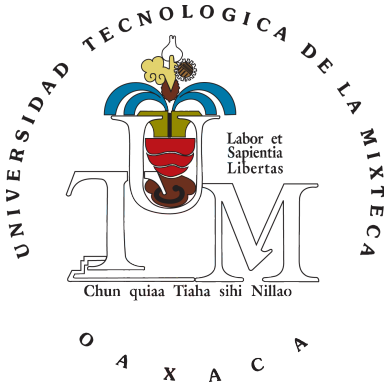


UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

DETECCIÓN DE ANSIEDAD A PARTIR DEL ANÁLISIS DE TEXTOS CORTOS MEDIANTE APRENDIZAJE SUPERVISADO Y SEMISUPERVISADO

TESIS

PARA OBTENER EL TITULO DE

INGENIERO EN COMPUTACIÓN

PRESENTA:

BARUC CISNEROS CRUZ

DIRECTOR DE TESIS:

DR. CHRISTIAN EDUARDO MILLÁN HERNÁNDEZ

CO-DIRECTOR:

DR. EDUARDO SÁNCHEZ SOTO

H. CD. DE HUAJUAPAN DE LEÓN, OAXACA.

FEBRERO DE 2025

Dedicado a mis papás, Gersain Cisneros Estrada y Leticia R. Cruz Montesinos, quienes con su dedicación y esfuerzo hicieron posible que este sueño se convirtiera en realidad.

Dedicado a mi hermano Pablo, a mi tío Ómar, a mis tías Agueda, Érica y Concepción, y a mi prima Melisa. Gracias por su apoyo constante y palabras alentadoras.

Agradecimientos

A mis padres, por darme la oportunidad de estudiar una ingeniería. Siempre estaré muy agradecido con mi padre por su entrega, ejemplo y perseverancia de todos los días para apoyarme incondicionalmente. A mi madre por su cuidado, cariño y esfuerzo incansable para proporcionarme siempre lo esencial en mi crecimiento académico. A los dos por estar siempre conmigo y brindarme la fortaleza necesaria para superarme cada día. Por todo esto y muchas cosas más, gracias.

Expreso mi más profundo agradecimiento al Dr. Christian Eduardo Millán Hernández, director de esta tesis, por permitirme desarrollar este proyecto bajo su guía. Gracias por su tiempo, conocimiento, consejos y paciencia a lo largo de esta tesis. Todo mi respeto y admiración hacia usted.

Un agradecimiento especial al Dr. Eduardo Sánchez Soto, por sus consejos y retroalimentación en el proceso de la realización de esta tesis. A la especialista Denisse Millán Hernández, Licenciada en Psicología y Maestra en Educación, por su ayuda con lo relacionado en el área de la Ansiedad como trastorno mental.

A mi tío Ómar y mi tía Agueda, por sus sabios consejos y apoyo de siempre, asimismo quiero agradecer a mi amigo Jairo CC y a una persona muy especial, Citlali RH, por estar siempre presentes, sobre todo en estos últimos años. Gracias por su complicidad, apoyo, compañía y los buenos momentos compartidos.

A la Universidad Tecnológica de la Mixteca, por brindarme la invaluable oportunidad de realizar mis estudios y por inculcar en mí un profundo orgullo de haber pertenecido a esta institución académica.

Resumen

La ansiedad, como respuesta emocional, se manifiesta ante situaciones de incertidumbre, estrés o miedo y su intensidad puede variar. Aunque esta respuesta es natural, puede convertirse en un trastorno significativo de salud mental cuando se presenta de manera intensa y recurrente. Este tipo de padecimiento impacta negativamente diversos aspectos de la vida cotidiana de las personas. En los últimos años, la prevalencia de la ansiedad ha aumentado considerablemente, lo que resalta en la necesidad de desarrollar herramientas de detección que apoyen a los profesionales de la salud mental.

El aprendizaje automático (ML, por sus siglas en inglés de *Machine Learning*), combinado con el Procesamiento de Lenguaje Natural (NLP, por sus siglas en inglés de *Natural Language Processing*) ofrecen una solución prometedora para abordar esta problemática. Estas tecnologías permiten identificar patrones y señales emocionales en la forma en que las personas se expresan por escrito, haciendo posible el análisis y la clasificación de textos. Este tipo de procesamiento incluye múltiples etapas: preprocesamiento de datos, extracción y selección de características relevantes, y la aplicación de algoritmos de ML que realizan las predicciones necesarias.

En esta tesis se propone el desarrollo de modelos de ML orientados a la identificación de ansiedad en textos escritos por personas, utilizando NLP. Este enfoque busca no solo analizar la forma en que las personas se expresan, sino también identificar patrones lingüísticos y emocionales que puedan ser indicadores de ansiedad.

Para alcanzar este objetivo, se emplean dos enfoques principales de aprendizaje: supervisado y semisupervisado. Estos enfoques se complementan mediante la implementación de diversos métodos de preprocesamiento para preparar los textos de manera efectiva, y técnicas de extracción de características basadas en modelos de lenguaje, que permiten representar la información contenida en los textos. Además, se evalúan diferentes algoritmos de ML para determinar cuáles ofrecen los mejores resultados en la tarea de clasificación.

El desarrollo de estos modelos representa una contribución al campo de la detección de trastornos psicológicos mediante el análisis automatizado de textos. La combinación de ML y

NLP proporcionan herramientas útiles para los profesionales de la salud mental y abren nuevas posibilidades para la investigación en el área de la salud emocional. Este trabajo busca sentar las bases para la implementación de sistemas que ayuden en la identificación temprana de ansiedad, promoviendo intervenciones más oportunas y efectivas.

Índice general

Índice de figuras

Índice de Tablas

Capítulo 1

Introducción

1.1. Antecedentes

La salud puede entenderse como la ausencia de algún malestar físico o enfermedad. Sin embargo, para que un individuo se considere saludable, debe de gozar de un estado completo de bienestar no solo físico, sino también social y mental (OMS, 2014). En los últimos años, sobre todo durante y posterior a la pandemia por COVID-19, se ha considerado un reto el mantener una salud mental apropiada. De acuerdo con la Organización Mundial de la Salud (OMS), poseer una salud mental adecuada implica tener una respuesta idónea ante situaciones estresantes de la vida cotidiana, así como un rendimiento productivo en aspectos laborales y sociales (OMS, 2017). No obstante, existen padecimientos que amenazan la salud mental, por ejemplo, ansiedad, depresión y estrés. La ansiedad es una emoción humana normal y natural; se trata de una reacción que se experimenta ante situaciones de estrés, peligro o incertidumbre, es una respuesta adaptativa que nos ayuda a enfrentar situaciones desafiantes que alteran el bienestar personal. (APA, 2017; CUN, 2020). La ansiedad se caracteriza por sentimientos de preocupación, nerviosismo, tensión y aprehensión, que puede manifestarse de diferentes maneras, desde síntomas físicos como palpitaciones y sudoración hasta síntomas emocionales como inquietud e irritabilidad (Beidel and Brooke, 1997).

El miedo y la ansiedad están estrechamente relacionados, el primero representa una respuesta inminente ante una amenaza percibida en el presente, mientras que la ansiedad está dirigida hacia el futuro, presta atención anticipada a una posible amenaza (OMS, 2022a). La ansiedad ayuda a prepararse y practicar acciones prudentes, de tal forma que se van adoptando medidas oportunas frente a situaciones potencialmente peligrosas; clínicamente la ansiedad es el miedo sin saber a qué (Fernández López et al., 2012). Ahora bien, cuando el miedo y la preocupación son persistentes alcanzando niveles muy altos, de tal forma que debilitan e interfieren las actividades diarias del individuo, se convierte en un trastorno (APA, 2017; OMS, 2022d).

Un trastorno mental es una perturbación de la cognición; es decir, de aspectos como el razonamiento, la memoria y la resolución de problemas, se trata de una alteración en la regulación

de emociones y/o el comportamiento, que se refleja en la disfunción de procesos psicológicos, biológicos o del desarrollo (OMS, 2022b). La OMS menciona que estos trastornos se tratan de una condición de salud diagnosticable y diferente de sentir tristeza, estrés o miedo de manera normal. Por lo tanto, en un trastorno de ansiedad los síntomas son suficientemente graves como para provocar angustia y afectación importante en el estado de ánimo de un individuo causando una discapacidad funcional importante (OMS, 2022d). En consecuencia, padecer un trastorno mental origina un daño significativo a nivel personal, familiar, social, educativo o áreas de productividad.

El diagnóstico tradicional de la ansiedad se apoya en la agrupación de síntomas físicos y conductuales. Además, este diagnóstico suele depender de evaluaciones clínicas y entrevistas con profesionales de la salud mental, lo que puede ser costoso, llevar tiempo y estar sujeto a sesgos (Tasnim et al., 2023). Sin mencionar, que cuando un individuo recurre con un especialista es probablemente porque su salud ya esta siendo afectada por problemas de ansiedad o por cualquier otra enfermedad mental. Por lo tanto, es esencial contar con métodos y herramientas precisas y accesibles para la detección de la ansiedad. Estos recursos permiten a los profesionales de la salud aplicar estrategias efectivas para prevenir la progresión de la ansiedad y evitar que se convierta en un trastorno de ansiedad.

En los últimos años, se han realizado investigaciones dedicadas al desarrollo de modelos de aprendizaje automático (ML, por sus siglas en inglés de *Machine Learning*) para identificar a personas en riesgo de padecer algún trastorno de ansiedad. Dichos modelos pueden ser útiles como herramientas en la detección de estos trastornos y otros padecimientos a través del análisis de diferentes tipos de datos, como por ejemplo: características fisiológicas (Elgendi et al., 2022), audio (Tasnim et al., 2023) o texto. Los modelos son capaces de detectar signos de angustia, inestabilidad emocional u otros indicadores de condiciones de salud mental (Nova, 2023). También logran detectar señales o cambios en los patrones del lenguaje que indican problemas mentales emergentes y monitorear el progreso del paciente con el tiempo. Además, reducen el estigma asociado con la salud mental, ofrecen un análisis objetivo y automatizado, lo que permite a las personas expresar sus pensamientos, sentimientos y emociones de manera más abierta sin temor a ser juzgadas por otro ser humano (Nova, 2023).

El uso del Procesamiento de Lenguaje Natural (NLP, por sus siglas en inglés de *Natural Language Processing*), que es una rama de la inteligencia artificial, se enfoca en la interacción entre computadoras y el lenguaje humano y ha posibilitado el desarrollo de nuevas herramientas para el análisis de texto (Sarkar, 2016). Actualmente, gracias a fuentes de texto como las redes sociales, donde las personas expresan sus estados de ánimo y formas de pensar, surge la oportunidad de que, con la ayuda del NLP y la accesibilidad a esos datos de texto, se pueda realizar la detección de ansiedad de manera efectiva. En este contexto, diversos estudios han explorado

diferentes metodologías para identificar y medir niveles de ansiedad a partir del análisis de textos, utilizando técnicas que van desde estrategias de preprocesamiento de textos hasta modelos de ML basados en diferentes enfoques.

En el trabajo de Byers et al. (2023) se propone la implementación de un modelo de ML para la detección de ansiedad en estudiantes. El modelo se desarrolla a partir de datos obtenidos de la transcripción de entrevistas realizadas a estudiantes sobre sus experiencias con la ansiedad. Las entrevistas fueron divididas en sentencias que permitieron crear un corpus de 1,187 documentos que fueron etiquetados por expertos en salud mental para identificar cuatro niveles de ansiedad: sin ansiedad, baja, media y alta. Durante el preprocesamiento de los datos se realizó la eliminación de signos de puntuación y conversión a letras minúsculas. Para la extracción de características se utilizó N-Gramas y posteriormente se realizó una selección de características utilizando *LASSO*, X^2, $L1$ y algoritmos basados en árboles. Durante el desarrollo del modelo se utilizó una búsqueda en malla para determinar los mejores hiperparámetros de los algoritmos: Bayes Ingenuo (NB, por sus siglas en inglés de *Naive Bayes*), Árbol de decisión (DT, por sus siglas en inglés de *Decision Tree*), Máquina de Soporte Vectorial (SVM, por sus siglas en inglés de *Support Vector Machine*) y Regresión Logística. La SVM con las características seleccionadas por X^2, obtuvo los mejores resultados con un intervalo de confianza de 7.2 % y una exactitud de 59.1 %

En el trabajo de Nova (2023), se realizó una clasificación de trastornos mentales a partir de análisis de textos extraídos de la red social Reddit. Específicamente de cuatro categorías o subreddits donde se comenta sobre el trastorno límite de personalidad, trastorno bipolar, depresión y ansiedad. Además, se consideró un conjunto que abarcaba esquizofrenia y enfermedad mental. Los textos están compuestos por el título y el contenido de las publicaciones. En el preprocesamiento se eliminaron URL's, signos de puntuación y palabras vacías, así como la extracción de características aplicando el método TF-IDF (por sus siglas en inglés de *Term Frequency - Inverse Document Frequency*). Los algoritmos que se utilizaron fueron: LightGBM (acrónimo en inglés de *Light Gradient Boosting Machine*), Bayes Ingenuo Multinomial y Perceptrón Multicapa (MLP, por sus siglas en inglés de *Multilayer Perceptron*). Sin embargo, LightGBM obtuvo los mejores resultados con una exactitud de 72.4 % en títulos y 77 % en el texto de las publicaciones.

En otro trabajo, realizado por Yu et al. (2023), se desarrolló un modelo para predecir el estado de ansiedad de usuarios de la red social Sina Weibo en China. Para lo cual, se invitó a los usuarios a llenar una escala de autoevaluación de ansiedad (SAS, por sus siglas en inglés de *Self-Rating Anxiety Scale*) y posteriormente se recopilaron todos los textos originales de sus publicaciones en la red social. Esto con la finalidad de construir un conjunto de datos etiquetados para el entrenamiento del modelo. Durante la extracción de características, los textos se

analizaron mediante el diccionario SC-LIWC (por sus siglas en inglés de *Simplified Chinese-Linguistic Inquiry and Word Count*), para contar el número de palabras de cada tipo. Dicho diccionario puede calcular el porcentaje de múltiples palabras con significado psicológico o lingüístico, que incluyen palabras de dimensiones emocionales, cognitivas y sociales. Se emplearon cuatro algoritmos de regresión para generar el modelo: Regresión Lineal (LR, por sus siglas en inglés de *Linear Regression*), Máquina de Vectores de Soporte (SVR, por sus siglas en inglés de *Support Vector Regression*), Regresor *XGBoost* y Regresor *AdaBoost*. Sin embargo, el algoritmo que dio mejor resultados fue el Regresor *XGBoost* con un Error Absoluto Medio de 4.57 y una correlación de Pearson de 0.32 entre los resultados reales y los obtenidos por el modelo.

En la investigación desarrollada por Saifullah et al. (2021), se realizó la detección de ansiedad en comentarios tomados de YouTube®. Se consideró que los datos que no representaban un caso de ansiedad están asociados a sentimientos positivos y con ansiedad aquellos que expresan sentimientos negativos. Las etapas de preprocesamiento de esta investigación incluyeron, la detección de emociones basada en el análisis de sentimientos (positivos y negativos) y pruebas de validación. Durante el preprocesamiento se realizó la tokenización, filtrado, derivación, etiquetado y conversión de emoticones a cada uno de los textos. Para el análisis fueron evaluados seis diferentes algoritmos: K Vecinos más Cercanos (K-NN, por sus siglas en ingles de *K Nearest Neighbors*), Bernoulli, DT, SVM, Bosque Aleatorio (RF, por sus siglas en inglés de *Random Forest*) y XGBoost. Entre los que destacó RF con una exactitud de 84.99 % al extraer características aplicando el método de Bolsa de Palabras y 82.63 % con el método TF-IDF.

En el trabajo realizado por Asra et al. (2021), se propone DASentimental un modelo de ML semisupervisado para identificar niveles de Depresión, Ansiedad y Estrés (DAS, por sus siglas en inglés de *Depression, Anxiety and Stress*) en textos. Para la creación del modelo se destaca el uso de cuatro conjuntos de datos. Un conjunto de datos compuesto por una colección de recuerdos emocionales (ERT, por sus siglas en inglés de *Emotional Recall Task*). Así también un conjunto de datos que mapea la memoria semántica humana a través de asociaciones libres (The Small World of Words). Asimismo, se consideró un conjunto de datos que incluía normas de valencia y excitación de diferentes palabras y para poner a prueba el modelo se utilizó un corpus sobre notas de suicidio. En el proceso, se realizó la limpieza de datos y representación vectorial de las variables (características), así como de las respuesta de los niveles DAS, en donde los vectores de características fueron sometidos a una regularización $L2$. En seguida se realizó el entrenamiento, validación cruzada y selección del modelo de regresión de mejor rendimiento para estimar los niveles DAS a partir de los datos ERT. Posteriormente, se estimaron los niveles de DAS de las notas de suicidio analizando las secuencias de palabras emocionales en cada una, y se validó el etiquetado predicho por DASentimental a través de normas afectivas independientes. Los algoritmos empleados para el desarrollo del modelo fueron: DT, MLP y

una Red Neuronal Recurrente (LSTM, por sus siglas en inglés de *Long Short Term Memory*). Cada uno de estos modelos fue evaluado en términos de Error Cuadrático Medio (MSE, por sus siglas en inglés de *Mean Square Error*) y correlación de Pearson. Los resultados indicaron que MLP, entrenado con secuencias de palabras, proporciona las mejores predicciones, obteniendo una correlación de Pearson en validación cruzada de 0.7 para depresión, 0.44 para ansiedad y 0.52 para estrés.

En el trabajo de Heri Cahyana et al. (2022) se realiza la detección de odio mediante el análisis de textos recuperados de comentarios de YouTube. Los textos permitieron la creación de un corpus con 13,169 documentos de los cuales 2,370 fueron etiquetados por expertos para clasificar tres estados de odio: "Mucho odio", "Odio" y "Sin odio". Durante el preprocesamiento de los textos se utilizó TF-IDF como método de extracción de características y se desarrollo un modelo de ML a partir de la implantación del algoritmo K-NN con un enfoque de aprendizaje semisupervisado, en donde, se tomaron los textos etiquetados para entrenar un primer modelo y pseudoetiquetar los demás textos. Para mejorar la clasificación se definió un umbral mínimo del 80 % de los votos a favor de alguna de las tres clases, para considerar un nuevo elemento como parte del conjunto de entrenamiento o de lo contrario repetir el proceso de pseudoetiquetado. Al final el modelo se evaluó con un conjunto de datos de 1,317 ejemplos etiquetados por los expertos, obteniendo como resultado 59.68 % en exactitud.

En las investigaciones revisadas sobre la detección de ansiedad mediante ML se ha identificado el uso de distintas características para la creación de conjuntos de datos, entre las más destacadas están el uso de señales fisiológicas como las frecuencias cardíacas, respiración y señales de electrocardiogramas, características extraídas del habla y por último las características extraídas de textos. En cuanto al tipo de ML, se utiliza principalmente el aprendizaje supervisado y semisupervisado. Una ventaja que tiene el uso de textos escritos es que permite que las personas expresen sus pensamientos de forma más abierta, y adicionalmente el uso de una computadora evita sentirse juzgado por otra persona, que en este caso será el experto de la salud metal (Nova, 2023).

De acuerdo con los trabajos previos revisados, existen diversas opciones para abordar la detección de ansiedad a partir del análisis de textos. Los trabajos presentan diferentes modelos de lenguaje y algoritmos o enfoques de ML. Entre estos enfoques, el aprendizaje supervisado y el semisupervisado destacan como áreas prometedoras. Esto señala una área de oportunidad para la implementación de herramientas de NLP, las cuales permiten explorar los beneficios y fortalezas específicas que estos dos tipos de aprendizaje pueden aportar. Por lo tanto, en esta tesis se aborda principalmente el estudio de los enfoques supervisado y semisupervisado.

1.2. Planteamiento del Problema

El problema del trastorno de ansiedad genera una discapacidad funcional significativa. Esto implica que quienes lo padecen enfrentan repercusiones en diversas áreas de su vida, por ejemplo, en el ámbito laboral, se puede observa una disminución considerable en la productividad; en lo social, se tiende a evitar eventos o convivencias sociales debido a una timidez excesiva; en lo académico, se evidencia una pérdida de interés, un bajo rendimiento y dificultades para concentrarse (Mayo Clinic, 2021). Por consiguiente, es crucial realizar la detección temprana de la ansiedad con el fin de proporcionar el apoyo adecuado a la persona afectada y mejorar su calidad de vida. En este contexto, se han desarrollado herramientas computacionales enfocadas en la detección de ansiedad a través del análisis de textos y el ML, herramientas de apoyo eficaces para el profesional de la salud metal.

Las investigaciones basadas en la detección de ansiedad proponen diversos enfoques. En particular, las que analizan de manera automática los textos escritos por los individuos utilizan NLP y ML; en general, se plantea un primer enfoque para la construcción de herramientas computacionales para la detección de ansiedad que utilizan algoritmos de aprendizaje supervisado, es decir, que requiere de datos etiquetados, se utiliza algún examen o instrumento adicional para determinar la presencia de ansiedad en la persona y etiquetar la muestra de entrenamiento; mientras que, en otras propuestas, no es necesario contar con etiquetas en los textos, en consecuencia se utilizan técnicas de aprendizaje semisupervisado, esto es, se utiliza alguna técnica de agrupación, por ejemplo, en el análisis de sentimientos, para determinar si la polaridad del texto escrito es positivo o negativo y de esta forma asociarlo a la salud mental.

Por lo tanto, en esta tesis se plantea el problema de la detección de ansiedad a partir de textos escritos por una persona. La propuesta de este trabajo parte de las investigaciones sobre NLP basadas en el análisis de la forma de escribir de una persona, donde se afirman y se muestran descubrimientos sobre estilos de escritura, análisis de sentimientos, detención del engaño, identificación de rasgos de personalidad, entre otros (H. Andrew et al., 2013).

Entonces, se plantea la siguiente pregunta de investigación: ¿Es posible detectar la presencia de ansiedad en una persona mediante aprendizaje automático a partir del análisis de textos cortos?

1.3. Justificación

La Organización Panamericana de la Salud (OPS) junto con la OMS revelan que los trastornos de ansiedad son el segundo trastorno mental que más afecta en la mayoría de los países de la Región de las Américas (OMS y OPS, 2018). Asimismo, la OMS informa que de acuerdo

con el Instituto de Sanimetría y Evaluación Sanitaria, que en el 2019 una de cada ocho personas en el mundo, es decir, aproximadamente novecientos setenta millones de personas padecían un trastorno mental, siendo los más comunes la ansiedad y la depresión (OMS, 2022d). Para el 2020 estas cifras aumentaron considerablemente debido a la pandemia por COVID-19, las estimaciones mostraron un incremento del 26 % y 28 % para la ansiedad y depresión, respectivamente (OMS, 2022c).

Por otro lado, la OMS en conjunto con la OPS, informaron que de acuerdo con los Centros para el Control y Prevención de Enfermedades (CDCP, por sus siglas en inglés de *Centers for Disease Control and Prevention*), en Estados Unidos la cantidad de adultos entre 18 y 29 años, con síntomas de un trastorno de ansiedad y depresión aumento del 36.4 % al 42.5 %, mientras que el porcentaje de personas que reportaron atención de salud mental no cubierta, incremento del 9.2 % al 11.7 % entre agosto del 2020 y febrero del 2021 debido a la pandemia por COVID-19. Por otro lado señalan que en una encuesta realizada en el 2021, más de la mitad de los participantes en países como Chile, Brasil, Perú y Canadá expusieron que su salud mental había empeorado desde el comienzo de la pandemia (OMS y OPS, 2021). En México, de acuerdo con el Instituto Nacional de Estadística y Geografía (INEGI) (2021) y los resultados obtenidos de la primera Encuesta Nacional de Bienestar Autorreportado (ENBIARE), muestran que 19.3 % de la población adulta presentó síntomas de ansiedad severa, mientras otro 13.3 % reveló síntomas de ansiedad mínima o en algún grado.

Sin duda la ansiedad se ha convertido en un problema de salud mental significativo en la sociedad actual. Las estadísticas revelan que millones de personas a nivel global experimentan síntomas de ansiedad, lo que tiene un impacto tangible en su bienestar emocional y calidad de vida, sin embargo, lo que resulta aún más alarmante, es la tendencia ascendente en la prevalencia de este padecimiento observada en los últimos años, lo que subraya la urgente necesidad de abordar este problema de manera más efectiva.

El uso de herramientas computacionales de apoyo para la detección de ansiedad permite a los expertos de la salud mental realizar diagnósticos oportunos. La presente investigación explora las propuestas del estado del arte, con la finalidad de comparar los métodos basados en aprendizaje supervisado y semisupervizado para la detección de ansiedad. Por otro lado, la implementación de herramientas computacionales que analizan los textos escritos por una persona mediante procesamiento de lenguaje, requiere utilizar un modelo de lenguaje para representar los textos en un formato propio, tanto para los algoritmos de aprendizaje supervisado como para los semisupervisados. Por lo que los resultados de este trabajo permitirán conocer la eficacia de los modelos de lenguaje propuestos en el estado del arte.

Dado que es necesario la adquisición de un conjunto de textos escritos por una persona,

en este trabajo se propone la creación de un conjunto de datos etiquetados (con la presencia o ausencia de ansiedad), es decir, que implica la aplicación de algún examen o instrumento para la detección de ansiedad. Por lo tanto, el conjunto de datos resultante podrá ser utilizado en futuros trabajos relacionados con la detección de ansiedad a partir de textos cortos.

1.4. Hipótesis

Las características léxicas de los textos cortos escritos por una persona tienen influencia para detectar la ansiedad, tanto en algoritmos de aprendizaje supervisado como en algoritmos de aprendizaje semisupervisado.

1.5. Objetivos

En el marco de esta investigación, primero se expondrá el objetivo general que orienta el estudio, seguido por la definición de los objetivos específicos que constituyen los pasos necesarios para su consecución.

1.5.1. Objetivo general

Detectar la ansiedad a partir del análisis de textos cortos mediante aprendizaje supervisado y semisupervisado.

1.5.2. Objetivos específicos

1. Realizar una investigación documental sobre aprendizaje supervisado y semisupervisado en algoritmos de aprendizaje automático para la detección de ansiedad mediante procesamiento de lenguaje natural.

2. Creación de un corpus a partir de textos cortos para la detección de ansiedad.

3. Proponer un método para la implementación de aprendizaje supervisado y semisupervisado para la detección de ansiedad mediante análisis de textos cortos.

4. Evaluar los modelos obtenidos de distintos algoritmos de aprendizaje supervisado y semisupervisado implementados con el método propuesto para la detección de ansiedad mediante análisis de textos cortos.

5. Comparar los resultados de la evaluación de los algortimos de aprendizaje supervisado y semisupervisados para la detección de ansiedad mediante análisis de textos cortos.

1.6. Metas

1. Elaboración de un reporte de investigación documental sobre métodos que implementan un enfoque de aprendizaje supervisado y semisupervisado para la detección de ansiedad mediante análisis de textos cortos.

2. Implementación de modelos de aprendizaje automático utilizando métodos de aprendizaje supervisado con mejor desempeño de acuerdo al estado del arte.

3. Implementación de modelos de aprendizaje automático utilizando métodos de aprendizaje semisupervisado con mejor desempeño de acuerdo al estado del arte.

4. Elaboración de un reporte sobre la comparación de los resultados de la evaluación de los métodos implementados para la detección de ansiedad.

5. Elaboración del documento de tesis para la obtención del título de Ingeniero en Computación

1.7. Alcances y limitaciones

1. Los datos recolectados para la creación del corpus consisten únicamente en información de alumnos de la Universidad Tecnológica de la Mixteca durante el periodo del curso propedéutico de 2023. Por lo tanto, las observaciones obtenidas están limitadas a este conjunto específico de personas.

2. Los participantes son hablantes del idioma español, por lo que la redacción de los documentos que conforman el corpus está limitada a este idioma.

3. El tamaño del corpus puede influir en el rendimiento de algunos algoritmos de aprendizaje automático, particularmente en un enfoque de aprendizaje supervisado.

4. Los modelos creados en esta investigación no están destinados a sustituir el juicio de los profesionales en salud mental, en cambio, están diseñados para servir como una herramienta adicional que apoya el trabajo de los expertos.

5. Este trabajo se centra exclusivamente en el uso de una biblioteca de algoritmos de aprendizaje automático para generar modelos, sin incluir el desarrollo de un software.

6. El tiempo disponible es insuficiente para realizar pruebas adicionales con una mayor variedad de modelos de lenguaje y algoritmos de aprendizaje automático.

1.8. Estructura de la tesis

La presente investigación se enfoca en la detección de ansiedad a través del análisis de textos, utilizando métodos de ML tanto supervisado como semisupervisado. El estudio explora cómo estos enfoques pueden ser empleados para identificar niveles de ansiedad en textos, evaluando la eficacia de cada método en la clasificación de contenido textual relacionado con la ansiedad.

El proceso de investigación descrito en el resto del documento está organizado en varias secciones, siguiendo la estructura que se presenta a continuación. Esta disposición facilita la comprensión del estudio al detallar cada aspecto del proceso de investigación.

En el *Capitulo 2. Marco Teórico*, se establecen los principios del aprendizaje supervisado y semisupervisado. Se examinan los modelos de lenguaje utilizados en el estudio. Además, se detallan los algoritmos de ML, proporcionando el contexto necesario para entender los métodos utilizados en la detección de ansiedad a través del análisis de textos.

En el *Capítulo 3. Método Propuesto*, se ofrece una descripción detallada de los pasos seguidos para llevar a cabo los experimentos correspondientes a los enfoques supervisado y semisupervisado. Se aborda el proceso de creación de la base de datos, que incluye la recolección y el preprocesamiento de los textos empleados en el estudio. Asimismo, se describe la implementación de los modelos de ML, especificando los parámetros utilizados para evaluar su rendimiento.

En el *Capítulo 4. Resultados*, se presentan los hallazgos obtenidos de los experimentos realizados con los enfoques supervisado y semisupervisado, siguiendo el método descrito anteriormente. Se ofrece un análisis detallado del desempeño de cada modelo en la detección de ansiedad y se discuten los resultados en el contexto de la investigación, comparando la eficacia de los modelos y evaluando su impacto en el análisis de textos.

En el *Capítulo 5. Conclusiones*, se resumen los hallazgos principales de la investigación, destacando los resultados obtenidos con los enfoques supervisado y semisupervisado en la detección de ansiedad. Se reflexiona sobre la efectividad de los modelos evaluados y su relevancia en el análisis de textos.

Capítulo 2

Marco teórico

En esta sección se abordan los temas fundamentales que constituyen el sustento teórico de la presente investigación, con un enfoque especial en el Procesamiento de Lenguaje Natural (NLP, por sus siglas en inglés de *Natural Lenguage Processing*) y el aprendizaje automático (ML, por sus siglas en inglés de *Machine Learning*) . Se realiza una exploración detallada de diversos algoritmos de ML y se destaca la relevancia de la selección de características como un elemento clave para la optimización del rendimiento de los modelos. Asimismo, se presentan las técnicas de evaluación implementadas y se efectúa una revisión de trabajos relacionados, estableciendo las bases conceptuales que guían el análisis de los resultados obtenidos en este estudio.

2.1. Procesamiento de Lenguaje Natural

El NLP representa un campo interdisciplinario en la confluencia de la informática, la inteligencia artificial y la lingüística computacional. Se enfoca principalmente en la comprensión y el análisis de las interacciones entre las computadoras y los lenguajes humanos, conocidos como lenguajes naturales. En esencia, el NLP se puede definir como la capacidad de las tecnologías computacionales para llevar a cabo de manera automática o semiautomática la interpretación y procesamiento del lenguaje utilizado por los seres humanos (Thanaki, 2017).

El NLP comprende un amplio espectro de tareas, entre las cuales se incluye el *Reconocimiento de voz*, que permite a los sistemas computacionales identificar y convertir la voz humana en texto o comandos accionables (Rabiner and Juang, 1993); así como la *Generación de Texto* que implica la creación automatizada de contenido textual coherente y significativo, emulando el estilo del lenguaje humano (Goodfellow et al., 2016); el *Análisis de Texto* o minería de texto, que se define como el proceso mediante el cual se extrae información a partir de datos textuales para transformar datos no estructurados en formas estructuradas, con el propósito de identificar patrones y generar conocimientos útiles para su aplicación en distintos contextos (Sarkar, 2016); la *Traducción Automática*, que transforma textos entre diferentes idiomas, procesando estructuras gramaticales y semánticas con el objetivo de obtener traducciones coherentes (Jurafsky

and Martin, 2008); el *Análisis de Sentimientos* que se utiliza para identificar, extraer y cuantificar emociones, opiniones y sentimientos expresados en textos; la *Generación de Resumenes*, en donde se identifica y extrae la información más relevante de un documento, sintetizándola de manera que se preserven los conceptos clave y la coherencia del contenido original (Sarkar, 2016); la *Clasificación de Texto*, que es el proceso de asignar documentos de texto a una o mas clases, basándose en un conjunto predefinido de clases (Sarkar, 2016).

En particular la *Clasificación de textos* resulta fundamental para esta tesis por enfocarse en la detección de ansiedad a través del análisis y categorización de textos, en la Figura 2.1 se ilustra la representación conceptual del proceso de clasificación de texto donde se observa cómo varios documentos pueden ser clasificados en diversas clases. Al inicio, todos los documentos están agrupados en un *corpus* (conjunto de datos, donde cada punto de datos individual se representa como un fragmento de texto único, denominado *documento* (Müller and Guido, 2016)). Posteriormente al pasar por un sistema de clasificación de textos, cada documento se asigna a una clase especifica previamente definida.

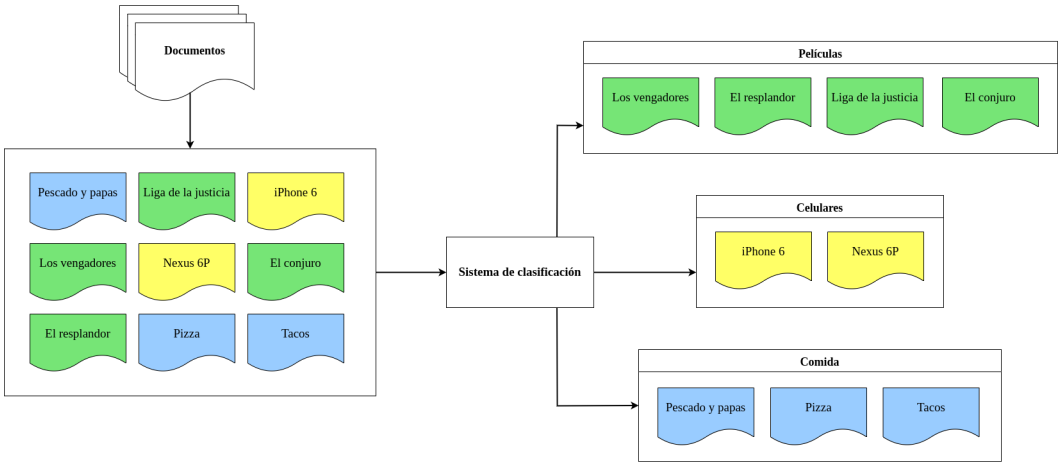


Figura 2.1: Representación conceptual de clasificación de textos (Sarkar, 2016).

Un *sistema de clasificación de texto* debe ser capaz de clasificar cada documento correctamente de acuerdo con las propiedades inherentes de dicho documento (Sarkar, 2016). La creación de un sistema de clasificación de texto consta de varias etapas que se desarrollan tanto en la fase de entrenamiento como en la fase de prueba. A continuación, se describe la secuencia de tareas básicas:

1. **Preparación de los conjuntos de datos de entrenamiento y prueba:** el primer paso consiste en seleccionar y organizar los datos que se utilizarán para entrenar el modelo y para evaluarlo.

2. **Normalización de texto:** esta etapa implica una serie de pasos para ordenar, limpiar y estandarizar datos textuales en un formato adecuado para ser utilizado como entrada en

aplicaciones de NLP, análisis y sistemas.

3. **Extracción de características:** a partir de los documentos normalizados, se extraen características relevantes que describen el contenido de manera numérica.
4. **Entrenamiento del modelo:** se selecciona un algoritmo de ML y se entrena el modelo utilizando los datos preparados.
5. **Predicción y evaluación del modelo:** se utiliza el modelo entrenado para predecir la clase de nuevos documentos y se evalúa su rendimiento en función de los datos de prueba.
6. **Implementación del modelo:** finalmente, el modelo se implementa para clasificar documentos nuevos en un entorno real.

En la fase de entrenamiento del sistema de clasificación de texto, los documentos se someten primero a un proceso de limpieza y estandarización en el módulo de normalización. A continuación, se aplican diversas técnicas para extraer características significativas, transformando los textos en matrices o vectores numéricos, ya que los algoritmos de ML requieren datos en este formato. Una vez obtenidas las características, se selecciona un algoritmo de ML para entrenar el modelo con los datos disponibles (Sarkar, 2016).

En la fase de predicción, el sistema sigue un enfoque similar. Los documentos del conjunto de prueba también pasan por los módulos de normalización y extracción de características, garantizando la consistencia en el preprocesamiento. Posteriormente, el modelo entrenado utiliza los patrones aprendidos durante la fase de entrenamiento para predecir la clase correspondiente a cada nuevo documento (Sarkar, 2016).

La Figura 2.2 ilustra este flujo de trabajo, destacando los pasos clave en las fases de entrenamiento y predicción. Cabe señalar que ambos procesos comparten dos componentes esenciales: la *normalización del texto* y la *extracción de características*.

2.2. Preprocesamiento de texto

De acuerdo con la sección anterior el preprocesamiento de texto es un paso fundamental en el análisis de datos textuales, ya que permite transformar texto no estructurado en un formato que puede ser interpretado y analizado de manera efectiva por modelos de ML. Este proceso abarca una serie de técnicas diseñadas para estructurar y estandarizar el texto, mejorando la capacidad de predicción de los modelos (Sarkar, 2016).

Por lo general, los corpus de texto y otros datos textuales en su formato original no están bien formateados ni estandarizados. El preprocesamiento, implica el uso de una variedad de técnicas

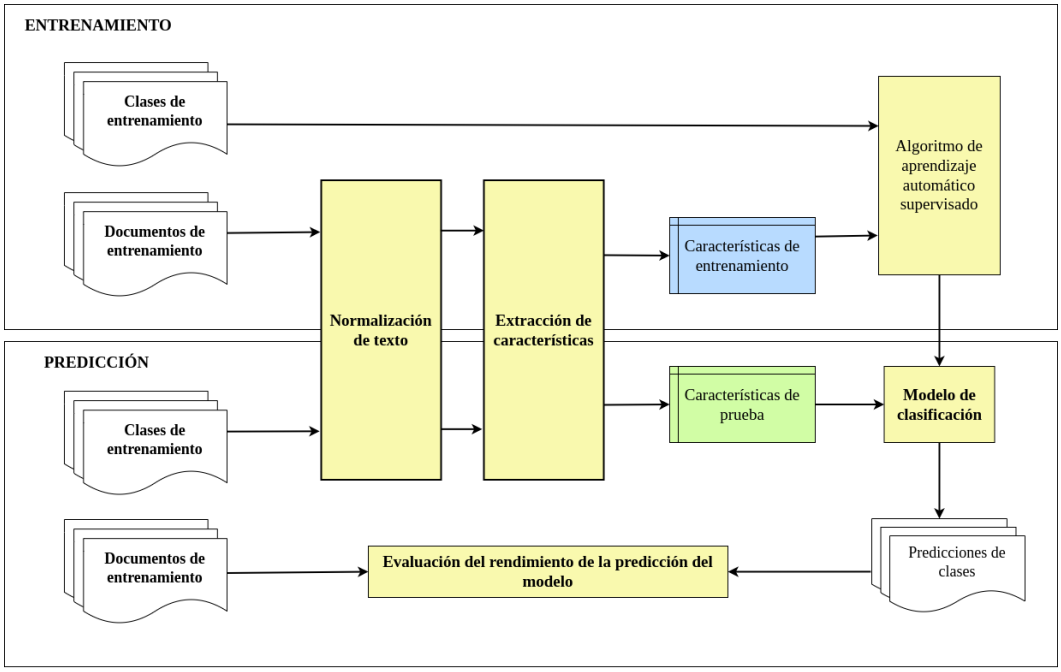


Figura 2.2: Etapas de implementación de un sistema automatizado de clasificación de textos (Sarkar, 2016).

para convertir el texto sin formato en secuencias bien definidas de componentes lingüísticos que tienen una estructura y una notación estándar. Un sistema de preprocesamiento de texto es una parte esencial de cualquier aplicación de NLP y análisis de texto. La razón principal es que todos los componentes textuales que se obtienen después del preprocesamiento (ya sean palabras, frases, oraciones o cualquier otro token) forman los bloques de construcción básicos de entrada que se introducen en las etapas posteriores de la aplicación que realiza análisis, incluido el aprendizaje de patrones y la extracción de información (Sarkar, 2016).

