UNIVERSIDAD
TORCUATO DI TELLA

Departamento de Economía
Tipo de documento: Tesis de maestría

CC BY NC SA

Maestría en Economía

ChatGPT vs. Modelos de Tradicionales de Machine Learning para Sentiment Analysis

Autoría: Harari, Nicolás
Fecha: 2025

¿Cómo citar este trabajo?

Harari, N. (2025). "ChatGPT vs. Modelos de Tradicionales de Machine Learning para Sentiment Analysis". [Tesis de maestría. Universidad Torcuato Di Tella]. Repositorio Digital Universidad Torcuato Di Tella

https://repositorio.utdt.edu/handle/20.500.13098/13585

El presente documento se encuentra alojado en el Repositorio Digital de la **Universidad Torcuato Di Tella** bajo una licencia Creative Commons Atribución-No Comercial-Compartir Igual 4.0 Internacional
Dirección: https://repositorio.utdt.edu

Biblioteca Di Tella

UNIVERSIDAD TORCUATO DI TELLA

DEPARTAMENTO DE ECONOMÍA

MAESTRÍA EN ECONOMÍA

ChatGPT vs. Modelos de Tradicionales de Machine Learning para Sentiment Analysis

Alumno: Nicolás Harari

Tutor: Andrés Gago

Fecha: Junio 2025

ChatGPT vs. Modelos de Tradicionales de *Machine Learning* para Sentiment Analysis

Nicolás Harari

Junio 2025

Resummen: Los datos textuales se utilizan cada vez más en economía. Desarrollos recientes como los *Large Language Models* (LLMs) presentan nuevas y emocionantes alternativas para extraer datos de texto. En este trabajo, comparo el rendimiento de ChatGPT3.5 con modelos de aprendizaje automático anteriores como Regresiones Logísticas Regularizadas y Máquinas de Vectores de Soporte. A diferencia de estas últimas, ChatGPT no requiere de entrenamiento previo, pero no es de código abierto; es opaco, costoso y relativamente lento de usar en comparación con los modelos tradicionales. Encuentro que el rendimiento de ChatGPT 3.5 es similar al de modelos simples sin necesitar de realizar entrenamiento previo ni de etiquetado de datos. Esto es especialmente interesante dado que la base de datos utilizada está en portugués, un idioma para el cual los recursos de análisis de datos no están tan extendidos.

Palabras clave: LLMs, Sentiment Analysis, ChatGPT3.5, Lasso, Ridge, SVM.

1. Introducción

Los datos textuales son ubicuos en el mundo moderno. En economía y finanzas, representan una fuente central para estudios en diferentes campos. Hay quienes leen registros históricos para proporcionar un "enfoque narrativo" al análisis de políticas como Romer and Romer (1989, 2023), aquellos que buscan manualmente registros policiales (Fryer, 2019) o reseñas de Yelp (Davis et al., 2019) para explorar diferencias raciales, o aquellos que utilizan noticias para entender factores de movimientos de precios de acciones (Engelberg and Parsons, 2016; Cutler et al., 1989).

Como los datos textuales son, generalmente, no estructurados analizarlos y categorizarlos manualmente es difícil. Por esta razón, durante los últimos 25 años, el uso de algoritmos informáticos para extraer información almacenada en estos textos ha aumentado considerablemente. Una de las aplicaciones más populares es la posibilidad de obtener el sentimiento o emoción —de ahora en adelante, Análisis de Sentimientos (SA, *Sentiment Analisis* en inglés)— de varias fuentes. Esto se puede hacer de múltiples maneras, pero las más comunes son soluciones basadas en diccionarios y diferentes modelos de aprendizaje automático. Recientemente, los *Large Language Models* (LLMs)

como ChatGPT han surgido como posibles alternativas, aunque sus beneficios sobre los métodos tradicionales aún deben ser probados.

Por lo tanto, el propósito de este trabajo es comparar el rendimiento de ChatGPT3.5 [1] fin con el de modelos de aprendizaje automático más establecidos. Mi motivación es principalmente práctica: ChatGPT como herramienta no es de código abierto; es opaco, costoso y relativamente lento de usar. A su favor, ChatGPT no requiere de entrenamiento previo ni de datos etiquetados. Esto puede justificar su uso, si su rendimiento equipara o supera a los modelos tradicionales. Para probar esto, utilizo una base de datos de libre acceso que contiene reseñas con puntajes y texto de usuarios de un mercado en línea. Al final, mis resultados no son muy diferentes de experimentos mucho más exhaustivos realizados por Wang et al. (2023) y Zhang et al. (2023), quienes encontraron que si bien los LLMs son buenos para el Análisis de Sentimientos, no superan (para el año 2023) a los modelos actuales del estado del arte. Sin embargo, mi resultado es más extremo porque los modelos que utilizo son menos sofisticados.

Como se mencionó anteriormente, hay una gran cantidad de diferentes modelos de aprendizaje automático y enfoques para extraer datos de texto no estructurado. La forma más básica, y a menudo muy efectiva, es usar algún diccionario pre-elaborado de palabras anotadas con su *polaridad* —ya sea que indiquen opiniones positivas o negativas— y ver cuántas veces aparecen en el documento. Para muchas aplicaciones, este método es suficiente y produce excelentes resultados. Solo para nombrar algunos, Fraccaroli et al. (2022) usan y comparan tres diccionarios diferentes para explorar el vínculo entre la ideología política y el sentimiento con respecto al banco central, Koch et al. (2022) lo usan para entender el impacto de la opinión en las noticias en los movimientos de acciones después del BREXIT y Esposito et al. (2023) utilizan una lista hecha a mano de palabras patrióticas y divisivas para ver los efectos de *Birth of a Nation* en la narrativa de la Reconciliación. Como algunas palabras pueden tener un sentimiento negativo o positivo adjunto en algunos contextos pero ser neutrales en otros —como la palabra "*deuda*" en finanzas— hay diccionarios especializados para diferentes aplicaciones, como el seminal documento de Loughran y McDonald de 2011 (Ver también Hanna et al., 2020; Obaid and Pukthuanthong, 2022; Dybowski and Adämmer, 2018).

Estos enfoques son rápidos y fáciles de hacer en idiomas como el inglés, donde los recursos son abundantes. Sin embargo para otros idiomas esta no es una alternativa. En portugués tiene pocos de estos diccionarios generales y virtualmente ninguno especializado. Como menciona Pereira (2021), traducir estos diccionarios palabra por palabra generalmente no es una buena idea, ya que se puede perder el contexto. En esas aplicaciones, una posible alternativa es construir un diccionario de clasificación desde cero, lo cual es costoso y requiere conocimiento tanto del idioma como del tema.

[1] Los resutlados expuestos en este trabajo se obtuvieron utilizando el ChatGPT3.5 en noviembre de 2023. En los últimos años el avance de estas herramientas ha sido considerable. Al momento de la entrega de este trabajo, el modelo más simple ofrecido por OpenAI en su plataforma es GPT-4.1 mini, que es más poderoso que el utilizado para este trabajo. Al día de la fecha, estos resultados son una cota inferior al rendimiento de estos modelos.

