Universidad de las Ciencias Informáticas

Facultad 2

Clasificación automática de la polaridad en los comentarios que realizan los usuarios en Cubadebate

Trabajo de diploma presentado para optar por el título de Ingeniero en Ciencias Informáticas

Autora: Camila González Nápoles

Tutores: MSc. Eliana Bárbara Ril Valentin

MSc. Héctor Raúl González Diez

La Habana, 5 de junio de 2019

Declaración de autoría

Declaro por este medio que yo Camila González Nápoles, con carnet de identidad 96110707538, soy la autora principal del presente trabajo de diploma, que tiene por título: **Clasificación automática de la polaridad en los comentarios que realizan los usuarios en Cubadebate**, y que autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste firman la presente a los ____ días del mes de _____ del año _____.

_____

Camila González Nápoles

_____ _____

MSc. Eliana Bárbara Ril Valentin MSc. Héctor Raúl González Diez

Dedicatoria

Camila González Nápoles

Dedicárselo en primer lugar a mis padres pues han sido mi apoyo, mi pi constante, por su amor incondicional y la persona que soy ahora se los debo a ellos.

A mi hermano, que ha significado mi libro de soluciones, gracias por tu paciencia infinita y por siempre estar ahí con los brazos abiertos.

A mi esposo, que más decir que a él le debo mi mejor experiencia en la vida, lo mejor que me ha ocurrido. Gracias por sus consejos y su ayuda en esta investigación, el aguantar conmigo las derrotas y celebrar mis victorias.

A mi abuela Teresita que aunque se que quisiera estar aquí ahora y mi abuela Josefina que la vida nos privó de su presencia hoy, se sentirían orgullosas y llorarían de la emoción por mi.

A mi alocada cuñada Yahi, una parte importante de nuestra familia y la queremos mucho.

A mis abuelos de acogida Feri y Víctor, que me han dado su apoyo y el amor de abuelos.

Y a mi brujilla, mi suegra gracias por tus consejos y ser otra madre para mi.

Agradecimientos

Camila González Nápoles

Agradezco en primer lugar a mis tutores, a Héctor por confiar en mi y darme este tema a desarrollar ayudándome a mi crecimiento personal y profesional y a Baby por sea cual hubiese sido el tema me hubiera ayudado, desde segundo año siendo mi profesora se lo dije, que ella algún día sería mi tutora, la considero un modelo a seguir.

Agradezco a todos los profesores de la UCI, que de alguna manera u otra han formado parte de mi formación.

Agradezco a todos mis amigos de la UCI, que han hecho que este viaje haya sido una de las mejores expriencias de mi vida. Principalmente a Eleany, amiga y hermana desde el primer día que entramos por esa puerta, Liana y Arianna que aunque no hayan proseguido con nosotros son amigas y hermanitas que quedaron para la vida y Chabelly inseparables desde que nos juntamos, con la que he compartido todo. No existe nada con los que pueda agradecer por estar estos años juntas.

A José Ángel, Raudel y Roman amigos que me han ayudado en todo lo que les he pedido incondicionalmente y mira que puedo llegar a ser bastante insistente.

A mis amigas de la vida Elisa García, Sandra, Lianet y Elisa Perdomo, han sido mi propia familia con las que he compartido momentos únicos a lo largo de mi vida y siempre han estado estado ahí para mí.

Por último, agradecer a mis amigos de la universidad por todo lo que hemos aprendido los unos de los otros, por las experiencias vividas que nos ayudaron a crecer, especialmente a Alexis, Leyan, Vega, Pedro, Luisma.

Resumen

La minería de opinión es un proceso de extracción de nuevos conocimientos y datos textuales no estructurados mediante los métodos automáticos de detección y extracción de opiniones. El principal problema de los datos que se recopilan del sitio web de noticias Cubadebate es que se encuentran en forma no estructurada, lo que dificulta la identificación de la opinión pública y a su vez el sentimiento que transmiten en ella. El objetivo de esta investigación se centró en clasificar automáticamente la polaridad para determinar la intencionalidad de los usuarios a través de sus comentarios en el sitio web de noticias Cubadebate. Para ello, en la presente investigación se realizó un estudio sobre los principales algoritmos de procesamiento del lenguaje natural y minería de opinión para el análisis de sentimientos utilizando la herramienta de aprendizaje automático máquina de soporte vectorial. Además, se siguieron los pasos propuestos por Fayyad para descubrir conocimiento, realizando un procesamiento de los 5 artículos más comentados del sitio web de noticias Cubadebate y así conocer si sus usuarios tenían valoraciones positivas, negativas o neutras respecto a los artículos publicados. Se realizaron las pruebas Accuracy score, Precision-Recall y Predict, en aras de verificar la eficacia de los algoritmos escogidos.

Palabras clave: análisis de sentimiento, conocimiento, minería de opinión, procesamiento del lenguaje natural.

Índice General

Índice de figuras

Introducción

Con el avance de la tecnología, la comunicación ha sido testigo de horizontes más amplios. La distancia ya no es un factor para las dificultades en la comunicación, todo esto se atribuye a las redes sociales, como lo son los sistemas de información de noticias. Estas han proporcionado una plataforma para el intercambio de ideas, puntos de vista y sentimientos en todo el mundo con la ayuda de Internet y, por lo tanto, disminuyendo el tiempo de responder peticiones o en intercambios de ideas, entre otros. Millones de datos se generan y difunden a través de estas redes sociales que han atraído enormes intereses de la comunidad de investigación y la industria, y se puede recopilar mucha información útil a partir de estos enormes datos [1]. La Internet ha posibilitado un acceso más fácil y rápido a la información, la creación de espacios en línea para el intercambio de datos y a su vez crean plataformas para compartir ideas y opiniones.

Los sitios web de noticias son plataformas creadas con el fin de llevar los periódicos a un nivel digital y tener un mayor acceso a la mayor cantidad de personas. Por ejemplo, desde cualquier parte del mundo se pueden consultar las noticias publicadas en sistemas digitales como NewYork Times, BBC Mundo y en Cuba: Cubadebate, Granma, Trabajadores, Juventud Rebelde, entre otros. En dichos sitios, el usuario (lector) es el eje principal, puesto que él contribuye mediante comentarios o posts respecto a cualquier tema que se publique, observándose como influye la información sobre las personas. Estos posts o comentarios tienen características diversas ya que tienen formato de tipo texto, por lo que no pueden ser clasificados por categoría ni su búsqueda puede ser tan sencilla puesto que no son datos organizados y no están estructurados para ser guardados en una base de datos.

Sin embargo, en esta era de la tecnología moderna, hay un recurso que se tiene en abundancia: los datos estructurados y los no estructurados. Es por ello que, a partir de la segunda mitad del siglo XX, el aprendiza-

je automático emergió como una disciplina dentro de la inteligencia artificial que involucró el desarrollo de algoritmos de aprendizaje para obtener conocimiento de esos datos y hacer predicciones. En lugar de requerir que los humanos deriven reglas y construyan modelos manualmente para analizar grandes cantidades de datos, el aprendizaje automático ofrece una alternativa más eficiente para capturar el conocimiento en datos y así mejorar gradualmente el rendimiento de los modelos predictivos y tomar decisiones basadas en datos [3].

Los datos estructurados consisten en la información que suele mostrarse en una base de datos de manera organizada, puesto que son informaciones sencillas de buscar. Son archivos de texto que están clasificados, por lo que pueden ser ordenados y procesados fácilmente. Mientras que los datos no estructurados es el resto de la información que no es fácil de buscar como audios, vídeos, imágenes y publicaciones en las redes sociales. Ejemplos de datos estructurados son los almacenados como la base de datos de un hospital o la de Etecsa. Por otra parte ejemplo de datos no estructurados son los correos electrónicos, archivos de textos, archivos pdf, imágenes, vídeos, audios y publicaciones en medios sociales [2].

A su vez, existe una gran cantidad de datos no estructurados en línea, datos que hasta ahora no se podían utilizar. Esos datos, crecen todo el tiempo y se agregan a diario cuando, por ejemplo, se exponen los gustos y disgustos de los contenidos publicados en las redes sociales. La necesidad de analizar una gran cantidad de datos no estructurados crea una nueva ciencia llamada Minería de opinión, es un dominio de investigación que trata sobre los métodos automáticos de detección y extracción de opiniones presentes en el texto [4].

Tales datos pueden ser analizados usando técnicas de minería de datos, minería web y minería de textos en diversas aplicaciones de la vida real. La aplicación de estas técnicas sobre los datos provenientes de los sistemas de noticias puede revelar patrones sobre los individuos inmersos en el ambiente compartido y producir conocimiento que antes no era factible encontrar debido a la variedad y complejidad de la información [5]. Numerosos foros, periódicos digitales, redes sociales, sitios web, informes de noticias y recursos web adicionales sirven como plataformas para expresar opiniones, que pueden ser utilizados para comprender las opiniones en cualquier sistema de información de noticias. Estas reflejan diferentes sentimientos de satisfacción o no de lo que se discute a nivel nacional e internacional para el análisis de este tipo de información se

estudia el término análisis de sentimiento. Para llevar a cabo estas tareas, las comunidades de investigación y los académicos están trabajando rigurosamente en el término análisis de sentimiento [6].

El análisis de sentimiento es la tarea de detectar, extraer y clasificar opiniones, sentimientos y actitudes sobre diferentes temas, como expresados en datos textuales [7, 2]. Esta definición tiene como objetivo observar el estado de ánimo de un usuario en relación con cualquier tema y así extraer opiniones, identificar los sentimientos expresados y clasificar su polaridad, dígase positivo, negativo y neutro. Análisis de sentimiento y Minería de opinión son términos interrelacionados pero cumplen diferentes objetivos.

En los países en desarrollo, los medios en línea y sociales están tomando el lugar de los medios sin conexión rápidamente, lo que alienta a la gente común a participar en discusiones políticas y les permite presentar pensamientos unilaterales sobre temas globales de forma interactiva. Los medios en línea proporcionan la plataforma para compartir ampliamente ideas y público alentador para discusiones grupales con puntos de vista abiertos. Proporcionan mejores medios para obtener una respuesta rápida y comentarios sobre diferentes temas y entidades globales en forma de publicaciones textuales, noticias, imágenes y vídeos [7]. Por lo tanto, el análisis de sentimiento se puede utilizar para analizar las opiniones de las personas y así analizar la polaridad en los comentarios que realizan en los medios de información de noticias.

Por otra parte, en Cuba aún no se ha definido una infraestructura que de manera automática le permita a un usuario conocer en un sistema de noticias si un artículo es aceptado, neutro o no aceptado por la población. Actualmente, un desarrollador tiene que navegar a través de todo el conjunto de comentarios, leer manualmente cada uno de ellos y comprobar si contiene información relevante. Al no tener presente todas las opiniones de los usuarios, hace que dicho proceso no sea factible si la aplicación recibe cientos de comentarios por día, como sucede con los sistemas de información de noticias.

Entre los sistemas de información del país con mayor acceso se encuentra el sitio web de noticias Cubadebate, con una cultura de usuarios que de manera sistemática acceden y dan sus opiniones acerca de los contenidos que se publican en forma de post. Cubadebate incluye entre su directorio de prensa 15 sitios web

nacionales y 3 sitios internacionales, posee su página en Facebook, Twitter y Youtube y posee 8 canales RSS. Posee más 8000 suscriptores de usuarios activos en la aplicación móvil y supervisa más de 1000 comentarios diario. Por lo tanto, sirve como recurso para extraer opiniones publicadas por personas de diferentes lugares del país y con distinta cultura.

En este sentido, el análisis de dicha información resulta de gran utilidad para el grupo editorial de este portal. Siendo Cubadebate un recurso en el cual los usuarios sienten la libertad de opinar sobre cada contenido que se genera. En general, hay temáticas muy polémicas del ámbito nacional, donde el número de comentarios puede llegar a superar los 4000 en un artículo. Estos contenidos (Posts) tienen características muy particulares por lo general son textos cortos, descritos de manera informal o términos escritos con errores de tipiado. En los Post puede además el autor del comentario usar un enfoque positivo o a favor del contenido o un enfoque negativo con términos mal intencionados.

Teniendo en cuenta lo anteriormente expuesto, para el grupo editorial del sitio web de noticias Cubadebate es muy complejo clasificar la intencionalidad del autor cuando emite un comentario. Pues tomando en cuenta lo anterior dicho es importante saber las tendencias en las diferentes temáticas del país planteadas en Cubadebate como son económica, política, deportivas, salud, culturales entre otras para la toma de decisiones al conocer la opinión del pueblo y definir si el artículo es positivo, negativo o neutro. Es muy importante puesto que en esta era digital donde la comunicación y la tecnología es una de las principales vías de desarrollo de cualquier país

En correspondencia con lo antes expuesto se puede plantear como **problema de investigación**: ¿Cómo determinar de manera automática la intencionalidad de los usuarios a través de sus comentarios en el sitio web de noticias Cubadebate?, definiéndose así como **objeto de estudio** la minería de opinión en el sitio web de noticias Cubadebate, para dar solución al problema de investigación se traza como **objetivo general** : Clasificar automáticamente la polaridad para determinar la intencionalidad de los usuarios a través de sus comentarios en el sitio web de noticias Cubadebate, por lo que se plantea en el **campo de acción** método de clasificación supervisado para la minería de opinión en el sitio web de noticias Cubadebate.

Para darle cumplimiento al objetivo general se trazan los siguientes **objetivos específicos:**

- Caracterizar el marco teórico-conceptual de los algoritmos de procesamiento del lenguaje natural y minería de opinión para el análisis de sentimientos con énfasis en las herramientas de aprendizaje automático disponible para ello.
- Implementar algoritmos para descubrir conocimientos, mediante técnicas del procesamiento del lenguaje natural, que permita determinar automáticamente la polaridad en los comentarios que realizan usuarios en los sitios web de noticias nacionales.
- Validar los algoritmos escogidos mediante el diseño de experimentos comparando los resultados con algoritmos estudiados en el capítulo 1 y con su implementación en otras herramientas

Para el desarrollo de esta investigación se utiliza los siguientes **métodos teóricos**:

- **Analítico-Sintético:** se empleó para analizar las técnicas del procesamiento del lenguaje natural para escoger la más factible de implementar en la solución propuesta.
- **Medición:** se empleó para verificar la eficacia entre los algoritmo escogido de análisis de sentimiento con los demás algoritmos de clasificación.

Además de los siguientes **métodos empíricos**:

- **Revisión documental:** se empleó para analizar los documentos publicados referente a técnicas del procesamiento del lenguaje natural en sitios web de noticias nacionales similares a esta investigación.
- **Estadística:** se empleó para comparar de manera visual el algoritmo que se escogió de análisis de sentimiento con otros en la validación y también compara los algoritmos que se implementan entre cada dataset analizados del sitio web de noticias Cubadebate.

El presente documento de trabajo de diploma está compuesto por introducción, tres (3) capítulos, conclusiones generales, recomendaciones y referencias bibliográficas que fueron utilizadas en el desarrollo de la investigación.

Capítulo 1 : Se describe la metodología KDD para la extracción del conocimiento. Además, conceptos fundamentales como Minería de opinión y el análisis de sentimientos. Se aborda el preprocesamiento del lenguaje natural. También los métodos, modelos de representación vectorial y aprendizaje automático. Todo lo anteriormente investigado sobre la base de las herramientas y tecnologías a utilizar y aplicaciones de la minería de opinión encontradas.

Capítulo 2: Se implementan los pasos a seguir para desarrollar el proceso de solución propuesto, así como las librerías y algoritmos que se utilizan. La metodología adoptada en este trabajo está basada en las etapas del proceso de descubrimiento del conocimiento.

Capítulo 3: Se realiza la validación de la solución propuesta, mediante las métricas de precisión y recall para la precisión del algoritmo de clasificación a partir de las métricas de evaluación de rendimiento, que comprueban la eficiencia con que se ejecutan.

Capítulo 1

La minería de opinión para el descubrimiento de conocimiento

En el presente capítulo se describe el proceso de Descubrimiento de conocimiento, la minería de opinión en sistemas de información de noticias y el análisis de sentimiento. También se presentan los métodos de aprendizaje y los modelos de representación vectorial y aprendizaje automático. Por último, se describen las herramientas y tecnologías necesarias para desarrollar la solución.

1.1. Minería de opinión en sistemas de información de noticias

Antes de explicar la Minería de opinión en un sistema de noticias, lo primero es saber que es un sistema de información de noticias. Es un sistema, automático o manual, que comprende personas, máquinas y/o métodos organizados para agrupar, procesar, transmitir y diseminar datos que representan información para el usuario. Un sistema de información es cualquier equipo o sistema interconectado o subsistema de equipos de computación o telecomunicación que es usado en la adquisición, almacenamiento, manipulación, administración, movimiento, control, presentación, conmutación, intercambio, transmisión, o recepción de voz y/o datos, e incluye software, firmware, y hardware [8].