2.2.1. Normalización de texto

La normalización de texto es un paso esencial en el preprocesamiento de datos textuales, ya que prepara el texto para su análisis y para el uso en modelos de ML. Este proceso abarca varias técnicas que permiten transformar y limpiar el texto, garantizando que los datos estén en el formato adecuado para extraer información relevante (Sarkar, 2016). Entre las técnicas más comunes en la normalización de texto se encuentran:

- Limpieza de texto.
- Tokenización.
- Eliminación de caracteres especiales.
- Eliminación de palabras vacías.

La *limpieza de texto* consiste en eliminar tokens o caracteres extraños e innecesarios que suelen estar presentes en los texto. Estos elementos superfluos pueden afectar la calidad del

análisis posterior, por lo que es un paso crucial antes de aplicar otras técnicas (Sarkar, 2016).

La *tokenización* de texto es otra técnica fundamental dentro del proceso de normalización. Este paso consiste en dividir o segmentar oraciones en las palabras que las componen. Dado que una oración es una colección de palabras, la tokenización transforma esa colección en una lista que permite reconstruir y analizar la estructura del texto. Esto se puede realizar antes o después de la eliminación de caracteres no deseados, dependiendo del problema y los datos en cuestión (Sarkar, 2016).

La *eliminación de caracteres especiales* es otro paso clave. Este proceso implica remover símbolos y signos de puntuación que no aportan valor al análisis, especialmente en el contexto de extracción de información o creación de modelos de ML. A menudo, los caracteres especiales no influyen en el significado de las palabras, por lo que su eliminación simplifica el procesamiento del texto (Sarkar, 2016).

Finalmente, la *eliminación de palabras vacías*, que son aquellas con poco o ningún significado, es una práctica habitual. Estas palabras, que suelen aparecer con frecuencia en los textos, se descartan para centrarse en aquellas que aportan más significado y contexto, lo cual optimiza el análisis y facilita la extracción de características importantes (Sarkar, 2016).

2.2.2. Extracción de características

De acuerdo a la sección anterior, la normalización de texto es un proceso fundamental para garantizar la calidad de los textos y optimizar su análisis. Una vez que el texto ha sido limpiado y estructurado adecuadamente, el siguiente paso consiste en la extracción de características, donde se transforman estos datos normalizados en representaciones numéricas útiles para los modelos de ML. En esta sección, se profundizará en dicho proceso.

En el ámbito del ML, las características corresponden a atributos o propiedades medibles asociadas a cada observación dentro de un conjunto de datos. Estas características, frecuentemente de naturaleza numérica, pueden presentarse como valores absolutos o clasificarse en categorías, las cuales se convierten en variables binarias mediante un proceso conocido como codificación *one-hot*. La adecuada representación de las características es esencial, dado que los algoritmos de ML dependen de ellas para identificar patrones que puedan generalizarse a nuevos datos. (Sarkar, 2016).

Los algoritmos de extracción de características desempeñan un papel crucial en la preparación y transformación de los datos. Estos algoritmos están diseñados para identificar y resaltar

patrones, estructuras y representaciones significativas dentro de los datos. Actúan como herramientas de preprocesamiento, convirtiendo datos complejos o no estructurados en representaciones más simples, comprensibles y adecuadas para ser procesadas por los algoritmos de ML (Isabelle, 2006). Estas representaciones simplificadas pueden incorporar variables relevantes, atributos clave o dimensiones reducidas, optimizadas para la resolución de un problema específico (Isabelle, 2006).

Particularmente en el tratamiento de textos, se enfrenta el reto de transformar esa información no estructurada en vectores numéricos. Dicha conversión resulta imprescindible, ya que los algoritmos de ML, en su esencia, operan mediante la optimización matemática, buscando minimizar pérdidas y errores conforme aprenden a partir de los datos. Por tanto, la extracción adecuada de características numéricas a partir de textos es fundamental para garantizar un aprendizaje efectivo (Sarkar, 2016).

Una de las técnicas empleadas para la extracción de características en textos es el modelo de vector de términos. Este enfoque se define como un modelo matemático y algebraico que permite transformar y representar documentos de texto en vectores numéricos, donde cada dimensión corresponde a términos específicos presentes en el conjunto de documentos (Sarkar, 2016). Matemáticamente, esta técnica se puede expresar de la siguiente forma:

Supongamos que tenemos un documento D en un espacio vectorial de documentos EV. El número de dimensiones o columnas asociado a cada documento en este espacio corresponderá al número total de términos o palabras distintos presentes en todos los documentos del espacio vectorial. Así, el espacio vectorial puede ser denotado como

$$EV = \{W_1, W_2, ..., W_n\}$$

donde hay n palabras distintas en todos los documentos. Ahora podemos representar el documento D en este espacio vectorial como

$$D = \{w_{D1}, w_{D2}, ..., w_{Dn}\}$$

donde w_{Dn} denota el peso de la palabra n en el documento D. Este peso es un valor numérico y puede representar cualquier cosa, desde la frecuencia de esa palabra en el documento hasta la frecuencia promedio de ocurrencia (Sarkar, 2016).

Modelos de lenguaje

En el contexto del análisis y clasificación de textos, los modelos de lenguaje son esenciales para la extracción de características. Algoritmos como los modelos de Bolsa de Palabras, N-

Gramas y TF-IDF ofrecen técnicas para convertir el contenido textual en estructuras numéricas que preservan información relevante. Estos modelos permiten no solo la representación simplificada del texto, sino también una comprensión de los patrones y relaciones subyacentes en los datos, lo que facilita su análisis y mejora el rendimiento de los sistemas de ML. A continuación, se describen cada uno de estos modelos:

Bolsa de palabras

La Bolsa de Palabras (BoW, por sus siglas en inglés de *Bag Of Words*) es una técnica simple pero efectiva que descarta la mayor parte de la estructura del texto de entrada, como los párrafos, oraciones y formato, y solo se cuenta la frecuencia con la que aparece cada palabra en cada texto del corpus (Müller and Guido, 2016).

La representación de BoW para un corpus de documentos consta de tres pasos (Figura 2.3):

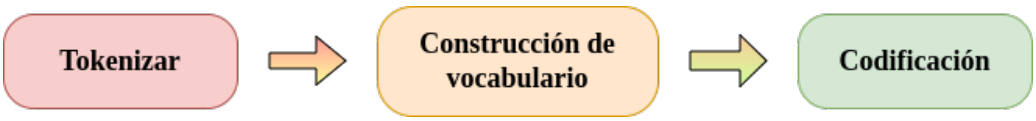


Figura 2.3: Pasos para la generación de una BoW.

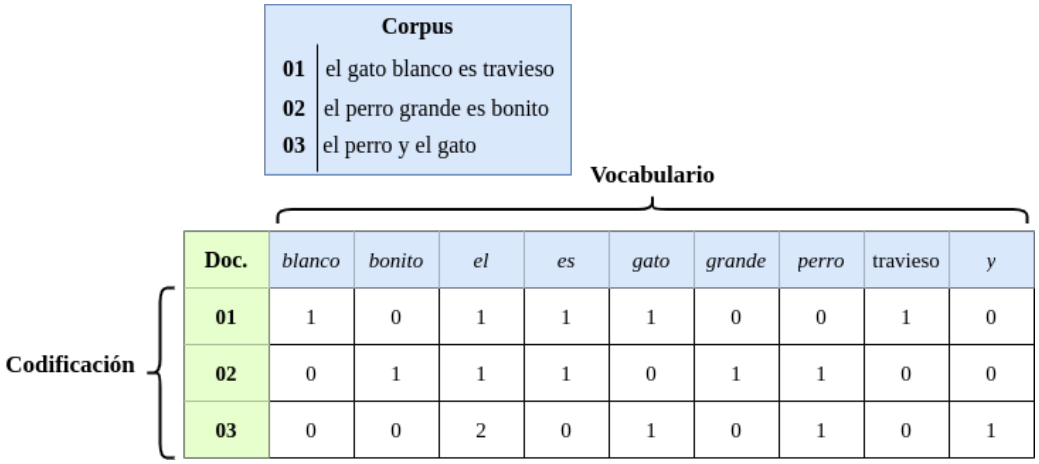
1. **Tokenizar.** Consiste en dividir cada documento o texto en las palabras que aparecen en él, también llamadas tokens.

2. **Construcción de vocabulario.** Significa que se debe generar un vocabulario de todas las palabras que aparecen en cualquiera de los documentos y enumerarlas, o bien, ordenarlas de manera alfabética.

3. **Codificación.** Para cada documento se cuenta con que frecuencia aparece cada una de las palabras del vocabulario.

El resultado obtenido es que, para cada palabra del vocabulario generado a partir del corpus completo, se crea una representación vectorial para cada documento, basada en la frecuencia con la que dicha palabra aparece en cada texto. Esto implica que la representación numérica asigna una característica única a cada palabra presente en el conjunto de datos. Es importante destacar que el orden de las palabras en los textos originales no tiene relevancia alguna en la representación de características bajo el enfoque de la BoW (Müller and Guido, 2016).

En la Figura 2.4 se muestra un ejemplo de la implementación del modelo de BoW, en el cual se define un vocabulario compuesto por todas las palabras únicas del corpus, y se codifican los documentos en vectores de acuerdo con la frecuencia de las palabras que los conforman y que están incluidas en dicho vocabulario.



Doc.	*blanco*	*bonito*	*el*	*es*	*gato*	*grande*	*perro*	travieso	*y*
01	1	0	1	1	1	0	0	1	0
02	0	1	1	1	0	1	1	0	0
03	0	0	2	0	1	0	1	0	1

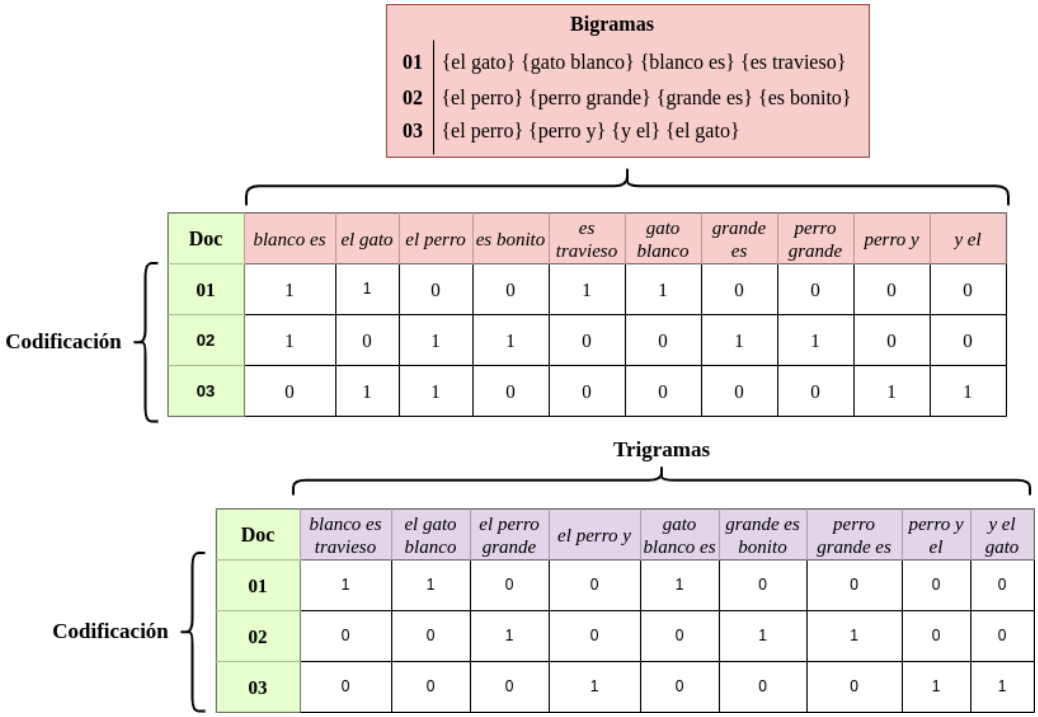
Figura 2.4: Ejemplo de representación de textos con BoW.

Una de las principales limitaciones de emplear una representación de bolsa de palabras radica en la pérdida del orden de las palabras. Sin embargo, existe una manera de capturar el contexto en el uso de esta representación, que implica no solo considerar el recuento de tokens individuales, sino también el recuento de conjuntos de tokens que aparecen en secuencia (Müller and Guido, 2016).

N-Gramas

Los N-Gramas son secuencias consecutivas de tokens o elementos de texto (como palabras o caracteres) extraídas de un corpus. Estas secuencias pueden tener una longitud variable, definida por n, lo que determina cuántos tokens consecutivos conforman cada grupo. Cuando n es igual a 2, las secuencias se denominan Bigramas; cuando n es igual a 3, se denominan Trigramas. De manera general, el término N-Gramas se refiere a cualquier conjunto de secuencias de tokens adyacentes, donde n representa el número de tokens que componen cada secuencia (Müller and Guido, 2016). Los N-Gramas permiten analizar patrones de co-ocurrencia y dependencias locales entre palabras o caracteres, capturando más información que los análisis basados en palabras individuales al considerar secuencias de palabras adyacentes en el texto (Müller and Guido, 2016).

En la Figura 2.5 se presenta un ejemplo de la codificación de documentos mediante N-Gramas, en particular Bigramas y Trigramas. Para obtener la representación vectorial de los documentos, se realiza una división de cada texto en secuencias consecutivas de palabras, a partir de las cuales se definen vocabularios de secuencias únicas de Bigramas y Trigramas. Posteriormente, se lleva a cabo el recuento de dichos N-Gramas, previamente definidos en los vocabularios, con el fin de generar la representación numérica de cada documento.



Doc	blanco es	el gato	el perro	es bonito	es travieso	gato blanco	grande es	perro grande	perro y	y el
01	1	1	0	0	1	1	0	0	0	0
02	1	0	1	1	0	0	1	1	0	0
03	0	1	1	0	0	0	0	0	1	1

Doc	blanco es travieso	el gato blanco	el perro grande	el perro y	gato blanco es	grande es bonito	perro grande es	perro y el	y el gato
01	1	1	0	0	1	0	0	0	0
02	0	0	1	0	0	1	1	0	0
03	0	0	0	1	0	0	0	1	1

Figura 2.5: Ejemplo de representación de textos con N-Gramas.

TF-IDF

El método TF-IDF (por sus siglas en inglés de *Term Frequency - Inverse Document Frequency*) es una técnica que permite evaluar la importancia relativa de una palabra o término en un documento dentro de un conjunto de documentos o corpus más amplio; es decir, dar cierto grado de importancia a cualquier palabra que aparezca con frecuencia en algún documento en comparación con el resto de documentos del corpus. Si una palabra aparece con frecuencia en un documento pero no está presente en el resto de documentos, quiere decir que se trata de una palabra muy descriptiva del contenido del documento (Müller and Guido, 2016). Este método se basa en dos componentes:

1. **Frecuencia del Termino (TF, Term Frequency)**

Este valor mide la frecuencia con la que aparece un término específico en un documento en relación con el número total de palabras en ese documento. Cuanto más frecuente sea un término en un documento, mayor será la puntuación de TF para ese término en ese documento (Müller and Guido, 2016).

2. **Frecuencia inversa del documento (IDF, Inverse Document Frequency)**

Este valor mide la importancia del término en el conjunto de documentos. Si un término es muy común en todos los documentos, su IDF será bajo, lo que indica que no es un término discriminativo. Por otro lado, si un término es raro en el conjunto de documentos, su IDF será

alto, lo que indica que es un término más relevante (Müller and Guido, 2016).

La formula para calcular TF-IDF es la siguiente:

$$tf_{i,j} \times idf_i$$

donde $tf_{i,j}$ es la Frecuencia del Termino i en el documento j y se calcula como:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_j}$$

donde:

- n_i, j es la cantidad total de veces que aparece el termino i en el documento j.
- $\sum_k n_j$ es la cantidad total de términos en el documento j.

y idf_i es la Frecuencia Inversa del Documento del termino i y se calcula como:

$$idf_i = log\left(\frac{N+1}{df_i+1}\right)+1$$

donde:

- N es el número total de documentos.
- df_i es el número total de documento en los que aparece el termino i.

Corpus	
01	el gato blanco es travieso
02	el perro grande es bonito
03	el perro es amigo del gato

$$w_{i,j} = tf_{i,j} \times idf_i$$

$$tf_i = \frac{n_{i,j}}{\sum_k n_j} \quad idf_i = log\left(\frac{N+1}{df_i+1}\right)+1$$

$$w_{gato,01} = \frac{1}{5} \times log\left(\frac{3+1}{2+1}\right)+1 = 0.258$$

Doc.	*amigo*	*blanco*	*bonito*	*del*	*el*	*es*	*gato*	*grande*	*perro*	travieso
01	0.000	**0.339**	0.000	0.000	0.2	0.2	**0.258**	0.000	0.000	**0.339**
02	0.000	0.000	0.339	0.000	0.2	0.2	0.000	0.339	0.258	0.000
03	0.282	0.000	0.000	0.282	0.166	0.166	0.215	0.000	0.215	0.000

Figura 2.6: Ejemplo de representación de textos con TF-IDF.

En la Figura 2.6 se ilustra el proceso de representación vectorial de documentos utilizando la técnica TF-IDF. En particular, se detalla el cálculo del valor correspondiente a la palabra "gato" en el documento 01, donde inicialmente se obtiene la Frecuencia del Término (tf) a partir

del cociente entre la frecuencia de la palabra y el total de palabras en el documento. Posteriormente, se calcula la Frecuencia Inversa de los Documentos (idf) utilizando el número total de documentos y el número de documentos en los que aparece la palabra. Este procedimiento se aplica a cada palabra contenida en el vocabulario, generando así un vector que representa cada documento del corpus.

Linguistic Inquiry and Word Count

LIWC (por sus siglas en inglés de *Linguistic Inquiry and Word Count*) es un diccionario de análisis de texto diseñado para evaluar diversos aspectos psicológicos, emocionales y lingüísticos en un corpus de texto (Pennebaker and Tausczik, 2010). LIWC clasifica las palabras en categorías que reflejan emociones, procesos cognitivos y aspectos sociales, mediante el análisis de la frecuencia de términos en cada una de estas categorías, lo que permite obtener una medida cuantitativa de los estados emocionales y psicológicos expresados en el texto (LIWC, 2023). En la Figura 2.7 se muestra un ejemplo con el resultado obtenido por LIWC a partir del análisis del siguiente texto:

> *En la imagen podemos ver que es una familia con cierto desorden a su alrededor, cosas que no van en su lugar, el papá o al menos el hombre de la imagen se nota estresado, la mamá viendo que pasa y haciendo lo suyo en la cocina, la niña se nota alegre y ayudando a ordenar el desastre que hay en la cocina. Hay botellas en el bote de basura y una rota, probablemente existen problemas de consumo de alguna sustancia por parte de algún integrante de la familia o se cayeron sin querer. La sala es un lugar ordenado, no se nota desastre o que algo este fuera del lugar pero esta solo. Fácilmente es un lugar al que uno de ellos puede ir y no estar dentro de el desastre de la cocina El perrito lleva un utensilio en la boca y nadie se da cuenta de eso, de ser lo contrario se lo estarían quitando porque es algo que no debe de hacer. De la misma manera en la cocina por más desorden que haya en la cocina, no hay nada que no pertenezca al lugar, hay frutas, verduras, trastes, utensilios, basura, muebles, un comedor, cosas de verdad pertenecen a ese lugar. También podría describir la imagen como que los padres o los adultos buscan que la niña no se de cuenta de algo, puede ser que previamente haya existido una pelea entre ambos y fingen que todo esta bien para no preocupar a la hija, y es el motivo de los gestos que el señor muestra, preocupado, estresado, nervioso, ansioso.*

Debido a las características de LIWC para identificar y medir la frecuencia de palabras, se ha utilizado para la detección de ansiedad y otras condiciones psicológicas, ya que permite obtener la frecuencia de palabras relacionadas con el estrés, la preocupación y otros indicadores emocionales (Yu et al., 2023).

las pautas establecidas por los datos de entrenamiento. Una vez que se ha construido el modelo, se puede evaluar su precisión utilizando datos de prueba. Si el modelo supera satisfactoriamente las etapas de entrenamiento y prueba, está preparado para aplicarse en situaciones del mundo real (Theobald, 2017).

Los sistemas de ML pueden clasificarse en función de la disponibilidad de datos etiquetados y del tipo de supervisión que reciben durante su proceso de entrenamiento (Géron, 2023). En el presente trabajo se exploran tres categorías de ML.

- Aprendizaje Supervisado.
- Aprendizaje no Supervisado.
- Aprendizaje Semisupervisado.

2.3.1. Aprendizaje Supervisado

El *aprendizaje supervisado* consiste en entrenar un modelo para identificar patrones mediante el establecimiento de relaciones entre variables de entrada y los resultados esperados, utilizando datos previamente etiquetados. En este caso, los datos etiquetados se refieren a aquellos en los que cada conjunto de características está asociado a un valor de salida conocido (Theobald, 2017). A lo largo del proceso de entrenamiento, el modelo recibe un conjunto de características (variables de entrada) junto con sus respectivas salidas correctas, lo que le permite aprender a realizar un mapeo preciso entre entradas y salidas. Una vez finalizado el entrenamiento, el modelo adquiere la capacidad de realizar predicciones con precisión sobre datos nuevos y no observados (Géron, 2023).

El desafío en los algoritmos de aprendizaje supervisado radica en disponer de una cantidad suficiente de datos que sean representativos de todas las posibles variaciones, incluyendo valores atípicos y anomalías. Los datos utilizados deben ser relevantes y, en caso de ser extraídos de un conjunto de datos más amplio, deben ser seleccionados de manera aleatoria (Theobald, 2017). Los algoritmos de aprendizaje supervisado se clasifican en algoritmos de clasificación y algoritmos de regresión, algunos de los cuales se abordarán más adelante.

Clasificación

La *clasificación* se refiere al proceso de asignar automáticamente una etiqueta a un ejemplo no etiquetado, seleccionada de un conjunto predefinido de posibilidades (Burkov, 2019). Esta tarea se puede dividir en dos categorías principales: la clasificación binaria, que consiste en distinguir entre dos clases, y la clasificación multiclase, que abarca la asignación de ejemplos a más de dos clases (Müller and Guido, 2016).

Regresión

La regresión es un problema que consiste en predecir un valor real a partir de un ejemplo sin una etiqueta previamente definida. Este problema se centra en modelar las relaciones entre variables y realizar predicciones de valores continuos (Burkov, 2019).

2.3.2. Aprendizaje no Supervisado

El *aprendizaje no supervisado* se distingue por la ausencia de datos etiquetados, lo que implica que no existe retroalimentación explícita incorporada en el proceso. En su lugar, el proceso se fundamenta en la agrupación de datos y en el ajuste del algoritmo basado en los patrones o estructuras que se descubren durante el análisis (Theobald, 2017). El objetivo de un algoritmo de aprendizaje no supervisado es generar un modelo que tome un conjunto de características o variables como entrada y lo convierta en otro conjunto o en un valor que sea útil para abordar una tarea práctica (Burkov, 2019). Una de las ventajas de emplear algoritmos no supervisados es que posibilitan la detección de patrones en los datos que, de otra manera, podrían pasar desapercibidos (Theobald, 2017). Los algoritmos no supervisados incluyen algoritmos de agrupamiento y algoritmos de reducción de bidimensionalidad.

Un ejemplo clásico de aprendizaje no supervisado lo constituyen los algoritmos de agrupamiento, los cuales se encargan de asociar datos que comparten características similares, representados habitualmente mediante puntos (Theobald, 2017). Una tarea estrechamente vinculada a este tipo de aprendizaje es la reducción de dimensionalidad, cuyo objetivo es simplificar los datos sin perder información significativa. Esto se puede lograr, por ejemplo, mediante la combinación de múltiples características correlacionadas en una sola (Géron, 2023). Además, un enfoque de aprendizaje no supervisado puede ser útil en la identificación de reglas de asociación, cuyo propósito es analizar grandes volúmenes de datos y descubrir relaciones relevantes entre los distintos atributos (Géron, 2023).

2.3.3. Aprendizaje Semisupervisado

Por otro lado, el *aprendizaje semisupervisado* se presenta como una solución eficiente cuando el proceso de etiquetado de datos resulta largo y costoso, lo que da lugar a una situación común, en la que se cuenta con una gran cantidad de instancias sin etiquetar y un número limitado de instancias etiquetadas. Algunos algoritmos son capaces de manejar este tipo de datos parcialmente etiquetados, lo cual se conoce como aprendizaje semisupervisado (Géron, 2023). El objetivo de un algoritmo de aprendizaje semisupervisado es similar al de uno supervisado; sin embargo, al incorporar un gran volumen de ejemplos no etiquetados, se espera que el algoritmo logre mejorar su capacidad para encontrar o calcular un modelo más preciso (Burkov, 2019).

Es posible que parezca contraproducente aumentar el número de ejemplos sin etiquetar en el aprendizaje, ya que esto aparentemente introduce más incertidumbre en el problema. No obstante, al agregar ejemplos sin etiquetar, en realidad estamos incorporando más información sobre el problema: una muestra más extensa tiende a representar de manera más precisa la distribución de probabilidad de la cual provienen los datos que si están etiquetados. En teoría, un algoritmo de ML debería ser capaz de aprovechar esta información adicional e ir mejorando (Burkov, 2019).

La mayoría de los algoritmos de aprendizaje semisupervisados son en realidad una combinación de enfoques entre supervisado y no supervisado. Por ejemplo, podríamos emplear un algoritmo de agrupamiento para agrupar instancias similares y luego asignar a cada instancia no etiquetada la etiqueta más común dentro de su grupo. Una vez etiquetado el conjunto completo de datos, se vuelve posible aplicar cualquier algoritmo de aprendizaje supervisado (Géron, 2023).

2.4. Algoritmos de Clasificación

En esta sección se presentan y describen algunos de los algoritmos empleados en el proceso de clasificación, con el fin de proporcionar una visión general de sus principales características y enfoques, facilitando una mejor comprensión de su papel en ML.

2.4.1. K - vecinos más cercanos

El algoritmo K Vecinos más Cercanos (K-NN del ingles *K Nearest Neighbors*) es un algoritmo de aprendizaje supervisado basado en instancias o similitudes que se utiliza en la agrupación, regresión y clasificación de datos (Isabelle, 2006). K-NN es generalmente preciso y fácil de comprender, se trata de un algoritmo de agrupación dentro del ML que se implementa para clasificar puntos de datos en determinada clase de acuerdo con su posición con respecto a los puntos de datos más cercanos conocidos (Theobald, 2020).

Desarrollar un modelo de clasificación utilizando el algoritmo K-NN consiste en realizar un adecuado preprocesamiento de datos y definir el valor de k, es decir, el número de vecinos más cercanos que se pretende considerar al momento de hacer las predicciones (Manning et al., 2008). K-NN memoriza todos los ejemplos de entrenamiento y luego compara un dato completamente desconocido con respecto a ellos, para que finalmente le asigne la etiqueta correspondiente de la clase con mayor frecuencia dentro de sus k vecinos más cercanos y en caso de empate realizar la elección al azar (Isabelle, 2006). El número elegido de vecinos identificados, definido por k, es crucial para determinar los resultados, por lo general se hace en función de la experiencia o del conocimiento del problema. Para seleccionar el valor de k, se recomienda

que sea impar para que los empates sean menos probables, por ejemplo, es común darle valores de 3 y 5 (Manning et al., 2008).

Existen variaciones del algoritmo K-NN, principalmente en el uso de distintas medidas de similitud (Isabelle, 2006), por ejemplo, la cercanía de dos puntos de datos se puede determinar en función de la distancia; durante la práctica es muy común utilizar la distancia euclidiana, sin embargo, se pueden emplear otras como: la distancia de Chebychev, la distancia de Mahalanobis, la distancia de Hamming o la similitud del coseno (Burkov, 2019).

En la Figura 2.8 podemos observar un conjunto de puntos de datos que han sido categorizados en dos grupos o clases diferentes; además, se puede calcular la distancia entre dos puntos cuales quiera dentro del conjunto. Al introducir un nuevo punto de datos para predecir la clase a la que pertenece, el algotitmo K-NN lo hace en función de la relación que tiene con los puntos de datos cercanos que ya existen (Theobald, 2017). Primero, se establece el valor de k para determinar cuantos puntos de datos se pretenden considerar al llevar a cabo la clasificación. Por ejemplo, si k tiene un valor de 3, se analiza la relación del nuevo punto con los tres puntos de datos más cercanos y el resultado son: dos puntos de datos de la Clase B y un punto de datos de la Clase A. Por lo tanto, la predicción del algoritmo para determinar la categoría del nuevo punto de datos se basara en la mayoría de sus vecinos, con dos de los tres puntos de datos la Clase B será la seleccionada (Theobald, 2017). En la Figura 2.8 se puede observar que la clasificación cambiará dependiendo de si k se establece en 3 o 7, de modo que es mejor probar numerosas combinaciones de k para encontrar el mejor ajuste y evitar establecer k demasiado bajo o demasiado alto. Establecer k demasiado bajo aumentará el sesgo y conducirá a una clasificación errónea y establecer k demasiado alto lo hará computacionalmente costoso (Theobald, 2020).

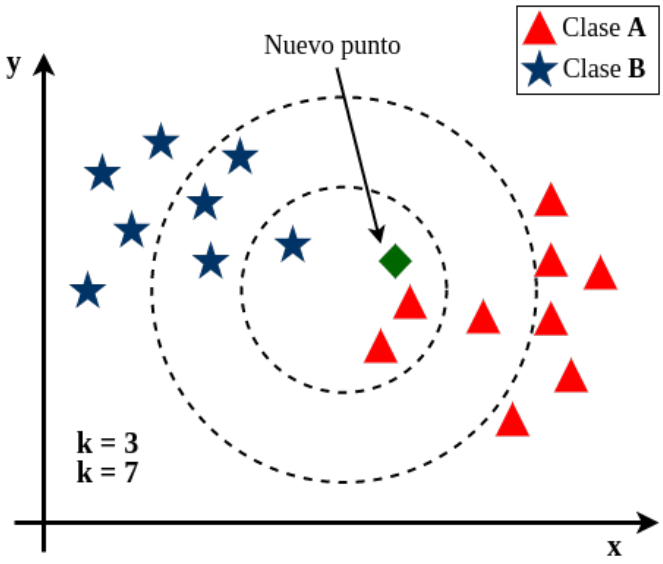


Figura 2.8: Algoritmo K-NN (Theobald, 2017).

Por lo general, dentro del ML se busca tener la mayor cantidad de datos posibles durante el entrenamiento. En cambio, para K-NN utilizar un conjunto de datos muy grande conlleva una grave penalización de eficiencia al momento de clasificar (Manning et al., 2008), esto significa que la cantidad de puntos de datos en el conjunto de datos es proporcional al tiempo que lleva ejecutar una sola predicción, lo que puede generar tiempos de procesamiento más lentos, debido a que se exige almacenar un conjunto de datos completo y calcular la distancia o medida de similitud entre los nuevos puntos de datos y todos los puntos de datos existentes. Como resultado, la ejecución que requiere de un cálculo intensivo de operaciones por cada nuevo dato. Por lo tanto, es posible que este algoritmo no se recomiende para trabajar con conjuntos de datos grandes (Theobald, 2017).

Otra desventaja al utilizar K-NN es la alta dimensión o numerosas características de los datos. Medir las múltiples distancias entre puntos de datos en un espacio de tres o cuatro dimensiones puede resultar agotador para los recursos informáticos y complicado de clasificar con precisión (Theobald, 2020).

2.4.2. Perceptrón Multicapa

Dentro del ML existe una técnica conocida como Red Neuronal Artificial (RNN, por sus siglas en ingles de *Artificial Neural Network*), este nombre fue inspirado por el parecido con el cerebro humano, ya que permiten el procesamiento de datos a través de capaz de análisis. Las RNN están formadas por neuronas interconectadas que interactúan entre sí, tal como sucede con la estructura neuronal del cerebro (Theobald, 2017).

Las neuronas de una RNN están organizadas por capas que se encuentran apiladas una sobre otra de tal forma que se pueden dividir en tres secciones: capa de entrada, capa oculta y capa de salida. Durante el proceso de análisis la capa de entrada consta de todos los datos sin procesar como texto, que se divide entre las neuronas para identificar características generales, luego cada neurona enviá información a las neuronas de la capa oculta en donde se analizan y procesan las características de entrada. A medida que la información fluye entre las capas, se vuelve menos abstracta y más especifica hasta llegar al resultado final que se muestra en la capa de salida (Theobald, 2017). En la Figura 2.9 se muestra la representación de una RNN formada por una capa de entrada que se compone de tres neuronas, dos capas ocultas con cuatro neuronas cada una y la capa de salida que esta formada por dos neuronas.

Un Perceptrón Multicapa (MLP, por sus siglas en inglés de *Multi-Layer Perceptron*) puede ser conceptualizado como una generalización de un modelo lineal que lleva a cabo procesamientos en multiples capas para llegar a una decisión, tanto categórica (clasificación) como continua (regresión) (Theobald, 2020) (Müller and Guido, 2016). Un MLP es una RNN compuesta por

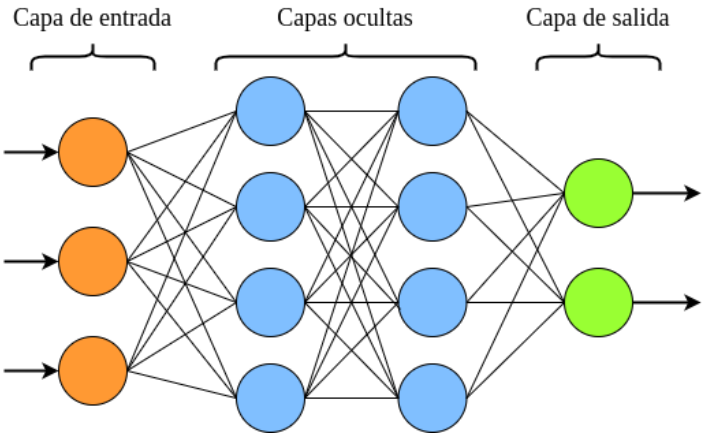


Figura 2.9: Representación de una RNA.

capas de perceptrones interconectados que procesan información que fluye en una sola dirección (de la capa de entrada a la capa de salida) por tener una arquitectura *feedforward* (Müller and Guido, 2016) (Theobald, 2020). Sin embargo, el uso de perceptrones con una función de activación *escalonada* o *paso* dentro de una red neuronal presenta una desventaja significativa, ya que su salida es binaria. Esto implica que pequeños cambios en los pesos o sesgos de cualquier perceptrón pueden inducir resultados polarizantes (Theobald, 2020). Para mitigar este efecto, una alternativa es emplear la función de activación *logística* o *sigmoide*, la cual genera valores entre 0 y 1, proporcionando mayor flexibilidad para absorber variaciones menores en los pesos sin producir cambios abruptos en los resultados (Theobald, 2020). En la Tabla 2.1 se presentan las fórmulas de diversas funciones de activación, incluyendo la *escalonada*, la *logística*, la tangente hiperbólica (*tanh*) y la unitaria lineal rectificada (*ReLU*).

Función	**Formula**
escalonada o *paso*	$\begin{cases} 0 & \text{si } z \leq umbral \\ 1 & \text{si } z > umbral \end{cases}$
logística o *sigmoide*	$\frac{1}{1-e^{-z}}$
tanh	$\frac{2}{1+e^{-2z}} - 1$
ReLU	$max(0,z)$

Tabla 2.1: Funciones de activación en MLP.

Durante el proceso de entrenamiento de un MLP se utiliza el algoritmo de retropropagación (del inglés *backpropagation*) y su funcionamiento se basa en tener una función de activación diferente de la función *escalonada* dentro de la estructura de las neuronas de la red, por ejemplo, la función *logística*, debido a que el algoritmo ocupa el gradiente descendiente y la función *paso* contiene unicamente segmentos planos, por lo que no hay ningún gradiente con el que cual trabajar, mientras que la función *logística* tiene una derivada distinta de cero y definida en todas partes, lo que permite que el descenso del gradiente haga algunos progresos en cada paso (Géron, 2023).

El algoritmo *backpropagation* funciona bien con otras funciones de activación como la función *tanh* que al igual que la función *logística* es continua y diferenciable pero su valor de salida esta definido entre -1 y 1. De igual forma se tiene la función *ReLU* la cual es continua pero no es diferenciable en cero, con esta función la pendiente cambia abruptamente lo que puede provocar que el descenso del gradiente rebote, además su derivada es cero cuando se evaluá con cualquier número menor que cero, sin embargo, en la práctica funciona muy bien y tiene la ventaja de ser rápida de calcular (Géron, 2023).

En términos generales, los MLP resultan idóneos para analizar conjuntos de datos extensos y complejos, sin estar limitados por restricciones temporales o computacionales. Al compararlos con algoritmos que demandan menos recursos computacionales, como los árboles de decisión y la regresión logística, se observa que estos últimos son más eficientes al trabajar con conjuntos de datos más pequeños. Sin embargo, debido a la abundancia de hiperparámetros, los MLP requieren más tiempo y esfuerzo para su ajuste en comparación con otros algoritmos. En cuanto al tiempo de procesamiento, los MLP tienden a ejecutarse más lentamente que la mayoría de las técnicas de aprendizaje supervisado. (Theobald, 2020).

2.4.3. Máquinas de Soporte Vectorial

Una Máquina de Vectores de Soporte (SVM, por las siglas en inglés de *Support Vector Machine*) es un algoritmo de ML que destaca por su eficiencia y capacidad para adaptarse rápidamente a diversos problemas. Aunque puede emplearse para tareas de regresión, sobresale por su versatilidad en tareas de clasificación, tanto lineal como no lineal, especialmente en conjuntos de datos de tamaño moderado (Géron, 2023).

Como técnica de clasificación, SVM se utiliza para filtrar datos de una variable objetivo binaria o multiclase, poniendo gran importancia en la ubicación de la linea limite de clasificación (Theobald, 2020). Durante el entrenamiento, las SVM evalúan la importancia de cada punto de datos para definir el límite de decisión entre las clases (Müller and Guido, 2016). Este limite no sólo separa las clases, sino que también se mantiene lo más alejado posible de las instancias

de entrenamiento más cercanas, generando un espacio conocido como margen máximo de separación, lo que mejora la clasificación de las clases (Kelleher, 2020). Generalmente, solo un subconjunto de los puntos de entrenamiento es relevante para establecer el límite de decisión: aquellos situados en el borde entre las clases. Estos puntos se denominan vectores de soporte y dan nombre al algoritmo (Müller and Guido, 2016). Además habrá al menos un vector de soporte para cada clase, sin embargo, no existe un límite máximo en la cantidad total de vectores de soporte (Kelleher, 2020). Todo lo anterior, se ilustra en la Figura 2.10.

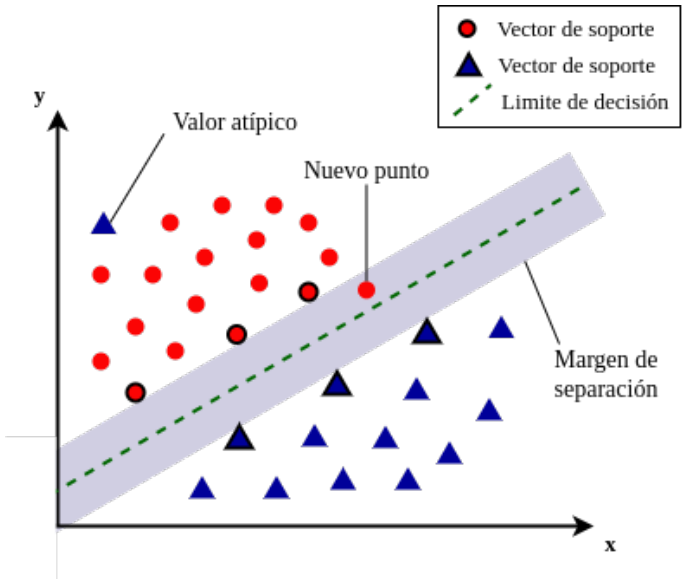
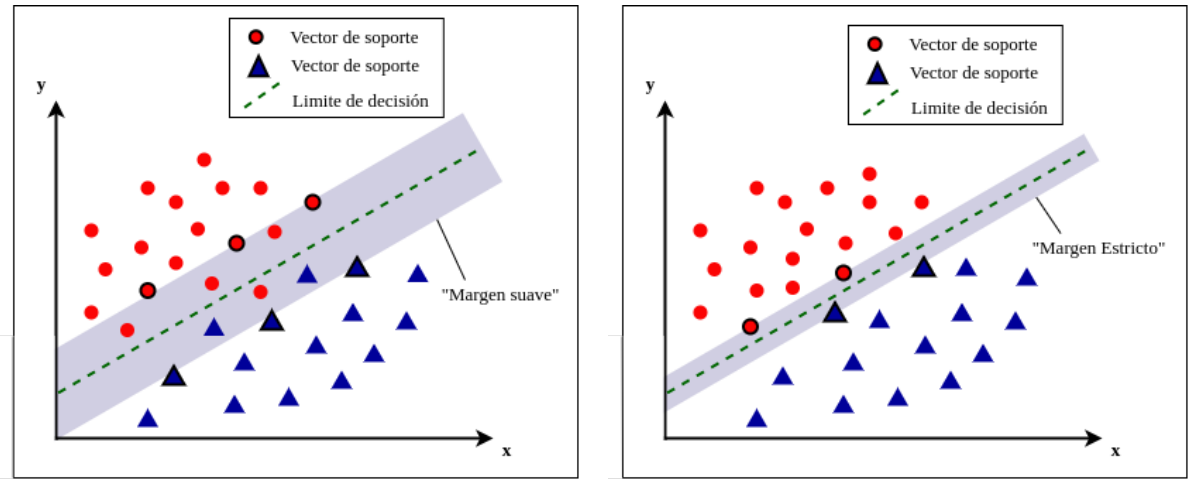


Figura 2.10: Clasificación con SVM.

