Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

Estrategias para el análisis de sentimientos en textos extraídos de Twitter utilizando técnicas de aprendizaje profundo.

Tesis Profesional

Para obtener el grado de Licenciatura en Ingeniería en

Ciencias de la Computación.

Presenta:

Jessica Olivares Lopez

Asesor:

Dr. Abraham Sánchez López

Noviembre 2022

Agradecimientos

Los principales valores en la formación de un individuo se basan en la humildad, respeto y apatía. Aprender a identificar, agradecer y valorar a las personas, elementos y oportunidades, que se han obtenido a lo largo del camino de una meta, es uno de los valores esenciales del ser humano.

En este pequeño pero significativo apartado quiero agradecer en primer lugar a mi familia, a mamá, papá y hermano que siempre han estado ahí, en cada proceso de mi crecimiento cómo persona. Este logro, es mío, de ellos y para ellos. Para quienes ya no están, porque seguramente también lo estarían celebrando, pero permanecen en cada recuerdo y momento de mi vida. Sin su instrucción, apoyo, amor y aliento, este camino hubiese sido realmente difícil.

Me gustaría agradecer grandemente a mi asesor, el Dr. Abraham Sánchez López por su instrucción como asesor y su ardua dedicación cómo docente. El tiempo y esfuerzo en cada proceso de realización de esta tesis han sido una pieza fundamental en el trabajo realizado.

También me gustaría agradecer a mis amigos, con los que compartí conocimiento, pero lo más importante, experiencias durante mi periodo de vida como universitaria.

En último lugar, me gustaría agradecer a cada docente que ha sido parte de mi vida académica, la estudiante que soy ahora es gracias al esfuerzo y dedicación de cada uno de ellos.

"Cuando la gente te diga que eso no es bueno, sigue así. Tengo un muy buen principio para ayudar a las personas a seguir haciéndolo, que es; sus intuiciones son buenas o no lo son. Si tus intuiciones son buenas, debes seguirlas y eventualmente tendrás éxito. Si tus intuiciones no son buenas, no importa lo que hagas".

~ Geoffrey Hinton ~

Índice General

índice de ilustraciones

índice de tablas

1 Introducción

El análisis de sentimientos es una de las aplicaciones de la Clasificación de textos del Procesamiento del lenguaje natural (PLN), básicamente asigna una categoría apropiada al contenido de una oración, texto o documento, a partir del procesamiento de texto (previamente no estructurado). Esta clasificación se hace mediante la asignación de una polaridad de sentimiento: positivo, negativo o neutro a una oración o un documento, o a partir de la asignación de una emoción que se identifique en la oración.

Actualmente el análisis de sentimientos es una herramienta potente en diversas aplicaciones para distintas industrias. Se puede aplicar a diferentes sectores; textil, automotriz, alimenticio, manufacturera, etcétera, o incluso en dependencias gubernamentales, universidades, entre otros más, pero específicamente en espacios donde día a día se recolectan datos. Por lo que, cada vez es de mayor la importancia el uso de este tipo de estrategias. A partir de los datos que por años han sido generados y almacenados, se pueden hacer inferencias que ayuden a validar, sustentar o detener cualquier tipo de toma de decisiones.

En este trabajo de tesis se recolecta, implementa y evalúa una serie de estrategias basadas en machine learning y análisis de datos para el análisis de sentimientos. Iniciando desde la recolección de datos no estructurados en formato de texto recuperados de Twitter. Con los cuales se realizaron experimentos para análisis de sentimientos desde distintos puntos de referencia, de manera supervisada y no supervisada.

El conjunto de datos está integrado con texto extraído de Twitter, específicamente con tweets relacionados a la inteligencia artificial, lo cual

adicionalmente proporciona un pequeño panorama respecto a la opinión pública de esta área de la ciencia, a partir del análisis y la exploración de datos.

Este conjunto de estrategias puede ser aplicables o adaptables a cualquier conjunto de datos en formato de texto para problemas de análisis de sentimientos sin importar el dominio del texto, sin embargo, el desempeño o resultado de cada una está variado por la integridad de datos, el problema o tema principal de los datos, la variación de parámetros, entre otros más factores.

1.1 Objetivos

1.1.1 Objetivo general

Recolectar, implementar y evaluar estrategias de análisis de sentimientos basados en algoritmos de Machine Learning para textos específicos extraídos de Twitter, con el fin de valorar modelos y descartar inferencias a partir del análisis.

1.1.2 Objetivos específicos

1. Recolectar Tweets mediante la API de Twitter para la construcción de un conjunto de datos de opiniones y/o comentarios respecto a la inteligencia artificial.
2. Realizar procesos de limpieza y exploración de datos, mediante herramientas visuales y estrategias de limpieza para problemas de procesamiento de lenguaje natural.
3. Mediante una herramienta basada en la nube realizar una clasificación de sentimientos por polaridad de sentimiento para referencia y valoración de los modelos resultantes.
4. Implementar algoritmos para la extracción de características basados en técnicas de Deep Learning.
5. Entrenar modelos de análisis de sentimientos supervisados y no supervisados con algoritmos de Machine Learning.
6. Implementar herramientas visuales para el análisis de sentimientos.
7. Valorar los modelos obtenidos a partir de los resultados y deducir conclusiones.
8. Proporcionar un análisis general de la opinión pública en Twitter respecto a la inteligencia artificial a partir de los resultados obtenidos.

1.2 Estructura de la tesis

Capítulo 1

En el estado del arte se encuentra toda la literatura necesaria para la realización de este trabajo de tesis. Partiendo desde la interrogativa ¿qué es el análisis de sentimientos? Hasta la definición de algunos temas y términos esenciales para la comprensión del trabajo.

Capítulo 2

En esta sección se muestran los procesos de extracción de datos, análisis y limpieza. Estos procesos son considerados como parte del procesamiento y exploración de datos, los cuales son de alta importancia en cualquier trabajo de Machine Learning, pues la calidad de datos influye en el desempeño y comportamiento de los algoritmos, además, la exploración de datos permite esclarecer qué técnicas son las más adecuadas.

Capítulo 3

Ahora con los datos analizados y preprocesados, se realiza la tarea de clasificación de sentimientos. Mediante una exploración de algoritmos de Machine Learning para la clasificación de sentimientos, empleando algoritmos supervisados y no supervisados, desde algoritmos básicos hasta algoritmos más complejos basados en Deep Learning.

Capítulo 4

En este capítulo se hace énfasis en el análisis de datos, mostrando las observaciones e inferencias obtenidas, adyacentes al análisis de sentimientos, cómo, por ejemplo: qué polaridad predomina en los Tweets, las emociones

inferidas en los Tweets, las palabras frecuentes que han sido empleadas, extracción de palabras y oraciones importantes, la relación de las palabras y el contenido del corpus en general con el contexto de inteligencia artificial.

Capítulo 5

La última sección de este trabajo de tesis es para las conclusiones generales del trabajo realizado en los capítulos anteriores, además de incluir opciones de trabajos futuros, que pudieran proporcionar mejores resultados en trabajos o problemas similares.

1 Capítulo 1

Estado del arte

1.1 Análisis de sentimientos

El procesamiento de lenguaje natural, o por sus siglas PLN, es una de las más grandes áreas de la inteligencia artificial que por décadas se ha dedicado a la interpretación y comunicación del lenguaje máquina y lenguaje humano.

El análisis de sentimientos es una de las técnicas de mayor relevancia de esta área, generalmente para la evaluación de todo el contenido en texto generado en la web [4]. Sin embargo, el análisis de sentimientos es un área multidisciplinar, pues además del Procesamiento de lenguaje natural intervienen disciplinas cómo lingüística y psicología. El hecho de definir las categorías en las que será clasificado un texto basado en la premisa de "sentimiento" es una cuestión psicológica muy ambigua y diversa, pues existe una variedad de posturas al respecto, hay cientos de estados emocionales que forman parte de la condición humana. Una clasificación o representación de las emociones humanas popular, es la rueda de Plutchik. El psicólogo Robert Plutchik sugiere que hay 8 emociones evolutivas humanas, es decir, emociones que han formado parte de la supervivencia humana y que han sido transferidas de generación en generación. Estas emociones son las siguientes:

1. Ira
2. Miedo
3. Tristeza
4. Repugnancia

5. Sorpresa
6. Expectación
7. Confianza
8. Alegría

En la representación de Plutchik muestra que cada una de estas emociones centrales puede intensificarse, atenuarse o incluso combinarse para producir cualquier estado emocional. Además, en la rueda de Plutchik, cada emoción tiene una emoción opuesta, por ejemplo, la tristeza es una emoción directamente opuesta a la felicidad. En la ilustración 1 se muestra la rueda de Plutchik, dónde se visualizan cada una de las emociones, y los estados emocionales derivados de estas emociones.

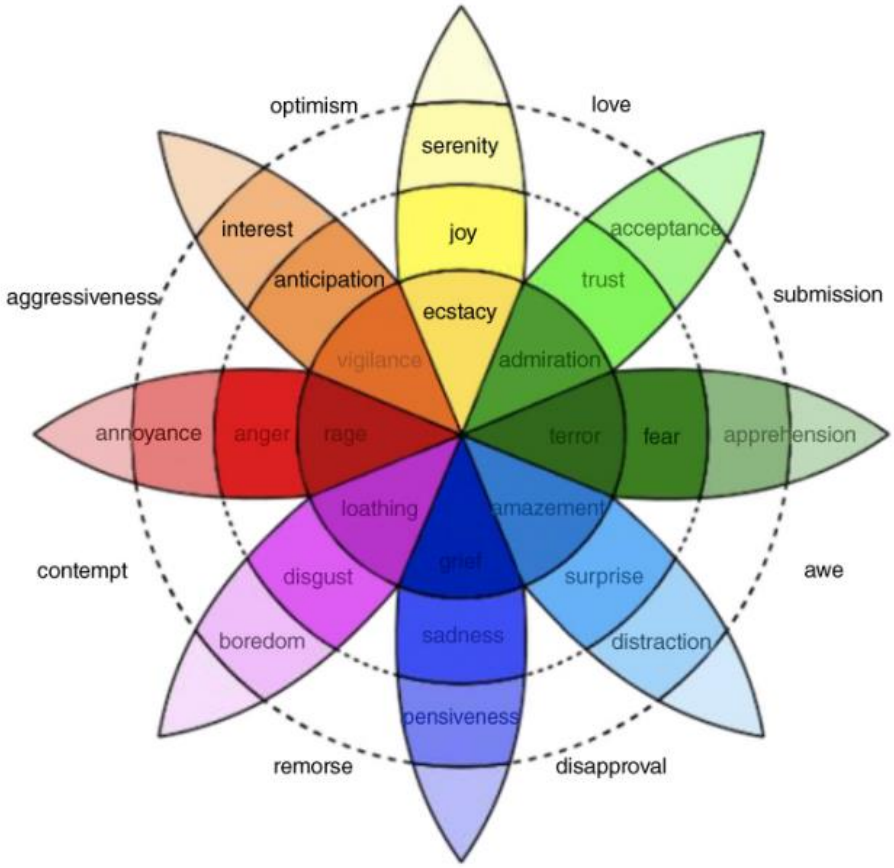

Ilustración 1La rueda Plutchik de emociones con ocho emociones principales, imagen recuperada de [9].

Sin embargo, muchos de los trabajos realizados para análisis de sentimientos, están basados en la clasificación del texto de acuerdo con la polaridad del sentimiento, bueno, malo o neutral, en su mayoría. Lo cual facilita considerablemente el problema de análisis de sentimientos, además, la polaridad puede ser más precisa porque solo hay dos o tres clases distintas, que hace más fácil de distinguirlas entre sí, mientras que una clasificación por emociones puede ser más ambigua, pues una oración puede involucrar más de una emoción.

Generalmente el análisis de sentimientos hace la categorización de texto no procesado en polaridades de acuerdo con las siguientes categorías:

1. Positivo
2. Negativo
3. Neutro

Los casos de uso comunes o aplicaciones para el análisis de sentimientos se destacan el seguimiento de los comentarios de los clientes, la orientación de las personas para mejorar un servicio, el seguimiento de cómo un cambio en el producto o servicio afecta cómo se sienten los clientes, entre muchos otros más. Desde encuestas de opinión, foros en internet, buzón de sugerencias, hasta estrategias de marketing creativas. Este tipo de estrategias ha redefinido por completo la forma en que operan y toman decisiones las empresas.

1.2 Inteligencia artificial

La introducción de la red 4.0, es considerada por algunos la cuarta revolución industrial, pues ha revolucionado la manera en la que las industrias se mueven, ejemplo de ello, es la extrema necesidad del uso de tecnologías, cómo BigData, Internet of Things, blockchain, entre otras, pero la más relevante de estas es la "Inteligencia Artificial", una disciplina de las ciencias computacionales que ha modificado la manera de resolver problemas en los últimos años, a pesar de estar presente desde ya hace varias décadas, pero que gracias al acceso y

avance tecnológico ha establecido un gran potencial y ha tenido mayor presencia en la vida cotidiana del ser humano.

Antes de definir, inteligencia artificial, sería interesante profundizar en ¿qué es inteligencia? algo tan controversial e incluso complejo de comprender. Las personas suelen asociar con "inteligencia": el razonamiento, la resolución de problemas, el pensamiento abstracto, la rapidez de efectuar una tarea, el pensamiento crítico, etcétera. L. S. Gottfredson define inteligencia como "*Capacidad mental general que incluye la habilidad de razonar, planificar, resolver problemas, pensar en abstracto, comprender ideas complejas, aprender rápido y aprender de la experiencia, que es más que una destreza académica o del aprendizaje por medio de libros*" [2]. Estas actividades son consideradas como cognitivas, para el ser humano, pues están relacionadas con la manera en la que se procesa y responde a la información inferida. Por su parte, artificial, explicado de manera muy sencilla, es algo que no es realizado de manera "natural" o por la naturaleza, es decir, algo construido por el hombre.

Ahora entonces, la inteligencia artificial o por sus sigas IA, es un campo de la ciencia que se dedica al estudio y asimilación de la inteligencia humana, su objetivo principal es la simulación del proceso de respuesta humana a la recepción de información, empleando programas o sistemas informáticos. En los últimos años la IA ha experimentado una mejora sustancial en varios dominios, como la visión artificial, la robótica, los vehículos autónomos, los videojuegos, la traducción de idiomas, diagnósticos médicos, reconocimiento de voz, etcétera [8], debido al avance tecnológico que se ha hecho más accesible, el acceso a mayor capacidad de procesamiento y almacenamiento, pues algoritmos complejos, que eran difíciles de implementar, ahora pueden ser procesados incluso desde la nube. Las tecnologías centrales detrás de estos avances en IA son sobre los ejes de aprendizaje automático y el aprendizaje profundo.

En la ilustración 2, se muestra una parte de campos o áreas de estudio y su relación entre ellos mediante un diagrama de Venn, entre estas áreas se

encuentra el PLN, que cómo ya se mencionó anteriormente, es un área que se dedica al estudio de lenguaje humano-máquina, y dónde el análisis de sentimientos forma parte de los problemas o aplicaciones.

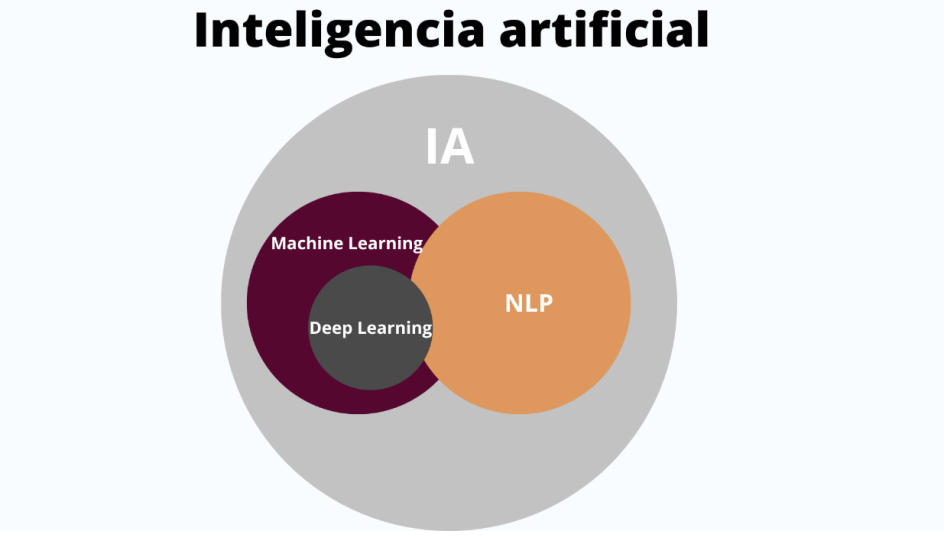

Ilustración 2 Diagrama de Venn de las áreas de la Inteligencia artificial.

1.3 Machine Learning

Infiriendo sobre la ilustración 1, se puede decir que el machine learning es un campo o área de la inteligencia artificial, esto es, todo machine learning es Inteligencia artificial, pero no toda inteligencia artificial es machine learning. En su traducción al español, se puede entender como aprendizaje automático, y cómo su nombre lo indica, los modelos de machine learning tratan de "aprender" patrones de datos que son procesados y analizados, algunas veces etiquetados, otras no, para la predicción de datos o clasificación de datos. En un modelo tradicional de machine learning, se tiene la siguiente estructura: entrada /salida, programa y modelo.

Esta área de la inteligencia artificial está fuertemente familiarizada con la estadística, pues en su mayoría, los algoritmos, métricas, elementos de apoyo visual son o están basados en estadística. Las aplicaciones de los algoritmos y/o

técnicas de machine learning tienen una amplia gama de aplicaciones, cómo: reconocimiento de voz, visión computacional, vehículos autónomos, diagnóstico de enfermedades, recomendaciones, Big Data, búsqueda en la web, identificación de personas, detector de spam, etcétera. En la siguiente ilustración se muestran los diferentes algoritmos de machine learning y sus clasificaciones.

