UNIVERSIDAD TORCUATO DI TELLA

Tesis de Maestría en Gestión y Análisis de datos

---

# Mejora en la atención al cliente usando datos de Twitter y técnicas de aprendizaje automático

---

Sofia Nazarena Di Buccio

Director
Carlos Diuk

Co-director
Augusto Villa Monte

Resumen

El comercio electrónico ha experimentado una aceleración sin precedentes en el último tiempo. Por su parte, las empresas han intentado adaptarse para satisfacer la creciente demanda de consumo brindando, en su mayoría, atención al cliente bajo una modalidad de "autoservicio". De hecho, varias de ellas han utilizado redes sociales como medio de soporte en pos de construir una imagen de disponibilidad e inmediatez. No obstante, los agentes acaban recibiendo los problemas más complejos, dejando en evidencia la falta de capacitación y herramientas provistos para la tarea. A lo largo de este trabajo, se diseña un tablero de negocio para dar visibilidad del estado de situación respecto a la atención brindada así como, también, para poder responder ante distintos casos de atención al cliente de forma *data-driven* en un marco de recursos limitados. Se hace foco, particularmente, en Mercado Libre ya que cuenta con un volumen considerable de transacciones y brinda soporte vía Twitter. Recolectando datos de esta fuente, se explora y modela la probabilidad de requerir atención al cliente, la satisfacción del usuario, las palabras clave del texto, los tópicos y la ubicación del *tweet*. Para ello, se aplican técnicas de aprendizaje supervisado como no supervisado, se emplean ensambles y se utilizan expresiones regulares.

UNIVERSIDAD TORCUATO DI TELLA

Master in Management & Analytics

---

# Enhancing customer service using Twitter data and machine learning techniques

---

Sofia Nazarena Di Buccio

Advisor
Carlos Diuk

Co-Advisor
Augusto Villa Monte

Abstract

Online commerce has recently experienced an unprecedented acceleration. For their part, companies have tried to adapt to meet the growing consumer demand by providing, for the most part, customer service under a "self-service" modality. In fact, several of them have used social networks as a means of support in order to build an image of availability and immediacy. However, the agents end up receiving the most complex problems, revealing the lack of training and tools provided for the task. Throughout this work, a business dashboard is designed to give visibility of the status of the service provided, as well as to be able to respond to different customer cases in a data-driven manner within a limited resources framework. The focus is particularly on Mercado Libre since it has a considerable volume of transactions and provides support via Twitter. Collecting data from this source, it is explored and modeled the probability of requiring customer service, user satisfaction, the keywords of the text, the topics and the location of the tweet. For this, supervised and unsupervised learning techniques are applied, ensembles are designed and regular expressions are used.

# Índice



# Índice de Tablas

# Índice de Figuras

# 1. Introducción

## 1.1 Motivación

El comercio de bienes y servicios en Internet, potenciado por el continuo desarrollo de tecnologías y el aumento en la seguridad de los pagos *online*, crece de manera exponencial todos los años (Beetrack, 2021). Actualmente, se estima que 2.14 mil millones de personas compran en línea, lo que representa aproximadamente una cuarta parte de la población mundial (Coppola, 2021). Mientras que un 92% de los compradores realiza una transacción *online* al año, un 67% lo hace una vez al mes, un 25% una vez por semana y un 4% una vez al día (Orús, 2022).

La modalidad de compra virtual es aquella preferida por muchos de los consumidores generando, sin lugar a dudas, un punto de inflexión para los negocios. La compra *online* ha dejado de ser un comportamiento excepcional y se ha convertido en un hábito para poder resolver necesidades diarias (Cámara Argentina de Comercio Electrónico, 2021).

La aparición de un tipo de cliente dispuesto a comprar de forma remota, incentivado o no por una necesidad concreta, ha puesto en evidencia el diferencial que cada empresa debe ofrecer para poder retener clientes, más aún cuando estos se encuentran empoderados por la disponibilidad de información a tan solo un click de distancia.

La lealtad de los consumidores y su comportamiento dependen fuertemente de la experiencia que hayan tenido previamente con el negocio (Salesforce, 2020). Además, según Walker (2017), la atención al cliente se ha convertido en un factor tan influyente como el precio en la decisión de adquirir un producto o servicio.

Es por estos motivos que, hoy en día, el soporte brindado durante el proceso de compra se ha vuelto una potencial fuente de ingreso para los comercios electrónicos. De hecho, las compañías que priorizan dar una buena experiencia al consumidor generan un 66% más de ingresos que aquellas que no (Walker, 2017).

Las empresas han optado por utilizar tecnologías que permitan brindar un soporte del estilo "autoservicio": se espera que los clientes puedan resolver la mayoría de las dificultades durante el proceso de compra con asistencia virtual automatizada (DeLisi et al., 2021).

Si bien es verdad que el 81% de los consumidores prefieren resolver los obstáculos por sí solos antes que hablar con un agente (Zendesk, 2019), el enfoque adoptado ha generado que los problemas más complejos recaigan en representantes con escasas herramientas para resolver conflictos de forma rápida y personalizada.

Al no implementar cambios ni invertir en la capacitación del personal, las empresas han dado lugar a que se genere una brecha entre la atención que los clientes esperan y la que obtienen, resultando en una estable caída de satisfacción en diferentes industrias (DeLisi et al., 2021).

### 1.1.1 Uso de redes sociales

Aproximadamente un 60% de los consumidores esperan que las empresas respondan de forma inmediata ante un conflicto en el proceso de compra (Walker, 2017). Frente a esta necesidad, las empresas han optado por brindar atención al cliente por vías como Twitter, Instagram y Facebook.

Esta apertura de múltiples canales de comunicación permite alcanzar un mayor grupo de consumidores en comparación a los medios tradicionales, contribuyendo a que la empresa se muestre disponible y consciente de los problemas que la vinculan (Commbox, 2021).

No obstante, las redes sociales se utilizan para múltiples propósitos como, por ejemplo, para generar lealtad en los consumidores y dar lugar al reconocimiento de marca, promoviendo interacción social para influir de forma positiva en las ventas (Caramela, 2020). Esto último suele denominarse retorno de inversión social.

### 1.1.2 Uso de inteligencia artificial

Parte de mejorar la atención que recibe el usuario se ha basado, principalmente, en la recopilación de grandes volúmenes de datos acerca de los clientes para luego poder interpretarlos y extraer información que permita ofrecer un mejor servicio durante el proceso de compra.

En este contexto, se introduce el aprendizaje automático — una de las tantas aplicaciones de la inteligencia artificial (IA) — como una herramienta que permite modelar datos crudos en busca de patrones observables para mejorar la calidad de la asistencia brindada al cliente.

Las empresas son conscientes de la existencia de estas tecnologías e, incluso, muchas han apostado a ellas dado que permiten brindar una mejor atención. De esta forma, se hace posible generar una personalización del servicio y anticipar las necesidades de los clientes en el negocio de forma proactiva (Chen, 2020).

Varias técnicas de aprendizaje automático ya son observables en el dominio de estudio. Múltiples aplicaciones han sido desarrolladas para poder ayudar a brindar soporte de forma inteligente teniendo en cuenta las interacciones generadas en las redes sociales.

Los *features* más frecuentes incluyen: monitoreo de conversaciones y menciones (muchas veces en tiempo real con *stream listening*), análisis de sentimiento (satisfacción del consumidor e imagen generada), análisis de competencia, respuestas automatizadas utilizando palabras clave (*bots*) y características de *engagement* de los posteos.

Las herramientas existentes, en su mayoría, intentan penetrar el flujo de trabajo de los representantes de atención al cliente, embebiendo y centralizando los canales de comunicación más importantes en una única interfaz para poder dar un seguimiento adecuado a los reclamos con tickets y chats en vivo de forma integrada.

Las soluciones en el mercado suelen incorporar, también, la opción de crear reglas duras de negocio para priorizar casos en el proceso de encolado. Generalmente, se utilizan características del usuario para dar visibilidad sobre la actividad y la influencia del mismo. En el caso particular de Twitter, por ejemplo, se refiere comúnmente al número de seguidores y mensajes publicados.

## 1.2 Contribución

En este proyecto, se propone identificar y medir de forma probabilística la necesidad que cada usuario posee en recibir atención personalizada. Este *feature,* aún no desarrollado por las aplicaciones actuales, permitiría realizar una alocación de representantes (recurso limitado) de forma *data-driven*.

Para acotar el alcance, se adapta la propuesta descrita a Mercado Libre. Dicho ecosistema de soluciones integradas posee, dentro de sus unidades de negocio, un *marketplace* donde se llevan a cabo transacciones de forma electrónica entre vendedores y compradores de productos o servicios (MELI, 2021). Respecto a sus operaciones en 2021, se observa una base de 139.5 millones de usuarios así como, también, alrededor de 1014.3 millones de productos vendidos (Compte, 2022). Dado que esta empresa otorga soporte por medio de redes sociales, se hace foco en la atención al cliente que brinda en Twitter con la cuenta @MLAyuda.

Utilizando como input principal el *corpus* de *tweets* vinculados a Mercado Libre, se aplican técnicas de aprendizaje automático como modelos de clasificación supervisados y *embeddings* para calcular la probabilidad de que un comprador necesite atención personalizada.

Además, se modelan características del *corpus* que permiten entender mejor las necesidades de los usuarios y el *feature* de interés. Estas incluyen: la satisfacción del cliente, los tópicos del *tweet*, la ubicación del usuario y las palabras clave (*keywords*) halladas en el texto. Para estos casos, se investigan tanto técnicas de aprendizaje no supervisado como supervisado, se hace uso de ensambles y se definen heurísticas con expresiones regulares. Al desarrollar estas variables para una empresa en particular, en algunos casos fue necesario etiquetar los datos para construir las características previamente mencionadas.

Todos estos *features* se consolidan en un tablero que también contiene atributos propios de los *tweets* (por ejemplo: favoritos y re-posteos). Dicha herramienta se diseña para el uso exclusivo del negocio, dando lugar a la exploración de la data en pos de generar accionables que mejoren el servicio de atención al cliente brindado.

## 1.3 Organización

Respecto a la estructura del documento, se detallan los métodos y procedimientos utilizados para el análisis en la Sección 2. Estos incluyen la recolección, la exploración y el modelado de los datos.

Los resultados obtenidos son consolidados en la Sección 3, donde se introduce un tablero diseñado específicamente para que el negocio pueda tomar decisiones orientadas en datos. Con el objeto de entender las potenciales aplicaciones de la herramienta, se expande sobre cada uno de los componentes ejemplificando al menos una modalidad de uso.

La Sección 4 presenta una discusión acerca de los logros y las limitaciones encontradas a lo largo del análisis, recomendando mejoras para futuras iteraciones. El documento finaliza con una serie de apéndices que dan mayor detalle sobre la Sección 2 y, por último, la bibliografía.

# 2. Método y Procedimiento

Este apartado detalla la recolección, exploración y modelado de los datos. Para asegurar la reproducibilidad de los resultados, el código se encuentra disponible en un repositorio de Github[1]. Durante el desarrollo, se programó en lenguaje *Python* utilizando Jupyter Notebooks como entorno de ejecución.

## 2.1 Recolección de datos

Los datos utilizados para el desarrollo de este proyecto fueron recolectados usando *tweepy*[2], un *wrapper* de la API[3] de Twitter que facilita el consumo de la información disponibilizada por la red social. Con el propósito de realizar *requests* a los distintos *endpoints* de la API, se gestionó una cuenta de desarrollador. Esto se hace a través de un formulario[4] online donde se especifica, entre otras cuestiones, el caso de uso de los datos. Una vez creada la cuenta, se pueden generar las credenciales necesarias para autenticar los *requests* – una *key,* un *secret* y un *token* – en el portal web de desarrolladores.

Para obtener *tweets* que pudiesen estar relacionados a Mercado Libre, se generan dos consultas parametrizadas teniendo en cuenta: por un lado, las menciones a @ML_Ayuda y @Mercadolibre y, por otro lado, las ocurrencias de palabras clave que vincularan a la empresa en ausencia de menciones explícitas. En este último caso, se consideran posteos donde se pueda leer "mercadolibre" y alguna de las siguientes palabras: "compra", "venta", "vendedor", "seguimiento", "envío", "pago" y "correo".

Estas pautas de recolección fueron el resultado de varias iteraciones que permitieron ajustar los parámetros de las consultas. Si bien la búsqueda de palabras clave es acotada, el objetivo de dicho criterio fue poder salvar casos evidentes en los que se estuviese hablando de la empresa sin mencionarla, más aún sabiendo que la gran mayoría (84.39%) de los *tweets* se recopilan bajo la norma de menciones.

Una vez almacenados los *tweets* de interés con su correspondiente metadata, se procura obtener el texto completo (sin truncar) para cada uno de ellos. Además, se reúne información acerca de las cuentas seguidas (*following*[5]*)* por los usuarios emisores así como las que siguen (*followers*) a estos últimos.

Dado que el acceso a estos datos se obtuvo bajo permisos limitados, sólo fue posible acceder a los *tweets* de los últimos 7 días respecto a la fecha de consulta con cuota de *requests* por hora. Para poder asegurar una muestra variada, se ejecutó múltiples veces la recolección con intervalos de una a dos semanas para evitar el *overlay* de datos.

Con esta metodología, se obtuvo una muestra de 35,481 *tweets.* Para más detalles acerca de las fechas de recolección y las variables relevantes del dataset, referirse al Apéndice A.

---

[1] El repositorio mencionado se halla en https://github.com/sdibuccio/intelligent_customer_service
[2] Referencias a la documentación en https://www.tweepy.org/
[3] Documentación de la API (v2) en https://developer.twitter.com/en/docs/twitter-api
[4] El formulario se encuentra en https://developer.twitter.com/en/portal/petition/essential/basic-info
[5] Notar que los los seguidos (*following)* se denominan amigos (*friends)* en la documentación de la API.

## 2.2 Exploración

En esta sección, se identifican y se extraen las variables que podrían ser relevantes para el problema. Dado que se recopilan características tanto de los usuarios como de los *tweets*, se analiza cada conjunto por separado para realizar una exploración de los datos.

### 2.2.1 Tweet Data

Se cuenta con 35,481 *tweets* en idioma español. Dado que un 3% de estos se han guardado múltiples veces por actualizaciones de la metadata, se opta por retener los registros más recientes. También se descarta un 13.6% de mensajes por ser *retweets* que no aportan información adicional, guardando el número de veces que fue retuiteado un mensaje.

A continuación se investigan distintas características del *corpus* constituido por 30,662 mensajes. Estas son: los retuits, los favoritos, la ubicación donde se emite cada *tweet*, las menciones a distintas cuentas, las respuestas entre mensajes y los *hashtags*. Además, se exploran las palabras más frecuentes en la muestra como una primera aproximación a comprender el contenido del texto, identificando distintas dificultades ligadas al lenguaje utilizado en Twitter.

Para comenzar, se observa el número de retuits y de favoritos. Estas variables valen cero en un 97.11% y 89.01% de los casos respectivamente. En aquellas ocasiones donde esto no es cierto, se observan comúnmente cifras inferiores a 10 y, sólo en algunas instancias puntuales, valores altos.

Refiriendo a la dimensión geográfica de los datos, se observa que la ubicación del mensaje posee un alto porcentaje de campos faltantes o nulos. De hecho, sólo se da a conocer el país donde se emitió el *tweet* para un 2.9% de los mensajes. Analizando dicho subset de casos, se encuentra que la mayoría se concentran en Argentina, México, Chile y Colombia.

Considerando las menciones de los *tweets*, aproximadamente un 67% se vinculan a cuentas de Mercado Libre (ver Figura 1 para más detalle) y parecen perseguir el objetivo de obtener soporte. Vale la pena mencionar que un 73.1% de los *tweets* son respuestas a otros posts y, en su mayoría, contestan a @ML_Ayuda (52.1%) o a @Mercadolibre (4.7%). Esto no es sorpresivo, ya que es el resultado de la metodología de recolección utilizada.

También es posible observar menciones a cuentas no relacionadas a Mercado Libre. Muchas de ellas prestan servicios de atención al cliente (por ejemplo, DigitalHouse[6] y Globant[7]), otras representan la defensa al consumidor (por ejemplo, Profeco[8] y BAConsumidor[9]), o son más bien cuentas independientes que realizan publicaciones masivas. La Figura 2 da más detalles sobre la frecuencia de menciones para dichas cuentas.

---

[6] Referir a https://twitter.com/_digitalhouse para el perfil correspondiente en Twitter.
[7] Referir a https://twitter.com/Globant para el perfil correspondiente en Twitter.
[8] Referir a https://twitter.com/Profeco para el perfil correspondiente en Twitter.
[9] Referir a https://twitter.com/BAconsumidor perfil correspondiente en Twitter.

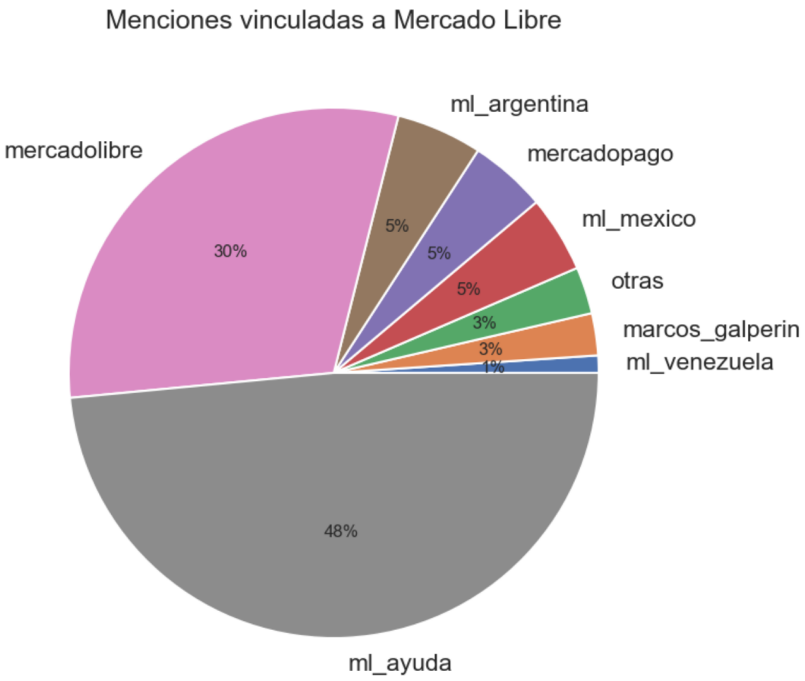


Figura 1. Composición de menciones vinculadas a Mercado Libre por usuario. Si la cuenta tiene menos del 1% de menciones, se la agrupa en "otras".

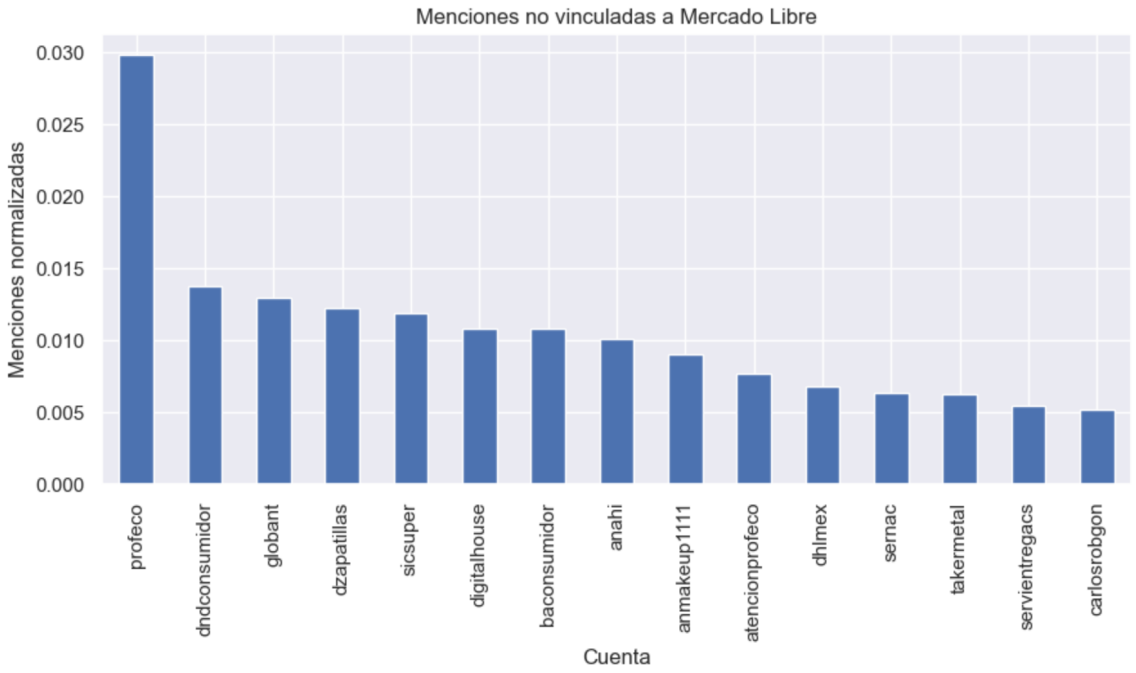


Figura 2. Menciones no vinculadas a Mercado Libre con más del 0.05% de apariciones en la muestra.

Respecto a los hashtags presentes en la muestra, los tópicos cubiertos son bastante variados. La Figura 3 muestra las apariciones más frecuentes de un total de 1,110 hashtags existentes. En dicha figura, es posible observar noticias relacionadas con el COVID-19, estafas en compras por internet, publicaciones de productos para venta y otros usos que fuera de contexto son difíciles de interpretar.

