UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

CENTRO UNIVERSITARIO UAEM VALLE DE MÉXICO

Identificación de lenguaje ofensivo en textos en español utilizando técnicas de aprendizaje supervisado y lexicones

T E S I S

Que para obtener el Título de

MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

P r e s e n t a

Ing. Jesús Donovan Maldonado Mondragón

Tutor Académico:

Dr. Asdrúbal López Chau

Tutor(es) Adjunto(s):

Dra. Maricela Quintana López

Dr. Saturnino Job Morales Escobar

VALLE DE MÉXICO

Atizapán de Zaragoza, Edo. de Méx. Enero 2024

Resumen

En la era de las redes sociales, se ha observado un aumento en la expresión de odio y discriminación en línea, generando riesgos psicológicos y físicos para individuos o grupos específicos. Este fenómeno ha impulsado la necesidad de detectar y abordar el lenguaje ofensivo en estos entornos. Actualmente, las validaciones manuales y los sistemas diseñados principalmente para el idioma Inglés abordan esta problemática, pero hay una falta de enfoques específicos para otros idiomas, incluido el Español.

Esta tesis se centra en desarrollar un método eficaz y robusto para detectar lenguaje ofensivo en Español dirigido a la comunidad LGBTIQ+, utilizando seis métodos de aprendizaje supervisado: Neural Network, Decision Tree, Support Vector Machine, Naive Bayesian Classifier, Logistic Regression y Random Forest. El desafío de desequilibrio de clases en datos web se aborda mediante un corpus de 6,716 documentos de Twitter, previamente etiquetados y seleccionados de un conjunto más extenso, junto con técnicas de vectorización de bolsa de palabras y un lexicón de polaridad personalizado.

Este enfoque combina la capacidad de aprendizaje supervisado con un lexicón específico para la comunidad LGBTIQ+, capturando las complejidades y matices del lenguaje ofensivo en este contexto. La investigación se enfoca en mejorar la capacidad de detectar comentarios ofensivos en línea, ofreciendo una contribución significativa al abordar la expresión de odio en plataformas digitales.

Abstract

In the era of social media, there has been an observed increase in the expression of hatred and discrimination online, posing psychological and physical risks for specific individuals or groups. This phenomenon has driven the need to detect and address offensive language in these environments. Currently, manual validations and systems designed primarily for English language address this issue, but there is a lack of specific approaches for other languages, including Spanish.

This thesis focuses on developing an effective and robust method to detect offensive language targeted at the LGBTIQ+ community in Spanish. It employs six supervised learning methods: Neural Network, Decision Tree, Support Vector Machine, Naive Bayesian Classifier, Logistic Regression, and Random Forest. The challenge of class imbalance in web data is addressed through a corpus of 6,716 Twitter documents, previously labeled and selected from a larger set, along with bag-of-words vectorization techniques and a customized polarity lexicon.

This approach combines the power of supervised learning with a lexicon tailored to the LGBTIQ+ community, capturing the complexities and nuances of offensive language in this context. The research aims to enhance the ability to detect offensive comments online, providing a significant contribution to addressing the expression of hatred on digital platforms.

Índice

Índice de figuras

Índice de tablas

Índice de ecuaciones

CAPÍTULO 1

1. Introducción

El surgimiento de las redes sociales supuso una revolución en el mundo digital actual y la forma de comunicarse. Estas plataformas nacieron con el propósito de conectar a las personas, permitiéndoles a su vez expresar y compartir ideas sobre diferentes puntos de vista públicamente.

Sin embargo, la libertad de los usuarios de poder expresarse libremente en internet provoca que en ocasiones se generen publicaciones y comentarios de odio hacia otras personas, fomentando el racismo y/o la discriminación hacia un individuo o grupo en específico (Jha et al., 2020). Esto puede generar varios riesgos físicos y psicológicos, que pueden derivar en depresión, ansiedad o incluso situaciones de riesgo para la vida.

Por lo anterior, surge la necesidad de detectar comentarios en redes sociales con lenguaje que incite al odio. Actualmente, algunas de las validaciones para identificar esto se implementan manualmente según el juicio subjetivo de revisores humanos. Otras validaciones se realizan con sistemas diseñados principalmente para el idioma Inglés (Jha et al., 2020) (Pronoza et al., 2021b) .

Existen varios métodos para detectar el discurso de odio, comúnmente basados en técnicas de aprendizaje automático; para los métodos de aprendizaje no supervisado, es importante mencionar el Procesamiento del Lenguaje Natural (Sengupta et al., 2022) y los algoritmos basados en léxico (Bashar et al., 2021), aunque son algunos de los más utilizados, gran parte de los sistemas diseñados con ellos están basados en el idioma Inglés, y hay significativamente menos trabajos para otros idiomas.

Los métodos de detección de lenguaje ofensivo basados en algoritmos de aprendizaje supervisado se ven afectados por el desequilibrio de clases cuando se

entrenan con datos de la web. Esto se debe a que los comentarios analizados son en un gran porcentaje comentarios sin relevancia o relación en los datos recopilados (etiquetados con la clase Neutral), lo que afecta aún más el rendimiento de los modelos predictivos.

Con base en lo anterior, en esta tesis se presenta un análisis comparativo del desempeño de seis métodos de aprendizaje supervisado para la detección de lenguaje ofensivo en Español dirigido a la comunidad LGBTIQ+. Este análisis se realiza a partir de un corpus de 6,716 documentos obtenidos de la plataforma Twitter previamente etiquetados y seleccionados de un corpus total de 126,000 tuits mediante un muestreo aleatorio simple sin reemplazo. Los clasificadores elegidos para la comparación son los más utilizados en el área de análisis de sentimientos: Neural Network (NN), Decision Tree (DT), Support Vector Machine (SVM), Naive Bayesian Classifier (NB), Logistic Regresión (LR) y Random Forest (RF). Estos métodos se combinan con un enfoque de vectorización de bolsa de palabras y un lexicón de polaridad generado a partir de datos reales de redes sociales del mismo corpus obtenido.

El objetivo principal de esta investigación es desarrollar un método eficaz y robusto para identificar y clasificar el lenguaje ofensivo dirigido a la comunidad LGBTIQ+. Se propone una combinación de métodos de aprendizaje supervisado y un lexicón personalizado que se adapte específicamente a este dominio. A través de este enfoque se busca mejorar la capacidad de detectar y abordar los comentarios ofensivos en línea.

El enfoque de aprendizaje supervisado permite entrenar algoritmos utilizando un conjunto de datos etiquetados, donde se identifican y clasifican los comentarios ofensivos relacionados con la comunidad LGBTIQ+. Estos comentarios se recopilan de manera exhaustiva y se etiquetan de manera precisa considerando las diversas formas en que puede manifestarse el discurso de odio en línea.

Además, se desarrolla un lexicón personalizado que contiene términos y expresiones específicas utilizadas en el contexto de la comunidad LGBTIQ+. Este

lexicón se construye a partir del mismo corpus con el fin de capturar de manera precisa las características y matices del lenguaje ofensivo relacionado con la identidad de género y orientación sexual.

1.1 Antecedentes

El número de personas que utilizan internet diariamente se ha incrementado vertiginosamente en los últimos años. De acuerdo con el estudio realizado por Simon Kemp para el portal DataReportal (Kemp, 2022), el número de usuarios activos en internet ascendió a 4.95 billones a inicios de 2022, lo que representa el 62.5% de la población mundial.

Las personas del siglo XXI han interactuado la mayor parte de su vida en un mundo habilitado para funcionar con internet y socializar a través de él. La facilidad de comunicarse con tan solo un clic ha convertido a las redes sociales como Facebook, Twitter o Instagram entre otras, en un medio muy popular para expresar opiniones sobre eventos, personalidades, o productos de forma pública.

Desafortunadamente, muchos usuarios que interactúan en las redes sociales a menudo corren el riesgo de ser atacados a través del uso de lenguaje ofensivo (Perera & Fernando, 2021), lo cual afecta a las personas en diferentes aspectos, no sólo en cuestiones de carácter físico sino también en problemas psicológicos como la depresión e incluso generar situaciones que pueden atentar contra la vida.

El concepto de lenguaje ofensivo se puede entender en más de un sentido, en (Erjavec & Kovačič, 2012) se define al lenguaje ofensivo como cualquier tipo de comunicación que sea de carácter abusivo, insultante, intimidante, acosador o que incite a la violencia o discriminación hacia alguna persona o grupo vulnerable debido a su etnia, género, orientación sexual, religión u otra característica. Por otro lado, en (Kucuk, 2016) se describe al lenguaje ofensivo como cualquier hostilidad o violencia que se genere hacia alguna persona o grupo debido a su raza, religión u otros factores, enfocándose principalmente en los grupos de minorías raciales, étnicas, religiosas y culturales, las mujeres o la comunidad LGBTIQ+ (término

formado por las siglas de las palabras Lesbiana, Gay, Bisexual, Transgénero, Intersexual y Queer).

La generación de mensajes con lenguaje ofensivo por parte de los usuarios en redes sociales crea la necesidad de la implementación de servicios para su detección. Cuando una red social da libertad a los usuarios para expresar sentimientos, puntos de vista, experiencias, etc., el mayor problema que se genera radica potencialmente en el abuso del lenguaje para realizar ofensas, promover el racismo (Jha et al., 2020), discriminar por pertenecer a algún grupo étnico (Pronoza et al., 2021b), entre otros.

Aunque detectar malas palabras u ofensas parece ser una tarea sencilla, en realidad no es fácil; por ejemplo, existen idiomas que contienen una serie de símbolos o que mezclan palabras con otros idiomas (como en el caso del spanglish) que dificultan esta tarea (Sreelakshmi et al., 2020). En otros la estructura gramatical no es tan simple como la del idioma Inglés, para estos casos existen varias propuestas de solución para la detección de lenguaje ofensivo, como por ejemplo (Kanclerz et al., 2020), donde se trabajó con un modelo que combina varias técnicas para detección de sentimientos en múltiples idiomas como el polaco, ruso o portugués entre otros.

Asimismo, la manera de escribir interviene en gran medida en el significado del texto, el sarcasmo y la ironía por ejemplo son dos temas que pueden influir en esto (Sulis et al., 2016). El sarcasmo, por una parte, se refiere a una expresión de forma cruel sobre una situación en específico, y la ironía se define más concretamente como burla disimulada. Este tipo de textos son uno de los grandes problemas que enfrentan los algoritmos actuales, sobre los cuales se han desarrollado diversos modelos que hasta el momento han alcanzado un índice de aceptación, sin embargo, aún se encuentra alejado de una identificación “perfecta” de sarcasmo (Ortega-Bueno et al., 2022).

A través de los años, se han desarrollado diversas técnicas para detección de lenguaje ofensivo, como por ejemplo, la detección de palabras de carácter ofensivo (Del Bosque & Garza, 2014), donde se emplean técnicas de lenguaje

natural y enfoques supervisados realizando un etiquetado manual, las cuales han mostrado buenos resultados; de igual forma podemos mencionar técnicas de aprendizaje no supervisado, donde se emplean el uso de diccionarios que no requieren datos etiquetados para su funcionamiento (Mubarak et al., 2017). Sin embargo, estos últimos se encuentran limitados, ya que continuamente se añaden nuevas palabras, agresiones o modismos al lenguaje cotidiano, lo que provoca la constante actualización de estos, de otra manera se verían comprometidos en su funcionamiento.

De igual forma los algoritmos que implementan técnicas de aprendizaje supervisado, se ven afectados por la calidad de la información, ya que, en el etiquetado de los datos, se enfrentan a la clasificación de forma subjetiva, debido a que el criterio del etiquetador está ligado a aspectos sociales, culturales y creencias (Park & Fung, 2017).

Con base en lo anterior, se puede afirmar que la detección de textos ofensivos se ve influenciada en los dos aspectos mencionados: la constante evolución del lenguaje, y el etiquetado de la información; por lo tanto, el presente trabajo propone un método mixto que planea involucrar ambos conceptos. Por una parte se utilizarán técnicas de aprendizaje supervisado, para lo cual se requerirá de un pre-procesamiento, extracción de características y etiquetado de la información; y por otra parte, se usará un lexicón de propósito específico (diccionario de ofensas) para mejorar la identificación de lenguaje ofensivo. Estas herramientas ya han sido utilizadas anteriormente por varios autores, por ejemplo,(Tiara et al., 2015); sin embargo, dichos trabajos se enfocan a idiomas diferentes al Español, por lo que no pueden ser aplicados directamente al Español de México.

Debido a que no se cuenta con un lexicón en Español orientado al contexto a utilizar en este proyecto (Lenguaje Ofensivo dirigido a la comunidad LGBTIQ+), se realizará la creación de éste.

1.2 Planteamiento del Problema

El lenguaje ofensivo contra grupos de personas, empresas, instituciones o el gobierno mismo puede conducir a consecuencias de diversos tipos, y estas últimas pueden ser de mayor magnitud cuando los discursos de odio son difundidos de manera masiva, como en las redes sociales. Ejemplos de las consecuencias mencionadas son: incitar a la violencia contra ciertas personas o grupos, llevar a la quiebra a una empresa, o provocar revueltas sociales. La identificación automática del uso de este tipo de lenguaje tiene tanta importancia, que actualmente es un tema de investigación muy activo en la comunidad científica del área de computación, por ejemplo, se pueden mencionar los trabajos de (Eronen et al., 2021) (Lin et al., 2022) (Liu et al., 2022) (Dhungana Sainju et al., 2021) (Florio et al., 2020) (Perifanos & Goutsos, 2021) (Mohapatra et al., 2021) (Fang et al., 2021).

En la revisión de la literatura sobre la identificación automática de lenguaje ofensivo en textos, se encontró que la mayoría de los métodos propuestos están diseñados para otros idiomas (principalmente el Inglés); sin embargo, para el Español existe una cantidad mucho menor de trabajos realizados y la aplicación de las técnicas actuales para identificación de lenguaje ofensivo, diseñadas para otros idiomas, no funciona correctamente para el idioma Español debido a la estructura del lenguaje, y al uso de regionalismos, palabras inglesas, expresiones usadas en redes sociales, etc. El simple uso de un traductor Español a Inglés, o la aplicación directa de técnicas diseñadas específicamente para otro idioma no es suficiente para identificar lenguaje ofensivo correctamente.

Por otra parte, resulta importante mencionar que, para el análisis de sentimientos, se han generado lexicones específicos (por ejemplo, SenticNet, Bin, Affin, etc.), mismos que son usados actualmente como herramienta para identificar la polaridad (con valores positivo, negativo o neutral) de textos. Estos lexicones fueron creados considerando el contexto de los textos, y el lenguaje usado en redes sociales. Un enfoque similar podría servir para identificación de lenguaje ofensivo. Por lo tanto, la creación de este lexicón en Español es un área de oportunidad que se atiende en esta investigación.

1.3 Pregunta de Investigación

¿Es posible mejorar la precisión en la detección de lenguaje ofensivo en Español de México en un contexto en específico combinando métodos de aprendizaje supervisado con lexicones?

1.4 Objetivos

1.4.1 Objetivo General

Diseñar un método para la identificación automática de lenguaje ofensivo hacia miembros de la comunidad LGBTIQ+ en textos escritos en idioma Español, basado en técnicas de aprendizaje supervisado y un lexicón creado específicamente para este propósito.

1.4.2 Objetivos Específicos

1. Crear un corpus con documentos en Español que contengan lenguaje ofensivo hacia la comunidad LGBTIQ+, recolectados de la plataforma Twitter.
2. Realizar pre-procesamiento y etiquetado a los elementos del corpus como paso previo a la extracción de características. Para el etiquetado, se generarán reglas específicas al caso de estudio de esta investigación.
3. Extraer características altamente discriminativas entre los documentos pre-procesados, con la finalidad de vectorizar los datos no estructurados.
4. Generar y evaluar modelos predictivos para identificar lenguaje ofensivo en textos en Español y determinar cuáles modelos ofrecen un mejor desempeño.
5. Proponer una metodología para la creación de un lexicón orientado a la identificación de lenguaje ofensivo.
6. Adaptar las técnicas basadas en lexicón de análisis de sentimientos para identificar lenguaje ofensivo usando el lexicón generado.

7. Combinar los mejores modelos predictivos y la técnica basada en lexicón para la identificación de lenguaje ofensivo, empleando un enfoque de ensamble.

1.5 Hipótesis

Es posible diseñar e implementar un método para identificación de lenguaje ofensivo en idioma Español de México para su aplicación en mensajes en contra de la comunidad LGBTIQ+, basado en técnicas de aprendizaje automático y lexicones, el cual tendrá una exactitud de clasificación igual o superior a otros métodos.

1.6 Justificación

Las agresiones en las redes sociales son un tema de preocupación en la actualidad debido a que el daño que provoca es cada vez más significativo. Twitter es una de las redes sociales más utilizadas por su simplicidad. Según indican las mediciones sobre el uso de esta red social, tan solo en 2020 fueron escritos más de 500 millones de Tuits diariamente (Stats, 2021).

Como se ha mencionado, algunos usuarios utilizan las redes sociales no sólo para interactuar, sino también para hacer un uso mal intencionado promoviendo las ofensas y textos de odio.

Diversos algoritmos se han desarrollado para la identificación de este tipo de textos en redes sociales; sin embargo, la gran mayoría se encuentran en otros idiomas como es el caso de (Pérez-Landa et al., 2021), el cual se enfoca en el idioma coreano, y de (Fortuna et al., 2021) quienes en mayor medida utilizan el idioma Inglés como referencia.

Estos algoritmos, aunque utilizan diversas técnicas de clasificación, al enfocarse en sus respectivos idiomas, no son funcionales al tratar de implementarlos con textos en el idioma Español, ya que las reglas gramaticales y palabras son completamente diferentes.

La construcción de un método basado en técnicas de aprendizaje supervisado en el idioma Español ayudará a categorizar mensajes en este idioma e identificar el sentimiento generado, con lo que se podrán considerar tres clases de mensajes: a) mensajes ofensivos, b) mensajes no ofensivos y c) mensajes no relacionados.

Algunos autores han desarrollado modelos que incluyen el idioma Español (Pérez-Landa et al., 2021), sin embargo, dichos modelos se basan en reconocimiento de lenguaje natural, los cuales tratan de identificar la oración con una estructura general, no obstante, el idioma Español utiliza muchas palabras ambiguas, lo que representa un gran reto para ellos.

1.7 Delimitación o alcances de la investigación

- Se realizará la recolección de datos por medio de la plataforma Twitter tomando como referencia hashtags y usuarios relacionados con un contexto en específico.
- Se usarán mensajes en Español de México.
- Se utilizará únicamente el contenido de texto plano y emojis, las imágenes o videos vinculados al mensaje serán descartados.
- No se cubrirá el análisis semántico de la oración.
- Se considera que cada documento sólo contiene mensajes de un solo tipo, es decir, ofensivo, no ofensivo o neutral. No se consideran documentos con varios tipos de comentarios.
- La identificación de sarcasmo e ironía quedan fuera del alcance de este trabajo.
- La identificación de textos con palabras ambiguas como albures queda fuera del alcance de esta tesis.

1.8 Publicaciones derivadas de esta investigación

- **Ponencia:** "Comparación de métodos de aprendizaje supervisado para la detección de lenguaje ofensivo en textos en español"**, Congreso Internacional Multidisciplinario – 18 de octubre de 2022.**

- **Artículo: J. D. Maldonado Mondragón, A. López-Chau, M. Quintana López, and S. J. M. Escobar**, "Comparación de métodos de aprendizaje supervisado para la detección de lenguaje ofensivo en textos en español" ***J. CIM*****, vol. 10, no. 1, p. 2187-2198, 2022.**

- **J. D. Maldonado Mondragón, A. López-Chau, M. Quintana López, and S. J. M. Escobar**, "Augmenting TF-IDF with Lexicon Bow for improving Hate Speech Detection on Spanish language"***, en proceso.***

1.9 Organización de la tesis

I. Introducción:

En este apartado se proporciona un contexto histórico, teórico y empírico sobre el tema de estudio. Se presentan investigaciones previas relevantes, tendencias actuales y posibles vacíos de conocimiento que justifican la necesidad de realizar la presente investigación sobre la identificación de lenguaje ofensivo dirigido a la comunidad LGBTIQ+ en textos en Español.

II. Marco Teórico:

En este capítulo se exploran los conceptos clave relacionados con el tema de investigación. Se presentan definiciones, teorías y enfoques relevantes que contribuyen al entendimiento del fenómeno del lenguaje

ofensivo y su impacto en la comunidad LGBTIQ+. Además, se discuten términos y conceptos específicos que serán utilizados a lo largo del estudio.