Por otra parte, la Minería de opinión (MO) es una extensión de la minería de datos, que utiliza técnicas de procesamiento de lenguaje natural para extraer la opinión de diferentes personas de las redes sociales. La

minería de opinión nos permite diseñar a través de técnicas del procesamiento del lenguaje natural algoritmos que permitan a una computadora de manera automática clasificar la intencionalidad del autor cuando emite un comentario. Esta analiza sistemáticamente cada texto y se da cuenta de qué parte contiene la palabra opinada, qué se está opinando y quién ha escrito la opinión [9]. La extracción de opiniones es un enfoque de recuperación de información de esta misma o sentimiento que se expresa en el texto. Usando la minería de opinión, se puede construir un sistema para recopilar y clasifique opiniones sobre un producto. La minería de opinión puede ser útil por ejemplo, analizar los comentarios en un sistema de noticias, ya que puede ayudar a saber si la población acepta, no acepta o se mantiene neutra con la noticia [10].

La extracción de opinión por tanto es un enfoque de recuperación de información para extraer la opinión o sentimiento que se expresa en el texto. La extracción de opiniones puede ser útil para analizar opiniones en las redes sociales, ya que puede ayudar a dar un criterio sobre publicaciones, a construir una buena imagen del tema raíz de la publicación entre comentaristas. Muchas personas piensan que la opinión y el análisis de sentimientos son lo mismo, pero en realidad son diferentes. Se puede decir que el análisis de sentimiento es parte de la minería de opinión [11]. Esta última extrae y analiza la opinión de las personas sobre una entidad, mientras el análisis de sentimiento identifica el sentimiento expresado en un texto y luego lo analiza [12]. La minería de opinión nos permite entonces diseñar a través de técnicas del procesamiento del lenguaje natural, algoritmos que permitan a una computadora de manera automática clasificar la intencionalidad del autor cuando emite un comentario [13].

Además, existen las siguientes desventajas: el primero, es que una palabra que se considera positiva en una situación puede considerarse negativa en otra situación y una segunda desventaja, es que las personas no siempre expresan opiniones de la misma manera [14].

1.2. Análisis de sentimiento

El análisis de sentimiento (AS) es el estudio computacional de las opiniones, actitudes y emociones de las personas hacia una entidad. La entidad puede representar individuos, eventos o temas. Estos temas son más

probablemente cubiertos por opiniones. El análisis de sentimiento por otra parte, se considera el estudio del pensamiento y sentimiento del usuario hacia un producto [7].

Una manera de clasificar comentarios que se extraen mediante la minería de opinión en afirmaciones positivas, negativas y neutrales es que sean definida por 5 tuplas según [15].

($E_j, A_{jk}, SO_{ijkl}, H_i, T_l$):

- E_j es el objetivo de la entidad
- A_{jk} es el aspecto/característica de la entidad E_j
- SO_{ijkl} es el valor del sentimiento de la opinión desde el poseedor de la opinión
- H_i es el poseedor de la opinión
- T_l el tiempo donde la opinión fue expresada.

El análisis de sentimiento se puede considerar un proceso de clasificación, primero ocurre una revisión de productos, luego se identifica el sentimiento mediante palabras o frases, se seleccionan las características de estas opiniones y se clasifican para determinar su polaridad [15].

En el proceso de clasificación existen 3 niveles principales en el AS: nivel de documento, nivel de oración y nivel de aspecto [7], estos niveles se utilizan en la tercera etapa de KDD, Minería de datos, cuando se estudia el documento, luego se profundiza a nivel de oración y se analiza a nivel de palabra.

- Nivel de documento, tiene como objetivo clasificar un documento (comentario) donde expresa una opinión con sentimiento positivo o negativo. Eso considera todo el documento una unidad de información básica (hablando de un tema).
- Nivel de oración, apunta a clasificar sentimientos expresados en cada oración. El primer paso es identificar si la oración es subjetiva u objetiva. Si la oración es subjetiva, el AS determinará a nivel de la oración

si la oración expresa opiniones positivas o negativas. No hay diferencia fundamental entre documento y nivel de oración porque las oraciones son solo documentos cortos. La clasificación de texto en el nivel del documento o en el nivel de la oración no proporciona el detalle necesario de las opiniones necesarias sobre todos aspectos de la entidad que se necesita en muchas aplicaciones, para obtener estos detalles; por lo que se necesita ir al nivel de aspecto.

- Nivel de aspecto, tiene como objetivo clasificar el sentimiento con respecto a los aspectos específicos de las entidades. El primer paso es identificar las entidades y sus aspectos. Los poseedores de opinión pueden dar diferentes opiniones para diferentes aspectos de la misma entidad como esta frase "La calidad de voz de este teléfono no es buena, pero la vida de la batería es larga".

En la investigación, se utilizan los 3 niveles, el primer nivel, cuando se seleccionan la fuente de datos y se analiza el texto extraído. El segundo nivel, en la etapa de preprocesamiento de datos cuando se analiza oración por oración al igual que el 3er nivel en el preprocesamiento de datos cuando se divide cada oración en una lista de palabras y se analiza cada una de ellas. Al igual que se puede analizar de manera inversa cuando de una lista de palabras ya desglosadas en el preprocesamiento de datos analizas en el contexto a nivel de aspecto.

Los Dataset son fuentes de datos utilizadas para guardar toda la información referida a los post o comentarios. Los sitios de redes sociales y microblogging son considerados una fuente de información porque las personas comparten y discuten sus opiniones sobre un tema determinado libremente. También se utilizan como fuentes de datos en el proceso del AS [16]. Habiendo definido los principales niveles en el proceso en el AS se pueden definir como modelos de entidad en el primer nivel de los modelos de documento de opinión como [2]:

- **Modelo de entidad**: Una entidad ei está representada por ella misma, también como una serie finita de aspectos $A_i = a_{i1}, a_{i2,\ldots,ain}$. Puede ser ei expresado con cualquier serie finita de estas expresiones de entidad $e_{ei1}, e_{ei2}, \ldots, e_{eis}$. Estos aspectos $a_{ij} \in A_i$ de esta entidad ei pueden ser expresados con cualquiera de estas series finitas de expresiones $a_{eij1}, a_{eij2}, \ldots, a_{eijm}$.

- **Modelo de opinión de documento**: Sea la colección de documentos D cuenta con una serie de entida-

des $e_1, e_2, \ldots, e_r$ y de una serie desde un conjunto de opiniones huéspedes $h_1, h_2, \ldots, h_p$ y una forma particular de puntos.

Finalmente, dado un conjunto de documentos de opinión el análisis consta de las siguientes 6 tareas principales:

1. Extracción de entidades y categorización: extraer todas las expresiones de entidades en D, y categorice o agrupe expresiones de entidad sinónimas en agrupaciones (o categorías). Cada agrupación de expresiones de entidad indica una única entidad.

2. Extracción de aspecto y categorización: extraer todas las expresiones de aspecto de las entidades, y categorice estas expresiones de aspecto en grupos. Cada grupo de expresiones de aspecto de la entidad e_i representa un aspecto único a_{ij}.

3. Extracción del titular de opinión y categorización: extraer titulares de opiniones a partir de texto o datos estructurados y categorizarlos. La tarea es análoga a las dos tareas anteriores.

4. Extracción y estandarización del tiempo: extraer los tiempos en que se dan opiniones y se estandarizan diferentes formatos de hora. La tarea es también análoga a las tareas anteriores.

5. Clasificación de sentimiento de aspecto: determinar si una opinión sobre un aspecto a_{ij} es positivo, negativo o neutral, o asigna un número valoración del sentimiento al aspecto.

6. Generación de quintuples de opinión: producir todos los quintuples de opinión $(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$ expresado en el documento D basado en los resultados de las anteriores tareas. Esta tarea es aparentemente muy simple, pero de hecho es muy difícil en muchos casos.

En [2] se presenta diferentes tareas posibles y trabajos publicados sobre AS y minería de opinión. Las principales tareas enumeradas son la subjetividad y la clasificación de sentimientos, el AS basado en aspectos, la generación de léxicos de sentimientos, el resumen de opiniones, el análisis de opiniones comparativos, la búsqueda y recuperación de opiniones, la detección de spam de opinión y la calidad de las revisiones. Según [15] expone que el campo del AS se centra en cinco problemas específicos: el AS a nivel de documento, el

AS a nivel de oración, el AS basado en aspectos, el AS comparativa y adquisición de léxico de sentimientos. También enumeraron algunos problemas abiertos, como la declaración de composición del AS, el reconocimiento automático de entidades, la discusión sobre múltiples entidades en la misma revisión, la detección de sarcasmo y la clasificación de subjetividad a un nivel más fino.

Aunque en [7], se presenta seis tareas como resumen de Bing Liu de las etapas que presentó en AS. Estas son: análisis de sentimiento, detección de emociones, construcción de recursos, transferencia de aprendizaje, clasificación de sentimientos y selección de características. Las que se utilizan en la investigación son: análisis de sentimiento, clasificación de sentimientos obteniendo a su vez la clasificación de la polaridad, construcción de recursos, esta tarea se implementa en el modelo de representación vectorial y transferencia de aprendizaje se observa como Recuperación de información dentro del preprocesamiento de datos . Según Medhat expone que el análisis de sentimiento como tarea es un problema de clasificación. Estas son las etapas que se utilizarán siendo las más actuales.

Por tanto, mediante el análisis de sentimiento se puede conocer y analizar las opiniones de un público cubano, tanto los que viven dentro como fuera del país con respecto a temas tanto nacionales como de índole internacional. Es una herramienta más para descubrir para descubrir conocimiento con el objetivo de conocer si se acepta, disgusta o no revela nada sobre procesos, sucesos mediante un artículo.

1.2.1. Clasificación de sentimientos

La clasificación de sentimiento es la determinación de la orientación del sentimiento de un texto dado en dos o más clases . Este puede ser realizado utilizando el aprendizaje automático, así como los enfoques basados en el léxico. La máquina de aprendizaje produce la máxima precisión entre sus algoritmos, mientras que la orientación semántica proporciona una mejor generalidad [7].

La máquina de aprendizaje se puede dividir en enfoques supervisados y no supervisados, este último se utiliza cuando es difícil encontrar estos documentos etiquetados, que a investigar este no es el caso, por lo que se utiliza enfoque supervisado [17]. Se llama así debido a la presencia de la variable de resultado para guiar el

proceso de aprendizaje. Este enfoque hace uso de un gran número de documentos de formación etiquetados. Para aproximaciones supervisadas, se necesita dos conjuntos de datos anotados, uno para cada entrenamiento y prueba [6] como se muestra en la siguiente Figura 1.1.

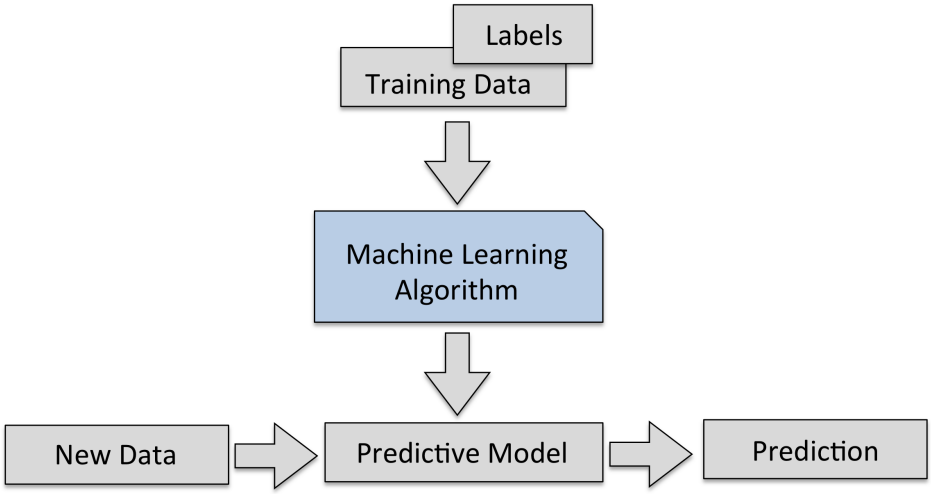

Figura 1.1: EL algoritmo máquina de aprendizaje [Tomado de [3]]

El **objetivo principal** del aprendizaje supervisado es aprender un modelo a partir de datos que estén clasificados manualmente y que nos permitan hacer predicciones sobre datos invisibles o futuros. El término supervisado se refiere a un conjunto de muestras donde las señales de salida deseadas (etiquetas) ya son conocidas [6].

El modelo predictivo aprendido por un algoritmo de aprendizaje supervisado puede asignar cualquier etiqueta de clase que se presentó en el conjunto de datos de entrenamiento a una nueva instancia sin etiqueta. Un ejemplo típico de una tarea de clasificación de clases múltiples es el reconocimiento de caracteres a mano. Aquí, se podría recopilar un conjunto de datos de entrenamiento que consta de varios ejemplos escritos a mano de cada letra en el alfabeto. Ahora, si un usuario proporciona un nuevo carácter escrito a mano a través de un dispositivo de entrada, el modelo predictivo podrá predecir la letra correcta en el alfabeto con cierta precisión [3].

Los clasificadores para el aprendizaje supervisado son: Árbol de decisión, Máquina de soporte vectorial, Red Neural, Naïve Bayes y Máxima Entropía. Por otra parte, en el aprendizaje basado en el léxico, se divide en el enfoque

basado en el diccionario que este utilizará un diccionario existente, que es una colección de palabras de opinión junto con su fuerza de sentimiento positiva (+ve) o negativa (-ve), a su vez. Además, el enfoque basado en corpus se basa en la probabilidad de ocurrencia de una palabra de sentimiento junto con un conjunto de palabras positivo o negativo al realizar una búsqueda en una gran cantidad de textos [6].

La clasificación de sentimientos se ocupa de determinar la polaridad de una oración: positiva, negativa o neutral. Por lo tanto, la clasificación del sentimiento es también denominado como determinación de polaridad. La determinación de la polaridad se ha realizado para revisiones de productos, foros, blogs, artículos de noticias y micro-blogs. Por lo tanto, necesita un preprocesamiento de alto nivel, así como técnicas de análisis más inteligentes [6].

Técnicas de clasificación de sentimientos

Las técnicas de clasificación de sentimientos se pueden dividir aproximadamente en el enfoque de aprendizaje automático, enfoque basado en el léxico y enfoque híbrido. El enfoque basado en el léxico se basa en un léxico del sentimiento, una colección de términos de sentimiento conocidos y precompilados. Es dividido en enfoque basado en diccionario, este depende de encontrar palabras semilla de opinión, y luego busca en el diccionario de sus sinónimos y antónimos. Y el enfoque basado en corpus, utiliza métodos estadísticos o semánticos para encontrar la polaridad del sentimiento, este comienza con una lista inicial de palabras de opinión, y luego busca otras palabras de opinión en un corpus grande para ayudar con orientaciones específicas del contexto [7], este enfoque posee dos técnicas: Estadística y Semántica. Siendo este enfoque no factible puesto que tendrías que tener otra fuente de datos (corpus) construida. Sería más trabajoso el proceso y más largo.

El enfoque de aprendizaje automático (ML) aplica los algoritmos de máquina de aprendizaje y usa características lingüísticas, siendo este el enfoque a utilizar, como se muestra en la Figura 1.2. Este se divide en no supervisado y supervisado. Este último se puede clasificar de cuatro maneras: clasificación de árbol de decisiones, clasificación lineal, clasificación probabilística, clasificación basada en reglas. Estas clasificaciones tienen sus técnicas: la clasificación lineal posee dos: Red Neural y Máquina de soporte vectorial (SVM) y la clasificación probabilística posee tres técnicas: Naïve Bayes, Máxima Entropía y Red Bayesiana.

La técnica escogida fue Máquina de soporte vectorial (SVM), de la clasificación lineal. La teoría de la SVM está basada

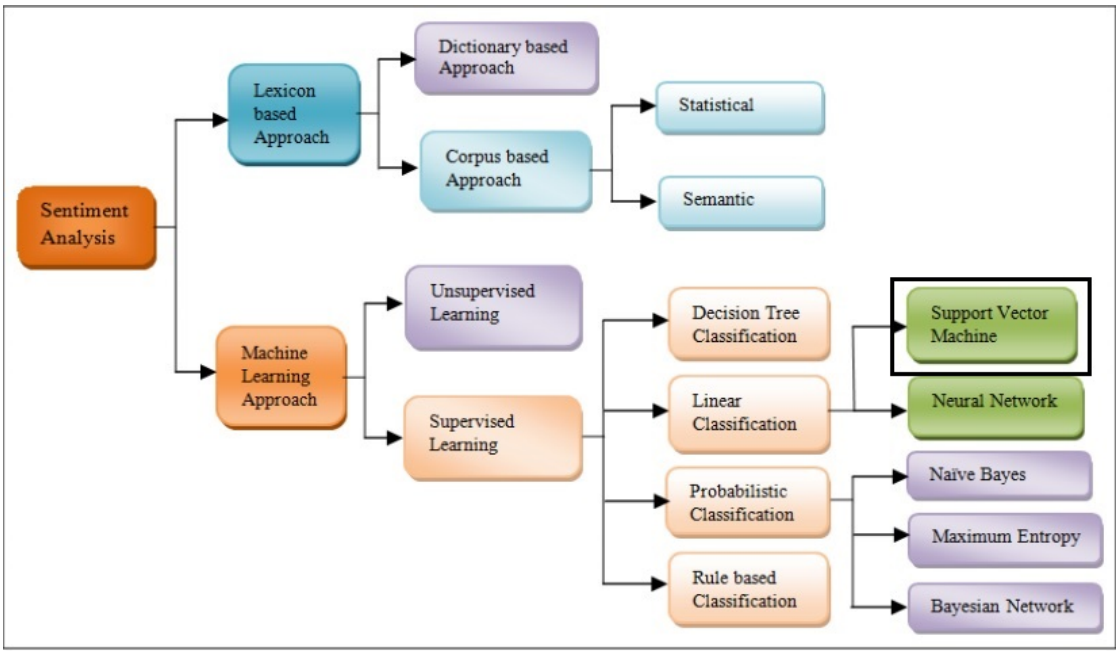

Figura 1.2: Técnicas de análisis de sentimiento [Tomado de [15]]

en la idea de minimización de riego estructural. En muchas aplicaciones, las SVM han mostrado tener gran desempeño, más que las máquinas de aprendizaje tradicional como las redes neuronales y han sido introducidas como herramientas poderosas para resolver problemas de clasificación [18]. Además, las SVM son básicamente clasificadores para 2 clases, aunque se puede cambiar la formulación del algoritmo para permitir clasificación multiclase.

