Tecnológico de Monterrey

Análisis de grandes volúmenes de datos

TC4034.10

Proyecto:
Entrega Final

Equipo 37:

Alejandro Díaz Villagómez	A01276769
Alonso Pedrero Martínez	A01769076
César Iván Pedrero Martínez	A01366501
Emiliano Saucedo Arriola	A01659258

Docentes:
Dr. Iván Olmos Pineda
Mtra. Verónica Sandra Guzmán de Valle
Mtro. Alberto Daniel Salinas Montemayor

Fecha de entrega:
22 de junio del 2025

Análisis de Sentimientos en Reseñas de Amazon mediante Técnicas de Big Data con PySpark

Resumen

El presente trabajo tiene como finalidad el desarrollo de una solución basada en técnicas de Big Data para el análisis de reseñas de productos publicadas en la plataforma Amazon. Aprovechando las capacidades de PySpark, se llevó a cabo el procesamiento y análisis de un conjunto de datos masivo que incluye millones de opiniones de usuarios, enfocándose en la identificación del sentimiento general expresado (positivo o negativo), la detección de patrones de comportamiento del consumidor y la extracción de información valiosa que pueda ser utilizada para la toma de decisiones estratégicas en entornos comerciales.

La motivación principal de este proyecto radica en la creciente necesidad de comprender de manera automatizada las percepciones de los consumidores en plataformas digitales, tanto por parte de fabricantes como de proveedores de servicios. A través de una serie de etapas metodológicas bien definidas (que incluyen la caracterización de la población, el diseño de una muestra representativa, el entrenamiento de modelos de aprendizaje automático y la evaluación mediante métricas apropiadas) se demuestra la viabilidad y efectividad de la propuesta planteada.

Los resultados obtenidos permiten confirmar que, mediante un análisis eficiente y escalable de grandes volúmenes de datos, es posible no solo clasificar opiniones con altos niveles de precisión, sino también derivar implicaciones prácticas para la mejora de productos, la atención al cliente y el diseño de sistemas de recomendación. Finalmente, se discuten oportunidades de mejora y líneas futuras de trabajo orientadas a la integración de modelos más sofisticados y la expansión a otros dominios de análisis.

Introducción

En la actualidad, el volumen de información generada diariamente por los usuarios en plataformas digitales representa una fuente invaluable de conocimiento para empresas y organizaciones. Particularmente en el comercio electrónico, las reseñas de productos se han consolidado como uno de los insumos más ricos y representativos para comprender el comportamiento del consumidor, evaluar la percepción de los productos y detectar áreas de mejora en tiempo real.

Amazon, como uno de los principales referentes globales del e-commerce y proveedor líder de infraestructura tecnológica mediante su plataforma AWS, constituye un entorno privilegiado para el análisis de datos masivos. Su posicionamiento como marketplace líder no solo le otorga una base de clientes diversa y activa, sino que también ofrece una enorme cantidad de datos útiles para desarrollar modelos de análisis de sentimiento, extracción de conocimiento y detección de patrones.

Este proyecto nace del interés profesional por desarrollar soluciones de análisis automatizado en contextos reales de negocio. En nuestra experiencia como consultores de software, es frecuente que clientes con tiendas en línea busquen herramientas que les permitan interpretar rápidamente las opiniones de sus usuarios y anticiparse a posibles problemas o tendencias. De ahí la pertinencia de construir una solución que combine la potencia del procesamiento distribuido (a través de PySpark) con técnicas de aprendizaje automático para el análisis de sentimientos en texto.

Los objetivos generales del proyecto se estructuran de la siguiente manera:

- Desarrollar una solución escalable para el análisis de sentimiento en reseñas de productos.
- Aplicar técnicas de Big Data para procesar eficientemente un volumen masivo de datos textuales.
- Detectar patrones recurrentes y generar información procesable para apoyar la toma de decisiones estratégicas.

De manera complementaria, se identifican aplicaciones prácticas que incluyen:

- Mejora de productos basada en retroalimentación de usuarios.
- Identificación y resolución proactiva de problemas mediante atención al cliente.
- Optimización de sistemas de recomendación basados en sentimiento y calificaciones.

Para alcanzar estos objetivos, se trabajó con un conjunto de datos titulado Amazon Reviews, disponible en la plataforma Kaggle. Este dataset contiene más de 3 GB de información estructurada, centrada en productos electrónicos, e incluye atributos como el identificador del producto, la calificación otorgada, la opinión textual y una variable adicional de sentimiento binario. Esta estructura y volumen lo convierten en una fuente idónea para aplicar técnicas de procesamiento paralelo y aprendizaje automático.

Propuesta de solución

El desarrollo del presente proyecto plantea una solución integral y escalable para el análisis automatizado de sentimientos en reseñas de productos publicadas en Amazon, combinando técnicas de procesamiento distribuido con PySpark y modelos de aprendizaje supervisado. Esta propuesta busca responder eficazmente a los desafíos inherentes al tratamiento de datos masivos, garantizando tanto la representatividad estadística de la muestra como la solidez y eficiencia computacional del modelo construido.

