

# **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

## **Facultad de Ingeniería en Electricidad y Computación**

Desarrollo de un sistema de recomendación de compra utilizando  
algoritmos de aprendizaje automático de sugerencia para la promoción  
efectiva de combos y ofertas

### **PROYECTO DE TITULACIÓN**

Previo la obtención del Título de:

#### **Máster en Ciencias de Datos**

Presentado por:

Bravo Muentes Kevin Hernán

Viñan Camacho Ana Belén

GUAYAQUIL - ECUADOR

Año: 2024

## DEDICATORIA

A Dios, mis padres, mis hermanas, mi sobrino Sebas y mi novia, gracias por ser mi refugio en los días difíciles, mi alegría en los días buenos y mi motor para seguir adelante. Cada paso que doy está lleno del amor que ustedes me han dado. Esto es para ustedes, porque este viaje no lo hice solo.

*Kevin Bravo*

A mis padres, abuelas y tías, por enseñarme con su ejemplo el valor del esfuerzo y la perseverancia.

A mis hermanas, por su cariño incondicional y apoyo en cada paso de este camino.

A mi novio, mi gran apoyo durante esta tesis, por su paciencia, motivación y por creer en mí incluso en los momentos más desafiantes.

A mis sobrinos, Cristina y Sebastián, porque su alegría y curiosidad me inspiran cada día, y su amor da sentido a cada esfuerzo. Que nunca dejen de soñar y luchen siempre por sus metas con valentía.

*Ana Belén Viñan*

## AGRADECIMIENTOS

A Dios, por ser mi luz y fortaleza en cada paso de este camino.

A mi madre y padre, por enseñarme con su tenacidad el valor del estudio y la importancia de nunca rendirse.

A Anita Viñán, por su paciencia, su apoyo incondicional y por ser una compañera excepcional en este viaje.

Y nuestro tutor de tesis, por su guía en este largo trabajo.

A todos ustedes, mi más sincero agradecimiento, pues este logro es tanto mío como suyo.

*Kevin Bravo*

Agradezco a Dios, por darme la sabiduría y fortaleza necesarias para superar cada desafío en este camino académico.

A la SENESCYT, por la oportunidad brindada a través de la beca que me permitió cursar esta maestría, contribuyendo al fortalecimiento de mi crecimiento profesional.

A Kevin, mi compañero de tesis, por su apoyo y compromiso durante este proceso, haciendo más llevadero cada desafío.

A nuestro tutor Sergio Bauz Olvera, por su invaluable guía durante el desarrollo de este trabajo.

*Ana Belén Viñan*

## **DECLARACIÓN EXPRESA**

Nosotros Kevin Hernán Bravo Muentes y Ana Belén Viñan Camacho acordamos y reconocemos que: La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá a los autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores. El o los estudiantes deberán procurar en cualquier caso de cesión de sus derechos patrimoniales incluir una cláusula en la cesión que proteja la vigencia de la licencia aquí concedida a la ESPOL.

La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, secreto empresarial, derechos patrimoniales de autor sobre software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por mí/nosotros durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que me/nos corresponda de los beneficios económicos que la ESPOL reciba por la explotación de mi/nuestra innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique a los autores que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Kevin Hernán Bravo  
Muentes

Ana Belén Viñan  
Camacho

## **COMITÉ EVALUADOR**

---

**Sergio Bauz Olvera, Ph.D.**

PROFESOR TUTOR

---

**Allan Avendaño, MSc.**

PROFESOR EVALUADOR

## RESUMEN

En el sector retail, mejorar la oferta de productos es fundamental para aumentar las ventas y mejorar la experiencia del cliente. Este estudio desarrolla un sistema de recomendación de productos para equipos comerciales y de ventas, con el objetivo de sugerir combinaciones de productos que mejoren la rentabilidad. Se implementaron tres algoritmos de aprendizaje automático: K-Nearest Neighbors (KNN), Alternating Least Squares (ALS) y Neural Collaborative Filtering (NCF), evaluando su desempeño con Precisión, Recall y F1-Score.

Se analizaron datos históricos de compras para identificar patrones de consumo y segmentar productos. Luego, los modelos fueron entrenados y evaluados para determinar el de mejor desempeño en la generación de recomendaciones. Los resultados indicaron que el modelo NCF obtuvo la mejor precisión, proporcionando sugerencias más acertadas.

Además, se diseñó un prototipo de aplicación interactiva que permite a los equipos comerciales seleccionar productos y recibir recomendaciones en tiempo real. Este enfoque facilita la planificación de promociones y mejora la toma de decisiones basada en datos.

Se concluye que el sistema desarrollado representa una herramienta útil para la gestión de productos en retail, mejorando la experiencia del cliente y la eficiencia de las estrategias comerciales.

**Palabras Clave:** Sistemas de recomendación, Retail, Aprendizaje automático, Ciencia de datos.



## ***ABSTRACT***

In the retail sector, improving product offerings is essential for increasing sales and enhancing the customer experience. This study develops a product recommendation system aimed at commercial and sales teams, with the goal of suggesting product combinations that improve profitability. Three machine learning algorithms were implemented: K-Nearest Neighbors (KNN), Alternating Least Squares (ALS), and Neural Collaborative Filtering (NCF), evaluating their performance using Precision, Recall, and F1-Score.

Historical purchase data was analyzed to identify consumption patterns and segment products. The models were then trained and evaluated in different scenarios to determine the best-performing model for generating recommendations. The results show that NCF achieved the best performance, providing more accurate suggestions. Additionally, an interactive application prototype was developed, allowing commercial teams to select products and receive real-time recommendations. This approach facilitates promotion planning and improves data-driven decision-making.

It is concluded that the developed recommendation system serves as a valuable tool for product management in retail, enhancing the customer experience and the efficiency of commercial strategies.

**Keywords:** Recommendation systems, Retail, Machine learning, Data science.

## ÍNDICE GENERAL

COMITÉ EVALUADOR.....	7
RESUMEN .....	I
<i>ABSTRACT</i> .....	II
ÍNDICE GENERAL .....	III
ABREVIATURAS.....	VII
ÍNDICE DE FIGURAS .....	VIII
ÍNDICE DE TABLAS.....	IX
CAPÍTULO 1 .....	1
1.    Introducción .....	1
1.1    Descripción del problema .....	3
1.2    Justificación del problema .....	4
1.3    Objetivos .....	5
1.3.1    Objetivo General .....	5
1.3.2    Objetivos Específicos .....	5
1.4    Metodología.....	6
1.4.1    Recopilación de Datos: .....	6
1.4.2    Preprocesamiento de Datos: .....	7
1.4.3    Implementación de Algoritmos de aprendizaje automático: .....	7
1.4.4    Desarrollo de un prototipo en un framework de visualización: .....	8
1.4.5    Evaluación de modelos: .....	8
1.4.6    Implementación y Monitoreo: .....	8
1.5    Resultados esperados .....	8
1.6    Dataset.....	9
CAPÍTULO 2 .....	11

2.	ESTADO DEL ARTE.....	11
2.1	Marco referencial.....	11
2.1.1	Servicios de Recomendación Basados en Filtrado Colaborativo Adaptativo Extendidos con KNN. ....	11
2.1.2	Algoritmo Alternating Least Squares (ALS) para recomendaciones en tiempo real .....	12
2.1.3	Algoritmo Neural Collaborative Filtering (NCF) para sistemas de recomendación.....	13
2.2	Sistemas de recomendación.....	15
2.2.1	Filtrado Colaborativo .....	16
2.2.2	Basados en Contenido.....	18
2.2.3	Híbridos .....	20
2.2.4	Otros Sistemas .....	23
2.3	Algoritmos de recomendación .....	24
2.3.1	K-NN.....	25
2.3.2	ALS .....	28
2.3.3	NCF .....	33
2.4	Comparación Teórica de los Algoritmos de Recomendación .....	36
2.5	Relación entre sistemas y algoritmos de recomendación .....	37
2.6	Medidas de desempeño .....	38
2.6.1	Precisión (Precision) .....	38
2.6.2	Recall.....	38
2.6.3	F1 Score .....	39
2.6.4	RMSE (Root Mean Square Error).....	39
2.6.5	MAE (Mean Absolute Error) .....	40
2.6.6	AUC (Área Bajo la Curva) .....	40

2.6.7	Precision@k.....	42
CAPÍTULO 3 .....		43
3.	METODOLOGÍA .....	43
3.1	Esquema de la metodología .....	43
3.1.1	Ingesta y Transformación de Datos.....	44
3.1.2	Segmentación de clientes .....	45
3.1.3	Implementación de Algoritmos .....	46
3.1.4	Métricas de Evaluación del Modelo .....	51
3.1.5	Generación y Almacenamiento de Recomendaciones .....	51
3.1.6	Implementación del Dashboard.....	52
3.2	Infraestructura para procesamiento y almacenamiento.....	55
3.2.1	Procesamiento de Datos .....	56
3.2.2	Escalabilidad.....	56
3.3	Almacenamiento de Datos.....	56
3.3.1	Consideraciones Futuras .....	57
3.3.2	Criterios de selección de combos.....	58
CAPÍTULO 4 .....		59
4.	RESULTADOS .....	59
4.1	Recolección de Datos y Estrategias para la Evaluación del Proyecto .....	59
4.1.1	Análisis de Segmentación de Clientes .....	60
4.1.2	Selección de Productos Relevantes .....	63
4.1.3	Selección de hiperparámetros.....	63
4.1.4	Back Testing .....	64
4.1.5	Comparación de Modelos .....	65
4.2	Puesta en Marcha y Funcionamiento.....	66
4.2.1	Limitaciones de Recursos y Estrategia de Implementación en Fases ...	67

4.2.2	Producción y Visualización de Recomendaciones .....	67
4.3	Pruebas de Funcionalidad .....	71
4.3.1	Evidencias de Funcionamiento del Prototipo .....	72
4.3.2	Pruebas de Tiempo de Respuesta .....	72
4.3.3	Pruebas de Precisión .....	72
4.3.4	Pruebas de Usabilidad y Experiencia de Usuario del Prototipo .....	73
4.3.5	Pruebas de Impacto Simulado en la Tasa de Conversión .....	73
4.4	Análisis Costo/Beneficio .....	74
4.4.1	Desglose de Costos del Prototipo .....	74
4.4.2	Proyección de Costos para una Implementación en Entorno Real .....	75
4.4.3	Proyección de Beneficios .....	76
4.4.4	Consideraciones Finales .....	77
CAPÍTULO 5	.....	78
5.	Conclusiones Y Recomendaciones .....	78
5.1	Conclusiones .....	78
5.2	Recomendaciones .....	79
BIBLIOGRAFÍA	.....	80

## **ABREVIATURAS**

KNN K-Nearest Neighbors

ALS Alternating Least Squares

NCF Neural Collaborative Filtering

## ÍNDICE DE FIGURAS

Figura 1-1 <i>Metodología del proyecto.</i> .....	6
Figura 2-1 <i>Evolución Histórica de los Algoritmos de Recomendación</i> .....	15
Figura 2-2 <i>Clasificación de Sistemas de Recomendación.</i> .....	16
Figura 2-3 <i>Clasificación de Algoritmos de Recomendación.</i> .....	24
Figura 2-4 <i>Flujograma del Algoritmo k-Nearest Neighbors (k-NN).</i> .....	26
Figura 2-5 <i>Flujograma del Algoritmo Alternating Least Squares (ALS).</i> .....	30
Figura 2-6 <i>Flujograma del Algoritmo Neural Collaborative Filtering (NCF).</i> .....	34
Figura 2-7 <i>Ilustración de Curva ROC.</i> .....	41
Figura 3-1 <i>Metodología del Proyecto.</i> .....	43
Figura 3-2 <i>Proceso de Implementación del Algoritmo k-NN.</i> .....	47
Figura 3-3 <i>Proceso de Implementación del Algoritmo ALS.</i> .....	49
Figura 3-4 <i>Proceso de Implementación del Algoritmo NCF.</i> .....	50
Figura 3-5 <i>Proceso de Generación y Almacenamiento de Recomendaciones.</i> .....	52
Figura 3-6 <i>Ventajas del uso de Streamlit.</i> .....	53
Figura 3-7 <i>Funcionalidades clave del prototipo del dashboard</i> .....	54
Figura 3-8 <i>Proceso iterativo de mejora del dashboard.</i> .....	55
Figura 3-9 <i>Opciones de escalabilidad para el sistema</i> .....	57
Figura 4-1 <i>Distribución de clientes por sexo</i> .....	59
Figura 4-2 <i>Determinación del número óptimo de clústers mediante el Método del Codo</i> .....	60
Figura 4-3 <i>Segmentación de clientes mediante K-Means (2D)</i> .....	61
Figura 4-4 <i>Visualización tridimensional de la segmentación de clientes con k-Means.</i> .....	63
Figura 4-5 <i>Pantalla de autenticación de usuario.</i> .....	68
Figura 4-6 <i>Pantalla de selección de productos.</i> .....	69
Figura 4-7 <i>Pantalla de combos recomendados.</i> .....	70
Figura 4-8 <i>Resumen y análisis de los combos recomendados.</i> .....	71

## ÍNDICE DE TABLAS

Tabla 1-1 <i>Descripción del dataset inicial</i> .....	10
Tabla 2-1 <i>Matriz comparativa de Algoritmos de Recomendación</i> . ....	37
Tabla 3-1 <i>Configuración de hiperparámetros para el algoritmo k-NN</i> .....	47
Tabla 3-2 <i>Configuración de hiperparámetros para el algoritmo ALS</i> .....	48
Tabla 3-3 <i>Configuración de hiperparámetros para el algoritmo NCF</i> . ....	50
Tabla 4-1 <i>Hiperparámetros óptimos seleccionados</i> .....	64
Tabla 4-2 <i>Comparación de modelos – Back testing</i> .....	65
Tabla 4-3 <i>Comparación de modelos en la Implementación</i> .....	66
Tabla 4-4 <i>Impacto del Sistema de Recomendación en la Tasa de Conversión</i> .....	74
Tabla 4-5 <i>Desglose de Costos del Prototipo</i> .....	75
Tabla 4-6 <i>Proyección de Costos para una Implementación en Entorno Real</i> .....	76
Tabla 4-7 <i>Beneficio proyectado de la Implementación del Sistema de recomendación</i> . .....	77



# CAPÍTULO 1

## 1. INTRODUCCIÓN

Actualmente, los sistemas de recomendación han adquirido una importancia crucial debido al gran volumen de información disponible y la necesidad de personalizar las experiencias de los usuarios (Ricci et al., 2011). Estos sistemas son herramientas esenciales en diversas plataformas, tales como tiendas en línea, servicios de streaming de música y video, y aplicaciones sociales, proporcionando una interfaz personalizada que puede aumentar significativamente la satisfacción y retención del cliente (Adomavicius & Tuzhilin, 2005). En el sector retail, la personalización de las ofertas y promociones constituye un eje estratégico para diferenciarse en un entorno competitivo y dinámico (Linden et al., 2003).

Los sistemas de recomendación funcionan analizando datos históricos a gran escala para determinar tendencias y correlaciones que no son evidentes de forma preliminar, utilizan algoritmos de aprendizaje automático para procesar esta información y generar sugerencias personalizadas que pueden incluir productos, servicios o contenidos específicos (Koren et al., 2009). Estos sistemas permiten ofrecer una experiencia satisfactoria para el usuario y a su vez, optimizan estrategias de marketing de las empresas al dirigir ofertas personalizadas que tienen más probabilidades de venta (Ricci et al., 2011).

El propósito del proyecto es desarrollar un Sistema de recomendación de productos para optimizar la oferta de combos y promociones en el sector retail. Este sistema está diseñado específicamente para compradores dentro de una empresa comercializadora de productos de venta masiva, con el fin de respaldar decisiones estratégicas informadas, mediante la personalización de combos basados en patrones de consumo, con ello se busca no solo incrementar las ventas, sino también proporcionar un mayor valor al cliente final.

Este sistema se diseñará para emplear técnicas avanzadas de machine learning, como el filtrado colaborativo y los modelos de factores latentes, para analizar patrones de compra y proporcionar recomendaciones personalizadas (Hu et al., 2008). La implementación se realizará en varias fases, iniciando con la recopilación y preprocesamiento de datos, seguida por la aplicación de algoritmos de aprendizaje automático y el desarrollo de un prototipo de aplicación en un framework de visualización interactiva.

La personalización en el comercio minorista permite incrementar el interés en los clientes, además puede impactar directamente en su fidelización. Las estrategias de recomendación basadas en datos precisos y modelos predictivos permiten a las empresas ofrecer productos y promociones que están alineados con las preferencias individuales de los consumidores, lo que resulta clave para tener una posición relevante en el mercado (Verhoef et al., 2015).

En el contexto específico de Ecuador, la implementación de un sistema de recomendación efectivo puede significar una ventaja competitiva significativa. Dada la diversidad del mercado y las variaciones en las preferencias del consumidor, un sistema que pueda adaptarse rápidamente a los cambios y ofrecer recomendaciones precisas es crucial. Además, el potencial de manejar una gran cantidad de información histórica de ventas permitirá a las empresas ecuatorianas no solo mejorar sus estrategias de marketing, sino también mejorar el control de stock.

Finalmente, la evaluación del sistema de recomendación se llevará a cabo utilizando diversas métricas de rendimiento, para asegurar que las recomendaciones sean tanto relevantes como oportunas. Se espera que este proyecto no solo incremente la personalización de combos en ofertas y promociones, sino que también establezca un nuevo estándar en rendimiento de las estrategias de mercadeo basadas en datos en el sector retail ecuatoriano.

## **1.1 Descripción del problema**

El sector retail enfrenta importantes desafíos en la gestión de combos y ofertas debido a la alta competitividad del mercado. A nivel mundial, los minoristas deben ajustarse a las preferencias dinámicas de los clientes y a la creciente demanda de personalización en las ofertas. Las promociones y descuentos son estrategias clave para atraer y retener clientes, pero su implementación efectiva sigue siendo un reto importante. La incapacidad de gestionar estas ofertas de manera eficiente puede llevar a una disminución en las ventas y a una experiencia del cliente menos favorable (Brynjolfsson et al., 2013; Rigby, 2011).

En Latinoamérica, la situación es similar, con empresas que enfrentan una intensa competencia y consumidores que buscan valor en cada compra. Las promociones basadas en datos y la personalización de ofertas han comenzado a ganar terreno, pero muchas empresas todavía dependen de enfoques tradicionales y empíricos. Esta dependencia de métodos intuitivos, en lugar de análisis de datos precisos, limita la capacidad de los minoristas para captar efectivamente el interés del consumidor y mejorar sus estrategias de marketing (Verhoef et al., 2015).

En Ecuador, el sector retail ha experimentado cambios significativos debido a la implantación del modelo Hard-Discount en 2019 y la crisis económica desencadenada por la pandemia. Estos factores han impulsado a los consumidores a priorizar el ahorro y buscar alternativas más accesibles, intensificando rivalidad entre empresas del mercado. Ante esto, la exigencia de implementar estrategias de ventas más efectivas se ha vuelto crítica para los minoristas que buscan mantenerse competitivos en este entorno desafiante.

La gestión de combos y ofertas en el sector retail ecuatoriano se lleva a cabo frecuentemente de manera empírica, una práctica dificulta que las empresas satisfagan oportunamente los intereses y demandas de los consumidores. Estas promociones, basadas más en intuición que en análisis de datos, a menudo no consiguen atraer eficazmente el interés del cliente, lo que puede ocasionar una disminución en las ventas y además un deterioro en la experiencia de compra del

consumidor, resultando en un desempeño económico insatisfactorio para la empresa (Hübner et al., 2016).

