eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Computación

Aprendizaje profundo aplicado a la detección temprana de riesgos de juego patológico

Xabier Larrayoz Vicuña

Dirección
Maite Oronoz Anchordoqui
Alicia Pérez Ramírez

19 de junio de 2023

Agradecimientos

Deseo expresar mi sincero agradecimiento a todas las personas que contribuyeron de manera significativa en la realización de este trabajo.

En primer lugar, agradezco al Gobierno Vasco por otorgarme una beca de investigación que hizo posible llevar a cabo este estudio. Su apoyo financiero ha sido fundamental para cubrir los gastos relacionados con el proyecto y permitirme dedicar mi tiempo y esfuerzo a la investigación.

También quiero agradecer al grupo de investigación IXA por proporcionarme los recursos y guía necesaria durante todo el proceso. En especial a Arantza Casillas, cuya orientación y valoraciones han sido indispensables en el desarrollo de este proyecto.

A todos ellos, mi más profundo agradecimiento por su apoyo, confianza y motivación. Sin su apoyo, este trabajo no habría sido posible.

Resumen

La salud mental juega un papel muy importante en nuestras vidas, sin embargo, todavía se percibe como un tabú en nuestra sociedad. Muchos pacientes, junto con sus familias, se ven obligados a sufrirlo en silencio, sin recibir el apoyo y la comprensión necesaria. A falta de una implicación real por parte de la sociedad, las redes sociales juegan un papel crucial. Cada vez más personas entienden las redes como espacios donde compartir sus experiencias y preocupaciones, incluyendo aquellas relacionados con la salud mental. Es importante destacar que los textos escritos en las redes sociales pueden ser indicativos de posibles problemas, muchos pacientes con trastornos mentales comienzan a dar indicios y síntomas en estas plataformas. Sin embargo, debido a la gran cantidad de datos generados a diario, resulta inviable tratar esta información utilizando los medios tradicionales. Es en este punto donde las técnicas de inteligencia artificial han avanzado lo suficiente como para tener aplicaciones en varios campos y, por tanto, tienen la capacidad y el deber de ayudar a mejorar la detección y la prevención de problemas de salud mental.

En este Trabajo de Fin de Grado se aplican una serie de técnicas de aprendizaje profundo con el objetivo de desarrollar un sistema eficiente y preciso que asista en la detección de posibles indicios de trastornos mentales, como puede ser la ludopatía. Para ello, se participará en una de las competiciones más importantes de este campo, la edición del 2023 de eRisk. Para participar en esta tarea compartida sobre detección precoz en Internet de juego patológico, se desarrollará un modelo que, aplicado a las redes sociales, sea capaz de monitorizar a un paciente e identificar las primeras señales de una posible recaída.

Índice de contenidos

Índice de figuras

Índice de tablas

CAPÍTULO 1

Introducción

1.1. Descripción del proyecto

Un trastorno mental es una afección que altera el pensamiento, el estado de ánimo y/o el comportamiento de una persona, afectando a su vida diaria. Puede ser causado por diferentes factores genéticos, biológicos, psicológicos y ambientales. Los trastornos más comunes son la ansiedad, la depresión y los trastornos bipolares. Según la Organización Mundial de la Salud en 2019, aproximadamente una de cada ocho personas en el mundo padecía un trastorno mental. A raíz de la pandemia y el correspondiente confinamiento, un porcentaje de la población ha desarrollado algún tipo de desorden, con aumentos del 26 % y el 28 % de la ansiedad y los trastornos depresivos graves respecto al año anterior [3].

La ludopatía o juego patológico es un trastorno mental que se caracteriza por la necesidad constante de jugar a juegos de azar, ignorando las consecuencias de las acciones que se realizan. Llega a afectar a individuos de cualquier ámbito, independientemente de su nivel social o situación económica. La incapacidad de controlar los propios deseos lleva a que entre las personas que optan por un tratamiento, dos de cada tres personas tengan una recaída en algún momento del mismo. La ludopatía puede ocasionar problemas graves como el endeudamiento, la pérdida de empleo además de originar otros trastornos como la depresión. La detección temprana de posibles reincidencias es fundamental para mejorar las tasas de éxito de las terapias, reduciendo la carga en los sistemas de atención médica.

El avance de la inteligencia artificial ha permitido procesar grandes cantidades de datos, y mediante el uso de algoritmos de aprendizaje profundo, es posible, por ejemplo, detectar signos tempranos de futuros trastornos mentales. Al aplicar estos avances en un modelo predictivo, se facilita el diagnóstico y el tratamiento en las primeras etapas de la enfermedad [1, 4, 5].

Es por ello por lo que en este proyecto nos centraremos en desarrollar un modelo capaz de detectar posibles recaídas de jugadores compulsivos. Monitorizaremos los mensajes de los jugadores en las redes sociales, buscando posibles comportamientos que indiquen una próxima reincidencia. Dicho de otra forma, dada una serie de mensajes de un usuario, el sistema necesita clasificar a los usuarios según los rasgos que presenten de ludopatía. Para

ello se participará en la competición llamada eRisk[1], donde los participantes desarrollan sistemas que puedan detectar problemas de salud mental en las redes sociales.

Todo el trabajo de investigación que estoy presentando surge gracias a la oportunidad que me ha brindado la beca Ikasiker (BOPV del 11/07/2022). Esta beca, otorgada por la administración del País Vasco, es una ayuda directa para estudiantes que deseen iniciar con tareas de investigación. La beca me ha proporcionado los recursos y el tiempo necesario para llevar a cabo mi proyecto de investigación, pudiendo iniciar mi carrera de investigación en las condiciones y el ámbito deseado.

1.2. Motivación

La motivación para elegir este trabajo se basa en mi deseo de querer especializarme en el área de la inteligencia artificial y el Procesamiento del Lenguaje Natural (*Natural Language Processing, NLP*), el cual es un sector en constante evolución y crecimiento. En plena revolución de los modelos del lenguaje, veo necesaria la tarea de desarrollar sistemas que aporten nuevas soluciones a los problemas de la sociedad.

Por otro lado, veo importante abordar el estigma y la discriminación asociada con los jugadores patológicos. Por eso mismo, una mejora en el proceso de detección de posibles recaídas, mejoraría las tasas de éxito de los tratamientos a la vez que aliviaría en gran medida el sufrimiento que tienen que experimentar muchas familias.

Además, trabajar con datos reales obtenidos de las redes sociales, me permite afrontar uno de los mayores retos que siguen en el procesamiento del lenguaje natural, el uso de lenguaje no correctamente escrito, muchas veces ambiguo y con frases incompletas. Este tipo de texto se caracteriza por un uso complejo del lenguaje, lo que dificulta su correcto procesamiento. En el pasado, se ha solido optar por otras fuentes de datos para entrenar los modelos. Sin embargo, en las redes sociales, los usuarios pueden expresarse tal y como son, aportando información valiosa que no se encuentra disponible en artículos o informes.

1.3. Objetivos

La finalidad de este trabajo es desarrollar un sistema que asista en la detección de trastornos de la ludopatía. Con esto en mente, se han establecido una serie de objetivos a cumplir en este trabajo.

1. Estudio del estado del arte
    a) Analizar las aproximaciones planteadas en los antecedentes
    b) Realizar una evaluación de las diferentes arquitecturas propuestas previamente en los trabajos analizados para distinguir sus fortalezas y debilidades
2. Implementación de un clasificador del riesgo del usuario a volver a caer en el juego, monitorizando su presencia en las redes sociales
    a) Ofrecer una nueva aproximación a este tipo de problemas de detección

[1] https://erisk.irlab.org/

   b) Mejorar y optimizar el sistema para que pueda tener aplicaciones en un entorno real

3. Participación en la competición eRisk 2023

   a) Realizar un estudio de los datos disponibles

   b) Evaluar la efectividad del sistema desarrollado

En el primero de los objetivos y con el fin de dar pie a los posteriores, se propone realizar un estudio exhaustivo de las ediciones pasadas de la competición eRisk. El interés es comprender los enfoques y técnicas presentadas, analizando el éxito logrado. Al estudiar estos antecedentes se busca obtener conocimientos que puedan ser aplicados en el desarrollo del modelo.

El segundo objetivo tiene el propósito de desarrollar una aproximación innovadora y original para abordar la detección de posibles recaídas en jugadores compulsivos. El estudio de los antecedentes permitirá la identificación de oportunidades de mejora en los enfoques existentes y a proponer nuevas estrategias, logrando un mejor rendimiento y precisión.

El tercer objetivo consiste en participar en la competición eRisk 2023. Esto implica desarrollar el sistema dentro de los plazos establecidos, y demostrar su validez mediante resultados precisos y confiables. Siendo necesario familiarizarse con la plataforma de experimentación, así como conectarse a otra máquina para llevar a cabo las evaluaciones. En resumen, para poder ejecutar los procesos de manera eficiente, se ha hecho uso de las máquinas de IXA mediante una conexión remota, con mayores recursos de procesamiento y memoria. Mientras que, para la fase de test, se solicitaba la comunicación constante con los servidores de la organización. Para concluir este objetivo, es necesaria la capacidad de presentar y explicar los resultados de manera clara y concisa.

Por último, se tiene la intención de establecer una base sólida para continuar con futuras investigaciones dentro del procesamiento del lenguaje natural. Siendo determinante, adoptar una correcta línea de trabajo y hábitos que permitan trabajar tanto en grupo como de manera independiente.

En cuanto a la competición eRisk, se trata de una de las competiciones que engloba Conference and Labs of the Evaluation Forum (CLEF) [2], una conferencia anual que se centra en la evaluación de sistemas de acceso a la información. Incluye diversas tareas que involucran el procesamiento de imágenes o la identificación de sexismo en redes sociales [6, 7]. Otras tareas que involucra eRisk son la búsqueda de síntomas de depresión y la medición de la gravedad de los signos de trastornos alimentarios.

La competición ofrece a los participantes la oportunidad de aplicar sus conocimientos en inteligencia artificial para abordar desafíos específicos. El estudio constante requerido fomenta la investigación y el intercambio de conocimientos dentro de un mismo campo.

El desafió que nos concierne en este trabajo se define como un problema de clasificación binaria. El sistema debe de ser capaz de monitorizar y detectar los primeros indicios de que un usuario pretende jugar de manera compulsiva tan pronto como sea posible. Estudiando la actividad de los usuarios en las redes sociales el sistema deberá etiquetar a los usuarios con tendencias de juego compulsivas. No solo se tendrá en cuenta la certeza de cada decisión,

[2] https://www.clef-initiative.eu/

sino que además el número de mensajes que se necesitan para dicha resolución se verá reflejado con las métricas de *ERDE* y *F-latency* propuestas por los organizadores (ofrecemos detalle sobre las métricas de evaluación en la sección 5.4). Como una tarea adicional, se pide a los participantes realizar un ordenamiento de los usuarios en función de su predisposición de sufrir dicho trastorno.

Previo al desarrollo del sistema, será necesario estudiar y tratar el formato de los datos proporcionados por la competición. Cada usuario del conjunto de datos tiene asociado una secuencia de mensajes que ha publicado en la plataforma social Reddit. Estos mensajes son presentados con las fechas de publicación correspondientes para realizar un trazado a lo largo del tiempo. Todos los datos requerirán de un procesamiento para poder ser utilizados en la detección de ludopatía.

CAPÍTULO 2

Planificación del proyecto

En este capítulo, se abordarán los aspectos fundamentales relacionados con la estructuración y organización de la investigación, estableciendo una planificación temporal para el trabajo a realizar. En los siguientes apartados se exponen tanto el planteamiento para realizar el trabajo, los riesgos que pueden producirse y un análisis de la dedicación de tiempo. La planificación repercutirá en la calidad del desarrollo del TFG.

2.1. Estructura de Descomposición del Trabajo

Una parte decisiva para el correcto desarrollo de este trabajo es el control de los recursos, en especial del tiempo. A diferencia de otros proyectos, es necesario cumplir los plazos establecidos por la competición, dentro del margen de 6 meses que establece la beca Ikasiker y a la vez, cumplimentando las 300 horas de trabajo que requiere un trabajo fin de grado (12 ECTS). En la estructura de descomposición del trabajo (EDT) de la figura 2.1, se puede ver la estructura jerárquica de las tareas necesarias para cumplir con los objetivos establecidos.

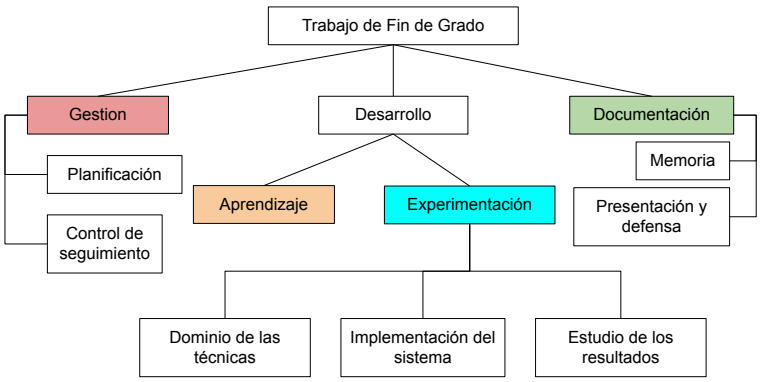

Figura 2.1: Diagrama EDT

A continuación, se describirán cada una de las agrupaciones:

1. **Gestión:** En esta categoría se hará el control del tiempo, estableciendo el espacio disponible para la implementación y la realización de las siguientes tareas de este trabajo.

   a) **Planificación:** En este apartado se realizarán todas las actividades relacionadas con la planificación, gestión y coordinación del proyecto.

   b) **Control de seguimiento:** Durante toda la vida del trabajo se realizará un control periódico del estado en el que se encuentra el proyecto, asegurando el éxito final. Incluye las reuniones para supervisar el rumbo del proyecto.

2. **Desarrollo:** Concentra la mayor carga de trabajo y responsable de generar el sistema.

   a) **Aprendizaje:** En esta fase del trabajo es necesario realizar un estudio de los antecedentes y del estado del arte existente. Los resultados de esta fase afectarán al diseño del modelo propuesto. Además, se estudiarán las técnicas y aproximaciones requeridas para afrontar la experimentación.

   b) **Experimentación:** Periodo más largo del proyecto, incluye tanto el diseño, la implementación y la evaluación de los sistemas candidatos.

      1) **Dominio de las técnicas:** Un ciclo previo a la implementación del sistema es necesaria una total comprensión y manejo de las técnicas de aprendizaje profundo. Es necesario poder replicar los resultados tanto de los antecedentes como de las técnicas con mejores resultados en el momento actual
      2) **Implementación del sistema:** En esta etapa se implementarán una serie de variaciones de la arquitectura seleccionada, de donde saldrá el modelo final.
      3) **Estudio de los resultados:** Terminada la fase de creación de los modelos, es necesario determinar el rendimiento y robustez de los modelos desarrollados. Para ello se recrearán las condiciones de la edición 2022 de la competición y se realizará una evaluación con los datos correspondientes de aquella edición.

3. **Documentación:** Este paquete de trabajo abarca tanto la creación de una memoria como la presentación y defensa del trabajo.

   a) **Memoria:** Tanto el seguimiento como los resultados generados serán registrados en un documento que describirá la arquitectura utilizada y plasmará todo el conocimiento adquirido a lo largo de este trabajo.

   b) **Presentación y defensa:** Finalmente se realizará una presentación como recurso complementario para la defensa oral, detallando las diferentes partes de este trabajo y se dará respuesta a las preguntas y dudas que le surjan al tribunal correspondiente.

2.2. Riesgos y medidas preventivas

En el desarrollo del proyecto se identifican algunos riesgos y se proponen medidas para prevenir su aparición.

Uno de los riesgos más significativos de este trabajo sería el incumplimiento de los plazos previstos, provocando la descalificación inmediata de la competición. Para evitar esto, se ha establecido una dedicación específica para cada sección, incluyendo un margen de error para subsanar cualquier imprevisto.