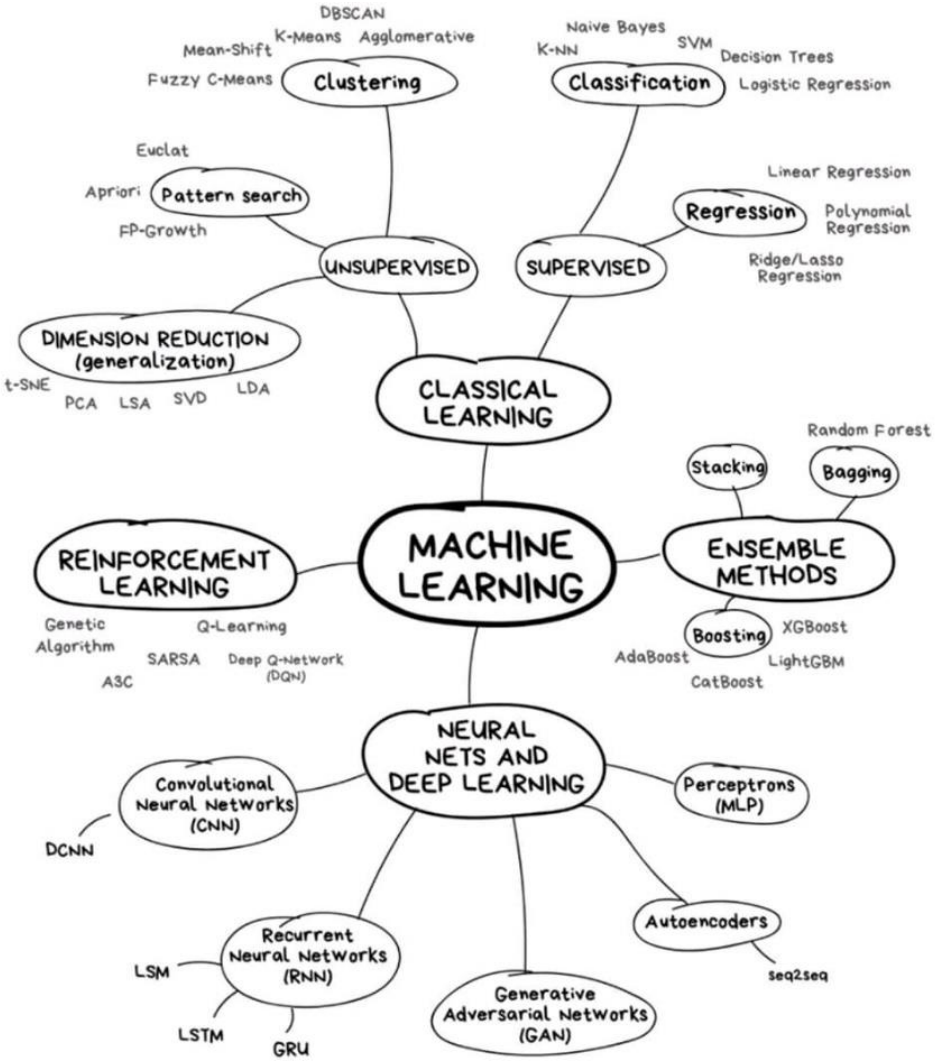

Ilustración 3 Estructura orgánica del Machine Learning (Imagen recuperada de [12]).

1.3.1 Deep Learning

Deep Learning es una de las áreas más modernas y complejas del Machine Learning, los algoritmos de Deep Learning están apoyados en la estructura y funcionamiento del cerebro humano [11], usan la arquitectura de una red neuronal, la cual es una organización jerárquica de neuronas artificiales con conexiones a otras neuronas. Una de las principales bondades de una neurona es que puede aprender fácilmente sin la necesidad de tener datos etiquetados manualmente, en algunos casos.

En la siguiente figura se muestra la estructura básica de una red neuronal, este modelo es una red de 2 capas, para definir el número de capas de la red no es considerada la capa de entrada. En la capa de entrada se reciben todos los datos u observaciones del conjunto X. La capa oculta de la red neuronal se encuentra entre las capas de entrada y salida, en la ilustración esta capa está representada por el color amarillo, esta puede ser de cualquier cantidad, pero a partir de 2 capas es considerada como profunda. Por último, se tiene la capa de salida, la cual proporciona el valor calculado por la red, puede tener más de una salida, el número dependerá del problema; una salida para cada clase.

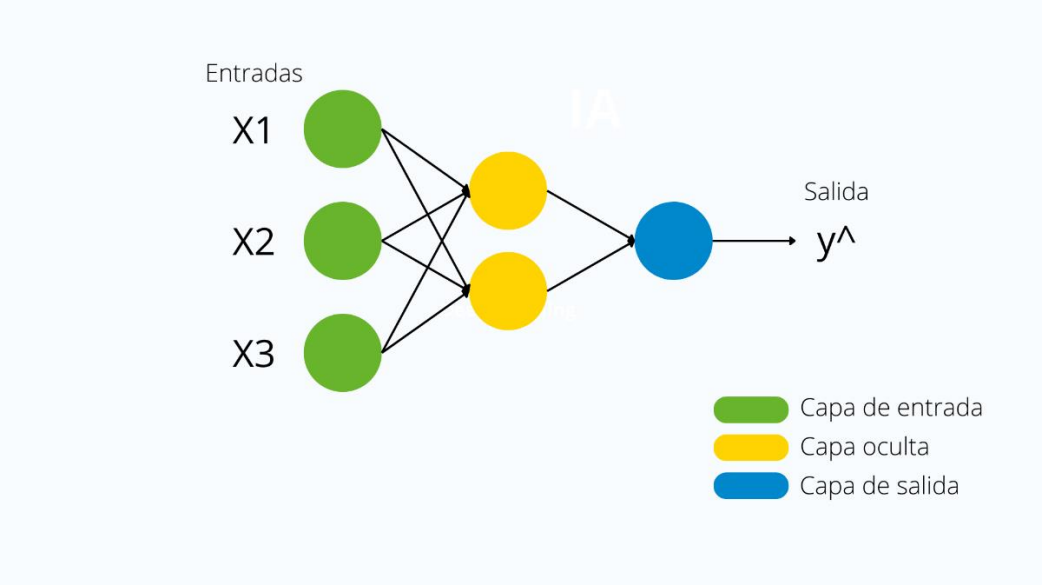

Ilustración 4 Estructura básica de una red neuronal.

Existen más conceptos y/o elementos de una red neuronal, cómo pesos, parámetros, hiper parámetros, funciones de activación, función de pérdida, función de costo, entre muchos más, de los que no se entrará a detalle, pero que son consideramos para la implementación de los modelos en capítulos posteriores.

1.3.2 Aprendizaje supervisado

Este tipo de aprendizaje se denomina supervisado, porque se requiere de una supervisión, generalmente humana, en el proceso de etiquetamiento de datos, se dispone de datos etiquetados, es decir, de una variable objetivo-observada o clasificada. Las técnicas más utilizadas actualmente en el aprendizaje supervisado son las redes neuronales, máquinas de soporte vectorial, los clasificadores bayesianos, y la familia de algoritmos basados en árboles [10].

1.3.3 Aprendizaje no supervisado

Por su parte, para el aprendizaje no supervisado no se dispone de una variable objetivo-observada o clasificada, los algoritmos de aprendizaje no supervisado son los encargados de encontrar patrones o relaciones entre ellos, para realizar tareas, cómo clasificación, dónde se tienen algoritmos basados en técnicas de clustering. Y aunque actualmente no hay técnicas lo suficientemente sofisticadas para tareas no supervisadas, existen alternativas muy eficientes de Deep Learning para la extensa cantidad de datos no etiquetados que se disponen. El aprendizaje no supervisado es ideal cuando se manejan datos no estructurados, por ejemplo, imágenes, video, audio y texto.

1.4 Algoritmos de Deep Learning para análisis de texto

En general existen distintos métodos de aprendizaje profundo no supervisado, estos métodos se pueden clasificar en cuatro distintas categorías.

Clustering analysis. Es un enfoque Long-standing, la clave es el descubrimiento de múltiples clústeres consistentes en los datos de entrenamiento.

Sample specificity learning. Considera cada observación como una clase. Este tipo de supervisión no modela explícitamente los límites de decisión de clase como el análisis de conglomerados. Obteniendo clases más ambiguas y características menos discriminatorias.

Self-supervised learning. Los métodos existentes varían esencialmente en el diseño de la supervisión auxiliar no supervisada. Por lo general, dicha supervisión auxiliar está diseñada a mano para explotar cierta información intrínsecamente disponible en los datos de entrenamiento no etiquetados.

Generative models. El modelo generativo está basado en principios de aprender la verdadera distribución de datos del conjunto de entrenamiento sin supervisión. Los modelos generativos más utilizados y eficientes incluyen Máquinas de Boltzmann Restringidas, Autoencoders y Generative Adversarial Network.

1.5 Neural bag of words

La bolsa de palabras o bag of words, es una de las técnicas más populares y sencillas para la extracción de características de texto en problemas de análisis de texto. Este modelo transforma cada documento en un vector y a cada palabra dentro de un documento se le asigna un score, cada puntuación se coloca en la ubicación correspondiente en la representación. Existen distintos métodos de puntuación:

Binary. Cada palabra es señalada mediante dos posibles valores presente (1) o ausente (0).

Count. Se indica el valor de la frecuencia de cada palabra en un documento.

TF-IDF. Cada palabra se puentea según la frecuencia de esta, mide qué tan común es cada palabra en todos los documentos.

Freq. Cada palabra es punteada según su frecuencia en cada documento.

Se convierte cada documento en un vector listo para el entrenamiento de un modelo de redes neuronales simple cómo un Multi Layer Perceptron (MLP) y este puede funcionar como clasificador en un enfoque de análisis de sentimientos, donde por cada entrada se obtiene una salida cómo: negativa, positiva o neutra.

1.6 Word embedding

Es una forma de representar texto basada en la idea de que palabras que tienen el mismo o similar significado tienen representaciones similares. Se representan como vectores con valores reales en un espacio vectorial predefinido. Cada palabra se asigna a un vector y estos valores vectoriales se aprenden de una manera que se asemeja a una red neuronal. Al ser entrenadas las Word embeddings recopilan más información en menos dimensiones.

Existen varios algoritmos que se pueden utilizar para obtener una representación de Word embeddings a partir de un conjunto de datos de texto.

1. **Word2Vec.** Es un modelo de una red neuronal con una sola capa oculta, que a partir de un corpus entrena la red. Se encuentra disponible de dos formas: Continuous Bag-of-Words (CBOW) o el modelo Skip-Gram.

2. **The Global Vectors for Word Representation or GloVe.** Algoritmo de aprendizaje no supervisado para aprender la representación vectorial, es decir, la incrustación de palabras para varias palabras.

1.7 Twitter API

Tweepy es una API desarrollada en Python que provee acceso a todos los métodos RESTful API de Twitter [17] mediante los cuales se puede acceder al contenido de esta plataforma. La idea es obtener Tweets que estén relacionados con la inteligencia artificial, es decir, que en su contenido se encuentra texto en este contexto, mediante las herramientas de esta API para la creación y estructuración de un dataset o conjunto de datos.

1.8 Azure Text Analysis

Microsoft Azure es una de las plataformas más usadas para realizar procesos en la nube. Entre los servicios que son ofrecidos mediante Cognitive Services de Azure para el lenguaje mediante Text Analysis se encuentran: named entity recognition, key phrase extraction, custom text classification, entre otros, como el análisis de sentimientos y la minería de opiniones. Cognitive Service for Language, es una colección de algoritmos de inteligencia artificial y aprendizaje automático en la nube que sirve para desarrollar aplicaciones inteligentes que involucran el lenguaje escrito. Estas funciones ayudan a descubrir lo qué piensan u opinan las personas sobre un tema mediante la extracción y procesamiento de texto.

1.9 Métricas

Las métricas son estrategias de machine learning que ayudan a medir el desempeño de los modelos, existen una gran cantidad de métricas, cada una de ellas tiene sus especificaciones y dependiendo de estas se decide cuáles son las más adecuadas para valorar un algoritmo, en esta sección se muestran algunas de las métricas que han sido empleadas para evaluar el desempeño de los algoritmos implementados en el capítulo 3, de esta manera, hay métricas que se

adecuan para aprendizaje supervisado y también para aprendizaje no supervisado.

1.9.1 Accuracy

El accuracy o precisión, es una métrica que se utiliza para evaluar los modelos de clasificación, mide la exactitud con la que un modelo realizó las predicciones, a partir de la división del número de predicciones correctas sobre el número total de predicciones.

1.9.2 Unsupervised Clustering Accuracy (ACC)

ACC es el equivalente no supervisado de la precisión de la clasificación. ACC se diferencia de la métrica de precisión habitual en que utiliza una función de mapeo para encontrar el mejor mapeo entre la salida de asignación de clúster del algoritmo con la etiqueta. Este mapeo es necesario porque un algoritmo no supervisado puede usar una etiqueta diferente a la etiqueta correcta para representar el mismo clúster.

1.9.3 Normalized Mutual Information (NMI)

NMI es una métrica teórica de información que mide la información mutua entre las asignaciones de clúster y las etiquetas del dataset. Está normalizado por el promedio de entropía de las etiquetas y las asignaciones de grupos [16].

1.9.4 Homogeneidad

Cuantifica cuántos clústeres contienen solo miembros de una sola clase. Esta métrica es independiente de los valores absolutos de las etiquetas: una permutación de los valores de las etiquetas de clase o clúster no cambiará el valor de la puntuación de ninguna manera [14].

1.9.5 Completeness o Integridad

Cuantifica cuántos miembros de una clase determinada se asignan a los mismos grupos [15].

1.9.6 V-measure

La media armónica de integridad y homogeneidad es idéntica a Normalized mutual information pero con la opción aritmética para promediar [13].

1.9.7 Rand-Index

Mide la frecuencia con la que los pares de puntos de datos se agrupan de forma coherente según el resultado del algoritmo de agrupación y la asignación de clase de verdad básica.

1.9.8 Adjusted Rand Index (ARI)

El índice de Rand calcula una medida de similitud entre dos agrupamientos considerando todos los pares de muestras y contando los pares que se asignan en el mismo o en diferentes agrupamientos en los agrupamientos predichos y verdaderos. El índice Rand ajustado es la versión corregida por azar del índice Rand.

1.10 Desbalanceo de clases

La desproporcionalidad de observaciones en una muestra en Machine Learning se conoce como desbalanceo de clases [5], se da cuando en la muestra o conjunto de datos tienen diferencia en cuanto a la cantidad de observaciones por clase de la variable objetivo. Este problema puede repercutir directamente en la eficiencia del modelo. Para problemas de clasificación, el hecho de tener una clase mayoritaria y una minoritaria representa un peso mayor o menor sobre el total de la muestra. Existen posibles soluciones a este problema, un conjunto de técnicas se basa en la modificación de hiper parámetros de los

algoritmos para mejorar el desempeño del modelo, trabajando sobre los datos desbalanceados, y por otra parte se tienen las técnicas de balanceo de datos o resampling, las cuales eliminan o agregan observaciones e incluso clases para lograr conjuntos o muestras de datos balanceados.

1.10.1 Resampling

Este tipo de técnicas están centradas la estructura de la composición de los datos, para la solución del desbalanceo de datos modificando la distribución de algunas o varias de las clases. Las técnicas de resampling se pueden clasificar en dos categorías undersampling y oversampling [5], los cuales básicamente eliminan observaciones de la clase mayoritaria y el otro replica o genera instancias de la clase minoritaria, respectivamente.

1.10.1.1 Undersampling

Las técnicas de undersampling son todas aquellas técnicas de balanceo de datos que tienen como propósito igualar las distribuciones desbalanceadas eliminando observaciones de la clase mayoritaria. Esta eliminación se puede hacer de manera aleatoria con Random Undersampling, o mediante selección de características donde se puede emplear algoritmos de machine learning para la selección de qué observaciones serán eliminadas, cómo K-near neighbours.

1.10.1.2 Oversamplig

Al contrario de las técnicas de undersampling que eliminan observaciones de las distribuciones para igualar las distribuciones. Las técnicas de oversampling aumentan el tamaño de la muestra original, tiene como propósito replicar instancias de la clase minoritaria a partir de las otras instancias.

2 Capítulo 2

Extracción y exploración de datos

2.1 Recopilación y preparación de datos

Uno de los objetivos de este trabajo es recuperar datos de Twitter, obteniendo así datos no estructurados, a los cuales es necesario aplicar estrategias de preprocesamiento para poder realizar el análisis de sentimientos. El primer paso, es consolidar un conjunto de datos que contenga la información y estructura necesaria para realizar la tarea de análisis de sentimientos.

El proceso de creación de un conjunto de datos consta de tres procesos; adquisición de datos, limpieza de datos y etiquetamiento de datos.

2.1.1 Adquisición de datos

Para la adquisición de datos se hizo uso de la API de Twitter, bajo una cuenta de desarrollador. La consulta de extracción de datos se describe en los parámetros de la ilustración 5.

Ilustración 5 Estructura de consulta de extracción de texto de Twitter.

Se realizó un filtro de los Tweets que en su contenido se encontrarán los términos de la ilustración 5, inteligencia artificial, deep learning y machine learning esencialmente, al ser considerados unos de los temas más relacionados y

populares de la inteligencia artificial. Obteniendo así un total de 12,000 observaciones, que en este contexto son Tweets.

2.1.2 Limpieza de datos

El proceso de limpieza de datos es uno de los procesos que define el éxito de un modelo de Machine Learning. Para tareas o problemas de Procesamiento del Lenguaje Natural se realizan algunas estrategias básicas que facilitan la manipulación del texto. El proceso de limpieza de esta propuesta está compuesto de dos partes tal y cómo se ilustra en la siguiente figura.

