Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Análisis Semántico de Opiniones en Twitter sobre la Vacunación del COVID-19

Cinthia Tavarez Mora

Dirigido por:
Agustina Bouchet e Irene Mariñas del Collado

UNIVERSIDAD DE OVIEDO
Facultad de Ciencias
Máster Universitario en Modelización e Investigación Matemática, Estadística y Computación

Octubre de 2023

Resumen

Utilizando como marco de datos los recopilados por el grupo *Panacea Lab* de la Universidad Estatal de Georgia, se seleccionaron y extrajeron un total de 11,787 *tweets* publicados durante el periodo comprendido entre el 1 de marzo 2020 al 15 de abril 2020 en la plataforma de redes sociales *Twitter*. El objetivo es identificar las opiniones de los usuarios sobre el proceso de vacunación contra el COVID-19. Como metodología, este estudio sigue una combinación de técnicas y modelos de *Procesamiento de Lenguaje Natural* y aprendizaje automático supervisado y no supervisado. Se aplica un modelo *Word2Vec* con las arquitecturas de *Continuous Bag of Words* y *Muestreo Negativo* para representar las palabras en vectores de valores reales. Además, se emplea la *Frecuencia de Términos-Frecuencia Inversa de Documentos* y *Análisis de Componentes Principales* para seleccionar un ranking de palabras clave y reducir la dimensionalidad de los datos. A continuación se aplica un análisis de clúster con el algoritmo *k-medias*. Los resultados permitieron identificar 8 etiquetas sobre las cuales giraron las opiniones de los usuarios en este periodo, 4 de ellas pertenecientes al modelo de creencias en el ámbito de la salud. Finalmente, se usaron modelos de clasificación, como *Random Forest* y *Máquinas de Soporte Vectorial*, para evaluar la capacidad de clasificación mediante distintas métricas.

Palabras claves: Vacunas, COVID-19, Procesamiento de Lenguaje Natural, redes sociales, *Twitter*, *Word2Vec*, clúster, opiniones.

Índice general

Índice de acrónimos

API	Application Programming Interfaces. 12–14, 18, 19, 44
App	Applicación. 19, 20
AUC	Area Under of Curve. 81, 84, 88, 89, 94–96
CBOW	Continuous Bag of Words. 34, 36–39, 42, 43, 60, 99
CH	Índice de Calinski Harabasz. 58, 65
COVID-19	Coronavirus 2019. 1, 7, 10–16, 18–21, 27, 46, 48, 65, 97–99, 101
DB	Índice de Davies Bouldin. 58, 65
HBM	Health Belief Medical. 8, 13, 14, 64, 75, 76, 97, 98, 100
ID	Índice de Dunn. 59
Id	Identificadores. 19, 20
IDF	Inverse of Documents Frequency. 25
IH	Índice de Hartigan. 58, 65
NCE	Noise Contraction Estimation. 40, 41
NLP	Natural Language Processing. 9, 10, 15, 16, 22, 23, 28, 31, 99
NNLM	Neural Net Language Model. 31, 32, 34–36
OMS	Organizacion Mundial de la Salud. 7
PCA	Principal Component Analysis. 16, 50, 51, 59, 60, 64, 76, 99, 100

POS	Part of Speech. 24
RF	Random Forest. 13, 69–71, 79, 80, 83, 84, 88, 89, 92–94, 96, 100
RN	Redes Neuronales. 31–34, 36, 37, 39, 101
ROC	Receiver Operating Characteristic Curve. 81, 84, 89–91
RRSS	Redes Sociales. 9–11, 13–15, 18, 20, 21, 98–100, 102
SIR	Suceptibles, Infectados y Recuperados. 13, 15
SSE	Sum of Square Error. 57, 65, 66
SVM	Support Vector Machine. 13, 17, 72–74, 87, 92–96, 100, 101
TF	Terms of Frequency. 25
TF-IDF	Term Frequency-Inverse Document Frequency. 25–27, 60, 97–100
TP	Top de Palabras. 62, 63

Capítulo 1

Introducción

A partir del mes de diciembre del año 2019, el brote surgido por el virus respiratorio SARS-CoV-2 que es el causante de la enfermedad del Coronavirus 2019 (COVID-19)[1] se convierte en una de las emergencias de salud más relevante en la última década lo que llevo a declararla como pandemia por la Organizacion Mundial de la Salud (OMS) en marzo del año 2020.

Para finales del año 2020 se inicia una carrera científica por encontrar la vacuna contra el virus, del que a la fecha del 13 de agosto 2023 se ha reportado 769 millones de casos confirmados y más de 6,9 millones de muertes en todo el mundo (World Health Organization, 2023). Como parte de la carrera por reducir el número de casos y muertes, se inició el proceso de desarrollo y estudio de vacunas contra el virus. Finalmente, para febrero del año 2021, se habían autorizado al menos 10 vacunas para uso en humanos (algunas con autorización solo para circulación nacional, como el caso de Cuba, que desarrolló su propia vacuna). Esto marca el inicio de un nuevo punto de inflexión en el debate sobre la pandemia, sus efectos y las diferentes posturas. Es decir, el éxito o fracaso del proceso de vacunación no dependía únicamente de la efectividad de las vacunas (se discutió ampliamente el hecho de que las vacunas no previenen completamente los contagios por el

[1]Nombrado así en la intervención del Director General de la Organizacion Mundial de la Salud en la conferencia de prensa el 11 de febrero de 2020.

virus, pero sí reducen su gravedad y los síntomas), sino también de la disposición de los ciudadanos a participar en los distintos planes de vacunación elaborados por los países.

Lo anterior aborda uno de los temas relativamente más estudiados alrededor en el ámbito de salud humana y especialmente sobre las vacunas: *«la hesitación hacia las vacunas»*, o mejor dicho, las creencias, percepciones y decisiones que influyen en sí los individuos optan o no por la vacunación como una medida de salud. Es en esta área que se plantea lo que se conoce como el **Modelo de Creencias de Salud** (*Health Belief Medical*, HBM). El modelo HBM surgió en el año 1958, después de que las campañas de vacunación contra la polio en Estados Unidos no estaban obteniendo los resultados esperados. Fue entonces cuando un grupo de psicólogos sociales del Servicio de Salud Pública, conformado por Irwin Rosenstock, Mayhew Derryberry y Bárbara Carriger, presentó un modelo basado en 4 constructos: la **Susceptibilidad percibida**, la **Gravedad percibida**, los **Beneficios percibidos**, las **Barreras percibidas** (Rodriguez-Insuasti et al., 2020). Desde entonces se ha convertido en uno de los marcos teóricos más utilizados en el área del análisis de la conducta que explica el comportamiento de los individuos en relación con la salud.

Los componentes del modelo HBM se derivan de la hipótesis de que la conducta de los seres humanos gira en torno a dos aspectos ante la toma de decisiones: 1) el valor de la meta pautada y 2) la probabilidad de lograr esta meta. Si esos aspectos se traducen al ámbito de la salud podría definirse con los términos *evitación* y *prevención*, el primero está relacionado con las acciones del individuo para evitar la enfermedad o la muerte si ya la padece, mientras el segundo está relacionado con las conductas que previenen la enfermedad o mejoran la salud ante esta.

La *Susceptibilidad percibida* es una dimensión importante que valora cómo los sujetos varían en la percepción de la propia vulnerabilidad a enfermar, desde el sujeto que niega cualquier posibilidad de contraer una enfermedad, pasando por el que admite la posibilidad "estadística" de que le ocurra un problema de salud, pero que no considera la posibilidad real de que le ocurra, hasta el sujeto que expresa su convencimiento de estar en peligro cierto de contraer una enfermedad. Así pues, esta se refiere fundamentalmente a la percepción subjetiva que tiene cada ser humano sobre el riesgo de caer enfermo. La *Gravedad percibida* se refiere a las creencias sobre la gravedad o no de contraer una determinada enfermedad o dejarla sin tratar una vez contraída y se aborda desde dos tipos de consecuencias relacionadas con la pérdida de la salud, por un lado, las consecuencias médico-clínicas (como muerte, incapacidad o dolor), y, por otro lado, las posibles consecuencias sociales (tales como la merma en las relaciones sociales, los efectos de la enfermedad sobre la capacidad laboral del sujeto o sobre sus relaciones familiares, etc.)(Moreno San Pedro and Gil Roales-Nieto, 2003).

Las creencias del sujeto respecto a la efectividad relativa que las diferentes conductas que este puede asumir puedan tener a la hora de enfrentarse con la enfermedad, es lo que se ha considerado como la dimensión de los *Beneficios percibidos*. Así, por ejemplo, aunque una persona esté asustada y se sienta amenazada por un trastorno de salud concreto, no seguirá las recomendaciones de su médico, al menos que las perciba como eficaces para enfrentar ese trastorno, asumiendo que las creencias del sujeto respecto a los cursos de acción de que dispone están influidas por las normas e incluso presiones del grupo social al que pertenece. Por último, otra dimensión del modelo tiene que ver con que determinadas *Barreras percibidas* se opongan a la ejecución de la conducta en cuestión, como pueden ser, por ejemplo, determinados aspectos potencialmente negativos de un curso de acción concreto. Así, un individuo puede considerar un determinado curso de acción como realmente efectivo para enfrentarse a un trastorno de salud, pero, al mismo tiempo, puede verlo como costoso, desagradable o doloroso. Estos aspectos negativos de la conducta de salud funcionarían como barreras para la acción que interaccionan con las anteriores dimensiones (Moreno San Pedro and Gil Roales-Nieto, 2003).

El acceso a estas creencias y/o percepciones se hace complejo a través de las prácticas tradicionales que conllevan el levantamiento de encuestas o sondeos, estudios de grupo o estudios longitudinales que suelen requerir una serie de recursos no solo económicos sino de tiempo para su obtención y procesamiento hasta obtener resultados. Ante esto surge como herramienta de alto potencial el uso de los discursos y temas relevantes de discusión en las Redes Sociales (RRSS), a través de la explotación del volumen de datos disponible en estas plataformas, con la aplicación de una metodología desde las diversas técnicas disponibles en el ámbito del Procesamiento del Lenguaje Natural (*Natural Language Processing*, NLP).

A medida que la interacción en línea se ha convertido en una parte integral de la vida cotidiana, las redes sociales (RRSS) generan enormes cantidades de datos en forma de publicaciones, comentarios y opiniones de usuarios. La aplicación de técnicas avanzadas de NLP permite analizar y comprender estos datos de manera eficiente, capturando las voces individuales y las tendencias emergentes. A diferencia de los métodos tradicionales de obtención de opiniones, como las encuestas, que consumen tiempo y recursos significativos para recopilar y procesar la información, el NLP automatiza este proceso al analizar grandes volúmenes de texto de manera rápida y precisa. Esto no solo agiliza la obtención de información, sino que también permite capturar sentimientos y opiniones en tiempo real, lo que resulta invaluable para la toma de decisiones informadas en entornos dinámicos.

La capacidad del NLP para extraer información de las RRSS se basa en su capacidad para interpretar y analizar patrones semánticos sutiles. A través del procesamiento de

grandes conjuntos de datos textuales, el NLP puede identificar temas recurrentes, detectar cambios en la opinión pública y evaluar el sentimiento general en torno a un tema específico. Este enfoque es especialmente eficiente, los modelos NLP permiten a las organizaciones acceder a información en tiempo real sobre la percepción y la actitud de los usuarios, lo que facilita la adaptación ágil de estrategias y decisiones. En última instancia, el uso de NLP para extraer información de las RRSS se erige como una alternativa poderosa y eficiente a los métodos tradicionales, permitiendo una comprensión más profunda y oportuna de las opiniones y tendencias que moldean el mundo digital.

1.1. Revisión literaria

Las plataformas de RRSS permiten a millones de personas de todo el mundo compartir opiniones de forma constante sobre cualquier tópico personal o de debate público. Esto último constituye uno de los aspectos de mayor potencial dado las limitaciones que ofrecen las técnicas tradicionales de poder capturar en tiempo real y de manera instantánea las opiniones de las personas.

Esto abre un campo de estudio para comprender las decisiones de las personas en diversos aspectos de la vida, como la salud, a través del uso de *big data* y la explotación de datos provenientes de estas aplicaciones de RRSS. Durante y después de la pandemia del COVID-19, se llevaron a cabo investigaciones para identificar la ecología en torno a la decisión de vacunarse utilizando técnicas estadísticas y modelos matemáticos.

La identificación de las posturas que influyen en la decisión de vacunarse conlleva el beneficio de poder planificar estrategias efectivas para abordar los temores y las dudas que la población pueda tener en relación con las vacunas. Estas estrategias también pueden proporcionar evidencia para abordar la resistencia a la vacunación en otros contextos y explicar por qué no se ha logrado la erradicación total de los grupos antivacunas.

El estudio *«Using big data to understand the online ecology of COVID-19 vaccination hesitancy»* Teng and Khong (2022) aborda las duda sobre la vacuna contra el COVID-19. Consiste en un conjunto de datos de 43,203 comentarios de *YouTube* específicamente de los comentarios de los usuarios en videos colocados por medios corporativos que estuvieran relacionados con «la eficacia de las vacunas» delimitados previamente por los autores. Aplican modelos de aprendizaje no supervisado para la creación de clústeres de términos provenientes de los textos previamente pre-procesados. Los autores usan como marco el

modelo de actualización de creencias de salud y sus 4 constructos (susceptibilidad, severidad, barreras y beneficios) y otros aspectos contextuales como la confianza en la ciencia, confianza en gobierno, confianza en las RRSS, ideología política, la información errónea difundida, confianza en farmacéuticas, entre otros. Adicionalmente, para investigar las relaciones casuales entre las variables resultantes (a partir de las etiquetas de los clústeres) y la intención de vacunación, se utilizan modelos de regresión múltiple. Los autores demuestran que entre las razones de la reticencia a la vacunación figuran las preocupaciones sobre la seguridad de las vacunas, los posibles efectos secundarios, la falta de confianza en el Gobierno y las empresas farmacéuticas. Además, la difusión de información errónea de los movimientos contra las vacunas alimentaron la vacilación de la vacunación y socavaron la confianza pública en las vacunas contra la COVID-19.

En esta revisión, exploramos uno de los estudios más citados[2] en el ámbito. Este estudio, realizado por Loomba et al. (2021), emplea datos de panel recopilados a través de una encuesta en línea que involucra a un grupo de 8,001 individuos. Estos participantes se dividen en dos grupos: un grupo de control y un grupo de tratamiento. Los individuos del primer grupo fueron expuestos a información con validez previamente verificada, mientras el segundo grupo fue expuesto a imágenes de información errónea sobre COVID-19 previamente seleccionada por los investigadores a partir de RRSS basadas en criterios tales como actualidad y alcance de la cuenta que lo comparte. Como modelo de estimación de los efectos del tratamiento utilizan una regresión logística ordenada bayesiana jerárquica. A todos los encuestados de ambos grupos se les pidió que proporcionaran su intención de recibir una vacuna COVID-19 antes y después de estar expuestos a la información recolectada, estableciendo la *intención de vacunación* antes y después de la aplicación del tratamiento como una variable ordenada en un rango de *Likert*[3] de 1 a 4. Este rango constituiría la variable dependiente para principalmente estimar: 1) el tratamiento de la información errónea en el cambio en la intención de vacunación en relación con la información fáctica, y 2) cómo estos tratamientos afectan de manera diferente a los individuos por sus características socio-demográficas. Entre sus principales hallazgos se encuentran que la exposición a la información errónea indujo una disminución en la intención de 6.2 puntos porcentuales en el Reino Unido y 6.4 puntos porcentuales en los Estados Unidos

[2]Hasta la fecha de redacción de este documento, el estudio ha acumulado 713 citas en la plataforma de la revista Nature: https://www.nature.com/articles/s41562-021-01056-1.

[3]Creada por Rensis Likert en 1932. La escala tipo *Likert* es un instrumento de medición o recolección de datos cuantitativos utilizado dentro de la investigación. Es un tipo de escala aditiva que corresponde a un nivel de medición ordinal (Luna, 2012).

entre aquellos que declararon que definitivamente aceptarían una vacuna.

Otro estudio de naturaleza similar es el llevado a cabo por Carrieri et al. (2023). En esta investigación, los autores emplearon datos obtenidos de la *Encuesta en línea sobre vida, trabajo y COVID-19*, realizada durante la semana del 12 al 1 de abril de 2021. Esta encuesta es gestionada por EUROFOUND[4] y consiste en un estudio transversal repetido a lo largo de varias semanas, con la participación de aproximadamente 35,000 personas residentes en la Unión Europea (UE). El muestreo empleado en esta encuesta es de tipo bola de nieve[5]. La novedad del estudio constituye el establecimiento de un modelo que permite medir el impacto del veto e investigación que se realizara en Europa y otros países a la vacuna AstraZeneca debido a los casos de trombosis reportados en personas luego de ser inoculados, incluyendo una variable que recoge el efecto del periodo en la que se divulgaron los casos y posterior veto con la decisión de vacunarse planteada como pregunta en el cuestionario.

Otros estudios que también explotan las informaciones a partir de la Interfaz de Programación de Aplicaciones (*Application Programming Interfaces*, API)[6] de *Twitter* son Pierri et al. (2022) y Wang et al. (2021). En el primero, los autores se centran en identificar y medir hasta qué punto las tasas de vacunación contra COVID-19 y la reticencia a la vacuna están asociadas con niveles de información errónea en línea sobre las vacunas. También se busca extraer la evidencia que sustente la direccionalidad desde la desinformación en línea hasta la vacilación de la vacuna. Usando una base de datos previamente

[4]EUROFOUND es una agencia tripartita de la Unión Europea que proporciona conocimientos para el desarrollo de políticas sociales, de empleo y relacionadas con el trabajo. Para acceder a la encuesta, puedes visitar el siguiente enlace: `https://www.eurofound.europa.eu/`.

[5]En el muestreo de bola de nieve, se inicia con un pequeño grupo de participantes que tienen las características que se desean estudiar. Luego, se les pregunta a estos participantes si conocen a otras personas que también cumplen con esas características y estarían dispuestas a participar en la investigación. Estas nuevas personas se convierten en parte de la muestra y, a su vez, se les pregunta si conocen a otras personas que cumplan con los criterios, y así sucesivamente. Este proceso continúa hasta que se alcance un tamaño de muestra adecuado o se haya agotado la red de contactos.

[6]Es un conjunto de reglas y protocolos que permiten que diferentes aplicaciones o componentes de software se comuniquen entre sí y compartan datos o funcionalidades. Las API permiten que los desarrolladores de software accedan a ciertas características o datos de una aplicación o servicio sin necesidad de conocer los detalles internos de cómo funciona esa aplicación.

elaborada por el Observatorio de *Social Media* de la Universidad de Indiana en EE.UU compuesta de más de 50 millones de *tweets* recopilados durante los días comprendidos entre el 4 de enero al 25 de marzo mediante la API de la plataforma con palabras claves relacionadas con las vacunas contra el COVID-19. La relevancia de este estudio reside en la incorporación de la identificación geográfica de los *tweets* recolectados, lo que hizo posible fusionar los datos recopilados mediante las RRSS con datos socioeconómicos relacionados con cada uno de los estados del país, pudiendo modelar la relación de estas posturas con ciertas características como la edad, sexo y otras, así como también medir los efectos entre los grupos.

En Wang et al. (2021), si bien no se aborda el tema sobre la vacunación COVID-19, si explotan el uso de un *dataset* sobre COVID-19 previamente capturado en este caso por el grupo *Panacea Lab*[7] con el objetivo de investigar las creencias de salud relacionadas al virus así como los posibles factores de impacto asociados con las fluctuaciones en las creencias de salud en las RRSS. Esto se lleva a cabo usando modelos de actualización de creencias sobre Salud (HBM) en sus 4 constructos a medir. Como caso especial de estudio, los autores se enfocaron en la difusión de información sobre el uso de hidrocloroquina (HCQ) y cloroquina (CQ) para el tratamiento de los síntomas asociados al COVID-19. Los *tweets* recopilados por los autores fueron etiquetados como positivos o negativos por estar relacionados con uno de los constructos del HBM, lo que significa que podrían asignarse al menos a uno de los cuatro. Por lo tanto, cada *tweet* podría tener hasta cinco etiquetas. Las reglas que determinarían la asociación de un *tweet* a uno de los 4 constructos se establecen mediante el entrenamiento previo de los datos con un subconjunto del universo total de *tweets* que fueron etiquetados manualmente, por una parte, de los colaboradores del estudio y debatidos por los autores para llegar a un consenso sobre su identificación. Para evaluar los *tweets* recopilados y etiquetados según los 4 constructos, utilizan y comparan varios modelos de aprendizaje automático supervisado y no supervisado (bosques aleatorios (*Random Forest*,RF), k-medias, Máquinas de Vector Soporte (*Support Vector Machine*, SVM) y otros) midiendo su rendimiento a través del indicador Kappa. Como aspecto adicional, los autores evalúan si la difusión de información constituía una infodemia, usando un indicador básico en los modelos epidemiológicos como es el número básico de reproducción. Para esto emplean el modelo clásico SIR (iniciales que provienen de la forma en que clasifican los 3 grupos: Susceptibles, Infectados y Recuperados), considerando a los usuarios que tuitearon sobre COVID-19 como la población susceptible. Entre esta población, estar infectado significaba que un usuario tuiteaba sobre creencias de salud

[7]Enlace al grupo: https://github.com/thepanacealab/covid19_twitter.

definidas en el alcance del modelo HBM establecido, y la recuperación indicaba que un usuario había dejado de tuitear sobre creencias de salud. Por lo tanto, el contacto con personas infectadas podría considerarse como la lectura de *tweets* relacionados con creencias de salud publicadas por otros usuarios. Algunos hallazgos relevantes de este estudio incluyen que tanto los eventos científicos como las publicaciones científicas, así como los eventos no científicos y los discursos de los políticos, demostraron ser comparables en su capacidad para influir en las tendencias de creencias de salud en las RRSS a través de una prueba de *Kruskal-Wallis*[8]. No se observaron diferencias significativas entre la influencia de los eventos científicos y no científicos tanto para los beneficios percibidos como para las barreras.

Otros estudios se concentran en la creación de una base de datos en torno al COVID-19 a partir de RRSS, como en Muric et al. (2021). Este estudio recopila las informaciones de posturas, así como informaciones sesgadas sobre la vacuna COVID-19 a partir de *Twitter*. Los autores utilizan un muestreo bola de nieve siguiendo a Pierri et al. (2022) el cual consiste en utilizar palabras claves que estén ligadas fuertemente a la negación del uso o beneficio de la vacuna (ej. #vaccinekills). Luego utilizaron estas palabras como semilla para ampliarlo a otras que usualmente eran utilizadas junto a estas e incorporándolas como nuevas al conjunto inicial de palabras semillas. El proceso se realizó varias veces hasta alcanzar el máximo de palabras claves de la API de *Twitter*. El aspecto más distintivo de este estudio es la clasificación de estas informaciones según las inclinaciones políticas de las cuentas. Para cada cuenta en el conjunto de datos, mantuvieron un registro de todos los *retweets* y los *tweets* originales que contenían un nombre de dominio afiliado a los medios de comunicación seleccionados. Cada uno de estos medios y sus cuentas en la plataforma de RRSS asociadas fueron colocados en un espectro político (izquierda, izquierda inclinada, centro, derecha inclinada, derecha) según las calificaciones proporcionadas por el servicio no partidista *AllSides*[9]. El sesgo político de cada cuenta se calculó como el sesgo político

[8]La prueba de *Kruskal-Wallis* es una prueba no paramétrica utilizada para determinar si hay diferencias estadísticamente significativas entre dos o más grupos independientes en una variable dependiente ordinal o continua. Es una alternativa a la prueba de análisis de varianza (ANOVA) cuando los supuestos necesarios para el ANOVA no se cumplen. La hipótesis nula (H_0) establece que no hay diferencias significativas entre los grupos en la variable dependiente. En otras palabras, sostiene que las muestras de los grupos independientes provienen de la misma población subyacente y que cualquier diferencia observada es atribuible al azar.

[9]*AllSides* es una compañía estadounidense que evalúa el sesgo político de medios de comunicación prominentes y presenta diferentes versiones de noticias similares de fuentes de derecha, izquierda y centro

promedio de todos los medios de comunicación de los que compartió contenido.

Por último, en Cinelli et al. (2020) los autores abordan un análisis masivo usando datos de RRSS de diferentes plataformas: *Twitter*, *Instagram*, *YouTube*, *Reddit* y *Gab* analizando el *engagement* y el interés por el tema COVID-19 y proporcionando una valoración diferencial sobre la evolución del discurso a escala global para cada plataforma y sus usuarios. Para unificar los datos de cada plataforma realizaron una selección de las publicaciones que contenían al menos un enlace web que enlaza a un sitio web fuera de la plataforma de RRSS relacionada. Posteriormente, se procedió a clasificar los medios en dos categorías: confiables y no confiables, utilizando la clasificación proporcionada por *MediaBias/FactCheek*. Esta clasificación se basa en el sesgo y se divide en cuatro categorías diferentes, una de las cuales es «Factual / Sourcing». En esta categoría, cada medio de comunicación recibe una etiqueta que refleja su nivel de confiabilidad, expresada en tres categorías: «Conspiración-Pseudociencia», «Pro-Ciencia» o «Cuestionable». Para evaluar los temas en torno a los cuales se concentra la percepción del debate, COVID-19, ejecutan un algoritmo *Partitioning Around Medoids* para agrupar las palabras de referencia en cada tema y extraer solo texto de la data procesada. Luego de obtener una base de datos acotada al objetivo del estudio, utilizan modelos para datos epidemiológicos: 1) un modelo fenomenológico que enfatizan la reproducibilidad de datos empíricos sin conocimientos sobre los mecanismos de crecimiento, y 2) un modelo SIR que intentan incorporar tales mecanismos. Con cada modelo, los autores estiman un número de reproducción básico para cada plataforma encontrando valores bastantes críticos, incluso considerando intervalos de confianza, lo que indica la posibilidad de una infodemia.

1.2. Objetivos

Como parte de este estudio se establecieron los siguientes objetivos:

1. Analizar los datos de *Twitter* relacionados con la vacunación del COVID-19 utilizando técnicas de NLP y modelado de temas, con el objetivo de identificar patrones

políticos. Se centran en la clasificación de publicaciones en línea en una escala de cinco puntos: izquierda, izquierda, centro, inclinación derecha y derecha.

y tendencias en las opiniones de los usuarios sobre la vacunación.

2. Investigar y clasificar los mensajes en *Twitter* para identificar aquellos que expresan hesitación hacia la vacunación, con el fin de comprender las preocupaciones y motivaciones detrás de esta actitud.

3. Demostrar como el uso de técnicas de *Machine Learning* y NLP pueden ser buenas herramientas cuando no se cuenta con el tiempo y recursos económicos para capturar las respuestas de los individuos a ciertos eventos y/o fenómenos mediante técnicas tradicionales.

1.3. Estructura del trabajo

En este trabajo, se presenta una aplicación de técnicas de *Word Embedding* y modelos de clúster para capturar y evaluar las discusiones predominantes sobre la vacunación del COVID-19 en los países de Iberoamérica[10] a través de la plataforma de *Twitter*. Derivado de todo lo anterior, este trabajo se estructura como sigue:

En el capítulo 2, se aborda la recopilación de datos, que es un componente esencial del proceso de investigación. Además, se examina el preprocesamiento de los datos, incluyendo la separación de cada palabra en un texto, el tratamiento de palabras vacías o *Stop words*, la lematización y la aplicación de la técnica de la frecuencia inversa de términos en un texto para la representación de documentos.

En el capítulo 3 se expone el contexto teórico detrás la representación vectorial de palabras y el concepto de *Word Embedding*. Se exploran los fundamentos de estas técnicas y su importancia en el análisis de datos textuales. También se presentan ejemplos prácticos para comprender cómo estas representaciones pueden mejorar la comprensión y el procesamiento de texto.

