

UNIVERSIDAD DE VALLADOLID



E.T.S.I. TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS  
DE TELECOMUNICACIÓN

**Modelización de algoritmos de Deep Learning en  
evaluar la respuesta emocional en redes sociales**

Autor:

**Dña. Ainhoa Santamaría Puras**

Tutor:

**Dña. Noemí Merayo Álvarez**

---

**TÍTULO: Modelización de algoritmos de Deep Learning en evaluar la respuesta emocional en redes sociales**

**AUTOR: Dña. Ainhoa Santamaría Puras**

**TUTOR: Dña. Noemí Merayo Álvarez**

**DEPARTAMENTO: Teoría de la Señal y Comunicaciones e Ingeniería Telemática**

---

---

**TRIBUNAL**

**PRESIDENTE:**

**SECRETARIO:**

**VOCAL:**

**SUPLENTE:**

**SUPLENTE:**

---

---

**FECHA:**

**CALIFICACIÓN:**

---

## **Resumen de TFG**

El trabajo realizado aborda la problemática de analizar las respuestas emocionales en las redes sociales mediante técnicas de aprendizaje automático. En los últimos años, las redes sociales se han convertido en una herramienta cada vez más utilizada en la sociedad, teniendo un impacto significativo en ella. Esto ha llevado a que los usuarios comenten más en los posts, generando una necesidad de comprender las emociones que estos posts despiertan en los usuarios.

El objetivo de este Trabajo de Fin de Grado es llevar a cabo un estudio para analizar el impacto emocional de comentarios de Instagram y TikTok ante revelaciones de problemas de salud mental realizado por famosos. Se ha elegido estas dos redes sociales por ser dos de las más utilizadas entre los jóvenes.

En primer lugar, se ha ampliado un corpus con el que se va a entrenar el modelo, etiquetando cada comentario con sus correspondientes polaridades y emociones. Posteriormente, los mensajes extraídos del corpus se procesan para eliminar información redundante y normalizar los datos. Luego, estos mensajes se pasan por un modelo de clasificación basado en Deep Learning, y se evalúa el rendimiento del modelo utilizando diferentes métricas de precisión. Este modelo se integra en una interfaz donde se puede cargar un archivo o introducir un texto por teclado, mostrando visualmente los resultados. Además, estos resultados pueden descargarse en un archivo para su futura interpretación.

## **Palabras clave**

Deep Learning, redes neuronales, redes sociales, post, corpus, respuesta emocional, interfaz.

## **Abstract**

The work carried out addresses the issue of analyzing emotional responses on social media using machine learning techniques. In recent years, social media has become an increasingly utilized tool in society, having a significant impact on it. This has led to users commenting more on posts, creating a need to understand the emotions these posts evoke in users.

The objective of this Final Degree Project is to carry out a study to analyze the emotional impact of Instagram and TikTok comments in the face of revelations of mental health problems made by celebrities. These two social networks have been chosen because they are two of the most used among young people.

First, the corpus used to train the model is analyzed, with each comment being tagged with its corresponding polarities and sentiments. Subsequently, the extracted messages are processed to remove redundant information and normalize the data. Then, these messages are passed through a Deep Learning-based classification model, and the model's performance is evaluated using different precision metrics. This model is integrated into an interface where a file can be uploaded or text can be entered via keyboard, visually displaying the results. Additionally, these results can be downloaded in a file for future interpretation.

## **Keywords**

Deep Learning, neural networks, social media, post, corpus, emotional response, interface.

# Agradecimientos

*Quiero dedicar este trabajo a mi Madre. Gracias por enseñarme tus valores, darme tu amor, ser mi ejemplo de lucha y fortaleza. Aunque ya no estes físicamente a mi lado, sigues siendo mi faro, mi inspiración. Este logro es también tuyo, mamá.*

*También quiero expresar mi agradecimiento a Noemí, quien desde el inicio ha sido fundamental en el desarrollo y éxito de este proyecto.*

# Índice

<b>Agradecimientos .....</b>	<b>v</b>
<b>Índice .....</b>	<b>vi</b>
<b>Índice de figuras .....</b>	<b>1</b>
<b>Índice de tablas .....</b>	<b>2</b>
<b>Índice de ecuaciones .....</b>	<b>2</b>
<b>1 Introducción.....</b>	<b>3</b>
1.1 Motivación.....	3
1.2 Objetivos.....	3
1.2.1 Objetivo Principal.....	3
1.2.2 Objetivos Específicos .....	4
1.3 Metodología del trabajo.....	4
1.3.1 Fase de Documentación.....	4
1.3.2 Fase de análisis .....	5
1.3.3 Fase de Pruebas.....	5
1.3.4 Fase de escritura del informe .....	5
1.4 Estructura de la Memoria.....	5
<b>2 Estado del arte .....</b>	<b>7</b>
2.1 Inteligencia Artificial y Deep Learning.....	7
2.1.1. Descripción de Deep Learning .....	7
2.1.2. Diseño del algoritmo hibrido basado en Deep Learning .....	9
2.2 Impacto de las redes sociales Instagram y TikTok.....	10
<b>3 Herramientas utilizadas.....</b>	<b>13</b>
3.1 Introducción.....	13

---

3.2	Python .....	13
3.3	Tensorflow y Keras.....	14
3.4	NLTK.....	14
3.5	TikTok .....	14
3.6	Instagram .....	15
3.7	Anaconda .....	15
<b>4</b>	<b>Ampliación del Corpus de Salud Mental.....</b>	<b>16</b>
4.1	Introducción .....	16
4.2	Descripción del corpus de salud mental .....	16
4.2.1	Metodología de selección y descarga de posts .....	19
4.2.2	Descripción de los posts descargados y proceso de etiquetado .....	20
<b>5</b>	<b>Análisis de la respuesta emocional en redes sociales: Polaridad y Emociones.....</b>	<b>27</b>
5.1	Introducción .....	27
5.2	Análisis de polaridad en salud mental .....	27
5.2.1	Definición del corpus de datos etiquetado .....	27
5.2.2	Análisis de resultados de polaridad: Positivo, Negativo, Neutro .....	28
5.3	Análisis de emociones en salud mental .....	36
5.3.1	Definición del corpus de datos etiquetado .....	36
5.3.2	Resultados de emociones para el dataset de salud mental .....	38
5.4	Conclusiones .....	46
<b>6</b>	<b>Interfaz gráfica para la implementación del modelo .....</b>	<b>48</b>
6.1	Introducción .....	48
6.2	Desarrollo de la Interfaz .....	48
6.2.1	Descarga de las herramientas necesarias para el desarrollo .....	48
6.2.2	Creación de la ventana y elementos gráficos.....	50
6.2.3	Creación de clases.....	53

---

6.3	Diseño y ejecución de la Interfaz gráfica.....	56
6.4	Conclusiones.....	59
<b>7</b>	<b>Conclusiones y líneas futuras .....</b>	<b>60</b>
7.1	Conclusiones.....	60
7.2	Líneas futuras.....	61
<b>8</b>	<b>Bibliografía.....</b>	<b>62</b>



# Índice de figuras

Figura 1: Explicación categorías [8].....	7
Figura 2: Ejemplo de red neuronal [8].....	8
Figura 3: Ejemplo real de etiquetado manual .....	26
Figura 4: distribución del corpus por polaridad.....	28
Figura 5: modelo tras optimización de 3 polaridades .....	32
Figura 6: matriz de confusión para configuración optima 3 polaridades .....	34
Figura 7: matriz de confusión para configuración optima 3 polaridades antiguo corpus .	35
Figura 8: resultados para configuración óptima 3 polaridades .....	35
Figura 9: distribución del corpus por emociones.....	36
Figura 10: distribución del corpus por emoción gratitud.....	37
Figura 11: distribución del corpus por emoción amor/admiración.....	38
Figura 12: modelo tras optimización de 6 emociones .....	42
Figura 13: matriz de confusión para configuración óptima nuevo corpus .....	45
Figura 14: matriz de confusión para configuración óptima antiguo corpus .....	46
Figura 15: Entorno gráfico de Anaconda.....	49
Figura 16: Importación de bibliotecas Python.....	50
Figura 17: Error en las librerías .....	50
Figura 18: Creación de la ventana .....	51
Figura 19: Distribución de comentarios por emociones en % .....	52
Figura 20: Distribución de todos los comentarios en emociones .....	53
Figura 21: Interfaz gráfica en MacOS .....	56
Figura 22: Resultado de ejecutar algoritmo introduciendo un comentario.....	57
Figura 23: Ventana emergente para cargar archivo .....	58
Figura 24: Resultado obtenido de ejecutar algoritmo con csv/Excel.....	58
Figura 25: Ventana emergente para guardar resultados en excel/csv .....	59

## Índice de tablas

Tabla 1. Post y número de comentarios descargados .....	25
Tabla 2: resultados obtenidos para variación de neuronas y filtros LSTM .....	29
Tabla 3: resultados obtenidos para variación de tasas de dropout.....	29
Tabla 4: resultados obtenidos para variación de tasa de aprendizaje .....	30
Tabla 5: resultados obtenidos para variación de tamaño del vocabulario .....	31
Tabla 6: resultados obtenidos para variación de batch_size .....	31
Tabla 7: parámetros optimizados .....	32
Tabla 8: valores de métricas para configuración óptima 3 polaridades.....	33
Tabla 9: resultados obtenidos para variación de neuronas y filtros LSTM sin capas ocultas .....	39
Tabla 10: resultados obtenidos para variación de tasas de dropout.....	40
Tabla 11: resultados obtenidos para variación de tasa de aprendizaje .....	40
Tabla 12: resultados obtenidos para variación de tamaño del vocabulario .....	41
Tabla 13: resultados obtenidos para variación del batch_size .....	42
Tabla 14: parámetros optimizados .....	43
Tabla 15: valores de métricas para configuración optima 6 emociones.....	43

## Índice de ecuaciones

Ecuación 1: Expresión matemática de una Neurona .....	9
---	---

# 1

## Introducción

### 1.1 Motivación

La razón principal que motivó la realización de este trabajo es el potencial actual de la inteligencia artificial, más concretamente el Deep Learning y sus múltiples aplicaciones.

También hay que tener en cuenta el auge de las redes sociales en la actualidad, las cuales son utilizadas tanto por jóvenes como por adultos y se han convertido en un medio de comunicación muy potente, lo que las convierte en una herramienta perfecta para analizar cómo afectan a la salud mental y a las emociones que despiertan.

Otro factor que se tuvo en cuenta para la realización de este proyecto fue las posibilidades que ofrecía la inteligencia artificial a los creadores de contenido, ayudándolos a obtener conclusiones de los datos que obtienen al subir sus posts, en este caso, en el ámbito de la respuesta emocional de sus seguidores.

Por último, se eligieron las dos redes sociales más utilizadas actualmente para realizar el estudio, las cuales son Instagram y TikTok.

### 1.2 Objetivos

#### *1.2.1 Objetivo Principal*

El propósito central de este Trabajo de Fin de Grado es aplicar técnicas de Deep Learning para analizar la respuesta emocional, específicamente el nivel de polaridad y las emociones expresadas en mensajes de redes sociales, particularmente en Instagram y TikTok.

Como punto de partida se utiliza un algoritmo previamente desarrollado [1] en un TFG anterior, basado en técnicas de Deep Learning. Este algoritmo se aplicará a distintos conjuntos de datos provenientes de diversas redes sociales y con temáticas variadas relacionadas con problemas de salud mental, con el fin de analizar su comportamiento en diferentes contextos.

El objetivo final de este trabajo es desarrollar un modelo de Deep Learning versátil y flexible que asegure la máxima precisión en la clasificación de mensajes según su polaridad o las emociones expresadas. Además, se creará una interfaz que permita a cualquier usuario utilizar este modelo fácilmente.

### **1.2.2 Objetivos Específicos**

Durante este Trabajo de Fin de Grado se han desarrollado los objetivos específicos que se enumeran a continuación:

1. Análisis del estado del arte en técnicas de Deep Learning y redes neuronales.
2. Aplicación y optimización de un modelo de Deep Learning sobre un corpus de Instagram y TikTok, con el objetivo de realizar un análisis que permita una clasificación precisa de emociones.
3. Realización de una interfaz gráfica para poder aplicar el modelo previamente optimizado de manera más sencilla y visual y obtener resultados al instante.

## **1.3 Metodología del trabajo**

La metodología empleada para alcanzar los objetivos se ha estructurado en las fases que se detallarán a continuación.

### **1.3.1 Fase de Documentación**

Esta fase ha sido de gran relevancia por diversos motivos. Primero, ha permitido establecer una base sólida para el problema, facilitando la comprensión de la importancia del análisis de emociones en redes sociales. Además, ha sido esencial para seleccionar las técnicas o enfoques más adecuados para realizar trabajos de aprendizaje automático.

### **1.3.2 Fase de análisis**

Esta fase es una de las más importantes, puesto que ayuda a comprender y extraer la información relevante. En esta fase se analizan los algoritmos previamente creados con su correspondiente documentación. Estos modelos se entrenan y evalúan utilizando técnicas de validación cruzada.

Para la realización de esta fase se ha partido de los siguientes trabajos:

- TFG de Jesús Herrero Llanos [1].
- TFM de José Carlos Sobrino [2].
- TFM de Julia Isabel Medrano Sanz [3].
- TFM de Iván Arévalo Núñez [4].
- TFG de Miguel Ángel Arias Navarro [5]

### **1.3.3 Fase de Pruebas**

Fase dedicada a realizar pruebas con el objetivo de evaluar el rendimiento y la efectividad del modelo desarrollado. Durante esta etapa, también se ajustaron los hiperparámetros para optimizar el rendimiento del modelo en diversos conjuntos de datos (corpus) de diferentes temáticas y redes sociales.

### **1.3.4 Fase de escritura del informe**

Fase en la cual se ha redactado este Trabajo de Fin de Grado, y donde se muestran y explican los resultados y el proceso llevado a cabo.

## **1.4 Estructura de la Memoria**

La memoria del proyecto se compone de varios capítulos, cada uno enfocado en distintos objetivos, los cuales se detallarán a continuación.

Capítulo 1: Introducción del problema, objetivos a alcanzar y metodología para la resolución de este.

Capítulo 2: Estado del arte sobre inteligencia artificial para contextualizar al lector y facilitar la comprensión del documento.

Capítulo 3: Exposición de las herramientas software utilizadas en el estudio y análisis.

Capítulo 4: Ampliación del corpus con comentarios obtenidos de publicaciones que tratan sobre temas de la salud mental en Instagram y TikTok en busca de mejorar la precisión del modelo.

Capítulo 5: Estudio de clasificación de polaridades y emociones en los corpus ampliados de Instagram y TikTok, con análisis detallado de los resultados obtenidos.

Capítulo 6: Realización de la interfaz gráfica para poder implementar el modelo y, por tanto, ser utilizado por cualquier usuario.

Capítulo 7: Conclusiones del estudio y propuestas de líneas futuras de investigación.

# 2

## Estado del arte

### 2.1 Inteligencia Artificial y Deep Learning

#### 2.1.1. Descripción de Deep Learning

La inteligencia artificial [6] es un campo de la informática que se enfoca en crear y desarrollar sistemas y máquinas que emulen la inteligencia humana. Para lograr esto, se utilizan algoritmos y técnicas de aprendizaje automático, permitiendo llevar a cabo diversas tareas como el procesamiento de grandes volúmenes de datos, el reconocimiento de patrones y la toma de decisiones. El objetivo principal es desarrollar sistemas que ejecuten las tareas asignadas de manera eficiente y precisa, incluso en situaciones complejas.

Una parte de la inteligencia artificial es lo que se denomina Deep Learning. Este agrupa una parte de las técnicas de aprendizaje automático, comúnmente conocido como Machine Learning, que se basa en modelos de redes neuronales y tienen un gran número de aplicaciones. El Deep Learning, gracias a que probablemente sea la más dinámica dentro de la inteligencia artificial, está haciendo que este en pleno auge. En la Figura 1 se muestra los distintos niveles y categorías dentro de la inteligencia artificial.

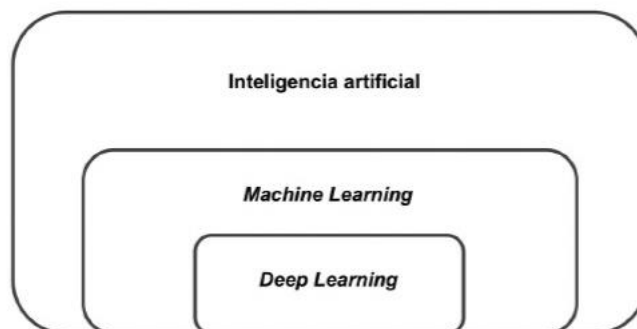


Figura 1: Explicación categorías [9]

Muchos de las aplicaciones y herramientas que utilizan la inteligencia artificial para tareas analíticas y físicas sin la ayuda del ser humano, son impulsados por Deep Learning. Esta tecnología se encuentra en muchos de los aparatos cotidianos del día a día, como por ejemplo los controles de televisión por voz.

En cuanto a su algoritmo, se diferencia de Machine Learning en que elimina la mayor parte del procesamiento de datos previo. Las técnicas de Deep Learning han mejorado la capacidad de clasificar, reconocer, detectar y describir [7].

