Universidad de Valladolid

Escuela de Ingeniería Informática de Valladolid

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Mención Ingeniería del Software

Análisis Comparativo de Modelos de Aprendizaje Supervisado para el Reconocimiento de Emociones en Texto

Autor:

Izan Jiménez Girón

Tutor:

Joaquín Adiego Rodríguez

Cotutora:

Beatriz Juanes Mayfield

“La curiosidad es la mecha en la vela del aprendizaje.”
— William Arthur Ward

Agradecimientos

Quisiera expresar mi más sincero agradecimiento a Joaquín Adiego Rodríguez, tutor de este trabajo, por su orientación constante, su implicación y la claridad con la que ha sabido guiar cada etapa del proyecto. Su apoyo académico y humano ha sido fundamental para dar forma y sentido a esta investigación.

Extiendo también un agradecimiento especial a Beatriz Juanes Mayfield, cotutora de este trabajo, por brindarme la oportunidad de dar continuidad práctica a su trabajo previo y por inspirar parte de este proyecto con su visión.

A mi familia y seres queridos, gracias por su apoyo incondicional, por acompañarme en cada paso del camino y por recordarme siempre el valor de seguir aprendiendo con pasión y esfuerzo.

Finalmente, a todo el profesorado del Grado en Ingeniería Informática, por haber compartido no solo conocimientos, sino también el entusiasmo y la vocación por esta disciplina que tanto me ha motivado durante estos años.

Resumen

Hacer que una máquina pueda sentir emociones puede parecer hoy en día una meta lejana, casi imposible. Sin embargo, el primer paso para acercarnos a ello es dotarla de la capacidad de reconocerlas. Esta tarea, enmarcada dentro de la computación afectiva, ha sido objeto de estudio desde hace años, especialmente en su aplicación al texto, donde aún persisten numerosos desafíos.

Detectar emociones en lenguaje escrito implica enfrentarse a ambigüedades, dobles sentidos, ironías y matices culturales difíciles de codificar. Aunque ya existen modelos capaces de clasificar textos en categorías generales como “positivo”, “negativo” o “neutro”, estos sistemas aún están lejos de identificar con precisión emociones específicas como tristeza, alegría, miedo o enfado, especialmente en dominios desafiantes como las redes sociales, donde la informalidad, los contextos implícitos y el lenguaje figurado son comunes.

Este Trabajo Fin de Grado propone una aproximación exploratoria a este problema, combinando una revisión de las tecnologías actuales en Procesamiento del Lenguaje Natural (NLP) con una experimentación práctica. A través de la comparación de distintos modelos supervisados clásicos, se pretende evaluar sus capacidades en la tarea de clasificación emocional, identificar los límites actuales de estos sistemas y señalar posibles líneas de mejora futuras.

Abstract

Making a machine feel emotions may seem like an impossible goal today. However, the first step toward this challenge is enabling it to recognize them. This task, framed within affective computing, has been a subject of research for years—particularly when applied to textual data, where significant challenges remain.

Emotion detection in written language faces ambiguities, double meanings, irony, and cultural nuances that are difficult to model. While existing systems can classify text into broad sentiment categories such as “positive,” “negative,” or “neutral,” they are still far from accurately identifying specific emotions like sadness, joy, fear, or anger—especially in complex domains like social media, where informal and figurative language is prevalent.

This Final Degree Project presents an exploratory approach to this issue, combining a review of current Natural Language Processing (NLP) technologies with a practical evaluation. By comparing various classic supervised learning models, the project aims to assess their effectiveness in emotion classification tasks, understand their current limitations, and propose potential avenues for future improvement.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1. Introducción

1.1. Contexto y motivación

En los últimos años, el reconocimiento automático de emociones se ha consolidado como un campo de estudio estratégico dentro del ámbito de la inteligencia artificial, con aplicaciones que abarcan desde el marketing hasta la salud mental, pasando por la educación, la seguridad digital o el entretenimiento interactivo. Detectar emociones humanas a partir de datos, ya sean textos, imágenes, voz o señales fisiológicas, supone una herramienta de valor creciente para interpretar con mayor profundidad las interacciones humanas, en un mundo cada vez más digitalizado.

Este Trabajo de Fin de Grado se desarrolla como una continuación especializada y aplicada del Trabajo Fin de Máster “Análisis Emocional con Integración de IA en Proyectos de Cambio Organizacional en Empresas Familiares”, elaborado por Beatriz Juanes Mayfield. Aquel proyecto partía de un enfoque multisensorial, integrando imagen, audio y texto para analizar emociones en contextos estructurados y corporativos, como parte de procesos de transformación empresarial. Una de sus principales conclusiones fue la necesidad de desarrollar herramientas basadas en IA que complementen las metodologías tradicionales de evaluación emocional, especialmente en entornos de cambio organizacional.

Inspirado por esa reflexión, el presente trabajo decide especializarse en el análisis emocional exclusivamente a través del lenguaje textual, y concretamente desde una perspectiva más informal, cotidiana y espontánea: el lenguaje usado en redes sociales. Esta elección no solo permite acotar y enfocar técnicamente el estudio, sino que abre la puerta a explorar uno de los contextos más dinámicos y complejos en el tratamiento del lenguaje natural. En redes como Twitter, Instagram, TikTok o Reddit, el volumen de datos generados cada día es enorme, y el texto escrito refleja una riqueza emocional especialmente matizada, aunque difícil de interpretar: uso de jerga, emojis, memes, ironía, errores ortográficos y referencias culturales en constante evolución.

A partir de esta base, el objetivo del trabajo es evaluar la viabilidad de aplicar tanto modelos clásicos de *Machine Learning* como modelos de *Deep Learning* para detectar emociones en textos provenientes de estos entornos. En lugar de limitarse al análisis tradicional de sentimientos (positivo, negativo, neutro), se busca identificar emociones específicas como alegría, tristeza, miedo, enfado, sorpresa, entre otras.

Este tipo de tecnología tiene aplicaciones prácticas inmediatas y muy diversas. Algunos ejemplos de ello son:

- **Monitorización emocional de campañas de marketing:** comprender cómo reacciona emocionalmente el público ante anuncios, productos o marcas, permitiendo adaptar estrategias de comunicación más efectivas basadas en emociones específicas, y no solo en polaridades.
- **Moderación emocional de contenidos:** detección temprana de contenidos agresivos, discursos de odio o publicaciones con carga emocional negativa, contribuyendo a la creación de entornos digitales más seguros y saludables.
- **Videojuegos adaptativos:** en entornos interactivos, como videojuegos, las emociones detectadas en el texto pueden usarse para modificar dinámicamente la narrativa, la dificultad o la música, generando experiencias más personalizadas y envolventes.
- **Análisis de guiones y críticas culturales:** detectar emociones dominantes en guiones, reseñas o críticas escritas permite estudiar la percepción emocional de obras culturales, y puede aportar valor en campos como el cine, la literatura o el diseño de experiencias.

Así, este estudio no solo se plantea como una prueba de concepto técnica, sino como una exploración aplicada, consciente de los retos que implica trabajar con lenguaje informal. Frente a la creciente sofisticación del *Deep Learning*, se valorará también la eficacia de modelos más ligeros y accesibles de *Machine Learning* tradicional, especialmente adecuados para proyectos de pequeña o mediana escala.

En definitiva, este TFG busca profundizar en un enfoque muy concreto del análisis emocional mediante inteligencia artificial: la detección de emociones en texto coloquial procedente de redes sociales. Esta línea de trabajo parte de una reflexión más amplia sobre la necesidad de desarrollar nuevas herramientas tecnológicas capaces de captar aquello que tradicionalmente ha sido difícil de medir: la emoción humana en su forma más viva, ambigua y cambiante.

1.2. Objetivos

Este Trabajo de Fin de Grado tiene como objetivo principal evaluar la viabilidad de aplicar inteligencia artificial al reconocimiento de emociones en texto, concretamente en el contexto del lenguaje informal propio de las redes sociales. Para ello, se plantean una serie de objetivos que permitirán abordar el problema desde un enfoque técnico y progresivo.

Objetivos generales

- Introducirse en el campo del reconocimiento emocional en texto desde una perspectiva de inteligencia artificial, aprendiendo sus fundamentos teóricos y prácticos.
- Buscar, comparar y seleccionar sets de datos adecuados para tareas de análisis emocional, priorizando aquellos que contengan lenguaje coloquial, etiquetas claras y volumen suficiente para experimentar.
- Explorar diferentes enfoques de inteligencia artificial aplicables al problema, entre los que se incluyen *Machine Learning* y *Deep Learning*, analizando sus ventajas, inconvenientes, limitaciones y requisitos, con el fin de determinar cuál es más apropiado para este caso concreto.
- Aprender a utilizar herramientas del ecosistema Python, como Scikit-learn, Pandas o PyTorch, fundamentales para el desarrollo de soluciones basadas en inteligencia artificial.
- Implementar y entrenar distintos modelos tanto de *Machine Learning* clásico como de *Deep Learning*, adaptando la selección de algoritmos a las características del problema y de los datos disponibles, con el fin de analizar su comportamiento y comparar su rendimiento en tareas de clasificación emocional.
- Evaluar cuantitativamente los resultados obtenidos, utilizando métricas adecuadas para analizar el rendimiento, la estabilidad y la eficiencia de los distintos enfoques implementados.
- Concluir si es viable aplicar modelos de IA al reconocimiento de emociones en este tipo de datos y condiciones, y justificar la elección final del enfoque más adecuado.
- A partir de los resultados obtenidos, identificar líneas de mejora, así como posibles vías de desarrollo futuro que permitan perfeccionar el uso de inteligencia artificial para el reconocimiento emocional en texto.

1.3. Organización y estructura de la memoria

La memoria de este Trabajo de Fin de Grado se encuentra organizada en los siguientes capítulos:

- **Capítulo 1. Introducción:** Se presenta el contexto general del problema, la motivación del trabajo, los objetivos planteados y una visión global de la estructura del documento.
- **Capítulo 2. Planificación:** Describe la planificación temporal del proyecto, incluyendo la organización de tareas, cronograma estimado y metodología seguida para la ejecución del trabajo.
- **Capítulo 3. Estado del arte:** Se exploran los enfoques actuales en el reconocimiento emocional en texto, revisando proyectos e investigaciones previas relacionadas con este campo. Además, se identifican las principales líneas de trabajo, tendencias existentes y normas legales.
- **Capítulo 4. Diseño:** Se presenta la estructura del sistema, incluyendo los requisitos del proyecto, la organización modular del código y los principales casos de uso. Se incluyen los diagramas necesarios.
- **Capítulo 5. Búsqueda y selección de datos:** Se detalla el proceso de búsqueda, comparación y selección de conjuntos de datos adecuados para el análisis emocional, valorando criterios como el tipo de emociones etiquetadas, volumen de datos, lenguaje utilizado y adecuación al dominio de redes sociales.
- **Capítulo 6. Elección de tecnologías:** Se justifican las herramientas y librerías utilizadas en el desarrollo del trabajo. Se argumenta su necesidad en función de los requisitos del proyecto y se detallan sus funcionalidades clave.
- **Capítulo 7. Análisis de modelos y resultados:** Se explican los fundamentos y configuraciones clave de cada modelo aplicado, detallando sus parámetros principales. Posteriormente, se presentan y comparan los resultados obtenidos, analizando el rendimiento y comportamiento de cada uno de los modelos. Por último, se extraen las principales conclusiones derivadas del análisis.
- **Capítulo 8. Conclusiones y trabajo futuro:** A partir de los resultados obtenidos, se identifican posibles mejoras y direcciones de desarrollo futuro que permitan ampliar el alcance del trabajo o perfeccionar el rendimiento de los modelos evaluados.
- **Bibliografía:** Se referencian de las fuentes de información utilizadas durante el proyecto.

Capítulo 2. Planificación

Este capítulo describe la planificación y organización del proyecto, combinando la metodología CRISP-DM (*Cross-Industry Standard Process for Data Mining*), ampliamente utilizada en proyectos de análisis de datos, con el enfoque ágil de Scrum, que permite estructurar el trabajo de forma iterativa y adaptable. Esta integración metodológica ha permitido abordar las fases clásicas de un proyecto de ciencia de datos con una gestión ágil y orientada a resultados, facilitando el seguimiento y adaptación continua del Trabajo de Fin de Grado.

2.1. Metodología CRISP-DM

La metodología CRISP-DM [62] es un modelo de proceso ampliamente utilizado en proyectos de minería de datos y ciencia de datos. Fue diseñado para ser independiente de herramientas, industrias o aplicaciones específicas, y se caracteriza por ser flexible, iterativo y cíclico, permitiendo adaptar el flujo de trabajo a las necesidades reales del proyecto.

CRISP-DM se estructura en seis fases principales, que no se siguen necesariamente de forma estrictamente secuencial, ya que muchas veces el trabajo requiere iteraciones y retrocesos. A continuación, se describen cada una de las fases que se aprecian en la figura 2.1:

- **Primera etapa, comprensión del negocio:** Esta fase se centra en entender profundamente el problema que se desea resolver, desde una perspectiva estratégica o de negocio. No se trata aún de analizar datos, sino de clarificar los objetivos que motivan el proyecto, identificar las necesidades del usuario o del contexto de aplicación, y definir los criterios que permitirán valorar el éxito del trabajo. La finalidad es traducir esas metas generales en objetivos técnicos concretos que puedan guiar el análisis posterior.
- **Segunda etapa, comprensión de los datos:** Una vez definidos los objetivos del proyecto, se inicia la exploración de los datos disponibles. Esta etapa tiene como propósito familiarizarse con el contenido, calidad y estructura del conjunto de datos. Se analizan aspectos como la cantidad de información, la presencia de valores nulos, la distribución de las variables y posibles anomalías. También se pueden generar visualizaciones y estadísticas descriptivas que ayuden a extraer un primer conocimiento sobre los datos y a orientar las decisiones de preparación futura.
- **Tercera etapa, preparación de los datos:** Esta es una de las fases más técnicas y exigentes del proceso. Consiste en transformar, limpiar y estructurar los datos para que estén listos para ser utilizados en los modelos de análisis. Las tareas comunes incluyen

la eliminación de registros irrelevantes, el tratamiento de valores atípicos, la conversión de formatos, la creación de nuevas variables, la codificación de categorías o la normalización de datos. El objetivo es obtener un conjunto de datos fiable, coherente y representativo del problema a estudiar.

- **Cuarta etapa, modelado:** En esta fase se seleccionan y aplican algoritmos estadísticos o de aprendizaje automático para construir modelos que permitan resolver el problema planteado. La elección del modelo dependerá del tipo de datos y de los objetivos definidos previamente. También es habitual realizar pruebas con distintos algoritmos y ajustar sus parámetros. Esta etapa requiere conocimientos técnicos específicos y una comprensión adecuada del comportamiento de los modelos utilizados.
- **Quinta etapa, evaluación:** Una vez entrenado el modelo, es fundamental evaluar su rendimiento. Esta fase no solo implica medir métricas cuantitativas, sino también interpretar los resultados a nivel cualitativo. Se analiza si el modelo cumple con los objetivos del negocio definidos en la primera etapa, si hay patrones erróneos o inesperados y si es necesario volver a fases anteriores para ajustar el enfoque. La evaluación es clave para garantizar que los resultados obtenidos son válidos y útiles.
- **Sexta etapa, despliegue:** Finalmente, se prepara la implementación de los resultados del proyecto. Esta etapa varía dependiendo del contexto: puede implicar integrar el modelo en un sistema productivo, generar un informe con conclusiones o simplemente documentar el trabajo para su consulta futura. Lo importante es que los hallazgos sean accesibles y aplicables para los usuarios finales o tomadores de decisiones. En contextos académicos, esta fase suele centrarse en la presentación formal de los resultados y la justificación metodológica del trabajo realizado.

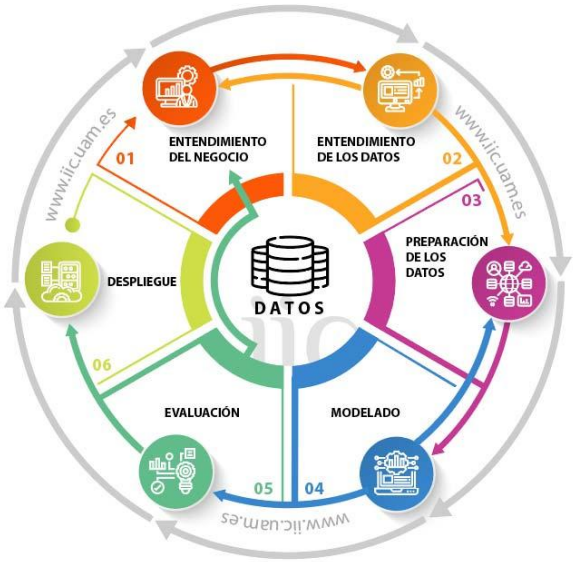

Figura 2.1. Etapas metodología CRISP-DM. Fuente [63]

2.2. Marco de trabajo Scrum

Scrum [64] es un marco de trabajo ágil orientado a la gestión y desarrollo de productos complejos, muy utilizado en el ámbito del software y, más recientemente, en la ciencia de datos. Su enfoque se basa en ciclos iterativos e incrementales llamados *sprints*, en los que se planifica, ejecuta, revisa y ajusta el trabajo con una alta frecuencia. Esta dinámica permite una mayor adaptabilidad ante cambios, mejora la organización del equipo y fomenta la entrega continua de valor.

A diferencia de las metodologías tradicionales, Scrum no impone una secuencia rígida de tareas, sino que propone una estructura flexible basada en ciclos iterativos. Aunque en su forma original contempla varios roles diferenciados, en proyectos académicos realizados de forma individual, como es el caso de este trabajo, todos los roles son asumidos por una única persona. La planificación y seguimiento del proyecto se ha gestionado principalmente a través de un *product backlog*, que contiene y prioriza todas las tareas relevantes. A partir de este backlog se han organizado los distintos *sprints*, lo que ha permitido mantener una visión clara de los objetivos y adaptarse progresivamente a los avances del desarrollo.

El uso de Scrum aporta una estructura de trabajo adaptable que facilita el seguimiento de tareas, el control del tiempo y la organización de objetivos intermedios, lo cual es especialmente útil en proyectos de investigación y desarrollo como los basados en ciencia de datos.

La integración de ambos modelos consiste en utilizar las fases de CRISP-DM como base metodológica global del proyecto, y organizar el avance dentro de cada fase mediante *sprints* definidos por Scrum.

Esta combinación permite beneficiarse de la claridad estructural de CRISP-DM sin renunciar a la agilidad y capacidad de adaptación que aporta Scrum. Así, se logra una gestión eficaz del tiempo y de los recursos, al tiempo que se garantiza un desarrollo coherente, controlado y bien documentado.

2.3. Descripción y planificación de los sprints

A continuación, se describen en detalle los distintos *sprints* en los que se ha dividido el desarrollo del proyecto:

Sprint 1: Comprensión del negocio (18 - 28 de febrero)

En esta fase inicial, el objetivo es entender el propósito general del proyecto, establecer los objetivos principales y definir el enfoque que se seguirá. En el contexto académico, esto implica también coordinar las expectativas con el tutor del TFG y delimitar claramente el problema a resolver. Durante este período se mantuvo una primera reunión de trabajo para acordar la dirección del proyecto, así como para revisar documentos de referencia, como el TFM previo utilizado como guía metodológica. Esta etapa constituye la base conceptual sobre la que se construirá el resto del proyecto.

Sprint 2: Comprensión de los datos (1 – 24 de marzo)

Una vez definidos los objetivos, se procede al análisis preliminar de los datos disponibles. Esta etapa corresponde al momento en que se explora el estado del arte del análisis emocional y del procesamiento del lenguaje natural (NLP), así como a la búsqueda, evaluación y selección del conjunto de datos principal. El proceso incluyó la revisión de diferentes fuentes y la elección de un corpus que presentara una adecuada cobertura emocional, calidad de etiquetado y volumen. El análisis inicial de estos datos permite identificar posibles limitaciones y oportunidades para el modelado posterior.

Sprint 3 - Preparación de los datos (25 de marzo – 7 de abril)

Aquí se realiza la transformación del conjunto de datos seleccionado para su uso en los modelos de aprendizaje automático. Esta fase implica tareas como limpieza, normalización, eliminación de ruido textual, y el formateo adecuado de las etiquetas. La preparación de los datos es una de las fases más exigentes del proceso, ya que de su calidad dependerá en gran medida el rendimiento de los modelos. Con los datos listos, se sientan las bases técnicas para iniciar el desarrollo experimental.

Sprint 4 – Modelado (8 - 21 de abril)

Una vez que los datos están correctamente preparados, se abordan distintas estrategias de modelado. Esta etapa se divide en varias fases internas: en primer lugar, se implementan modelos clásicos como regresión logística y Naive Bayes, que permiten establecer una línea base de rendimiento. Posteriormente, se introducen modelos más sofisticados como Random Forest y SVM, con el objetivo de explorar mayores niveles de precisión. Finalmente, se incorpora un modelo basado en *deep learning*, concretamente BERT, lo que permite comparar el rendimiento de técnicas tradicionales frente a enfoques modernos de procesamiento de lenguaje natural. Cada uno de estos modelos fue ajustado y evaluado dentro de su propio sprint, siguiendo principios iterativos de mejora.

Sprint 5 - Evaluación (22 de abril – 19 de mayo)

Una vez construidos los modelos, se realiza una evaluación comparativa entre ellos. En esta fase se aplican métricas cuantitativas como precisión, *recall* y *F1-score*, además de realizar un análisis cualitativo de los errores. El propósito es determinar cuál de los enfoques ofrece mejores resultados para el análisis emocional y justificar, en términos técnicos, la elección de un modelo final. Este proceso de evaluación también permite validar si los objetivos definidos en la fase de comprensión del negocio han sido alcanzados.

Fase final – Despliegue / Comunicación de resultados (20 de mayo – 6 de julio)

En un TFG académico, la fase de despliegue se traduce en la redacción y entrega del trabajo final. Aquí se integran todos los elementos del proyecto: la fundamentación teórica, las decisiones técnicas, los resultados obtenidos y su análisis. También se contempla la revisión y validación del trabajo por parte del tutor, así como las tareas de corrección de estilo, maquetación y generación del documento final en formato PDF. Esta etapa asegura que todo el conocimiento generado durante el desarrollo del proyecto se comunique de forma clara y rigurosa.

2.4. Diagrama de Gantt

Para una mejor comprensión de la planificación temporal del proyecto, en la figura 2.2 se presenta un diagrama de Gantt que recoge de forma visual todas las tareas previstas, estructuradas en fases o *sprints*, junto con su duración estimada.

Nombre	Fecha de inicio	Fecha de fin	Duración
Reconocimiento de emociones en texto	18/2/25	15/6/25	118
Fase 1	18/2/25	2/3/25	13
Definición de objetivos y alcance del proyecto	18/2/25	20/2/25	3
Revisión del TFM de referencia	21/2/25	25/2/25	5
Revisión preliminar del estado del arte	26/2/25	2/3/25	5
Fase 2	5/3/25	26/3/25	22
Estudio de IA, NLP y análisis emocional	5/3/25	13/3/25	9
Búsqueda y selección del dataset principal	14/3/25	22/3/25	9
Exploración de librerías Python	23/3/25	26/3/25	4
Fase 3	27/3/25	2/4/25	7
Limpieza y normalización del dataset	27/3/25	28/3/25	2
Procesamiento de etiquetas y estructuras	29/3/25	1/4/25	4
Configuración del entorno de trabajo y repositorio	2/4/25	2/4/25	1
Fase 4	3/4/25	29/4/25	27
Implementación de Regresión logística	3/4/25	7/4/25	5
Implementación de Naive Bayes	8/4/25	12/4/25	5
Implementación de Random Forest	13/4/25	17/4/25	5
Implementación de SVM	18/4/25	22/4/25	5
Implementación de BERT	23/4/25	29/4/25	7
Fase 5	30/4/25	19/5/25	20
Entrenamiento de modelos	30/4/25	9/5/25	10
Evaluación de resultados	10/5/25	14/5/25	5
Comparativa de resultados	15/5/25	19/5/25	5
Fase 6	20/5/25	15/6/25	27
Redacción técnica y académica del documento	20/5/25	8/6/25	20
Validación por parte del tutor	9/6/25	13/6/25	5
Corrección de estilo, maquetación y entrega	14/6/25	15/6/25	2

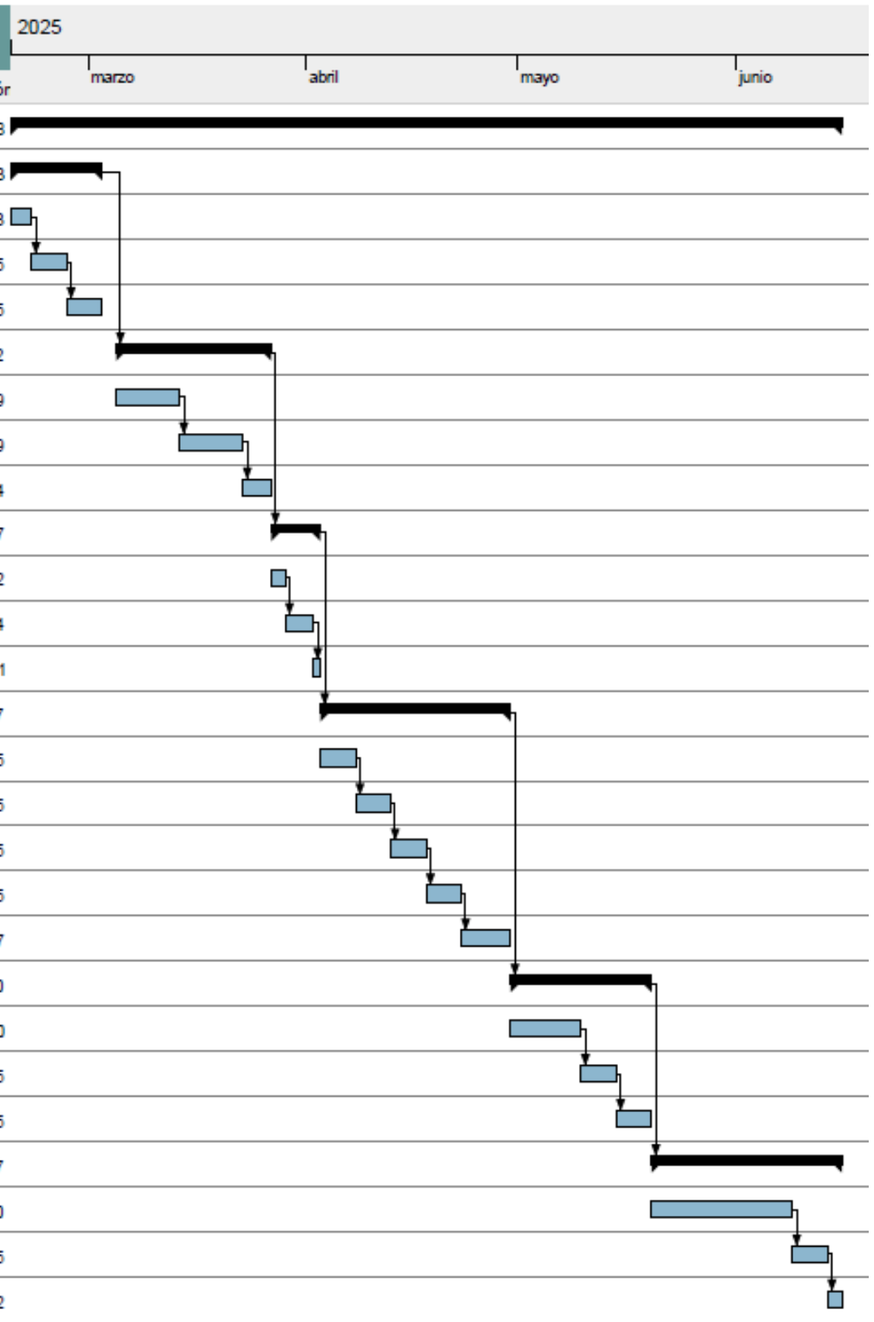

Figura 2.2. Diagrama de Gantt del proyecto. Realizado con GanttProject [23]

Es importante señalar que no se han planificado tareas en paralelo. Esto se debe a que el trabajo es realizado íntegramente por una única persona, lo que condiciona la ejecución secuencial de las actividades. Aun así, resulta conveniente elaborar un diagrama de Gantt, ya que permite visualizar con claridad la secuencia temporal de las tareas a realizar y facilita el seguimiento del progreso global del proyecto.

