ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UN MODELO DE SÍNTESIS Y RANKING DE HECHOS RELEVANTES APLICADO A MEDIOS DE PRENSA DIGITALES EN BASE A MINERÍA DE DATOS Y RETROALIMENTACIÓN DEL USUARIO FINAL

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE MAGÍSTER EN SISTEMAS DE INFORMACIÓN
MENCIÓN INTELIGENCIA DE NEGOCIOS Y ANALÍTICA DE DATOS MASIVOS

FERNANDO ANDRÉS CEVALLOS SALAS

fernando.cevallos03@epn.edu.ec

DIRECTOR: LORENA KATHERINE RECALDE CERDA, PhD.

lorena.recalde@epn.edu.ec

CODIRECTOR: EDISON FERNANDO LOZA AGUIRRE, PhD.

edison.loza@epn.edu.ec

Quito, agosto 2024

AVAL

Certificamos que el presente trabajo fue desarrollado por Fernando Andrés Cevallos Salas, bajo nuestra supervisión.

_____

PhD. Lorena Katherine Recalde Cerda
DIRECTOR DEL TRABAJO DE TITULACIÓN

_____

PhD. Edison Fernando Loza Aguirre
CODIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Fernando Andrés Cevallos Salas, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

_____

Fernando Andrés Cevallos Salas

DEDICATORIA

A mi tío Jorge Cevallos Serrano.

Q.E.P.D.

AGRADECIMIENTO

A mi madre, a mi padre y a mis hermanos.

A la PhD. Lorena Recalde Cerda y al PhD. Edison Loza Aguirre.

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

El crecimiento de usuarios en Internet, incrementado más aún durante la pandemia de COVID-19, a la par del crecimiento exponencial de noticias electrónicas; han generado la necesidad de crear nuevas herramientas que permitan analizar y sintetizar las ideas que buscan transmitirse a través de estas noticias. El presente proyecto de titulación presenta un modelo para clusterización de noticias, síntesis de hechos relevantes y ranking. El modelo se fundamenta en prácticas de minería de datos haciendo uso de modelos Large Language Models (LLMs) y Machine Learning (ML), y permite la retroalimentación en base a la valoración del usuario final. El modelo conforma una métrica compuesta que permite filtrar la información que se considera útil y un conocimiento valioso para el usuario.

El desarrollo de este modelo ha sido realizado siguiendo el marco de referencia de la metodología Cross Industry Standard Process for Data Mining. Esto ha permitido un avance gradual para poder ir afinando los resultados en base a las variables de entrada y salida de cada una de las fases seguidas.

Los resultados obtenidos tras implementar los modelos desarrollados han sido satisfactorios. Se generaron 7.761 clústers de diarios afines y se ha podido sintetizar el contenido relevante para el usuario. Se deduce del análisis que la media del contenido tras ser puntuado es de 54,13 sobre 100 puntos. A la vez se construyó un modelo de regresión para poder predecir la autoridad que generaría en Internet el titular con un score de predicción del 91,85%.

PALABRAS CLAVE: aprendizaje de máquina, BERT, inteligencia artificial, modelos de lenguaje de gran tamaño, prensa digital, sistema de puntuaciones

ABSTRACT

Internet users' growth, which has been increased even more during the COVID-19 pandemic, along with the exponential growth of electronic news, has generated the need to create new tools which allow analyzing and synthesizing the ideas which want to be transmitted. In this project a model for news clustering, synthesis of relevant facts and ranking is presented. The model is based on data mining practices using large language models (LLM) and machine learning (ML), and allows feedback based on the end user's assessment. The model, based on several factors, creates a composite metric that allows filtering the information that is considered useful and valuable knowledge for the end user.

The development of this model has been carried out following the reference framework of the Cross Industry Standard Process for Data Mining methodology. The CRISP-DM methodology has allowed gradual progress while being able to refine the results based on the input and output variables of each of the phases followed.

Obtained results after implementing the developed models have been satisfactory. A total of 7,761 clusters of related news stories have been generated, and the relevant content for the end user has been synthesized. It can be deduced from the analysis that the average of the content after being scored is 54.13 in a scale of 100 points. At the same time, a regression model was built to predict the authority that the news story would generate on the Internet, the prediction score obtained has been of 91.85%.

KEYWORDS: artificial intelligence, BERT, digital newspapers, large language models, machine learning, ranking system

1. INTRODUCCIÓN

1.1 Planteamiento del Problema

En el comienzo del tercer milenio, nos encontramos inmersos en un mundo lleno de información que circula a lo largo y ancho de nuestro planeta. Esta gran cantidad de datos ha generado nuevas tendencias en el ejercicio comunicativo, lo que ha venido a crear un nuevo medio de comunicación, Internet, que tendría que sumarse y adaptar a los medios ya conocidos como la prensa escrita [1]. La cantidad de información que se genera cada día en Internet es impresionante. Se estima que el volumen de datos en el mundo podría alcanzar los 163 zetabytes para el año 2025 y que, en ese mismo año, la persona promedio interactuará 4.800 veces al día con dispositivos electrónicos [2]. Esta gran cantidad de información, que crece a pasos agigantados, ha sido nombrada como big data.

La era digital afecta a todos los ámbitos de los medios de comunicación tradicionales como la prensa escrita, los cuales utilizan las redes, se reinventan y crecen continuamente con nuevas herramientas y servicios para aprovechar al máximo las ventajas, los atractivos y los valores añadidos que proporcionan las tecnologías de la información y de la comunicación [3]. A medida que big data se incrementa y que Internet engloba cada vez más modelos de negocio, se presentan nuevos desafíos. Entre ellos, el más prominente es la capacidad de obtener información de calidad y valor para cada usuario individual. La ingente cantidad de datos generados puede resultar abrumadora para el usuario común, lo que impulsa la necesidad de crear herramientas innovadoras que procesen y ofrezcan información de calidad y personalizada, en sintonía con las necesidades de cada individuo. Para que la big data se traduzca en conocimiento útil para el usuario final, debe ser de fácil acceso. Esto exige un meticuloso trabajo de indexación y síntesis, transformando la vastedad de datos en información relevante y organizada.

A pesar del auge de Internet y las redes sociales, la prensa escrita conserva un rol fundamental en las comunidades. Su credibilidad y objetividad, avaladas por equipos profesionales de la industria, la convierten en una fuente de información de mayor confianza. Los medios de comunicación como los diarios electrónicos son los encargados de mantenernos informados día a día, no solo del acontecer nacional, sino también de todos los hechos importantes de trascendencia internacional; de ahí que todos sus medios traspasen barreras, permitiendo así que incluso los compatriotas radicados en el exterior tengan la posibilidad de mantenerse siempre informados de todo lo concerniente a nuestro país y el mundo [4]. En este contexto digital, la prensa escrita debe adaptarse y convertirse

en un actor fundamental en Internet. Su larga trayectoria y probada utilidad como fuente de información, incluso desde antes del auge de la red, la convierten en un activo invaluable para el ecosistema digital.

La incursión de la prensa escrita en el ámbito digital trae consigo nuevos desafíos específicos para este modelo de negocio. Entre ellos, destaca el intenso aumento de la competencia de editoriales y la proliferación de las redes sociales. Las noticias en Internet son un producto cuyo contenido es hasta cierto punto de producción sencilla, tamaño pequeño, ciclo de vida rápido y bajo costo [5]. La irrupción de la prensa de forma digital ha reconfigurado el modelo de negocio, intensificando la competencia y ha dado lugar a una nueva industria con sus propias definiciones y características. En muchos aspectos el efecto de Internet sobre la prensa se puede comparar al que ha tenido sobre la música, que ha enfrentado una mayor competencia (en este caso debido a la piratería) y que como consecuencia ha necesitado reinventarse, con resultados aún impredecibles, después de más de diez años [6].

Actualmente, proliferan los diarios electrónicos, atrayendo a usuarios que buscan información a su gusto o por sus editoriales favoritas. Sin embargo, la elección individual puede limitar la interacción con la big data, dejando sin uso una parte de su vasto contenido. Es imperativo aprovechar al máximo la información recopilada en la big data, pues de otro modo, su recolección carece de sentido. Sin las herramientas adecuadas, la interacción con la información se torna limitada, generando una percepción de baja calidad en la misma. Equipar a los usuarios con las herramientas tecnológicas adecuadas es importante para optimizar su análisis de la información. Estas herramientas les permitirán explorar la big data de manera eficiente y obtener una visión organizada de los datos, facilitando la identificación de patrones y la extracción de conocimiento valioso.

1.2 Objetivo General

Desarrollar un modelo de síntesis y ranking de hechos relevantes aplicado a medios de prensa digitales en base a minería de datos y retroalimentación del usuario final.

1.3 Objetivos Específicos

- Realizar una revisión previa de la literatura existente para identificar las mejores alternativas que permitan desarrollar los componentes del sistema.

- Definir las fases de almacenamiento y procesamiento de datos con los distintos indicadores que intervienen en el mecanismo de puntuaciones para la construcción de cada componente.
- Diseñar un mecanismo de puntuación de noticias electrónicas que se defina en base a una métrica compuesta en la que intervengan indicadores como la reputación del diario, opinión de usuario, correlación con otras noticias, entre otras.
- Realizar un análisis de las noticias electrónicas para poder agruparlas en base a la similitud de sus títulos y puntuar su contenido para sintetizar los hechos importantes.
- Probar los componentes desarrollados con un dataset de medios digitales.

1.4 Marco Teórico

1.4.1 Web Scraping

El web scraping es el conjunto de técnicas que se utilizan para obtener automáticamente información de un sitio web en lugar de hacerlo manualmente copiándolo [7]. Estas técnicas ofrecen una amplia gama de opciones para los diversos objetivos que se busquen. La selección de las técnicas de web scraping dependen de varios factores. Entre los principales factores se pueden mencionar:

- **Tipos de datos:** El tipo de datos que se necesita extraer condiciona la técnica utilizada. Los tipos de datos pueden ser imágenes, texto, entre otros.
- **Tecnología del sitio web:** Las páginas web están desarrolladas con diferentes tecnologías como HTML, Javascript o APIs de consumo. Las técnicas seleccionadas deben soportar estas tecnologías para una extracción adecuada.

Las técnicas de web scraping que se pueden utilizar son [8] [9]:

- **Expresiones Regulares:** Una expresión regular es un patrón de texto consistente en una combinación de caracteres alfanuméricos y caracteres especiales denominados meta caracteres [10]. En base a expresiones regulares se puede identificar estos patrones en el texto y filtrarlos.

- **HTML Parsing:** Consiste en el uso de librerías previamente diseñadas para analizar código HTML. Estas librerías permiten extraer información como imágenes, texto, enlaces, y demás componentes basados en las etiquetas de HTML.

- **XPath:** Xpath permite analizar contenido XML. El objetivo principal de XPath es abordar partes de un documento XML. En apoyo de este propósito principal, también proporciona instalaciones básicas para manipulación de cadenas, números y booleanos. XPath utiliza una sintaxis compacta, no XML, para facilitar su uso. XPath opera sobre la estructura lógica y abstracta de un Documento XML, en lugar de su sintaxis superficial. XPath recibe su nombre del uso de una notación de ruta como la URL para posteriormente navegar a través de la estructura jerárquica de un documento XML [11].

- **Selectores CSS:** Los selectores CSS son componentes específicos de estilos que se encuentran integrados en el documento HTML. Estos selectores pueden ser utilizados como una palabra de búsqueda para encontrar información concreta. Los selectores CSS ofrecen un mecanismo altamente eficiente y conveniente para filtrar la información en base al aspecto visual que la página provee.

- **Web APIs:** En la actualidad, un elevado número de sitios web implementa una arquitectura basada en APIs para la gestión de su backend. La interacción con estas APIs, mediante solicitudes previamente elaboradas, nos permite obtener la información específica que buscamos. La principal ventaja de esta técnica reside en que la información devuelta por las APIs ya ha sido procesada y organizada en una estructura predefinida, utilizando formatos estandarizados como JSON y XML. Esto elimina la necesidad de realizar tediosas tareas de filtrado y análisis.

Actualmente existe una amplia gama de librerías y frameworks para realizar web scraping. A continuación, se detallan los principales que tienen compatibilidad con el lenguaje de programación Python.

- **Beautiful Soup:** Es una librería de Python para extraer datos de archivos HTML y XML. Provee una amplia funcionalidad para navegar, buscar y modificar el árbol de análisis. Normalmente ahorra horas de trabajo a los programadores [12].

- **Scrapy:** Es un completo framework de código abierto que permite a los programadores de Python desarrollar web crawlers [13]. Posee una arquitectura

sólida, robusta y escalable que puede ser utilizada en proyectos de gran envergadura.

- **PyQuery:** Es una librería de web scraping con sintaxis similar a jQuery para Python. Permite realizar operaciones de forma cómoda y rápida utilizando la gramática jQuery [14]. Esto permite que los desarrolladores de jQuery tengan una opción para realizar web scraping aprovechando sus conocimientos anteriores. A la vez, PyQuery ofrece integración con LXML, permitiendo al desarrollador complementarse con los beneficios de esta librería para el análisis de documentos de mayor complejidad.

- **LXML:** Una librería para Python que permite procesar código HTML y XML. Posee un alto rendimiento en procesamiento lo que la hace atractiva para los desarrolladores. LXML tiene un tiempo de respuesta para extracción de contenido de 200 ms en condiciones críticas [15].

1.4.2 Large Language Models (LLMs)

Los Large Language Models (LLMs) son una colección de modelos de inteligencia artificial que se especializan en el procesamiento del lenguaje. Estos modelos se entrenan con grandes cantidades de datos, permitiéndoles realizar una amplia gama de tareas con una precisión notable. Entre las tareas que se pueden realizar con estos modelos se puede destacar el entendimiento del lenguaje, responder preguntas, resumir texto, identificar textos similares, entre otros. A medida que los LLMs continúan desempeñando un papel vital tanto en la investigación como en el uso diario, su evaluación en las tareas se vuelve cada vez más relevante. En los últimos años, se han realizado importantes esfuerzos para examinar los LLMs [16].

Entre los principales modelos desarrollados se puede mencionar:

- **Generative Pre-trained Transformer (GPT):** Es la arquitectura de modelo de lenguaje de aprendizaje profundo desarrollada por OpenAI. El modelo utiliza una técnica de aprendizaje basada en transformadores, que le permite aprender de grandes cantidades de datos y generar respuestas de alta calidad. El modelo presenta alta eficacia en la generación de texto y una eficacia media para tareas como traducción y resumen de texto [17].

- **Bidirectional Encoder Representations from Transformers (BERT):** Un modelo desarrollado por Google que tiene capacidad de aprendizaje en base a la palabra y

su contexto dentro de las oraciones. La eficacia del modelo es alta en comprensión de lectura y media en tareas de resumen de texto y respuestas a preguntas [18].

- **Text-To-Text Transfer Transformer (T5):** Un modelo desarrollado por Google AI que se basa en la arquitectura transformer de manera similar a GPT pero con modificaciones. T5 es entrenado con objetivos de eliminación de ruido y un enfoque a transferencia de texto a texto. Por lo que T5 es altamente efectivo en tareas que requieren transformar texto como traducir texto y generarlo. El modelo presenta eficacia media en tareas de resumen de texto y respuesta a preguntas [19].
- **Federated Adversarial Learning for Controllable Text Generation (FALCON):** Desarrollado por el Instituto de Innovación Tecnológica (IT) de los Emiratos Árabes Unidos. Es un modelo de código abierto que se basa en la arquitectura transformer pero con marcadas diferencias de T5 y GPT. Estas diferencias están enfocadas en la comprensión del significado del texto. Falcon, a diferencia de otros LLMs, ha sido entrenado con una gran cantidad de texto en idioma árabe. El modelo tiene una alta eficacia en generación de texto y una eficacia media en tareas de traducción y resumen de texto [20].

A continuación, se profundiza en el modelo BERT y sus variantes. LLM que será utilizado para la construcción de los modelos en el presente proyecto.

1.4.3 Bidirectional Encoder Representations from Transformers

BERT, acrónimo de Bidirectional Encoder Representations from Transformers, es un modelo de vectorización de última generación desarrollado en 2018 por Google [21]. Su principal característica reside en su capacidad para comprender bidireccionalmente el significado de las palabras en función del contexto en el que se encuentran. A diferencia de los modelos tradicionales, que generan representaciones fijas para cada palabra, BERT asigna vectores dinámicos que se adaptan al contexto específico.

Así por ejemplo en la frase “el cabeza de grupo”, la palabra “cabeza” adquiere un significado distinto al de la frase “se golpeó la cabeza”. El modelo BERT relaciona cada una de las palabras, de esta manera la palabra “cabeza” será, en cada caso, relacionada con cada una de las demás palabras identificando de esta manera su significado distinto. Esta capacidad del modelo BERT de poder interpretar las palabras en base a su contexto le dan una cualidad especial que puede ser utilizada en varios campos de estudio.

La arquitectura del modelo BERT está fundamentada en sus transformadores, que se basan en una red neuronal para poder procesar cadenas de texto.

BERT, al igual que otros modelos LLMs basa su funcionamiento en dos fases de proceso, el pre-entrenamiento y fine tuning.

1.4.3.1 Pre-entrenamiento

En la fase de pre-entrenamiento, BERT se entrena utilizando el modelo de lenguaje enmascarado (Masked Language Modeling MLM). Este modelo toma oraciones del corpus de entrenamiento y oculta aleatoriamente el 15% de las palabras. Luego, BERT intenta predecir las palabras ocultas utilizando el contexto de las palabras visibles restantes. Esto le permite a BERT aprender el significado y las relaciones entre las palabras en un contexto [22].

Tras el entrenamiento con el modelo de lenguaje enmascarado, BERT se enfrenta a un nuevo desafío: predecir la siguiente oración (Next Sentence Prediction NSP). Este proceso consiste en tomar dos oraciones conocidas y unirlas, creando una única secuencia de palabras. A continuación, BERT utiliza su conocimiento del lenguaje y el contexto de las dos oraciones para predecir la tercera oración que debería seguir.

En la fase de pre-entrenamiento, BERT necesita una gran cantidad de información para desarrollar su capacidad de comprender el lenguaje. Cuanta más calidad y cantidad de datos tenga, mejor será el modelo pre-entrenado.

El modelo original de BERT fue entrenado con dos conjuntos de datos en inglés:

- **BookCorpus:** Un conjunto de texto con más de 10.000 libros.
- **Wikipedia en inglés:** La enciclopedia en línea, con millones de artículos.

Estos conjuntos tienen un tamaño de 16 GB sin comprimir y proporcionaron a BERT una amplia base de conocimiento sobre el lenguaje [23].

1.4.3.2 Afinación (Fine Tuning)

El pre-entrenamiento de BERT es un proceso que genera un modelo genérico con una comprensión profunda del lenguaje, sin especialización en ninguna tarea específica. BERT por sí mismo es un modelo poderoso capaz de realizar diversas tareas, como traducción, redacción, respuesta a preguntas e incluso generación de código. La afinación es un paso posterior que orienta el modelo a una tarea específica, mejorando su precisión y eficiencia.

Las principales tareas en las que se puede utilizar el modelo son: clasificación de texto, respuesta a preguntas, resumen de textos, generación de texto, búsqueda de texto, entre otras.

1.4.4 Variantes del Modelo BERT

Algunas variantes han sido propuestas al modelo inicial de BERT, en las que se fomentan algunas mejoras [24]. A continuación, se mencionan las más importantes:

1.4.4.1 RoBERTa

Esta versión es más tolerante que BERT a errores y ruido en los datos al ser puesta a prueba. Corresponde a una versión mejorada en la que se ha pre-entrenado el modelo con 10 veces más datos que BERT original. Para el pre-entrenamiento de RoBERTa se ha considerado 160 GB de datos correspondientes a las siguientes fuentes [25]:

- **BookCorpus y Wikipedia en inglés:** Estos son los datos originales utilizados para entrenar BERT (16 GB).
- **CC-NEWS:** Los datos contienen 63 millones de artículos de noticias en inglés rastreados entre septiembre de 2016 y febrero de 2019 (76 GB después del filtrado).
- **OpenWebText:** Usa una recreación de código abierto del corpus WebText. El texto es contenido web extraído de URLs compartidas en Reddit con al menos tres votos a favor (38 GB).
- **STORIES:** Un conjunto de datos que contiene un subconjunto de datos de Common Crawl filtrados para que coincidan con el estilo de historia de los esquemas de Winograd (31 GB).

En el pre-entrenamiento de RoBERTa se ha eliminado la fase de predicción de la siguiente oración (NSP) y durante la fase de MLM se ha modificado este modelo para incluir máscaras dinámicas que cambian durante cada iteración de entrenamiento. Esto obliga a RoBERTa a seguir aprendiendo de la información contextual.

1.4.4.2 DistilBERT

El modelo de DistilBERT (BERT destilado) es un modelo de propósito general más pequeño, que ha sido entrenado mediante la transferencia de conocimientos desde BERT. Previamente el modelo BERT es reducido o “destilado” en un 40% de su tamaño, conservando el 97% de su eficacia y aumentando al 60% su eficiencia [26].

El modelo DistilBert fue desarrollado por Hugging Face y publicado por primera vez en 2019 [27]. Es mucho más ligero y rápido que BERT por lo que puede ser utilizado en aplicaciones en las que el rendimiento sea importante como las que se ejecutan en dispositivos móviles o dispositivos de borde.

1.4.4.3 ALBERT

El modelo Albert utiliza una técnica llamada factorización de matriz para reducir la dimensionalidad de la representación del lenguaje aprendida por BERT. Esto reduce el tamaño del modelo y lo hace más eficiente sin sacrificar precisión [28].

1.4.4.4 XLNet

XLNet se basa en un modelo de lenguaje AR (autorregresivo) bidireccional, mientras que BERT se basa en un modelo AE (codificador automático) bidireccional. En este sentido, XLNET está pre-entrenado en función de las posibles permutaciones de palabras de contexto que rodean una palabra objetivo. También, XLNET tiene en cuenta las dependencias entre palabras [29]. En el modelado en base a permutaciones se baraja el orden de las palabras dentro de la oración, posteriormente se predicen las palabras enmascaradas haciendo uso del modelo MLM haciendo uso del contexto permutado. Esta técnica que busca innovar en el modelo XLNet permite que el modelo aprenda de manera más profunda.

XLNet está basado en la tecnología Transformer-XL que se caracteriza por manejar oraciones largas de manera efectiva [29]. Esto permite que XLNet pueda ser utilizado para tareas que involucren grandes cantidades de contenido, por ejemplo resumir textos.

1.4.4.5 Sentence BERT Masked and Permuted Network (sBERT MPNET)

Sentence BERT una modificación del modelo BERT que utiliza redes siamesas y tripletes que pueden derivar en la generación de embeddings de oraciones semánticamente significativas que se pueden comparar mediante similitud cosenoidal. Esto permite utilizar BERT para determinadas tareas nuevas que hasta ahora no eran aplicables para BERT. Estas tareas incluyen comparación de similitudes semánticas a gran escala, clusterización y búsqueda semántica de información [30]. Esta versión modificada de BERT está diseñada principalmente para realizar comparativa y análisis de oraciones y tiene soporte para varios idiomas.

BERT adopta el modelado de lenguaje enmascarado (Masked Language Modeling MLM) para el pre-entrenamiento previo y es uno de los modelos más exitosos. XLNet introduce el modelado de lenguaje permutado (Permutative Language Modeling PLM) para incluir mejoras. Sin embargo, XLNet no aprovecha toda la información de posición de una oración y, por lo tanto, sufre una discrepancia de posición entre el pre-entrenamiento y el ajuste que BERT originalmente podría solventar. La variante basada en MPNET aparece como

un novedoso método de pre-entrenamiento que hereda las ventajas de BERT y XLNet y evita ambas limitaciones [31].

1.4.5 Modelos Supervisados de Aprendizaje de Máquina

El aprendizaje supervisado, ampliamente utilizado en tareas de clasificación y regresión, requiere que la máquina aprenda en base a varios ejemplos con entradas y salidas predefinidas [32]. Entre las ventajas de implementar este tipo de modelos destacan su alta exactitud en la tarea de predicción y flexibilidad durante la implementación. Su versatilidad permite entrenarlos independientemente de la naturaleza o giro de negocio al que vaya a enfocarse el modelo. Sin embargo, el principal inconveniente que se puede encontrar en este tipo de modelos radica en la necesidad de tener datos etiquetados que, dependiendo del giro de negocio, pueden ser de difícil obtención.

Las tareas en las que se puede utilizar este tipo de modelos son:

- **Clasificación:** Permite asignar un conjunto de características a una determinada categoría. Por ejemplo, el definir si un correo electrónico califica como spam o no.
- **Regresión:** Predice un valor numérico continuo a partir de las características. Por ejemplo, predecir el score que tendrá un titular en Internet.

1.4.5.1 Artificial Neural Networks (ANN)

Están inspiradas en el cerebro humano. Constituyen una red compleja de neuronas interconectadas entre sí. Una neurona calcula una salida utilizando una función de activación que considera la suma ponderada de todas sus entradas [33]. La función de activación juega un papel importante en el entrenamiento y el desempeño de una Red Neuronal Artificial. Estas funciones le aportan las propiedades no lineales necesarias para poder desarrollar sus tareas [34].

Existe una vasta cantidad de funciones de activación. A continuación, se detallan las más utilizadas:

- **Función sigmoide:** Utilizada con frecuencia para clasificaciones binarias. Una función en forma de S.
- **Función hiperbólica:** Utilizada con frecuencia en regresiones. Una función similar a la sigmoide pero restringida a un rango de -1 y 1.
- **Rectified Linear Unit (RELU):** Esta función es 0 para valores negativos y la identidad para valores positivos. Es ampliamente utilizada por su eficiencia de cálculo y resultados efectivos.

- **Gaussian Error Linear Unit (GELU):** Esta función de activación se ha convertido en un método dominante, superando a funciones tradicionales como Rectified Linear Unit (RELU) [35]. La función GELU es muy similar a RELU pero su incremento para valores positivos así como su caída en los valores negativos son más suaves, lo que permite una mejor adaptación al modelo. La función GELU puede ser definida como:

$$GELU(x) = x\,\varphi(x) \tag{1.1}$$

donde $\varphi(x)$ corresponde a la función de distribución acumulativa Gaussiana.

La Figura 1.1, muestra la función RELU y la función GELU graficadas. Se puede apreciar sus diferencias.