III. Metodología:

En este capítulo se desglosa a detalle los pasos para lograr la identificación de lenguaje ofensivo en contra de la comunidad LGBTIQ+. Se describe la metodología utilizada para la obtención del corpus describiendo los pasos aplicados desde la extracción, criterios de búsqueda y procesamiento de los mensajes. Se explica el pre-procesamiento y etiquetado de documentos. Se describe la metodología para la generación de un lexicón para identificar lenguaje ofensivo en Español. Se describe el proceso completo para generar modelos predictivos de lenguaje ofensivo y su evaluación. Se aclara la forma de combinar el método basado en lexicón y los modelos predictivos para identificar lenguaje ofensivo en textos en Español

IV. Resultados:

En este capítulo se presenta un análisis exhaustivo del rendimiento del método propuesto. Se evalúa la efectividad y eficiencia del enfoque utilizado en relación con los objetivos establecidos. Se presentan los resultados obtenidos a través de experimentos y se analiza su significancia. Se incluyen métricas de evaluación y se comparan los resultados con enfoques existentes en el campo.

V. Conclusiones:

En este último capítulo se presentan las conclusiones del estudio. Se resumen las contribuciones y hallazgos clave del método propuesto. Se

discuten las implicaciones prácticas y teóricas de los resultados obtenidos. Además, se abordan las limitaciones identificadas durante el proceso y se sugieren áreas de mejora o investigación futura.

CAPÍTULO 2

2. Marco Teórico

2.1 Lenguaje Ofensivo

Las comunicaciones verbales o escritas que son consideradas como discursos de odio (Pérez-Landa et al., 2021) o que hacen uso de un lenguaje ofensivo (Sulis et al., 2016), contienen ciertas características, como presencia de groserías, vocabulario discriminatorio o despectivo o uso de sobrenombres. El objetivo principal de estos mensajes siempre es el producir un daño emocional o social al destinatario.

De manera general, se puede decir que el lenguaje ofensivo *"es aquel que utiliza expresiones discriminatorias, despectivas o groseras con el fin de realizar daño a un grupo o persona".* Para esta tesis, el lenguaje ofensivo se define como:

"En mensajes de texto en Español de México es el contenido en documentos que tienen un carácter hostil, insultante, agresivo, abusivo, intimidante, acosador, que incite a la violencia o de discriminación hacia la comunidad LGBTIQ+".

Existen otros términos estrechamente relacionados con el concepto de lenguaje ofensivo, como los mostrados a continuación:

- Bullying cibernético (Cyber bullying): el Bullying cibernético se refiere a todas las ofensas que se originan de forma continua y repetitiva a través de medios digitales. Estas ofensas pueden clasificarse en diversas categorías, como peleas en línea, acoso, solicitud de información personal para compartirla sin consentimiento, exclusión de listas de amigos, distribución no consentida de imágenes de carácter sexual, entre otras (Willard, 2007). A diferencia del lenguaje ofensivo en general, el Bullying cibernético se dirige a un objetivo específico, se produce de manera frecuente y puede darse durante un período prolongado.
- Odio cibernético (Cyber hate): se refiere a la difusión de odio a través de internet utilizando medios digitales con el objetivo de difamar, acosar, agredir

o humillar a una persona o grupo en particular (Watanabe et al., 2018). Este comportamiento se caracteriza por el uso de lenguaje ofensivo y denigrante.

- Discurso de odio (Hate speech): se refiere al uso de un lenguaje agresivo, violento y ofensivo que está dirigido a un grupo de personas en función de su género, grupo étnico, raza o religión (Watanabe et al., 2018). Esta definición es bastante similar a la del lenguaje ofensivo, ya que ambos implican el uso de expresiones que buscan denigrar o perjudicar a un grupo específico.

2.2 Aprendizaje Automático

El aprendizaje automático se puede definir como el proceso de brindar a las computadoras la capacidad de aprender mediante la enseñanza o entrenamiento, para que puedan usar este conocimiento para realizar otras tareas (Vinet & Zhedanov, 2011).

(Vemuri, 2020) define el aprendizaje automático como el proceso de solucionar problemas prácticos a través de recopilar un conjunto de datos y construir algorítmicamente un modelo estadístico basado en ellos.

Los algoritmos de aprendizaje automático tratan de aprender de los datos proporcionados, y cuantos más datos estén disponibles para aprender, estos tendrán mejor desempeño.

Los humanos son expertos en reconocer patrones complejos de manera natural; son capaces de identificar y discriminar objetos de manera visual, auditiva, táctil, además de identificar aromas y sabores. Sin embargo, para las computadoras esto es una tarea complicada. En esta tesis, se aborda el problema de identificar el uso de lenguaje ofensivo en mensajes de texto empleando métodos de aprendizaje supervisado, apoyados por una lista de palabras con ponderación, llamada lexicón. Para poder lograr esta identificación, se requiere un conjunto de textos a analizar, que en su conjunto forman un corpus. A cada elemento del corpus se le denomina documento.

Los métodos de aprendizaje automático se pueden clasificar en 3 grupos:

1. Aprendizaje supervisado. En el aprendizaje supervisado, se suministra al modelo de aprendizaje un conjunto de datos de entrenamiento etiquetados que consisten en una entrada y su respectiva respuesta correcta. El objetivo principal es que el modelo aprenda a realizar predicciones precisas sobre nuevas entradas basándose en la relación existente entre las entradas y las salidas proporcionadas en los datos de entrenamiento. Por ejemplo, si se suministra un conjunto de imágenes etiquetadas con categorías como "perro" o "gato", se espera que el modelo pueda clasificar nuevas imágenes en las categorías correctas de manera acertada (Chapelle et al., 2009).

2. Aprendizaje no supervisado. En el aprendizaje no supervisado, el modelo de aprendizaje es entrenado utilizando un conjunto de datos no etiquetados. En lugar de seguir una guía basada en una respuesta correcta, en el aprendizaje no supervisado el objetivo principal es que el modelo descubra patrones y relaciones en los datos de manera autónoma. (Kalita, 2015).

3. Aprendizaje semi-supervisado. El aprendizaje semi-supervisado es una técnica que combina elementos del aprendizaje supervisado y no supervisado. A diferencia de ser entrenado exclusivamente con datos etiquetados o no etiquetados, el modelo de aprendizaje semi-supervisado se entrena utilizando una combinación de ambos tipos de datos. Esto permite al modelo aprovechar la información proporcionada por las etiquetas mientras explora y descubre patrones en los datos no etiquetados. Este enfoque resulta especialmente útil en situaciones en las que hay una cantidad limitada de datos etiquetados pero una gran cantidad de datos no etiquetados, permitiendo un mejor aprovechamiento de los recursos disponibles y una mejora en la capacidad de generalización del modelo (Sengupta et al., 2022).

Con el objetivo de desarrollar un modelo capaz de identificar documentos con lenguaje ofensivo, en esta tesis se propone realizar un análisis del desempeño de métodos de aprendizaje supervisado en la detección de dicho lenguaje. Estos métodos serán complementados con un lexicón generado a partir del corpus seleccionado, el cual estará específicamente enfocado en el contexto de la comunidad LGBTIQ+. Para lograrlo, se llevará a cabo el etiquetado de documentos, clasificándolos en aquellos que contienen lenguaje ofensivo y aquellos que no lo poseen, lo cual servirá como base para el entrenamiento y evaluación del modelo. Este proceso suele ser llevado a cabo por un humano, quien etiqueta un conjunto de datos. A partir de estos datos etiquetados, los métodos de aprendizaje automático generalizan el conocimiento del experto mediante la creación de un modelo predictivo. El objetivo de este modelo es simular de la mejor manera posible el proceso realizado por el etiquetador, permitiendo así la identificación automática de nuevos casos basándose en el conocimiento adquirido.

Debido a que en general no es posible aplicar métodos de aprendizaje automático de manera directa a datos no estructurados, como lo son los documentos y sus etiquetas, es necesario aplicar un preprocesamiento. Además, los documentos pueden contener caracteres, palabras o símbolos que no son útiles para identificar lenguaje ofensivo. Las etiquetas usadas en esta tesis son *Ofensa*, para los documentos que contienen lenguaje ofensivo, *No ofensa* para los que no contienen ese tipo de lenguaje y *Neutral* para los que no tienen alguna relación relevante con el tema LGBTIQ+.

Para convertir un documento en un tipo de dato estructurado, es necesario convertirlo en un vector. La vectorización de documentos es un proceso en el aprendizaje automático en el que se representan los documentos en forma de vector. Existen varias técnicas para vectorizar un documento, como la representación de frecuencia inversa de documentos (TF-IDF), las representaciones distribuidas de Word embedding, hashing vectorizer o la bolsa de

palabras, siendo esta última la más común de todas. Sin embargo, no siempre es la técnica con la que se pueden obtener los mejores resultados (Ilie et al., 2021).

Los documentos vectorizados, y las etiquetas asignadas a ellos pueden ser usados como entrada a un método de aprendizaje supervisado, para predecir etiquetas de nuevos documentos.

La figura 1 muestra de manera gráfica el proceso general anteriormente explicado.

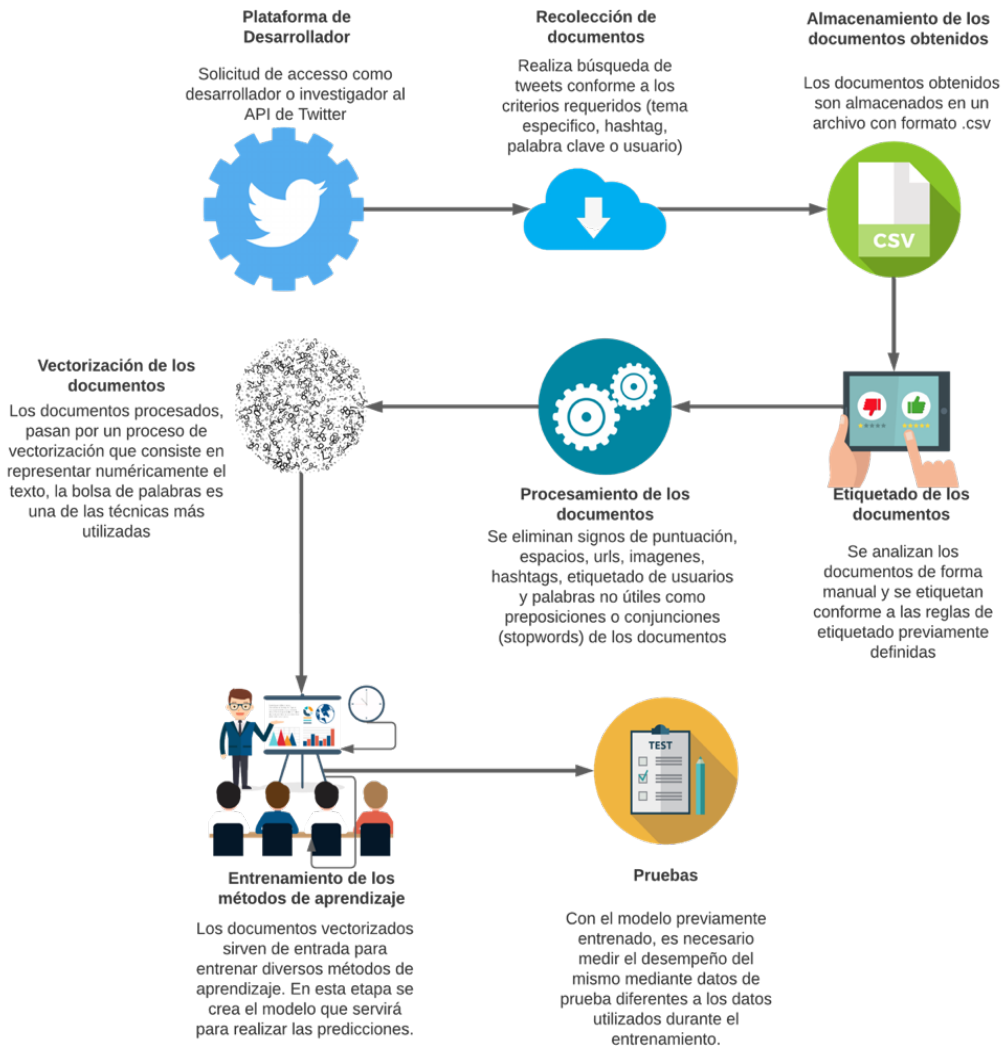


Figura 1. Proceso General para la identificación de lenguaje ofensivo en textos en español de México. Elaboración propia.

La aplicación de técnicas de aprendizaje automático para identificación de lenguaje ofensivo puede ser apoyada mediante el uso de lexicones. Un lexicón es una lista de palabras extraídas de un contexto específico, las cuales tienen asignada una ponderación que da un significado de intensidad. El uso de lexicones ha sido exitosamente aplicado en análisis de sentimientos (Bhowmik et al., 2022) (Sharma & Dutta, 2021) (Wang et al., 2021), por lo que extender su uso a la identificación de lenguaje ofensivo tiene sentido.

2.2.1 Métodos de aprendizaje supervisado utilizados para clasificación

Existen varios métodos de aprendizaje supervisado que se utilizan comúnmente para la clasificación. Estos métodos se basan en el uso de datos de entrenamiento etiquetados, donde cada muestra de datos está asociada con una etiqueta o clase conocida. A continuación, se mencionan algunos de los métodos de aprendizaje supervisado más utilizados para la clasificación (Halim et al., 2020):

1. K vecinos más cercanos (KNN)
2. Máquinas de Soporte Vectorial o Máquinas de Vectores Soporte (SVM)
3. Redes Neuronales (NN)
4. Árboles de Decisión (DT)
5. Clasificador Naive Bayes (NB)
6. Regresión Logística (LR)
7. Bosque aleatorio (Random Forest: RF)

2.2.1.1 K vecinos más cercanos (KNN del Inglés K-Nearest Neighbors)

Es un algoritmo de aprendizaje supervisado que clasifica nuevos datos en función de su posición con respecto a otros datos cercanos. Se puede considerar similar a un sistema de votación(Vemuri, 2020). Por ejemplo, suponiendo que un individuo entra a un restaurante donde no conoce los platillos que se sirven. El

individuo analiza su entorno y observa los platillos de las *K* mesas más cercanas en el restaurante (por ejemplo, *K* = 3). Al notar que la mayoría de las mesas optan por un tipo de sopa en particular (la cual llamaremos Clase A), el individuo decide ordenar lo mismo.

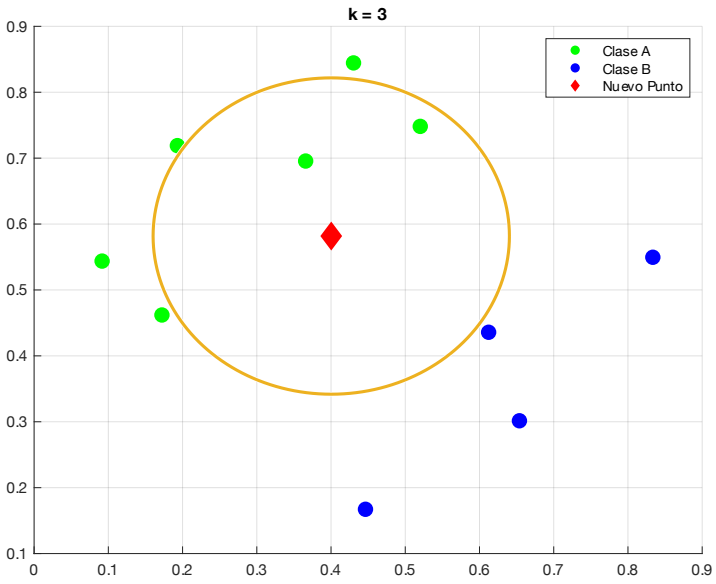


Figura 2. Ejemplo de KNN usado para la predicción de un nuevo punto dadas 2 clases. Elaboración propia.

En la Figura 2, se presentan dos clases diferentes de clientes: aquellos que solicitaron la primera sopa se categorizan como Clase A, mientras que los que eligieron la segunda se categorizan como Clase B. El nuevo punto para clasificación se representa como un rombo en la imagen. Utilizando el algoritmo KNN, es posible predecir la categoría de este nuevo punto basado en su posición en relación con los puntos cercanos.

KNN se basa en la idea de que los puntos de datos similares tienden a estar cerca unos de otros en el espacio de características. Al determinar la clase predominante entre los k vecinos más cercanos de un nuevo punto, se asigna al nuevo punto la misma clase.

Para realizar la clasificación, el algoritmo KNN calcula las distancias entre todos los puntos y utiliza únicamente los K valores más cercanos. En el ejemplo de la Figura 2, se muestra al individuo junto con la distribución de los clientes que solicitaron cada tipo de sopa. En este caso, se configuró un valor de K = 3, lo que significa que se considerarán únicamente los 3 valores más cercanos. Observamos que los 3 valores más cercanos corresponden a la Clase A, por lo tanto, al ser la mayoría, el clasificador establecerá esta clase como la categoría del nuevo punto.

Es importante mencionar que el valor de K puede influir en el resultado de la clasificación. En la Figura 3, se modificó el valor de K a 6, y se puede observar que en esta ocasión se incluyeron puntos de la Clase B. A medida que el valor de K continúa incrementando, se considerarán más valores, lo que puede provocar errores en la precisión de la clasificación.

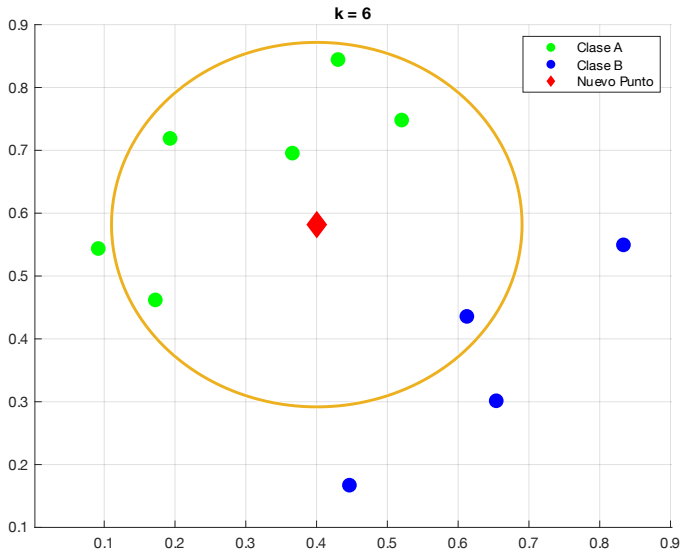


Figura 3. Ejemplo de KNN usado para la predicción con K=6. Elaboración propia.

Para determinar las distancias entre cada par de puntos, es necesario realizar el cálculo de disimilitud entre objetos, el cual mide la distancia o similitud entre dos objetos. La disimilitud cuantifica qué tan distantes están dos objetos entre

sí en función de las características o atributos utilizados en el modelo (Tan et al., 2018).

La disimilitud se puede medir utilizando diversos métodos, dependiendo de las características de los objetos a comparar. Algunas de las técnicas más comunes para medir la disimilitud incluyen el cálculo de la distancia Euclidiana, la distancia Manhattan y la distancia Minkowski, entre otras (Chiu & Tavella, 2020), las cuales se describen brevemente a continuación:

1. **Distancia Euclidiana.** La distancia Euclidiana *d,* es la medida más utilizada para calcular la distancia entre dos puntos, se determina utilizando una línea recta entre el punto *X* y el punto *Y*, en diversos espacios dimensionales (una, dos, tres o más dimensiones), la cual se determina por la siguiente Ecuación 1 (Tan et al., 2018):

$$\boldsymbol{d(x,y)} = \sqrt{\sum_{\boldsymbol{k=1}}^{\boldsymbol{n}} (\boldsymbol{X_k} - \boldsymbol{Y_k})^2}$$

Ecuación 1. Fórmula para calcular la distancia Euclidiana

(Tan et al., 2018)

Donde:

n	Es el número de dimensiones a tratar
X_k y Y_k	Son todos los atributos o componentes para evaluar.

2. **Distancia Manhattan.** También conocida como distancia de ciudad o distancia de bloque. Esta distancia se define como la suma de las diferencias absolutas entre las coordenadas de dos puntos en cada dimensión. Por ejemplo, se requiere calcular la distancia entre dos puntos de una ciudad, tomando en cuenta que se encuentra dividida en forma de cuadricula

generada por las calles, la distancia Manhattan, calcula la suma total de las distancias entre calles para llegar al objetivo. La fórmula para el cálculo de la distancia Manhattan se muestra en la Ecuación 2 (Chiu & Tavella, 2020).

$$\boldsymbol{d(x,y) = (\sum_{k=1}^{n} |X_i - Y_i|)}$$

Ecuación 2. Fórmula para calcular la distancia Manhattan

(Chiu & Tavella, 2020)

Donde:

n Es el número de dimensiones a tratar

X_k y Y_k Son todos los atributos o componentes para evaluar.