La SVM ha demostrado ser altamente efectiva en la categorización de textos tradicionales, en general, superando a Naïve Bayes. Son clasificadores de gran margen, en lugar de probabilísticos, en contraste con Naïve Bayes y Máxima Entropía, logran una precisión de búsqueda significativamente mayor que otro algoritmo según [19, 18, 20, 21, 22, 23, 24].

Los clasificadores de máquina de soporte vectorial tienen como principio fundamental determinar los separadores lineales en el espacio de búsqueda que mejor puede separar las diferentes clases. Hay 2 clases x, o y hay 3 hiperplanos A, B y C. El hiperplano A proporciona la mejor separación entre las clases, debido a la distancia normal de cualquiera de los puntos que son los datos, por lo que representa el margen máximo de separación. El objetivo de la SVM es producir un modelo que permita predecir los valores de clasificación (identificar la clase) en la etapa de prueba conociendo solo los atributos [25].

La clasificación SVM es utilizada en categorización de texto por la escasa naturaleza del texto, en el que pocas características son irrelevantes, pero tienden a estar correlacionadas entre sí y generalmente organizados en categorías linealmente separables. La máquina de soporte vectorial puede construir una superficie de decisión no lineal en el espacio original de características mediante la asignación de las instancias de datos, a un espacio interno del producto donde las clases pueden ser separadas linealmente con un hiperplano. Las SVM se utilizan en muchas aplicaciones, entre las que se encuentran las revisiones de clasificación según su calidad. Chen y Tseng han usado dos multiclase basadas en los enfoques de SVM: SVM uno-contra-todos y SVM multiclase de una sola máquina para clasificar las revisiones [26].

Se puede plantear este problema como una tarea de clasificación binaria en la que se refiere a dos clases como 1 (clase positiva) y -1 (clase negativa) para simplificar. Luego se definirá una función de activación $\phi(Z)$ que toma una combinación lineal de ciertos valores de entrada x y un vector de peso correspondiente w, donde z es la llamada entrada de red $(z = w_1 * x_1 + ... + w_m * x_m)$

$$W = \begin{pmatrix} W_1 \\ ... \\ W_m \end{pmatrix}$$

La función objetivo de SVM trata de maximizar el margen, $\frac{2}{||w||}$ bajo la restricción de que las muestras se clasifican correctamente, que se puede escribir como se muestra en la Figura 1.3:

$$w_0 + w^t x \geq 1 \; si \; y = 1$$

$$w_0 + w^t x < 1 \; si \; y = -1$$

Figura 1.3: Fórmula de SVM

1.3. Descubrimiento de conocimiento

El Descubrimiento de conocimiento (KDD, por sus siglas en inglés), es un conjunto de procesos introducidos al análisis de las bases de datos [27]. Fayadd en [28] lo define como:

> ‘’procesos no comunes de identificación de patrones válidos, novedoso, potencialmente útiles y finalmente comprensibles en los datos”

Su propia definición se debe a que comprende los procesos siguientes: Selección de datos, Preprocesamiento de datos, Transformación de datos, Minería de datos, Interpretación y Evaluación del conocimiento descubierto. Por lo que vale puntualizar que se utilizará KDD como metodología de trabajo.

Aunque hay varias definiciones del proceso KDD, se puede observar que todos los autores coinciden en que este es un proceso con el cual es posible obtener información útil de grandes almacenes de datos. A través de los años han variado las etapas de KDD aunque manteniendo su objetivo y los siguientes procesos [28]:

1. **Selección de datos**: En esta primera etapa del proceso se realiza una selección de las fuentes de datos, estas pueden ser bases de datos y/o archivos. Además, se eliminan los datos inconsistentes y se combinan las diferentes fuentes de datos que fueron seleccionadas en un Data Warehouse.

   Se debe integrar todos los datos a analizar. Existen varias formas de combinar las distintas bases de datos para crear el repositorio. Una posibilidad es simplemente hacer una copia de las bases de datos integrantes (eliminando inconsistencias y redundancias). Lo importante en el proceso de extracción de datos es llevar todas las fuentes de datos a un formato común y consistente para proceder con la carga del dataset correspondiente.

   La limpieza de datos es un proceso que ha sido identificado como el elemento más laborioso en la construcción del dataset. Para los datos de entrada, la limpieza debe hacerse antes de cargar los mismos. No hay ningún aspecto para la limpieza de datos que sea específico de un almacenamiento de datos y que no pueda aplicarse a una base de datos anfitriona. Sin embargo, dado que los datos de entrada deben ser examinados y formateados consistentemente, se aprovecha esta oportunidad para comprobar la validez y calidad. El reconocimiento de datos erróneos e incompletos es difícil de automatizar, y la limpieza que requiere una corrección de errores automática puede resultar incluso más ardua. Cada vez que una operación manual es realizada, se debe tratar de identificar el camino tomado para luego eliminar el proceso manual y realizar procesos automáticos que lo reemplacen. En algunos casos, se debe ser capaz de modificar los sistemas contenedores de los datos originales y así no tener que realizar el proceso cada vez que se necesiten estos datos.

Además del problema de irrelevancia (datos no relevantes) existen otros problemas que afectan la calidad de los datos. Uno de estos problemas es la presencia de valores que no se ajustan al comportamiento general de los datos. Estos datos anómalos pueden representar errores o pueden ser valores. Una forma de detectar estos atributos cuando son nominales, es ver si existen tantos valores del atributo como instancias.

La presencia de datos faltantes o perdidos (missing values) puede ser también un problema pernicioso que puede conducir a resultados poco precisos. No obstante, es necesario reflexionar primero sobre el significado de los valores faltantes antes de tomar alguna decisión sobre cómo tratarlos, ya que estos pueden deberse a causas muy diversas, como un mal funcionamiento del dispositivo que hizo la lectura del valor, o cambios efectuados en los procedimientos usados durante la recolección de los datos o al hecho de que los datos se recopilen desde fuentes diversas. Estos tres problemas son sólo tres ejemplos que muestran la necesidad de la limpieza de datos, es decir, de mejorar su calidad.

2. **Preprocesamiento y transformación de los datos**: En este paso del proceso se seleccionan los atributos que serán utilizados y son transformados en un formato apropiado para el análisis que será realizado posteriormente con la minería de datos. Las dos primeras etapas del proceso KDD son las etapas en las que se consume más tiempo dado que es aquí donde se debe tener especial cuidado en la "limpieza" que exista en los datos, ya que sin calidad en ellos no habrá calidad en los resultados obtenidos a través de la minería de datos.

   Este proceso incluye, entre sus tareas, la construcción de nuevos atributos. Automáticamente, aplicando alguna operación o función a los atributos originales. Una fuerte motivación para esta tarea surge cuando los atributos originales no tienen mucho poder predictivo por sí solos o los patrones dependen de variaciones lineales de las variables originales.

   El tipo de datos también puede modificarse para facilitar el uso de técnicas que requieren tipos de datos específicos. Así, algunos atributos podrán ser "numerizados" para reducir el espacio y poder usar técnicas numéricas. Por ejemplo, se pueden reemplazar los valores del atributo "sexo" por números enteros. El proceso inverso consiste en discretizar los atributos continuos, es decir, transformar valores numéricos en atributos discretos o nominales. Los atributos discretizados pueden tratarse como atributos categóricos con un número más pequeño de valores.

La idea básica es partir de los valores de un atributo continuo en una pequeña lista de intervalos, tal que, cada intervalo es visto como un valor discreto del atributo.

Otras transformaciones típicas incluyen las siguientes tareas: unificación de nombres de campos, división de las fechas en campos separados (por ejemplo, año, mes y día), mapeos de una representación a otra (por ejemplo, los valores booleanos "true" y "false" a "1" y "0" respectivamente); también códigos numéricos a campos de textos, mapeos de múltiples representaciones a una única (por ejemplo, unificar los formatos de los números de teléfonos a una representación común).

3. **Minería de datos**: Es la parte medular del proceso KDD puesto que se perfeccionan las técnicas y algoritmos que se encargan de extraer y representar el conocimiento de forma adecuada para la toma de decisiones. Se combinan técnicas potenciando las ventajas de cada una y atenuando sus debilidades. La minería de datos tiene como propósito la identificación de un conocimiento obtenido a partir de las bases de datos que aporten hacia la toma de decisión.

   Una vez recogidos los datos de interés, se puede decidir qué tipo de patrón quiere descubrir. El tipo de conocimiento que se desea extraer va a marcar claramente la técnica de minería de datos a utilizar. Según como sea la búsqueda del conocimiento se puede distinguir entre:

   - **Minería de dato directa**: se sabe claramente lo que se busca, generalmente predecir unos ciertos datos o clases.
   - **Minería de datos indirecta**: no se sabe lo que se busca, se trabaja con los datos.

   En el primer caso, algunos sistemas de minería de datos se encargan generalmente de elegir el algoritmo más idóneo entre los disponibles para ir determinando el patrón a buscar. En la presente investigación se trabajará con minería de datos directa, pues ya se sabe lo que se busca: definir si el artículo analizado es positivo, negativo o neutral dependiendo a los comentarios que se entran como parámetros y se definen las clases de un principio: positiva, negativa o neutra. Entre la tipología de minería de datos, se cuentan con técnicas que crean modelos predictivos y/o descriptivos.

4. **Interpretación y evaluación del conocimiento descubierto**: Se procede al análisis de los resultados descubiertos. Incluye a su vez la resolución de posibles inconsistencias con otros conocimientos anteriores a la investigación.

1.4. Preprocesamiento de datos en lenguaje natural

El análisis de sentimiento, la minería de opinión y la clasificación de sentimiento son áreas interrelacionadas de investigación que utilizan diversas técnicas tomadas del procesamiento del lenguaje natural (NLP, por sus siglas en inglés), recuperación de información, minería de datos estructurado y no estructurados. Este último se refiere a los datos disponibles en todo el mundo (como texto, habla, audio, vídeo)

Varios pasos se requieren para realizar la extracción de opinión a partir de textos dados, ya que los textos para la extracción de opinión están llegando de varios recursos en diverso formato. La adquisición de datos y el preprocesamiento de datos son las subtareas más comunes requeridas para la minería de textos y el AS.

Para determinar si el algoritmo de aprendizaje automático no solo funciona en el conjunto de entrenamiento, también se divide aleatoriamente el conjunto de datos de entrenamiento y prueba separado. Se utiliza el conjunto de entrenamiento para entrenar y optimizar el modelo de aprendizaje automático, mientras el conjunto de prueba hasta el final para evaluar el modelo final.

Los datos del mundo actual a menudo son incompletos, inconsistentes e inciertos y la mayoría contienen muchos errores. El preprocesamiento de datos es un método que previene resolver estos problemas. Estos tendrán ayuda para obtener mejores resultados de algoritmos clasificadores como: tokenización, eliminación de palabras de parada, derivación, partes de Etiquetado de voz (POS), y extracción y representación de características.

- La tokenización se usa para romper una oración en palabras, frases, símbolos u otros símbolos significativos al eliminar los signos de puntuación. La tokenización es una técnica fundamental para la mayoría de las tareas de NLP, se divide una frase o documento en fichas que son palabras o frases.
- La eliminación de palabras de parada, consiste en quitar los signos, los puntos; todo lo que no sean palabras.

- Stemmer o derivación es el proceso para traer una palabra a su forma de raíz, mientras ignora otros POS de la palabra.
- El etiquetado POS se realiza para reconocer diferentes partes del discurso en el texto, que es bastante esencial para el procesamiento del lenguaje natural.
- La extracción y representación de características se utiliza para crear las clases en que vas a colocar cada palabra que posean significados parecidos o iguales. La selección de características tiene como objetivo: Impedir que la máquina de aprendizaje se sobreentrene, producto del entrenamiento con datos que causen un mal ajuste de la función objetivo y mejorar la eficiencia computacional, ya que mucho de los algoritmos no pueden manejar de una manera óptima un espacio vectorial de alta dimensión. A su vez, la selección de características permite dos enfoques, la selección de características y construcción de características, ésta última consiste en introducir nuevas características que tienen como meta representar la mayor información posible de la representación original mientras se minimiza el número de atributos. Este último paso se puede o no realizar, no es necesario pues puede realizarse como un paso adicional.

1.5. Modelos de representación vectorial

La selección de un buen modelo de representación de documentos es un aspecto clave en la categorización de tareas de textos. El enfoque habitual, denominado modelo Bolsa de palabras (BoW), considera que las palabras son índices simples en un vocabulario de término. En este modelo la información del campo de recuperación y los documentos se representan como vectores indexados por esas palabras. Una colección de documentos compuesta por n documentos indexados y m términos representados por una matriz documento-término de $[N * M]$.

Los N vectores renglón representan los n documentos; y el valor asignado a cada componente refleja la importancia o frecuencia ponderada que produce el término, frase o concepto T_i en la representación semántica del documento d_j. Cada componente de un vector (documento) representa el peso que la palabra correspondiente ha asociado en ese documento. Lo que representa m es la cardinalidad del diccionario y representa la contribución del término T_i para la representación semántica del documento d_j [29].

$$D_j = (W_{1j}, W_{2j}, W_{3j}, \ldots W_{mj})$$

Bolsa de palabras brinda un método para extraer características de documentos de texto. Estas características se pueden utilizar para entrenar algoritmos de aprendizaje automático. Crea un vocabulario de todas las palabras únicas que aparecen en todos los documentos del conjunto de capacitación. En términos simples, es una colección de palabras para representar una oración con recuento de palabras y, en general, sin tener en cuenta el orden en que aparecen [20]. BoW es un enfoque ampliamente utilizado con:

- Procesamiento natural del lenguaje
- Recuperación de información de documentos
- Clasificaciones de documentos

Entre las limitaciones de las representaciones de BoW, se encuentra la dispersión de los vectores resultantes y la pérdida de cualquier información sobre ubicaciones de palabras dentro de documentos. BoW es una de las más utilizadas en las tareas de categorización de texto según [30, 20, 23, 31].
Esta representación popular es simple de implementar, rápida de obtener y se puede utilizar bajo diferentes esquemas de ponderación. Sin embargo, los ordenamientos de las palabras en el documento se ignoran y la información semántica y conceptual están perdidas.

Representaciones de palabras basadas en vectores

Existen diversos métodos en las representaciones de palabras basadas en vectores, considerando cuatro métodos del área de distribución (SOA[1] , LSA[2] , LDA[3] , DOR[4]) y un representante del enfoque de representación distribuida (Word2Vec). En la investigación se utilizará LDA, puesto que trata de una manera de descubrir automáticamente categorías o temas, a partir de una colección de documentos y ordenarlos en lista en función de la similitud que presenten las palabras. En las representaciones de palabras basadas en vectores (VWRs), cada palabra tiene asociado un vector. El valor de cada dimensión corresponde a una entidad, denominada Word feature, que podría tener una interpretación semántica o gramatical. VWRs se supone que supera limitaciones del modelo BoW al permitir capturar una estructura relacional más rica del léxico. Esto se logra mediante la codificación de similitudes continuas (no binarias) entre palabras como distancia o ángulo entre vectores de palabras en una alta dimensión espacio [29].

- Asignación Latente de Dirichlet (LDA)

[1] Atributos de segundo orden
[2] Análisis semántico latente
[3] Asignación Latente de Dirichlet
[4] Representación del suceso del documento

LDA es un modelo generativo probabilístico para colecciones de datos discretos. Es un modelo iterativo que comienza a partir de un número fijo de temas. Cada tema se representa como una distribución sobre palabras, y cada documento se representa como una distribución sobre temas. Aunque los tokens en sí mismos no tienen significado, las distribuciones de probabilidad sobre las palabras proporcionadas por los temas ofrecen un sentido de las diferentes ideas contenidas en los documentos. El modelado de temas es una técnica para identificar los grupos de palabras (llamado tema) de una colección de documentos que contiene la mejor información de la colección. Se utilizó la asignación de Dirichlet latente para generar funciones de modelado de temas [29, 32].

Este es un modelo bayesiano jerárquico de tres niveles, en el que se modela cada elemento de una colección como una mezcla finita sobre un conjunto subyacente de K temas. Cada tema es, a su vez, modelado como una mezcla infinita sobre un conjunto subyacente de probabilidades de temas. Básicamente, LDA observa cada documento y asigna aleatoriamente cada palabra del documento a una de los temas k.

