

**Comparación entre algoritmos de Machine Learning en la implementación del análisis
de sentimientos para textos en español**

**Proyecto de grado para optar por el título de
Ingeniero en Sistemas**

**UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
FACULTAD DE INGENIERÍA
INGENIERÍA EN SISTEMAS
BOGOTÁ D.C.
2021**

**Comparación entre algoritmos de Machine Learning en la implementación del
análisis de sentimientos para textos en español**

**JOAQUIN SUAREZ MOSQUERA
20112020016**

**Proyecto de grado para optar por el título de
Ingeniero en Sistemas**

**Directora
LUZ DEICY ALVARADO PhD.**

**UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
FACULTAD DE INGENIERÍA
INGENIERÍA EN SISTEMAS
BOGOTÁ D.C.
2021**

Tabla de contenido

Contenido

1	Introducción.....	7
2	Justificación	9
3	Planteamiento del problema	12
4	Objetivos.....	13
4.1	General.....	13
4.2	Específicos:	13
5	Marco Referencial	14
5.1	Antecedentes	14
5.2	Marco teórico.....	17
5.2.1	Lengua y lenguaje	17
5.2.2	Procesamiento de lenguaje natural (PNL)	18
5.2.3	Análisis de sentimientos	21
5.2.4	Preprocesamiento de textos.....	23
5.2.5	Procesamiento de Textos.	25
5.2.6	Métricas de Rendimiento	31
5.2.7	TWITTER	32
5.2.8	SCIKIT LEARN	33
5.2.9	CONDA	33
5.2.10	Tweepy.....	34
5.2.11	PATTERN.....	37
5.2.12	Coronavirus.....	38
6	Desarrollo de la Propuesta.....	40
6.1	Tratamiento de textos	40
6.1.1	Acceso a los textos.....	40
6.1.2	Etiquetado de los Tweets	41
6.1.3	Convenciones usadas	42
6.1.4	Limpieza de los textos	42
6.1.5	Eliminación de signos de puntuación	43
6.1.6	Eliminación de palabras vacías (stopwords).....	43
6.1.7	Lematizar las palabras.....	44
6.1.8	Eliminación de urls	45
6.1.9	Casos especiales.....	45

6.2	Análisis de los textos	47
6.2.1	Representación matricial.....	47
6.2.2	Entrenamiento de algoritmos	52
6.2.3	Cálculo de métricas.....	54
6.3	Desarrollo del aplicativo.....	55
6.3.1	Capa de presentación o cliente web	56
6.3.2	Capa de Procesamiento	61
6.3.3	Capa de persistencia.....	63
7	Pruebas realizadas.....	64
8	Resultados Obtenidos	66
8.1	Resultados para etiquetado manual	66
8.2	Resultados para etiquetado automático	67
9	Análisis de Resultados.....	69
10	Conclusiones.....	70
11	Trabajo Futuro	71

Lista de Figuras

Figura 1 Países con mayor número de publicaciones en análisis de sentimientos	9
Figura 2 Enfoques análisis de sentimientos.....	22
Figura 3 Máquina de soporte vectorial en dos dimensiones	27
Figura 4 Neurona Artificial.....	29
Figura 5 Capas en Redes Neuronales.....	30
Figura 6 Configuración de objetos OAuthHandler y Api de tweepy	34
Figura 7 Llamado al Cursor de Tweepy.....	35
Figura 8 Función para tratar ítems del cursor.....	37
Figura 9 Guardado de tweets en texto plano	40
Figura 10 Eliminación de signos de puntuación.....	43
Figura 11 Función para eliminar las palabras vacías	44
Figura 12 Modulo pattern para español	44
Figura 13 Preprocesamiento de Tweets.....	46
Figura 14 Perceptrón multicapa de Scikitlearn.....	52
Figura 15 Algoritmo Bayesiano Multinomial	53
Figura 16 Entrenamiento SVM con variante “One vs One”.....	54
Figura 17 Cálculo de métricas	55
Figura 18 Arquitectura general del sistema	56
Figura 19 Opción de carga de Archivos	57
Figura 20 Vista de textos preprocesados	57
Figura 21 Configuración de vocabulario	58
Figura 22 Datos de entrenamiento obtenidos	59
Figura 23 Formulario para entrenar algoritmos.....	59
Figura 24 Panel de pruebas.....	60
Figura 25 Métricas obtenidas de los textos	61
Figura 26 Módulos servidor.....	62

Lista de tablas

Tabla 1 Información morfológica y semántica de un lecionario.....	21
Tabla 2 Representación matricial de textos.....	24
Tabla 3 Matriz de confusión para dos clases	31
Tabla 4 Parámetros cursor Tweepy	35
Tabla 5 Campos Objeto tweet de Tweepy.....	36
Tabla 6 Convención para consolidado de etiquetas de tweets	41
Tabla 7 Etiquetas enfoque automático	41
Tabla 8 Etiquetas de tweets	42
Tabla 9 Convención para dos etiquetas	42
Tabla 10 Frecuencias de palabras en muestra de Tweets.....	48
Tabla 11 Matriz frecuencia termino-Tweet	49
Tabla 12 Factor tf-idf Tweets de prueba.....	51
Tabla 13 Campos para colección del Algoritmo	63
Tabla 14 Escenarios de prueba con 3 Etiquetas.....	64
Tabla 15 Escenarios de prueba con 2 Etiquetas.....	64
Tabla 16 Resultados 8000 Tweets.....	66
Tabla 17 Resultados 6000 Tweets.....	66
Tabla 18 Resultados 4000 Tweets.....	67
Tabla 19 Resultados 2000 Tweets.....	67
Tabla 20 Resultados 12000 Tweets.....	67
Tabla 21 Resultados 9000 Tweets.....	68
Tabla 22 Resultados 6000 tweets 2 etiquetas.....	68
Tabla 23 Resultados 3000 Tweets.....	68

1 Introducción

Desde el surgimiento de la web 2.0, la interacción de millones de personas con plataformas como redes sociales o blogs ha generado una cantidad enorme de datos en internet, que se evidencia en el contenido alojado en sitios web producto de los comentarios y opiniones expresadas por sus usuarios, creando una fuente aprovechable de información.

Paralelamente a este fenómeno social, la comunidad científica y otras áreas de conocimiento han buscado formas de estructurar, procesar y clasificar todo el contenido que se encuentra en las redes sociales o en algunos sitios web, para entender cuáles son los gustos, deseos o sentimientos de las personas, mientras que otros sectores como el político, empresarial o social, con base en la información extraída, tienen la posibilidad de orientar decisiones (estratégicas, operativa,) o acciones a realizar. En el caso político, el expresidente de Estados Unidos Barack Obama basó su campaña en el análisis de redes sociales para construir una propuesta que incluía intereses de distintos sectores del pueblo norteamericano (Zenith, 2013). Por otro lado, en el ámbito empresarial, *Microsoft* anunció en el año 2019 que eliminaría de sus programas la herramienta de dibujo *Paint*, dado que el remplazo de este daría paso a nuevos aplicativos con más opciones, provocó una reacción negativa por parte de millones de personas a nivel mundial que a través de redes sociales expresaban su descontento, haciendo que finalmente *Microsoft* decidiera conservar este programa (Barredo A., 2019).

Dado que uno de los tipos de fuentes de datos más representativo para los humanos, alojado en Internet es el que viene en formato de texto, se ha usado en el análisis de sentimientos para procesar y clasificar grandes cantidades en distintas áreas como política, marketing empresarial y campañas presidenciales (Hootsuite, 2019). El análisis de sentimientos o minería de opinión se encarga de tratar elementos del lenguaje natural, ya sean palabras o frases, para extraer información subjetiva y asignarles un sentimiento o polaridad. Aunque inicialmente esta tarea se enfocaba en hacer un tratamiento lingüístico sobre los textos y su estructura interna, actualmente se creó la vertiente que aborda el problema como uno de clasificación. Con el *Machine Learning* este proceso puede hacerse mediante el aprendizaje supervisado, que, a partir de la representación vectorial de un conjunto de textos con su respectiva etiqueta o clase, entrena un modelo y lo usa para clasificar nuevos ejemplos de los que se desconoce su categoría.

En este trabajo se recolectaron textos en español de la red social Twitter acerca de la covid-19 durante un periodo aproximado de tres meses, después se usaron dos enfoques de etiquetado para los textos: uno manual y el otro automático. En el primero se solicitó a varias personas

llevar a cabo este proceso sobre un mismo lote de Tweets asignándoles sentimientos de “positivo”, “negativo” o “neutro”, donde los resultados se compararon y se usaron para generar una muestra final de textos con etiqueta. Para el enfoque automático se obtuvieron las palabras con mayor frecuencia de aparición y se les dio un valor numérico de -1 si el sentimiento que expresaba era “negativo” o 1 en caso de ser “positivo”, posteriormente la polaridad de cada Tweet se calculó con reglas de validación y la sumatoria de sentimientos asociados a las palabras. Posteriormente se usaron técnicas de limpieza en los textos para eliminar el ruido o los datos que no eran relevantes como los signos de puntuación, urls y *stopwords*. Con el uso de herramientas computacionales y librerías se cambiaron las palabras (en la medida posible) a su raíz léxica. Después, para transformar los Tweets en vectores de entrenamiento de los algoritmos de *machine learning*, se armó una matriz a partir del vocabulario de las palabras y su frecuencia de aparición en todos los textos. Finalmente se entrenaron los modelos de Redes Neuronales, Maquinas de Soporte Vectorial y Naïve Bayes, con los que se llevaron a cabo diferentes pruebas, obteniendo sus respectivas métricas de rendimiento.

2 Justificación

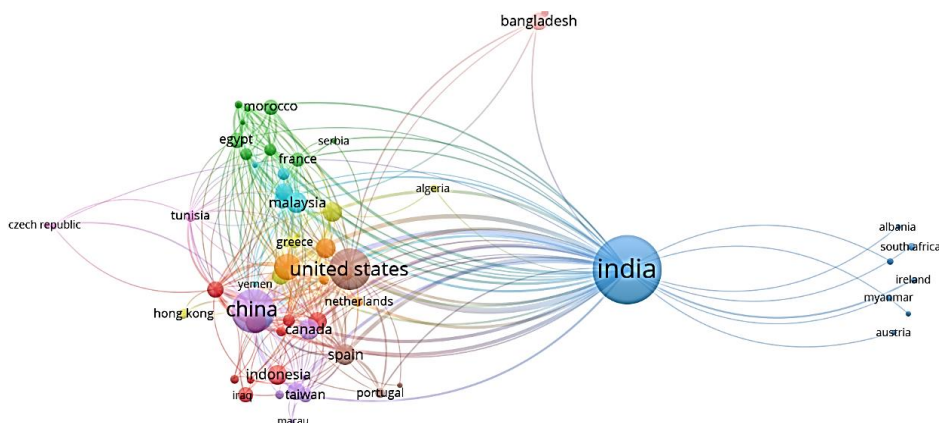
Las cifras a nivel mundial sobre el uso de las redes sociales tienden a crecer con el paso de los días. De acuerdo con la información publicada por Hootsuite para el año 2020, había 3,8 mil millones de personas registradas en redes sociales, con un incremento del 9.2% comparado con el año anterior (Estadísticas generales de redes sociales, 2020).

Por otra parte, las empresas también reconocen la importancia de sus campañas en redes sociales. Según la entrevista de *Social Media Trends*, para el año 2018 el porcentaje de personas en medios que pueden ser alcanzadas para ofrecer algún tipo de producto es de un 28%, así mismo en la entrevista se afirma que el 41% de los usuarios de internet a nivel mundial usan las redes sociales para encontrar alguna marca o producto en vez de los buscadores tradicionales y que la información allí generada es útil para las organizaciones.

En algunos sectores como el político y académico se ha despertado el interés de conocer y entender el comportamiento de las personas en estos medios digitales. Las elecciones presidenciales de Estados Unidos de los años 2008 y 2012 que basaron sus campañas electorales en la difusión de mensajes en redes sociales y el análisis de los datos allí generados, demostraron el poder que tienen estas últimas, ya que para llegar a la casa blanca como primer mandatario es necesario superar los 270 votos electorales de 538 posibles y Barak Obama en el año 2008 obtuvo 365, en contraste de John McCain que logró un total 173, para el año 2016 Obama consiguió la reelección con 332 votos electorales sobrepasando a Mitt Romney que alcanzó los 206 (Zenith, 2013).

Figura 1

Países con mayor número de publicaciones en análisis de sentimientos



Nota. La dimensión de los círculos representa la magnitud de trabajos realizados y las líneas que los conectan las referencias hechas entre distintos países. Recuperado de Scopus, <https://www.scopus.com/home.uri>.

Como se observa en la Figura 1, países como India, China y Estados Unidos con sus idiomas hindi, mandarín e inglés respectivamente, han hecho aportes significativos en la investigación académica del análisis de sentimientos, también de cómo los trabajos realizados en un idioma pueden servir de referencia para aplicarlos en otro completamente distinto, por ejemplo, España ha tomado referencias de Portugal (portugués) y Canadá (inglés y francés). En esa misma dirección, el presente trabajo busca aplicar la técnica de análisis de sentimientos sobre textos en español de temáticas sociales y políticas acerca del coronavirus y la pandemia de la COVID-19, extraídos de la red social Twitter, usando metodologías que ya se han implementado para otros países o idiomas.

Al construir mecanismos que permitan clasificar textos de manera automática, se reduce considerablemente el tiempo invertido en comparación con un ser humano que realiza esta tarea manualmente, dado que las personas tienen un promedio de lectura de 200 palabras por minuto según lo indica Dhanendran (Dhanendran A, 2014), mientras que las computadoras pueden procesar una cantidad mayor al trabajar en el orden de los milisegundos, y con los avances tecnológicos se pueden emplear para llevar a cabo tareas en el procesamiento del Lenguaje Natural, donde los esfuerzos se han centrado principalmente en mejorar e innovar las técnicas con las cuales trabaja esta área.

Como lo indica Darina Lynkova (Lynkova, D., 2021), los algoritmos de *machine learning* *Naive Bayes* y Maquinas de Soporte Vectorial, hacen parte del top 10 de los modelos de aprendizaje supervisado más usados en la actualidad, por otro lado, en el trabajo de Celina Beltran (Beltrán C. & Barbona I., 2017), se evidencia que las redes neuronales artificiales con su variante de perceptrón multicapa, obtuvieron mejores resultados en la clasificación de textos comparados con algoritmos como el Vecino más Cercano, Maquinas de Soporte Vectorial, Árboles de Clasificación, Regresión Logística, entre otros. Por esta razón en el presente trabajo se usarán los algoritmos *Naive Bayes*, Maquinas de Soporte Vectorial y Redes Neuronales para comparar su desempeño en la clasificación de Tweets relacionados a la COVID-19, empleando diferentes configuraciones y variaciones en la cantidad de datos de entrenamiento.

Por otra parte, la pandemia COVID-19 es un tema de interés mundial, con repercusión en todas las esferas de la sociedad, ya que millones de personas murieron, empresas quebraron, varios hospitales colapsaron en la atención de pacientes, el PIB de muchos países bajó por las restricciones en actividades económicas dentro y fuera de sus fronteras, adicionalmente en redes sociales se ha divulgado información de casos reportados, decesos y el avance en proceso de vacunación, también muchas personas expresan sus sentimientos u opiniones acerca de esta temática. Por lo tanto, si se desea tratar grandes cantidades de texto relacionados con la

COVID-19, es necesario construir sistemas automatizados, en los que se empleen algoritmos de *machine learning* y dentro de éstos validar cuál es más eficiente a la hora de tratar frases u opiniones, todo esto mediante la aplicación del análisis de sentimientos, con el fin de hacer seguimiento a las tendencias, o posturas, acerca de esta pandemia, expresadas por las personas en Twitter.

3 Planteamiento del problema

Hoy en día la Internet es una tecnología esencial de la sociedad; tanto personas como empresas usan sitios web y redes sociales para llevar a cabo promociones de marcas, lanzamiento de nuevos productos o servicios, publicación de eventos etc. Además, buscan un acercamiento a públicos de interés o nichos de mercado para mejorar imagen o reputación que los lleve a lograr sus objetivos.

En la web 2.0, con el surgimiento de redes sociales y sitios web dinámicos se ha generado una gran cantidad de datos por parte de millones de usuarios, sectores como el político, académico y empresarial se han enfocado en cómo aprovechar este contenido, para analizar el comportamiento social en Internet con el fin de obtener información que les pueda ser de utilidad, y así usarla de acorde con sus intereses en diferentes contextos, ya sean políticos, ambientales o sociales.

Hoy en día gracias a los avances tecnológicos y científicos se han desarrollado mecanismos y herramientas como *Brandwatch*, *Semantria* y *Rapidminer*, entre otras, que permiten el tratamiento de textos en diferentes idiomas, siendo el inglés uno de los más trabajados (GetApp, s.f.). Donde el trabajo se centra en definir cómo las máquinas tratan las palabras, frases y oraciones para obtener información relevante para los seres humanos.