3. **Distancia Minkowski**. La distancia Minkowski *d,* es una medida utilizada para determinar la diferencia máxima entre atributos de objetos (Tan et al., 2018).

   La ecuación de la distancia Minkowski puede generalizarse en la Ecuación 3.

$$d(x,y) = \left(\sum_{k=1}^{n} |X_k - Y_k|^2\right)^{\frac{1}{r}}$$

Ecuación 3. Fórmula para calcular la distancia Minkowski

(Tan et al., 2018)

Donde:

r Actúa como parámetro estableciendo:

$r = 1$ - Distancia Manhattan

$r = 2$ - Distancia Euclidiana

$r = 3$ - Distancia Minkowski

El cálculo de disimilitud es esencial en el algoritmo KNN, ya que se utiliza para determinar qué puntos son los más cercanos a un nuevo punto a clasificar. Al evaluar las distancias entre los puntos de entrenamiento y el nuevo punto, se pueden identificar los K puntos más cercanos y utilizar su información para realizar la clasificación.

La selección de la métrica de disimilitud adecuada es crucial, ya que debe adaptarse a las características de los datos y a la naturaleza del problema. Un cálculo preciso de la disimilitud es fundamental para obtener resultados confiables en el proceso de clasificación mediante el algoritmo KNN. La elección cuidadosa de la métrica garantiza que se capturen de manera efectiva las diferencias y similitudes entre los objetos, lo que permitirá una clasificación más precisa y acorde con los datos.

2.2.1.2 Máquinas de Soporte Vectorial (SVM, del Inglés Support Vector Machines)

Es un método de aprendizaje automático utilizado ampliamente para la clasificación de problemas de aprendizaje supervisado. Tiene como objetivo reconocer patrones en dos clases o categorías diferentes las cuales se encuentran limitadas por un margen denominado “hiperplano” (Cortes & Vapnik, 1995). En la Figura 4 se muestra la clasificación de varios objetos, los cuales están colocados en el plano por algún parámetro X definido (por ejemplo, peso o tamaño). En este ejemplo se define el hiperplano lineal de forma ilustrativa aproximadamente a la mitad del plano, dividiendo los objetos en dos secciones. La máquina de soporte vectorial toma este margen para realizar la clasificación, retomando el ejemplo anterior, todos aquellos que se encuentren por encima del margen pertenecerán a una categoría y los que se encuentren por debajo a otra.

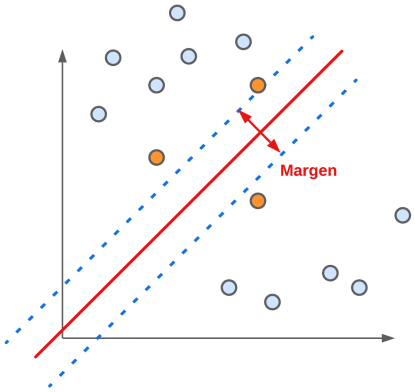


Figura 4. Ejemplo de separación de clases con margen máximo. Elaboración propia.

Las Máquinas de Soporte Vectorial se destacan por su capacidad para manejar problemas con un gran número de características, es decir no están limitadas a resolver problemas solo en dos dimensiones (como el expuesto en la Figura 2), si no que pueden trabajar con **n** número de dimensiones e hiperplanos para la clasificación conforme a las necesidades (Chih-Wei Hsu, Chih-Chung Chang, 2008).

Las máquinas de soporte vectorial son una herramienta poderosa y versátil para la resolución de problemas de aprendizaje supervisado, pero es importante considerar sus fortalezas y debilidades al utilizarlas en un problema específico.

Por ejemplo, (Ben-Hur & Weston, 2010), destaca la robustez de las máquinas de soporte vectorial frente al sobreajuste, lo que las hace adecuadas para el trabajo con conjuntos de datos pequeños o con una relación desequilibrada entre las clases.

Sin embargo, de igual forma, en (Williams, 2003) se señala que las máquinas de soporte vectorial pueden ser sensibles a la escala de los datos y a la presencia de ruido en los datos de entrenamiento, por lo que es de vital importancia, realizar un preprocesamiento adecuado antes de su utilización.

2.2.1.3 Redes Neuronales Artificiales (ANN del Inglés Artificial Neural Networks)

Las redes neuronales son un tipo de aprendizaje automático el cual se inspira en el funcionamiento del cerebro humano (Vemuri, 2020).

El sistema nervioso humano puede verse de manera general como un sistema basado en 3 componentes las cuales se representan en la Figura 5. El cerebro, es el punto principal de este sistema, el cual se encuentra conformado por una serie de neuronas interconectadas entre si las cuales forman una especie de red, que continuamente, a través de los receptores, reciben y perciben información que utilizan para tomar y realizar acciones a través de los efectores (Vemuri, 2020). En la Figura 5 se ilustra este sistema: los receptores, quienes son los encargados de convertir los estímulos del cuerpo humano en pulsos eléctricos los cuales son procesados por las neuronas(red neuronal). Por último, los efectores convierten estos pulsos eléctricos procesados por la red neuronal en respuestas de salida (Vinet & Zhedanov, 2011).

Figura 5. Representación en diagrama a bloques del sistema nervioso humano (Vinet & Zhedanov, 2011)

En el cerebro humano, las neuronas son la unidad mínima de procesamiento de información, las cuales crean una conexión entre sí mediante fibras llamadas axones. Los axones son usados para transmitir impulsos nerviosos de una neurona a otra cuando las neuronas son estimuladas. La conexión entre un axón y una neurona se hace mediante dendritas; el punto de unión entre dendrita y neurona se llama sinapsis(Tan et al., 2018).

Una red neuronal, trabaja de manera muy similar, interconectando nodos y ensamblándolos por medio de relaciones entre ellos. Existen diversos modelos para

definir una red neural, el más simple es el modelo llamado perceptrón (Tan et al., 2018).

- Perceptrón: Consiste en dos tipos de nodos (nodos de entrada y nodos de salida). En el perceptrón cada nodo de entrada está conectado al nodo de salida mediante pesos o ponderaciones, los cuales emulan la fuerza sináptica entre una neurona y un axón. El perceptrón define 2 valores posibles de salida *y* (1,0). Su forma de trabajo se basa en la suma de los pesos ponderados de entrada y de un factor de sesgo o bias *t, cuyo* resultado es evaluado para determinar el valor de salida. La ecuación 4 muestra un ejemplo de 2 valores de entrada con un bias *t=0.2* los cuales son representados de forma gráfica en la Figura 6 (Tan et al., 2018).

$$y = \begin{cases} 1, si\ 0.8_{x1} + \ 0.1_{x2} - 0.2 > 0; \\ 0, si\ 0.8_{x1} + \ 0.1_{x2} - 0.2 < 0; \end{cases}$$

Ecuación 4. Ejemplo de cálculo de salidas para perceptrón. Elaboración propia.

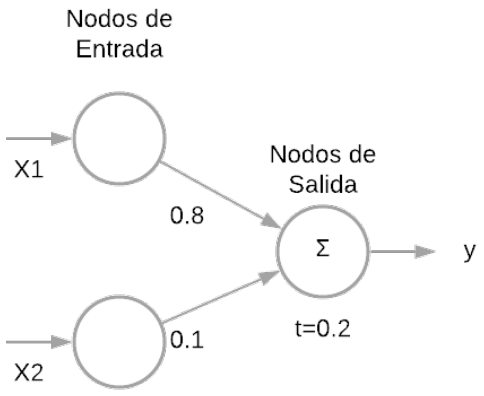


Figura 6. Modelo de Perceptrón con 2 entradas. Elaboración propia.

- Modelo Multicapa. Una red neuronal artificial puede contener una serie de capas intermedias llamadas capas ocultas las cuales contienen nodos embebidos (nodos ocultos). La estructura resultante de este modelo se ilustra en la Figura 7, donde se presenta una arquitectura de tipo Feed-Forward, en

la cual la información de la red se propaga en una dirección única, es decir, hacia adelante.

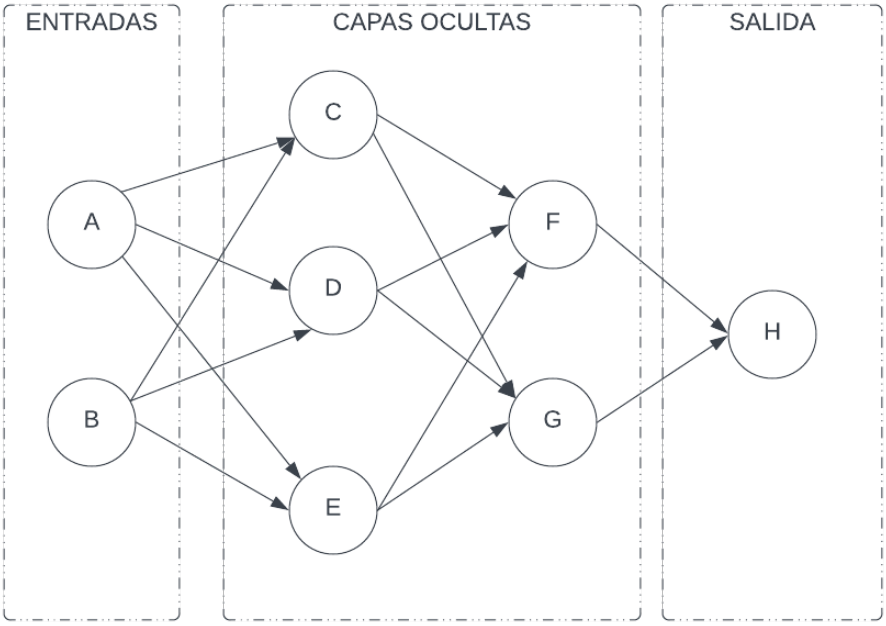


Figura 7. Representación de una red neuronal artificial con arquitectura Feed-Foward. *Elaboración propia*

Cada neurona está caracterizada por un valor numérico denominado función de activación. Vinculado con cada unidad, existe una función de salida, la cual transforma el estado actual de activación en señal de salida. Esta señal es enviada de forma unidireccional a otras neuronas de la red, donde se modifica de acuerdo con una regla determinada en cada interacción. Las señales modificadas que han llegado a la j-esima unidad, se combinan entre ellas generando la entrada total.

Una función de activación, establece el nuevo estado de la neurona, tomando la entrada total calculada y el estado de activación anterior, a continuación se mencionan cuatro de las más usadas (Vemuri, 2020).

1. **Función Lineal.** La función lineal es una de las operaciones más simples que se realizan en una red neuronal. Esta función toma una de las entradas(*x*) y la transforma mediante una multiplicación por una matriz de pesos(*w*), al producto generado se le suma el valor del bias(*b*) o sesgo (Ecuación 5). Como resultado de esta operación se obtiene una representación en forma de vector representado por la Figura 8 (Chiu & Tavella, 2020).

$$\boldsymbol{y = wx + b}$$

Ecuación 5. Fórmula de la función lineal

(Chiu & Tavella, 2020)

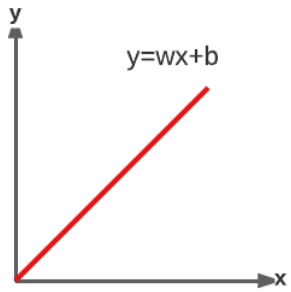


Figura 8. Representación de una función lineal

(Chiu & Tavella, 2020)

2. **Función Sigmoide.** Convierte los valores de entrada en intervalos simples comprendidos entre 0 y 1. Valores muy pequeños tienden a estar más cerca del cero y valores muy grandes al 1. Esta función se representa por la fórmula expuesta en la Ecuación 6 y representada por la Figura 9 (Vinet & Zhedanov, 2011).

$$f(x) = \frac{1}{1 - e^{-x}}$$

Ecuación 6. Fórmula de la función sigmoidal

(Vinet & Zhedanov, 2011)

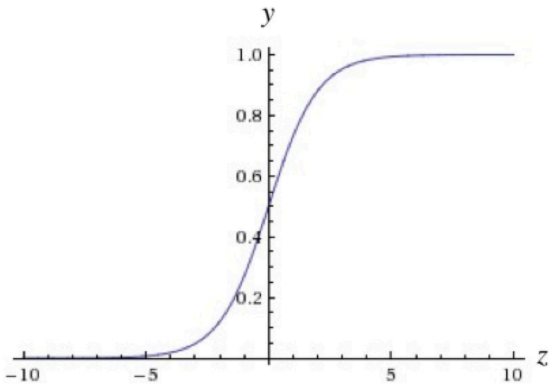


Figura 9. Representación de una función sigmoide o logarítmica

(Vinet & Zhedanov, 2011)

3. **Función Tangente o hiperbólica.** Utiliza una forma similar a la sigmoide, sin embargo, los intervalos son diferentes, en lugar de comprender entre 0 y 1, la función tangente utiliza valores entre -1 y 1. A diferencia de la función sigmoide, esta función involucra valores cero como se muestra en la Figura 10. Esta función se representa por la fórmula expuesta en la ecuación 7 (Vinet & Zhedanov, 2011).

$$f(x) = \tan(x)$$

Ecuación 7. Fórmula de la función tangente

(Vinet & Zhedanov, 2011)

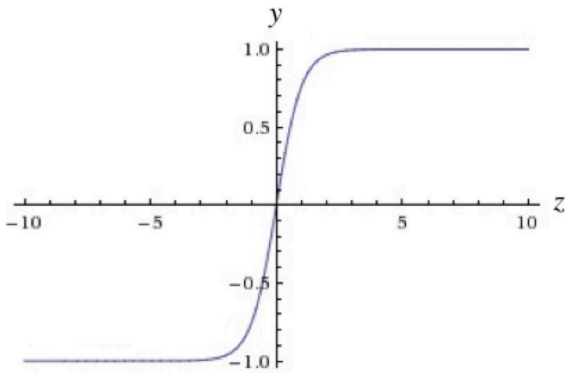


Figura 10. Representación de una función Tangente o hiperbólica

(Vinet & Zhedanov, 2011)

4. **Función Umbral.** Esta función es utilizada cuando las salidas de la red representan valores binarios únicamente. La salida de la neurona se activa cuando el estado de activación es mayor a cierto valor del umbral o menor al mismo. Se podría representar esta función utilizando la Ecuación 8 y la Figura 11 (Vinet & Zhedanov, 2011).

$$f(x) = \begin{cases} 1, si\ X > 0; \\ 0, si\ X < 0; \end{cases}$$

Ecuación 8. Fórmula de la función Umbral

(Vinet & Zhedanov, 2011)

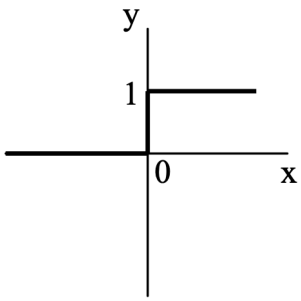


Figura 11. Representación de una función Umbral o Escalonada

(Vinet & Zhedanov, 2011)

2.2.1.4 Árboles de Decisión (DT del Inglés Decision Tree)

En el ámbito del aprendizaje automático y la minería de datos, se emplea el método de clasificación de Árboles de Decisión para llevar a cabo predicciones y facilitar la toma de decisiones. Este algoritmo desarrolla un modelo jerárquico en el que cada nodo denota una decisión y cada hoja representa un resultado(Figura 12). El funcionamiento del algoritmo implica la evaluación de diversos atributos y características de los datos de entrada, con el objetivo de determinar la mejor manera de dividir los datos en subconjuntos más pequeños, iterando hasta alcanzar una solución óptima (Vemuri, 2020).

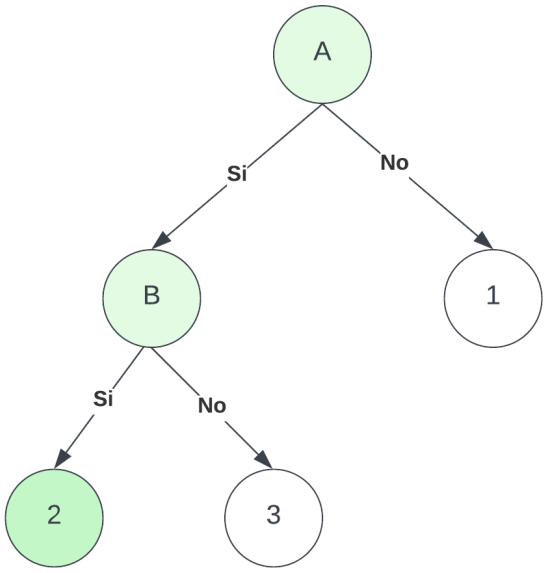


Figura 12. Representación gráfica del método de árbol de decisión. Elaboración propia.

Los árboles de decisión tienen 3 diferentes tipos de nodos:

- **Nodo Raíz (root node).** Es el nodo inicial el cual se encuentra en la cima del árbol de decisión. Este nodo contiene la población total de nodos debajo de él, lo cual significa que es la primera capa de conocimiento por la cual se alimenta el árbol. Por lo general el nodo raíz representa la decisión principal que se debe tomar y el tema de esa decisión en particular (Vemuri, 2020).

- **Nodos Internos.** Los nodos internos son todos aquellos nodos que tienen divisiones debajo de ellos. Cada nodo representa una nueva decisión que debe de ser evaluada para crear nuevos resultados. Cada decisión es una sub-decisión de la decisión anterior y todas, por ende, de la decisión principal. (Vemuri, 2020).

- **Nodos Hoja o terminal.** Los nodos hojas son nodos que no están divididos, los cuales representan la decisión final del árbol de decisión. A cada nodo hoja se le asigna una etiqueta de clase (Vemuri, 2020).

Para alcanzar un resultado final, una decisión debe de moverse desde el nodo raíz a través de las ramas (resultados posibles) hacia los nodos hoja (resultados finales). Este proceso se lleva a cabo a través de varias reglas de clasificación que determinan la probabilidad de ocurrencia de un hecho y cómo se categoriza la información en el árbol. (Vemuri, 2020).

Uno de los principales problemas en los árboles de decisión es el tamaño de estos, el cual en muchas ocasiones suele volverse extremadamente complejo hablando computacionalmente. Para abordar este problema, se han desarrollado diversos algoritmos para realizar la partición de los datos, reduciendo así el espacio de búsqueda y, en consecuencia, haciéndolo más eficiente. Uno de estos algoritmos, es el algoritmo de Hunt (Tan et al., 2018).

El algoritmo de Hunt divide el árbol de decisión de forma recursiva en subsets de datos. A continuación, se describe brevemente los pasos involucrados en él, teniendo como ejemplo un conjunto de datos de entrenamiento D_t asociados con el nodo *t* y un conjunto de etiquetas representadas como $y = \{y_1, y_2, \ldots . . y_c\}$.

Paso 1. Si todos los elementos en D_t pertenecen a la misma clase y_t, entonces *t* será un nodo hoja y se le asignará la etiqueta y_t.

Paso 2. Si D_t contiene elementos de diferentes clases, un atributo es seleccionado para dividir el conjunto en conjuntos más pequeños. Todos los elementos de D_t pasan a ser nodos hijo de él y el proceso se repite hasta no poder ser posible más divisiones (Tan et al., 2018).

Para determinar el mejor camino para dividir elementos de un árbol, se han desarrollado diferentes formas de medición. Estas métricas son definidas con base en la distribución de clases del conjunto de datos, tanto antes como después de ser divididos. Para medir que tan bien fue dividido un árbol de decisión, se utiliza como base el grado de impureza, el cual determina un valor numérico de la mezcla de atributos divididos en el árbol teniendo como meta un grado cero de impureza. Las métricas más utilizadas son (Tan et al., 2018):

- Entropía

$$H' = -\sum_{i=0}^{c-1} p(i|t)\ log_2\ p(i|t)$$

Ecuación 9. Ecuación de la métrica de Entropía

- Gini Index

$$Gini(k) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2$$

Ecuación 10. Ecuación de la métrica de Gini index

- Ganancia de información

$$1 - max_i\ [p(i|t)]$$

Ecuación 11. Ecuación de la métrica de Ganancia de la información

Donde:

C Es el número de clases

0 $0\ log_2 0 = 0$ en terminos de entropía

p(i|t) Probabilidad condicional de que una instancia pertenezca a la clase i

2.2.1.5 Clasificador de Naive Bayes (NB)

El clasificador Naive Bayes es un algoritmo de aprendizaje supervisado probabilístico que se utiliza en clasificación. Se llama “Naive” (ingenuo) debido a que supone independencia condicional entre los atributos de los datos (Tan et al., 2018), es decir considera que la presencia o ausencia de un atributo en particular no está ligada a la ausencia o presencia de otra, como por ejemplo, para determinar la estatura promedio de un grupo, no es necesario saber su nivel académico.