El capítulo 4 se centra en dos técnicas esenciales en el análisis de datos: el Análisis de Componentes Principales (*Principal Component Analysis*, PCA) y el Análisis de Clúster.

[10]Conformado por los países de América Latina y el Caribe más España y Portugal.

Se explica en qué consisten estos métodos y cómo se aplican. Se presentan los resultados del análisis de clúster, destacando patrones y agrupaciones identificadas.

El capítulo 5 se dedica al proceso de clasificación de datos. Se muestran los resultados obtenidos a través de dos modelos diferentes: bosques aleatorios y SVM. Se evalúan y comparan estos resultados para analizar la efectividad de los modelos en la clasificación de los datos recopilados.

En el capítulo 6, se presenta una discusión de los hallazgos y resultados obtenidos en relación con los resultados de otros estudios revisados. Asimismo, se resumen las conclusiones generales del trabajo. Se destacan las principales contribuciones de la investigación y se proporciona una visión general de los objetivos alcanzados. Además, se plantean posibles direcciones para investigaciones futuras.

Capítulo 2

Recopilación y preprocesamiento de datos

En este capítulo, se explorarán las técnicas esenciales para la recopilación y preprocesamiento de datos. Se abordarán los métodos de adquisición de información y se destacarán las estrategias de limpieza y transformación que permitirán que los datos sean aptos para su análisis posterior. Este capítulo proporcionará una comprensión detallada de cómo obtener, organizar y depurar los datos, preparándolos para desvelar información valiosa en investigaciones posteriores.

2.1. Recopilación de datos

Para este estudio se han aprovechado las disponibilidades de la aplicación de *Twitter* a través de su API y la data recolectada por el grupo *Panacea Lab* sobre *tweets* relacionados con el virus del COVID-19 desde el periodo de marzo 2020 hasta abril 2022.

Como ocurre en muchas RRSS, la propia plataforma pone a disposición de los usuarios una API que permite extraer información. Aunque en la mayoría de casos se trata de *web*

services API[1], con frecuencia existen librerías que permiten interactuar con la API desde diversos lenguajes de programación. Un ejemplo de ello es *Tweepy*, un *paquete* de *Python* que se comunica con la API de *Twitter*.

La App de *Twitter* ofrece la posibilidad de extraer información a través de un usuario de desarrollador. Esto implica una conexión con su API, aunque en la actualidad ya existen librerías que permiten interactuar con esta desde algunos de los software más utilizados y populares, como *Rstudio*, *Python*, *Java*, *SAS*, entre otros. Sin embargo, es importante tener en cuenta que esta extracción de datos tiene ciertas limitaciones en cuanto al acceso y al número específico de consultas. Esto puede incluir la delimitación del alcance temporal de las consultas. Adicionalmente, la App contiene acuerdos de confidencialidad que prohíben la difusión de los datos recopilados como forma de respeto de la privacidad de sus usuarios.

No obstante, como parte del apoyo a las actividades de investigación académica, la App suele ofrecer privilegios de usuario que permiten la extracción de datos históricos. A pesar de su potencial, esta tarea requiere cierto nivel de recursos computacionales y tiempo. Para subsanar estas limitaciones, se optó por acceder a los datos de la aplicación utilizando como marco de referencia los Identificadores (Id) de los *tweets* recopilados por *Panacea Lab*[2]. Estos Id proporcionan acceso a todo el flujo de opiniones, reportes y discursos relacionados con la enfermedad del COVID-19, incluido el proceso de vacunación. La extracción de estos *tweets* se realizó mediante el uso de la librería *Tweeppy* en *Python*, y con esta metodología se obtuvieron los $11,787$ *tweets* que conforman la base de datos de este estudio.

En ese sentido, la recolección consiste en enlazar la base de datos de los Id disponibles con la aplicación generando consultas[3] en la API y extrayendo información y campos relevantes. Este proceso es lo que se conoce como *hidratación*.

---

[1]Es un conjunto de protocolos y estándares que permiten que diferentes aplicaciones o sistemas se comuniquen entre sí a través de la red *WWW*.

[2]Disponibles en el siguiente enlace: `http://www.panacealab.org/covid19/`.

[3]Las limitaciones de consultas a base de datos establecen un mínimo de tiempo de espera después de cada consulta, lo que hace que se deba repetir el proceso varias veces antes de alcanzar el número de registros requeridos.

Dado que los datos recopilados por el grupo se almacenan como archivos diarios y contiene *tweets* de todos los idiomas y subtemas relacionados con el virus, se implementaron filtros que acotaron las informaciones hacia el tema de interés de este estudio. Estos filtros se detallan en los puntos siguientes:

1. Sólo seleccionar los *tweets* en español. De esta forma, se incorpora como novedad el análisis de las opiniones sobre la vacuna COVID-19 en los países de habla hispana.
2. Uso de palabras claves, que aseguran que los *tweets* recopilados tienen como foco las posturas alrededor del proceso de vacunación.
3. Usuarios no repetidos. La plataforma de RRSS tiene dos tipos de Identificadores (Id), el primero se utiliza para identificar la publicación, mientras el segundo tipo se refiere al identificador del usuario en la App. En este estudio se aplicó un filtro para que no se repitieran estos últimos (los Id de usuarios) con la finalidad de evitar que la consulta se hiciera sobre un Id de usuario repetido[4].
4. Periodo de búsqueda. Se estableció un periodo de 30 días comprendidos entre el 15 de marzo de 2021 al 15 de abril de 2021. La selección de este período se basó en los eventos que ocurrieron durante ese tiempo, en particular, el veto experimentado por la farmacéutica AstraZeneca. AstraZeneca, en colaboración con la Universidad de Oxford, desarrolló una de las primeras vacunas contra el COVID-19 que obtuvo la aprobación para su uso en seres humanos. Este evento generó una serie de reacciones masivas en la población mundial, muchas de las cuales estaban impregnadas de ideas conspiratorias relacionadas con la efectividad, los efectos y la autenticidad de la vacuna y sus componentes.

La estrategia previamente mencionada se evaluó considerando una serie de ventajas y desventajas que son fundamentales para comprender la reproducibilidad de este estudio.

En cuanto a las ventajas, se destaca que el marco de datos proporcionado por *Panacea Lab* permite el uso de la paquetería *Tweeppy* en *Python*, lo que facilita la obtención del estado de los Id y otros componentes. Además, esta estrategia ofrece acceso a información

[4]Se estima que en media una persona postea al menos 5 *tweet* al día y esto puede aumentar cuando se trata de cuentas de alto alcance, es decir, con un volumen alto de seguidores.

histórica y garantiza que el contexto de los datos esté relacionado con la pandemia del COVID-19.

También se identificaron desventajas relevantes. Por la antigüedad de los *tweets*, se observó que aproximadamente el 30-40 % de las cuentas estaban suspendidas o que los mensajes ya no estaban disponibles. Además, la estrategia no diferencia entre los *hashtags* y el texto en los *tweets*, lo que requirió una revisión manual para eliminar posibles *tweets* no relacionados que utilizaban hashtags virales del tema. Por último, se encontró una limitación en el acceso a consultas, ya que solo se pueden realizar 1,000 consultas cada 15 minutos, lo que implicó desarrollar un código más especializado para generar consultas de manera recursiva.

2.2. Preprocesamiento de los datos

El preprocesamiento de datos es una etapa relevante en el análisis de datos y en la minería de datos, es donde los datos crudos se preparan y transforman para que sean adecuados y útiles para su posterior análisis. Su objetivo es mejorar la calidad y la eficiencia del análisis de datos, así como reducir la posibilidad de obtener resultados incorrectos debido a problemas en los datos.

En el preprocesamiento de análisis de texto, la limpieza se lleva a cabo con el propósito de eliminar todo lo que no contribuye a la estructura o contenido de los datos. Este proceso puede variar según el interés del análisis y la fuente de donde se recopila la información. Por ejemplo, al trabajar con datos que proviene de RRSS se esperaría eliminar ciertos caracteres (@) o iconos (emojis), enlace de páginas web y etiquetas (#). Existen diversas técnicas de preprocesamiento para texto. En este estudio se utilizaron principalmente aquellas que estuvieron presentes en la literatura revisada: *tokenización*, *Stop words*, *lemmatizacion* y la frecuencia de términos relevantes.

2.2.1. Tokenizacion

Se procedió a utilizar una función de limpieza de patrones no informativos, signos de puntuación, caracteres sueltos, números y enlaces. Esto también genero un proceso denominado *tokenización*.

La *tokenización* es un proceso usado en el análisis de NLP que consiste en convertir secuencias de caracteres (textos, párrafo, oración) en sus unidades más pequeñas que además contienen sentido semántico. Estas unidades es lo que se denomina *token*. La cantidad de *tokens* a obtener va a depender de la limpieza de información «no significativa» que se hayan aplicado, es decir, el proceso se realiza luego de la limpieza de caracteres, iconos, números y enlaces a los *tweet* recolectados. Se destaca que cada *tweets* representa un «documento»[5] al que se le debe aplicar el proceso de limpieza y *tokenización*.

La tabla 2.1 muestra el resultado del proceso de *tokenizacion* en los datos utilizados. A la izquierda se observa el texto original, es decir, tal como se recolectó de la aplicación de *Twitter* en formato texto. A la derecha se observa el resultado correspondiente, luego de aplicar la limpieza y *tokenizacion* obteniéndose una lista de los términos (cada uno es un *token*) que componen el texto asociado.

Texto original	**Texto tokenizado**
Ya están entrando las personas mayores de 60 a...	[entrando, personas, mayores, años, instalacio...
Catastrófica la incidencia en la República Che...	[catastrófica, incidencia, república, checa, v...
SemanaSanta2021 de Nuevo León a...	[semanasanta, autoridades, nuevo, león, anunci...
Gráfico con la evolución de los ingresos ho...	[gráfico, evolución, ingresos, hospitalarios, ...
Cómo el pueblo se va a tomar en serio las adve...	[cómo, pueblo, va, tomar, serio, advertencias,...

Tabla 2.1: Resultado del proceso de tokenización.

2.2.2. Palabras vacías o *Stop Words*

Stop Words o palabras vacías, es un término utilizado en el área de informática y hace referencia a aquellas palabras que no están registradas por los motores de búsqueda en internet como *Google*, las cuales carecen de sentido cuando se escriben solas o sin la palabra clave. Su uso está acotado al idioma en que se escriba o hable, pero aportan muy poco significado semántico por sí solas. Se componen de conjunciones, artículos, preposiciones y

[5]En lo adelante se usara de manera indistinta las palabras *tweet* y documento.

adverbios. El término *Stop Words* surge porque en informática son palabras que se filtran, bien sea antes o después del procesamiento de los datos en un proceso de NLP (o texto). Algunas de estas palabras son, por ejemplo: *[«en, y, con, la, el»]* entre otras.

La figura 2.1 presenta en forma de «nube» las palabras principales. Básicamente, se observa la no existencia de ninguna de las *Stop words* eliminando el peso que tendrían en el cálculo de la frecuencia de términos. Un aspecto principal de estas representaciones es el tamaño de la fuente, donde las palabras más frecuentes o importantes se representan con un tamaño de fuente más grande, lo que las hace más visibles y destacadas. En este caso, las palabras más frecuentes son «vacuna» y «covid» lo cual es coherente con el marco contextual que se está trabajando. Otras palabras frecuentes serían: «país», «vacunar», «adulto mayor» y «dosis», entre otras.

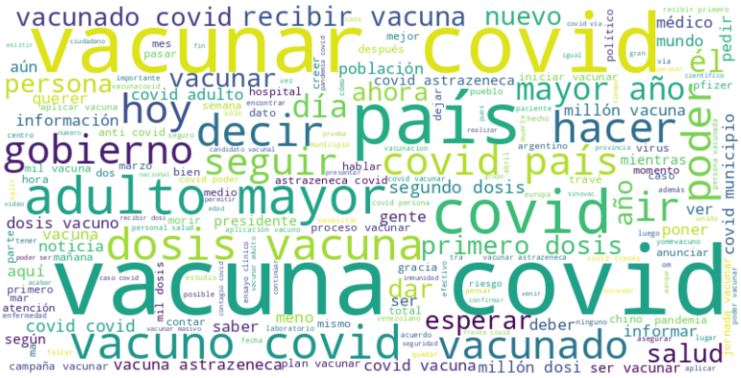


Figura 2.1: Nube de palabras resultante, luego de eliminar las *Stop word*.

2.2.3. Lematización

La *lematización* es un proceso lingüístico que consiste en agrupar las formas flexionadas o derivadas de una palabra en su forma canónica o *lemma* para que puedan representarse y/o estudiarse como un solo elemento. La lematización es una técnica ampliamente empleada en la lingüística computacional y el aprendizaje automático, especialmente en NLP. Su popularidad se debe a su capacidad para reducir la dimensionalidad del vocabulario y agrupar palabras relacionadas en el análisis de texto. Este enfoque mejora la precisión de tareas que involucran modelos de aprendizaje, tanto supervisado como no supervisado.

En el proceso aplicado a los datos recolectados se utiliza el paquete desarrollado por Honnibal et al. (2020) en *Python* que permite utilizar cualquiera de sus *pipelines*[6] de procesamiento de idiomas. En este caso se trabaja con «es_core_news_sm», que es un modelo con palabras en español. Estos modelos contienen reglas gramaticales, información léxica y estadísticas que el algoritmo utiliza para realizar tareas de procesamiento de lenguaje natural.

El algoritmo utiliza información sobre la categoría gramatical (sustantivo, verbo, adjetivo, etc.) y realiza lo que se conoce como etiquetado gramatical o análisis morfosintáctico (*Part of Speech*, POS), donde cada palabra en un texto se etiqueta con su categoría gramatical y otras características relacionadas con su función en la oración.

La tabla 2.2 muestra el resultado del proceso de *lemmatización*. A la izquierda se muestra el texto ya *tokenizado* mientras a la derecha se presentan los términos en su versión raíz. Palabras como «entrando» se convierte en la versión infinitivo del verbo y queda expresado como «entrar» o «personas» se convierte en su versión singular «persona». Esto muestra la eficiencia del modelo en detectar el contexto de las palabras más allá de solo su forma semántica.

Texto tokenizado	**Texto lemmatizado**
[entrando, personas, mayores, años, instalacio...	[entrar, persona, mayor, año, instalación, uaa...
[catastrófica, incidencia, república, checa, v...	[catastrófico, incidencia, república, checo, i...
[semanasanta, autoridades, nuevo, león, anunci...	[semanasantar, autoridad, nuevo, león, anuncia...
[gráfico, evolución, ingresos, hospitalarios, ...	[gráfico, evolución, ingreso, hospitalario, co...
[cómo, pueblo, va, tomar, serio, advertencias,...	[cómo, pueblo, ir, tomar, serio, advertencia, ...

Tabla 2.2: Resultado del proceso de *lemmatización*.

[6]Son una serie de pasos secuenciales interconectados que se utilizan en informática y programación para procesar datos o realizar tareas de manera automatizada y eficiente.

2.2.4. Frecuencia de Término-Inversa de Frecuencia de Documento

La Frecuencia de Término-Inversa de Frecuencia de Documento (*Term Frequency-Inverse Document Frequency*, TF-IDF) es una técnica estadística de amplio uso en el procesamiento de lenguaje natural para medir la importancia relativa de una palabra en un documento dentro de una colección de documentos. La TF-IDF combina dos conceptos:

1. **Frecuencias de términos** (*Term of Frequency*, TF): una manera sencilla de medir la importancia de un término dentro de un documento es utilizando la frecuencia con la que aparece. Sin embargo, es un hecho bien conocido que la longitud total de los documentos puede variar desde muy pequeña hasta muy grande en relación con la capacidad computacional, por lo que es posible que un término pueda ocurrir con más frecuencia en documentos grandes en comparación con documentos pequeños. Para corregir este problema, la ocurrencia de cualquier término en un documento se divide por el total de términos presentes en ese documento para encontrar la frecuencia de términos (Qaiser and Ali, 2018).

   Por lo tanto, el cálculo del TF está dado por la siguiente fórmula:

   $$TF(t,d) = \frac{n_t}{longitud_d}, \tag{2.1}$$

   donde n_t es el número de veces que aparece el término t en el documento d y $longitud_d$ es la cantidad de documentos analizados.

   Esta aproximación, aunque simple, tiene la limitación de atribuir mucha importancia a aquellas palabras que aparecen muchas veces, aunque no aporten información selectiva. Cuando se calcula la frecuencia de término de un documento, se puede observar que el algoritmo trata a todas las palabras clave por igual, sin importar si es una palabra de parada como «de», lo cual es incorrecto. Todas las palabras clave tienen diferente importancia (Qaiser and Ali, 2018).

2. **Frecuencia Inversa** (*Inverse Document Frequency*, IDF): para subsanar el problema mencionado anteriormente se ponderan los valores TF multiplicándolos por la inversa de la frecuencia con la que el término en cuestión aparece en el resto de los documentos. La frecuencia inversa de documento asigna un peso más bajo a las palabras frecuentes y un peso mayor a las palabras poco frecuentes.

   El cálculo de IDF está dado por la fórmula:

$$IDF(t) = log\left(\frac{n_d}{n_{(d,t)}}\right), \tag{2.2}$$

donde n_d es el número total de documentos y $n_{(d,t)}$ es el número de documentos que contienen el término t.

El objetivo de esta técnica, en lugar de utilizar las frecuencias brutas de los *token* en un documento, es reducir la influencia de los *token* que aparecen con alta frecuencia en un cuerpo de texto. Estos *token* tienden a ser menos informativos en comparación con las características que ocurren raramente en el corpus[7] de entrenamiento. Además, se da prioridad a estas características menos frecuentes, ya que pueden proporcionar un contexto más relevante (Pedregosa et al., 2011a).

Finalmente, para calcular el valor del TF-IDF se utiliza la fórmula:

$$\text{TF-IDF} = TF \times IDF. \tag{2.3}$$

En la tabla 2.3, se muestra el resultado de la aplicación del método TF-IDF al conjunto de documentos utilizado en este estudio. El TF-IDF es una técnica utilizada en procesamiento de lenguaje natural que asigna un puntaje a cada palabra (o *tokens*) en función de su importancia en relación con el conjunto de documentos analizados. Los cinco primeros *tokens* con los puntajes de TF-IDF más altos son «vacunar», «vacuna», «país», «dosis» y «vacuno».

[7]Conjunto lo más extenso y ordenado posible de datos o textos científicos, literarios, etc., que pueden servir de base a una investigación (Real Academia Española, 2023).

Token	Conteo	TF	IDF	TF-IDF
vacunar	4,218	0.027	1.104	0.029
vacuna	6,064	0.038	0.746	0.029
país	2,308	0.015	1.793	0.026
dosis	1,833	0.012	1.975	0.023
vacuno	1,337	0.008	2.198	0.019
mayor	1,070	0.007	2.446	0.017
año	1,031	0.007	2.498	0.016
poder	1,006	0.006	2.534	0.016
vacunado	912	0.006	2.604	0.015
salud	862	0.005	2.676	0.015

Tabla 2.3: Top 10 de *tokens* con mayor valor del TF-IDF.

Asimismo, las palabras «vacunar» y «vacuna» tienen un valor de TF-IDF casi idéntico, lo que sugiere que son términos clave en los documentos relacionados con la vacunación. «País» y «dosis» también son términos relevantes, aunque menos frecuentes. Por otro lado, palabras como «mayor» y «año» tienen un valor de TF-IDF más bajo, lo que indica que son menos relevantes en este contexto. Estos resultados proporcionan información valiosa para identificar las palabras más importantes en el conjunto de documentos y entender mejor los temas clave relacionados con la vacunación contra el COVID-19.

Capítulo 3

Word Embedding: Representación vectorial de palabras

En este capítulo, se proporciona una base teórica para que se comprendan los conceptos clave sobre la incrustación y vectorización de palabras y su uso en modelos NLP antes de adentrarse en la arquitectura del modelo *Word2Vec*.

A través de los modelos de espacio vectorial podemos representar objetos de manera algebraica mediante vectores continuos en un espacio multidimensional y usando métricas de distancias poder medir la similitud entre estos. Tradicionalmente, la forma en cómo se construyen estos espacios semánticos está basada en el uso de matrices de co-ocurrencia que registran la frecuencia con la que N elementos específicos co-ocurren en un conjunto de datos. En el contexto del NLP, una matriz de co-ocurrencia de palabras registra cuántas veces dos palabras diferentes aparecen juntas en un mismo contexto, como en una oración o un documento.

Pero, ¿cómo pueden las co-ocurrencias de palabras denotar similitud semántica? La idea principal aquí es la hipótesis distribucional (establecido por el lingüista británico John Rupert Firth, en su publicación Firth (1957)), según la cual *«una palabra se caracteriza por la compañía que mantiene»*. Es decir, aquellas palabras que aparecen en contextos similares son propensas de tener significados similares. Un ejemplo de esto serían *Júpiter* y *Venus*, ya que generalmente aparecen en contextos similares, como por ejemplo: sistema solar, estrella, planeta y astronomía. Por lo tanto, se pueden recopilar estadísticas de co-ocurrencia de palabras e inferir relaciones semánticas (Pilehvar and Camacho-Collados,

2021).

Cuando se trabaja con matrices de co-ocurrencia introducimos un concepto importante como es el tamaño de la ventana de palabras. El tamaño de la ventana es un hiperparámetro importante, ya que influye en como se capturan las relaciones contextuales entre las palabras. Ventanas más grandes pueden introducir más ruido, mientras que ventanas más pequeñas se centran en un contexto más estrecho y específico.

En la figura 3.1 usando la sentencia "*The quick brown fox jumps over the lazy dog.*" se muestra las palabras de entrenamiento que se generan al seleccionar la palabra objetivo (en recuadro azul) y las palabras de contexto (recuadro en blanco) de un texto usando una ventana de tamaño 2.

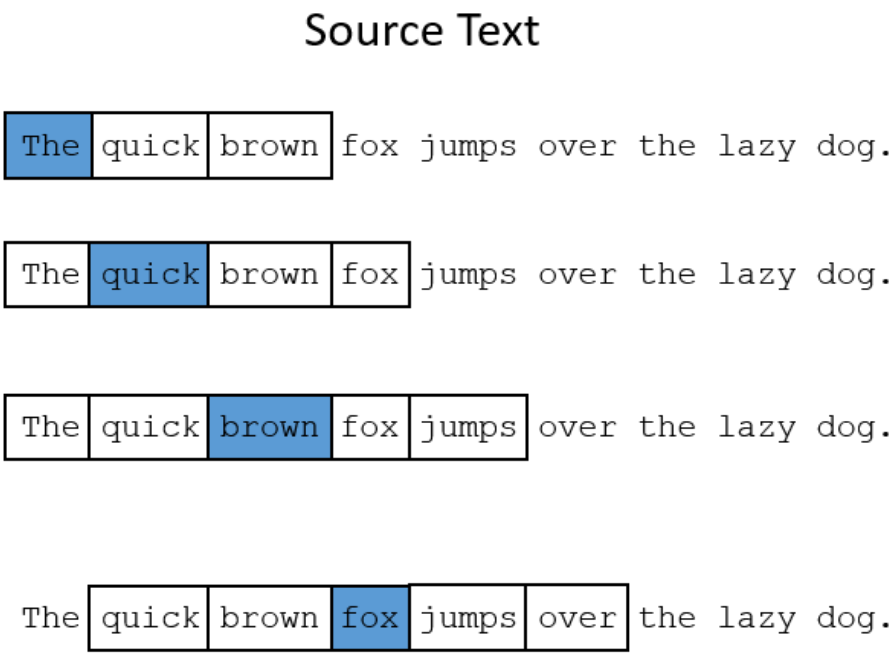


Figura 3.1: Ejemplo de la generación de *Word Embedding* en modelos *Word2vec* (McCormick, 2019).

En la figura 3.2 se observa un ejemplo de matriz de co-ocurrencia para tres sentencias

y una ventana de palabras igual a 1. La ventana de palabras define cuántas palabras circundantes (contexto) se tendrán en cuenta para aprender la representación vectorial de la palabra objetivo: *«I enjoy flying.»*, *«I like NLP.»* y *«I like deep learning.»*. La matriz recoge las coincidencias entre las palabras de cada sentencia, indicando el número de veces que están relacionadas, por ejemplo: [*«I»*, *«like»*] resulta ser dos, dado que ambas palabras aparecen juntas dos veces en la segunda y tercera oración. Igual que un valor 1 para [*«I»*, *«enjoy»*] que co-ocurren en la sentencia 1.

1. I enjoy flying.
2. I like NLP.
3. I like deep learning.

$$X = \begin{array}{c} \\ I \\ like \\ enjoy \\ deep \\ learning \\ NLP \\ flying \\ . \end{array} \begin{array}{c} \begin{array}{cccccccc} I & like & enjoy & deep & learning & NLP & flying & . \end{array} \\ \begin{bmatrix} 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Figura 3.2: Matriz de co-ocurrencia para 3 sentencias y una ventana de 1 (Chaubard et al., 2019).

Sin embargo, existen varios obstáculos para inferir la semántica de las palabras a partir de estadísticas de co-ocurrencia que van desde la ambigüedad de las palabras, por ejemplo: estrella puede ser utilizada en contexto sobre universo, sistema solar, o en contexto sobre actuación, deporte u otras; hasta y no menos relevante el aumento de la complejidad computacional que requiere a medida que aumenta el número de palabras y/o documentos a revisar.

Es aquí donde se incorpora las técnicas de aprendizaje automático que con el avance experimentado en los últimos años permiten entrenar modelos más complejos con grandes corpus de texto, permiten reducir la complejidad computacional superando la eficiencia

y resultados de los modelos más simples. La idea es diseñar un modelo cuyos parámetros sean los vectores de palabras. Luego, entrenar el modelo en un objetivo específico. En cada iteración, ejecutamos el modelo, evaluamos los errores y seguimos una regla de actualización que tiene cierta noción de penalizar los parámetros del modelo que causaron el error (Chaubard et al., 2019).

Este proceso ya es conocido y es lo que se funciona detrás de modelos de Redes Neuronales (RN), que en el caso del NLP se verían implementados en los Modelos de Lenguaje de Redes Neuronales (*Neural Net Language Model, NNLM*) los cuales, a diferencia de los modelos tradicionales, que consideran las palabras como unidades discretas y atómicas, utilizan representaciones distribuidas de palabras. Cada palabra se representa como un vector numérico de valores continuos, lo que permite que las palabras con significados y contextos similares tengan representaciones cercanas en el espacio vectorial, esto es lo que se conoce como vector de palabras o *embedding*. Se construyen aprovechando redes neuronales, principalmente popularizados después de 2013, con la introducción de *Word2vec* desarrollado por un equipo de *Google* y presentado en Mikolov et al. (2013a).

Los *embedding* o *Word Embedding* son una representación vectorial de valores reales de palabras al incrustar tanto significados semánticos como sintácticos obtenidos de un corpus grande y no etiquetado (Wang et al., 2019). Es una herramienta poderosa utilizada ampliamente en tareas modernas de NLP, incluido el análisis semántico, la recuperación de información, el análisis de dependencias, la respuesta a preguntas y la traducción automática.

3.1. Fundamentos del modelo *Word2Vec*

Los modelos NNLM se utilizan para predecir la probabilidad de ocurrencia de una palabra, dada su historia contextual, en una secuencia de palabras. Cada palabra se representa como un vector numérico de valores continuos, lo que permite que las palabras con significados y contextos similares tengan representaciones cercanas en el espacio vectorial y es una de sus muchas ventajas versus los modelos tradicionales que tratan las palabras como unidades atómicas que se representan como índices de un vocabulario. La elección de estos modelos se basa en razones válidas, son simples, robustos y la noción de que si se entrenan con grandes corpus de palabras superarían a otros sistemas, que si bien pudieran trabajar con menos datos, en términos de su modelación matemática,

serían mucho más complejos. Ejemplo de esto son los modelos *N-gram*[1] que se pueden entrenar con billones de palabras.

