



UNIVERSIDAD  
DE MÁLAGA



## **ESCUELA DE INGENIERÍAS INDUSTRIALES**

**Departamento: Lenguajes y Ciencias de la Computación**

**Área de conocimiento: Lenguajes y Sistemas Informáticos.**

# **TRABAJO FIN DE GRADO**

## **Análisis de Sentimiento en Redes Sociales**

Grado en ingeniería mecánica

Autor: Rafael Reyes Podadera

Tutor: Lola Burgueño Caballero

MÁLAGA, junio de 2023



# Resumen

La comunicación verbal mediante lenguaje natural es la principal herramienta que tienen las personas para comunicarse entre sí. Esta comunicación, ya sea oral o escrita, nos permite expresar cualquier tipo de idea o emoción a otros individuos. En las últimas décadas, la llegada de las nuevas tecnologías y las redes sociales ha evolucionado y revolucionado la forma de comunicarse.

En el campo de la Inteligencia Artificial hay todo un desarrollo de décadas de investigación dedicada, no solo a procesar este tipo de lenguaje de forma automatizada, sino también que también pretende lograr el aprendizaje automático por parte de máquinas, capaces de conocer, interpretar e incluso producir lenguaje natural casi como si fueran humanos..

Este TFG hace uso de estas técnicas de Inteligencia Artificial para el procesamiento de lenguaje natural con el objetivo de entender e identificar cómo las personas expresan sus sentimientos sobre un tema específico y cómo estos sentimientos pueden ser utilizados para la toma de decisiones en diferentes campos. Para esto, usa el texto que los usuarios de forma pública publican en redes sociales como Twitter.

# Abstract

Communication using natural language is the main means of communication used by people to communicate. This communication, whether oral or written, allows us to express ideas and emotions. In recent years, the rise of new technologies and social networks have evolved and revolutionized the way in which we communicate.

In the field of Artificial Intelligence, research has been carried out for decades, not only to process this natural language in an automated way, but also to achieve automated learning by machines, so that they are capable of knowing, interpreting and even producing natural language as if they were humans or close enough to it.

This final year project uses Artificial Intelligence techniques for natural language processing in order to understand and identify how people express their feelings on a specific topic and how these feelings can be used for decision-making in different fields. For this, we use the text that users publicly post on social networks such as Twitter.

# Índice general

<b>Introducción</b>	<b>7</b>
1.1 Motivación	7
1.2 Metodología de desarrollo	8
1.3 Estructura de la memoria	8
1.4 Objetivos del estudio	9
1.5 Plan de Trabajo	10
<b>Estudio del estado</b>	<b>11</b>
2.1 Estudio del estado del arte	11
2.2 Preprocesado	12
2.3 Clasificación	12
2.3.1 Procesamiento de lenguaje natural	13
2.3.1.1 Diccionario	14
2.3.2 Aprendizaje automático	14
2.3.2.1 Naive Bayes	15
2.3.2.2 Redes Neuronales	16
2.3.2.3 SVM	17
<b>Tecnologías empleadas en el desarrollo</b>	<b>19</b>
3.1 Introducción	19
3.2 Python y el procesamiento de lenguaje natural	19
3.3 Librerías de procesamiento de lenguaje natural en Python	19
3.4 Plataformas para el desarrollo de software	23
<b>Desarrollo</b>	<b>24</b>
4.1 Lectura de la información mediante la API de twitter	24
4.2 Preprocesamiento de los datos	25
4.3 Clasificación de los datos	28
4.3.1 TextBlob	28
4.3.2 Vader	31
4.3.3 Pattern	32
4.3.4 Flair	34
4.3.5 Transformers	35
4.3.5.1 Modelo "cardiffnlp/twitter-roberta-base-emotion"	36
4.3.5.2 Modelo "michellejeli/emotion_text_classifier"	38
<b>Evaluación</b>	<b>41</b>
<b>Conclusiones y líneas futuras</b>	<b>47</b>
<b>Bibliografía</b>	<b>51</b>

# Índice de figuras

2.1 Clasificación de algoritmos de análisis de sentimientos	14
2.2 Fórmula de Naive Bayes	16
2.3 Representación del clasificador Naive Bayes	17
2.4 Representación de la arquitectura de red neuronal	18
4.1 Conjunto inicial de los datos	26
4.2 Funciones para obtener la subjetividad y polaridad de un texto utilizando TextBlob	30
4.3 Generación de un DataFrame con polaridad y subjetividad de tweets utilizando TextBlob	30
4.4 Clasificación de polaridad de tweets en negativo, neutral o positivo utilizando TextBlob	31
4.5 Clasificación de subjetividad, polaridad y análisis utilizando TextBlob	31
4.6 Funciones para obtener la subjetividad y polaridad de un texto utilizando Vader	32
4.7 Clasificación de subjetividad, polaridad y análisis utilizando Vader	33
4.8 Funciones para obtener la subjetividad y polaridad de un texto utilizando Pattern	34
4.9 Clasificación de subjetividad, polaridad y análisis utilizando Pattern	35
4.10 Función para obtener la polaridad de un texto utilizando Flair	36
4.11 Clasificación de la polaridad y análisis utilizando Flair	36
4.12 Carga del modelo "cardiffnlp/twitter-roberta-base-emotion"	38
4.13 Función para obtener el sentimiento de tweets utilizando el modelo cardiffnlp/twitter-roberta-base-emotion	38
4.14 Clasificación de los tweets utilizando cardiffnlp/twitter-roberta-base-emotion	39
4.15 Carga del modelo "michellejeli/emotion_text_classifier"	40
4.16 Función para obtener el sentimiento de tweets utilizando el modelo michellejeli/emotion_text_classifier	41
4.17 Clasificación de los tweets utilizando michellejeli/emotion_text_classifier	41
5.1 Muestra de los 25000 tweets	42
5.2 Distribución de Sentimientos con TextBlob	43
5.3 Distribución de Sentimientos con VADER	44
5.4 Distribución de Sentimientos con Pattern	45
5.5 Distribución de Sentimientos con Flair	45
5.6 Distribución de Sentimientos con cardiffnlp	46
5.7 Distribución de Sentimientos con michellejeli	47



# Capítulo 1.

## Introducción

### 1.1 Motivación

En las últimas décadas, las redes sociales han revolucionado la forma en que las personas interactúan entre sí y se comunican en línea. Desde el primer lanzamiento de Friendster en 2002 hasta la llegada de Facebook en 2004 y la explosión de Twitter en 2006, las redes sociales han crecido exponencialmente y se han convertido en una parte fundamental de la vida diaria de muchas personas. Hoy en día, millones de usuarios utilizan las redes sociales para compartir información, mantenerse en contacto con amigos y familiares, realizar negocios y mucho más.

A medida que ha aumentado la presencia de las redes sociales, se ha vuelto cada vez más importante para las empresas, organizaciones y otros grupos comprender la opinión pública sobre sus marcas, productos o servicios. Las personas comparten sus opiniones públicamente en línea y estas opiniones pueden afectar significativamente la reputación de una empresa o marca. Por lo tanto, comprender la opinión pública en las redes sociales es una parte fundamental de la gestión de marca en la era digital.

El análisis de sentimientos es una técnica utilizada para extraer y analizar el tono emocional detrás de un texto. Este análisis de sentimientos puede ser aplicado a los mensajes que se publican en línea en redes sociales. Esto se lleva a cabo mediante técnicas de procesamiento de lenguaje natural y aprendizaje automático, las cuales identifican y clasifican los sentimientos detrás de los mensajes en línea. El análisis de sentimientos se ha convertido en una herramienta crucial para comprender los sentimientos de los usuarios y clientes en línea y ha demostrado ser útil en una amplia variedad de aplicaciones, desde la detección de la satisfacción del cliente hasta la prevención de crisis.

En este trabajo, examinamos en detalle el análisis de sentimientos en las redes sociales. Comenzaremos describiendo los diferentes enfoques que se pueden usar para el análisis, el uso de modelos previamente entrenados y los enfoques basados en reglas. Luego, evaluamos los desafíos y las limitaciones del análisis de sentimientos en las redes sociales, incluida la presencia de jerga, faltas de ortografía y emoticonos, así como la presencia de ironía y sarcasmo. Además, discutimos cómo el análisis de sentimientos en las redes sociales es probabilístico y puede tener un margen de error.

Después de explorar las técnicas y los desafíos del análisis de sentimientos en las redes sociales, examinamos el impacto de esta técnica en la toma de decisiones. El análisis de sentimientos en las redes sociales tiene un impacto significativo en la toma de decisiones, ya que proporciona una comprensión en tiempo real de las percepciones y opinión pública sobre temas importantes. Por lo tanto, es importante saber que existen empresas y organizaciones que pueden utilizar el análisis de sentimientos de las redes sociales para tomar decisiones estratégicas e informadas.

## 1.2 Metodología de desarrollo

Para la realización de este trabajo se ha seguido la metodología de desarrollo iterativa e incremental.

Cada iteración es un período de tiempo corto (en torno a 1-4 semanas) en los que se planifica, se diseña, se desarrolla y se prueba una funcionalidad, en nuestro caso, cada iteración ha consistido en abordar cada uno de los objetivos detallados en la Sección XYZ.

## 1.3 Estructura de la memoria

Las tareas que se han llevado a cabo en el trabajo son:

- Estudio del estado del arte: Es un término utilizado para referirse al estado actual, en el presente, de los avances, técnicas y tecnologías que se emplean en una rama del conocimiento como es, en este caso, el Análisis de Sentimientos. Es necesario realizar este estudio como primera fase de este proyecto.
- Aprendizaje de las Tecnologías que se emplearán en el desarrollo: En concreto se usa el lenguaje de programación Python y librerías desarrolladas por terceros para el procesamiento de lenguaje natural y en concreto para el análisis de sentimientos. También se estudiarán metodologías y plataformas para el desarrollo de software.
- Desarrollo: En primer lugar, se utiliza la API de Twitter para obtener la información necesaria para el análisis. La API de Twitter proporciona acceso a los datos de Twitter, como tweets y seguidores, y permite a los desarrolladores recuperar estos datos mediante solicitudes HTTP. Una vez que se recopilan datos sobre tweets, se realiza un preprocesamiento para limpiar y preparar los datos para su análisis. Después del preprocesamiento, se clasifican los datos en función de su contenido y su sentimiento.
- Resultados y conclusiones: En cuanto a los resultados y conclusiones, el análisis que utiliza las librerías de clasificación de sentimientos permite obtener una visión

más completa y precisa sobre el contenido y sentimiento de los datos recopilados a través de la API de Twitter y el preprocesamiento de los mismos. Se concluye que la elección de la librería a utilizar depende de los objetivos específicos del análisis y las características de los datos.

## 1.4 Objetivos del estudio

En este trabajo, se explora el uso de técnicas de análisis de sentimientos para extraer información valiosa de las redes sociales. El objetivo principal del estudio es evaluar la eficacia de diferentes técnicas de modelado del lenguaje y clasificación de texto para determinar la polaridad del sentimiento en las publicaciones de las redes sociales. Los objetivos del estudio son:

- Evaluar las diferentes posibilidades para extraer información procedente de redes sociales.
- Preprocesar la información descargada para que sea susceptible de ser analizada y clasificada.
- Analizar la eficacia de las técnicas de modelado del lenguaje por métodos de aprendizaje automático y procesamiento de lenguaje natural.
- Contrastar la eficacia de las diferentes técnicas de clasificación de texto mediante aprendizaje automático.
- Clasificar el texto según su polaridad, es decir, determinar si el texto es positivo, negativo o neutro en cuanto a su sentimiento o actitud y en cuanto a su subjetividad se refiere a la medida en que el texto está basado en opiniones o creencias personales en lugar de hechos objetivos.
- Discutir los resultados y proponer líneas futuras de trabajo.

A través de este trabajo, se espera obtener una mayor comprensión de las técnicas de análisis de sentimientos y su aplicación en el contexto de las redes sociales.

## 1.5 Plan de Trabajo

En este apartado se exponen las tareas relacionadas con la gestión de proyectos, ya que la correcta gestión de esfuerzos es una tarea crítica y determinante en el desarrollo de cualquier proyecto, y en consecuencia, en el cumplimiento de sus objetivos.

El proyecto se desarrolla como Trabajo de Fin de Grado en el Grado de Ingeniería Mecánica de la Universidad de Málaga. Tiene una carga de 12 créditos que equivalen a 300 horas de trabajo. El proyecto, aunque se acordó un tiempo antes, se considera que comienza el 13 de febrero con una finalización del 5 de junio de 2023. Si tenemos en cuenta este período, consta de aproximadamente 16 semanas, durante las cuales se dedican de 8 a 12 horas de trabajo por semana..

# Capítulo 2.

## Estudio del estado

### 2.1 Estudio del estado del arte

El análisis de sentimientos en redes sociales es un campo de investigación en constante evolución, debido al creciente uso de las redes sociales y la necesidad de comprender mejor cómo se relacionan los usuarios con ellas. El estudio del estado del arte en este campo se centró en identificar los avances más recientes en cuanto a técnicas y herramientas de análisis de sentimientos en redes sociales, así como en la identificación de las líneas de investigación futuras más prometedoras.