Con esta asunción de independencia, en lugar de calcular las probabilidades para cada elemento de la clase dado el evento *Y*, solo se estima la probabilidad de cada atributo X dado Y **(**probabilidades condicionales), lo que reduce drásticamente la cantidad de datos a trabajar, formalmente se representa en la Ecuación 12.

$$P(X|Y = y) = \prod_{i=1}^{nd} P(X_i|Y = y)$$

Ecuación 12. Fórmula para el cálculo de probabilidades en el algoritmo Naive Bayes

(Tan et al., 2018)

Donde:

X es el conjunto *d* de atributos

El funcionamiento básico del algoritmo Naive Bayes utiliza el teorema de Bayes mostrado en la Ecuación 13 para calcular la probabilidad de que una muestra pertenezca a una clase determinada, dada la distribución de probabilidades de entrada.

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

Ecuación 13. Fórmula del Teorema de Bayes

(Tan et al., 2018)

Para entrenar el clasificador Naive Bayes, primero es necesario realizar el cálculo de la desviación estándar de cada una de las características para cada clase. Estas estadísticas se utilizan para modelar la distribución de probabilidad de las características de entrada para cada clase.

A continuación se calculan las probabilidades a priori de cada una de las clases con base en la proporción de muestras en el conjunto de entrenamiento al que pertenece cada clase (Vemuri, 2020).

El modelo Naive Bayes tiene 3 principales tipos:

- Gaussiano. Asume que los datos de entrada tienen una distribución gaussiana. Es el más utilizado para datos continuos.
- Multinomial. Utiliza la frecuencia de palabras en el documento para poder realizar la predicción, este es uno de los modelos más utilizados.
- Bernoulli. Similar al modelo multinomial, la diferencia radica en los valores de salida, los cuales son únicamente una representación booleana (si / no) al momento de la predicción (Vinet & Zhedanov, 2011).

2.2.1.6 Regresión Logística (LR del Inglés Logistic Regression)

La Regresión Logística es una técnica de aprendizaje supervisado que se utiliza en la clasificación de datos. En lugar de predecir un valor numérico continuo como en la regresión lineal, la Regresión Logística se utiliza para predecir una probabilidad de pertenencia a una clase específica, y esta probabilidad se mapea a una clase binaria mediante una función de activación. Esta función se llama función sigmoide y se parece a una curva en forma de “S”. La función sigmoide transforma los datos de entrada en valores de salida en un rango de 0 a 1, teniendo como resultado de la variable dependiente un valor booleano como se muestra en la Ecuación 14 (Theobald, 2020).

$$y = \frac{1}{1 + e^{-x}}$$

Ecuación 14. Función Sigmoide en el algoritmo de regresión logística

(Theobald, 2020)

Donde:

X es la variable independiente que se desea transformar

e constante de Euler, 2.718

La curva generada por la función sigmoide puede ser utilizada para convertir cada uno de los puntos en un valor numérico entre 0 y 1 (sin alcanzar estos límites). Al aplicar la fórmula, la función sigmoide convierte la variable independiente en probabilidades en relación con la variable dependiente. Aquellos valores más cercanos a cero tendrán menos probabilidad de ocurrir, mientras que los valores más cercanos a uno, tienden a ser más probables de ocurrir (Theobald, 2020).

A las probabilidades calculadas, es posible asignarles una clase. En el caso de clasificación binaria, la Figura 13 muestra un ejemplo de la función sigmoide con diversos puntos graficados, aquellos puntos que se encuentren por encima de 0.5 son clasificados como clase A y los que están por debajo de 0.5 clase B. Puntos con valor igual a 0.5 no pueden ser clasificados, sin embargo, por la naturaleza de la función sigmoide, este escenario es muy poco probable que llegue a ocurrir.

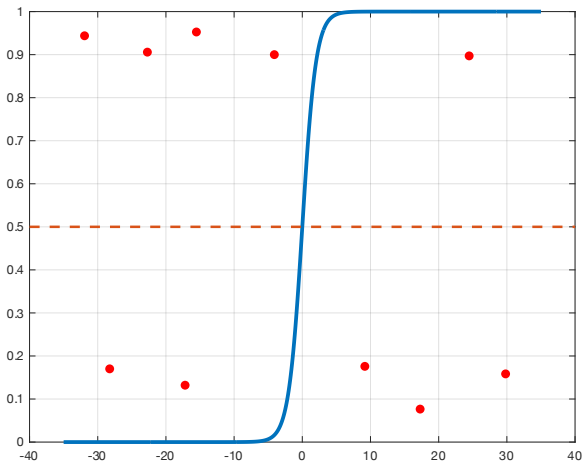


Figura 13. Implementación de la función sigmoide representada gráficamente. Elaboración propia.

La regresión logística es uno de los modelos más simples en el aprendizaje supervisado. Además de ser fácil de implementar, se puede utilizar como base para problemas de clasificación binaria principalmente. Este modelo describe y estima la relación que tiene una variable dependiente y una independiente (Vemuri, 2020).

El modelo de regresión logística se utiliza principalmente para predecir la probabilidad de que una variable binaria, por ejemplo, un correo electrónico, sea spam o no. El objetivo principal es predecir el valor de una variable dependiente tomando como base una o más variables independientes (Vemuri, 2020).

2.2.1.7 Bosque Aleatorio (RF del Inglés Random Forest)

El método de clasificación de bosque aleatorio es de tipo ensamble, lo que significa que varios clasificadores son usados al mismo tiempo para predecir la clase de una instancia. La característica principal de RF es que todos los clasificadores son árboles de decisión diferentes, pero entrenados para predecir datos que siguen una distribución similar a las muestras en el conjunto de datos de entrenamiento (Tan et al., 2018).

Para generar un modelo de clasificación RF con un conjunto de datos de entrenamiento, existen varias opciones, entre las cuales están las siguientes:

a) Elegir conjunto de atributos de manera pseudo aleatoria. Esto hace que cada árbol de decisión generado tome en cuenta solamente $k < d$ atributos, siendo k un número natural, y d el total de atributos en el conjunto de datos de entrenamiento. Debido a la naturaleza aleatoria de la selección de atributos, cada árbol usa diferentes características al resto de los árboles. Los árboles construidos generalmente no son sujetos a poda. Estos árboles en su conjunto forman un bosque, y las predicciones que produce este último son usadas en un esquema de mayoría de votos. Este tipo de bosque se llama Forest-RI, o bosque con entradas aleatorias (Tan et al., 2018).

b) Generar nuevos atributos. Otra forma de crear árboles de decisión para formar un bosque consiste en usar combinaciones lineales de atributos

(elegidos aleatoriamente) en cada nodo interno del árbol. Los coeficientes usados en las combinaciones lineales generalmente tienen un valor real en el intervalo de (-1.0 a 1.0). A este tipo de bosque se le llama Forest-RC, o bosque de características aleatorias (Tan et al., 2018).

c) Selección pseudo aleatoria de mejor atributo. Esta manera de generar árboles de decisión consiste en ordenar los atributos en orden decreciente con respecto a la calidad de los puntos de separación. Después, se elige pseudo aleatoriamente uno de ellos, considerando solamente los K primeros atributos (Tan et al., 2018).

d) Selección pseudo aleatoria de muestras. Este método consiste en particionar el conjunto de datos en M subconjuntos usando selección aleatoria simple con reemplazo. De esta manera, en uno o más subconjuntos puede estar presente un mismo objeto. Los árboles de decisión son construidos usando uno de los subconjuntos generados (Tan et al., 2018).

La Figura 14 muestra una representación general del proceso de creación de árboles de decisión para el clasificador RF.

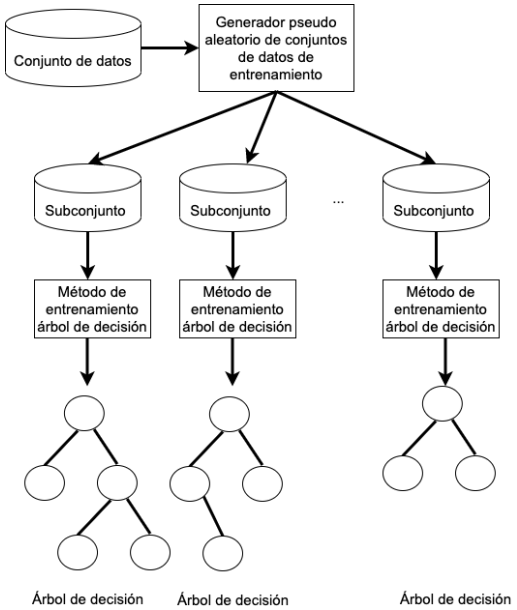


Figura 14. Método general de generación de árboles para RF.

(Tan et al., 2018)

2.2.2 Métricas de desempeño de clasificadores

Las métricas de desempeño de clasificadores son herramientas útiles para evaluar la eficacia de un modelo de clasificación para predecir clases correctamente. El desempeño de un clasificador se mide en un rango de 0 a 1, siendo 1 el 100% de los casos alcanzados. Las métricas de desempeño en clasificadores son fundamentales en la evaluación y selección de un modelo de clasificación adecuado. En general, la elección de una métrica de desempeño depende del problema en cuestión. A continuación, se describen algunas de las métricas más comunes (Vinet & Zhedanov, 2011).

2.2.2.1 Matriz de confusión (Confusion Matrix)

Una matriz de confusión es una tabla que describe el rendimiento de un modelo de aprendizaje automático. Cada una de las filas representa la clase real y las columnas la clase predicha de cada modelo. El nombre matriz de confusión se da por el hecho que es una herramienta que se utiliza para poder identificar dónde el modelo se confundió (Theobald, 2020).

La matriz de confusión suele tener cuatro celdas cuando la clase es binaria, que son las siguientes:

Verdaderos Positivos (TP): Son los casos positivos que el modelo identificó correctamente como positivos.

Verdaderos Negativos (TN): son los casos negativos que el modelo identificó correctamente como negativos.

Falsos Positivos (FP): Son los casos que el modelo identificó como positivos y sin embargo son negativos.

Falsos Negativos (FN): Son los casos que el modelo identificó como negativos y sin embargo son positivos.

La Figura 15 muestra la matriz de confusión que proporciona una forma visual de representación de esta información, la cual es útil para evaluar el desempeño de un modelo de aprendizaje ante diferentes situaciones.

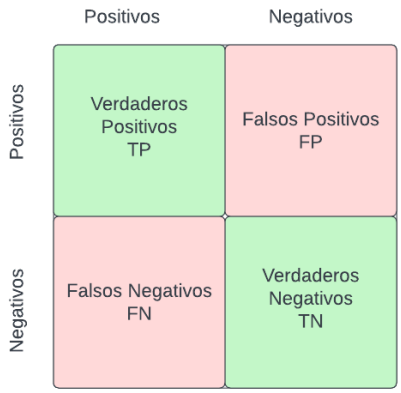


Figura 15. Representación de una Matriz de Confusión. Elaboración propia.

2.2.2.2 Precisión (Precision)

La precisión es una medida comúnmente utilizada en aprendizaje automático para evaluar la calidad de un modelo de clasificación. La precisión se refiere al número de veces que un modelo clasifica correctamente una muestra (TP) en comparación con el número total de clasificaciones realizadas por el modelo (TP + FP) (Theobald, 2020). La precisión se puede representar por la siguiente Ecuación 15:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Ecuación 15. Ecuación para cálculo de la precisión en un modelo de aprendizaje automático (Theobald, 2020)

2.2.2.3 Exhaustividad (Recall)

Recall o exhaustividad es una métrica importante en el aprendizaje automático que se utiliza para evaluar el rendimiento de un modelo de clasificación.

El Recall mide la proporción de ejemplos positivos correctamente identificados (TP) por el modelo en comparación con el número total de ejemplos positivos presentes en los datos (TP + FN) (Theobald, 2020) como se muestra en la Ecuación 16.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Ecuación 16. Cálculo de la exhaustividad en un modelo de aprendizaje automático (Theobald, 2020)

2.2.2.4 F1-Score

El F1-score es una métrica ampliamente utilizada en el aprendizaje automático para evaluar el rendimiento de un modelo de clasificación. El F1-score combina la precisión y la recall para proporcionar una visión balanceada del rendimiento de un modelo.

La precisión mide cuán bien un modelo identifica ejemplos positivos en los datos, mientras que la métrica recall mide la proporción de ejemplos positivos correctamente identificados por el modelo en comparación con el número total de ejemplos positivos presentes en los datos. El F1-score es una combinación ponderada de ambas métricas, donde se busca un equilibrio entre la precisión y la métrica recall (Theobald, 2020). El F1-score se puede calcular a partir de la precisión y la métrica recall mediante la Ecuación 17:

$$F1 - score = 2 * \frac{(precision * recall)}{(precision + recall)}$$

Ecuación 17. Ecuación para calcular F1-Score en un modelo de aprendizaje automático (Theobald, 2020)

Es importante destacar que el F1-score es especialmente útil en problemas de clasificación con desequilibrio de clases, donde la cantidad de ejemplos positivos es significativamente menor que la cantidad de ejemplos negativos. En estos casos, utilizar solo la precisión o la recall puede no ser representativo del rendimiento real del modelo.

2.2.3 Métodos de balance de clases

El desbalance de clases se refiere a la situación en la que una o varias clases de un conjunto de datos tienen una cantidad significativamente menor de ejemplos que otras clases. Este tipo de datos suelen ser un problema para los modelos de aprendizaje automático, debido a que el modelo puede tener dificultad para aprender patrones importantes de esta clase minoritaria.

Por ejemplo, si se desea construir un modelo para realizar el diagnóstico de una enfermedad rara donde solo el 1% de los pacientes la padecen, entonces el conjunto de datos estaría desbalanceado, siendo la clase “pacientes no enfermos” la que predomina. En estos casos el modelo de aprendizaje puede generar problemas traduciéndose en un deficiente rendimiento y errores en la clasificación.

Para abordar el problema de desbalance de clases, existen diversas técnicas siendo la técnica de submuestreo de clase mayoritaria y la técnica de sobre muestreo de clase minoritaria las más populares. Estas técnicas pueden ayudar a equilibrar la representación de clases de un conjunto de datos y así mejorar el rendimiento de este (Brownlee, 2020).

2.2.3.1 Técnica de submuestreo de clase mayoritaria

El submuestreo de clase mayoritaria consiste en eliminar aleatoriamente algunas instancias de la clase de mayor presencia en el conjunto seleccionado hasta igualar la cantidad de instancias entre todas las clases implicadas. La técnica de submuestreo no discrimina entre los datos a eliminar, simplemente permite que sean seleccionados de forma aleatoria.

Una de las principales desventajas de utilizar esta técnica, consiste en que al utilizar de forma aleatoria la eliminación de datos, puede remover miembros importantes (Brownlee, 2020).

2.2.3.2 Técnica estadística de sobre muestreo de minorías sintéticas(SMOTE)

SMOTE (Synthetic Minority Over-sampling Technique) es una técnica de sobremuestreo utilizada para abordar el problema de desequilibrio de clases. En lugar de eliminar instancias de la clase mayoritaria, SMOTE aumenta el número de instancias en las clases minoritarias generando nuevas instancias sintéticas basadas en las existentes.

SMOTE funciona tomando cada instancia de las clases minoritarias generando nuevas instancias sintéticas. Para ello, se seleccionan aleatoriamente una de las instancias de clases minoritarias y localiza sus K vecinos más cercanos. Basados en estos resultados, genera nuevas instancias sintéticas interpolando las características de la instancia seleccionada con sus K vecinos más cercanos.

La ventaja de SMOTE es que a diferencia del submuestreo que disminuye el número de instancias de la clase mayoritaria, SMOTE aumenta el número de instancias sin perder información.

Una de las desventajas de SMOTE, es que puede generar instancias sintéticas muy similares a las originales, lo que puede llevar a un sobreajuste del modelo predictivo (Brownlee, 2020).

2.3 Vectorización de documentos

Los documentos de tipo texto, son afectados principalmente por la gramática, las reglas aplicadas a cada idioma, la longitud de la oración, en otros, lo que origina que cada documento tenga una estructura y tamaño diferente. A este tipo de datos se le conoce como datos no estructurados.

Los datos no estructurados, para que puedan ser entendidos por una máquina, es necesario modificar su estructura debido a que la mayoría de los algoritmos de aprendizaje supervisado requieren que los datos sigan un formato altamente estructurado. Los datos estructurados, se ejemplifican en un formato de tablas, donde cada columna constituye una característica del documento, como la frecuencia de las palabras o la presencia de ciertas palabras y en conjunto forman un vector de características que representa el documento en cuestión; a este proceso se le conoce como vectorización de documentos.

Existen varios métodos de vectorización de documentos, de los cuales podemos mencionar:

1. Bolsa de palabras
2. TF-IDF
3. Word embeddings
4. Hashing vectorizer

A continuación, se explicará cada una de estas representaciones de los documentos.

2.3.1 Bolsa de Palabras (BoW del Inglés Bag Of Words)

Es uno de los métodos más populares de vectorización de documentos. Este método consiste en representar cada documento en un vector de valores reales. En cada componente del vector que representa a un documento, se almacena el conteo de la frecuencia de aparición de cada una de las palabras en todo el corpus (George, 2022). Expresado matemáticamente, se muestra en la ecuación 18:

$$D_{vs} = \{ w_{D1} + w_{D2} + w_{D3}, \ldots\ldots, w_{DN} \}$$

Ecuación 18. Representación matemática de la bolsa de palabras

(Žižka et al., 2019)

Donde:

w_D = Cada una de las frecuencias de las palabras en el corpus.

A manera de ejemplo, considere los dos documentos siguientes, en los cuales se han reemplazado las letras acentuadas por sus equivalentes sin acento, los signos de puntuación e interrogación, y se les ha transformado a minúsculas:

1. “hola buenos días”
2. “hola cómo estás”

La representación de la bolsa de palabras resultaría como se muestra a continuación:

	Hola	buenos	días	cómo	estás
hola buenos días	1	1	1	0	0
hola como estas	1	0	0	1	1

Los valores incluidos en cada una de las columnas indican el número de apariciones de dicha palabra en la frase. El vector que representan a la frase “hola buenos días” sería 11100, mientras que para “hola cómo estás” es 10011. Los problemas más notables de la representación de documentos con bolsa de palabras son los siguientes: los vectores obtenidos son altamente dispersos y no se considera el orden de aparición de las palabras en el documento.

2.3.2 TF-IDF

Es una técnica de vectorización de documentos muy utilizada para la tarea de clasificación y búsqueda de información. Esta técnica se compone de dos términos, por una parte, el TF (Term Frequency o Frecuencia de Término), y por otra el IDF (Inverse Document Frequency o frecuencia inversa de documentos).

La frecuencia de término, se refiere a la cantidad de veces que una palabra aparece en un documento (similar a la bolsa de palabras). Por otro lado, la frecuencia inversa de documento mide la rareza de una palabra tomando como referencia el corpus completo.

Para calcular TF-IDF, es necesario multiplicar la frecuencia del término por la frecuencia inversa de documento. Este cálculo da como resultado un valor numérico (vector) que representa la importancia de un término en relación con su importancia dentro del corpus (George, 2022).

Existen diversas formas de calcular la frecuencia inversa de un documento, siendo la más común la expuesta en la Ecuación 19:

$$IDF = \log\left(\frac{N}{n}\right)$$

Ecuación 19. Fórmula para calcular la frecuencia inversa de un término

(Žižka et al., 2019)

Donde:

Log: Es la función logaritmo natural

N = Número total de documentos en el corpus

n = Número de documentos que contienen el término

A continuación, se muestra un ejemplo de la representación TF-IDF de los dos documentos presentados en la sección previa al cual se le añade un documento adicional para fines del algoritmo.

1. “hola buenos días”
2. “hola cómo estás”
3. “bien gracias”

Tomando como referencia la palabra “hola” en el documento 1, se calcula el TF (Frecuencia de término) de la palabra en el documento:

$$TF(hola, documento\ 1) = \frac{1}{3} = 0.33$$

A continuación, se calcula el IDF (Frecuencia inversa de documento) de la palabra “hola” en el conjunto de documentos:

$$IDF(hola) = \log\left(\frac{3}{2}\right) = 0.1760$$

Finalmente, se calcula el TF-IDF de la palabra “hola” en el documento 1 multiplicando el TF por el IDF obtenidos:

$$TF - IDF(hola, documento1) = (0.33)(0.1760) = 0.05808$$

Por lo tanto, el valor de TF-IDF de la palabra hola el documento 1 es de 0.5808. Este proceso se repite para cada una de las palabras en todos los documentos para determinar la importancia relativa de cada palabra en el corpus.