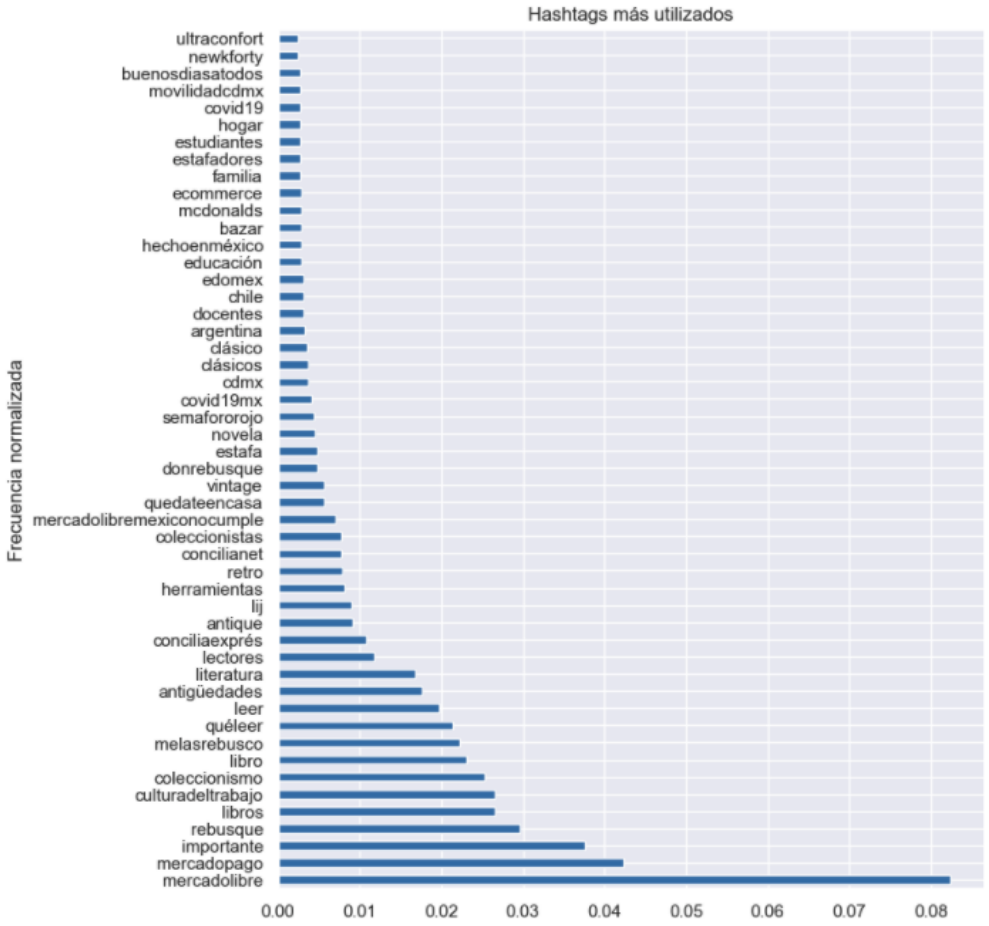


Figura 3. Top 50 hashtags más utilizados en la muestra.

Como último paso de esta fase exploratoria, se contabilizan las palabras más frecuentes en la muestra para obtener una intuición acerca del contenido de los *tweets*. Con este propósito, se limpia y procesa el texto de acuerdo a la metodología descrita en el Apéndice B. Este método reduce la cantidad de palabras en aproximadamente un 61%, reteniendo un promedio de 20 palabras relevantes por *tweet* con un piso mínimo de 5 y un máximo de 31. Observando la Figura 4, se nota que aquellas palabras con mayor cobertura en la muestra parecen reflejar rigurosamente el dominio de atención al cliente.


Figura 4. Nube de palabras creada a partir de los *tokens* de cada *tweet*.

Una cuestión a notar es que el lenguaje varía en función de la red social utilizada (Candale, 2017). En el caso particular de Twitter, los usuarios comparten ideas e información en mensajes que no pueden superar los 280 caracteres. Por este motivo, se observa un uso excesivo de acrónimos, acortamientos de palabras y falta de signos de puntuación. También se ven afectadas las estructuras de las oraciones y, muchas veces, se mezclan distintos idiomas en una misma palabra.

Según Farzindar & Inkpen (2015), los emisores en Twitter usan un tono informal y conversacional para comunicar sus pensamientos en forma de corriente de conciencia (*stream of consciousness*). Esto incentiva la interacción entre múltiples usuarios, dando lugar a frecuentes cambios de tema. El texto no solo se vuelve dinámico y continuo sino que, también, ambiguo dada la falta de información contextual.

Si bien esta sección no pretende profundizar en el procedimiento de tokenización, es relevante mencionar que muchos de los desafíos encontrados provienen de las características propias del texto de Twitter. A estos desafíos se añade, también, el número limitado de herramientas disponibles para trabajar el *corpus* en español en comparación al idioma inglés.

### 2.2.2 User Data

El dataset recopilado contiene 9,330 usuarios diferentes. Para comprender mejor los datos disponibles se describe la ubicación, el número de posteos emitidos y, también, el número de seguidores y seguidos. Además, se investiga la calidad del contenido provista por cada usuario, buscando correlaciones con el número de seguidores y la cantidad de *tweets* emitidos.

Respecto a los seguidores de cada usuario, se observa una distribución con alta variación donde un 50% de las cuentas tienen menos de 100 seguidores. Si bien la distribución de esta última variable posee variación, el valor es más bajo respecto a aquel observado para los seguidores. La Tabla 1 amplía sobre estadísticas de estas distribuciones.

| MÉTRICA | SEGUIDORES DEL USUARIO | SEGUIDOS DEL USUARIO |
|---|---|---|
| MIN | 0 | 0 |
| P25 | 14 | 69 |
| P50 | 95 | 241 |
| P75 | 414 | 660 |
| P80 | 603 | 840 |
| P90 | 1719 | 1560 |
| MAX | 9,916,088 | 179,145 |
| DESVÍO | 99,758.87[10] | 5,061 |

Tabla 1. Tabla de estadísticas descriptivas para los seguidores y seguidos del usuario.

[10] Nótese que el desvío se encuentra directamente afectado por el valor máximo de la distribución (en el orden de los millones). Para referencia, el desvío computado excluyendo dicho valor es 26,039.89.

El campo de ubicación geográfica es de texto libre y presenta un nivel de complejidad de consumo diferente que el resto de las variables. Por este motivo, se lo analiza en profundidad en la Sección 2.3.1.

La interacción de los usuarios con Mercado Libre varía notablemente a lo largo de los 49 días cubiertos por la muestra. Aproximadamente un 99% de los usuarios postean menos de 16 veces. Exponiendo la cantidad de mensajes generados por este subset de cuentas (Figura 5), se observa que la mayoría de los emisores generan hasta dos *tweets.* De hecho, la mediana corresponde al posteo de un único *tweet.*

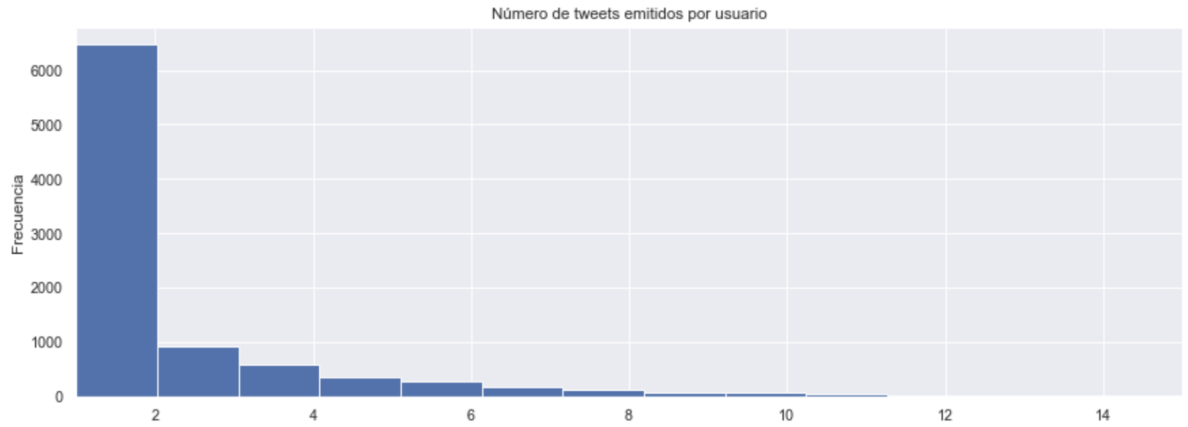


Figura 5. Histograma de la cantidad de *tweets* encontrados para aquellos usuarios que generan menos de 16 posteos relacionados a Mercado Libre.

Respecto al 1% de las cuentas restantes, algunas persiguen el objetivo de brindar soporte mientras que otras utilizan la red social para generar ventas, *i.e.*, los *tweets* se vuelven publicaciones de productos y servicios. La Figura 6 muestra el comportamiento para dichas cuentas excluyendo @ML_Ayuda y @mercadopago ya que cubren el 18% del subset relevado.

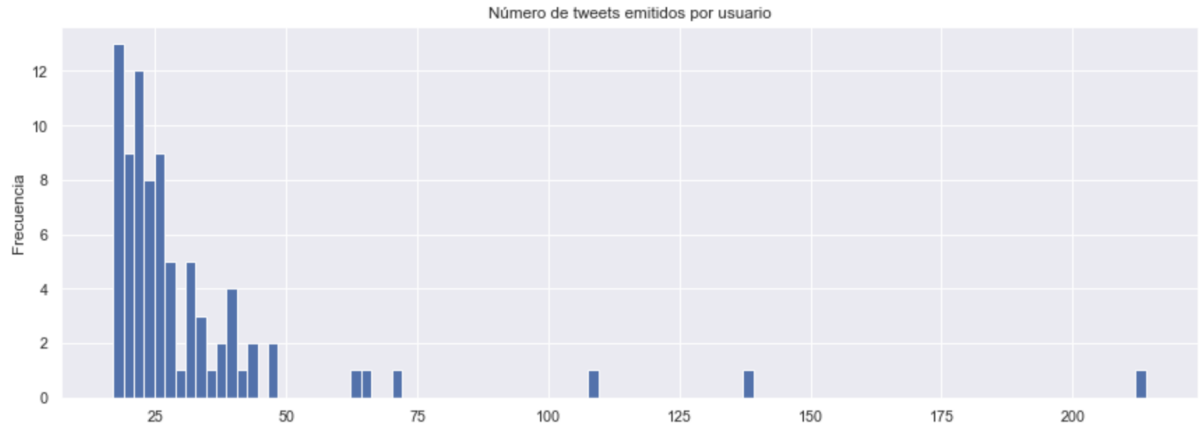


Figura 6. Histograma de la cantidad de *tweets* encontrados para aquellos usuarios que generan más de 16 posteos relacionados a Mercado Libre.

Para continuar, se intenta evaluar la calidad del contenido que proveen los usuarios que generan más de un *tweet*. Esto surge de haber encontrado varias cuentas que publican textos prácticamente idénticos y con leves modificaciones. La idea es identificar y eliminar dichos *tweets* ya que no agregan valor al *corpus*.

Para evaluar la calidad del contenido, se utiliza el método descrito en el Apéndice C sobre el 48% de los usuarios que cumplen con el requerimiento mínimo de posteos. El proceso permite identificar un 42.13% de usuarios que generan al menos un *tweet* de baja calidad, afectando un 35% de los textos del *corpus*. En promedio, una cuenta genérica postea 4 textos irrelevantes, lo que representa un 37% de sus publicaciones totales.

La Figura 7 muestra la cantidad de *tweets* totales y aquellos identificados como poco relevantes para cada cuenta categorizada por número de seguidores. Dicha gráfica excluye a @ML_ayuda y a @mercadopago — que publican un 78.08% y un 64.40% de textos irrelevantes respectivamente — porque poseen alta cobertura y vuelven compleja la visualización en la escala seleccionada.

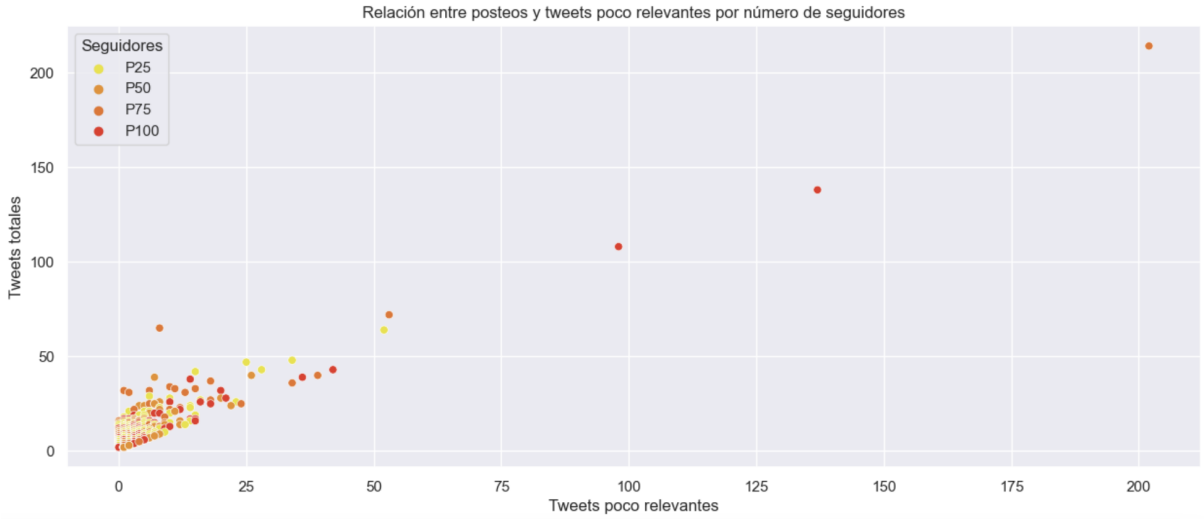


Figura 7. Gráfica de dispersión entre tweets totales y tweets poco relevantes, donde el color de cada punto representa el percentil al que pertenece cada cuenta considerando el número de seguidores. Se excluye @ML_Ayuda y @mercadopago.

Analizando la relación entre la cantidad de *tweets* generados y aquellos identificados como poco relevantes por el proceso, el coeficiente de pearson toma un valor de $r = 0.91$ significativo al 1%. Esto permite afirmar que existe una correlación directa fuerte entre las variables en cuestión.

Además, se considera la relación entre la cantidad de seguidores de cada cuenta y la calidad del contenido generado. Si bien se evalúa la posibilidad de que exista una relación no lineal para dichas variables, no se encuentra evidencia concluyente. Tampoco se pudo establecer una relación entre la cantidad de posteos totales y los seguidores de cada cuenta. Este comportamiento podría explicarse teniendo en cuenta que sólo se analiza un subconjunto de *tweets*, *i.e,* aquellos que poseen alguna relación con Mercado Libre.

## 2.3 Modelado

Esta sección desarrolla la construcción de características relevantes del *corpus* que, más adelante, habilita una toma de decisiones basada en datos. Se busca modelar inicialmente la necesidad de requerir atención personalizada para cada cliente pero, también, se generan otras variables de interés que complementan y enriquecen dicho aporte.

### 2.3.1 Inferencia del país de origen

Se comienza por inferir el país de origen de cada mensaje incluido en el *corpus*. Aunque esta información debería ser inherente a cada *tweet*, en la sección 2.2 se ha observado que sólo un 1.89% de los textos poseen este dato. Por este motivo, se diseñan reglas que intentan localizar cada mensaje de la muestra en alguno de los 18 países[11] donde opera Mercado Libre utilizando la información del usuario y el contenido de cada *tweet*.

Antes que nada, se genera una tabla que posee las ciudades y estados (provincias) de cada país cubierto por la empresa en cuestión. Un 8% de las ciudades se repiten en más de un país y un 4% en más de una provincia. En el caso de encontrar información que localice un *tweet* en alguna de estas ciudades con repetición, no se podrá inferir su país con certeza. Como regla general, las coincidencias exactas prevalecerán por encima de las que no lo son.

Con esta tabla de referencia se intenta interpretar el campo de ubicación del usuario que se encuentra disponible en un 61.4% de los casos. Dado que dicho campo es de texto libre, se limpia de forma básica (caracteres especiales, mayúsculas, espacios y puntuación) cada una de las palabras para poder procesarlas.

Una vez normalizado el campo de ubicación del usuario, se compara con aquellos valores guardados en la tabla de referencia. Se realiza una búsqueda secuencial, teniendo en cuenta los nombres de los países, de los estados y de las ciudades registradas, en ese orden de prioridad. Dicho proceso se interrumpe si, en la categoría considerada, se encuentra una coincidencia que hace posible inferir el país al cual pertenece el *tweet*. Con este proceso, la proporción de *tweets* con ubicación crece de 1.89% a 48.4%.

En muchos casos, se observa que el campo de ubicación posee abreviaciones y faltas de ortografía que dificultan el proceso de coincidencia ya descrito. Un usuario que escribe su ubicación como "cdad de bs as", por ejemplo, queda fuera del alcance de la búsqueda exacta a pesar de que se entiende que se refiere a Argentina.

Para obtener cobertura sobre estos casos, se definen lógicas difusas (*fuzzy*) que evalúan la ubicación del usuario con las particiones territoriales de la tabla de referencia y, también, con algunas abreviaciones frecuentes. Se calcula para cada par de textos comparados un puntaje de similitud basado en una función robusta al orden de las palabras y que puede lidiar con coincidencias parciales. Así se infiere la ubicación del *tweet* cuando existe un único país cuyo porcentaje de similitud supere el 70%. Esta metodología permite aumentar la proporción de *tweets* con ubicación a un 52.35%.

Respecto al remanente de *tweets* sin ubicación, se hace uso del texto del mensaje para intentar identificar la ubicación. Nuevamente, se normalizan las palabras para poder realizar comparaciones. Se empieza por buscar coincidencias de nombres de países en el texto del mensaje, sólo asignando ubicaciones cuando un único país resulta de la evaluación. El porcentaje de *tweets* con data de ubicación, incorporando esta técnica, alcanza un 61.85%.

[11] México, Guatemala, El Salvador, Honduras, Nicaragua, República Dominicana, Costa Rica, Panamá, Colombia, Venezuela, Ecuador, Brasil, Perú, Paraguay, Bolivia, Argentina, Uruguay y Chile.

Aplicando la misma lógica *fuzzy* introducida previamente para la búsqueda de países, la proporción final de *tweets* con ubicación alcanza un 62.16%, representando unos 19,062 mensajes en total.

### 2.3.2 Extracción de términos clave

Una forma de caracterizar los *tweets* (o documentos) del *corpus* es identificando términos clave que representan su contenido. En general, los algoritmos utilizados con este fin seleccionan potenciales candidatos y, en función de propiedades calculadas para cada uno de ellos, se genera un puntaje que permite ordenarlos. El conjunto final de términos clave se define, comúnmente, utilizando un punto de corte (*threshold)*.

En este apartado, se explica tanto la elección como el funcionamiento de cada algoritmo empleado para obtener las palabras clave de cada *tweet*. Además, se detalla el preprocesamiento del texto que, en algunos casos, depende del método de extracción elegido. Luego, se genera una heurística para poder combinar los candidatos obtenidos con cada método, creando un puntaje que permite retener el término más frecuente por cada *tweet*.

Los métodos de extracción pueden ser tanto supervisados como no supervisados y, además, pueden estar basados en grafos o en cálculos estadísticos (Godec, 2021). Dado que no se cuenta con un dataset etiquetado en función de términos clave, se limita el alcance de esta sección haciendo uso de algoritmos no supervisados. Dentro de dicha categoría, se consideran métodos populares con implementación en *Python* que no dependen de un dominio ni de un idioma en particular.

Con este criterio, se seleccionan los siguientes métodos: RAKE (*Rapid Automatic Keyword Extraction*), YAKE (*Yet Another Keyword Extractor)* y TextRank. Los primeros dos se basan en cálculos estadísticos mientras que el tercero basa su cómputo en grafos. A pesar de haber buscado algoritmos diseñados específicamente para textos cortos, muchos de ellos no tienen el código fuente disponible para ser utilizado o bien son de carácter supervisado. De todas maneras, los tres métodos seleccionados han demostrado ser útiles para la extracción de términos clave en *tweets* (Farinha, 2018).

Respecto al preprocesamiento del texto, en esta ocasión, se parte de 21,467 *tweets* que han sido caracterizados como relevantes en la fase exploratoria del usuario (Sección 2.2.2). Primero se normaliza el texto removiendo caracteres especiales y acentos, luego se quitan menciones, emails, dígitos, hashtags, links y *stopwords*. Sólo se procede a buscar términos clave para aquellos *tweets* que después del procesamiento retienen al menos 10 palabras, lo que deja un 74.84% del universo inicial.

Considerando las mayúsculas, sólo serán retenidas al aplicar YAKE puesto que construye propiedades en función de esta característica. Nótese que los signos de puntuación no son tratados de ninguna forma. Esto se debe a que cada algoritmo le da un uso particular como detallaremos más adelante.

Si bien cada uno de los algoritmos seleccionados intenta lidiar con la existencia de *stopwords,* se decide no depender enteramente de dichas técnicas y, en su lugar,

preprocesar el texto previamente. El motivo por el cual esta decisión fue tomada es porque las métricas utilizadas para identificar las palabras en cuestión se basan en la frecuencia y, al estar trabajando con textos cortos, puede que no sea lo más adecuado.

A continuación se describe brevemente el funcionamiento de cada uno de los algoritmos seleccionados. El objetivo es poder comprender la forma en la que cada uno: (i) genera el conjunto de potenciales candidatos a términos clave, (ii) ordena dicho conjunto y, (iii) selecciona los términos clave por documento.