En cuanto al funcionamiento de esta tecnología, consta de varias capas de nodos conectados entre sí, formando las llamadas redes neuronales (Figura 2), que son capaces de simular las acciones del cerebro de un ser humano. Estas redes pueden estar formadas por cientos de capas neuronales. Esta herramienta destaca por el autoaprendizaje y entrenamiento al que se le somete. Durante este entrenamiento, el algoritmo ajusta las neuronas para reducir la diferencia entre los resultados obtenidos y los que se esperan obtener [8]. Dentro de una red neuronal, se pueden distinguir hasta tres capas o niveles [9]:

- De entrada: Capa que recibe la información de forma directa proveniente de las fuentes externas de la red.
- Ocultas: No tienen contacto directo con el exterior de la red. El número de niveles puede ser de 1 a un número bastante elevado, y estas pueden estar interconectadas de diferentes formas.
- Salida: Mandan la información obtenida de la red al exterior.

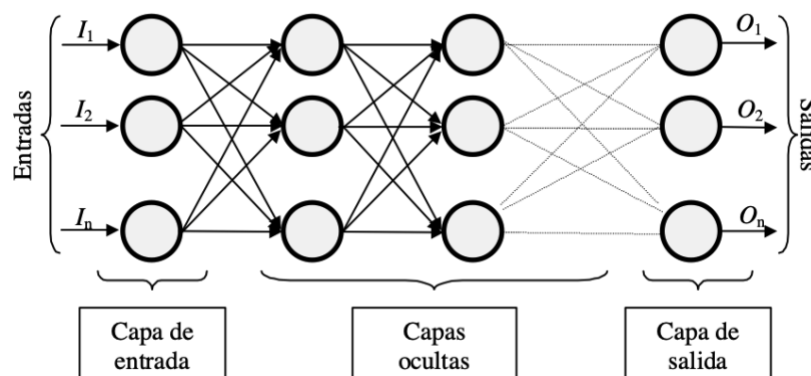


Figura 2: Ejemplo de red neuronal [9]



Estas neuronas, si hablamos desde el punto de vista matemático, la podemos ver como la siguiente función [10]:

$$y = w_1x_1 + \dots + w_nx_n + b$$

Ecuación 1: Expresión matemática de una Neurona

Donde cada variable de la ecuación:

- $y$ : es el valor de salida
- $x_1$  hasta  $x_n$ : son los valores de entrada
- $w_1$  hasta  $w_n$ : son los pesos que determinan la influencia en el resultado final de cada valor de entrada
- $b$ : llamado bias, se trata de un parámetro adicional para que la función no esté condicionada a pasar por el origen.

En cuanto a la aplicación del Deep Learning, este suele funcionar bien con grandes flujos de datos a diferencia de antiguos algoritmos de aprendizaje. Con la aparición del Big Data se necesitan este tipo de algoritmos que tiene mucha potencia para analizar y tratar datos. Actualmente, se utilizan en sectores como la medicina para realizar autodiagnósticos de imagen, en la industria del automóvil con la conducción asistida e incluso en el comercio para enviar mensajes personalizados a cada usuario [8].

## **2.1.2. Diseño del algoritmo híbrido basado en Deep Learning**

En este trabajo se va a usar un modelo híbrido basado en el aprendizaje profundo desarrollado por Miguel Ángel Arias en su Trabajo Fin de Grado [5]. Para mejorar la capacidad de resolver problemas específicos, el modelo combina las ventajas de una Red Neuronal Recurrente (RNN) y una Red Neuronal Convolutiva (CNN).

Las CNN se utilizan para obtener características locales y espaciales de palabras individuales y secuencias de palabras, capturando así los patrones relevantes para el procesamiento de lenguaje natural, que es lo que se busca con los comentarios de Instagram y

TikTok. Las capas convolucionales agregarán filtros de convolución al texto, lo que proporcionará características de nivel superior para el análisis y clasificación de contenidos.

El modelo híbrido utiliza el hecho de que las capas convolucionales pueden extraer características locales y estructurales del texto, lo que hace que los datos de partida sean una representación de características más pequeña. Esta representación se transmite a las capas recurrentes para obtener dependencias a largo plazo e información temporal de la secuencia.

La ventaja de esta combinación es que las capas convolucionales reducen la carga de trabajo de las capas recurrentes porque las características relevantes se han extraído previamente, lo que reduce la dimensión del problema. La estructura en sí mejora la precisión y la eficacia del modelo.

Combinar características locales con dependencias temporales mejorará la capacidad de la red para capturar información, lo que permitirá abordar problemas más complejos.

El modelo con el que trabajaremos se basa en los siguientes componentes básicos: una capa de incrustación, una capa convolucional unidimensional (Conv1D), una capa MaxPooling, una capa LSTM y finalmente una capa densa que clasificará los mensajes. en 2 o 3 capas (2 o 3 polos) dependiendo del problema que estemos investigando en un momento dado

## **2.2 Impacto de las redes sociales Instagram y TikTok**

Las redes sociales [11] son sitios web donde personas físicas o jurídicas interactúan y crean relaciones en base a sus intereses o actividades (ya se trate de aficiones, temas de actualidad o vínculos de índole personal y/o laboral) de forma rápida y sin apenas limitaciones físicas o geográficas.

En los últimos años, especialmente después de la pandemia causada por el COVID 19, el uso de las redes sociales está en un claro auge. Durante ese periodo de tiempo, se popularizaron otro tipo de tendencias en este tipo de aplicaciones que siguen siendo muy populares en la actualidad. A continuación, se mostrarán datos actuales, donde se podrá ver la tendencia de los últimos años en el uso de estas aplicaciones en España.

Según un estudio realizado en 2023 [12], 9 de cada 10 españoles tienen una cuenta en alguna red social y la mayoría la utilizan de manera habitual. Estos datos hacen de España uno de los principales consumidores de redes sociales de Europa y del mundo. Entre sus usuarios, destacan los pertenecientes al rango de entre 18 y 24 años. Estos son los que mayor tiempo consumen en las redes sociales, con una media de 1 hora y 53 minutos. No obstante, el uso también ha aumentado en el resto de los grupos de edad incluido el de edad más adulta. Esto ha sido posible a la gran digitalización de la población española, ya que más de un 78% de la población española tiene acceso a internet.

En cuanto a las redes sociales más utilizadas, esta lista la lidera Whatsapp (86%) y Facebook, seguido de Instagram y TikTok. Estas tres últimas con una utilización superior al 74% [13].

- Facebook: durante años estuvo encabezando el ranking de ser la más utilizada, gracias a ser muy variada: permite subir fotos, videos, chatear, jugar a juegos, etc. Esta red social es una de las favoritas de la generación Millennials y la Generación X. Pero actualmente, debido a la filtración de datos de esta red social y la aparición de otras como Instagram, ha hecho que Facebook este en declive [14].
- Instagram: Es considerada la red social más importante en la actualidad, con 1000 millones de usuarios. La mayor parte de los usuarios pertenecen al rango entre 16 a 23 años. La plataforma integra fotografía y video, con la opción novedosa de Stories fugaces, los cuales se borran de manera automática pasadas 24 horas. Además, las marcas han captado esta tendencia y utilizan esta red social para vender y promocionar sus productos. Cabe mencionar que Instagram goza cada vez de mayor popularidad entre la población española y su número de usuarios no ha cesado de crecer, especialmente durante los dos últimos años [15].
- TikTok: Se podría decir que es la red social del momento. Se trata de una red social para crear y subir videos cortos, de máximo 60 segundos. Lo que la hace diferente al resto de redes sociales es su algoritmo personalizado a cada usuario, sirviendo así videos de temas frecuentados por el usuario. Además, nunca sabes cual va a ser el siguiente video [14].

- Whatsapp: Es la aplicación de mensajería más utilizada a nivel mundial. Permite conectar con otras personas pudiendo mandar mensajería instantánea, videollamadas y llamadas de voz. Esta ha añadido una novedad la cual permite subir todos, las cuales se eliminan a las 24 horas (servicio ofrecido por Instagram también). Entre los usuarios de esta aplicación, en España el 67% de las personas entre 55 y 64 años la utilizan de manera habitual. Esto supone la aplicación más usada en esta franja de edad [16].

Si nos centramos en las dos más utilizadas entre la población joven (Instagram y TikTok), destacan la influencia que tienen en esta población ciertos creadores de contenido, los conocidos como Influencers. Suelen estar en el rango de los 20 a 35 años y ganan dinero con sus publicaciones y sus respectivas visualizaciones. Parte de estas publicaciones son colaboraciones de pago, es decir, publicitan un producto o una idea financiada de manera externa a la red social. Estas personas son seguidas por miles e incluso millones de usuarios, por lo que cada post subido llega a muchísimas personas. Esto conlleva a que la población de este rango de edad los tenga como referentes y como modelos a seguir. Esto a veces puede ser un problema, ya que esa idealización puede provocar un impacto negativo.

# 3

## Herramientas utilizadas

### 3.1 Introducción

Para llevar a cabo el proyecto ha sido necesario utilizar las herramientas y plataformas software que a continuación se describen.

### 3.2 Python

Python [17] es un lenguaje de programación creado por Guido van Rossum a principios de los años 90. Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible. Se trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos. Un lenguaje interpretado o de script es aquel que se ejecuta utilizando un programa intermedio llamado intérprete, en lugar de compilar el código a lenguaje máquina que pueda comprender y ejecutar directamente una computadora (lenguajes compilados).

Las características que han hecho a python ser uno de los lenguajes más utilizados en la actualidad son las siguientes [18]:

- Sintaxis sencilla y legible: La sintaxis es fácil de comprender y aprender comparado con otros lenguajes de programación. Esto hace que cualquiera pueda programar algo sencillo en Python en unas pocas horas, aunque para completo manejo del lenguaje es necesario muchas más horas.
- Lectura simple: el idioma que emplea es el inglés y al carecer de corchetes y comas, hace que cualquier persona sea capaz de entender lo que hace el código con una simple lectura.

- Lenguaje de alto nivel: los programadores no necesitan recordar la memoria ni la arquitectura del sistema, como si sucede en lenguajes de bajo nivel.
- Código abierto: es gratis y se puede modificar y redistribuir libremente.

### 3.3 Tensorflow y Keras

Tanto TensorFlow como Keras son librerías de código abierto empleadas para trabajar con redes neuronales en Python.

Tensorflow [19] es una biblioteca de código abierto para la computación numérica rápida. Fue creado y mantenido por Google y lanzado bajo la licencia de código abierto Apache 2.0. La API es nominalmente para el lenguaje de programación Python, aunque hay acceso a la API C++ subyacente. A diferencia de otras bibliotecas numéricas destinadas a su uso en Deep Learning como Theano, TensorFlow fue diseñado para su uso tanto en investigación y desarrollo como en sistemas de producción, sobre todo RankBrain en la búsqueda de Google y el divertido proyecto DeepDream. Puede ejecutarse en sistemas de CPU individuales, GPU, así como en dispositivos móviles y sistemas distribuidos a gran escala de cientos de máquinas.

Keras [20] es una biblioteca de alto nivel, la cual se ejecuta sobre Tensorflow. Permite la combinación de capas de diferentes tipos, para así, crear redes neuronales más complejas y a su vez ofrece funciones como pueden ser la tokenización o One-Hot, las cuales se usarán en el proyecto. En definitiva, Keras simplifica la creación de redes neuronales.

### 3.4 NLTK

Natural Language Toolkit (NLTK) es una colección de módulos, conjuntos de datos y tutoriales diseñados para facilitar la investigación y la educación en lingüística computacional y procesamiento del lenguaje natural. Esta herramienta está implementada en Python y se distribuye bajo la licencia de código abierto GPL [21].

### 3.5 TikTok

Plataforma de redes sociales y aplicación móvil para crear y publicar videos cortos, de entre 15 y 60 segundos [22]. En los últimos años, y sobre todo a partir de la emergencia

sanitaria causada por el virus COVID-19, su uso ha aumentado de manera exponencial entre los jóvenes entre 16 y 24 años. Su interfaz distinta al resto de redes sociales y su algoritmo personalizado a cada usuario dependiendo del uso de cada uno de ellos hace que TikTok sea una plataforma muy atractiva para el usuario [23].

### **3.6 Instagram**

Instagram es una aplicación de red social utilizada por más de mil millones de personas en todo el mundo para compartir fotos, vídeos y mensajes. Instagram te permite seguir personas y ser seguidos por ellos, pero a diferencia de Facebook, no es necesario que ambos se sigan para que uno siga a otro y viceversa. Los usuarios con cuentas privadas pueden controlar quien los sigue. Esta red social permite realizar videos en directo, publicar post, videos de hasta 60 segundos o historias (fotos o videos que son eliminados cuando transcurren 24 horas) [24].

### **3.7 Anaconda**

Anaconda [25] es un software gratuito que proporciona un conjunto de herramientas diseñadas para la investigación y la ciencia. La instalación de Anaconda brinda acceso a diferentes entornos que permiten codificar en Python o R. Estos entornos, también conocidos como entornos de desarrollo integrado (IDE), son plataformas o aplicaciones que facilitan enormemente el desarrollo de código. Los IDE contienen muchas funciones útiles para escribir, editar y depurar código, visualizar e inspeccionar datos, almacenar variables, presentar resultados y colaborar en proyectos.

La descarga del kit de herramientas también proporciona una enorme selección de funciones prediseñadas que la comunidad Python ha codificado en el pasado. Estas funciones se reagrupan en lo que se llaman bibliotecas y se pueden descargar fácilmente a través de Anaconda. Además, también simplifica el proceso de mantener todas estas bibliotecas actualizadas.

# 4

## Ampliación del Corpus de Salud Mental

### 4.1 Introducción

Este capítulo se centrará en describir el corpus inicial, así como la ampliación del corpus con nuevos comentarios obtenidos de publicaciones en redes sociales. El objetivo será aumentar la precisión y robustez del modelo para poder clasificar los comentarios en función de su polaridad (positiva, negativa o indeterminado) y emoción (Amor/Admiración, Gracitud, Comprensión/empatía/identificación, Tristeza o Ira/desprecio/burla). Para ello se ha utilizado una metodología para la selección de posts y de influencers, la cual se detallará en este capítulo.

### 4.2 Descripción del corpus de salud mental

El corpus inicial corresponde a un corpus de salud mental en redes sociales etiquetado con polaridad y estigma. En particular, el corpus inicial consistía en 2.007 comentarios etiquetados con polaridad (positiva, negativa, neutral) y estigma, provenientes de comentarios en publicaciones de Instagram sobre revelaciones de salud mental de influencers. A este se le han añadido otros 2038 comentarios etiquetados de igual manera que los iniciales provenientes de Instagram y de TikTok en publicaciones de la misma temática que los comentarios del corpus inicial.

1. **Polaridad:** Consiste en asignar un valor positivo, negativo o neutral/indefinido a los comentarios en respuesta a la revelación o descripción de la sintomatología en la publicación. La polaridad positiva refleja comprensión, ánimo o incluso admiración por la publicación. Por ejemplo, “*Animo bonita!! Todo saldrá bien*”. La polaridad negativa se asigna cuando la persona expresa opiniones negativas, generalmente cuestionando la publicación con comentarios irónicos, sarcásticos o



incluso burlones y despectivos. Por ejemplo, "*Pues se nota que no has tenido nunca un ataque de ansiedad real*". La polaridad neutral o indefinida se asigna en casos donde no se detecta una opinión clara o esta puede interpretarse en ambos sentidos. Por ejemplo, "*puedes hablar de lo que quieras*" "*Increíble texto*".

2. Estigma: Las respuestas estigmatizantes a los comentarios son comportamientos en los que se expresan creencias y emociones negativas hacia los problemas de salud mental. El estigma se manifiesta de diversas formas, incluyendo el rechazo y la ira hacia la persona, que pueden extenderse al desprecio o la burla, minimizando su problema. Por ejemplo, "*Buffff!! Cuentista!!!*"; "*Madre mía chica uff, flipo y flipo la verdad*"; "*Esta señora no está bien*". Debido a que socialmente sabemos que "el estigma está mal", muchos comentarios de rechazo se hacen de manera irónica o sarcástica. Por ejemplo, "*Los ricos también lloran???*"; "*Otras chonis el dinero fácil ahora a contar miseria*". Además, se muestra enojo argumentando que tales publicaciones "trivializan o comercializan" la salud mental. Por ejemplo, "*Si trabajaras y no vivieras del cuento no tendrías depresión*". Otras veces, el estigma se manifiesta como lástima o pena por la persona. Por ejemplo, "*Ayyy q lastima*"; "*Me parte el corazón verte así :(*".