El presente proyecto aplicará el análisis de sentimientos a textos recolectados en español de la red social Twitter, que tengan relación con la COVID-19 como temática central, usando técnicas de limpieza de textos y algoritmos de *Machine Learning*, para buscar una forma de procesarlos y construir sistemas de clasificación que realicen esta tarea de manera automática obteniendo resultados similares a como lo haría una persona.

4 Objetivos

4.1 General

Comparar el comportamiento de los algoritmos de *Machine Learning*: *Naive Bayes*, Redes Neuronales y Maquinas de soporte Vectorial en la aplicación del análisis de sentimientos a textos en español obtenidos de la red social Twitter, sobre temas relacionados con la COVID 19.

4.2 Específicos:

- Usar la Librería Tweepy de Python para descargar diariamente un estimado de 100 a 300 Tweets, dadas las restricciones del uso gratuito del API.
- Obtener los textos preprocesados a partir de la muestra inicial.
- Construir la matriz de entrada para los algoritmos de *machine learning* mediante la representación tf-idf o mediante el conteo de términos.
- Implementar diferentes versiones de los algoritmos de *machine learning Naive Bayes*, Maquinas de soporte Vectorial y Redes Neuronales, aplicándolos a las matrices de representación de textos.
- Aplicar métricas de rendimiento que permitan determinar cuál de los algoritmos de *machine learning* evidencia mayor efectividad en la clasificación de textos.

5 Marco Referencial

5.1 Antecedentes

Las elecciones presidenciales de Estados Unidos en el año 2008 y 2012 cambiaron la forma en la que se realizan las campañas con el fin de llegar a la casa blanca, donde el enfoque publicitario tradicional fue remplazado por uno basado en el análisis del contenido en redes sociales. El entonces candidato Barack Obama y su equipo enfocaron su esfuerzo en estar presentes en las principales plataformas como MySpace, Twitter y Facebook, adicionalmente crearon el sitio My.BarackObama.com, con la finalidad de acercarse más a los ciudadanos, organizar actividades, llevar a cabo discusiones y recolectar fondos, esto le ayudó a Barack Obama ser presidente en el año 2008.

Para las elecciones del año 2012 en Estados Unidos, con la masificación del uso del internet y el aumento de miles a millones de usuarios en redes sociales, Barack Obama con su equipo se centraron en el análisis de datos que los usuarios de internet generaban, así mismo crearon un grupo de trabajo que operaría durante la campaña y estaba constituido principalmente por programadores, matemáticos y expertos de internet. Gracias a esto, analizaron la información obtenida de los votantes, lograron identificar a quienes aún no se habían decidido por su voto y a su vez encontraron la forma de llegar a ellos. En Florida, y según las bases de datos consultadas, las mujeres jóvenes de 35 años tenían un gusto particular por algunas series de televisión, así usaron los espacios publicitarios para dar a conocer la campaña presidencial. Aunque su rival Romney también estaba realizando una campaña en redes sociales, Barack Obama ganó las elecciones presidenciales, con el análisis preciso de la información (Zennit,2013).

Por su parte, en el ámbito académico también se han llevado a cabo grandes esfuerzos encaminados a encontrar nuevas y mejores formas de analizar y estructurar la información obtenida de redes sociales. A continuación, se mencionarán algunos de los trabajos relacionados con el análisis de sentimientos.

Li Jie y Qui Lirong (Jie Li & Lirong Qiu, 2017) aplicaron el análisis de sentimientos a blogs sociales de Internet, que tenían comentarios u opiniones muy cortos y no excedían un total de 140 caracteres. Su trabajo tenía como eje central el análisis del contenido de los textos, estructura interna y relación entre palabras, con el fin de establecer su respectiva categoría o polaridad a través de un cálculo que ellos mismos plantearon. Entrenaron algoritmos usados

por *Machine Learning* y lexicon, obteniendo mejores resultados comparado con trabajos previos.

Andrea Salinca (Salinca A., 2016) se enfocó en reseñas de negocios en inglés usando un conjunto de metadatos proporcionados por Yelp Dataset Challenge (que contiene información de 61000 negocios, 366000 usuarios y un total de 1,6 millones de opiniones). Su trabajo se basó en dos etapas, en la primera se enfocó en extraer características de los textos basándose en los sacos de palabras¹ y observar sus frecuencias. Ya en la segunda etapa, usó 4 algoritmos de machine learning (naive Bayes, regresión logística, vectores lineales de soporte y un clasificador estocástico de gradiente descendente) para la clasificación de textos (que estaban en inglés), finalmente presenta una comparación del rendimiento de los 4 algoritmos usados.

Woldemariam (Woldemariam, Y., 2016) usó redes neuronales recursivas y algoritmos basados en lexicón con el fin de analizar los comentarios realizados en foros de discusión, en el caso del lexicón, se basó en un diccionario de sentimientos para asignar la polaridad a cada palabra, y la etiqueta que correspondía a esa frase o texto era la suma de todas las polaridades. Para el caso de las redes neuronales recursivas procesó los textos, etiquetando cada una de las palabras, mediante un análisis sintáctico.

Luis Rodríguez (Rodríguez L., 2019) implementó el análisis de sentimientos sobre Tweets relacionados al sistema de transporte masivo Transmilenio, aplicando técnicas de minería de datos con la finalidad de limpiar los textos, estructurarlos y extraer de estos la información necesaria para identificar en los mensajes contenido relacionado a comportamientos que afectarán el funcionamiento normal del sistema o la integridad ciudadana.

Paola Guarnizo y Andes Monroy (Monroy A. & Guarnizo P., 2020) usaron la técnica de análisis de sentimientos sobre Tweets relacionados a la temática de la Jurisdicción Especial para la Paz (JEP), donde aplicaron un proceso de obtención de textos, limpieza, lematización de palabras, extracción de características, con el fin de entrenar los algoritmos de *machine learning* *Random Forest*, *Naive Bayes* y Máquinas de Soporte Vectorial para clasificar los Tweets asignándoles una etiqueta, ya sea “positivo”, “negativo” o “neutro”, dividiendo la muestra inicial en conjuntos de entrenamiento y pruebas, aplicando métricas de rendimiento sobre estos últimos. Como se observa este trabajo usa solamente tres etiquetas y no utiliza redes neuronales, que es uno de los algoritmos de *machine learning* más usados a nivel mundial.

¹ Los sacos de palabras o del inglés BOW, hace referencia a una representación vectorial de palabras.

Daniel Ruiz y Yohana Velandia (Ruiz D., Delgado Y., 2017) construyeron un aplicativo para llevar a cabo la minería de opinión para textos obtenidos de Twitter en diferentes temáticas, en el que definieron una serie de reglas con el fin de preprocesar los textos y eliminar los datos que no eran necesarios. Con el lexicón “Galeras” realizaron una asignación de polaridad a los Tweets usando las etiquetas (positivo, negativo y neutro), teniendo presente los modificadores y(o) con cuantificadores para el sentimiento. Finalmente usaron diferentes estrategias para establecer el grado de favorabilidad de los Tweets mediante cálculos matemáticos.

Fernando Osorio y Alberto Arango (Osorio F, Arango A., 2021) llevaron a cabo en análisis de sentimiento mediante machine learning usando la covid como eje temático, donde emplearon el paquete informático de rapidminder con la finalidad de realizar las tareas de tokenización, limpieza y eliminación de stopwords sobre los textos descargados. Para generar los datos de entrenamiento, se basaron en un corpus con 6775 textos que contienen las etiquetas de: ira, alegría, miedo y tristeza. Por último, entrenaron los modelos de Naive Bayes, Naive bayes Kernel y Deep learning, con el fin de clasificar los Tweets.

Rafael Sanchez (Sanchez R., 2019), implementó el análisis de sentimientos en temáticas relacionadas con el turismo y política a diferentes cuentas de lugares representativos en España mediante un enfoque semántico y el uso de los dendogramas. El proceso de etiquetado de Tweets consistió en identificar las palabras con mayor frecuencia de aparición y asignarles un valor numérico de acuerdo con el grado de intensidad de la emoción representada, en un rango de -5 a 5 para indicar “muy negativo” y “muy positivo” respectivamente. Finalmente agrupó las palabras por frecuencia de aparición, cuenta y tipo de sentimiento para establecer la percepción de los usuarios respecto a estas.

Luciana Dubiau y Juan Ale (Dubiau L., Ale J, s.f) aplicaron la técnica de análisis de sentimientos con un enfoque basado en *machine learning*, para textos recolectados de un sitio web en español en el cual los usuarios dejaban sus opiniones acerca de diversos restaurantes, comentando acerca del servicio, ambiente y comida. Además, daban una valoración de (malo, regular, bueno y excelente) asociada a un nivel de puntuación. Con este insumo los autores categorizaron los textos como positivo y negativo, basándose en el puntaje asignado al restaurante. Finalmente usaron los algoritmos de Naive Bayes, Árboles de decisión, Máquinas de soporte Vectorial y adaptación del clasificador de Turney para entrenamiento y clasificación de los textos de los restaurantes.

5.2 Marco teórico

El ser humano es considerado como el único ser vivo con la capacidad de emplear la comunicación con fines heurísticos, que además de recibir genéticamente las pautas de comportamiento para su supervivencia, aprende de experiencias y así puede actuar de diferentes maneras frente a diversas situaciones. También le ha permitido intercambiar información con semejantes con el fin de mejorar muchas de sus actividades y lograr avances significativos. Este proceso lo ha logrado gracias al desarrollo de la lengua y el lenguaje (Chordá, F., 2000).

5.2.1 Lengua y lenguaje

La lengua es conocida como un conjunto de signos orales y escritos que permiten llevar a cabo la comunicación entre las personas, dentro de una comunidad lingüística. En cambio, el lenguaje es la facultad que poseen los seres humanos para aprender nuevas lenguas, ya sean orales o escritos (Aitchinson, 1992). Existen varios tipos de lenguajes y algunos de ellos son:

- **Lengua Natural:** Es aquel que se usa cotidianamente y ha construido un conjunto de signos a través del tiempo con la intención de comunicar a las personas. Una característica relevante, es que permite arbitrariedad en su uso, es decir que no impone restricciones en la forma en la que se emplea, y a su vez hace que la gramática de este tipo lenguaje se vaya modificando constantemente, dependiendo del contexto o el lugar donde se desenvuelven sus habitantes (Vásquez & Huerta, 2009).
- **Lenguaje Formal:** Es un tipo de lenguaje que ha sido desarrollado por el ser humano para expresar situaciones del ámbito científico. Un lenguaje formal es aquel que está perfectamente formado en el sentido que se eliminan las ambigüedades semánticas, este tipo de lenguaje se usa para modelar teorías en el ámbito de las matemáticas o la física. También está constituido por un alfabeto, un conjunto de reglas, y axiomas (Vásquez & Huerta, 2009).

EL origen de la lengua y el lenguaje es un enigma que aún no se ha podido esclarecer completamente, según aproximaciones realizadas, en un lapso de 50.000 a 100.000 años atrás apareció algún tipo de lenguaje hablado, y según las evidencias encontradas, hace 5000 años se desarrollaron las primeras formas escritas, mostrando el proceso de evolución y transformación continua de este mismo (Yule G., 2007).

5.2.2 Procesamiento de lenguaje natural (PNL)

El uso correcto del lenguaje en los seres humanos está estrechamente relacionado con la capacidad cognitiva y se asocia con su inteligencia, por lo que desarrollar una máquina que pueda entender y responder correctamente a una persona mediante un dialogo común se consideraría un acto “inteligente” (Kumar E. ,2011).

Alan Turing fue el primero en plantear en su famoso “*Test de Turing*” el intercambio de textos en lenguaje natural (según el postulado inicial) de una persona con una máquina, mientras son evaluados por un interlocutor que observa dicha interacción, si no indica acertadamente cuál de estos es el ordenador, se puede decir que la prueba fue superada y que la maquina tiene cierto grado de inteligencia (Copeland B,200). A pesar de que la computación estaba aún en sus inicios, este planteamiento fue importante para que áreas como lingüística, ciencias de la computación y estadística abordaran esta temática y realizaran grandes avances.

Con el auge de la computación e inteligencia artificial, se llevaron a cabo los primeros esfuerzos de la comunidad científica en plantear y (o) construir elementos computacionales que trataran el lenguaje natural. *Shanon* a partir de procesos discretos en las cadenas de Markov, aplicó modelos probabilísticos para automatizar el procesamiento del lenguaje, Chomsky uso las máquinas de estado para modelar una gramática como un lenguaje de estados, *Warren Weaver* en 1945 planteó una máquina traductora, donde un equipo de investigación de Georgetown llevaría a cabo su primera implementación haciendo traducciones del idioma inglés al ruso y viceversa (Hutchins et al, 1955). Posteriormente se abordó el tratamiento de los lenguajes formales como una secuencia de símbolos mediante el uso del álgebra y teoría de conjuntos, incluyendo a las gramáticas libres de contexto que dieron paso a los lenguajes de programación. Otros trabajos como sistemas simples de preguntas y respuestas bajo dominios específicos usando coincidencia de patrones y búsqueda de palabras claves, se desarrollaron poco después de la creación del procesamiento de Lenguaje natural como área de la inteligencia artificial en el workshop de 1956 (Moor J.,2006). Posteriormente Roger Shank y sus colegas trabajaron en programas para el entendimiento del lenguaje enfocándose en conocimiento conceptual humano, como scripts, textos y organización de la memoria. También otros sistemas lógicos como LUNAR se crearon con el fin de entender el lenguaje natural mediante la lógica de predicados (Woods W et al,1972).

El continuo desarrollo tecnológico brindó a las computadoras una mayor capacidad de cómputo, un hecho que representó una mejora significativa en las aplicaciones implementadas para el tratamiento el lenguaje natural. Además, el uso de datos y probabilidades se

estandarizaron en los algoritmos creados para etiquetar partes de voz, análisis sintáctico, resolución de referencias y procesamiento de discurso. Por lo que, a través de la historia, con todos los aportes de conocimiento en el procesamiento del lenguaje natural, el ser humano científica y tecnológicamente ha:

“combinado las tecnologías de la ciencia computacional como son: la inteligencia artificial, el aprendizaje automático o la inferencia estadística, y la lingüística aplicada, para hacer posible la comprensión y comunicación con computadoras mediante el lenguaje natural, con el fin de hacer determinadas tareas como la traducción automática, los sistemas de diálogo interactivos, el análisis de opiniones entre otros” (Futurizable, 2017).

Esta área de conocimiento tiene diversas aplicaciones, Hernández y Gómez (Hernández M., Gómez J, 2013) mencionan algunas de estas y son:

- ***Recuperación y extracción de información:*** Se encarga de obtener información relevante en un repositorio de textos no estructurados, aplicando un proceso de transformación que puede ser de tipo matemático o basado en sus propiedades. La finalidad de esta tarea es obtener una fuente de datos estructurada con las secciones que interesan de los textos.
- ***Minería de datos:*** Tiene como fin encontrar posibles relaciones o patrones ocultos dentro de los textos, por lo que es necesario que los datos se encuentren estructurados y esto se puede lograr mediante un proceso de normalización.
- ***Traducción automática:*** Su objetivo es tomar textos que se encuentren en un lenguaje y convertirlos a otro conservando su significado. Para esto realiza una representación intermedia de los textos, después de acuerdo con las reglas gramaticales del Lenguaje destino, se modifica dicha representación y finalmente los textos se convierten al idioma destino.
- ***Sistemas de búsquedas de respuestas:*** Estos se aplican principalmente en los motores de búsqueda y su finalidad es brindar respuestas acertadas a búsquedas solicitadas por usuarios. En esta aplicación se usan la extracción y recuperación de información y el mayor reto se presenta en la respuesta generada, por lo que es necesario calificar los resultados obtenidos.
- ***Generación de resúmenes automáticos:*** Se pueden dar en dos niveles: documento y documentos y también de tipo abstractivo o extractivo. En el primero se basa en una colección de textos párrafos, frases o términos que estén asociados con el significado

del texto original, en el abstractivo se basa en técnicas de parafraseo para producir síntesis y aún se encuentran en desarrollo.

Moreno (Moreno A., s.f.) menciona que un sistema de procesamiento de lenguaje natural tiene algunos componentes que constituyen la base de su funcionamiento y pueden estar o no presentes dependiendo de su aplicación. Dichos componentes son:

- **Fonológico:** Es el encargado de estudiar los sonidos producidos por los seres humanos a la hora de hablar, la relación de las palabras con el significado que representan y de cómo este proceso hace posible la comunicación.
- **Morfológico:** Se encarga de analizar la estructura interna de las palabras, extrayendo toda la información necesaria de estas: morfemas, lemas, rasgos flexivos y unidades léxicas, que se pueden ser usar en otros niveles de procesamiento.
- **Sintáctico:** Es el tratamiento de las oraciones de acuerdo con un modelo gramatical lógico o sintáctico. Donde las palabras y un conjunto de reglas se combinan para analizar y(o) generar posibles elementos que sean aceptados dentro de un lenguaje ya sean frases, oraciones, discursos entre otros.
- **Semántico:** Con el significado de las palabras se encarga de dar interpretación a las oraciones.
- **Pragmático:** Analiza las oraciones en diferentes contextos y la repercusión de su significado en ellos.