Otro camino es posible cuando se proporciona algún tipo de métrica de clasificación junto con los datos textuales. Las reseñas en línea son un ejemplo perfecto de esto, incluso se utilizan en textos introductorios para Procesamiento del Lenguaje Natural como Bird et al. (2009) (en este caso para reseñas de películas) y Zheng and Casari (2018) con datos de Yelp. El uso de técnicas como *bag-of-words* y *frecuencia de términos – frecuencia inversa de documentos* (de ahora en adelante, *tf-idf*, utilizado y explicado con más detalle en la Sección 3), proporciona una limpieza de datos necesaria, se puede utilizar para extraer la información necesaria para entrenar modelos de aprendizaje automático que pueden ser útiles fuera de la muestra. Como generalmente estos tipos de clasificación de texto son problemas lineales (Joachims, 1998), la mayoría de los trabajos (simples) en estos utilizan modelos lineales como Máquinas de Vectores de Soporte (SVM), Modelos de Regresión Lineal o Logística Regularizados como Lasso/-Ridge o Naïve Bayes.

Sin embargo, con el reciente aumento de los LLMs, existe una tercera alternativa. ChatGPT de OpenAI (y sus competidores) proporciona una interfaz general donde los investigadores pueden preguntar a este "asistente virtual" para clasificar, extraer o interpretar datos textuales. El trabajo con estos modelos no está tan extendido pero está ganando tracción ya que el potencial reclamado es masivo. Algunos ejemplos recientes incluyen Liang et al. (2023), que prueba la capacidad de GPT-4 para proporcionar comentarios sobre artículos de investigación, Fatouros et al. (2023), que utiliza ChatGPT 3.5 para análisis de sentimientos en un contexto financiero y Lopez-Lira and Tang (2023), quien compara varios LLMs en análisis de sentimientos y pronósticos de rendimiento. El desafío inherente en casi todo el trabajo con estas plataformas es que su interfaz de usuario es radicalmente diferente de otros modelos de aprendizaje automático. Enfoques más tradicionales a veces pueden categorizarse como "cajas negras" —en el sentido de que los parámetros del modelo no son fácilmente comprensibles para el investigador y que existen opciones amigables para el usuario donde no siempre es necesario codificar las funciones reales— pero son, sin embargo, resultados de entrenar algún modelo con los datos de entrenamiento y ver sus resultados con los datos de prueba separados. Esto es radicalmente diferente con ChatGPT: aquí el código es "la indicación", (el texto que el investigador alimenta al asistente virtual) los datos de prueba no se conocen (ya que son propietarios) y los datos de prueba son el conjunto de datos del investigador. Ampliaré más sobre estos problemas en el resto de las siguientes secciones.

Este trabajo está inspirado en el trabajo que realicé como asistente de investigación bajo Andrés Gago para el artículo que coescribió con Jose María Abad, Vicente Bermejo y Felipe Carozzi (2023). Allí utilizan datos de noticias de España durante la Gran Recesión para ver si los incumbentes municipales reelegidos son más reacios que los funcionarios recién elegidos a solicitar ayuda del gobierno central porque hacerlo se vería como una admisión de incompetencia. Como una extensión de su modelo, codifiqué una aplicación ChatGPT en `python` que utiliza el chatbot para extraer información clave sobre las consideraciones mediáticas de estos políticos locales. Uso ese código aquí con algunos cambios menores.

El uso de aprendizaje automático en reseñas en línea es un problema estándar en la literatura. Solo para nombrar dos ejemplos que utilizan el mismo tipo de modelos que estoy aplicando en este trabajo, Jabbar et al. (2019) utilizan SVM para construir una aplicación de análisis de sentimientos en tiempo real y Lin (2020) extrae datos utilizando *tf-idf* para luego comparar modelos de regresión logística regularizados, SVM y Random Forest.

Este trabajo está ordenado de la siguiente manera. En la primera sección muestro el conjunto de datos utilizado en esta aplicación y algunos hechos empíricos sobre las reseñas, a saber, que la inclusión de una reseña escrita está correlacionada con una reducción de alrededor de $0.5/5$ en el puntaje (o estrellas) que un consumidor deja en el producto. En la tercera sección muestro la limpieza de datos necesaria para ejecutar los modelos de aprendizaje automático, así como explico su uso y motivación. En la cuarta sección discuto los pasos necesarios para ejecutar el programa mencionado anteriormente con ChatGPT, las indicaciones utilizadas y los diferentes parámetros utilizados. En la quinta sección muestro los resultados de los tres modelos y los comparo.

Todo el código utilizado en este trabajo está escrito en `python` y está disponible bajo solicitud.

2. Descripción de los datos

En este trabajo utilizo el conjunto de datos proporcionado por Olist and Sionek (2018), disponible a través de Kaggle. Contiene alrededor de cien mil pedidos realizados en el sitio de comercio electrónico brasileño Olist, una plataforma que proporciona sitios web para pequeños minoristas que venden sus productos en grandes mercados en línea, como Amazon, Ebay o Mercado Livre. Este conjunto de datos es extenso: combina información sobre el pedido —como el precio y el cargo de flete, el método de pago, el tiempo de envío (por nombrar algunos) con información sobre vendedores y compradores. Sin embargo, el enfoque principal de este trabajo son las reseñas, tanto basadas en puntuaciones como en texto.

En el comercio minorista en línea, especialmente cuando se vende directamente a los consumidores, las reseñas son un aspecto clave de la adquisición de clientes. En ausencia de un espacio físico donde el cliente pueda examinar el producto, las reseñas en línea funcionan como el boca a boca virtual: ayudan a los consumidores a calibrar sus expectativas, evitar malas compras o convencerlos de cerrar un trato (Dellarocas, 2003). Esto es especialmente importante para bienes *menos* populares (Zhu and Zhang, 2010).

Combino todos los conjuntos de datos y limpio los datos para su análisis. Primero, por razones de seguridad, los datos de geolocalización para consumidores y vendedores se proporcionan en forma de los primeros dígitos de su código postal. El conjunto de datos de geolocalización contiene varios valores de latitud y longitud para cada

	order id	seller id	category	review text	review score	has text	days to deliver	total price per houndred
número	103363	103363	101859	44680	103363.00	103363.00	103363.00	103363.00
únicas	97634	3090	73	37339	-	-	-	-
media	-	-	-	-	4.08	0.43	16.18	1.56
desvío	-	-	-	-	1.35	0.50	29.76	2.16
mínimo	-	-	-	-	1.00	0.00	0.00	0.00
50 %	-	-	-	-	5.00	0.00	10.00	1.01
máximo	-	-	-	-	5.00	1.00	208.00	136.64

Cuadro 1: Estadísticas descriptivas para el conjunto de datos final después de la limpieza. De izquierda a derecha, se encuentra el identificador único de pedido y vendedor y la categoría (en portugués) de los productos en los pedidos. `Review text` almacena tanto el título como el cuerpo de la reseña textual dejada por el usuario, mientras que `review score` almacena la puntuación o revisión numérica. A continuación, `has text` es una variable booleana que toma uno siempre que el usuario deje una reseña textual en el pedido. Finalmente, `days to deliver` representa los días que realmente tomó el paquete para llegar al cliente y `total price per houndred` es el precio pagado por el cliente en cientos de reales, incluido el envío.

código postal. Los promedio para obtener una estimación general para cada ubicación donde se realizaron los pedidos. Como se puede ver en Figura 1, hay pedidos que no se realizaron dentro del territorio continental de Brasil. Como método de fitlrado, aproximo la forma de brasil por un rectángulo y elimino todas las observaciones del conjunto de datos que no se ajustan a él. Luego, combino las observaciones restantes en los conjuntos de datos de pedidos, pagos, consumidores, vendedores y reseñas.

Figura 1: Distribución geográfica de la ubicación promedio asociada con cada código postal. La figura de la izquierda muestra todos los códigos postales en la base de datos. A la derecha, solo se grafican los códigos postales de los consumidores en la zona general de Brasil.