Por otro lado, el conjunto final de etiquetas incluye las siguientes emociones: Amor/Admiración, Comprensión/Empatía/Identificación, Gratitud, Tristeza, Ira/Desprecio/Burla. A estas, añadimos la categoría "Neutral" para los casos en los que resulta imposible categorizar el mensaje con una emoción. A continuación, se presenta una descripción más detallada de cada emoción con algunos ejemplos reales que ayudan a categorizar los comentarios:

- Amor/Admiración: Emociones presentes en los modelos de Fredrickson (2013) y Plutchick (2001) donde la admiración, aprobación y amor están estrechamente relacionados. Suelen ser mensajes con contenido positivo con elogios. Por ejemplo, "*eres un campeón*"; afecto, por ejemplo, "*te queremos*". A veces son mensajes fortalecedores que buscan reforzar la confianza en sí mismo de la persona, fomentando una actitud positiva y resiliente, como en "*tú eres lo primero, estamos seguros de que puedes hacerlo*"; "*Ante todo, eres hermoso. Por encima de todo, quienes te amamos siempre te apoyaremos. Eres maravilloso y puedes superar todo esto Buena suerte*". Y

*"Cuando no estás bien, es necesario PARAR y redescubrirte a ti mismo. Estamos seguros de que puedes manejarlo".*

- Gratitud: Presente en modelos recientes (Ekman, 2004; Fredrickson, 2013), los mensajes implican un agradecimiento sincero por compartir contenido relacionado con la salud mental en las redes sociales. Por ejemplo, *"gracias por visibilizar este problema"*, *"Gracias por esta publicación y por lo que compartes"* o *"Aprecio que hayas publicado esto"*.
- Comprensión/empatía/identificación: Presente en Plutchick (2001). Implica interés y comprensión del mensaje, incluyendo la identificación personal con la situación o contexto, poniéndose en el lugar del otro. Por ejemplo, *"A todos nos pasa y todos hemos pasado por momentos como este"*. A menudo provocan revelaciones de problemas de salud mental al afirmar haber pasado por lo mismo. *"Te entiendo tanto... Quiero salir de esta depresión"*; *"Anteayer tuve un ataque de ansiedad brutal. Pensé que me estaba muriendo y tuve que ir a urgencias, creyendo que era un ataque al corazón y temiendo no salir de allí. Solo quienes lo han vivido saben cómo es..."* y *"Te entiendo tanto... Espero que este nuevo año nos traiga más salud y oportunidades. Quiero salir ya de esta depresión"*.
- Tristeza: Esta emoción primaria (Ekman, 2004) se produce por eventos que no son agradables y que denotan pesadez. Incluye muchas manifestaciones de lástima hacia la persona. Por ejemplo, *"pobrecita"* *"qué lástima"*. También hay comentarios que expresan tristeza por la despedida de la influencer después de su anuncio de dejar las redes sociales. Por ejemplo, *"te extrañaremos"*. También incluye expresiones en las que la persona que escribe anuncia que siente pesar al leer el mensaje: *"Se me parte el corazón al escuchar esto; es una lástima que te sientas así"*.
- Ira/desprecio/burla: Esta categoría involucra respuestas de irritación y ataques a la persona como ridícula y superficial (Ekman, 2004). Por ejemplo, *"subir fotos llorando es un nivel más alto de ridiculez"*; *"cada vez que lloras, subes una foto... estoy tan enfadado..."*; *"eres tan asquerosa"*; *"eres tan repugnante"*; *"eres tan ridícula..."*; *"eres tan ridícula"*; *"eres tan ridículo"*. En otras instancias, lo que hacen es sembrar dudas y menospreciar la forma en que han experimentado sus problemas de salud mental, desacreditándolos: *"No sabes lo que es tener depresión. Si tuvieras depresión, no te"*

*sentirías con ánimos de grabarte", "Tengo ansiedad. Cuando tengo ataques de ansiedad, ni siquiera puedo moverme... Está claro que no sabes lo que te está pasando".*

- Neutral: Esta categoría corresponde a mensajes que proporcionan consejos como "*deberías desconectar de las redes sociales y descansar*", así como mensajes que hacen referencia a temas religiosos como "*que Dios esté contigo*". También incluye mensajes en los que se preguntan cosas que no expresan ninguna emoción: "*Hola, ¿qué psicólogos te están tratando, si se puede preguntar, por supuesto...?*", "*¿Alguien sabe si se han ordenado los horarios de la semana pasada?*".

#### **4.2.1 Metodología de selección y descarga de posts**

La metodología de selección y descarga de posts es un componente esencial del análisis y de la respuesta emocional en redes sociales. Se realizó una búsqueda activa de posts en la que los protagonistas trataran de la salud mental desde un punto de vista personal, contando sus experiencias y problemas. Además, para asegurarnos de la influencia de estos posts, únicamente se consideraron validos aquellos que fueran de personas *influencers* españolas con gran cantidad de seguidores en la red social. La metodología utilizada fue la siguiente:

- Identificación de posts. Se buscaron y seleccionaron posts que trataran de los problemas de salud mental desde un punto personal, como procesos de depresión, ansiedad u otros problemas psicológicos. Para ello, se utilizaron palabras clave y hashtags relacionados con la salud mental para una búsqueda más eficaz.
- Influencia del autor. De todos los posts que cumplieron con el criterio explicado en el primer punto, solo se consideraron validos aquellos cuyos autores tuvieran al menos 10 mil seguidores en la plataforma. De esta manera, la elección de influencers garantiza una mayor visibilidad, interacción y una mayor variedad de comentarios.
- Descarga de datos. Se utilizó la herramienta online export comments [26] para poder descargar los comentarios en formato xlsx y que el proceso de selección fuera más sencillo.

Es esencial llevar a cabo esta metodología para así poder obtener el mejor resultado posible.

#### **4.2.2 Descripción de los posts descargados y proceso de etiquetado**

En esta sección, se tratarán los posts descargados y el proceso de etiquetado de estos, los cuales serán utilizados para el entrenamiento y aprendizaje del modelo. Los posts descargados fueron seleccionados mediante la metodología descrita en el anterior capítulo. Estos son protagonizados en su totalidad por mujeres españolas con cierta influencia en las redes sociales, en este caso TikTok e Instagram. Por lo tanto, el objetivo es ampliar el corpus inicial que tan solo estaba centrado en Instagram y nutrirlo con otras redes sociales usadas masivamente por jóvenes como es TikTok. Estos posts tratan el mismo tema de diferentes maneras. La mayoría son revelaciones de salud mental personales para dar visibilidad. Generalmente, reciben más comentarios positivos y son bien recibidos por sus seguidores. Sin embargo, parte de los posts seleccionados representan una situación no real para mostrar posibles problemas que sufren los influencers en el ámbito de la salud mental, los cuales, fueron recibidos de manera desfavorable y recibieron numerosas críticas, predominando los comentarios negativos. La variedad de posts nos permite obtener un corpus más diverso para un mejor entrenamiento del modelo.

Para llevar a cabo la selección de comentarios se han tenido en cuenta varios aspectos como la variedad de comentarios, la longitud o la forma de expresión de estos. Se recopilaron comentarios diversos con el fin de obtener una cantidad significativa de comentarios de cada emoción y polaridad, asegurando así una mayor precisión en los resultados del modelo. Estos comentarios fueron guardados en un archivo xlsx y fueron etiquetados de manera manual teniendo en cuenta palabras clave para cada emoción:

- Amor/Admiración: te quiero, guapa, un abrazo.
- Gratitud: Gracias, bravo.
- Comprensión/empatía/identificación: Te entiendo, totalmente de acuerdo.
- Tristeza/Pena: Pobrecita, lloro, triste.
- Enfado/desprecio/burla: Payasa, mentira, dame tu dinero.

A continuación, se explicará de manera detallada cada uno de los post descargados y utilizados para el entrenamiento del modelo, así como las protagonistas de estos, datos cuyo resumen se encuentra en la Tabla 1:

- Lorenaoliversig  
[https://www.tiktok.com/@lorenaoliversig/video/7078688776817593606?is\\_from\\_webapp=1&web\\_id=7296446936789435936](https://www.tiktok.com/@lorenaoliversig/video/7078688776817593606?is_from_webapp=1&web_id=7296446936789435936): Lorena Oliver es una creadora de contenido española, con más de 30.000 seguidores. En sus redes sociales habla acerca de sus viajes y sobre todo de temas de la salud mental. El video seleccionado trata sobre el PAS (Personas Altamente Sensibles). Estas siglas se refieren a personas que son más sensibles a estímulos emocionales intensos o situaciones de un alto nivel de estrés. El video fue grabado en un coche de manera espontánea, para así mostrar una situación donde se mostrará realmente lo que sufren las personas que lo padecen. La falta de preparación previa en el video aumenta su credibilidad al otorgarle un aspecto más auténtico y espontáneo. Eso fue un aspecto clave a la hora de decantarme por él. Los comentarios que recibió este video fueron sobre todo positivos, destacando los mensajes de comprensión e identificación son las personas PAS. Sin embargo, las siglas de este término que describe a este tipo de personas fueron utilizado de manera negativa para realizar bromas y comentarios ofensivos. Estos comentarios fueron recopilados en el corpus debido a la complejidad de ver comentarios negativos evidentes en la mayoría de los videos, lo que ayudaría a un mejor aprendizaje del modelo.
- Rosariomatew  
[https://www.tiktok.com/@rosariomatew/video/7289369940199607584?is\\_from\\_webapp=1&web\\_id=7296446936789435936](https://www.tiktok.com/@rosariomatew/video/7289369940199607584?is_from_webapp=1&web_id=7296446936789435936): Rosario Cerdán Mateu es una influencer española con más de medio millón de seguidores en TikTok, sobre todo conocida por sus participaciones en realitys shows. El post seleccionado trata sobre lo que sufren por dentro las personas con ansiedad. En él se muestra mirando por la ventana de un transporte público. Sin embargo, lo especial de este video lo aporta el audio. Un trozo de la canción *In this shirt de the irrepresible* junto con una frase de la película un monstruo viene a verme la cual dice: “*Despierta, despierta, resiste, despierta, despierta*” hace que este video despierte el lado más sentimental del usuario. Sin embargo, a pesar de esto parte de los comentarios que recibió este

post son negativos, donde es criticada por ir maquillada o por sobreactuar para algunos usuarios. A pesar de eso, muchos de los comentarios fueron de comprensión, apoyo y ánimo.

- Carlaflila  
[https://www.tiktok.com/@carlaflila/video/7153256794624691462?is\\_from\\_webapp=1&web\\_id=7296446936789435936](https://www.tiktok.com/@carlaflila/video/7153256794624691462?is_from_webapp=1&web_id=7296446936789435936): Carla Flila es una de las mayores tiktokers españolas de la actualidad con 4,5 millones de seguidores. Lleva varios años dando visibilidad al tema de la salud mental mostrando a sus seguidores sus problemas con la ansiedad, la cual la llevaron a autolesionarse. En el video seleccionado donde dice lo siguiente: *“Más de 1 año con ansiedad, me sentía vacía y perdida. Pero ahora puedo chillar que por fin me he encontrado a mí misma, no me dan ataques de ansiedad y se lo que necesito para ser feliz”*, se muestra una persona feliz, animada y con vitalidad. El post fue bien recibido, con numerosos comentarios positivos en los que los seguidores expresaban todo su apoyo y aprecio. Destacan las muestras de cariño hacia la influencer y los pocos comentarios negativos que recibió.
- Un dia a la vez bel <https://www.instagram.com/reel/Cu-1CYRGOM/>: Bel Santacruz es una influencer y psicóloga española con más de 100 mil seguidores en TikTok. En su perfil aborda temas psicológicos mientras comparte videos sobre su vida privada. El post escogido fue aquel donde nos muestra un ataque de ansiedad sufrido por ella misma y donde su pareja la ayuda a superarlo. Además, acompaña el video con el siguiente texto: *“Estaba teniendo un ataque de ansiedad cuando Jor llegó a casa. Y solo se acercó, y me ayudo a regular mis emociones, las validó. Me recordó que no estoy sola, y que, aunque puedo con todo sola, también puedo apoyarme en él.”* El post fue mal recibido por los seguidores, quienes la acusaron de fingir, ya que se había grabado durante un ataque. Estos comentarios ayudaron a completar el corpus con comentarios negativos y así conseguir un mejor entrenamiento del modelo.
- Aliceinyoutubeland  
[https://www.tiktok.com/@aliceinyoutubeland/video/7288040793430920480?is\\_from\\_webapp=1&sender\\_device=pc&web\\_id=7296446936789435936](https://www.tiktok.com/@aliceinyoutubeland/video/7288040793430920480?is_from_webapp=1&sender_device=pc&web_id=7296446936789435936): Alicia es una influencer española conocida por sus videos de YouTube y con más de 50 mil

seguidores en TikTok, muestra su día a día en las redes sociales tratando todos los temas cotidianos. Destaca por mostrarse de manera natural, tanto vestida de manera casual, elegante o en pijama. Esto hace que sea muy cercana a sus seguidores. En el post de TikTok muestra dos situaciones muy opuestas. Al principio, muestra lo que la gente ve en las redes sociales, donde aparece ella mostrándose alegre, de fiesta, haciendo bromas. Pero a la mitad del video, este cambia completamente y muestra lo que no se ve en las redes sociales. En este fragmento muestra videos suyos llorando, su medicación contra la ansiedad, mensajes como: *“Pero solo me quiero morir”*, *“Es que solo quiero llorar”*, *“No puedo más”*. Todos los comentarios fueron positivos, predominando los de Amor/Admiración.

- CrisBlanco

[https://www.tiktok.com/@crisblancofdz/video/7288273751198698784?is\\_from\\_webapp=1&web\\_id=7296446936789435936](https://www.tiktok.com/@crisblancofdz/video/7288273751198698784?is_from_webapp=1&web_id=7296446936789435936): Influencer española con casi 4 millones de seguidores en TikTok. Se define a sí misma como *“Soy algo así como tu amiga virtual”*. Trata todo tipo de temas dirigiéndose de forma directa a los usuarios, dando la sensación de que te habla de manera personal. El 10 de octubre, día mundial de la salud mental, subió un post donde dice lo siguiente: *“No se ve desde fuera o al menos parece que nadie quiere verlo, pero tú lo sientes y tanto que lo sientes. Días grises llenos de angustia en los cuales tu cabeza es tu peor enemiga. No sabes cómo ni a quién acudir para pedir ayuda, por lo que decides fingir una sonrisa para no incomodar a los demás. Sin embargo, poco a poco las actividades del día a día se van haciendo cada vez más difíciles al punto de convertirse en imposibles y aun así la gente sigue sin entenderlo. No entienden que no es que no quiera, es que no puedo. Nadie disfruta con la ANSIEDAD, con la DEPRESIÓN, con la sensación de que eres un fracaso, de que eres un bicho raro que no vale para nada. Y, aun así, te dicen que sonrías y que mires el lado positivo. Si fuera tan fácil, ya lo habría hecho”*. Estas palabras van acompañadas de imágenes y videos suyos, sufriendo ataques de ansiedad, tomando la medicación, mostrando los momentos de bajón anímico. Una parte de los comentarios fueron dando las gracias por esas palabras y dar voz a las personas que lo sufren. La otra parte de comentarios fueron de tristeza y gran parte de apoyo. Los comentarios con emoción de gratitud fueron recopilados debido a que en la mayoría de los posts

seleccionados suelen ser mínimos este tipo de comentarios. Esto ayudo a tener más muestras de este tipo y por tanto tener un corpus más diverso.

- PaulaGonu <https://www.tiktok.com/@paulagonu/video/6818186597771513094>: Influencer española con más de medio millón de seguidores en TikTok y casi 2 millones en Instagram. Sus posts tratan sobre todo de su vida diaria, consejos de moda y belleza, viajes y reflexiones personales. El post seleccionado tiene dos partes diferenciadas. La primera mitad del video representa *“Cómo la gente cree que es un ataque de ansiedad”*. En dicho post se muestra en un video con sensación de agobio, respiración agitada y de sensación de ahogo. En la segunda mitad del video representa *“Yo teniendo un ataque de ansiedad”*. En él se la puede observar con la mirada perdida y pensativa. A todo esto, hay que añadir que grabó el video con una peluca y no con su pelo natural. Este video recibió numerosos comentarios negativo, de odio y enfado. La mayoría de los comentarios eran del estilo: *“Esto contribuye al estigma de los trastornos mentales... totalmente en contra de bromear sobre esto”*, *“Ojalá a todas estas típicas personas inventándose tener ansiedad o el típico de “soy bipolar” les pase algún día de vrdr”* o *“Mucho cuento es lo que tienes.”*. Estos comentarios ayudaron a coger un mayor número de muestras negativas para el entreno del modelo.
- Its.Bellido [https://www.tiktok.com/@its.bellido/video/7033475131833945349?is\\_from\\_webapp=v1](https://www.tiktok.com/@its.bellido/video/7033475131833945349?is_from_webapp=v1): Lucia Bellido es actualmente una de las influencers y tiktokers más populares en España con 10 millones de seguidores en esta red social. Es una creadora de todo tipo de contenidos, ya sean retos, bailes o tutoriales. El post escogido es un tipo de video muy famoso en esta plataforma. Consiste en ir bajando dedos si lo que dice en el audio te ha sucedido o te sucede. Se plantean 10 situaciones y en este video lucia bellido baja los dedos en las siguientes situaciones: *“Baja un dedo si tienes problemas para dormir”*, *“Baja un dedo si eres de los que sobre piensan muchísimo las cosas”*, *“Baja un dedo si te sientes roto por dentro”*, *“Baja un dedo si estas aprendiendo a amarte a ti mismo pero te es difícil”*, *“Baja un dedo si tienes amigos tóxicos o amigos falsos”*, *“Baja un dedo si te pones triste sin razón alguna”*, *“Baja un dedo si te sientes perdido o perdida y sin saber que hacer”*, *“Baja un dedo si te sientes ansioso o deprimido”*. Los comentarios



recibidos eran bastante diversos, pero destacan comentarios de tristeza y pena, por ejemplo: “A mí también me quedaron 2 dedos es tan difícil que la gente te comprenda y a siempre me siento solo” o “he llorado y todo”.