Sin embargo, las técnicas simples tienen sus límites en muchas tareas. Por ejemplo, la cantidad de datos relevantes en un dominio específico para el reconocimiento automático del habla es limitada. Generalmente, el rendimiento está dominado por el tamaño de los datos de habla transcrito de alta calidad. Por lo tanto, existen situaciones en las que simplemente aumentar la escala de las técnicas básicas no resultará en un progreso significativo, y debemos centrarnos en técnicas más avanzadas (Mikolov et al., 2013a).

Para esto, los *Word Embedding* modelados a partir de RN han demostrado ser mucho mejores. De esto se trata el modelo implementado por Mikolov et al. (2013a), que ha sido bien recibido y replicado por muchos otros autores con resultados satisfactorios, basándose en el trabajo previo de otros autores que trabajan con representaciones de vectores continuos y/o con la implementación de modelos de Redes Neuronales Recurrentes que aprenden conjuntamente una representación de vectores de palabras y que contiene una capa de proyección lineal y una capa oculta no lineal.

Word2vec es un modelo, NNLM el cual aprende conjuntamente una representación vectorial de palabras y un modelo de lenguaje estadístico mediante una RN *feedforward* que incluye una capa de proyección lineal y una capa oculta no lineal. Se utiliza un vector $one - hot$ de N dimensiones que representa la palabra como entrada, donde N es el tamaño del vocabulario. La entrada se proyecta primero en la capa de proyección. Luego, se utiliza una operación *softmax* para calcular la distribución de probabilidad sobre todas las palabras en el vocabulario (Wang et al., 2019).

La codificación $one - hot$ es bastante sencilla, se trata una codificación binaria en donde colocaremos un «1» en la posición correspondiente a la palabra «objetivo», y 0 en todas las demás posiciones. A continuación, se presenta un ejemplo de *one-hot*:

[1]Un N-gram es una secuencia de N palabras, por ejemplo: un 2-gram (llamado bigrama) es una secuencia de dos palabras como «por favor», «gire su», o «su tarea», y un 3-gram (trigrama) es una secuencia de tres palabras como «por favor gire», o «gire su tarea». Los modelos N-gram se usan para estimar la probabilidad de la última palabra de un N-gram dadas las palabras previas (Jurafsky et al., 2014).

$$
\begin{aligned}
Corpus: \ & El\ gato\ esta\ durmiendo \\
El \quad &= [1 \quad 0 \quad 0 \quad 0] \\
Gato \quad &= [0 \quad 1 \quad 0 \quad 0] \\
Esta \quad &= [0 \quad 0 \quad 1 \quad 0] \\
Durmiendo \quad &= [0 \quad 0 \quad 0 \quad 1]
\end{aligned}
$$

La figura 3.3 hace referencia a la arquitectura de modelo de RN para lenguaje con solo 3 capas que es el que corresponde con el modelo *Word2vec*, la primera capa es la que proyecta el vector de entrada con la codificación $one - hot$, la capa oculta que en el caso del modelo *Word2vec* es lineal y una capa de salida donde los vectores resultantes son previamente transformados por la función *softmax*. La salida de la red es un único vector (del mismo tamaño del vocabulario utilizado) que contiene, para cada palabra del vocabulario, la probabilidad de que una palabra cercana seleccionada al azar coincida con esa palabra del vocabulario.

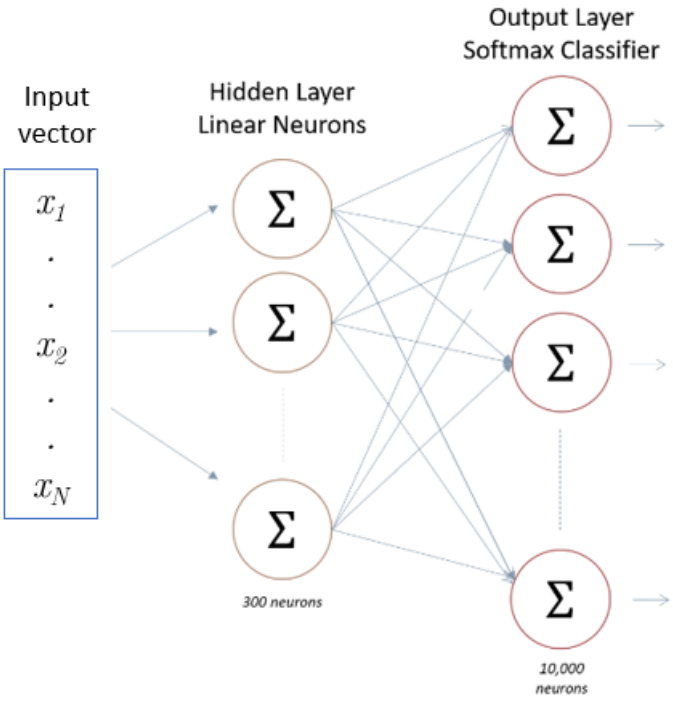


Figura 3.3: Modelo de Red Neuronal para lenguaje *Word2vec*(McCormick, 2019).

La función *softmax* se formula de la forma siguiente:

$$g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}. \tag{3.1}$$

donde z_m representa cada elemento del vector z y k es el tamaño del vector. Cada resultado es normalizado dividiendo por la suma del exponencial de todo el vector. La operación de normalización permite que cada elemento en el vector de salida sume 1.

Para predecir la clase de una muestra, el modelo calcula un puntaje para cada clase y hace pasar el vector de puntaje por la función *softmax*. El concepto matemático, detrás explicado de manera simple, es la conversión de un vector de k números reales en otro vector de igual tamaño con valores en el rango [0, 1]. Lo que da como resultado la distribución de probabilidad de los k valores que representan los vectores de *embedding*.

El trabajo de Mikolov et al. (2013a) sigue la misma línea que ha planteado en trabajos posteriores desde la publicación en 2009 de *Neural network based language models for higly inflective languages* (Mikolov et al., 2009) donde los vectores de palabras se aprenden primero utilizando una red neuronal con una sola capa oculta. Estos vectores se utilizan luego para entrenar el modelo NNLM. De esta manera, los vectores de palabras se aprenden incluso sin construir completamente el NNLM.

En ese sentido, *Word2vec* es una implementación de RN para estimación de palabras en un texto, y su resultado no es un simple algoritmo, más bien, es una familia de arquitecturas y optimizaciones de modelos que se pueden utilizar para aprender las *embeddings* de palabras a partir de conjuntos de datos grandes. Es decir, el modelo internamente posee metodologías con alto potencial y que se han de seleccionar según la finalidad y los datos disponibles. Específicamente, en Mikolov et al. (2013a) los autores proponen dos metodologías para la predicción de las palabras a través de la red: **Continuous Bag of Words, CBOW** y **Skip-gram**. El primer método proviene de los modelos de bolsa de palabras *Bag of Words* donde el orden de las palabras en el historial no influye en la proyección que predice una palabra a partir de las palabras denominadas «contexto» situadas en una ventada hacia delante y detrás de la palabra «central» que representa la palabra a estimar. Una explicación más detallada de este método se estaría presentando en la Sección 3.2.1. Mientras que con la segunda metodología se realiza la operación opuesta, es decir, se estiman las palabras contexto a través de una palabra central.

En el primer trabajo, Mikolov et al. (2013a) se plantea el uso de un único método de optimización basado en la misma metodología de los modelos NNLM usando *softmax* jerárquico. Con este método, en lugar de calcular directamente las probabilidades para todas las palabras en el vocabulario, se utiliza la estructura de un árbol binario de Huffman[2] para agrupar las palabras y calcular probabilidades más eficientemente. Las palabras más frecuentes se representan cerca de la raíz del árbol y las menos frecuentes en los niveles más bajos. Esto reduce drásticamente la cantidad de cálculos requeridos para obtener las probabilidades, lo que acelera el proceso de entrenamiento ofreciendo mayor eficiencia.

Posteriormente, en un segundo trabajo Mikolov et al. (2013b) proponen una metodología alternativa denominada **Muestreo Negativo** (*Negative sampling*) en lugar de considerar todas las palabras en el vocabulario en cada iteración, el muestreo negativo selecciona un pequeño número de palabras negativas de manera arbitraria (palabras que no están en el contexto) para cada palabra objetivo. En esencia, en lugar de calcular y ajustar los valores de todas las palabras en cada iteración, el muestreo negativo solo se enfoca en un subconjunto pequeño y aleatorio de palabras.

Dicha muestra negativa posee su distribución de probabilidades y es denominada «ruido». Por lo tanto, la tarea consiste en distinguir la palabra objetivo de las muestras obtenidas de la distribución de ruido utilizando regresión logística, donde hay n ejemplos negativos para cada muestra de datos.

Con el avance de las técnicas y la capacidad de los software de hoy en día se han ido mejorando y adicionando nuevos modelos que siguen la misma línea de *Word2vec* con mejoras o técnicas diferentes, pero que siguen aportando al mundo del análisis de texto basado en *Word Embedding*.

[2]Creados por David Huffman en 1952. Huffman describió un algoritmo que codificaba símbolos de un mensaje a transmitir, de tal manera que los símbolos con una frecuencia de aparición más alta eran codificados con secuencias de longitud más corta que los símbolos con una frecuencia de aparición más baja. El algoritmo propuesto creaba códigos de longitud variable y sin prefijos únicos decodificables. La codificación que propone utiliza la estructura de datos de árbol para extraer las secuencias de código de cada símbolo distinto que aparece en el mensaje fuente (Xezonakis and Leivadaros, 2021).

3.2. Implementación práctica

La idea detrás de *Word2vec* ha sido estudiada y aplicada en diversos estudios, por lo que, hoy en día, existen diversos paquetes que modelan la técnica. Para los fines de este estudio se usa la establecida en *Python* centrándonos en los algoritmos y optimizadores utilizados para el modelo que se propone: modelo de Bolsa de palabras con Vectores Continuos (CBOW) y Muestreo negativo.

3.2.1. Método CBOW

El primer modelo propuesto por Mikolov et al. (2013a) es el modelo de CBOW. Este trabaja prediciendo la palabra central a estimar a partir de las palabras que la acompañan en una misma sentencia y le dan contexto, estableciendo una ventana que determina el número de estas palabras de contexto.

El procedimiento maximiza la probabilidad:

$$P(w_i|w_{i-c}, w_{i-c+1}, ..., w_{i-c}, w_{i+c}, ..., w_{i+c-1}, w_{i+c}), \tag{3.2}$$

donde w_i es la palabra en la posición i y c el tamaño de la ventana. Se supone que M representa el conjunto de palabras del corpus de texto, definiremos los parámetros principales como:

(i) una matriz entrada $V \in \mathbb{R}^{N \times |M|}$, donde N es el tamaño del espacio de *embeddings* tal que la columna i-ésima de V es el vector de *embedded* de N-dimensiones para la palabra w_i,

(ii) un vector de salida $U \in \mathbb{R}^{|M| \times N}$ donde la j-esima fila representa el vector de *embedded* de N-dimensión para la palabra de salida u_j.

El proceso se corresponde con una RN para un modelo NNLM con una estructura «simple» de una capa de entrada, una capa oculta y una capa de salida, como se muestra

en la figura 3.4. Se resumirá el proceso en pasos, dando por sentado el conocimiento medio sobre modelos de RN que no es el fin de este estudio desarrollar. Sin embargo, si es de interés del lector, puede hacer una revisión al trabajo de Schmidt (2019) o en libros como el de Montavon (2020).

La figura 3.2.1 representa la RN del modelo *Word2vec* para un modelo CBOW que se distingue por la primera capa (lado izquierdo) con la introducción de un vector de entrada de varios términos (que serían codificados como 1) que representan las palabras de contexto para estimar la palabra objetivo.

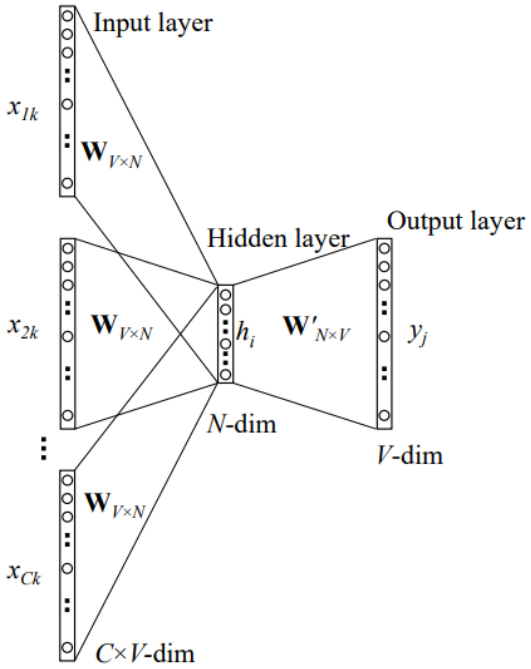


Figura 3.4: Modelo con CBOW (Rong, 2014).

Matemáticamente, se explica en los siguientes pasos:

1. En la primera capa, la capa de entrada, se tiene la representación vectorial de las palabras en el texto codificado como *one-hot* $(x_1, x_2, x_3....x_N)$. Se denota W como una matriz de tamaño $V \times N$. Cada fila de esta matriz es de tamaño del vector V de la palabra asociada. Se tendría:

$$\mathbf{h} = \mathbf{W}^T x := v_{wi}^T. \tag{3.3}$$

Esto implica que la función de enlace (activación) de las unidades de la capa oculta es lineal, es decir, pasa directamente la suma ponderada de sus entradas a la siguiente capa (Rong, 2014).

2. Desde la capa oculta hasta la capa de salida, hay una matriz de peso diferente $\mathbf{W}' = w'_{ij}$, de tamaño $N \times V$. Utilizando estos pesos que se generan de manera aleatoria siguiendo una distribución uniforme con valores entre [-1,1], se puede calcular un puntaje u_j para cada palabra en el vocabulario. Se tendría:

$$u_j = {v'_{wj}}^T \mathbf{h}, \tag{3.4}$$

donde v'_{wj} representa la j-esima columna en $\mathbf{W}'$. A partir de este punto se introduce la función *softmax* para generar la distribución de probabilidad de los vectores de palabras:

$$P(w_c|w_i) = y_j = \frac{exp(u_j)}{\sum_{j'=1}^{|V|} exp(u_{j'})}, \tag{3.5}$$

donde y_j es el resultado de la j-esima unidad de la capa de salida. Al sustituir (3.3) y (3.4), la función *softmax* queda como:

$$P(w_c|w_i) = \frac{exp({v'_{wj}}^T v_{wi})}{\sum_{j'=1}^{|V|} exp({v'_{wj'}}^T v_{wi})}. \tag{3.6}$$

donde v_w y v'_w son representaciones de la palabra w. El primero proviene de las filas de la matriz de $\mathbf{W}$ que es la matriz de pesos que se inicializa en el procesamiento entre la capa de entrada y la capa oculta, mientras la segunda proviene de las columnas de $\mathbf{W}'$ que es la matriz que se encuentran entre la matriz de la capa oculta y la capa de salida. En ese sentido, se tiene que v_w es el vector de entrada y v'_w es el vector de salida de la palabra w.

En la ecuación (3.3) se supone que la ventana de palabras contexto es 1, es decir, un modelo CBOW con una sola palabra contexto, pero en general se trabaja con modelos de múltiples palabras, lo que modificaría su planteamiento debido a que bajo un modelo múltiple se emplea el vector promedio de los *embedding* de las palabras contexto, en ese orden, se reestructura la ecuación como:

$$\begin{aligned}\mathbf{h} &= \frac{1}{c}\mathbf{W}^T(x_1 + x_2 + x_3 + ... + x_c) \\ &= \frac{1}{c}(v_{w1} + v_{w2} + v_{w3} + ... + v_{wc})^T\end{aligned} \tag{3.7}$$

El modelo CBOW se obtiene al minimizar la pérdida de entropía cruzada entre el vector de probabilidad y el vector embebido de la palabra de salida. Esto se logra al minimizar la siguiente función objetivo (Wang et al., 2019):

$$\begin{aligned}&- \log P(w_O|w_{i1}, ..., w_{ic}) \\ &= -u_{j*} + \log \sum_{j'=1}^{V} \exp(u_{j'}) \\ &= -{v'_{wO}}^T \cdot \mathbf{h} + \log \sum_{j'}^{V} \exp({v'_{wj}}^T \cdot \mathbf{h}),\end{aligned} \tag{3.8}$$

donde $P(w_O|w_{i1}, ..., w_{ic})$ representa la probabilidad condicional de observar la palabra de salida real w_O (denotando su índice en la capa de salida como $j*$) dado el contexto de entrada w_i con relación a los pesos ($\mathbf{W}$).

Si bien cuando se habla de modelos de RN, se suele utilizar la función *softmax* para obtener la función de distribución de los vectores de salida, esta función no es la más eficiente. Como se observa en (3.5) el denominador requiere una suma sobre todo el tamaño del vocabulario. Por lo que, cuando se tiene un tamaño de vocabulario grande, se vuelve costoso y lento normalizar todos y cada uno de los ejemplos de entrenamiento, sumando los resultados de cada palabra de vocabulario, y se introducen técnicas de muestreo eficientes.

En la aplicación de la metodología de *Word2vec* se cuenta con dos técnicas adicionales que según los resultados revisados obtiene resultados más eficientes cuando se trabaja con corpus grandes. El ejercicio empírico de este estudio fue realizado con uno de ellos, específicamente con muestreo negativo, y se describe en la siguiente sección.

3.2.2. Muestreo negativo

La idea general de *Word2vec* se puede interpretar en que aquellas palabras que aparecen en el mismo contexto tendrán representaciones vectoriales similares. En (3.6) el numerador reflejaría esto asignando un valor mayor para palabras similares a través del producto de puntos de los dos vectores. Si las palabras no aparecen en el contexto de la otra, las representaciones serán diferentes y, como resultado, el resultado será un valor pequeño.

Sin embargo, el principal reto surge cuando se calcula el denominador, que es un factor normalizador que debe computarse sobre todo el vocabulario. Dado que el tamaño del vocabulario puede alcanzar cientos de miles o incluso varios millones de palabras, el cálculo se vuelve una tarea imposible en términos del costo computacional. Aquí es donde entra en juego el muestreo negativo y hace que este cálculo sea factible.

El modelo de Estimación Contrastiva de Ruido (*Noise Contraction Estimation*, NCE) es un método poderoso en la estimación de parámetros para modelos log-lineales, que evita el cálculo de la función de partición o sus derivadas en cada paso de entrenamiento (Ma and Collins, 2018). En Mikolov et al. (2013b) los autores mencionan que el muestreo negativo es una simplificación del modelo, NCE procurando mantener la calidad en las representaciones vectoriales necesarias para el modelo *skip-gram* en el cual el objetivo principal es predecir las palabras circundantes (contexto) a partir de una palabra de entrada (objetivo) y en el que se utiliza todo el conjunto de entrenamiento, por lo que puede ser computacionalmente más costoso que el muestreo negativo. Otros autores también resaltan la similitud de ambos métodos.

En NCE se etiquetan la palabra objetivo verdadera con 1 y muestras aleatorias de las palabras objetivo incorrectas con 0. Esto es como entrenar un modelo para predecir: «¿Cuál de estas palabras es real y cuál es ruido?», para esto utiliza un modelo de regresión logística para responder a esta cuestión planteada. Básicamente, el modelo postula que un buen modelo debe poder diferenciar entre datos y ruido mediante regresión logística (Jost, 2019). Dado la amplia base teórica y práctica que existe sobre los modelos de regresión logística, sería un ejercicio simple. Se quiere las probabilidades logarítmicas de que una clase provenga de la verdadera distribución de palabras P en lugar de una distribución de ruido Q:

$$RegLog = log\left(\frac{P}{Q}\right) = log(P) - log(Q). \tag{3.9}$$

Se desconoce la distribución real de P, mientras la distribución Q es la que se usa para generar nuestras muestras negativas. Esto compara la distribución de datos, que se está tratando de aprender, con una distribución de ruido de referencia, de ahí el nombre de «Estimación de Contraste de Ruido». La idea general es convertir un problema de clasificación multinomial en un problema de clasificación binaria. Es decir, en lugar de usar *softmax* para estimar una verdadera distribución de probabilidad de la palabra de salida, se usa una regresión logística binaria.

La principal diferencia entre el muestreo negativo y NCE es que esta última necesita tanto las muestras como las probabilidades numéricas de la distribución de ruido, mientras que el muestreo negativo utiliza solamente muestras. Y aunque NCE maximiza aproximadamente la probabilidad logarítmica del *softmax*, esta propiedad no es importante para su aplicación (Mikolov et al., 2013b).

En pocas palabras, al definir una nueva función objetivo, el muestreo negativo tiene como objetivo maximizar la similitud de las palabras en el mismo contexto y minimizarla cuando ocurren en diferentes contextos. Sin embargo, en lugar de hacer la minimización de todas las palabras del diccionario, excepto las palabras de contexto, selecciona aleatoriamente una cantidad de palabras ($2 \leq k \leq 20$) dependiendo del tamaño del entrenamiento y las utiliza para optimizar el objetivo. La función objetivo propuesta por los autores es la siguiente:

$$log\ \sigma({v'_{w_O}}^T v_{w_i}) + \sum_{i=1}^{k} E_{w_i}\ P_n(w) \left[log\ \sigma(-{v'_{w_i}}^T v_{w_i})\right]. \tag{3.10}$$

donde σ es la función sigmoide, y $P_n(w)$ es la distribución de ruido con las muestras negativas.

La derivación de (3.10) proviene de la aplicación de la función sigmoide que se utiliza para clasificar si una muestra determinada es verdadera o falsa en función de la probabilidad calculada y se formula como:

$$\sigma(x) = \frac{1}{1 + exp(-x)} \tag{3.11}$$

El muestreo negativo convierte la tarea de clasificación múltiple en una tarea de clasificación binaria. El nuevo objetivo es predecir, para cualquier conjunto de palabras, (w, c) que se posicionan cerca una de la otra en los datos utilizados como entrenamiento, siendo w una palabra central y c su contexto. Por lo tanto, se puede denotar $P(D = 1|w, c)$ como la probabilidad de que provengan de los datos del corpus de texto. Entonces, la probabilidad de que no provengan de los datos del corpus de texto será $P(D = 0|w, c) = 1 - P(D = 1|w, c)$. Estableciendo los parámetros de entrenamiento de la distribución de probabilidad como θ y los parámetros fuera de estos como D' (Karimi, 2023), se tiene que optimizar lo siguiente:

$$\arg\max_{\theta} \prod_{(w,c)\in D} P(D = 1|w, c; \theta) \prod_{(w,c)\in D'} P(D = 0|w, c; \theta), \tag{3.12}$$

al sustituir $P(D = 0|w, c; \theta)1 - P(D = 1|w, c; \theta)$ y convirtiéndolo en un máximo de suma de logaritmos, se tiene:

$$\arg\max_{\theta} \sum_{(w,c)\in D} \log\, P(D = 1|w, c; \theta) + \sum_{(w,c)\in D'} \log\, (1 - P(D = 1|w, c; \theta)). \tag{3.13}$$

Se calcula $P(D = 1|w, c; \theta)$ usando la función sigmoide:

$$P(D = 1|w, c; \theta) = \sigma(v_c.v_w) = \frac{1}{1 + e^{-v_c.v_w}}, \tag{3.14}$$

donde v_w y v_c son las representaciones vectoriales de la palabra principal y de contexto, respectivamente. Por lo tanto, la ecuación (3.13) se convierte en:

$$\arg\max_{\theta} \sum_{(w,c)\in D} \log\, \sigma(v_c.v_w) + \sum_{(w,c)\in D'} \log\, (\sigma(-v_c.v_w)), \tag{3.15}$$

que representa la misma fórmula que la ecuación (3.10) aplicada en todo el corpus del texto. Para el modelo CBOW la ecuación (3.10) quedaría:

$$-log\ \sigma({v'_{w_O}}^T \mathbf{h}) - \sum_{k=1}^{K} log\ \sigma(-{\hat{v}'_{w_k}}^T \mathbf{h}). \tag{3.16}$$

En la fórmula anterior $(\hat{v}'_{w_k}|k = 1....K)$ representa la muestra tomada a partir de $P_n(w)$.

3.3. Resultados *Word Embedding*

En la tabla 3.1 se muestran los parámetros utilizados en la aplicación del modelo *Word2vec* en el paquete disponible en la plataforma de *Python*. Los parámetros se detallan a continuación:

- El *tamaño del vector* es el número de dimensiones sobre las cuales se van a representar las palabras. Cuanto mayor sea este valor, más grande será el espacio vectorial y, por lo tanto, los vectores tendrán más dimensiones. La literatura concuerda con un valor en un rango de $[100 - 1000]$.
- El *método* es el explicado en la Sección 3.2.1, es decir, CBOW.
- *Ventana* se refiere al tamaño de la ventana contextual utilizado para entrenar el modelo. Controla la cantidad de palabras vecinas que se consideran al predecir una palabra objetivo.
- *Mínimo conteo* es el número mínimo de veces que una palabra debe aparecer en el corpus para ser considerada en el entrenamiento. Las palabras que aparecen con poca frecuencia pueden ser excluidas para mejorar la calidad del modelo.
- *Epoch* es el número de iteraciones completas a través del conjunto de datos de entrenamiento.
- *Tamaño de muestra* es un parámetro asociado al método de Muestreo Negativo y se refiere al número de palabras a seleccionar en el *samplig*, es decir, la cantidad de muestras de palabras no relacionadas que se toman para cada palabra objetivo.

Parámetro	Selección
Tamaño del vector	50/100
Metodo	CBOW
Mínimo conteo	5
Ventana	5
Epoch	10
Tamaño de sampling	10

Tabla 3.1: Parámetros seleccionados para el modelo *Word2vec*.

Estas dimensiones donde se representarían las palabras son aprendidas automáticamente por el modelo a partir de grandes cantidades de texto y se optimizan para capturar relaciones semánticas y lingüísticas entre palabras en ese contexto particular. En otras palabras, las dimensiones en un vector de palabras representan características abstractas y específicas del contexto en el que se entrena el modelo. El aprendizaje de estas dimensiones se basa en las co-ocurrencias de palabras en oraciones o documentos, y el modelo intenta organizar estas dimensiones de manera que palabras similares tengan representaciones cercanas en el espacio vectorial.

Usando estos parámetros en conjunto con los métodos seleccionados, se entrenó un modelo para obtener la representación vectorial de los 11,787 documentos recolectados mediante la API de *Twitter*. La figura 3.5 permite de manera visual ver la relación que poseen las palabras a partir de su representación vectorial.

Las palabras que aparecen cercanas tienen vectores similares en el espacio vectorial original. Esto indica que estas palabras tienen significados semánticos[3] similares o están relacionadas en el contexto en el que se entrenó el modelo. Por ejemplo, las palabras *mayor* y *municipio*, así como *poder*, *gobierno* y *salud*, representan conjuntos de palabras con significados similares o relacionados.

Por otro lado, cuanto más lejos estén dos palabras, menos similares son en el contexto

[3]Los patrones o espacios semánticos se refieren a las relaciones de significado y contexto entre las palabras dentro de un conjunto de datos de texto tales como: analogías, sinonimia, hiperonimia, antonimia y otras.

en el que se entrenó el modelo. También algunas palabras a lo largo de las figuras se mantienen alejadas y sin agrupación con relación al resto. Estas palabras pudieran ser del tipo atípicas o tendrían significados muy distintos en comparación con las demás. Es el caso de la palabra *marzo* por ejemplo, que en la figura 3.5 se encuentra solitaria en todos los gráficos a partir de (b).

