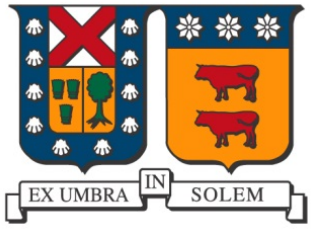
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO - CHILE



“ANÁLISIS DEL SENTIR SANSANO EN TIEMPOS DE PANDEMIA MEDIANTE TÉCNICAS DE MACHINE LEARNING Y VISUALIZACIÓN DE DATOS”

ISIDORA UBILLA ZAVALA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERA CIVIL EN INFORMÁTICA

Profesor Guía: José Luis Martí Lara
Profesor Correferente: Nicolás Torres Rudloff

Octubre - 2023

DEDICATORIA

Dedico mi memoria a mi madre Marlene y a mi abuela Rosa. Son dos mujeres excepcionales que me han ayudado a ser quien soy y a alcanzar mis sueños.

AGRADECIMIENTOS

Primero, agradecer a mi madre Marlene, quien con su amor, esfuerzo y dedicación me ha proporcionado todas las herramientas necesarias para ser la mujer que soy hoy en día. Agradezco eternamente a mi abuela Rosa, que ha cuidado de mí desde que nací y que seguirá haciéndolo como si todavía fuera una niña. Por otro lado, debo agradecer a los hombres importantes de mi familia. Comenzando con mis hermanos Iván, Tomás y Joaquín, por su amor, cariño, compañía y confianza en su hermana mayor. A mi padre Leonardo por las lecciones de vida, su ejemplo de esfuerzo y por permitirme jugar en su computadora cuando era niña. Hoy, estoy saliendo de la carrera de informática en parte gracias a su familiarización temprana con las computadoras. Además, agradezco a mi abuelo Ramón por su compañía, cuidados y preocupación, especialmente durante mis días universitarios. Finalmente a Mia, mi cachorra que lleva conmigo desde que comencé mi travesía en la informática.

Agradezco a mi pareja Tomás por ser un gran compañero, por celebrar mis triunfos y acompañarme en ellos, por el tiempo que dedicas para ayudarme, tu cariño, el apoyo y, sobre todo, el amor y alegría que me brindas.

La universidad me permitió conocer a Jered, Pedro, Camilo, Rodrigo, Javier, Alfredo, Benjamín, Gonzalo, Clemente y muchos más. Por otro lado, a Fernanda, que me ha acompañado desde el colegio. Son personas maravillosas que han alegrado infinitamente mis días durante este camino. Muchos de ellos confiaron en mí, me brindaron oportunidades muy valiosas, se tomaron el tiempo de enseñarme y guiarme con consejos. Agradezco a mis amigos, tanto a los que están presentes como a los que estuvieron. Sin ellos, no hubiera sido lo mismo.

Agradezco a los profesores, en especial a José Luis Martí Lara, quien siempre fue una guía y apoyo en todos los aspectos durante mi trayecto en la carrera. Al profesor Nicolás Torres Rudloff, por guiarme en esta etapa final, aconsejarme y confiar en mí. Y al profesor Luis Hevia, por su flexibilidad cuando más la necesitaba. Agradezco a ellos y a todos los que me formaron como ingeniera civil en informática.

Finalmente, agradezco a esa Isidora del pasado, quien tuvo la fortaleza, determinación, dedicación y esfuerzo para seguir adelante. Sin ella, no estaría aquí hoy.

RESUMEN

Resumen— Este trabajo de memoria se centra en el análisis de sentimientos de las confesiones de estudiantes de la Universidad Técnica Santa María publicadas en la red social *Instagram* durante pandemia. Para lograrlo, se aplicaron técnicas de aprendizaje automático, específicamente utilizando modelos de procesamiento del lenguaje natural, algoritmos de clúster y visualizaciones de datos. Los resultados destacan que las emociones y sentimientos expresados en los textos tienden a ser más negativos que positivos. Además, se identificó que el algoritmo Fuzzy-C Means con un parámetro m=1.01 se desempeñó como el mejor modelo de clusterización entre los probados. La relevancia de este trabajo radica en su potencial para comprender las necesidades de los alumnos y ofrecer apoyo temprano a aquellos que puedan requerirlo. En un contexto de creciente importancia de la salud mental, la detección anticipada de posibles signos de angustia o depresión a través del análisis de texto en las publicaciones de *Instagram* se convierte en una herramienta valiosa.

Palabras Clave— Procesamiento Lenguaje Natural, Análisis de sentimiento, Aprendizaje Automático.

ABSTRACT

***Abstract*—** This thesis work focuses on sentiment analysis of confessions shared by students of Universidad Técnica Santa María on the platform Instagram during pandemic. To achieve this, machine learning techniques were applied, specifically utilizing natural language processing models, clustering algorithms, and data visualizations. The results reveal that the emotions and sentiments expressed in the texts tend to be more negative than positive. Additionally, it was identified that the Fuzzy-C Means algorithm with a parameter m=1.01 performed as the best clustering model among those tested. The significance of this work lies in its potential to comprehend the needs of students and provide early support to those who may require it. In a context of increasing emphasis on mental health, the early detection of possible signs of distress or depression through text analysis in Instagram posts becomes a valuable tool.

***Keywords*—** Natural Language Processing, Sentiment Analysis, Machine Learning

GLOSARIO

BN: Bayesian Network.
CNN: Redes Neuronales Convolucionales.
CSV: Comma Separated Values.
CRISP-DM: Cross Industry Standard Process for Data Mining.
DI: Departamento de Informática.
FCM: Fuzzy C-Means.
GMM: Gaussian Mixture Model.
KDD: Knowledge Discovery in Databases.
ME: Máxima Entropía.
NB: Naive Bayes.
NLG: Natural Language Generation.
NLP: Natural Language Processing.
NLU: Natural Language Understanding.
OCR: Optical Character Recognition.
OMS: Organización Mundial de la Salud.
PCA: Principal Component Analysis.
RNA: Redes Neuronales Artificiales.
RNN: Redes Neuronales Recurrentes.
SEMMA: Sample, Explore, Modify, Model, Assess.
SVM: Support Vector Machines.
TA: Trastornos de Ansiedad.
TDM: Trastorno de Depresión Mayor.
TI: Tecnologías de la Información y Comunicación.
UNESCO: Organización de las Naciones Unidas para la Educación, la Cultura y la Ciencia.
UTFSM: Universidad Técnica Federico Santa María.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

La presente investigación, bajo el título .ªnálisis del Sentir Sansano en Tiempos de Pandemia Mediante Técnicas de Machine Learning y Visualización de Datos,"se sumerge en el desafiante contexto de la pandemia de COVID-19. Su objetivo principal es llevar a cabo un análisis de sentimiento en las publicaciones escritas por estudiantes de la Universidad Técnica Federico Santa María en la red social *Instagram*. La importancia de este estudio radica en comprender y cuantificar los sentimientos y emociones de los sansanos durante este período.

La propuesta de solución se basa en la aplicación de técnicas de procesamiento de lenguaje natural, modelos de clasificación de sentimientos y algoritmos de aprendizaje no supervisado, se busca identificar patrones emocionales, tendencias y factores subyacentes que han influido en el sentir de la comunidad sansana durante la pandemia.

Para proporcionar una visión general de la estructura de esta memoria, el documento se divide en varios capítulos. El Capítulo 1, **Definición del Problema**, establece el contexto y los objetivos de la investigación, destacando la importancia de comprender el impacto emocional de la pandemia en los estudiantes.

El Capítulo 2, **Marco Conceptual**, proporciona las bases teóricas necesarias para comprender las técnicas utilizadas, incluyendo el reconocimiento óptico de caracteres, el procesamiento de lenguaje natural, el análisis de sentimientos y las metodologías de minería de datos.

El Capítulo 3, **Propuesta de Solución**, describe en detalle la metodología empleada, desde la selección de datos hasta el preprocesamiento y la minería de datos.

El Capítulo 4, **Análisis de Resultados y Validación**, presenta los hallazgos de la investigación, incluyendo análisis de sentimientos, visualizaciones de datos y evaluación de resultados.

Finalmente, se presentan las **Conclusiones** que se desprenden de todo el trabajo realizado, desde la metodología utilizada, hasta lo aprendido con la experiencia de realizar este trabajo de titulación.

CAPÍTULO 1
DEFINICIÓN DEL PROBLEMA

En este capítulo, se presenta el contexto en el que surge el problema que se abordará en este trabajo de titulación. Además, se detallan el objetivo general y los objetivos específicos diseñados para resolver dicho problema.

1.1. Contexto

Durante los primeros dos años de la pandemia de COVID-19 (2020-2021), causada por el virus SARS-CoV2, gran parte de la población del planeta se vio en la obligación de entrar en confinamiento como medida sanitaria para evitar la rápida propagación del virus. Esta situación afectó el estilo de vida de miles de personas en ámbitos como la educación, la economía, el trabajo, la vida social, el comercio y, sobretodo, la salud. En consecuencia, el aislamiento social trajo consigo muchos efectos colaterales, entre ellos destaca el fuerte deterioro de la salud mental de la población global, sentimientos como la soledad, pesimismo, desmotivación, ansiedad, miedo, incertidumbre ante el futuro y fatiga de la rutina monótona en los hogares.

La Organización Mundial de la Salud (OMS) alertó sobre la importancia de la salud mental en el contexto de pandemia, con un estudio científico que dio a conocer el alarmante aumento de un 25 % en Trastornos Depresivo mayor (TDM) y en Trastornos de Ansiedad (TA) en todo el mundo durante sólo el primer año de confinamiento [Organization, 2022]. Dentro de la población más afectada se encuentran aquellas personas que vivían en lugares con las mayores tasas diarias de infección de COVID-19, especialmente las mujeres y los jóvenes entre los 20 y 24 años (edad en la que se suele cursar la universidad o estudios superiores), quienes demostraron ser más susceptibles a tener pensamientos suicidas o comportamiento autodestructivos.

En el ámbito de la educación se tuvo que responder al desafío de enseñar a distancia dado el cierre temporal de miles de escuelas, institutos, universidades y centros educativos de todo tipo, según cifras entregadas por la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO), para mediados de Mayo del 2020 alrededor de 1.200 millones de estudiantes de todo el mundo y todos los niveles de enseñanza, habían dejado de asistir a clases presenciales en sus respectivos centros educacionales, y más de 160 millones correspondían a estudiantes América Latina y el Caribe [Cepal y UNESCO, 2020]

Las clases debieron adaptarse rápidamente a una modalidad online [Salakhova *et al.*, 2022], lo que supuso un montón de obstáculos para el sistema educacional. Un claro ejemplo de esto es la falta de infraestructura por parte de los estudiantes o profesores, la imposibilidad de los universitarios para concentrarse en un ambiente de estudio adaptado a los hogares, la

falta de una buena conectividad a la red por parte de los actores que participan en las clases [Seminara, 2021], entre otros problemas similares que impiden alcanzar la misma efectividad en el aprendizaje que si se logra en una clase presencial.

En el caso de Chile y producto de lo anterior, se produjo un aumento en el número de universitarios que optaron por congelar su carrera y en los que deciden retirarse directamente [Emol, 2020]. Durante el primer semestre de clases online (2020), en la Universidad Técnica Federico Santa María (UTFSM) se contabilizaban 744 alumnos que habían decidido congelar la carrera al 30 de junio, mientras que 425 decidieron retirarse de los estudios [Tercera, 2020], números que causan preocupación y que debería ser materia de estudio para determinar los factores que conllevan a los sansanos a decidir por estas opciones.

El malestar existente por parte de los estudiantes de todas las sedes de la UTFSM, que surge a raíz de los diversos problemas detallados previamente y bajo el contexto de pandemia de COVID-19, se vio reflejado en una gran cantidad de publicaciones que se relacionan a problemas de salud mental en la red social de Instagram mediante páginas orientadas a las “confesiones anónimas”. Estos espacios virtuales son creados y administrados por estudiantes de la misma universidad solo por diversión, donde los sansanos pueden descargar sus desahogos, preguntas, confesar problemas y opiniones respecto a cualquier tema que se desee mediante un texto escrito en un formulario de Google de forma completamente anónima. Luego, una vez que los moderadores los reciben, son subidos a las páginas donde todos los seguidores pueden leer lo que se ha confesado por otros estudiantes, sentirse identificados o discrepar con la publicación y comentar al respecto.

Durante los últimos años los estudiantes universitarios han concentrado sus esfuerzos en visibilizar la importancia de la salud mental y la necesidad de crear medidas para cuidar de ella, debido a la gran exigencia que supone estudiar una carrera universitaria. En la UTFSM durante el año 2019, previo a la pandemia, se realizó una paralización en los campus, donde uno de los principales focos de la movilización era la salud mental que ya afectaba a una gran cantidad de estudiantes, situación que empeoró con la llegada de la educación remota.

Dada la relevancia de comprender los pensamientos, preocupaciones y comentarios de los estudiantes universitarios, y la falta de análisis en relación con sus experiencias durante la pandemia, este estudio se propone investigar cómo se sintieron los alumnos durante el período comprendido entre los años 2020 y 2021, cuando las clases se llevaron a cabo en modalidad en línea.

Dada la trascendencia de comprender los pensamientos, inquietudes y comentarios de los estudiantes universitarios en el contexto de la pandemia, y dada la carencia de análisis exhaustivos sobre sus experiencias, este estudio se propone como un esfuerzo integral para explorar en profundidad cómo se sintieron los alumnos durante el periodo abarcado entre los años 2020 y 2021, cuando las clases se llevaron a cabo exclusivamente en modalidad en línea debido a la circunstancia global. A través del análisis de confesiones y testimonios enviados por los estudiantes en la plataforma Instagram durante este período, se busca obtener una comprensión más profunda y matizada de sus vivencias.

Este estudio aspira a constituir un aporte significativo al ámbito académico y a la toma de decisiones en el ámbito educativo. Se espera que los hallazgos obtenidos puedan informar a los profesionales de la educación acerca de las necesidades y preocupaciones de los estudiantes en situaciones similares en el futuro. Además, se espera que este análisis contribuya a la creación de políticas y estrategias más efectivas para la enseñanza en línea y, en última instancia, mejore la calidad de la educación superior en situaciones de crisis como la que vivimos durante la pandemia.

1.2. Objetivos

1.2.1. Objetivo General

Categorizar y analizar los sentimientos de textos escritos por sansanos publicados en la red social *Instagram*, mediante técnicas de procesamiento de lenguaje natural y visualización de datos, con el propósito de comprender mejor sus emociones y experiencias durante este período.

1.2.2. Objetivos Específicos

Se plantean los siguientes objetivos específicos para poder cumplir el objetivo general previamente descrito:

- Investigar e implementar modelos para el procesamiento del lenguaje natural aplicados en problemas de análisis de sentimientos, para la comprensión y evaluación de las emociones expresadas en los textos.
- Evaluar la calidad de los modelos de clasificación de textos con métricas de desempeño para determinar la efectividad del modelo, identificar fortalezas y debilidades y, seleccionar el mejor para el análisis de sentimientos.
- Diseñar visualizaciones para identificar los grupos obtenidos a partir de los textos analizados, para facilitar una comprensión más clara y accesible de los patrones y tendencias presentes en los datos.

CAPÍTULO 2
MARCO CONCEPTUAL

2.1. Reconocimiento Óptico de Caracteres

El Reconocimiento Óptico de Caracteres (OCR, por sus siglas en ingles, *Optical Character Recognition*) , es un proceso mediante el cual un computador es capaz de reconocer y extraer el texto presente en una imagen para su posterior utilización, principalmente en la creación de documentos de texto.

En la actual era digital, se ha vuelto cada vez más común la digitalización de documentos. Esto se debe al auge de los dispositivos móviles y las Tecnologías de la Información y Comunicación (TIC). En diferentes sectores, como la gestión de facturas, boletas y documentos legales, se emplean documentos impresos que contienen valiosa información que requiere ser convertida en formato digital para una adecuada administración. No obstante, realizar este proceso de manera manual resulta agotador y tedioso. Es en este contexto donde adquiere relevancia el OCR, ya que permite automatizar la digitalización al extraer automáticamente el texto presente en las imágenes. De esta manera, se facilita la transformación de documentos impresos en documentos digitales que pueden ser procesados eficientemente por software empresarial.

Para detectar y extraer los caracteres dentro de una imagen, se sigue un flujo compuesto por 6 pasos [Boulid *et al.*, 2017], tal como se observa en la Figura 1.

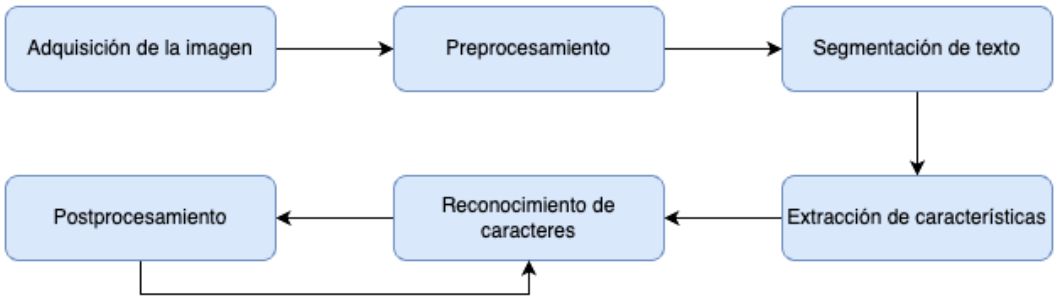


Figura 1: Pipeline OCR
Fuente: Elaboración propia

Dichos pasos se mencionan a continuación:

1. **Adquisición de la imagen:** El primer paso consiste en obtener la imagen que contiene el texto que se desea reconocer. Esta imagen puede ser una fotografía tomada con una

cámara, un escaneo de un documento impreso o cualquier otro formato de imagen.

2. **Preprocesamiento:** Antes de aplicar el OCR, es común realizar una serie de operaciones de preprocesamiento en la imagen para mejorar la calidad y facilitar la extracción del texto. Esto puede incluir la eliminación de ruido, ajuste de contraste, corrección de perspectiva y segmentación de texto.

3. **Segmentación de texto:** En este paso se identifican y se separan las regiones de la imagen que contienen texto. Esto implica detectar y delimitar cada uno de los caracteres, palabras o bloques de texto presentes en la imagen.

4. **Extracción de características:** Se extraen características representativas de cada segmentación, para esto se pueden obtener características del tipo forma, trazos, direcciones, intersecciones, etc.

5. **Reconocimiento de caracteres:** Una vez que la imagen ha sido segmentada, se aplica un algoritmo de reconocimiento de caracteres para convertir las regiones de texto en caracteres reconocibles. Este paso puede basarse en el análisis de características visuales de los caracteres o utilizar técnicas de aprendizaje automático.

6. **Postprocesamiento:** Después de realizar el reconocimiento de caracteres, se pueden aplicar técnicas adicionales para mejorar la precisión y la coherencia del texto reconocido. Esto puede incluir correcciones ortográficas, análisis de contexto y corrección de errores.

2.1.1. Motores OCR

En el ámbito del OCR, existen motores destacados que ofrecen soluciones efectivas para extraer texto de imágenes y documentos en este idioma. Estos motores están diseñados específicamente para trabajar con la complejidad y diversidad del español, brindando resultados precisos y confiables. Con su capacidad para reconocer y procesar texto en español, estos motores abren posibilidades en áreas como la gestión documental, traducción automática e investigación.

A continuación, se presentan algunos de los mejores motores de OCR en español que brindan un procesamiento de texto optimizado en este idioma:

- **Tesseract:** Desarrollado por *Google*, *Tesseract* es un motor OCR de código abierto ampliamente utilizado que ofrece alta precisión y reconocimiento de texto en más de 100 idiomas [OCR, 2023]. Es altamente flexible y puede trabajar con diversos formatos de imagen, ofreciendo funciones avanzadas como la detección de columnas y el reconocimiento de tablas.

- **ABBYY FineReader:** ABBYY FineReader es un motor OCR comercial, de pago, conocido por su alta precisión y velocidad de procesamiento [abb,]. Ofrece soporte para más de 200 idiomas, capacidad para extraer datos estructurados y funciones avanzadas de corrección y edición.

- **Microsoft Azure OCR:** Desarrollado por Microsoft, Microsoft Azure OCR es un motor OCR basado en la nube con una API fácil de usar [PatrickFarley y eric urban, 2023]. Ofrece soporte para varios idiomas, buena precisión y la ventaja de la integración con otros servicios en la plataforma Azure.

- **Adobe Acrobat OCR:** Adobe Acrobat OCR es un motor OCR de pago, incluido en la suite de productos Adobe Acrobat [Adobe,]. Es conocido por su precisión, capacidad para preservar el formato y la estructura del documento original, y permite la extracción de texto de imágenes en archivos PDF.

2.1.2. Bibliotecas para OCR en Python

Python ofrece varias bibliotecas para realizar OCR en español. Estas herramientas permiten extraer texto de imágenes y documentos en este idioma de manera eficiente y precisa. Con esta variedad de opciones, es posible automatizar tareas de extracción y análisis de información en documentos en español, abriendo oportunidades en la gestión documental, investigación, traducción automática y más.

A continuación, se mencionan algunas de las mejores librerías de OCR disponibles:

- **Pytesseract:** *Pytesseract* es un paquete de Python que actúa como una interfaz para utilizar el motor *Tesseract* [madmaze, 2022]. Es fácil de usar y permite extraer texto de imágenes y documentos con solo unas pocas líneas de código. También ofrece opciones para ajustar configuraciones y mejorar la precisión.

- **OpenCV:** OpenCV es una biblioteca de visión por computadora ampliamente utilizada que también ofrece funciones para el procesamiento de imágenes y el reconocimiento de texto [OpenCV, 2023]. Puede realizar operaciones de preprocesamiento en imágenes, como eliminación de ruido, mejora de contraste y segmentación de texto, antes de aplicar técnicas de OCR.

- **PyOCR:** PyOCR es una interfaz sencilla que permite utilizar varios motores OCR, incluyendo Tesseract, OCRopus y GOCR. Ofrece funciones para extraer texto de imágenes y archivos PDF, y facilita la integración de diferentes motores OCR en un único flujo de trabajo [pyo, 2023].

2.2. Procesamiento del Lenguaje Natural

El procesamiento del lenguaje natural (NLP, son las siglas de *Natural Language Processing* en inglés,) es una rama de la inteligencia artificial que se ocupa de la interacción entre los seres humanos y computadores mediante el uso del lenguaje humano. Su objetivo principal es permitir a las máquinas comprender, interpretar y generar texto o voz de manera similar a como lo hacen los seres humanos. El funcionamiento de NLP se debe a la combinación de modelos de lingüística computacional con modelos estadísticos de aprendizaje profundo y aprendizaje automático, de esta forma se permite interpretar el lenguaje humano [IBM, 2021].

2.2.1. Componentes de NLP

NLP trabaja con datos no estructurados para convertirlos en datos estructurados que la computadora pueda procesar como texto o audio y formular respuestas relevantes al contexto [AI, 2020]. En la literatura, el procesamiento del lenguaje natural (NLP) se divide comúnmente en dos subconjuntos principales [Khurana *et al.*, 2023], tal como se muestra en la Figura 2: la Comprensión del Lenguaje Natural (NLU) y la Generación del Lenguaje Natural (NLG).

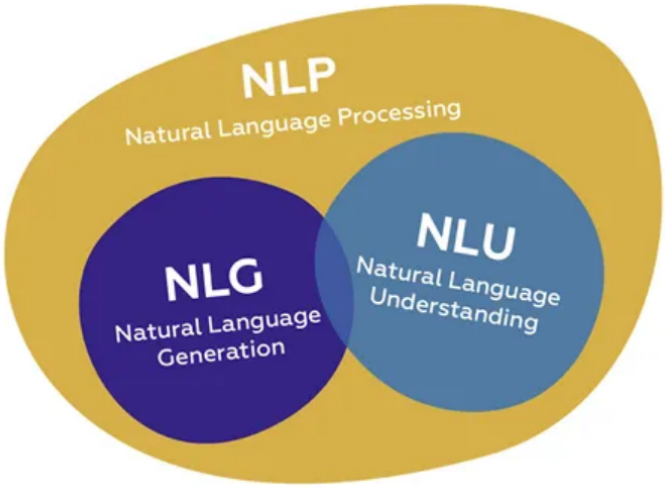


Figura 2: Componentes procesamiento lenguaje natural
Fuente: NLP, NLU y NLG: ¿Cuál es la diferencia? Una guía completa - Medium

- **Comprensión del lenguaje natural** (NLU): Se basa en la inteligencia artificial para capacitar a los computadores en comprender y extraer significado del lenguaje humano en diversas formas, como texto, voz o video. Su principal objetivo es permitir que las máquinas interpreten y comprendan el lenguaje humano de manera similar a como lo hacen las personas. Al aplicar técnicas y algoritmos de procesamiento del lenguaje natural, la NLU permite a las máquinas analizar y extraer conceptos, entidades, emocio-

nes y otros elementos del lenguaje, brindando una mayor capacidad de comprensión y comunicación con los usuarios.

- **Generación del lenguaje natural** (NLG): Es el segundo subconjunto de NLP. Se enfoca en la capacidad de las computadoras para crear texto o discurso de forma automática. Su objetivo principal es permitir que las máquinas generen contenido en lenguaje humano de manera coherente y comprensible. A través de técnicas de inteligencia artificial y procesamiento del lenguaje natural, la NLG permite a las computadoras producir informes, descripciones, respuestas y otros tipos de texto de forma automatizada. Esto tiene aplicaciones en áreas como redacción automática de noticias, generación de contenido en redes sociales y asistencia virtual, donde se requiere una producción eficiente de texto en lenguaje humano.

2.2.2. Aplicaciones de NLP

Existen diversas aplicaciones donde procesar el lenguaje natural es de gran utilidad para empresas u organizaciones de diversas áreas y, en general, se pueden encontrar múltiples ejemplos en nuestra vida cotidiana [Dojo, 2022]. En la Figura 3 se muestran alguna de las aplicaciones más importantes de NLP.

- **Traducción de lenguajes:** Para lograr traducir de un texto o un audio de un idioma origen X a otro idioma objetivo Z manteniendo el mismo valor semántico del mensaje original se aplica NLP, las que se presentan a continuación:
- **Asistentes virtuales:** La popularidad de los asistentes virtuales como Google Assistant, Siri, Alexa, Cortana, entre otros, aumenta a medida que la tecnología progresa. Para poder sonar como un humano y comunicarse con los usuarios se utiliza NLP de tal forma de comprender las conversaciones.
- **Análisis de documento:** Utilizar NLP para procesar datos no estructurados, como documentos de texto, correos, contenido de redes sociales, entre otros, es una gran ayuda cuando se trata de clasificar según su contenido u obtener información que puede servir para la toma de decisiones informada.
- **Buscadores de internet:** Para buscar en los navegadores de internet se suelen escribir palabras claves que permitan entregarnos la información más relevante relacionada a nuestra consulta, pero para lograr esto los motores de búsqueda aplican NLP para comprender y completar con parámetros más precisos, de tal forma de obtener las mejores recomendaciones de búsqueda.
- **Texto predictivo:** Otra aplicación muy común de NLP es el texto predictivo que nos ofrecen los smartphones, luego de escribir cierto conjunto de palabras se nos sugieren nuevas palabras que tienen sentido con el contexto de las utilizadas anteriormente,

se logra mediante modelos predictivos que aprenden del usuario para adaptarse a los patrones de escritura y reglas gramaticales del lenguaje. El mismo funcionamiento aplica para la corrección de gramática, ortografía y puntuación.