Mejorar la distribución de temas LDA para cada palabra en cada documento, asume que esta palabra está en un tema incorrecto y trata de encajar en otros temas maximizando la probabilidad de que la palabra sea con otras palabras con el mismo contexto. Este proceso se repite varias veces hasta que la distribución tópica no cambia sustancialmente. Se trata de una manera de descubrir automáticamente categorías o temas, a partir de una colección de documentos.

Entre las herramientas para el procesamiento del lenguaje natural se utilizará Word2Vec,puesto que obtiene vectores de palabras con características significativas, es decir, las palabras relacionadas están en el mismo grupo si aparecen en contextos similares, tienen similares significado y / o tener relaciones semánticas [29]. Es un método de aprendizaje de representación utilizado para obtener representaciones de palabras distribuidos (también nombradas incrustaciones de palabras). El método aprende un modelo usando pocos documentos etiquetados [32].

1.6. Modelo de clasificación y aprendizaje automático

Ya escogido el método de aprendizaje automático supervisado, proporcionan algoritmos de agrupamiento y algoritmos de asociación mostrados en la Figura 1.4. Anteriormente se escogió para la clasificación de sentimiento la técnica SVM, entonces se elegirá como modelo de aprendizaje automático SVM, para seguir la misma linea que se comenzó, o

sea el mismo algoritmo se enseña para que lo haga automáticamente. Un enfoque común para la evaluación de la calidad de los resultados de agrupamiento es usar índices de validación de grupos, los cuales tienen por objetivo encontrar un conjunto de grupos que se ajuste a una partición natural de los datos, usualmente definidos combinando compacidad y separabilidad.

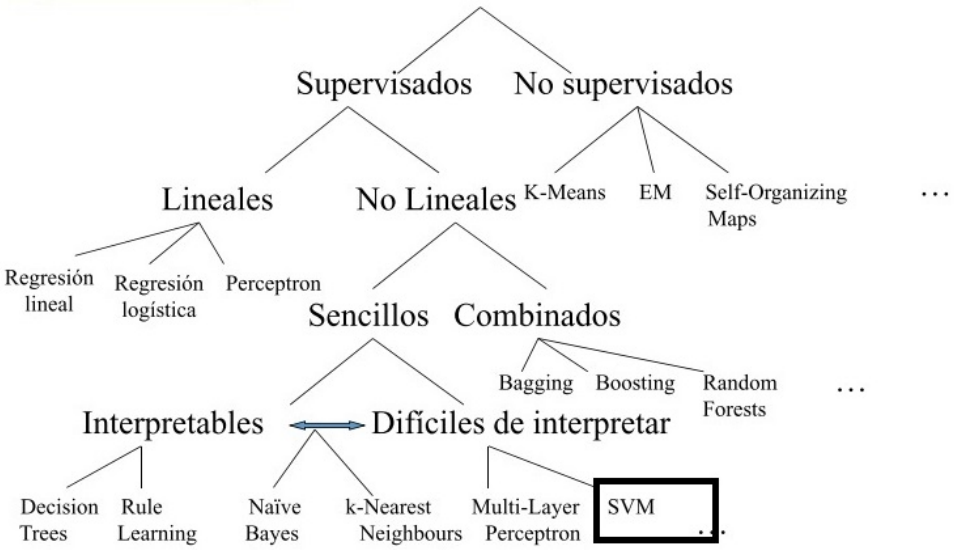

Figura 1.4: Métodos de aprendizaje inductivo [Tomado de [22]]

1.7. Análisis de herramientas existentes

En la actualidad la MO se usa en diferentes negocios y ramas, principalmente en sistemas de información de noticias como lo son en blogs, microblogs, revistas digitales entre otros. Hoy en día, los periódicos digitales se ha convertido en una comunicación muy popular entre los usuarios de Internet. Millones de usuarios compartiendo opiniones de millones de publicaciones que se realizan al día. Por lo tanto, los sitios web de microblogging y sistemas de información de noticias son fuentes ricas de datos para la extracción de opiniones y el análisis de sentimientos.

Uno de los ejemplos de soluciones en el análisis de sentimientos es **SentiWordNet**, que siendo un recurso léxico producido para preguntar a un clasificador automatizado. Asocia a cada uno de los sincronismos WordNet (versión 2.0), un triplete de puntuaciones numéricas (Positivo, Negativo, Objetivo) [33]. Para dar sentido a la enorme cantidad de datos

sociales disponibles en la Web, se requiere la adopción de enfoques novedosos para la comprensión del lenguaje natural que puedan dar una estructura a dichos datos, de manera que puedan ser agregados y analizados más fácilmente. En este contexto, la computación sentic se puede explotar para tareas de NLP que requieren la inferencia de información semántica y / o afectiva asociada con el texto, desde el análisis de grandes datos sociales hasta la gestión de datos, metadatos de la comunidad en línea y el análisis de las dinámicas de interacción de redes sociales [34].

Una de las principales redes sociales que utiliza análisis de sentimiento es **Twitter** en [35]. Explica como realiza el proceso de conversión de la opinión en estadísticas usando el corpus, construyen un clasificador de sentimiento, que se puede establecer como positivo, negativo y neutralizado para un documento mediante la técnica de clasificación de sentimiento SVM y la herramienta TreeTagger for POS-tagging.

Por otra parte, **Brandwatch** es una herramienta de monitoreo de redes sociales que permite obtener una idea de la opinión pública general sobre ciertos temas. Utiliza un proceso de reglas para ayudar a entender mejor las diferentes maneras en que el contexto puede afectar al sentimiento. Es una herramienta abarcadora, el precio más básico es de 800 dólares y en la medición de 'sentimiento' no es posible conocer en más detalle al usuario que emitió el comentario negativo o positivo.

Las mismas capacidades que tendría un motor de búsqueda orientado a la revisión también podrían servir como la base para la creación y el mantenimiento automatizado de sitios web de revisión y agregación de opinión, es decir, como una alternativa a sitios como **Epinions** que solicitan comentarios y opiniones, uno podría imaginar sitios que proactivamente recopilen dicha información. Los temas no tienen por qué limitarse a las revisiones de productos, sino que podrían incluir opiniones en sistemas de información en noticias. Uno de los problemas que se presentan es resumir las opiniones de los usuarios, también los errores que comete el usuario en su escritura. Las calificaciones podrían ser fijas: hay casos en que los usuarios han seleccionado claramente una calificación baja accidentalmente cuando sus revisiones indica una evaluación positiva. Además, existe alguna evidencia de que las calificaciones de los usuarios pueden estar sesgadas o que necesitan corrección, y clasificadores automatizados podrían proporcionar tales actualizaciones [2].

Clipping 2.0 ofrece a sus clientes (Administraciones públicas, Marcas, grandes empresas, agencia de noticias y particulares) una imagen clara de cómo los medios de comunicación y las redes sociales hablan de su marca, para ayudarles

a comprender las conversaciones y los sentimientos mediante la medición de la imagen corporativa, de relaciones públicas y de marca.

PosNeg Opinion es una herramienta desarrollada por el Departamento de Ciencia de la Computación en la Universidad Central de las Villas, para que el usuario analice un gran cúmulo de opiniones de manera sencilla, ya que se convierten los ficheros XML a texto plano. Fue desarrollada completamente en JAVA, por lo que es multiplataforma, necesita como entrada un fichero XML con todas las opiniones a analizar y como salida muestra cuántas son positivas y cuántas negativas a petición del usuario también retorna el porcentaje, además de una lista con las opiniones negativas y positivas [36]. Aunque es una aplicación de escritorio para devolver las opiniones negativas y positivas y no tiene en cuenta clasificaciones o grupos específicos.

Aunque todas estas herramientas son muy buenas y resolverían el problema, no se podrían utilizar ya que son corporativas y no tienen el código libre expuesto. Pagar la licencia o altos precios para su utilización, debido a que fueron concebidas con fines comerciales, para poder implementar algunas de estas en algún sistema de noticias en Cuba es imposible, puesto que siendo un país con una ley de bloqueo impuesta ninguna corporación de estas crearía tal relación de negocios. Las herramientas de monitoreo de redes sociales como Brandwatch son abarcadoras y algunas de ellas no controlan los artículos de noticias generales o foros. PosNeg Opinion es una aplicación de escritorio para devolver las opiniones negativas y positivas, sin tener en cuenta clasificaciones o grupos específicos.

1.8. Herramientas y tecnologías

Entre las herramientas y tecnologías que son necesarias utilizar para desarrollar la investigación son las descritas a continuación.

1.8.1. Lenguaje de programación

Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas [37].

Se utilizan las siguientes bibliotecas:

NLTK: (Natural Language Toolkit): es un conjunto de técnicas que permiten el análisis y manipulación del lenguaje natural. Se utiliza para la creación de programas en Python que interpretan el lenguaje humano. Permite realizar tareas de transformación y limpieza de documentos tales como: eliminar caracteres especiales, signos de puntuación, convertir todo el texto en minúscula, eliminar palabras comunes o sin significado (conocidas como palabras de paradas) de la lengua en la que está escrito tales como: el, para, de, por, y, un, entre otras [38, 39].

Se investigan las diferentes librerías de paquetes para el preprocesamiento de datos en NLP y que son compatibles en Python puesto que las características innatas de este lenguaje lo convierten en la mejor opción para cualquier proyecto de NLP según [39]. Estas librerías son: NLTK siendo la que se utiliza, CoreNLP, TextBlob, SpaCy, Poligloto y Gensim, aunque también se utiliza.

NLTK, apoya a las tareas como la clasificación, la tokenización, la derivación, el etiquetado, el análisis y el razonamiento semántico. Esta biblioteca es la principal herramienta para el procesamiento de lenguaje natural. Cumple con los requisitos para lo que se persigue, es la biblioteca más popular para el procesamiento del lenguaje natural (NLP) escrita en Python y tiene una gran comunidad detrás [39], siendo esta una gran ventaja. Es una librería para fines académicos. Las demás igual servirían para cumplir el objetivo del desarrollo de la investigación, pero se observó que todas utilizan nltk como base, esto influye si se obtuviera otra para trabajar pues se lograran combinar bien. Aunque fue necesario utilizar la librería de Gensim, para recopilar el modelo Word2vec y utilizarlo en esta investigación.

Se ecnontró en el repositorio GitHub y Sonatype Nexus Repository Manager(repositorio) de la Universidad de Ciencias Informáticas todas las librerías y los paquetes que son necesarios, por ejemplo:

- NLTK versión 3.2.4, aporta todas las funciones necesarias para el preprocesamiento de datos.
- Scikit-Learn, trae consigo algoritmos como SVM y funciones como la lista de stop-words, el algoritmo LDA y otras funciones necesarias en la transformación de datos.
- MatplotLib, librería utilizada para transformaciones numéricas, además de dar formato a la figura que representara SVM.
- Gensim, utilizada como herramienta de Word2vec para la transformación del modelo bolsa de palabras, es una biblioteca de código abierto de Python para modelado de temas, indexación de documentos y recuperación de similitudes [40].

NumPy: es el paquete fundamental para la computación científica con Python. Contiene entre otras cosas: un potente objeto de matriz N-dimensional, herramientas para integrar código C / C ++ y Fortran, álgebra lineal útil, transformada de Fourier y la capacidad de generar números aleatorios. También puede ser utilizado como un eficiente contenedor multidimensional de datos genéricos. Se pueden definir tipos de datos arbitrarios. Esto permite a NumPy integrarse de forma transparente y rápida con una amplia variedad de bases de datos [37, 40].

Sklearn: es una biblioteca de código abierto para tareas de aprendizaje automático para el lenguaje de programación Python. Cuenta con varios algoritmos de clasificación, regresión y agrupación; entre los que se encuentran K-means y DBSCAN, y está diseñada para interoperar con las bibliotecas numéricas y científicas de Python, NumPy y SciPy [41, 40].

Panda: es una biblioteca de código abierto que proporciona estructuras de datos de alto rendimiento y fácil de usar y herramientas de análisis de datos para el lenguaje de programación Python. En particular, ofrece estructuras de datos y operaciones para manipular tablas numéricas y series temporales [42].

Word2Vector es una biblioteca de código abierto y libre que provee una implementación de las arquitecturas BoW para calcular representaciones vectoriales de palabras. La herramienta tiene como entrada un corpus textual y produce los vectores palabra como salida. Primero construye un vocabulario del texto de entrenamiento y luego aprende representaciones vectoriales de palabras. Una forma simple de verificar las representaciones aprendidas es encontrar las palabras más cercanas para una palabra especificada por el usuario. Para observar regularidades fuertes en el espacio vectorial de palabras es necesario entrenar los modelos en grandes conjuntos de datos [32].

MatplotLib: es un módulo de dibujo de gráficas 2D para Python [43, 44].

Otras herramientas a utilizar:
Anaconda: La distribución de código abierto Anaconda facilita el proceso de implementar soluciones relacionadas con manipulación de datos en Python y cuenta con más de seis millones de usuarios. Incluye cientos de paquetes de código abierto, así como el paquete conda y el administrador de entorno virtual para Windows, Linux y MacOS. Conda facilita y agiliza el proceso de instalar, ejecutar y actualizar entornos complejos como SciPy. Anaconda es la base de millones

de proyectos, así como de aprendizaje automático de Amazon Web Services y Anaconda para Microsoft en Azure y Windows. Se decidió que la versión Anaconda 3-2.5.0 es la más adecuada para trabajar con la versión del lenguaje de programación seleccionado, debido a que contiene una amplia colección de librerías empleadas para facilitar el trabajo.
Git es un sistema de control de versiones creado por Linus Torvalds y la comunidad de desarrollo de Linux en 2005 con el objetivo de sustituir el software que empleaban para realizar esta función. El control de versiones es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que se pueda recuperar versiones específicas más adelante. Entre las ventajas que ofrece Git están la velocidad, el diseño sencillo y la capacidad de manejar grandes proyectos. En el presente trabajo se escogió la versión Git 2.19.1.0 para el control de versiones por la compatibilidad que tiene con el IDE seleccionado anteriormente.
Para la presente investigación se utilizará Python 3.7 puesto que para el procesamiento de datos en NLP tiene como requisitos instalar versiones de Python de 3.0 en adelante según [3], además por la variedad de librerías que contiene para el trabajo con el análisis de sentimientos y aprendizaje automático.

1.8.2. Entorno Integrado de Desarrollo (IDE)

Para la implementación se propone utilizar PyCharm 2018.3.4 que es un IDE basado en IntelliJ IDEA que ofrece las siguientes funciones [45]:

- Auto-completamiento de código.
- Señalamiento de errores con soluciones fáciles.
- Posibilita una fácil navegación para proyectos y código.
- Mantiene el código bajo control de chequeos, asistencia de pruebas, refactorizaciones y un conjunto de inspecciones que posibilitan codificar de forma limpia y sostenible.

1.8.3. Editor de texto

Como editor de texto que se usó en la realización del informe y la presentación fue el TeXstudio (LaText).
Puesto que es un programa para la composición tipográfica de textos científicos y una opción disponible para edición de textos con contenido matemático tales como artículos, reportes, libros, entre otros. TEX es en la práctica un estándar

para publicaciones científicas en áreas como matemática, física, computación. Además, es un lenguaje que nos permite preparar automáticamente un documento de apariencia estándar y de alta calidad [46].

1.8.4. Gestor bibliográfico

BibTEX es una de las dos opciones para realizar bibliografías en el editor de texto Latex. Para este caso se realiza una "base de datos" de los libros en un archivo de texto aparte. Este archivo se debe guardar en la misma carpeta del documento con extensión .bib. Este archivo se puede realizar con el Bloc de Notas en Windows o el Editor de Textos en Linux, en general funciona con cualquier editor de texto plano [46].

1.9. Conclusiones parciales

Después de la investigación realizada se definió que:

- Análisis de sentimiento y Minería de opinión son términos interrelacionados pero cumplen diferentes objetivos.
- En esta investigación se seguirán las etapas del proceso de descubrimiento de conocimiento propuestas por Fayyad de 1996.
- En la clasificación de sentimientos se escoge la técnica enfoque basados en máquina de aprendizaje supervisado Máquina de soporte vectorial.
- Como modelo de representación vectorial se definió Asignación Latente de Dirichlet.
- Como Modelo de aprendizaje automático se seguirá con Máquina de soporte vectorial.
- Se escogió algunas de las etapas del análisis de sentimientos definidas por Medhat en el 2014.

Capítulo 2

Propuesta de solución

En el presente capítulo se describen los pasos a seguir para desarrollar el proceso KDD. Además de las técnicas, bibliotecas de lenguaje de programación y algoritmos que se utilizan. La metodología adoptada en este trabajo está basada en las etapas que caracterizan el proceso KDD.

2.1. Esquema general de la propuesta de solución

Pasos a seguir:

1. Realizar la selección de los datos (comentarios) a partir de los artículos más comentados del sitio web de noticias Cubadebate.
2. Preprocesar los comentarios seleccionados, limpiándolos de cualquier signo o palabra que no aporten significado al contenido y convirtiendo las palabras a sus palabras raíces.
3. Transformar el texto a vectores para obtener la matriz término-documento.
4. Aplicar el algoritmo SVM, siendo el algoritmo de Minería de texto y análisis de sentimiento.