Como lo menciona Rafael Marín (Marín R., s.f.), en las diferentes aplicaciones del procesamiento de lenguaje natural se emplea un recurso muy importante conocido como el **lexicón computacional**, este es un diccionario que contiene información codificada y estructurada de los lemas bajo un criterio morfológico en un formato de fácil acceso por los computadores, que incluye su clasificación gramatical y posibles formas flexivas (masculino singular). En la Tabla 1 se puede ver el ejemplo de una estructura léxica para los lemas *gato*, *sobre* y *gato de agua* con información tanto gramatical como semántica asociada a estos.

Tabla 1

Información morfológica y semántica de un leuario

Información morfológica		Información semántica
Formario	Lemario	Acepcionario
sobre-Prep.	sobre-Prep.	sobre-1 (<<Encima de.>>) sobre-2 (<<Además de.>>)
sobre-Nom Masc. Sing.	sobre-Nom.	sobre-1 (<<Envoltorio...>>)
sobres-Nom Masc. Plur.		sobre-2 (<<Juego...>>)
gata-Nom. Fem. Sing.	gato-Nom	gato-1 (<<Mamífero>>)
gata-Nom. Fem. Plur.		gato-2 (<<Bolso...>>)
gato-Nom. Masc. Sing.		gato-3 (<<Dinero...>>)
gatos-Nom. Masc. Plur.		gato-4 (<<Instrumento...>>)
gato de agua-Nom. Masc. Sing.	gato de agua-Nom.	gato de agua (<<Ratonera ..>>)
gatos de agua-Nom. Masc. Plur.		

Nota. Tomada de *El tratamiento computacional del léxico y sus aplicaciones* (p. 466), por Marin R., s.f.

5.2.3 Análisis de sentimientos

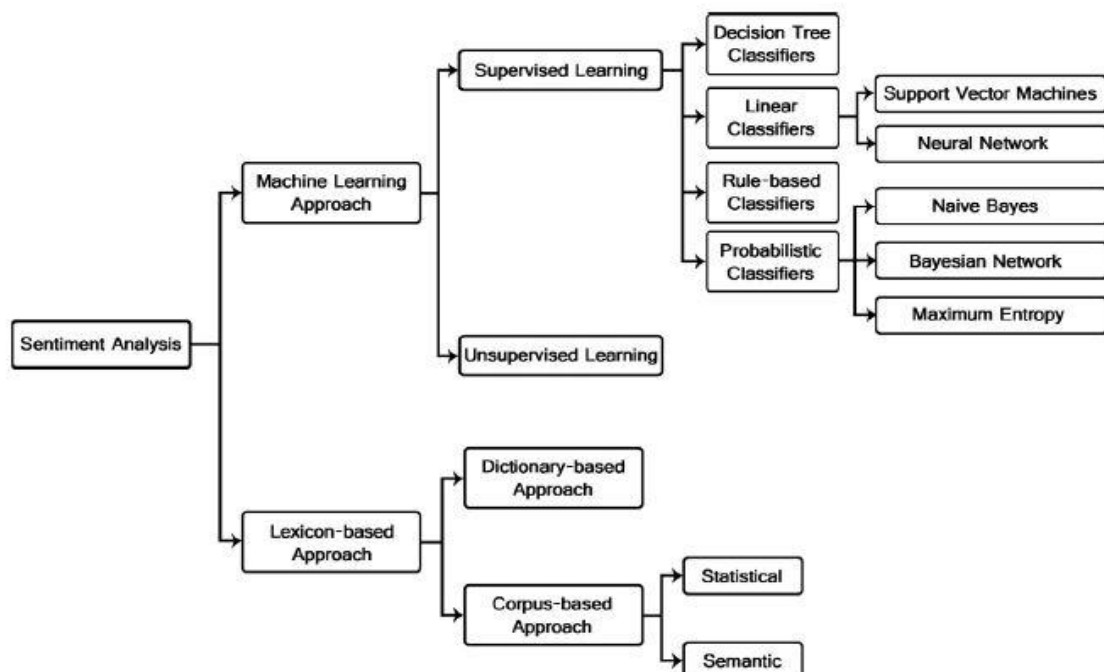
Es considerado como una tarea encargada de tratar computacionalmente elementos subjetivos presentes dentro del lenguaje natural como son las opiniones o los sentimientos, vienen en palabras, frases o textos completos y pertenecen a un tema en específico, ya sea en deportes, salud, política, entre otros (Yates & Berthier, 1999).

El objetivo de esta tarea es clasificar una serie de textos, asignándole una polaridad a cada frase, que en la mayoría de los casos suele ser dicotómica “positivo” o “negativo”, aunque hay casos donde se asignan rangos numéricos y dependiendo de dichos rangos, se establece la polaridad del texto.

Los sentimientos asociados a cada texto pueden estar presentes de manera explícita, por ejemplo, la frase “El árbol es hermoso” contiene un adjetivo que describe el árbol, mientras que la frase “La carga del celular duró solamente dos horas” presenta un sentimiento u opinión que no es fácil de identificar, por esta razón la mayoría de los problemas abordados, son aquellos en donde los textos pueden ser relacionados a una opinión de manera explícita (Liu, 2006).

Figura 2

Enfoques análisis de sentimientos



Nota. Tomada de *Sentiment classification techniques* (p. 14), por Medhat, Hassan & Korashy 2014, *Sentiment analysis algorithms and applications: A survey*.

Como lo mencionan Medhat, Hassan y Korashy (Medhat, 2014), y se observa en la Nota., en el análisis de sentimientos existen los enfoques basados en lexicones o *Machine Learning*, a continuación, se describirán algunos aspectos acerca de ellos.

- **Basados en Lexicon:** Usa las palabras de opinión tanto positivas como negativas para expresar estados que pueden ser o no deseados, también se emplean frases con información adicional que se denominan lexicones de opinión. En esta línea de trabajo hay dos enfoques que son los basados en diccionarios y en corpus. El primero emplea un conjunto inicial con palabras de opinión recolectadas manualmente y se agregan de manera automática más de bases de datos léxicas como WordNet o Tesoros, tanto sinónimos y antónimos del conjunto de partida adicional, el proceso termina cuando no se encuentran más palabras, adicionalmente se emplean sistemas de reglas que mejoran la eficiencia en su aplicación, ya que presentan un inconveniente y es que no tienen la capacidad de encontrar palabras de opinión en un dominio o contexto específico. En el segundo enfoque resuelve el problema del primero mediante el análisis de patrones o características sintácticas de los textos, donde se usan los adjetivos como semillas de

opinión y mediante restricciones lingüísticas se encuentran palabras de opinión adicional, con el conector (y) se puede indicar que el sentimiento expresado va en la misma línea, mientras que (pero) puede ser empleado como una expresión adversa.

- **Enfoque en Machine Learning:** En este enfoque se aborda el problema mediante la clasificación supervisada utilizando características sintácticas o lingüísticas de los textos. Donde se cuenta con registros pertenecientes a una clase en particular y se emplean para entrenar un modelo, el cual se usa con el fin de asignar a nuevos registros de los que se desconoce clase su respectiva etiqueta. Existen diferentes tipos de modelos que se pueden entrenar los cuales están basados en probabilidad o en clasificaciones lineales.

5.2.4 Preprocesamiento de textos

Los textos en sí representan datos con que se pretende extraer la información, pero es necesario tratar esos textos antes de aplicar el análisis de sentimientos y así clasificarlos. Ariel Pérez (Pérez Vera, 2017) plantea una forma de hacerlo y es como se menciona a continuación:

5.2.4.1 **Representación de Documentos.** Existen diversas formas de representación de los textos, algunos de los métodos más usados se mencionan a continuación:

- **N-gramas:** Es empleado para crear una secuencia de elementos a partir un texto, que pueden ser sílabas, palabras o fonemas. Aunque existen n-gramas que pueden contener pares de palabras o más, esto dependerá del caso de estudio (Sidorov, Velasquez, Stamatacos, Alexander, Hernández, 2012). Por ejemplo, para la frase “*Amor con amor se paga*”, su representación mediante bi-gramas se daría de la forma: “*amor con*”, “*con amor*”, “*amor se*”, “*se paga*”.
- **Representación Matricial:** Este enfoque busca una forma matemática de representar una colección textos. Consiste básicamente en armar una matriz de tamaño $m \times n$, donde m es el número de filas que vienen a ser la cantidad de textos en la colección y n las columnas (palabras) presentes en la colección de textos. Para llenar la matriz se establece la relación entre la palabra y la frase, esta puede darse de varias maneras: ya sea de forma binaria (0 si no aparece la palabra y 1 en caso contrario) o con el número de veces que aparece la palabra en el texto (Perez, 2017). Para las frases “*el niño es*

veloz”, “la niña es veloz”, “el perro es veloz”, la matriz de representación se observa en la Tabla 2.

Tabla 2

Representación matricial de textos

	La	El	Niña	niño	Perro	Es	veloz
Frase 1	0	1	0	1	0	1	1
Frase 2	1	0	1	0	0	1	1
Frase 3	0	1	0	0	1	1	1

5.2.4.2 **Reducción de dimensionalidad.** El análisis de los textos en muchas ocasiones resulta ser muy complicado, y es necesario eliminar elementos contenidos que no aportan información relevante o que no permiten realizar una buena clasificación. Por ejemplo: conectores, signos de puntuación y preposiciones son algunos de ellos.

5.2.4.3 **Lematización.** Está basado en un análisis morfológico de las palabras y se usa para transformarlas a su respectiva raíz léxica, es decir, se encuentra un lema que es su forma aceptada dentro de todas las posibles. Esto ayuda mucho en el análisis de sentimientos, ya que reduce considerablemente la dimensionalidad del problema a trabajar.

5.2.4.4 **Extracción de características.** Uno de los algoritmos más usados para extraer las características es el tf-idf (frecuencia de términos-frecuencia inversa de documentos) que se aplica principalmente en recuperación de información y es usado para medir la importancia de una palabra que pertenece a un documento. Viene dada por la ecuación (1).

$$w_d = f_{w,d} * \log(|D|/f_{w,D}) \quad (1)$$

Donde $f_{w,d}$ es la frecuencia de aparición del término w en el documento d , $f_{w,D}$ es el número de veces que se encuentra en D . Finalmente $|D|$ es el total de documentos (Salton & Buckley, 1988, Berger, et al, 2000).

5.2.5 Procesamiento de Textos.

Actualmente esta tarea se lleva a cabo usando *Machine Learning*, que permite procesar y clasificar conjuntos grandes de textos. Algunos de los algoritmos más usados dentro del análisis de sentimientos son:

- Clasificador bayesiano.
- Máquinas de soporte vectorial.
- Redes neuronales.

5.2.5.1 Clasificador bayesiano. Tiene como fundamento el Teorema de Bayes (Bayes, 1764). En el que se consideran dos conjuntos de sucesos denotados como A y B y se busca calcular la probabilidad de ocurrencia del evento en un conjunto, dada la probabilidad del otro. En el presente caso A es el primer conjunto de eventos y B el segundo. La probabilidad del suceso A viene denotada por $P(A)$, la de B como $P(B)$ y la de B dado A es $P(B/A)$. Entonces la probabilidad a posteriori de A en el que se verifica B, es decir $P(A/B)$ está dada por la ecuación (2).

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)} \quad (2)$$

Al implementar el teorema de Bayes en problemas de clasificación, para un texto que se ha caracterizado mediante un conjunto de palabras $x_1, x_2, x_3 \dots x_n$, con el fin de determinar su pertenencia a una de las clases $c_1, c_2, c_3, \dots, c_m$. Y de tal manera que su probabilidad se maximiza, encontrando el argumento de la máxima probabilidad como lo indica la ecuación (3).

$$Arg_c[Max P(C|x_1, x_2, \dots, x_n)] \quad (3)$$

Si el conjunto de palabras es expresado como: $X = \{x_1, x_2, x_3 \dots x_n\}$, el problema quedaría de la forma: $Arg_c[Max P(C/X)]$. Por lo tanto, para la calcular la probabilidad a posteriori usando el teorema de Bayes, debe emplearse la ecuación (4):

$$P(C|x_1, x_2, x_3, \dots, x_n) = \frac{P(x_1, x_2, x_3, \dots, x_n|C)P(C)}{P(x_1, x_2, x_3, \dots, x_n)} \quad (4)$$

Entonces el problema de clasificación se puede escribir de una forma más compacta usando la regla de Bayes según lo indica la ecuación (5):

$$Arg_c \left[Max \left[P(C|X) = \frac{P(X|C)P(C)}{P(X)} \right] \right] \quad (5)$$

El denominador $P(X)$ puede ser considerado como una constante ya que no varía entre las clases, es decir no incluye ningún cálculo relacionado con las clases. Finalmente, el problema de clasificación usando el teorema de Bayes está dado por la ecuación (6).

$$Arg_c [Max[P(C|X) = \alpha P(X|C)P(C)]] \quad (6)$$

Si se desea resolver el problema con el enfoque bayesiano es necesario conocer la probabilidad a priori de la clase $P(C)$, también la probabilidad de las palabras dada la clase $P(X/C)$. Para calcular la probabilidad a posteriori $P(C/X)$, se requiere obtener previamente las probabilidades a priori con los datos de entrenamiento (Sulcar, n.d). (Ashraf M et all, s.f.) plantean una forma de hallar $P(X|C)$ para un clasificador multinomial bayesiano, como se ilustra en la ecuación (7).

$$P(X|C) = \left(\sum_n f_n \right)! \prod_n \frac{P(w_n|C)^{f_n}}{f_n!} \quad (7)$$

Donde el conteo de las palabras n para el conjunto de palabras X viene dado por f_n . Finalmente $P(w_n|C)^{f_n}$ es la probabilidad de obtener la palabra n dada la clase C , que se puede según la ecuación (8) :

$$\hat{P}(w_n|C) = \frac{1 + F_{nc}}{N + \sum_{x=1}^N F_{xc}} \quad (8)$$

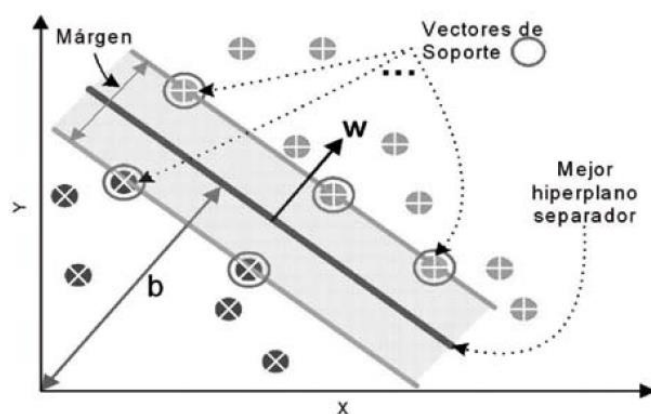
El valor de F_{xc} es el conteo de todas las palabras que pertenecen a la clase C .

5.2.5.2 **Máquinas de soporte vectorial.** Están fundamentadas en la teoría de aprendizaje estadístico desarrollada por V. Vapnik y A. Chervonekis en el año 1995 (Vapnik, Chervonekis, 1995). Pueden ser usadas para abordar problemas de clasificación y regresión, con un buen desempeño aun cuando se cuente con pocos datos de entrenamiento.

En las máquinas de soporte vectorial se abordan dos tipos de problemas, los linealmente separables y clasificadores no lineales. Para el caso lineal se dispone de los vectores de palabra que son de la forma $(x_1, w_1, z_1, \dots, u_1)$, $(x_2, w_2, z_2, \dots, u_2)$, ..., $(x_n, w_n, z_n, \dots, u_n)$ ² con x_n, w_n, z_n y $u_n \in \mathbb{R}^n$. Donde cada escalar y_i denominado etiqueta, corresponde a una de las dos clases que se denotan como +1 y -1. El vector de etiquetas puede ser representado como $Y = (y_1, y_2, \dots, y_n)$.

Figura 3

Máquina de soporte vectorial en dos dimensiones



Nota. Recuperada de *Hiperplano de separación óptima $w \cdot x + b$ para el caso bidimensional* (p. 76). por Jiménez & Rengifo, 2010, Facultad de Ciencias Naturales y Exactas Universidad del Valle.

Como se muestra en la Figura 3, la idea es encontrar un hiperplano capaz de clasificar los vectores de palabras maximizando la distancia de este con los ejemplos más cercanos de cada clase, las máquinas de soporte vectorial tienen la particularidad de que cumplen con dicha condición y al hiperplano que lo maximiza lo denominan hiperplano de margen o separación óptima (HSO). Que viene expresado como: $(w \cdot x) + b = 0$ donde $w, x \in \mathbb{R}^n, b \in \mathbb{R}$. Aquí el trabajo es encontrar el vector de

² En el apartado 5.2.4.1 se indica cómo se pueden construir vectores a partir de las palabras y los valores que este puede tener asociados

pesos w . Por otra parte, la función de la distancia (ejemplo más cercano al hiperplano separador) viene dada por la ecuación (9).

$$d(x, w, b) = \frac{|(w \cdot x) + b|}{\|w\|} \text{ con } \|w\| = \sqrt{(w \cdot w)} \quad (9)$$

Para el caso de los clasificadores no lineales (cuando no existe una superficie de decisión lineal), se mapea el vector de entrada X a uno de mayor dimensión denominado espacio de características $F = \{\phi(x) | x \in X\}$ de la forma (10).

$$x = \{x_1, x_2, \dots, x_n\} \rightarrow \phi(x) = \{\phi(x)_1, \phi(x)_2, \dots, \phi(x)_n\} \quad (10)$$

Con el fin de plantear una función kernel K tal que para todo $x, z \in X$. Quedaría expresado como lo indica (11).

$$K(x, z) = \langle \phi(x) \cdot \phi(z) \rangle \quad (11)$$

Donde $\phi(x) \cdot \phi(z)$ representa el producto punto dentro del espacio de características de una dimensión arbitraria. Encontrado este espacio de mapeo apropiado, se resuelve el problema como se hace en el caso lineal (Jiménez & Rengifo, 2010).

