lingüística computacional en el Departamento de Informática y Ciencias de la Información de la Universidad de Pensilvania. Desde entonces, se ha desarrollado y ampliado con la ayuda de docenas de colaboradores. Ahora se ha adoptado en cursos en docenas de universidades y sirve como base para muchos proyectos de investigación de procesamiento de lenguaje natural. Bird et al. (2009).

- **Tweepy:** Librería de Python que facilita el acceso a la API de Twitter, tiene muchas funcionalidades. Es utilizada en el trabajo de investigación para extraer tweets de los usuarios escogidos y construir así el *dataset*.

- **RE:** *Regular Expression Operations* es una librería de Python que ofrece operaciones con expresiones regulares que facilitan la limpieza de datos en el proyecto.

- **SAS:** *Sentiment-Analysis-Spanish* es una librería que utiliza redes neuronales convolucionales para predecir oraciones en español. El modelo fue entrenado con más de 800 mil comentarios de usuarios en internet. J. Bello (2021).

- **Autocorrect:** Librería en Python que sirve para determinar si una palabra existe o no en el diccionario de un idioma determinado.

- **PySentimiento:** Librería de código abierto para tareas de procesamiento de lenguaje natural en español e inglés, entrenada con *datasets* de terceros. Pérez et al. (2021).

- **CVXOPT:** *Python Software for Convex Optimization* es una librería de código libre basada en Python para optimizar el tratamiento de matrices y operaciones sobre las mismas, además es útil para la resolución de problemas lineales y cuadráticos. Andersen et al. (2022).

- **WordCloud:** Librería para generar nubes de palabras con diversos parámetros, como número y probabilidad de palabras verticales y horizontales, cantidad mínima de letras por palabra y una máscara para obtener gráficos con formas propias.

- **Pickle:** Herramienta de Python que proporciona una interfaz para guardar y cargar objetos Python en archivos binarios. Es fácil de usar y es compatible con la mayoría de las versiones de este lenguaje. Sin embargo, puede ser inseguro ya que puede ejecutar código malicioso.

- **TDLib:** Biblioteca de programación de código abierto desarrollada por Telegram para construir aplicaciones de mensajería. TDLib permite a los desarrolladores acceder a las funciones de Telegram de manera eficiente y segura, ofreciendo una amplia variedad de herramientas para el envío y recepción de mensajes, la gestión de chats y la interacción con usuarios.

- **Telethon:** Librería que permite a los desarrolladores construir aplicaciones en Python que se comuniquen con Telegram y realizar acciones como enviar y recibir mensajes, gestionar chats y canales, y mucho más. Telethon utiliza la TDLib de Telegram como base y ofrece una interfaz fácil de usar.

- **Asyncio:** Biblioteca de Python diseñada para manejar tareas asíncronas de manera eficiente. Permite a los programadores crear programas que pueden realizar varias tareas simultáneamente sin bloquear el hilo principal y sin tener que usar hilos adicionales.

Capítulo 3

Desarrollo del tema de tesis

3.1. Creación del dataset

Para entrenar el modelo desarrollado en la investigación es necesario un *dataset* robusto, libre de ruido y con información pertinente. Son poco comunes los repositorios de noticias en habla hispana, y aun más reducidos aquellos repositorios de textos peruanos con las características necesarias para este proyecto, es por esta razón que se diseña y construye un *dataset* propio de noticias utilizando la API de la red social Twitter para investigación.

3.1.1. Generación de token y credenciales

Para el empleo de la API de Twitter es necesario generar las credenciales que otorgan todos los permisos. Este proceso tarda entre 2 a 3 días dependiendo del servicio de atención de la red social. Para ello se redacta una solicitud indicando la aplicación que se dará con esta herramienta y la finalidad educativa o de investigación del proyecto.

Tras responder las preguntas requeridas por el personal de Twitter y enviar las evidencias digitales sobre los detalles del proyecto que se realiza, se nos concedió el permiso para utilizar su servicio en nivel *Elevated*, el cual tiene un límite de dos millones de tweets al mes y cincuenta peticiones cada quince minutos como se mostró en la Tabla 2.1, cantidad suficiente para construir la primera versión del *dataset*.

3.1.2. Desarrollo del script para extraer publicaciones de cuentas de twitter

El script que genera nuestro *dataset* es desarrollado en Python, ya que tiene múltiples librerías para una fácil conexión con la API de Twitter como TweePy, este script tiene como parámetros el nombre único de usuario de la cuenta que extraeremos los mensajes y la cantidad máxima de tweets que se extrae en cada ejecución. Para las primeras pruebas se recuperan los últimos mil tweets de cada noticiero seleccionado en la red social, dichos textos solo serán almacenados si cumplen con la condición de tener al menos 40 palabras.

Para la selección de noticieros confiables en nuestro país, se consideró el estudio realizado por el Instituto *Reuters* como describe Chacón (2022), el cual muestra porcentajes de confiabilidad para un grupo de noticieros preseleccionados en nuestro contexto nacional, como se puede observar en la Figura 3.1.

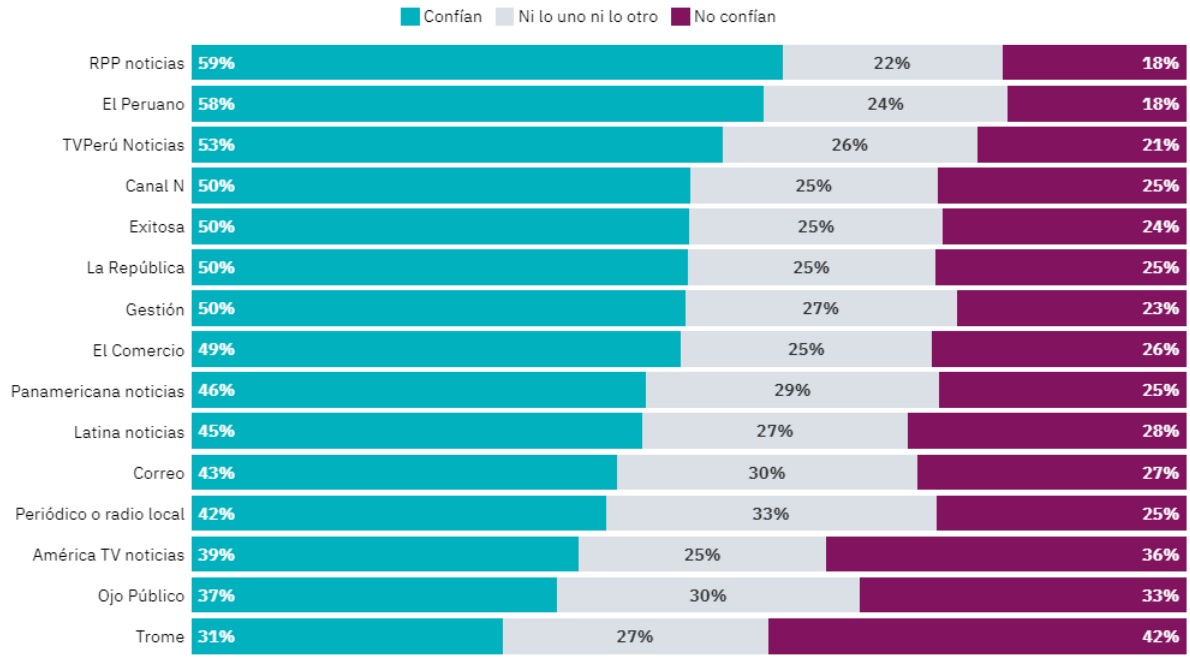


Figura 3.1: Puntuación de confianza en cada medio seleccionado por *Reuters Institute*

Fuente: Chacón (2022). Puntuación de confianza en medios peruanos

Es así como, en la Tabla 3.1 se muestran los 10 noticieros con más credibilidad del estudio mencionado, la mayoría de las publicaciones que realizan estas cuentas son revisadas arduamente y respaldadas por fuentes confiables. Entre los noticieros seleccionados se tienen: medios de televisión, radio o diarios con más de 30 años de emisión en el país.

Tabla 3.1: Noticieros peruanos etiquetados como confiables en el proyecto

	Noticiero	**Usuario Twitter**	**Credibilidad**
1	RPP	@RPPNoticias	59 %
2	Diario El Peruano	@DiarioElPeruano	58 %
3	TV Perú Noticias	@noticias_tvperu	53 %
4	Canal N	@canalN_	50 %
5	Exitosa Noticias	@exitosape	50 %
6	La República	@larepublica_pe	50 %
7	Diario Gestión	@Gestionpe	50 %
8	El Comercio	@elcomercio_peru	49 %
9	Panamericana noticias	@PTV_Noticias	46 %
10	Latina Noticias	@Latina_Noticias	45 %

Por otro lado, en la Tabla 3.2 se puede observar los noticieros y cuentas populares consideradas de poca credibilidad en la red social Twitter, sus publicaciones son, en su mayoría, comentarios propios y de odio, no es información respaldada en fuentes confiables, tampoco utilizan un lenguaje formal e imparcial, incluso algunas de ellas fueron bloqueadas por la propia red social pero reabrieron sus cuentas con nuevos nombres, es por esto que no cumplen los requerimientos necesarios como para considerarse confiables. La selección de

esta lista de cuentas en el trabajo de investigación es expresada de manera neutral con los puntos de vista políticos, económicos y sociales. Estas cuentas hacen publicaciones periódicas compartiendo su punto de vista sobre la situación actual del país hacia sus seguidores, difundiendo de esta manera la desinformación. Cabe resaltar que, al momento de redactar este documento, Twitter bloqueó algunos de estos noticieros por contenido inapropiado o falso, dichas cuentas incluyen: @malditaternura, @jcd46 y @peru_memoria. Se espera que durante los siguientes meses otro grupo de estas cuentas seleccionadas también serán bloqueadas.

Tabla 3.2: Noticieros peruanos etiquetados como engañosos en el proyecto

	Noticiero	**Usuario Twitter**	**Antigüedad**
1	MalditaTernua	@malditaternura	2020
2	PolloFarsantePe	@PolloFarsantePe	2021
3	JusticieroPE	@justicierope	2021
4	el_analista2020	@el_analista2020	2020
5	Lord	@blackdragon1802	2010
6	Sinmermelada	@Sinmermelada3	2021
7	Derecha Unida	@derecha_Unida_	2022
8	el comentarista	@elcomentaristam	2021
9	Rafael Rey Rey	@reyreysincorona	2011
10	El Joker	@eljokerpe	2021
11	POLITICAL BEAR	@jcd46	2011
12	Noticias que no verás en Willax	@vekace82_laley	2022
13	Perú Memoria	@peru_memoria	2022
14	El Analista Fiu Fiu	@elanalista2022	2022
15	PikanteDKuy2	@PIKANTEDEKUY2	2022
16	Warner	@padre_n	2020
17	El Justiciero Rojo	@justicierojope	2012
18	Fachos Nunca Más	@AntifachosPE	2016

Por medio de la API de Twitter podemos extraer fácilmente las últimas publicaciones de las cuentas mencionadas anteriormente. De esta forma, utilizando la librería Pandas para un mejor manejo de los datos, se generan el fichero *dataset.csv* con noticias concatenadas y etiquetadas como verdaderas y falsas. Nuevamente, durante esta extracción de tweets se valida la cantidad mínima de palabras en cuarenta para reducir el error en la predicción de la veracidad de cada noticia, como se puede ver en el fragmento de Código 3.1.

```
1  def saveNewsCsv(noticieros, nroTweets, fileName, date_since):
2    news_list = []
3    for noticieroT in noticieros:
4      # Recover tweets from news
5      for status in tweepy.Cursor(api.user_timeline, screen_name=
           noticieroT, exclude_replies=True, include_rts=False,
           tweet_mode='extended').items(nroTweets):
6      # Only if it passes minimum required
7        if (len(status.full_text.split(" ")) > nro_min_palabras and
             fechaDentroRango(status.created_at, date_since)):
8          news_list.append([status.full_text, status.created_at,
             status.id])
9    # Convert array to DataFrame
10   news_list = pd.DataFrame(news_list, columns=['text', 'created_at',
       'id'])
```

```
11  news_list.to_csv(fileName)
12  files.view(fileName)
```

Código 3.1: Método que extrae noticias utilizando la API de Twitter.

Es así como, en la Tabla 3.3, se muestra un extracto de cuatro noticias al azar del *dataset* construido, se consideran cuatro columnas para los datos:

- **Etiqueta**: *Label* de las noticias, toma valores de *fake* o *true*.
- **Texto**: Noticia completa en bruto, tweet sin limpieza de ruido.
- **Creación**: Fecha de creación del tweet extraída por la API.
- **Id**: Id único del tweet extraído por la API.

Tabla 3.3: Extracto de tweets recuperados utilizando la API de Twitter

Etiqueta	**Texto**	**Creación**	**Id**
true	Conozca la confesión de Hugo Espino, al detalle su narración donde cuenta el papel de la primera dama y los viajes de Yenifer Paredes por diferentes distritos del Perú en busca de posibles licitaciones.	2022-08-21 22:50:00	1561485799329107968
true	Una banda de delincuentes asaltó a una mujer que acababa de recoger a su hijo del colegio en San Juan de Lurigancho. Al momento de la captura se les encontró un arma de fuego.	2022-08-18 02:26:37	1560090764083871744
fake	Este sujeto funge de periodista y se encarga de hacer el registro de todos los compañeros para que después sean atacados. Ojo con él. Hoy en el ataque de la Resistencia a la feria de niños llegó con el bloque fascista.	2022-05-15 04:30:50	1525695174856392704
fake	Aquí los mononeuronales con el brazo bien en alto. Después lloran diciendo que no son fascistas. En la foto el pelao Roger Ayachi, el loco Álvaro Subiria que hace unos días decía que una señora de 1.50 de estatura lo había agredido.	2021-10-18 04:55:47	1449962390536269830

Al finalizar este proceso, el *dataset* actual contiene más de cuatro mil textos, sin embargo no se tiene la misma cantidad de noticias falsas y verdaderas ya que los textos engañosos tienden a presentarse amplios, cumpliendo así en mayor cantidad con la condición del mínimo de palabras en esta red social. Es por ello que se hace una reducción de la cantidad

de textos falsos y adición de un mayor número de noticias reales hasta que el *dataset* presente aproximadamente 50 % para cada clase, de esta manera se realiza el etiquetado con mayor equidad. Para el control de versiones del proyecto se utilizó un repositorio de GitLab como fuente principal, donde se encuentra el *dataset* generado y código fuente de esta investigación. Dongo (2023).

Esto es suficiente como primera instancia, conforme se avance el proyecto se recopilarán nuevas noticias, las cuales pueden ser decisivas para modificar las predicciones del modelo. En capítulos siguientes se explica de manera más detallada cómo afectan estas noticias actuales a la precisión del modelo y de qué manera situaciones extremas, como los problemas bélicos, la crisis económica mundial o los cambios políticos a nivel nacional, afectan de manera negativa nuestra predicción.

3.1.3. Limpieza de ruido

A pesar de contener tweets con un mínimo de 40 palabras, se necesita validar que éstas sean útiles dentro del *dataset*, varias de ellas son ruido y no tienen valor predictivo para el modelo, como los emojis, hashtags, URLs u otro tipo de información de poca utilidad para la investigación, es por ello que se requiere limpiar el *dataset* de este tipo de impurezas para evitar incongruencias en la predicción y obtener resultados más confiables.