## **1.2 Justificación del problema**

Este proyecto apunta a transformar la gestión de combos y ofertas en una empresa comercializadora de productos de venta masiva a nivel nacional, mediante el desarrollo de un sistema de recomendación basado en técnicas de aprendizaje automático para identificar patrones latentes en los datos de compra, permitiendo una mejor selección de combinaciones de productos, con el fin de incrementar las ventas y ofrecer una experiencia más satisfactoria para el cliente.

Para lograrlo, es esencial analizar los patrones de compra que pueden utilizarse para recomendar productos de manera efectiva, alineando así las recomendaciones con los objetivos comerciales y operativos de la empresa. De esta manera, el sistema no solo generará recomendaciones precisas, sino que también contribuirá a fortalecer la estrategia comercial.

En este estudio, se evaluarán al menos tres metodologías de aprendizaje automático. Primero, el Filtrado Colaborativo basado en Vecinos que se centra en las similitudes entre usuarios o productos para hacer recomendaciones. Este enfoque es eficaz para encontrar similitudes claras, pero puede enfrentar problemas de escalabilidad y falta de datos iniciales suficientes (Sarwar et al., 2001). En segundo lugar, los Modelos de Factores Latentes, como el ALS (Alternating Least Squares), que utilizan técnicas de descomposición matricial para descubrir patrones latentes en las interacciones usuario-producto, siendo útiles para manejar datos dispersos y efectivos incluso con datos limitados (Koren et al., 2009; Lops et al., 2011).

Finalmente, el Neural Collaborative Filtering (NCF), una técnica que emplea redes neuronales para modelar interacciones entre usuarios y productos, superando restricciones de métodos antiguos. Este enfoque permite mejorar la precisión de

las recomendaciones, permitiendo generar predicciones más personalizadas en comparación con técnicas convencionales (He et al., 2017).

La implementación de estos modelos predictivos permitirá una personalización avanzada de las ofertas, asegurando que cada promoción sea relevante y oportuna. Esto no solo favorecerá las estrategias de marketing, sino que también permitirá mejorar la oferta de combos y la experiencia del cliente, proporcionando recomendaciones más alineadas con sus preferencias y hábitos de compra. Como resultado, se espera incrementar la fidelización de clientes mediante promociones más efectivas y dirigidas.

### **1.3 Objetivos**

#### **1.3.1 Objetivo General**

Desarrollar un sistema de recomendación de productos usando algoritmos de aprendizaje automático de sugerencia para mejorar la oferta de combos y la experiencia del cliente.

#### **1.3.2 Objetivos Específicos**

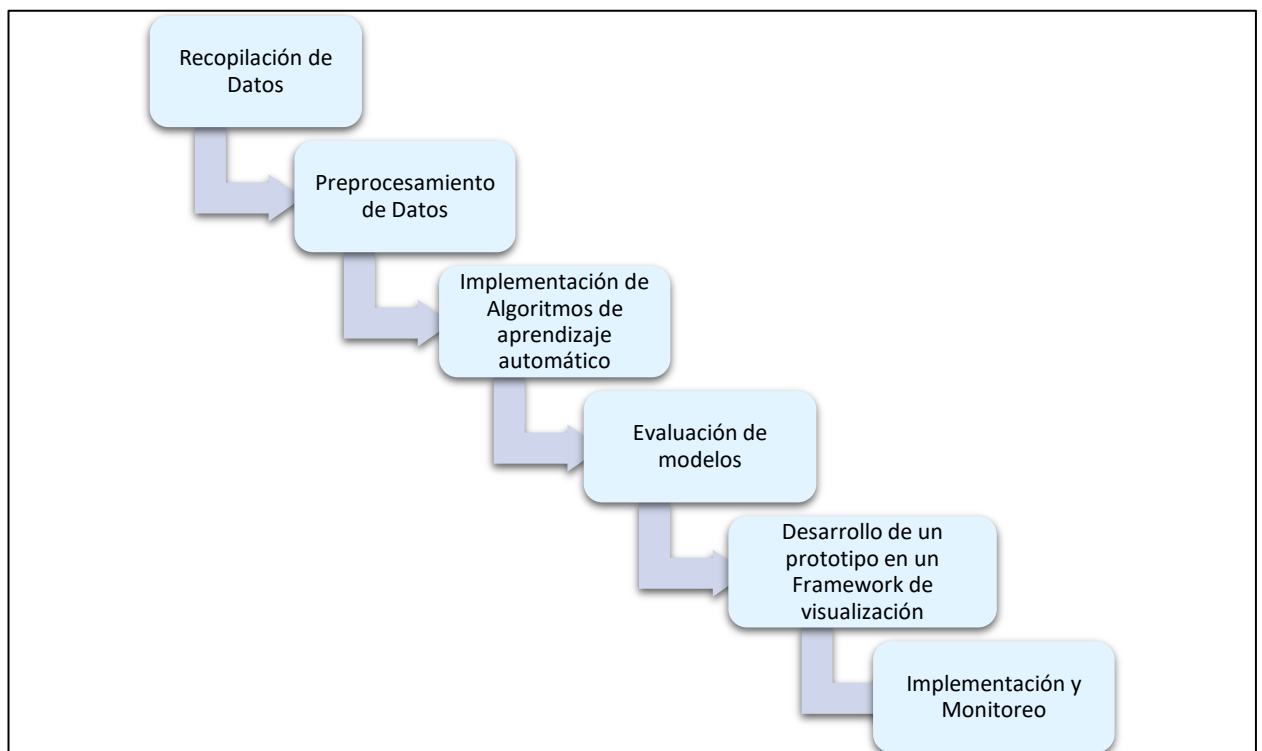
1. Implementar algoritmos de aprendizaje automático de sugerencia para identificar patrones de compra conjunta entre productos.
2. Realizar visualizaciones interactivas dinámicas empleando un framework de visualización para monitorear el comportamiento de compra y la eficacia de las recomendaciones.
3. Evaluar la eficiencia del modelo utilizando un análisis de rendimiento para mejorar la tasa de conversión y la experiencia del cliente.

## 1.4 Metodología

La metodología de este proyecto aprovecha técnicas avanzadas de análisis de datos y machine learning, incluyendo algoritmos de aprendizaje automático para la recomendación de combos y ofertas. La **Figura 1-1** ilustra las fases de desarrollo del proyecto:

**Figura 1-1**

***Metodología del proyecto.***



*Nota.* Elaboración propia.

### 1.4.1 Recopilación de Datos:

Se utilizarán datos históricos de ventas que incluyen registros de transacciones de Enero a Septiembre del 2024. Estos datos serán obtenidos de la empresa retail y abarcan las transacciones del top 25 de tiendas con mayor venta. La recopilación de datos es crucial para asegurar que el modelo tenga una base sólida de información.

#### **1.4.2 Preprocesamiento de Datos:**

Los datos serán sometidos a un proceso de limpieza y transformación para garantizar su calidad y utilidad. Este proceso incluirá la eliminación de datos duplicados, el tratamiento de valores faltantes y la normalización de datos numéricos. La preparación de los datos a través de este paso es esencial para el análisis y el desarrollo del modelo.

#### **1.4.3 Implementación de Algoritmos de aprendizaje automático:**

Se aplicarán algoritmos de aprendizaje automático como:

- Filtrado colaborativo basado en vecinos (User-Based y Item-Based) que se centra en las similitudes entre usuarios o productos para hacer recomendaciones (Pinela, 2017). A diferencia de IB, UB calcula la similitud entre productos directamente, en lugar de identificar a los K vecinos más cercanos.
- Alternating Least Squares (ALS), una técnica eficiente en sistemas de recomendación se utiliza para identificar patrones de compra combinada entre productos, especialmente cuando se trabaja con grandes volúmenes de datos. ALS se basa en la factorización de matrices, dividiendo la matriz de usuario-producto en dos matrices de menor tamaño. Esta técnica es ideal para manejar datos dispersos y mejorar la precisión de las recomendaciones al identificar patrones subyacentes en las interacciones entre usuarios y productos (Rendle, 2012).
- Neural Collaborative Filtering (NCF) es un enfoque de vanguardia en los sistemas de recomendación que aprovecha el poder de las redes neuronales para modelar las interacciones entre usuarios y productos, diferenciándose de los métodos tradicionales basados en factorización matricial, NCF permite capturar relaciones no lineales y patrones

complejos en los datos de interacción, mejorando la personalización y precisión de las recomendaciones. Su flexibilidad lo convierte en una alternativa eficiente para escenarios donde las preferencias del usuario pueden cambiar dinámicamente (Wang et al., 2019).

#### **1.4.4 Desarrollo de un prototipo en un framework de visualización:**

Desarrollaremos un prototipo de aplicación con un framework de visualización para monitorear en tiempo real el comportamiento de consumo y la eficacia de las recomendaciones, lo que permitirá una mejor toma de decisiones estratégicas en marketing.

#### **1.4.5 Evaluación de modelos:**

La evaluación de los modelos se basará en métricas de rendimiento como precisión, recall y F1-score, las cuales miden la calidad y relevancia de las recomendaciones. (Fader & Hardie, 2016).

#### **1.4.6 Implementación y Monitoreo:**

En esta etapa, se desarrollará un prototipo funcional que permita validar la viabilidad del sistema de recomendación. La implementación completa se prevé en una fase posterior, donde se integrará con los sistemas operativos de la empresa. Además, se establecerá un proceso de monitoreo continuo, para ajustar y mejorar el sistema según las necesidades cambiantes del mercado y el comportamiento de los clientes.

### **1.5 Resultados esperados**

Al finalizar este proyecto, esperamos obtener los siguientes resultados:

- Desarrollo y evaluación de algoritmos de recomendación para generar sugerencias de combos y ofertas más relevantes para los clientes, utilizando el historial de compras y preferencias pasadas. Se aplicarán distintos enfoques, y la implementación final utilizará el algoritmo que proporcione el



mejor desempeño en cuanto a precisión y relevancia de las recomendaciones.

- Un prototipo de aplicativo interactivo que proporcione insights en tiempo real sobre el comportamiento de compra en los clientes y efectividad en las recomendaciones. Esto ayudará a hacer ajustes rápidos en la estrategia de marketing.
- Mejorar la experiencia del cliente, al ofrecer promociones que se ajusten mejor a lo que el/los clientes deseen.
- Impacto económico positivo: Con las mejoras en las estrategias con las áreas de la compañía y siendo más precisos en la selección de ofertas, se tiene como objetivo incrementar las ventas y usar los recursos de manera más eficiente, lo que mejorara el rendimiento económico en la empresa.

## 1.6 Dataset

Se dispone de un dataset que contiene datos correspondientes a las ventas de productos que estuvieron en oferta en el periodo comprendido desde Enero hasta Septiembre del 2024 en una empresa retail. Donde se consideraron únicamente las transacciones del top 25 de las tiendas con mayor venta.

Referente a la disponibilidad del dataset, se tuvo que realizar una solicitud al equipo de Data Science, debido a que, esta información es privada y pertenece a la compañía. Si se requieren más datos, se debe realizar una nueva solicitud para la actualización.

Se cuenta con un solo archivo de formato csv que proporciona la siguiente información en la **Tabla 1-1**:

**Tabla 1-1**

***Descripción del dataset inicial.***

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
sucursal_id	Identificador de la sucursal o tienda	Categorico
producto_id	Identificador del producto	Categorico
fecha	Fecha de la transacción	Temporal
factura_id	Identificador de la factura	Categorico
cantidad	Cantidad vendida en unidades	Numérico
pvp	Precio de venta al público	Numérico
costo	Costo del producto	Numérico
valor_pvp	Valor total pagado	Numérico
cliente_id	Identificador del cliente	Categorico
clase_id	Identificador de la familia de productos	Categorico
subclase_id	Identificador de la categoría subfamilia	Categorico

*Nota.* Elaboración propia.

La disponibilidad del dataset es de 11 millones de registros (filas) y 10 columnas donde se realizó un ajuste correcto al tipo de dato, ya sea de tipo categorico, numerico y temporal.

En el próximo capítulo se presentarán los fundamentos teóricos que sustentan los modelos de aprendizaje automatico para la recomendación de productos. Esta revisión teorica permitirá entender las metodologías y algoritmos clave para el desarrollo del sistema.

# CAPÍTULO 2

## 2. ESTADO DEL ARTE

Este capítulo ofrece una revisión teórica de información referente a Sistemas de recomendación aplicados al sector retail, centrado en técnicas y algoritmos de aprendizaje automático. Se presentan trabajos relevantes que proporcionan un marco referencial para este proyecto, abordando metodologías de preprocesamiento y análisis de datos esenciales para la implementación del sistema de recomendación propuesto.

### 2.1 Marco referencial

A continuación, se mencionan algunos trabajos referentes al proyecto objeto de estudio:

#### 2.1.1 Servicios de Recomendación Basados en Filtrado Colaborativo Adaptativo Extendidos con KNN.

El artículo realizado por Nguyen et al. (2023) presenta un enfoque adaptativo para los Sistemas de recomendación, fundamentado en el algoritmo K-Nearest Neighbors (KNN). Esta técnica adaptativa busca superar las restricciones de los antiguos métodos de filtrado colaborativo al ajustar dinámicamente la influencia de los vecinos más cercanos, considerando su impacto en la relevancia y semejanza con los usuarios. Esto permite una personalización más precisa de las recomendaciones al incorporar un análisis más detallado de las interacciones y preferencias de los usuarios.

En relación al desempeño, el estudio muestra que el método adaptativo KNN mejora significativamente las sugerencias proporcionadas, en comparación con otros sistemas. Los resultados obtenidos destacan un aumento en la precisión y relevancia de las recomendaciones, dando una experiencia satisfactoria a los usuarios. La evaluación de la efectividad se realiza mediante métricas estándar como la precisión y el recall, así como a través de pruebas

de usuario que corroboran la mejora en la experiencia general de recomendación.

Sin embargo, el artículo también identifica varios desafíos asociados con el método propuesto. Un reto importante es la demanda de un ajuste fino de los parámetros del modelo para mantener el equilibrio entre la adaptabilidad y la complejidad computacional. Además, el algoritmo requiere una cantidad significativa de datos para entrenar y ajustar el modelo, lo cual puede ser un obstáculo en contextos con información limitada o en tiempo real.

El análisis del estudio revela que, a pesar de estos desafíos, el enfoque adaptativo basado en KNN ofrece una solución prometedora para mejorar los sistemas de recomendación. La adaptabilidad del modelo permite una personalización más precisa, lo que puede ser especialmente útil en aplicaciones que requieren una alta precisión en la recomendación. El artículo concluye que la metodología adaptativa favorece a la mejora continua de los servicios de recomendación, aunque su implementación efectiva dependerá de la capacidad para manejar las complejidades y limitaciones del entorno de datos.

### **2.1.2 Algoritmo Alternating Least Squares (ALS) para recomendaciones en tiempo real**

El trabajo de Hu et al. (2008) aborda el uso del algoritmo Alternating Least Squares (ALS) en sistemas de recomendación que manejan datos implícitos, como clics y visualizaciones de productos. Este método optimiza las recomendaciones al minimizar iterativamente los errores entre las predicciones y las interacciones observadas, resultando en recomendaciones más precisas y en tiempo real. El ALS se distingue por su habilidad para descomponer la matriz de interacciones en factores latentes, permitiendo identificar patrones ocultos en los datos de usuario.

El ALS es especialmente eficaz en contextos donde los datos explícitos, como calificaciones, son escasos. En el comercio minorista, donde las interacciones implícitas son más comunes y reflejan mejor las preferencias de los usuarios, el ALS se convierte en una herramienta esencial. Este algoritmo maneja grandes volúmenes de datos y ofrece recomendaciones rápidas, mejorando la experiencia del cliente. La integración del ALS en sistemas en tiempo real permite a los minoristas ajustar sus ofertas y promociones dinámicamente según las interacciones de los usuarios.

La implementación del ALS en plataformas de comercio electrónico permite personalizar recomendaciones a nivel individual, aumentando la retención de clientes y las tasas de conversión. Los autores concluyen que el ALS es efectivo para predecir preferencias implícitas, y su escalabilidad y eficiencia lo hacen ideal para sistemas de recomendación en tiempo real.

### **2.1.3 Algoritmo Neural Collaborative Filtering (NCF) para sistemas de recomendación**

El artículo de He et al. (2017) titulado "Neural Collaborative Filtering" presenta un enfoque innovador para sistemas de recomendación basado en el uso de redes neuronales profundas. A diferencia otras metodologías que emplean factorización matricial, el NCF integra la capacidad de modelado no lineal de las redes neuronales para capturar interacciones complejas entre usuarios y productos.

El modelo NCF propuesto en el artículo consta principalmente de dos componentes: una capa de embeddings que ubica a los usuarios y productos en un espacio denso de atributos latentes, y por otra parte, una red neuronal que aprende a predecir la probabilidad de interacción entre un usuario y un producto. Este enfoque permite al modelo capturar relaciones no lineales y mejorar la precisión de las recomendaciones.

Los resultados del estudio demuestran que el NCF supera a los métodos tradicionales de recomendación en varios escenarios, especialmente cuando se trata de datos dispersos y no lineales. El artículo muestra que el NCF permite proporcionar una mejora significativa en métricas como la precisión y el recall en relación a otras técnicas basadas en factorización matricial como ALS.

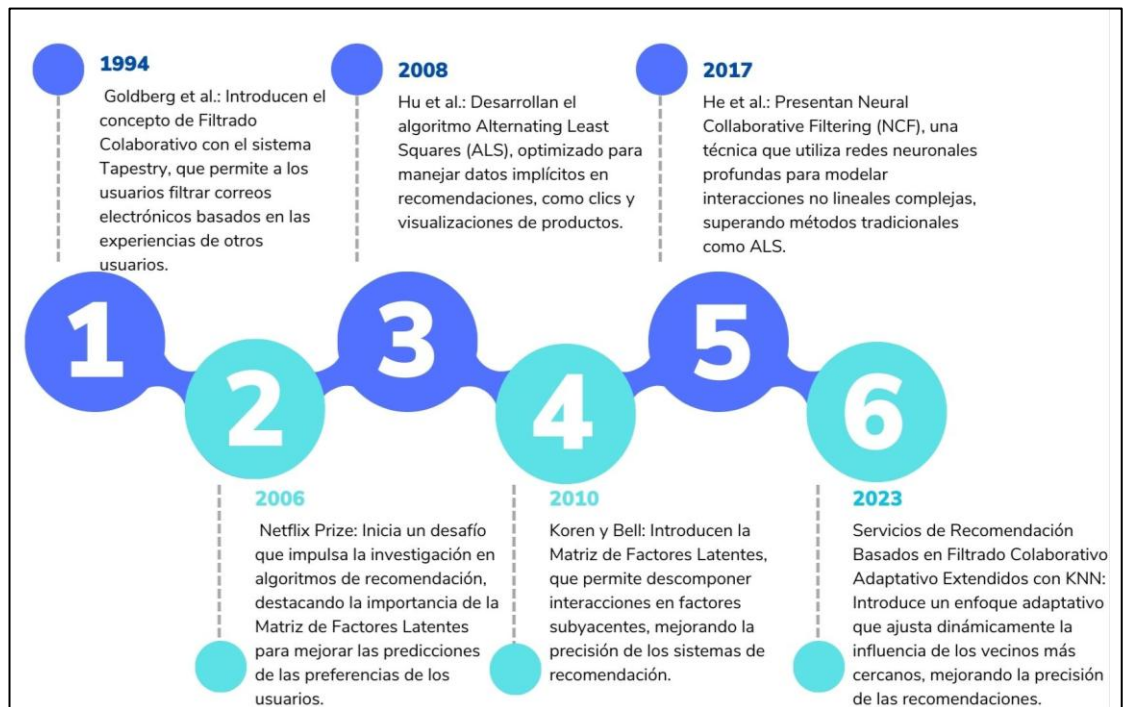
Sin embargo, los autores también destacan algunos desafíos, como el hecho de necesitar una gran cantidad de información histórica para poder entrenar eficientemente el modelo y la complejidad computacional asociada con el entrenamiento de redes neuronales profundas. A pesar de estas limitaciones, el NCF representa un avance importante en la personalización de recomendaciones, ofreciendo un marco flexible que puede adaptarse a diferentes tipos de datos y aplicaciones.