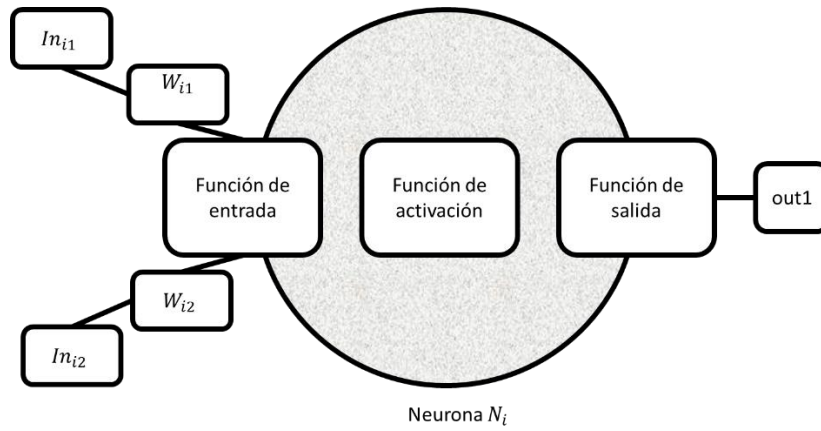
5.2.5.3 Redes Neuronales Artificiales. Según Jorge Matich (Matich, 2001), se pueden definir como:

- Una forma de computación bio inspirada.
- Un modelo matemático, organizado por elementos y niveles de procesamiento.
- Un sistema de computación que se compone de varias unidades que interactúan procesando información alterando su estado de acuerdo con las entradas recibidas.

Las redes neuronales también son consideradas como un sistema capaz de emular algunas de las actividades que realizan los seres humanos tales como memorizar y asociar hechos. Su unidad básica es la neurona artificial que consta de tres elementos esenciales: función de entrada, activación y salida.

Figura 4

Neurona Artificial



Nota. Tomada de *ejemplo de una neurona con 2 entradas y 1 salida* (p. 13), por Matich J., 2014, *Redes Neuronales: Conceptos Básicos y Aplicaciones*.

La función de entrada recibe múltiples valores (que para la neurona N_i de la Figura 4 serían i_{ni1} y i_{ni2} , con pesos w_{i1}, w_{ni2} respectivamente) y los transforma en uno solo. El conjunto de n elementos $ni_i = (ni_{i1}, ni_{i2}, \dots, ni_{in})$ se conoce como vector de entradas. Con la regla de propagación se unifican los valores que ingresan a cada neurona, esto se puede realizar usando cualquiera de las siguientes opciones:

- **Sumatoria de entradas ponderadas:** Es el valor total de la suma de cada una de las entradas multiplicada con su peso correspondiente (12).

$$\sum_j (n_{ij} w_{ij}) \text{ con } j = 1, 2, \dots, n \quad (12)$$

- **Productoria de entradas ponderadas:** Es el producto de todas las entradas multiplicadas por sus correspondientes pesos (13).

$$\prod_j (n_{ij} w_{ij}) \text{ con } j = 1, 2, \dots, n \quad (13)$$

- **Máximo de entradas ponderas:** Solamente tiene en cuenta el valor de mayor fuerza, multiplicado por su correspondiente peso (14).

$$\text{Max}_j (n_{ij} w_{ij}) \text{ con } j = 1, 2, \dots, n \quad (14)$$

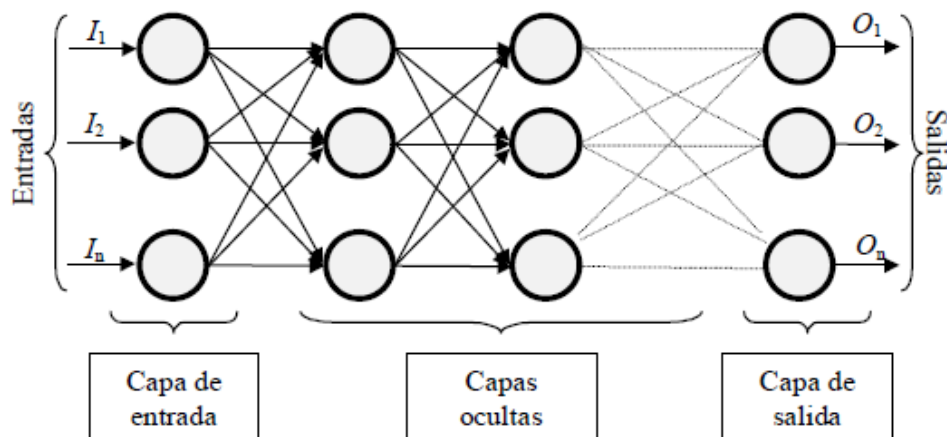
La función de activación se encarga de tomar el valor de entrada global que se puede denotar como (gn_i) y transformarlo en uno nuevo que active o inhiba a la neurona artificial, sus rangos de activación son valores numéricos. Para calcular el estado de la neurona se toma (gn_i) restándole un valor denominado umbral θ_i , y se evalúa usando alguna de las funciones más conocidas como la lineal, sigmoidea o tangente hiperbólica (Matich,2001).

Finalmente, la función de salida es un valor dentro de una red neural que se transfiere de una neurona a otra, este puede ser binario $\{-1,1\}$, $\{0,1\}$ o estar comprendido dentro de un rango $[-1, +1]$, $[0,1]$ o $(-\infty, \infty)$.

La Figura 5 muestra la forma en que las redes neuronales artificiales se organizan para dar solución a muchas de las tareas asignadas, esto es a través de tres niveles o capas encargadas de procesar los datos de entrada y se conocen como capas de: entrada, oculta (puede estar constituida por varias y dependen de su topología) y de salida.

Figura 5

Capas en Redes Neuronales



Nota. Tomada de *ejemplo de una red neuronal totalmente conectada* (p. 12), por Matich J., 2014, Redes Neuronales: Conceptos Básicos y Aplicaciones.

Para usar redes neuronales en el análisis de sentimientos, la capa de entrada recibirá los datos proporcionados por la matriz de representación de textos, donde se emplearán tantas neuronas como palabras se presenten. Las capas ocultas pueden variar en su número y cantidad de neuronas, estas últimas recibirán los valores correspondientes a las salidas de sus antecesoras aplicando a su ingreso la regla de propagación definida (ya sea de sumatoria, productoria o entradas ponderadas), y finalmente para la capa de

salida de usará una sola neurona que corresponderá a la etiqueta de la palabra correspondiente.

5.2.6 Métricas de Rendimiento

Con la finalidad de evaluar un modelo entrenado de *machine learning*, es necesario construir la matriz de confusión, y a partir de esta calcular una serie de valores que darán su respectivo indicador de desempeño. Según Ron Kohavi y Foster Provost (kohavi,1998), la matriz de confusión es una relación entre los datos actuales y predichos.

Tabla 3

Matriz de confusión para dos clases

		Actual	
		positivo	Negativo
predicho	Positivo	VP	FN
	Negativo	FP	VN

En la Tabla 3 se puede ver un ejemplo de para dos clases con etiquetas “positivo” y “negativo”, donde los valores dentro de la relación corresponden a los ejemplos clasificados como:

- Positivos siendo positivos (VP).
- Negativo siendo positivo (FN).
- Positivo siendo negativo (FP).
- Negativo siendo negativo (VN).

A partir de estos se calculan los indicadores de desempeño de un modelo de *machine learning* y son los que se mencionan a continuación:

- **Accuracy:** Es el número de ejemplos clasificados correctamente frente al total de ejemplos, este valor se puede calcular mediante la fórmula (15).

$$Accuracy = \frac{(VP + VN)}{(VP + VN + FP + FN)} \quad (15)$$

- **Precision:** Indica el número de predicciones acertadas para una clase sobre el total realizadas sobre esta. Para la etiqueta “positivo” se indica la forma de calcularla, según la ecuación (16).

$$Presicion = \frac{(VP)}{(VP + FN)} \quad (16)$$

- **Recall:** Corresponde a los ejemplos clasificados correctamente sobre el total de la muestra para una clase, en la ecuación (17) se observa la forma de calcularlo para la etiqueta “positivo”.

$$Recall = \frac{(VP)}{(VP + FP)} \quad (17)$$

5.2.7 TWITTER

Es un servicio de Microblogging creado y lanzado en el año 2006 por Jack Dorsey, que cuenta actualmente con casi 500 millones de usuarios que producen más de 500 millones de Tweets al día (Matt Ahlgren, 2020). El elemento más importante dentro de esta red es el tuit y según la RAE es un “Mensaje digital que se envía a través de la red social Twitter y que no puede rebasar un número limitado de caracteres.” (Real Academia Española, s.f.). La red social cuenta con las siguientes características:

- Permite publicar texto (con un máximo de 280 caracteres), imágenes o videos.
- Los tuits son públicos, aunque se puede limitar su visualización si así se desea.
- Se pueden seguir otros usuarios.
- Para iniciar o referenciar un tema en específico se hace uso de los hashtags #xxxx
- Usando @ seguido del usuario, se le puede mencionar en un tuit.
- Los hilos sirven para mantener el contexto de una serie de Tweets pertenecientes a un mismo usuario.
- Se puede compartir un tuit publicado por otro usuario con la opción de retuitear.
- Con la opción de “me gusta” se puede indicar que un Tweet es de interés.
- Las listas permiten organizar la información relevante para un usuario.
- Se pueden mantener conversaciones privadas sobre tweets usando mensajes directos.

- Se consideran línea de tiempo a lo que una persona ve de manera predeterminada en el Inicio de Twitter.

5.2.8 SCIKIT LEARN

Es una librería de aprendizaje automático desarrollada por Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort y Vincent Michel (Pedregosa et al, 2011) para *Python*, que permite llevar a cabo el entrenamiento de diversos modelos para tareas de clasificación, regresión, clustering. Además, funciones de preprocesado, reducción de dimensionalidad o selección de modelos. Esta librería está construida a partir de otras como:

- ***Numpy***: Un paquete esencial en la computación científica con Python que provee un objeto de matriz multidimensional denominado *ndarray*, que junto a una serie de rutinas que se encuentran en código compilado, permite realizar operaciones rápidas que incluyen manipulación matemática, clasificación, transformadas discretas de Fourier, entre otras.
- ***Scipy***: Es una librería que contiene una colección de funciones y algoritmos matemáticos desarrollados para *numpy* con la finalidad de crear entornos de procesamiento de datos.
- ***Pandas***: Provee un conjunto de funcionalidades para cargar y tratar datos que vienen de diferentes fuentes como archivos (Json, Excel, Csv) o bases de datos, con varias opciones para el filtrado, limpieza y generación de estadísticas básicas. Además, cuando se trabajan con tablas se pueden hacer cambios estructurales, concatenación con otras y obtención de filas o columnas sin necesidad de iterar sobre todos los datos.
- ***Matplotlib***: Es un paquete que permite realizar diversos gráficos a través de una serie de datos que se encuentren en listas de Python.

5.2.9 CONDA

Es un gestor de paquetes y entornos desarrollado en Python, con la capacidad de trabajar con diferentes lenguajes de programación y sus dependencias, facilitando la administración de diferentes ambientes de desarrollo, evitando conflictos entre versiones (Conda, s.f).

5.2.10 Tweepy

Es una librería desarrollada por Twitter en Python con la finalidad de consumir el api creada por ellos mismos, para usarla es necesario crear una cuenta la cual tendrá asociada las credenciales de *consumer_key*, *consumer_secret*, *access_key*, *access_secret* con sus respectivos valores, que se usan como mecanismo de autenticación mediante OAUTH2 con Twitter. Un usuario registrado puede interactuar con la plataforma a través de Tweepy, y realizar las siguientes acciones:

- Crear un Tweet.
- Buscar Tweets.
- Obtener los 20 Tweets más recientes.
- Obtener un Tweet.
- Actualizar un Tweet.
- Eliminar un Tweet.
- Actualizar un Tweet con contenido multimedia.
- Hacer un Retweet.
- Deshacer un Retweet.

En la Figura 6 se indica la forma en la que se invocan los métodos de *OAuthHandler* y *set_access_token* para obtener un token de acceso y usarlo en la clase *API* de Tweepy.

Figura 6

Configuración de objetos *OAuthHandler* y *Api* de *tweepy*.

```
10 import tweepy
11 auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
12 auth.set_access_token(access_key, access_secret)
13 api = tweepy.API(auth)
14
```

Nota. Fuente Elaboración propia

Tweepy tiene un gran número de funciones para realizar acciones en Twitter, pero con la finalidad de obtener los tweets mediante paginación se puede usar el objeto *Cursor* de *Tweepy*, que facilita la iteración entre líneas de tiempo, mensajes directos o listas de usuarios, al llamarlo es necesario pasar los parámetros que se mencionan en la Tabla 4.

Tabla 4Parámetros cursor *Tweepy*

Nombre	Descripción
<i>Tweet_mode</i>	El usar el valor extended, permite obtener toda la información del tweet, incluyendo todos los caracteres.
<i>include_entities</i>	Cuando se asigna el parámetro en True incluye información relacionada con el tweet, como hashtags, url, archivos multimedia.
lang	Lenguaje en el que se desea obtener la información, al asignarse en “es” descarga tweets en español
q	Es un query de búsqueda en el que se pueden ingresar distintos valores como fechas desde-hasta y el tema en particular del que se desea buscar.

Al invocar la función `items` del *Cursor* se devuelven los elementos dentro de la paginación, donde recibe un único parámetro y corresponde a la cantidad de tweets que se obtendrán. En la Figura 7 se puede observar una función que llama al *Cursor*.

Figura 7Llamado al *Cursor* de *Tweepy*

```
def obtener_Tweets(query,cantidad,apiCreacion):  
    cursor_twitter = tweepy.Cursor(apiCreacion.search,q=query, lang="es",tweet_mode='extended',include_entities=True)  
    cursor_final = cursor_twitter.items(cantidad)  
    print(cursor_final)  
    return cursor_final  
  
cursor_api_Final = obtener_Tweets('covid 19',500,api)  
print(cursor_api_Final)
```

Nota. Fuente: Elaboración propia.

El objeto *tweet* dentro de *Tweepy* contiene información acerca de este aparte del texto y fecha de creación, algunos de los campos contenidos en este con los que se mencionan en la Tabla 5.

Tabla 5Campos Objeto tweet de *Tweepy*

Atributo	Tipo	Descripción
<i>created_at</i>	String	Fecha de creación del Tweet
<i>id</i>	Base 64	Identificador en base 54
<i>Id_str</i>	String	Identificador en string
<i>Text</i>	String	Texto en formato utf-8 acerca del estado del tweet.
<i>Full_Text</i>	String	Texto completo cuando un tweet supera los 140 caracteres.
<i>Source</i>	String	Sitio de origen del tweet, puede ser sitio web o api.
<i>Truncated</i>	String	Indica si un tweet ha sido truncado, es decir que termina en ... dado que se presenta un límite en los caracteres que es de 140, el texto original se encontrara en el retweet status.
<i>in_reply_to_status_id</i>	String	Si es una respuesta al tweet tendrá la referencia
<i>user</i>	Objeto	Contiene la información del usuario que ha postado el tweet.

Como la función *items* del *Cursor* devuelve un iterable, una de las formas de tratar con este elemento es recorrerlo y almacenar los datos de interés de cada objeto dentro de una lista. En la Figura 8 se observa una función para almacenar los textos de los tweets en una lista.

Figura 8

Función para tratar ítems del cursor

```
74 def crearlistaTweets(items_cursor):  
75     lista_tweets = []  
76     for tweet in items_cursor:  
77         if 'retweeted_status' in dir(tweet):  
78             tweet_f=tweet.retweeted_status.full_text.encode('latin-1',errors='ignore').decode('latin-1')  
79             if tweet_f not in lista_tweets:  
80                 lista_tweets.append(tweet_f)  
81         else:  
82             tweet_f = tweet.full_text.encode('latin-1',errors='ignore').decode('latin-1')  
83             if tweet_f not in lista_tweets:  
84                 lista_tweets.append(tweet_f)  
85     return lista_tweets
```

Nota. Fuente: Elaboración propia.