Por otro lado, la incapacidad de desarrollar un modelo eficiente, pese a no acarrear problemas relacionados con la competición, supone el incumplimiento de varios de los objetivos propuestos para este trabajo. Para evitar este desenlace, se ha prestado especial interés al estudio de los antecedentes y el estado del arte.

Finalmente, para afrontar la posible pérdida parcial o total del trabajo realizado, se realizarán copias de seguridad locales de forma periódica, a la vez que se hacen uso de plataformas como GitHub u Overleaf.

2.3. Planificación temporal

En esta sección se establece la planificación estimada para realizar las tareas anteriores. En la tabla 2.1 se presentan las tareas expuestas en el EDT, junto al periodo de tiempo en el que se realizarán y la dedicación prevista para realizarlas.

Fase	Inicio	Fin	Dedicación (h)
Gestión	01/01/2023	20/05/2023	15
Planificación	01/01/2023	10/01/2023	10
Control de seguimiento	01/01/2023	20/05/2023	5
Desarrollo	10/01/2023	10/04/2023	185
Aprendizaje	10/01/2023	10/02/2023	30
Experimentación	20/01/2023	10/04/2023	155
Dominio de las técnicas	20/01/2023	05/02/2023	15
Implementación del sistema	05/02/2023	05/04/2023	120
Estudio de los resultados	20/03/2023	10/04/2023	20
Documentación	20/02/2023	20/06/2023	100
Memoria	20/02/2023	20/05/2023	80
Presentación y defensa	20/05/2023	20/06/2023	20
Total	01/01/2023	20/06/2023	300

Tabla 2.1: Planificación temporal

En la figura 2.2 se detalla un diagrama sobre la planificación y gestión del proyecto. Este diagrama proporciona una visión clara y estructurada, de las tareas y los hitos a lo largo del tiempo. Ha permitido visualizar las dependencias entre las distintas etapas del proyecto, identificando posibles retrasos, además de establecer una secuencia lógica de actividades. Adicionalmente, se registra tanto los días en los que se realizarán las reuniones, como la fecha final de la competición.

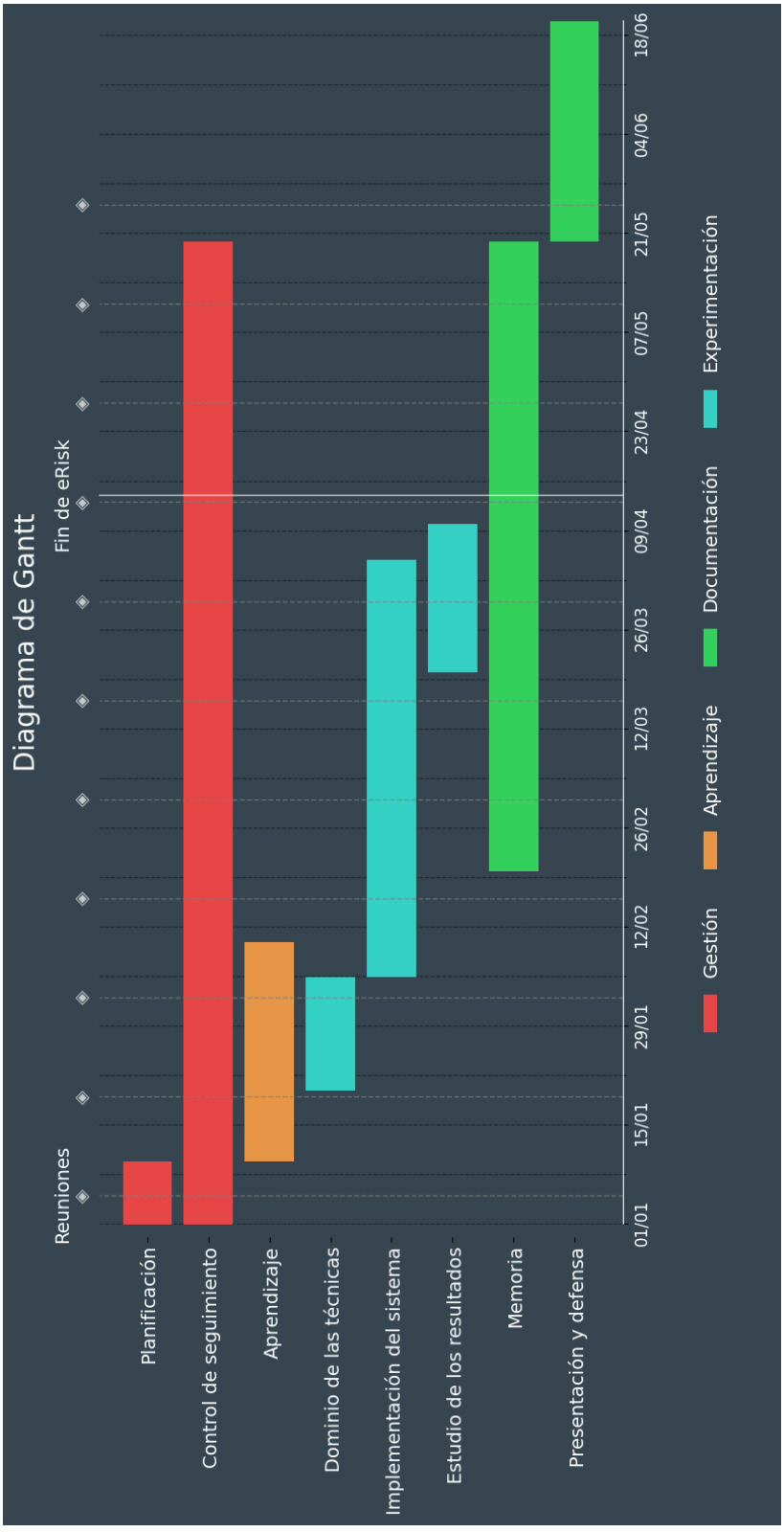

Figura 2.2: Diagrama de Gantt

CAPÍTULO 3

Antecedentes

En los últimos años, el uso de la IA para abordar tareas relacionadas con la salud mental ha obtenido mayor presencia. La IA, y en particular el NLP, ha demostrado ser una herramienta poderosa en la detección de signos de trastornos mentales. En estudios previos, se ha utilizado NLP sobre informes médicos electrónicos para asistir en la identificación de conductas suicidas [8, 9], logrando una precisión de 0,47.

La figura 3.1 muestra la tendencia creciente de la investigación en detección de enfermedades mentales, enfatizando la preferencia a enfoques basados en Aprendizaje profundo a partir de 2020.

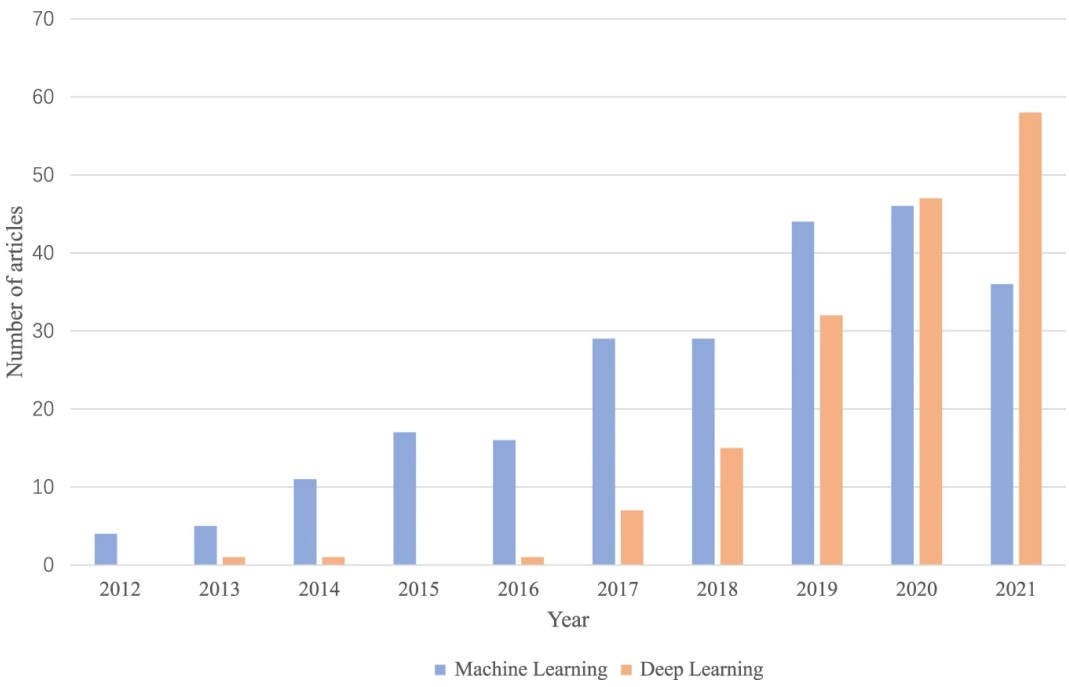

Figura 3.1: La tendencia de la cantidad de artículos que contienen métodos basados en aprendizaje automático y aprendizaje profundo para detectar enfermedades mentales de 2012 a 2021 [1]

En este contexto, el 59 % de los métodos utilizados para la detección de enfermedades

mentales están basados en técnicas tradicionales de Machine Learning, como Support Vector Machine (SVM) [10], AdaBoost [11] o Decision Trees [12] , a pesar del reciente interés en el Aprendizaje profundo que ha mostrado un mejor rendimiento [1]. Sin embargo, tal y como refleja la figura 3.2, gran parte de las soluciones planteadas se concentran en unos pocos trastornos mentales. Destaca la ausencia de estudios en la detección de la adicción al juego pese a ser una de las enfermedades más comunes en todo el mundo.

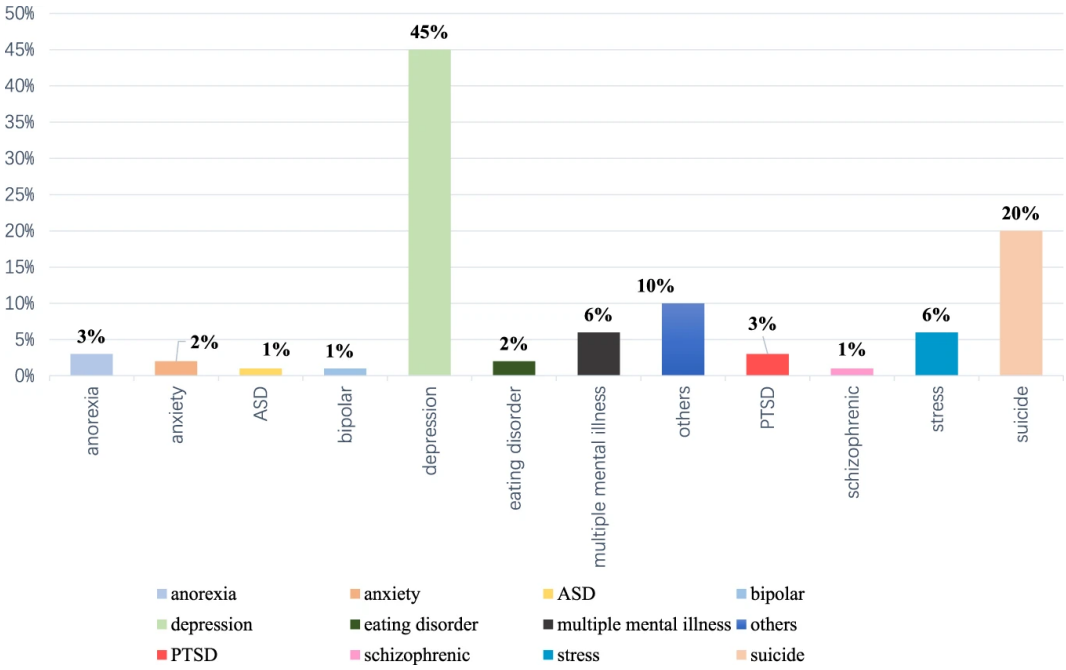

Figura 3.2: Distribución de los trabajos de detección de trastornos mentales según el tipo de trastorno [1]

Son muy escasos los trabajos que implementan un sistema que trate la evolución del trastorno del paciente a lo largo del tiempo. Prácticamente la totalidad de los datasets disponibles presenta un formato con entradas independientes, impidiendo tratar una serie de entradas de un mismo usuario.

Las competiciones se han convertido en una forma popular de demostrar los avances en el campo de la inteligencia artificial. Los investigadores pueden poner a prueba sus modelos en una plataforma estandarizada, compartiendo sus descubrimientos y comparando estos resultados con los de ediciones anteriores. En el campo del procesamiento de imágenes, destaca ImageNet [13], una competición anual de clasificación de imágenes. Algunos de los ganadores de ediciones pasadas, como AlexNet [14], Inception [15], o ResNet [16], provocaron grandes avances. Estas competiciones no solo impulsan el desarrollo y la mejora de nuevas técnicas de IA, sino que también establecen el estado del arte para dichas tareas.

Las competiciones Social Media Mining for Health (SMM4H) [17], Conference and Labs of the Evaluation Forum (CLEF) y CLPsych [18] se enfocan en la aplicación de técnicas de aprendizaje automático en el ámbito de la salud mental. En 2022, el ganador de la competición CLPsych obtuvo una precisión del 68,9 % en la detección de cambios de humor en tweets. Años antes, en 2019, se registró una tasa parecida en la identificación en base a sus mensajes de usuarios en riesgo de suicidio.

Durante varios años una de las tareas a abordar en el CLEF eRisk ha consistido en la detección temprana de riesgo de ludopatía. Procesando secuencialmente las interacciones de los usuarios en las redes sociales, el sistema debía detectar los primeros indicios de ludopatía lo antes posible.

En la edición anterior se realizaron enfoques muy diferentes. El grupo SINAI [19] planteó un diseño basado en las características del lenguaje. Utilizando los últimos 50 mensajes del usuario, se obtenía un vector el cual se complementaba con características de los mensajes, como el número de palabras, la diversidad léxica y la complejidad de las oraciones. Finalmente se pasaba por un modelo de red neuronal alimentada hacia adelante (FFNN). El grupo BLUE [20], propuso entrenar un clasificador BERT, empleando un conjunto de datos adicional generado de algunas comunidades sobre salud mental de Reddit. El equipo UNED-NLP [21], participó con un sistema que se basaba en técnicas de *Approximate Nearest Neighbors* (ANN) [22] para detectar mensajes positivos.

CAPÍTULO 4

Datos

4.1. Descripción cualitativa

El dataset provisto en esta edición eRisk 2023, en la tarea "Task 2: Early Detection of Signs of Pathological Gambling", está compuesto por la unión de los datos utilizados en las dos ediciones anteriores (eRisk 2021 y 2022). El contenido consta de un conjunto de ficheros eXtended Markup Language (XML), donde cada uno de ellos contiene una serie de publicaciones escritas por un usuario en una red social. Adicionalmente se acompañan los datos con un fichero que vincula a cada usuario con una etiqueta. Esta etiqueta, que toma el valor 1 o 0, asigna a los usuarios como ludópatas o de control respectivamente. En la Tabla 4.1, se muestra como el etiquetado se asigna a nivel de usuario, y no existe una relación directa en función del mensaje publicado.

subject1090	**1**	**subject1317**	**0**
Which site is that man? 22.5k per month max is fucking cheap. Is this sportsinteraction?		For a long time I thought I would leave and never come back and then I got to this point where…	
Fuck me. Already lost half my paycheck in 1 hr,got paid today and already fucking lost half of it on blackjack. Like what the fuck is wrong with me?? I hate this fucking addiction.		I guess in a certain way letting it go would be like abandoning that former self of me that was scared to death of what was happening…	
I want to stop. But I can't stop.		Yes the friends thing is an interesting distinction. For me I was actually recently married and…	

Tabla 4.1: Muestra de ejemplo del etiquetado ofrecido para cada usuario

La tabla 4.2 es una muestra del tipo de contenido que podemos encontrar para un usuario. Por cada mensaje se guarda tanto la fecha, el título y el cuerpo del mensaje publicado en Reddit. Los usuarios aprovechan estas plataformas para expresar sus preocupaciones y rabia contenida, sin las limitaciones que a menudo se llegan a dar en consultas o grupos de ayuda.