En conclusión, el artículo sugiere que la integración de redes neuronales en sistemas de recomendación abre nuevas posibilidades para mejorar la eficacia de las sugerencias, aunque su implementación requiere consideraciones cuidadosas respecto a la infraestructura y los recursos computacionales disponibles.

La siguiente línea de tiempo destaca las contribuciones clave en el desarrollo de algoritmos de recomendación, proporcionando una visión clara de cómo ha evolucionado el estado del arte en este campo.

**Figura 2-1**

### ***Evolución Histórica de los Algoritmos de Recomendación***



*Nota.* Elaboración propia.

En las siguientes secciones, se abordarán los conceptos clave de los sistemas y algoritmos de recomendación, herramientas esenciales para personalizar y mejorar la experiencia del usuario, empleadas para recomendar productos, servicios o contenido en diversos contextos, incluyendo el comercio minorista y otras aplicaciones.

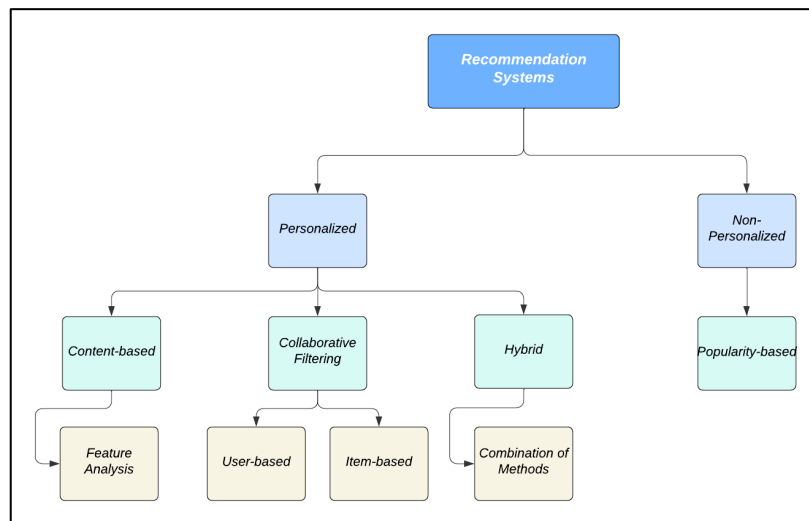
## **2.2 Sistemas de recomendación**

Los sistemas de recomendación constituyen soluciones útiles para sugerir productos a los usuarios, basándose en diversos criterios. Estos sistemas tienen un rol muy importante en mejorar la satisfacción de los usuarios y aumentar la eficiencia de las recomendaciones en diversos contextos, como su uso extendido en plataformas digitales hoy en día, mediante la entrega de recomendaciones personalizadas (Pazzani & Billsus, 2007).

Existen varias categorías de sistemas de recomendación, cada una con características y enfoques únicos:

**Figura 2-2**

***Clasificación de Sistemas de Recomendación.***



*Nota.* Elaboración propia.

## **2.2.1 Filtrado Colaborativo**

“El filtrado colaborativo establece que si dos usuarios han tenido elecciones similares anteriormente, tenderán a hacerlo de nuevo más adelante” (Schafer et al., 2007).

El Filtrado colaborativo se clasifica en dos categorías principales: basado en usuarios (User-Based) y basado en ítems (Item-Based).

### **2.2.1.1 User-base**

“El filtrado colaborativo basado en usuarios estima los intereses de un usuario a partir de los intereses de otros usuarios con elecciones similares” (Ricci et al., 2015). Este método se fundamenta en la hipótesis que usuarios con gustos afines en el pasado tendrán gustos afines en el futuro.



Según Goldberg et al. (1992) y Herlocker et al. (1999), el pronóstico de la valoración de un usuario para un ítem se calcula considerando la media de calificaciones y la similitud entre usuarios. La ecuación correspondiente es la siguiente:

**Modelo Matemático:**

$$r_{u,i} = u + b_u + b_i + \sum_{v \in N(u)} w_{uv} * (r_{v,i} - u - b_v) \quad (1)$$

Donde:

- $r_{u,i}$  es la predicción de la valoración que el usuario  $u$  concede al ítem  $i$ .
- $u$  es la media global de todas las calificaciones.
- $b_u$  y  $b_i$  son los sesgos del usuario y del ítem respectivamente.
- $N(u)$  es el conjunto de usuarios vecinos de  $u$ .
- $w_{uv}$  es la similitud entre los usuarios  $u$  y  $v$ .

### **2.2.1.2 Item-based**

El filtrado colaborativo basado en ítems (item-based collaborative filtering) se enfoca en la similitud entre los ítems en lugar de los usuarios. En contraste con el enfoque basado en usuarios, “Los sistemas basados en productos sugieren productos parecidos a los que han recibido una evaluación favorable por parte del usuario.” (Adomavicius & Tuzhilin, 2005). Este método es eficaz para sistemas con grandes conjuntos de datos porque la similitud entre ítems tiende a ser más estable que la similitud entre usuarios.

Según Sarwar et al. (2001), el pronóstico de la puntuación de un usuario para un ítem se calcula considerando la media de calificaciones y la similitud entre productos. La ecuación que modela este proceso es la siguiente:

### Modelo Matemático:

$$r_{u,i} = u + b_u + b_i + \sum_{j \in N(i)} w_{ij} * (r_{v,j} - u - b_j) \quad (2)$$

Donde:

- $r_{u,i}$  es la predicción de la valoración que el usuario  $u$  otorga al ítem  $i$ .
- $u$  constituye el promedio general de todas las calificaciones.
- $b_u$  y  $b_i$  son los sesgos del usuario y del ítem respectivamente.
- $N(i)$  es el conjunto de ítems vecinos de  $i$ .
- $w_{ij}$  es la similitud entre los ítems  $i$  y  $j$ .

#### 2.2.2 Basados en Contenido

Los sistemas de recomendación basados en contenido emplean los atributos de los productos junto con las distinciones explícitas o implícitas de los usuarios para generar sugerencias. Este enfoque parte del principio de que los ítems recomendados comparten similitudes con aquellos en los que el usuario ha mostrado interés previamente. (Pazzani & Billsus, 2007).

“Enfoque que asegura que las recomendaciones sean relevantes incluso cuando no hay datos suficientes sobre otros usuarios” (Lops et al., 2011).

Este tipo de sistemas sugieren ítems mediante un análisis comparativo entre las propiedades particulares de los productos y los perfiles de usuarios. Estos perfiles se construye a partir de valoraciones previas del usuario.

### 2.2.2.1 Análisis de Características

#### Modelo Matemático

- **Vector de Características del ítem:** Cada ítem  $i$  está modelado mediante un vector de atributos  $x_i$ .
- **Perfil del Usuario:** El usuario  $u_k$  tiene un perfil generado a partir de una combinación ponderada de los vectores de características de los ítems que ha valorado, donde las calificaciones determinan su relevancia.

$$u_k = \frac{\sum_{i \in I_k} r_{ki} x_i}{\sum_{i \in I_k} r_{ki}} \quad (3)$$

Donde  $I_k$  corresponde a la agrupación de ítems que el usuario  $k$  ha valorado y  $r_{ki}$  es la puntuación que el usuario  $k$  asignó al ítem  $i$ .

- **Similitud del Coseno:** La similitud entre el perfil del usuario y un nuevo ítem se calcula mediante la similitud del coseno.

$$\text{sim}(u_k, x_j) = \frac{u_k x_j}{|u_k| |x_j|} \quad (4)$$

- **Puntuación Predicha:** La puntuación predicha para el ítem  $j$  para el usuario  $k$  se puede expresar como:

$$\hat{r}_{kj} = \text{sim}(u_k, x_j) \quad (5)$$

## Parámetros e Hiperparámetros

### Parámetros:

- $x_i$ : Vectores de características de los ítems.
- $u_k$ : Perfiles de los usuarios.
- $r_{ki}$ : Calificaciones dadas por los usuarios a los ítems.

### Hiperparámetros:

- Método de extracción de características (e.g., TF-IDF para texto, embeddings para imágenes).
- Métrica de similitud (e.g., coseno, euclidiana).
- Peso de las características (en caso de usar ponderaciones).

### Flujo de Trabajo

- **Recopilación de Datos:** Obtener datos de los ítems y la manera en que los usuarios han interactuado con ellos.
- **Construcción del Perfil del Usuario:** Crear perfiles de usuario a partir de los ítems que han evaluado, ponderados por las calificaciones.
- **Cálculo de Similitud:** Calcular la afinidad entre las preferencias del usuario y los nuevos ítems.
- **Generación de Recomendaciones:** Ordenar los ítems en función de la similitud y recomendar los más similares.

### 2.2.3 Híbridos

Los sistemas híbridos (hybrid recommendation systems) combinan múltiples enfoques de recomendación con el objetivo de incrementar la fiabilidad de las recomendaciones. Estos sistemas pueden combinar filtrado colaborativo, métodos basados en contenido y otros enfoques.

### 2.2.3.1 Combinación de Métodos

#### 1. Método de Ponderación (Weighted Hybrid)

Combina las puntuaciones de diferentes técnicas asignando un peso a cada una.

**Modelo Matemático:**

$$r_{u,i} = \alpha * r_{u,i}^{colab} + \beta * r_{u,i}^{contenido} \quad (6)$$

Donde:

- $r_{u,i}$  es la puntuación predicha para el usuario  $u$  y el ítem  $i$ .
- $r_{u,i}^{colab}$  es la predicción del filtrado colaborativo.
- $r_{u,i}^{contenido}$  es la predicción del sistema basado en contenido.
- $\alpha$  y  $\beta$  son los pesos asignados a cada método (Burke, 2002) y (Adomavicius & Tuzhilin, 2005).

#### 2. Método de Combinación de Características (Feature Combination)

El método de combinación de características permite integrar datos de los usuarios y de los ítems, aprovechando múltiples enfoques en un único modelo. Según Adomavicius & Tuzhilin (2005) este método incorpora simultáneamente características extraídas de sistemas basados en contenido y de filtrado colaborativo, mejorando así la habilidad del sistema para ofrecer sugerencias más acertadas.

**Modelo Matemático:**

$$z_{u,i} = [x_u^{CB}, y_i^{CB}, x_u^{CF}, y_i^{CF}] \quad (7)$$

Donde:

- $x_u^{CB}$  y  $y_i^{CB}$  son las características de usuarios e ítems basadas en contenido.
- $x_u^{CF}$  y  $y_i^{CF}$  son las características basadas en filtración colaborativa.

### 3. Método de Filtrado Mixto (Mixed Hybrid)

El método de filtrado mixto combina múltiples técnicas de recomendación para mejorar la precisión y diversidad de las sugerencias generadas. Según Burke (2002) este enfoque permite generar recomendaciones por separado utilizando diferentes métodos y luego combinar los resultados mediante una función de mezcla.

**Modelo Matemático:**

$$\widehat{R}_u = MIX ( \widehat{R}_u^{CF}, \widehat{R}_u^{CB} ) \quad (8)$$

Donde:

- Mix es una función de mezcla que combina las recomendaciones de filtración colaborativa  $\widehat{R}_u^{CF}$  y basadas en contenido  $\widehat{R}_u^{CB}$ .

### 4. Método de Métodos Combinados (Meta-Level Hybrid)

El método de nivel meta utiliza una técnica de recomendación para generar características que luego son empleadas como entrada en otro modelo de recomendación. Burke (2002) señala que este enfoque permite combinar la fortaleza de diferentes métodos, optimizando la calidad de las predicciones al utilizar información generada por filtrado colaborativo como insumo para otro modelo.

**Modelo Matemático:**

$$z_{u,i} = MODEL_{CF}(x_u, y_i) \quad (9)$$

Donde:

- $MODEL_{CF}$  es un modelo de filtración colaborativa que genera características utilizadas por otro modelo de recomendación.

## 2.2.4 Otros Sistemas

### 2.2.4.1 Popularity-based

Los sistemas basados en popularidad (popularity-based recommendation systems) recomiendan los ítems más populares independientemente del perfil del usuario. Son útiles en situaciones donde no hay suficientes datos sobre el usuario.

#### 1. Frecuencia de Interacción

##### Modelo Matemático:

El modelo más básico calcula la popularidad de un ítem basándose en la frecuencia con la que los usuarios han interactuado con él. Esta interacción puede ser en forma de vistas, compras, descargas, etc.

$$\text{pop}(i) = \sum_{u \in U} I_{ui} \quad (10)$$

Donde:

- $\text{pop}(i)$  es la popularidad del ítem  $i$ .
- $I_{ui}$  es una variable indicadora que es 1 si el usuario  $u$  ha interactuado con el ítem  $i$  y 0 en ausencia de interacción.
- $U$  es la agrupación de la totalidad de usuarios.

#### 2. Promedio de Calificaciones

Otro enfoque mide la popularidad basándose en el promedio de las calificaciones que el ítem ha recibido.

$$\text{pop}(i) = \frac{\sum_{u \in U} r_{ui}}{|U_i|} \quad (11)$$

Donde:

- $r_{ui}$  indica la puntuación que el usuario  $u$  otorgó al ítem  $i$ .
- $U_i$  es el conjunto que agrupa a los usuarios que han valorado al ítem  $i$ .

### 3. Combinación de Frecuencia y Calificaciones

Una combinación de ambos enfoques puede proporcionar un balance entre la cantidad de interacciones y la calidad percibida de los ítems.

$$\text{pop}(i) = \alpha * \left( \sum_{u \in U} I_{ui} \right) + \beta * \left( \frac{\sum_{u \in U} r_{ui}}{|U_i|} \right) \quad (12)$$

Donde:

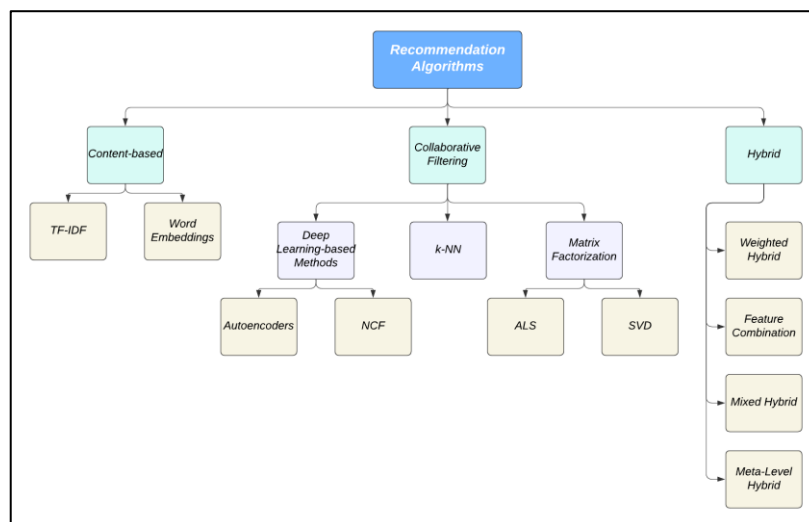
- $\alpha$  y  $\beta$  son pesos que controlan la influencia de la frecuencia de interacciones y el promedio de calificaciones, respectivamente.

### 2.3 Algoritmos de recomendación

Por otro lado, los algoritmos de recomendación se basan en técnicas matemáticas y computacionales para procesar datos y generar sugerencias. Estos algoritmos son esenciales para los sistemas de recomendación, ya que determinan la calidad y relevancia de las recomendaciones proporcionadas a los usuarios (Kouzes et al., 2009). Los algoritmos principales incluyen:

Figura 2-3

#### ***Clasificación de Algoritmos de Recomendación.***



Nota. Elaboración propia.



En este proyecto, se han seleccionado tres algoritmos fundamentales para la implementación del sistema de recomendación, considerando su capacidad para manejar eficientemente datos transaccionales y generar recomendaciones altamente personalizadas. La elección de estos algoritmos no solo responde a su reconocida eficacia en el ámbito de la recomendación, sino también a su probado rendimiento en contextos que comparten características clave con el presente proyecto. A continuación, se presentan los algoritmos escogidos para su implementación en este estudio:

### 2.3.1 K-NN

#### 1. Descripción

El algoritmo k-Nearest Neighbors (k-NN) se utiliza para identificar ítems similares a aquellos con los que un usuario ha interactuado previamente. Este método de aprendizaje supervisado es ampliamente utilizado tanto para tareas de clasificación como de regresión. La premisa básica de k-NN es que ítems similares tendrán calificaciones similares. Para determinar la similitud entre ítems, el algoritmo considera la proximidad en un espacio de características utilizando métricas de distancia como la euclidiana, la similitud del coseno, entre otras (Aggarwal, 2016).

La similitud entre dos ítems  $i$  y  $j$  se puede calcular manejando diversas métricas. Las métricas más comunes es la similitud del coseno, que se define como:

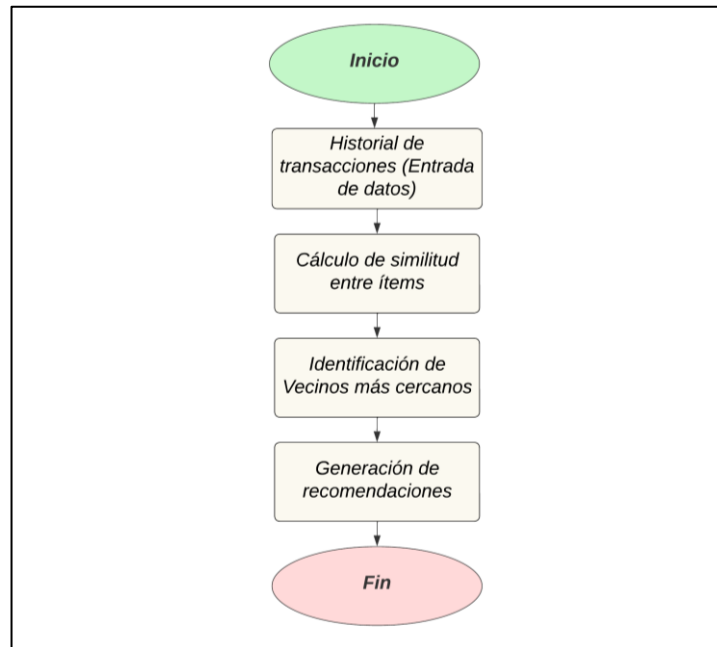
$$sim(i, j) = \frac{\sum_{u \in U} r_{u,i} \cdot r_{u,j}}{\sqrt{\sum_{u \in U} r_{u,i}^2} \cdot \sqrt{\sum_{u \in U} r_{u,j}^2}} \quad (13)$$

donde  $r_{u,i}$  es la valoración del usuario  $u$  para el ítem  $i$  y  $U$  es el conjunto de todos los usuarios (Koren et al., 2009).