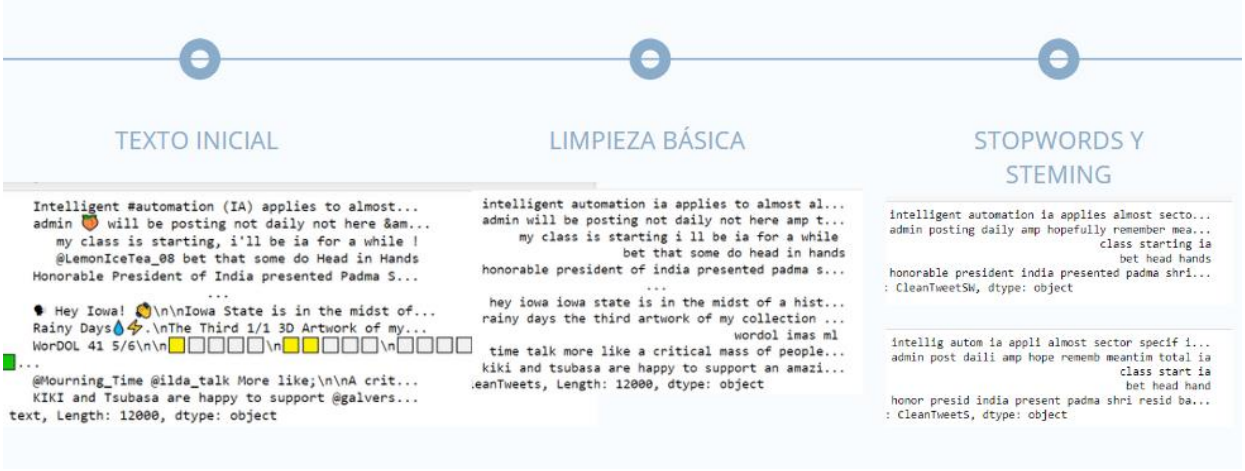

Ilustración 6 Proceso de limpieza del conjunto de datos.

La limpieza básica consiste en la eliminación de algunos elementos, pues al ser extraídas las observaciones de un medio social como Twitter, en el cual el texto lo integran elementos adicionales como: usuarios, menciones, hashtags, enlaces o URL y emoticones. Estos elementos deben ser eliminados, pues pueden introducir ruido y reducir el desempeño o aprendizaje de un modelo.

La segunda parte de la limpieza de datos es más general para cualesquiera tareas de Lenguaje de Procesamiento Natural y consiste en la realización de las siguientes tareas [9].

- **Convertir el texto a minúsculas**. Esto facilita algunos procesos de exploración que se muestran más adelante.
- **Eliminar espacios dobles.** Se hace la sustracción de espacios dobles, porque no aportan nada al contenido del texto.
- **Eliminar números.** La eliminación de números previene la errónea interpretación de números en los modelos.
- **Eliminar StopWords**. Consiste en la eliminación de palabras cortas que regularmente se ocupan con mayor frecuencia en el idioma y que podrían no añadir valor al contexto, ejemplos de estas palabras son: conjunciones, artículos, preposiciones, etcétera.
- **Steming.** Este proceso consiste en convertir una palabra a su palabra base.

2.1.3 Etiquetamiento de datos

Finalmente, una vez recolectados y estructurados los datos, se procede a la asignación de una etiqueta, en particular, una polaridad de sentimiento. Esta asignación se realizó de manera semi-supervisada. Mediante un flujo de trabajo en Python que solicitó respuesta del servicio de Text Analysis de Azure, el cual, entre otras tareas, puede asignar una polaridad de sentimiento a un conjunto de documentos, o textos. Además de una supervisión manual sobre la etiqueta asignada por Text Analysis.

Text Analysis, asigna una de las siguientes categorías o polaridad de texto: neutral, positive, negative y mixed. También devuelve puntuaciones de confianza entre 0 y 1 para cada documento y oraciones dentro del documento.

Una vez realizada la solicitud de las 12,000 observaciones recolectadas se obtiene la siguiente cantidad de observaciones para cada categoría de sentimiento: neutral; 6162, positive; 2965, negative; 2290 y mixed; 583, relación que se muestra en la tabla 1.

Tabla 1 Número de observaciones por sentimiento.

Sentimiento	Observaciones
neutral	6162
positive	2965
negative	2290
mixed	583

2.2 Exploración de datos

La exploración de datos como su nombre lo indica es el proceso en el cual se realiza una exploración o examinación de los datos, para entender la composición o integración de estos y obtener las primeras observaciones e inferencias sobre ellos. Para el procesamiento del lenguaje natural existen varias técnicas que permiten visualizar y deducir conclusiones eficientemente. Se realizaron algunas de estas técnicas, con las que se obtuvieron peculiares observaciones que se irán detallando brevemente.

En primer lugar, se obtuvieron dos histogramas de la frecuencia de palabras a partir de la composición de n gramas. N – gramas es una subsecuencia, donde n es el número de elementos de la subsecuencia, esta técnica es usualmente utilizada en el procesamiento del lenguaje natural para el tratamiento de textos.

Los histogramas de las siguientes ilustraciones corresponden a la división del texto en 2 y 3 gramas, bigramas y trigramas, respectivamente. Para obtener este gráfico es necesario obtener la lista de bigramas y trigramas con su respectivo número de ocurrencias a partir del corpus.

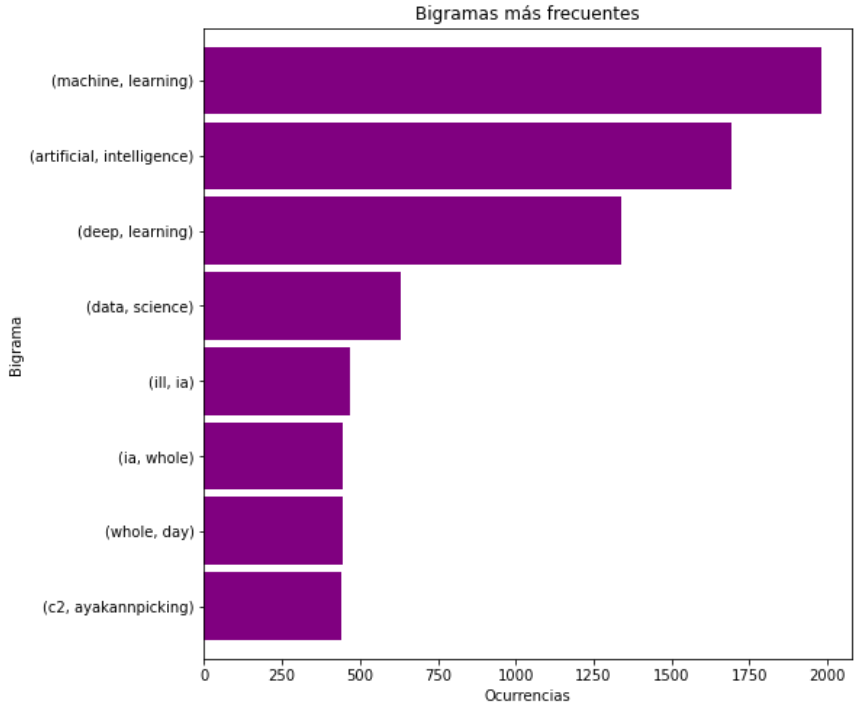

Ilustración 7 Histograma de Bigramas más frecuentes.

Al comparar ambos histogramas, la implementación de bigramas favorece al comportamiento de los datos, pues retomando el contexto, se tiene mayor coherencia del contenido con respecto de la división en trigramas donde, los subconjuntos de palabras no tienen mucha relevancia al contexto del problema. Entre los bigramas más relevantes del histograma de la ilustración 7 se encuentran, machine-learning, artificial-intelligence, deep-learning y data-sicience, todos estos términos que son directamente relacionados con la "Inteligencia artificial".

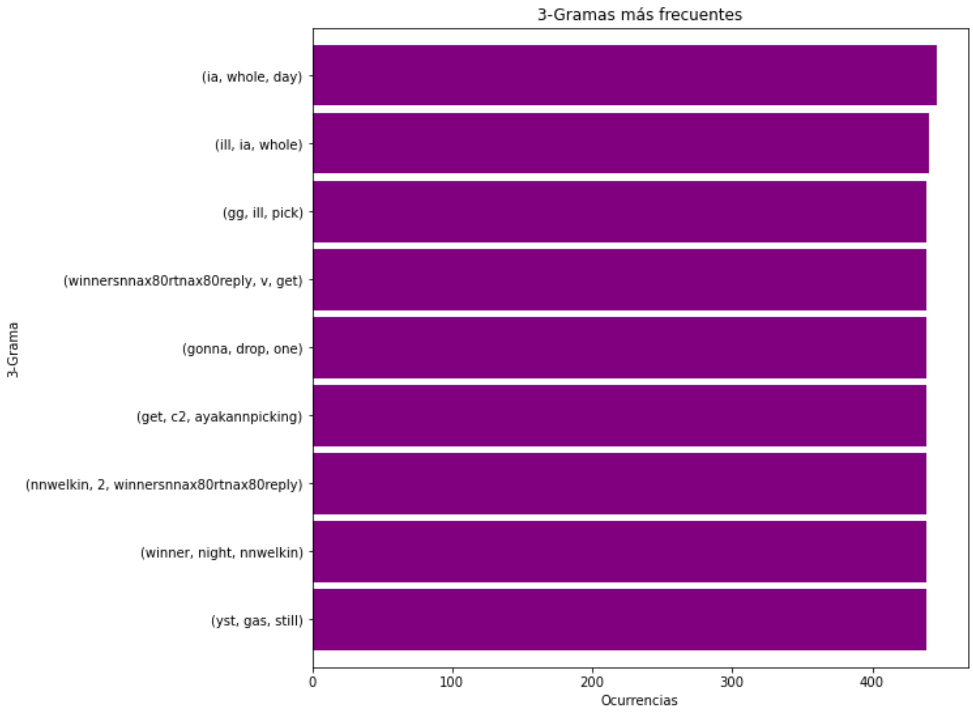

Ilustración 8 Histograma de Bigramas más frecuentes.

A partir de la lista de bigramas obtenidos, también se puede obtener una representación semántica, la cual muestra la relación que hay entre distintos bigramas establecidos en el paso anterior.

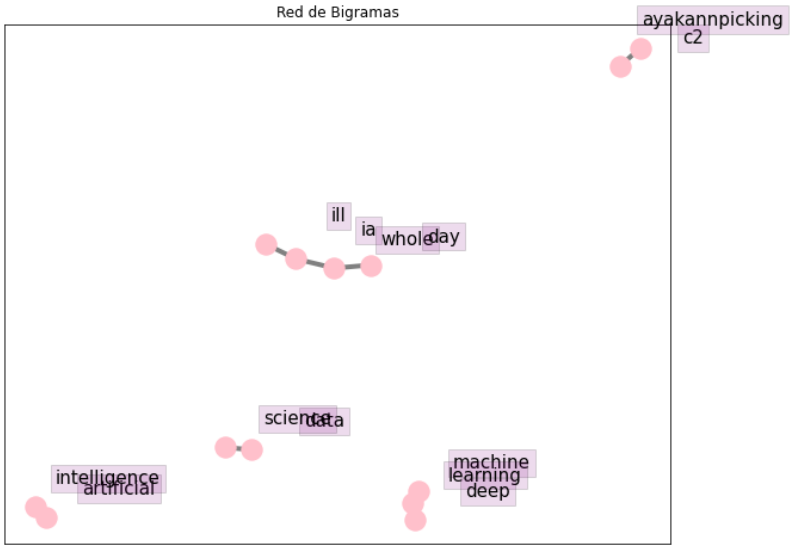

Ilustración 9 Red de Bigramas integrada por las 8 bigramas con mayor ocurrencia.

La visualización de bigramas es de ayuda para confirmar que los datos si están constituidos en base al contenido deseado, es decir, con opiniones respecto a la Inteligencia artificial.

En el Capítulo 1, se habló de las representaciones de texto, entre las que se encuentran Word embeddings o incrustaciones de palabras, las cuales son una representación de texto muy popular en Deep Learning, esta es la representación de texto que se emplea en los algoritmos que han sido implementados y valorados en capítulos posteriores. Sin embargo, esta misma representación también puede ser eficiente desde el proceso de exploración de datos, las incrustaciones de palabras son representaciones vectoriales que permiten entre otras cosas conocer su relación entre ellas.

Mediante la ayuda del proyector de Embeddings de TensorBoard [3] una herramienta que permite visualizar gráficamente los vectores de incrustaciones, para facilitar la interpretación de las relaciones al visualizar Word embeddings se muestra la visualización de las Word Embeddings, dónde cada punto es el vector correspondiente a la representación de la palabra que contiene la etiqueta. Para poder visualizar las incrustaciones específicas del dataset, cómo se muestra en las ilustraciones 10, 11 y 12, es necesario hacer un entrenamiento previo, que genera los vectores de características de 4 dimensiones correspondientes a las representaciones del vocabulario de las observaciones del conjunto de datos.

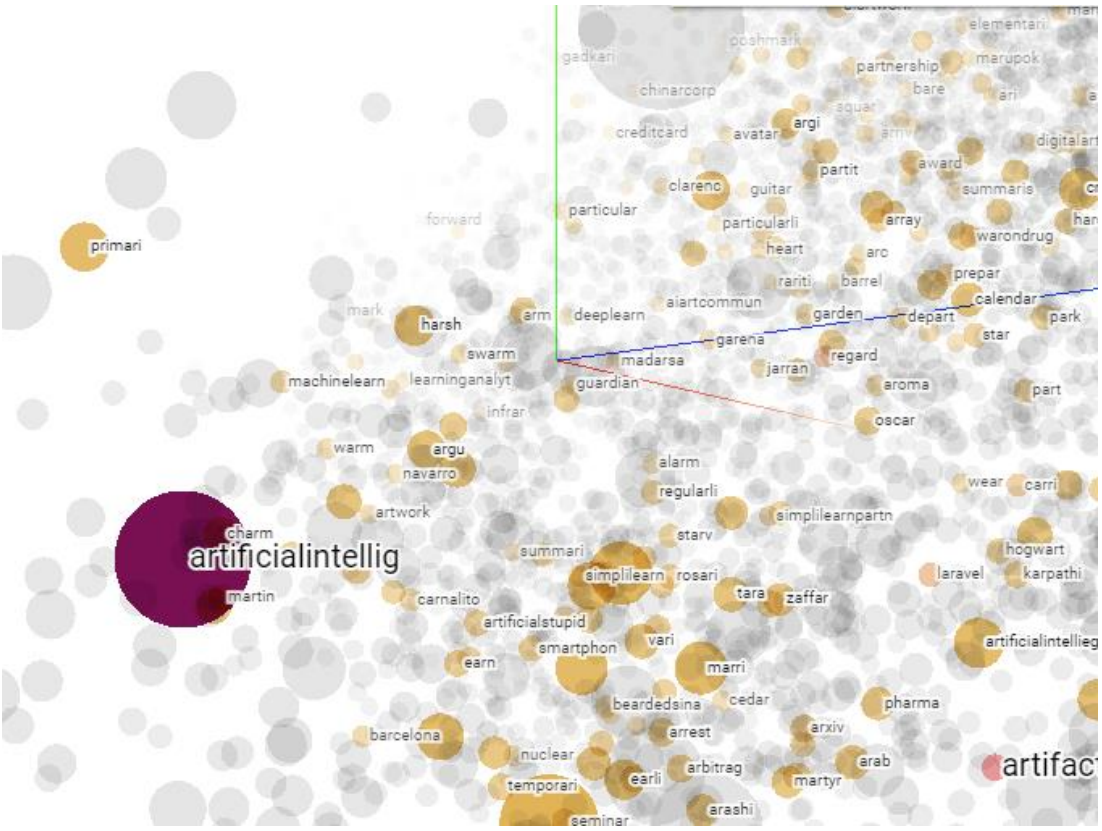

Ilustración 10 Visualización de espacio vectorial de Word Embeddings cercanas a "artificialintelling".

Esta herramienta permite a partir de una palabra, encontrar su representación y las palabras más cercanas, es decir, las palabras relacionadas a partir del cálculo de la distancia (coseno o euclidiana). En las figuras, se muestra la representación o visualización de representaciones de "artificialintelling", "learning" y "neuralnetwork". De las cuales se puede corroborar contextualmente la relación de los términos mostrados en las figuras, por ejemplo, para la palabra "learning", se indica que hay relación con los términos "learn", "thinking", "knowledge", entre otras más. Y efectivamente, existe una relación conceptual de estos términos y así mismo funciona para los otros dos ejemplos de las ilustraciones 11 y 12.

Ilustración 11 Visualización de espacio vectorial de Word Embeddings cercanas a "learning".

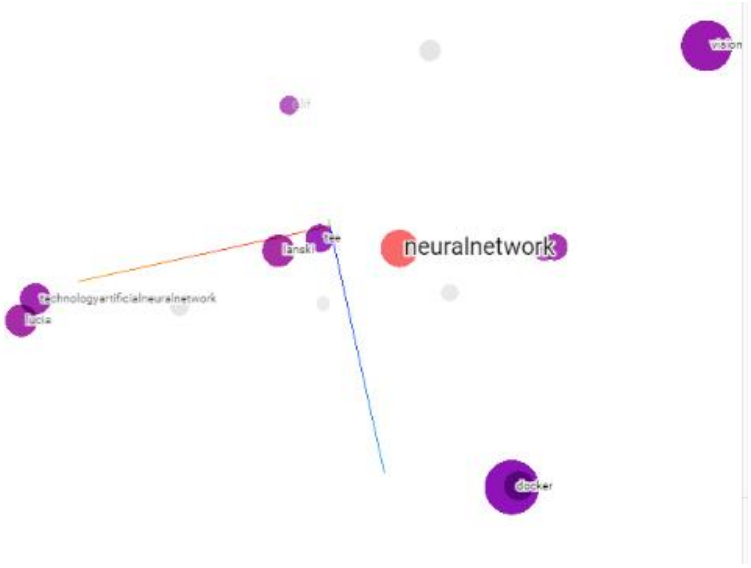

Ilustración 12 Visualización de espacio vectorial de Word Embeddings cercanas a "neuralnetwork".

2.3 Selección y balanceo de datos

En el histograma de la ilustración 13 se muestran las observaciones que se tienen para cada una de las categorías de sentimiento. De la cual a primera vista se puede ver una desproporción de observaciones, teniendo más del doble de observaciones con polaridad neutra respecto de observaciones con polaridad positive. Mientras que las observaciones positive y negative no son tan diferentes, por otra parte, se tiene las observaciones de polaridad mixed, con un número muy inferior de observaciones sobre las otras categorías o polaridades de sentimiento.