Esta limpieza se realiza utilizando expresiones regulares mostradas en la Tabla 3.4, las cuales no validan todos los casos, pero tienen la funcionalidad de eliminar gran parte del ruido dentro de cada noticia. La limpieza se realiza removiendo, en cada texto, los elementos coincidentes a las expresiones regulares en el mismo orden de la tabla, la librería *re* es importante para este procedimiento.

Tabla 3.4: Expresiones regulares para la limpieza del *dataset*

	Función	**Expresión Regular**
1	Usuarios	@[A-Za-z0-9]+
2	Correos	[A-Za-z0-9]+@[A-Za-z0-9À-ÿ]+
3	Hashtags	@[A-Za-z0-9]+
4	URLs	https?://\S+
5	Números y puntuación	[^a-zA-ZÀ-ÿ]

Tras limpiar los elementos coincidentes a las expresiones regulares, se valida que las palabras restantes no sean *stop words*, sino palabras diferenciadoras como sustantivos, adjetivos, verbos o adverbios, los artículos o preposiciones, se consideran *stop words* ya que no aportan información útil para las predicciones del modelo, se consideran ruido. La limpieza de estos *tokens* se realiza eficientemente al importar una lista de *stop words* en español utilizando la libreria NLTK.

Existe un detalle más para finalizar la limpieza del *dataset*, es necesario añadir palabras especiales a la lista de *stop words*, sin embargo esta parte será visible cuando se construyan los histogramas de palabras, donde será necesaria dicha modificación para la precisión de las predicciones. La descripción detallada de este tema se realizará en la Sección 3.2 dedicada a la implementación del modelo NB.

Otros puntos importantes a considerar en esta etapa son el análisis de la colección de

los datos y la extracción de características, sin embargo y dado que el algoritmo NB utiliza solo histogramas de palabras y se enfoca directamente en el texto, no requiere estos datos adicionales para hacer sus predicciones. Es por ello que serán considerados en la Sección 3.3 de la máquina de vector de soporte, ya que son muy importantes para el desarrollo y precisión de este modelo. Además en la SubSección 4.2.1 del siguiente capítulo se medirá la precisión de dichas características por medio de puntajes y *benchmarkings* de las *features* seleccionadas para el trabajo.

3.2. Desarrollo de modelo predictivo utilizando clasificador Naive Bayes

Como se mencionó en las bases teóricas del Capítulo 2, NB es un algoritmo de predicción ingenuo pero muy preciso, ya que calcula probabilidades de cada noticia en base a las probabilidades de cada palabra en referencia a todo el *dataset* utilizando el teorema de Bayes. La implementación del algoritmo NB, al igual que en los scripts anteriores para la construcción del *dataset* y la limpieza del mismo, se realiza en Python utilizando *Google Colab*, una plataforma muy útil para la implementación de modelos de *machine learning*, al otorgar acceso gratuito a sus máquinas virtuales *Python 3 Computing Engine Backend*, con las características suficientes para llevar a cabo el proyecto.

3.2.1. Histograma de palabras

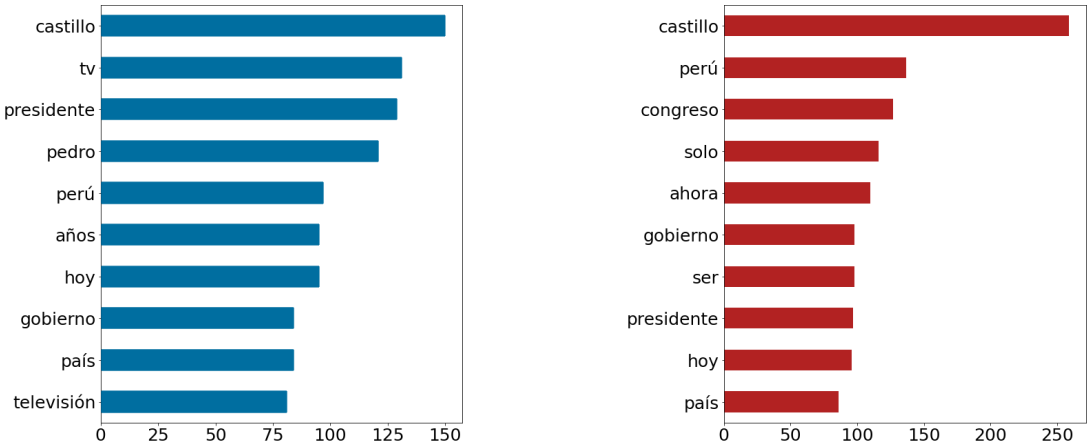
Se construyen dos histogramas de palabras utilizando el Código 3.2, uno generado a partir de la frecuencia de palabras dentro de los textos verdaderos y el otro dentro de los falsos, dichos histogramas serán utilizados para calcular la probabilidad de cada palabra referente a la cantidad total de palabras de cada noticia.

```
1  def histogram(hist_word, hist_count, hist_prob, nro_words, text):
2    text = text.split()
3    for word in text:
4      # Verify if exists, always initialize in 1
5      if word in hist_word:
6        hist_count[hist_word.index(str(word))] = hist_count[hist_word.
             index(str(word))] + 1
7      else:
8        hist_word.append(word)
9        hist_count.append(1)
10   return hist_word, hist_count, hist_prob
```

Código 3.2: Método para la construcción de los histogramas de palabras

La limpieza de ruido realizada en la Sección 3.1.3 elimina los *stop words* del dataset, sin embargo, existen palabras que no aportan a la predicción del modelo, tales como: nombres de los noticieros y monosílabos sin sentido, por esta razón es necesario adicionar estos *tokens* a la lista de *stop words* de la librería NLTK y realizar nuevamente la limpieza del *dataset*.

Tras realizar la nueva limpieza, en la Figura 3.2 se aprecia el top de palabras más utilizadas para cada clase de noticias. Aparentemente, las palabras que se muestran ahora son



(a) Respecto a clúster de noticias reales (b) Respecto a clúster de noticias falsas

Figura 3.2: Top 10 palabras más utilizadas por cada clase

de mayor utilidad para el clasificador NB, pero como se observa, existen múltiples *tokens* repetidos y utilizados comúnmente en ambos casos como: castillo, presidente, país, entre otros. Es por ello que en realidad los *tokens* diferenciadores no necesariamente son los que pertenecen al top de más utilizados, sino a los más característicos de cada clase.

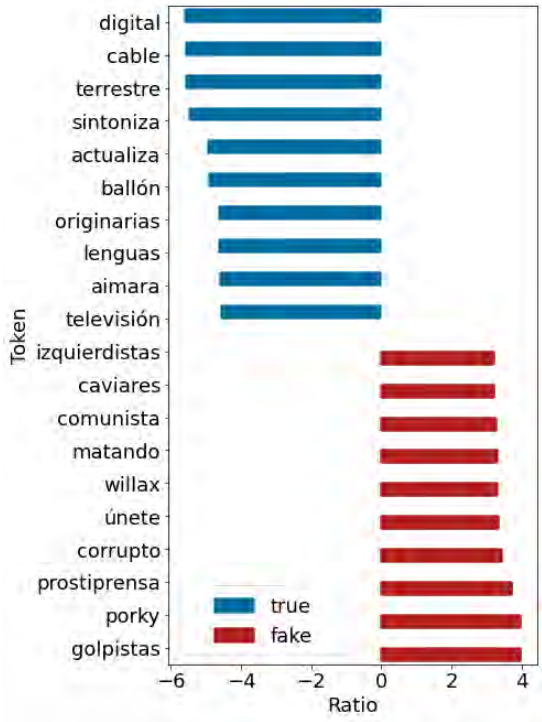


Figura 3.3: Top 10 palabras más diferenciadoras por cada clase

Es así como, en la Figura 3.3, se observa el top de palabras más características en cada tipo de noticias. Un valor más alto del *ratio* sugiere que una palabra es más característica o distintiva en el contexto de tweets falsos en comparación con los verdaderos, y viceversa. Es por ello que palabras como: digital, terrestre, cable, sintoniza, entre otras pertenecen a la clase de noticias verdaderas. Y palabras como: corruptos, golpistas, comunismo y demás, pertenecen a las falsas.

Con los histogramas construidos, es posible determinar la probabilidades individuales

Tabla 3.5: Ejemplo de funcionamiento para algoritmo NB

Texto					**Propor. V**	**Probab. V**
Las	personas	sufrieron	un	accidente		
	14.19×10^{-4}	0.424×10^{-4}		1.482×10^{-4}	0.4455	3.9723×10^{-12}
Texto					**Propor. F**	**Probab. F**
Las	personas	sufrieron	un	accidente		
	7.674×10^{-4}	0.357×10^{-4}		0.535×10^{-4}	0.5545	0.8127×10^{-12}

de cada palabra dentro de un nuevo texto, y de esta forma, las probabilidades del texto por cada clase, como se observa en la Tabla 3.5, donde se considera también la proporción de noticias para el cálculo, ya que la cantidad de noticias por clase no es la misma en el *dataset*. Para finalizar, se comparan ambas probabilidades, se determina la mayor y se etiqueta la noticia ingresada; es así como, en el ejemplo anterior, el texto sería etiquetado como verdadero. Esta probabilidad tiende a ser un número muy pequeño, es por ello que el algoritmo desarrollado recalcula la probabilidad resultante multiplicándola por 100 cada vez que sea menor a 0.01. Luego se almacena la cantidad de decimales en una variable usada posteriormente y, de esta manera, se determina la probabilidad mayor basada en potencias de 100, como se muestra en la regla condicional:

$$f(x) = \begin{cases} \text{si } x < 0{,}01 & x * 100 \wedge decimales+ = 1 \\ \text{si } x \geq 0{,}01 & x \end{cases} \tag{3.1}$$

3.3. Desarrollo de modelo predictivo utilizando Support Vector Machine

A pesar de la clara simplicidad pero alto potencial que muestra la técnica de NB para la predicción de la veracidad de los textos, un algoritmo como SVM puede ser una gran alternativa ampliando el enfoque que se tiene para solucionar este problema.

3.3.1. Análisis del contenido de las colecciones de datos

Si bien para este punto se tienen los datos del *dataset* libres de ruido, es necesario llevar a cabo un análisis de las particularidades de cada colección para entenderlas de mejor manera y comprender el tipo de información que almacenan. Este análisis se debe realizar sobre los datos en bruto, sin modificaciones, ya que alteran, en muchos casos, los resultados esperados. De manera similar que el análisis realizado por Espejel et al. (2022), se estudian los siguientes aspectos:

Primero, la longitud o número de *tokens* de las noticias por cada clase, para ello se estandariza todos los textos en minúscula, sin signos de puntuación pero preservando las *stop words*. Los resultados se muestran en la Figura 3.4. Si bien todos los textos tienen como mínimo 40 palabras, se observa que las noticias veraces, representadas por la línea azul, tienden a posicionarse en su mayoría entre las 40 y 50 palabras, mientras que las falsas, representadas por la línea de color rojo, pueden llegar a contener entre 40 y 60 palabras, este dato es curioso, ya que en longitudes menores a 40 los resultados son inversos.

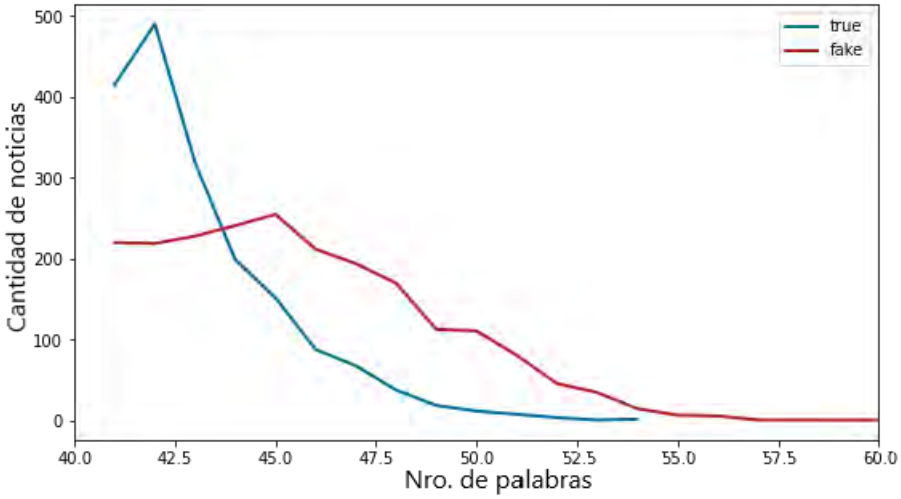


Figura 3.4: Distribución de noticias por número de palabras

Segundo, la proporción de *stop words* en las colecciones de datos, aspecto distinto a la cantidad de *stop words* utilizada en Espejel et al. (2022), ya que en el presente trabajo se considera más confiable la proporción porque varía de acuerdo a la longitud de cada noticia, la cual se obtuvo en el anterior aspecto. La Figura 3.5 presenta la proporción de estas palabras respecto a la longitud de *tokens* de cada noticia. Como se puede apreciar, la línea azul de las noticias verdaderas tiene uso y cantidades de *stop words* similares a la roja que representa a las *fake news*, se puede observar también, que la mayoría de noticias tienen valores ubicados entre 0.4 y 0.6 en promedio, lo que demuestra que para redactar una noticia se utilizan entre 40 % y 60 % de estas palabras.

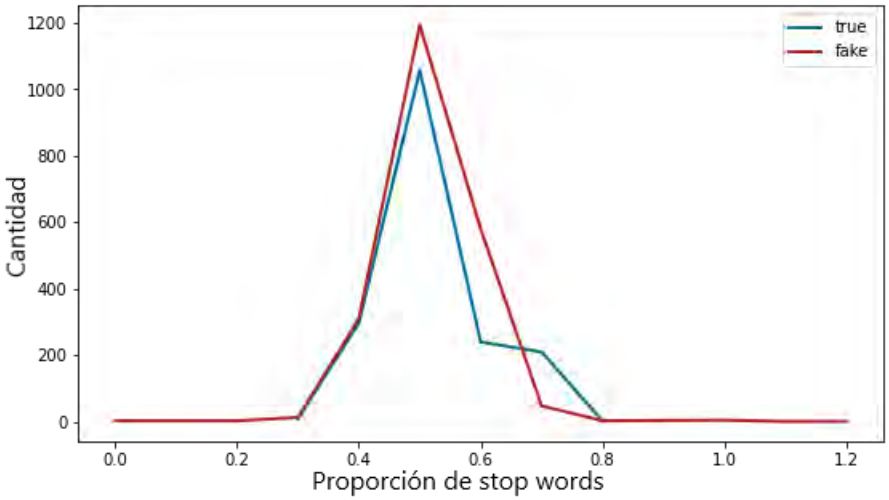


Figura 3.5: Distribución de noticias por proporción de *stop words*

Tercero, los términos más frecuentes por cada etiqueta, para lo cual son necesarios los histogramas de palabras construidos en la Subsección 3.2.1 del modelo *Naive Bayes* anteriormente descrito, donde se estudia a mayor profundidad la limpieza de los *tokens* sin valor semántico y los grupos de palabras más diferenciadoras de cada clase.