Entre los avances más significativos en el análisis de sentimientos en redes sociales se encuentran el desarrollo de algoritmos de aprendizaje profundo (*deep learning*), que permiten una mayor precisión en la identificación de la polaridad de los textos; la aplicación de procesamiento de lenguaje natural, que permiten una mejor comprensión del contexto y la detección de más sutiles emociones; y la integración de métodos de análisis de redes sociales, que permiten una mejor comprensión de las relaciones sociales e influyen en la propagación de sentimientos en la red.

Asimismo, el estudio del estado del arte del análisis de sentimientos en redes sociales ha identificado varias líneas de investigación prometedoras. Estas áreas incluyen la integración de múltiples modalidades de datos como imágenes y videos, la identificación de sentimientos en diferentes idiomas y culturas, la detección de emociones más complejas y sutiles, y la aplicación de técnicas de análisis de sentimientos en redes sociales en diferentes campos. como la política, la salud y el marketing.

Además, el estudio de estado del arte también identificó algunos desafíos y limitaciones en el análisis de sentimientos en las redes sociales, como la necesidad de abordar el tema de la subjetividad, la falta de datos etiquetados y la dificultad de generalizar los resultados a diferentes contextos.

En conclusión, el estudio del estado del arte del análisis de sentimientos en redes sociales ha permitido identificar importantes avances y áreas de investigación futura en este campo en constante evolución. Esto puede ayudar a los investigadores y profesionales a comprender mejor cómo interactúan los usuarios con las redes sociales, así como a mejorar la precisión y la eficacia de las herramientas y técnicas de análisis de sentimientos de las redes sociales.

## 2.2 Preprocesado

El preprocesamiento en el análisis de sentimientos es un paso importante que implica procesar y limpiar datos textuales antes de aplicar técnicas de análisis de sentimientos.

El preprocesamiento puede incluir las siguientes técnicas:

- Tokenización: dividir el texto en palabras o tokens individuales.
- Eliminación de puntuación: elimine los signos de puntuación como comas, puntos y signos de interrogación.
- Eliminación de palabras vacías: Elimina palabras comunes que no tienen un alto valor semántico, como "él", "ella" y "de".
- Stemming y lematización: reducir las palabras a su forma básica o raíz, como "comer" o "comiendo" se convierte en "come".
- Eliminación de caracteres especiales y números: Elimina caracteres especiales y números que no brindan información útil para el análisis de sentimientos.

El objetivo del preprocesamiento es convertir el texto en una forma que pueda ser analizada y procesada por algoritmos de análisis de sentimientos. Además, el procesamiento previo también puede ayudar a reducir el ruido y aumentar la precisión del análisis de sentimientos.

## 2.3 Clasificación

Para llevar a cabo la clasificación se pueden utilizar diferentes técnicas, por un lado, un acercamiento basado en *Machine Learning* (Aprendizaje automático) o basados en lexicones. A continuación se ilustra una representación en árbol de cómo se pueden catalogar estos diferentes clasificadores que más adelante se van a desarrollar los que se han utilizado en este trabajo.

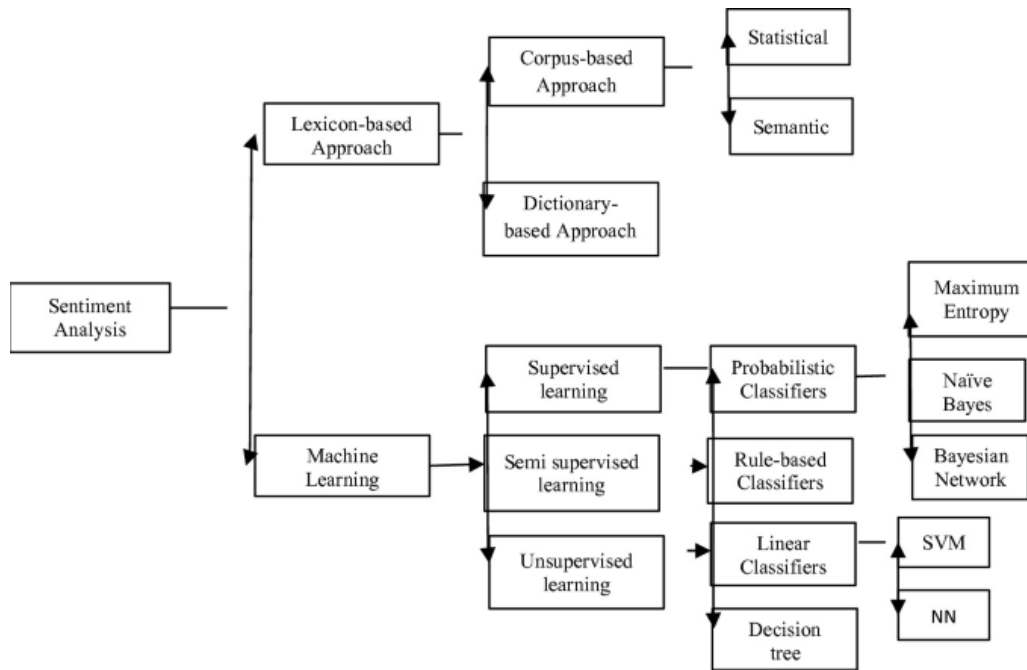


Figura 2.1: Clasificación de algoritmos de análisis de sentimientos

### 2.3.1 Procesamiento de lenguaje natural

El procesamiento del lenguaje natural (NLP) es una técnica clave en el análisis de sentimientos de última generación. Se utiliza para analizar y comprender el lenguaje humano en una amplia gama de aplicaciones, incluido el análisis de sentimientos de las redes sociales.

El NLP se basa en identificar patrones y relaciones entre las palabras de un texto para determinar su polaridad y tono emocional. Esta técnica utiliza varios métodos para analizar el lenguaje, como la tokenización, que consiste en dividir el texto en palabras y oraciones, y el etiquetado de partes del discurso, que identifica las partes gramaticales de cada palabra.

Además, el NLP también utiliza técnicas como el análisis sintáctico y el análisis semántico para comprender la estructura y el significado de las oraciones en un texto. Por ejemplo, el análisis sintáctico ayuda a identificar la relación entre las palabras de una oración, mientras que el análisis semántico se centra en comprender el significado de las palabras y su contexto.

En el análisis de sentimientos, el NLP es esencial para identificar términos clave en un texto y determinar su polaridad. Esto se logra comparando las palabras encontradas con un diccionario de sentimientos, que asocia las palabras con su polaridad positiva, negativa o neutra.

### 2.3.1.1 Diccionario

Un clasificador de análisis de sentimientos que utiliza un diccionario es un enfoque en el que se crea un diccionario de palabras con una polaridad específica (por ejemplo, positiva, negativa o neutral) y se utiliza para determinar la polaridad de un texto en función de la frecuencia y el contexto de las palabras que aparecen en él. Este enfoque se basa en la idea de que ciertas palabras tienen una connotación emocional positiva o negativa, y que la presencia y frecuencia de estas palabras en un texto pueden indicar la polaridad general del mismo. Los clasificadores de análisis de sentimientos que utilizan diccionarios a menudo se basan en técnicas de procesamiento de lenguaje natural (NLP) y pueden ser útiles para analizar grandes cantidades de texto de manera rápida y eficiente.

### 2.3.2 Aprendizaje automático

El análisis de sentimientos en *machine learning* es una técnica de procesamiento de lenguaje natural que utiliza algoritmos de aprendizaje automático para clasificar el tono emocional de un texto. Este enfoque implica la extracción de características del texto, como las palabras clave, la estructura de la oración, la entonación y el contexto, para entrenar un modelo de clasificación de sentimientos.

El modelo de aprendizaje automático se entrena con un conjunto de datos etiquetados previamente, donde cada texto está asociado con una etiqueta de sentimiento (positivo, negativo o neutral). A partir de este conjunto de datos de entrenamiento, el modelo aprende a identificar patrones en los datos que le permiten predecir el sentimiento de nuevos textos que no han sido etiquetados previamente.

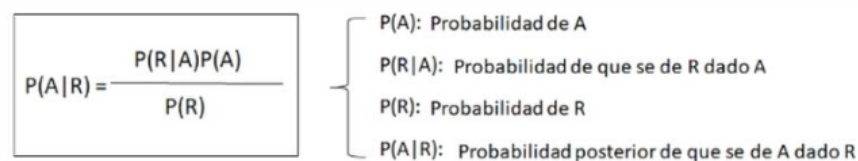
Una vez que el modelo ha sido entrenado, se puede utilizar para analizar grandes cantidades de datos no etiquetados, como publicaciones de redes sociales, reseñas de productos y comentarios de clientes. Al aplicar el modelo entrenado a estos nuevos datos, se pueden identificar patrones en los sentimientos que pueden ser útiles para la toma de decisiones empresariales o para la investigación de mercado. El análisis de sentimientos de los machine learning se utiliza ampliamente en la industria para analizar la opinión del cliente, la satisfacción del usuario y la percepción de la marca.

En los siguientes apartados presentamos las técnicas y algoritmos que hemos usado para hacer análisis de sentimientos.

### 2.3.2.1 Naive Bayes

La clasificación por Naive Bayes es un algoritmo de aprendizaje automático supervisado que se utiliza en una variedad de aplicaciones, como la clasificación de texto, la detección de spam y la detección de sentimientos. Es un algoritmo popular debido a su simplicidad y eficiencia, y a menudo funciona bien incluso en conjuntos de datos pequeños.

El Naive Bayes se basa en el teorema de Bayes, que establece que la probabilidad de que ocurra un evento A dado que ocurre un evento B es igual a la probabilidad de que ocurra B dado que ocurre A multiplicada por la probabilidad de que ocurra A, dividida por la probabilidad de que ocurra B.



The diagram shows the Naive Bayes formula in a box on the left:  $P(A|R) = \frac{P(R|A)P(A)}{P(R)}$ . To the right of the box is a list of definitions for the variables in the formula, enclosed in a large right-facing curly bracket. The definitions are: P(A): Probabilidad de A; P(R|A): Probabilidad de que se de R dado A; P(R): Probabilidad de R; and P(A|R): Probabilidad posterior de que se de A dado R.

Figura 2.2: *Fórmula de Naive Bayes*

En el contexto de la clasificación, el teorema de Bayes se puede utilizar para calcular la probabilidad de que un documento pertenezca a una categoría dada, dada la presencia de ciertas características en el documento.

El "naive" en Naive Bayes se refiere a la suposición de independencia condicional, lo que significa que se asume que cada característica es independiente de las demás, dada la categoría. A pesar de esta suposición simplificada, Naive Bayes a menudo funciona bien en la práctica.

El proceso de clasificación por Naive Bayes implica primero entrenar un modelo con un conjunto de datos de entrenamiento que contiene documentos previamente etiquetados con sus categorías correspondientes. Durante el entrenamiento, el modelo utiliza el teorema de Bayes para calcular la probabilidad de que cada característica aparezca en cada categoría. Estas probabilidades se utilizan para calcular la probabilidad de que un documento pertenezca a cada categoría posible, dadas las características del documento.

Una vez que el modelo ha sido entrenado, se puede utilizar para clasificar nuevos documentos. Para clasificar un nuevo documento, el modelo calcula la probabilidad de que el documento pertenezca a cada categoría posible, dadas las características del documento. El documento se clasifica en la categoría con la probabilidad más alta.

Las fortalezas de la clasificación por Naive Bayes incluyen su simplicidad y eficiencia, su capacidad para funcionar bien en conjuntos de datos pequeños y su capacidad para trabajar con características categóricas y numéricas. Sin embargo, las suposiciones simplificadas de independencia condicional pueden ser

problemáticas en algunos casos, y el rendimiento del modelo puede verse afectado por la presencia de características irrelevantes o correlacionadas.

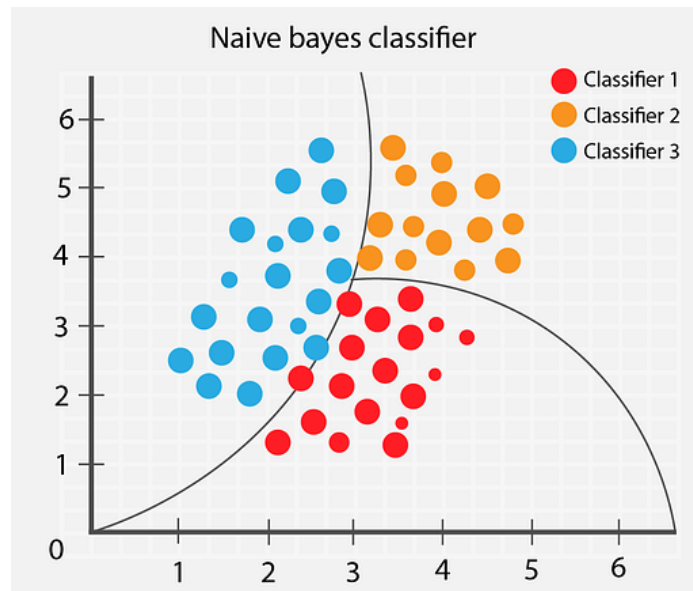


Figura 2.3: Representación del clasificador Naive Bayes

### 2.3.2.2 Redes Neuronales

Una red neuronal es un modelo matemático inspirado en el funcionamiento del cerebro humano, que utiliza múltiples capas de unidades de procesamiento interconectadas para realizar una tarea específica. En el caso del análisis de sentimientos, una red neuronal se entrena utilizando grandes conjuntos de datos de textos etiquetados para aprender a identificar patrones y características importantes en los textos que indican polaridad y tono emocional.