5.2.11 PATTERN

Es un módulo de minería web creado por Smedt. y Daelemans (Smedt T. & Daelemans W., 2012), para el lenguaje de programación Python en su versión 2.5, el cual consta de los siguientes componentes:

- **Minería web:** Contiene la clase URL, que es una extensión de *urllib2.Request*, de Python. Esta se usa con la finalidad de acceder a diferentes direcciones web y obtener el contenido de estas (código fuente o el maquetado de HTML). Adicionalmente cuenta con la opción de *search engine*, que le permite conectarse con diferentes servicios como Google, Wikipedia. Facebook con la finalidad de retornar información de estos en forma de objetos.
- **Procesamiento de lenguaje natural:** Cuenta con un etiquetador para la parte del discurso (*part-of-speech*), que permite identificar en una oración sustantivos, adjetivos o verbos en una oración. Adicionalmente brinda la opción para conjugar verbos, pluralizar o singularizar Sustantivos. Este cuenta con diferentes interfaces para soportar idiomas como el español, inglés y francés.
- **Machine learning:** Cuenta con la implementación de algoritmos para la clasificación (*Naive Bayes*, *k-nearest neighbor*, *single-layer averaged perceptrón* y *support vector machine*) o agrupamiento en el que solamente usa *k-means clustering*.
- **Visualización en canvas:** Consiste en un conjunto de funcionalidades para el análisis de gráficos y visualización en el navegador. Estos se pueden emplear en el análisis de grafos o relaciones semánticas.

5.2.12 Coronavirus

Como lo mencionan Maguiña, Gastelo y Tequen (Maguiña, 2020) los coronavirus hacen parte de una familia extensa de virus que puede transmitirse tanto en animales como en los seres humanos. En las personas puede presentar desde resfriado común hasta casos más graves como el síndrome respiratorio de Oriente Medio (MERS, 2012) o el síndrome respiratorio agudo severo (SRAS,2003).

5.2.12.1 COVID-19

Es una enfermedad infecciosa producida por el virus SARS-COV2 descubierta recientemente y que se originó en Wuhan (China) en diciembre del año 2019. Los síntomas que una persona generalmente desarrolla son fiebre, tos seca y cansancio, por otro lado, otros menos comunes se mencionan a continuación:

- dolores y molestias la congestión nasal
- dolor de cabeza
- conjuntivitis.
- dolor de garganta
- diarrea
- pérdida del gusto o el olfato.
- erupciones cutáneas o cambios de color en los dedos de las manos o los pies

Según la OMS el 80% de las personas se recuperan de la enfermedad sin la necesidad de asistencia médica, también que 1 de cada 5 personas presenta gravedad y tiene dificultades para respirar. Donde las personas mayores con problemas de salud tales como diabetes, cáncer, hipertensión arterial y/o afecciones pulmonares, son las más propensas a desarrollar la enfermedad (Organización Mundial de la Salud OMS,2021).

La forma en la que se propaga la COVID-19 es mediante la inhalación de gotículas infectadas que alguna persona expulsa al estornudar, toser o al hablar. También puede contraerse si hay superficies como barandas, mesas con gotículas infectadas, que al tocarlas con las manos y después llevarlas a boca y nariz transmite el virus al organismo (Ministerio de Sanidad, Consumo y Bienestar Social de España, 2021),

Las medidas que se establecen para la prevención y cuidado contra la COVID-19 van desde el lavado de manos, el uso de mascarillas, el distanciamiento social (1 metro) y el aislamiento

durante catorce días si se llega a presentar levemente alguno de los síntomas anteriormente mencionados (Organización Mundial de la Salud OMS,2021).

La pandemia COVID-19 ha afectado significativamente la vida de los seres humanos a nivel mundial. Según Scottie Andrew (Andrew S, 2020), en 211 países se reportaron casos del virus donde 2.780.000 personas han perdido la vida, 1600 millones de niños no podían regresar a sus escuelas por las ordenes de confinamiento, cambiando el modelo de educación presencial a uno virtual. La economía a nivel mundial presentó muchos inconvenientes ya que se cancelaron importaciones y exportaciones, se perdieron muchos empleos, las empresas comenzaron a perder liquidez y según el Banco Mundial (Banco Mundial, 2020) los países más afectados son aquellos que dependen del comercio exterior, además el PIB en muchas naciones presentó una contracción y millones de personas quedarán en la pobreza extrema. Los sistemas de salud colapsaron ya que el número de unidades de cuidado intensivo requeridas para el tratamiento de los pacientes no eran suficientes para atender la demanda.

Para evitar nuevas propagaciones del virus y con el fin de reactivar los distintos sectores de la sociedad, se han desarrollado vacunas que están compuestas por antígenos que son formas del virus muertas o débiles que brindan al organismo la capacidad de reconocerlas en un futuro, adicionalmente de adyuvantes que mejoran la respuesta de la vacuna, finalmente de conservantes y estabilizantes que son útiles para los procesos de transporte y mantienen la eficacia de estas mismas. Las organizaciones Johnson & Johnson/Janssen, AstraZeneca/ Universidad de Oxford, Moderna y Pfizer/BioNTech se encargaron de producirlas y son las que cuentan con aprobación para ser aplicadas en las personas, y a nivel mundial se están usando para inmunización (Unión Europea, 2021). En Colombia y según el Ministerio de Salud (MinSalud,s.f) se adquirieron 41.5 millones de dosis para alcanzar un total de 35.250.000 de colombianos, comenzando el 21 de febrero el proceso de vacunación las personas de la tercera edad y pertenecientes al sector de la salud. Donde al 21 de septiembre del año 2021 ya se ha inmunizado un 32.1% de la población colombiana.

6 Desarrollo de la Propuesta

En esta sección se lleva a cabo el desarrollo la propuesta del presente trabajo de grado, basada en la que planteó Eman Younis (Younis E., 2015) para el tratamiento de los Tweets. Adicionalmente se mencionan detalles del aplicativo desarrollado mediante la metodología Scrum.

6.1 Tratamiento de textos

En esta sección se describe el proceso de cómo los Tweets son descargados, etiquetados, tratados y transformados con la finalidad de obtener los datos necesarios para entrenar y realizar las pruebas con los algoritmos de *machine learning* planteados para este trabajo.

6.1.1 Acceso a los textos

Con la finalidad de obtener los tweets relacionados con la covid-19, se llevó a cabo la creación de una cuenta en Twitter, adicionalmente hacer el registro en la sección de desarrolladores para obtener las credenciales de acceso al Api de Twitter mediante Tweepy.

Como existen limitaciones de descarga para las cuentas gratuitas, se creó un script *script* en Python para obtener y guardar los tweets en un archivo de texto, en la Figura 9 se muestra una función que genera un fichero a partir de una lista de textos, que se ejecutó diariamente y así finalmente se consiguieron un total de 20000 tweets en un lapso de 2 meses.

Figura 9

Guardado de tweets en texto plano

```
def escribirArchivoTweets(listaTweets,query,fecha_formateada):
    nombreArchivo = '%s%s.txt' % (query,fecha_formateada)
    archivo = open(nombreArchivo,'w')
    for tweet in listaTweets:
        archivo.write(tweet)
        archivo.write('\n')
    archivo.close()
```

Nota. Fuente: Elaboración propia.

6.1.2 Etiquetado de los Tweets

En esta sección se usaron dos enfoques para la clasificación de los tweets, en el primero se trataron 8000 de forma manual con el apoyo de 3 personas a quienes se les solicitó asignar a una misma muestra de tweets su respectiva etiqueta, que al final se consolidaron en una sola teniendo presente la mayoría, en la se puede observar la forma en la que se hizo.

Tabla 6

Convención para consolidado de etiquetas de tweets

Persona 1	Persona 2	Persona 3	Etiqueta Final
Etiqueta 1	Etiqueta 1	Etiqueta 2	Etiqueta 1
Etiqueta 2	Etiqueta 2	Etiqueta 3	Etiqueta 2
Etiqueta 3	Etiqueta 2	Etiqueta 3	Etiqueta 3
Etiqueta 1	Etiqueta 3	Etiqueta 3	Etiqueta 3
Etiqueta 1	Etiqueta 3	Etiqueta 1	Etiqueta 1
Etiqueta 1	Etiqueta 2	Etiqueta 3	Aleatorio entre las 3 etiquetas

Para el enfoque de asignación de etiquetas automático, se usó un diccionario de sentimientos con términos positivos y negativos, después se verificó si las palabras de cada tweet se encontraban o no dentro del diccionario, asignándoles un valor numérico como lo indica la Tabla 7.

Tabla 7

Etiquetas enfoque automático

Palabra	Sentimiento
Positiva	1
Negativa	-1
No contenida	0

Posteriormente se sumaron los valores numéricos para cada tweet y el resultado se asociaba con su respectivo texto, así se procesaron un total de 12000 tweets teniendo presente solamente las palabras.

6.1.3 Convenciones usadas

Las etiquetas empleadas para el primer enfoque de etiquetado fueron de “positivo” si el tweet estaba relacionado con algo positivo durante la pandemia un ejemplo puede ser la vacunación de las personas, recuperación o descenso en el número de decesos, “negativo” en el caso de muertes de personas y “neutro” para cuando no se evidenciaba algún impacto en el tweet, siguiendo la convención que se muestra en la Tabla 8.

Tabla 8

Etiquetas de tweets

Etiqueta	
-1	negativo
0	neutro
1	positivo

En el enfoque automático se establecieron dos etiquetas, teniendo presente los valores numéricos calculados automáticamente para cada tweet, si este era mayor o igual a cero, entonces se asignaba como “positivo” y en caso contrario era “negativo”. En la Tabla 9 se muestra la convención empleada.

Tabla 9

Convención para dos etiquetas

Sumatoria	Etiqueta	Valor Numérico
Menor que cero	Negativo	-1
Mayor o igual a cero	positivo	1

6.1.4 Limpieza de los textos

Las frases son el insumo de entrenamiento de los diferentes algoritmos de *Machine Learning*, pero no todos los elementos contenidos en ella aportan un significado relevante dentro del

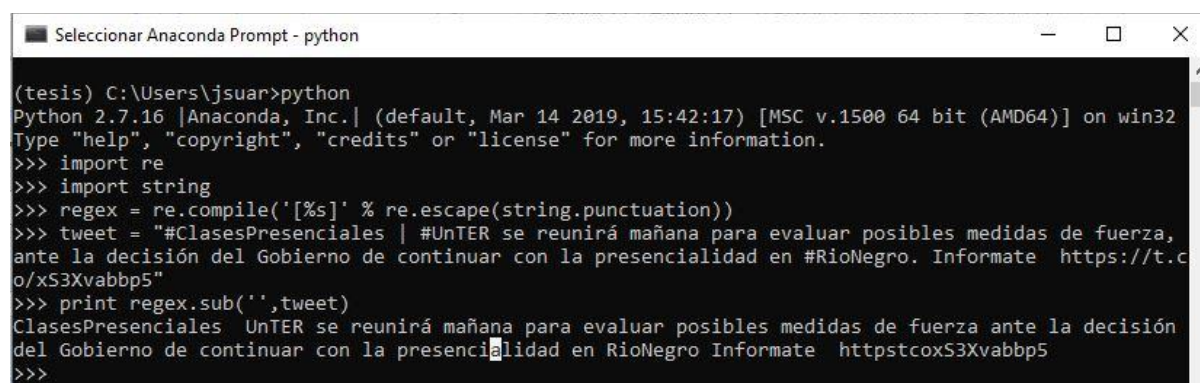
contexto de la minería de opinión o análisis de sentimientos. Por lo tanto, para todos los textos en esta fase se debe llevar a cabo la:

6.1.5 Eliminación de signos de puntuación

Por ejemplo, el Tweet *“#ClasesPresenciales | #UnTER se reunirá mañana para evaluar posibles medidas de fuerza, ante la decisión del Gobierno de continuar con la presencialidad en #RioNegro. Informate <https://t.co/xS3Xvabbp5>”*, posee conectores como puntos y comas que para un lector normal indican pausas o enlace de ideas o conceptos, pero no aportan alguna información relevante en minería de opinión.

Figura 10

Eliminación de signos de puntuación



```
Seleccionar Anaconda Prompt - python

(tesis) C:\Users\jsuar>python
Python 2.7.16 [Anaconda, Inc.] (default, Mar 14 2019, 15:42:17) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import re
>>> import string
>>> regex = re.compile('[%s]' % re.escape(string.punctuation))
>>> tweet = "#ClasesPresenciales | #UnTER se reunirá mañana para evaluar posibles medidas de fuerza,
ante la decisión del Gobierno de continuar con la presencialidad en #RioNegro. Informate https://t.co/xS3Xvabbp5"
>>> print regex.sub('',tweet)
ClasesPresenciales UnTER se reunirá mañana para evaluar posibles medidas de fuerza ante la decisión
del Gobierno de continuar con la presencialidad en RioNegro Informate httpstcoxS3Xvabbp5
>>>
```

Nota. Fuente: Elaboración propia.

Para eliminar todos los signos de puntuación se trabajó con el módulo de expresiones regulares de Python *re*, en el cual se compiló una expresión regular que servía para identificar y eliminar estos símbolos. En la Figura 10 se puede observar cómo se llevó a cabo esta tarea.

6.1.6 Eliminación de palabras vacías (stopwords)

Por otro lado, algunas palabras como conectores o proposiciones que no aportan ningún significado léxico se pueden eliminar y así reducir el tamaño del vocabulario obtenido), algunos de estos términos son: *“a, los, con, y, de, la”*.

Figura 11

Función para eliminar las palabras vacías

```
from stop_words import get_stop_words

stop_words = get_stop_words('spanish')

def eliminar_stop_words(frase_entrada):
    palabras = []
    frase_con_espacios = " ".join(frase_entrada.split())
    for palabra in frase_con_espacios.split(' '):
        palabra_decodificada = palabra.decode('string_escape')
        if palabra_decodificada not in stop_words:
            palabras.append(palabra_decodificada)
    return " ".join(palabras)
```

Nota. Fuente: Elaboración propia.

En la Figura 11 se muestra la manera en la que se eliminaron las palabras vacías mediante el uso del paquete de **“stopwords”** de Python para el idioma en español, que dentro de un arreglo carga todas las palabras que son consideradas como vacías. También se ilustra una función que permite eliminar estas palabras para una frase.

6.1.7 Lematizar las palabras

En un conjunto de textos se evidencian variaciones de una misma palabra, y a pesar de que tengan el mismo significado, representa un aumento del vocabulario final obtenido, por tanto, es necesario aplicar un tratamiento a estas mediante la lematización, que consiste en obtener la raíz léxica de una palabra mediante la consulta de algún recurso computacional previamente construido. Al importar el módulo para Python *pattern*, se pueden consultar los lemas de una palabra, en la Figura 12 se ilustra un ejemplo con la conjugación en gerundio simple “informando” para “informar”.

Figura 12

Modulo pattern para español

```
>>> import pattern.es as lemEsp
>>> lemEsp.parse('informando', lemmata=True)
u'informando/VBG/B-VP/O/informar'
```

Nota. Fuente: Elaboración propia.

6.1.8 Eliminación de urls

Mediante una expresión regular, se identificaron las urls contenidas en los Tweets y se eliminaron, dado que estas no aportan ningún valor semántico.

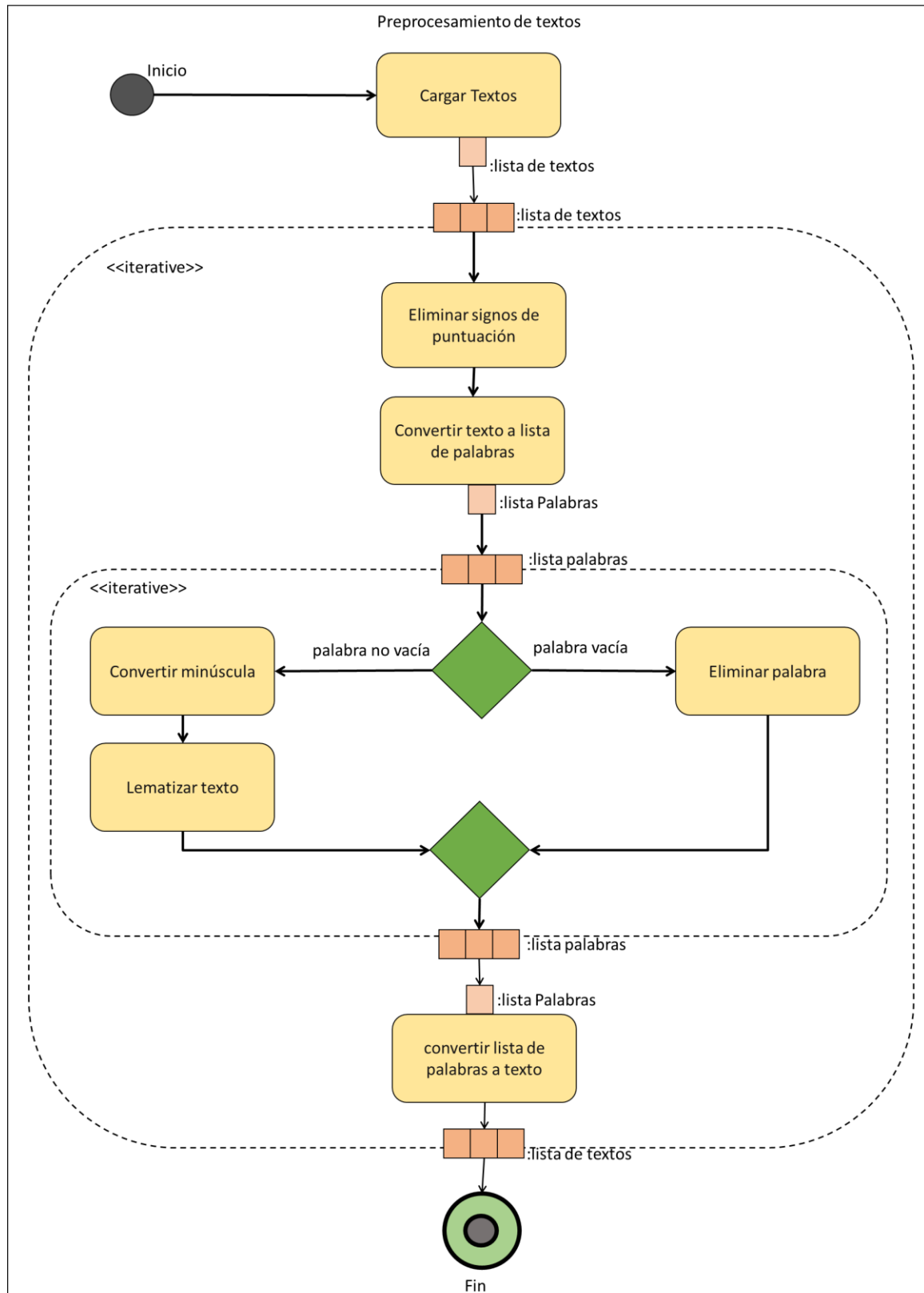
6.1.9 Casos especiales

Las menciones (palabras que comienzan con un @ y hace referencia a un usuario) y los hashtags (que inician mediante un #) son palabras relevantes dentro del marco de análisis de sentimientos para textos extraídos de Twitter, por lo sobre estos se eliminaron los símbolos mencionados anteriormente.