Influencer	Followers (million)	Instagram Post	Number of responses	Number of selected comments
Lorena Oliversig	0.339	<a href="https://www.tiktok.com/@lorenaoliversig/video/7078688776817593606?is_from_webapp=1&amp;web_id=7296446936789435936">https://www.tiktok.com/@lorenaoliversig/video/7078688776817593606?is_from_webapp=1&amp;web_id=7296446936789435936</a>	1925	134
Rosariomatew	0.588	<a href="https://www.tiktok.com/@rosariomatew/video/7289369940199607584?is_from_webapp=1&amp;web_id=7296446936789435936">https://www.tiktok.com/@rosariomatew/video/7289369940199607584?is_from_webapp=1&amp;web_id=7296446936789435936</a>	210	38
Carlafila	4.5	<a href="https://www.tiktok.com/@carlafila/video/7153256794624691462?is_from_webapp=1&amp;web_id=7296446936789435936">https://www.tiktok.com/@carlafila/video/7153256794624691462?is_from_webapp=1&amp;web_id=7296446936789435936</a>	240	118
Un_día_a_la_vez_bel	0.271	<a href="https://www.instagram.com/reel/Cu-1CYRGOM/">https://www.instagram.com/reel/Cu-1CYRGOM/</a>	1150	153
Aliceinyoutubeland	0.053	<a href="https://www.tiktok.com/@aliceinyoutubeland/video/7288040793430920480?is_from_webapp=1&amp;sender_device=pc&amp;web_id=7296446936789435936">https://www.tiktok.com/@aliceinyoutubeland/video/7288040793430920480?is_from_webapp=1&amp;sender_device=pc&amp;web_id=7296446936789435936</a>	16	11
CrisBlanco	3.9	<a href="https://www.tiktok.com/@crisblancofdz/video/7288273751198698784?is_from_webapp=1&amp;web_id=7296446936789435936">https://www.tiktok.com/@crisblancofdz/video/7288273751198698784?is_from_webapp=1&amp;web_id=7296446936789435936</a>	756	140
PaulaGonu	0.657	<a href="https://www.tiktok.com/@paulagonu/video/6818186597771513094">https://www.tiktok.com/@paulagonu/video/6818186597771513094</a>	987	138
Its.Bellido	10	<a href="https://www.tiktok.com/@its.bellido/video/7033475131833945349?is_from_webapp=1">https://www.tiktok.com/@its.bellido/video/7033475131833945349?is_from_webapp=1</a>	624	162

Tabla 1. Post y número de comentarios descargados

En resumen, el etiquetado de los posts descargados ha sido esencial para establecer una base sólida para el entrenamiento y aprendizaje de nuestro modelo. Al seleccionar una variedad de posts protagonizados mayormente por mujeres influyentes en TikTok, el contenido refleja las distintas maneras de afrontar y tratar el tema de la salud mental. Esta diversidad es esencial para un correcto entrenamiento del modelo y hacer que este sea más robusto y preciso a la hora de categorizar las emociones y polaridades.

El proceso de etiquetado manual, basado en palabras clave específicas para cada emoción, ha permitido construir un conjunto de datos detallado y bien definido. Esta metodología no solo contribuye a la precisión del modelo, sino que también garantiza que las diferentes emociones y reacciones del público estén representadas de manera equilibrada. En la Figura 3 se muestra un ejemplo del etiquetado manual llevado a cabo.

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

1621	Ayyyyy , que pena me ha dado leerte!! menos mal que acabas con "soy feliz, mentiría si digo que no	Negativa	Tristeza/Pena	I
1622	Como me gustaría que alguien me apoyara así 😞😞😞	Negativa	Tristeza/Pena	I
1623	YO,COMO TANTOS MAS,SOLO DESEAMOS QUE X FAVOR,LUCHA MI PRINCESA,NO ESTAS SOLA	Positiva	Amor/Admiración	I
1624	Yo ya simplemente después de diez años soy un cuerpo que camina,pero tus cancion	Positiva	Gratitud	I
1625	Yo voy al psicologo tambien y me parecio una experiencia que nunca olvidare	Positiva	Comprensión/Empatía/Identificación	I
1626	Yo voy al psicólogo para desahogame de todo lo q me quedo para mi misma🙄	Positiva	Comprensión/Empatía/Identificación	I
1627	Yo tengo depresión moderada y aunque me cuesta hago deporte ,trabajo e intento aunque no	Positiva	Comprensión/Empatía/Identificación	I
1628	yo tengo ansiedad y mi pareja es el que me contiene , aunque me dijo tu te haces la fuerte pero yo	Positiva	Comprensión/Empatía/Identificación	I
1629	Yo tengo ansiedad y depresión y varias cosas más y también voy a terapia eres muy fuerte lo c	Positiva	Comprensión/Empatía/Identificación	I
1630	Yo te quiero vale ?❤️	Positiva	Amor/Admiración	I
1631	Yo te entiendo cada día me esfuerzo para sonreír	Positiva	Comprensión/Empatía/Identificación	I
1632	Yo te celebro siempre amiga. Te quiero	Positiva	Amor/Admiración	I
1633	yo tb soy PAS BELLEZA, ánimo no estás solaaaaaa🙄🙄🙄🙄🙄	Positiva	Comprensión/Empatía/Identificación	I
1634	Yo tb estoy en este punto, me escondo y me desahogo pq hay días en los q no soy🙄	Positiva	Comprensión/Empatía/Identificación	I
1635	Yo también voy a terapia. También comparto y explico que también tengo mis momentos mal	Positiva	Comprensión/Empatía/Identificación	I
1636	Yo también veo belleza en esas fotos...GRACIAS Dulceida por compartirlas	Positiva	Gratitud	I

Figura 3: Ejemplo real de etiquetado manual

# 5

## **Análisis de la respuesta emocional en redes sociales: Polaridad y Emociones**

### **5.1 Introducción**

Este capítulo de la memoria se centrará en aplicar el modelo de predicción basado en aprendizaje profundo y realizar una optimización de los hiperparámetros utilizados para analizar las respuestas emocionales presentes en los corpus del apartado anterior, enfocado en detectar y clasificar las emociones y la polaridad expresadas en cada mensaje.

### **5.2 Análisis de polaridad en salud mental**

#### ***5.2.1 Definición del corpus de datos etiquetado***

El corpus de mensajes de Instagram y TikTok consta de 4373 mensajes, los cuales están clasificados con su correspondiente polaridad, que puede ser de 3 tipos diferentes: Positiva, Negativa e Indeterminada. Del corpus se obtienen los datos de partida, esto es, los mensajes y su polaridad, que posteriormente se usarán para el entrenamiento del modelo de Deep Learning.

En este caso, la distribución de los mensajes es la siguiente: 2751 positivos, 1281 negativos y 340 indeterminados, y su distribución se muestra en la Figura 4.

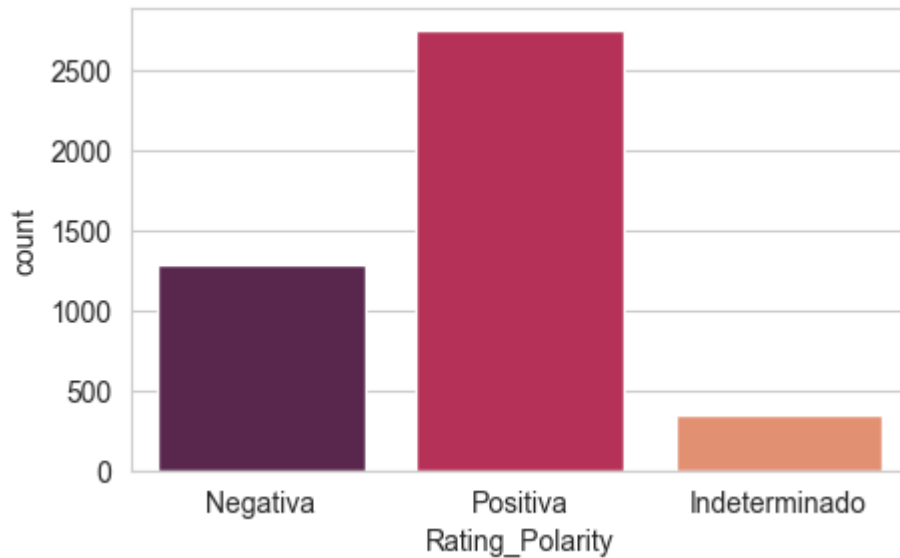


Figura 4: distribución del corpus por polaridad

Dado que cada mensaje solo puede pertenecer a una polaridad, estamos ante un problema de clasificación de etiqueta única y multiclase.

Se puede observar en la Figura 8 que las polaridades negativa y positiva tienen una mayor cantidad de muestras en comparación con la indeterminada, lo que podría ocasionar un problema de desbalance, haciendo así que el modelo prediga con mayor probabilidad que el mensaje pertenece a las clases con más muestras.

### ***5.2.2 Análisis de resultados de polaridad: Positivo, Negativo, Neutro***

En el siguiente apartado se analizarán los resultados de accuracy, precisión, recall y F1-score para cada una de las clases del corpus, esto es, polaridad positiva, negativa y neutra. Para ello se comenzará ajustando el número de neuronas y filtros de las capas, luego se seguirá optimizando el hiperparámetro de dropout de las capas y filtros de las capas, a la tasa de aprendizaje y por último el tamaño del vocabulario para obtener el número de palabras más óptimas. El tamaño del Kernel en todo momento se mantendrá con un valor constante de 8.

#### **Ajuste del número de neuronas y filtros**

Como se puede observar de los resultados obtenidos tanto en la Tabla 2, la combinación más precisa es la de 160 neuronas y 64 filtros para la capa LSTM.

Número de neuronas en capa convolucional	Filtros para capa LSTM	Dropout para capa convolucional	Dropout para capa LSTM	Tamaño del Kernel	Accuracy Validación cruzada
192	96	0.2	0.3	8	77.58%
192	64	0.2	0.3	8	77.54%
180	256	0.2	0.3	8	76.76%
180	96	0.2	0.3	8	77.58%
160	128	0.2	0.3	8	77.81%
160	64	0.2	0.3	8	78.16%
150	256	0.2	0.3	8	76.79%
128	128	0.2	0.3	8	76.47%
128	64	0.2	0.3	8	77.28%

Tabla 2: resultados obtenidos para variación de neuronas y filtros LSTM

### Ajuste de las tasas de dropout

Como se puede observar en la tabla 3, la variación en las tasas para la capa convolucional y la capa LSTM afecta al resultado final, siendo la combinación más óptima aquella con los valores de 0.2 y 0.5 respectivamente.

Número de neuronas en capa convolucional	Filtros para capa LSTM	Dropout para capa convolucional	Dropout para capa LSTM	Tamaño del Kernel	Accuracy Validación cruzada
160	64	0.2	0.3	8	78,16%
160	64	0.3	0.3	8	78.03%
160	64	0.4	0.3	8	78.03%
160	64	0.5	0.3	8	77.25%
160	64	0.6	0.3	8	76.33%
160	64	0.7	0.3	8	76.40%
160	64	0.8	0.3	8	76.14%
160	64	0.2	0.4	8	77.71%
160	64	0.2	0.5	8	78.30%
160	64	0.2	0.6	8	77.05%
160	64	0.2	0.7	8	77.61%
160	64	0.2	0.8	8	77.81%

Tabla 3: resultados obtenidos para variación de tasas de dropout

### Ajuste de tasa de aprendizaje del optimizador Adam

Tal y como se observa en la Tabla 4 para la tasa de aprendizaje se observa que, como para el caso de dos polaridades, se produce variación en la métrica de precisión del modelo al modificar el valor de dicho parámetro. Se produce una diferencia de un 29,3% entre la tasa óptima y la tasa que devuelve el peor resultado, de ahí la importancia de utilizar la tasa de aprendizaje apropiada.

Tasa de aprendizaje	Accuracy Validación cruzada
0.006	76.40%
0.008	75.52%
0.001	78.30%
0.02	62.64%
0.04	55.35%

Tabla 4: resultados obtenidos para variación de tasa de aprendizaje

### Reducción del número total de palabras únicas del corpus

La reducción del número de palabras únicas del corpus afecta a la precisión del modelo, en concreto, se observa en la Tabla 5 que al disminuir el número de palabras se obtienen mejores resultados, teniendo cuidado de no descartar demasiadas palabras como para que se produzca una pérdida de información relevante para el modelo. El valor óptimo se observa para 4980 palabras, con un valor de accuracy de 82.4%. Se han realizado en este caso saltos de 200 palabras para obtener un barrido más completo de los resultados.

Tamaño vocabulario	Accuracy Validación cruzada
Todas (6780)	78.30%
6580	79.60%
6380	79.70%
6180	78.66%
5980	79.60%
5780	79.77%
5580	80.26%
5380	81.47%
5180	81.40%
4980	82.4 %
4780	80.49%
4580	80.16%
4380	79.28%
4180	78.59%
3980	77.58%
3780	78.56%

3580	78.69%
3380	76.73%
3180	78.07%
2980	79.01%
2780	78.30%
2580	78.56%
2380	78.03%
2180	77.77%
1980	78.33%

Tabla 5: resultados obtenidos para variación de tamaño del vocabulario

Es apreciable cómo la utilización de un menor número de palabras, descartando aquellas de menor relevancia nos conduce en un principio a un aumento en la precisión del modelo, pero que al seguir disminuyendo el número ocurre que precisión empieza a disminuir, ya que llegará un momento en el que se comenzarán a descartar palabras importantes del corpus.

### Ajuste del tamaño del batch size

En la Tabla 6 se muestran los valores de las métricas obtenidos para la configuración con una mayor tasa de precisión, esto es, un batch\_size igual a 64.

batch size	Accuracy Validación cruzada
32	82.4%
64	82.74 %
128	80.71%
256	79.54%
512	81.76%

Tabla 6: resultados obtenidos para variación de batch\_size

### Resumen de hiperparámetros del modelo final y resultados

En la Figura 5 se aprecia la concatenación de los subconjuntos train y test, la configuración del bucle, los valores finales para el número de neuronas de la capa convolucional, en este caso 160, el número de filtros LSTM, en este caso 64, las tasas de dropout que son 0.2 y 0.5 respectivamente, el optimizador Adam con un learning rate de 0.001, el tamaño del vocabulario de 4980 el valor del batch\_size de 64. La configuración de parámetros que ha permitido optimizar el modelo se representa en la Tabla 7 a modo de resumen, con dichos parámetros y valores alcanzados.

```
from keras.models import Sequential
from sklearn.utils.multiclass import type_of_target
from tensorflow.keras.callbacks import EarlyStopping
import numpy as np
from keras.layers import Embedding, Flatten, Dense, LSTM, Conv1D, GlobalMaxPooling1D, MaxPooling1D, Bidirectional, GRU
from sklearn.model_selection import KFold
import statistics
from tensorflow.keras.optimizers import Adam

X = np.concatenate((X_train, X_test), axis=0)
y = np.concatenate((y_train, y_test), axis=0)

#y=np.argmax(y,axis=1)
print(type_of_target(y))
acc_per_fold=[]
loss_per_fold=[]
print(type_of_target(y))
kf=KFold(n_splits=5, shuffle=True, random_state=999)
cvscores=[]
for train, test in kf.split(X_train, y_train):
    model = Sequential()
    embedding_layer = Embedding(vocab_size, 200, input_length=maxlen)
    #embedding_layer = Embedding(vocab_size, 300, input_length=maxlen, weights=[embedding_matrix],trainable=False)

    model.add(embedding_layer)

    model.add(Conv1D(160, 8, activation='relu'))

    model.add(MaxPooling1D(10))

    model.add(LSTM(64, dropout=0.2, recurrent_dropout=0.5))

    #Si queremos 3 clases usaremos 3 neuronas en la última capa densa, por el contrario usaremos 4
    model.add(Dense(3, activation='softmax'))
    #Si queremos 2 clases usaremos 3 neuronas en la última capa densa, por el contrario usaremos 4
    #model.add(Dense(2, activation='softmax'))
    model.compile(optimizer=Adam(learning_rate=0.01), loss='categorical_crossentropy', metrics=['accuracy'])
    model.summary()

    early_stop = EarlyStopping(monitor = 'accuracy', mode = 'max', verbose = 1, patience = 5)
    model.fit(X[train], y[train], epochs=100, batch_size=64, verbose=1, validation_data = (X[test], y[test]), callbacks=[early_stop])

    scores = model.evaluate(X[test], y[test], verbose=1)
    print(f'Score for fold : {model.metrics_names[0]} of {scores[0]}; {model.metrics_names[1]} of {scores[1]*100}%')
    acc_per_fold.append(scores[1] * 100)
    loss_per_fold.append(scores[0])
print(acc_per_fold)
print(statistics.mean(acc_per_fold))
```

Figura 5: modelo tras optimización de 3 polaridades

Parámetros	Valores
Número Neuronas	160
Número filtros	64
Tasa dropout	0.2
Tasa recurrent dropout	0.5
Tasa de aprendizaje	0.001
Vocab_size	4980
Batch_size	64

Tabla 7: parámetros optimizados

En la Tabla 8 se observan los siguientes valores de calidad para cada una de las clases por separado:

- Polaridad negativa: precisión del 73%, lo cual nos dice el porcentaje de las predicciones negativas que en realidad lo eran. Recall del 84% nos indica que de todos los mensajes que en realidad eran negativos solo se escogieron como tal el 81%. Si comparamos con el corpus anterior vemos un mínimo empeoramiento,



pasando del 78% a un 73% actual en la precisión, pero se mejora el Recall pasando de un 67% a un 85% actual.