La Figura 3.6 muestra las nubes de palabras generadas utilizando la librería de Python *WordCloud*, ambos gráficos fueron producidos con la configuración: *max-words* = 40, *min_word_length* = 3 y *prefer_horizontal* = 0.5. La Figura 3.6a, de la izquierda, muestra una nube de palabras conteniendo las 40 palabras más utilizadas en las noticias clasificadas

como verdaderas, mientras que en la Figura 3.6b, de lado contrario, se observan las 40 palabras más utilizadas para nuestra colección de noticias falsas. Donde las palabras de mayor tamaño representan alta frecuencia y las de menor, baja frecuencia. Si comparamos ambos gráficos podemos notar algunas diferencias visuales, como el tamaño de las palabras, si interpretamos este patrón se concluye que las noticias falsas abusan, en su mayoría, del mismo grupo de palabras, mientras que las verdaderas utilizan términos más variados.

(a) Respecto a clúster de noticias reales

(b) Respecto a clúster de noticias falsas

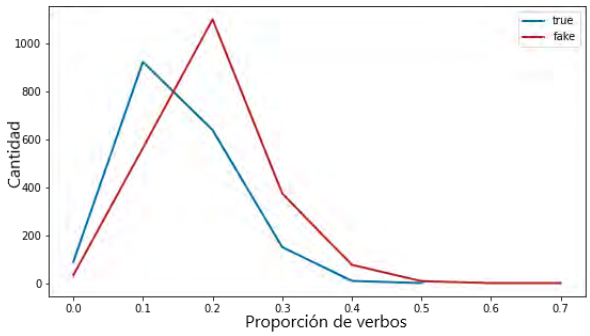
Figura 3.6: *Word Clouds* de las colecciones de datos

Finalmente, la frecuencia de elementos gramaticales como: sustantivos, verbos y adjetivos, los cuales indican el estilo de redacción de cada instancia. Para esta tarea se utiliza la librería *PySentimiento*, que tiene muchas capacidades de Natural Language Processing (NLP), las cuales incluyen análisis sentimental, emocional y gramatical. La Figura 3.7 muestra gráficos de proporciones en función a la longitud de cada noticia para los elementos gramaticales mencionados, los valores obtenidos fueron redondeados para mostrar un gráfico más limpio y claro. Para los tres casos se consideran las líneas de color azul y rojo como la cantidad de noticias verdaderas y falsas por proporción de los elementos gramaticales respectivamente. Se observa en todos los gráficos que la diferencia entre clases es mínima, por lo cual no se considera esta característica como diferenciadora.

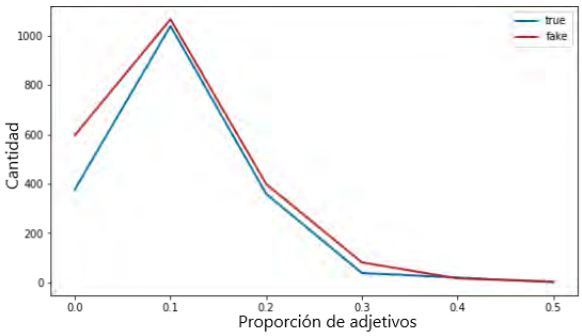
Además, se llevó a cabo un análisis de correlación de palabras para determinar la relación entre ambas clases, como se muestra en la Figura 3.8, existe una correlación positiva, el valor del eje Y (true) se incrementa ligeramente a medida que aumenta el valor del eje X (fake), observamos, además, algunas palabras que pertenecen claramente a cada clase. Se descubrió también que la cantidad de palabras únicas utilizadas en las *fake news* es de 12445, mientras que en las verdaderas se contaron 9305 palabras, de las cuales 4663 son comunes entre ambas clases, representando el 37.47 % de las noticias falsas y 50.11 % de las verdaderas.

3.3.2. Extracción de características

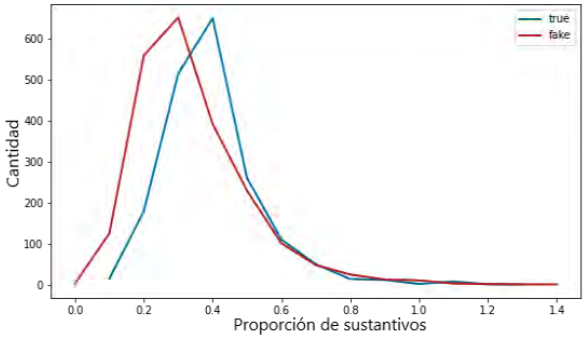
La extracción de características es un proceso crítico para la construcción de un DS de alta calidad. Es una etapa crucial en el desarrollo del modelo y es determinante para la precisión en la predicción final. Sin una extracción precisa de características, es difícil lograr resultados precisos y confiables en el modelo. Como mencionan Xue et al. (2021), la principal



(a) Por proporción de verbos



(b) Por proporción de adjetivos



(c) Por proporción de sustantivos

Figura 3.7: Cantidad de noticias por proporción POS

dificultad en la detección de noticias falsas consiste en encontrar las características que las hacen diferentes de las verdaderas.

Aunque para esta etapa se dispone de datos limpios, también es útil tener acceso a las colecciones previas a la limpieza. Como se explicó en la sección anterior, las palabras vacías o *stop words* no proporcionan un valor semántico significativo, pero son esenciales en la comunicación natural humana. Por lo tanto, como se menciona en la investigación de Espejel et al. (2022), su eliminación no es siempre recomendable en tareas de NLP. Las características consideradas en el proyecto se agrupan en:

Análisis sentimental

Una de las características más importantes que se considera en el proyecto es el análisis sentimental. Cada texto es calificado y puntuado como positivo, negativo o neutral. Para ello se utilizan las librerías externas SAS y PySentimiento. Sin embargo, al hacer pruebas con ambas librerías, se demostró que la primera de ellas no reconoce el contexto dado por los signos de puntuación como se muestra los ejemplos de la Tabla 3.6:

Tabla 3.6: Comparación de característica *Positive* generada por librerías utilizadas

Texto	**SAS**	**PySentimiento**
Yo no, estoy muy feliz en mi trabajo.	0.0097	0.9832
Yo no estoy muy feliz en mi trabajo.	0.0097	0.0017
No! estoy triste porque los tengo cerca!	0.0020	0.0014
No estoy triste, porque los tengo cerca.	0.0020	0.9706

Se observa que la librería SAS no cuenta con la capacidad de diferenciar el contexto

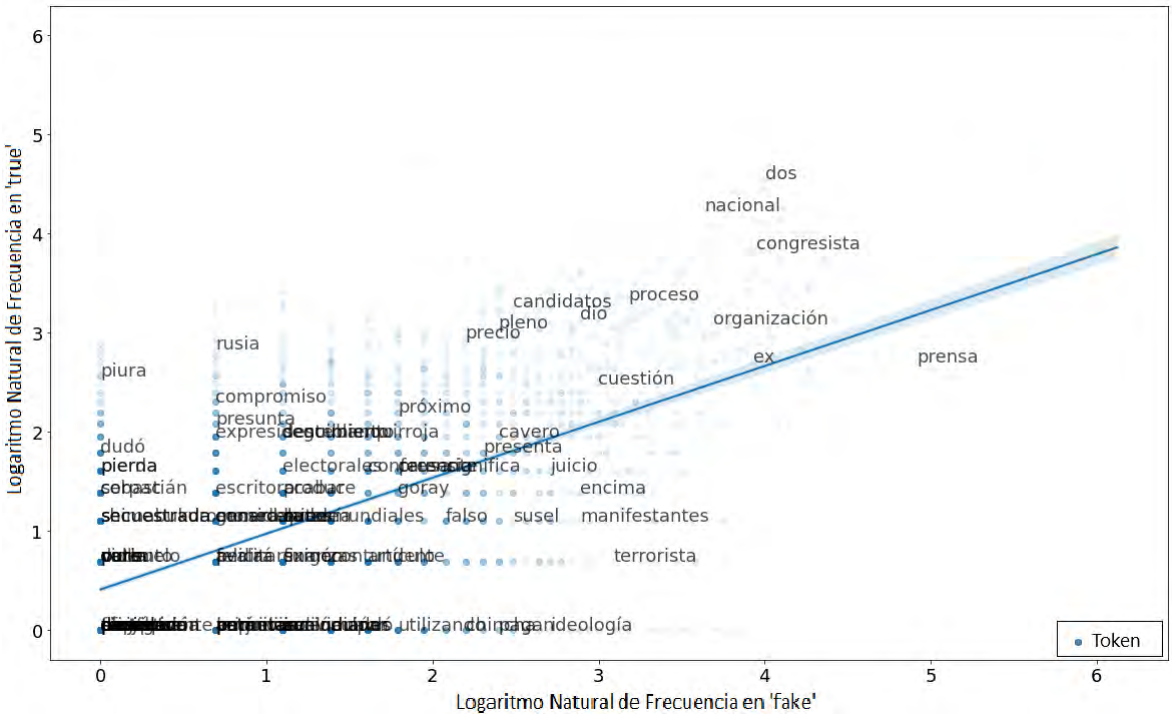


Figura 3.8: Gráfico de correlación de palabras

de un texto basado en sus signos de puntuación, a diferencia de PySentimiento, la cual es capaz de identificar de manera clara que ambos pares de oraciones son distintos. Por esta razón, se optó por utilizar esta última librería, la cual genera tres características: *Positive*, *Negative* y *Neutral* para esta subsección, las cuales tienen valores comprendidos entre 0 y 1, expresando así su puntaje de positividad, negatividad o neutralidad respectivamente.

La importancia de este análisis se basa en que las noticias verdaderas tienden a tener un sentimiento en su mayoría neutral, mientras que las noticias falsas tienden a ser negativas en su mayoría, generando de esta manera una separación entre los clústeres; sin embargo en la práctica esto no se da completamente, las noticias verdaderas también tienen valores de sentimiento negativo y las *fake news* pueden no expresar sentimientos y mostrarse neutrales.

Análisis emocional

De manera similar al análisis sentimental, esta característica es útil para calificar los textos otorgándoles puntajes respecto a emociones como ira, alegría, tristeza, entre otros. Para ello se utiliza nuevamente la librería *PySentimiento* que trabaja con el *dataset EmoEvent* en la publicación titulada *Proceedings of the 12th Language Resources and Evaluation Conference* del Arco et al. (2020), esta librería abarca un gran número de emociones, sin embargo para el proyecto solo se utilizan las más resaltantes como son: *Fear* (Miedo), *Surprise* (Sorpresa), *Sadness* (Tristeza), *Anger* (Ira), *Joy* (Alegría) y *Disgust* (Disgusto), estas *features* pueden tener valores entre 0 y 1, siendo 1 una mayor expresión de la respectiva emoción en un texto y 0 la inexistencia de la misma.

Análisis de odio

Este análisis se realizará también utilizando la librería *PySentimiento*, se extraerán dos características claves para el proyecto: *Hateful* y *Aggressive* (Odio y Agresividad). Estas son consideradas cruciales porque han demostrado ser diferenciadoras. Si bien ambos grupos de noticias pueden presentar sentimientos negativos o neutrales y emociones similares, las noticias falsas tienden a mostrar más odio y agresividad que las verdaderas. Como en el análisis anterior, estas características serán valoradas con puntajes entre 0 y 1.

Non dictionary words

Para esta característica se utilizó la librería externa de Python *autocorrect*, es así como se recorre cada noticia, verificando si sus *tokens* o palabras existen en el diccionario de la lengua española. El puntaje final tiene valores entre 0 y 1, siendo 0 un texto conformado por el 100 % de palabras existentes en el diccionario y 1 un texto conformado en su totalidad por palabras inexistentes en el diccionario.

Se considera que las noticias falsas tienden a demostrar generalmente informalidad, por ello presentan errores gramaticales y utilizan palabras inexistentes en el diccionario, mientras que las noticias verdaderas tienen a utilizar un lenguaje más culto y formal. Es por ello que esta característica será resultado de una proporción entre palabras inexistentes en el diccionario y el total de palabras de cada noticia.

Después de realizar los análisis respectivos, se encontró que los clústeres generados en 2 dimensiones (considerando pares de características) se sobreponen y no son diferenciadores. Esto será descrito de manera más detallada en el siguiente capítulo. Con el resto de características obtenidas, como se muestra en la Tabla 3.7, y el análisis exploratorio de los datos de las subsecciones anteriores, se tiene un mayor conocimiento del problema, el cual es utilizado para llevar a cabo el desarrollo del SVM. Dicha implementación se basa en el blog de Mathieu Blondel, quien incorpora *kernels*, *soft margin* y programación cuadrática utilizando la librería Python Software for Convex Optimization (CVXOPT). Blondel (2010). En dicha implementación se definen los kernels: lineal, gausiano y polinomial para el funcionamiento del modelo, cuyos resultados se describen en el siguiente capítulo.

Tabla 3.7: Extracto de *dataset* al analizar cuatro características

Etiqueta	**Texto**	**Odio**	**Agresividad**	**Alegría**	**Negatividad**
true	Una banda de delincuentes asaltó a una mujer que acababa de recoger a su hijo del colegio en San Juan de Lurigancho. Al momento de la captura se les encontró un arma de fuego.	0.04861	0.02117	0.00798	0.39645
true	Conoce quienes son los dos miembros de seguridad señalados por Bruno Pacheco como los encargados de cobrar las coimas en los ascensos policiales, como operaban, qué papel jugaban y como era la estructura de esta organización	0.01131	0.01258	0.00248	0.07071
true	La vida de lujos de Tyson, uno de los capos de la droga en Perú que operaba desde una exclusiva residencia en las Casuarinas. Seguimientos, capturas, toneladas de droga incautadas y su conexión con poderosas mafias de narcotráfico en Europa.	0.02570	0.02042	0.02216	0.01884
fake	Acaban de llamar por teléfono a mi esposa y parece que va a recuperar una importante consultoría en uno de los principales ministerios. Estamos todos muy felices en casa, pasaremos unas lindas fiestas.	0.02080	0.01400	0.96010	0.00590
fake	Ni bien se fue la OEA los representantes de los partidos de oposición se reúnen en secreto para planificar una suspensión al presidente. No les basta su denuncia por traición a la patria rechazada por el TC, y ahora buscan una segunda oportunidad.	0.01716	0.01812	0.00205	0.31631

Capítulo 4

Benchmarking y pruebas de rendimiento

4.1. Naive Bayes

Al momento de redactar este documento, el *dataset* utilizado está actualizado al 8 de enero de 2023 con un total de 4326 noticias, divididas en porcentajes similares para cada clase. Para los tests realizados, se considera un 80 % del *dataset* para *training* y 20 % para *validation*, porcentajes que obtienen los resultados con mayor precisión y que fueron determinados por prueba y error durante la investigación.

4.1.1. Matriz de confusión

Para determinar la precisión del modelo desarrollado se utiliza el *validation dataset*. Se determina la veracidad de cada texto de validación utilizando los histogramas construidos y se obtiene la clase más probable a la que pertenece cada noticia según la predicción del modelo. Finalmente, se compara la etiqueta predicha frente a la real, se construye la matriz de confusión de acuerdo a la Tabla 2.2 y se determina su *accuracy* según la ecuación (2.1).

