Detección Temprana de Anorexia y Bulimia en Redes Sociales

Máster en Big Data

Trabajo Fin de Máster

Autor:
Alicia Ruiz Simón
Tutor/es:
Jose Antonio García Díaz
Rafael Valencia García

2 de Junio de 2025

Facultad
Informática
Universidad
Murcia

Detección Temprana de Anorexia y Bulimia en Redes Sociales

Autor
Alicia Ruiz Simón

Tutor/es
Jose Antonio García Díaz
Departamento de Informática y Sistemas
Rafael Valencia García
Departamento de Informática y Sistemas

Facultad de Informática

Máster en Big Data

DIS Departamento de Informática y Sistemas

UNIVERSIDAD DE MURCIA

Murcia, 2 de Junio de 2025

Agradecimientos

Quiero expresar mi agradecimiento a mis compañeros de máster. Compartir esta experiencia fue un privilegio; su apoyo, colaboración y las innumerables horas de estudio hicieron el camino mucho más llevadero y enriquecedor. A mis tutores, gracias por su invaluable guía y paciencia. Su conocimiento y dedicación fueron fundamentales para el desarrollo de este trabajo.

A mi familia, gracias por su amor incondicional y su ánimo constante en cada etapa de este proceso. Y, por último, a mi pareja, gracias por tu infinita comprensión, tu paciencia y por ser mi mayor apoyo y motivación en todo momento. Sin todos, este logro no hubiera sido posible.

Declaración firmada sobre originalidad del trabajo

D./Dña. **Alicia Ruiz Simón**, con DNI **48736048Z**, estudiante de la titulación de **Máster en Big Data** de la Universidad de Murcia y autor del TF titulado "**Detección Temprana de Anorexia y Bulimia en Redes Sociales**".

De acuerdo con el Reglamento por el que se regulan los Trabajos Fin de Grado y de Fin de Máster en la Universidad de Murcia (aprobado C. de Gob. 30-04-2015, modificado 22-04-2016 y 28-09-2018), así como la normativa interna para la oferta, asignación, elaboración y defensa delos Trabajos Fin de Grado y Fin de Máster de las titulaciones impartidas en la Facultad de Informática de la Universidad de Murcia (aprobada en Junta de Facultad 27-11-2015)

Declaro:

Que el Trabajo Fin de Máster presentado para su evaluación es original y de elaboración personal. Todas las fuentes utilizadas han sido debidamente citadas. Así mismo, declara que no incumple ningún contrato de confidencialidad, ni viola ningún derecho de propiedad intelectual e industrial

Murcia, a 2 de Junio de 2025

Fdo.: Alicia Ruiz Simón
Autora del TFM

Resumen

Este trabajo aborda la detección temprana de trastornos alimenticios, específicamente anorexia y bulimia, a través del análisis de texto en español. A diferencia de estudios previos que han utilizado enfoques de clasificación binaria —en los que se determina únicamente si un usuario padece o no un trastorno—, se propone aquí una formulación del problema como una tarea de clasificación multietiqueta, permitiendo predecir el grado de la presencia de condiciones en los textos analizados. Esta elección responde a la complejidad clínica de los trastornos alimenticios, que a menudo presentan síntomas solapados, y que no pueden representarse adecuadamente mediante un enfoque binario tradicional.

El núcleo metodológico del estudio está basado en el uso de modelos de lenguaje preentrenados en español, concretamente BETO y ALBETO, aplicados bajo dos paradigmas distintos: por un lado, mediante fine-tuning completo del modelo, adaptando todos sus pesos a la tarea específica; y por otro lado, mediante el uso de embeddings. Esta doble estrategia permite evaluar las ventajas y limitaciones de cada enfoque, especialmente en contextos donde los datos disponibles presentan desbalance de clases y una cantidad limitada de ejemplos para las categorías minoritarias.

Para preservar la coherencia en la distribución de etiquetas durante el proceso de entrenamiento, validación y prueba, se emplea un esquema de división de datos mediante Multilabel Stratified Shuffle Split, garantizando así que cada subconjunto refleje adecuadamente la presencia de cada una de las etiquetas. El preprocesamiento de los textos se realiza mediante los tokenizadores correspondientes a cada modelo, ajustando el tamaño máximo de secuencia e incluyendo las etiquetas como matrices multietiqueta binarizadas. Las métricas de evaluación incluyen F1-score micro y macro, así como medidas adicionales que reflejan el rendimiento específico por etiqueta, dada la naturaleza multietiqueta y desbalanceada del problema.

Este estudio no solo contribuye al desarrollo de herramientas automáticas de detección temprana de trastornos alimenticios, sino que también pone en evidencia la importancia de adoptar enfoques más flexibles y representativos del espectro clínico real. Al permitir una clasificación multietiqueta, se facilita una comprensión más precisa de los mensajes de los usuarios, abriendo la puerta a futuras aplicaciones en el ámbito de la salud mental digital y la intervención psicosocial asistida por inteligencia artificial.

Extended Abstract

The increasing prevalence of eating disorders such as anorexia nervosa and bulimia nervosa represents a significant global public health concern, especially among adolescents and young adults. These disorders are complex psychiatric conditions characterized by disordered eating behaviors, intense fears of weight gain, and distorted body image perceptions. Over the past decade, the emergence of social media platforms has profoundly transformed the way individuals communicate, express emotional distress, and seek or offer support. In this context, platforms such as Telegram have gained relevance not only as spaces for open dialogue but also as digital environments where harmful ideologies, including the normalization or promotion of disordered eating behaviors, may proliferate. This dual nature of online spaces—offering both support and risk—motivates the exploration of new technological approaches for the early detection of such mental health risks.

This work is framed within the field of digital mental health and seeks to develop an automated system for the early detection of anorexia and bulimia through the analysis of textual content generated by users on social media, specifically on Telegram. The primary focus is on analyzing Spanish-language content, addressing a gap in the current research, which has predominantly focused on English-language data. Unlike most previous approaches that rely on binary classification models—distinguishing only between the presence or absence of a disorder—this study proposes a multilabel classification approach that enables the identification of nuanced and overlapping patterns of disordered behavior. This formulation is more clinically appropriate, as eating disorders often present a spectrum of symptoms that may co-exist or evolve over time, making binary categorizations insufficient for capturing the complexity of these conditions.

The research uses the MentalRiskES dataset, a Spanish-language corpus derived from real conversations in public Telegram groups that address various mental health topics. This corpus contains anonymized and ethically collected user data, annotated by multiple human raters to indicate the potential presence and type of eating disorders. Each user is represented through their messages, grouped into a single textual input, and labeled with one or more of the following categories: users suffering from an eating disorder (bs), users who suffer and promote harmful behaviors (bsf), users who suffer but oppose the disorder (bsa), users who suffer and do not clearly fall into the prior

categories (bso), and users from the control group (bc), which includes unaffected individuals and professionals. The corpus provides a rich, though imbalanced, dataset for evaluating various classification models and techniques.

A distinctive characteristic of this study lies in its emphasis on multilabel annotation and prediction, which marks a departure from the prevailing binary paradigms in computational mental health. By allowing a single user to be associated with multiple disorder-related labels, the system reflects the often overlapping and non-exclusive nature of eating disorder behaviors. This design acknowledges that an individual may simultaneously exhibit contradictory behaviors—such as promoting disordered eating in one context while expressing regret or seeking help in another. Such psychological complexity cannot be captured by traditional binary classifiers and thus requires models that are capable of understanding multifaceted linguistic cues. This multilabel perspective not only enhances the granularity of detection but also aligns the computational framework more closely with the clinical understanding of eating disorders as dynamic, evolving conditions.

To address this task, several methodological steps were undertaken. Initially, traditional machine learning approaches were applied using Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) vectorizations, paired with classifiers such as Naïve Bayes and Support Vector Machines (SVM). These models served as a baseline and achieved acceptable performance on the most frequent labels. However, they consistently failed to detect rare classes, largely due to the severe imbalance in class distribution. For instance, while the control category (bc) and primary disorder category (bs) had ample examples, categories such as bsa and bso were extremely underrepresented, comprising less than two percent of the total labeled data.

To overcome the limitations of traditional models, more advanced techniques were implemented, leveraging the capabilities of deep learning and contextualized language representations. Specifically, the study evaluated the performance of embeddings derived from pretrained transformer models, such as BETO—a BERT-based model trained on Spanish corpora—and MarIA, a model trained on a large and diverse collection of Spanish biomedical, legal, and general texts. These embeddings, extracted without altering the internal weights of the models, were used as input to external multilabel classifiers, typically based on logistic regression. This approach improved performance over the BoW models, particularly in capturing the semantics and emotional content of the messages. BETO consistently outperformed MarIA, suggesting that domain-specific training (in this case, colloquial or general Spanish rather than formal domains) plays a crucial role in tasks related to mental health detection.

The most significant performance gains, however, were observed through the application of fine-tuning techniques. Full model fine-tuning was conducted using BETO and ALBETO, the latter being a computationally lighter variant designed to offer effi-

ciency with minimal performance trade-offs. These models were trained end-to-end on the labeled dataset, with their weights adjusted to optimize for the classification task. To preserve the distribution of labels across training, validation, and test sets, a stratified multilabel splitting strategy was used. This methodological choice was essential for ensuring that each subset of the data maintained a representative distribution of the multilabel annotations, thus enabling fairer and more reliable evaluations.

Fine-tuned BETO demonstrated the highest overall performance, achieving a micro-averaged F1-score of approximately 0.95 and a macro-averaged F1-score above 0.83 when focusing on the three main labels—bs, bsf, and bc—and excluding the sparsely represented bsa and bso labels. ALBETO also performed strongly, offering slightly lower but still competitive results, especially when computational resources were limited. These outcomes highlight the effectiveness of transformer-based language models, particularly when adapted through fine-tuning, in detecting subtle linguistic markers of psychological distress in real-world textual data.

An important finding of the study was that reducing the number of target classes—specifically, by excluding the extremely rare labels—significantly improved model stability and performance. This pragmatic decision, while reducing the clinical granularity of the predictions, enhanced the models' reliability and robustness, enabling more consistent detection of users at risk. However, this also underscores a key challenge in working with real-world, unbalanced clinical data: achieving a balance between sensitivity to rare conditions and the need for accurate, generalizable models.

The linguistic characteristics of the texts analyzed in this study align with previous findings in computational mental health research. Users expressing symptoms of eating disorders frequently employ first-person pronouns, emotionally charged vocabulary, and references to body image, food, and control. These patterns were particularly pronounced in the bs and bsf categories, reflecting the internal struggle, emotional distress, and sometimes normalization of harmful behaviors. Transformer models, with their ability to capture contextual and emotional nuances, proved particularly adept at modeling these patterns, in contrast to traditional vectorization techniques that fail to consider semantic relationships between words.

This research contributes not only to the technical field of natural language processing but also to the broader domain of public mental health. By demonstrating the feasibility and effectiveness of automated, language-based detection tools, the study opens the door to future applications in clinical and preventive contexts. Such tools could support healthcare professionals by providing early warnings, facilitating triage, and enabling targeted interventions, especially in resource-constrained environments where mental health services are limited or stigmatized.

Nevertheless, several ethical considerations must be addressed before implementing such systems in real-world scenarios. Automated analysis of user-generated content

raises questions related to privacy, consent, data ownership, and the potential for false positives or negatives. These concerns are particularly acute in mental health contexts, where the consequences of misclassification can be profound. The MentalRiskES dataset, fortunately, was constructed following rigorous anonymization and ethical review protocols. However, any deployment of models derived from such data must be accompanied by transparent, ethical frameworks that prioritize user safety, autonomy, and the principle of "do no harm."

There are also technical limitations to consider. The dataset, although rich, is relatively small for deep learning standards and exhibits strong class imbalance. The messages originate from a single platform and language, which limits generalizability to other contexts or populations. Moreover, the study does not incorporate temporal or behavioral features, such as the frequency or timing of message posting, which could further enhance predictive performance. Future research may benefit from integrating these aspects, as well as exploring multimodal data sources—including images and videos—that are common on platforms like Instagram or TikTok and highly relevant to body image-related content.