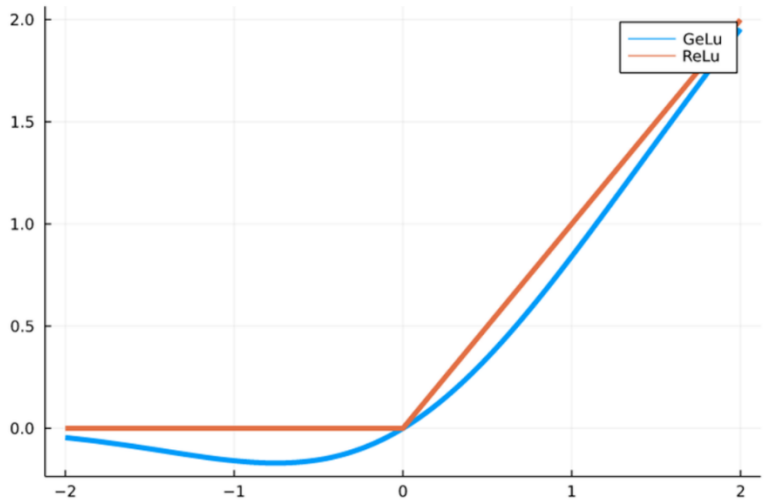

Figura 1.1. Funciones RELU y GELU [36]

1.4.5.2 K Nearest Neighbors (KNN)

KNN es un algoritmo de alto rendimiento que calcula los vecinos más cercanos similares a un registro determinado [37]. KNN basa su proceso en un mapa de los datos de entrenamiento establecidos en un espacio de distancia unidimensional basado en las similitudes calculadas, para luego etiquetar la consulta según la etiqueta más dominante o promedio de los k vecinos más cercanos, y posteriormente calcular la regresión o clasificación según sea necesario [38].

KNN es un algoritmo que no necesariamente requiere entrenamiento, por lo que puede ser utilizado para tareas de imputación de valores, así como clasificación y regresión. Por lo que KNN es considerado como un algoritmo altamente versátil.

KNN es sensible a la elección de k, la dificultad en su implementación radica en la selección de los k vecinos que debe identificar el algoritmo.

1.4.5.3 Support Vector Machine (SVM)

SVM construye un hiperplano o un conjunto de hiperplanos en un espacio de alta o infinita dimensión, que puede usarse para clasificación, regresión u otras tareas [39]. El hiperplano construido debe ser tal que se maximice adecuadamente la distancia entre puntos de una clase respecto de otra. El vector de soporte comprende a los puntos más cercanos, de ambas clases, al hiperplano que los separa. Estos puntos son importantes ya que definen el hiperplano que se utilizará.

Cuando se agrega un nuevo punto al sistema, el modelo está en capacidad de definirlo en una de las clases haciendo uso del hiperplano definido. Esto permite que el algoritmo se utilice de forma eficaz para tareas de regresión y clasificación.

SVM hace uso de una función kernel previamente definida. La función kernel permite al modelo aprender relaciones lineales y no lineales entre las características de los datos. La función de kernel opera sobre el espacio vectorial realizando operaciones que permiten separar y agrupar los puntos [40]. Entre las principales funciones de kernel que se pueden utilizar para la construcción de un modelo SVM se puede mencionar:

- **Lineal:** Un kernel simple y eficiente computacionalmente. Opera con datos linealmente separables entre sí.
- **Polinómico:** Aplica funciones polinómicas a un grado definido. Es sensible al grado que se define pues un grado no adecuado puede producir overfitting.
- **Radial Basis Function (RBF):** Crea un espacio de alta dimensión en base a las características. Es uno de los más populares debido a su alta versatilidad y rendimiento.
- **Sigmoide:** Similar al polinómico, pero hace uso de la función sigmoide para la separación.

1.4.6 Modelos No Supervisados de Aprendizaje de Máquina

Trabajar con métodos no supervisados implica no tener un atributo etiqueta o clase con valores predefinidos en el conjunto de datos. Entonces, los algoritmos agrupan los datos dentro de un número de clases preestablecidas o no, partiendo de criterios de distancia o similitud. De esta forma se agrupan las clases que sean similares entre sí y distintas con las otras clases [41].

El aprendizaje no supervisado basa su funcionamiento en la detección de patrones o características comunes en base al análisis estadístico. Es ampliamente utilizado en tareas de clusterización, detección de anomalías, detectores de plagio y uso de material con copyright, entre otros.

1.4.6.1 Clusterización

El propósito de las tareas de clusterización es el de agrupar un conjunto de datos en grupos más pequeños (clústers), basándose en características comunes determinadas. Los objetivos de las tareas de clusterización son muy variados como por ejemplo segmentación de clientes, agrupación de documentos, identificación de imágenes similares, entre otras.

1.4.7 Elasticsearch

Elasticsearch es un motor de búsqueda y analítica de datos de código abierto creado por Shay Banon y publicado en febrero de 2010 [42]. Elasticsearch ha evolucionado desde un proyecto de búsqueda de texto simple hasta un sistema de análisis de datos complejo [43]. Elasticsearch tiene una arquitectura horizontal altamente escalable, lo que significa que se puede añadir más nodos para seguir atendiendo las necesidades de crecimiento de usuarios del negocio. A la vez, permite realizar búsquedas de baja latencia proveyendo un alto rendimiento a las aplicaciones.

Elasticsearch está basado en la biblioteca de software Apache Lucene, una poderosa librería para indexar y buscar texto. Elasticsearch almacena la información a manera de documentos. Es decir, no debe confundirse con una base de datos relacional ya que Elasticsearch no tiene la capacidad de relacionar los documentos por sí misma.

Este enfoque permite buscar información de manera eficiente, donde los datos sin esquema, en forma de documentos JSON arbitrariamente complejos, se pueden entregar a Elasticsearch para su indexación. Los resultados de las búsquedas son proporcionados mediante una interfaz RESTful [44].

Elasticsearch está estructurado de varios componentes que trabajan de manera coordinada para proporcionar un servicio de búsqueda y análisis indexado de manera distribuida. Entre los componentes de Elasticsearch se encuentran:

- **Nodos:** Es la unidad básica de un clúster. Cada nodo ejecuta una instancia del servicio Elasticsearch con todas las funciones de almacenamiento, indexación y comunicación con los demás nodos del clúster.

- **Clúster:** Conjunto de nodos que se interconectan entre sí para trabajar como un sistema unificado. Su uso permite la escalabilidad horizontal del sistema.
- **Índices:** Es un almacén de documentos, en el cual se registran en partes fraccionadas llamadas Shard.
- **Shard:** Particiones horizontales de un índice que se distribuyen entre los nodos del clúster. Los shards son el núcleo de la arquitectura de Elasticsearch [45].
- **Réplica:** Son copias de un shard que se almacenan en diferentes nodos de un clúster para garantizar la redundancia.
- **Documentos:** Objetos JSON que contienen la información específica.
- **Mappings:** Metadatos que definen la estructura de los documentos en el índice.

Elasticsearch puede lograr respuestas de búsqueda rápidas porque, en lugar de buscar el texto directamente, busca un índice [46]. Las búsquedas en Elasticsearch se basan en un lenguaje de dominio específico (Domain Specific Language DSL) cuyas consultas son de sintaxis sencilla y están estructuradas en formato JSON.

Elasticsearch posee integración a herramientas de machine learning permitiendo aplicar técnicas de procesamiento de lenguaje natural (Natural Language Processing NLP), almacenamiento y búsqueda vectorial. Además de soportar algoritmos como KNN para realizar búsquedas, incrementando su rendimiento en base a su arquitectura indexada. Elasticsearch cuenta con un sistema de score, el cual además de computar los resultados óptimos para la consulta DSL permite obtener el grado de correlación entre las tramas de texto y/o vectores consultados [47]. Es por estas razones que la herramienta Elasticsearch es ideal para realizar la fase de afinación (fine tuning) de los LLMs.

1.5 Revisión de Literatura

Existen algunos casos de estudio previos que abordan la temática de la clusterización de titulares y puntuaciones. A continuación, se detallan los más relevantes.

1.5.1 Clusterización de Titulares

Para el uso de LLMs para conformación de clústers y análisis asociados de su eficacia se pueden citar los siguientes estudios de relevancia.

- Kurisinkel & Chen [48], hacen uso de un algoritmo de clusterización basado en una función submodular monotónica para extraer los eventos de mayor importancia. En base a estos eventos realiza la clusterización y posteriormente hace un resumen

haciendo uso de técnicas LLM basadas en indicadores ROUGE (Recall-Oriented Understudy for Gisting Evaluation). Para resumir el contenido del clúster su método implica tres pasos principales: extracción del evento principal, extracción de contexto y reescritura. La evaluación la realizan utilizando métricas objetivas y evaluadores humanos. Así se afirma la eficacia de su enfoque, ya que indican han superado las líneas de base potenciales, demostrando excelencia tanto en el contenido, cobertura, coherencia e informatividad.

- Fatemi et al. [49], emplean la plataforma Expert.AI para clasificar las noticias en base al LLM GPT. La plataforma Expert.AI tiene modelos entrenados basados en el marco de noticias estructurados IPCT (International Press Communication Council). Afirman que el uso de lenguajes LLM pre-entrenados como GPT tienen un potencial significativo para mejorar la eficiencia y precisión en la tarea de clasificación de titulares. Destacan la importancia de explorar y aprovechar las capacidades de estos modelos para automatizar estas tareas en el campo del periodismo y la investigación de medios. Su estudio de clasificación es del tipo supervisado, la etiqueta conocida previamente corresponde a la IPCT que es un estándar de la industria que abarca titulares por temática. El mecanismo de evaluación de la clasificación que utiliza es la comparación en base al recall, f1 score y precisión.
- Nakshatri et al. [50], proponen un método propio de clusterización que está basado en la detección de eventos clave en las noticias construido con el uso del LLM GPT. Este método resume cada noticia mediante el LLM GPT para identificar los eventos clave. Posterior, se clusteriza las noticias en base a estos resultados. Para la evaluación utilizan tres métricas: pureza de la entidad, cobertura y coherencia de la entidad. Estas métricas tienen por objetivo ayudarles a medir el grado de coherencia con el que se ha formado el clúster. Para la implementación utilizan el API de pago de OPEN AI.
- Zhou et al. [51], realizan un análisis comparativo del LLM GPT, BERT clásico y diversas variantes de BERT para comparar su entendimiento del lenguaje y el grado de sentido común que puede llegar a tener cada uno de estos LLM. De su análisis se puede concluir que mientras GPT se muestra superior en resultados a BERT clásico, las variantes del modelo BERT como XLNet y RoBERTa tienen mejores resultados que GPT. Este estudio nos permite tener un enfoque de las bondades que se puede percibir de cada modelo.

El presente proyecto propone un modelo de clusterización, que se fundamenta en la detección de patrones al considerar varias métricas. Las métricas tomadas en cuenta son la similitud patrones de texto y la similitud vectorial haciendo uso del LLM sBERT MPNET, variante mejorada del LLM BERT. La clusterización se realiza a nivel del título de noticia. Esto se debe a que el titular es la primera fuente sintetizada de información de una noticia. Se busca ir más allá operando sobre los clústers formados para puntuar el contenido e identificar el que sea de mayor relevancia.

1.5.2 Sistema de Puntuación

Existen algunos estudios sobre sistemas de puntuación que hacen uso de machine learning para predecir el éxito de contenido en Internet, algunos de ellos enfocados en titulares de noticias. Se pueden citar los siguientes:

- Wu [52], propone un sistema de recomendaciones que hace uso de KNN para hacer ranking de comentarios de usuario, basándose en la similitud cosenoidal al ser transformados sobre un modelo LLM BERT customizado. Concluyen resultados satisfactorios al implementar su modelo tras realizar búsquedas KNN para obtener los mejores scores para recomendación.
- Jääskeläinen et al. [53] proponen la medición del ranking de noticias mediante regresiones lineales sobre modelos neuronales basados en métricas simples como las visitas de las páginas por los suscriptores, las vistas de las páginas por lectores que no están suscritos y el número de suscriptores que se ha ganado en consecuencia de la publicación de la historia. Este modelo se basa en el número de visitas y asume la medida del éxito de una noticia a esta métrica única. Para el entrenamiento del modelo se hace uso de un dataset de datos internos de una empresa de comunicación colaboradora.
- Tsagkias et al. [54] proponen una predicción basada en el número de comentarios de los usuarios sobre las noticias publicadas. Su estudio aborda la posibilidad de predecir la popularidad de una noticia en base al volumen de comentarios que los usuarios generan sobre ella. Para ello, clasifican las noticias en base a criterios de probabilidad de recepción de comentarios y probabilidad de volumen de comentarios en dos métricas (bajo volumen y alto volumen). El análisis se basa en una predicción usando estas dos fases de clasificación.
- Szabo & Huberman [55] abordan la posibilidad de puntuar contenido en Internet en las primeras horas en que ha sido publicado. Esto permite predecir la popularidad que el contenido generará en los siguientes días. Proponen un enfoque en la

popularidad a largo plazo del contenido en línea a partir de mediciones tempranas del acceso. Utilizan dos portales para compartir contenido, Youtube y Digg, y aseguran que al modelar la acumulación de vistas y votos sobre el contenido ofrecido por estos servicios se puede predecir la dinámica a largo plazo de los envíos individuales a partir de los datos iniciales.

- McCreadie et al. [56] hacen uso de la blogosphere para pronosticar la importancia de las noticias. Para ello, clasifican en base a un marco de aprendizaje las noticias en tiempo real.
- Gruhl et al. [57] utilizan varias fuentes de contenido de Internet como libros, blogs, páginas web, entre otros; para predecir el ranking en ventas en base a consultas y la identificación de la tendencia de los picos de ventas de su análisis. La popularidad del contenido está basada en el número de ventas.

El presente proyecto aborda un enfoque distinto de ranking que se basa en una métrica compuesta que está definida por la reputación del titular en Internet y el análisis de relevancia del contenido de un titular en comparación a los demás dentro de un clúster. Hace uso de varias técnicas como análisis de similitud textual, vectorial y otros factores combinados que ayuden a identificar la reputación de un titular en Internet. Posteriormente se construye una métrica compuesta que permite identificar el contenido de mayor relevancia de una noticia en un determinado clúster.

2. METODOLOGÍA

Para el proyecto propuesto se hará uso de la metodología Cross Industry Standard Process for Data Mining (CRISP-DM).

CRISP-DM es una metodología de minería de datos que fue desarrollada a mediados de la década de 1990 por un consorcio de empresas europeas para servir como marco no propietario para proyectos en esta área [58]. La metodología se ha convertido en un estándar de facto en la industria de la minería de datos independientemente del tema del proyecto [59].

La metodología CRISP-DM permite regresar a pasos anteriores lo que la hace una herramienta altamente flexible, de esta manera los resultados del proyecto pueden refinarse de modo gradual. Dependiendo de los resultados obtenidos entre las distintas fases se puede volver a pasos anteriores para realizar modificaciones y/o correcciones [60].

La metodología CRISP-DM ha sido seleccionada por ser una metodología estandarizada abierta de minería de datos que tiene la ventaja de poder adaptarse a distintos proyectos por su gran flexibilidad. El proyecto propuesto abarca varias fases de minería de datos, por lo que una metodología flexible, como CRISP-DM, es recomendable.

La metodología CRISP-DM está compuesta por seis etapas [61]:

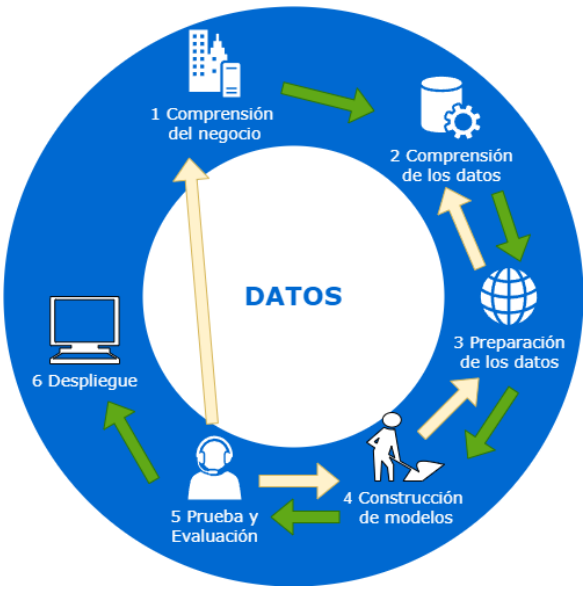

Figura 2.1. Etapas del modelo CRISP-DM [62]

- **Comprensión del negocio:** Abarca un estudio inicial del modelo de negocio y la problemática a abordar. Esta etapa es fundamental ya que durante su transcurso se establecen las bases para el proyecto de minería de datos. Su objetivo principal es comprender a fondo lo relacionado al modelo de negocio. El interés en la etapa de comprensión del negocio debe enfocarse en asegurar al proyecto una sólida integración al contexto del negocio para que se adapte a la apreciación de la dinámica local y el modelo tenga una utilidad evidente.
- **Comprensión de los datos:** Se enfoca en la exploración de los datos para identificar la información relevante que ayude a alcanzar los objetivos planteados. En esta etapa el analista de datos se familiariza con la estructura de los datos para que, en base a la comprensión del negocio anteriormente realizada, pueda abordar el problema de manera objetiva.
- **Preparación de los datos:** Los datos son preparados mediante su limpieza y transformación para permitir mayor facilidad durante el tratamiento en las siguientes etapas. Esta etapa puede abarcar varias actividades como: imputación de valores faltantes, codificación de variables categóricas, eliminación de características, normalización de datos, entre otras.
- **Construcción de modelos:** En esta etapa se seleccionan los algoritmos para construir los modelos de datos. Posteriormente se realiza el entrenamiento de los modelos con los datos que fueron preparados en la etapa previa.
- **Prueba y evaluación:** Se pone a prueba los modelos diseñados teniendo como entrada el conjunto de datos. En caso de existir aspectos omitidos o no considerados se identifican en esta etapa para su corrección. La ejecución del modelo depende de los algoritmos seleccionados y los modelos se pueden evaluar con métricas para revisar si cumplen los objetivos planteados o si requieren ser corregidos [63]. En caso de haberse realizado varios modelos se selecciona el que tenga la mejor evaluación en base a las métricas de resultados obtenidas en esta etapa.
- **Despliegue:** Esta etapa puede abarcar desde la documentación hasta la puesta en producción dependiendo del entorno y giro de negocio en el que se implemente la metodología.

A continuación, en las siguientes secciones, detallaremos cómo cada una de estas etapas se instancian en el proyecto.

2.1 Comprensión del negocio

El vertiginoso crecimiento de los diarios electrónicos ha transformado el panorama informativo, inundando el mundo con una gran cantidad de titulares. Si bien el acceso a la información se ha democratizado gracias a Internet, la proliferación de fuentes y la similitud de las publicaciones presentan desafíos para los lectores: la síntesis y el análisis crítico de la información.

La creciente tendencia a consumir noticias de una sola fuente, generalmente la de mayor preferencia personal, limita la exposición a diversas perspectivas y puede generar una visión fragmentada de la realidad.

En vista de estas necesidades y de las técnicas ya existentes analizadas, se plantea la implementación de una arquitectura de sistema que sea apoyo para lograr las metas del proyecto. La Figura 2.2 muestra la arquitectura del sistema.

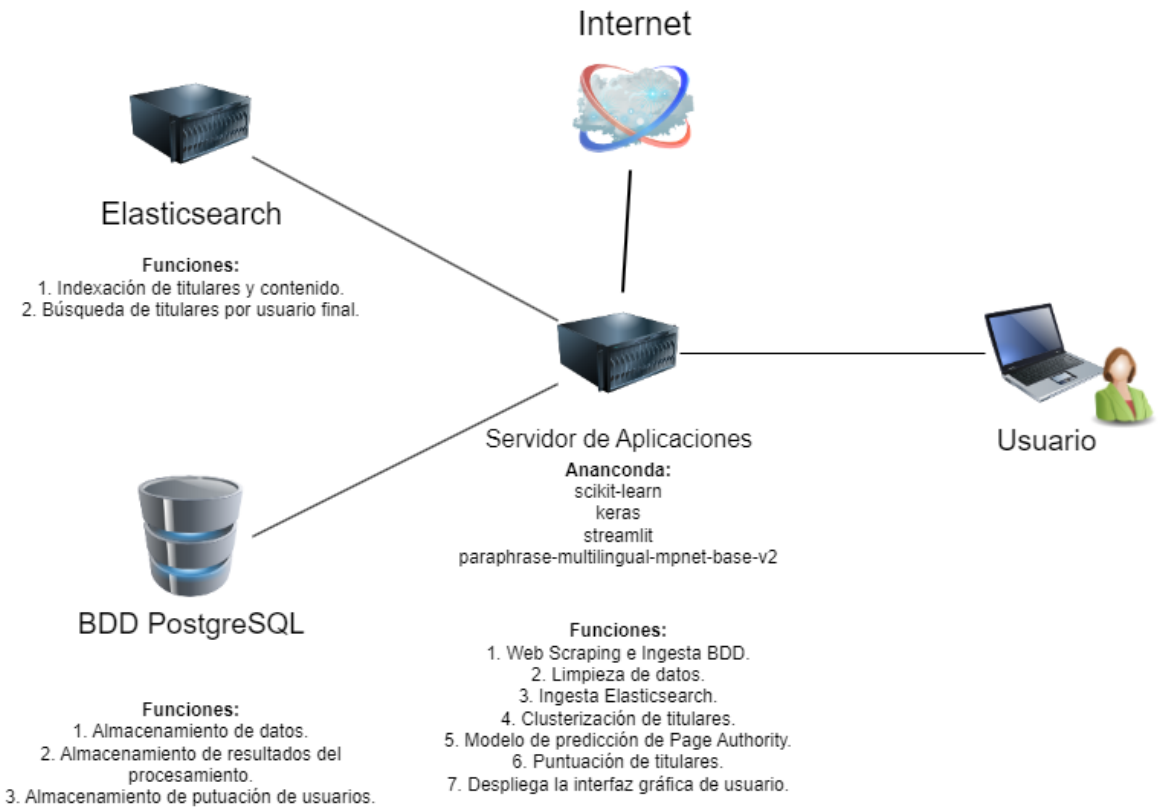

Figura 2.2. Componentes del sistema (elaborado por el autor)

Para la gestión de datos se ha incluido en la arquitectura una base de datos relacional implementada con el motor open-source PostgresSQL, que se encarga de almacenar y organizar la información de forma eficiente. También se integra una instancia de servicio Elasticsearch que funciona como herramienta de análisis para el procesamiento y

búsqueda para el usuario final, permitiendo una exploración profunda de los datos con tiempos de latencia bajos.

El procesamiento se realiza en un servidor con el framework Anaconda 3. El modelo "paraphrase-multilingual-mpnet-base-v2", una variante de BERT basada en sBERT MPNET, se ha cargado en este servidor. Este modelo genera vectores de 768 dimensiones, donde la relación cosenoidal refleja el significado de la frase. Es importante destacar que esta versión del modelo soporta 50 idiomas entre ellos el idioma español. La Tabla 2.1 muestra las características del modelo paraphrase-multilingual-mpnet-base-v2.

Tabla 2.1. Características del modelo paraphrase-multilingual-mpnet-base-v2 [64]

Modelo Base:	Maestro: paraphrase-mpnet-base-v2; Estudiante: xlm-roberta-base
Máxima longitud de secuencia:	128
Dimensión del vector:	768
Usa Normalización:	No
Funciones Factibles de Puntuación:	Similitud cosenoidal (util.cos_sim)
Tamaño:	970 MB
Datos de entrenamiento:	Modelo multilenguaje variante de paraphrase-mpnet-base-v2, extendido a 50 idiomas

El modelo paraphrase-multilingual-mpnet-base-v2 se encuentra disponible en [64] para su descarga. Este modelo está basado en el modelo paraphrase-mpnet-base-v2 que originalmente fue diseñado para soportar el idioma inglés. Esta variante difiere en su soporte multiidioma y permite generar vectores que pueden ser asociados mediante similitud cosenoidal.

Para la gestión de la interfaz gráfica de usuario se implementa la librería streamlit para Python. Streamlit es una librería de código abierto para la construcción de aplicaciones web interactivas.

A continuación, se detalla la estrategia de implementación de la metodología, así como los insumos de entrada y productos de salida de cada una de las etapas que serán abordadas. Esto nos permite tener un mejor enfoque de cómo se abordará el problema en cuestión.

2.1.1 Recopilación y Almacenamiento de Datos

La Figura 2.3, muestra las fases que se han seguido para la recopilación y almacenamiento de datos. Las fases deben seguirse de manera ordenada ya que la salida de la fase anterior

será entrada de la siguiente fase. Una vez culminada las fases de recopilación y almacenamiento de datos se debe proceder con las fases de procesamiento de datos.

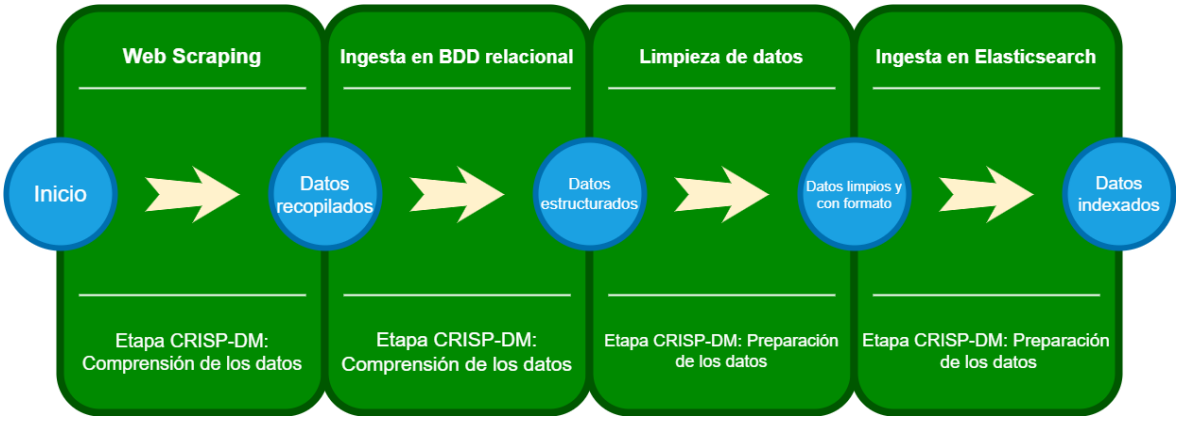

Figura 2.3. Fases de la recopilación y almacenamiento de datos (elaborado por el autor)

Las fases que se contemplan corresponden a las actividades que deben ser realizadas durante la etapa de comprensión y preparación de los datos de la metodología CRISP-DM. Las fases consideradas son:

- **Web Scraping:** En esta fase se recopila las noticias electrónicas de las páginas web correspondientes a las editoriales.
- **Ingesta en BDD relacional:** Los datos son filtrados de su formato original y almacenados de manera organizada en la base de datos relacional.
- **Limpieza de datos:** Los datos son preparados mediante una limpieza en la que se explora y aborda las posibles falencias.
- **Ingesta en Elasticsearch:** Los datos son indexados en la instancia Elasticsearch para un mayor rendimiento durante su análisis, esto por cuanto se operará con vectores de 768 dimensiones.

2.1.2 Procesamiento de Datos

La Figura 2.4, muestra las fases que se han seguido para el procesamiento de datos. Estas fases deben realizarse de manera ordenada ya que la salida de la fase anterior es la entrada de la siguiente fase. Se contemplan tres componentes que requieren desarrollo para el análisis de los datos. Estos son:

- **Componente de clusterización:** Permite agrupar las noticias por titular.
- **Componente de predicción de noticias en base a sus parámetros:** Permite predecir el grado de éxito de la noticia en base a regresiones.

- **Componente de puntuación:** Permite puntuar el contenido de cada línea de noticia en el clúster, para de esta manera poder extraer el contenido relevante dentro de un determinado clúster.