A fin de tener una ligera idea global del contenido de los mensajes, se muestran las nubes de palabras de los mensajes positivos (clase 1) y negativos (clase 0) respectivamente en las figuras 4.1 y 4.2. Estas figuras se han extraído automáticamente mediante las bibliotecas Gensim [23] y Wordcloud [24]. Se realizó un preprocesamiento de los textos, que incluye la eliminación de palabras vacías (*stopwords*). Posteriormente, separando los textos en función

Título	Fecha	Texto
Having no money; worst part of recovering	2018-02-20 19:26:16	I am now 30 days clean but effectively penniless, I am currently -3k in the red. Always thinking twice before spending money. It sucks
I actually dont miss it	2018-02-21 14:33:21	It has got to a point where when I have the urge to gamble I just shrug my shoulders and cant be bothered to do it. Too much hassle.
I hate being in debt	2018-05-28 15:24:57	Feels chrostophobic Anxiety argh
Anxiety from being in debt	2018-06-19 08:25:03	Does anyone else feel this way, strong levels of anxiety after being in debt?

Tabla 4.2: Muestra del contenido del conjunto de datos con las publicaciones de Reddit de un usuario ludópata

de la edición y la pertenencia de clase del usuario se han generado cuatro modelos. Estos modelos asignan un peso a cada palabra en función de su frecuencia y relevancia en el corpus. Finalmente, se han extraído las palabras más relevantes para cada grupo y se han visualizado con la biblioteca Wordcloud.

Analizando los términos más relevantes que se presentan en los textos, podemos observar la complejidad detrás de esta tarea. Tanto los jugadores como los usuarios de control presentan un lenguaje similar en estas plataformas. A pesar de pertenecer a categorías distintas, el uso de las mismas palabras clave en contextos similares dificulta la tarea de diferenciarlos. Un desafío tanto para los modelos entrenados con reportes y artículos médicos que presentan un lenguaje más formal, como para los profesionales de la salud.

(a) Positivos 2021 **(b)** Positivos 2022

Figura 4.1: Nubes de palabras de los usuarios de clase 1 (con tendencia al juego compulsivo)

(a) Negativos 2021 **(b)** Negativos 2022

Figura 4.2: Nubes de palabras de los usuarios de clase 0 (i.e. con ausencia de tendencias de juego compulsivo)

Dado el contenido sensible que conlleva el conjunto de datos proporcionado, la organización ha aplicado una serie de medidas para garantizar la privacidad de los usuarios, muy habituales en estos casos. En primer lugar, los nombres de usuario han sido debidamente

alterados para mantener el anonimato. Asimismo, el contenido de cada mensaje ha sido analizado para borrar cualquier información personal que puede llevar a la identificación del usuario.

4.2. Descripción cuantitativa

Una vez ofrecida la descripción cualitativa de los datos, mediante la tabla 4.3 se muestra una descripción cuantitativa del conjunto de datos disponible para el entrenamiento, así como el conjunto empleado para la evaluación por parte de la organización. La partición empleada en el desarrollo del modelo se menciona en la sección 6.2. El conjunto de datos 2023 disponible para los participantes (que no es más que la unión de los conjuntos 2021 y 2022) consta de un total de 4.427 usuarios con un total de 2.298.412 mensajes en total. Si llamo $|\Sigma_{2021}| = 588 \times 10^3$ al vocabulario disponible en 2021, y $|\Sigma_{2022}| = 748 \times 10^3$ al de 2022. El vocabulario total es $|\Sigma| = |\Sigma_{2021} \bigcup \Sigma_{2022}| = 1,15 \times 10^6$. Por otra parte, el conjunto de evaluación, utilizado para comparar las soluciones de los participantes, ha consistido en una nueva colección de usuarios. Una vez que los sistemas se comunicaban con el servidor, iban obteniendo secuencialmente los mensajes para realizar la evaluación. Este conjunto consta de un total de 2.174 usuarios con un total de 1.324.507 mensajes en total. El nuevo vocabulario alberga 372.905 palabras que no se habían visto en el conjunto de entrenamiento disponible, $|\Sigma|$, o lo que llamamos palabras *Out of Vocabulary (OOV)*.

	Disponible		**Evaluación**
	2021	**2022**	**2023**
Usuarios totales	2.348	2.079	2.174
Ludópatas (1)	164	81	-
Control (0)	2.184	1.998	-
Post totales	$1{,}13 \times 10^6$	$1{,}16 \times 10^6$	$1{,}32 \times 10^6$
Post/usuario (Avg $\pm$ Stdev)	481 ± 521	561 ± 573	609 ± 579
Palabras totales	33×10^6	39×10^6	41×10^6
Palabras/post (Avg $\pm$ Stdev)	37 ± 141	42 ± 121	36 ± 118
|Vocabulario|	588×10^3	748×10^3	10^6
OOV	-	-	370×10^3

Tabla 4.3: Descripción cuantitativa de los conjuntos de datos empleados: eRisk 2021, 2022 y 2023. Describimos la distribución de usuarios por clase; el número de textos (posts) por usuario (user); la longitud de los textos medida en número de palabras (total y por mensaje); el tamaño del vocabulario y las palabras fuera del vocabulario (OOV) en el test

Describimos, a continuación, el contenido de esta tabla de modo más gráfico y mostraremos, así, la dispersión de los datos disponibles desde distintas perspectivas.

Como se puede apreciar en la figura 4.3, ambas ediciones presentaron un conjunto de datos desbalanceado, es decir, la distribución de la clase está lejos de ser uniforme. Siendo el caso donde los jugadores clasificados como ludópatas no superan el 7 % o el 4 % en sus respectivos conjuntos. Los antecedentes en clasificación supervisada destacan el desbalanceo de clases (*class skew*, o *class imbalance*) como uno de los factores que dificultan notablemente la capacidad de los algoritmos de inferencia para aprender y generalizar correctamente la clase minoritaria [25].

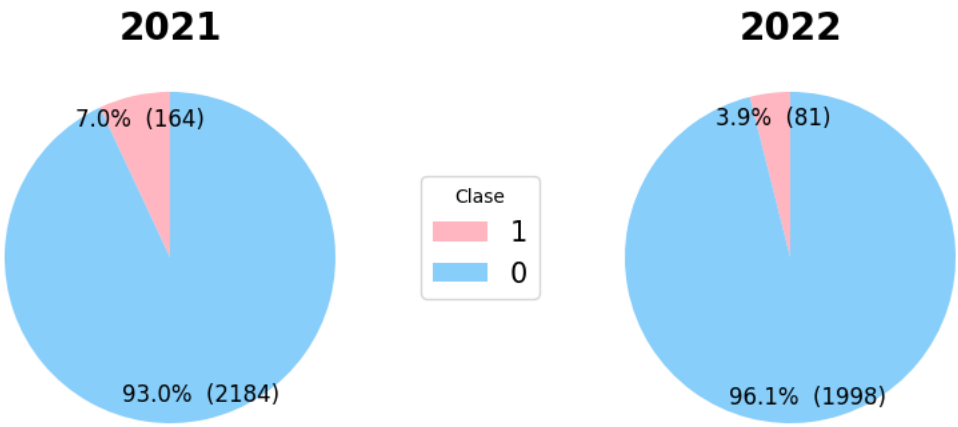

Figura 4.3: Distribución de clases de los usuarios de las ediciones 2021 y 2022

La desviación estándar de los mensajes por usuario es bastante grande y esto nos motivó a estudiar más en detalle este aspecto. Así pues, la figura 4.4 representa la distribución de mensajes disponibles por usuario e indica los diferentes cuartiles que se obtienen para cada edición. Asimismo, revela una concentración importante en torno a 250 mensajes, pero con una serie de candidatos con valores atípicos. Se observó que muchos de estos mensajes se reducen a simples interacciones con otros usuarios, como puede verse en los ejemplos de la Tabla 4.4.

Boi	Bruh
u/uwutranslator	Bro true
Yo tf?	Yes, yes it does
Uhhhhh...	GG bro GG
Repost	Thanks

Tabla 4.4: Ejemplos de interacciones triviales por parte de los usuarios

Por consiguiente, al quedarse reducido en simples respuestas o discusiones triviales, no se reflejan las experiencias y emociones relacionadas con el juego que pueden ser de gran interés. A la vista de estas distribuciones optamos por restringir el entrenamiento del modelo a usuarios con al menos 10 mensajes y excluimos los 20 usuarios con mayor presencia en el dataset. Esto último lo hicimos con el objetivo de mejorar la calidad de los mensajes, puesto que ese subgrupo de usuarios solo aportaba ruido y reduciría la eficacia del entrenamiento. Este proceso se profundizará en la sección 5.2.1.

En la tabla 4.3 también intuimos que hay variedad en la longitud de los mensajes de texto. Es por esto que estudiamos la distribución de palabras por mensaje de los textos en detalle en la figura 4.5. Como se puede ver, en ambas ediciones únicamente un tercio de los mensajes está compuesto por más de 30 palabras, una práctica muy habitual dentro de las redes sociales. Dada la gran cantidad de mensajes formados con menos de 5 palabras, se estudió la posibilidad de descartar este subgrupo, no obstante, después de un análisis más profundo, un importante número de sujetos periódicamente compartían el número de días sin apostar, acompañado de un breve resumen de su estado mental, lo que justificó su permanencia, se presenta una muestra en la Tabla 4.5.

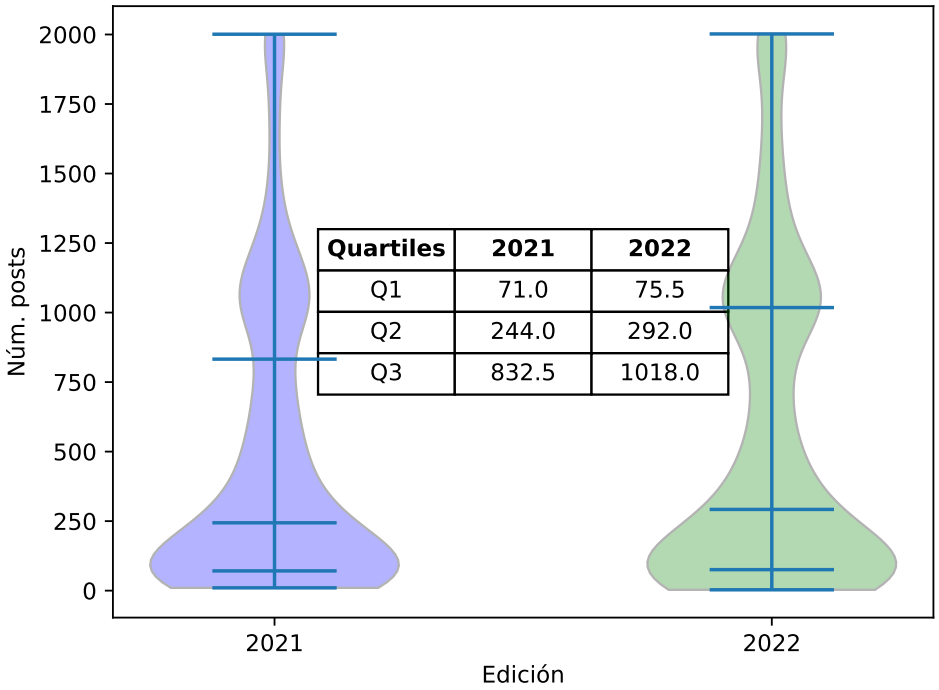

Figura 4.4: Distribución de mensajes por usuario

day 18: still suffering from anxiety.
day 21 - no gambling urges, but life is really hard.
day 1: hopefully never again a day 1 - i lost my last $ of savings.
day 236 - time flies, cant believe this number is so high.
back to day 0 - i hate to say this but im back to day 0.
day 96 update - i am feeling good about life.

Tabla 4.5: Ejemplos de registros por parte de los usuarios

A continuación, se presenta un análisis a posterior de los datos de la competición eRisk 2023. En la tabla 4.3 se intuye el parecido respecto a los datos disponibles, tanto en el número de usuarios como en el resto de las categorías, presentando unos valores dentro de lo esperado. Las gráficas 4.7 y 4.6 realizan una comparación respecto la edición de 2021 de las distribuciones en el número de mensajes y el número de palabras. En general, los usuarios de la edición de 2023, muestran una mayor interacción en las redes. Sin embargo, a grandes rasgos, la edición disponible de 2021 es un buen representativo del conjunto de test. La longitud de los mensajes es similar en ambas ediciones, vemos que la mitad de los mensajes se encuentran por debajo de las 15 palabras y dos tercios por debajo de 20. Tal y como hemos adelantado, los usuarios pertenecientes a la edición de 2023 han publicado más mensajes que sus predecesores, mientras que en 2021 la mediana se determinó en 244 mensajes, en 2023 alcanza los 407 mensajes.

