SISTEMA DE RECOMENDACIÓN INTELIGENTE PARA E-COMMERCE BASADO EN BIG DATA Y MACHINE LEARNING

Manuel Alejandro Andrade Franco, Jairo David Acevedo Jaramillo, Gabriel Arcángel Santiago Castañeda, Javier Ignacio González Álvarez, Andrés Puerta González

INTRODUCCIÓN

En la actualidad, el comercio electrónico ha experimentado un crecimiento acelerado, transformando la manera en que los consumidores acceden y compran productos. Plataformas como Amazon han sido clave en esta transformación, ofreciendo una vasta gama de productos y servicios, donde las opiniones de los usuarios juegan un papel crucial en la toma de decisiones de compra. Las reseñas de productos no solo sirven como una forma de retroalimentación directa, sino que también representan un recurso invaluable para las empresas que buscan entender las preferencias de los consumidores y mejorar sus estrategias comerciales. El análisis de estas reseñas y calificaciones ha llevado al desarrollo de sistemas de recomendación que, mediante el uso de algoritmos avanzados, personalizan la experiencia de compra, sugiriendo productos que probablemente interesen a los usuarios según sus preferencias previas (Ricci et al., 2015).

El objetivo general de este proyecto es desarrollar un sistema de recomendación de productos para plataformas de comercio electrónico que utilice las calificaciones de estrellas proporcionadas por los usuarios, optimizando la precisión y personalización de las sugerencias mediante la implementación de algoritmos de aprendizaje automático avanzados y técnicas de Big Data. Para lograr esto, se implementarán algoritmos como ALS (Alternating Least Squares), que se utiliza para el filtrado colaborativo, y K-means, que facilitará la segmentación de usuarios y productos. Además, se incorporarán modelos adicionales como KNNBasic y SVD (Singular Value Decomposition). El modelo KNNBasic emplea un enfoque basado en vecinos más cercanos para encontrar similitudes entre usuarios o productos, proporcionando una solución efectiva en entornos con datos densos. Por su parte, el modelo SVD descompone la matriz de calificaciones en componentes latentes, permitiendo capturar relaciones complejas entre usuarios y productos, lo que mejora la precisión en escenarios con datos dispersos (Deviane Puspita et al., 2021; Koren et al., 2009).

El Amazon Customer Reviews Dataset será la base de datos utilizada en este estudio, proporcionando millones de reseñas y calificaciones de productos. La primera fase del proyecto se enfocará en analizar y preprocesar este dataset, con el objetivo de identificar patrones en las calificaciones de estrellas y determinar cómo estos patrones pueden ser utilizados para predecir las preferencias de los usuarios en función de sus evaluaciones previas. Este análisis inicial permitirá obtener una comprensión más profunda de los comportamientos de los usuarios y cómo las calificaciones reflejan sus intereses y preferencias (Linden et al., n.d.; Schafer et al., 2007).

Además, se evaluarán y compararán diferentes algoritmos de aprendizaje automático, incluidos los enfoques de filtrado colaborativo y modelos basados en contenido, para determinar cuál de estos ofrece el mejor desempeño en la generación de recomendaciones precisas y personalizadas. El filtrado colaborativo será implementado utilizando el algoritmo ALS, que descompone la matriz de interacciones entre usuarios y productos para predecir calificaciones no observadas. Por su parte, el K-means se utilizará para segmentar tanto a los usuarios como a los productos en grupos homogéneos, lo que permitirá mejorar la personalización de las recomendaciones al considerar patrones comunes

en grupos específicos. Adicionalmente, el modelo KNNBasic contribuirá a la personalización mediante la identificación de patrones entre usuarios o productos similares, mientras que el SVD complementará el análisis al identificar factores latentes que impactan las preferencias de los usuarios, incluso en escenarios de alta dispersión de datos (Padhy et al., 2024; Zhou et al., 2008).

Un aspecto crítico de este proyecto será la optimización de los hiperparámetros de los algoritmos seleccionados para maximizar la precisión y personalización de las recomendaciones. Para ello, se emplearán técnicas de validación cruzada y métodos de optimización avanzados, con el fin de mejorar el rendimiento y la efectividad del sistema de recomendación en la plataforma de comercio electrónico. Este proceso garantizará que el modelo no solo sea preciso, sino también eficiente, capaz de adaptarse a un entorno de datos masivos y proporcionar recomendaciones de alta calidad a los usuarios (Jain et al., 2022)

Este sistema de recomendación, al combinar ALS para el filtrado colaborativo, K-means para la segmentación de usuarios y productos, KNNBasic para la identificación de similitudes y SVD para el análisis de factores latentes, busca generar recomendaciones personalizadas que no solo mejoren la experiencia del usuario, sino que también optimicen la toma de decisiones de compra, contribuyendo a aumentar las tasas de conversión y la fidelización de usuarios en plataformas de comercio electrónico. Al integrar estos algoritmos de machine learning y Big Data, el proyecto proporcionará un sistema escalable y robusto que puede manejar grandes volúmenes de datos y ofrecer resultados precisos y adaptados a las preferencias individuales de los consumidores.

Objetivo General

Desarrollar un sistema de recomendación de productos para plataformas de comercio electrónico que utilice las calificaciones de estrellas proporcionadas por los usuarios, con el fin de mejorar la precisión y personalización de las recomendaciones a través de la implementación de algoritmos avanzados de aprendizaje automático y técnicas de Big Data.