La estrategia metodológica implementada se fundamenta en el análisis de un conjunto de datos extenso, cuyo objetivo central es la predicción del sentimiento (positivo o negativo) expresado en cada reseña. Para alcanzar este fin, se diseñó un flujo de trabajo sistemático que abarca desde la caracterización inicial de la población y la construcción rigurosa de una muestra representativa, hasta la selección, entrenamiento y validación de modelos predictivos.

El proceso completo se estructura en las siguientes etapas clave:

1) Caracterización de la población
2) Construcción de una muestra representativa (M) y particiones Mi
3) Selección de métricas e identificación de algoritmos de aprendizaje
4) Segmentación de datos y configuración del preprocesamiento
5) Ajuste de hiperparámetros mediante validación cruzada

Cada una de estas fases metodológicas será descrita en detalle en los apartados siguientes, destacando tanto las decisiones técnicas adoptadas como los fundamentos estadísticos y computacionales que sustentan su implementación.

1) Caracterización de la población

La caracterización de la población constituye una etapa crítica en todo proceso de análisis basado en muestreo. Su objetivo es identificar las propiedades generales de la población objetivo (*P*) mediante un conjunto de variables que capturen diferencias y similitudes significativas entre los registros, y que sirvan como base para diseñar una estrategia de particionamiento representativa, libre de sesgos y alineada con los objetivos analíticos del proyecto.

Para ello, se seleccionaron variables con alto poder explicativo, dominio discretizable, baja proporción de valores nulos y relevancia directa en el contexto del análisis de sentimientos. Esta selección permite generar particiones homogéneas y comparables, facilitar la interpretación de resultados y detectar patrones comportamentales relevantes.

En primera instancia, se descartaron aquellas variables que, por su naturaleza, no aportaban valor analítico en esta etapa o introducían alta cardinalidad innecesaria:

- Identificadores únicos como customer_id, review_id, product_id, product_parent, y product_title actúan como claves o agrupadores, pero no representan comportamientos colectivos.
- Variables constantes como marketplace y product_category no ofrecen variabilidad interna, al contener un único valor en todo el dataset.
- Campos textuales como review_headline y review_body, aunque esenciales para el análisis de sentimiento, presentan cardinalidad extremadamente alta y no son adecuados para la segmentación estadística inicial.

Por otro lado, las siguientes variables han sido identificadas como relevantes para caracterizar los patrones de comportamiento en las reseñas de Amazon:

Nombre	Dominio (posibles valores)	Comentarios adicionales
star_rating	{1, 2, 3, 4, 5}	Indicador directo de satisfacción; presenta sesgo positivo característico.

helpful_votes	Enteros ≥ 0	Mide el impacto social de una reseña; distribución long-tail significativa.
total_votes	Enteros ≥ 0	Complementa helpful_votes; permite calcular ratios de utilidad percibida.
sentiment	{0, 1}	Variable objetivo del proyecto; correlacionada con star_rating.
verified_purchase	{'Y', 'N'}	Refleja la autenticidad de la reseña; mayor credibilidad en valores 'Y'.
vine	{'Y', 'N'}	Distingue reseñas orgánicas de aquellas incentivadas (programa Vine de Amazon).
review_date	Fechas entre 1999 y 2015	Permite detectar estacionalidades y cambios de comportamiento a lo largo del tiempo.

Cada una de estas variables aporta una dimensión única al análisis, ya sea desde una perspectiva temporal, de credibilidad, de popularidad o de valoración explícita del producto.

Para el diseño de la muestra estratificada y la construcción de particiones *Mi*, se eligió un subconjunto de tres variables categóricas: **star_rating**, **verified_purchase** y **vine**. Esta decisión se fundamenta en los siguientes criterios:

- Relevancia analítica directa:
  - star_rating refleja la evaluación explícita del usuario, estrechamente vinculada con el sentimiento.
  - verified_purchase permite distinguir reseñas de compradores auténticos, clave para evitar sesgos.
  - vine identifica reseñas incentivadas, útiles para analizar diferencias en el tono o contenido.

- Viabilidad práctica:
  - Incluir todas las variables conduciría a una explosión combinatoria de particiones con escasa representatividad en algunas celdas. El uso de estas tres variables permite mantener interpretabilidad sin perder granularidad analítica.

Esta caracterización fue esencial para garantizar que los modelos entrenados fueran evaluados sobre segmentos de la población con comportamiento representativo y coherente, evitando conclusiones sesgadas o limitadas a subconjuntos no generalizables.

2) Construcción de una muestra representativa (M) y particiones Mi

Una vez caracterizada la población, se procedió a la construcción de una muestra representativa denominada *M*, con el propósito de reducir la escala del conjunto de datos original sin perder su riqueza estructural ni comprometer la validez estadística de los análisis posteriores. Esta etapa es especialmente relevante en entornos de Big Data, donde el volumen de información puede superar la capacidad operativa de los sistemas y dificultar tanto el entrenamiento de modelos como la evaluación exploratoria detallada.