Se comienza introduciendo RAKE, un método estadístico que opera en cada documento de forma individual, siendo lo suficientemente flexible para aplicarse sobre distintos dominios así como, también, sobre colecciones dinámicas que no siguen una convención en particular (Rose et al., 2010). Este algoritmo utiliza los signos de puntuación del texto para poder delimitar las palabras y frases candidatas a ser relevantes. Además, el número de palabras que constituyen un término clave se encuentra parametrizado.

Siguiendo a Rose et al. (2010), se trabaja con cada documento particular para extraer las palabras que lo constituyen, generando a partir de estas una matriz de co-ocurrencias. Así, se da a conocer la cantidad de veces que cada palabra aparece en conjunto con otras (grado) y, también, su frecuencia dentro del documento. Se continúa generando un set de potenciales candidatos a ser términos clave que pueden componerse por una o más palabras adyacentes. Tomando el ratio entre el grado y la frecuencia, se crea un puntaje para cada palabra existente en la matriz. Luego, cada candidato obtiene la suma de los puntajes de las palabras que lo componen.

Haciendo uso del ratio definido anteriormente, se procede a ordenar de forma descendente el set de candidatos a términos clave de cada documento. Según los autores (Rose et al., 2010) es conveniente retener el primer tercio de los candidatos con mayor puntaje para obtener los términos clave de cada documento del *corpus*.

En la misma línea, se puede detallar el funcionamiento de YAKE, otro método de carácter estadístico que manipula sólo los datos contenidos dentro de los documentos (Campos et al. 2020). A diferencia de RAKE, este procedimiento incluye el uso de las mayúsculas para delimitar el conjunto de potenciales términos relevantes de cada texto.

Respecto a la generación de *features*, YAKE genera propiedades relacionadas a la frecuencia de los términos pero, también, crea variables que cubren el uso de mayúsculas, la ubicación dentro del documento, la aparición en distintas oraciones y, por último, las relaciones entre palabras. El Apéndice D amplía en profundidad cada uno de dichos cálculos.

Cada una de las variables mencionadas se ponderan en un único puntaje para poder reflejar la importancia de cada candidato. Se le otorga peso a cada uno de los *features* y, también, se normaliza por la frecuencia para evitar distorsionar el puntaje de candidatos largos (ver ecuaciones 8, 10 y 11 del Apéndice D).

Una vez obtenido el puntaje final, se ordenan los candidatos de forma ascendente y se calcula la similitud para cada par en la lista. Cuando el valor de la similitud excede un determinado punto de corte, se retiene sólo el candidato más relevante según el puntaje.

El último algoritmo seleccionado, TextRank, ejecuta la extracción de términos clave utilizando grafos (Mihalcea & Tarau, 2004). La importancia de cada palabra se determina de forma global teniendo en cuenta las relaciones entre vértices: se pondera tanto la cantidad de votos obtenidos (*votes in*) como los generados (*votes out*).

Partiendo de un documento de texto cualquiera, el método en cuestión genera los *tokens* correspondientes y los anota con *part-of-speech tagging* (POST). Dicho proceso permite etiquetar las palabras en función de su significado sintáctico o gramatical. Con una ventana de palabras determinada, se crea una matriz de co-ocurrencia y se inicializan los puntajes de cada vértice en 1. Luego, se itera el cálculo del vértice (ver ecuación 12 del Apéndice D) hasta encontrar convergencia respecto de un *threshold*. En última instancia, se ordenan los candidatos y se toman los primeros N como términos clave.

Rememorando que se busca sintetizar el contenido de cada *tweet*, se utiliza el set de términos clave obtenidos mediante las tres metodologías detalladas para crear una heurística que retenga aquel término de mayor frecuencia por *tweet*. La idea es construir un *output* que permita contextualizar el mensaje rápidamente y que dé lugar a distintos análisis descritos más adelante durante la consolidación de *features* (Sección 3).

Se propone generar para cada algoritmo un puntaje para el mejor término rankeado, comparando dicho término con las mejores tres nominaciones del resto. Para ello, se evalúa la existencia de palabras en común entre pares de términos clave: se retorna 1 si la intersección no es vacía, sino se retorna 0.

Siendo $kw_{n,\,a}$ el n-ésimo término clave sugerido por el algoritmo ***a*** y $compare$ la función de la comparación entre términos descrita, el puntaje se calcula con la siguiente fórmula.

(1) $$score\,(kw_{1,\,a}) \;=\; \sum_{j \neq a} \sum_{n=1}^{3} compare\,(kw_{1,\,a},\, kw_{n,\,j}) \qquad a,\, j \in \{rake,\, yake,\, textrank\}$$

De esta forma, se obtendrá un total de tres términos clave por *tweet*, uno por cada algoritmo considerado, con sus respectivos puntajes de validación. Seleccionaremos aquel candidato de mayor puntaje, siempre que sea superior a cero, para representar el contenido del documento.

Utilizando esta heurística, es posible identificar un término clave para un 85.59% de los *tweets* (13,429). Observando la Figura 8, RAKE genera la mayor cantidad de términos clave validados en comparación con YAKE y TextRank, siendo capaz de identificar una sugerencia para un 47.78% de los documentos considerados.

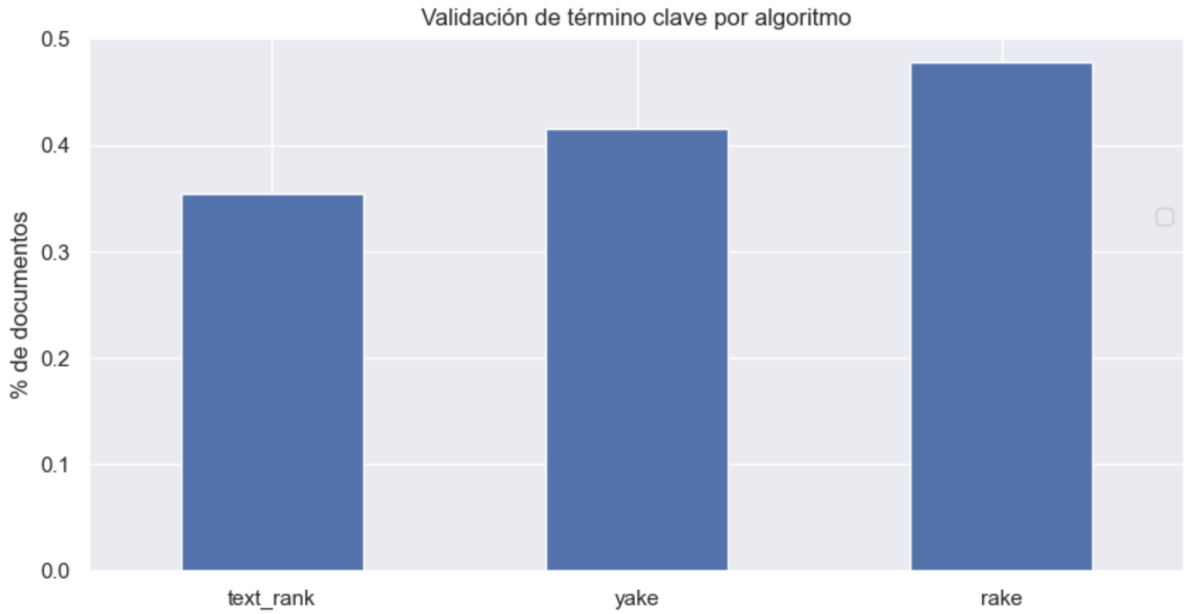


Figura 8. Gráfica de barras para el porcentaje de documentos con términos clave por algoritmo. Nótese que si múltiples algoritmos generan el mismo puntaje en un documento cualquiera, dicho documento será contado más de una vez (no excluyente). Por ese motivo la suma de los porcentajes es superior al 100%.

Si bien parece que RAKE genera una mayor cantidad de términos clave validados, es importante observar la distribución de los puntajes correspondientes para entender en profundidad la calidad de las sugerencias. En la Figura 9 puede observarse que a pesar de que la mediana de las tres distribuciones es idéntica ($P50 = 1$), en el caso de RAKE, el percentil 75 es levemente más alto ($P75 = 2$).

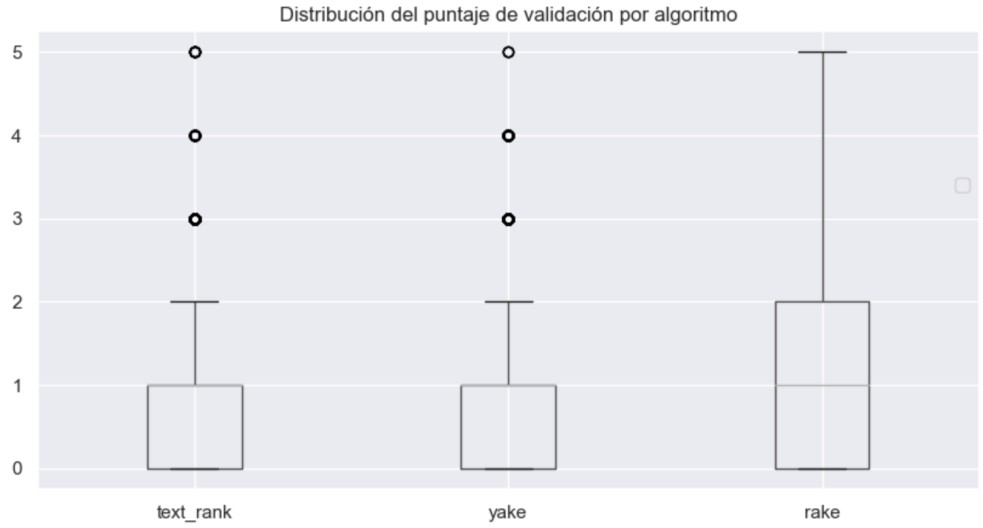


Figura 9. Diagrama de cajas para el puntaje generado por cada algoritmo. Nótese que el puntaje toma un mínimo de 0 y un podría alcanzar un máximo de 6.

Sólo se retendrá para cada *tweet* el término clave de mayor puntaje. La Figura 9 permite analizar el comportamiento de dicho puntaje. Nótese que, para un 16.41% de *tweets*, la sugerencia de término clave no es validada por al menos otro método. Además, es más frecuente encontrar dos o tres métodos coincidentes respecto a las sugerencias generadas. Aunque la cantidad de palabras contenidas en cada texto podría influir en la probabilidad de

tener un puntaje de validación más alto, esta afirmación no posee sustento en los datos de la muestra.

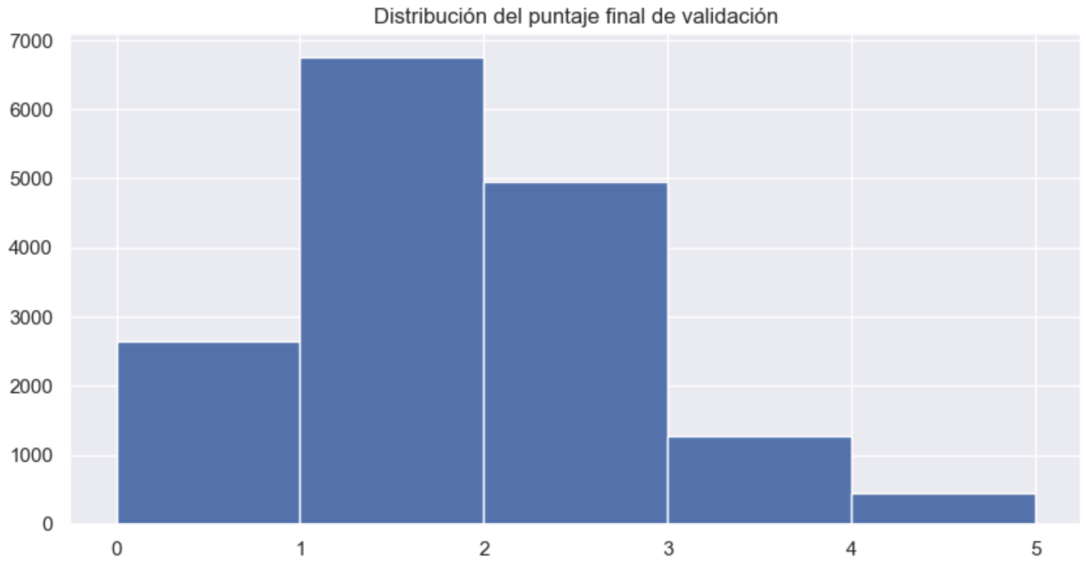


Figura 10. Histograma del puntaje de validación para las sugerencias de término clave generadas.

Como resultado final, la heurística utilizada permite asignar una sugerencia de término clave para un 83.58% (13,429) del universo contemplado. Dichas sugerencias se muestran en forma de nube de palabras en la Figura 11.


Figura 11. Nube de palabras en función de los términos clave generados para la muestra utilizando la heurística descrita en la sección.

### 2.3.3 Análisis de sentimiento

La polaridad de los textos es, sin lugar a dudas, una característica importante a la hora de analizar la muestra relevada. Haciendo foco particularmente en el dominio de atención al cliente, una variable de este estilo puede utilizarse para entender el contexto de cada *tweet* y, de esa forma, priorizar casos de forma inteligente.

Distintos algoritmos propios del análisis de sentimiento pueden emplearse para establecer la polaridad de los textos en la muestra. Algunos usan léxicos predefinidos para formular heurísticas mientras que otros requieren de modelos más complejos, frecuentemente hallados en el entorno del aprendizaje automático (Rao, 2019).

Con la intención de definir la polaridad de los *tweets*, en esta sección se eligen y describen tres modelos distintos, discutiendo sobre las herramientas disponibles para dicha tarea. Se obtiene un puntaje por cada modelo y se evalúa su comportamiento generando etiquetas para una muestra del *corpus*. Analizando distintas métricas de ajuste, se propone generar un ensamble en pos de mejorar la polaridad obtenida para cada *tweet*.

El primer modelo[12] seleccionado posee una arquitectura basada en redes convolucionales y ha sido entrenado con 800,000 reseñas en español de distintos sitios web como, por ejemplo, Ebay. Haciendo uso de este modelo, se obtiene la probabilidad de que un texto tenga polaridad positiva o negativa. Respecto a la calidad de dicha predicción, según los autores, el rendimiento del algoritmo sobre textos nunca antes vistos alcanza un *accuracy* promedio del 88%. Sin embargo, de existir grandes diferencias entre los *tweets* del dataset y aquellos textos contemplados para el entrenamiento, se podría observar una disminución de la capacidad predictiva del modelo.

El segundo algoritmo considerado pertenece a la línea de investigación de Aguilar et al. (2020) que explora el multilingüismo (*code-switching)* en el procesamiento del lenguaje natural. Los autores proponen *benchmarks* para distintas tareas como: NER (*named entity recognition*), LID (*language identification*), SA (*sentiment analysis*) y POST (*part-of-speech tagging*). Considerando que el "espanglish" aparece a menudo en la muestra de *tweets* relevada, se decide utilizar el modelo de análisis de sentimiento sugerido por los autores[13]. La arquitectura de dicho modelo se basa en BERT (*Bidirectional Encoder Representations from Transformers*) y ha probado alcanzar una *accuracy* promedio del 60% a la hora de inferir polaridad (positivo, neutral, negativo) en data nueva.

La tercera metodología elegida plantea el uso de heurísticas construidas a partir de léxicos que contienen métricas de intensidad de distintas palabras. Las reglas de VADER (*Valence Aware Dictionary and sEntiment Reasoner*) están pensadas, particularmente, para el dominio de las redes sociales donde, en general, hallamos textos cortos con uso de emojis y puntuación copiosa (Hutto & Gilbert, 2015). Considerando el desempeño del modelo[14], la clasificación del sentimiento (positivo, neutral, negativo) en *tweets* alcanza un puntaje F1 de 0.96.

Las herramientas disponibles para realizar análisis de sentimiento con *Python* en español son limitadas, de hecho, VADER sólo se encuentra implementado para textos en idioma inglés. Con el objetivo de experimentar con dicho algoritmo, se traducen los textos empleando una API *open-source* – LibreTranslate[15] – que basa su código en Argos Translate. No obstante, hay que tener en cuenta que la calidad de las traducciones podría

[12] En https://github.com/sentiment-analysis-spanish/sentiment-spanish se encuentra el código correspondiente al entrenamiento del modelo.
[13] Los detalles de implementación se hallan en https://pypi.org/project/codeswitch/
[14] Referir a https://www.nltk.org/_modules/nltk/sentiment/vader.html para más especificaciones.
[15] Para más detalles, referir a https://pypi.org/project/libretranslate/ .

verse afectada por las abreviaciones, los errores de ortografía y la falta de puntuación propias del lenguaje utilizado en Twitter.

Contemplando nuevamente aquellos 21,467 *tweets* caracterizados como relevantes (ver Sección 2.2.2), se procede a utilizar los distintos algoritmos mencionados. Dado que los puntajes de polaridad obtenidos no varían en el mismo rango de valores, se normalizan las variables (ver Apéndice E) y se grafica su distribución en la Figura 12 para comparar los resultados de los distintos modelos. Mientras que VADER categoriza la mayoría de los textos como neutrales, CNN RESEÑAS frecuentemente identifica los textos como negativos. Respecto a CODESWITCH, se comporta en general de forma uniforme pero posee un leve sesgo hacia puntajes por encima del 0.8.


Figura 12. Distribución de los puntajes de polaridad normalizados entre 0 y 1 para los distintos algoritmos utilizados. A mayor probabilidad, mayor chance de que el tono del texto sea positivo.

Puesto que los modelos poseen sesgos distintos, se apartan 505 *tweets* del universo inicial contemplado para medir el ajuste individual de cada algoritmo. Particularmente, se incluyen casos relevantes que podrían no encontrarse en una muestra aleatoria a menos que la cantidad de mensajes sea grande. En el Apéndice E se encuentran detallados los criterios de selección y la metodología de rotulado.

Una vez etiquetados los tweets seleccionados, se observa que la polaridad se compone de la siguiente forma: un 33% de casos son negativos, un 58% de casos son neutrales y un 9% de casos son positivos. Si bien es viable cuantificar el desempeño de los modelos empleados midiendo el ajuste de cada una de las clases de polaridad, tener una estimación continua sobre la insatisfacción del usuario puede ser de mayor utilidad para priorizar casos de atención al cliente. Por este motivo, se busca analizar particularmente la capacidad que tiene cada algoritmo para clasificar *tweets* de polaridad negativa, manipulando la probabilidad de pertenecer a dicha clase en vez de considerar un output categórico.

De esta manera, se re-definen las etiquetas de la muestra en forma dicotómica para reflejar si un *tweet* es de polaridad negativa o no. Además, se calcula el complemento de los puntajes normalizados de cada algoritmo para obtener la probabilidad de que un *tweet* sea de polaridad negativa. Luego, se comparan estas variables para medir el desempeño de cada algoritmo.

Respecto a las métricas de evaluación de los modelos, se estima la precisión (exactitud) del valor predicho utilizando dos funciones de pérdida — BRIER SCORE LOSS y LOG LOSS — que difieren principalmente en la escala y en la penalización del error. En primer lugar, el error correspondiente a BRIER SCORE LOSS varía entre cero y uno mientras que aquel resultante de LOG LOSS no se encuentra acotado. En segundo lugar, la función LOG LOSS penaliza de forma más estricta las predicciones incorrectas en comparación con BRIER SCORE LOSS. Para más información sobre estas métricas, ver las Ecuaciones 1 y 2 del Apéndice E.

Otro aspecto a considerar para entender el ajuste de los modelos es la correctitud del ordenamiento generado en las predicciones. Para ello, se utiliza el área bajo la curva (AUC) ROC (acrónimo de Receiver Operating Characteristic), que brinda información complementaría respecto a la estimación de precisión. La Tabla 2 muestra los cálculos de ajuste contemplados.

| IDENTIFICADOR DE MODELO | IDIOMA SOPORTADO | DETALLES DEL MODELO | MÉTRICAS | | |
|---|---|---|---|---|---|
| | | | ROC AUC | BRIER SCORE LOSS | LOG LOSS |
| VADER | Inglés | Uso de léxicos y heurísticas dirigidas al uso de redes sociales. | 0.6866 | **0.2050** | **0.5999** |
| CNN RESEÑAS | Español | Red neuronal convolucional ya entrenada con reseñas. | 0.6911 | **0.4537** | **4.1785** |
| CODESWITCH | Español / Inglés | Modelo basado en BERT para poder lidiar con el multilingüismo. | **0.7109** | 0.2184 | 0.6436 |
| ALEATORIO | - | Generado por $\hat{y} \sim U[0,1]$. | **0.4990** | 0.3342 | 1.0011 |

Tabla 2. Métricas de ajuste para cada uno de los modelos utilizados sobre la muestra (505 casos) etiquetada. En el caso del modelo ALEATORIO, las métricas son el promedio de unas 100 simulaciones de predicción aleatoria. En azul (rojo) se muestran los mejores (peores) valores por cada métrica.

Analizando los resultados obtenidos, se observa que las predicciones más precisas corresponden a VADER mientras que el mejor puntaje de ordenamiento se obtiene con CODESWITCH. La Figura 13 muestra los detalles sobre la comparación de métricas de los modelos considerando el algoritmo ALEATORIO como punto de partida. Las estimaciones muestran que VADER y CODESWITCH superan al modelo base tanto en precisión como en correctitud de ordenamiento. En el caso particular de CNN RESEÑAS, si bien el puntaje de ordenamiento se encuentra por encima del *baseline*, no se puede decir lo mismo de su precisión.