En la Figura 13 se muestra la forma en la que se llevó a cabo el preprocesado de todos los Tweets recolectados acerca de la covid-19. Donde se obtiene una nueva muestra usada como insumo para la representación matricial de textos.

Figura 13

Preprocesamiento de Tweets



Nota. Fuente: Elaboración propia

6.2 Análisis de los textos

Como los algoritmos de *machine learning* solamente reciben datos numéricos de entrada para ser procesados mediante diferentes técnicas, es necesario hacer una transformación sobre los textos para que cada algoritmo pueda ser entrenado y usado en clasificación.

6.2.1 Representación matricial

Una de las representaciones más comunes es la matricial, que tiene en cuenta la relación frases-palabras, donde a cada palabra contenida en la frase se le asigna un valor numérico que se puede calcular usando el factor tf-idf o solamente teniendo en cuenta la frecuencia de esta. La forma en la que se obtuvo la representación matricial en el presente trabajo fue de la siguiente manera:

- Tomar todos los textos preprocesados, ingresarlos a un script que leyera cada uno de ellos y los transformara en un vector de palabras. Con esto se tiene una colección de vectores de palabras.
- Contar el número de veces que aparece cada palabra dentro de los textos.
- Obtener la frecuencia para cada una de las palabras que aparecen en cada texto.
- Construir una matriz donde las columnas serán todas las palabras usadas dentro del entrenamiento y las filas los textos. El valor que se asignará en la relación será el obtenido por el factor tf-idf.

Para ilustrar a modo de ejemplo el proceso de construcción de la matriz de representación de textos, se usarán los siguientes Tweets:

- *“El ejército estadounidense se ha ofrecido a proporcionar algunas vacunas contra el coronavirus de Johnson & Johnson para las tropas surcoreanas, dijo el lunes el Ministerio de Defensa, mientras Corea del Sur lucha contra la escasez de inyecciones de COVID-19. <https://t.co/tAEbFJOITY>”*
- *Cada día hay más evidencias de que las vacunas contra el coronavirus también frenan la transmisión en TODOS LOS GRUPOS DE EDAD. Un aullido.*
- *“Un helicóptero cargado con trabajadores de la salud y dosis de vacuna contra el coronavirus despegó de Labrea, en la parte sur del Amazonas. No ven la tv”*
- *“La gente de la región tiene más miedo a la vacuna que al #coronavirus y ninguna mujer quiere ponérsela. Huyen de ella como de la peste. <https://t.co/ficSQynvwy>”*

Después de preprocesar los Tweets de acuerdo con lo mencionado en la sección de limpieza de los textos, quedarían de la forma:

- “*ejército estadounidense ofrecer proporcionar vacuna contra coronavirus johnson & johnson tropa surcoreana decir lunes ministro defensa mientras corea sur luchar contra escasez inyección covid-19*”.
- “*cada día haber evidencia vacuna contra coronavirus también frenar transmitir todo grupo edad aullar*”.
- “*helicóptero cargar trabajador salud dosis vacuna contra coronavirus despegar labrea parte sur Amazonas ver tv*”

Con la finalidad de hacer los cálculos necesarios para obtener la matriz de representación de textos, es necesario conocer cuantas veces aparecen las palabras en los textos o vocabulario, en la Tabla 10 se puede observar esta relación para la muestra de Tweets.

Tabla 10

Frecuencias de palabras en muestra de Tweets

Palabra	Frecuencia
<i>ejército</i>	1
<i>estadounidense</i>	1
<i>ofrecer</i>	1
<i>proporcionar</i>	1
<i>vacuna</i>	3
<i>coronavirus</i>	3
<i>contra</i>	3
<i>johnson</i>	2
<i>&</i>	1
<i>tropa</i>	1
<i>surcoreana</i>	1
<i>decir</i>	1
<i>lunes</i>	1
<i>ministro</i>	1
<i>defensa</i>	1
<i>mientras</i>	1
<i>corea</i>	1
<i>sur</i>	2
<i>luchar</i>	1
<i>escasez</i>	1
<i>inyección</i>	1
<i>covid-19</i>	1
<i>cada</i>	1
<i>haber</i>	1
<i>día</i>	1

<i>evidencia</i>	1
<i>también</i>	1
<i>frenar</i>	1
<i>transmitir</i>	1
<i>todo</i>	1
<i>grupo</i>	1
<i>edad</i>	1
<i>aullar</i>	1
<i>helicóptero</i>	1
<i>cargar</i>	1
<i>trabajador</i>	1
<i>salud</i>	1
<i>dosis</i>	1
<i>despegar</i>	1
<i>parte</i>	1
<i>labrea</i>	1
<i>amazonas</i>	1
<i>ver</i>	1
<i>tv</i>	1

El siguiente paso es construir la matriz donde se relaciona la frecuencia de términos por frases (que para el presente caso serían los Tweets). La manera en que se genera es teniendo en cuenta todas las palabras en la muestra y la cantidad de Tweets, entonces, para cada Tweet se relaciona el número de veces que aparece el termino asociado, en la Tabla 11 se puede observar el resultado para los Tweets de ejemplo.

Tabla 11

Matriz frecuencia termino-Tweet

Palabra	Tweet 1	Tweet 2	Tweet 3
ejército	1	0	0
estadounidense	1	0	0
ofrecer	1	0	0
proporcionar	1	0	0
vacuna	1	1	1
coronavirus	1	1	1
contra	1	1	1
johnson	2	0	0
&	1	0	0
tropa	1	0	0
surcoreana	1	0	0
decir	1	0	0
lunes	1	0	0
ministro	1	0	0

defensa	1	0	0
mientras	1	0	0
corea	1	0	0
sur	1	0	0
luchar	1	0	0
escasez	1	0	0
inyección	1	0	0
covid-19	1	0	0
cada	0	1	0
haber	0	1	0
día	0	1	0
evidencia	0	1	0
también	0	1	0
frenar	0	1	0
transmitir	0	1	0
todo	0	1	0
grupo	0	1	0
edad	0	1	0
aullar	0	1	0
helicóptero	0	0	1
cargar	0	0	1
trabajador	0	0	1
salud	0	0	1
dosis	0	0	1
despegar	0	0	1
parte	0	0	1
labrea	0	0	1
amazonas	0	0	1
ver	0	0	1
tv	0	0	1

Finalmente se construye la matriz con el factor $tf-idf$, su proceso de generación es similar al de la Tabla 11, solamente que los campos se llenan calculado su respectivo valor mediante el planteamiento de la ecuación (1). Para la palabra ejército y el Tweet número 1, $f_{w,d}$ sería 1, por otro lado $\log(|D|/f_{w,D})$ es 0,47712125. En la Tabla 12 se puede observar los valores para todos los Tweets y palabras.

Tabla 12

Factor tf-idf Tweets de prueba

Palabra	Tweet 1	Tweet 2	Tweet 3
ejército	0,47712125	0	0
estadounidense	0,47712125	0	0
ofrecer	0,47712125	0	0
proporcionar	0,47712125	0	0
vacuna	0	0	0
coronavirus	0	0	0
contra	0	0	0
johnson	0,35218252	0	0
&	0,47712125	0	0
tropa	0,47712125	0	0
surcoreana	0,47712125	0	0
decir	0,47712125	0	0
lunes	0,47712125	0	0
ministro	0,47712125	0	0
defensa	0,47712125	0	0
mientras	0,47712125	0	0
corea	0,47712125	0	0
sur	0,47712125	0	0
luchar	0,47712125	0	0
escasez	0,47712125	0	0
inyección	0,47712125	0	0
covid-19	0,47712125	0	0
cada	0	0,47712125	0
haber	0	0,47712125	0
día	0	0,47712125	0
evidencia	0	0,47712125	0
también	0	0,47712125	0
frenar	0	0,47712125	0
transmitir	0	0,47712125	0
todo	0	0,47712125	0
grupo	0	0,47712125	0
edad	0	0,47712125	0
aullar	0	0,47712125	0
helicóptero	0	0	0,47712125
cargar	0	0	0,47712125
trabajador	0	0	0,47712125
salud	0	0	0,47712125
dosis	0	0	0,47712125
despegar	0	0	0,47712125
parte	0	0	0,47712125
labrea	0	0	0,47712125
amazonas	0	0	0,47712125

ver	0	0 0,47712125
tv	0	0 0,47712125

6.2.2 Entrenamiento de algoritmos

Mediante las librerías *scikitlearn*, *Numpy* y *joblib* de *python*, se abordaron los diferentes algoritmos de *machine learning* propuestos para el presente trabajo, donde los datos de entrenamiento debían transformarse en dos listas de *Numpy*, una correspondiente a los ejemplos de entrenamiento y otra para las etiquetas asignadas previamente. La implementación en código para cada uno estos, consistía en importar las clases asociadas a los algoritmos y crear sus respectivas ejemplificaciones (o instancias), después mediante la función *fit* se entrenaron los modelos. A continuación, se ilustra el proceso para cada uno de estos:

6.2.2.1 Redes Neuronales

Se usó la variante de perceptrón multicapa, con la clase *MLPClassifier*, que viene con diversos valores por defecto, aquellos que se modificaron para llevar a cabo las diferentes pruebas fueron: número de capas ocultas y neuronas dentro de estas que se pasan como una tupla en el parámetro *hidden_layer_size*, y las épocas de entrenamiento con *max_iter*. La forma en la que se implementó con *Scikitlean* se puede ver en Figura 14.

Figura 14

Perceptrón multicapa de Scikitlearn

```
from sklearn.neural_network import MLPClassifier
import random
from sklearn.preprocessing import StandardScaler
import numpy as np
from joblib import dump, load

def training_neural_network(self, matrix_training, name, hidden_layers_config, epochs):
    data_training = [x[0] for x in matrix_training]
    labels = [x[1] for x in matrix_training]
    X = np.asarray(data_training, float)
    y = labels
    clf = MLPClassifier(solver='lbfgs', alpha=1e-5,
                       hidden_layer_sizes=hidden_layers_param,
                       random_state=1, max_iter=epochs)
    clf.fit(X, y)
    final_name = '%s.joblib' % (name)
    dump(clf, name)
    return clf
```

Nota. Fuente: Elaboración propia.

6.2.2.2 Naive Bayes

Fue implementada la variante de **MultinomialNB**, que es utilizada en la clasificación de textos, con distribuciones multinomiales. En la Figura 15 se observa la forma de entrenamiento.

Figura 15

Algoritmo Bayesiano Multinomial

```
from sklearn.preprocessing import StandardScaler
import numpy as np
from joblib import dump, load
from sklearn.naive_bayes import MultinomialNB

def training_naive_bayes(self, matrix_training, name):
    data_training = [x[0] for x in matrix_training]
    labels = [x[1] for x in matrix_training]
    X=np.asarray(data_training,float)
    y = labels
    clf = MultinomialNB()
    clf.fit(X, y)
    example=np.array(data_training[0])
    example=example.reshape(1, -1)
    example.shape
    final_name = '%s.joblib' % (name)
    dump(clf,name)
    return clf
```

Nota. Fuente: Elaboración propia.

6.2.2.3 Máquinas de soporte vectorial

Con la finalidad de hacer una clasificación de varias clases, el módulo SVM de Scikitlearn contiene un parámetro denominado ‘ovo’ de ‘one vs one’, que permite trabajar con múltiples categorías, esto se puede ver en la Figura 16.

Figura 16

Entrenamiento SVM con variante “One vs One”

```
from sklearn import svm
import random
from sklearn.preprocessing import StandardScaler
import numpy as np
from joblib import dump, load

def train_svm_machine(self, matrix_training, name):
    data_training = [x[0] for x in matrix_training]
    labels = [x[1] for x in matrix_training]
    X = np.asarray(data_training, float)
    y = labels
    clf = svm.SVC(decision_function_shape='ovo')
    clf.fit(X, y)
    example = np.array(data_training[0])
    example = example.reshape(1, -1)
    example.shape
    final_name = '%s.joblib' % (name)
    dump([clf, name])
    return clf
```

Nota. Fuente: Elaboración propia.

6.2.3 Cálculo de métricas

Con el fin de calcular y obtener los diferentes valores que dan los indicadores de desempeño *accuracy*, *precisión*, *recall* y *f1* de los algoritmos, se usó la opción de *metrics* brindada por *scikit learn*, en esta se usan dos listas: una con las etiquetas obtenidas en la clasificación y otra que contiene las salidas esperadas. La librería se encarga de calcular para cada una de las métricas sus respectivos valores, en el caso de la matriz de confusión, es construida con su respectiva dimensionalidad dependiendo del número de clases, en la Figura 17 se ilustra la forma en la que se calcula cada métrica.

Figura 17

Cálculo de métricas

```
@classmethod
def calculate_confusion_matrix(self,true_labels,predicted_labels):
    m_confusion = confusion_matrix(true_labels, predicted_labels)
    return m_confusion.tolist()

@classmethod
def calculate_acuracy(self,true_labels,predicted_labels):
    return accuracy_score(true_labels,predicted_labels)

@classmethod
def calculate_f1(self,true_labels,predicted_labels):
    return f1_score(y_true, y_pred, average='macro')

@classmethod
def get_recall(self,true_labels,predicted_labels):
    return recall_score(true_labels,predicted_labels, average='macro')
```

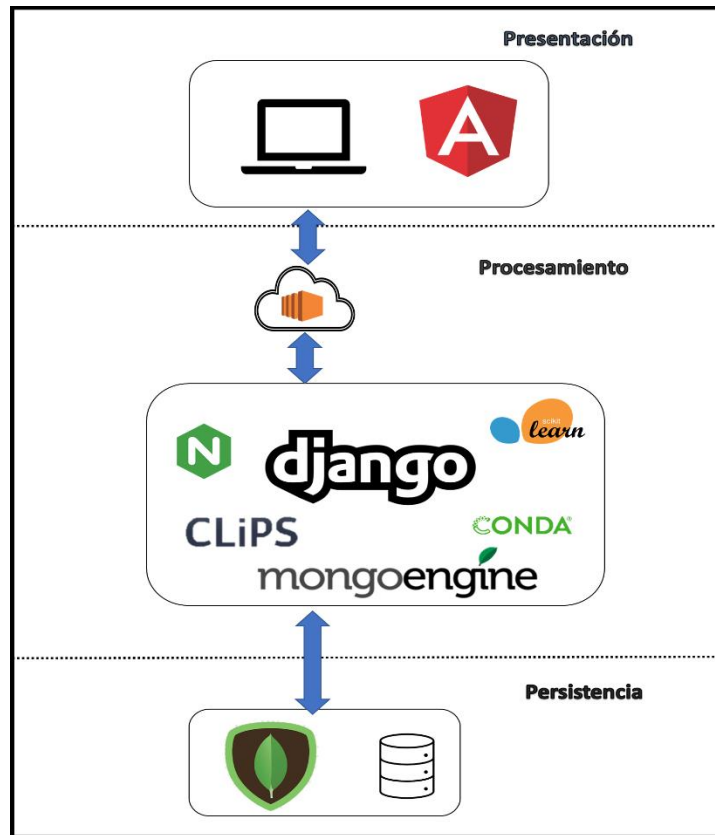
Nota. Fuente: Elaboración propia.