2.5. Herramientas para la gestión Agile

Durante el desarrollo del Trabajo de Fin de Grado, se han empleado diferentes herramientas digitales con el objetivo de facilitar la planificación, organización y seguimiento del proyecto. Aunque la planificación no ha sido excesivamente estructurada en su inicio, se ha hecho uso de varias aplicaciones para mejorar progresivamente la gestión del tiempo y las tareas. Las herramientas principales han sido:

Microsoft Teams [1]

Utilizado como plataforma principal de comunicación con el tutor del TFG. A través de Teams se han realizado las reuniones periódicas para revisar el progreso del trabajo, resolver dudas y establecer próximos objetivos. Ha sido clave para mantener una supervisión continua y organizada.

Google Calendar [2]

Esta herramienta ha servido como agenda personal para fijar fechas clave del proyecto: reuniones, entregas parciales, hitos importantes, etc. Ha permitido tener una visión global del calendario del TFG, así como establecer recordatorios de tareas pendientes.

GitLab

Aunque no se ha utilizado de forma intensiva como plataforma de control de versiones, GitLab se ha empleado para almacenar el código de forma segura, versionarlo cuando ha sido necesario y tener un repositorio ordenado y accesible. Esto permite trazabilidad del desarrollo técnico y mejora la organización del código fuente.

Se puede acceder al repositorio mediante la referencia [3].

2.6. Análisis de riesgos

El PMBOK [4], estándar elaborado por el PMI (*Project Management Institute*), establece un marco de buenas prácticas para la gestión de proyectos, incluyendo la gestión de riesgos como un área clave. Según este estándar, un riesgo es cualquier evento incierto que puede afectar positiva o negativamente a los objetivos del proyecto.

La gestión de riesgos es fundamental para anticipar problemas, reducir incertidumbre y mejorar la toma de decisiones, especialmente en proyectos complejos o innovadores, como los que integran inteligencia artificial o cambios organizativos. El PMBOK propone identificar, analizar, planificar respuestas y hacer seguimiento de los riesgos desde las primeras etapas del proyecto, permitiendo así una ejecución más controlada y eficaz.

Para calcular el grado de cada riesgo se ha hecho uso de la matriz de riesgos, mostrada en la tabla 2.1

Probabilidad/Impacto	Bajo	Medio	Alto
Baja	Bajo	Bajo	Medio
Media	Bajo	Medio	Alto
Alta	Medio	Alto	Alto

Tabla 2.1. Matriz de riesgos utilizada

Dicho esto, durante el desarrollo del proyecto se han identificado una serie de riesgos potenciales que podrían haber afectado negativamente al avance y consecución de los objetivos. A continuación, se enumeran los más relevantes, junto con su posible impacto y las medidas tomadas para mitigarlos o responder ante ellos:

Riesgo 1	*Retraso en el inicio del proyecto*
Descripción	El TFG comenzó más tarde de lo habitual en relación con la fecha de entrega establecida, reduciendo el margen total.
Probabilidad	Alta
Impacto	Medio
Grado del riesgo	Alto
Prevención	Establecer una planificación ajustada desde el inicio.
Mitigación	Reorganización del calendario y priorización de tareas clave.

Tabla 2.2. Riesgo 1: retraso del proyecto

Riesgo 2	*Falta de experiencia previa en inteligencia artificial*
Descripción	Primer contacto real con técnicas de IA, así como con tecnologías para su implementación.
Probabilidad	Baja
Impacto	Alto
Grado del riesgo	Medio
Prevención	Dedicación inicial a formación autodidacta.
Mitigación	Aprendizaje progresivo y aplicación práctica en tareas reales del TFG.

Tabla 2.3. Riesgo 2: falta de experiencia

Riesgo 3	*Limitaciones técnicas y computacionales*
Descripción	No se dispone de recursos potentes para entrenar modelos complejos.
Probabilidad	Alta
Impacto	Bajo
Grado del riesgo	Medio
Prevención	Elegir técnicas acordes a los recursos disponibles.
Mitigación	Uso de modelos ligeros y pruebas limitadas con modelos potentes.

Tabla 2.4. Riesgo 3: limitaciones

Riesgo 4	*Dificultad para encontrar conjuntos de datos adecuados*
Descripción	Encontrar conjuntos de datos numerosos, de calidad y que contengan las emociones requeridas para el proyecto resulta complicado.
Probabilidad	Baja
Impacto	Alto
Grado del riesgo	Medio
Prevención	Explorar múltiples fuentes desde el inicio y definir bien los criterios de búsqueda.
Mitigación	Se compararon diversas opciones antes de seleccionar la más adecuada.

Tabla 2.5. Riesgo 4: dificultades con los datos

Riesgo 5	*Dificultades con las métricas de evaluación*
Descripción	Requiere comprender a fondo el significado de cada métrica y su interpretación en el contexto del proyecto.
Probabilidad	Baja
Impacto	Alto
Grado del riesgo	Medio
Prevención	Estudio previo de las métricas más comunes en clasificación.
Mitigación	Se realiza un esfuerzo por interpretar correctamente las métricas más relevantes y presentar los resultados con claridad.

Tabla 2.6. Riesgo 5: dificultades con las métricas

Riesgo 6	*Coordinación con el tutor y cotutores*
Descripción	Dificultad para encontrar horarios compatibles entre varias personas involucradas.
Probabilidad	Medio
Impacto	Alto
Grado del riesgo	Alto
Prevención	Uso de herramientas como Teams para prever agendas.
Mitigación	Planificación anticipada de reuniones y mayor autonomía en las diferentes fases.

Tabla 2.7. Riesgo 6: coordinación con tutor y cotutores

Riesgo 7	*Carga de trabajo adicional al proyecto*
Descripción	Paralelamente se cursa la asignatura Sistemas Empotrados y se realizan prácticas en empresa con un contrato de 6 horas diarias.
Probabilidad	Alto
Impacto	Medio
Grado del riesgo	Alto
Prevención	Evaluar desde el inicio la carga total y planificar un calendario realista, priorizando hitos clave del TFG.
Mitigación	Establecer una buena organización del tiempo, dividir las tareas por bloques semanales y mantener comunicación constante con el tutor.

Tabla 2.8. Riesgo 7: carga de trabajo

Riesgo 8	*Enfermedad o indisponibilidad del trabajador*
Descripción	Existe la posibilidad de que el estudiante sufra una enfermedad o cualquier situación personal que le impida trabajar temporalmente.
Probabilidad	Baja
Impacto	Alto
Grado del riesgo	Medio
Prevención	Mantener hábitos saludables y procurar una buena organización personal que reduzca el estrés y la sobrecarga.
Mitigación	Establecer un margen de seguridad en la planificación y priorizar tareas clave con antelación para minimizar el impacto en caso de baja temporal.

Tabla 2.9. Riesgo 8: enfermedad o indisponibilidad

2.7. Costes

El desarrollo de este Trabajo Fin de Grado ha implicado una planificación meticulosa y una considerable inversión tanto de recursos humanos como materiales. A continuación, se presenta un desglose detallado de ambos tipos de costes asociados al proyecto.

Costes materiales

Dado que este proyecto se enmarca en un Trabajo Fin de Grado de carácter académico, no ha supuesto ningún coste económico directo. Todas las herramientas empleadas durante su desarrollo han sido de uso gratuito, incluyendo entornos de desarrollo como Visual Studio Code, lenguajes y librerías de código abierto como Python, así como plataformas de trabajo colaborativo como GitLab, Microsoft Teams y Google Calendar, entre otras.

Del mismo modo, los datos utilizados para entrenar y evaluar los modelos proceden de fuentes públicas y abiertas, por lo que no ha sido necesario realizar ningún gasto en adquisición de sets de datos ni licencias de uso.

El único coste material que podría considerarse es el del equipo informático utilizado. En concreto, se ha empleado un ordenador de sobremesa (ver apartado 7.6.1) con un precio de adquisición de 1.584 €, comprado con anterioridad y destinado a un uso personal. Si se asume una vida útil estimada de 8 años (96 meses), y que se ha utilizado de forma exclusiva durante 2 meses para este proyecto, se puede calcular un coste proporcional del uso del equipo de la siguiente manera:

$$Amortización\ de\ Hardware = \frac{1584€}{96\ meses} \times 2\ meses = 33\ €$$

Este valor de 33 € representa un coste teórico aproximado del uso del equipo durante el tiempo dedicado al TFG. A efectos prácticos, sin embargo, el coste material puede considerarse cero, ya que no ha sido necesario adquirir ningún recurso físico adicional específicamente para el desarrollo del proyecto.

Costes humanos

Se ha estimado el valor económico del trabajo realizado en este proyecto académico en función de los perfiles profesionales que, en un entorno real, asumirían las distintas fases del desarrollo. Para ello, se han tomado como referencia sueldos anuales medios en España, según Glassdoor [68]:

- Data Scientist: 39.000 €/año [69]
- Ingeniero de IA / Machine Learning Engineer: 36.000 €/año [70]
- Data Analyst: 27.000 €/año [71]

A continuación, en la tabla 2.10, se calculan los costes estimados:

Fase del proyecto	*Perfil profesional*	*Duración estimada (h)*	*Tarifa (€/h)*	*Coste estimado (€)*
Preparación de datos	Data Scientist	40 h	18,75 €	750,00 €
Modelado	Ingeniero de IA / ML	60 h	17,31 €	1.038,60 €
Evaluación de resultados	Data Scientist	105 h	18,75 €	1.968,75 €
Redacción y entrega	Data Analyst	95 h	12,98 €	1.233,10 €
Total estimado		300 h		4.990,45 €

Tabla 2.10. Tabla resumen de los costes humanos

Esta estimación de costes permite simular cuánto supondría económicamente desarrollar un proyecto similar en un entorno profesional, donde intervendrían perfiles especializados. Aunque en este caso todas las tareas han sido asumidas por una única persona en un contexto académico, sin experiencia profesional específica en cada área, este desglose resulta útil para dimensionar el valor del trabajo realizado y comprender qué recursos serían necesarios en un entorno corporativo real.

Capítulo 3. Estado del arte

3.1. Introducción al problema

La correcta gestión emocional en entornos profesionales se ha convertido en un factor clave para el éxito de procesos como la toma de decisiones, la adaptación al cambio o la mejora del clima organizacional. Esta creciente sensibilidad hacia el papel de las emociones ha impulsado el interés por desarrollar herramientas que permitan analizarlas de forma más objetiva, continua y automatizada.

Tradicionalmente, la evaluación emocional se ha basado en técnicas como entrevistas [65], encuestas o la observación directa. Sin embargo, estos métodos presentan importantes limitaciones: requieren intervención humana, son difícilmente escalables, y están condicionados por la subjetividad tanto del evaluador como del evaluado. Además, su carácter puntual dificulta capturar la naturaleza dinámica y cambiante de las emociones.

En este contexto, la inteligencia artificial y, en particular, el procesamiento del lenguaje natural (NLP), han abierto nuevas posibilidades para el análisis emocional basado en texto. Los modelos de aprendizaje automático y aprendizaje profundo permiten detectar patrones emocionales en mensajes escritos sin necesidad de interacción directa, lo que reduce el sesgo humano y facilita el análisis en tiempo real. Este enfoque resulta especialmente prometedor en entornos donde las interacciones se producen en formato textual, como redes sociales, plataformas colaborativas o sistemas de mensajería digital.

Este trabajo se inscribe en esta línea de investigación, centrando su atención en el análisis emocional de textos procedentes de contextos informales y coloquiales. El objetivo es explorar la viabilidad de aplicar modelos existentes de inteligencia artificial a este tipo de lenguaje, caracterizado por su variabilidad, riqueza expresiva y matices culturales, con el fin de evaluar su rendimiento y utilidad en escenarios reales.

3.2. Antecedentes

Un antecedente de especial relevancia para este Trabajo de Fin de Grado es el Trabajo Fin de Máster titulado “Análisis Emocional con Integración de IA en Proyectos de Cambio Organizacional en Empresas Familiares” [5], realizado por Beatriz Juanes Mayfield en el Máster en Dirección de Proyectos de la Universidad de Valladolid. Este estudio aborda de manera transversal la problemática de la gestión emocional en los procesos de cambio organizacional, poniendo el foco en un tipo de entidad particularmente sensible a estos desafíos: la empresa familiar. En este contexto, se reflexiona sobre cómo los factores emocionales influyen en la toma de decisiones, el liderazgo, la resistencia al cambio o la comunicación interna.

El TFM parte de una hipótesis clara: las herramientas tradicionales para evaluar emociones como las entrevistas estructuradas, cuestionarios u observación directa, son valiosas pero insuficientes, ya que dependen en gran medida de la percepción subjetiva del evaluador y del contexto en el que se aplican. Además, tienden a ofrecer una visión estática, puntual y a menudo sesgada del estado emocional del individuo o grupo. Por ello, el trabajo propone la incorporación de técnicas de inteligencia artificial como medio para aportar una mayor objetividad, continuidad y precisión en la medición emocional, especialmente en entornos de transformación estratégica.

Uno de los principales aportes del trabajo radica en la presentación de un enfoque multimodal de detección emocional, que combina diferentes fuentes sensoriales: texto, imagen y audio. A través de una revisión exhaustiva de la literatura, se describen las tres líneas principales de análisis:

- NLP (*Natural Language Processing*), para la detección de emociones a partir de texto escrito.
- FER (*Facial Emotion Recognition*), mediante el reconocimiento de expresiones faciales en imágenes o vídeo.
- SER (*Speech Emotion Recognition*), a partir del análisis del tono, la prosodia y otras características del habla.

Aunque el TFM no desarrolla ni implementa una solución técnica completa, sí plantea una propuesta metodológica sólida, diseñada para servir como base en proyectos posteriores. Se detallan aspectos como los posibles modelos de referencia, las arquitecturas viables, las fuentes de datos, las estrategias de validación y las consideraciones éticas necesarias al trabajar con datos sensibles. El trabajo también analiza diferentes escenarios de aplicación, subrayando la utilidad de estas tecnologías para anticipar bloqueos emocionales, identificar resistencias al cambio o reforzar el liderazgo empático dentro de la empresa.

En el capítulo 3 de este TFM, en lo relativo al canal textual, uno de los más ricos y complejos en términos de contenido emocional, el Trabajo Fin de Máster de Beatriz Juanes Mayfield

expone cómo las tecnologías de inteligencia artificial, y en particular el Procesamiento del Lenguaje Natural (NLP) [6], permiten extraer información significativa a partir de grandes volúmenes de datos no estructurados. En un contexto en el que más del 80% de los datos generados no tienen una forma fija, pueden ser de correos electrónicos, comentarios en redes sociales [7], encuestas, etc. se hace imprescindible aplicar técnicas avanzadas que posibiliten el análisis automático de sentimientos, intenciones o emociones subyacentes en el lenguaje escrito.

El TFM profundiza en el estado actual de estas tecnologías, destacando el uso de algoritmos de aprendizaje profundo para interpretar patrones semánticos y lingüísticos que reflejan estados emocionales. Este enfoque permite, por ejemplo, identificar emociones como alegría, tristeza, sorpresa o rechazo en textos de diversa naturaleza, desde opiniones de clientes hasta comunicaciones internas en organizaciones. A través de un proceso estructurado que incluye el preprocesamiento del texto [8] [9], la extracción de características y la posterior clasificación emocional, se facilita una comprensión más profunda del contenido emocional implícito.

Tal como se recoge en el TFM en la figura 3.1 se muestran ejemplos concretos de cómo se forman los vectores que permiten representar palabras de forma numérica en el espacio semántico. Y en la figura 3.2 se observa la proximidad semántica entre distintas palabras representadas en un espacio vectorial n-dimensional.” Esta representación permite que los algoritmos de procesamiento de lenguaje natural (NLP) analicen las palabras no solo por su forma o frecuencia, sino por las relaciones de significado que existen entre ellas, lo cual es fundamental para tareas como la detección de emociones a partir del texto.

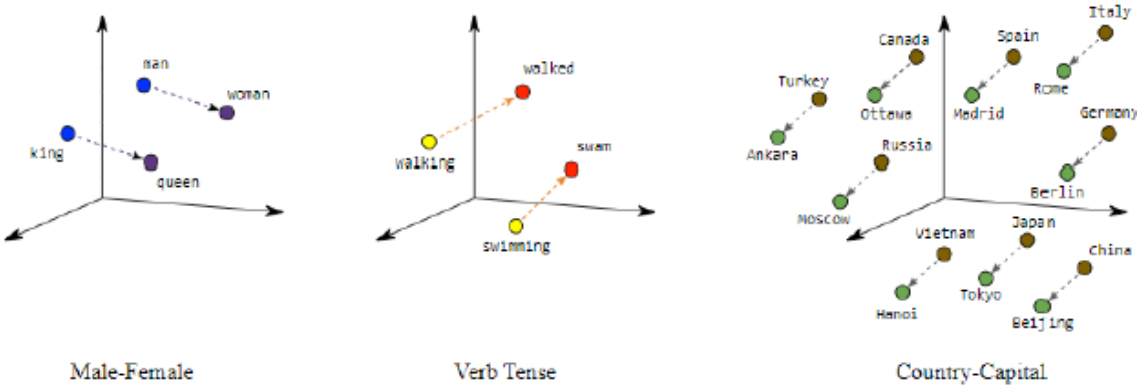

Figura 3.1. Proceso de tokenización de palabras para NLP. Fuente: [10]

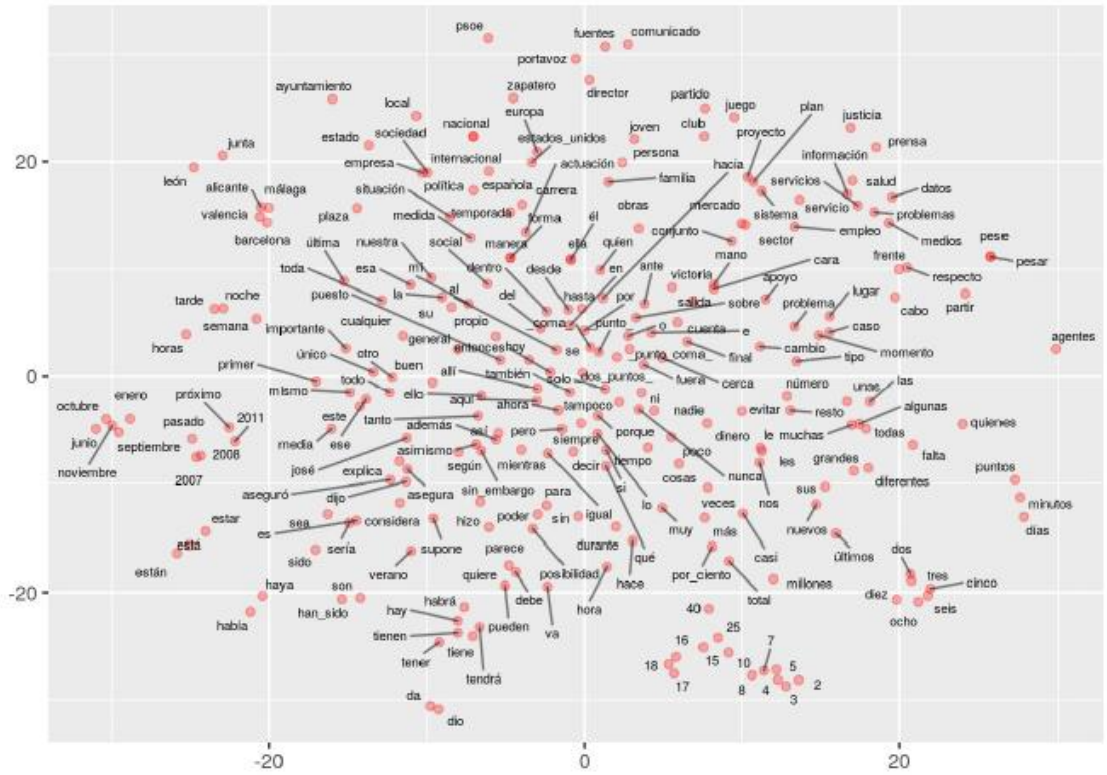

Figura 3.2. Relación de palabras en espacio N-dimensional para NLP. Fuente: [11]

Asimismo, el estudio revisa herramientas [12] actuales basadas en NLP que ya están siendo utilizadas en el entorno empresarial, como HubSpot Service Hub, Talkwalker, Repustate, Lexalytics, Empathic Workplace o KeenCorp. Estas plataformas permiten analizar el sentimiento presente en textos de clientes o empleados y ofrecen funcionalidades para visualizar tendencias y tomar decisiones estratégicas basadas en dicha información.

Sin embargo, tal y como subraya el trabajo, una de las principales limitaciones de estas herramientas es su orientación hacia el análisis de satisfacción del cliente o del clima laboral de forma genérica. En cambio, el análisis emocional vinculado a procesos de cambio organizacional plantea un reto distinto: los estados emocionales que surgen en estos contextos pueden tener implicaciones críticas en la gestión del proyecto y en la eficacia del cambio implementado. Por ello, el TFM propone la necesidad de adaptar o entrenar específicamente estas herramientas para capturar de forma más precisa los sentimientos particulares que emergen durante este tipo de procesos.

Esta perspectiva innovadora sitúa el canal textual no solo como un recurso de análisis pasivo, sino como una fuente activa de conocimiento emocional, capaz de influir directamente en la toma de decisiones durante la dirección de proyectos de transformación organizacional.

3.3. Social media listening

En los últimos años, el análisis de redes sociales, o *social media listening* [13], ha evolucionado hasta convertirse en una herramienta esencial para comprender cómo las emociones se manifiestan y propagan en línea. La cantidad masiva de contenido generado por los usuarios permite estudiar no solo qué se dice, sino cómo se siente, y cómo esas emociones influyen en la toma de decisiones o en fenómenos sociales más amplios.

Desde un punto de vista técnico, la evolución de esta disciplina ha estado marcada por el paso de enfoques tradicionales basados en diccionarios de sentimientos, como AFINN, SentiWordNet o DepecheMood, hacia el uso de arquitecturas complejas de aprendizaje automático y profundo. Modelos como LSTM, GRU o BiLSTM fueron los primeros en captar secuencias de emociones en el lenguaje, pero con la llegada de los modelos *transformer* (ver figura 3.3), la capacidad de análisis semántico y contextual ha mejorado notablemente. Un ejemplo de ello se encuentra en los resultados del reto SemEval-2019 [14], donde los modelos basados en *transformers* superaron consistentemente a las redes neuronales recurrentes, logrando métricas superiores al 75 % en tareas de detección de emociones en textos breves.