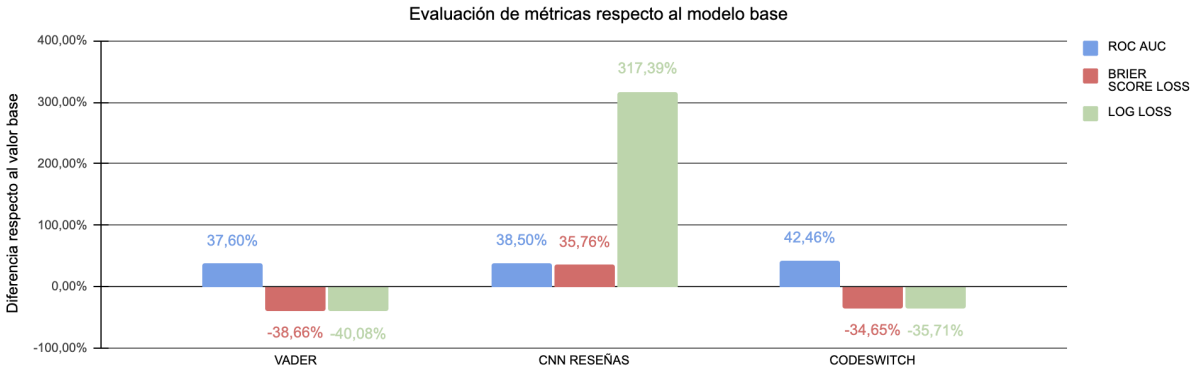


Figura 13. Evaluación de métricas de ajuste de los modelos respecto al modelo ALEATORIO base. Superar el modelo *benchmark* debe reflejar diferencias negativas para las funciones de pérdida y diferencias positivas para el valor de ROC AUC.

En pos de mejorar las predicciones generadas sobre los *tweets*, se ensamblan los tres modelos utilizados para intentar aumentar la precisión y la robustez de nuestras estimaciones (Brownlee, 2021). Con este objetivo, se proponen dos maneras de combinar las probabilidades de los algoritmos considerados: por un lado, se calcula un promedio de las predicciones y, por otro lado, se utiliza un proceso de AutoML[16] para encontrar el mejor ajuste entre varios modelos de aprendizaje automático. Provisto que el alcance de este trabajo es acotado, la segunda sugerencia busca optimizar tiempo e incrementar las chances de encontrar una combinación óptima.

Si bien el promedio no requiere entrenamiento alguno para poder generar el ensamble, el modelo de aprendizaje supervisado sí lo precisa. Por eso se divide la muestra etiquetada en 303 *tweets* destinados a la calibración del modelo y otros 202 para medir la eficiencia de este último. Ambos conjuntos poseen aproximadamente un 33% de casos con polaridad negativa. La Tabla 3 muestra los resultados obtenidos utilizando los ensambles propuestos. En el Apéndice E se pueden encontrar más detalles acerca del entrenamiento del modelo de AutoML.

| **IDENTIFICADOR DE ENSAMBLE** | **DETALLES DEL MODELO** | **MÉTRICAS** | | |
|---|---|---|---|---|
| | | **ROC AUC** | **BRIER SCORE LOSS** | **LOG LOSS** |
| PROMEDIO DE LAS PROBABILIDADES | Se toma el promedio de los puntajes re-escalados entre 0 y 1. | 0.7929 | 0.2201 | 0.6294 |
| AUTOML TPOT | Modelo basado en un árbol de decisión (*random forest*). | **0.8116** | **0.1609** | **0.4964** |

Tabla 3. Métricas de ajuste para los ensambles generados calculados sobre el dataset de test. En azul se muestran los mejores valores por cada métrica.

Analizando los resultados obtenidos, se encuentra que AUTOML TPOT supera las mejores métricas obtenidas al considerar los algoritmos de forma individual: el puntaje de ROC AUC aumenta aproximadamente un 12%, el BRIER SCORE LOSS disminuye casi un 22% y, por último, identificamos una disminución del 17% en LOG LOSS.

[16] La documentación acerca de la librería se encuentra en https://github.com/EpistasisLab/tpot

### 2.3.4 Inferencia de tópicos

El tópico encontrado en cada *tweet* es otro atributo del texto que puede ser útil al momento de manipular casos de atención al cliente ya que da noción del motivo del reclamo presentado. Puesto que no se cuenta con etiquetas que permitan identificar el tópico de cada *tweet*, se utilizan técnicas de aprendizaje no supervisado para inferirlos.

Si bien fueron explorados los textos previamente en la Sección 2.2, se decide realizar una pequeña muestra aleatoria a partir de la cual observar los temas abordados en cada uno de los *tweets*.

Dentro de dicha muestra, se encuentran los siguientes tópicos:

(i) quejas sobre la atención al cliente proporcionada
(ii) reclamos relacionados al correo (demoras en el envío, cancelaciones, impresión de etiquetas, despachos, recepción de producto incorrecto)
(iii) difusión de productos o servicios (con vínculo al sitio, ofertas sobre precios)
(iv) problemas con cuentas de usuario (validación de identidad y gestión de claves)
(v) inconvenientes en el proceso de compra (medios de pago, demora en envío de factura, cancelaciones y retención de dinero, imposibilidad de comprar y vender)
(vi) conflictos entre clientes y vendedores
(vii) dudas genéricas sobre el sitio (cómo publicar, cómo crear cuenta, etc)

La metodología de esta sección indaga dos potenciales alternativas. Por un lado, se manipulan las distribuciones de probabilidad palabra-tópico y tópico-documento con el objeto de identificar el tema abarcado por cada *tweet*. Por otro lado, se generan representaciones vectoriales de los documentos (*embeddings*), cuyo tamaño se reduce a dos dimensiones para poder, luego, realizar un agrupamiento (*clustering)* que potencialmente revele los tópicos presentes en el dataset.

#### Distribuciones de probabilidad palabra-tópico y tópico-documento

En esta sección, se adopta un enfoque probabilístico con el objetivo de realizar una asignación de tópicos a cada *tweet*. Para poder llevar a cabo dicha asignación, se hace uso de dos algoritmos que serán descritos brevemente a continuación: LDA (acrónimo de Latent Dirichlet Allocation) y Guided LDA.

LDA es un modelo generativo que permite obtener tanto (i) la probabilidad de que una palabra pertenezca a un tópico como (ii) la probabilidad de que un tópico esté asociado a un documento (Campbell et al., 2015). Esta metodología supone que tópicos similares harán uso de palabras similares, así como también que los documentos pueden hablar de múltiples tópicos. Cada documento estará representado por un número predefinido de *k* tópicos, donde cada tópico es una distribución que le asigna una probabilidad a cada palabra en el diccionario contemplado.

Ahora bien, *k* no es el único hiperparámetro de utilidad para el algoritmo. Al modelar las probabilidades de interés como distribuciones de Dirichlet, se añaden otros dos hiperparámetros (vectores) adicionales: α y β. El primero refiere a la densidad documento-tópico: cuando α aumenta, se espera observar un mayor mix de tópicos por

documento. El segundo refiere a la densidad tópico-palabra limitando, análogamente, la distribución de palabras por tópico: al aumentar β, mayor será el número de palabras probables por tópico.

Respecto al funcionamiento de LDA[17], se comienza asignando aleatoriamente cada palabra a un potencial tópico para cada documento. Durante esta fase de inicialización, es posible asignar distintos pesos a los tópicos para generar sesgos o bien distribuir las palabras de forma uniforme. Para actualizar el tópico asignado a una palabra *p* por cada documento se sigue el siguiente proceso: (i) se asume que el resto de las palabras se encuentran bien asignadas, (ii) se computa el set de términos *T* que suelen aparecer en conjunto con *p* y (iii) se asigna el tópico más frecuente en T a *p.* Este proceso se itera múltiples veces hasta lograr la convergencia del modelo.

Es en este contexto que Daumé III et al. (2012) introducen Guided LDA. Esta es una mera extensión del algoritmo descrito previamente, donde se propone usar palabras representativas del *corpus* como "semillas" para generar tópicos de interés. Por este motivo, si ya se tiene un entendimiento o intuición sobre los textos, es posible inicializar los pesos de ciertos términos para ayudar a la convergencia del modelo (LDA).

Vale la pena mencionar que, para este tipo de modelado, los documentos se consideran bolsas de palabras (o en inglés *bag of words*). Esto implica que cualquier tipo de información sintáctica como, por ejemplo, el orden de los términos o el rol gramatical, no será tenido en cuenta.

Para experimentar con los algoritmos, se parte nuevamente del set de tweets *tokenizados* (ver Apéndice B). Si bien dicho conjunto ha sido sometido a una limpieza de *stopwords*, el resultado de LDA no se verá afectado. Si una palabra aparece con alta frecuencia en el *corpus*, seguramente se observe en la mayoría de los tópicos a encontrar, por lo que no será útil para modelar.

Se comienza por intentar modelar los tópicos de la muestra de *tweets* sin utilizar ningún tipo de semilla. Para tener una referencia sobre la calidad de los tópicos, se hace uso de métricas de coherencia. No se eligen métricas como la perplejidad, por ejemplo, ya que se ha demostrado que correlaciona negativamente con la evaluación humana de tópicos (Chang et al, 2009). En este caso, se evalúa el aprendizaje del algoritmo y la interpretabilidad de los tópicos utilizando Umass, una métrica de carácter más bien intrínseco, que mide el grado de similaridad semántico entre las palabras de un mismo tópico partiendo de una matriz de co-ocurrencia (Röder et al., 2015).

A continuación, se experimenta con el número de tópicos a encontrar en el *corpus* (parámetro *k*) como, también, con el número de pasadas que el algoritmo realiza durante el entrenamiento. Si bien Umass parece decrecer a medida que incrementa el valor de *k,* se elige *k* = 6 con 10 pasadas de entrenamiento. Analizando las palabras más relevantes por cada tópico hallado, es posible notar que ciertos términos se corresponden con los temas explorados inicialmente. Por eso, se itera este primer resultado utilizando semillas que

[17] Ver https://tedboy.github.io/nlps/generated/generated/gensim.models.LdaMulticore.html para detalles de implementación del algoritmo de LDA.

ayuden a la convergencia del modelo. Para más detalles sobre la elección del valor de *k,* las semillas utilizadas y las palabras más relevantes por tópico ver el Apéndice F.

Posteriormente, se busca entender la distribución de contenidos en la muestra. Para ello, se asigna a cada *tweet* el tópico más probable, requiriendo una mínima probabilidad del 20%. Si bien la Figura 14 parece mostrar que algunos temas aparecen con mayor frecuencia que otros, al indagar las distribuciones tópico-documento con detalle, se observa que un mismo *tweet* puede estar hablando de múltiples asuntos al mismo tiempo.

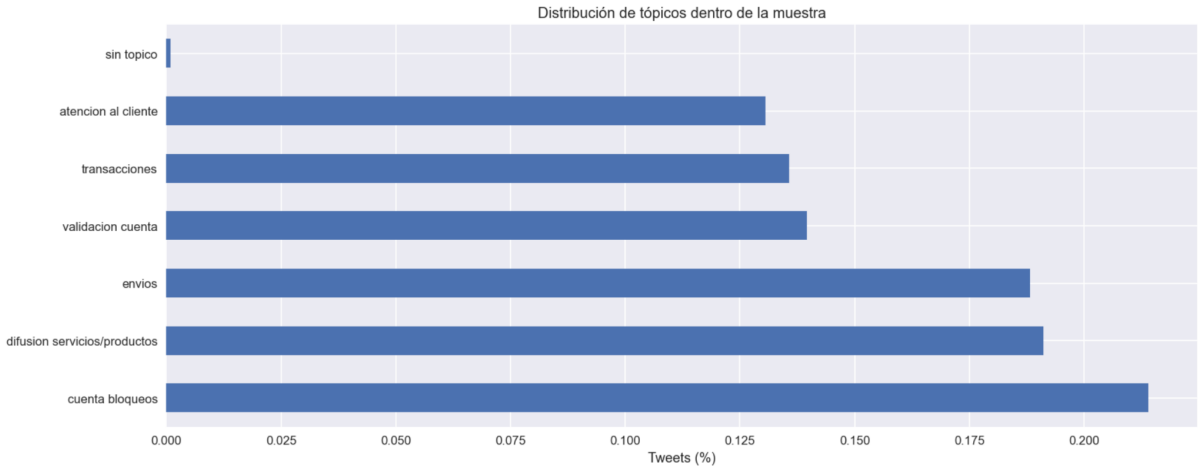


Figura 14. Distribución del tópico más probable para cada *tweet* en la muestra requiriendo una mínima probabilidad del 20%.

Siguiendo este último hallazgo, se contabiliza la cantidad de temas con al menos una probabilidad del 20% por documento: un 50.24% de los *tweets* se ven asignados a un solo tópico, un 41.27% a dos, un 8.37% a tres y menos del 1% a 4. Puesto que un porcentaje considerable de la muestra habla de dos temas a la vez, se genera una matriz de co-ocurrencias para entender qué combinaciones de contenido son usuales o, de forma contraria, poco frecuentes. La Figura 15 muestra dicha matriz normalizando los valores de cada celda por aquellos posicionados sobre la diagonal (volviéndose no simétrica) y la Tabla 4 introduce algunos casos de ejemplo.

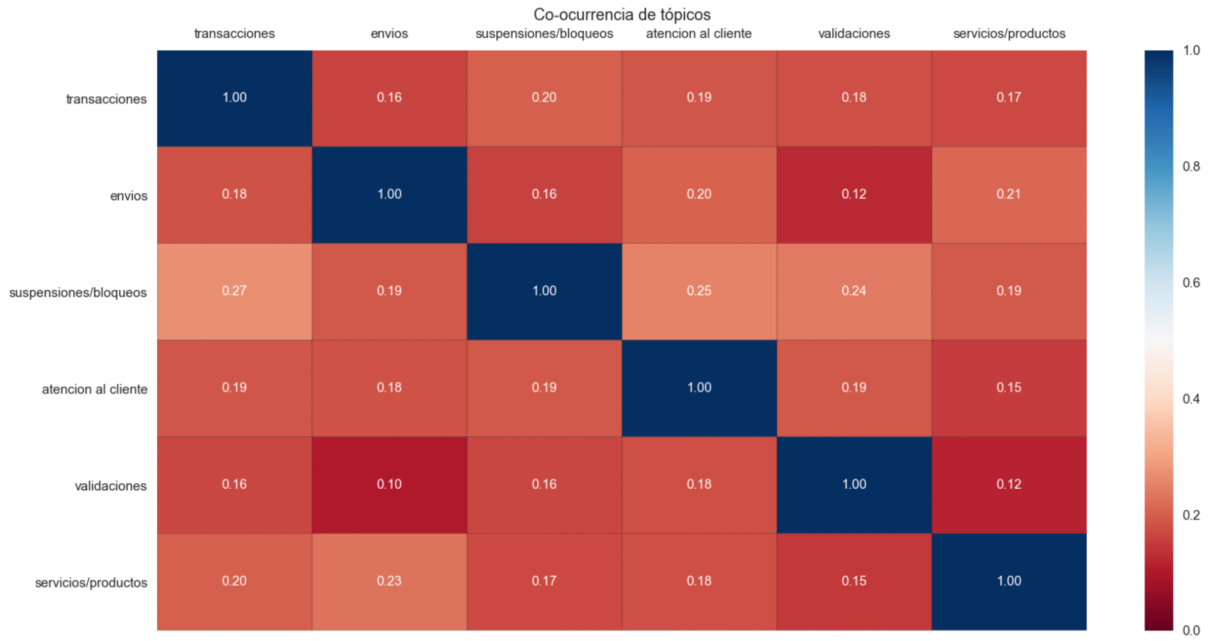


Figura 15. Co-ocurrencia de los tópicos presentes. Los valores de la matriz se encuentran **normalizados por columna**. Por ejemplo, para el 18% de los tweets donde el tópico es "transacciones", también se observa "envíos".

| TEXTO | PRIMER TÓPICO | SEGUNDO TÓPICO |
|---|---|---|
| @ML_Ayuda Hola, ya me dieron solucion y mi cuenta de mercado pago, que es supuestamente donde abonaron mi dinero. Las dos están activas, pero mi saldo está en cero, tengo comprobantes en emails de que me lo abonaron, que procede? | BLOQUEO / SUSPENSIONES (66.99%) | TRANSACCIONES (28.63%) |
| @Mercadolibre @ML_Argentina hola, estoy esperando devolución de compra que hice el 28 de septiembre.. No me entregaron el producto, nunca llegó en 30 días y tuve que pedir reintegro.. | TRANSACCIONES (45.55%) | ENVÍOS (33.70%) |
| @ML_Ayuda tengo un problema con una compra que hice en diciembre y la plataforma no me permite escalar la situación y no hay ningún medio de comunicación directo | TRANSACCIONES (41.59%) | ATENCION AL CLIENTE (52.81%) |
| Don Rebusque vende taba antigua 👉 #Importante NO HAGO ENVÍOS https://t.co/YlPsIKuUGD #Antigüedades #Vintage #Retro #Juego #Campo #Rural #Tradición #Artesanía #MercadoLibre #MercadoPago\n#Rebusque #MeLasRebusco #DonRebusque | ENVÍOS (60.00%) | DIFUSIÓN DE SERVICIOS Y PRODUCTOS (26.00%) |
| Hola @ML_Ayuda me aparece que rechazaron la compra con mi tarjeta, anteriormente ya había hecho compras y verifique mi identidad. Ayuda https://t.co/S4J27DJzHf | VALIDACIONES DE CUENTA (62%) | TRANSACCIONES (32.41%) |
| @ML_Ayuda Hola, buenas noches, desde hace horas, no me deja ingresar a mi cuenta de mercado libre... Tengo un paquete que está por llegar. | ENVÍOS (52.18%) | VALIDACIONES DE CUENTA (39.41%) |
| @GilGarciaD @ML_Ayuda @ML_Mexico @Mercadolibre Espero te funcione. A mi me funcionó. De hecho ya hice mi compra y no podía desde hace como 10 días me salía el mismo error que a ti. Pero ahí verificando nuevamente mis datos ya me la activo. | VALIDACIONES DE CUENTA (36.04%) | BLOQUEO / SUSPENSIONES (43.70%) |

Tabla 4. Algunos ejemplos que corresponden a más de un tópico.

Haciendo hincapié en ciertas relaciones de la matriz, se percibe que la mayoría son efectivamente factibles. Analizando *tweets* que caen en distintas categorías, se puede agregar un poco de contexto para algunas de las relaciones altas y bajas de la tabla. Como

primer ejemplo, se investiga el 27% de los documentos que, siendo asignados a transacciones, también son asignados a bloqueos de cuenta. Estos casos suelen tener que ver con reclamos sobre retención de dinero o imposibilidad de vender y comprar a causa de tener las cuentas suspendidas.

Un segundo ejemplo de interés está definido por el 23% de los documentos asignados a envíos que, también, se relacionan a la difusión de servicios y productos. Esta coyuntura refleja casos donde un vendedor indica las condiciones de envío de una publicación o potenciales beneficios como, por ejemplo, promesa de entrega en el día. De forma similar, el 20% de los documentos de transacciones asignados a difusión de productos o servicios cobran sentido en un contexto donde se utilizan medios de pago o rebajas para promocionar publicaciones.

Un tercer ejemplo viene dado por el 24% de los casos asignados a validaciones que, inevitablemente, se vinculan con suspensiones de cuenta. Esta relación se explica fácilmente ya que, muchas veces, para desbloquear cuentas se requiere un método de autenticación proveyendo datos personales para validar la identidad.

Una última combinación a ser resaltada por ser poco usual es aquella determinada por el 10% de los documentos asignados a envíos que, además, pueden hablar de validación de identidad. En general, estos *tweets* suelen relacionarse con la imposibilidad de dar seguimiento a un envío puesto que requieren de validaciones de cuenta.

En líneas generales, los resultados obtenidos en la muestra trabajada parecen tener sentido. No obstante, se detectan algunos motivos por los cuales el algoritmo puede otorgar probabilidades erróneas. Puesto que se parte de un modelo de bolsa de palabras, la falta de contexto genera confusión en los tópicos que, muchas veces, hacen uso de los mismos términos pero con significado distinto. Además, las faltas ortográficas de los *tweets* dificultan la generación de un diccionario limpio para poder generar los vectores de representación. Es notable mencionar que, al trabajar con texto de corta longitud, la matriz documento-término puede volverse poco relevante para modelar.

### Vectores-documento, reducción de dimensionalidad y clustering

Este apartado tiene como objetivo mostrar otra alternativa al problema de asignación de tópicos. En particular, se busca generar una representación vectorial de cada documento para luego reducir su dimensionalidad y generar *clusters* que den a conocer potenciales tópicos en la muestra. Este enfoque posee tres componentes que se detallarán a continuación.

En primer lugar, se elige una forma de vectorizar los documentos del *corpus* ya que se puede (i) entrenar *embeddings* sobre la data o (ii) utilizar embeddings ya entrenados sobre muestras más grandes. Si bien esta última opción parece atractiva para aprovechar el volumen de datos de entrenamiento, podría suceder que no haya un buen *fit* con el *corpus*, especialmente teniendo en cuenta que se hace foco en un dominio particular (atención al cliente) y que los textos están sujetos a los rasgos del propio lenguaje utilizado en Twitter.

Antes que nada, se evalúa la opción de utilizar embeddings ya entrenados. Para ello, se manipula la librería spaCy[18], seleccionando un modelo entrenado con noticias en español que genera vectores de tamaño 96 para cada uno de los documentos. La dimensión del *embedding* no es algo despreciable puesto que agrega complejidad para visualizar los datos y, también, afecta la estimación de los algoritmos de *clustering.* Si la data del *corpus* no es lo suficientemente compleja, se podría introducir ruido.

Con el objetivo de retener la mayor cantidad de información posible durante la reducción de dimensionalidad de los *embeddings*, se analiza el uso de UMAP[19] (McInnes et al., 2020). A diferencia de PCA (Principal Components Analysis), esta técnica no supone relaciones lineales y, a diferencia de t-SNE (t-Distributed Stochastic Neighbor Embedding), logra preservar mejor la estructura de los datos proyectados (Sivarajah, 2021).