- Polaridad positiva: precisión del 91%, lo cual nos dice el porcentaje de las predicciones positivas que en realidad lo eran. Recall del 89% nos indica que de todos los mensajes que en realidad eran positivos solo se escogieron como tal el 89%. Si comparamos con el corpus anterior podemos observar una ligera diferencia, con una mejora de la precisión, pasando del 85% al 91% y una disminución del Recall, pasando del 94% al 89% en Recall.
- Polaridad Indeterminada: precisión del 41%, lo cual nos dice el porcentaje de las predicciones de polaridad indeterminada que en realidad lo eran. Recall del 27% nos indica que de todos los mensajes que en realidad eran de polaridad indeterminada solo se escogieron como tal el 27%. Si comparamos con los resultados obtenidos con el corpus anterior, observamos que se disminuye mínimamente el nivel de precisión de un 42% a un 41%. En cuando a Recall, observamos una mínima mejora, pasando del 22 % al 27%.

Estos ligeros cambios pueden ser debido a que estamos integrando comentarios de una nueva red social, TikTok, donde el formato de las publicaciones tiene un patrón distinto y existe una mayor diversidad lingüística.

Polaridad	Precision	Recall	F1-score	Support
Negativo	73%	84%	78%	348
Negativo (corpus anterior)	78%	67%	72%	186
Positivo	91%	89%	90%	859
Positivo (corpus anterior)	85%	94%	89%	451
Indeterminado	41%	27%	32%	105
Indeterminado (corpus anterior)	42%	22%	29%	49
Global (weighted avg)	82,74%	83%	82%	1312
Global (weighted avg) (corpus anterior)	80%	81%	80%	686

Tabla 8: valores de métricas para configuración óptima 3 polaridades

Por otro lado, la matriz de confusión aporta una representación visual de las predicciones realizadas por el modelo y compararlas con las clases a las que verdaderamente pertenecen los datos tratados. En este caso en concreto (Figura 6), se

aprecia como el modelo experimenta el mayor grado de dificultad a la hora de clasificar los mensajes de polaridad Indeterminada, debido a la posible ambigüedad que representan los mismos, ya que en muchos casos los mensajes suelen tener inherente una polaridad y no son totalmente neutros. También se puede deber al desbalance existente entre las clases de datos, como se puede observar en la Figura 4, puesto que el número de mensajes de polaridad indeterminada es bastante inferior al resto de clases.

Otro aspecto destacable es la mínima mejora que experimenta el modelo con el nuevo corpus respecto al anterior. Esto se puede observar también de manera visual en la Figura 7, donde adquieren un color más fuerte los resultados que eran positivos y se predijeron como positivos y viceversa. Con el nuevo corpus esos tonos más oscuros desaparecen para dar paso a tonos claros casi blancos lo que nos indica una mayor precisión en las predicciones del modelo. En la polaridad donde más se acentúa esta no mejora en la polaridad Indeterminado. Esto se puede deber a que al incorporar al corpus un mayor número de palabras, estas hayan sido categorizadas como indeterminado, pero tenga un mayor valor en cualquiera de las otras dos polaridades.

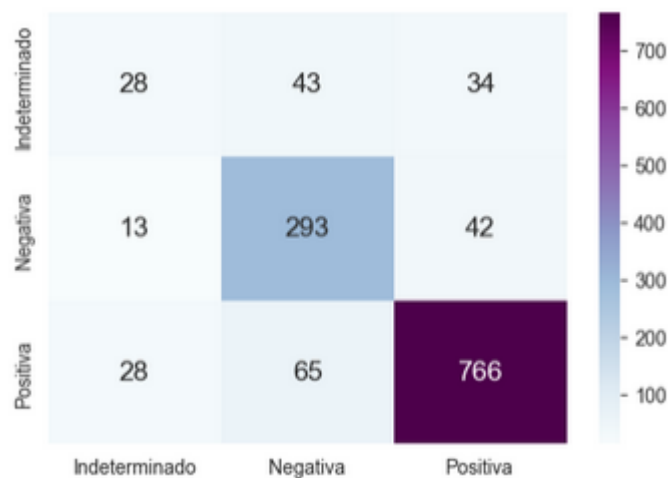


Figura 6: matriz de confusión para configuración óptima 3 polaridades

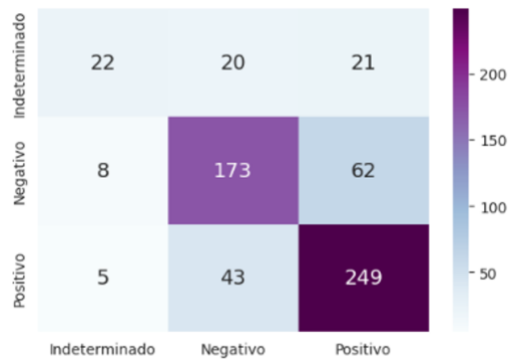


Figura 7: matriz de confusión para configuración óptima 3 polaridades antiguo corpus



También se crea un informe que muestra las principales métricas de la clasificación (classification\_report).

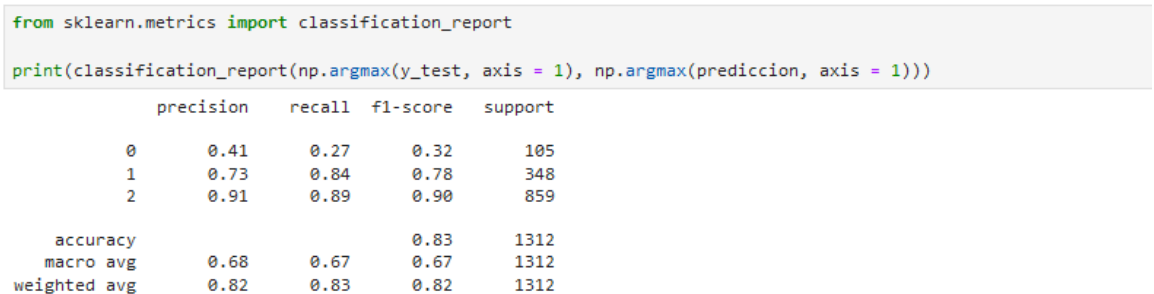


Figura 8: resultados para configuración óptima 3 polaridades

En la Figura 8 se observa una representación de los valores finales obtenidos una vez se han optimizado todos los hiperparámetros.

## 5.3 Análisis de emociones en salud mental

### 5.3.1 Definición del corpus de datos etiquetado

El corpus de mensajes de Instagram y TikTok (4373 mensajes), también están clasificados con su correspondiente emoción, que pueden ser: Enfado/desprecio/burla, Pena/tristeza, Gratitud, Empatía/Comprensión/Identificación, Amor/Admiración, Indeterminado. Del corpus se obtienen los datos de partida, esto es, los mensajes y la emoción que evoca, y estos datos posteriormente se usarán para el entrenamiento del modelo de Deep Learning. Su distribución se muestra en la Figura 9.

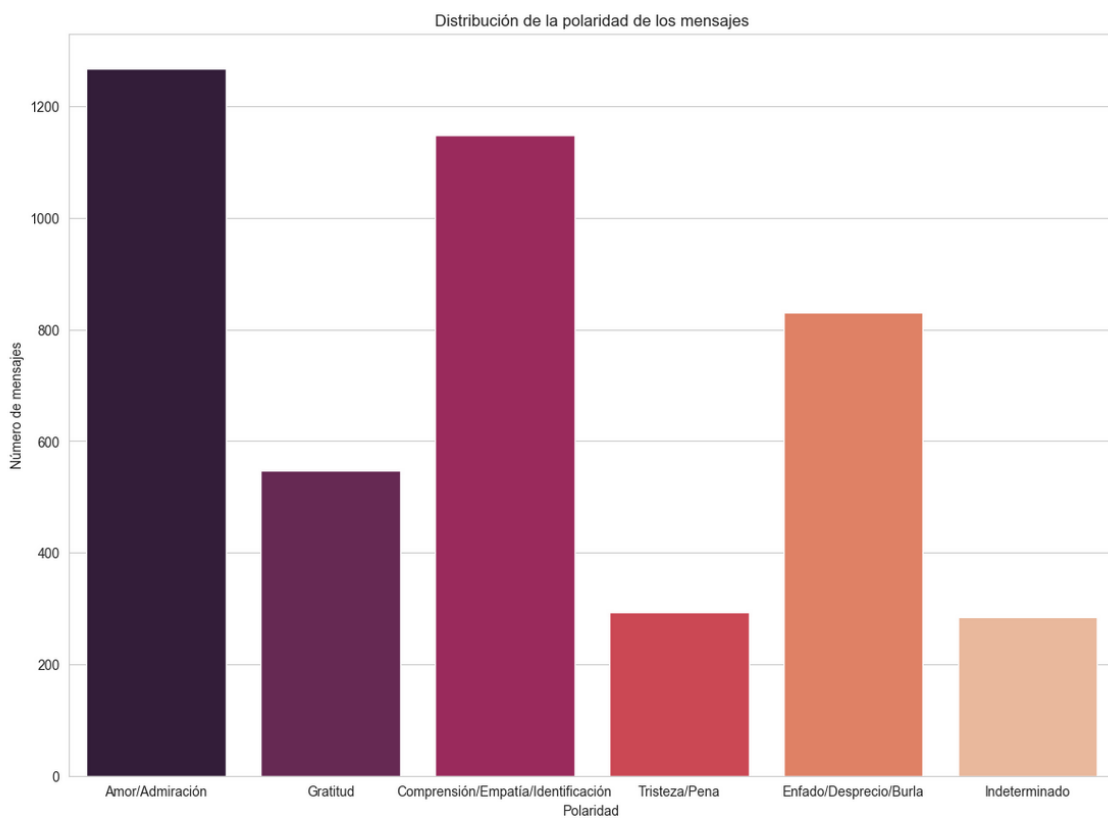


Figura 9: distribución del corpus por emociones

En este caso, la distribución de los mensajes es la siguiente: 830 de Enfado/desprecio/burla, 293 de Pena/tristeza, 548 de Gratitud, 1148 de Empatía/Comprensión/Identificación, 1267 de Amor/Admiración y 285 de indeterminado.

Dado que cada mensaje solo puede pertenecer a una polaridad, estamos ante un problema de clasificación de etiqueta única y multiclase. Se puede observar en la Figura 9 que las emociones de amor/admiración y Comparación/Empatía/Identificación tienen una

mayor cantidad de muestras en comparación con el resto, lo que podría ocasionar un problema de desbalance, haciendo así que el modelo prediga con mayor probabilidad que el mensaje pertenece a la clase con más muestras. En la Figura 10, se presenta una nube de palabras, la cual representa de manera gráfica las palabras de mayor relevancia para una emoción específica, en este caso la emoción de gratitud. Podemos observar que la emoción gratitud es de las que menos comentarios tiene, sin embargo, en la nube de palabras aparecen bastantes. Esto se debe a que emociones como Amor/Admiración o Empatía/Comprensión/Identificación también contienen palabras de gratitud, de ahí el tener palabras repetidas en la nube de la emoción gratitud (Figura 10) y en la nube de emoción Amor/Admiración (Figura 11).



Figura 10: distribución del corpus por emoción gratitud

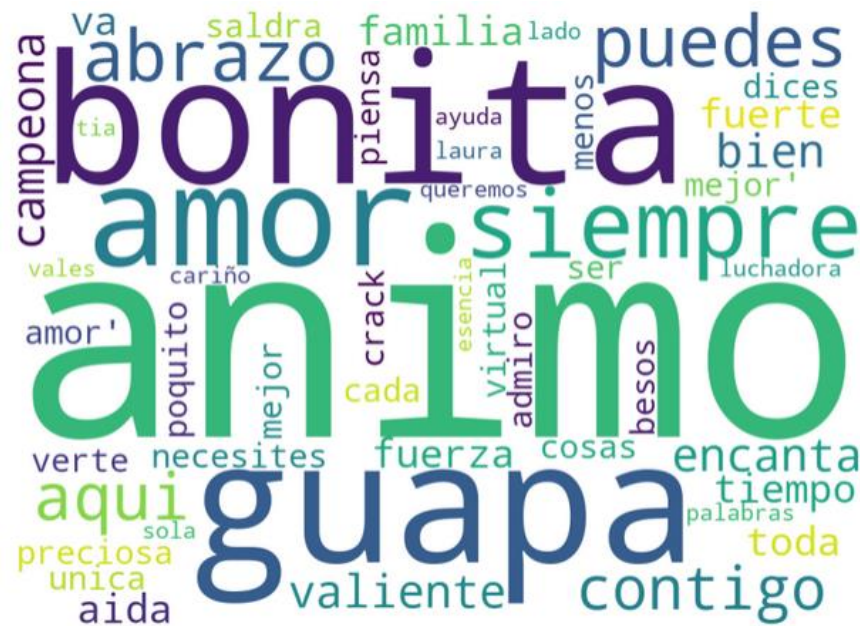


Figura 11: distribución del corpus por emoción amor/admiración

### ***5.3.2 Resultados de emociones para el dataset de salud mental***

El modelo fue entrenado con un conjunto de datos, los cuales fueron separados en dos subconjuntos: 30% de los datos fueron utilizados para prueba y los datos restantes (70%) fueron utilizados para entrenamiento. Para entrenar el modelo se realizaron diferentes ajustes de parámetros para obtener el resultado óptimo. Estos parámetros fueron: Número de neuronas en la capa convolucional, filtros para capa LSTM, dropout para capa LSTM, dropout para capa convolucional, tasa de aprendizaje, tamaño del vocabulario y tamaño del batch\_size. El tamaño del Kernel se mantuvo en todo momento constante de valor 8. En la fase de entrenamiento se siguieron los siguientes pasos:

- **Preparación:** Se crearon dos subconjuntos de datos, resultado de concatenar los subconjuntos de entrenamiento y de prueba. El resultado es un conjunto 'X' (resultado de concatenar 'X\_train' y 'X\_test') y un subconjunto 'Y' (resultado de concatenar 'y\_train' y 'y\_test')
- **Validación cruzada:** Se definieron 2 listas ('acc\_per\_fold' y 'loss\_per\_fold') para guardar los resultados y se empleó el método KFold con 5 divisiones.

- **Bucle de validación cruzada:** Este bucle iteró según los índices de entrenamiento generados por el método KFold.

En la fase de prueba se tuvieron en cuenta diferentes valores como precisión, recall y F1-score que nos proporcionan información sobre la capacidad del modelo para categorizar los comentarios de manera correcta.

A continuación, se analizarán los resultados de accuracy, precisión, recall y F1-score para cada una de las clases del corpus, esto es, Enfado/desprecio/burla, Pena/tristeza, Gratitud, Empatía/Comprensión/Identificación, Amor/Admiración y Indeterminado. Para ello se comenzará ajustando el número de neuronas y filtros de las capas, luego se seguirá optimizando el hiperparámetro de dropout de las capas y filtros de las capas, a continuación, se ajustará la tasa de aprendizaje y por último se ajustará el tamaño del vocabulario para obtener el número de palabras más óptimas. El tamaño del Kernel en todo momento se mantendrá con un valor constante de 8.

### Ajuste del número de neuronas y filtros

Como se puede observar de los resultados obtenidos tanto en la Tabla 9, la combinación más precisa es la de 192 neuronas y 96 filtros para la capa LSTM.

Número de neuronas en capa convolucional	Filtros para capa LSTM	Dropout para capa convolucional	Dropout para capa LSTM	Tamaño del Kernel	Accuracy Validación cruzada
192	96	0.2	0.3	8	65.57 %
192	64	0.2	0.3	8	64.92%
180	256	0.2	0.3	8	64.85 %
180	96	0.2	0.3	8	64.85%
160	128	0.2	0.3	8	64.72 %
160	64	0.2	0.3	8	65.47 %
150	256	0.2	0.3	8	64.52%
128	128	0.2	0.3	8	64.72%
128	64	0.2	0.3	8	65.11%

Tabla 9: resultados obtenidos para variación de neuronas y filtros LSTM sin capas ocultas

### Ajuste de las tasas de dropout

La variación de los hiperparámetros con 6 emociones no afecta en gran medida a la precisión del modelo como podemos observar en la Tabla 10. Entre las diferentes pruebas realizadas se puede observar una diferencia de un 2% entre la mejor combinación de valores de los hiperparámetros (0.2 para capa convolucional y 0.5 para capa LSTM) y la peor combinación (0.4 para capa convolucional y 0.3 para capa LSTM).