Para las reseñas, combino los títulos y el texto principal en una sola entrada para facilitar su uso. Respecto a las variables temporales, *Olist* tiene una marca de tiempo de compra, entrega y entrega esperada. Utilizo estas variables para generar variables que muestren el tiempo esperado hasta que se entregue el producto y la diferencia entre el tiempo esperado y el tiempo real de entrega. Para aquellas observaciones en las que el artículo no se entregó, se imputa el máximo de días de espera en los datos. Las estadísticas descriptivas para el conjunto de datos final resultante se pueden encontrar en Tabla 1.

2.1. Reseñas textuales y numéricas

En este apartado me centro en las reseñas dejadas por los clientes. Como se indicó anteriormente, estas se dividen en dos partes: una basada en puntuación (o estrellas) que va de 1 a 5 y un comentario escrito. En el gráfico izquierdo en Figura 2, muestro la distribución de las reseñas de puntuación entre todas las observaciones. La mayoría de ellas son reseñas de cinco estrellas o "excelente" (casi el 60 %), seguidas por 4 estrellas (alrededor del 19 %), luego, en orden, 1 estrella (11 %), 3 estrellas (8 %) y 2 estrellas (3 %). Respecto de las reseñas escritas, este patrón cambia ligeramente, ya que no todas las entradas que tienen reseñas de puntuación contienen comentarios escritos. En el el gráfico derecho en Figura 2, se muestra que la distribución de las puntuaciones en el subconjunto que tiene texto está significativamente sesgada a la izquierda, siendo las reseñas negativas de una estrella las segundas más populares. Una prueba de U de Mann-Whitney me permite rechazar la hipótesis de que las distribuciones de las reseñas de puntuación con y sin texto son las mismas. Esto parece indicar que la presencia de reseñas escritas es un signo de incomodidad del consumidor.

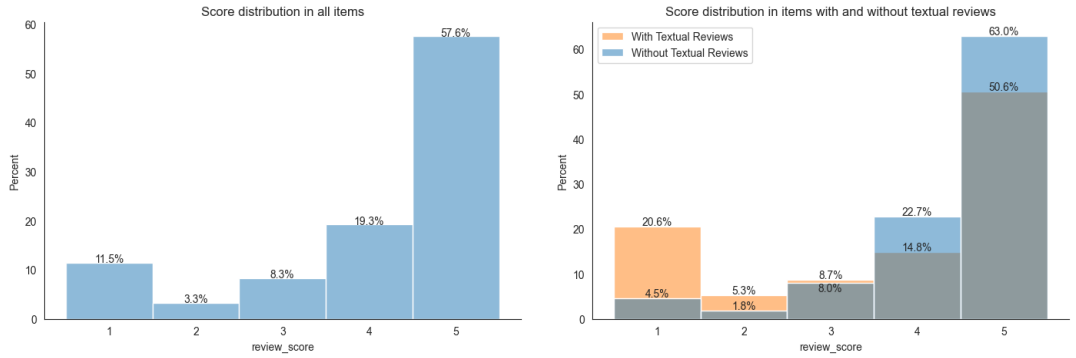

Figura 2: Histogramas para las reseñas de puntuación numérica para todas las reseñas en el subconjunto final (izquierda) y diferenciando entre reseñas con y sin texto (derecha)

Para comprender mejor la relación entre la puntuación de la reseña y la presencia de una reseña de texto, corro la siguiente regresión:

$$\begin{aligned} \text{review score}_i = \beta_0 + \beta_1 \text{has text}_i + \beta_2 \text{total price per houndred}_i \\ + \beta_3 \text{days to deliver}_i + \epsilon_i \end{aligned} \tag{1}$$

donde review score$_i$ representa el número de estrellas que dejó el usuario (de 1 a 5), has text$_i$ es 1 cuando la reseña contiene datos textuales, total price es el precio combinado del flete y el producto del pedido y days to deliver$_i$ el tiempo de espera a ser entregado.

Todos los coeficientes son estadísticamente significativos y, como era de esperar con la interpretación de que las reseñas escritas tienden a ser negativas, sus valores son negativos. El mayor efecto corresponde a la presencia de datos textuales, que se traduce en una reducción promedio de 0.7 en la reseña de estrellas. En primer lugar, los consumidores penalizan el tiempo total: un mes reduce la calificación promedio en medio punto de estrella. Por otro lado, a mayor precio peores son las reseñas: un au-

	Dependent variable: review score
	(1)
Intercept	4.631***
	(0.012)
days to deliver	-0.016***
	(0.000)
delivered minus expected	0.001***
	(0.000)
has text	-0.603***
	(0.008)
total price per houndred	-0.017***
	(0.002)
Observations	103363
R^2	0.164
Adjusted R^2	0.164
Residual Std. Error	1.233 (df=103358)
F Statistic	5054.513*** (df=4; 103358)

Cuadro 2: Resultados para la regresión OLS en la ecuación Ecuación 1. Las segunda y tercera columnas muestran efectos fijos para vendedores y categoría de productos respectivamente (*p<0.1; **p<0.05; ***p<0.01).

mento del precio del producto de 3000 reales equivale a la pérdida de media estrella (alrededor de 800 dólares en la cotización promedio de 2018). Como verificación de robustez, también ejecuto esta regresión con efectos fijos en vendedores y productos, encontrando que prácticamente no hay diferencia en los resultados. Estos se pueden encontrar en Tabla 2.

Como último paso, hago el experimento inverso: verifico si éstas variables tienen algún efecto en la probabilidad de escribir una reseña. Para esto, corro el siguente modelo *probit*:

$$\begin{aligned}\text{has text}_i = \Phi[&\beta_0 + \beta_1 \text{review score}_i + \beta_2 \text{total price}_i \\ &+ \beta_3 \text{days to deliver}_i + \epsilon_i]\end{aligned} \tag{2}$$

donde Φ es la función de distribución acumulada de la distribución normal estándar. Los resultados se pueden encontrar en Tabla 3. Como era de esperar, una buena puntuación de estrellas disminuye la probabilidad de encontrar una reseña escrita, mientras que el precio del producto hace aumentar tiene el efecto contrario. Para finalizar podemos ver que la espera no influye en la probabilidad en la que los usuarios dejen reseñas textuales.

3. Análisis de Sentimiento de la Manera Tradicional

Como se indicó anteriormente, en el conjunto de datos de *Olist*, las opiniones de los clientes aparecen de dos formas distintas:

1. Como una puntuación numérica, es decir, el número de "estrellas" y,
2. una reseña basada en texto **opcional**.

	Dependent variable: has text
	(1)
Intercept	0.772***
	(0.015)
days to deliver	0.000
	(0.000)
review score	-0.239***
	(0.003)
total price per houndred	0.021***
	(0.002)
Observations	103363

Cuadro 3: Resultados para la regresión Probit en Ecuación 2.(*p<0.1; **p<0.05; ***p<0.01).

Utilizaré estas métricas para entrenar y comparar modelos de Regresión Logística Regularizada (Lasso/Ridge) y Máquinas de Vectores de Soporte para extraer la opinión del cliente de los datos. Como se indicó en la introducción, este enfoque es especialmente útil al trabajar con idiomas donde no está disponible un diccionario preconstruido, siendo el portugués un ejemplo principal (Pereira, 2021). Esta sección se organiza de la siguiente manera: primero, explico los pasos que tomo para limpiar los datos textuales y la normalización aplicada antes de ejecutar los modelos. Después, justifico los modelos que elegí en función de la literatura existente. Es importante remarcar que estos pasos, tomados para entrenar los modelos Lasso/Ridge y SVM, no son necesarios para ChatGPT, que tokeniza los datos de forma independiente.