- **Resúmenes automáticos:** El poder resumir extensos informes, crear resúmenes y titulares de noticias, generar *abstracts* de papers científicos o obtener una síntesis de cualquier documento en general es posible mediante NLP, esto se logra mediante tres enfoques: extracción de frases claves, generación de resumen por abstracción y un híbrido de los dos previos.

- **Análisis redes sociales:** Las redes sociales han tomado un rol importante en la vida cotidiana de las personas. El uso masivo de estas genera toneladas de información valiosa de los usuarios que puede ser procesada para generar estadísticas. Mediante NLP es posible extraer preferencias de los usuarios, con esto las empresas pueden identificar el público objetivo al que desean promocionar sus productos o servicios. Esto no se limita solo a las corporaciones, las organizaciones gubernamentales, partidos políticos y organizaciones internacionales buscan realizar procesamiento del lenguaje para el análisis de redes sociales con tal de guiar la toma de decisiones, la gestión de crisis, identificación de tendencias y marketing .

- **Chatbots:** Una de las aplicaciones de NLP más populares en el último tiempo. Con ejemplos como el chatbot ChatGPT, impulsado con inteligencia artificial, puede mantener una conversación de alto nivel con el usuario en tiempo real, comprendiendo sus solicitudes y generando respuestas en texto con la información relevante. Las empresas utilizan los chatbot para servicio al cliente y/o soporte técnico, responder las dudas de los usuarios y entregar las soluciones más comunes a sus problemas.

- **Filtrado de correos:** El poder identificar y filtrar los correos no deseados que diariamente reciben los usuarios en sus bandejas de entradas es posible mediante el procesamiento de campos como remitente, asunto, archivos adjuntos y contenido del correo, de esta forma los gestores de correo electrónico pueden separar *emails* en dos grandes categorías: importantes y *spam*.

- **Análisis de sentimientos:** Internet está plagado de opiniones escritas en texto: *reviews* de productos, restaurantes y servicios, comentarios a contenido de redes sociales, pensamientos a modo de expresión de todos los temas posibles. Toda esta información se encuentra cargada de emociones y sentimientos que puede ser procesada para determinar si el sentimiento predominante es positivo, neutro o negativo, obteniendo valores y estadísticas muy valiosas para las empresas.

Figura 3: Aplicaciones de procesamiento de lenguaje natural
Fuente: Elaboración propia

2.2.3. Modelos pre-entrenados de NLP

Un modelo pre-entrenado en el ámbito de NLP se refiere a un modelo de aprendizaje profundo que ha sido previamente entrenado en grandes conjuntos de datos textuales con el fin de capturar patrones y características lingüísticas [Qiu *et al.*, 2020]. Estos modelos se entrenan en tareas generales del lenguaje, como la predicción de palabras o la clasificación de texto, en lugar de en tareas específicas. A través de este entrenamiento en *corpus* extensos, los modelos pre-entrenados adquieren conocimientos lingüísticos y semánticos que pueden ser transferidos y aplicados en diversas tareas específicas de NLP. Esto permite aprovechar el conocimiento previamente adquirido, ahorrando tiempo y recursos al no requerir entrenar un modelo desde cero para cada tarea.

Entre los modelos pre-entrenados más populares para realizar NLP, está ***Google BERT***. *Bidirectional Encoder Representations from Transformers* es un modelo pre-entrenado de lenguaje desarrollado por *Google*. Fue entrenado utilizando una enorme cantidad de datos no etiquetados, que incluyen aproximadamente 2.5 mil millones de palabras de *Wikipedia* y 800

millones de palabras de *Google Books corpus* [Devlin *et al.*, 2018]. Durante el entrenamiento, BERT se enfocó en la tarea de predicción de palabras faltantes en oraciones, lo que le permitió capturar el contexto y las relaciones entre palabras en ambos sentidos. Este enfoque bidireccional lo diferencia de otros modelos anteriores. BERT ha demostrado ser altamente efectivo en una variedad de tareas de procesamiento del lenguaje natural, como clasificación de texto, extracción de información, respuesta a preguntas y generación de texto. Su capacidad para comprender el significado de las palabras en función de su contexto ha impulsado importantes avances en el campo del NLP.

Actualmente, *Generative Pretrained Transformer* 4, mejor conocido como **GPT-4**, es el último modelo de lenguaje extenso multimodal desarrollado por *OpenAI*. Se trata de un modelo de aprendizaje profundo que combina el procesamiento del lenguaje natural y la generación de texto, siendo capaz de aceptar imágenes y texto como entradas [OpenAI, 2023]. Durante su proceso de entrenamiento, GPT-4 utiliza técnicas avanzadas de aprendizaje automático para comprender la relación entre las palabras y su orden en el lenguaje humano. Como resultado, puede generar respuestas que son prácticamente indistinguibles de las que daría un ser humano. Una de las características sobresalientes de GPT-4 es su capacidad para alcanzar un rendimiento similar al humano en diversas tareas profesionales y académicas. Se ha demostrado que supera los exámenes simulados de abogacía con una puntuación que se encuentra en el 10 % superior de los participantes. Esto muestra el nivel de sofisticación y precisión que ha alcanzado este modelo en la generación de texto.

También se cuenta con otro modelo llamado **ELMo**, por sus siglas en inglés Embeddings from Language Models, es un modelo pre-entrenado de lenguaje desarrollado por *Allen Institute for Artificial Intelligence* [Institute, 2021]. Utiliza una arquitectura de redes neuronales recurrentes bidireccionales para capturar el significado contextual de las palabras en un texto. Se entrena en un *corpus* de texto masivo, aprendiendo a predecir palabras basándose en su contexto anterior y posterior. ELMo genera una representación de palabras altamente contextualizada que abarca tanto características complejas del uso de palabras, como la sintaxis y la semántica, como también la variación de estos usos en diferentes contextos lingüísticos para modelar la polisemia. Estos vectores de palabras son resultados de funciones aprendidas a partir de los estados internos de un modelo de lenguaje bidireccional profundo, el cual se entrena previamente en un extenso *corpus* de texto. Al integrarlos fácilmente en modelos existentes, ELMo logra mejoras significativas en diversas áreas desafiantes de procesamiento del lenguaje natural, tales como responder preguntas, relacionar textos y analizar sentimientos.

Transformer-XL es un modelo de lenguaje desarrollado por *Google Brain* que soluciona la limitación de la longitud de contexto en los modelos basados en la arquitectura *Transformer* [Dai *et al.*, 2019]. Utiliza una estructura de atención recurrente y un mecanismo de memoria para capturar relaciones a largo plazo en el texto. El modelo se entrena en dos etapas: primero en un *corpus* masivo de texto para aprender patrones lingüísticos y luego se ajusta finamente en tareas específicas utilizando conjuntos de datos y técnicas de optimización. Con su arquitectura basada en atención y memoria recurrente, *Transformer-XL* supera las limita-

ciones de contexto y muestra un rendimiento eficaz en diversas tareas de procesamiento del lenguaje natural.

Otro modelo popular es **RoBERTa**, *Robustly Optimized BERT*, desarrollado por investigadores de *Facebook AI*, una variante del modelo BERT. RoBERTa es un modelo de lenguaje basado en transformadores que utiliza auto-atención para procesar secuencias de entrada y generar representaciones contextualizadas de las palabras en una oración [Liu *et al.*, 2019]. Una diferencia clave entre RoBERTa y BERT es que RoBERTa fue entrenado en un conjunto de datos mucho más grande y utilizando un procedimiento de entrenamiento más efectivo. En particular, RoBERTa fue entrenado en un conjunto de datos de 160GB de texto, que es más de 10 veces más grande que el conjunto de datos utilizado para entrenar BERT. Además, RoBERTa utiliza una técnica de enmascaramiento dinámico durante el entrenamiento que ayuda al modelo a aprender representaciones más robustas y generalizables de las palabras. Se ha demostrado que RoBERTa supera a BERT y otros modelos de vanguardia en una variedad de tareas de procesamiento del lenguaje natural, incluyendo traducción de idiomas, clasificación de texto y respuesta a preguntas. También se ha utilizado como modelo base para muchos otros modelos exitosos de procesamiento del lenguaje natural y se ha convertido en una opción popular para aplicaciones de investigación e industria.

En resumen, los modelos pre-entrenados de NLP han transformado el campo del procesamiento del lenguaje natural al proporcionar representaciones lingüísticas ricas y generalizables. Estos modelos se entrenan en grandes conjuntos de datos para capturar el contexto y la semántica del lenguaje de manera efectiva. Al utilizar estas representaciones preentrenadas, se elimina la necesidad de entrenar modelos desde cero, lo que ahorra tiempo y recursos. Estos modelos pre-entrenados se pueden adaptar a tareas específicas, como el análisis de sentimientos, la generación de texto y la traducción automática, lo que facilita el desarrollo de aplicaciones NLP más sofisticadas y precisas. Además, los modelos preentrenados han democratizado el acceso a técnicas avanzadas de procesamiento del lenguaje natural, permitiendo a más personas y organizaciones aprovechar los beneficios de la inteligencia artificial en el análisis y comprensión del lenguaje.

2.3. Análisis de Sentimientos

El análisis de sentimientos es una aplicación del procesamiento del lenguaje natural utilizada para evaluar y comprender las emociones y opiniones expresadas en textos, ya sean comentarios, reseñas, publicaciones en redes sociales, entre otros casos. A través diversas técnicas de clasificación, se puede determinar si un texto es positivo, negativo o neutro, permitiendo a las empresas y organizaciones comprender la percepción del público hacia sus marcas, productos, servicios o eventos. Este tipo de análisis es útil al obtener información valiosa para la toma de decisiones, el desarrollo de estrategias de marketing y la gestión de la reputación, contribuyendo a mejorar la experiencia del cliente y potenciar el crecimiento empresarial.

Según la literatura [Medhat *et al.*, 2014], el análisis de sentimientos puede ocurrir en tres

niveles distintos:

- **A nivel de documentos:** En este nivel el análisis de sentimientos se realiza a partir de todo el documento que almacena la opinión o reseña; la idea es clasificar este documento de opinión completamente en positivo, negativo o neutral. Los análisis se consideran dentro de este nivel cuando el documento que contiene el comentario está enfocado en un solo producto o entidad, y es escrito por un único usuario.
- **A nivel de oraciones:** El análisis de sentimientos busca determinar la polaridad de cada oración que compone la reseña. Se consideran dos pasos, primero se realiza una clasificación de subjetividad, indicando si la sentencia es del tipo objetiva o subjetiva. En caso de que la clasificación sea subjetiva, entonces se continúa con el siguiente paso, identificar si la oración es positiva o negativa.
- **A nivel de aspectos:** En este nivel se logra obtener un análisis de sentimientos más granular. Esto es posible teniendo en consideración la entidad en cuestión, quien comenta sobre la entidad, el contexto de la opinión, los aspectos mencionados de la entidad y la opinión de esos aspectos. Debido a esta lista de parámetros que se toman en cuenta para este nivel, es posible determinar qué es lo que realmente le gusta y disgusta a las personas sobre la entidad. Si bien es un nivel más profundo de análisis, es más complejo de realizar.

En la Figura 4 se visualizan las técnicas de clasificación de sentimientos. Existen dos enfoques principales: el enfoque basado en aprendizaje automático y el enfoque basado en lexicones[Shivaprasad y Shetty, 2017]. El primero se centra en el uso de algoritmos de aprendizaje automático para categorizar textos según su sentimiento, ya sea positivo, negativo o neutro. Dentro de este enfoque, se distinguen dos sub-enfoques principales: el aprendizaje supervisado y el aprendizaje no supervisado.

Por otro lado, el enfoque **basado en lexicones** tiene como objetivo determinar el valor semántico de un texto o documento. Utiliza diccionarios o lexicones predefinidos que contienen palabras asociadas a un sentimiento específico. En este enfoque, cada palabra del texto se compara con el diccionario y se le asigna un puntaje de sentimiento en función de su presencia y el valor asociado en el lexicón. Estos puntajes se pueden sumar o promediar para obtener un puntaje total de sentimiento del texto. Este enfoque se caracteriza por ser más simple y rápido de implementar, ya que no requiere el entrenamiento de modelos. Sin embargo, una limitación importante es la cobertura y precisión de los lexicones, ya que no capturan el contexto y la ambigüedad de las palabras en diferentes contextos.

Ambos enfoques presentan ventajas y desventajas. El enfoque basado en aprendizaje automático brinda mayor flexibilidad y adaptabilidad a diferentes dominios y contextos, pero requiere conjuntos de datos de entrenamiento etiquetados y un proceso de entrenamiento. Por otro lado, el enfoque basado en lexicones es más rápido de implementar, pero su eficacia depende de la calidad y cobertura del lexicón utilizado. En la práctica, se suele combinar ambos enfoques para obtener una clasificación de sentimientos más precisa y robusta.

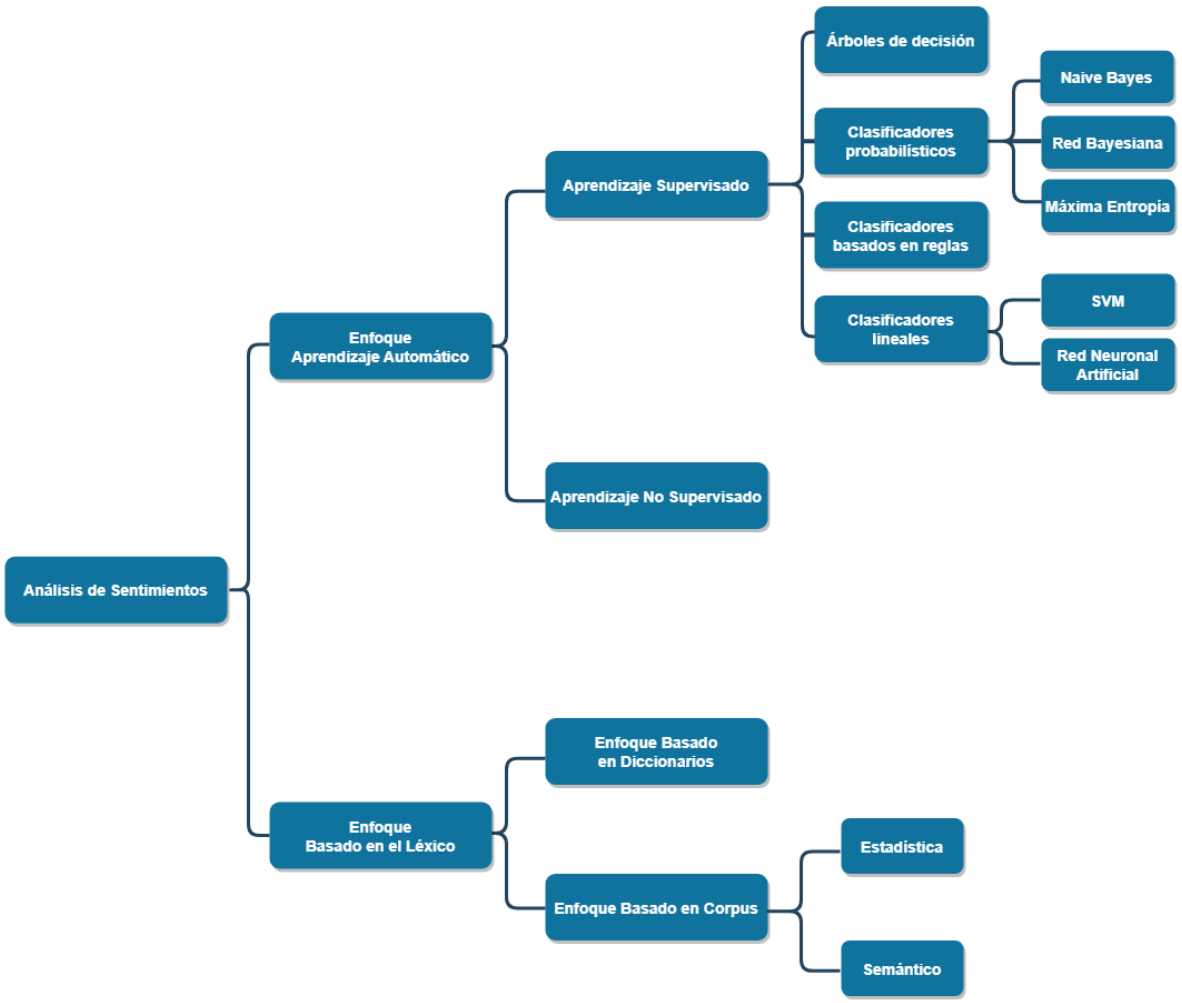


Figura 4: Técnicas de clasificación de sentimientos
Fuente: Elaboración propia

2.3.1. Aprendizaje supervisado para clasificación de sentimientos

El enfoque de aprendizaje supervisado utiliza un conjunto de datos etiquetados para entrenar un modelo capaz de predecir la polaridad de una entrada determinada, basándose en los patrones encontrados en el conjunto de entrenamiento. Este tipo de aprendizaje consta de dos pasos: entrenamiento del modelo de clasificación y predicción. Durante el entrenamiento, el modelo se alimenta y aprende de una parte del conjunto de datos inicial, lo que le permite detectar los patrones más comunes que definen cada clase. Luego, cuando se introduce una nueva entrada, el modelo es capaz de predecir la clase a la que pertenece.

En el análisis de sentimientos, se emplean varios clasificadores de aprendizaje supervisado, los cuales han demostrado ser efectivos (ver Figura 5). Entre ellos, se encuentran los clasificadores probabilísticos, los árboles de decisión, los clasificadores basados en reglas, los

clasificadores lineales y el aprendizaje profundo. Esta variedad de opciones permite abordar el análisis de sentimientos desde diferentes perspectivas y utilizar el enfoque más adecuado para obtener resultados precisos y relevantes en cada caso.

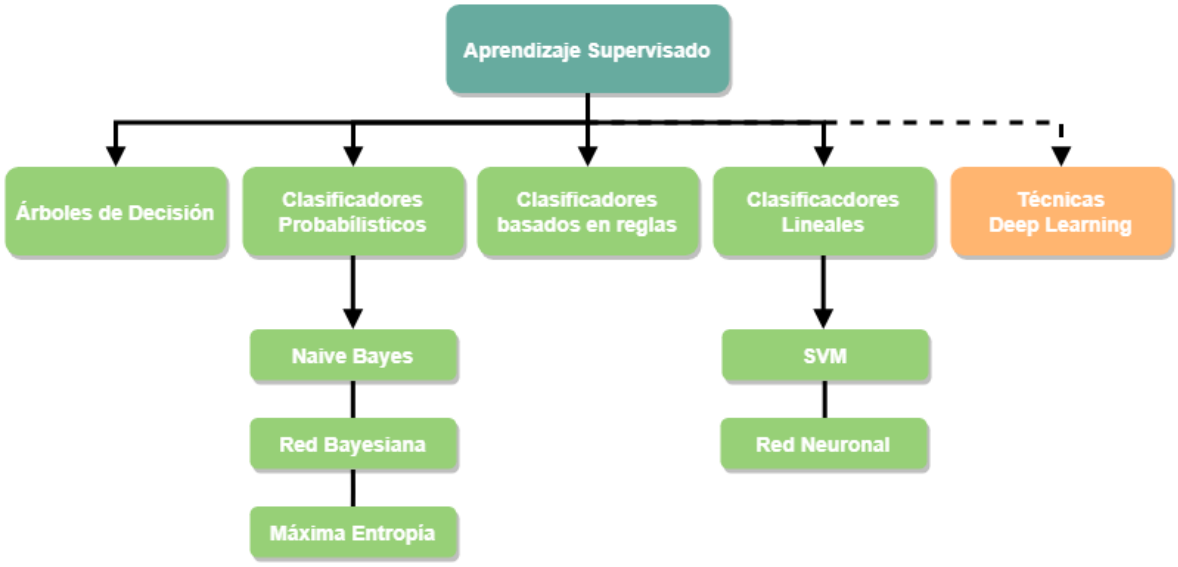


Figura 5: Algoritmos de aprendizaje supervisado para clasificación de sentimientos
Fuente: Elaboración propia

2.5.1.1. Árboles de decisión

Un árbol de decisión es una estructura jerárquica en forma de árbol que se utiliza en el análisis de sentimientos. Se basa en dividir el conjunto de datos en ramas según preguntas o condiciones relacionadas con las características del texto. Cada nodo interno del árbol representa una pregunta, y las ramas salientes indican las respuestas o decisiones posibles. El objetivo es separar eficazmente las clases de sentimiento mientras se desciende por el árbol, lo que permite realizar clasificaciones precisas. El árbol se construye evaluando características relevantes del texto y aplicando reglas de decisión en cada nodo. Esto proporciona una interpretación clara de cómo se clasifica el sentimiento y qué características son más importantes. Los árboles de decisión son valiosos para el análisis de sentimientos debido a su interpretabilidad y capacidad para capturar relaciones y patrones en los datos, facilitando la comprensión y la toma de decisiones basadas en el sentimiento del texto.

Entre los algoritmos más utilizados de árboles de decisión se encuentra ID3 [Quinlan, 1986], C4.5 [Salzberg, 1994], C5, y CART [Breiman *et al.*, 1984]. En general el funcionamiento de los arboles de decisión se pueden describir en base al algoritmo ID3, dado que los algoritmos C4.5 Y C5 son versiones mejoradas de este. Recibe un conjunto de muestras representadas como un arreglo multidimensional llamado `X` y un conjunto de atributos `A`. El algoritmo calcula la entropía `(H)` (1) y la ganancia de información `(IG)` (3) para seleccionar el atributo más informativo como el nodo raíz del árbol. Luego, se repite iterativamente el proceso de

particionar los datos en ramas hasta que todas las particiones sean homogéneas o se cumpla un criterio de parada. El árbol resultante se utiliza para clasificar nuevas muestras según las reglas de decisión establecidas durante su construcción [Sangüeza y Terrádez Gurrea, 2016].

$$\mathsf{H}(X, C) = -\sum_{i=1}^{C} p_i \log_2(p_i) \tag{1}$$

$$p_i = \frac{|C_i|}{|X|} \tag{2}$$

$$\mathsf{IG}(X, A_k) = \mathsf{H}(X, C) - \mathsf{E}(X, A_k) \tag{3}$$

$$\mathsf{E}(X, A_k) = \sum_{v \in valores(A_k)} \frac{|C_v|}{|X|} \times \mathsf{H}(X, C_v) \tag{4}$$

Los árboles de decisión tienen ventajas y desventajas para esta aplicación de NLP [Thorn, 2018]. Entre las ventajas se encuentran su interpretabilidad, lo que facilita comprender las decisiones tomadas y las características relevantes; su capacidad para capturar relaciones no lineales entre las características del texto y las clases de sentimiento; la habilidad de manejar características mixtas y diferentes tipos de información en el proceso de clasificación; y su robustez frente a datos ruidosos o valores atípicos. Sin embargo, las desventajas incluyen la tendencia al sobreajuste si se construyen árboles demasiado complejos, la dificultad de manejar conjuntos de datos grandes y la posibilidad de generar modelos demasiado complejos para ser interpretados adecuadamente.

2.5.1.2. Clasificadores probabilísticos

En el análisis de sentimientos, los clasificadores probabilísticos son ampliamente utilizados debido a su enfoque basado en la probabilidad de pertenencia a cada clase de sentimiento. Algunos de los clasificadores probabilísticos más populares en esta área son Naive Bayes, las redes bayesianas y el clasificador de máxima entropía [Mehta y Pandya, 2020]. Estos clasificadores permiten abordar el análisis de sentimientos al calcular la probabilidad de que un texto dado pertenezca a una clase de sentimiento específica. Utilizan características del texto, como palabras o frases, y aplican técnicas estadísticas para asignar probabilidades a cada clase. De esta manera, los clasificadores probabilísticos proporcionan una medida cuantitativa de la probabilidad de sentimiento asociada a un texto determinado.

El algoritmo de **Naive Bayes** (NB) es un clasificador probabilístico frecuentemente utilizado para minería de datos. Se basa en el teorema de Bayes y hace una suposición ingenua de independencia entre las características del conjunto de datos, lo que significa que considera que las características son independientes entre sí cuando se trata de predecir la clase de

una instancia. Este modelo es un clasificador popular para el análisis de sentimientos debido a su simplicidad y eficiencia. Puede trabajar con conjuntos de datos grandes y pequeños, además de manejar características categóricas y numéricas. Es simple de implementar, por lo que existen múltiples bibliotecas que permiten utilizar este clasificador. Sin embargo, tiene limitaciones, como la suposición de independencia condicional entre características, que puede no ser válida en todos los casos, sobretodo en casos del mundo real, y la incapacidad para capturar relaciones complejas entre características y sentimientos [Gandhi, 2018].

Una **Red Bayesiana** (BN) está representada por un grafo acíclico dirigido (DAG), $G = (V, D)$, y una colección de tablas de probabilidad condicional [Ruozzi, 2018]. Un DAG es un tipo de grafo dirigido en el que no hay ciclos dirigidos, lo que significa que no hay una secuencia de aristas dirigidas que comience en un vértice, y, al seguir la dirección de las flechas, eventualmente regrese al mismo vértice inicial. En otras palabras, se puede recorrer el DAG siguiendo las flechas en una sola dirección sin quedar atrapado en un bucle. Las BN se utilizan en el análisis de sentimientos para modelar las relaciones probabilísticas entre variables relevantes, como palabras clave, características lingüísticas, contexto y etiquetas de sentimiento. Los nodos representan estas variables y los arcos direccionales muestran sus dependencias probabilísticas. Con una BN, se captura la incertidumbre inherente al análisis de sentimientos y se realizan inferencias sobre las probabilidades de diferentes etiquetas de sentimiento basadas en las características observadas.

El clasificador de **Máxima Entropía** (ME) es un modelo de clasificación probabilístico que se basa en el principio de entropía máxima. Su objetivo es encontrar la distribución de probabilidad más uniforme o equilibrada dentro de las restricciones o información proporcionada [Nigam *et al.*, 1999]. En otras palabras, busca encontrar el modelo que maximice la entropía, considerando las restricciones impuestas por los datos de entrenamiento.

2.5.1.3. Clasificadores basados en reglas

Los clasificadores basados en reglas trabajan con un conjunto de reglas predefinidas que determinan la polaridad del sentimiento en un texto [Dey *et al.*, 2020]. Las reglas son diseñadas bajo una serie de criterios que siguen el principio *if*, *then*. Estas reglas pueden incluir palabras clave, frases específicas o combinaciones de palabras que se asocian con una determinada polaridad de sentimiento. El clasificador aplica las reglas al texto de entrada y asigna una etiqueta de sentimiento correspondiente [Virmani, 2022]. Aunque los clasificadores basados en reglas pueden ser efectivos en escenarios específicos, su rendimiento puede verse limitado por la capacidad de las reglas para capturar la complejidad y sutilezas de las expresiones de sentimiento en diferentes contextos.