Para predecir la clase a la que pertenece un nuevo punto, se mide la distancia a cada uno de los vectores de soporte. La decisión de clasificación se basa en estas distancias, lo que resalta la importancia de los vectores aprendidos durante el entrenamiento (Müller and Guido, 2016). Si bien, las SVM suelen ser muy efectivas, son sensibles a la configuración de parámetros y al escalado de los datos (Theobald, 2020). En particular, es necesario que todas las características tengan una escala similar, por tal motivo es importante hacer un ajuste previo de la escala de las característica para que sus valores sean comparables (Müller and Guido, 2016).

El limite de decisión en las SVM puede ser modificado para ignorar casos mal clasificados en los datos de entrenamiento utilizando algún hiperparámetro de penalización que permite cierto nivel de soporte (se establece el margen de separación) a nuevos puntos de datos que pueden infringir el limite de decisión (Theobald, 2020). Un valor bajo de dicho hiperparámetro ensancha el margen (margen suave) y aumenta la flexibilidad del modelo, permitiendo que algunos puntos crucen el margen. Esto mejora la generalización del modelo haciéndolo menos propenso al sobreajuste (Géron, 2023), lo cual se puede observar en la Figura 2.11a. En contraste, un valor alto del hiperparámetro penaliza fuertemente las clasificaciones incorrectas, lo que hace que el margen sea más estrecho (margen estricto), como se ve en la Figura 2.11b. Esto puede llevar al

modelo a ajustarse demasiado a los datos de entrenamiento, resultando en un mal rendimiento al clasificar nuevos datos debido al sobreajuste (Theobald, 2020). Por lo tanto, en un modelo de SVM se busca lograr un equilibrio entre un margen amplio (más errores) y un margen estrecho (menos errores) haciendo que el modelo sea lo suficientemente estricto y flexible para regular hasta que punto los casos mal clasificados (en el lado equivocado del margen) son ignorados (Theobald, 2020).



(a) Margen con un valor de *C* bajo (b) Margen con un valor de *C* alto

Figura 2.11: Márgenes en SVM.

A pesar de que los clasificadores SVM lineales son muy eficientes y a menudo ofrecen un rendimiento bastante bueno, muchos conjuntos de datos están lejos de ser linealmente separables (Géron, 2023). En muchos casos, puede ser muy difícil clasificarlos correctamente debido a la presencia de ruido, errores en el etiquetado o valores atípicos que distorsionan los resultados (Burkov, 2019). Afortunadamente existen variantes avanzadas que permiten clasificar datos no lineales mediante el llamado truco del núcleo (*Kernel Trick* en inglés) (Géron, 2023). Esta técnica consiste en mapear los datos de un espacio de baja dimensión a un espacio de alta dimensión cuando no se pueden clasificar utilizando un límite de decisión lineal en su espacio original. Por ejemplo, al pasar de un espacio bidimensional a uno tridimensional, podemos utilizar un plano lineal para dividir los datos dentro de un área tridimensional. En otras palabras, *Kernel Trick* permite clasificar puntos de datos con características no lineales utilizando una clasificación lineal en una dimensión superior (Theobald, 2020).

Al implementar *Kernel Trick* en la SVM se utilizan funciones para no calcular las nuevas dimensiones de los datos, las cuales podrían ser representaciones muy grandes y afectar el rendimiento del entrenamiento del modelo. Por lo tanto, un *kernel* es una función que toma dos puntos en el espacio de entrada y devuelve el producto punto de esos puntos en un espacio de características de mayor dimensión, sin tener que calcular explícitamente las coordenadas en ese espacio más alto (Müller and Guido, 2016).

Existen diferentes funciones utilizadas como kernel dentro de una SVM, entre las más comunes se encuentran el *kernel polinomial*, el *kernel radial* (RBF, por sus siglas en inglés de *Radial basis función*) o también conocido como *kernel Gaussiano* y el *kernel sigmoid*. Sin embargo, entrar a profundidad en cada uno de ellos esta fuera del alcance de esta tesis.

Las SVM son capaces de establecer límites de decisión complejos, funcionan muy bien con datos tanto de baja como de alta dimensión, es decir, con pocas o muchas características, pero no se adaptan bien a un gran número de muestras, debido a que pueden presentar problemas en tiempo de ejecución y uso intensivo de memoria (Müller and Guido, 2016). Una de las principales ventajas de los modelos SVM es su resistencia al sobreajuste (Kelleher, 2020). Además, es importante destacar que las SVM sobresalen en la identificación de valores atípicos en conjuntos de datos complejos, es menos sensible a dichos puntos de datos y minimiza su impacto en la ubicación del límite de decisión (Theobald, 2020). No obstante, presentan varias desventajas importantes, necesitan un preprocesamiento cuidadoso de los datos y un ajuste exacto de los parámetros. Finalmente, se debe considerar que los modelos SVM son difíciles de interpretar, lo que puede dificultar el hecho de comprender por qué se realizó una predicción específica (Müller and Guido, 2016).

2.4.4. Árboles de Decisión

Los Árboles de Decisión (DT, por sus siglas en inglés de *Decision Tree*) son algoritmos de ML altamente versátiles, comúnmente empleados en tareas de clasificación y regresión (Géron, 2023). El entrenamiento de un DT implica aprender una secuencia de preguntas (del tipo "sí/no") que permiten llegar a la respuesta correcta de la manera más eficiente posible. Estas preguntas se denominan pruebas (no confundirse con el conjunto de datos de prueba) y se aplican a los datos, los cuales por lo general se presentan en forma de características continuas sobre un valor o clase objetivo (Müller and Guido, 2016). Por lo tanto, las pruebas suelen adoptar la forma: "¿Es la característica i mayor que el valor a?". De esta manera al construir un DT, el algoritmo evalúa todas las pruebas posibles y selecciona aquella que garantice una partición óptima de los datos, recuperando la mayor cantidad de información sobre la variable objetivo (Theobald, 2020).

El proceso de construcción del DT se ilustra en la Figura 2.12, en donde se comienza con un nodo superior, también llamado raíz, que representa el conjunto de datos completo y actúa como punto de partida. Desde la raíz, el algoritmo divide los datos en subconjuntos basados en las pruebas seleccionadas (Müller and Guido, 2016). En cada nodo de ramificación del gráfico, se examina una característica específica del vector de características: si el valor de la característica está por debajo de un umbral específico, se continua por la rama izquierda; en caso contrario,

se sigue por la rama derecha. Estas divisiones, también conocidas como aristas se unen nuevas hojas, o nodos, formando puntos de decisión (Burkov, 2019). Este proceso se repite utilizando los datos recopilados en cada nueva hoja hasta que se alcanza una hoja que ya no genera nuevas ramas, resultando en lo que se llama un nodo terminal (Theobald, 2020).

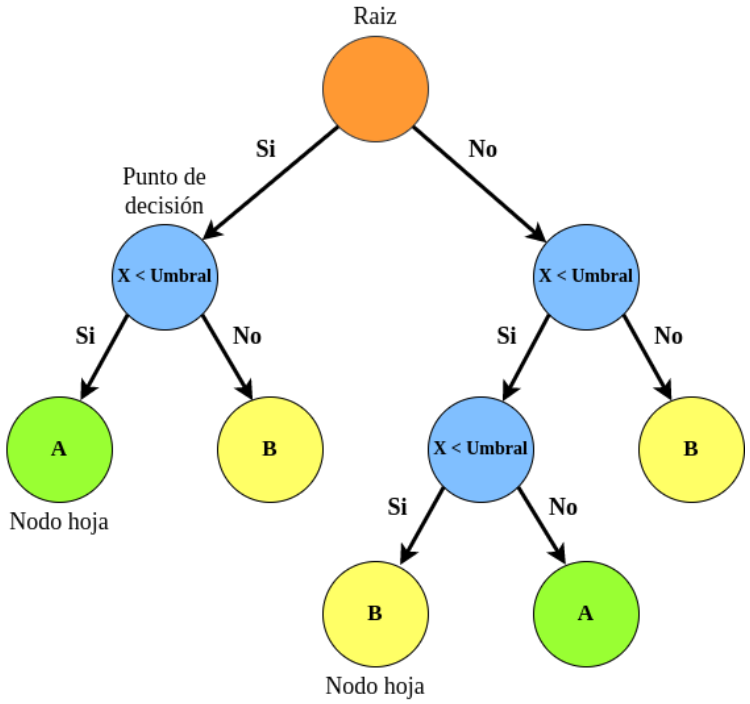


Figura 2.12: Representación de DT.

Este procedimiento recursivo de división de los datos continúa hasta que cada región de la partición (cada hoja del árbol) contiene únicamente una clase objetivo, lo que se conoce como una hoja pura. Una hoja pura es aquella en la que todos los puntos de datos comparten el mismo valor objetivo (Müller and Guido, 2016). Cuando se alcanza el nodo hoja, se toma la decisión sobre la clase a la que pertenece el ejemplo. Para hacer una predicción con un DT, se toma un nuevo punto de datos y se verifica en qué región de la partición del espacio de características se encuentra dicho punto. Esto se logra atravesando el árbol desde la raíz y siguiendo las ramas izquierda o derecha según se cumplan o no las pruebas en cada nodo (Müller and Guido, 2016). Al alcanzar una hoja, se predice el valor objetivo mayoritario en esa región (o el valor único en el caso de hojas puras). El objetivo al construir un DT es mantenerlo lo más pequeño posible. Esto se consigue eligiendo variables que dividan los datos en grupos homogéneos, disminuyendo la impureza o heterogeneidad de los nodos en las ramas subsecuentes (Theobald, 2020).

Una de sus ventajas más destacadas es su transparencia y facilidad de interpretación. Los modelos desarrollados a partir de DT, poseen la capacidad de operar eficazmente con conjuntos de datos reducidos, al mismo tiempo que requieren una menor cantidad de recursos computacionales (Theobald, 2020). Además, funcionan bien con características en diferentes escalas

lo que elimina la necesidad de normalización o estandarización previa y su capacidad para ser representados gráficamente facilita su explicación a personas no expertas, haciendo que sean una herramienta valiosa en diversas aplicaciones (Müller and Guido, 2016).

Los DT son susceptibles de sobreajustar el modelo a los datos de entrenamiento, ya que analizan y decodifican con precisión los patrones encontrados, lo que puede provocar que no clasifiquen correctamente los datos de prueba. Por ello, controlar la complejidad de los árboles de decisión es fundamental para evitar el sobreajuste y mejorar la capacidad de generalización del modelo (Theobald, 2020). Este problema puede originarse al construir DT sin ninguna restricción, volviéndose excesivamente profundos hasta alcanzar únicamente hojas puras que logran una precisión del 100 % en el conjunto de entrenamiento, debido a que cada punto de datos se clasifica correctamente según la clase mayoritaria de su hoja. Sin embargo, esta precisión no garantiza un buen desempeño en datos nuevos(Müller and Guido, 2016).

Para evitar el sobreajuste, se utilizan dos estrategias principales en la construcción de DT: la poda previa y la poda posterior. La poda previa detiene el crecimiento del árbol antes de que todas las hojas sean puras, aplicando criterios como limitar la profundidad máxima, restringir el número máximo de hojas o establecer un mínimo de puntos en un nodo antes de dividirlo. Por otro lado, la poda posterior implica construir el árbol completo y luego eliminar los nodos que aportan poca información relevante (Géron, 2023).

2.4.5. Bosque Aleatorio

En el ámbito del ML, uno de los principales objetivos es mejorar el rendimiento de los modelos. Un ejemplo destacado de cómo se puede lograr esto es el Bosque Aleatorio (RF, por sus siglas en inglés de *Random Forest*), que pertenece a los métodos de ensamble. Los ensambles combinan múltiples modelos de ML para generar soluciones más precisas. (Müller and Guido, 2016). Un RF se define esencialmente como una colección de DT, en la que cada árbol presenta ligeras variaciones respecto a los demás (Theobald, 2020), esto se puede ilustrar en la Figura 2.13.

La construcción de numerosos árboles, cada uno adaptado de manera distinta a los datos, ayuda a reducir el riesgo de sobreajuste al promediar sus predicciones. Este método permite que el modelo global, al combinar las salidas de todos los árboles, ofrezca una solución más estable y menos influenciada por las variaciones del conjunto de datos de entrenamiento (Müller and Guido, 2016).

Los RF reciben su nombre al introducir aleatoriedad en la construcción de los árboles con el propósito de asegurar que cada árbol sea distinto. Este proceso se realiza a través de dos

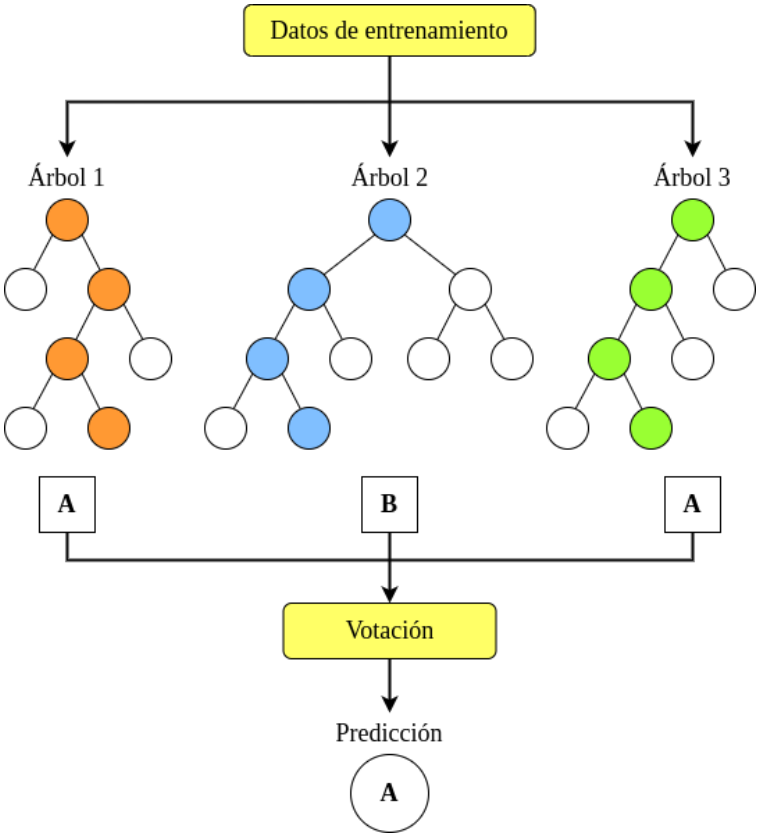


Figura 2.13: Representación de RF.

métodos principales: en primer lugar, mediante la selección aleatoria de los puntos de datos utilizados para construir cada árbol; y en segundo lugar, mediante la selección aleatoria de las características en cada punto de división durante el entrenamiento (Müller and Guido, 2016).

Para realizar una predicción, el algoritmo inicia el proceso generando una estimación para cada árbol dentro del RF. En el contexto de la clasificación, se aplica una estrategia conocida como *"voto ponderado"*. En este enfoque, cada árbol emite una predicción *"suave"*, lo que significa que proporciona una probabilidad para cada una de las posibles etiquetas de salida, en lugar de una clasificación definitiva. Una vez que todos los árboles han generado sus probabilidades, estas se promedian para obtener una estimación general. El resultado final es la clase cuya probabilidad promedio es la más alta (Müller and Guido, 2016). Este método permite que el modelo aproveche la información combinada de todos los árboles, lo que generalmente conduce a una mayor precisión en la predicción final (Theobald, 2020).

Los RF, empleados tanto en regresión como en clasificación, son métodos ampliamente utilizados en ML debido a su alta eficacia y su capacidad para funcionar bien sin ajuste intensivo de parámetros ni necesidad de escalamiento de datos. Estos modelos manejan eficazmente grandes volúmenes de datos y permiten el entrenamiento en paralelo en múltiples núcleos de CPU. Sin embargo, requieren más memoria y tienen tiempos de entrenamiento y predicción más largos

en comparación con otros modelos (Müller and Guido, 2016).

2.4.6. Naive Bayes Multinomial.

El algoritmo Naive Bayes Multinomial (NB) es una técnica de predicción y clasificación particularmente adecuada para datos discretos. Es ampliamente utilizado en NLP, especialmente en tareas de clasificación de texto (IBM, sf). NB es una variante específica del algoritmo Naive Bayes, el cual se basado en la aplicación del teorema de Bayes. Este teorema permite calcular la probabilidad posterior de una clase dado un conjunto de características. Sin embargo, la principal característica del algoritmo Naive Bayes es la suposición de que cada característica es independiente de las demás (Sarkar, 2016). De manera matemática, esto se puede expresar: dada una variable de clase y y un conjunto de n características representadas como un vector de características $X = \{x_1, x_2, ...x_n\}$, utilizando el teorema de Bayes podemos definir la probabilidad de que ocurra y dado el conjunto de características (Sarkar, 2016)

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

Donde,

- $P(y|X)$ es la probabilidad de la clase y dada el conjunto de características X. Esta es la probabilidad posterior, que queremos calcular (Thanaki, 2017).
- $P(y)$ es la probabilidad previa de la clase y. Representa nuestro conocimiento inicial sobre la distribución de las clases antes de observar los datos (Thanaki, 2017).
- $P(X|y)$ es la estimación de la probabilidad del vector de características X dado que la clase es y. Esto representa la verosimilitud, es decir, cuán probable es observar las características X si supiéramos que la clase es y (Thanaki, 2017).
- $P(X)$ es la probabilidad previa del vector de características X. Esta es una constante de normalización que asegura que las probabilidades posteriores sumen uno (Thanaki, 2017).

Ahora, bajo el supuesto de independencia de que:

$$P(x_i|y, x_1, x_2, ..., x_{i-1}, x_{i+1}, ..., x_n) = P(x_i|y)$$

y para toda i, se puede representar como:

$$P(y|x_1, x_2, ..., x_n) = \frac{P(y) \times \prod_{i=1}^{n} P(x_i|y)}{P(y|x_1, x_2, ..., x_n)}$$

donde x varia de 1 a n. Como $P(x_1, x_2, ..., x_n)$ es constante, el modelo se puede expresar así:

$$P(y|x_1,x_2,...,x_n)\ \alpha\ P(y)\times\prod_{i=1}^{n}P(x_1|y)$$

Esto implica que, bajo los supuestos de independencia entre las características, donde cada una es condicionalmente independiente de las demás, la distribución condicional de la variable de clase y que se va a predecir se puede expresar mediante la siguiente ecuación matemática (Sarkar, 2016):

$$P(y|x_1,x_2,...,x_n)=\frac{1}{Z}P(y)\times\prod_{i=1}^{n}P(x_i|y)$$

Donde la medida de evidencia, $Z=p(x)$, es un factor de escala constante que depende de las variables características. A partir de esta ecuación, podemos construir el clasificador Niave Bayes combinándolo con la regla de decisión Máximo a Posteriori (MAP). Esta regla permite representar el clasificador como una función matemática que puede asignar una etiqueta de clase predicha $\hat{y}=C_k$ para algún k utilizando la siguiente representación (Sarkar, 2016):

$$\hat{y}=\underset{k\in\{1,2,...,K\}}{\arg\max}\ P(C_k)\times\prod_{i=1}^{n}P(x_i|C_k)$$

Los diferentes clasificadores basados en teorema de Bayes difieren principalmente por las suposiciones que hacen con respecto a la distribución de $P(x_1|C_k)$. NB se basa en la distribución multinomial, que modela el número de ocurrencias de cada una de las características en un conjunto de eventos (Sarkar, 2016). Esta distribución se puede ver como $p_y=\{p_{y1},p_{y2},...,p_{yn}\}$ para cada etiqueta de clase y, mientras que n es el número total de características. De la ecuación anterior, $p_{yi}=P(x_i|y)$, es la probabilidad de la característica i en cualquier muestra de observación que tenga un resultado o clase. EL parámetro p_y se calcula con una versión suavizada de la estimación de máxima verosimilitud, representada como (Sarkar, 2016):

$$\hat{y_{yi}}=\frac{F_{yi}+\alpha}{F_y+\alpha n}$$

donde,

- $F_{yi}=\sum_{x\in T}x_i$ es la frecuencia de aparición de la característica i en una muestra para la etiqueta de clase y en el conjunto de entrenamiento T (Sarkar, 2016).
- $\sum_{i=1}^{|T|}F_{yi}$ es la frecuencia total de todas las características para la etiqueta de clase y (Sarkar, 2016).

Hay cierta cantidad de suavizado con la ayuda de $\alpha\geq 0$, lo que tiene en cuenta las características que no están presentes en los puntos de datos de entrenamiento y ayuda a deshacerse de los problemas relacionados con la probabilidad cero. Algunas configuraciones específicas para este parámetro se utilizan con bastante frecuencia. El valor de $\alpha=1$ se conoce como suavizado de Laplace y $\alpha<1$ se conoce como suavizado de Lidstone (Sarkar, 2016).

2.5. Algoritmos de regresión

En esta sección se abordarán y describirán algunos algoritmos utilizados en el ámbito de la regresión, con el propósito de ofrecer una visión general de sus características distintivas y enfoques, destacando así su importancia dentro de ML.

2.5.1. Regresión Lineal

La regresión lineal (LR, por sus siglas en inglés, *Linear Regression*) es un método estadístico ampliamente utilizado en el ML para modelar la relación entre una variable dependiente y una o más variables independientes, mediante una función lineal basada en las características del conjunto de datos de entrada (Burkov, 2019).

La LR se desarrolla a partir de una colección de ejemplos etiquetados $\{(x_i, y_i)\}_{i=1}^{N}$, donde N es el tamaño del conjunto, x_i, es el vector de características de dimensión D del ejemplo i, y_i es el valor objetivo real asociado a dicho ejemplo y se busca construir un modelo $f_{w,b}(x)$ como una combinación lineal de las características del ejemplo x (Burkov, 2019):

$$f_{w,b}(x) = w \cdot x \times b,$$

donde w es un vector de parámetros de dimensión D y b es un número real. La notación $f_{w,b}$ indica que el modelo f está parametrizado por dos valores: w y b.

Este modelo se utiliza para predecir el valor desconocido de y para un x de entrada, de la siguiente manera $y \leftarrow f_{w,b}(x)$. Sin embargo, dos modelos parametrizados por dos pares diferentes (w, b) producirá dos predicciones diferentes al aplicarse al mismo ejemplo. Por lo tanto, el objetivo es encontrar los valores óptimos (w^*, b^*) que minimicen el error de predicción (Burkov, 2019).

2.5.2. Regresión de Vectores de Soporte

Las SVM son ampliamente conocidas por su aplicación en tareas de clasificación, pero también pueden adaptarse para problemas de regresión, denominándose en estos casos Regresion de Vectores de Soporte (SVR por sus siglas en inglés de *Support Vector Regression*). La clave para utilizar SVM en regresión en lugar de clasificación radica en ajustar el objetivo: mientras que en clasificación se busca ajustar el margen más grande posible entre dos clases limitando las violaciones de margen, en SVR se intenta ajustar tantas instancias como sea posible dentro de un margen, limitando las violaciones de margen, es decir, las instancias que quedan fuera del margen. El ancho de este margen está controlado por un hiperparámetro ε. (Géron, 2023).

SVR utiliza una función de pérdida insensible (*epsilon-insensitive loss*) que ignora errores menores a ε. La idea principal es encontrar una función $f(x)$ que tenga una desviación máxima de ε del valor real para todos los datos de entrenamiento. Esta función de pérdida se define de manera que solo se penalizan los errores que exceden ε, permitiendo cierta flexibilidad en las predicciones (Schlkopf and Smola, 2001). Reducir ε aumenta el número de vectores de soporte, lo que ayuda a regularizar el modelo. Además, agregar más instancias de entrenamiento dentro del margen no afecta las predicciones del modelo, por lo tanto, se dice que el modelo es insensible a ε (Géron, 2023).

El objetivo en SVR es encontrar un hiperplano $f(x) = \langle w, x \rangle + b$ que tenga la mayor cantidad de puntos dentro de un margen de ancho 2ε. Este margen se define por dos líneas paralelas, una por encima y otra por debajo del hiperplano central. Además, SVR minimiza una combinación del error y la complejidad del modelo, resolviendo un problema de optimización que involucra variables de holgura para permitir que algunos puntos estén fuera del margen de ε, y un parámetro C que controla el equilibrio entre la complejidad del modelo y el error permisible (Schlkopf and Smola, 2001).

Para abordar tareas de regresión no lineal, SVR puede utilizar el truco del kernel. Este truco consiste en usar una función de kernel $K(x_i, x_j)$ que transforma los datos a un espacio de mayor dimensión donde es más fácil encontrar un hiperplano lineal que resuelva el problema. Los kernels más comunes son el kernel lineal, el kernel polinomial y el kernel RBF, cada uno con sus propias características y aplicaciones (Géron, 2023) .

SVR tiene varias ventajas, como su capacidad para manejar alta dimensionalidad y el uso eficiente de memoria, ya que solo utiliza un subconjunto de los datos de entrenamiento (vectores de soporte). Sin embargo, también presenta desventajas, como la complejidad en la elección del kernel y sus parámetros, y su sensibilidad a la escala de los datos, lo que hace recomendable normalizarlos antes de su uso.

2.6. Ensamble de modelos: Boosting

En el ML se han desarrollado diversas técnicas para mejorar la precisión de los resultados mediante la combinación de varios modelos. Boosting es uno de los algoritmos más efectivos de esta técnica, el cual combina varios modelos "débiles" para construir un modelo "fuerte" y más preciso (Theobald, 2020). Los modelos débiles, que individualmente tienen un desempeño apenas superior al de una selección aleatoria, se integran en un modelo final más robusto y confiable mediante una estrategia de corrección secuencial (Theobald, 2020), esto se puede ilustrar en la Figura 2.14.

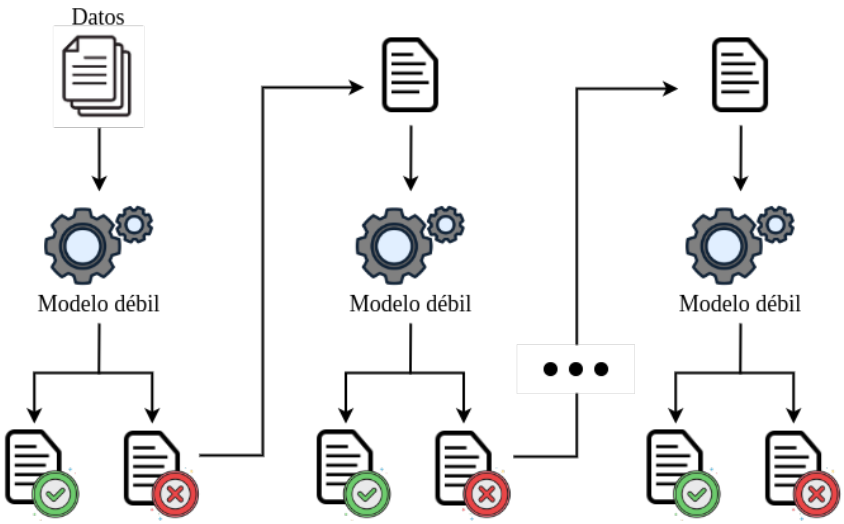


Figura 2.14: Representación de Boosting.

Una variante destacada de Boosting es el impulso de gradiente (en inglés *Gradient Boosting*). Este método optimiza la precisión del modelo mediante la construcción secuencial de DT (Theobald, 2020). A diferencia de otros enfoques de Boosting que ajustan los pesos de las instancias, *Gradient Boosting* se enfoca en corregir los errores de los modelos previos y cada nuevo árbol se entrena para abordar estos errores (Géron, 2023). Por lo tanto, se añaden estimadores uno a uno al conjunto, con el objetivo de minimizar los errores y mejorar el rendimiento global del modelo. Este enfoque interdependiente permite que los DT utilicen la información derivada de los árboles anteriores, en contraste con otros métodos que generan árboles de manera independiente (Theobald, 2020).

2.6.1. Regresor AdaBoost

AdaBoost (abreviatura en inglés de *Adaptive Boosting*) es un algoritmo de ML basado en el método de Boosting y aunque fue desarrollado inicialmente para problemas de clasificación, AdaBoost ha sido adaptado para tareas de regresión, donde tiene la capacidad de predecir valores continuos. El propósito principal de AdaBoost es crear un predictor potente al combinar de manera secuencial una serie de modelos débiles. En el contexto de la regresión, los predictores débiles son típicamente modelos simples como DT de baja profundidad. AdaBoost adapta estos modelos simples a los errores de los modelos anteriores, concentrándose iterativamente en los ejemplos que han sido mal predichos (Hastie et al., 2009).

El proceso del algoritmo comienza con la asignación de un peso inicial uniforme a cada ejemplo del conjunto de datos (Efron and Hastie, 2016). Estos pesos indican la importancia de cada ejemplo en la iteración actual. En cada iteración t, se entrena un nuevo modelo h_t utilizando los datos ponderados (Hastie et al., 2009). Este modelo busca predecir el valor de la variable objetivo minimizando el error ponderado mediante el ajuste de sus parámetros en función de

los pesos asignados a los ejemplos (Efron and Hastie, 2016).

Para un ejemplo i, el peso w_i en la iteración $t+1$ se actualiza según la fórmula:

$$\mathrm{w}_i^{(t+1)} exp(\alpha_t \cdot |y_i - h_t(x_i)|)$$

donde $\mathrm{w}_i^{(t)}$ es el peso del ejemplo i en la iteración t, α_t es la ponderación del modelo en la iteración t, y_i es el valor real del ejemplo i, y $h_t(x_i)$ es la predicción del modelo h_t para el ejemplo i. La actualización de los pesos asegura que los ejemplos mal predichas recibirán más atención en las siguientes iteraciones (Efron and Hastie, 2016).

Luego, se calcula el error del modelo h_t evaluando la diferencia entre las predicciones y los valores reales, teniendo en cuenta los pesos de los ejemplos. Posteriormente, se determina una ponderación α_t para el modelo h_t basada en su error (Hastie et al., 2009). Los modelos con menor error reciben mayor importancia, aumentando su influencia en la predicción final. Los pesos de los ejemplos se actualizan para enfatizar aquellos que fueron mal predichos, los cuales reciben mayor peso en las iteraciones subsiguientes, obligando a los modelos futuros a concentrarse en estos ejemplos difíciles (Efron and Hastie, 2016). Finalmente, las predicciones finales se obtienen mediante una combinación ponderada de las predicciones de todos los modelos entrenados. La combinación se realiza sumando las predicciones ponderadas de cada modelo (Hastie et al., 2009). La fórmula general para la predicción final es:

$$\hat{y} = \sum_{t=1}^{T} \alpha_t \cdot h_t(x)$$

donde $\hat{y}$ es la predicción final, α_t es la ponderación del modelo en la iteración t, y $h_t(x)$ es la predicción del modelo h_t para la muestra x (Hastie et al., 2009).

AdaBoost tiene varias ventajas, puede convertir modelos débiles en un modelo fuerte y preciso. Además, al enfocarse iterativamente en las muestras mal predichas, el algoritmo mejora continuamente la precisión (Bishop, 2006). Su flexibilidad permite utilizarse con una variedad de modelos básicos, aunque comúnmente se emplean DT de baja profundidad (Efron and Hastie, 2016). Sin embargo, también presenta algunas limitaciones. AdaBoost puede ser sensible al ruido en los datos, ya que los ejemplos mal predichos reciben más peso, incluyendo los errores debidos al ruido (Hastie et al., 2009). Además, el proceso iterativo puede ser computacionalmente intensivo, especialmente para grandes conjuntos de datos (Theobald, 2020).

2.6.2. Regresor XGBoost

El algoritmo XGBoost (abreviatura en inglés de *eXtreme Gradient Boosting*) es una herramienta avanzada de ML que utiliza un conjunto de DT basados en la técnica de *Gradiend Boosting*. Este método está diseñado para ser altamente eficiente y escalable, permitiendo crear modelos predictivos precisos minimizando una función de pérdida y controlando la complejidad de los árboles (Bentéjac et al., 2021).

La función de pérdida en XGBoost se expresa como:

$$L_{xgb} = \sum_{i=1}^{N} L(y_i, F(x_i)) + \sum_{m=1}^{M} \omega(h_m)$$

donde $\omega(h)$ se define como:

$$\omega(h) = \gamma T + \frac{1}{2}\alpha \|x\|^2$$

En la fórmula anterior, T es el número de hojas en el árbol, y w representa las puntuaciones de salida de las hojas. Esta función de pérdida se utiliza en el criterio de división de los DT, lo que permite la poda anticipada de nodos. Un valor mayor de γ resulta en árboles más simples, ya que controla la ganancia mínima necesaria para dividir un nodo. Además, XGBoost emplea otras técnicas de regularización, como limitar la profundidad de los árboles, para reducir la complejidad del modelo y mejorar la eficiencia del entrenamiento (Bentéjac et al., 2021).

XGBoost emplea técnicas de aleatorización, como el submuestreo aleatorio de datos y columnas, para mitigar el sobreajuste y acelerar el proceso de entrenamiento. La adaptación de la función de pérdida es posible mediante la definición de gradientes y hessianos específicos, ajustándose así a diversos problemas. Para abordar la escasez de datos, XGBoost excluye automáticamente entradas con valores cero o faltantes al calcular ganancias, y gestiona nodos predeterminados para estos valores. También ofrece restricciones monótonas y de interacción de características, que garantizan una salida consistente y limitan las combinaciones de atributos en las divisiones del árbol (Bentéjac et al., 2021).

Para mejorar la velocidad de entrenamiento, XGBoost reduce la complejidad computacional utilizando almacenamiento en columnas comprimidas y preordenación de datos, lo que permite realizar divisiones en paralelo(Bentéjac et al., 2021). Además, usa métodos basados en percentiles para probar un subconjunto de divisiones candidatas, calculando su ganancia mediante estadísticas agregadas, lo que acelera el proceso sin comprometer la precisión (Wade, 2020).

2.6.3. LightGBM

LightGBM (acrónimo en inglés de *Light Gradient Boosting Machine*) es una implementación del DT de impulso de gradiente (GBDT, por sus siglas en inglés de *Gradient Boosting Decision Tree*) desarrollada por Microsoft®. Este algoritmo se ha destacado por su alta eficiencia, velocidad y capacidad para manejar grandes conjuntos de datos, lo que lo convierte en una opción popular en el ámbito del ML (Bentéjac et al., 2021).

El algoritmo LightGBM se basa en los principios de *Gradient Boosting* y una de las características distintivas es su estrategia de división basada en hojas (*leaf-wise*) como se ilustra en la Figura 2.15, que lo diferencia de otros algoritmos de Boosting que utilizan una estrategia de crecimiento nivel por nivel (*level-wise*). Esta técnica divide primero las hojas con la mayor reducción de pérdida, lo que conduce a una reducción más rápida de la misma y, en general, a modelos más precisos[1]. Para mejorar aún más su eficiencia, LightGBM emplea dos técnicas avanzadas: GOSS (del inglés *Gradient-based One-Side Sampling*) y EFB (del inglés *Exclusive Feature Bundling*). GOSS reduce el número de datos utilizados en cada iteración seleccionando todos los datos con grandes gradientes y solo una parte de los datos con gradientes pequeños y ajusta los pesos de los datos seleccionados para mantener la contribución proporcional de los gradientes. Esto reduce el tiempo de cómputo manteniendo la precisión del modelo. Por otro lado, EFB agrupa características que son exclusivas entre sí (es decir, características que rara vez tienen valores no nulos simultáneamente) en una sola característica, reduciendo así la dimensión del conjunto de datos y el uso de memoria sin perder información relevante (Bentéjac et al., 2021).

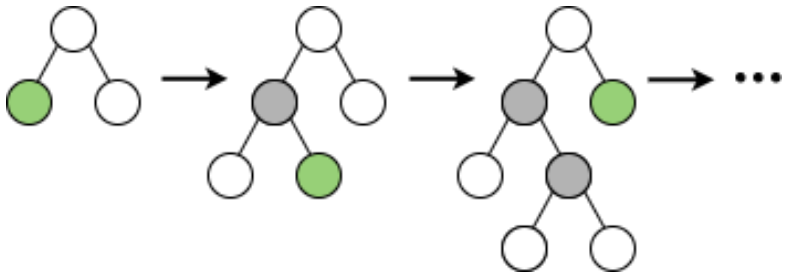


Figura 2.15: División basa en hojas (*leaf-wise*) en LightGBM.

Además, LightGBM convierte los datos continuos en histogramas antes de buscar los puntos de división óptimos. Este enfoque reduce significativamente el tiempo de cómputo y el uso de memoria, mejorando la eficiencia del algoritmo. Otra ventaja importante es su capacidad para manejar características categóricas de forma nativa sin necesidad de preprocesamiento adicional, como la codificación *one-hot*. Esto simplifica el flujo de trabajo y mejora la eficacia del modelo[1].

[1] `https://lightgbm.readthedocs.io/en/stable/#welcome-to-lightgbm-s-documentation`

LightGBM también soporta el entrenamiento distribuido y el aprendizaje paralelo, lo que lo hace adecuado para entornos de computación en paralelo y en clústeres, permitiendo así el manejo eficiente de grandes volúmenes de datos. Además, incluye técnicas de regularización como $L1$ y $L2$ para prevenir el sobreajuste y mejorar la capacidad de generalización del modelo[1].

Este algoritmo es utilizado en una amplia gama de aplicaciones, incluyendo la clasificación, regresión y ranking, donde se requiere ordenar elementos, como en los motores de búsqueda y sistemas de recomendación. Entre las principales ventajas de LightGBM se encuentran su eficiencia y velocidad, siendo significativamente más rápido y eficiente en comparación con otras implementaciones de boosting y su capacidad para manejar datos categóricos directamente simplifica el preprocesamiento de los datos[1].

2.7. Selección de características

La selección de características es un componente clave en el ML, que tiene como objetivo identificar un subconjunto óptimo de variables a partir de un conjunto más amplio para mejorar el desempeño del modelo predictivo (Liu and Motoda, 1998). Este proceso se lleva a cabo siguiendo criterios específicos, como la relevancia y redundancia de las características, así como su importancia en los resultados finales. Su propósito es maximizar la precisión del modelo, reducir su complejidad y prevenir el sobreajuste (Liu and Motoda, 1998). Al descartar características irrelevantes o redundantes, se logra no solo acortar el tiempo de entrenamiento, sino también evitar complicaciones relacionadas con la alta dimensionalidad, la cual puede afectar negativamente la capacidad del modelo para generalizar sobre nuevos datos cuando hay un exceso de variables (Liu and Motoda, 1998).

En consecuencia, la selección de características puede describirse como la tarea de identificar aquellas variables que mejor contribuyen al rendimiento del modelo. La intención es establecer una correspondencia adecuada entre las características y el resultado, reduciendo la dimensionalidad sin sacrificar la capacidad predictiva (Kuhn and Johnson, 2019). Existen múltiples algoritmos y técnicas diseñados para llevar a cabo este proceso, a continuación, se presentarán algunos de los enfoques.

2.7.1. Árboles extremadamente aleatorios

El algoritmo de Árboles Extremadamente Aleatorios o ExtraTrees (abreviatura en inglés de *Extremely Randomized Trees*) se basa en la creación de un conjunto de DT, utilizando un proceso que introduce una alta aleatoriedad en su construcción, que a diferencia de otros métodos

basados en árboles, se distingue por dos características principales: la selección completamente al azar de puntos de corte para dividir los nodos y el uso de toda la muestra de entrenamiento (en lugar de una muestra bootstrap) para hacer crecer los árboles (Geurts et al., 2006).

ExtraTrees selecciona un subconjunto aleatorio de características durante la construcción de los árboles y divide los nodos de cada árbol eligiendo divisiones al azar entre un conjunto predefinido de posibles divisiones dentro del subconjunto de características, por lo tanto, en lugar de buscar los umbrales de división más discriminantes, se generan umbrales al azar para cada característica candidata y se elige el mejor de estos umbrales aleatorios. Este enfoque reduce aún más la varianza del modelo, aunque puede incrementar ligeramente el sesgo (Géron, 2023). Durante la fase de predicción, ExtraTrees emplea el método de votación mayoritaria, es decir, cada árbol en el bosque emite una decisión de clasificación, y la clase que recibe el mayor número de votos se considera la predicción final del modelo (Geurts et al., 2006).