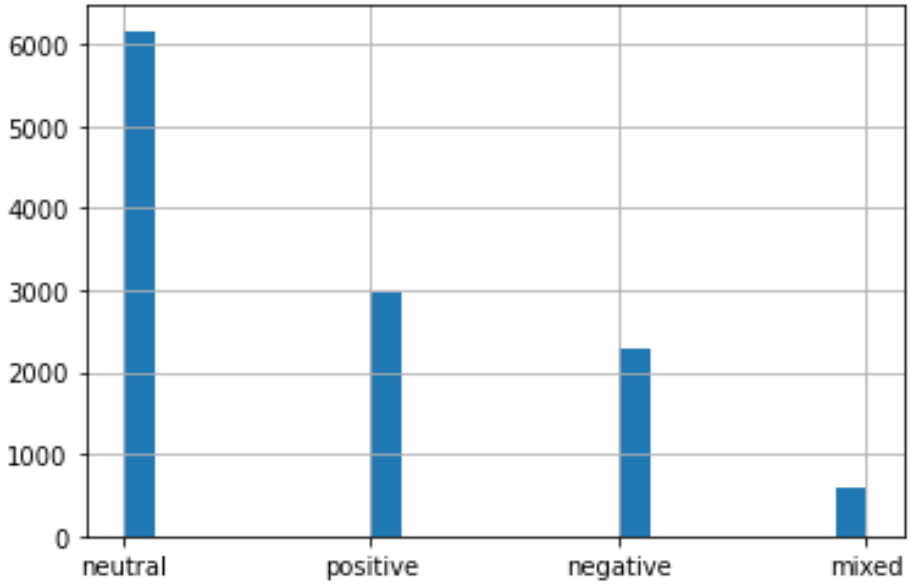

Ilustración 13 Histograma de la distribución de datos respecto a la variable objetivo (sentiment).

Por lo anterior se puede decir que la distribución de datos está desbalanceada, y este problema podría afectar el desempeño de los modelos, al introducir un sesgo de información sobre las clases mayoritarias y minoritarias; neutral y mixed, respectivamente. Para evitar problemas en fases futuras, se aplican técnicas de balanceo de clases, cómo las mencionadas en el capítulo 1.

Para esta distribución en específico, lo deseable es igualarla, es decir, tener el mismo número de observaciones negativas, positivas, neutras y mixtas. Para no reducir considerablemente el número de instancias de cada clase se eliminaron las muestras de la clase mixed, reduciendo el número de observaciones totales a 11,417.

Ahora considerando la información de la Tabla 1, donde se muestra el número de observaciones por cada clase, se selecciona de las clases, positive, negative y neutral, la clase con menor número de características y se toma ese como número de observaciones para cada clase de la distribución.

La eliminación de observaciones o instancias para el balanceo de clases es lo que se conoce como undersampling, y ha sido aplicado a la distribución de datos, de forma aleatoria para al final tener 2,290 observaciones de cada clase (véase ilustración 14).

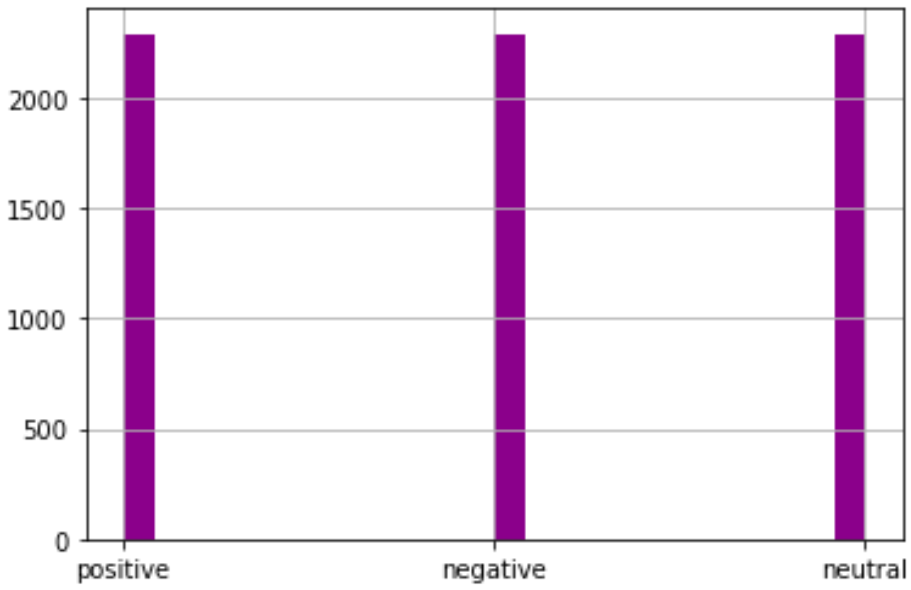

Ilustración 14 Histograma de la distribución de datos respecto a la variable objetivo (sentiment) después del balanceo de datos.

Retomando la estructura del conjunto de datos que fue recabado de Twitter, la agregación de las columnas correspondientes al proceso de limpieza

de texto (Twitter), la columna que contiene la polaridad de sentimiento, y las puntuaciones de cada polaridad, se tienen los siguientes campos en el conjunto de datos.

- **Id.** Número de identificación del Twitter
- **Username.** Nombre del usuario que publicó el Twitter.
- **Date**. Fecha de publicación del Twitter
- **Location**. Locación de publicación del Twitter.
- **Text**. Contenido del Tweet.
- **Hashtags**. Menciones en el Twitter seguidas del símbolo '#'.
- **CleanTweetS**. Contenido del Twitter después del proceso de limpieza.
- **Sentiment**. Polaridad de sentimiento asignado al contenido del Twitter.
- **positive_score**. Puntuación positiva del contenido del Twitter.
- **neutral_score**. Puntuación neutra del contenido del Twitter.
- **negative_score**. Puntuación negativa del contenido del Twitter.

Para reducir la dimensionalidad del dataset y así mismo parámetros en los algoritmos que se evaluaron, debido a que podrían generar mayor tiempo de ejecución, sólo se seleccionan las columnas de la ilustración 15, sobre las cuales se trabajará para realizar la tarea de análisis de sentimientos en diversos algoritmos.

	text	CleanTweetS	sentiment	positive_score	neutral_score	negative_score
	Jean For Genesis 8 and 8.1 Female\n\nConverted...	jean genesi femal convert daz studio credit wo...	positive	0.99	0.01	0.00
	A great read for the #innovation community! GL...	great read innov commun glide new creativ arti...	positive	1.00	0.00	0.00

Ilustración 15 Conjunto de datos o dataset final.

3 Capítulo 3

Implementación de estrategias para el análisis de sentimientos

Una característica importante para la realización de alguna tarea no supervisada es que las observaciones no están etiquetadas previamente, cómo es el caso del conjunto de datos. Sin embargo, se debía implementar una forma de tener datos etiquetados que ayude a evaluar el comportamiento de los modelos, y aunque no sean entrenados con estas etiquetas en los modelos no supervisados, permita comparar su desempeño y además generar modelos de manera supervisada.

Para lo anterior, se realizó el etiquetado del conjunto de datos mediante el apoyo de una herramienta en la nube, que permite categorizar texto, mediante el análisis de sentimientos, presentada en el capítulo anterior. Esta función de análisis de sentimientos proporciona etiquetas de polaridad de sentimiento (como "negativo", "neutral" y "positivo") basadas en la puntuación de confianza más alta encontrada por el servicio a nivel de oración y documento.

Es importante recordar que los procesos de limpieza y preproceso de datos es parte esencial para la realización de los algoritmos que se estarán valorando y/o evaluado, estos dos pasos se han realizado y mostrado en el capítulo previo a este. Por lo cual no se mostrarán nuevamente, pero se da por hecho que son parte esencial de cada implementación de los algoritmos.

3.1 Kmeans

La agrupación K Means es un algoritmo no supervisado, que se utiliza cuando se tienen datos no etiquetados, es decir, datos sin categorías, o no clasificados. El objetivo de este algoritmo es encontrar grupos en los datos, con el número de grupos o clústeres, definidos por el valor de K.

El algoritmo funciona de manera iterativa para asignar cada punto de datos a uno de los k clústeres. Los puntos de datos se agrupan en función de la similitud de las características. Obteniendo como resultados del algoritmo centroides de los clústeres. Cada centroide de un clúster es un conjunto de valores de características que definen los grupos resultantes.

En lugar de definir grupos antes de examinar los datos, la agrupación permite encontrar y analizar los grupos que se han formado orgánicamente.

El clustering K Means es una buena alternativa para empezar a explorar un conjunto de datos sin etiquetas.

Algoritmo básico:

1. Inicializar los clústeres centroides.
2. Asignar los puntos de datos a los clústeres.
3. Actualizar los centroides.
4. Repetir el paso 2 y 3 hasta que se cumpla la condición de parada.

Este algoritmo es muy sencillo de implementar gracias a algunas bibliotecas que se encuentran disponibles, dónde sólo es necesario introducir el valor de los parámetros bajo los cuales se entrenará el modelo, entre ellos, el número de clústeres y el número de interacciones.

Para determinar el número de clústeres óptimo existe un método denominado Elbow que consiste en trazar la variación explicada en función del

número de clústeres y elegir el codo de la curva como el número de clústeres a utilizar. En la figura siguiente se muestra el método Elbow aplicado al conjunto de datos, dónde se puede apreciar que el número de clústeres apropiado para este modelo es 3, el cual coincide con el número de categorías que se desean obtener.

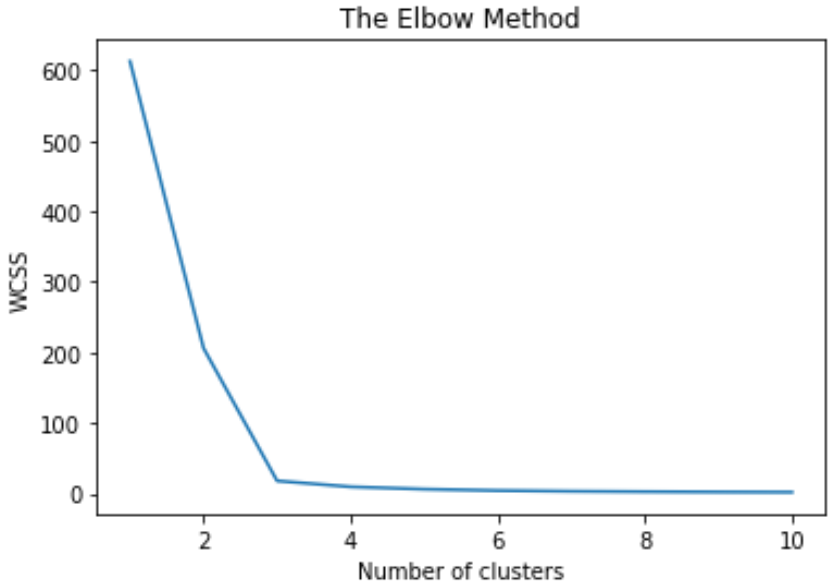

Ilustración 16 Elbow Method

Los algoritmos de clustering son fundamentalmente métodos de aprendizaje no supervisados, entonces las métricas de evaluación del modelo resultante se deben adaptar a este enfoque. Para las comparaciones de los desempeños se realizaron cuatros configuraciones distintas, con las siguientes especificaciones.

Kmeans con TF-IDF

Implementando Kmeas mediante una extracción de características con TF-IDF para el proceso de vectorización con 3 clústeres y 100 épocas.

Kmeans con LSA y TF-IDF

Implementando Kmeas mediante una representación de características con TF-IDF para el proceso de vectorización con 3 clústeres y 100 épocas más una reducción del espacio vectorial con SVD. En esta configuración primero se hace una reducción del espacio vectorial, para que k-means sea más estable mediante TruncatedSVD, dado que los resultados de SVD no están normalizados, se hace la normalización nuevamente para mejorar el resultado de KMeans. El uso de SVD para reducir la dimensionalidad de los vectores de documentos TF-IDF a menudo se conoce como latent semantic analysis (LSA) o por su traducción al español análisis semántico latente.

Kmeans con LSA y vectores Hashed

La misma configuración que la anterior, es decir, 3 clústeres con 100 épocas, más una reducción del espacio vectorial con SVD, pero ahora con Hashed para el proceso de extracción de características.

Hashes word es un espacio dimensional fijo, los vectores de conteo de palabras se normalizan para que cada uno tenga una norma l2 igual a 1, lo que parece ser importante para que k-means funcione en un espacio dimensional alto.

MiniBachKmeans con LSA

Ahora con la representación de Hashes words, la reducción de dimensionalidad del espacio vectorial se entrenó con un algoritmo de clustering similar a Kmeans, MiniBachKmeans bajo los mismos parámetros, 3 clústeres y 100 épocas.

En la figura 16 se tienen los resultados obtenidos y los tiempos de ejecución sobre los modelos de clustering que fueron entrenados. Todas las métricas de evaluación de agrupamiento tienen un valor máximo de 1 (para un resultado de

agrupamiento óptimo) siendo los valores más altos los mejores. Los valores del Índice Rand Ajustado cercanos a 0 corresponden a un etiquetado aleatorio.

Se puede observar que el comportamiento del clustering no es efectivo, incluso haciendo métodos de reducción de dimensionalidad. El modelo que mejor comportamiento alcanzó fue K-means con TF-IDF, sin embargo, los valores están por debajo del valor medio óptimo. Esto puede ser por la consistencia de los datos, donde de cierta manera existe una agrupación diferente a la que se desea, por polaridad de sentimiento y en cambio se tiene algo relacionado por los temas con los que fue recuperada la información o simplemente porque el algoritmo no es el adecuado para este problema.

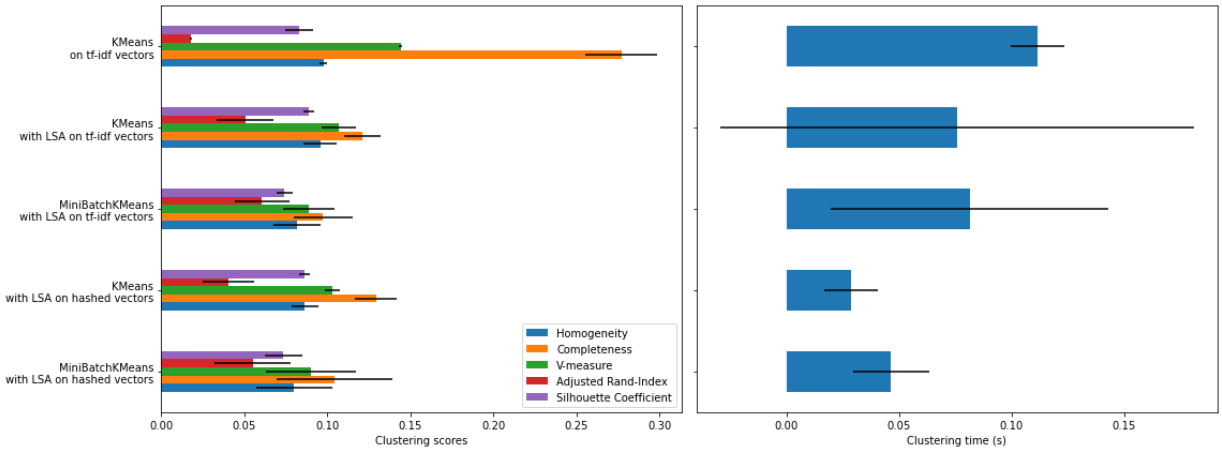

Ilustración 17 Resultados de las métricas para la evaluación de los modelos de Clustering con 3 clústeres.

Las cuatro configuraciones anteriores de clustering también fueron realizadas para 5 clústeres con el fin de ver el comportamiento de los modelos, los tiempos de ejecución son más grandes para la mayoría. En cuanto a las métricas también hay un aumento para algunas de ellas en ciertas configuraciones, lo que podría corroborar la idea anterior, de que no necesariamente las etiquetas estas relacionadas con el comportamiento de los clústeres.

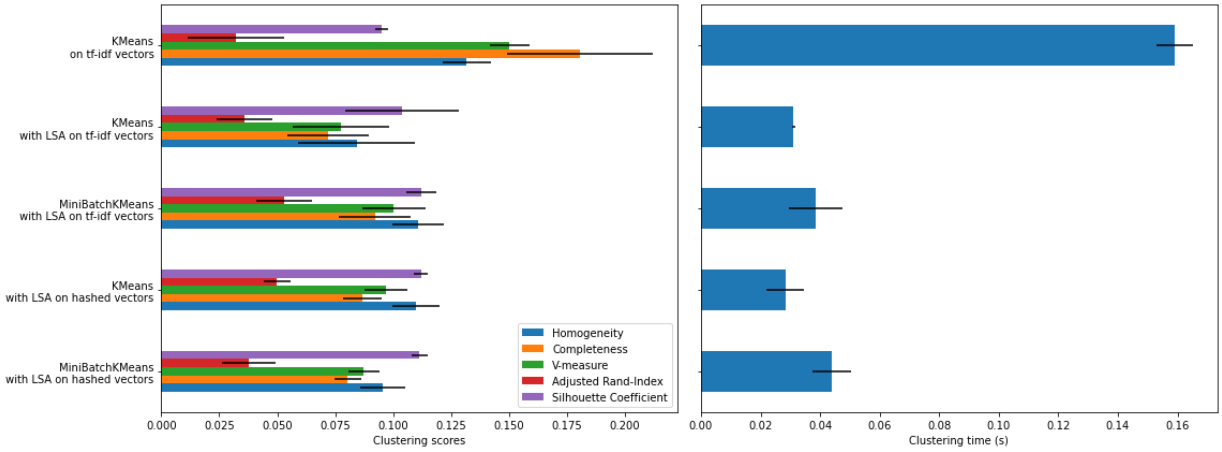

Ilustración 18 Resultados de las métricas para la evaluación de los modelos de Clustering con 5 clústeres

Al generar los 3 clústeres, se obtienen los clústeres de la ilustración 19, donde se puede observar una breve separación de los términos, por tema, y esta puede ser debido al hecho de que la extracción se realizó de ese modo, sin embargo, aunque se tiene una organización conceptual, no se ven definidos los sentimientos o alguna polaridad de sentimiento inferidos en cada clúster, a excepción del primero clúster donde, se agrupan algunas palabras que podrían considerarse como positivas.

```
Cluster 0: dl ia like amp know time need year day think realli im good feel love say want work today let
Cluster 1: learn ml artifici intellig ai machin deep use amp new data machinelearn python technolog develop work artificialinte
llig model datasci note
Cluster 2: ill pick ga winner repli data gg ayaka yst welkin oh gonna night drop end rt today base day ia
```

Ilustración 19 K-means con 3 clústeres.

