

Diseño e implementación de motor de afinidad para personalización comercial B2B en consumo masivo

Lic. Abril Noguera

Carrera de Especialización en Inteligencia Artificial

Director: Ing. Juan Pablo Rodríguez Varela (ITBA)

Jurados:

MSc. Lautaro González (UTDT)
Esp. Ing. Carlos David Rodríguez Sánchez (FIUBA)
Esp. Ing. Félix Alejandro Guglielmi Parra (UNLP)

Resumen

En la presente memoria se describe el diseño e implementación de un motor de afinidad orientado a la personalización comercial en un entorno de negocio del sector de consumo masivo. Se desarrolló un sistema de recomendación capaz de estimar la relevancia de cada producto para cada cliente a partir de datos transaccionales, señales digitales y características contextuales, con el objetivo de generar listas priorizadas de sugerencias.

Para su desarrollo fueron fundamentales los conocimientos adquiridos en la carrera, tales como aprendizaje automático, aprendizaje profundo, validación de modelos, ingeniería de atributos y prácticas de MLOps para la trazabilidad y despliegue del sistema.

Índice general

Resumen	I
1. Introducción general	1
1.1. Marco de la propuesta	1
1.2. Definición del problema	2
1.3. Estado del arte	4
1.3.1. Referencias en sistemas de recomendación	4
1.3.2. Sistemas de recomendación en B2B	4
1.3.3. Caso de implementación	5
1.3.4. Lecciones aprendidas	5
1.4. Motivación	6
1.5. Objetivos y alcance	7
2. Introducción específica	9
2.1. Sistemas de recomendación	9
2.1.1. Funcionamiento de los sistemas de recomendación	9
2.1.2. Tipos de <i>feedback</i>	10
2.1.3. Filtrado colaborativo	10
2.1.4. Sistemas basados en contenido	11
2.2. Fuentes de información	11
2.3. Herramientas utilizadas	12
2.3.1. Plataformas de procesamiento distribuido	12
2.3.2. Gestión del ciclo de vida de modelos	12
2.3.3. Bibliotecas de aprendizaje automático y profundo	12
2.3.4. Bibliotecas de visualización	13
2.3.5. Control de versiones y colaboración	13
2.3.6. Consideraciones finales	13
3. Diseño e implementación	15
3.1. Diseño de solución	15
3.2. Análisis exploratorio de los datos	16
3.2.1. Curvas de concentración de clientes y productos	16
3.2.2. Patrones de diversidad en el portafolio	18
3.2.3. Correlaciones entre variables transaccionales y digitales	20
3.2.4. Observaciones preliminares del análisis exploratorio	21
3.3. Preparación e ingeniería de los datos	22
3.3.1. Fuentes y estructura general	22
3.3.2. Construcción de la matriz cliente-producto	23
3.3.3. Diseño de atributos de cliente y producto	24
Evaluación de representatividad	24
3.4. Desarrollo de modelos	25
3.4.1. Filtrado colaborativo con ALS	26

Configuración del modelo	26
Optimización de hiperparámetros	26
Resultados y conclusiones	27
3.4.2. Modelo híbrido con LightFM	27
Evaluación experimental del modelo LightFM	28
Optimización bayesiana de hiperparámetros	29
Resultados y conclusiones	29
3.4.3. Modelo basado en contenido con arquitectura Two-Tower	29
Diseño y configuración del modelo	30
Ensamble híbrido con ALS	31
Resultados y conclusiones	31
3.4.4. <i>Neural Collaborative Filtering</i>	31
Diseño y configuración	32
Resultados y conclusiones	33
3.5. Implementación	33
3.5.1. Diseño del <i>pipeline</i> de procesamiento	33
3.5.2. Integración con la infraestructura tecnológica	33
3.5.3. Estrategias de versionado y monitoreo	34
4. Ensayos y resultados	35
4.1. Metodología de evaluación	35
4.2. Modelos <i>baseline</i>	37
4.2.1. Modelo nulo	37
4.2.2. Modelo aleatorio	37
4.2.3. Modelo basado en reglas	38
4.2.4. Resultados de los modelos <i>baseline</i>	38
4.2.5. Comparación global de desempeño frente a los <i>baseline</i>	38
4.3. Análisis de resultados	40
4.3.1. Resultados de performance	40
4.3.2. Resultados de eficiencia computacional	41
4.3.3. Evaluación final del desempeño y eficiencia	42
4.4. Análisis de robustez	42
5. Conclusiones	45
5.1. Conclusiones generales	45
5.2. Próximos pasos	46
A. Optimización de pesos por evento y ventana temporal	49
A.1. Estrategia de optimización	49
A.2. Pesos óptimos por ventana temporal	49
A.3. Pesos óptimos por tipo de evento	50
A.4. Análisis e interpretación	50
B. Ensayos experimentales del modelo LightFM	53
B.1. <i>Test 1</i> – Modelo base con contexto categórico reducido	53
B.2. <i>Test 2</i> – Modelo con selección explicativa de atributos discretizados	54
B.3. <i>Test 3</i> – Análisis y depuración de representaciones latentes	55
B.4. Resultados comparativos de los ensayos experimentales	56
Bibliografía	59

Índice de figuras

2.1. Ejemplo de representación de marcas de cerveza en un espacio de atributos	10
3.1. Arquitectura de alto nivel del sistema de recomendación.	16
3.2. Concentración de productos en el portafolio.	17
3.3. Concentración de clientes.	17
3.4. Histograma de diversidad de portafolio: número de productos distintos por cliente.	18
3.5. <i>Log log plot</i> de la popularidad de productos.	19
3.6. Mapa de calor de co-ocurrencia entre los 10 productos más relevantes.	19
3.7. Matriz de correlación entre variables transaccionales y digitales. . .	20
3.8. Correlación de variables con la compra en el mes siguiente.	21
3.9. Proyecciones PCA de clientes y productos	25
3.10. Arquitectura del modelo <i>Two Towers</i> ¹	30
3.11. Arquitectura del modelo <i>Neural Collaborative Filtering</i> híbrido ²	32
4.1. Comparación de Precision@10 frente a modelos baseline	39
4.2. Comparación de Recall@10 frente a modelos baseline	39
4.3. Distribución del <i>affinity score</i> por mes	43