Another promising direction for future work involves the use of explainable AI techniques to shed light on the decision-making processes of the models. Understanding which features or expressions contribute most to a classification decision could help build trust among clinicians and users alike and potentially reveal new linguistic markers of distress. Moreover, incorporating feedback from mental health professionals into the annotation and evaluation process could improve the clinical validity of the models and align them more closely with diagnostic criteria and therapeutic practices.

In conclusion, this thesis illustrates the potential of modern NLP techniques, particularly transformer-based language models fine-tuned on domain-specific data, to address complex and sensitive problems such as the early detection of eating disorders through user-generated texts. It bridges the fields of computational linguistics, clinical psychology, and data science to provide a proof of concept for innovative, language-driven mental health technologies. While challenges remain in terms of data quality, ethical considerations, and clinical integration, the findings offer a compelling case for the continued exploration of AI-driven tools to support mental health monitoring, prevention, and intervention in the digital age. By harnessing the expressive power of language and the analytical strength of machine learning, such tools can contribute meaningfully to the global effort to promote psychological well-being and respond more swiftly and effectively to the early signs of mental illness.

Índice general

Índice de Códigos

1 Introducción

Los trastornos de la conducta alimentaria (TCA), como la anorexia nerviosa y la bulimia nerviosa, representan un grave problema de salud pública a nivel mundial, con consecuencias físicas, psicológicas y sociales que afectan de manera profunda a quienes los padecen. Según la Organización Mundial de la Salud (Organización Mundial de la Salud, 2019), los TCA son enfermedades mentales complejas que se manifiestan a través de patrones anómalos de alimentación y una preocupación patológica por el peso y la imagen corporal. La anorexia nerviosa, caracterizada por la restricción extrema de la ingesta calórica y un temor intenso a ganar peso, y la bulimia nerviosa, definida por episodios recurrentes de ingesta excesiva seguidos de conductas compensatorias como el vómito autoinducido, afectan mayoritariamente a adolescentes y adultos jóvenes, especialmente mujeres.

En los últimos años, el auge de las plataformas digitales y redes sociales ha transformado radicalmente la manera en que los individuos se comunican, expresan y buscan apoyo emocional. En este contexto, las redes sociales han adquirido un papel ambivalente: por un lado, han facilitado el acceso a información y comunidades de apoyo, pero por otro, han propiciado entornos donde se normalizan o incluso promueven conductas perjudiciales relacionadas con los TCA. Estudios recientes han documentado cómo los usuarios, especialmente adolescentes, utilizan plataformas como Instagram, TikTok o Telegram para compartir contenido relacionado con dietas extremas, control de peso y conductas alimentarias nocivas (Villar del Saz Bedmar and Baile Ayensa, 2023; Arcelus et al., 2019). Este fenómeno, conocido como ”pro-ana” o ”pro-mía”, representa una amenaza significativa para la salud mental de jóvenes en situación de vulnerabilidad.

Telegram, en particular, destaca como un canal relevante para el análisis del lenguaje y la detección de señales tempranas de trastornos mentales. Su estructura descentralizada, la posibilidad de mantener el anonimato y la proliferación de canales temáticos han convertido a esta aplicación en un espacio donde los usuarios pueden expresar libremente sus pensamientos, emociones y preocupaciones, incluyendo aquellos relacionados con la salud mental. Esto ofrece una oportunidad única para el análisis computacional de textos con fines clínicos y preventivos, especialmente en el ámbito de la salud pública digital (Cisternas-Osorio et al., 2022).

En este marco, surge el interés por aplicar tecnologías del procesamiento del lenguaje natural (PLN) y herramientas de inteligencia artificial para el análisis auto-

matizado de grandes volúmenes de datos textuales generados por los usuarios. Este enfoque se alinea con las tendencias actuales en salud digital y medicina preventiva, que promueven el uso de métodos no invasivos y pasivos para la monitorización del estado emocional y cognitivo de los individuos (Calvo et al., 2017; Chancellor and De Choudhury, 2019). Diversos estudios han demostrado que ciertos patrones lingüísticos, como el uso excesivo de pronombres en primera persona, términos asociados a la negatividad emocional o referencias al cuerpo y la comida, pueden constituir marcadores predictivos de la presencia de TCA y otros trastornos mentales (Birnbaum et al., 2020).

El dataset MentalRiskES (Mármol-Romero et al., 2023) representa un recurso valioso para la investigación en este campo, al proporcionar datos anonimizados obtenidos de interacciones reales en plataformas de mensajería. MentalRiskES se enfoca en la detección automática de riesgos de salud mental a partir de datos lingüísticos generados en español, lo cual resulta especialmente relevante dado que la mayoría de las investigaciones y herramientas existentes se han desarrollado en inglés. Esta base de datos ha sido construida con criterios éticos rigurosos y contiene anotaciones especializadas que permiten el entrenamiento de modelos de clasificación orientados a la identificación de conductas autolesivas, ideación suicida, depresión, ansiedad y, en el caso que nos ocupa, trastornos alimentarios.

El análisis textual de mensajes escritos por usuarios en plataformas como Telegram no solo permite detectar indicios de TCA, sino también comprender mejor los contextos y dinámicas que subyacen a estas patologías. La investigación en este ámbito puede contribuir al desarrollo de sistemas de alerta temprana que permitan intervenir antes de que los síntomas se agraven, favoreciendo así una atención más oportuna y personalizada. Además, este enfoque puede facilitar la identificación de factores de riesgo emergentes y tendencias socioculturales que inciden en la aparición de los TCA, como la presión estética, la idealización de la delgadez o la exposición a discursos tóxicos en línea (Ruiz-Centeno et al., 2025).

La combinación del análisis lingüístico automatizado y el estudio de datos extraídos de redes sociales como Telegram abre nuevas posibilidades para la prevención y detección temprana de los trastornos de la conducta alimentaria. En este sentido, el presente trabajo se inscribe dentro de un campo interdisciplinario que integra conocimientos de la psicología clínica, la lingüística computacional y la ciencia de datos, con el objetivo de contribuir a la mejora del bienestar psicológico de los individuos mediante herramientas tecnológicas innovadoras.

A pesar del avance en los tratamientos y estrategias de prevención, uno de los principales desafíos en la lucha contra los trastornos alimentarios sigue siendo la detección temprana. Numerosos estudios han demostrado que cuanto antes se identifica un TCA, mayor es la probabilidad de una intervención efectiva y menor el riesgo de

desarrollar complicaciones crónicas o recurrencias (Treasure et al., 2020). Sin embargo, los síntomas iniciales suelen pasar desapercibidos tanto para los entornos familiares como para los profesionales de salud, en parte debido al estigma social que rodea los trastornos mentales y a la tendencia de los afectados a ocultar o minimizar sus síntomas. En este contexto, el análisis de textos escritos espontáneamente por los propios usuarios en plataformas digitales se presenta como una fuente rica de información que puede revelar indicadores sutiles y persistentes de malestar psicológico.

La manera en que los individuos se expresan lingüísticamente puede ofrecer señales reveladoras sobre su estado emocional y cognitivo. En el caso de los TCA, investigaciones previas han identificado ciertos rasgos lingüísticos recurrentes entre quienes padecen estas condiciones, como el uso frecuente de términos relacionados con el cuerpo, la comida y el control, así como expresiones de autoevaluación negativa, desesperanza o perfeccionismo (Sukunesan et al., 2021). Estos patrones pueden no ser evidentes en conversaciones presenciales o entrevistas clínicas, pero sí pueden emerger en contextos digitales donde los usuarios se sienten más libres para compartir sus pensamientos sin miedo a ser juzgados. A través del uso de algoritmos de análisis semántico, modelos de aprendizaje profundo y técnicas de vectorización textual, es posible detectar y cuantificar estas señales lingüísticas para desarrollar modelos predictivos capaces de identificar perfiles de riesgo.

En este sentido, el uso de redes neuronales, clasificadores basados en transformers y modelos contextualizados del lenguaje como BERT o BETO (Alaparthi and Mishra, 2021), ha abierto nuevas oportunidades para mejorar la precisión y sensibilidad de los sistemas de detección automática de TCA. Estos modelos no solo permiten captar el significado explícito de las palabras, sino también inferencias contextuales, emociones subyacentes y relaciones sintácticas complejas, aspectos fundamentales cuando se trata de analizar textos de carácter emocional, introspectivo o subjetivo. A diferencia de los enfoques tradicionales basados únicamente en palabras clave, estas técnicas modernas permiten un análisis más fino y personalizado del lenguaje.

Telegram se convierte así en una plataforma especialmente relevante para este tipo de investigaciones. A diferencia de otras redes más visuales o abiertas como Instagram o Twitter, Telegram permite una comunicación más íntima, directa y menos expuesta públicamente. Los grupos cerrados, los canales temáticos y la posibilidad de mantener conversaciones anónimas fomentan un tipo de interacción más introspectiva y menos condicionada por la imagen pública. Esta dinámica propicia la aparición de discursos más genuinos sobre el malestar psicológico, que pueden ser de enorme utilidad para el análisis computacional del lenguaje y la detección temprana de síntomas de TCA. Además, la naturaleza asincrónica y textual de las interacciones en esta red facilita la recolección, anonimización y procesamiento de los datos, respetando siempre los principios éticos y legales que rigen la investigación en salud mental y tecnología (Cisternas-Osorio et al., 2022).

Así, la combinación de datos textuales provenientes de Telegram con métodos avanzados de procesamiento del lenguaje natural y aprendizaje automático ofrece una vía prometedora no solo para la detección temprana de anorexia y bulimia, sino también para el desarrollo de herramientas de monitoreo continuo, intervención personalizada y generación de alertas clínicas en tiempo real. Estas herramientas podrían integrarse en sistemas de salud digital, programas de prevención escolar o entornos terapéuticos, abriendo la puerta a una nueva era de medicina personalizada y preventiva en el campo de la salud mental.

1.1 Motivación

Desde la irrupción de la pandemia por COVID-19, diversos estudios han alertado sobre un preocupante incremento en la prevalencia de los trastornos de la conducta alimentaria (TCA), especialmente entre adolescentes y mujeres jóvenes en España. Según los datos recopilados en el estudio de Hurtado et al. (2024), las consultas por TCA en atención primaria han experimentado un aumento significativo desde el año 2020, duplicando los casos en algunos contextos como se observa en la figura 1.1. Este fenómeno no solo revela un cambio en los patrones de salud mental de la población joven, sino que también pone de manifiesto la urgencia de desarrollar herramientas innovadoras que permitan una detección temprana y eficaz de este tipo de trastornos.

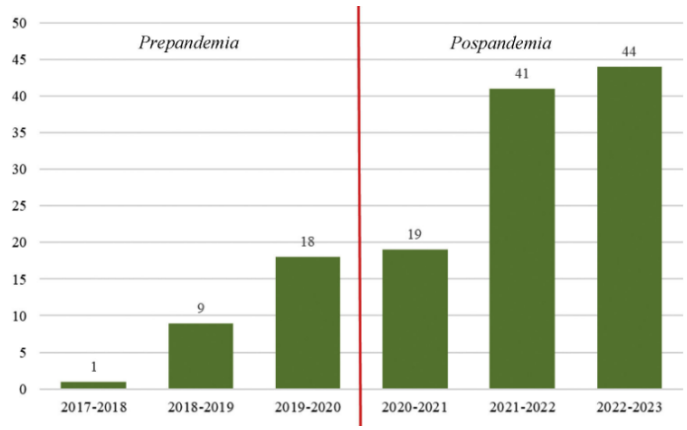

Figura 1.1: Pacientes incluidos en el estudio por periodos anuales. FEDs: Trastornos de la Alimentación y la Conducta Alimentaria. **Fuente: Adaptado de Hurtado et al. (2024).**

Asimismo, este estudio se alinea con los principios del **Objetivo de Desarrollo Sostenible (ODS) número 3** de la Agenda 2030 de las Naciones Unidas, enfocado en garantizar una vida sana y promover el bienestar para todos en todas las edades. Aunque el trabajo no plantea un sistema de monitoreo automático ni pretende sustituir

el juicio clínico, sí ofrece una contribución tecnológica que podría servir como base para futuras herramientas de apoyo a la labor de los profesionales de la salud mental. En definitiva, este proyecto aporta evidencia empírica que puede ser útil tanto en la investigación académica como en el diseño de soluciones preventivas dentro del ámbito de la salud pública digital.