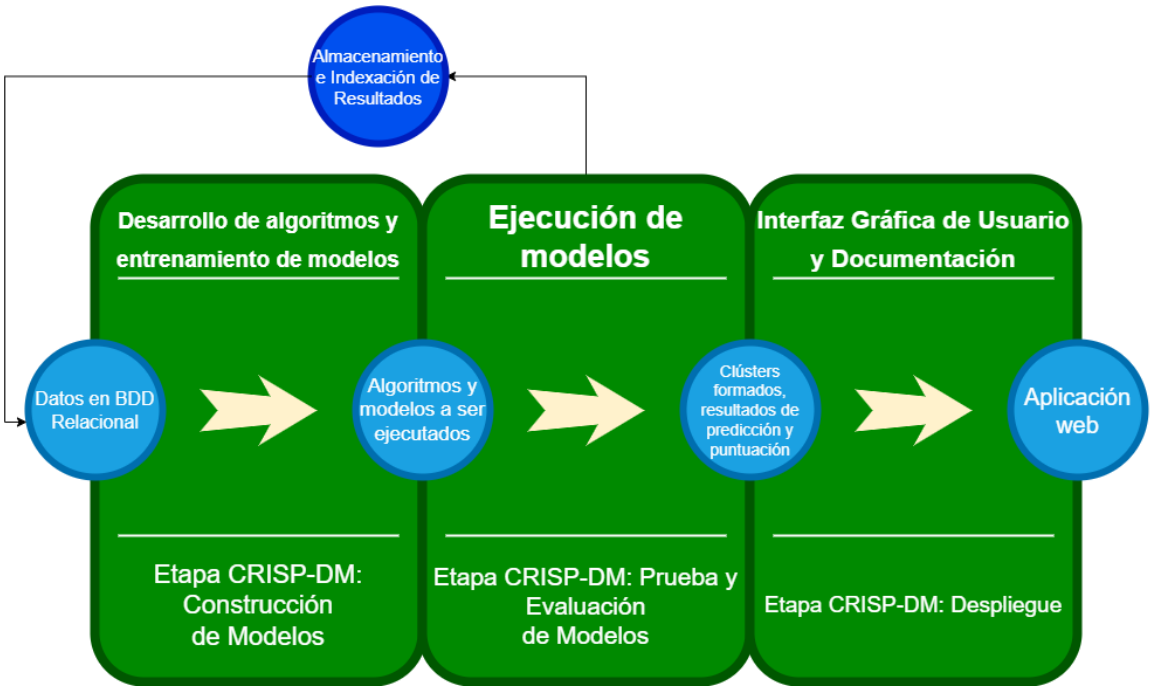

Figura 2.4. Fases del procesamiento de datos (elaborado por el autor)

Las fases de procesamiento descritas en la Figura 2.4 deben ser realizadas para cada uno de los componentes a desarrollarse. Estas fases culminan con el despliegue. Las fases contempladas son:

- **Desarrollo de algoritmos y entrenamiento de modelos:** Se construyen los algoritmos necesarios y se entrenan los modelos para cumplir los objetivos de cada componente.
- **Ejecución de modelos:** Se ejecuta cada modelo. Si los resultados son satisfactorios se almacenan en la base de datos y la instancia Elasticsearch. Los resultados obtenidos pueden ser entrada del siguiente componente. Un ejemplo claro son las agrupaciones procedentes del componente de clusterización, que son utilizadas posteriormente para delimitar el ranking del componente de puntuación.
- **Interfaz Gráfica de Usuario y Documentación:** Abarca la construcción de una interfaz para poder observar las noticias procesadas y puntuarlas. Además, se documenta el proyecto realizado.

2.2 Comprensión de los Datos

Se realizaron las tareas de recopilación y almacenamiento de los datos. Los datos fueron catalogados identificando cuales son requeridos de ser almacenados por ser relevantes para el estudio.

2.2.1 Recopilación de Titulares

Se realizó una recopilación de 49.630 titulares de 24 editoriales de Ecuador, Argentina y España. Los titulares se almacenaron en una base de datos relacional.

Para la recopilación se utilizó web scraping para los diarios ecuatorianos y argentinos, y un dataset obtenido de [65] para los titulares de España. La combinación de estos métodos nos permite tener un dataset final amplio y diverso para el proyecto, lo que nos permitirá alcanzar los objetivos.

2.2.1.1 Web Scraping

El primer paso consistió en la obtención de las URLs de las páginas web que contenían las noticias de interés. Estas URLs se almacenaron en una base de datos para su posterior procesamiento.

Luego, se procedió a la descarga y preprocesamiento de cada una de las URLs. Este proceso permitió extraer el contenido de las noticias y almacenarlo en la base de datos de forma organizada y accesible.

Mediante un análisis de las páginas web y su contenido de noticias electrónicas se pudo identificar las características que podían ser utilizadas. Estas características se detallan en la Tabla 2.2.

El preprocesamiento consistió en analizar el código HTML de las páginas descargadas para separar la información objetivo. Para ello, se utilizó la técnica de web scraping de selectores CSS. El contenido HTML de los diarios electrónicos descargados fue filtrado mediante la identificación de sus etiquetas CSS. Esto permitió obtener el contenido filtrado de las distintas partes que conforman la noticia.

Las líneas de contenido de cada noticia recolectada fueron separadas para su posterior análisis y puntuación. Este proceso permitió descomponer el texto en unidades más pequeñas y manejables.

Tabla 2.2. Información objetivo del web scraping (elaborado por el autor)

Característica	Descripción
Titulo	Título de la noticia (encabezado).
Autor 1	Autor de la notica.
Autor 2	Coautor de la notica en caso de existir.
Año	Año de publicación.
Mes	Mes de publicación.
Día	Día de publicación.
Hora	Hora de publicación.
Minuto	Minuto de publicación.
Diario	Diario que ha publicado la noticia (editorial).
Contenido	Contenido de las noticias, separado por líneas.

El web scraper se implementó utilizando la librería request de Python para la descarga de las noticias electrónicas. Para el preprocesamiento del contenido HTML se utilizó la librería Beautiful Soup, por su alta versatilidad en el procesamiento de este tipo de contenido.

La elección de estas librerías permitió realizar un proceso eficiente de extracción y preprocesamiento de las noticias, sentando las bases para el análisis posterior.

2.2.1.2 Dataset Adicional

Un dataset adicional, compuesto por noticias sobre economía y temas generales de España, fue fusionado con las noticias descargadas mediante web scraping. Este dataset, en formato CSV, fue cargado en la base de datos e integrado con las demás noticias utilizando sentencias SQL.

Esta integración permitió ampliar y diversificar el conjunto de datos disponible para el proyecto, enriqueciendo el análisis posterior.

2.2.2 Diseño de Base de Datos

Se ha optado por una base de datos relacional para el almacenamiento de datos, debido a la gran cantidad de información que se requiere analizar y las múltiples fases que comprende el proceso de análisis.

La base de datos nos permite recopilar los resultados obtenidos en cada fase y acceder a ellos en las fases posteriores, facilitando el seguimiento del proceso y la obtención de resultados finales.

La base de datos diseñada se compone de dos conjuntos de tablas. Las primeras, utilizadas durante la recolección, almacenan la información preprocesada descargada de las fuentes web. Sobre estas tablas también se ejecutaron las tareas de limpieza. Las segundas, con un diseño más sofisticado, contienen la información codificada para su análisis posterior. Estas tablas ya manejan conceptos de integridad referencial y hacen uso de las tablas de catálogos. La Figura 2.5 ilustra este diseño, detallando los elementos que conforman la base de datos: tablas de recolección (noticias, lineas_noticias) y tablas con información codificada (noticias_codificado, lineas_noticias_codificado, catálogos, y demás). Este diseño permite un almacenamiento eficiente y flexible de la información, facilitando el acceso y la manipulación de los datos durante las diferentes fases del proyecto.

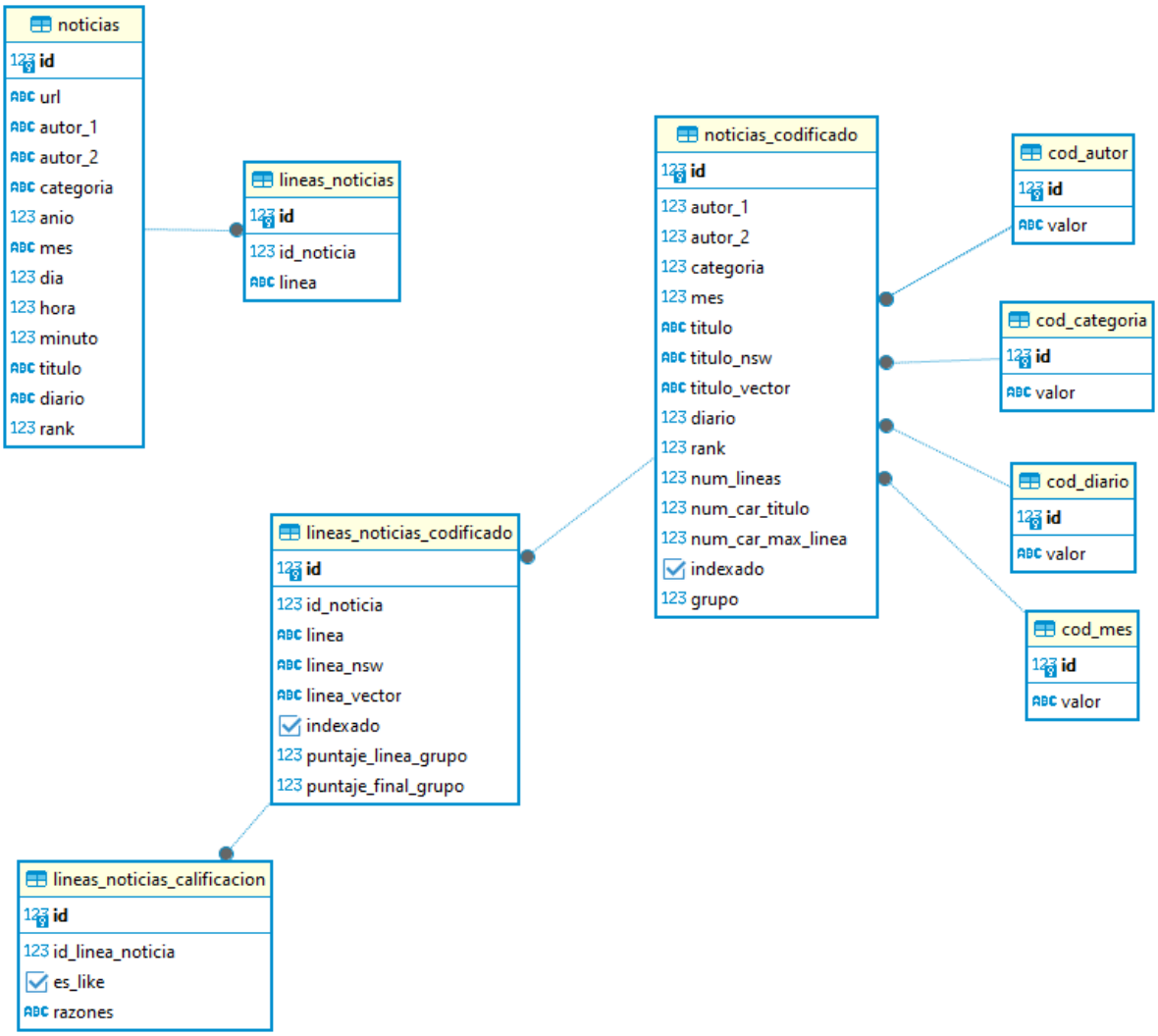

Figura 2.5. Diseño de base de datos relacional – BDD PostgreSQL (elaborado por el autor)

2.2.3 Diccionario de Datos

Esta sección describe las diferentes tablas y campos que conforman la base de datos relacional, incluyendo su descripción y las funciones para las cuales fueron creadas.

2.2.3.1 Tabla NOTICIAS

La tabla NOTICIAS almacena los datos de los titulares de las noticias, con un registro por cada noticia publicada. Los campos considerados se muestran en la Tabla 2.3. Esta tabla se llena mediante el proceso de web scraping.

Tabla 2.3. Campos de la tabla NOTICIAS (elaborado por el autor)

Campo	Tipo	Descripción/Función
ID	Numérico autogenerado	Clave primaria, identificador único de noticia.
URL	Texto	Dirección URL de la noticia.
AUTOR_1	Texto	Primer autor de la noticia.
AUTOR_2	Texto	Coautor de la noticia en caso de existir.
CATEGORIA	Texto	Categoría de la noticia. Por ejemplo: tecnología, economía, política, etc.
ANIO	Numérico	Año de publicación.
MES	Texto	Mes de publicación.
DIA	Numérico	Día de publicación.
HORA	Numérico	Hora de publicación.
MINUTO	Numérico	Minuto de publicación.
TITULO	Texto	Título de la noticia.
DIARIO	Texto	Corresponde a la editorial de entre las 24 consideradas en la que fue publicado el artículo.
RANK	Numérico	Page Authority de la página en Internet. Un valor entre 0 y 100 que determina el grado de popularidad del artículo en Internet. Ha sido recolectado de forma manual de [66].

2.2.3.2 Tabla LINEAS_NOTICIAS

La tabla LINEAS_NOTICIAS almacena el contenido de cada noticia separado por líneas. Tiene una relación foránea a la tabla de noticias. Esta tabla se llena durante la fase de web scraping. La Tabla 2.4 muestra los campos de la tabla LINEAS_NOTICIAS.

Tabla 2.4. Campos de la tabla LINEAS_NOTICIAS (elaborado por el autor)

Campo	Tipo	Descripción/Función
ID	Numérico autogenerado	Clave primaria, identificador único de línea de contenido.
ID_NOTICIA	Numérico	Identificador único de noticia. Relación foránea al campo ID de la tabla NOTICIAS.
LINEA	Texto	Línea de contenido de noticia.

2.2.3.3 Tabla NOTICIAS_CODIFICADO

La tabla NOTICIAS_CODIFICADO es una versión limpia y codificada de la tabla NOTICIAS, basada en catálogos predefinidos. Se crea durante la fase de limpieza de datos y, además de los campos de la tabla original, incluye metadatos sobre la noticia y otros campos útiles para el procesamiento, almacenamiento y gestión de los resultados obtenidos en las diferentes fases del proyecto. A la vez, en esta tabla se han eliminado los campos que no son de utilidad para el análisis. La Tabla 2.5 muestra los campos que comprenden esta tabla.

2.2.3.4 Tabla LINEAS_NOTICIAS_CODIFICADO

La tabla LINEAS_NOTICIAS_CODIFICADO se presenta como una evolución depurada y limpia de la tabla LINEAS_NOTICIAS. Su diseño meticuloso incorpora campos específicos para el procesamiento, almacenamiento y gestión eficiente de los resultados relacionados con las líneas de contenido de cada artículo. La Tabla 2.6 muestra los campos que la conforman.

2.2.3.5 Tabla LINEAS_NOTICIAS_CALIFICACION

La tabla LINEAS_NOTICIAS_CALIFICACION almacena las puntuaciones ingresadas por el usuario a través de la interfaz gráfica de usuario. La Tabla 2.7 muestra los campos considerados en el diseño de esta tabla.

Tabla 2.5. Campos de la tabla NOTICIAS_CODIFICADO (elaborado por el autor)

Campo	Tipo	Descripción/Función
ID	Numérico	Clave primaria, identificador único de noticia. El valor corresponde al de la tabla original NOTICIAS.
AUTOR_1	Numérico	Autor de la noticia codificado. Relación foránea al campo ID de la tabla catálogo COD_AUTOR.
AUTOR_2	Numérico	Coautor de la noticia codificado en caso de existir. Relación foránea al campo ID de la tabla catálogo COD_AUTOR.
CATEGORIA	Numérico	Categoría de la noticia codificada. Relación foránea al campo ID de la tabla COD_CATEGORIA.
MES	Numérico	Mes de publicación. Relación foránea al campo ID de la tabla COD_MES.
TITULO	Texto	Título completo de la noticia.
TITULO_NSW	Texto	Título de la noticia con remoción de las palabras vacías (stop words).
TITULO_VECTOR	Texto	Vector sBERT MPNET correspondiente al título.
DIARIO	Numérico	Editorial codificada en la que se publicó la noticia. Relación foránea al campo ID de la tabla COD_DIARIO.
RANK	Numérico	Page Authority de la página.
NUM_LINEAS	Numérico	Número de líneas con las que cuenta el artículo.
NUM_CAR_TITULO	Numérico	Número de caracteres que conforman el título de la noticia.
NUM_CAR_MAX_LINEA	Numérico	Número de caracteres de la línea de mayor longitud del contenido del titular.
INDEXADO	Booleano	Indica si la noticia ha sido indexada o no en el servicio Elasticsearch.
GRUPO	Numérico	Clúster al que pertenece la noticia.

Tabla 2.6. Campos de la tabla LINEAS_NOTICIAS_CODIFICADO (elaborado por el autor)

Campo	Tipo	Descripción/Función
ID	Numérico	Clave primaria, identificador único de línea de contenido. El valor corresponde al de la tabla original LÍNEAS_NOTICIAS.
ID_NOTICIA	Numérico	Identificador único de noticia. Relación foránea al campo ID de la tabla NOTICIAS_CODIFICADO.
LINEA	Texto	Línea de contenido de noticia.
LINEA_NSW	Texto	Línea de noticia removidas las palabras vacías (stop words).
LINEA_VECTOR	Texto	Línea de noticia convertida a vector sBERT MPNET.
INDEXADO	Booleano	Indica si la línea de noticia ya ha sido indexada en el servicio Elasticsearch.
PUNTAJE_LINEA_GRUPO	Numérico	Puntaje de la línea dentro del clúster.
PUNTAJE_FINAL_GRUPO	Numérico	Puntaje de la línea teniendo en cuenta el campo PUNTAJE_LINEA_GRUPO y su Page Authority.

Tabla 2.7. Campos de la tabla LINEAS_NOTICIAS_CALIFICACION (elaborado por el autor)

Campo	Tipo	Descripción/Función
ID	Numérico autogenerado	Clave primaria, identificador único de calificación realizada.
ID_LINEA_NOTICIA	Numérico	Identificador único de línea de noticia que ha sido calificada. Relación foránea al campo ID de la tabla LÍNEAS_NOTICIAS_CODIFICADO.
ES_LIKE	Booleano	Indica si la calificación es like, en caso de ser falso es dislike.
RAZONES	Texto	Indica las razones o motivos de la calificación que ha expresado el usuario final.

2.2.3.6 Tablas Catálogo

Las tablas catálogo son un componente fundamental para la codificación eficiente de las características del dataset. Su función principal radica en la codificación numérica de las características de los datos, permitiendo un análisis más adecuado y un manejo más preciso de la información.

Las tablas catálogo creadas son las siguientes:

- **Tabla COD_AUTOR:** Catálogo con la lista completa de autores.
- **Tabla COD_CATEGORIA:** Catálogo con la lista de categorías, por ejemplo: tecnología, política, etc.
- **Tabla COD_DIARIO:** Catálogo de las 24 editoriales contempladas para el análisis.
- **Tabla COD_MES:** Catálogo con los meses del año.

La Tabla 2.8 muestra los campos de los que se encuentran estructuradas cada una de las tablas catálogo.

Tabla 2.8. Campos de las tablas catálogo (elaborado por el autor)

Campo	Tipo	Descripción/Función
ID	Numérico autogenerado	Clave primaria, identificador único.
VALOR	Texto	Corresponde al valor textual del ítem del catálogo que lo describe en sí mismo.

2.2.3.7 Vista VW_PUNTAJES

La vista VW_PUNTAJES facilita el acceso a los resultados del puntaje basado en una métrica compuesta. Esta vista, utilizada por la interfaz gráfica de usuario, presenta la información de forma clara y organizada al usuario final. La Tabla 2.9 describe los campos que conforman la vista.

2.2.4 Procedimientos Almacenados y Funciones

Para optimizar la eficiencia de ejecución ante la gran cantidad de datos del dataset, se implementaron procesos almacenados y una función en la base de datos.

Tabla 2.9. Campos de la vista VW_PUNTAJES (elaborado por el autor)

Campo	Tipo	Tabla Original	Descripción/Función
ID	Numérico	LINEAS_NOTICIAS_CODIFICADO (ID)	Identificador de línea de contenido del artículo.
ID_NOTICIA	Numérico	NOTICIAS_CODIFICADO (ID)	Identificador de noticia a la que pertenece la línea de contenido.
TITULO	Texto	NOTICIAS_CODIFICADO (TITULO)	Título de la noticia a la que pertenece la línea de contenido.
LINEA	Texto	LINEAS_NOTICIAS_CODIFICADO (LINEA)	Línea de contenido.
PUNTAJE	Numérico	Métrica compuesta calculada en base a: - LINEAS_NOTICIAS_CODIFICADO (PUNTAJE_FINAL_GRUPO) - LINEAS_NOTICIAS_CALIFICACION (ES_LIKE)	Métrica compuesta con el puntaje final de la línea de contenido.
GRUPO	Numérico	NOTICIAS_CODIFICADO (GRUPO)	Clúster al que pertenece la línea de contenido.
DIARIO	Texto	COD_DIARIO (VALOR)	Editorial que publicó la noticia a la que pertenece la línea de contenido.

Los procesos almacenados y funciones creados son:

- **LIMPIAR_LINEAS:** Proceso almacenado que limpia las líneas “basura” de la tabla LINEAS_NOTICIAS_CODIFICADO buscando patrones de texto.
- **REMOVER_SW_ENTRADA:** Función que acepta como variable de entrada una cadena de texto y devuelve la cadena de texto con las palabras vacías (stop words) removidas.
- **REMOVER_SW_NOTICIAS_CODIFICADO:** Proceso almacenado que remueve las palabras vacías (stop words) del campo TITULO de la tabla NOTICIAS_CODIFICADO y lo almacena en el campo TITULO_NSW.

- **REMOVER_SW_LINEAS_NOTICIAS_CODIFICADO:** Proceso almacenado que remueve las palabras vacías (stop words) del campo LINEA de la tabla LINEAS_NOTICIAS_CODIFICADO y lo almacena en el campo LINEA_NSW.
- **CALCULAR_PARAMETROS_NUMERICOS:** Proceso almacenado que calcula los metadatos de las noticias. Llena los campos NUM_LINEAS, NUM_CAR_TITULO y NUM_CAR_MAX_LINEA de la tabla NOTICIAS_CODIFICADO.
- **CALCULAR_PUNTAJE_FINAL_GRUPO:** Calcula el puntaje final de la línea dentro del grupo. Llena el campo PUNTAJE_FINAL_GRUPO de la tabla LINEAS_NOTICIAS_CODIFICADO.

2.3 Preparación de los Datos

En esta etapa, en base a los resultados obtenidos en la etapa anterior, se analizaron los datos para definir una estrategia de limpieza y posteriormente indexarlos en la instancia Elasticsearch.

2.3.1 Limpieza de Datos

La limpieza de datos se abordó en dos subfases. La primera se enfocó en la tabla NOTICIAS, limpiando la información general de los titulares. En la segunda fase, se limpió el contenido textual de los artículos en la tabla LINEAS_NOTICIAS.

2.3.1.1 Limpieza de la Tabla NOTICIAS

La limpieza inicia con la carga de la tabla NOTICIAS a un dataframe de la librería Pandas. Pandas es una librería open source de Python para análisis de datos, que proporciona a Python la capacidad de trabajar con tablas realizando operaciones con un rendimiento óptimo [67].

Se realizó una exploración de los datos y se identificó las columnas que no son relevantes para el análisis. Estas columnas son:

- **Año:** El año del titular resulta irrelevante debido a que todas las noticias son del año actual, no provee más información.
- **Url:** La URL fue de utilidad durante la fase de web scraping; sin embargo, al ser definida por cuestiones técnicas no se considera una variable relevante.
- **Día:** El día es bastante variable y muchas veces obedece a la casualidad. Por lo que no se considera una variable objetiva para el análisis.
- **Hora y minuto:** Similar a la variable día. No son variables objetivas para el análisis.

Se ha considerado mantener la variable mes, esto debido a que existen fechas de días festivos en los que se publican determinados tipos de titulares.

En un paso posterior, se realizó la detección de variables con valores nulos. La Figura 2.6 ilustra las características del conjunto de datos, incluyendo el conteo de valores obtenidos a partir del proceso de extracción. Esta información permite identificar las variables con mayor cantidad de datos faltantes, lo que resulta importante para la etapa posterior de imputación y análisis.

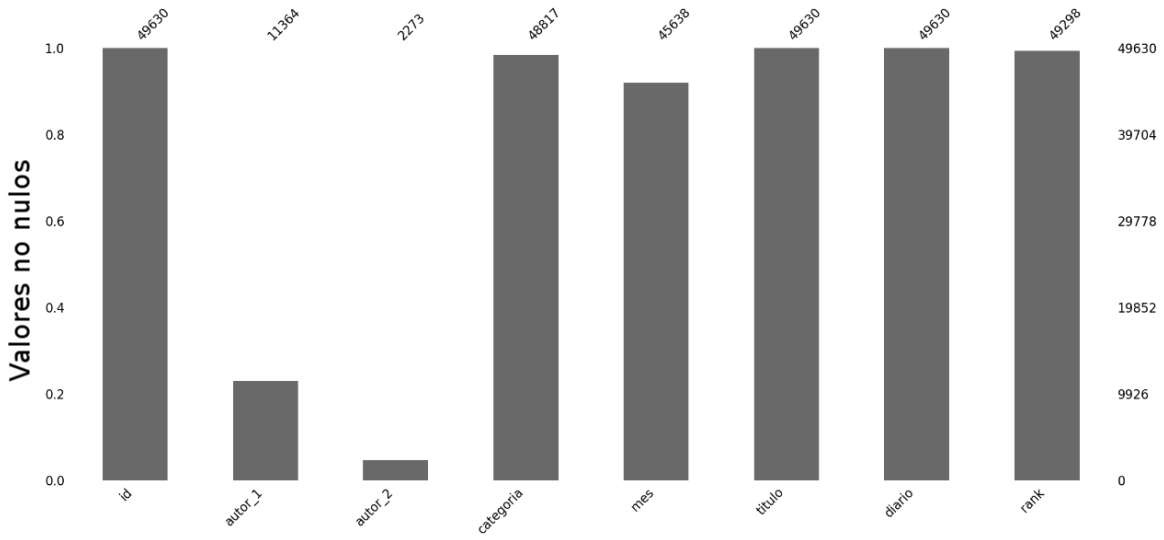

Figura 2.6. Valores faltantes en el dataset (elaborado por el autor)

En el presente análisis, se identificaron las variables con valores nulos: autor_1, autor_2, categoría, mes y rank.

- **Autoría:** La omisión de autores en los titulares se debe principalmente a la costumbre generalizada de algunas editoriales de publicar de manera anónima ciertas noticias. En este caso la editorial opta por atribuir la autoría a la propia editorial, por lo que se imputan estos valores nulos con el nombre de la editorial.
- **Categoría:** Las noticias sin una categoría específica, generalmente son de temática general, se clasificaron en la categoría "Mundo", que aborda temas de acontecer global.
- **Mes:** La mayoría de los titulares pertenecían al mes de enero, por lo que se imputó la variable "mes" por moda.
- **Rank:** Existen titulares que no tienen un ranking de Page Authority definido a la fecha, es por este motivo que se ha optado por tomar el promedio por editorial para llenar estos valores faltantes.