2.5.1.4. Clasificadores lineales

Un clasificador lineal utiliza una función lineal para encontrar una línea o hiperplano en el conjunto de características que permita separar y clasificar datos en diferentes clases. Los clasificadores lineales son atractivos debido a su simplicidad y eficiencia, pero pueden tener

dificultades para capturar relaciones más complejas y sutilezas en los datos debido a su naturaleza lineal. Entre los modelos más populares utilizados en el análisis de sentimientos se encuentran las ***Support Vector Machines*** (SVM) y las redes neuronales.

Las SVM, son un tipo de algoritmos utilizado para problemas de clasificación lineal o regresiones. Su objetivo es encontrar hiperplanos óptimos que separen las distintas clases de datos de la mejor manera posible, maximizando el margen entre ellas [MathWorks,]. El hiperplano óptimo es la línea o superficie que divide las muestras de diferentes clases en el espacio de características, posicionándose de manera que maximice la distancia entre las muestras más cercanas de cada clase, lo que se conoce como ”margen”. Este margen proporciona tolerancia a errores y evita el sobreajuste del modelo. Las muestras más cercanas al hiperplano son llamadas ”vectores de soporte”, éstas influyen directamente en la definición del margen.

En el análisis de sentimientos, SVM puede identificar la polaridad emocional de un texto, permitiendo detectar si el contenido es positivo, negativo o neutral. SVM aprovecha la capacidad de separar eficientemente datos en espacios de alta dimensión, lo que lo convierte en una herramienta efectiva para tareas de clasificación de texto y análisis de sentimientos en diversas aplicaciones, como en redes sociales, comentarios de usuarios o en la industria de la opinión pública.

Una **Red Neuronal Artificial** (RNA) es un modelo computacional inspirado en la estructura y funcionamiento del cerebro humano. Consiste en un conjunto de unidades interconectadas, llamadas neuronas artificiales o nodos, que trabajan en conjunto para realizar tareas específicas, como el aprendizaje y reconocimiento de patrones en datos

El funcionamiento de una red neuronal artificial se basa en procesar información en capas interconectadas de nodos o neuronas artificiales. La información ingresa a través de una capa de entrada, donde cada nodo recibe datos. A medida que los datos pasan por las conexiones ponderadas, las neuronas calculan una salida y la transmiten a capas subsiguientes, incluyendo capas ocultas si las hay. Estas capas ocultas procesan los datos de manera iterativa, ajustando los pesos de las conexiones en función del error entre las predicciones de la red y los resultados deseados. Finalmente, la capa de salida proporciona la respuesta de la red. En el proceso de entrenamiento, se minimiza el error y se ajustan los pesos de manera que la red pueda realizar predicciones precisas en tareas como clasificación, regresión o reconocimiento de patrones.

Las redes neuronales artificiales para el análisis de sentimientos implican entrenar la red con datos etiquetados, donde se conoce el sentimiento de cada texto. La red aprende a identificar características y relaciones lingüísticas complejas que indican el sentimiento. El proceso incluye la creación de representaciones de entrada a partir de los datos de texto, que luego se introducen en las capas de la red. Estas capas, compuestas por nodos interconectados, procesan los datos de entrada y ajustan iterativamente los pesos de las conexiones para minimizar el error de predicción.

Varias arquitecturas de redes neuronales artificiales, como redes neuronales recurrentes

(RNN), redes neuronales convolucionales (CNN) y modelos basados en transformadores como BERT, han demostrado un rendimiento excepcional en tareas de análisis de sentimientos. Estos modelos no solo pueden manejar el lenguaje, sino también capturar información contextual y semántica, mejorando su precisión en la detección de matices de sentimiento, sarcasmo y más.

2.3.2. Aprendizaje no supervisado para clasificación de sentimiento

En contextos en los que el conjunto de datos carece de etiquetas, el enfoque de aprendizaje no supervisado emerge como una solución crucial. En esta dinámica, los algoritmos exploran los datos en búsqueda de patrones y similitudes con el propósito de formar agrupaciones (clústeres) basadas en los distintos valores de polaridad asociados a cada texto dentro del dataset. Entre las técnicas primordiales para este enfoque se incluyen el *clustering* y la reducción de dimensionalidad.

En lo que respecta al *clustering*, se descompone en cinco enfoques principales [Developers,]:

1. **Basado en Jerarquía**: Los algoritmos de *clustering* jerárquico son técnicas que organizan los datos en una estructura jerárquica de clústeres, lo que permite visualizar tanto grupos individuales como agrupaciones más amplias que contienen subgrupos. Estos algoritmos son útiles para representar la relación entre diferentes niveles de agrupación en los datos.

2. **Basado en Centroides**: En esta categoría, se busca el centro de cada clúster y las instancias se organizan según su proximidad a estos centros. El método *K-Means* es un claro ejemplo, donde los centroides representan el corazón de cada clúster.

3. **Basado en Distribución**: Estos algoritmos modelan la distribución de los datos y crean clústeres en función de cómo se ajustan a las distribuciones. Un ejemplo es el algoritmo de Modelo de Mezcla Gaussiana (GMM), que asume que los datos provienen de diferentes distribuciones gaussianas y trata de ajustar clústeres en función de esas distribuciones.

4. **Basado en Densidad**: En esta categoría, los clústeres se generan en áreas de alta densidad de puntos. Algoritmos como *DBSCAN* identifican regiones densas y pueden manejar clústeres de formas y tamaños irregulares, adaptándose a las variaciones en la densidad de los datos.

5. **Basado en Difusión**: Estos algoritmos consideran cómo los puntos se difunden en el espacio y crean clústeres según cómo se propaga esta difusión. Un ejemplo es el algoritmo *Fuzzy C-Means*, que asigna grados de pertenencia a cada instancia en múltiples clústeres, lo que permite que las instancias pertenezcan parcialmente a más de un grupo.

Como se mencionó anteriormente, los algoritmos basados en jerarquía representan un método de agrupación de datos que organiza elementos similares en grupos de manera jerárquica. Se dividen en dos enfoques: el aglomerativo, que fusiona clústeres semejantes, y el divisivo, que separa clústeres en subclústeres. El resultado es un dendrograma, un árbol que ilustra la estructura de agrupación en distintos niveles [Patlolla, 2018]. Esta técnica es útil para descubrir patrones en datos sin requerir un número predeterminado de clústeres. Sin embargo, puede volverse costosa para conjuntos de datos grandes y carece de la capacidad de deshacer etapas de agrupación.

El algoritmo básico de agrupamiento jerárquico, en el caso aglomerativo, se puede explicar de la siguiente manera:

1. **Calcular la matriz de proximidad**: Se evalúa la distancia entre todos los pares de puntos de datos y se crea una matriz que refleja las similitudes.
2. **Considerar cada punto como un clúster individual**: Al principio, se trata cada punto de datos como su propio clúster independiente.
3. **Repetir**: En cada iteración,
    a) Los dos clústeres más cercanos se fusionan en uno solo.
    b) La matriz de proximidad se actualiza para reflejar esta nueva estructura de clústeres.
4. **Continuar la fusión**: Se repite el paso 3 hasta que solo quede un clúster, lo que significa que todos los puntos de datos están agrupados en una única entidad.

Este proceso de fusión sucesiva se basa en la idea de que los elementos más similares se agruparán primero, creando una jerarquía de clústeres en función de su similitud. En el caso divisivo, el enfoque es similar, pero en lugar de fusionar clústeres, se dividen en subclústeres más pequeños. Cabe señalar que el proceso es irreversible, y el dendrograma resultante muestra cómo los clústeres se formaron a diferentes niveles de similitud.

El algoritmo más popular del enfoque basado en centroides es ***K-Means***. Es un algoritmo de *clustering* que tiene como objetivo agrupar un conjunto de datos en clústeres coherentes y distintos. Funciona mediante la asignación iterativa de puntos de datos a los centroides de clústeres, calculando luego nuevos centroides basados en los puntos asignados [Srivastava, 2021]. El objetivo es minimizar la suma de las distancias cuadradas entre los puntos y sus centroides correspondientes, lo que resulta en clústeres donde los puntos son similares entre sí y diferentes de los puntos en otros clústeres. *K-Means* es ampliamente utilizado en la segmentación de datos y en la identificación de patrones en diversas aplicaciones.

El funcionamiento de *K-Means* se puede reducir a los siguientes 5 pasos:

1. **Inicialización**: El proceso comienza eligiendo de manera aleatoria `k` centroides, donde `k` es el número deseado de clústeres. Los centroides son puntos en el espacio de características que representarán los centros de cada clúster.

2. **Asignación de Puntos**: Cada punto de datos se asigna al centroide más cercano en función de una medida de distancia, como la distancia euclidiana. Esto crea `k` grupos iniciales.

3. **Actualización de Centroides**: Una vez que los puntos son asignados a los centroides, se calcula el centroide promedio de cada grupo. Estos centroides actualizados representarán las nuevas posiciones centrales de los clústeres.

4. **Reasignación de Puntos**: Los puntos son reasignados a los nuevos centroides recalculados según la distancia. Esto puede resultar en una redistribución de puntos entre los clústeres.

5. **Pasos 3 y 4 Repetidos**: Los pasos de actualización de centroides y reasignación de puntos se repiten iterativamente hasta que los centroides ya no cambien significativamente o se alcance un número máximo de iteraciones.

El algoritmo **Gaussian Mixture Model (GMM)** es un modelo de clusterización basado en distribución. GMM asume que los datos están compuestos por varias distribuciones gaussianas (también conocidas como distribuciones normales) combinadas. Cada distribución gaussiana representa un clúster o grupo en los datos. La idea detrás de un GMM es que los datos se generan a partir de varias fuentes, cada una representada por una distribución gaussiana diferente. Cada distribución gaussiana tiene sus propios parámetros, como la media y la varianza. Estos parámetros se estiman a partir de los datos mediante técnicas de optimización, como el algoritmo de Expectation-Maximization (EM) [Beheshti, 2023].

El algoritmo GMM se puede explicar en los siguientes pasos:

1. **Normalización de datos:** Ajustar los datos para asegurar que cada característica tenga un promedio de cero y una variación unitaria.

2. **Inicialización de parámetros:** Elegir una suposición inicial para los vectores promedio, las matrices de covarianza y los pesos de las distribuciones Gaussianas componentes.

3. **Algoritmo EM para GMM:**

    a) **Paso de Esperanza (E-step):** Calcular la probabilidad de que cada punto de datos pertenezca a cada cluster, basándose en las estimaciones actuales de los parámetros del modelo.

    b) **Paso de Maximización (M-step):** Actualizar los parámetros del modelo para maximizar la probabilidad logarítmica completa esperada de los datos observados, dadas las probabilidades obtenidas en el paso E. Los nuevos valores se obtienen usando la estimación de máxima probabilidad (MLE).

c) Repetir el paso E y el paso M hasta alcanzar la convergencia, es decir, hasta que la probabilidad logarítmica de los datos observados deje de aumentar significativamente o se alcance un número máximo de iteraciones.

4. **Determinación del número óptimo de clusters:** Usar un criterio de selección de modelos como el Criterio de Información Bayesiana (BIC) o el Criterio de Información de Akaike (AIC) para decidir el número ideal de clusters.

En el caso de los clústeres basados en densidad destaca fuertemente ***DBSCAN***. Es un algoritmo de *clustering* que tiene como objetivo agrupar puntos de datos en regiones de alta densidad, mientras etiqueta los puntos de baja densidad como ruido [Gajjar, 2020]. Funciona encontrando áreas densas en el espacio de características y conectando los puntos dentro de esas áreas en clústeres. Los puntos que están lejos de todas las áreas densas se consideran como ruido. El algoritmo define dos parámetros críticos: `epsilon` (ϵ), que establece la distancia máxima entre dos puntos para que sean considerados vecinos, y `minPts`, el número mínimo de puntos dentro de un radio ϵ para formar un clúster. DBSCAN es especialmente útil para identificar clústeres de formas y tamaños irregulares en conjuntos de datos con ruido o outliers, y no asume una distribución específica de los datos.

El funcionamiento de DBSCAN se resume en los siguientes pasos:

1. Inicialmente el algoritmo comienza con un punto de datos de inicio no visitado al azar. Todos los puntos dentro de una distancia ε se clasifican como puntos de vecindario.

2. Se requiere un número mínimo de puntos `minPts` dentro de la vecindad para iniciar el proceso de agrupación. De lo contrario, el punto se etiqueta como 'ruido'.

3. Todos los puntos dentro de la distancia ε se convierten en parte del mismo clúster. Repite el procedimiento para todos los puntos nuevos agregados al grupo del clúster. Continúa hasta que visite y etiquete cada punto dentro de la vecindad ε del clúster.

4. Al completar el proceso, comienza de nuevo con un nuevo punto no visitado, lo que lleva al descubrimiento de más clústeres o ruido. Al final del proceso, asegúrate de marcar cada punto como parte del clúster o como ruido.

El algoritmo más destacable del enfoque basado en difusión es ***Fuzzy C-Means*** (FCM) es una técnica de agrupamiento suave, que es una extensión del algoritmo clásico de *K-Means*. FCM es especialmente útil cuando los puntos de datos pueden pertenecer a múltiples clústeres con diferentes grados de pertenencia, en lugar de una pertenencia estricta a un solo clúster. En FCM, cada punto de datos se asocia con valores de pertenencia difusos (μ), que indican el grado en que el punto pertenece a cada clúster. Además, el algoritmo utiliza un parámetro llamado `m` (*fuzziness parameter*), para controlar la difusión de los valores de pertenencia y ajustar la suavidad de las asignaciones. Con un valor menor de `m`, los grados de pertenencia se vuelven más difusos, lo que permite una mayor superposición entre los clústeres. Por otro

lado, un valor mayor de `m` hace que los grados de pertenencia sean más nítidos, similar a una asignación más dura en el K-Means convencional.

El proceso de FCM se describe a continuación:

1. **Asignación de número fijo de clústeres (k)**: Se asume un número fijo de clústeres para los cuales se busca agrupar los datos.
2. **Inicialización**: Se inicializan aleatoriamente los centroides (μ_k) asociados a los clústeres. Luego, se calcula la probabilidad de que cada punto de datos x_i sea miembro de un clúster k dado, es decir, $P(\text{punto } x_i \text{ tiene etiqueta } k|x_i, k)$. Esto indica la cercanía difusa del punto al centroide del clúster.
3. **Iteración**: Se recalcula el centroide del clúster como el centroide ponderado, considerando las probabilidades de pertenencia de todos los puntos de datos x_i. Los puntos de datos tienen diferentes grados de pertenencia a diferentes clústeres, por lo que se utilizan estos grados para ponderar sus contribuciones al cálculo del centroide.
4. **Terminación**: Se repiten los pasos de asignación de pertenencia y recálculo del centroide hasta que el proceso converja (es decir, los centroides y las probabilidades de pertenencia no cambien significativamente) o hasta que se alcance un número especificado de iteraciones definido por el usuario. Es posible que el proceso quede atrapado en máximos o mínimos locales durante las iteraciones.

Este proceso de FCM permite ajustar iterativamente los grados de pertenencia de los puntos a los clústeres y encontrar centroides que representen mejor las formas y estructuras de los datos. La flexibilidad para asignar grados de pertenencia difusos en lugar de una pertenencia binaria permite que FCM maneje situaciones donde los puntos tienen asociación mixta con múltiples clústeres.

La **reducción de dimensionalidad** es una estrategia esencial en el análisis de datos que busca abordar los retos asociados con conjuntos de datos de alta dimensionalidad [Chaitanyanarava, 2020]. Esta técnica busca transformar los datos originales en un espacio de menor dimensión para mejorar la eficiencia computacional, facilitar la visualización y la interpretación, prevenir la maldición de la dimensionalidad y reducir el ruido y el sobreajuste. Métodos como el Análisis de Componentes Principales (PCA), t-SNE y LDA son utilizados para lograr esta reducción, aunque es importante considerar que esta acción podría conllevar la pérdida de información, lo que hace necesario evaluar cuidadosamente el equilibrio entre la simplificación y la preservación de datos relevantes para la tarea analítica o predictiva en cuestión.

El **Análisis de Componentes Principales (PCA)** es una de las técnicas más utilizadas en estadísticas y análisis de datos para reducir la dimensionalidad de un conjunto de datos, mientras conserva la mayor parte de su información relevante. Se aplica principalmente en situaciones

donde se trabaja con conjuntos de datos que tienen múltiples variables, lo que puede dificultar su análisis y visualización. PCA busca transformar los datos originales en un nuevo sistema de coordenadas en el que las nuevas dimensiones, llamadas çomponentes principales", las que son una combinación lineal de las variables originales, y además son independientes entre sí. Las componentes están ordenadas de manera que la primera componente explica la mayor cantidad de variabilidad en los datos, la segunda componente explica la siguiente mayor cantidad de variabilidad, y así sucesivamente [JavaTpoint,]. Esto permite capturar las direcciones principales en las que los datos varían más.

El algoritmo PCA se puede explicar en los siguientes pasos:

1. **Normalización de datos:** Se deben normalizar los datos para asegurar que todas las variables tengan la misma escala y no dominen el análisis debido a valores grandes.
2. **Cálculo de la matriz de covarianza:** Se calcula la matriz de covarianza a partir de los datos normalizados. Esta matriz muestra cómo las variables se relacionan entre sí.
3. **Cálculo de autovectores y autovalores:** Se calculan los autovectores y autovalores de la matriz de covarianza. Los autovectores representan direcciones de máxima variación, mientras que los autovalores indican la cantidad de variabilidad en esas direcciones.
4. **Selección de componentes principales:** Ordenar los autovectores según los autovalores en orden descendente. Esto define la secuencia de componentes principales, donde la primera explica la mayor variabilidad.
5. **Proyección de datos:** Proyectar los datos originales en el nuevo sistema de coordenadas definido por los autovectores, reduciendo la dimensionalidad.
6. **Reducción de dimensionalidad:** Si se busca reducir dimensiones, selecciona las primeras n componentes principales que conserven una cantidad adecuada de variabilidad.

Otro algoritmo muy utilizado es **t-SNE (t-Distributed Stochastic Neighbor Embedding)**. Es una técnica de reducción de dimensionalidad no lineal utilizada en análisis de datos y visualización. A diferencia del PCA, que se centra en reducir la dimensionalidad conservando la estructura global, t-SNE busca preservar las relaciones locales entre puntos en un espacio de alta dimensión, al representarlos en uno de menor dimensión [Interactive Chaos,].

El objetivo principal de t-SNE es reducir un conjunto de datos con muchas características a uno de menor dimensión, normalmente a 2 o 3, de forma que puntos cercanos en el espacio de alta dimensión también lo sean en el de baja dimensión, y viceversa. Esto facilita la visualización de agrupamientos y patrones difíciles de apreciar en dimensiones más altas.

t-SNE es particularmente útil en datos complejos y no lineales, donde las relaciones entre puntos son complejas y no pueden capturarse eficazmente mediante técnicas lineales como PCA.

Los pasos clave del algoritmo t-SNE son:

1. **Medición de similitudes:** Se calcula una medida de similitud, generalmente se utiliza distancia euclidiana, entre todos los pares de puntos en el espacio de alta dimensión.
2. **Construcción de probabilidades conjuntas:** Se crean distribuciones de probabilidad que representan similitudes entre puntos. Estas probabilidades se suavizan y normalizan para obtener probabilidades conjuntas.
3. **Definición de distribuciones de probabilidad en el espacio de baja dimensión:** Se establecen distribuciones de probabilidad similares en el espacio de baja dimensión, con el objetivo de que puntos cercanos en el espacio de alta dimensión también lo sean en el espacio de baja dimensión.
4. **Optimización:** Se busca minimizar la divergencia entre las distribuciones de probabilidad en ambos espacios mediante métodos de optimización numérica. Esto implica ajustar ubicaciones de puntos en el espacio de baja dimensión para lograr una representación que conserve las relaciones de similitud deseadas.

Un proceso importante del aprendizaje no supervisado es la **validación de clústeres**, para así conocer de la calidad de los resultados de algoritmos de *clustering*. Esto es importante para evitar utilizar modelos que no desempeñan bien al momento de clústerizar, así como en situaciones donde se desee comparar dos algoritmos de *clustering* [Kassambara, ceso].

En la literatura, las estadísticas de validación de clústeres se categorizan en 3 clases:

- **Validación interna de clústeres**. Utilizan la información interna del proceso de *clustering* para evaluar la calidad de una estructura de estás sin hacer referencia a información externa. También puede utilizarse para estimar el número de clústeres y el algoritmo de *clustering* adecuado sin ningún dato externo.

  El **Silhouette Score** es una métrica interna que evalúa qué tan bien son asignados los puntos a sus propios clústeres en comparación con otros cercanos. Cuanto mayor sea el valor, mejor será la calidad del agrupamiento. Si se acerca a 1, indica que los objetos están bien asignados a sus clústeres, mientras que cerca de -1 sugiere asignaciones incorrectas.

  Calinski-Harabasz evalúa la calidad de los clústeres midiendo la relación entre la dispersión intra clústeres y la dispersión extra clústeres. Valores más altos indican clústeres más definidos y separados, lo que sugiere una mejor agrupación.

  El índice **Davies-Bouldin** evalúa la calidad de los clústeres basándose en la distancia entre los centros de los clústeres y su dispersión. Un valor más bajo indica clústeres más compactos y mejor definidos, mientras que valores más altos indican que los clústeres están menos cohesionados y más dispersos.

- **Validación externa de clústeres**. Consiste en comparar los resultados de un análisis de clústeres con un resultado conocido externamente, como etiquetas de clases proporcionadas externamente. Mide en qué medida las etiquetas de clústeres coinciden con las etiquetas de clase proporcionadas externamente. Dado que conocemos el número ”verdadero”de clústeres de antemano, este enfoque se utiliza principalmente para seleccionar el algoritmo de *clustering* adecuado para un conjunto de datos específico.

  Una métrica de validación externa es el **Purity Score**. Se enfoca en comparar cómo se distribuyen las clases reales entre los clústeres generados por un algoritmo de *clustering*. Si un clústeres contiene principalmente instancias de una sola clase, entonces su pureza será alta. En cambio, si un clústeres tiene una mezcla de diferentes clases, su pureza será menor.

- **Validación relativa de clústeres**. Evalúan la estructura de *clustering* al variar diferentes valores de parámetros para el mismo algoritmo (por ejemplo, variar el número de clústeres k). Se utiliza generalmente para determinar el número óptimo de clústeres.

2.3.3. Enfoque basado en diccionario para clasificación de sentimientos

El enfoque basado en diccionario para el análisis de sentimientos implica la creación manual de un diccionario inicial de palabras clave relacionadas con sentimientos y su polaridad. Luego, se utiliza un *corpus* relevante, como *WordNet* [Miller *et al.*, 1990], *SentiWords*, *Vader*, entre otros; para expandir este diccionario mediante la incorporación de sinónimos y antónimos apropiados. Este proceso de expansión continúa hasta que ya no se pueden agregar nuevas palabras al diccionario. Posteriormente, se realiza una evaluación manual para corregir posibles errores y refinar el diccionario.

Este enfoque es sencillo y rápido de implementar, ya que solo requiere la comparación de palabras con un diccionario preexistente. Sin embargo, presenta algunas limitaciones. Por un lado, depende en gran medida de la calidad y cobertura del diccionario utilizado, ya que las palabras no incluidas en el *corpus* no serán consideradas en la clasificación. Además, no captura las relaciones o matices más complejos entre las palabras y el sentimiento, lo que puede afectar la precisión de la clasificación en casos ambiguos o contextos particulares [Elia, 2023].

2.3.4. Enfoque basado en *corpus* para clasificación de sentimientos

El análisis de sentimientos basado en *corpus* utiliza patrones sintácticos para identificar las palabras asociadas al sentimiento y comprender su contexto. Aunque no es tan preciso como los enfoques basados en diccionarios, proporciona una comprensión más profunda de cómo las palabras expresan sentimientos en diferentes contextos. Dependiendo de la calidad del *corpus* y del algoritmo utilizado, esta técnica puede ser muy útil para el análisis de sentimientos.

2.3.5. Bibliotecas para clasificación de sentimientos en Python

En Python existen muchas bibliotecas que permiten realizar análisis de sentimientos bajo los distintos enfoques que se mencionaron previamente. A continuación se listan las más populares:

- **TextBlob:** Es una biblioteca de código abierto orientada a tareas comunes de procesamiento de lenguaje natural (NLP), construida en base a NLTK (*Natural Language Toolkit*), otra biblioteca popular para NLP. Mediante los diversos métodos que ofrece *TextBlob*, es posible realizar análisis de sentimientos, procesamiento de texto, traducciones a diversos idiomas, extracción de frases clave, tokenización y mucho más.

  El análisis de sentimientos de *TextBlob* funciona bajo el enfoque basado en lexicón. Un sentimiento se define por su orientación semántica y la intensidad de cada palabra en la oración. Esto requiere un diccionario predefinido que clasifique la polaridad de las palabras negativas y positivas. A cada palabra del texto se le asigna un puntaje, el sentimiento final se calcula mediante alguna operación de agrupación, como tomar un promedio de todos los sentimientos. La polaridad final del texto se encuentra entre los valores [-1,1], -1 define un sentimiento negativo y 1 es un sentimiento positivo.

  TextBlob ofrece una serie ventajas, como su facilidad de uso y una interfaz sencilla, lo que la hace ideal para principiantes en procesamiento de lenguaje natural. Además, cuenta con una amplia gama de funcionalidades para realizar las diversas tareas de procesamiento de texto, entre ellas: análisis de sentimientos. Además, *TextBlob* ofrece soporte para varios idiomas mediante traducciones del texto origen y es útil para tareas rápidas de procesamiento de texto en proyectos pequeños o prototipos.

  Las principales desventajas de *TextBlob* son su enfoque basado en reglas y lexicones predefinidos, lo que puede limitar su precisión en comparación con métodos más avanzados de aprendizaje automático. Además, no proporciona funcionalidades de modelado de lenguaje profundo, lo que puede restringir su capacidad para comprender textos complejos y contextos sofisticados. Otra desventaja importante es su traducción de textos al inglés para realizar el análisis de sentimientos, dado que existe el riesgo de que se altere o se pierda el significado original del texto. La traducción automática puede introducir errores y matices que pueden afectar la precisión del análisis de sentimientos.

- **Vader:** (*Valence Aware Dictionary for sEntiment Reasoning*) es un modelo de análisis sentimientos bajo el enfoque de léxico y *rule-based* para texto escrito en inglés, su aplicación esta orientado a las redes sociales pero es extrapolable a otro tipo de análisis como las críticas de películas y artículos de opinión. Para determinar la polaridad de un texto (positivo, neutro, negativo) se utiliza un diccionario de características léxicas con puntajes de sentimiento junto a un grupo de heurísticas.