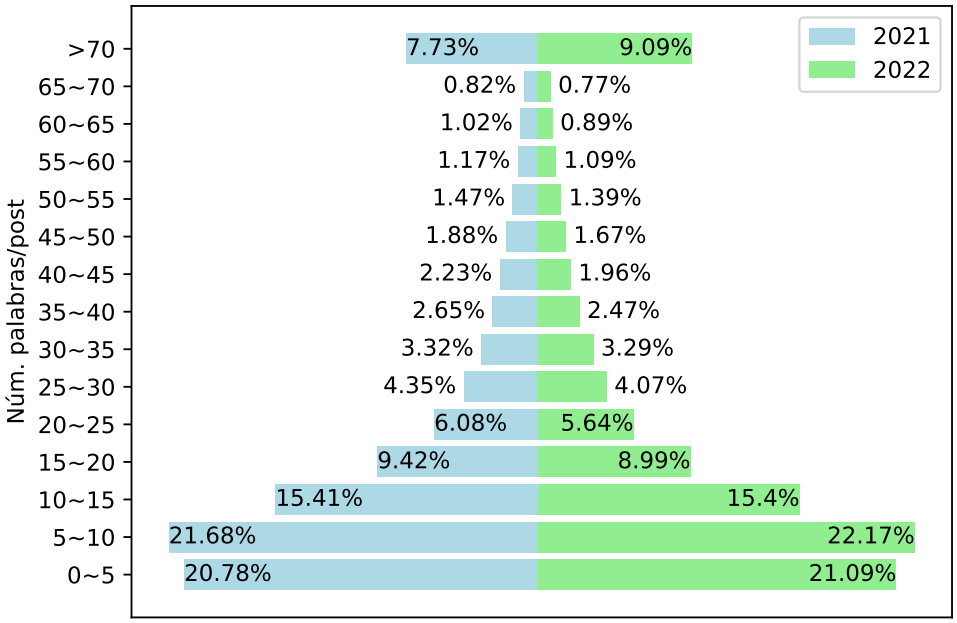

Figura 4.5: Distribución del número de palabras por mensaje

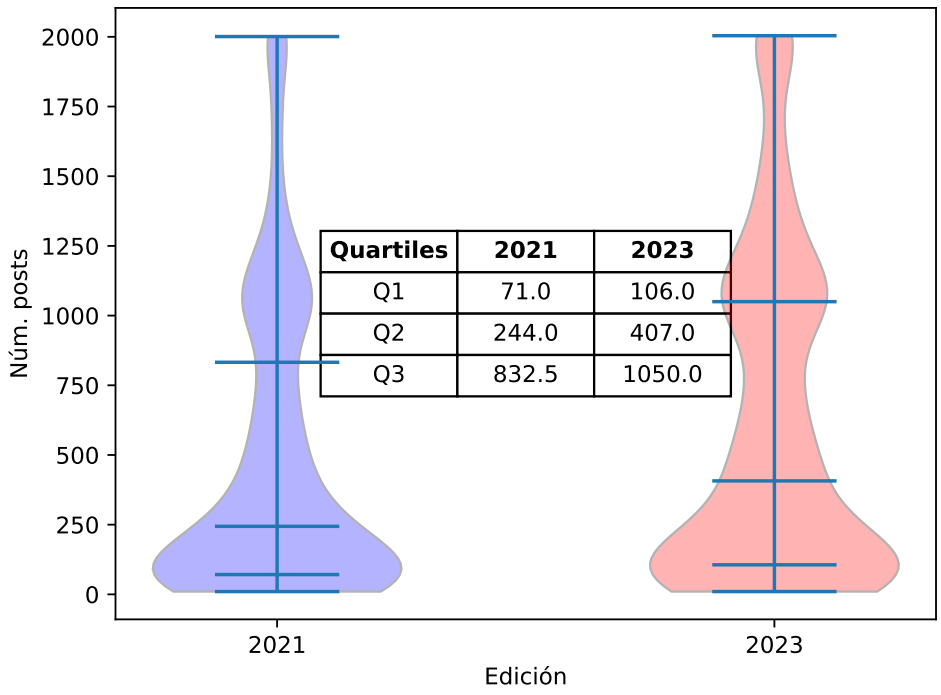

Figura 4.6: Distribución de mensajes por usuario

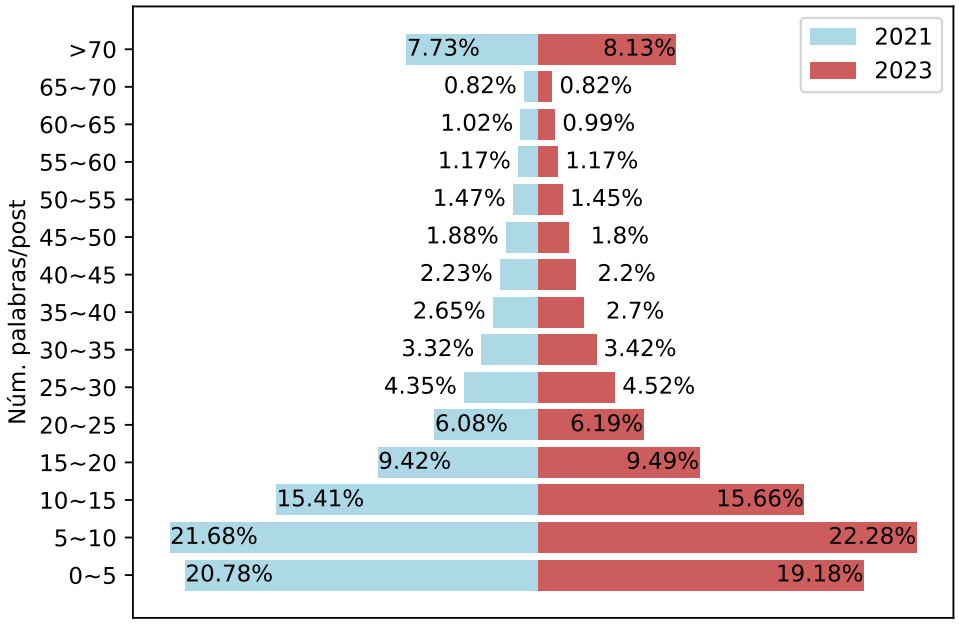

Figura 4.7: Comparativa entre las ediciones 2021 y 2023 en el número de palabras por mensaje

CAPÍTULO 5

Marco metodológico

La tarea se define como un problema de clasificación binaria. El sistema debe de ser capaz de monitorizar y detectar los primeros indicios de un jugador compulsivo tan pronto como sea posible. Estudiando la actividad de los usuarios en las redes sociales, el sistema deberá etiquetar a los usuarios con tendencias de juego compulsivo. No solo se tendrá en cuenta la certeza de cada decisión, el número de mensajes que se han necesitado para dicha decisión se verá reflejado con las métricas de *ERDE* y *F-latency* propuestas por los organizadores (ofrecemos detalle sobre las métricas de evaluación en la sección 5.4). Como una tarea adicional, se pide a los participantes realizar un ordenamiento de los usuarios en función de su predisposición a sufrir dicho trastorno y el nivel de gravedad. A falta de muestras para realizar un entrenamiento, se ha optado por utilizar la probabilidad de sufrir dicho trastorno como un indicador de la gravedad para realizar la ordenación (se profundizará en detalle en la sección 5.2.6).

En base a las fortalezas y debilidades estudiadas en los antecedentes , capítulo 3, en este apartado describimos la aproximación por la que nos decantamos. En la sección 5.2 describimos la arquitectura propuesta y a continuación, sección 5.3, remarcamos algunos parámetros a los que puede ser sensible esta arquitectura y que debemos tener en cuenta en la fase experimental. Para finalizar, en la sección 5.4 describimos las métricas empleadas para evaluar la calidad de los modelos en este contexto.

5.1. Conceptos previos

Para poder comprender las decisiones próximas es necesario presentar una sección donde se describirán algunos conceptos claves que ubican este trabajo.

5.1.1. Entrenamiento de redes neuronales

A la hora de ajustar los parámetros de una red neuronal, existen varios tipos de entrenamiento. En función de los datos que tengamos estaremos ante un caso u otro. En el entrenamiento supervisado, se utiliza un conjunto de datos etiquetados para entrenar el modelo, mientras que el entrenamiento no supervisado no requiere etiquetas para los datos.

Por otro lado, el entrenamiento semi-supervisado utiliza una combinación de los dos casos anteriores. La elección de un caso frente a otro dependerá de la tarea que se esté abordando.

Independientemente del tipo de entrenamiento, existen diversas técnicas que se pueden agregar. La base del *transfer learning* radica en partir de un modelo entrenado y adaptarlo a la tarea en la que nos encontremos. En el *ensambling* se combina múltiples modelos para mejorar el rendimiento de la red. Modificar la función de pérdida durante el entrenamiento puede ayudar a evitar el sobreajuste, mientras que, añadiendo nuevas funciones de pérdida, se puede mejorar el rendimiento del modelo en varias tareas a la vez [26].

5.1.2. Transformer-based Pretrained Language Models

Los Transformer-based Pretrained Language Models (T-PTLMs) [26] son modelos de aprendizaje profundo que han sido entrenados previamente en grandes cantidades de datos de texto. Desde la presentación de *Bidirectional Encoder Representations from Transformers (BERT)* [27], estos modelos han logrado nuevos estados del arte en diversas tareas del procesamiento del lenguaje natural, incluyendo la clasificación de texto y el análisis de sentimiento. Utilizando la arquitectura de los transformers, generan representaciones de alta calidad del lenguaje natural, que posteriormente se emplean como entrada para modelos de clasificación de texto, mejorando así su rendimiento.

5.1.3. Representación del lenguaje

Emplear una correcta representación de los datos de entrada determinará la eficacia que podrá lograr el modelo para abstraer la información relevante. En función del dominio de la tarea se necesitará un tipo de representación especifica. En el caso de las redes sociales, se presenta un reto mayor debido al uso de un lenguaje informal y variado, lo que complica el proceso de una efectiva representación.

Tanto en este como en anteriores trabajos, es necesaria una total comprensión de los datos de entrada. Tanto el contexto como la forma de expresión de cada mensaje influye en la calidad de las representaciones generadas por los modelos. Para entornos propios de una red social, es fundamental una comprensión general del lenguaje, pese a perder el vocabulario especializado que podría tener un campo específico, como el de la salud. Por ejemplo, en algunos trabajos se utilizaban coloquialismos y emoticonos, los cuales influían en la calidad de la representación de los datos. En este sentido, un tratamiento de procesamiento del lenguaje resultó esencial para el éxito del modelo [26].

5.1.4. Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) es un modelo probabilístico utilizado para analizar y clasificar grandes conjuntos de datos de texto. Partiendo de los patrones de la entrada, es capaz de identificar los temas latentes o más significativos. En muchos trabajos se ha utilizado como método para reducir la dimensionalidad de la entrada. Sin embargo, en este trabajo, se han utilizado las distribuciones que genera como características adicionales en la red neuronal. Todo esto con la intención de aprender patrones más complejos en los textos.

Como se muestra en la figura 5.1, se asume la existencia de un número determinado de temas, cada uno de los cuales es una distribución de probabilidad sobre las palabras del vocabulario. A su vez, un documento se representa por la presencia de ciertos temas, espe-

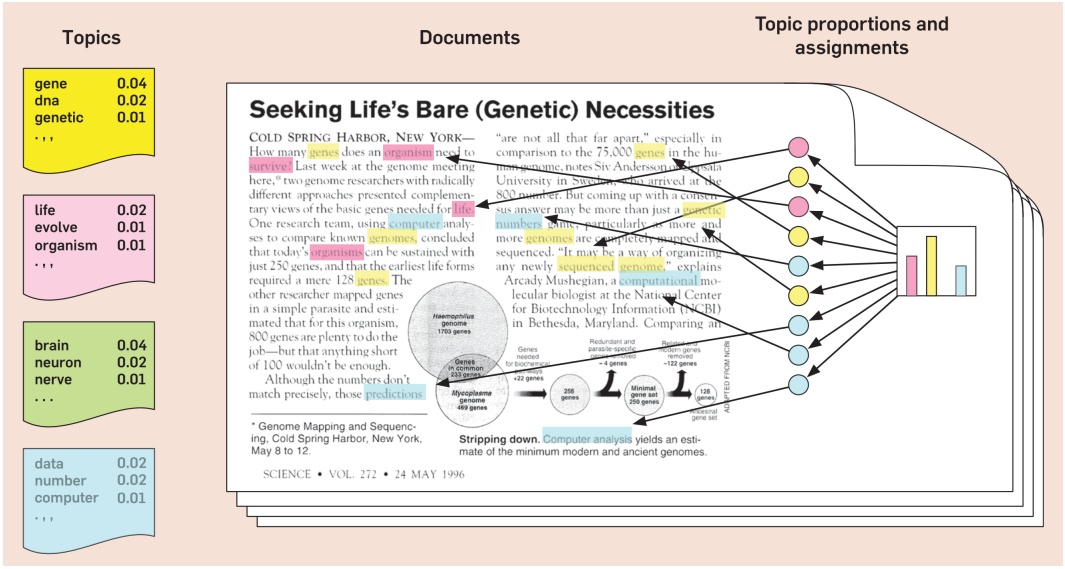

Figura 5.1: Ejemplo para ilustrar Latent Dirichlet Allocation [2]: cada documento se representa como una distribución de tópicos (histograma derecho) y cada tópico se representa como una distribución sobre el vocabulario (a la izquierda con un color para cada tópico)

cíficamente, como una distribución de tópicos. Estas representaciones de los documentos (el vector de la presencia de cada tópico en el documento) se utilizan posteriormente en modelos de clasificación o clustering [2]. Cada tópico se representa como una distribución sobre las palabras. A la izquierda de la figura se muestran las palabras de cada tópico, junto con la probabilidad o presencia de la palabra en el tópico. Por ejemplo, el tópico amarillo de la izquierda trata de genética, mientras que el tópico azul trata de ciencias de datos. Cada documento se representa como una distribución de la presencia de los tópicos en el documento actual. En la figura se representa como un histograma de colores a la derecha del documento. LDA define los tópicos latentes en la colección de documentos de modo que mejor se discriminen los documentos entre sí en base a la distribución de tópicos.

5.2. Arquitectura del modelo

Las tendencias actuales del estado del arte sugieren que las redes neuronales tienen la capacidad de afrontar el desafío descrito. Su capacidad de procesamiento las vuelve idóneas para trabajar con grandes cantidades de datos. Sin embargo, se requiere de una entrada de tipo numérico para poder trabajar con ellas. En el caso de nuestro problema, el input está en formato de texto, lo que requiere de una conversión previa para poder ser utilizado. El diseño del sistema propuesto consiste en una primera fase de procesamiento del texto, donde se obtiene una representación vectorial por cada post, para finalmente acabar en una red neuronal.

La información que viene en texto plano queda convertida en un vector numérico que puede ser utilizado como entrada de una red neuronal. Una representación adecuada permite capturar complejas características del texto, como el significado o el sentimiento expresado, necesarias en tareas como clasificación o generación de texto, donde se necesita un completo entendimiento del lenguaje.

Durante muchos años, se han utilizado representaciones a nivel de palabra. Métodos como Word2Vec [28], GloVe [29] o FastText [30] tuvieron mucho éxito en tareas como la detección de similitud entre palabras o el análisis de sentimiento. Sin embargo, la necesidad de trabajar con un contexto más global del texto en algunas tareas terminó en el desarrollo de nuevas soluciones.

Modelos como el Universal Sentence Encoder (USE) [31], Sentence-BERT (SBERT) [32] y Transformer-based Pretrained Language Models (T-PTLMs), permiten generar representaciones de oraciones. De esta forma, se obtiene una representación mucho más compleja y global al tener en cuenta las interacciones y relaciones de las palabras. Esto es necesario en tareas como la generación y clasificación de texto.

SBERT es una variación del modelo preentrenado BERT que utiliza estructuras de redes siamesas y de tripletas para generar las representaciones. Dichas estructuras permiten aprender la similitud y las diferencias entre diferentes entradas. Por otra parte, USE utiliza una arquitectura basada en redes neuronales convolucionales y recurrentes para generar los vectores. La variante Dynamic Aggregation of Network (DAN) utiliza la técnica de agregación dinámica de redes, para mejorar el resultado. En lugar de tratar cada nodo de una red de manera aislada, permite que los nodos intercambien información para mejorar el rendimiento.

En este trabajo hemos optado por probar tanto DAN como SBERT para generar las representaciones. La versión de DAN genera un vector de tamaño 512, mientras que SBERT trabaja con un vector de dimensión 384. El uso de un vector de características de mayor tamaño, como el que ofrece DAN, puede proporcionar una mayor cantidad de información. Sin embargo, las diferencias en el desarrollo de ambos modelos determinarán cuál se ajuste mejor a los datos ofrecidos y al tamaño de la capa de entrada de la red final.

Dado una secuencia de l posts consecutivos escritos por el usuario k, denotado como $\mathbf{t}_k = (t_k^1, t_k^2, \ldots, t_k^l)$ con t_k^j siendo el j-ésimo post de la sucesión, el objetivo es obtener una etiqueta de clasificación a nivel de usuario ($\hat{u}_k$) para distinguir a los Jugadores Patológicos de los usuarios de Control. Para ello, en nuestro enfoque se procesa cada post (t_k^j) y se calcula una etiqueta a nivel de post (c_k^j) mediante la arquitectura presentada en la figura 5.2. Con esta información se obtiene, a continuación, la etiqueta a nivel de usuario.

Los procesos implicados y las estrategias de entrenamiento aplicadas se detallan en las siguientes secciones.