Una de las principales ventajas de las redes neuronales es su capacidad para identificar patrones complejos y no lineales en los datos, lo que las hace especialmente efectivas en el análisis de sentimientos. Las redes neuronales pueden procesar grandes volúmenes de datos en paralelo y adaptarse automáticamente a las características del texto para lograr una alta precisión en la clasificación de la polaridad.

Las redes neuronales pueden ser entrenadas utilizando diferentes arquitecturas, como las redes neuronales convolucionales (CNN) y las redes neuronales recurrentes (RNN). Las CNN son adecuadas para el análisis de sentimientos en textos cortos, como los mensajes de Twitter, mientras que las RNN son útiles para el

análisis de sentimientos en textos más largos, como reseñas de productos o artículos de noticias.

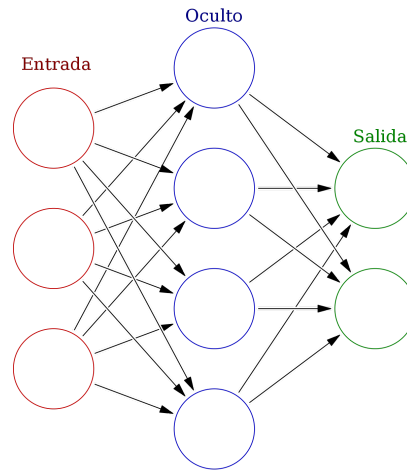


Figura 2.4: Representación de la arquitectura de red neuronal

### 2.3.2.3 SVM

El clasificador SVM (*Support Vector Machine*) es un modelo de aprendizaje automático supervisado que se utiliza en problemas de clasificación y regresión. En el caso de la clasificación de sentimientos, SVM se utiliza para predecir la polaridad de un texto (positivo, negativo o neutral) en función de las características del texto.

SVM se basa en la idea de encontrar un hiperplano que separe dos clases de puntos en un espacio de características de alta dimensión. El hiperplano se define como el límite de decisión óptimo que maximiza la distancia entre los puntos de dos clases cercanas, también conocidos como vectores de soporte. El objetivo de SVM es maximizar el margen entre el hiperplano y los vectores de soporte.

En el caso de la clasificación de sentimientos, SVM utiliza como entrada una matriz de características, donde cada fila representa un texto y cada columna representa una característica del texto, como la frecuencia de una palabra específica, la longitud del texto o la presencia de ciertos patrones lingüísticos. La matriz de características se utiliza para entrenar el clasificador SVM, que busca aprender una función que pueda predecir la polaridad del texto de entrada.

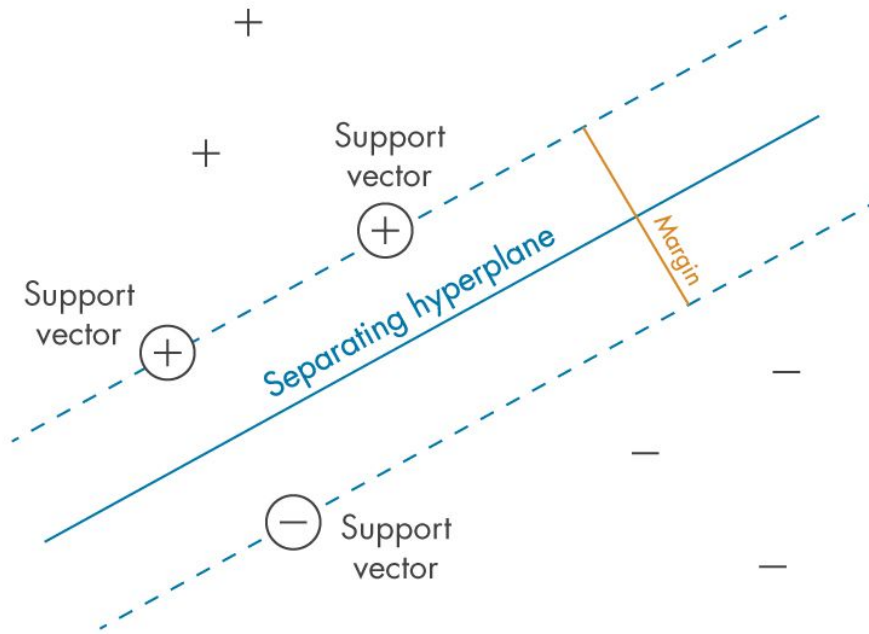


Figura 2.5: Representación del clasificador SVM

# Capítulo 3.

## Tecnologías empleadas en el desarrollo

### 3.1 Introducción

El procesamiento de lenguaje natural y el aprendizaje automático son dos áreas que han sido fundamentales para el análisis de sentimientos aunque su relación puede parecer muy estrecha. En esencia, el lenguaje natural (NLP) es un campo de la informática que se centra en la comprensión y la generación de lenguaje humano, mientras que el aprendizaje automático es un subcampo de la inteligencia artificial que se centra en el desarrollo de algoritmos y modelos que pueden aprender a partir de datos y hacer predicciones.

En particular, Python y las librerías de terceros desarrolladas para el procesamiento de lenguaje natural han sido muy utilizadas en la implementación de algoritmos para el análisis de sentimientos. En este trabajo, se presentan las tecnologías que se emplearán en el desarrollo de un análisis de sentimientos basado en Python y se estudiarán las metodologías y plataformas para el desarrollo de software.

### 3.2 Python y el procesamiento de lenguaje natural

Python es un lenguaje de programación que ha ganado mucha popularidad en el ámbito del procesamiento de lenguaje natural (NLP) debido a su sintaxis simple y fácil de entender. Además, ofrece una gran cantidad de bibliotecas y herramientas de terceros que se pueden utilizar para el procesamiento de texto y análisis de sentimientos.

Python cuenta con librerías y herramientas específicas para el procesamiento de lenguaje natural, como TextBlob, Flair, Vader, Pattern entre otros. Estas bibliotecas proporcionan herramientas para el preprocesamiento de texto, análisis de sentimientos, etiquetado de partes del discurso, extracción de entidades, análisis sintáctico, entre otras funcionalidades que son esenciales en el procesamiento de lenguaje natural.

### 3.3 Librerías de procesamiento de lenguaje natural en Python

A continuación, se presentan las librerías de procesamiento de lenguaje natural en Python que se han utilizado para el análisis de sentimientos debido a que son algunas de las más populares y ampliamente utilizadas en la comunidad de procesamiento de lenguaje natural y aprendizaje automático en Python.

- Tweepy: Es una librería de Python que facilita el acceso a la API de Twitter, permitiendo hacer búsquedas y extraer datos de esta red social. Tweepy se encarga de la autenticación y autorización necesarias para acceder a la API, y provee métodos para hacer búsquedas y obtener información de tweets, usuarios, listas, etc. *(Documentación de Tweepy)*
- Pandas: Es una librería de Python que se utiliza para el análisis y manipulación de datos. Pandas ofrece estructuras de datos como Series y Data Frames, que permiten trabajar con datos tabulares y realizar diversas operaciones como filtrado, agregación, unión, etc. *(Documentación de Pandas)*
- Numpy: Es una librería de Python que se utiliza para realizar operaciones matemáticas y numéricas. Numpy ofrece estructuras de datos como arreglos y matrices, que permiten trabajar con datos numéricos y realizar operaciones como multiplicación de matrices, cálculo de estadísticas, etc. *(Documentación de NumPy)*
- re: es una librería de Python que se utiliza para trabajar con expresiones regulares. Las expresiones regulares son patrones que se utilizan para buscar y manipular texto de forma eficiente. La librería provee métodos para compilar y buscar patrones en texto. *(Documentación de re)*
- Matplotlib: Es una librería de Python para la creación de gráficos y visualizaciones de datos. Proporciona una gran variedad de gráficos, como histogramas, gráficos de líneas y de barras, y gráficos de dispersión. Es esencial para la visualización de datos en el análisis de sentimientos y otras tareas de análisis de datos. *(Documentación de Matplotlib)*
- Emoji: La librería emoji es un módulo de Python que permite trabajar con emojis. Proporciona una manera fácil de encontrar, imprimir y reemplazar emojis en cadenas de texto. Además, también puede convertir emojis en su forma de texto y viceversa. *(Documentación de emoji)*
- Matplotlib: Permite la creación de gráficos de líneas, gráficos de dispersión, histogramas, diagramas de barras, gráficos de contorno, gráficos de superficie y muchas otras visualizaciones. Matplotlib es altamente personalizable y ofrece un control preciso sobre todos los aspectos de la visualización, incluidos los ejes, etiquetas, colores, estilos y tamaños. Además, se puede integrar con otras bibliotecas de Python, como NumPy y Pandas, para trabajar fácilmente con datos estructurados. *(Documentación de Matplotlib)*

Aparte de estas librerías comunes para el procesamiento de lenguaje natural y aprendizaje automático en Python se han utilizado modelos de librerías de procesamiento de lenguaje natural:

- TextBlob utiliza un clasificador basado en el modelo de aprendizaje supervisado Naive Bayes. Este clasificador se entrena con un conjunto de datos etiquetados previamente (en el caso del análisis de sentimiento, con textos clasificados como positivos, negativos o neutrales) y luego utiliza la probabilidad condicional de cada palabra en el texto para asignar una polaridad al mismo. *(Documentación de TextBlob)*
- VADER (Valence Aware Dictionary and Sentiment Reasoner) es una librería de procesamiento de lenguaje natural (NLP) diseñada específicamente para el análisis de sentimientos en texto. Fue desarrollada por el equipo de investigación de la Universidad de Georgia y utiliza un conjunto de reglas lingüísticas y un diccionario de palabras que asigna puntuaciones de polaridad a las palabras en el texto. VADER es muy popular debido a su facilidad de uso y a su precisión en la identificación de la polaridad de un texto. Además, es capaz de manejar adecuadamente el lenguaje coloquial y las emociones expresadas en el texto, lo que lo hace especialmente útil para el análisis de sentimientos en redes sociales y otras plataformas en línea. *(Documentación de VADER)*
- Pattern es una biblioteca de procesamiento de lenguaje natural (NLP) que utiliza una variedad de técnicas, incluyendo aprendizaje automático y lexicones, para realizar tareas como análisis de sentimientos, etiquetado de partes del discurso, extracción de información y más. Si bien Pattern utiliza redes neuronales en algunas de sus funcionalidades, no se limita a ellas y utiliza una amplia gama de técnicas de NLP. En particular, en este TFG se usa la función de análisis de sentimientos sentiment de la librería Pattern. El módulo sentiment de Pattern utiliza un enfoque híbrido para la clasificación de sentimientos, que combina un diccionario léxico con reglas heurísticas con un modelo de aprendizaje automático basado en SVM para la clasificación de sentimientos. *(Documentación de Pattern)*
- Flair es una biblioteca de procesamiento de lenguaje natural de código abierto que utiliza redes neuronales para tareas de lenguaje natural, como etiquetado de partes del discurso, reconocimiento de entidades nombradas, análisis de sentimientos, clasificación de texto y más. Flair permite el uso de modelos de lenguaje pre-entrenados o entrenados personalizados y ofrece una interfaz de usuario fácil de usar para interactuar con los modelos. También es conocido por su capacidad para manejar idiomas diferentes al inglés, lo que lo convierte en una opción popular para aplicaciones de lenguaje natural multilingües. *(Documentación de Flair)*

- Transformers es una biblioteca de procesamiento de lenguaje natural (NLP) de código abierto que se enfoca en el uso de modelos basados en transformadores, una arquitectura de redes neuronales, para una variedad de tareas en NLP. Esta biblioteca fue desarrollada por Hugging Face y se ha vuelto muy popular en la comunidad de NLP debido a su amplia colección de modelos pre-entrenados y su fácil implementación.

La biblioteca Transformers permite utilizar modelos pre-entrenados de vanguardia, como BERT, GPT, RoBERTa, entre otros, para realizar tareas como clasificación de texto, generación de texto, traducción automática, extracción de información, análisis de sentimientos y más. Estos modelos han sido entrenados en grandes conjuntos de datos y capturan patrones lingüísticos complejos, lo que les permite ofrecer un rendimiento sobresaliente en diversas tareas de procesamiento de lenguaje natural. (*Documentación de Transformers*)

### 3.4 Plataformas para el desarrollo de software

En el desarrollo de cualquier proyecto de software, es importante tener en cuenta la plataforma en la que se llevará a cabo el desarrollo, ya que esto puede influir en la elección de herramientas, librerías y lenguajes de programación a utilizar. En el caso del análisis de sentimientos, se recomienda utilizar una plataforma que permita trabajar con lenguajes de programación como Python y que ofrezca soporte para el procesamiento de lenguaje natural.