Utilizando distintas medidas de distancia para proyectar los vectores con UMAP en dos dimensiones, no es posible identificar agrupaciones que permitan distinguir tópicos (Figura 16). Esto probablemente significa que el ajuste de los *embeddings* no es el adecuado, ya que la exploración inicial provee evidencia de que sí existen tópicos dentro del dataset.

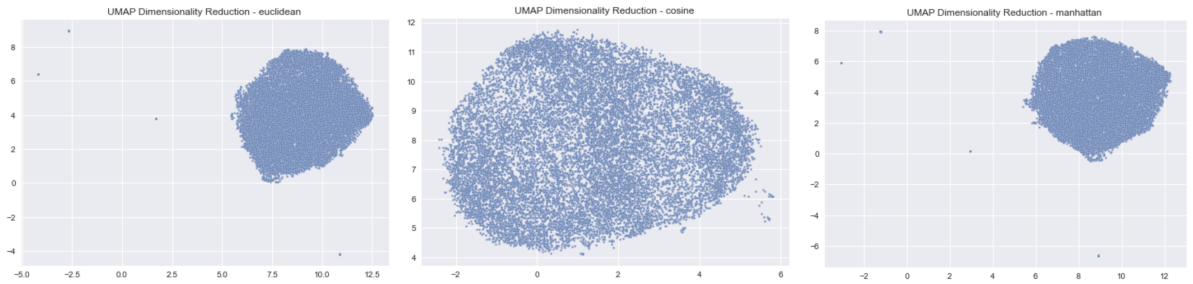


Figura 16. Proyecciones en un plano bidimensional generadas utilizando UMAP sobre los embeddings de Spacy. Los vectores fueron normalizados previamente por comodidad pero el algoritmo no es sensible a las escalas.

Para obtener representaciones vectoriales que ajusten mejor, es posible entrenar *embeddings* empleando, por ejemplo, el modelo Doc2Vec[20]. A diferencia de las representaciones que pueden obtenerse siguiendo un modelo de bolsa de palabras, este modelo no supervisado permite agregar la noción de contexto a los vectores y, también, facilita elegir el tamaño del *embedding*. Para más detalles sobre los parámetros utilizados en el entrenamiento, ver el Apéndice F.

La Figura 17 muestra los vectores resultantes del entrenamiento del modelo Doc2Vec, ya proyectados en un plano bidimensional haciendo uso, nuevamente, de UMAP. En este caso, se nota rápidamente la existencia de aglomeraciones de puntos que permiten pensar, a priori, que es posible formar *clusters*. Es interesante observar los puntos hallados entre concentraciones que, pareciera, conectan potenciales temáticas.

[18] Para más detalles, referir a https://spacy.io/
[19] Ver https://umap-learn.readthedocs.io/en/latest/ para información sobre la implementación en *Python* para UMAP utilizada.
[20] Ver https://radimrehurek.com/gensim/models/doc2vec.html para la implementación del algoritmo.

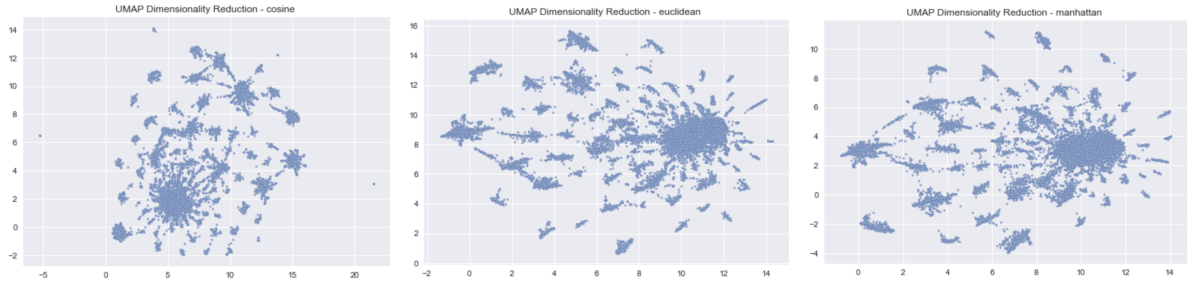


Figura 17. Proyecciones en un plano bidimensional generadas utilizando UMAP sobre los embeddings entrenados con Doc2Vec. Los vectores fueron normalizados previamente por comodidad pero el algoritmo no es sensible a las escalas.

Con el objetivo de identificar las concentraciones de puntos, se realiza *density-based clustering* empleando DBSCAN[21] (Ester et al., 1996). A diferencia de otros métodos como *k-means*, este algoritmo: (i) no precisa de un número de *clusters* para hallar, (ii) es capaz de descartar observaciones de tipo ruidosas y (iii) no busca generar agrupaciones del mismo tamaño ni con forma esférica.

Respecto al funcionamiento del método per se, se utilizan distancias entre puntos (siendo afectado por la escala de las unidades) para formar *clusters* que cumplen con una densidad determinada. Para ajustar el algoritmo, se experimenta particularmente con dos parámetros: *epsilon* y *min_points*. Mientras que el primero refiere a qué tan cerca deben estar los puntos para ser parte de una misma agrupación, el segundo es básicamente el requerimiento mínimo para ser considerado una región densa. Nótese que tanto incrementar *epsilon* como reducir *min_points* genera, potencialmente, un aumento de ruido en los *clusters*.

Ester et al. (1996) proponen elegir *epsilon* calculando, para cada observación, la distancia respecto a su vecino más cercano de forma que, luego, se visualicen estas distancias ordenadas (*sorted k-dist graph*) para identificar el punto de mayor curvatura. Todo lo que quede por encima del valor elegido, será considerado ruido.

Ajustando los parámetros, se elige *epsilon* = 0.08 y se varía *min_samples* para entender el impacto generado sobre el número de clusters y, también, sobre el porcentaje de observaciones clasificadas como ruido por DBSCAN. Para tener una referencia acerca de la calidad de los *clusters*, se calcula el coeficiente de *silhouette*, que mide la hermeticidad de las agrupaciones (distancia *intra-cluster*) y su separabilidad (distancia *inter-cluster*). Nótese que ciertos *tweets* hablan de múltiples tópicos, por lo que eventualmente la separabilidad de los *clusters* se puede ver afectada. Los detalles de esta experimentación se encuentran en el Apéndice F.

A pesar que esta alternativa no refleja bien la naturaleza de la muestra de *tweets,* porque sólo asigna un tópico por documento, se puede utilizar *min_samples* para generar un análisis con mayor granularidad. Por este motivo, esta sección tiene un propósito exploratorio y, eventualmente, para definir el tópico de cada documento se utilizan los resultados obtenidos previamente.

[21] Referirse a https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html para más detalles sobre el código de la librería.

### 2.3.5 Probabilidad de requerir atención personalizada

El último atributo modelado para cada *tweet* es la probabilidad de requerir atención personalizada. Esta característica se torna relevante en un contexto de recursos limitados, donde es necesario poder definir qué casos priorizar de forma *data-driven*.

En esta ocasión, se utilizan técnicas de aprendizaje supervisado, por lo que se comienza creando un dataset y definiendo rótulos que identifiquen la variable objetivo. Luego, se modela la variable de interés bajo una arquitectura BERT, extrayendo distintos *features* del texto para iterar el ajuste de las predicciones.

Para generar el dataset de entrenamiento, se parte de los tópicos obtenidos en la Sección 2.3.4. Se asigna, nuevamente, el tema de cada *tweet* como aquel de mayor chance, siempre que la probabilidad supere el 25%. Con el objetivo de asegurar variedad en la muestra, se seleccionan aleatoriamente 200 documentos por cada uno de los tópicos identificados. Además, se escogen otros 300 documentos al azar del set de *tweets* que no fueron agrupados bajo ninguna temática para poder cubrir casos potencialmente nuevos. Consolidando ambos criterios, se obtiene un conjunto de 1500 documentos.

Habiendo generado la muestra, se construye el criterio con el que se rotulan los casos. La variable objetivo contempla los *tweets* donde se observa un potencial accionable sobre el cual puede trabajar un agente. La idea es excluir casos donde un usuario: (i) comparte una mala experiencia de compra en la que ya no es posible intervenir, (ii) escribe una queja pero no deja claro si busca ayuda o, (iii) menciona a la empresa sin tratar asuntos de atención al cliente.

Con el propósito de ampliar el dataset, se incluyen 45 casos donde los mensajes son escritos por cuentas que brindan soporte en Twitter. Por definición, estos documentos no están asociados a requerir servicios de atención al cliente y, por eso, pueden ser clasificados automáticamente. Como resultado, se obtiene un 23% de casos negativos en la variable de interés.

Respecto al proceso de etiquetado manual, existieron ocasiones particulares en las que fue complejo definir la variable objetivo. Mientras que algunos *tweets* contienen poca información para crear los rótulos, otros poseen un tono más bien irónico que crea confusión sobre si realmente el usuario requiere atención personalizada.

Dado que se cuenta con un dataset limitado, se opta por usar *transfer learning* partiendo de un modelo pre-entrenado sobre un *corpus* de gran tamaño. Devlin et al. (2019) proponen usar la arquitectura BERT basada en redes neuronales para generar representaciones bidireccionales a partir de texto sin rotular. La salida de este modelo puede luego ser ajustada para realizar otro tipo de tareas relacionadas al procesamiento de lenguaje natural.

En este caso particular, se plantea un problema de clasificación utilizando como input los vectores de BERT provenientes de un modelo entrenado en español (Cañete et al., 2020). En pos de generar *features* que puedan dar información adicional, se extrae utilizando expresiones regulares si los textos contienen: (i) emails, (ii) links, (iii) menciones a una

cuenta soporte o a un usuario, (iv) hashtags y (v) fechas. Estas variables se incluirán en los *embeddings* pre-existentes haciendo uso de *tokens* personalizados.

A continuación, se realiza el *fine-tuning* de BERT utilizando un 50% de los casos para entrenamiento, un 20% de los casos para validación y un 30% de los casos para evaluación externa. Se experimenta, particularmente, con los *tokens* personalizados que buscan extraer información adicional. La Tabla 5 muestra el desempeño del modelo por cada iteración ejecutada para predecir la probabilidad de que un usuario requiera atención al cliente. En el Apéndice G se encuentran los detalles de la arquitectura del modelo.

| ITERACIÓN DEL MODELO | FEATURES ADICIONALES BASADOS EN EL TEXTO | MÉTRICAS | |
|---|---|---|---|
| | | ROC AUC | BRIER SCORE LOSS |
| 1 | Sin *features* adicionales, sólo se convierte el texto a minúsculas. | 0.9503 | 0.0957 |
| 2 | Se incluye la transformación de [1] y se agrega el token '[mayus]' si se utiliza al menos una palabra completa en mayúsculas. | 0.9586 | 0.0921 |
| 3 | Se incluyen los *features* de [2] y se reemplazan los emails por el token '[email]'. | 0.9606 | 0.0793 |
| 4 | Se incluyen los *features* de [3] y se reemplazan los links por el token '[link]'. | 0.9689 | 0.0798 |
| 5 | Se incluyen los *features* de [4] y se diferencian las cuentas soporte de aquellas que son de usuarios utilizando los tokens '[soporte]' y '[usuario]' respectivamente | 0.9701 | 0.0664 |
| 6 | Se incluyen los *features* de [5] y se reemplazan los hashtags por el token '[hashtag]'. | 0.9642 | 0.0691 |
| 7 | Se incluyen los *features* de [6] y se reemplazan las fechas por el token '[fecha]'. | 0.9722 | 0.0604 |

Tabla 5. Métricas de evaluación del modelo sobre *test* para cada iteración realizada.

Nótese que entre la primera y la séptima iteración se observa un aumento del 1.30% en la calidad del ordenamiento de las predicciones y una reducción del 38.88% de la función de pérdida. Para poder contextualizar los valores de las métricas presentadas, se construye un *benchmark* a partir de una distribución uniforme entre 0 y 1, obteniendo un valor de 51.65% para ROC-AUC y un 0.3256 para BRIER SCORE LOSS. Esto significa que la séptima iteración del modelo supera ampliamente el modelo base.

# 3. Tablero de Negocio

En esta sección se consolidan en un tablero de Data Studio, las características obtenidas de los tweets tanto en la fase de modelado como en la de recolección de los mismos[22]. El objetivo es poder proporcionar una interfaz interactiva para que el negocio pueda explorar y manipular la información en pos de tomar decisiones basadas en datos. Dicha interfaz posee **dos aplicaciones** que se detallan a continuación.

## 3.1 Aplicación: Desempeño regional

La vista de desempeño regional en el tablero (Figura 18) busca describir, para un periodo de tiempo determinado, el estado general de cada país donde opera la empresa. Diseñada para el uso de gerentes o puestos afines, se pretende dar a conocer rápidamente la situación de rendimiento.


Figura 18. Vista de la hoja del tablero correspondiente al desempeño regional.

Como indicadores principales, se presenta la satisfacción del usuario y la probabilidad de requerir atención al cliente. De forma complementaria, se incluye la cobertura obtenida a través del proceso de recolección. Dichas métricas pueden ser analizadas teniendo en cuenta dos dimensiones: la temporal y la geoespacial.

Respecto a la dimensión temporal (Figura 19), se muestra la evolución de los indicadores a través del tiempo y su valor puntual dentro de un periodo seleccionado. Esto permite evaluar las tendencias de las series temporales así como, también, identificar la estacionalidad y los potenciales ciclos.

[22] En https://datastudio.google.com/reporting/29bf9546-6cae-4f3d-8b31-43f354728039 se encuentra el tablero mencionado.

Respecto a la dimensión geoespacial, se desglosan las series temporales por país para dar lugar a un análisis de tipo comparativo (Figura 19). Además, se añade un mapa de burbujas donde se identifica, con el tamaño, la cantidad de *tweets* hallados por región y, con el color, el valor de un KPI a seleccionar (Figura 20). Con dichos recursos, se busca otorgar herramientas al usuario para que pueda identificar países que requieran una mejora en su rendimiento.

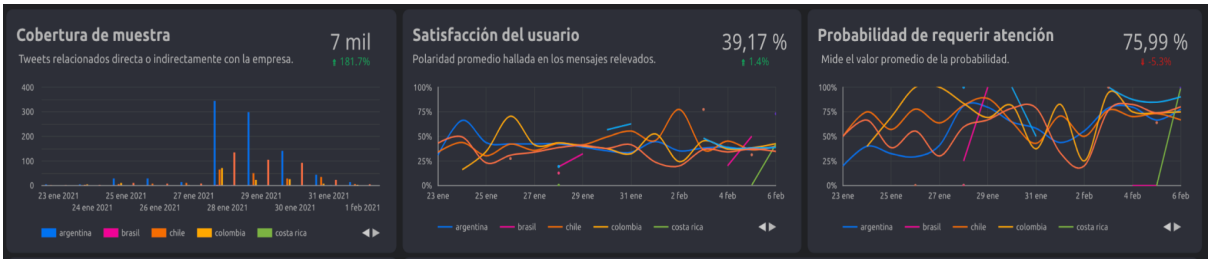


Figura 19. Componentes del tablero que dan lugar a analizar la dimensión temporal y geoespacial de los indicadores. Los colores de las series distinguen los distintos países.


Figura 20. Componente geoespacial adicional del tablero. (i) A la izquierda el mapa de burbujas y (ii) a la derecha el selector de métrica.

Otro aspecto contemplado es el contenido de los mensajes que refieren, directa o indirectamente, a la empresa. Para cada *tweet,* se define el tópico principal como aquel más probable siempre que supere el 25% y el tópico secundario como el segundo más probable cuando supera el 15%. Considerando esta heurística, se incluyen indicadores que permiten dar seguimiento al número de mensajes que no poseen un tópico principal o secundario.

Además, se introduce la cantidad de tweets por tópico principal en términos relativos y, desglosando nuevamente por país, el mismo valor en términos absolutos . Para entender la relación entre tópicos principales y secundarios, se proporciona una tabla de frecuencia de doble entrada que resalta las combinaciones más comunes con un mapa de calor. La Figura 21 muestra en detalle el componente de tópicos en el tablero.

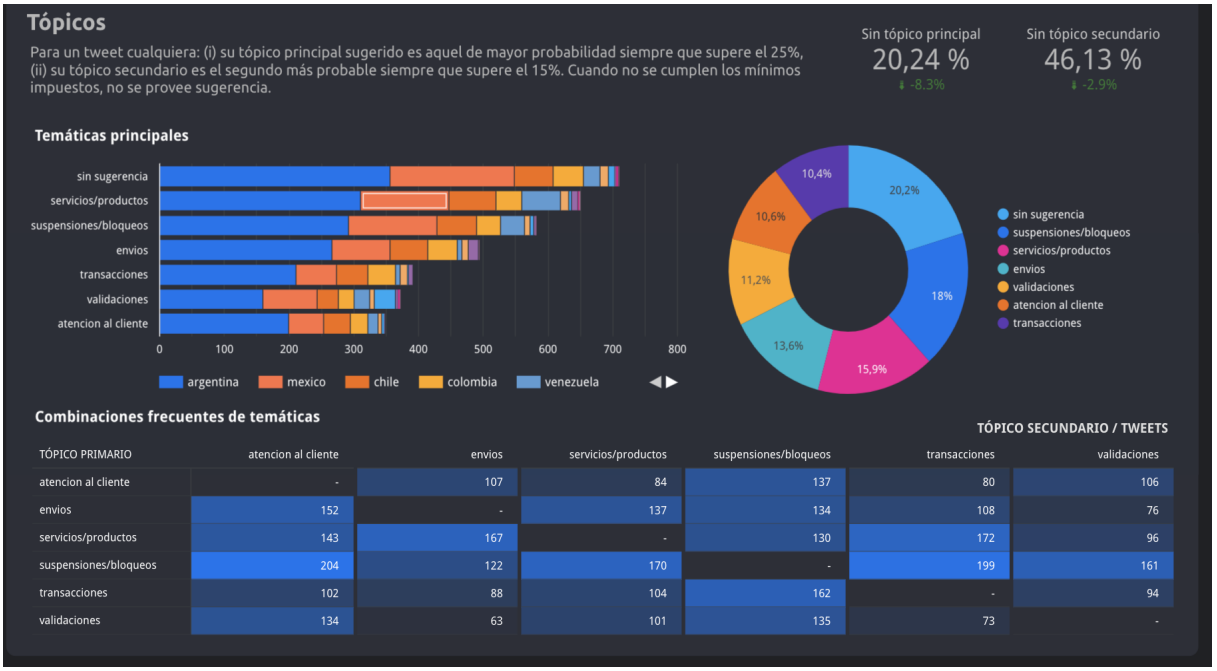


Figura 21. Componente de tópicos del tablero. (i) Arriba a la derecha se observan los indicadores de cobertura para los tópicos, (ii) al medio a la izquierda un gráfico de barras con valores absolutos desglosado por país, (iii) al medio a la derecha una gráfica de torta para la comparación porcentual entre tópicos principales, (iv) debajo una tabla de doble entrada con columnas caracterizadas por mapas de calor.

Como potencial caso de uso se sugiere, primero, realizar un análisis comparativo entre regiones para poder comprender qué países deben ser intervenidos y, luego, investigar a detalle las métricas presentadas para el área de interés. Esto último comprende tanto la evolución en el tiempo de los indicadores así como, también, los cambios en la composición de tópicos hallados en los mensajes.

## 3.2 Aplicación: Monitoreo de tweets

La vista de monitoreo de *tweets* en el tablero (Figura 22) está preparada para el uso, principalmente, de agentes de atención al cliente. En este caso, se incorporan elementos que ayudan a entender los mensajes de forma detallada para que los usuarios tomen decisiones más informadas tanto de corto como de largo plazo.


Figura 22. Vista de la hoja del tablero correspondiente al monitoreo de tweets.

Antes que nada, nótese que en la parte superior de la vista se añaden filtros que dan lugar a reducir el set de mensajes mostrados (Figura 23). Pueden aplicarse condiciones sobre: (i) el país, (ii) el usuario emisor, (iii) el tópico principal, (iv) la ventana temporal, (v) el puntaje de satisfacción, (vi) la probabilidad de requerir atención personalizada y, además, (vii) el contenido del texto del mensaje. Para entender el efecto de los filtros, se proveen algunas métricas que describen el subset exhibido.


Figura 23. Componente de filtrado del tablero. (i) En la parte superior los filtros de la vista y, (ii) en la inferior las estadísticas del subset filtrado.

El componente tabular del tablero tiene como objetivo poder brindar contexto sobre el texto del mensaje así como, también, sobre el usuario emisor (Figura 24). Considerando una toma de decisiones más bien de corto plazo, donde un agente debe elegir qué casos priorizar, se sugiere aplicar filtros sobre los mensajes y, luego, ordenar la tabla por las variables presentadas para obtener perspectiva sobre distintas situaciones de criticidad.

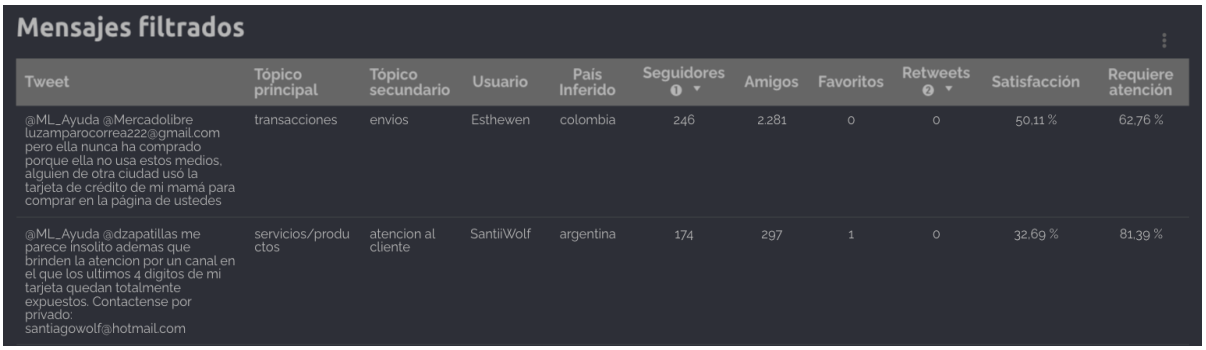


Figura 24.Visualización del componente tabular del tablero.