Tabla 4.1: Matriz de confusión algoritmo Naive Bayes

		Predicción	
		Verdadero	Falso
Etiqueta real	Verdadero	322	17
	Falso	18	404

La Tabla 4.1 muestra los resultados del test de precisión representados por medio de la matriz de confusión, donde se observa lo siguiente:

- Las noticias verdaderas etiquetadas correctamente como verdaderas (VV) son 322 representando el 94.99 % de las noticias verdaderas
- Las noticias falsas etiquetadas correctamente como falsas (FF) son 404 representando un 95.73 % de las noticias falsas

- Las noticias verdaderas etiquetadas erróneamente como falsas (VF) son 17 representando un 5.01 % de las noticias verdaderas
- Las noticias falsas etiquetas erróneamente como verdaderas(FV) son 18 representando un 4.27 % de las noticias falsas

Aplicando la fórmula de precisión, el modelo de predicción basado en NB con un *dataset* de cuatro mil noticias dividido 80 % para *training* y 20 % *validation* tiene una precisión de 95.4 %.

Si bien los resultados demuestran una alta precisión del algoritmo en este tipo de tareas, existen detalles por los cuales la predicción puede ser errónea como la sintaxis o palabras repetidas, estos puntos serán descritos a mayor detalle en el Capitulo 7.

4.2. Support Vector Machine

Los resultados expuestos en la sección anterior demuestran que la técnica NB tiene alto potencial y precisión en la detección de noticias falsas, el análisis realizado para el algoritmo NB es simple ya que no presenta múltiples parámetros a comparación del SVM, el cual puede obtener múltiples resultados en base a la configuración de dichos parámetros.

Aunque mostrar los resultados obtenidos al realizar el *benchmarking* del SVM utilizando pares de características, variando la proporción de datos de entrenamiento, los kernels utilizados y diferentes valores para la constante C podría ser información interesante, en este proyecto se considera que no es relevante, ya que el problema es demasiado complejo para resolverlo con conjuntos reducidos de características. Por lo tanto, las pruebas siguientes se basan en algunas características diferenciadoras mencionadas, mientras que las pruebas posteriores utilizarán todas las características obtenidas en el análisis previo de sentimiento y emoción. De esta forma, se determinará el mejor hiperplano para la tarea en cuestión.

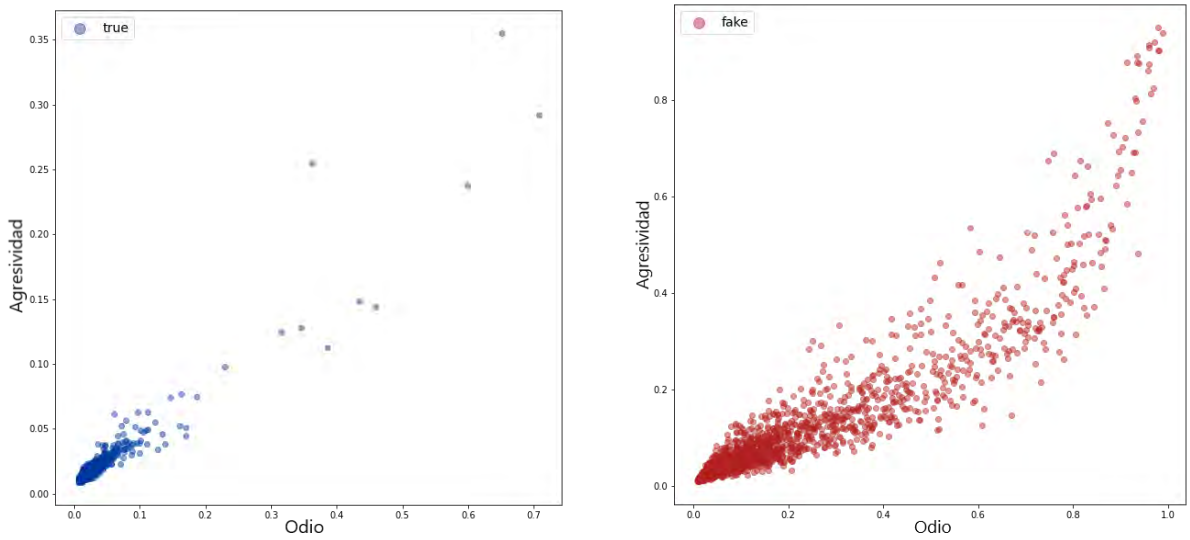
4.2.1. Para características en duplas

Para su correcto funcionamiento es necesario tener características muy diferenciadoras en el *dataset*, como se mencionó anteriormente, cada característica genera una dimensión. En esta subsección se trabaja con un hiperplano lineal, ya que el espacio tiene 2 dimensiones y, por lo tanto, 2 características diferenciadoras. Como se mencionó en la Sección 3.3.2 dedicada al preprocesamiento de los datos, las características seleccionadas para el proyecto son las siguientes: *Hateful* (Odio), *Agressive* (Agresividad), *Fear* (Miedo), *Surprise* (Sorpresa), *Sadness* (Tristeza), *Anger* (Ira), *Joy* (Alegría), *Disgust* (Disgusto), *Positive* (Positividad), *Negative* (Negatividad) y *Neutral* (Neutralidad).

La detección de noticias falsas es un problema complejo que no puede resolverse solo con un par de características. Sin embargo, analizando gráficos como los mostrados en las Figuras 4.1, 4.2 y 4.3, se puede obtener información relevante. Estos gráficos muestran que las noticias verdaderas obtienen puntajes menores en características como *Odio* y *Agresividad*, mientras que las *fake news* incitan al odio y la agresión en mayor proporción. Además, las noticias falsas obtienen puntajes altos en la característica *Ira* pero bajos en *Miedo*, en

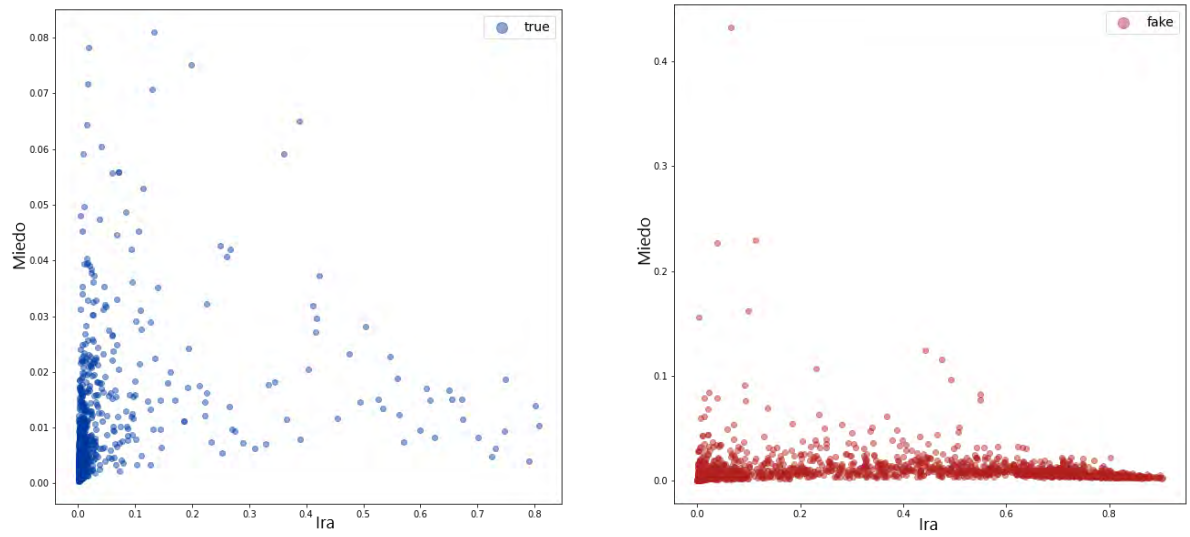
comparación con las noticias reales. Por último, se observa que las *fake news* muestran un mayor nivel de *Disgusto* que las noticias verdaderas.

Aunque se han obtenido buenos resultados de precisión e interpretación usando algunas de las características obtenidas, es importante utilizar un espacio de N dimensiones para mejorar la predicción del modelo.



(a) Clúster para noticias verdaderas (b) Clúster para noticias falsas

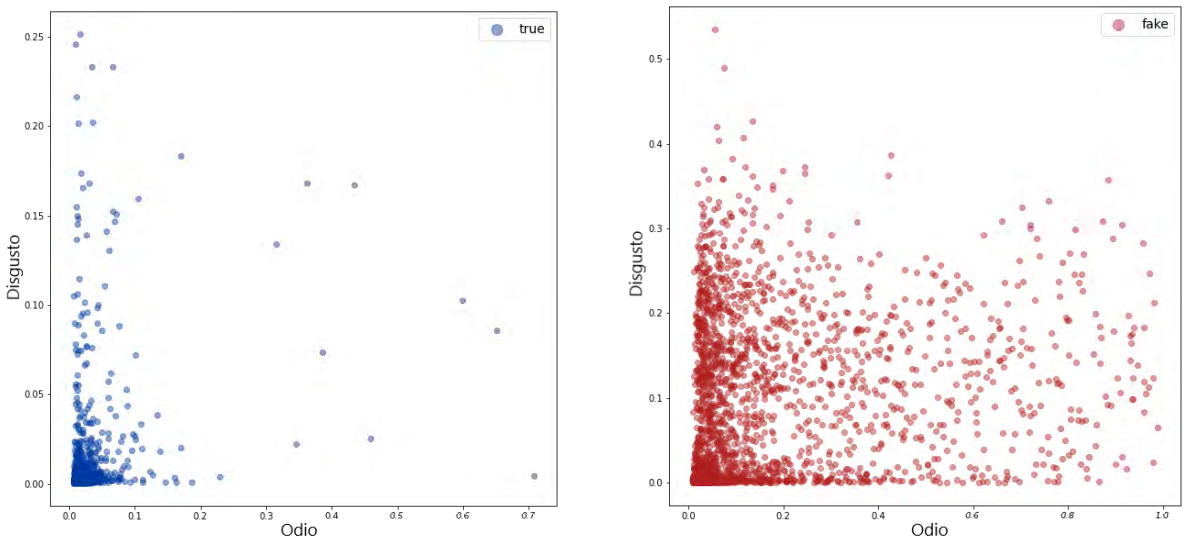
Figura 4.1: Clústeres de noticias respecto a dupla *Odio-Agresividad*



(a) Clúster para noticias verdaderas (b) Clúster para noticias falsas

Figura 4.2: Clústeres de noticias respecto a dupla *Ira-Miedo*

Al finalizar las pruebas de precisión utilizando el kernel lineal, la dupla que obtuvo el mayor puntaje de precisión en promedio para los valores de la constante C frente a las otras fue *Odio-Disgusto*, con un puntaje de precisión de 86.68 %.



(a) Clúster para noticias verdaderas (b) Clúster para noticias falsas

Figura 4.3: Clústeres de noticias respecto a dupla *Odio-Disgusto*

Matriz de confusión

La matriz de confusión obtenida para el algoritmo SVM sobre la dupla de características *Odio-Disgusto* se muestra a continuación:

Tabla 4.2: Matriz de confusión algoritmo SVM para dupla *Odio-Disgusto*

		Predicción	
		Verdadero	Falso
Etiqueta real	Verdadero	357	62
	Falso	30	347

La Tabla 4.2 muestra los resultados del test de precisión representados por medio de una matriz de confusión, donde se observa lo siguiente:

- Las noticias verdaderas etiquetadas correctamente como verdaderas (VV) son 357 representando el 85.2 % de las noticias verdaderas.
- Las noticias falsas etiquetadas correctamente como falsas (FF) son 347 representando un 92.04 % de las noticias falsas.
- Las noticias verdaderas etiquetadas erróneamente como falsas (VF) son 62 representando un 14.8 % de las noticias verdaderas.
- Las noticias falsas etiquetas erróneamente como verdaderas(FV) son 30 representando un 7.96 % de las noticias falsas.

Aplicando la fórmula de precisión, el modelo de predicción basado en SVM para la dupla de características *Odio-Disgusto* con un *dataset* dividido 80 % para *training* y 20 % *validation* tiene una precisión de 88.44 %.

4.2.2. Para características en general

Como se explicó anteriormente, son necesarias las pruebas de precisión utilizando todas las características. Para ello se consideran parámetros como la proporción de *training dataset* y *testing dataset*, los *kernels* lineal, gaussiano y polinomial y la constante C con valores de 1, 10 y 100.

Tabla 4.3: *Benchmarking* de características en general para SVM dividiendo *dataset* 70/30

C	Lineal	Gaussiano	Polinomial
1	0.8727	0.8325	0.8903
10	0.8835	0.8668	0.9037
100	0.8953	0.8794	0.9070

La Tabla 4.3 muestra la precisión del modelo dividiendo el *dataset* en 70 % para entrenamiento y 30 % para *testing* con valores de 1, 10 y 100 para la constante C, considerando los *kernels* lineal, gaussiano y polinomial. Con los resultados obtenidos, la configuración más óptima es representada por el *kernel trick* polinomial, específicamente cuando la constante C toma el valor de 100, dando un porcentaje de precisión de 90.7 %.

Tabla 4.4: *Benchmarking* de características en general para SVM dividiendo *dataset* 80/20

C	Lineal	Gaussiano	Polinomial
1	0.8756	0.8417	0.8932
10	0.8844	0.8744	0.9058
100	0.8945	0.8781	0.9045

De la misma forma, la Tabla 4.4 muestra la precisión del modelo con una proporción de 80 % del *dataset* para entrenamiento y 20 % para *testing* utilizando valores de 1, 10 y 100 para la constante C, y los 3 *kernel tricks* mencionados en el marco teórico. Se observa que las configuraciones óptimas son, nuevamente, las que utilizan el kernel polinomial, específicamente cuando la constante C toma el valor de 10 y 100, dando porcentajes de precisión de 90.58 % y 90.45 % respectivamente.

Tabla 4.5: *Benchmarking* de características en general para SVM dividiendo *dataset* 90/10

C	Lineal	Gaussiano	Polinomial
1	0.8819	0.8643	0.8944
10	0.8869	0.8894	0.9070
100	0.8894	0.8869	0.8995

Para finalizar con las pruebas de precisión, la Tabla 4.5 que se obtiene, muestra los valores predictivos del modelo dividiendo el *dataset* en un 90 % para entrenamiento y 10 % para *testing* con la constante C tomando valores de 1, 10 y 100 y utilizando los *kernels* lineal, gaussiano y polinomial mencionados en las pruebas anteriores. Se puede apreciar que las configuraciones más óptimas son las que utilizan el kernel polinomial, específicamente cuando la constante C obtiene el valor de 10, con un porcentaje de acierto del 90.7 %.

Para concluir, la configuración de parámetros que obtuvo el mayor puntaje en las tres pruebas utilizando el modelo SVM y todas las características fue para un kernel polinomial con una constante C de 10 o 100, obteniendo un porcentaje de precisión de hasta 90.7 %. La

proporción de división del *dataset* no se muestra relevante, sin embargo, para pruebas futuras se utilizará el 80/20 para entrenamiento y pruebas respectivamente por ser la proporción intermedia del Benchmarking Method (BM). Estos parámetros son utilizados en el modelo híbrido implementado en la Sección 5.1.