  Entre las ventajas de *Vader* es posible destacar que es una herramienta de código abierto escrito en *Python* y de acceso gratuito para la comunidad. Es sencillo de utilizar

y está orientado al análisis de sentimientos en redes sociales, dado que puede procesar lenguaje característico de estas plataformas, por ejemplo, reconociendo acrónimos (*lol*), caras sonrientes en texto (*:D*) y *slang* (*cachai*). Otro punto importante a favor es que utiliza un diccionario léxico con heurísticas de reglas, esto permite que sea independiente del contexto que otros enfoques, esto deriva en un alto nivel de precisión en la clasificación de sentimientos.

En cuanto a las desventajas, *Vader* está diseñado para trabajar en inglés, si bien existe la opción de trabajar en otros idiomas, esto lo hace traduciendo al inglés, lo que puede afectar la precisión del modelo. Otra desventaja es que puede tener problemas al identificar patrones de texto más complejos dado que no es un enfoque de aprendizaje automático.

- **PySentimiento:** Es una biblioteca de código abierto en *Python* diseñada para fomentar la investigación en análisis de sentimientos y minería de opiniones en redes sociales. Ofrece modelos de vanguardia basados en redes neuronales artificiales *Transformer*, específicamente se utiliza el modelo RoBERTuito, un modelo de RoBERTa entrenado con alrededor de 5000 *tweets* en español, para el análisis de sentimientos y el análisis de emociones de manera rápida y sencilla [Pérez *et al.*, 2021]. Actualmente, *PySentimiento* cuenta con soporte para español, inglés, italiano y portugués.

  Esta biblioteca presenta ventajas significativas en el análisis de sentimientos en textos de redes sociales en español. Gracias a su entrenamiento específico en *tweets* en español, logra un rendimiento óptimo al analizar textos provenientes de plataformas como *Instagram*. Además de analizar sentimientos, *PySentimiento* proporciona funcionalidades para diversas tareas de NLP. Estas incluyen la detección de discurso de odio, la detección de ironía, el análisis de emociones, el etiquetado de entidades y partes del discurso, así como la detección contextualizada de discurso de odio y el análisis de sentimientos dirigido. La capacidad de realizar estas variedades de análisis amplía su utilidad y permite su aplicación en diferentes casos de uso, brindando flexibilidad y versatilidad en la comprensión y clasificación de textos en español en redes sociales.

  Sin embargo, también existen algunas limitaciones en *PySentimiento*. Su entrenamiento específico en *tweets* en español puede afectar su rendimiento al analizar textos que difieren en patrones y estructuras de las redes sociales. Además, puede tener dificultades para adaptarse a dominios especializados o vocabulario técnico específico, lo que podría afectar su precisión y relevancia en esos contextos. Aunque *PySentimiento* está diseñado para analizar textos en español, su rendimiento en otros idiomas puede ser limitado. Es importante tener en cuenta estas limitaciones al utilizar *PySentimiento*, para asegurar una interpretación adecuada de los resultados y considerar su aplicación en diferentes contextos y dominios lingüísticos.

2.4. Metodología para proyectos de Minería de Datos

La minería de datos es un campo que requiere de un enfoque organizado y estructurado para lograr resultados exitosos en los proyectos. Para abordar la complejidad de estos proyectos, se utilizan metodologías específicas que guían cada fase de manera adecuada. Algunas de las metodologías más populares y ampliamente utilizadas en la minería de datos son KDD (*Knowledge Discovery in Databases*), SEMMA (*Sample*, *Explore*, *Modify*, *Model*, *Assess*) y CRISP-DM (*Cross Industry Standard Process for Data Mining*). Estas metodologías proporcionan un marco de trabajo eficiente y efectivo para la gestión de proyectos de minería de datos.

2.4.1. Proceso KDD

KDD es un enfoque integral para extraer conocimiento útil y significativo a partir de grandes conjuntos de datos. Consiste en un proceso iterativo y cíclico, que consta de varias etapas, mostradas en la Figura 6, interrelacionadas diseñadas para identificar patrones, relaciones y tendencias ocultas en los datos.

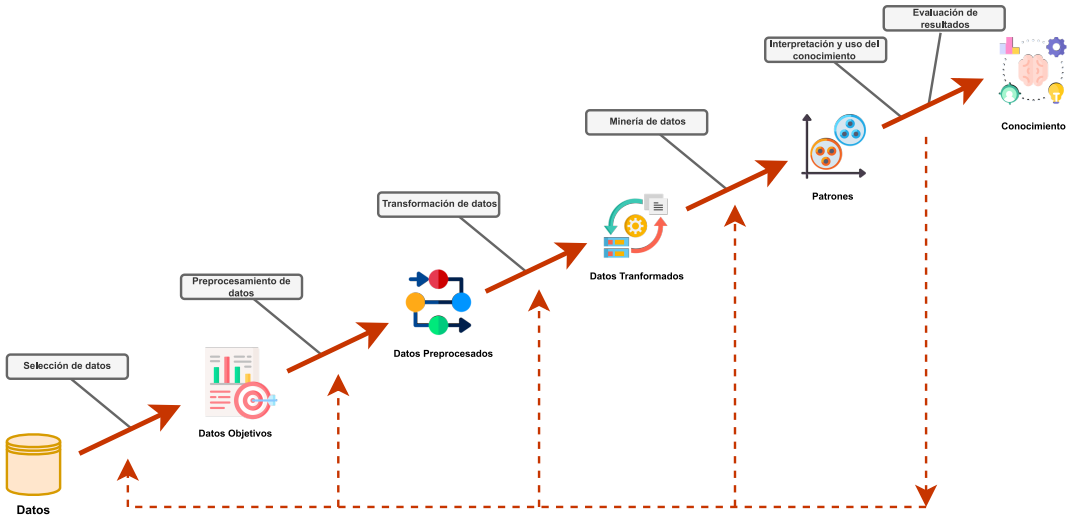


Figura 6: Etapas Proceso KDD
Fuente: Elaboración Propia

El proceso KDD se compone de las siguientes etapas:

1. **Selección de datos:** En esta etapa, se seleccionan los datos relevantes para el análisis. Se definen los criterios de selección y se recopilan los datos necesarios para el proyecto.

2. **Preprocesamiento de datos:** Los datos recolectados se someten a un trabajo de limpieza y preprocesamiento para eliminar ruido, datos inconsistentes o faltantes. Además, se pueden aplicar técnicas de transformación y normalización para preparar los datos de manera adecuada.

3. **Transformación de datos:** En esta etapa, se aplican técnicas de transformación a los datos preprocesados para resaltar características importantes o reducir la dimensionalidad de los datos. Esto puede incluir técnicas como la discretización, la reducción de dimensiones o la normalización.

4. **Minería de datos:** Aquí es donde se aplican técnicas y algoritmos de minería de datos para descubrir patrones y relaciones ocultas en los datos. Esto implica la exploración sistemática y automatizada de los datos utilizando algoritmos de clasificación, regresión, agrupamiento, asociación, entre otros.

5. **Evaluación de resultados:** Los resultados obtenidos de la etapa de minería de datos se evalúan para determinar su calidad y relevancia. Se utilizan métricas y técnicas de validación para evaluar la precisión y el rendimiento de los modelos generados.

6. **Interpretación y uso del conocimiento:** Los conocimientos extraídos de los datos se interpretan y se utilizan para tomar decisiones informadas. Esto implica la comunicación de los resultados a través de informes, visualizaciones o modelos predictivos que se pueden utilizar en diferentes ámbitos, como el comercio, la medicina, la seguridad, entre otros.

El proceso KDD se puede aplicar a una amplia gama de proyectos en los que se busca descubrir conocimiento a partir de los datos [Timarán-Pereira *et al.*, 2016]. Algunos ejemplos incluyen la identificación de perfiles de clientes fraudulentos, la exploración de relaciones ocultas entre síntomas y enfermedades, el análisis de características técnicas para diagnosticar el estado de equipos y máquinas, la optimización de procesos empresariales, la recomendación de productos y el descubrimiento de patrones de compra en las canastas de mercado de los clientes.

2.4.2. Metodología SEMMA

SEMMA es un proceso estándar creado por el instituto SAS [Azevedo y Santos, 2008]. El ciclo completo de SEMMA se presenta en la Figura 7, donde se describen brevemente las etapas correspondientes.

1. **Muestreo:** En esta etapa, se selecciona una muestra representativa del conjunto de datos completo. El objetivo es trabajar con una porción de los datos que sea lo suficientemente representativa para realizar análisis y construir modelos.

2. **Exploración:** En este paso, se realiza un análisis exploratorio de los datos. Se utilizan técnicas y herramientas de visualización para comprender la estructura de los datos, identificar patrones, tendencias y relaciones entre las variables. El objetivo es obtener una visión general de los datos y generar hipótesis iniciales.

3. **Modificación:** Se aplican técnicas de limpieza, transformación y manipulación de los datos. Esto puede incluir la eliminación de valores atípicos, la normalización de variables, la creación de nuevas variables derivadas, o cualquier otra transformación necesaria para preparar los datos para el análisis posterior.

4. **Modelado:** Se construyen modelos estadísticos o algoritmos de aprendizaje automático utilizando los datos preparados en la etapa anterior. Estos modelos pueden ser utilizados para realizar predicciones, clasificaciones o descubrir patrones más profundos en los datos. Se selecciona el modelo más adecuado según los objetivos del proyecto.

5. **Evaluación:** En esta última etapa, se evalúa el desempeño de los modelos construidos. Se realiza una validación cruzada o se utiliza un conjunto de datos de prueba para evaluar la precisión y la eficacia del modelo. Además, se realiza un análisis de los resultados obtenidos y se documenta el proceso seguido.

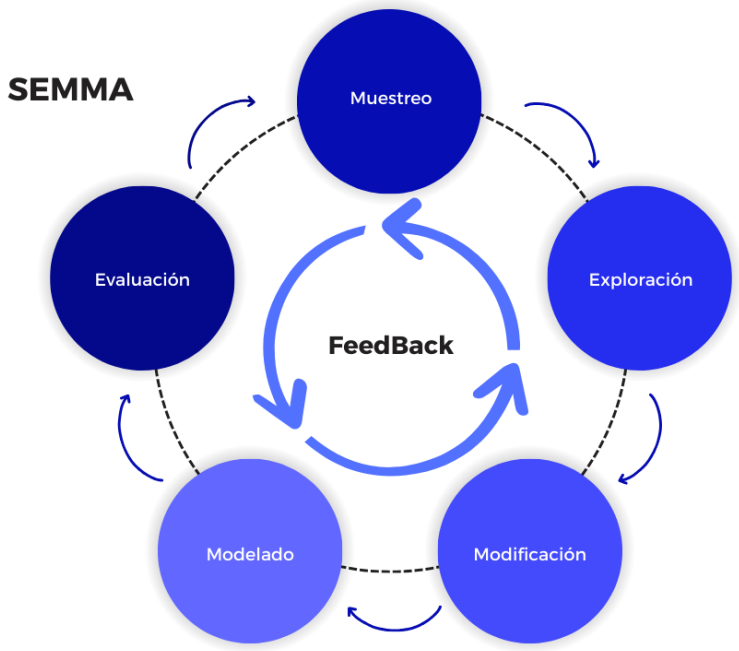


Figura 7: Ciclo SEMMA
Fuente: Elaboración propia

2.4.3. Metodología CRISP-DM

La metodología CRISP-DM proporciona una estructura y guía para el desarrollo de proyectos de ciencia de datos a través de un ciclo de vida flexible compuesto por 6 fases, como se muestra en la Figura 8 [Sngular, 2023]. Es importante destacar que CRISP-DM no es una metodología rígida, sino que puede adaptarse y personalizarse según las necesidades específicas de cada proyecto. Se permite la retroalimentación y la posibilidad de avanzar o retroceder entre las fases, lo que permite ajustes y mejoras en cualquier etapa del proceso para lograr resultados óptimos.

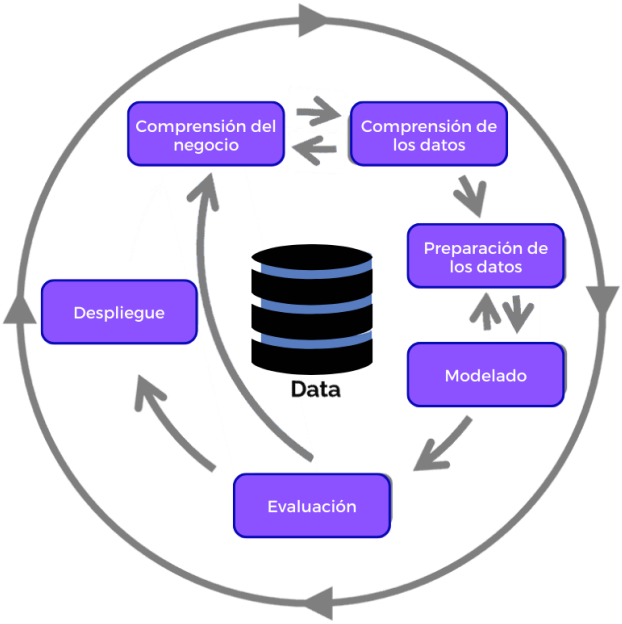


Figura 8: Ciclo CRISP-DM
Fuente: Elaboración Propia

Las etapas de la metodología CRISP-DM son las siguientes:

1. **Comprensión del negocio:** En esta fase, se establecen los objetivos del proyecto y se comprenden los requerimientos y problemas del negocio. Se identifican los factores críticos de éxito y se definen las metas a alcanzar.

2. **Comprensión de los datos:** Se recopilan y exploran los datos disponibles para comprender su estructura, calidad y relevancia para el proyecto. Se realizan tareas de limpieza, integración y selección de datos, y se identifican posibles patrones y relaciones.

3. **Preparación de los datos:** En esta etapa, se transforman los datos en un formato adecuado para su análisis. Se aplican técnicas de limpieza, transformación y selección de variables, y se crean conjuntos de datos preparados para el modelado.

4. **Modelado:** Se seleccionan y aplican técnicas de modelado de datos para construir modelos predictivos o descriptivos. Se utilizan algoritmos y herramientas de minería de datos para crear modelos que capturen patrones y relaciones en los datos.

5. **Evaluación:** Se evalúan y validan los modelos construidos utilizando métricas y técnicas apropiadas. Se verifica la calidad de los modelos y se realiza una evaluación detallada de su rendimiento.

6. **Despliegue:** En esta fase, se implementan los resultados del proyecto en un entorno de producción. Se generan informes, se integran los modelos en sistemas existentes y se lleva a cabo la monitorización y seguimiento de los resultados obtenidos.

CAPÍTULO 3
PROPUESTA DE SOLUCIÓN

En este capítulo, se presenta la propuesta de solución diseñada para abordar el desafío de comprender y analizar el sentimiento de la comunidad sansana en la red social *Instagram* durante tiempos de pandemia. El objetivo central de esta propuesta es satisfacer los objetivos planteados para el desarrollo de esta memoria de investigación.

3.1. Metodología de trabajo

Debido a que el trabajo realizado para esta memoria corresponde a la categoría de proyecto de minería de datos, se propone utilizar el proceso KDD, descrito en la sección 2.4, para llevar a cabo el análisis de sentimiento en los textos recolectados desde *Instagram*. La Figura 9 enseña las etapas de KDD que son aplicadas como parte de la metodología de trabajo en el contexto de la memoria.

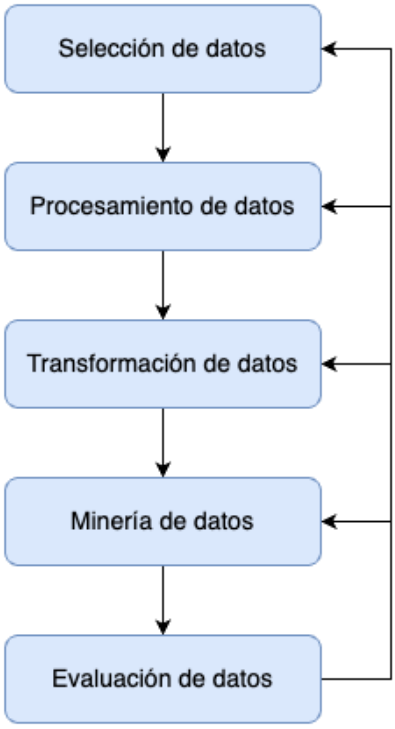


Figura 9: Metodología de trabajo
Fuente: Elaboración propia

Es importante destacar que las primeras tres etapas se abordarán completamente en el presente capítulo de la memoria, mientras que la etapa de minería de datos será dividida en dos

partes, la primera parte se desarrolla en este capítulo y la segunda etapa, en conjunto con la evaluación de resultados, se desarrollarán en el siguiente capítulo.

3.1.1. Selección de los datos

Los datos seleccionados para este trabajo de memoria provienen de la red social *Instagram*. A diferencia de Twitter, donde la información se presenta en formato de texto, en *Instagram* hay imágenes. Por lo tanto, el conjunto de datos objetivo se limita a todas aquellas imágenes subidas a las cuentas de confesiones creadas por estudiantes de la Universidad Técnica Federico Santa María durante el período de la pandemia y las clases en línea, que abarca desde comienzos de marzo de 2020 hasta finales de marzo de 2022. Se considera marzo de 2022 debido a que el tema de la pandemia aún era recurrente en las confesiones en ese momento. La selección de estas confesiones se realiza en función del período temporal relevante para el contexto del problema, lo que suma un total de 4292 imágenes.

La etapa de selección y extracción de los datos, ilustrado en la Tabla 1, se subdivide en 5 actividades. En primer lugar, se realiza la selección de imágenes desde la red social *Instagram*, centrándose en las confesiones publicadas en dos cuentas específicas: @confesiones.usm, con un total de 4256 publicaciones, y @sansano.anonimo, con un total de 36 publicaciones, correspondientes al período del contexto del problema.

Actividad	**Medio**
Selección de imágenes	Cuentas de Instagram
Descarga de imágenes	Biblioteca Instaloader
Obtención de imágenes	Python
Extracción de texto	Tesseract OCR
Obtención de texto	Python

Tabla 1: Actividades etapa de selección y extracción de datos

La segunda actividad implica la descarga de las imágenes seleccionadas, utilizando la biblioteca *Instaloader* [Community, 2023], una herramienta *open-source* desarrollada en *Python*. Esta biblioteca ofrece la capacidad de descargar diversos datos de *Instagram*, como fotos, vídeos, historias, perfiles de usuarios y metadata, ya sea mediante la línea de comandos o código. Para este problema en particular, se aplicaron filtros específicos, como fechas y tipos de contenido, para descargar únicamente imágenes seleccionadas. La posibilidad de personalizar estos filtros es fundamental para obtener los datos que se habían seleccionado en la etapa anterior. En la Tabla 2 se enseñan los filtros utilizados en el algoritmo de descarga de imágenes.

Fecha desde	Fecha hasta	Cuenta 1	Cuenta 2	Contenido
01-03-2020	30-03-2022	@confesiones.usm	@sansano.anonimo	Imágenes

Tabla 2: Filtros aplicados para la descarga de imágenes

Para la tercera actividad se cuenta con un total de 4292 imágenes correspondientes a confesiones; un ejemplo es la Figura 10. También se incluyen imágenes de publicidad, memes y otros tipos de contenido que no se corresponden con los datos objetivo. Debido a que los filtros no permiten seleccionar específicamente las imágenes relacionadas con las confesiones, fue necesario realizar una limpieza manual para eliminar aquellas imágenes que no eran relevantes para el análisis final. El objetivo final es trabajar con archivos de texto, por lo tanto, se requiere extraer el contenido de cada una de las confesiones y almacenarlo en un formato adecuado para su posterior preprocesamiento.

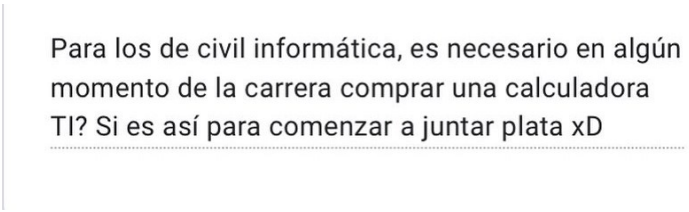


Figura 10: Imagen de confesión extráida desde Instagram
Fuente: @confesiones.usm

Para la cuarta actividad se realiza la extracción de los textos a partir de las imágenes utilizando la biblioteca *Python Tesseract* [Hoffstaetter, 2022]. Esta biblioteca *open-source* proporciona una interfaz para aprovechar el potencial del motor de OCR conocido como *Tesseract OCR* [Tesseract OCR Developers, 2022]. Este motor tiene la capacidad de convertir imágenes con texto en texto editable, lo que resulta fundamental para su posterior procesamiento. Una de las ventajas más destacables de *Tesseract* es su habilidad para detectar y reconocer texto en varios idiomas, incluyendo el español. Esto es especialmente valioso en situaciones en las que trabajamos con imágenes que contienen texto en diferentes idiomas, como en el caso de las confesiones universitarias.

Finalmente, en la quinta actividad se obtiene el texto correspondiente a cada confesión. En algunas confesiones es posible detectar información adicional, como la fecha y hora en que se envió a través del formulario de *Google*, o la sede asociada a la confesión. Toda esta información se guarda en un archivo CSV para su posterior preprocesamiento y análisis. El archivo tiene una estructura compuesta por seis columnas, las cuales se describen a continuación:

- **id:** id de la confesión.
- **id_2:** id secundario creado a partir de date_ig_utc.

- **date_ig_utc:** fecha y hora de subida de la confesión a *Instagram*.
- **campus:** campus asociado a la confesión.
- **date_submitted_aaaa_mm_dd:** fecha y hora de envío de la confesión por formulario de Google.
- **text:** texto de la confesión.

Durante las siguientes etapas del proceso, es de vital importancia contar con una manipulación efectiva de los datos. Para lograrlo, se lleva a cabo la transferencia de la información contenida en el archivo CSV, que almacena los datos en un formato tabular, luego se importa en *Python* y se crea un *dataframe* de Pandas que permitirá una manipulación más eficiente de los datos. Este paso es fundamental para garantizar que los datos estén listos y adecuados para su procesamiento en las etapas posteriores del proceso.

3.1.2. Preprocesamiento de datos

La etapa de preprocesamiento juega un papel fundamental al asegurar una buena calidad de preparación de los datos, lo cual repercute de forma significativa en la obtención de resultados fiables y precisos en las etapas de transformación de datos y minería de datos. Durante esta etapa, se lleva a cabo una minuciosa operación de limpieza de los datos. Se comienza con la normalización de todo el texto a letras minúsculas, de esta forma es posible facilitar la coincidencia de palabras clave y evita inconsistencias como diferencias de mayúsculas y minúsculas al buscar palabras clave o realizar operaciones de coincidencia de patrones.

El uso de *emojis* en las redes sociales se ha vuelto una forma popular y efectiva de comunicación. Estos símbolos visuales permiten expresar emociones, transmitir ideas y añadir un toque personal a los mensajes o comentarios en las redes sociales. Los *emojis* ofrecen una manera rápida y concisa de comunicar sentimientos, ya que a menudo pueden transmitir en un solo símbolo lo que tomaría varias palabras describir. Por ejemplo, un *emoji* sonriente (😀) puede indicar felicidad, mientras que un *emoji* triste (☹) puede transmitir tristeza. Estos símbolos permiten capturar la emoción detrás del mensaje y facilitan la interpretación del tono y el contexto de esta. Para el proceso de extracción de los textos a partir de las imágenes de confesiones, se utilizó *Tesseract* como herramienta OCR. *Tesseract* está diseñado principalmente para reconocer y convertir texto escrito en caracteres alfanuméricos y algunos caracteres especiales. Sin embargo, los *emojis* son símbolos visuales más complejos y están fuera del alcance de la capacidad de reconocimiento de caracteres de *Tesseract*. Como resultado, al extraer el texto de imágenes que contienen *emojis*, es común que aparezcan caracteres alfanuméricos en lugar de los símbolos visuales originales. Este fenómeno puede afectar la calidad y la comprensión del texto resultante, ya que los *emojis* aportan un contexto emocional y expresivo que los caracteres alfanuméricos no logran transmitir de la misma

manera. Para evitar este problema, se limpiaron los textos extraídos, aplicando funciones específicas diseñadas para eliminar todos los caracteres alfanuméricos, números y palabras de una sola letra. Al eliminar estos elementos, se busca obtener un texto más limpio y legible, eliminando los caracteres no deseados que pueden haberse generado debido a las limitaciones del reconocimiento de *emojis*. Esta etapa de limpieza ayuda a garantizar que el texto final sea más coherente y comprensible, centrándose en la esencia y el contenido relevante del mensaje original, sin la interferencia de caracteres no deseados.

Cuando los sansanos escriben sus confesiones, suelen utilizar un lenguaje informal lleno de modismos chilenos. Estos modismos, aunque le dan un toque auténtico y local a sus mensajes, pueden ser un desafío para los modelos de análisis de sentimiento. Muchos de estos modismos no son comprendidos por completo por los modelos de procesamiento de lenguaje natural, dado que no son entrenados con esta clase de expresiones, lo que puede afectar la precisión en la interpretación de las emociones y sentimientos transmitidos. Por lo tanto, es importante considerar esta propiedad lingüística presente en los textos extraídos, y ajustar los algoritmos y modelos para capturar adecuadamente la intención y el tono detrás de las confesiones en el contexto cultural específico de los sansanos. La solución para abordar el desafío de los modismos en los textos de confesiones es reemplazar las palabras específicas que pueden causar problemas en la interpretación que puedan asignar los modelos de análisis de sentimiento. Palabras como “mechón”, “sansano”, “pololo(a)”, “u”, “ap”, “profe”, “apañar” y entre varias más, pueden ser sustituidas por términos más generales y comprensibles para los modelos. Al realizar estos reemplazos mediante la búsqueda de palabras especificas (primer paso del preprocesamiento), se busca que los modelos puedan captar de manera más precisa la intención emocional del mensaje y brindar una interpretación adecuada.

A modo de resumen, la Tabla 3 ejemplifica cada paso del preprocesamiento, mencionados previamente, a través de una confesión real enviada a la cuenta @confesiones.usm. Es importante destacar el paso de **Reemplazo**, el cual implica el cambiar palabras clave, para este ejemplo se sustituyeron las siguientes:

- cabros $\rightarrow$ chicos
- mechon $\rightarrow$ novato
- plata $\rightarrow$ dinero

El texto final que se proporciona a cada modelo después del preprocesamiento es el resultado de aplicar los diferentes pasos de extracción, normalización a minúsculas, limpieza y reemplazo.