2.3.3 Word Embeddings

Word Embedding se basa en la idea de que las palabras que tienen significado similar deben ser representadas por vectores similares. La técnica más popular es Word2Vec, la cual utiliza redes neuronales para aprender la representación vectorial de las palabras. Word2Vec utiliza un enfoque CBOW (Continuous bag-of-words) y otro modelo llamado skip-gram.

En el modelo CBOW, se toma como entrada un conjunto de palabras con las cuales trata de predecir la palabra central del conjunto. El modelo skip-gram, a diferencia del anterior, toma como entrada una palabra y trata de predecir las palabras que la rodean. Ambos modelos generan una representación vectorial de cada palabra en el corpus.

Word Embeddings es especialmente utilizado para el procesamiento de lenguaje natural, sin embargo, también tiene algunas limitantes. Por ejemplo, no siempre es capaz de identificar la relación de las palabras y suele tener problemas en capturar el significado de las palabras en contextos específicos o culturales (George, 2022).

2.3.4 Hashing Vectorizer

El hashing vectorizer transforma las palabras en números únicos utilizando una función hash para ello. Este enfoque tiene la ventaja de ser eficiente en términos de recursos computacionales utilizados en su implementación, ya que utiliza el número entero generado por el hash para indexar los vectores de características donde cada elemento del vector es la frecuencia de aparición de la palabra y no requiere realizar cálculos utilizando el corpus completo (George, 2022).

El funcionamiento del hashing vectorizer consiste en tomar cada palabra w de cada documento d en el corpus. Estas palabras son sometidas a una función

hash h adecuada que puede ser MD5, SHA-1 o FNV-1a la cual devuelve un número entero único y se crea un vector de características V de tamaño fijo M donde cada elemento del vector se inicializa en cero. Para calcular el índice se utiliza la Ecuación 20.

$$i = h(w) mod\ M$$

Ecuación 20. Ecuación para calcular el índice de una palabra en el algoritmo de Hashing vectorizer (Žižka et al., 2019)

Donde:

i = índice a calcular

h = Función hash

w = Palabra a convertir

mod M= Se utiliza para asegurarse que el valor esté dentro de los límites del vector

Para calcular la frecuencia de aparición de cada palabra se utiliza el índice generado para obtener el valor de la palabra w en el documento y se suma 1 $V_{[d][i]} = V_{[d][i]} + 1$, esta adición de 1 tiene como propósito incrementar el valor asociado a la palabra "w" cada vez que se encuentra en el texto, contribuyendo así al seguimiento preciso de su frecuencia de aparición.

A continuación, se muestra un ejemplo de la representación Hashing Vectorizer de dos documentos.

1. “hola buenos días”
2. “hola cómo estás”

Suponiendo que se desea presentar los datos como un vector de 5 características utilizando esta técnica. Como primer paso, se inicializan los vectores en cero:

Vector 1 = [0 , 0 , 0 , 0 , 0]

Vector 2 = [0 , 0 , 0 , 0 , 0]

Como siguiente paso, se separan las palabras en tokens:

1. [hola, buenos, días]
2. [hola, cómo, estás]

Se elige la función hash a utilizar, esta función asignará un valor único a la palabra en cuestión. Para fines de ejemplificación, se asume con valores aleatorios el resultado de un algoritmo MD5 a cada palabra.

$$hash - MD5(hola) = 107$$

$$hash - MD5(buenos) = 405$$

$$hash - MD5(dias) = 134$$

$$hash - MD5(cómo) = 512$$

$$hash - MD5(estás) = 901$$

Al valor resultante, se obtiene el valor del módulo del hash con el tamaño total del vector de características para asegurar que el índice se encuentre entre el rango de 0 y el tamaño máximo del vector.

$$indice(hola) = 107\ MOD\ 5 = 2$$

$$indice(buenos) = 405\ MOD\ 5 = 0$$

$$indice(dias) = 134\ MOD\ 5 = 4$$

$$indice(cómo) = 512\ MOD\ 5 = 2$$

$$indice(estás) = 901\ MOD\ 5 = 1$$

El Hashing Vectorizer representa cada palabra asignándole un valor de índice, el cual se utiliza sumándole 1 al valor obtenido.

$$vector(documento\ 1, indice(hola))) = vector(documento\ 1, 2) += 1$$

$$vector(documento\ 1, indice(\ buenos)) = vector(documento\ 1, 0) += 1$$

$$vector(documento\ 1, indice(dias)) = \ vector(documento\ 1,4) += 1$$

$$vector(documento\ 2, indice(\ hola)) = vector(documento\ 2, 2) += 1$$

$$vector(documento\ 2, indice(como)) = \ vector(documento\ 2, 2) += 1$$

$$vector(documento\ 2, indice(\ estas)) = vector(documento\ 2, 1) += 1$$

La representación vectorial para los documentos es la siguiente:

Vector 1 = [1, 0, 1, 0, 1]

Vector 2 = [0, 1, 2, 0, 0]

2.4 Revisión del Estado del Arte

Durante la revisión del estado del arte sobre la identificación de lenguaje ofensivo, se encontraron principalmente trabajos enfocados en idiomas diferentes al Español; mientras este último se ha trabajado en una menor medida. Los artículos analizados se centran en dos enfoques principales para abordar el problema de identificación de lenguaje ofensivo: el uso de lexicones y la aplicación de técnicas de ML. Estas propuestas han servido de base para la formulación de la solución planteada en esta tesis.

En la sección 2.3.1 se muestran los enfoques para la identificación de lenguaje ofensivo basados en lexicones, en la sección 2.3.2 se describen los sistemas basados en métodos de aprendizaje automático y por último, en la sección 2.3.3 los sistemas basados en aprendizaje profundo con el mismo propósito.

2.4.1 Identificación de lenguaje ofensivo basado en lexicones

Los métodos basados en lexicones son comúnmente utilizados para la identificación de lenguaje ofensivo en textos. Se puede conceptualizar a un lexicón como una lista de palabras que guardan alguna relación con un tópico. Estos sistemas utilizan dicho conjunto de palabras (lexicón) para identificar si un texto contiene lenguaje ofensivo o identificar un sentimiento en específico (Tiara et al., 2015).

En (Pamungkas et al., 2020), se plantea un método para identificar la misoginia en distintos idiomas a partir de documentos extraídos de Twitter. En dicho trabajo se llevó a cabo la implementación de HurtLex, un lexicón de acceso libre disponible en internet (https://github.com/valeriobasile/hurtlex). HurtLex es un recurso meticulosamente creado con el propósito de ofrecer una herramienta que permita la detección automática de palabras ofensivas, cuenta con una extensa lista de términos en múltiples idiomas, incluyendo el Español, junto son su nivel de agresividad asociado.

Los resultados obtenidos al aplicar HurtLex al corpus se utilizan para entrenar diversos métodos de aprendizaje supervisado y aprendizaje profundo. Con la inclusión de HurtLex como una característica en el modelo mostró una mejora significativa en su capacidad para detectar instancias de discursos misóginos en los documentos analizados.

Los resultados obtenidos con HurtLex son bastante buenos en términos de precisión, mostrando valores aproximadamente de 0.80 para el idioma Español y 0.91 para Inglés. Para otros idiomas, la precisión se sitúa en torno al 0.70, esto utilizando una combinación del lexicón HurtLex y SVM como método de aprendizaje supervisado, lo cual muestra que el rendimiento del modelo varía dependiendo al idioma analizado. Un problema con HurtLlex es en la creación o expansión del lexicón, ya que el etiquetado se realiza de manera manual.

De igual forma, (Kocoń et al., 2021), utiliza un lexicón de acceso abierto (Hatebase.org), el cual contiene una extensa base de datos sobre palabras orientadas al odio. La diferencia principal con HurtLex, es que Hatebase contiene términos únicamente enfocados al idioma Inglés. Además, HurtLex se enfoca en un tipo especifico de odio, en (Kocoń et al., 2021) tratan de identificar los mensajes que emitan el sentimiento de odio en general, consideran tres clases principales de comentarios: Ofensivos, No Ofensivos, y Muy ofensivos.

Es importante tener en cuenta que el enfoque de identificación de lenguaje ofensivo basado en lexicones tiene varias limitaciones. En primer lugar, la lista de palabras ofensivas puede ser limitada, y no contener todas las palabras que se usan en los mensajes de texto con contenido ofensivo. En segundo lugar, algunas palabras pueden tener diferentes significados dependiendo del contexto en el que se utilizan, lo que puede generar falsos positivos o falsos negativos en la identificación de lenguaje ofensivo. Otra limitación es el idioma, ya que, si el texto analizado está en uno diferente al de la lista de palabras, la simple traducción podría dar lugar a errores en la predicción.

En (Pronoza et al., 2021a), los autores se centran en la detección de ofensas dirigidas a grupos étnicos en idioma ruso. Los autores explican que la detección de este tipo de discurso es un desafío, lo que coincide con lo encontrado en otros artículos (Plaza-Del-Arco et al., 2020) (Vrysis et al., 2021)(Chandrasekharan et al., 2017). En (Pronoza et al., 2021a), también se consideró la naturaleza informal del lenguaje, los regionalismos y modismos utilizados en redes sociales. En los experimentos usaron más de 2.6 millones de mensajes de usuarios que mencionan grupos étnicos. El lexicón generado con estos mensajes le llamaron RuEthnoHate. Este lexicón ayuda a contextualizar el análisis y a capturar mejor la carga emocional asociada a determinados términos. Con el fin de construir una muestra representativa, se tomaron 12,000 documentos etiquetados en 3 distintas clases: Positivo, Neutral y Negativo, siendo esta última clase la que implica discurso de odio étnico. De estos, 2040 mensajes son etiquetados como Negativos, 1315 Positivos

y 8697 como Neutrales. El mejor resultado obtenido fue un F1-Score de 0.824 con el modelo Convers- RuBERT, utilizando la representación de grupo étnico y texto.

Por otro lado, en (Tiara et al., 2015) se describe un enfoque diferente, los autores se enfocan en evaluar el sentimiento generado en textos obtenidos de Twitter sobre programas de televisión indonesio. El proceso consta de tres etapas: preprocesamiento de datos, método basado en lexicones y el empleo de una SVM. En la etapa de la aplicación de lexicones, se extraen las palabras de opinión y se les asigna un valor de polaridad. Además, se manejan las negaciones y se calcula la distancia entre las palabras de opinión y las entidades. Para realizar la evaluación, se utiliza una SVM combinado con TF-IDF. Los resultados muestran que la combinación de métodos basados en léxicos y SVM puede utilizarse para analizar el sentimiento en programas de televisión con una tasa de precisión del 0.8. Los autores concluyen que el método basado en léxicos podría mejorarse mediante el uso de un diccionario que proporcione etiquetas según el nivel de fuerza de cada palabra de opinión. Este enfoque es similar al propuesto en esta tesis.

Ambos artículos (Pronoza et al., 2021a) y (Tiara et al., 2015) se enfocan en un contexto en específico, dirigidos a un idioma en particular.

Por último, (Tang et al., 2014) expone una perspectiva diferente para la creación del lexicón y abordar el tema del análisis de sentimientos. Al utilizar datos provenientes de internet (específicamente de la plataforma Twitter), los autores de ese artículo reconocieron que el uso de lexicones pequeños no es suficiente para la identificación correcta de lenguaje ofensivo debido a la naturaleza de los mensajes, ya que estos contienen lenguaje informal, argot y expresiones multi-palabra que no son cubiertas por los lexicones tradicionales.

Los autores de este artículo se enfocaron en la creación de un lexicón utilizando información de Twitter y aplicando técnicas de representación de aprendizaje. Su enfoque se basó en el uso de una arquitectura de red neuronal hibrida con una función de perdida. De igual forma, los autores incorporaron palabras del “diccionario urbano” como semillas para expandir la lista de palabras

en el lexicón. El “diccionario urbano” es un recurso valioso que contiene una variedad de términos de jerga y expresiones utilizadas en los tweets, específicamente en idioma Inglés (similar al ocupado en esta tesis). Esta estrategia permitió capturar mejor la diversidad de sentimientos y matices presentes en los mensajes de Twitter.

2.4.2 Sistemas de identificación del lenguaje ofensivo basados en aprendizaje automático

Los sistemas de identificación del lenguaje ofensivo basados en aprendizaje automático son aquellos que utilizan técnicas de inteligencia artificial, específicamente de aprendizaje automático para identificar y clasificar lenguaje ofensivo en textos.

En estos sistemas, se entrena un modelo de aprendizaje automático utilizando un conjunto de datos de texto etiquetados que contiene ejemplos de lenguaje ofensivo y no ofensivo. A partir de estos datos, el modelo aprende a identificar patrones y características en el lenguaje ofensivo y puede aplicar este conocimiento para clasificar nuevos textos como ofensivos o no ofensivos.

En (Del Bosque & Garza, 2014), se diseñó una escala para medir el nivel de agresividad en los tweets, centrándose en palabras relacionadas con el acoso escolar. Esta escala, que se mide en un rango de 0 a 10, fue creada utilizando diversas metodologías. Los autores observaron que la detección de comentarios basados en odio está estrechamente relacionada con la identificación de emociones en un texto. Sin embargo, reconocieron que realizar la clasificación de textos únicamente basándose en las emociones no es suficiente, ya que existen expresiones lingüísticas más elaboradas que no necesariamente expresan una emoción especifica.

Los resultados obtenidos demostraron que el modelo de regresión lineal es altamente efectivo para clasificar este tipo de documentos a gran escala. Además, se observó que el uso excesivo de lenguaje agresivo en los tweets facilita la identificación de mensajes de odio.

Estos hallazgos son importantes porque ayudan a comprender la complejidad y los desafíos asociados con la identificación de mensajes de odio. También resaltan la necesidad de combinar enfoques y técnicas diferentes para lograr una evaluación más precisa y completa de textos provenientes de redes sociales.

En un estudio posterior realizado por (Khanday et al., 2022), se abordó el problema de detección de discursos de odio durante la pandemia de COVID-19. Evaluaron ocho métodos de aprendizaje supervisado diferentes y encontraron que los clasificadores de árboles de decisión y aumento de gradiente estocástico fueron efectivos en la detección de este tipo de discursos, obteniendo valores altos en n términos de f1-score (entre 0.96 y 0.97). El estudio menciona que los algoritmos simples basados en palabras no logran descubrir contenido ofensivo sutil en las redes sociales debido a la complejidad del lenguaje natural, como se mencionó anteriormente. Por lo tanto, es necesario utilizar técnicas de aprendizaje automático más avanzadas y complejas. En este estudio, se utilizaron solo dos clases para clasificar los tweets: “odio” y “No odio”, y se etiquetaron los documentos manualmente para mejorar la precisión en diferentes idiomas y contextos.

En otro estudio realizado por (Park & Fung, 2017), se propuso un enfoque de dos etapas para la detección de textos con lenguaje racista y sexista. Los autores se centraron en el uso de redes neuronales convolucionales (CNN) para encontrar características relevantes en los textos. Este enfoque obtuvo resultados prometedores en términos de f1-score los cuales rondan entre 0.824 y 0.827. Sin embargo la dificultad principal radicó en la definición misma de lenguaje ofensivo debido a la subjetividad y la falta de contexto. Por lo tanto, el etiquetado de los documentos fue un proceso esencial para obtener resultados óptimos.

En concordancia con los estudios anteriores, (Cruz et al., 2022) cuyo estudio se basa en idioma Inglés, afirmaron que el uso de técnicas de aprendizaje automático por si solas no es suficiente para identificar correctamente el lenguaje ofensivo. Por lo tanto, propusieron un marco de trabajo que analiza relación de múltiples técnicas de extracción de características y algoritmos de clasificación. Este marco de trabajo consta de tres partes principales: generación, selección e integración. En la generación, se entrenan diferentes clasificadores, ya sea homogéneos(utilizando el mismo algoritmo de aprendizaje) o heterogéneos (utilizando diferentes algoritmos de aprendizaje). Luego, en la etapa de selección, se eligen los clasificadores que obtuvieron los mejores resultados para formar el conjunto final. En la última etapa, la integración, se combinan las salidas de los clasificadores seleccionados. Esta técnica conocida como Sistema de Clasificación Múltiple (MCS, por sus siglas en Inglés), ha demostrado ser muy efectiva para mejorar la precisión de clasificación en comparación con el uso de un solo clasificador.

Los estudios mencionados resaltan la importancia de enfrentar los desafíos y la complejidad relacionadas con la identificación del lenguaje ofensivo y los discursos de odio en las redes sociales.

En este contexto, se hace hincapié en la necesidad de combinar diferentes enfoques y técnicas para abordar de manera efectiva esta problemática. Por un lado, se destaca la importancia de la medición de la agresividad en los tweets como un indicador clave para identificar mensajes ofensivos. Esto implica analizar el uso de palabras relacionadas con el acoso escolar u otras formas de agresión verbal, y asignarles un nivel de agresividad en una escala predefinida.

Además, la identificación de emociones en un texto se revela como otro aspecto crucial para detectar discursos de odio. Si bien reconocen que las emociones por sí solas no son suficientes para abordar esta tarea de manera exhaustiva, sí representan una pista importante para comprender el tono y la intención detrás de las palabras utilizadas en los mensajes.

Para mejorar la precisión en la detección, se sugiere el uso de algoritmos de aprendizaje supervisado avanzados. Estos algoritmos tienen la capacidad de aprender patrones y características sutiles en los textos ofensivos, lo que permite una clasificación más precisa y automatizada. Además, la utilización de múltiples clasificadores en conjunto proporciona resultados aún más robustos, ya que cada uno puede aportar diferentes perspectivas y enfoques para abordar la detección de lenguaje ofensivo.

En última instancia, estos hallazgos resaltan la importancia de abordar los desafíos y la complejidad inherentes a la identificación del lenguaje ofensivo y los discursos de odio en las redes sociales. La combinación de enfoques como la medición de agresividad, la identificación de emociones y el uso de algoritmos de aprendizaje supervisado avanzados, junto con la utilización de múltiples clasificadores, ofrece una estrategia prometedora para mejorar la precisión en la detección.

2.4.3 Sistemas de identificación de lenguaje ofensivo basados en aprendizaje profundo

Los sistemas de identificación de lenguaje ofensivo basados en aprendizaje profundo han demostrado ser eficaces en la detección de contenido ofensivo y discriminatorio en diferentes plataformas en línea. Estos sistemas utilizan algoritmos de aprendizaje profundo, como redes neuronales convolucionales (CNN) y redes neuronales recurrentes (RNN), para capturar patrones y características lingüísticas que son indicativas de lenguaje ofensivo (Zhang & Wallace, 2015).

En (Sreelakshmi et al., 2020) se presenta una metodología para abordar el desafío de detectar discursos de odio en texto de redes sociales en código mixto Hindi-Inglés.

Utilizando modelos pre-entrenados como fastText, word2vec y doc2vec, se lleva a cabo una comparación exhaustiva para evaluar su rendimiento en esta tarea

específica. Los resultados obtenidos indican que fastText, en combinación con el clasificador SVM-RBF, ofrece una representación de características más efectiva, lo que demuestra su capacidad para identificar y clasificar adecuadamente los discursos de odio en texto de redes sociales.

Uno de los hallazgos destacados de este estudio es que las características a nivel de caracteres proporcionan información más valiosa que las características a nivel de palabras y documentos en el contexto del código mixto. Esto resulta particularmente relevante en países como India, donde existe una gran diversidad lingüística y una presencia significativa de poblaciones multilingües y bilingües. La detección de discursos de odio se complica aún más debido a las variaciones no estándar en la ortografía y la gramática presentes en el código mixto. Además, la falta de restricciones en la expresión de opiniones en los sitios web de redes sociales permite que los usuarios emitan comentarios abusivos y adversos con la intención de dañar la imagen y el estatus de otras personas en la sociedad.

La metodología propuesta en este artículo logra una precisión del 0.85 y un recall destacado de 0.81. Estos resultados son de gran importancia, ya que proporcionan una base sólida para futuros avances en la detección de contenido perjudicial en línea.

Uno de los enfoques populares en este campo es el uso de modelos de procesamiento de lenguaje natural (NLP) pre-entrenados, como BERT (Bidirectional Encoder Representations from Transformers), que han demostrado un rendimiento destacado en tareas de clasificación de texto. A estos modelos pre-entrenados se les da un ajuste fino utilizando conjuntos de datos etiquetados para la identificación de lenguaje ofensivo.