3.1. Limpieza de Datos e Ingeniería de Características

Para los modelos de aprendizaje automático tradicionales, divido el conjunto de datos y entreno un modelo para predecir la polaridad de las reseñas utilizando las puntuaciones como objetivo. Para esto, defino la variable *positivo* como

$$\text{positivo}_i = \begin{cases} 1 & \text{si } review_i \geq 4 \\ 0 & \text{si } review_i \leq 3 \end{cases}$$

para reducir las categorías a entrenar.

Continúo con la limpieza de las reseñas. En Figura 3, se puede ver un ejemplo con algunos problemas comunes: letras acentuadas y emojis. Otros pueden incluso hacer referencia a fechas, dinero o jerga. Todos estos son problemáticos y su tratamiento no es trivial.

> Cada vez que compro más fico satisfecha parabéns pela honestidade com seus clientes 👏👏👏👏👏👏👏👏

Figura 3: Una muestra de reseña con emojis y acentos.

La ruta más sencilla a seguir es eliminar todos los elementos no textuales del texto por completo. Sin embargo, esto implicaría perder información valiosa. Por lo tanto, sigo (y expando) los pasos que Sangani (2021) muestra en su publicación de blog para salvar estos datos:

1. Convierto todos los emojis a su descripción en portugués.
2. Luego, reemplazo todas las referencias de dinero (es decir, R$4) con la palabra portuguesa "dinheiro".
3. Reemplazo todas las referencias a fechas (es decir, 1/2/2003) con la palabra portuguesa "fecha".
4. Siguiendo a Pereira (2021), reemplazo modismos, jerga o abreviaturas comunes en portugués brasileño con palabras más descriptivas, es decir, "kkk" (literalmente "jajaja"), se reemplaza por "rir", o risa.
5. Luego, elimino todos los signos de puntuación y letras duplicadas (ya que no son válidas en portugués). Como los acentos son importantes en este idioma, decido no reemplazarlos.
6. Utilizo una lista elaborada por `nltk` para eliminar todas las palabras que no añaden ningún significado semántico, conocidas en la literatura como *stopwords* (Bird et al., 2009).
7. Finalmente, utilizando una función también de `nltk`, transformo todas las palabras en sus formas de raíz (*stemming*) para reducir la dimensionalidad de los datos. De esta manera, en lugar de tener dos entradas separadas para "reir" y "risa", las transformamos en "rei" (Zheng and Casari, 2018; Bird et al., 2009).

En Figura 4, presento el resultado de aplicar este procedimiento al ejemplo anterior.

```
Demojize:
Cada vez que compro más fico satisfecha parabens pela honestidade
com seus clientes
:maos_aplaudindo::maos_aplaudindo::maos_aplaudindo::maos_aplaudindo:
Keeps only:
Cada vez que compro más fico satisfecha parabens pela honestidade
com seus clientes
maos aplaudindo maos aplaudindo maos aplaudindo maos aplaudindo
Stems the sentence:
cad vez compr fic satisfeit parabens pela honest client mao
aplaud mao aplaud mao aplaud mao aplaud
```

Figura 4: El resultado de aplicar los pasos de limpieza de datos en una muestra de reseña.

El siguiente paso es elegir cómo representar numéricamente los datos en nuestro modelo. Como Zheng and Casari (2018); Bird et al. (2009) recopilan, hay dos grandes (y simples) estrategias para esto en la literatura. Primero, *bag of words*, conceptualiza cada documento como un vector que indica si la palabra dada aparece o no en el texto. Aquí, todas las palabras se cuentan igual. El problema con este enfoque es que hay

palabras que, aunque no son *stopwords*, aparecen en muchos documentos — y por lo tanto proporcionan poca información sobre la naturaleza de un documento dado. Un ejemplo directo es la palabra *"produto"* (producto), una palabra *a priori* neutral que puede aparecer en reseñas positivas o negativas y que proporciona poca información sobre el sentimiento del comprador. Por lo tanto, una forma de controlar la *importancia relativa* que cada palabra tiene en cada documento es usar la segunda estrategia, *tf-idf*. Esto pondera la relevancia de cada palabra considerando con qué frecuencia aparece en un documento sobre su presencia general en todos los documentos. La función viene con el paquete de aprendizaje automático `sktlearn` y se define como

$$\text{tf-idf}(pal, doc) = (\text{veces que } pal \text{ aparece en } doc) \times \left[\log \left(\frac{1+n}{1+(\substack{\text{documentos en que} \\ pal \text{ apparece}})} \right) + 1 \right]$$

Por estas razones, elijo la segunda alternativa. En Tabla 4, muestro el efecto de aplicar esta métrica de regularización en una entrada pre-limpiada.

	estipul	bem	ant	receb	praz
tf-idf	0.693235	0.427493	0.351994	0.345367	0.305773

Cuadro 4: Ejemplo de aplicación del proceso de regularización *tf-idf* a una entrada limpia.

3.2. Regresión Logística Regularizada y Máquinas de Vectores Soporte

Para esta aplicación, elegí entrenar Modelos de Regresión Logística Regularizados (Lasso/Ridge) y Máquinas de Vectores Soporte (o en este caso, también Clasificadores de Vectores Soporte). Estos modelos son relativamente viejos, pero aún se utilizan como referencia por su interpretabilidad y bajo costo computacional.

Para ambos tipos de modelos, comienzo separando los datos en un conjunto de entrenamiento y un conjunto de prueba y uso el primero para encontrar los mejores hiperparámetros utilizando la función `gridsearch` de *sklearn*. En el caso de las regresiones regularizadas, también elijo mediante validación cruzada entre modelos *Lasso* y *Ridge*. Utilizo el *roc auc* como métrica de decisión. Al final del ejercicio, probaré el rendimiento del modelo en la muestra de prueba. Los resultados se presentan en la Sección 5 de este documento.

4. Análisis de sentimientos con ChatGPT

En comparación con los modelos anteriores, los LLM representan un cambio radical. Primero, aunque se debe programar la aplicación para comunicarse con la API de OpenAI, las instrucciones reales para extraer los datos del texto se realizan a través de "mensajes" o "indicaciones" (en ingés, *prompt*) a este asistente virtual. Segundo, la interacción con la aplicación es opaca: no hay una razón clara por la que el chatbot responde de una manera u otra y descubrir el mejor mensaje para extraer los datos

generalmente se logra mediante ensayo y error. Y tercero, como este programa es relativamente nuevo, no hay una "mejor manera" con el chatbot. Esto dificulta encontrar el mensaje óptimo.

Lo que recojo aquí es el producto de la experiencia que obtuve trabajando con Andrés Gago para su artículo (Abad et al., 2023). En las siguientes subsecciones mostraré las indicaciones, algunas reglas generales y la estrategia de prueba general para perfeccionar la comunicación con el *chatbot*. Esto de ninguna manera pretende ser un trabajo exhaustivo sobre el tema.

En este caso comencé con una muestra pequeña de 115 reseñas seleccionadas del conjunto de datos. Con estas, iteré ligeramente sobre las mejores indicaciones usando métricas que mostraré en las siguientes líneas. Una vez determinada la indicación final, ejecuté el código en toda la base de datos. Todo el trabajo con ChatGPT se realizó utilizando el valor de `temperature` más bajo posible, que controla la "creatividad" del modelo. Esto se hace para evitar alucinaciones y resultados extraños.