ExtraTrees suele ser más rápido en comparación con métodos que buscan divisiones óptimas, esto lo hace adecuado para grandes conjuntos de datos (Mueller and Luca, 2021). Sin embargo, además de todas las características, ExtraTrees se puede utilizar particularmente en la selección de características debido a su capacidad para evaluar la contribución de cada característica en la clasificación a lo largo de todos los árboles en el bosque, proporcionando una medida de importancia que ayuda a identificar las variables más relevantes que influyen significativamente en las predicciones del modelo. Esta capacidad es crucial para reducir la bidimensionalidad de las características y mejorar la eficiencia de los modelos (Géron, 2023).

2.7.2. Análisis de Varianza

El Análisis de Varianza o ANOVA (por su abreviatura en inglés de *Analysis of Variance*) es una técnica estadística utilizada para determinar si existen diferencias significativas entre las medias de tres o más grupos independientes (Devore, 2015). Este método se basa en comparar la variabilidad observada dentro de los grupos con la variabilidad entre los grupos, con el objetivo de evaluar si las diferencias en las medias de los grupos son lo suficientemente grandes como para no ser simplemente producto del azar (Moore et al., 2007).

En el ML dentro del proceso de selección de características, ANOVA permite evaluar la capacidad de cada característica para distinguir entre diferentes clases de la variable objetivo y seleccionar aquellas características que aportan información valiosa y relevante para el modelo predictivo (Moore et al., 2007).

La formulación del problema en ANOVA implica definir las hipótesis:

- Hipótesis nula (H_0): establece que las medias de la característica en cuestión son iguales

en todas las clases de la variable objetivo (Moore et al., 2007).

- Hipótesis alternativa (H_a): sostiene que al menos una de las medias de la característica es diferente entre las clases (Moore et al., 2007).

ANOVA utiliza la estadística F para evaluar la relación entre la variabilidad entre grupos y la variabilidad dentro de los grupos. Un valor alto de la estadística F sugiere que la variabilidad entre las medias de los grupos es mayor que la variabilidad interna, indicando que las diferencias entre los grupos no se deben al azar (Devore, 2015). La fórmula de la estadística F se define como la razón de la variabilidad entre grupos sobre la variabilidad dentro de los grupos, ajustada por sus respectivos grados de libertad (Moore et al., 2007).

La variabilidad entre grupos o suma de cuadrados entre grupos (SSBG, por sus siglas en inglés de *Sum of Squares Between Groups*), mide la variabilidad de las medias de las características entre los diferentes grupos o categorías de la variable objetivo. Se calcula para determinar cuánto varían las medias de los grupos en comparación con la media global de la característica (Moore et al., 2007). La fórmula para calcular SSBG es:

$$SSBG = \sum_{i=1}^{k} n_i(\bar{X}_i - \bar{X})^2$$

donde:

- k es el número de grupos.
- n_i es el número de observaciones en el grupo i.
- $\bar{X}_i$ es la media de la característica en el grupo i.
- $\bar{X}$ es la media global de la característica, calculada como la media de todas las observaciones en todos los grupos.

La variabilidad dentro de los grupos o Suma de Cuadrados Dentro de Grupos (SSWG, por sus siglas en inglés de *Sum of Squares Within Groups*) o , representa la dispersión de las observaciones dentro de los grupos alrededor de sus medias de grupo respectivas(Moore et al., 2007). La fórmula para calcular $SSWG$ es:

$$SSWG = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (X_{ij} - \bar{X}_i)^2$$

donde

- k es el número de grupos.
- n_i es el número de observaciones en el grupo i.
- X_{ij} es el valor de la característica para la observación j en el grupo i.
- $\bar{X}_i$ es la media de la característica en el grupo i.

La fórmula para calcular la estadística F es:

$$F = \frac{MSBG}{MSWG}$$

donde:

La Media de Cuadrados Entre Grupos (MSBG, por sus siglas en inglés de *Mean Square Between Groups*) se calcula dividiendo *SSBG*, entre sus grados de libertad (Moore et al., 2007). La formula es:

$$MSBG = \frac{SSBG}{k-1}$$

donde, k es el número de grupos y, $k-1$ son los grados de libertad entre grupos, que representan el número de grupos menos uno.

La Media de Cuadrados Dentro de Grupos (MSWG, por sus siglas en inglés de *Mean of Squares Within Groups*) se calcula dividiendo $SSWG$ entre sus grados de libertad (Moore et al., 2007). La fórmula es:

$$MSWG = \frac{SSWG}{N-k}$$

donde, N es el número total de observaciones y $N-k$ son los grados de libertad dentro de grupos, que representan el número total de observaciones menos el número de grupos.

La estadística F sigue una distribución específica conocida como distribución F, que se utiliza para calcular el valor p. El valor p proporciona una medida cuantitativa para evaluar la significancia estadística de las diferencias observadas en los datos. La significancia estadística se evalúa comparando el valor p con el nivel de significancia predefinido (α). Si el valor p es menor que α, se rechaza la hipótesis nula (H_0) en favor de la hipótesis alternativa (H_a). Esto indica que existen diferencias significativas en las medias entre al menos algunos de los grupos. En caso contrario, si el valor p es mayor o igual a α, no se rechaza la hipótesis nula. Esto sugiere que no hay suficiente evidencia para afirmar que las medias de los grupos son significativamente diferentes (Devore, 2015).

En la selección de características mediante ANOVA, los valores p se emplean para determinar la significancia estadística de las características en relación con la variable objetivo. Las características que presentan valores p menores que el nivel de significancia se consideran significativas, lo que indica que tienen una capacidad destacada para diferenciar entre los grupos de la variable objetivo y, por ende, son relevantes para el modelo predictivo. Por el contrario, las características con valores p mayores o iguales al nivel de significancia no evidencian diferencias significativas en las medias entre los grupos y, por lo tanto, pueden ser consideradas

menos informativas (Moore et al., 2007).

2.7.3. Coeficiente de Correlación de Pearson

El coeficiente de correlación de Pearson (r) es una medida estadística que indica la dirección y fuerza de la relación entre dos variables cuantitativas (Moore et al., 2007). Se calcula mediante:

$$r = \frac{1}{n-1} \sum \left(\frac{x_i - \bar{x}}{s_x} \right) \left(\frac{y_i - \bar{y}}{s_y} \right)$$

donde:

- n es el número total de pares de las variables X y Y.
- x_i y y_i son los valores individuales de las variables X y Y, respectivamente.
- $\bar{x}$ y $\bar{y}$ son las medias de las variables X y Y, respectivamente.
- s_x es la desviación estándar de las variables X.
- s_y es la desviación estándar de las variables Y.

La correlación r siempre es un número entre -1 y 1. Mientras r se acerca o es igual a -1, la relación entre las variables es negativa, es decir, a medida que una variable aumenta la otra disminuye. Por otro lado, cuando r se acerca o es igual a 1, la relación entre las variables es positiva, esto significa que cuando una variable aumenta la otra también lo hace. Finalmente, si r es cercano o igual a 0 indica que la relación entre las variables es débil, esto quiere decir que no hay una tendencia discernible en la forma en que una variable cambia en relación con la otra (Moore et al., 2007).

En el ámbito del ML, el coeficiente de correlación permite identificar características (o variables) redundantes en un conjunto de datos. Su importancia radica en la capacidad para evaluar la relación entre pares de características. Aquellas características que presentan una correlación cercana a -1 o 1 ofrecen información redundante al modelo, dado que aportan datos similares. Por consiguiente, mantener ambas características no añade valor adicional y puede incrementar innecesariamente la complejidad del modelo. Por lo tanto, se recomienda eliminarlas para simplificar el modelo y optimizar su rendimiento.

2.8. Evaluación

La evaluación de un modelo de ML se enfoca en la medición y el análisis de su eficacia al realizar predicciones precisas en datos desconocidos. Este proceso es esencial en el desarrollo de modelos de ML, ya que permite determinar si un modelo es idóneo para cumplir su propósito. La evaluación involucra múltiples etapas y técnicas, que pueden variar dependiendo del tipo de problema y del modelo en cuestión.

2.8.1. Métricas de evaluación

Las métricas de evaluación están vinculadas a las tareas de ML. Existen diferentes métricas para las tareas de clasificación, regresión, agrupación, etc. A continuación se describen algunas de las métricas de evaluación utilizadas en modelos de aprendizaje supervisado y modelos de aprendizaje semisupervisado.

Si consideramos una clasificación binaria teniendo en cuenta las clases *Positivo* y *negativo* entre las que el modelo deberá realizar las predicciones, tenemos:

- **TP:** Cantidad de elementos *positivos* que el modelo identifico correctamente al clasificar.
- **TN:** Cantidad de elementos *negativos* que el modelo identifico correctamente al clasificar.
- **FP:** Cantidad de elementos que el modelo identifico incorrectamente como *positivos* a la hora de hacer la clasificación.
- **FN:** Cantidad de elementos que el modelo identifico incorrectamente como *negativos* a la hora de hacer la clasificación.

Exactitud

Esta métrica cuantifica con qué frecuencia el clasificador acierta en sus predicciones. Se obtiene al calcular la proporción entre el número de predicciones acertadas y el número total de predicciones realizadas. En otras palabras, la exactitud nos brinda una medida simple y directa de la capacidad del clasificador para acertar en la clasificación de los ejemplos en el conjunto de datos, evaluando qué parte de las predicciones coincide con la verdad absoluta en comparación con el total de predicciones realizadas. (Zheng, 2015).

$$Exactitud = \frac{TP+TN}{P+N}$$

Precisión

La métrica de precisión desempeña un papel clave al abordar la siguiente cuestión: De todos los elementos que el clasificador ha identificado como relevantes, ¿cuántos de ellos resultan ser genuinamente relevantes? (Zheng, 2015). En términos más concretos, la precisión cuantifica la proporción de predicciones etiquetadas como positivas que realmente corresponden a casos positivos en el conjunto de datos. Esta métrica se emplea específicamente cuando el objetivo principal consiste en minimizar la cantidad de falsos positivos, es decir, cuando se busca asegurar que las predicciones positivas realizadas por el modelo sean altamente confiables y reflejen con precisión las instancias verdaderamente positivas en el conjunto de datos (Müller and Guido, 2016).

$$Precision = \frac{TP}{TP + FP}$$

Exhaustividad

La tasa de verdaderos positivos, exhaustividad o recall, se encarga de medir cuántas de las instancias positivas reales son correctamente identificadas por las predicciones positivas del modelo. Esta métrica adquiere una importancia particular cuando el objetivo primordial consiste en asegurar que todas las muestras positivas sean detectadas, evitando así la omisión de casos verdaderamente positivos. En otras palabras, se utiliza cuando la minimización de los falsos negativos, es decir, la omisión de ejemplos que deberían haber sido identificados como positivos, es una consideración crítica para el problema en cuestión (Müller and Guido, 2016).

$$Exhaustividad = \frac{TP}{TP + FN}$$

Medida F

Si bien, la precisión y el recall son medidas muy importantes, observar solo una de ellas no le brindará una visión completa. Una forma de resumirlos es la medida F o f-score, esta métrica combina la precisión y el recall en una sola medida, la cual es útil cuando deseas tener en cuenta tanto la capacidad de un modelo para identificar correctamente ejemplos positivos (recall), como su capacidad para evitar falsos positivos (precision) (Müller and Guido, 2016) (Zheng, 2015).

$$Medida\ F = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

Sensibilidad

La sensibilidad, también conocida como tasa de verdaderos positivos, cuantifica la proporción de instancias positivas que son correctamente identificadas por el modelo. Esta métrica evalúa la eficacia del modelo en la predicción de la clase positiva, proporcionando una medida de su capacidad para detectar correctamente los casos que pertenecen a esta categoría (Brownlee, 2020).

$$Sensibilidad = \frac{TP}{TP + FN}$$

Especificidad

La especificidad, que representa la tasa de verdaderos negativos, es el complemento de la sensibilidad y evalúa la capacidad del modelo para predecir correctamente la clase negativa (Brownlee, 2020).

$$Especificidad = \frac{TN}{TN+FP}$$

Exactitud Balanceada

La exactitud balanceada (en inglés *Balanced Accuracy*) es una métrica de evaluación utilizada en problemas de clasificación, especialmente cuando se trata de conjuntos de datos desequilibrados[2]. Su propósito es proporcionar una medida más equitativa del rendimiento del modelo, evitando que las métricas de evaluación se vean infladas debido a la desproporción en las clases[2].

Para problemas de clasificación multiclase, la exactitud balanceada se define como el promedio de la exhaustividad obtenida en cada clase[2]. Es decir, calcula la precisión media de cada clase, dando igual importancia a cada una, independientemente de su frecuencia en el conjunto de datos.

$$Exactitud\ balanceada = \frac{1}{n_{clases}} \sum_{i=1}^{n_{clases}} \frac{TP_i}{TP_i + FN_i}$$

donde n_{clases} es el número total de clases, TP_i es el número de verdaderos positivos para la clase i, y FN_i es el número de falsos negativos para la clase i.

En el caso de clasificación binaria, la exactitud balanceada es igual a la media aritmética de la sensibilidad y la especificidad:

$$Exactitud\ balanceada = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right)$$

De este modo, en conjuntos de datos equilibrados, la exactitud balanceada es igual a la precisión general. Sin embargo, en conjuntos de datos desbalanceados, ofrece una evaluación más justa al dar un peso igual a cada clase, lo que ayuda a mitigar el sesgo hacia las clases mayoritarias.

2.8.2. Validación cruzada

La validación cruzada (CV, por sus siglas en inglés de *Cross Validation*) es un método estadístico ampliamente utilizado para evaluar el rendimiento de generalización de un modelo de manera más robusta y completa que simplemente dividir los datos en un conjunto de entrenamiento y un conjunto de prueba. En lugar de una sola división, la validación cruzada divide los datos en múltiples partes llamadas *pliegues* y entrena varios modelos en un proceso iterativo

[2] `https://scikit-learn.org/stable/modules/model_evaluation.html#balanced-accuracy-score`

(Müller and Guido, 2016).

En la Figura 2.16 se ilustra la forma más común de validación cruzada, conocida como *validación cruzada k-fold*, donde *k* representa un número específico. En este enfoque, los datos se dividen en *k* pliegues de aproximadamente igual tamaño. Luego, se construyen y evalúan *k* modelos diferentes. El primer modelo se entrena utilizando el primer pliegue como conjunto de prueba y los pliegues restantes (2, 3, 4, 5) como conjunto de entrenamiento. Luego, se construye otro modelo, utilizando el segundo pliegue como conjunto de prueba y los pliegues 1, 3, 4, 5 como conjunto de entrenamiento, y así sucesivamente. Este proceso se repite para cada uno de los *k* pliegues (Müller and Guido, 2016).

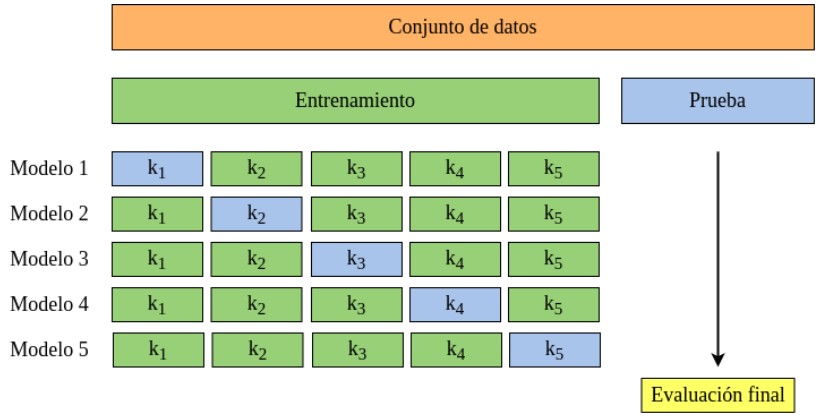


Figura 2.16: Validación Cruzada K-Folds.

Para cada una de estas divisiones de datos en conjuntos de entrenamiento y prueba, se calcula una métrica de rendimiento, como la precisión. El rendimiento general del modelo se determina como el promedio de las métricas de rendimiento calculadas en todos los *k* pliegues. Esto proporciona una evaluación más confiable del rendimiento del modelo, ya que se asegura de que el modelo se ajuste y se evalúe en diferentes subconjuntos de datos, lo que ayuda a reducir el sesgo y aumentar la robustez de la evaluación del rendimiento (Zheng, 2015) (Müller and Guido, 2016).

2.9. Trabajos relacionados

En los últimos años, el análisis de textos ha emergido como una herramienta en la identificación de trastornos psicológicos, como la ansiedad. Dentro de este campo, múltiples enfoques han sido propuestos, entre ellos los métodos supervisados y semisupervisados. Estos enfoques no solo permiten una comprensión más profunda de los patrones lingüísticos asociados con la ansiedad, sino que también facilitan la creación de modelos predictivos que pueden ser implementados en diversas aplicaciones prácticas. A continuación, se presentan algunas de las investigaciones en este ámbito, que han contribuido en el desarrollo de técnicas para la detección de

ansiedad a través del análisis de textos.

2.9.1. Enfoques de aprendizaje supervisado

El estudio de Byers et al. (2023) presenta la implementación de un modelo de ML para la detección de ansiedad en estudiantes. Este modelo se desarrolló a partir de datos obtenidos de la transcripción de diez entrevistas realizadas a estudiantes sobre sus experiencias con la ansiedad. Las entrevistas se segmentaron en sentencias, creando así un corpus de 1,187 documentos, los cuales fueron etiquetados por expertos en salud mental para identificar cuatro niveles de ansiedad: sin ansiedad, presente en el 24.3 % de los casos; baja, con un 46.25 %; media, con un 27.93 %; y alta, observada en el 1.52 % de los documentos. Durante el preprocesamiento de los datos se convirtió el texto a minúsculas y se eliminaron signos de puntuación. Para la extracción de características, se emplearon N-Gramas (unigramas, bigramas y trigramas), seguidos de una selección de características utilizando *LASSO*, X^2, $L1$ y algoritmos basados en árboles. En el desarrollo del modelo, se llevó a cabo una búsqueda en malla para optimizar los hiperparámetros, junto con una validación cruzada de $k = 10$ en los algoritmos NB, DT, SVM y Regresión Logística. La SVM, con características seleccionadas por X^2, mostró los mejores resultados, con un intervalo de confianza del 7.2 % y una exactitud del 59.1 %.

En el trabajo de Nova (2023), se abordó la clasificación de trastornos mentales mediante el análisis de textos obtenidos de la red social Reddit. Los textos permitieron la creación de un corpus con 10,000 documentos, los cuales fueron organizados en tres clases: trastorno limite de la personalidad (TLP), que comprendió el 35.34 % de los textos; trastorno bipolar, con un 7.11 %; y una categoría denominada “otros”, que representó el 57.55 % e incluyó condiciones como ansiedad, depresión, esquizofrenia y otras enfermedades mentales. Los datos analizados consistieron en los títulos y contenidos de las publicaciones. Durante el preprocesamiento, se eliminaron URL’s, signos de puntuación y palabras vacías, y se llevó a cabo la extracción de características mediante el método TF-IDF. Los algoritmos empleados para la clasificación incluyeron LightGBM, NB y MLP. LightGBM demostró ser el más efectivo, alcanzando una exactitud del 72.4 % en la clasificación de títulos y del 77 % en el análisis de los textos completos.

En el trabajo realizado por Yu et al. (2023), se desarrolló un modelo para predecir el estado de ansiedad de usuarios de la red social Sina Weibo en China. Para este propósito, se invitó a los usuarios a completar la Escala de Autoevaluación de Ansiedad (SAS, por sus siglas en inglés de *Self-Rating Anxiety Scale*), cuyos resultados se expresaron en valores enteros dentro de un rango de 0 a 100. Posteriormente, se recopilaron los textos originales de sus publicaciones en la red social, conformando un conjunto de datos compuesto por 1,039 textos etiquetados con los resultados de la autoevaluación, los cuales fueron utilizados para el entrenamiento del mode-

lo. Durante el proceso de preprocesamiento de los datos, se eliminaron elementos no deseados del texto, como caracteres especiales y URLs. En la fase de extracción de características, se analizaron los textos utilizando el diccionario SC-LIWC (por sus siglas en inglés de *Simplified Chinese-Linguistic Inquiry and Word Count*), que permite contar el número de palabras de cada tipo y calcular el porcentaje de palabras con significado psicológico o lingüístico, abarcando dimensiones emocionales, cognitivas y sociales. En la etapa de selección de características, se calcularon los coeficientes de correlación entre todas las características y se contabilizó cuántas veces el coeficiente de correlación entre cada característica y otras superó un umbral de 0.4. Se estableció que una característica sería considerada redundante y, por ende, eliminada si presentaba una alta correlación con al menos otras 30 características. Posteriormente, se empleó el algoritmo RF para calcular la importancia de las características restantes, seleccionándose las 22 más significativas. Para la generación del modelo se utilizaron cuatro algoritmos de regresión: LR, SVR, Regresor XGBoost y Regresor AdaBoost. Sin embargo, el Regresor XGBoost fue quien obtuvo mejores resultados, con un Error Absoluto Medio de 4.57 y una correlación de Pearson de 0.32 entre los resultados reales y los obtenidos por el modelo.

2.9.2. Enfoques de aprendizaje semisupervisado

En la investigación desarrollada por Saifullah et al. (2021), se realizó la detección de ansiedad en comentarios tomados de YouTube®. Se consideró que los datos que no representaban un caso de ansiedad estaban asociados a sentimientos positivos, mientras que aquellos que expresaban sentimientos negativos se etiquetaron como ansiedad. El conjunto de datos utilizado en el estudio constaba de 4,862 comentarios, con una distribución del 33.96 % para la clase positiva y del 66.04 % para la clase negativa. Las etapas de preprocesamiento incluyeron la detección de emociones basada en el análisis de sentimientos (positivos y negativos) y pruebas de validación. Durante el preprocesamiento, se llevaron a cabo tareas como la tokenización, filtrado, derivación, etiquetado y conversión de emoticones en cada uno de los textos. Para el análisis, se evaluaron seis algoritmos: K-NN, Bernoulli, DT, SVM, RF y XGBoost. Los mejores resultados se obtuvieron con K-NN en términos de precisión, mientras que XGBoost sobresalió en la recuperación. Sin embargo, RF destacó en exactitud, alcanzando un 84.99 % al extraer características utilizando el método de Bolsa de Palabras y un 82.63 % con el método TF-IDF.

En el trabajo realizado por Asra et al. (2021), se propone DASentimental, un modelo de ML semisupervisado para identificar niveles de Depresión, Ansiedad y Estrés (DAS, por sus siglas en inglés de *Depression, Anxiety and Stress*) en textos. Para la creación del modelo, se destaca el uso de cuatro conjuntos de datos. Un conjunto compuesto por una colección de recuerdos emocionales (ERT, por sus siglas en inglés de *Emotional Recall Task*), obtenidos de 200 participantes, quienes expresaron las emociones experimentadas en el último mes mediante listas

de palabras. Además, se les aplicó la prueba DASS-21 (por sus siglas en inglés de *Depression, Anxiety and Stress Scale*), una escala estandarizada para medir sus niveles DAS. Así también un conjunto de datos que mapea la memoria semántica humana a través de asociaciones libres (*The Small World of Words*), esto significa que se tienen asociaciones conceptuales donde una palabra provoca el recuerdo de otra. Asimismo, se consideró un conjunto de datos que incluye normas de valencia y excitación para más de 20,000 palabras. Por último, para poner a prueba el modelo, se empleó un corpus sobre notas de suicidio. Durante el proceso, se realizó la limpieza de datos y la representación vectorial de las variables (características), así como de las respuestas de los niveles DAS, aplicando una regularización $L2$ a los vectores de características. A continuación, se llevó a cabo el entrenamiento, validación cruzada y selección del modelo de regresión de mejor rendimiento para estimar los niveles DAS a partir de los datos ERT. Posteriormente, se estimaron los niveles de DAS en las notas de suicidio analizando las secuencias de palabras emocionales en cada una, y se validó el etiquetado predicho por DASentimental a través de normas afectivas independientes. Los algoritmos empleados para el desarrollo del modelo fueron DT, MLP y una Red Neuronal Recurrente (LSTM, por sus siglas en inglés de *Long Short Term Memory*). Cada uno de estos modelos fue evaluado en términos de Error Cuadrático Medio (MSE, por sus siglas en inglés de *Mean Square Error*) y correlación de Pearson. Los resultados indicaron que MLP, entrenado con secuencias de palabras, proporcionó las mejores predicciones, obteniendo una correlación de Pearson en validación cruzada de 0.7 para depresión, 0.44 para ansiedad y 0.52 para estrés.

En el trabajo de Heri Cahyana et al. (2022), se abordó la detección de odio mediante el análisis de textos extraídos de comentarios en YouTube®. Este análisis permitió la creación de un corpus compuesto por 13,169 documentos, de los cuales 2,370 fueron etiquetados por expertos para clasificar tres niveles de odio: *"Mucho odio"*, *"Odio"* y *"Sin odio"*. Durante el preprocesamiento de los textos, se empleó el método TF-IDF para la extracción de características, y se desarrolló un modelo de aprendizaje automático utilizando el algoritmo K-NN con un enfoque semisupervisado. En este enfoque, los textos etiquetados se utilizaron para entrenar un modelo inicial, que luego se empleó para pseudoetiquetar los textos restantes. Tras la predicción de etiquetas para los datos no etiquetados, se implementó un proceso de votación para determinar la etiqueta final de cada registro. Este proceso consideró dos tipos de ponderaciones: una relacionada con la polaridad del discurso de odio y otra con la polaridad del discurso sin odio, calculadas en función de la confianza del modelo en sus predicciones. Si alguna de estas puntuaciones superaba un umbral del 80 % de votos en favor de alguna de las clase, el registro se consideraba etiquetado con suficiente confianza y se añadía al conjunto de datos de entrenamiento. En caso contrario, el registro se sometía a un nuevo ciclo de pseudoetiquetado. Si después de tres ciclos el registro no alcanzaba la confianza requerida, se procedía a un etiquetado manual por parte de expertos. Al final de este proceso, el modelo fue evaluado con un conjunto de datos de prueba con 1,317 ejemplos nuevos etiquetados por expertos, obteniendo

una exactitud del 59.68 %.

2.10. Resumen

El NLP se define como la capacidad de las tecnologías computacionales para interpretar y procesar de manera automática o semiautomática el lenguaje humano. Esta tecnología permite el desarrollo de diversas tareas, como el reconocimiento de voz, la generación de resúmenes, el análisis de sentimientos, la traducción automática y la clasificación de textos. En particular, la clasificación de textos es fundamental en este trabajo, ya que se utiliza para la detección de ansiedad.

El proceso de clasificación de textos implica el desarrollo de un sistema que consta de varias etapas, entre las cuales destacan el preprocesamiento de los datos, la normalización del texto, la extracción de características, el entrenamiento del modelo, la predicción y la evaluación del mismo. Las etapas de normalización y extracción de características son particularmente relevantes. La normalización tiene como objetivo transformar y limpiar los textos para garantizar que los datos estén en el formato adecuado para la extracción de información relevante. Una vez normalizados los textos, se aplican técnicas o algoritmos de extracción de características, conocidos como modelos de lenguaje, que permiten convertir los textos en representaciones numéricas que describen su contenido.

El ML juega un papel crucial en los sistemas de clasificación de textos, ya que permite el desarrollo de modelos que, a partir de algoritmos, identifican patrones en los datos y, basándose en esos patrones, realizan predicciones sobre datos nuevos. El proceso de creación de un modelo varía según el tipo de aprendizaje, que puede ser supervisado, no supervisado o semisupervisado, dependiendo de la disponibilidad de datos etiquetados. En el aprendizaje supervisado, los algoritmos pueden clasificarse en dos grandes grupos: algoritmos de clasificación y de regresión. La diferencia principal entre ellos radica en el tipo de valor que predicen: los algoritmos de clasificación trabajan con etiquetas categóricas, mientras que los algoritmos de regresión predicen valores continuos o reales.

Entre los algoritmos de clasificación más utilizados se encuentran K-NN, SVM y DT. Por otro lado, los algoritmos de regresión incluyen la LR y la SVR. Además, existen algoritmos avanzados tanto para clasificación como para regresión, que se basan en el enfoque de Boosting, un método que combina secuencialmente modelos débiles para mejorar la precisión. Algunos de los algoritmos de este tipo son LightGBM, regresor AdaBoost y regresor XGBoost.

Cuando se utilizan estos algoritmos, ya sean de clasificación o regresión, el objetivo es alcanzar el mejor rendimiento posible del modelo. Una técnica clave para lograr este objetivo es

la selección de características, la cual puede realizarse mediante algoritmos de ML que proporcionan información sobre la importancia de las características, o bien, a través de métodos estadísticos que eliminan características redundantes.

Para evaluar el rendimiento de un modelo, se emplean diversas métricas y técnicas que proporcionan información cuantitativa sobre los resultados obtenidos. Una métrica a destacar es la exactitud balanceada, que tiene en cuenta el desequilibrio en los conjuntos de datos utilizados para el entrenamiento. Esta métrica calcula el promedio de la recuperación correcta por clases, proporcionando una visión más equitativa del rendimiento en conjuntos de datos desbalanceados. Además de las métricas de evaluación, una técnica comúnmente utilizada es la validación cruzada k-fold. Esta técnica divide el conjunto de datos de entrenamiento en varios subconjuntos, y de manera iterativa, cada uno de estos subconjuntos actúa como conjunto de prueba mientras los restantes se usan para entrenar el modelo. Finalmente, se calcula un promedio de la métrica de evaluación obtenida, lo que permite una visión general del rendimiento del modelo.

Capítulo 3

Método propuesto

En este Capítulo se presenta una propuesta para la identificación de ansiedad a partir del análisis de textos cortos escritos por una persona mediante la implementación de algoritmos de aprendizaje automático, (ML, por sus siglas en inglés de *Machine Learning*). En el Capítulo 1 de esta tesis, se planteó la siguiente pregunta de investigación: *¿Es posible detectar la presencia de ansiedad en una persona mediante aprendizaje automático a partir del análisis de textos cortos?*. Por lo que a continuación se presenta una propuesta para realizar la detección de ansiedad y un diseño experimental que permitirá realizar la comparación entre los trabajos previos y la implementación de los enfoques de ML supervisado y semisupervisado.

3.1. Descripción general del método

La Figura 3.1 presenta un diagrama que detalla el método implementado en esta investigación, compuesto por siete etapas. En la primera etapa, se realiza la creación de la base de datos. La segunda etapa abarca el proceso de limpieza de datos, seguida por la tercera etapa, que corresponde a la separación de los datos. En la cuarta y quinta etapa, se desarrollan los pasos experimentales para los enfoques de aprendizaje supervisado y semisupervisado, respectivamente. La sexta etapa se enfoca en el entrenamiento de modelos de ML, y, finalmente, en la séptima etapa se lleva a cabo la evaluación de dichos modelos.

3.1.1. Creación de la Base de Datos

En esta sección se describe el proceso seguido para la creación de la base de datos empleada en las etapas posteriores. Dicho proceso se ilustra en la Figura 3.2, donde se presenta la secuencia de fases implementadas. Comenzando con la adquisición de los datos, seguida por la verificación de resultados por parte de un especialista en salud mental, y culminando con la creación del corpus.

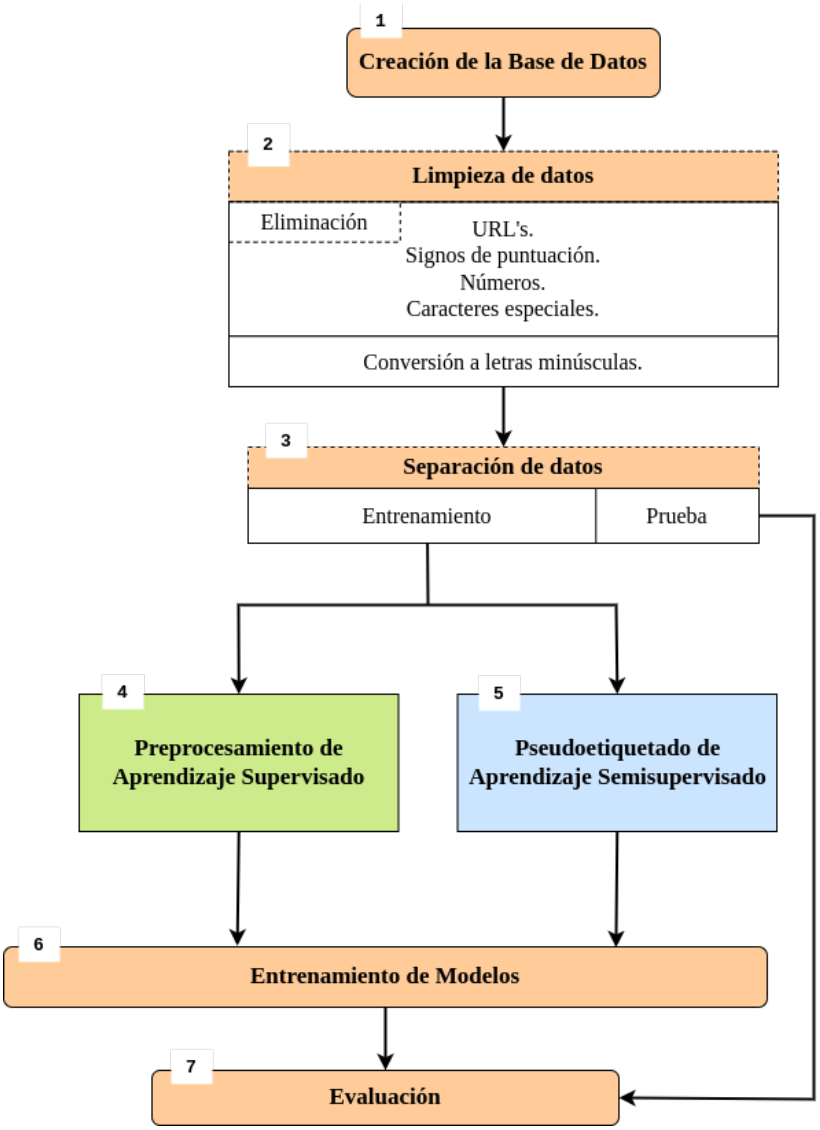


Figura 3.1: Diagrama general del método propuesto.

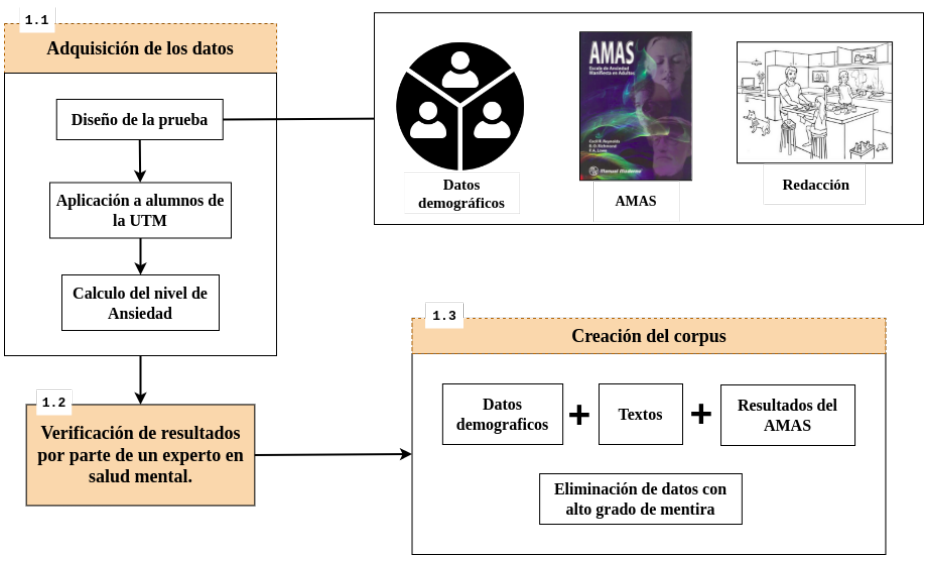


Figura 3.2: Creación de la base de datos.

Adquisición de los datos

El proceso comienza con la adquisición de los datos, cuyo primer paso consiste en el diseño de un instrumento para la recolección de textos cortos. Para este propósito se implementa un cuestionario, el cual consta de varias secciones: primero, preguntas para obtener datos demográficos de los participantes; luego, incluye los reactivos de la Escala de Ansiedad Manifiesta en Adultos (AMAS, por sus siglas en inglés de *Adult Manifest Anxiety Scale*) en su versión *C* (AMAS-C), adaptada para estudiantes (Reynolds et al., 2007), que permite evaluar el nivel de ansiedad de los encuestados. Finalmente, como último elemento del cuestionario, se solicita a los participantes redactar un texto en base a su percepción de una imagen tomada del trabajo de Tasnim et al. (2023). En la investigación de Tasnim et al. (2023), también pidieron a un grupo de participantes realizar una tarea similar; sin embargo, los datos recopilados para la detección de ansiedad consistieron en grabaciones de audio.

Una vez diseñado el cuestionario, se aplica a un grupo de alumnos de la Universidad Tecnológica de Mixteca (UTM). Tras la recolección de los datos, se procede al cálculo del nivel de ansiedad de cada participante mediante la automatización de la evaluación de acuerdo con las reglas autocalificables de AMAS-C, para obtener como resultado el estado emocional de los participantes en términos de cinco niveles de ansiedad: *Baja*, *Esperada*, *Elevación Leve*, *Clinicamente Significativa* y *Extrema*.

Revisión de resultados

Los resultados de la prueba calculados de forma automática se someten a una revisión por parte de un especialista en salud mental. Este profesional tiene la tarea de verificar y validar los niveles de ansiedad obtenidos por el script, evaluando la exactitud y coherencia de estas mediciones en comparación con los criterios establecidos por el AMAS-C. Mediante esta revisión, se aseguró que los valores calculados no solo reflejan de manera precisa el estado de ansiedad de los participantes, sino que también se mantienen alineados con los estándares clínicos y científicos de la evaluación de la ansiedad. Este proceso de validación es fundamental, ya que garantiza la fiabilidad de las mediciones obtenidas, permitiendo que los resultados sean utilizados con confianza en el análisis de la ansiedad en el contexto de la investigación.

Creación del corpus

El último paso de esta primera etapa corresponde a la creación del corpus. En un primer paso, se consideran los datos demográficos de los participantes, como edad, género, atención en salud metal y sí trabajan para solventar gastos de educación. En un segundo paso, se integran los textos redactados por los participantes junto con los resultados de la prueba AMAS-C. Constituyendo así la base de datos completa. Finalmente, en un último paso se lleva a cabo un

proceso de eliminación de datos con alto grado de mentira, con el fin de depurar el corpus y asegurar que solo se conserven datos válidos y confiables para el análisis posterior.

3.1.2. Limpieza de datos

En esta Etapa, se llevan a cabo acciones para depurar los datos de texto obtenidos en la fase previa, asegurando así su calidad y uniformidad para el análisis. El proceso de limpieza de datos se enfoca en la eliminación de elementos irrelevantes o que podrían introducir ruido en los resultados, tales como direcciones URL, signos de puntuación, números y caracteres especiales, los cuales no aportan información significativa para el análisis del contenido. Además, se convierten todos los textos a letras minúsculas, permitiendo así una estandarización homogénea de los datos y facilitando el procesamiento posterior, al evitar que palabras idénticas con distintas capitalizaciones sean tratadas como términos diferentes.

3.1.3. Separación de datos

La separación de los datos se realiza con el fin de generar un subconjunto destinado al entrenamiento del modelo y otro para su evaluación a través de pruebas. En este caso, se emplea una división estratificada con una proporción del 80 % de los datos asignados al entrenamiento y el 20 % restante a las pruebas. Cabe destacar que la estratificación garantiza que las clases en el conjunto de datos mantengan la misma proporción en ambos subconjuntos, asegurando que la distribución de las clases sea representativa en cada uno de ellos.

3.1.4. Preprocesamiento de aprendizaje supervisado

La Etapa 4 se centra en la implementación del aprendizaje supervisado, estructurándose en tres métodos distintos de preprocesamiento. Cada uno de estos métodos se aplica de manera independiente, ejecutando acciones específicas que permiten optimizar la calidad de los datos. Este enfoque organizado en métodos individuales facilita la evaluación comparativa de las distintas técnicas de preprocesamiento así como identificar cuál proporciona los resultados más robustos y precisos.

3.1.5. Pseudoetiquetado de aprendizaje semisupervisado

La Etapa 5 corresponde a los pasos seguidos durante el proceso de aprendizaje semisupervisado. En esta etapa, se destacan dos conjuntos de datos de entrenamiento: el conjunto A, que corresponde a los datos etiquetados, y el conjunto B, al cual se le eliminan etiquetas. La Tabla 3.1 detalla los conjuntos de datos empleados en cada experimento. El corpus construido en este trabajo de tesis se identifica como UTMente-Ansiedad.

Experimento	Datos de Entrenamiento A	Datos de Entrenamiento B
1	SocialMedia-Anxiety	UTMente-Ansiedad-Entrenamiento etiquetado B
2	UTMente-Ansiedad-Entrenamiento etiquetado B	SocialMedia-Anxiety
3	UTMenteII-Ansiedad	UTMente-Ansiedad-Entrenamiento etiquetado B

Tabla 3.1: Experimentos en aprendizaje semisupervisado.