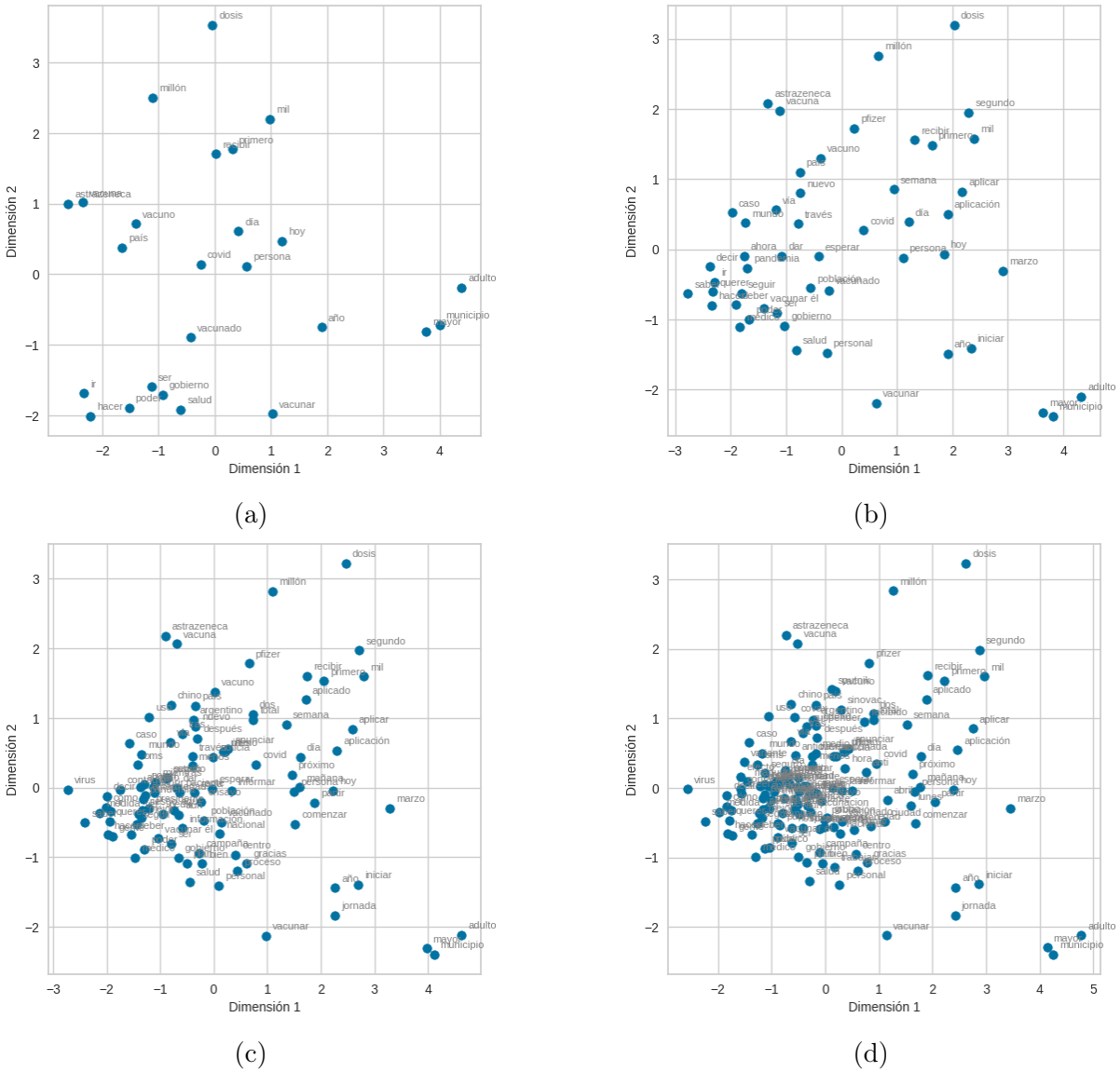


Figura 3.5: Diagrama de dispersión de palabras: (a) Dispersión del top 25 de palabras.(b) Dispersión del top 50 de palabras. (c) Dispersión del top 100 de palabras.(d) Dispersión del top 150 de palabras.

Si bien existe poca literatura sobre la evaluación de los modelos de generación de *Word Embedding* aquellos que abordan el tema concuerdan con la utilización de algunas medidas, como la similitud entre palabras, que correlaciona la distancia entre los vectores de palabras y la similitud semántica percibida por el ser humano. El objetivo es medir qué tan bien la noción de similitud percibida por los humanos es capturada por las representaciones de vectores de palabras y validar la hipótesis distributiva según la cual el significado de las palabras está relacionado con el contexto en el que ocurren(Wang et al., 2019). La medida comúnmente utilizada para esta evaluación es la **similitud del coseno** (Wang et al., 2019) definida como:

$$cos(W_x, W_y) = \frac{W_x \cdot W_y}{||W_x|| \cdot ||W_y||}, \tag{3.17}$$

donde W_x y W_y son dos vectores de palabras. Esta medida calcula la correlación entre todas las dimensiones del vector, independientemente de su relevancia para un par de palabras determinado o para un grupo semántico. La medida es robusta dado la normalización por el tamaño de los vectores y es computacionalmente económico, por lo que, se puede utilizar para un número grande de puntuaciones de un modelo.

La tabla 3.2 muestra el top 10 de palabras que mayor frecuencia tienen en los documentos recopilados y las 3 primeras palabras que mayor valor de similitud tienen con relación a estas. Es importante considerar la coherencia que deben tener con relación a la palabra observada, por ejemplo: la palabra **Vacuna** tiene una mayor proximidad en el espacio vectorial y, por lo tanto, similitud con *vacuno*, *laboratorio* y *aprobado* las cuales se enmarcan en el marco contextual vacuna COVID-19 que se está estudiando.

Palabras	**Top 3**	**Valor de Similitud**
Vacunar	vacunación	0.778
	inmunización	0.613
	inoculación	0.513
Vacuna	vacuno	0.741
	laboratorio	0.569
	aprobado	0.526
País	latino	0.765
	américa	0.732
	eu	0.724

Dosis	trimestre	0.736
	componente	0.721
	biológico	0.700
Vacuno	vacuna	0.741
	oxford	0.564
	estudiar	0.509
Mayor	edad	0.687
	elegible	0.673
	adelante	0.663
Año	edad	0.710
	residente	0.707
	años	0.699
Poder	contagiar	0.707
	sano	0.706
	querido	0.700
Vacunado	muerto	0.639
	inmunizado	0.638
	pasar	0.610
Salud	médico	0.734
	docente	0.686
	educación	0.681

Tabla 3.2: Top de 10 palabras con mayor frecuencia y sus 3 palabras con mayor valor de similitud.

Algunos contextos en los que se puede encontrar estas palabras serían los siguientes:

> *@ «Evalúan reactivar turismo cubano en el mediano plazo con posibilidad de ofrecer una vez aprobada e inmunizada la población local vacunación contra covid a turistas, la vacuna cubana más avanzada aún con ensayos de fase es soberana»*

@«En estados unidos ya nació una bebé con anticuerpos contra covid luego de que su madre recibiera la vacuna del laboratorio moderna»

@ «Este es el calendario de vacunación contra el covid para la próxima semana en nuestro país, revisa más información aquí #yomevacuno »

Otra prueba que se suele utilizar es la **analogía de palabras** que, como se describe en Mikolov et al. (2013c), es un test del tipo *«a es a b lo que c es a ...»* representado de la forma:

$$a : b :: c : d?. \tag{3.18}$$

El algoritmo *Word2Vec* busca los vectores de incrustación x_a, x_b, x_c (todos normalizados a una norma unitaria) y calcula el valor de y que es la representación en un espacio continuo de la palabra que se espera sea la mejor respuesta. Puede que no exista ninguna palabra en esa posición exacta, por lo que, se buscará la palabra cuyo vector de incrustación tenga la mayor similitud de coseno con y como respuesta, lo que se escribiría como:

$$y = x_b - x_a + x_c = cos(x_b x_a + x_c, x_d). \tag{3.19}$$

Se realiza el test para la prueba *«COVID es a enfermedad, lo que vacuna es a...»*: $enfermedad : covid :: vacuna :?$. La figura 3.6 muestra el resultado de las palabras que son análogas a *vacuna*, según el valor de similitud del coseno. En este caso por primera respuesta se tiene *dosis*, lo cual concuerda con el marco contextual en que se trabaja, que es la vacunación contra el COVID-19[4].

[4]Durante el proceso de vacunación se concretó que se requería más de una sola aplicación de la vacuna, por lo que, se inició a utilizar la palabra dosis como analogía de vacuna COVID-19 diferenciando si era la primera, segunda o hasta cuarta según laboratorio y disponibilidad.

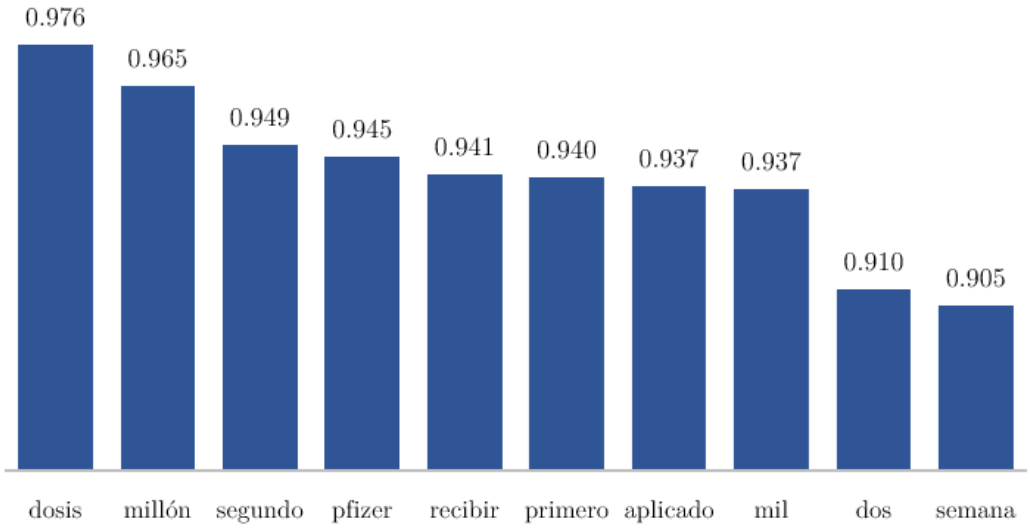


Figura 3.6: Palabras análogas según su valor de similitud del coseno.

Se afirma que muchos modelos obtienen menos del 30 % de las relaciones reales entre palabras con esta pruebas de analogía, lo que sugiere que no todas las relaciones pueden ser identificadas de esta manera. En particular, las relaciones semánticas léxicas como la sinonimia y antonimia son las más difíciles (Wang et al., 2019). También la precisión de esta prueba disminuye a medida que aumenta la distancia entre las palabras. Otro inconveniente es la subjetividad, pues las analogías son principalmente procesos del razonamiento y la lógica humanos, lo que por demás pudiera fluctuar fruto de los contextos de grupos poblacionales. Sin embargo, son las pruebas más utilizadas en el análisis de *Word Embedding* y suelen ser incorporadas en su evaluación.

Capítulo 4

Análisis de Componentes Principales y clúster

Siguiendo la metodología de Teng and Khong (2022), en este estudio se propone realizar un análisis de clúster para obtener los grupos de palabras del corpus que poseen ciertas similitudes y se agrupan dentro de un mismo grupo, tal que se podría determinar discursos en ellos y categorizar el tema central de estos. Previo al análisis de clúster se ejecutó un Análisis de Componentes Principales (PCA).

4.1. Análisis de Componentes Principales

En *Word2vec* se toma como entrada una serie de corpus de texto y se los representa en un espacio vectorial normalmente de un número alto de dimensiones, asignando cada palabra única en el corpus a un vector correspondiente en el espacio.

El uso de un número alto de dimensiones ($D > 100$ por ejemplo) puede permitir que los vectores capturen relaciones más complejas entre las palabras y, potencialmente, representen mejor las características semánticas y sintácticas de las palabras. Sin embargo, también aumenta la complejidad computacional y puede requerir más datos de entrenamiento. Esto es relevante de cara a la aplicación de un análisis de clúster.

El análisis de componentes principales (PCA) es la técnica más antigua y conocida de análisis de datos multivariados. Fue acuñada por primera vez por Karl Pearson en 1901 y desarrollada de manera independiente por Harold Hotelling en 1933. Es una técnica que utiliza principios matemáticos subyacentes sofisticados para transformar un número posiblemente correlacionado de variables en un número menor de variables llamadas componentes principales (Mishra et al., 2017) y quizás su uso más común es como el primer paso para analizar conjuntos de datos grandes.

Una de las principales aplicaciones de esta técnica es reducir la dimensionalidad de un conjunto de datos en el que hay un gran número de variables interrelacionadas, al tiempo que se retiene la mayor parte de la variación presente en el conjunto de datos. Esta reducción se logra mediante la transformación en un nuevo conjunto de variables, denominado componentes principales, que no están correlacionados y que están ordenados, de manera que los primeros retengan la mayor parte de la variación presente en todas las variables originales.

En esencia, PCA busca encontrar las direcciones en las cuales los datos más varían y proyecta los datos originales en esas direcciones principales. Se supone una distribución de nuestros datos con 3 variables, geométricamente PCA se vería como se muestra en la figura . El punto morado representa la media del conjunto de observaciones. El vector que define la primera componente principal Z_1 (línea naranja) sigue la dirección en la que las observaciones varían más. La proyección de cada observación sobre esa dirección equivale al valor de la primera componente para dicha observación o puntuación.

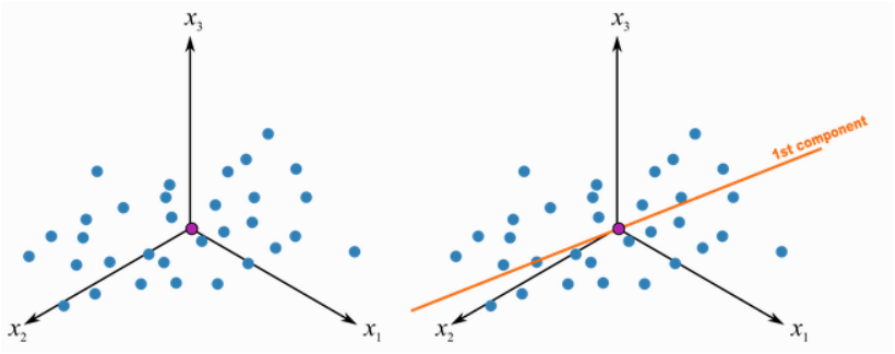


Figura 4.1: Primer componente en una distribución con 3 variables (Gil, 2018).

Las puntuaciones miden la distancia desde el origen a cada proyección, teniendo cada observación x_i su propia puntuación para cada componente m. Desde el punto de vista geométrico, las puntuaciones se calculan teniendo en cuenta la ecuación para obtener el

coseno de un ángulo en un triángulo rectángulo con la siguiente fórmula:

$$cos(\theta) = \frac{L_{i,m}}{||x_i||} = \frac{x_i' p_m}{||x_i||\ ||p_m||}. \tag{4.1}$$

donde $L_{i,m}$ representa la longitud de la proyección de la observación x_i sobre la componente principal m. x_i' es el valor proyectado de la observación x_i sobre la componente principal m. p_m es el vector de la componente principal m. $||x_i||$ y $||p_m||$ se refieren a la norma (longitud) del vector de observación x_i y es la norma (longitud) del vector de la componente principal m respectivamente.

Con $x_i' p_m$, igual a $(1 \times 1) = (1 \times p)(p \times 1)$, es decir, cada puntuación es una combinación lineal del valor original de su observación x_i y el vector de carga.

Esta componente Z_1 estará representada por un vector de cargas $(p \times 1)$ con inicio en el origen, que indica la dirección de la línea y se obtiene por combinación lineal de las variables originales. Se pueden entender como nuevas variables obtenidas al combinar de una determinada forma las variables originales. La primera componente principal de un grupo de variables $(X_1, X_2, .., X_p)$ es la combinación lineal normalizada de dichas variables que tiene mayor varianza (Amat Rodriguez, 2017):

$$Z_1 = \phi_{1,1} X_1 + \phi_{2,1} X_2 + ... + \phi_{p,1} X_p \tag{4.2}$$

donde los términos $\phi_{1,1}, \phi_{2,1}, .., \phi_{p,1}$ son los que definen a la componente. $\phi_{1,1}$ es la carga de la variable X_1 en la primera componente principal. Las cargas pueden interpretarse como el peso/importancia que tiene cada variable en cada componente y, por lo tanto, ayudan a conocer qué tipo de información recoge cada una de las componentes (Gil, 2018).

Que la combinación lineal sea normalizada implica que:

$$\sum_{j=1}^{p} \phi_{j,1}^2 = 1. \tag{4.3}$$

La segunda componente, Z_2 (ver en figura 4.2), sigue la segunda dirección en la que los datos muestran mayor varianza y que no está correlacionada con la primera componente.

La condición de no correlación entre componentes principales equivale a decir que sus direcciones son perpendiculares.

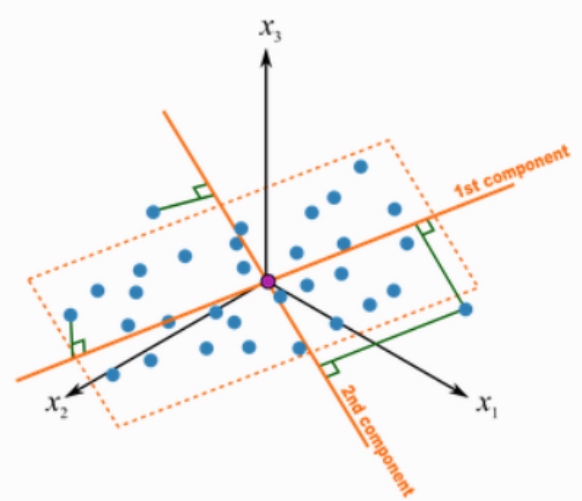


Figura 4.2: Primera y segunda componente en una distribución con 3 variables (Gil, 2018).

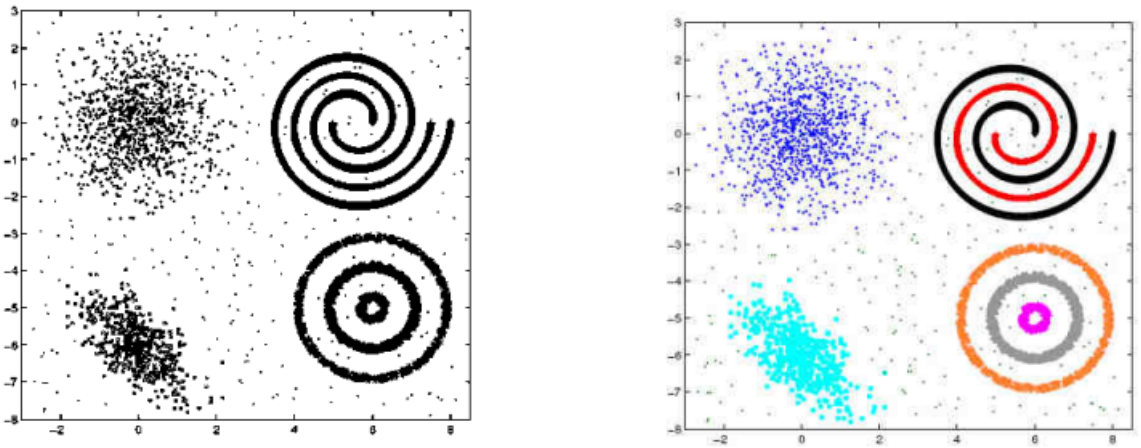
El proceso se repite de forma iterativa hasta calcular todas las posibles componentes $(min(n-1,p))$ con n tamaño de datos y p número de variables. El orden de importancia de las componentes viene dado por la magnitud del autovalor asociado a cada autovector.

4.2. Análisis de Clúster

El desarrollo de la metodología de clúster ha sido un esfuerzo interdisciplinario. Taxónomos, científicos sociales, psicólogos, biólogos, estadísticos, matemáticos, ingenieros, científicos de la computación, investigadores médicos y otros que recopilan y procesan datos reales han contribuido a la metodología de agrupamiento. La agrupación de datos apareció por primera vez en el título de un artículo de 1954 que trataba datos antropológicos. La agrupación de datos también se conoce como análisis Q, tipología, agrupación y taxonomía, dependiendo del campo en el que se aplique. Desde entonces es vasta la bibliografía desarrollada sobre su uso y aplicación en diferentes áreas. Los algoritmos de agrupamiento se pueden dividir ampliamente en dos grupos: jerárquicos y de partición. Los algoritmos de agrupamiento jerárquicos encuentran de forma recursiva grupos anidados, ya sea en modo aglomerativo comenzando con cada punto de datos en su propio

grupo, se fusionan sucesivamente el par de grupos más similares para formar una jerarquía de grupos) o en modo divisivo (de arriba hacia abajo) (comenzando con todos los puntos de datos en un solo grupo, se divide recursivamente el grupo en grupos más pequeños). Los algoritmos de agrupación de partición encuentran todos los grupos simultáneamente como una partición de los datos y no imponen una estructura jerárquica (Jain, 2010).

La figura 4.3 muestra gráficamente la forma en como se espera la aglomeración con datos aplicando análisis de clúster. La idea general es crear un algoritmo automático que descubra los agrupamientos naturales (figura 4.3 (b)) en los datos reales proporcionados (figura 4.3 (a)). También evidencia la diversidad de clúster que pueden existir, observando como los siete clústeres en (a) denotados por siete colores diferentes en (b) difieren en forma, tamaño y densidad.



(a) (b)

Figura 4.3: Diferentes agrupaciones mostradas de manera teórica como datos reales versus agrupación deseada: (a) Datos reales. (b) Agrupación detectada (Jain, 2010).

De los tipos de algoritmos de clúster existentes, este estudio se centra en los de partición y entre ellos el k-medias. Se supone que se tiene un conjunto de datos $X = \{x_i\}$ $i = 1, 2 \ldots n$, donde x_i representa una observación en el espacio n-dimensional. El análisis de clústeres busca agrupar estas observaciones en k clústeres $C_1, C_2, \ldots, C_k$ de manera que la similitud entre las observaciones dentro de un mismo clúster sea alta y la similitud entre clústeres sea baja.

Dado un conjunto de datos y un número k de clústeres dados, se quiere encontrar una asignación de observaciones a clústeres que minimice la suma del error cuadrado de las

distancias intra-clúster que se denota como:

$$J(C_k) = \sum_{x \in C_k} \|x_i - \mu_k\|^2, \tag{4.4}$$

donde C_k es el k-ésimo clúster, μ_k es el centroide del clúster C_i y $\|\cdot\|$ representa una métrica de distancia, que puede ser elegida en función del conocimiento previo de los datos (euclidiana, manhattan, coseno y otras)[1].

El algoritmo de k-medias, también es conocido como el **algoritmo Lloyd**. Fue propuesto por primera vez por Stuart Lloyd en 1957 y es uno de los algoritmos de partición más reconocidos. Su funcionamiento es bastante sencillo, comienza con una estimación inicial de centros o etiquetas y luego actualiza de manera iterativa las etiquetas y los centros hasta que se alcance la convergencia (Lu and Zhou, 2016).

La creación de los centroides iniciales se puede realizar de manera aleatoria, teniendo en cuenta que los centroides deben estar suficientemente separados entre sí, para que no se tenga todos los elementos a distancias similares de todos los centroides. La idea consiste en establecer a cada punto el centroide más cercano, calculando el valor de los centroides como la media de los datos que tenga asignados, de forma que, en sucesivas iteraciones, los centroides recalculados se aproximarán cada uno a su región de referencia (Colome Abril, 2012).

El proceso se lleva a cabo para minimizar la suma de las distancias al cuadrado entre los puntos de datos y todos los centroides resultante de aplicar la ecuación en (4.4) incluyendo una componente binaria de decisión para la actualización de los centroides que se define en:

$$J(C) = q_{ik} \sum_{i=1}^{m} \sum_{i=1}^{K} \|x_i - \mu_K\|^2, \tag{4.5}$$

[1]En este estudio como métrica se emplea la distancia euclidiana.

donde q_{ik} es una variable binaria que asume un valor en función de una regla de actualización de los valores del centroide, que se define como Dabbura (2018):

$$q_{ik} = \begin{cases} 1 \text{ ; si} & k = \text{argmin}_j \ \|x_i - \mu_j\|^2 \\ 0 \text{ ; si} & \text{otro caso} \end{cases} \tag{4.6}$$

Por lo tanto, en la actualización de los centroides se tiene:

$$\mu_i = \frac{\sum_{i=1}^{m} q_{ik} x_i}{\sum_{i=1}^{m} q_{ik}}. \tag{4.7}$$

¿Qué pasa cuando $q_{ik} = 0$? en este escenario la media o centroide se quedaría con el valor anterior asignado (Mackay, 2003).

Como el objetivo del *clustering* es agrupar objetos similares en el mismo clúster y objetos diferentes ubicarlos en diferentes clústeres, las métricas de validación interna están basadas usualmente en los dos siguientes criterios (Guzmán León, 2013):

- Cohesión: El miembro de cada clúster debe ser lo más cercano posible a los otros miembros del mismo clúster.
- Separación: Los clústeres deben estar ampliamente separados entre ellos. Existen varios enfoques para medir esta distancia entre clúster: distancia entre el miembro más cercano, distancia entre los miembros más distantes o la distancia entre los centroides.

Es así como la selección del valor óptimo de k en algoritmos de *clustering*, como k-medias, es un paso crítico en el proceso de su modelización. Si bien no existe un método único que funcione en todos los casos, existen varias técnicas y enfoques que se pueden utilizar para determinar el valor adecuado de k. Como elemento relevante para evaluación

del valor del k óptimo, se define el error SSE, que es la suma de las distancias al cuadrado (*Sum of Square Error*, SSE) dentro de cada clúster, cuya forma de cálculo es:

$$SSE = \sum_{i=1}^{k} \sum_{j=1}^{n_i} d(x_{ij}, c_i)^2,$$

donde k es el número de clústeres, n_i es el número de puntos en el clúster i, x_{ij} es el punto j en el clúster i,μ_i y μ_j son los centroides de los clústeres i y j, $d(\mu_i, \mu_j)$ es la distancia (euclidiana) entre los centroides μ_i y μ_j. El SSE se vuelve progresivamente más pequeño con las iteraciones de agrupamiento hasta que se estabiliza.

En este caso no se estaría usando el SSE directamente, pero su medición es una componente importante en los 6 índices que se han de utilizar como referencia para seleccionar, k los cuales son:

1. **Método del Codo (elbow)**: se utiliza para determinar el número de clústeres k en un conjunto de datos. El método consiste en graficar la variación explicada en función de k y elegir el codo de la curva como el número de clústeres a utilizar (Reyes-Figueroa, 2021). Su fórmula es la siguiente:

$$\text{varianza explicada} = 1 - \frac{SSE}{\sum_{i=1}^{n}(x_i - \bar{x})^2}. \tag{4.8}$$

   Se entiende como varianza explicada la relación entre la varianza intra-grupos contra la varianza total. El punto donde la suma de las distancias cuadradas comienza a aplanarse se llama *«elbow»* (codo). Este punto representa un equilibrio entre la complejidad del modelo y su capacidad para explicar los datos.

2. **Método de la Silueta (Silhouette)**: el índice de silueta mide la calidad de un clúster en función de cuán similar es cada punto de datos con respecto a los demás puntos de su mismo clúster en comparación con los puntos en otros clústeres (Chen et al., 2002). Se calcula como:

$$s(x) = \frac{b(x) - a(x)}{\text{máx}\{a(x), b(x)\}}, \tag{4.9}$$

donde $a(x)$ representa la distancia promedio entre el punto x y todos los demás puntos en el mismo clúster al que pertenece y $b(x)$ es la distancia promedio entre el punto x y todos los puntos en el clúster vecino más cercano al que no pertenece. Cuanto mayor sea esta distancia, mejor separado estará el punto x de los clústeres vecinos.

La silueta varía en un rango de $[-1, +1]$: un valor alto indica que el objeto se corresponde bien con su propio grupo y no con los grupos vecinos.