5.2.1. Pre-procesamiento

Como preparación previa inicial y dado el origen de los datos, cada publicación de un usuario constaba de un título y un cuerpo. El enfoque propuesto fue la concatenación de ambas partes para formar un único mensaje, solucionando situaciones donde una de las dos partes estaba vacía. Una vez realizado esto y con el objetivo de poder utilizarlo de manera eficiente, se realizaron una serie de preprocesamientos en cada mensaje. Inicialmente, se realizó una conversión a minúsculas y una limpieza de caracteres innecesarios, tales como signos de puntuación. Posteriormente, con ayuda de la librería NLTK (Natural Language Toolkit) [33], que proporciona una lista de palabras comunes conocida como "stop words", se eliminaron las palabras que no aportan significado en el análisis de texto. Una vez terminado esta primera fase, se habría reducido el nivel de ruido de una manera importante para poder ser utilizado en la generación de representaciones semánticas. La limpieza toma un mayor

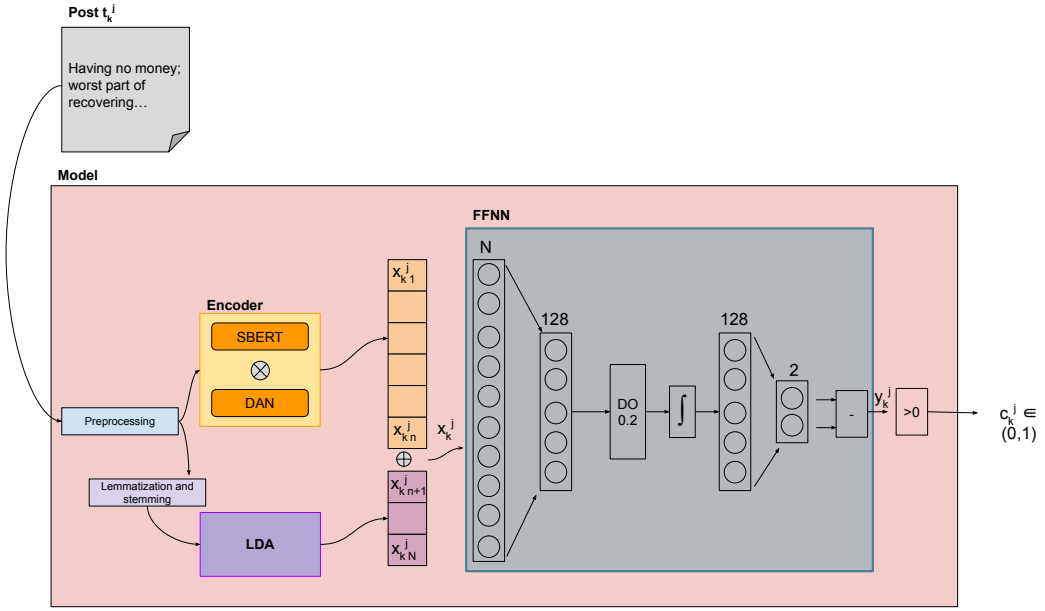

Figura 5.2: Diseño del modelo para la clasificación binaria a nivel de post. El post j de un usuario k (t_k^j) se representa como una matriz numérica ($x_{k1}^j, \ldots, x_{kN}^j$). La entrada de la red FFNN es la concatenación del vector generado por el *Encoder* y el modelo LDA. La salida de la FFNN (c_k^j) es la etiqueta a nivel de post

peso de lo habitual debido a la presencia de ruido en textos de esta índole.

En el caso del modelo que utiliza LDA, se requiere de un procesamiento más profundo. Con el objetivo de mejorar la calidad de los resultados obtenidos, es necesario reducir la dimensionalidad del vocabulario y asegurar que palabras similares se traten de la misma manera. Dicho esto, la biblioteca NLTK ofrece herramientas para aplicar la lematización y el stemming. Al aplicar estas herramientas al texto previamente procesado, se obtiene una reducción de las palabras a su forma base, provocando que las palabras que comparten la misma raíz se traten como una sola entidad. Al combinar ambas técnicas se busca capturar más variantes de palabras en el texto.

La tabla 5.1 muestra las versiones de un texto tras aplicar distintas técnicas de preprocesamiento. La última versión sería la necesaria en el caso de aplicar LDA.

Original	Preprocessing	Lemmatization & stemming
Having no money; worst part of recovering	money worst part recovering	money worst part recover

Tabla 5.1: Comparación del texto original y las versiones preprocesadas, donde Preprocessing se refiere a aplicar las transformaciones mencionadas, y Lem. & Stem. genera la forma base

5.2.2. Representación vectorial a nivel de post

Tal y como se ha mencionado anteriormente, para trabajar con entradas en formato de texto, es necesario convertirlas en datos numéricos. Tenemos que conseguir convertir cada post, t_k^j, en un vector numérico de tamaño fijo, $\mathbf{x}_k^j = (x_{k1}^j, x_{k2}^j, \ldots, x_{kN}^j) \in \mathbb{R}^N$ que serviría como entrada a la FFNN. Nótese que $t_k^j \in \Sigma^*$ siendo Σ el vocabulario de entrada.

Se exploraron dos estrategias principales para obtener una representación numérica ($\mathbf{x}_k^j$) dada una entrada (t_k^j): codificación y LDA. Podemos utilizar sólo una estrategia o ambas y hacer uso de la representación concatenada dando lugar, así, a una representación vectorial más larga, como en (5.1), en la que la vectorización mediante el codificador dio lugar a una matriz $N_{encoder} = n$ dimensional, $v_{encoder}(t_k^j) = (x_{k,1}^j, x_{k,2}^j, \ldots, x_{k,n}^j)$ y el LDA produjo una matriz $N_{LDA} = N - n + 1$ dimensional, $v_{LDA}(t_k^j) = (x_{k,n+1}^j, x_{k,n+2}^j, \ldots, x_{k,N}^j)$. El texto, representado como una matriz de tamaño fijo cuya dimensión (N) depende de la representación utilizada (codificación, LDA o ambas). La representación vectorial resultante $\mathbf{x}_k^j \in \mathbb{R}^N$ es, de hecho, la entrada para el clasificador.

$$\begin{aligned} v : \Sigma^* &\longrightarrow \mathbb{R}^N \\ t_k^j &\longrightarrow v(t_k^j) = (x_{k,1}^j, x_{k,2}^j, \ldots, x_{k,n}^j, x_{k,n+1}^j, x_{k,n+2}^j, \ldots, x_{k,N}^j) = \mathbf{x}_k^j \end{aligned} \tag{5.1}$$

De esta forma, se han utilizado tanto SBERT como la variante de USE, DAN, para generar representaciones semánticas de las oraciones. Tal y como se estudia en el apartado 5.3.1, la representación empleada tiene un gran efecto en el rendimiento final del modelo. En el caso de utilizar LDA, la codificación empleada como entrada de la red puede extenderse. La distribución de probabilidad de palabras generada por LDA se convierte en un vector que puede agregarse como información adicional.

A continuación, se dan detalles de cada estrategia explorada para obtener la representación, el codificador en la sección 5.2.2.1 y LDA en la sección 5.2.2.2.

5.2.2.1. Encoder a nivel de post

SBERT

Para poder utilizar el modelo pre-entrenado SBERT, Python tiene disponible la biblioteca "Sentence-transformers", que facilita la aplicación de varios modelos previamente entrenados. Entre todas las variantes disponibles, se optó por la versión “all-MiniLM-L6-v2". Este modelo fue entrenado con un dataset de más de mil millones de pares de textos, en las tareas de codificación y el cálculo de similitud semántica, logrando codificar sentencias en un vector de dimensión 384. La elección de este modelo se basó en la presencia de mensajes de Reddit en el dataset de entrenamiento, que representan un 62 % del total. Debido a esto, el modelo reconocerá gran parte del vocabulario presente en esta tarea.

DAN

En esta ocasión, se ha empleado la biblioteca "TensorFlow Hub"que proporciona un repositorio de modelos pre-entrenados de alta calidad. La cuarta versión del modelo de Google está optimizado para ofrecer representaciones de alta calidad con una salida de tamaño 512. El modelo se entrenó en múltiples tareas y con una amplia variedad de datos, lo que le permite ser aplicado tanto para clasificación, clustering y otras tareas de procesamiento de lenguaje natural.

5.2.2.2. Modelado de tópicos

Latent Dirichlet Allocation (LDA) es un modelo probabilístico capaz de identificar los temas latentes en los mensajes. Permite extraer la distribución de temas de cada entrada y

lo utilizamos como características adicionales para representar la entrada. Configuramos LDA para extraer 20 temas de los mensajes, lo que lleva a una representación $\mathbf{x}_k^j \in \mathbb{R}^{20}$.

Para el modelado de tópicos, se hizo uso de la biblioteca de Python "Tomotopy". Su sencilla interfaz permite construir y entrenar modelos de manera eficiente. Además, su uso eficiente de la memoria lo hace adecuado para manejar grandes volúmenes de datos. Adicionalmente, proporciona una serie de parámetros ajustables para adaptar el modelo a las necesidades del proyecto, como el número de tópicos a generar.

5.2.2.3. Combinación de atributos descriptores

En cuanto a la combinación de atributos descriptores, se lleva a cabo la concatenación del vector de representación textual con el vector generado por el modelo LDA. Esta técnica permite aprovechar la información capturada tanto por el *encoder* como por el modelo LDA. Otros métodos posibles hubiesen sido realizar una suma o la multiplicación, incluso mediante aprendizaje automático. Sin embargo dada la naturaleza de los atributos y los resultados de los antecedentes se ha inferido que una combinación mediante la técnica de concatenación sería más adecuada.

5.2.3. FFNN

La construcción de la red neuronal se realizó mediante "PyTorch", una biblioteca de aprendizaje automático que ofrece una amplia gama de herramientas y funciones tanto para la construcción como para el entrenamiento. La red neuronal que hemos construido consta de dos capas, con un número total de parámetros que varía entre 49.538 y 65.922, dependiendo del *encoder* utilizado, y con 2.560 parámetros adicionales en caso de aplicar LDA.

Para el entrenamiento, se ha empleado una tasa de aprendizaje (*learning rate*) de 5×10^{-5} junto con 5 épocas. Tal y como se mencionó anteriormente, se ha entrenado aplicando la técnica de *Drop-out* con un valor de 0,2, desactivando aleatoriamente el 20 % de las neuronas de la capa intermedia, con el objetivo de evitar posibles sobreajustes y mejorar la generalización del modelo. Por otro lado, el entrenamiento se realiza utilizando el algoritmo AdamW, que combina el método de optimización Adam con la técnica de *weight decay*. Ambas técnicas son muy comunes en el entrenamiento de redes neuronales y han demostrado ser eficaces para mejorar el rendimiento. Se aplicó un enfoque de entrenamiento iterativo en el que, en cada época, todos los mensajes de un usuario se procesan secuencialmente para actualizar los parámetros.

5.2.4. Clasificación a nivel de post

La confianza estimada en la etiqueta de post $y_k^j \in \mathbb{R}$ es un valor escalar intermedio obtenido en nuestro sistema, interpretado como la puntuación de confianza de que el post t_k^j contiene rasgos de lenguaje relacionados con el Juego Patológico. Cuando y_k^j toma valores positivos, la etiqueta intermedia a nivel de post c_k^j se asigna a la clase 1; de lo contrario, se asigna a la clase 0, es decir, $g(z) = sign(z)$. Cabe destacar que la transformación $f(\cdot)$ se logra mediante la red FFNN.

Estos procesos se resumen formalmente en (5.2) y se representan gráficamente en la figura 5.2.

$$\begin{array}{rcccc} g \circ f : \mathbb{R}^N & \longrightarrow & \mathbb{R} & \longrightarrow & \{0,1\} \\ \mathbf{x}_k^j & \longrightarrow & f(\mathbf{x}_k^j) = y_k^j & \longrightarrow & g(y_k^j) = c_k^j \end{array} \tag{5.2}$$

5.2.5. Clasificación a nivel de usuario

La etiqueta de referencia (*gold label*) del usuario u_k es la etiqueta del usuario (ya sea 'Control' o 'Jugadores Patológicos') para el usuario k-ésimo según el estándar de referencia, es decir, la etiqueta esperada para el sujeto. Con la información obtenida a nivel de post (como se indica en la sección 5.2.4), se estima la etiqueta a nivel de usuario ($\widehat{u_k}$). El rendimiento del sistema se evalúa, de hecho, en función de la diferencia entre las etiquetas a nivel de usuario predichas ($\widehat{u_k}$) y esperadas (u_k).

Sin embargo, cabe destacar una sutileza en la organización de la tarea: no todos los posts del usuario k se presentan conjuntamente, sino que los posts se presentan al sistema de forma secuencial, uno a uno en su turno. Es decir, para el usuario k en el momento de tiempo 1, solo contamos con el post t_k^1, mientras que en el momento de tiempo l habremos visto una secuencia de l posts, $(t_k^1, t_k^2, \ldots, t_k^l)$. Para cada mensaje, el sistema debe proporcionar una evaluación a nivel de usuario. Por lo tanto, en la l-ésima ronda, el sistema ha proporcionado una secuencia de l salidas $\widehat{\mathbf{u}}_k = (\widehat{u}_k^1, \widehat{u}_k^2, \ldots, \widehat{u}_k^l)$.

Dado el planteamiento establecido por la competición, una vez que el sistema clasifica a un usuario con la clase 1, solo se considerará el momento de tiempo en el que se realiza para la evaluación, ignorando el resto de asignaciones para rondas posteriores.

En el momento en que el sistema necesita calcular la etiqueta actual a nivel de usuario $\widehat{u}_k^{l+1}$, cuenta con el post actual t_k^{l+1} y todo el historial:

- secuencia pasada de posts e, inherentemente, su codificación correspondiente: $\mathbf{t}_k = (t_k^1, t_k^2, \ldots, t_k^l)$ y $(v(t_k^1), v(t_k^2), \ldots, v(t_k^l))$
- secuencia pasada de la confianza estimada en la etiqueta de post: $\mathbf{y}_k = (y_k^1, y_k^2, \ldots, y_k^l)$
- secuencia pasada de las etiquetas intermedias a nivel de post: $\widehat{\mathbf{c}}_k = (c_k^1, c_k^2, \ldots, c_k^l)$

Toda esta información está disponible y puede ser utilizada para generar $\widehat{u}_k^{l+1}$ como se indica en (5.3).

$$\begin{array}{rcl} h : \Sigma^* \times (\Sigma^*)^l \times \mathbb{R}^l \times \{0,1\}^l & \longrightarrow & \{0,1\} \\ (t_k^{l+1}, \mathbf{t}_k, \mathbf{y}_k, \widehat{\mathbf{c}}_k) & \longrightarrow & h(t_k^{l+1} | \mathbf{t}_k, \mathbf{y}_k, \widehat{\mathbf{c}}_k) = \widehat{u_k}^{l+1} \end{array} \tag{5.3}$$

Sin embargo, en nuestro enfoque, la etiqueta a nivel de usuario $\widehat{u}_k^{l+1}$ se estima como se muestra en (5.4). Es decir, calculamos la etiqueta a nivel de usuario basándonos únicamente en la etiqueta a nivel de post actual, sin tener en cuenta la información previa. Cabe mencionar que en futuros esfuerzos, podrían aprovecharse las capacidades de $h(\cdot)$ utilizando toda la información disponible.

$$\widehat{u_k}^{l+1} = h(t_k^{l+1} | \mathbf{t}_k, \mathbf{y}_k, \widehat{\mathbf{c}}_k) = g(f(v(t_k^{l+1}))) = \widehat{c}_k^{l+1} \tag{5.4}$$

5.2.6. Clasificación de Ranking

Tal y como se mencionó en el inicio del capítulo, se ha propuesto una tarea adicional, donde los participantes tienen que realizar una ordenación de los usuarios en función de la gravedad del trastorno. Al carecer de información sobre la pauta por la que se regirán los jueces o ejemplos con los que entrenar el modelo en dicha tarea, se ha optado por una aproximación que reutilice el modelo diseñado para la tarea principal. El valor que se le asigna a un usuario por cada mensaje, será igual a la confianza estimada de la etiqueta de post, y_k^j.

5.3. Sensibilidad del modelo a distintos parámetros

Con objeto de explorar la sensibilidad del sistema ante distintos parámetros, en este apartado hemos entrenado diferentes variantes del sistema. En esta serie de pruebas se han variado los parámetros que hemos considerado más sensibles para el sistema, que son los siguientes: el tipo de representación, la calidad del etiquetado, el uso del resultado de LDA como característica adicional y el ajuste de la función de pérdida empleada. A continuación, en las secciones que van de la 5.3.1 a la 5.3.3, exploramos los resultados preliminares obtenidos con cada una de estas variantes.