En el estudio llevado a cabo por (Davidson et al., 2017), se investigó el uso de modelos de aprendizaje profundo, específicamente CNN (Redes Neuronales Convolucionales) y LSTM (Memoria de Corto Plazo de Largo Plazo), para abordar la clasificación del contenido ofensivo en tweets. El objetivo principal del estudio fue

comparar el desempeño de estos modelos de aprendizaje profundo con los enfoques tradicionales que se basan en características lingüísticas.

Los resultados revelaron que los modelos de aprendizaje profundo superaron de manera significativa a los enfoques tradicionales en términos de clasificación precisa del contenido ofensivo en los tweets analizados obteniendo: precisión de 0.91, recall 0.90 y f1-score 0.90. Esto indica que la capacidad de los modelos de aprendizaje profundo para aprender patrones y características más complejas y abstractas en el texto les proporcionó una ventaja en la identificación de contenido ofensivo.

Además, el estudio destacó la importancia de incorporar características adicionales, como la información contextual y la interacción entre palabras, para mejorar aún más la precisión de la clasificación. Estas características adicionales permiten capturar mejor la semántica y el contexto de los mensajes, lo que resulta en una mejor comprensión de la intención ofensiva detrás de los textos.

En otro estudio relevante realizado por (Nobata et al., 2016), se enfocaron en la detección de comentarios tóxicos en línea utilizando un enfoque basado en CNN (Redes Neuronales Convolucionales). El objetivo principal fue desarrollar un sistema capaz de identificar de manera precisa y eficiente los comentarios ofensivos en diversos contextos en línea.

Los resultados obtenidos en este estudio fueron muy prometedores. El enfoque basado en CNN logró un alto rendimiento en términos de precisión (0.773), recall (0.794) y f1-score (0.783) en la detección de comentarios tóxicos. La capacidad de las redes neuronales convolucionales para aprender automáticamente características relevantes a partir de datos sin procesar permitió capturar patrones lingüísticos complejos que son indicativos de comentarios ofensivos.

Además, el estudio exploró el uso de características léxicas y características derivadas de modelos de lenguaje pre-entrenados, como word2vec. Se descubrió

que la combinación de estas características mejoró aún más la capacidad del sistema para identificar comentarios ofensivos. Las características léxicas proporcionaron información valiosa sobre el contenido específico de las palabras utilizadas, mientras que las características derivadas de modelos de lenguaje pre-entrenados capturaron relaciones semánticas y contextuales entre las palabras.

Estos hallazgos destacan la importancia de combinar diferentes tipos de características en el desarrollo de sistemas de detección de comentarios tóxicos en línea. La integración de características léxicas con representaciones de lenguaje más sofisticadas obtenidas de modelos de lenguaje pre-entrenados puede mejorar significativamente la capacidad del sistema para identificar y clasificar comentarios ofensivos de manera precisa.

CAPÍTULO 3

3. Metodología

La presente tesis se centra en el logro de 7 objetivos específicos los cuales se resumen en la Figura 16. Estos objetivos son el núcleo de la investigación y proporcionan una visión general de los aspectos abordados en el estudio

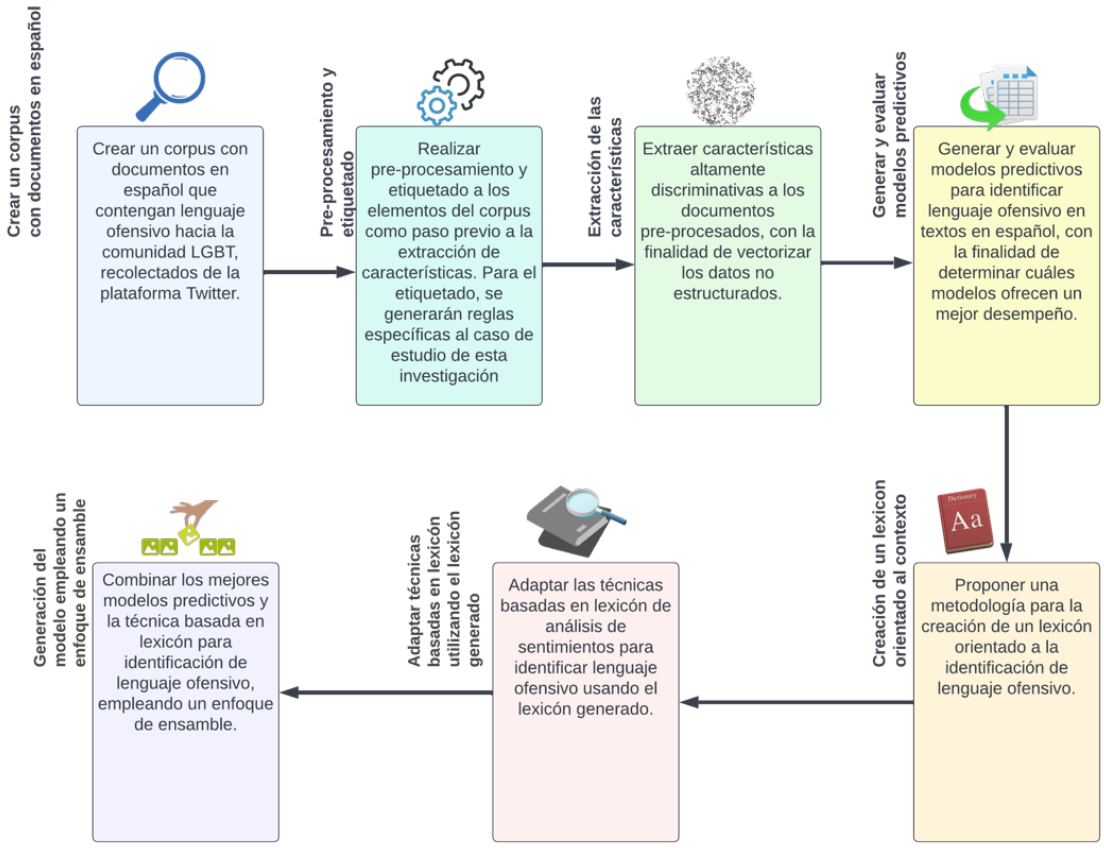


Figura 16. Resumen de Metodología. Elaboración propia.

3.1 Creación de un corpus con documentos en Español

Para obtener los datos necesarios para el entrenamiento y las pruebas del algoritmo, se utilizó la plataforma Twitter para crear un corpus de documentos en Español. La plataforma cuenta con una amplia cantidad de mensajes en forma de texto plano escritos por sus usuarios que abarcan una variedad de temas y pueden contener emoticones, videos o imágenes, con una longitud máxima de 280 caracteres.

El proceso de obtención de los datos se dividió en varias etapas. En primer lugar, se procedió a crear una cuenta de desarrollador o investigador en Twitter para obtener acceso a su API (Application Programming Interface). La API proporciona una serie de microservicios que permiten realizar consultas y manipular datos en la plataforma. Esta etapa fue fundamental para establecer una conexión con Twitter y poder obtener la información necesaria.

Una vez obtenido el acceso a la API, se procedió a desarrollar un script en Python utilizando la biblioteca externa Tweepy. Tweepy, la cual es una herramienta que facilita el manejo de las operaciones relacionadas con Twitter, como la consulta de tuits basada en parámetros de búsqueda, la obtención de tuits de usuarios específicos y la exploración de los temas de tendencia en tiempo real. La utilización de Tweepy agilizó considerablemente el proceso de extracción de datos y proporcionó flexibilidad en la obtención de información relevante.

El script desarrollado se centró en la búsqueda de tweets que cumplieran con un algoritmo específico propuesto en esta tesis. El algoritmo de búsqueda se basó en una serie de palabras clave relacionadas con la comunidad LGBTIQ+, como "LGBT", "LGBT++", "Besoton", "SoyHomosexual", "MexicoLGBTTT", "ScruffLatino" y "QueAscoSerHomosexual". Estas palabras clave se utilizaron para filtrar los tuits y limitar los resultados a aquellos que fueran relevantes para el estudio.

Además, se establecieron ciertos criterios para la selección de los tuits. Se descartaron los retuits y las respuestas, centrándose únicamente en los tuits originales. También se limitó la búsqueda a tuits escritos en Español, ya que el objetivo era construir un corpus de documentos en este idioma.

Una vez obtenidos los tuits relevantes, se almacenaron en un archivo CSV, donde cada tuit se representó como un documento individual. Esto permitió organizar y estructurar los datos de manera adecuada para su posterior procesamiento.

En base a los documentos obtenidos, se realizó un procesamiento en memoria (sin modificar el archivo CSV). Este procesamiento consistió en eliminar las palabras que no representan valor al estudio, como conjunciones o preposiciones. Luego, se creó una lista con todas las palabras únicas en los documentos y se ordenaron por frecuencia de aparición, de mayor a menor. Utilizando este ordenamiento, el revisor tuvo que seleccionar manualmente las 10 palabras siguientes que se emplearían en la búsqueda posterior, las cuales debían ser diferentes a las primeras 10. Esto indica que el script no se limita a una sola ejecución.

Para garantizar la disponibilidad de un número suficiente de documentos para el entrenamiento y las pruebas del algoritmo, se repitieron esos pasos hasta obtener el número de documentos deseados. Después de obtener una cantidad inicial de documentos, se revisaron manualmente los primeros resultados, descartando aquellos que ya habían sido incluidos en iteraciones anteriores o aquellos que no aportaban valor al estudio, según el criterio del revisor. Luego, se continuó con la extracción de nuevos documentos utilizando los mismos pasos y criterios establecidos previamente. Este proceso se repitió hasta obtener el número deseado de documentos para el corpus.

La repetición de los pasos permitió garantizar la variedad y relevancia de los documentos recopilados, evitando la inclusión excesiva de duplicados y maximizando la diversidad de información en el corpus final.

En total, se recopilaron 126,000 documentos durante un período específico comprendido entre el 25-03-2022 y el 01-07-2022. Estos documentos representan una muestra significativa de los tuits en Español relacionados con la temática LGBTIQ+ en ese período.

En la Tabla 1 se detallan los atributos extraídos para cada mensaje obtenido. Estos atributos pueden incluir información como el contenido del tuit, el autor, la fecha de publicación y otros metadatos relevantes.

Tabla 1 Atributos obtenidos por mensaje utilizando la librería Tweepy

Campo	Descripción
full_text	Texto del Tuit en formato extendido de 280 caracteres
Created	Fecha de creación del Tuit
id	Id del Tuit
isRetweet	Bandera para indicar si el mensaje es un retuit
SearchCriteria	Parámetro utilizado para realizar la búsqueda

3.2 Preprocesamiento y etiquetado de los datos

3.2.1 Etiquetado de los datos

El etiquetado de los datos es un paso crucial en el análisis de sentimientos. El correcto funcionamiento de los métodos supervisados depende de este proceso, por lo que es fundamental leer detenidamente los textos y asignar las etiquetas que mejor correspondan al sentimiento especificado (López-Chau et al., 2020).

Como se mencionó, se propusieron tres categorías para el etiquetado: "Ofensa", "No ofensa" y "Neutral". Cada documento etiquetado debe cumplir las siguientes reglas para ser considerado en alguna de las citadas categorías:

- Ofensa: Un documento se etiqueta en esta categoría si contiene al menos una grosería, un mensaje que incite al odio, a la discriminación por orientación de género, o que denigre a un sexo en específico o a la ausencia de sexo definido. También se consideran maldiciones o comentarios denigrantes hacia la comunidad LGBTIQ+.

- No ofensa: Un documento se considera en esta categoría si al leerse se identifican mensajes de apoyo a la comunidad LGBTIQ+ o palabras de aliento.

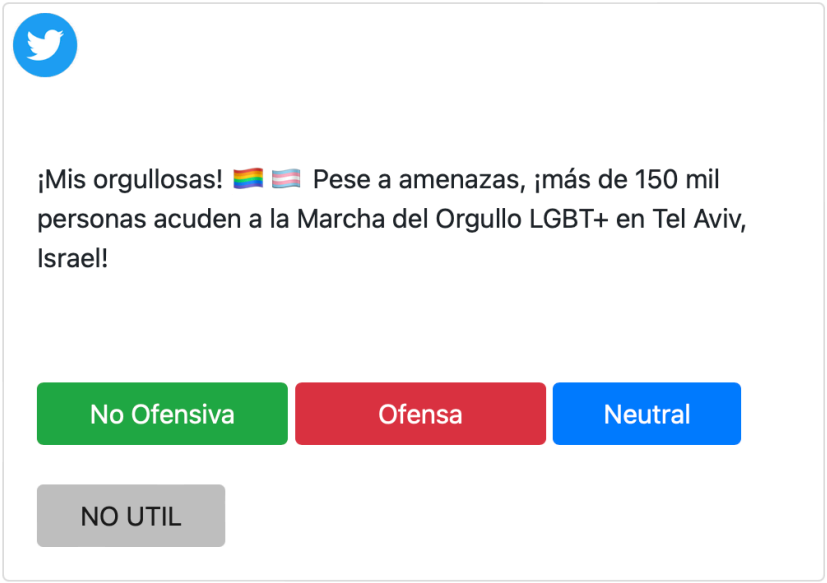
- Neutral: Un documento se etiqueta como neutral si su contenido no está relacionado con la comunidad LGBTIQ+, como temas de marketing, publicidad u otros temas no vinculados. También se consideran mensajes que únicamente etiquetan personas o colocan hashtags, así como invitaciones a conferencias, programas de radio, televisión, podcasts, talleres, entre otros.

Estas reglas fueron propuestas por el autor de la tesis después de realizar un análisis manual de 1000 documentos.

El etiquetado se realizó sobre una muestra de 6,716 documentos seleccionados mediante un muestreo aleatorio simple sin reemplazo. Se llevó a cabo de forma manual, utilizando una aplicación web desarrollada específicamente por el autor de esta tesis para este propósito como se muestra en la Figura 17. La aplicación fue diseñada para mostrar tuits almacenados en una base de datos de forma interactiva. Cada tuit se presentaba de manera individual, lo que permitía al usuario etiquetador asignar una categoría a cada uno de ellos. Se proporcionaron cuatro opciones de categoría: "Ofensa", "No ofensa", "Neutral" y "No útil". El botón "No útil" se utiliza para los textos vacíos o aquellos que carecían de un significado

coherente. Al seleccionar esta opción, los tuits correspondientes son eliminados del corpus. Los tuits etiquetados son guardados en la misma base de datos con la respectiva etiqueta de la clase y posteriormente exportados en el formato original de los documentos en un archivo CSV.

Figura 17. Aplicación web diseñada para el etiquetado de los datos utilizando una base de datos de tuits orientados a la comunidad LGBTIQ+



Una vez realizado el etiquetado en los tuits seleccionados, se llevó a cabo una validación aplicando el siguiente criterio: se seleccionaron aleatoriamente 364 tuits y otra persona volvió a etiquetarlos. Luego se compararon las etiquetas anteriores con las nuevas y no se encontraron diferencias. La cantidad de 364 tuits se determinó utilizando la fórmula de población finita, con un nivel de confianza del 95% y un margen de error del 5%.

El número de documentos en cada clase se muestra a continuación:

- Ofensa: 1,965 documentos

- No ofensa: 2,101 documentos

- Neutral: 2,650 documentos

Durante esta etapa, es evidente que existe un desbalance de clases, siendo la clase Neutral aquella que contiene la mayor cantidad de documentos etiquetados.

3.2.2 Preprocesamiento de textos

El preprocesamiento de los textos obtenidos de redes sociales es una etapa importante debido a la diversidad de formatos presentes, como emoticones, videos, imágenes, símbolos y caracteres especiales. Para llevar a cabo esta tarea, se siguieron los pasos propuestos por (López-Chau et al., 2020).

Las tareas realizadas en el preprocesamiento de textos incluyen:

- Eliminación del contenido no relevante, como imágenes o videos.
- Eliminación de stopword como preposiciones o conjunciones.
- Normalización del texto a UTF-8, lo que implica dar formato a los caracteres que se encuentren en una codificación diferente, como los caracteres ASCII.

  En la Tabla 2 se muestra un ejemplo de un mensaje en su formato original y el resultado después del preprocesamiento.

Tabla 2 Texto original vs texto pre-procesado

Texto de Entrada	Texto de Salida
'Fue un gusto cerrar nuestra campa<U+00F1>a con la alegr<U+00ED>a y el <U+00E1>nimo de los guanajuatenses. Desde Le<U+00F3>n, les pido que el 1<U+2026> https://t.co/y145kNcaG1'	‘Fue un gusto cerrar nuestra campaña con la alegría y el ánimo de los guanajuatenses. Desde León, les pido que el 1…”

- Identificación de las menciones de usuarios (@NombreDeUsuario), las cuales fueron reemplazadas por la palabra clave USERNAME.
- Identificación de los hashtags o temas (#), los cuales fueron reemplazados por la palabra clave TOPIC.

- Eliminación de las URL, las cuales comienzan con el texto "http".
- Eliminación de caracteres especiales, como $, %, ^, *, ¡, !, ¿,? y los signos de puntuación.
- Cambio de las vocales acentuadas por la misma vocal sin acentos.
- Eliminación de números.
- Eliminación de tags HTML, los cuales comienzan con los símbolos "<" y ">" respectivamente.

Es importante destacar que el procesamiento mencionado fue aplicado a la totalidad de los tuits seleccionados para la tesis. Para lograr esto de manera eficiente, se desarrolló un script en Python específicamente diseñado para llevar a cabo dicho procesamiento.

3.3 Extracción de características

Los documentos que se utilizaron en esta tesis fueron divididos en dos conjuntos distintos: uno destinado a pruebas y otro para entrenamiento. En total, se utilizaron 6,716 documentos en este proceso de división. El 80% de los documentos se asignó al conjunto de entrenamiento, mientras que el 20% restante se reservó para el conjunto de pruebas. Esta división estratégica permitió utilizar una cantidad significativa de datos para entrenar el modelo, asegurando que el modelo fuera capaz de generalizar y evaluar su rendimiento en datos no vistos previamente.

Antes de aplicar los algoritmos de aprendizaje supervisado, fue necesario realizar un proceso de extracción de características en el bloque de entrenamiento. Este proceso es esencial en el análisis de texto, ya que permite convertir los documentos de texto en representaciones numéricas o vectores de características comprensibles para los modelos de aprendizaje automático.

En esta tesis, se emplearon tres técnicas diferentes para la extracción de características:

- TF-IDF: El objetivo es destacar las palabras más relevantes para cada documento.
- Bolsa de Palabras: Esta técnica no considera el orden de las palabras, sino solo su ocurrencia.
- Hashing Vectorizer: Cada documento se representa mediante un vector donde solo se indican los índices correspondientes a las palabras presentes en el documento.

Una vez que se obtuvieron las representaciones vectoriales utilizando cada una de estas técnicas, se llevó a cabo una evaluación del rendimiento de los seis métodos de aprendizaje supervisado seleccionados en la tesis. Se analizó cómo cada combinación de técnica de extracción de características y método de aprendizaje afectaba la precisión, el recall y el F1-Score.

Este análisis exhaustivo permitió determinar cuál era la combinación más efectiva para la identificación de mensajes ofensivos orientados a la comunidad LGBTIQ+ en los documentos seleccionados. Se compararon los resultados obtenidos con cada técnica de extracción de características y método de aprendizaje, y se seleccionó la combinación que proporcionaba los mejores resultados en términos de las métricas definidas.

En el contexto de esta tesis, se determinó que la técnica de TF-IDF fue la más efectiva en la clasificación de sentimientos en los documentos. Esta técnica sobresalió debido a su capacidad para resaltar las palabras más relevantes en cada documento y capturar su importancia en la representación vectorial. Por lo tanto, se decidió utilizar esta técnica como base para los pasos posteriores del análisis.

Una vez que se obtienen los valores vectorizados de los documentos utilizando la técnica de TF-IDF, se guardan en un archivo CSV para su posterior utilización. Estos vectores representan de manera numérica las características relevantes de los documentos y se utilizarán como entrada para el modelo de aprendizaje automático en etapas posteriores del análisis de comentarios ofensivos.

3.3.1 Balanceo de clases

Durante la fase de análisis de los datos destinados al entrenamiento, se identificó un desbalance en las clases debido a la naturaleza de los datos recopilados de diversas fuentes en internet. Se observó una mayor prevalencia de mensajes de tipo Neutral en comparación con otras categorías. Esta disparidad en las clases planteaba un desafío, ya que los métodos de aprendizaje supervisado tienden a favorecer la clase con más instancias, lo que podría sesgar la clasificación y afectar la precisión y el rendimiento general del modelo.