3. **Índice de Calinski Harabasz, (CH)**: mide la compacidad del clúster calculando la suma de los cuadrados de las distancias entre cualquier punto y el punto central en el mismo clúster. Luego mide el grado de separación, que es la suma de los cuadrados de la distancia entre el centro de cada clúster y los otros centros. La relación entre el grado de separación y la compacidad es el índice CH (Rui, 2021) definido como:

$$CH(k) = \frac{tr(B_k)}{tr(Cov_k)} \times \frac{(N-k)}{(k-1)}, \tag{4.10}$$

donde N es el número de todos los puntos, k es el número de clústeres, B_k es la matriz de covarianza entre clústeres, Cov_k es la matriz de covarianza del mismo clúster, y tr es la traza de la matriz. Cuanto mayor sea el índice CH, mejor será el rendimiento del agrupamiento.

4. **Índice de Davies Bouldin, (DB)**: mide la similitud promedio entre cada clúster y su clúster más similar. Un valor más bajo indica una mejor calidad de agrupamiento (Rui, 2021). El índice se calcula como:

$$DB = \frac{1}{N} \sum_{i=1}^{N} \text{máx} \left(\frac{\bar{S}_i + \bar{S}_j}{||\mu_i - \mu_j||_2} \right)_{j \neq i}, \tag{4.11}$$

donde N es el número de clústeres, S_i es la distancia promedio entre cada punto en el clúster i y el centroide del clúster, S_j es la distancia promedio entre cada punto del clúster j y el centroide del clúster, y $||(\mu_i, \mu_j)||_2$ mide la distancia entre los centroides de los 2 clústeres.

5. **Índice de Hartigan, (IH)**: compara la inercia intraclúster con la inercia interclúster. El k óptimo se elige cuando ya no hay una mejora significativa en la inercia intraclúster (Guzmán León, 2013). Y se calcula como:

$$IH = log \left(\frac{SSE}{SSB} \right), \tag{4.12}$$

donde SSB, representa la suma de las distancias al cuadrado entre los centroides de los clústeres:

$$SSB = \sum_{j=1}^{k} n_j \quad (\mu_j, \bar{x})^2.$$

donde k el número de clústeres, n_j el número de elementos en el clúster j, c_j el centroide del clúster j y $\bar{x}$ es la media del los datos en el dataset, por lo que, $d(\mu_j, \bar{x})$ representa la distancia entre ambos puntos.

El valor de k que resulta en el índice de Hartigan más alto se considera el óptimo. Un índice de Hartigan más alto indica una mejor calidad del *clustering*.

6. **Índice de Dunn, (ID)**: es una medida de la calidad de una partición de un conjunto de datos. Consiste en medir la relación entre la distancia máxima que separa dos elementos clasificados juntos y la distancia mínima que separa dos elementos clasificados por separado. Valores más altos indican una mejor calidad de agrupamiento (Reyes-Figueroa, 2021).

   El índice se calcula de la siguiente forma:

$$ID = \frac{\text{mín}_{1 \leq i < j \leq k} \ d(\mu_i, \mu_j)}{\text{máx}_{1 \leq j \leq k} \ \nabla j}, \tag{4.13}$$

   donde $d(\mu_i, \mu_j)$ es la función de distancia entre clústeres. Mientras en el denominador ∇j representa el cálculo del diámetro para cada grupo(esto es, la máxima distancia entre dos de sus elementos), y muestra que se debe maximizar la función de distancia intraclúster, que consiste en encontrar el clúster con el diámetro más grande. Este índice no se basa en una distancia en particular, por lo tanto, se puede utilizar en una amplia variedad de situaciones y métricas.

4.3. Resultados de PCA y *clustering*

En este apartado se presentan los resultados de la aplicación de la técnica PCA y el análisis de clúster a los *Word Embedding* generados a partir del modelo *Word2vec*. Antes de entrar en los resultados obtenidos a través del PCA y el análisis de clúster de los *Word Embeddings* generados por el modelo *Word2vec*, es importante aclarar los términos que se utilizarán en esta sección:

- Cuando se hace referencia a *Dimensiones* se estará hablando del tamaño del vector que se utilizó para la representación vectorial de los *Word Embedding* en el modelo *Word2vec* expuesto en la Sección 3.
- Para el caso del PCA se utilizará el término *Componentes* para referirnos al total de componentes utilizados como resultado de la técnica en los datos. Sin embargo, en todos los modelos ejecutados se trabajó sobre la selección de los 2 primeros componentes resultantes.
- Por último, *Escenario* es el término que refiere a la combinación única de parámetros utilizada en cada uno de los modelos de clúster ejecutado.
- Se usará de manera indistinta los términos «documentos» y «*tweets*» para referirnos al conjunto de *tweets* recopilados para este estudio.

En ese sentido, se plantearon un total de tres grandes escenarios, según el número de palabras claves (top de palabras), ramificados cada uno en 2 sub-escenarios, variando el tamaño de la dimensión de vectorización (ver tabla 3.1) terminando con 6 sub-escenarios. Los resultados permitieron elegir el valor óptimo de número de clústeres que elegir para modelar según los datos previamente procesados con PCA. El proceso de creación de estos escenarios se compone de una serie de pasos que se detallan a continuación:

- Aplicación de *Word2Vec* con CBOW y Muestreo Negativo. Para cada uno de los sub-escenarios se aplicó el modelo *Word2Vec*. Se realizaron dos configuraciones para el tamaño de vector de palabras: 50 y 100 dimensiones (D). Esto dio como resultado un total de seis ejecuciones de modelos *Word2Vec*, tres con un tamaño de vector de 50D y otros tres con 100D.
- Cálculo de TF-IDF para la selección de top de palabras. Después de obtener los vectores de palabras en los pasos anteriores, se aplicó la técnica de TF-IDF para calcular la importancia de las palabras en los documentos. Se generaron tres conjuntos de palabras clave utilizando diferentes umbrales de selección: 100, 150 y 200 palabras clave (top de palabras).
- Reducción de dimensionalidad con PCA. En cada uno de los escenarios y sub-escenarios, se aplicó el (PCA) a los vectores de palabras resultantes. El objetivo era reducir la dimensionalidad de los datos de alta dimensionalidad a solo dos dimensiones para permitir una visualización efectiva y una comprensión más profunda de la estructura de los datos.

- Análisis de clúster con k-medias. Primero se ejecutó un algoritmo de k-medias iterativo con diferentes valores de k en un rango de [0-20]. Después, se calcularon los valores de los seis índices introducidos anteriormente y se evaluó cómo variaban estos índices a medida que k aumentaba. La selección del valor óptimo de k se basó en la consistencia de los resultados de estos índices en los diferentes escenarios.

Los resultados para cada índice se pueden visualizar en la figura 4.4. La línea azul representa la evolución del valor de k, la línea punteada negra representa el valor de k óptimo para el modelo de clúster y la línea verde representa la evolución del tiempo medido en segundos de procesamiento para cada modelo. Con relación al tiempo de procesamiento se observa como es progresivo en todos los casos, lo cual es coherente debido a aumento del número de clústeres a calcular, sin embargo, es relativamente bajo en términos de segundos, apenas llegando a los 0.09 segundos como el máximo de tiempo relativo cuando se calcula el método del Elbow.

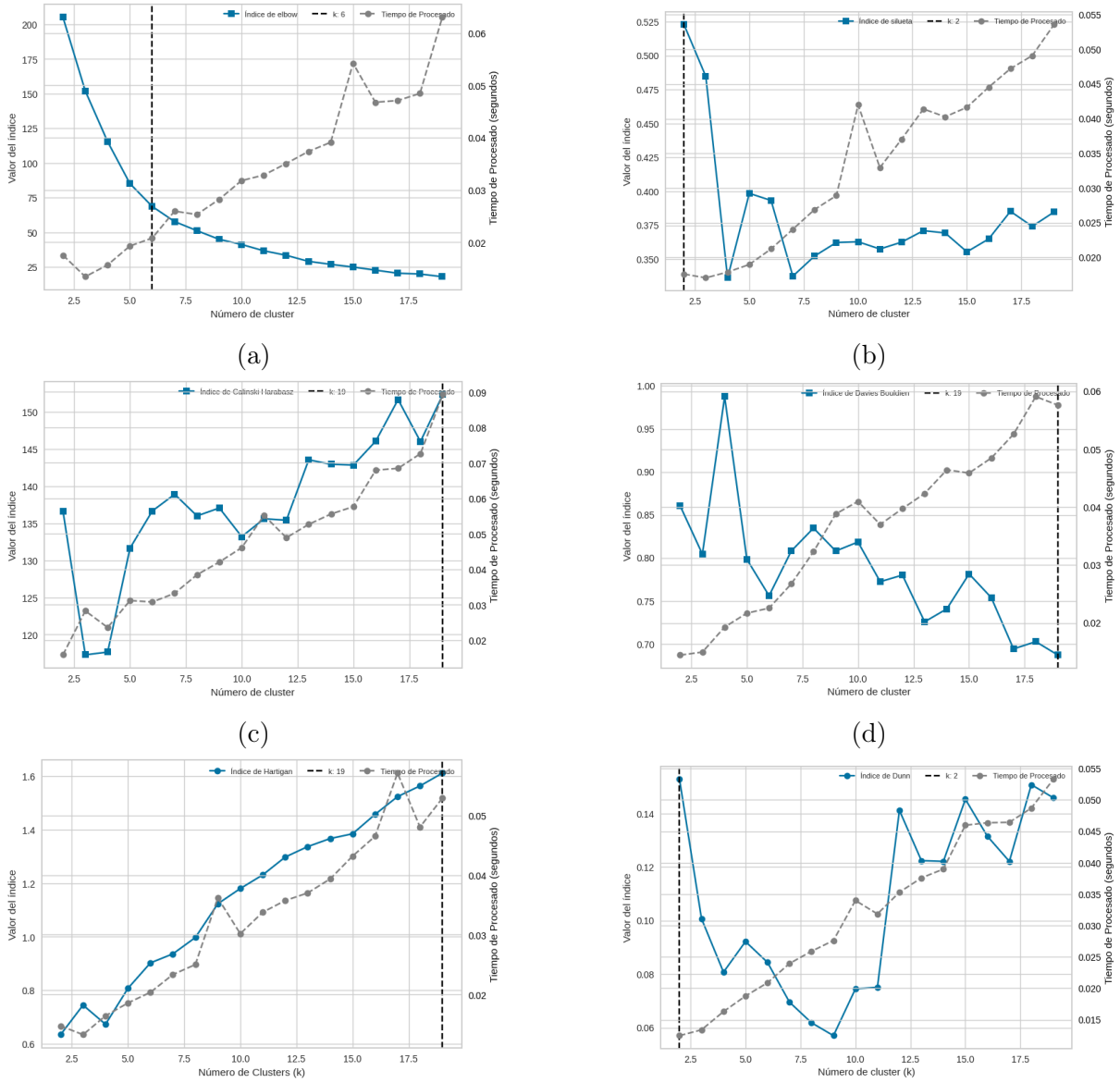


Figura 4.4: Resultados gráficos del cálculo de cada uno de los 6 índices para determinar el k óptimo: (a) Método del Elbow.(b) Índice de la silueta. (c) Índice de Calinski Harabasz. (d) Índice de Davies Bouldin. (e) Índice de Hartigan. (f) Índice de Dunn.

En la tabla 4.1, se resumen los resultados del proceso de selección del valor óptimo de k para diferentes configuraciones de palabras clave (Top de Palabras, TP) y dimensiones de vectores de palabras (D). Se utilizaron los índices introducidos en la Sección 4.2, para

evaluar la calidad de los clústeres. En cada celda de la tabla, se presenta un escenario y se indica el valor de k que se consideró óptimo según el índice correspondiente y la configuración específica de palabras clave y dimensiones de vectores de palabras.

En resumen, los resultados muestran:

- Top Palabras 200: para dimensiones de vectores de palabras de 100 y 50, los índices «Elbow» y «Silueta» sugieren un valor de k igual a 7. Mientras, los índices «Davies Bouldin» y «Hartigan» respaldan k igual a 18 para ambas dimensiones. En general, en este escenario los índices arrojan igual resultado dado cualquier valor de D(50,100).
- Top Palabras 100: para dimensiones de vectores de palabras de 100, el índice «Elbow» sugiere un valor de k igual a 6. Los demás índices presentan valores diferentes de k. Para dimensiones de vectores de palabras de 50, los índices, los índices «Calinski Harabasz», «Davies Bouldin», «Hartigan» y coinciden en $k = 19$.
- Top Palabras 150: para dimensiones de vectores de palabras de 100, los índices «Dunn» y «Silueta» sugieren $k = 2$. En ambas dimensiones los índices «Calinski Harabasz», «Davies Bouldin» y «Hartigan» coinciden en $k = 19$.

Top Palabras	**Dimensiones**	**Elbow**	**Silueta**	**CH**	**DB**	**H**	**ID**
200	100	7	2	15	18	18	3
	50	7	2	15	18	18	6
100	100	6	2	19	16	18	3
	50	6	2	19	19	19	3
150	100	6	2	19	19	19	2
	50	7	2	19	19	19	9

Tabla 4.1: Valor de k óptimo según indicador usando k-medias.

La figura 4.5 muestra la varianza explicada para los dos primeros componentes según los escenarios planteados, la elección de la cantidad de componentes principales a seleccionar surge del equilibrio entre reducir la dimensionalidad de los datos y mantener suficiente información para realizar análisis o tareas posteriores. En términos del (TP)

no hay tantas diferencias en los porcentajes de varianza explicada de los primeros dos componentes al aplicar el PCA, es en la selección del tamaño del vector de *embedding* donde se encuentran diferencias, siendo que el uso de 100D ofrece el mejor resultado con cualquier top de palabras seleccionadas.

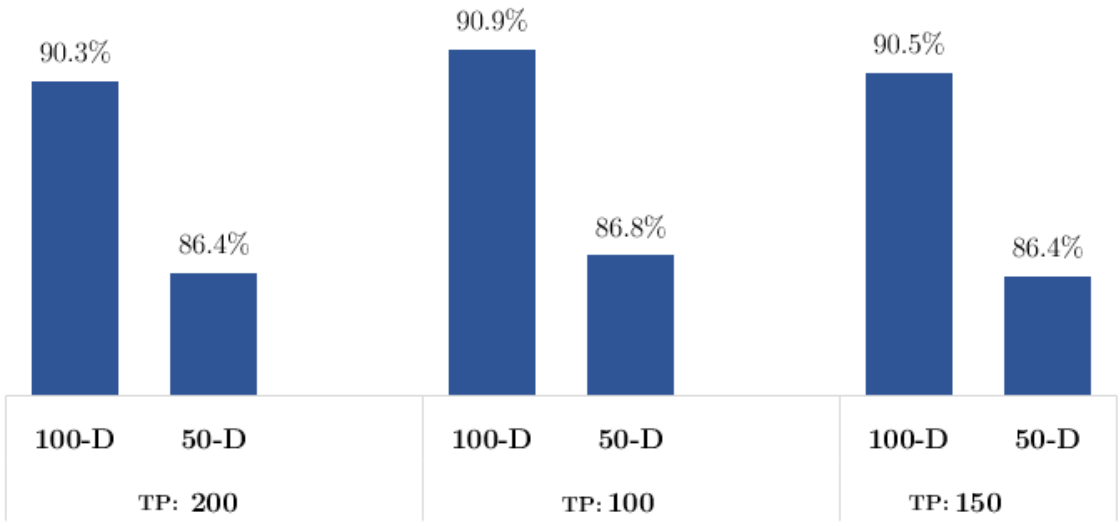


Figura 4.5: Valor de varianza explicada para los dos primeros componentes según escenarios.

Para la selección del escenario se tiene en cuenta la consistencia en el valor de k resultante en el set de índices calculados, el porcentaje de varianza explicada que ofrece los primeros dos componentes principales, así como la coherencia de los términos agrupados en los clústeres resultantes con relación al contexto de análisis utilizado que en este caso es la adaptación de los 4 conceptos del modelo HBM. Siguiendo la tabla 4.1 el valor de k que proporcionó resultados consistentes en términos de los seis índices se consideró como el valor óptimo de k para ese escenario en particular. Esta metodología permitió una selección robusta de k, que fue relevante para la interpretación y análisis adecuados de los resultados de los clústeres en cada escenario. Al elegir el valor de k de esta manera, se garantizó que los clústeres resultantes fueran representativos y coherentes en función de múltiples criterios de evaluación, lo que fundamenta la calidad de los resultados de la investigación.

Basado en la consistencia en el valor de k que proporciona cada índice, se tienen como potenciales escenarios la combinación de palabras y dimensiones siguientes: (100,50),

(150,100) y (150,50) las cuales presentan resultados similares para los índices de CH, DB y IH. Otras elecciones como el valor de k que arroja el método de la silueta (que resulta constante en cada escenario) presentaría el inconveniente de no permitir identificar las narrativas establecidas en el modelo creencias de salud, cuyos 4 conceptos resumirían las ideas que subyacen detrás del proceso de vacunación contra el COVID-19.

Basado en los criterios mencionados, el escenario elegido sería el que combina 150 palabras con 100 dimensiones, donde 3 de sus índices son consistente en la elección del k. Adicionalmente, se tiene un buen valor de porcentaje de varianza explicada con solo dos componentes (90.5 %). En cuanto al valor del error (SSE) se confirmó la tendencia expuesta en la teoría de que a medida que se exploraban diferentes valores de k disminuye el error.

La figura 4.6 muestra la disminución progresiva a medida que aumenta el número de clústeres. Esto sugiere que a medida que se aumenta el número de clústeres, los puntos de datos se ajustan de manera más precisa a los centroides de sus respectivos clústeres. Sin embargo, es importante señalar que la curva de SSE tiende a estabilizarse en algún punto, lo que indica que un valor excesivamente alto de k puede llevar a una sobre-agrupación. Por lo tanto, la elección del número óptimo de clústeres debe basarse en un equilibrio entre la reducción de SSE y la capacidad de los clústeres para capturar patrones significativos en los datos. En ese sentido, la curva comienza a aplanarse de manera horizontal, sugiriendo que para k mayores a 19 se podría tener esta sobre-agrupación a costa de una mayor reducción de este indicador. En ese sentido, $k = 19$ resultaría ser un punto óptimo.

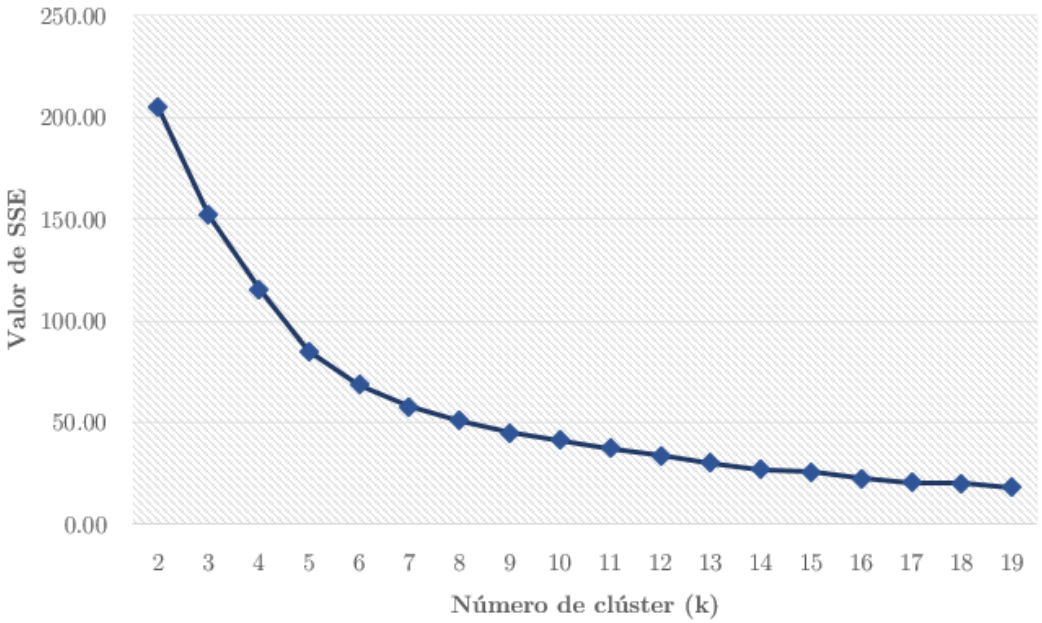


Figura 4.6: Valor de SSE (intra-clúster) según número de clúster.

Al aplicar el algoritmo de k-medias usando $k = 19$ se hace a través del paquete *sklearn* de *Python* que utiliza la distancia euclidiana y con un método de inicialización de centroides denominado como *k-medias++* que selecciona los centroides iniciales del clúster utilizando un muestreo basado en una distribución de probabilidad empírica de la contribución de los puntos a la inercia total, lo cual ayuda a acelerar la convergencia. El algoritmo implementado es el *k-medias++ codicioso* y difiere del algoritmo estándar al realizar varios intentos en cada paso de muestreo y elegir el mejor centroide entre ellos (Pedregosa et al., 2011b).

El resultado de manera gráfica se presenta en la figura 4.7 en la que se observa como hay grupos que apenas poseen un solo *tokens*. En la tabla 4.2 cada uno de los 19 clústeres representados en la tabla contiene un conjunto de palabras relacionadas semánticamente, donde un valor más alto indica que las palabras dentro del clúster están más cerca entre sí en comparación con otros clústeres. Estos resultados son esenciales para comprender la estructura semántica de los datos y en la identificación de patrones y relaciones significativas entre palabras en el contexto de este estudio.

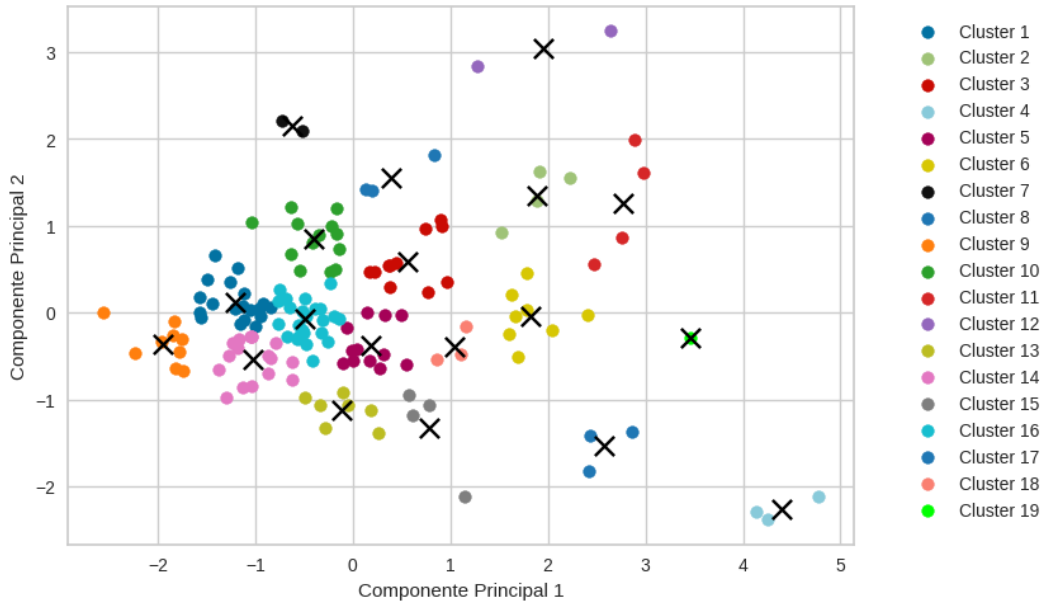


Figura 4.7: Representación de los grupos obtenidos con k-medias para $k = 19$.

El análisis de los resultados del *clustering* muestra patrones interesantes en la distribución de los términos según su longitud y su similitud semántica. Se observa que los clústeres presentan una variación significativa en términos de índices de silueta, lo que sugiere diferencias en la cohesión de los grupos. Destacan especialmente los clústeres 4, 7 y 8, que exhiben índices de silueta más altos, indicando una mayor homogeneidad interna y una separación efectiva.

Por otro lado, el clúster 19 presenta un índice de silueta, sea igual a cero. Un valor de silueta de igual a cero indica que los términos agrupados en este no presentan similitud sustancial. En el caso de este clúster se observa que solo posee un término, «marzo», lo cual es la razón por la que el índice sea 0.

En ese mismo orden, los clústeres 1 y 16 tienen un número significativamente mayor de términos, es decir, son más densos en comparación con los demás. Es probable que haya en estos algunos patrones semánticos específicos que los caracterizan y los agrupan como semejantes.

Clúster	**Términos en el clúster**	Índice de Silueta
1	caso, ahora, querer, médico, campaña, cómo, paciente, uso, próximo, total, hospital, sanitario, seguro, sputnik, creer, vacunada, abril, recibido, anti	0.338
2	astrazeneca, recibir, semana, enfermedad	0.448
3	vacunar, aquí, nacional, tras, bien, mientras, oms, hora, edad, ola, faltar	0.341
4	año, municipio, hacer	0.767
5	salud, población, esperar, presidente, gracias, partir, momento, anticuerpo, grupo, punto, avanzar, mismo	0.335
6	covid, día, hoy, poner, mañana, último, argentino, inmunidad	0.368
7	vacuna, primero	0.801
8	poder, iniciar, jornada	0.670
9	adulto, gobierno, decir, mundo, noticia, contagio, menos, ver, medida	0.510
10	dosis, nuevo, través, saber, centro, virus, informar, aplicado, estudio, fase, clínico, trabajador, pueblo	0.396
11	ser, dar, aplicar, aplicación	0.159
12	vacuno, millón	0.199
13	persona, ir, personal, información, riesgo, mes, trabajar	0.437
14	vacunado, mil, seguir, vacunar él, deber, pandemia, gente, chino, dato, mejor, cada, público, suspender, contar, dejar, lunes, frente	0.328
15	país, proceso, después, comenzar	0.260
16	segundo, vía, plan, dos, anunciar, pedir, morir, según, aún, ensayo, vez, muerte, efecto, mar, variante, medio, buen, pasar, vacunacion, maduro, ministro, acceso, importante, sinovac, atención	0.378
17	mayor, pfizer, covax	0.456
18	ciudad, parte, ninguno	0.543
19	marzo	0.000

Tabla 4.2: Agrupación de términos según el clúster, para $k = 19$.

Capítulo 5

Modelos de Clasificación

Desde siempre, la inteligencia artificial ha sido impulsada por la ambición de comprender y descubrir relaciones complejas en los datos. Es decir, encontrar modelos que no solo puedan generar predicciones precisas, sino también utilizarse para extraer conocimiento de manera comprensible. Guiada por este objetivo doble, la investigación en aprendizaje automático ha dado lugar a extensos cuerpos de trabajo en una miríada de direcciones (Louppe, 2015). Entre estos, los métodos basados en árboles se destacan como uno de los métodos efectivos y útiles, capaces de producir resultados confiables y comprensibles en casi cualquier tipo de datos. Los árboles de decisión son el fundamento de muchos de los estados de artes de los modelos de supervisión automática y entre ellos de los modelos de bosques aleatorios (*Random Forest, RF*).

Los *Random Forest* fueron introducidos por Breiman (2001), quien se inspiró en trabajos previos realizados por Amit and Geman (1997). Aunque no es introducido formalmente como tal hasta 2001, los RF son una extensión de la idea de *bagging* de (Breiman, 1996). Los RF pueden utilizarse tanto para una variable de respuesta categórica, denominada en Breiman (2001) como «clasificación», como para una respuesta continua, conocida como «regresión». Del mismo modo, las variables predictoras pueden ser tanto categóricas como continuas.