1.2 Objetivos

El objetivo principal de este trabajo es desarrollar un modelo de aprendizaje automático capaz de predecir si un usuario padece anorexia, bulimia u otras condiciones intermedias, a partir del análisis automático de sus mensajes. A diferencia de estudios previos que han abordado esta problemática desde una perspectiva binaria —es decir, clasificando simplemente entre presencia o ausencia de un trastorno alimenticio— este trabajo propone un enfoque multietiqueta, que permite identificar de manera simultánea distintas condiciones asociadas. Esta aproximación reconoce la complejidad clínica de los trastornos alimentarios, los cuales pueden coexistir o presentar síntomas compartidos, lo que motiva el uso de modelos capaces de capturar esta superposición.

Además del objetivo general, este estudio persigue una serie de objetivos específicos que enriquecen el análisis y mejoran la aplicabilidad del sistema propuesto:

1. Evaluar el rendimiento de distintos enfoques de modelado textual, comparando el fine-tuning directo de modelos de lenguaje preentrenados (como BETO y ALBETO) frente al uso de embeddings fijos como entrada para clasificadores más simples.

2. Diseñar un esquema de preprocesamiento y partición de datos que conserve la distribución real de etiquetas mediante técnicas de multilabel stratification, evitando así el sesgo hacia clases mayoritarias y permitiendo una evaluación más justa del modelo.

3. Explorar el impacto del desbalance de clases en modelos multietiqueta y proponer estrategias para mitigar sus efectos, tales como el uso de funciones de pérdida ponderadas, re-muestreo o selección cuidadosa de métricas (como F1-score micro y macro) que reflejen adecuadamente el rendimiento en cada etiqueta.

4. Contribuir a la detección temprana de trastornos alimenticios en entornos digitales mediante técnicas automáticas de procesamiento del lenguaje natural (PLN), con la finalidad de facilitar el trabajo de profesionales de la salud, así como aportar evidencia empírica para futuras investigaciones en el ámbito de la salud mental computacional.

2 Estado del Arte

El análisis automático de texto ha recorrido un largo camino desde sus inicios en la lingüística computacional de mediados del siglo XX hasta los actuales sistemas basados en aprendizaje profundo. Esta evolución metodológica ha tenido un impacto significativo en múltiples disciplinas, y una de las más sensibles y prometedoras es la del análisis del lenguaje para la detección temprana de problemas de salud mental a través de redes sociales.

2.1 Orígenes del análisis textual: enfoques estadísticos y lingüísticos

Las primeras aproximaciones al análisis automático del lenguaje natural se desarrollaron en el contexto de la lingüística generativa de Chomsky (1957) en los años cincuenta, donde se postulaba una gramática universal capaz de describir las estructuras sintácticas del lenguaje. Sin embargo, la complejidad inherente al lenguaje natural pronto evidenció las limitaciones de los sistemas basados exclusivamente en reglas. Durante los años ochenta y noventa, comenzaron a ganar terreno los enfoques estadísticos, que abandonaban la gramática formal a favor de modelos probabilísticos capaces de captar regularidades en grandes corpus textuales. En esta etapa surgieron técnicas como la bolsa de palabras (Bag of Words) y los modelos de frecuencia ponderada como TF-IDF (Salton and Buckley, 1988) que permitían representar los textos como vectores en espacios multidimensionales. Aunque estas representaciones ignoraban el orden y el contexto de las palabras, fueron ampliamente utilizadas por su simplicidad y eficacia. Clasificadores como Naive Bayes o las máquinas de soporte vectorial (SVM), popularizadas en el ámbito del procesamiento de textos por Joachims (1998), fueron herramientas habituales en tareas de categorización de documentos y análisis de sentimientos.

Durante la primera década del siglo XXI, la investigación en PLN se vio enriquecida por el desarrollo de métodos que buscaban capturar el significado semántico de las palabras mediante representaciones densas en vectores de baja dimensión. Modelos como Latent Semantic Analysis (LSA) o Latent Dirichlet Allocation (LDA) permitieron representar documentos como combinaciones de tópicos latentes, marcando un avance

importante en tareas como la detección de temas o la recuperación de información. Sin embargo, el verdadero cambio de paradigma llegó con los modelos de word embeddings como Word2Vec Mikolov et al. (2013) y GloVe (Pennington et al., 2014), que introdujeron la noción de contexto semántico en la representación de palabras. Estas técnicas distribuían el significado de una palabra en un vector que codificaba su uso en múltiples contextos, lo cual resultó especialmente útil en aplicaciones como la clasificación emocional, la extracción de relaciones semánticas y la detección de similitud textual.

A partir de 2014, el análisis textual comenzó a aplicarse de forma sistemática al ámbito de la salud mental, gracias en parte al auge de las redes sociales como plataformas de expresión espontánea y emocional. En un estudio pionero, Coppersmith et al. (2014) analizaron tuits de usuarios que declaraban diagnósticos clínicos previos de depresión o trastorno de estrés postraumático (TEPT), identificando patrones lingüísticos que correlacionaban con dichos diagnósticos. Estos trabajos demostraron que era posible detectar señales tempranas de riesgo psicológico a partir de los textos compartidos en línea. Paralelamente, De Choudhury et al. (2013) profundizaron en el uso de Reddit y Twitter para predecir estados como la depresión postparto, las autolesiones o la ideación suicida. Estos estudios combinaron características lingüísticas con variables temporales y conductuales, abriendo la puerta a una nueva generación de modelos predictivos en salud mental.

2.2 Modelos de lenguaje preentrenados con transformadores

El cambio de paradigma hacia los modelos preentrenados de gran escala comenzó a consolidarse entre 2017 y 2018 con la introducción de la arquitectura Transformer, que prescinde de las estructuras recurrentes tradicionales (como las LSTM o GRU) y se basa completamente en mecanismos de atención. La propuesta original de Vaswani et al. (2017) en el artículo *Attention is All You Need* demostró que este enfoque podía alcanzar resultados de vanguardia en traducción automática sin recurrir a arquitecturas secuenciales. Su capacidad para paralelizar el entrenamiento y capturar dependencias a largo plazo en las secuencias de texto abrió una nueva vía para el modelado contextual.

Posteriormente, Radford et al. (2018) presentaron el modelo GPT (Generative Pretrained Transformer), que incorporaba una fase de preentrenamiento no supervisado seguido de una adaptación supervisada a tareas específicas, lo cual inspiró una amplia adopción del paradigma *pretrain-finetune*. Sin embargo, el modelo que consolidó este enfoque y revolucionó el campo fue BERT (Bidirectional Encoder Representations from Transformers), desarrollado por Devlin et al. (2019). BERT introdujo dos ideas clave: el preentrenamiento mediante la tarea de *Masked Language Modeling* (MLM), y el uso

de contexto bidireccional, es decir, considerar simultáneamente el contexto izquierdo y derecho de cada palabra. Estas mejoras permitieron a BERT superar el estado del arte en múltiples benchmarks de la comunidad, como GLUE o SQuAD.

La aparición del modelo BERT (Bidirectional Encoder Representations from Transformers) en 2018, desarrollado por Google, supuso una revolución en el PLN. Basado en la arquitectura de transformadores propuesta por Vaswani et al. (2017), BERT introdujo la capacidad de comprender el contexto bidireccional de las palabras, lo que le permitía superar a los modelos anteriores en prácticamente todas las tareas de PLN. A partir de este modelo se han desarrollado múltiples variantes que mejoran su eficiencia, como RoBERTa, ALBERT y DistilBERT, entre otros. En el ámbito hispanohablante, comenzaron a aparecer modelos entrenados específicamente en español, entre los cuales destacan BETO (Rosenbrock et al., 2021), ALBETO (Cañete et al., 2022) y MarIA (Gutiérrez-Fandiño et al., 2022). Estos modelos han sido entrenados con grandes corpus en español, incluyendo Wikipedia, noticias, legislación y conversaciones en foros, y han demostrado un rendimiento muy competitivo en tareas de clasificación, detección de emociones y análisis sintáctico. Gracias al fine-tuning, estos modelos pueden adaptarse a tareas específicas como la detección de señales de riesgo en salud mental con resultados muy superiores a los enfoques tradicionales.

Desde entonces, se ha producido una explosión de variantes que optimizan el rendimiento, reducen el coste computacional o se adaptan a lenguas específicas. Entre ellas destacan RoBERTa (Liu et al., 2019), que elimina algunas restricciones del entrenamiento de BERT; ALBERT (Lan et al., 2020), que reduce el número de parámetros mediante la compartición de pesos; o DistilBERT (Sanh et al., 2019), que proporciona una versión más ligera y rápida del modelo original. Además, se han lanzado proyectos a gran escala como T5 (Raffel et al., 2020) o BART (Lewis et al., 2020), que combinan objetivos de reconstrucción secuencial y han sido aplicados exitosamente en tareas generativas como el resumen automático y la traducción.

La adopción de estos modelos en el análisis de salud mental también ha sido rápida y significativa. Por ejemplo, Matero et al. (2019) y Ji et al. (2022) han aplicado variantes de BERT y RoBERTa para detectar señales de depresión, ansiedad o riesgo suicida en Reddit y Twitter, logrando mejoras notables frente a enfoques anteriores. Autores como Holgado-Apaza et al. (2023) han utilizado BETO para detectar emociones en Twitter, mientras que Martínez-Castaño et al. (2021) han desarrollado sistemas para detectar ansiedad en foros médicos mediante embeddings contextualizados. En este contexto, el corpus *MentalRiskES* (Mármol-Romero et al., 2023), desarrollado a partir de conversaciones de Telegram, representa una fuente única de datos para el estudio de riesgos psicológicos en español. Diversos trabajos han utilizado este corpus para entrenar modelos de clasificación binaria pero la mayoría de estos estudios no abordan la clasificación multiclase. Esta limitación abre la puerta a nuevas investigaciones que busquen una caracterización más fina de los riesgos.

3 Metodología

3.1 Descripción del corpus *MentalRiskES*

El presente trabajo se basa en el corpus facilitado por los organizadores de la tarea compartida MentalRiskES, una iniciativa orientada al desarrollo de tecnologías de procesamiento del lenguaje natural aplicadas a la detección de trastornos mentales en entornos digitales. Este corpus ha sido construido a partir de mensajes recopilados en grupos públicos de la plataforma de mensajería Telegram, seleccionados por su relación temática con diversos trastornos psicológicos. La lengua principal de los grupos es el español, lo cual hace de este conjunto de datos una referencia especialmente significativa para la investigación en salud mental en el ámbito hispanohablante.

Telegram es una plataforma de mensajería cifrada, basada en la nube, que permite la comunicación individual o grupal de forma gratuita (Cisternas-Osorio et al., 2022). Su uso generalizado y el acceso a grupos públicos temáticos lo convierten en una fuente de datos espontánea y representativa para el estudio de conversaciones relacionadas con la salud mental. A partir de estos grupos, se extrajeron los historiales de conversación de cientos de usuarios que posteriormente fueron sometidos a un proceso exhaustivo de anonimización, eliminando nombres, alias, números de teléfono y cualquier dato sensible. Para garantizar la calidad lingüística del corpus, se aplicaron criterios de filtrado que excluyeron mensajes con menos de tres tokens, se reemplazaron elementos como URLs o hashtags por etiquetas estándar, y se transformaron los emojis en sus equivalentes textuales.

Una vez preprocesados, los mensajes se agruparon por usuario y se establecieron límites en la cantidad de mensajes por sujeto (mínimo de 10 y máximo de 50 para el corpus de trastornos alimentarios, y hasta 100 en los demás casos), truncando los historiales cuando se superaban dichos umbrales. Cada usuario fue anotado manualmente por diez personas a través de la plataforma Prolific, utilizando la herramienta de anotación Doccano. El proceso de anotación, que se prolongó durante aproximadamente cuatro meses, permitió asignar a cada usuario una probabilidad de riesgo basada en la proporción de anotadores que consideraron que existían evidencias de un trastorno.