Para abordar este desafío, se decidió utilizar la técnica SMOTE. Esta técnica permitió aumentar el número de instancias en la clase minoritaria, logrando así un equilibrio en la cantidad de instancias entre todas las clases. A través de SMOTE, se generaron instancias sintéticas basadas en los patrones existentes en la clase minoritaria, enriqueciendo de esta manera el conjunto de datos y evitando una sobre-representación de la clase mayoritaria. De esta forma, se logró mitigar el desbalance de clases y se obtuvo un conjunto de datos de entrenamiento más equitativo y representativo.

Es importante destacar que el proceso de aplicación de SMOTE se realizó exclusivamente en el conjunto de datos de entrenamiento, asegurando que la distribución original del conjunto de pruebas se mantuviera intacta. Esto garantizó una evaluación más precisa del modelo y su capacidad para generalizar en datos no vistos. Al preservar la distribución original en el conjunto de pruebas, pudimos simular condiciones similares a las del mundo real, donde el modelo se encontraría con datos desequilibrados. De esta manera, se pudo evaluar de manera más efectiva la capacidad del modelo para generalizar y clasificar correctamente las clases minoritarias en situaciones reales. Este enfoque nos brindó resultados más realistas y confiables en el contexto de la clasificación de sentimientos.

3.4 Generar y evaluar modelos predictivos

Los algoritmos de aprendizaje supervisado son fundamentales en la construcción de modelos de conocimiento para abordar problemas específicos. Estos algoritmos se basan en el uso de datos etiquetados, donde cada muestra tiene asignada una etiqueta que representa la clase o categoría a la que pertenece. La calidad y el manejo adecuado de los datos son aspectos críticos para obtener predicciones precisas y confiables (Vitiugin et al., 2021)(Vemuri, 2020).

En el contexto de esta tesis, se han considerado seis métodos de aprendizaje supervisado para el análisis de sentimientos: Redes neuronales (NN), Árboles de decisión (DT), Máquina de vectores de soporte (SVM), Naïve Bayes (NB), Regresión logística (LR) y Random Forest (RF). Estos métodos fueron seleccionados debido a su amplia aplicación y su capacidad para abordar problemas de clasificación con buenos resultados.

Cada método requiere un ajuste de hiperparámetros para optimizar su rendimiento. En esta tesis, se utilizó una técnica de búsqueda en cuadrícula con validación cruzada para encontrar los mejores valores de los hiperparámetros. Esta técnica implica explorar diferentes combinaciones de valores dentro de un rango amplio y evaluar el rendimiento del modelo utilizando validación cruzada.

Para esta tesis, se seleccionaron los hiperparámetros más comunes para cada método de aprendizaje supervisado. Se consideró un amplio rango de valores para cada hiperparámetro, lo que permitió una exploración de las diferentes combinaciones posibles. La validación cruzada se utilizó para evaluar el rendimiento de cada combinación de valores de hiperparámetros y determinar cuáles producían los mejores resultados.

La Tabla 3 muestra los resultados obtenidos a través de este proceso de ajuste de hiperparámetros para cada uno de los métodos considerados. Estos

resultados representan las combinaciones de hiperparámetros que lograron maximizar el rendimiento de cada modelo en la tarea de análisis de sentimientos.

Tabla 3 Configuración de hiperparámetros por búsqueda en cuadricula por validación cruzada

Clasificador	**Hiperparámetros**
NN	solver:'adam'
	activation:'tanh'
	hidden_layer_sizes:(10,30,10)
	alpha:0.0001
	learning_rate_'constant'
DT	max_depth:17
	min_samples_leaf:1
SVM	kernel:'rbf'
	gamma: 2.8571
NB	var_smoothing:2.3101e-05
LR	penalty:'l2'
	tol:1e-4
RF	max_depth:17

El ajuste de hiperparámetros es esencial para obtener el máximo rendimiento de los métodos de aprendizaje supervisado. Al encontrar los valores óptimos, se asegura que los modelos estén correctamente configurados y puedan generalizar de manera efectiva a datos no vistos previamente. Esta metodología rigurosa garantiza que se obtengan los mejores resultados posibles con cada método y se

logre un modelo de clasificación óptimo para el problema específico abordado en esta tesis.

3.5 Creación de lexicón orientado al contexto

El proceso de creación del lexicón de polaridad fue cuidadosamente diseñado para garantizar su calidad y utilidad en la tarea de identificación de textos ofensivos dirigidos a la comunidad LGBTIQ+. En primer lugar, se llevó a cabo la separación de todos los documentos basándose en la etiqueta asignada a cada uno de ellos. Esta clasificación permitió agrupar los documentos según el sentimiento identificado, ya sea "Ofensa", "No ofensa" o "Neutral".

Una vez completada la separación, se realiza un proceso de preprocesamiento en memoria para eliminar los StopWords, como preposiciones y conjunciones, presentes en cada bloque de texto. Este paso es fundamental para reducir el ruido y enfocarse en las palabras más relevantes para la clasificación de polaridad.

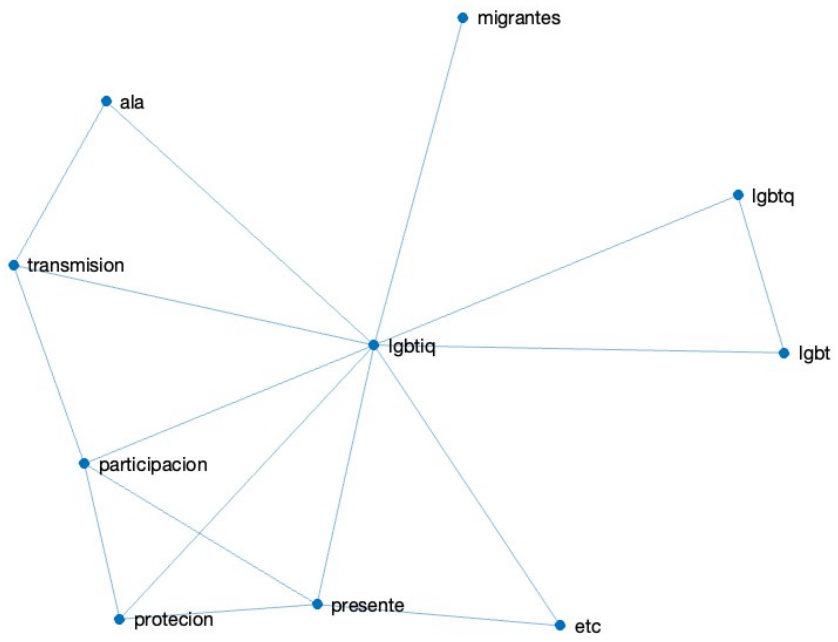
A continuación, se procede a obtener todas las palabras únicas presentes en cada clase y se almacenan en tres archivos CSV diferentes, cada uno identificado con el nombre de la clase correspondiente.

Posteriormente, se lleva a cabo un proceso de disyunción exclusiva entre conjuntos. El objetivo principal de este proceso es identificar las palabras que se encuentran exclusivamente en la lista de cada clase, es decir, aquellas palabras que no se comparten entre las clases. De esta manera, se obtiene un conjunto de palabras clave que resultan distintivas para cada clase, eliminando aquellas palabras que podrían generar ambigüedad o interferir en el análisis de polaridad.

Una vez obtenidas las palabras únicas, es necesario calcular el valor de polaridad de cada una de ellas para determinar su grado de ofensividad. Para este propósito, se utiliza una técnica de propagación de grafos basada en un artículo de (Velikovich et al., 2010) .

Se puede imaginar un grafo como una red de conexiones entre frases, donde cada palabra se representa como un punto en el grafo. Las conexiones entre las palabras se visualizan mediante líneas que las conectan. Estas líneas muestran la similitud entre las palabras en términos de su significado y contexto. En la Figura 18 se presenta un ejemplo de esta conexión utilizando la palabra " lgbtiq ", donde se muestran las palabras que están conectadas a ella en las frases y que suelen aparecer junto a esta palabra en la web. Esto nos permite obtener medidas de similitud entre las frases.

Figura 18 Representación del grafo generado, utilizando como ejemplo las conexiones realizadas con la palabra "lgbtiq"



En nuestra variante del método, se inicia con la inicialización de las semillas de polaridad de las palabras según su categoría. Los términos ofensivos se inicializan con un valor de 1.0, los no ofensivos con -1.0 y los neutrales con 0.0. Cabe mencionar que el autor original utiliza únicamente 2 clases. A continuación, se ejecuta un bucle en el que se propaga la polaridad a través de los nodos vecinos. La propagación se realiza identificando la mejor ruta desde cada nodo semilla hasta

cada nodo candidato, donde la mejor ruta se define como aquella con el peso más alto. En cada iteración, se actualizan las polaridades de cada frase no etiquetada como la suma ponderada de las polaridades de sus vecinos, utilizando el peso de cada vecino que representa la similitud del coseno entre la frase no etiquetada y el vecino. Cada nodo toma el promedio ponderado de los valores de polaridad de sus vecinos de la iteración anterior. El algoritmo se detiene cuando se alcanza la convergencia, es decir, cuando las polaridades de todas las frases no etiquetadas dejan de cambiar.

Una vez que se ha propagado la polaridad a través del grafo, se asigna una polaridad a cada frase candidata en función de su polaridad final. Si la polaridad final es mayor que el umbral establecido, se le asigna una polaridad positiva. Si es menor que el umbral, se le asigna una polaridad negativa. En caso de que sea igual al umbral, se le asigna un valor de 0, lo cual corresponde a la clase neutral. Este proceso se repite para todas las frases en el grafo, lo que resulta en la construcción de un léxico de polaridad a partir del grafo de similitud de frases.

Como resultado de este proceso, se obtiene una lista de palabras únicas con sus respectivas polaridades positivas (correspondientes a la clase Ofensivo), negativas (correspondientes a la clase No ofensivo) y valores de 0 (correspondientes a la clase Neutral). Este lexicón de polaridad es utilizado en etapas posteriores del análisis.

3.6 Adaptar técnicas basadas en lexicón utilizando el lexicón generado

3.6.1 Aumentando atributos usando lexicón

En el paso anterior de la metodología, se lleva a cabo la transformación del lexicón orientado a la comunidad LGBTIQ+ en su representación vectorial utilizando la técnica de bolsa de palabras, que se referirá como lexicón BoW de ahora en adelante. Los valores resultantes, junto con la representación TF-IDF de los documentos del conjunto de entrenamiento, se guardan en un archivo separado para su posterior reutilización sin necesidad de repetir el proceso de transformación.

La representación TF-IDF de los documentos seleccionados para el entrenamiento se combina con el lexicón BoW, agregándolo como parte de los atributos en la representación TF-IDF y aumentando así el número total de atributos. Por ejemplo, si la representación TF-IDF inicialmente tenía 10 atributos y el lexicón BoW contenía 5, el modelo resultante contendrá un total de 15 atributos.

Con el objetivo de mejorar el rendimiento del modelo, se realiza la selección de las características más relevantes utilizando el criterio de selección chi2 (chi cuadrada). Este criterio permite identificar aquellas características que tienen una relación significativa con la variable objetivo y que aportan más información. Utilizando chi2, se seleccionan las 1000 mejores características para el modelo.

El nuevo modelo resultante, que combina el lexicón BoW y la representación TF-IDF de los documentos, se utiliza como entrada para el entrenamiento de los modelos de aprendizaje supervisado.

3.7 Generación del modelo empleando un enfoque de ensamble

En el paso final del proceso de identificación de mensajes ofensivos dirigidos a la comunidad LGBTIQ+, se utiliza el modelo combinado de Lexicón BoW y TF-IDF como entrada para los métodos seleccionados de aprendizaje supervisado con su configuración correspondiente.

En este paso, se analizan manualmente los resultados utilizando el F1-Score como métrica de comparación. Se evalúan los métodos de aprendizaje supervisado y se selecciona aquel que tenga dos o más categorías con valores más altos como el mejor disponible para este contexto.

CAPÍTULO 4

4. Resultados

Para seleccionar el método de vectorización a utilizar, se realizó un experimento preliminar. En este experimento, se recolectaron 5,000 tuits mediante una técnica de muestreo simple sin reemplazo. Estos tuits fueron etiquetados según los criterios previamente establecidos. A continuación, se presenta la distribución de documentos por cada clase:

- Ofensa: 365
- No Ofensa: 200
- Neutral: 4435

Es importante destacar que este proceso permitió obtener una muestra representativa de los diferentes tipos de tuits en nuestro conjunto de datos.

Al tener una cantidad significativa de tuits en la clase “Neutral”, se decidió llevar a cabo un equilibrio de clases utilizando la técnica SMOTE. Esto nos permitió incrementar la cantidad de ejemplos en las clases minoritarias y asegurar un conjunto de datos balanceados para el entrenamiento de los modelos, igualando la cantidad de documentos a 4435 por cada clase.

Se llevó a cabo una comparación del rendimiento de los seis clasificadores en la tarea de detección de lenguaje ofensivo en textos en Español. Esta comparación se realizó utilizando tres métricas clave: precisión, exhaustividad y F1-Score. Además, se evaluaron tres tipos diferentes de técnicas de vectorización: TF-IDF, Bolsa de Palabras y Hashing Vectorizer. Los resultados obtenidos se presentan en la Tabla 4, donde se utilizó una técnica de 10 validaciones cruzadas(10 cross-validation).

En la Tabla 4, se destacan en negrita los mejores resultados en términos de F1-Score. Estos resultados representan el desempeño más sobresaliente en la identificación de lenguaje ofensivo en textos en Español.

Esta evaluación exhaustiva permite seleccionar el método de vectorización más efectivo y los clasificadores más adecuados para abordar esta tarea específica. Los resultados obtenidos proporcionan una base sólida para tomar decisiones informadas sobre las estrategias a seguir en futuros análisis de lenguaje ofensivo.

Tabla 4 Comparativa de los diferentes tipos de vectorización y clasificadores propuestos en términos de F1-Score

Clases	Clasificador	TF-IDF	Hashing Vectorizer	Bag of Words
Ofensa	NN	0.83	0.88	0.70
	DT	0.95	0.87	0.70
	SVM	**1.00**	**1.00**	0.77
	NB	0.98	0.69	0.93
	LR	**1.00**	0.72	0.83
	RF	0.95	0.97	0.79
No Ofensa	NN	0.12	0.75	0.13
	DT	0.89	0.78	0.69
	SVM	**0.99**	**0.99**	0.93
	NB	0.97	0.52	0.93
	LR	**0.99**	0.49	0.83
	RF	0.93	0.94	0.79

Clases	Clasificador	TF-IDF	Hashing Vectorizer	Bag of Words
Neutral	NN	0.81	0.94	0.62
	DT	0.94	0.86	0.59
	SVM	**1.00**	0.99	0.93
	NB	0.99	0.71	0.76
	LR	0.99	0.62	0.72
	RF	0.98	0.98	0.67

Durante el desarrollo del experimento, se determinó que la técnica de vectorización más efectiva para esta tesis es TF-IDF. Es importante destacar que se identificó una tendencia hacia la clase "Neutral" en los datos recopilados. Aunque esta validación arrojó resultados aceptables en la clasificación, es crucial tener en cuenta que la gran mayoría de los datos utilizados fueron generados sintéticamente, lo que implica que esta validación no refleja una situación 100% real.

Para abordar esta limitación, se tomó la decisión de utilizar un conjunto de datos completamente nuevo y balanceado en el experimento final y en la creación del léxico de polaridad. Se buscó que la cantidad de documentos en cada una de las clases fuera muy similar, lo que garantiza una representación equilibrada de las diferentes categorías.

El nuevo conjunto de datos se conformó por un total de 6,716 documentos, distribuidos de la siguiente manera:

- Ofensa: 1,965 documentos
- No ofensa: 2,101 documentos
- Neutral: 2,650 documentos

Aunque ahora los datos están más equilibrados en términos de cantidad, se decidió implementar la técnica SMOTE para realizar el balanceo de clases. Sin embargo, a diferencia del experimento anterior, en esta ocasión se generaron datos sintéticos en menor cantidad para el balanceo. Por lo tanto, se asume que estos datos sintéticos no interferirán significativamente en la obtención de las métricas utilizadas y no afectarán la validez de los resultados obtenidos.

Con el nuevo corpus balanceado, se procede a realizar la partición de los datos, asignando un 20% para pruebas y un 80% para entrenamiento. Los datos de entrenamiento se transforman en su representación vectorial utilizando la técnica TF-IDF como se mencionó. Los valores resultantes de esta vectorización se almacenan en un archivo CSV para su posterior utilización en el proceso de entrenamiento y evaluación de los modelos.

Posteriormente, se lleva a cabo un experimento utilizando estos datos y los métodos de aprendizaje supervisado, con el objetivo de comparar su rendimiento con el uso del lexicón generado.

Para configurar los métodos de aprendizaje supervisado, se emplea una técnica de búsqueda exhaustiva en rejilla. Esta técnica implica probar sistemáticamente diferentes combinaciones de parámetros para cada clasificador. Se establece una lista de valores candidatos para uno o más parámetros de cada clasificador, y se evalúa el rendimiento utilizando validación cruzada. Se consideran métricas como la exactitud, precisión, F1-score. Los detalles de la configuración óptima encontrada para estos clasificadores se mencionan en el capítulo 3 de esta tesis, proporcionando información detallada sobre las configuraciones que brindan los mejores resultados.

Utilizando la configuración óptima obtenida, se procede a validar el corpus seleccionado utilizando la vectorización TF-IDF en conjunto con los métodos de aprendizaje supervisado elegidos en esta investigación. Los resultados obtenidos se presentan en la Tabla 5, donde se exhiben las métricas de precisión, exhaustividad y F1-score. Los mejores resultados se resaltan en negritas, indicando

el desempeño más destacado en la detección de lenguaje ofensivo en textos en Español. Estos resultados son fundamentales para respaldar la toma de decisiones informadas en futuros análisis de lenguaje ofensivo y proporcionan una base sólida para el desarrollo de estrategias efectivas en este campo de investigación.

Tabla 5 Desempeño de las predicciones a un corpus de 6,716 documentos utilizando TF-IDF

Clases	Clasificador	Precision	Recall	F1-Score
Ofensa	**NN**	**0.76**	**0.75**	**0.75**
	DT	0.61	0.81	0.70
	SVM	0.69	0.78	0.73
	NB	0.71	0.22	0.33
	LR	0.58	0.86	0.70
	RF	0.62	0.84	0.71
No Ofensa	NN	0.68	0.70	0.69
	DT	0.81	0.62	0.70
	SVM	0.73	0.62	0.67
	NB	0.69	0.25	0.36
	LR	0.83	0.62	0.71
	RF	**0.83**	**0.66**	**0.73**
Neutral	**NN**	**0.72**	**0.72**	**0.72**
	DT	0.56	0.39	0.46

Clases	Clasificador	Precision	Recall	F1-Score
	SVM	0.73	0.68	0.70
	NB	0.33	0.92	0.49
	LR	0.58	0.27	0.37
	RF	0.59	0.37	0.46

El lexicón generado a partir del procesamiento del algoritmo basado en grafos, aplicado al conjunto de datos de entrenamiento, consta de 3,315 términos etiquetados como "Neutrales", 3,766 como "No Ofensivos" y 2,564 como "Ofensivos".

El funcionamiento normal de un lexicón consiste en realizar la sumatoria de los valores ponderados de cada palabra por documento analizado. El resultado final de esta sumatoria determina el sentimiento asociado al documento. Sin embargo, al realizar pruebas implementando esta metodología tradicional, se pudo determinar que los valores de los resultados mostrados en la tabla anterior no presentaban cambios significativos en comparación con los nuevos resultados obtenidos mediante el lexicón.

Ante esta situación, se decidió explorar una implementación diferente para aprovechar al máximo el lexicón generado. En lugar de utilizarlo directamente en la sumatoria de valores ponderados, se optó por transformarlo en una representación de bolsa de palabras.

Esta representación de bolsa de palabras consiste en contar la frecuencia de aparición de cada término del lexicón en cada documento. Posteriormente, se combinó esta representación de bolsa de palabras con la técnica de vectorización TF-IDF, que asigna un peso a cada término en función de su relevancia en el documento y en el corpus en general.

La combinación de la representación de bolsa de palabras y TF-IDF mostró mejoras significativas en los resultados. Al evaluar los clasificadores utilizando esta combinación, se obtuvieron métricas más favorables en la clasificación de los documentos. Los detalles y los resultados obtenidos se presentan en la Tabla 6.

En la Tabla 6, se resaltan en negritas los valores con mejor desempeño, lo que indica las mejoras más destacadas logradas mediante la utilización conjunta de la representación de bolsa de palabras y TF-IDF. Estos resultados refuerzan la efectividad de esta combinación para el análisis de lenguaje ofensivo orientado a la comunidad LBGTIQ+.