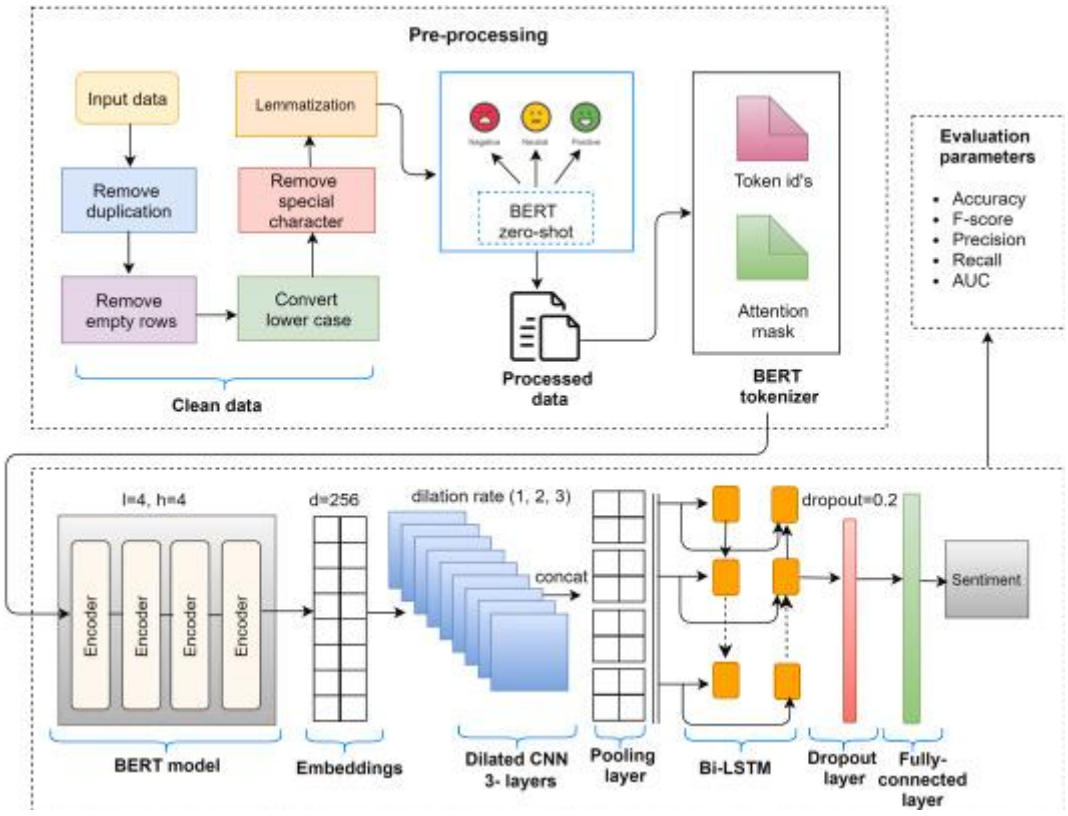

Figura 3.3. Funcionamiento de un modelo transformer, en este caso BERT. Fuente: [15]

En cuanto a su aplicabilidad, diversos estudios han demostrado que las emociones expresadas en redes sociales siguen ciertos patrones de difusión que pueden correlacionarse con fenómenos sociales o mediáticos. Por ejemplo, se ha comprobado que emociones como la ira tienden a viralizarse con mayor rapidez y alcance que otras como la tristeza o la alegría. Estas dinámicas se vuelven especialmente relevantes en contextos de crisis, marketing político o campañas de salud pública. De hecho, investigaciones recientes (Zhou et al., 2023) han analizado cómo los picos de contenido emocional en redes como Twitter y Reddit coinciden con la publicación de noticias destacadas, revelando que ciertos eventos provocan oscilaciones emocionales masivas, medibles en tiempo real a través de herramientas de *social media listening*.

A nivel de investigación, estas herramientas también permiten analizar los llamados *emotion triggers*, es decir, elementos dentro de los textos que despiertan una reacción emocional específica, lo cual resulta útil en estudios narrativos o de dinámica conversacional en comunidades digitales. Plataformas como Reddit, con hilos extensos y contextos emocionales prolongados, se han convertido en entornos de estudio ideales para esta línea de trabajo.

No obstante, el análisis emocional en redes sociales todavía enfrenta importantes desafíos. Por un lado, el lenguaje informal, el uso de ironía, el sarcasmo y la ambigüedad lingüística dificultan la correcta interpretación de los mensajes. Por otro, las diferencias culturales y contextuales generan variabilidad en la expresión de emociones, lo que obliga a entrenar modelos específicos para cada dominio. Además, la precisión de estos modelos tiende a disminuir cuando se aplican a entornos distintos a aquellos para los que fueron entrenados, lo que hace necesario el uso de enfoques transferibles y adaptativos.

Desde una perspectiva comercial, empresas como Talkwalker, Repustate o Lexalytics han desarrollado soluciones que aplican estas metodologías en tiempo real, integrando no solo análisis de sentimiento, sino también detección de ironía, etiquetado emocional y análisis semántico profundo para clientes de sectores como la comunicación, el marketing, la política o la salud.

En resumen, el *social media listening* se ha convertido en una disciplina transversal con aplicaciones tanto científicas como industriales, impulsada por el avance de modelos de inteligencia artificial cada vez más precisos. Sin embargo, sigue siendo necesario abordar cuestiones éticas, metodológicas y técnicas que garanticen un uso responsable, eficaz y justo de estas tecnologías en contextos sociales diversos.

3.4. BERT

BERT [16] [17] (*Bidirectional Encoder Representations from Transformers*) es un modelo de lenguaje desarrollado por Google en 2018 que ha supuesto un punto de inflexión en el procesamiento del lenguaje natural (NLP). A diferencia de modelos previos que leían los textos de manera unidireccional, BERT introduce una arquitectura verdaderamente bidireccional basada en el mecanismo de atención de los *transformers*. Esto le permite captar simultáneamente el contexto que rodea a cada palabra desde ambos extremos de la frase, generando así representaciones semánticas más precisas.

Su entrenamiento consta de dos fases clave: por un lado, el *Masked Language Modeling* (ver figura 3.4), en el que algunas palabras del texto se ocultan para que el modelo aprenda a predecirlas basándose en el contexto completo. Y, por otro lado, el *Next Sentence Prediction* (ver figura 3.5), mediante el cual el modelo evalúa si una oración sigue lógicamente a otra, permitiéndole manejar relaciones entre frases. Esta estrategia de preentrenamiento lo dota de una comprensión contextual robusta, y lo convierte en una base muy versátil para tareas específicas, como la clasificación de sentimientos, la detección de emociones, o la respuesta automática a preguntas, mediante un proceso posterior de *fine-tuning*.

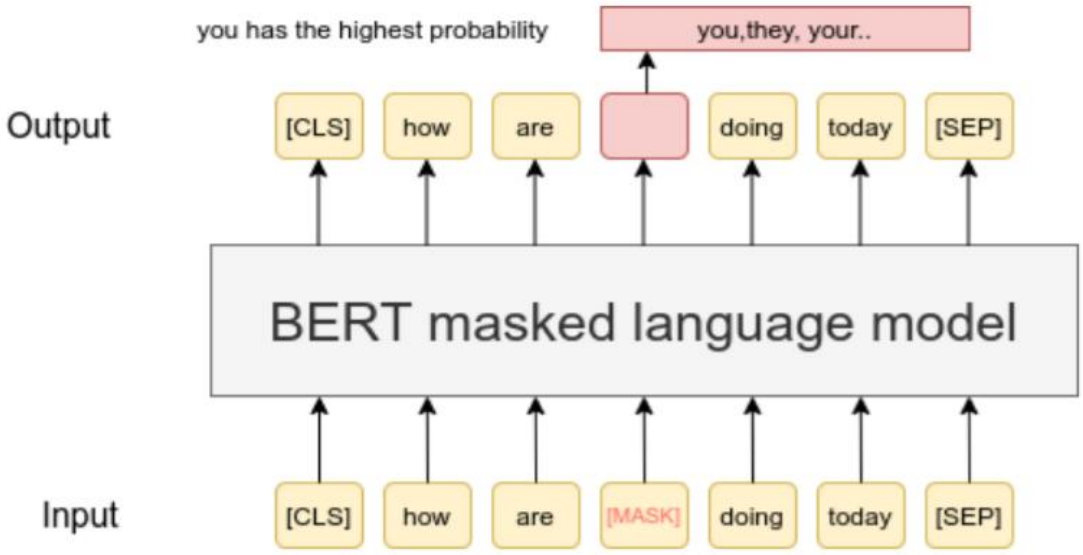

Figura 3.4. Proceso de Masked Language Modeling. Fuente: [18]

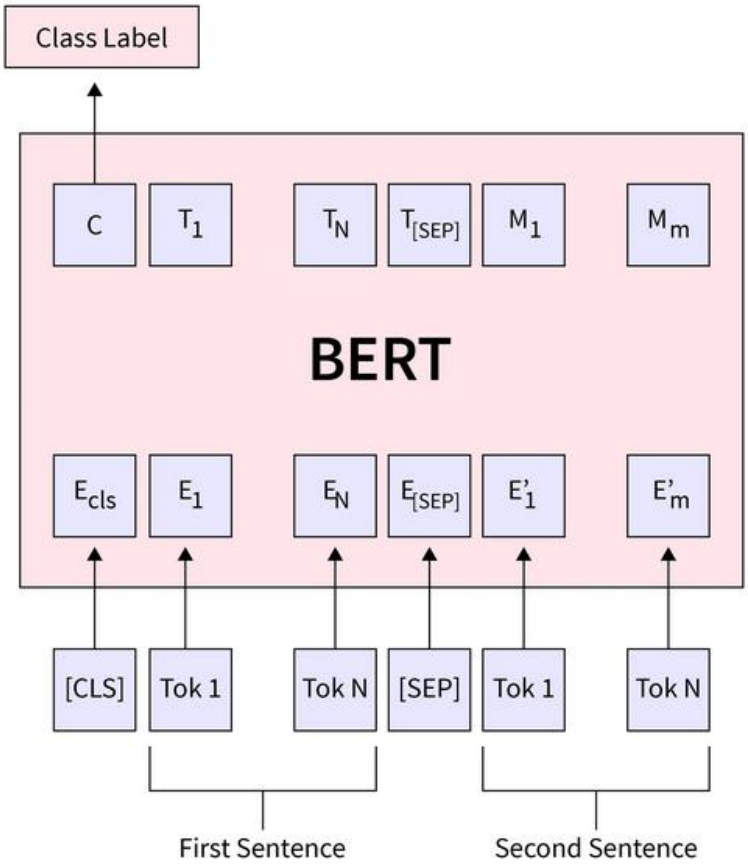

Figura 3.5. Proceso de Next Sentence Prediction. Fuente: [19]

Según Zhang et al. (2020), BERT ha demostrado ser altamente eficaz [20] incluso en dominios críticos como el médico, donde el lenguaje técnico y ambiguo requiere interpretaciones precisas del contexto. En su estudio, los autores utilizaron BERT para extraer automáticamente emociones, síntomas y afirmaciones desde notas clínicas no estructuradas, alcanzando niveles de precisión que superaban con claridad a los métodos tradicionales basados en reglas o léxicos predefinidos. Este enfoque no solo redujo el trabajo manual, sino que también mejoró la capacidad de los sistemas para detectar información implícita.

Además, la arquitectura de atención de BERT le permite enfocarse dinámicamente en las partes más relevantes de una oración para realizar inferencias. Esta capacidad resulta especialmente valiosa en textos breves y contextualmente densos, donde el significado de una palabra o expresión depende en gran medida del entorno lingüístico inmediato.

En definitiva, BERT constituye una herramienta fundamental en el procesamiento moderno del lenguaje, ofreciendo una comprensión profunda y contextualizada que ha mejorado de forma sustancial el rendimiento en múltiples tareas de análisis automático del lenguaje humano.

3.5. EmoNet

EmoNet es un sistema avanzado de detección de emociones diseñado para abordar los desafíos del análisis emocional en textos breves, ambiguos y propios de entornos informales como las redes sociales. Su principal fortaleza radica en la combinación de enfoques simbólicos y conexionistas, integrando conocimiento léxico explícito con representaciones aprendidas mediante redes neuronales profundas.

Una de las variantes más influyentes de EmoNet [21] fue desarrollada por Abdul-Mageed y Ungar (2017), quienes construyeron un corpus masivo de más de 1,6 millones de tuits utilizando una técnica de etiquetado débil. En este proceso, se recurrió a hashtags emocionales inspirados en la rueda de emociones de Plutchik (ver figura 3.6) como etiquetas automáticas. Posteriormente, se aplicaron filtros para garantizar la calidad del contenido: se eliminaron duplicados, tuits con múltiples etiquetas emocionales, mensajes con URLs y aquellos que no estaban en inglés. Una muestra aleatoria del corpus fue revisada manualmente, validando una precisión del 90 %.

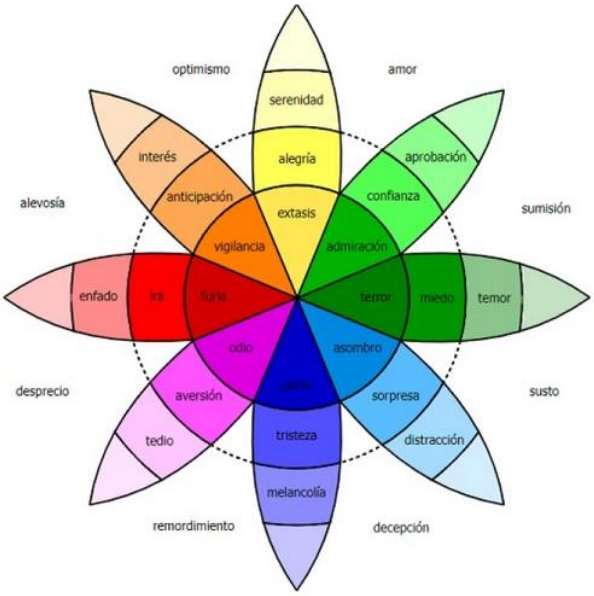

Figura 3.6. Rueda de emociones propuesta por Plutchik. Fuente: [22]

Desde el punto de vista arquitectónico, EmoNet se apoya en redes neuronales recurrentes con compuertas GRNN (*Gated Recurrent Neural Networks*), lo que le permite modelar de manera efectiva la secuencia y la dependencia contextual de las palabras en una oración. Esta capacidad resulta esencial para capturar el matiz emocional que puede depender del orden o la combinación de palabras. El sistema logró una media del 87,6 % en F1-score para la

clasificación de 24 emociones diferentes, y una impresionante precisión del 95,7 % al agruparlas según los ocho ejes emocionales primarios definidos por Plutchik.

Otra implementación relevante de EmoNet, propuesta por Yang et al. (2017), adopta un modelo híbrido que combina una red neuronal convolucional (CNN) con vectores de palabras obtenidos mediante *embeddings* preentrenados, y características léxicas extraídas de recursos como el NRC Emotion Lexicon. Esta estrategia permite al modelo detectar patrones sintácticos locales y, al mismo tiempo, utilizar conocimiento emocional explícito. Este enfoque fortalece la robustez del sistema frente a expresiones no literales, metáforas, sarcasmos o modismos culturales, típicos del lenguaje informal en redes.

En ambos casos, EmoNet fue evaluado frente a clasificadores tradicionales como Naive Bayes o Perceptrón, y superó ampliamente sus resultados gracias a su capacidad para aprender representaciones semánticas complejas. Su rendimiento también fue competitivo respecto a modelos neuronales más recientes, especialmente en tareas de clasificación emocional en textos breves.

En resumen, EmoNet constituye una propuesta sólida y eficaz en el campo del análisis emocional automático. Su diseño híbrido, que combina el conocimiento simbólico con la potencia de las redes neuronales, no solo mejora la precisión, sino que también permite interpretar mejor el significado emocional del lenguaje en contextos reales. Esta combinación lo convierte en un modelo altamente aplicable tanto en investigación como en aplicaciones comerciales centradas en el procesamiento emocional del lenguaje natural.

3.6. Directrices éticas para una IA fiable

Las directrices [24] establecen que una IA confiable debe cumplir con tres componentes principales:

- **Legalidad:** la IA debe respetar todas las leyes y regulaciones aplicables, como el RGPD (Reglamento General de Protección de Datos) y otros marcos legales europeos.
- **Ética:** el desarrollo y uso de IA debe seguir principios éticos como el respeto a los derechos humanos, la autonomía individual, la justicia, la no maleficencia y el bienestar social.
- **Solidez técnica y social:** los sistemas deben ser técnicamente robustos, seguros y fiables, sin dejar de estar alineados con valores sociales y humanos.

Siete requisitos clave para una IA fiable:

- **Agencia y supervisión humanas:** los sistemas de IA deben permitir la intervención humana significativa y respetar la autonomía del usuario.
- **Robustez técnica y seguridad:** los sistemas deben ser resilientes ante errores o manipulaciones, garantizando ciberseguridad y fiabilidad.
- **Privacidad y gobernanza de los datos:** es esencial garantizar la protección de datos personales, control del usuario sobre su información y gobernanza transparente de los datos utilizados.
- **Transparencia:** los procesos, decisiones y capacidades de los sistemas de IA deben ser comprensibles para humanos, incluyendo trazabilidad y explicabilidad.
- **Diversidad, no discriminación y equidad:** la IA debe evitar sesgos injustos y promover la inclusión.
- **Bienestar social y ambiental:** la IA debe contribuir positivamente a la sociedad y al medio ambiente.
- **Responsabilidad:** debe establecerse una rendición de cuentas clara, incluyendo mecanismos para evaluar el impacto del sistema y mitigar sus efectos adversos.

Dado que el presente proyecto utiliza datos generados por personas y aplica técnicas avanzadas de inteligencia artificial, es imprescindible considerar el marco normativo vigente en la Unión Europea, así como adoptar buenas prácticas de gestión del riesgo.

En junio de 2024 entró en vigor el Reglamento (UE) 2024/1689 del Parlamento Europeo y del Consejo, conocido como la Ley de Inteligencia Artificial [25] (AI Act). Esta normativa establece un sistema de clasificación de los sistemas de IA en función de su nivel de riesgo, tal y como se ve en la figura 3.7:

- **Riesgo inaceptable**, prohibido por la ley.
- **Riesgo alto**, que requiere cumplimiento estricto de requisitos técnicos y organizativos.
- **Riesgo limitado**, sujeto a ciertas obligaciones de transparencia.
- **Riesgo mínimo**, sin obligaciones específicas adicionales.

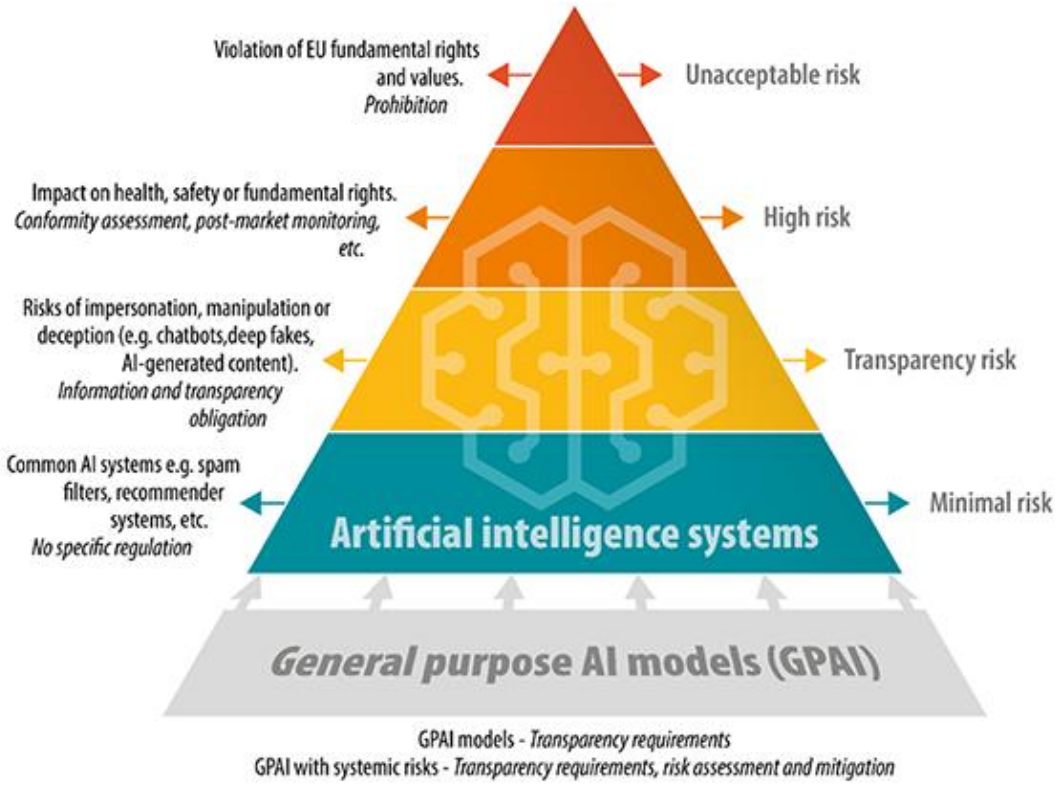

Figura 3.7. Niveles de riesgo establecidos por el reglamento del parlamento europeo. Fuente: [25]

Los sistemas de alto riesgo, como aquellos que analizan emociones humanas, deben cumplir con obligaciones específicas en materia de gobernanza de datos, trazabilidad, supervisión humana, transparencia y gestión formal de riesgos. Esto incluye la implementación de un sistema de evaluación continua del comportamiento del modelo, así como el mantenimiento de registros y documentación técnica adecuados.

En el contexto de la estrategia nacional de digitalización, España ha impulsado un *sandbox* regulatorio para sistemas de IA, gestionado por la SEDIA, que permite desarrollar y validar tecnologías de forma segura y conforme al nuevo Reglamento (UE) 2024/1689. Este proyecto adopta un enfoque de gestión de riesgos alineado con la guía PMBOK y adaptado al marco normativo europeo, mediante una matriz que permite identificar, evaluar y mitigar riesgos asociados, garantizando así una ejecución más eficiente y coherente con los principios de seguridad, transparencia y fiabilidad exigidos por la legislación vigente.

Capítulo 4. Diseño

Aunque el sistema desarrollado no consiste en una aplicación con interfaz gráfica o entorno web, sino en un programa que se ejecuta a través de la línea de comandos, la fase de diseño sigue siendo una parte fundamental del proceso de desarrollo.

El diseño permite estructurar y organizar el proyecto de forma clara y escalable, facilitando tanto el desarrollo como el mantenimiento de este. A través del diseño se definen los módulos principales, las responsabilidades de cada componente, las relaciones entre ellos y los flujos de ejecución. Además, esta etapa ayuda a anticipar posibles problemas, mejorar la reutilización del código y garantizar una mayor calidad en el producto final.

En esta sección se presentan los requisitos del sistema, la estructura del proyecto y los casos de uso más relevantes, acompañados de sus respectivos diagramas de secuencia. Estos elementos permiten entender cómo se organiza internamente el sistema y cómo interactúan sus componentes durante la ejecución.

4.1. Requisitos

Aunque Scrum es una metodología ágil que se apoya fundamentalmente en historias de usuario para definir y priorizar funcionalidades desde la perspectiva del usuario final, en el contexto de este Trabajo Fin de Grado se ha optado por el uso de requisitos. Esta decisión se justifica principalmente por motivos de claridad y estructura académica, ya que los requisitos permiten una redacción más técnica, directa y alineada con los estándares tradicionales de documentación en proyectos universitarios. Además, al tratarse de un proyecto desarrollado de forma individual y sin la participación directa de un cliente real, el uso de historias de usuario resultaba menos práctico.

Requisitos funcionales

- **RF1 – Carga del set de datos:** El sistema debe permitir importar sets de datos en formato CSV con datos textuales para su análisis emocional.
- **RF2 – Formato de datos de entrada:** El sistema debe aceptar sets de datos que sigan el formato de entrada `id, sentiment, content`, garantizando que la estructura de los datos sea compatible con el flujo de procesamiento y análisis posterior.
- **RF3 – Limpieza y normalización del texto:** El sistema debe implementar un proceso de limpieza y normalización del contenido textual, que incluya al menos operaciones de

tokenización, con el objetivo de preparar adecuadamente los datos para el entrenamiento de los modelos.

- **RF4 – Entrenamiento de modelos clásicos:** El sistema debe permitir entrenar los modelos tradicionales Regresión Logística, Naive Bayes, Random Forest y SVM para llevar a cabo tareas de clasificación emocional.
- **RF5 – Entrenamiento de modelo basado en BERT:** El sistema debe permitir integrar y entrenar el modelo preentrenado bert-base-uncased para llevar a cabo tareas de clasificación emocional.
- **RF6 – Evaluación de modelos:** El sistema debe evaluar el rendimiento de los modelos entrenados utilizando el conjunto de datos principal, el cual se dividirá en un 80 % para entrenamiento y 20 % para validación. Las métricas de evaluación generadas deben ser: precisión, *recall*, *F1-score*, matriz de confusión, curvas de aprendizaje, ROC-AUC y *log loss*.
- **RF7 – Validación con conjunto de datos externo:** El sistema debe permitir la evaluación de los modelos entrenados utilizando un conjunto de datos externo, generando las mismas métricas de evaluación especificadas en RF6, con excepción de las curvas de aprendizaje.
- **RF8 – Visualización de resultados:** El sistema debe generar gráficos e informes visuales que faciliten la interpretación de los resultados obtenidos en la evaluación de los modelos.
- **RF9 – Clasificación de emociones:** El sistema debe ser capaz de clasificar textos en cuatro categorías emocionales: alegría, tristeza, miedo e ira.

Requisitos no funcionales

- **RNF1 – Portabilidad:** El sistema debe ser portable y funcionar correctamente en cualquier sistema con Python 3.13.3+ instalado.
- **RNF2 – Rendimiento:** El tiempo de entrenamiento de los modelos debe mantenerse dentro de límites razonables, acordes con la complejidad del modelo. Se pueden llegar a admitir tiempos de hasta 5 horas.
- **RNF3 – Escalabilidad:** El sistema debe poder ampliarse para trabajar con otros modelos o incorporar más emociones sin cambios estructurales importantes.
- **RNF4 – Seguridad y anonimato:** Todos los datos procesados deben anonimizarse si provienen de redes sociales u otras fuentes sensibles, para cumplir el Reglamento de IA y la GDPR.
- **RNF5 – Legibilidad del código:** El código debe estar debidamente comentado, organizado en módulos y seguir la guía de estilo PEP8 [26].