## 2. Diagrama de funcionamiento

Figura 2-4

**Flujograma del Algoritmo *k*-Nearest Neighbors (*k*-NN).**



*Nota.* Elaboración propia.

## 3. Flujo de trabajo

Este flujo de trabajo sigue la metodología descrita en Aggarwal (2016; Koren et al., 2009).

- **Inicio**

El proceso se inicia con la definición del objetivo: generar recomendaciones personalizadas basadas en el historial de transacciones de los usuarios. Este objetivo establece el marco para el procesamiento de datos y la aplicación del algoritmo k-NN.

- **Historial de Transacciones (Entrada de Datos)**

Se recopilan y organizan los datos históricos de transacciones en una matriz usuario-producto. Esta matriz, que representa las interacciones entre

usuarios y productos, es fundamental para el análisis de similitudes entre ítems.

- ***Cálculo de Similitud entre Ítems***

El sistema calcula la similitud entre productos utilizando métricas como la similitud del coseno. Este cálculo determina qué productos son más similares entre sí, basándose en las interacciones registradas en la matriz usuario-producto.

- ***Identificación de Vecinos Más Cercanos***

Una vez calculadas las similitudes, el algoritmo identifica los  $k$  ítems más cercanos (vecinos) para cada producto de interés. Estos vecinos son fundamentales, ya que influyen directamente en las recomendaciones que se generarán.

- ***Generación de Recomendaciones***

El sistema genera recomendaciones para los usuarios basándose en los productos identificados como vecinos más cercanos. Estos ítems son aquellos que, según el análisis, tienen mayor probabilidad de interesar al usuario.

- ***Recomendaciones Personalizadas (Salida)***

Finalmente, se presentan las recomendaciones personalizadas al usuario. Estas recomendaciones, derivadas del análisis previo, buscan mejorar la experiencia de compra al sugerir productos relevantes.

#### **4. Parámetros e Hiperparámetros**

- ***$k$ : Número de vecinos más cercanos a considerar.***

Este parámetro define cuántos ítems cercanos se tomarán en cuenta para hacer una recomendación. El valor de  $k$  determina cuántos vecinos se consideran para las recomendaciones, afectando el equilibrio entre precisión y generalización. Un valor bajo puede ser sensible al ruido, mientras que obtener un valor alto puede diluir la relevancia. En la práctica,

el valor óptimo de  $k$  se selecciona a través de técnicas de validación cruzada (Aggarwal, 2016).

- **Distancia: Tipo de métrica de distancia (euclidiana, coseno, Manhattan, etc.)**

La métrica de distancia utilizada influye significativamente en la calidad de las recomendaciones. La métrica del coseno es comúnmente utilizada en sistemas de recomendación porque mide la orientación entre vectores, lo que es particularmente útil en el caso de datos dispersos. Además de las métricas mencionadas, existen otras opciones como la distancia euclidiana o Manhattan, que pueden ser más adecuadas según los datos. La elección de la métrica correcta puede mejorar la precisión del sistema (Koren et al., 2009).

## **5. Medidas de Desempeño**

- Precisión
- Recall
- F1 Score
- RMSE (Root Mean Square Error)

### **2.3.2 ALS**

#### **1. Descripción**

Alternating Least Squares (ALS) es un algoritmo de factorización de matrices utilizado comúnmente en sistemas de recomendación, especialmente efectivo para manejar datos implícitos, como clics, visualizaciones y compras. ALS descompone la matriz usuario-ítem en dos matrices más pequeñas para minimizar errores de predicción. Este método alterna entre la actualización de las características del usuario y las características del ítem, utilizando la técnica de mínimos cuadrados para encontrar la mejor aproximación (Hu et al., 2008).

## 2. Modelo Matemático

La matriz de interacciones  $R$  se factoriza en dos matrices  $P$  y  $Q$  tal que:

$$R \approx PQ^T \quad (14)$$

donde  $P \in \mathbb{R}^{m \times k}$  y  $Q \in \mathbb{R}^{n \times k}$ , con  $m$  siendo el número de usuarios,  $n$  el número de ítems, y  $k$  el número de factores latentes. El cálculo de la predicción de la calificación del usuario  $u$  para el ítem  $i$  se realiza mediante la siguiente fórmula:

$$\hat{r}_{ui} = p_u^T q_i \quad (15)$$

donde  $p_u$  es el vector de características que describe al usuario  $u$  y  $q_i$  es el vector de características que describe al ítem  $i$ .

Para ajustar las matrices  $P$  y  $Q$ , se minimiza la función de error regularizada:

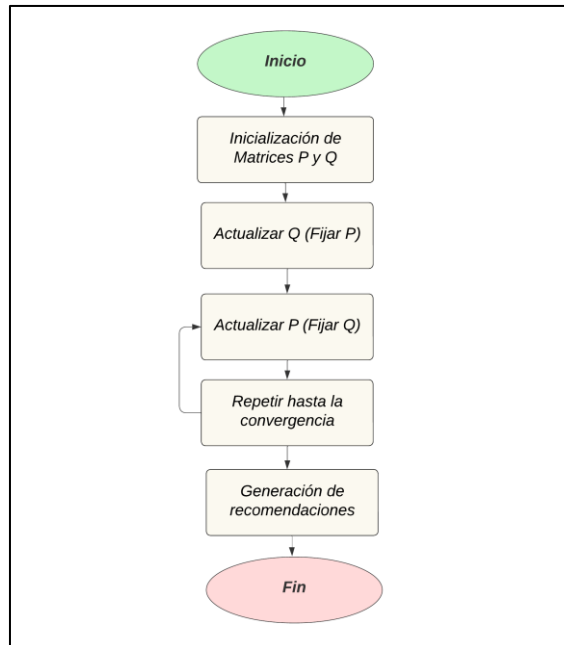
$$\min_{P, Q} \sum_{(u, i) \in \mathbb{R}} (r_{ui} - p_u^T q_i)^2 + \lambda (\|p_u\|^2 + \|q_i\|^2) \quad (16)$$

donde  $\lambda$  es el parámetro de regularización que penaliza los valores grandes de los factores para evitar el sobreajuste.

### 3. Diagrama de Funcionamiento

Figura 2-5

*Flujograma del Algoritmo Alternating Least Squares (ALS).*



*Nota.* Elaboración propia.

### 4. Parámetros e Hiperparámetros

- **Factores Latentes ( $k$ ):**

Este parámetro define el número de dimensiones en el espacio de factores latentes. Los factores latentes representan características ocultas que los usuarios e ítems comparten y que no están explícitamente disponibles en los datos originales. Un valor de  $k$  mayor permite al modelo capturar patrones más complejos en los datos, pero también incrementa el riesgo de sobreajuste y el costo computacional (Hu et al., 2008).

El valor de  $k$  debe seleccionarse con cuidado, generalmente a través de validación cruzada, para equilibrar la capacidad del modelo y la prevención de sobreajuste.

- **Número de Iteraciones (num\_iters):**

Este parámetro define cuántas veces se repetirá el proceso de actualización de las matrices  $P$  y  $Q$ . El número de iteraciones influye tanto en la precisión de las predicciones del modelo como en el tiempo de cómputo. Un mayor número de iteraciones puede mejorar la precisión, pero también aumenta el tiempo necesario para completar el entrenamiento. Es crucial encontrar un equilibrio entre ambos factores (Koren et al., 2009).

En la práctica, el número de iteraciones se ajusta hasta que la función de error converge, lo que significa que las actualizaciones posteriores no mejoran significativamente el rendimiento del modelo.

- **Regularización ( $\lambda$ ):**

Este parámetro controla la magnitud de la penalización aplicada a los valores grandes de los factores latentes en las matrices  $P$  y  $Q$ . Es crucial aplicar técnicas de regularización para prevenir el sobreajuste, que ocurre cuando el modelo se ajusta demasiado a los datos de entrenamiento y no puede generalizar correctamente a datos nuevos. (Hu et al., 2008).

El valor de  $\lambda$  determina el equilibrio entre la complejidad del modelo y su capacidad para generalizar. Un valor alto de  $\lambda$  reduce la complejidad y mejora la generalización, mientras que un valor bajo puede resultar en un modelo demasiado complejo que se sobreajusta a los datos de entrenamiento.

## 5. Flujo de Trabajo

- ***Inicio***

El proceso comienza con la preparación para la descomposición de la matriz de interacciones usuario-ítem mediante el algoritmo ALS. Este paso establece el contexto para el procesamiento subsiguiente.

- ***Inicialización de Matrices  $P$  y  $Q$***

Se inicializan las matrices  $P$  y  $Q$  con valores aleatorios. Estas matrices representan las características latentes de usuarios e ítems, y su

inicialización aleatoria es el punto de partida para el ajuste iterativo posterior.

- **Actualización Alternada de  $Q$  (Fijar  $P$ )**

Con  $P$  fijado, se actualiza la matriz  $Q$  resolviendo un problema de mínimos cuadrados. Este paso ajusta  $Q$  para minimizar el error de predicción dado  $P$ .

- **Actualización Alternada de  $P$  (Fijar  $Q$ )**

A continuación, se fija  $Q$  y se actualiza  $P$  utilizando la misma técnica de mínimos cuadrados. Este proceso iterativo asegura que ambas matrices converjan hacia una solución óptima.

- **Repetición hasta la Convergencia**

Los pasos de actualización alternada se repiten hasta que la función de error converja, lo que indica que las matrices  $P$  y  $Q$  han alcanzado una solución estable y suficientemente precisa.

- **Generación de Recomendaciones**

Una vez que el algoritmo ha convergido, se utilizan las matrices  $P$  y  $Q$  para predecir las calificaciones de ítems que los usuarios no han interactuado previamente. Estas predicciones se emplean para generar recomendaciones personalizadas para cada usuario.

- **Fin**

El proceso concluye con la presentación de las recomendaciones generadas, finalizando así el ciclo de ALS en el sistema de recomendación.

## 6. Medidas de Desempeño

- RMSE (Root Mean Square Error)
- MAE (Mean Absolute Error)



### 2.3.3 NCF

#### 1. Descripción

Gracias a su capacidad para modelar relaciones no lineales complejas, Neural Collaborative Filtering (NCF) se ha convertido en un enfoque avanzado en los sistemas de recomendación. A diferencia de los métodos tradicionales de filtrado colaborativo, que se basan en técnicas lineales, NCF utiliza redes neuronales para capturar patrones más sofisticados en las interacciones entre usuarios e ítems. Este método es especialmente útil en escenarios donde las interacciones usuario-ítem son muy diversas y no lineales (He et al., 2017).

#### 2. Modelo Matemático:

El modelo NCF consiste en dos componentes principales:

- **Modelo Generalizado:** Una red neuronal que combina las características del usuario y del ítem.
- **Modelo Lineal:** Similar a la factorización de matrices tradicional.

La predicción de la calificación (usuario  $u$ , ítem  $i$ ) se calcula como:

$$\hat{r}_{ui} = MLP(P_u, Q_i) \quad (17)$$

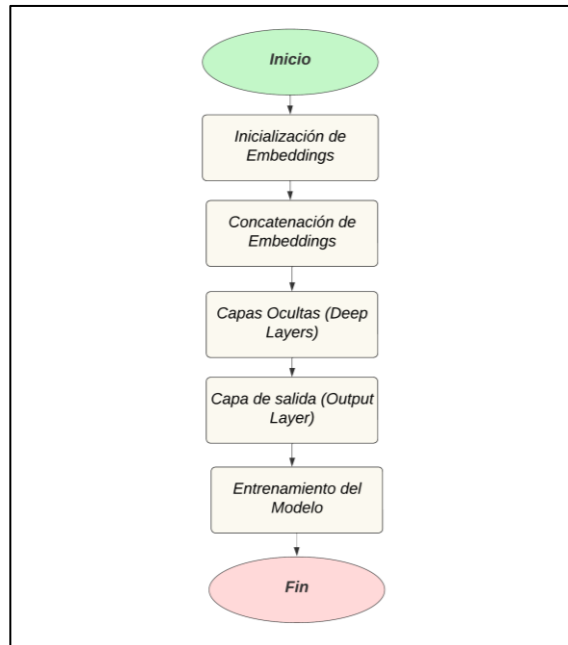
donde:

- $MLP$  es una red neuronal multicapa.
- $P_u$  y  $Q_i$  son representaciones vectoriales que capturan las características latentes del usuario  $u$  y del ítem  $i$ .

### 3. Diagrama de Funcionamiento

Figura 2-6

*Flujograma del Algoritmo Neural Collaborative Filtering (NCF).*



*Nota.* Elaboración propia.

### 4. Parámetros e Hiperparámetros:

- **Número de Embeddings (k):**

Define el tamaño de los embeddings para usuarios e ítems. Un tamaño mayor de  $k$  puede capturar más características latentes, pero también aumenta la complejidad del modelo.

Se selecciona generalmente a través de validación cruzada para encontrar el tamaño óptimo que equilibre capacidad de modelado y eficiencia.

- **Número de Capas y Neuronas por Capa:**

La arquitectura de la red neuronal, definida por su profundidad y el número de neuronas en cada capa, influye en la capacidad del modelo para aprender patrones complejos. Sin embargo, una mayor complejidad implica un mayor riesgo de sobreajuste y un tiempo de entrenamiento más prolongado.

Este parámetro se ajusta basándose en experimentación y validación, asegurando que el modelo no sea demasiado complejo para los datos disponibles.

- **Tasa de Aprendizaje (Learning Rate):**

La tasa de aprendizaje controla la velocidad de ajuste de los pesos. Una tasa alta puede acelerar la convergencia, pero también puede llevar a mínimos locales. Una tasa baja asegura un ajuste más fino pero puede hacer el entrenamiento muy lento.

Ajustar la tasa de aprendizaje es crucial para encontrar un equilibrio entre velocidad de convergencia y precisión del modelo.

## **5. Flujo de Trabajo**

- ***Inicio***

El proceso comienza con la preparación para el entrenamiento de un modelo NCF, que combina técnicas de filtrado colaborativo con la capacidad de las redes neuronales para capturar relaciones no lineales.

### **2.3.3.1 Inicialización de Embeddings**

Se inicializan los embeddings de usuario e ítem, vectores que representan las interacciones en un espacio de características reducido.

- ***Concatenación de Embeddings***

Los embeddings de usuario e ítem se concatenan para formar una entrada combinada que será procesada por la red neuronal. Esta combinación toma la relación entre el usuario-ítem a nivel de características latentes.

### **2.3.3.2 Capas Ocultas (Deep Layers)**

La red neuronal, a través de sus diversas capas ocultas, procesa la entrada concatenada, lo que le permite aprender representaciones complejas de las interacciones entre usuarios e ítems, lo que a su vez mejora la precisión de sus predicciones.

#### **2.3.3.3 Capa de Salida (Output Layer)**

La última capa de la red neuronal produce una estimación de la interacción usuario-ítem, que puede entenderse como la probabilidad de que el usuario muestre interés por el ítem en cuestión.

- **Entrenamiento del Modelo**

El modelo se entrena ajustando los pesos a través del backpropagation, minimizando la función de pérdida mediante optimización, generalmente usando un optimizador como Adam.

#### **2.3.3.4 Generación de Recomendaciones**

Una vez entrenado, el modelo se utiliza para predecir las interacciones entre usuarios e ítems que aún no han sido observadas, generando recomendaciones personalizadas.

- **Fin**

El proceso concluye con la generación y presentación de las recomendaciones al usuario.

### **7. Medidas de Desempeño**

- AUC (Área Bajo la Curva)
- Precision@k

## **2.4 Comparación Teórica de los Algoritmos de Recomendación**

Para seleccionar y justificar los algoritmos de recomendación implementados en este proyecto, es esencial realizar una comparación teórica de sus características fundamentales. La siguiente tabla resume los aspectos clave de los algoritmos seleccionados: k-NN, ALS, y NCF. Esta comparación considera criterios como el tipo de algoritmo, la escalabilidad, la flexibilidad, y la complejidad computacional, proporcionando una visión detallada de las ventajas y desventajas de cada enfoque.

**Tabla 2-1**

***Matriz comparativa de Algoritmos de Recomendación.***

<b>Criterio</b>	<b>k-NN (k-Nearest Neighbors)</b>	<b>ALS (Alternating Least Squares)</b>	<b>NCF (Neural Collaborative Filtering)</b>
Tipo de Algoritmo	Filtrado Colaborativo	Factorización de Matrices	Filtrado Colaborativo con Deep Learning
Modelo de Similitud	Vecinos más cercanos	Latente	Redes Neuronales
Manejo de datos dispersos	Moderado	Excelente	Excelente
Escalabilidad	Baja a moderada	Alta	Alta
Flexibilidad	Alta	Moderada	Alta
Capacidad de generalización	Limitada	Moderada	Alta
Complejidad computacional	Baja a Moderada	Alta	Alta
Ventajas	Simplicidad, fácil de interpretar	Maneja bien datos dispersos	Captura relaciones no lineales complejas
Desventajas	Escalabilidad, sobreajuste	Complejidad computacional	Requiere mayor poder computacional y datos.

*Nota.* \*Fuente: Información adaptada de He et al., 2017; Koren et al., 2009; Rendle, 2012.

## **2.5 Relación entre sistemas y algoritmos de recomendación**

La diferencia fundamental entre los sistemas y los algoritmos de recomendación radica en su propósito y función. A diferencia de los sistemas de recomendación, que actúan como la estructura que organiza y muestra las recomendaciones a los usuarios, los algoritmos de recomendación son las herramientas que permiten a estos sistemas analizar datos y crear sugerencias personalizadas (Aggarwal,

2016). Es decir, los algoritmos son los métodos que operan dentro de los sistemas para realizar las tareas de recomendación.

Esta relación es sinérgica: los sistemas de recomendación necesitan de los algoritmos para funcionar eficazmente, y los algoritmos, a su vez, se aplican en el contexto de un sistema para tener un impacto práctico y real en la experiencia del usuario (Adomavicius & Tuzhilin, 2005). Es esencial comprender ambos componentes y su relación para diseñar soluciones de recomendación efectivas.

## **2.6 Medidas de desempeño**

A continuación, las siguientes métricas son las más comunes para evaluar sistemas de recomendación y sirven como guía para entender las diferentes maneras de medir su eficacia:

### **2.6.1 Precisión (Precision)**

La precisión mide qué proporción de los elementos recomendados son realmente relevantes. Se calcula de la siguiente manera:

$$\text{Precisión} = \frac{\text{Elementos relevantes recomendados}}{\text{Total de elementos recomendados}} \quad (18)$$

Esta métrica es crucial en los sistemas de recomendación, ya que refleja la exactitud de las sugerencias ofrecidas al usuario (Herlocker et al., 2004).

### **2.6.2 Recall**

El recall representa la proporción de elementos relevantes que el modelo logró recomendar correctamente, en comparación con el total de elementos relevantes. La siguiente fórmula muestra cómo se calcula:

$$Recall = \frac{Elementos\ relevantes\ recomendados}{Total\ de\ elementos\ relevantes} \quad (19)$$

Esta métrica es especialmente importante cuando es necesario recuperar la mayor cantidad posible de elementos relevantes (Fader & Hardie, 2016).