Texto original	Cabros soy mechon y aun no me llega la plata del cae, tipo ya firme y todo se supone, a alguien le ha llegado algo o no?? 🥺🥺🥺
Texto extraído	Cabros soy mechon y aun no me llega la plata del cae, tipo ya firme y todo se supone, a alguien le ha llegado algo o no?? ° W * +
Minúsculas	cabros soy mechon y aun no me llega la plata del cae, tipo ya firme y todo se supone, a alguien le ha llegado algo o no?? ° w * +
Limpieza	cabros soy un mechon aun no me llega la plata del cae tipo ya firme todo se supone alguien le ha llegado algo no
Reemplazo	chicos soy un novato aun no me llega la dinero del cae tipo ya firme todo se supone alguien le ha llegado algo no

Tabla 3: Ejemplo de aplicación de los pasos de un Procesamiento de Datos

Finalizando al etapa de preprocesamiento, se cuenta con un *dataframe* de Pandas que consta de seis columnas con información relacionada a la confesión, tal como se muestra en la Tabla 4.

Id	Id2	date_ig_utc	campus	datesub	text
1	202003100544140	2020-03-10_05-44-14_UTC	NaN	NaN	que hago si quiero participar del centro de al...
2	202003100544320	2020-03-10_05-44-32_UTC	NaN	NaN	katha con th de ingeniería comercial me tienes...
3	202003100544470	2020-03-10_05-44-47_UTC	NaN	NaN	habrá algún estudiante que le guste la música ...

Tabla 4: Dataframe previo a la transformación

3.1.3. Transformación de datos

Durante la etapa de transformación de datos, se llevó a cabo una importante tarea de reducción de dimensionalidad al eliminar las columnas adicionales del *dataframe*. Estas columnas, que carecían de relevancia para el análisis de sentimiento, fueron eliminadas para enfocar el proceso en la columna de texto, que contiene la información central para el análisis. Para lograr esto, se utilizaron métodos de la biblioteca Pandas, reconocida por su especialización en el procesamiento y manipulación de datos. Mediante el uso de Pandas, se procedió a eliminar las columnas del *dataframe* que no ofrecían información útil para el análisis de sentimiento, como aquellas relacionadas con identificadores, fechas y campus asociados a las confesiones. Este proceso de reducción del *dataframe* permite trabajar de manera más precisa y eficiente con los modelos de análisis de sentimientos que se emplearon en la siguiente etapa de minería de datos.

Para llevar a cabo el análisis, únicamente se necesitan las columnas 'Id' y 'Text', mientras que las columnas relacionadas con fechas, la columna 'campus' e 'Id2' no brindan información útil. Por lo tanto, se realiza una reducción de dimensionalidad para eliminar estos atributos que carecen de relevancia para el análisis. Una vez concluida la etapa de transformación, se obtiene un *dataframe* con una estructura similar a la mostrada en la Tabla 5.

Id	text
1	que hago si quiero participar del centro de al...
2	katha con th de ingeniería comercial me tienes...
3	habrá algun estudiante que le guste la música ...

Tabla 5: Dataframe posterior a la transformación

3.1.4. Minería de datos

La etapa de minería de datos se enfoca en extraer conocimiento valioso de conjuntos de datos extensos. En el contexto del análisis de sentimientos, la minería de datos implica el uso de técnicas y algoritmos para descubrir patrones, relaciones y estructuras ocultas en los datos textuales. El objetivo principal es comprender y analizar las opiniones, actitudes y sentimientos expresados por los sansanos en las confesiones enviadas a las cuentas de *Instagram*. Esto proporciona una perspectiva más profunda sobre las percepciones de los estudiantes de la UTFSM durante el período de pandemia y clases en línea. Mediante esta exploración, se busca obtener una comprensión más completa y significativa de la experiencia y emociones de la comunidad en este contexto específico.

La metodología utilizada en la etapa de minería de datos consiste en una evaluación inicial de los datos mediante modelos de clasificación de sentimientos que permiten determinar la polaridad de los textos, en este caso, se emplean tres modelos diferentes: *Vader*, *PySentimiento y TextBlob*. La utilización de estos tres modelos se justifica debido a que cada uno tiene sus propias ventajas y enfoques, mencionadas en la sección 2.3.5. Posteriormente, se realiza una comparación de los resultados obtenidos entre los tres modelos. Se evalúa la polaridad que cada uno de ellos asigna a una misma confesión y se implementa un algoritmo de votación para determinar el valor de sentimiento predominante para cada confesión, basándose en los resultados de la votación. A partir de esta evaluación, se selecciona uno de los tres modelos para su posterior uso.

Una vez seleccionado el modelo, se continúa con un proceso de extracción de características del texto de las confesiones. Esta extracción de características permite obtener información relevante sobre cada confesión y sus características emocionales. Luego, se aplican técnicas de *clustering* sobre el conjunto de datos para agrupar las confesiones según su similitud en términos de sentimiento y contenido emocional. De esta manera, se obtiene una percepción clara de cómo se comportan las agrupaciones de confesiones en función de sus emociones expresadas.

Finalmente, se lleva a cabo la visualización y análisis de los resultados obtenidos. Mediante herramientas de visualización adecuadas, se presentan los grupos de confesiones identificados en mediante clústeres, lo que facilita la comprensión de los patrones emocionales y temáticos presentes en la comunidad sansana durante el período de pandemia y clases en línea. Estas visualizaciones y análisis permiten una mejor interpretación de los datos recopila-

dos y brindan una perspectiva más profunda sobre las experiencias emocionales compartidas por la comunidad.

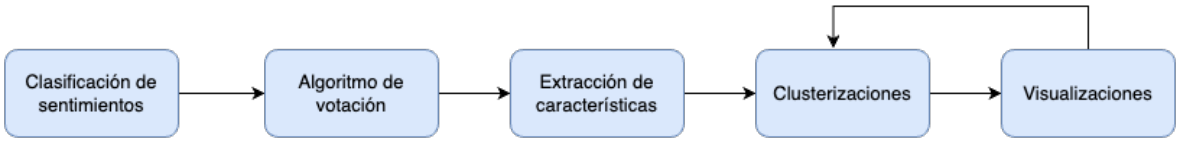


Figura 11: Actividades de la etapa minería de datos
Fuente: Elaboración propia

Profundizando en cada etapa mencionada anteriormente. En primer lugar, con el objetivo de evaluar el tono emocional de los textos de confesiones, se emplearon tres modelos previamente mencionados en el capítulo 1.2.2: *TextBlob*, *Vader y PySentimiento*. Estos modelos permiten evaluar la polaridad de los textos utilizando diversos enfoques y técnicas de clasificación, además de presentar sus propias ventajas, lo que brinda una perspectiva más amplia para el análisis.

Para el primer modelo de análisis de sentimiento se utilizó el proporcionado por la biblioteca *TextBlob*. Dado que este modelo está entrenado en inglés, se requiere realizar una traducción previa del texto para poder llevar a cabo el análisis. *TextBlob* ofrece una función de traducción, por lo tanto, se utiliza dicho método para traducir el texto original y luego se aplica el analizador sobre el texto traducido. El resultado obtenido es un valor numérico para cada texto, con un rango entre -1 y 1, donde -1 representa un sentimiento negativo y 1 un sentimiento positivo. Finalmente, los resultados se almacenan en un *dataframe* con columnas como 'ID', 'Text' y 'SENT', donde 'SENT' indica el sentimiento predominante determinado por el modelo. Al igual que *PySentimiento*, *TextBlob* toma cierto tiempo en realizar el análisis, dado que primero realiza el paso de traducir cada texto, por este motivo se guardó la estructura de datos resultante en un archivo .csv para usos posteriores.

El segundo modelo es *Vader*. Aunque está diseñado para analizar textos en inglés, ofrece una solución para tratar con textos en otros idiomas. Sin embargo, esta solución es externa a la biblioteca en sí. Por lo tanto, se decidió utilizar la funcionalidad de traducción proporcionada por *TextBlob*. De esta manera, es posible mantener la consistencia en la traducción para un mismo texto, ya que se utiliza la misma traducción tanto para *TextBlob* como para *Vader*. Esto facilita la detección de las diferencias de asignación entre los modelos. El resultado del análisis es un valor compuesto obtenido de la normalización de los puntajes de positividad, negatividad y neutralidad. Se considera que un valor compuesto mayor o igual a 0.05 indica un sentimiento positivo, un valor menor a -0.05 indica un sentimiento negativo, y los valores en el rango entre -0.05 y 0.05 se asignan como neutrales. Con los resultados, se genera un *dataframe* con tres columnas: 'ID', 'Text' y 'SENT'. Por último, está estructura de datos se guarda en un archivo .csv para su posterior uso.

El tercer modelo utilizado es *PySentimiento* debido a su capacidad para trabajar con textos

de redes sociales en español. En primera instancia, se llamó al método de análisis, indicando el idioma a analizar como parámetro, y proporcionando cada una de las confesiones almacenadas en el *dataframe*. Como resultado, se obtuvo un diccionario con las probabilidades de que cada texto fuese positivo, negativo o neutro, junto con el sentimiento con mayor probabilidad. A partir de este diccionario, se reestructuró un nuevo *dataframe* que incluía las columnas de 'ID', 'Text', 'NEG', 'NEU', 'POS' y 'SENT'. Es importante mencionar que la ejecución del modelo para analizar los textos presentó cierta lentitud, por lo que se decidió guardar el *dataframe* resultante del análisis en un archivo .csv para su posterior visualización o análisis estadístico.

Una vez obtenidos los resultados de las evaluaciones de clasificación de sentimiento, se procedió a realizar una exhaustiva comparación entre los valores asignados por cada modelo a cada texto. Esta comparación entregó una visión clara del rendimiento individual de cada modelo y permite identificar las confesiones en las cuales los tres coinciden unánimemente en la evaluación del sentimiento. Asimismo, se buscó detectar las discrepancias y revisar los casos donde existe una mayoría de acuerdo entre los modelos. Posteriormente, se seleccionó el modelo que mejor se adapte y funcione en base a la evaluación de los resultados obtenidos para abordar este problema. Esta meticulosa evaluación se realizó con el propósito de obtener conclusiones precisas y fundamentadas, es por esto que se establecen los siguientes conceptos fundamentales que guían todo el proceso de comparación y selección:

- **Total de consensos:** se refiere al número total de ocasiones en las que los tres modelos están de acuerdo en la clasificación o valoración del sentimiento para ese texto específico. En otras palabras, cada vez que los tres modelos coinciden en asignar el mismo valor de sentimiento al texto, se cuenta como un *consenso*.

  Considerar que el total de consensos no necesariamente implica que la clasificación o valoración del sentimiento sea correcta o precisa, ya que los modelos podrían estar de acuerdo en una clasificación incorrecta. Sin embargo, un alto total de consensos puede sugerir una mayor confianza en los resultados obtenidos.

- **Total de acuerdos por mayoría:** se refiere al número de ocasiones en la que, al comparar las clasificaciones de sentimiento de los tres modelos para un mismo texto, al menos dos de ellos coinciden en una clasificación particular. En otras palabras, se considera un *acuerdo por mayoría* cuando la mayoría de los modelos están de acuerdo en una misma clasificación de sentimiento.

  Es necesario destacar que este valor no garantiza necesariamente la precisión absoluta, ya que el modelo restante podría tener una clasificación diferente, y es posible que esa clasificación alternativa sea la correcta. Por lo tanto, el acuerdo por mayoría es una medida a considerar en el análisis, pero no es concluyente por sí sola.

- **Total de desacuerdos:** se refiere al número total de veces en las que los 3 modelos no están de acuerdo en la clasificación o valoración del sentimiento para un mismo texto. Es decir, cada vez que al menos uno de los modelos asigna un valor de sentimiento diferente a los otros modelos, se considera un *desacuerdo*.

Es importante tener en cuenta que los desacuerdos no necesariamente indican que una clasificación es correcta y la otra incorrecta. Puede haber casos en los que los diversos modelos ofrezcan diferentes perspectivas válidas sobre el sentimiento del texto. Sin embargo, un gran total de desacuerdos puede sugerir una mayor variabilidad y falta de consistencia en los resultados obtenidos.

El algoritmo de votación es sencillo: toma como entrada las columnas de ID y sentimiento para *PySentimiento*, *Vader y TextBlob*. A partir de estos datos, se genera un *dataframe* que muestra claramente las votaciones de cada modelo para cada ID de confesión. Luego, se procede a iterar por cada fila y comparar los tres valores obtenidos en la etapa anterior para cada texto. Si los tres modelos coinciden en la clasificación, se considera un *consenso* y se almacena el sentimiento en la columna destinada para los votos. En caso de que dos modelos voten por un mismo valor de sentimiento y el tercero difiera, esto se denomina *acuerdo por mayoría*, y se guarda el valor del sentimiento que presente una mayoría. Si ninguno de los tres modelos está de acuerdo en la clasificación, se almacena un 1. Para cada caso se tiene una variable que contabiliza la cantidad de consensos, mayoría y desacuerdos.

Algoritmo 1: Algoritmo votación

Entrada: Arreglo con los ID y los sentimientos de pysent, tblob y vader
Salida : Dataframe con los votos

$consensos \leftarrow 0$;
$mayoria \leftarrow 0$;
$desacuerdos \leftarrow 0$;
$resultado \leftarrow [\,]$;
$votos \leftarrow df[id, pysent, tblob, vader, resultado]$;
para $row \in votos$ **hacer**
    $id, ps, tb, vd, res \leftarrow row$;
    si $ps == tb == vd$ **entonces** `// consenso`
        $row['res'] \leftarrow ps$;
        $consensos+ = 1$;
    si no, si $TotalIguales(ps, tb, vd) == 2$ **entonces** `// mayoría`
        $row['res'] \leftarrow Mayoria(ps, tb, vd)$;
        $mayoria+ = 1$;
    en otro caso `// desacuerdo`
        $row['res'] \leftarrow 1$;
        $desacuerdos+ = 1$;
    fin
fin

Como resultado del algoritmo, se obtiene el *dataframe* votos presentado en la Tabla 6. El resumen de los resultados serán expuestos y discutidos en el próximo capítulo.

ID	TB	PS	VD	Resultado
1	NEG	NEU	POS	1
2	POS	POS	POS	POS
3	POS	NEU	POS	POS
4	POS	NEG	NEU	1
5	NEG	NEG	NEG	NEG
...	...	...	...	...
4288	POS	NEG	NEG	NEG
4289	NEG	NEU	NEU	NEU
4290	NEU	NEU	POS	NEU
4291	POS	POS	POS	POS
4292	NEG	NEG	POS	NEG

Tabla 6: Dataframe resultante de la votación

Los próximos tres pasos en este proceso abarcan la extracción de características, los algoritmos de clúster y la visualización de datos. Para justificar las decisiones tomadas en estas etapas posteriores, es esencial basarse en los resultados proporcionados por el algoritmo de votación. En el próximo capítulo, se llevará a cabo el análisis de los resultados de la fase de minería de datos, lo que permitirá realizar una evaluación exhaustiva de los mismos.

CAPÍTULO 4
ANÁLISIS DE RESULTADOS Y VALIDACIÓN

En el presente capítulo, se llevan a cabo análisis de resultados y validación. Se revisan las clasificaciones de sentimientos obtenidas, se evalúa el rendimiento de los modelos y se profundiza en la extracción de características. Además, se exploran los sentimientos y las emociones desglozadas a través de visualizaciones. Se discute sobre la reducción de dimensionalidad, se examinan y comparan los grupos de datos mediante diversos agrupamientos. Finalmente se validan los hallazgos con coeficientes de clusterización.

4.1. Análisis de la Clasificación de Sentimientos Obtenida por Modelo

Luego de aplicar los tres modelos de análisis de sentimiento al conjunto de confesiones que componen el conjunto de datos, los resultados presentados en la Figura 12 muestran una similitud en la clasificación para los sentimientos positivos, negativos y neutros, entre *TextBlob y Vader*.

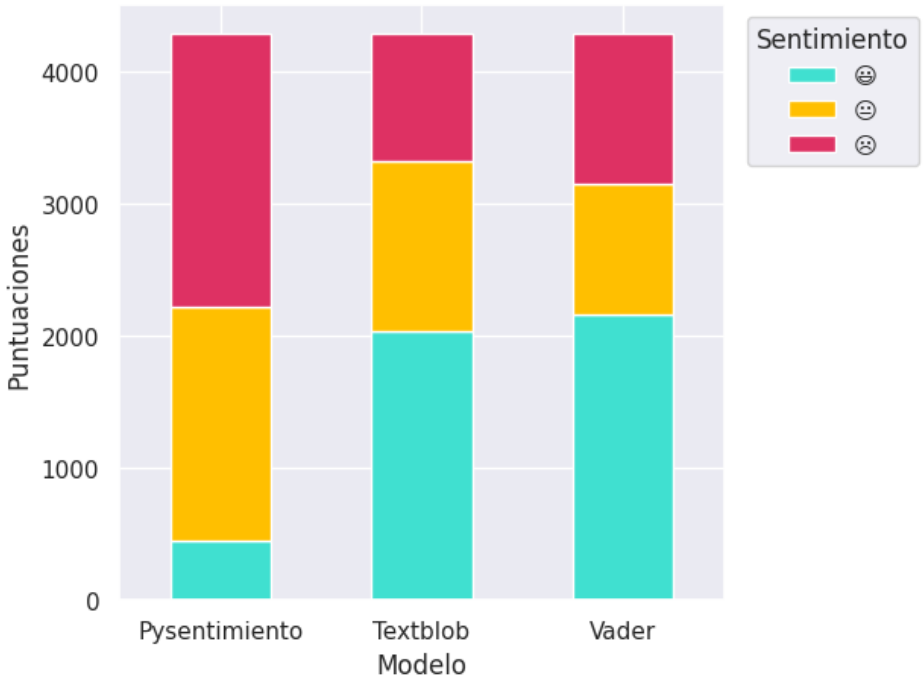


Figura 12: Comparación de clasificación de sentimientos entre modelos
Fuente: Elaboración propia

Ambos modelos muestran una paridad en términos de cómo etiquetan cada tipo de sentimiento. Detallando las puntuaciones obtenidas por cada uno de los modelos, los resultados son los siguientes: *PySentimiento* clasificó 449 confesiones como positivas, 1772 como neutras y 2071 como negativas. Por su parte, *TextBlob* arrojó 2032 clasificaciones positivas, 1296 neutras y 964 negativas. En el caso de *Vader* , se identificaron 2161 como positivas, 992 como neutras y 1139 como negativas. Estos valores encapsulan las interpretaciones de cada modelo sobre el tono emocional de las confesiones analizadas.

PySentimiento presenta diferencias significativas en sus clasificaciones en comparación con los otros dos modelos. Sus evaluaciones de los sentimientos en las confesiones divergen sustancialmente en algunos casos. Esta disparidad evidencia la divergencia en la forma en que *PySentimiento* interpreta y asigna sentimientos en relación con *TextBlolb* y *Vader* .

Este análisis destaca la importancia crucial de emplear distintos modelos al clasificar el sentimiento en textos sin previa etiqueta. Es especialmente relevante al tratar con textos en español u otros idiomas distintos del inglés, ya que el inglés cuenta con un mayor número de desarrollos e investigaciones avanzadas relacionadas a la comprensión de este idioma. Considerar diversos métodos de clasificación no solo permite verificar la concordancia entre los modelos, sino también detectar las posibles diferencias y, en última instancia, llegar a una conclusión sólida e informada respecto a la clasificación final de cada texto.

Aunque TextBlob y *Vader* tienden a mostrar más consistencia en sus resultados, las clasificaciones variables de *PySentimiento* sugieren que interpretar el sentimiento en textos en español puede ser más complicado, además de resaltar la necesidad de desarrollar más modelos de aprendizaje profundo entrenados con textos en español.

4.2. Análisis de Resultados Obtenidos por Algoritmo de Votación

Con el objetivo de analizar el rendimiento de los tres modelos utilizados en la clasificación de sentimientos, se lleva a cabo una comparación empleando el algoritmo de votación descrito en la sección anterior. Los resultados totales de los consensos, acuerdos por mayoría y los desacuerdos obtenidos mediante la votación se presentan en la Tabla 7.

Tipo	Total	Porcentaje
Consensos	1218 de 4292	28.37 %
Acuerdos	3872 de 4292	90.21 %
Desacuerdos	420 de 4292	9.78 %

Tabla 7: Resultados obtenidos con el Algoritmo de Votación

A partir de los resultados obtenidos es posible resaltar los siguientes puntos:

- **El porcentaje de consensos alcanza el 28.37 %**. Este valor da un indicio bastante positivo, ya que equivale a un poco más de la cuarta parte de las confesiones evaluadas. Esto significa que en estos casos los tres modelos coincidieron en la clasificación del sentimiento para un mismo texto.
- **Los acuerdos por mayoría son el aspecto más notable de la votación**. Con un valor que alcanza el 90.21 %, este resultado refleja que en la gran mayoría de los casos al menos dos de los modelos estuvieron de acuerdo en asignar el mismo sentimiento a la misma confesión. Es importante mencionar que, según su definición entregada en el capítulo anterior, los acuerdos por mayoría abarcan los casos de consenso. Por ende, al no considerar los consensos, el número total de acuerdos por mayoría en los cuales dos modelos concuerdan en el sentimiento mientras uno discrepa es de 2654 casos.
- **Aproximadamente un décimo de los casos son desacuerdos**. Solamente el 9.78 % de los casos muestran desacuerdos totales en la clasificación del sentimiento para un mismo texto. A pesar de ser una proporción menor que puede sugerir un buen desempeño por parte de los modelos según los resultados obtenidos del algoritmo de votación, es importante revisar en mayor profundidad aquellos casos de discrepancias.

El caso de los acuerdos por mayoría es de interés para el análisis, dado que el algoritmo de votación permite determinar aquellos casos donde solo dos modelos estuvieron de acuerdo, mientras que el tercero discrepa. Al comparar los modelos en pares, se evidencia una notoria afinidad entre estos al momento de votar por un mismo sentimiento. En la Tabla 13 se detalla la cantidad de acuerdos por mayoría logrados por cada par de modelos.

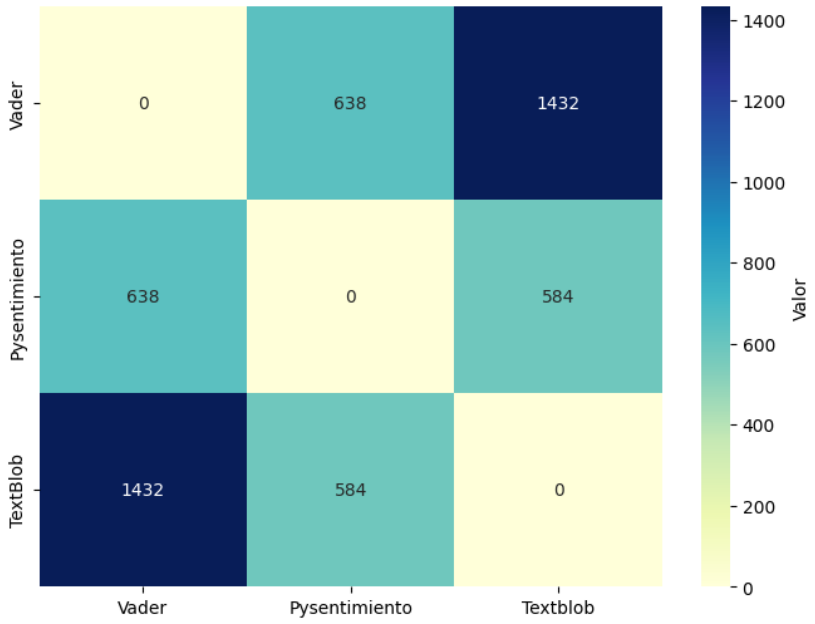


Figura 13: Acuerdos por mayoría entre pares de modelos
Fuente: Elaboración propia

De estos resultados es destacable el total de acuerdos por mayoría alcanzados entre *Vader* y *TextBlob*, equivalente al 53.95 % de las votaciones para este escenario. Por otra parte, se tiene que la menor cantidad de acuerdos por mayoría es entre *PySentimiento* y *TextBlob*, alcanzando un 22 % de los casos.

Desarrollando el punto sobre la alta tasa de acuerdos por mayoría entre *Vader* y *TextBlob*, cabe destacar que ambos modelos están entrenados con textos en ingles, por otra parte, para utilizar ambos modelos se trabajó con el método de traducción que incluye la biblioteca *TextBlob*, lo que puede justificar en parte la afinidad que se obtuvo entre ambos modelos.

Ampliando el punto sobre la alta coincidencia entre *Vader* y *TextBlob*, esto se debe en parte a que ambos modelos se entrenaron con textos en inglés, mientras que las confesiones analizadas en esta investigación están en español en su forma original, con modismos, jergas y faltas de ortografía. Para trabajar con ambos modelos, se aplicó el método de traducción de la biblioteca *TextBlob*. Esto podría haber influido en la coincidencia al proporcionar una interpretación uniforme del contenido original en otro idioma. Sin embargo, es esencial enfatizar esta traducción podría haber excluido muchas palabras sin una equivalencia directa en inglés, un ejemplo de esto son los garabatos y groserías, las cuales en español ofrecen contexto en el mensaje que se intenta expresar.

En este punto de la etapa de minería de datos, se dispone de una clasificación de sentimiento con 3872 confesiones en las que los modelos coinciden mayoritariamente en la clase de sentimiento asignada para un mismo texto. Esta cifra equivale a un porcentaje considerable de las confesiones analizadas. Es por esto que, para las siguientes actividades de la etapa de minería de datos y la etapa de evaluación se opta por trabajar con este subconjunto de los datos originales. Esta decisión está fundamentada en la premisa de que estas confesiones en las que existe consenso brindan una base robusta y coherente para continuar el análisis de sentimientos.

Para determinar qué tan preciso es cada modelo, se comparan las clasificaciones que cada uno de ellos realizó con el resultado que se obtiene a través del algoritmo de votación. De esta manera, se calcula el porcentaje de veces en que cada modelo acierta al predecir el mismo resultado que la votación. Los resultados se enseñan en la Tabla 8.

Modelo	**Aciertos**	**Porcentaje de Aciertos**
Vader	3288 de 3872	84.92 %
Pysent	2440 de 3872	63.02 %
TextBlob	3234 de 3872	83.52 %

Tabla 8: Evaluación de la precisión de los modelos en comparación con la votación

Los resultados obtenidos señalan que tanto *Vader* como *TextBlob* alcanzan los porcentajes más altos de aciertos en relación a los resultados finales de la votación, destacándose en

conseguir una buena precisión de sus predicciones. Por otra parte, *PySentimiento* alcanza un porcentaje de aciertos más bajo en comparación con los otros dos modelos. La similitud entre los niveles de aciertos entre *TextBlob* y *Vader* puede atribuirse a lo previamente comentado, ambos modelos están entrenados en inglés y comparten el mismo traductor proporcionado por *TextBlob*. Dada esta similitud en la clasificación, se crea una dinámica en la que se forma una mayoría durante el proceso de votación, lo que relega a *PySentimiento* a una posición minoritaria. Este fenómeno está dirigido por la similitud entre los enfoques de clasificación de sentimiento entre *TextBlob* y *Vader* , además se ve potenciada por su entrenamiento en un idioma común y el uso del mismo recurso de traducción. Esto resulta en una preeminencia en el resultado de la votación en contraste con las clasificaciones divergentes de *PySentimiento*.