Índice de tablas

1.1. Ventajas y desventajas de enfoques en recomendación	6
1.2. caption corto	6
3.1. Tasa de recompra por combinación de señales	21
3.2. Métricas descriptivas de la matriz cliente–producto	24
3.3. Resultados comparativos de los ensayos con LightFM	28
4.1. Métricas de performance	36
4.2. Métricas de eficiencia computacional	37
4.3. Resultados comparativos de los modelos <i>baseline</i>	38
4.4. Desempeño comparado de los modelos avanzados	40
4.5. Resumen de eficiencia computacional	41
4.6. Estabilidad temporal del desempeño	43
4.7. Desempeño por unidad de negocio	43
4.8. Desempeño por clasificación de cliente	44
A.1. Pesos óptimos por ventana temporal	50
A.2. Pesos óptimos por tipo de evento	50
B.1. Resultados comparativos de los ensayos con LightFM	57

Dedicado a... [OPCIONAL]

Capítulo 1

Introducción general

Este capítulo tiene como propósito contextualizar el trabajo dentro del ámbito del consumo masivo y, en particular, de los modelos de negocio entre empresas (B2B). Se expone la relevancia que adquiere la personalización comercial en este sector y los desafíos que surgen al gestionar un portafolio amplio de productos frente a una base heterogénea de clientes. A partir de esta perspectiva se describe el problema central que motiva el desarrollo de un motor de afinidad y se señalan las limitaciones de los enfoques tradicionales de recomendación en entornos de alta variabilidad y escasez de datos.

Asimismo, se realiza una revisión introductoria de los principales sistemas de recomendación y de sus alcances en diferentes contextos, donde se destacan las particularidades que distinguen al escenario de negocio entre empresas. Finalmente, se presentan la motivación, la relevancia y los objetivos del trabajo, con el fin de ofrecer al lector una visión clara del problema abordado, de la importancia de su resolución y del recorrido que seguirá la memoria en los capítulos posteriores.

1.1. Marco de la propuesta

La industria del consumo masivo constituye uno de los motores más importantes de la economía, caracterizada por un volumen elevado de transacciones, la alta frecuencia de compra y la amplia variedad de productos que la conforman. La magnitud de este sector, junto con la fuerte competencia existente, obliga a las compañías a buscar permanentemente mecanismos que les permitan diferenciarse y mejorar la relación con sus clientes.

En este entorno, la relación comercial se establece entre una empresa proveedora y una red extensa de clientes minoristas que funcionan como canales de llegada al consumidor final. Estos clientes presentan una gran diversidad en cuanto a tamaño, ubicación geográfica, recursos disponibles y patrones de demanda. La heterogeneidad de la red de distribución genera que cada establecimiento tenga necesidades distintas y reaccione de manera diferente frente a la oferta de productos. Bajo estas condiciones, una estrategia comercial homogénea resulta insuficiente, ya que no logra capturar las particularidades de cada cliente ni ofrecerle productos que se ajusten de manera adecuada a su realidad.

La necesidad de personalización surge entonces como un factor estratégico central. Adaptar la oferta a las características específicas de cada cliente no solo incrementa la probabilidad de aceptación de los productos sugeridos, sino que también permite optimizar el uso del canal comercial, fortalecer la relación de largo

plazo y generar un impacto positivo en la eficiencia general del negocio. Las sugerencias ajustadas al contexto trascienden la idea de recomendar lo más vendido en términos absolutos: implica comprender la dinámica particular de cada cliente y priorizar aquellos productos que, dentro de un portafolio amplio, resulten más relevantes para su operación cotidiana.

A esta diversidad se suman factores que aumentan la complejidad del sector. La estacionalidad en la demanda, la influencia de promociones y campañas comerciales, la variabilidad en las preferencias de los consumidores finales y la constante rotación de productos dentro del catálogo configuran un escenario cambiante y difícil de predecir. La incorporación de artículos nuevos en el portafolio plantea, además, el desafío de la falta de contexto e información histórica que da perspectiva para guiar las recomendaciones.

En este marco, contar con herramientas que permitan personalizar la relación con cada cliente resulta indispensable. Un sistema capaz de priorizar los productos más relevantes para cada establecimiento aporta ventajas significativas: mejora la precisión de las recomendaciones, amplía la visibilidad de productos estratégicos, optimiza la gestión de los recursos comerciales y contribuye a consolidar vínculos más sólidos con los clientes minoristas. De esta manera, las recomendaciones a medida se convierten en un pilar fundamental para la sostenibilidad y la competitividad en el sector del consumo masivo.

1.2. Definición del problema

La empresa en la que se desarrolla este trabajo pertenece al sector del consumo masivo y opera bajo un modelo de venta directa a una red amplia y heterogénea de clientes minoristas. Esta red está compuesta por autoservicios, kioscos y comercios tradicionales distribuidos en todo el territorio nacional, lo que permite alcanzar una cobertura superior a los trescientos mil puntos de venta. La magnitud de esta operación, sumada a la diversidad de formatos y capacidades de los clientes, convierte a la personalización en una necesidad estratégica. A ello se suma la complejidad de un portafolio que incluye un gran número de marcas y presentaciones, lo que multiplica las posibles combinaciones cliente–producto y genera un desafío de gestión a gran escala.

El reto principal radica en estimar con precisión el interés que cada cliente podría tener en cada producto dentro del portafolio. Hoy en día, las decisiones comerciales se apoyan principalmente en el historial de ventas o en la popularidad general de los artículos, lo que conduce a una oferta relativamente homogénea. Este enfoque ignora las particularidades de los clientes y no captura la relevancia contextual de los productos. El problema central se expresa, entonces, en la ausencia de mecanismos que permitan calcular un nivel de afinidad entre cliente y producto capaz de reflejar con realismo el grado de interés que un artículo puede despertar en un punto de venta específico en un momento determinado.

Este desafío se ve amplificado por una serie de batallas que la empresa enfrenta de manera cotidiana en su estrategia comercial. La primera de ellas es la necesidad de pasar de un enfoque reactivo, basado en compras históricas, hacia una estrategia proactiva que permita anticipar tendencias de consumo y orientar la oferta en consecuencia. Para ello es indispensable contar con una herramienta

que adapte las recomendaciones de manera dinámica y alineada con el comportamiento observado en cada cliente.

Otra dimensión crítica es la optimización de recursos. La magnitud de la red comercial hace imposible abordar a todos los clientes con la misma intensidad, por lo que resulta fundamental identificar en qué productos y clientes concentrar los esfuerzos. Un motor de afinidad que jerarquice oportunidades de mayor impacto ofrece al equipo comercial la posibilidad de planificar visitas y diseñar ofertas más focalizadas, lo que mejora la eficiencia del canal.