Estas estrategias de imputación permiten abordar la ausencia de datos de manera coherente, fortaleciendo el conjunto de datos para análisis posteriores.

Tras la imputación de datos, se procedió a la creación de tablas de catálogo para las variables autores, categoría, diario y mes. Estas tablas de catálogo contienen los valores únicos de cada variable, codificados numéricamente para facilitar el análisis y la eficiencia en el procesamiento.

La codificación se realizó utilizando el campo ID de la base de datos, que asigna un valor único a cada registro durante la inserción. Al mismo tiempo que se creaban las tablas de catálogos, las columnas del dataframe de Pandas se codificaron con el valor numérico (ID) asignado.

Este proceso de codificación permite una mejor comprensión de los datos, al convertir valores textuales en valores numéricos fácilmente interpretables por los modelos de análisis. Además, facilita el manejo y análisis de grandes conjuntos de datos, optimizando el almacenamiento y rendimiento [68].

El dataframe codificado se almacenó en la tabla LINEAS_NOTICIAS_CODIFICADO. Luego, el proceso almacenado REMOVER_SW_NOTICIAS_CODIFICADO eliminó las palabras vacías (stop words) del campo TITULO, llenando el campo TITULO_NSW. Finalmente, el modelo paraphrase-multilingual-mpnet-base-v2 transformó los títulos, sin palabras vacías, en vectores de 768 dimensiones, y fueron almacenados en el campo TITULO_VECTOR. Este proceso facilita el manejo, análisis y procesamiento de estos vectores de gran dimensión.

2.3.1.2 Limpieza de la Tabla LINEAS_NOTICIAS

Con el objetivo de eliminar contenido irrelevante para el análisis, se realizó la limpieza de la tabla LINEAS_NOTICIAS. Para ello, se pobló la tabla LINEAS_NOTICIAS_CODIFICADO y se ejecutó el proceso almacenado LIMPIAR_LINEAS.

Este proceso identifica y elimina líneas "basura" mediante la búsqueda de patrones específicos. Las líneas "basura" son aquellas que se encuentran entre el contenido del texto, que incluyen palabras como "Suscríbete al diario", "Anuncio", entre otras, y que no aportan valor al análisis que se requiere realizar. La limpieza de estas líneas "basura" permite obtener un conjunto de datos más preciso y relevante para el análisis posterior, mejorando la calidad y confiabilidad de los resultados.

Posteriormente, se ejecutó el proceso almacenado CALCULAR_PARAMETROS_NUMERICOS para completar los campos de metadatos de las noticias: NUM_LINEAS, NUM_CAR_TITULO y NUM_CAR_MAX_LINEA.

Para optimizar el análisis de las líneas de noticias, se ejecutó el proceso almacenado REMOVER_SW_LINEAS_NOTICIAS_CODIFICADO. Este proceso elimina las palabras vacías (stop words) del campo LINEA, llenando el campo LINEA_NSW en la tabla LINEAS_NOTICIAS_CODIFICADO.

Para concluir el procesamiento de las líneas de noticias, se realizó la conversión de estas a vectores. Este proceso se llevó a cabo utilizando el modelo paraphrase-multilingual-mpnet-base-v2, mediante el cual se transformó las líneas de texto, sin palabras vacías, en vectores de 768 dimensiones. Estos vectores, que representan la información semántica de las líneas de noticias, se almacenaron en la columna LINEA_VECTOR de la tabla LINEAS_NOTICIAS_CODIFICADO.

2.3.2 Ingesta de Titulares en Elasticsearch

Una vez finalizada la limpieza de datos y obtenidas las representaciones vectoriales de los encabezados de noticias, se procede a su ingesta desde la base de datos hacia la instancia de Elasticsearch.

Para la ingesta de los titulares, se ha creado un índice en Elasticsearch denominado "encabezados". La Figura 2.7 muestra la estructura de los documentos a ser guardados en este índice. Este índice alberga los 49.630 encabezados de noticias que serán utilizados durante la fase de clusterización.

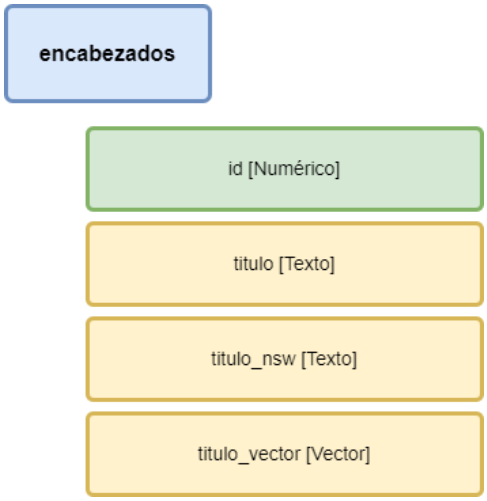

Figura 2.7. Campos del documento del índice de encabezados (elaborado por el autor)

El índice "encabezados" posee una estructura que permite almacenar las distintas versiones del campo título de la noticia. Esto facilita la ejecución de consultas utilizando diversas técnicas, ya sea basadas en la representación textual sin palabras vacías (campo "titulo_nsw") o en la representación vectorial (campo "titulo_vector"). Permitiendo retornar la respuesta completa del título (campo “titulo”) y el identificador único (campo “id”).

El campo “id”, identifica inequívocamente al documento dentro del índice. Es un campo integrado a la estructura de documento de Elasticsearch [42]. El campo “id” de cada documento en Elasticsearch se ha llenado con el campo ID en la tabla NOTICIAS_CODIFICADO. Esto permite recuperar información adicional de la base de datos si es necesario.

2.3.3 Ingesta de Contenido de Noticas en Elasticsearch

La ingesta de contenido tiene como objetivo almacenar e indexar las líneas de información de forma organizada en la instancia Elasticsearch. Esta organización se basa en índices que han sido creados uno para cada clúster, estos permitirán una mayor eficiencia durante el análisis posterior.

Tras la definición de los clústeres, en las fases anteriores, las noticias fueron agrupadas en base a su campo GRUPO dentro de la tabla NOTICIAS_CODIFICADO. De esta manera, se creó una estructura que facilita el acceso a los grupos. A la vez, las líneas fueron removidas de palabras vacías (*stop words*) y fueron transformadas mediante el modelo paraphrase-multilingual-mpnet-base-v2 en vectores de 768 dimensiones que fueron almacenados en la base de datos. Esta información debe ser ingestada en índices dentro del motor de búsquedas para que se pueda obtener rendimientos óptimos durante el procesamiento.

La Figura 2.8 muestra la estructura del índice creado. Se crea un índice para cada clúster que lleva por nombre el prefijo “cl_” seguido del número de clúster. En este índice se almacenarán las líneas de contenido de cada noticia que conforma el clúster.

El índice abarca las distintas transformaciones de la línea de contenido, su forma original, su transformación con palabras vacías removidas y su transformación vectorial. En base a esta información almacenada se puede ejecutar el proceso de puntuación de contenido.

El campo ID juega un papel fundamental en la interconexión entre el documento del motor de búsquedas y el registro de la base de datos relacional. Al almacenar el valor del campo ID de la tabla LINEAS_NOTICIAS_CODIFICADO dentro del campo de identificación del

documento Elasticsearch, se crea una referencia directa e inequívoca entre ambos elementos.

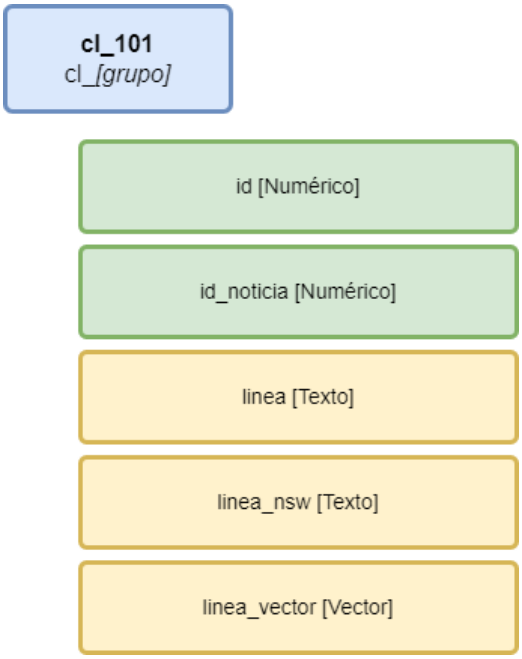

Figura 2.8. Estructura de índice para un determinado clúster (elaborado por el autor)

Para la estructura de los índices también se ha considerado el almacenamiento del identificador de noticia. Esto permite obtener de manera inmediata, tras la búsqueda, el código único de noticia que corresponde al campo ID de la tabla NOTICIAS_CODIFICADO.

2.4 Construcción de Modelos

En la construcción de modelos se aborda la construcción de los tres componentes que integran el sistema:

- **Agrupador de titulares:** Este componente permite agrupar en base a la búsqueda textual y semántica los titulares por similitud haciendo uso de un algoritmo personalizado para el análisis de los datos.
- **Ranking de contenido:** Permite hacer un ranking de cada línea de contenido basándose en una métrica compuesta que considera: la comparativa de noticias en un determinado clúster y la autoridad de la página (Page Authority) publicada del titular en Internet.

Adicional, se ha entrenado un modelo para predecir la autoridad de la página en Internet ya que este valor puede no estar disponible, especialmente si el titular es reciente.

- **Predictor de autoridad en Internet:** Un modelo supervisado de machine learning entrenado en base a los parámetros de la editorial y metadatos del titular que permite predecir el grado de autoridad de la noticia en Internet y que sirve como parte de la métrica compuesta de ranking.

2.4.1 Agrupador de Titulares

El titular periodístico en sí mismo sitúa mucho más la noticia: la valora, le confiere relieve, destaca especialmente un aspecto del texto, entre otros aspectos relevantes. El encabezamiento en sí mismo resume la noticia y la clarifica [69]. La agrupación de noticias según su titular, tomando como base su significado, es una tarea necesaria para el análisis de datos. La identificación semántica se hace importante por cuanto la información es recolectada desde diversas fuentes y cada una posee estilos de redacción distintos.

Para el proceso de clusterización de titulares se ha implementado un algoritmo de agrupamiento que, en base a la identificación de la similitud entre los títulos de las noticias permite, asignar los grupos. El algoritmo, a través de un sistema de búsqueda cuidadosamente diseñado, extrae resultados objetivos para un análisis preciso.

2.4.1.1 Estructura del Buscador de Texto

El buscador de texto para clusterización ha sido diseñado para identificar titulares con una significativa relación entre sí. Su estructura permite no solo detectar similitudes, sino también medir su grado de similitud (score), posibilitando una comparación precisa de los resultados.

Para la estructura del buscador de textos para clusterización se hace uso del motor de búsquedas que computa los resultados en base a varios algoritmos. El cálculo del score engloba varias fases y algoritmos. En una fase inicial, la frase a buscar es identificada dentro del índice haciendo uso del algoritmo BM25.

BM25 (Okapi BM25) es un algoritmo que permite puntuar las búsquedas en base a las palabras ingresadas [70]. Este algoritmo tiene su fundamento en el modelo probabilístico de recuperación de información BIR (Binary Independent Retrieval) [71]. La función de ranking BM25 se implementa mediante bolsas de palabras. Una bolsa de palabras es un modelo que representa un documento como un conjunto de palabras sin tomar en cuenta su orden ni gramática. La idea detrás de dicho modelo es que los documentos de texto pueden representarse mediante un histograma de distribución estadística, una función de densidad de probabilidad de sus palabras más significativas, enfocándose en los vocablos de gran importancia para reconocer el tema del texto [72].

El algoritmo BM25 permite realizar un análisis literal enfocado en el modo en el que se encuentra escrita la oración de búsqueda. No es capaz en sí mismo en identificar el contexto ni la semántica, pero puede identificar patrones de texto de similitud con bastantes resultados satisfactorios. El algoritmo es ejecutado haciendo uso de la cadena de texto del titular con las palabras vacías removidas, el cual busca entre los campos correspondientes a los titulares con palabras vacías removidas que fueron almacenados en el índice durante el proceso de ingesta. De esta manera logra identificar sus similares.

Los resultados del algoritmo BM25 deben superar un score mínimo del 65% para ser considerados para la agrupación. Este umbral funciona como un filtro que garantiza la precisión de los clústers.

El valor de umbral seleccionado debe tomar en cuenta las siguientes definiciones: Un valor mayor de umbral genera clústers con mayor similitud textual entre sus titulares y permite un análisis más preciso y específico de las relaciones entre los textos. Mientras que un menor umbral permite agrupar noticias de manera más general y puede ser útil para obtener una visión global de los temas presentes en un conjunto de datos.

El valor de 65% ha sido seleccionado de manera práctica en base a la evaluación de resultados. Se ha demostrado que este umbral produce clústers con un buen balance entre precisión y generalidad.

El score obtenido tras haber ejecutado el algoritmo BM25 representa el 70% de la puntuación global.

El buscador de textos para clusterización no solo se limita a la coincidencia de palabras clave. Para una comprensión más profunda de los textos, también toma en cuenta la semántica y el contexto de las palabras dentro de la oración. Para ello, se ejecuta una búsqueda utilizando el algoritmo KNN. Este algoritmo identifica los vecinos vectoriales más cercanos dentro del índice. Este algoritmo utiliza un vector para buscar sus relacionados. Este vector tiene 768 dimensiones y corresponde a la transformación ejecutada en base al modelo sBERT MPNET. El algoritmo busca similitudes semánticas entre los textos comparando los vectores que fueron almacenados en el índice durante el proceso de ingesta. Cuanto más similares sean los vectores, más probable es que los textos tengan un significado similar.

De manera análoga a la búsqueda anterior el resultado es un score que permite medir el grado de similitud semántica del titular con los demás. Para ser considerados para la agrupación, los resultados deben superar un score mínimo del 67%.

Este umbral está definido de manera similar al anterior, pero para el análisis semántico tiene como características: Un mayor umbral genera clústers con mayor precisión semántica y permite un análisis más profundo y específico de las relaciones semánticas. Mientras que un menor umbral permite agrupar semánticamente noticias de manera más general y puede ser útil para obtener una visión global de los temas presentes en un conjunto de datos.

El valor de 67% ha sido seleccionado de manera práctica en base a la evaluación de resultados. Se ha demostrado que este umbral produce clústers con un buen balance entre precisión semántica y generalidad.

El proceso de clusterización culmina con la ponderación de los dos scores obtenidos: 30% para la similitud semántica (KNN) y 70% para la frecuencia de palabras (BM25). Solo los resultados que superen ambos umbrales (65% en BM25 y 67% en KNN) son considerados para la agrupación.

Adicionalmente, se exige que los resultados se encuentren dentro los primeros mejores. Se seleccionan los 17 primeros resultados para cada algoritmo, que representan el 70% de las 24 editoriales en total. Esta medida pretende asegurar que la información esté relacionada, ya que se espera que un 70% de las editoriales como promedio hayan abordado un titular en común.

La implementación de estos umbrales conlleva una serie de verificaciones que buscan garantizar que solo se tome en cuenta para la clusterización titulares que estén relacionados entre sí.

La Figura 2.9 muestra el proceso de puntuación de manera simplificada. Este proceso es ejecutado para cada uno de los titulares que no tengan un grupo definido, permitiendo obtener los titulares relacionados.

El proceso de búsqueda identifica los titulares que son candidatos a ser agrupados en un mismo clúster. Sin embargo, no todos los candidatos son idóneos. Para asegurar la precisión y la calidad de los clústers, se utiliza un algoritmo de agrupamiento como filtro final.

2.4.1.2 Algoritmo de Agrupamiento

Para tomar la decisión de integrar una noticia a un determinado clúster, se hace uso de un algoritmo de agrupamiento. Este algoritmo se fundamenta en el uso del buscador de texto estructurado anteriormente.

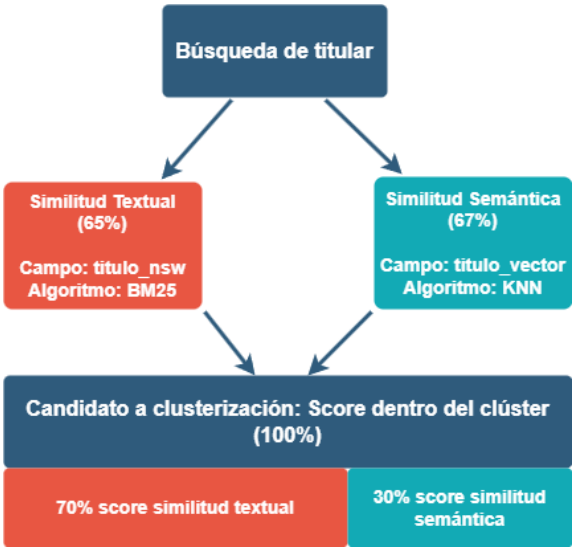

Figura 2.9. Puntuación de titulares relacionados para clusterización (elaborado por el autor)

El algoritmo de clusterización funciona de forma iterativa, partiendo de un titular raíz que no tiene un grupo definido. El objetivo es encontrar un grupo adecuado para este titular.

El proceso seguido es el siguiente:

1. Se busca el titular raíz utilizando el algoritmo de búsqueda.
2. Se analizan los resultados de la búsqueda (titulares candidatos).
3. Para cada titular candidato, se realiza una nueva búsqueda.
4. Si el titular raíz aparece entre los candidatos de la nueva búsqueda, se confirma una relación inequívoca entre ambos titulares.
5. En este caso, se forma un clúster con el titular raíz y los titulares candidatos que cumplan la relación inequívoca.

Este proceso iterativo se repite hasta que no se encuentren más relaciones inequívocas. La Figura 2.10 ilustra este proceso.

En algunos casos, los titulares candidatos pueden pertenecer a clústers ya formados. Para determinar a qué clúster se debe integrar la noticia raíz, se realiza un análisis inteligente que toma en cuenta la cohesión interna de los candidatos dentro de cada clúster.

El proceso seguido es el siguiente:

1. Se calcula la suma de los scores de todos los miembros candidatos que pertenecen al mismo clúster.

2. Se selecciona el clúster con el mayor score acumulado.
3. La noticia raíz y todas las demás noticias que cumplen la relación inequívoca se integran al clúster seleccionado.

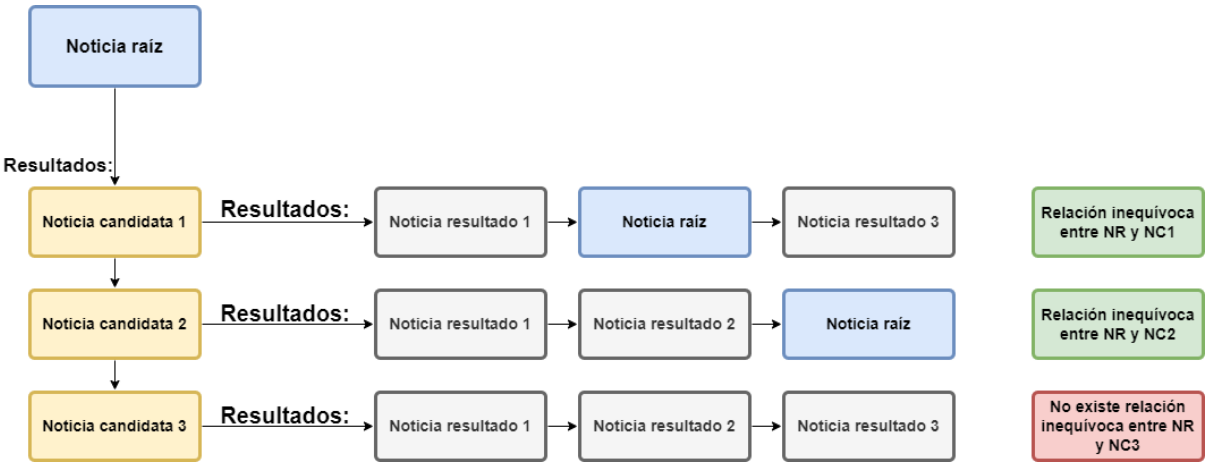

Figura 2.10. Proceso de conformación de un clúster (elaborado por el autor)

La Figura 2.11 ilustra el proceso de asignación de grupo cuando existen candidatos pertenecientes a grupos previos.

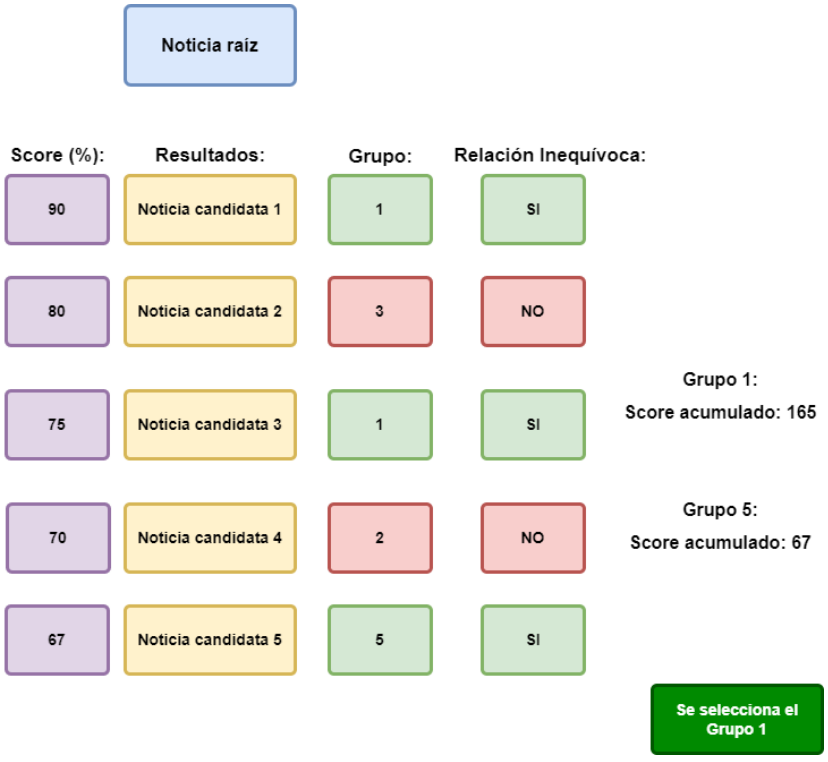

Figura 2.11. Asignación de grupo cuando existen candidatos pertenecientes a un clúster (elaborado por el autor)

La Figura 2.12 muestra el diagrama de flujo general del algoritmo de clusterización. Las transacciones, sobre todo las que involucran vectores de 768 dimensiones, demandan un alto rendimiento computacional. Para optimizar el algoritmo de clusterización, se ha implementado una rutina específica que busca mejorar la eficiencia del proceso. Esta rutina tiene como objetivo definir la agrupación del titular de manera directa sin realizar la identificación inequívoca de cada uno de sus candidatos, proceso último que puede resultar en costo computacional.

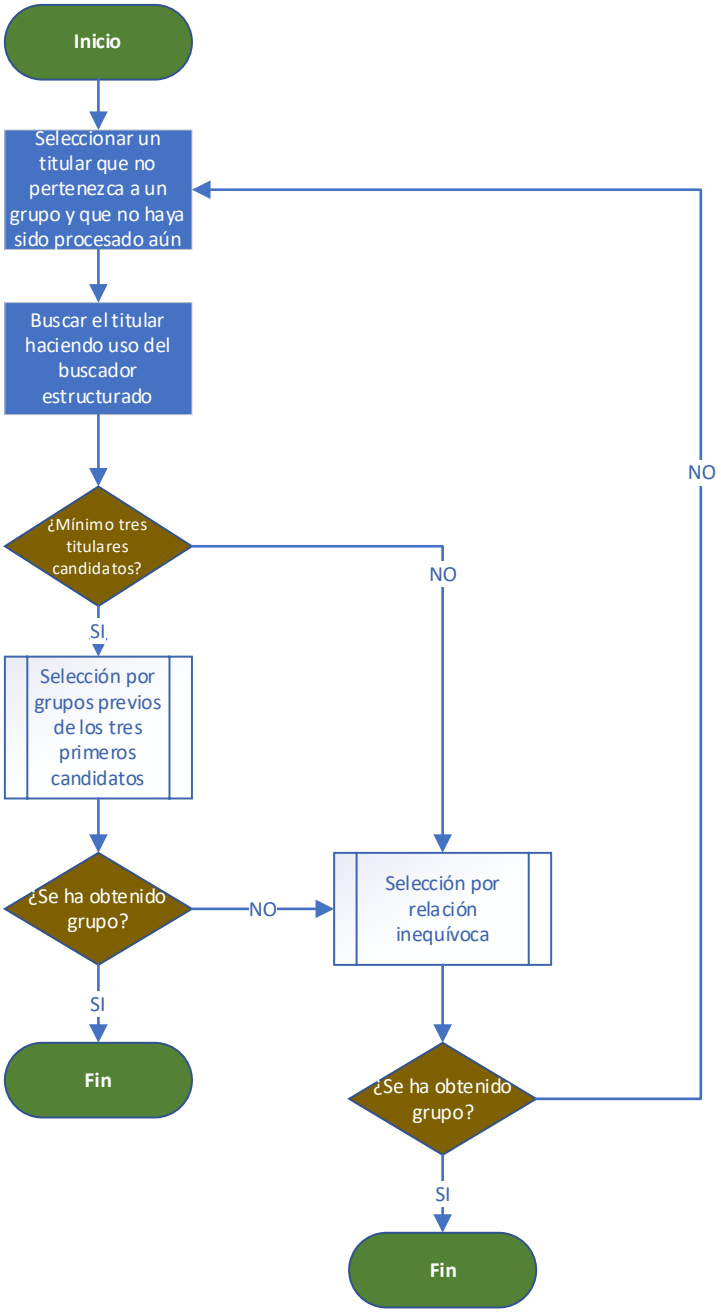

Figura 2.12. Diagrama de flujo del algoritmo de agrupamiento (elaborado por el autor)

2.4.1.3 Algoritmo para Mejorar la Eficiencia

En aras de proveer de un mayor rendimiento al algoritmo, se ha desarrollado una rutina. Esta, tomando como base los clústers ya formados y la puntuación obtenida en la búsqueda, se encarga de identificar con mayor rapidez el clúster al que debería pertenecer la noticia raíz. De esta manera, se logra una agrupación más veloz y eficiente de las noticias, permitiendo una mejor explotación del hardware.

La presente rutina se basa en los tres primeros resultados de la búsqueda para determinar el clúster al que debería pertenecer el titular. Para ello, se requiere que el titular tenga al menos tres resultados. La Figura 2.13 ilustra este proceso, donde se analiza el score de los tres primeros resultados para identificar el grupo más adecuado para el titular. Esta estrategia se fundamenta en la influencia que tiene el pertenecer a un grupo preexistente y obtener una alta puntuación como candidato para un titular.