3.1.6. Entrenamiento de modelos

En la Etapa 6 se realiza el entrenamiento de los modelos de ML. De acuerdo con las etapas previas, en este punto los datos están debidamente preparados para ser procesados por los algoritmos correspondientes, lo que permite el desarrollo de los modelos. En primer lugar, dentro del proceso de aprendizaje supervisado, se contempla el uso de los siguientes algoritmos:

- K Vecinos más cercanos (K-NN).
- Máquina de Soporte Vectorial (SVM).
- Árboles de decisión (DT).
- Bosque Aleatorio (RF).
- Bayes Ingenuo Multinacional (NB).
- LightGBM.
- Perceptrón Multicapa (MLP).

Por otro lado, para el proceso de aprendizaje semisupervisado, los algoritmos empleados en el entrenamiento de los modelos son los siguientes:

- Máquina de Soporte Vectorial (SVM).
- Bayes Ingenuo Multinomial (NB).
- LightGBM.

Otro aspecto crucial en esta etapa es la validación de los modelos generados a partir de los distintos algoritmos empleados. Para ello, se utiliza la técnica de validación cruzada con k-fold con un valor de $k = 5$, permitiendo evaluar el rendimiento del modelo mediante el calculo de la exactitud balanceada. Además, se obtiene un intervalo de confianza mediante la repetición del proceso de validación cruzada en cinco iteraciones. Este enfoque tiene como objetivo establecer expectativas claras sobre el rendimiento del modelo y obtener una comprensión precisa de la fiabilidad de los resultados, garantizando así una evaluación rigurosa y fundamentada.

3.1.7. Evaluación

Por último, en la Etapa 8 se lleva a cabo la evaluación de los modelos desarrollados en la etapa anterior, empleando el conjunto de datos de prueba definido en la Etapa 3. Esta evaluación tiene como objetivo medir el rendimiento de cada modelo en condiciones independientes de entrenamiento, permitiendo así una evaluación confiable de su capacidad predictiva mediante la métrica de exactitud balanceada, que proporciona un indicador claro de la efectividad general de cada modelo.

3.2. Descripción de preprocesamiento para el Aprendizaje Supervisado

En esta sección se describe en detalle la Etapa 4 del método, ilustrada en la Figura 3.3, que presenta los pasos implementados en cada uno de los métodos de preprocesamiento aplicados en el enfoque de aprendizaje supervisado. Esta figura permite visualizar el flujo de trabajo específico seguido en cada método, destacando las variaciones y decisiones adoptadas en cada caso. A continuación, se presenta una descripción de cada uno de estos métodos.

3.2.1. Método de preprocesamiento I

El primer método de preprocesamiento consiste en la eliminación de las palabras vacías, con excepción de aquellas que expresan negatividad, basado en el trabajo de Nova (2023), esto debido a que al escribir de manera negativa puede estar asociado con la presencia de ansiedad en una persona. Posteriormente, se lleva a cabo la extracción de características mediante el uso de los métodos de Bolsa de Palabras (BoW, por sus siglas en inglés de *Bag Of Words*), N-Gramas (Bigramas y Trigramas) y TF-IDF (por sus siglas en inglés de *Term Frequency - Inverse Document Frequency*).

3.2.2. Método de preprocesamiento II

El segundo método de preprocesamiento comienza con la extracción de características mediante los métodos de BoW, N-Gramas (Bigramas y Trigramas) y TF-IDF. En seguida, se lleva a cabo una selección de características utilizando algoritmos basados en árboles de decisión (DT, por sus siglas en inglés de *Decision Tree*), como bosque aleatorio (RF, por sus siglas en inglés de *Random Forest*) y árboles extremadamente aleatorios (ExtraTrees, por sus abreviatura en inglés de *Extremely Randomized Trees*), complementada con el método estadístico ANOVA (por su abreviatura en inglés de *Analysis of Variance*).

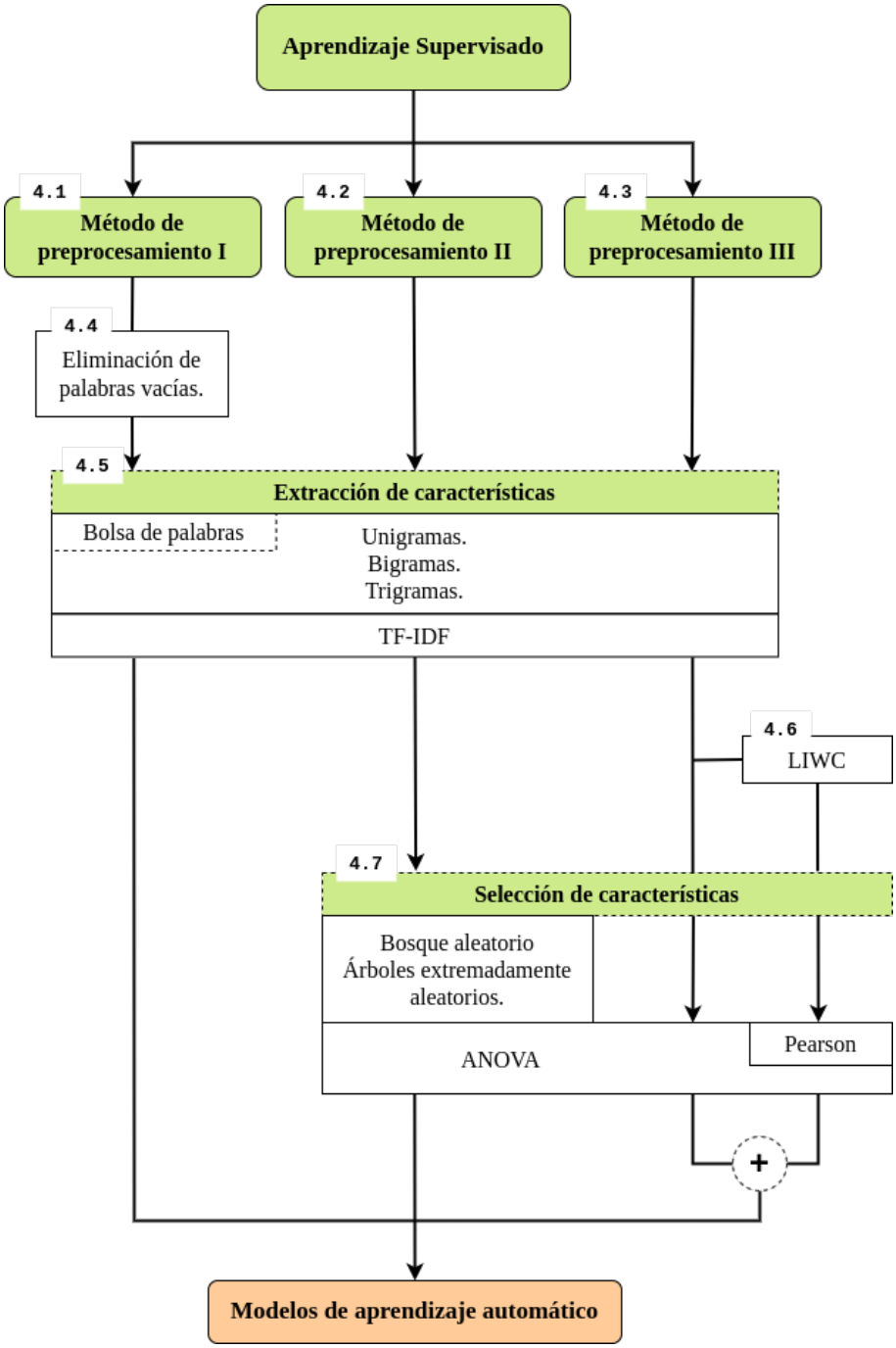


Figura 3.3: Preprocesamiento para aprendizaje supervisado.

3.2.3. Método de preprocesamiento III

El tercer método de preprocesamiento inicia con la extracción de características utilizando las técnicas de BoW, N-Gramas (bigramas y trigramas) y TF-IDF, cuyas salidas se someten a un proceso de selección mediante el método ANOVA. Adicionalmente, se emplea el diccionario LIWC (por sus siglas en inglés de *Linguistic Inquiry and Word Count*) para extraer otro conjunto de características. Estas últimas pasan por un proceso de selección que se realiza en dos etapas: primero, se calcula la correlación de Pearson entre todas las características para descartar aquellas que se correlacionaran con un valor mayor a 0.4 y con respecto a más de 3 características diferentes, con el objetivo de eliminar información redundante. En la segunda etapa, las características resultantes de este filtrado son sometidas nuevamente a selección utilizando el método ANOVA. Finalmente, el conjunto de características seleccionado incluye las obtenidas tras ambos procesos.

3.3. Descripción de pseudoetiquetado para el Aprendizaje Semisupervisado

En la presente sección se detallan los pasos correspondientes a la Etapa 5 del método, los cuales se ilustran en la Figura 3.4. Esta Figura muestra los pasos realizados durante los experimentos desarrollados bajo el enfoque de aprendizaje semisupervisado, proporcionando una representación clara del flujo de trabajo seguido en cada uno de ellos.

- **Extracción de características:** corresponde al proceso de obtención de características mediante los métodos de BoW, N-Gramas (Bigramas y Trigramas) y TF-IDF, aplicados al conjunto de datos *A*. Es importante señalar que las características extraídas de este conjunto se replican de manera equivalente en el conjunto de datos *B*, con el propósito de asegurar la compatibilidad de la información entre ambos conjuntos.

- **Selección de características:** este paso se realiza utilizando el método ANOVA aplicado al conjunto de datos *A*. No obstante, con el fin de asegurar la compatibilidad informativa con el conjunto de datos *B*, se emplean las mismas características seleccionadas en *A*.

- **Entrenamiento base:** consiste en tomar las características seleccionadas del conjunto de datos *A* e implementar el entrenamiento de modelos de aprendizaje supervisado a través los siguientes algoritmos.

  - Máquina de Soporte Vectorial (SVM).
  - Bayes Ingenuo Multinomial (NB).
  - LightGBM.

- **Eliminación de etiquetas:** se lleva a cabo en el conjunto de datos *B*, permitiendo obtener exclusivamente las características al separar los datos de sus correspondientes etiquetas.

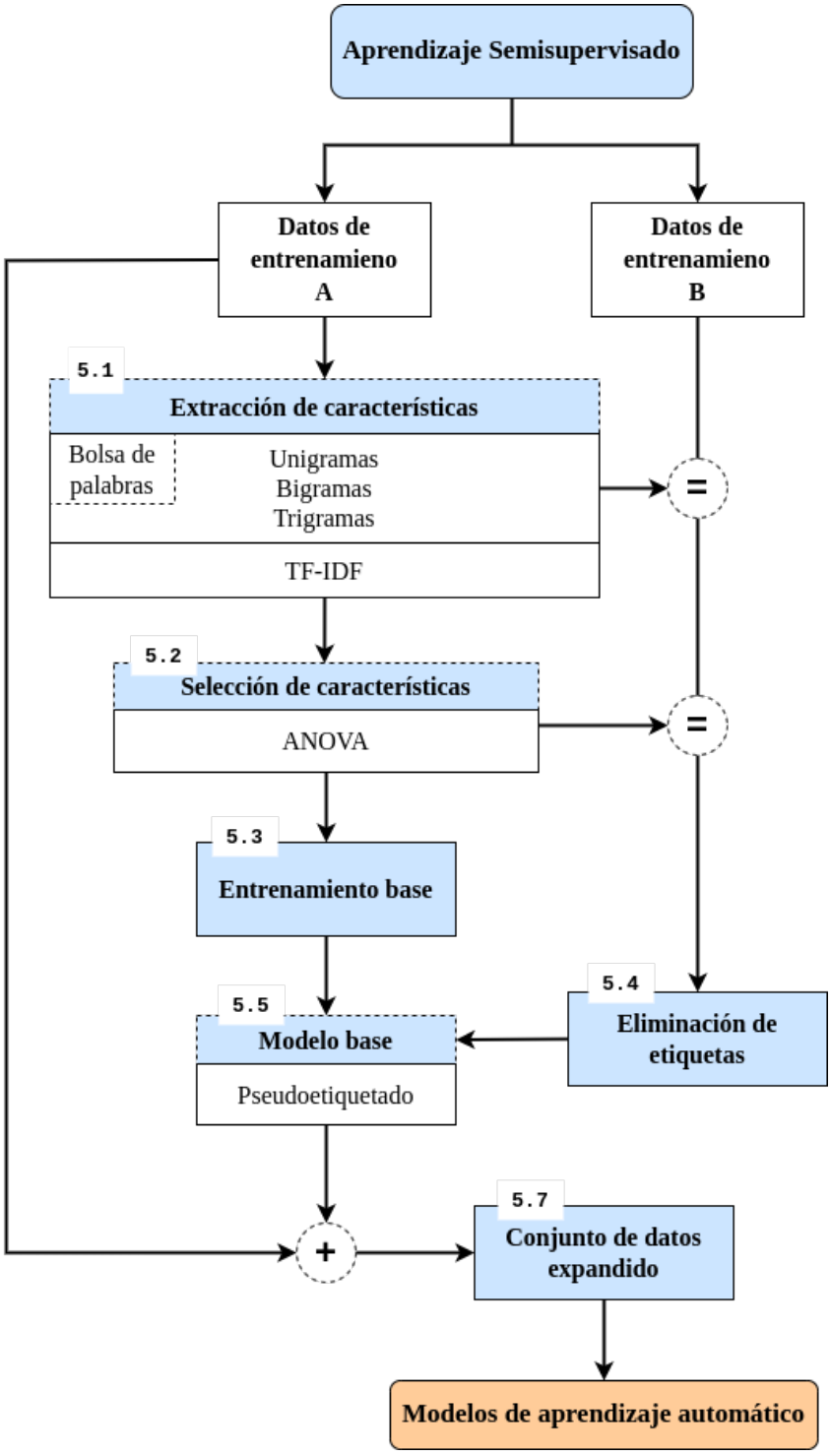


Figura 3.4: Pseudoetiquetado para aprendizaje semisupervisado.

- **Modelo base:** se desarrolla a partir del entrenamiento base, se emplea para llevar a cabo el proceso de pseudoetiquetado de los datos del conjunto *B* obtenido de la eliminación de etiquetas. El pseudoetiquetado consiste en asignar a los datos no etiquetados, las etiquetas obtenidas por el modelo base al hacer las predicciones, considerándolas como verdad fundamental.

- **Conjunto de datos expandido:** se conforma al integrar los datos del conjunto *A*, utilizados en el entrenamiento base, junto con los datos pseudoetiquetados por el modelo base, constituyendo ambos en un único conjunto. Esto permite que el conjunto final sea más robusto y se utilice para entrenar un nuevo modelo.

3.4. Resumen

El método propuesto en este capítulo se organiza en una serie de etapas, cada una de las cuales desempeña un papel fundamental dentro del proceso. A continuación, se presenta un resumen de las etapas que componen este método.

La Etapa 1 se enfoca en la adquisición de datos mediante el diseño y la aplicación de un cuestionario, el cual se emplea como herramienta para recolectar información de un grupo de participantes. Al finalizar este proceso, se obtiene un corpus etiquetado, que servirá como insumo clave para el análisis posterior.

La Etapa 2 se centra en la normalización de los datos, abarcando diversas acciones destinadas a asegurar que los textos se encuentren en un formato uniforme y adecuado para su posterior procesamiento. Entre estas acciones se incluyen la eliminación de elementos como URL, signos de puntuación, números y caracteres especiales, así como la conversión de todo el contenido textual a letras minúsculas.

La Etapa 3 implica la separación de los datos en subconjuntos destinados al entrenamiento y la prueba del modelo. Esta división se realiza de manera estratificada, asegurando una distribución equilibrada de las clases en ambos subconjuntos. El 80 % de los datos se asigna al conjunto de entrenamiento, mientras que el 20 % restante se reserva para la fase de prueba.

La Etapa 4 detalla los pasos involucrados en la ejecución de los tres métodos de preprocesamiento utilizados en el aprendizaje supervisado. Cada uno de estos procesos se desarrolla en pasos secuenciales y garantizan que los datos estén debidamente preparados antes de su uso en los modelos de ML.

La Etapa 5 describe los pasos del proceso que se lleva a cabo en los tres experimentos de

aprendizaje semisupervisado. Estos experimentos se distinguen por los conjuntos de datos utilizados en las fases de entrenamiento base y pseudoetiquetado.

La Etapa 6 se enfoca en el uso de algoritmos de ML para el entrenamiento de modelos. En esta etapa, se realiza la validación de los modelos mediante el método de validación cruzada con k-fold y se calcula el promedio de la exactitud balanceada obtenida en todas las iteraciones, lo que permite una evaluación más robusta del rendimiento del modelo.

Finalmente, en la Etapa 7 se retoma el conjunto de datos de prueba para evaluar el rendimiento de los modelos calculando el valor de la exactitud balanceada.

Capítulo 4

Resultados

En este capítulo se presentan los resultados de la experimentación al aplicar los métodos de preprocesamiento del estado del arte y del método de preprocesamiento propuesto en el capítulo anterior, tanto para el enfoque de aprendizaje supervisado como semisupervisado en el conjunto de datos denominado como UTMente-Ansiedad y para el conjunto de datos Social Media-Anxiety (Saha, 2022).

Es importante indicar que la hipótesis planteada al inicio de esta tesis asevera que: *"Las características léxicas de los textos cortos escritos por una persona tienen influencia para detectar la ansiedad, tanto en algoritmos de aprendizaje supervisado como en algoritmos de aprendizaje semisupervisado"*. En este capítulo se muestran los resultados y comparación de la evaluación de distintos modelos de aprendizaje automático (ML, por sus siglas en inglés de *Machine Learning*) después de aplicar el método propuesto en el capítulo anterior, donde se plantean tres diferentes estrategias de preprocesamiento para extraer las características léxicas de los textos, para realizar la detección de la ansiedad en textos cortos. Los textos utilizados fueron obtenidos de alumnos que participaron en una encuesta durante el curso propedéutico impartido por la Universidad Tecnológica de la Mixteca (UTM) en el periodo de agosto a septiembre de 2023.

La experimentación se realizó utilizando el lenguaje Python 3.12.0 y con la biblioteca especializada en algoritmos de ML Scikit-Learn 1.3.1, así como el diccionario LIWC (por sus siglas en inglés de *Linguistic Inquiry and Word Count*). El equipo de cómputo utilizado tenía las siguientes características: procesador Intel Core i5 y 24 GB de memoria RAM; en un sistema operativo Ubuntu 20.04.

4.1. Creación de la Base de Datos

En esta sección se describe la obtención y creación de la base de datos UTMente-Ansiedad, la cual está compuesta por textos cortos y por un valor de puntuación T del instrumento de autoevaluación AMAS (por sus siglas en inglés de *Adult Manifest Anxiety Scale*) en su versión

C (Reynolds et al., 2007). Aplicado a alumnos del nivel superior durante el curso propedéutico impartido por la Universidad Tecnológica de la Mixteca en el periodo de agosto a septiembre de 2023.

4.1.1. Obtención de datos

A partir del diseño y aplicación de un instrumento titulado: *Detección de rasgos de ansiedad en estudiantes* (ver Apéndice A para detalles del cuestionario), se construyó la base de datos. El instrumento fue aplicado en formato electrónico en Google Forms® y está compuesto de cuatro secciones. En la primera sección, se incluye un aviso de privacidad de los datos dirigido a los participantes, aunque en el diseño del instrumento no se recolecta información personal sensible o que identifique a los participantes. En la segunda sección, se solicita información demográfica: edad, género, si trabaja para financiar sus estudios; y por último si ha recibido algún diagnóstico o tratamiento relacionada con la ansiedad u otros trastornos mentales. La tercera sección está compuesta por cuarenta y nueve preguntas obtenidas de la prueba de autoevaluación conocida como AMAS en su versión *C* (AMAS-C), la cual permite medir el nivel de ansiedad presente en adultos. Finalmente, en la cuarta sección se presenta una imagen (ver Figura 4.1) donde se solicitó al participante la redacción de un texto corto de 1,300 caracteres, a partir de lo que le inspiró la imagen mostrada en la Figura 4.1.

Figura 4.1: Imagen utilizada en el instrumento: *Detección de rasgos de ansiedad en estudiantes*. Recuperada del trabajo de Tasnim et al. (2023)

El instrumento se aplicó a 443 estudiantes de la Universidad Tecnológica de la Mixteca, durante el curso propedéutico de todas las carreras en el mes de septiembre de 2023. Una vez aplicado el instrumento, se realizó una evaluación automática de las respuesta de cada participante siguiendo las instrucciones de la *Forma Autocalificable AMAS-C* (Reynolds et al., 2007), la cual representa una guía con los pasos correspondientes para obtener una puntuación

T asociada a un diagnóstico de ansiedad y se puede consultar en la página 68 del AMAS-C (Reynolds et al., 2007). El proceso comienza con la suma de las respuestas afirmativas de los participantes de acuerdo a seis parámetros considerados: *Inquietud/hipersensibilidad*, *Ansiedad fisiológica*, *Ansiedad para los exámenes*, *Preocupación/estrés social* y *Mentira*. estos resultados según AMAS-C se denominan puntuaciones naturales y están asociadas a una puntuación T. Estos pasos se programaron en un script el cual se puede ver a detalle en el Apéndice C. Para validar los resultados obtenidos, la Licenciada en Psicología y Maestra en Educación Denisse Millán Hernández realizó de manera manual una revisión y evaluación de una submuestra aleatoria compuesta de 120 cuestionarios. El resultado de este procedimiento permitió verificar la eficacia del programa computacional para la evaluación automática ya que no se encontraron discrepancias entre los resultados obtenidos por la psicóloga y el programa computacional para la evaluación automática.

En este proceso de evaluación, se descartaron 76 aplicaciones, ya que de acuerdo a la hoja de perfil de la *Forma Autocalificable AMAS-C*, en la puntuación obtenida, es necesario cumplir una puntuación mínima de T en la sección descrita como mentira. El manual AMAS-C especifica que dicho rubro está diseñado para identificar la distorsión intencional o el sesgo de respuesta del participante. Por lo que, debido a los resultados obtenidos de estos participantes se determinó que fueran descartados para evitar una tendencia errónea en los registros de la base de datos para la detección de ansiedad.

Una vez validados los resultados de la evaluación de AMAS-C generados de manera automática, los valores de T obtenidos fueron utilizados para realizar el etiquetado de los textos de cada participante y crear el conjunto de datos UTMente-Anxiedad. Para este paso se propusieron dos tipos de criterios:

- **Etiquetado de Tipo A**. En este etiquetado se utiliza el valor de la puntuación T calculado en la *Forma Autocalificable AMAS-C*. De acuerdo con la Tabla 4.1 obtenida en (Reynolds et al., 2007), se obtienen cinco clases para definir la ansiedad: Extrema, Clínicamente significativa, Elevación leve, Esperada y Baja.

- **Etiquetado de Tipo B**. En este etiquetado, y de acuerdo con las recomendaciones realizadas por la Licenciada en Psicóloga y Maestra en Educación Denisse Millán Hernández, se utiliza el valor de puntuación T calculado en la *Forma Autocalificable AMAS-C* para clasificar como con *"ansiedad"* los textos con puntuaciones de T mayores a 65 y los textos con puntuaciones de T menores a tal valor se clasifican como *"sin ansiedad"*.

[3]Extracto de la tabla de Interpretación de las puntuaciones T de las escalas de ansiedad del AMAS-C (Reynolds et al., 2007).

Puntuación T	Categoría descriptiva
≥ 75	Extrema
$65 - 74$	Clínicamente significativa
$55 - 64$	Elevación leve
$45 - 54$	Esperada
≤ 44	Baja

Tabla 4.1: Interpretación de las puntuaciones T de Ansiedad Total.[3]

4.1.2. Análisis exploratorio de datos de UTMente-Ansiedad

Después de la obtención de los datos y del cálculo de la puntuación T obtenida para cada participante, se realiza un análisis exploratorio de datos con la finalidad de conocer algunos aspectos de los participantes y del conjunto de datos obtenido.

No. de columna	Nombre	Descripción	Tipos de datos
1	Marca temporal	Fecha y hora de la aplicación de la encuesta	Fecha
2	Edad	Edad del participante	Numérico
3	Género	Variable categórica con los valores: hombre, mujer u otro	Texto
4	Diagnóstico	Indica si el participante ha recibido algún diagnostico o tratamiento previo relacionado con la ansiedad u otros trastornos mentales	Booleano
5	Trabajo	Indica se trabaja actualmente para financiar estudios o gastos educativos	Booleano
6 a la 54	AMAS-C_RP#	Respuesta a las preguntas de la 1 a la 49 del AMAS-C	Texto
55	Descripción	Descripción de la imagen de la Figura 4.1	Texto
56	Puntuación T	Valor obtenido por el evaluador automático	Numérico
57	Nivel de ansiedad	Etiquetado tipo A. Valor categórico ordinal: Baja: 0; Clínicamente significativa: 1; Elevación leve: 2; Esperada: 3; Extrema: 4	Numérico
58	Ansiedad	Etiquetado de tipo B. Indica si tiene ansiedad o no	Booleano

Tabla 4.2: Base de datos UTMente-Ansiedad.

Los datos contienen 367 registros y 58 columnas. Dado que el conjunto de datos fue obtenido con un formulario de Google® se validó que el usuario no dejará secciones o preguntas

sin contestar, por lo tanto, no se obtuvieron registros con información nula o incompleta. Las columnas de los datos se muestran en la Tabla 4.2. Donde las columnas del 2 al 5 representan variables demográficas. En cuanto al género, participaron 236 hombres, 127 mujeres y 4 que indicaron su género como: "otro". La edad media es de 18 años, con una edad máxima de 24 y una mínima de 17 años.

Al considerar los niveles de ansiedad de AMAS-C (etiquetado de tipo A) se especifican cinco niveles de ansiedad: *Baja*, *Esperada*, *Elevación leve*, *Clínicamente significativa* y *Extrema*. En la Figura 4.2, se muestra la frecuencia de los tipos de ansiedad en la base de datos. Los niveles de ansiedad con mayor frecuencia son: *Elevación leve* y *Clínicamente significativa*, con números de 104 y 101 diagnósticos, respectivamente. Para ansiedad *Extrema* se presenta en 36 casos del total de los 367 ejemplos que conforman el conjunto de datos. Estas proporciones de datos cumplen con las características de un conjunto de datos desbalanceado al no existir proporciones similares de número de ejemplos entre los cinco niveles de ansiedad.

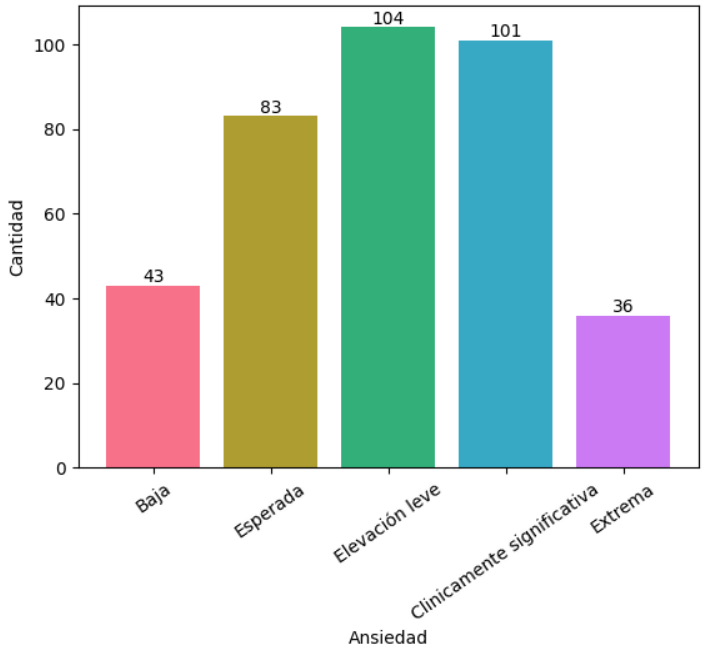


Figura 4.2: Frecuencia de niveles de ansiedad AMAS-C en el conjunto de datos UTMente-Ansiedad.

En la Figura 4.3 se muestra la relación entre el género y los niveles de ansiedad para el conjunto de datos UTMente-Ansiedad, de acuerdo al género de los participantes las mujeres tienen una participación de 50 y 21 en los niveles de ansiedad *Clínicamente significativa* y *Extrema*, respectivamente. A diferencia de los hombres, que presentan menor frecuencia en estos niveles de ansiedad. En los niveles restantes los hombres son los que representan la proporción mayor.

En cuanto a los participantes que indicaron haber recibido algún diagnóstico o tratamiento previo relacionado con la ansiedad u otros trastornos mentales, se muestra en todos los niveles

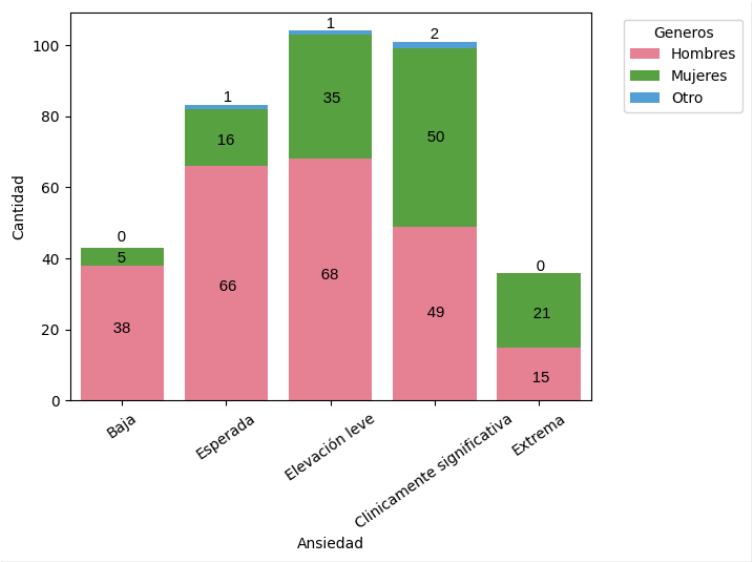


Figura 4.3: Relación del género y los niveles de ansiedad AMAS-C en el conjunto de datos UTMente-Ansiedad.

de ansiedad del conjunto de datos UTMente-Ansiedad que la proporción mayor son aquellos participantes que no han recibido algún diagnóstico en comparación de aquellos que sí han recibido un diagnóstico, incluso en el nivel de ansiedad *extrema* (ver Figura 4.4).

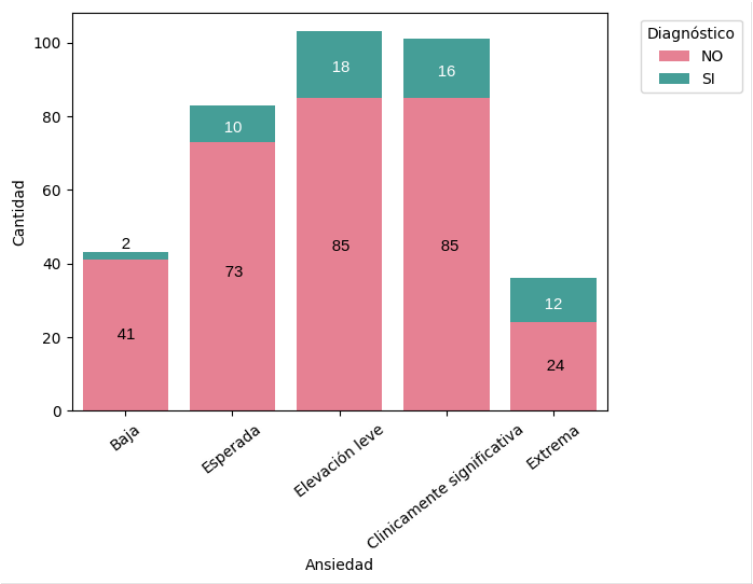


Figura 4.4: Relación de diagnóstico previo y niveles de ansiedad AMAS-C en el conjunto de datos UTMente-Ansiedad.

En cuanto al análisis de los participantes que indicaron que trabajan para financiar estudios o gastos educativos, se muestra en todos los niveles de ansiedad del conjunto de datos UTMente-Ansiedad, que la proporción mayor son aquellos participantes que no trabajan en relación a los que indicaron que si tiene alguna actividad laboral (ver Figura 4.5) .

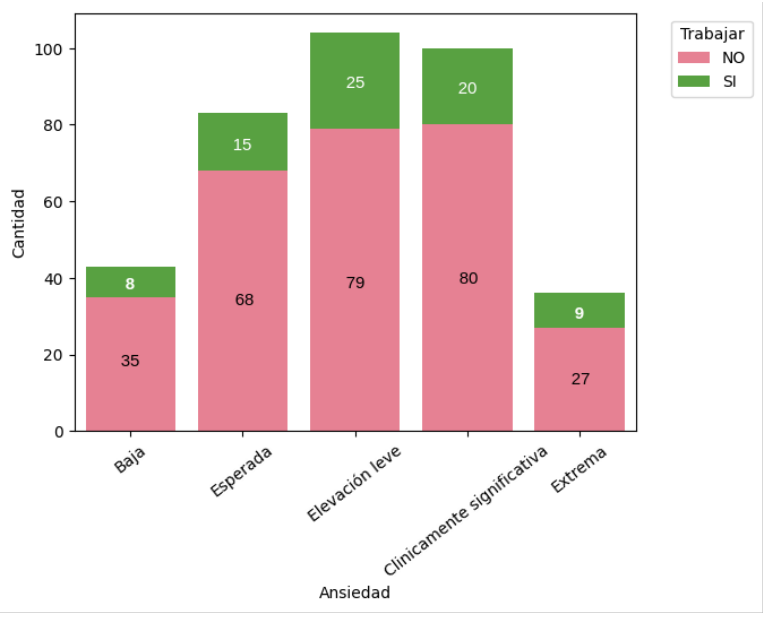


Figura 4.5: Relación de la respuesta si trabaja actualmente para financiar estudios o gastos educativos frente a los niveles de ansiedad AMAS-C en el conjunto de datos UTMente-Ansiedad.

Por último, al considerar los niveles de ansiedad con el etiquetado de tipo B donde solo se define la presencia o ausencia de ansiedad. En la Figura 4.6, se muestra la frecuencia de los tipos de ansiedad en la base de datos.

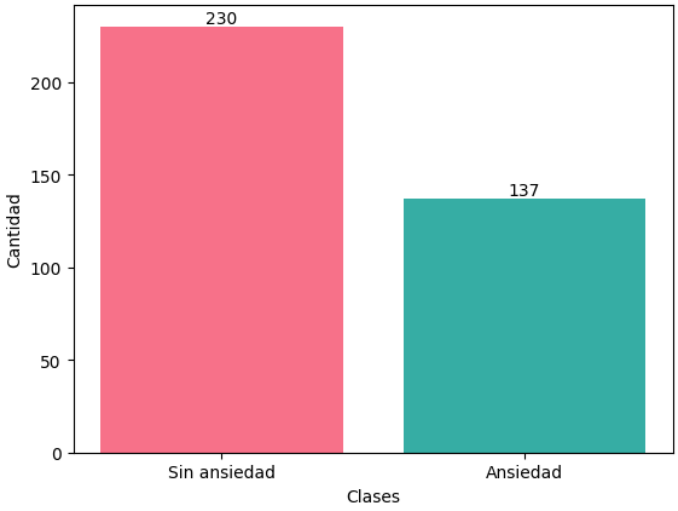


Figura 4.6: Clases binarias.

La información demográfica fue utilizada para describir la pluralidad de los alumnos participantes. Por lo que, este análisis muestra una consistencia entre los resultados obtenidos y los esperados de acuerdo a la muestra de participantes que conforman el conjunto de datos UTMente-Ansiedad.

Con respecto a los textos cortos generados por cada participante, también se realizó un

análisis exploratorio de los datos de texto. El corpus muestra una riqueza léxica de 0.059 y 6,873 palabras únicas, con una longitud promedio del texto de 1,609.52 caracteres y con 139.23 palabras únicas en promedio por cada participante. En la Figura 4.7 se muestra a través de una nube de palabras o *wordcloud* las palabras utilizadas en el corpus, donde el tamaño de cada palabra refleja la frecuencia de uso en el corpus.

Figura 4.7: Nube de palabra de UTMente-Ansiedad.

Texto	Ansiedad
Primeramente se puede observar a una familia de 3 integrantes, con una mascota en la parte de su cocina integral, así como igual se aprecia un angulo de su pequeña sala que tiene donde se logran ver algunos electrodomésticos como son la televisión donde se puede apreciar como es la convivencia a la hora de preparar los alimentos...	Baja
La madre esta haciendo la comida para su familia, y su esposo le esta ayudando a cocinar partiendo las cebollas y rebanando aparte verduras en cambio su hija lo que esta haciendo es que esta jugando las verduras y a su vez esta leyéndoles algún trabajó que hizo lo mas seguro en su escuela...	Esperada
En la imagen podemos ver que es una familia con cierto desorden a su alrededor, cosas que no van en su lugar, el papá o al menos el hombre de la imagen se nota estresado, la mamá viendo que pasa y haciendo lo suyo en la cocina, la niña se nota alegre y ayudando a ordenar el desastre que hay en la cocina. Hay botellas en el bote de basura y una rota, probablemente existen problemas...	Elevación Leve
En la cocina es un desastre la mujer esta cocinando y no se da cuenta que las cosas de arriba se van a caer y pueden hacer que el material caliente de la olla le salpique y la queme, mientras la llave de fregadero esta abierta lo que genera que se este desperdiciando el agua...	Clínicamente significativa
Son una familia con bastante desorden en general, hay un amplio desorden dentro de la habitación, también bastante estrés de parte del padre debido a que su hija no esta siendo ordenada y esta tomando cosas que no debería tomar, el padre esta frustrado mientras que la madre trabaja haciendo la comida y su mascota tiene una cuchara que se utilizaría en la cocina, la madre muestra una sonrisa ante todo esto pero no parece ser feliz...	Extrema

Tabla 4.3: Ejemplos de textos en UTMente-Ansiedad.

En la Tabla 4.3 se muestran algunos fragmentos de ejemplo de los textos que conforman el conjunto de datos UTMente-Ansiedad.

En cuanto a la frecuencia de las palabras se observa en la Figura 4.8 los Unigramas más comunes en el corpus UTMente-Ansiedad. Palabras como *familia*, *niña*, *cocina*, *perro*, *padre*, *comida*, entre otras; muestran una relación entre lo que la persona escribió y la imagen incluida en el cuestionario (ver Figura 4.1).

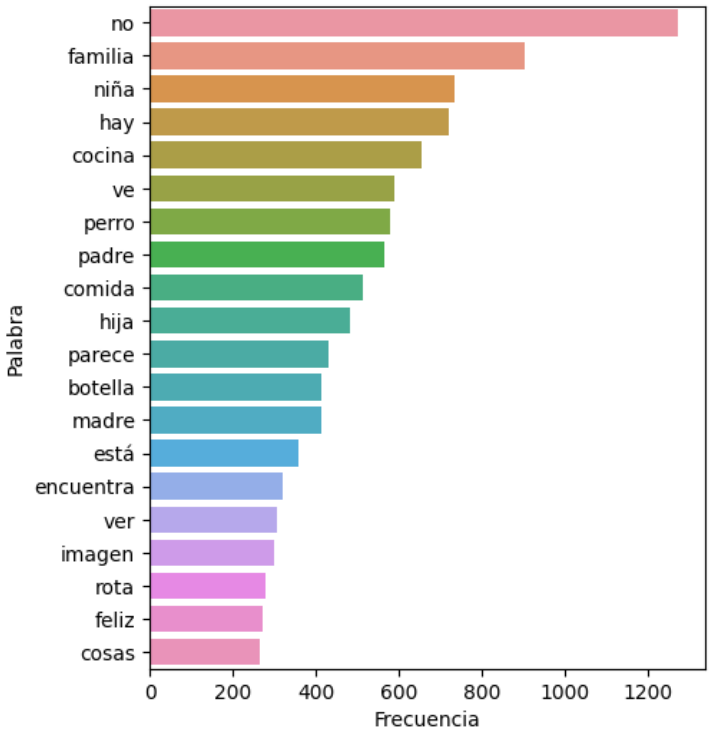


Figura 4.8: Top 20 de la frecuencia de Unigramas en UTMente-Ansiedad.

Adicionalmente, se realizó una gráfica con las frecuencias de los Bigramas más comunes en el corpus UTMente-Ansiedad, destacando los Bigramas: *{botella, rota}*, *{hay, botella}* y *{botella, vidrio}* (ver Figura 4.9).

El uso de Trigramas es una forma común de conocer la relación que se guarda entre palabras dentro del texto, por lo que en la Figura 4.10 se muestran el top 20 de las frecuencias más comunes, donde: *{hay, botella, rota}*, *{botella, vidrio, rota}* y *{hay, botella, vidrio}* son las más frecuentes en UTMente-Ansiedad.