Reglas de negocio

- **RN1 – Datos sensibles:** No se pueden almacenar ni utilizar datos que permitan identificar directamente a usuarios individuales.
- **RN2 – Validación del conjunto de datos:** Solo se pueden utilizar sets de datos que tengan una procedencia legal clara y estén etiquetados para tareas de análisis emocional.
- **RN3 – Tipos de emociones:** La clasificación debe basarse en un conjunto fijo de emociones acordad. En este caso el conjunto de Ekman [28].
- **RN4 – Evaluación estandarizada:** Todos los modelos deben evaluarse sobre el mismo conjunto de prueba para garantizar la comparabilidad de resultados.

4.2. Casos de uso

CU01	*Entrenamiento y validación de un modelo*
Descripción	El usuario selecciona un modelo de aprendizaje automático para ser entrenado y validado con un conjunto de datos previamente configurado.
Actor	Usuario
Precondiciones	El usuario ha configurado correctamente la ruta del conjunto de datos que se desea utilizar.
Postcondiciones	El sistema presenta las métricas de evaluación del modelo entrenado. Las métricas y gráficas se almacenan en una carpeta de resultados.
Flujo	1. El usuario ejecuta el programa. 2. El sistema solicita al usuario que seleccione un modelo de entrenamiento. 3. El usuario elige uno de los modelos disponibles. 4. El sistema carga los datos de entrenamiento. 5. El sistema entrena el modelo seleccionado. 6. El sistema evalúa el modelo con métricas estándar. 7. El sistema evalúa el modelo con un conjunto de validación externa.
Flujo alternativo	3.1A El usuario selecciona el modelo BERT. 3.2A El sistema entrena y valida el modelo BERT.
Excepciones	3.1B Si el usuario escoge un modelo no reconocido, el sistema muestra un mensaje de error. 3.2B El programa finaliza sin ejecutar entrenamiento ni evaluación.

Tabla 4.1. Caso de uso 01: entrenamiento y validación de un modelo.

Dado que el sistema es lo suficientemente simple y cuenta con un único caso de uso principal, no se considera necesario un diagrama de casos de uso tradicional. Este único caso de uso representa la secuencia principal del sistema, desde la selección del modelo hasta la evaluación final.

A continuación, en la figura 4.1, se muestra esta secuencia de forma más precisa mediante un diagrama de secuencia, que ilustra las interacciones entre los componentes durante la ejecución del flujo principal.

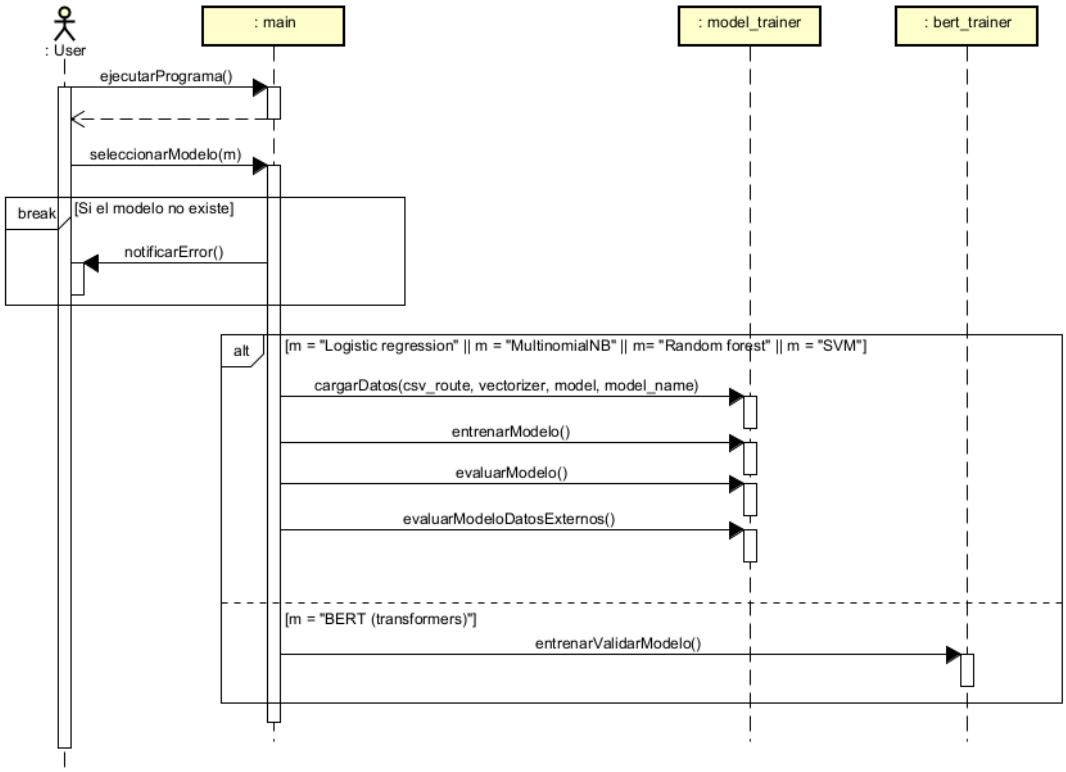

Figura 4.1. Diagrama de secuencia entrenamiento y validación de un modelo. Fuente: Astah [27]

4.3. Estructura del proyecto

Por último, en la figura 4.2 se ilustra la estructura del sistema mediante un sencillo diagrama modular de estilo Uses Style. Este diagrama permite visualizar de forma clara las dependencias entre las clases y paquetes del proyecto, facilitando la comprensión de cómo se organiza y colabora internamente el sistema.

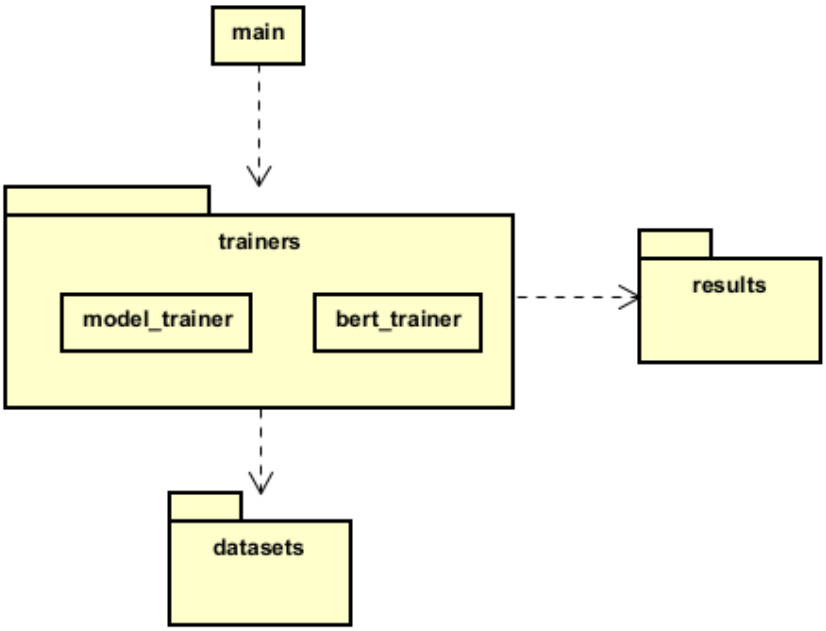

Figura 4.2. Diagrama uses style del proyecto. Fuente: Astah [27]

Capítulo 5. Búsqueda y selección de datos

El primer paso fundamental en el desarrollo consiste en la búsqueda de datos adecuados para entrenar el modelo de inteligencia artificial. En particular, se requieren textos previamente etiquetados, es decir, datos que ya cuenten con una clasificación o anotación que sirva como base para el aprendizaje supervisado del modelo. La obtención de este tipo de datos es crucial, ya que sin ellos no sería posible llevar a cabo el entrenamiento de forma efectiva.

Durante esta fase inicial se contemplaron dos enfoques principales para conseguir los datos etiquetados necesarios:

- **Generación de datos mediante inteligencia artificial:** Esta opción consiste en utilizar otra IA para generar textos junto con sus etiquetas. Si bien este enfoque podría resultar útil en determinadas situaciones, presenta el riesgo de introducir sesgos o errores, además de que el proceso de validación de los datos generados sería complejo y costoso.
- **Búsqueda de sets de datos ya existentes:** La segunda opción, y finalmente la elegida, consiste en localizar conjuntos de datos ya creados por otros usuarios o investigadores. Este enfoque se ha considerado más fiable, siempre que se preste la debida atención a la calidad y procedencia de los datos.

En la búsqueda de fuentes externas de datos, se han explorado diferentes repositorios y plataformas. Aunque en algunos casos la búsqueda no fue especialmente fructífera, se identificó una plataforma con gran potencial para este propósito: Kaggle [29].

Kaggle es una plataforma ampliamente conocida dentro del ámbito de la ciencia de datos y el aprendizaje automático. Se puede describir como una especie de red social orientada a la analítica de datos, en la que los usuarios pueden compartir y descargar sets de datos, publicar notebooks (cuadernos interactivos con código y resultados), participar en competiciones de ciencia de datos y colaborar con otros miembros de la comunidad.

Entre las principales ventajas de Kaggle destacan las siguientes:

- **Accesibilidad y comunidad:** La plataforma facilita el acceso a una gran variedad de recursos y permite interactuar con una comunidad activa de profesionales, investigadores y entusiastas de la ciencia de datos.
- **Variedad de sets de datos:** Se pueden encontrar conjuntos de datos sobre temas muy diversos, lo cual la convierte en un excelente punto de partida para realizar pruebas y experimentaciones.
- **Documentación y ejemplos:** Muchos sets de datos vienen acompañados de descripciones detalladas, notebooks de ejemplo y comentarios de la comunidad, lo que facilita su comprensión y uso.

No obstante, también es importante tener en cuenta ciertas limitaciones:

- **Calidad variable:** Al tratarse de una plataforma abierta, cualquier usuario puede subir contenido, por lo que la calidad de los sets de datos puede variar significativamente. Algunos conjuntos de datos pueden estar incompletos, mal etiquetados o no contar con información contextual suficiente.
- **Uso limitado en producción:** Aunque Kaggle es ideal para la fase de experimentación y prototipado, no siempre es la fuente más adecuada para obtener datos destinados a aplicaciones finales o sistemas en producción. En estos casos, se recomienda recurrir a fuentes oficiales, validadas o especializadas.
- **Dificultad para construir un set de datos diverso y homogéneo:** En Kaggle, la mayoría de los conjuntos de datos han sido creados de forma independiente, lo que implica diferencias en estructura, formato y temática. Esta variedad dificulta su combinación directa, ya que cada uno puede pertenecer a un ámbito distinto. Unificar múltiples conjuntos de datos para cubrir una amplia gama de situaciones requiere un esfuerzo considerable de normalización, limpieza y alineación conceptual, lo que hace muy costosa la creación de un conjunto representativo y coherente.

En definitiva, Kaggle resulta ser una herramienta muy útil durante la fase exploratoria del proyecto. A pesar de la necesidad de revisar cuidadosamente los datos encontrados para asegurar su calidad, su uso ha permitido avanzar en el desarrollo del modelo sin necesidad de generar manualmente grandes volúmenes de datos etiquetados.

Para esta investigación se ha seleccionado un conjunto de datos disponible en la plataforma Kaggle, concretamente el denominado "*Emotions*" [30]. Este set de datos resulta especialmente adecuado por dos motivos:

- **Cobertura emocional:** Incluye ejemplos etiquetados con las principales emociones consideradas en este trabajo, lo que permite llevar a cabo una clasificación multiclase coherente con los objetivos del proyecto.
- **Adecuación al contexto digital:** El conjunto está compuesto por tweets ya preprocesados, lo que garantiza una alta calidad textual sin ruido excesivo. Además, mantiene un estilo de lenguaje informal y propio de redes sociales, alineado con el tipo de comunicación emocional que se desea analizar, lo que lo convierte en un recurso ideal para comenzar a entrenar y validar modelos de análisis emocional en entornos reales.

Además, se ha incorporado un segundo set de datos también de Kaggle, conocido como "*Emotion Dataset for Emotion Recognition Tasks*" [31], para llevar a cabo una evaluación externa del modelo. Este conjunto de datos también está compuesto por tweets etiquetados y preprocesados, conservando el lenguaje coloquial y el enfoque emocional del conjunto principal, lo que permite comprobar la capacidad del modelo para generalizar correctamente sobre datos no vistos. Aunque ambos conjuntos pertenecen al mismo dominio, presentan

diferencias en la jerga utilizada, lo que añade un reto adicional para la generalización del modelo.

Además de estos, existen otros conjuntos de datos relevantes para la detección de emociones en texto que, aunque no se han empleado directamente, resultan de gran interés para investigaciones futuras:

- **GoEmotions [32]:** Este conjunto de datos, desarrollado por Google, contiene más de 58.000 comentarios de Reddit etiquetados con 27 emociones distintas y una clase neutral. Está diseñado para entrenar y evaluar modelos de detección emocional en texto, y se considera uno de los recursos más completos en este ámbito. Aunque es más complejo que el set de datos utilizado en este trabajo, su estructura lo hace adaptable a tareas similares.
- **EmoBank [33]:** EmoBank es un corpus de unas 10 000 oraciones en inglés anotadas según un modelo dimensional de emociones, basado en tres ejes: Valencia, Activación y Dominancia. A diferencia de los enfoques categóricos, permite representar emociones de forma continua, capturando matices más sutiles. El conjunto de datos proviene de diversas fuentes como artículos, reseñas y contenido generado por usuarios, lo que aporta variedad estilística y contextual. Aunque más complejo, su formato puede adaptarse para tareas similares a las planteadas en este trabajo.

Más adelante, en el capítulo 6, apartado 6.4.2, se profundizará en la librería de Python pandas, una herramienta fundamental para la manipulación y análisis de datos. Por el momento, se utilizará para obtener un resumen estadístico básico sobre la cantidad de ejemplos que componen los conjuntos de datos seleccionados:

```
Distribución de clases:
sentiment
joy        141067
sadness    121187
anger       57317
fear        47712
```

Figura 5.1. Distribución del set de datos principal

```
Distribución de clases:
sentiment
joy        704
sadness    550
anger      275
fear       212
```

Figura 5.2. Distribución del set de datos externo

Como se puede apreciar en la figura 5.1, el set de datos principal cuenta con una gran cantidad de ejemplos, de los cuales probablemente no se utilicen todos. Sin embargo, esta abundancia garantiza la disponibilidad de datos suficientes para realizar todas las pruebas necesarias y, además, deja margen para futuras investigaciones. Por su parte, en la figura 5.2, el conjunto de datos externo ofrece un volumen adecuado de muestras para llevar a cabo una validación efectiva del modelo.

A diferencia de muchos enfoques tradicionales en el análisis de sentimientos, que se centran en clasificaciones generales como “positivo”, “negativo” o “neutro”, este trabajo opta por un análisis más específico y granular basado en emociones concretas. Esta decisión se fundamenta en la necesidad de capturar mejor la riqueza emocional presente en el lenguaje cotidiano, especialmente en contextos digitales como las redes sociales, donde las emociones se expresan de forma más matizada. Para ello, se han seleccionado cuatro emociones básicas: ira, tristeza, alegría y miedo, todas ellas ampliamente reconocidas en la literatura psicológica. En particular, se toma como referencia el modelo de emociones básicas propuesto por Paul Ekman [28], quien identificó un conjunto reducido de emociones universales presentes en todas las culturas. Aunque el modelo de Ekman incluye otras emociones como el asco o la sorpresa, las seleccionadas en este proyecto representan un espectro emocional lo suficientemente amplio y frecuente en textos reales como para permitir una clasificación significativa y aplicable en múltiples contextos.

Capítulo 6. Elección de tecnologías

6.1. Lenguaje de programación

En el desarrollo de sistemas de inteligencia artificial, la elección del lenguaje de programación es una decisión estratégica que puede influir significativamente en la eficiencia, escalabilidad y facilidad de mantenimiento del proyecto. En este contexto, Python [34] se ha consolidado como el lenguaje por excelencia dentro del campo, tanto en entornos académicos como en la industria, debido a una combinación de factores técnicos y prácticos que lo hacen especialmente adecuado para este tipo de aplicaciones.

Una de sus principales ventajas radica en su sintaxis sencilla, legible y expresiva, que permite escribir código de forma clara y directa. Esta característica resulta especialmente valiosa, ya que facilita la comprensión del código y permite a los desarrolladores centrarse en la lógica del modelo y en el tratamiento de los datos, en lugar de lidiar con una sintaxis compleja o estructuras innecesariamente rígidas. Esta accesibilidad también lo convierte en una herramienta ideal para equipos multidisciplinares, donde pueden colaborar perfiles técnicos y no técnicos de forma más fluida.

Otra de las razones que justifican su uso es el amplio ecosistema de herramientas y bibliotecas específicamente orientadas al desarrollo de soluciones inteligentes. Este entorno de desarrollo, que abarca desde el procesamiento de datos hasta el entrenamiento de modelos y su despliegue en producción, ha sido adoptado y mejorado constantemente por una comunidad global muy activa. Es importante mencionar que su disponibilidad contribuye significativamente a reducir los tiempos de desarrollo y a facilitar la implementación de soluciones robustas y escalables.

Asimismo, cuenta con una comunidad de usuarios muy extensa y consolidada, lo que representa un gran valor añadido. Esta comunidad genera constantemente documentación, tutoriales, foros de discusión y soluciones a problemas comunes, lo que permite resolver incidencias de forma ágil y fomentar el aprendizaje continuo. Esta característica no solo reduce las barreras de entrada para nuevos desarrolladores, sino que también facilita el trabajo colaborativo y la mejora continua del código.

Otro aspecto para destacar es su versatilidad y facilidad de integración con otras tecnologías. Se adapta bien a entornos de desarrollo locales, servidores y plataformas en la nube, lo que permite desplegar modelos de inteligencia artificial en diferentes escenarios sin necesidad de realizar adaptaciones complejas. Además, su compatibilidad con servicios web, bases de datos y entornos de computación distribuida lo convierte en una opción flexible para proyectos que requieren escalabilidad y acceso a grandes volúmenes de datos.

Por último, cabe señalar que se trata de un lenguaje de código abierto, con una evolución constante respaldada por una comunidad y una industria que lo impulsan de forma activa. Esta característica garantiza su sostenibilidad a largo plazo, evita la dependencia de proveedores propietarios y asegura el acceso continuo a mejoras, actualizaciones y nuevas funcionalidades.

En resumen, la elección de esta tecnología para el presente trabajo de fin de grado se fundamenta en su simplicidad sintáctica, su amplia adopción en el ámbito de la inteligencia artificial, la riqueza de su ecosistema, el soporte comunitario y su capacidad de integración con distintas plataformas. Estas cualidades la convierten en una herramienta robusta y eficaz para el desarrollo de soluciones inteligentes, frente a otras alternativas.

6.2. Enfoque IA

Antes de continuar con la elección de las tecnologías y librerías que se van a utilizar en el presente Trabajo de Fin de Grado, es fundamental comprender el enfoque que adoptará la inteligencia artificial dentro del proyecto. Esto permite conocer las distintas posibilidades que existen para abordar el problema, entender en qué consiste cada enfoque y, sobre todo, saber cuándo conviene utilizar uno u otro. En este caso, se explorarán dos ramas principales de la inteligencia artificial: el *Machine Learning* (aprendizaje automático) y el *Deep Learning* (aprendizaje profundo). A partir del análisis de sus características, ventajas, desventajas y diferencias clave, se podrá tomar una decisión justificada sobre cuál es el enfoque más adecuado para el reconocimiento de emociones en texto.

6.2.1. Machine Learning

El *Machine Learning* [35] es una rama de la inteligencia artificial que permite a las máquinas aprender a partir de datos sin ser programadas explícitamente para cada tarea específica. En lugar de seguir reglas predefinidas, los algoritmos de *Machine Learning* identifican patrones en los datos y hacen predicciones o toman decisiones basadas en la experiencia previa.

Existen tres tipos principales de aprendizaje en *Machine Learning*:

- **Aprendizaje supervisado:** el modelo se entrena con datos etiquetados.
- **Aprendizaje no supervisado:** el modelo encuentra patrones ocultos en datos no etiquetados.
- **Aprendizaje por refuerzo:** el modelo aprende mediante prueba y error, recibiendo recompensas o penalizaciones.

Ventajas del Machine Learning

- Requiere menos datos para entrenar en comparación con *Deep Learning*.
- Menor consumo de recursos computacionales.
- Mayor interpretabilidad de los modelos (especialmente en modelos como la regresión o los árboles de decisión).
- Más rápido de entrenar y probar en conjuntos de datos pequeños o medianos.

Desventajas del Machine Learning

- El rendimiento se estanca en tareas complejas donde los datos tienen alta dimensionalidad o no linealidades difíciles de modelar.
- Menor capacidad para procesar datos sin preprocesamiento o ingeniería de características.
- Puede requerir un esfuerzo manual significativo para seleccionar y extraer características relevantes.

6.2.2. Deep Learning

El *Deep Learning* [36] es una subárea del *Machine Learning* que se basa en redes neuronales artificiales profundas. Estas redes están compuestas por múltiples capas (de ahí el término "profundo") que permiten al sistema aprender representaciones jerárquicas de los datos.

Gracias a su arquitectura, *el Deep Learning* ha mostrado resultados extraordinarios en tareas como reconocimiento de voz, visión por computadora, procesamiento del lenguaje natural (NLP), entre otros.

Las redes más comunes incluyen [37]:

- Redes neuronales de retroalimentación (FF)
- Redes neuronales recurrentes (RNN)
- Redes de memoria a corto y largo plazo (LSTM)
- Transformers y modelos como BERT o GPT en NLP

Ventajas del Deep Learning

- Excelente rendimiento en tareas complejas y con grandes volúmenes de datos.
- Capacidad de aprender representaciones automáticamente, sin necesidad de una ingeniería de características intensiva.
- Se adapta bien a problemas con datos no estructurados como texto, imágenes y audio.
- Últimos avances en NLP, como los modelos basados en Transformers, superan ampliamente a los enfoques tradicionales.

Desventajas del Deep Learning

- Requiere grandes cantidades de datos para entrenar correctamente.
- Necesita hardware potente (como GPUs) para un entrenamiento eficiente.
- Tiempo de entrenamiento elevado.
- Baja interpretabilidad: es difícil entender qué está "pensando" la red neuronal.
- Mayor complejidad de implementación y ajuste.

6.3. Enfoque para el trabajo

En el contexto de este Trabajo de Fin de Grado, el objetivo principal es desarrollar un sistema de reconocimiento de emociones en texto, una tarea propia del procesamiento del lenguaje natural (NLP). Este tipo de problema es inherentemente complejo debido a la ambigüedad del lenguaje humano, el contexto semántico y la sutileza de las emociones.

Tras analizar ambas aproximaciones, se concluye que *Deep Learning* es, en general, la opción más adecuada para esta clase de tareas. Modelos como LSTM, GRU, BERT y otros basados en *Transformers* han demostrado un desempeño muy superior al de los métodos tradicionales de *Machine Learning* en el reconocimiento de emociones, al poder captar mejor la semántica del lenguaje y aprender representaciones contextuales más ricas.

Sin embargo, la elección final para el desarrollo del sistema será el uso de *Machine Learning*, y esto se debe a dos limitaciones prácticas:

- **Cantidad de datos disponible:** el *Deep Learning* necesita grandes corpus etiquetados para lograr buenos resultados, y el presente proyecto cuenta con un volumen de datos limitado.
- **Capacidad computacional:** entrenar modelos profundos requiere recursos de hardware avanzados (como GPUs), los cuales no están disponibles en el entorno de trabajo actual.

Por estas razones, aunque el *Deep Learning* es tecnológicamente más avanzado para el problema propuesto, se optará por implementar modelos clásicos de *Machine Learning*, que son más viables dadas las condiciones del proyecto.

6.4. Herramientas seleccionadas para el desarrollo del proyecto

Una vez definido el enfoque de inteligencia artificial que se aplicará al problema, y considerando tanto las limitaciones prácticas como los objetivos del proyecto, se procede a describir las herramientas principales que se utilizarán durante el desarrollo del sistema de detección de emociones en texto.

6.4.1. Scikit-learn

Scikit-learn [38] es una de las librerías más consolidadas del ecosistema Python para la implementación de algoritmos de *Machine Learning* clásico. Incluye una amplia gama de modelos supervisados y no supervisados, así como utilidades para preprocesamiento, evaluación y selección de modelos.

Razones de su elección:

- **Simplicidad y eficiencia:** Permite implementar modelos con pocas líneas de código y un rendimiento adecuado para conjuntos de datos pequeños o medianos.
- **Modelos listos para usar:** Ofrece algoritmos ampliamente utilizados como SVM, regresión logística, árboles de decisión, entre otros.
- **Buena integración:** Se combina de forma fluida con Pandas, NumPy y otras librerías del entorno Python.
- **Documentación clara y comunidad activa:** Facilita el aprendizaje y la resolución de problemas comunes.

Limitaciones:

No está diseñado para tareas complejas de *Deep Learning* ni permite aprovechar procesamiento en GPU. Por tanto, resulta menos adecuado para redes neuronales profundas o grandes volúmenes de datos no estructurados. Aun así, en este proyecto es apropiado dada la naturaleza del conjunto de datos y la infraestructura disponible.

6.4.2. Pandas

Pandas [39] es una librería de código abierto para Python orientada a la manipulación y análisis de datos estructurados. Su estructura principal, el *DataFrame*, permite trabajar de forma muy eficiente con datos en formato tabular.

Razones de su elección:

- **Carga y transformación de datos:** Facilita la lectura desde múltiples fuentes como CSV, Excel, JSON o SQL.
- **Manipulación sencilla:** Permite filtrar, agrupar, transformar y combinar datos de forma eficiente.
- **Exploración de datos:** Ofrece herramientas básicas para análisis estadístico y detección de patrones.
- **Manejo de valores nulos:** Incluye funciones intuitivas para identificar y tratar datos faltantes.
- **Alta compatibilidad:** Se integra sin problemas con otras librerías utilizadas en el proyecto.