### 2.6.3 F1 Score

El F1 Score es una métrica útil para evaluar modelos de clasificación, ya que combina la precisión y el recall en una sola medida. Su cálculo se realiza mediante la siguiente fórmula:

$$F1\ Score = 2 \times \frac{Precisión \times Recall}{Precisión + Recall} \quad (20)$$

Es útil en escenarios donde es crucial mantener un equilibrio entre precisión y recall (Herlocker et al., 2004).

### 2.6.4 RMSE (Root Mean Square Error)

El RMSE es una métrica que evalúa la precisión de las predicciones del modelo al calcular la diferencia entre los valores predichos y los valores reales observados, según la siguiente fórmula:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (Valor\ real - Valor\ predicho)^2} \quad (21)$$

Es una métrica de error comúnmente utilizada en modelos de recomendación, ya que penaliza los grandes errores más que los pequeños (Koren et al., 2009).

### 2.6.5 MAE (Mean Absolute Error)

El MAE calcula el valor medio de los errores absolutos, donde los errores se definen como la diferencia entre las predicciones y los valores reales. Se define como

$$MAE = \frac{1}{N} \sum_{i=1}^N |Valor\ real - Valor\ predicho| \quad (22)$$

A diferencia del RMSE, el MAE no penaliza tanto los grandes errores, siendo útil para entender el error promedio de las predicciones (Fader & Hardie, 2016).

### 2.6.6 AUC (Área Bajo la Curva)

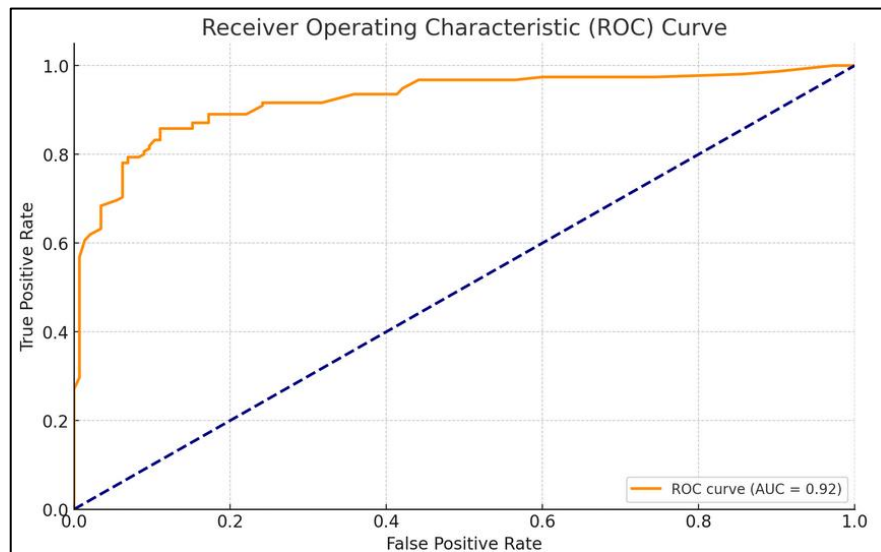
El AUC, o área bajo la curva ROC, es una medida clave para determinar la capacidad de un modelo de clasificación binaria para distinguir entre clases. En sistemas de recomendación, esta métrica proporciona información valiosa sobre la precisión del modelo en un rango completo de umbrales (Receiver Operating Characteristic) (Hanley & McNeil, 1982).

Para una mejor comprensión de la curva ROC y su aplicación en la evaluación de modelos de clasificación, se presentará un ejemplo ilustrativo. Este ejemplo permitirá visualizar cómo se relacionan las tasas de verdaderos y falsos positivos en la evaluación del rendimiento del modelo.



**Figura 2-7**

***Ilustración de Curva ROC.***



*Nota.* Elaboración propia.

La **Figura 2-7** ilustra la curva ROC de un modelo de clasificación binaria, una representación gráfica que permite visualizar el rendimiento del modelo al mostrar la relación entre la tasa de verdaderos positivos (TPR o sensibilidad que se encuentra en el eje vertical) y la tasa de falsos positivos (FPR que se encuentra en el eje horizontal). Un modelo eficaz se caracteriza por una curva que se desplaza hacia la esquina superior izquierda del gráfico, lo que indica que el modelo es capaz de identificar correctamente la mayoría de los casos positivos y minimizar los errores de clasificación.

En este caso, la curva ROC está bastante cerca de ese vértice, lo que indica que el modelo tiene un buen desempeño. El AUC de 0.92 obtenido por el modelo indica su excelente capacidad para discriminar entre clases positivas y negativas. Específicamente, el modelo tiene un 92% de probabilidad de asignar correctamente una etiqueta positiva a un caso positivo en comparación con un caso negativo. Un valor de AUC más cercano a 1 sugiere un rendimiento superior del modelo en la clasificación.

En contraste, si la curva estuviera cerca de la diagonal punteada, que representa un AUC de 0.5, indicaría que el modelo está haciendo predicciones al azar. La diagonal punteada actúa como una referencia para ver qué tan lejos está el modelo de hacer predicciones sin valor.

Finalmente, este gráfico nos muestra que el modelo es eficaz para identificar correctamente las clases con pocos falsos positivos, lo que lo convierte en una buena opción para el problema de clasificación abordado. Este tipo de visualización es crucial cuando se están comparando modelos, ya que la curva ROC facilita la elección del mejor clasificador según el contexto.

#### 2.6.7 Precision@k

Esta métrica mide la precisión de las recomendaciones dentro de las primeras  $k$  posiciones. Se calcula con la fórmula:

$$\text{Precisión@}k = \frac{\text{Elementos relevantes en las primeras } k \text{ posiciones}}{k} \quad (23)$$

Es útil para entender la relevancia de las recomendaciones más importantes para el usuario (He et al., 2017).

En el próximo capítulo, se describirán las etapas clave del desarrollo y la validación del modelo utilizado en este proyecto, con el fin de garantizar la calidad y precisión de los resultados obtenidos..

# CAPÍTULO 3

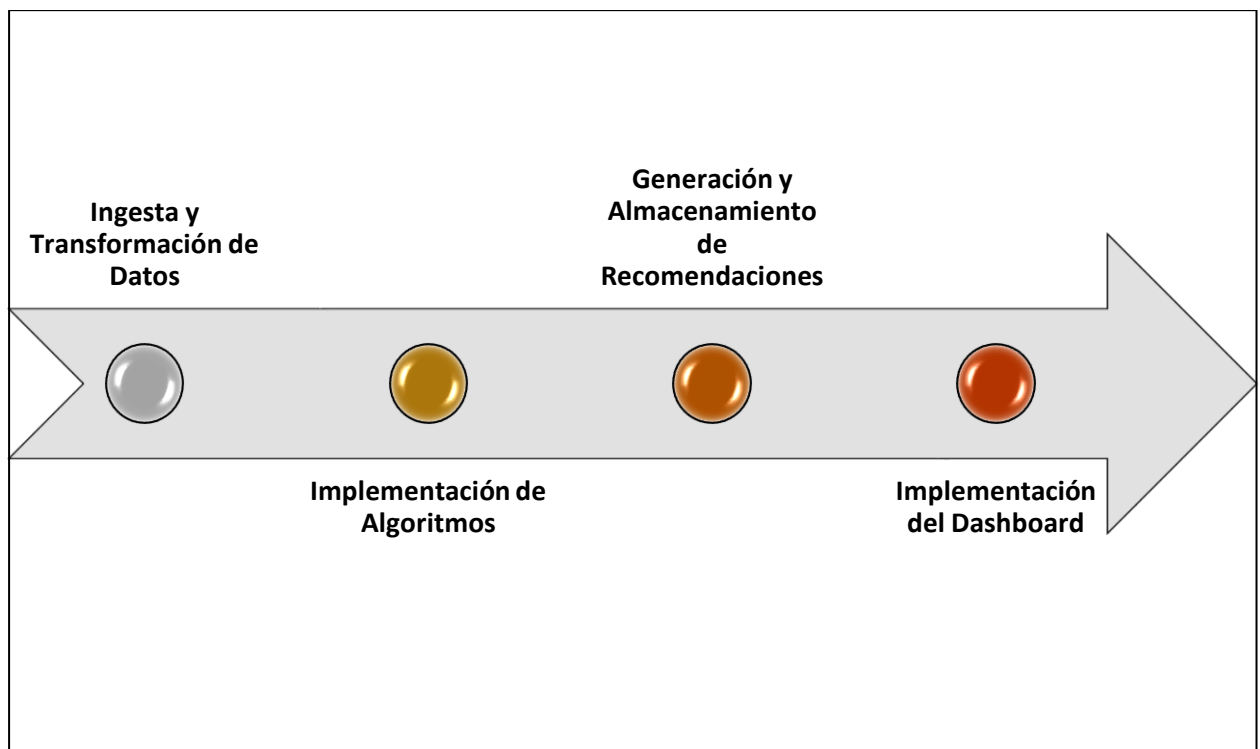
## 3. METODOLOGÍA

### 3.1 Esquema de la metodología

En este capítulo, la **Figura 3-1** ilustra un recorrido estructurado de las etapas clave que nos guiarán paso a paso hacia nuestros objetivos. Cada fase ha sido diseñada estratégicamente para asegurar que avancemos de manera firme y coherente, enfrentando desafíos y aprovechando oportunidades. Este enfoque nos permitirá transformar nuestras ideas en acciones concretas y resultados tangibles.

**Figura 3-1**

***Metodología del Proyecto.***



*Nota.* Elaboración propia.

### **3.1.1 Ingesta y Transformación de Datos**

#### **3.1.1.1 Extracción de datos**

Para la obtención del dataset, se realizó una extracción de datos desde el sistema ERP de la empresa retail, utilizando consultas SQL para seleccionar transacciones de ventas realizadas entre enero y septiembre del 2024. Se filtraron únicamente las 25 tiendas con mayor volumen de ventas y los productos en oferta durante este período.

El equipo de Data Science proporcionó los datos en formato CSV, asegurando la correcta tipificación de las variables (categóricas, numéricas y temporales). Se aplicaron procesos de limpieza para eliminar registros nulos, duplicados o inconsistentes.

El dataset proporcionado contiene 11 millones de registros y 10 columnas clave, estructurados para su análisis y posterior aplicación en los modelos de recomendación. Para futuras actualizaciones, será necesario solicitar nuevas extracciones de datos.

#### **3.1.1.2 Análisis exploratorio**

El análisis exploratorio de datos (EDA) es una etapa clave para asegurar que la información utilizada sea adecuada y esté en condiciones óptimas para su posterior modelado. Para este proyecto, el análisis exploratorio se enfocó en la revisión inicial del dataset, examinando la calidad de los datos e identificando patrones y relaciones preliminares entre las variables.

Inicialmente, el dataset consistía en 11 millones de registros correspondientes a transacciones realizadas en un conjunto de 25 tiendas de mayor venta. Como parte de este proceso, se seleccionaron categorías de productos de alta rotación, tales como limpieza, perfumería, bebidas y comestibles, y se aplicaron filtros adicionales para enfocarnos en clientes con más de tres visitas y un ticket promedio de \$22. Tras aplicar estos filtros, el conjunto de datos se redujo a aproximadamente 2.6 millones de registros, que fueron los utilizados para el análisis posterior.

Para validar la integridad de los datos, se realizó una revisión de los valores nulos en las principales variables, así como una verificación de la existencia

de registros duplicados. Ambos análisis confirmaron que el conjunto de datos no presentaba problemas significativos, lo que garantizó su idoneidad para los modelos posteriores.

Además, se llevaron a cabo análisis preliminares de correlación entre las principales variables numéricas relacionadas con las transacciones de venta. Esto permitió identificar relaciones consistentes con lo esperado, como una fuerte correlación entre el precio de venta al público (PVP), y el valor total pagado (valor\_pvp). Estos resultados sugieren que el incremento en cualquiera de estos factores influye directamente en el valor total de la compra, lo cual es coherente con la naturaleza de las transacciones comerciales.

Este análisis inicial proporcionó una base sólida para avanzar hacia las siguientes etapas del proyecto, permitiendo un entendimiento claro de la estructura del conjunto de datos antes de la aplicación de los algoritmos de recomendación.

### **3.1.2 Segmentación de clientes**

Para potenciar la aplicación de los algoritmos de recomendación y mejorar la eficiencia del sistema, es fundamental llevar a cabo una segmentación de clientes utilizando técnicas de clustering. Se aplica el algoritmo K-Means, el cual agrupa a los clientes en función de características clave como la frecuencia con la que los usuarios compran, el valor promedio de compra por factura y la diversidad de productos adquiridos.

La selección de estas variables se fundamenta en su impacto en los patrones de compra y su utilidad para la generación de recomendaciones personalizadas. Para definir el número óptimo de clusters, se recurre a emplear el del método del codo, midiendo la variación de la inercia en función de la cantidad de agrupaciones.

Esta segmentación permite reducir la dimensionalidad del problema, priorizar las recomendaciones para los clientes con mayor relevancia comercial y optimizar la asignación de recursos computacionales en la etapa de implementación de los algoritmos de recomendación.

### 3.1.3 Implementación de Algoritmos

En el capítulo 2, se presentó una visión general de los algoritmos de recomendación, los cuales comúnmente se basan en la relación usuario-producto. Sin embargo, en este capítulo se aborda su aplicación específica dentro del sistema desarrollado, donde la recomendación se basa en una combinación producto-producto. Este enfoque considera productos que han sido comprados conjuntamente en transacciones previas, permitiendo capturar patrones de compra recurrentes y estructurar recomendaciones más alineadas con el comportamiento histórico de los clientes.

Esta sección se enfoca en la aplicación de los algoritmos de recomendación escogidos para este estudio. Los algoritmos k-Nearest Neighbors (k-NN), Alternating Least Squares (ALS), y Neural Collaborative Filtering (NCF), que forman el núcleo del sistema, determinando la calidad y precisión de las recomendaciones basadas en los datos transaccionales disponibles.

#### 3.1.3.1 *k-Nearest Neighbors (k-NN)*

- **Objetivo:** Identificar productos similares basados en el historial de compras para recomendar aquellos que han sido comprados en conjunto.
- **Datos Utilizados:**
  - **Matriz producto-producto:** Representa la frecuencia con la que dos productos han sido adquiridos en la misma transacción.
  - **Matriz de similitud:** Calculada para medir la similitud entre productos en función de sus compras conjuntas.

- **Hiperparámetros y configuración:**

**Tabla 3-1**

***Configuración de hiperparámetros para el algoritmo k-NN***

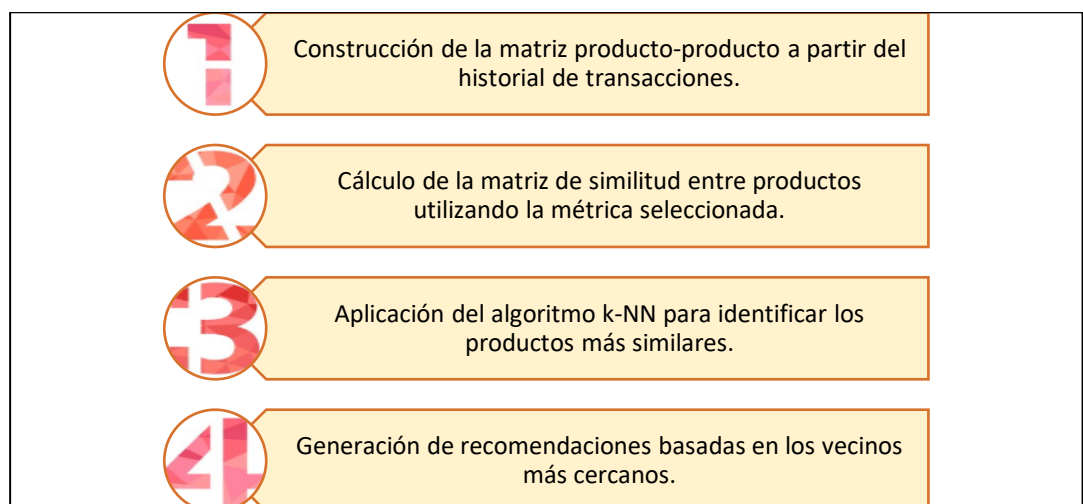
Hiperparámetros	Descripción	Valores probados
Número de vecinos (k)	Determina cuántos productos similares se consideran en la recomendación.	5, 10, 20
Métrica de similitud	Método usado para calcular la distancia entre productos.	Coseno, Euclidiana
Peso de las interacciones	Asigna mayor importancia a productos con más compras conjuntas.	Binario, Frecuencia

*Nota.* Elaboración propia.

- **Proceso de Implementación:**

**Figura 3-2**

***Proceso de Implementación del Algoritmo k-NN.***



*Nota.* Elaboración propia.

### 3.1.3.2 Alternating Least Squares (ALS)

- **Objetivo:** Identificar patrones de compra conjunta entre productos mediante la factorización de matrices, permitiendo generar recomendaciones basadas en compras previas.
- **Datos Utilizados:**
  - **Matriz producto-producto:** Representa la frecuencia con la que dos productos han sido comprados juntos en una misma transacción.
  - **Matrices factorizadas:** Se descompone la matriz producto-producto en dos matrices de menor dimensión, capturando patrones latentes en las compras conjuntas.
- **Hiperparámetros y configuración:**

Tabla 3-2

#### *Configuración de hiperparámetros para el algoritmo ALS.*

Hiperparámetros	Descripción	Valores probados
Número de factores (rank).	Dimensión de las matrices factorizadas.	10, 20, 50
Número de iteraciones	Cantidad de veces que se optimizan los valores en ALS.	10, 20, 50
Parámetro de regularización ( $\lambda$ )	Controla la penalización para evitar sobreajuste.	0.01, 0.1, 1
Método de optimización	Técnica utilizada para minimizar la función de error.	ALS, SGD

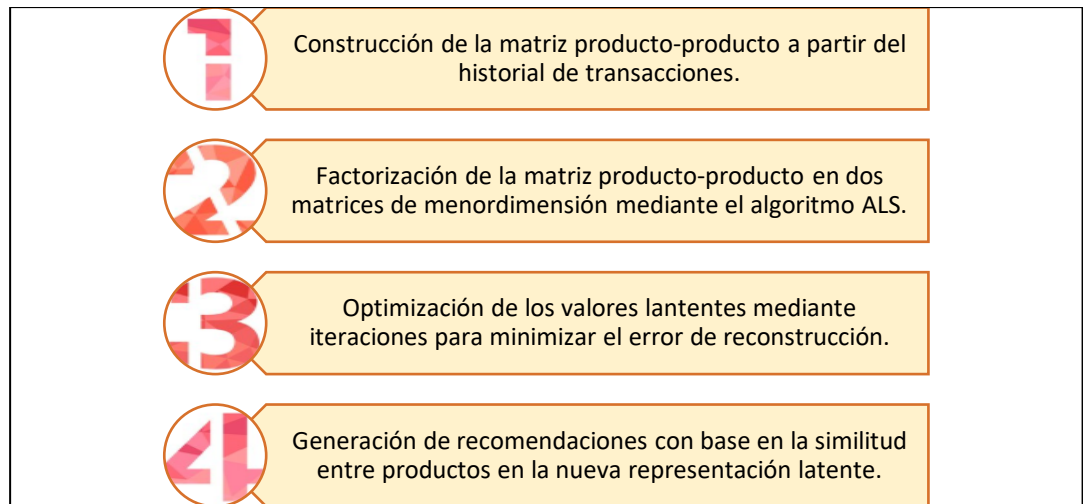
*Nota.* Elaboración propia.



- **Proceso de Implementación:**

**Figura 3-3**

***Proceso de Implementación del Algoritmo ALS.***



*Nota.* Elaboración propia.