6.3 Desarrollo del aplicativo

Dado de que la metodología implementada para el desarrollo del software fue Scrum, al final de cada una de las iteraciones y Sprints, se obtuvo un aplicativo que brinda la posibilidad de llevar a cabo el proceso de análisis de sentimientos para textos en español, desde la carga de los archivos de Tweets en Excel, hasta el entrenamiento de algoritmos, incluyendo los cálculos de sus respectivas métricas.

Figura 18

Arquitectura general del sistema



Nota. Fuente: Elaboración propia.


Como se observa en la en la Figura 18, a nivel general hay tres capas del sistema que son de presentación, servicio y persistencia, de las cuales se abarcarán más detalles a continuación:

6.3.1 Capa de presentación o cliente web

Consiste en una *single page application* SPA desarrollado en angular, que le brinda al usuario las opciones para llevar a cabo completamente el proceso de análisis de sentimientos, desde la carga de los textos, hasta el entrenamiento y prueba de los diferentes algoritmos de *machine learning*, obteniendo también las distintas métricas de rendimiento. En la Figura 19 se observa la opción de carga del archivo de Excel, que posteriormente se enviara al servidor para el preprocesamiento de textos.

Figura 19

Opción de carga de Archivos



Cargar Archivo Excel

Eliminar palabras con:

☐ @

☐ #

Seleccionar archivo Ningún archi... seleccionado

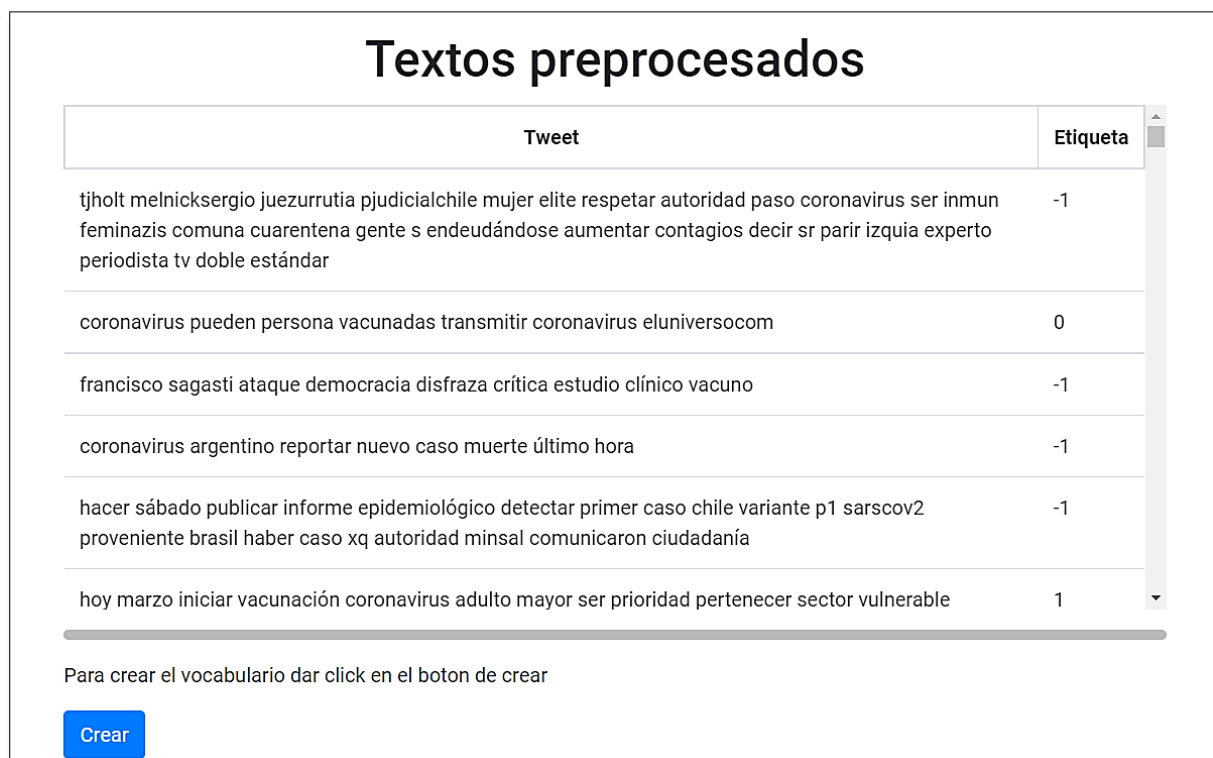
Leer

Nota. Fuente: Elaboración propia.

Una vez el servidor finaliza el preprocesamiento de los textos, retorna una lista de estos con la etiqueta asignada inicialmente, en la Figura 20 se muestra la forma en la que se observan los tweets preprocesados.

Figura 20

Vista de textos preprocesados



Tweet	Etiqueta
tjholt melnicksergio juezurrutia pjudicialchile mujer elite respetar autoridad paso coronavirus ser inmun feminazis comuna cuarentena gente s endeudándose aumentar contagios decir sr parir izquia experto periodista tv doble estándar	-1
coronavirus pueden persona vacunadas transmitir coronavirus eluniversocom	0
francisco sagasti ataque democracia disfraz crítica estudio clínico vacuno	-1
coronavirus argentino reportar nuevo caso muerte último hora	-1
hacer sábado publicar informe epidemiológico detectar primer caso chile variante p1 sarscov2 proveniente brasil haber caso xq autoridad minsal comunicaron ciudadanía	-1
hoy marzo iniciar vacunación coronavirus adulto mayor ser prioridad pertenecer sector vulnerable	1

Para crear el vocabulario dar click en el boton de crear

Crear

Nota. Fuente: Elaboración propia.

Una vez se han obtenido los textos preprocesados el sistema brinda al usuario la posibilidad de construir el vocabulario a partir de estos, adicionalmente la opción de eliminar palabras por frecuencia mínima de aparición, esto con la finalidad de reducir la dimensionalidad del problema, en la Figura 21 se muestran las opciones que el sistema brinda al usuario.

Figura 21





Configuración de vocabulario

Vocabulario

Tamaño

354

eliminar

Palabra	Frecuencia	Eliminar
cooperativaencasa	63	
haber	590	
chino	153	
reporte	62	
vacunación	350	
hoy	382	
fin	61	

Copiar datos

copiar

Obtener datos entrenamiento

Obtener

Nota. Fuente: Elaboración propia.

Además, se pueden eliminar aquellas palabras que no se desea incluir aun después de preprocesar los textos y filtrarlos por frecuencia, solamente dando clic en el icono de eliminar en la fila del término correspondiente. Una vez completado este proceso de filtrado, se procede a la construcción de las matrices de representación de textos, que serán los datos de entrenamiento para los diferentes algoritmos de *machine learning*. En la Figura 22 se puede observar el resultado que se obtiene.

Figura 22

Datos de entrenamiento obtenidos

Datos de entrenamiento													
cooperativaencasa	haber	chino	reporte	vacunación	hoy	fin	coronavirus	vacunar	diario	tres	campana	cama	fallecim
0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	2	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0

Nota. Fuente: Elaboración propia.

Adicionalmente el sistema despliega un formulario que solicita el nombre y el tipo de algoritmo a entrenar, para el caso particular de las redes neuronales, se cargan unos cuadros de texto adicionales para ingresar la información del número de capas ocultas, cantidad de neuronas dentro de estas, y épocas de entrenamiento. En la Figura 23 se puede ver el formulario de carga para entrenar una máquina de soporte vectorial.

Figura 23

Formulario para entrenar algoritmos

Entrenar Algoritmos	
Tipo	svm
Nombre	
Entrenar	

Nota. Fuente: Elaboración propia.

Una vez finalizado el proceso de entrenamiento se despliega la información del algoritmo y un cuadro de texto para realizar pruebas ingresando manualmente un Tweet, también brinda la opción de cargar un archivo de Excel con Tweets de prueba. En la Figura 24 se muestra la información relacionada con el algoritmo entrenado.

Figura 24

Panel de pruebas

Algoritmo Entrenado

Nombre Algoritmo:
svm_test_8000

Tipo:
svm

Tamaño vocabulario
354

Seleccionar archivo

Ningún archivo seleccionado

Palabra	Etiqueta	Esperado
@melnicksergio @JuezUrrutia @PJudicialChile 8M, mujeres de elite del 1% no respetan la autoridad. Qué paso con coronavirus?, Son inmunes las feminazis?. Comunas en cuarentena, gente s/trabajo, estado endeudándose. Cuando aumenten los contagios ¿Qué dirá Sr. París, Izquierda, "los expertos" y periodistas TV?. Doble estándar.	1	<input type="text" value="-1"/>
#8Mar #Coronavirus ¿Pueden las personas vacunadas transmitir el coronavirus? https://t.co/BUe58zgFG - @eluniversocom https://t.co/rVB6huR504	-1	<input type="text" value="0"/>
Francisco Sagasti: Ataque contra la democracia se "disfraza" de críticas a un estudio clínico de una vacuna. https://t.co/zDsa8NAhe8 https://t.co/m01xzBeIMs	-1	<input type="text" value="-1"/>
Coronavirus en Argentina: reportan 2.922 nuevos casos y 10 muertes en las últimas 24 horas https://t.co/9ryfV1TpL5	-1	<input type="text" value="-1"/>

Nota. Fuente: Elaboración propia.

Finalmente, para el algoritmo entrenado y con las pruebas realizadas, se pueden calcular las diferentes métricas asociadas a este mismo, esto lo hace tomando los valores esperados para la salida en cada prueba contra el valor de obtenido del texto retornado por el algoritmo. En la Figura 25 se ilustra la información que el aplicativo despliega en las métricas.

Figura 25

Métricas obtenidas de los textos

calcular eficacia		
<input type="button" value="Enviar"/>		
-1	0	1
1090	0	82
220	0	4
560	0	43
Acuraccy		
0.5667833916958479		
F1		
0.5667833916958479		
Recall		
0.5667833916958479		
Preision		
0.5667833916958479		

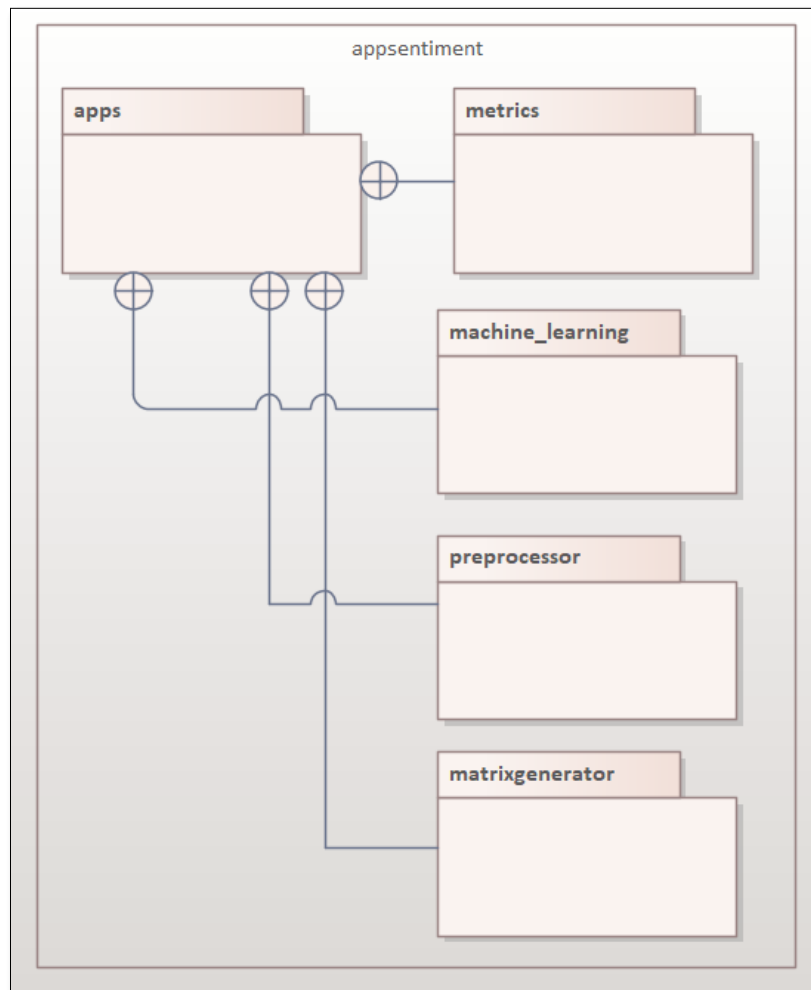
Nota. Fuente: Elaboración propia.

6.3.2 Capa de Procesamiento

Con la finalidad de recibir y procesar todas las peticiones que venían desde la capa de presentación, se implementó un servidor REST en Django, adicionalmente se conectaba a la capa de persistencia para guardar la información de los algoritmos entrenados. La Figura 26 se muestra cómo se encuentra estructurada esta capa mediante un diagrama de paquetes.

Figura 26

Módulos servidor



Nota. Fuente: Elaboración propia.

Dado que un proyecto en Django consiste en un conjunto de apps, en el presente proyecto se crearon las siguientes:

- ***Preprocessor***: Es la parte encargada del preprocesamiento de los Tweets enviados por cada petición que viene de parte del cliente WEB.
- ***Matrixgenerator***: Construye las matrices de representación de textos a partir de los vocabularios y Tweets.
- ***Machine_learning***: Tiene como función el uso de la librería scikit-learn para entrenar y probar los diferentes algoritmos de *machine learning*, guardando la información de estos en la base de datos, además hace un volcado de la información en un archivo con extensión. `.joblib`.

- **Metrics:** Encargado de obtener las métricas de los diferentes algoritmos, recibe los datos de las pruebas realizadas y los valores esperados a manera de arreglos y genera la información solicitada a partir de estas.

6.3.3 Capa de persistencia

En esta se usan bases de datos no relacionales con la finalidad de almacenar la información relacionada con los algoritmos. Los campos almacenados de cada algoritmo se pueden ver en la Tabla 13.

Tabla 13

Campos para colección del Algoritmo

Campo	Tipo
Nombre	String
Id	String
Cantidad datos	Number
Tipo	String
Archivo	String

El campo Archivo hace referencia al nombre del fichero con extensión. *joblib* que se genera, carga y prueba dependiendo de la acción solicitada desde la capa de presentación.

7 Pruebas realizadas

Los escenarios empleados en el presente trabajo de grado variaron a partir de la obtención de distintas representaciones matriciales de los textos, mediante la modificación del vocabulario (la muestra total de las palabras), cantidad de datos de entrenamiento, y el enfoque para la asignación de polaridad para los textos. En la Tabla 14 se muestran los escenarios configurados para los tweets etiquetados de forma manual.

Tabla 14

Escenarios de prueba con 3 Etiquetas

No Escenario	Cantidad tweets entrenamiento	Vocabulario Obtenido	Vocabulario Inicial	Frecuencia mínima Palabra
1	8000	231	19807	50
2	6000	190	15215	50
3	4000	186	11573	40
4	2000	57	7061	40

En la Tabla 15 se relacionan los diferentes vocabularios obtenidos para el caso de los tweets etiquetados automáticamente.

Tabla 15

Escenarios de prueba con 2 Etiquetas

No Escenario	Cantidad tweets entrenamiento	Vocabulario Obtenido	Vocabulario Inicial	Frecuencia mínima Palabra
1	12000	535	30664	50
2	9000	410	25001	50
3	6000	294	18770	45
4	3000	141	11547	40

Como se observa en Tabla 14 y Tabla 15, al comparar el tamaño de los vocabularios obtenidos a partir de la inicial, estos difieren en grandes proporciones y se realizó con la finalidad de

reducir la dimensionalidad del problema y usar palabras que en las muestras resultantes aparezcan en la mayor cantidad de tweets posibles.

8 Resultados Obtenidos

En esta sección se relacionan las diferentes métricas de rendimiento obtenidas de los algoritmos de *machine learning* para los escenarios de prueba y enfoques de etiquetado planteados (manual y automático).

8.1 Resultados para etiquetado manual

Desde la Tabla 16 hasta la Tabla 19, se muestran los resultados de las métricas obtenidas para el enfoque manual de etiquetado. En la Tabla 16 se pueden ver los resultados de 8000 Tweets y 231 palabras.

Tabla 16

Resultados 8000 Tweets

	Acuraccy	F1	Recall	Precision
Naive bayes	0.401080	0.401080	0.401080	0.401080
Red Neuronal	0.404684	0.404684	0.404684	0.404684
SVM	0.53496	0.53496	0.53496	0.53496

Para los 6000 tweets se ilustra el rendimiento de los algoritmos en la Tabla 17, con un total de 190 palabras.

Tabla 17

Resultados 6000 Tweets

	Acuraccy	F1	Recall	Precision
Naive bayes	0.43690	0.43690	0.43690	0.43690
Red Neuronal	0.435727	0.435727	0.435727	0.435727
SVM	0.5994	0.599424	0.599424	0.599424

En el caso de los 4000 Tweets se obtuvo un tamaño de vocabulario muy cercano al de los 6000, en la Tabla 18 se muestran los resultados del respectivo escenario.