Tabla 6 Desempeño de las predicciones a un corpus de 6,716 documentos utilizando TF-IDF con aumento de atributos

Clases	Clasificador	Precision	Recall	F1-Score
Ofensa	NN	0.90	0.89	0.89
	DT	0.88	0.88	0.88
	SVM	0.82	0.92	0.87
	NB	0.61	0.96	0.75
	LR	0.87	0.90	0.89
	RF	**0.91**	**0.90**	**0.90**
No Ofensa	NN	0.81	0.83	0.82
	DT	0.83	0.84	0.83
	SVM	0.86	0.73	0.79
	NB	0.90	0.40	0.55
	LR	0.85	0.82	**0.84**
	RF	0.84	0.84	**0.84**

Clases	Clasificador	Precision	Recall	F1-Score
Neutral	**NN**	**0.85**	**0.85**	**0.85**
	DT	0.80	0.80	0.80
	SVM	0.86	0.80	0.83
	NB	0.77	0.35	0.48
	LR	0.86	0.83	0.85
	RF	0.82	0.84	0.83

CAPÍTULO 5

5. Conclusiones

En síntesis, esta investigación ha logrado su objetivo central al desarrollar un método efectivo para la identificación de lenguaje ofensivo dirigido a la comunidad LGBTIQ+. La meta fue alcanzada a través de la evaluación exhaustiva de seis métodos de aprendizaje supervisado, respaldados por un lexicón específico creado a partir de un corpus de documentos reales de la plataforma Twitter.

La elección estratégica de Twitter como fuente principal de datos proporcionó ventajas sustanciales al permitir una adquisición eficiente de información mediante un script automatizado en Python. Esta decisión facilitó la obtención rápida de documentos reales centrados en el tema, contribuyendo significativamente a la calidad del corpus.

El proceso de etiquetado se simplificó mediante una aplicación web especializada, mejorando la eficiencia al eliminar información no pertinente. El preprocesamiento de textos, con una atención especial a la limpieza de datos, resultó crucial para garantizar la interpretación precisa de los algoritmos supervisados.

El desafío de obtener un conjunto balanceado se abordó con éxito mediante una estrategia de equilibrio de corpus, que aseguró la representatividad de las categorías, ya que, debido a que el corpus se construyó a partir de datos provenientes de internet, los documentos destacaban por ser de tipo “neutral” en mayor medida.

La experimentación con la vectorización TF-IDF, enriquecida con un lexicón específico, impulsó un progreso significativo, alcanzando un aumento del 15% en la precisión de la clasificación. El método de Bosque Aleatorio emergió como el clasificador más eficaz para este estudio, con puntuaciones entre 0.84 y 0.90 en términos de F1-Score.

Cada objetivo establecido dentro de la tesis fue alcanzado de manera exitosa, a modo de resumen, se listan a continuación los principales puntos por objetivo:

Generar y evaluar modelos predictivos para identificar lenguaje ofensivo en textos en Español: Este objetivo se cumplió mediante la implementación y evaluación de diversos modelos, siendo el Bosque Aleatorio el más eficaz en nuestros experimentos. Esta eficacia se resalta al comparar los resultados con el estudio similar realizado por (Plaza-del-Arco et al., 2021), donde nuestro enfoque demostró un rendimiento superior.

Proponer una metodología para la creación de un lexicón orientado a la identificación de lenguaje ofensivo: La metodología desarrollada, fundamentada en relaciones entre palabras y prescindiendo de WordNet, posibilitó la creación de un lexicón específico que demostró una mejora significativa en la precisión. Al contrastar con estudios similares, como lo es (Pamungkas et al., 2020), nuestra aproximación resalta por su efectividad y aporte distintivo al campo.

Adaptar las técnicas basadas en lexicón de análisis de sentimientos: La adaptación fue exitosa, evidenciada por la mejora en la precisión de la identificación de lenguaje ofensivo mediante la inclusión del lexicón específico.

Combinar los mejores modelos predictivos y la técnica basada en lexicón mediante un enfoque de ensamble: La estrategia de ensamble, integrando el Bosque Aleatorio y enriqueciendo TF-IDF con el lexicón, resultó ser eficaz, con puntuaciones F1-Score notables.

En consecuencia, esta investigación no solo logró sus objetivos establecidos, sino que también ha contribuido significativamente al avance en la detección precisa de lenguaje ofensivo en entornos específicos de diversidad. Los resultados respaldan la relevancia y aplicabilidad de este trabajo en el ámbito de la inteligencia artificial y análisis de sentimientos.

En perspectiva de futuras investigaciones, se identifican diversas áreas que podrían enriquecer y ampliar la contribución de este estudio. Estos posibles caminos incluyen:

- Implementación de Técnicas de Clustering para agilizar la etiquetación de documentos, permitiendo así una clasificación eficiente de un mayor volumen en un tiempo reducido.
- Exploración de Word Embeddings con el objetivo de obtener representaciones más cohesivas y menos dispersas de los documentos.
- Evaluación de Métodos Alternativos para Lexicones para ampliar y diversificar las herramientas disponibles para la identificación de lenguaje ofensivo.
- Adaptación a Entornos Multilingües, explorando la aplicabilidad y eficacia de las metodologías desarrolladas en contextos que involucren otros idiomas

Estas áreas de investigación prospectivas no solo permitirían una evolución natural de esta tesis, sino que también contribuirían a la mejora continua de las técnicas y herramientas desarrolladas, consolidando aún más su impacto en el campo de la detección de lenguaje ofensivo.

6. Referencias

Bashar, M. A., Nayak, R., Luong, K., & Balasubramaniam, T. (2021). Progressive domain adaptation for detecting hate speech on social media with small training set and its application to COVID-19 concerned posts. *Social Network Analysis and Mining*, *11*(1), 69. https://doi.org/10.1007/s13278-021-00780-w

Ben-Hur, A., & Weston, J. (2010). A user's guide to support vector machines. *Methods in Molecular Biology (Clifton, N.J.)*, *609*, 223–239. https://doi.org/10.1007/978-1-60327-241-4_13

Bhowmik, N. R., Arifuzzaman, M., & Mondal, M. R. H. (2022). Sentiment analysis on Bangla text using extended lexicon dictionary and deep learning algorithms. *Array*, *13*, 100123. https://doi.org/10.1016/j.array.2021.100123

Brownlee, J. (2020). Imbalanced Classification with Python: Choose Better Metrics. In *Balance Skewed Classes, and Apply Cost-Sensitive Learning* (Vol. 1). Machine Learning Mastery. https://books.google.com/books?hl=en&lr=&id=jaXJDwAAQBAJ&oi=fnd&pg=PP1&dq=%22intrusion+detection%22%7C%22anomaly+intrusion+detection%22%7C%22network+intrusion+detection%22%7C%22anomaly+based+network+intrusion+detection%22+%22imbalance+problem%22%7C%22imba

Chandrasekharan, E., Pavalanathan, U., Srinivasan, A., Glynn, A., Eisenstein, J., & Gilbert, E. (2017). You Can't Stay Here: The Efficacy of Reddit's 2015 Ban Examined Through Hate Speech. *Proc. ACM Hum.-Comput. Interact.*, *1*(CSCW). https://doi.org/10.1145/3134666

Chapelle, O., Scholkopf, B., & Zien Eds., A. (2009). Semi-Supervised Learning. *IEEE Transactions on Neural Networks*, *20*(3), 542. https://doi.org/10.1109/TNN.2009.2015974

Chih-Wei Hsu, Chih-Chung Chang, and C.-J. L. (2008). A Practical Guide to

Support Vector Classification. *BJU International*, *101*(1), 1396–1400. http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

Chiu, S., & Tavella, D. (2020). Introduction to Data Mining. In *Data Mining and Market Intelligence for Optimal Marketing Returns* (US ed). Addison Wesley. https://doi.org/10.4324/9780080878096-12

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Cruz, R. M. O., de Sousa, W. V., & Cavalcanti, G. D. C. (2022). Selecting and combining complementary feature representations and classifiers for hate speech detection. *Online Social Networks and Media*, *28*, 100194. https://doi.org/10.1016/j.osnem.2021.100194

Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International Conference on Web and Social Media, ICWSM 2017* (pp. 512–515). https://doi.org/10.1609/icwsm.v11i1.14955

Del Bosque, L. P., & Garza, S. E. (2014). Aggressive Text Detection for Cyberbullying. In A. Gelbukh, F. C. Espinoza, & S. N. Galicia-Haro (Eds.), *Human-Inspired Computing and Its Applications* (pp. 221–232). Springer International Publishing.

Dhungana Sainju, K., Mishra, N., Kuffour, A., & Young, L. (2021). Bullying discourse on Twitter: An examination of bully-related tweets using supervised machine learning. *Computers in Human Behavior*, *120*, 106735. https://doi.org/https://doi.org/10.1016/j.chb.2021.106735

Eronen, J., Ptaszynski, M., Masui, F., Smywiński-Pohl, A., Leliwa, G., & Wroczynski, M. (2021). Improving classifier training efficiency for automatic cyberbullying detection with Feature Density. *Information Processing & Management*, *58*(5), 102616.

https://doi.org/https://doi.org/10.1016/j.ipm.2021.102616

Fang, Y., Yang, S., Zhao, B., & Huang, C. (2021). Cyberbullying Detection in Social Networks Using Bi-GRU with Self-Attention Mechanism. *Information*, *12*(4). https://doi.org/10.3390/info12040171

Florio, K., Basile, V., Polignano, M., Basile, P., & Patti, V. (2020). Time of Your Hate: The Challenge of Time in Hate Speech Detection on Social Media. *Applied Sciences*, *10*(12). https://doi.org/10.3390/app10124180

Fortuna, P., Soler-Company, J., & Wanner, L. (2021). How well do hate speech, toxicity, abusive and offensive language classification models generalize across datasets? *Information Processing & Management*, *58*(3), 102524. https://doi.org/https://doi.org/10.1016/j.ipm.2021.102524

George, A. (2022). *Python Text Mining Perform Text Processing, Word Embedding, Text Classification and Machine Translation* (P. Publications (ed.); Primera Ed). PBP Publications. www.bpbonline.com

Halim, Z., Waqar, M., & Tahir, M. (2020). A machine learning-based investigation utilizing the in-text features for the identification of dominant emotion in an email. *Knowledge-Based Systems*, *208*, 106443. https://doi.org/https://doi.org/10.1016/j.knosys.2020.106443

Ilie, V. I., Truica, C. O., Apostol, E. S., & Paschke, A. (2021). Context-Aware Misinformation Detection: A Benchmark of Deep Learning Architectures Using Word Embeddings. *IEEE Access*, *9*, 162122–162146. https://doi.org/10.1109/ACCESS.2021.3132502

Jha, V. K., Hrudya, P., Vinu, N. V., Vijayan, V., & Prabaharan, P. (2020). DHOT-Repository and Classification of Offensive Tweets in the Hindi Language. *Procedia Computer Science*, *171*, 2324–2333. https://doi.org/10.1016/J.PROCS.2020.04.252

Kalita, D. J. (2015). *Supervised and Unsupervised Document Classification-A*

survey.

Kanclerz, K., Milkowski, P., & Kocon, J. (2020). Cross-lingual deep neural transfer learning in sentiment analysis. *Procedia Computer Science*, *176*, 128–137. https://doi.org/10.1016/J.PROCS.2020.08.014

Kemp, S. (2022). *DIGITAL 2022: Global Overview Report*. https://datareportal.com/reports/digital-2022-global-overview-report

Khanday, A. M. U. D., Rabani, S. T., Khan, Q. R., & Malik, S. H. (2022). Detecting twitter hate speech in COVID-19 era using machine learning and ensemble learning techniques. *International Journal of Information Management Data Insights*, *2*(2), 100120. https://doi.org/https://doi.org/10.1016/j.jjimei.2022.100120

Kocoń, J., Figas, A., Gruza, M., Puchalska, D., Kajdanowicz, T., & Kazienko, P. (2021). Offensive, aggressive, and hate speech analysis: From data-centric to human-centered approach. *Information Processing & Management*, *58*(5), 102643. https://doi.org/https://doi.org/10.1016/j.ipm.2021.102643

Kucuk, S. U. (2016). *What Is Hate?* Brand Hate. https://doi.org/10.1007/978-3-319-41519-2_1

Lin, S.-Y., Kung, Y.-C., & Leu, F.-Y. (2022). Predictive intelligence in harmful news identification by BERT-based ensemble learning model with text sentiment analysis. *Information Processing & Management*, *59*(2), 102872. https://doi.org/https://doi.org/10.1016/j.ipm.2022.102872

Liu, J., Yang, Y., Fan, X., Ren, G., Yang, L., & Ning, Q. (2022). Offensive-Language Detection on Multi-Semantic Fusion Based on Data Augmentation. *Applied System Innovation*, *5*(1). https://doi.org/10.3390/asi5010009

López-Chau, A., Valle-Cruz, D., & Sandoval-Almazán, R. (2020). Sentiment Analysis of Twitter Data Through Machine Learning Techniques. In M. Ramachandran & Z. Mahmood (Eds.), *Software Engineering in the Era of*

Cloud Computing (pp. 185–209). Springer International Publishing. https://doi.org/10.1007/978-3-030-33624-0_8

Mohapatra, S. K., Prasad, S., Bebarta, D. K., Das, T. K., Srinivasan, K., & Hu, Y.-C. (2021). Automatic Hate Speech Detection in English-Odia Code Mixed Social Media Data Using Machine Learning Techniques. *Applied Sciences*, *11*(18). https://doi.org/10.3390/app11188575

Mubarak, H., Darwish, K., & Magdy, W. (2017). Abusive Language Detection on Arabic Social Media. *Proceedings of the First Workshop on Abusive Language Online*, 52–56. https://doi.org/10.18653/v1/W17-3008

Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., & Chang, Y. (2016). Abusive language detection in online user content. *25th International World Wide Web Conference, WWW 2016*, 145–153. https://doi.org/10.1145/2872427.2883062

Ortega-Bueno, R., Rosso, P., & Medina Pagola, J. E. (2022). Multi-view informed attention-based model for Irony and Satire detection in Spanish variants. *Knowledge-Based Systems*, *235*, 107597. https://doi.org/10.1016/J.KNOSYS.2021.107597

Pamungkas, E. W., Basile, V., & Patti, V. (2020). Misogyny Detection in Twitter: a Multilingual and Cross-Domain Study. *Information Processing & Management*, *57*(6), 102360. https://doi.org/https://doi.org/10.1016/j.ipm.2020.102360

Park, J. H., & Fung, P. (2017). *One-step and Two-step Classification for Abusive Language Detection on Twitter*. arXiv. https://doi.org/10.48550/ARXIV.1706.01206

Perera, A., & Fernando, P. (2021). Accurate Cyberbullying Detection and Prevention on Social Media. *Procedia Computer Science*, *181*, 605–611. https://doi.org/10.1016/J.PROCS.2021.01.207

Pérez-Landa, G. I., Loyola-González, O., & Medina-Pérez, M. A. (2021). An explainable artificial intelligence model for detecting xenophobic tweets.

Applied Sciences (Switzerland), *11*(22), 128–137. https://doi.org/10.3390/app112210801

Perifanos, K., & Goutsos, D. (2021). Multimodal Hate Speech Detection in Greek Social Media. *Multimodal Technologies and Interaction*, *5*(7). https://doi.org/10.3390/mti5070034

Plaza-del-Arco, F. M., Molina-González, M. D., Ureña-López, L. A., & Martín-Valdivia, M. T. (2021). Comparing pre-trained language models for Spanish hate speech detection. *Expert Systems with Applications*, *166*, 114120. https://doi.org/https://doi.org/10.1016/j.eswa.2020.114120

Plaza-Del-Arco, F. M., Molina-González, M. D., Ureña-López, L. A., & Martín-Valdivia, M. T. (2020). Detecting Misogyny and Xenophobia in Spanish Tweets Using Language Technologies. *ACM Transactions on Internet Technology*, *20*(2). https://doi.org/10.1145/3369869

Pronoza, E., Panicheva, P., Koltsova, O., & Rosso, P. (2021). Detecting ethnicity-targeted hate speech in Russian social media texts. *Information Processing & Management*, *58*(6), 102674. https://doi.org/https://doi.org/10.1016/j.ipm.2021.102674

Sengupta, A., Bhattacharjee, S. K., Akhtar, M. S., & Chakraborty, T. (2022). Does aggression lead to hate? Detecting and reasoning offensive traits in hinglish code-mixed texts. *Neurocomputing*, *488*, 598–617. https://doi.org/https://doi.org/10.1016/j.neucom.2021.11.053

Sharma, S. S., & Dutta, G. (2021). SentiDraw: Using star ratings of reviews to develop domain specific sentiment lexicon for polarity determination. *Information Processing and Management*, *58*(1), 102412. https://doi.org/10.1016/j.ipm.2020.102412

Sreelakshmi, K., Premjith, B., & Soman, K. P. (2020). Detection of Hate Speech Text in Hindi-English Code-mixed Data. *Procedia Computer Science*, *171*,

737–744. https://doi.org/10.1016/J.PROCS.2020.04.080

Stats, I. L. (2021). *Twitter Usage Statistics*. https://www.internetlivestats.com/twitter-statistics/

Sulis, E., Irazú Hernández Farías, D., Rosso, P., Patti, V., & Ruffo, G. (2016). Figurative messages and affect in Twitter: Differences between irony, sarcasm and not. *Knowledge-Based Systems*, *108*, 132–143. https://doi.org/https://doi.org/10.1016/j.knosys.2016.05.035

Tan, P.-N., Steinbach, M., Karpatne, A., & Kumar, V. (2018). *Introduction to Data Mining (2nd Edition)* (2nd ed.). Pearson.

Tang, D., Wei, F., Qin, B., Zhou, M., & Liu, T. (2014). Building large-scale twitter-specific sentiment lexicon : A representation learning approach. *COLING 2014 - 25th International Conference on Computational Linguistics, Proceedings of COLING 2014: Technical Papers*, 172–182.

Theobald, O. (2020). *Machine Learning for Absolute Beginners: A Plain English Introduction (Third Edition) (Machine Learning with Python for Beginners Book 1)*. Scatterplot Press. https://ammroc-files.edgegroup.ae/s3fs-public/jobs/pdf-machine-learning-for-absolute-beginners-a-plain-english-introduc-oliver-theobald-pdf-download-free-book-63cb5bb.pdf

Tiara, Sabariah, M. K., & Effendy, V. (2015). Sentiment analysis on Twitter using the combination of lexicon-based and support vector machine for assessing the performance of a television program. *2015 3rd International Conference on Information and Communication Technology (ICoICT)*, 386–390. https://doi.org/10.1109/ICoICT.2015.7231456

Vemuri, V. K. (2020). The Hundred-Page Machine Learning Book. In *Journal of Information Technology Case and Application Research* (Vol. 22, Issue 2). Andriy Burkov. https://doi.org/10.1080/15228053.2020.1766224

Vinet, L., & Zhedanov, A. (2011). A “missing” family of classical orthogonal

polynomials. In *Journal of Physics A: Mathematical and Theoretical* (Vol. 44, Issue 8). O'Reilly Media. https://doi.org/10.1088/1751-8113/44/8/085201

Vrysis, L., Vryzas, N., Kotsakis, R., Saridou, T., Matsiola, M., Veglis, A., Arcila-Calderón, C., & Dimoulas, C. (2021). A web interface for analyzing hate speech. *Future Internet*, *13*(3), 80. https://doi.org/10.3390/fi13030080

Wang, Z., Yin, Z., & Argyris, Y. A. (2021). Detecting Medical Misinformation on Social Media Using Multimodal Deep Learning. *IEEE Journal of Biomedical and Health Informatics*, *25*(6), 2193–2203. https://doi.org/10.1109/JBHI.2020.3037027

Willard, N. E. (2007). Cyberbullying and cyberthreats: Responding to the challenge of online social aggression, threats, and distress. In *Cyberbullying and cyberthreats: Responding to the challenge of online social aggression, threats, and distress.* Research Press.

Williams, C. K. I. (2003). Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. In *Journal of the American Statistical Association* (Vol. 98, Issue 462). MIT Press. https://doi.org/10.1198/jasa.2003.s269

Zhang, Y., & Wallace, B. (2015). *A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification*. http://arxiv.org/abs/1510.03820

Žižka, J., Dařena, F., & Svoboda, A. (2019). Text Mining with Machine Learning. In *Text Mining with Machine Learning*. Crc Press. https://doi.org/10.1201/9780429469275