Limitaciones:

No está diseñada para análisis complejos de aprendizaje automático ni para visualización avanzada. Su propósito principal es la manipulación y exploración de datos estructurados.

6.4.3. Matplotlib

Matplotlib [40] es una biblioteca de visualización para Python ampliamente utilizada para la creación de gráficos estáticos, animados e interactivos en dos dimensiones. Es ideal para representar visualmente información cuantitativa de forma clara y precisa.

Razones de su elección:

- **Variedad de visualizaciones:** Permite generar histogramas, diagramas de barras, gráficos de líneas, de dispersión, matrices de confusión, etc.
- **Personalización:** Ofrece control sobre todos los aspectos del gráfico: colores, tamaños, estilos, etiquetas, títulos, etc.
- **Soporte para análisis exploratorio:** Es fundamental para comprender el comportamiento de los datos y modelos.
- **Compatibilidad:** Se integra fácilmente con NumPy, pandas y Scikit-learn.

Limitaciones:

Aunque muy potente para gráficos básicos y personalizables, puede resultar menos intuitiva que otras bibliotecas modernas para ciertos tipos de visualización compleja.

6.4.4.PyTorch

PyTorch [41] es una librería de código abierto desarrollada por *Facebook AI Research*, especializada en el desarrollo y entrenamiento de modelos de *Deep Learning*. Proporciona una gran flexibilidad y potencia para construir redes neuronales complejas mediante una interfaz intuitiva y dinámica, ideal para investigación y producción.

Razones de su elección:

- **Flexibilidad y dinamismo:** Permite construir y modificar modelos de manera dinámica, facilitando la experimentación y el desarrollo de arquitecturas complejas.
- **Soporte para computación en GPU:** PyTorch aprovecha la aceleración mediante GPUs, lo que es fundamental para entrenar modelos profundos y manejar grandes volúmenes de datos.
- **Amplio ecosistema:** Cuenta con herramientas para visión por computador, procesamiento de lenguaje natural, y otras áreas, además de integración con librerías como NumPy y Pandas.
- **Comunidad activa y documentación:** Tiene una comunidad creciente y recursos actualizados que apoyan el aprendizaje y la resolución de problemas.

Limitaciones:

Requiere mayor complejidad y recursos computacionales. Para conjuntos de datos pequeños o entornos limitados, puede resultar innecesario. En este proyecto se opta por alternativas más simples, como Scikit-learn, por adecuarse mejor a las características del problema y el entorno disponible.

Capítulo 7. Análisis de modelos

Para analizar el rendimiento de distintos modelos en la tarea de reconocimiento automático de emociones en texto, se ha seguido un enfoque uniforme que permite establecer comparaciones justas y consistentes. Estos modelos se resumen en la tabla 7.1, donde se especifica tanto su denominación como la clase o librería correspondiente utilizada en su implementación.

Modelo para analizar	Implementación
Regresión Logística	LogisticRegression
Naive Bayes Multinomial	MultinomialNB
Random Forest	RandomForestClassifier
Support Vector Machines (SVM)	SVC
BERT	bert-base-uncased

Tabla 7.1. Tabla de modelos analizados en el proyecto

En las próximas secciones se describen estos modelos evaluados, detallando tanto su funcionamiento como los parámetros utilizados en cada uno, con una explicación razonada de las decisiones tomadas durante la configuración.

Todos los modelos han sido entrenados bajo las mismas condiciones para garantizar la equidad en la evaluación. Concretamente, se ha extraído una muestra aleatoria de 50.000 ejemplos del conjunto original de datos, manteniendo la proporción original de clases presentes en el set de datos completo. Esto asegura que la distribución de emociones en la muestra utilizada para el entrenamiento sea representativa del dominio general. La muestra se ha dividido en un 80% para entrenamiento y un 20% para validación, y ha servido como base para calcular las métricas principales. Para reforzar la robustez de la evaluación, también se ha utilizado el conjunto de datos externo ya mencionado, formado por 1.700 ejemplos adicionales. Esto permite comprobar la capacidad de generalización de cada modelo frente a datos no vistos previamente, funcionando como una forma sencilla de validación cruzada externa.

En cuanto a la representación del texto, se ha optado por la técnica TF-IDF [42] [43], ampliamente utilizada en procesamiento del lenguaje natural por su capacidad para identificar la importancia de las palabras dentro de un corpus. Esta técnica transforma cada documento en un vector numérico, como se puede observar en la figura 7.1, donde a cada término se le asigna un peso, calculado según la fórmula mostrada en la figura 7.2. Dicho peso refleja tanto la frecuencia de la palabra en el documento como su relevancia respecto al resto del conjunto. Así, las palabras muy frecuentes en todos los textos reciben un peso bajo, mientras que aquellas más características de un documento reciben un peso mayor. Este enfoque resulta

especialmente útil en tareas como la clasificación de emociones, donde ciertas palabras clave pueden marcar la diferencia entre categorías.

Además, se ha limitado el vocabulario a las 5.000 características más informativas y se han tenido en cuenta tanto unigramas (palabras individuales) como bigramas (pares de palabras consecutivas), lo cual permite capturar no solo el significado aislado de términos, sino también patrones frecuentes de coocurrencia que pueden ser expresivos emocionalmente.

La única excepción a esta configuración común es el modelo Naive Bayes Multinomial, que requiere una representación más ajustada a sus supuestos estadísticos. En su caso, se ha limitado la vectorización a unigramas y se ha aplicado suavizado sublineal de frecuencias, ya que este modelo asume una distribución multinomial que no se adapta bien a bigramas ni a frecuencias extremas.

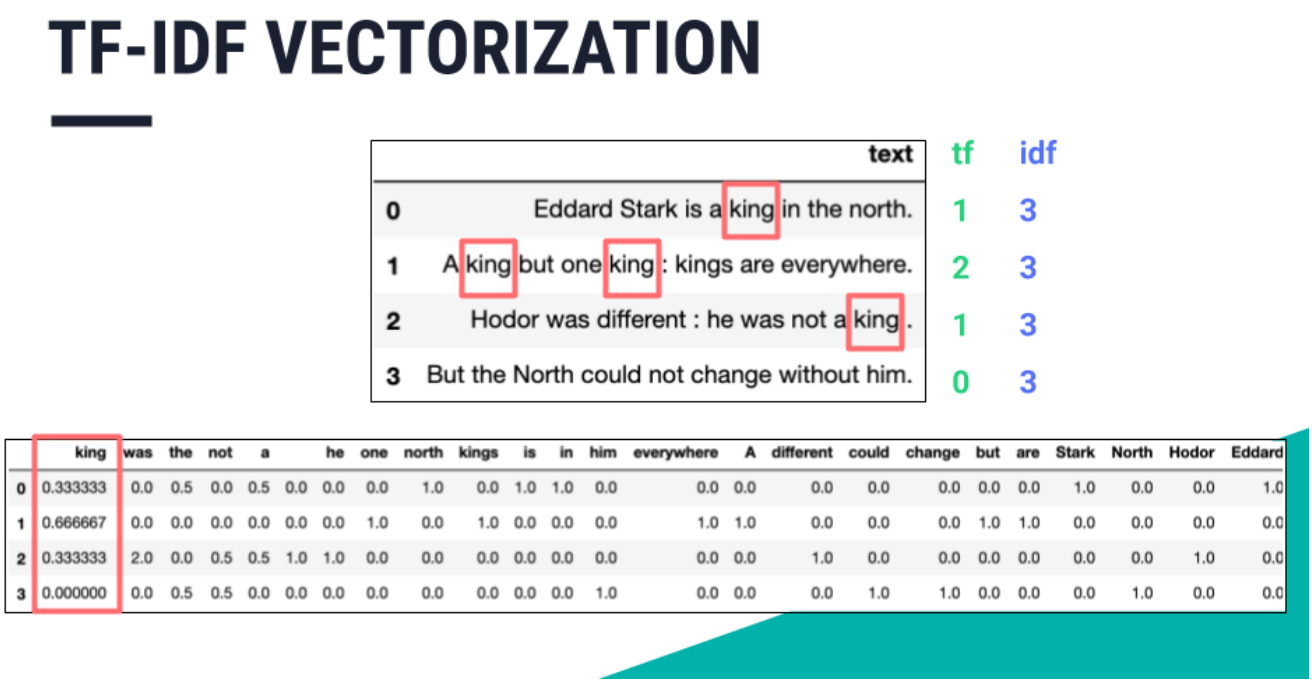

Figure 7.1. Ejemplo de la obtención de las frecuencias inversas de las palabras de un documento. Fuente [66]

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

Figure 7.2. Fórmula para calcular el peso de cada palabra en el documento. Fuente [67]

7.1. Métricas

Con el fin de medir el rendimiento del modelo de clasificación entrenado, se han seleccionado diversas métricas ampliamente utilizadas en el ámbito del aprendizaje automático. Estas permiten obtener una visión detallada del comportamiento del modelo tanto sobre el conjunto de prueba como sobre un conjunto externo para validación cruzada. A continuación, se describen las principales métricas [44] utilizadas:

Evaluación del modelo: Precision, Recall, F1-score y Support

Una de las formas más habituales de evaluar modelos de clasificación supervisada es a través del análisis de métricas derivadas de la comparación directa entre las etiquetas reales y las predicciones realizadas por el modelo. Estas métricas permiten medir de manera detallada la calidad de las decisiones tomadas por el sistema para cada clase, y son fundamentales para interpretar el rendimiento más allá de una simple tasa de acierto general.

Las métricas consideradas son:

- **Precision:** Indica la proporción de verdaderos positivos entre todas las predicciones positivas realizadas por el modelo. Una alta *precision* implica que, cuando el modelo predice una clase determinada, tiene una alta probabilidad de estar en lo correcto. Es especialmente relevante cuando el coste de una falsa alarma es alto.
- **Recall (o sensibilidad):** Mide la proporción de verdaderos positivos sobre el total de ejemplos reales de una clase. Es útil para evaluar hasta qué punto el modelo es capaz de identificar correctamente todos los casos pertenecientes a una categoría concreta, y resulta crítica en contextos donde es más importante detectar todos los casos positivos, aunque se cometan algunos errores.
- **F1-score:** Se trata de la media armónica entre *precision* y *recall*, y proporciona un valor equilibrado cuando se desea mantener un compromiso entre ambas métricas. Es particularmente útil cuando existen desequilibrios entre las clases del conjunto de datos, ya que evita que una sola métrica (como la precisión global) oculte deficiencias en categorías con menos representación. En este trabajo, el *F1-score* se empleará como métrica principal para la comparación entre modelos y configuraciones. Su

capacidad para reflejar tanto la capacidad de detección como la precisión en la clasificación lo convierte en un indicador robusto y adecuado para el problema de clasificación emocional, donde los errores pueden tener consecuencias distintas dependiendo del tipo de emoción mal clasificada.

- **Support:** Hace referencia al número total de ejemplos reales de cada clase presentes en los datos de evaluación. Aunque no constituye una métrica de rendimiento en sí misma, es una medida importante para interpretar correctamente el resto de las métricas, ya que contextualiza los resultados según la distribución de clases.

Matriz de Confusión

La matriz de confusión es una herramienta fundamental para evaluar modelos de clasificación, ya que permite observar visualmente cómo se distribuyen los aciertos y errores del modelo en cada clase. En ella, las filas representan las clases reales y las columnas las predicciones realizadas, mostrando cuántas veces se ha acertado o fallado para cada categoría.

Esta representación resulta especialmente útil cuando se trabaja con múltiples clases, ya que permite identificar patrones de error, como una confusión sistemática entre dos emociones similares. Además, es complementaria a las métricas tradicionales como *precision* o *recall*, ya que proporciona una visión más detallada del comportamiento del modelo en cada clase específica.

ROC-AUC (Receiver Operating Characteristic - Area Under Curve)

La métrica ROC-AUC (*Receiver Operating Characteristic – Area Under Curve*) evalúa la capacidad del modelo para distinguir entre clases. En el contexto multiclase, se suele utilizar un enfoque de tipo "*One-vs-Rest*", calculando un área bajo la curva ROC para cada clase frente al resto y promediando los resultados. Esta métrica refleja el equilibrio entre la tasa de verdaderos positivos (sensibilidad) y la tasa de falsos positivos, proporcionando una medida global del rendimiento del modelo.

El valor del AUC varía entre 0 y 1, donde 1 indica una clasificación perfecta y 0.5 equivale a una predicción aleatoria. Es especialmente útil cuando se desea evaluar el comportamiento del modelo más allá de un simple umbral de decisión, y en casos con clases desbalanceadas, ya que no depende directamente de la distribución de clases ni del número exacto de predicciones acertadas.

Log Loss

La métrica *Log Loss* (pérdida logarítmica) mide la incertidumbre de las predicciones del modelo. Evalúa no solo si el modelo acierta, sino con qué nivel de confianza lo hace. Penaliza más fuertemente las predicciones incorrectas realizadas con alta seguridad, por lo que un modelo que predice con mayor probabilidad las clases correctas obtendrá una puntuación más baja.

Su valor también se sitúa entre 0 y 1, donde valores más cercanos a 0 indican mejores resultados. En clasificación multiclase, se calcula a partir de las probabilidades asignadas por el modelo a cada clase para cada muestra. Es una métrica especialmente útil cuando se necesita tener en cuenta no solo la precisión, sino también la calidad probabilística de las predicciones.

Curvas de Aprendizaje

Las curvas de aprendizaje [45] permiten analizar el comportamiento del modelo a medida que se incrementa la cantidad de datos de entrenamiento. Representan gráficamente la evolución del rendimiento tanto en el conjunto de entrenamiento como en el de validación, en función del número de muestras utilizadas.

Este tipo de análisis es útil para detectar problemas como el sobreajuste (cuando el modelo aprende demasiado bien los datos de entrenamiento y generaliza mal) o el subajuste (cuando el modelo no es capaz de capturar patrones relevantes). Además, ayudan a decidir si añadir más datos podría mejorar el rendimiento o si es necesario modificar la complejidad del modelo.

Evaluación sobre un conjunto externo

Además de evaluar el modelo sobre el conjunto de prueba habitual, se realiza una evaluación adicional sobre un conjunto externo de validación. Este conjunto, que proviene de una fuente diferente, pero con características similares al conjunto original, permite analizar la capacidad de generalización del modelo. En esta validación externa se obtendrán de nuevo las métricas de *precision*, *recall*, *F1-score*, matriz de confusión, ROC-AUC y *log loss*, proporcionando una visión más robusta del rendimiento del sistema en condiciones reales y no vistas durante el entrenamiento.

7.2. Regresión Logística

7.2.1.Fundamentos y Funcionamiento

La regresión logística [46] es uno de los modelos más clásicos y utilizados para tareas de clasificación, especialmente cuando se trata de predecir un resultado binario. Su principal valor reside no solo en su rapidez y eficiencia, sino también en su interpretabilidad, ya que permite observar cómo cada variable influye en la probabilidad de pertenencia a una clase. A través de una función sigmoide o logística, transforma combinaciones lineales de características en valores comprendidos entre 0 y 1, lo que la hace particularmente útil para estimar probabilidades.

Durante el entrenamiento, el modelo ajusta los coeficientes de las variables independientes mediante una estrategia de máxima verosimilitud, buscando los parámetros que hacen más probable la observación de los datos reales. Esto le proporciona una base estadística sólida y permite la incorporación de técnicas de regularización como L1 o L2, especialmente útiles en contextos con un elevado número de variables, como el análisis de texto con representaciones tipo TF-IDF.

Aunque originalmente concebida para clasificación binaria, la regresión logística puede extenderse a problemas multiclase mediante estrategias como *One-vs-Rest*. Entre sus ventajas destaca su bajo coste computacional, su capacidad para manejar datos dispersos y la posibilidad de obtener predicciones en forma de probabilidad, lo que permite realizar análisis más matizados. Además, sus coeficientes ofrecen interpretaciones directas sobre la influencia de cada palabra o característica.

Sin embargo, presenta ciertas limitaciones. Su capacidad para modelar relaciones es exclusivamente lineal, por lo que puede resultar insuficiente ante estructuras lingüísticas complejas o emociones expresadas de forma implícita. Frente a modelos más avanzados como SVM o redes neuronales, puede quedarse atrás en precisión, especialmente en tareas donde los matices emocionales no se representan de forma lineal. Aun así, su simplicidad, transparencia y rendimiento competitivo la convierten en una herramienta muy valiosa en clasificación textual.

7.2.2. Parámetros aplicados

A continuación, se describen los principales parámetros [47] configurados para el modelo de `Logistic Regression`:

- **C**: Este parámetro controla la intensidad de la regularización. Cuanto menor sea su valor, mayor será la penalización aplicada a los coeficientes del modelo, lo que tiende a reducir el riesgo de sobreajuste. En este caso se ha fijado en 1.0, lo que representa una regularización estándar. Este valor ofrece un equilibrio adecuado entre permitir que el modelo aprenda patrones relevantes y evitar que se ajuste demasiado a los datos del entrenamiento, especialmente en contextos con alta dimensionalidad como el análisis de texto.
- **penalty**: Se refiere al tipo de regularización aplicado. Se ha optado por la regularización L2, que tiende a reducir el valor de los coeficientes sin anularlos completamente. Esta forma de regularización es adecuada cuando se trabaja con datos dispersos, como ocurre habitualmente con representaciones textuales del tipo bolsa de palabras o TF-IDF, ya que permite al modelo conservar una mayor cantidad de información útil.
- **solver**: El *solver* especifica el algoritmo de optimización que utilizará el modelo. En este caso, se ha escogido el método denominado `liblinear`. Este algoritmo es robusto y eficaz en conjuntos de datos de tamaño medio, y ofrece compatibilidad con la regularización L2 y con la opción de ponderar las clases automáticamente. Es una opción segura cuando se busca estabilidad en los resultados y una implementación bien soportada.
- **class_weight**: Este parámetro permite ajustar el peso relativo que se otorga a cada clase durante el entrenamiento. Se ha establecido como `balanced`, lo que significa que el modelo compensa de forma automática las diferencias en la frecuencia de las clases. Esta estrategia resulta especialmente importante en problemas de clasificación emocional, donde ciertas emociones son mucho más frecuentes que otras. Ponderar adecuadamente las clases ayuda a que el modelo no tienda a favorecer las categorías mayoritarias.
- **max_iter:** Finalmente, se ha definido un número máximo de iteraciones de mil para garantizar que el modelo disponga del tiempo suficiente para converger durante el proceso de ajuste. Este valor alto es útil cuando se trabaja con conjuntos de datos complejos o con muchas variables, ya que algunos algoritmos requieren más pasos para encontrar una solución estable.

7.2.3.Resultados

El modelo de regresión logística muestra un rendimiento sólido al analizar el *F1-score*, tal y como se observa en la figura 7.1. Los resultados indican que el modelo logra identificar correctamente las distintas emociones manteniendo una consistencia notable entre las clases, sin favorecer de manera significativa a ninguna en particular. Esto sugiere que la regresión logística, a pesar de su simplicidad y linealidad, es capaz de capturar patrones relevantes en los datos textuales, ofreciendo un desempeño competitivo cuando el conjunto de datos está bien representado y equilibrado. En este contexto, el *F1-score* confirma que el modelo no solo acierta frecuentemente, sino que también lo hace con una distribución de errores relativamente baja entre las emociones evaluadas.

Evaluación del modelo logistic regression:

	precision	recall	f1-score	support
anger	0.90	0.89	0.90	1540
fear	0.88	0.92	0.90	1285
joy	0.95	0.97	0.96	3786
sadness	0.95	0.92	0.94	3389
accuracy			0.93	10000
macro avg	0.92	0.93	0.92	10000
weighted avg	0.93	0.93	0.93	10000

Figura 7.1. Evaluación para logistic regression con set de datos principal

La matriz de confusión, que se aprecia en la figura 7.2, confirma el excelente rendimiento global reflejado por los F1-scores. Las clases se distinguen claramente, con errores reducidos y dispersos. No hay una tendencia a confundir todas las clases con una sola, lo cual indica un comportamiento más equilibrado y una comprensión más matizada de las diferentes emociones.

Además del *F1-score*, otras métricas complementarias refuerzan la solidez del modelo de regresión logística. El valor de ROC-AUC macro, calculado bajo un enfoque multiclase *One-vs-Rest*, alcanza un notable 0.9934, lo que evidencia una capacidad excepcional del modelo para distinguir entre las distintas emociones, incluso cuando estas presentan cierta superposición semántica. Este resultado sugiere que el clasificador mantiene un equilibrio eficaz entre sensibilidad y especificidad a lo largo de todas las clases. Por otro lado, el *log loss* también presenta un valor bajo de 0.3767, lo cual indica que no solo acierta en sus predicciones, sino que además lo hace con un alto grado de confianza. Esto revela una adecuada calibración probabilística, es decir, el modelo asigna probabilidades realistas y

consistentes a sus decisiones, lo que es especialmente importante en tareas donde la interpretación de la incertidumbre es clave.

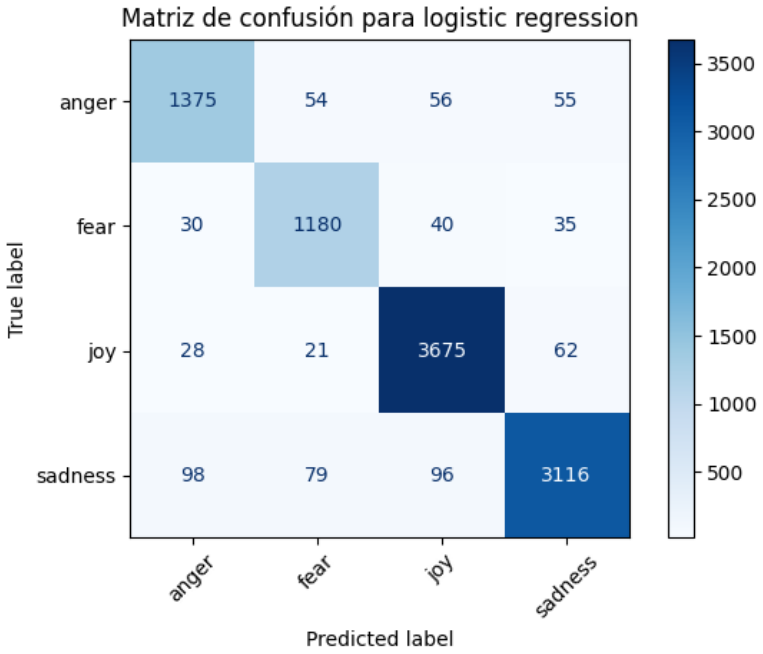

Figura 7.2. Matriz de confusión para logistic regression con set de datos principal

Las curvas de aprendizaje (ver figura 7.3) muestran una tendencia coherente con la de un modelo que mejora progresivamente a medida que dispone de más datos. La puntuación F1 en el conjunto de entrenamiento se mantiene alta y estable, mientras que la curva correspondiente al conjunto de validación crece de forma constante, alcanzando un valor cercano al 0.93. La brecha entre ambas se reduce, lo que indica que el modelo generaliza bien y no incurre en un sobreajuste severo. No obstante, también se observa que la tasa de mejora en la curva de validación empieza a atenuarse, lo que sugiere que el modelo podría estar aproximándose a un punto de saturación en su capacidad de aprendizaje con los datos disponibles. En consecuencia, aunque añadir más datos podría seguir aportando beneficios, estos serían previsiblemente más marginales, indicando que quizás sea momento de explorar mejoras arquitectónicas o metodológicas para seguir aumentando el rendimiento.

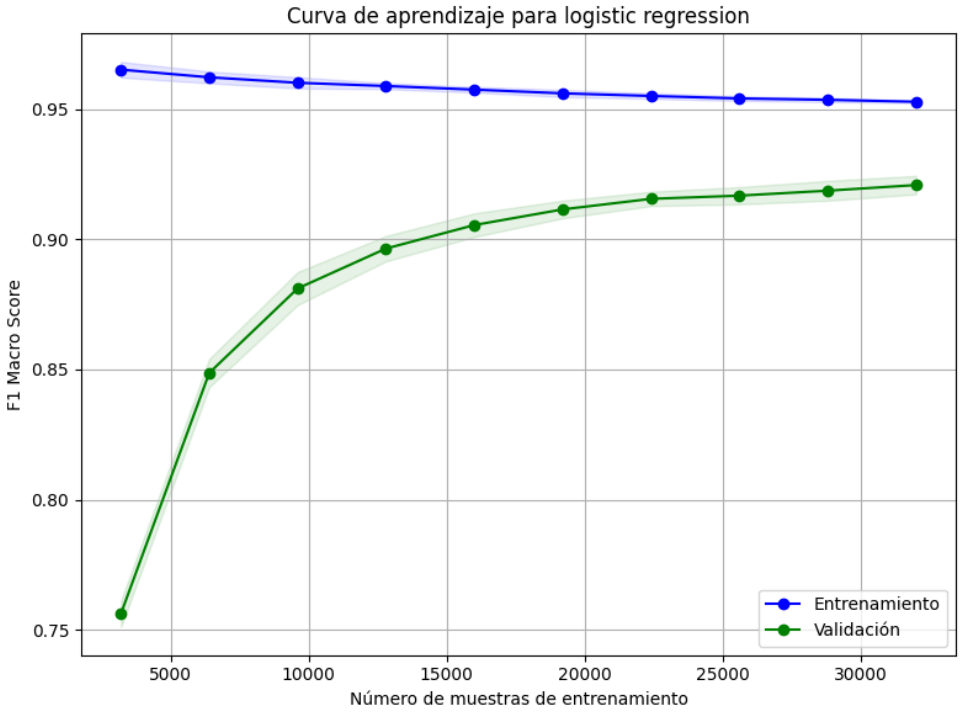

Figura 7.3. Curvas de aprendizaje para logistic regression

Al evaluar el modelo de regresión logística sobre un conjunto de datos externo no visto durante el entrenamiento, se observó una caída significativa en su capacidad de generalización, reflejada principalmente en la disminución del *F1-score macro* a 0.24, que se puede ver en la figura 7.4. Esta métrica indica que el modelo no logra mantener un rendimiento consistente al enfrentarse a datos que difieren del dominio de entrenamiento.