5.3.1. Influencia del etiquetado a nivel de post

Dado que las etiquetas de referencia se proporcionan a nivel de usuario y entrenamos nuestra red FFNN utilizando posts, necesitábamos obtener los posts etiquetados para el entrenamiento. Es decir, en la etapa de entrenamiento, la confianza estimada en la etiqueta de post (y_k^j) debe compararse con una confianza deseada o esperada ($\mathbf{y'}_k$), y el problema subyacente radica en que la confianza a nivel de post no se proporciona directamente. Este *silver standard* se muestra en la figura 5.4. Como referencia de confianza en la etiqueta de post, $\mathbf{y'}_k = (y'^1_k, \ldots, y'^l_k)$, en este trabajo exploramos dos estrategias alternativas de asignación de *silver standard*:

- Etiquetado de mensajes basado en el usuario (EBU): Consiste en asignar a cada mensaje la etiqueta del usuario. Es decir, si un usuario es positivo todos sus mensajes serán etiquetados como positivos, es decir, todos los componentes de este *array*, y'_k, serán iguales a u_k.

- *Approximated Nearest Neighbors* (ANN): Las entradas se etiquetan utilizando un método iterativo. Primero, cada mensaje hereda la etiqueta del usuario. A continuación, utilizando la técnica ANN, se reasignan las etiquetas, dando a los mensajes más cercanos las mismas etiquetas, obteniendo $\mathbf{y'}_k$ como en [21].

Las referencias heurísticas a nivel de post empleadas tienen un profundo impacto en la etapa de entrenamiento y, como es de esperar, deben seleccionarse cuidadosamente. Trabajos futuros pueden abordar estrategias alternativas de asignación de referencias.

En la figura 5.3 se han resumido las dos formas de etiquetado. El usuario conlleva una serie de mensajes (representados por 5 mensajes en gris a la izquierda de la figura). Deseamos asignar una etiqueta a cada mensaje, dando lugar a un vector de dimensión igual

al número de mensajes. Hemos explorado dos tipos de etiquetado: en el etiquetado EBU, los mensajes heredan la etiqueta de usuario; en el etiquetado ANN, los mensajes se etiquetan según ANN.

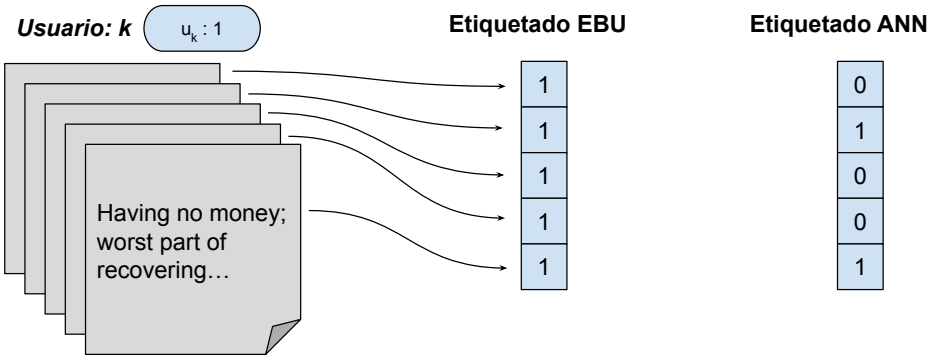

Figura 5.3: Tipos de etiquetado: EBU y ANN

Como se mencionó anteriormente, el uso de un etiquetado u otro tiene un impacto directo en el entrenamiento de la red neuronal. Diferentes etiquetas implican valores distintos de la función de pérdida y, por lo tanto, un modelo con parámetros diferentes. Mientras que en enfoques como el propuesta por el grupo UNSL, con técnicas de ANN para la clasificación, es necesario un etiquetado preciso. En el contexto de una red neuronal, esto implica disminuir la cantidad de muestras pertenecientes a la clase 1, lo que agrava el desequilibrio de los datos y aumenta el sesgo hacia la clase 0. Tal y como se reflejará en la sección 6.2, esto ha resultado en una disminución de 20 puntos en la métrica *F1*.

5.3.2. Influencia de la representación vectorial

En este apartado estudiamos la influencia de las diferentes vectorizaciones generadas y nos centramos en determinar la conveniencia de incorporar o no LDA.

La representación utilizada, además de determinar el tamaño de la entrada de la red neuronal, condiciona el rendimiento final del sistema. Las diferencias entre los modelos SBERT y DAN hacen que cada uno genere un vector que representa la misma sentencia de formas distintas. Aunque la versión de DAN generaba un vector de mayor tamaño y, por ende, permite obtener representaciones más complejas, el modelo SBERT ha demostrado un mejor rendimiento, indicando una mejor comprensión del lenguaje natural (*Natural Language Understanding*), tal y como muestran los experimentos posteriores.

Por otro lado, el uso de LDA en el input de la red neuronal ha demostrado ser beneficioso en varias aplicaciones. Incorporar información de tópicos latentes suele mejorar la representación de texto. Sin embargo, debido a la naturaleza compleja de los datos, no solo ha agregado complejidad al modelo, sino que también ha reducido en 10 puntos la métrica *F1*, como se verá posteriormente.

5.3.3. Influencia de la función de coste

Como se menciona en la sección 4, los datos, lejos de ser uniforme, presentan un desequilibrio notable. Como consecuencia, la red neuronal puede verse sesgada y obtener una baja precisión en la clase minoritaria. Para hacer frente a este problema, existen diversas estrategias para alterar la proporción de clases mediante sobre-muestreo y sub-muestreo. Siguiendo un enfoque más cercano a asignar pesos de clase, hemos optado por aplicar una función de pérdida basada en la entropía cruzada durante el entrenamiento de la red neuronal. La estrategia seguida en nuestro trabajo se esboza en la Figura 5.4.

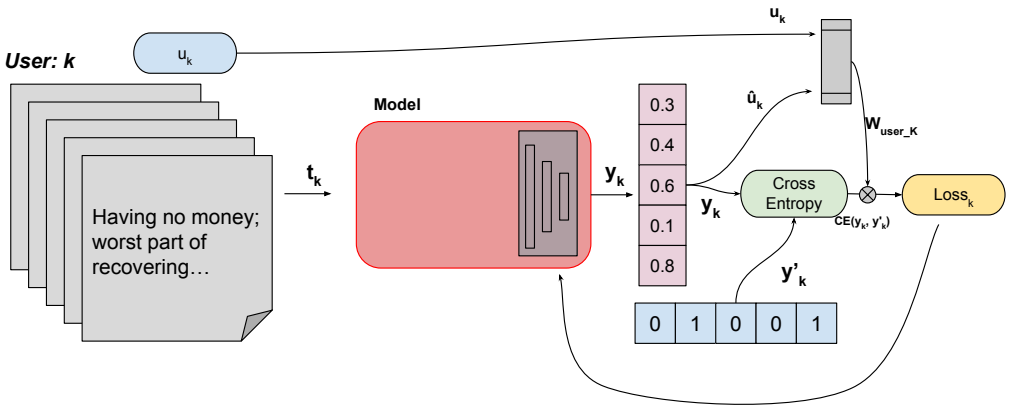

Figura 5.4: Enfoque de entrenamiento del sistema. Se utiliza la Entropía Cruzada para calcular la función de pérdida y actualizar el modelo para un usuario dado. Se mostró un acercamiento al modelo en la Figura 5.2

Con la secuencia de posts del usuario k, el modelo calcula una secuencia de valores de confianza, post por post, y se obtiene $\mathbf{y}_k$. Con esto, como se mencionó en la sección 5.2.4, se calcula una secuencia de etiquetas a nivel de post, es decir, $\widehat{\mathbf{c}}_k$. En la etapa de entrenamiento, el sistema estima una etiqueta a nivel de usuario teniendo en cuenta todas las etiquetas a nivel de post, como se muestra en (5.5), lo que significa que una etiqueta igual a 1 a nivel de post en la secuencia, es suficiente para clasificar al usuario como ludópata. En la etapa de entrenamiento, la etiqueta a nivel de usuario se estima comprendiendo todas las etiquetas a nivel de post del usuario. La etiqueta a nivel de usuario estimada se puede comparar con la etiqueta verdadera (u_k) para actualizar el modelo.

$$\widehat{u_k} = \begin{cases} 1, & \text{si } \exists i \quad 1 \leq i \leq l \quad : \quad c_k^i = 1 \\ 0, & \text{en caso contrario} \end{cases} \tag{5.5}$$

En la etapa de entrenamiento, la secuencia de etiquetas calculadas ($\mathbf{y}_k$) se compara con las etiquetas de referencia propuestas ($\mathbf{y'}_k$) presentadas en la sección 5.3.1. Esta comparación se cuantifica como la pérdida mediante la función de pérdida de Entropía Cruzada, $H(\mathbf{y'}_k, \mathbf{y}_k)$, implementada en PyTorch [34]. La pérdida dependiente del usuario se captura mediante un factor de ponderación que permite una penalización según se muestra en (5.6).

$$Loss_k = W_{user_k} \cdot H(\mathbf{y'}_k, \mathbf{y}_k) \tag{5.6}$$

Nuestro enfoque de entrenamiento no penaliza de igual manera los falsos positivos y los falsos negativos, de hecho, el factor de ponderación utilizado por nuestro equipo se muestra en (5.7).

$$W_{user_k} = \begin{cases} 4 & \text{si } \widehat{u_k} = 0 \wedge u_k = 1 \\ 2 & \text{si } \widehat{u_k} = 1 \wedge u_k = 0 \\ 1 & \text{si } \widehat{u_k} = u_k \end{cases} \tag{5.7}$$

En los experimentos, se utilizó una penalización de 2 para los falsos positivos y de 4 para los falsos negativos. Los falsos negativos críticos se penalizaron duplicando la pérdida en aquellos casos en los que el sistema predijo incorrectamente una instancia positiva. Estos valores (1, 2 y 4) se determinaron mediante un análisis de sensibilidad y en base al objetivo de priorizar los falsos positivos sobre los falsos negativos en un intento de no pasar por alto a los jugadores patológicos.

Hay margen de mejora en la etapa de entrenamiento. Por un lado, la estrategia de etiquetado a nivel de usuario, es decir, (5.5), podría calcularse teniendo en cuenta la marca de tiempo y no solo la secuencia de etiquetas de los posts, sin embargo, la función propuesta es computacionalmente económica y adecuada para abordar el desequilibrio de clases. Por otro lado, la función de pérdida y el peso de penalización podría ser estudiada con mayor profundidad para lograr un equilibrio adecuado.

5.4. Evaluación

Todos los experimentos se han evaluado siguiendo un esquema de evaluación *hold-out* manteniendo las particiones de entrenamiento y evaluación para poder compararnos con los antecedentes. Podríamos haber hecho *repeated hold-out* o *k-fold cross validation* para tener una idea de la variabilidad del modelo y la sensibilidad ante ligeras variaciones en los datos. No obstante, debido a las limitaciones en el tiempo, se optó por dejarlo como futura mejora.

Las métricas de evaluación son esenciales en cualquier tarea de clasificación, puesto que nos permite medir la calidad del modelo desarrollado. Dada su importancia se ha optado por recrear las condiciones de evaluación que se usarán en la competición. De esta forma, durante la implementación se puede comparar con modelos de ediciones pasadas.

Las métricas presentadas a lo largo de este capítulo son las mismas que se definen en el artículo [35].

5.4.1. Métricas de clasificación

5.4.1.1. Métricas estándares en predicción

Sea un usuario u de un conjunto de usuarios U, denotado mediante $u \in U$. El sistema analiza k_u mensajes de u y con esa información clasifica al usuario de la forma $d_u \in \{0, 1\}$; la etiqueta real (etiqueta esperada) se define como $g_u \in \{0, 1\}$ (donde g: *gold*). De esta forma las métricas estándares tales como *Precision (P)*, *Recall (R)* y *F-measure (F)* se definen según las expresiones (5.8)-(5.10).

$$P = \frac{|u \in U : d_u = g_u \wedge g_u = 1|}{|u \in U : d_u = 1|} \tag{5.8}$$

$$R = \frac{|u \in U : d_u = g_u \wedge g_u = 1|}{|u \in U : g_u = 1|} \tag{5.9}$$

$$F = 2 \cdot \frac{P \cdot R}{P + R} \tag{5.10}$$

5.4.1.2. Métricas específicas en predicción temprana

Estas métricas, (5.8)-(5.10), sólo toman en cuenta el rendimiento del sistema después de haber visto k_u mensajes, sin embargo, no favorecen a los sistemas que son capaces de dar su predicción consumiendo menos mensajes (con el menor k_u posible), es decir, a los sistemas que son capaces de anticiparse en sus predicciones. Para tomar en cuenta no sólo la calidad de las predicciones sino también la dinámica de los sistemas, se define la métrica $latency_{TP}$, según la expresión (5.11), que penaliza el retraso del sistema a la hora de detectar casos positivos. Dado un conjunto de usuarios $U = \{u1, u2, u3, u4\}$, donde los usuarios $u1,\ u2\ y\ u4$ pertenecen a la clase positiva y las predicciones del sistema para ellos son correctas. Estableciendo que se han necesitado 12, 10 y 8 mensajes para llegar a ese resultado. Tendremos que $latency_{TP} = median\{12, 10, 8\} = 10$, es decir, que el sistema en promedio detecta los casos positivos después de procesar 10 mensajes.

$$latency_{TP} = \text{median}\{k_u : u \in U, d_u = g_u \wedge g_u = 1\} \tag{5.11}$$

De la misma forma, la velocidad del modelo se mide con la métrica $speed$ (5.12). En la fórmula se aplica una función de penalización a la cantidad de mensajes necesaria para etiquetar correctamente a un usuario de la clase positiva. Dado un valor de *speed* igual a 1 refleja un sistema que identifica los casos con el primer mensaje, mientras que valores cercanos a 0 describen un sistema que necesita cientos de mensajes para realizar la misma clasificación.

$$speed = (1 - \text{median}\{penalty(k_u) : u \in U, d_u = g_u \wedge g_u = 1\}) \tag{5.12}$$

Tomando un factor de penalización basado en el retraso medio, con un parámetro p ajustado a $0,0078$, tal y como indica la organización.

$$penalty(k_u) = -1 + \frac{2}{1 + \exp(-p \cdot (k_u - 1))} \tag{5.13}$$

Finalmente, la métrica $F_{latency}$ (la métrica *F-score* ponderada según la latencia) combina la efectividad de la decisión con el retraso implícito. Es decir, $speed$ actúa como factor de ponderación de la métrica *F* y esta ponderación refleja la latencia en la respuesta.

$$F_{latency} = F_1 \cdot speed \tag{5.14}$$

Dada esta combinación entre la métrica *F* y *speed*, solo aquel sistema que necesite de un único mensaje para realizar la clasificación mantendrá el mismo valor para *F* como de $F_{latency}$. Por otro lado, un modelo que necesite de 10 mensajes verá disminuido el valor de $F_{latency}$ un 35 % frente a *F*.

Asimismo, siendo *ERDE* una medida que penaliza las respuestas correctas tardías, es decir, la penalización crece a medida que se retrase en asignar a un usuario como positivo, siendo el número de mensajes procesados la medida de tiempo. Se propone una variante, cuya penalización dependa del porcentaje de mensajes vistos. Se puede ver una descripción completa en [36].

5.4.2. Métricas de ranking

Como una forma de evaluación alternativa, la edición de 2019 presentó la tarea complementaria de ordenación. Por cada uno de los mensajes analizados, los participantes asignaban al usuario un nivel de riesgo teniendo en cuenta el riesgo visto. De esta forma se construyen rankings de los usuarios por cada cantidad de mensajes analizados.

Cada clasificación puede ser evaluada mediante métricas estándar de recuperación de información (*Information Retrieval* denotado por IR). Estas métricas se utilizan para medir la calidad y el rendimiento de los sistemas. Entre las métricas destacan el *P@K* y *NDCG*.

Precision at K (P@K): se refiere a la precisión en los primeros k, mide la proporción de usuarios relevantes presentes en las primeras k posiciones del ranking. Por lo tanto, P@K será un valor acotado entre 0 y 1, donde para obtener un 1 será necesario la aparición de todos los elementos relevantes, independientemente del orden en el que aparezcan.

$$P@K = \frac{\text{Número de elementos relevantes en los primeros k}}{\text{k}} \tag{5.15}$$

Normalized Discounted Cumulative Gain (NDCG): es una métrica que tiene en cuenta tanto la relevancia de los elementos como su posición en la clasificación [37].