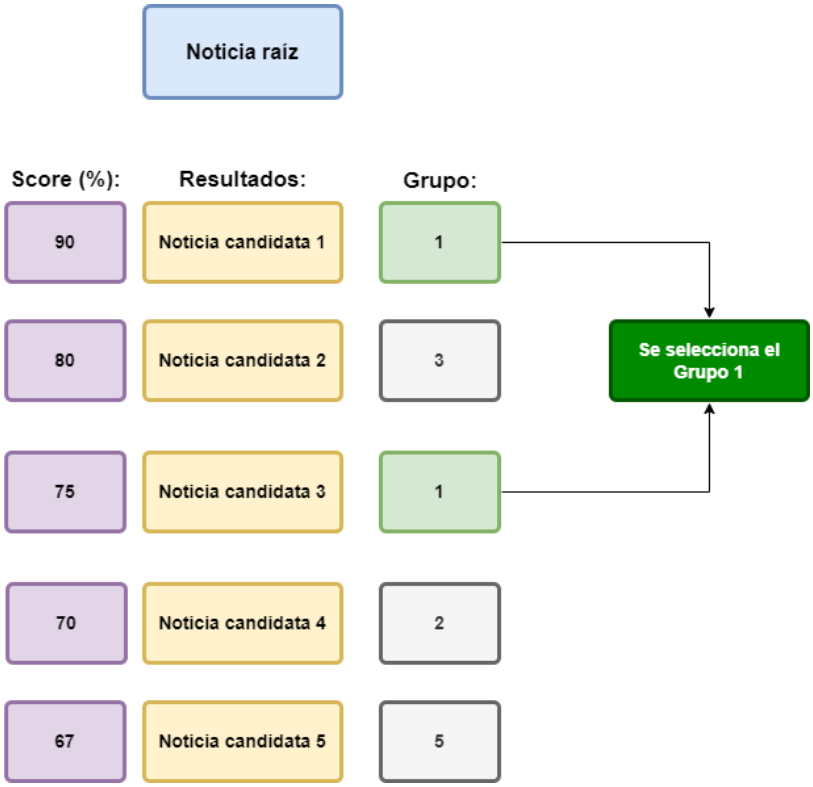

Figura 2.13. Proceso de clusterización basado en mejores candidatos con grupos previos (elaborado por el autor)

Para optimizar la asignación de clústers, la rutina implementa un proceso secuencial de análisis. En primer lugar, se verifica si la primera y segunda noticias candidatas comparten el mismo clúster. Si se cumple esta condición, dicho clúster se asigna directamente al titular buscado (noticia raíz). En caso contrario, se repite el análisis con la primera y tercera

noticias candidatas. Si estas dos noticias también pertenecen al mismo clúster, este se asigna a la noticia raíz. Si ninguna de las dos condiciones anteriores se cumple, se procede a realizar el proceso de clusterización habitual.

2.4.2 Ranking de Contenido

2.4.2.1 Estructura General de Sistema de Ranking

En Internet no solo basta con generar tráfico a la web, se debe tener en cuenta que también el contenido debe ser de calidad [73]. Actualmente, en la nueva era digital, donde el contenido abunda y la competencia es cada vez más feroz, ya no basta con una sola métrica para garantizar la calidad. Se requiere un enfoque multifacético que evalúe diversos aspectos para determinar el verdadero valor de la información.

La calidad, que puede ser definida como el grado de satisfacción o insatisfacción del usuario final [74], requiere de una evaluación integral de las características de un servicio o producto que permita identificar su utilidad, importancia, reputación y otras más que puedan determinar la percepción por parte del usuario.

Esta sección presenta un sistema de puntuación para identificar, dentro de cada clúster de noticias, el contenido de mayor calidad para el usuario final. Se ha realizado un análisis riguroso de los datos siguiendo un proceso que identifica las características clave que diferencian la calidad del contenido dentro del mismo clúster.

Esta fase del proyecto tiene como punto de partida los clústers formados tras haber seguido el proceso de fases anteriores y en base a estos resultados ejecuta nuevos procesos que permiten computar las puntuaciones de contenido.

Una vez formados los grupos o clústers a los que pertenecen cada noticia, es necesario que el contenido sea organizado de tal manera que se presente el de mayor calidad primero. Para ello se requiere identificar este contenido y priorizarlo. El objetivo es el de discernir la relevancia de cada línea dentro del contexto de un clúster en base a un sistema de puntajes. Esto permite evaluar integralmente el contenido de las noticias dentro del clúster.

El sistema de puntajes implementado va más allá de una simple medición, proporcionando una métrica compuesta del contenido en base a varios factores. Este sistema se basa en tres pilares fundamentales:

- **Relevancia del contenido:** Se analiza la relevancia de las palabras dentro del clúster.

- **Reputación en Internet:** Para comprender mejor la recepción de una noticia por parte del público, se emplea un indicador clave: la autoridad que la noticia posee en Internet (Page Authority). Esta métrica refleja el nivel de aceptación y confianza que los usuarios depositan en la información. En el caso de noticias nuevas que aún no han acumulado esta información, se utiliza un método predictivo basado en técnicas de aprendizaje automático. Este método permite estimar la autoridad de la noticia con un alto grado de precisión.
- **Puntuaciones de usuario final:** Se consideran las valoraciones de los usuarios que han interactuado con el contenido.

La Figura 2.14 muestra la ponderación de cada uno de estos factores en la construcción del puntaje final.

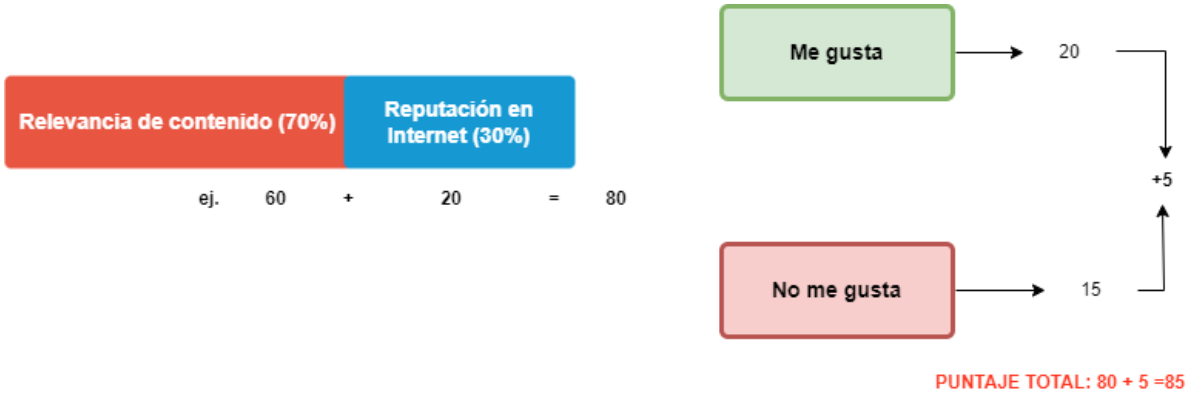

Figura 2.14. Distribución ponderada para métrica compuesta de puntuación de noticias (elaborado por el autor)

La relevancia del contenido es el factor principal en la puntuación inicial, con un peso del 70%. La reputación en Internet de la noticia en particular también juega un papel importante, con un peso del 30%. Estos dos factores forman una métrica que constituye la puntuación inicial de la línea de contenido dentro del clúster. La puntuación inicial no es definitiva. Los usuarios tienen la posibilidad de valorar las noticias, aumentando o disminuyendo su puntaje. Esta participación permite afinar las puntuaciones de las noticias y reflejar con mayor precisión la opinión del público.

Se ha realizado un análisis línea a línea del contenido para identificar los puntos de mayor enfoque dentro de las noticias, examinando cada una de ellas dentro de su clúster correspondiente.

2.4.2.2 Ranking por Relevancia del Contenido

El objetivo de la evaluación de la relevancia del contenido es identificar los patrones que determinan la importancia de una línea de contenido dentro de un clúster. Este proceso se basa en un análisis meticuloso que se centra en la similitud y el significado de cada palabra dentro del clúster. Este análisis no toma en cuenta factores de índole general como características de la edición, que serán tomados en cuenta en un análisis posterior.

Para determinar la puntuación de una noticia, es fundamental identificar previamente el clúster al que pertenece. Esto debido a que la relevancia se calcula en comparación con otras noticias dentro del mismo clúster. Posteriormente se realiza una ingesta del contenido en la instancia del motor de búsquedas y se procede a puntuar cada una de las líneas de contenido.

El conjunto de datos utilizado en este proyecto comprende 537.146 líneas de contenido, lo que implica un volumen considerable de información. Al trabajar con transformaciones vectoriales de 768 dimensiones sobre este conjunto, es importante estructurar e indexar la información de manera adecuada. Esto permite optimizar el procesamiento y garantizar una buena eficiencia durante la ejecución de las tareas. La estructuración e indexación correctas facilitan el acceso a la información, optimizan las operaciones matemáticas y minimizan el tiempo de procesamiento. En consecuencia, se logra un mejor aprovechamiento de los recursos computacionales y se agilizan las tareas de análisis.

Para el cálculo de la puntuación de relevancia de contenido se hace uso del buscador de textos que realiza la búsqueda en base al modo de escritura de las palabras y también basándose en su semántica. Este proceso se ejecuta en cada uno de los índices correspondientes a los grupos formados.

El proceso de puntuación de cada línea de contenido comienza con su preparación previa. La línea se libera de palabras vacías (*stop words*) que no aportan valor, para luego ser segmentada en palabras (tokenización). Cada palabra emprende un proceso a través del clúster, donde es comparada con el resto del contenido mediante el algoritmo BM25. De esta comparación, surgen los resultados más similares, teniendo un umbral máximo de 17 resultados. Estos resultados son acompañados de su puntuación de similitud tras la búsqueda.

La búsqueda se basa en la coincidencia entre la palabra y el campo de la línea de contenido sin las palabras vacías (“linea_nsw”). De esta manera, al no considerar en ningún momento

las palabras vacías, se asegura una evaluación basada en contenido que aporte valor dentro de la oración.

La Figura 2.15 ilustra este proceso. La salida de este proceso son los resultados con su correspondiente puntaje sobre 100 puntos. El proceso culmina con el cómputo de la puntuación promedio de los resultados que se puedan obtener. De esta manera para el buscador BM25 se ha obtenido una puntuación promedio para la palabra sobre 100 puntos. Esta búsqueda se enfoca en la coincidencia de las palabras escritas literalmente.

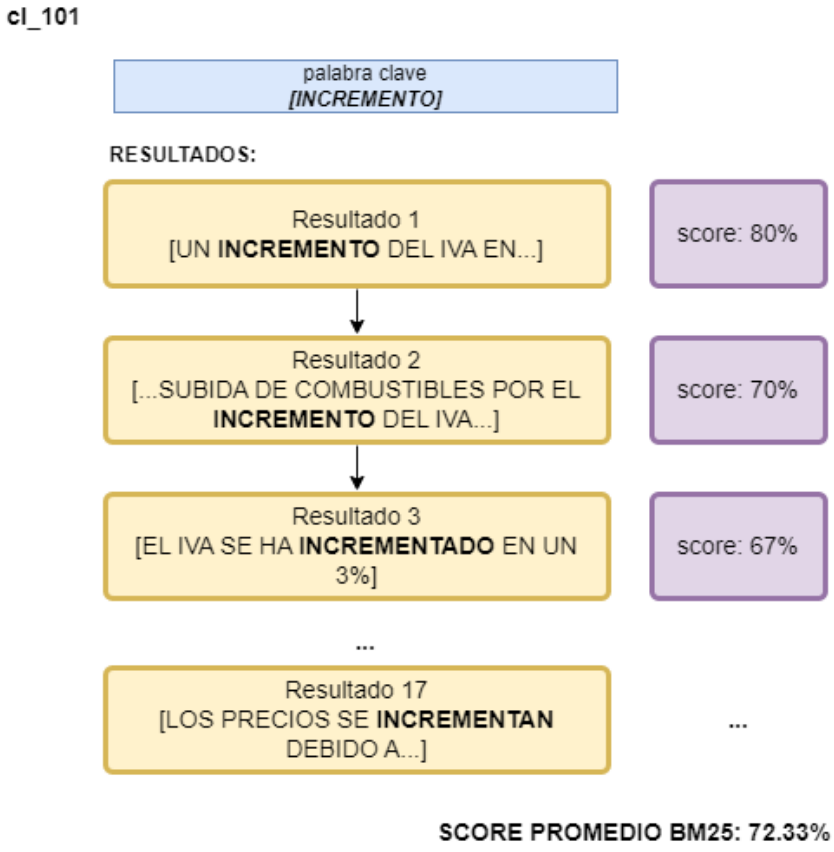

Figura 2.15. Búsqueda BM25 de palabra dentro del clúster para obtener puntaje de palabra (elaborado por el autor)

Posteriormente se realiza la búsqueda haciendo uso del algoritmo KNN. Para este proceso la palabra es transformada a un vector de 768 dimensiones haciendo uso del modelo paraphrase-multilingual-mpnet-base-v2. El vector es buscado para encontrar vectores similares. El proceso se ejecuta sobre el campo “linea_vector” almacenado en el índice. El proceso retorna los resultados con su puntaje de similitud vectorial sobre 100 puntos. El umbral máximo es de 17 resultados. Esta búsqueda se enfoca en la semántica.

La Figura 2.16 ilustra este proceso. Los resultados obtenidos son promediados para obtener el puntaje KNN sobre 100 puntos. Este resultado se promedia con el resultado anterior, basado en el algoritmo BM25. El resultado final corresponde a la valoración de la palabra en la línea de contenido.

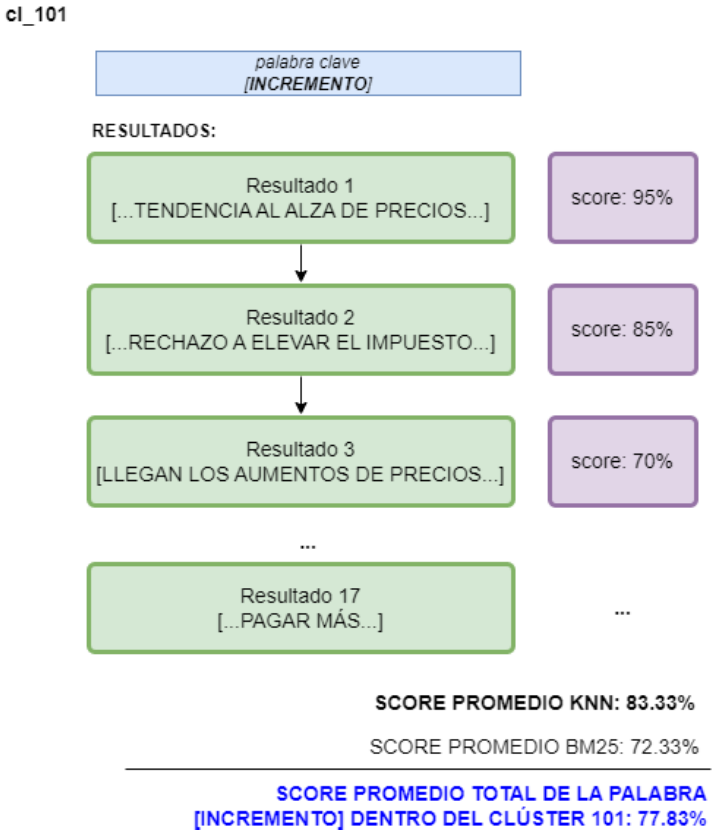

Figura 2.16. Búsqueda KNN de palabra dentro del clúster para obtener puntaje de palabra (elaborado por el autor)

Tras haber obtenido el puntaje de cada una de las palabras no vacías de la oración, los puntajes son promediados. Este promedio, que corresponde al total de las palabras, es el puntaje total de la línea de contenido.

El puntaje de línea permite identificar el grado de relevancia de las palabras dentro del clúster en base a su escritura literal y su significado. Una palabra que sea muy importante presentará tendencia a aparecer ya sea literalmente o en base a su semántica en la mayor parte del contenido y con una mayor puntuación.

La métrica final de la línea de contenido obtenida de este proceso será ponderada al 70% para conformar la métrica compuesta.

En la Figura 2.17 se sintetiza en un diagrama de flujo el proceso seguido para obtener la puntuación de las líneas de contenido. El proceso busca determinar la puntuación de cada palabra, la cual finalmente otorgará relevancia a la línea de contenido.

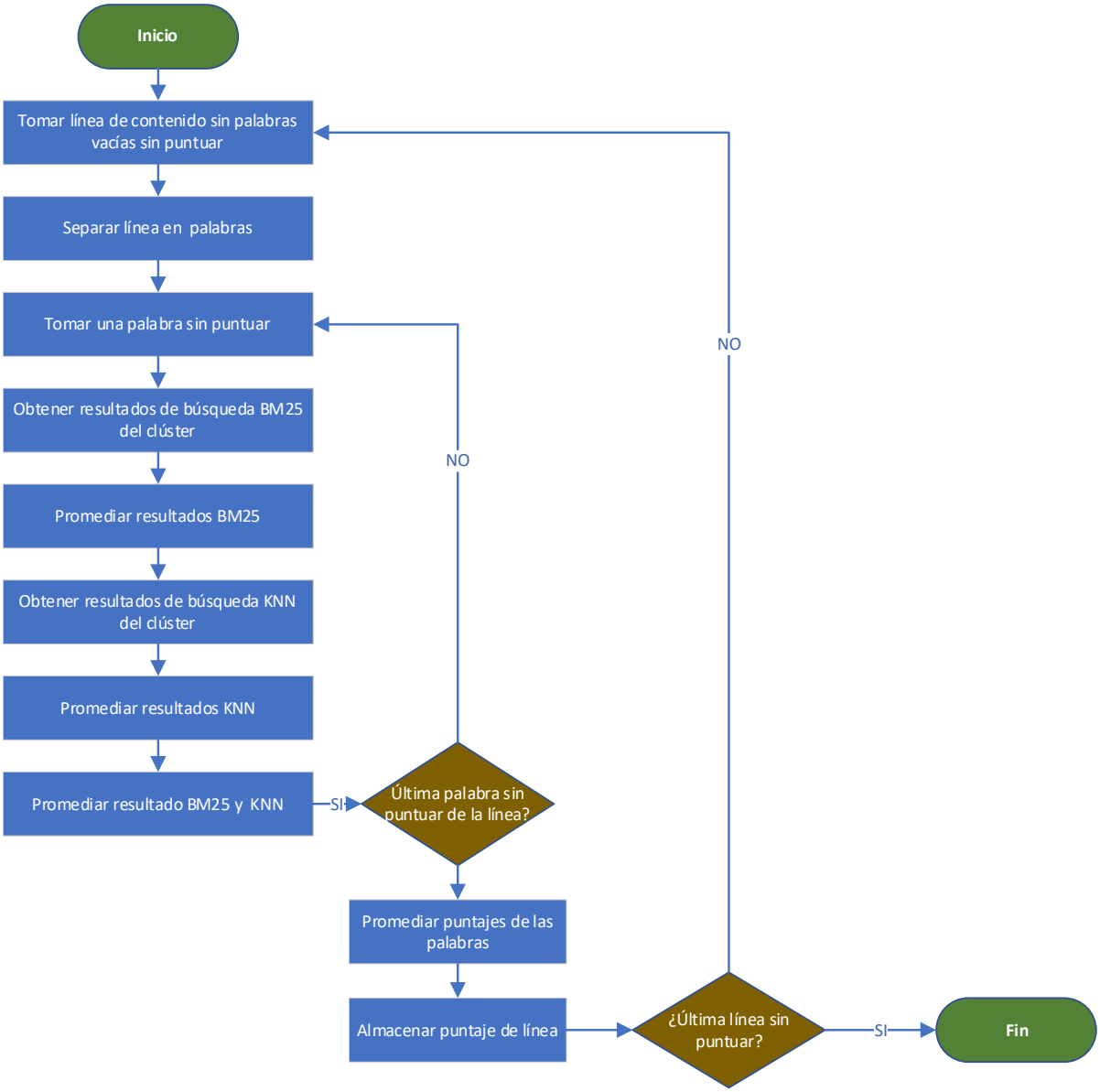

Figura 2.17. Diagrama de flujo de puntuación de contenido (elaborado por el autor)

2.4.2.3 Puntuación por Reputación en Internet

El grado de importancia y visualización de una página en Internet se puede resumir en una palabra: “reputación”. Internet consiste en billones páginas, cada una con una gran cantidad de información disponible [75]. Las páginas con mayor contenido de utilidad generan una mayor reputación progresivamente con el pasar del tiempo. Estas páginas

tras haber adquirido reputación aparecen primero en los buscadores y son los contenidos más consumidos por los usuarios.

Existen varias maneras de medir la reputación en Internet de una página web, de las cuales derivan una serie de algoritmos como: Page Authority, Page Rank, Semrush's Authority Score, entre otros. Estos algoritmos conllevan una serie de métricas que, basándose en varios factores, buscan medir con la mayor exactitud y precisión posibles el grado de reputación que una página web tiene en Internet.

La métrica de puntuación basada en la reputación de la noticia en Internet abarca el 30% de la puntuación global. Esta métrica se basa en la autoridad de la página (Page Authority) en que se publicó el titular. Page Authority es una métrica robusta y un buen indicador de cuán buena es una página web en particular [76]. Page Authority no debe confundirse con Page Domain. Page Authority mide el grado de reputación de una página en particular mientras que Page Domain de un sitio web o dominio específico.

Page Authority es una puntuación de clasificación basada en un número entero entre 0 y 100 y se calcula como una métrica compuesta basada en 40 parámetros que incluyen mozRank, mozTrust, dominios raíz de enlace, número total de enlaces, entre otros. La autoridad de página se calcula para una sola página utilizando una escala logarítmica para obtener una puntuación [77].

Page Authority es una métrica fundamental para evaluar la capacidad de una página web para posicionarse en los resultados de búsqueda (Search Engine Results Page SERP). Debido a su precisión y confiabilidad, Page Authority se ha convertido en una herramienta indispensable para los profesionales del SEO (Search Engine Optimization) y el marketing. Permite evaluar el potencial de una página web, identificar áreas de mejora y desarrollar estrategias para optimizar su posicionamiento en los buscadores.

El valor de Page Authority de cada noticia electrónica descargada nos ofrece una valiosa medida de su aceptación en Internet. Esta métrica, obtenida durante la fase de recopilación de datos, nos permite evaluar la relevancia y el impacto de cada noticia. Sin embargo, existen varios motivos por los cuales una noticia puede carecer de un valor de Page Authority, por ejemplo, si se ha publicado recientemente y aún no han tenido tiempo suficiente para definirse un valor. Para completar el vacío que se generaría en las nuevas noticias que se integren al sistema a futuro, se propone el entrenamiento de un modelo de aprendizaje automático, que en base a la información del dataset permita predecir el valor de Page Authority de la noticia.

Este modelo, entrenado en base a las características que definen la noticia, permite predecir el éxito de un titular en Internet. Se basa en las características específicas del titular y orienta la métrica al modelo de negocio de las editoriales. De esta manera, las editoriales pueden optimizar las decisiones de redacción de sus titulares, identificar las temáticas y formatos que mejor puntuación obtengan que generarán una mejor sintonía con su audiencia y generarán mayor impacto.

2.4.2.4 Puntuación de Usuario Final

Luego de calcular el puntaje de relevancia del contenido y la reputación de la noticia, se ponderan estos valores al 70% y 30% respectivamente, creando un puntaje inicial. Este puntaje se somete a la votación del usuario final a través de una aplicación web, involucrando al usuario en el proceso de evaluación.

Cada voto a favor o en contra suma o resta un punto al puntaje inicial, permitiendo que los resultados ordenados por puntaje suban o bajen según la opinión de la comunidad. De esta manera, se crea un sistema de calificación dinámico que combina la precisión del algoritmo con la perspectiva humana.

Los resultados, que están ordenados por puntaje, pueden subir o bajar dependiendo del número de puntuaciones que tengan a favor o en contra. La Figura 2.18 muestra un ejemplo de cómo las votaciones de usuario conllevan el cambio en el orden en el que se muestran los resultados, prevaleciendo en la parte superior los mejor puntuados.

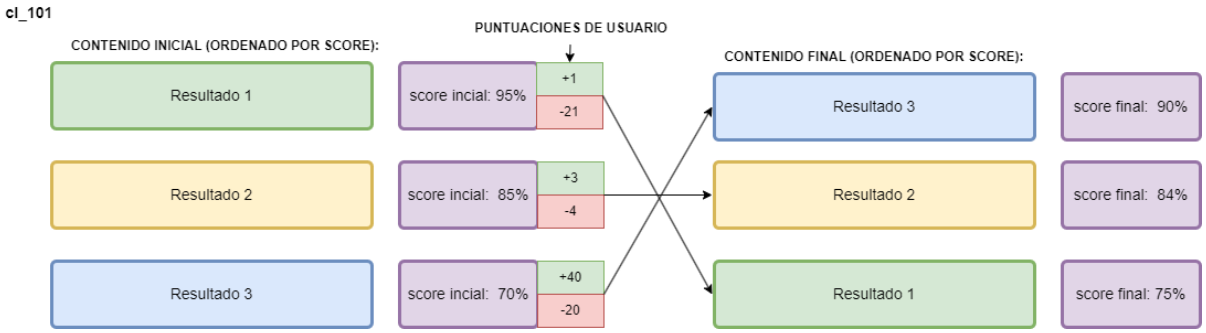

Figura 2.18. Impacto de las puntuaciones de usuario en la métrica compuesta (elaborado por el autor)

El puntaje ponderado inicial, es una semilla inicial. A medida que el sistema atrae más visitantes, las puntuaciones de los usuarios se integran al proceso, nutriendo y refinando la puntuación de las noticias.

A la par de la valoración del usuario se toma en cuenta los comentarios que pueda emitir y que son de ayuda para una retroalimentación en la construcción del sistema. Estos son recolectados mediante formularios que ayudan al usuario a expresar su opinión del porqué de su votación a favor o en contra.

2.4.2.5 Almacenamiento de Puntuaciones

A continuación, se describe el proceso de almacenamiento y cálculo de los diferentes puntajes utilizados en el proyecto. Se abarca el almacenamiento de los puntajes de relevancia de contenido, reputación de titular, métrica inicial compuesta y puntajes de usuario. También se explica el proceso para obtener el puntaje final utilizando todas las métricas consideradas y la vista que se utiliza para mantener la interfaz de usuario actualizada.

Los puntajes de relevancia de contenido obtenidos son almacenados en la base de datos, en el campo PUNTAJE_LINEA_GRUPO de la tabla LINEAS_NOTICIAS_CODIFICADO.

Los puntajes de reputación de titular son almacenados en el campo RANK de la tabla NOTICIAS_CODIFICADO.

La métrica inicial compuesta se calcula en base al proceso almacenado CALCULAR_PUNTAJE_FINAL_GRUPO y se almacena en el campo PUNTAJE_FINAL_GRUPO de la tabla LINEAS_NOTICIAS_CODIFICADO.

Los puntajes de usuario se almacenan en la tabla LINEAS_NOTICIAS_CALIFICACION. Cada uno de los registros dentro de esta tabla representan una puntuación de usuario. Esta tabla posteriormente puede ser filtrada mediante el conteo de sus registros para obtener el valor total de votaciones a favor o en contra.