Una de las plataformas más populares para el desarrollo de software en Python es Google Colab ya que es una plataforma de desarrollo de software basada en la nube que permite a los usuarios escribir, ejecutar y colaborar en código Python en un entorno de Jupyter Notebook. La plataforma es proporcionada por Google y está diseñada para hacer que el desarrollo y la colaboración de proyectos de aprendizaje automático sean más accesibles y eficientes para los usuarios.

Entre las características principales de Google Colab se encuentran la capacidad de acceder a recursos de computación en la nube de Google de forma gratuita, incluyendo CPUs, GPUs y TPUs para acelerar el entrenamiento de modelos de aprendizaje profundo, así como la integración con Google Drive y GitHub para el almacenamiento y la colaboración de proyectos.

Además, Google Colab incluye una gran variedad de librerías preinstaladas para el procesamiento de datos, el aprendizaje automático y la visualización de datos, lo que hace que sea más fácil para los usuarios comenzar a trabajar en proyectos de forma inmediata sin tener que preocuparse por la instalación de las librerías necesarias.

# Capítulo 4.

## Desarrollo

### 4.1 Lectura de la información mediante la API de Twitter

API es una abreviatura de Application Programming Interfaces. Una API es un mecanismo que permite a los componentes software comunicarse entre sí mediante un conjunto de definiciones y protocolos. Twitter ofrece una API con un amplio conjunto de funcionalidades que brindan un amplio acceso a los datos públicos de la red social.

Para acceder a la API, lo primero que se debe hacer es registrar una aplicación que se destine a la extracción de un conjunto de tweets que servirán como base para nuestros análisis. Esto se puede hacer desde la página web [developer.twitter.com/en/portal/projects](https://developer.twitter.com/en/portal/projects), donde después de proporcionar la información necesaria, se te darán unas claves de autenticación que te permitirán acceder a los servicios de la API.

Las claves son: clave del consumidor, clave secreta del consumidor, token de acceso y token de acceso secreto. Una vez que hayamos completado esta parte, la extracción de los datos la realizaremos mediante Tweepy, una librería de Python que nos permite acceder a la API de Twitter.

Lo primero que debemos hacer es configurar nuestra cuenta para poder acceder a la API correctamente:

1. Autenticarnos mediante OAuthHandler, proporcionando la clave y clave secreta del consumidor (consumer key y consumer secret).
2. Proporcionar nuestro token de acceso para poder trabajar con la API mediante el método set access token, que recibirá nuestro token de acceso y su clave (access token y access token secret).
3. Una vez que hayamos hecho la configuración bajo el nombre de auth, crearemos una instancia de la clase "API" a la que le pasaremos la configuración como parámetro y con la que comenzaremos a trabajar.

Para extraer tweets, es necesario usar un objeto llamado cursor que nos permite recorrer diferentes colecciones de tweets, como por ejemplo los timelines o listas de usuarios, entre otros. El cursor tiene varios parámetros que se pueden configurar, pero el más importante es indicarle qué tipo de tweets queremos buscar, es decir,

cuál es el criterio de búsqueda específico que vamos a utilizar. Por ejemplo, si queremos buscar tweets que contengan una palabra clave determinada, configuraremos el cursor para que busque todos los tweets que incluyan esa palabra.

Esto se logrará pasando como argumento el método `api.search`, donde le indicamos que queremos tweets que contengan el hashtag que queramos, en este caso que contengan la palabra España sin contar los retweets (`q='España -filter:retweets'`) que estén escritos en español (`lang='es'`) y que se solicitan hasta 100 tweets que cumplan con los criterios de búsqueda especificados (`count=100`).

Nuestro iterador recoge objetos "Tweet" que contienen muchos atributos como pueden ser el texto, su identificador, el identificador del usuario, la fecha en la que se escribió, los hashtags incluidos, etc. Queremos recoger simplemente el texto escrito, es decir, el "Tweet" propiamente dicho. Para almacenar esta información, se utiliza un DataFrame, una estructura de datos bidimensional comúnmente empleada en el análisis de datos y programación en Python. El DataFrame se asemeja a una tabla de base de datos o a una hoja de cálculo de Excel, donde los datos se organizan en filas y columnas. Cada columna en un DataFrame representa una variable o característica, mientras que cada fila representa una observación o instancia.

## 4.2 Preprocesamiento de los datos

Una vez obtenido el conjunto de datos que se va a utilizar, es recomendable examinarlo detenidamente antes de iniciar el proceso de preprocesamiento, con el fin de obtener una comprensión general de cómo está estructurado y qué información contiene.

```
1)@https_peritazzz sostiene conversaciones con una inteligencia artificial*
2)Diccionario para ser un experto en inteligencia artificial: guía rápida con lo
3)🤖 "Videos creados a partir de diferentes herramientas tecnológicas de inteli
4)#InteligenciaArtificial en la #porcicultura.
👉 La inteligencia artificial ha ganado una gran atención para resolver... https://
5)Varios empleados de Google pidieron detener la inteligencia artificial Bard ht
```

Figura 4.1: *Conjunto inicial de los datos*

Al analizar los datos que hemos extraído de las redes sociales, especialmente de Twitter, nos encontramos con características particulares que requieren un enfoque

específico, como las menciones, los hashtags, las palabras mal escritas y los emojis. Es importante tener en cuenta que el objetivo de esta etapa es preparar adecuadamente los datos para que los algoritmos de extracción de características puedan procesarlos de manera efectiva.

En esta fase del proceso, utilizamos expresiones regulares, que son patrones específicos que nos ayudan a identificar y modificar ciertos elementos en el texto, como menciones, hashtags, palabras mal escritas o emojis. Para llevar a cabo esta tarea, usamos el módulo `'re'` de Python que está diseñado para trabajar con expresiones regulares. La función `'re.sub()'` es especialmente útil, ya que nos permite reemplazar los patrones que encontramos en el texto con otro texto o simplemente eliminarlos. En resumen, los pasos que seguimos son identificar los patrones que deseamos modificar, y luego utilizar la función `'re.sub()'` para hacer los cambios necesarios en el texto.

A continuación, se describe el proceso de preprocesamiento más común y esencial utilizado para limpiar un tweet. Cada paso tiene una razón específica para su aplicación, ayudando a eliminar el ruido y mejorar la calidad de los datos para el análisis de sentimientos.

- Eliminación de URLs:

```
text=re.sub('((www\.[^\s]+)|(https?://[^\s]+))', '', text)
```

En ocasiones, un tweet puede contener un enlace a una página web externa. Sin embargo, estos enlaces no son útiles para nuestro análisis, por lo que debemos eliminarlos. Para hacerlo, solo consideraremos aquellos enlaces que comienzan con "www.", "https://" o "http://" y terminan en el primer espacio que se encuentre.

- Eliminación de saltos de línea:

```
text=re.sub("\n", "", text)
```

Podemos eliminar los saltos de línea que se codifican como `\n` ya que pueden causar problemas en el análisis de los datos.

- Eliminación de menciones:

```
text=re.sub('@\s','@',text)  
text=re.sub('(@[^\s]+)|(w/)', '', text)
```

En Twitter, una mención es cuando se incluye el nombre de usuario de otra persona precedido por el símbolo `@` en un tweet, lo que notifica a esa

persona. Es importante tener en cuenta que los tweets pueden contener menciones mal escritas, como aquellas con un espacio entre el símbolo @ y el nombre de usuario, o aquellas que utilizan "w/" en lugar de @ para abreviar la palabra "with". Por lo tanto, es necesario prestar atención a estas menciones para su correcto tratamiento en el preprocesamiento de datos.

- Eliminación de Hashtags:

```
text = re.sub(r'#([\s]+)', r'1', text)
```

Podemos identificar los hashtags como cadenas de caracteres precedidas por el símbolo # y seguidas de caracteres que no sean un espacio en blanco. En las redes sociales, los hashtags se usan para etiquetar temas y permitir que los usuarios encuentren y participen en conversaciones relacionadas. En nuestro preprocesamiento de datos, eliminaremos solamente el símbolo # ya que la información contenida en el hashtag es valiosa.

- Conversión de mayúsculas a minúsculas

```
text = text.lower()
```

Este paso es esencial para considerar dos palabras como idénticas aunque tengan diferencias en las letras mayúsculas o minúsculas.

- Eliminación de emojis:

```
text = emoji.demojize(text)  
text = re.sub('_', ' ', text)
```

La eliminación de emojis en el análisis de sentimientos de tweets puede ser útil para reducir el ruido en los datos y evitar que los emojis influyan en la detección del sentimiento del texto. En algunos casos, los emojis pueden ser utilizados para transmitir sentimientos, pero en otros casos, pueden ser utilizados simplemente para añadir énfasis o humor sin cambiar significativamente el sentimiento general del tweet. Al eliminar los emojis, se puede mejorar la precisión del análisis de sentimientos y se pueden obtener resultados más confiables.

- Eliminación de caracteres no alfanuméricos:

```
text = re.sub('[^\w\s#@]', "", text)
```

Eliminar los caracteres no alfanuméricos es una técnica comúnmente utilizada para limpiar los datos y prepararlos para el análisis de sentimientos

en los tweets. Los caracteres no alfanuméricos, como los signos de puntuación o los símbolos, no suelen contener información relevante sobre el sentimiento que se desea analizar y pueden distorsionar los resultados del análisis.

Además, la eliminación de estos caracteres ayuda a reducir el ruido en los datos, lo que hace que sea más fácil identificar las palabras clave que expresan la emoción del tweet. Esto mejora la precisión del análisis de sentimientos y ayuda a obtener resultados más significativos.

## 4.3 Clasificación de los datos

La clasificación de datos es una tarea fundamental en el análisis de sentimientos. Se utiliza para asignar una etiqueta de polaridad (positiva, negativa o neutral) a un texto determinado, como puede ser un tweet. Esta tarea se ha vuelto cada vez más importante en la era de las redes sociales, donde las empresas y las organizaciones pueden monitorear la opinión pública y la satisfacción del cliente en tiempo real.

Existen diferentes librerías con diferentes algoritmos que permiten realizar esta tarea, en nuestro caso hemos escogido algunas de las más populares.

### 4.3.1 TextBlob

TextBlob es una buena opción para el análisis de sentimientos en Twitter por varias razones:

1. Fácil de usar: TextBlob es fácil de implementar y tiene una sintaxis sencilla que permite a los usuarios realizar el análisis de sentimientos en pocos pasos.
2. Alto rendimiento: TextBlob está implementado en Python, lo que le permite aprovechar el alto rendimiento de este lenguaje de programación. Además, utiliza técnicas de procesamiento de lenguaje natural (NLP) avanzadas para proporcionar resultados precisos y fiables.
3. Preprocesamiento de texto integrado: TextBlob tiene una serie de herramientas integradas para preprocesar el texto, como la eliminación de stopwords, la lematización y el etiquetado POS (part-of-speech), lo que ayuda a mejorar la precisión de los resultados.

4. Idiomas múltiples: TextBlob soporta varios idiomas, lo que lo hace adecuado para su uso en diferentes contextos y regiones.

Se va a comenzar con la extracción de la subjetividad y la polaridad de los tweets dados.

```
def get_subjectivity(text): #obtención de la subjetividad
    return TextBlob(text).sentiment.subjectivity

def get_polarity (text): #obtención de la polaridad
    return TextBlob(text).sentiment.polarity
```

Figura 4.2: *Funciones para obtener la subjetividad y polaridad de un texto utilizando TextBlob*

La función *get\_subjectivity* obtiene la subjetividad de un texto utilizando la librería TextBlob. En la escala de subjetividad utilizada por TextBlob, el valor 0 representa un texto completamente objetivo, mientras que el valor 1 representa un texto completamente subjetivo, basado únicamente en opiniones o sentimientos personales.

La función *get\_polarity* obtiene la polaridad de un texto utilizando la librería TextBlob. La polaridad es un valor entre -1 y 1 que indica el sentimiento general del texto, donde -1 indica sentimiento negativo, 0 indica sentimiento neutral y 1 indica sentimiento positivo

Se van a añadir dos nuevas columnas al DataFrame "df" que contiene tweets, llamadas "Subjectivity" y "Polarity". Estas columnas se van a rellenar utilizando las funciones *get\_subjectivity* y *get\_polarity*, respectivamente, que calculan la subjetividad y la polaridad de cada tweet en la columna "Tweets". Para aplicar estas funciones a cada tweet en la columna "Tweets", se utiliza la función *apply*. Al utilizar *df.head()*, se mostrarán las primeras filas del DataFrame actualizado con las nuevas columnas de subjetividad y polaridad. En resumen, las columnas de subjetividad y polaridad indicarán en qué medida cada tweet es subjetivo y en qué medida expresa una opinión positiva o negativa.

```
df['Subjectivity'] = df['Tweets'].apply(get_subjectivity) #generar el dataframe
df['Polarity'] = df['Tweets'].apply(get_polarity)

df.head()
```

Figura 4.3: Generación de un DataFrame con polaridad y subjetividad de tweets utilizando TextBlob

El siguiente paso es clasificar cada tweet en el DataFrame como negativo, neutral o positivo según su puntaje de polaridad.

```
def analysis(score):
    if score < 0:
        return 'Negative'
    elif score == 0:
        return 'Neutral'
    else:
        return 'Positive'

df['Analysis'] = df['Polarity'].apply(analysis)

df
```

Figura 4.4: Clasificación de polaridad de tweets en negativo, neutral o positivo utilizando TextBlob

Y obtendremos cada uno de los tweets clasificados por la subjetividad, la polaridad y el análisis.