Para la estrategia de muestreo empleada, se adoptó un enfoque de muestreo estratificado proporcional, basado en las combinaciones únicas de las variables categóricas seleccionadas previamente: **verified_purchase** y **vine**. Esta técnica consiste en dividir la población en estratos homogéneos definidos por estas combinaciones, y luego seleccionar una muestra de cada estrato en proporción a su representación en la población total.

El motivo por el cual se excluyó la variable **star_rating** de esta etapa fue metodológico: al haber sido identificada como altamente correlacionada con la variable objetivo **sentiment**, incluirla podría inducir un sesgo no deseado en la muestra. Preservar únicamente variables neutrales garantiza una segmentación más imparcial y evita filtraciones de la variable objetivo hacia el diseño de la muestra, reforzando la validez del experimento.

El tamaño de la muestra se fijó en aproximadamente el 10% del total de registros, lo que equivale a más de 690,000 observaciones. Esta proporción fue seleccionada considerando un equilibrio entre representatividad estadística y viabilidad computacional, asegurando la preservación de patrones relevantes sin incurrir en costos operativos excesivos.

A partir de los conteos originales en la población para cada combinación de estrato (verified_purchase, vine), se calcularon los tamaños proporcionales deseados en la muestra, redondeando a los enteros más cercanos. Luego, se extrajo de forma independiente cada subconjunto de registros utilizando una lógica condicional por estrato, aplicando muestreo aleatorio sin reemplazo cuando el número de registros disponibles excedía lo requerido, y conservando todos los registros en caso contrario. Este procedimiento permitió:

- Asegurar la preservación de las proporciones reales observadas en la población original.
- Garantizar la representación adecuada de estratos minoritarios, como las reseñas del programa Vine (≈0.5%).
- Evitar el sobreajuste o la subrepresentación de perfiles relevantes para el análisis de sentimiento.

El resultado fue una muestra *M* de 691,829 registros, cifra levemente superior al tamaño proyectado debido a redondeos y a la inclusión total de ciertos estratos con baja frecuencia. Esta desviación es metodológicamente aceptable, dado que se mantiene la fidelidad estructural de la muestra respecto a la población completa.

Finlamente, una vez construida la muestra, se definieron las particiones *Mi* como subconjuntos disjuntos de *M*, agrupados según las mismas combinaciones de variables utilizadas en la estratificación (verified_purchase, vine). Este enfoque permite:

- Facilitar el análisis segmentado por perfil de usuario.
- Evaluar el comportamiento diferencial de los modelos en distintos subgrupos.
- Implementar estrategias de validación más controladas y específicas.

Al verificar la distribución final de las particiones generadas, se constató que las proporciones de cada estrato se conservaron con alta precisión respecto a las proporciones originales, validando así la correcta ejecución del proceso de muestreo.

3) Selección de métricas e identificación de algoritmos de aprendizaje

La selección del modelo de aprendizaje y sus respectivas métricas de evaluación representa un paso esencial en el diseño de una solución robusta, especialmente cuando se trabaja con grandes volúmenes de datos textuales en entornos distribuidos. En este proyecto se implementaron dos enfoques complementarios: uno supervisado, orientado a la clasificación binaria del sentimiento, y otro no supervisado, enfocado en la identificación de agrupamientos latentes dentro de las reseñas.

a) Modelo supervisado: Random Forest Classifier

Para abordar la tarea de clasificación del sentimiento (positivo o negativo), se seleccionó el algoritmo **RandomForestClassifier** del módulo pyspark.ml.classification. Esta decisión se fundamentó en criterios tanto técnicos como metodológicos, dada la capacidad del modelo para balancear precisión, interpretabilidad y eficiencia computacional en contextos de Big Data.

Las razones principales que justifican su elección son:

- Adecuación a problemas de clasificación binaria, como el presente, donde sentiment $\in \{0, 1\}$.
- Compatibilidad con datos numéricos y variables categóricas codificadas, sin requerir transformaciones adicionales.
- Tolerancia a outliers y ruido, ideal para variables como helpful_votes o total_votes, que presentan distribuciones asimétricas.
- No requiere normalización, lo que reduce el esfuerzo de preprocesamiento.
- Capacidad de manejar desbalanceo de clases, gracias a su naturaleza basada en muestreo bootstrap.
- Interpretabilidad del modelo, mediante análisis de importancia de variables.
- Eficiencia en entornos distribuidos, al tener un costo computacional razonable cuando se entrena en PySpark.

Otras alternativas como LogisticRegression, GBTClassifier y MultilayerPerceptronClassifier fueron consideradas, pero descartadas por requerimientos adicionales (normalización, tuning exhaustivo de hiperparámetros o mayores recursos computacionales) que no ofrecían ventajas significativas en este contexto.

b) Modelo no supervisado: K-Means Clustering

En la fase exploratoria no supervisada se optó por utilizar el algoritmo **KMeans**, con el objetivo de descubrir patrones latentes en las reseñas a partir de sus características numéricas codificadas. Dicho modelo fue seleccionado por las siguientes razones:

- Alta escalabilidad, con complejidad computacional baja, ideal para conjuntos de millones de registros.
- Compatibilidad con PySpark, a través de una implementación optimizada que se integra con pipelines de transformación y evaluación.
- Simplicidad interpretativa, facilitando la lectura de los centroides generados y su correspondencia con posibles perfiles de usuario.