La constante rotación del portafolio también representa un desafío de gran magnitud. Una proporción significativa de los productos se renueva cada año, lo que obliga a dar visibilidad a artículos sin historial de ventas y, al mismo tiempo, sostener el desempeño de categorías tradicionales. Este problema de arranque en frío limita la capacidad de los enfoques tradicionales para recomendar productos nuevos o poco frecuentes, lo que retrasa su incorporación en los puntos de venta y afecta el posicionamiento de la innovación en el mercado.

De manera similar, la inserción de nuevos clientes en la red sin historial de compras constituye un reto adicional. Cada semana se incorporan comercios que aún no cuentan con registros transaccionales suficientes para perfilar sus preferencias. Estos clientes suelen recibir sugerencias genéricas o basadas en promedios de segmentos, lo que reduce el atractivo de la oferta inicial y dificulta su integración temprana al canal digital. Una solución efectiva debería ser capaz de recomendar productos relevantes aun en ausencia de historial, al aprovechar señales contextuales y patrones de clientes similares.

La estacionalidad y las promociones constituyen otro factor de complejidad. La demanda de determinados productos fluctúa de manera pronunciada según la época del año o las campañas comerciales en curso. Un producto que en un período presenta alta relevancia puede perder vigencia en el siguiente, lo que provoca que reglas estáticas de recomendación queden rápidamente obsoletas. Para sostener la efectividad en este entorno dinámico se requiere un sistema flexible y capaz de adaptarse a variaciones temporales.

En conjunto, estos factores configuran un escenario donde la falta de personalización impacta de manera directa en los resultados del negocio. Sin un mecanismo que integre de manera sistemática los datos disponibles, se generan listas de productos poco relevantes para los clientes, se desperdician oportunidades de venta cruzada y se dificulta la adopción de innovaciones. Asimismo, el equipo comercial se ve limitado por información fragmentada, lo que reduce su capacidad de diseñar acciones específicas y de extraer valor de la gran cantidad de datos generados en el canal digital.

La solución propuesta apunta a superar estas limitaciones mediante el desarrollo de un motor de afinidad que calcule de forma periódica la relevancia de cada producto para cada cliente, y que integra señales transaccionales, interacciones digitales y atributos contextuales. Este motor tiene como objetivo generar rankings personalizados que orienten las recomendaciones tanto en el canal digital como en la gestión directa del equipo comercial. De esta forma, se busca avanzar hacia una estrategia más precisa, escalable y alineada con los objetivos de negocio, lo que habilita una gestión proactiva de portafolio y mejora la relación con los clientes de la red.

1.3. Estado del arte

El estado del arte permite ubicar este trabajo dentro de la evolución de los sistemas de recomendación. En esta sección se revisan los principales *benchmarks* en entornos B2C (del inglés, *Business to Customer*), los aportes de la literatura en contextos B2B y un caso de implementación en Brasil, para finalmente sintetizar los aprendizajes y señalar la brecha que orienta esta propuesta.

1.3.1. Referencias en sistemas de recomendación

El campo de los sistemas de recomendación se consolidó en los últimos veinte años como una de las áreas más dinámicas dentro de la inteligencia artificial aplicada. Sus desarrollos se originaron en entornos de consumo directo al público, donde el volumen de usuarios y la abundancia de señales digitales permitieron mejorar rápidamente la precisión y escalabilidad. A lo largo de este proceso, distintos hitos se transformaron en referencias obligadas y definieron *benchmarks* de la disciplina.

Uno de los puntos de inflexión fue el concurso Netflix Prize [1], que impulsó avances en factorización matricial y consolidó métricas de ranking como *recall* y *precision* en el análisis de desempeño. En paralelo, Amazon desarrolló un motor de recomendaciones basado en filtrado colaborativo *item-to-item*, reconocido por su capacidad de escalar en catálogos extensos y mantener robustez frente a grandes volúmenes de transacciones. MovieLens [2] se transformó en el dataset académico más utilizado, al servir como estándar para comparar algoritmos y validar resultados de manera consistente. Finalmente, plataformas como Spotify y YouTube llevaron la disciplina hacia modelos secuenciales y de aprendizaje profundo, capaces de personalizar en tiempo real a partir de interacciones en sesiones cortas.

Estos casos muestran cómo los sistemas de recomendación se convirtieron en el núcleo de la personalización digital y establecieron estándares en cuanto a precisión, escalabilidad y diversidad. Al mismo tiempo, reflejan un sesgo hacia contextos de B2C, donde las interacciones con consumidores finales son abundantes, explícitas y fácilmente trazables.

1.3.2. Sistemas de recomendación en B2B

En entornos de negocio entre empresas, la adopción de sistemas de recomendación es mucho más incipiente. La literatura identifica que, a diferencia de lo que ocurre en B2C, los procesos de compra en B2B suelen involucrar múltiples actores, ciclos de decisión más largos y una relación de largo plazo entre proveedor y cliente. Estas particularidades hacen que las soluciones desarrolladas para consumo final no se trasladen de forma directa.

El estudio presentado en [3] resalta el potencial de estas herramientas en B2B, al destacar que pueden reducir los costos de búsqueda, fortalecer vínculos comerciales y facilitar la introducción de productos en portafolios complejos. Sin embargo, también identifica desafíos clave: la necesidad de integrar datos dispersos de distintas fuentes, la importancia de la interpretabilidad para ganar confianza en decisiones de compra de alto valor y la dificultad de escalar modelos en contextos de menor densidad transaccional.

Si bien existe un reconocimiento académico del valor que los sistemas de recomendación pueden aportar en B2B, las implementaciones concretas son todavía escasas y carecen de estandarización. Esto genera una brecha significativa entre el potencial identificado y la práctica real, que representa una oportunidad de innovación para sectores como el consumo masivo.

1.3.3. Caso de implementación

Un antecedente particularmente relevante proviene de la propia organización, a través de la implementación de un sistema de recomendación en Brasil dentro de la plataforma digital BEES [4]. Este desarrollo tuvo como objetivo priorizar productos para cada punto de venta a gran escala, con el fin de reemplazar procesos manuales que en el pasado se realizaban en planillas y que resultaban poco eficientes.