Objetivos Específicos

1. Realizar un análisis exhaustivo y preprocesamiento del Amazon Customer Reviews Dataset, con el propósito de identificar patrones relevantes en las calificaciones de estrellas (star_rating) y determinar de qué manera estos patrones pueden ser utilizados para predecir las preferencias de productos de los usuarios, basándose en sus interacciones y evaluaciones anteriores.
2. Evaluar y comparar el rendimiento de diversos algoritmos de aprendizaje automático, como el filtrado colaborativo y los modelos basados en contenido, para identificar cuál de estos enfoques genera las recomendaciones más precisas y personalizadas, optimizando así la experiencia del usuario en una plataforma de comercio electrónico.
3. Optimizar los hiperparámetros de los algoritmos seleccionados mediante el uso de técnicas de validación cruzada y otros métodos de optimización avanzados, con el objetivo de maximizar la precisión, la efectividad y la personalización de las recomendaciones, mejorando el rendimiento general del sistema de recomendación dentro de la plataforma de comercio electrónico.

METODOLOGÍA GENERAL

El análisis y modelado de los datos se desarrolló bajo un enfoque sistemático orientado al manejo de grandes volúmenes de datos y la extracción de patrones relevantes para sistemas de recomendación. La metodología empleada se estructuró en tres etapas principales:

1. **Carga y preprocesamiento de datos**:

El preprocesamiento del dataset combinó el uso de herramientas avanzadas para manejar grandes volúmenes de datos y optimizar la eficiencia del flujo de trabajo. Los datos, almacenados en un bucket de Amazon S3 en formato Parquet, se cargaron y transformaron en un DataFrame de Spark para aprovechar su capacidad de procesamiento distribuido. Este formato permitió optimizar la velocidad de acceso y compresión de los datos, mientras que Spark facilitó la ejecución paralela de tareas en múltiples nodos, maximizando la escalabilidad del sistema. Las columnas clave seleccionadas incluyeron identificadores únicos de usuarios y productos (customer_id, product_id), calificaciones (star_rating), votos (helpful_votes, total_votes) y la información de compra verificada (verified_purchase), que se transformó a un formato binario para facilitar su análisis (Gershinsky, 2018; Zaharia et al., 2010).

De manera complementaria, se empleó la librería Dask para etapas específicas del preprocesamiento, como la eliminación de valores nulos, la identificación de valores repetidos y la consolidación del dataset para trabajos exploratorios (Rocklin, 2015). Dask permitió particionar el dataset en fracciones manejables (particiones), distribuyendo las tareas en una cola de procesamiento que ejecuta operaciones de forma secuencial o paralela en memoria. Este enfoque, basado en el método de ejecución "perezosa" (lazy execution), optimiza el uso de la memoria RAM al realizar operaciones únicamente cuando se requiere el resultado específico (.compute()), escribiendo los resultados intermedios en disco para liberar memoria. Asimismo, se calculó la dispersión del dataset, revelando que menos del 1% de las combinaciones usuario-producto contenían información relevante, un patrón característico en sistemas de recomendación. Este flujo de trabajo integrado aseguró una preparación eficiente y robusta de los datos, adaptándose a los retos de escalabilidad y heterogeneidad típicos de entornos de Big Data.

1. **Análisis exploratorio de datos (EDA)**:

Se realizó un análisis estadístico detallado para entender la distribución de las variables y las correlaciones entre ellas. Se encontró una tendencia hacia calificaciones altas (promedio de 4.17) y relaciones débiles entre star_rating y los votos, lo que indica que las calificaciones no dependen directamente de estos factores. Se implementaron técnicas como la normalización y la reducción de dimensionalidad mediante PCA para capturar patrones latentes y preparar los datos para el modelado.

2. **Modelado predictivo y de agrupamiento**:

- **Modelo ALS**: Se implementó la técnica de Alternating Least Squares para predecir calificaciones, configurando iteraciones, regularización y factores latentes. Se probaron tres tamaños de muestra (20%, 40% y 100%), evaluando el modelo con el RMSE. Los resultados mostraron una mejora progresiva al incrementar la cantidad de datos utilizados.
- **Modelo K-Means**: Se agruparon usuarios y productos en clústeres homogéneos con y sin PCA, evaluando los resultados con el índice de Silhouette. Los cinco clústeres identificados

mostraron características diferenciadas, aunque el predominio de un clúster (99.7% de los datos) sugiere áreas de mejora en la segmentación.

EXPLORACIÓN Y MODELADO DE DATOS

Metodología de investigación

El estudio se desarrolló utilizando datos de un sistema de recomendaciones cargados desde un bucket de Amazon S3 en formato Parquet. La metodología se estructuró en tres etapas. Primero, se realizó una limpieza y preprocesamiento inicial, seleccionando columnas clave como customer_id, product_id, y star_rating, y eliminando valores nulos o inconsistentes. A continuación, se llevó a cabo un análisis exploratorio (EDA) para evaluar la estructura y características del conjunto de datos, calculando correlaciones y realizando normalizaciones para preparar los datos para el modelado. Por último, se implementaron modelos de agrupamiento con y sin reducción de dimensionalidad para segmentar los usuarios y productos.