4.1.3. Separación de datos de Entrenamiento y Prueba en UTMente-Ansiedad

El conjunto de datos UTMente-Ansiedad fue separado en datos de entrenamiento y prueba con una proporción de: 80 % como UTMente-Ansiedad-Entrenamiento y 20 % como UTMente-

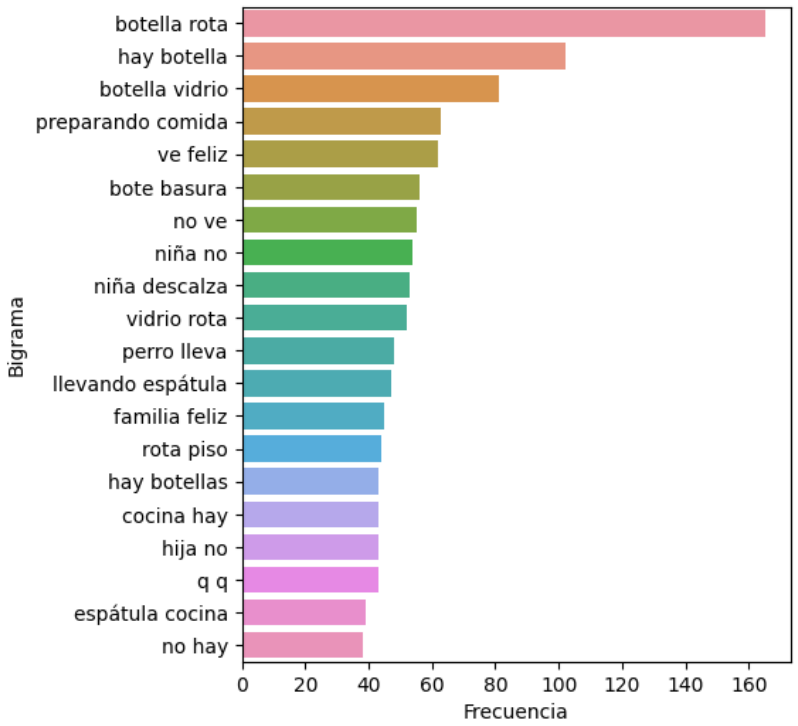


Figura 4.9: Top 20 de la frecuencia de Bigramas en UTMente-Ansiedad.

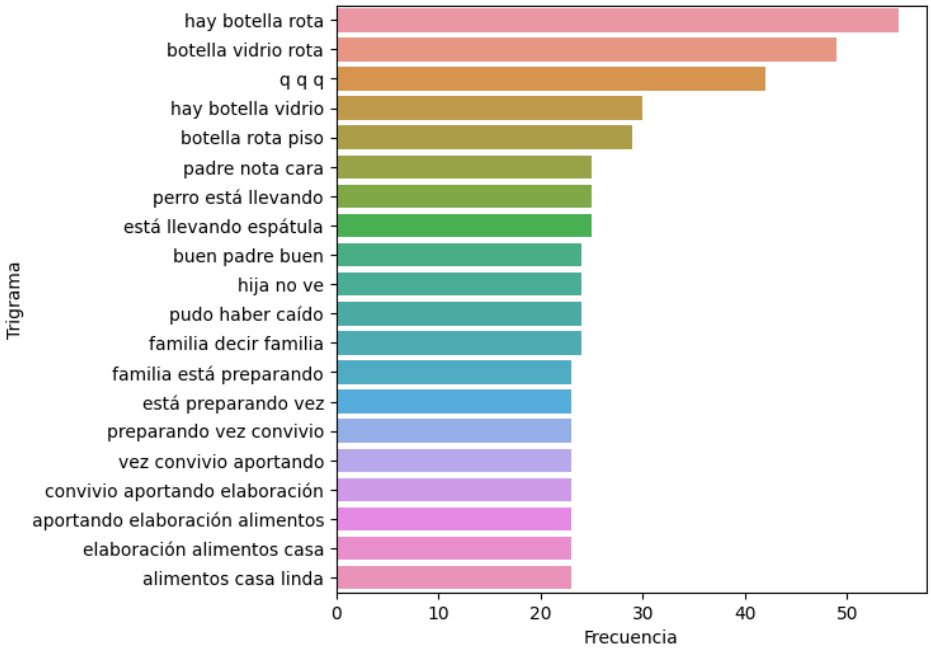


Figura 4.10: Top 20 de la frecuencia de Trigramas en UTMente-Ansiedad.

Ansieada-Prueba. La distribución de los datos en ambos subconjuntos muestran un desbalance (ver Tabla 4.4), es decir, se observa que existen más ejemplos de la clase *Elevación leve* y de la clase *Clínicamente significativa*. Por lo que es necesario considerar este aspecto al momento de analizar los resultados de evaluación de los modelos obtenidos en las siguientes secciones.

Clase	UTMente-Ansiedad-Entrenamiento	UTMente-Ansiedad-Prueba
Baja	34	9
Esperada	66	17
Elevación leve	83	21
Clínicamente significativa	81	20
Extrema	29	7

Tabla 4.4: Distribución de tipo de ansiedad en el conjunto UTMente-Ansiedad.

4.2. Métodos supervisados de clasificación en UTMente-Ansiedad con etiqueta Tipo A

En la experimentación de esta sección, se realizó el entrenamiento de distintos modelos a partir de siete algoritmos de ML (K-Vecinos más cercanos (K-NN), Bosque Aleatorio (RF), Máquina de Soporte Vectorial (SVM), Árboles de Decisión (DT), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM), Naive Bayes Multinomial (NB) y Perceptrón Multicapa (MLP)) para realizar la clasificación de ansiedad sobre el conjunto de datos UTMente-Entrenamiento con etiquetado Tipo A (clases: *Baja*, *Esperada*, *Elevación leve*, *Clínicamente significativa* y *Extrema*). Para el entrenamiento se implementaron métodos de preprocesamiento encontrados en trabajos previos y tres métodos propuestos, descritos en el capítulo anterior:

- Método de preprocesamiento I (Nova, 2023).
- Método de preprocesamiento I-BoWnG.
- Método de preprocesamiento II-RF (Byers et al., 2023).
- Método de preprocesamiento II-ET (Byers et al., 2023).
- Método de preprocesamiento II-ANOVA.
- Método de preprocesamiento III (Yu et al., 2023).
- Método de preprocesamiento III-BoW.

A continuación se muestran los resultados obtenidos, de acuerdo al método de preprocesamiento. Con el objetivo de evaluar adecuadamente el desempeño de los modelos se utiliza la validación cruzada con k-fold con un $k = 5$, cada fold tuvo una distribución aleatoria estratificada de las clases. Utilizando los resultados de los cinco experimentos se calcularon los intervalos de confianza y el promedio de la exactitud balanceada (Acc B) de cada modelo.

4.2.1. Método de preprocesamiento I con UTMente-Entrenamiento etiquetado Tipo A.

Mediante el método de preprocesamiento original propuesto en (Nova, 2023), se realiza una extracción de características con TF-IDF (por sus siglas en inglés de *Term Frequency - Inverse Document Fre quency*), dando un total de 5,874 características para entrenar los algoritmos de ML (K-Vecinos más cercanos (K-NN), Bosque Aleatorio (RF), Máquina de Soporte Vectorial (SVM), Árboles de Decisión (DT), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM), Naive Bayes Multinomial (NB) y Perceptrón Multicapa (MLP)) para la clasificación de ansiedad sobre el conjunto de datos UTMente-Entrenamiento con etiquetado Tipo A. A continuación se muestran los resultados sobre el conjunto de datos UTMente-Entrenamiento.

Método de preprocesamiento I-TFIDF con UTMente-Entrenamiento etiquetado Tipo A.

Al utilizar el método de preprocesamiento I con la extracción de características TF-IDF se entrenaron y probaron siete modelos de ML mediante una validación cruzada con k-fold, donde el valor de $k = 5$. En la Tabla 4.5 se muestran los resultados de la exactitud balanceada. El mejor modelo fue Árbol de Decisión (DT) con una exactitud balanceada de 52.21 % ($\pm$ 0.85) en el entrenamiento y 20.98 % ($\pm$ 1.33) en la prueba, sin embargo, la diferencia de desempeño en el entrenamiento y prueba muestra que el modelo presenta sobreajuste, en comparación con los modelos de Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM), Máquina de Soporte Vectorial (SVM) y Perceptrón Multicapa (MLP) que alcanzan una exactitud balanceada de 20.2 % ($\pm$ 0.66), 20.0 % ($\pm$ 0.00) y 20.0 % ($\pm$ 0.00) en la prueba, y un 35.86 % ($\pm$ 0.4), 24.62 % ($\pm$ 0.13) y 20.0 % ($\pm$ 0.00) en el entrenamiento, respectivamente, para cada modelo.

Modelo	**Entrenamiento**		**Prueba**	
	Acc B	IC	Acc B	IC
DT	**52.21**	**± 0.85**	**20.98**	**± 1.33**
LigthGBM	35.86	± 0.4	20.2	± 0.66
SVM	24.62	± 0.13	20.0	± 0.0
MLP	20.0	± 0.0	20.0	± 0.0
NB	29.35	± 0.25	19.62	± 0.21
RF	29.34	± 0.44	19.27	± 0.89
KNN	51.58	± 0.65	15.41	± 1.17

Tabla 4.5: Resultados de tipo de preprocesamiento I en UTMente-Entrenamiento etiquetado Tipo A.[4]

[4]Resultados de los siete modelos: K-Vecinos más cercanos (K-NN), Bosque Aleatorio (RF), Máquina de Soporte Vectorial (SVM), Árboles de Decisión (DT), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM), Naive Bayes Multinomial (NB) y Perceptrón Multicapa (MLP) con el promedio de la métrica de exactitud balanceada (Acc B) con una validación cruzada de 5 folds, sobre el conjunto de datos UTMente-Entrenamiento etiquetado Tipo A. El intervalo de Confianza (IC) se calculó repitiendo el experimento cinco veces.

Método de preprocesamiento I-BoWnG con UTMente-Entrenamiento etiquetado Tipo A.

De acuerdo a los modelos de lenguaje estudiados en el marco teórico es posible que la Bolsa de Palabras (BoW, por sus siglas en inglés de *Bag Of Words*) con N-Gramas (BoWnG) en sus variantes de Unigramas (BoW1G), Bigramas (BoW2G) y Trigramas (BoW3G) sean implementados. Por lo que, en este trabajo, se propuso como alternativa al modelo TF-IDF del método de preprocesamiento PI-BoWnG. En la Tabla 4.6 se muestra el resultado de evaluación del entrenamiento y prueba de una validación cruzada con k-fold con un valor de $k = 5$, donde se observa que el desempeño del mejor modelo de ML con el modelo de Bigramas y con Máquina de Soporte Vectorial (SVM) alcanzó una exactitud balanceada de 20.00 % ($\pm$ 0.0) y de 28.33 % ($\pm$ 0.0), en entrenamiento y prueba, respectivamente. Por lo que, el método de preprocesamiento PI-BoW1G mejora el desempeño de modelos, en particular de aquellos algoritmos que no se sobreajustan. Adicionalmente la exactitud balanceada del modelo SVM con PI-BoW1G supera al método de procesamiento I propuesto por Nova (2023).

Modelo / Modelo	**BoW1G**	**BoW2G**	**BoW3G**
SVM	**20.00** ($\pm$ **0.0**)	18.70 ($\pm$ 0.36)	40.02 ($\pm$ 0.60)
	28.33 ($\pm$ **0.0**)	17.31 ($\pm$ 0.53)	17.53 ($\pm$ 1.34)
K-NN	31.65 ($\pm$ 0.31)	34.21 ($\pm$ 3.18)	34.21 ($\pm$ 3.18)
	19.73 ($\pm$ 0.99)	19.75 ($\pm$ 0.47)	19.75 ($\pm$ 0.47)
RF	29.77 ($\pm$ 0.74)	30.13 ($\pm$ 0.71)	26.37 ($\pm$ 0.62)
	20.84 ($\pm$ 1.23)	20.11 ($\pm$ 0.82)	20.07 ($\pm$ 0.19)
DT	45.09 ($\pm$ 1.62)	36.36 ($\pm$ 0.84)	33.28 ($\pm$ 0.33)
	19.84 ($\pm$ 1.24)	19.55 ($\pm$ 1.14)	20.45 ($\pm$ 0.51)
LigthGBM	31.36 ($\pm$ 0.52)	30.10 ($\pm$ 0.71)	21.28 ($\pm$ 0.19)
	17.57 ($\pm$ 1.10)	21.24 ($\pm$ 1.20)	19.17 ($\pm$ 0.69)
NB	36.75 ($\pm$ 0.48)	95.03 ($\pm$ 0.15)	99.70 ($\pm$ 0.0)
	19.04 ($\pm$ 1.12)	18.87 ($\pm$ 1.19)	20.90 ($\pm$ 1.07)
MLP	20.00 ($\pm$ 0.0)	20.00 ($\pm$ 0.0)	20.00 ($\pm$ 0.0)
	20.00 ($\pm$ 0.0)	20.00 ($\pm$ 0.0)	20.00 ($\pm$ 0.0)

Tabla 4.6: Resultados de tipo de preprocesamiento I-BoWnG modificado en UTMente-Entrenamiento etiquetado Tipo A.[4]

4.2.2. Método de preprocesamiento II con UTMente-Entrenamiento etiquetado Tipo A.

Al utilizar el método de preprocesamiento II (Byers et al., 2023) se entrenaron y evaluaron nuevamente siete modelos de ML mediante una validación cruzada con k-fold, donde el valor de $k = 5$. Los algoritmos utilizados para la generación de los modelos son: K-Vecinos más cercanos (K-NN), Bosque Aleatorio (RF), Máquina de Soporte Vectorial (SVM), Árboles de Decisión (DT), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM), Naive Bayes Multinomial (NB) y Perceptrón Multicapa (MLP).

En este método de preprocesamiento se utiliza la extracción de características con el modelo de BoW con N-Gramas (BoWnG) en tres variantes: Unigramas (BoW1G), Bigramas (BoW2G) y Trigramas (BoW3G). En adición, se realiza una selección de características mediante los métodos: Árboles Extremadamente Aleatorios (ET) y Árboles Aleatorios (FR), que son métodos basados en árboles como en la propuesta de (Byers et al., 2023). Además, se implementa un método propuesto en esta tesis con ANOVA (por su abreviatura en inglés de *Analysis of Variance*) como una técnica adicional para la selección de características (PII-ANOVA). Para más detalles sobre la cantidad total y las características seleccionadas por estos métodos se puede consultar el Apéndice B.1.

Método de preprocesamiento II con selección de características basada en Árboles Aleatorios en UTMente-Entrenamiento etiquetado Tipo A.

En esta sección se muestran los resultados de la evaluación de los modelos obtenidos de los algoritmos de ML entrenados con BoW basada en N-Gramas (BoWnG), y con el método de selección de características con Árboles Aleatorios (PII-RF), permitiendo un total de de 2,030 en BoW1G, 11,646 en BoW2G y 22,047 en BoW3G características seleccionadas. El mejor resultado se obtuvo con Árboles Aleatorios (RF) con la técnica de extracción de características de BoW con Unigramas (BoW1G) con una exactitud balanceada de 55.35 % (± 2.0) en entrenamiento y de 30.36 % (± 1.49) en prueba (ver en la Tabla 4.7. El modelo de Máquina de Soporte Vectorial (SVM) y Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM) obtuvieron una exactitud balanceada de 28.33 % (± 0.00) y 20.00 % (± 1.44) en la prueba, sin embargo sus resultados en el entrenamiento (59.70 % y 40.81 %, respectivamente) muestran un sobreajuste.

Modelo / Modelo	**BoW1G**	**BoW2G**	**BoW3G**
RF	**55.35** (± **2.0**)	25.58 (± 0.30)	32.92 (± 0.62)
	30.36 (± **1.49**)	19.79 (± 1.09)	19.23 (± 0.30)
SVM	59.70 (± 0.0)	12.09 (± 0.67)	22.22 (± 0.52)
	28.33 (± 0.0)	19.99 (± 1.86)	20.69 (± 2.90)
LigthGBM	40.81 (± 0.55)	27.39 (± 0.38)	26.05 (± 0.40)
	20.0 (± 1.44)	18.79 (± 0.92)	19.04 (± 0.65)
DT	34.51 (± 0.82)	33.34 (± 0.67)	23.13 (± 0.29)
	20.49 (± 0.71)	19.75 (± 0.34)	20.47 (± 0.36)
MLP	20.0 (± 0.0)	20.0 (± 0.0)	20.06 (± 0.05)
	20.0 (± 0.0)	20.0 (± 0.0)	20.0 (± 0.0)
NB	40.93 (± 0.52)	40.36 (± 1.25)	32.06 (± 0.25)
	19.76 (± 0.94)	18.83 (± 0.76)	20.10 (± 0.11)
K-NN	46.87 (± 0.33)	31.65 (± 0.70)	71.95 (± 1.28)
	19.60 (± 0.53)	20.92 (± 1.44)	19.42 (± 0.32)

Tabla 4.7: Resultados del preprocesamiento PII-RF en UTMente-Entrenamiento etiquetado Tipo A.[5]

Para esta sección, el modelo de lenguaje TF-IDF es implementado como una alternativa en este método de preprocesamiento II, en donde se seleccionaron 20,030 características. En la Tabla 4.8 se muestra el resultado de evaluación del entrenamiento y prueba de una validación cruzada con k-fold con un valor $k = 5$. Se observa que el desempeño del mejor modelo de ML para el modelo de TF-IDF fue con Máquina de Soporte Vectorial (SVM) con una exactitud balanceada de 24.07 % ($\pm$ 0.23) y de 28.33 % ($\pm$ 0.0), en entrenamiento y prueba, respectivamente. Por lo que, los resultados obtenidos no superan a los modelos obtenidos con el método originalmente planteado por (Byers et al., 2023) (ver Tabla 4.7).

Modelo	**Entrenamiento**		**Prueba**	
	Acc B	IC	Acc B	IC
SVM	**24.07**	**$\pm$ 0.23**	**28.33**	**$\pm$ 0.0**
K-NN	34.81	$\pm$ 0.77	21.47	$\pm$ 0.95
RF	30.2	$\pm$ 1.29	20.67	$\pm$ 1.17
DT	33.54	$\pm$ 0.92	20.08	$\pm$ 1.45
MLP	20.0	$\pm$ 0.0	20.0	$\pm$ 0.0
LigthGBM	52.15	$\pm$ 1.14	18.84	$\pm$ 1.29
NB	27.77	$\pm$ 0.29	18.77	$\pm$ 0.63

Tabla 4.8: Resultados del preprocesamiento PII-RF modificado en UTMente-Entrenamiento etiquetado Tipo A.[5]

Método de preprocesamiento II con selección de características basada en Árboles Extremadamente Aleatorios en UTMente-Entrenamiento etiquetado Tipo A.

Al igual que en la sección 4.2.2 se muestran los resultados de la evaluación para BoW basada en N-Gramas (BoWnG), a excepción de que el método de selección utilizado es por Árboles Extremadamente Aleatorios (PII-ET), permitiendo un total de de 2,030 en BoW1G, 11,646 en BoW2G y 22,047 en BoW3G características seleccionadas. El mejor resultado se obtuvo con Máquina de Soporte Vectorial (SVM) con el modelo de BoW con Unigramas (BoW1G) con una exactitud balanceada de 40.0 % ($\pm$ 0.0) en entrenamiento y de 28.33 % ($\pm$ 0.0) en prueba (ver en la Tabla 4.9). Sin embargo, en el modelo basado en Trigramas (BoW3G) con SVM muestran la mejor relación de desempeño, sin presentar sobreajuste con una exactitud balanceada de 22.38 % ($\pm$ 0.33) y 21.98 % ($\pm$ 1.75) en entrenamiento y prueba, respectivamente.

[5]Resultados de los siete modelos: K-Vecinos más cercanos (K-NN), Bosque Aleatorio (RF), Máquina de Soporte Vectorial (SVM), Árboles de Decisión (DT), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM), Naive Bayes Multinomial (NB) y Perceptrón Multicapa (MLP) con el promedio de la métrica de exactitud balanceada (Acc B) con una validación cruzada de 5 folds, sobre el conjunto de datos UTMente-Entrenamiento etiquetado tipo A. El intervalo de confianza (IC) se calculó mediante la repetición de los experimentos cinco veces.

Modelo / Modelo	BoW1G	BoW2G	BoW3G
SVM	**40.0** (± **0.0**)	**18.46** (± **0.74**)	**22.38** (± **0.33**)
	28.33 (± **0.0**)	**24.70** (± **2.99**)	**21.98** (± **1.75**)
K-NN	30.13 (± 0.81)	36.15 (± 0.73)	27.06 (± 0.31)
	21.28 (± 0.91)	21.74 (± 0.79)	19.30 (± 0.79)
NB	42.03 (± 0.32)	40.89 (± 0.54)	31.91 (± 0.18)
	20.42 (± 0.41)	19.28 (± 0.72)	20.05 (± 0.09)
MLP	20.0 (± 0.0)	20.0 (± 0.0)	20.0 (± 0.0)
	20.0 (± 0.0)	20.0 (± 0.0)	20.0 (± 0.0)
RF	40.95 (± 1.77)	44.41* (± 0.23)	29.17 (± 0.15)
	19.99 (± 1.23)	19.65 (± 1.25)	20.37 (± 0.60)
DT	43.73 (± 2.16)	40.63 (± 2.82)	25.27 (± 0.48)
	19.72 (± 1.59)	19.15 (± 0.84)	20.95 (± 0.53)
LightGBM	68.52 (± 1.22)	45.99 (± 0.58)	32.54 (± 0.57)
	22.09 (± 1.24)	20.68 (± 1.16)	17.68 (± 0.59)

Tabla 4.9: Resultados del preprocesamiento PII-ET en UTMente-Entrenamiento etiquetado Tipo A.[6]

Para el modelo de lenguaje TF-IDF, en donde se seleccionaron 20,030 características. La Tabla 4.10 se muestra el resultado de la evaluación del entrenamiento y prueba de una validación cruzada con un k-fold con un valor $k = 5$, donde se observa que el desempeño del mejor modelo de ML para el modelo de TF-IDF fue con Máquina de Soporte Vectorial (SVM) con una exactitud balanceada de 36.17 % (± 0.35) y de 27.73 % (± 1.05), en entrenamiento y prueba, respectivamente. Por lo que, los resultados obtenidos no superan a los modelos obtenidos con el método originalmente planteado por (Byers et al., 2023) (ver Tabla 4.9).

Método de preprocesamiento II con selección de características basada en ANOVA en UTMente-Entrenamiento etiquetado Tipo A.

A continuación se muestran los resultados de la exactitud balanceada para el método propuesto, mediante el modelo de lenguaje de BoW basado en N-Gramas (BoWnG), y con el método de selección de características ANOVA (PII-ANOVA), permitiendo un total de de 2,030 en BoW1G, 11,646 en BoW2G y 22,047 en BoW3G características seleccionadas. En el caso de la selección con ANOVA el mejor resultado se obtuvo con Máquinas de Soporte Vectorial (SVM) en los tres modelos de extracción de características, pero particularmente el mejor resultado se obtuvo en Trigramas (BoW3G) con una exactitud balanceada de 73.48 % (± 0.99) en

[6]Resultados de los siete modelos: K-Vecinos más cercanos (K-NN), Bosque Aleatorio (RF), Máquina de Soporte Vectorial (SVM), Árboles de Decisión (DT), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM), Naive Bayes Multinomial (NB) y Perceptrón Multicapa (MLP) con el promedio de la métrica de exactitud balanceada (Acc B) con una validación cruzada de 5 folds, sobre el conjunto de datos UTMente-Entrenamiento etiquetado Tipo A. El intervalo de confianza (IC) se calculó mediante la repetición de los experimentos cinco veces.

Modelo	Entrenamiento		Prueba	
	Acc B	IC	Acc B	IC
SVM	**36.17**	**± 0.35**	**27.73**	**± 1.05**
NB	26.65	± 0.37	20.13	± 0.17
RF	46.93	± 2.31	18.70	± 1.52
K-NN	54.65	± 0.72	18.75	± 0.69
DT	58.88	± 1.25	19.90	± 1.54
MLP	20.0	± 0.0	20.0	± 0.0
LigthGBM	52.84	± 1.03	19.54	± 0.49

Tabla 4.10: Resultados del preprocesamiento PII-ET modificado en UTMente-Entrenamiento etiquetado Tipo A.[7]

entrenamiento y de 71.81 % (± 2.84) en prueba (ver en la Tabla 4.11).

Modelo / Modelo	BoW1G	BoW2G	BoW3G
SVM	**49.26** (± **0.45**)	**57.61** (± **0.68**)	**73.48** (± **0.99**)
	58.93 (± **1.84**)	**71.44** (± **0.67**)	**71.81** (± **2.84**)
NB	43.28 (± 0.37)	40.83 (± 0.56)	91.16 (± 0.16)
	28.29 (± 0.65)	25.04 (± 1.13)	30.40 (± 1.29)
RF	61.99 (± 0.67)	43.04 (± 1.29)	45.73 (± 0.85)
	23.21 (± 1.62)	21.98 (± 0.73)	20.74 (± 1.45)
LigthGBM	58.35 (± 0.84)	80.41 (± 0.73)	51.49 (± 1.02)
	21.44 (± 0.83)	22.07 (± 0.7)	22.27 (± 0.68)
K-NN	37.55 (± 0.51)	27.26 (± 0.28)	26.64 (± 0.87)
	22.42 (± 1.38)	21.23 (± 0.52)	20.24 (± 0.29)
DT	38.18 (± 1.32)	34.39 (± 1.86)	22.95 (± 0.33)
	20.72 (± 1.24)	19.68 (± 1.23)	20.2 (± 0.52)
MLP	20.0 (± 0.0)	20.03 (± 0.03)	20.05 (± 0.05)
	20.0 (± 0.0)	20.0 (± 0.0)	20.0 (± 0.0)

Tabla 4.11: Resultados del preprocesamiento PII-ANOVA en UTMente-Entrenamiento etiquetado Tipo A.[7]

Para la implementación propuesta, basada en el modelo de lenguaje TF-IDF, en donde se seleccionaron 20,030 características. La Tabla 4.12 se muestra el resultado de evaluación del entrenamiento y prueba de una validación cruzada con un k-fold con un valor $k = 5$, donde se observa que el desempeño del mejor modelo de ML para el modelo de TF-IDF fue con Máquina de Soporte Vectorial (SVM) con una exactitud balanceada de 49.03 % (± 0.94) y de 59.67 %

[7]Resultados de los siete modelos: K-Vecinos más cercanos (K-NN), Bosque Aleatorio (RF), Máquina de Soporte Vectorial (SVM), Árboles de Decisión (DT), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM), Naive Bayes (NB) y Perceptrón Multicapa (MLP) con el promedio de la métrica de exactitud balanceada (Acc B) con una validación cruzada de 5 folds, sobre el conjunto de datos UTMente-Entrenamiento etiquetado tipo A. El intervalo de confianza (IC) se calculó mediante la repetición de los experimentos cinco veces.

($\pm$ 1.13), en entrenamiento y prueba, respectivamente. Por lo que, los resultados obtenidos no superan a los modelos obtenidos en la Tabla 4.11 (preprocesamiento II con selección de características basada en ANOVA).

Modelo	**Entrenamiento**		**Prueba**	
	Acc B	IC	Acc B	IC
SVM	**49.03**	**$\pm$ 0.94**	**59.67**	**$\pm$ 1.13**
NB	35.69	$\pm$ 0.36	25.75	$\pm$ 0.9
K-NN	31.67	$\pm$ 0.52	25.33	$\pm$ 0.56
RF	64.6	$\pm$ 1.42	22.65	$\pm$ 1.89
DT	48.20	$\pm$ 1.86	21.21	$\pm$ 1.51
MLP	20.0	$\pm$ 0.0	20.0	$\pm$ 0.0
LigthGBM	91.22	$\pm$ 0.66	21.49	$\pm$ 1.45

Tabla 4.12: Resultados del preprocesamiento II-ANOVA en UTMente-Entrenamiento etiquetado Tipo A.[8]

4.2.3. Método de preprocesamiento III con UTMente-Entrenamiento etiquetado Tipo A.

En esta sección se presentan los resultados de utilizar el método de preprocesamiento III. En este método Yu et al. (2023) realiza una predicción de un valor numérico que representa el nivel de ansiedad. Por lo que, se utilizan los algoritmos de regresión: Regresión Lineal (LR, por sus siglas en inglés de *Linear Regression*), Regresor AdaBoost, Regresor XGBoost y Regresión de Vectores de Soporte (SVR, por sus siglas en inglés de *Support Vector Regression*).

Método de preprocesamiento III con LIWC en UTMente-Entrenamiento etiquetado Tipo A.

De acuerdo, a lo descrito en el capítulo anterior sobre este método de procesamiento, se utiliza la extracción de características mediante LIWC (PIII-LIWC), el cual calcula el nivel del uso de palabras relacionadas a emociones positivas y negativas, autoreferencias, temas de género, procesos cognitivos y aspectos sociales. En adición, se realiza una selección de características mediante correlación de Pearson y selección basada en Árboles. En la Tabla 4.13 se muestran los resultados de evaluación para los cuatro modelos obtenidos después de realizar el entrenamiento mediante el Error Absoluto Medio (MAE, por sus siglas de *Mean Absolute Error*) y la

[8]Resultados de los siete modelos: K-Vecinos más cercanos (K-NN), Bosque Aleatorio (RF), Máquina de Soporte Vectorial (SVM), Árboles de Decisión (DT), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM), Naive Bayes Multinomial (NB) y Perceptrón Multicapa (MLP) con el promedio de la métrica de exactitud balanceada (Acc B) con una validación cruzada de 5 folds, sobre el conjunto de datos UTMente-Entrenamiento etiquetado Tipo A. El intervalo de confianza (IC) se calculó mediante la repetición de los experimentos cinco veces.

Correlación de Pearson. Para obtener los resultados se utiliza la validación cruzada con k-fold, donde el valor de k es igual a 5. El modelo SVR obtuvo en la evaluación sobre el conjunto de prueba en promedio un MAE de 9.587, es decir, que el valor de T que se predice alcanza hasta una diferencia de 9.587 puntos con respecto al valor esperado.

Modelo	**Entrenamiento**		**Prueba**	
	MAE	Pearson	MAE	Pearson
SVR	**9.543**	**0.265**	**9.587**	**0.086**
XGBoost	8.433	0.733	9.742	0.003
AdaBoost	7.612	0.634	10.115	− 0.025
LR	9.079	0.339	10.174	0.095

Tabla 4.13: Resultados del preprocesamiento PIII-LIWC en UTMente-Entrenamiento etiquetado Tipo A.[9]

Dado que en el presente trabajo se realiza una comparativa para la clasificación de ansiedad y no de regresión, es decir, de la predicción de un valor numérico que represente el nivel de ansiedad, los resultado obtenidos por los modelos anteriores, son utilizados para convertir la salida de predicción de cada modelo de regresión en una clase, siguiendo la propuesta para la generación de las clases de ansiedad, descrita en la Tabla 4.1. Por lo que, se realiza una segunda evaluación de los resultados de estos modelos adaptados a la clasificación. En la Tabla 4.14 se muestra que el modelo de Regresión Lineal a multiclase (LR-2MC) presenta el mejor desempeño al alcanzar una exactitud balanceada de 20.06 % en el entrenamiento y un 20.69 % en conjunto de datos de prueba.

Modelo	Entrenamiento	Prueba
	Acc B	Acc B
LR-2MC	20.06	20.69
XGBoost-2MC	20.32	20.0
SVR-2MC	20.0	20.0
AdaBoost-2MC	23.16	18.19

Tabla 4.14: Resultados de clasificación de tipo de preprocesamiento PIII-LIWC en UTMente-Entrenamiento etiquetado Tipo A.[10]

[9]Resultados de los cuatro modelos: Regresion Lineal (LR), Regresor AdaBoost, Regresor XGBoost y Support Vector Regressor (SVR) con la métrica Error Cuadrático Medio (MAE) y Correlación de Pearson mediante validación cruzad k-fold, sobre el conjunto de datos UTMente-Entrenamiento con etiquetado Tipo A.

[10]Resultados en cuatro modelos: Regresion Lineal (LR), Regresor AdaBoost, Regresor XGBoost y Support Vector Regressor (SVR), con la métrica de exactitud balanceada (Acc B) mediante validación cruzada k-fold sobre el conjunto de datos UTMente-Entrenamiento etiquetado Tipo A.

Método de preprocesamiento III con BoWnG+LIWC en UTMente-Entrenamiento etiquetado Tipo A.

En esta sección se utiliza el método propuesto basado en el trabajo de Yu et al. (2023) donde se utiliza la extracción de características mediante LIWC, pero adicionalmente, en esta propuesta se realiza una combinación de la extracción de características de BoW basadas en N-Gramas más el diccionario LIWC (PIII-BoWnG+LIWC) para el entrenamiento de los modelos de clasificación a partir de los algoritmo de ML: K-Vecinos más cercanos (K-NN), Bosque Aleatorio (RF), Máquina de Soporte Vectorial (SVM), Árboles de Decisión (DT), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM), Naive Bayes Multinomial (NB) y Perceptrón Multicapa (MLP). Adicionalmente se realiza una la selección de características mediante correlación de Pearson y el método ANOVA (para más detalles sobre la cantidad total y las características seleccionadas se puede consultar el Apéndice B.2). En la Tabla 4.15, se muestran los resultados de la evaluación de los modelos obtenidos, donde la Máquina de Soporte Vectorial (SVM) obtuvo los mejores resultados en exactitud balanceada con Trigramas con una 72.12 % ($\pm$ 0.43) en entrenamiento y un 73.73 % ($\pm$ 1.66) en prueba, en promedio en una validación cruzada con un k-fold con un $k = 5$, y una repetición de cinco experimentos para la obtención del intervalo de confianza.

Modelo / Modelo	**BoW1G+LIWC**	**BoW1G+LIWC**	**BoW1G+LIWC**	**TF-IDF+LIWC**
SVM	**47.21** ($\pm$ **1.17**)	**63.63** ($\pm$ **1.13**)	**72.12** ($\pm$ **0.43**)	**46.95** ($\pm$ **1.02**)
	55.87 ($\pm$ **2.72**)	**67.91** ($\pm$ **2.79**)	**73.73** ($\pm$ **1.66**)	**53.72** ($\pm$ **3.41**)
K-NN	28.95 ($\pm$ 0.54)	27.32 ($\pm$ 0.56)	45.52 ($\pm$ 0.45)	40.43 ($\pm$ 0.47)
	18.56 ($\pm$ 0.54)	18.69 ($\pm$ 0.69)	21.45 ($\pm$ 0.82)	18.96 ($\pm$ 1.61)
NB	37.04 ($\pm$ 0.32)	35.06 ($\pm$ 0.36)	31.45 ($\pm$ 0.44)	26.59 ($\pm$ 0.36)
	27.07 ($\pm$ 1.0)	22.21 ($\pm$ 0.32)	20.8 ($\pm$ 0.14)	22.03 ($\pm$ 0.58)
LigthGBM	51.56 ($\pm$ 0.25)	37.58 ($\pm$ 0.57)	40.04 ($\pm$ 0.42)	33.52 ($\pm$ 0.55)
	20.37 ($\pm$ 0.64)	20.56 ($\pm$ 1.0)	20.79 ($\pm$ 1.89)	19.55 ($\pm$ 1.07)
RF	57.63 ($\pm$ 2.12)	43.92 ($\pm$ 0.46)	35.86 ($\pm$ 1.0)	43.74 ($\pm$ 1.7)
	22.87 ($\pm$ 1.34)	20.38 ($\pm$ 1.36)	19.92 ($\pm$ 0.87)	20.25 ($\pm$ 0.48)
DT	47.24 ($\pm$ 2.39)	30.0 ($\pm$ 0.52)	34.11 ($\pm$ 1.77)	53.28 ($\pm$ 3.22)
	21.4 ($\pm$ 0.95)	20.38 ($\pm$ 0.65)	19.79 ($\pm$ 0.95)	20.36 ($\pm$ 1.8)
MLP	20.0 ($\pm$ 0.0)	20.0 ($\pm$ 0.0)	20.0 ($\pm$ 0.0)	20.0 ($\pm$ 0.0)
	20.0 ($\pm$ 0.0)	20.0 ($\pm$ 0.0)	20.0 ($\pm$ 0.0)	20.0 ($\pm$ 0.0)

Tabla 4.15: Resultados de clasificación de tipo de preprocesamiento PIII-BoWnG+LIWC en UTMente-Entrenamiento etiquetado Tipo A.[11]

[11]Resultados de los siete modelos: K-Vecinos más cercanos (K-NN), Bosque Aleatorio (RF), Máquina de Soporte Vectorial (SVM), Árboles de Decisión (DT), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM), Naive Bayes (NB) y Perceptrón Multicapa (MLP) con el promedio de la métrica de exactitud balanceada (Acc B) con una validación cruzada de 5 folds, sobre el conjunto de datos UTMente-Entrenamiento etiquetado tipo A. El intervalo de confianza (IC) se calculó mediante la repetición de los experimentos cinco veces.

4.2.4. Comparación de resultados en UTMente-Ansiedad-Prueba etiquetado Tipo A.

En esta sección se realiza una evaluación de los modelos propuestos en las secciones: 4.2.1 (PI-BoWnG), 4.2.2 (PII-BoWnG), 4.2.3 (PIII-BoWnG+LIWC) y sus variantes respectivas utilizando TF-IDF con la finalidad de comparar los resultados de los métodos propuestos y de los modelos de lenguaje utilizados, para esta evaluación se utiliza el conjunto de datos UTMente-Ansiedad-Prueba etiquetado Tipo A descrito en la sección 4.1.3.

En la Figura 4.11 se muestran los resultados de la evaluación de los métodos propuestos para la BoW basada en Unigramas (BoW1G) en los modelos obtenidos a partir de los algoritmos de ML: K-Vecinos más cercanos (K-NN), Bosque Aleatorio (RF), Máquina de Soporte Vectorial (SVM), Árboles de Decisión (DT), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM), Naive Bayes Multinomial (NB) y Perceptrón Multicapa (MLP). De acuerdo con los resultados, los modelos de SVM para los métodos de procesamiento II (PII-BoW1G) y III (PIII-BoW1G) obtienen una exactitud balanceada de 57.93 % y 55.07 %, respectivamente, destacando de los demás modelos evaluados.

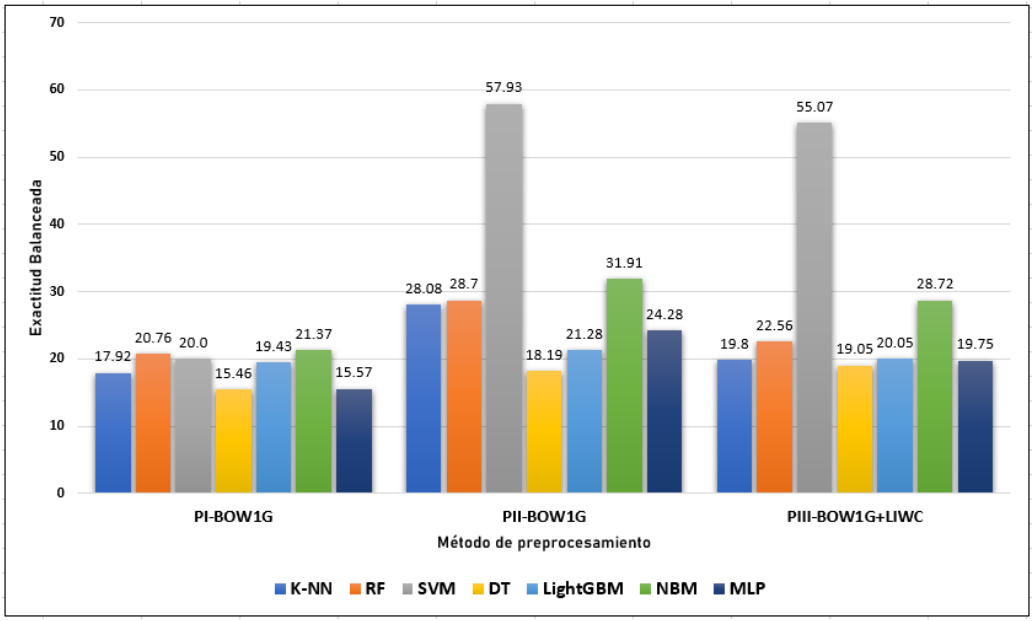


Figura 4.11: Resultados de evaluación de los modelos PI-BoW1G, PII-BoW1G, PIII-BoW1G+LIWC en UTMente-Ansiedad-Prueba etiquetado Tipo A.