A continuación, se muestra en la Figura 2.1 los pasos de la metodología escogida KDD explicada anteriormente, la solución:

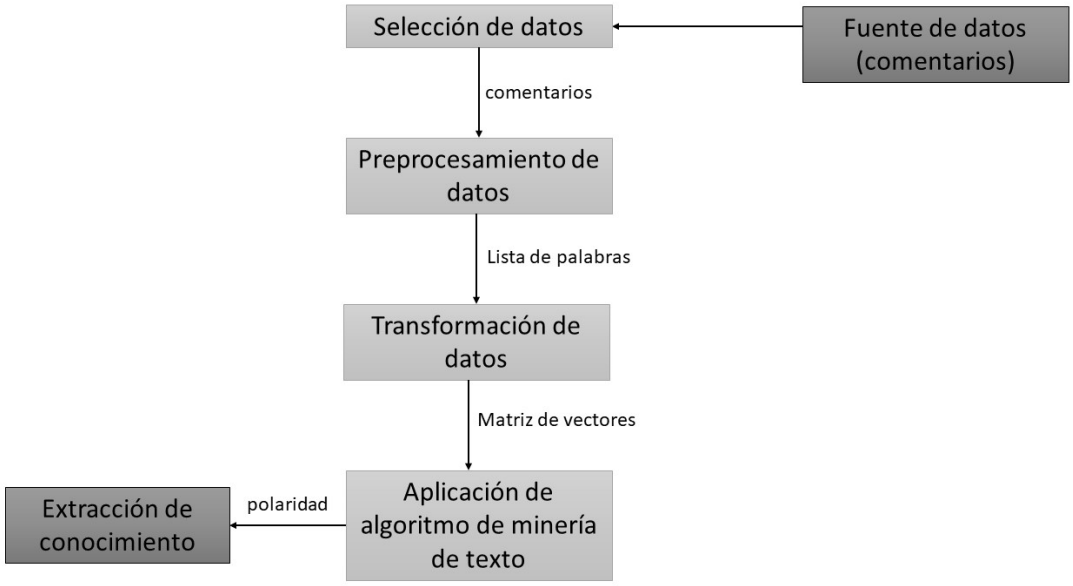

Figura 2.1: Esquema de KDD implementado [elaboración propia]

2.2. Selección de los datos

Al realizar la selección de las fuentes de datos, se escogen los comentarios de artículos publicados en la sección, noticias más comentadas, del sitio web de noticias Cubadebate. Son seleccionan 5 noticias con diferentes temáticas e incluso diferentes cantidades de comentarios, los más polémicos entre sus ramas. Se combinan los comentarios seleccionados en un bloc de notas, puesto que los datos son textos y es una base de datos para textos.

1. La primera fuente es el artículo “ETECSA. Internet en el móvil a partir del seis de diciembre”, obteniendo 2013 comentarios.

2. La segunda fuente es el artículo “Esta página es toda tuya Comparte con Cubadebate tu homenaje a Fidel Castro”, donde se realizaron 4945 comentarios.

3. La tercera fuente es el artículo “En su 20 cumpleaños, Mailen regala las primeras fotos tras el accidente aéreo”, obteniendo 737 comentarios

4. La cuarta fuente es el artículo “Leinier Domínguez se nacionaliza por Estados Unidos-Federación Cubana expresa desacuerdo”, con 584 comentarios

5. La quinta fuente, el artículo “Si de alimentos se trata. Miradas a la industria nacional” con 420 comentarios.

2.3. Preprocesado y transformación de los datos

En esta etapa ya seleccionados los datos (comentarios) se realiza un preprocesamiento para convertir los datos en un lenguaje que la PC pueda entender y transformarlo mediante algoritmos escogidos en el capítulo anterior, para descubrir el conocimiento y cumplir con el objetivo de la investigación.

2.3.1. Preprocesado de datos

A continuación, en la Figura 2.2, se muestran los pasos del preprocesamiento de los comentarios. Existirá como **entrada** de datos el texto a analizar (comentarios) y tendrá como **salida** ya palabras clasificadas en categorías, o sea datos preprocesados. Esto es un importante paso en cualquier proceso de minería de datos, prácticamente implica transformar los datos en bruto en un comprensible formato de modelos de lenguaje natural. Para el procesamiento de datos y su transformación se utilizará como ejemplo el artículo 1 (Dataset 1) en toda la investigación aunque se implementó para todos los dataset.

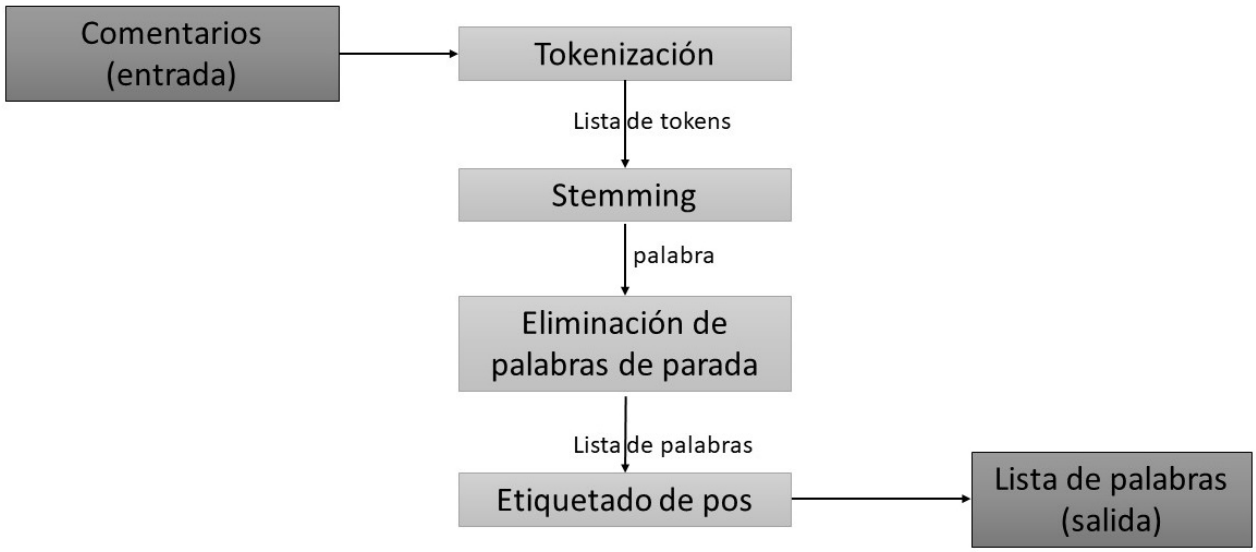

Figura 2.2: Pasos de preprocesamiento de datos [elaboración propia]

Para la **tokenización** se divide la fuente de datos seleccionada en palabras, frases, símbolos u otros elementos significativos llamados tokens. La lista de tokens se convierte en entrada para el procesamiento posterior, como el análisis o la minería de textos. Se utiliza de la librería **nltk**, el método *word_tokenize*, que sirve para dividir una oración en palabras [47]. Se carga el texto de entrada (comentarios o Posts) y se le aplica *word_tokenize*, además se eliminan los espacios en blanco y los números. Se tendrá como salida palabras guardadas en una lista de tokens como se muestra a continuación:

'Que','se', 'recupere', 'pronto','Muchas', 'felicidades', 'y', 'que', 'siga', 'recuperándose', 'rápido', 'y'

En cada idioma siempre existirá un conjunto de palabras de uso común, tales como artículos, pronombres o adverbios. En la librería **sklearn** se encuentra el método *stop_words*, una lista de palabras que busca eliminar la cabeza de la distribución de palabras, donde se ubican todas aquellas palabras de uso común que no aportan significado discriminatorio a los documentos ejemplo: "a", "antes", "otra", "otro". Esta lista se filtra en la lista de palabras producidas por el proceso de indexación.

Stemming:

En el siguiente ejemplo se muestra el proceso de reducir las palabras a sus palabras raíces.

$Palabra$	$Stem$	$Lemma$
$Estudio$	$Estudio$	$Estudio$
$Estudiando$	$Estudiando$	$Estudio$
$Hermoso$	$Hermoso$	$Hermoso$
$Hermosamente$	$Hermosamente$	$Hermoso$

A continuación, se observa ya implementada la función *def_clean*, tokenizado el texto, se transforma las mayúsculas en minúsculas, se eliminan las palabras de parada y se llevan a su forma raíz. Se obtiene una lista de tokens mostrada anteriormente, después de haber realizado el proceso de Stemmer con la función *WordNetLemmatizer* y *PorterStemmer*, de la librería nltk.

'recupere', 'pronto', 'felicidades', 'siga', 'recuperándose', 'rápido', 'fortaleza', 'tenido', 'feliz'

Eliminación de palabras de parada:

En la siguiente Figura 2.3 se muestra esta lista de stop_words, así se sabrá diferenciar entre palabras que no añaden al contenido a palabras que si son pilares para la clasificación. Esta lista se tomó de la librería nltk, en su inicio estaba en inglés, por lo que se tradujo al español y se le agregaron palabras.

"así", "a", "arriba",
"alrededor","atrás","al","aún","alguien","atravesado","antemano", "a", "abajo",
"acerca", "arriba", "algo", "alguna ","ambos","además","aparte", "bien",
"casi", "cada", "cualquiera", "cómo", "como" ,"cualquiera", "casualmente" "de", "
hecho ", "debajo", "dentro", "debido", "durante", "modo", "después", "desde",
"sincero", "algunos", "un", "debajo", "hasta", "eso", "esa", "mismos", "luego",
"allí", "en", "e", "etc", "incluso", "siempre", "todos", "en", "donde", "que",
"mientras", "donde", "en contra", "encontrado", "cuatro", "de", "frente",
"completo", "más", "obtener", "dar", "ir", "encontrar", "fuego", "cinco", "para",
"antiguo", "anteriormente", "cuarenta", "factura", "frecuentemente", "hacia", "ha"

Figura 2.3: Lista stop_words [elaboración propia]

Etiquetado de pos

Se utiliza el algoritmo *p*os_tag de nltk que va a reconocer de la lista de palabras en el texto que son adjetivos, adverbios, verbos, sustantivos y así se vería ya reconocido el patrón. Siendo:

- nombre: [”NN”, ”NNS”, ”NNP”, ”NNPS”]
- pronombre: [”PRP”, ”PRP”, ”WP”, ”WP”]
- verbo: [”VB”, ”VBD”, ”VBG”, ”VBN”, ”VBP”, ”VBZ”]
- adjetivo: [”JJ”, ”JJR”, ”JJS”]
- adverbio: [”RB”, ”RBR”, ”RBS”, ”WRB”]

A continuación se muestra como identifica cada palabra en la oración.

(’recupere’, RB"), (’pronto’, "JJ"), (’felicidades’, "NNS"), (’siga’, "VBP"), (’recuperándose’, "JJ")

Pseudocódigo del algoritmo de pre-procesamiento

Entrada: Diccionario con los comentarios en D
Salida: Comentarios pre-procesados en P

Crear diccionario P para guardar los comentarios procesados, una T para guardar lista de tokens
Para cada comentario en D Hacer
Transformar texto a minúscula
Eliminar signo de puntuación
Eliminar palabras de parada
Guardar en una lista de tokens T

Retornar T

Una nube de términos es un grupo de palabras claves etiquetadas en diferentes ubicaciones, formas, tamaños o colores, en forma de nube. Normalmente las de mayor tamaño y colores intensos, reflejan las temáticas de mayor importancia (por relevancia, volumen de contenido o actualidad) siendo las menos significativas aquellas más pequeñas y de colores más degradados. El principal objetivo de la nube de términos es facilitar al usuario la búsqueda de información relevante al indicarle las palabras más frecuentes en los grupos de comentario, representadas por esas etiquetas.
Luego de haber preprocesado el texto, se realiza un algoritmo para ayudar a la visualización de palabras más frecuentes. Una herramienta de visualización, del paquete wordcloud ayuda a crear nubes de palabras colocando palabras en un lienzo al azar, con tamaños proporcionales a su frecuencia en el texto. En la Figura 2.4 se muestra el resultado de wordcloud, y en la Figura 2.5 se muestra contabilizado las palabras que aparecen.

Dificultades que se encontraron:

- La lista de stop_words.py venía en la librería nltk en inglés, por lo que se creó una lista de palabras de parada en español.

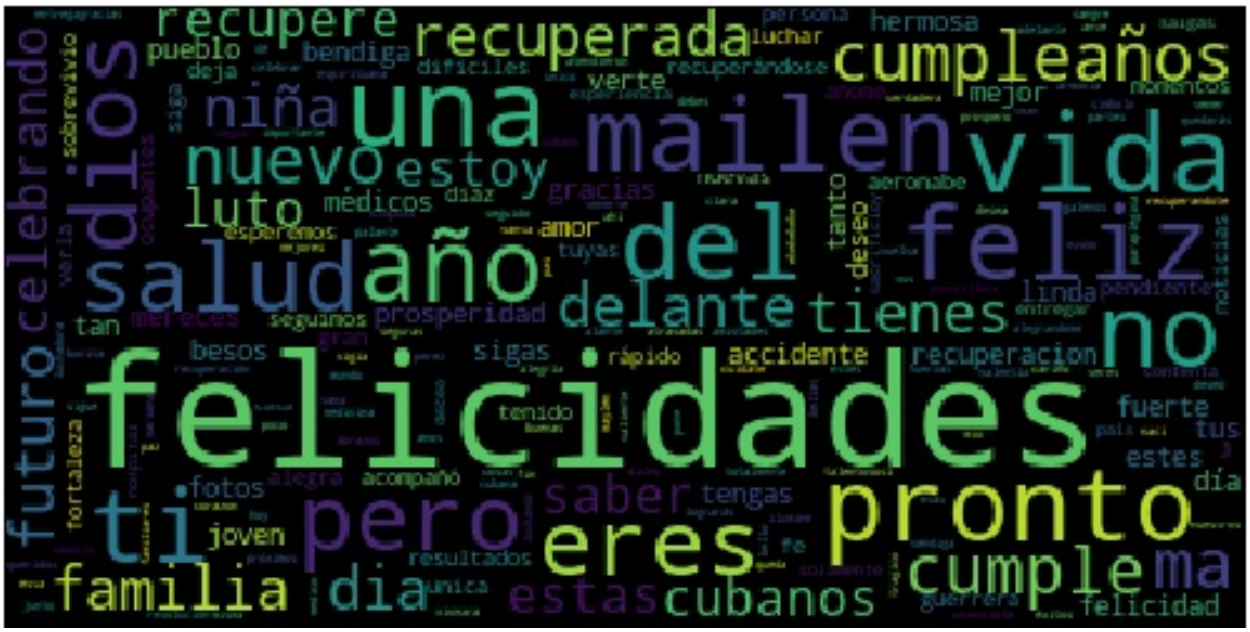

Figura 2.4: Palabras frecuentes ya preprocesado el texto->wordcloud [elaboración propia]

- En el preprocesado de datos se observó que cuando se tiene como entrada en esta fase mucha carga de datos demora más de 6 minutos mostrar resultados y cuando los muestra los muestra incompletos.

2.3.2. Transformación de los datos

Ya aplicado el procesamiento de lenguaje natural se aplica la transformación, mediante modelos de representación vectorial, es un aspecto clave en la categorización de tareas de textos. Este tiene como propósito categorizar documentos en un número fijo de categorías predefinidas. Cada documento puede ser clasificado en múltiples, exactamente una o en ninguna categoría en absoluto. La clasificación de documentos es vista como una tarea de aprendizaje supervisado, el objetivo es utilizar el aprendizaje automático para clasificar automáticamente los documentos en categorías, basadas en documentos previamente etiquetados [20].