Por lo anterior, también se implantó con 5 clústeres para observar el comportamiento de las agrupaciones. Aquí se puede ver más palabras coloquiales en los clústeres a diferencia de los 3 clústeres, qué en su mayoría se tenían palabras técnicas.

```
Cluster 0: ia sorri im day ive today good time morn gonna like work feel hi miss come amp exam guy school
Cluster 1: artifici intellig ai use technolog artificialintellig learn futur new machin human nft help develop read world amp r
obot phase chatbot
Cluster 2: ill pick ga winner repli gg ayaka yst welkin oh gonna night drop end rt base today day ia want
Cluster 3: learn machin deep data ai use python scienc code machinelearn book scientist datasci skill bigdata good model small
necessari core
Cluster 4: ml dl amp like new know need note play year think live say let day want love look peopl time
```

Ilustración 20 K-means con 5 clústeres.

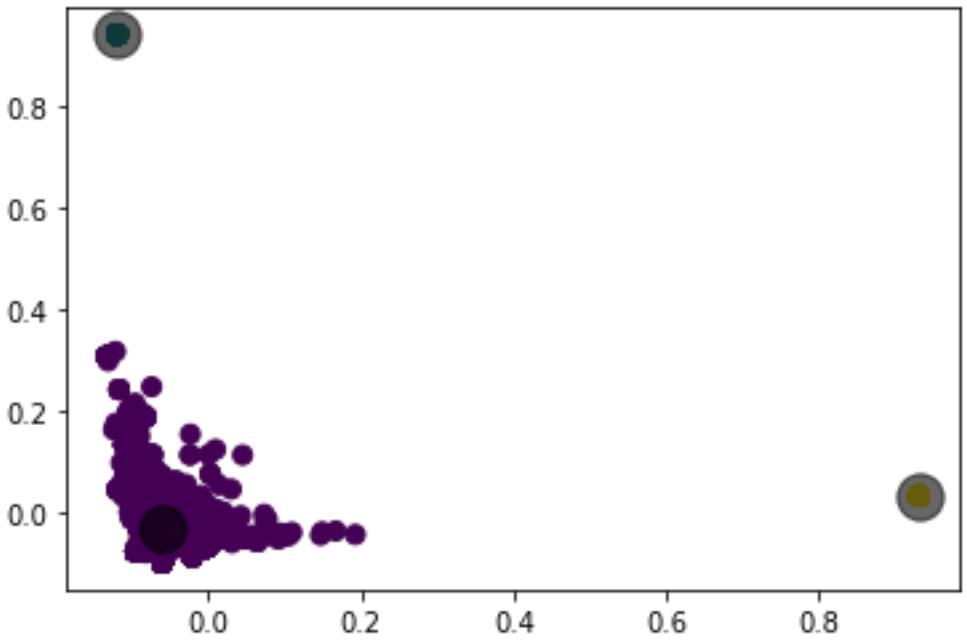

Ilustración 21 Ilustración 20 Dispersión de los centroides con 3 clústeres.

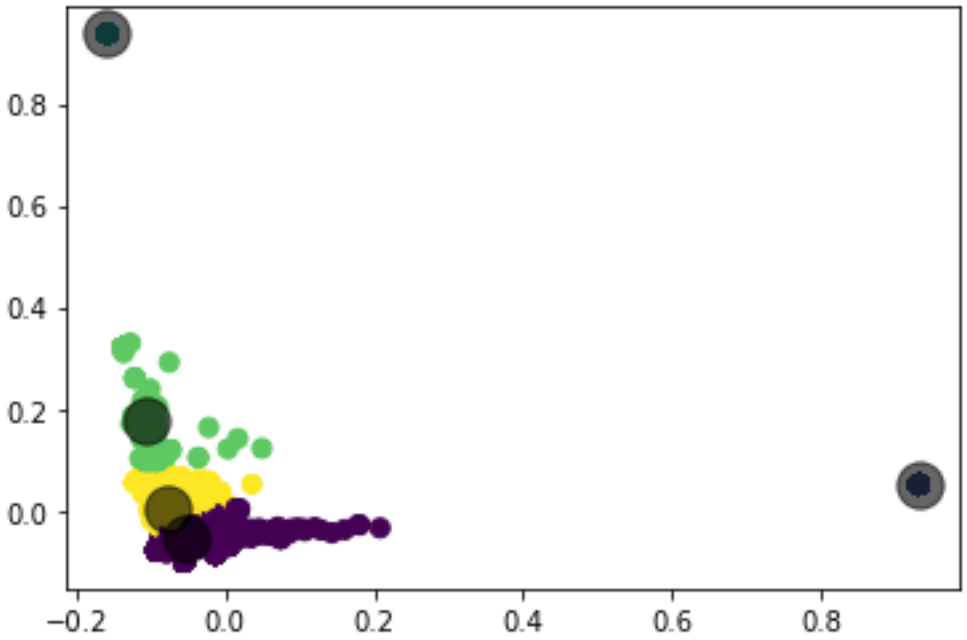

Ilustración 22 Dispersión de los centroides con 5 clústeres.

Dado que tanto KMeans como MiniBatchKMeans optimizan una función objetivo no convexa, no se garantiza que su agrupación sea óptima para un inicio aleatorio determinado. Aún más, en datos escasos de alta dimensión, como texto vectorizado mediante el enfoque de bolsa de palabras, k-means puede inicializar centroides en puntos de datos extremadamente aislados. Esos puntos de datos pueden permanecer en sus propios centroides todo el tiempo, como se observa en la distribución de los centroides con 3 clústeres.

3.2 Lbl2Vec

Lbl2Vec un algoritmo no supervisado para problemas de clasificación de documentos y recuperación de documentos [1]. Genera vectores de etiquetas, documentos y vectores de palabras automáticamente mediante la definición de palabras claves (keywords) predefinidas manualmente. Está basado en la idea de que muchas keywords semánticamente similares pueden representar un tema o clase, en este caso en particular, un sentimiento. En el primer paso, el algoritmo crea una incrustación conjunta de documentos y vectores de palabras. Una vez que los documentos y las palabras se incrustan en un espacio vectorial, el objetivo del algoritmo es aprender vectores de etiquetas a partir de palabras clave, estas previamente definidas, ocasionalmente de manera manual, donde cada conjunto de palabras clave representan un tema o clase. Finalmente, el algoritmo puede predecir la afiliación de documentos a una clase del vector de documento.

Algoritmo:

1. Obtiene palabras clave definidas manualmente para cada categoría de interés.
2. Crea embedded documents y vectores de palabras usando Doc2Vec
3. Encuentra los documentos de vectores que son similares a las keywords para cada sentimiento.
4. Limpia los vectores de outliers.
5. Calcula el centroide de cada documento limpio.
6. Calcula las similitudes del vector de etiquetas para cada vector de etiqueta y vector de documento.

El algoritmo promete buenos resultados, al utilizar una representación vectorial de las palabras y los documentos, además de utilizar el algoritmo Local Outlier Factor (LOF), un algoritmo para la identificación de outliers presentes en el conjunto de datos, a partir de la densidad de los vecinos más cercanos.

Este algoritmo es un algoritmo fácil de probar, gracias a la biblioteca Lbl2Vec, dónde, sólo es necesario preparar los datos y seleccionar los parámetros óptimos para el mejor desempeño de este. A continuación, se enlistan cada uno de los parámetros necesarios para el entrenamiento de un modelo de clasificación de documentos mediante esta biblioteca.

- Keywords list. Lista iterable de listas con palabras clave descriptivas para cada categoría, clase o tema.
- Tagged documents: Lista iterable de elementos gensim.models.doc2vec.TaggedDocument.
- Label names: Lista iterable de nombres personalizados para cada etiqueta. Los nombres de las etiquetas y las palabras clave de la misma clase deben tener el mismo índice.
- Similarity threshold. Valor de umbral para que solo los documentos con una mayor similitud con las palabras clave de descripción respectivas que este en este umbral se utilicen para calcular la incrustación de la etiqueta.
- Mínimo número de documentos: Número mínimo de documentos que se utilizan para calcular la incrustación de la etiqueta.
- Épocas: Número de iteraciones sobre el corpus.

Al ser un algoritmo no supervisado, se empleó el conjunto de datos sin etiquetas para la fase de entrenamiento. Creando así un corpus que contiene cada observación del conjunto de datos. En la siguiente ilustración se muestran los datos cargados en un dataFrame.

:

	CleanTweetS	text
0	intellig autom ia appli almost sector specif i...	Intelligent #automation (IA) applies to almost...
1	admin post daili amp hope rememb meantim total ia	admin ð□□□ will be posting not daily not here ...
2	class start ia	my class is starting, i'll be ia for a while !
3	bet head hand	@LemonIceTea_08 bet that some do Head in Hands
4	honor presid india present padma shri resid ba...	Honorable President of India presented Padma S...
...	...	...
11995	hey iowa iowa state midst histor turnaround ce...	ð□□£ Hey Iowa! ð□□□\n\nIowa State is in the mi...
11996	raini day third artwork collect avail foundat ...	Rainy Daysð□□§â□¡ï¸□.\nThe Third 1/1 3D Artwor...
11997	wordol ima ml	WorDOL 41 5/6\n\nð□□¨â¬□â¬□â¬□â¬□\nð□□¨ð□□¨â¬□...
11998	time talk like critic mass peopl final realiz ...	@Mourning_Time @ilda_talk More like;\n\nA crit...
11999	kiki tsubasa happi support amaz japanes projec...	KIKI and Tsubasa are happy to support @galvers...

12000 rows × 2 columns

Ilustración 23 Carga del conjunto de datos para Lbl2Vec

Para realizar el entrenamiento de este algoritmo se realizó una división del conjunto de datos, con el fin de evaluar el comportamiento del algoritmo mediante F1 Score. Por lo qué, se definió el 30% de observaciones del conjunto de datos de prueba (Test Set) y el 70% para el conjunto de entrenamiento (Train Set).

La lista de palabras claves fue construida a partir del Lexicón de Sentiment Analysis VADER (Valence Aware Dictionary and Sentiment Reasoner) [6], mediante el indexado de la categorización de palabras: positivas, neutras y negativas. VADER es una herramienta de análisis de sentimientos basada en reglas y léxico, específicamente para los sentimientos expresados en las redes sociales, pero que incluso puede funcionar bien en textos de otros dominios.

Finalmente, el algoritmo se entrenó con una lista de palabras clave que contiene la relación de la tabla 2, con cierto número de palabras clave por clase, o polaridad de sentimiento, entre las palabras también se encuentran, conjuntos de caracteres asociados a emojis, cómo ':)',':|' ':/', '(-%'',') ':', ')-':', entre muchos otros más, de igual forma están etiquetados según la polaridad de sentimiento que representan.

Tabla 2 Relación de palabras clave del indexado creado con el lexicón de Sentiment Analysis VADER.

	Positive	Neutral	Negative
Núm. Palabras clave	2986	703	3813

Para los otros parámetros del algoritmo, se definieron los valores de la tabla 3.

Tabla 3 Parámetros para Lbl2Vec

Parámetro	Valor
similarity_threshold	0.43
min_num_docs	2000
epochs	100

Los resultados de este algoritmo se evaluaron con la métrica, F1 score obteniendo un valor de 0.7403697617091208, que se puede considerar un valor aceptable tomando en cuenta, que es un algoritmo no supervisado, y qué el conjunto de datos fue recuperado de internet, es decir, que se ha tratado con los datos desde inicio a fin.

Tabla 4 F1 Score con Lbl2Vec

	Train	Test
F1 Score	0.7403697617091208	0.7150849150849151

3.3 LSTM

Las redes neuronales recurrentes (RNN) son una forma de redes neuronales artificiales muy popular en los últimos años, se caracterizan por memorizar secuencias de patrones de entrada de longitud arbitraria mediante la captura de conexiones entre tipos de datos secuenciales. Sin embargo, debido a la falla de los gradientes estocásticos, los RNN no pueden detectar dependencias a largo plazo en secuencias largas. Se propusieron varios modelos RNN, en particular LSTM, para abordar este problema. Las redes LSTM son extensiones de RNN diseñadas para aprender datos secuenciales (temporales) y sus conexiones a largo plazo con mayor precisión que las RNN estándar. Se usan comúnmente en aplicaciones de aprendizaje profundo, para problemas con datos no estructurados en reconocimiento de voz, procesamiento de lenguaje natural, etcétera.

En la ilustración de abajo se muestra una unidad LSTM típica que se repite a lo largo de toda la secuencia.

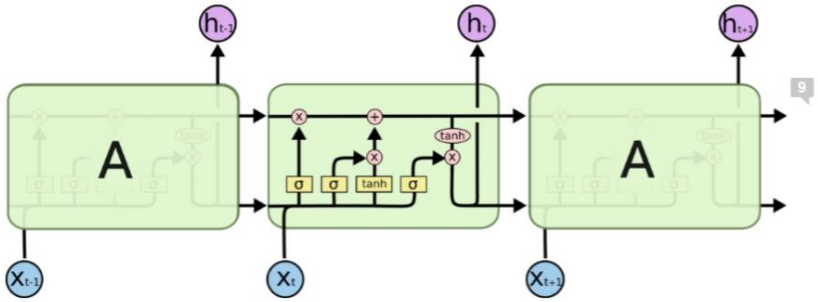

Ilustración 24 El módulo repetitivo en un LSTM que contiene cuatro capas, imagen recuperada de [18].

3.3.1 Análisis de sentimientos con LSTM

LSTM es un algoritmo de aprendizaje supervisado, por lo que, se trabajó con el conjunto de datos etiquetado previamente con Azure. El primer paso por

realizar para generar el modelo de LSTM es: cargar los datos, cómo estos ya han sido previamente procesados, están listos para ser utilizados en el modelo de LSTM.

Primero es necesario hacer la división del conjunto de datos, en train y test. Esta división corresponde al 70% para el conjunto de entrenamiento y 30% para el conjunto de prueba.

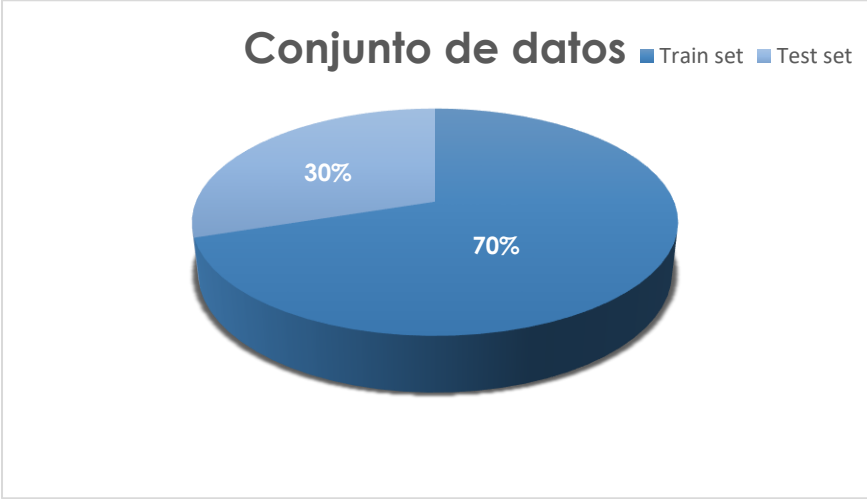

Ilustración 25 División de conjunto de datos en Train set y Test set.

Antes de ingresar al modelo LSTM, los datos deben pasar por el proceso de padding y tokenización.

- **Tokenización**: La API de tokenizer incorporado de Keras se ha implementado en el conjunto de datos, este divide las oraciones en palabras y crea un diccionario de todas las palabras únicas encontradas y sus números enteros asignados de forma única. Cada oración se convierte en una matriz de números enteros que representan todas las palabras individuales presentes en ella.
- **Sequence Padding**: Convierte un vector en una matriz que representa cada oración en el conjunto de datos, se llena con ceros a la izquierda para hacer que el tamaño de la matriz sea igual a la máxima longitud y así asegurar que todas las colecciones tengan la misma longitud.

3.3.2 Componentes del modelo LSTM

Antes de detallar el modelo de LSTM, se describe brevemente cada uno de los componentes del modelo.

3.3.2.1 Embedding layer

La embeddind layer permite hacer la representación de cada documento, mediante Word Embeddings, convierte cada palabra en un vector de longitud fija de tamaño definido. El vector resultante es denso y tiene valores reales en lugar de solo 0 y 1. La longitud fija de los vectores de palabras ayuda a representar las palabras de una mejor manera junto con dimensiones reducidas. En modelos de Deep learning se puede hacer el entrenamiento de los vectores, o definir un modelo preentrenado como los vistos en el capítulo 1. Para la definición de esta capa, se hizo el preentrenamiento de las incrustaciones mediante GloVe.

3.3.2.2 LSTM bidirectional

Los LSTM bidireccionales entrenan dos en lugar de un LSTM en la secuencia de entrada. El primero en la secuencia de entrada y el segundo en una copia invertida de la secuencia de entrada. Esto puede proporcionar contexto adicional a la red y dar como resultado un aprendizaje más rápido e incluso más completo sobre el problema.

3.3.2.3 Spatial 1D version of Dropout

Realiza la misma función que Dropout, sin embargo, descarta mapas de características de una dimensión completos en lugar de elementos individuales. Si los marcos adyacentes dentro de los mapas de características están fuertemente correlacionados, entonces Dropout no regularizará las activaciones y de lo contrario solo dará como resultado una disminución efectiva de la tasa de aprendizaje. En este caso, SpatialDropout1D ayudará a promover la independencia entre los mapas de características.

3.3.2.4 Dense layer

La capa densa implementa la operación de salida, la cual es igual a la función activación del producto punto de la entrada, y el kernel más el sesgo, el kernel es una matriz de pesos y el sesgo es un vector de sesgo, ambos creados por la capa.

3.3.2.5 Dropout

La capa Dropout establece aleatoriamente las unidades de entrada en 0 con una frecuencia de la tasa (definida en los parámetros) en cada paso durante el tiempo de entrenamiento, lo que ayuda a evitar el sobreajuste.

3.3.2.6 Funciones de activación

Las funciones de activación introducen la no linealidad en la red. Sin linealidad, la red estaría realizando asignaciones lineales entre la entrada, que no sería nada sino una ecuación lineal multivariada. Las funciones de activación controlan el umbral que decide lo que la neurona proporcionaría como salida. Existe una variedad de funciones de activación en la literatura, el uso de ellas depende del problema y el propósito en el modelo, en este apartado sólo se describen las funciones de activación que integran el modelo entrenado.