En total, se generaron tres subconjuntos del corpus, correspondientes a distintos trastornos: trastornos de la conducta alimentaria (TCA) o, su versión en inglés *eating*

disorders (ED), depresión y un tercer grupo de usuarios con desórdenes no especificados. Cada usuario se representa mediante un archivo en formato `JSON` que contiene la secuencia temporal de sus mensajes, junto con información de metadatos como la fecha o el identificador del mensaje. La estructura del corpus permite tanto enfoques de clasificación binaria como estrategias más avanzadas de regresión o clasificación multiclase.

Además, se incluye un archivo llamado `gold_label.csv` que relaciona cada usuario con sus etiquetas tanto de clasificación como de regresión. En este análisis se utilizan únicamente las etiquetas de clasificación, que incluyen cinco categorías: *bs* (suffer), *bsa* (suffer+against), *bsf* (suffer+in favour), *bso* (suffer+other) y *bc* (control). El sistema asigna una de estas etiquetas a cada usuario según sus mensajes. "Suffer" indica que el usuario padece una enfermedad del tipo TCA. "Suffer+against" identifica a personas que sufren el trastorno y buscan o ofrecen ayuda para superarlo, mostrando una postura en contra del trastorno. "Suffer+in favour" agrupa a quienes sufren y fomentan el trastorno. "Suffer+other" describe a quienes sufren pero no encajan en las categorías anteriores. Finalmente, "control" incluye a usuarios sin síntomas, especialistas o personas que no están relacionadas con el trastorno.

Cabe destacar que el mencionado corpus proporcionado por *MentalRiskES* incluía los datos preprocesados pero también aquellos sin preprocesar, datos crudos, que fueron los que se aplicaron a este trabajo para realizar un preprocesamiento de los datos.

3.2 Modelos empleados en *MentalRiskES*

En primer lugar, veamos cómo otros investigadores han empleado distintos modelos para este conjunto de datos. Para la resolución de la primera tarea de *MentalRiskES*, centrada en la detección de trastornos alimentarios, los participantes experimentaron con una diversidad de modelos y estrategias, aunque con un claro predominio de arquitecturas basadas en *Transformers*. La mayoría de propuestas se enmarcan en dos grandes líneas: el ajuste fino (*fine-tuning*) de modelos de lenguaje preentrenados en español, y la aplicación de técnicas tradicionales de aprendizaje automático a partir de representaciones vectoriales de los textos.

Entre los modelos más utilizados destacan variantes monolingües y multilingües entrenadas específicamente para tareas en español, como BETO, RoBERTa-BNE y MarIA, así como modelos multilingües como XLM-RoBERTa o mDeBERTa. En general, los sistemas más competitivos fueron aquellos que realizaron *fine-tuning* sobre estos modelos, utilizando el historial de mensajes del usuario como unidad de entrada. Algunas propuestas optaron por agrupar mensajes en bloques temporales o semánticos para capturar mejor la evolución del discurso, mientras que otras aplicaron técnicas de

aumento de datos, como la retrotraducción, para enriquecer los conjuntos de entrenamiento.

A nivel de tareas, se abordaron dos variantes: una clasificación binaria para detectar la presencia o ausencia de trastornos alimentarios (tarea 1.a) y una regresión simple para predecir la probabilidad de que un usuario padezca uno de estos trastornos (tarea 1.b). En ambas subtareas, los modelos basados en *Transformers* superaron con claridad a los enfoques clásicos, especialmente en la clasificación binaria, donde se lograron puntuaciones de *Macro-F1* superiores a 0.96. No obstante, también se observaron resultados competitivos con técnicas más tradicionales como Naïve Bayes, en particular cuando se emplearon vectores de caracteres o vocabularios adaptados al dominio.

En cuanto a la tarea de regresión, los modelos preentrenados como RoBERTa Base lograron los mejores resultados en términos de error cuadrático medio (RMSE), alcanzando valores tan bajos como 0.178 y coeficientes de correlación de Pearson superiores al 0.90, lo que indica una gran proximidad a las valoraciones humanas. Aunque algunos sistemas exploraron enfoques de ensamblado (*ensembles*) o enriquecimiento con variables adicionales, las mejoras frente a los modelos base fueron generalmente modestas.

En conjunto, los resultados de esta edición muestran una clara ventaja de los modelos Transformer preentrenados, especialmente aquellos específicos para español, aunque también evidencian que ciertos enfoques más ligeros o adaptativos —como los basados en caracteres o la personalización del vocabulario— pueden ofrecer rendimientos competitivos, particularmente en tareas donde el lenguaje utilizado presenta un léxico muy específico, como ocurre con los trastornos alimentarios.

3.3 Modelos utilizados en este trabajo

En este trabajo se han empleado cuatro configuraciones distintas de modelos de lenguaje preentrenados en español con el objetivo de detectar señales de trastornos de la conducta alimentaria en mensajes publicados en grupos públicos de Telegram. Estos modelos representan distintas estrategias de aprovechamiento del conocimiento lingüístico aprendido durante el preentrenamiento: por un lado, el uso de representaciones estáticas mediante extracción de embeddings; y por otro, el ajuste fino o *fine-tuning* del modelo completo para la tarea específica. En cualquier caso, y para diferenciar este estudio de los mencionados en la sección anterior, se ha llevado a cabo con todos los modelos un análisis multietiqueta, en contraposición con la clasificación binaria que se llevaba a cabo en la primera tarea del estudio de *MentalRiskES*.

En primer lugar, se ha utilizado el modelo **BETO** en dos variantes. BETO es

una implementación del modelo BERT entrenado exclusivamente con corpus en español, como Wikipedia y OpenSubtitles, lo que lo convierte en una buena opción para tareas de procesamiento del lenguaje natural en castellano (Rosenbrock et al., 2021). En su versión basada en *embeddings*, se ha extraído la representación contextual de los mensajes utilizando las capas superiores del modelo, y estas representaciones han servido como entrada a clasificadores externos, como redes neuronales densas o modelos tradicionales. En la versión con *fine-tuning*, BETO se ha ajustado directamente sobre los datos etiquetados del corpus *MentalRiskES*, adaptando todos los parámetros del modelo a la tarea de clasificación multiclase definida en este trabajo.

En segundo lugar, se ha empleado el modelo **MarIA**, un modelo BERT entrenado sobre el corpus del Plan Nacional de Lengua Española, que incluye una gran diversidad de géneros textuales, desde noticias hasta textos jurídicos y científicos (Gutiérrez-Fandiño et al., 2022). En este caso, se ha optado exclusivamente por el uso de *embeddings* sin ajuste fino, extrayendo las representaciones del CLS o mediante agregación media de los tokens para cada mensaje. Esta decisión se fundamenta en el interés por comparar el rendimiento entre modelos ajustados y no ajustados, así como en la estabilidad que ofrecen los embeddings preentrenados ante posibles sesgos de sobreajuste.

Por último, se ha utilizado **ALBETO**, un modelo más ligero que BETO, optimizado para mayor eficiencia computacional sin una pérdida significativa de rendimiento (Cañete et al., 2022). En este trabajo, ALBETO se ha empleado exclusivamente mediante *fine-tuning*, ajustando el modelo completo sobre los datos anotados. Esta opción permite comparar su rendimiento con respecto a modelos más pesados como BETO, especialmente en contextos donde los recursos computacionales pueden ser limitados.

La selección de estos cuatro enfoques —BETO con embeddings, MarIA con embeddings, BETO con ajuste fino y ALBETO con ajuste fino— permite realizar un análisis comparativo no solo entre diferentes arquitecturas, sino también entre paradigmas metodológicos. Así, se pretende evaluar empíricamente si el *fine-tuning* sobre un corpus pequeño y sensible como *MentalRiskES* proporciona mejoras sustanciales frente al uso directo de representaciones preentrenadas sin ajuste, y hasta qué punto influyen el tamaño y la configuración del modelo en la detección de riesgo en mensajes sociales relacionados con trastornos alimentarios.

3.4 Resumen Metodológico

El desarrollo de este trabajo ha seguido una secuencia estructurada de etapas que permiten abordar de forma sistemática el problema de detección de trastornos alimentarios en mensajes provenientes de redes sociales. El primer paso consistió en el preprocesamiento del corpus original proporcionado por *MentalRiskES*. Este corpus,

en su formato original, contenía mensajes sin tratar, por lo que fue necesario aplicar una serie de transformaciones para homogeneizar los datos: se convirtieron los *emojis* en sus correspondientes descripciones textuales y se prepararon las entradas para su procesamiento por modelos de lenguaje.

En primer lugar, se implementó un modelo base para la clasificación de texto mediante la implementación de modelos basados en representaciones Bag of Words (BOW). Este enfoque fundamental en el Procesamiento del Lenguaje Natural (PLN) transforma cada documento de texto en un vector numérico, esencialmente ignorando el orden de las palabras y tratándolas como elementos individuales dentro de una "bolsa". Para construir estos vectores, se define un vocabulario único a partir de todo el corpus de documentos; luego, la representación de cada texto se genera cuantificando la frecuencia con la que cada palabra del vocabulario aparece en él, o simplemente registrando su presencia binaria. Una vez convertidos en estos vectores numéricos, los datos se introdujeron en clasificadores clásicos de machine learning, como las Máquinas de Vectores de Soporte (SVM) y los algoritmos de Naive Bayes.

Una vez finalizada esta fase, se procedió al diseño y entrenamiento de dos tipos de enfoques. En primer lugar, se implementaron modelos basados en *embeddings* contextualizados. En este caso, los modelos BETO y MarIA se utilizaron como generadores de representaciones vectoriales para los mensajes, sin modificar sus parámetros internos. Estas representaciones se integraron en arquitecturas de clasificación externas, generalmente redes densas, que aprendían a partir de los embeddings la tarea de categorización multiclase definida en este trabajo.

Posteriormente, se ha llevado a cabo el fine-tuning de modelos de lenguaje basados en transformadores preentrenados, concretamente BETO y ALBETO, ambos adaptados al español. El fine-tuning ha consistido en ajustar estos modelos sobre el conjunto de entrenamiento para optimizar su rendimiento en la clasificación multilabel. Se han probado versiones con las cinco clases originales y versiones reducidas con solo tres clases, eliminando las etiquetas minoritarias para analizar el impacto en el desempeño.

Finalmente, se llevó a cabo una fase de comparación de resultados, tanto cuantitativa como cualitativa. Se evaluó el rendimiento de cada enfoque mediante métricas estándar como la precisión, la recuperación y el F1, y se analizaron posibles diferencias en la sensibilidad de los modelos ante determinadas clases. Este análisis permitió identificar fortalezas y limitaciones de cada configuración, así como reflexionar sobre el papel del *fine-tuning* frente al uso directo de embeddings en tareas clínicas con datos sensibles.

4 Desarrollo del trabajo

En este capítulo se detalla el proceso completo de desarrollo del trabajo, desde la carga de los datos hasta la aplicación de distintos modelos de aprendizaje automático y procesamiento del lenguaje natural. A lo largo de este proceso se emplearon diversas librerías de Python, fundamentales para la manipulación de datos, preprocesamiento y entrenamiento de modelos. A continuación se detallan las principales librerías instaladas y su propósito:

- **scikit-learn**: Librería fundamental para la implementación de modelos clásicos de aprendizaje automático, así como herramientas de evaluación y validación cruzada.
- **pandas**: Para la manipulación eficiente de estructuras de datos tabulares (DataFrames).
- **emoji**: Utilizada para convertir emojis en texto legible, permitiendo su inclusión en el análisis semántico.
- **transformers**: Proporciona acceso a modelos preentrenados de lenguaje como BERT, RoBERTa, y DistilBERT, facilitando su uso para tareas de clasificación de texto.
- **datasets**: Permite el acceso y gestión eficiente de conjuntos de datos, y está integrada con la librería `transformers`.
- **torch**: Framework de aprendizaje profundo utilizado para entrenar modelos basados en redes neuronales.
- **numpy**: Para realizar operaciones matemáticas de bajo nivel y manipulación de vectores y matrices.
- **iterative-stratification**: Librería utilizada para realizar particiones estratificadas multilabel, útil cuando se trabaja con múltiples etiquetas simultáneamente.