4.1. *The prompt*

La comunicación con el asistente virtual se realiza mediante dos tipos de mensajes. Primero tenemos un mensaje del sistema cuyo propósito es dar al modelo una visión general de su papel y tareas y, segundo, está el mensaje del *usuario*, con la solicitud real. Seguí la recomendación de esta guía de *prompt-engeneering* para modelos `gpt-3.5` de escribir toda la instrucción en el mensaje del usuario, en lugar de en la parte del `sistema`. Como se puede ver en el Apéndice Figura 9, este mensaje consiste en dos partes: primero, una descripción general de la tarea y el formato en el que necesito que responda (a través de un diccionario de Python). Luego, proporciono al chatbot la reseña y las dos preguntas. Aunque ChatGPT puede entender diferentes idiomas en la misma indicación, todo esto se hace en portugués para minimizar el ruido en el problema. Las preguntas, traducidas al español, son:

1. ¿De "1" (muy malo) a "5" (excelente), qué piensa que fue la evaluación numérica del usuario que dejó este comentario? Restrinja su respuesta a un número de "1" a "5".

2. ¿Qué opinión cree que tiene el usuario sobre este producto? Si es positiva, responda con "1". Si es negativa, responda con "0".

Estas preguntas se hacen para verificar la consistencia interna del modelo. La Tabla 5 muestra el resultado del cruce de las respuestas de ambas preguntas. Podemos ver que casi todas las respuestas pasan esta prueba: para aquellas entradas que ChatGPT considera positivas, asigna altas puntuaciones.[2]

[2]Los errores de API son comunes con esta aplicación y en gran medida inevitables. Dado que en mi configuración hago ambas preguntas simultáneamente, los errores de API aparecen en ambos casos a la vez.

	Predicted Score:	1	2	3	4	5	API Error
Predicted polarity	0	19	4	16	0	1	0
	1	0	0	2	3	59	0
	API Error	0	0	0	0	0	10

Cuadro 5: Tabla pivote que muestra la consistencia interna de los resultados de ChatGPT en la submuestra de *prompt engineering*. Las entradas cuentan el número de instancias en las que una revisión dada fue predicha para tener una cierta puntuación numérica (columnas) y una polaridad dada (filas).

Los resultados no son perfectos. Considere el caso de una reseña textual cuyo texto es sólo "10" (presumiblemente, diez puntos sobre diez). La respuesta de ChatGPT a nuestra consulta fue repetir "10", aunque no esté dentro de las opciones provistas en la indicación. En este caso imputé la respuesta "5", como una interpretación válida de esta respuesta. Al preguntar sobre la polaridad de la reseña, el modelo respondió negativo.

Antes de pasar a los resultados de los tres modelos, deben hacerse dos advertencias. Primero, como *OpenAI* no ha divulgado exactamente lo que usaron para entrenar el modelo, podría ser el caso de que este conjunto de datos estuviera incluido en los datos de entrenamiento del modelo. Aun así, la indicación fue construida específicamente para este ejercicio. En segundo lugar, el costo total de la ejecución del modelo completo fue de alrededor de 40 dólares, mientras que tanto los modelos LASSO/RIDGE como el SPV fueron gratuitos. Además, como OpenAI tiene tarifas y regulaciones específicas en torno al uso del modelo, el tiempo de ejecución del modelo ChatGPT fue de más de dos días completos, mientras que los otros modelos se ejecutan en menos de un segundo. Aunque puede requerirse alguna optimización para mi código, el principal problema fue esperar tanto las respuestas de ChatGPT como evitar sus límites de solicitudes por minuto.

5. Resultados

En esta sección, presento los resultados de ambos tipos de modelos: *Machine Learning* y ChatGPT. Comienzo discutiendo brevemente los resultados en la submuestra de entrenamiento tanto para los modelos Lasso/Ridge como para SVM y elijo los que mejor se ajustan. Luego muestro que, aunque ligeramente mejor para la Regresión Logística con Ridge, los resultados en la submuestra de prueba son bastante similares a los de SVM. Continúo con los modelos de ChatGPT, mostrando que los resultados son similares a los modelos anteriores. Aunque se podría interpretar que el uso de LLM no implica un beneficio para los modelos más tradicionales (o incluso, que es peor, dado que es más costoso y lento) es importante remarcar que en la mayor parte de las aplicaciones no se tienen datos clasificados. Tener un resultado a la par con modelos tradicionales sin tener que hacer un entrenamiento previo puede ser un gran beneficio.

5.1. Ridge y Lasso

Como se explicó en secciones anteriores, utilizo la función `gridsearch` de `sklearn` en la submuestra de entrenamiento para elegir tanto el parámetro de regularización c como entre Ridge y Lasso usando la palabra clave `penalty`. El mejor modelo tiene un *roc auc* de alrededor de 0.95, con un parámetro de regularización $c = 1.638$ y una penalización de Ridge. En el gráfico izquierdo en Figura 5 se puede ver que los Modelos de Ridge superan a Lasso para todo el espacio de búsqueda. Luego pruebo el rendi-

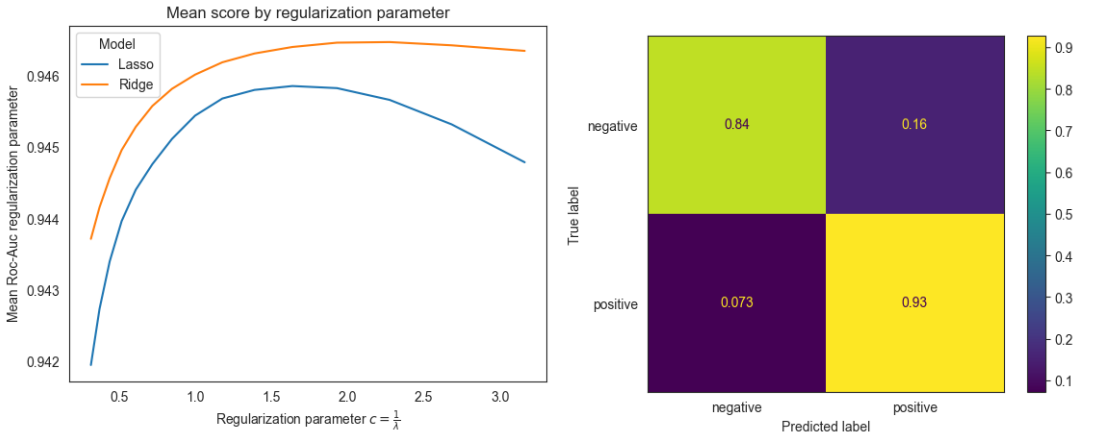

Figura 5: La figura izquierda muestra la puntuación media por especificación en la submuestra de entrenamiento para ambos modelos. Dado que Ridge supera a Lasso en todo el espacio de búsqueda, lo utilizo como regularización preferida. En la derecha, muestro la matriz de confusión para el mejor modelo de Ridge en la submuestra de prueba. Los valores se refieren a la proporción de predicciones para cada categoría verdadera (filas).

miento de la mejor especificación en la submuestra de prueba. Los resultados están en Tabla 6, donde podemos ver una precisión de alrededor de 0.90 en ambas clasificaciones. En la imagen de la derecha de Figura 5 podemos ver la matriz de confusión para este modelo, normalizada como un porcentaje de la verdadera clasificación. Con esto, podemos ver que el modelo predice correctamente los mensajes positivos con más frecuencia que los negativos. Este será un patrón en todos los modelos. Una muestra de los resultados de la predicción se puede encontrar en Tabla 11 en el Apéndice.