3.3.2.6.1 ReLU

Rectified linear unit (ReLU) es probablemente la función de activación más utilizada, está dada por la función de la ecuación 1. Lo que hace que ReLU sea muy utilizada por su fórmula muy simple y que, al mismo tiempo, supera los problemas asociados con las funciones sigmoide y tanh. Sin embargo, si se observa el gráfico de la ilustración 26 de la función ReLU, se puede notar que, para valores negativos, la neurona con ReLU como función de activación nunca se activaría. Este problema se puede resolver usando leaky ReLU como función de activación.

$$ReLU\ (x) = max\ (0\ , x) \qquad \textit{Ecuación 1. Función ReLU}$$

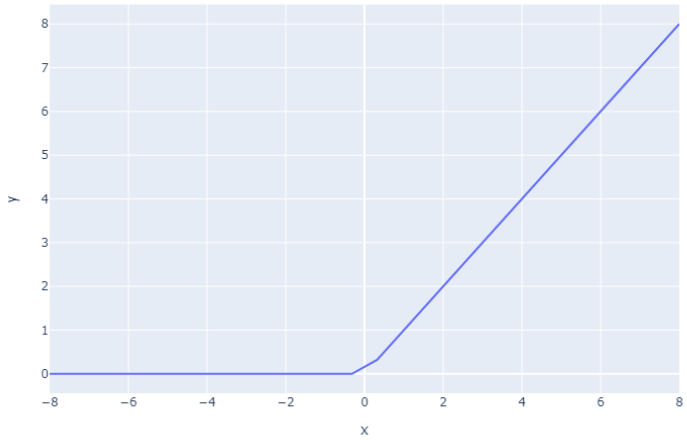

Ilustración 26 Gráfica de la función de activación ReLU.

3.3.2.6.2 Sigmoid.

La función de activación Sigmoid restringe la salida en un rango entre 0 y 1. Esta función de activación es la adecuada para una clasificación binaria. Está definida usando la siguiente ecuación matemática:

$$Sigmoid\ (x) = \frac{1}{1+ e^{-x}}$$ *Ecuación 2. Función Sigmoid*

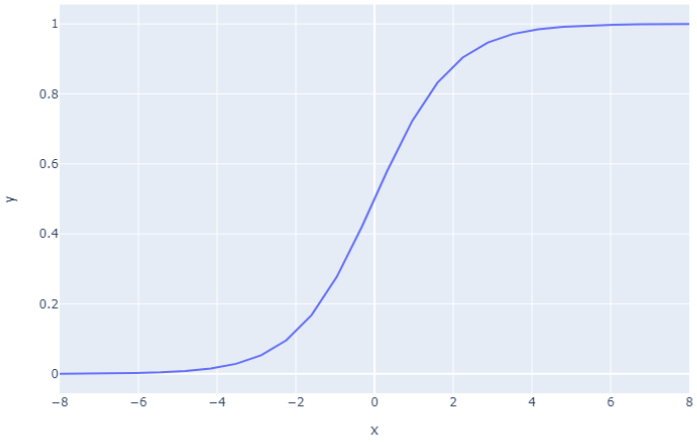

Ilustración 27 Gráfica de la función de activación Sigmoid.

3.3.3 Definición del modelo de LSTM

Para la implementación de LSTM, se hizo un modelo secuencial de Keras. El cual es una pila lineal constituida por capas, para la definición del modelo se incorporaron las siguientes capas:

- Una **embedding layer** de dimensión 45 que convierte cada palabra de la oración en un vector denso de longitud fija de tamaño 45. La dimensión de entrada se establece como el tamaño del vocabulario y la dimensión de salida (45). Por lo tanto, cada palabra de la entrada se representará mediante un vector de tamaño 45.
- Una capa de **Spatial 1D version of Dropout**.
- Una capa LSTM bidireccional de 45 unidades.
- Una **capa densa** totalmente conectada de 100 unidades con activación ReLU.
- La segunda **capa densa** totalmente conectada de 100 unidades con activación ReLU.
- Una **capa densa** de 3 unidades con salida de activación softmax, para la salida del modelo.

En la siguiente figura se puede observar la estructura del modelo generado con las capas descritas anteriormente, cada fila de la figura representa una capa del modelo. En la última capa densa con activación softmax se tiene cómo salida 3 neuronas, correspondientes a cada una de las clases.

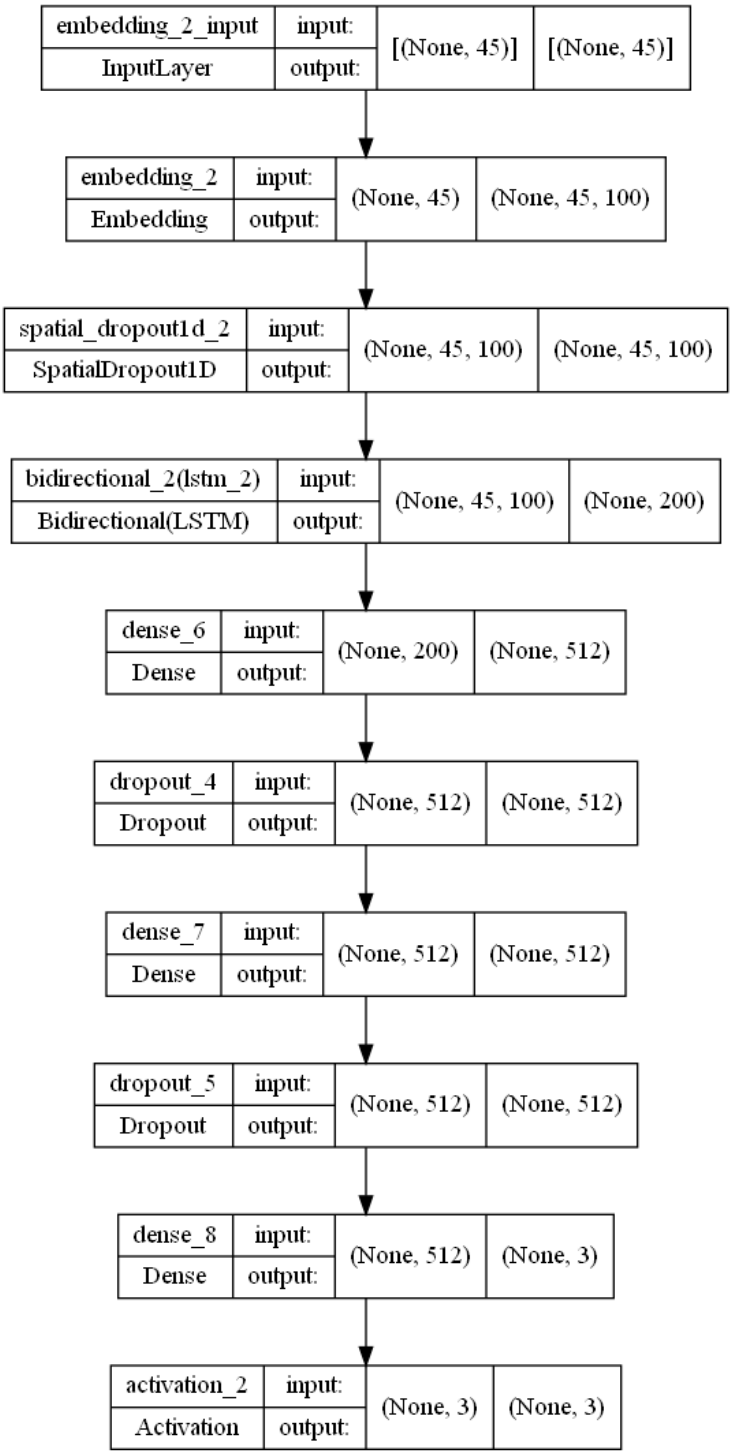

Ilustración 28 Gráfico del modelo de clasificación LSTM definido.

En la ilustración 26, se muestra un resumen de los parámetros del modelo, indicando el Shape o tamaño de las matrices con las que se realizará el

entrenamiento y el número de parámetros por cada capa de la red de LSTM, obteniendo un total de parámetros de 1,544,507.

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 45, 100)           1016600

 spatial_dropout1d_2 (Spatia  (None, 45, 100)          0
 lDropout1D)

 bidirectional_2 (Bidirectio  (None, 200)              160800
 nal)