Todas las librerías fueron instaladas desde el entorno de Google Colab mediante el comando `%pip`.

4.1 Carga de datos

Los datos utilizados para este trabajo fueron proporcionados por la Universidad de Murcia y se encuentran almacenados en su nube institucional. Para facilitar el acceso, los datos se descargan de forma automática desde Google Colab mediante comandos que emplean herramientas como `wget` y `unzip`. A continuación, se muestra el fragmento de código que realiza esta operación:

Código 4.1: Descarga de datos desde la nube de la Universidad de Murcia

```
# Crear carpeta para los datos
!mkdir -p Data

# Descargar gold_label.csv
!wget --no-check-certificate -O gold_label.csv "https://umubox.um.es/↩
    ↪ index.php/s/fug4fbpi26VcNUY/download"

# Descargar carpeta Data con los .json comprimidos
!wget --no-check-certificate -O Data.zip "https://umubox.um.es/index.php/↩
    ↪ s/gky5M4xlhg8UC8P/download"

# Descomprimir Data.zip dentro de Data/
!unzip -o Data.zip -d Data/

# Eliminar el zip para liberar espacio
!rm Data.zip
```

En primer lugar, se crea un directorio local llamado `Data` para almacenar los archivos descargados. Posteriormente, se descargan dos recursos desde la nube institucional: un archivo `CSV` llamado `gold_label.csv`, que contiene las etiquetas de los datos, y un archivo comprimido `Data.zip`, que contiene múltiples archivos `JSON` con los textos originales. El archivo comprimido se descomprime automáticamente en la carpeta designada y, una vez completado este proceso, se elimina para optimizar el uso del espacio en el entorno de ejecución.

4.1.1 Análisis de los datos

Antes de entrenar los modelos, se ha llevado a cabo un análisis estadístico exploratorio del conjunto de datos para comprender mejor la distribución de las etiquetas. En la Tabla 4.1 se presentan las estadísticas descriptivas para cada una de las cinco etiquetas multilabel: *bs* (suffer), *bsf* (suffer-in favour), *bsa* (suffer-against), *bso* (suffer-other) y *bc* (control).

Tabla 4.1: Estadísticas descriptivas de las etiquetas multilabel

Estadística	**bs**	**bsf**	**bsa**	**bso**	**bc**
count	335	335	335	335	335
mean	0.427	0.406	0.018	0.003	0.573
std	0.495	0.492	0.133	0.055	0.495
min	0	0	0	0	0
25%	0	0	0	0	0
50%	0	0	0	0	1
75%	1	1	0	0	1
max	1	1	1	1	1

Como se puede observar, la media de las etiquetas *bs* y *bsf* es cercana a 0.4, lo que indica una distribución relativamente equilibrada entre clases positivas y negativas. Por el contrario, las etiquetas *bsa* y especialmente *bso* presentan una media próxima a cero, reflejando una fuerte desproporción y anticipando posibles dificultades de los modelos para aprender representaciones útiles de estas clases minoritarias. La etiqueta *bc*, correspondiente a usuarios considerados como controles, aparece con una frecuencia mayoritaria.

Para analizar con mayor claridad la proporción entre ejemplos positivos y negativos en cada clase, se ha generado la Tabla 4.2, que muestra el número exacto de instancias para cada valor binario.

Tabla 4.2: Distribución binaria de etiquetas (conteo de clases 0 y 1)

Etiqueta	**Clase 0**	**Clase 1**
bs	192	143
bsf	199	136
bsa	329	6
bso	334	1
bc	143	192

La tabla confirma la fuerte descompensación en las clases *bsa* y *bso*, donde las clases positivas representan únicamente el 1.8% y 0.3% de los casos, respectivamente. Esta asimetría podría influir negativamente en el rendimiento de los modelos en estas categorías, especialmente si no se aplican técnicas de balanceo. En contraste, las etiquetas *bs*, *bsf* y *bc* presentan una distribución más equitativa, lo cual resulta más favorable para los modelos de aprendizaje supervisado.

4.2 Preprocesamiento

Previo al entrenamiento de los modelos, se realizó una fase de preprocesamiento sobre los textos con el objetivo de mejorar la calidad de los datos y extraer mayor valor semántico. Uno de los elementos más relevantes del preprocesamiento fue el tratamiento de los *emojis*, símbolos que suelen aparecer con frecuencia en mensajes informales y que pueden aportar significado emocional importante.

En lugar de eliminar los *emojis*, se optó por transformarlos en su equivalente textual mediante la librería `emoji`. Para ello, se diseñó una función que convierte los *emojis* en descripciones en español y limpia el formato para hacerlo más legible y procesable por los modelos. El código correspondiente es el siguiente:

Código 4.2: Conversión de emojis a texto legible

```
import emoji
import re

def replace_emojis(text):
    text = emoji.demojize(text, language="es") # Convertir emojis a texto
    text = re.sub(r":(\w+):", lambda m: m.group(1).replace("_", " "), text
        ↪ ) # Limpiar formato
    return text
```

Esta transformación permite que los modelos de lenguaje puedan interpretar los emojis como conceptos lingüísticos (por ejemplo, “cara pensativa”) en lugar de símbolos gráficos sin contexto, enriqueciendo así el análisis emocional y semántico de los textos.

4.3 Modelo base

Antes de aplicar modelos de clasificación, se realizó un preprocesamiento esencial para consolidar los datos. Los mensajes de cada usuario estaban distribuidos en múltiples archivos con formato `.json`, por lo que fue necesario recorrer todos los archivos, extraer los mensajes y agruparlos por usuario. Esta consolidación facilitó el posterior análisis, permitiendo representar cada usuario como un único documento textual.

Preprocesamiento. Unificación de mensajes por usuario

El siguiente fragmento de código lee cada archivo `.json`, extrae los mensajes, los concatena en un solo texto por usuario y guarda el resultado en un archivo `CSV`.

Código 4.3: Unificación de mensajes por usuario

```
data_folder = "Datos/preprocesados/"
output_file = os.path.join("Datos/", "usuarios_mensajes.csv")

subject_texts = {}

# Recorrer todos los archivos JSON
for filename in os.listdir(data_folder):
    if filename.endswith(".json"):
        subject_id = filename.replace(".json", "").strip().lower() # Obtener identificador del usuario
        with open(os.path.join(data_folder, filename), "r", encoding="utf-8") as f:
            messages = json.load(f)
            # Concatenar todos los mensajes en una sola cadena
            subject_texts[subject_id] = " ".join([msg["message"] for msg in messages])

# Convertir el diccionario a DataFrame
df_texts = pd.DataFrame(list(subject_texts.items()), columns=["usuario", "mensaje"])
df_texts.to_csv(output_file, index=False, encoding="utf-8")
```

Este procedimiento tiene como objetivo representar cada usuario mediante un solo documento, conformado por todos sus mensajes.

Vectorización y modelo de clasificación

Una vez preparado el conjunto de datos, se procede a aplicar técnicas de vectorización y modelado. El objetivo es construir un modelo base que actúe como punto de referencia frente a futuras mejoras con técnicas más avanzadas.

Se cargan los textos procesados junto con el archivo de etiquetas (`gold_label.csv`), que contiene las etiquetas asignadas a cada usuario. Luego, se unen ambos conjuntos de datos en un único DataFrame.

Código 4.4: Unificación de mensajes por con etiquetas

```
# Cargar datos
df = pd.read_csv("Datos/usuarios_mensajes.csv")
labels_df = pd.read_csv("gold_label.csv", delimiter="\t")

# Unir los textos con sus etiquetas usando el identificador del usuario
df = df.merge(labels_df, left_on="usuario", right_on="nick")
```

Las etiquetas corresponden a un esquema de clasificación multilabel, es decir, cada usuario puede tener múltiples etiquetas asociadas. Se extraen dichas columnas y se divide el conjunto de datos en entrenamiento y prueba.

Código 4.5: División de entrenamiento y prueba BoW

```
# Seleccionar etiquetas multilabel
etiquetas = ["bs", "bsf", "bsa", "bso", "bc"]
y = df[etiquetas]

# División en entrenamiento (80\%) y prueba (20\%)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(df["mensaje"], y, ↩
    ↪ test_size=0.2, random_state=42)
```

4.3.1 Vectorización mediante Bag of Words (BoW)

La técnica de Bag of Words transforma el texto en vectores numéricos, donde cada dimensión representa la frecuencia de una palabra. Se limita el vocabulario a las 5000 palabras más frecuentes para reducir la dimensionalidad.

Código 4.6: Vectorización con BoW

```
vectorizer = CountVectorizer(max_features=5000)
X_train_bow = vectorizer.fit_transform(X_train)
X_test_bow = vectorizer.transform(X_test)
```

Entrenamiento con Naïve Bayes

Se utiliza el clasificador Naïve Bayes Multinomial, adecuado para datos discretos como conteos de palabras. Dado que se trata de clasificación multilabel, se emplea la estrategia `OneVsRest`, que entrena un clasificador independiente para cada etiqueta.

Código 4.7: Entrenamiento de BoW con Naïve Bayes

```
nb_model = OneVsRestClassifier(MultinomialNB())
nb_model.fit(X_train_bow, y_train)
y_pred_nb = nb_model.predict(X_test_bow)

print("Naïve Bayes (Multi-label):")
print(classification_report(y_test, y_pred_nb, target_names=etiquetas))
```

Entrenamiento con SVM

También se entrena un clasificador SVM (Support Vector Machine) con kernel lineal, usando la misma estrategia de `OneVsRest`. Este modelo es más robusto frente a casos linealmente separables y suele ofrecer mejor rendimiento que Naïve Bayes en tareas de texto.

Código 4.8: Entrenamiento de BoW con SVM

```
svm_model = OneVsRestClassifier(SVC(kernel="linear", probability=True))
svm_model.fit(X_train_bow, y_train)
y_pred_svm = svm_model.predict(X_test_bow)

print("SVM (Multi-label):")
print(classification_report(y_test, y_pred_svm, target_names=etiquetas))
```

Ambos modelos son evaluados utilizando el informe de clasificación proporcionado por `sklearn`, que incluye métricas por etiqueta como precisión, recall y F1-score. Esta evaluación permite identificar qué etiquetas son más fáciles o más difíciles de predecir con un enfoque clásico basado únicamente en conteo de palabras. Este modelo base establece una línea de referencia inicial con métodos tradicionales, sobre la cual se compararán posteriormente los modelos basados en representaciones semánticas más sofisticadas como embeddings y técnicas de *fine-tuning*.

Resultados del modelo base

Tal como puede observarse en la tabla 4.3, se muestran los resultados obtenidos al aplicar los clasificadores Naïve Bayes y SVM sobre las representaciones Bag of Words del texto. Las métricas evaluadas son *precision*, *recall* y *f1-score*, calculadas por etiqueta y también en promedio.

Los resultados obtenidos con los modelos Naïve Bayes y SVM evidencian un buen desempeño para las etiquetas bs (suffer), bsf (suffer in favour) y bc (control), con métricas de precisión, recall y F1-score altas, cercanas o superiores al 90%. Estas categorías representan usuarios que sufren del trastorno y buscan ayuda o la fomentan de distintas maneras, así como usuarios considerados controles, que no muestran síntomas o están involucrados en otros roles.

Por otro lado, las etiquetas bsa (suffer against) y bso (suffer other) muestran métricas nulas en todas las métricas evaluadas. Esta baja precisión se explica por la escasa representación de estas clases en el conjunto de datos, ya que incluyen usuarios que sufren del trastorno pero tienen un comportamiento o interacción menos común o poco representado en los mensajes.