Luego de este análisis del algoritmo de votación y el rendimiento de los modelos, se tomó la decisión de continuar el análisis de sentimiento utilizando *PySentimiento* . Esta elección está respaldada por varios factores fundamentales que destacan su utilidad en el contexto de este problema.

En primer lugar, el hecho de que *PySentimiento* esté entrenado en español lo convierte en una opción sumamente adecuada para abordar el análisis de sentimiento en textos en este idioma. Esto asegura una comprensión sólida de los sentimientos expresados.

Además, es relevante destacar que, dado que el conjunto de datos original no está etiquetada y los modelos previamente probados no garantizan una clasificación precisa, se decidió por continuar la etapa de minería de datos con un enfoque de aprendizaje no supervisado. En este sentido, *PySentimiento* posee una tarea de NLP que permite desglosar las emociones contenidas en el texto. Dado que es un descriptor sólido del contenido textual, se utiliza para extraer características y llevar a cabo agrupaciones que ayuden a profundizar en el análisis.

Como se mencionó anteriormente, dado el desafío de la falta de etiquetas, y la incertidumbre en la clasificación precisa, se optó por aprovechar las capacidades de *PySentimiento* para descomponer las emociones contenidas en los textos, aportando así una base sólida para el análisis de aprendizaje no supervisado y la extracción de características relevantes.

En este contexto, se continuó trabajando sobre una submuestra de la data que ha sido identificada como aciertos por mayoría. Esta submuestra consiste en 2440 confesiones, que corresponden a aquellos casos en los que *PySentimiento* y el algoritmo de votación coinciden con el sentimiento final. Esta decisión se toma con el objetivo de enfocar el análisis en aquellos textos donde existe mayor concordancia entre los enfoques y, por ende, se espera una mayor confiabilidad en las características y patrones extraídos.

4.3. Extracción de Características

En la extracción de características se aprovecha la clasificación de sentimientos previamente realizada mediante el modelo *PySentimiento* en aquella etapa. Tras filtrar las confesiones,

se reduce la muestra a un total de 2440 registros. Para la creación del *dataframe* definitivo, destinado a los métodos de aprendizaje no supervisado, se combina la clasificación de sentimientos, que incluye el almacenamiento de las probabilidades de clasificación para cada sentimiento, y la clasificación de emociones, que añade la probabilidad correspondiente a la presencia de cada emoción en los textos analizados.

Para lograrlo, se utiliza el método de creación de un analizador, al cual se le suministra el tipo de tarea: en este contexto, la identificación de emociones. Asimismo, se especifica el idioma requerido, que es el español.

Las emociones que el analizador descompone son enlistadas a continuación:

1. **Joy:** Emoción de alegría.
2. **Sadness:** Emoción de tristeza.
3. **Anger:** Emoción de enojo.
4. **Surprise:** Emoción de sorpresa.
5. **Disgust:** Emoción de disgusto.
6. **Fear:** Emoción de miedo.
7. **Others:** Otras emociones.

El resultado final es un desglose de probabilidades de diversas emociones y sentimientos presentes en cada texto. La Tabla 9 se representa el *dataframe* que se utilizó para aplicar diversos métodos y algoritmos de aprendizaje no supervisado. Es importante resaltar que se agregan dos columnas adicionales, 'SENT' y 'EMOT', para destacar los valores finales de clasificación tanto para el sentimiento como para la emoción.

ID	Text	NEG	NEU	POS	sadness	fear	disgust	surprise	joy	anger	others	SENT	EMOT
0	...	0.047	0.179	0.774	0.008	0.044	0.010	0.193	0.191	0.009	0.545	POS	others
1	...	0.963	0.035	0.002	0.004	0.004	0.003	0.008	0.000	0.005	0.975	NEG	others
2	...	0.966	0.029	0.005	0.010	0.001	0.034	0.001	0.001	0.935	0.019	NEG	anger
3	...	0.078	0.868	0.054	0.005	0.007	0.002	0.015	0.013	0.001	0.955	NEU	others
4	...	0.014	0.083	0.903	0.002	0.002	0.001	0.009	0.055	0.001	0.928	POS	others
...	...	...	...	...	...	...	...	...	...	...	...	...	...

Tabla 9: Extracción de características mediante clasificación de sentimientos y emociones

4.4. Análisis de Sentimientos y Emociones Mediante Visualizaciones

Para realizar una exploración inicial y análisis de los datos, se emplean diversas visualizaciones que se presentan en esta sección.

La Figura 14 presenta la distribución de la clasificación de sentimientos en el conjunto de datos. El sentimiento negativo es predominante, con un total de 1145 ocurrencias, lo que equivale al 46.9 % de la muestra. Le sigue el sentimiento neutro con 902 ocurrencias, representando el 37 % de la muestra. En menor proporción, se encuentran los casos positivos, con 393 ocurrencias, equivalente al 16.1 % de los datos.

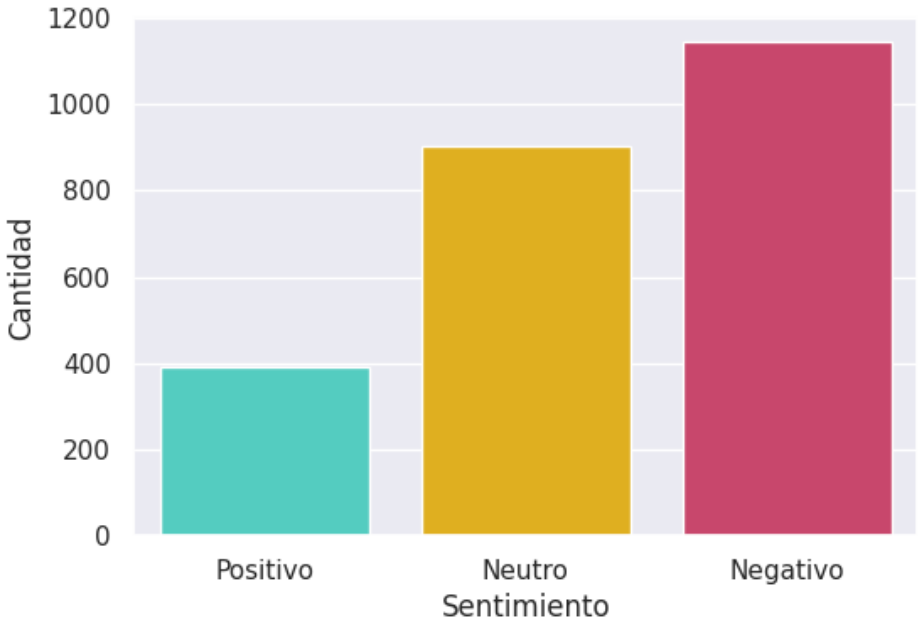


Figura 14: Distribución de sentimientos
Fuente: Elaboración propia

En cuanto a las emociones, la Figura 15 muestra la distribución de las emociones para la muestra. La categoría *"others"* es la emoción más predominante, con un total de 1568 ocurrencias, equivalente al 78.4 % del conjunto de datos. La emoción *"anger"* sigue en frecuencia, presentándose en 486 ocasiones, lo que representa el 24.3 % de las muestras. Las emociones *"sadness"* y *"joy"* tienen una proporción de alrededor del 8.8 %, unas 176 ocurrencias, y 8 %, es decir unas 160 ocurrencias, respectivamente. Por otro lado, las emociones menos comunes son *"fear"* y *"surprise"*, con un valor igual 32 (1.6 %), y 16 (0.8 %), respectivamente. Finalmente, la emoción *"disgust"* es la menos registrada, con apenas 2 casos, lo que equivale al 0.1 % del total.

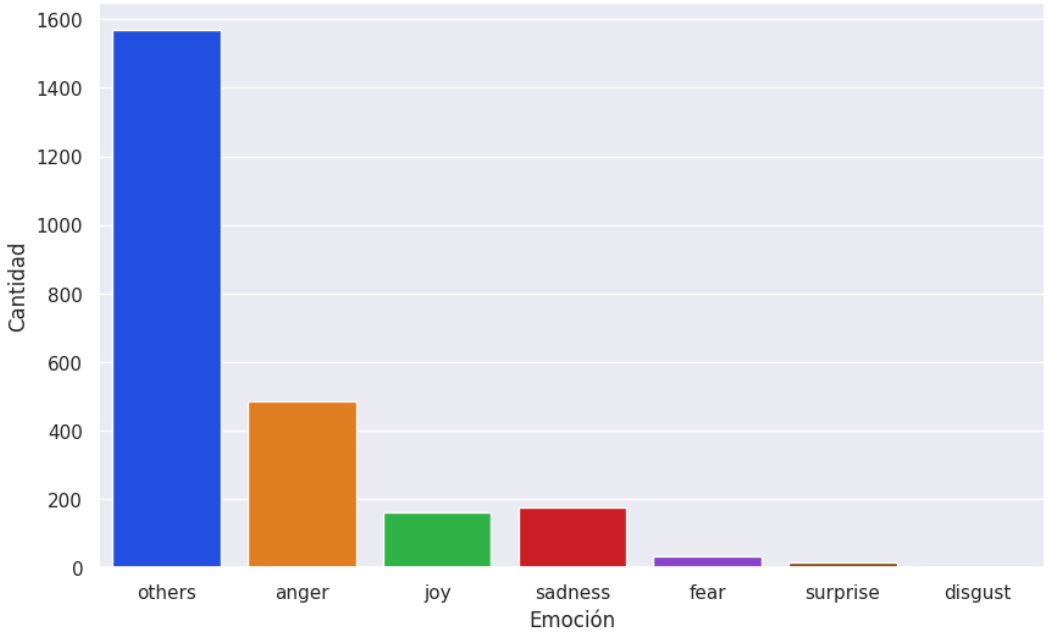


Figura 15: Distribución de emociones
Fuente: Elaboración propia

Un aspecto relevante adicional para el análisis es el que surge al examinar la Figura 16, donde se presenta la interrelación entre la distribución de sentimientos y las distintas emociones. Al explorar la disposición de las emociones en el contexto de los sentimientos, se observa que en las confesiones etiquetadas como positivas, se destacan principalmente *"Others"* con 238 ocurrencias, seguido de *"Joy"* con un valor de 146. Para las confesiones neutras *"Others"* tienen una fuerte presencia con 879 apariciones, mientras que para las negativas, *"Anger"*, *"Others"* y *"Sadness"*, presentan frecuencias de 484, 451 y 168, respectivamente.

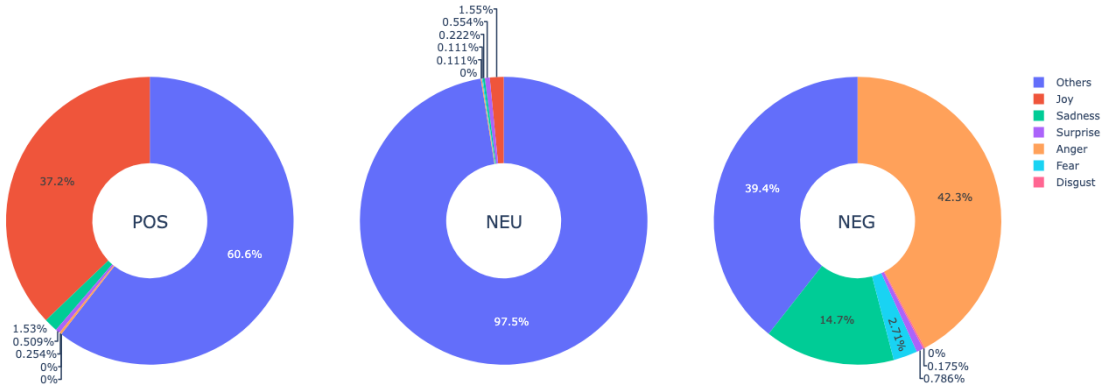


Figura 16: Distribución de emociones por sentimientos
Fuente: Elaboración propia

Con el objetivo de explorar las correlaciones presentes entre los sentimientos y las emociones, se introduce la Figura 17 en forma de una matriz simétrica. Dadas sus características, el enfoque analítico se dirige hacia la porción triangular superior de esta matriz.

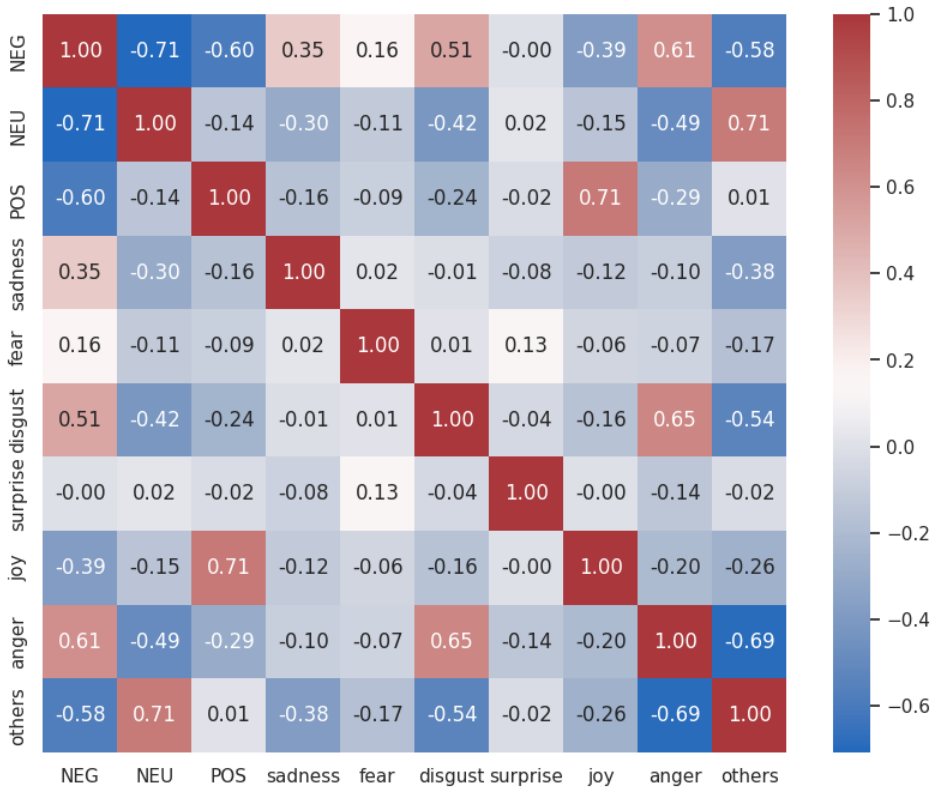


Figura 17: Matriz de correlación entre emociones y sentimientos
Fuente: Elaboración propia

Al examinar el sentimiento negativo, se destaca una correlación positiva con las emociones *"disgust"*, *"anger"* y *"sadness"*. Esto indica que las confesiones con connotación negativa suele estar cargadas de emociones de enojo, disguto y tristeza. Por otro lado, se observa una correlación negativa tanto con el sentimiento neutral como con el positivo, además de las emociones *"joy"* y *"others"*. Cabe mencionar que no existe una correlación entre el sentimiento negativo y la emoción *"surprise"*.

En relación al sentimiento neutro, se observa una correlación significativa con la emoción *"others"*, lo que podría interpretarse como una indicación de que las confesiones catalogadas como neutras suelen estar expresadas con emociones diferentes a las mencionadas previamente. Por otro lado, se destaca una correlación negativa con las emociones *"anger"*, *"disgust"* y *"sadness"*, es decir, los textos neutros no suelen tener una carga emocional asociada a estas emociones. Al igual que en el caso anterior, no se identifica una correlación entre la emoción *"surprise"* y el sentimiento neutro.

El sentimiento positivo exhibe una fuerte correlación con la emoción *"joy"*, lo que indica que los textos optimistas suelen incluir palabras que expresan alegría. Por otra parte, *"others"* y *"surprise"* no presentan correlación directa con el sentimiento positivo

En relación a las emociones, se destaca una notable correlación entre *"anger"* y *"disgust"*, lo cual sugiere que ambas emociones suelen expresarse simultáneamente, indicando la presencia de una carga emocional vinculada a la irritabilidad. Por otro lado, se observa una correlación negativa entre las emociones *"anger"* y *"others"*, así como entre *"disgust"* y *"others"*. Esta relación inversa podría señalar que las confesiones que contienen emociones de enojo o disgusto tienden a evitar el uso de términos más generales o diversos en su expresión.

Es de interés conocer aquellas palabras que resaltan con mayor recurrencia en las confesiones catalogadas como positivas. La Figura 18 presenta las 500 palabras más frecuentes para este sentimiento. Esta visualización proporciona una perspectiva sobre los temas y expresiones más comunes asociados con las confesiones positivas.

Figura 18: Nube de palabras asociadas al sentimiento positivo
Fuente: Elaboración propia

De la nube de palabras destacan términos como *"confieso"*, *"gusta"*, y *"amo"*, que reflejan sentimientos de aprecio y satisfacción. Asimismo, palabras como *"universidad"*, *"año"* y "genial"sugieren una conexión con experiencias educativas y momentos memorables. Es notorio que expresiones como *"vida"*, *"amor"* y *"amigos"* reflejan relaciones y emociones positivas, enriqueciendo el panorama de las confesiones optimistas.

En contraposición, la Figura 19 muestra las palabras relacionadas a las confesiones catalogadas como negativas, al mostrar las 500 palabras más frecuentes de esta clase. Entre las expresiones más destacadas se encuentran *"wn"* y *"wea"*, que reflejan un tono de descontento. Además, palabras como *"siento"*, *"gente"*, y *"quiero"* sugieren sentimientos de malestar y deseo insatisfecho. La presencia recurrente de términos como *"mierda"*, *"miedo"*, y *"puta"* revela una carga emocional intensa en las confesiones negativas. Se añade a esta lista la aparición significativa de palabras relacionadas con la pandemia, como *"año"*, *"online"*, *"semestre"*, que reflejan las preocupaciones y ansiedades derivadas de la situación sanitaria global. Es relevante notar que conceptos como *"universidad"*, *"profesor"* y *"clases"* también aparecen, posiblemente vinculados a momentos de estrés académico.

Figura 19: Nube de palabras asociadas al sentimiento negativo
Fuente: Elaboración propia

Un análisis de gran relevancia en este estudio se relaciona al contexto de la pandemia. A continuación, se presenta una lista de palabras directamente vinculadas al período de confinamiento, las cuales se han empleado para filtrar los textos dentro del conjunto de confesiones. Tras aplicar este filtro, se ha creado una nube de palabras, representada en la Figura 20, que exhibe las 500 palabras más frecuentes en los textos relacionados con el aislamiento social.

- covid19
- covid
- coronavirus
- pandemia
- cuarentena
- distanciamiento
- mascarilla
- vacuna
- infección
- contagio
- síntomas
- aislamiento
- pcr
- pruebas
- hospital
- cuidado
- virus
- epidemia
- prevención
- sana distancia
- teletrabajo
- confinamiento
- depresión
- tristeza
- triste
- sad
- extraño
- extrañar
- encierro
- ansiedad
- casa
- familia
- muerte
- enfermo
- enfermedad
- hospital
- certamen
- estrés
- online
- clases online
- educación
- presencialidad
- clases presenciales
- pruebas online

Las palabras más frecuentes evidencian sobre los temas y sentimientos más comunes en relación con la pandemia. Se observa que términos como *"casa"*, *"online"*, *"universidad"*, *"clases"*, *"familia"* y *"vida"* sugieren una adaptación a la nueva realidad de ese período de tiempo, además de la importancia de la educación y las relaciones familiares en este contexto. Además, palabras como *"gente"*, *"pareja"*, *"alguien"* y *"hablar"* se relacionan al tópico de mantener conexiones sociales a pesar de las restricciones. Palabras como *"cansado"*, *"depresión"*, *"triste"* y *"ganas"* reflejan los desafíos emocionales y mentales experimentados durante la pandemia, que convergen en las problemáticas de salud mental. La inclusión de términos como *"pandemia"* y *"seguir"* en la lista de palabras más repetidas resalta la conciencia colectiva sobre la situación global y el deseo de mantenerse informado y seguro. En conjunto, esta nube de palabras y el análisis de las palabras más repetidas revelan el panorama de las preocupaciones, experiencias y emociones de las personas durante la pandemia.

Figura 20: Nube de palabras asociadas al contexto de pandemia
Fuente: Elaboración propia

4.5. Reducción de Dimensionalidad

En el proceso de minería de datos, una etapa que no fue mencionada en la Figura 11, pero que es importante para el análisis, es la reducción de dimensionalidad. Mediante diversos métodos es posible comprender y visualizar patrones intrínsecos en los datos. Para esta tarea, se ponen a prueba dos métodos: el Análisis de Componentes Principales (PCA) y el *t-distributed Stochastic Neighbor Embedding* (t-SNE).

Inicialmente, se optó por realizar un experimento empleando la técnica PCA en el DataFrame presentado en la Tabla 9. Se consideraron exclusivamente aquellas columnas que contenían valores numéricos. Es relevante señalar que se omitió un paso intermedio de normalización, puesto que todos los valores ya se encontraban previamente normalizados.

Con el fin de visualizar los grupos presentes en los datos, se elige utilizar dos componentes principales para realizar una representación en 2D. Esta reducción de dimensionalidad simplifica la visualización de los grupos en un espacio fácilmente interpretable. En la Figura 21, se ilustra el resultado obtenido al aplicar PCA con dos componentes, donde cada punto se asocia con su respectiva clase.

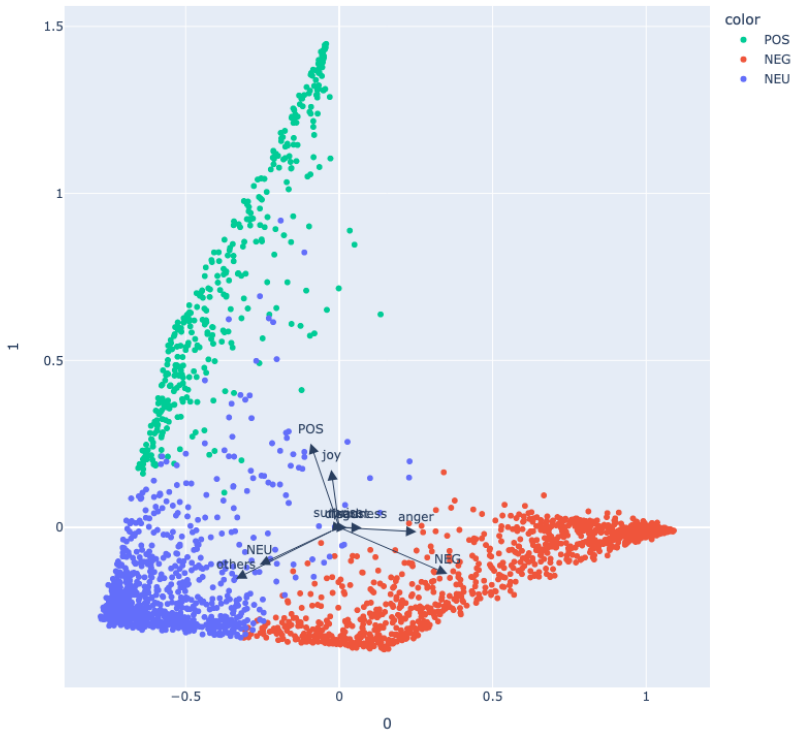


Figura 21: Reducción de dimensionalidad mediante PCA
Fuente: Elaboración propia

Luego de realizar una exploración inicial, se busca una aproximación alternativa para visualizar los datos con el objetivo de comparar y seleccionar la opción más adecuada. En esta línea, se emplea t-SNE como enfoque alternativo. Mediante esta técnica, se aplica la reducción de dimensionalidad con el propósito de lograr una representación visual en dos dimensiones, por lo tanto se utilizan dos componentes; además, se utiliza el parámetro de perplejidad, que regula la cantidad de vecinos de cada punto, este se fija en 30, dado que el conjunto de datos no es grande. En la Figura 22, se exhibe el resultado de esta reducción de dimensionalidad utilizando t-SNE, destacando las clases correspondientes de cada punto en el espacio visual resultante.

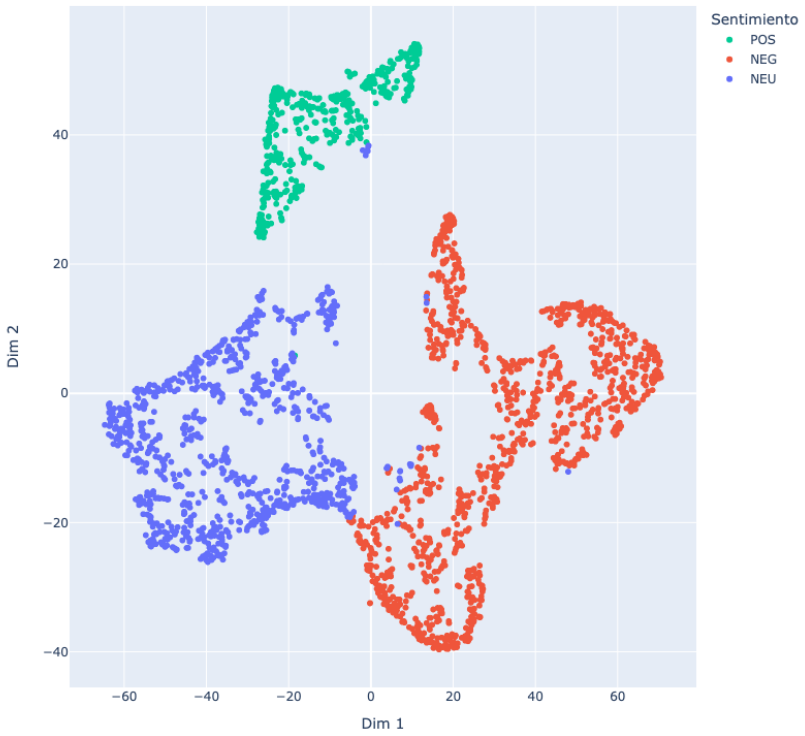


Figura 22: Reducción de dimensionalidad mediante t-SNE
Fuente: Elaboración propia

La comparación entre las visualizaciones generadas mediante PCA y t-SNE proporciona claras perspectivas diferenciadoras. En el caso de t-SNE, los grupos se destacan por su formación más nítida y sólida en comparación con las representaciones generadas por PCA, las cuales exhiben cierta difusión, especialmente entre los grupos 'POS' y 'NEU'. En relación a t-SNE, es notable la presencia de puntos que parecen dispersarse hacia agrupaciones distintas a las correspondientes, como por ejemplo, en la categoría 'NEG' donde se detectan puntos de carácter 'NEU'. Este fenómeno puede ser atribuido a la habilidad inherente de t-SNE para capturar relaciones no lineales y complejas en los datos, generando agrupaciones más acertadas. En contraste, PCA enfoca su atención en la linealidad, lo que podría limitar su capacidad para distinguir claramente agrupaciones con características no lineales. Por consiguiente, en este contexto, t-SNE emerge como la opción preferida para la visualización de grupos en el análisis de sentimiento, debido a su versatilidad para relaciones no lineales y su capacidad para producir agrupamientos más cohesivos y distintivos.