Se descartaron modelos como GaussianMixture, PowerIterationClustering y BisectingKMeans, debido a su mayor complejidad, requerimientos de configuración avanzada o falta de ventajas claras en un espacio de agrupamiento plano como el actual.

c) Métricas de evaluación

Dado que se trabaja con dos enfoques distintos (supervisado y no supervisado), se seleccionaron métricas adecuadas a la naturaleza de cada modelo:

Para el modelo supervisado (RandomForestClassifier):

- Se utilizaron accuracy, precision, recall y F1-score, todas en su versión ponderada. Estas métricas permiten una evaluación equilibrada del desempeño, especialmente en presencia de clases desbalanceadas.
- La matriz de confusión fue empleada para analizar visualmente los errores de clasificación (falsos positivos y negativos), proporcionando una visión más granular del rendimiento.

Para el modelo no supervisado (KMeans):

- Se utilizó el Silhouette Score, métrica estándar que mide la cohesión interna de los clústers en relación con su separación respecto a otros. Este indicador permite evaluar la calidad del agrupamiento sin necesidad de etiquetas externas.
- De forma complementaria, se llevó a cabo una inspección semántica de los centroides y su posible alineación con la variable sentiment, permitiendo valorar si los agrupamientos generados reflejan estructuras coherentes con los patrones observados previamente.

La selección de estas métricas se realizó con base en su solidez teórica, disponibilidad en PySpark y adecuación al volumen de datos manejado. Este diseño metodológico permite no solo evaluar el rendimiento del modelo desde múltiples perspectivas, sino también profundizar en el entendimiento de la estructura subyacente en las opiniones de los usuarios.

4) Segmentación de datos y configuración del preprocesamiento

Como paso previo al entrenamiento de modelos, se llevó a cabo la división de la muestra representativa *M* en subconjuntos de entrenamiento (*Tri*) y prueba (*Tsi*), utilizando una estrategia de segmentación estratificada. Esta estrategia se basó en las combinaciones únicas de las variables categóricas verified_purchase y vine, previamente definidas como clave para mantener la representatividad de los distintos perfiles dentro de la población.

Se definió una proporción de 70% para entrenamiento y 30% para prueba, práctica ampliamente aceptada en el ámbito del aprendizaje automático por ofrecer un balance adecuado entre aprendizaje del modelo y evaluación de su capacidad de generalización. El uso de una semilla aleatoria (seed=42) garantiza la reproducibilidad del proceso de partición, aspecto fundamental en entornos experimentales.

El procedimiento consistió en recorrer cada una de las combinaciones posibles de las variables de estratificación (particiones *Mi*) e implementar la función **randomSplit** de PySpark sobre cada subconjunto, de forma que se generaran divisiones controladas y mutuamente excluyentes. Este enfoque no solo asegura la proporción deseada en cada estrato, sino que también preserva el balance entre clases incluso en estratos de baja frecuencia, como las reseñas incentivadas (vine = 'Y').

En aras de validar la correcta ejecución del proceso de segmentación, se realizaron dos comprobaciones clave:

- No superposición entre *Tri* y *Tsi*, condición garantizada por el comportamiento interno de la función randomSplit, la cual produce subconjuntos disjuntos por diseño.
- Cobertura completa de la muestra original *M*, verificada al reconstruir la unión de todas las particiones ($Tri \cup Tsi$) y confirmar su equivalencia exacta con el tamaño de *M*. Esta verificación garantiza que no se han perdido ni duplicado registros durante la división.

Una vez definida la segmentación de los datos, se procedió a diseñar el pipeline de preprocesamiento que se utilizará durante la etapa de entrenamiento y evaluación de los modelos. Este pipeline contempla las transformaciones necesarias para preparar los datos estructurados y garantizar su compatibilidad con los algoritmos seleccionados.

Aunando a lo anterior, el pipeline fue diseñado siguiendo principios de eficiencia y escalabilidad, contemplando las siguientes etapas:

- Conversión de variables categóricas a índices numéricos:
    - Se empleó **StringIndexer** para transformar las variables verified_purchase, vine y sentiment en variables numéricas (verified_purchase_index, vine_index, label). Esta conversión es esencial ya que los modelos de Spark MLlib no operan directamente sobre cadenas de texto. Se utilizó handleInvalid="skip" para omitir de forma segura registros con valores no válidos.

- Verificación y tratamiento de valores nulos:
    - Se validó la ausencia de valores nulos en columnas clave del modelo (helpful_votes, total_votes, verified_purchase, vine, y sentiment). Esta verificación permitió garantizar la integridad del conjunto de datos, confirmando que no era necesario reestructurar el muestreo o modificar las divisiones previamente realizadas. Otras columnas, como identificadores o campos textuales, no fueron consideradas en esta etapa por no aportar valor directo al vector de características.

- Ensamblado de características:
    - Se utilizó un **VectorAssembler** para consolidar en un solo vector (raw_features) las variables numéricas y categóricas indexadas previamente (helpful_votes, total_votes, verified_purchase_index, vine_index). Esta etapa permite estandarizar la entrada esperada por los modelos de aprendizaje.