Matriz de confusión

La matriz de confusión generada para el algoritmo de SVM con los parámetros óptimos se muestra en la Tabla 4.6 descrita a continuación:

Tabla 4.6: Matriz de confusión de SVM utilizando configuración óptima

		Predicción	
		Verdadero	Falso
Etiqueta real	Verdadero	369	50
	Falso	25	352

Donde se observa lo siguiente:

- Las noticias verdaderas etiquetadas correctamente como verdaderas (VV) son 369 representando el 88.07 % de las noticias verdaderas.
- Las noticias falsas etiquetadas correctamente como falsas (FF) son 352 representando un 93.37 % de las noticias falsas.
- Las noticias verdaderas etiquetadas erróneamente como falsas (VF) son 50 representando un 11.93 % de las noticias verdaderas.
- Las noticias falsas etiquetas erróneamente como verdaderas(FV) son 25 representando un 6.63 % de las noticias falsas.

Aplicando la fórmula de precisión, el modelo de predicción basado en SVM para el grupo de características en general con la configuración óptima de kernel polinomial con un *dataset* dividido 80 % para *training* y 20 % *validation* tiene una precisión de 90.70 %.

A comparación del modelo anterior, los resultados utilizando la SVM en el proyecto presentan una menor precisión, esto puede deberse a varias razones, como el tamaño del *dataset*, la limpieza correcta del ruido de los datos, la fiabilidad de las librerías externas utilizadas para la extracción de características y la sintaxis de los textos de prueba, aspectos que deberán ser considerados en trabajos futuros. Sin embargo, la técnica NB presenta también carencias en predicciones específicas, es por ello que se plantea implementar un modelo híbrido que tenga el conocimiento de ambos algoritmos, la sólida capacidad léxica del NB y la potencia semántica y sintáctica del SVM.

Es así como se adiciona una nueva características a las descritas anteriormente para nuestro SVM, la cual es calculada utilizando el algoritmo de NB. En la siguiente sección se explica esta propuesta con mayor detalle.

Capítulo 5

Modelo híbrido

5.1. Mejora de modelo predictivo utilizando algoritmo híbrido

La generación de clústeres lo suficientemente diferenciadores con datos reales es un desafío complejo. Como se ha mencionado anteriormente, el uso de algoritmos de clasificación, como la SVM, depende en gran medida de las características extraídas del *dataset*. Sin embargo, la precisión de estas características es afectada por el rendimiento de las librerías externas utilizadas para el análisis sentimental, emocional y gramatical. Además, se ha observado que las características obtenidas pueden presentar valores similares, lo que dificulta la separación de los clústeres y la determinación de los hiperplanos adecuados para realizar una clasificación precisa.

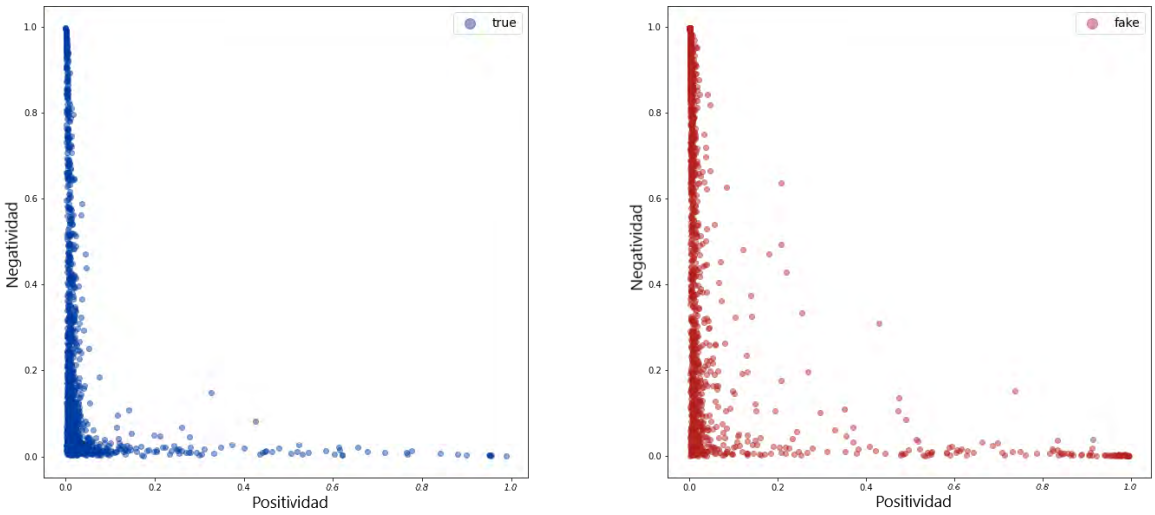
En complemento al capítulo anterior, donde se identificaron los clústeres que obtienen la mayor precisión según el BM, si se analiza la Figura 5.1 se puede observar la distribución de los clústeres en relación al análisis sentimental, en donde, tanto las noticias verdaderas como las falsas son principalmente negativas, y solo un pequeño porcentaje positivas. Esta dupla no es muy relevante para diferenciar ambas clases. Como se mencionó en la Sección 4.2 del capítulo anterior, si se considera el análisis de odio, se puede ver que las noticias falsas tienden a mostrar agresividad en mayor medida que las noticias verdaderas. Esta información es útil para diferenciar ambos tipos, pero no es suficiente para obtener resultados precisos.

Por otro lado tenemos el modelo estadístico NB, el cual ha demostrado tener porcentajes de precisión muy altos con respecto al SVM, utilizando una idea simple pero efectiva, un histograma de palabras, determinando la etiqueta de cada noticia por probabilidades de palabras dentro de cada texto.

Tabla 5.1: Complejidad sintáctica en el análisis de Naive Bayes

Texto	**Tokens**
O1. El criminal no sobrevivió al incendio.	[‘criminal’, ‘sobrevivió’, ‘incendio’]
O2. No, el criminal sobrevivió al incendio.	[‘criminal’, ‘sobrevivió’, ‘incendio’]
O3. Al incendio el sobrevivió criminal no.	[‘incendio’, ‘sobrevivió’, ‘criminal’]

Si bien el modelo NB funciona muy bien y tiene alta precisión para esta tarea, presenta problemas semánticos, ya que se enfoca en las palabras como entes cuantitativos sin



(a) Clúster para noticias verdaderas (b) Clúster para noticias falsas

Figura 5.1: Similitud de clústeres respecto a dupla *Positividad-Negatividad*

significado, esto puede originar errores de predicción para textos con sintaxis errónea pero palabras diferenciadoras, un caso similar ocurre en textos con palabras repetidas. En la Tabla 5.1, se observa que el algoritmo NB presenta complicaciones sintácticas al analizar textos. A pesar de ser altamente eficiente, este algoritmo puede devolver resultados similares para textos que contienen significados distintos o incoherentes. En el ejemplo mostrado, las oraciones **O1** y **O2** tienen sentidos distintos, mientras que **O3** carece de ello. Sin embargo, al ser tokenizadas, todas obtienen los mismos resultados, de esta forma, cada token obtenido tendría una probabilidad de 1/3, lo cual genera resultados ambiguos. Por otro lado, el modelo SVM, tiene mejor manejo de la sintaxis y la semántica ya que los análisis sentimental y emocional consideran el contexto dentro de cada texto, como se muestra en la Tabla 3.6 anteriormente descrita. De esta manera ambos algoritmos se complementan.

5.1.1. Creación de característica NB

Después de observar los resultados de precisión en la detección de noticias falsas utilizando los modelos NB y SVM se vio necesario realizar cambios en las *features* del modelo, las características recuperadas con las librerías de análisis sentimental requieren de una adicional, un puntuador de NB, que retorne la probabilidad mayor al ser clasificado como verdadero o falso, dicha probabilidad tendrá signo positivo para el primer caso y negativo para el segundo, es por ello que desde ahora, esta nueva característica será denominada como *SignoNB*, el cálculo de este valor se describe a continuación:

Al evaluar cada noticia, se puede obtener un puntaje normalizado utilizando los histogramas de NB, este puntaje varía de acuerdo a cómo fue etiquetada una noticia, como se muestra en el Algoritmo 1, pseudocódigo de una función llamada *prediccionSignoNB()*, la cual tiene cuatro parámetros: *nro_decimales_T*, *nro_decimales_F*, *prob_text_T*, *prob_text_F*. Estos parámetros son obtenidos por el algoritmo NB e indican la cantidad de decimales por probabilidad verdadera y falsa que tiene una noticia.

Es así como en el pseudocódigo se compara si el número de decimales de T es mayor que el de F, si es así se asigna a la variable valor_signo el valor de prob_text_F multiplicado

Algorithm 1 Generación de nueva característica utilizando los puntajes de Naive Bayes

```
1: procedure PREDICCIONSIGNONB(nro_decimales_T, nro_decimales_F, prob_text_T,
   prob_text_F)
2:     if nro_decimales_T > nro_decimales_F then
3:         valor_signo ← prob_text_F*-1
4:     else if nro_decimales_T < nro_decimales_F then
5:         valor_signo ← prob_text_T
6:     else
7:         if prob_text_T > prob_text_F then
8:             valor_signo ← prob_text_T
9:         else
10:            valor_signo ← prob_text_F*-1
11:        end if
12:    end ifreturn valor_signo
13: end procedure
```

por -1, ya que al tener mayor número de decimales hace que la probabilidad sea menor. Si el número de decimales de T es menor que el de F, se asigna a la variable valor_signo el valor de prob_text_T. Si el número de decimales de T es igual al de F, se compara si el valor de prob_text_T es mayor que el de prob_text_F, si es así se asigna a la variable valor_signo el valor de prob_text_T, en caso contrario, se asigna el valor de prob_text_F multiplicado por -1. El resultado final es la variable valor_signo, que contiene el valor de la nueva característica.

5.1.2. Benchmarking de características en general

A diferencia de los BM anteriores, en esta sección no se realizan las pruebas de duplas con la nueva característica, porque los resultados de pares obtenidos no se consideran útiles. Si bien los resultados de precisión de las duplas son altos, no se vio conveniente darle importancia a estos valores, ya que en realidad es equivalente a tomar la característica de NB y relacionarla una a una con las otras, desaprovechando realmente el conocimiento de la SVM. Como se muestra en la Figura 5.2, al relacionar la característica *Disgust* con SignoNB se obtienen dos clústeres separables visualmente, esto vuelve irrelevante el resto de propiedades. Es por ello que las pruebas siguientes consideran todas las características (antiguas y *SignoNB*) frente a los diversos *kernels* y valores para la constante *C*.

Tabla 5.2: *Benchmarking* de características en general para modelo híbrido dividiendo *dataset* 80/20

C	Lineal	Gaussian	Polinomial
1	0.9372	0.892	0.9447
10	0.9447	0.9309	0.9510
100	0.9447	0.9422	0.9535

La Tabla 5.2 muestra la precisión del modelo híbrido dividiendo, como en pruebas anteriores, el *dataset* de la manera más óptima, 80 % para entrenamiento y 20 % para *testing*, con valores de 1, 10 y 100 para la constante C y utilizando los tres *kernel trick* descritos en secciones pasadas. Se observa que las configuraciones con los resultados más óptimos son cuando el kernel es polinomial y la constante $C = 100$, dando un porcentaje de precisión de 95.35 %.

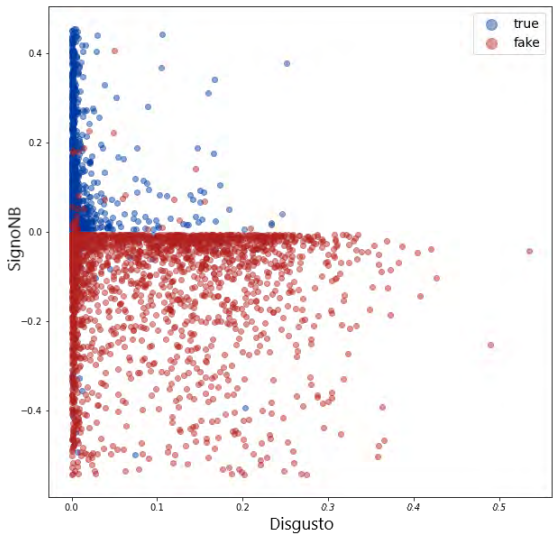


Figura 5.2: Clustering para dupla *Disgusto-SignoNB* en modelo híbrido

La característica de NB mejora en gran medida la precisión del modelo porque es un valor precalculado. Este modelo híbrido presenta mejores resultados ya que sostiene su predicción en fundamentos léxicos, como los histogramas de NB, y semánticos, como el análisis sentimental y emocional de la SVM.

5.1.3. Matriz de confusión

La matriz de confusión generada para el modelo híbrido con los parámetros óptimos de las pruebas de precisión se muestra en la Tabla 5.3 a continuación:

Tabla 5.3: Matriz de confusión para modelo híbrido

		Predicción	
		Verdadero	Falso
Etiqueta real	Verdadero	396	23
	Falso	14	363

Donde se observa lo siguiente:

- Las noticias verdaderas etiquetadas correctamente como verdaderas (VV) son 396 representando el 94.51 % de las noticias verdaderas.
- Las noticias falsas etiquetadas correctamente como falsas (FF) son 363 representando un 96.29 % de las noticias falsas.
- Las noticias verdaderas etiquetadas erróneamente como falsas (VF) son 23 representando un 5.49 % de las noticias verdaderas.
- Las noticias falsas etiquetas erróneamente como verdaderas(FV) son 14 representando un 3.71 % de las noticias falsas.

Aplicando la fórmula de precisión, el Modelo Híbrido (SVM-NB) para el grupo de características en general con la configuración óptima de kernel polinomial con un *dataset* dividido 80 % para *training* y 20 % *validation* obtiene una precisión de 95.35 %.

Los resultados obtenidos con el SVM-NB muestran una mejoría en la precisión de la detección de *fake news* en comparación con el uso únicamente del SVM. Este modelo tiene potencial para ser mejorado en el futuro mediante la incorporación de nuevas características y la optimización de la forma en que se utiliza el conocimiento previo del NB en el SVM. Sin embargo, es importante tener en cuenta que el tiempo y los recursos necesarios para el entrenamiento pueden aumentar a medida que se agreguen nuevas características al *dataset* y se requiera más tiempo para la extracción de características.

Capítulo 6

Prototipado de aplicación

6.1. Persistencia de conocimiento

El conocimiento adquirido por el modelo híbrido se almacena mediante la herramienta *Pickle*, la cual serializa y hace persistente el modelo entrenado SVM en un archivo que puede ser importado en el futuro y que conserva toda la información y conocimiento adquiridos durante el entrenamiento.