4.6. Clusterización

Durante el proceso de clusterización, se experimenta con tres modelos distintos: K-Means, Modelos de Mezcla Gaussiana y Fuzzy C-Means, utilizando variados valores para el parámetro de pertenencia en este último. Para facilitar la identificación de patrones, se emplea una transformación t-SNE de dos componentes, la cual fue presentada en la sección anterior. Al hacer uso de las capacidades visuales intrínsecas de t-SNE, se logra una mayor claridad en la identificación de las relaciones y subdivisiones complejas entre los clústeres en los gráficos generados. Esta sinergia planificada entre las técnicas de agrupamiento y la reducción de dimensionalidad mediante t-SNE emerge como una metodología robusta para desentrañar y comprender patrones significativos en los datos.

El proceso se inicia con la aplicación del modelo K-Means de la biblioteca *Scikit-Learn*, donde se establecen parámetros específicos. Se configura el número de clústeres igual a tres, coincidiendo con las agrupaciones presentes en los datos: positivo (’POS’), negativo (’NEG’) y neutro (’NEU’). Para asegurar la reproducibilidad y repetibilidad de los resultados, se fija una semilla aleatoria mediante el parámetro `random_state=0`. Una vez definida la configuración, el modelo se ejecuta y se evalúa utilizando cuatro métricas distintas: *Silhouette Score*, *Calinski-Harabasz Index*, *Davies-Bouldin Index y Purity Score*. Cada una de estas métricas aporta información valiosa sobre diversos aspectos de la calidad de los clústeres obtenidos. Posteriormente, se asignan los grupos generados a las etiquetas de sentimiento correspondientes, y el resultado se representa visualmente en la Figura 23.

En la Tabla 10, se presentan los valores obtenidos para cada coeficiente como resultado de la evaluación del modelo K-Means. Los resultados del modelo K-Means muestran un panorama alentador en términos de calidad de los grupos formados. El *Silhouette Score* de 0.489 indica una proximidad entre los puntos dentro de los agrupamientos y una distancia razonable con clústeres vecinos. El alto valor del índice *Calinski-Harabasz* que alcanza la cifra de 3014.354 refleja una cohesión intraclúster sólida y una separación efectiva entre clústeres. Además, el índice *Davies-Bouldin*, con un valor de 0.684, sugiere una buena discriminación entre clústeres, y el *Purity Score*, igual a 0.936, denota una captura precisa de las clases reales. En conjunto, estos resultados sugieren que el modelo K-Means logra una agrupación coherente y efectiva.

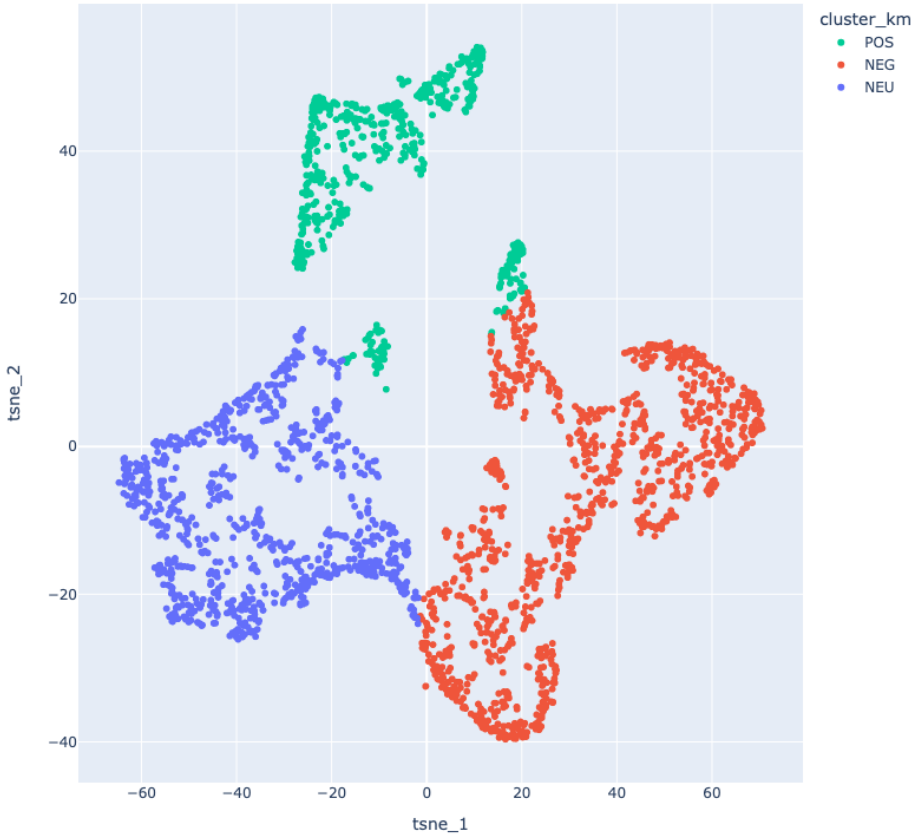


Figura 23: Visualización de asignaciones de clústeres mediante K-Means
Fuente: Elaboración propia

Modelo	**K-Means**
Silhouette Score	0,489 977
Calinski-Harabasz Index	3014,354 149
Davies-Bouldin Index	0,684 740
Purity Score	0,936 066

Tabla 10: Coeficientes de validación de clúster para K-Means

Después de la evaluación inicial con K-Means, se procede a emplear el algoritmo GMM de la biblioteca *Scikit-Learn*. Manteniendo la consistencia en los parámetros, se fijó el valor de `n_components=3` y se estableció `random_state=5` para asegurar la reproductibilidad y repetibilidad de los resultados. Una vez que los datos son ajustados al modelo, se calculan los coeficientes de validación del clúster correspondientes. Para completar el proceso, se realiza

el mapeo de los agrupamientos obtenidos con sus respectivas etiquetas de sentimiento y se genera la Figura 24 con el fin de lograr una mejor comprensión de los resultados.

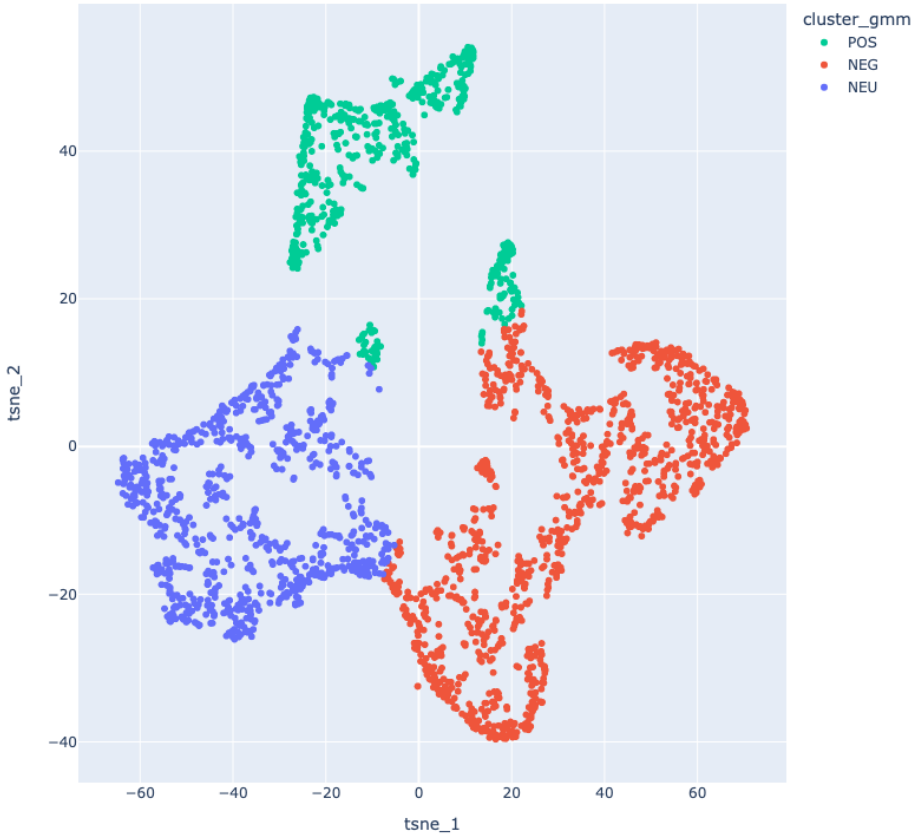


Figura 24: Visualización de asignaciones de clústeres mediante GMM
Fuente: Elaboración propia

Modelo	**GMM**
Silhouette Score	0,487 763
Calinski-Harabasz Index	2959,326 981
Davies-Bouldin Index	0,696 180
Purity Score	0,933 606

Tabla 11: Coeficientes de validación de clúster para GMM

Los valores para cada coeficiente se muestran en la Tabla 11. El modelo GMM exhibe una evaluación positiva en la calidad de los clústeres. Con un *Silhouette Score* de 0.487, expresa una cercanía adecuada dentro de los clústeres y una separación razonable entre ellos. El índice *Calinski-Harabasz*, valorado en 2959.326, destaca una cohesión intraclúster sólida y separación efectiva. A pesar de que el índice *Davies-Bouldin* es 0.696, lo que muestra una aceptable separación entre clústeres, aún puede mejorarse. Por último, el *Purity Score* de 0.933 demuestra una efectiva captura de las clases reales. De lo anterior se puede concluir que el modelo GMM logra agrupaciones coherentes y satisfactorias, con posibilidad de perfeccionar la separación entre clústeres.

En la fase final, se procede a evaluar el modelo *Fuzzy C-Means* de la biblioteca *skfuzzy*. En este contexto, se llevaron a cabo experimentos con diversos parámetros específicos: se fijó la cantidad de `centroids=3`, el error se configuró en `error=0.005`, y se definió un máximo de iteraciones igual a `maxiter=1000`. Con el propósito de garantizar la replicabilidad y reproducibilidad, se utilizó una semilla predefinida. Sin embargo, lo más destacado de este experimento radica en la exploración de distintos valores para el parámetro `m`, el cual controla el nivel de difusión en el algoritmo. Con el propósito de analizar su impacto, se llevaron a cabo pruebas con cinco valores diferentes: 1.01, 2, 3, 4 y 5. Luego de ejecutar el modelo, se calculan los coeficientes de validación pertinentes, y se procede con asignar los valores de sentimiento a los resultados derivados de la clusterización. Finalmente, se presentan visualizaciones ilustrativas que resumen los hallazgos obtenidos en este proceso.

Las Figuras 25, 26, 27, 28 y 29 exhiben los clústeres generados para cada configuración correspondiente. Paralelamente, en las Tablas 12 13, 14, 15 y 16 se detallan los valores alcanzados por cada coeficiente en función de su configuración correspondiente.

Entre los diversos resultados derivados de las diferentes configuraciones del modelo *Fuzzy C-Means*, destaca claramente por su rendimiento sobresaliente *Fuzzy C-Means* con valor de `m=1.01`. Este modelo presenta un coeficiente *Silhouette Score* de **0.490099**, indicando una óptima cohesión intraclúster y separación interclúster. Además, registra un valor mayor en el índice *Calinski-Harabasz*, con una cifra de **3014.327287**. Esto refuerza la noción de que las instancias dentro de cada clúster están cercanas entre sí, mientras que los diferentes clústeres están bien separados, en comparación con las otras configuraciones evaluadas. Por otro lado, el coeficiente de *Davies-Bouldin* es el más bajo entre todas las configuraciones, marcando **0.683076**. Esto demuestra una densidad intraclúster sólida y una separación interclúster efectiva, aunque aún queda margen para la mejora. Además, este modelo logra un destacado valor de *Purity Score*, con **0.936885**, lo que indica su habilidad para asignar con precisión las instancias a sus clústeres correspondientes. En conjunto, estos resultados subrayan la distintiva capacidad del modelo **FCM-1.01** para identificar con precisión y coherencia las estructuras en los datos.

Es esencial destacar una diferencia significativa en los resultados al comparar el modelo con el parámetro `m=5` con el resto de las configuraciones. Este modelo exhibe un rendimiento notablemente menos favorable en diversos aspectos. El *Silhouette Score*, indicador de la cohesión intraclúster y la separación interclúster, se sitúa en un valor considerablemente inferior

de **0.452416**. Además, el índice *Calinski-Harabasz*, que evalúa la coherencia y separación de los clústeres, registra su valor mínimo de **2731.584089** en este caso. En cuanto al índice *Davies-Bouldin*, registra el valor más alto de los experimentos, siendo igual **0.807470**. Asimismo, el *Purity Score*, que cuantifica la precisión de la asignación de instancias a los clústeres predominantes, se establece en **0.872541**, demostrando ser el más bajo entre todas las configuraciones exploradas. Esta serie de resultados resalta un desempeño menos satisfactorio del modelo con `m`=5, en contraposición a los demás modelos evaluados.

Es relevante subrayar que los modelos restantes, específicamente aquellos en los que se establecieron valores de `m` iguales a 2, 3 y 4, manifiestan un rendimiento promedio similar, aunque no destacable.

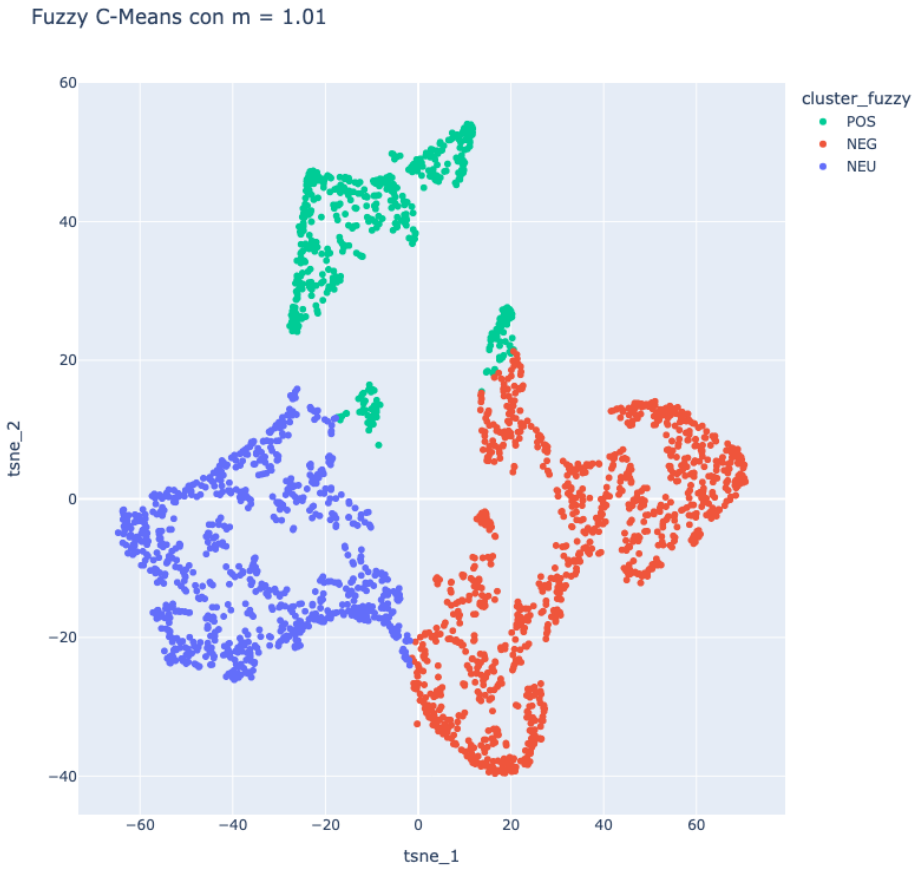


Figura 25: Visualización de asignaciones de clústeres mediante Fuzzy C-Means con m=1.01
Fuente: Elaboración propia

Modelo	**FCM-1.01**
Silhouette Score	0,490 099
Calinski-Harabasz Index	3014,327 287
Davies-Bouldin Index	0,683 076
Purity Score	0,936 885

Tabla 12: Coeficientes de validación de clúster para Fuzzy C-Means con m=1.01

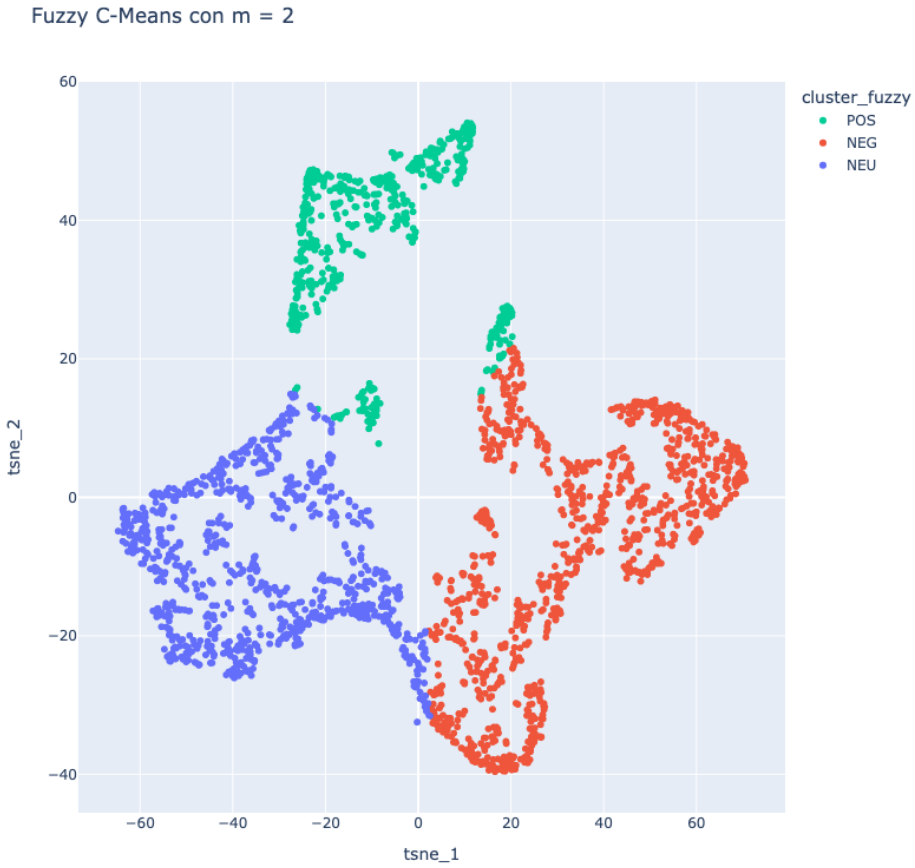


Figura 26: Visualización de asignaciones de clústeres mediante Fuzzy C-Means con m=2
Fuente: Elaboración propia

Modelo	FCM-2
Silhouette Score	0,484 684
Calinski-Harabasz Index	2992,605 725
Davies-Bouldin Index	0,687 330
Purity Score	0,919 672

Tabla 13: Coeficientes de validación de clúster para Fuzzy C-Means con m=2

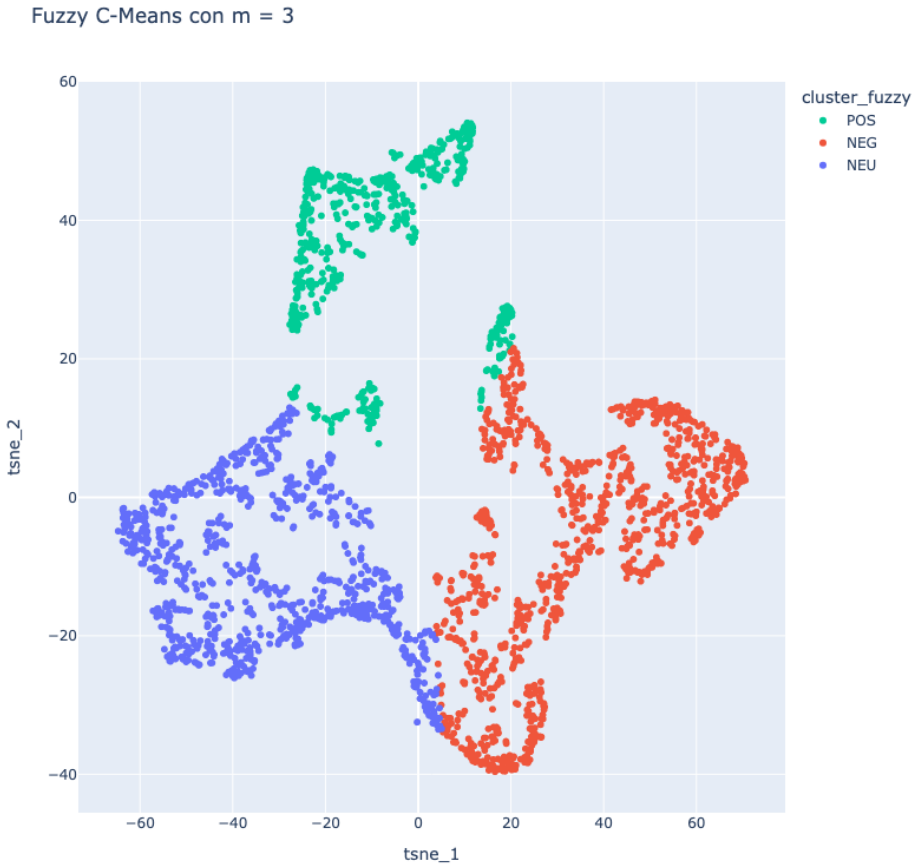


Figura 27: Visualización de asignaciones de clústeres mediante Fuzzy C-Means con m=3
Fuente: Elaboración propia

Modelo	FCM-3
Silhouette Score	0,477 921
Calinski-Harabasz Index	2952,073 902
Davies-Bouldin Index	0,709 346
Purity Score	0,900 000

Tabla 14: Coeficientes de validación de clúster para Fuzzy C-Means con m=3

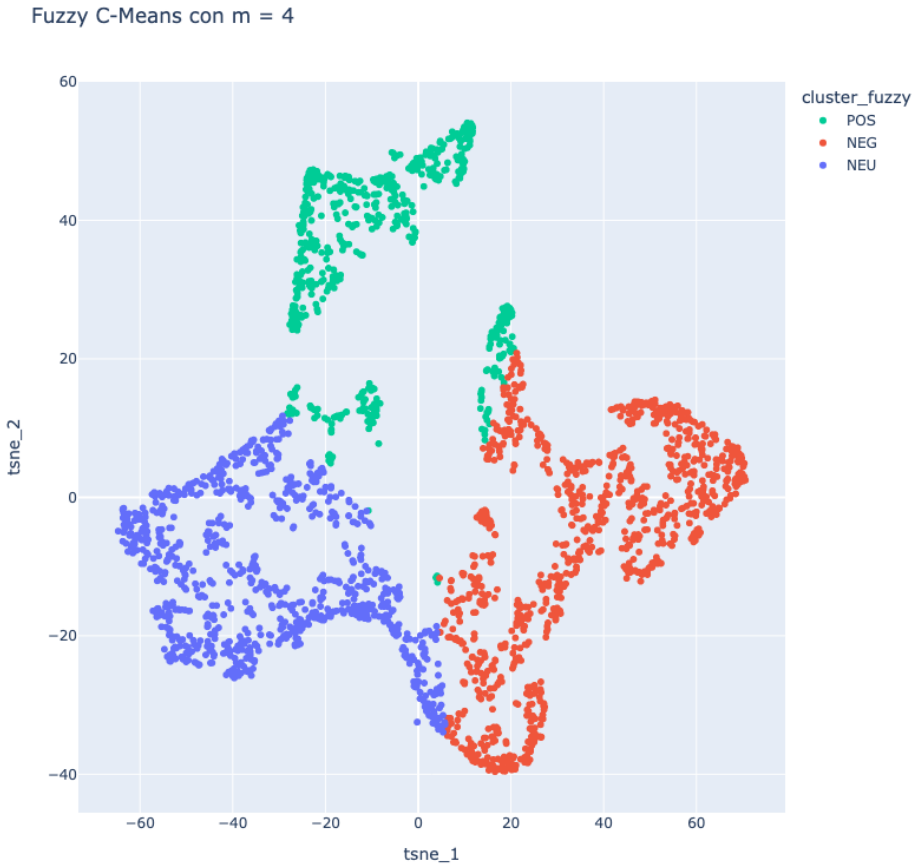


Figura 28: Visualización de asignaciones de clústeres mediante Fuzzy C-Means con m=4
Fuente: Elaboración propia

Modelo	FCM-4
Silhouette Score	0,468 524
Calinski-Harabasz Index	2884,652 379
Davies-Bouldin Index	0,745 174
Purity Score	0,882 787

Tabla 15: Coeficientes de validación de clúster para Fuzzy C-Means con m=4

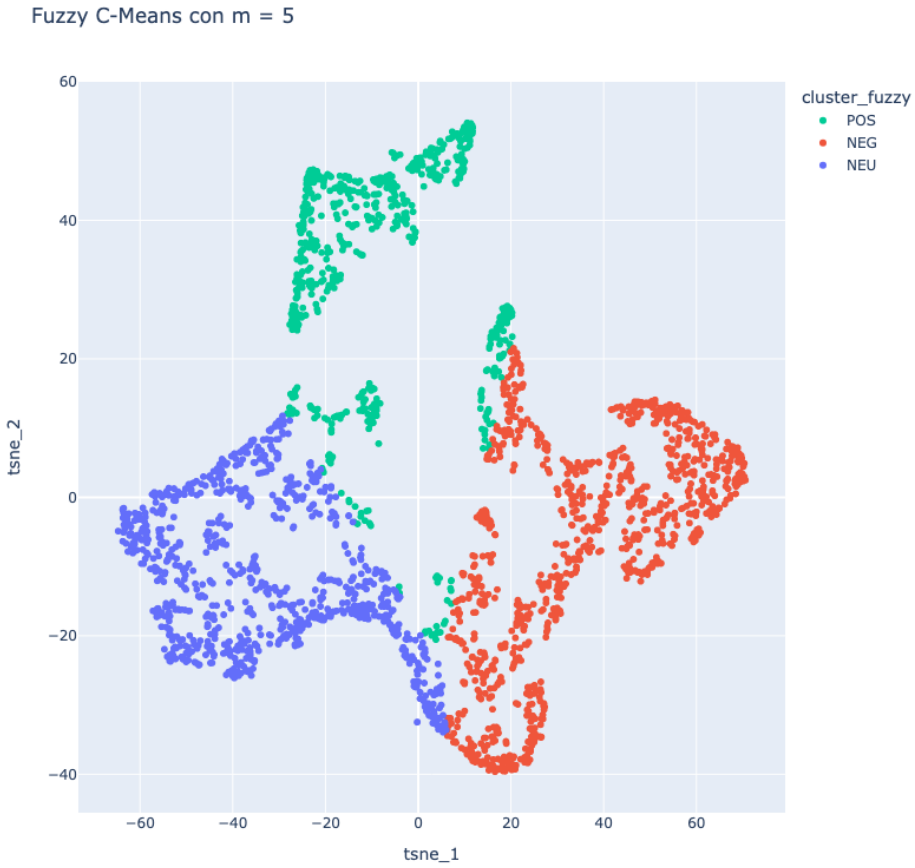


Figura 29: Visualización de asignaciones de clústeres mediante Fuzzy C-Means con m=5
Fuente: Elaboración propia

Modelo	**FCM-5**
Silhouette Score	$0{,}452\,416$
Calinski-Harabasz Index	$2731{,}584\,089$
Davies-Bouldin Index	$0{,}807\,470$
Purity Score	$0{,}872\,541$

Tabla 16: Coeficientes de validación de clúster para Fuzzy C-Means con m=5

4.7. Análisis de Coeficientes de Validación de Clustering

Los resultados derivados de la evaluación de cada modelo previamente analizado se encuentran resumidos en las Tabla 17.