Los árboles utilizados en los RF se basan en los árboles de partición binaria recursiva. Estos árboles dividen el espacio de los predictores mediante una secuencia de particiones binarias («divisiones») en variables individuales. El nodo «raíz» del árbol comprende todo el espacio de los predictores. Los nodos que no se dividen se llaman «nodos terminales» y

forman la partición final del espacio de los predictores. Cada nodo no terminal se divide en dos nodos descendientes, uno a la izquierda y otro a la derecha, según el valor de una de las variables predictoras. Para una variable predictora continua, una división se determina mediante un punto de división; los puntos en los que el predictor es menor que el punto de división van hacia la izquierda, el resto va hacia la derecha. Una variable predictora categórica X_i toma valores de un conjunto finito de categorías $S_i = \{s_{i,1}, ..., s_{i,m}\}$. Una división envía un subconjunto de estas categorías $S \subset S_i$ a la izquierda y las categorías restantes a la derecha (ver figura 5.1). La división específica que un árbol utiliza para dividir un nodo en sus dos descendientes se elige considerando todas las posibles divisiones en cada variable predictora y eligiendo la «mejor» según algún criterio (Cutler et al., 2012).

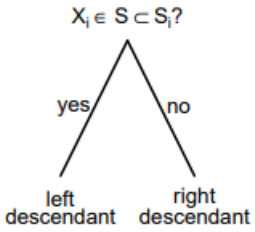


Figura 5.1: Ejemplo de partición binaria de una variable categórica X_i (Cutler et al., 2012).

Dado un conjunto de datos X compuesto por N observaciones que pertenecen a dos clases, y una serie de características, los árboles de decisión operan de la siguiente manera: primero, se selecciona una característica x y un umbral d que dividen a X en dos subconjuntos que son distintos según un criterio especificado, a partir de todas las características de x y todos los posibles valores de d. Luego, el conjunto de entrenamiento se divide en los dos grupos X_L y X_R dependiendo sí $x < d$ o $x >= d$. Este procedimiento se repite con X_L y X_R utilizando otra combinación (x, d). El proceso se realiza tantas veces hasta que no sea posible realizar más divisiones. En un RF, en lugar de entrenar un árbol en la totalidad del conjunto de entrenamiento, se entrena solo en una muestra de m casos seleccionados al azar con reemplazo del conjunto completo de N casos. Las muestras seleccionadas se llaman casos dentro de la bolsa; el resto se dejan de lado como casos fuera de la bolsa. Además, al determinar qué característica dividir en cada nodo, solo se considera un subconjunto de g de las G características (generalmente $g = G^{1/2}$) que son elegibles; este subconjunto se selecciona al azar sin reemplazo de manera independiente para cada nodo a partir del conjunto completo de G características (Amaratunga et al., 2008).

La novedad de los modelos de RF versus, los árboles de decisión, es el proceso de *ensemble* que se refiere a la forma en cómo lograr el equilibrio entre sesgo (árboles muy pequeños) y varianza (árboles muy grandes). En el caso de los RF esté proceso es *bagging* introducido por Leo Breinman, y que es un método para generar múltiples versiones de un predictor y utilizarlos para obtener un predictor agregado. La agregación promedia las versiones al predecir un resultado numérico y realiza una votación plural al predecir una clase. Las múltiples versiones se forman haciendo réplicas de arranque del conjunto de aprendizaje y utilizando estas como nuevos conjuntos de aprendizaje (Breiman, 1996).

A pesar de la sencillez con la que se puede resumir el proceso de construcción de un árbol, es necesario establecer una metodología que permita crear los diferentes subconjuntos de X_L y X_R o lo que es equivalente, decidir dónde se introducen las divisiones, en qué predictores y en qué valores de los mismos. Es en este punto donde se diferencian los algoritmos de árboles de regresión y clasificación (Amat Rodrigo, 2020).

El objetivo del algoritmo es dividir X utilizando la d^* óptima que maximiza la disminución de la impureza, la cual surge como medida de bondad de ajuste del modelo. Cuanto menor sea la impureza en un nodo, mejor se separarán las clases en ese nodo y, por lo tanto, mejor será el rendimiento del árbol de decisión.

Existen varias métricas comunes de impureza utilizadas en RF, como la Entropía y el Índice Gini:

- Entropía: mide la incertidumbre en un nodo. Cuanto menor sea su valor, más puro será el nodo. Se calcula como:

$$E = -\sum_{k=1}^{K} \hat{p}_{mk}\ log(\hat{p}_{mk}). \tag{5.1}$$

  donde $\hat{p}_{mk}$ representa la proporción de observaciones del nodo m que pertenecen a la clase k. Y K es el número de clases en el nodo m.

- Índice de Gini: mide la probabilidad de que un elemento seleccionado al azar sea clasificado incorrectamente si se clasifica de acuerdo con la distribución de probabilidad de las clases en el nodo. Un valor bajo de índice gini indica un nodo puro. Se calcula como:

$$G_m = \sum_{k=1}^{K} \hat{p}_{mk}\ (1 - \hat{p}_{mk}). \tag{5.2}$$

Otro modelo que se estará utilizando son las Máquinas de Soporte Vectorial (SVM) que se fundamentan en el *Maximal Margin Classifier*, que está basado en el concepto de hiperplano óptimo. SVM busca encontrar el hiperplano o la función que separe correctamente dos clases.

Dado un conjunto separable de ejemplos $S = \{(x_1, y_1), ...(x_n, y_n)\}$, donde $x_i \in \mathbb{R}^d$ e $y_i \in \{+1, -1\}$, se puede definir un hiperplano de separación como una función lineal que es capaz de separar dicho conjunto sin error (Suárez, 2014):

$$g(x) = (w_1 x_1 + ... + w_d x_d) + b =< w, x > +b. \tag{5.3}$$

donde w es un vector de peso que determina la orientación del hiperplano en el espacio de características y b representa un umbral que permite ajustar la posición del hiperplano. Es un término constante que se utiliza para trasladar el hiperplano. En resumen (5.3) es la función que evalúa si un punto x pertenece a una de las dos clases.

Gráficamente, el modelo SVM es un modelo de clasificación que mapea las observaciones como puntos en el espacio para que las categorías se dividan por dichos hiperplanos. Luego, las nuevas observaciones se pueden mapear en el espacio para la predicción (Barbona and Beltrán, 2016). El algoritmo SVM encuentra el hiperplano de separación óptimo utilizando un mapeo no lineal a una dimensión suficientemente alta. El hiperplano se define por las observaciones que se encuentran dentro de un margen optimizado por un hiperparámetro a los que se les asigna un coste (error). Estas observaciones se denominan vectores de soporte.

Resulta que el hiperplano que permite separar los puntos no es único, es decir, existen infinitos hiperplanos separables, representados por todos aquellos hiperplanos que son capaces de cumplir las restricciones dadas. Surge entonces la pregunta sobre si es posible establecer algún criterio adicional que permita definir un hiperplano de separación óptimo. Siguiendo la figura 5.2 primero, se define el concepto de margen de un hiperplano de separación, denotado por T, como la mínima distancia entre dicho hiperplano y el ejemplo más cercano de cualquiera de las dos clases (ver figura 5.2 izquierda). A partir de esta definición, un hiperplano de separación se denominará óptimo si su margen es de tamaño máximo (figura 5.2 derecha) donde los puntos dentro del margen (puntos rellenos de color) representan los soportes (Suárez, 2014).

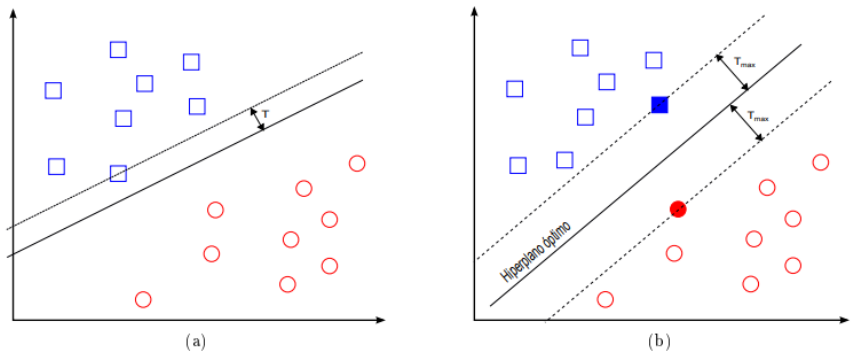


Figura 5.2: Margen de un hiperplano de separación lineal: (a) hiperplano de separación no-óptimo y su margen asociado (no máximo) (b) hiperplano de separación óptimo y su margen asociado (máximo) (Suárez, 2014).

Por lo tanto, dado una serie de puntos, X_i los cuales pertenecen a dos clases linealmente separables ω_1 y ω_2, la distancia de cualquier punto desde el hiperplano es igual a $\frac{|g(c)|}{||w||}$. SVM busca encontrar el valor de w tal que $g(x)$ sea igual a 1 para los puntos de datos más cercanos que pertenecen a la clase ω_1 y -1 para los más cercanos de ω_2. Esto se puede ver como tener un margen de (Awad and Khanna, 2015):

$$\frac{1}{||w||} + \frac{1}{||w||} = \frac{2}{||w||} \tag{5.4}$$

Esto conduce a un problema de optimización en el cual se desea minimizar la función objetivo:

$$\begin{aligned} &J(w) = \frac{1}{2}||w||^2, \\ &\text{s.a} \\ &y_i(< w, x_i > +b) \geq 1, \quad i = 1, \ldots, n. \end{aligned} \tag{5.5}$$

La ecuación (5.6) se refiere a una separación lineal y es lo que se denomina «margen duro» que clasificaría todos los datos de manera casi perfecta.

Cuando los datos no son completamente separables, se introducen ξ_i son un conjunto de variables positivas denominadas variables de holguras ($\xi_i, \quad i = 1, ..., n$) en la función objetivo de la SVM para permitir errores en la clasificación errónea. En este caso, la SVM no busca el margen duro. En cambio, SVM se convierte en un clasificador de margen suave; es decir, el modelo clasifica la mayoría de los datos correctamente, al tiempo que permite que el modelo clasifique erróneamente algunos puntos en las proximidades del límite de separación (Awad and Khanna, 2015). Entonces, la ecuación (5.5) se convierte en:

$$\begin{aligned} &J(w, \xi) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\xi_i, \\ &\text{s.a:} \\ &y_i(< w, x_i > +b) + \xi_i - 1 \geq 0, \quad \xi_i \geq 0, \quad i = 1, ..., n. \end{aligned} \tag{5.6}$$

donde ξ_i son el conjunto de variables de holguras que permite medir el coste asociado al número de ejemplos no separables, C es una constante suficientemente grande que varía según el objetivo de optimización. A medida que se aumenta C, se obtiene un margen más estrecho y se pone más énfasis en minimizar el número de clasificaciones erróneas. A medida que se disminuye C, se permiten más violaciones, ya que maximizar el margen entre las dos clases se convierte en el objetivo (Awad and Khanna, 2015).

El gran inconveniente de esta es la presunción de una separación lineal. SVM hacen esto de una manera específica, utilizando kernels, los cuales se pueden definir de la siguiente manera (Suárez, 2014):

$$K(x, x') =< \Phi(x)\Phi(x') >= (\phi_1(x), \phi_1(x'), ..., \phi_m(x)\phi_m(x') \tag{5.7}$$

donde $\Phi : \mathbb{X} \to \mathcal{F}$ es la función de transformación que hace corresponder cada vector de entrada x con un punto del espacio de características tal que $\phi(x)$ es una función no lineal.

Y según la separación de las clases se puede elegir entre una función lineal, polinómica, radial y otras. La solución tiene la interesante propiedad de que solo las observaciones sobre o dentro del margen afectan al hiperplano. Estas observaciones se conocen como vectores de soporte (Rodrigo Amat, 2017).

Si existe una relación no lineal entre las características y las clases, pero esta relación no es extremadamente compleja o de alto grado, entonces el kernel polinomial podría ser una buena elección, ya que puede capturar esa relación de manera efectiva sin ser demasiado complejo. Sin embargo, cuando la relación no lineal es altamente compleja, el kernel polinomial de bajo grado puede no ser suficiente y podría requerir un kernel de grado más alto o incluso el uso de un kernel radial. En este caso se usará la función polinomial fundamentada en los resultados de los modelos testeados iterando el tipo de kernel, concluyendo que el polinomial provee una separación mejor que el kernel radial y se descarta la lineal porque los datos no tienen separación perfecta que se pueda modelar en un hiperplano. En ese orden, el kernel quedaría como:

$$K(x, x') = (x \cdot x' + C)^d. \tag{5.8}$$

donde C representa el costo de ajustar cada punto a la región que le corresponde, mientras d es un parámetro que especifica el grado del polinomio que se utilizará en la función de kernel polinomial.

Se entrena un modelo usando la función *poly* en *Python* en el se intenta maximizar el ancho del margen entre clases mediante un límite de clase polinómica, la función permite tunear 2 hiperparámetros que serían C y d.

5.1. Resultados del modelo con *Random Forest*

Primero se expone las reglas de clasificación que están sustentadas en los conceptos del modelo, HBM aunque se incluyen otros que fueron identificados en la lectura de los documentos.

Comúnmente la metodología seguida en la literatura disponible requiere de un equipo de investigadores, los cuales realizan diferentes tareas en el proceso de revisión y etiquetado, de forma que se pueda reducir la subjetividad en la lectura de los documentos, pero también se utilizan las revisiones de fuentes y literatura donde se aplicaron los constructos a temas de salud especialmente sobre las vacunas.

Para este estudio también se siguió una metodología de etiquetado manual y revisión

de documentación disponible sobre estudios similares (ver Sección 1.1). En ese sentido, siguiendo con los resultados del análisis de PCA y clúster presentados en la metodología tabla 4.2. La metodología seguida para el proceso de etiquetado y los modelos de clasificación se describe en los pasos siguientes:

1. Se seleccionaron aleatoriamente 5,000 *tweets* de la base de datos completa, una vez pre-procesada (ver Sección 2).
2. Se utilizaron los 3 primeros términos de cada clúster para seleccionar una muestra del 30 % de los *tweets* en los que aparecieron alguno de estos.
3. Después, de forma manual se revisaron las narrativas de los *tweets* dentro del 30 % seleccionado que contenían cada clúster y se etiquetaron estos *tweets* en función de los 4 conceptos del HBM. En el proceso se identificaron 4 etiquetas adicionales.
4. Finalmente, se identificaron un total de 3,602 documentos (*tweets*) aquellos que compartieran relevancia con los clústeres que reflejaban una de las narrativas estudiadas.

En lo que se refiere al punto dos, la tabla 5.1 muestra el top 3 de los términos de cada clúster, sus frecuencias y la proporción de frecuencia que ocupan del total de frecuencias en cada clúster. Esto con la finalidad de validar que los mismos representan una proporción significativa de la narrativa que se capturó al revisarlos.

ID	Top 3 de términos	Conteo para top 3	Conteo Total del clúster	Proporción (%) del top 3
1	caso, ahora, querer	972	3606	26.96
2	astrazeneca, recibir, semana	1921	2089	91.96
3	vacunar, aquí, nacional	4728	5981	79.05
4	año, municipio, hacer	2316	2316	100.00
5	salud, población, esperar	1555	2978	52.22
6	covid, día, hoy	11466	12300	93.22
7	vacuna, primero	6866	6866	100.00
8	poder, iniciar, jornada	1505	1505	100.00
9	adulto, gobierno, decir	1809	3172	57.03
10	dosis, nuevo, través	2624	4464	58.78
11	ser, dar, aplicar	1401	1701	82.36
12	vacuno, millón	2041	2041	100.00
13	persona, ir, personal	1768	2600	68.00
14	vacunado, mil, seguir	1994	5046	39.52
15	país, proceso, después	2762	2966	93.12
16	segundo, vía, plan	984	4374	22.50
17	mayor, pfizer, covax	1513	1513	100.00
18	ciudad, parte, ninguno	410	410	100.00
19	marzo	370	370	100.00

Tabla 5.1: Conteo y proporción del top 3 de los términos según clúster.

Finalmente, se obtienen los constructos con el clúster que le corresponde. Estos constructos adaptados al marco contextual provienen de la lectura de dos estudios de los expuestos en la Sección 1.1 presentados por Teng and Khong (2022) y Wang et al. (2021).

ID clúster	Top 3 términos	Etiqueta	Constructos
1	caso, ahora, querer	Gravedad Percibida (Gravedad)	Narrativa alrededor de la veracidad de que con la aplicación de la vacuna se terminaría la pandemia/contagio del virus.
2	astrazeneca, recibir, semana	Barreras Percibida (Barreras)	Narrativa donde se expone preocupación por los efectos secundarios de la vacuna.
10	dosis, nuevo, través	Beneficios Percibido (Beneficios)	Narrativa positiva sobre la vacuna y las dosis necesarias, incluyendo motivación para aplicarse más de una dosis.
9	adulto, gobierno, decir	Vulnerabilidad	Narrativa sobre el riesgo que padecen los adultos mayores ante la enfermedad y por ende la necesidad de disponer de vacunas para estos como población vulnerable.
16	segundo, vía, plan	Susceptibilidad Percibida (Susceptibilidad)	Narrativa sobre el riesgo de contagio expresado en reclamo sobre el costo de oportunidad de asistir a vacunarse versus solo cuidarse en casa.
8	poder, iniciar, jornada	Jornadas de Vacunación (Jornada)	Narrativa en la que se expresa confianza/desconfianza sobre el plan de vacunación de su país/región.
13	persona, ir, personal	Personal Médico	Narrativa sobre la disposición de vacunas para el personal médico como primera línea en la lucha contra la enfermedad.
5	salud, población, esperar	Riesgo Sanitario	Narrativa que gira en torno a la visualización del virus como un tema de salud pública nacional, expresado en reclamos de la atención y gestión sanitaria disponible.

Tabla 5.2: Etiquetas y sus narrativas para los clústeres seleccionados.

La potencia de estos modelos frente a los modelos *bagging* «tradicionales» reside en la elección aleatoria de los predictores, lo que permite disminuir la posibilidad de que variables con alta correlación se repitan en la iteración de los modelos, haciéndolos relativamente similares.

Para proceder con la modelación se realizó una conversión de la variable que contenía las etiquetas para generar así 8 bases de datos en las cuales se generó una variable como *dummy* con los valores [1,0] donde *1=Positivo y 0= Negativo*. Esta categorización se hizo de acuerdo a la regla de que *"son positivos los casos donde el documento está relacionado con la etiqueta en cuestión y negativo todo lo demás"*. Previo a la modelización se realizó la división de la base de datos en 70 % de los datos para el proceso de entrenamiento de los modelos y 30 % para el testeo de prueba de ajuste. Esta división es fundamental en el desarrollo de modelos de clasificación, ya que permite evaluar la capacidad de genera-

lización de los modelos. El conjunto de entrenamiento se utiliza para ajustar el modelo, mientras que el conjunto de prueba se utiliza para evaluar su rendimiento en datos no vistos. Esta separación ayuda a identificar posibles problemas de sobreajuste y garantiza que el modelo sea capaz de hacer predicciones precisas en nuevas observaciones.

Se entrenó un modelo RF usando diferentes parámetros para cada uno de los constructos identificados. Uno de los atributos más importante de este proceso es poder ejecutar una técnica de *resampling* mediante validación cruzada o *boostrapping* para poder modelar múltiples modelos y evaluarlos sin necesidad de usar los datos de prueba. Esto permite evaluar cada modelo internamente solo con los datos de entrenamiento, lo que además permite calcular indicadores como el error fuera de muestra (*out of bag, OBB*).

Otro aspecto es poder tunear el modelo mediante sus hiperparámetros, esto es, dejándole un intervalo de valores aleatoriamente en cada uno de estos y para cada modelo iterativo que realiza. Para el proceso de tunear se selecciona el argumento *ResearchGrid* de la función utilizada en *Python* que recibe un vector de valores o caracteres que indican los valores que seleccionar los hiperparámetros.

En un primer modelo se tunearon y/o se seleccionaron los parámetros en la tabla 5.3.

Parámetro	**Rango/valores**
El número de árboles en el bosque	[100,300,500]
La profundidad máxima de los árboles en el bosque	[20,25,30,35]
El número mínimo de muestras requeridas para dividir un nodo interno	2
El número mínimo de muestras requeridas en un nodo hoja	2

Tabla 5.3: Parámetros tuneados en el modelo *Random Forest*.

Para el proceso de construcción de los árboles se utiliza el método de *Bootsraping* mediante el cual se crean un gran número de muestras con reposición de los datos observados. Esto significa que se seleccionan muestras del conjunto de datos original de manera aleatoria, pero se permite que una misma muestra pueda ser seleccionada varias veces y que algunas muestras puedan no ser seleccionadas en absoluto (Ledesma, 2008). Esto crea conjuntos de datos de entrenamiento «bootstrap» que son similares pero ligeramente diferentes entre sí. Esto significa que cada árbol se ajusta a una versión ligeramente diferente de los datos originales, lo cual mejora el rendimiento del modelo.

También se combinan estos hiperparámetros del funcionamiento interno del RF con

la técnica de *Cross validation* o validación cruzada que permite evaluar el modelo RF. La idea de la validación cruzada es dividir el conjunto de datos en un conjunto de entrenamiento y un conjunto de prueba. El conjunto de prueba se puede utilizar para probar el modelo que se construyó en el conjunto de entrenamiento. Esto se hace repetidamente para un (posiblemente grande) número de divisiones de los datos. Los resultados de los procedimientos de prueba se combinan de manera adecuada y se utilizan para la selección y evaluación subsiguiente del modelo (Friedl and Stampfer, 2006). Esta técnica se aplica con la finalidad de estimar la precisión y la generalización del modelo *Random Forest* en términos de su capacidad de generalización en datos no vistos.

Como regla de partición de un árbol se usa el índice de gini. Que es el criterio que tiene por defecto el paquete utilizado en *Python*. También se seleccionaron valores fijos para los dos últimos parámetros debido a que para cualquier otro valor el resultado de clasificación empeoraba y de esta forma se puede reducir el tiempo de procesamiento de cada uno de los modelos al tener menos combinaciones y, por lo tanto, iteraciones.

Basándose en los vectores de valores proporcionados como hiperparámetros, el entrenamiento del modelo hace posible la selección de los hiperparámetros que mejor rendimiento arrojan usando como indicador la *Exactitud* que mide la eficacia general de un clasificador.

Usando estos «mejores hiperparámetros» se realiza el modelo de clasificación de RF y se obtienen las matrices de confusión que sirven para evaluar el rendimiento de un modelo de aprendizaje automático. La matriz muestra la relación entre las predicciones del modelo y las verdaderas etiquetas o clases de un conjunto de datos de prueba.

La figura 5.3, muestra de manera teórica los elementos de una matriz de confusión. Esta matriz es una herramienta esencial en problemas de clasificación binaria, ya que permite evaluar el rendimiento de un modelo al comparar las predicciones con los valores reales. En esta matriz, los elementos diagonales representan las predicciones correctas (verdaderos negativos y verdaderos positivos), mientras que los elementos fuera de la diagonal indican los errores de predicción (falsos positivos y falsos negativos). Esta información es fundamental para medir la precisión y la capacidad de un modelo para distinguir entre las clases de interés.

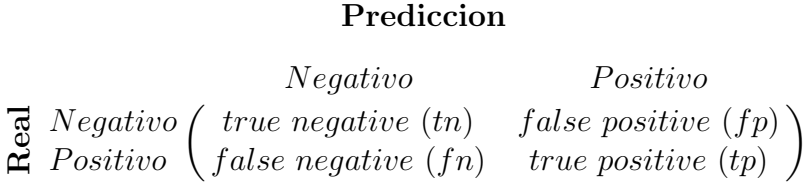


Figura 5.3: Ejemplo de matriz de confusión para clasificación binaria usada en este estudio.

Como parte de la medición de la efectividad del modelo de clasificación se utilizan una serie de indicadores que miden ciertos aspectos en relación con el resultado de la matriz de confusión. La tabla 5.4 muestra estos indicadores, entre los que se encuentra la *Especificidad*, que mide la eficacia del clasificador con relación a los casos etiquetados como negativos. El último indicador se refiere al Área Bajo la Curva ROC (*Area Under of Curve*, AUC) que evalúa que tan bien un modelo puede distinguir entre las clases positivas y negativas. La métrica se deriva de la Curva ROC (*Receiver Operating Characteristic Curve*). En general se utiliza como una representación gráfica del rendimiento del modelo a diferentes umbrales de clasificación. La Curva ROC muestra la tasa de verdaderos positivos en el eje vertical y la tasa de falsos positivos en el eje horizontal a medida que se ajusta el umbral de clasificación. El AUC mide el área bajo esta curva y su valor varía entre 0 y 1.

Indicador:	**Formulación:**	**Evalua:**
Exactitud	$\frac{tp+tn}{tp+fn+fp+tn}$	Eficacia general de un clasificador.
Precisión	$\frac{tp}{tp+fp}$	Concordancia de clase de las etiquetas de datos con las etiquetas positivas dadas por el clasificador.
Sensibilidad	$\frac{tp}{tp+fn}$	Efectividad de un clasificador para identificar etiquetas positivas.
F1-score	$\frac{(\beta^2+1)tp}{(\beta^2+1)tp+\beta^2 fn+fp}$	Relaciones entre las etiquetas positivas de los datos y las dadas por un clasificador.
Especificidad	$\frac{tn}{fp+tn}$	Con qué eficacia un clasificador identifica etiquetas negativas.
AUC	$\frac{1}{2}\left(\frac{tp+tn}{tp+fn}+\frac{tn}{tn+fp}\right)$	La capacidad del clasificador para evitar clasificaciones falsas.

Tabla 5.4: Indicadores de evaluación para clasificación binaria usados en este estudio, (Sokolova and Lapalme, 2009).

Los resultados presentados en la tabla 5.5 son las matrices de confusión de los 8 modelos de clasificación binaria, cuando se aplica el modelo a los datos de prueba. Esto resulta en un error promedio por clase de 2 % para los casos etiquetados como **Negativos** y de alrededor de un 80 % para los casos etiquetados como **Positivos**. Es decir, el modelo tiene baja *Especificidad*, lo que se traduce en poca potencia para clasificar correctamente a los casos Positivos calculada como la proporción de casos tp o tn en relación con los casos totales en cada clase.

Gravedad	*Negativo*	*Positivo*
Negativo	935	42
Positivo	94	10

Suceptibilidad	*Negativo*	*Positivo*
Negativo	970	50
Positivo	57	4

Barrera	*Negativo*	*Positivo*
Negativo	726	140
Positivo	141	74

Jornada	*Negativo*	*Positivo*
Negativo	880	91
Positivo	89	21

Beneficio	*Negativo*	*Positivo*
Negativo	695	119
Positivo	168	99

Peronal médico	*Negativo*	*Positivo*
Negativo	927	56
Positivo	89	9

Vulnerabilidad	*Negativo*	*Positivo*
Negativo	804	118
Positivo	114	45

Riesgo salud	*Negativo*	*Positivo*
Negativo	957	57
Positivo	63	4

Tabla 5.5: Matriz de confusión para los 8 modelos de clasificación binaria con *Random Forest*.

Este comportamiento se reporta en cada uno de los 8 modelos con diferencias relativas entre ellos, por ejemplo el modelo *Susceptibilidad* y *Riesgo de Salud* tienen peor rendimiento en la etiqueta de casos **Positivos** que otros como *Beneficios* o *Barreras*. A partir de estos resultados, se calcula el reporte de indicadores de evaluación plausibles para casos de clasificación binaria que se muestran en su formulación matemática en la tabla 5.4.

Basado en los resultados de la distribución de casos positivos y negativos en las matrices de confusión, se esperaría que los indicadores sean el reflejo del rendimiento en la capacidad de clasificación de los modelos, como por ejemplo la baja sensibilidad relacionada con la correcta clasificación de los casos verdaderamente positivos (tp) o la alta especificidad relacionada con la correcta clasificación de los casos verdaderamente negativos (tn).