Análisis de los datos

El conjunto de datos cuenta con **109,828,726 filas** y **7 columnas**, con un esquema que incluye identificadores únicos de usuarios y productos, calificaciones (star_rating), y votos (helpful_votes, total_votes). El análisis de densidad y esparcidad reveló que, aunque hay **27,535,894 usuarios únicos** y **15,219,332 productos únicos**, el dataset tiene una dispersión cercana al **100%**, lo que es común en sistemas de recomendación y refleja la naturaleza esporádica de las interacciones.

El resumen estadístico mostró que la media de las calificaciones es **4.17** con una desviación estándar de **1.27**, indicando una tendencia hacia calificaciones altas (Figura 1A). La correlación entre las variables mostró una relación positiva fuerte entre helpful_votes y total_votes (**0.987**), mientras que las relaciones entre star_rating y los votos fueron débiles y negativas (-0.02 y -0.04), lo que sugiere que las calificaciones no dependen directamente de los votos.

El análisis con reducción de dimensionalidad mediante PCA mostró que las dos primeras componentes principales capturan patrones clave en los datos. Por ejemplo, usuarios con interacciones promedio o atípicas se agruparon de manera diferenciada en el espacio de PCA. Esto permitió alimentar modelos de agrupamiento más robustos y reducir el ruido.

Se utilizaron modelos K-Means con y sin PCA para agrupar los datos. Sin PCA, el modelo obtuvo un **Silhouette Score de 0.9999**, lo que indica una alta cohesión y separación en los clústeres. Al incorporar PCA, se mantuvo un Silhouette Score similar, pero con una mejora en la interpretabilidad, asignando observaciones a cinco clústeres con diferencias significativas en patrones de interacción.

IMPLEMENTACIÓN Y EVALUACIÓN DE UN MODELO PREDICTIVO BASADO EN ALS

Metodología de investigación

Este estudio implementa un modelo predictivo basado en la técnica de **Alternating Least Squares (ALS)**, adaptado para sistemas de recomendación en entornos con grandes volúmenes de datos. La metodología se estructuró en tres etapas principales. En la primera etapa, se realizó la carga y preprocesamiento de los datos almacenados en formato **Parquet** en un bucket de Amazon S3, seleccionando las columnas esenciales (customer_index, product_index, star_rating) para alimentar

el modelo. En la segunda etapa, se configuraron y evaluaron diferentes versiones del modelo ALS variando el tamaño de la muestra (20%, 40% y 100% de los datos) para optimizar la precisión y escalabilidad. Finalmente, el modelo fue evaluado utilizando el error cuadrático medio (RMSE) como métrica principal, complementado con un análisis detallado de predicciones para identificar discrepancias y validar el desempeño.

El modelo ALS se configuró con 10 iteraciones máximas, un parámetro de regularización de 0.1 y 10 factores latentes. La división de los datos en conjuntos de entrenamiento (80%) y prueba (20%) se realizó de manera aleatoria y consistente en todas las iteraciones. Se utilizó la estrategia coldStartStrategy='drop' para manejar combinaciones de usuario-producto no vistas en el conjunto de entrenamiento, garantizando la integridad de las predicciones en el conjunto de prueba.

Análisis de los datos

El análisis comenzó con el cálculo de la distribución de la columna star_rating, identificando una tendencia predominante hacia calificaciones altas (5.0 y 4.0), que representan más del 75% de las evaluaciones totales. Este patrón sugiere una preferencia positiva entre los usuarios, aunque también podría reflejar un sesgo en las evaluaciones. Las calificaciones bajas (1.0 y 2.0) fueron significativamente menos frecuentes, indicando la necesidad de un modelo robusto que capte patrones incluso en datos desequilibrados.

Los resultados del RMSE muestran una mejora progresiva al incrementar el tamaño de los datos:

- **20% de los datos:** RMSE de **2.2730**, indicando un error moderado y limitaciones en la precisión debido al tamaño reducido del conjunto de entrenamiento.
- **40% de los datos:** RMSE de **2.0497**, reflejando una mejora sustancial al incorporar una muestra más representativa de la población.
- **100% de los datos:** RMSE de **1.8136**, logrando el mejor desempeño al utilizar el conjunto completo, capturando patrones más detallados y reduciendo el error.

El análisis de las predicciones mostró una precisión adecuada en muchos casos, como el usuario 25809837 con el producto 540, donde una calificación real de **5.0** fue predicha como **4.8441**. Sin embargo, también se identificaron discrepancias significativas, como el usuario 18073726 con el producto 2003, donde una calificación real de **3.0** fue predicha como **-0.4690**, lo que sugiere áreas de mejora en la optimización del modelo. Estas discrepancias podrían deberse a la naturaleza esporádica de ciertos datos o a la falta de relaciones sólidas entre algunos usuarios y productos.