Tabla 4.3: Métricas comparativas de clasificación multilabel para Naïve Bayes y SVM

Etiqueta	**Naïve Bayes**			**SVM**		
	Prec.	Rec.	F1	Prec.	Rec.	F1
bs	0.93	0.90	0.92	0.93	0.93	0.93
bsf	0.93	0.86	0.89	0.96	0.90	0.93
bsa	0.00	0.00	0.00	0.00	0.00	0.00
bso	0.00	0.00	0.00	0.00	0.00	0.00
bc	0.92	0.95	0.93	0.95	0.95	0.95
Micro avg	0.93	0.90	0.91	0.95	0.92	0.93
Macro avg	0.56	0.54	0.55	0.57	0.56	0.56
Weighted avg	0.92	0.90	0.91	0.94	0.92	0.93
Samples avg	0.93	0.91	0.92	0.94	0.93	0.93

La distribución desbalanceada y la insuficiente cantidad de ejemplos para bsa y bso limitan la capacidad de los modelos para aprender patrones característicos, resultando en una imposibilidad práctica de clasificar correctamente estas etiquetas. En vistas de estos resultados, en futuros análisis que utilicen modelos basados en técnicas de fine-tuning se eliminarán las etiquetas bsa y bso para concentrar el aprendizaje en las clases mejor representadas y facilitar la construcción de modelos más robustos y con mejores capacidades predictivas.

4.4 Modelos con embeddings

El objetivo de esta sección es clasificar los mensajes de usuarios en cinco posibles etiquetas que reflejan su relación con un trastorno emocional: `bs`, `bsf`, `bsa`, `bso` y `bc`. Para mejorar el rendimiento, se aplicaron modelos de lenguaje preentrenados que transforman los mensajes en vectores semánticos (embeddings). Estos embeddings permiten representar el contenido emocional y contextual de los mensajes, capturando significados más profundos que enfoques tradicionales como Bag-of-Words o TF-IDF.

Preprocesamiento. Unificación de datos

Se comenzó cargando los mensajes de usuarios y las etiquetas manuales correspondientes. Estas se unificaron por nombre de usuario (`nick`) mediante un `merge`, y se eliminaron las columnas redundantes:

Código 4.9: Unificación de mensajes y etiquetas

```
usuarios_mensajes = pd.read_csv("Datos/usuarios_mensajes.csv")
labels = pd.read_csv("gold_label.csv", sep='\t')

etiquetas = ["bs", "bsf", "bsa", "bso", "bc"]
usuarios_mensajes_labels = usuarios_mensajes.merge(
    labels[["nick"] + etiquetas],
    left_on="usuario",
    right_on="nick"
)
usuarios_mensajes_labels = usuarios_mensajes_labels.drop(columns=["nick"]
    ↪ ])
```

El resultado es un `DataFrame` con una columna de mensajes y cinco columnas binarias que indican la pertenencia del usuario a cada una de las etiquetas.

Funciones para modelos con embeddings

Para codificar los mensajes, se usó la librería `sentence-transformers` que permite aplicar modelos BERT para obtener vectores representativos de cada texto:

Código 4.10: Función para generar embeddings

```
def get_embeddings(model_name, textos):
    model = SentenceTransformer(model_name)
    return np.array([model.encode(text) for text in textos])
```

Se implementó una función de evaluación basada en regresión logística multietiqueta con validación cruzada estratificada de 5 divisiones. Se reportaron métricas de accuracy y F1 macro, además de informes de clasificación por clase.

Código 4.11: Evaluación del modelo multilabel

```
def evaluar_modelo_multilabel(X, y, nombre_modelo):
    clf = OneVsRestClassifier(LogisticRegression(max_iter=1000))
    kf = KFold(n_splits=5, shuffle=True, random_state=42)
    f1_scores, accuracies = [], []

    for train_idx, test_idx in kf.split(X):
        clf.fit(X[train_idx], y.iloc[train_idx])
        y_pred = clf.predict(X[test_idx])

        f1_scores.append(f1_score(y.iloc[test_idx], y_pred, average="macro
            ↪ "))
```

```
        accuracies.append(np.mean(y_pred == y.iloc[test_idx]))

    clf.fit(X, y)
    y_pred_final = clf.predict(X)
    report = classification_report(y, y_pred_final, target_names=y.columns)

    return {
        "accuracy": np.mean(accuracies),
        "f1_macro": np.mean(f1_scores),
        "classification_report": report
    }
```

4.4.1 Modelo BETO

Al aplicar el modelo BETO para generar embeddings de los mensajes y entrenar un clasificador multietiqueta, se obtuvieron los resultados mostrados en la tabla 4.4. Se obtuvo un F1 macro promedio de 0.72 y una accuracy promedio de 0.96. Estos resultados indican un buen rendimiento general, especialmente en etiquetas con mayor cantidad de ejemplos como `bs` (suffer), `bsf` (suffer in favour) y `bc` (control).

Etiqueta	Precision	Recall	F1-score	Soporte
bs	0.99	0.94	0.96	143
bsf	0.98	0.96	0.97	136
bsa	1.00	0.50	0.67	6
bso	0.00	0.00	0.00	1
bc	0.95	0.99	0.97	192
Promedio micro	0.97	0.96	0.97	478
Promedio macro	0.79	0.68	0.72	478
Promedio ponderado	0.97	0.96	0.97	478
Promedio por muestra	0.97	0.97	0.97	478

Tabla 4.4: Reporte de clasificación del modelo BETO

Sin embargo, el modelo tiene dificultades para predecir clases poco representadas como `bsa` (suffer against) y `bso` (suffer other), lo cual es esperable en escenarios de desequilibrio de clases. A pesar de ello, la precisión en etiquetas frecuentes demuestra la capacidad del modelo para captar la estructura semántica del discurso relacionado con los trastornos mentales.

4.4.2 Modelo MarIA

MarIA es un modelo tipo RoBERTa entrenado sobre textos biomédicos y clínicos en español. Dado su enfoque especializado, se evaluó con la hipótesis de que podría ser útil en la identificación de patrones discursivos relacionados con la salud mental, como la depresión o la ansiedad.

Tal como se puede comprobar en la tabla 4.5, el modelo obtuvo un F1 macro promedio de 0.56 y una accuracy promedio de 0.92. Aunque la precisión en etiquetas frecuentes como `bs`, `bsf` y `bc` sigue siendo alta, el rendimiento general es inferior al de BETO, especialmente en clases minoritarias.

Etiqueta	Precision	Recall	F1-score	Soporte
bs	0.96	0.92	0.94	143
bsf	0.95	0.89	0.92	136
bsa	0.00	0.00	0.00	6
bso	0.00	0.00	0.00	1
bc	0.94	0.97	0.96	192
Promedio micro	0.95	0.92	0.94	478
Promedio macro	0.57	0.56	0.56	478
Promedio ponderado	0.94	0.92	0.93	478
Promedio por muestra	0.94	0.94	0.94	478

Tabla 4.5: Reporte de clasificación del modelo MarIA

Esto sugiere que, a pesar de su entrenamiento sobre datos biomédicos, MarIA no se adapta tan bien al lenguaje informal y subjetivo de las redes sociales o foros, donde el contenido emocional y coloquial es predominante. Además, las clases menos representadas siguen mostrando una capacidad predictiva limitada.

Comparación de resultados

Los resultados de la tabla 4.6 muestran que BETO obtiene un desempeño claramente superior a MarIA en términos de F1 macro, lo que sugiere una mejor capacidad de balance entre clases. En cambio, MarIA tiende a sobreajustarse a las clases mayoritarias, fallando en las minoritarias (`bsa` y `bso`) con F1 igual a 0. Esto puede atribuirse a la especialización de MarIA en textos clínicos formales, que difieren del lenguaje emocional e informal de los mensajes. Por su parte, BETO, al haber sido entrenado en corpus generales en español, se adapta mejor a ese tipo de discurso.

Modelo	Accuracy promedio	F1 Macro promedio
BETO	0.96	0.72
MarIA	0.92	0.56

Tabla 4.6: Comparación entre modelos BETO y MarIA

Las clases `bsa` y `bso` tienen muy pocos ejemplos, lo cual dificulta al modelo aprender patrones representativos.

4.5 Modelos con Fine-Tuning

En esta sección se presentan los experimentos de fine-tuning realizados con modelos basados en arquitecturas BERT entrenadas para español, concretamente BETO y ALBETO. Para cada modelo, se trabajó con dos configuraciones de etiquetas: una completa con cinco clases y otra simplificada con tres clases. La idea principal fue adaptar estas potentes representaciones lingüísticas a la tarea multilabel de clasificación de mensajes.

Para asegurar la reproducibilidad, se configuró una semilla fija que controla la aleatoriedad en todas las librerías involucradas en el proceso. Además, se desactivó el paralelismo en los tokenizadores para evitar conflictos en la ejecución.

Código 4.12: Configuración de semilla y entorno para reproducibilidad

```
seed = 42
random.seed(seed)
np.random.seed(seed)
torch.manual_seed(seed)
torch.cuda.manual_seed_all(seed)
os.environ['PYTHONHASHSEED'] = str(seed)
os.environ['TOKENIZERS_PARALLELISM'] = 'false'
```

Esta configuración es clave para que los resultados sean comparables y reproducibles entre distintas ejecuciones y entornos.

4.5.1 Modelo BETO

Primero, se carga el tokenizador oficial de BETO para procesar el texto de entrada, convirtiendo los mensajes en secuencias numéricas que el modelo puede interpretar.

Código 4.13: Tokenizador BETO

```
tokenizer = AutoTokenizer.from_pretrained("dccuchile/bert-base-spanish-wwm-cased")
```

A continuación, se definen los inputs y las etiquetas multilabel a partir del DataFrame original. Aquí, las etiquetas corresponden a las cinco categorías originales del problema, representadas como vectores binarios.

Código 4.14: Preparación de inputs y etiquetas multilabel

```
X = df["mensaje"].values
y = df[["bs", "bsf", "bsa", "bso", "bc"]].values
```

Para dividir el conjunto de datos en entrenamiento, validación y prueba, se utiliza una técnica llamada *Multilabel Stratified Shuffle Split*. A diferencia de un split aleatorio simple, esta técnica asegura que la distribución de todas las etiquetas multilabel se mantenga proporcional en cada subconjunto, lo que es especialmente importante para etiquetas poco frecuentes.

Primero, se separa un 60% de datos para entrenamiento y un 40% temporal para validación y prueba.

Código 4.15: Primer split entre entrenamiento y prueba en fine-tuning

```
msss = MultilabelStratifiedShuffleSplit(n_splits=1, test_size=0.4, random_state=42)
train_idx, temp_idx = next(msss.split(X, y))
```

Luego, se divide ese 40% temporal en dos partes iguales de 20% cada una para validación y test, respectivamente.

Código 4.16: Segundo split entre validación y prueba en fine-tuning

```
msss2 = MultilabelStratifiedShuffleSplit(n_splits=1, test_size=0.5, random_state=42)
val_idx, test_idx = next(msss2.split(X[temp_idx], y[temp_idx]))
```

Una vez definidos los conjuntos, se carga el modelo BETO para clasificación multilabel configurado con cinco salidas, una por cada clase. Se especifica que el problema es de multilabel para que el modelo use la función de pérdida adecuada, y se proporcionan los diccionarios que mapean entre índices y etiquetas.

Código 4.17: Carga del modelo BETO para multilabel

```
model = AutoModelForSequenceClassification.from_pretrained(
    "dccuchile/bert-base-spanish-wwm-cased",
    num_labels=len(etiquetas),
```

```
    problem_type="multi_label_classification",
    id2label=int_to_label,
    label2id=label_to_int
)
```

Para evaluar correctamente la tarea multilabel, se define una función que aplica la función sigmoide a las predicciones del modelo para convertirlas en probabilidades entre 0 y 1. Luego, se utiliza un umbral (por defecto 0.5) para decidir si cada etiqueta se activa o no. Sobre estas predicciones binarias se calculan métricas de desempeño multilabel como F1 micro, macro, weighted y exactitud.

Código 4.18: Función para cálculo de métricas multilabel

```
def multi_label_metrics(predictions, labels, threshold=0.5):
    sigmoid = torch.nn.Sigmoid()
    probs = sigmoid(torch.Tensor(predictions))
    y_pred = np.zeros(probs.shape)
    y_pred[np.where(probs >= threshold)] = 1

    f1_micro = f1_score(labels, y_pred, average='micro')
    f1_macro = f1_score(labels, y_pred, average='macro')
    f1_weighted = f1_score(labels, y_pred, average='weighted')
    accuracy = accuracy_score(labels, y_pred)

    return {
        "f1_micro": f1_micro,
        "f1_macro": f1_macro,
        "f1_weighted": f1_weighted,
        "accuracy": accuracy
    }
```

Esta función es utilizada durante el entrenamiento para monitorizar el rendimiento en validación mediante la función `compute_metrics`, que adapta la salida del trainer para que use nuestras métricas personalizadas.