Las tablas 5.6 y 5.7 muestran los valores resultantes de los indicadores de bondad de ajuste para en modelo RF. En promedio, los modelos lograron un buen nivel de *Exactitud* del 77.73 %, lo que significa que el 77.73 % de las predicciones realizadas por los modelos son correctas. Sin embargo, la *Precisión* es relativamente baja. La *Sensibilidad* también es baja, variando entre modelos como los casos de *Barreras*, *Beneficios* y *Vulnerabilidad*, en los que el indicador sobrepasa el 20 %. Esto indica que el modelo tiende a generar un

número significativo de falsos positivos y presenta dificultades para identificar verdaderos positivos en el conjunto de datos, lo cual concuerda con los resultados expuestos en las matrices de confusión. La *Especificidad* es alta, con un rango de 80-95 %, lo que sugiere que los modelos son efectivos para identificar verdaderos negativos. Como resultado, el *F1-score*, que combina *Precisión* y *Sensibilidad*, indica un desequilibrio entre la capacidad de predecir positivos y negativos.

Indicadores	**Gravedad**	**Barreras**	**Beneficios**	**Personal Médico**	**Vulnerabilidad**
Exactitud	87.42	74.19	72.80	86.40	78.72
Precision	19.23	35.05	43.72	16.44	28.48
Sensibilidad	9.62	34.88	35.21	12.24	29.56
Especificidad	95.70	83.95	85.14	93.79	87.20
F1-score	12.82	34.97	39.00	14.04	29.01
AUC	62.00	69.00	73.00	68.00	72.00

Tabla 5.6: Indicadores de evaluación de los modelos *Random Forest* (Parte 1).

Indicadores	**Susceptibilidad**	**Jornada**	**Riesgo salud**
Exactitud	90.47	83.81	88.71
Precision	8.00	20.18	6.35
Sensibilidad	6.56	20.00	5.97
Especificidad	95.49	91.04	94.18
F1-score	7.21	20.09	6.15
AUC	46.00	71.00	57.00

Tabla 5.7: Indicadores de evaluación de los modelos *Random Forest* (Parte 2).

La figura 5.5 muestra la curva ROC resultante para cada modelo estimado con RF. Los resultados del AUC varían entre modelo, destacando un rendimiento superior en los factores *Beneficio* con un AUC del 73.0 % y *Jornada* con un AUC del 71.0 %. Por otro lado, *Susceptibilidad* obtiene el AUC más bajo con un 46.0 %. El AUC promedio es de 69.17 % lo que indica un rendimiento moderado en términos de la capacidad de discriminación del modelo, exceptuando de este promedio los modelos de *Susceptibilidad* y *Riesgo de Salud*, que muestran mal rendimiento también en este indicador.

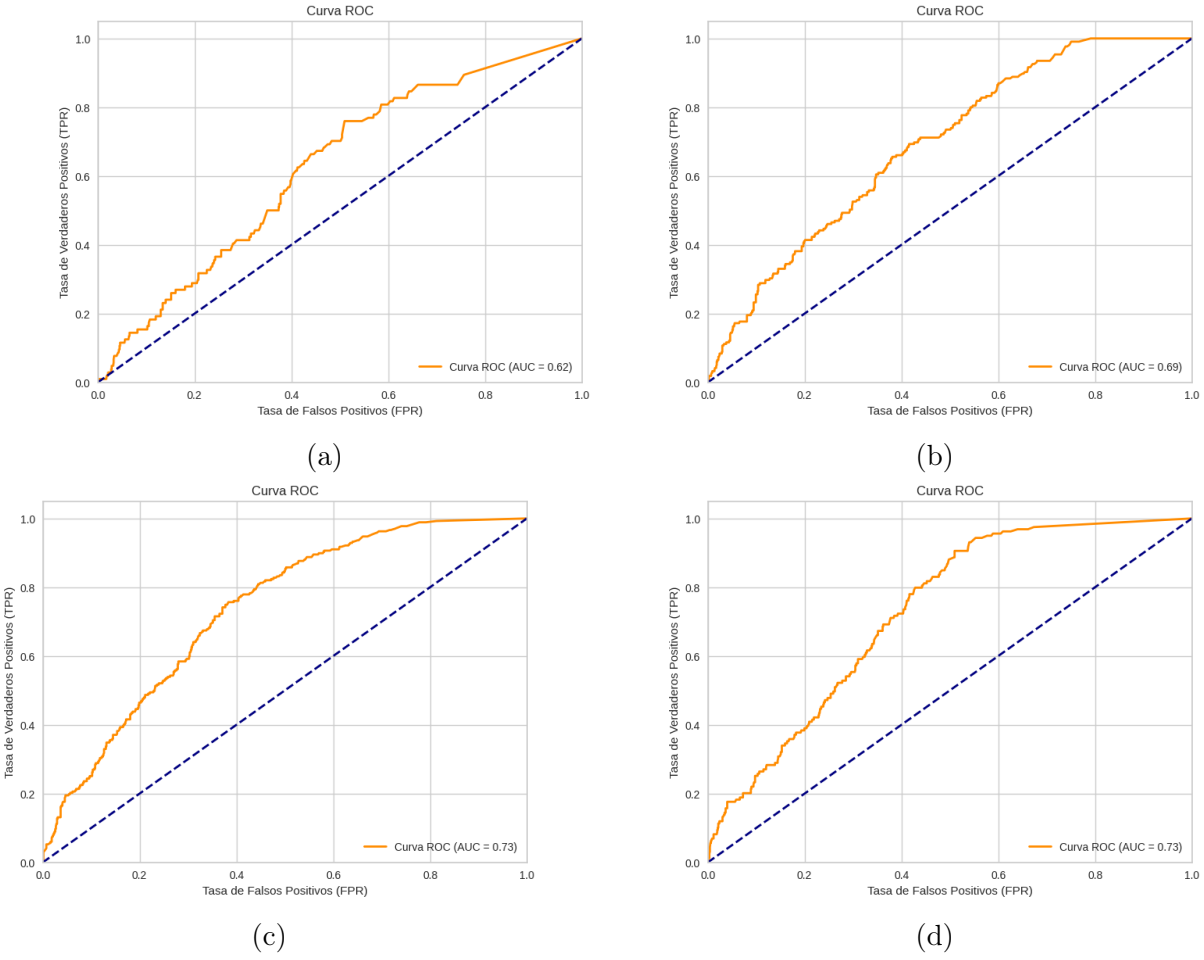


Figura 5.4: Curvas de ROC como resultado del modelo *Random Forest*: (a) Gravedad.(b) Barreras.(c) Beneficios.d) Vulnerabilidad. (Parte 1).

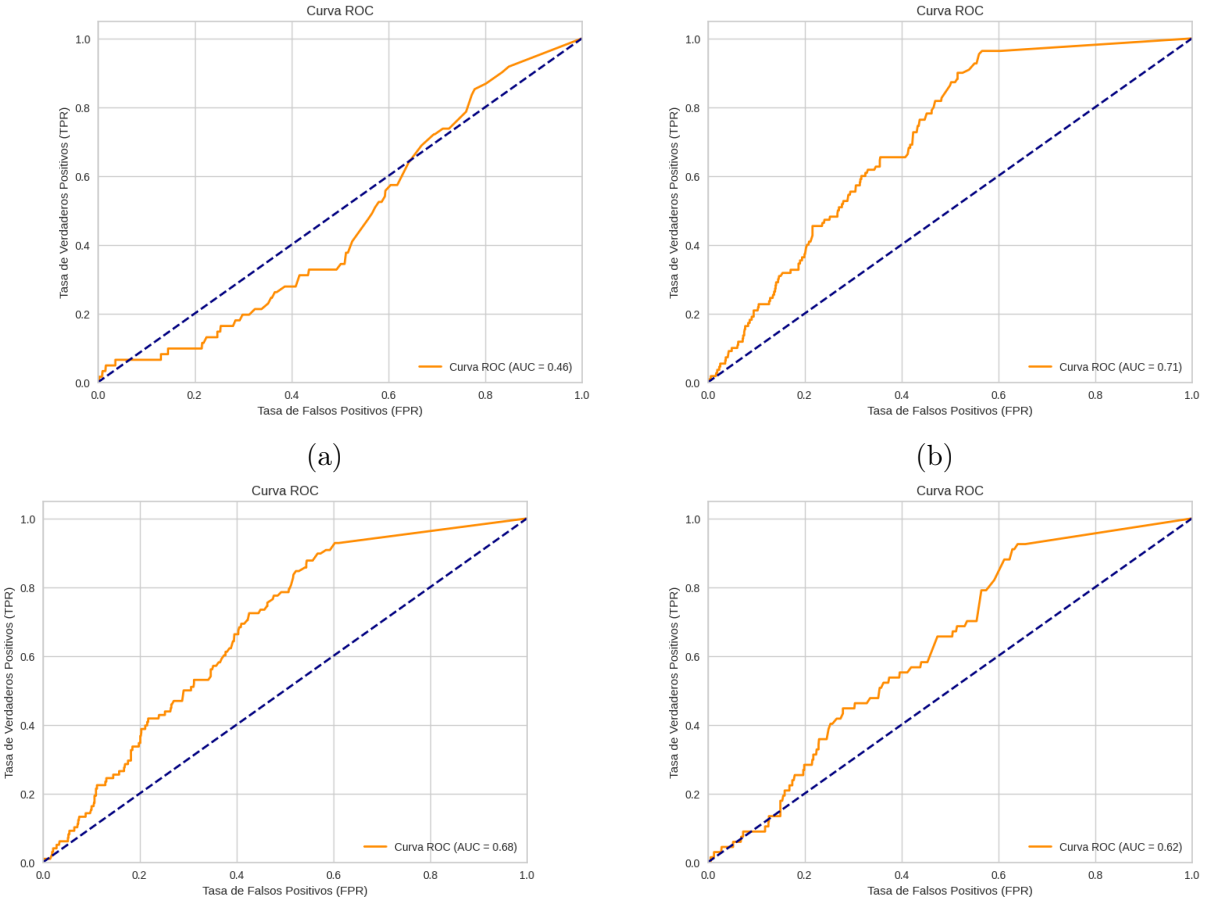


Figura 5.5: Curvas de ROC como resultado del modelo *Random Forest*: (a) Susceptibilidad.(b) Jornadas.(c) Personal Médico. (d) Riesgo de salud.(Parte 2).

Los resultados anteriores pudieran estar siendo afectados por un desequilibrio entre las clases, lo cual una de sus consecuencias es la obtención de un rendimiento generalmente muy sesgado contra la clase con las frecuencias más pequeñas. Por ejemplo, si los datos tienen una mayoría de muestras pertenecientes a la primera clase y muy pocas en la segunda clase, gran parte de los modelos predictivos maximizarán la precisión al predecir que todo será de la primera clase. Como resultado, generalmente hay un desequilibrio entre los indicadores de *Sensibilidad* y la *Especificidad*.

Una técnica para resolver el efecto del desequilibrio de las clases en la muestra de entrenamiento que genera la baja en la potencia de especificación es **balancear** los datos, técnica que se utiliza para contrarrestar el problema común en la clasificación de

documentos en el que una clase tiene significativamente menos instancias que otra, lo que puede llevar a un rendimiento deficiente del modelo de aprendizaje automático al favorecer la clase mayoritaria. Nuestra estrategia de balanceo de clases se centra en igualar el número de muestras en las clases minoritarias a las de las clases mayoritarias. A través de la implementación del paquete *scikit-learn* (Pedregosa et al., 2011a) en *Python* se puede indicar al algoritmo que hará uso de esta técnica seleccionando una de sus opciones siguientes:

- Muestreo balanceado: permite asignar un peso diferente a las clases en función de su importancia relativa. Esto significa que se le asignaría más peso a la clase minoritaria para que el modelo preste más atención a esa clase durante el entrenamiento.

$$Peso_i = N_{muestra}/(N_{clases} \cdot N_{clase_i}). \tag{5.9}$$

  donde $N_{muestra}$ representa el número de muestra en los datos, N_{clases} es el número de clases etiquetadas y N_{clase_i} el número de casos en la $clase_i$.

- Muestreo submuestra equilibrada: es el mismo modus operandi qué balanceado, pero excepto que los pesos se calculan basado en la muestra de *Bootstrap* para cada árbol creado.

Para este estudio se utiliza el modelo balanceado, ya que está disponible en la modelación no solo de *Random Forest*, sino de modelos SVM. Además, en los experimentos de modelación no se observaron diferencias entre usar una opción u otra.

Las matrices de confusión de la tabla 5.8 presentan la clasificación que se logra cuando se inserta la técnica de balanceo. Se observa una pequeña, pero poco significativa mejora en el poder de clasificación de los casos verdaderamente positivos del modelo. Sin embargo, esta mejora se logra a costa de la disminución de la potencia en clasificar los verdaderos casos negativos que tiene una reducción significativa en relación con la tabla 5.5 del modelo anterior. En términos generales, son los modelos de *Barrera* y *Personal Médico* que muestran un aumento mayor en cuanto a casos tp que antes eran clasificados como fp, con 6 casos redistribuidos cada uno.

Gravedad	*Negativo*	*Positivo*
Negativo	909	68
Positivo	89	15

Suceptibilidad	*Negativo*	*Positivo*
Negativo	968	52
Positivo	57	4

Barrera	*Negativo*	*Positivo*
Negativo	715	151
Positivo	135	80

Jornada	*Negativo*	*Positivo*
Negativo	876	95
Positivo	87	23

Beneficio	*Negativo*	*Positivo*
Negativo	680	134
Positivo	164	103

Peronal médico	*Negativo*	*Positivo*
Negativo	891	92
Positivo	84	14

Vulnerabilidad	*Negativo*	*Positivo*
Negativo	780	142
Positivo	111	48

Riesgo salud	*Negativo*	*Positivo*
Negativo	941	73
Positivo	62	5

Tabla 5.8: Matriz de confusión para los 8 modelos de clasificación binaria con *Random Forest* con clases balanceadas.

Las tablas 5.9 y 5.10 muestran los valores resultantes de los indicadores de bondad de ajuste para en modelo RF balanceado. En términos de los indicadores utilizados para evaluar estos resultados, la variación con relación al modelo anterior no es significativa, cómo se esperaría se obtiene un valor de sensibilidad un poco mayor debido a la redistribución de casos y una leve reducción proporcional también del indicador de *Especificidad*, tanto la *Exactitud* como el AUC aumentan en un rango de [0-2.5] puntos.

Indicadores	**Gravedad**	**Barreras**	**Beneficios**	**Personal Médico**	**Vulnerabilidad**
Exactitud	85.38	73.45	72.43	83.26	76.97
Precisión	17.86	34.48	43.46	12.61	26.32
Sensibilidad	14.42	37.21	38.58	14.29	31.45
Especificidad	92.94	82.45	83.54	90.13	84.82
F1-score	15.96	35.79	40.87	13.40	28.65
AUC	64.00	70.00	74.00	68.00	73.00

Tabla 5.9: Indicadores de evaluación de los modelos *Random Forest* balanceado (Parte 1).

Indicadores	**Susceptibilidad**	**Jornada**	**Riesgo salud**
Exactitud	89.92	83.16	87.51
Precision	7.14	19.49	6.41
Sensibilidad	6.56	20.91	7.46
Especificidad	94.90	90.22	92.80
F1-score	6.84	20.18	6.90
AUC	46.00	72.00	62.00

Tabla 5.10: Indicadores de evaluación de los modelos *Random Forest* balanceado (Parte 2).

La figura 5.7 muestra los resultados de la curva ROC para el modelo RF balanceado. Se observa un desempeño ligeramente mejor en comparación con el primer modelo. El factor *Beneficios* continúa siendo el más destacado, con un valor de AUC igual a 74.0 %, mostrando su alta capacidad predictiva en cuanto a riesgos para la salud. Por otro lado, *Susceptibilidad* sigue siendo el factor con el AUC más bajo, permaneciendo en 46.0 %. En general, estos resultados sugieren una mejora en la capacidad del modelo para predecir los constructos en comparación con el modelo anterior, con un aumento del 1 punto porcentual en el AUC en casi todos los casos.

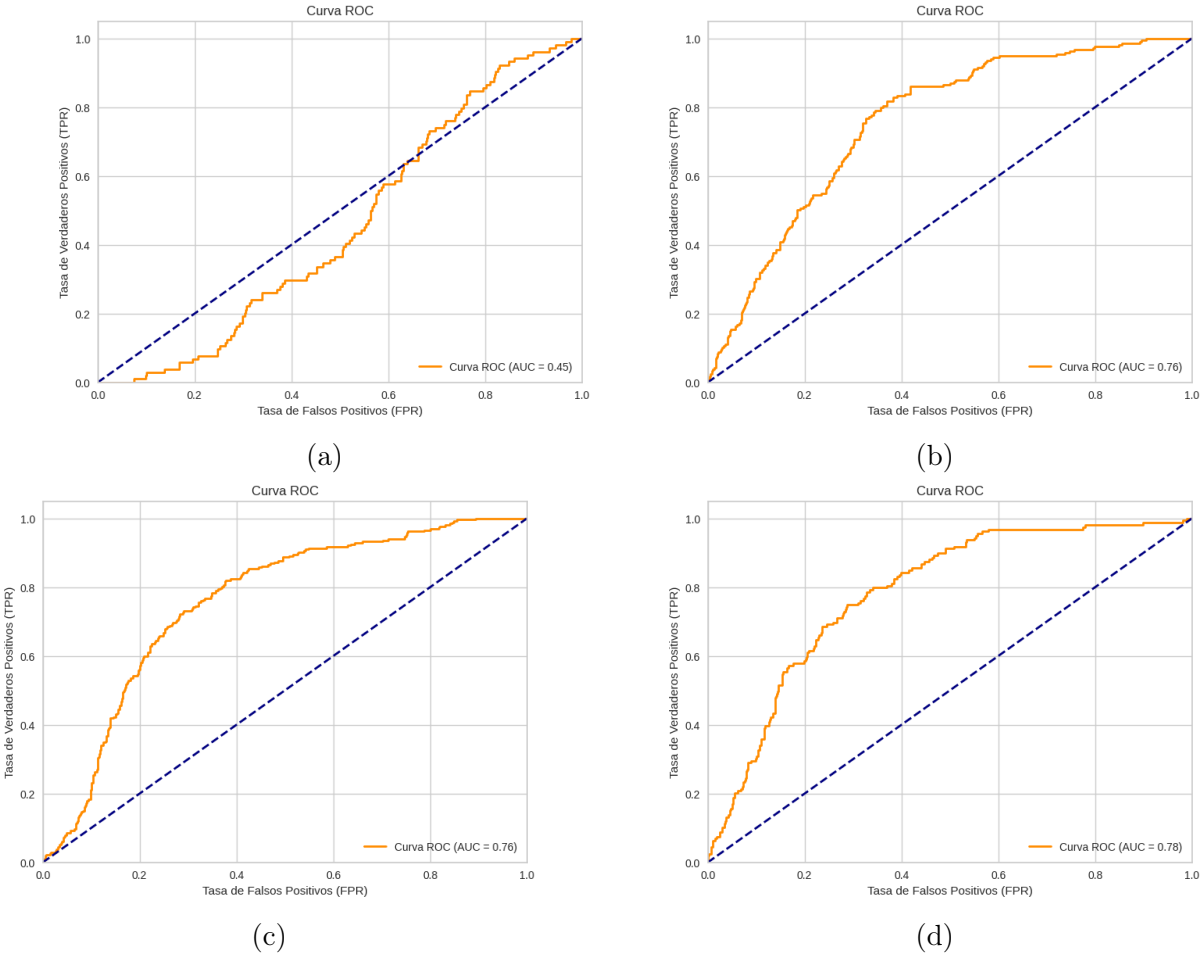


Figura 5.6: Curvas de ROC como resultado del modelo *Random Forest* balanceado: (a) Gravedad.(b) Barreras.(c) Beneficios.d) Vulnerabilidad (Parte 1).

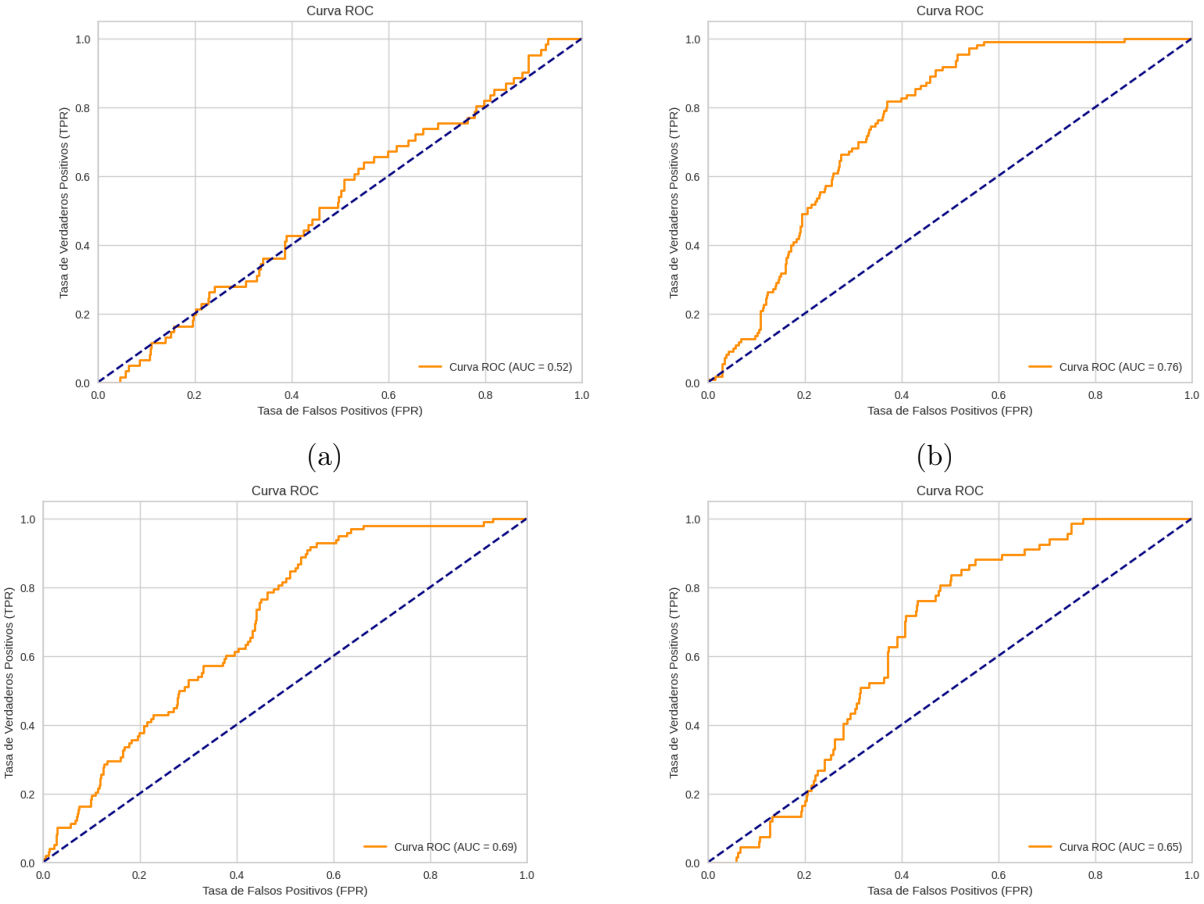


Figura 5.7: Curvas de ROC como resultado del modelo *Random Forest* balanceado: (a) Susceptibilidad.(b) Jornadas.(c) Personal Médico. (d) Riesgo de salud (Parte 2).

5.2. Resultados del modelo SVM

El sobreajuste (*overfitting*) es un fenómeno común en modelos de aprendizaje automático, especialmente cuando se utilizan modelos complejos o cuando el conjunto de datos de entrenamiento es relativamente pequeño en comparación con la complejidad del modelo. Esto ocurre cuando un modelo se ajusta demasiado a los detalles y el ruido en el conjunto de datos de entrenamiento en lugar de capturar las relaciones subyacentes y generalizables. La regularización es una técnica común en el aprendizaje automático para evitar el sobreajuste y mejorar la generalización del modelo.

En los modelos SVM, se puede aplicar la regularización mediante el parámetro de penalización de margen, comúnmente conocido como el parámetro C que representa el costo del ajuste y controla el equilibrio entre maximizar el margen entre las clases y minimizar el error en la clasificación. Ajustar este valor permite controlar el grado de regularización en un modelo SVM. Un valor pequeño de C aplicará una regularización fuerte, lo que significa que el modelo dará prioridad a maximizar el margen, incluso si eso significa cometer algunos errores en la clasificación. Mientras un valor grande de C aplicará una regularización débil, lo que significa que el modelo dará prioridad a minimizar el error en la clasificación, incluso si eso significa reducir el margen.

En este caso, dado que las etiquetas han sido generadas a través de la revisión manual de los documentos, los resultados pueden padecer de sobreajuste y ser la causa del bajo desempeño del modelo en la clasificación cuando se usa los datos de prueba. En ese sentido, usando el modelo SVM se tunean los parámetros de forma que se pueda hacer uso de esta técnica de regularización y validar si hay diferencias en los resultados cuando se usa RF. Cabe destacar que los modelos estimados se hacen haciendo uso de la técnica de balanceado de clases anteriormente descrita.

Los resultados de la tabla 5.11 muestran que se logra una reducción significativa del error en cuanto a la clasificación de los casos verdaderamente positivos (tp) esto a costa de la reducción de la eficiencia en la clasificación de los casos verdaderamente negativos (tn). Esto puede pasar con los modelos SVM que buscan maximizar el margen entre las clases y en escenarios con valores pequeños, de C este margen es más amplio, lo que puede terminar beneficiando la clasificación de casos positivos.

Gravedad	*Negativo*	*Positivo*
Negativo	471	506
Positivo	32	72

Suceptibilidad	*Negativo*	*Positivo*
Negativo	232	788
Positivo	14	47

Barrera	*Negativo*	*Positivo*
Negativo	546	320
Positivo	39	176

Jornada	*Negativo*	*Positivo*
Negativo	552	419
Positivo	16	94

Beneficio	*Negativo*	*Positivo*
Negativo	557	257
Positivo	68	199

Peronal médico	*Negativo*	*Positivo*
Negativo	443	540
Positivo	9	89

Vulnerabilidad	*Negativo*	*Positivo*
Negativo	509	413
Positivo	20	139

Riesgo salud	*Negativo*	*Positivo*
Negativo	443	571
Positivo	8	59

Tabla 5.11: Matriz de confusión para los 8 modelos de clasificación binaria con SVM con clases balanceadas.

Estos resultados tienen un efecto en los indicadores de evaluación también. Las tablas 5.12 y presentan 5.12 estos resultados y se observa que efectivamente hay una reducción de la *Exactitud* con relación al obtenido en los modelos RF, probablemente causada por el efecto de la drástica reducción de los casos *tn* en el denominador del indicador.

Indicadores	**Gravedad**	**Barreras**	**Beneficios**	**Personal Médico**	**Vulnerabilidad**
Exactitud	46.438	66.79	69.935	49.214	59.94
Precision	9.365	35.48	43.640	14.149	25.18
Sensibilidad	88.060	81.86	74.532	90.816	87.42
Especificidad	43.688	63.05	68.428	45.066	55.21
F1-score	16.930	49.51	55.048	24.484	39.10

Tabla 5.12: Indicadores de evaluación de los modelos SVM balanceado (Parte 1).

Indicadores	Susceptibilidad	Jornada	Riesgo salud
Exactitud	25.81	59.76	46.44
Precision	5.63	18.32	9.37
Sensibilidad	77.05	85.45	88.06
Especificidad	22.75	56.85	43.69
F1-score	10.49	30.18	16.93

Tabla 5.13: Indicadores de evaluación de los modelos SVM balanceado (Parte 2).