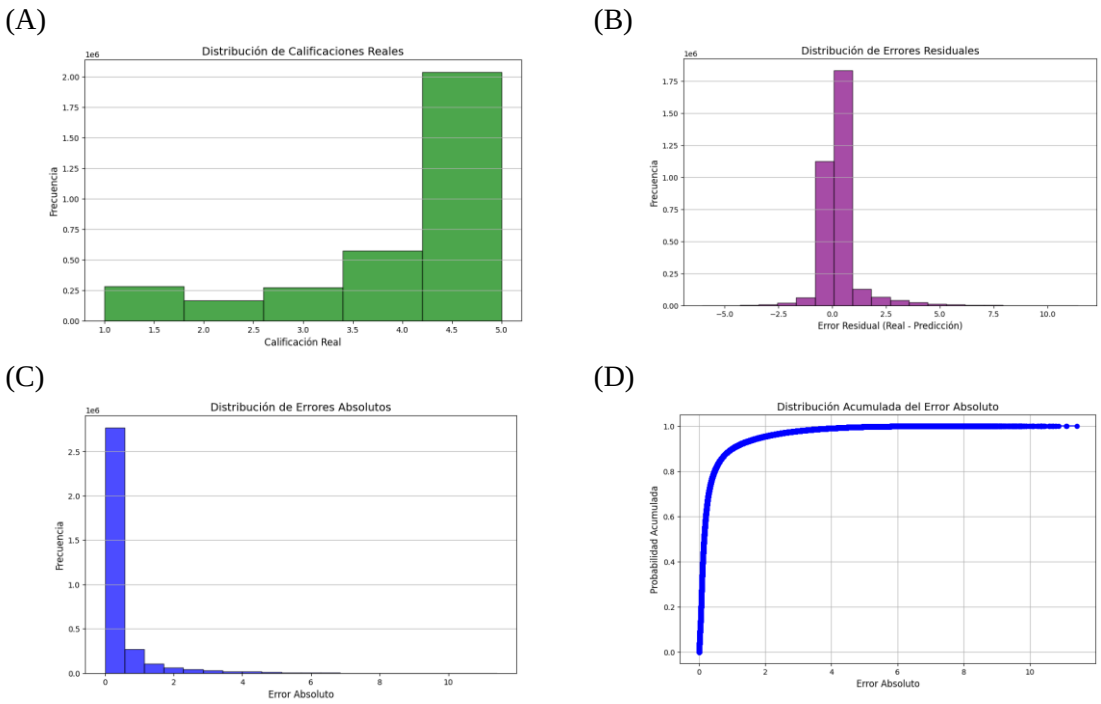

Figura 1. (A) Distribución de errores absolutos. (B) Distribución de calificaciones reales. (C) Distribución de errores residuales. (D) Distribución acumulada de errores absolutos.

La distribución de calificaciones reales muestra una tendencia marcada hacia valores altos, predominando las calificaciones de 4.0 y 5.0, lo que sugiere una percepción generalmente positiva de los usuarios. Las calificaciones bajas (1.0 y 2.0) son menos frecuentes, evidenciando un posible sesgo hacia la satisfacción o una menor disposición a registrar evaluaciones negativas. Esto resalta la importancia de que el modelo maneje adecuadamente datos desbalanceados, capturando tanto las preferencias mayoritarias como las excepciones (Figura A). La distribución de errores residuales, concentrada alrededor de 0, sugiere un balance adecuado entre sobreestimaciones y subestimaciones, aunque los valores extremos reflejan limitaciones en la captura de relaciones menos comunes entre usuarios y productos (Figura C). Además, la mejora del RMSE al incrementar el tamaño del conjunto de datos subraya la importancia de un conjunto representativo para reducir discrepancias.

El modelo predictivo basado en ALS demuestra resultados significativos en términos de precisión y robustez en la predicción de calificaciones. Las distribuciones analizadas muestran que los errores absolutos se concentran mayoritariamente cerca de 0, lo que evidencia que el modelo captura patrones predominantes con alta fidelidad. Sin embargo, se observan valores atípicos, posiblemente asociados a datos esporádicos o inconsistencias en las interacciones usuario-producto. La distribución acumulativa del error absoluto refuerza esta observación, indicando que más del 95% de los errores se encuentran por debajo de 2.0, lo que refleja un desempeño aceptable en la mayoría de los casos (Figuras D y B). En general, el modelo exhibe un balance positivo entre precisión y escalabilidad, siendo una herramienta prometedora para aplicaciones de recomendación en contextos complejos.

IMPLEMENTACIÓN Y EVALUACIÓN DE UN MODELO K-MEANS PARA DATOS DE CALIFICACIÓN DE USUARIOS

Metodología de investigación

Se diseñó un flujo de trabajo estructurado comenzando con la carga de datos desde un bucket de Amazon S3, almacenados en formato Parquet para optimizar la compresión y la velocidad de lectura. Posteriormente, se transformaron los datos en un DataFrame de Spark para aprovechar su capacidad de procesamiento distribuido. Las columnas clave seleccionadas incluyeron star_rating, helpful_votes, total_votes y verified_purchase, esta última transformada a un formato binario para facilitar el análisis numérico.

Para implementar el modelo K-Means, se ensamblaron estas variables en un vector de características mediante VectorAssembler. Se configuró el algoritmo con 5 clústeres, utilizando la métrica de distancia euclidiana y ajustando el modelo sobre el conjunto de datos completo. Finalmente, las predicciones se evaluaron mediante el índice de Silhouette, una métrica que mide la cohesión y separación de los clústeres.