El algoritmo principal implementado fue un filtrado colaborativo para feedback implícito, concretado mediante factorización matricial con el método *Alternating Least Squares (ALS)*. El modelo utilizó como insumos tanto el historial de compras como señales digitales generadas en la aplicación, e incluyó búsquedas, visualizaciones de productos e interacciones con el carrito de compras. De este modo, se logró reducir sustancialmente la cantidad de recomendaciones enfocándolas en productos con mayor interés para el cliente, lo que marcó un avance significativo en la capacidad de personalizar la oferta a cada punto de venta.

Los resultados demostraron la viabilidad de este tipo de soluciones en un entorno B2B real y de gran escala. Sin embargo, también dejaron en evidencia limitaciones relevantes. La dependencia casi exclusiva del historial transaccional reforzó el problema del arranque en frío, tanto para productos recién incorporados como para clientes nuevos sin registros suficientes. Además, el sistema presentó limitaciones en diversidad de recomendaciones, ya que tendía a reforzar productos populares, y careció de un componente explícito para alinear los resultados con prioridades estratégicas de negocio.

El mismo documento identifica líneas de mejora hacia el futuro, como la incorporación de modelos híbridos que integren atributos de clientes y productos, el desarrollo de algoritmos de *clustering* para agrupar unidades de negocio con características similares y la inclusión de mecanismos que permitan diversificar resultados. Estas observaciones resultan especialmente valiosas para orientar el diseño de una solución adaptada al contexto argentino.

1.3.4. Lecciones aprendidas

El recorrido presentado permite extraer tres conclusiones principales. En primer lugar, los benchmarks internacionales muestran que los sistemas de recomendación son capaces de transformar industrias enteras cuando logran combinar precisión, escalabilidad y diversidad. En segundo lugar, la literatura sobre B2B reconoce la oportunidad de trasladar estos beneficios, pero también evidencia la falta de soluciones maduras que contemplan las particularidades de este tipo de relaciones comerciales. Finalmente, el caso de Brasil demuestra que es posible implementar un motor de recomendaciones en un contexto de consumo masivo B2B, pero también que persisten limitaciones en arranque en frío, diversidad y alineación con objetivos de negocio.

A modo de síntesis, la tabla 1.1 resume las ventajas y desventajas de cada uno de los enfoques revisados, e incluye la brecha identificada en el contexto argentino que motiva el desarrollo de un motor de afinidad adaptado a la realidad local. Este resumen permite enfatizar la necesidad de avanzar hacia un sistema que integre señales transaccionales y digitales, incorpore criterios estratégicos de negocio y se apoye en técnicas modernas de aprendizaje automático y profundo. El objetivo es superar las restricciones de los enfoques tradicionales y aportar un valor diferencial en la gestión comercial de la empresa en Argentina.

TABLA 1.1. Ventajas y desventajas de los enfoques revisados.

Enfoque / Caso	Ventajas principales	Desventajas principales
Benchmarks B2C (Netflix, Amazon, etc.)	Alta precisión y escalabilidad. Abundancia de datos y señales digitales. Estándares de evaluación consolidados.	Contextos con abundancia de <i>feedback</i> explícito/implícito, poco comparables al B2B. No consideran objetivos de negocio específicos.
Literatura B2B	Reconoce particularidades de clientes empresariales. Identifica beneficios en reducción de costos y fortalecimiento de relaciones.	Pocas implementaciones reales. Escasa estandarización de métricas y datasets. Desafíos de interpretabilidad y escalabilidad.
Caso Brasil (BEES)	Demostró viabilidad en gran escala. Integró compras e interacciones digitales. Mejora clara frente a procesos manuales.	Dependencia fuerte del histórico transaccional (arranque en frío). Limitaciones en diversidad y alineación con objetivos estratégicos.
Brecha en Argentina	Oportunidad de adaptar aprendizajes globales y regionales. Potencial de integrar señales contextuales y digitales. Aplicación de técnicas modernas de aprendizaje automático y profundo.	Falta de solución probada en el contexto local. Mayor heterogeneidad y escala que en otros países.

TABLA 1.2. caption largo más descriptivo.

Especie	Tamaño	Valor
Amphiprion Ocellaris	10 cm	\$ 6.000
Hepatus Blue Tang	15 cm	\$ 7.000
Zebrasoma Xanthurus	12 cm	\$ 6.800

1.4. Motivación

La definición del problema mostró que la empresa enfrenta limitaciones para identificar con precisión qué productos resultan más relevantes para cada cliente

en cada momento, debido a factores como la rotación del portafolio, la estacionalidad de la demanda y la incorporación de nuevos clientes sin historial. El estado del arte, por su parte, evidencia que si bien existen avances notables en sistemas de recomendación y casos aplicados en entornos B2C, aún persiste una brecha en cuanto a soluciones robustas y adaptadas a escenarios B2B de consumo masivo.

La motivación de este trabajo surge de esa intersección: un problema claramente identificado en la operación local y un campo de conocimiento que ofrece enfoques valiosos pero todavía insuficientes para resolverlo en toda su complejidad. El diferencial de esta propuesta reside en integrar múltiples fuentes de información, transaccionales, digitales y contextuales, dentro de un motor de afinidad diseñado específicamente para el mercado argentino. Además, el trabajo incorpora la orientación explícita a objetivos de negocio y el uso de prácticas modernas de aprendizaje automático, aprendizaje profundo y MLOps, con el fin de garantizar escalabilidad, trazabilidad y alineación estratégica.

En este sentido, el trabajo no busca reproducir soluciones existentes, sino avanzar hacia un sistema que combine la rigurosidad técnica con la aplicabilidad práctica en un contexto desafiante, y que aporte un valor diferencial tanto en la gestión comercial de la empresa como en la evolución del conocimiento sobre sistemas de recomendación en consumo masivo B2B.

1.5. Objetivos y alcance

El propósito general de este trabajo es desarrollar un motor de afinidad que permita generar recomendaciones personalizadas de productos para cada cliente de la red de la empresa. El sistema se plantea como una herramienta capaz de integrar información transaccional, señales digitales y atributos contextuales con el fin de optimizar la gestión comercial, mejorar la efectividad de las sugerencias y facilitar la adopción de categorías estratégicas.