Comparando los resultados de los modelos de clasificación binaria entre RF y SVM, se pueden observar algunas diferencias significativas. Los modelos RF tienen un rendimiento de precisión más alto en la mayoría de las categorías, con una *Exactitud* promedio más alta en comparación con el SVM. Esto sugiere que los modelos RF tienden a hacer una mejor predicción global. También RF supera al SVM en cuanto a los resultados de *Precisión*, esto dado que en el primero se tienden a tener menos falsos positivos en comparación con el SVM, lo que es beneficioso en aplicaciones donde la minimización de los falsos positivos es importante. En cuanto a *Sensibilidad* y *Especificidad*, los modelos SVM tienen una mayor sensibilidad en la mayoría de las categorías en comparación con RF. Esto significa que los SVM tienen una mejor capacidad para detectar casos positivos en comparación con RF. La especificidad es relativamente similar en ambos modelos, aunque en algunos casos el SVM supera al RF.

Asimismo, los modelos RF tienen puntajes *F1-score* más altos en la mayoría de las categorías en comparación con los modelos SVM. Esto sugiere que los modelos RF encuentran un mejor equilibrio entre precisión y sensibilidad. Finalmente, los modelos SVM superan a los modelos RF en términos de AUC en varias categorías.

Dada la cantidad de indicadores de evaluación que se dispone, elegir un indicador que evalué de forma veraz y segura los modelos de clasificación es un debate bastante documentado en la literatura y en el caso de clasificación de documentos no es la excepción. En Sokolova and Lapalme (2009) se encuentran sugerencias para seleccionar los indicadores a tomar en cuenta cuando se trabaja con clasificación de documentos. Los autores realizan una serie de pruebas (los autores establecen 8) en la matriz de confusión que consisten básicamente en realizar operaciones elementales de matrices como suma y multiplicación, probando si existe o no invariabilidad en los indicadores, los cuales se corresponden con los mostrados en la tabla 5.4. En relación con sus resultados del estudio, concluyen que los indicadores *Precisión* y *Especificidad* pueden ser más confiables cuando el etiquetado manual sigue reglas rigurosas para una clase negativa. Mientras, en ausencia de tales re-

glas, el desacuerdo entre las etiquetas de datos y las etiquetas negativas asignadas por un clasificador puede depender de factores subjetivos y fluctuar.

También establecen que, en general, cuando se trata de la clasificación de documentos, generalmente se tiene una clase positiva que está bien definida. Sin embargo, la clase negativa suele ser bastante heterogénea y está compuesta por documentos no relacionados que se componen de «todo lo que no es positivo». Esto es lo que provoca que los conjuntos de datos para la clasificación de documentos a menudo estén significativamente desequilibrados. En cuyo caso, la presencia de una clase negativa que complementa la clase positiva favorece el uso de *F1-score* como indicador debido a la métrica que este evalúa en conjunto con otro indicador como el AUC para complementar el análisis.

En este estudio se encuentra similar escenario donde se ha realizado el proceso de etiquetado siguiendo la regla de que «todo lo que no es positivo es negativo», razón por la cual se enfrentan desequilibrios entre las clases. Por lo que, siguiendo a estos autores, se puede seleccionar los mejores de los 8 modelos, según el valor de indicadores (tabla 5.14) de *F1-score*, *Precisión* y AUC. Siendo que los mejores modelos serían los estimados mediante SVM balanceado que en todos los 8 modelos posee mejor valor del indicador, resaltando como los mejores en clasificar los casos para *Beneficios*, *Barreras*, *Vulnerabilidad* y *Jornada*.

Modelos	**Gravedad**	**Barreras**	**Beneficios**	**Personal Médico**	**Vulnerabilidad**
RF	12.82	34.97	39.00	14.04	29.01
RF balanceado	15.96	35.79	40.87	13.40	28.65
SVM balanceado	16.93	49.51	55.05	24.48	39.10

(a) Primera parte de la tabla con F1-score para los modelos.

Modelos	**Susceptibilidad**	**Jornada**	**Riesgo Salud**
RF	7.21	20.09	6.15
RF balanceado	6.84	20.18	6.90
SVM balanceado	10.49	30.18	16.93

(b) Segunda parte de la tabla con F1-score para los modelos.

Tabla 5.14: Indicador F1-score para los 8 modelos estimados según tipo de modelo.

En este capítulo, se ha explorado y evaluado el rendimiento de dos poderosos modelos de clasificación como son SVM y RF. El objetivo principal era identificar el mejor enfoque para abordar un problema de clasificación. Para ello, se ha entrenado y evaluado un total de ocho modelos diferentes, combinando distintas configuraciones de hiperparámetros y técnicas de preprocesamiento de datos.

Los resultados han arrojado una comprensión profunda del comportamiento de estos modelos en el conjunto de datos específico. Se observa que en términos de *Exactitud* los modelos RF se obtuvieron modelos relativamente buenos con más de 80 % de *Exactitud* de clasificación, como el caso de los modelos para Beneficios, *Barreras* y *Gravedad*, estos porcentajes se reducen cuando se modela usando SVM, en la que ninguno modelo obtiene este mismo nivel quedando por debajo del 60 % en la mayoría de los casos, sin embargo, si se obtienen mejores resultados en otros indicadores sobre todo en lo que los casos de *Sensibilidad*, *F1-score* y AUC. Esto probablemente influenciado por el aumento en los casos clasificados como verdaderos positivos frente a la reducción de casos clasificados como casos verdaderamente negativos, es decir, los 8 modelos bajo SVM son más sensibles a los casos *verdaderamente positivos* que en los modelos estimados con RF.

En resumen, este capítulo ha sido fundamental para el proceso de toma de decisiones y ha permitido identificar las fortalezas y debilidades de los modelos SVM y RF en relación con nuestra tarea de clasificación. Estos resultados ayudan a cumplir con el objetivo 2 planteados en la Sección 1.2.

Capítulo 6

Discusión y conclusiones

En este capítulo se presentan las conclusiones y discusiones derivadas de esta investigación. A lo largo de este estudio, se exploró de forma exhaustiva diversas técnicas de procesamiento de datos, modelos de aprendizaje automático y análisis de textos para comprender en profundidad los patrones y las percepciones relacionadas con la vacunación contra el COVID-19.

Abordando los resultados desde la literatura revisada, se puede discutir algunos puntos de los resultados y metodologías que han sido utilizados. Por ejemplo, en Teng and Khong (2022) abordan el uso de la técnica de TF-IDF y el análisis de clúster con la finalidad de generar variables determinantes de la «intención de vacunarse» para ser modeladas en regresiones múltiples. También en Muric et al. (2021) se aborda una metodología de etiquetado manual para crear modelos de clasificación y analizar la relación de estas etiquetas con la evolución de casos y otras variables de incidencia del virus.

En Teng and Khong (2022) los autores logran identificar 9 variables, de las cuales 4 son los constructos que se han utilizado en el presente estudio y que son parten del modelo HBM, los demás son libremente relacionadas con narrativas identificadas en la revisión de los documentos, en ese orden de las variables identificada por los autores una es la confianza que tienen las personas en el gobierno referido a la duda sobre los motivos detrás del anuncio de la eficacia de la vacuna Pfizer. En el análisis de clúster que se ha presentado en la Sección 4.3 también se identificaron narrativas que repercuten en la confianza de los individuos en la planificación del gobierno de la jornada de vacunación, Evidenciándose como un aspecto relevante en cuanto al proceso de vacunación del COVID-19.

Los autores también utilizan la técnica de TF-IDF para generar las principales palabras que formarían el análisis de clúster que realizaron, teniendo palabras como «vacuna», «ser», «persona», «año» y «saber» entre las de mayor frecuencia, algo que incluso con las diferencias gramaticales del lenguaje es común con lo resultados que se han obtenido evidencia de que el formato no es tan relevante cuando se trata de la información que se difunde en RRSS en el contexto de vacunación COVID-19.

Siguiendo la línea de investigación de Carrieri et al. (2023), se analiza el impacto del veto a la vacuna AstraZeneca en la intención de vacunarse. Este aspecto es relevante en los resultados obtenidos y fue considerado en el diseño de la recopilación de información que abarca el período en el que se inició el veto en Europa. Este impacto se reflejó principalmente en los términos del clúster 2, el cual aborda la narrativa asociada a las barreras. Cabe destacar que el clúster 2 demostró un buen desempeño en términos de los modelos de clasificación.

Con relación a la evaluación de los modelos se revisó el estudio de Sokolova and Lapalme (2009) donde se presenta una metodología para la selección de indicadores de evaluación de los modelos de clasificación a partir de la matriz de confusión. Replicando el tablero que presentan los autores para medir las «invariabilidad» se obtuvieron iguales escenarios y se evalúa cuáles son las implicaciones de estas medidas frente a los resultados de los modelos de clasificación aquí presentados. Los autores relatan sobre la existencia de una clase positiva definida y una clase negativa heterogénea que agrupa dentro de ella todo lo que no se ha podido identificar, por lo que, hay muchos más ejemplos de documentos en la clase negativa que en la clase positiva. En otras palabras, la clase negativa es heterogénea y contiene una variedad de tipos de documentos que no están relacionados entre sí. Con esto se plantea elegir las medidas como *F1-score* para evaluar la bondad de ajuste del modelo de clasificación. Es desde este punto donde se toman como referencia los modelos de *Barreras*, *Beneficios*, *Vulnerabilidad* y *Jornada* como los mejores, y que al menos 3 de estos también se reflejan en algunos de los estudios revisados, ya que refleja dos de los constructos del modelo HBM y una de las variables que más se repiten en que es la confianza en los organismos sobre todo en los gubernamentales.

No se puede dejar de lado, como gran limitación frente estos trabajos, la participación de un equipo de revisión que realiza los etiquetados manuales. La dependencia de estos para poder revisar un mayor número de documentos, y asegurar una menor subjetivad, si bien puede ser un aporte a las metodologías, es un inconveniente cuando no se dispone de iguales recursos. Sin embargo, el ejercicio realizado en el estudio presentado se mantuvo transparente en cuanto al proceso llevado cabo y se espera que en futuros trabajos sea mejorado.

El uso de las RRSS como fuente de información para estudiar las narrativas que influyen en una acción de relevancia, como la vacunación, utilizando la modelización matemática, es una estrategia efectiva, sin costos adicionales asociados a técnicas de muestreo o captura de información que a menudo conllevan tasas de rechazo. Asimismo, el estudio de las narrativas centradas en las vacunas quizás no sea nuevo en la literatura, pero que sea abordado para solo documentos que estén en español, si genera un aporte a la literatura. Dado que durante la revisión literaria no se encontró ninguna aplicación que no fuera a documentos en inglés u otros idiomas.

A modo de repaso general, en este estudio se han recopilado 11,787 *tweets* de la aplicación de RRSS *Twitter* a partir de los datos proporcionados por el grupo *Panacea Lab*. El grupo se enfocó en la captura de los *tweets* publicado en el periodo de marzo 2020 a abril 2022 relacionados con el COVID-19, entre ellos, la vacunación. Para obtener estas opiniones se ha utilizado las técnicas de NLP y aprendizaje automático disponibles. Se comenzó por la aplicación de preprocesamiento que implica la limpieza y transformación de los textos en listas de términos denominados token que contuvieran y/o representan la esencia de la información compartida en cada *tweet* recopilado. En ese sentido, se aplicó la *lemmatizacion* de los textos para trabajar con la raíz de cada palabra. De esta forma se hizo más eficiente el proceso de análisis de frecuencia, que fue realizado con la técnica de TF-IDF, previamente eliminadas las palabras vacías que por su función gramatical suelen tener una frecuencia alta.

Después del preprocesamiento de los datos se obtuvieron las *Word Embedding* de cada término, es decir, su representación vectorial mediante el modelo de Mikolov et al. (2013b) denominado *Word2vec*, usando una arquitectura basada en CBOW y muestreo negativo. Con este modelo se obtuvieron la representación vectorial de cada término o *token* de los *tweets* recopilados. Este modelo es ampliamente reconocido por su capacidad para capturar de forma eficiente las representaciones vectoriales de palabras que conservan significado semántico. Se exploraron algunas métricas como la similitud del coseno o las pruebas de analogías impulsadas por Mikolov et al. (2013c) para evaluar los resultados del modelo, No obstante, este aspecto ha sido muy poco incursionado y por eso se han planteado las recomendadas por los mismos autores como son la similitud del coseno y la analogía de palabras.

Las representaciones vectoriales permitieron que se realizara el análisis de clúster usando un modelo k-medias que permitió la agrupación de los términos en clúster. Además, se pudo representar narrativas por su similitud semántica. Sin embargo, debido a que la representación vectorial se realiza para un tamaño de vector de cierta magnitud, $D \geq 50$ se aplicó un análisis de PCA seleccionando los primeros dos componentes que explicaban

más entre el 80 % y el 90 % de la varianza de los datos. Esta técnica permite reducir la dimensionalidad de los datos manteniendo la información relevante. Este análisis PCA se realizó a una muestra de los términos previamente seleccionados mediante un ranking de palabras con mayor indicador obtenido en la técnica TF-IDF.

Mediante la creación de escenarios que combinaban los parámetros de tamaño del vector de la representación vectorial en los que se usaron dos tamaños 50D y 100D y el número de palabras a seleccionar del ranking, alternándolo entre 100,150 y 200 palabras, de esta forma se crearon 6 escenarios. Se seleccionó el valor de k óptimo dado los valores obtenidos de un set de 6 indicadores que analizan la cohesión y separación de los clústeres para cada valor de k iterado. Como resultado se obtuvo un k óptimo igual a 19 con un D=100 y 150 palabras.

Después de realizar el proceso de agrupación y clasificación, se puede concluir que algunas narrativas son más fáciles de capturar y definir que otras, lo que puede estar relacionado con cuán efectivo es el discurso en cada caso. Esto se evidenció con los modelos de *Barreras* y *Beneficios* que resultaron ser los mejores en las narrativas etiquetadas. En otros modelos como la *Gravedad* y *Susceptibilidad* se evidenció límites difusos entre ambos, dado que expresan preocupaciones similares sobre el problema evaluado. Otras como *Vulnerabilidad* y *Jornadas* también presentaron resultados relativamente mejores, lo que puede inferirse como consecuencia de que no responden a la aplicación de un *framework* teórico previo, sino que fueron etiquetados como resultado de la misma revisión. Lo anterior cumple con el Objetivo 1 planteado en la Sección 1.2, y tiene su potencial debido a que no se trata de analizar las opiniones desde un punto de vista positivo o negativo, sino desde sus diferencias y fuerza de formalización como narrativa común dentro de una comunidad o círculo (en este caso de la plataforma de RRSS).

Luego de obtener los clústeres, se realizó una revisión manual de una muestra de los *tweets* recopilados. Se identificaron las narrativas relacionadas con los 4 constructos del modelo HBM, que analiza cómo se forman las creencias y decisiones de salud. Durante la revisión manual, se aplicaron 8 etiquetas siguiendo la metodología utilizada por Wang et al. (2021). Esta metodología permitió identificar narrativas vinculadas a temas de confianza y reclamos de índole social relacionados con la vacunación. Luego, se emplearon modelos de clasificación, como RF y SVM, para evaluar la *Exactitud*, *Precisión*, *Especificidad* y otros indicadores. Estos modelos se utilizaron para medir la capacidad de clasificación en función de los datos reales y asegurar que las etiquetas se correspondieran con precisión a las narrativas. Este proceso permitió cumplir el Objetivo 2 establecido en la Sección 1.2.

En términos generales, los modelos de clasificación ejecutados muestran un buen nivel de *Exactitud*, lo que indica que son capaces de realizar predicciones precisas en sus respectivos dominios de aplicación. Sin embargo, la precisión y la sensibilidad promedio son relativamente bajas, lo que indica un desafío en la identificación de verdaderos positivos. La *Especificidad* es alta, lo que sugiere una buena capacidad para identificar verdaderos negativos. Los valores de *F1-score* indican un equilibrio mixto entre *Precisión* y *Sensibilidad* en los modelos.

Los resultados globales evidencian que no es solo el desbalance de las clases lo que lleva a tener un error alto cuando se clasifican los casos que se etiquetaron como positivos, dado que la aplicación de técnicas de balanceo no mejoro los resultados.

En ese mismo orden se destaca la importancia de evaluar la selección entre los modelos de supervisión, especialmente cuando se trabaja en la clasificación de documentos, en los que involucra un proceso de etiquetado manual. Esto se debe a que las clases negativas no pueden definirse con un grado absoluto de certeza, dadas las limitaciones de tiempo asociadas a la revisión exhaustiva de la totalidad de los documentos. En esta investigación, se evidenció que el modelo SVM sobresale en la identificación de la clase positiva, gracias a la capacidad de ajustar los márgenes que influyen en su enfoque de clasificación. Esto se vuelve especialmente relevante cuando se trabaja con clases negativas menos definidas, como es común en el análisis de documentos.

Por último, en resumen de los pasos anteriores y las conclusiones obtenidas, se evidencia la existencia de narrativas identificables con respecto a las creencias relacionadas con la vacunación contra el COVID-19. Este logro ha sido posible gracias a la combinación de diversas metodologías y la aplicación de técnicas de análisis de datos y aprendizaje automático, sin recurrir a enfoques estadísticos tradicionales como encuestas o sondeos. Además, se ha logrado capturar narrativas que giraron en torno a un evento que tuvo su inicio y pico de actividad en el pasado, lo cual representa un valor agregado del uso de estas técnicas.

Dado el trabajo realizado se quedan algunas líneas futuras que se pueden y se detallan a continuación. Trabajar con modelos que incorporen multiclases en lugar de clasificación binaria, lo cual permitiría ver no solo la clasificación de los casos verdaderos, sino la distribución de estos entre las clases etiquetadas, validando si existen distribuciones compartidas entre unas y otras. Se podría abordar la aplicación con modelos de supervisión automática más complejos que han demostrado también tener buena potencia en la clasificación de documentos como las RN, entre ellas las redes neuronales recurrentes.

Asimismo, siguiendo con la línea de modelos de representación vectorial, se pueden aplicar diferentes modelar *Word Embedding* disponibles como *GloVe* o *Bert* para evaluar y comparar la potencia entre los, esto además sería un aporte a la literatura disponible en cuanto al mejor modelo para vectorización de documentos que provienen de RRSS.

Otro punto sería la incorporación de datos que provienen de otras fuentes/plataformas que han sido también utilizados en la literatura, como son: *Instagram*, *Facebook* o *YouTube* donde también se puede capturar información de opiniones de sus usuarios a través de los comentarios y publicaciones. La unificación de estos datos en una única base de datos puede ser un desafío, pero como muestran en Muric et al. (2021) es interesante de realizar dado que entre estas la rapidez con la que se difunden las informaciones es similar. De igual manera, sería interesante abordar no solo las narrativas que domina un tema en específico, como es el caso de este estudio, sino como se propagan estas en términos de las redes de usuarios que son una comunidad en las RRSS, es decir, ver la dispersión de estas creencias mediante un modelo que considere la interacción de los usuarios tanto desde sus seguidores como los *repost*.

En resumen, el futuro de la investigación en este campo ofrece emocionantes oportunidades para explorar. Algunos de los retos mencionados aquí son solo una muestra de los múltiples enfoques que prometen contribuir significativamente al avance del análisis de opiniones en entornos digitales.

Bibliografía

Amaratunga, D., Cabrera, J., and Lee, Y.-S. (2008). Enriched random forests. *Bioinformatics*, 24(18):2010–2014.

Amat Rodrigo, J. (2020). Árboles de decisión, random forest, gradient boosting y c5.0. Recuperado de: https://cienciadedatos.net/documentos/33_arboles_decision_random_forest_gradient_boosting_c50.

Amat Rodriguez, J. (2017). Análisis de componentes principales (principal component analysis, pca) y t-sne. Recuperado de: https://cienciadedatos.net/documentos/35_principal_component_analysis.

Amit, Y. and Geman, D. (1997). Shape quantization and recognition with randomized trees. *Neural Computation*, 9(7):1545–1588.

Awad, M. and Khanna, R. (2015). *Support Vector Machines for Classification*, pages 39–66.

Barbona, I. and Beltrán, C. (2016). Supervised classification method support vector machine applied to automatic text classificatio. *Revista de Epistemología y Ciencias Humanas*.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Carrieri, V., Guthmuller, S., and Wübker, A. (2023). Trust and covid-19 vaccine hesitancy. *Scientific Reports*, 13.

Chaubard, F., Fang, M., Genthial, G., Mundra, R., and Socher, R. (2019). Word vectors i: Introduction, svd and word2vec 2. part i. In *Proceedings of CS224n: Natural Language Processing with Deep Learning*, pages 1–14, Stanford University, USA.

Chen, G., Jaradat, S. A., Banerjee, N., Tanaka, T. S., Ko, M. S. H., and Zhang, M. Q. (2002). Evaluation and comparison of clustering algorithms in analyzing es cell gene expression data. *Statistica Sinica*, 12:241–262.

Cinelli, M., Quattrociocchi, W., and Galeazzi, A. (2020). The covid-19 social media infodemic. *Scientific Reports*, 10.

Colome Abril, J. (2012). Aproximación al reajuste automático de centroides mediante la heurística de lloyd para resolver el problema de las k-medias. *Universitat Oberta de Catalunya*.

Cutler, A., Cutler, D. R., and Stevens, J. R. (2012). *Random Forests*, pages 157–175. Springer New York, New York, NY.

Dabbura, I. (2018). K-means clustering: Algorithm, applications, evaluation methods, and drawbacks. Recuperado de: https://github.com/ImadDabbura/blog-posts/blob/master/notebooks/Kmeans-Clustering.ipynb.

Firth, J. (1957). *Papers in Linguistics, 1934-1951*. Oxford University Press.

Friedl, H. and Stampfer, E. (2006). *Cross-Validation*. John Wiley Sons, Ltd.

Gil, C. (2018). Análisis de componentes principales (pca). Recuperado de: https://rpubs.com/Cristina_Gil/PCA.

Guzmán León, E. (2013). Métricas para la validación de clustering. In *Apuntes: Mineria de datos, Universidad Nacional de Colombia. Facultad de Ingeniería de Sistemas y Computation*.

Honnibal, M., Montani, I., Van Landeghem, S., and Boyd, A. (2020). spaCy: Industrial-strength Natural Language Processing in Python.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666. Award winning papers from the 19th International Conference on Pattern Recognition (ICPR).

Jost, Z. (2019). Noise contrastive estimation.a gentle introduction. Recuperado de: https://towardsdatascience.com/noise-contrastive-estimation-246446ea9aba.

Jurafsky, D., Martin, J., Norvig, P., and Russell, S. (2014). *Speech and Language Processing*. Pearson Education.

Karimi, A. (2023). Nlp's word2vec: Negative sampling explained. Recuperado de: https://www.baeldung.com/cs/nlps-word2vec-negative-sampling.

Ledesma, R. (2008). Introducción al bootstrap: Desarrollo de un ejemplo acompañado de software de aplicación. *Tutorials in Quantitative Methods for Psychology*, 4.

Loomba, S., de Figueiredo, A., and Piatek, S. (2021). Measuring the impact of covid-19 vaccine misinformation on vaccination intent in the uk and usa. *Nat Hum Behav*, (5):337—-348.

Louppe, G. (2015). *Understanding Random Forests: From Theory to Practice*. PhD thesis, University of Liège.

Lu, Y. and Zhou, H. H. (2016). Statistical and computational guarantees of lloyd's algorithm and its variants. *CoRR*.

Luna, S. (2012). Manual práctico para el diseño de la escala likert. *Xihmai*, 2.

Ma, Z. and Collins, M. (2018). Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. *CoRR*, abs/1809.01812.

Mackay, D. (2003). *Chapter 20. An Example Inference Task: Clustering*, pages 284–292. Cambridge University Press.

McCormick, C. (26 abril 2019). Word2vec tutorial - the skip-gram model. Recuperado de: http://www.mccormickml.com.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space.

Mikolov, T., Kopecky, J., Burget, L., Glembek, O., and Cernocky, J. (2009). Neural network based language models for highly inflective languages. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4725–4728.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546.

Mikolov, T., Yih, S. W.-t., and Zweig, G. (2013c). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2013)*. Association for Computational Linguistics.

Mishra, S., Sarkar, U., Taraphder, S., Datta, S., Swain, D., Saikhom, R., Panda, S., and Laishram, M. (2017). Principal component analysis. *International Journal of Livestock Research*, page 1.

Montavon, G. (2020). *Introduction to Neural Networks*, pages 37–62. Springer International Publishing, Cham.

Moreno San Pedro, E. and Gil Roales-Nieto, J. (2003). El modelo de creencias de salud: Revisión teórica, consideración crítica y propuesta alternativa. i: Hacia un análisis funcional de las creencias en salud. *International Journal of Psychology and Psychological Therapy*, 3:91–109.

Muric, G., Wu, Y., and Ferrara, E. (2021). Covid-19 vaccine hesitancy on social media: Building a public twitter data set of antivaccine content, vaccine misinformation, and conspiracies. *JMIR Public Health Surveill*, 7(11):e30642.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011a). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011b). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pierri, F., Perry, B., and DeVerna, M. e. a. (2022). Online misinformation is linked to early covid-19 vaccination hesitancy and refusal. *Scientific Reports*, 12.

Pilehvar, M. T. and Camacho-Collados, J. (2021). *Word Embeddings*, pages 25–40. Springer International Publishing, Cham.

Qaiser, S. and Ali, R. (2018). Text mining: Use of tf-idf to examine the relevance of words to documents. *International Journal of Computer Applications*, 181.

Real Academia Española (2023). Diccionario de la lengua española. [versión 23.6 en línea].

Reyes-Figueroa, A. (2021). Métricas para evaluar algoritmos de agrupamiento. In *Introducción a la Minería de datos, Universidad del Valle de Guatemala.*

Rodrigo Amat, J. (2017). Máquinas de vector soporte (support vector machines, svms). Recuperado de: https://cienciadedatos.net/documentos/34_maquinas_de_vector_soporte_support_vector_machines.

Rodriguez-Insuasti, H., Mendoza-Zambrano, D., and Vasquez-Giler, M. (2020). El modelo de creencia de salud (hbm): un análisis bibliométrico. *FACSalud UNEMI*, 4:43 – 54.

Rong, X. (2014). Word2vec parameter learning explained. *CoRR*, abs/1411.2738.

Rui, Y. (2021). An improved k-means clustering algorithm for global earthquake catalogs and earthquake magnitude prediction. *Journal of Seismology*, 25(3):1005–1020.

Schmidt, R. M. (2019). Recurrent neural networks (rnns): A gentle introduction and overview. *CoRR*, abs/1912.05911.

Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437.

Suárez, E. J. C. (2014). *Tutorial sobre Máquinas de Vectores Soporte (SVM)*. Dpto. de Inteligencia Artificial, ETS de Ingeniería Informática, Universidad Nacional de Educación a Distancia (UNED). Versión inicial: 2013. Última versión: 11 Julio 2014.

Teng, S., J. N. and Khong, K. (2022). Using big data to understand the online ecology of covid-19 vaccination hesitancy. *Humanities and Social Sciences Communications*, 9.

Wang, B., Wang, A., Chen, F., Wang, Y., and Kuo, C.-C. J. (2019). Evaluating word embedding models: methods and experimental results. *APSIPA Transactions on Signal and Information Processing*, 8(1).

Wang, H., Li, Y., Hutch, M., Naidech, A., and Luo, Y. (2021). Using tweets to understand how covid-19–related health beliefs are affected in the age of social media: Twitter data analysis study. *J Med Internet Res*, 23(2):e26302.

World Health Organization (2023). Covid-19 weekly epidemiological update, edition 156, 17 august 2023. Technical documents.

Xezonakis, I. S. and Leivadaros, S. (2021). N-ary huffman encoding using high-degree trees - A performance comparison. *CoRR*, abs/2105.07073.

Appendices

Apéndice A

Códigos en *Python* utilizados

Se puede acceder al código fuente de este estudio elaborado en *Python* mediante el enlace aquí depositado, el cual redirigirá a un repositorio en la plataforma Github. Estos códigos permitirán explorar y comprender en detalle la implementación de las metodologías y técnicas utilizadas para la recolección, preprocesamiento, y modelización de los datos.

Enlace: https://github.com/cinthiaEquivocada/TFM2023.git