Análisis de los datos

El análisis inicial de los datos mostró una distribución sesgada hacia calificaciones altas (5.0 y 4.0), que constituyeron la mayoría de las observaciones. Las calificaciones bajas (1.0 y 2.0) fueron menos comunes, lo que refleja una tendencia positiva entre los usuarios hacia los productos evaluados. Adicionalmente, el modelo K-Means identificó cinco clústeres con distribuciones heterogéneas:

- **Clúster 0**: Representó el 99.7% de las observaciones, agrupando a usuarios con características comunes pero menos distintivas.
- **Clúster 3**: Incluyó 227,774 observaciones, destacándose por características diferenciadas relacionadas con votos útiles y calificaciones medias.
- **Clústeres 1, 2 y 4**: Representaron grupos más pequeños, posiblemente usuarios con comportamientos atípicos o productos poco evaluados.

El índice de Silhouette obtenido fue de **0.996**, indicando una alta cohesión interna en los clústeres y una clara separación entre ellos. Sin embargo, la predominancia del clúster 0 sugiere que el modelo podría beneficiarse de una reconfiguración para capturar mejor la variabilidad latente en los datos.

IMPLEMENTACIÓN Y EVALUACIÓN DE UN MODELO KNNBasic y SVD PARA DATOS DE CALIFICACIÓN DE USUARIOS

Metodología de investigación

El desarrollo de los modelos de recomendación se centró en la descomposición de la matriz de interacciones usuario-producto (customer_id-product_id) para identificar factores latentes que reflejan relaciones subyacentes entre las preferencias de los usuarios y las características de los productos. Para implementar estos modelos, se utilizó la librería *Surprise* (Simple Python Recommendation System Engine), diseñada para sistemas de recomendación y que permite evaluar el desempeño de los modelos mediante métricas como RMSE, precisión, recall y F1-score (Hug, 2020).

En el modelo KNNBasic, se realizó un muestreo estratificado aleatorio del 1% del dataset original, compuesto por 109 millones de registros distribuidos en 37 categorías de productos. Este procedimiento buscó garantizar que el dataset final representara de manera precisa las interacciones reales entre usuarios y productos. Además, se evaluaron diferentes tamaños de dataset, desde el 0.03% hasta el 1%, con el objetivo de analizar cómo el volumen de datos influye en el rendimiento del modelo.

Por otro lado, se implementó un modelo SVD (Singular Value Decomposition), basado en la factorización matricial para predecir calificaciones faltantes en la matriz usuario-producto. Este enfoque descompone la matriz original en componentes más pequeñas, representando factores latentes que capturan características implícitas tanto de usuarios como de productos. En este modelo, se trabajó con un conjunto de datos más grande, cercano a 1 millón de registros, debido a su menor costo computacional en comparación con el modelo KNNBasic (Koren et al., 2009).

Análisis de los datos

Los resultados obtenidos para ambos modelos destacan la estabilidad y consistencia de sus métricas de desempeño. En el caso del modelo KNNBasic, el RMSE se estabilizó en 1.2751 a partir del 0.4% del dataset original, mientras que las métricas de precisión, recall y F1-score alcanzaron valores de 0.782 a partir del 0.1% de los datos. Como se observa en la gráfica, estas métricas muestran una estabilización significativa con tamaños reducidos del dataset, lo que resalta la eficiencia del modelo para manejar conjuntos de datos más pequeños. Sin embargo, este modelo exige un alto costo computacional, principalmente en la elaboración de matrices de interacción y similitud, lo que limitó el análisis a una muestra representativa de 32,951 registros (0.03% del dataset original).

En contraste, el modelo SVD logró un RMSE de 1.2650 y valores de precisión, recall y F1-score de 0.778, 0.781 y 0.779, respectivamente, utilizando cerca de 1 millón de registros. Aunque los resultados son similares a los del modelo KNNBasic, el menor costo computacional asociado al modelo SVD permitió trabajar con un conjunto de datos más grande. Sin embargo, al comparar ambos modelos, se concluye que un aumento significativo en el tamaño del dataset no genera una mejora sustancial en las métricas de desempeño, sugiriendo que el esfuerzo computacional adicional puede no ser justificable más allá de un tamaño óptimo del dataset.

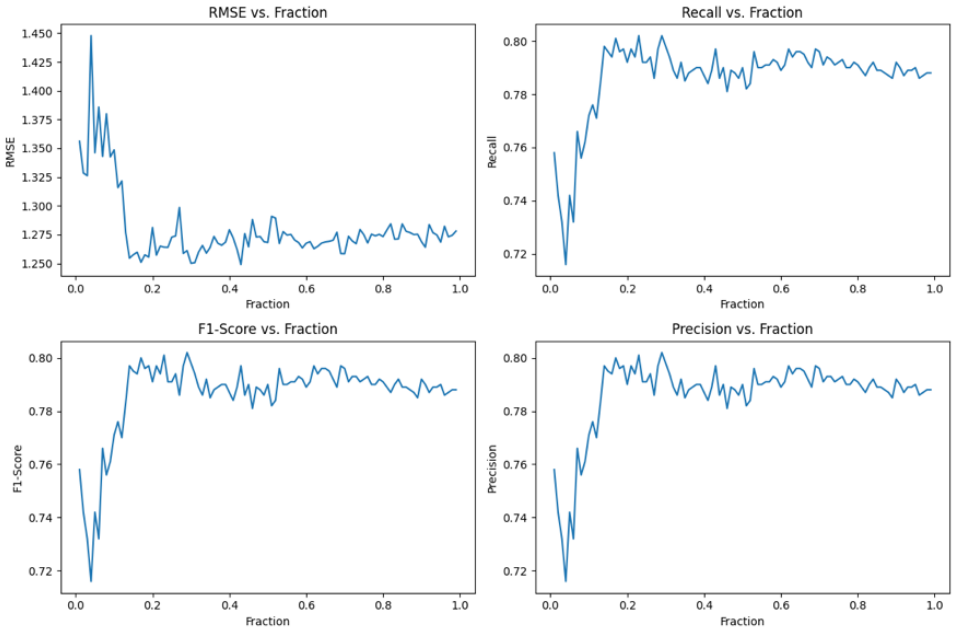

Figura 2. Relación entre el tamaño del dataset y las métricas de desempeño del modelo KNNBasic.