CAPÍTULO 6

Marco experimental

La parte experimental de este trabajo, se arraiga en la experimentación y resultados obtenidos con el conjunto de datos de la competición eRisk 2022, donde estudiamos la influencia de los parámetros mencionados en la sección 5.3. Los resultados obtenidos, imitando las condiciones de los antecedentes, se recogen en la sección 6.2. Posteriormente, una vez estudiado el efecto de los diferentes parámetros se realizó la prueba de varias variantes del sistema por parte de la competición. En la sección 6.3 se encuentra un análisis detallado de los resultados obtenidos en la competición eRisk 2023.

6.1. Hardware

6.1.1. Máquinas de cálculo

Todos y cada uno de los experimentos realizados se han ejecutado en los servidores proporcionados por el grupo IXA. Una descripción detallada con la información de la *CPU* puede ser encontrada en la tabla 6.1. Adicionalmente, el dispositivo estaba equipado con 4 *GPUs Titan XP*, lo que ha permitido completar el entrenamiento de los modelos en un breve periodo de tiempo.

No se ha necesitado de un periodo de adaptación, debido a la experiencia previa colaborando con dicho grupo y en dichas máquinas. Los recursos disponibles me han permitido llevar a cabo numerosas pruebas con celeridad y hallar el diseño definitivo. El entrenamiento de la red neuronal del modelo se ha llevado a cabo con un tiempo de 3 minutos. Esta fase de experimentación incluye los correspondientes fracasos iniciales empleando diferentes arquitecturas, que comprenden tanto técnicas de aprendizaje automático como los diseños planteados por los concursantes de eRisk 2022.

6.1.2. Protocolo de comunicación con los servidores

Hay un aspecto técnico que desconocíamos cuando iniciamos la competición y es el relativo a la comunicación con el servidor en la fase de evaluación. Cada equipo dispone de un *token* para identificarse con el servidor. Una vez iniciada la comunicación, el servidor proporciona iterativamente los mensajes de los usuarios.

Propiedad	Valor
Arquitectura	x86_64
CPU op-mode(s)	32-bit, 64-bit
Byte Order	Little Endian
CPU(s)	40
On-line CPU(s) list	0-39
Thread(s) per core	2
Core(s) per socket	10
Socket(s)	2
NUMA node(s)	2
Vendor ID	GenuineIntel
CPU family	6
Model	79
Model name	Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz
Stepping	1
CPU MHz	1273.535
CPU max MHz	3400.0000
CPU min MHz	1200.0000
BogoMIPS	4794.77
Virtualization	VT-x
L1d cache	32K
L1i cache	32K
L2 cache	256K
L3 cache	25600K

Tabla 6.1: Especificaciones de la máquina

La figura 6.1 muestra el formato de envió del servidor. A continuación, se aplica el modelo para generar una estimación de la clase a la que pertenece cada usuario junto con el nivel de riesgo. Una vez completado el proceso y enviado al servidor, se obtendrá el siguiente subgrupo de mensajes

Para facilitar todo este proceso de comunicación entre el servidor y el modelo del participante, la organización proporcionaba una plantilla en varios lenguajes de programación[1].

6.2. Resultados obtenidos sobre el test 2022

Todas las variantes exploradas en esta sección han sido entrenadas con los datos referentes a la edición de 2021. Como test, se ha utilizado el conjunto de datos descrito en la tabla 6.2. Dicho conjunto de evaluación consiste en 81 jugadores compulsivos y 1.998 de control. De media, cada usuario realizó un total 180 y 507 publicaciones respectivamente en un intervalo de 489 y 664 días. Los jugadores compulsivos empleaban una media de 30 palabras frente a las 22 de los usuarios de control.

Para evaluar el nivel de los sistemas que hemos propuesto en la sección 5.3, hemos simulado las mismas condiciones que la edición 2022. La tabla 6.3 refleja algunas de las actuaciones más notables de los participantes del año 2022 [35]. Tanto UNED-NLP con su

[1] http://gitlab.irlab.org/javier/erisk-dummy-client/

```
[
    {
        "id": 18752,
        "number": 0,
        "nick": "subject3798",
        "redditor": 18702,
        "title": "...",
        "content": "...",
        "date": "..."
    },
    {
        "id": 18772,
        "number": 0,
        "nick": "subject7495",
        "redditor": 18703,
        "title": "...",
        "content": "...",
        "date": "..."
    },
        ...
]
```


Figura 6.1: Formato de la comunicación. Number indica el número de ronda, number=0 representa el primer envío de mensajes. ID es un identificador del mensaje, Redditor es un identificador para el usuario

	Jugadores patológicos	*Control*
Núm. de sujetos	81	1998
Núm. de mensajes	14.627	1.014.122
Promedio de mensajes por sujeto	180,58	507,56
Promedio de días desde el primer hasta el último mensaje	$\approx$ 489,7	$\approx$ 664,9
Promedio de palabras por mensaje	30,4	22,2

Tabla 6.2: Principales estadísticas de la colección de pruebas de eRisk 2022

cuarto intento, como el grupo SINAI en su segunda versión, lograron unos resultados notorios en la métrica *F1* de 0,869 y 0,808 respectivamente. Los diseños consistieron en técnicas de *Approximate Nearest Neighbors* y el uso de características adicionales relacionadas con la volumetría, la diversidad léxica, las métricas de complejidad y los puntajes emocionales, que proporcionaron información valiosa a la propuesta de SINAI. Otros intentos, como el del grupo UNSL y NLPGroup-IISERB, obtuvieron un gran éxito respecto al *Recall*, a costa de una baja precisión con lo que terminaron con un *F1* inferior.

Habiendo establecido las referencias y visto la complejidad que supone realizar una buena actuación, los resultados, reflejados en la tabla 6.4, de la versión del sistema que hace uso de SBERT, un etiquetado básico y la función de pérdida modificada (presentado en la subsección 5.3), ha logrado superar las propuestas mencionadas con un *F1-score* de 0,88 para la clase positiva (número 1). Además de la mejora explícita, demuestra un mejor rendimiento al no procesar los datos de forma tan compleja como SINAI y resuelve la limitación de técnicas como el vecino más próximo en una población grande. Por otro lado, dada la función de pérdida modificada es posible lograr el equilibrio deseado frente a los

Modelo	Precision	Recall	F1-score
UNED-NLP R0	0.285	0.975	0.441
UNED-NLP R1	0.555	0.938	0.697
UNED-NLP R2	0.296	**0.988**	0.456
UNED-NLP R3	0.536	0.926	0.679
UNED-NLP R4	**0.809**	0.938	**0.869**
SINAI R2	**0.908**	0.728	**0.808**
BioInfo_UAVR R1	0.067	**1.000**	0.126
RELAI R2	0.052	0.963	0.099
BLUE R0	0.260	0.975	0.410
BioNLP_UniBuc R4	0.046	**1.000**	0.089
UNSL R1	0.461	0.938	0.618
NLPGroup-IISERB R3	0.140	**1.000**	0.246
stezmo3 R4	0.160	0.901	0.271

Tabla 6.3: Mejores resultados de la edición de 2022

falsos positivos y negativos.

Class	Precision	Recall	F1-score	Support
0	0.99	0.99	0.99	1998
1	0.88	0.88	0.88	81
Accuracy	-	-	0.99	2079
Macro avg	0.94	0.94	0.94	2079
Weighted avg	0.99	0.99	0.99	2079

Tabla 6.4: Resultados de la variante de SBERT, con función de coste modificada

Los sistemas restantes han mostrado un rendimiento muy variado según la configuración elegida. El proceso de refinar las etiquetas, el cual resultó un éxito en el modelo de UNED-NLP, ha supuesto una pérdida valiosa en el rendimiento, obteniendo un *F1-score* de 0,53, como podemos ver en la tabla 6.5. Esto se puede atribuir a la importancia de no trabajar con ruido en los modelos basados de ANN, mientras que en mi modelo supone reducir las muestras de instancias de clase 1 en un conjunto desbalanceado.

Class	Precision	Recall	F1-score	Support
0	0.98	1.00	0.99	1998
1	0.89	0.38	0.53	81
Accuracy	-	-	0.97	2079
Macro avg	0.93	0.69	0.76	2079
Weighted avg	0.97	0.97	0.97	2079

Tabla 6.5: Resultados de la variante de DAN, con función de coste modificada y etiquetado refinado

El uso de LDA, a pesar del éxito demostrado en otros trabajos, ha supuesto una pérdida del rendimiento con una reducción de 0,88 a 0,64 en el *F1-score*, reflejada en la tabla 6.6. Las condiciones del conjunto de datos, estudiado en capítulos anteriores, no han facilitado una correcta integración en el modelo, aumentando la complejidad sin una mejora en la precisión.

Class	Precision	Recall	F1-score	Support
0	1.00	0.96	0.98	1998
1	0.48	0.95	0.64	81
Accuracy	-	-	0.96	2079
Macro avg	0.74	0.95	0.81	2079
Weighted avg	0.98	0.96	0.96	2079

Tabla 6.6: Resultados de la variante de DAN, con función de coste modificada y LDA

Por otro lado, la comparación de las tablas 6.7 y 6.8 muestra una superioridad del modelo SBERT frente a DAN, con valores de 0,86 y 0,71 en el *F1-score* respectivamente, para estas condiciones. El uso de DAN en la solución planteada por el grupo UNED-NLP, puede ser explicada debido al uso de un vector de dimensionalidad más alta frente a la que lograrían usando SBERT, lo que facilitaría generar un espacio más beneficioso para las técnicas de ANN. Por otro lado, el uso de siamesas y de tripletas en el entrenamiento de SBERT ha resultado más conveniente para el diseño planteado en este trabajo.

Finalmente, las tablas 6.8 y 6.9 muestran los efectos de utilizar la función de pérdida modificada, presentada en el apartado 5.3.3. No solo se han obtenido mejores resultados en la métrica *F1*, 0,78 frente al 0,71 anterior, sino que se ha logrado un mayor control frente a posibles falsos positivos, obteniendo una precisión de 0,83 frente al 0,58 obtenido anteriormente.

Class	Precision	Recall	F1-score	Support
0	0.99	1.00	0.99	1998
1	0.90	0.81	0.86	81
Accuracy	-	-	0.99	2079
Macro avg	0.95	0.91	0.93	2079
Weighted avg	0.99	0.99	0.99	2079

Tabla 6.7: Resultados de la variante de SBERT

Class	Precision	Recall	F1-score	Support
0	1.00	0.97	0.98	1998
1	0.58	0.94	0.71	81
Accuracy	-	-	0.97	2079
Macro avg	0.79	0.96	0.85	2079
Weighted avg	0.98	0.97	0.97	2079

Tabla 6.8: Resultados de la variante de DAN

Class	Precision	Recall	F1-score	Support
0	0.99	0.99	0.99	1998
1	0.83	0.73	0.78	81
Accuracy	-	-	0.98	2079
Macro avg	0.91	0.86	0.88	2079
Weighted avg	0.98	0.98	0.98	2079

Tabla 6.9: Resultados de la variante de DAN, con función de coste modificada

6.3. Resultados obtenidos sobre el test 2023

Como test se ha utilizado el conjunto de datos descrito en la tabla 6.10. Consiste en 103 jugadores compulsivos y 2.071 de control. De media, cada usuario realizó un total 327 y 516 publicaciones respectivamente en un intervalo de 675 y 878 días. Los jugadores compulsivos empleaban una media de 28 palabras frente a las 20 de los usuarios de control.

	Jugadores patológicos	*Control*
Núm. de sujetos	103	2.071
Núm. de mensajes (publicaciones y comentarios)	33.719	1.069.152
Promedio de mensajes por sujeto	327,33	516,25
Promedio de días entre el primer y el último mensaje	$\approx$ 675	$\approx$ 878
Promedio de palabras por mensaje	28,9	20,47

Tabla 6.10: Principales estadísticas de la colección de pruebas de eRisk 2023

En esta prueba se llegaron a enviar 5 variantes que usaban DAN (explicado en la sección 5.2) para su representación y la misma función de pérdida modificada. La tabla 6.11 muestra la configuración de cada una de las variantes. Dada la limitación en el número de variantes a utilizar, se ha priorizado una mayor diversidad en las configuraciones frente a variantes que empleaban SBERT. Adicionalmente, se ha utilizado un modelo combinado, que hace uso de todas las variantes mencionadas para crear un único resultado. La decisión se determina mediante un OR, donde es condición necesaria y suficiente que una de las variantes asigne al usuario con la clase 1 para que esta versión así lo haga. La motivación detrás de esta acción no es otra que aumentar el *Recall*, a costa de ciertos falsos positivos que son preferibles a falsos negativos.

Run	Representación	LDA	Etiqueta	Entrenamiento
0	DAN	No	EBU	2021
1	DAN	Sí	EBU	2021
2	DAN	No	ANN	2021
3	DAN	No	EBU	2021 $\cup$ 2022
4	$OR_{i=0}^{3}(Run_i)$			

Tabla 6.11: Ejecuciones presentadas: Descripción de las configuraciones exploradas. La segunda columna se refiere a la estrategia de codificación (explicada en la sección 5.2.2.1), LDA se refiere a la incorporación de LDA en la representación vectorial a nivel de post (como en la sección 5.2.2.2), Label indica el etiquetado utilizado en el entrenamiento (sección 5.3.1), y finalmente, la edición del conjunto de entrenamiento utilizado (sección 4)

La tabla 6.12 muestra los mejores resultados obtenidos por algunos participantes en distintas métricas. Tanto el grupo de ELiRF-UPV, NLP-UNED-2 y algunos de mis modelos suponen un incremento considerable en el estado del arte establecido en las ediciones 2021 y 2022. La mejora en todas las métricas refleja un mayor entendimiento y comprensión en torno a esta tarea. Sin embargo, las dificultades de muchos equipos en desarrollar un sistema efectivo demuestran la complejidad que reside en ella.

Entre todas las variaciones del sistema, la primera configuración ha obtenido puntuaciones competitivas en todas las métricas, con un fuerte dominio en el *Recall*. En un entorno donde los falsos negativos tienen un gran peso, en aplicaciones reales supone de

gran interés tener un alto *Recall*, compensando pequeñas diferencias en otras medidas. Por otro lado, como ya se ha visto en el desarrollo la utilización de LDA ha supuesto una pérdida general en el rendimiento del sistema, mientras que el entrenamiento en base a un etiquetado más preciso se ha reflejado en una pérdida de la precisión del sistema. En general, estos resultados sugieren que hay múltiples enfoques efectivos para abordar el problema.

Team	Run	P	R	F1	$ERDE_5$	$ERDE_{50}$	latencyTP	speed	latency-weighted F1
ELiRF-UPV	0	**1.000**	0.883	**0.938**	**0.026**	**0.010**	**4.0**	**0.988**	**0.927**
Xabi_EHU	0	0.846	**0.961**	**0.900**	**0.030**	**0.012**	**8.0**	**0.973**	**0.875**
Xabi_EHU	1	0.89	0.864	0.877	0.035	0.017	12.0	0.957	0.839
Xabi_EHU	2	0.79	0.913	0.847	0.036	0.015	13.0	0.953	0.807
Xabi_EHU	3	0.829	0.942	0.882	0.033	0.013	12.0	0.957	0.844
Xabi_EHU	4	0.756	**0.961**	0.846	0.031	0.013	**8.00**	**0.973**	0.823
UNSL	2	0.752	0.854	0.800	0.048	0.013	14.0	0.949	0.759
BioNLP-IISERB	0	**0.933**	0.68	0.787	0.038	0.037	62.0	0.766	0.603
NLP-UNED-2	1	**0.957**	0.883	**0.919**	0.034	0.016	13.0	0.953	**0.876**

Tabla 6.12: Resultados de clasificación por modelo

En lo referente a la tarea adicional de ordenación de usuarios, la tabla 6.13 muestra el rendimiento de los mejores equipos en dicha tarea. Tal y como se puede ver, priorizar un mejor *Recall* ha supuesto en un rendimiento inferior comparado a otros equipos. A falta de información adicional, el emplear la propia probabilidad de la clase como nivel de riesgo del usuario, puede no ser un buen indicador para lo que deseaba la competición. No obstante, en lo referente a las métricas *P@10* y *NDCG@10* se han obtenido resultados competitivos en todas las categorías.