En el caso de los métodos propuestos para la bolsa de palabras basada en Bigramas (BoW2G) en los modelos obtenidos a partir de los algoritmos de ML: K-Vecinos más cercanos (K-NN), Bosque Aleatorio (RF), Máquina de Soporte Vectorial (SVM), Árboles de Decisión (DT), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM), Naive Bayes Multinomial (NB) y Perceptrón Multicapa (MLP). En la Figura 4.12 se muestran los resultados de la evalua-

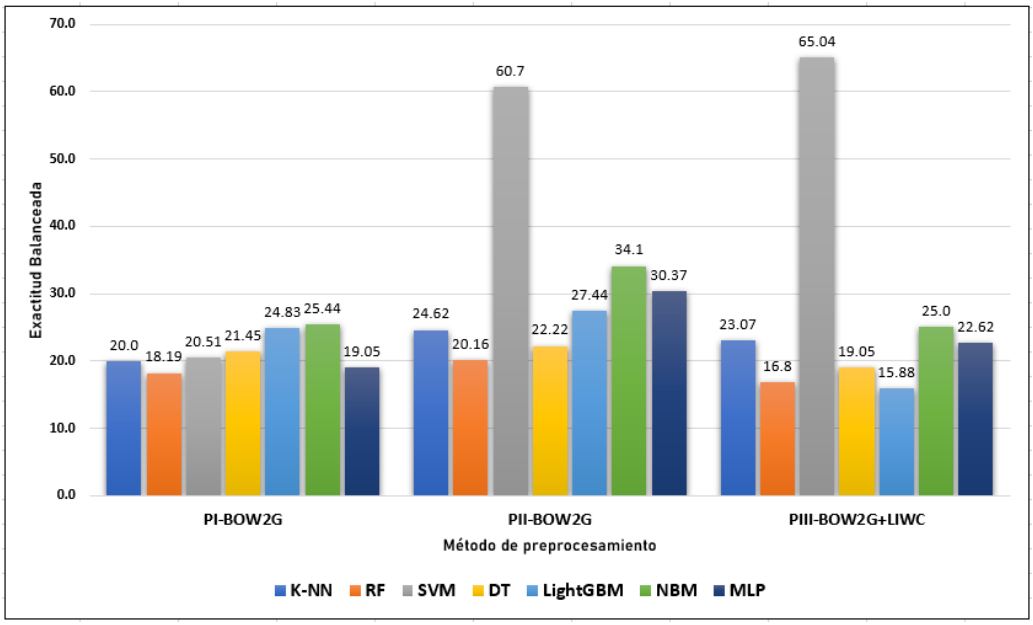


Figura 4.12: Resultados de evaluación de los modelos PI-BoW2G, PII-BoW2G, PIII-BoW2G+LIWC en UTMente-Ansiedad-Prueba etiquetado Tipo A.

ción, donde los modelos de Máquinas de Soporte Vectorial para los métodos de procesamiento II (PII-BoW2G) y III (PIII-BoW2G) obtienen un exactitud balanceada de 60.7 % y 65.04 %, respectivamente, destacando de los demás modelos evaluados.

En los métodos propuestos para la BoW basada en Trigramas (BoW3G) en los modelos obtenidos a partir de los algoritmos de ML: K-Vecinos más cercanos (K-NN), Bosque Aleatorio (RF), Máquina de Soporte Vectorial (SVM), Árboles de Decisión (DT), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM), Naive Bayes Multinomial (NB) y Perceptrón Multicapa (MLP). En la Figura 4.13 se muestran los resultados de la evaluación, donde los modelos de Máquinas de Soporte Vectorial para los métodos de procesamiento II (PII-BoW3G) y III (PII-BoW3G) obtiene una exactitud balanceada de 52.45 % y 51.53 %, respectivamente, destacando de los demás modelos evaluados.

En la Figura 4.14 se presentan los resultados de la evaluación obtenidos en los métodos propuestos basados en TF-IDF para los modelos implementados a partir de los algoritmos de ML: K Vecinos más cercanos (K-NN), Bosque Aleatorio (RF), Máquina de Soporte Vectorial (SVM), Árboles de Decisión (DT), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM), Naive Bayes Multinomial (NB) y Perceptrón Multicapa (MLP). Se puede observar que los modelos de Máquinas de Soporte Vectorial para los métodos de procesamiento II (PII-TFIDF) y III (PII-TFIDF+LIWC) obtienen una exactitud balanceada de 56.02 % y 48.03 %, respectivamente, destacando de los demás modelos evaluados.

En las Tablas 4.16 se presentan los mejores resultados obtenidos por los modelos de apren-

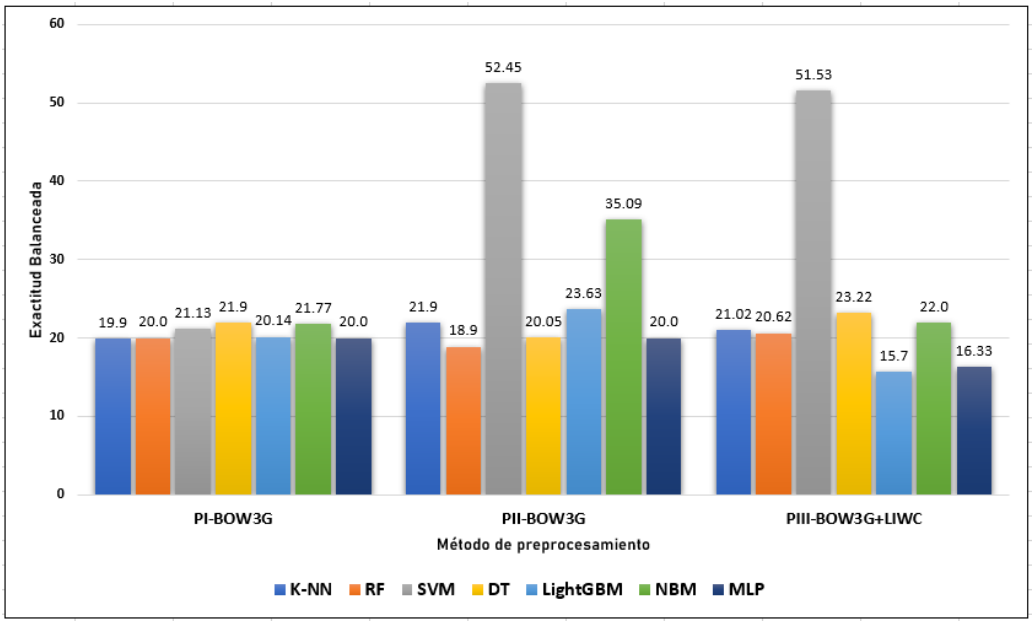


Figura 4.13: Resultados de evaluación de los modelos PI-BoW3G, PII-BoW3G, PIII-BoW3G+LIWC en UTMente-Ansiedad-Prueba etiquetado Tipo A.

dizaje supervisado en la etapa de entrenamiento para cada método propuesto de procesamiento, además se añaden los resultados obtenidos de los métodos de procesamiento originales. En la Tabla 4.16a se observan los resultados de la exactitud balanceada en la etapa de entrenamiento y prueba para el procesamiento I, donde el método propuesto con Naive Bayes Multinomial en su implementación con Bigramas (PI-BoW2G-NB) obtuvo 93.41 % y 25.44 % en entrenamiento y prueba respectivamente, superando 52.37 % en entrenamiento y 24.69 % en prueba del método original que involucra TF-IDF y K Vecinos más Cercanos como algoritmo de ML.

En la Tabla 4.16b se presentan los resultados obtenidos en exactitud balanceada en la etapa de entrenamiento y prueba para el procesamiento II, donde el método propuesto con Máquina de Soporte Vectorial en su implementación con Bigramas (PII-BoW2G-SVM) obtuvo 59.55 % y 60.7 % en entrenamiento y prueba respectivamente, superando 12.5 % en entrenamiento y 29.08 % en prueba del método original que involucra una selección de características extraídas a partir de Bigramas y la Máquina de Soporte Vectorial como algoritmo de ML.

En la Tabla 4.16c se pueden observar los resultados obtenidos en exactitud balanceada en la etapa de entrenamiento y prueba para el procesamiento III, donde el método propuesto con Máquina de soporte Vectorial en su implementación con Bigramas (PIII-BoW2G+LIWC-SVM) obtuvo 65.93 % y 65.04 % en entrenamiento y prueba respectivamente, superando 20.06 % en entrenamiento y 20.69 % en prueba del método original que involucra el diccionario LIWC y Regresión Lineal como algoritmo de ML.

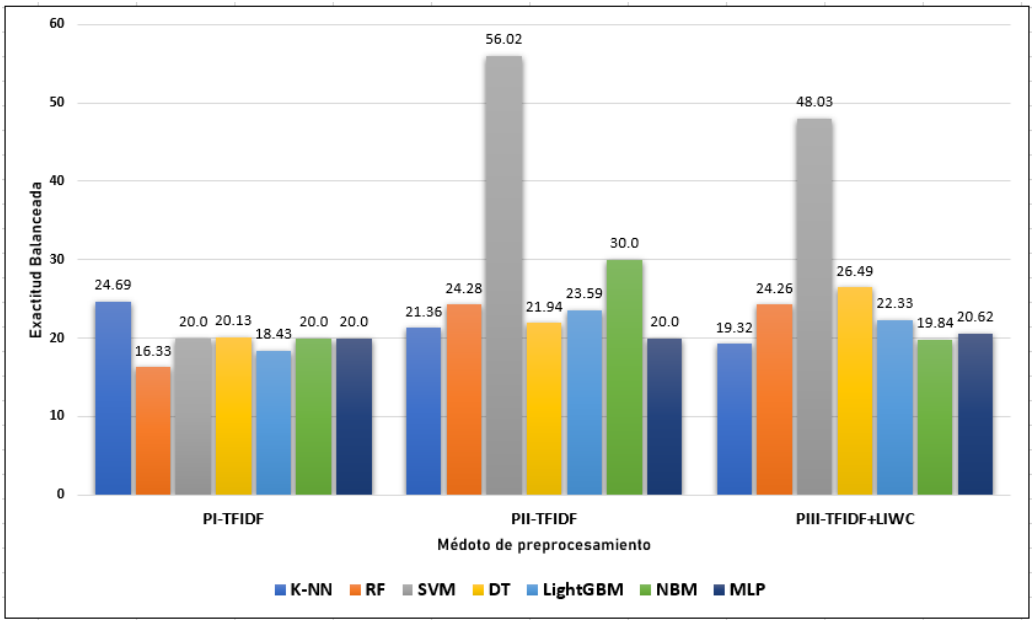


Figura 4.14: Resultados de la evaluación de los modelos PI-TFIDF, PII-TFIDF, PIII-TFIDF+LIWC en UTMente-Ansiedad-Prueba etiquetado Tipo A.

Modelo	Entrenamiento	Prueba
PI-BoW2G-NB	**93.41**	**25.44**
PI-TFIDF-KNN	52.37	24.69
PI-BoW3G-NB	**99.70**	**21.77**
PI-BoW1G-NB	**36.34**	**21.37**

(a) Resultados del Preprocesamiento I.

Modelo	Entrenamiento	Prueba
PII-BoW2G-SVM	**59.55**	**60.7**
PII-BoW1G-SVM	**52.52**	**57.93**
PII-TFIDF-SVM	**40.31**	**56.02**
PII-BoW3G-SVM	**71.67**	**52.45**
PII-BoW2G-SVM-ET	12.50	29.08

(b) Resultados del Preprocesamiento II.

Modelo	Entrenamiento	Prueba
PIII-BoW2G+LIWC-SVM	**65.93**	**65.04**
PIII-BoW1G+LIWC-SVM	**46.53**	**55.07**
PIII-BoW3G+LIWC-SVM	**71.37**	**51.53**
PIII-TFIDF+LIWC-SVM	**46.77**	**48.03**
PIII-LIWC-LR-2MC	20.06	20.69

(c) Resultados del Preprocesamiento III.

Tabla 4.16: Comparación de los resultados de clasificación de los mejores modelos de aprendizaje supervisado en UTMente-Entrenamiento etiquetado Tipo A.

4.3. Métodos semisupervisados de clasificación en UTMente-Ansiedad etiquetado Tipo B

En esta sección se presentan los resultados de tres métodos basados en aprendizaje semisupervisado para detección de ansiedad en UTMente-Ansiedad etiquetado Tipo B, es decir, se determina si la persona padece o no, ansiedad. Dado que en el aprendizaje semisupervisado, parte de tener un conjunto etiquetado a partir del cual se construirá un modelo que permite realizar el pseudoetiquetado de otro conjunto de datos, se utilizarán dos conjuntos etiquetados manualmente. Asimismo, se lleva a cabo una selección de características en cada conjunto de datos, esto como parte del método propuesto descrito en el Sección 3.3 y para ver detalles sobre la cantidad total y las características seleccionadas se puede consultar el Apéndice B.3.

El conjunto de datos SocialMedia-Anxiety fue publicado por Saha (2022) en Kaggle, los textos fueron extraídos de redes sociales como: Facebook®, Twitter® y otros. El conjunto de datos fue etiquetado manualmente por cuatro estudiantes universitarios, sin embargo, no se especificó cual era la especialidad y conocimiento de los estudiantes que etiquetaron sobre los problemas de salud mental asociados con la ansiedad. El conjunto de datos SocialMedia-Anxiey está compuesto por 6980 textos escritos en idioma inglés, donde 733 y 6247 son textos etiquetados con ansiedad y sin ansiedad, respectivamente.

El segundo conjunto de datos UTMenteII-Ansiedad fue obtenido de los casos descartados en la creación de UTMente-Ansiedad (ver sección 4.1.1) incluye 76 textos en español, compuesto por 34 y 42 datos etiquetados como ansiedad y sin ansiedad, respectivamente. El etiquetado fue realizado de forma manual con el apoyo de la Licenciada en Psicología y Maestra en Educación Denisse Millán Hernández.

4.3.1. SocialMedia-Anxiety a UTMente-Ansiedad-Entrenamiento etiquetado Tipo B

Para el primer método semisupervisado se utiliza SocialMedia-Anxiety como base, para realizar el pseudoetiquetado de UTMente-Ansiedad-Entrenamiento. Primero se realizó la eliminación de números, signos de puntuación y caracteres distintos a letras, conversión a minúsculas, y expansión de contracciones para el idioma inglés. Adicional se realizó una traducción del corpus al idioma español. Posteriormente la extracción de características se realizó mediante los modelos de lenguaje TF-IDF y BoW con N-Gramas (BoWnG) en tres variantes: Unigramas (BoW1G), Bigramas (BoW2G) y Trigramas (BoW3G), para finalmente realizar una selección de características mediante el método de ANOVA.

Se entrenaron y evaluaron tres modelos de ML mediante una validación cruzada con k-

fold, donde el valor de $k = 5$. Los algoritmos utilizados para la generación de los modelos son: Máquina de Soporte Vectorial (SVM), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM) y Naive Bayes Multinomial (NB). En la Tabla 4.17 se muestra el resultado de los tres modelos, en donde Naive Bayes Multinomial (NB) obtuvo los mejores resultados en exactitud balanceada utilizando Unigramas, Bigramas y TF-IDF con 98.66 % (± 0.04), 99.25 % (± 0.02), 96.48 % (± 0.02) en entrenamiento respectivamente y 90.60 % (± 0.34), 90.36 % (± 0.16) y 91.26 % (± 0.20) en prueba respectivamente, en promedio en una validación cruzada con un k-fold con un $k = 5$, y una repetición de cinco experimentos para la obtención del intervalo de confianza.

Modelo / Modelo	**BoW1G**	**BoW2G**	**BoW3G**	**TF-IDF**
NB	**98.66** (± **0.04**)	99.25 (± 0.02)	60.09 (± 0.29)	96.48 (± 0.02)
	90.60 (± **0.34**)	90.36 (± 0.16)	54.46 (± 0.54)	91.26 (± 0.20)
SVM	82.17 (± 0.16)	72.65 (± 0.11)	67.69 (± 0.1)	79.08 (± 0.17)
	89.65 (± 0.25)	74.16 (± 0.65)	64.37 (± 0.41)	88.1 (± 0.58)
LightGBM	87.8 (± 0.24)	69.71 (± 0.21)	51.97 (± 0.32)	88.87 (± 0.19)
	85.88 (± 0.3)	66.88 (± 0.19)	51.19 (± 0.17)	86.46 (± 0.09)

Tabla 4.17: Resultados de evaluación del entrenamiento de SocialMedia-Anxiety.[12]

La segunda parte del primer método semisupervisado consta de pseudoetiquetar UTMente-Ansiedad-Entrenamiento utilizando los tres algoritmos de ML mostrados en la Tabla 4.17. Para evaluar la efectividad del pseudotiquetado se recuperaron las etiquetas originales de UTMente-Ansiedad-Entrenamiento etiquetado Tipo B y se calculó la exactitud balanceada para cada algoritmo con respecto a cada modelo de lenguaje, los resultados se muestran en la Tabla 4.18, donde la Máquina de Soporte Vectorial en su implementación con Trigramas obtuvo el mejor resultado con una exactitud balanceada de 52.33 %.

Modelo/Modelo	**BoW1G**	**BoW2G**	**BoW3G**	**TF-IDF**
SVM	48.93	51.02	**52.33**	48.69
LightGBM	47.51	51.73	50.41	48.24
NB	50.0	50.0	49.56	50.0

Tabla 4.18: Resultados de evaluación del pseudoetiquetado de UTMente-Ansiedad-Entrenamiento etiquetado Tipo B.[13]

[12]Resultados de la métrica de exactitud balanceada mediante validación cruzada k-fold de los algoritmos: Máquina de Soporte Vectorial (SVM), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM) y Naive Bayes Multinomial (NB) con el promedio de la métrica de exactitud balanceada (Acc B) con una validación cruzada de 5 folds, sobre el conjunto de datos SocialMedia-Anxiety. El intervalo de confianza (IC) se calculó mediante la repetición de los experimentos cinco veces.

[13]Resultados de la métrica de exactitud balanceada de los algoritmos: Máquina de Soporte Vectorial (SVM), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM) y Naive Bayes Multinomial (NB) sobre el pseudoetiquetado del conjunto de datos UTMente-Ansiedad-Entremaniento etiquetado Tipo B.

4.3.2. UTMente-Ansiedad-Entrenamiento etiquetado Tipo B a SocialMedia-Anxiety

Para el segundo método semisupervisado se utiliza UTMente-Ansiedad-Entrenamiento etiquetado Tipo B como base, para realizar el pseudoetiquetado de SocialMedia-Anxiety. Después de realizar la limpieza de los textos de SocialMedia-Anxiety como se hizo en el experimento anterior, se realizó una traducción de los textos al idioma español para coincidir con el idioma del conjunto de datos UTMente-Ansiedad-Entrenamiento. Posteriormente la extracción de características se realizó mediante los modelos de lenguaje TF-IDF y BoW con N-Gramas (BoWnG) en tres variantes: Unigramas (BoW1G), Bigramas (BoW2G) y Trigramas (BoW3G), para finalmente realizar una selección de características mediante el método de ANOVA.

Se entrenaron y evaluaron tres modelos de ML mediante una validación cruzada con k-fold, donde el valor de $k = 5$. Los algoritmos utilizados para la generación de los modelos son: Máquina de Soporte Vectorial (SVM), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM) y Naive Bayes Multinomial (NB). En la Tabla 4.19 se muestra el resultado de los modelos, en donde la Máquina de Soporte Vectorial obtuvo los mejores resultados en exactitud balanceada utilizando Bigramas y Trigramas con 100.0 % ($\pm$ 0.0), 100.0 % ($\pm$ 0.0) en entrenamiento respectivamente y 91.35 % ($\pm$ 0.63) y 91.57 % ($\pm$ 0.33) en prueba respectivamente, en promedio en una validación cruzada con un k-fold con un $k = 5$, y una repetición de cinco experimentos para la obtención del intervalo de confianza.

Modelo / Modelo	BoW1G	BoW2G	BoW3G	TF-IDF
SVM	79.81 ($\pm$ 0.54)	**100.0** ($\pm$ **0.0**)	**100.0** ($\pm$ **0.0**)	77.72 ($\pm$ 0.69)
	84.24 ($\pm$ 1.83)	**91.24** ($\pm$ **0.63**)	**91.57** ($\pm$ **0.33**)	81.51 ($\pm$ 1.4)
NB	99.91 ($\pm$ 0.04)	60.89 ($\pm$ 0.24)	56.01 ($\pm$ 0.43)	100.0 ($\pm$ 0.0)
	86.32 ($\pm$ 1.07)	50.0 ($\pm$ 0.0)	50.0 ($\pm$ 0.0)	81.33 ($\pm$ 1.22)
LightGBM	71.81 ($\pm$ 0.48)	71.11 ($\pm$ 0.38)	65.31 ($\pm$ 0.54)	71.28 ($\pm$ 1.07)
	53.24 ($\pm$ 1.35)	51.49 ($\pm$ 2.41)	54.20 ($\pm$ 2.29)	52.03 ($\pm$ 2.02)

Tabla 4.19: Resultados de evaluación del entrenamiento de UTMente-Ansiedad-Entrenamiento etiquetado Tipo B.[14]

La segunda parte del segundo método semisupervisado consta de pseudoetiquetar SocialMedia-Anxiety utilizando los tres algoritmos de ML mostrados en la Tabla 4.19. Para evaluar la efectividad del pseudotiquetado se recuperaron las etiquetas originales de SocialMedia-Anxiety y se calculó la exactitud balanceada para cada algoritmo con respecto a cada modelo de lenguaje, los

[14]Resultados de la métrica de exactitud balanceada mediante validación cruzada k-fold de los algoritmos: Máquina de Soporte Vectorial (SVM), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM) y Naive Bayes Multinomial (NB) con el promedio de la métrica de exactitud balanceada (Acc B) con una validación cruzada de 5 folds, sobre el conjunto de datos UTMente-Ansiedad-Entrenamiento etiquetado Tipo B. El intervalo de confianza (IC) se calculó mediante la repetición de los experimentos cinco veces.

resultados se muestran en la Tabla 4.20, donde Naive Bayes Multinomial en su implementación con Unigramas obtuvo el mejor resultado con una exactitud balanceada de 57.61 %.

Modelo/Modelo	BoW1G	BoW2G	BoW3G	TF-IDF
NB	**57.61**	49.99	50.0	56.68
SVM	51.64	50.44	50.13	55.38
LightGBM	50.1	50.12	50.09	50.36

Tabla 4.20: Resultados de evaluación del pseudoetiquetado de SocialMedia-Anxiety.[15]

4.3.3. UTMenteII-Ansiedad a UTMente-Ansiedad-Entrenamiento etiquetado Tipo B

Para el tercer método semisupervisado se utiliza UTMenteII-Ansiedad como base, para realizar el pseudoetiquetado de UTMente-Ansiedad-Entrenamiento. Primero se realizó la limpieza de los textos en UTMenteII-Ansiedad mediante eliminación de números, signos de puntuación y caracteres distintos a letras y conversión a minúsculas. Posteriormente la extracción de características se realizó mediante los modelos de lenguaje TF-IDF y BoW con N-Gramas (BoWnG) en tres variantes: Unigramas (BoW1G), Bigramas (BoW2G) y Trigramas (BoW3G), para finalmente realizar una selección de características mediante el método de ANOVA.

Se entrenaron y evaluaron tres modelos de ML mediante una validación cruzada con k-fold, donde el valor de $k = 5$. Los algoritmos utilizados para la generación de los modelos son: Máquina de Soporte Vectorial (SVM), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM) y Naive Bayes Multinomial (NB). En la Tabla 4.21 se muestra el resultado de los tres modelos, en donde la Máquina de Soporte Vectorial obtuvo los mejores resultados en exactitud balanceada, particularmente al utilizar Bigramas con 99.91 % ($\pm$ 0.16) en entrenamiento y 98.13 % ($\pm$ 0.14) en prueba, como promedio durante la validación cruzada con un k-fold definiendo un $k = 5$, y una repetición de cinco experimentos para la obtención del intervalo de confianza.

La segunda parte del tercer método semisupervisado consta de pseudoetiquetar UTMente-Ansiedad-Entrenamiento utilizando los tres algoritmos de ML mostrados en la Tabla 4.21. Para

[15]Resultados de la métrica de exactitud balanceada de los algoritmos: Máquina de Soporte Vectorial (SVM), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM) y Naive Bayes Multinomial (NB) sobre el pseudoetiquetado del conjunto de datos SocialMedia-Anxiety.

[16]Resultados de la métrica de exactitud balanceada mediante validación cruzada k-fold de los algoritmos: Máquina de Soporte Vectorial (SVM), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM) y Naive Bayes Multinomial (NB) con el promedio de la métrica de exactitud balanceada (Acc B) con una validación cruzada de 5 folds, sobre el conjunto de datos UTMenteII-Ansiedad. El intervalo de confianza (IC) se calculó mediante la repetición de los experimentos cinco veces.

Modelo / Modelo	BoW1G	BoW2G	BoW3G	TF-IDF
SVM	99.54 (± 0.26)	**99.91 (± 0.16)**	100.0 (± 0.0)	99.37 (± 0.39)
	97.18 (± 0.73)	**98.13 (± 0.14)**	72.0 (± 1.93)	95.12 (± 1.7)
NB	100.0 (± 0.0)	99.9 (± 0.17)	70.33 (± 1.02)	100.0 (± 0.0)
	97.85 (± 1.44)	83.12 (± 3.58)	50.0 (± 0.0)	97.67 (± 1.61)
LightGBM	77.92 (± 0.78)	72.85 (± 1.6)	50.31 (± 0.54)	76.63 (± 0.42)
	70.14 (± 4.73)	65.04 (± 3.78)	49.54 (± 0.8)	64.46 (± 1.29)

Tabla 4.21: Resultados de evaluación del entrenamiento de UTMenteII-Ansiedad.[16]

evaluar la efectividad del pseudotiquetado se recuperaron las etiquetas originales de UTMente-Ansiedad-Entrenamiento etiquetado Tipo B y se calculó la exactitud balanceada para cada algoritmo con respecto a cada modelo de lenguaje, los resultados se muestran en la Tabla 4.22, donde el LigthGBM en su implementación con TF-IDF obtuvo el mejor resultado con una exactitud balanceada de 54.10 %.

Modelo/Modelo	BoW1G	BoW2G	BoW3G	TF-IDF
LightGBM	53.36	44.11	53.38	**54.1**
NB	51.71	47.78	49.46	52.04
SVM	50.13	47.0	45.42	52.11

Tabla 4.22: Resultados de evaluación del pseudoetiquetado de UTMente-Ansiedad-Entrenamiento etiquetado Tipo B.[17]

4.3.4. Comparación de resultados en UTMente-Ansiedad-Prueba etiquetado Tipo B

En esta sección se presentan los resultados del entrenamiento y evaluación obtenidos de los modelos de ML: Máquina de Soporte Vectorial (SVM), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM) y Naive Bayes Multinomial (NB) al ser entrenados con el conjunto de datos base utilizados en cada método semisupervisado pero incluyendo los datos pseudoetiquetados de la segunda etapa en los métodos semisupervisados.

En la Tabla 4.23 se muestran los resultados obtenidos del primer método semisupervisado, donde la Máquina de Soporte Vectorial en su implementación con Unigramas obtuvo el mejor resultado con una exactitud balanceada de 77.84 % y 69.18 % en entrenamiento y prueba respectivamente.

[17]Resultados de la métrica de exactitud balanceada de los algoritmos: Máquina de Soporte Vectorial (SVM), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM) y Naive Bayes Multinomial (NB) sobre el pseudoetiquetado del conjunto de datos UTMente-Ansiedad-Entrenamiento etiquetado Tipo B.

Modelo / Modelo	BoW1G	BoW2G	BoW3G	TF-IDF
SVM	**77.84**	74.5	68.21	74.5
	69.18	51.4	53.88	46.35
LightGBM	91.94	74.34	59.45	92.44
	57.92	52.25	52.1	56.83
NB	97.34	98.63	97.94	96.02
	50.0	50.0	48.91	50.0

Tabla 4.23: Resultados de Entrenar con SocialMedia-Anxiety en conjunto con el pseudoetiquetado de UTMente-Ansiedad-Entrenamiento etiquetado Tipo B y probar en UTMente-Ansiedad-Prueba etiquetado Tipo B.[18]

En la Tabla 4.24 se presentan los resultados obtenidos del segundo método semisupervisado, donde la Máquina de Soporte Vectorial en su implementación con Bigramas obtuvo el mejor resultado con una exactitud balanceada de 74.83 % y 97.83 % en entrenamiento y prueba respectivamente.

Modelo / Modelo	BoW1G	BoW2G	BoW3G	TF-IDF
SVM	65.84	**74.83**	**86.06**	**81.11**
	42.31	**97.83**	**91.77**	**83.85**
NB	95.87	50.0	50.0	95.75
	81.06	50.0	50.0	84.63
LightGBM	76.16	96.96	91.72	96.64
	56.13	49.07	52.72	44.72

Tabla 4.24: Resultados de Entrenar con UTMente-Ansiedad-Entrenamiento etiquetado Tipo B en conjunto con el peseudoetiquetado de SocialMedia-Anxiety y probar en UTMente-Ansiedad-Prueba etiquetado Tipo B.[18]

En la Tabla 4.25 se muestran los resultados obtenidos del tercer método semisupervisado, donde Naive Bayes Multinomial en su implementación con Unigramas obtuvo el mejor resultado con una exactitud balanceada de 97.07 % y 60.40 % en entrenamiento y prueba respectivamente.

4.4. Resumen

Con la aplicación de un cuestionario como instrumento de recolección de datos aplicado a alumnos del curso propedéutico impartido en la Universidad Tecnológica de la Mixteca en agosto de 2023 se obtuvieron 443 registros que cuentan con datos demográficos de los participantes, así como las respuestas de los reactivos que conforman el manual AMAS-C y una

[18]Resultados de la métrica de exactitud balanceada de los algoritmos: Máquina de Soporte Vectorial (SVM), Método de Ensamblaje de Refuerzo de Gradientes Ligero (LigthGBM) y Naive Bayes Multinomial (NB), sobre el conjunto de prueba UTMente-Ansiedad-Prueba etiquetado Tipo B.

Modelo / Modelo	BoW1G	BoW2G	BoW3G	TF-IDF
NB	**97.07**	65.46	51.39	98.35
	60.4	50.0	50.0	56.13
LightGBM	94.5	93.79	50.0	92.75
	60.02	59.32	50.0	59.63
SVM	76.62	63.21	96.57	72.21
	50.70	57.22	51.94	56.13

Tabla 4.25: Resultados de Entrenar con UTMenteII-Ansiedad en conjunto con el peseudoetiquetado de UTMente-Ansiedad-Entrenamiento etiquetado Tipo B y probar en UTMente-Ansiedad-Prueba etiquetado Tipo B.[18]

descripción en texto sobre la percepción de cada participante con respecto una imagen presentada. La evaluación de acuerdo al AMAS-C permitió etiquetar los registros con una puntuación T, conformando así la base de datos denominada UTMente-Ansiedad. Sin embargo del total de registros se descartaron 76 por identificar distorsión intencionada por parte de los participantes de acuerdo con el parámetro de *mentira* que incluye el manual AMAS-C. Los 367 registros aceptados llevaron a la definición de un corpus que se dividio en datos de entrenamiento y prueba (UTMente-Ansiedad-Entrenamiento y UTMente-Ansiedad-Prueba), en donde cada documento fue categorizado en con dos tipos diferentes de etiquetado según la puntuación T. En primer caso se manejo un etiquetado multiclase considerando cinco niveles de ansiedad al cual se le denominado etiquetado Tipo A y en el segundo caso, los textos se dividieron en dos clases: *Con ansiedad* y *Sin ansiedad*, definido como etiquetado Tipo B.

A lo largo de los experimentos se abordaron dos enfoques según el tipo de aprendizaje de los modelos: Supervisado y Semisupervisado. En el primer enfoque se utilizaron los datos con etiquetado Tipo A del conjunto UTMente-Ansiedad-Entrenamiento y se emplearon en tres métodos propuestos de preprocesamiento. En cada método se aplicaron los modelos de lenguaje de BoW, N-Gramas (Bigramas y Trigramas) y TF-IDF en el proceso de extracción de caracteristicas y solo en el tercer método se agregaron las caracteristicas extraídas con el diccionario LIWC. Además en los métodos de preprocesamiento II y III se realizo una selección de características. El mejor resultado se obtuvo con el método III al evaluar el modelo desarrollado a partir de Máquina de Soporte Vectorial y Bigramas (PIII-BoW2G+LIWC-SVM) en el conjunto UTMente-Ansiedad-Prueba etiquetado Tipo A.

En el segundo enfoque con aprendizaje Semisupervisado se llevaron a cabo tres experimentos en donde se extrajeron características aplicando los modelos de lenguaje de BoW, N-Gramas (Bigramas y Trigramas) y TF-IDF, ademas de que se realizó una selección de características mediante el método de ANOVA. Los datos utilizados en los experimento fueron: el conjunto de datos UTMente-Ansiedad entrenamiento etiquetado Tipo B y dos conjunto de datos adiciona-

les, SocialMedia-Anxiety y UTMenteII-Ansiedad. El primero se recupero de Kaggle publicado por Saha (2022) y el segundo se creo a partir de los 76 registros descartados, a los cuales se les realizo un etiquetado manual por parte de la Licenciada en Psicología y Maestra en Educación Denisse Millán Hernández. Por cada experimento que se ocupo un conjunto de datos para entrenar un modelo base que posteriormente se utilizo para pseudoetiquetar alguno de los otros dos conjuntos de datos y de esta forma crear un conjunto de datos mas robusto que se utilizo para entrenar un nuevo modelo y ser evaluado por el conjunto de datos de prueba. De los tres experimentos realizados el que mejor resultado obtuvo fue donde se utilizo el conjunto de datos de UTMente-Ansiedad entrenamiento etirquetado Tipo B y el conjunto de datos SocialMedia-Anxiety con pseudoetiquetas. Al momento de realizar la evaluación con el conjunto UTMente-Prueba etiquetado Tipo B se obtuvo el mejor resultado con el modelo desarrollado a partir de Máquina de Soporte Vectorial y Bigramas.

Finalmente, los resultados obtenidos de los métodos propuestos tanto en el enfoque de aprendizaje Supervisado con en el Semisupervisado superaron los resultado de los métodos implementados en el estado del arte.

Capítulo 5

Conclusiones

La ansiedad, al desarrollarse y convertirse en un trastorno, afecta significativamente la salud mental de quienes la padecen, impactando de manera negativa diversos aspectos de su vida cotidiana. En los últimos años, la presencia de la ansiedad en la sociedad ha incrementado, posicionándose como uno de los principales problemas de salud mental. Este fenómeno se ha hecho evidente en el ámbito académico, y la Universidad Tecnológica de la Mixteca no ha sido la excepción. Los resultados obtenidos en este trabajo evidencian la existencia de niveles considerables de ansiedad entre los alumnos evaluados, tanto a nivel significativo como extremo. Dado el impacto que la ansiedad tiene en la sociedad y, particularmente, en la vida de los estudiantes, resulta esencial desarrollar herramientas que asistan a los expertos en su detección temprana. El uso de aprendizaje automático (ML, por sus siglas en inglés de *Machine Learning*) en conjunto con el Procesamiento de Lenguage Natural (NLP, por sus siglas en inglés de *Natural Lenguaje Processing*) adquiere gran relevancia para el desarrollo de dichas herramientas, que tienen la capacidad de identificar patrones de ansiedad en los textos escritos por las personas. Los resultados presentados en este trabajo demuestran la efectividad de este enfoque

Los resultados de esta investigación confirman que es posible detectar la ansiedad en personas mediante el uso de ML a partir del análisis de textos cortos. Sin embargo, la eficiencia de estos resultados depende de las etapas previas de tratamiento de los datos antes de aplicar cualquier algoritmo de ML. Además, se muestra que las propuestas implementadas en este trabajo superan los resultados obtenidos en investigaciones del estado del arte. De esta manera, el objetivo de la investigación se cumplió, ya que los resultados alcanzados muestran que se logró realizar un análisis efectivo de textos cortos y, con el apoyo de técnicas de ML, llevar a cabo la detección de la ansiedad.

Para alcanzar los objetivos específicos de este estudio, se comenzó con una revisión documental sobre el aprendizaje supervisado y semisupervisado, centrada en la detección de ansiedad mediante el uso de NLP. En el segundo capítulo, se recopiló información relevante sobre estos conceptos, mientras que en la sección de resultados se detalló la creación de un corpus

conformado por textos escritos por estudiantes, etiquetados con la escala AMAS (por sus siglas en inglés de *Adult Manifest Anxiety Scale*) en su versión *C* (AMAS-C). De este modo, se logró cumplir uno de los principales objetivos del estudio.

En el Capítulo tres se describió en detalle la metodología empleada, la cual abarcó tanto el aprendizaje supervisado como el semisupervisado. Los resultados fueron evaluados mediante la métrica de exactitud balanceada, lo que permitió comparar el rendimiento de diversos algoritmos bajo diferentes técnicas de preprocesamiento en el enfoque supervisado y con varios conjuntos de datos en el enfoque semisupervisado. En particular, el algoritmo de Máquina de Soporte Vectorial (SVM, por sus siglas en inglés de *Support Vector Machine*) mostró un desempeño destacado al combinarse con la tercera técnica de preprocesamiento en el enfoque supervisado, y en el segundo experimento realizado bajo el enfoque semisupervisado.

Respecto a la hipótesis de este trabajo, no fue rechazada. En el aprendizaje supervisado, se alcanzó una exactitud balanceada del 65.04 % con la técnica de preprocesamiento III, mientras que en el aprendizaje semisupervisado, se obtuvo una exactitud balanceada del 97.83 % en el segundo experimento. Estos resultados sugieren, con base en la métrica utilizada, que los algoritmos lograron un desempeño adecuado. Además, se observó una correlación entre el contenido escrito por los participantes y su estado emocional, lo que refleja cómo la forma de escribir de una persona está influenciada por su estado mental y la posible presencia de ansiedad.

De acuerdo con los fundamentos teóricos abordados en el marco teórico, se destacó la importancia de utilizar una métrica adecuada para evaluar el rendimiento de los modelos, considerando el desbalance en los datos. En este sentido, el uso de la exactitud balanceada en lugar de la exactitud tradicional utilizada en estudios previos, permitió obtener una evaluación más precisa de los resultados.

Además, en el proceso de extracción de características, los N-gramas, especialmente los Bigramas, sobresalieron frente a otros modelos de lenguaje. Según lo explicado en el marco teórico, los Bigramas permiten una mejor obtención de información al considerar secuencias consecutivas de palabras en los textos, lo que facilita a los algoritmos de ML identificar patrones más precisos relacionados con la presencia de ansiedad. A esto se suma la selección de características, una etapa crucial para evitar el sobreajuste de los algoritmos, ya que elimina aquellas características que pueden introducir ruido y afectar negativamente su desempeño. Este proceso fue particularmente relevante al comparar los resultados de las técnicas de preprocesamiento II y III en el enfoque de aprendizaje supervisado, donde ambas incluyeron selección de características, en contraste con la técnica de preprocesamiento I. Los resultados obtenidos con la técnica III mostraron mejoras significativas, lo que subraya la importancia de esta etapa en la optimización del rendimiento de los modelos.

Por último, como se discute en las ventajas de las SVM, este algoritmo demostró una capacidad superior para manejar la bidimensionalidad inherente a las características extraídas de textos. En los experimentos realizados, las SVM lograron adaptarse mejor a esta complejidad en comparación con otros algoritmos evaluados, lo que explica su destacado rendimiento en los diferentes tipos de preprocesamiento aplicados.

5.1. Aportaciones

- La construcción de un conjunto de datos compuesto por textos cortos orientado a la detección de ansiedad. Este corpus facilita la aplicación de técnicas de NLP y ML en la detección de cinco niveles de ansiedad mediante el análisis de textos.
- Los resultados de esta tesis muestran la importancia de las técnicas de preprocesamiento en los textos, así como la elección adecuada de una métrica de evaluación para obtener un mejor rendimiento de los modelos teniendo en cuenta el desbalance de los datos.
- La obtención de modelos de aprendizaje supervisado y semisupervisado que mejoran los resultados del estado del arte para la detección de ansiedad a partir del análisis de textos.

5.2. Trabajo futuro

A partir de la experiencia obtenieda en este trabajo se consideran dos líneas de investigación que pueden ser exploradas por el lector o por un tercero con la intención de mejorar la detección de ansiedad a partir del análisis de textos. La primera de ellas se enfoca en implementar otras técnicas de preprocesamiento en los textos analizados. Entre las posibles mejoras, se podrían incorporar procesos de corrección de errores ortográficos, que ayudarían a depurar el texto y asegurar que las palabras se representen de manera consistente en todo el conjunto de datos. Asimismo, la lematización y el *stemming* son enfoques que podrían ser explorados, ya que ambos permiten reducir las palabras a su forma base o raíz, lo que contribuye a minimizar la variabilidad en los términos utilizados, optimizando así la representación del texto.

Otra dirección interesante para investigaciones futuras es experimentar con algoritmos de ML diferentes a los utilizados en este proyecto. Al probar modelos alternativos, se podría evaluar si alguna de estas técnicas ofrece mejoras en el rendimiento de los modelos. Además, la combinación de estos algoritmos con técnicas avanzadas de extracción de características, como la utilización de embeddings de palabras mediante modelos como Word2Vec, podría abrir nuevas oportunidades para mejorar la detección de ansiedad. La evaluación del impacto de estas técnicas sobre los resultados generaría una comprensión más profunda de qué enfoques son más eficaces para este tipo de análisis y contribuiría al desarrollo de modelos más robustos y precisos.