USO DE HERRAMIENTAS DE BIG DATA

La arquitectura utilizada en este proyecto combina Apache Spark y Amazon S3 para garantizar un manejo eficiente de grandes volúmenes de datos. Spark, configurado con memoria dedicada para ejecutores y controlador (entre 2 y 8 GB), se utilizó como motor de procesamiento distribuido, lo que permitió realizar tareas en paralelo y maximizar el rendimiento computacional. Además, el formato Parquet optimizó la compresión y el acceso a los datos, reduciendo los tiempos de carga y almacenamiento. Este enfoque no solo garantizó la eficiencia actual, sino que también sentó las bases para escalar el sistema frente a futuros incrementos en el volumen y la velocidad de los datos.

En cuanto a los modelos implementados, se utilizaron enfoques avanzados como ALS (Alternating Least Squares), KNNBasic y SVD (Singular Value Decomposition), cada uno diseñado para abordar aspectos específicos del problema. ALS permitió el filtrado colaborativo al descomponer la matriz de interacciones usuario-producto en factores latentes, capturando relaciones complejas en los datos. KNNBasic, implementado mediante la librería *Surprise* (Simple Python Recommendation System Engine), empleó un enfoque basado en vecinos más cercanos para identificar similitudes entre usuarios o productos, lo que resultó efectivo en entornos con datos densos. Por su parte, SVD descompuso la matriz usuario-producto en componentes latentes, reduciendo el ruido en los datos y mejorando la precisión en escenarios de alta dispersión.

Además de Spark y Surprise~~,~~ se utilizaron otras librerías para respaldar el flujo de trabajo. Es importante anotar que la biblioteca Surprise no trabaja bajo sistemas distribuidos de tareas, y particularmente no tiene una versión para correr bajo Sp-ark. Esto representaba un reto importante teniendo en cuenta el gran volumen de datos que se estaba manejando, las limitaciones que ofrece Google Collab en cuanto a recursos computacionales, y y las limitaciones de recursos de las máquinas locales (PC). Mediante la librería Dask se realizó e~~empleó en e~~l preprocesamiento de datos, al~~para~~ particionar el dataset en fracciones manejables y ejecutar tareas de manera distribuida. Esto resultó

fundamental para la eliminación de valores nulos, la consolidación de datos y el manejo eficiente de memoria mediante ejecución perezosa. Tambien para preprocesamiento de datos en pasos previos al modelamiento con la librería Surprise. Para el análisis exploratorio y visualización de resultados, se integraron librerías como Pandas, NumPy y Matplotlib, que facilitaron la normalización, el cálculo de métricas y la representación gráfica de los resultados obtenidos. Esta combinación de herramientas y modelos garantizó un sistema de recomendación robusto, escalable y adaptado a las necesidades específicas del proyecto.

ENTREGABLES Y SU DESCRIPCIÓN

https://github.com/Javsk891/Sistemas-de-Recomendaci-n---EAFIT

CONCLUSIONES Y TRABAJO FUTURO

El desarrollo del sistema de recomendación basado en técnicas avanzadas de filtrado colaborativo y modelos de agrupamiento ha permitido implementar un marco sólido para la personalización en plataformas de comercio electrónico. Los resultados obtenidos destacan la efectividad de los modelos ALS, KNNBasic y SVD, que demostraron un desempeño consistente en métricas clave como RMSE, precisión, recall y F1-score. En particular, el modelo SVD mostró ser una alternativa eficiente en términos computacionales al trabajar con conjuntos de datos más grandes sin sacrificar significativamente la precisión. La integración de herramientas de Big Data, como Apache Spark y Amazon S3, optimizó el manejo de datos masivos, mientras que el uso de Dask facilitó el preprocesamiento eficiente, empleando maquinas locales con pocos recursos disponibles (RAM). No obstante, la alta esparsidad de los datos y el predominio de interacciones positivas plantean desafíos que requieren soluciones más adaptativas. Este proyecto evidenció también la importancia de una planificación detallada en la gestión de recursos técnicos, como los entornos en AWS, y la necesidad de tiempos adecuados para la limpieza y normalización de datos. Aunque se presentaron contratiempos, la capacidad del equipo para adaptarse permitió completar el proyecto exitosamente, identificando áreas donde se puede mejorar, especialmente en la optimización de los procesos iniciales.