Evaluando logistic regression con dataset externo:

	precision	recall	f1-score	support
anger	0.17	0.15	0.16	275
fear	0.13	0.12	0.12	212
joy	0.38	0.41	0.39	704
sadness	0.30	0.28	0.29	550
accuracy			0.29	1741
macro avg	0.24	0.24	0.24	1741
weighted avg	0.29	0.29	0.29	1741

Figura 7.4. Evaluación para logistic regression con set de datos externo

Por otro lado, la figura 7.5, muestra la matriz de confusión revela que muchas instancias de las emociones enfado, miedo y tristeza son clasificadas erróneamente como alegría, que se convierte en la clase dominante en las predicciones. Esta tendencia puede estar relacionada con la forma en que el modelo ha aprendido a representar las emociones, alegría podría tener un patrón lingüístico más distintivo o estable frente a variaciones de contexto, mientras que las emociones negativas, más sutiles o contextualmente dependientes, presentan mayor ambigüedad. Este sesgo de predicción sugiere que el modelo es sensible a cambios en la distribución estilística o contextual del lenguaje, lo que refuerza la necesidad de incorporar estrategias de robustecimiento, como entrenamiento con datos más heterogéneos o técnicas de adaptación de dominio, para mejorar su desempeño en escenarios reales.

Además, las métricas ROC-AUC y *log loss* refuerzan esta conclusión: el valor AUC de 0.4909 es prácticamente equivalente a una clasificación aleatoria, y el *log loss* externo, de 1.9019, refleja una alta incertidumbre y falta de calibración en las predicciones. En conjunto, estos resultados confirman que, fuera del entorno para el que fue entrenado, el modelo pierde su fiabilidad tanto en la precisión de sus decisiones como en el grado de confianza que asigna a ellas.

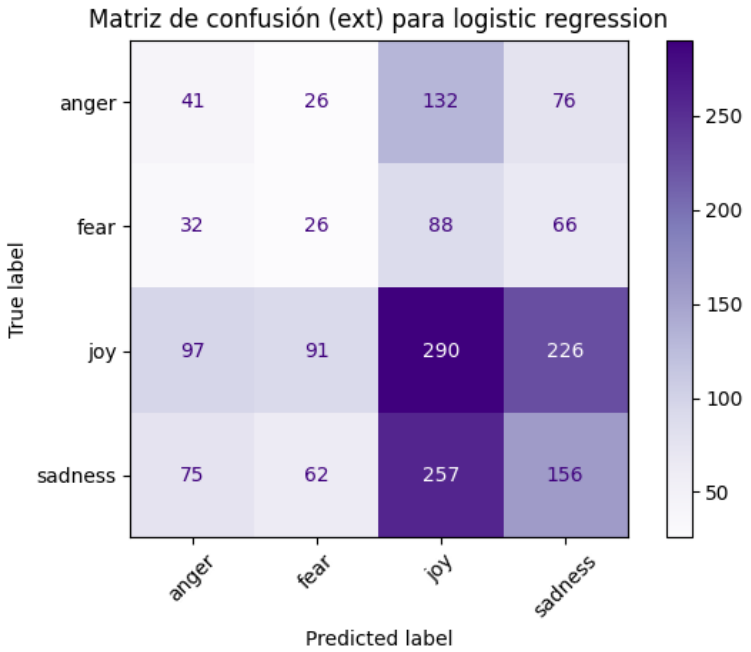

Figura 7.5. Matriz de confusión para logistic regression con set de datos externo

7.3. Naive Bayes Multinomial

7.3.1. Fundamentos y Funcionamiento

El clasificador Naive Bayes [48] es uno de los algoritmos más veteranos y eficaces dentro del aprendizaje automático supervisado, especialmente en tareas de clasificación de texto. Su funcionamiento se basa en el teorema de Bayes, que permite estimar la probabilidad de que un texto pertenezca a una clase determinada dado un conjunto de características observadas, como la presencia o frecuencia de palabras. La particularidad de este modelo es la suposición de independencia condicional entre atributos: se asume que, dadas las clases, las características son independientes entre sí. Aunque esta hipótesis raramente se cumple en la práctica, especialmente en lenguaje natural, la simplificación resultante permite una gran eficiencia computacional sin perder rendimiento en muchos escenarios.

En el contexto del procesamiento del lenguaje natural, el enfoque Multinomial Naive Bayes ha demostrado ser particularmente efectivo, ya que modela directamente la frecuencia de aparición de palabras en documentos, lo que lo hace muy adecuado para trabajar con representaciones como bolsas de palabras o TF-IDF. Así, frente a un texto, el modelo calcula para cada clase la probabilidad de que dicho texto haya sido generado por esa categoría, considerando la frecuencia de cada término.

Entre sus principales ventajas destacan su extrema rapidez tanto en el entrenamiento como en la predicción, su bajo coste computacional y su buen desempeño incluso en conjuntos de datos de alta dimensionalidad y dispersos. Además, funciona adecuadamente con pocos datos y ofrece resultados competitivos en contextos ruidosos o cuando se ha aplicado un buen preprocesamiento al texto.

No obstante, presenta limitaciones importantes. Su principal debilidad es la suposición de independencia entre palabras, que no refleja la naturaleza secuencial ni la estructura sintáctica del lenguaje. Como consecuencia, el modelo puede tener dificultades para capturar matices emocionales más sutiles o relaciones contextuales entre términos. Además, es sensible a clases desbalanceadas y a la mala distribución de datos.

A pesar de estas limitaciones, el Multinomial Naive Bayes continúa siendo una herramienta fundamental, tanto como línea base comparativa en experimentos como en aplicaciones reales donde la rapidez y la simplicidad son factores clave.

7.3.2. Parámetros aplicados

A continuación, se describen los principales parámetros [49] configurados para el modelo de `MultinomialNB`:

- **alpha**: Este parámetro controla el grado de suavizado de Laplace aplicado a las probabilidades estimadas. El suavizado es una técnica fundamental en modelos probabilísticos que asigna una pequeña probabilidad incluso a eventos no observados durante el entrenamiento, evitando así que una palabra desconocida anule la probabilidad total de una clase. Se ha optado por un valor de 0.5, que introduce un nivel moderado de suavizado. Esta elección busca un equilibrio entre conservar la influencia de las palabras realmente observadas y evitar penalizar en exceso a las clases por palabras ausentes, algo común en textos breves o variados como los utilizados en tareas de detección emocional.
- **fit_prior**: Este parámetro define si el modelo debe aprender las probabilidades previas de las clases a partir de los datos de entrenamiento. Se ha activado esta opción, es decir, se permite que el modelo estime automáticamente la distribución de clases. Esta decisión es coherente con el hecho de que las emociones no están uniformemente distribuidas en los textos, y conocer la frecuencia relativa de cada clase puede ayudar al modelo a ajustar sus predicciones de forma más realista, especialmente cuando algunas categorías emocionales son menos frecuentes.

7.3.3.Resultados

En el conjunto de entrenamiento, el modelo Multinomial Naive Bayes, tal y como se ve en la figura 7.6, logró un rendimiento sólido con un F1-score macro de 0.84. Esta métrica sugiere que, aunque no alcanza la precisión de la regresión logística, mantiene una buena capacidad para clasificar de forma equilibrada entre las distintas emociones. Destaca especialmente el excelente desempeño para las clases alegría y tristeza, con *F1-scores* de 0.92 y 0.89 respectivamente. Sin embargo, el modelo muestra mayor dificultad al clasificar enfado y miedo, que presentan valores de F1 más bajos, lo que indica cierta confusión entre emociones negativas, particularmente cuando comparten vocabulario o tono emocional. A pesar de esto, el rendimiento global es alto y coherente con la naturaleza probabilística del modelo.

Evaluación del modelo MultinomialNB:

	precision	recall	f1-score	support
anger	0.94	0.70	0.80	1540
fear	0.94	0.61	0.74	1285
joy	0.87	0.97	0.92	3786
sadness	0.84	0.94	0.89	3389
accuracy			0.87	10000
macro avg	0.90	0.81	0.84	10000
weighted avg	0.88	0.87	0.87	10000

Figura 7.6. Evaluación para multinomialNB con set de datos principal

La matriz de confusión (ver figura 7.7) evidencia claramente las limitaciones citadas. En el caso de enfado, de las 1540 muestras totales, 182 fueron clasificadas erróneamente como alegría y 255 como tristeza, lo que representa un desvío considerable hacia emociones más positivas o neutras. De forma similar, miedo muestra 200 clasificaciones incorrectas como alegría y 248 como tristeza, indicando una tendencia del modelo a desdibujar las emociones negativas y canalizarlas hacia clases más prevalentes o mejor representadas. Este comportamiento sugiere que, aunque `MultinomialNB` logra captar bien las emociones dominantes, enfrenta dificultades en mantener la separación semántica cuando las emociones comparten estructuras lingüísticas similares o aparecen en contextos ambiguos.

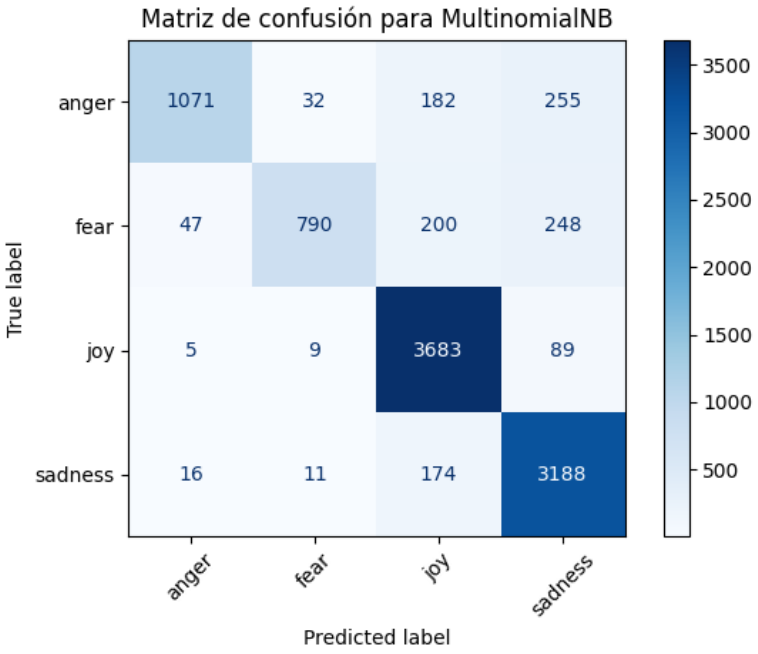

Figura 7.7. Matriz de confusión para multinomialNB con set de datos principal

En términos de discriminación y calibración, `MultinomialNB` también ofrece resultados positivos. El valor ROC-AUC macro alcanzó 0.9859, lo que indica que el modelo es competente al distinguir entre clases incluso cuando hay solapamientos emocionales. Por otro lado, el *log loss* fue de 0.5653, algo superior al de regresión logística, pero aún lo suficientemente bajo como para indicar que las predicciones son razonablemente confiables desde el punto de vista probabilístico. Es decir, aunque el modelo comete más errores, lo hace con cierta prudencia en la asignación de probabilidades, sin ser excesivamente confiado en decisiones erróneas.

Las curvas de aprendizaje, que se aprecian en la figura 7.8, para este modelo muestran una evolución más progresiva que en el caso anterior. La puntuación F1 en el conjunto de validación aumenta de forma continua con el tamaño del conjunto de entrenamiento, alcanzando un valor próximo a 0.84, pero con una brecha más pronunciada respecto al conjunto de entrenamiento. Esto sugiere que el modelo sigue aprendiendo al incorporar más datos, pero su capacidad está limitada por la naturaleza de la simplificación que impone su supuesto de independencia entre características. La mejora se va ralentizando a medida que crecen las muestras, lo que indica que, aunque escalar los datos ayuda, el modelo puede estar alcanzando un límite estructural de rendimiento.

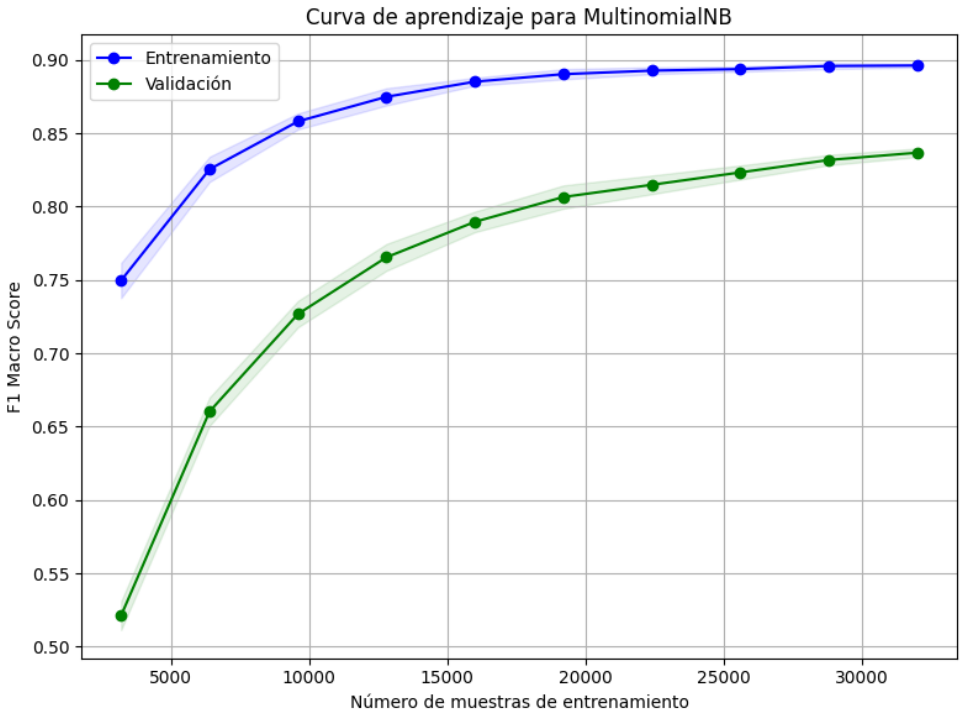

Figura 7.8. Curvas de aprendizaje para multinomialNB

Cuando se aplicó a un conjunto de datos externo, `MultinomialNB` sufrió una caída significativa en el rendimiento (ver figura 7.9), alcanzando un *F1-score* macro de apenas 0.24, al igual que regresión logística. Aunque la clase alegría mantiene una ventaja comparativa con un F1 de 0.42, las demás clases, especialmente enfado y miedo, presentan resultados muy bajos, con *F1-scores* de 0.12 y 0.09 respectivamente. Esta pérdida de rendimiento vuelve a evidenciar una dificultad para generalizar a dominios distintos al de entrenamiento. La matriz de confusión, que se puede ver en la figura 7.10, refuerza esta interpretación: una vez más, la mayoría de las emociones son clasificadas como alegría, que domina las predicciones del modelo. Esto, al igual que antes, es posible que se atribuya a una representación más estable de dicha emoción frente a cambios en estilo o contexto. La ROC-AUC se redujo a 0.4932, cercana al azar, y el *log loss* externo aumentó a 1.6335, revelando una gran pérdida de confianza y precisión en las probabilidades estimadas. Estos resultados refuerzan la necesidad de incluir mecanismos que permitan mayor adaptación al dominio, como reentrenamiento con datos externos o enfoques híbridos más sofisticados.

Evaluando MultinomialNB con dataset externo:

	precision	recall	f1-score	support
anger	0.15	0.10	0.12	275
fear	0.12	0.08	0.09	212
joy	0.39	0.46	0.42	704
sadness	0.30	0.32	0.31	550
accuracy			0.31	1741
macro avg	0.24	0.24	0.24	1741
weighted avg	0.29	0.31	0.30	1741

Figura 7.9. Evaluación para multinomialNB con set de datos externo

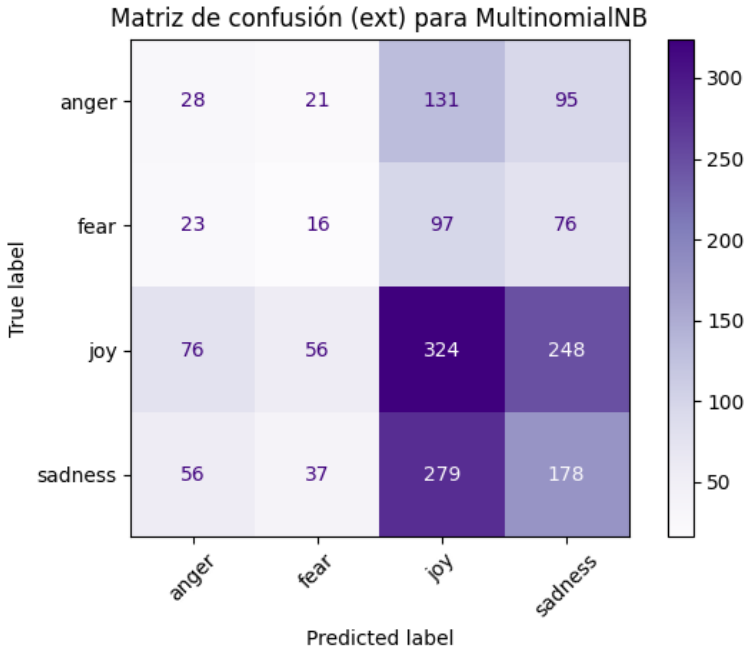

Figura 7.10. Matriz de confusión para multinomialNB con set de datos externo

7.4. Random Forest

7.4.1.Fundamentos y Funcionamiento

Random Forest [50] es un algoritmo de aprendizaje supervisado basado en el enfoque de ensamblado (*ensemble learning*), diseñado para mejorar la precisión de predicciones mediante la combinación de múltiples árboles de decisión. Su funcionamiento se apoya en la idea de que, aunque un solo árbol pueda ser inestable o propenso al sobreajuste, un conjunto de árboles diversos puede generalizar mejor los patrones subyacentes del conjunto de datos. Cada árbol se entrena sobre una muestra aleatoria del conjunto original, mediante la técnica de *bagging*, y en cada división del árbol solo se consideran un subconjunto aleatorio de características, lo que introduce variedad entre los árboles y reduce la correlación entre ellos.

El resultado final en tareas de clasificación se obtiene por votación mayoritaria de todos los árboles, mientras que en regresión se toma el promedio de sus predicciones. Esta estrategia permite a Random Forest manejar con eficacia datos de alta dimensionalidad, como aquellos representados mediante bolsas de palabras o TF-IDF en procesamiento de texto, y ser resistente al ruido y a las variables irrelevantes. Además, permite estimar la importancia relativa de cada característica en la predicción, proporcionando información útil sobre qué atributos contribuyen más al rendimiento del modelo.

Entre sus principales ventajas se encuentra su capacidad para capturar relaciones no lineales y combinaciones complejas de atributos, su robustez frente a datos ruidosos o desbalanceados y su buena adaptabilidad a distintos tipos de datos. También es menos sensible a la necesidad de ajuste fino de parámetros en comparación con otros modelos avanzados.

Sin embargo, también presenta limitaciones. Su entrenamiento puede ser computacionalmente costoso cuando se incrementa el número de árboles o la cantidad de datos, y al tratarse de un conjunto de modelos, su interpretabilidad se ve reducida en comparación con un árbol individual. Además, su rendimiento puede verse afectado negativamente al trabajar con vectores de texto muy dispersos, como ocurre con ciertas representaciones de texto en NLP, a menos que se aplique una adecuada selección o reducción de características.

En definitiva, Random Forest ofrece una combinación sólida de precisión, resiliencia y flexibilidad, lo que lo convierte en una opción muy útil como modelo de referencia o punto de partida en problemas de clasificación de texto, incluido el reconocimiento de emociones.

7.4.2. Parámetros aplicados

A continuación, se describen los principales parámetros [51] configurados para el modelo de `RandomForestClassifier`:

- **n_estimators**: Este parámetro indica el número total de árboles que compondrán el bosque. A mayor número de árboles, el modelo tiende a ser más estable y preciso, ya que las predicciones se basan en un mayor consenso. En este caso, se han empleado 200 árboles, un valor suficientemente alto como para asegurar una buena generalización, pero sin comprometer en exceso el tiempo de entrenamiento.
- **max_depth**: Controla la profundidad máxima que puede alcanzar cada árbol individual. Limitar esta profundidad es una medida efectiva para evitar el sobreajuste, ya que árboles demasiado profundos pueden memorizar los datos de entrenamiento. Se ha fijado un valor de 30, que permite capturar relaciones complejas sin perder capacidad de generalización.
- **max_features**: Define cuántas características se consideran aleatoriamente en cada división de los nodos. Se ha utilizado la raíz cuadrada del número total de características, una estrategia habitual y eficaz en conjuntos de texto vectorizados como los generados con TF-IDF, donde la dimensionalidad es alta y se busca introducir diversidad entre los árboles.
- **class_weight**: Este ajuste, como ya se ha dicho, permite compensar desequilibrios en la distribución de clases. Al emplear la opción de pesos balanceados, el modelo otorga mayor importancia a las clases menos representadas, lo cual es fundamental en tareas de clasificación emocional donde algunas emociones pueden aparecer con mucha menor frecuencia que otras.

Además, se ha utilizado un valor fijo en la semilla aleatoria para garantizar la reproducibilidad de los resultados, y se ha habilitado el uso de todos los núcleos del procesador para acelerar el entrenamiento en entornos *multicore*.

7.4.3.Resultados

El modelo Random Forest, entrenado sobre el conjunto principal, muestra un rendimiento sólido, como se puede ver en la figura 7.11, con un *F1-score* macro de 0.87, comparable al obtenido con regresión logística. Las clases miedo y alegría son particularmente bien reconocidas, alcanzando *F1-scores* de 0.87 y 0.89, respectivamente. Por su parte, enfado y tristeza también obtienen buenos resultados, ambos con un F1 cercano a 0.85. Este desempeño equilibrado entre clases revela que el modelo es capaz de capturar patrones relevantes en distintos tipos de emociones, gracias a su capacidad de modelar relaciones no lineales entre características. La puntuación ROC-AUC de 0.9712 confirma la buena discriminación entre clases, mientras que el valor *log loss* de 1.1224, si bien más alto que en otros modelos, sugiere que el modelo es algo menos calibrado en cuanto a la confianza de sus predicciones, a pesar de acertar frecuentemente.

Evaluación del modelo random forest:

	precision	recall	f1-score	support
anger	0.89	0.81	0.85	1540
fear	0.85	0.90	0.87	1285
joy	0.83	0.97	0.89	3786
sadness	0.95	0.79	0.86	3389
accuracy			0.87	10000
macro avg	0.88	0.87	0.87	10000
weighted avg	0.88	0.87	0.87	10000

Figura 7.11. Evaluación para random forest con set de datos principal

La matriz de confusión (ver figura 7.12) refuerza estas observaciones. La clase alegría es nuevamente la mejor clasificada con 3659 aciertos de 3786, mientras que miedo también muestra un buen desempeño con 1162 aciertos sobre 1285 muestras. En contraste, tristeza, aunque presenta un alto número de verdaderos positivos (2676), evidencia una notable confusión con alegría (512 casos), lo que afecta su *recall*. También se observa que enfado se confunde principalmente con alegría, más concretamente en 164 casos, aunque conserva una proporción aceptable de predicciones correctas. En general, el modelo mantiene un rendimiento robusto y bastante equilibrado, aunque muestra una tendencia a confundir emociones con componentes afectivos parcialmente solapados, como tristeza y alegría.

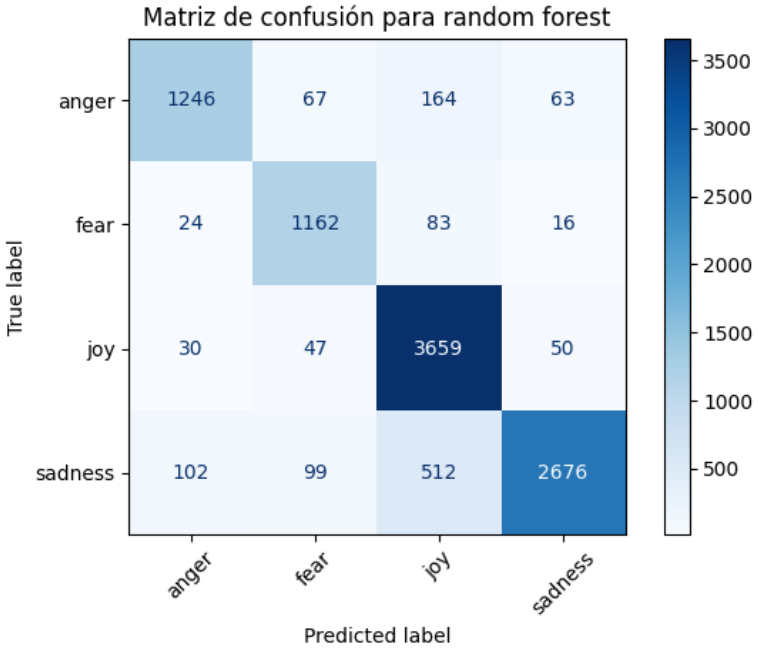

Figura 7.12. Matriz de confusión para random forest con set de datos principal

Por otro lado, la figura 7.13 muestra la curva de aprendizaje del modelo. Se observa cómo el *F1-score* en entrenamiento desciende de forma gradual conforme aumenta el número de muestras, mientras que la curva de validación se mantiene prácticamente estable tras una mejora inicial, en torno a un valor de 0.86. Esto indica que el modelo generaliza de forma consistente en el dominio original, aunque la ligera separación entre ambas curvas sugiere que podría beneficiarse de una mayor cantidad de datos anotados.