- Escalamiento de variables:
    - Finalmente, se aplicó un StandardScaler para normalizar las características. Esta transformación es particularmente importante para modelos como KMeans, que se basan en distancias euclidianas. Se configuró el escalado con varianza unitaria (withStd=True) y sin centrado en la media (withMean=False), estrategia común en entornos distribuidos donde la centración puede aumentar innecesariamente el costo computacional.

El diseño de este pipeline garantiza que todas las transformaciones puedan aplicarse de manera uniforme y reproducible tanto al conjunto de entrenamiento como al de prueba, asegurando integridad metodológica y eficiencia operativa en el procesamiento de grandes volúmenes de datos.

5) Ajuste de hiperparámetros mediante validación cruzada

Una vez definidos los conjuntos de entrenamiento y el pipeline de preprocesamiento, se procedió al ajuste de hiperparámetros utilizando técnicas de validación cruzada, con el objetivo de optimizar el desempeño de los modelos tanto supervisados como no supervisados bajo evaluación.

a) Modelo supervisado: Random Forest

Para la clasificación del sentimiento de las reseñas, se utilizó un RandomForestClassifier, modelo robusto y eficaz en tareas con variables tanto categóricas como numéricas. La estrategia de ajuste consistió en la aplicación de una validación cruzada estratificada con 3 folds, sobre el conjunto de entrenamiento completo. Se evaluó el desempeño de diferentes combinaciones de hiperparámetros, definidos mediante una cuadrícula de búsqueda (ParamGridBuilder) que incluía:

- numTrees: número de árboles en el bosque (valores evaluados: 10 y 50)
- maxDepth: profundidad máxima del árbol (valores evaluados: 5 y 10)
- minInstancesPerNode: mínimo de instancias por nodo hoja (valores evaluados: 1 y 5)

El pipeline de ajuste integró todas las etapas previas de transformación de datos, incluyendo la indexación de variables categóricas, ensamblado del vector de características, escalamiento de variables y codificación de la etiqueta. Como métrica de evaluación se empleó el F1-score, dada su capacidad para balancear precisión y exhaustividad, especialmente útil en contextos con posibles desbalances de clase.

Tras entrenar las diferentes configuraciones sobre un conjunto de más de 480,000 registros, el modelo con mejor desempeño fue el que empleó *50 árboles, una profundidad máxima de 10 y una instancia mínima por nodo de 1*. Esta configuración refleja una preferencia del modelo por una alta capacidad de particionamiento, lo cual es razonable en un escenario de alta dimensionalidad y gran volumen de datos. La validación cruzada, al asegurar la rotación de los subconjuntos de entrenamiento y prueba, permitió obtener estimaciones más confiables del desempeño general esperado.

b) Modelo no supervisado: K-Means

En paralelo, se exploró una técnica no supervisada mediante el algoritmo KMeans, con el objetivo de identificar estructuras latentes en los datos sin hacer uso de la variable objetivo. El análisis se centró en evaluar diferentes valores de k (número de clústers) dentro del rango [2, 5], aplicando el mismo pipeline de preprocesamiento que en el modelo supervisado, salvo por la omisión de la etapa de indexación de la etiqueta.

Para cada valor de k, se entrenó un modelo sobre el conjunto completo de entrenamiento y se evaluó su desempeño utilizando el Silhouette Score, métrica que mide la coherencia interna de los clústers generados. El mejor resultado se obtuvo para k = 2, con un Silhouette Score perfecto de 1.0000, indicando una separación clara y natural entre dos grupos en el espacio de características (lo cual es poco común y digno de exploración adicional). Aunque los valores para k = 4 y k = 5 también fueron elevados (0.9745), la simplicidad del modelo con k = 2 y su puntuación sobresaliente respaldan su elección como la configuración óptima.

Experimentación

Finalizadas las etapas de construcción de la muestra representativa, preprocesamiento, selección de algoritmos, ajuste de hiperparámetros y validación cruzada, esta sección presenta los resultados obtenidos a partir del proceso de experimentación. El objetivo principal es evaluar la capacidad predictiva de los modelos desarrollados y analizar su idoneidad para ser aplicados en contextos reales.

Se examinan tanto el desempeño del modelo supervisado (**RandomForestClassifier**), como posteriormente se integrará el análisis del modelo no supervisado (**KMeans**), lo cual permite comparar enfoques complementarios en tareas de análisis de sentimiento.