	Tweets	Subjectivity	Polarity	Analysis
0	mobilegpt al rescate asistente personal de ia ...	0.300000	0.000000	Neutral
1	tiktok está trabajando en una nueva herramient...	0.000000	0.000000	Neutral
2	las cosas con la inteligencia artificial avanz...	1.000000	-0.600000	Negative
3	yuval noah harari y la inteligencia artificial...	1.000000	-0.600000	Negative
4	cómo aplicar la inteligencia artificial en las...	1.000000	-0.600000	Negative
...	...	...	...	...
95	inteligencia artificial y su impacto social y ...	0.388889	-0.155556	Negative
96	curso de inteligencia artificial con wordpress...	1.000000	-0.600000	Negative
97	te llama la atención la inteligencia artificia...	1.000000	-0.600000	Negative
98	rompí la inteligencia artificial esa transcrib...	1.000000	-0.600000	Negative
99	es falso fue generado por inteligencia artifi...	1.000000	-0.600000	Negative

Figura 4.5: Clasificación de subjetividad, polaridad y análisis utilizando TextBlob

### 4.3.2 Vader

Existen varios motivos para elegir Vader como herramienta para analizar sentimientos en Twitter, entre ellos:

1. Diseñado específicamente para redes sociales: Vader fue desarrollado por investigadores que buscaban una herramienta para analizar los sentimientos en redes sociales, por lo que fue diseñado específicamente para este fin.
2. Alto rendimiento: Vader es una herramienta muy rápida y eficiente en el procesamiento de grandes volúmenes de datos, lo que la hace especialmente útil para analizar grandes cantidades de tweets en tiempo real.
3. Capacidad de manejar jerga y emoticonos: Vader está diseñado para ser capaz de comprender el lenguaje coloquial, incluyendo jerga y emoticonos, lo que es esencial para analizar los sentimientos en Twitter, donde el uso de este tipo de lenguaje es muy común.
4. Alta precisión en el análisis de sentimientos: Según estudios comparativos, Vader tiene una alta precisión en la clasificación de sentimientos en comparación con otras herramientas, lo que la convierte en una opción confiable para el análisis de sentimientos en Twitter.

Se va a comenzar con la extracción de la subjetividad y la polaridad de los tweets dados.

```
def get_subjectivity(text):
    sia = SentimentIntensityAnalyzer()
    sentiment_dict = sia.polarity_scores(text)
    return sentiment_dict['compound']

def get_polarity(text):
    sia = SentimentIntensityAnalyzer()
    sentiment_dict = sia.polarity_scores(text)
    if sentiment_dict['compound'] > 0:
        return 1
    elif sentiment_dict['compound'] < 0:
        return -1
    else:
        return 0
```

Figura 4.6: *Funciones para obtener la subjetividad y polaridad de un texto utilizando Vader*

La función `get_subjectivity` utiliza la función `polarity_scores` de la librería VADER para obtener un diccionario con las puntuaciones de polaridad del texto, es decir, el grado de positividad o negatividad del texto. Luego, la función retorna el valor de la clave 'compound', que es la puntuación compuesta que representa el sentimiento general del texto.

La función `get_polarity` también utiliza la función `polarity_scores` de la librería VADER para obtener el diccionario con las puntuaciones de polaridad del texto. En este caso, la función evalúa la puntuación compuesta y retorna 1 si es positiva, -1 si es negativa y 0 si es neutral.

Al igual que antes se van a crear dos nuevas columnas "Subjectivity" y "Polarity" en un DataFrame para luego la clasificación de cada tweet como negativo, neutral o positivo según su puntaje de polaridad.

	Tweets	Subjectivity	Polarity	Analysis
0	mobilegpt al rescate asistente personal de ia ...	0.000	0	Neutral
1	tiktok está trabajando en una nueva herramient...	0.000	0	Neutral
2	las cosas con la inteligencia artificial avanz...	0.000	0	Neutral
3	yuval noah harari y la inteligencia artificial...	-0.296	-1	Negative
4	cómo aplicar la inteligencia artificial en las...	0.000	0	Neutral
...	...	...	...	...
95	inteligencia artificial y su impacto social y ...	0.000	0	Neutral
96	curso de inteligencia artificial con wordpress...	0.000	0	Neutral
97	te llama la atención la inteligencia artificia...	0.000	0	Neutral
98	rompí la inteligencia artificial esa transcrib...	0.000	0	Neutral
99	es falso fue generado por inteligencia artifi...	0.000	0	Neutral

Figura 4.7: Clasificación de subjetividad, polaridad y análisis utilizando Vader

### 4.3.3 Pattern

Algunos de los motivos por los que se elige Pattern para analizar sentimientos en Twitter son:

1. Facilidad de uso: Pattern es una librería muy fácil de usar y no requiere conocimientos avanzados de programación para su implementación.

2. Idiomas múltiples: Pattern soporta varios idiomas, lo que permite analizar sentimientos en diferentes idiomas, incluyendo inglés, español, francés, alemán, italiano, portugués, holandés, entre otros.
3. Open source: Pattern es una librería de código abierto y gratuita, lo que la hace accesible para cualquier persona interesada en utilizarla.
4. Buen rendimiento: Pattern ha demostrado un buen rendimiento en el análisis de sentimientos en diferentes estudios comparativos, lo que la hace una opción confiable para analizar sentimientos en Twitter.

Al igual que antes se va a comenzar con la extracción de la subjetividad y la polaridad de los tweets dados.

```
def get_subjectivity(text):
    sentiment_dict = sentiment(text)
    return sentiment_dict[1]

def get_polarity(text):
    sentiment_dict = sentiment(text)
    if sentiment_dict[0] > 0:
        return 1
    elif sentiment_dict[0] < 0:
        return -1
    else:
        return 0
```

Figura 4.8: *Funciones para obtener la subjetividad y polaridad de un texto utilizando Pattern*

La función `get_subjectivity(text)` utiliza la librería `sentiment` importada desde `vader` para calcular la subjetividad de un texto dado, en este caso un tweet. La función devuelve el valor de subjetividad que se encuentra en el índice 1 del diccionario `sentiment_dict`, que se obtiene al aplicar la función `sentiment` al texto.

Por otro lado, la función `get_polarity(text)` también utiliza la librería `sentiment` para calcular la polaridad de un texto dado. La función devuelve un valor numérico que representa la polaridad del texto, el cual se determina comparando el valor de polaridad que se encuentra en el índice 0 del diccionario `sentiment_dict` con 0. Si el valor es mayor que 0, se considera como positivo y se devuelve 1; si es menor que 0, se considera como negativo y se devuelve -1; y si es igual a 0, se considera como neutral y se devuelve 0.

Al igual que antes se van a crear dos nuevas columnas "Subjectivity" y "Polarity" en un DataFrame para luego la clasificación de cada tweet como negativo, neutral o positivo según su puntaje de polaridad.

	Tweets	Subjectivity	Polarity	Analysis
0	mobilegpt al rescate asistente personal de ia ...	0.300000	0	Neutral
1	tiktok está trabajando en una nueva herramient...	0.000000	0	Neutral
2	las cosas con la inteligencia artificial avanz...	1.000000	-1	Negative
3	yuval noah harari y la inteligencia artificial...	1.000000	-1	Negative
4	cómo aplicar la inteligencia artificial en las...	1.000000	-1	Negative
...	...	...	...	...
95	inteligencia artificial y su impacto social y ...	0.388889	-1	Negative
96	curso de inteligencia artificial con wordpress...	1.000000	-1	Negative
97	te llama la atención la inteligencia artificia...	1.000000	-1	Negative
98	rompí la inteligencia artificial esa transcrib...	1.000000	-1	Negative
99	es falso fue generado por inteligencia artifi...	1.000000	-1	Negative

Figura 4.9: Clasificación de subjetividad, polaridad y análisis utilizando Pattern

### 4.3.4 Flair

Los motivos por los que se han elegido Flair para analizar sentimientos en Twitter son:

1. Mejor rendimiento en tareas de procesamiento del lenguaje natural, como la detección de sentimientos, en comparación con otras bibliotecas de Python como TextBlob y NLTK.
2. Soporte para varios idiomas, lo que permite el análisis de sentimientos en diferentes idiomas, lo que es útil para empresas y organizaciones que operan en diferentes países.
3. Integración con otras herramientas de procesamiento del lenguaje natural, como Word Embeddings y Transformers, que permiten el análisis de sentimientos más avanzado y preciso.

En general, Flair es una biblioteca de procesamiento del lenguaje natural moderna y poderosa que puede ser útil para el análisis de sentimientos en Twitter y otras tareas relacionadas con el procesamiento del lenguaje natural.

Al igual que en los pasos anteriores, se procederá con la extracción de la polaridad de los tweets, ya que la subjetividad no puede ser calculada con Flair ya que es una biblioteca de procesamiento de lenguaje natural que se enfoca en el análisis del lenguaje a nivel de sentencia, palabra y carácter sin ser capaz de proporcionar una medida subjetiva.

```
def get_polarity(text):
    sentence = Sentence(text)
    classifier.predict(sentence)
    # Obtenemos la polaridad de la etiqueta con mayor score
    label = sentence.labels[0]
    if label.value == 'NEGATIVE':
        return -1 * label.score
    else:
        return label.score
```

Figura 4.10: Función para obtener la polaridad de un texto utilizando Flair

La función `get_polarity(text)` utiliza el modelo de análisis de sentimientos de Flair para clasificar el sentimiento del texto pasado como argumento. Primero, crea un objeto `Sentence` a partir del texto y luego lo procesa a través del modelo de clasificación utilizando la función `predict()`. Luego, se obtiene la polaridad del sentimiento a través de la etiqueta con el puntaje más alto. Si la etiqueta es "NEGATIVE", se devuelve el puntaje multiplicado por -1 para indicar una polaridad negativa, de lo contrario se devuelve el puntaje.

En este caso se va a crear una columna "Polarity" en un DataFrame para luego la clasificación de cada tweet como negativo, neutral o positivo según su puntaje de polaridad.

	Tweets	Polarity	Analysis
0	mobilegpt al rescate asistente personal de ia ...	0.964245	Positive
1	tiktok está trabajando en una nueva herramient...	0.960351	Positive
2	las cosas con la inteligencia artificial avanz...	-0.957466	Negative
3	yuval noah harari y la inteligencia artificial...	0.988587	Positive
4	cómo aplicar la inteligencia artificial en las...	-0.859012	Negative
...	...	...	...
95	inteligencia artificial y su impacto social y ...	-0.896796	Negative
96	curso de inteligencia artificial con wordpress...	-0.997239	Negative
97	te llama la atención la inteligencia artificia...	0.599413	Positive
98	rompí la inteligencia artificial esa transcrib...	-0.998025	Negative
99	es falso fue generado por inteligencia artifi...	-0.996028	Negative

Figura 4.11: Clasificación de la polaridad y análisis utilizando Flair

## 4.3.5 Transformers

En este proyecto, se van a utilizar dos modelos pre-entrenados de la biblioteca Transformers. Los modelos pre-entrenados son redes neuronales que han sido entrenadas en grandes conjuntos de datos para capturar patrones lingüísticos y comprender el lenguaje de manera general. Estos modelos pre-entrenados están disponibles para su uso y pueden aplicarse a una variedad de tareas de procesamiento de lenguaje natural (NLP) sin requerir un entrenamiento adicional.

### 4.3.5.1 Modelo "cardiffnlp/twitter-roberta-base-emotion"

Hay varios motivos para elegir el modelo "cardiffnlp/twitter-roberta-base-emotion" para analizar tweets de Twitter:

1. Específico para tweets: Este modelo ha sido entrenado específicamente en un gran conjunto de datos de tweets, lo que lo hace más adecuado para comprender el lenguaje y las emociones expresadas en este tipo de texto corto y con características propias de Twitter.
2. Clasificación de emociones: El modelo está diseñado para clasificar emociones en textos cortos. Puede identificar emociones como felicidad, tristeza, enojo, miedo, sorpresa y disgusto en los tweets, lo que proporciona una comprensión más profunda del contenido emocional presente en los mensajes.
3. Arquitectura Roberta: El modelo se basa en la arquitectura Roberta, que es una variante de los modelos de lenguaje basados en Transformer. Esta arquitectura ha demostrado un rendimiento sobresaliente en una amplia gama de tareas de procesamiento de lenguaje natural, lo que garantiza una capacidad efectiva para capturar los patrones lingüísticos y emocionales en los tweets.
4. Pre-entrenado: El modelo se encuentra pre-entrenado en una gran cantidad de datos, lo que significa que ya ha aprendido representaciones de palabras y estructuras lingüísticas comunes. Esto le permite comprender mejor el contexto y proporcionar predicciones más precisas sobre las emociones expresadas en los tweets.
5. Amplia disponibilidad: El modelo "cardiffnlp/twitter-roberta-base-emotion" está disponible en la biblioteca Transformers, que es una de las bibliotecas más populares y ampliamente utilizadas para el procesamiento de lenguaje natural y el aprendizaje automático en Python.