Modelo bolsa de palabras

A diferencia de los seres humanos, las computadoras solo pueden entender los números. Por lo que en el procesamiento de lenguaje natural se tiene que representar palabras en un formato numérico que sea comprensible para las

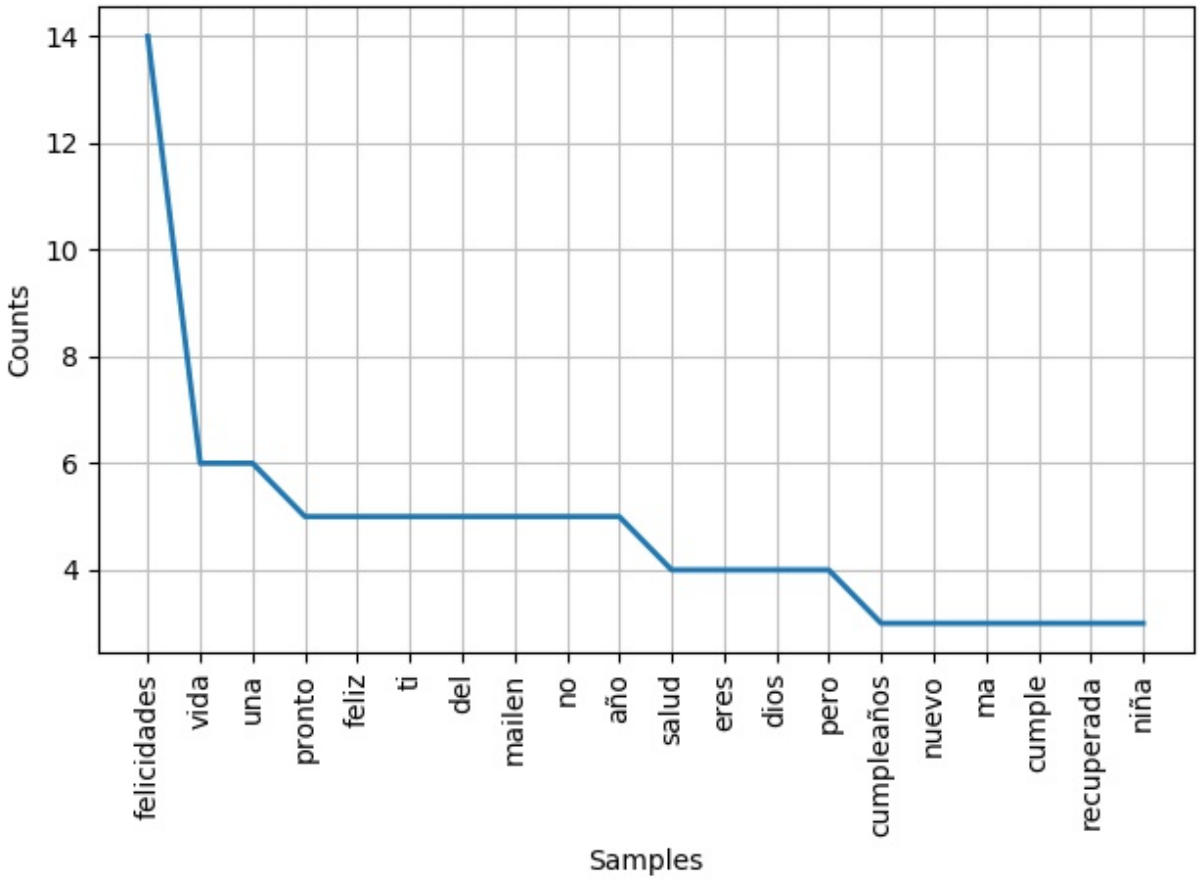

Figura 2.5: Palabras frecuentes ya preprocesado el texto->diagrama [elaboración propia]

computadoras. La inserción de palabras se refiere a las representaciones numéricas de las palabras.
Actualmente existen varios enfoques de integración de palabras aunque los que se utilizarán:

- Bolsa de palabras
- Esquema TF-IDF
- Word2Vec

Llamamos vectorización al proceso general de convertir una colección de documentos de texto en vectores de características numéricas. Esta estrategia específica (tokenización, conteo y normalización) se denomina representación de Bolsa de palabras. Los documentos se describen por ocurrencias de palabras mientras se ignora por completo la información

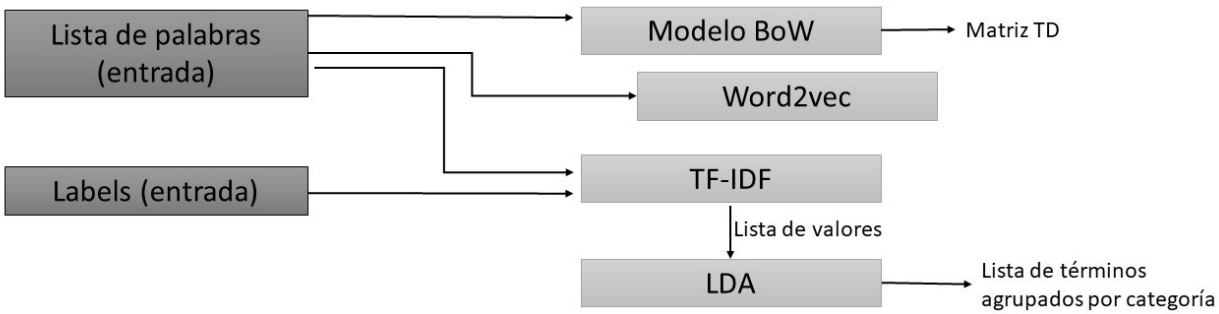

Figura 2.6: Pasos para la transformación de datos [elaboración propia]

de posición relativa de las palabras en el documento. La entrada al modelo será la lista de tokens *f*inal_words y la salida serán los vectores. Crea un diccionario de palabras únicas del corpus.

Matriz término-documento

Para obtener la matriz término-documento se necesitó la función *CountVectorizer* que mediante esta se transformará la lista *final_words* en 0 y 1 que serán los valores de la matriz ya como último mediante la librería *pandas* defines las columnas y las filas, quedando como resultado la Figura 2.7, donde cada columna representa un término de la lista *final_words*, cada renglón un documento y cada elemento o celda de la matriz la ocurrencia del término en el documento.

En total mostró la matriz TD, 329 FILAS x 211 columnas, para el caso del Dataset 1.

Word2Vec

Es necesario este método para convertir las palabras en algún conjunto de vectores numéricos para luego utilizarlos en la transformación. La idea subyacente aquí es que las palabras similares tendrán una distancia mínima entre sus vectores. Con bolsa de palabras, el orden de las palabras en la historia no influye la proyección y predice la palabra actual basado en el contexto.

Entrenar el modelo es bastante sencillo. Simplemente se crea una instancia de Word2Vec, extraída de la librería Gensim y pasa por entrada la lista *final_words*. Word2Vec usa todos estos tokens para crear internamente un vocabulario

M-TD:	abrazos	accidente	acompañó	adelante	aeronabe	ahí	alante	alegr
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0
17	0	0	1	0	0	0	0	0

Figura 2.7: Matriz término-documento [elaboración propia]

(conjunto de palabras únicas). En la Figura 2.8 muestra que existe un vocabulario de 29 palabras únicas.

```
Word2Vec(vocab=29, size=150, alpha=0.025)
```

Figura 2.8: Vocabulario de word2vec [elaboración propia]

Ahora incluso se podría usar Word2Vec para calcular la similitud entre dos palabras en el vocabulario invocando la función de similitud *most_similar* y pasándole las palabras relevantes. Si se realiza una similitud entre dos palabras idénticas, la puntuación será 1.0, ya que el rango de la puntuación de similitud siempre estará entre [0.0-1.0].

TF-IDF

Para alimentar los modelos predictivos con los datos de texto, primero se debe convertir el texto en vectores de valores numéricos adecuados para el análisis estadístico. Esto se puede lograr con las utilidades de *sklearn.feature_extraction.text*,

generalmente es un proceso de colección de documentos de texto en vectores de características numéricas. Existen muchos modelos para convertir datos de texto en vectores, uno de los más utilizados en todos los proyectos académicos que sirve como base es Frecuencia de término- documento inverso (TF-IDF), puesto que reduce las palabras repetidas y t dice cual aparece con más frecuencia [18].

Tf-idf se utiliza para determinar qué palabras de un corpus pueden ser favorable al uso en función de la frecuencia del documento de cada palabra. La representación de Tf-idf se encuentra entre los mejores enfoques para retirar documentos y etiquetarlos. Esta técnica calcula un valor para cada palabra en un documento a través de una inversa proporción de las frecuencias de la palabra en un cierto documento y al porcentaje de documentos a los que la palabra aparece en el documento [20].

- Frecuencia de término: Esto resume con qué frecuencia aparece una palabra dentro de un documento.
- Frecuencia inversa de documentos: Esto reduce las palabras que aparecen mucho en los documentos.

El esquema de pesos más usado es la frecuencia de aparición de los términos en los documentos (Term Frequency / Inverse Document Frequency; TF-IDF) para expresar el peso de un término en un documento es el producto de su frecuencia de aparición en dicho documento (TF) y su frecuencia inversa de documento (IDF):

$tf - idf(w, d) = tf(w, d) * idf(w, d)$

El corpus, que es donde se tiene guardado los dataset, se dividirá en dos conjuntos de datos, entrenamiento y prueba. Los conjuntos de datos de entrenamiento serán usados para ajustar al modelo y las predicciones en los datos de prueba. Para realizar esto el método *train_test_split* de la librería de datos sklearn. Los datos de entrenamiento tendrán el 70 % del corpus y los datos de prueba el restante 30 %, que será el parámetro *test_size=0.3*. Siquiendo esto, se tiene ejemplo en el dataset 1, 230 datos para entrenamiento y 99 para pruebas.

```
Datos de entrenamiento entrenamiento y prueba del predictor: (230, 166) (99, 166)
```

Figura 2.9: Cantidad de datos para entrenamiento y prueba [elaboración propia]

Definido los datos de pruebas y entrenamiento se está listo para transformar datos categóricos de tipo string en conjuntos de datos con valores numéricos, donde el modelo puede entenderlo. Finalmente, se transforma *Train_X* y *Test_X* en vectores *Train_X_Tfidf* y *Test_X_Tfidf*.

La Figura 2.10 muestra los conjuntos de datos procesados por TF-IDF, donde se realizó la transformación numérica

y después en las columnas contendrán para cada fila una lista de números enteros únicos y su importancia asociada calculada por TF-IDF. Los valores de tf-idf más altos que tienen las palabras implican que tienen una relación más estrecha en el documento que aparecen:

```
(0, 13)   1
(1, 62)   1
(2, 66)   1
(3, 75)   1
(4, 203)  1
(5, 3)    1
(6, 61)   1
(7, 76)   1
(8, 31)   1
(9, 12)   1
(10, 45)  1
(11, 171) 1
(12, 81)  1
(13, 35)  1
(14, 175) 1
```

Figura 2.10: Resultado de TF-IDF [elaboración propia]

LDA

Ya como último paso de la transformación de datos, se debe implementar el algoritmo de representación gráfica y estadística LDA. Se crea un objeto para el modelo LDA y se debe entrenarlo en la matriz de documento-término. En dos pasos se realiza el algoritmo:

1. Los artículos / documentos se producen a partir de una mezcla de temas. Cada artículo pertenece a cada uno de los temas en cierta medida (cada artículo se compone de una distribución de temas).

2. Cada tema es un modelo generativo que genera palabras del vocabulario con ciertas probabilidades. Las palabras que suelen aparecer juntas tendrán más probabilidad (cada tema está hecho de alguna distribución de palabras).

Pseudocódigo del algoritmo LDA

Entrada: Diccionario de comentarios D

Salida: Listado de términos ordenados por frecuencia L

Pre-procesar D

Crear el corpus T a partir de D

Obtener listado de términos ordenados por frecuencia L

Retornar L

2.4. Clasificador Máquina de soporte vectorial

Una vez ejecutados los algoritmos, los datos transformados se someten a la clasificación para detectar su polaridad. Permitirán con la información que se extraerá, a los analistas humanos completar el análisis iniciado por el texto mediante la visualización una herramienta.

En la implementación del algoritmo SVM o cualquier otro tipo de clasificador, es necesario pasarle un conjunto de datos que ya estén manualmente clasificados. En este caso *label_data*, extrae de un excel y guarda en un arreglo los comentarios clasificados manualmente en -1(negativo), 0(neutro), 1(positivo), enseñandole al clasificador como debe comportarse. Se muestra en la Figura 2.11 el *label_data* con la polaridad.

Al realizar este mismo proceso en todos los dataset se observa un desbalance en los datos, representado en las Figura 2.12, 2.13, 2.14, este problema existe cuando hay una gran diferencia entre los porcientos de los datos, siendo esto un caso peculiar llamado **problema de balance**. Este problema se presenta en un conjunto de datos que tienen una gran cantidad de datos de cierto tipo (clase mayoritaria), mientras que el número de datos de otro tipo es considerablemente menor. El desbalance de los datos presenta problemas y compromete el proceso de aprendizaje, puesto que los algoritmos de la máquina de aprendizaje espera datos con una distribución balanceada para su clasificación.

Aunque en las Figura 2.14, 2.15 se observa un balance entre los datos. Dígase que polaridad 1 (positivo los datos), polaridad 0 (neutro los datos), polaridad -1 (negativo los datos).

1
1
1
1
1
1
1
1
1
1
1
1
1
0
0
1
1
1
1
1

Figura 2.11: Label_data del dataset 1 (Artículo 1) [elaboración propia]

Para problemas de desbalance se aplica *SGDClassifier*, clasificador de máquina de soporte vectorial, este estimador implementa modelos lineales regularizados con aprendizajes de descenso de gradiente estocástico (SGD). Este clasificador encontrará el hiperplano de separación óptimo usando SVC reemplazándolo con SGDClassifier y observando las diferencias para clases que están desbalanceada.

El objetivo de SGDClassifier es encontrar un separador hiperplano óptimo para clases desbalanceadas. Primero se en-

Figura 2.12: Balance de datos del dataset 1 (Artículo 1): 90 % de datos con polaridad 1 y 10 % con datos de polaridad 0 [elaboración propia]

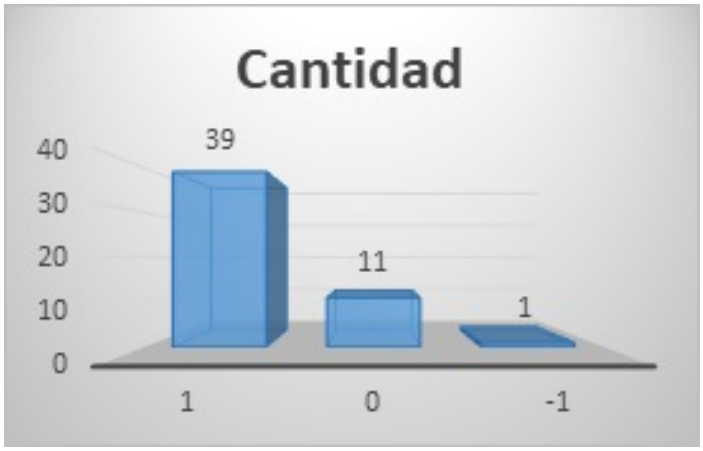

Figura 2.13: Balance de datos del dataset 2 (Artículo 2): 76.47 % de datos con polaridad 1, 21.56 % de datos con polaridad 0 y 1.96 % con datos de polaridad -1 [elaboración propia]

cuentra el plano de separación con SVC plano como se muestra en la Figura 2.17, donde muestra los punto de datos que es un peso diferente de acuerdo con su importancia relativa en la clase, de modo que los diferentes puntos de datos tengan una contribución diferente al aprendizaje de la superficie de decisión siendo esta idea la idea básica de la máquina de soporte vectorial ponderada (weighted y non weighted). Se realiza el mismo proceso para mostrar SGDClassifier en la Figura 2.18 aunque luego se traza el hiperplano de separación con corrección automática para problema de desbalance y observándose.

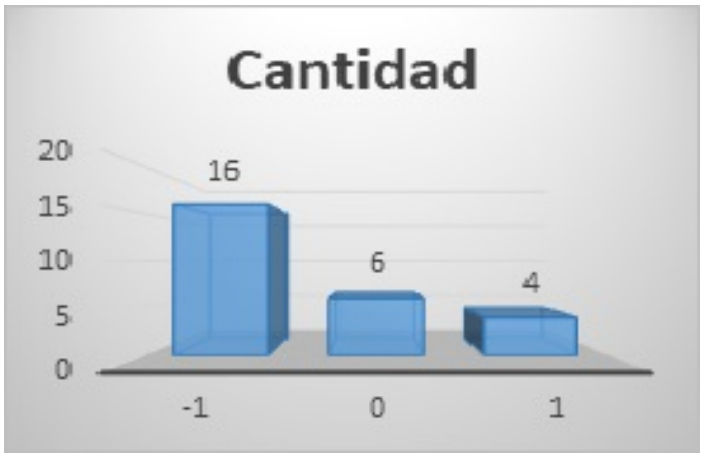

Figura 2.14: Balance de datos del dataset 3 (Artículo 3): 61.53 % de datos con polaridad -1, 23.07 % de datos con polaridad 0 y 15.38 % de datos con polaridad 1

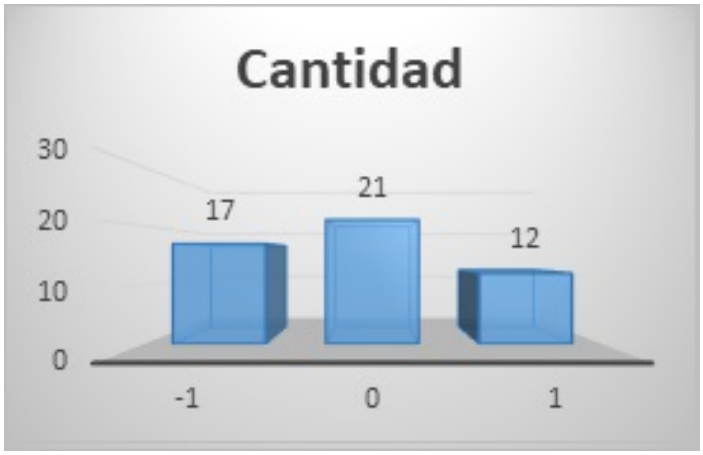

Figura 2.15: Balance de datos del dataset 4 (Artículo 4): 34 % de datos con polaridad -1, 42 % de datos con polaridad y 24 % de datos con polaridad 1 [elaboración propia]

Se utilizará los datos de entrenamiento y los datos de prueba para predecir mediante el clasificador SVM. Aplicado ya el modelo máquina de soporte vectorial en la Figura 2.18, pues sirve para mostrar en un hiperplano la información extraída de los datos de entrenamiento y los datos predictores. Para la implementación del clasificador SVM, se probaron diferentes *Kernels*: linear, sigmoidal, polynomial y RBF. Obteniendo los 3 últimos los mismos valores, esocgiendo RBF para desarrollar.