 dense_6 (Dense)             (None, 512)               102912

 dropout_4 (Dropout)         (None, 512)               0

 dense_7 (Dense)             (None, 512)               262656

 dropout_5 (Dropout)         (None, 512)               0

 dense_8 (Dense)             (None, 3)                 1539

 activation_2 (Activation)   (None, 3)                 0

=================================================================
Total params: 1,544,507
Trainable params: 527,907
Non-trainable params: 1,016,600
_________________________________________________________________
```

Ilustración 29 Resumen del modelo de clasificación LSTM definido.

Después de realizar pruebas con el desempeño de diferentes modelos de LSTM, este modelo fue el final, se realizó el entrenamiento con 200 épocas sobre el conjunto de datos de entrenamiento, y así mismo se realizó la validación con el conjunto de prueba, obteniendo cómo resultado los valores de la tabla 5, usando como métrica a Accuracy.

Tabla 5 LSTM Accuracy

	Train	Test
Accuracy	0.70	0.62

Se puede apreciar que el desempeño es mejor para el conjunto de entrenamiento, la diferencia no es mucha entre ambos valores, por lo que no se podría considerar como un problema de overfitting. En las ilustraciones 30 y 31 se muestra el comportamiento del accuracy y la pérdida a través del transcurso de las épocas en el entrenamiento del modelo.

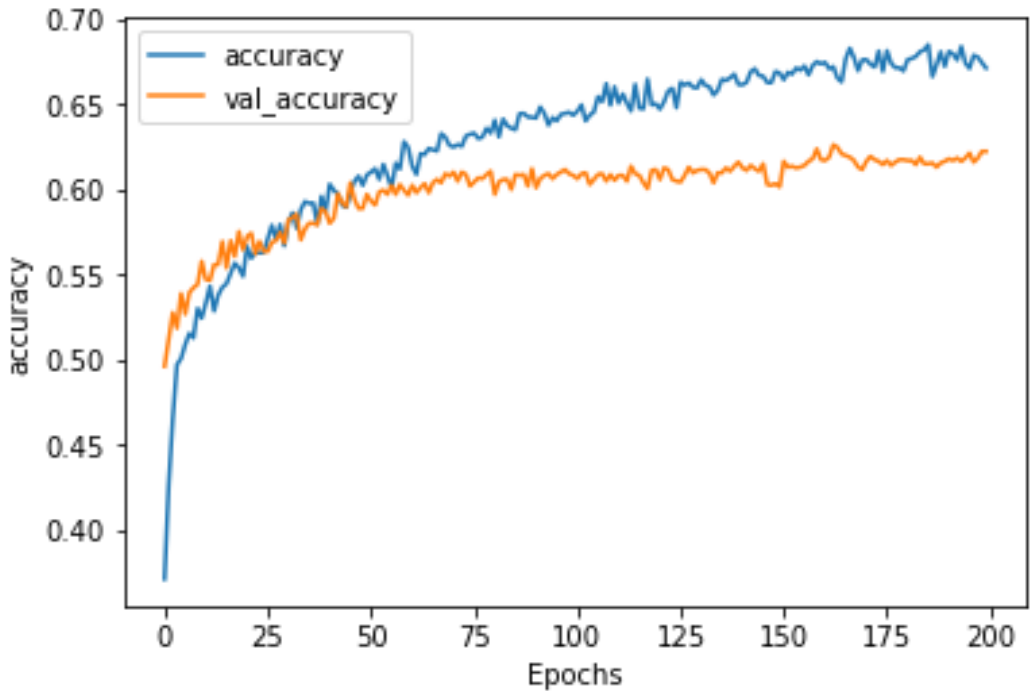

Ilustración 30 Accuracy del modelo vs Accuracy del conjunto de prueba.

La curva de aprendizaje del conjunto de entrenamiento muestra que tan bien el modelo aprendió, mientras que la curva de aprendizaje del conjunto de validación representa que tan bien el modelo se comporta con nuevas observaciones. Un buen ajuste se identifica por una pérdida (loss) de entrenamiento y validación que disminuye hasta un punto de estabilidad con una brecha mínima entre los dos valores de pérdida finales. En la ilustración 31 que corresponde al comportamiento de la pérdida del modelo, se puede observar

que, si hay una brecha mínima entre los dos valores de pérdida, sin embargo, no se ve bien definido el punto de equilibrio.

Este comportamiento del modelo se puede deber a distintos factores, desde la integridad de los datos hasta la configuración del mismo modelo. Se realizaron pruebas incorporando más capas al modelo, pero se presentaban problemas de overfitting, así mismo también sobre este modelo se modificaron el número de épocas para el entrenamiento, pero al aumentarlas también se presentaban problemas de overfitting, por lo que la configuración mostrada fue la que mejor comportamiento logró. En futuros trabajos se podría experimentar aún más en la construcción del modelo de LSTM, para mejorar los valores de precisión de este.

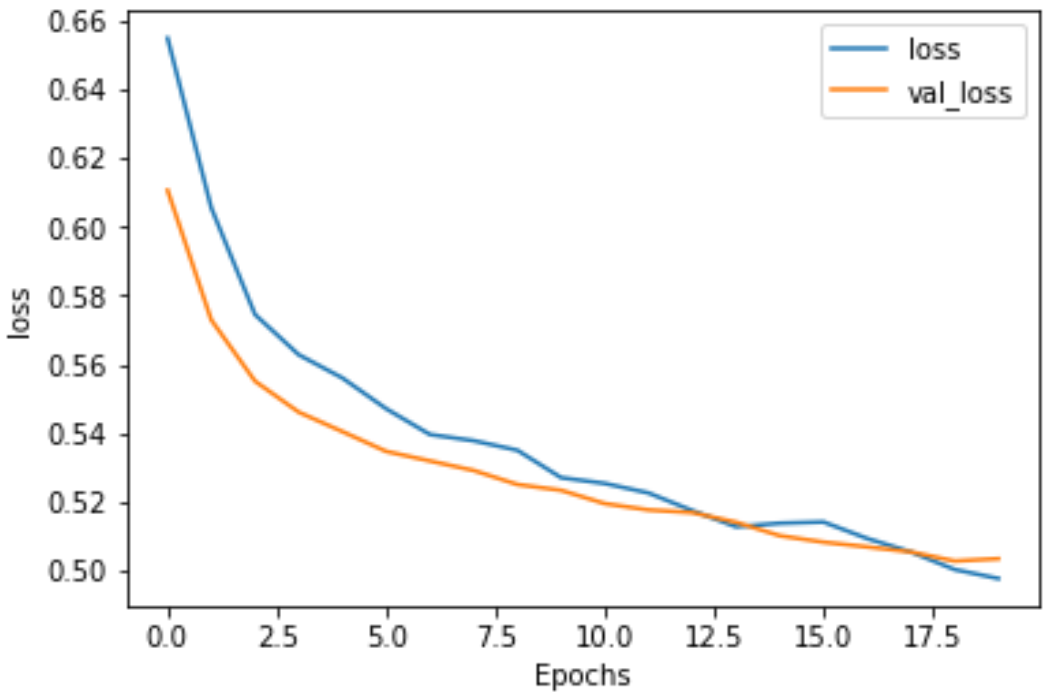

Ilustración 31 Perdida del modelo vs Perdida del conjunto de prueba.

4 Capítulo 4

Resultados del análisis de sentimientos

En esta sección se proporcionan y analizan una serie de gráficos qué han sido preparados y procesados mediante técnicas de minería de datos, en R y Python, con el fin de mostrar que tan certera fue la extracción de datos, es decir, si los Tweets realmente contienen opiniones o contenido relacionado con la inteligencia artificial, y de ser así, ¿cómo es el comportamiento de las opiniones?, ¿cómo perciben los usuarios la inteligencia artificial en Twitter? ¿qué tan formal es el lenguaje ocupado para hacer referencia a esta disciplina en redes sociales? ¿qué emociones son expresadas en los Tweets?, etcétera. Todas estas respuestas se pueden obtener a partir de la exploración y análisis de los datos.

El análisis de datos consiste en la recolección, transformación y organización de datos, con el propósito de obtener conclusiones, hacer predicciones y ayudar en la toma de decisiones, esta sección, no es más que un proceso de análisis de datos que permitirá resolver las interrogantes anteriores y mostrar los resultados obtenidos.

Es importante resaltar que este análisis se ha realizado, después de la clasificación de sentimientos, pues con esos datos, se podrá valorar realmente el comportamiento de los usuarios, mediante sus Tweets.

La clasificación obtenida mediante Azure es relevante, y ha sido tomada cómo punto de referencia, porque esta fue punteada a partir de un léxico proporcionalmente más grande al que se ha creado en este trabajo para los modelos valorados, lo que introduce mayores valores de presión.

En primer lugar, se tiene que a partir de los resultados obtenidos con la clasificación de Text Analytics de Azure, el 51 % de las observaciones fueron categorizadas como neutras, el 25 % positivas, 19 % como negativas y sólo el 5 % como mixed que es una combinación entre positivas y negativas. De lo que se puede inferir que mayoritariamente se tiene un flujo positivo de texto en los Tweets, teniendo en segundo lugar los comentarios positivos, y por otra parte, las opiniones neutras, pueden considerarse relativamente "buenas", en un ambiente dónde regularmente se tiene mucho contenido negativo, cómo Twitter.

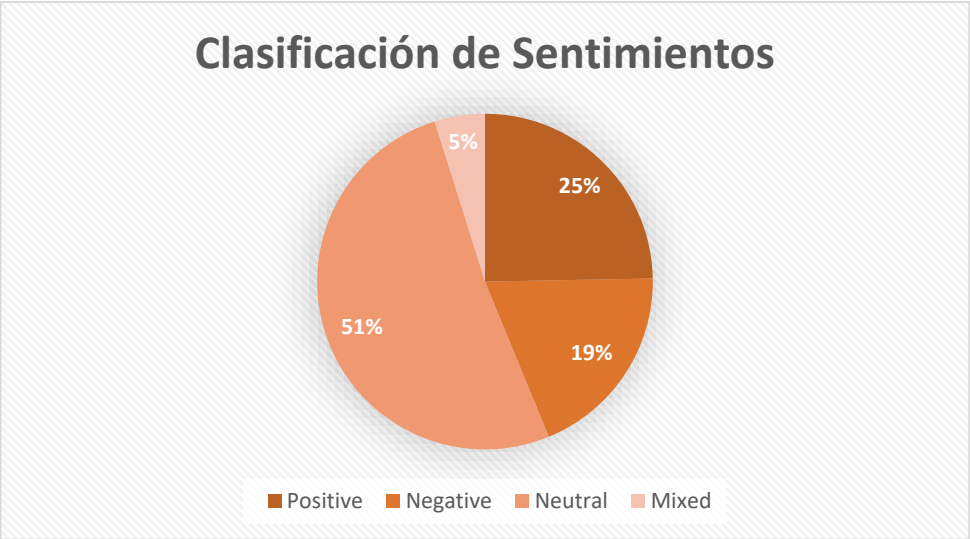

Ilustración 32 Gráfica de porcentajes de cada polaridad se sentimiento con la clasificación de Azure.

Después de mostrar datos muy generales, mediante la gráfica de barras, lo que es un poco más descriptivo. Ahora se tratará de inferir las respuestas de las preguntas iniciales, a partir de la extracción y exploración de palabras clave del corpus obtenido del conjunto de datos.

4.1 Extracción de palabras clave

Los procesos de extracción de palabras claves en minería de texto, son muy eficientes, permiten profundizar en el contenido textual de los conjuntos de datos.

Existen muchas técnicas y herramientas que se pueden emplear en la extracción de palabras clave, y en este pequeño apartado se muestra la exploración de algunas de estas técnicas con el fin responder a las preguntas planteadas al inicio de este capítulo.

Primero se iniciará con la extracción de las palabras que más han sido utilizadas en los contenidos de los Tweets, es decir, las palabras con mayor frecuencia. Los resultados obtenidos se pueden visualizar fácilmente en un histograma, ver ilustración 33. Entre estas palabras frecuentes se encuentran: learn, ml, machine, ai, deep, inteligent, winner. De lo que se puede deducir la respuesta a la primera pregunta establecida, efectivamente, la adquisición de datos fue certera, pues realmente tiene contenido relacionado con términos, temas y opiniones sobre "Inteligencia artificial". Según esta representación la palabra más frecuente en el corpus es "learn" lo que directamente se puede relacionar con Mahine Learning.

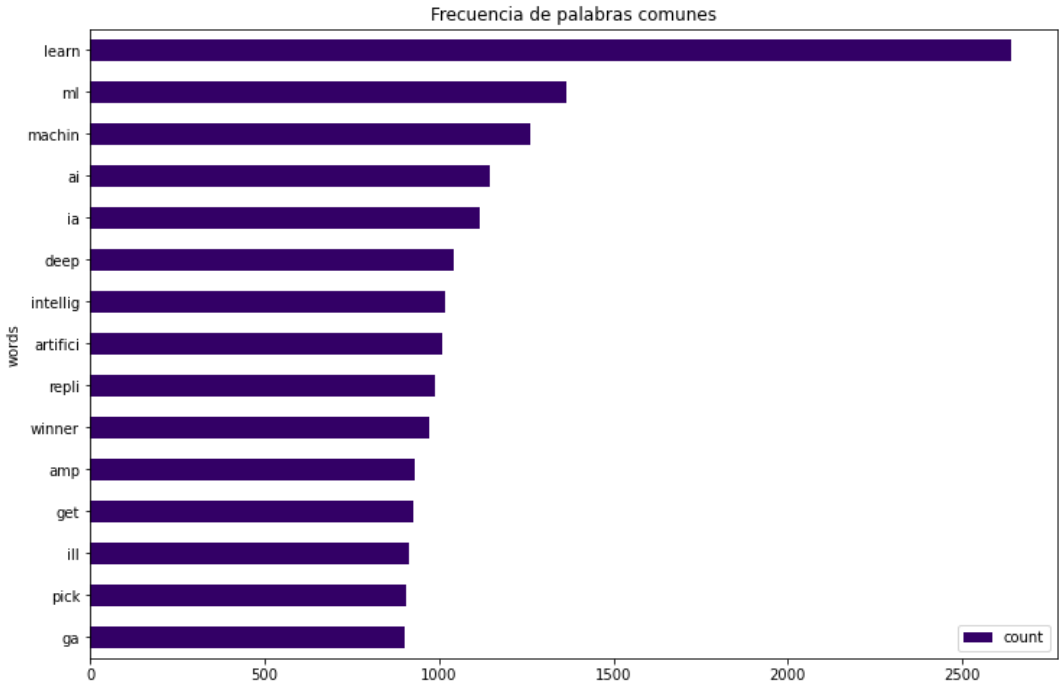

Ilustración 33 Histogramas de las palabras más frecuentes del vocabulario del conjunto de datos.

Entre la inmensa cantidad de paquetes disponibles en R para PLN, se encuentra udpipe [14], un paquete que facilita el procesamiento del texto permite tokenizar, etiquetar, lematizar y analizar dependencias. Con esto no sólo se pueden visualizar las palabras más frecuentes, ahora a partir de este etiquetamiento, se puede obtener la frecuencia de los sustantivos y adjetivos encontrados en el corpus.

La ilustración 34 muestra la frecuencia de sustantivos obtenidos con udpipe, donde se muestran incluso de algunos que aún no eran relevantes en comparación de las palabras más frecuentes mostradas en la ilustración 33. Al extraer sólo sustantivos, se pueden encontrar términos más relevantes y certeros que son empleados en "Inteligencia Artificial" como: "data", "code", "python", "scient", "collect", "program", etcétera. Estos sustantivos son más cercanos a un trabajo de inteligencia artificial, y no tan superficiales cómo los que se percibían tomando en cuenta solamente la frecuencia de las palabras.

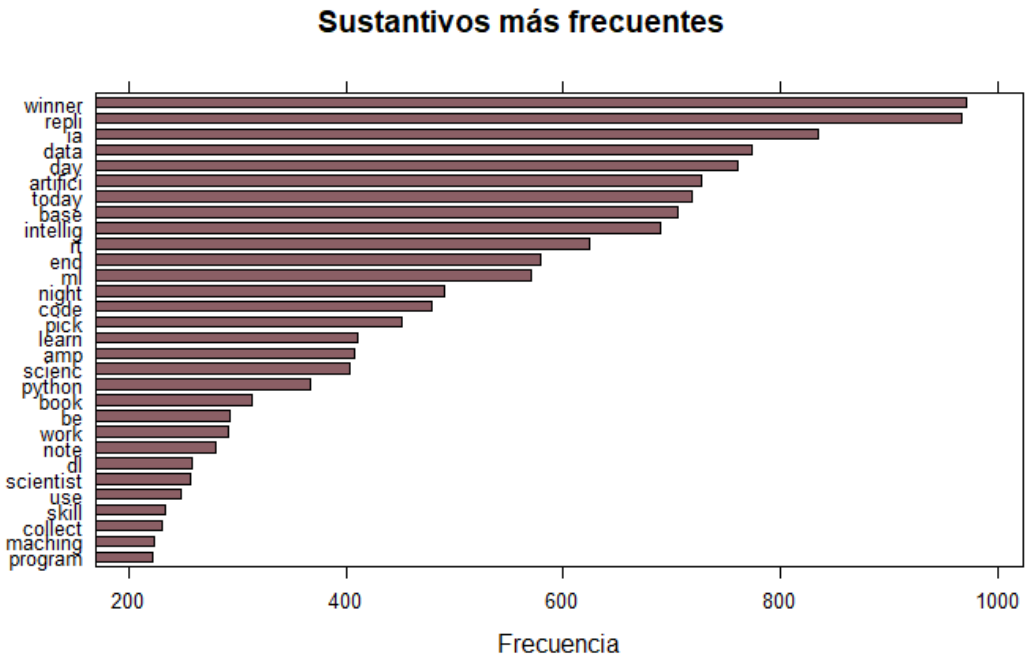

Ilustración 34 Frecuencias de los sustantivos del corpus.

4.1.1 Co-ocurrencias de sustantivos y adjetivos

Además de hacer el etiquetamiento de sustantivos udpipe también etiqueta los adjetivos por cada documento del conjunto de datos. Con esto se puede obtener una expresión, integrada por varias palabras, mediante la extracción de las colocaciones (palabras que se suceden), las co-ocurrencias de palabras dentro de cada oración (que tan frecuentemente se encuentran dos palabras incluso si son omitidas 2 palabras en el medio). Esto mediante la selección de los sustantivos y adjetivos por cada observación.

En la ilustración 35 se visualizan las co-ocurrencias mediante un gráfico de red para los 90 sustantivos y adjetivos co-ocurrentes más frecuentes donde la línea púrpura muestra la relación encontrada entre estas. En la parte inferior de la ilustración se puede observar una red donde varias co-ocurrencias están relacionadas entre sí al ser formadas con una misma palabra en común "data", y al formar las relaciones, se encuentra más contenido relevante cómo: data scientist, small data, good data, data subset, entre otras.

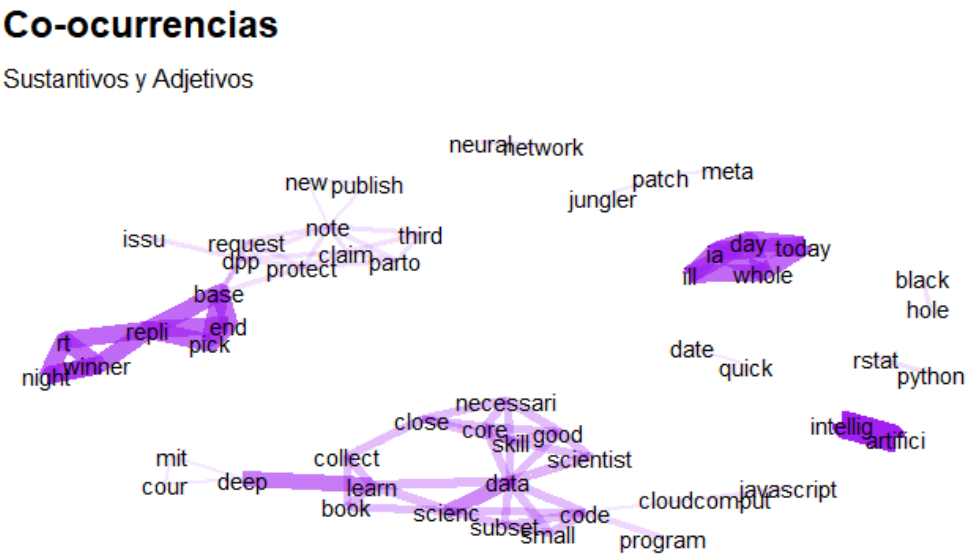

Ilustración 35 Co-ocurrencias de los sustantivos y adjetivos más frecuentes.

4.1.2 TextRank

TextRank es un modelo de clasificación basado en gráficos para el procesamiento de texto que se puede usar para encontrar las oraciones más relevantes en el texto y también para encontrar palabras clave mediante la construcción de una red de palabras, al observar si las palabras se suceden unas a otras y adicionalmente de esta red, se aplica el algoritmo 'Google Pagerank' para extraer palabras relevantes, después las palabras relevantes que se suceden se combinan para obtener las oraciones más importantes [15].

Para encontrar las oraciones más relevantes en el texto, se construye un gráfico donde los vértices del gráfico representan cada oración en un documento y los bordes entre oraciones se basan en la superposición de contenido, es decir, calculando la cantidad de palabras que tienen 2 oraciones en común.

Con esta técnica al hacer la extracción de oraciones importantes, se puede obtener un resumen del corpus, retomando solo las oraciones más importantes. En la ilustración 36 se muestra una nube de palabras construida a partir de las oraciones con al menos una frecuencia de 20 en el corpus compuesta por 8 gramas. También se realizaron pruebas modificando el número de gramas y el número mínimo de frecuencia, pero esta configuración es la más significativa para el corpus. Entre las oraciones más resaltadas se encuentra una particularmente que se acerca más a un comentario proveniente de la comunidad científica "new note rnn amp tranformer lecture", "Training GAN, RNN, Python, pytorch", incluyendo otras como "learn course mit" y "Coding Udemy feature course maching learning" que podrían ser publicidad para cursos de Machine Learning. Pero en general, si hay varios temas relevantes de inteligencia artificial de los cuales se hablan en Twitter. Destacando principalmente quizá la popularidad de lenguajes o bibliotecas utilizadas por la comunidad, ofertas de cursos en línea y comentarios de trabajos de inteligencia artificial. Algo interesante es que, en este resumen, se puede apreciar en su mayoría un

lenguaje apropiado, en dos sentidos, relevante a los temas y segundo pacifista, lo que da una buena impresión.

Ilustración 36 Nube de oraciones importantes extraídas con TextRank

4.1.3 Nubes de palabras positivas y negativas

Una nube de palabras es un gráfico que ya se ha empleado para representar palabras o texto relevante dentro de un corpus. En esta sección se muestra una nube de palabras construida a partir del subconjunto de observaciones que fueron clasificadas con polaridad positiva por Azure. La ilustración 37 muestra la correspondiente nube de palabras, dónde el tamaño de fuente de las palabras es proporcional a la frecuencia de esta dentro del corpus. Entre las palabras extraídas, la mayoría son tecnicismos de Inteligencia artificial,

pero algunas otras como" good data", "skill good", "feel free" están directamente asociadas a palabras que se pueden encontrar en un texto o documentos con polaridad positiva.

Ilustración 37 Word cloud del vocabulario de las observaciones (Tweets) clasificadas cómo positivas.

Ahora para la nube de palabras construidas a partir del subjuntivo de observaciones clasificadas con polaridad negativa, se tienen menos palabras técnicas con respecto a la nube de palabras anterior. Lo cual es una observación interesante, pues de alguna manera el modelo, clasificó la terminología asociada al campo de la inteligencia artificial como palabras con score positivo. Pero al igual que la anterior, también hay palabras que generalmente suelen asociarse a comentarios negativos, como: "gonna drop", "replay end", entre otras. Esta nube de palabras se muestra a continuación en la siguiente ilustración, de igual

manera el tamaño de la fuente es una asociación con la frecuencia de la palabra.

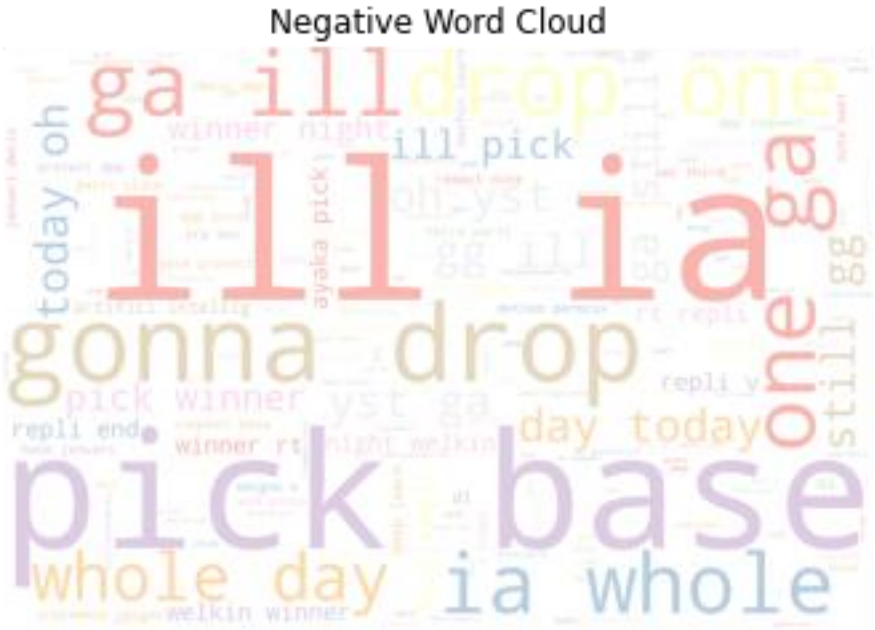

Ilustración 38 Word cloud del vocabulario de las observaciones (Tweets) clasificadas cómo negativas.

4.2 Análisis de sentimientos con emociones

En el capítulo 2 del estado del arte se dio una pequeña introducción de lo que es el análisis de sentimientos en PLN, haciendo un poco de énfasis en la clasificación por emociones y por polaridad de sentimiento, también se señaló que la categorización de emociones aún es algo complejo para las máquinas. En el capítulo 3 se obtuvieron polaridades de sentimientos, ahora en esta sección se muestra un pequeño trabajo realizado con el conjunto de datos para extraer las emociones encontradas en los Tweets del conjunto de datos, para lo cual se hizo uso del paquete syuzhet de R.

Syuzhet es un paquete que está integrado por cuatro diccionarios de opiniones y proporciona un método para acceder a una herramienta de

extracción de opiniones robusta, pero computacionalmente costosa, desarrollada en el grupo PLN de Stanford [7].

Algo interesante de Syuzhet es que clasifica las emociones de acuerdo con clasificación la rueda de Plutchik, que está integrada por ocho emociones centrales o primarias, que en distintos niveles pueden derivar otras emociones o estados.

Para ver cuáles son las emociones con mayor presencia en el corpus integrado por los Tweets, se presenta un gráfico de barras, con las emociones obtenidas por syuzhet. Para cada barra se suman todos los valores de la columna correspondiente a la emoción. La grafica de puntuaciones de sentimientos muestra que hay más observaciones positivas respecto de las demás, además las emociones como expectación (anticipation) y confianza (trust), también tienen más cantidad de observaciones respecto de Repugnancia (disgust) y tristeza (sadness).

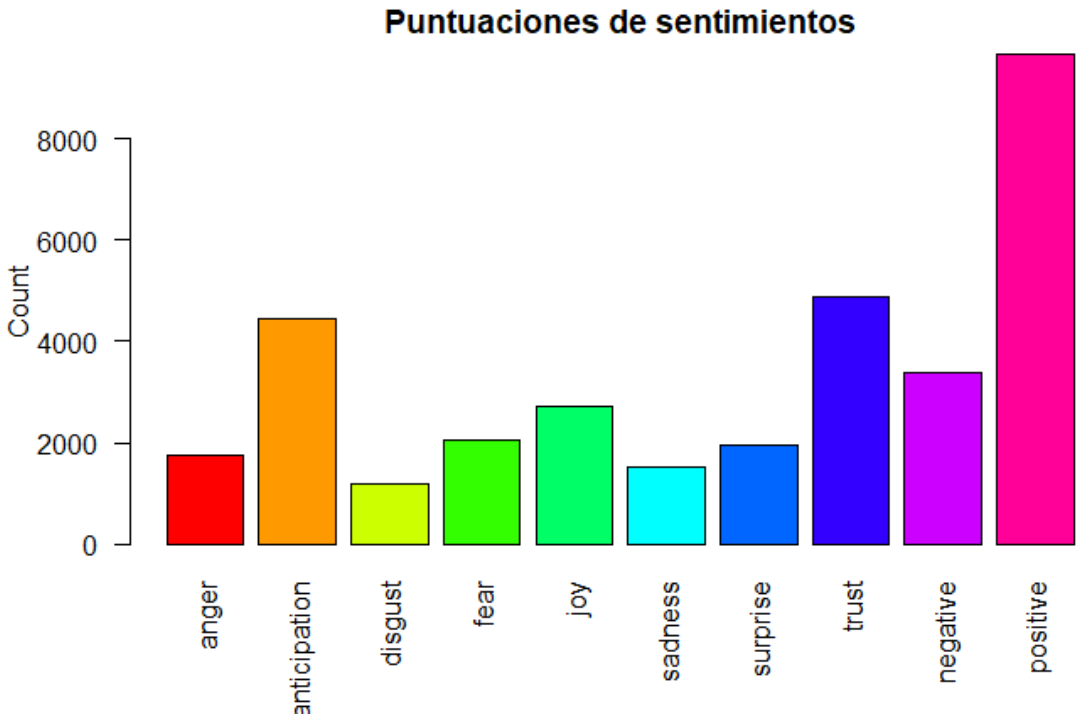

Ilustración 39 Gráfica de barras de las emociones obtenidas por el paquete Syuzhet.

En general, se puede decir que predominan las emociones positivas, mediante un conteo de la clasificación realizada con el paquete Syuzhet se muestra una tabla comparativa con el número de palabras en el corpus asociadas por cada emoción de la rueda de Plutchik ordenada de manera ascendente. Dónde parcialmente casi todas las cantidades de palabras corresponden con la relación de la gráfica de puntuaciones, a excepción de miedo (fear) y sorpresa (surprise), que no tienen la misma posición, desde la comparativa de la gráfica y la tabla. Pero en general la proporción de palabras es similar a la proporción de observaciones clasificadas en la emoción.

Tabla 6 Tabla comparativa del número de palabras asociadas a una emoción.

	Disgust	Sadness	Anger	Fear	Surprise	Joy	Anticipation	Trust
Núm. Palabras	**1121**	**1408**	**1565**	**1713**	**1734**	**2243**	**3234**	**3434**

Algo muy importante de señalar, es que generalmente una palabra suele estar asociada a más de una emoción, es por lo que, si se sumara cada uno de los totales de palabras por emoción excedería por mucho el tamaño del vocabulario del corpus, esto porque las palabras de vocabulario han ido asignadas a más de una emoción en particular.