1) RandomForestClassifier

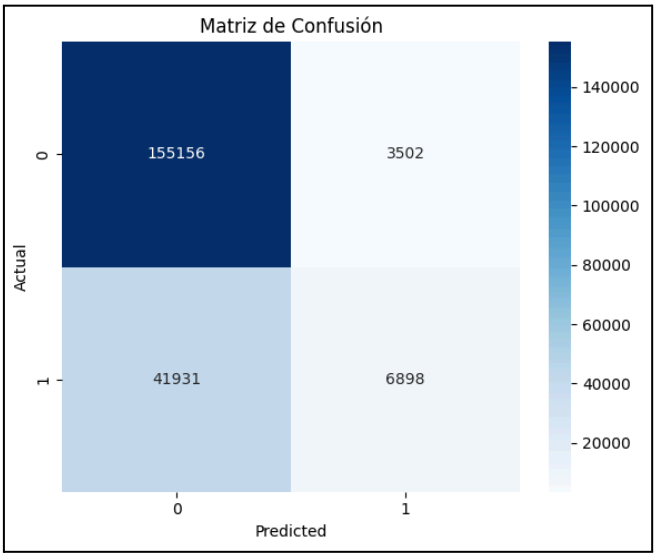
El modelo alcanzó un desempeño global con las siguientes métricas sobre el conjunto de prueba:

Accuracy	F1-score	Precision	Recall
0.78	0.72	0.76	0.78

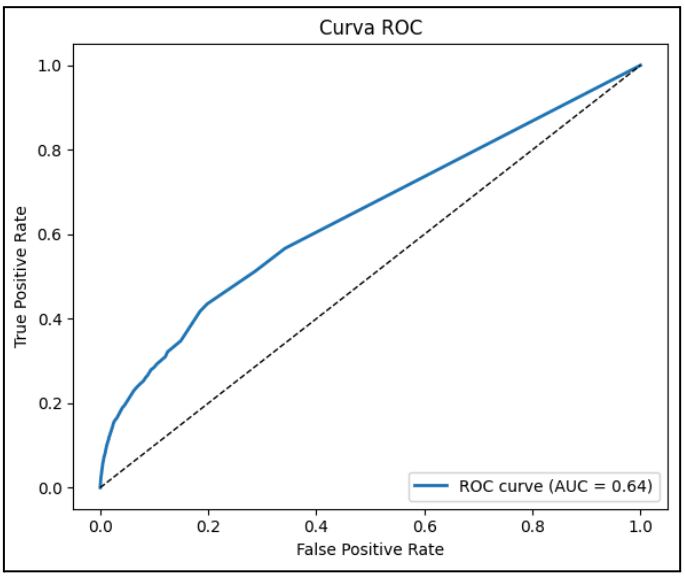
Estos valores reflejan una capacidad razonablemente alta del modelo para generalizar sobre datos no vistos, especialmente si se considera la presencia de cierto desbalance en las clases. La precisión elevada sugiere una baja proporción de falsos positivos, mientras que el recall indica una adecuada capacidad para capturar la mayoría de los ejemplos positivos reales. El F1-score ponderado permite balancear ambas métricas y demuestra un rendimiento sólido, sin que haya un compromiso excesivo entre sensibilidad y especificidad.

Con el fin de proporcionar una visión más completa del desempeño del modelo, se generaron las siguientes representaciones gráficas:

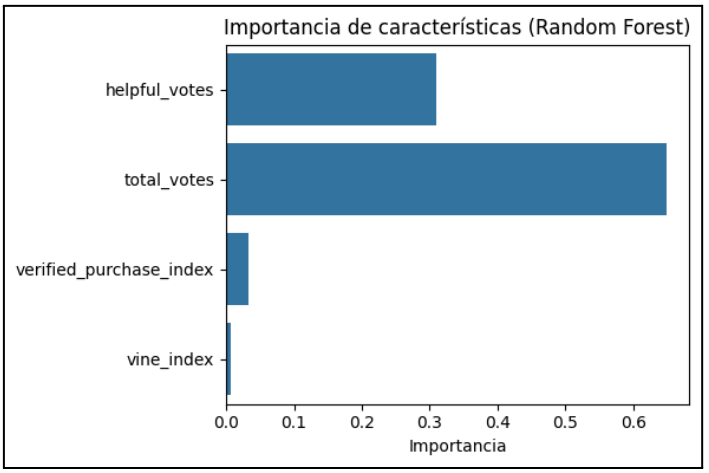
- **Matriz de confusión**: Esta gráfica permite visualizar de manera explícita las predicciones correctas e incorrectas del modelo. Se observa un alto número de verdaderos negativos (155,156) y verdaderos positivos (6,898), con una proporción moderada de falsos negativos (41,931) que indica una oportunidad de mejora en la detección de la clase minoritaria. Aun así, la relación entre verdaderos positivos y falsos positivos (3,502) resulta favorable y refuerza la robustez del modelo.



- **Curva ROC y AUC**: El área bajo la curva (AUC) fue de 0.64, lo cual representa una capacidad moderada de discriminación entre clases. Aunque este valor podría mejorarse, es aceptable dadas las características del dataset y la naturaleza ruidosa de algunas variables. La curva evidencia que el modelo logra una mejor tasa de verdaderos positivos que una predicción aleatoria, pero también sugiere posibles márgenes de optimización adicionales, por ejemplo mediante técnicas de balanceo de clases o ingeniería de características más avanzada.



- **Importancia de características**: El análisis de importancia de variables indica que total_votes y helpful_votes son los principales contribuyentes a la predicción del sentimiento, mientras que verified_purchase_index y vine_index tienen menor relevancia. Este hallazgo es coherente con el comportamiento observado en etapas exploratorias, y respalda la interpretación de que el volumen y utilidad percibida de los votos son factores determinantes en la valoración del sentimiento expresado.