**3.1.3.3 Neural Collaborative Filtering (NCF)**

- **Objetivo:** Aplicar redes neuronales profundas para modelar interacciones no lineales entre productos comprados juntos, permitiendo identificar relaciones complejas en los datos de compra.
- **Datos Utilizados:**
  - **Conjunto de interacciones producto-producto:** Dataset de compras conjuntas en transacciones, representadas de forma binaria o ponderada.
  - **Embeddings de productos:** Representaciones vectoriales aprendidas durante el entrenamiento del modelo, capturando similitudes entre productos comprados juntos.

- **Hiperparámetros y configuración:**

**Tabla 3-3**

***Configuración de hiperparámetros para el algoritmo NCF.***

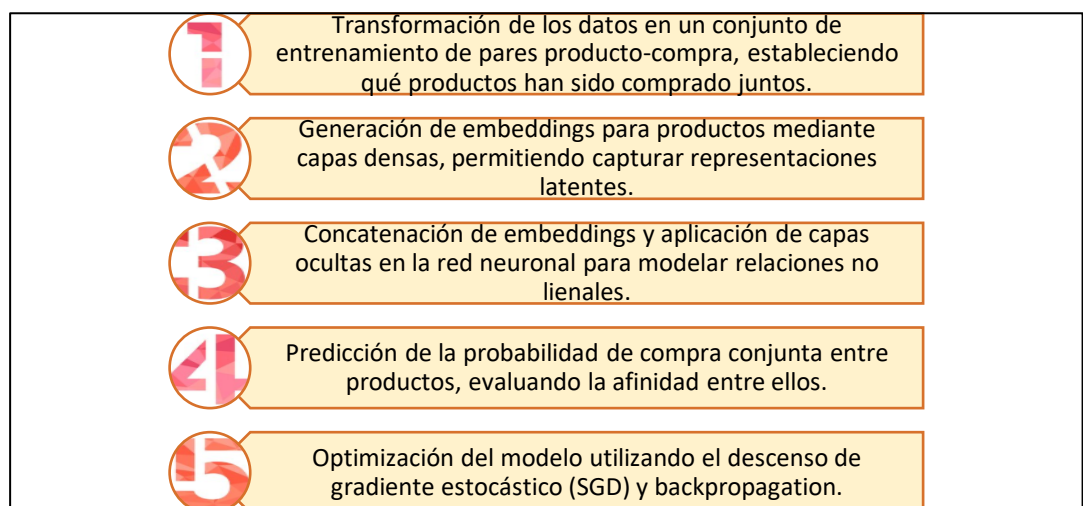
Hiperparámetros	Descripción	Valores probados
Número de capas ocultas	Cantidad de capas en la red neuronal.	2, 3, 4
Número de neuronas por capa	Número de nodos en cada capa oculta.	32, 64, 128
Función de activación	Transformación aplicada en cada capa.	ReLU, Sigmoid
Batch size	Cantidad de muestras procesadas por iteración.	32, 64, 128
Learning rate	Tasa de aprendizaje del modelo.	0.001, 0.01

*Nota.* Elaboración propia.

- **Proceso de Implementación:**

**Figura 3-4**

***Proceso de Implementación del Algoritmo NCF.***



*Nota.* Elaboración propia.

### **3.1.4 Métricas de Evaluación del Modelo**

Para analizar el desempeño del sistema de recomendación, se emplean las métricas Precisión, Recall y F1-Score. Estas métricas son adecuadas porque el sistema no predice calificaciones explícitas ni clasificaciones binarias, sino que genera rankings de productos recomendados.

- Precisión mide la proporción de productos recomendados que realmente fueron relevantes para el usuario.
- Recall evalúa cuántos de los productos realmente relevantes fueron incluidos en las recomendaciones.
- F1-Score proporciona un balance entre precisión y recall, permitiendo evaluar el desempeño global del sistema.

Estas métricas fueron seleccionadas en lugar de otras como RMSE y MAE, ya que estas últimas están diseñadas para modelos que predicen calificaciones numéricas y no rankings de recomendaciones. Asimismo, la Matriz de Confusión y la Curva ROC no son adecuadas, pues están diseñadas para problemas de clasificación binaria y no consideran la calidad del orden de las recomendaciones.

Por lo tanto, Precisión, Recall y F1-Score ofrecen una evaluación más representativa de cómo el sistema genera recomendaciones relevantes para los usuarios.

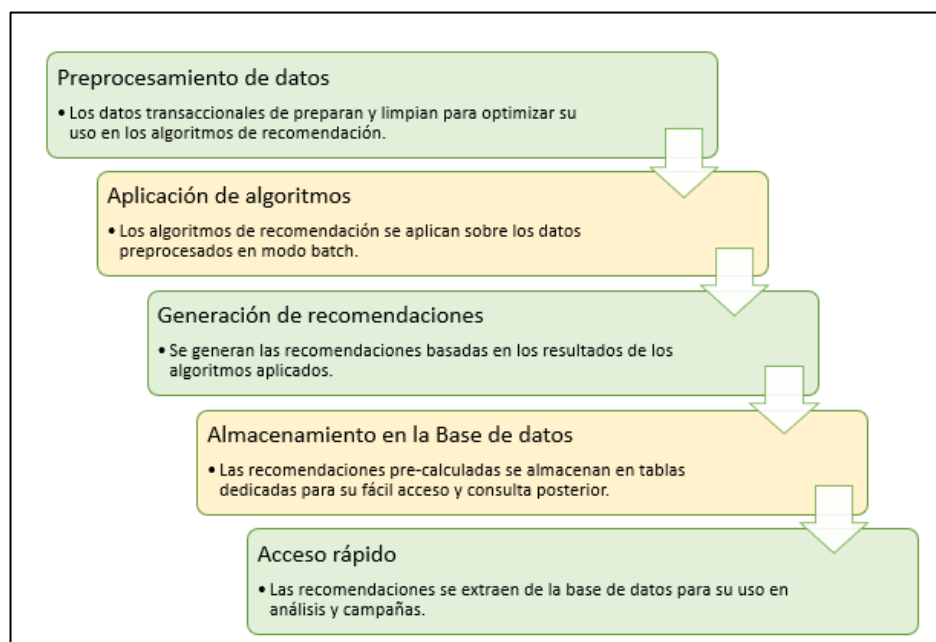
### **3.1.5 Generación y Almacenamiento de Recomendaciones**

La generación y almacenamiento de recomendaciones es una fase clave en el sistema de recomendación. Debido a que el cálculo de recomendaciones en tiempo real puede resultar costoso en términos computacionales, este proyecto se enfoca en la generación de recomendaciones pre-modeladas y su almacenamiento. Este enfoque permite que las recomendaciones se generen en lotes (batch), lo que mejora el rendimiento al minimizar los costos asociados con el procesamiento en tiempo real. Las recomendaciones pre-elaboradas se almacenan para su fácil acceso, permitiendo que el sistema responda rápidamente a las consultas sin recalculación los resultados cada vez.

La **Figura 3-5** presenta el proceso de generación y almacenamiento de recomendaciones. Este proceso comienza con el preprocesamiento de datos y termina con el acceso rápido a las recomendaciones generadas, permitiendo un rendimiento óptimo en la ejecución del sistema.

**Figura 3-5**

***Proceso de Generación y Almacenamiento de Recomendaciones.***



*Nota.* Elaboración propia.

### **3.1.6 Implementación del Dashboard**

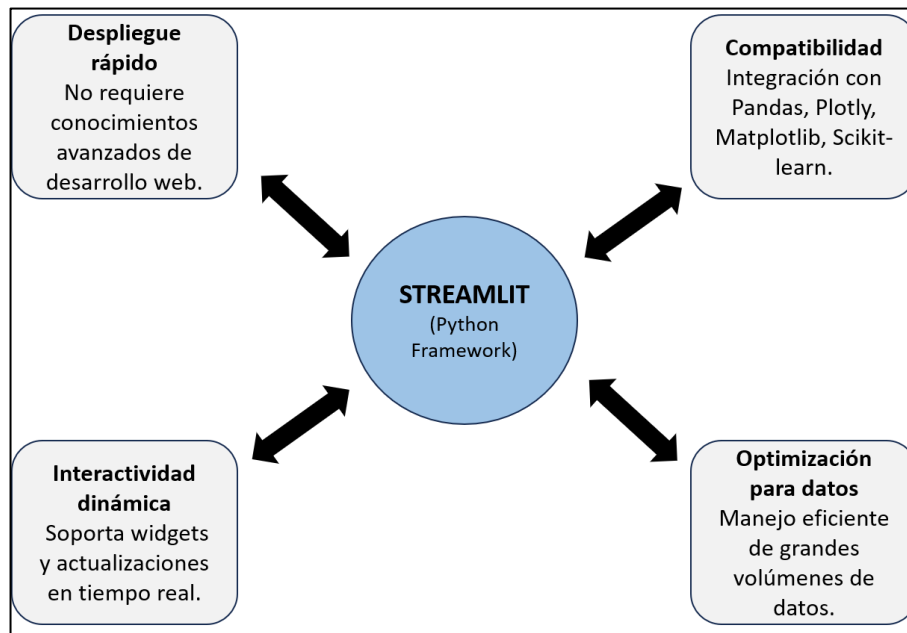
Para la implementación del dashboard, se ha seleccionado Streamlit, una solución basada en Python que facilita la creación de aplicaciones web. Streamlit permite integrar visualizaciones dinámicas con un backend en Python sin necesidad de conocimientos avanzados en desarrollo web.

Streamlit se eligió por su enfoque intuitivo, rápida implementación, compatibilidad con bibliotecas populares como Pandas, Plotly y Scikit-learn, y su capacidad para manejar big data de forma eficiente.

En la **Figura 3-6** se ilustran estas características, que justifican su selección para el desarrollo del sistema.

**Figura 3-6**

***Ventajas del uso de Streamlit.***



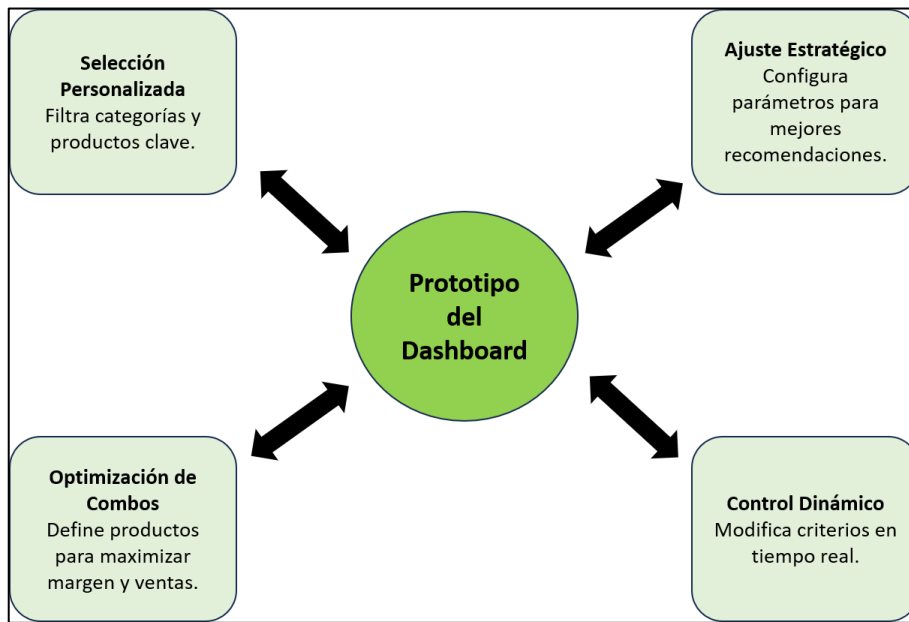
*Nota.* Elaboración propia.

### ***3.1.6.1 Prototipo de Visualización***

El prototipo del dashboard es clave para obtener feedback temprano y realizar ajustes. La **Figura 3-7** presenta los atributos más relevantes que incluirá el prototipo de visualización, enfocadas en ofrecer una interfaz intuitiva, gráficos dinámicos e interactividad a través de un panel de control.

**Figura 3-7**

***Funcionalidades clave del prototipo del dashboard***



*Nota.* Elaboración propia.

**3.1.6.2 Desarrollo Iterativo**

Durante el desarrollo del dashboard, se implementarán ciclos de mejora continua basados en el feedback de los stakeholders y las pruebas de usabilidad. La **Figura 3-8** presenta los componentes clave del enfoque iterativo, orientado a asegurar que el dashboard cumpla con las expectativas de los usuarios y presente los datos de manera efectiva.

**Figura 3-8**

***Proceso iterativo de mejora del dashboard.***



*Nota.* Elaboración propia.

### **3.1.6.3 Despliegue y Acceso**

- **Despliegue Local:** Inicialmente en un servidor local para pruebas internas.
- **Despliegue en la Nube:** Una vez validado, se considerará el despliegue en la nube para escalabilidad.

## **3.2 Infraestructura para procesamiento y almacenamiento**

Esta sección describe la infraestructura diseñada para el procesamiento y almacenamiento de datos del sistema de recomendación. Debido a que este proyecto se enfoca en un prototipo o proyecto piloto, la infraestructura se ha optimizado para operar en un entorno in-house, utilizando recursos locales que garantizan escalabilidad futura.

### **3.2.1 Procesamiento de Datos**

El procesamiento de datos se efectuará en un entorno in-house, utilizando una máquina o laptop local. Este enfoque resulta ideal para el prototipo, ya que proporciona control total sobre el entorno de desarrollo, facilitando iteraciones rápidas y flexibles del sistema. Además, este método garantiza la capacidad de ajustar los procesos de acuerdo con las demandas particulares del proyecto.

#### **Método de Procesamiento en Batch:**

- Los datos se procesarán en batch, ejecutando los algoritmos de recomendación sobre un conjunto de datos predefinido. Este método es eficiente para el volumen de datos manejado en el prototipo.
- La capacidad computacional de la máquina local es suficiente para procesar los datos transaccionales de un subconjunto de clientes (aproximadamente 1000).

### **3.2.2 Escalabilidad**

Aunque el sistema se ejecuta en un entorno local para el prototipo, ha sido diseñado para ser escalable. Esto permite una futura migración a infraestructuras más robustas, como servidores dedicados o entornos en la nube, si se decide avanzar hacia un despliegue a gran escala.

## **3.3 Almacenamiento de Datos**

#### **• Base de Datos In-house:**

- El almacenamiento de la información se llevará a cabo en una base de datos designada específicamente para este proyecto piloto. Esta base contiene transacciones de un período específico, organizadas en tablas optimizadas para el acceso y procesamiento por los algoritmos.
- Las inferencias generadas por los modelos de recomendación se almacenarán en tablas separadas, permitiendo su fácil acceso para la ejecución de campañas y análisis.



- **Tabla de Recomendaciones:**

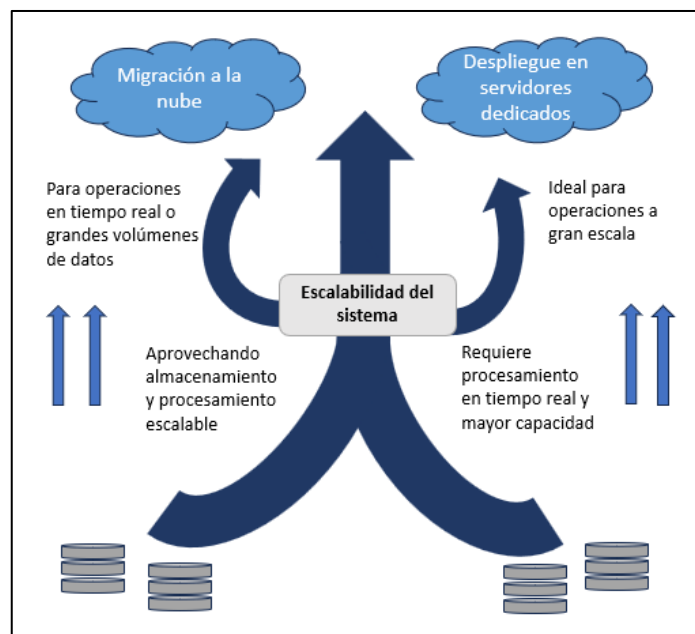
- Las recomendaciones generadas se almacenarán en una tabla específica, que será el soporte para la toma de decisiones durante el periodo de ejecución del batch. Esta tabla podrá ser consultada según sea necesario, sin requerir recálculo bajo demanda.

### 3.3.1 Consideraciones Futuras

En las consideraciones futuras, se destacan dos opciones principales para la escalabilidad del sistema: la migración a la nube y el despliegue en servidores dedicados. La **Figura 3-9** presenta estas opciones, enfocadas en mejorar el rendimiento y la capacidad de procesamiento a gran escala.

**Figura 3-9**

***Opciones de escalabilidad para el sistema***



*Nota.* Elaboración propia.

### 3.3.2 Criterios de selección de combos

La selección final de los combos recomendados es realizada por los compradores, quienes eligen entre las opciones generadas por el sistema en función de diversos criterios comerciales. El sistema propuesto presenta opciones de combos optimizados, pero la decisión de compra está influenciada por factores estratégicos internos, tales como:

- **Margen de rentabilidad:** Los compradores suelen priorizar combos con un margen atractivo para maximizar la rentabilidad de la venta.
- **Acuerdos con proveedores:** En algunos casos, las compras están sujetas a acuerdos comerciales que requieren alcanzar cierto volumen o monto de compra con determinados proveedores.
- **Popularidad y rotación de inventario:** Se favorecen productos con alta rotación y demanda en el mercado para evitar acumulación de inventario.

Estos criterios aseguran que las opciones presentadas a los compradores no solo optimicen la recomendación basada en datos históricos, sino que también se alineen con las estrategias comerciales del negocio.

# CAPÍTULO 4

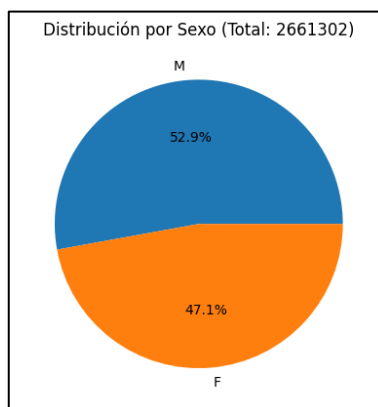
## 4. RESULTADOS

### 4.1 Recolección de Datos y Estrategias para la Evaluación del Proyecto

El propósito del proyecto es desarrollar un sistema de recomendación para optimizar la selección de combos mediante el análisis de patrones de compra. La validación de los objetivos se realizó a través de una serie de estrategias que permitieron evaluar el rendimiento del modelo y su efectividad en un entorno real. Estas estrategias abarcan desde la segmentación de clientes y la validación de productos clave, hasta el análisis de desempeño y la comparación de modelos. A continuación, se detallan cada una de estas etapas y los resultados obtenidos. Antes de iniciar las siguientes etapas, se ejecutó una depuración de datos, eliminando registros que no cumplían con ciertos criterios (clientes fidelizados, productos activos, alta rotación, ticket promedio). Como resultado, se obtuvo una base de datos más precisa, donde el 52,9% de las compras fueron realizadas por mujeres, representando un total de 2,6 millones de registros.

**Figura 4-1**

***Distribución de clientes por sexo***



*Nota.* Elaboración propia.