Una línea de trabajo futura sería integrar datos contextuales, como el tiempo de interacción o información demográfica de los usuarios, para enriquecer las predicciones y aumentar la personalización. La incorporación de estos datos podría facilitar el desarrollo de modelos híbridos que combinen filtrado colaborativo y técnicas basadas en contenido. También es fundamental explorar técnicas que reduzcan el costo computacional del modelo KNNBasic, como la implementación de algoritmos más rápidos, la reducción de dimensionalidad mediante PCA o el uso de infraestructuras más avanzadas, como GPU o clusters de alto rendimiento. Además, abordar el sesgo hacia calificaciones altas y mejorar la equidad en las recomendaciones mediante métodos de balanceo de datos o métricas que valoren la diversidad y la justicia en las sugerencias fortalecería el sistema. Adaptar el sistema para manejar flujos de datos en tiempo real representaría una extensión lógica que permitiría generar recomendaciones dinámicas basadas en las interacciones recientes de los usuarios. Por último, incluir métricas orientadas a la experiencia del usuario, como la novedad y la diversidad, complementaría las métricas actuales y ofrecería una evaluación más completa del impacto del sistema en el entorno de comercio electrónico. Estas propuestas contribuirían a mejorar la robustez, escalabilidad y adaptabilidad del sistema en un entorno de constante evolución.

EJECUCIÓN DEL PLAN

La limpieza de datos y el manejo de valores atípicos requirieron más tiempo del previsto, no solo por la complejidad inherente a los datos, sino también por las dificultades en la gestión de los entornos dispuestos para estudiantes en AWS. Este retraso impactó directamente la fase de preprocesamiento, obligando a ajustar los tiempos. Sin embargo, gracias a la capacidad de adaptación del equipo, las fases posteriores, como la Implementación de Modelos y la Evaluación y Validación, lograron alinearse casi por completo con los tiempos establecidos en el plan inicial.

Como lección aprendida, queda claro que tanto la calidad de los datos como la gestión de entornos técnicos, como los ofrecidos en AWS, son aspectos críticos que requieren una planificación más detallada. Incorporar mayores márgenes de tiempo y anticipar posibles dificultades técnicas habría permitido mitigar el impacto de los imprevistos y evitar retrasos acumulativos. Aunque se enfrentaron contratiempos y se limitó parcialmente el alcance inicial, el equipo mostró flexibilidad y logró completar las fases finales del proyecto con éxito.

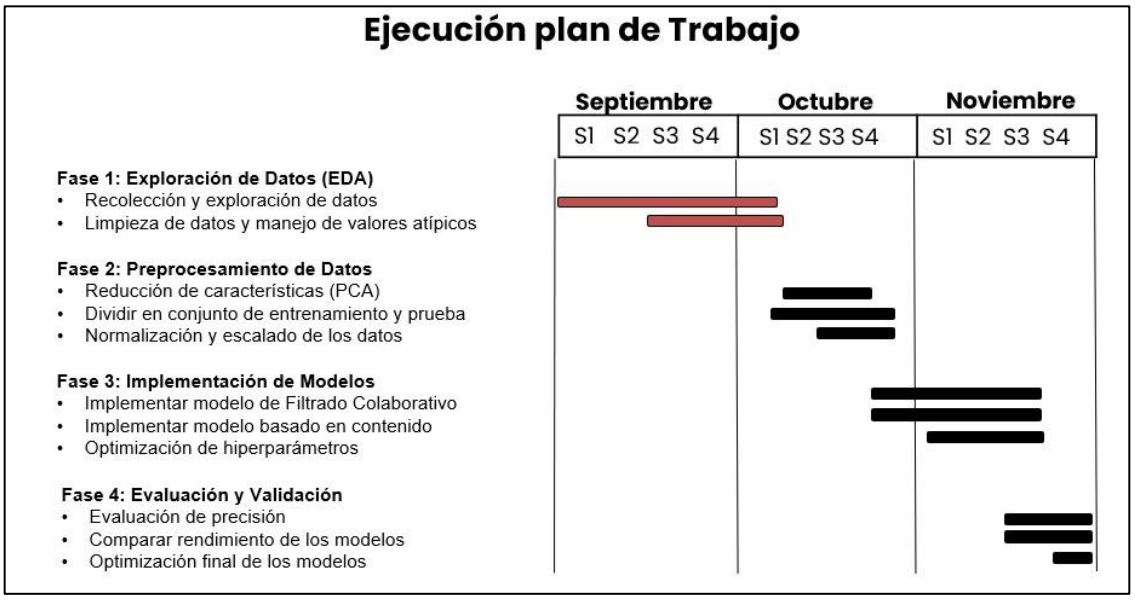

Figura 3. Ejecución y cronograma del plan de trabajo para el desarrollo del sistema de recomendación.

IMPLICACIONES ÉTICAS

El desarrollo de sistemas de recomendación como el implementado en este trabajo implica consideraciones éticas críticas. En primer lugar, la recopilación y análisis de grandes volúmenes de datos personales deben cumplir estrictamente con las normativas de privacidad y protección de datos, como el GDPR o la Ley de Protección de Datos Personales (Mittelstadt et al., 2016). Además, el sesgo inherente en los datos, como la predominancia de calificaciones altas, podría perpetuar desigualdades o favorecer ciertos productos de manera desproporcionada, afectando la equidad del sistema de recomendación (Friedman & Nissenbaum, 1996). También es fundamental garantizar la transparencia del modelo, asegurando que los usuarios entiendan cómo se generan las recomendaciones y permitiendo auditorías independientes que evalúen posibles impactos negativos,

como la exclusión de ciertos grupos de usuarios (Jobin et al., 2019). Finalmente, se debe evitar la manipulación de resultados que prioricen intereses comerciales sobre la experiencia genuina del consumidor.