Código 4.19: Función para adaptar métricas al Trainer

```
def compute_metrics(p):
    preds = p.predictions[0] if isinstance(p.predictions, tuple) else p.↩
        ↪ predictions
    return multi_label_metrics(predictions=preds, labels=p.label_ids)
```

Finalmente, se configuran los parámetros de entrenamiento, como tasa de aprendizaje, tamaño de batch, número de épocas, y criterios para guardar el mejor modelo. Se especifica que la métrica principal para elegir el mejor modelo sea el F1 micro, dado

que es una métrica robusta para multilabel.

Código 4.20: Configuración de argumentos para el entrenamiento

```
training_args = TrainingArguments(
    output_dir="./beto-finetuned",
    eval_strategy="epoch",
    save_strategy="epoch",
    learning_rate=2e-5,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    num_train_epochs=5,
    weight_decay=0.01,
    load_best_model_at_end=True,
    metric_for_best_model="eval_f1_micro",
    report_to="none",
    seed=42,
)
```

Con todo listo, se crea un objeto `Trainer` que dirige el proceso de entrenamiento y evaluación, recibiendo el modelo, los datos tokenizados de entrenamiento y validación, el tokenizador y la función de métricas.

Código 4.21: Definición del Trainer para fine-tuning

```
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=encoded_train,
    eval_dataset=encoded_eval,
    tokenizer=tokenizer,
    compute_metrics=compute_metrics,
)
```

El entrenamiento se lanza con `trainer.train()`, y una vez finalizado se evalúa el rendimiento en el conjunto de prueba con `trainer.evaluate(encoded_test)`. Los datos se almacenan para una posterior comparación.

BETO con tres clases

El procedimiento para la versión de BETO con tres clases es muy similar al anterior. La única diferencia fundamental está en la selección de etiquetas, donde se trabaja con un subconjunto reducido de tres clases,las más frecuentes o relevantes para simplificar el problema.

El preprocesamiento de datos, división de conjuntos, carga del tokenizador, y

configuración del modelo se realizan de forma análoga, adaptando el número de salidas a tres. El código para la carga del modelo cambiaría así:

Código 4.22: Modelo BETO para tres clases

```
model = AutoModelForSequenceClassification.from_pretrained(
    "dccuchile/bert-base-spanish-wwm-cased",
    num_labels=3, # Ajustado a tres clases
    problem_type="multi_label_classification",
    id2label=int_to_label,
    label2id=label_to_int
)
```

Todo lo demás permanece igual, incluyendo la configuración del entrenamiento, la función de métricas multilabel y el uso del `Trainer`. Este enfoque permite comparar el comportamiento del modelo con menos clases y analizar si la simplificación mejora el rendimiento.

Comparación de resultados de BETO

Tabla 4.7: Comparación de resultados de fine-tuning de BETO con 5 clases y 3 clases

Métrica	BETO 5 clases	BETO 3 clases
Pérdida de evaluación (eval_loss)	0.2020	0.1227
F1 micro (eval_f1_micro)	0.895	0.941
F1 macro (eval_f1_macro)	0.538	0.566
F1 ponderado (eval_f1_weighted)	0.890	0.936
Exactitud (eval_accuracy)	0.896	0.940

Tal como se observa en la tabla 4.7, los resultados muestran que al simplificar el problema de clasificación multilabel reduciendo el número de clases de cinco a tres (eliminando las minoritarias), el modelo BETO obtiene mejoras claras en varias métricas clave. Primero, la pérdida de evaluación disminuye significativamente (de 0.202 a 0.123), lo que indica que el modelo se ajusta mejor y hace predicciones más precisas en general para el conjunto de prueba.

El F1 micro, que resume el rendimiento global tomando en cuenta el total de verdaderos positivos, falsos positivos y falsos negativos, aumenta de aproximadamente 0.895 a 0.941, mostrando que el modelo mejora su capacidad de identificar correctamente las etiquetas activas cuando se reducen las clases minoritarias. El F1 macro, que calcula la métrica por clase y luego promedia, también aumenta ligeramente, reflejando una mejora en el equilibrio de desempeño entre las diferentes categorías.

En conjunto, estos resultados sugieren que eliminar clases minoritarias puede

facilitar el aprendizaje y mejorar la actuación del modelo BETO en este problema multietiqueta, posiblemente porque reduce el ruido y el desequilibrio en las etiquetas. Sin embargo, esto también implica perder información de ciertas categorías que podrían ser relevantes en escenarios prácticos, por lo que esta decisión debe valorarse en función del caso de uso.

4.6 Modelos ligeros

4.6.1 Modelo ALBETO

En esta subsección se presenta el proceso seguido para el fine-tuning del modelo ALBETO, siguiendo una metodología similar a la aplicada con BETO. ALBETO es una variante más ligera de BERT optimizada para el español, que permite realizar tareas de clasificación con eficiencia computacional. Para este modelo se utilizó el tokenizador y el modelo preentrenado de la siguiente manera:

Código 4.23: Carga del modelo ALBETO para clasificación multilabel

```
model_albeto = AutoModelForSequenceClassification.from_pretrained(
    "dccuchile/albert-base-spanish",
    num_labels=len(etiquetas),
    problem_type="multi_label_classification",
    id2label=int_to_label,
    label2id=label_to_int
)
```

Al igual que con BETO, se prepararon los datos separando los textos y las etiquetas multilabel. Se realizó una división estratificada en tres conjuntos: entrenamiento, validación y prueba, utilizando `MultilabelStratifiedShuffleSplit` para preservar la distribución de las etiquetas. El proceso de entrenamiento siguió las mismas pautas, configurando los argumentos del `Trainer` para optimizar con la métrica de F1 micro, ajustando parámetros como tasa de aprendizaje, tamaño de lote y número de épocas, asegurando reproducibilidad mediante la configuración de semillas para todas las librerías involucradas. El entrenamiento se llevó a cabo mediante `trainer.train()`, y la evaluación final se realizó sobre el conjunto de prueba con `trainer.evaluate(encoded_test)`,

Comparación de resultados de ALBETO

Se pueden observar en la tabla 4.8 los resultados obtenidos tras el fine-tuning de ALBETO con el conjunto completo de 5 clases y con la versión reducida de 3 clases (eliminando las etiquetas minoritarias). Los valores de las métricas son los siguientes:

Tabla 4.8: Comparación de resultados de fine-tuning de ALBETO con 5 clases y 3 clases

Métrica	ALBETO 5 clases	ALBETO 3 clases
Pérdida de evaluación (eval_loss)	0.429	0.396
F1 micro (eval_f1_micro)	0.556	0.715
F1 macro (eval_f1_macro)	0.314	0.409
F1 ponderado (eval_f1_weighted)	0.521	0.691
Exactitud (eval_accuracy)	0.588	0.687

De la tabla 4.8 se puede observar que, al igual que ocurrió con BETO, la reducción de clases minoritarias de cinco a tres mejora notablemente las métricas de evaluación del modelo ALBETO. La pérdida de evaluación disminuye, indicando que el modelo logra ajustarse mejor a los datos de validación. El F1 micro, que mide la calidad global de las predicciones, aumenta considerablemente, reflejando que ALBETO es capaz de identificar correctamente una mayor proporción de etiquetas activas.

Además, el F1 macro mejora, lo que señala un mejor equilibrio en el desempeño a lo largo de las diferentes clases, mientras que el F1 ponderado también crece, confirmando que el modelo maneja mejor las etiquetas con más soporte. La exactitud del modelo también experimenta un salto significativo, reforzando la idea de que el modelo generaliza mejor cuando se eliminan las clases más raras. Este comportamiento sugiere que el manejo del desequilibrio de clases es crucial para mejorar el rendimiento en tareas multilabel con ALBETO, y que simplificar la tarea puede ser beneficioso cuando las clases minoritarias tienen poca representación o relevancia práctica.

4.7 Comparación de Modelos

En esta sección se realiza una comparación exhaustiva de los distintos modelos evaluados en el problema de clasificación multilabel. Se consideran tanto modelos basados en fine-tuning de transformadores como BETO y ALBETO, como modelos tradicionales basados en representaciones BOW (Bag of Words), y modelos que emplean embeddings preentrenados sin fine-tuning. Los resultados incluyen las métricas de F1 macro, exactitud (accuracy) y F1 micro, que permiten analizar tanto el desempeño global como la precisión balanceada entre las clases y el manejo del desequilibrio.

Como se observa en la tabla 4.9, el modelo BETO entrenado con fine-tuning sobre tres clases obtiene el mejor rendimiento general, alcanzando un F1 macro de 0.566, una exactitud del 94.03% y un F1 micro del 94.12%. Esto refleja que BETO es especialmente efectivo al trabajar con un conjunto de clases reducido y balanceado, logrando un equilibrio entre sensibilidad y precisión para todas las etiquetas consideradas.

Los modelos basados en BOW con SVM y Naive Bayes también presentan un desempeño sólido, con exactitudes del 91.04% y 89.55% respectivamente, y valores de F1 macro que rondan el 0.56 y 0.55, lo que indica que técnicas tradicionales siguen siendo competitivas en algunos escenarios, especialmente cuando la capacidad computacional o el tiempo de entrenamiento son limitados. En contraste, los modelos que utilizan embeddings preentrenados sin realizar fine-tuning, como BETO y MarIA con embeddings, obtienen resultados inferiores en F1 macro y F1 micro, aunque con una precisión bastante alta. Esto sugiere que el fine-tuning es clave para adaptar los modelos de lenguaje a la tarea específica y mejorar su capacidad discriminativa.

ALBETO muestra un desempeño inferior respecto a BETO en ambos escenarios, tanto con el conjunto completo de cinco clases como con la versión de tres clases. Aunque mejora al reducir el número de clases, sus métricas se sitúan claramente por debajo de las obtenidas por BETO y los métodos basados en BOW, posiblemente debido a la menor capacidad del modelo ALBETO para capturar las complejidades del problema o a limitaciones en el ajuste fino.

Tabla 4.9: Comparación de resultados entre todos los modelos evaluados

Modelo	**Tipo**	**Macro F1**	**Accuracy**	**Micro F1**
BETO tres clases	Fine-tuned	0.5659	0.9403	0.9412
BOW - SVM	BOW	0.5616	0.9104	0.9319
BOW - Naive Bayes	BOW	0.5483	0.8955	0.9110
BETO con embeddings	Embeddings	0.5452	0.9463	0.5452
MarIA con embeddings	Embeddings	0.5389	0.9409	0.5389
BETO completo	Fine-tuned	0.5377	0.8955	0.8947
ALBETO tres clases	Fine-tuned	0.4095	0.6866	0.7151
ALBETO completo	Fine-tuned	0.3138	0.5882	0.5556

Esta comparación revela la importancia del fine-tuning específico para mejorar la capacidad de los modelos de lenguaje, destacando la superioridad de BETO en esta tarea particular de clasificación multilabel en español. Por otro lado, aunque ALBETO es un modelo eficiente, su rendimiento queda por debajo, lo que podría deberse a su arquitectura más compacta o a la necesidad de un ajuste más detallado.

Los modelos basados en BOW y clasificadores tradicionales siguen siendo opciones válidas y competitivas, especialmente cuando se considera la simplicidad y velocidad de entrenamiento, pero generalmente no alcanzan la precisión y equilibrio de los modelos transformadores fine-tuneados. Finalmente, el uso de embeddings preentrenados sin ajuste fino resulta insuficiente para obtener resultados óptimos, subrayando la relevancia de adaptar los modelos al dominio y la tarea específicos.