Tabla 18

Resultados 4000 Tweets

	Acuraccy	F1	Recall	Precision
Naive bayes	0.41310327	0.41310327	0.41310327	0.41310327
Red Neuronal	0.3895973	0.3895973	0.3895973	0.3895973
SVM	0.545636	0.545636	0.545636	0.545636

Ya para el último escenario de etiquetado manual con un total de 2000 palabras de vocabularios, se observa el rendimiento de los algoritmos en la Tabla 19.

Tabla 19

Resultados 2000 Tweets

	Acuraccy	F1	Recall	Precision
Naive bayes	0.5752726	0.5752726	0.5752726	0.5752726
Red Neuronal	0.570785	0.570785	0.570785	0.570785
SVM	0.59479	0.59479	0.59479	0.59479

8.2 Resultados para etiquetado automático

A continuación, se muestran los resultados obtenidos de las métricas de rendimiento de los algoritmos para el enfoque de etiquetado automático. En la Tabla 20 se relacionan los datos para 12.000 tweets.

Tabla 20

Resultados 12000 tweets

	Acuraccy	F1	Recall	Precision
Naive bayes	0.694227	0.694227	0.694227	0.694227
Red Neuronal	0.5695	0.5695	0.5695	0.5695
SVM	0.862878	0.862878	0.862878	0.862878

Para 9000 tweets como escenario, en la Tabla 21 se relacionan los resultados.

Tabla 21

Resultados 9000 tweets

	Acuraccy	F1	Recall	Precision
Naive bayes	0.825313	0.825313	0.825313	0.825313
Red Neuronal	0.818090	0.818090	0.818090	0.818090
SVM	0.862095	0.862095	0.862095	0.862095

En el caso de los 6000, en la Tabla 22 se relacionan los resultados obtenidos.

Tabla 22

Resultados 6000 tweets 2 etiquetas

	Acuraccy	F1	Recall	Precision
Naive bayes	0.795632	0.795632	0.795632	0.795632
Red Neuronal	0.755125	0.755125	0.755125	0.755125
SVM	0.86131021	0.86131021	0.86131021	0.86131021

Finalmente, en la Tabla 23 se relacionan los resultados para el caso de etiquetado automático y 3000 tweets.

Tabla 23

Resultados 3000 tweets

	Acuraccy	F1	Recall	Precision
Naive bayes	0.806064	0.806064	0.806064	0.806064
Red Neuronal	0.797067	0.797067	0.797067	0.797067
SVM	0.863378	0.863378	0.863378	0.863378

9 Análisis de Resultados

- Los algoritmos entrenados a partir de los tweets etiquetados automáticamente presentaron un mayor desempeño comparados con los que se entrenaron con etiquetas asignadas manualmente.
- En todos los escenarios planteados se observa que el algoritmo SVM es el que tiene un mayor rendimiento.
- A pesar de las variaciones de la muestra y el vocabulario para los escenarios de etiquetado manual, se observa que el SVM se mantiene en un rango de 53 a 59% de efectividad en el enfoque Manual.
- Para el enfoque automático, se observa que el algoritmo SVM mantuvo un porcentaje de efectividad superior al 86% en todos los escenarios planteados.
- Las redes neuronales en el etiquetado manual mostraron bajo rendimiento ya que en algunos escenarios no superaron el 40% de efectividad.
- Los algoritmos de redes neuronales y Bayes presentan una disminución en su rendimiento cuando la cantidad de datos y vocabulario incrementaban. Esto se observó en los dos enfoques de etiquetado.
- Una de las posibles causas del bajo rendimiento de los algoritmos posiblemente se dio por la selección de características, que para el presente proyecto se basó en la frecuencia de aparición de términos.
- Dado el bajo desempeño de los algoritmos de Redes neuronales y Bayes para algunos escenarios, sería conveniente usar variantes de estos que puedan mejorar los resultados en otros casos de aplicación.
- Aplicar un modelo de selección de características más sofisticado puede mejorar el desempeño de los algoritmos de *machine learning*.

10 Conclusiones

- La comparación de algoritmos de *Machine Learning* en la clasificación de tweets extraídos de esta red social acerca de la COVID-19, fue posible gracias a las diferentes herramientas lingüísticas, computacionales y matemáticas que permitieron la transformación de un conjunto de textos a matrices numéricas con la finalidad de usarlos como datos de entrenamiento, adicionalmente el cálculo de métricas permitió conocer cuál algoritmo presentaba un mejor desempeño, que para el presente trabajo fueron las Maquinas de soporte Vectorial.
- Aunque el lenguaje para el ser humano tiene sus reglas, connotaciones, contextos y las personas emplean muchos procesos complejos a la hora de comunicarse de manera intuitiva, dentro del marco del procesamiento de lenguaje natural y con la finalidad de reducir el costo de tratamiento computacional de un problema, es necesario que los algoritmos trabajen con representaciones mapeadas con el fin de simplificar procesos y obtener mejores resultados, lo que implica trabajar con dominios de problemas restringidos y omitir información lingüística que puede llegar a ser útil.
- La construcción de recursos léxicos es de suma importancia en el análisis de sentimientos, ya que reducen significativamente la dimensionalidad de un problema en cuestión, al transformar las palabras a su respectiva raíz léxica. También brinda la capacidad de llevar a cabo un preprocesamiento usando enfoques lingüísticos, donde la limitación se evidencia en la cantidad de información consignada en estos, a pesar de que cuentan con un numero grande de registros, son muy pocos comparándolos con los elementos gramaticales de una lengua.
- La Ingeniería en sistemas juega un rol importante para llevar a cabo este tipo de proyectos, ya que debe trabajar de manera conjunta con expertos en diferentes áreas de conocimiento como matemáticas, lingüística, programación, diseño de software e inteligencia artificial, con la finalidad de construir soluciones en el ámbito del procesamiento de lenguaje natural.

11 Trabajo Futuro

- Usar nuevos enfoques para la construcción de recursos léxicos, que permitan una estandarización y la adición de nuevos registros, en los que se pueda consolidar toda la información gramatical posible de una lengua, brindando facilidad y velocidad a la hora de consultarla.

Bibliografía

- Hootsuite. (2019). *Digital in 2019 - Social Media Marketing & Management Dashboard - Hootsuite*. [en línea] Available at: <https://hootsuite.com/pages/digital-in-2019> [Accessed 23 Mar. 2019].
- Blog Hootsuite. 2020. Estadísticas Generales De Redes Sociales. [online] Available at: <https://blog.hootsuite.com/es/125-estadisticas-de-redes-sociales/> [Accessed 29 November 2020].
- Bloggin Zenith. (2013). *El triunfo de Obama en Internet: caso de estudio de las campañas de 2008 y 2012 (II) - Bloggin Zenith*. [online] Available at: <https://blogginzenith.zenithmedia.es/el-triunfo-de-obama-en-internet-caso-de-estudio-de-las-campanas-de-2008-y-2012-ii/> [Accessed 22 Mar. 2019].
- Baeza-Yates & Ribeiro-Neto Berthier, Modern Information Retrieval. Addisonwesley, Wokingham, UK, 1999.
- Liu, B. (2006). Web Data Mining, chapter Opinion Mining. Springer.
- Alfaro R., Allende H. (2010), Text Representation in Multi-label Classification: Two New Input Representations 10th International Conference on Adaptive and Natural Computing Algorithms (ICANNGA'11).
- Li, J., & Qiu, L. (2017). A sentiment analysis method of short texts in microblog. Proceedings - 2017 IEEE International Conference on Computational Science and Engineering and IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, CSE and EUC 2017, 1, 776–779. doi: 10.1109/CSEEUC.2017.153.
- Salinca, A. (2016). Business Reviews Classification Using Sentiment Analysis. Proceedings - 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2015, 247–250. doi: 10.1109/SYNASC.2015.46.
- Woldemariam, Y. (2016). Sentiment analysis in a cross-media analysis framework. Proceedings of 2016 IEEE International Conference on Big Data Analysis, ICBDA 2016. doi: 10.1109/ICBDA.2016.7509790.
- About.twitter.com. (2019). *About*. [online] Available at: https://about.twitter.com/en_us.html [Accessed 13 Apr. 2019].
- AITCHINSON, Jean. “Animales que intentan hablar. ¿Es el lenguaje algo exclusivo de los humanos?”. En *El mamífero articulado*. Madrid, Alianza, 1992, págs. 39-69
- Liu, B. (2006). Web Data Mining, chapter Opinion Mining. Springer.
- Baeza-Yates & Ribeiro-Neto Berthier, Modern Information Retrieval. Addisonwesley, Wokingham, UK, 1999.
- Augusto Cortez Vásquez, Hugo Vega Huerta, J. P. Q. (2009). Procesamiento de lenguaje natural. *Revista de Ingeniería de Sistemas e Informática*, 6, 10
- Jiménez Moscovitz, L., & Rengifo Rengifo, P. (2010). AL INTERIOR DE UNA MÁQUINA DE SOPORTE VECTORIAL. *Revista De Ciencias*, 14, 73-85.
- BAYES, T. (1764). LII. A demonstration of the second rule in the essay towards the solution of a problem in the doctrine of chances, published in the *Philosophical Transactions*, Vol. LIII. Communicated by the Rev. Mr. Richard Price, in a letter to Mr. John Canton, M.A. F. R. S. *Philosophical Transactions Of The Royal Society Of London*, 54, 296-325. doi: 10.1098/rstl.1764.0050.

- Sucar, L. E. (2008). Clasificadores Bayesianos: de Datos a Conceptos. *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 1–2.
- Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5), 988–999. <https://doi.org/10.1109/72.788640>.
- Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5), 988–999. <https://doi.org/10.1109/72.788640>.
- Futurizable. (2017). Estado del arte en el desarrollo de chatbots a nivel mundial. 2020, noviembre 24, de s | Futurizable Recuperado de <https://futurizable.com/chatbot/>
- Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, & \Edouard Duchesnay (2011). Scikit-learn: Machine Learning in Python *Journal of Machine Learning Research*, 12(85), 2825-2830.
- Matt Ahlgren. (2020). 50 + ESTADÍSTICAS Y HECHOS DE TWITTER PARA 2020. 2020, noviembre 28, de websitehostingrating Recuperado de <https://www.websitehostingrating.com/es/twitter-statistics/>.
- Ashraf M., K., Eibe, F., Bernhard, P. and Geoffrey, H., n.d. Multinomial Naive Bayes for Text Categorization Revisited. Department of Computer Science, University of Waikato, Hamilton, New Zealand, p.12.
- Dhanendran Anthony 2014. Qué tan rápido puede uno llegar a leer - BBC News Mundo. [online] Available at: <https://www.bbc.com/mundo/noticias/2014/09/140925_vert_fut_leer_super_velocidad_np> [Accessed 28 February 2021].
- Pérez Vera, S., 2017. Análisis Y Clasificación De Textos Con Técnicas Semi Supervisadas Aplicado A Área Atención Al Cliente. pregrado. PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO.
- Chordá, F. (2000). Historia y Lenguaje. "La civilización de las expresiones variadas", de Mihai Nadin: Una síntesis. *Historia, Antropología Y Fuentes Orales*, (23), 61-81. Retrieved March 2, 2021, from <http://www.jstor.org/stable/27753021>.
- Hernandez M., & Gomez J. (2013). Aplicaciones de Procesamiento de Lenguaje Natural. *Revista Politécnica*, 32(1), 87-96. https://revistapolitecnica.epn.edu.ec/ojs2/index.php/revista_politecnica2/article/view/32/pdf.
- Cortez Vásquez, A., Vega Huerta, H., & Pariona Quispe, J. (s. f.). Procesamiento de lenguaje natural. Recuperado 6 de marzo de 2021, de https://sisbib.unmsm.edu.pe/bibvirtual/Publicaciones/risi/2009_n2/v6n2/a06v6n2.pdf
- Moreno, A. (s.f.). Procesamiento del lenguaje natural ¿qué es? Instituto de Ingeniería del Conocimiento. Recuperado 6 de marzo de 2021 de <https://www.iic.uam.es/inteligencia/que-es-procesamiento-del-lenguaje-natural/>
- Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4), 1093-1113. <https://doi.org/10.1016/j.asej.2014.04.011>.
- Barredo A. (24 de abril del 2019). Microsoft decide no eliminar Paint de Windows. *La Vanguardia*. Recuperado el 9 de marzo del 2021 de <https://www.lavanguardia.com/tecnologia/20190424/461836854094/paint-windows-10.html>

- Organización mundial de la salud. (9 de marzo de 2021). *Preguntas y respuestas sobre la enfermedad por coronavirus (COVID-19)*. <https://www.who.int/es/emergencias/diseases/novel-coronavirus-2019/advice-for-public/q-a-coronaviruses>.
- Marín, R. (s.f.). El tratamiento computacional del léxico y sus aplicaciones. CNRS - Université de Lille.
- Ariel Pérez Vera. (abril del 2017). Análisis y Clasificación de Textos con Técnicas Semi Supervisadas Aplicado a Área Atención al Cliente (tesis de pregrado). Pontificia Universidad Católica de Valparaíso, Chile.
- Real Academia Española. (s.f.). Cultura. En Diccionario de la lengua española. Recuperado el 27 de marzo de 2021.
- Cifuentes, Valery (octubre del 2018). Así es el salario de los profesores en las cinco mejores universidades del país. *La República*. <https://www.larepublica.co/alta-gerencia/asi-es-el-salario-de-los-profesores-en-las-mejores-universidades-del-pais-2780195>.
- Andrew, S. (2020, 21 abril). Este es el impacto devastador de la pandemia de coronavirus en cifras. CNN. <https://cnnespanol.cnn.com/2020/04/21/este-es-el-impacto-devastador-de-la-pandemia-de-coronavirus-en-cifras/>.
- Banco Mundial. (2020). La COVID-19 (coronavirus) hunde a la economía mundial en la peor recesión desde la Segunda Guerra Mundial. (2020, 8 junio). World Bank. <https://www.bancomundial.org/es/news/press-release/2020/06/08/covid-19-to-plunge-global-economy-into-worst-recession-since-world-war>.
- Ministerio de Salud. (s. f.). Vacunación contra COVID-19. [minsalud.gov.co](https://www.minsalud.gov.co). Recuperado 28 de marzo de 2021, de <https://www.minsalud.gov.co/salud/publica/Vacunacion/Paginas/Vacunacion-covid-19.aspx>.
- Younis, E. (febrero del 2015). Sentiment Analysis and Text Mining for Social Media Microblogs using Open-Source Tools: An Empirical Study. *International Journal of Computer Applications*.
Guarnizo p. & Monroy a. (2020). Implementación de un modelo de análisis de sentimientos con Respecto a la JEP basado en minería de datos en twitter. *Universidad Católica De Colombia*.
- Rodriguez, L. (2019). Análisis de datos de sentimientos enfocados al servicio de Transporte masivo Transmilenio s.a aplicando tecnologías big data. *Fundación universitaria los libertadores*.
- Beltran C. & Barbona I. (2017). Una revisión de las técnicas de clasificación supervisada en la clasificación automática de textos. *Universidad Nacional de Rosario, Argentina*.
- Secretaria de Estado de Sanidad (15 de enero de 2021). *Enfermedad por coronavirus, COVID-19*. <https://www.mscbs.gob.es/profesionales/saludPublica/ccayes/alertasActual/nCov/documentos/ITCoronavirus.pdf>
- Maguiña Vargas, C., Gastelo Acosta, R. and Tequen Bernilla, A., 2020. El nuevo Coronavirus y la pandemia del Covid-19. *Revista Médica Herediana*, 31(2), pp.125-131.
- Unión europea (2021). *Estrategia de las vacunas contra el coronavirus*. https://ec.europa.eu/info/live-work-travel-eu/coronavirus-response/public-health/eu-vaccines-strategy_es.

- GetApp. (s. f.). Natural Language Processing (NLP). Recuperado 24 de abril de 2021, de <https://www.getapp.com.co/directory/3763/natural-language-processing-nlp/software>.
- Lynkova, D. (2021, 27 enero). Top 10 Machine Learning Algorithms: Why Are They So Important in 2021? TechJury. <https://techjury.net/blog/machine-learning-algorithms/#gref>
- *Tweepy Documentation*. (s. f.). Tweepy. <https://docs.tweepy.org/en/latest/>
- *Conda*. (s. f.). Conda. Recuperado 25 de agosto de 2021, de <https://docs.conda.io/en/latest/>
- De Smedt, T. & Daelemans, W. (2012). Pattern for Python. *Journal of Machine Learning Research*, 13: 2031–2035.
- Copeland, B. J. (2000). *Minds and Machines*, 10(4), 519–539. doi:10.1023/a:1011285919106.
- Kumar, E. (2011). *Natural language processing*. IK International Pvt Ltd.
- W. John Hutchins, Leon Dostert, & Paul Garvin (1955). The Georgetown-I.B.M. experiment. In *In* (pp. 124–135). John Wiley & Sons.
- Moor, J. (2006). The Dartmouth College artificial intelligence conference: The next fifty years. *Ai Magazine*, 27(4), 87-87.
- Yule, G. (2007). *El lenguaje*. Ediciones Akal.
- Woods, W., Kaplan, R., Kaplan, R., Nash-Webber, B., Language Research Foundation, Bolt, Newman, & Manned Spacecraft Center (1971). *The Lunar Sciences Natural Language Information System: Final Report*. Bolt Beranek and Newman.