Bibliografía

APA (2017). American Psychological Association (APA). Más allá de la preocupación. Disponible: https://www.apa.org/topics/anxiety/preocupacion#:~:text=La%20ansiedad%20es%20una%20reacci%C3%B3n,con%20el%20paso%20del%20tiempo Recuperado: 17/08/2023.

Asra, F., Li, Y., Hills, T. T., and Stella, M. (2021). Dasentimental: Detecting depression, anxiety, and stress in texts via emotional recall, cognitive networks, and machine learning. *Big Data and Cognitive Computing*, 5(4).

Beidel, D. C. and Brooke, S. (1997). Anxiety disorders. In Turner, S. M. and Hersen, M. E., editors, *Adult Psychopathology and Diagnosis*, chapter 11, pages 239 – 409. John Wiley & Sons Inc.

Bentéjac, C., Csörgő, A., and Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 1st ed. 2006. corr. 2nd printing edition.

Brownlee, J. (2020). *Imbalanced Classification with Python: Better Metrics, Balance Skewed Classes, Cost-Sensitive Learning*. Machine Learning Mastery. Independently Published, 1.2 edition.

Burkov, A. (2019). *The hundred page machine learning book*. Andriy Burkov.

Byers, M., Trahan, M., Nason, E., Eigege, C., Moore, N., Washburn, M., and Metsis, V. (2023). Detecting Intensity of Anxiety in Language of Student Veterans with Social Anxiety Using Text Analysis. *Journal of Technology in Human Services*, 41:125–147.

CUN (2020). Clinica Universidad de Navarra. Ansiedad. Síntomas, ataque de ansiedad, control y tratamiento. Disponible: https://www.cun.es/enfermedades-tratamientos/enfermedades/ansiedad#:~:text=La%20ansiedad%20es%20anormal%20cuando,empeorar%20si%20no%20se%20tratan Recuperado: 17/08/2023.

Devore, J. L. (2015). *Probability and Statistics for Engineering and the Sciences*. Cengage Learning, 9 edition.

Efron, B. and Hastie, T. (2016). *Computer Age Statistical Inference: Algorithms, Evidence, and Data Science*. Institute of Mathematical Statistics Monographs. Cambridge University Press, 1 edition.

Elgendi, M., Galli, V., Ahmadizadeh, C., and Menon, C. (2022). Dataset of psychological scales and physiological signals collected for anxiety assessment using a portable device. *Data*, 7(9).

Fernández López, O., Jiménez Hernández, B., Alfonso Almirall, R., Sabina Molina, D., and Cruz Navarro, J. (2012). Manual para diagnóstico y tratamiento de trastornos ansiosos. *MediSur*, 10:466 – 479.

Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63:3–42.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. The MIT Press, final edition edition.

Géron, A. (2023). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc., 3 edition.

H. Andrew, S., Eichstaedt, J. C., Kern, M. L., Dziurzynski, L., Ramones, S. M., Agrawal, M., Shah, A., Kosinski, M., Stillwell, D., Seligman, M. E. P., and Ungar, L. H. (2013). Personality, gender, and age in the language of social media: The open-vocabulary approach. *PLOS ONE*.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. Springer Series in Statistics. Springer, 2nd ed. 2009. corr. 3rd printing 5th printing. edition.

Heri Cahyana, N., Saifullah, S., Fauziah, Y., Sasmito Aribowo, A., and Drezewski, R. (2022). Semi-supervised Text Annotation for Hate Speech Detection using K-Nearest Neighbors and Term Frequency-Inverse Document Frequency. *International Journal of Advanced Computer Science and Applications*, 13:147–151.

IBM (s.f.). International business machines (IBM). What are Naïve Bayes classifiers? Disponible: `https://www.ibm.com/topics/naive-bayes` Recuperado: 12/02/2024.

Instituto Nacional de Estadística y Geografía (INEGI) (2021). Instituto Nacional de Estadística y Geografía (INEGI). Resultados de la primera encuesta nacional de bienestar autorreportado (ENBIARE). Disponible: `https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2021/EstSociodemo/ENBIARE_2021.pdf` Recuperado: 18/09/2023.

Isabelle, G. (2006). *Feature Extraction Foundations and Applications. Pattern Recognition*. Springer.

Jurafsky, D. and Martin, J. H. (2008). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, 2 edition.

Kelleher, B. M. N. . A. D. . J. D. (2020). *Fundamentals of Machine Learning for Predictive Data Analytics : Algorithms, Worked Examples, and Case Studies*. MIT Press, 2 edition.

Kuhn, M. and Johnson, K. (2019). *Feature Engineering and Selection: A Practical Approach for Predictive Models*. Chapman and Hall/CRC Data Science Series. Chapman and Hall/CRC, 1 edition.

Liu, H. and Motoda, H. (1998). *Feature Selection for Knowledge Discovery and Data Mining*. The Springer International Series in Engineering and Computer Science. Springer, 1 edition.

LIWC (2023). Welcome to liwc-22. Disponible: `https://www.liwc.app/` Recuperado: 22/04/2024.

Manning, C. D., Raghavan, P., and Schutze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

Mayo Clinic (2021). Trastornos de ansiedad. Disponible: `https://www.mayoclinic.org/es/diseases-conditions/anxiety/symptoms-causes/syc-20350961` Recuperado: 22/09/2023.

Moore, D. S., McCabe, G. P., and Craig, B. (2007). *Introduction to the Practice of Statistics*. W. H. Freeman, 6th edition edition.

Mueller, J. P. and Luca, M. (2021). *Machine Learning For Dummies*. For Dummies, 2 edition.

Müller, A. C. and Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media, 1 edition.

Nova, K. (2023). Machine learning approaches for automated mental disorder classification based on social media textual data. *Contemporary Issues in Behavioral and Social Sciences*, 7(1):70–83.

OMS (2014). Organización mundial de la salud (OMS). DOCUMENTOS BÁSICOS. 48° edición. Disponible: `https://apps.who.int/gb/bd/s/` Recuperado: 16/08/2023.

OMS (2017). Organización Mundial de la Salud (OMS). Depression and other common mental disorders: global health estimates. Technical documents.

OMS (2022a). Organización Mundial de la Salud (OMS). Clasificación Internacional Enfermedades (CIE) (11 ed.). Disponible: `https://icd.who.int/browse11/l-m/es#/http%3a%2f%2fid.who.int%2ficd%2fentity%2f1336943699` Recuperado: 19/08/2023.

OMS (2022b). Organización Mundial de la Salud (OMS). Clasificación Internacional Enfermedades (CIE) (11 ed.). Disponible: `https://icd.who.int/browse11/l-m/es#/http%3a%2f%2fid.who.int%2ficd%2fentity%2f334423054` Recuperado: 19/08/2023.

OMS (2022c). Organización Mundial de la Salud (OMS). Salud mental y COVID-19: datos iniciales sobre las repercusiones de la pandemia. Disponible: `https://www.who.int/es/publications/i/item/WHO-2019-nCoV-Sci_Brief-Mental_health-2022.1` Recuperado: 21/08/2023.

OMS (2022d). Organización Mundial de la Salud (OMS). Trastornos mentales. Disponible: `https://www.who.int/es/news-room/fact-sheets/detail/mental-disorders` Recuperado: 20/08/2023.

OMS y OPS (2018). Organización Mundial de la Salud y Organización Panamericana de la Salud. The Burden of Mental Disorders in the Region of the Americas. . Disponible: `https://iris.paho.org/handle/10665.2/49578`. Recuperado: 18/09/2023.

OMS y OPS (2021). Organización Mundial de la Salud y Organización Panamericana de la Salud. Boletín Desastres N.131.- Impacto de la pandemia COVID-19 en la salud mental de la población. Disponible: https://www.paho.org/es/boletin-desastres-n131-impacto-pandemia-covid-19-salud-mental-poblacion#:~:text=Las%20mujeres%2C%20los%20j%C3%B3venes%2C%20las,entre%20los%20grupos%20m%C3%A1s%20afectados. Recuperado: 18/09/2023.

Pennebaker, J. and Tausczik, Y. (2010). The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of Language and Social Psychology*, 29:24–54.

Rabiner, L. and Juang, B.-H. (1993). *Fundamentals of speech recognition*. PH, ph edition.

Reynolds, C. R., Richmond, B., and Lowe, P. (2007). *Escala de Ansiedad Manifiesta en Adultos (AMAS)*. El Manual Moderno.

Saha, S. (2022). Students anxiety and depression dataset. kaggle. Disponible: https://www.kaggle.com/datasets/sahasourav17/students-anxiety-and-depression-dataset/ Recuperado: 10/08/2023.

Saifullah, S., Fauziyah, Y., and Sasmito Aribowo, A. (2021). Comparison of machine learning for sentiment analysis in detecting anxiety based on social media data. *Jurnal Informatika*, 15(1):45.

Sarkar, D. (2016). *Text Analytics with Python: A Practical Real-World Approach to Gaining Actionable Insights from your Data*. Apress, 1st ed. edition.

Schlkopf, B. and Smola, A. J. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Adaptive Computation and Machine Learning. The MIT Press, 1st edition.

Tasnim, M., Ehghaghi, M., Diep, B., and Novikova, J. (2023). Depac: a corpus for depression and anxiety detection from speech.

Thanaki, J. (2017). *Python Natural Language Processing: Advanced machine learning and deep learning techniques for natural language processing*. Packt Publishing, 1 edition.

Theobald, O. (2017). *Machine Learning A Visual Starter Course*. Scatterplot Press, 1 edition.

Theobald, O. (2020). *Machine Learning for Absolute Beginners: A Plain English Introduction (Third Edition)*. Scatterplot Press.

Wade, C. (2020). *Hands-On Gradient Boosting with XGBoost and scikit-learn: Perform accessible Python machine learning and extreme gradient boosting with Python*. PACKT Publishing LTD.

Yu, Y., Li, Q., and Liu, X. (2023). Automatic anxiety recognition method based on microblog text analysis. *Frontiers in Public Health*, 11.

Zheng, A. (2015). *Evaluating machine learning models : a beginner's guide to key concepts and pitfalls*. O'Reilly Media, first edition. edition.

Apéndice A

Instrumento: Detección de rasgos de ansiedad en estudiantes

Este formulario cuenta con 4 partes, la primera se compone de un aviso de privacidad, en caso de pasar a la siguiente sección se responderán 49 preguntas especializadas que permiten la identificación de rasgos ansiedad en estudiantes universitarios y finalmente se solicita redactar un texto libre acerca de una imagen. Es importante que sus respuestas sean lo más objetivas posibles para obtener datos adecuados que ayuden en el desarrollo de una investigación enfocada en la detección de ansiedad mediante aprendizaje automático.

A.1. Aviso de privacidad

La información recuperada será utilizada con fines académicos para estudiar los rasgos de ansiedad en estudiantes. No se solicita información personal que pueda identificar a los participantes. Todos los datos se adquieren de manera anónima y serán utilizados de forma responsable. Para cualquier duda o aclaración enviar un correo a cicb990222@gs.utm.mx . Si estas de acuerdo con lo anterior da siguiente para comenzar con el formulario.

A.2. Reactivos sobre datos demográficos

A continuación se muestran 4 preguntas para ayudar a la clasificación de los resultados. (No se preguntan datos personales, como Nombre, Dirección o numero telefónico).

- ¿Qué edad tienes?.
- ¿Cómo te identificas en términos de género?.
- ¿Has recibido algún diagnóstico o tratamiento previo relacionado con la ansiedad u otros trastornos mentales?
- ¿Estás trabajando actualmente para financiar tus estudios o tus gastos educativos?.

A.3. Reactivos de AMAS-C

Esta sección cuenta con 49 preguntas para detectar rasgos de ansiedad en estudiantes universitarios.

1. Pareciera que los demás hacen cosas con mayor facilidad que yo.
2. Me preocupo demasiado por las pruebas o exámenes.
3. Siento que a los demás les desagrada la forma en que hago las cosas.
4. Me cuesta trabajo tomar decisiones.
5. Tengo problemas para conciliar el sueño la noche anterior a una prueba importante.
6. Estoy preocupado(a) gran parte del tiempo.
7. Me preocupo incluso por las pruebas breves y poco importantes.
8. Siempre soy amable.
9. Siento que alguien me va a decir que hago las cosas mal.
10. Siempre soy educado(a).
11. No importa cuánto estudie para un examen, nunca es suficiente.
12. Los demás son más felices que yo.
13. Me preocupa lo que otros piensen de mi.
14. Resolver una prueba se me dificulta más a mi que a los demás.
15. Me preocupa hacer lo correcto.
16. Siempre soy bueno(a).
17. En la mayoría de los exámenes, espero que mi calificación será peor de lo que resulta ser.
18. Me preocupa lo que vaya a suceder.
19. Es frecuente que se me describa como intranquilo(a).
20. Me es difícil concentrarme en mis estudios.
21. Siempre soy agradable con todos.
22. Es fácil que hieran mis sentimientos cuando me llaman la atención.
23. Sin importar cuánto estudie para un examen, de todos modos me siento nervioso(a).
24. Siempre digo la verdad.
25. Me pongo nervioso(a) cuando las cosas no resultan bien para mi.
26. Con frecuencia me siento solo(a) cuando estoy con otras personas.
27. Odio tener exámenes.
28. Nunca me enojo.
29. Me preocupa cómo me está yendo en mis estudios.
30. Me preocupo cuando me acuesto a dormir.
31. Es frecuente que me sienta enfermo(a) antes de una prueba.
32. Estoy nervioso(a).

33. Los exámenes me ponen nervioso(a).
34. Con frecuencia me siento inquieto(a).
35. Me preocupa el futuro.
36. Mis músculos se sienten tensos.
37. Después de un examen, me siento preocupado(a) hasta que me entero de mi resultado.
38. Me preocupo mucho por el pasado.
39. Me siento nervioso(a) cuando tengo una prueba, incluso si estoy bien preparado(a).
40. Es frecuente que me sienta acelerado(a) o intranquilo(a).
41. Siempre me preocupo por las pruebas o exámenes.
42. Me siento solo(a) aun cuando estoy acompañado(a) por otras personas.
43. Es fácil que hieran mis sentimientos.
44. A veces me preocupo tanto por una prueba que me duele la cabeza.
45. Es frecuente que sienta mi cuerpo tenso.
46. Es frecuente que me sienta cansado(a).
47. En ocasiones noto que mi corazón late con mucha rapidez.
48. Me simpatizan todas las personas que conozco.
49. En ocasiones me preocupo acerca de cosas que en realidad no tienen importancia.

A.4. Descripción de imagen

En esta sección se deberá describir la imagen que se le presenta con aproximadamente 300 palabras.

50. Describe con tus propias palabras lo que esta sucediendo en la siguiente imagen en 300 palabras como mínimo o en su defecto con 1300 caracteres.

Apéndice B

Características seleccionadas

El Apéndice B contiene un registro de las características seleccionadas tras aplicar procesos de selección en los métodos propuestos de preprocesamiento I y III para el aprendizaje supervisado, así como en los tres métodos planteados para el aprendizaje semisupervisado. Este análisis tiene como objetivo mejorar el rendimiento de los modelos de aprendizaje automático.

B.1. Características seleccionadas en el método de preprocesamiento II

En este apartado se presentan las características seleccionadas mediante los algoritmos de Árboles Aleatorios, Árboles Extremadamente Aleatorios y el método ANOVA, obtenidas en el segundo método de preprocesamiento de aprendizaje supervisado. A continuación, se mencionan las características que conforman los conjunto de datos basados en los métodos de extracción de características aplicados: Bolsa de Palabras en su versión de unigramas (BoW1G), bigramas (BoW2G) y trigramas (BoW3G), ademas del método TF-IDF.

- **Características seleccionadas del conjunto BoW1G**

Las características extraídas mediante el método de Bolsa de Palabras en su versión de unigramas obtuvo un total de 6,089, de las cuales se seleccionaron 2,030. En los cuadros de texto siguientes se muestran segmentos de las primeras 25 características de mayor importancia, determinadas por los algoritmos de Árboles Aleatorios, Árboles Extremadamente Aleatorios y el método ANOVA.

Árboles Aleatorios

modo, ve, haciendo, llevan, necesitan, quer, alado, madre, primera, se, esten, grifo, de, utensilio, que, identificar, asi, mutuamente, mujer, disfrutara, aceite, ha, probablemente, limpio, infelicidad, ...

Árboles Extremadamente Aleatorios

alacenas, ta, riesgo, espacio, resolver, repisa, detalle, desperdicie, problema, cortarse, madera, percatan, tienen, buro, piernas, yendo, importarle, adultos, preste, recetas, demuestra, bebidas, suela, gustan, rotas, ...

ANOVA

abandon, abandonar, abierta, abriendo, abrir, abrirla, absten, absurdo, aburre, aburri, acabaron, acabo, acartonadas, acci, accidente, accidentes, acciones, aceite, aceptaran, acercar, acomodados, acomodar, aconsejaron, acostada, acostumbrado, ...

- **Características seleccionadas del conjunto BoW2G**

El método de Bolsa de Palabras en su versión de bigramas generó un total de 34,938 características, de las cuales se seleccionaron 11,646. A continuación, se muestra un extracto con las 25 características de mayor relevancia de acuerdo con los algoritmos de Árboles Aleatorios, Árboles Extremadamente Aleatorios y el método ANOVA.

Árboles Aleatorios

{adem, de}, {tambi, podemos}, {pero, pudo}, {preocupado, de}, {se, vea}, {pero, est}, {adem, el}, {libros, en}, {que, veo}, {el, esta}, {observar, la}, {con, lo}, {parecer, se}, {peque, se}, {cocinar, el}, {cara, de}, {ganas, de}, {con, estar}, {no, tiene}, {una, hoja}, {veo, la}, {desorden, en}, {cara, no}, {sarten, con}, {vista, se}, ...

Árboles Extremadamente Aleatorios

{igual, espero}, {parece, preocupado}, {nada, para}, {preocupado, de}, {es, su}, {verse, como}, {pared, hay}, {sarten, sin}, {esta, situaci}, {siento, no}, {manos, sucias}, {esta, desordenado}, {planta, grande}, {al, rev}, {al, contrario}, {paliado, la}, {esta, sentada}, {atenci, lo}, {mientras, tanto}, {ayudando, cocinar}, {sala, hay}, {adem, el}, {aunque, parece}, {junto, un}, {aparte, del}, ...

ANOVA

{abajo, en}, {abajo, puede}, {abandon, una}, {abandonar, el}, {abierta, adentro}, {abierta, ah}, {abierta, al}, {abierta, cuando}, {abierta, del}, {abierta, el}, {abierta, en}, {abierta, eso}, {abierta, hay}, {abierta, la}, {abierta, los}, {abierta, parece}, {abierta, tanto}, {abierta,tienen}, {abierta, una}, {abiertas, cerradas}, {abiertas, con}, {abiertas, de}, {abiertas, hay}, {abiertas, porque}, {abierto, con}, ...

- **Características seleccionadas del conjunto BoW3G**

El conjunto de características extraídas con Bolsa de Palabras en formato de trigramas obtuvo 66,140 características, de las cuales se seleccionaron 22,047. En los siguientes cuadros de texto se presenta una muestra de las 25 características más destacadas, definidas por los algoritmos de Árboles Aleatorios, Árboles Extremadamente Aleatorios y el método ANOVA.

Árboles Aleatorios

{desordenada, ya, que}, {la, falta, de}, {ella, esa, tambi}, {pueda, que, alguien}, {ni, est, sentada}, {mam, la, hija}, {creo, que, es}, {verduras, la, chica}, {los, adultos, tiene}, {est, cortando, los}, {no, podr, vivir}, {haciendo, el, perro}, {ayudando, leer, el}, {de, caerse, por}, {cajon, que, podria}, {vean, como, hace}, {le, parece, preocuparle}, {esta, lo, que}, {tomates, con, un}, {siento, que, la}, {que, pueden, la}, {familia, que, se}, {podr, ser, una}, {est, cocinando, la}, {un, cuchillo, la}, ...

Árboles Extremadamente Aleatorios

{sido, un, accidente}, {no, tiene zapatos}, {la, hija, que}, {de, la, ventana}, {de, cosas, se}, {parece, una, familia}, {para, ella, debido}, {actividades, en, su}, {se, aprecia, una}, {que, en, las}, {espacio, de, convivencia}, {esta, ayudando, cocinar}, {una, cama, con}, {que, est, dentro}, {descalza, en, alg}, {mas, asumiendo, asi}, {primero, que, me}, {eso, es, lo}, {do, una, botella}, {tal, vez, sea}, {tres, integrantes, realizando}, {la, boca, tambien}, {decir, que, aunque}, {el, contorno, de}, {hay, un, costal}, ...

ANOVA

{abajo, de, esta}, {abajo, de, isla}, {abajo, en, el}, {abajo, puede, contener}, {abandon, una, magn}, {abandonar, el, dibujo}, {abierta, adentro, de}, {abierta, ah, se}, {abierta, al, fondo}, {abierta, al, lado}, {abierta, del, grifo}, {abierta, dentro, de}, {abierta, dentro, hay}, {abierta, el, humo}, {abierta, en, ella}, {abierta, eso ya}, {abierta, hay, basura}, {abierta, hay, un}, {abierta, la, llave}, {abierta, la, mam}, {abierta, la, ventana}, {abierta, lo, cual}, {abierta, lo, que}, {abierta, los, especias}, {abierta, no, hay}, ...

- **Características seleccionadas del conjunto TF-IDF**

Con el método TF-IDF se obtuvo el mismo número de características que con la Bolsa de Palabras en su versión de unigramas, es decir, un total 6,089 de las cuales se seleccionaron 2,030. A continuación se presentan las primeras 25 características de mayor importancia, determinadas por los algoritmos de Árboles Aleatorios, Árboles Extremadamente Aleatorios y el método ANOVA.

Árboles Aleatorios

cortarse, ahora, normal, ase, tiro, mujer, expresi, como, dinero, ve, forzado, esa, pas, haciendo, ser, mas, preocupado, ingredientes, el, tienen, es, se, por, cocinar, ha, ...

Árboles Extremadamente Aleatorios

resolver, empezar, repisa, cocinar, agobiado, yendo, lvenado, por, estilo, bebidas, callo, detalle, fue, gustos, notan, piernas, rompiera, no, vac, basura, llama, sentir, calientes, tranquila, distinguir, ...

ANOVA

abandon, abandonar, abriendo, abrir, abrirla, absten, absurdo, aburre, aburri, acabaron, acabo, acartonadas, accidente, accidentes, aceite, aceptaran, acercar, acomodados, acomodar, aconsejaron, acostada, acostumbrado, actitudes, actividad, acto, ...

B.2. Características seleccionadas de LIWC en el método de preprocesamiento III

En esta sección se presentan las características obtenidas mediante el método de preprocesamiento III en el contexto del aprendizaje supervisado. Este proceso incluyó la extracción de características utilizando el diccionario LIWC y como parte del método propuesto, se realizó una selección de características en dos etapas: en la primera, se calculó la correlación de Pearson y en la segunda, se aplicó el método ANOVA para mejorar aún más la selección.

- **Primer filtro de selección de características con correlación de Pearson**

Se aplicó correlación de Pearson a un total de 69 características extraídas mediante el diccionario LIWC. Tras este primer filtro, se seleccionaron 52 características. A continuación, se presenta un extracto de 25 de ellas.

BoW1G

WC, WPS, Sixltr, Funct, PronPer, TuUtd, ElElla, PronImp, Articulo, VerbAux, Pasado, Adverb, Prepos, Conjunc, Negacio, Cuantif, verbYO, verbNOS, verbosEL, verbELLOS, Subjuntiv, Humanos, Ansiedad, Enfado, Triste, ...

- **Selección de características con ANOVA**

Las características seleccionadas tras aplicar el primer filtro, basado en la correlación de Pearson, fueron procesadas mediante el método ANOVA, obteniendo un total de 22 características. En el cuadro de texto que se presenta a continuación, se detallan dichas características.

BoW2G

WPS, Articulo, VerbAux, Pasado, Negacio, Cuantif, verbosEL, verbELLOS, Humanos, Ansiedad, Enfado, Insight, Discrep, Tentat, Percept, Oir, Salud, Sexual, Tiempo, Trabajo, Logro, Comma.

B.3. Características seleccionadas en el Aprendizaje Semisupervisado

Como parte de los métodos propuestos en el enfoque de aprendizaje semisupervisado, se realizó un proceso de selección de características aplicando en método ANOVA para los tres conjuntos de datos utilizados en el entrenamiento base del método. En esta sección, se describen las características seleccionadas conforme a los métodos de extracción de características aplicados a cada conjunto de datos, los cuales incluyen Bolsa de Palabras en sus versiones de unigramas (BoW1G), bigramas (BoW2G) y trigramas (BoW3G), así como TF-IDF.

- **Conjunto de datos SocialMedia-Anxiety**

En el conjunto de datos SocialMedia-Anxiety, se extrajeron un total de 13,459 características utilizando el método de Bolsa de Palabras en su versión de BoW1G. Posteriormente, mediante el método ANOVA, se seleccionaron 4,486 características relevantes. A continuación, se presenta un extracto de 25 de estas características.

BoW1G

abajo, abandonado, abierto, abra, abraza, abrazame, abrazar, abre, abri, abrire, abrirlo, abro, absoluto, absorberlas, abuela, abuelo, acaba, acaban, acabara, acabas, acabe, acabo, accion, acciones, acechando, ...

La aplicación de Bolsa de Palabras en su versión de BoW2G resultó en un total de 46,418 características extraídas, de las cuales se seleccionaron 15,472 mediante el proceso de selección. En el cuadro de texto a continuación, se presenta un extracto con 25 de estas características.

BoW2G

{abierta, mi}, {abierto, esta}, {abraza, aunque}, {abrazado, por}, {abrazame, cuando}, {abre, tu}, {abrir, la}, {abrir, las}, {abrir, mi}, {abrir, twitter}, {abrir, twt}, {abrire, mas}, {abrirlo, seguido}, {abro, whatsapp}, {absorberlas, sin}, {abuelo, es}, {abuelo, realmente}, {aburrido, confundido}, {aburrido, de}, {aburrido, escalofriante}, {acaba, de}, {acaban, de}, {acabara, el}, {acabas, de}, {acabe, bonito}, ...

Mediante la aplicación de Bolsa de Palabras en su versión de BoW3G, se obtuvieron 59,980 características, de las cuales se seleccionaron 22,046 como resultado del proceso de selección.

A continuación, se incluye un extracto que muestra 20 de estas características.

BoW3G

{abierta, el, humo}, {abierto, la, bolsa}, {accidente, tambi, la}, {ah, no, es}, {ah, no, se}, {al, borde, de}, {al, contrario, que}, {al, costado, de}, {al, fondo, se}, {al, igual, que}, {zapatos, para, que}, {zapatos, para, tambi}, {zapatos, pod, pisar}, {zapatos, podr, ir}, {zapatos, porque, hay}, {zapatos, puestos, hay}, {zapatos, tiene, al}, {zona, de, alimentos}, {zona, de, riesgo}, {zonas, que, utilizamos}

En cuanto a las características extraídas mediante el método TF-IDF, se obtuvieron un total de 13,459, de las cuales se seleccionaron 4,486. En el cuadro de texto a continuación, se presenta un extracto de 25 de las características seleccionadas.

TF-IDF

abajo, abandonado, abdominales, abierto, abiertos, abra, abraza, abrazado, abrazame, abrazar, abre, abri, abrire, abrirlo, abro, absoluto, absorbe, absorberlas, abuela, abuelo, acaba, acabado, acaban, acabara, acabas, ...

- **Conjunto de datos UTMente-Ansiedad-Entrenamiento etiquetado Tipo B**

En el conjunto de datos UTMente-Ansiedad-Entrenamiento etiquetado Tipo B, se extrajeron 6,046 características empleando el método de Bolsa de Palabras en su versión de BoW1G. Posteriormente, se seleccionaron 2,015 características relevantes utilizando el método ANOVA. En seguida, se presenta un extracto con 25 de estas características.

BoW1G

abierta, abominacion, abre, abriendo, abrirla, abrumara, absurdo, abuelita, abuelos, acabo, accesorios, accidentarse, acciones, aceite, aceptar, aceptaran, aceptenme, acepto, acercarlos, aclarar, acomodados, acomodarlo, acompana, acompanandolos, acostumbrado, ...

Mediante el método de Bolsa de Palabras en su versión de BoW2G, se obtuvieron 34,308 características, de las cuales 11,436 fueron seleccionadas como resultado del proceso de selección. En el cuadro de texto a continuación, se muestra un extracto de 25 de estas características.

BoW2G

{abajo, con}, {abajo, parece}, {abajo, puede}, {abierta, ademas}, {abierta, al}, {abierta, desordenada}, {abierta, el}, {abierta, en}, {abierta, eso}, {abierta, esto}, {abierta, exista}, {abierta, los}, {abierta, nadie}, {abierta, parece}, {abierta, pareciera}, {abierta, podria}, {abierta, provocando}, {abierta, realmente}, {abierta, tanto}, {abiertas, al}, {abiertas, dejando}, {abiertas, hay}, {abiertas, pareciera}, {abiertas, porque},{abiertas, pueden} ...

La aplicación de Bolsa de Palabras en su versión de BoW3G generó un total de 65,177 características, de las cuales se seleccionaron 21,725 tras el proceso de selección. A continuación, se presenta un extracto con 25 de estas características.

BoW3G

{abierta, el, humo}, {abierto, la, bolsa}, {accidente, tambien, la}, {accion, que, esta}, {ademas, el, perro}, {ahi, no, es}, {ahi, no, se}, {ahi, ya, que}, {al, borde, de}, {al, contrario, que}, {al, costado, de}, {al, fondo, se}, {al, igual, que}, {al, igual, se}, {al, mi, parecer}, {al, mismo, tiempo}, {al, no, estar}, {al, no, saber}, {al, padre, la}, {al, parecer, esta}, {al, parecer, la}, {al, parecer, lo}, {al, parecer, no}, {al, parecer, por}, {al, parecer, se}, ...

Con respecto a la aplicación del método TF-IDF, se extrajeron 6,046 características, de las cuales se identificaron 2,015 como relevantes. En el cuadro de texto a continuación, se muestra un extracto con 25 de las características seleccionadas.

TF-IDF

abierta, abominacion, abre, abriendo, abrirla, abrumara, absurdo, abuelita, abuelos, acabaron, acabo, accesorios, accidentarse, acciones, aceite, aceptar, aceptaran, aceptenme, acepto, acercarlos, aclarar, acomodados, acomodarlo, acompana, acompanandolos, ...

- **Conjunto de datos UTMenteII-Ansiedad**

En el conjunto de datos UTMenteII-Ansiedad, se generaron 2,310 características a través del método de Bolsa de Palabras en su versión de BoW1G. Posteriormente, el método ANOVA permitió identificar 770 características como relevantes. A continuación, se muestra un extracto con 25 de estas características.

BoW1G

abrir, accidente, accidentes, acerca, adelante, afecte, agradable, agradables, agua, ahi, alacena, alado, alcoholica, algo, algunas, algunos, almuerzo, alrededor, alzando, anormal, ante, aparentar, aportar, aprecia, aqui, ...

Utilizando el método de Bolsa de Palabras en su variante BoW2G, se generaron 9,943 características, de las cuales 3,314 fueron seleccionadas tras el proceso de selección. En el cuadro de texto siguiente se presenta un extracto que incluye 25 de estas características en el conjunto de datos UTMenteII-Ansiedad.

BoW2G

{abierta, se}, {abierta, una}, {abierto, {accidente, por}, {accidente, ya}, {acerca, de}, {adelante, en}, {ademas, al}, {ademas, que}, {ademas, se}, {agradable, por}, {agua, en}, {agua, se}, {al, estar}, {al, fondo}, {al, igual}, {al, mismo}, {al, no}, {al, parecer}, {al, senor}, {al, ver}, {alacena, abierta}, {alacena, donde}, ...

Mediante la aplicación de Bolsa de Palabras en su versión de BoW3G, se obtuvieron 16,160 características, de las cuales se seleccionaron 5,387 por medio del método ANOVA. A continuación, se incluye un extracto que muestra 25 de estas características.

BoW3G

{abierto, el, grifo}, {accidente, en, el}, {accidente, ya, que}, {ademas, de, estar}, {agradable, por, que}, {al, fondo, se}, {al, mismo, tiempo}, {al, parecer, la}, {al, parecer, un}, {al, perro, con}, {al, ver, al}, {alado, de, la}, {algo, le, preocupara}, {algo, parecido, en}, {algunas, botellas, en}, {apreciar, una botella}, {arriba, donde, esta}, {arriba, esta, abierta}, {arriba, esta, abierto}, {atencion, es, que}, {atencion, la, comida}, {atras, de, la}, {ayudando, su, esposa}, {bolsa, que, esta}, {bolsa, que, parece}, ...

Con respecto a la aplicación del método TF-IDF, se extrajeron 2,310 características, de las cuales se identificaron 770 como relevantes. En el cuadro de texto a continuación, se muestra un extracto con 25 de las características seleccionadas del conjunto de datos UTMenteII-Ansiedad.

TF-IDF

abiertos, abrir, accidente, accidentes, acciones, acerca, acorde, actos, actuada, adecuado, adecuaran, adelante, afectaria, afecte, agobiado, agradable, agradables, agravando, agua, ahi, alacena, alado, alcoholica, alguien, algunas, ..

Apéndice C

Script de automatización para la evaluación del AMAS-C

El Apéndice C contiene el script de lenguaje de programación Python utilizado para automatizar el proceso de evaluación de las respuestas obtenidas por los participantes en la segunda sección del cuestionario aplicado para la adquisición de datos, siguiendo las reglas de evaluación establecidas por el AMAS-C.

```
data = pd.read_csv('../../../DataSets_Origen/AMAS.csv')
# Recuperacion de las respuestas
#Se selecciona unicamente las preguntas de la prueba
question1 = "1.- Pareciera que los demas hacen cosas con mayor facilidad que yo."
question49 = "49.- En ocasiones me preocupo acerca de cosas que en realidad no tienen importancia."
answers_df = data.loc[:,question1:question49]
answers_df.head()

# Preguntas de acuerdo a la categoria.
#
# |   Categoria                  |   Variable   |
# |------------------------------|--------------|
# | Inquietud/hipersencibilidad  |      ihs     |
# | Ansiedad fisiologica         |      fis     |
# | Ansiedad ante los examenes   |    examen    |
# | Preocupacion/estres social   |      soc     |
# | Mentira                      |    mentira   |
# | Ansiedad total               |      tot     |

#Estos arrays tienen los indices de las preguntas
#En la matriz de evalucion estos indices seran las i's de las coornenadas (i,j)
```

```
22 ihs = np.array([3,5,12,14,17,21,24,28,34,42,45,48])
23 fis = np.array([18,29,31,33,35,39,44,46])
24 examen = np.array([1,4,6,10,13,16,19,22,26,30,32,36,38,40,43])
25 soc = np.array([0,2,8,11,25,37,41])
26 #Si la afirmacion de una pregunta aparece en los arreglos
      anteriores
27 #Automaticamente se suma a la columna de "Ansiedad total"
28 mentira = np.array([7,9,15,20,23,27,47])
29 # Calculo de puntuaciones naturales
30 ptes_naturales = []
31 numIndividuo = answers_df.shape[0] #Numero de individuos
32 for ni in range(numIndividuo):
33         matrizEvalucion = np.zeros((49, 6)) #Matriz de evalucion
34         individuo = answers_df.iloc[ni] #Fila n de respuestas a las
               preguntas
35         individuo = np.array(individuo)
36         numRes = 0
37         for i in range(len(individuo)+1): #la i tiene el indice de
              la pregunta
38                 if i != 3:
39                         valor = individuo[numRes]
40                         numRes=numRes+1
41                 if valor=='Si' and i!=3:
42                         #Se verifica a que tipo de pregunta
                              pertenece
43                         if i in ihs:
44                                 matrizEvalucion[i][0] = 1
45                                 matrizEvalucion[i][5] = 1 #Ansiedad
46                         elif i in fis:
47                                 matrizEvalucion[i][1] = 1
48                                 matrizEvalucion[i][5] = 1
49                         elif i in examen:
50                                 matrizEvalucion[i][2] = 1
51                                 matrizEvalucion[i][5] = 1
52                         elif i in soc:
53                                 matrizEvalucion[i][3] = 1
54                                 matrizEvalucion[i][5] = 1
55                         elif i in mentira: #Mentira
56                                 matrizEvalucion[i][4] = 1
57         puntuacionesNaturales = np.sum(matrizEvalucion, axis=0) #
              calculo de las puntuaciones normales (se hacen las sumas
```

```
            de las respuestas afirmativas)
58          puntuacionesNaturales = puntuacionesNaturales.tolist()
59          ptes_naturales.append(puntuacionesNaturales)
60  ptes_naturales = np.array(ptes_naturales)
61  ptes_naturales = pd.DataFrame(ptes_naturales)
62  #Nombre a las columnas del dataFrame
63  ptes_naturales.columns = ['Inquietud/hipersencibilidad','Ansiedad␣
        fisiologica','Ansiedad␣ante␣los␣examenes','Preocupaciones␣
        sociales/estres','Mentira','Ansiedad␣total']
64  # Valores T
65  # Normas para universitarios - Inquietud/hipersencibilidad (IHS)
66  tablaEvalucionIHS = {0:[28,1],1:[32,4],2:[36,8],3:[39,16],
        4:[43,27],5:[46,37],6:[50,47],7:[53,59],8:[57,72],
        9:[60,83],10:[64,92],11:[67,97],12:[71,99]}
67  # Normas para universitarios - Ansiedad ante los examenes (Examen)
68  tablaEvalucionExamen =  {0:[37,5],1:[40,15],2:[42,27],3:[45,38],
        4:[48,49],5:[51,59],6:[53,66],7:[56,73],8:[59,80],9:[62,85],
        10:[65,89],11:[67,93],12:[70,96],13:[73,98],14:[76,99],
        15:[78,100]}
69  # Normas para universitarios - Ansiedad fisiologica (FIS)
70  tablaEvalucionFIS = { 0:[39,12], 1:[44,34], 2:[48,52], 3:[53,67],
        4:[58,78], 5:[62,85], 6:[67,91], 7:[71,97], 8:[76,99] }
71  # Normas para universitarios - Preocupacion / estres (SOC)
72  tablaEvalucionSOC = { 0:[40,16], 1:[46,44], 2:[52,65], 3:[58,79],
        4:[64,88], 5:[69,94], 6:[75,97], 7:[81,99] }
73  # Normas para universitarios - Mentira
74  tablaEvalucionMentira = { 0:[35,7], 1:[40,20], 2:[45,34],
        3:[49,49], 4:[54,61], 5:[59,77], 6:[63,91], 7:[68,98] }
75  # Normas para universitarios - Ansiedad total (TOT)
76  tablaEvalucionTOT = { 0:[32,0], 1:[33,1],2:[34,2],3:[35,4],
        4:[37,6],5:[38,10],6:[39,13],7:[40,17],8:[41,21],9:[43,26],
        10:[44,31],11:[45,36],12:[46,41],13:[48,45],14:[49,50],
        15:[50,55],16:[51,60],17:[53,65],18:[54,70],19:[55,73],
        20:[56,76],21:[58,79],22:[59,81],23:[60,83],24:[61,86],
        25:[63,88],26:[64,89],27:[65,91],28:[66,93],29:[68,94],
        30:[69,95],31:[70,96],32:[71,97],33:[73,97],34:[74,98],
        35:[75,98],36:[76,99],37:[77,99],38:[79,100],39:[80,100],
        40:[81,100],42:[84,100] }
77  # Calculo de valores T
78  valoresT = []
79  numIndividuo = ptes_naturales.shape[0] #Numero de individuos
```

```
for ni in range(numIndividuo):
        listValT = []
        individuo = ptes_naturales.iloc[ni] #Fila n de respuestas a
            las preguntas
        individuo = np.array(individuo)
        for i in range(len(individuo)): #la i tiene la subcategoria
                valor = individuo[i]
                #Busqueda del valor T correspondiente en cada tabla
                    de las subcategorias
                if i == 0: #IHS
                        T = tablaEvalucionIHS[valor][0]
                        listValT.append(T)
                elif i == 1: #FIS
                        T = tablaEvalucionFIS[valor][0]
                        listValT.append(T)
                elif i == 2: #EXAMEN
                        T = tablaEvalucionExamen[valor][0]
                        listValT.append(T)
                elif i == 3: #SOC
                        T = tablaEvalucionSOC[valor][0]
                        listValT.append(T)
                elif i == 4: #MENTIRA
                        T = tablaEvalucionMentira[valor][0]
                        listValT.append(T)
                else: #TOT
                        if valor == 41:
                                valor = 42 #LA PSICOLOGA LO
                                    CONSIDERO ASI
                        T = tablaEvalucionTOT[valor][0]
                        listValT.append(T)
        valoresT.append(listValT)
valoresT = np.array(valoresT)
valoresT = pd.DataFrame(valoresT)
#Nombre a las columnas del dataFrame
valoresT.columns = ['Inquietud/hipersencibilidad','Ansiedad␣
   fisiologica','Ansiedad␣ante␣los␣examenes','Preocupaciones␣
   sociales/estres','Mentira','Ansiedad␣total']
```