A partir de este objetivo general se desprenden metas específicas que orientan el desarrollo. En primer lugar, se busca analizar en detalle las fuentes de datos disponibles y transformarlas en insumos útiles para el modelado. Sobre esta base, se plantea la construcción de variables que reflejen el comportamiento de compra, las características de los productos y el contexto de cada cliente. Un segundo objetivo es implementar y comparar distintos enfoques de modelado, desde métodos de referencia hasta técnicas de factorización, modelos híbridos y arquitecturas profundas, con el objetivo de evaluar su desempeño con métricas de ranking como *recall@K*, *MAP@K*, cobertura y diversidad. De manera complementaria, se incluye la necesidad de diseñar estrategias que permitan afrontar el arranque en frío, tanto de productos recién incorporados al portafolio como de clientes nuevos sin historial de compras. Finalmente, se busca establecer un pipeline de entrenamiento y despliegue con prácticas de MLOps que garantice trazabilidad, reproducibilidad y escalabilidad del sistema.

El alcance del trabajo se limita a la construcción y evaluación de un prototipo funcional en un entorno controlado con datos reales de la empresa. Esto implica el análisis y preparación de la información, el desarrollo de modelos de recomendación y la evaluación de su desempeño a través de métricas definidas, e incluye escenarios de robustez frente a la incorporación de productos y clientes nuevos.

También, se contempla el diseño conceptual de la integración del motor con el canal digital y el apoyo al trabajo del equipo comercial.

Capítulo 2

Introducción específica

Este capítulo presenta los conceptos y componentes centrales que sustentan el trabajo. Se introducen los sistemas de recomendación y sus enfoques principales, se describen las fuentes de información empleadas y se detallan las plataformas y herramientas utilizadas para el procesamiento de datos, el modelado y la gestión de experimentos, que conforman la base tecnológica de la solución propuesta.

2.1. Sistemas de recomendación

Los sistemas de recomendación constituyen una de las aplicaciones más extensas de la inteligencia artificial [5, 6], con un papel central en la reducción de la sobrecarga de información y en la optimización de decisiones de consumo. Su finalidad es generar sugerencias personalizadas que se ajusten a las características de cada cliente, lo que incrementa la relevancia de los productos ofrecidos y mejora la experiencia general de interacción con la empresa.

2.1.1. Funcionamiento de los sistemas de recomendación

El eje central del enfoque consiste en identificar relaciones de similitud entre productos, clientes o interacciones [5]. Estas relaciones pueden establecerse desde diferentes perspectivas. En primer lugar, es posible medir la similitud entre productos, lo que permite agrupar aquellos que suelen adquirirse en conjunto o que comparten atributos comunes. En segundo lugar, puede analizarse la similitud entre clientes, de modo que las preferencias observadas en un grupo con comportamientos semejantes permitan anticipar las elecciones de otros con perfiles cercanos. Por último, también resulta clave la similitud entre interacciones, que considera el historial de comportamientos de un cliente, como sus compras o búsquedas, para anticipar futuras decisiones.

Un ejemplo ilustrativo, representado en la figura 2.1, puede plantearse en la industria de bebidas. Supongamos que cada marca de cerveza se representa como un vector en un espacio definido por atributos, como tradicional versus innovador y masivo versus *premium*. En ese espacio, una lager clásica de gran consumo quedaría ubicada cerca de otras variedades tradicionales y de alcance masivo, mientras que una IPA artesanal o una edición limitada se situarían en la región asociada a lo *premium* e innovador. El sistema de recomendación aprovecha esta representación para calcular distancias o similaridades entre productos. Si un cliente suele elegir artículos situados en torno al cuadrante de *premium-tradicional*, el modelo infiere que probablemente muestre interés

por otras marcas que ocupan posiciones cercanas en ese mismo espacio vectorial. De esta manera, la proximidad entre vectores se convierte en un indicador de afinidad, que guía la generación de recomendaciones personalizadas.

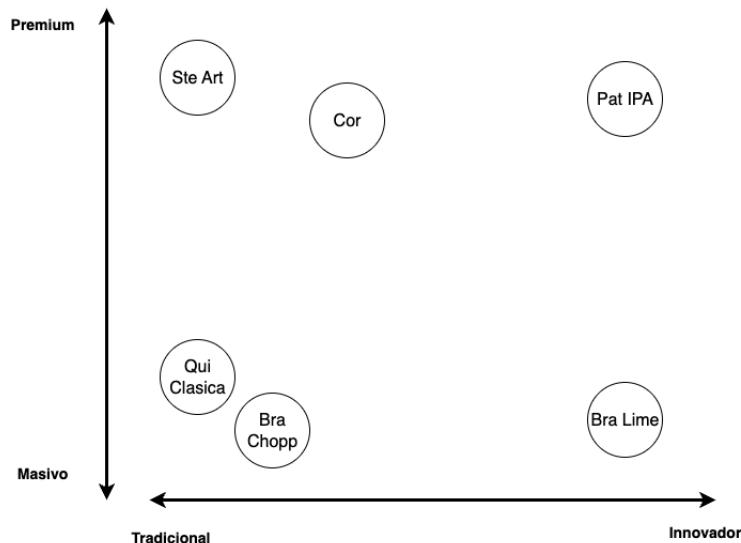


FIGURA 2.1. Ejemplo de representación de marcas de cerveza en un espacio de atributos.

2.1.2. Tipos de *feedback*

El tipo de información disponible para alimentar un sistema de recomendación también es determinante. Se distinguen dos formas principales de retroalimentación [7]. La retroalimentación explícita consiste en la valoración directa que realizan los clientes sobre los productos, como calificaciones numéricas, encuestas o reseñas. La retroalimentación implícita, en cambio, se infiere del comportamiento de los clientes, ya sea a través de sus compras, búsquedas o interacciones digitales. En el ámbito B2B, donde no es común que los clientes asignen calificaciones explícitas, predominan las señales implícitas, lo que plantea desafíos adicionales para la construcción de modelos precisos.

2.1.3. Filtrado colaborativo

El filtrado colaborativo se apoya en la hipótesis de que usuarios similares tienden a preferir ítems similares, lo que puede abordarse mediante enfoques *user-based* o *item-based* [8]. Su implementación moderna se basa en factorización matricial [9].

En la modalidad *user-based*, se recomienda a un cliente productos que fueron consumidos por otros con patrones de compra semejantes. En la modalidad *item-based*, se priorizan productos que suelen aparecer en conjunto en los historiales de distintos clientes.