Dicha visualización puede utilizarse, por ejemplo, para minimizar el riesgo de propagación de malas experiencias relacionadas a la empresa en Twitter. En este caso, conviene filtrar aquellos mensajes donde existe un bajo puntaje de satisfacción y una alta probabilidad de requerir atención al cliente. Luego, para darle importancia a los emisores de mayor influencia en redes, se puede arreglar de forma descendente la tabla por el número de seguidores. Aunque este es solo uno de los tantos usos que se le puede dar al componente, se deja en evidencia la forma en la que puede emplearse para tomar decisiones más informadas.

Con el propósito de poder desglosar y entender problemáticas recurrentes, se añade tanto la gráfica de tendencias como la de composición de mensajes (Figura 25). Mientras que la primera muestra rápidamente los términos clave en una nube de palabras, la segunda hace uso de burbujas para representar las palabras más frecuentes por tópico principal en función de la satisfacción del usuario y la probabilidad de requerir atención al cliente. Nótese que el color de la burbuja representa el tópico y su tamaño la cantidad de *tweets*. Si una misma palabra se encuentra en más de un tópico, las burbujas se superpondrán.

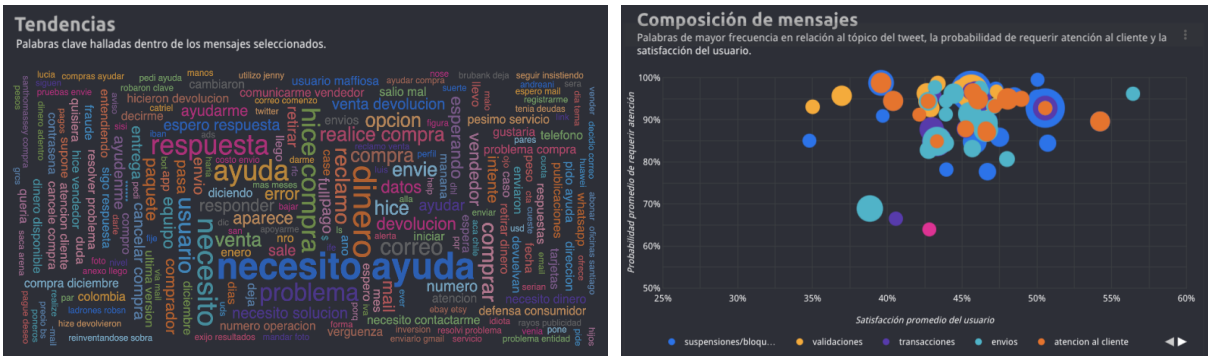


Figura 25. Visualización de términos relevantes en el tablero. (i) A la izquierda la gráfica de tendencias y (ii) a la derecha la gráfica de composición de mensajes.

Se sugiere hacer uso de las gráficas mencionadas para poder planificar, a largo plazo, acciones que eviten la generación de casos recurrentes de atención al cliente. Examinando los términos clave y la composición de los tópicos, se pretende generar una respuesta más eficiente por parte de los agentes y, eventualmente, de la empresa.

Para ejemplificar un caso de uso, se puede partir de la visualización de tendencias para afirmar que "dinero" es un término frecuente en los *tweets*. Naturalmente, una aserción de este estilo requiere de mayor contexto para poder generar algún tipo de accionable. Por eso, corresponde aplicar un filtro de texto para retener aquellos mensajes que contienen el término en cuestión, de manera que se puedan comprender mejor las potenciales problemáticas asociadas a "dinero".

Analizando la composición de los mensajes, se percibe que la mayoría posee palabras asociadas al tópico de transacciones. Para conocer aún más acerca del contenido de los *tweets*, se puede volver a la nube de palabras y, también, se pueden buscar ejemplos en el componente tabular de la vista. Esta información puede emplearse con distintos propósitos como, por ejemplo, para mejorar e iterar las preguntas frecuentes de la web.

# 4. Conclusiones

## 4.1 Logros

Aquellas empresas que brindan una buena experiencia de compra generan, eventualmente, un aumento de ingresos. Sin embargo, la mayoría de los negocios ha adoptado una modalidad de "autoservicio" que complejiza las problemáticas recibidas por los agentes de atención al cliente. De esta forma, se deja en evidencia una falta de herramientas y capacitación necesarias para alcanzar la calidad del servicio que los consumidores esperan.

A lo largo de este trabajo, se ha buscado crear una herramienta de negocio que permita situar estratégicamente recursos limitados y atender de forma inteligente los casos de atención al cliente. Al perseguir una mejora en el servicio de soporte brindado, se intenta obtener una mayor lealtad por parte de los consumidores para diferenciar a una empresa cualquiera.

Puesto que las redes sociales ayudan a construir una imagen de disponibilidad e inmediatez, muchas compañías han utilizado estas plataformas como medio para brindar soporte. Siendo Mercado Libre una de ellas, se hizo foco en dicha empresa para reducir el alcance del estudio, recolectando datos de Twitter accesibles vía API con consultas parametrizadas.

Una vez obtenido el *corpus* de *tweets* vinculados directa o indirectamente a Mercado Libre, se exploraron y modelaron distintas características de interés. Estas incluyen: la probabilidad de requerir atención al cliente, la satisfacción del usuario, las palabras clave del texto, los tópicos y la ubicación del tweet. Con este fin, se aplicaron técnicas de aprendizaje supervisado como no supervisado, se emplearon ensambles y se utilizaron expresiones regulares.

La herramienta de negocio propuesta consolida la fase de modelado en un tablero diseñado para dos potenciales aplicaciones. Por un lado, se pretende dar visibilidad sobre el estado general y regional de la empresa, caracterizando el rendimiento de los distintos países donde hay actividad. Por otro lado, se detallan los *tweets* con mayor granularidad para poder monitorear casos, tanto a corto como largo plazo, en un marco de recursos limitados.

Si bien el alcance de la investigación cubre sólo a Mercado Libre, nada impide repetir el proceso de recolección y análisis de datos para otra empresa que brinde soporte al consumidor vía Twitter. Para ello, probablemente se deban refinar los parámetros de búsqueda de la API y, además, ajustar el modelado a las peculiaridades del rubro.

## 4.2 Limitaciones y Extensiones

Respecto a las dificultades y restricciones detectadas durante el desarrollo del trabajo, se pueden mencionar principalmente tres: (i) el tipo de lenguaje e idioma utilizado en Twitter, (ii) la ausencia de datos etiquetados y, por último, (iii) la actualización de los modelos y análisis presentados. También es posible discutir extensiones del trabajo considerando, por un lado, modificaciones en las técnicas de modelado y, por otro lado, las oportunidades de

experimentación que existirían al incorporar la herramienta implementada en el ecosistema de trabajo de los agentes de atención al cliente en Mercado Libre.

Comúnmente, dependiendo de la red social que se utilice, se suelen encontrar mutaciones del lenguaje. En el caso particular de los *tweets*, los textos parecen reflejar el flujo de pensamientos del hablante, evadiendo principalmente signos de puntuación. El ahorro de caracteres es la regla instaurada, derivando en textos cortos con frecuente uso de abreviaciones y símbolos (emojis). El lenguaje usual es informal e irregular, con errores gramaticales y oraciones mal formuladas que no logran organizar el contenido. En este marco de hábitos construidos se encuentra, además, la mezcla de distintos idiomas en un mismo texto e, incluso, en una misma palabra.

Si bien los mensajes de Twitter son naturalmente complejos para analizar, también existe una falta de herramientas que permitan manipular el texto en idioma español. Durante la fase de modelado, se tradujeron los documentos al inglés en pos de ampliar el conjunto de librerías de *python* disponibles para modelado. No obstante, se descubre que las peculiaridades del lenguaje limitan la calidad de las traducciones. Estas últimas serán tan buenas como la correctitud gramatical del texto. Para futuras iteraciones, se debería trabajar aún más en la normalización de los mensajes.

Más adelante, se puede comentar acerca de la ausencia de datos etiquetados. A lo largo del trabajo, se tuvieron que generar datasets para poder llevar a cabo el modelado de algunos de los *features* considerados. Aunque se podrían haber empleado librerías como Snorkel[23] para obtener muestras de mayor tamaño reduciendo el trabajo manual, la prioridad fue obtener una muestra acotada asegurando la calidad de las etiquetas.

De todas formas, la mayoría de las librerías que sirven para dicho proceso requieren la especificación de funciones de etiquetado. Puesto que al crear los datasets de entrenamiento se enfrentaron casos donde existía cierto grado de ambigüedad, el uso de estas herramientas no parece trivial. No obstante, aumentar el tamaño muestral podría potencialmente generar un aumento en el rendimiento de la probabilidad de requerir atención al cliente y, también, en el ensamble examinado durante el análisis de sentimiento.

Por último, se mencionan limitaciones propias del proceso de modelado. Si bien los análisis realizados a lo largo del trabajo deben ser actualizados, esto es estrictamente necesario para la inferencia de tópicos. A pesar de que la intención fue modelar las temáticas recurrentes, de no ajustar el parámetro *k*, no se podrán incorporar potenciales nuevos tópicos. Aunque no se espera que la aparición de nuevos temas sea frecuente, si lo fuese, se debería seleccionar otro tipo de algoritmo capaz de manejar mejor el dinamismo de los tópicos.

El valor que aporten los *features* estudiados en este trabajo depende, inevitablemente, de los cambios que sufran los *tweets* a lo largo del tiempo. A medida que la información se modifique, se deberán ajustar los modelos para que la herramienta de negocio propuesta no se vuelva obsoleta.

[23] Ver https://www.snorkel.org/ para la documentación de la librería Snorkel.

Para continuar, se detallan algunas potenciales extensiones del modelado de datos planteado para: (i) el análisis de sentimiento, (ii) la inferencia de tópicos y (iii) la extracción de *features* de relevancia.

Considerando el análisis de sentimiento, podría incorporarse al ensamble propuesto una heurística adicional utilizando el léxico del Hedenometer[24]. Construído a partir de cuatro *corpus* diferentes – Twitter, Google Books, New York Times y Music Lyrics – el diccionario asocia una medición de felicidad para las 10,000 palabras de mayor frecuencia. Dicha métrica representa un promedio de puntajes que van del 1 (tristeza) al 9 (felicidad) y que son obtenidos por medio de un etiquetado manual (Amazon's Mechanical Turk).

En relación a la inferencia de tópicos, dentro del enfoque de representaciones vectoriales, sería interesante experimentar con otros algoritmos de agrupamiento como, por ejemplo, *hierarchical agglomerative clustering*[25]. Dicha técnica vincula sucesivamente observaciones bajo distintos criterios disponibles (*ward, complete linkage, average linkage* y *single linkage*), generando así relaciones de jerarquía entre los *clusters* que pueden visualizarse por medio de dendrogramas. Proporcionando información acerca de la estructura de los datos, se daría lugar al análisis de las relaciones entre tópicos así como, también, se habilitaria la formación de tópicos combinados.

Otra cuestión atractiva para explorar en futuras extensiones es la detección de comunidades. Haciendo uso de técnicas relacionadas a *network analysis*, se podrían generar *features* adicionales para iterar el ajuste de los modelos supervisados presentados a lo largo del trabajo. Sería viable partir tanto de los vínculos entre las cuentas de Twitter como de los contenidos de los mensajes de los usuarios. Incluso puede pensarse como una alternativa para resolver la inferencia del país de origen bajo otra metodología.

Otra cuestión a explorar potencialmente en futuras extensiones para mejorar el modelado propuesto a lo largo del trabajo es la detección de comunidades (*network analysis)* para poder extraer características que permitan mejorar el modelado propuesto a lo largo del trabajo. Teniendo en cuenta los vínculos entre las cuentas de Twitter (por ejemplo, seguidores y seguidos) es posible identificar *features* que

Para concluir, se discuten distintas oportunidades de experimentación que podrían surgir si la solución desarrollada estuviese implementada y en uso dentro del ecosistema de Mercado Libre. De ser este el caso, se podrían capturar métricas de desempeño (*performance*) directamente relacionadas al negocio como, por ejemplo, el tiempo de resolución de reclamos. Así, se volvería viable evaluar el comportamiento de dichos indicadores tanto de forma individual como, también, en conjunto con otros *features* de interés presentados a lo largo del trabajo.

Además, se daría lugar a nuevos análisis vinculados con el rendimiento de cada agente, especialmente si la empresa en cuestión compartiera datos internos sobre los casos de

[24] Ver https://hedonometer.org/words/labMT-es-v2/ para más detalles.
[25] En https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html se puede hallar una implementación del algoritmo.

atención al cliente. El análisis de dichos datos potencialmente aportaría valor para iterar y repensar las decisiones de modelado actuales.

Otro aspecto a explorar es la adopción de la herramienta propiamente dicha. Obtener métricas ligadas al negocio sería de utilidad para contextualizar el manejo actual de los distintos componentes del tablero. Incluso, podría diseñarse un flujo de *AB testing* para lanzar nuevas funcionalidades.

# Apéndice A

## Variables contenidas en el dataset

Debajo (Tabla A1) se muestra un detalle acerca de la cantidad de *tweets* por fechas de recolección.

| FECHA DE RECOLECCIÓN | NÚMERO DE TWEETS |
|---|---|
| 2020-12-06 | 4,814 |
| 2020-12-20 | 6,602 |
| 2021-01-04 | 6,259 |
| 2021-01-12 | 5,090 |
| 2021-01-31 | 6,448 |
| 2021-02-06 | 6,268 |
| **TOTAL** | **35,481** |

Tabla A1. Cantidad de *tweets* por día en recolección.

Las variables más relevantes contenidas en el dataset para cada *tweet* se describen a continuación (Tabla A2).

| NOMBRE | TIPO DE DATO | DESCRIPCIÓN |
|---|---|---|
| created_at | timestamp | Fecha de emisión del tweet. |
| id | integer | Número identificatorio del tweet. |
| tweet | string | Texto completo no truncado del tweet. |
| entities | json | Contiene los hashtags, símbolos y menciones de usuario. |
| in_reply_to_user_id | int | ID del usuario al que se le está contestando. |
| in_reply_to_user_screen_name | string | Nombre de usuario al que se le está contestando. |
| user | json | Contiene atributos del usuario emisor del tweet como id, nombre, nombre de usuario, ubicación, descripción, verificación de cuenta, número de seguidores y seguidos, entre otros. |
| language | str | Idioma del *tweet* |
| place | json | Contiene atributos del lugar de emisión del *tweet* como las coordenadas, el polígono del país donde se encuentra y otras características. |
| retweet_count | int | Número de retweets. |
| favorite_count | int | Número de favoritos. |
| possibly_sensitive | bool | No refiere al contenido del *tweet* sino, más bien, a la inclusión de vínculos a sitios web en el texto. |

Tabla A2. Descripción de las variables.

Nótese que las fechas de recolección no se encuentran separadas por la misma cantidad de días. Por este motivo, se generan periodos donde no existe cobertura de muestra para los *tweets* (Figura A3).


Figura A3. Cantidad de *tweets* por día en recolección.

# Apéndice B

## Generación de *tokens* para cada *tweet*

En esta sección se describe principalmente la tokenización de los tweets. Dicho proceso consiste en romper el texto en *tokens*, en ese caso palabras, que dejan lugar a distintos tipos de análisis semánticos como, por ejemplo, la evaluación del contenido de los mensajes mediante la frecuencia de palabras. Además, se discute sobre los principales desafíos encontrados y las metodologías consideradas que fueron descartadas en el proceso.

Se comienza eliminando links y menciones propios de los *tweets.* Luego, se utiliza el diccionario en español del paquete de *spacy* para poder remover las potenciales *stopwords*. Dado que dichas palabras están predefinidas, se agregan algunos términos específicos del dominio de atención al cliente y, también, aquellos que no tengan al menos 4 letras. Este último parámetro se definió de forma iterativa observando los cambios en las palabras más frecuentes y la pérdida de palabras relevantes para el dominio de interés (por ejemplo: "mal") considerando la cobertura de cada una.

Analizando el universo de palabras remanentes, se encuentran casos donde las palabras están mal escritas. Esto resulta de faltas de ortografía no intencionales (errores de tipeo*)* como intencionales en el caso de mezclas de idioma entre el español y el inglés ("espanglish").

Frente a esta problemática, se trata de enmendar las faltas de ortografía no intencionales con correctores ortográficos *open-source*. Si bien a simple vista parece una buena solución, la mayoría de estos correctores suelen rendir mejor donde el error de ortografía es la excepción y no la regla. Al aplicar estos correctores, no solo se subsanan pocos errores sino que, también, se genera un problema adicional al cambiar el sentido de muchas de las palabras.

Para poder dimensionar dicho cambio, se mide la alteración del significado sintáctico utilizando un *part-of-speech (POS) tagger.* Con esta técnica, se observa que un 30% de las palabras de la muestra sufren una modificación y, de estas, un 22% cambia su significado sintáctico. Aunque este proxy no tiene en cuenta el significado semántico de las palabras y está sujeto al desempeño del *POS tagger*, el porcentaje de palabras afectadas es lo suficientemente alto como para decidir no incluir estos correctores en el proceso de tokenización.

Otro método que podría mejorar la contabilización posterior de los *tokens* es el uso de un lematizador. No obstante, al aplicar esta técnica se observa nuevamente que muchas palabras cambian su sentido. En términos del proxy generado: un 42% de las palabras de la muestra sufren una modificación y, de esa proporción, un 30% altera su significado sintáctico.

Tratando de rescatar un potencial uso del lematizador, se utiliza un enfoque conservador y se intenta aplicar esta herramienta solamente a estructuras sintácticas relacionadas a

verbos que son, aproximadamente, un 10% del universo de palabras. Esto permitiría resolver las múltiples conjugaciones propias de los verbos. De esta manera, se lematizan sólo estas estructuras: un 48% sufre alguna alteración pero, de esa proporción, sólo un 20% (217 palabras) cambia su estructura. Si bien se pretende reducir el riesgo de modificar el sentido de las palabras, no es posible asegurar que no existan errores de medición generados por las etiquetas del *POS tagger*.

Otro desafío identificado durante este proceso fue encontrar herramientas para poder trabajar el *corpus* en idioma español. Esto refiere a la escasez de instrumentos comparado con aquellos disponibles para trabajar texto en idioma inglés.

En la tabla B se muestran algunas estadísticas relacionadas al proceso de tokenización final refiriéndose, particularmente, al número de palabras retenidas y eliminadas por *tweet*. Adicionalmente, se muestran gráficas de caja para las distribuciones completas (Figura B1 y B2).

Como puede observarse, existe una reducción de palabras promedio del 61% (20) para cada texto, reteniendo aproximadamente 12 palabras por *tweet*. Además, la tokenización implementada produce una reducción en la variación de palabras por texto.

| MÉTRICA | NÚMERO DE PALABRAS | | REDUCCIÓN DE PALABRAS | |
|---|---|---|---|---|
| | TWEET TOKENIZADO | TWEET ORIGINAL | VALOR ABSOLUTO | PORCENTAJE |
| PROMEDIO | 12.22 | 32.45 | 20.23 | 0.61 |
| DESVÍO ESTÁNDAR | 5.03 | 12.46 | 8.28 | 0.07 |
| MÍNIMO | 5 | 6 | 0 | 0.00 |
| MEDIANA | 12 | 32 | 20 | 0.62 |
| MÁXIMO | 31 | 71 | 54 | 0.85 |

Tabla B. Estadísticas descriptivas sobre el impacto generado en la cantidad de palabras de cada *tweet* por el procesamiento utilizado.

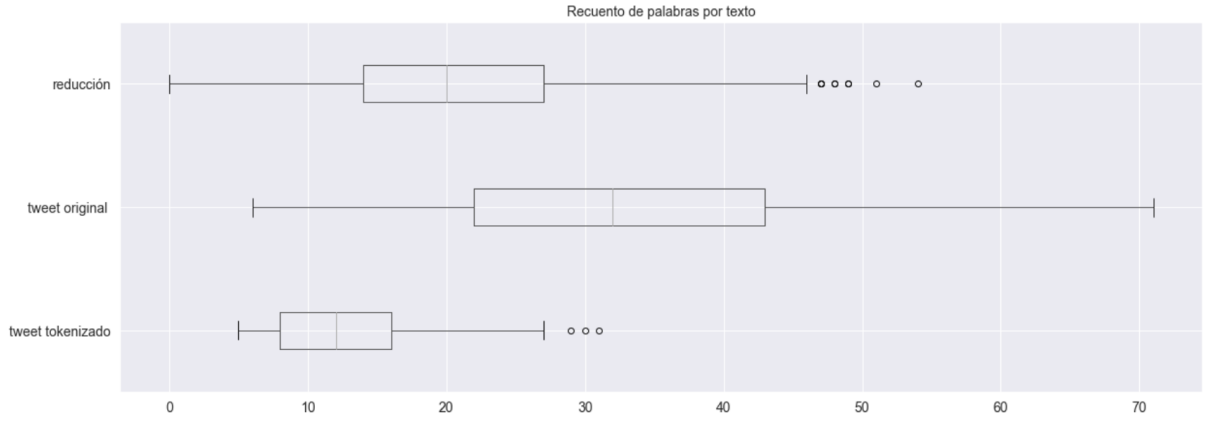

Figura B1. Gráfica de caja para los recuentos de palabras por textos en valor absoluto.