Ahora con ayuda nuevamente de una nube de palabras, se representan las palabras más frecuentes clasificadas por syuzhet para cada emoción, en las ilustraciones 40 y 41. En la primera nube de palabras se tiene a las emociones de felicidad, tristeza, ira y miedo, y sobre ese subconjunto del corpus, es mayor la frecuencia de las palabras asociadas a la felicidad y la tristeza, por otra parte, para ira y miedo apenas se alcanza a extraer unas cuantas palabras, esto porque al construir la nube de palabras se da preferencia las palabras más frecuentes.

Ilustración 40 Nubes de palabras para las emociones: felicidad, tristeza, ira y miedo.

La segunda nube de palabras basada en el análisis de emociones se muestra en la ilustración 41, dónde se extraen las palabras más relevantes para cada una de las siguientes emociones: Expectación, repugnancia, sorpresa y confianza, emociones de las cuales se muestran mayoritariamente palabras asociadas a repugnancia y sorpresa.

Visualmente comparando ambas nubes de palabras se puede observar que hay palabras que son asociadas a más de una emoción, cómo se mencionó anteriormente, sin embargo, es parte de la lingüística y psicología de este tipo de problemas, es decir, una emoción puede ser muy subjetiva a contextos e incluso muchas veces, si la expresión es auditiva, también puede depender la emoción del tipo de sonido que se utiliza para ser expresada. Esa situación en uno de los muchos retos del PLN, en los cuales se ha trabajado arduamente durante las últimas décadas, y en efecto hoy en día hay muchas técnicas, que ayudan a la solución de ellos, cómo los modelos multimodales, donde al mismo tiempo se validan distintos tipos de datos para obtener resultados más precisos.

Ilustración 41 Nubes de palabras para las emociones: Expectación, repugnancia, sorpresa y confianza.

5 Capítulo 5

Conclusiones

El análisis de sentimientos es un problema del PLN, que como muchos otros más problemas retoman el estado del arte de otras disciplinas, lo que hace que sea una de las áreas desafiantes de la inteligencia artificial.

El principal propósito del desarrollo de este trabajo de tesis fue la exploración de distintas alternativas de machine learning para el análisis de sentimientos en textos no etiquetados. Esto porque la mayoría de los datos disponibles se encuentran no procesados, sin etiquetas o clasificación, generalmente distribuidos en grandes bases de datos, o cómo es el caso de este trabajo, información recuperada desde medios digitales, como redes sociales. Por lo que, es necesario realizar estos procesos manualmente, el etiquetamiento muchas de las veces con ayuda de expertos, siendo una tarea ardua y costosa.

Con el fin de buscar alternativas para realizar este proceso de una manera no manual se propusieron algunas soluciones basadas en machine learning para realizar el proceso de análisis de sentimientos de manera no supervisada. Sin embargo, para tener un punto de evaluación mediante métricas más precisas fue necesario hacer una clasificación mediante el etiquetamiento de una polaridad de sentimiento con el uso de una herramienta de la nube, que además de permitir hacer una comparación con el comportamiento de los algoritmos no supervisados, también abriera la posibilidad de emplear técnicas supervisadas, con el fin de evaluar el comportamiento de los datos en ambos escenarios.

Los algoritmos no supervisados explorados es este trabajo de tesis fueron K-means y Lbl2Vec, los cuales permitieron explorar dos tipos de técnicas no

supervisadas, Clustering analysis y Self-supervised learning, respectivamente. De esta exploración se obtuvieron las siguientes observaciones.

5.1 Observaciones del Clustering Analysis

Para el Clustering Analysis se realizó la evaluación de dos alternativas, K means y otro algoritmo basado en K-meas, MiniBachKmeans, para evaluar el comportamiento de los datos, aplicando clustering para la solución del problema de análisis de sentimientos.

K-meas en su implementación básica, con 3 clústeres y 100 épocas, no mostro resultados realmente interesantes con respecto a la separación de clústeres asociados a una polaridad se sentimiento, sin embargo, al explorar más los clústeres, se descubrió que hacia un tipo de asociación entre las palabras que estaban relacionadas a ciertos temas de inteligencia artificial, esto puede estar relacionado a la forma en la que se recuperaron los datos, es decir, la consulta, que consistía en extraer tweets que contuvieran ciertas palabras que están relacionadas con temas de la inteligencia artificial.

MiniBachKmeans por su parte, mostro un ligero comportamiento menor al de kmeans, para 3 clústeres, sin embargo, al hacer el entrenamiento del modelo con 5 clusteres los valores de las métricas mejoraron en algunas configuraciones, es decir, el agrupamiento de los datos mostro mejores resultados, a pesar de que la configuración optima obtenida con el método Elbow era de 3 clústeres.

Las métricas empleadas para la evaluación de k means y MiniBachKmeans fueron al igual que los algoritmos, no supervisadas, para obtener comparaciones justas de los resultados. Métricas de las cuales apenas se alcanzó un resultado por debajo de la media de los resultados óptimos, de lo que se puede concluir que el clustering no es una solución óptima para el análisis de sentimientos en concreto, puede ser una estrategia interesante para descubrir conocimiento de los datos, en la etapa de análisis y exploración de datos, e incluso en el proceso de data

mining, o para otros problemas semejantes como la categorización de documentos, en casos donde las etiquetas no han sido asignadas, es decir, a partir de los clústeres obtenidos, se puede definir que etiquetas se les asigna a los datos.

Otra alternativa interesante, es el uso de un autoencoder con k-means [4], que ha comprobado obtener buenos resultados, al evaluar los datos en diferentes dimensiones, y así entrenar un modelo con los conocimientos obtenidos en esas transformaciones en las distintas capas a través de la red con los datos. Debido a que cuando la dimensión del espacio de características de entrada es muy alta, el agrupamiento se vuelve ineficaz obteniendo métricas de similitud poco confiables, cómo lo sucedido en este experimento. La transformación de datos del espacio de características de alta dimensión a un espacio dimensional más bajo, en el que realizar el agrupamiento es una solución intuitiva, se puede lograr aplicando técnicas de reducción de dimensiones más eficientes que PCA, cómo una red neuronal profunda.

5.2 Observaciones de Self-supervised learning

Este tipo de algoritmos se caracterizan porque estar acompañados de una supervisión auxiliar no supervisada. Por lo general, supervisión que está diseñada a mano para explotar cierta información intrínsecamente disponible en los datos de entrenamiento no etiquetados. Lbl2Vec es un algoritmo de Deep learning que genera vectores de etiquetas, documentos y vectores de palabras automáticamente mediante la definición de palabras claves predefinidas manualmente. Lo interesante de este trabajo en la implementación de este algoritmo fue la construcción de un diccionario para cada clase de las polaridades de sentimientos, defiendo palabras positivas, negativas y neutras mediante el SentimentAnalysis VADER.

Después de realizar pruebas con distintas configuraciones de parámetros para Lbl2Vec, se obtuvieron buenos resultados, según el valor alcanzado con F1

Score, obteniendo un valor aproximado de 0.74, considerando que el valor máximo es 1, los resultados fueron buenos. Adicionalmente a esta métrica también se realizó una comparación de los resultados obtenidos con este modelo respecto de los resultados de Azure, con lo que sorprendentemente se descubrió que las etiquetas de sentimiento si cambian en un 35% de las observaciones del conjunto de datos, pero esto es claramente comprensible, pues para empezar Lbl2Vec hace la clasificación con 3 polaridades de sentimiento, mientras que Azure lo realiza con 4. Además, muy posiblemente el algoritmo de Azure este entrenado con un vocabulario considerablemente más grande al creado en la implementación de Lbl2Vec.

5.3 Observaciones del trabajo supervisado

Para el trabajo supervisado solo se exploró con LSTM, dado que en la literatura se encuentran trabajos interesantes con buenos resultados para análisis de sentimientos donde se hacen implementaciones con LSTM.

El uso de un modelo preentrenado para la capa de incrustaciones de palabras o embeding layer, proporcionó mayor estabilidad al modelo, esta representación permite obtener mejores relaciones semánticas de las palabras durante el entrenamiento. Al hacer este preentrenamiento se obtuvieron los vectores de incrustaciones resultantes del vocabulario del conjunto de datos, junto con las relaciones semánticas que GloVe proporciona, lo que adiciona mayor robustes al modelo.

Además, la integración del modelo en la capa oculta, con la capa bidireccional de LSTM y las dos capas densas lograron obtener un Acuracy aceptable, es cierto que, los valores óptimos podrían ser más altos, pero también es interesante destacar que este conjunto de datos es completamente nuevo y no hay otros trabajos realizados sobre el mismo, como pudiera ser para cualquier otro conjunto de datos obtenido de algún repositorio dedicado a la investigación.

Son datos reales y totalmente procesados de inicio a fin, por lo que, los resultados se consideran buenos.

Una LSTM, proporciona buenos resultados para un análisis de sentimientos basado en polaridad de sentimiento, y se puede decir que también seria efectivo para un análisis de sentimientos basado en emociones, en ese caso, el modelo se modificaría en la capa de salida, aumentando el número de neuronas de acuerdo con la relación de clases en las que se clasificarían las emociones, pero dado la integridad de este conjunto de datos hasta el momento, este experimento aun no es posible, pues sería necesario tener los datos etiquetados de acuerdo a emociones.

5.4 Observaciones del análisis de sentimientos

El proceso de análisis y exploración de datos es uno de los procesos más relevantes de cualquier trabajo de machine learning, la consistencia e integridad de datos es lo que define el funcionamiento de los modelos.

Para visualización y análisis de datos para PLN, R proporciona varios paquetes que ayudan a comprender y deducir conclusiones interesantes en un corpus. Este proceso de análisis ayudó a comprender un poco el comportamiento de los usuarios en Twitter con respecto a la inteligencia artificial, donde en esencia se ocupa para comunicar o mencionar algunos trabajos desarrollados en el área, publicidad u oferta de cursos generalmente en línea, y también deducir la popularidad de herramientas como lenguajes de programación y librerías o paquetes. Inferencias que con el análisis de sentimientos basado en polaridad o emociones no hubieran sido obtenidas, y que pudieran ser relevantes para algún caso de estudio en particular, como, por ejemplo, ¿qué tipo de cursos son de interés para la población? y ¿qué herramientas despiertan más interés para ser aprendidas? Por lo anterior, comprender y explorar los datos, es un proceso muy importante y relevante en la toma de decisiones.

5.5 Observaciones genérales

El aprendizaje no supervisado tiene muchos desafíos importantes en la investigación, sin embargo, hay muchos modelos y algoritmos desarrollados que permiten modelar los datos y obtener buenos resultados. Alternativas como el uso de Transformers, para entrenar modelos de análisis de sentimientos que pudieran ser más robustos, sin necesidad de tener datos etiquetados, al ser modelos preentrenados, de lo que también se podrían obtener deducciones importantes, y sería una buena implementación a futuro, que enriquezca este trabajo.

Por su parte el aprendizaje supervisado, provee resultados más precisos, además de que hay mucha información e investigación disponible sobre la cual trabajar. A diferencia del aprendizaje no supervisado para tareas de PLN. Por lo que, las mejores estrategias para tareas de PLN en general hasta el momento se derivan de técnicas supervisadas o semi- supervisadas.

En este pequeño trabajo de tesis se realizó una exploración mínima de las alternativas existentes, logrando resultados satisfactorios, pero principalmente se obtuvo un conocimiento muy relevante, y al mismo tiempo se descubrió un interés especial por el área, por lo que seguramente se seguirá trabajando en este campo. Pues, como ya se dijo hay muchos retos del PLN, en los que pequeñas aportaciones pueden ser muy significativas.

6 Bibliografía

[1] Braun, D. & Matthes, F. (2021). Lbl2Vec: An Embedding-based Approach for Unsupervised Document Retrieval on Predefined Topics. Proceedings of the 17th International Conference on Web Information Systems and Technologies. https://doi.org/10.5220/0010710300003058

[2] David Gallardo-Pujol i Antonio Andrés Pueyo (ed.). «5-Estabilitat i desenvolupament de la Intelligencia i el QI». Psicología de les Diferències Individuals (en catalán)

[3] Embedding projector - visualization of high-dimensional data. (s. f.). tensorflow.org. Recuperado 21 de agosto de 2022, de https://projector.tensorflow.org/

[4] Gao, L., Liu, X. & Yin, J. (2017). Improved Deep Embedded Clustering with Local Structure Preservation. Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. https://doi.org/10.24963/ijcai.2017/243

[5] García, J. (s. f.). Comparativa de técnicas de balanceo de datos. Aplicación a un caso real para la predicción de fuga de clientes. [Tesis de maestría]. Universidad de Oviedo.

[6] Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

[7] Jockers, M. (2020, 24 noviembre). Introduction to the Syuzhet Package. Recuperado 14 de octubre de 2022, de https://cran.r-

project.org/web/packages/syuzhet/vignettes/syuzhet-vignette.html

[8] Kristombu Baduge, S., Thilakarathna, S., Perera, J., Arashpour, M., Sharafi, P., Teodosio, B., Shringi, A. & Mendis, P. (2021). Artificial intelligence and smart vision for building and construction 4.0: Machine and deep learning methods and applications. sciencedirect, 141(104440). https://doi.org/10.1016/j.autcon

[9] Kwartler, T. (2017, 24 julio). Text Mining in Practice with R (1.a ed.). Wiley.

[10] Machine learning, una pieza clave en la transformación de los modelos de negocio. (2018). Management Solutions. Recuperado 15 de mayo de 2022, de https://www.managementsolutions.com/sites/default/files/publicaciones/esp/machine-learning.pdf

[11] Moolayil, J. (2018). Learn Keras for Deep Neural Networks: A Fast-Track Approach to Modern Deep Learning with Python (English Edition) (1st ed.). Apress.

[12] Prasad, R. (2017). Machine-Learning-Concepts. Machine-Learning-Concepts. Recuperado 2 de abril de 2022, de https://github.com/free-to-learn/Machine-Learning-Concepts

[13] Silge, J. & Robinson, D. (2017). Text Mining with R: A Tidy Approach. O'Reilly.

[14] Sklearn metrics homogeneity_score. (2007). scikit-learn. Recuperado 18 de junio de 2022, de https://scikit-learn.org/stable/modules/generated/sklearn.metrics.homogeneity_score.html

[15] Sklearn metrics completeness_score. (s. f.). scikit-learn. Recuperado 18 de junio de 2022, de https://scikit-learn.org/stable/modules/generated/sklearn.metrics.completeness_score.html

[16] sklearn.metrics.normalized_mutual_info_score. (s. f.). scikit-learn. Recuperado 21 de octubre de 2022, de https://scikit-learn.org/stable/modules/generated/sklearn.metrics.normalized_mutual_info_score.html

[17] Tweepy Documentation. (s. f.). Tweepy Documentation. Recuperado 10 de enero de 2022, de https://docs.tweepy.org/en/stable/

[18] Understanding LSTM Networks -- colah's blog. (s. f.). colah's blog. Recuperado 20 de febrero de 2022, de http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[19] Wijffels, J. (2020, 12 octubre). Textrank for summarizing text. Recuperado 14 de octubre de 2022, de https://cran.r-project.org/web/packages/textrank/vignettes/textrank.html

[20] Wijffels, J. (2022, 24 marzo). CRAN - Package udpipe. CRAN. Recuperado 15 de octubre de 2022, de https://cran.r-project.org/web/packages/udpipe/index.html