Para obtener el puntaje final haciendo uso de todas las métricas consideradas se utiliza la vista VW_PUNTAJES que hace uso de las valoraciones de usuario (tabla LINEAS_NOTICIAS_CALIFICACION) y la métrica total de línea (campo PUNTAJE_FINAL_GRUPO de LINEAS_NOTICIAS_CODIFICADO).

La Figura 2.19 resume este proceso. La interfaz de usuario consulta la vista VW_PUNTAJES por lo que siempre se encuentra actualizada en la puntuación final de titulares.

2.4.3 Modelo Supervisado para Predecir Autoridad en Internet

Se ha desarrollado un modelo de regresión mediante aprendizaje de máquina supervisado. La Tabla 2.10 muestra las variables independientes utilizadas que corresponden a los

parámetros que describen la noticia publicada y la variable dependiente a estimar. El valor por predecir es la variable continua “Page Authority” que mide el grado de reputación que puede llegar a tener el titular en Internet.

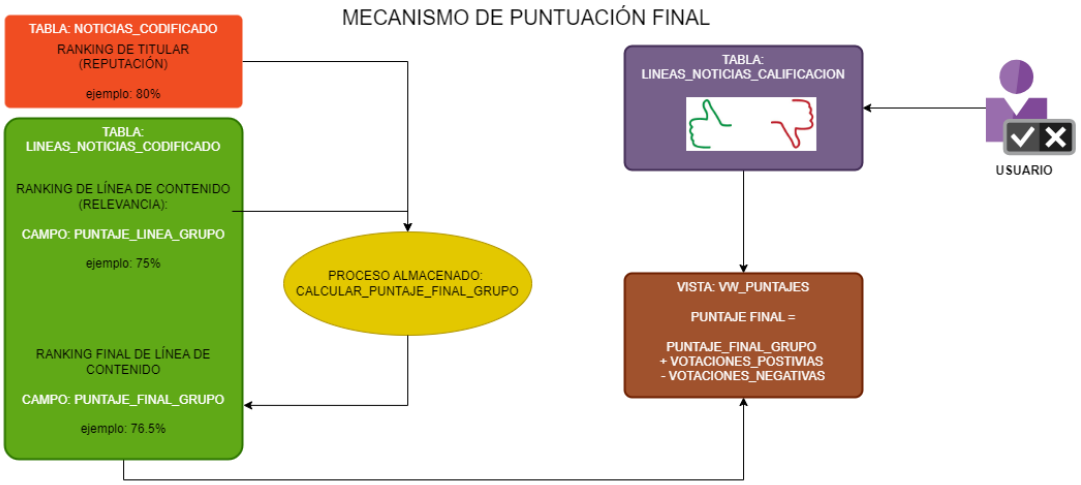

Figura 2.19. Mecanismo de puntuación final (elaborado por el autor)

Esta variable, ya sea que se tenga el valor real recolectado o el valor predicho para nuevos titulares que se integren al dataset, representará el 30% de la métrica compuesta inicial del sistema.

Tabla 2.10. Variables del modelo (elaborado por el autor)

Característica	Descripción	Tipo de Datos
autor_1	Autor principal	Texto
autor_2	Autor secundario	Texto
categoría	Tema de enfoque (política, tecnología, otros)	Texto
mes	Mes de publicación	Numérico
editorial	Editorial que ha publicado la noticia	Texto
num_lineas	Número de líneas que constituyen el contenido de la noticia	Numérico
num_car_titulo	Número de caracteres que contiene el título	Numérico
num_car_max_linea	Número de caracteres que tiene la línea de mayor longitud del contenido.	Numérico
rank	*Variable dependiente del modelo. Corresponde al valor de Autoridad de la Página (Page Authority). Una métrica compuesta de 0 a 100 que mide el grado de éxito de una página en Internet.*	*Numérico*

A diferencia del puntaje inicial, centrado en el contenido y su relevancia, este puntaje se fundamenta en las características que describen de manera global a la noticia, su temática, su origen y metadatos.

Para la construcción de los modelos se han considerado tres algoritmos de aprendizaje de máquina: Artificial Neural Networks (ANN), K Nearest Neighbors (KNN) y Support Vector Machine (SVM). Los modelos han sido entrenados en base al dataset recopilado y se ha comparado las métricas de evaluación para seleccionar el más adecuado.

2.4.3.1 Modelo ANN

Para el desarrollo del modelo en base a Artificial Neural Networks (ANN), se ha utilizado la librería Keras. Keras es una librería de código abierto para Python que provee una interfaz de la librería TensorFlow [78]. Keras permite la construcción, ejecución y gestión de redes neuronales.

El modelo de la red neuronal artificial se ha desarrollado con cuatro capas de funciones de densidad. La primera capa consta de nueve neuronas, la segunda de seis, la tercera de tres y la última de una neurona. Todas las capas de la red neuronal utilizan la función de activación GELU.

Se ha hecho uso del optimizador “adam”. Adaptive Moment Estimation (Adam) es un optimizador de descenso de gradiente estocástico que adapta la tasa de aprendizaje para cada parámetro, permitiendo obtener una menor pérdida [79].

Como métrica de pérdida, producto del entrenamiento, se ha tomado como referencia el error medio cuadrático (Mean Squared Error MSE). Para el entrenamiento del modelo se han realizado 250 iteraciones que han permitido disminuir el grado de pérdida a 2.8.

La Figura 2.20 muestra la curva de pérdida durante el entrenamiento del modelo. Esta gráfica ha sido computada para los datos de entrenamiento (loss) y para los datos de validación (val_loss). Se puede apreciar cómo la pérdida en la exactitud de la predicción decrece a medida que el modelo se entrena.

El valor del error absoluto promedio (Mean Absolute Error MAE) también fue calculado en base al uso de los datos de testing y es de 1,16. Este valor significa que el error promedio cometido durante el cálculo del Page Authority, métrica que computa la autoridad de la página en Internet del 0 al 100%, es del 1,16%. La raíz del error medio cuadrático (Root Mean Squared Error RMSE) para los datos de testing es del 1,99. Este valor nos permite ofrece una estimación del error promedio en base a la desviación estándar de los valores

residuales. El modelo ofrece una exactitud promedio del 91,85% durante la predicción del Page Authority para una noticia de una de las 24 editoriales analizadas.

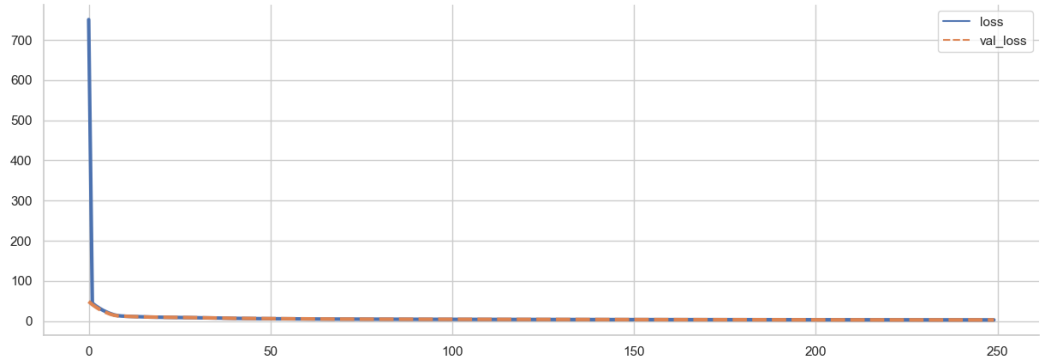

Figura 2.20. Pérdida durante el entrenamiento del modelo (elaborado por el autor)

2.4.3.2 Modelo KNN

El segundo algoritmo que se ha considerado para el desarrollo del modelo de predicción ha sido KNN. Este algoritmo requiere como parámetro de entrada el valor de k, que es el número de vecinos más cercanos a obtener durante su ejecución.

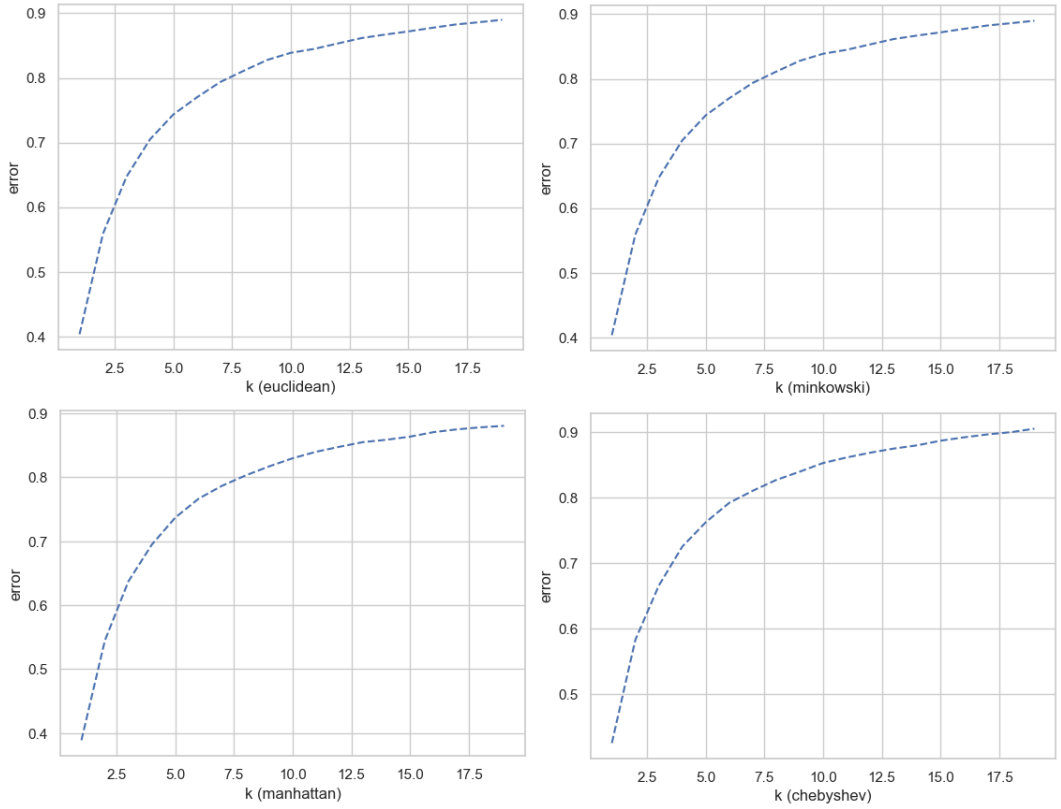

Figura 2.21. Valor promedio de error para distintos valores de k (elaborado por el autor)

Para identificar un valor de k óptimo se ha ejecutado un algoritmo para obtener, para los valores de k del 1 al 20, el valor promedio de error que se cometería con cada uno. La Figura 2.21 muestra los resultados obtenidos, se puede observar que el error aumenta con el aumento de k. Se opta por tomar un valor de k de tres, puesto que ofrece una buena relación de bajo grado de error frente a un número de k óptimo.

Para la construcción del algoritmo se ha ejecutado haciendo uso de las cuatro métricas soportadas por la librería scikit-learn implementada: Euclidiana, Minkowski, Manhattan y Chebyshev. La Tabla 2.11 muestra los resultados obtenidos con cada una de las métricas. Los mejores resultados han sido obtenidos haciendo uso de la métrica Manhattan. Esta métrica nos provee del menor error absoluto, menor raíz del error medio cuadrático y mayor exactitud durante la ejecución.

Las métricas de distancia para el cálculo de los vecinos son esenciales para determinar su similitud. Las métricas soportadas por la librería scikit-learn implementada son:

- **Distancia Euclidiana:** La más conocida e intuitiva. Se calcula como la raíz cuadrada de la suma de las diferencias al cuadrado entre las coordenadas de cada punto en cada dimensión [80].
- **Distancia Minkowski:** Es una generalización de la distancia euclidiana que permite un mayor control sobre la sensibilidad a la distancia en diferentes dimensiones [81].
- **Distancia Manhattan:** Es la suma de las diferencias absolutas entre las coordenadas de cada punto en cada dimensión. Se calcula como la suma de las distancias en cada dimensión sin considerar la raíz cuadrada [82].
- **Distancia Chebyshev:** Se define como la mayor diferencia entre las coordenadas de los dos puntos en cada dimensión [83].

Tabla 2.11. Ejecución KNN para cada métrica soportada (elaborado por el autor)

Métrica de Distancia	Error Absoluto Promedio	Raíz del Error Medio Cuadrático	Score de Predicción (%)
Euclidiana	1,02	1,86	86,71
Minkowski	1,02	1,86	86,71
Manhattan	0,93	1,68	89,50
Chebyshev	1,18	2,14	82,04

2.4.3.3 Modelo SVM

El tercer algoritmo con el que se ha construido el modelo es SVM. Para su configuración se ha hecho uso de un kernel RBF. Radial Basis Function (RBF), conocido como kernel gaussiano, corresponde a una función matemática que mapea los datos de entrada a un espacio de mayor dimensión donde se espera que sea más sencillo encontrar patrones lineales o no lineales para realizar la clasificación o regresión [84].

El error absoluto promedio obtenido tras evaluar el modelo entrenado es del 1,43 y la raíz del error medio cuadrático es del 2,54. Mientras que el porcentaje de exactitud del modelo es del 74,94%.

En el caso de los kernels lineal, sigmoide y polinomial, los resultados obtenidos no superaron un puntaje del 54,50%, por lo que se los descarta como opciones viables para el desarrollo del modelo.

2.5 Prueba y Evaluación

El proceso de pruebas y evaluación ha sido ejecutado para cada uno de los tres modelos desarrollados. La Tabla 2.12 muestra las consideraciones para la prueba y evaluación de cada modelo creado.

Tabla 2.12. Consideraciones de prueba y evaluación de modelos (elaborado por el autor)

Modelo	Pruebas y Evaluación
Agrupación de titulares	Se analizó el 13% de los 7.761 clústers formados, correspondientes a un total de 1.008 clústers analizados para verificar el correcto funcionamiento. En base a estos análisis se ha calibrado el valor de los umbrales.
Ranking de contenido	Tras haber realizado el ranking de noticias se ha obtenido las métricas descriptivas del ranking obtenido para poder observar los rasgos estadísticos de la predicción obtenida. También, se realizó la verificación en base a la interfaz de usuario creada para constatar la calidad de las puntuaciones.
Modelo supervisado para predecir autoridad en Internet	Se evaluaron los modelos de regresión creados en base al cálculo del error absoluto promedio, raíz del error medio cuadrático y el score de predicción. Para de esta manera seleccionar el algoritmo que mejores resultados ha producido.

Los resultados finales obtenidos tras haber refinado los modelos se detallan en el Capítulo 3.

2.6 Despliegue

En esta etapa se documentaron los modelos desarrollados, revisión de la literatura y se desplegó la interfaz gráfica de usuario.

Para la construcción de la aplicación web se ha hecho uso de la librería streamlit de Python, que permite la construcción de interfaces gráficas.

La aplicación web se compone de tres pantallas principales. La primera pantalla permite la búsqueda de titulares mediante un buscador estructurado que admite la búsqueda por palabras clave y semántica. La búsqueda muestra la puntuación de cada titular, permitiendo al usuario seleccionar el que desea leer y acceder al contenido de su clúster completo. La Figura 2.22 muestra la pantalla de búsqueda de titulares.

Figura 2.22. Pantalla de búsqueda de titulares (elaborado por el autor)

El motor de búsqueda utilizado en la aplicación web es similar al que se empleó para la clusterización de titulares, descrito en la sección 2.4.1.1. A este buscador se le ha retirado los umbrales de similitud textual (65%) y similitud semántica (67%) que limitaban las respuestas durante la clusterización para aumentar la exactitud. Esto por cuanto el objetivo de este buscador es el de mostrar la mayor cantidad de información similar organizada por puntuación y no se requiere sea tan restrictivo como se requirió durante la etapa de clusterización. Los valores con menor precisión, filtrados por umbrales en la etapa de clusterización, se mostrarán en los últimos resultados con una puntuación inferior.

Para este caso, la cadena de texto a ser utilizada por el buscador es la que ingresa el usuario haciendo uso de la interfaz gráfica. Por lo tanto, se eliminan las palabras vacías de la cadena de texto y luego se convierte en un vector de 768 dimensiones para que pueda ser procesada por el motor de búsqueda. Este proceso permite que la aplicación web encuentre de manera eficiente los titulares más relevantes para la consulta del usuario.

Durante el proceso de búsqueda se hace uso del índice de titulares que fue creado en la instancia Elasticsearch. Este índice contiene la información de todos los titulares o encabezados de las noticias.

Los resultados tras ejecutar la búsqueda son ordenados en base a su puntuación de manera descendente, posicionando los mejor puntuados en la parte superior de la tabla de resultados.

Entre los resultados obtenidos al ejecutar la búsqueda se retorna el campo “id” del documento Elasticsearch que fue almacenado de manera que coincida con el campo ID de noticia de la base de datos relacional. De esta manera es posible obtener la información de clúster para la noticia que el usuario decida seleccionar.

Al seleccionar un titular, se abre la pantalla con el contenido del clúster. La pantalla indica el clúster al que pertenece el titular y muestra las líneas de contenido ordenadas por puntuación final, facilitando la identificación de la información relevante. La Figura 2.23 muestra esta pantalla, donde se observa el contenido organizado por puntuación final, junto al titular original y el diario del cual se ha obtenido la información. Desde esta pantalla se puede seleccionar puntuar las líneas de contenido específicas.

La pantalla de contenido hace uso de la vista VW_PUNTAJES, para obtener la lista de contenidos de un clúster con sus determinadas puntuaciones.

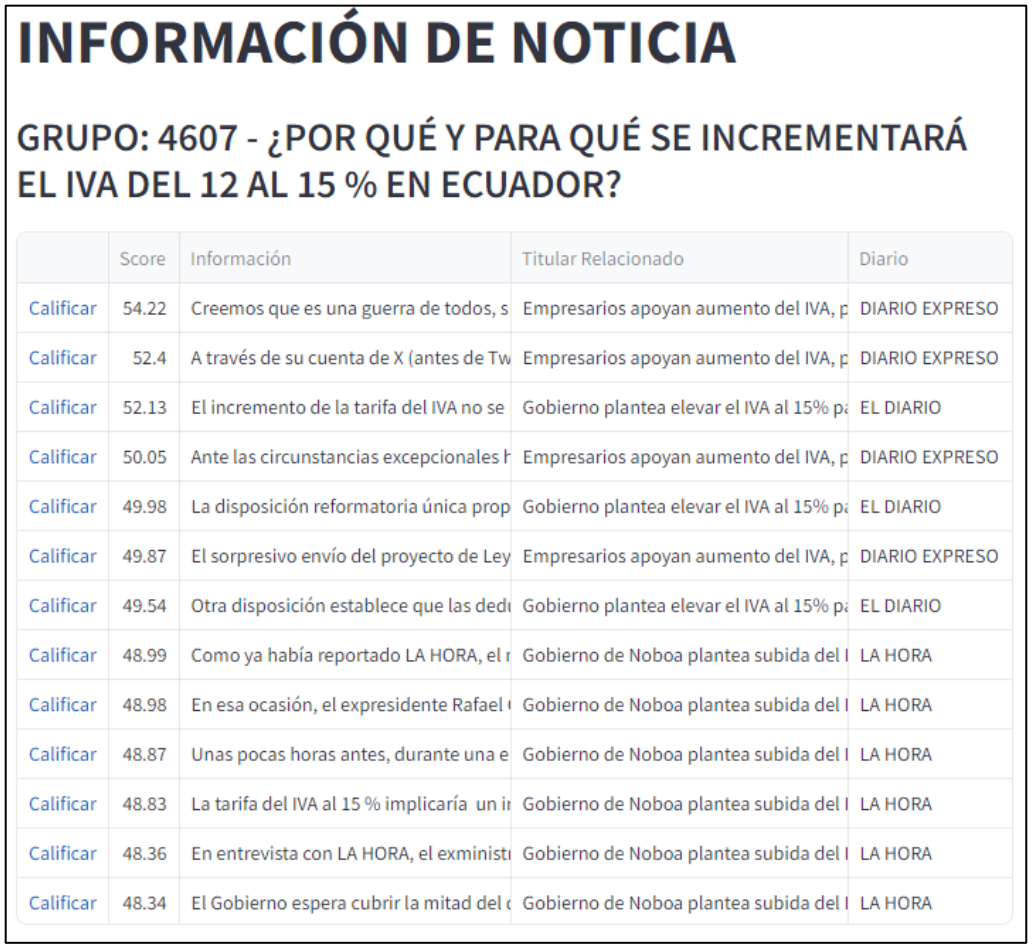

Figura 2.23. Pantalla de contenido de clúster de titular seleccionado (elaborado por el autor)

Al seleccionar una línea de contenido para calificar, se presenta un formulario donde el usuario puede ingresar su puntuación y comentarios o razones de su valoración. La evaluación debe basarse en la utilidad del contenido y no en la reacción emocional que este genere al usuario. Esto es indicado de manera oportuna en la pantalla de puntuación de usuario. La Figura 2.24 muestra esta pantalla.

La valoración de usuario es un valor booleano que indica si el contenido le ha gustado o no. Las razones o motivos corresponden a un campo multilínea de texto libre en el que el usuario puede ingresar la información que crea pertinente que ayude a mejorar el sistema.

Tras haber puntuado el contenido, se muestra una pantalla de agradecimiento para el usuario final. La Figura 2.25 muestra esta pantalla.

Figura 2.24. Pantalla de puntuación de contenido de usuario final (elaborado por el autor)

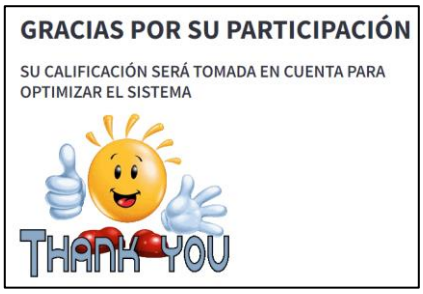

Figura 2.25. Pantalla de agradecimiento a usuario por su participación (elaborado por el autor)

3. RESULTADOS

3.1 Agrupación de Titulares

Los resultados del proceso de asignación de clústers se almacenan de forma organizada en la base de datos. Para ello, se utiliza el campo GRUPO dentro de la tabla NOTICIAS_CODIFICADO. Este campo guarda el valor numérico que identifica el clúster al que ha sido asignada cada noticia, permitiendo la recuperación de la información.

La Figura 3.1 ilustra la arquitectura de hardware sobre la que se ejecutó el proceso de clusterización. Para el almacenamiento de datos, se ha implementado en un servidor la base de datos relacional y el motor de búsquedas. Adicionalmente, se cuenta con un servidor independiente para la ejecución del algoritmo de clusterización. Esta configuración ha procesado ininterrumpidamente los grandes volúmenes de datos.

Figura 3.1. Hardware utilizado para la ejecución del proceso de clusterización de titulares (elaborado por el autor)

El proceso de clusterización permitió formar 7.761 clústers de noticias. Luego de la ejecución del proceso de clusterización, se llevó a cabo una evaluación meticulosa del 13% de los clústers creados, con el objetivo de verificar la calidad de su conformación. Esta revisión exhaustiva permitió confirmar la óptima calidad de los clústeres y definir los umbrales adecuados para el funcionamiento del algoritmo, garantizando así la calidad de la agrupación de las noticias.

La Figura 3.2 presenta algunos ejemplos de los clústers conformados, evidenciando la clara relación temática entre los titulares agrupados. Cabe destacar que la configuración de los umbrales juega un rol fundamental en la granularidad de la clusterización. Disminuir los valores de los umbrales genera clústeres con una tendencia más general, mientras que aumentarlos conduce a clústeres con una mayor especificidad temática. La evaluación de la calidad de la clusterización debe estar en función de los objetivos del negocio. En este caso, la agrupación por temática ha sido óptima con los valores umbrales propuestos, logrando una clusterización adecuada y alineada con las metas del proyecto.

grupo numeric (8)	titular character varying (250)
8	Unidad del Trolebús se incendió en Quito debido a una falla mecánica
8	Conato de incendio en unidad de Trolebús tras falla mecánica
8	Quito: emergencia en el Trolebús por una avería mecánica
8	Una falla mecánica causó un incendio en unidad del Trolebús, en Quito
8	Incendio de unidad de Trolebús en Quito se atribuye a falla mecánica

grupo numeric (8)	titular character varying (250)
20	Viceministro de Gobernabilidad, Esteban Torres, aseguró que se tiene previsto reducir la nómina estatal
20	"El Gobierno tiene previsto reducir su nómina", dice viceministro Esteban Torres

grupo numeric (8)	titular character varying (250)
35	Viral.- El enfrentamiento de un grupo de chinos y un pianista en una estación de tren de Londres
35	El curioso cruce entre un grupo de chinos y un pianista youtuber inglés en una estación de tren en Londres

grupo numeric (8)	titular character varying (250)
55	Conozca lo que dispone el estado de excepción decretado por el Presidente Daniel Noboa
55	¿Qué pasará con los eventos públicos durante el estado de excepción decretado por Daniel Noboa?
55	¿Qué implica el estado de excepción decretado por Daniel Noboa?

Figura 3.2. Ejemplos de clústers conformados (elaborado por el autor)

3.2 Ranking de Contenido

El cálculo de las puntuaciones en base a métricas compuestas ha requerido de una infraestructura de hardware para ser ejecutado por la naturaleza estructurada de las métricas y el gran volumen de datos a ser procesado.

La Figura 3.3 ilustra el hardware que se ha hecho uso para ejecutar el cómputo de la métrica inicial de puntuación. Se ha hecho uso de un servidor para datos y tres nodos de iguales características para ejecutar los cálculos.

La distribución de la carga en los tres nodos se ha realizado de manera equitativa. Distribuyendo los clústers de manera administrativa entre cada uno de tres los nodos.

El servidor de datos ha sido implementado con el sistema operativo Ubuntu Server 22.04 y contiene las instancias de base de datos y motor de búsquedas Elasticsearch. Los nodos han sido implementados mediante la plataforma Google Colab que ofrece una gestión adecuada de los recursos para operaciones de esta naturaleza.