Figura B2. Gráfica de caja para la proporción de palabras reducidas por texto.

# Apéndice C

## Calidad de contenido por usuario

El objetivo de esta sección es poder describir el proceso diseñado para poder identificar la calidad de los *tweets* publicados por una cuenta cualquiera en el *corpus*. Durante la exploración de datos, se hallaron casos donde algunas cuentas publican el mismo contenido con leves modificaciones en el texto e, incluso, es posible encontrar respuestas estandarizadas que pertenecen, en su mayoría, a cuentas que brindan soporte al usuario.

Dado que estos *tweets* no aportan diversidad al contenido del *corpus,* se implementa un proceso basado en coincidencias aproximadas entre cadenas de texto. Si bien existe una gran variedad de algoritmos diseñados con este fin, algunos se adaptan mejor que otros a los casos observados en la muestra.

En esta oportunidad, se busca un puntaje de comparación que sea robusto al desorden de las palabras en los textos a comparar y que, eventualmente, pueda computar coincidencias parciales. Teniendo estos requerimientos en cuenta, se opta por utilizar la función *token_set_ratio* implementada en *python* del paquete *fuzzywuzzy*[26]*.*

Utilizando el algoritmo elegido, se evalúa la similitud de *tweets* para 4,408 cuentas con al menos una publicación (aproximadamente un 48% de los usuarios). Cuando el porcentaje de coincidencia para dos textos de una misma cuenta supera el 70%, el contenido se considera irrelevante.

La tabla C muestra estadísticas sobre el proceso utilizado al igual que las figuras C1 y C2. Esta última gráfica sólo representa el 95% de la distribución puesto que la métrica posee alta variación y los valores absolutos dificultan la visualización (escala).

| MÉTRICA | PUBLICACIONES IRRELEVANTES POR USUARIO (%) | PUBLICACIONES IRRELEVANTES POR USUARIO (#) |
|---|---|---|
| PROMEDIO | 0.37 | 4.82 |
| DESVÍO ESTÁNDAR | 0.18 | 93.42 |
| MÍNIMO | 0.03 | 1 |
| PERCENTIL 25 | 0.25 | 1 |
| MEDIANA | 0.33 | 1 |
| PERCENTIL 75 | 0.50 | 2 |
| PERCENTIL 90 | 0.65 | 4 |
| PERCENTIL 95 | 0.70 | 6 |
| MÁXIMO | 0.99 | 4011 |

[26] Documentación disponible en https://pypi.org/project/fuzzywuzzy/ .

Tabla C. Estadísticas descriptivas sobre el número de *tweets* irrelevantes por publicación.


Figura C1. Diagrama de caja para el porcentaje de textos irrelevantes por cuenta.

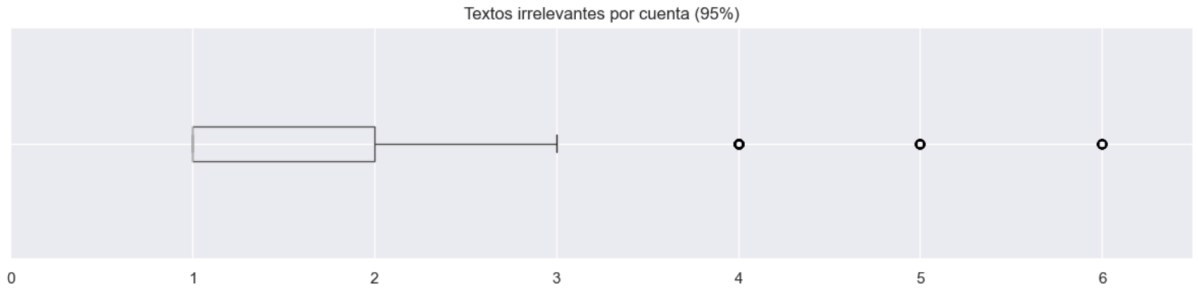


Figura C2. Diagrama de caja para la cantidad de textos irrelevantes por cuenta considerando el 95% de la distribución.

El proceso utilizado permite identificar al menos un texto irrelevante para un 42% de los usuarios, acumulando un 35% de *tweets* en la muestra. Además, se entiende que una cuenta genérica del dataset publicaría alrededor de unos 5 *tweets* irrelevantes representando aproximadamente un 37% de sus posteos.

# Apéndice D

## Algoritmos de extracción de palabras clave

Esta sección describe en forma detallada las variables generadas por cada algoritmo de extracción seleccionado. Además, se indica cuáles fueron los parámetros definidos al aplicar el procesamiento al *corpus*. Vale la pena mencionar que, dada la naturaleza no supervisada de las técnicas seleccionadas, se han iterado dichos parámetros hasta alcanzar un resultado razonable.

## RAKE

### Métricas

#### Frecuencia

Refiere a la cantidad de veces que aparece la palabra en el subset de candidatos.

(D1) $freq(w)$

#### Grado

Refiere a la cantidad de veces que aparece la palabra en conjunto con otras (co-ocurrencia) en el subset de candidatos.

(D2) $deg(w)$

#### Ratio Grado / Frecuencia

Favorece aquellas palabras que ocurren frecuentemente y en candidatos a términos clave más largos.

(D3) $deg(w) \,/\, freq(w)$

### Parámetros

#### Selección de candidatos

- mínimo de caracteres para considerar una palabra clave = 4
- máximo de palabras contenidas en un término clave = 2
- frecuencia mínima para ser considerado un término clave = 1

#### Manipulación de *stopwords*

- percentil de frecuencia de palabras para considerar *stopwords* adicionales = P80
- mínimo de caracteres para considerar un *stopword* = 1
- máximo de caracteres para considerar un *stopword* = 3

# YAKE

## Métricas

### Mayúsculas

La siguiente función de puntaje es creciente respecto a la cantidad de apariciones del término en mayúsculas.

(D4) $$T_{case} = \frac{max(TF(U(t)), TF(A(t))}{ln(TF(t))}$$

Donde

- $TF(U(t))$es el número de ocurrencias donde el término candidato *t* inicia con una mayúscula, excluyendo el inicio de las oraciones.
- $TF(A(t))$ es el número de veces que el término candidato *t* es marcado como un acrónimo.
- $TF(t)$ es la frecuencia del término candidato *t.*

### Posición

La siguiente función de puntaje considera la importancia del término candidato teniendo en cuenta que las palabras más relevantes suelen aparecer al principio de los documentos. Esto suele ser cierto en el caso de textos científicos y de aquellos que relatan noticias. El valor de (D5) incrementa a medida que el término aparece más hacia el final de los documentos.

(D5) $$T_{position} = ln(ln(3 + Median(Sen_t)))$$

Donde

- $Sen_t$ es el set de posiciones de las oraciones donde el candidato *t* occure.

Dado que la mediana puede devolver un valor de cero (candidato ocurre en una sola oración), la constante C = 3 se agrega para garantizar que el puntaje sea mayor a cero. Además, el doble logaritmo pretende suavizar diferencias extremas entre términos localizados al principio y al final del documento.

### Frecuencia Normalizada

La siguiente función de puntaje pretende capturar la relevancia del término en función de la cantidad de apariciones normalizadas. El valor de dicho cálculo incrementará a medida que la frecuencia del término esté por encima del valor promedio de ocurrencias del documento.

(D6) $$TF_{norm} = \frac{TF(t)}{MeanTF + 1 * \sigma}$$

Donde

- $TF(t)$ es el número de ocurrencias del término *t.*

- $MeanTF$ es el valor medio de las ocurrencias de los términos del documento.
- $\sigma$ es la dispersión de las ocurrencias de los términos del documento.

### Relación con el contexto

La siguiente función de puntaje tiene como objetivo determinar la dispersión de un término candidato considerando el contexto en el que se presenta. Esta fórmula sigue la idea de que cuanto mayor sea el número de términos que co-ocurren con el candidato tanto a la izquierda como a la derecha (ambos lados), menos significante será para el documento.

(D7)
$$DL\,[DR] = \frac{\left|A_{t,w}\right|}{\sum_{k \in A_{t,w}} CoOccur_{t,\,k}}$$

Donde

- $DL\,[DR]$ es la dispersión a la izquierda (derecha) del candidato *t.*
- $\left|A_{t,w}\right|$ es el número de diferentes términos que ocurren a la izquierda o derecha de los términos parseados utilizando una ventana de tamaño *w*.
- $CoOccur_{t,k}$ mide la co-ocurrencia con el término.

La ecuación D7 luego es utilizada para construir el cálculo final que intenta dar información acerca del contexto del término. Cuanto mayor sea el puntaje de (D8), menor es la relevancia del término. Es por este motivo que este *feature* en particular tiene la capacidad de distinguir *stopwords*.

(D8)
$$T_{rel} = 1 + (DL + DR) * \frac{TF(t)}{MaxTF}$$

### Número de oraciones diferentes

Esta propiedad mide qué tan frecuentemente un término candidato aparece en diferentes oraciones. Cuanto mayor sea este valor, mayor será la probabilidad de que el término candidato sea relevante.

(D9)
$$T_{sentence} = \frac{SF(t)}{\#\ sentences}$$

Donde

- $SF(t)$ es la frecuencia de aparición del término *t* en las distintas oraciones del documento.
- $\#\ sentences$ es el número de oraciones totales del documento.

### Puntaje del término

Una vez calculadas las variables que describen la relevancia del término candidato, dichas métricas se combinan para formar un único puntaje que decrece a medida que el término se vuelve importante.

(D10) $$S(t) = \frac{T_{rel} * T_{position}}{T_{case} + \frac{TF_{Norm}}{T_{rel}} + \frac{T_{Sentence}}{T_{Rel}}}$$

Es interesante mencionar el uso de $T_{rel}$ en la fórmula. Nótese que sólo se le da importancia a $T_{sentence}$ , $T_{Norm}$ y $T_{position}$ cuando el término es relevante.

Dado que esta técnica permite generar términos clave construidos por más de un término utilizando *n-grams,* en los casos donde los candidatos posean más de una palabra, se combinarán los puntajes $S(t)$ de la siguiente manera.

(D11) $$S(kw) = \frac{\prod_{t \in kw} S(t)}{KF(kw) * (1 + \sum_{t \in kw} S(t))}$$

Si sólo se tuviese en cuenta el nominador, podría ocurrir que se favorezca levemente más aquellos candidatos más largos y, por ese motivo, es que se normaliza dicho valor por la suma de todos los puntajes ponderados por la frecuencia del candidato.

## Parámetros

- máximo de palabras contenidas en un término clave (*n grams)* = 3
- ventana definida para obtener los *n grams* = 2
- función de similitud para identificar pares de candidatos similares = Levenshtein
- punto de corte para eliminar candidatos similares = 0.9

# TextRank

## Métricas

### Puntaje del vértice

En este caso, la función utilizada para calcular el puntaje de cada vértice es idéntica a la que se utiliza en PageRank solo que se aplica sobre texto en vez de *web surfing.*

(D12) $$S(V_i) = (1 - d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j)$$

Donde

- d es un valor entre 0 y 1 que representa la probabilidad de saltar aleatoriamente de un vértice a otro en el grafo.
- $In\ (V_i)$ es el set de vértices que apuntan a $V_i$ (predecesores)
- $Out\ (V_i)$ es el set de vértices a los que apunta $V_i$ (sucesores)
- $|\ Out\ (V_i)\ |$ es el número de links sucesores.

Nótese que los puntajes iniciales asignados a los vértices no deberían afectar el resultado final si la cantidad de iteraciones es suficiente para alcanzar la convergencia.

## Parámetros

- tamaño de la ventana de co-ocurrencia definida para el grafo = 3
- retención de palabras acorde a significado sintáctico (POS) = ["NOUN", "PROPN", "VERB"]*
- probabilidad *d* de saltar de un vértice a otro aleatoriamente = 0.85**
- punto de corte de convergencia = 1e-5
- iteraciones = 10

* Si bien Mihalcea y Tarau (2004) demuestran que los mejores resultados se obtienen reteniendo sólo sustantivos y adjetivos, dado que el dominio del *corpus* trabajado es de atención al cliente, se cree que podría perderse información relevante y, por ese motivo, se retienen sustantivos y verbos.
** Este valor se define utilizando el valor de referencia de PageRank.

# Apéndice E

## Análisis de sentimiento

## Manipulación de los puntajes

### Normalización

Los modelos utilizados generan puntajes de polaridad que varían en distintos rangos. Para poder interpretar y comparar fácilmente los resultados de los algoritmos elegidos, se normalizan los valores para que se encuentren en el rango [0,1]. La Tabla E1 refleja dicho proceso.

| MODELO | RANGO DE VALORES | TIPO DE NORMALIZACIÓN | FÓRMULA $f(x)$ |
|---|---|---|---|
| VADER | [-1, 1] | MAX-MIN | $(x + 1)/2$ |
| CNN RESEÑAS | [0,1] | - | - |
| CODESWITCH | [0.5, 1] | MAX-MIN | $(x - 0.5)/0.5$ |

Tabla E1. Normalización de los puntajes de acuerdo a los modelos utilizados.

### Categorización

Teniendo en cuenta la normalización previa que permite comparar los puntajes de los modelos, se propone la siguiente forma de categorizar la variable.

| PUNTAJE NORMALIZADO | CATEGORÍA |
|---|---|
| $p > 0.475$ | POSITIVO |
| $0.475 \leq p \leq 0.525$ | NEUTRAL |
| $p < 0.525$ | NEGATIVO |

Tabla E2. Categorización de los puntajes normalizados.

## Ajuste de los modelos

### Criterios de selección de muestra

En pos de poder evaluar el ajuste individual de cada uno de los modelos utilizados, se genera una muestra de 505 casos que cumple con los siguientes criterios.

| CRITERIO DE SELECCIÓN | DETALLE | CASOS (#) |
|---|---|---|
| TODOS LOS MODELOS COINCIDEN | Considerando la categorización del puntaje normalizado, seleccionamos casos donde todos los modelos coinciden / discrepan. | 50 |
| TODOS LOS MODELOS DISCREPAN | | 50 |
| CNN RESEÑAS PROBABILIDADES ALTAS | Puesto que la distribución de probabilidad | 50 |

| | | |
|---|---|---|
| CNN RESEÑAS PROBABILIDADES MODERADAS | se encuentra sumamente sesgada hacia puntajes negativos, se asegura que la muestra contenga casos positivos y neutrales. | 50 |
| ALEATORIO | Muestreo aleatorio. | 305 |
| **TODOS LOS CRITERIOS** | | **505** |

Tabla E3. Detalle de criterios de muestreo utilizados para etiquetar casos de interés.

No se seleccionan todos los casos de forma aleatoria para que se garantice representar casos relevantes en una muestra de menor tamaño.

## Proceso de etiquetado de muestra

Una vez seleccionada la muestra de casos, se procedió a etiquetar de forma manual cada uno de los *tweets* del conjunto. Para realizar esta tarea se siguieron distintas pautas para poder rotular cada texto (Tabla E4). No obstante, puesto que la revisión se realizó por una única persona, el procedimiento puede poseer cierto grado de subjetividad.

| ETIQUETA | DETALLE |
|---|---|
| NEGATIVO | ● El texto transmite inquietud / ansiedad / estrés / ironía.<br>● El texto contiene adjetivos calificativos negativos.<br>● El texto contiene insultos / groserías / lenguaje vulgar. |
| NEUTRAL | ● No posee adjetivos calificativos positivos / negativos.<br>● No es posible identificar una postura positiva / negativa en la frase evaluada sino más bien cordial. |
| POSITIVO | ● El texto transmite gratitud / aprobación / alegría / satisfacción.<br>● El texto contiene adjetivos calificativos positivos. |

Tabla E4. Pautas contempladas a la hora de etiquetar manualmente la muestra.

## Métricas de evaluación

En esta sección se amplia acerca de las funciones de pérdida utilizadas para medir el desempeño de los modelos respecto a la precisión o exactitud del valor predicho. Si bien las métricas seleccionadas son similares, utilizan distintas penalizaciones y varían en rangos diferentes.

### Brier Score Loss

Esta función de pérdida pretende medir la distancia entre las predicciones y el valor real de la observación. Se define básicamente como la diferencia cuadrática media.

(E1) $$BrierScoreLoss(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 = \frac{1}{N}\left(\sum_{y_i=0}\hat{y}_i^{\,2} + \sum_{y_i=1}(\hat{y}_i - 1)^2\right)$$

Donde

- N es la cantidad total de observaciones

- $\widehat{y_i}$ es una predicción probabilística entre 0 y 1
- $y_i$ es la clase real de la observación (binaria)

### Log Loss Score

Esta función de pérdida, también conocida como *Binary Cross Entropy*, penaliza la distancia entre el valor predicho y el real teniendo en cuenta el logaritmo.

(E2) $$LogLoss(y\ ,\widehat{y}\ ) = -\frac{1}{N}\sum_{i=1}^{n-1}\left(y_i\ log\ \widehat{y_i} + (1-y_i)\ log(1-\widehat{y_i})\right)$$

Donde

- N es la cantidad total de observaciones
- $\widehat{y_i}$ es una predicción probabilística entre 0 y 1
- $y_i$ es la clase real de la observación (binaria)

## Ensamble AutoML

### Entrenamiento

Para poder ensamblar los puntajes obtenidos de cada algoritmo se hace uso de TPOT[27], una librería de *python* que permite probar distintos modelos de aprendizaje automático en pos de hallar el mejor ajuste. Seleccionando los parámetros introducidos a continuación, el modelo realiza distintos experimentos y retiene el algoritmo que mejores métricas pudo proveer bajo las condiciones de entrenamiento.

| PARÁMETRO | DETALLE | VALOR |
|---|---|---|
| TIEMPO MÁXIMO DE BÚSQUEDA | Tiempo total que se ejecutará el algoritmo. | 60 minutos |
| PARADA TEMPRANA | Número de iteraciones máximas donde se tolera que no exista una mejora en la función de puntaje. | 3 iteraciones |
| FUNCIÓN DE PUNTAJE | Función para maximizar. | LogLoss (negativa) |
| VALIDACIÓN CRUZADA (K-FOLDS) | Número de conjuntos a utilizar (*K*) para realizar validación cruzada. | 3 |
| GENERACIONES | Dejamos el valor sugerido por defecto. | 100 |
| TAMAÑO DE POBLACIÓN | Dejamos el valor sugerido por defecto. | 100 |
| RATIO DE MUTACIÓN | Dejamos el valor sugerido por defecto. | 0.9 |
| RATIO CROSSOVER | Dejamos el valor sugerido por defecto. | 0.1 |
| ESTADO ALEATORIO | Para asegurar que se pueda reproducir el resultado. | 1 |

Tabla E5. Parámetros de entrenamiento seleccionados para hallar el mejor ensamble utilizando TPOT.

[27] La documentación acerca de la librería se encuentra en https://github.com/EpistasisLab/tpot

# Apéndice F

## Inferencia de tópicos

# Enfoque probabilístico

### Métricas de coherencia

Las métricas de coherencia, a diferencia de otras estimaciones, intentan medir la calidad de los tópicos y su interpretabilidad (Röder et al., 2015). Dentro de este conjunto, es posible hallar métricas de carácter intrínseco como extrínseco.

Debajo presentamos Umass, una métrica intrínseca que, basada en la co-ocurrencia de palabras, intenta medir qué tanto ha aprendido el algoritmo acerca de un tópico en particular.

(F1) $$score(w_i, w_j, \epsilon) = log\left(\frac{D(wi, wj) + \epsilon}{D(wj)}\right)$$

Donde

- $w_i$ y $w_j$ son palabras de un tópico en particular
- $D(w_i, w_j)$ cuenta la ocurrencia en conjunto de ambas palabras por documento
- $D(w_i)$ mide la cantidad de documentos donde se observa $w_i$

Respecto al funcionamiento de la métrica, se parte de la distribución de probabilidad que genera un tópico para las palabras contenidas dentro de un *corpus*. Luego, se seleccionan los N términos de mayor probabilidad (por ejemplo, N=20) y se realiza una segmentación en pares $(w_i, w_j)$.

Por último, se estima $score(w_i, w_j)$ como se indica en la ecuación previa (F1) y se agregan dichos valores mediante un promedio. Nótese que la ecuación no es simétrica.

A pesar de que Umass se define para un tópico en particular, se puede reportar el promedio de las estimaciones que se obtienen para *k* tópicos como métrica global. De acuerdo con este método, la calidad del modelado mejora a medida que Umass incrementa.

## LDA

#### Parámetros

Para entrenar el algoritmo de carácter no supervisado, primero se crea un diccionario de términos seleccionando aquellas palabras que ocurren en al menos 15 documentos y que, además, no aparecen en más del 80% de los documentos. Esto deja aproximadamente 1,836 *tokens*. Para elegir la cantidad de pasadas para entrenar y el número *k* de tópicos a encontrar, se varían dichos parámetros y se evalúa Umass en cada iteración. Si bien la métrica de Umass se define para un tópico en particular, en este caso la gráfica reporta el promedio de las estimaciones que se obtienen para *k* tópicos.