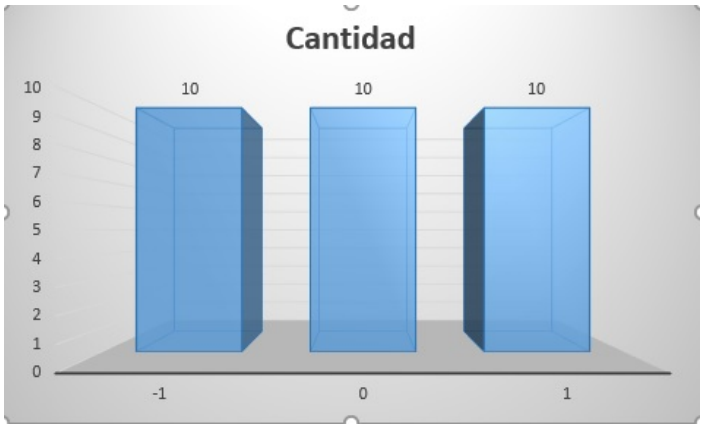

Figura 2.16: Balance de datos del dataset 5 (Artículo 5): 33.33 % de datos con polaridad -1, 33.33 % de datos con polaridad 0, 33.33 % de datos con polaridad 1 [elaboración propia]

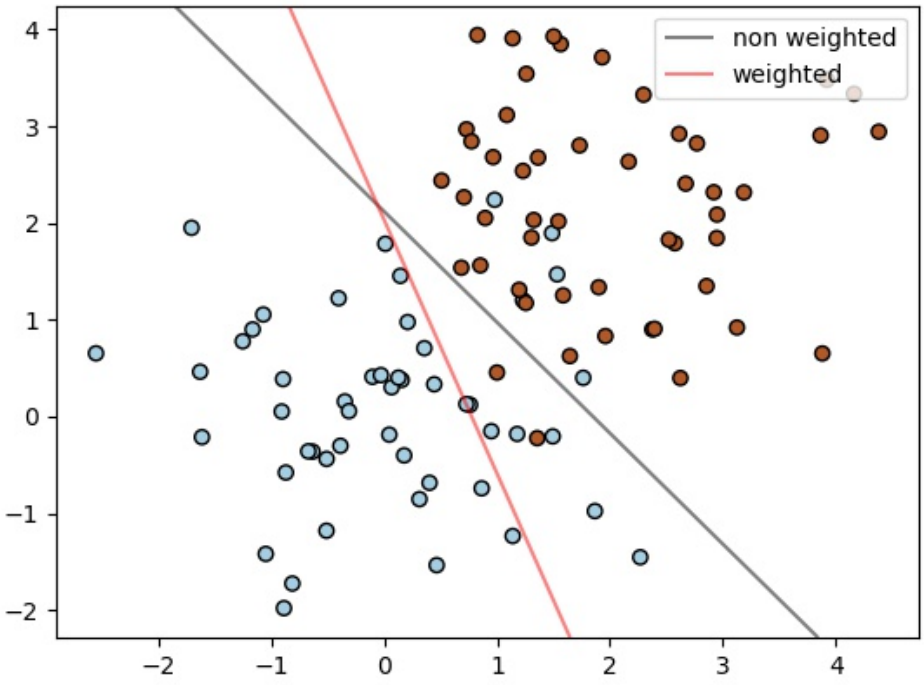

Figura 2.17: SVC

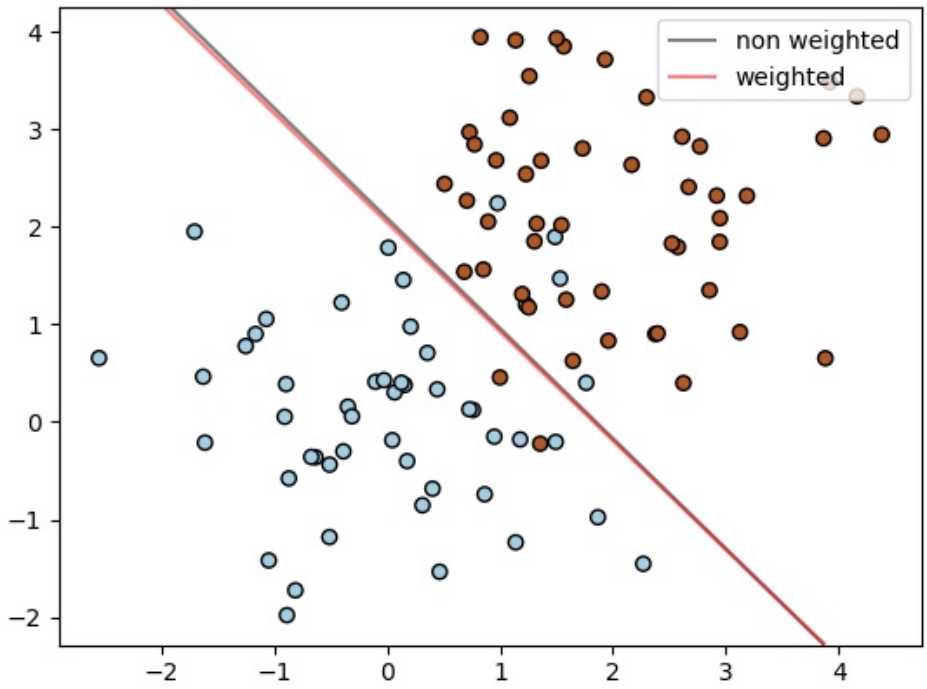

Figura 2.18: SVM tipo SGDClassifier [elaboración propia]

Al aplicar la predicción de SVM con la librería *NumPy* mostrará el resultado de la clase dominante siendo *[Felicidades]* que se entiende por el contexto del artículo como una clase positiva, aunque lo que significa que está ignorando por completo a la clase minoritaria en favor de la clase mayoritaria, esto ocurre cuando existe un problema de balance.

En la Figura 2.19 muestra el resultado de aplicar *predict* a el dataset 1 (Artículo 1), muestra que la palabra felicidades es la que predomina la polaridad, siendo esta positiva entonces se puede llegar a la conclusión de que el dataset 1 (Artículo 1) es positivo.

```
Predición de SVM: ['felicidades' 'felicidades' 'felicidades' 'felicidades
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
```

Figura 2.19: Aplicado función *predict* a dataset 1 [elaboración propia]

2.5. Conclusiones parciales

Presentada la propuesta de la solución para descubrir conocimiento de los comentarios de usuarios en el sitio web de noticias Cubadebate, se concluye que:

1. Se logró la transformación de datos de texto en una matriz término-documento con los algoritmos TF-IDF, SVM y LDA para convertir los datos en números donde reflejen en el algoritmo de clasificación de sentimientos si son positivos, negativos y neutros.

2. La obtención de la solución para la detección de polaridad en comentarios de los usuarios sobre los artículos del sitio web de noticias Cubadebate, posibilitó incorporar un nuevo instrumento para el análisis de comentarios de usuarios.

3. Interpretando el conocimiento o la información extraída se obtuvo que el algoritmo máquina de soporte vectorial con la función *predict* muestra el sentimiento que predomina en el artículo.

Capítulo 3

Validación de los algoritmos implementados

En el presente capítulo se realiza la validación de la investigación realizada mediante varios criterios para determinar la precisión con que se escogieron los algoritmos y métodos en el anterior capítulo y así responder al objetivo que se trazó.

3.1. Entorno de pruebas

Para la realización de las pruebas a la solución desarrollada se trabajó con las tecnologías y herramientas seleccionadas (ver epígrafe 1.8) empleando como plataforma el sistema operativo Windows 10 Pro de 64 bits. Las características de hardware sobre el que se trabajó fue una laptop TOSHIBA Satellite L55 con un microprocesador Intel(R) Core (TM) i3-5005U a 2.00 GHz y una memoria RAM DDR 3 de 4GB.

3.2. Aplicación de métricas de evaluación de rendimiento en algoritmos implementados

La evaluación experimental de un clasificador usualmente mide la efectividad, que para este caso particular corresponde a la habilidad de tomar la decisión de clasificación correcta. Para poder evaluar las soluciones de los distintos algoritmos de clasificación que se aplican, es necesario disponer de un conjunto de datos, sobre el cual se ejecutan estos algoritmos, los cinco dataset de entrenamiento utilizados y que cuentan con diferentes particularidades. Las métricas que se emplean dadas por la librería sklearn para evaluar los resultados obtenidos, son:

- **accuracy_score**: mide la precisión que posee algún algoritmo sobre la fuente de datos indicada
- **predict**: da como resultado la predicción del clasificador sobre la fuente de datos entrada.

Para analizar el clasificador escogido se decide compararlo con otros clasificadores, se escoge a *Naive Bayes*, puesto que es el segundo clasificador más utilizado para la categorización de textos y el clasificador *RandomForestClassifier* un enfoque basado en conjuntos para identificar datos relevantes [48], aunque como otros clasificadores no está diseñado para tratar con problemas de datos no balanceados.

En la figura Figura 3.1 se observa una ilustración con los valores resultantes de *accuracy* y el algoritmo que mayor valores tiene sobre la fuente de datos es el clasificador RandomForest, por lo que prueba que el algoritmo escogido en la investigación no es el más eficiente sobre la precisión de sus datos. El modelo escogido, la máquina de soporte vectorial tiene una precisión de 4.34 % de precisión que no lo hace viable.

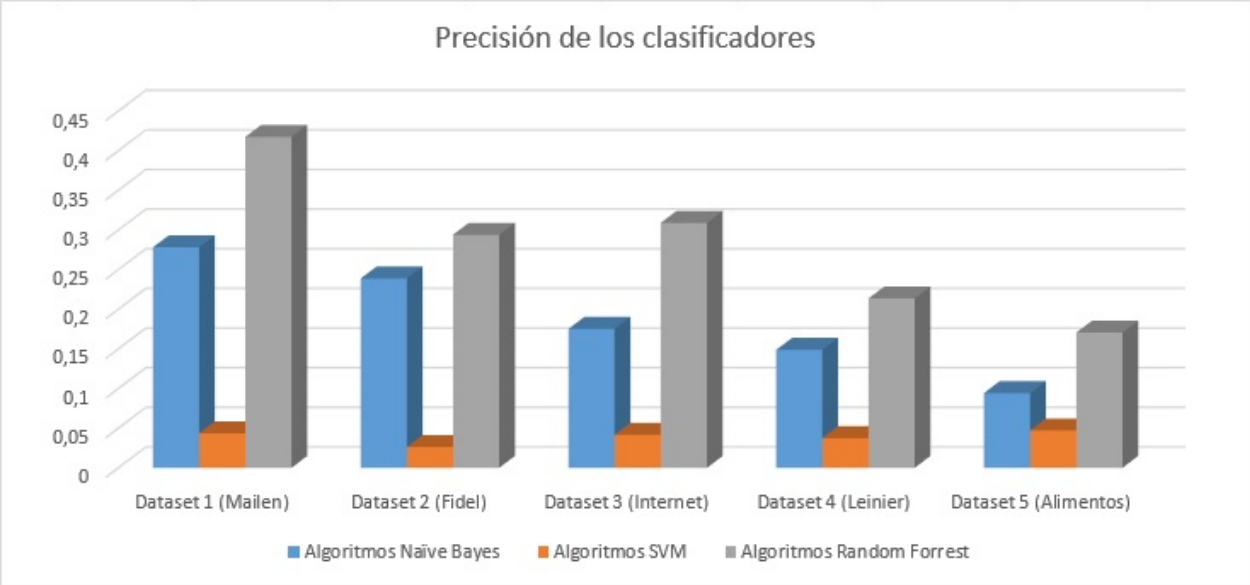

Figura 3.1: Precisión entre los clasificadores. Tomado por [elaboración propia]

Se necesita datos para usar en la demostración, así que se obtuvo el conjunto de datos de las colecciones de prueba de recuperación de texto. El conjunto de datos se dividirá en dos partes las X, que son datos predictores, siendo las variables de entrada (matriz término-documento) y las Y que son los datos tarjeta (labels). También cada tipo de datos se dividirá en un conjunto de datos para entrenamientos y otro para pruebas. Se muestra a continuación la precisión hallada a los

conjuntos de datos divididos y siendo 1 su máximo valor entonces se puede proseguir adelante en los próximos pasos obteniendo una buena precisión en los datos. En la Figura 3.1 se observan los resultados obtenidos a partir de los valores que da como resultado la precisión de cada clasificador escogido sobre la fuente de datos existente.

Como conclusión se demuestra con la métrica *accuracy* los valores altos los proporciona el clasificador *RandomForestClassifier*, superando a *Naive Bayes* y *SVM*, siendo el algoritmo escogido en las 5 dataset probada da entre los valores 0 y 0.1 nunca llegando a este último valor. Por lo que nos da la validación del clasificador como mejor para utilizar *RandomForestClassifier* con la métrica *accuracy*. Para asegurarse de las validación del clasificador, se evaluará con otras medidas de evaluación.

3.2.1. Medidas Precision y Recall para la efectividad en categorización de texto

La efectividad de los clasificadores en la mayoría de los casos es medida en términos de las clásicas nociones de Recuperación de información, como son Precision π y Recall ρ, pero adaptadas al caso de categorización de texto. π está definida como la probabilidad condicional [30].

$$P(\Phi(d_x, c_i)) = T|\Phi(d_x, c_i) = T$$

La probabilidad de que si un documento al azar d_x es clasificado bajo c_i, la decisión es correcta.

Estos valores deben ser entendidos como probabilidades subjetivas, ya que miden la expectación del usuario con respecto a si el sistema se comportará correctamente al momento de clasificar un documento cualquiera bajo una categoría ci. Estas probabilidades pueden ser estimadas en términos de una tabla de contingencia, ver Figura 3.2 para la categoría ci en un conjunto de prueba (test) dado, donde, FPi (falsos positivos) es el número de documentos del conjunto de entrenamiento que han sido incorrectamente clasificados bajo ci; TNi (verdaderos negativo), TPi (falsos negativos), FNi (falsos negativo) se ajustan a la misma definición dependiendo el caso.

Con estos resultados se estima la predicción y precisión de la investigación realizada a través de los algoritmos utilizados, por un conjunto de datos. Primero se carga los conjunto de datos para analizar. Luego se procesan los textos y se construye un espacio con los vectores asignados a cada post. Ya como último se imprimen los resultados de aplicar las métricas ver Figura 3.3.

Categoría ci		Expertos en juzgar	
		Si	No
Clasificadores	Si	TP	FP
	No	FN	TN

Figura 3.2: Contingencia para i categorías. Tomado por [30]

```
from sklearn.metrics import accuracy_score, precision_recall_curve, precision_score, recall_score

print((np.unique(y_train_pred)))

print('Precisión del clasificador X %.7f' % accuracy_score(y_true='', y_pred=''))

print('Precision average macro de X %.7f' % precision_score(y_true='', y_pred='', average='macro'))

print('Precision average micro de X %.7f' % precision_score(y_true='', y_pred='', average='micro'))

print('Recall average macro de X %.7f' % recall_score(y_true='', y_pred='', average='macro'))

print('Recall average micro de X %.7f' % recall_score(ty_true='', y_pred='', average='micro'))
```

Figura 3.3: Métricas elegidas para evaluación [elaboración propia]

Se escoge para precision-recall el average tipo macro y micro, se observan que los dos se comportan iguales con respecto a sus valores por lo que se escoge a desarrollar macro también porque este tipo de métrica, calcula las métricas para cada label y encuentra la media no ponderada. Ya evaluados los clasificadores sobre la fuente de datos elegidas, dan los mismos resultados validados arriba sobre la métrica **accuracy**, RandomForestClassifier es el clasificador con mayores valores.

Ya entonces aplicadas las métricas Precision-Recall, ver Figura 3.4, 3.5 sobre los algoritmos para comparar Máquina de soporte vectorial (escogido), Naive Bayes y RandomForest, este ultimo posee los mejores valores para escoger el algoritmo a desarrollar. Obteniendo una precision macro de 18 % y 19 % en Recall, concluyendo es que si un comentario es analizado bajo el clasificador RandomForest posee una probabilidad de un 18 % de ser correcto. Aunque estos valores siguen siendo bajos para ser la probabilidad de escoger el clasificador correcto.