#### 4.1.1 Análisis de Segmentación de Clientes

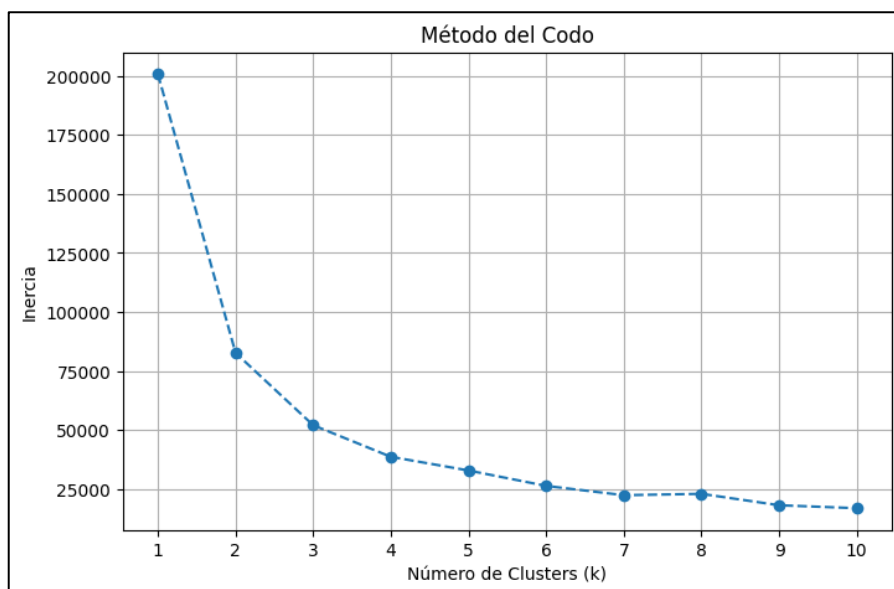
El análisis de segmentación fue esencial para identificar los grupos de clientes más relevantes. Se aplicó el algoritmo K-Means para generar cuatro clusters basados en frecuencia de compra, valor promedio de compra y diversidad de productos. A partir de esta segmentación, se determinó que los **Clústers 3 y 4** constituían los segmentos más adecuados para las recomendaciones, ya que representan a los clientes con la mayor frecuencia de compra y el mayor valor promedio de compra. Esto los convierte en un grupo prioritario para estrategias de fidelización y aumento de ventas. A continuación se detalla el proceso:

En primer lugar, se utilizó el método del codo, que permite identificar el número óptimo de clusters, evitando una segmentación excesiva que dificulte el análisis. Donde se determinó que:

- Con 3 o 4 clusters se logra un buen balance entre simplicidad y precisión al agrupar clientes.
- Más clusters no aportan mucha mejora, pero sí complican el análisis.

**Figura 4-2**

***Determinación del número óptimo de clústers mediante el Método del Codo***



*Nota.* Elaboración propia.

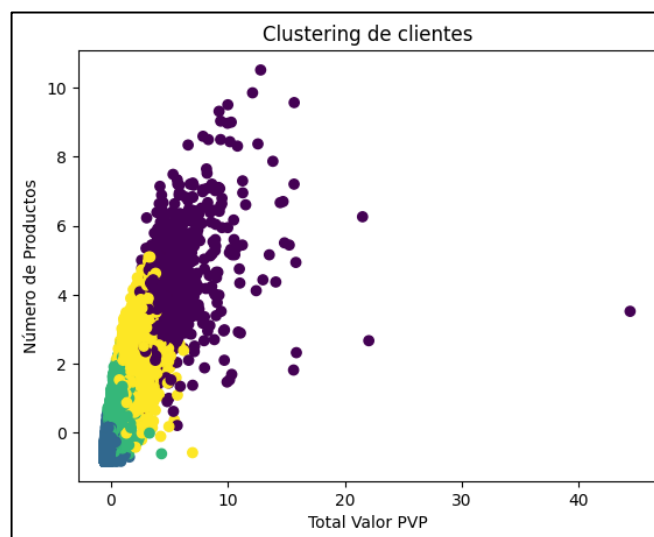
Luego, se aplicó el algoritmo K-Means, obteniendo el resultado de la clusterización de clientes basado en las variables Total Valor PVP (gasto total) y Número de Productos (diversidad de productos comprados). En la **Figura 4-3**, presentada a continuación, cada punto representa un cliente, y los colores identifican los grupos (clusters) formados.

Es decir:

- Los clusters separan a los clientes según su comportamiento de compra. Por ejemplo, clientes con alto gasto y mayor diversidad de productos están en una región distinta a los clientes con bajo gasto y poca diversidad de productos. Esto permite identificar patrones de consumo, segmentando a los clientes para estrategias específicas, como promociones o fidelización.

**Figura 4-3**

***Segmentación de clientes mediante K-Means (2D)***



*Nota.* Elaboración propia.

El gráfico tridimensional presentado en la **Figura 4-4**, permite visualizar los resultados de un análisis que busca agrupar a los clientes en segmentos homogéneos basados en características comunes, en este caso, tres variables:

- **Total Valor PVP (Escalado):** Representa el valor total de las compras realizadas por cada cliente, normalizado para facilitar la comparación.
- **Número de Productos (Escalado):** Indica la cantidad de productos adquiridos por cada cliente.
- **Frecuencia de Compra (Escalado):** Refleja la frecuencia con la que un cliente realiza compras.

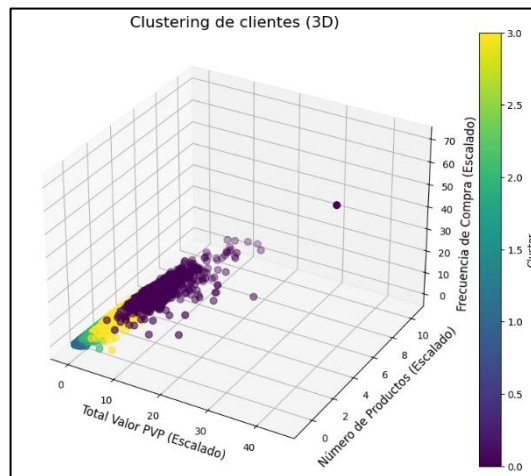
Cada punto en el gráfico representa un cliente individual, y su posición en el espacio tridimensional está determinada por los valores de estas tres variables. Los diferentes colores de los puntos indican a qué cluster pertenece cada cliente.

La disposición de los puntos en la **Figura 4-4** revela patrones interesantes y permite identificar distintos perfiles de clientes:

- **Cluster 1 (Colores más fríos):**
  - ✓ **Características:** Clientes que realizan un número bajo de compras, con un valor total bajo y una frecuencia de compra baja.
  - ✓ **Interpretación:** Probablemente se trate de clientes nuevos o poco activos, que podrían requerir estrategias de activación y fidelización.
- **Cluster 2 (Colores intermedios):**
  - ✓ **Características:** Clientes con un comportamiento de compra más equilibrado, con valores intermedios en las tres variables.
  - ✓ **Interpretación:** Podrían ser considerados como clientes regulares, que representan una base sólida para la empresa.
- **Cluster 3 y 4 (Colores más cálidos):**
  - ✓ **Características:** Clientes que realizan un número elevado de compras, con un valor total alto y una frecuencia de compra alta.
  - ✓ **Interpretación:** Son los clientes de mayor valor para la empresa, ya que generan una gran parte de los ingresos. Representan una oportunidad para programas de fidelización más personalizados y ofertas exclusivas.

**Figura 4-4**

***Visualización tridimensional de la segmentación de clientes con k-Means.***



*Nota.* Elaboración propia.

Teniendo como resultado en el alcance del dataset en un 1 millón de registros a utilizarse en el proyecto.

#### **4.1.2 Selección de Productos Relevantes**

Simultáneamente, se realizó un análisis de productos para identificar los más demandados y relevantes para el sistema de recomendaciones.

De acuerdo con el análisis de demanda, se seleccionó el top 200 de productos con mayores ventas como base de los productos a recomendar. Enfocar el sistema en estos productos permite optimizar la relevancia de las recomendaciones y aumentar la satisfacción del cliente, asegurando que las recomendaciones se concentren en productos de alta demanda.

#### **4.1.3 Selección de hiperparámetros**

Para garantizar el mejor desempeño de cada modelo, se realizaron experimentos con diferentes combinaciones de hiperparámetros, definidos en el Capítulo 3. La selección de los valores óptimos se basó en el mejor equilibrio entre Precisión,

Recall y F1-Score, asegurando estabilidad en las recomendaciones sin comprometer la capacidad de generalización del modelo.

A continuación, se presentan los valores óptimos seleccionados para cada algoritmo:

**Tabla 4-1**

***Hiperparámetros óptimos seleccionados***

Algoritmo	Hiperparámetro	Valor óptimo
KNN	Número de vecinos (k)	10
	Métrica de similitud	Coseno
	Peso de interacciones	Frecuencia
ALS	Número de factores (rank)	20
	Número de iteraciones	50
	Parámetro de regularización ( $\lambda$ )	0.1
	Método de optimización	ALS
NCF	Número de capas ocultas	3
	Número de neuronas por capa	64
	Función de activación	ReLU
	Batch size	64
	Learning rate	0.001

*Nota.* Elaboración propia.

Estos valores fueron determinados tras evaluar múltiples combinaciones en escenarios controlados. Los hiperparámetros seleccionados permitieron a cada modelo alcanzar su mejor rendimiento, como se observa en las próximas secciones.

#### **4.1.4 Back Testing**

Tras definir los segmentos clave de clientes y productos, se realizó un back testing para evaluar la capacidad predictiva del sistema de recomendaciones. En este proceso, se particionó la data en conjuntos de entrenamiento y prueba: los datos



de enero a marzo de 2024 se utilizaron para entrenar el modelo, mientras que los datos de abril de 2024 se reservaron para pruebas y validación. Esta metodología permitió evaluar la eficacia del modelo considerando data que no fue usada en el entrenamiento, lo cual garantiza que las recomendaciones generadas sean pertinentes y robustas en un entorno simulado de producción.

Para el análisis de rendimiento, se emplearon las métricas de Precisión, Recall y F1-Score, centrándose en las cinco principales recomendaciones para asegurar su pertinencia para el usuario final. Los resultados se presentan a continuación:

**Tabla 4-2**

***Comparación de modelos – Back testing***

Modelo	Precisión	Recall	F1-Score
KNN	0.6500	0.6000	0.6250
ALS	0.7000	0.6500	0.6750
NCF	0.7800	0.7200	0.7500

*Nota.* Elaboración propia.

Estos resultados muestran que NCF obtuvo el mejor rendimiento en todas las métricas, lo que indica que genera recomendaciones más precisas y ordenadas de manera efectiva.

**4.1.5 Comparación de Modelos**

En esta fase crítica del proyecto, se realizó una comparación exhaustiva de tres algoritmos de recomendación: K-Nearest Neighbors (KNN), Alternating Least Squares (ALS) y Neural Collaborative Filtering (NCF). El objetivo era identificar el modelo con el mejor rendimiento en términos de precisión, relevancia y eficiencia, asegurando que las recomendaciones ofrecidas fueran útiles para los usuarios en un contexto realista.

Para esta comparación, se emplearon datos de mayo a agosto de 2024 para el entrenamiento y los datos de septiembre de 2024 para las pruebas y validación.

Esta división permitió evaluar cómo los modelos se comportaban con un conjunto de datos actualizado y nunca antes visto, simulando un entorno de producción. Se seleccionaron los 200 productos más relevantes en términos de ventas para asegurar que las recomendaciones fueran significativas y enfocadas en productos de alta demanda.

Cada modelo fue evaluado utilizando las métricas de rendimiento: Precisión, Recall y F1-Score. Estas métricas permiten evaluar la efectividad de las recomendaciones, midiendo tanto la exactitud como la capacidad del sistema para capturar todos los elementos relevantes.

**Tabla 4-3**

***Comparación de modelos en la Implementación***

Modelo	Precisión	Recall	F1-Score
KNN	0.9787	0.6578	0.7832
ALS	0.9800	0.6782	0.7895
NCF	0.9867	0.6904	0.8012

*Nota.* Elaboración propia.

Como se indica en la **Tabla 4-3**, el modelo NCF superó a los otros algoritmos, logrando la mayor precisión y F1-score, lo que lo convierte en la opción más eficaz para este sistema. El modelo ALS también demostró robustez; y KNN, aunque es el modelo más sencillo, presentó un desempeño inferior en comparación a los otros dos. La elección del modelo más preciso se basó en la evaluación de estas métricas, las cuales reflejan la capacidad del modelo para generar recomendaciones precisas y relevantes.

## **4.2 Puesta en Marcha y Funcionamiento**

La implementación del sistema de recomendación se realizó en tres fases, abarcando el análisis exploratorio, la clusterización y el desarrollo final del modelo.

Este enfoque permitió optimizar los recursos disponibles y superar limitaciones técnicas sin comprometer la precisión del modelo.

#### **4.2.1 Limitaciones de Recursos y Estrategia de Implementación en Fases**

La implementación inicial del sistema se realizó en un entorno de desarrollo con Python versión 3.10. En lugar de un entorno de producción completo, se utilizó un entorno controlado que permitió realizar pruebas exhaustivas de los modelos y ajustar sus configuraciones sin necesidad de infraestructura adicional en la nube. Inicialmente, el sistema trabajaba con un conjunto de datos de aproximadamente 11 millones de registros, lo que excedía las capacidades de memoria disponibles. Para optimizar el procesamiento, se implementó una estrategia de segmentación que incluyó trabajar únicamente con el clúster de clientes con mayor frecuencia de compra y seleccionar el top 200 de productos con mayores ventas, como fue descrito previamente en las secciones de Análisis de Segmentación de Clientes y Selección de Productos Relevantes.

Esta delimitación estratégica permitió reducir el volumen de datos a 1 millón de registros, haciendo que el procesamiento fuera más eficiente sin comprometer la calidad ni la relevancia de las recomendaciones.

#### **4.2.2 Producción y Visualización de Recomendaciones**

La solución final integra un prototipo desarrollado en el framework de visualización Streamlit, el cual proporciona a los usuarios una interacción fluida con las recomendaciones generadas por el sistema. Este aplicativo optimiza el proceso de selección de productos y creación de combos, permitiendo reducir el tiempo requerido de una semana a solo 1-2 horas. La integración en los sistemas internos permite al equipo comercial acceder a las recomendaciones de manera eficiente, mejorando la capacidad de respuesta y facilitando la toma de decisiones estratégicas.

La interfaz inicial muestra como el usuario ingresa sus credenciales para acceder al sistema. Se buscó que el diseño sea limpio, intuitivo y de rápida comprensión, reduciendo la curva de aprendizaje para nuevos usuarios. La pantalla presentada

a continuación ratifica que el prototipo cumple con el criterio de **usabilidad**, ya que facilita un acceso seguro y rápido al sistema.

**Figura 4-5**

***Pantalla de autenticación de usuario.***



*Nota.* Elaboración propia.

Una vez autenticado, el sistema guía al usuario hacia la selección de categorías, subcategorías y productos. La navegación está diseñada de forma secuencial y lógica, evitando pasos innecesarios y priorizando la eficiencia operativa.

Se destacan los siguientes aspectos:

- **Filtros claros y personalizables:** Permiten al usuario seleccionar de manera precisa los productos que desea analizar o incluir en los combos.
- **Organización visual:** Los elementos están dispuestos de forma jerárquica, lo que simplifica la identificación de las opciones.
- **Feedback positivo de los usuarios:** Durante las pruebas, los usuarios resaltaron la claridad de los filtros y la facilidad para encontrar los productos deseados.

**Figura 4-6**

***Pantalla de selección de productos.***

Selecciona los Productos para Recomendación

Seleccione una Categoría

Limpieza del Hogar

Seleccione una Subcategoría

LAVADO DE ROPA

Seleccione hasta 4 productos:

DETERGENTE CL... SURVIZANTE SU...

Siguiente

Mostrar iconos ocultos

*Nota.* Elaboración propia.

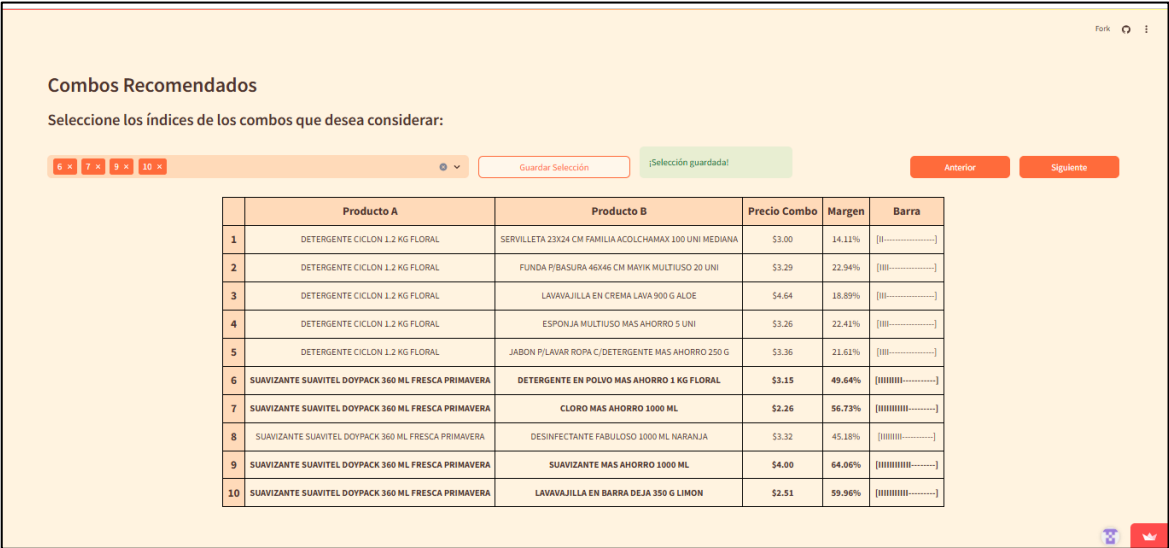
Luego de seleccionar los criterios, el sistema genera una lista de recomendaciones en forma de combos optimizados para maximizar el valor comercial. Esta funcionalidad se validó mediante pruebas que confirmaron que los resultados son coherentes con los criterios seleccionados.

Pruebas realizadas:

- **Validación de resultados:** Se verificó que las recomendaciones sean coherentes con los filtros aplicados por el usuario.
- **Eficiencia operativa:** Se midió el tiempo que toma generar los combos recomendados, logrando reducirlo de días a tan solo minutos.

Figura 4-7

Pantalla de combos recomendados.

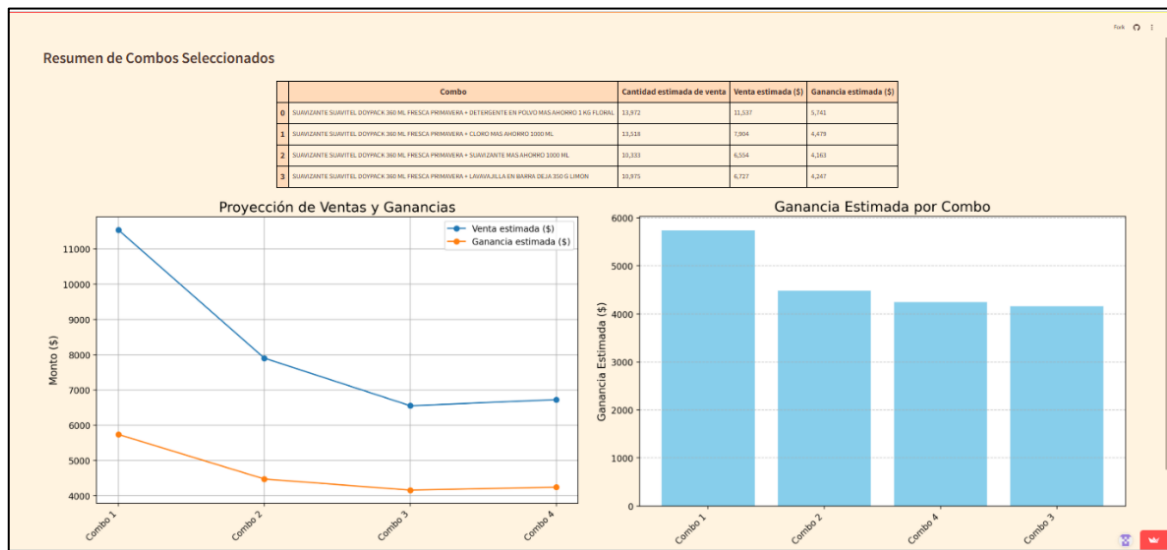


Nota. Elaboración propia.