Es así como se exportan 5 archivos necesarios para el prototipo:

- fictus_detector_model.pickle: Modelo SVM-NB entrenado con las características de ambos algoritmos. Al importar este archivo ya se tiene la capacidad de realizar predicciones, solo es necesario declarar las funciones con las que se entrenó para definir los parámetros necesarios para la predicción, los cuales son los valores de las características para el análisis sentimental, emocional y de odio, además de *SignoNB*.
- hate_speech_analyzer.pickle: Archivo que contiene los datos correspondientes del create_analyzer preentrenado para *hateful* utilizando la libreria PySentimiento.
- emotion_analyzer.pickle: Archivo que contiene los datos correspondientes del create_analyzer preentrenado para *emotion* utilizando la libreria PySentimiento.
- sentiment_analyzer.pickle: Archivo que contiene los datos correspondientes del create_analyzer preentrenado para *sentiment* utilizando la libreria PySentimiento.
- naive_bayes_analyzer.pickle: Extractor de características preentrenado de NB para obtener los valores correspondientes de *SignoNB*. Contiene también todos los histogramas del entrenamiento del modelo.

Se decidió almacenar el conocimiento adquirido por los analizadores de sentimiento mediante archivos serializados y, de esta manera, evitar generarlos cada vez que el prototipo sea ejecutado. Es así como se puede recuperar el conocimiento almacenado en el *storage* del proyecto simplemente importándolo en lugar de descargarlo y generarlo cada vez utilizando la librería *PySentimiento*.

Además de importar los archivos serializados que almacenan el conocimiento de los modelos entrenados, es necesario definir en el prototipo las funciones correspondientes que

se utilizaron en su entrenamiento. De esta forma, se garantiza la utilización completa del conocimiento adquirido por el modelo.

6.2. Desarrollo de prototipo

Dado que el enfoque principal de este proyecto es el modelo de predicción y no la descripción detallada del prototipo de aplicación, se presentará una explicación breve y general de esta etapa, sin profundizar en los aspectos técnicos de su desarrollo.

El modelo de predicción desarrollado en este proyecto es versátil y puede ser aplicado en diferentes plataformas, incluyendo aplicaciones de escritorio, extensiones de navegador, aplicaciones móviles, entre otras. Además de mostrar los porcentajes de características obtenidas, se pueden incluir gráficos estadísticos para una mejor interacción con los usuarios. En este trabajo, se utilizó el modelo entrenado para crear un bot de Telegram en Python llamado *FictusDetector*, el cual funciona como detector de noticias falsas.

6.2.1. Creación de bot

Para la creación y funcionamiento se utilizarán las APIs de Telegram, además de la herramienta BotFather que facilita el proceso de creación de bots en Telegram, brindando una interfaz intuitiva y fácil de usar para definir el nombre, descripción e imagen del mismo. Al crearlo, se obtiene un token único que se utilizará para mantenerlo en un estado de escucha y responder a los eventos enviados por los usuarios.

Es necesario también, generar un *api_id* y un *api_hash* como credenciales para utilizar la Telegram Database Library (TDLib). En el proyecto importamos Telethon, una librería que brinda múltiples funciones para trabajar con la TDLib, y *Asyncio*, que nos permite ejecutar funciones asíncronas en la aplicación, como *send_messages()* o *get_entity()*. Estas funciones son utilizadas para enviar mensajes a una entidad específica (usuario, canal o bot) y obtener entidades a partir de su nombre o ID, respectivamente.

Para el proyecto, se implementó el bot como un *daemon* y se realizó el *deploy* en una máquina virtual proporcionada por *Google Cloud*. De esta manera, el bot se mantiene listo para recibir y responder a las solicitudes en todo momento.

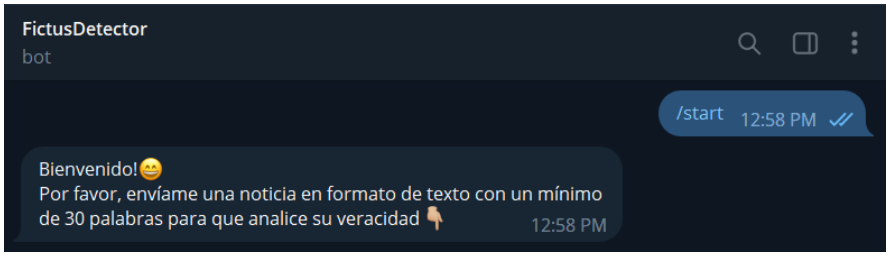


Figura 6.1: Mensaje al iniciar chat por primera vez con *FictusDetector*

6.2.2. Configuración e interfaz

De acuerdo con la documentación de Telegram, los bots no pueden iniciar una conversación por sí mismos, Telegram (2023). Por tanto, nuestro bot se ha programado para estar en modo escucha, esperando que se desencadenen eventos. De esta manera, después de cargar el modelo entrenado con *Pickle* y configurar el bot con la función *TelegramClient*, el bot está listo para recibir y responder a cualquier *EventHandler* que se desencadene. La interfaz incluye una ventana de conversación en la que los usuarios pueden escribir mensajes y recibir respuestas automatizadas por parte del bot.

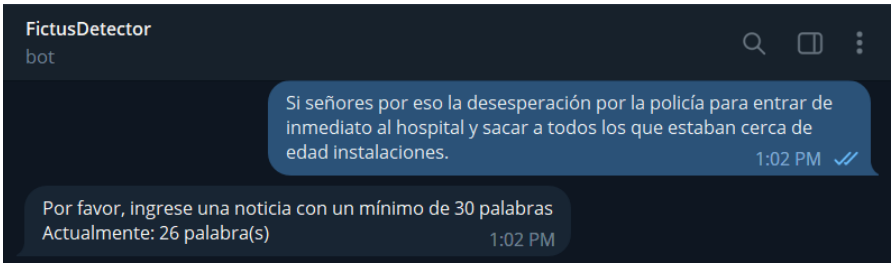


Figura 6.2: Respuesta al no cumplir requisito de palabras por *FictusDetector*

6.2.3. Funcionamiento

Para utilizar el bot, los nuevos usuarios deben buscar el nick @FictusDetectorBot en Telegram, e iniciar una conversación, como se muestra en la Figura 6.1. Una vez hecho este paso ya pueden empezar a enviar sus noticias para su verificación.

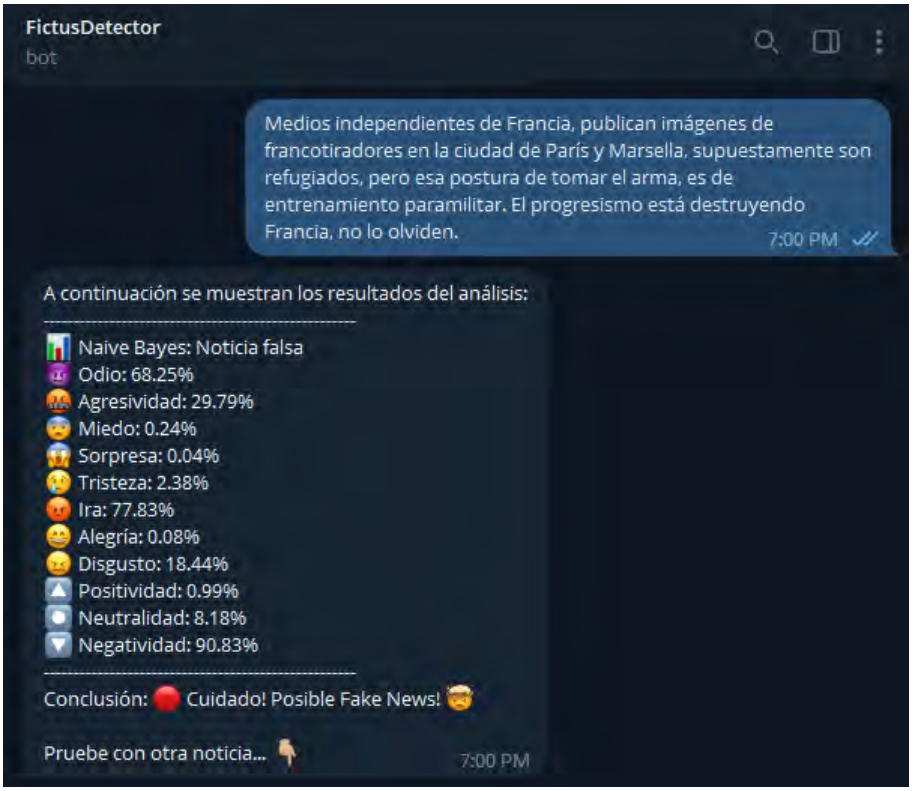


Figura 6.3: Análisis y respuesta de *FictusDetector* respecto a noticia de prueba

Al recibir un evento, se valida la cantidad mínima de tokens a 30 para considerar una noticia útil para el análisis, de no ser el caso se retorna un aviso como se observa en la Figura 6.2. Si cumple con el mínimo de palabras se llama a una función predictora con el objeto del evento como parámetro, el cual incluye el mensaje del usuario, su ID, entre otros datos. Esta función utilizará el modelo SVM-NB preentrenado para determinar la probabilidad de veracidad de la noticia analizando los valores de cada característica. Finalmente, se envía la respuesta al usuario mediante la función *send_message()* de Telethon. La respuesta incluye la predicción de veracidad del texto y las estadísticas de las características analizadas.

La Figura 6.3 muestra el mensaje de respuesta del bot desarrollado frente a una noticia enviada, el cual incluye la clasificación en general del texto y los puntajes obtenidos respectivamente por cada analizador, incluido el de Naive Bayes. Como se observa en el ejemplo, la noticia es etiquetada como falsa por el algoritmo NB representado por la característica *SignoNB*, y recibe nuevamente una calificación de falsa por el modelo híbrido, ya que en general, el resto de características la clasifican como tal, esto demuestra que el modelo SVM-NB realiza predicciones más robustas utilizando ambos conocimientos.

Capítulo 7

Análisis y discusión de resultados

7.1. Análisis de resultados respecto a los objetivos

1. El *dataset* construido en este proyecto es el apropiado, ya que genera resultados con valores de precisión de 95.4 % y 90.45 % para el clasificador NB y el optimizar SVM respectivamente. Sin embargo, esta colección de datos puede mejorarse, buscando fuentes de información más favorables y diferenciadoras para la recolección de noticias en otras plataformas.

2. La limpieza y procesamiento de los datos obtenidos fue realizada con éxito, eliminando elementos gramaticales carentes de significado como *stop words*, signos de puntuación, valores numéricos, entre otros. Sin embargo, se enfrentaron desafíos al momento de realizar la extracción de características, ya que los clústeres generados presentaron poca diferenciación entre ellos. Por lo tanto, para obtener resultados más precisos, se deben considerar las características en conjunto y no solo en pares. Como se menciona previamente, la principal dificultad en esta tarea radica en encontrar las características que diferencian ambas clases de noticias.

3. En el Capítulo 5, se mostró que el modelo híbrido, que complementa el conocimiento de NB y SVM obtiene una precisión final de 95.35 %, un valor menor al de NB por sí solo, pero con un análisis más completo, considerando más variables y contexto.

4. Con los resultados obtenidos, se observa que el modelo NB demuestra una alta precisión en la detección de noticias falsas, utilizando solo dos histogramas de palabras. Sin embargo, la precisión del algoritmo SVM depende en gran medida de las características extraídas del *dataset*. Es por ello que es esencial contar con librerías precisas para el análisis sentimental y emocional en español para mejorar los resultados del SVM. En resumen, aunque NB es un algoritmo sencillo, demuestra resultados efectivos en comparación a SVM que requiere características adicionales para mejorar su precisión.

7.2. Discusión de resultados respecto a los antecedentes

1. En el trabajo presentado en Ahmad et al. (2020), se utiliza el software *LIWC2015* para extraer 93 características diferentes. Aunque las versiones más recientes de esta herra-

mienta tienen mayores capacidades, para este proyecto se optó por utilizar librerías gratuitas como *PySentimiento* y SAS para analizar las características sentimentales y emocionales de los textos. En lugar de utilizar un software específico, se desarrollaron scripts para extraer las propiedades de cada texto. En Ahmad et al. (2020) se realizan pruebas con cuatro *datasets* diferentes, divididos en un 70/30, obteniendo resultados del 98 %, 31 %, 54 % y 88 % respectivamente, utilizando un SVM lineal. A pesar de tener un número menor de propiedades, se obtuvieron resultados similares al dividir nuestro *dataset* en un 80/20 para el SVM del proyecto.

2. Al comparar los histogramas y las nubes de palabras de los trabajos presentados en Zhang et al. (2019) y el actual, es evidente que existen grandes diferencias debido al contexto político, económico y social distinto. Aunque ambos proyectos utilizan la API de Twitter, *FakeDetector*, software del antecedente, se basa en 14 mil textos extraídos de *Politifact*, una cuenta dedicada específicamente a la detección y lucha contra las noticias falsas en el ámbito político, lo que permite encontrar diferencias entre ambas clases de manera más sencilla. Por otro lado, el presente proyecto se basa en 4 mil noticias locales extraídas de los medios de comunicación más populares en el Perú, lo que genera diferentes modos de redacción, múltiples temas y textos irrelevantes, por ello puede ocasionar confusiones en las predicciones realizadas. Aunque en Zhang et al. (2019) no se muestran los resultados de precisión, se menciona que *FakeDetector* es un modelo altamente efectivo y que utiliza aprendizaje profundo y métodos avanzados de *machine learning*.

3. En el trabajo presentado en Aldwairi and Alwahedi (2018) se utiliza nuevamente un software para minería de datos llamado *WEKA*. Este software es especialmente útil para el procesamiento, análisis y clasificación de grandes colecciones de datos debido a que incluye múltiples algoritmos implementados para esta tarea, como NB y *Random Tree*. Asimismo, se realizó un proceso de BM para medir la precisión de cada algoritmo implementado. Los resultados obtenidos son similares a los presentados en el trabajo mencionado, donde se muestra una precisión del 98.7 % para el algoritmo de NB.

4. En el presente proyecto se llevó a cabo un proceso de BM tomando como referencia el trabajo presentado en Younus Khan et al. (2021). Donde se realizaron diversas pruebas de rendimiento para evaluar los modelos pre-entrenados BERT, DistilBERT, RoBERTa, ELECTRA y ELMo, con el objetivo de determinar cuál de ellos proporciona mejores resultados de precisión utilizando una menor cantidad de datos. Se consideraron características léxicas, sentimentales y *n-grams* para las múltiples pruebas. Además, se compararon los resultados de estos algoritmos avanzados con modelos de *deep learning* y modelos tradicionales de *machine learning* como SVM y NB. Se realizaron pruebas sobre 3 *datasets*: *Liar*, *Fake or real news* y *Combined Corpus*, obteniendo resultados del 56 %, 67 % y 71 % para el SVM, y 60 %, 86 % y 93 % para NB, respectivamente. Estos resultados demuestran una precisión similar a los presentados en nuestro proyecto. Por otro lado, los modelos BERT obtuvieron una precisión del 62 %, 98 % y 96 % para cada *dataset*.

5. En primer lugar, tal como se presenta en Espejel et al. (2022), se llevó a cabo un análisis exhaustivo de las colecciones de datos con el objetivo de comprender de manera más precisa la información contenida en los textos de nuestro *dataset*. Los resultados gráficos indican que las diferencias gramaticales entre ambas clases de noticias son mínimas. Además, se realizaron análisis adicionales, como la longitud de los textos y las palabras más frecuentes, lo que nos llevó a concluir que no es posible diferenciar las clases de

noticias únicamente por la cantidad de verbos, adjetivos o sustantivos utilizados, ni por la cantidad de palabras o *stop words*.