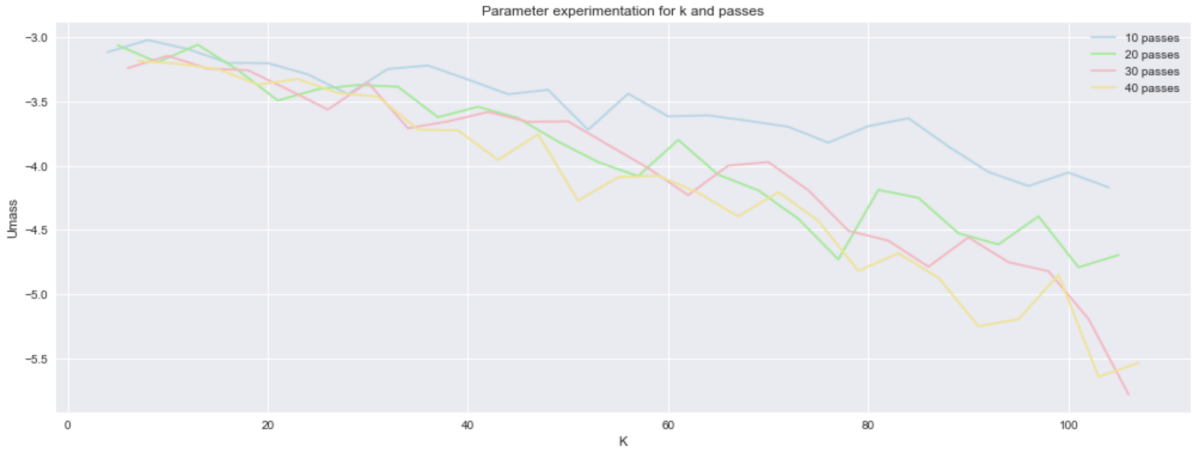


Figura F1. Evaluación de Umass acorde el número de pasadas en el *corpus* para entrenar el algoritmo y la cantidad de tópicos a encontrar.

### Resultados

Una vez entrenado el modelo (10 pasadas) con *k=6* (número de tópicos a encontrar), se vuelve factible interpretar cada uno de los tópicos en función de las palabras que los componen. Incluso, se pueden identificar las palabras más relevantes observando la distribución de probabilidad asignada por cada tópico a los términos del *corpus*.

A continuación, se introducen las palabras de mayor probabilidad por tópico encontrado durante el proceso de entrenamiento del algoritmo LDA (Tabla F2).

| TÓPICO | PALABRAS RELEVANTES |
|---|---|
| 1 | **reclamo**, vendedor, dinero, **llegó**, **devolución**, **paquete**, comprador, hacer, **compra**, nunca, compre, **devolver**, **envío** |
| 2 | **cuenta**, dinero, necesito, respuesta, **correo**, mail, **ayuda**, problema, hacer, dias, solucion, caso, ningún, **ingresar** |
| 3 | **compra**, **pago**, aparecer, **tarjeta**, hacer, necesito, **dinero**, **pagar**, **cancelar**, **credito**, **devolucion**, fecha, enero, realice |
| 4 | **envio**, productos, venta, **envios**, compra, comprar, vendedor, hacer, **domicilio**, **entrega**, gratis, dias, **full**, comprador, **paquete** |
| 5 | **servicio**, **cliente**, **atencion**, estan, hacen, empresa, gente, vender, plata, **resolver**, plataforma, bien, **tiempo**, **meses**, usuarios |
| 6 | **datos**, número, pide, **foto**, paginar, **identidad**, **celular**, **fotos**, error, **validar**, publicación, **telefono**, dirección, sistema, poner |

Tabla F2. Palabras relevantes por tópico obtenido utilizando k = 6 y 10 pasadas de entrenamiento.

## GuidedLDA

### Parámetros

En este caso, se formulan algunas palabras que van a actuar como semillas para cada tópico durante el entrenamiento. La idea es poder ayudar a la convergencia de los tópicos.

| TÓPICO | SEMILLAS | POTENCIAL TÓPICO |
|---|---|---|

| | | |
|---|---|---|
| **1** | factura, dinero, transferencia, operación, banco, tarjeta, crédito, devolución, compra, pago | Transacciones |
| **2** | virtual, malo, atención, ayuda, espera, soporte, chat, mensaje | Atención al cliente |
| **3** | envío, correo, demora, paquete, sucursal, domicilio, seguimiento, despacho, llegar | Correo y Envíos |
| **4** | cuenta, bloqueo, suspensión, acceso, recuperar, retención | Suspensión de cuentas |
| **5** | verificación, error, contraseña, habilitar, datos, entrar | Validación de cuentas |
| **6** | stock, productos, tienda, link, vendo, precio, oferta, beneficios, nuevo, barato | Difusión de servicios o productos |

Tabla F3. Semillas elegidas para cada potencial tópico a encontrar en la data.

### Resultados

| TÓPICO | TOP PALABRAS | ETIQUETA |
|---|---|---|
| **1** | compra, pago, dinero, devolución, comprar, tarjeta, pagar, crédito, factura, devolver, banco, hacer, operacion, hice, vendedor | Transacciones |
| **2** | mail, ayuda, caso, reclamo, esperando, respuesta, mensaje, necesito, atencion, problema, soporte, contacto, chat, espera, ningún | Atención al cliente |
| **3** | envio, compra, vendedor, paquete, correo, cancelar, llegó, envios, llegar, venta, numero, seguimiento, domicilio, comprador, entrega | Correo y Envíos |
| **4** | cuenta, dinero, necesito, días, hace, respuesta, suspendida, suspendieron, solución, meses, problema, correo, ayuda, bloqueada, hacer | Suspensión de cuentas |
| **5** | cuenta, datos, entrar, numero, usuario, celular, recuperar, foto, acceder, identidad, ingresar, hacer, validar, deja, correo | Validación de cuentas |
| **6** | productos, link, nuevo, precio, tienda, envio, vendo, stock, empresa, barato, oferta, dueno, beneficios, están, gente | Difusión de servicios o productos |

Tabla F4. Top de palabras por tópico obtenido utilizando k = 6 y N=10 pasadas de entrenamiento.

# Representación vectorial

## Doc2Vec

### Parámetros

| PARÁMETRO | DETALLE | VALOR |
|---|---|---|
| EPOCHS | Número de pasadas sobre el dataset para entrenamiento. | 30 |
| VECTOR SIZE | Tamaño del vector generado. | 96 |
| NEGATIVE SAMPLING | Cantidad de observaciones elegidas para hacer *negative sampling*. | 3 |
| WINDOW SIZE | Tamaño de la ventana en la que se considera contexto. | 4 |

| | | |
|---|---|---|
| MIN COUNT | Mínima cantidad de documentos donde debe aparecer cada palabra. | 5 |
| ALPHA | Learning rate. Dejamos el valor sugerido por defecto. | 0.025 |
| DM | Este valor controla la estructura del modelo: PV-DM (DM=1) o PV-DBOW (DM=0) | 1 |
| DM CONCAT | En este caso, se especifica cómo agrupar los word-embeddings a nivel documento. Al seleccionar DM CONCAT = 1 se aprende la mejor forma de combinarlos. | 1 |
| SEED | Para asegurar que se pueda reproducir el resultado. | 42 |

Tabla F5. Parámetros utilizados para entrenar Doc2Vec y obtener los vectores para cada documento de la muestra.

# DBSCAN

## Métricas de evaluación

Debajo se muestra la fórmula de *Silhouette* que permite medir la calidad de los *clusters* a través de la separabilidad de los *clusters* (inter-cluster distance) así como, también, de el nivel de hermeticidad de los mismos (intra-cluster distance).

(F2) $$Silhouette\ Score\ = \frac{(b-a)}{max(a,b)}$$

Donde

- *a* es el promedio de distancias intra-cluster
- *b* es el promedio de distancias inter-cluster

## Parámetros

Durante la fase de exploración de parámetros, se evalúa qué valor de *epsilon* ajusta mejor a la data (Figura F5) así como también, el efecto de variar min_samples una vez definido *epsilon=0.08* tanto en ruido y clusters (Figura F6) como en la calidad de agrupaciones (Figura F7).

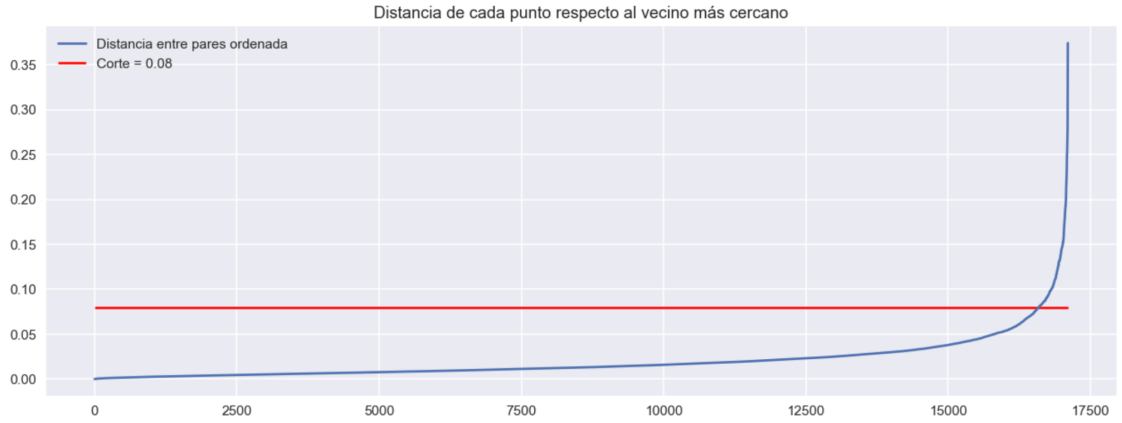


Figura F5. Distancias para cada observación respecto a su vecino más cercano ordenadas de forma ascendente.

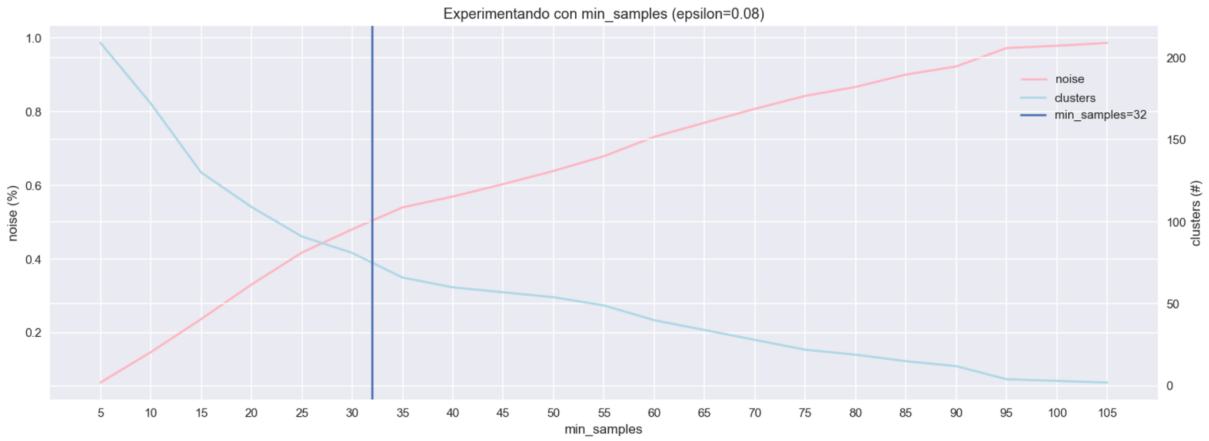


Figura F6. Efecto de variar min_samples en el número de clusters generados y el porcentaje etiquetado como ruido.

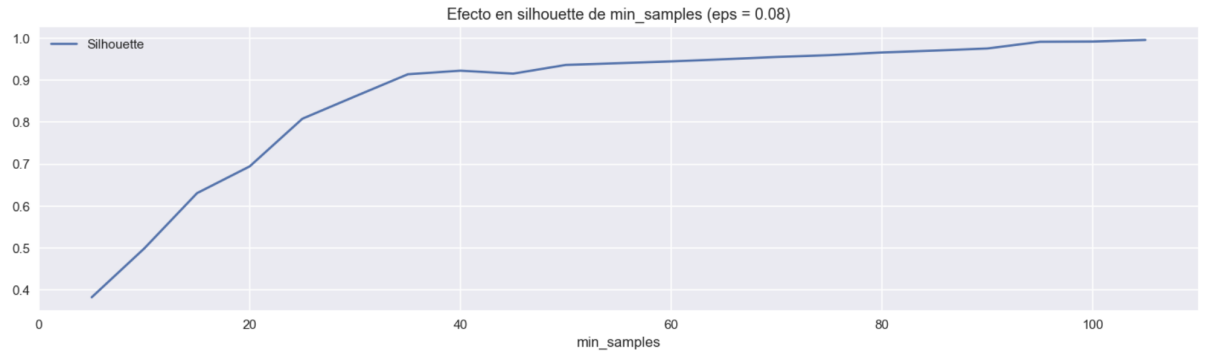


Figura F7. Valores de referencia de *silhouette* para cada uno de los min_samples seleccionados.

# Apéndice G

## Probabilidad de requerir atención personalizada

## Modelo de clasificación

En este caso, se utiliza DCCUCHILE/BERT-BASE-SPANISH-WWM-UNCASED, un modelo de BERT que ya ha sido entrenado con una gran cantidad de documentos en español (Cañete et al., 2020). Tanto el tokenizador como el modelo se ajustan levemente para poder incluir los *tokens* personalizados en pos de generar más información acerca de cada *tweet*.

Respecto al clasificador en sí, se construye una arquitectura de red neuronal partiendo del modelo BERT. Se añade una capa de *dropout* (30%) a la red original y, luego, se consolida la salida bajo una capa linear. La Figura G1 ejemplifica este proceso.

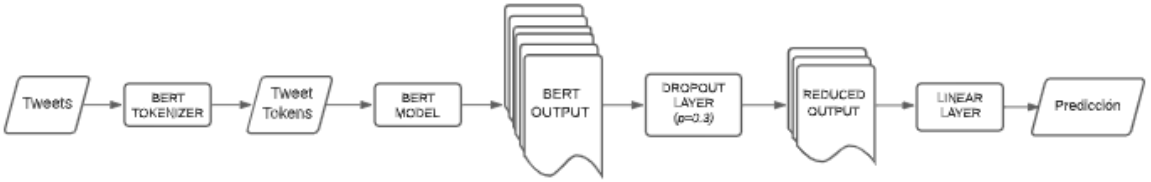


Figura G1. Proceso de modelado del clasificador para un *tweet* cualquiera.

Para la calibración de los pesos presentes en la arquitectura, se utiliza *cross-entropy* para estimar la función de pérdida. La Tabla G2 incluye otros parámetros que cobran relevancia durante el proceso de entrenamiento.

| PARÁMETRO | DETALLE | VALOR |
|---|---|---|
| EPOCHS | Número de pasadas sobre el dataset para entrenamiento. | 8 |
| BATCH SIZE | Número de lotes de datos a ser considerados. | 6 |
| LOSS FUNCTION | Función a optimizar durante el entrenamiento. | cross entropy loss |
| OPTIMIZER CLASS | Método de optimización a ser utilizado en el entrenamiento. | AdamW |
| LEARNING RATE | Define el ritmo de aprendizaje durante el entrenamiento. | 2e-5 |

Tabla G2. Parámetros más relevantes utilizados para el *fine-tuning* de BERT.

# Bibliografía


Aguilar, G., Kar, S., & Solorio, T. (2020). LinCE: A Centralized Benchmark for Linguistic Code-switching Evaluation. *Department of Computer Science, University of Houston*. https://arxiv.org/pdf/2005.04322v1.pdf

Beetrack. (2021). *Evolución del comercio electrónico: fases y futuro*. Recuperado el 10 de abril del 2022, a partir de https://www.beetrack.com/es/blog/evolucion-del-comercio-electronico

Brownlee, J. (2021). *Why Use Ensemble Learning?*. Machine Learning Mastery. Recuperado el 12 de diciembre del 2021, a partir de https://machinelearningmastery.com/why-use-ensemble-learning/

Cámara Argentina de Comercio Electrónico. (2021). *Hot Sale alcanzó 3 millones de usuarios: tendencias de la octava edición*. Recuperado el 22 de agosto del 2021, a partir de https://www.cace.org.ar/noticias-hot-sale-alcanzo-3-millones-de-usuarios-tendencias-de-la-octava-edicion

Campbell, J. C., Hindle, A., & Stroulia, E. (2015). Latent Dirichlet Allocation. *The Art and Science of Analyzing Software Data*, 139–159. doi:10.1016/b978-0-12-411519-4.00006-9

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., & Jatowt, A. (2020). YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*, 509, 257–289. https://doi.org/10.1016/j.ins.2019.09.013

Candale, C.V. (2017). Las características de las redes sociales y las posibilidades de expresión abiertas por ellas. La comunicación de los jóvenes españoles en Facebook, Twitter e Instagram. *Colindancias: Revista de la Red de Hispanistas de Europa Central*, 8, 201–218. https://dialnet.unirioja.es/servlet/articulo?codigo=6319192

Caramela, S.(2020). *How to use social media for customer service*. Business News Daily. Recuperado el 22 de agosto del 2021, a partir de https://www.businessnewsdaily.com/5917-social-media-customer-service.html

Cañete, J., Chaperon, G., Fuentes, R., Ho, J.H., Kang, H., & Pérez, J. (2020). Spanish Pre-Trained BERT Model and Evaluation Data. En PML4DC at ICLR 2020. https://users.dcc.uchile.cl/~jperez/papers/pml4dc2020.pdf

Chang, J., Gerrish S., Wang C., Boyd-Graber J., & Blei D. (2009). Reading tea leaves: How humans interpret topic models. *Advances in Neural Information Processing Systems,* 22, 288-296.https://proceedings.neurips.cc/paper/2009/file/f92586a25bb3145facd64ab20fd554ff-Paper.pdf

Chen, M. (2020). *Customer experience, artificial intelligence and machine learning*. Medium. Recuperado el 22 de agosto del 2021, a partir de

https://towardsdatascience.com/customer-experience-artificial-intelligence-and-machine-learning-748d8c1e1127

Commbox. (2021). The role of social media in customer service, a social media guide. Recuperado el 22 de agosto del 2021, a partir de https://www.commbox.io/the-role-of-social-media-in-customer-service-a-social-media-guide/

Commbox. (2021). *Artificial intelligence and deep learning for customer service.* Recuperado el 22 de agosto del 2021, a partir de https://www.commbox.io/artificial-intelligence-and-deep-learning-for-customer-service/

Compte, J. (2022). *Mercado Libre imparable: cuánto ganó en su año récord de usuarios y transacciones*. Cronista. Recuperado el 5 de abril del 2022, a partir de https://www.cronista.com/negocios/647041/

Coppola, D. (2021). *Number of digital buyers worldwide from 2014 to 2021*. Statista. Recuperado el 20 de abril del 2022, a partir de https://www.statista.com/statistics/251666/number-of-digital-buyers-worldwide/

Daumé III, H., Jagarlamudi, J., & Udupa, R. (2012). Incorporating Lexical Priors into Topic Models. *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics,* 204-213. https://aclanthology.org/E12-1021.pdf

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. https://arxiv.org/pdf/1810.04805.pdf

Ester, M., Kriegel, H., Sander, J, & X., Xu. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Institute for Computer Science, University of Munich,* 226-231. https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf

Farinha, A. F. N. (2018). *Extracting Keywords from Tweets* (Master's dissertation). http://hdl.handle.net/10400.26/28594

Farzindar, A. & Inkpen, D. (2015). Natural Language Processing for Social Media. *Synthesis Lectures on Human Language Technologies*,8, 1-166. https://doi.org/10.2200/S00659ED1V01Y201508HLT030

Godec, P. (2021). *Keyword Extraction Methods — The Overview - Towards Data Science*. Medium.https://towardsdatascience.com/keyword-extraction-methods-the-overview-35557350f8bb

Hutto, C. J., & Gilbert, E. (2015). *VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text*. Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media, Ann Arbor, Michigan. https://www.researchgate.net/publication/275828927_VADER_A_Parsimonious_Rule-based_Model_for_Sentiment_Analysis_of_Social_Media_Text

McInnes L., Healy, J., & Melville J. (2020). *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. https://arxiv.org/pdf/1802.03426.pdf

MELI. (2021). *Historia de Mercado Libre: nuestros primeros pasos, nuestro recorrido*. Mercado Libre. Recuperado el 23 de agosto del 2021, a partir de https://www.mercadolibre.com.ar/institucional/somos/historia-de-mercado-libre

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing Order into Text. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 404–411. https://aclanthology.org/W04-3252

Orús, A. (2022). *Porcentaje de compradores online a nivel mundial en 2021*. Statista. Recuperado el 20 de abril del 2022, a partir de https://es.statista.com/estadisticas/1243580/frecuencia-de-comra-online-a-nivel-mundial/

Rao, P. (2019). *Fine-grained Sentiment Analysis in Python (Part 1) - Towards Data Science*. Medium. Recuperado el 24 de abril del 2021, a partir de https://towardsdatascience.com/fine-grained-sentiment-analysis-in-python-part-1-2697bb111ed4

Röder, M., Both, A., & Hinneburg, A. (2015). Exploring the Space of Topic Coherence Measures. *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. https://doi.org/10.1145/2684822.2685324

Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010). Automatic Keyword Extraction from Individual Documents. In M. W. Berry & J. Kogan (Eds.), *Text Mining: Applications and Theory,* 1–20. Wiley. https://doi.org/10.1002/9780470689646.ch1

Salesforce. (2020). *What are customer expectations, and how have they changed?* Salesforce. Recuperado el 22 de agosto del 2021, a partir de https://www.salesforce.com/resources/articles/customer-expectations/?sfdc-redirect=369

Sivarajah, S. (2021). *Dimensionality reduction for data visualization: PCA vs TSNE vs UMAP vs LDA*. Medium. Recuperado el 29 de enero del 2022, from https://towardsdatascience.com/dimensionality-reduction-for-data-visualization-pca-vs-tsne-vs-umap-be4aa7b1cb29

Walker. (2017). *Customers 2020: a progress report more insight for a new decade*. https://walkerinfo.com/docs/WALKER-Customers2020-ProgressReport.pdf

Zendesk. (2019). *How is machine learning being used in customer service?*. Recuperado el 22 de agosto del 2021, a partir de https://www.zendesk.com.mx/blog/machine-learning-used-customer-service/