Número de neuronas en capa convolucional	Filtros para capa LSTM	Dropout para capa convolucional	Dropout para capa LSTM	Tamaño del Kernel	Accuracy Validación cruzada
192	96	0.2	0.3	8	65.57 %
192	96	0.3	0.3	8	64.26%
192	96	0.4	0.3	8	64.07%
192	96	0.5	0.3	8	64.46%
192	96	0.6	0.3	8	65.21%
192	96	0.7	0.3	8	65.60%
192	96	0.8	0.3	8	64.39%
192	96	0.2	0.4	8	64.53%
192	96	0.2	0.5	8	66.23%
192	96	0.2	0.6	8	64.66%
192	96	0.2	0.7	8	64.49%
192	96	0.2	0.8	8	66.09%

Tabla 10: resultados obtenidos para variación de tasas de dropout

### Ajuste de tasa de aprendizaje del optimizador Adam

Como resultado del estudio de la tasa de aprendizaje (Tabla 11) se puede observar que, dependiendo del valor que se le asigne, se produce una diferencia de aproximadamente un 55,5% en los resultados obtenidos. A continuación, se muestran los valores de las métricas obtenidos para la configuración con una mayor tasa de precisión, esto es, una tasa de aprendizaje igual a 0.001.

Tasa de aprendizaje	Accuracy Validación cruzada
0.006	65.77%
0.008	64.52%
0.001	66.23%
0.02	62.76%
0.04	36.08%

Tabla 11: resultados obtenidos para variación de tasa de aprendizaje



### Reducción del número total de palabras únicas del corpus

Como se mostraba en el estudio de polaridades, al disminuir el número de palabras se obtienen mejores resultados, teniendo cuidado de no descartar demasiadas palabras como para que se produzca una pérdida de información relevante para el modelo, tal y como se muestra en la Tabla 12. Teniendo en cuenta esto, el mejor tamaño del vocabulario, obteniendo un 68,97% de precisión es 3380 palabras.

Tamaño vocabulario	Accuracy Validación cruzada
Todas (6780)	66.23%
6580	65.31%
6380	67.96%
6180	65.87%
5980	68.06%
5780	67,73%
5580	67,57%
5380	68,06%
5180	66,72%
4980	67,07%
4780	67,27%
4580	67,53%
4380	66,85%
4180	67,27%
3980	67,79%
3780	68,84%
3580	68,32%
3380	68,97%
3180	68,61%
2980	67,60%
2780	68,61%
2580	68,55%
2380	68,91%
2180	67,57%
1980	68,3%

Tabla 12: resultados obtenidos para variación de tamaño del vocabulario

Es apreciable cómo la utilización de un menor número de palabras nos conduce en un principio a un aumento en la precisión del modelo, pero que al seguir disminuyendo el número la precisión empieza a disminuir, ya que llegará un momento en el que se comenzarán a descartar palabras importantes del corpus.

### Ajuste del tamaño del batch size

En la Tabla 13 se muestran los resultados obtenidos para el proceso de optimización del batch\_size, donde el mejor resultado se observa para un valor de 256.

batch size	Accuracy Validación cruzada
32	45.08 %
64	52.36 %
128	57.86%
256	69,69%
512	68.97%

Tabla 13: resultados obtenidos para variación del batch\_size

## Resumen de hiperparámetros del modelo final y resultados

En la Figura 12 se muestra el resultado final del modelo una vez se han realizado todos los ajustes de hiperparámetros que se han ido mencionando paso a paso.

```
from keras.models import Sequential
from sklearn.utils.multiclass import type_of_target
from tensorflow.keras.callbacks import EarlyStopping
import numpy as np
from keras.layers import Embedding, Flatten, Dense, LSTM, Conv1D, GlobalMaxPooling1D, MaxPooling1D, Bidirectional, GRU
from sklearn.model_selection import KFold
import statistics
from tensorflow.keras.optimizers import Adam

X = np.concatenate((X_train, X_test), axis=0)
y = np.concatenate((y_train, y_test), axis=0)
filters = 180
units = 96
#y=np.argmax(y,axis=1)
print(type_of_target(y))
acc_per_fold=[]
loss_per_fold=[]
kf=KFold(n_splits=5, shuffle=True, random_state=999)
#ACUERDATE
cv_scores=[]
for train, test in kf.split(X_train, y_train):
    model = Sequential()
    embedding_layer = Embedding(vocab_size, 200, input_length=maxlen)
    #embedding_layer = Embedding(vocab_size, 300, input_length=maxlen, weights=[embedding_matrix],trainable=False)

    model.add(embedding_layer)

    model.add(Conv1D(192, 8, activation='relu'))

    model.add(MaxPooling1D(10))

    model.add(LSTM(96, dropout=0.2, recurrent_dropout=0.5))

    #Si queremos 6 clases (número de emociones) usaremos 6 neuronas en la última capa densa
    model.add(Dense(6, activation='softmax')), #hay que poner la , para que entienda que no son la misma línea esta y la de abajo luego vuelve a
    model.compile(optimizer=Adam(learning_rate=0.01), loss='categorical_crossentropy', metrics=['accuracy'])

    model.summary()

    early_stop = EarlyStopping(monitor = 'accuracy', mode = 'max', verbose = 1, patience = 5) #cambio un poco el patience para que se entrene más
    model.fit(X[train], y[train], epochs=100, batch_size=256, verbose=1, validation_data = (X[test], y[test]), callbacks=[early_stop])

    scores = model.evaluate(X[test], y[test], verbose=1)
    print(f'Score for fold : {model.metrics_names[0]} of {scores[0]}; {model.metrics_names[1]} of {scores[1]*100}%')
    acc_per_fold.append(scores[1] * 100)
    loss_per_fold.append(scores[0])
print(acc_per_fold)
print(statistics.mean(acc_per_fold))
```

Figura 12: modelo tras optimización de 6 emociones

Se aprecia la concatenación de los subconjuntos train y test, la configuración del bucle, los valores finales para el número de neuronas de la capa convolucional, en este caso 192, el número de filtros LSTM, en este caso 96, las tasas de dropout que son 0.2 y 0.5 respectivamente, el optimizador Adam con un learning rate de 0.001 y el valor del

batch\_size de 256. La configuración de parámetros que ha permitido optimizar el modelo se representa en la Tabla 14 a modo de resumen.

Parámetros	Valores
Número Neuronas	192
Número filtros	96
Tasa dropout	0.2
Tasa recurrent dropout	0.5
Tasa de aprendizaje	0.001
Vocab size	3380
Batch size	256

Tabla 14: parámetros optimizados

En la Tabla 15 se muestran los valores para todas las métricas con el corpus actual y con el corpus anterior y así poder comparar resultados y posibles mejoras. Se observa que las dos emociones que mejor reconoce son Amor/admiración y Gratitud, mientras que las emociones que más confunde son Indeterminado y Tristeza/Pena, las cuales pueden guardar características en común que lleven a confusión a la hora de que el modelo las clasifique. Además, podemos comparar los resultados de cada emoción con los obtenidos con el anterior corpus. Vemos que en alguna de las emociones los resultados obtenidos son ligeramente peores que las pruebas pasadas. Esto puede ser debido a que estamos integrando comentarios de una nueva red social, TikTok, donde el formato de las publicaciones tiene un patrón distinto y existe una mayor diversidad lingüística.

Emoción	Precision	Recall	F1-score	Support
Amor/admiración	81%	82%	82%	400
Amor/admiración (corpus anterior)	77%	81%	79%	207
Comprensión/Empatía/Identificación	62%	69%	65%	333
Comprensión/Empatía/Identificación (corpus anterior)	66%	77%	71%	190
Enfado/Desprecio/Burla	70%	77%	73%	236
Enfado/Desprecio/Burla (corpus anterior)	84%	68%	75%	147
Gratitud	91%	84%	87%	178
Gratitud (corpus anterior)	85%	85%	65%	61
Tristeza/Pena	46%	42%	44%	73
Tristeza/Pena (corpus anterior)	60%	51%	55%	41
Indeterminado	38%	20%	26%	92
Indeterminado (corpus anterior)	34%	27%	30%	41
Global (weighted avg)	69,69%	72%	71%	1312
Global (weighted avg) (corpus anterior)	73%	72%	72%	687

Tabla 15: valores de métricas para configuración óptima 6 emociones

Por otro lado, la matriz de confusión de la Figura 13, muestra como el modelo experimenta el mayor grado de dificultad a la hora de clasificar los mensajes las emociones de tristeza y pena, debido a la posible ambigüedad que representan los mismos. También se puede deber al desbalance existente entre las clases de datos, como se puede observar en la Figura 9. Se puede destacar también el hecho de que se predicen muchos mensajes como emoción de amor, aunque no pertenezcan a la misma, esto se aprecia en la primera columna de la matriz de confusión, suceso el cual puede deberse al desbalance entre el tamaño de esta clase y el resto.

Comparando con los resultados obtenidos con el corpus anterior (Figura 14), podemos observar un pequeño empeoramiento en alguna de las clases y en otras una pequeña mejora. Esto se puede deberse a diferentes aspectos como el incremento de palabras del corpus, la similitud entre las diferentes emociones o una mayor desigualdad entre el número de comentarios de cada emoción. Cuantos más comentarios etiquetados, mayor es la probabilidad de confusión debido a que entre emociones comparten palabras clave. Mientras que con el antiguo corpus se llega a un 73% de precisión, con el nuevo se reduce hasta el 69,69%.

Error! Use the Home tab to apply Título 1 to the text that you want to appear here.



Figura 13: matriz de confusión para configuración óptima nuevo corpus

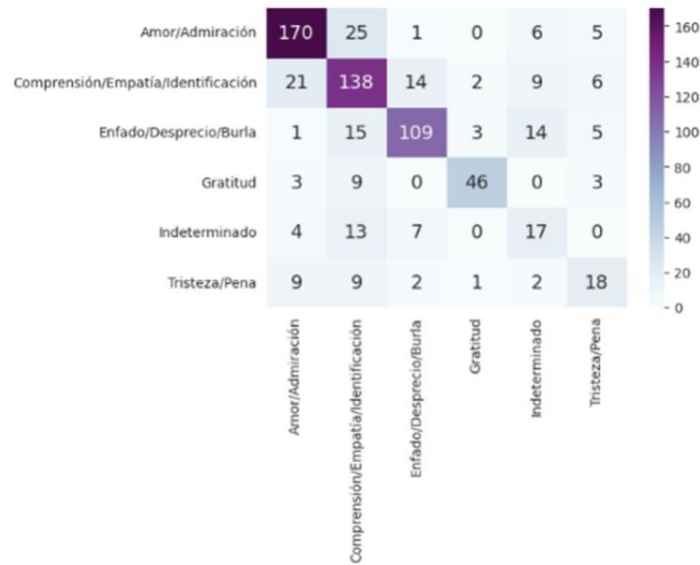


Figura 14: matriz de confusión para configuración óptima antiguo corpus

## 5.4 Conclusiones

Como primera conclusión se puede decir, una vez realizadas todas las pruebas para cada uno de los casos, que el hiperparámetro que mayor repercusión generó al ser modificado fue batch size, que al modificarlo las precisiones aumentaban o disminuían en gran medida. Otro de los parámetros que al modificarlos obteníamos una gran mejora fue la Tasa de aprendizaje. Esta dependiendo del valor asignado mejoraba hasta en un 50% la precisión del modelo.

El estudio para medir la respuesta emocional respecto a la polaridad ha dado como resultado para Instagram y TikTok una precisión del 82,74%. Si nos fijamos en los tres tipos de polaridad, tuvo una mayor precisión para positivo que para negativo. Estos resultados se pueden explicar teniendo en consideración que el corpus está formado en su mayor parte por mensajes de polaridad positiva, quedando con esto demostrada la importancia de contar con una base de datos extensa, pues las clases mayoritarias se han clasificado mejor.

En cuanto al modelo de emociones se han obtenido un 69,69% con el nuevo corpus. Si comparamos con los resultados obtenidos con el anterior corpus, podemos observar un pequeño empeoramiento en el nivel de precisión al incorporar un mayor número de comentarios. En cuanto a las emociones con mayor precisión nos

encontramos a Amor/admiración y Gritud. Esto se debe a una mayor cantidad de muestras, y por tanto, un mayor entrenamiento de estas.

Estos resultados nos hacen darnos cuenta de la importancia de tener un corpus variado con un número de comentarios similar para cada emoción o polaridad, pero a la vez los resultados nos muestran que un corpus con mayor número de comentarios no implica una mejora debido a la similitud de ciertas palabras entre ciertas polaridades y emociones.

# 6

## Interfaz gráfica para la implementación del modelo

### 6.1 Introducción

Este capítulo de la memoria se centrará en implementar una interfaz gráfica para que el usuario final pueda utilizar el modelo de manera fácil e intuitiva. Para ello se han utilizado la herramienta de Anaconda llamada Jupyter Notebook para el desarrollo de esta.

En los siguientes puntos de este capítulo se explica de manera detallada los pasos llevados a cabo y los resultados obtenidos, así como los problemas que han ido surgiendo a lo largo del proceso.

### 6.2 Desarrollo de la Interfaz

La interfaz ha sido desarrollada mediante la herramienta Jupyter Notebook en lenguaje Python. Para ello ha sido necesario la instalación de librerías específicas y de Anaconda.

#### ***6.2.1 Descarga de las herramientas necesarias para el desarrollo***

Lo primero a la hora de desarrollar, es comprobar si se disponen de las herramientas necesarias. Si no se disponen de ellas, se tendrán que descargar e instalar de la manera correspondiente a cada una. En este caso, para realizar el desarrollo de la interfaz fueron necesarias varias instalaciones que a continuación se detallan en las siguientes secciones.

##### **6.2.1.1 Anaconda**

Anaconda es un software gratuito que proporciona un conjunto de herramientas diseñado para la investigación y la ciencia. La instalación de Anaconda brinda acceso a diferentes entornos que le permiten codificar en Python [27]. Dentro de Anaconda se



**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

encuentran varias herramientas, pero en este caso la única que utilizaremos será Jupyter Notebook.

Para la instalación de Anaconda, basta con ir a la página oficial <https://www.anaconda.com/download/success> y seleccionar el ejecutable dependiendo del sistema operativo que dispongamos. Cuando ya se ha ejecutado e instalado, se abrirá Anaconda haciendo doble clic y nos aparece lo mostrado en la Figura 15. Como podemos observar, ya estará instalado Jupyter Notebook para su utilización.

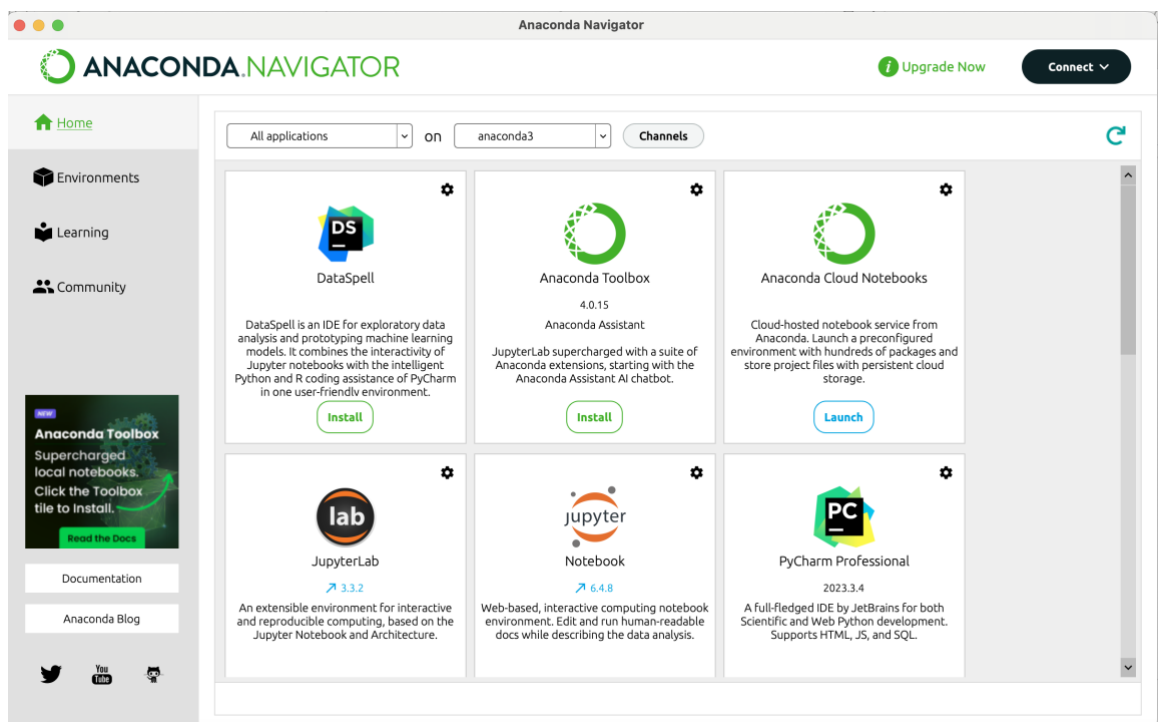


Figura 15: Entorno gráfico de Anaconda

### 6.2.1.2 Bibliotecas de Python

Para poder desarrollar ciertas funcionalidades en la interfaz ha sido necesario la instalación de las bibliotecas pertinentes. Para ello, en el terminal se ha utilizado el siguiente comando para cada una de las bibliotecas: `pip install tensorflow`. Este es el ejemplo de cómo instalar la biblioteca tensorflow.