5.2. Máquinas de Vectores de Soporte

Repito el mismo procedimiento para el clasificador de Máquinas de Vectores de Soporte. Usando `gridsearch` obtenemos el mejor parámetro de regularización, $c = 0.316$ con un *roc auc* de alrededor de 0.945. En Figura 6 podemos ver que el resultado del modelo es prácticamente idéntico al del ejemplo de Ridge, aunque con una eficiencia ligeramente menor en la clase positiva. Podemos corroborar este resultado en Tabla 6, donde el SVM tiene una precisión general de 0.89, casi idéntica a la del modelo de Ridge. Ambos resultados parecen indicar que las técnicas de aprendizaje automático tradicionales son formas simples y confiables de obtener la polaridad de oraciones simples como reseñas. Es importante recordar que, aunque ambos modelos necesitan

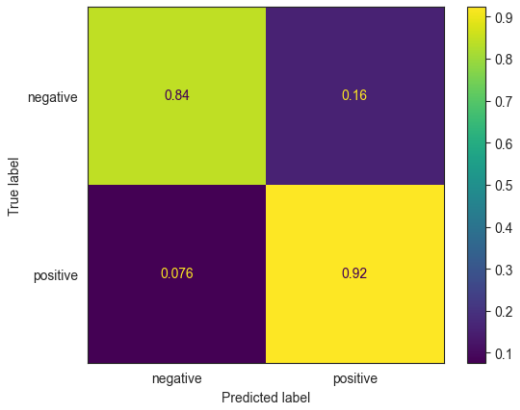

Figura 6: Matriz de Confusión para el mejor modelo SVM en la submuestra de prueba. Los valores están normalizados como un porcentaje de los valores verdaderos (filas) de cada categoría.

que los datos se limpien y normalicen, una vez que este procedimiento se resuelve, son rápidos, confiables y gratuitos de usar, en contraste con ChatGPT.

	Ridge				SVM				ChatGPT3.5			
	Precision	Recall	F1-Score	Support	Precision	Recall	F1-Score	Support	Precision	Recall	F1-Score	Support
Negative	0.86	0.85	0.85	3614	0.86	0.83	0.84	3669	0.85	0.90	0.87	15027
Positive	0.92	0.93	0.92	7063	0.91	0.93	0.92	7008	0.95	0.92	0.93	28610
Accuracy	-	-	0.90	10677	-	-	0.89	10677	-	-	0.91	43637
Macro avg	0.89	0.89	0.89	10677	0.89	0.88	0.88	10677	0.90	0.91	0.90	43637
Weighted avg	0.90	0.90	0.90	10677	0.89	0.89	0.89	10677	0.91	0.91	0.91	43637

Cuadro 6: Comparación de los tres modelos en la clasificación binaria.

5.3. ChatGPT

Para los resultados de ChatGPT, primero comienzo comprobando la consistencia interna, como se hizo con la submuestra de ingeniería de indicaciones. En Tabla 7, presento una tabla pivote comparando el resultado del modelo para ambas preguntas, Q1 (pidiendo predecir la puntuación de la reseña colocada por el consumidor) y Q2 (simplemente indicando la polaridad). Afortunadamente, vemos que ChatGPT es internamente consistente en la muestra completa, clasificando como positivos los que predicen con alta puntuación y viceversa. Tenga en cuenta que la mayoría de las reseñas de cinco estrellas/negativas se ingresaron, donde la respuesta real del chatbot fue "10", repitiendo textualmente la prueba de la reseña.

	Predicted Score:	1	2	3	4	5
	0	7716	4704	2683	101	31
Predicted polarity:	1	0	9	964	2989	23061

Cuadro 7: Tabla pivote que muestra la consistencia interna de los resultados de ChatGPT en la muestra completa. Las entradas cuentan el número de instancias en las que una determinada reseña fue predicha tanto para tener una puntuación numérica específica (columnas) como una determinada polaridad (filas). Dado que los errores de la API son aleatorios y no afectan el resultado general, los excluyo.

Un resultado común que puede surgir al trabajar con ChatGPT es que las respuestas no permanecen consistentes cuando se les pregunta en diferentes iteraciones. Este fue

un gran problema que encontré mientras trabajaba para Andrés Gago en su artículo (Abad et al., 2023), así que lo verifico aquí, volviendo a probar una pequeña muestra de 200 reseñas con las mismas preguntas. Afortunadamente, en Tabla 8 muestro que el porcentaje de respuestas que permanecen iguales es alto: justo por debajo del 90 % para la primera pregunta y casi todas para la segunda.

Question	Match Count	Percentage
Q1	175	89.29
Q2	193	98.47

Cuadro 8: Esta tabla muestra en una submuestra de 200 reseñas el recuento de coincidencias y el porcentaje de respuestas repetidas cuando se hacen dos veces con la misma entrada.

Ahora compruebo las respuestas del modelo contra los valores reales de las reseñas de puntuación. En Tabla 9 y Figura 7 se puede ver que el modelo predice en su mayoría correctamente las reseñas de puntuación, con una precisión general del 65 %. Si bien este número puede parecer bajo en comparación, este ejercicio es mucho más difícil que el binario y hay que recordar que no entrené ni calibré ChatGPT de ninguna manera aparte del *prompt engeneering*.

	precision	recall	f1-score	support
1.0	0.755547	0.660648	0.704918	8917.000000
2.0	0.190774	0.423244	0.263002	2306.000000
3.0	0.253364	0.257361	0.255347	3804.000000
4.0	0.349718	0.172590	0.231119	6472.000000
5.0	0.820460	0.877089	0.847830	22138.000000
accuracy	0.650366	0.650366	0.650366	0.650366
macro avg	0.473973	0.478186	0.460443	43637.000000
weighted avg	0.654666	0.650366	0.644605	43637.000000

Cuadro 9: Estadísticas para la predicción del número de estrellas por parte de ChatGPT. Podemos ver que el modelo evalúa correctamente alrededor del 65 % de las entradas.

Antes de pasar al caso binario, vale la pena señalar un punto ciego aparente del bot de chat en esta aplicación: las reseñas de cuatro estrellas. Tenga en cuenta que, en Figura 7, más de la mitad de las verdaderas reseñas de cuatro estrellas se clasifican incorrectamente como de cinco estrellas. Una posible explicación para esto es que puede que no sea culpa de ChatGPT, sino un reflejo de que las reseñas de puntuación pueden no referirse exactamente a su contraparte escrita. Podría ser que los clientes no sean consistentes con sus reseñas y un humano también las clasificaría incorrectamente.