En conjunto, tanto los indicadores numéricos como las visualizaciones confirman que el modelo supervisado implementado presenta un rendimiento competitivo y consistente. Su capacidad de generalización, su estabilidad frente a ruido y su eficiencia computacional lo convierten en una alternativa viable para ser desplegada en un entorno productivo real, especialmente si se considera su facilidad de interpretación y mantenimiento. No obstante, existen áreas de mejora asociadas a la clase minoritaria, lo que podría explorarse en futuras iteraciones mediante reentrenamiento con técnicas de balanceo sintético o ensambles más sofisticados.

2) KMeans

Dado que no se cuenta con valores verdaderos para evaluar el desempeño del modelo KMeans, se empleó el Silhouette Score como métrica principal de evaluación. Este indicador cuantifica la cohesión interna de los clústeres y su separación entre ellos, siendo un valor de referencia común en tareas de clustering. Inicialmente, al intentar aplicar el modelo sobre el conjunto de prueba (all_tsi_union), se observó que todos los registros fueron asignados a un único clúster, impidiendo el cálculo del Silhouette Score y sugiriendo una baja diversidad estructural en dicha partición.

Por esta razón, el modelo se evaluó sobre el conjunto de entrenamiento (all_tri_union), obteniéndose un Silhouette Score perfecto de 1.0000. Aunque este resultado parecería indicar una segmentación ideal, es importante analizar con cautela su significado. En contextos reales, un valor tan elevado suele ser poco frecuente y puede reflejar:

- Una separación extrema entre clústeres causada por valores atípicos o distribuciones desequilibradas.
- Efectos de variables mal normalizadas o con alta varianza.

- La detección de un clúster artificial generado por pocos registros con características muy distintas.

Al extraer los centroides generados por el modelo, se observó lo siguiente:

- **Centroide 0**: Presenta valores bajos y homogéneos en todas las variables, representando un grupo común de reseñas con baja interacción.
- **Centroide 1**: Exhibe valores anómalamente altos en las variables de votos (helpful_votes y total_votes), con ceros en las demás. Esto sugiere un grupo altamente atípico, posiblemente asociado a reseñas virales o con visibilidad inusualmente elevada.

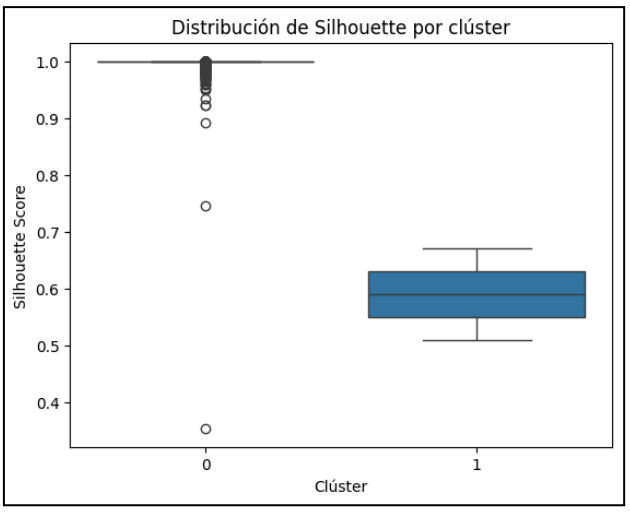
Esta marcada separación entre los centroides ayuda a explicar el valor perfecto del Silhouette Score, ya que la distancia entre ambos centros es considerablemente grande y la variabilidad interna dentro de cada grupo es mínima. Además, al analizar la distribución del sentimiento por clúster, se encontró lo siguiente:

Clúster	Sentimiento Negativo (0)	Sentimiento Positivo (1)	Total
0	114,225	370,115	484,340
1	0	2	2

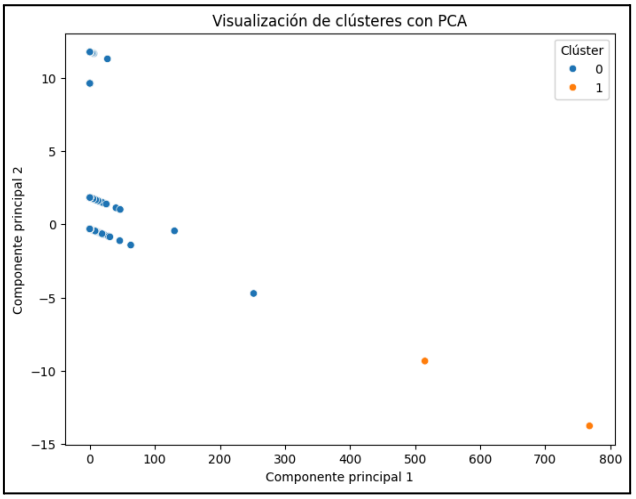
Este desbalance extremo refuerza la idea de que el clúster minoritario representa un caso excepcional más que una estructura generalizada del conjunto de datos. En consecuencia, aunque el agrupamiento es matemáticamente óptimo, no resulta práctico desde un punto de vista interpretativo o productivo.