En cuanto a las diferencias con el trabajo mencionado, se destaca que en este proyecto se considera no solo la cantidad de elementos gramaticales por cada noticia, sino también la proporción de cada elemento respecto a la cantidad total de palabras de la noticia, lo que resulta en una mayor precisión. Es importante mencionar que los datos utilizados en Espejel et al. (2022) son documentos completos basados en noticias multinacionales, mientras que nuestro *dataset* incluye tweets de noticieros conocidos y con poca credibilidad únicamente en el Perú. Así mismo, los modelos implementados en el antecedente utilizan Bag of Words (BoW), Term Frecuency – Inverse Document Frequency (TF-IDF), además de otro conjunto de características POS y de análisis sentimental. Sin embargo, los resultados obtenidos en el trabajo mencionado son del 77.97 % para el SVM utilizando BoW, los cuales muestran puntajes menores a los obtenidos en este trabajo de investigación.

El modelo híbrido implementado en este proyecto es un enfoque innovador que combina las ventajas de dos algoritmos diferentes, NB y SVM. El modelo precalcula las etiquetas correspondientes a cada noticia utilizando el algoritmo NB, lo que mejora la precisión del SVM al aprovechar tanto el conocimiento léxico proporcionado por NB como el semántico y sintáctico proporcionado por el contexto de cada noticia y el significado que trae cada texto al relacionar palabras. Este enfoque es único en la investigación y tiene el potencial de ser mejorado en trabajos futuros.

7.3. Complicaciones durante la investigación

7.3.1. Creación del dataset

- Para el proyecto, se crearon varios conjuntos de datos. Inicialmente, se recopilaron noticias desde principios de 2021 y se consideraron falsos temas como guerra o ufología. Sin embargo, debido a los acontecimientos ocurridos a principios y mediados de 2022, esto cambió drásticamente. Los conflictos bélicos entre Rusia y Ucrania abrieron la posibilidad de que las noticias referentes a guerras y otros conflictos internacionales puedan ser verdaderas. Esto causó contradicciones en las predicciones del modelo. Un problema similar ocurrió con temas de ovnis y ufología, cuyo tópico era considerado ciencia ficción, exageración o conspiración; sin embargo, tras el informe mundial de la National Aeronautics and Space Administration (NASA) a mediados de 2022, se confirmó la existencia de objetos voladores no identificados como se menciona en el artículo *¿Qué han dicho el Gobierno de EE.UU. y la NASA sobre los ovnis?* de CNN en español. Reyes Haczek (2022). Lo cual cambió la forma en que se clasificaban los textos relacionados con estos temas.
- Dado que las cuentas de Twitter que difunden noticias falsas tienen una credibilidad reducida, en algunos casos esta red social las bloquea, elimina o cambia su estado a privado para evitar la propagación de información errónea. Éste es el caso de cuentas como @malditaternura, @jcd46 y @peru_memoria, las cuales fueron eliminadas en el transcurso de la redacción del presente trabajo. Este hecho ha llevado a una reducción de las fuentes de datos en el proyecto, ya que la API de Twitter ya no tiene acceso a la información de estas cuentas bloqueadas.

- En algunos casos, cuentas y noticieros falsos de Twitter evitan ser censurados reemplazando letras por números, como el ejemplo de la palabra "MU3RT3", cuya estructura evita la censura del algoritmo de esta red social. Para solucionar esto, en el proyecto se desarrolló una expresión regular que elimina los números de este tipo de palabras, dejando tokens sin sentido e inexistentes en el Diccionario de la Real Academia Española (DRAE). Por lo tanto, el análisis ortográfico utilizado en el proyecto, basado en la técnica Non Dictionary Words (NDW), detecta estos casos y les asigna un puntaje más alto a los textos conformados por palabras inexistentes en el diccionario español. De esta manera, se diferencian de las noticias verdaderas.

- Se encontraron problemas de codificación en UTF-8 y Latin-1 con el conjunto de datos generado en formato CSV al abrirlo desde Excel. Los datos generados no se mostraban con el formato correcto como en la plataforma Google Colab utilizando la librería *Pandas*. Por esta razón, se decidió mantener el conjunto de datos en ese formato y trabajar con él.

7.3.2. Naive Bayes

- Como se ha mencionado en el desarrollo del proyecto, esta técnica presenta una debilidad en cuanto a la sintaxis. El algoritmo solo determina las probabilidades evaluando las palabras de manera individual, en lugar de considerar el contexto y sentido de las expresiones. Esto puede ocasionar que conjuntos de palabras sin sentido sean clasificados erróneamente como verdaderos.

- En este estudio se ha observado que existen palabras que se repiten con frecuencia en los textos analizados. En algunos casos, esto puede ser beneficioso para la clasificación, ya que refuerza la etiqueta correcta de la noticia. Sin embargo, en otros puede ser perjudicial, especialmente cuando la palabra que se repite con frecuencia no corresponde a la etiqueta de la noticia. Esto podría tener un gran peso en la probabilidad y cambiar la etiqueta de la noticia a una clase errónea, contrariando las probabilidades de otros *tokens* menos frecuentes pero correctos para la etiqueta del texto.

7.3.3. Support Vector Machine

- Para la implementación del algoritmo SVM en este proyecto, se consideraron varios estudios previos existentes. Aunque existen diversas maneras de desarrollar esta técnica, no se encontró un pseudocódigo claro y conciso para su implementación. Por esta razón, se decidió trabajar sobre la implementación propuesta por Mathieu Blondel, tal y como se describió previamente. Adicionalmente, no se utilizaron las librerías de SVM de *SKLearn*, ya que éstas no ofrecen la misma flexibilidad que el algoritmo descrito.

- La detección de noticias falsas es un problema complejo, ya que uno de los factores clave para lograrlo es identificar las características que las diferencian de las noticias verdaderas. En este proyecto se utilizaron varios softwares para minería de datos, pero muchos de ellos no resultaron ser precisos como versiones de prueba o *trial*. Por lo tanto, se decidió utilizar librerías para esta tarea, aunque se encontraron problemas similares. Solo la librería *PySentimiento* cumplió con las expectativas para el análisis sentimental.

- Como se mostró en el Capítulo 4, los hiperplanos que separan únicamente tuplas de características no resultan efectivos para diferenciar textos verdaderos y falsos. Se considera como posible solución utilizar la técnica de BoW para la representación de los datos del SVM, sin embargo, como se ha demostrado en estudios previos, como el trabajo de Espejel et al. (2022), es posible que tampoco se obtengan resultados precisos. Esto deberá ser verificado en trabajos futuros. De la misma manera, al no poder separar claramente los clústeres de noticias, no se pudo gráficar los hiperplanos correspondientes en las figuras anteriores.

- Un aspecto que se mejoró en este trabajo fue la forma en que se utilizaron las librerías de análisis sentimental y emocional en las noticias. Al iniciar el proyecto, no se tomaba en cuenta el contexto que podría tener cada noticia, ya que se aplicaba el análisis al texto limpio, sin ruido ni signos de puntuación. Sin embargo, se observó que la librería PySentimiento detecta el contexto a través de los signos de puntuación, como se demostró en la Tabla 3.6 de capítulos anteriores, al aplicarla a un texto limpio, sin ruido, se distorsionaba el análisis y cada texto mostraba emociones diferentes. Este problema fue resuelto gracias al trabajo de Espejel et al. (2022), por el cual se mejoró la precisión final del SVM hasta en un 9 %.

- Determinar la configuración de parámetros óptima para el algoritmo de SVM requiere una gran cantidad de tiempo. Por esta razón, se desarrolló un script para automatizar este proceso con los parámetros previamente definidos. Sin embargo, documentar los resultados obtenidos por cada combinación de parámetros es un trabajo extenso; por esta razón, en el BM realizado en la Sección 4.2 presenta únicamente los resultados obtenidos con el análisis de todas las características en general, y no por duplas o ternas.

7.3.4. Modelo híbrido

- A pesar de tener los algoritmos necesarios para esta etapa, se enfrentaron problemas al momento de considerar y normalizar la nueva característica basada en el algoritmo NB para utilizarla adecuadamente en la SVM. Inicialmente se intentó generar un valor distinto a los obtenidos anteriormente utilizando las probabilidades en potencias de 10. Sin embargo, esto causó problemas en los casos con probabilidades extremadamente grandes y pequeñas, con más de 30 decimales. Finalmente, se propuso utilizar el valor NB ya existente e invertir su signo para los casos de noticias falsas.

Conclusiones

1. La construcción del *dataset* de noticias es un proceso complejo que requiere un gran conocimiento. Uno de los mayores desafíos fue encontrar fuentes confiables de información para recolectar las noticias. A pesar de esto, se logró construir un conjunto de datos de alta calidad que cumplió con las necesidades de la investigación.

2. Antes de procesar los datos, se realizó un análisis de las particularidades de cada colección, con el objetivo de identificar las características distintivas entre noticias verdaderas y falsas. Se utilizaron diferentes librerías para la extracción de características, y se encontró que algunas de éstas requerían datos en bruto para obtener resultados precisos. La limpieza de ruido se llevó a cabo con el fin de mejorar la calidad de los datos, lo cual, a su vez, mejoró la precisión de los resultados finales.

3. Para aplicar el clasificador NB, se construyeron dos histogramas de palabras con los datos limpios y procesados previamente. Por otro lado, para optimizar el algoritmo SVM fue necesario contar con un conjunto sólido de características distintivas. La combinación de ambos algoritmos predictivos contribuyó de manera significativa a la eficacia del modelo híbrido desarrollado.

4. Los resultados obtenidos mediante las diversas pruebas de *benchmarking* mostraron un alto nivel de precisión en el modelo desarrollado. En concreto, el algoritmo híbrido obtuvo una precisión del 95.35 % utilizando el conocimiento del clasificador NB complementado con el optimizador SVM. Este resultado indica un alto grado de eficacia en la detección de noticias falsas.

Recomendaciones

1. Una recomendación importante para futuros estudios es ampliar y mejorar el dataset existente. Una forma de lograrlo es desarrollar un sistema de actualización automática del dataset, con un periodo de actualización semanal o mensual. De esta manera, se podría contar con información actualizada y más precisa, lo cual permitiría mejorar la precisión en la detección de noticias falsas.

2. Se recomienda ampliar el alcance del estudio para incluir formatos digitales adicionales como imágenes, audio e incluso video, ya que estos son algunos de los formatos más comunes utilizados en la propagación de noticias falsas. Esto permitiría evaluar la efectividad de los modelos desarrollados en un rango más amplio y proporcionar una comprensión más completa del problema en cuestión.

3. Los modelos desarrollados en esta investigación tienen un gran potencial para ser aplicados en una variedad de plataformas, como aplicaciones móviles o web. Estos podrían ser utilizados para detectar noticias falsas en redes sociales o como una extensión de navegador capaz de detectar *fake news* en múltiples páginas web. Estas aplicaciones podrían ayudar con la detección de noticias falsas y contribuir a la lucha contra la desinformación en el Perú.

4. La obtención de datos a través de la API de Twitter y el *web scraping* puede generar diferencias significativas en la calidad de los datos, originando, de esta manera, errores en las predicciones. Por esta razón, se recomienda utilizar una única técnica de extracción de datos para evitar este problema y garantizar una mayor homogeneidad y precisión en los datos recolectados.

5. En esta investigación se ha observado la importancia de la representación de los datos en la precisión del modelo. Aunque se realizó un análisis exploratorio de los datos para determinar las características diferenciadoras y se logró definir de manera satisfactoria las mismas, se recomienda investigar técnicas de representación de datos más avanzadas, como la utilización de plataformas de pago para la extracción de características como LIWC2022 y WEKA, o la vectorización de los datos con el objetivo de mejorar los resultados obtenidos.

Bibliografía

(2021a). Data dictionary: Standard v1.1. url: https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/object-model/tweet. [recuperado el 25-09-2022].

(2021b). Twitter api v2. url: https://developer.twitter.com/en/docs/twitter-api. [recuperado el 24-09-2022].

Adhanom Ghebreyesus, T. (2021). Fighting misinformation in the time of covid-19, one click at a time. url: https://www.who.int/news-room/feature-stories/detail/fighting-misinformation-in-the-time-of-covid-19-one-click-at-a-time. [recuperado el 15-01-2022].

Agarwal, D. (2021a). Guide for feature extraction techniques. url: https://www.analyticsvidhya.com/blog/2021/04/guide-for-feature-extraction-techniques/. [recuperado el 11-05-2022].

Agarwal, D. (2021b). Introduction to svm(support vector machine) along with python code. url: https://www.analyticsvidhya.com/blog/2021/04/insight-into-svm-support-vector-machine-along-with-code/. [recuperado el 08-09-2022].

Agarwal, D. (2022). Introduction to svm(support vector machine) along with python code. url: https://www.analyticsvidhya.com/blog/2021/04/insight-into-svm-support-vector-machine-along-with-code/. [recuperado el 06-02-2023].

Ahmad, I., Yousaf, M., and Yousaf, S. (2020). *Fake News Detection Using Machine Learning Ensemble Methods*. Hindawi, Peshawar, Pakistan.

Aldwairi, M. and Alwahedi, A. (2018). *Detecting Fake News in Social Media Networks*. Elsevier, Abu Dhabi, Emiratos Árabes Unidos.

Alpaydın, E. (2010). *Introduction to Machine Learning*. The MIT Press, Massachusetts.

Amat Rodrigo, J. (2020). Máquinas de vector de soporte (svm) con python. url: https://www.cienciadedatos.net/documentos/py24-svm-python.html. [recuperado el 03-12-2022].

Andersen, M., Dahl, J., and Vandenberghe, L. (2022). *Python Sfotware for Convex Optimization*. [recuperado el 03-12-2022].

Berg, S. (2022). Numpy: About us. url: https://numpy.org/about/. [recuperado el 01-09-2022].

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media, Inc., Gravenstein Highway North, Sebastopo.

Blondel, M. (2010). Support vector machines. url: https://gist.github.com/mblondel/586753. [recuperado el 08-11-2022].

Chacón, L. M. C. (2022). La cobertura electoral partidista y la creciente hostilidad hacia la prensa contribuyeron a la disminución de la confianza en los medios de perú en un año en el que se convirtió en el país con la mayor tasa de mortalidad por covid-19 del mundo. url: https://reutersinstitute.politics.ox.ac.uk/es/digital-news-report/2022/peru. [recuperado el 31-06-2023].

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Cambridge, England.

del Arco, F. M. P., Strapparava, C., Lopez, L. A. U., and Martín-Valdivia, M. T. (2020). Emoevent: A multilingual emotion corpus based on different events. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1492–1498.

@DiarioElPeruano (2023). Siete presuntos autores del delito. url: https://twitter.com/DiarioElPeruano/status/1675297476947918848. [recuperado el 01-07-2023].

Domo (2022). Hybrid machine learning. url: https://www.domo.com/glossary/what-is-hybrid-machine-learning. [recuperado el 03-12-2022].

Dongo, D. (2023). Repositorio del código fuente de la tesis. GitLab repository. Este repositorio contiene el código fuente relacionado con la tesis y se encuentra disponible en línea en la siguiente dirección: `https://gitlab.com/diefrek/fake-news-detection-source-code`. Para consultas póngase en contacto con el autor en `dongoyoshiro@gmail.com` o a través de su perfil de LinkedIn en `www.linkedin.com/in/diego-yoshiro-dongo`.