ASPECTOS LEGALES Y COMERCIALES

El diseño e implementación de sistemas de recomendación deben abordar aspectos legales y comerciales fundamentales. Desde el punto de vista legal, la protección de datos personales es prioritaria, y el cumplimiento de normativas como el Reglamento General de Protección de Datos (GDPR) en Europa asegura que los datos recopilados sean tratados con transparencia y consentimiento explícito por parte de los usuarios. Estas normativas buscan garantizar derechos como el acceso, rectificación y eliminación de información personal (Mittelstadt et al., 2016; Wachter et al., 2017). En el ámbito comercial, los sistemas de recomendación representan una ventaja competitiva al personalizar la experiencia del cliente, pero también pueden plantear desafíos como el posible sesgo algorítmico que afecte la equidad de las sugerencias, lo que podría derivar en demandas o pérdidas reputacionales. Las empresas deben equilibrar el cumplimiento normativo con la generación de valor comercial, manteniendo la confianza del usuario como eje central de sus estrategias (Binns, 2018; Morley et al., 2021).

BIBLIOGRAFÍA

Binns, R. (2018). Fairness in Machine Learning: Lessons from Political Philosophy. *Proceedings of Machine Learning Research*, *81*(81), 1–11.

Deviane Puspita, A., Permadi, V. A., Anggani, A. H., & Christy, E. A. (2021). Musical Instruments Recommendation System Using Collaborative Filtering and KNN. *Proceedings Universitas Muhammadiyah Yogyakarta Undergraduate Conference*, *1*(2), 1–6.

Friedman, B., & Nissenbaum, H. (1996). Bias in Computer Systems. *ACM Transactions on Information Systems*, *14*(3), 330–347.

Gershinsky, G. (2018). *Efficient Analytics on Encrypted Data*. 121–121. https://doi.org/10.1145/3211890.3211907

Hug, N. (2020). Surprise: A Python library for recommender systems. *Journal of Open Source Software*, *5*(52), 2174. https://doi.org/10.21105/joss.02174

Jain, G., Mahara, T., & Sharma, S. C. (2022). Performance Evaluation of Time-based Recommendation System in Collaborative Filtering Technique. *Procedia Computer Science*, *218*, 1834–1844. https://doi.org/10.1016/j.procs.2023.01.161

Jobin, A., Ienca, M., & Vayena, E. (2019). The global landscape of AI ethics guidelines. *Nature Machine Intelligence*, *1*(9), 389–399. https://doi.org/10.1038/s42256-019-0088-2

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *8*(42), 30–37.

Linden, G., Smith, B., & York, J. (n.d.). *Amazon.com Recommendations*. http://computer.org/internet/

Mittelstadt, B. D., Allo, P., Taddeo, M., Wachter, S., & Floridi, L. (2016). The ethics of algorithms: Mapping the debate. *Big Data and Society*, *3*(2). https://doi.org/10.1177/2053951716679679

Morley, J., Floridi, L., Kinsey, L., & Elhalal, A. (2021). From What to How: An Initial Review of Publicly Available AI Ethics Tools, Methods and Research to Translate Principles into Practices. In *Philosophical Studies Series* (Vol. 144, pp. 153–183). Springer Nature. https://doi.org/10.1007/978-3-030-81907-1_10

Padhy, N., Suman, S., Priyadarshini, T. S., & Mallick, S. (2024). A Recommendation System for E-Commerce Products Using Collaborative Filtering Approaches. *The 3rd International Electronic Conference on Processes*, 50. https://doi.org/10.3390/engproc2024067050

Ricci, F., Shapira, B., & Rokach, L. (2015). Recommender systems: Introduction and challenges. In *Recommender Systems Handbook, Second Edition* (pp. 1–34). Springer US. https://doi.org/10.1007/978-1-4899-7637-6_1

Rocklin, M. (2015). Dask: Parallel Computation with Blocked algorithms and Task Scheduling. In *PROC. OF THE 14th PYTHON IN SCIENCE CONF*. https://www.youtube.com/watch?v=1kkFZ4P-XHg

Schafer, B., Frankowski, D., & Sen, S. (2007). Collaborative Filtering Recommender Systems. In Springer (Ed.), *The adaptive web: methods and strategies of web personalization* (pp. 291–324). https://www.researchgate.net/publication/200121027

Wachter, S., Mittelstadt, B., & Floridi, L. (2017). Why a Right to Explanation of Automated Decision-Making Does Not Exist in the General Data Protection Regulation. *International Data Privacy Law*, *7*(2), 76–99. https://doi.org/10.1609/aimag.v38i3.2741

Zaharia, M., Chowdhury, M., Franklin, M. J., & Shenker, S. (2010). Spark: Cluster Computing with Working Sets. *Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing (HotCloud)*, *10*, 1–7.

Zhou, Y., Wilkinson, D., Schreiber, R., & Pan, R. (2008). Large-scale Parallel Collaborative Filtering for the Netflix Prize. In *Algorithmic Aspects in Information and Management: 4th International Conference, AAIM 2008* (pp. 337–348). Springer.