Durante las pruebas iniciales, los usuarios indicaron que necesitaban visualizar ciertas variables clave dentro del prototipo para poder tomar decisiones estratégicas con mayor rapidez. Estas variables han sido incluidas en la versión actual. A continuación, se presenta la pantalla final del aplicativo donde se muestran los combos elegidos con visualizaciones sobre la proyección de ventas, ganancias estimadas y margen de contribución.

**Figura 4-8**

***Resumen y análisis de los combos recomendados.***



*Nota.* Elaboración propia.

En esta sección final, el sistema presenta un análisis detallado de los combos seleccionados por el usuario. Los datos incluyen:

- **Proyección de ventas:** Gráficos de barra que muestran la expectativa de ingresos con base en los combos elegidos.
- **Ganancias estimadas y márgenes de contribución:** Visualizaciones que ayudan a evaluar la rentabilidad de las recomendaciones.
- **Resumen visual de combos seleccionados:** Ayuda al usuario a comparar y priorizar las opciones disponibles.

### 4.3 Pruebas de Funcionalidad

Dado que el aplicativo se encuentra en fase de prototipo, se realizaron pruebas específicas para validar su funcionalidad, eficiencia y usabilidad. El sistema aún no es completamente funcional, sino que representa un modelo inicial diseñado para evaluar la viabilidad y utilidad de las recomendaciones en un entorno controlado. Actualmente, recoge las ideas y necesidades del equipo comercial, cumpliendo con los requisitos mínimos de operación.

En una implementación final, se incorporarán más variables y mejoras técnicas.

#### **4.3.1 Evidencias de Funcionamiento del Prototipo**

El prototipo del sistema de recomendación fue presentado al equipo comercial de la empresa para una evaluación preliminar. Durante esta sesión, se explicó el funcionamiento del sistema y los usuarios interactuaron con las funcionalidades principales, incluyendo la generación de recomendaciones y las visualizaciones interactivas. Los comentarios obtenidos fueron cualitativos, basados en observaciones de cómo los usuarios utilizaban el sistema, y se identificaron oportunidades para perfeccionar la interfaz y enriquecer la experiencia del usuario.

#### **4.3.2 Pruebas de Tiempo de Respuesta**

El tiempo de respuesta fue un aspecto crítico evaluado durante las pruebas del prototipo. Se realizaron simulaciones para medir cuánto tiempo tomaba generar recomendaciones, especialmente bajo condiciones de uso intensivo. Los resultados mostraron que el modelo NCF podía responder en un promedio de 1.3 segundos, cumpliendo con las expectativas de eficiencia para un sistema de recomendación y demostrando que el prototipo es capaz de manejar interacciones rápidas y fluidas.

#### **4.3.3 Pruebas de Precisión**

La precisión del sistema fue evaluada a través de pruebas basadas en datos históricos. Se compararon las recomendaciones generadas por el prototipo con los patrones de compra registrados en el pasado para verificar la relevancia y exactitud de las sugerencias. Las métricas empleadas fueron **Precisión**, **Recall** y **F1-Score**, enfocándose en las cinco principales recomendaciones. El modelo NCF se destacó en estas métricas, evidenciando que las sugerencias eran consistentes y alineadas con las necesidades del usuario final.



#### 4.3.4 Pruebas de Usabilidad y Experiencia de Usuario del Prototipo

El segundo objetivo se centró en proporcionar una experiencia de usuario efectiva mediante visualizaciones interactivas. Para validar este aspecto, se llevaron a cabo pruebas de usabilidad específicas para prototipos, donde se observó a los usuarios del equipo comercial interactuar con las visualizaciones desarrolladas en Streamlit. Las observaciones incluyeron:

- **Facilidad de Navegación:** Se analizó si los usuarios podían navegar intuitivamente por las visualizaciones y comprender la información presentada sin necesidad de asistencia adicional.
- **Interacción con Recomendaciones:** Se evaluó cómo los usuarios interpretaban y utilizaban las recomendaciones generadas, y si estas les parecían relevantes y útiles para la toma de decisiones comerciales.

Los resultados de estas observaciones fueron positivos, indicando que el prototipo es fácil de usar y que las visualizaciones son claras y efectivas, aunque se identificaron áreas para futuras mejoras.

#### 4.3.5 Pruebas de Impacto Simulado en la Tasa de Conversión

Para cumplir con el tercer objetivo de evaluar la eficiencia del modelo en mejorar la tasa de conversión, se realizaron pruebas simuladas en un entorno de prototipo. Se utilizaron datos históricos y escenarios hipotéticos para analizar el impacto potencial del sistema:

Se evaluaron distintos escenarios de ventas, con y sin las recomendaciones del sistema, para analizar su impacto en las decisiones de compra de los clientes.

**Análisis Comparativo:** Se compararon escenarios de ventas con y sin las recomendaciones del sistema, simulando cómo podría cambiar el comportamiento de compra de los clientes. Los resultados mostraron un incremento teórico en las tasas de conversión, lo que sugiere que, una vez implementado en un entorno real, el sistema podría tener un impacto positivo significativo. Se presentan a continuación, los resultados obtenidos en el análisis:

**Tabla 4-4**

***Impacto del Sistema de Recomendación en la Tasa de Conversión.***

Escenario	Tasa de Conversión (%)	Incremento (%)
Sin recomendaciones	3.2%	-
Con recomendaciones	4.8%	+50%

*Nota.* Elaboración propia.

Estas pruebas confirman que el prototipo podría mejorar la experiencia del usuario y optimizar las tasas de conversión. Sin embargo, se reconoce que se requerirán pruebas más exhaustivas en un entorno de producción para validar completamente estos resultados.

#### **4.4 Análisis Costo/Beneficio**

La evaluación del costo/beneficio del sistema de recomendación es fundamental para determinar su viabilidad económica y su impacto en la estrategia comercial. El desarrollo del prototipo se realizó con cero costos adicionales, aprovechando herramientas gratuitas y equipos personales. Sin embargo, en un entorno real, la inversión necesaria incluiría infraestructura, soporte técnico y licencias.

A continuación, se presentan los costos del prototipo y la proyección de costos para una implementación a gran escala, lo que permite evaluar la diferencia en términos de inversión y retorno.

##### **4.4.1 Desglose de Costos del Prototipo**

La siguiente tabla presenta un desglose detallado de los costos asociados a esta fase inicial:

**Tabla 4-5*****Desglose de Costos del Prototipo***

<b>Categoría</b>	<b>Detalle</b>	<b>Costo estimado</b>
Desarrollo del Prototipo	Implementación con Python y Streamlit Cloud (versiones de prueba gratuitas)	\$0
Equipos	Uso de 2 laptops personales (2 equipos de los autores)	\$0 (sin costo adicional)
Recursos Humanos	Trabajo no remunerado del equipo de desarrollo (horas dedicadas)	\$0
Gastos Operativos	Utilización de recursos gratuitos en Streamlit Cloud	\$0
<b>Total Costos Implementación Real</b>		<b>\$0</b>

*Nota.* Elaboración propia.

**4.4.2 Proyección de Costos para una Implementación en Entorno Real**

En un entorno de producción, sería necesario invertir en infraestructura tecnológica más robusta, licencias de software, y soporte técnico, lo que incrementaría significativamente los costos operativos y de implementación. A continuación, se detallan los costos estimados:

**Tabla 4-6*****Proyección de Costos para una Implementación en Entorno Real***

<b>Categoría</b>	<b>Detalle</b>	<b>Costo estimado</b>
Infraestructura	Servidores dedicados o servicios en la nube, bases de datos seguras	\$50,000 anuales
Licencias de Software	Herramientas de visualización y plataformas de desarrollo	\$5,000 anuales
Equipos	Adquisición de hardware especializado (servidores, laptops de alto rendimiento)	\$10,000
Soporte y Mantenimiento	Costos asociados al monitoreo continuo, actualizaciones y soporte técnico	\$20,000 anuales
Recursos Humanos	Personal especializado para desarrollo y soporte técnico (anual)	\$30,000
<b>Total Costos Implementación Real</b>		<b>\$115,000</b>

*Nota.* Elaboración propia.

**4.4.3 Proyección de Beneficios**

El beneficio proyectado del sistema de recomendación, una vez implementado en un entorno real, se basa en el incremento potencial de las ventas y la mejora en la retención de clientes. Si se estima que el sistema podría generar un aumento de \$1 millón en ventas anuales, con un costo operativo de \$300,000, el beneficio neto ascendería a \$700,000, lo que representaría un retorno de inversión significativo.

**Tabla 4-7**

***Beneficio proyectado de la Implementación del Sistema de recomendación.***

Concepto	Valor estimado
Incremento en Ventas anuales	\$1,000,000
Costos operativos anuales	\$300,000
Beneficio Neto	\$700,000

*Nota.* Elaboración propia.

#### **4.4.4 Consideraciones Finales**

Es crucial destacar que el desarrollo del prototipo se llevó a cabo utilizando recursos limitados, como Python y Streamlit Cloud en sus versiones gratuitas, y fue ejecutado en laptops personales del equipo de proyecto. En un entorno de producción, sin embargo, se requerirá una infraestructura más sólida y un presupuesto adecuado para asegurar el mantenimiento y la eficiencia operativa del sistema. Este análisis evidencia que, aunque el costo inicial del prototipo fue mínimo, la inversión necesaria para su implementación real sería considerable. Sin embargo, los beneficios proyectados justifican dicha inversión, dado el potencial de incremento en las ventas y mejora en la experiencia del cliente.

# CAPÍTULO 5

## 5. CONCLUSIONES Y RECOMENDACIONES

En este capítulo se exponen las conclusiones derivadas del desarrollo de este proyecto y las recomendaciones propuestas para su mejora y futuras implementaciones.

### 5.1 Conclusiones

- El desarrollo del prototipo del sistema de recomendación permitió validar la viabilidad de aplicar modelos de aprendizaje automático para mejorar la selección de combos en el sector retail.
- La implementación en Streamlit facilitó una interfaz intuitiva y funcional, reduciendo significativamente el tiempo de selección de productos, pasando de aproximadamente una semana a 1-2 horas.
- Se determinó que el modelo NCF presentó el mejor rendimiento teniendo en cuenta las métricas de precisión y recall, logrando recomendaciones más relevantes en comparación con KNN y ALS.
- El modelo actualmente no captura la estacionalidad, ya que se basa en los datos disponibles, los cuales presentan un comportamiento homogéneo a lo largo del tiempo. Sin embargo, esto no significa que el sistema no pueda considerar la estacionalidad en futuras versiones. A medida que se incorporen más datos históricos y se disponga de información sobre variaciones de demanda en diferentes períodos del año, el modelo podrá ajustar dinámicamente sus predicciones y capturar tendencias estacionales.
- La segmentación de clientes y la selección del top 200 de productos más vendidos mejoraron la precisión del sistema, asegurando que las recomendaciones sean estratégicamente relevantes.
- Se realizaron pruebas de funcionalidad con el equipo comercial, evidenciando que el prototipo cumple con los requisitos mínimos de operación y puede evolucionar en futuras versiones con mayor cantidad de variables.

- Los resultados del análisis de impacto en ventas indicaron que el sistema podría aumentar la tasa de conversión, aunque su efectividad final deberá ser validada en un entorno de producción.

## **5.2 Recomendaciones**

Para mejorar el sistema de recomendación y facilitar su implementación en entornos reales, se sugieren las siguientes mejoras:

- Ampliar el conjunto de datos con información histórica para que el modelo pueda capturar la estacionalidad y ajustar sus predicciones según la demanda en diferentes períodos del año.
- Explorar modelos híbridos que combinen diversas metodologías para mejorar la precisión de las recomendaciones.
- Evaluar la integración del sistema con plataformas empresariales, como SAP o CRM, para que las recomendaciones sean aprovechadas directamente en la gestión comercial.
- Expandir el sistema a nuevos segmentos de clientes, explorando su aplicación en usuarios con menor frecuencia de compra para identificar estrategias de fidelización.

# BIBLIOGRAFÍA

- Adomavicius, G., & Tuzhilin, A. (2005a). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734–749. <https://doi.org/10.1109/TKDE.2005.99>
- Adomavicius, G., & Tuzhilin, A. (2005b). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734–749. <https://doi.org/10.1109/TKDE.2005.99>
- Aggarwal, C. C. (2016). *Recommender Systems*. <https://doi.org/10.1007/978-3-319-29659-3>
- Brynjolfsson, E., Hu, Y., & Rahman, M. (2013). Competing in the Age of Omnichannel Retailing. *MIT Sloan Management Review*, 54(4), 23–29.
- Burke, R. (2002a). Hybrid recommender systems: Survey and experiments. *User Modelling and User-Adapted Interaction*, 12(4), 331–370. <https://doi.org/10.1023/A:1021240730564/METRICS>
- Burke, R. (2002b). Hybrid recommender systems: Survey and experiments. *User Modelling and User-Adapted Interaction*, 12(4), 331–370. <https://doi.org/10.1023/A:1021240730564/METRICS>
- Fader, P. S., & Hardie, B. G. S. (2016). *Customer-Base Analysis: Using Predictive Analytics to Drive Growth*. University of Pennsylvania Press.
- Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12), 61–70. <https://doi.org/10.1145/138859.138867/ASSET/74169BF6-30C5-4F64-80AC-093322AB4CF5/ASSETS/138859.138867.FP.PNG>
- Hanley, J. A., & McNeil, B. J. (1982). The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. *Radiology*, 143(1), 29–36.
- He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017a). Neural collaborative filtering. *26th International World Wide Web Conference, WWW 2017*, 173–182. <https://doi.org/10.1145/3038912.3052569>



- He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017b). Neural collaborative filtering. *26th International World Wide Web Conference, WWW 2017*, 173–182. <https://doi.org/10.1145/3038912.3052569>
- Herlocker, J. L., Konstan, J. A., Borchers, A., & Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1999*, 230–237. <https://doi.org/10.1145/312624.312682/ASSET/B5021BF9-14D1-4A89-8876-877E8F0A7F91/ASSETS/312624.312682.FP.PNG>
- Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1), 5–53. <https://doi.org/10.1145/963770.963772>
- Hu, Y., Volinsky, C., & Koren, Y. (2008a). Collaborative filtering for implicit feedback datasets. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 263–272. <https://doi.org/10.1109/ICDM.2008.22>
- Hu, Y., Volinsky, C., & Koren, Y. (2008b). Collaborative filtering for implicit feedback datasets. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 263–272. <https://doi.org/10.1109/ICDM.2008.22>
- Hübner, A., Kuhn, H., & Wollenburg, J. (2016). Last mile fulfilment and distribution in omni-channel grocery retailing: A strategic planning framework. *International Journal of Retail and Distribution Management*, 44(3), 228–247. <https://doi.org/10.1108/IJRDM-11-2014-0154/FULL/XML>
- Koren, Y., Bell, R., & Volinsky, C. (2009a). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30–37. <https://doi.org/10.1109/MC.2009.263>
- Koren, Y., Bell, R., & Volinsky, C. (2009b). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30–37. <https://doi.org/10.1109/MC.2009.263>
- Koren, Y., Bell, R., & Volinsky, C. (2009c). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30–37. <https://doi.org/10.1109/MC.2009.263>

- Kouzes, R. T., Anderson, G. A., Elbert, S. T., Gorton, I., & Gracio, D. K. (2009). The changing paradigm of data-intensive computing. *Computer*, 42(1), 26–34. <https://doi.org/10.1109/MC.2009.26>
- Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1), 76–80. <https://doi.org/10.1109/MIC.2003.1167344>
- Lops, P., de Gemmis, M., & Semeraro, G. (2011). Content-based Recommender Systems: State of the Art and Trends. *Recommender Systems Handbook*, 73–105. [https://doi.org/10.1007/978-0-387-85820-3\\_3](https://doi.org/10.1007/978-0-387-85820-3_3)
- Nguyen, L. V., Vo, Q. T., & Nguyen, T. H. (2023). Adaptive KNN-Based Extended Collaborative Filtering Recommendation Services. *Big Data and Cognitive Computing 2023, Vol. 7, Page 106*, 7(2), 106. <https://doi.org/10.3390/BDCC7020106>
- Pazzani, M. J., & Billsus, D. (2007). Content-Based Recommendation Systems. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4321 LNCS, 325–341. [https://doi.org/10.1007/978-3-540-72079-9\\_10](https://doi.org/10.1007/978-3-540-72079-9_10)
- Pinela, C. (2017). *Recommender Systems — User-Based and Item-Based Collaborative Filtering* | by Carlos Pinela | Medium. <https://medium.com/@cfpinela/recommender-systems-user-based-and-item-based-collaborative-filtering-5d5f375a127f>
- Rendle, S. (2012). Factorization Machines with libFM. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3). <https://doi.org/10.1145/2168752.2168771>
- Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to Recommender Systems Handbook. *Recommender Systems Handbook*, 1–35. [https://doi.org/10.1007/978-0-387-85820-3\\_1](https://doi.org/10.1007/978-0-387-85820-3_1)
- Ricci, F., Shapira, B., & Rokach, L. (2015). Recommender Systems: Introduction and Challenges. *Recommender Systems Handbook, Second Edition*, 1–34. [https://doi.org/10.1007/978-1-4899-7637-6\\_1](https://doi.org/10.1007/978-1-4899-7637-6_1)
- Rigby, D. (2011). The Future of Shopping. *Harvard Business Review*, 89(12), 65–76.
- Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001a). Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th International Conference on World Wide Web, WWW 2001*, 285–295. <https://doi.org/10.1145/371920.372071>

- Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001b). Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th International Conference on World Wide Web, WWW 2001*, 285–295. <https://doi.org/10.1145/371920.372071/ASSET/CFB8B952-6F16-43A6-8125-16F950D0D3E3/ASSETS/371920.372071.FP.PNG>
- Schafer, J. Ben, Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative Filtering Recommender Systems. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4321 LNCS, 291–324. [https://doi.org/10.1007/978-3-540-72079-9\\_9](https://doi.org/10.1007/978-3-540-72079-9_9)
- Verhoef, P. C., Kannan, P. K., & Inman, J. J. (2015). From Multi-Channel Retailing to Omni-Channel Retailing. Introduction to the Special Issue on Multi-Channel Retailing. *Journal of Retailing*, 91(2), 174–181. <https://doi.org/10.1016/J.JRETAI.2015.02.005>
- Wang, Z., Wang, W., Wang, C., Gu, X., Li, B., & Meng, D. (2019). Community Focusing: Yet Another Query-Dependent Community Detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 329–337.