EC, R. (2023). Embajador de chile: Perú asumirá presidencia de alianza del pacífico hasta primer trimestre del 2024. url: https://elcomercio.pe/politica/actualidad/embajador-de-chile-peru-asumira-presidencia-de-alianza-del-pacifico-hasta-primer-trimestre-del-2024-ultimas-noticia/. [recuperado el 01-07-2023].

EFE, A. (2023). Muere el escritor y periodista cubano carlos alberto montaner. url: https://elcomercio.pe/mundo/actualidad/carlos-alberto-montaner-muere-el-escritor-y-periodista-cubano-video-madrid-miami-ee-uu-ultimas-noticia/. [recuperado el 01-07-2023].

@eljokerpe (2023). Inputado por delito contra menor. url: https://twitter.com/eljokerpe/status/1636557593144438785. [recuperado el 01-07-2023].

@elzorrotacneno (2023). Vacunas con propósitos letales. url: https://twitter.com/elzorrotacneno/status/1674048230722510850. [recuperado el 01-07-2023].

Espejel, R., Calderón, S., Ortega, M., Camacho, B., and Márquez, V. (2022). *Detección automática de noticias falsas usando representaciones textuales tradicionales y soluciones basadas en aprendizaje profundo*, volume 10. Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI.

@exitosape (2023). Informe movimiento telúrico en en arequipa. url: https://twitter.com/exitosape/status/1675318318490591232. [recuperado el 01-07-2023].

Fabian, P., Gaël, V., and Alexandre, G. (2011). Scikit-learn: Machine learning in python.

Fassihi, F. (2018). Fake news. url: https://news.un.org/en/audio/2018/05/1008682. [recuperado el 15-01-2022].

Hernández Sampieri, R., Fernández Collado, C., and Baptista Lucio, P. (2014). *Metodología de la investigación*. McGraw-Hill, México D.F, 6ta edition.

Howard, J. and Gugger, S. (2020). *Deep Learning for Coders with fastai & PyTorch*. O'Reilly Media, Canada.

J. Bello, H. (2021). Sentiment analysis for sentences in spanish. url: https://pypi.org/project/sentiment-analysis-spanish/. [recuperado el 20-10-2022].

JavaTPoint (s.f.). Naïve bayes classifier algorithm. url: https://www.javatpoint.com/machine-learning-naive-bayes-classifier2. [recuperado el 15-01-2022].

Jin, C. and Wang, L. (2012). *Dimensionality dependent PAC-Bayes margin bound. Advances in Neural Information Processing Systems.* [recuperado el 12-05-2022].

Joshi, P. (2013). What is k-means clustering? url: https://prateekvjoshi.com/2013/06/06/what-is-k-means-clustering/. [recuperado el 12-05-2022].

Kurtzleben, D. (2018). Did fake news on facebook help elect trump? here's what we know. url: https://www.npr.org/2018/04/11/601323233/6-facts-we-know-about-fake-news-in-the-2016-election. [recuperado el 31-08-2022].

@Liberfach0 (2023). Francotiradores en francia. url: https://twitter.com/Liberfach0/status/1674834592719618050. [recuperado el 01-07-2023].

Matos, E. (2023). Temblor en perú hoy: epicentro del último sismo este 1 de julio, vía igp. url: https://larepublica.pe/sociedad/2023/07/01/temblor-de-hoy-peru-2023-en-vivo-donde-fue-el-epicentro-magnitud-que-dice-el-igp-y-ultimas-noticias-instituto-geofisico-del-peru-sismos-de-hoy-en-lima-ica-callao-atmp-20093. [recuperado el 01-07-2023].

Meyer, C. D. (2000). *Matrix Analysis and Applied Linear Algebra*. [recuperado el 07-09-2022].

OPS (2020). Entender la infodemia y la desinformación en la lucha contra la covid-19. url: https://iris.paho.org/bitstream/handle/10665.2/52053/Factsheet-Infodemic_spa.pdf. [recuperado el 31-08-2022].

Parzen, E. (2021). *Teoría moderna de probabilidades y sus aplicaciones*. [recuperado el 11-05-2022].

Perú21, R. (2023a). Economía peruana: Qué le espera a nuestro país para los próximos seis meses. url: https://peru21.pe/economia/incertidumbre-economia-peruana-economia-peruana-que-le-espera-a-nuestro-pais-para-los-proximos-6-meses-noticia/. [recuperado el 01-07-2023].

Perú21, R. (2023b). Inpe: identifican a internos extranjeros que al cumplir su sentencia serán expulsados del país. url: https://peru21.pe/lima/inpe-identifican-a-internos-extranjeros-que-al-cumplir-su-sentencia-seran-expulsados-del-pais-reclusos-penal-de-lurigancho-inpe-migraciones-noticia/. [recuperado el 01-07-2023].

posesodegerasa (2023a). El chip injertable. url: https://astillasderealidad2.blogspot.com/2021/12/el-chip-injertable.html. [recuperado el 01-07-2023].

posesodegerasa (2023b). El engaño .ºmicron", ¿cuánto tiempo puede proseguir la farsa de las "variantes¿ url: https://astillasderealidad2.blogspot.com/2021/12/el-engano-omicron-cuanto-tiempo-puede.html. [recuperado el 01-07-2023].

posesodegerasa (2023c). Los niños, inmunes al covid, están muriendo ahora a causa de la inyección. url: https://astillasderealidad2.blogspot.com/2021/12/los-ninos-inmunes-al-covid-estan.html. [recuperado el 01-07-2023].

Pérez, J. M., Giudici, J. C., and Luque, F. (2021). pysentimiento: A python toolkit for sentiment analysis and socialnlp tasks.

Reyes Haczek, A. (2022). ¿qué han dicho el gobierno de ee.uu. y la nasa sobre los ovnis? url: https://cnnespanol.cnn.com/2022/07/02/ivnis-nasa-estados-unidos-orix/. [recuperado el 17-01-2023].

Rodríguez-Andrés, R. (2018). Trump 2016: ¿presidente gracias a las redes sociales? pages 1–2.

Roman, L. (2023a). El dengue no se da por "falta de calcio", sino por un virus. url: https://larepublica.pe/verificador/2023/06/01/el-dengue-no-se-da-por-falta-de-calcio-sino-por-un-virus-94764. [recuperado el 01-07-2023].

Roman, L. (2023b). La nasa no admitió que el cambio climático ocurre "solo" de forma natural. url: https://larepublica.pe/verificador/2023/06/22/la-nasa-no-admitio-que-el-cambio-climatico-ocurre-solo-de-forma-natural-234498. [recuperado el 01-07-2023].

Roman, L. (2023c). Pepe mujica no tuiteó este mensaje a favor del gobierno de gustavo petro. url: https://larepublica.pe/verificador/2023/06/24/pepe-mujica-no-tuiteo-este-mensaje-a-favor-del-gobierno-de-gustavo-petro-1808112. [recuperado el 01-07-2023].

@RPPNoticias (2023a). Banda integrada por hermanos. url: https://twitter.com/RPPNoticias/status/1675320778764460038. [recuperado el 01-07-2023].

@RPPNoticias (2023b). Nombramiento de ministros. url: https://twitter.com/RPPNoticias/status/1675341665874108418. [recuperado el 01-07-2023].

@RPPNoticias (2023c). Twitter restringiría temporalmente la lectura de tuits. url: https://twitter.com/RPPNoticias/status/1675270480863850497. [recuperado el 01-07-2023].

Telegram (2023). Telegram apis. url: https://core.telegram.org/api. [recuperado el 01-02-2023].

@Ufologopedro1 (2023). Objeto volador en polonia. url: https://twitter.com/Ufologopedro1/status/1672333491671908354. [recuperado el 01-07-2023].

Xue, J., Wang, Y., Tian, Y., Li, Y., Shi, L., and Wei, L. (2021). Detecting fake news by exploring the consistency of multimodal data. *Inf. Process. Manage.*, 58(5).

Younus Khan, J., Islam Khondaker, T., and Afroz, S. (2021). *A benchmark study of machine learning models for online fake news detection.* Elsevier, Dhaka, Bangladesh.

Zhang, J., Dong, B., and S. Yu, P. (2019). *FAKEDETECTOR: Effective Fake News Detection with Deep Diffusive Neural Network.* The University Press of Florida, Florida, USA.

Anexos

Anexo A: *Testing* utilizando noticias externas al *dataset*

A continuación se muestran los resultados obtenidos utilizando el modelo híbrido para analizar noticias externas al *dataset*:

Tabla 7.1: *Testing* utilizando noticias externas al *dataset*

N	Noticia	Etiqueta	Predicción
1	Culminado el primer semestre del año, el balance de la economía peruana no es muy favorable, y es que a los rezagos de la pandemia se sumaron los conflictos sociales ocurridos en la primera parte del año, el ruido político y la tensa presencia del Fenómeno del Niño. Fuente: Perú21 (2023a)	True	True
2	El escritor y periodista cubano exiliado Carlos Alberto Montaner, que padecía una enfermedad neurodegenerativa y desde 2022 había trasladado su residencia de Miami a Madrid, falleció en su domicilio madrileño acompañado de sus seres queridos, informaron a EFE fuentes de su entorno. Fuente: EFE (2023)	True	True
3	El Perú asumirá la presidencia pro témpore de la Alianza del Pacífico, tal y como acordaron los cuatro países miembro, el 1 de agosto en Santiago y la mantendrá hasta el primer trimestre del 2024, según informó el embajador de Chile en el Perú, Óscar Fuentes Lira. Fuente: EC (2023)	True	True
4	Con la finalidad de regularizar su identificación hasta que se haga efectiva su salida del territorio peruano tras cumplir su condena, en el penal de Lurigancho se realizó el acto de enrolamiento de 272 internos extranjeros, a través de la alianza el Instituto Nacional Penitenciario (INPE) y la Superintendencia Nacional de Migraciones. Fuente: Perú21 (2023b)	True	True

5	Es habitual que en nuestro país ocurran varios movimientos sísmicos debido a que se encuentra ubicado en el Cinturón de Fuego del Pacífico. Ante ello, el Instituto Geofísico del Perú (IGP) se encarga de brindar información oficial sobre el epicentro, magnitud y hora exacta de los temblores ocurridos en el territorio nacional durante el día a través de sus redes sociales. Fuente: Matos (2023)	True	True
6	Elon Musk anunció que Twitter restringiría temporalmente la lectura de tuits, con el fin de reducir el uso masivo de datos por parte de terceros para contener el uso de datos por la inteligencia artificial. Fuente: @RPPNoticias (2023c)	True	True
7	Siete presuntos autores del delito de violación sexual fueron incluidos en el Programa de Recompensas del @MininterPeru, que ofrece beneficios económicos de entre S/ 50,000 y S/ 80,000 por información que facilite su ubicación y captura. Fuente: @DiarioElPeruano (2023)	True	True
8	A punto de cumplir dos años de este nuevo gobierno (compartido entre Castillo y Boluarte) se marcó un nuevo récord. Con el nombramiento de César Vásquez como ministro de Salud, el Perú alcanzó los 106 nombramientos de ministros desde julio de 2021. Fuente: @RPPNoticias (2023b)	True	True
9	Una banda integrada por dos hermanos y su cómplice se habrían hecho pasar como colectiveros para cometer sus fechorías. Tras los asaltos, los agraviados denunciaron que les arrojaban un líquido irritante en los ojos. Fuente: @RPPNoticias (2023a)	True	Fake
10	Un sismo se registró hace instantes a 18 kilómetros al oeste de Atico, Caraveli - Arequipa. Según informó el IGP, el movimiento telúrico tuvo una magnitud de 5.1 y una profundidad de 41 kilómetros. Fuente: @exitosape (2023)	True	True
11	Graban "magnífica.^{es}fera en Polonia. de Junio 2023. Una de las imágenes mas claras de una .^{es}fera", grabadas y fotografiadas en el mundo, ocurrió hace unos días en Polonia, cuando el objeto sobrevolaba la ciudad, a plena luz del día... Es momento de creer. Fuente: @Ufologopedro1 (2023)	Fake	Fake
12	Un estudio reciente publicado en The Lancet volvió a confirmar que la prevalencia del virus está aumentando en personas completamente vacunadas. Después de inspeccionar nuevas infecciones en Alemania, los investigadores encontraron que la tasa de casos entre las personas de 60 años o más completamente vacunadas ha aumentado del 16,9 % en julio al 58,9 % en octubre. Fuente: posesodegerasa (2023c)	Fake	True

13	Darpa presentó a principios de 2021 el chip injertable, un biochip implantable con Wireless fabricado por la empresa Microsemi que puede ser inyectado parenteralmente en el organismo humano y que, como reconocen cientos de estudios y patentes, modifica el ADN. Ese chip envía información externa y registra cada modificación genética. Fuente: posesodegerasa (2023a)	Fake	True
14	Nada muta más rápido que un virus inexistente, excepto quizás los pronunciamientos de Tony Fauci sobre la "pandemia". A principios de 2020, todo comenzó con un "virus" que nadie había aislado. Es decir, un fantasma, una falsificación, una estafa, una no entidad. SIN VIRUS HASTA HOY. Fuente: posesodegerasa (2023b)	Fake	Fake
15	El troll Gerardo Lipe es un imputado por el DELITO de actos contra UN MENOR de 14 AÑOS, y es parte de la organización criminal de Trolls que desde hace 10 años difama y calumnia. Fuente: @eljokerpe (2023)	Fake	Fake
16	Medios independientes de Francia, publican imágenes de francotiradores en la ciudad de París y Marsella, supuestamente son refugiados, pero esa postura de tomar el arma, es de entrenamiento paramilitar. El progresismo está destruyendo Francia, no lo olviden. Fuente: @Liberfach0 (2023)	Fake	Fake
17	"Los ataques mediáticos derrotaron a Bolivia y Perú. Ahora van por Colombia. Los medios tradicionales persisten en su campaña de desinformación y tergiversación contra el gobierno de @petrogustavo. No les importa la verdad, sino buscan derrocar a la izquierda. Fuerza", se lee en un mensaje del 24 de junio, que proviene del perfil "José Mujica". Fuente: Roman (2023c)	Fake	Fake
18	La NASA admite que el cambio climático sólo ocurre de forma normal y natural debido a los cambios en la órbita solar de la Tierra, no debido a la actividad industrial humana o a los combustibles fósiles. El único factor importante que afecta al clima en la Tierra es el Sol. Fuente: Roman (2023b)	Fake	True
19	El dengue da por falta de calcio, si sienten los síntomas del dengue, toma calcio con vitamina D. Te aseguro que de los dos tipos de dengue, tanto el clásico como el hemorrágico, se te quita. No hay riesgo de muerte con el tratamiento. Asegura medico. Fuente: Roman (2023a)	Fake	True
20	Las llamadas vacunas COVID tienen dos propósitos fundamentales: 1. La hibridación del ser humano con fines transhumanistas. 2. La preparación de otra pLandemia, más letal, que permita la instauración de una tiranía global jamás vivida, planeada desde hace décadas y conocida como el Nuevo Orden Mundial. Fuente: @elzorrotacneno (2023)	Fake	Fake