Para reforzar el análisis del modelo no supervisado, se generaron diversas visualizaciones:

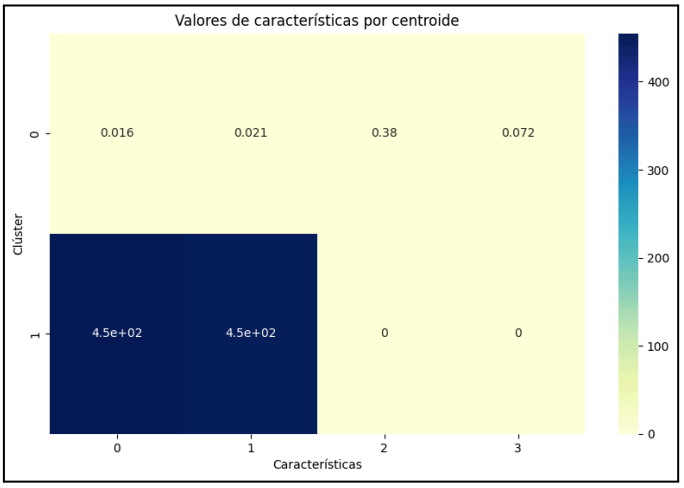
- **Gráfico de Silhouette Score por clúster**: A pesar del valor promedio perfecto, la distribución interna muestra que el clúster dominante contiene gran parte de la variabilidad.



- **Proyección PCA (Análisis de Componentes Principales)**: Se aprecia una separación clara entre los clústeres, aunque el clúster minoritario aparece aislado con apenas un par de observaciones.



- **Mapa de calor de los centroides**: Refleja la gran disparidad entre los clústeres en al menos dos variables, lo que sugiere que la separación se basa en pocas características dominantes.



Aunque el modelo KMeans logró una segmentación altamente separable en términos matemáticos, su utilidad práctica es cuestionable debido a:

- La presencia de un clúster extremadamente pequeño y poco representativo.
- La falta de diversidad en los clústeres para tareas como segmentación de clientes, análisis de comportamiento o detección de patrones generales.

En un entorno productivo, sería recomendable considerar:

- Una revisión de las variables utilizadas, posiblemente filtrando outliers.
- Aplicar técnicas de reducción de dimensionalidad o métodos de clustering más robustos (e.g., DBSCAN o Gaussian Mixture Models).

- Evaluar la estabilidad del modelo sobre distintas muestras para asegurar que la estructura no depende exclusivamente de pocos registros extremos.

Aunque el modelo de KMeans mostró una separación clara y cuantitativamente destacada, su aplicabilidad real es limitada bajo las condiciones actuales del conjunto de datos. Este hallazgo demuestra la importancia de complementar los indicadores estadísticos con un análisis cualitativo y visual del comportamiento del modelo.

Conclusiones y trabajo futuro

El desarrollo de este proyecto representó una oportunidad invaluable para comprender y aplicar conceptos clave del análisis de datos en un contexto de Big Data. A través del uso de herramientas como PySpark, se logró implementar modelos de aprendizaje supervisado y no supervisado sobre un conjunto de reseñas masivo, reforzando habilidades tanto técnicas como analíticas.

Uno de los principales aprendizajes fue reconocer la importancia de elegir las tecnologías adecuadas para enfrentar problemas reales donde el volumen de datos supera las capacidades tradicionales. Así mismo, se evidenció el papel crucial que juega el preprocesamiento de datos, el conocimiento del dominio y la construcción de una muestra representativa, no solo como base para obtener resultados sólidos y confiables, sino también como una estrategia necesaria para no comprometer la capacidad computacional disponible. Además, se destacó que la correcta visualización de resultados (mediante gráficas y tablas relevantes) permitió comunicar hallazgos de forma clara tanto a audiencias técnicas como no especializadas.

Desde el punto de vista práctico, se consolidó un modelo supervisado robusto capaz de predecir el sentimiento de los consumidores con un desempeño satisfactorio, especialmente útil para sistemas de recomendación o análisis de reputación. No obstante, se identificó la necesidad de mejorar el tratamiento de la clase minoritaria y explorar estrategias más sofisticadas de feature engineering. Por su parte, el modelo no supervisado demostró una separación técnica efectiva, pero su aplicabilidad aún es limitada. En trabajos futuros, se propone fortalecer esta línea mediante mejoras en la definición de clústeres y su alineación con patrones de comportamiento de compra.

Finalmente, una línea de desarrollo particularmente prometedora radica en la incorporación de técnicas de Procesamiento de Lenguaje Natural (NLP). Dado que el corpus original contiene textos ricos en información, su análisis podría potenciar aún más la capacidad predictiva de los modelos y abrir nuevas vías de segmentación y personalización. Este enfoque representa un paso natural para ampliar el valor del proyecto y adaptarlo a escenarios reales de negocio donde el texto juega un papel central.

Anexos

- *Código: ProyectoFinal_Equipo37.ipynb*
- Video: *ProyectoFinal_Equipo37.mp4*