Se va a comenzar con la carga del modelo y con la búsqueda con la palabra clave.

```

# Cargar modelo de clasificación de sentimientos
classifier = pipeline("text-classification", model="cardiffnlp/twitter-roberta-base-emotion")

# Realizar búsqueda de tweets con la palabra clave
hashtag = api.search_tweets(q='Artificial intelligence', lang='en', count=100)

```

Figura 4.12: Carga del modelo "cardiffnlp/twitter-roberta-base-emotion"

El modelo "cardiffnlp/twitter-roberta-base-emotion" se utiliza en inglés porque ha sido entrenado específicamente en datos en inglés para realizar la clasificación de emociones en texto. Esto significa que el modelo ha aprendido a reconocer patrones y características del lenguaje inglés para asignar las emociones adecuadas a los textos.

Al igual que el resto de librerías mencionadas anteriormente, este modelo también requiere un preprocesamiento de los tweets antes de ser alimentados al modelo. Sin embargo, a diferencia de los modelos anteriores, no es necesario eliminar los emojis, ya que este modelo es capaz de analizarlos y tenerlos en cuenta en la clasificación de emociones.

Siguiendo los pasos anteriores, ahora procederemos a extraer la polaridad de los tweets utilizando este modelo.

```

results = []
for index, row in df.iterrows():
    texto_tweet = row['Tweets']
    sentimiento = classifier(texto_tweet)[0]['label']
    results.append({'Tweet': texto_tweet, 'Sentimiento': sentimiento})

```

Figura 4.13: Función para obtener el sentimiento de tweets utilizando el modelo cardiffnlp/twitter-roberta-base-emotion

El código recorre cada tweet en un dataframe y clasifica su sentimiento utilizando el modelo pre-entrenado. Luego, almacena los resultados de la clasificación en una lista de diccionarios. Cada diccionario contiene el texto del tweet y su respectiva clasificación de sentimiento.

El código seguidamente imprime el dataframe que contiene los tweets y sus clasificaciones de sentimientos.

	Tweet	Sentimiento
0	when you consider that the trillions of doll...	anger
1	my suggestion to anc necthe anc while busy thi...	anger
2	metatechnews provides the latest updates ins...	optimism
3	my top 10 ai related films 1 2001 a space od...	optimism
4	digital analytics and aificial intelligence ...	anger
..	...	...
81	jamesvgingerich sam babic hyland svp and cio s...	optimism
82	this is real hype cody ai is an advanced a...	optimism
83	openai the creators of chatgpt have authored...	anger
84	hej and dont know if aificial intelligence...	joy
85	does she have any intelligence herself real o...	joy

```
[86 rows x 2 columns]
anger      39
optimism  26
sadness    11
joy        10
```

Figura 4.14: *Clasificación de los tweets utilizando cardiffnlp/twitter-roberta-base-emotion*

#### 4.3.5.2 Modelo "michellejeli/emotion\_text\_classifier"

Hay varios motivos por los cuales se podría elegir el modelo "michellejeli/emotion\_text\_classifier" para analizar tweets de Twitter:

1. Específicamente entrenado para analizar emociones en texto: Este modelo ha sido entrenado específicamente para clasificar emociones en texto, lo cual lo hace adecuado para analizar los diferentes tipos de emociones expresadas en los tweets.
2. Precisión en la clasificación de emociones: El modelo ha sido entrenado en un conjunto de datos amplio y diverso, lo que le permite tener un alto nivel de precisión en la clasificación de emociones en texto. Esto es importante para obtener resultados confiables y significativos al analizar los sentimientos expresados en los tweets.
3. Disponibilidad de etiquetas emocionales relevantes: El modelo proporciona etiquetas emocionales relevantes, lo que significa que puede identificar y clasificar una variedad de emociones específicas, como felicidad, tristeza, enojo, miedo, sorpresa y disgusto. Esto permite un análisis más detallado y completo de los sentimientos presentes en los tweets.
4. Compatibilidad con el procesamiento de texto en varios idiomas: El modelo puede manejar textos en diferentes idiomas, lo que es beneficioso para analizar tweets de Twitter que pueden estar escritos en diferentes idiomas.
5. Facilidad de uso y accesibilidad: La integración del modelo en la biblioteca de Transformers proporciona una interfaz sencilla y fácil de usar para utilizar el modelo en aplicaciones de procesamiento de lenguaje natural. Esto facilita su implementación y uso en el análisis de los tweets de Twitter.

6. Rendimiento en el contexto de redes sociales: Dado que este modelo ha sido entrenado específicamente con datos de Twitter, es probable que tenga un buen rendimiento al analizar los tweets de esta plataforma. Los tweets suelen tener características particulares, como una longitud limitada, lenguaje informal, uso de emoticonos y hashtags, entre otros. El hecho de que el modelo haya sido entrenado con estos datos específicos puede mejorar su capacidad para comprender y clasificar las emociones expresadas en los tweets de manera más precisa y adecuada al contexto de las redes sociales.

Al igual que con el modelo anterior se va a comenzar con la carga del modelo y con la búsqueda con la palabra clave.

```
# Cargar modelo de clasificación de sentimientos
classifier = pipeline("sentiment-analysis", model="michellejeli/emotion_text_classifier")

# Realizar búsqueda de tweets con la palabra clave
hashtag = api.search_tweets(q='Artificial intelligence', lang='en', count=100)
```

Figura 4.15: Carga del modelo "michellejeli/emotion\_text\_classifier"

El modelo "michellejeli/emotion\_text\_classifier" tiene la capacidad de ser utilizado en otros idiomas además del inglés. A diferencia del modelo "cardiffnlp/twitter-roberta-base-emotion" que está específicamente entrenado en datos en inglés, el modelo "michellejeli/emotion\_text\_classifier" puede ser aplicado a textos en diferentes idiomas para clasificar las emociones.

Esto se debe a que el modelo ha sido entrenado en un conjunto de datos más diverso que incluye múltiples idiomas, lo que le permite capturar patrones y características de lenguaje comunes a través de diferentes idiomas. Por lo tanto, si se desea utilizar este modelo para analizar tweets en un idioma distinto al inglés, es posible obtener resultados de clasificación de emociones precisos.

Sin embargo, es importante tener en cuenta que el rendimiento del modelo puede variar según el idioma, ya que su desempeño óptimo se logra en el idioma en el que fue entrenado originalmente (que en este caso es el inglés). Por lo tanto, al utilizar el modelo en otros idiomas, es posible que se observen diferencias en la precisión de la clasificación de emociones.

Al igual que el resto de las librerías mencionadas anteriormente, este modelo requiere un preprocesamiento de los tweets antes de ser alimentados al modelo. Sin embargo, a diferencia de otros modelos, no es necesario eliminar los emojis, ya que el modelo "michellejeli/emotion\_text\_classifier" es capaz de analizarlos y tenerlos en cuenta en la clasificación de emociones. Esto permite que los emojis, que son una forma común de expresión emocional en los tweets, influyan en la clasificación de emociones realizada por el modelo.

Siguiendo los pasos anteriores, ahora procederemos a extraer la polaridad de los tweets utilizando este modelo.

```

results = []
for index, row in df.iterrows():
    texto_tweet = row['Tweets']
    sentimiento = classifier(texto_tweet)[0]['label']
    results.append({'Tweet': texto_tweet, 'Sentimiento': sentimiento})

```

Figura 4.16: *Función para obtener el sentimiento de tweets utilizando el modelo michellejeli/emotion\_text\_classifier*

Se crea una lista vacía llamada "results" para almacenar los resultados de clasificación de sentimientos de cada tweet. Luego, se itera sobre cada fila del dataframe, obteniendo el texto del tweet y clasificando su sentimiento utilizando el modelo "classifier". El resultado de la clasificación se guarda en un diccionario junto con el texto del tweet y se agrega a la lista "results". Al finalizar, la lista "results" contiene los diccionarios que representan cada tweet y su clasificación de sentimientos.

El código seguidamente imprime el dataframe que contiene los tweets y sus clasificaciones de sentimiento.

	Tweet	Sentimiento
0	dear god please say this is a parody uh o...	fear
1	aificial intelligence vs emotional intellige...	neutral
2	aificial intelligencethere is a reason why a...	neutral
3	hieloni wrote a screenplay that u might find...	neutral
4	new ideas around multiagent systems palaificia...	neutral
..	...	...
76	what could possibly go wrong here why not appo...	neutral
77	aificial intelligence model could help diagnos...	neutral
78	a frightening future is aificial intelligence ...	fear
79	google magi a whole new google search engine	neutral
80	hes 100 aificialbut there is zero intell...	sadness

```

[81 rows x 2 columns]
neutral    67
fear       5
joy        3
sadness    3
anger      2
surprise   1

```

Figura 4.17: *Clasificación de los tweets utilizando michellejeli/emotion\_text\_classifier*

# Capítulo 5.

## Evaluación

En este trabajo se ha llevado a cabo un análisis de sentimientos en redes sociales, en particular en Twitter, y se han evaluado varios algoritmos para determinar su efectividad en la clasificación de sentimientos.

El conjunto de datos utilizado para evaluar los algoritmos es conocido como "Sentiment140" y es ampliamente utilizado en la comunidad de procesamiento de lenguaje natural y análisis de sentimientos. Este conjunto de datos consiste en 1.6 millones de tweets en inglés que han sido etiquetados con sentimientos conocidos.

El dataset Sentiment140, también conocido como el dataset de Twitter, clasifica los tweets en dos categorías principales: positivos y negativos. La clasificación de los tweets se realiza utilizando una escala de polaridad, donde los tweets positivos se etiquetan como "4" y los tweets negativos se etiquetan como "0".

Es importante destacar que el dataset Sentiment140 no incluye una categoría neutral. Esto se debe a que el enfoque principal del conjunto de datos se centra en capturar la polaridad positiva y negativa de los tweets, sin considerar una categoría intermedia de neutralidad.

Se ha escogido una muestra aleatoria de 25,000 tweets (de los cuales 12,500 han sido clasificados como positivos y los restantes 12,500 como negativos) de este conjunto de datos para limitar el tamaño del conjunto de datos y facilitar el procesamiento. Esta muestra nos permite obtener una visión representativa del conjunto de datos completo sin tener que lidiar con la totalidad de los 1.6 millones de registros. Al fijar una semilla (en este caso, la semilla 42) para la generación de la muestra aleatoria, aseguramos que obtengamos la misma muestra en cada ejecución, lo que nos permite obtener resultados consistentes y comparables.

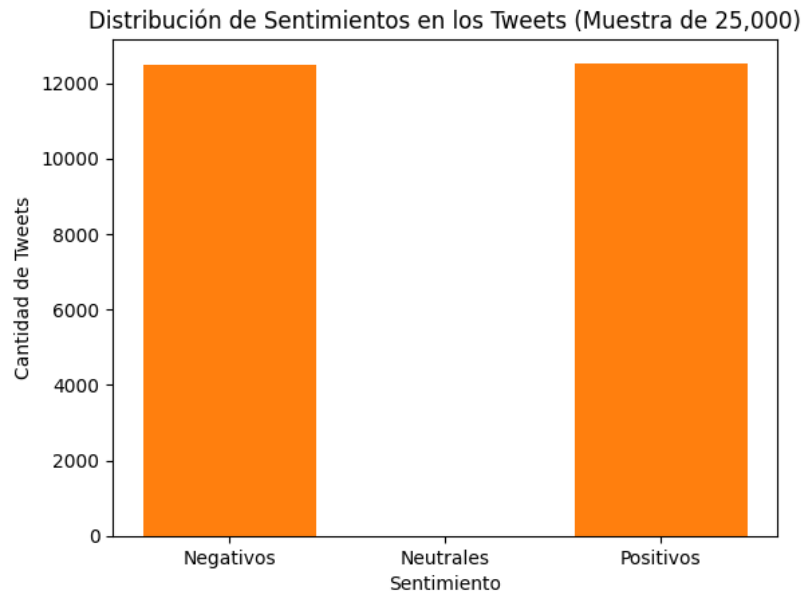


Figura 5.1: *Muestra de los 25000 tweets*

En esta evaluación de los algoritmos de análisis de sentimientos, utilizaremos dos métricas principales: el accuracy como medida de la precisión de los algoritmos, y el tiempo de ejecución como medida del rendimiento.