El filtrado colaborativo suele implementarse mediante técnicas de factorización matricial. Dado un conjunto de m usuarios y n productos, se construye una matriz de interacciones $R \in \mathbb{R}^{m \times n}$, donde cada celda refleja el vínculo entre un cliente y un producto. El objetivo consiste en aproximar esta matriz como el producto de dos matrices de menor dimensión, como se puede observar en 2.1.

$$R \approx U \cdot V^T \quad (2.1)$$

donde $U \in \mathbb{R}^{m \times k}$ representa a los usuarios en un espacio latente de dimensión k , y $V \in \mathbb{R}^{n \times k}$ representa a los ítems en ese mismo espacio. La predicción de la afinidad del usuario i con el ítem j se calcula en 2.2 como el producto escalar entre los vectores latentes correspondientes.

$$\hat{r}_{ij} = u_i \cdot v_j^T \quad (2.2)$$

Este modelo permite capturar relaciones complejas entre clientes y productos a partir de información implícita, aunque presenta limitaciones frente al problema del arranque en frío, cuando no existe historial suficiente de interacciones.

2.1.4. Sistemas basados en contenido

Otro enfoque ampliamente utilizado es el de los sistemas basados en contenido, que centran la recomendación en las características de los productos y en el perfil de cada cliente [10]. En este caso, se representa a cada producto por un vector de atributos y se construye un perfil para cada cliente que refleja la importancia relativa de esos atributos en función de sus elecciones pasadas. La predicción de relevancia para recomendar un producto j a un cliente i puede expresarse de manera simplificada como en 2.3.

$$\hat{r}_{ij} = w_i \cdot x_j \quad (2.3)$$

donde x_j es el vector de atributos del producto y w_i representa el perfil del cliente. Este método permite recomendar productos nuevos o poco frecuentes siempre que exista información suficiente sobre sus atributos, lo que lo convierte en un complemento natural del filtrado colaborativo.

2.2. Fuentes de información

El funcionamiento de los sistemas de recomendación se apoya en diversas fuentes de información [5, 6] que, al combinarse, permiten construir una representación más completa de la relación entre usuarios y productos. Estas fuentes pueden clasificarse en tres grandes categorías: datos transaccionales, señales de interacción y atributos contextuales.

Los datos transaccionales reflejan las operaciones efectivamente realizadas, como compras, alquileres o reproducciones. Constituyen una evidencia directa de preferencia, ya que expresan decisiones concretas de los usuarios respecto a determinados productos o servicios.

Las señales de interacción incluyen registros de comportamiento que no necesariamente culminan en una transacción, pero que aportan información implícita de interés. Las señales implícitas, como visualizaciones o clics, resultan especialmente valiosas en contextos digitales [7, 11]. Ejemplos de este tipo de datos son las

visualizaciones de fichas de producto, las búsquedas realizadas en una plataforma, las adiciones y eliminaciones en un carrito digital o las calificaciones otorgadas. Estas interacciones permiten identificar patrones de consideración más allá de la compra final.

Finalmente, los atributos contextuales corresponden a características adicionales tanto de los usuarios como de los productos. Del lado de los usuarios, se pueden incluir variables demográficas, geográficas o vinculadas al canal de consumo. Del lado de los productos, se consideran atributos como categoría, marca, segmento o características técnicas. Este conjunto de información enriquece la representación de afinidad, al capturar heterogeneidades que condicionan las recomendaciones.

De esta forma, la integración de datos transaccionales, señales de interacción y atributos contextuales constituye la base informativa sobre la cual se construyen los diferentes enfoques de recomendación. La disponibilidad y calidad de estas fuentes son determinantes para el desempeño de los modelos y para la capacidad de generar sugerencias precisas y relevantes.

2.3. Herramientas utilizadas

El desarrollo de este trabajo se apoyó en un conjunto de herramientas tecnológicas que facilitaron la gestión integral del ciclo de vida del sistema de recomendación. A continuación, se detallan las principales plataformas, bibliotecas y entornos empleados, junto con su función específica en el proceso.

2.3.1. Plataformas de procesamiento distribuido

El procesamiento y consolidación de grandes volúmenes de información se llevó a cabo en la plataforma Databricks [12], que integra un entorno colaborativo con un motor de cómputo distribuido basado en Apache Spark [13]. Esta herramienta permitió orquestar la ingestión de datos, ejecutar transformaciones a gran escala mediante PySpark y garantizar reproducibilidad en los flujos de trabajo. El uso de Databricks resultó fundamental para integrar múltiples fuentes y preparar los insumos que alimentaron las etapas de análisis y modelado.

2.3.2. Gestión del ciclo de vida de modelos

Para la gestión del ciclo de vida de los modelos se empleó MLflow [14], plataforma que facilita el registro de experimentos, parámetros, métricas y versiones de modelos. Esta herramienta permitió mantener trazabilidad entre las distintas ejecuciones, asegurar comparabilidad de resultados y almacenar los artefactos generados (modelos entrenados y estructuras derivadas). De este modo, se consolidó un repositorio ordenado que garantizó reproducibilidad y control en la experimentación.

2.3.3. Bibliotecas de aprendizaje automático y profundo

En el desarrollo de los modelos se emplearon distintas bibliotecas que constituyen estándares en la comunidad científica y profesional. Se utilizó MLLib para

implementar el filtrado colaborativo mediante factorización matricial con el algoritmo Alternating Least Squares [15], mientras que LightFM [16] permitió construir un modelo híbrido que combina señales de interacción implícita con atributos de clientes y productos. Adicionalmente, se recurrió a PyTorch como entorno de deep learning para el diseño de arquitecturas neuronales capaces de capturar relaciones no lineales y complejas en los datos.

2.3.4. Bibliotecas de visualización

Para el análisis visual y la generación de gráficos se utilizaron bibliotecas como Matplotlib [17] y Seaborn [18], que facilitaron la representación gráfica tanto de la información explorada como de los resultados obtenidos en las distintas fases del trabajo.

2.3.5. Control de versiones y colaboración

La organización y versionado del código se gestionaron mediante GitHub [19], que permitió estructurar los repositorios, registrar cambios de manera sistemática y facilitar la colaboración. El uso de esta plataforma aseguró orden en el desarrollo, trazabilidad de modificaciones y una integración eficiente de los distintos componentes del sistema.

2.3.6. Consideraciones finales

En conjunto, estas herramientas brindaron una infraestructura robusta para abordar todas las etapas del desarrollo del sistema de recomendación, desde la preparación de los datos hasta la evaluación y almacenamiento de modelos. Cabe destacar que la calidad de una implementación no depende únicamente del algoritmo utilizado, sino también de la solidez del entorno técnico que la respalda. El uso articulado de estas herramientas permitió asegurar la reproducibilidad de los resultados, la eficiencia en el manejo de datos, la trazabilidad de las decisiones y la escalabilidad del sistema desarrollado. En el contexto de una solución real, contar con esta base técnica resulta clave para garantizar tanto la calidad técnica como la posibilidad de evolución futura del sistema.