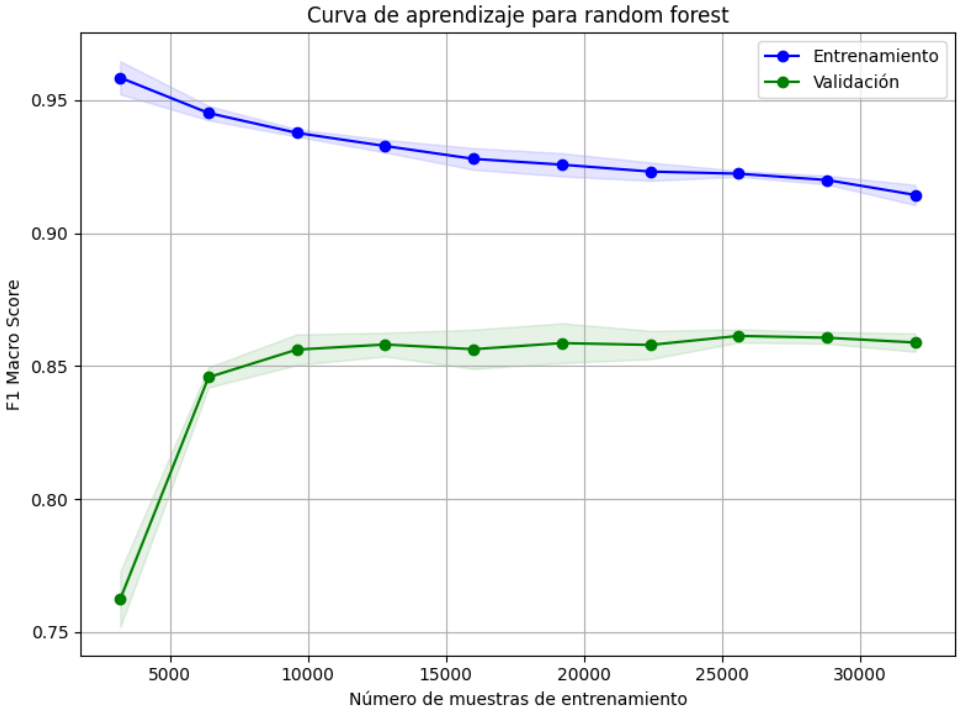

Figura 7.13. Curvas de aprendizaje para random forest

Al evaluar el modelo con un set de datos externo, su rendimiento cae significativamente, logrando apenas, como se puede ver en la figura 7.14, un *F1-score macro* de 0.24, similar al observado con otros modelos en condiciones fuera de dominio. Alegría vuelve a ser la clase con mayor capacidad de recuperación, mientras que enfado y miedo quedan rezagadas con 0.15 y 0.11, respectivamente. Este patrón de desempeño se confirma en la matriz de confusión (ver figura 7.15), donde una gran parte de las predicciones de enfado, miedo y triste son absorbidas por la clase alegría, que actúa como una especie de "imán" emocional, acumulando predicciones incorrectas.

La puntuación ROC-AUC de 0.4994 indica una pérdida total de discriminación, situándose en el nivel del azar, y un *log loss* 1.3737 refleja una baja confianza y mala calibración en las asignaciones probabilísticas. Estos resultados vuelven a poner en evidencia la escasa generalización del modelo fuera del dominio original, acentuando la importancia de trabajar con datos más diversos o aplicar técnicas de adaptación de dominio si se desea aplicar el modelo en entornos reales heterogéneos.

Evaluando random forest con dataset externo:

	precision	recall	f1-score	support
anger	0.17	0.14	0.15	275
fear	0.12	0.11	0.11	212
joy	0.40	0.50	0.44	704
sadness	0.30	0.23	0.26	550
accuracy			0.31	1741
macro avg	0.25	0.25	0.24	1741
weighted avg	0.30	0.31	0.30	1741

Figura 7.14. Evaluación para random forest con set de datos externo

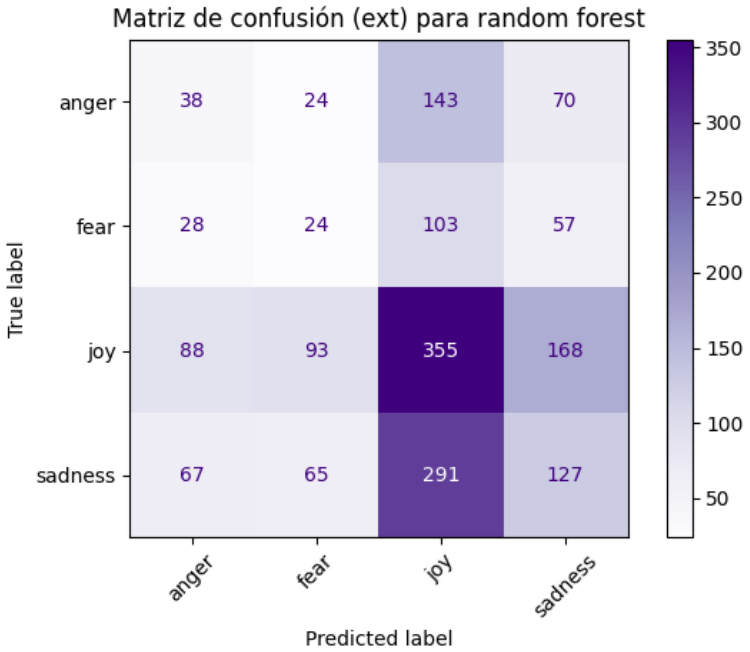

Figura 7.15. Matriz de confusión para random forest con set de datos externo

7.5. Support Vector Machines (SVM)

7.5.1.Fundamentos y Funcionamiento

Los *Support Vector Machines* (SVM) [52] son una familia de algoritmos ampliamente utilizados para tareas de clasificación, conocidos por su capacidad de generalizar bien incluso en contextos complejos y con pocos datos. El principio fundamental que guía a estos modelos es la búsqueda del hiperplano óptimo: una frontera que separe las clases de manera que se maximice el margen, es decir, la distancia entre dicho hiperplano y los puntos más cercanos de cada clase. Estos puntos limítrofes se denominan vectores soporte y son los que determinan la posición y orientación del hiperplano.

Maximizar el margen no solo mejora la separación entre clases, sino que también proporciona una mayor capacidad de generalización, reduciendo el riesgo de sobreajuste frente al ruido o variaciones menores en los datos de entrenamiento. Para problemas en los que las clases no son linealmente separables, SVM recurre al llamado "truco del kernel", una técnica que transforma los datos originales a un espacio de mayor dimensión donde sí pueden separarse mediante un hiperplano. Esta transformación se realiza sin calcular explícitamente las nuevas coordenadas, lo que permite una eficiencia sorprendente en problemas no lineales. Entre los kernels más utilizados destacan el lineal, el polinómico y el RBF (*Radial Basis Function*), especialmente eficaz en clasificación compleja.

SVM destaca por su rendimiento en conjuntos de datos con alta dimensionalidad y bajo número de muestras, como ocurre a menudo en tareas de procesamiento de lenguaje natural.

Entre sus principales ventajas, SVM ofrece una excelente capacidad de generalización cuando se trabaja con representaciones vectorizadas del texto, como TF-IDF, y muestra buenos resultados en conjuntos de tamaño reducido o con clases parcialmente solapadas. Además, su formulación robusta permite trabajar con datos ruidosos de forma eficaz. Sin embargo, también presenta limitaciones, como el elevado coste computacional al escalar a grandes volúmenes de datos o la ausencia de salidas probabilísticas nativas, lo que puede dificultar ciertos análisis. Asimismo, su menor interpretabilidad frente a modelos más simples puede ser un inconveniente en entornos donde esa interpretabilidad del modelo es crucial.

En conjunto, SVM constituye una herramienta poderosa y versátil en clasificación textual, especialmente útil cuando se requiere precisión en contextos complejos y con estructuras no lineales.

7.5.2. Parámetros aplicados

A continuación, se describen los principales parámetros [53] configurados para el modelo de SVC:

- **C:** Este parámetro regula el equilibrio entre un margen de separación amplio y el número de errores de clasificación permitidos. Un valor más alto obliga al modelo a clasificar correctamente la mayor cantidad de puntos posible, mientras que uno más bajo permite mayor tolerancia a errores a cambio de un margen más amplio. En este trabajo se ha empleado el valor estándar de uno, que proporciona un punto de partida sólido para evitar tanto el sobreajuste como un margen excesivamente permisivo.
- **kernel:** Define el tipo de transformación aplicada al espacio de características antes de encontrar el hiperplano de separación. Se ha elegido un kernel lineal, ya que es especialmente adecuado para texto vectorizado mediante técnicas como TF-IDF, donde los datos suelen ser dispersos, pero linealmente separables. Esta elección favorece además un entrenamiento más rápido y una mayor interpretabilidad del modelo.
- **class_weight:** Este parámetro, al igual que en otros modelos, permite asignar pesos diferentes a cada clase con el fin de compensar desequilibrios en la distribución de los datos. Al aplicar un ajuste automático basado en la frecuencia de cada clase, se evita que el modelo favorezca sistemáticamente las clases mayoritarias, lo cual es crucial en tareas de clasificación emocional con clases desbalanceadas.
- **probability:** Habilita el cálculo posterior de probabilidades calibradas asociadas a las predicciones. Aunque este proceso introduce un coste computacional adicional, resulta necesario para obtener métricas probabilísticas como el *log loss* o el AUC, que permiten una evaluación más matizada del comportamiento del modelo.

7.5.3.Resultados

El modelo SVM ha mostrado un rendimiento sobresaliente sobre el conjunto de entrenamiento principal. En términos de métricas, como se muestra en la figura 7.16, ha alcanzado un *F1-score macro* de 0.93, siendo el más alto entre todos los modelos evaluados.

Evaluación del modelo SVM:

	precision	recall	f1-score	support
anger	0.87	0.92	0.89	1540
fear	0.87	0.95	0.90	1285
joy	0.97	0.96	0.97	3786
sadness	0.97	0.91	0.94	3389
accuracy			0.94	10000
macro avg	0.92	0.93	0.93	10000
weighted avg	0.94	0.94	0.94	10000

Figura 7.16. Evaluación para SVM con set de datos principal

La matriz de confusión de la figura 7.17 revela que las predicciones son muy precisas en todas las clases. Las confusiones más frecuentes se dan entre tristeza y enfado, aunque en una magnitud mucho menor comparado con modelos anteriores. Esto se refleja también en la alta precisión y *recall* por clase: alegría alcanza un *F1-score* de 0.97, mientras que tristeza y miedo obtienen 0.94 y 0.90 respectivamente. Además, el valor de ROC-AUC de 0.9910 y un bajo *log loss* de 0.1947 refuerzan la robustez del modelo sobre datos vistos.

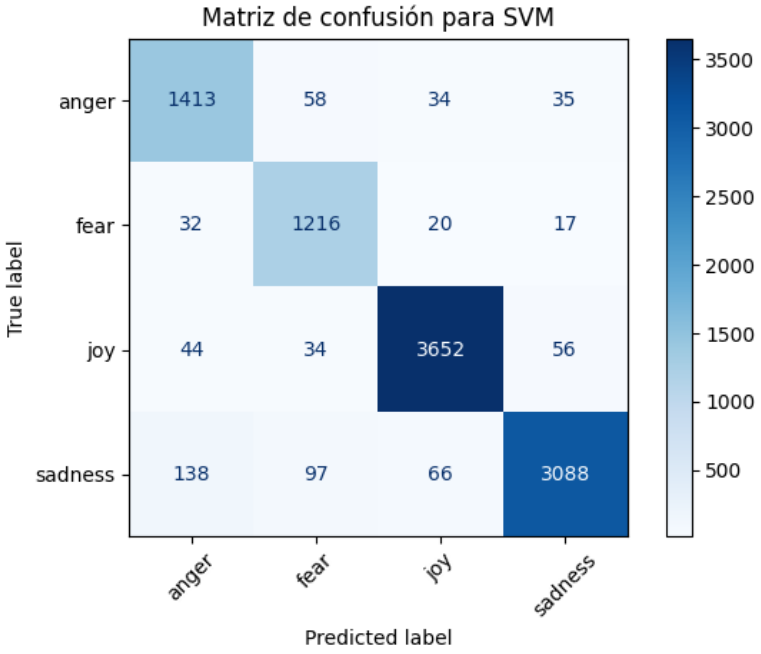

Figura 7.17. Matriz de confusión para SVM con set de datos principal

La curva de aprendizaje (ver figura 7.18) del modelo SVM confirma un comportamiento altamente favorable: mientras la puntuación F1 en entrenamiento se mantiene cercana a 0.96, la validación se estabiliza por encima del 0.91 a medida que aumentan las muestras, lo que indica buena capacidad de generalización dentro del conjunto original. No obstante, también se observa que el rendimiento en validación tiende a estabilizarse, lo que sugiere que el modelo podría estar alcanzando un límite en cuanto a la ganancia adicional de rendimiento con más datos, al menos bajo esta configuración actual.

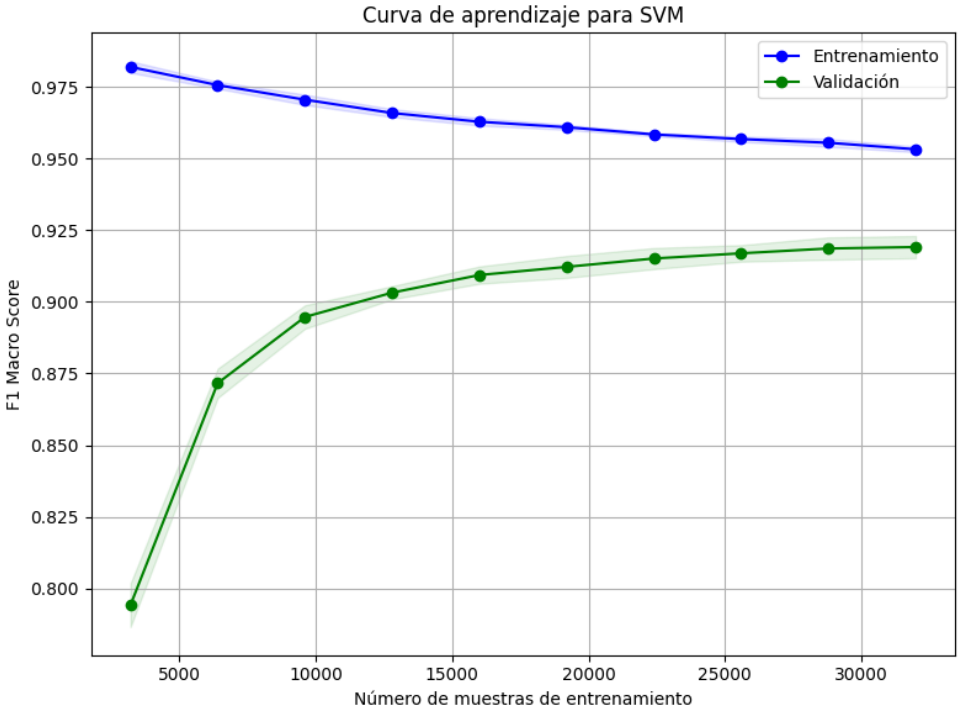

Figura 7.18. Curvas de aprendizaje para SVM

El comportamiento en el set de datos externo, que se puede visualizar en las figura 7.19 y figura 7.20 evidencia nuevamente una caída considerable. Aunque las métricas no son tan bajas como en algunos modelos previos, el *F1-score macro* desciende a 0.24, y la precisión se ve comprometida especialmente en clases como miedo y enfado, donde el modelo apenas supera un 15% de precisión. A pesar de que alegría se mantiene como la clase mejor reconocida, sigue habiendo una tendencia del modelo a predecir excesivamente esta clase, como lo muestra la matriz de confusión. Esto sugiere un posible sobreajuste al dominio del conjunto de entrenamiento. El ROC-AUC externo cae hasta 0.4935, y el *log loss* externo asciende a 5.0933, lo cual confirma la alta incertidumbre del modelo fuera de su dominio original.

Evaluando SVM con dataset externo:

	precision	recall	f1-score	support
anger	0.16	0.15	0.16	275
fear	0.13	0.14	0.13	212
joy	0.39	0.43	0.41	704
sadness	0.30	0.26	0.28	550
accuracy			0.30	1741
macro avg	0.25	0.25	0.24	1741
weighted avg	0.29	0.30	0.29	1741

Figura 7.19. Evaluación para SVM con set de datos externo

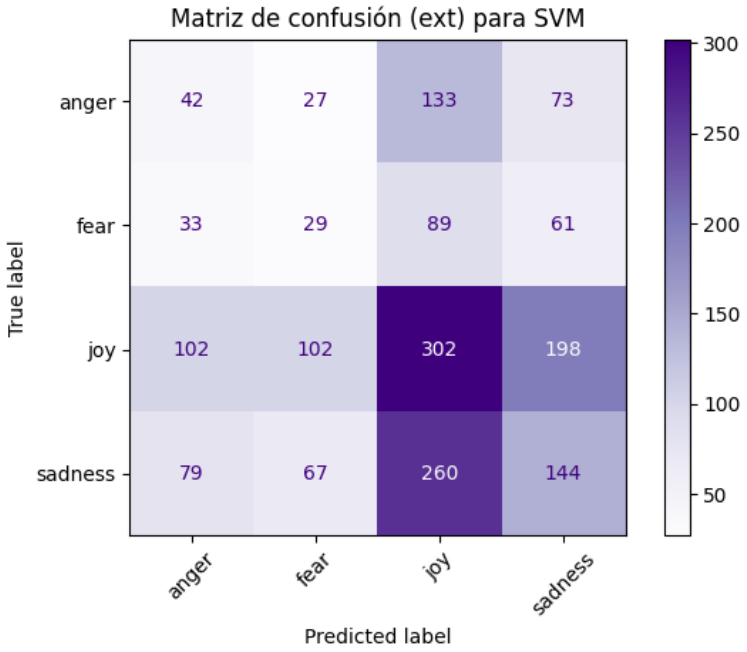

Figura 7.20. Matriz de confusión para SVM con set de datos externo

7.6. Prueba con enfoque Deep Learning usando PyTorch

Aunque en este trabajo se ha planteado que el uso de técnicas de *Deep Learning* no es viable debido a las limitaciones en la cantidad de datos y la capacidad computacional disponible, se ha realizado una prueba práctica con un modelo de *Deep Learning* para contrastar resultados y entender mejor las diferencias en rendimiento y recursos requeridos.

Para esta prueba se ha utilizado la librería PyTorch, reconocida por su potencia y flexibilidad en el desarrollo de modelos de *Deep Learning*. En concreto, se ha empleado el modelo `bert-base-uncased`, una versión preentrenada del modelo BERT [54] que se ha convertido en uno de los referentes en tareas avanzadas de procesamiento de lenguaje natural. BERT se caracteriza por su arquitectura basada en transformadores, capaz de capturar contextos complejos y relaciones semánticas profundas en el texto.

7.6.1. Especificaciones del equipo utilizado

Dado que el entrenamiento de modelos de *Deep Learning*, especialmente modelos grandes como BERT, requiere una gran cantidad de recursos computacionales, resulta fundamental contextualizar la prueba realizada con las capacidades reales del sistema utilizado. Esta información permite valorar la viabilidad práctica del uso de modelos complejos en entornos no especializados, y proporciona una referencia para futuras investigaciones.

Las pruebas se llevaron a cabo en un equipo con las siguientes características técnicas:

- Sistema operativo: Windows 11 Pro 64-bit
- Procesador: Intel i7-14700KF [55]
- Memoria RAM: 32 GB DDR4
- Placa base: MSI PRO B760-P DDR4 II [56]
- Tarjeta Gráfica: NVIDIA GeForce RTX 4070 SUPER con 12 GB de VRAM [57]

Este entorno proporciona una capacidad computacional muy por encima de la media en sistemas personales, especialmente gracias a la GPU RTX 4070 SUPER, que ofrece una excelente potencia de procesamiento paralelo para tareas intensivas como el entrenamiento de modelos BERT.

Aprovechando la potencia del equipo, se ha decidido utilizar una muestra de 50.000 ejemplos del conjunto de datos original para entrenar el modelo BERT. Esta decisión responde a la necesidad de comprobar hasta qué punto un modelo de *Deep Learning* como BERT puede superar a los enfoques clásicos de *Machine Learning* en tareas de clasificación emocional, especialmente cuando se dispone de un volumen de datos suficiente.

7.6.2. Resultados

El modelo preentrenado `bert-base-uncased` fue ajustado mediante *fine-tuning* para la tarea específica de detección de emociones en texto. Se entrenó el modelo durante tres épocas, cuyos resultados se pueden apreciar en la figura 7.21, figura 7.22 y figura 7.23 completas sobre el conjunto de entrenamiento:

	precision	recall	f1-score	support
0	0.95	0.92	0.94	1540
1	0.94	0.95	0.95	1285
2	1.00	1.00	1.00	3786
3	0.97	0.98	0.98	3389
accuracy			0.97	10000
macro avg	0.97	0.96	0.96	10000
weighted avg	0.97	0.97	0.97	10000

Figura 7.21. Evaluación de la primera época de entrenamiento BERT

	precision	recall	f1-score	support
0	0.94	0.94	0.94	1540
1	0.96	0.93	0.95	1285
2	0.99	0.99	0.99	3786
3	0.97	0.98	0.97	3389
accuracy			0.97	10000
macro avg	0.97	0.96	0.96	10000
weighted avg	0.97	0.97	0.97	10000

Figura 7.22. Evaluación de la segunda época de entrenamiento BERT

	precision	recall	f1-score	support
0	0.97	0.92	0.94	1540
1	0.91	0.99	0.95	1285
2	1.00	0.99	0.99	3786
3	0.97	0.97	0.97	3389
accuracy			0.97	10000
macro avg	0.96	0.97	0.96	10000
weighted avg	0.97	0.97	0.97	10000

Figura 7.23. Evaluación de la tercera época de entrenamiento BERT

El orden de las clases en el modelo BERT se mantiene consistente con el utilizado en los modelos clásicos de *Machine Learning*. En concreto, las etiquetas se corresponden con las siguientes emociones:

0 → enfado, 1 → miedo, 2 → alegría, 3 → tristeza.

Los resultados obtenidos mostraron una mejora notable en la precisión del modelo utilizando *Deep Learning*, superando claramente a los modelos tradicionales de *Machine Learning* implementados previamente. Esto confirma el potencial de los modelos basados en transformadores para capturar matices emocionales en el texto que otros enfoques no logran detectar con la misma eficacia.

No obstante, los tiempos de entrenamiento han sido considerablemente elevados, incluso con una GPU de última generación como la RTX 4070 SUPER, lo que refleja las altas exigencias computacionales de este tipo de modelos. Esta circunstancia refuerza la idea de que, aunque los modelos basados en transformadores ofrecen un rendimiento superior, su aplicación práctica está condicionada por la disponibilidad de recursos técnicos adecuados.

Además, el uso de modelos como BERT requiere no solo hardware especializado, como GPU o TPU, sino también un nivel técnico avanzado para la correcta configuración, ajuste y evaluación del modelo. Esto representa una barrera importante para proyectos con recursos limitados o que requieren ciclos rápidos de entrenamiento y evaluación.

7.7. Conclusiones de los resultados

Al analizar de manera conjunta el comportamiento de los modelos evaluados se revelan una serie de coincidencias notables, así como diferencias específicas que ayudan a comprender no solo cómo funcionan internamente, sino también por qué fallan cuando se enfrentan a datos que no pertenecen al dominio original. Cabe destacar que todos los modelos han sido ajustados utilizando los mejores parámetros encontrados tras un proceso de optimización cuidadoso, lo que garantiza que el rendimiento observado refleja su máximo potencial dentro del enfoque seleccionado. Esta evaluación, por tanto, no solo compara arquitecturas, sino también su adaptabilidad y robustez en condiciones realistas. La tabla 7.1 presenta un resumen consolidado de los resultados obtenidos, ofreciendo una visión más sintética y comparativa del rendimiento de los distintos modelos evaluados. Para ello, se recogen las métricas numéricas más relevantes analizadas a lo largo del trabajo: el *F1-score*, la puntuación ROC-AUC, el *log loss* y el número de aciertos y fallos de las matrices de confusión. De la misma forma, lo hace la tabla 7.2 para la evaluación del modelo con el set de datos externo.

Modelos/Métricas	*F1-score*	*ROC-AUC*	*Log loss*	*Aciertos*	*Fallos*
Regresión Logística	0.92	0.9934	0.3767	9346	654
Naive Bayes Multinomial	0.90	0.9859	0.5653	8732	1268
Random Forest	0.88	0.9712	1.1224	8743	1257
Support Vector Machines	0.92	0.9910	0.1947	9369	631
BERT	0.97	-	-	-	-

Tabla 7.2. Tabla de resultados con el set de datos principal

Modelos/Métricas (ext)	*F1-score*	*ROC-AUC*	*Log loss*	*Aciertos*	*Fallos*
Regresión Logística	0.24	0.4909	1.9019	513	1228
Naive Bayes Multinomial	0.24	0.4932	1.6335	536	1205
Random Forest	0.25	0.4994	1.3737	544	1197
Support Vector Machines	0.25	0.4935	5.0933	517	1224
BERT	-	-	-	-	-

Tabla 7.3. Tabla de resultados con el set de datos externo

Coincidencias clave entre los modelos

Una de las coincidencias más evidentes es que todos los modelos, sin excepción, muestran un buen rendimiento al trabajar con el conjunto de datos original. Las puntuaciones *F1 macro* se mantienen consistentemente altas, lo que indica que estos clasificadores son capaces de aprender y reproducir correctamente las distribuciones del corpus con el que fueron entrenados. Las métricas por clase refuerzan esta impresión: tanto alegría como tristeza son detectadas con gran precisión, lo que sugiere que estas emociones están representadas de manera clara y consistente en el conjunto original.