6.4. Análisis de resultados

El sistema propuesto ha logrado un rendimiento sobresaliente en las tareas de clasificación binaria y en la clasificación mediante ranking. Pese a ser una de las competiciones más relevantes dentro del ámbito del NLP y la salud, se ha conseguido superar el estado del arte establecido. Dado que el desarrollo del modelo se ha enfocado exclusivamente en la tarea principal sobre clasificación binaria, el rendimiento en la tarea de ordenación es inferior.

Con la debida información sobre la tarea de ordenación, se habría podido entrenar el modelo en ambas tareas de manera simultánea, lo cual habría mejorado el rendimiento en ambas tareas. El desbalanceo de datos ha sido un reto por afrontar durante todo el desarrollo, puesto que el modelo obtenía mejores resultados para la clase mayoritaria, pero es esa misma condición la que ha derivado en la creación de una función de pérdida específica. Este mismo desequilibrio en los datos, me ha impedido utilizar otras arquitecturas basadas en Aprendizaje automático, provocando que cualquier intento no lograran generalizar los

Team	Run	1 writing			100 writing			500 writing			1000 writing		
		P@10	NDCG@10	NDCG@100	P@10	NDCG@10	NDCG@100	P@10	NDCG@10	NDCG@100	P@10	NDCG@10	NDCG@100
Xabi_EHU	0	1.00	1.00	0.57	1.00	1.00	0.50	0.80	0.88	0.41	0.90	0.94	0.41
OBSER-MENH	0	1.00	1.00	0.64	1.00	1.00	0.55	1.00	1.00	0.48	1.00	1.00	0.50
ELiRF-UPV	0	1.00	1.00	0.59	1.00	1.00	0.91	1.00	1.00	0.95	1.00	1.00	0.94
UNSL	1	1.00	1.00	0.57	1.00	1.00	0.78	1.00	1.00	0.67	1.00	1.00	0.70
NLP-UNED-2	3	1.00	1.00	0.59	1.00	1.00	0.92	1.00	1.00	0.95	1.00	1.00	0.93

Tabla 6.13: Evaluación basada en el Ranking

datos y obtener un rendimiento satisfactorio. Este efecto se puede apreciar en los valores de las métricas separadas por clases, donde se obtienen resultados cercanos a 1 para la clase 0, a diferencia del caso de la clase 1.

Indudablemente, este mismo diseño puede ser extrapolado a otros trastornos mentales, incluso con textos en otros idiomas, con el *encoder* correspondiente. Los pesos de la función de pérdida modificada variarán según el balanceo de las clases, sin embargo, con una función de pérdida sin alterar, el modelo debería lograr buenos resultados. La aplicación de LDA, debe ser estudiada más en profundidad para lograr un incremento en el rendimiento.

CAPÍTULO 7

Conclusiones y trabajo futuro

7.1. Conclusiones

7.1.1. Objetivos alcanzados

En este trabajo se planteaba el reto de desarrollar un sistema que, mediante la aplicación de técnicas de aprendizaje profundo, fuese capaz de detectar indicios de ludopatía. Con ese objetivo en mente, se establecieron una serie de metas a alcanzar. El primer objetivo, el estudio del estado del arte, se llevó a cabo en las etapas iniciales del proyecto. El seguimiento de la evolución del estado del NLP aplicado a la salud mental, junto con las diferentes propuestas presentadas en las distintas ediciones del eRisk, me ha permitido ampliar mis conocimientos y aplicarlos en el diseño de una arquitectura propia.

El segundo objetivo planteado fue el desarrollo de un clasificador eficiente. Se logró alcanzar dicha meta mediante la implementación de un sistema capaz de etiquetar a un usuario según su riesgo de desarrollar ludopatía. El modelo presenta una arquitectura con un buen rendimiento, capaz de ser aplicado en situaciones reales.

El tercer objetivo, que consistía en participar en el eRisk 2023, se llevó a cabo con éxito. Tanto el estudio de los datos como la evaluación se realizaron dentro de los plazos estipulados. El modelo no solo evaluó a todos los usuarios que se utilizaron, sino que también consiguió realizarlo con un rendimiento ejemplar. Logrando una de las mejores posiciones en la historia de la competición, superando cualquier previsión inicial.

A nivel general, se puede considerar que se han cumplido los objetivos del proyecto, dado que se ha conseguido realizar un estudio exhaustivo de las propuestas surgidas en años anteriores, ofrecer un enfoque distinto capaz de aportar nuevas soluciones y realizar una actuación sobresaliente. Al mismo tiempo que se realizaba el trabajo, se ha hecho un acercamiento al campo de la investigación, enfrentando los desafíos propios de cualquier trabajo en la vanguardia y siguiendo de cerca los últimos avances.

7.1.2. Aportaciones científicas

El hecho de que el sistema desarrollado haya obtenido una de las mejores posiciones en esta competición es un indicador del alto nivel alcanzado. Todo este desarrollo queda

documentado en dos artículos, mostrados en la Tabla 7.1, a la espera de ser divulgados por la propia competición. La oportunidad de publicar un artículo supone un reconocimiento y reputación dentro de la comunidad científica. Además, permite compartir los avances y logros obtenidos que pueden derivar en nuevos descubrimientos.

Representation exploration and Deep learning applied to the early detection of pathological gambling risks
Xabier Larrayoz, Nuria Lebeña, Arantza Casillas and Alicia Pérez
OBSER-MENH at eRisk 2023: Deep Learning-Based Approaches for Symptom Detection in Depression and Early Identification of Pathological Gambling Indicators
Juan Martinez-Romo, Lourdes Araujo, Xabier Larrayoz, Maite Oronoz and Alicia Pérez

Tabla 7.1: Artículos presentados

7.1.3. Análisis de la desviación

Una vez terminado el proyecto se ha podido realizar una comparación entre el tiempo que se estimó y las horas necesarias para la realización de las tareas. La tabla 7.2 refleja la dificultad de realizar una estimación acorde al alcance del proyecto. Pese a cumplir todos y cada uno de los objetivos propuestos, dentro de los plazos acordados, se ha necesitado dedicar horas adicionales en cada una de las tareas. Algo previsible teniendo en cuenta, que se trata de un trabajo de investigación con la dificultad añadida de participar en una competición de renombre.

Fase	Horas estimadas	Horas empleadas	Desviación
Gestión	15	16	1
Planificación	10	10	0
Control de seguimiento	5	6	1
Desarrollo	185	220	35
Aprendizaje	30	40	10
Experimentación	155	180	25
Dominio de las técnicas	15	15	0
Implementación del sistema	120	145	25
Estudio de los resultados	20	20	0
Documentación	100	110	10
Memoria	80	90	10
Presentación y defensa	20	20	0
Total	300	346	46

Tabla 7.2: Estudio del tiempo requerido

7.1.4. Reflexión personal

En lo personal, este Trabajo de Fin de Grado ha supuesto un desafió en diferentes aspectos. El reto de participar en una competición de este nivel ha establecido el marco del programa, determinando los intervalos de desarrollo e investigación. Desde una etapa temprana, he aprendido a valorar la importancia de la planificación y organización.

La participación en una competición me ha permitido experimentar los desafíos adicionales que conlleva. Dicho entorno competitivo y exigente me ha permitido adaptarme y llevar a cabo un trabajo que ya de por sí era complicado.

En lo referente a la parte de investigación, este proyecto me ha permitido adentrarme en la vanguardia del campo del procesamiento del lenguaje natural. Ha sido una experiencia enriquecedora, ya que me ha enfrentado a los retos propios de cualquier investigación. El estudio constante de los últimos avances y tendencias me ha permitido ampliar mis conocimientos en esta área y me ha dado la capacidad de afrontar este reto. Toda esta experiencia me ha motivado para seguir profundizando en este ámbito.

7.2. Posibles mejoras y objetivos para el futuro

La limitación del tiempo a raíz de las fechas límite de la competición ha restringido el desarrollo de ciertas ideas y aspectos que requerían una mayor profundización. Por ejemplo, los resultados iniciales del rendimiento de algunas variaciones del sistema determinaron un descarte prematuro, abandonando el deseo de profundizar más en esos aspectos, como puede ser el planteamiento de agregar características adicionales mediante LDA o propiedades semánticas de la entrada. Otra perspectiva interesante para investigar sería entrenar la red utilizando dos tareas, clasificación y regresión. De esta forma, el modelo puede hacer un mejor uso de los datos y obtener una mejor generalización.

Sin embargo, a pesar de los posibles beneficios que podría aportar la aplicación de este modelo, es importante tener en cuenta las limitaciones éticas que existen, especialmente al trabajar en el ámbito de la salud y al utilizar las redes sociales como fuente de datos. Por eso mismo, en todo este trabajo se ha respetado el marco de trabajo propuesto en el artículo presentado en el primer taller sobre ética en el procesamiento del lenguaje natural [38].

Bibliografía

[1] Tianlin Zhang, Annika Schoene, Shaoxiong Ji, and Sophia Ananiadou. Natural language processing applied to mental illness detection: a narrative review. *NPJ Digital Medicine*, 5, 04 2022.

[2] David M. Blei. Introduction to probabilistic topic models. In *IEEE Signal Processing Magazine*, 2010.

[3] World Health Organization. Trastornos mentales. https://www.who.int/es/news-room/fact-sheets/detail/mental-disorders, 2022.

[4] Vankayala Tejaswini, Korra Sathya Babu, and Bibhudatta Sahoo. Depression detection from social media text analysis using natural language processing techniques and hybrid deep learning model. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, nov 2022. Just Accepted.

[5] Yuting Guo, Xiangjue Dong, Mohammed Al-Garadi, Abeed Sarker, Cécile Paris, and Diego Molla Aliod. Benchmarking of transformer-based pre-trained models on social media text classification datasets. *Australasian Language Technology Association*, 02 2021.

[6] Bogdan Ionescu, Henning Muller, Renaud Peteri, Johannes Ruckert, Asma Ben Abacha, Alba Garcia Seco de Herrera, Christoph M. Friedrich, Louise Bloch, Raphael Brüngel, Ahmad Idrissi-Yaghir, Henning Schäfer, Serge Kozlovski, Yashin Dicente Cid, Vassili Kovalev, Liviu-Daniel Stefan, Mihai Gabriel Constantin, Mihai Dogariu, Adrian Popescu, Jerome Deshayes-Chossart, Hugo Schindler, Jon Chamberlain, Antonio Campello, and Adrian Clark. Overview of the ImageCLEF 2022: Multimedia retrieval in medical, social media and nature applications. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, Proceedings of the 13th International Conference of the CLEF Association (CLEF 2022), Bologna, Italy, September 5-8 2022. LNCS Lecture Notes in Computer Science, Springer.

[7] Francisco Rodríguez-Sánchez, Jorge Carrillo-de Albornoz, Laura Plaza, Adrián Mendieta-Aragón, Guillermo Marco-Remón, Maryna Makeienko, María Plaza, Julio Gonzalo, Damiano Spina, and Paolo Rosso. Overview of exist 2022: sexism identification in social networks. *Procesamiento de Lenguaje Natural*, 69:229–240, 09 2022.

[8] Nicholas J Carson, Brian Mullin, Maria Jose Sanchez, Frederick Lu, Kelly Yang, Michelle Menezes, and Benjamin Lê Cook. Identification of suicidal behavior among psychiatrically hospitalized adolescents using natural language processing and machine learning of electronic health records. *PloS one*, 14(2):e0211116, 2019.

[9] Benjamin L Cook, Ana M Progovac, Pei Chen, Brian Mullin, Sherry Hou, and Enrique Baca-Garcia. Novel use of natural language processing (nlp) to predict suicidal ideation and psychiatric symptoms in a text-based mental health intervention in madrid. *Computational and mathematical methods in medicine*, 2016, 2016.

[10] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[11] Robert E Schapire. Explaining adaboost. In *Empirical inference*, pages 37–52. Springer, 2013.

[12] Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al. Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1):1–37, 2008.

[13] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, may 2017.

[15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[17] Davy Weissenbacher, Juan Banda, Vera Davydova, Darryl Estrada Zavala, Luis Gasco Sánchez, Yao Ge, Yuting Guo, Ari Klein, Martin Krallinger, Mathias Leddin, Arjun Magge, Raul Rodriguez-Esteban, Abeed Sarker, Lucia Schmidt, Elena Tutubalina, and Graciela Gonzalez-Hernandez. Overview of the seventh social media mining for health applications (#SMM4H) shared tasks at COLING 2022. In *Proceedings of The Seventh Workshop on Social Media Mining for Health Applications, Workshop & Shared Task*, pages 221–241, Gyeongju, Republic of Korea, October 2022. Association for Computational Linguistics.

[18] Adam Tsakalidis, Jenny Chim, Iman Bilal, Ayah Zirikly, Dana Atzil Slonim, Federico Nanni, Ps Resnik, Manas Gaur, Kaushik Roy, Becky Inkster, Jeff Leintz, and Maria Liakata. Overview of the clpsych 2022 shared task: Capturing moments of change in longitudinal user posts. In *Proceedings of the Eighth Workshop on Computational Linguistics and Clinical Psychology*, pages 184–198, Seattle, USA, 01 2022. Association for Computational Linguistics.

[19] Alba María Mármol-Romero, Salud María Jiménez-Zafra, Flor Miriam Plaza-Del-Arco, M. Dolores Molina-González, María-Teresa Martín-Valdivia, and Arturo Montejo-Ráez. Sinai at erisk@clef 2022: Approaching early detection of gambling and eating disorders with natural language processing. In *CEUR Workshop Proceedings*, volume 3180, pages 961–971. CEUR-WS, 2022.

[20] Ana-Maria Bucur, Adrian Cosma, and Liviu Dinu. Early risk detection of pathological gambling, self-harm and depression using bert. In *CEUR Workshop Proceedings*. CEUR-WS, 07 2021.

[21] Hermenegildo Fabregat, Andrés Duque, Lourdes Araujo, and Juan Martínez-Romo. Uned-nlp at erisk 2022: Analyzing gambling disorders in social media using approximate nearest neighbors. In *Conference and Labs of the Evaluation Forum*, 2022.

[22] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, page 604–613, New York, NY, USA, 1998. Association for Computing Machinery.

[23] Radim Rehurek and Petr Sojka. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2), 2011.

[24] Layla Oesper, Daniele Merico, Ruth Isserlin, and Gary D Bader. Wordcloud: a cytoscape plugin to create a visual semantic summary of networks. *Source code for biology and medicine*, 6(1):7, 2011.

[25] Justin Johnson and Taghi Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6:27, 03 2019.

[26] Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. Ammu : A survey of transformer-based biomedical pretrained language models, 2021.

[27] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[28] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[29] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[30] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information, 2017.

[31] Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder, 2018.

[32] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

[33] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. .ºReilly Media, Inc.", 2009.

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[35] Javier Parapar, Patricia Martín-Rodilla, David Losada, and Fabio Crestani. *Overview of eRisk 2022: Early Risk Prediction on the Internet*, pages 233–256. Springer-Verlag, 08 2022.

[36] Marcel Trotzek, Sven Koitka, and Christoph M. Friedrich. Utilizing neural networks and linguistic metadata for early detection of depression indications in text sequences. *IEEE Transactions on Knowledge and Data Engineering*, 32(3):588–601, mar 2020.

[37] Yining Wang, Liwei Wang, Yuanzhi Li, Di He, Tie-Yan Liu, and Wei Chen. A theoretical analysis of ndcg type ranking measures, 2013.

[38] Adrian Benton, Glen Coppersmith, and Mark Dredze. Ethical research protocols for social media health research. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 94–102, Valencia, Spain, April 2017. Association for Computational Linguistics.