	Naïve Bayes		SVM		Random Forrest	
	Precision (Macro)	Recall (Macro)	Precision (Macro)	Recall (Macro)	Precision (Macro)	Recall (Macro)
Dataset 1 (Mailen)	0,0907037	0,0963855	0,0002619	0,0060241	**0,1871653**	**0,1927711**
Dataset 2 (Fidel)	0,047032	0,0489237	0,0000514	0,0019569	**0,0744341**	**0,0763209**
Dataset 3 (Internet)	0,0314572	0,0339426	0,0001088	0,002611	**0,0889213**	**0,0913838**
Dataset 4 (Leinier)	0,0187866	0,0202808	0,0000585	0,0015601	**0,0421927**	**0,0436817**
Dataset 5 (Alimentos)	0,0077133	0,0095238	0,0000898	0,0019048	**0,0362929**	**0,0380952**

Figura 3.4: Tabla comparativa entre los clasificadores aplicando precision-recall. Tomado por [elaboración propia]

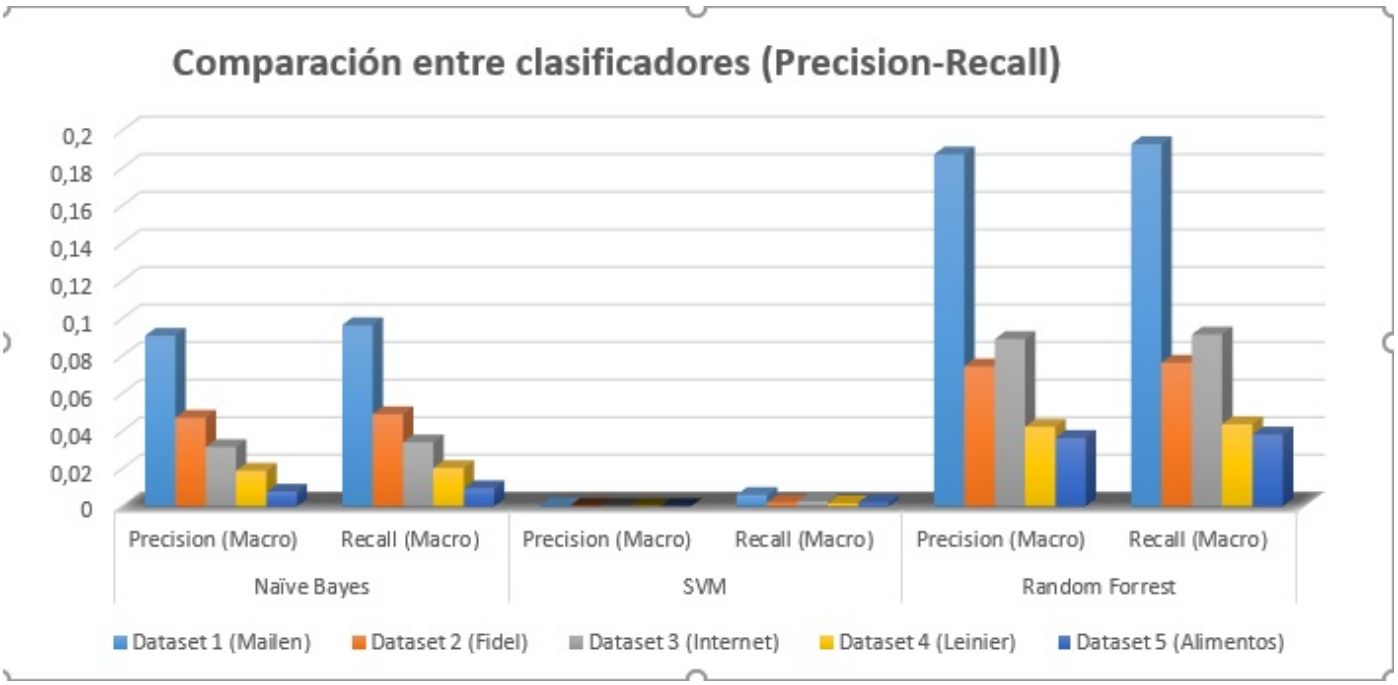

Figura 3.5: Ilustración comparativa entre los clasificadores aplicando precision-recall. Tomado por [elaboración propia]

Después de aplicar las métricas Accuracy-score y Precision-Recall se concluye que el algoritmo escogido (máquina de soporte vectorial) aunque era uno de los mejores para categorización de texto según [19, 18, 20, 21, 22, 23, 24], resultó ser el de menores valores en la validación. Realizando un análisis de este resultado se obtienen dos causas:

1. Los algoritmos se van a comportar de manera diferente en cada fuente de datos que se obtiene como entrada. En las bibliografías estudiadas se utilizaban base de datos que ya se encontraban estructuradas y categorizadas por temas diferentes a los sistemas de noticias, identificando emociones en otros sectores. Al contrario de esta

investigación donde se diseñó una base de datos a partir de los comentarios seleccionados en Cubadebate. Por lo que no se va a comportar de igual manera.

2. La clasificación a sistemas de noticias ha sido uno de los sectores menos investigado como se muestra en la Figura 3.6. Por lo que no se va a obtener los mismos resultados que en otros dominios, de ahí la importancia de adentrarse en este dominio e implementarlo, por su importancia social actual en el país.

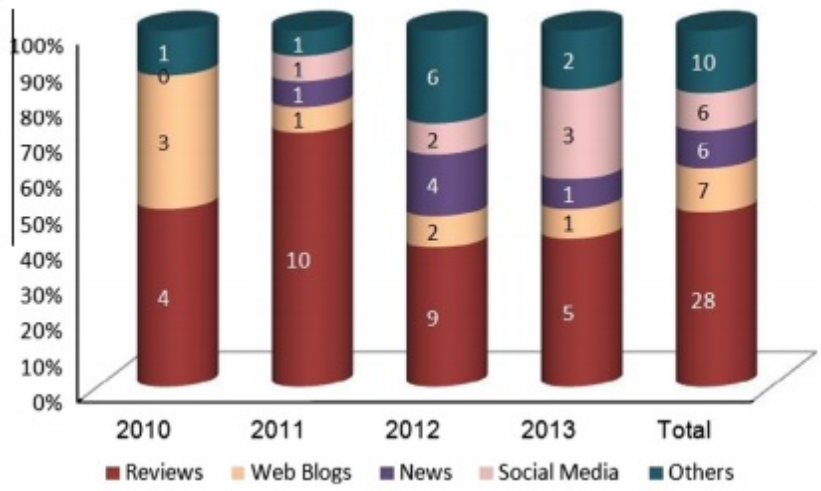

Figura 3.6: Número y porciento de artículos dirigidos a diferentes dominios de textos. Tomado de [7]

Realizando una comparación con respecto a las predicciones de los 5 dataset se observa en los Anexos 1, 2, 3, 4 y 5 las predicciones encontradas por cada uno, dando como resultado: Artículo 1 positivo y los Artículos 2, 3, 4 y 5 negativos. Obtenidos estos resultados se puede realizar un análisis social de porque son negativos los artículos señalados y darle solución.

3.3. Comparación entre los algoritmos implementados

A parte de las métricas también se comparan los algoritmos implementados sobre la base de los dataset. Algoritmos como word2vec, su vocabulario y su puntuación máxima, la puntuación del modelo LDA sobre las fuentes de datos y la matriz resultante de cada uno de los dataset.

En la Figura 3.5 se observan los valores de salida de cada algoritmo sobre los dataset. El tamaño de la matriz sobre cada fuente de datos, el vocabulario de palabras únicas y la medida de puntuación, y la precisión del modelo LDA sobre los dataset. En los resultados se muestra que el dataset 4 muestra mayores resultados de la mayor matriz término-documento, mayor precisión en el modelo LDA y score de la herramienta word2vec, aunque el dataset 3 posee mayor vocabulario de palabras únicas

	BoW	Word2vec (vocab)	Word2vec (score)	LDA
Dataset 1 (Mailen)	[329 rows x 211 columns]	29	(-8,3)	(-1891.59875188)
Dataset 2 (Fidel)	[1089 rows x 675 columns]	30	(-33,45)	(-6601.76419802)
Dataset 3 (Internet)	[823 rows x 492 columns]	32	(-24.5)	(-4839.96888777)
Dataset 4 (Leinier)	[1411 rows x 824 columns]	31	(-41.0)	(-8703.81609914)
Dataset 5 (Alimentos)	[1030 rows x 675 columns]	29	(-33.6)	(-6419.19830273)

Figura 3.7: Algoritmos escogidos sobre los dataset. Tomado por [elaboración propia]

3.4. Conclusiones parciales

En este capítulo se han presentado un conjunto de técnicas en calidad de prueba que permitieron contrastar empíricamente los resultados obtenidos y validar su efectividad. Estas pruebas han corroborado la precisión de la solución implementada y demostraron su utilidad.

- Se demostró que el clasificador escogido de análisis de sentimiento fue el que obtuvo menores valores en la precisión con el que analizaría un dataset, obteniendo RandomForest los mayores resultados, dando como conclusión que puede ser por la fuente de datos puesto que cada algoritmo se comporta diferente por los datos de entrada.

- La aplicación de mediciones basadas en la precisión y predicción del algoritmo verificó la correspondencia entre los datos obtenidos en los experimentos y aquellos obtenidos de la solución implementada, lo cual demostró además que los valores de los modelos Bolsa de palabra y el algoritmo LDA aumentan en la medida que se incorporan mayor cantidad de datos, elevando la precisión de los análisis y con ello la fiabilidad de los resultados obtenidos.

Conclusiones

En base a los resultados obtenidos se arribó a las siguientes conclusiones:

1. Al ejecutar los pasos del procesamiento del lenguaje natural se transformó a un lenguaje binario los comentarios seleccionados del sitio web de noticias Cubadebate. A su vez, el algoritmo clasificador de análisis de sentimientos permitió, clasificar el artículo en positivo, negativo o neutro y el modelo de aprendizaje automático y la máquina de soporte vectorial posibilitó el aprendizaje automático para futuras fuentes de datos.

2. Se clasificó automáticamente la polaridad a través de los comentarios de 5 artículos seleccionados en el sitio web de noticias Cubadebate, clasificando el artículo 1 como positivo y el artículo 2, 3, 4 y 5 como negativo.

3. Después de aplicar las métricas Accuracy-score y Precision-Recall se concluye que el algoritmo escogido (máquina de soporte vectorial) aunque era uno de los mejores para categorización de texto según [19, 18, 20, 21, 22, 23, 24], resultó ser el de menores valores en la validación.

Referencias bibliográficas

[1] T. K. Tran and T. T. Phan, "Mining opinion targets and opinion words from online reviews," *International Journal of Information Technology*, vol. 9, no. 3, pp. 239–249, 2017.

[2] B. Liu, "Sentiment analysis and opinion mining," *Synthesis lectures on human language technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[3] S. Raschka, *Python Machine Learning*. Packt Publishing Ltd, 2015.

[4] I. Smeureanu, C. Bucur, *et al.*, "Applying supervised opinion mining techniques on online user reviews," *Informatica Economică*, vol. 16, no. 2, pp. 81–91, 2012.

[5] S. J. Valbuena, S. A. Cardona, and A. Fernández, "Minería de datos sobre streams de redes sociales, una herramienta al servicio de la bibliotecología," *Información, cultura y sociedad*, no. 33, pp. 63–74, 2015.

[6] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: tasks, approaches and applications," *Knowledge-Based Systems*, vol. 89, pp. 14–46, 2015.

[7] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams engineering journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

[8] F. M. Gonzalez-Longatt, "Introducción a los sistemas de información: fundamentos," 2007.

[9] M. A. H. B. Puja Munjal, Aditi Gupta and S. Kumar, *Social Media Based Opinion Mining Using Lexical Sentiment Analysis*, Junio 2017.

[10] J. A. Balazs and J. D. Velásquez, "Opinion mining and information fusion: a survey," *Information Fusion*, vol. 27, pp. 95–110, 2016.

[11] C. C. Aggarwal and C. Zhai, *Mining text data*. Springer Science & Business Media, 2012.

[12] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[13] S. N. Manke and N. Shivale, "A review on: opinion mining and sentiment analysis based on natural language processing," *International Journal of Computer Applications*, vol. 109, no. 4, 2015.

[14] T. S. E. Raheesa Safrin, K.R.Sharmila, "Sentiment analysis on online product review,"

[15] P. Arya, A. mit Bhagat, and B. MANIT, "Deep survey on sentiment analysis and opinion mining on social networking sites and e-commerce website," *International Journal of Engineering Science*, vol. 4796, 2017.

[16] Y. Zheng, "Opinion mining from news articles," in *Recent Developments in Intelligent Computing, Communication and Devices*, pp. 447–453, Springer, 2019.

[17] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.

[18] A. Basu, C. Walters, and M. Shepherd, "Support vector machines for text categorization," in *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*, pp. 7–pp, IEEE, 2003.

[19] A. S. Manek, P. D. Shenoy, M. C. Mohan, and K. Venugopal, "Aspect term extraction for sentiment analysis in large movie reviews using gini index feature selection method and svm classifier," *World wide web*, vol. 20, no. 2, pp. 135–154, 2017.

[20] J. Lilleberg, Y. Zhu, and Y. Zhang, "Support vector machines and word2vec for text classification with semantic features," in *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, pp. 136–140, IEEE, 2015.

[21] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European conference on machine learning*, pp. 137–142, Springer, 1998.

[22] A. H. Mohammad, T. Alwada'n, and O. Al-Momani, "Arabic text categorization using support vector machine, naïve bayes and neural network," *GSTF Journal on Computing (JoC)*, vol. 5, no. 1, 2018.

[23] S. Sun, C. Luo, and J. Chen, "A review of natural language processing techniques for opinion mining systems," *Information fusion*, vol. 36, pp. 10–25, 2017.

[24] M. Lan, C.-L. Tan, H.-B. Low, and S.-Y. Sung, "A comprehensive comparative study on term weighting schemes for text categorization with support vector machines," in *Special interest tracks and posters of the 14th international conference on World Wide Web*, pp. 1032–1033, ACM, 2005.

[25] G. A. Betancourt, "Las máquinas de soporte vectorial (s)," *Scientia et technica*, vol. 1, no. 27, 2005.

[26] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," *Expert Systems with Applications*, vol. 57, pp. 117–126, 2016.

[27] M. Jaramillo and M. de Jesús, "Modelo teórico-metodológico basado en el kdd para la integración y explotación de datos bibliográficos de patentes," 2014.

[28] U. Fayyad, G. Piatetky-Shapiro, and P. Smyth, "The kdd process for extracting useful knowledge from volumes of data," *Communications of the ACM*, vol. 39, no. 11, pp. 27–34, 1996.

[29] M. P. Villegas, M. J. Garciarena Ucelay, J. P. Fernández, M. A. Álvarez Carmona, M. L. Errecalde, and L. Cagnina, "Vector-based word representations for sentiment analysis: a comparative study," in *XXII Congreso Argentino de Ciencias de la Computación (CACIC).*, 2016.

[30] A. A. H. Corey, *Desarrollo de un sistema de clasificación binaria automática de noticias con máquinas de aprendizaje*. PhD thesis, Pontificia Universidad Católica de Valparaiso, 2010.

[31] A. Nematzadeh, S. C. Meylan, and T. L. Griffiths, "Evaluating vector-space models of word representation, or, the unreasonable effectiveness of counting words near other words.," in *CogSci*, 2017.

[32] C. Torres López and L. Arco García, "Representación textual en espacios vectoriales semánticos," *Revista Cubana de Ciencias Informáticas*, vol. 10, no. 2, pp. 148–180, 2016.

[33] A. Esuli and F. Sebastiani, "Sentiwordnet: a high-coverage lexical resource for opinion mining," *Evaluation*, vol. 17, no. 1, p. 26, 2007.

[34] E. Cambria and A. Hussain, "Sentic computing," *Springer*, 2012.

[35] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining.," in *LREc*, vol. 10, pp. 1320–1326, 2010.

[36] M. Amores, L. Arco, and M. Artiles, "Posneg opinion: Una herramienta para gestionar comentarios de la web," *Revista Cubana de Ciencias Informáticas*, vol. 9, no. 1, pp. 20–12, 2015.

[37] W. McKinney, *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. .ºReilly Media, Inc.", 2012.

[38] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. .ºReilly Media, Inc.", 2009.

[39] N. Hardeniya, J. Perkins, D. Chopra, N. Joshi, and I. Mathur, *Natural Language Processing: Python and NLTK*. Packt Publishing Ltd, 2016.

[40] B. Srinivasa-Desikan, *Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras*. Packt Publishing Ltd, 2018.

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[42] W. McKinney, "pandas: a python data analysis library," *see http://pandas. pydata. org*, 2015.

[43] M. Lenz, "Plotting using inline:: Python and matplotlib," in *Perl 6 Fundamentals*, pp. 119–134, Springer, 2017.

[44] A. Yim, C. Chung, and A. Yu, *Matplotlib for Python Developers: Effective techniques for data visualization with Python*. Packt Publishing Ltd, 2018.

[45] Q. N. Islam, *Mastering PyCharm*. Packt Publishing Ltd, 2015.

[46] F. Mora and A. Borbón, "Edición de textos científicos latex2014," *Revista digital Matemática. Educación e Internet*, 2013.

[47] N. Madnani, "Getting started on natural language processing with python.," *ACM Crossroads*, vol. 13, no. 4, p. 5, 2007.

[48] M. Khabsa, A. Elmagarmid, I. Ilyas, H. Hammady, and M. Ouzzani, "Learning to identify relevant studies for systematic reviews using random forest and external information," *Machine Learning*, vol. 102, no. 3, pp. 465–482, 2016.

Capítulo 4

Anexos

```
Predición de SVM: ['felicidades' 'felicidades' 'felicidades' 'felicidades
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
 'felicidades' 'felicidades' 'felicidades' 'felicidades' 'felicidades'
```

Figura 4.1: Predicción de dataset 1 [elaboración propia]

```
Predición de SVM: ['no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
```

Figura 4.2: Predicción de dataset 2 [elaboración propia]

```
Predición de SVM: ['no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' '
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
```

Figura 4.3: Predicción de dataset 3 [elaboración propia]

```
Predición de SVM: ['no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
```

Figura 4.4: Predicción de dataset 4 [elaboración propia]

```
Predición de SVM: ['no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
```

Figura 4.5: Predicción de dataset 5 [elaboración propia]