Sin embargo, este problema no solo aparece en el lado superior de la distribución, sino también para las puntuaciones bajas. En Figura 8 muestro la distribución de respuestas que ChatGPT predice vis á vis la distribución real[3]. Aquí se puede ver el mismo hecho que en el gráfico derecho de Figura 7: el asistente virtual tiene más facilidad

[3]Tenga en cuenta que los porcentajes pueden diferir ligeramente de este gráfico a los anteriores debido a que aquí eliminé los errores de la API. Estos son en su mayoría impredecibles y aleatorios, por lo que el resultado general sigue siendo válido

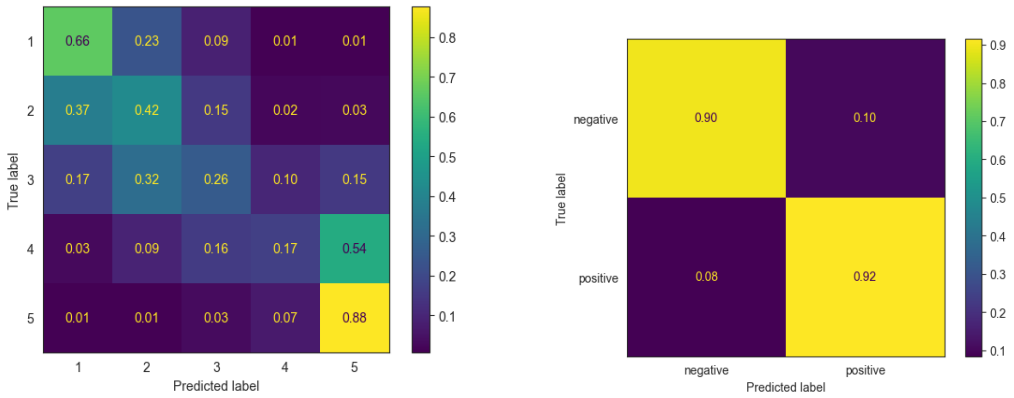

Figura 7: Matriz de confusión para la clasificación binaria (Q1, a la derecha) y para la predicción de puntuación (Q2, a la izquierda).

para clasificar valores extremos que intermedios, por lo que mientras que las cuatro estrellas están subrepresentadas, las dos estrellas es lo contrario en sus respuestas.

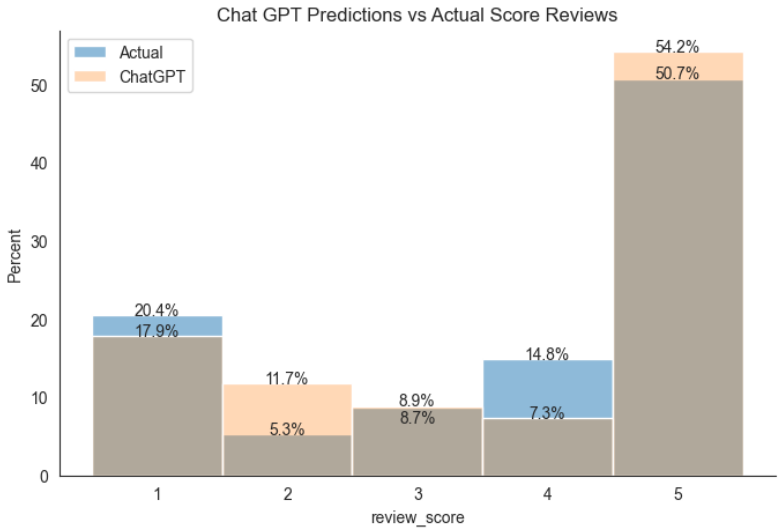

Figura 8: Distribución de respuestas proporcionadas por ChatGPT en las cinco categorías frente a la distribución real de respuestas.

Finalmente, ampliaré sobre los resultados respecto a la pregunta Q2, la evaluación de polaridad. En la Tabla 6 muestro que la precisión de ChatGPT en la clasificación binaria es ligeramente mejor que la de los modelos tradicionales, en 0.91. También se puede ver (o usando Figura 7 junto con Figura 6 y Figura 5) que el aumento en la eficiencia del modelo se debe en gran medida a su clasificación correcta de sentimientos negativos.

Esto puede resultar decepcionante, dado que el uso de este modelo fue costosa, lenta y no de código abierto, en comparación con las otros modelos. Sin embargo, es importante recordar que no realicé ningún entrenamiento en el modelo de ChatGPT y que está clasificando en gran medida correctamente el sentimiento en portugués, con sus desventajas respecto de otros idiomas más populares. Esto implica que la utilidad de ChatGPT es más importante en aquellos casos donde no hay clasificaciones disponibles para entrenar otros modelos.

6. Conclusiones

En este trabajo, se llevó a cabo un análisis completo de las reseñas de texto en un contexto de *e-commerce*. Después de algunas pruebas y exploración general de los datos, expliqué el proceso de limpieza de datos realizado para utilizar el texto en un entorno de aprendizaje automático tradicional. Después de reescalar los datos utilizando *tf-idf*, se utilizaron tres modelos lineales (Lasso, Ridge y SVC) para entrenar un clasificador de *análisis de sentimientos*. Estos resultados fueron satisfactorios, con una precisión general de alrededor de 0.90. Posteriormente, expliqué los pasos necesarios para utilizar un LLM popular, ChatGPT de OpenAI, como una herramienta de Análisis de Sentimientos. Discutí el proceso de *prompt engineering* y mostré los resultados en una pequeña submuestra de datos antes de continuar con el resto de la base de datos. A partir de aquí, comparé los tres modelos y encontré prácticamente ninguna diferencia en sus resultados. Si bien es cierto que ChatGPT tuvo un rendimiento ligeramente mejor en este entorno, es discutible si la mejora en el rendimiento justifica el costo, el tiempo de espera y la opacidad relacionados con el uso de este modelo propietario o no. Sin embargo, es importante recordar que ChatGPT logró un rendimiento relativamente bueno sin un conjunto de datos de entrenamiento (en un idioma no tan popular) y que no es obvio cómo las predicciones entrenadas de Ridge o SVM se desempeñarían con datos de texto en portugués no relacionados. Además, es importante remarcar que los *LLMs* han mejorado sustancialmente desde que se obtuvieron los resultados de este modelo. Estos tipos de modelos no desplazan totalmente los más tradicionales, pero resultan de gran utilidad y de muy buen rendimiento cuando no se tiene a disposición los datos para entrenar modelos alternativos.

Bibliografía

Abad, Jose, Vicente J. Bermejo, Felipe Carozzi, and Andres Gago (2023) "Government Turnover and External Financial Assistance," 10.2139/ssrn.4520859.

Bird, Steven, Ewan Klein, and Edward Loper (2009) *Natural Language Processing with Python*: O'Reilly.

Cutler, David M., James M. Poterba, and Lawrence H. Summers (1989) "What Moves Stock Prices?" *The Journal of Portfolio Management*, 15 (3), 4–12, 10.3905/jpm.1989.409212.

Davis, Donald R., Jonathan I. Dingel, Joan Monras, and Eduardo Morales (2019) "How Segregated Is Urban Consumption?" *Journal of Political Economy*, 127 (4), 1684–1738, 10.1086/701680.

Dellarocas, Chrysanthos (2003) "The Digitization of Word of Mouth: Promise and Challenges of Online Feedback Mechanisms," *Management Science*, 49 (10), 1407–1424, http://www.jstor.org/stable/4134013.

Dybowski, T.P. and P. Adämmer (2018) "The Economic Effects of U.S. Presidential Tax Communication: Evidence from a Correlated Topic Model," *European Journal of Political Economy*, 55, 511–525, 10.1016/j.ejpoleco.2018.05.001.

Engelberg, Joseph and Christopher A. Parsons (2016) "Worrying about the Stock Market: Evidence from Hospital Admissions," *The Journal of Finance*, 71 (3), 1227–1250, 10.1111/jofi.12386.

Esposito, Elena, Tiziano Rotesi, Alessandro Saia, and Mathias Thoenig (2023) "Reconciliation Narratives: "The Birth of a Nation" after the US Civil War," *American Economic Review*, 113 (6), 1461–1504, 10.1257/aer.20210413.

Fatouros, Georgios, John Soldatos, Kalliopi Kouroumali, Georgios Makridis, and Dimosthenis Kyriazis (2023) "Transforming Sentiment Analysis in the Financial Domain with ChatGPT," 10.48550/ARXIV.2308.07935.