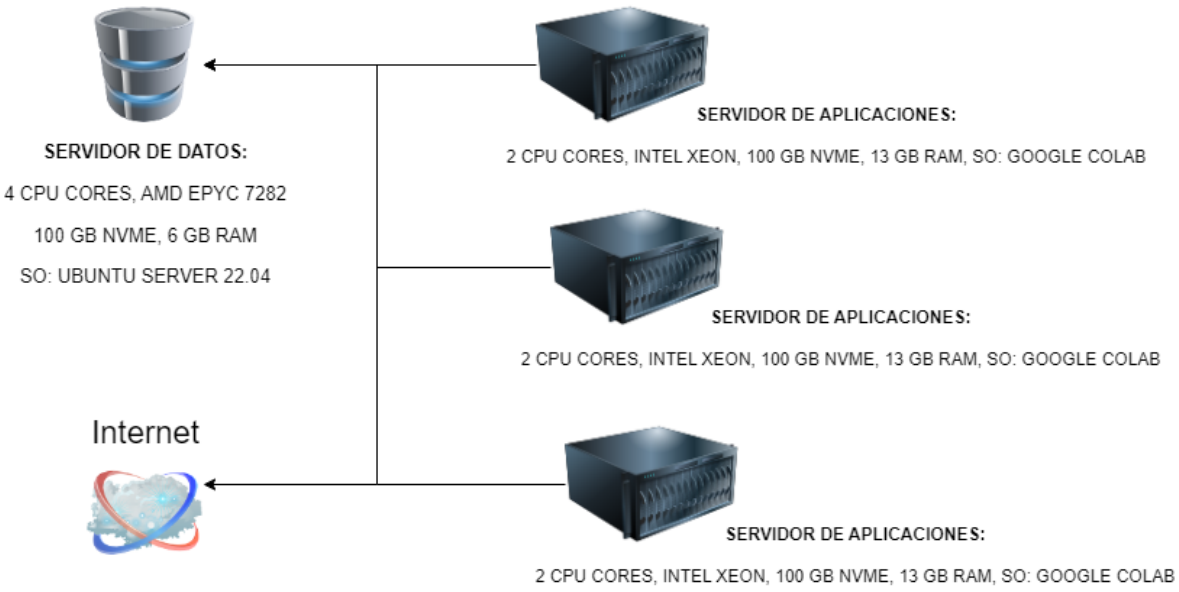

Figura 3.3. Hardware utilizado para calcular el puntaje inicial (elaborado por el autor)

El proceso de puntuación implementado ha permitido calcular un puntaje inicial para cada una de las 537.146 líneas de contenido analizadas. Este proceso, que se basa en una serie de algoritmos y técnicas predefinidas, tiene como objetivo organizar la información de forma descendente, priorizando la más relevante. El puntaje final obtenido para cada línea es un valor numérico que se encuentra dentro del rango de 0 a 100 puntos, donde un valor mayor indica una mayor relevancia.

Estos puntajes iniciales son de gran utilidad para identificar la información más relevante dentro de un clúster de noticias determinado. La Figura 3.4 muestra la distribución por rangos de los puntajes obtenidos, lo que permite observar una tendencia general en la dispersión de la información. Cabe destacar que estos puntajes no son definitivos, y posteriormente serán complementados con la participación del usuario a través de un proceso de votación.

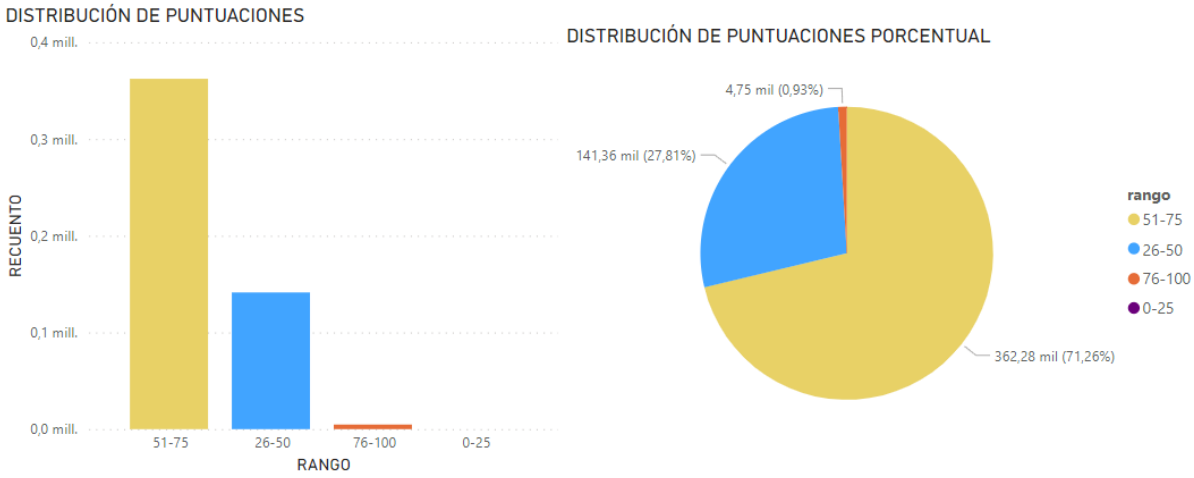

Figura 3.4. Distribución de puntajes de contenido (elaborado por el autor)

La votación del usuario tiene como objetivo incorporar la perspectiva humana en el proceso de evaluación de la información. De esta forma, se busca obtener una visión más completa y enriquecida de la relevancia del contenido, considerando no sólo los aspectos técnicos, sino también las preferencias y necesidades de los usuarios finales.

La Figura 3.5 muestra información descriptiva referente a los resultados de las puntuaciones de contenido. Para el puntaje final computado de las líneas de contenido se tiene un promedio de 54,13, un valor mínimo de 31, un valor máximo de 87, una desviación estándar de 7,07 y cuartiles de 49 (25%), 54 (50%) y 58 (75%).

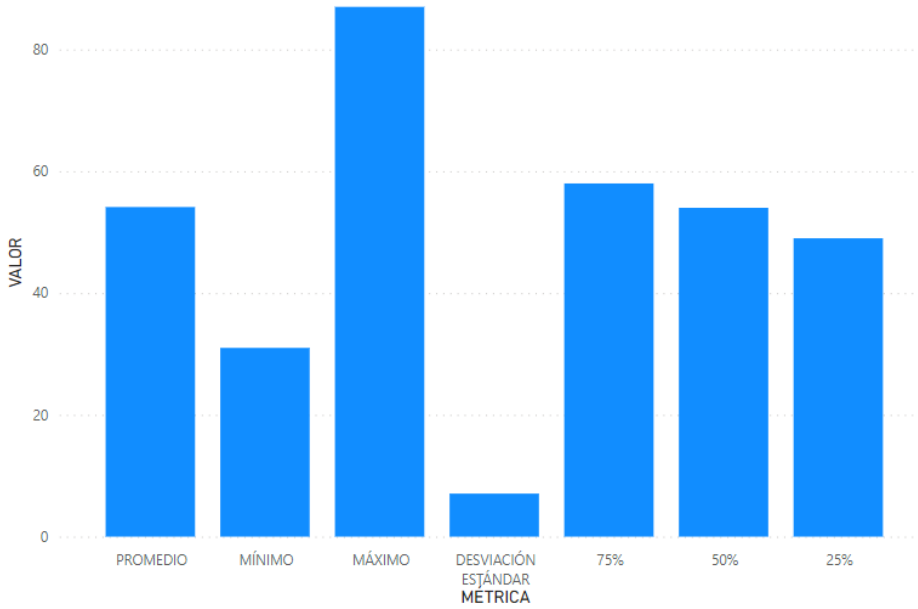

Figura 3.5. Métricas descriptivas del puntaje de contenido (elaborado por el autor)

3.3 Modelo de Aprendizaje Supervisado para Predecir Autoridad en Internet

La Tabla 3.1 muestra los resultados de error absoluto promedio, raíz del error medio cuadrático y score de predicción de cada uno de los modelos. En base a los resultados obtenidos se puede establecer que el modelo óptimo es el construido en base a redes neuronales artificiales.

Tabla 3.1. Resultados tras la evaluación de modelos (elaborado por el autor)

Tipo de Modelo	Error Absoluto Promedio	Raíz del Error Medio Cuadrático	Score de Predicción (%)
ANN	1,16	1,99	91,85
KNN (Manhattan)	0,93	1,68	89,50
SVM (RBF)	1,43	2,54	74,94

En los resultados obtenidos, podemos observar que el mayor score de predicción se obtiene en base al modelo ANN que es del 91,85%, seguido del modelo KNN con 89,50% y finalmente el modelo SVM con 74,94%. La diferencia en el score de predicción permite posicionar como mejor modelo al realizado en base a ANN con por lo menos una diferencia del 2,35% del siguiente modelo (KNN). En lo referente al cálculo del error, las métricas obtenidas han sido similares en los tres modelos teniendo los mejores resultados con el modelo KNN que presenta un error absoluto promedio (MAE) de 0,93 y la raíz del error medio cuadrático (RMSE) del 1,68. El modelo ANN presenta la segunda mejor métrica de MAE y RMSE, similar al modelo KNN, con el 1,16 y 1,99 respectivamente. Finalmente, el modelo SVM tiene un mayor MAE y RMSE del 1,43 y 2,54. Los valores obtenidos por estos tres modelos para el score de predicción, MAE y RMSE exhiben niveles muy aceptables en la calidad de los modelos.

Las Redes Neuronales Artificiales se destacan por su capacidad para modelar relaciones complejas y no lineales entre las variables, lo que se traduce en una mayor precisión en comparación con otros modelos como KNN y SVM. En este caso, el modelamiento en base a ANN ha sido de gran utilidad para obtener un resultado más satisfactorio.

Si bien es cierto que ANN ha superado las métricas de evaluación de KNN y SVM, se debe destacar que los valores obtenidos en base a estos dos algoritmos también han sido

considerablemente similares. Este análisis ha permitido percibir y cuantificar el grado de utilidad de los algoritmos de aprendizaje de máquina, que ha sido bastante representativo.

3.4 Caso de Uso: Uso del Sistema para Buscar una Noticia

A modo de ilustración, se presenta un caso de uso para la búsqueda de noticias. En este ejemplo, se detalla la diversa información que el sistema proporciona, incluyendo las secciones detalladas que componen la estructura de cada pantalla.

Al iniciar, se ingresa el titular deseado a través de la interfaz gráfica de usuario. Tras obtener los resultados, se despliega una lista ordenada de manera descendente por puntuación, donde el titular con mayor puntaje se ubica en la primera posición. Como se observa en la Figura 3.6, la pantalla muestra el listado de resultados y la información asociada a cada uno.

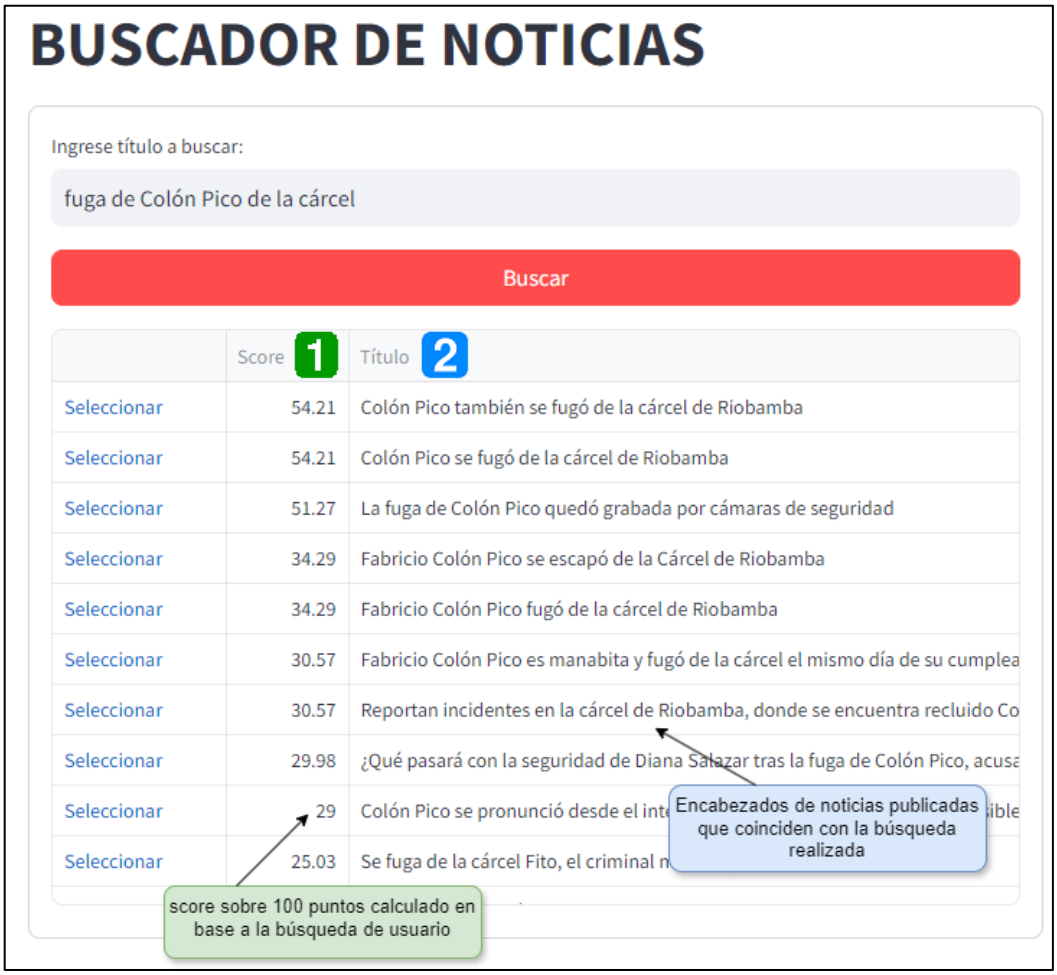

Figura 3.6. Elementos de la pantalla de búsqueda de titulares (elaborado por el autor)

La información mostrada es la siguiente:

1. **Score:** Este indicador refleja la relevancia del titular con respecto a la consulta del usuario, alcanzando un máximo de 100 puntos. Se obtiene tras eliminar palabras vacías del texto ingresado y realizar una búsqueda textual y semántica en el índice de titulares.

2. **Título:** Corresponde al encabezado de la noticia tal cual fue publicado en Internet.

Al pulsar el botón "Seleccionar", se despliega una pantalla que presenta la información consolidada del clúster de noticias seleccionado. En esta pantalla, el usuario puede observar el contenido de las noticias que conforman dicho clúster, organizadas de manera descendente según su puntuación. Dado que un clúster está compuesto por titulares de diversos medios de prensa, la información más relevante se ubica en la parte superior de la lista. Esta lista actúa como un filtro, priorizando la información de mayor importancia en la parte superior. La Figura 3.7 ilustra los resultados de la consolidación y el ranking de información para el clúster de la noticia seleccionada.

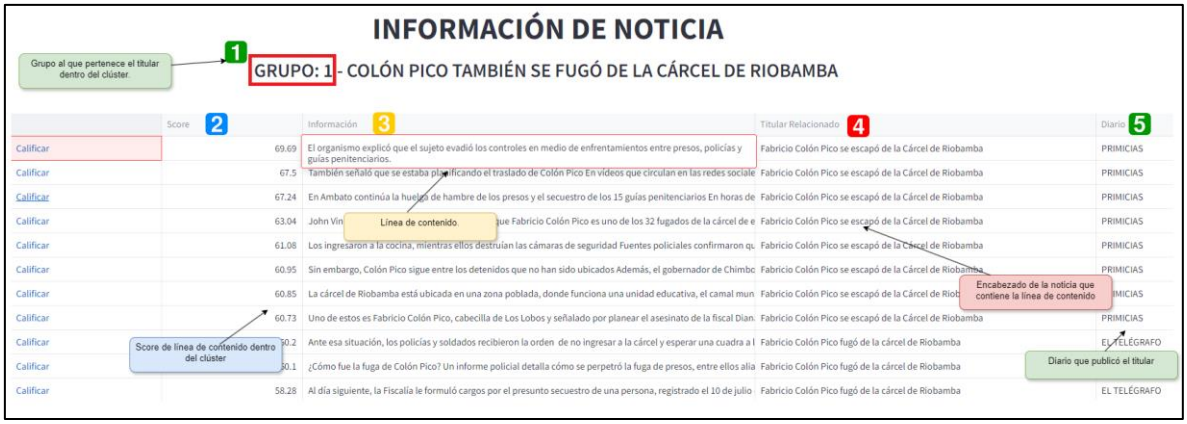

Figura 3.7. Elementos de la pantalla de contenido de clúster (elaborado por el autor)

La información que se puede obtener de la pantalla de contenido es la siguiente:

1. **Grupo:** Este identificador único indica el número de clúster o grupo al que pertenece la noticia. En este caso particular, corresponde al primer grupo que se ha formado, denominado grupo 1.

2. **Score:** Este indicador refleja la relevancia de cada una de las líneas de contenido dentro del clúster, evaluadas inicialmente sobre 100 puntos. Se trata de una métrica compuesta estimada que considera diversos factores. En este caso, la línea de contenido con mayor puntuación ha obtenido el valor de 69,69 puntos.

3. **Información:** Corresponde a la línea de contenido.
4. **Titular Relacionado:** Este campo presenta el texto del encabezado de la noticia en la que se publicó la línea de contenido. Es decir, indica el título de la noticia que contiene la línea de texto en cuestión.
5. **Diario:** Corresponde al nombre del diario en el que fue publicada la línea de contenido.

La puntuación inicial asignada a las líneas de contenido es una estimación sobre 100 puntos que sirve como punto de partida para que los usuarios puedan expresar su propia valoración. Esta puntuación inicial representa una base “semilla” que permite la organización inicial de las líneas de contenido y que se complementará con las opiniones de los usuarios finales.

Al pulsar el botón "Calificar", se despliega la pantalla de calificación de usuario final. Esta interfaz permite al usuario expresar su opinión sobre el contenido presentado, indicando si le ha resultado de utilidad o no. La Figura 3.8 muestra los elementos que componen esta pantalla y sus respectivas funciones.

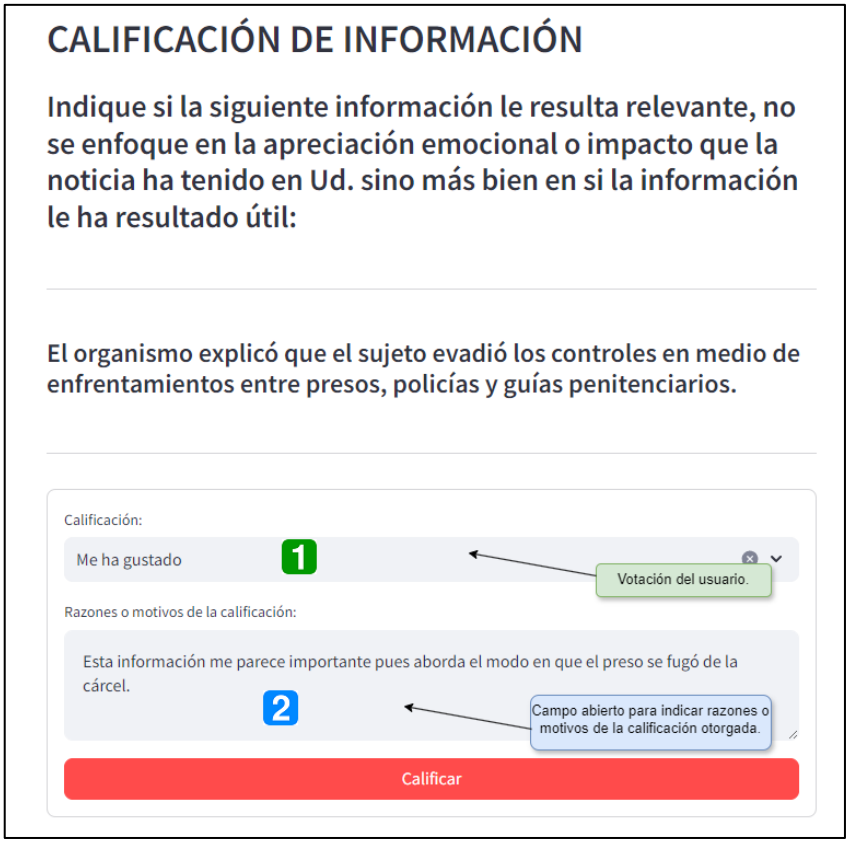

Figura 3.8. Elementos de la pantalla de puntuación de usuario (elaborado por el autor)

Los elementos que componen esta pantalla son los siguientes:

1. **Calificación:** Esta variable binaria refleja la opinión del usuario final sobre el contenido presentado, indicando si le ha resultado satisfactorio o no. La evaluación debe basarse en la relevancia de la información para el usuario, no en si la noticia en sí misma es de su agrado. En el ejemplo, la selección de "Me ha gustado" contribuye a incrementar la puntuación de la línea de contenido seleccionada.

2. **Razones o motivos de la calificación:** Este campo abierto brinda al usuario la oportunidad de expresar las razones o motivos que respaldan la calificación otorgada al contenido seleccionado. En el ejemplo, se observa un comentario que aporta valiosa retroalimentación sobre la relevancia de la información.

Luego de calificar el contenido seleccionado y regresar a la pantalla que muestra el ranking del contenido de clúster, se aprecia un incremento en la puntuación de la línea de contenido que fue calificada. Esto confirma que la valoración del usuario final ha sido tomada en cuenta. En este caso específico, la puntuación de la línea de contenido ha incrementado de 69,69 a 70,69 como muestra la Figura 3.9. Los incrementos y decrementos en base a las puntuaciones de usuario permiten refinar la organización del contenido, proporcionando una mayor exactitud al mostrar el contenido más relevante.

Figura 3.9. Incremento en el score de contenido tras ser puntuado por el usuario (elaborado por el autor)

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

Tras haber culminado el presente proyecto se ha podido concluir que:

- El modelo de lenguaje BERT se destaca por su alta precisión en la interpretación del lenguaje humano. Las variantes del algoritmo BERT han logrado superar significativamente las capacidades del modelo original, mejorando la precisión en la contextualización y su rendimiento para procesar oraciones largas. Estas características y mejoras han convertido a BERT y sus variantes en herramientas poderosas y versátiles para diversas tareas de procesamiento del lenguaje natural, con un alto grado de precisión y eficiencia, ampliando su aplicabilidad en diferentes sectores.

- Los Large Language Models están revolucionando el panorama de las aplicaciones. Su alta precisión, en constante evolución, permite una mejor comprensión del lenguaje natural, lo que se traduce en interacciones más fluidas e intuitivas con los usuarios. En el futuro, estos modelos irán adquiriendo mayor protagonismo por permitir un acceso más intuitivo a los usuarios y facilidad en las comunicaciones.

- Las operaciones con transformaciones vectoriales, si bien son esenciales para el funcionamiento de los modelos de lenguaje de gran tamaño, conllevan un alto costo computacional. Para optimizar la eficiencia en la ejecución de estos modelos, es importante seleccionar cuidadosamente el software y el hardware que se utilizará. Para las herramientas de software se deben seleccionar algoritmos, transformadores, bases de datos e indexadores que presenten buenos rendimientos. También se debe seleccionar un hardware adecuado. Esta combinación de software y hardware permitirá obtener una eficiencia óptima durante la ejecución de los modelos de lenguaje de gran tamaño.

- La elección del hardware en cada fase del proceso se basa en dos variables fundamentales: la cantidad de registros a procesar y la complejidad del procedimiento. En la etapa de clusterización de titulares, donde el volumen de datos y la complejidad de cálculo son menores, se utilizó un único nodo de procesamiento y un nodo de datos. En cambio, para la etapa de puntuación de contenido, que requiere procesar una mayor cantidad de datos y presenta mayor complejidad de cálculo, se implementaron tres nodos de procesamiento y un nodo de datos. Esta

configuración permite optimizar el rendimiento y la eficiencia del proceso en cada etapa, asegurando un procesamiento adecuado y preciso de la información.

- La evaluación óptima de los titulares de noticias requiere considerar una serie de características que los definen. No se puede confiar la puntuación a una sola métrica, ya que no captura su complejidad por completo. La implementación de métricas compuestas que combinan diversas características de las noticias permite una mejor precisión y predicción del contenido que puede ser más importante para los usuarios. Estas métricas ofrecen una visión más holística de la calidad de los titulares al identificar con mayor exactitud aquellos que son más relevantes, informativos y atractivos para la audiencia objetivo, brindándoles una mejor experiencia final.

- La inteligencia artificial y el machine learning están revolucionando el mundo tecnológico al posibilitar análisis mucho más profundos que los métodos tradicionales. Esta capacidad permite obtener conocimiento valioso de grandes cantidades de datos en tiempos razonables, lo que abre un sinfín de posibilidades en diversos campos.

- Los modelos de aprendizaje automático supervisados, como el utilizado para predecir el valor de puntuación basado en Page Authority, son herramientas de gran utilidad en el análisis predictivo. Permiten realizar estimaciones precisas de información no disponible a partir de datos históricos, posibilitando la anticipación a eventos y la toma de decisiones estratégicas. Estos modelos permiten identificar de las variables de entrada las que generan mejores resultados, permitiendo enfocar los esfuerzos en optimizarlas y obtener resultados más exitosos.

- Los resultados obtenidos han demostrado los beneficios de los modelos de aprendizaje automático para las tareas de clusterización y ranking de noticias digitales. Estos modelos lograron obtener resultados óptimos en tiempos razonables, lo que representa una gran ventaja en comparación con el tiempo y el esfuerzo que requeriría un ser humano para realizar las mismas tareas. La eficiencia y precisión de los modelos de aprendizaje automático los convierten en herramientas valiosas para el procesamiento y análisis de grandes cantidades de información en el ámbito de las noticias digitales.

- La implementación de la inteligencia artificial en los negocios trae consigo una serie de beneficios tangibles. En primer lugar, permite reducir significativamente los

costos de operación al automatizar tareas repetitivas y optimizar procesos. Esto libera tiempo y recursos humanos que pueden ser destinados a actividades más estratégicas.

- Los modelos de inteligencia artificial están reemplazando al análisis tradicional de datos gracias a su precisión en los resultados, la reducción de costos que generan, la mayor confiabilidad que desarrollan con el tiempo y el creciente número de usuarios que adoptan estas tecnologías. A diferencia de los métodos tradicionales, la inteligencia artificial ofrece un análisis más profundo y automatizado, lo que se traduce en una mejor toma de decisiones y un mayor retorno de la inversión.

- Es necesario invertir un esfuerzo en garantizar contenido de calidad a los usuarios finales. Esto muchas veces implica la implementación de diversos módulos enfocados en tareas específicas y que se integren a un sistema. Aunque el diseño puede ser más complejo, este enfoque permite a los usuarios percibir mejores resultados, lo que se traduce en una mayor calidad. La satisfacción del usuario final se ve potenciada por la experiencia fluida y eficiente que ofrece un sistema integrado con módulos especializados en la entrega de contenido de alta calidad.

- La aplicación de la metodología CRISP-DM ha sido fundamental para el desarrollo óptimo del proyecto. Esta metodología ofrece la flexibilidad necesaria entre sus etapas, brindando una estructura organizada y robusta que ha permitido abordar el proyecto de forma estructurada. La gradualidad en la evaluación de la eficacia de los modelos y la flexibilidad para retornar a etapas anteriores son aspectos clave que facilitaron el éxito del proyecto.

- Internet alberga un sinfín de información valiosa que puede ser recopilada mediante técnicas de web scraping. Esta información, al ser procesada y sintetizada, permite crear contenido de gran valor que, a simple vista, sería difícil de asimilar para el usuario navegando por la web de forma tradicional.

- La elección de la técnica de web scraping adecuada es fundamental para el éxito del proceso. Diversos factores deben ser considerados, como la tecnología del sitio web objetivo y el tipo de datos a obtener. Una técnica bien seleccionada facilitará la extracción de la información, optimizando el tiempo y los recursos empleados.