Capítulo 3

Diseño e implementación

Este capítulo aborda el proceso de construcción del sistema de recomendación, desde la concepción de la solución hasta su materialización en un entorno operativo. Se presentan las principales decisiones de diseño, el tratamiento de los datos y las técnicas empleadas para generar las recomendaciones, así como los lineamientos seguidos para asegurar que la propuesta resulte escalable, reproducible y alineada con los objetivos del negocio.

3.1. Diseño de solución

El sistema de recomendación se diseñó con el objetivo de estimar la afinidad entre clientes y productos en un entorno caracterizado por alta escala, heterogeneidad de perfiles y rotación constante del portafolio. El diseño de la solución se apoyó en una arquitectura en capas que permite integrar diversas fuentes de datos, transformarlas en estructuras analíticas consistentes, entrenar modelos capaces de capturar relaciones complejas y, finalmente, desplegar las recomendaciones en un flujo operativo estable y reproducible.

La primera capa corresponde a la ingesta de datos, instancia en la que se integran registros transaccionales, interacciones digitales y atributos contextuales. Los datos transaccionales reflejan las compras efectivas realizadas en distintos horizontes temporales, lo que aporta evidencia directa sobre las preferencias observadas. Las interacciones digitales, en cambio, ofrecen señales implícitas de interés a partir de búsquedas, visualizaciones de productos o modificaciones en el carrito. Finalmente, los atributos contextuales caracterizan tanto a los clientes, mediante variables asociadas a su canal de comercialización, localización o tamaño, como a los productos, a partir de propiedades como marca, categoría o segmento.

La segunda capa se orienta a la preparación de los datos. En esta etapa se construyen las matrices de interacciones cliente–producto y se generan representaciones temporales que permiten capturar la dinámica de la demanda. Asimismo, se aplican técnicas de tratamiento de valores faltantes y de codificación de atributos categóricos, con el fin de asegurar consistencia y compatibilidad entre las distintas fuentes. El resultado de este proceso es un conjunto de estructuras homogéneas que sientan las bases para la etapa de modelado.

El modelado constituye la tercera capa de la arquitectura. En este punto se combinan distintos enfoques con el fin de maximizar la capacidad predictiva y superar las limitaciones de cada técnica individual. El filtrado colaborativo implícito, implementado a través de factorización matricial con el método ALS, permite capturar patrones latentes a partir de historiales de compra extensos. Los modelos

basados en contenido complementan este enfoque al aprovechar descripciones de clientes y productos, y ofrecen una alternativa frente al problema del arranque en frío. Adicionalmente, se exploran modelos híbridos y de aprendizaje profundo capaces de integrar simultáneamente señales transaccionales y digitales, y de modelar relaciones no lineales entre las variables.

Finalmente, la capa de despliegue asegura la integración del motor de recomendación en el ecosistema tecnológico de la empresa. El *pipeline* resultante genera listas de productos priorizados para cada cliente, incorpora mecanismos de versionado y monitoreo de modelos, y permite evaluar su desempeño en forma continua. De este modo, la solución se diseñó no solo para alcanzar precisión en la generación de recomendaciones, sino también para garantizar escalabilidad, reproducibilidad y adaptabilidad frente a la evolución del portafolio y a los cambios en los objetivos estratégicos.

La arquitectura de la solución se representa en la figura 3.1. Allí se observa el flujo general del sistema, desde la integración de datos hasta la generación de recomendaciones. El diagrama sintetiza los módulos principales y sus interacciones, y ofrece una visión global que facilita comprender cómo se organiza el motor de afinidad.

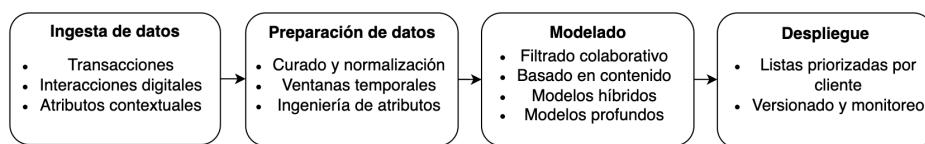


FIGURA 3.1. Arquitectura de alto nivel del sistema de recomendación.

3.2. Análisis exploratorio de los datos

El análisis exploratorio constituye una etapa fundamental para comprender la estructura y los patrones subyacentes en la información disponible antes de su utilización en modelos de recomendación. Su objetivo es identificar distribuciones, tendencias y relaciones entre variables que permitan caracterizar el comportamiento de los clientes y del portafolio de productos, así como anticipar posibles limitaciones o sesgos que afecten el desempeño de los algoritmos.

En esta sección se examinan distintas dimensiones de los datos, e incluye la concentración de clientes y productos, la diversidad de los portafolios de compra, las correlaciones entre variables transaccionales y digitales, y la presencia de sesgos asociados a la popularidad. Este análisis preliminar no solo proporciona una visión descriptiva del conjunto de datos, sino que también orienta decisiones posteriores de ingeniería de atributos y diseño de modelos, al revelar qué señales resultan más informativas y qué fenómenos requieren un tratamiento específico.

3.2.1. Curvas de concentración de clientes y productos

El análisis de concentración constituye un paso clave para comprender la distribución del consumo en entornos de negocio masivo. La figura 3.2 muestra la curva de concentración de productos, donde se observa que un reducido conjunto

concentra la mayor parte del volumen total. En particular, el 20 % de los productos explica cerca del 90 % de las ventas acumuladas, mientras que el resto conforma una extensa cola larga con niveles de rotación significativamente menores. Este comportamiento coincide con la ley de Pareto o principio 80/20 [20], ampliamente documentado en mercados de consumo masivo, donde la dinámica competitiva se organiza en torno a un pequeño núcleo de artículos de alta popularidad y una mayoría de baja incidencia [21].