Una vez instalado en nuestra ruta, tenemos que importarlo en el código para que, a la hora de ejecutar la interfaz, importe las bibliotecas y sus características para un perfecto desarrollo de esta. Para ello, al principio del código, se han escrito las líneas de código que se muestran en la Figura 16.

```
In [34]: from tkinter import *
import tkinter as tk
from tkinter import ttk
from tkinter import filedialog # Importar filedialog desde tkinter
from matplotlib.patches import Rectangle
import os
import pandas as pd
from matplotlib.figure import Figure
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing.sequence import pad_sequences
from keras.models import load_model
from keras.preprocessing.text import Tokenizer
import pickle # Importar la biblioteca pickle
from keras.preprocessing.text import Tokenizer
from keras.utils import pad_sequences
from keras.models import Model
import tensorflow as tf
from tensorflow.keras.models import load_model
import nltk
from nltk.tokenize import word_tokenize
import string
from keras.preprocessing.text import text_to_word_sequence
from keras.utils import pad_sequences
import numpy as np
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from collections import Counter
```

Figura 16: Importación de bibliotecas Python

Si a la hora de ejecutar el código en Jupyter nos sale un error como el mostrado en la Figura 17, nos indica que nos falta por importar la librería correspondiente.

```
NameError: name 'pandas' is not defined
```

Figura 17: Error en las librerías

### 6.2.2 Creación de la ventana y elementos gráficos

Para desarrollar la interfaz y todos sus elementos y acciones, se comenzó con la creación de una ventana gráfica. Para ello, se ha recurrido al paquete tkinter (*from tkinter import \**). Esta es la interfaz por defecto de Python para el kit de herramientas de GUI Tk. Tanto Tk como tkinter están disponibles en la mayoría de las plataformas Unix, así como en sistemas Windows [28]. Una de las decisiones que se tomaron respecto a la ventana fue la imposibilidad de reducir su tamaño y establecer así un tamaño fijo. De esta manera, se evitan problemas como la mala visibilidad o el desajuste de botones, textos y graficas. Todo esto se puede ver reflejado en el código mostrado en la Figura 18.

```
# Crear una instancia de la ventana principal
ventana = tk.Tk()

# Obtener las dimensiones de la pantalla
screen_width = ventana.winfo_screenwidth()
screen_height = ventana.winfo_screenheight()

# Obtener las dimensiones actuales de la ventana
width = ventana.winfo_width()
height = ventana.winfo_height()

# Establecer el tamaño mínimo de la ventana
ventana.minsize(1000, 1000)

ventana.title("Interface")
ventana.configure(background='#7c1324')

resultado_strvar = tk.StringVar()

# Establecer las dimensiones de la ventana
ventana.resizable(width=False, height=False)
ventana.geometry(f"{screen_width}x{screen_height}")
```

Figura 18: Creación de la ventana

Como disponemos de dos modelos (polaridad y emociones), se optó por la opción de añadir dos Frames para no tener que hacer una interfaz para cada una. Para ello se utilizó la clase `Frame()`. Un Frame es una región rectangular en la pantalla. Se utilizan para agrupar otros widgets en diseños complejos. [29]. También se añadieron botones y cuadros de texto con las clases `Button()` y `Label()` respectivamente.

Para los botones se utilizaron diferentes opciones dentro de la clase como fueron `text`, `command`, `padding` o `state`:

- Text: Indica el texto que aparecerá en el botón.
- Command: Se utiliza para llamar a una función o un método cuando se presiona el botón. La devolución puede ser un método, función o cualquier objeto.
- Padding: Indica la separación horizontal y vertical entre el botón y el resto de los elementos.
- State: Hay tres tipos de estados: Normal, disabled y active. En este proyecto se han utilizado los dos primeros estados.

Para los cuadros de texto se utilizaron diferentes opciones dentro de la clase como fueron `text` y `state`. Estas dos opciones han sido explicadas anteriormente, debido a que las opciones de las clases funcionan de igual manera.

Para la creación de gráficas se ha utilizado *matplotlib.pyplot*. Es una colección de funciones de estilo de comando que hacen que *matplotlib* funcione como MATLAB[30]. En este caso, ha sido utilizado para crear y añadir elementos a las gráficas. En estas, se muestran los resultados de ejecutar el algoritmo de manera visual, obteniendo una primera visión del resultado sin necesidad de descargar un Excel con los resultados. Las opciones utilizadas para la modificación de las gráficas han sido *figure()*, *plot()*, *yticks()*, *title()* y *subplots\_adjust()*. Pero con esta función sola no conseguimos que se muestre por pantalla en la interfaz. Para ello utilizamos la herramienta *FigureCanvasTkAgg*, lo que permite mostrar las gráficas en la posición que indiquemos. Todo esto se ha integrado dentro de una clase llamada *mostrar\_grafica1\_emociones* y *mostrar\_grafica2\_emociones* para la pestaña de emociones y en *mostrar\_grafica1\_polaridad* y *mostrar\_grafica2\_polaridad*. A continuación, voy a explicar que se muestra en cada una de las dos gráficas que se han implementado en la interfaz gráfica:

- Mostrar\_grafica1\_emociones: En esta primera gráfica, la cual se encuentra al lado izquierdo de la interfaz, se muestran las emociones en el eje x y en número de comentarios de cada una en el eje y en tanto por ciento. El tipo de gráfica es de barras, y nos muestra de manera visual, la cantidad de comentarios de cada tipo que contiene el Excel/csv cargado. Para polaridad, lo única diferencia es que en el eje x nos muestra las polaridades en vez de las emociones. En la Figura 19 se muestra un ejemplo de la gráfica en la pestaña de emociones.

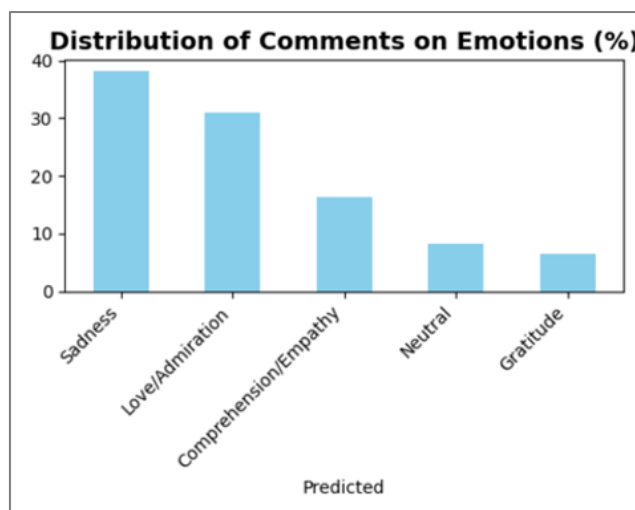


Figura 19: Distribución de comentarios por emociones en %

- Mostrar grafica2 emociones: En esta segunda gráfica, se va mostrando la evolución de las emociones de los comentarios introducidos en el Excel. De esta manera podemos observar la evolución de estos y ver si hay una evolución a lo largo del archivo previamente cargado. Para conseguirlo, se le ha asignado una codificación a cada emoción, en este caso ('Amor/Admiración': 6, 'Comprensión/Empatía': 4, 'Gratitud': 2, 'Indeterminado': 0, 'Pena/Tristeza': -2, 'Enfado/Desprecio/Burla': -4). Para polaridad, lo única diferencia es que en el eje x nos muestra las polaridades en vez de las emociones, con la siguiente codificación: ('Positive': 2, 'Neutral': 0, 'Negative': -2). En la Figura 20 se muestra un ejemplo de dicha gráfica en la pestaña de emociones.

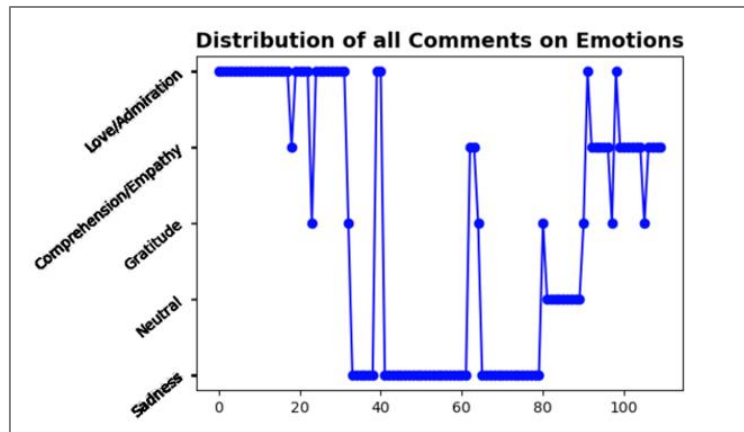


Figura 20: Distribución de todos los comentarios en emociones

### 6.2.3 Creación de clases

Con el objetivo de que el código resultara más accesible y comprensible para cualquier lector, se decidió crear clases que contuvieran las funciones a realizar en la interfaz. De esta manera, será más sencillo asignar una tarea o función a los elementos gráficos de la interfaz. Para ello, se han creado un total de 16 clases, siendo 1 de ellas cogida del archivo donde se define el modelo, la cual, define el procesado del texto eliminando palabras que provocaran confusión al modelo como son las conjunciones, nexos, etc. A continuación, se proce explicar alguna de estas clases:

- def draw\_separator (canvas): Dibuja una línea continua en la posición indicada en ella para separar elementos en la propia interfaz. A esta clase se le proporciona el atributo canvas al ser llamado, para indicar en que Frame queremos que se muestre.

- def boton\_algorithmo\_emociones(): Esta clase es llamada al pulsar el botón Run Algorithm. Su función es llamar a la función ejecutar\_algorithmo\_emociones(new\_text) y el resultado que devuelve mostrarlo en un cuadro de texto, el cual, esta deshabilitado para la escritura en la interfaz y así se asegura que lo mostrado en el sea solo el resultado de la predicción.
- def boton\_algorithmo\_polaridad(): Misma función que def boton\_algorithmo\_emociones(), con la diferencia de que en vez de llamar a la función ejecutar\_algorithmo\_emociones(new\_text), llama a ejecutar\_algorithmo\_polaridad(new\_text).
- def ejecutar\_algorithmo\_emociones(new\_text): Se definen las clases ('Love/Admiration', 'Gratitude', 'Comprehension/Empathy', 'Sadness','Anger/Mockery','Neutral'), en este caso las emociones, se carga el modelo y el tokenizador y se le proporciona el atributo new\_text que será obtenido del cuadro de texto de la interfaz. Este texto es procesado con el fin de facilitar la comprensión por parte del modelo y permitir la realización de la predicción. Esta se realiza, y el resultado se mostrará en otro cuadro de texto de la interfaz. Para realizar esto, esta clase devuelve el resultado obtenido, que en este caso será la emoción predicha.
- def ejecutar\_algorithmo\_polaridad(new\_text): Misma función que def ejecutar\_algorithmo\_emociones(new\_text) con la única diferencia de que en la definición de clases, en vez de poner las emociones, definimos las tres polaridades posibles ('Negative', 'Positive', 'Neutral') y que el modelo cargado es el de polaridad.
- def cargar\_archivo(): La función de esta clase es cargar un archivo csv/Excel para su futura predicción. Para ello, se implementó un filtro para restringir el tipo de archivo seleccionable y así evitar problemas de formatos o versiones. Dependiendo de si el archivo seleccionado es csv o Excel, se utilizará read\_excel o read\_csv. Si la carga se ha realizado de manera exitosa, se habilitará el botón Run Algorithm with Csv/Excel para así poder realizar la predicción de este. Si al cargar se hubiera producido algún error, se informaría mostrando la siguiente frase: “Error: The

*selected file is not an Excel workbook or a CSV file*”, y no se habilitará el botón Run Algorithm with Csv/Excel.

- def mostrar\_mensaje (): Muestra en la interfaz, durante 2 segundos, el siguiente mensaje cuando se ha cargado el archivo de manera satisfactoria: *"File uploaded successfully"*. Después de esos 2 segundos, se llama a la clase def borrar\_mensaje() para que
- def ejecutar\_excel\_emociones (): La función a realizar es muy similar a la de la clase def ejecutar\_algoritmo\_emociones(new\_text) con la diferencia de que en vez de introducir una cadena de caracteres se le introduce un archivo csv o Excel. El proceso es el siguiente: se carga el modelo y el tokenizador, se extraen los comentarios en una lista, se tokeniza y se pasan por el modelo, el cual realiza la predicción de cada uno de ellos y devuelve el resultado. Justo antes de devolver el resultado se llaman a las funciones mostrar\_grafica1\_emociones(df\_results) y mostrar\_grafica2\_emociones(df\_results) para que muestre por pantalla las gráficas con los resultados obtenidos.
- def ejecutar\_excel\_polaridad (): realiza la misma función que def ejecutar\_excel\_emociones () con la diferencia de que en vez de cargar el modelo de emociones se carga el modelo de polaridad y que las clases definidas son las 3 polaridades ('Negative', 'Positive', 'Neutral').
- def mostrar\_grafica1\_emociones(df\_results): Se le pasa el atributo df\_results desde la clase def ejecutar\_excel\_emociones () y mediante la librería matplotlib.pyplot, se obtiene la gráfica de barras. Esta muestra, en tanto por cierto %, los comentarios que hay para cada emoción. Para mostrarla en la interfaz, se utiliza la herramienta canvas.
- def mostrar\_grafica1\_polaridad(df\_results): Muy parecida a def mostrar\_grafica1\_emociones(df\_results) con la diferencia de que el atributo df\_results procede de la ejecución de la clase def ejecutar\_excel\_polaridad () .
- def descargar\_csv\_excel (): Descarga un Excel con los resultados obtenidos de ejecutar la clase def ejecutar\_excel\_emociones () y def ejecutar\_excel\_polaridad

(). Como ya se hizo en `def cargar_archivo()`, se da la opción al usuario de guardarlo en tres formatos diferentes ".xlsx" ".xls" o ".csv", con el fin de evitar problemas.

- `def mostrar_grafica2_emociones(df_results)`: La diferencia con la clase `def mostrar_grafica1_emociones(df_results)` es el tipo de gráfica que se muestra. En esta clase, se muestra una gráfica donde se puede observar la evolución del tipo de emoción que tiene cada comentario del archivo en orden. Para ello, se codifican las emociones para así lograr este tipo de gráfica.
- `def mostrar_grafica2_polaridad(df_results)`: Misma clase que `def mostrar_grafica2_emociones(df_results)` con la diferencia de que lo que se codifican son las tres polaridades.

### 6.3 Diseño y ejecución de la Interfaz gráfica

En este apartado se mostrará de manera visual los resultados de ejecutar el código explicado en los apartados anteriores. Al ejecutar la clase `def procesado(text)` y posteriormente ejecutar el resto del código, se muestra en una pestaña emergente el siguiente resultado, mostrado en la Figura 21 para un sistema operativo MacOS.

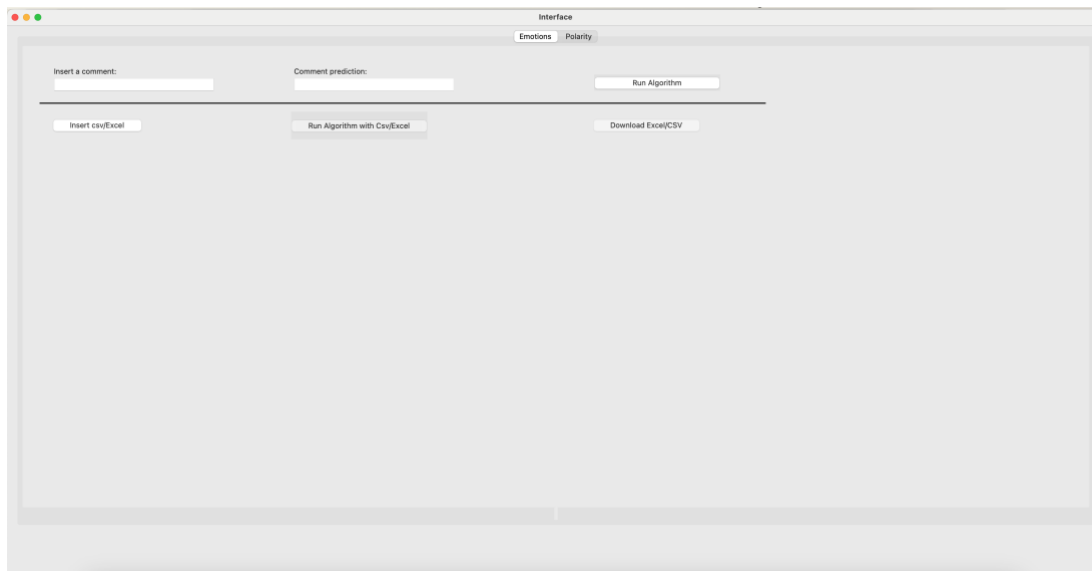


Figura 21: Interfaz gráfica en MacOS

Como se puede observar, tenemos dos partes claramente diferenciadas gracias a la línea continua que las separa. Por un lado, en la parte superior tenemos las funciones para introducir un comentario por teclado y obtener la predicción de este. Por el otro lado, en la



parte inferior se encuentran los botones para cargar y procesar un archivo y obtener la predicción de este.