Algoritmo	**Silhouette Score**	**Calinski-Harabasz Index**	**Davies-Bouldin Index**	**Purity Score**
K-Means	0.489977	3014.354149	0.684740	0.936066
GMM	0.487764	2959.326981	0.696185	0.933607
Fuzzy-1.01	0.490099	3014.327287	0.683076	0.936885
Fuzzy-2	0.484684	2992.605725	0.687330	0.919672
Fuzzy-3	0.477921	2952.073902	0.709346	0.900000
Fuzzy-4	0.468524	2884.652379	0.745174	0.882787
Fuzzy-5	0.452416	2731.584089	0.807470	0.872541

Tabla 17: Coef. de validación de clústeres para datos reducidos a 2 componentes con t-SNE

Para determinar cuál modelo presenta un mejor rendimiento en cada experimento, se busca alcanzar valores elevados en todas todos los coeficientes, excepto el Índice de Davies-Bouldin, que se busca minimizar, dado que un valor más cercano a cero es mejor. Con este propósito, se calcula una ponderación final que expresa todas las métricas para evaluar de manera más equilibrada el desempeño de los algoritmos.

La ponderación final se obtiene mediante la asignación de pesos a cada métrica. Se asignan los siguientes pesos a las métricas:

- **Peso Silhouette Score (PSS):** 1
- **Peso Calinski-Harabasz Index (PCHI) :** 1
- **Peso Davies-Bouldin Index (PDBI) :** -1
- **Peso Purity Score (PPS) :** 1

Con el objetivo de minimizar el impacto del índice de Davies-Bouldin, se le asigna un peso negativo distinto al resto de índices. La ponderación final para cada algoritmo se calcula utilizando la siguiente fórmula:

$$Pond.Final = SS + CHI - DBI + PS \tag{5}$$

Es importante destacar que el índice *Calinski-Harabasz* difiere de los demás coeficientes al operar en una escala distinta, presentando valores fuera del rango convencional entre 0 y 1. Por ende, resulta imperativo normalizar previamente estos valores para homogeneizar las escalas y facilitar una comparación precisa entre todas las métricas. Para lograr la homogeneización de los índices, se emplea la siguiente fórmula de normalización:

$$x_{norm} = \frac{(x - \text{min})}{(\text{max} - \text{min})} \tag{6}$$

En la Tabla 18, se presenta la columna "Ponderación Final" que muestra los resultados calculados para cada modelo, ordenados de mayor a menor. Por otra parte, en la columna "Ponderación Final Norm" se aplica una normalización a los valores de las ponderaciones finales utilizando la ecuación 6, basada en el mínimo y máximo valor posibles que se pueden obtener a partir de la ecuación 5. El valor mínimo alcanzable con todos los valores de coeficientes normalizados es -2, lo cual ocurre cuando SS=-1, CHI=0, DBI=+1, y PS=0. En contraste, el valor máximo posible es 3, que se alcanza cuando SS=+1, CHI=+1, DBI=0, y PS=+1.

Algoritmo	SS	CHI	CHI-Norm	DBI	PS	Pond. Final	Pond. Final Norm
Fuzzy-1.01	0.490099	3014.327287	0.9999050041	0.683076	0.936885	1.744	0.749
K-Means	0.489977	3014.354149	1	0.684740	0.936066	1.741	0.748
Fuzzy-2	0.484684	2992.605725	0.9230879535	0.687330	0.919672	1.640	0.728
GMM	0.487764	2959.326981	0.8053995957	0.696185	0.933607	1.531	0.706
Fuzzy-3	0.477921	2952.073902	0.7797495004	0.709346	0.900000	1.448	0.690
Fuzzy-4	0.468524	2884.652379	0.5413171748	0.745174	0.882787	1.147	0.629
Fuzzy-5	0.452416	2731.584089	0	0.807470	0.872541	0.517	0.503

Tabla 18: Coef. de validación de clústeres normalizados

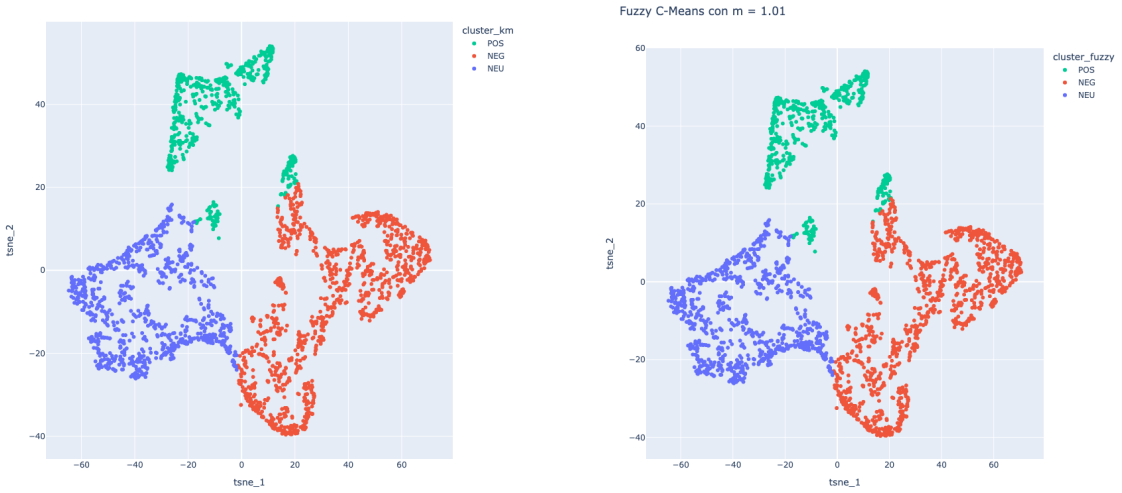
Basado en los resultados obtenidos, se puede observar que, el algoritmo Fuzzy-C Means con un parámetro `m=1.01` emerge como el de mayor valor de ponderación, para el experimento llevado a cabo utilizando t-SNE con dos componentes de reducción dimensional. Este algoritmo logra un equilibrio en las diversas métricas de validación, demostrando una sólida capacidad para definir clústeres con una separación efectiva en este espacio bidimensional. La combinación de t-SNE y Fuzzy-C Means parece ser una estrategia exitosa para el análisis exploratorio y la interpretación de los datos, ya que no solo produce resultados sólidos en términos de ponderación, sino que también facilita la comunicación visual de los resultados a través de la representación en dos dimensiones.

Además de los resultados destacados obtenidos por el algoritmo Fuzzy-C Means, es relevante mencionar el buen performance de la segunda opción considerada en el estudio: el algoritmo K-Means. Con una ponderación de 0.748, este modelo muestra una diferencia mínima en términos de ponderación en comparación con Fuzzy-C Means, que obtiene una ponderación de 0.749. Esta pequeña discrepancia en los valores de ponderación sugiere que ambos algoritmos están compitiendo en eficacia en términos de separación y definición de clústeres en el espacio bidimensional.

Es interesante observar cómo los coeficientes de ponderación para K-Means son muy cercanos a los de Fuzzy-C Means, lo que puede atribuirse a que ambos enfoques exploran estructuras similares en los datos. Aunque la ponderación es apenas ligeramente diferente, esta leve ventaja puede deberse a la naturaleza de asignación rígida de K-Means en comparación con la asignación difusa de Fuzzy-C Means.

Esta proximidad en los resultados sugiere que ambas estrategias, t-SNE en combinación con tanto Fuzzy-C Means como K-Means, pueden ofrecer una sólida base para el análisis exploratorio y la interpretación de los datos en este contexto específico.

En la Figura 30 se presenta una comparación visual de los clústeres logradas por ambos modelos.



(a) Clustering con K-Means

(b) Clustering con Fuzzy C-Means y m igual 1.01

Figura 30: Comparación de modelos con mejor rendimiento

Resulta evidentemente complicado percibir a simple vista las diferencias distintivas entre ellos. Los clústeres exhiben una notable similitud, presentando agrupaciones claramente definidas en ambos casos. Sin embargo, es particularmente intrigante observar que dentro del clúster positivo emergen dos subagrupaciones de menor tamaño: una inclinada hacia las instancias positivas y otra hacia las negativas. Esta dualidad podría ser atribuida a la existencia

de textos que poseen emociones positivas que rozan la categoría de la neutralidad, así como casos similares en la vertiente negativa. Este fenómeno resalta la complejidad de la clasificación y sugiere que ciertas instancias pueden presentar características intermedias entre las emociones polarizadas.

Al concluir este capítulo, se ha alcanzado con éxito el objetivo general de categorizar y analizar los sentimientos de textos escritos por sansanos publicados en *Instagram*, mediante técnicas de procesamiento de lenguaje natural y visualización de datos con el propósito de comprender mejor sus emociones y experiencias durante este período. A lo largo de los análisis minuciosos presentados, se ha arrojado luz sobre las complejidades de sus sentimientos y emociones en el contexto de pandemia, reflejando una presencia superior de textos negativos que positivos y neutros, en consecuencia, las emociones presentes en los textos son más enojo y tristeza que alegría. Desde la visualización de las clasificaciones de sentimientos realizadas por los tres modelos puestos a prueba, hasta el análisis de la distribución de emociones por sentimiento, y desde la reducción de dimensionalidad hasta la clusterización, se ha explorado diversas facetas de las experiencias enviadas por los sansanos a esta plataforma social, tanto en el ámbito emocional, sentimental y en las expresiones más utilizadas. Estos hallazgos proporcionan una valiosa visión de sus interacciones y emociones en un período particularmente desafiante. En resumen, este capítulo representa un paso significativo hacia una comprensión más completa de las vivencias de los estudiantes en el contexto de pandemia.

CONCLUSIONES

En este trabajo de titulación, se ha abordado de manera exhaustiva el objetivo general que se planteó inicialmente: categorizar y analizar los sentimientos presentes en los textos redactados por estudiantes de la comunidad sansana que participaron en la plataforma social *Instagram* durante el período de la pandemia. A través de la implementación y evaluación de modelos de NLP, se logró obtener una comprensión profunda de las emociones expresadas en estos textos. La evaluación rigurosa de los modelos, utilizando métricas de rendimiento, permitió determinar su eficacia en la clasificación de los contenidos sentimentales. Además, se desarrollaron visualizaciones intuitivas y efectivas que facilitan la identificación de grupos y tendencias emergentes a partir del análisis de los textos recopilados.

La metodología empleada en este estudio se basa en el proceso KDD, que consta de cinco etapas fundamentales: la selección de los datos, el procesamiento de datos y la transformación de los mismos, seguidos de la fase de minería de datos y, por último, la evaluación de los resultados obtenidos. Cada una de estas etapas presentó sus propios desafíos y dificultades que se enlistan a continuación:

1. **Selección de los datos :** La recopilación de datos desde *Instagram* implicó abordar problemas de privacidad y acceso a la información, así como la necesidad de obtener un conjunto de datos representativo y significativo. Cabe destacar que la biblioteca utilizada para esta tarea fue *Instaloader*, aunque si bien permitía trabajar con rangos de fechas para la descarga de imágenes, en muchas ocasiones se tuvo que interrumpir el proceso de descarga. Esto se debió a limitaciones relacionadas con la cantidad máxima de consultas permitidas, lo que requería una descarga en lotes más pequeños y, en consecuencia, ralentizaba el proceso. Además, un problema repetitivo fue que muchas de las imágenes descargadas no eran confesiones, sino contenido distinto como memes, publicidad y otros tipos de publicaciones. Esto generó la necesidad de filtrar y eliminar estas imágenes del conjunto de datos final, lo que implicó un esfuerzo adicional en la selección de datos.

2. **Procesamiento de datos :** El preprocesamiento de datos presentó desafíos relacionados con el uso de *Tesseract* para extraer el texto desde las confesiones. *Tesseract* no reconoció *emojis* ni emoticones, lo que resultó en caracteres incorrectos asociados a símbolos gráficos, generando la necesidad de una exhaustiva y tediosa limpieza de los textos. Otro problema asociado a esta etapa se debió a la cargada presencia de modismos y expresiones coloquiales en las confesiones, fue necesario aplicar una normalización de palabras para que los modelos posteriores pudieran entender los términos específicos utilizados por la comunidad sansana, lo que añadió complejidad al proceso de preprocesamiento.

3. **Minería de datos :** La principal dificultad en el proceso de minería de datos radicó en la limitación de los modelos empleados para clasificar los sentimientos en las confe-

siones escritas en español. Dos de los tres modelos utilizados estaban originalmente entrenados en inglés, lo que requirió un proceso de traducción intermedia para su adaptación al español. Dada la escasez de modelos específicos para el español, fue necesario desarrollar una metodología propia para lograr una clasificación precisa y justa de los sentimientos expresados en el texto.

En cuanto a los resultados obtenidos, a partir de la evaluación de tres modelos de clasificación de sentimientos, *PySentimiento*, *TextBlob y Vader*, junto con la implementación de un algoritmo de votación para concluir una etiqueta para cada texto de manera justa, se determina que la gran mayoría de las publicaciones presentan un sentimiento negativo. Este hallazgo refleja la realidad del periodo de pandemia, en el que una carga emocional negativa afectó significativamente a la comunidad sansana. Entre los modelos evaluados, *PySentimiento* se destaca como el más adecuado para el contexto del problema, debido a su rendimiento superior en comparación con los otros dos modelos, sumado a los métodos de análisis sentimental y emocional que ofrece la biblioteca.

Uno de los hallazgos más destacados de este análisis es la prevalencia de sentimientos negativos en las confesiones sansanas, seguidos por sentimientos neutros y positivos en orden descendente. Las emociones más comunes en estas confesiones incluyen 'others', 'anger', 'joy' y 'sadness', mientras que 'fea', 'surprise' y 'disgust' son menos frecuentes. Es relevante notar que entre las emociones negativas, las más prominentes son 'others', 'anger' y 'sadness', lo que proporciona una visión de las confesiones enviadas por la comunidad estudiantil durante el período de la pandemia.

En cuanto a las confesiones catalogadas como neutras, no exhiben un sentimiento particularmente dominante, a excepción de 'others'. Por otro lado, en las confesiones positivas, destaca la emoción 'joy', sugiriendo que los textos positivos conllevan mensajes relacionados con la alegría y el disfrute. Este hallazgo se respalda al observar las palabras que se asocian con cada sentimiento. Las palabras positivas están fuertemente vinculadas a temas de amistad, amor y momentos felices, mientras que las palabras negativas hacen referencia a la universidad, los sentimientos, el año, la vida, el tiempo y el miedo.

Al analizar las palabras filtradas por temas relacionados con la pandemia, emergen categorías que abordan cuestiones como el estar en casa, las clases en línea, la universidad, la vida, la familia, la tristeza, el cansancio y la depresión. Estos resultados permiten profundizar en las preocupaciones y experiencias compartidas por los estudiantes durante este período de tiempo.

Además, para explorar la agrupación de los textos en función de los sentimientos y emociones, se probaron tres modelos de clusterización diferentes, *K-Means*, *GMM y FCM*, incluyendo diversas configuraciones del parámetro que regula la difusión en el caso de FCM. Después de realizar una exhaustiva comparación utilizando diversos coeficientes de validación de clúster, se determinó que FCM con el parámetro de difusión 'm' igual a 1.01 y el algoritmo *K-Means* son los dos mejores modelos para la agrupación de textos. Las visualiza-

ciones derivadas de estos clústeres revelan que existen textos etiquetados como 'positivos' que, en realidad, presentan una carga emocional que roza la neutralidad y la negatividad. Esto sugiere que, aunque se clasifiquen como 'positivos', estos textos todavía están imbuidos de emociones que se asemejan a lo negativo y lo neutral.

El aprendizaje acumulado a lo largo de mis años de carrera, especialmente en cursos como Bases de Datos, Visualización de Datos, Minería de Datos, Introducción a la Ciencia de Datos, Estadística Computacional y Redes Neuronales, ha sido de vital importancia en la gestación y desarrollo de este trabajo de titulación. Estos cursos no solo me proporcionaron conocimientos teóricos sólidos sino también habilidades prácticas fundamentales, estableciendo las bases necesarias para llevar a cabo un proyecto de análisis de datos en consonancia con las exigencias de la industria. Es relevante destacar que el curso de Minería de Datos, a pesar de no ser un electivo comúnmente ofertado, fue de suma importancia al inspirarme y orientarme hacia la minería de texto como enfoque para el análisis de sentimientos.

Para este trabajo de título, se enfocó en una muestra de datos recopilados durante el periodo de pandemia, la cual es relativamente pequeña en comparación al total de confesiones disponibles actualmente en la cuenta de *Instagram*. Una de las líneas de investigación futura es aumentar la cantidad de datos recopilados para mejorar el análisis de sentimiento, utilizando otras técnicas distintas a las ya presentadas en este trabajo, como DBSCAN a modo de aprendizaje no supervisado o directamente entrenar modelos propios con *RoBERTuito*. Además, se plantea la inclusión de *emojis* en el análisis y extracción del texto, dado que estos símbolos gráficos están cada vez más integrados en nuestra forma de comunicación y pueden enriquecer el contexto emocional del mensaje. Identificar, comprender e interpretar adecuadamente los *emojis* permitiría un análisis más preciso de la carga emocional de los textos en redes sociales y cualquier otra plataforma de interacción. Otra área interesante para seguir investigando es la aplicación de un análisis predictivo de depresión a partir del texto en las publicaciones de *Instagram*. Detectar signos tempranos de depresión sería de gran importancia para brindar apoyo y atención a los estudiantes que lo necesiten. Por último, se sugiere desarrollar un estudio enfocado en la detección de tópicos específicos de las confesiones escritas por los sansanos para comprender mejor sus objetivos, inquietudes y necesidades particulares. Con estos avances, se mejoraría significativamente la capacidad de comprender el sentir de la comunidad sansana.

Finalizando, este trabajo de titulación me ha permitido profundizar en *Instagram* como una una herramienta y fuente de datos que es masivamente utilizada por los sansanos para expresar sus pesnamientos. Pero más allá de la exploración y colección de datos desde esta plataforma, se abre una valiosa posibilidad: comprender las necesidades de los alumnos y ofrecer apoyo temprano a quienes lo necesiten. En estos tiempos de creciente importancia de la salud mental, la detección anticipada de posibles signos de angustia o depresión a través del análisis de texto en las confesiones se convierte en una instrumento valiosa. Al conocer esta perspectiva, no solo se amplía el conocimiento de la comunidad sansana, sino que también se contribuye a la visibilización de las necesidades de los alumnos de la UTFSM.

REFERENCIAS BIBLIOGRÁFICAS

[abb,] Abbyy ocr sdk. https://www.abbyy.com/es/ocr-sdk/. Accedido el: 29/09/2023.

[pyo, 2023] (2023). Pyocr. https://pypi.org/project/pyocr/.

[Adobe,] Adobe. What is ocr and why is ocr software important? https://www.adobe.com/acrobat/guides/what-is-ocr.html.

[AI, 2020] AI, B. (2020). Nlp, nlu y nlg: ¿cuál es la diferencia? una guía completa.

[Azevedo y Santos, 2008] Azevedo, A. y Santos, M. (2008). Kdd, semma and crisp-dm: A parallel overview. pp. 182–185.

[Beheshti, 2023] Beheshti, A. (2023). Unsupervised learning: Clustering using gaussian mixture model (gmm).

[Boulid *et al.*, 2017] Boulid, Y., Souhar, A., y Elkettani, Y. (2017). Handwritten character recognition based on the specificity and the singularity of the arabic language. *International Journal of Interactive Multimedia and Artificial Intelligence*, 4:45–53.

[Breiman *et al.*, 1984] Breiman, L., Friedman, J., Stone, C., y Olshen, R. (1984). *Classification and Regression Trees*. Taylor and Francis.

[Cepal y UNESCO, 2020] Cepal y UNESCO (2020). La educación en tiempos de la pandemia de covid-19. p. 21 p.

[Chaitanyanarava, 2020] Chaitanyanarava (2020). A complete guide on dimensionality reduction.

[Community, 2023] Community, T. I. (Último acceso: 2023). Instaloader documentation. https://instaloader.github.io/index.html.

[Dai *et al.*, 2019] Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V., y Salakhutdinov, R. (2019). Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

[Developers,] Developers, G. Clustering algorithms. Online. Accessed on: [Fecha de acceso].

[Devlin *et al.*, 2018] Devlin, J., Chang, M.-W., Lee, K., y Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Dey *et al.*, 2020] Dey, R., Sarddar, D., Sarkar, I., Bose, R., y Roy, S. (2020). A literature survey on sentiment analysis techniques involving social media and online platforms. *International Journal of Scientific and Technology Research*, 9:1–8.

[Dojo, 2022] Dojo, D. S. (2022). 6 aplicaciones del procesamiento del lenguaje natural.

[Elia, 2023] Elia, F. (2023). Sentiment analysis dictionaries. *Baeldung*. Accedido el día Mes Año.

[Emol, 2020] Emol (2020). Universitarios congelan carreras por efectos del coronavirus.

[Gajjar, 2020] Gajjar, K. (2020). Cluster analysis with dbscan: Density-based spatial clustering of applications with noise. Accessed on: [Fecha de acceso].

[Gandhi, 2018] Gandhi, R. (2018). Naive bayes classifier.

[Hoffstaetter, 2022] Hoffstaetter, S. (2022). Pytesseract. `https://github.com/madmaze/pytesseract`. GitHub repository.

[IBM, 2021] IBM (2021). Natural language processing.

[Institute, 2021] Institute, A. (2021). Allennlp models: A collection of pre-trained models for natural language processing. `https://github.com/allenai/allennlp-models`.

[Interactive Chaos,] Interactive Chaos. Tutorial de machine learning - t-sne.

[JavaTpoint,] JavaTpoint. Principal component analysis.

[Kassambara, ceso] Kassambara, A. (Año de acceso). Cluster validation statistics: Must-know methods.

[Khurana *et al.*, 2023] Khurana, D., Koli, A., Khatter, K., y Singh, S. (2023). Natural language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications*, 82(3):3713–3744.

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

[madmaze, 2022] madmaze (2022). Pytesseract. `https://github.com/madmaze/pytesseract`.

[MathWorks,] MathWorks. Support Vector Machine (SVM). [Accessed 24-Jul-2023].

[Medhat *et al.*, 2014] Medhat, W., Hassan, A., y Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093–1113.

[Mehta y Pandya, 2020] Mehta, P. y Pandya, S. (2020). A review on sentiment analysis methodologies, practices and applications. 9.

[Miller *et al.*, 1990] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., y Miller, K. J. (1990). Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

[Nigam *et al.*, 1999] Nigam, K., Lafferty, J., y McCallum, a. (1999). Using maximum entropy for text classification. En *IJCAI-99 workshop on machine learning for information filtering*, volumen 1, pp. 61–67. Stockholom, Sweden.

[OCR, 2023] OCR, T. (2023). Tesseract ocr. `https://github.com/tesseract-ocr/tesseract`.

[OpenAI, 2023] OpenAI (2023). Gpt-4 technical report. *ArXiv*, abs/2303.08774.

[OpenCV, 2023] OpenCV (2023). Opencv-python. `https://github.com/opencv/opencv-python`.

[Organization, 2022] Organization, W. H. (2022). Mental health and covid-19: early evidence of the pandemic's impact: scientific brief, 2 march 2022. Technical documents.

[Patlolla, 2018] Patlolla, C. R. (2018). Understanding the concept of hierarchical clustering technique. *Towards Data Science*.

[PatrickFarley y eric urban, 2023] PatrickFarley y eric urban (2023). Ocr: reconocimiento óptico de caracteres. `https://learn.microsoft.com/es-es/azure/ai-services/computer-vision/overview-ocr`.

[Pérez *et al.*, 2021] Pérez, J. M., Giudici, J. C., y Luque, F. (2021). pysentimiento: A python toolkit for sentiment analysis and socialnlp tasks.

[Qiu *et al.*, 2020] Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., y Huang, X. (2020). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

[Quinlan, 1986] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106.

[Ruozzi, 2018] Ruozzi, N. (2018). CS6347 - Statistical Methods in AI and Machine Learning, Unit 2: Graphical Models, 2.1 Bayesian Networks. University of Texas at Dallas, CS 6347: Statistical Methods in Artificial Intelligence and Machine Learning, Class Notes.

[Salakhova *et al.*, 2022] Salakhova, V. B., Shukshina, L. V., Belyakova, N. V., Kidinov, A. V., Morozova, N. S., y Osipova, N. V. (2022). The problems of the covid-19 pandemic in higher education. *Frontiers in Education*, 7.

[Salzberg, 1994] Salzberg, S. L. (1994). C4.5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. *Machine Learning*, 16(3):235–240.

[Sangüeza y Terrádez Gurrea, 2016] Sangüeza, Ramon. Carracedo Garnateo, P. y Terrádez Gurrea, M. (2016). Clasificación árboles de decisión. Appears in Collections: Recursos de aprendizaje UOC.

[Seminara, 2021] Seminara, M. P. (2021). De los efectos de la pandemia covid -19 sobre la deserción universitaria: desgaste docente y bienestar psicológico estudiantil. 33:402–421.

[Shivaprasad y Shetty, 2017] Shivaprasad, T. K. y Shetty, J. (2017). Sentiment analysis of product reviews: A review. En *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pp. 298–301.

[Sngular, 2023] Sngular (2023). Crisp-dm: La metodología para poner orden en los proyectos. *Sngular Blog*.

[Srivastava, 2021] Srivastava, A. (2021). K-means clustering. Accessed on: [Fecha de acceso].

[Tercera, 2020] Tercera, L. (2020). La crisis golpea a universidades: aumenta deserción y 50 mil deudores del cae piden apoyo.

[Tesseract OCR Developers, 2022] Tesseract OCR Developers (2022). Tesseract ocr. `https://github.com/tesseract-ocr/tesseract`. GitHub repository.

[Thorn, 2018] Thorn, J. (2018). Decision trees explained. *Towards Data Science*.

[Timarán-Pereira *et al.*, 2016] Timarán-Pereira, S. R., Hernández-Arteaga, I., Caicedo-Zambrano, S. J., Hidalgo-Troya, A., y Alvarado-Pérez, J. C. (2016). El proceso de descubrimiento de conocimiento en bases de datos. En *Descubrimiento de patrones de desempeño académico con árboles de decisión en las competencias genéricas de la formación profesional*, pp. 63–86. Ediciones Universidad Cooperativa de Colombia, Bogotá.

[Virmani, 2022] Virmani, S. (2022). Rule-based classifier – machine learning.