- La eficacia del modelo predictivo desarrollado se ha visto potenciada por la alta capacidad de las redes neuronales artificiales para modelar relaciones complejas

no lineales, junto a su arquitectura de múltiples capas. Esta combinación ha permitido obtener resultados robustos y confiables, evidenciando el potencial de las ANN como herramienta de gran utilidad en el ámbito del modelado predictivo. La precisión y confiabilidad de las ANN las convierten en una opción invaluable para la toma de decisiones estratégicas en diversos campos, posibilitando la anticipación de escenarios futuros con un alto grado de certeza.

- Elasticsearch se destaca como una herramienta poderosa para el análisis y búsqueda de patrones en diversos tipos de datos. Su arquitectura escalable y distribución open source la convierten en una solución ideal para empresas de todos los tamaños. Es capaz de integrarse para usar algoritmos de aprendizaje de máquina como KNN y soportar una variada cantidad de operaciones.

- La calidad de los datos es un pilar fundamental para el éxito del entrenamiento de modelos. La etapa de preparación de datos, que incluye la limpieza y el tratamiento inicial de la información, es muy importante para garantizar que los modelos aprendan de manera adecuada. Un alto grado de calidad en los datos de entrada se traduce directamente en un mayor éxito en el entrenamiento y, por ende, en mejores resultados y predicciones más precisas. Es vital dedicar tiempo y esfuerzo a la preparación de los datos para obtener el máximo potencial de los modelos y alcanzar los objetivos del proyecto.

- La clusterización depende en gran medida del valor del umbral utilizado en la definición de las búsquedas. Un umbral alto genera clústers con un alto grado de similitud entre sus elementos, mientras que un umbral bajo produce clústers de temática más general que ofrecen una visión global del conjunto de datos. La elección del umbral adecuado es importante para obtener clústers que se ajusten a las necesidades específicas del proyecto.

4.2 Recomendaciones

Tras haber culminado el presente proyecto se puede recomendar:

- En un mundo cada vez más digitalizado, la inteligencia artificial y el aprendizaje automático se perfilan como tecnologías disruptivas con un enorme potencial para transformar las industrias y la sociedad en su conjunto. Ante este panorama, se recomienda la creación de más asignaturas relacionadas con estas áreas en los

planes de estudio ya que se convierte en una necesidad imperiosa para el campo laboral.

- La eficiencia y precisión del proceso de recopilación dependen en gran medida de la elección de la técnica adecuada de web scraping, por lo que se recomienda un análisis meticuloso de las opciones disponibles antes de iniciar la extracción de datos.
- Al trabajar con grandes cantidades de datos, el almacenamiento en una base de datos se convierte en una práctica esencial. Esto no solo facilita la consulta de los resultados en fases posteriores, sino que también permite un análisis más profundo y accesible mediante el uso de consultas SQL.
- En el contexto del aprendizaje supervisado, se recomienda entrenar al menos tres modelos diferentes. Esta práctica se fundamenta en la posibilidad de comparar las métricas de cada modelo utilizando la etiqueta conocida como referencia. Esta comparación permite identificar el modelo con el mejor rendimiento, es decir, aquel que ofrece una estimación más precisa y confiable.
- La organización del código es un factor fundamental para su posterior mantenimiento y comprensión. Un código bien organizado facilita las modificaciones y actualizaciones, a la vez que permite un mejor entendimiento de este por parte de los programadores.
- La elaboración de la documentación del proyecto no debe verse como una tarea disociada del desarrollo de este, sino como una herramienta activa que evoluciona a la par que el proyecto. Esta práctica aporta un valor inestimable al facilitar la consulta, el seguimiento y el apoyo en las fases sucesivas.
- Existen muchas características valiosas que pueden obtenerse de los Large Language Models. Estos pueden ser utilizados para un sin número de aplicaciones. Su capacidad para comprender y generar texto de manera similar a la humana abre un abanico de posibilidades en una amplia gama de aplicaciones. Es recomendable explorar estas posibilidades en futuros proyectos.
- A la hora de realizar un proyecto, es recomendable la implementación de una metodología robusta y probada que es fundamental para minimizar riesgos y asegurar el éxito de la iniciativa. En este caso, se ha optado por la metodología

CRISP-DM, una de las más reconocidas y utilizadas en el ámbito del análisis de datos.

- Previo a la ejecución de los modelos, es recomendable realizar un análisis exhaustivo de las diferentes técnicas y opciones de desarrollo disponibles. Este proceso de investigación permitirá seleccionar la alternativa óptima para alcanzar los objetivos específicos del proyecto.
- La provisión de una GUI para la visualización de los resultados es una práctica altamente recomendable. Una GUI bien diseñada facilita la comprensión de la información, permite una interacción intuitiva con los datos y ofrece una experiencia de usuario más rica. En caso de no ser posible implementar una GUI, se debe garantizar la existencia de un mecanismo accesible para la visualización de la información. Este mecanismo puede ser un archivo de texto, una hoja de cálculo, un panel de control o cualquier otra herramienta que permita a los usuarios acceder y comprender los resultados de manera eficiente.
- Se recomienda priorizar el uso de las variantes del modelo BERT en lugar del modelo original. Estas variantes, como RoBERTa, XLNet o sBERT, han sido desarrolladas para solventar diversos problemas que presenta BERT original, como la dificultad de entrenamiento o la baja eficiencia computacional.

5. REFERENCIAS BIBLIOGRÁFICAS

[1] T. Silva, & A. Cuadra, Relación entre periodismo e internet, tras el surgimiento de la prensa electrónica: análisis del diario el Mostrador (Disertación Doctoral, Universidad Academia de Humanismo Cristiano), 2001, pp. 1-5. [Online]. Disponible en: http://bibliotecadigital.academia.cl/xmlui/handle/123456789/5540

[2] J. Ribeiro, Big data for executives and market professionals, Publicación Independiente, 2019, pp. 14-35.

[3] Y. Huamani & F. Montesinos, Evolución de la prensa escrita al medio digital en los diarios El Sol y Diario del Cusco 2012-2017, 2023.

[4] F. Carrera, Los medios y su importancia, ene. 3, 2023. [Online]. Disponible en: https://www.lahora.com.ec/de-la-audiencia/los-medios-y-su-importancia

[5] A. Tatar, P. Antoniadis, M. Amorim & S. Fdida, Ranking News Articles Based on Popularity Prediction, 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Istanbul, Turkey, 2012, pp. 106-110, doi: 10.1109/ASONAM.2012.28

[6] G. Llobet, Los agregadores de noticias y la prensa escrita en la era de Internet. Papeles de economía española, 2015, p. 172. [Online]. Disponible en: https://www.proquest.com/docview/1774557228/fulltextPDF/894866668FC54149PQ/1?accountid=32496

[7] E. Vargiu, & M. Urru, Exploiting web scraping in a collaborative filtering-based approach to web advertising. Artif. Intell. Res., vol. 2(1), 2013, pp. 44-54, doi: 10.5430/air.v2n1p44

[8] R. Gunawan, A. Rahmatulloh, I. Darmawan, & F. Firdaus, Comparison of web scraping techniques: regular expression, HTML DOM and Xpath. In 2018 International Conference on Industrial Enterprise and System Engineering (ICoIESE 2018), Atlantis Press, 2019, pp. 283-287.

[9] C. Ariza, D. Conti, & W. Garzon, Automated Benchmarking for the Purchase of Second-hand Cars and Motorcycles in Colombia through Web Scraping and Prescriptive Analytics, 2023, doi: 10.21203/rs.3.rs-3693600/v1

[10] L. Linzo, M. Pardo & L. Rodríguez., Web classification wrappers. Tesis de grado. Universidad de la República (Uruguay), Facultad de Ingeniería, 2002, pp. 1-8.

[11] J. Clark & S. Derose, XML path language (XPath), 1999. [Online]. Disponible en: https://www.w3.org/TR/1999/REC-xpath-19991116

[12] L. Richardson, Beautiful soup documentation, 2019. [Online] Disponible en: https://media.readthedocs.org/pdf/beautiful-soup-4/latest/beautiful-soup-4.pdf

[13] I. Almaqbali, F. Al Khufairi, M. Khan, A. Bhat, & I. Ahmed, Web scrapping: Data extraction from websites. Journal of Student Research, 2019.

[14] Z. Liao, H. Ding, Y. Xia, & F. Ma, Web Crawler Based Study on Traffic Accident Data Acquisition for Operating Tunnels. In IOP Conference Series: Materials Science and Engineering, vol. 741(1), 2020, p. 12070, doi: 10.1088/1757-899X/741/1/012070

[15] S. Thivaharan, G. Srivatsun, & S. Sarathambekai, A survey on python libraries used for social media content scraping. In 2020 International Conference on Smart Electronics and Communication (ICOSEC), 2020, pp. 361-366, doi: 10.1109/ICOSEC49089.2020.9215357

[16] X. Yupen, Y. Jindong, Y. Linyi, H. Kaijie H., Y. Xiaoyuan, W. Cunxiang, W. Yidong, Y. Yue, S. Philip, Q. Yu, & X. Xing, A Survey on Evaluation of Large Language Models. ACM Trans. Intell. Syst. Technol, 2024, doi: 10.1145/3641289

[17] M. Fezari, A. Al-Dahoud & A. Al-Dahoud, Augmanting Reality: The Power of Generative AI. University Badji Mokhtar Annaba: Annaba, Algeria, 2023.

[18] G. Remón, Resumen de textos literarios en español por medio de modelos lingüísticos Transformers (Disertación Doctoral, ETSI_Informatica), 2020, pp. 3-25.

[19] Z. Chi, L. Dong, S. Ma, S. Mao, H. Huang, & F. Wei, mT6: Multilingual pretrained text-to-text transformer with translation pairs, 2021, doi: 10.48550/arXiv.2104.08692

[20] V. Câmara, R. Mendonca, A. Silva, & L. Cordovil, A Large Language Model approach to SQL-to-Text Generation. In 2024 IEEE International Conference on Consumer Electronics (ICCE), 2024, pp. 1-4, doi: 10.48550/arXiv.2401.14887

[21] S. Ravichandiran, Getting Started with Google BERT: Build and train state-of-the-art natural language processing models using BERT. Packt Publishing Ltd, 2021.

[22] B. Sigüenza, Transformers BERT for Question-Answering COVID-19, Universidad Nacional de Educación a Distancia, 2021. [Online]. Disponible en: http://e-spacio.uned.es/fez/eserv/bibliuned:master-ETSInformatica-TL-Bsiguenza/SiguenzaBernardo_TFM.pdf

[23] C. Ramírez, Programación de Inteligencia Artificial. Curso Práctico, Ra-Ma, vol. 1, 2023.

[24] B. Wang, & C. Kuo, Sbert-wk: A sentence embedding method by dissecting bert-based word models. IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, 2020, pp. 2146-2157, doi: 10.1109/TASLP.2020.3008390

[25] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen & V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019, doi: 10.48550/arXiv.1907.11692

[26] V. Sanh, L. Debut, J. Chaumond, & T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, 2019, doi: 10.48550/arXiv.1910.01108

[27] S. Lee, H. Lin, J. Park, E. Lim, & J. Woo, NLP Models Classifying Helpful Ratings in OpenTable Dataset, 2023.

[28] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, & R. Soricut, Albert: A lite bert for self-supervised learning of language representations, 2019, doi: 10.48550/arXiv.1909.11942

[29] H. Ghavidel, A. Zouaq, & M. Desmarais, Using BERT and XLNET for the Automatic Short Answer Grading Task. In CSEDU (1), 2020, pp. 58-67.

[30] N. Reimers, & I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084, 2019, doi: 10.48550/arXiv.1908.10084

[31] K. Song, X. Tan, T. Qin, J. Lu, & T. Liu, Mpnet: Masked and permuted pre-training for language understanding. Advances in neural information processing systems, vol. 33, 2020, pp. 16857-16867, doi: 10.48550/arXiv.2004.09297

[32] V. Nasteski, An overview of the supervised machine learning methods. Horizons. b, vol. 4(51-62), 2017, p. 56, doi: 10.20544/HORIZONS.B.04.1.17.P05

[33] X. Yang, Artificial neural networks. In Handbook of research on geoinformatics, 2009, pp. 122-128, IGI Global, doi: 10.4018/978-1-59140-995-3.ch016

[34] A. Rasamoelina, F. Adjailia & P. Sinčák, A Review of Activation Function for Artificial Neural Network, IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI), Herlany, Slovakia, 2020, pp. 281-286, doi: 10.1109/SAMI48414.2020.9108717

[35] M. Lee, GELU activation function in deep learning: a comprehensive mathematical analysis and performance, 2023, doi: 10.48550/arXiv.2305.12073

[36] M. Ureña & R. Martínez & R. González & M. Marchamalo, Automatic Building Height Estimation: Machine Learning Models for Urban Image Analysis. Applied Sciences, vol. 13, 2023, p. 5037, doi: 10.3390/app13085037

[37] H. Abu, A. Hassanat, O. Lasassmeh, A. Tarawneh, M. Alhasanat, H. Eyal & V. Prasath, Effects of distance measure choice on k-nearest neighbor classifier performance: a review. Big data, vol. 7(4), 2019, pp. 221-248, doi: 10.1089/big.2018.0175

[38] F. Ertuğrul & E. Mehmet, A novel version of k nearest neighbor: Dependent nearest neighbor, Applied Soft Computing, vol. 55, 2017, pp. 480-490, doi: 10.1016/j.asoc.2017.02.020

[39] V. Kecman, Support vector machines–an introduction. In Support vector machines: theory and applications, Berlin, 2005, pp. 1-47, doi: 10.1007/10984697_1

[40] N. Cristianini & B. Scholkopf, Support Vector Machines and Kernel Methods: The New Generation of Learning Machines. AI Magazine, vol. 23(3), 2002, p. 31, doi: 10.1609/aimag.v23i3.1655

[41] O. Spositto, G. Blanco, & L. Matteo, Técnicas de preprocesamiento de datos en modelos no supervisados aplicados al estudio genético de la raza Aberdeen Angus, Universidad Nacional de La Matanza. Departamento de Ingeniería e Investigaciones Tecnológicas, 2022, p. 7. [Online]. Disponible en: http://repositoriocyt.unlam.edu.ar/handle/123456789/1210

[42] R. Kuc & M. Rogozinski, Elasticsearch server. Packt Publishing Ltd., 2013.

[43] A. Andhavarapu, Learning Elasticsearch. Packt Publishing Ltd., 2017.

[44] M. Konda, Elasticsearch in Action, Second Edition. Estados Unidos: Manning, 2 ed., 2024, p. 23.

[45] Y. Pant, B. Joshi, & N. Adhikari, Performance Analysis of Shard Selection Techniques on Elasticsearch, 2021.

[46] M. Divya & S. Goyal, An advanced and quick search technique to handle voluminous data. Compusoft, 2 ed., 2013, pp. 171-175.

[47] Documentación Elasticsearch, 2024. [Online]. Disponible en: https://www.elastic.co/elasticsearch/machine-learning

[48] L. Kurisinkel & N. Chen, LLM Based Multi-Document Summarization Exploiting Main-Event Biased Monotone Submodular Content Extraction, Cornell University, 2023, pp. 1-7.

[49] B. Fatemi, F. Rabbi & A. Opdahl, Evaluating the Effectiveness of GPT Large Language Model for News Classification in the IPTC News Ontology, IEEE Access, 2023, pp. 145386-145394.

[50] N. Nakshatri, S. Liu, S. Chen & D. Roth, Using LLM for Improving Key Event Discovery: Temporal-Guided News Stream Clustering with Event Summaries, Association for Computational Linguistics, 2023, pp. 4162-4173.

[51] X. Zhou, Y. Zhang, L. Cui, & D. Huang, Evaluating Commonsense in Pre-Trained Language Models. Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34(5), 2020, pp. 9733-9740, doi: 10.1609/aaai.v34i05.6523

[52] R. Wu, RecBERT: Semantic recommendation engine with large language model enhanced query segmentation for k-nearest neighbors ranking retrieval. Intelligent and Converged Networks, 2024.

[53] A. Jääskeläinen, E. Taimela, & T. Heiskanen, Predicting the success of news: Using an ML-based language model in predicting the performance of news articles before publishing. In Proceedings of the 23rd International Conference on Academic Mindtrek, 2020, pp. 27-36, doi: 10.1145/3377290.3377299

[54] M. Tsagkias, W. Weerkamp, & M. De Rijke, Predicting the volume of comments on online news stories. In Proceedings of the 18th ACM conference on Information and knowledge management, 2009, pp. 1765-1768, doi: 10.1145/1645953.1646225

[55] G. Szabo & B. Huberman, Predicting the popularity of online content. Communications of the ACM, vol. 53(8), 2010, pp. 80-88, doi: 10.2139/ssrn.1295610

[56] R. Mccreadie, C. Macdonald & I. Ounis, A learned approach for ranking news in real-time using the blogosphere. In String Processing and Information Retrieval: 18th International Symposium, SPIRE 2011, Pisa, Italy Proceedings, vol. 18, 2011, pp. 104-116, doi: 10.1007/978-3-642-24583-1_11

[57] D. Gruhl, R. Guha, R. Kumar, J. Novak, & A. Tomkins, The predictive power of online chatter. In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, 2005, pp. 78-87, doi: 10.1145/1081870.1081883

[58] Q. Al-Radaideh, A. Assaf & E. Alnagi, Predicting stock prices using data mining techniques. In The International Arab Conference on Information Technology, ACIT'2013, 2013, pp. 1-8.

[59] C. Schröer, F. Kruse & J. Gómez, A Systematic Literature Review on Applying CRISP-DM Process Model, Procedia Computer Science, vol. 181, 2021, pp. 526-534, doi: 10.1016/j.procs.2021.01.199

[60] J. Espinosa, Aplicación de metodología CRISP-DM para segmentación geográfica de una base de datos pública. Ingeniería, investigación y tecnología, vol. 21(1), 2020, doi: 10.22201/fi.25940732e.2020.21n1.008

[61] M. Shayboun, C. Koch, & D. Kifokeris, Learning from Accidents: Machine Learning Prototype Development Based on the CRISP-DM Business Understanding, Proceedings of the Joint CIB W099 & W123 Annual International Conference 2021: Good health, Changes & innovations for improved wellbeing in construction, 2021.

[62] E. Turban, R. Sharda & D. Delen, Business intelligence: A managerial approach, Pearson Education, 2014, p. 233.

[63] U. Shafique, & H. Qaiser, A comparative study of data mining process models (KDD, CRISP-DM and SEMMA). International Journal of Innovation and Scientific Research, vol. 12(1), 2014, pp. 217-222.

[64] R. Nils, Modelos Pre-entrenados SBERT, 2024. [Online]. Disponible en: https://www.sbert.net/docs/pretrained_models.html

[65] Datamarket, Noticias económcias, 2024. [Online]. Disponible en: https://www.kaggle.com/datasets/datamarket/noticias-economicas

[66] Enzipe Software House, 2024, Disponible en: https://www.prepostseo.com/domain-authority-checker

[67] D. Chen, Pandas for everyone: Python data analysis. Addison-Wesley Professional, 2017.

[68] O. Rolik, K. Ulianytska, M. Khmeliuk, V. Khmeliuk, & U. Kolomiiets, Increase efficiency of relational databases using instruments of second normal form. In 2021 IEEE

3rd International Conference on Advanced Trends in Information Theory (ATIT), 2021, pp. 221-225, doi: 10.1109/ATIT54053.2021.9678605

[69] J. Sánchez, Títulos y titulares. Sobre las funciones de la titulación periodística. Communication & Society, vol. 3(1-2), 1990, pp. 173-183, doi: 10.15581/003.3.35523

[70] B. Dixit, Mastering Elasticsearch 5. x. Packt Publishing Ltd., 2017.

[71] M. Murata, H. Nagano, R. Mukai, K. Kashino, & S. Satoh, BM25 with exponential IDF for instance search. IEEE Transactions on Multimedia, vol. 16(6), 2014, pp. 1690-1699, doi: 10.1109/TMM.2014.2323945

[72] D. Jiménez & E. Román, Clasificación automática de fragmentos de vasijas arqueológicas mediante el modelo Bolsa de Palabras, Arqueología Computacional. Nuevos enfoques para la documentación, análisis y difusión del patrimonio cultural, México: Instituto Nacional de Antropología e Historia, 2017, pp. 111-126.

[73] A. Escudero, Optimización de la web de Leigh Perú como estrategia para dinamizarla y volverla atractiva en los motores de búsqueda, Universidad de Piura, 2020, p. 24. [Online]. Disponible en: https://pirhua.udep.edu.pe/bitstreams/9dc8e2d1-402f-4c9b-b74d-a9f2ada75520/download

[74] J. Miyahira, Calidad en los servicios de salud: ¿Es posible? Revista Médica Herediana, vol. 12(3), 2001, pp. 75-77. [Online]. Disponible en: http://www.scielo.org.pe/scielo.php?script=sci_arttext&pid=S1018-130X2001000300001&lng=es&tlng=es

[75] G. Kumar, N. Duhan & A. K. Sharma, Page ranking based on number of visits of links of Web page, 2nd International Conference on Computer and Communication Technology (ICCCT-2011), Allahabad, India, 2011, pp. 11-14, doi: 10.1109/ICCCT.2011.6075206

[76] Market Brew, 2024, https://marketbrew.ai/a/page-authority-seo

[77] S. Kumar, Can Webometrics Predict the Academic Rankings of Institutes?. The Journal of Prediction Markets, vol. 14(2), 2020, pp. 61-76, doi: https://doi.org/10.5750/jpm.v14i2.1816

[78] D. Grattarola & C. Alippi, Graph neural networks in tensorflow and keras with spektral [application notes], IEEE Computational Intelligence Magazine, vol. 16(1), 2021, pp. 99-106.

[79] N. Huda & M. Mubarok, A multi-label classification on topics of quranic verses (english translation) using backpropagation neural network with stochastic gradient descent and adam optimizer. In 2019 7th International conference on information and communication technology (ICoICT), 2019, pp. 1-5. doi: 10.1109/ICoICT.2019.8835362

[80] S. Aziz, Improving the KNN algorithm by using weighted euclidean distance, Journal of Software Engineering & Intelligent Systems, vol. 6(1), 2021, pp. 47-52. [Online]. Disponible en: https://www.academia.edu/download/68370304/V6N1_5.pdf

[81] D. Ratnasari, Comparison of Performance of Four Distance Metric Algorithms in K-Nearest Neighbor Method on Diabetes Patient Data. Indonesian Journal of Data and Science, vol. 4(2), 2023, pp. 97-108.

[82] P. Mulak & N. Talhar, Analysis of distance measures using k-nearest neighbor algorithm on kdd dataset. Int. J. Sci. Res, vol. 4(7), 2015, pp. 2319-7064.

[83] R. Ehsani & F. Drablos, Robust distance measures for knn classification of cancer data, Cancer informatics, vol. 19, 2020, doi: 10.1177/1176935120965542

[84] S. Aguilar, Igualación de canal no lineal mediante algoritmos kernelizados (Master's thesis), Universidad Carlos III de Madrid, 2009. [Online]. Disponible en: http://hdl.handle.net/10016/5943

6. ANEXOS

Los anexos se incluyen en el DVD que acompaña este documento

ANEXO A: Estructura de base de datos: Almacena el código SQL necesario para la construcción del esquema y procedimientos almacenados de la base de datos.

ANEXO A.1 ESQUEMA_BDD: Alacena el esquema de la base de datos.

ANEXO A.2 CALCULAR_PARAMETROS_NUMERICOS: Código SQL para crear procedimiento almacenado.

ANEXO A.3 LIMPIAR_LINEAS: Código SQL para crear procedimiento almacenado.

ANEXO A.4 CALCULAR_PUNTAJE_FINAL_GRUPO: Código SQL para crear procedimiento almacenado.

ANEXO A.5 REMOVER_SW_ENTRADA: Código SQL para crear procedimiento almacenado.

ANEXO A.6 REMOVER_SW_LINEAS_NOTICIAS_CODIFICADO: Código SQL para crear procedimiento almacenado.

ANEXO A.7 REMOVER_SW_NOTICIAS_CODIFICADO: Código SQL para crear procedimiento almacenado.

ANEXO B: Contenido de la base de datos relacional.

ANEXO B.1 cod_autor.csv: Contenido de la tabla COD_AUTOR.

ANEXO B.2 cod_categoria.csv: Contenido de la tabla COD_CATEGORIA.

ANEXO B.3 cod_diario.csv: Contenido de la tabla COD_DIARIO.

ANEXO B.4 noticias.csv: Contenido de la tabla NOTICIAS.

ANEXO B.5 noticias_codificado.csv: Contenido de la tabla NOTICIAS_CODIFICADO.

ANEXO B.6 lineas_noticias.csv: Contenido de la tabla LINEAS_NOTICIAS.

ANEXO B.7 lineas_noticias_codificado.csv: Contenido de la tabla LINEAS_NOTICIAS_CODIFICADO.

ANEXO C: Código fuente de los módulos del sistema.

ANEXO C.1 WEB SCRAPING.ipynb: Cuaderno Jupyter para realizar web scaping de las noticias digitales.

ANEXO C.2 LIMPIEZA: 2 cuadernos Jupyter de preparación y limpieza de datos.

ANEXO C.2.1 LIMPIEZA DE DATOS NOTICIAS.ipynb: Cuaderno Jupyter para limpieza de la información de la noticia sin contar con el contenido.

ANEXO C.2.2 LIMPIEZA DE DATOS LINEAS NOTICIAS.ipynb: Cuaderno Jupyter de limpieza del contenido de las noticias digitales.

ANEXO C.3 INGESTA A INSTANCIA ELASTICSEARCH: 2 cuadernos Jupyter para trasladar la información clave de la base de datos relacional a la instancia Elasticsearch.

ANEXO C.3.1 INGESTA ELASTIC SEARCH ENCABEZADOS.ipynb: Cuaderno Jupyter para trasladar la información de los titulares.

ANEXO C.3.2 INGESTA ELASTIC SEARCH LINEAS.ipynb: Cuaderno Jupyter para trasladar la información referente a las líneas de contenido.

ANEXO C.4 AGRUPACION NOTICIAS.ipynb: Cuaderno Jupyter para clusterizar los diarios digitales.

ANEXO C.5 MODELO SUPERVISADO PREDECIR RANKING.ipynb: Cuaderno Jupyter para entrenar y evaluar los modelos de predicción de Page Authority.

ANEXO C.6 PUNTAJE LINEAS.ipynb: Cuaderno Jupyter para generar los puntajes de contenido.

ANEXO C.7 escalador.pck: Almacena el escalador utilizado durante el entrenamiento de los modelos de predicción de Page Authority.

ANEXO C.8 modelo_predictivo.keras: Almacena el modelo ANN entrenado.

ANEXO D: Código fuente de interfaz gráfica de usuario.

ANEXO D.1 gui.py: Pantalla principal de la interfaz gráfica de usuario para consulta de titulares.

ANEXO D.2 pages/tab.py: Pantalla de muestra de información de contenido clusterizado.

ANEXO D.3 pages/calificar.py: Pantalla para realizar una puntuación de usuario.

ANEXO D.4 pages/agradecimiento.py: Pantalla de agradecimiento.

ANEXO D.5 pages/ConfigFile.properties: Archivo con los parámetros de conexión a la base de datos relacional y al servidor Elasticsearch.