El accuracy es una métrica comúnmente utilizada para evaluar la precisión de los modelos de clasificación. Representa la proporción de predicciones correctas realizadas por un algoritmo con respecto al total de predicciones realizadas, y por tanto viene dado por un número en el rango  $[0,1]$ . En el contexto del análisis de sentimientos, el accuracy nos indicará qué tan bien los algoritmos son capaces de clasificar correctamente los tweets en las categorías de sentimiento: negativo, neutral y positivo. Un accuracy más alto indica una mayor capacidad del algoritmo para clasificar correctamente los sentimientos.

El tiempo de ejecución se refiere al tiempo necesario para que un algoritmo complete su proceso de clasificación en un conjunto de datos determinado. Medir el tiempo de ejecución nos permite evaluar la eficiencia computacional de los algoritmos. En este caso, estaremos interesados en comparar el tiempo de ejecución de los diferentes algoritmos para determinar cuál de ellos es más rápido en realizar las clasificaciones de sentimientos.

Al evaluar tanto el accuracy como el tiempo de ejecución, obtendremos una visión integral de la eficacia y eficiencia de los algoritmos de análisis de sentimientos. El accuracy nos indicará qué tan bien clasifican los algoritmos, mientras que el tiempo de ejecución nos ayudará a identificar aquellos algoritmos que logran un equilibrio entre precisión y velocidad de procesamiento.

Se iniciará la evaluación con el algoritmo TextBlob, conocido por su simplicidad y eficiencia. Durante la ejecución, se ha observado un tiempo promedio de 31

segundos, lo que demuestra su capacidad para procesar grandes volúmenes de datos de manera rápida. En cuanto a su rendimiento, se ha obtenido un nivel de precisión (accuracy) de 0.43992, lo cual indica que el algoritmo tiene un desempeño deficiente en el análisis de sentimientos en redes sociales. Sin embargo, es importante tener en cuenta que este algoritmo se basa en un enfoque basado en reglas, lo que puede limitar su capacidad para captar matices y contextos más complejos.

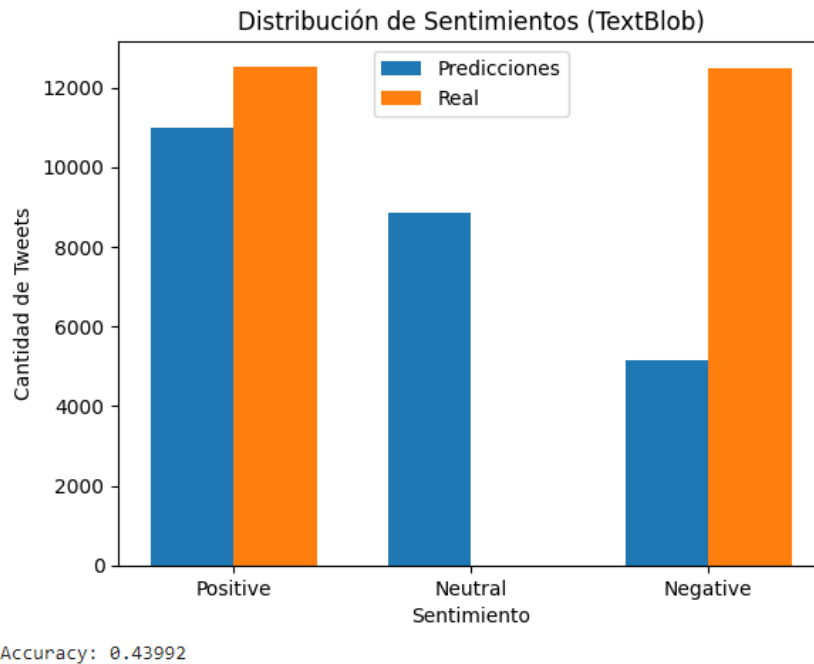


Figura 5.2: *Distribución de Sentimientos con TextBlob*

En el siguiente paso de nuestra evaluación, se analizó el algoritmo VADER, ampliamente utilizado en el análisis de sentimientos en redes sociales. Durante el análisis de los 25,000 tweets, se registró un tiempo de procesamiento de aproximadamente 5 minutos al ejecutar este algoritmo. Aunque su implementación puede requerir un mayor tiempo computacional en comparación con otros enfoques, los resultados obtenidos mostraron mejoras significativas en comparación con el enfoque anterior. El modelo logró alcanzar una precisión (accuracy) de 0.52536 en la clasificación de sentimientos. A pesar de este avance, es importante destacar que esta precisión todavía se considera relativamente baja y existe margen para mejoras adicionales en términos de desempeño y rendimiento. VADER se basa en léxicos y reglas predefinidas para determinar la polaridad de los sentimientos, lo que le permite capturar ciertos matices y contextos en el análisis de sentimientos en redes sociales.

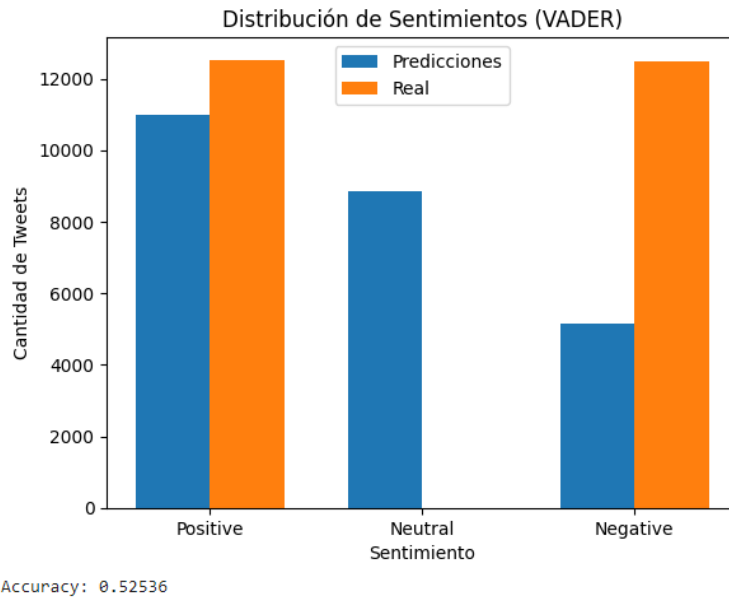


Figura 5.3: *Distribución de Sentimientos con VADER*

Continuando con nuestra evaluación, se ha procedido analizar el algoritmo Pattern en el contexto del análisis de sentimientos en redes sociales. Durante la ejecución de este algoritmo, se registró un tiempo de procesamiento de aproximadamente 30 segundos. Aunque Pattern requiere menos tiempo en comparación con otros enfoques, los resultados obtenidos revelaron una precisión (accuracy) de 0.44184 en la clasificación de sentimientos. Esto indica que Pattern tiene la capacidad de capturar la polaridad de los sentimientos expresados en los textos analizados, aunque su rendimiento puede variar dependiendo del contexto y las características específicas del conjunto de datos. Este algoritmo combina reglas y aprendizaje automático, lo que le permite adaptarse a diferentes dominios y contextos de análisis de sentimientos en redes sociales.

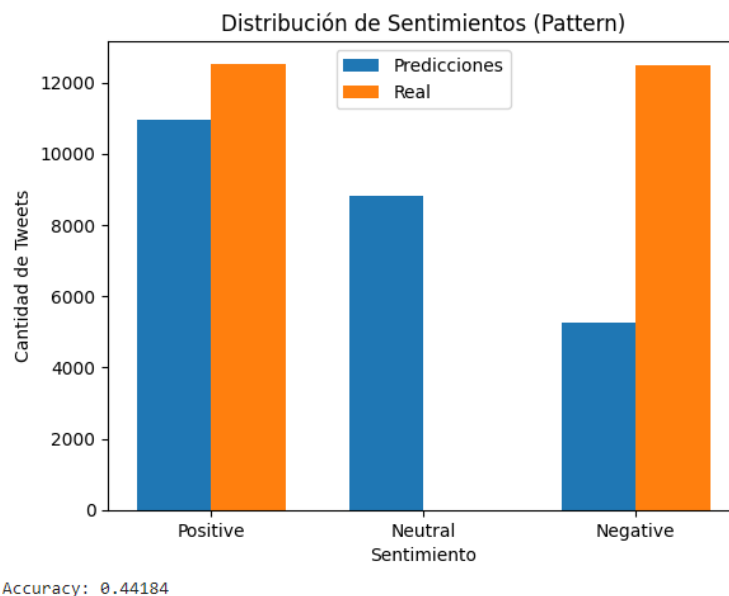


Figura 5.4: *Distribución de Sentimientos con Pattern*

Durante la ejecución del algoritmo Flair, registramos un tiempo de procesamiento de aproximadamente 39 minutos. Aunque Flair requiere más tiempo en comparación con otros enfoques, los resultados obtenidos revelan una precisión (accuracy) de 0.69412 en la clasificación de sentimientos. Esto indica que Flair demuestra una mayor capacidad para capturar la polaridad y los matices emocionales en los textos analizados, lo que lo convierte en una opción prometedora para el análisis de sentimientos en redes sociales. Flair utiliza redes neuronales para el análisis de sentimientos, lo que le permite capturar relaciones complejas y comprender el contexto en el que se expresan los sentimientos.

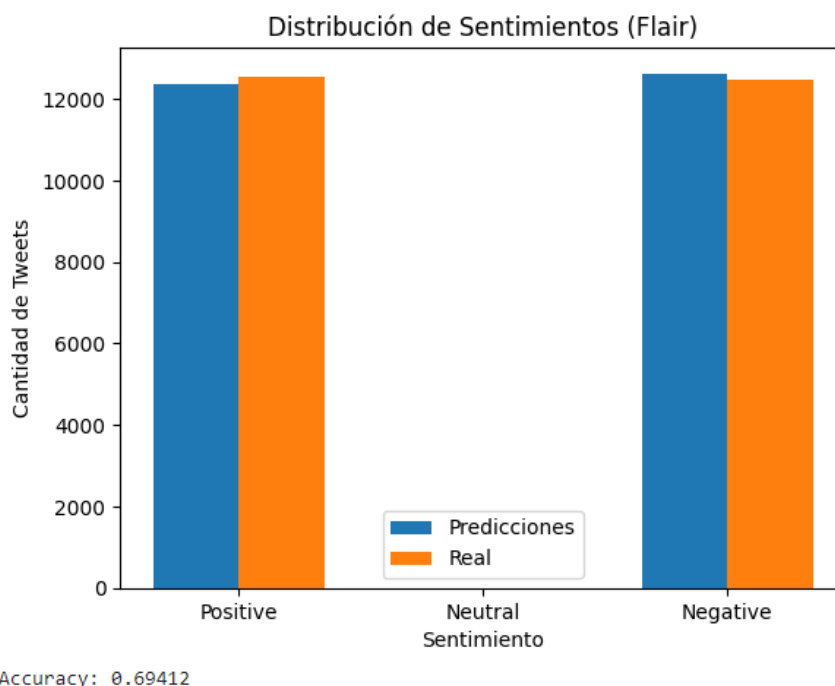


Figura 5.5: *Distribución de Sentimientos con Flair*

En los siguientes algoritmos pre-entrenados, se ha realizado un mapeo de las etiquetas predichas a las etiquetas objetivo con el fin de compararlas directamente con el conjunto de datos Sentiment140. Esto nos permite agrupar las etiquetas predichas en tres categorías principales: Negativo, Neutral y Positivo.

Para lograr esto, creamos un diccionario llamado "label\_mapping" que asocia las etiquetas predichas por los algoritmos a las etiquetas objetivo. En este diccionario, hemos establecido las siguientes correspondencias:

- Etiquetas Negativas: incluyen las emociones de enojo (anger) y tristeza (sadness).
- Etiquetas Neutrales: incluyen las emociones de neutralidad (neutral) y ausencia de emoción (no emotion).
- Etiquetas Positivas: incluyen las emociones de alegría (joy) y optimismo (optimism).

Luego, se ha aplicado este mapeo a las etiquetas predichas del conjunto de datos utilizando la función "apply" en la columna 'predicted\_sentiment'. Mediante una función lambda, buscamos la correspondencia de cada etiqueta predicha en el diccionario "label\_mapping" y asignamos la etiqueta objetivo correspondiente. De esta manera, logramos mapear las etiquetas predichas por los algoritmos pre-entrenados a las etiquetas objetivo del conjunto de datos Sentiment140. Esto permite realizar una comparación directa y evaluar la precisión de los algoritmos en función de las etiquetas conocidas.

Continuando la evaluación, se ha analizado el algoritmo basado en Transformers llamado `cardiffnlp/twitter-roberta-base-emotion` en el contexto del análisis de sentimientos en redes sociales. Durante la ejecución de este algoritmo, se registró un tiempo de procesamiento de aproximadamente 42 minutos. A pesar de requerir un tiempo más prolongado en comparación con otros enfoques, los resultados obtenidos revelaron una precisión (accuracy) del 0.78128 en la clasificación de sentimientos.