5 Conclusiones y vías futuras

5.1 Conclusiones generales

Este trabajo se ha centrado en la detección de trastornos de la conducta alimentaria, específicamente anorexia y bulimia, a partir del análisis automatizado de mensajes escritos por usuarios en plataformas digitales. A lo largo de este estudio, se ha desarrollado una propuesta metodológica basada en técnicas de procesamiento del lenguaje natural y aprendizaje automático, que permite abordar este problema desde una perspectiva multietiqueta. Esta aproximación no solo supera la limitación de los enfoques binarios realizados sobre este corpus con anterioridad, sino que ofrece una caracterización más matizada y realista de las manifestaciones lingüísticas asociadas a estos trastornos.

Desde el planteamiento inicial, el trabajo persiguió la construcción de un sistema capaz de distinguir entre múltiples perfiles de usuario según su relación con los TCA. Esta complejidad clínica fue adecuadamente reflejada en el diseño de la tarea como clasificación multilabel, lo cual supuso un reto tanto desde el punto de vista computacional como desde el tratamiento de los datos.

En cuanto a la experimentación, se exploraron diversos enfoques de modelado textual. En una primera fase, se implementaron modelos base clásicos como Naïve Bayes y máquinas de soporte vectorial (SVM), utilizando representaciones simples como Bag of Words. Estos modelos mostraron un rendimiento aceptable en clases mayoritarias, pero evidenciaron limitaciones sustanciales frente a categorías con menor representación, como ”suffer+against” o ”suffer+other”, cuyo bajo soporte dificultó el aprendizaje de patrones representativos. Estos resultados sirvieron como punto de referencia para evaluar enfoques más avanzados.

En una segunda fase, se aplicaron modelos de lenguaje preentrenados en español, como BETO, MarIA y ALBETO. Se abordaron dos paradigmas principales: el uso de embeddings fijos como entrada para clasificadores externos, y el fine-tuning completo de los modelos. La comparación entre ambos reveló diferencias significativas. El fine-tuning con modelos como BETO ofreció mejoras notables en métricas de evaluación globales, especialmente en F1-score macro, lo que confirma su capacidad para adaptarse de forma eficaz a las especificidades de la tarea. En contraste, los embeddings fijos, aunque más

eficientes computacionalmente, ofrecieron un rendimiento algo inferior, especialmente en la identificación de clases menos frecuentes.

BETO, en particular, demostró ser el modelo más robusto dentro del conjunto evaluado, gracias a su entrenamiento previo sobre corpus diversos en español que le permitieron adaptarse mejor al lenguaje emocional, subjetivo e informal presente en los mensajes analizados. Por su parte, MarIA, entrenado sobre textos biomédicos, mostró un rendimiento inferior, posiblemente debido a su menor exposición a registros lingüísticos informales o expresiones coloquiales.

Un aspecto metodológico relevante ha sido la adopción de técnicas de partición estratificada multilabel, que han permitido preservar la distribución original de etiquetas en los conjuntos de entrenamiento, validación y prueba. Este enfoque ha sido clave para garantizar una evaluación justa del modelo, evitando sesgos hacia clases mayoritarias. Además, se utilizaron métricas específicas para tareas multilabel, como el F1-score micro y macro, que proporcionan una visión equilibrada del rendimiento del modelo en contextos de desbalance de clases.

En conjunto, este trabajo demuestra que es posible construir sistemas automáticos capaces de identificar perfiles de riesgo en entornos digitales mediante el análisis lingüístico, y que el uso de modelos de lenguaje avanzados adaptados al español puede contribuir de manera significativa a la mejora de las herramientas de detección temprana en salud mental. Aunque todavía quedan retos por abordar, los resultados obtenidos representan un avance hacia sistemas más inteligentes, sensibles y útiles para la comunidad clínica.

5.1.1 Posibles mejoras

A pesar de los avances logrados, es importante reconocer que el sistema desarrollado presenta áreas de mejora que podrían reforzar tanto su precisión como su aplicabilidad. Uno de los principales desafíos ha sido el desbalance en la distribución de las etiquetas. Las clases minoritarias, como aquellas que representan usuarios que sufren el trastorno pero muestran una actitud crítica o ambigua hacia él, fueron especialmente difíciles de predecir. Esta escasez de ejemplos limita el aprendizaje efectivo de los modelos, lo cual podría mitigarse mediante la incorporación de nuevas técnicas de re-muestreo o generación de datos sintéticos.

Otra línea de mejora consiste en optimizar el proceso de entrenamiento mediante la introducción de funciones de pérdida ponderadas, que penalicen de manera diferencial los errores según la frecuencia de cada clase. Esta técnica permitiría que el modelo otorgue mayor atención a aquellas etiquetas con menos ejemplos disponibles. Del mismo modo, el ajuste fino del modelo podría beneficiarse de una búsqueda más exhaustiva

de hiperparámetros y del uso de técnicas como early stopping o regularización adaptativa, que pueden mejorar la generalización sin caer en sobreajuste, especialmente en dominios con alta sensibilidad social y clínica.

Finalmente, aunque los modelos de fine-tuning mostraron un mejor rendimiento, requieren una considerable carga computacional. En contextos donde los recursos son limitados, se podría explorar una mejor integración entre modelos ligeros y eficientes, y estrategias de aprendizaje por transferencia que minimicen el costo de entrenamiento sin sacrificar rendimiento.

5.2 Líneas futuras

Este trabajo abre numerosas posibilidades de investigación que pueden continuar y expandir lo aquí desarrollado. Una de las principales líneas futuras consiste en dotar al sistema de una capacidad de análisis temporal. Actualmente, los mensajes se procesan como bloques unificados por usuario, pero no se considera la evolución de su contenido a lo largo del tiempo. Incorporar una dimensión temporal permitiría analizar patrones de progresión o deterioro en el estado emocional del individuo, abriendo la puerta a modelos predictivos más finos y sensibles a cambios graduales en el discurso.

Otra línea prometedora es la integración multimodal. Aunque este trabajo se ha centrado exclusivamente en texto, las plataformas digitales incluyen otras formas de comunicación como imágenes, audio o incluso patrones de interacción (horarios de conexión, respuestas de otros usuarios, etc.). Incorporar estas señales no textuales podría enriquecer el modelo y proporcionar una representación más completa del estado del usuario.

Además, para que este tipo de tecnología sea útil en entornos reales, será fundamental su validación clínica. Esto implica una colaboración estrecha con profesionales de la salud mental, quienes pueden aportar una visión cualitativa sobre la interpretación de las predicciones, así como ayudar a definir protocolos éticos y de intervención. Un modelo, por preciso que sea, carece de sentido si no está enmarcado en una estrategia de atención responsable, segura y centrada en la persona.

Por último, una dirección especialmente relevante es el desarrollo de herramientas aplicadas para la monitorización automática y en tiempo real. La implementación de sistemas de alerta temprana que puedan integrarse en redes sociales o servicios de mensajería tendría un gran potencial preventivo, siempre que se acompañe de medidas adecuadas de privacidad, consentimiento y supervisión profesional. Esta línea de investigación no solo es técnica, sino también ética y social, y constituye uno de los retos más importantes para el futuro.

Bibliografía

Alaparthi, S. and Mishra, M. (2021). Bert: A sentiment analysis odyssey. *Journal of Marketing Analytics*, 9(2):118–126.

Arcelus, J., Mitchell, A. J., Wales, J., and Nielsen, S. (2019). Mortality rates in patients with anorexia nervosa and other eating disorders: A meta-analysis of 36 studies. *Archives of General Psychiatry*, 68(7):724–731.

Birnbaum, M. L., Ernala, S. N., Rizvi, A. F., De Choudhury, M., and Kane, J. M. (2020). A collaborative approach to identifying social media markers of schizophrenia and psychotic disorders. *Journal of Medical Internet Research*, 22(6):e16237.

Calvo, R. A., Milne, D. N., Hussain, M. S., and Christensen, H. (2017). Natural language processing in mental health applications using non-clinical texts. In *Natural Language Processing in Healthcare*, pages 11–25. Springer.

Cañete, J., Donoso, S., Bravo-Marquez, F., Carvallo, A., and Araujo, V. (2022). Albeto and distilbeto: Lightweight spanish language models. *arXiv preprint arXiv:2204.09145*.

Chancellor, S. and De Choudhury, M. (2019). Methods in predictive techniques for mental health status on social media: a critical review. *NPJ Digital Medicine*, 2(1):1–11.

Chomsky, N. (1957). *Syntactic Structures*. Mouton.

Cisternas-Osorio, R., Navarrete, A. J. L., Cabrera-Méndez, M., and Díez-Somavilla, R. (2022). Telegram para el ejercicio de la comunicación interna: Análisis de su uso en universidades hispanohablante. *Fonseca, Journal of Communication*, (25):77–93.

Coppersmith, G., Dredze, M., and Harman, C. (2014). Quantifying mental health signals in twitter. In *Proceedings of the Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, pages 51–60.

De Choudhury, M., Gamon, M., Counts, S., and Horvitz, E. (2013). Predicting depression via social media. In *Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Gutiérrez-Fandiño, A., Aguilar, C., Armengol-Estapé, J., Pàmies, M., and et al. (2022). Maria: Spanish language models. *Procesamiento del Lenguaje Natural*, 68:39–60.

Holgado-Apaza, L. A., Ancco-Calloapaza, C. L., Bedregal-Flores, O., Quispe-Layme, M., and Miranda-Castillo, R. (2023). Midiendo la carga emocional: Análisis de las emociones presentes en contenido de tweets sobre covid-19 en lima. *Revista Científica de Sistemas e Informática*, 3(2):e587–e587.

Hurtado, M. M., Rivada, Á. M., García, S. P., and Fariña, Y. R. (2024). Influencia de la pandemia por covid-19 en la incidencia de trastornos de la conducta alimentaria. In *Anales de Pediatría*, volume 101, pages 21–28. Elsevier.

Ji, S., Yu, R., Sui, Y., Li, Q., and Yu, H. (2022). Mental health analysis in social media posts with transformers. *Neurocomputing*, 469:364–373.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *European Conference on Machine Learning*, pages 137–142. Springer.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2020). Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations (ICLR)*.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Martínez-Castaño, R., López-Nores, M., García-Duque, J., and Blanco-Fernández, Y. (2021). Automatic anxiety detection using neural networks and contextualized embeddings in spanish medical forums. In *Actas del Congreso Español de Informática (CEDI)*.

Matero, M., Idnani, A., Son, Y., Giorgi, S., Vu, H. A., Schwartz, H. A., and Ungar, L. (2019). Suicide risk assessment with multi-level dual-context language and bert. In *Proceedings of the Sixth Workshop on Computational Linguistics and Clinical*

Psychology, pages 39–44.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mármol-Romero, A. M., Moreno-Muñoz, A., Plaza-del Arco, F. M., Molina-González, M. D., Martín-Valdivia, M. T., Ureña-López, L. A., and Montejo-Raéz, A. (2023). Overview of mentalriskes at iberlef 2023: Early detection of mental disorders risk in spanish. *Procesamiento del Lenguaje Natural*, 71:329–350.

Organización Mundial de la Salud (2019). Trastornos de la alimentación. Consultado el 26 de mayo de 2025.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. OpenAI Technical Report.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Rosenbrock, G., Trossero, S., and Pascal, A. (2021). Técnicas de análisis de sentimientos aplicadas a la valoración de opiniones en el lenguaje español. In *XXVII Congreso Argentino de Ciencias de la Computación (CACIC)(Modalidad virtual, 4 al 8 de octubre de 2021)*.

Ruiz-Centeno, C., Cueto-Galán, R., Pena-Andreu, J. M., and Fontalba-Navas, A. (2025). Problematic internet use and its relationship with eating disorders. *Frontiers in Public Health*, 13:1464172.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sukunesan, S., Huynh, M., Sharp, G., et al. (2021). Examining the pro-eating disorders community on twitter via the hashtag# proana: statistical modeling approach. *JMIR Mental Health*, 8(7):e24340.

Treasure, J., Stein, D., and Maguire, S. (2020). Has the time come for a staging model to map the course of eating disorders from high risk to severe enduring illness? *Early Intervention in Psychiatry*, 14(1):5–13.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, ., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Villar del Saz Bedmar, M. and Baile Ayensa, J. I. (2023). La influencia de las redes sociales como factor de riesgo en el desarrollo de la anorexia y la bulimia nerviosas durante la adolescencia. *Revista Tecnología, Ciencia y Educación*, (24):141–168.