Si nos centramos en la superior, se encuentra un cuadro de texto que pone *Insert a comment*, donde el usuario introducirá por teclado una palabra o una frase. Para que el modelo realice la predicción, se tendrá que hacer clic en el botón *Run Algorithm*. Inmediatamente después, aparecerá la emoción o la polaridad (dependiendo en que pestaña se esté interactuando) predicha en el cuadro de texto *Comment prediction*. Este proceso se podrá realizar todas las veces que el usuario lo desee. Un ejemplo de cómo se vería después de haber obtenido un resultado se muestra en la Figura 22.

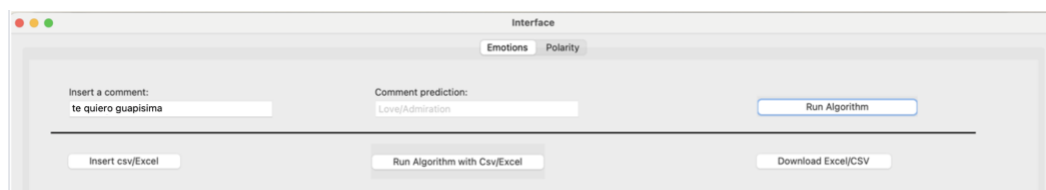


Figura 22: Resultado de ejecutar algoritmo introduciendo un comentario

Por otro lado, en la parte inferior de la línea continua se encuentran los botones para cargar y realizar la predicción de un archivo csv/Excel. En primer lugar, solo se encuentra habilitado el botón de *Insert csv/Excel*, mientras que los otros dos están deshabilitados. Cuando se hace clic sobre este botón, por pantalla se muestra una ventana emergente, como la mostrada en la Figura 23, para permitir seleccionar el archivo que se desea cargar. En la parte inferior de esta, se puede observar el filtro que aplicamos en el código para restringir el formato de los archivos que se pueden cargar. De esta manera, los archivos que cumplan con esta condición de formato saldrán con el icono normal, sin embargo, los que no cumplan con esta condición aparecerán con un color de icono más tenue.

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

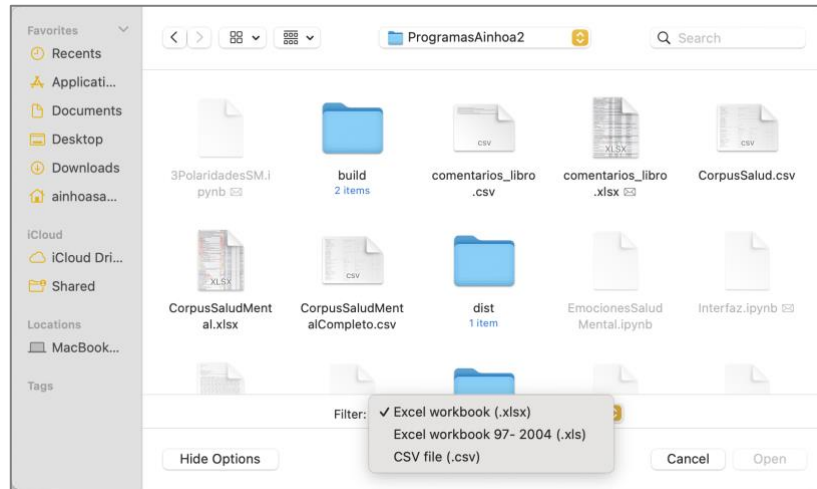


Figura 23: Ventana emergente para cargar archivo

Cuando el archivo se ha cargado de manera exitosa, se habilita el botón Run Algorithm with csv/Excel. Al hacer clic sobre él, se mostrará en la interfaz las dos graficas explicadas en el punto anterior. De esta manera, obtenemos una primera visión de los resultados obtenidos y nos aseguramos de un correcto funcionamiento. El resultado final será muy parecido mostrado en la Figura 24, solo dependerá los datos introducidos en el archivo para que las gráficas varíen en su valor.

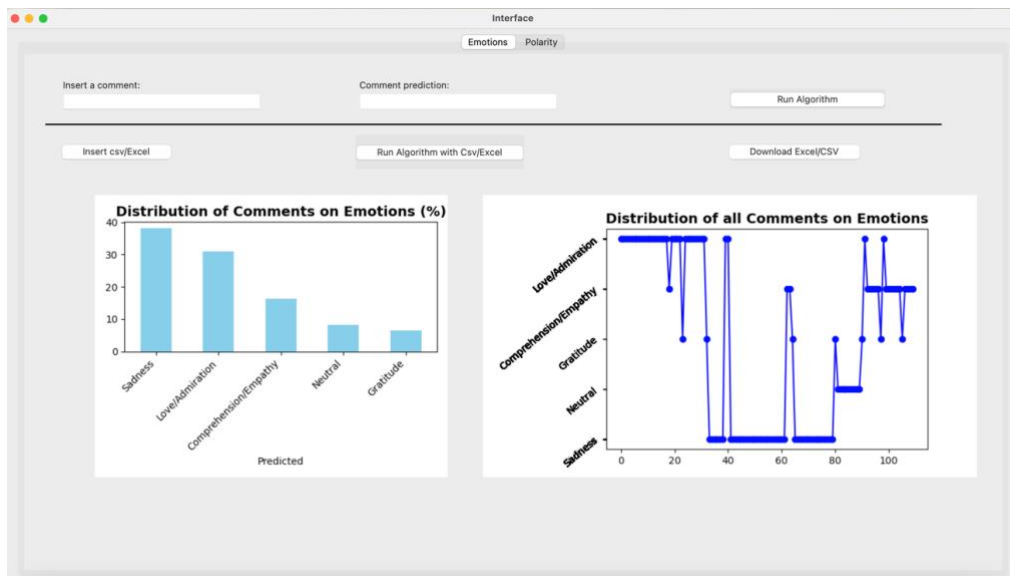


Figura 24: Resultado obtenido de ejecutar algoritmo con csv/Excel

En este punto, se habilitaría el botón Download Excel/CSV y nos permitiría descargar los resultados obtenidos en un archivo y así poder realizar diferentes tareas con los resultados. Al hacer clic en este, nos aparecerá una pestaña emergente como la mostrada

**Error! Use the Home tab to apply Título 1 to the text that you want to appear here.**

---

en la Figura 2. Podremos seleccionar la ruta donde se desee guardar y el nombre que se le quiere asignar, así como el formato.

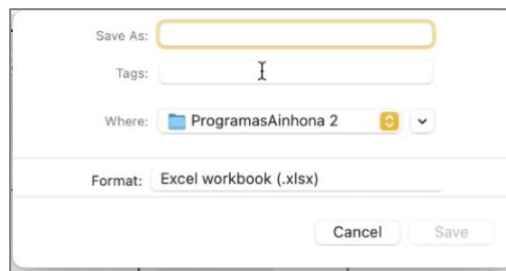


Figura 25: Ventana emergente para guardar resultados en excel/csv

Todo esto ha sido ejecutado y explicado desde la pestaña Emotions, pero funciona exactamente de la misma manera para la pestaña Polarity.

## 6.4 Conclusiones

En este capítulo se ha explicado de manera detallada el desarrollo y funcionamiento de la interfaz gráfica. De esta manera, se puede apreciar la multitud de opciones que nos proporcionan las librerías Python, así como la necesidad de crear una herramienta para que cualquier persona puede utilizar el modelo desarrollado sin la necesidad de tener conocimientos de programación. También se han tenido en cuenta posibles problemas como, por ejemplo, el formato del archivo cargado. Por eso se han tomado medidas preventivas para evitarlos. Al desarrollar graficas que muestren los resultados obtenidos de manera visual hace que el proceso de análisis de los resultados sea bastante rápido y visual dando así una primera visión de lo obtenido.

# 7

## Conclusiones y líneas futuras

### 7.1 Conclusiones

Este Trabajo de Fin de Grado se ha enfocado en la investigación y desarrollo de un modelo para la clasificación de mensajes utilizando técnicas de procesamiento de lenguaje natural y aprendizaje automático. Para ello, se ha empleado un modelo de Deep Learning basado en redes neuronales, combinando diferentes tipos de capas y utilizando redes neuronales convolucionales.

Respecto al entrenamiento del modelo mediante el nuevo corpus, el cual consta de casi el doble de palabras que el anterior, los resultados obtenidos son del 68,97% de precisión para emociones y del 82,74% de precisión para 3 polaridades. Cogiendo estos resultados y comparados con los resultados obtenidos con el anterior corpus, que fueron de 81,34% para 3 polaridades y 72.49% para emociones, concluimos que obtenemos una ligera mejora en polaridad, pero un empeoramiento más significativo en emociones. Esto se puede deber a la ambigüedad de las palabras, las cuales pueden estar presentes en más de una emoción y que dependiendo del contexto transmiten una emoción u otra.

Y con respecto a la interfaz, se ha comprobado el correcto funcionamiento de esta ejecutando el algoritmo, cargando datos e introduciéndolos por teclado. Además, se obtienen de manera correcta gráficos que muestran de primera mano los resultados obtenidos de ejecutar el algoritmo. Para que su uso fuera universal, todos los elementos (título, botones, graficas, resultados, etc.) han sido escritos en Inglés. Al ser programada mediante Python, nos asegura un correcto funcionamiento mediante diferentes sistemas operativos, obteniendo el mismo resultado con mínimas diferencias visuales que dependerán de cada sistema operativo.

## 7.2 Líneas futuras

Los resultados obtenidos ejecutando el algoritmo de polaridad y emociones, nos muestra una mínima mejoría y un pequeño empeoramiento respectivamente. Esto nos hace reflexionar sobre posibles mejoras para investigaciones futuras. Algunas de las líneas futuras a mejorar son:

- Mejora de la Interfaz: Depurar el código para que, al ejecutarlo, el tiempo de procesamiento sea menor que actualmente. También conseguir un menor tiempo de respuesta al hacer clic a los botones, al introducir el texto por teclado o al obtener las gráficas resultantes de los resultados obtenidos.
- Desarrollo de un ejecutable: Transformar el archivo Python .py en un archivo .exe para que sea más fácil para el usuario ejecutarlo. De esta manera también se podría difundir de manera más sencilla al no tener que transferir tantos archivos.
- Mejora del corpus: se ha explicado a lo largo del trabajo la importancia que tienen los datos a tratar en el estudio. Sería interesante por lo tanto conseguir una mejor calidad de los comentarios haciendo que las palabras y comentarios tengan la menor ambigüedad posible para no dar pie a confusiones entre clases.
- Realizar otras categorizaciones e integrarlas en los corpus: considerando aspectos como el impacto de los discursos de odio o los estigmas en redes sociales, lo cual puede proporcionar clasificaciones interesantes y relevantes en estos entornos virtuales.

Con este trabajo se han establecido las bases para futuras investigaciones sobre la clasificación de textos utilizando Deep Learning. Por lo tanto, las líneas de investigación futuras se resumen en: se buscará continuar desarrollando el modelo y de lo interfaz para aumentar los niveles de precisión alcanzados y un mejor rendimiento respectivamente.

# 8

## Bibliografía

[1] TFG Jesús Herrero Llanos. URL:  
<https://uvadoc.uva.es/handle/10324/57316>

[2] TFM José Carlos Sobrino Sande. URL:  
[https://github.com/jcsobrinio/TFM-Analisis\\_sentimientos\\_Twitter-UOC](https://github.com/jcsobrinio/TFM-Analisis_sentimientos_Twitter-UOC).

[3] TFM de Julia Isabel Medrano Sanz. URL:  
[https://oa.upm.es/65589/1/TFM%5C\\_Carlos\\_Hernandez%5C\\_P.pdf](https://oa.upm.es/65589/1/TFM%5C_Carlos_Hernandez%5C_P.pdf).

[4] TFM de Ivan Arévalo Nuñez. URL:  
[https://oa.upm.es/65589/1/TFM%5C\\_Carlos\\_Hernandez%5C\\_P.pdf](https://oa.upm.es/65589/1/TFM%5C_Carlos_Hernandez%5C_P.pdf).

[5] Arias Navarro, M. Á. (2023). *Análisis de la respuesta emocional en redes sociales usando técnicas de aprendizaje automático* [Trabajo Fin de Grado]. Universidad de Valladolid.

[6] Torres, J. (2020). Python deep learning: Introducción práctica con Keras y TensorFlow 2. Alpha Editorial.

[7] ¿Qué es deep learning? (s. f.). SAS.  
[https://www.sas.com/es\\_es/insights/analytics/deep-learning.html](https://www.sas.com/es_es/insights/analytics/deep-learning.html)

[8] Machine Learning vs Deep Learning. URL:  
<https://www.xataka.com/robotica-e-ia/machine-learning-y-deep-learning-como-entender-las-claves-del-presente-y-futuro-de-la-inteligencia-artificial>.

[9] Matich, D. J. (2001). Redes Neuronales: Conceptos básicos y aplicaciones. Universidad Tecnológica Nacional, México, 41, 12-16.

[10] Alonso, F. (2021, 17 septiembre). Redes neuronales y deep learning. Capítulo 2: La neurona. Future Space S.A. <https://www.futurespace.es/redes-neuronales-y-deep-learning-capitulo-2-la-neurona/>

[11] Tema: Las redes sociales en España. (2024, 23 mayo). Statista. <https://es.statista.com/temas/6566/las-redes-sociales-en-espana/#topicOverview>

[12] De Cozar, C. (2023, 17 julio). Redes sociales en España: un panorama con cifras sorprendentes. TIC Negocios. <https://ticnegocios.camaramadrid.es/servicios/tendencias/el-uso-de-las-redes-sociales-en-espana-datos-y-evolucion/>

[13] Statista. (2024, 22 mayo). Redes sociales favoritas de los millennials en España en 2023. <https://es.statista.com/estadisticas/934869/redes-sociales-preferidas-de-los-millennials-en-espana/>

[14] Del Campo Fernández-Paniagua, A. M. (2022, 26 octubre). Las Redes Sociales más utilizadas: cifras y estadísticas. Thinking For Innovation. <https://www.iebschool.com/blog/medios-sociales-mas-utilizadas-redes-sociales/>

[15] Redes sociales con más usuarios en España en 2023 | Statista. (2024, 4 abril). Statista. <https://es.statista.com/estadisticas/489153/porcentaje-de-internautas-en-las-redes-sociales-en-espana/>

[16] Fanego, I., & Fanego, I. (2024, 18 enero). Cuántos usuarios tiene WhatsApp en 2024 y otros datos interesantes - AppCritic. AppCritic. <https://appcritic.es/usuarios-whatsapp/#Numero de usuarios de WhatsApp en 2024>

[17] González Duque, R. (2011). Python para todos.

[18] Tokio School. (2024, 31 enero). ¿Qué es Python? Te contamos todo sobre este popular lenguaje | Tokio. <https://www.tokioschool.com/formaciones/cursos-programacion/python/que-es/>

[19] Brownlee, J. (2016). Deep learning with Python: develop deep learning models on Theano and TensorFlow using Keras. Machine Learning Mastery.

[20] Web oficial de Keras. URL: <https://keras.io/>.

[21] Bird, S. (2006, July). NLTK: the natural language toolkit. In Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions (pp. 69-72).

[22] Herrman, J. (2019). How TikTok is rewriting the world. *The New York Times*, 10, 412586765-1586369711.

[23] Quiroz, N. T. (2020). TikTok. *Revista argentina de estudios de juventud*.

[24] Instagram, P. O. (2016). Instagram. Online unter: <https://www.wissenschaftskommunikation.de/format/instagram>.

[25] Rolon-Mérette, D., Ross, M., Rolon-Mérette, T., & Church, K. (2016). Introduction to Anaconda and Python: Installation and setup. *Quant. Methods Psychol*, 16(5), S3-S11.

[26] Export Facebook, Instagram, Twitter, YouTube, TikTok, Vimeo Comments to CSV / Excel. (s. f.). *Export Facebook, Instagram, Twitter, YouTube, TikTok, Vimeo Comments to CSV / Excel*. [exportcomments.com. https://es.exportcomments.com/download-tiktok-comments/](https://es.exportcomments.com/download-tiktok-comments/)

[27] Rolon-Mérette, D., Ross, M., Rolon-Mérette, T., & Church, K. (2016). Introduction to Anaconda and Python: Installation and setup. *Quant. Methods Psychol*, 16(5), S3-S11.



[28] tkinter — Python interface to Tcl/Tk. (s. f.). Python Documentation. <https://docs.python.org/es/3/library/tkinter.html>

[29] Lundh, F. (1999). An introduction to tkinter. URL: [www.pythonware.com/library/tkinter/introduction/index.htm](http://www.pythonware.com/library/tkinter/introduction/index.htm).

[30] Matplotlib: Python plotting — Matplotlib 2.0.2 documentation. (s. f.). <https://matplotlib.org/2.0.2/index.html>