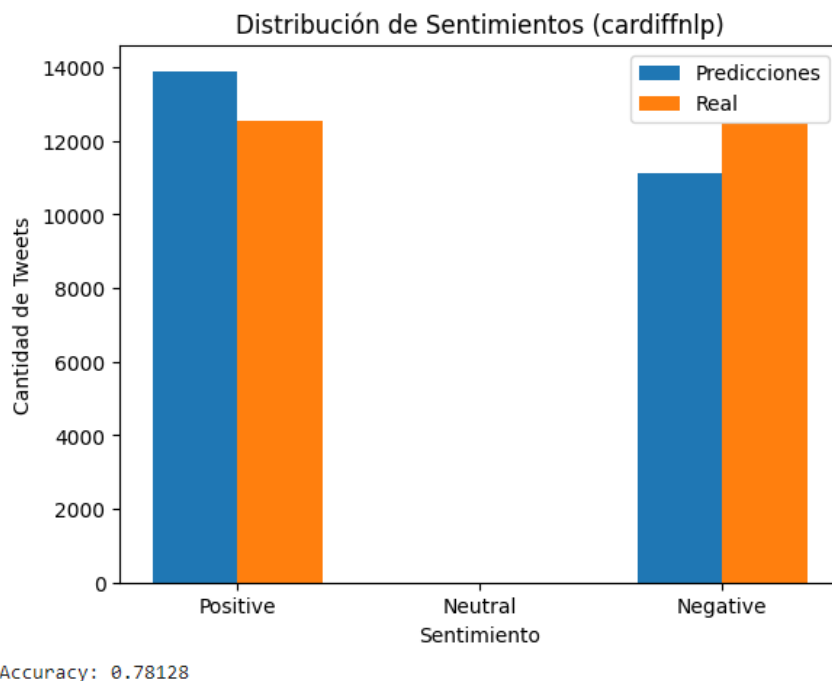


Figura 5.6: *Distribución de Sentimientos con cardiffnlp*

Por último, el modelo pre-entrenado `"michellejeli/emotion_text_classifier"` para el análisis de sentimientos en redes sociales. Durante su ejecución, se logró un tiempo de procesamiento de aproximadamente 32 minutos, con una precisión (accuracy) de 0.75342 en la clasificación de sentimientos.

Estos resultados destacan la capacidad del modelo `"michellejeli/emotion_text_classifier"` para capturar la polaridad y los matices emocionales en los textos analizados. Al aprovechar arquitecturas basadas en Transformers y mecanismos de atención, el modelo es capaz de comprender el

contexto en el que se expresan los sentimientos y capturar relaciones complejas en el texto.

Es importante tener en cuenta que el rendimiento del modelo puede verse afectado por diversos factores, como la calidad y cantidad de datos de entrenamiento, la selección de hiperparámetros y la capacidad computacional disponible. Sin embargo, hasta ahora, los resultados obtenidos respaldan la capacidad del modelo "michellejeli/emotion\_text\_classifier" para lograr una precisión significativa en el análisis de sentimientos en redes sociales.

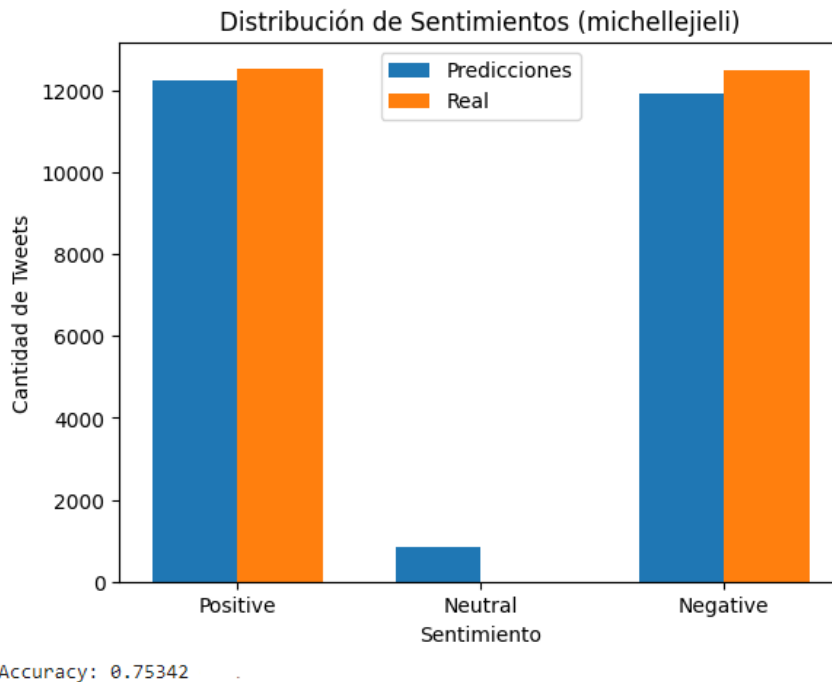


Figura 5.7: *Distribución de Sentimientos con michellejeli*

# Capítulo 6.

## Conclusiones y líneas futuras

Las conclusiones obtenidas señalan que, en términos generales, se observó un rendimiento eficiente en el análisis de sentimientos al evaluar los cinco algoritmos seleccionados (TextBlob, Vader, Pattern, Flair y Transformers). A pesar de ello, con el objetivo de alcanzar una mayor precisión y mejorar los resultados, surgió la necesidad de explorar nuevas alternativas, como los modelos pre-entrenados. Cada uno de estos algoritmos presenta sus propias fortalezas y limitaciones, y la selección del enfoque más adecuado dependerá del contexto y los requisitos particulares del proyecto en cuestión.

Comenzamos con algoritmos como TextBlob y Pattern, que tienen una mayor eficiencia en términos de tiempo de procesamiento, pero logran una precisión moderada en la clasificación de sentimientos.

Luego, exploramos el algoritmo VADER, que requiere más tiempo computacional pero obtuvo resultados alentadores en términos de precisión. VADER utiliza léxicos y reglas predefinidas para capturar la polaridad de los sentimientos expresados en los textos.

Continuamos con el algoritmo Flair, que utiliza redes neuronales y demoró más tiempo en procesar los datos. Sin embargo, logró una mayor precisión en la clasificación de sentimientos, lo que destaca su capacidad para capturar tanto la polaridad como los matices emocionales en los textos.

Luego, evaluamos el modelo basado en Transformers llamado "cardiffnlp/twitter-roberta-base-emotion". Aunque requirió un tiempo de procesamiento más prolongado, obtuvo una precisión del 0.78128 en la clasificación de sentimientos. Esto demuestra su capacidad para capturar la polaridad y los matices emocionales en los textos analizados. Al aprovechar arquitecturas basadas en Transformers, este modelo puede comprender el contexto en el que se expresan los sentimientos y capturar relaciones complejas en el texto.

Finalmente, evaluamos el modelo pre-entrenado "michellejeli/emotion\_text\_classifier". Aunque también requirió un tiempo de procesamiento prolongado (aproximadamente 32 minutos), logró una precisión del 0.75342 en la clasificación de sentimientos. Este modelo también muestra una capacidad significativa para capturar la polaridad y los matices emocionales en los textos analizados.

Es importante tener en cuenta que el rendimiento de cada algoritmo o modelo puede verse influenciado por diferentes factores, como la calidad de los datos de entrenamiento, la selección de hiperparámetros y la capacidad computacional disponible. Sin embargo, estos resultados respaldan la capacidad de ambos

modelos,"cardiffnlp/twitter-roberta-base-emotion" y "michellejeli/emotion\_text\_classifier", para lograr una precisión significativa en el análisis de sentimientos en redes sociales.

Basándonos en los resultados de la evaluación de diferentes algoritmos y modelos para el análisis de sentimientos en redes sociales, podemos extraer las siguientes líneas futuras:

1. Mejora de la precisión: Investigar y probar diferentes técnicas y modelos de aprendizaje automático para mejorar la precisión del análisis de sentimientos.
2. Análisis de sentimientos multilingüe: Ampliar el análisis de sentimientos para abarcar múltiples idiomas. Esto implica adaptar los modelos existentes a diferentes idiomas y culturas, o incluso desarrollar modelos específicos para cada idioma. También se podría investigar la transferencia de conocimientos entre idiomas para mejorar la eficiencia del proceso.
3. Detección de sarcasmo e ironía: Desarrollar técnicas y modelos capaces de detectar el sarcasmo y la ironía en los mensajes de las redes sociales. Estos elementos comunicativos a menudo presentan desafíos adicionales debido a su naturaleza subversiva y no literal.
4. Análisis de sentimientos en tiempo real: Adaptar los algoritmos y modelos existentes para realizar el análisis de sentimientos en tiempo real, lo que permite obtener resultados instantáneos y en tiempo casi real. Esto puede ser útil para monitorear la opinión pública durante eventos en vivo, campañas de marketing en tiempo real o seguimiento de la reputación de una marca en tiempo real.
5. Contexto y personalización: Considerar el contexto y la personalización en el análisis de sentimientos. Esto implica tener en cuenta factores como la demografía del usuario, el contexto de la publicación, la polaridad histórica del usuario y otros datos relevantes para proporcionar análisis de sentimientos más precisos y adaptados a cada situación.
6. Visualización de datos: Desarrollar técnicas de visualización efectivas para representar los resultados del análisis de sentimientos en redes sociales. Esto puede incluir gráficos interactivos, mapas de calor, nubes de palabras y otros métodos visuales que permitan comprender y comunicar de manera efectiva las tendencias y patrones de sentimiento en los datos.
7. Aplicaciones específicas: Explorar aplicaciones específicas del análisis de sentimientos en redes sociales, como la detección de noticias falsas, la identificación de tendencias emergentes, el análisis de opinión política o la

detección de eventos importantes en tiempo real. Cada aplicación puede requerir enfoques y técnicas específicas adaptadas a su contexto y objetivos.

Estas líneas futuras pueden ayudar a mejorar la efectividad y la aplicabilidad del análisis de sentimientos en redes sociales, permitiendo una comprensión más profunda de las emociones y opiniones expresadas en línea y su impacto en diferentes áreas y sectores.



# Bibliografía

- [1] Wang D, Szymanski B.K, Abdelzaher T, Heng Ji, Kaplan L (2018). The age of social sensing. IEEE Computer Society.
- [2] Sailunaz K, Dhaliwal M, Rokne J, Alhadj R (2018). Emotion detection from text and speech: a survey. Social Network Analysis and Mining.
- [3] Repustate. (2021) Sentiment analysis challenges and how to overcome them. [Online]. <https://www.repustate.com/blog/sentimentanalysis-challenges-with-solutions>
- [4] Tutorial de Análisis de Sentimientos en twitter | español (2021) YouTube. Available at: <https://www.youtube.com/watch?v=QdiVb0t26DI> (Accessed: January 11, 2023).
- [5] Felipe Martinez. (2021, 6 febrero). *Tutorial de Análisis de Sentimientos en Twitter | ESPAÑOL*. YouTube. <https://www.youtube.com/watch?v=QdiVb0t26DI>
- [6] Aguirre, C., & Solorio, T. (2016). Análisis de sentimientos en redes sociales: revisión sistemática.
- [7] Medina, D., & Montoyo, A. (2017). Análisis de sentimientos en redes sociales: estado del arte. Procesamiento de Lenguaje Natural.
- [8] Pérez-Rosas, V., Mihalcea, R., & Morency, L. P. (2013). Sentiment analysis of Spanish tweets. Proceedings of the Second Joint Conference on Lexical and Computational Semantics (\* SEM), Volume 2: Proceedings of the Seventh
- [9] Rangel, F., Rosso, P., & Potthast, M. (2015). SemEval-2015 task 10: Sentiment analysis in Twitter.
- [10] Taboada, M., & Alvarez-Rosete, A. (2015). Análisis de sentimientos. En F. Rangel, R. Rosso, & M. Potthast (Eds.).
- [11] Villena-Román, J., & Pérez-Rosas, V. (2016). Análisis de sentimientos en textos cortos: un enfoque basado en traducción automática y aprendizaje supervisado. Procesamiento de Lenguaje Natural.
- [12] Jain, V. K., Kumar, S., & Fernandes, S. L. (2017). Extraction of emotions from multilingual text using intelligent text processing and computational linguistics.
- [13] Kang, X., Ren, F., & Wu, Y. (2018). Exploring latent semantic information for textual emotion recognition in blog articles.

[14] Yat Mei Lee, S., Chen, Y., & Huang, C. (2010). A text-driven rule-based system for emotion cause detection.

[15] Chen, Y., & Skiena, S. (2014). Building Sentiment Lexicons for All Major Languages.

[16] Kralj Novak, P., Smailovic, J., Sluban, B., & Mozetic, I. (2015). Sentiment of Emojis.

[17] Análisis de Sentimientos con Python y NLTK - PildorasInformaticas. Disponible en: <https://www.youtube.com/watch?v=b5pFv9NB9fs>

[18] Análisis de Sentimientos con Python - CódigoFacilito. Disponible en: <https://www.youtube.com/watch?v=i4Tp7S9TIVQ>

[19] Análisis de Sentimientos en Twitter con Python y Tweepy - PildorasInformaticas. Disponible en: [https://www.youtube.com/watch?v=ImCp\\_qygGsM](https://www.youtube.com/watch?v=ImCp_qygGsM)

[20] Análisis de Sentimientos en Redes Sociales con Python - Geeky Theory. Disponible en: <https://www.youtube.com/watch?v=4G5F-c13SvU>