Sin embargo, también comparten un fenómeno que podría considerarse una debilidad: el predominio absoluto de la clase alegría. En todos los modelos, esta emoción no solo es la más reconocida, sino también la más predicha, incluso cuando no corresponde. Al observar las matrices de confusión del conjunto externo, es evidente que una gran parte de los errores se produce cuando el modelo asigna erróneamente la etiqueta alegría a textos que en realidad expresan enfado o miedo. Este patrón de comportamiento sugiere la presencia de un sesgo estructural, posiblemente derivado del estilo del corpus original, donde alegría podría estar representada de forma más explícita y frecuente, haciendo que los modelos tiendan a ella en situaciones de ambigüedad.

Otra similitud importante es la drástica caída de rendimiento cuando se enfrentan al set de datos externo. Mientras que en el entorno controlado del entrenamiento los modelos alcanzan métricas sobresalientes, al cambiar de dominio las puntuaciones se desploman: el F1 macro cae a valores cercanos al 0.24, y el ROC-AUC apenas supera el 0.49, lo cual se aproxima al

comportamiento de un modelo aleatorio. Esta caída generalizada pone en evidencia una limitación compartida: los modelos han aprendido muy bien los patrones específicos del conjunto original, pero han fallado en capturar una comprensión más profunda y generalizable de las emociones humanas expresadas en lenguaje natural.

Además, todos muestran serias dificultades para identificar emociones como enfado y miedo. La precisión para estas clases se reduce por debajo del 15% en el conjunto de datos externo, un comportamiento consistente en los cuatro modelos. Estas emociones, posiblemente más complejas o contextualmente dependientes, resultan ser mucho más difíciles de detectar con las representaciones actuales, que no logran capturar bien los matices que las definen.

Diferencias y particularidades de cada modelo

A pesar de estas similitudes estructurales, cada modelo también muestra comportamientos propios que merecen ser destacados. El modelo regresión logística, por ejemplo, se comporta de manera bastante estable y balanceada. Aunque no es el mejor en ninguna métrica específica, su rendimiento es consistente y sus predicciones están mejor calibradas, como lo demuestra su bajo valor de *log loss* externo. Esta característica sugiere que, aunque pueda acertar menos veces que otros modelos, es más honesto en sus predicciones probabilísticas, mostrando mayor humildad ante la incertidumbre.

Por su parte, el modelo de Random Forest se distingue por su comportamiento más errático en cuanto a confianza. Tiene el *log loss* más alto en el set de datos original, lo que indica que, aunque acierta, lo hace con menos seguridad. También muestra una tendencia a confundir tristeza con alegría, lo cual puede explicarse por la forma en que construye árboles de decisión: si el vocabulario emocional no está bien representado para ciertas clases, el modelo puede tender a generalizar en exceso, apoyándose en correlaciones poco robustas.

El modelo SVM, en cambio, es el que logra los mejores resultados en el entorno controlado. Sus métricas son sobresalientes, y la curva de aprendizaje muestra un crecimiento constante conforme se incrementa el tamaño del set de datos. No obstante, también es el que sufre la caída más brusca al enfrentarse al conjunto de datos externo: el *log loss* externo asciende a 5.0933, indicando una alta incertidumbre y falta de calibración fuera de su dominio. Esto sugiere un caso clásico de sobreajuste estructural: el modelo ha aprendido patrones muy específicos del corpus original, que no se replican en los nuevos textos. Además, SVM presenta una desventaja práctica importante: es el modelo que más tiempo tarda en entrenarse, ya que requiere una considerable potencia computacional, especialmente en tareas multiclase y con vectores de entrada de alta dimensión, como los generados por técnicas de NLP. Esto lo convierte en una opción potente pero costosa, tanto en recursos como en tiempo, lo cual puede ser un factor limitante en contextos con infraestructuras modestas o

necesidades de respuesta rápida. En contraste, los demás modelos evaluados presentan tiempos de entrenamiento muy bajos y requisitos del sistema mínimos, lo que los hace más accesibles en entornos con limitaciones técnicas.

Finalmente, el modelo de Naive Bayes, aunque más simple, se comporta de forma bastante coherente con sus limitaciones. Internamente alcanza un *F1 macro* aceptable, pero sufre mucho más que los otros al salir del entorno de entrenamiento. Esto es comprensible, ya que se basa en la independencia de características y depende fuertemente del vocabulario: cualquier cambio en las palabras utilizadas afecta gravemente su rendimiento. Aun así, su inclusión permite tener una línea base útil para comparar la eficacia de los modelos más complejos.

Reflexiones sobre la representación y la generalización

Este estudio pone también de relieve algunas limitaciones inherentes a la representación textual utilizada. El enfoque TF-IDF, si bien útil y de bajo coste computacional, no es capaz de capturar contexto ni relaciones semánticas complejas. Funciona bien cuando el dominio es estable y las palabras mantienen patrones predecibles, pero se vuelve ineficaz cuando el vocabulario o el estilo cambian, como ocurre con el set de datos externo. Esta rigidez en la representación explica en buena medida por qué todos los modelos, sin importar su complejidad, fallan al enfrentarse a un lenguaje ligeramente distinto.

Además, queda claro que los modelos no han logrado aprender una comprensión profunda y flexible de las emociones humanas, sino que se han apoyado en patrones léxicos muy concretos. El hecho de que todos ellos compartan errores similares sugiere que lo que han aprendido es más un reflejo del estilo del corpus original que una interpretación semántica robusta.

Capítulo 8. Conclusiones y trabajo futuro

En este último capítulo se procederá a una revisión de los objetivos planteados al inicio del proyecto, con el fin de analizar en qué medida han sido alcanzados, así como detectar posibles limitaciones o excepciones. Además, se expondrán diversas líneas de trabajo futuro que podrían derivarse de esta investigación. Dado el carácter exploratorio del presente trabajo, resulta especialmente relevante identificar nuevas oportunidades de mejora y evolución, ya sea en términos de precisión, escalabilidad o aplicación práctica.

8.1. Conclusiones

A lo largo del desarrollo del presente Trabajo Fin de Grado, se han abordado de forma sistemática todos los objetivos generales propuestos inicialmente. En primer lugar, se ha conseguido una introducción sólida al campo del reconocimiento emocional en texto, combinando tanto el estudio teórico de la literatura como la experimentación práctica con datos reales, lo que ha permitido adquirir una visión integral del problema desde la perspectiva de la inteligencia artificial.

En cuanto a la selección del set de datos, se han comparado distintas fuentes y conjuntos de datos, priorizando aquellos que presentaban lenguaje coloquial y etiquetas emocionales claras. Esta fase ha sido esencial para garantizar que los experimentos se realizaran sobre datos representativos y adecuados al problema tratado.

Se han explorado diversas aproximaciones basadas en Machine Learning y Deep Learning, implementando algoritmos clásicos como Naive Bayes, Regresión Logística, Random Forest y SVM, así como un modelo basado en BERT, representativo de las técnicas más avanzadas en procesamiento del lenguaje natural. Cada enfoque ha sido analizado en cuanto a sus fortalezas, limitaciones y requisitos computacionales, tal como se preveía en los objetivos.

Durante la fase de desarrollo, se ha trabajado intensamente con el ecosistema Python, utilizando herramientas como Pandas, Scikit-learn y PyTorch, lo que ha permitido aplicar los modelos de forma flexible y reproducible. En consecuencia, se han implementado y entrenado múltiples modelos, adaptando su arquitectura y parámetros a las características del corpus disponible.

La evaluación cuantitativa del rendimiento se ha llevado a cabo utilizando métricas relevantes como *F1-score*, precisión, *recall*, AUC y *log-loss*, lo que ha facilitado una comparación justa y rigurosa entre los distintos enfoques implementados. Los resultados han permitido extraer conclusiones significativas sobre la viabilidad de aplicar modelos de IA al análisis emocional textual en las condiciones establecidas.

Finalmente, el trabajo ha culminado en una reflexión crítica centrada en el análisis comparativo de los distintos modelos implementados. En el set de datos principal, todos han mostrado un rendimiento notable, con métricas elevadas y comportamiento estable. Sin embargo, al aplicar los modelos a un conjunto de datos externo, se han observado numerosos errores comunes, lo que pone de manifiesto las limitaciones actuales en cuanto a la capacidad de generalización. Por ello, no es posible concluir que uno de los modelos sea claramente superior a los demás, ya que cada uno presenta fortalezas y debilidades que dependen del dominio y la representación de los datos. Para un análisis más detallado de estas limitaciones, se remite al lector al Capítulo 7, apartado 7.7.

En resumen, todos los objetivos generales definidos al inicio del proyecto han sido cumplidos satisfactoriamente, con la excepción del objetivo relativo a la elección de un modelo óptimo. Este resultado refleja con claridad la complejidad del problema abordado y refuerza la necesidad de seguir investigando para desarrollar soluciones más precisas y generalizables en el reconocimiento emocional en texto.

8.2. Trabajo futuro

A partir del trabajo realizado, se abren múltiples caminos para continuar esta investigación, tanto desde una perspectiva técnica como aplicada y ética. El análisis automático de emociones en texto es un campo en expansión, con un alto potencial de impacto en sectores tan diversos como la salud mental, la educación, el marketing o la interacción persona-máquina. Sin embargo, para que estas soluciones sean realmente eficaces, inclusivas y seguras, es necesario avanzar en varias direcciones clave.

Una de las primeras líneas de desarrollo natural es la ampliación y diversificación del conjunto de datos utilizado. Aunque los resultados obtenidos son robustos, el rendimiento real de un modelo emocional se pone a prueba cuando se enfrenta a contextos variados, estilos lingüísticos diferentes o expresiones emocionales menos directas. Incorporar textos procedentes de redes sociales, foros, literatura, mensajes breves o incluso múltiples idiomas permitiría entrenar modelos más resilientes y menos sesgados. Esta riqueza lingüística y cultural es indispensable para construir herramientas que funcionen en entornos reales, más allá de los laboratorios o sets de datos estandarizados.

En paralelo, resulta imprescindible profundizar en el uso de técnicas avanzadas de aprendizaje profundo. En esta investigación se ha explorado inicialmente BERT como representante de los modelos pre entrenados, pero existen múltiples alternativas que pueden aportar ventajas específicas: RoBERTa, DeBERTa, XLNet o ELECTRA, entre otros. Cada uno presenta diferencias en su arquitectura o preentrenamiento que podrían traducirse en mejoras de rendimiento, eficiencia o capacidad de generalización. Asimismo, el diseño de

experimentos más complejos, como tareas multitarea, clasificación jerárquica o modelos adaptativos, podría llevar la calidad del análisis emocional a otro nivel. El objetivo es no limitarse a aplicar modelos existentes, sino profundizar en su adaptación y evolución para ajustarse mejor a la naturaleza emocional del lenguaje humano.

Una línea de desarrollo especialmente prometedora es la expansión multimodal del análisis emocional. Las emociones no se comunican únicamente a través de palabras, sino también mediante tono de voz, expresiones faciales, gestos o imágenes. Integrar estos canales permitiría construir modelos mucho más completos y cercanos a la percepción humana. Esta convergencia es especialmente útil en campos como la educación emocional, la atención psicológica automatizada o los asistentes conversacionales, donde captar matices afectivos es fundamental para ofrecer respuestas adecuadas y empáticas.

Además de aplicar y ajustar modelos existentes, una propuesta ambiciosa es desarrollar un modelo desde cero, entrenado específicamente para la tarea de análisis emocional. Esto permitiría optimizar cada etapa del proceso en función de las necesidades concretas del problema. Este enfoque puede suponer una mayor complejidad técnica, pero también un avance cualitativo, especialmente si se busca diseñar un sistema que no solo reconozca emociones, sino que las interprete con sensibilidad contextual y matices.

En esta misma línea de sofisticación técnica, es fundamental avanzar hacia una mayor aplicabilidad, calibración y transparencia de los modelos. A medida que estas herramientas se acercan a contextos sensibles, como la salud o la educación, resulta crucial que sus decisiones puedan ser comprendidas y auditadas. Técnicas como SHAP [58] o LIME [59] permiten identificar qué palabras o expresiones han motivado una determinada predicción, lo cual no solo facilita la validación por expertos humanos, sino que también es una vía poderosa para detectar sesgos o errores sistémicos. Además, la calibración probabilística permitirá que los modelos devuelvan estimaciones de confianza más realistas, algo clave para su uso responsable en entornos críticos.

También se considera esencial avanzar hacia una implementación práctica, donde los modelos se incorporen a sistemas interactivos con utilidad directa. Entre los posibles desarrollos se encuentran *plugins* para navegadores que analicen el tono emocional de textos en tiempo real, módulos para plataformas de e-learning que detecten frustración o desmotivación, herramientas de moderación de contenido sensible, o asistentes virtuales capaces de responder empáticamente. Este tipo de aplicaciones puede tener un impacto transformador tanto en la experiencia del usuario como en la comprensión colectiva del lenguaje emocional en entornos digitales.

Una extensión fascinante, y aún poco explorada, es el desarrollo de modelos capaces no solo de detectar emociones, sino también de generarlas. La generación emocional de tiene aplicaciones en narrativa automatizada, marketing afectivo, *chatbots* creativos o entrenamiento de habilidades sociales. Esta línea supone un reto desde el punto de vista del

control semántico, pero también abre un camino completamente nuevo para la interacción emocional hombre-máquina.

En este contexto, *Affective HCI* [60] (Interacción Afectiva Hombre-Máquina) se configura como una aplicación práctica clave dentro del campo más amplio de la Computación Afectiva [61]. Mientras que la computación afectiva estudia cómo dotar a los sistemas de la capacidad para procesar emociones humanas, *Affective HCI* se centra específicamente en cómo estos procesos pueden integrarse en las interfaces que median la interacción con los usuarios. Así, representa el puente entre la teoría emocional computacional y su implementación directa en tecnologías interactivas, permitiendo interfaces más empáticas, adaptativas y centradas en el usuario. Esta sinergia es esencial para diseñar experiencias más naturales e intuitivas en áreas como la educación, la salud, el entretenimiento o la atención al cliente.

Por último, cualquier evolución en este campo debe ir acompañada de una profunda reflexión ética, social y cultural. Las emociones son una parte íntima del ser humano, y su análisis automatizado plantea cuestiones delicadas sobre privacidad, manipulación, sesgo y consentimiento. Será necesario estudiar con rigor los posibles efectos secundarios del uso de estas tecnologías, especialmente en poblaciones vulnerables. Asimismo, será clave incorporar marcos normativos y de gobernanza que aseguren un desarrollo responsable, justo y respetuoso con la diversidad humana.

En definitiva, este trabajo sienta las bases de un campo que, lejos de estar cerrado, apenas comienza a mostrar su potencial. Las líneas de desarrollo propuestas no solo buscan avanzar técnicamente, sino también construir herramientas emocionalmente inteligentes que contribuyan al bienestar humano desde la tecnología, la empatía y la ética.

Bibliografía

[1] Microsoft Teams. Plataforma de comunicación y colaboración en línea: https://www.microsoft.com/es-es/microsoft-teams/group-chat-software

[2] Google Calendar. Herramienta de planificación y gestión de tiempo: https://calendar.google.com

[3] Repositorio del TFG: Detección de emociones en texto [GitLab]. Universidad de Valladolid: https://gitlab.inf.uva.es/izajime/tfg_deteccion_de_emociones#

[4] Gestión de riesgos. Área de conocimiento del PMBOK: https://www.soypm.website/area-de-conocimiento/gestion-de-riesgos/

[5] Juanes Mayfield, B. (2023–2024). Análisis emocional con integración de IA en proyectos de cambio organizacional en empresas familiares. Máster en Dirección de Proyectos, Universidad de Valladolid.

[6] Juanes Mayfield, B. (2023–2024). Análisis emocional con integración de IA en proyectos de cambio organizacional en empresas familiares. Amazon Web Services. (2024). ¿Qué es el Procesamiento de lenguaje natural (NLP)?: https://aws.amazon.com/es/what-is/nlp/

[7] Juanes Mayfield, B. (2023–2024). Análisis emocional con integración de IA en proyectos de cambio organizacional en empresas familiares. Pak, A., & Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. ACL Anthology: https://aclanthology.org/L10-1263/

[8] Juanes Mayfield, B. (2023–2024). Análisis emocional con integración de IA en proyectos de cambio organizacional en empresas familiares. Elastic. (s.f.). ¿Qué es el análisis de sentimientos? Elastic: https://www.elastic.co/es/what-is/sentiment-analysis

[9] Juanes Mayfield, B. (2023–2024). Análisis emocional con integración de IA en proyectos de cambio organizacional en empresas familiares. Insua Yañez, A., Gómez Rodríguez, C., & González Vázquez, S. (2019). Sistema Deep Learning para el análisis de sentimientos en opiniones de productos para la ordenación de resultados de un buscador semántico. Universidad de A Coruña.

[10] Juanes Mayfield, B. (2023–2024). Análisis emocional con integración de IA en proyectos de cambio organizacional en empresas familiares. GitHub. (2019). Semantica y Vectores de palabras.ipynb. GitHub: https://gist.github.com/harpiechoise/1e0a025ddfad5cc2ead3d5f5077d5791

[11] Juanes Mayfield, B. (2023–2024). Análisis emocional con integración de IA en proyectos de cambio organizacional en empresas familiares. González, F. (2020). Métodos analíticos. ITAM. URL: https://heuristic-bhabha-ae33da.netlify.app/representaci%C3%B3n-de-palabras-y-word2vec.html

[12] Juanes Mayfield, B. (2023–2024). Análisis emocional con integración de IA en proyectos de cambio organizacional en empresas familiares. Akinsanmi Ruiz, K. M.; Sánchez Carreras, D. (2023). “Aplicación de una herramienta de análisis emocional en una plataforma de medios sociales” URL: https://upcommons.upc.edu/bitstream/handle/2117/396309/405_Memoria_TFG.pdf?sequence=2&isAllowed=y

[13] Dwivedi, A., Yadav, A., & Mandal, S. (2023). Role of artificial intelligence in inachieving the Sustainable Development Goals (SDGs). Journal of Business Research, 162, 113974: https://www.sciencedirect.com/science/article/pii/S0148296323001091

[14] Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., & Specia, L. (2019). SemEval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. Proceedings of the 13th International Workshop on Semantic Evaluation: https://aclanthology.org/S19-2007/

[15] Loaiza-Ganem, G., Park, S., & Blei, D. M. (2022). Estimating the Number of Latent Causes. Patterns, 3(2), 100452: https://www.sciencedirect.com/science/article/pii/S2590005622000224

[16] ScienceDirect. (s.f.). Bidirectional Encoder Representations from Transformers: https://www.sciencedirect.com/topics/computer-science/bidirectional-encoder-representations-from-transformers

[17] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv: https://arxiv.org/abs/1810.04805

[18] Reimers, N. (s.f.). Unsupervised Learning with MLM. Sentence-Transformers: https://sbert.net/examples/sentence_transformer/unsupervised_learning/MLM/README.html

[19] Scaler. (s.f.). BERT Next Sentence Prediction: https://www.scaler.com/topics/nlp/bert-next-sentence-prediction/

[20] Davenport, T. H., & Kalakota, R. (2020). The potential for artificial intelligence in healthcare. Future Healthcare Journal, 7(2), 94–98: https://www.sciencedirect.com/science/article/pii/S1532046420302069

[21] Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017). Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: https://aclanthology.org/P17-1067/

[22] Psicología y Mente. (s.f.). La rueda de las emociones de Robert Plutchik: https://psicologiaymente.com/psicologia/rueda-emociones-robert-plutchik

[23] GanttProject. Herramienta de planificación de proyectos: https://www.ganttproject.biz/

[24] Comisión Europea. (2019). Directrices éticas para una inteligencia artificial fiable. Oficina de Publicaciones de la Unión Europea: https://op.europa.eu/es/publication-detail/-/publication/d3988569-0434-11ea-8c1f-01aa75ed71a1

[25] Secretaría de Estado de Digitalización e Inteligencia Artificial. (s.f.). Reglamento de Inteligencia Artificial - Sandbox regulatorio. Gobierno de España: https://avance.digital.gob.es/sandbox-IA/Paginas/reglamento-IA.aspx

[26] van Rossum, G., Warsaw, B., & Coghlan, N. (2001). PEP 8 – Style Guide for Python Code. Python Enhancement Proposals. Python Software Foundation: https://peps.python.org/pep-0008/

[27] Astah Professional. (n.d.). Model-based design tool for software modeling: https://astah.net/products/astah-professional/

[28] Paul Ekman Group. Universal Emotions: https://www.paulekman.com/universal-emotions/

[29] Kaggle. Datasets - Discover and publish data projects: https://www.kaggle.com/datasets

[30] Nelgiriyewithana. Emotions. Kaggle: https://www.kaggle.com/datasets/nelgiriyewithana/emotions

[31] Parul Pandey. Emotion Dataset for Emotion Recognition Tasks. Kaggle: https://www.kaggle.com/datasets/parulpandey/emotion-dataset

[32] Debarshi Chanda. GoEmotions Dataset. Kaggle: https://www.kaggle.com/datasets/debarshichanda/goemotions

[33] Jackksoncsie. EmoBank Dataset. Kaggle: https://www.kaggle.com/datasets/jackksoncsie/emobank

[34] Python Official Website - Python Software Foundation: https://www.python.org/

[35] MIT Sloan School of Management. (n.d.). Machine learning explained: https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained

[36] ScienceDirect. (n.d.). Deep learning: https://www.sciencedirect.com/topics/engineering/deep-learning

[37] Deep learning vs Machine learning - Google Cloud: https://cloud.google.com/discover/deep-learning-vs-machine-learning

[38] Scikit-learn Documentation: https://scikit-learn.org/stable/

[39] Pandas Documentation: https://pandas.pydata.org/docs/

[40] Matplotlib Documentation: https://matplotlib.org/

[41] PyTorch Documentation: https://pytorch.org/docs/stable/index.html

[42] Inverse Document Frequency. In: ScienceDirect Topics – Mathematics. Elsevier: https://www.sciencedirect.com/topics/mathematics/inverse-document-frequency

[43] Scikit-learn: TF-IDF Vectorizer: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

[44] Scikit-learn - Classification Metrics: https://scikit learn.org/stable/modules/model_evaluation.html

[45] Scikit-learn - Learning Curves: https://scikit-learn.org/stable/visualizations.html#learning-curves

[46] Elsevier. (s.f.). Logistic regression – an overview. ScienceDirect Topics: https://www.sciencedirect.com/topics/computer-science/logistic-regression

[47] Scikit-learn - LogisticRegression: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[48] Elsevier. (s.f.). Naive Bayes classifier – an overview. ScienceDirect Topics: https://www.sciencedirect.com/topics/engineering/naive-bayes-classifier

[49] Scikit-learn - MultinomialNB: https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html

[50] Elsevier. (s.f.). Random forest – an overview. ScienceDirect Topics: https://www.sciencedirect.com/topics/nursing-and-health-professions/random-forest

[51] Scikit-learn - RandomForestClassifier: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[52] Elsevier. (s.f.). Support vector machine – an overview. ScienceDirect Topics: https://www.sciencedirect.com/topics/computer-science/support-vector-machine

[53] Scikit-learn - SVC (Support Vector Classifier): https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[54] Hugging Face Transformers: https://huggingface.co/transformers/

[55] Intel Core i7-14700KF: https://www.pccomponentes.com/intel-core-i7-14700kf-34-56ghz-box

[56] MSI PRO B760-P DDR4 II: https://es.msi.com/Motherboard/PRO-B760-P-DDR4-II

[57] MSI GeForce RTX 4070 Ti SUPER Ventus 2X OC 16GB GDDR6X: https://www.pccomponentes.com/msi-geforce-rtx-4070-ti-super-ventus-2x-oc-16gb-gddr6x-dlss3

[58] Introducción a los valores SHAP: https://www.datacamp.com/es/tutorial/introduction-to-shap-values-machine-learning-interpretability

[59] Explicación modelo LIME: https://medium.com/latinxinai/explicabilidad-de-modelos-de-ml-lime-f9d0dceb5154

[60] Picard, R. W., Vyzas, E., & Healey, J. (2001). Toward machine emotional intelligence: Analysis of affective physiological state: https://www.sciencedirect.com/science/article/abs/pii/S1071581903000478

[61] Ding, Y., Xia, R., Zhang, C., & Wu, Y. (2024). Human-Centered Affective Computing: Recent Advances and Future Directions: https://spj.science.org/doi/10.34133/icomputing.0076

[62] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0: Step-by-step data mining guide. CRISP-DM Consortium.

[63] Haya, P. (2021, 29 de noviembre). La metodología CRISP-DM en ciencia de datos. Instituto de Ingeniería del Conocimiento (IIC), UAM: https://www.iic.uam.es/innovacion/metodologia-crisp-dm-ciencia-de-datos/

[64] Schwaber, K., & Sutherland, J. (2020). The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. Scrum.org: https://scrumguides.org

[65] - Riba Campos, C. (s.f.). La entrevista como técnica nuclear de la observación participante: https://openaccess.uoc.edu/server/api/core/bitstreams/20f83cda-f0af-461b-966e-622542b823fb/content

[66] Dataiku. TF-IDF Vectorization – Text Features for Machine Learning. Disponible en: https://knowledge.dataiku.com/latest/ml-analytics/nlp/concept-text-features-for-ml.html

[67] KeepCoding. Qué es el algoritmo TF-IDF Vectorizer. Disponible en: https://keepcoding.io/blog/que-es-el-algoritmo-tf-idf-vectorizer/

[68] Glassdoor. Ofertas de empleo y sueldos en España: https://www.glassdoor.es/Job/index.htm

[69] Glassdoor. Sueldo: Data Scientist en España: https://www.glassdoor.es/Sueldos/data-scientist-sueldo-SRCH_KO0,14.htm

[70] Glassdoor. Sueldo: Ingeniero de Inteligencia Artificial y Aprendizaje Automático en España: https://www.glassdoor.es/Sueldos/ingeniero-de-inteligencia-artificial-y-aprendizaje-automatico-sueldo-SRCH_KO0,61.htm

[71] Glassdoor. Sueldo: Analista de Datos en España: https://www.glassdoor.es/Sueldos/analista-de-datos-sueldo-SRCH_KO0,17.htm