Fraccaroli, Nicolò, Alessandro Giovannini, Jean-François Jamet, and Eric Persson (2022) "Ideology and Monetary Policy. The Role of Political Parties' Stances in the European Central Bank's Parliamentary Hearings," *European Journal of Political Economy*, 74, 102207, 10.1016/j.ejpoleco.2022.102207.

Fryer, Roland G. (2019) "An Empirical Analysis of Racial Differences in Police Use of Force," *Journal of Political Economy*, 127 (3), 1210–1261, 10.1086/701423.

Hanna, Alan J., John D. Turner, and Clive B. Walker (2020) "News Media and Investor Sentiment during Bull and Bear Markets," *The European Journal of Finance*, 26 (14), 1377–1395, 10.1080/1351847X.2020.1743734.

Jabbar, Jahanzeb, Iqra Urooj, Wu JunSheng, and Naqash Azeem (2019) "Real-Time Sentiment Analysis On E-Commerce Application," in *2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC)*, 391–396, Banff, AB, Canada: IEEE, May, 10.1109/ICNSC.2019.8743331.

Joachims, Thorsten (1998) "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," in Carbonell, Jaime G., Jörg Siekmann, G. Goos, J. Hartmanis, J. van Leeuwen, Claire Nédellec, and Céline Rouveirol eds. *Machine Learning: ECML-98*, 1398, 137–142, Berlin, Heidelberg: Springer Berlin Heidelberg, 10.1007/BFb0026683.

Koch, Alexander, Toan Luu Duc Huynh, and Mei Wang (2022) "News Sentiment and International Equity Markets during BREXIT Period: A Textual and Connectedness Analysis," *International Journal of Finance & Economics*, ijfe.2635, 10.1002/ijfe.2635.

Liang, Weixin, Yuhui Zhang, Hancheng Cao et al. (2023) "Can Large Language Models Provide Useful Feedback on Research Papers? A Large-Scale Empirical Analysis," 10.48550/ARXIV.2310.01783.

Lin, Xiaoxin (2020) "Sentiment Analysis of E-commerce Customer Reviews Based on Natural Language Processing," in *Proceedings of the 2020 2nd International Conference on Big Data and Artificial Intelligence*, 32–36, Johannesburg South Africa: ACM, April, 10.1145/3436286.3436293.

Lopez-Lira, Alejandro and Yuehua Tang (2023) "Can ChatGPT Forecast Stock Price Movements? Return Predictability and Large Language Models," July.

Loughran, Tim and Bill Mcdonald (2011) "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10Ks," *The Journal of Finance*, 66 (1), 35–65, 10.1111/j.1540-6261.2010.01625.x.

Obaid, Khaled and Kuntara Pukthuanthong (2022) "A Picture Is Worth a Thousand Words: Measuring Investor Sentiment by Combining Machine Learning and Photos from News," *Journal of Financial Economics*, 144 (1), 273–297, 10.1016/j.jfineco.2021.06.002.

Olist and André Sionek (2018) "Brazilian E-Commerce Public Dataset by Olist," 10.34740/KAGGLE/DSV/195341.

Pereira, Denilson Alves (2021) "A Survey of Sentiment Analysis in the Portuguese Language," *Artificial Intelligence Review*, 54 (2), 1087–1115, 10.1007/s10462-020-09870-1.

Romer, Christina D. and David H. Romer (1989) "Does Monetary Policy Matter? A New Test in the Spirit of Friedman and Schwartz," *NBER Macroeconomics Annual*, 4, 121–170, 10.1086/654103.

——— (2023) "Presidential Address: Does Monetary Policy Matter? The Narrative Approach after 35 Years," *American Economic Review*, 113 (6), 1395–1423, 10.1257/aer.113.6.1395.

Sangani, Raj (2021) "Regex Essential for NLP," https://towardsdatascience.com/regex-essential-for-nlp-ee0336ef988d, July.

Wang, Zengzhi, Qiming Xie, Zixiang Ding, Yi Feng, and Rui Xia (2023) "Is ChatGPT a Good Sentiment Analyzer? A Preliminary Study," 10.48550/ARXIV.2304.04339.

Zhang, Wenxuan, Yue Deng, Bing Liu, Sinno Jialin Pan, and Lidong Bing (2023) "Sentiment Analysis in the Era of Large Language Models: A Reality Check," 10.48550/ARXIV.2305.15005.

Zheng, Alice and Amanda Casari (2018) *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*: O'Reilly Media, 1st edition.

Zhu, Feng and Xiaoquan (Michael) Zhang (2010) "Impact of Online Consumer Reviews on Sales: The Moderating Role of Product and Consumer Characteristics," *Journal of Marketing*, 74 (2), 133–148, 10.1509/jm.74.2.133.

Appendix

ChatGPT Prompts

````
system: Você é um assistente de pesquisa.

user: Seu trabalho é ler comentários que os usuários escreveram em um site de
vendas online e responder ao seu conteúdo.
Dê sua resposta em um dicionário Python estruturado da seguinte forma:
```"Q1": "resposta à pergunta 1", "Q2": "resposta à pergunta 2", ...```
Por favor, não adicione nenhuma explicação adicional à sua resposta.
Não deixe nenhuma pergunta sem resposta.

<review>
Here the review text is inserted
</review>

- Q1: De 1 (muito ruim) a 5 (excelente), o que você acha que foi a avaliação
numérica do usuário que deixou este comentário?
Restrinja sua resposta a um número de 1 a 5
- Q2: Para você, como é a avaliação do usuário sobre esse produto?
Se você acha que a avaliação é positiva, responda com "1"
Se você acha que a avaliação é negativa, responda com "0"
````

Figura 9: Esqueleto general del *prompt*. El texto de la reseña se inserta dentro de los marcadores de código.
````

Muestra de casos del modelo

	order_id	review_score	review_text	positive
3	(...)	5	Recebi bem antes do prazo estipulado.	1
4	(...)	5	Parabéns lojas lannister adorei comprar pela ...	1
9	(...)	4	recomendo aparelho eficiente. no site a marca ...	1
12	(...)	4	Mas um pouco ,travando...pelo valor ta Boa.\r\n	1
15	(...)	5	Super recomendo Vendedor confiável, produto ok...	1

Cuadro 10: Una muestra de reseñas antes del proceso de limpieza.

	Negative	Positive
Recebi bem antes do prazo estipulado	0.0204224	0.979578
o pior vendedor do mundo. atenção ruim	0.838143	0.161857
Cada vez que compro mais fico satisfeita parabéns pela honestidade com seus clientes 👏👏👏👏👏👏	0.15943	0.84057
Super recomendo. Vendedor confiável, produto ok e entrega antes do prazo.	0.00234795	0.997652
GOSTARIA DE SABER O QUE HOUVE, SEMPRE RECEBI E ESSA COMPRA AGORA ME DECPCIONOU	0.686065	0.313935
Não chegou meu produto. Péssimo	0.999003	0.000997164

Cuadro 11: Resultados en una pequeña muestra del modelo Ridge.

Texto	Clasificación
Recebi bem antes do prazo estipulado	Positivo
o pior vendedor do mundo. atenção ruim	Negativo
Cada vez que compro mais fico satisfeita parabéns pela honestidade com seus clientes 👏👏👏👏👏👏	Positivo
Super recomendo. Vendedor confiável, produto ok e entrega antes do prazo.	Positivo
GOSTARIA DE SABER O QUE HOUVE, SEMPRE RECEBI E ESSA COMPRA AGORA ME DECPCIONOU	Negativo
Não chegou meu produto. Péssimo	Negativo

Cuadro 12: Resultados en una pequeña muestra del modelo SVM.