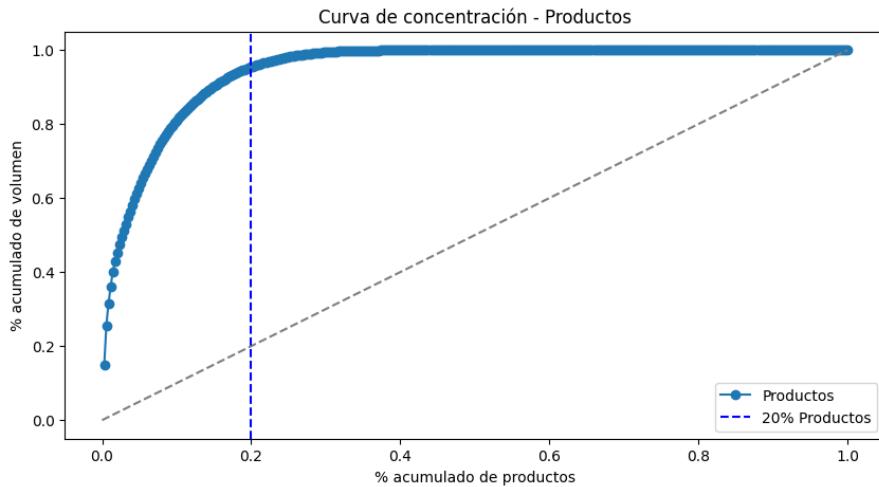


FIGURA 3.2. Concentración de productos en el portafolio.

De manera análoga, la figura 3.3 refleja la concentración del consumo en la base de clientes. Los resultados indican que cerca del 20 % de los puntos de venta generan alrededor del 80 % del volumen total, lo que pone de manifiesto la existencia de clientes estratégicos que concentran gran parte de la demanda. Esta distribución desigual plantea desafíos relevantes para el diseño de sistemas de recomendación, ya que las señales provenientes de clientes de alto volumen tienden a dominar los modelos, lo que genera sesgos hacia productos y comportamientos mayoristas.

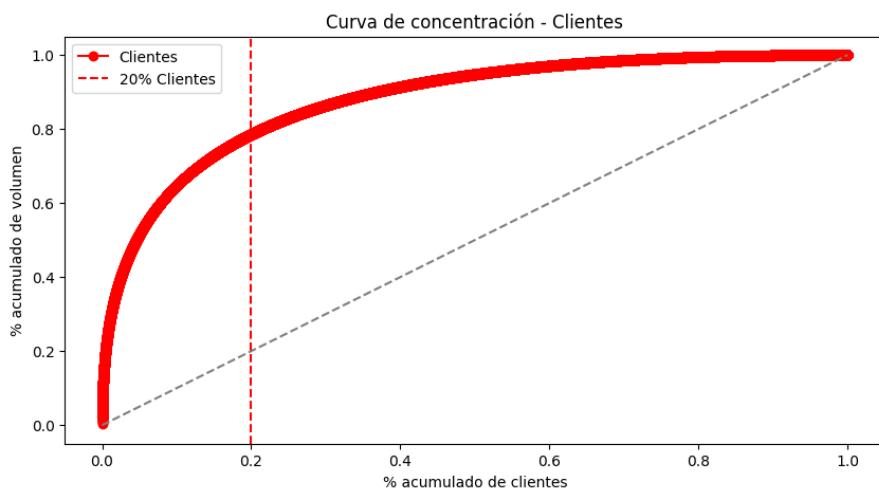


FIGURA 3.3. Concentración de clientes.

La evidencia empírica confirma así que tanto el portafolio de productos como la base de clientes presentan fuertes patrones de concentración. En consecuencia, un motor de recomendación que busque maximizar su impacto no solo debe capturar la afinidad entre clientes y productos más relevantes, sino también considerar mecanismos que favorezcan la diversidad y la exploración de la cola larga. Esta perspectiva resulta fundamental para equilibrar la explotación de los artículos de mayor rotación con la exposición de productos menos populares, lo que alinea los objetivos de negocio con la mejora de la experiencia del cliente.

3.2.2. Patrones de diversidad en el portafolio

El análisis de la diversidad en el portafolio de productos por cliente permite comprender la amplitud y heterogeneidad de los hábitos de consumo. La figura 3.4 muestra la distribución del número de productos distintos adquiridos por cliente en un mes. Los resultados evidencian que la mayoría de los puntos de venta concentra su demanda en un conjunto reducido de referencias, mientras que un número menor incorpora una mayor amplitud de marcas y presentaciones. Esta asimetría confirma la coexistencia de clientes de bajo rango de exploración con otros de portafolio más diversificado.

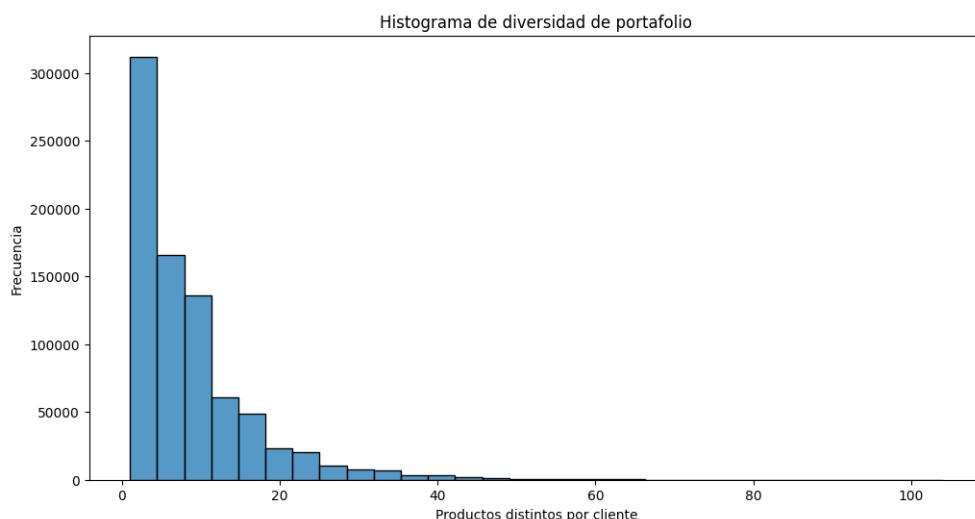


FIGURA 3.4. Histograma de diversidad de portafolio: número de productos distintos por cliente.

La figura 3.5 ilustra el fenómeno de concentración extrema en la demanda, donde unos pocos productos acumulan la mayoría de los pedidos mientras la gran mayoría registra volúmenes marginales. Para representar este patrón se utiliza un *log-log plot*, en el cual tanto el ranking de los productos como su número total de pedidos se expresan en escala logarítmica. Esta transformación permite visualizar con mayor claridad distribuciones de tipo cola larga, que en escalas lineales suelen quedar ocultas por la presencia de artículos extremadamente populares. El gráfico muestra una pendiente decreciente que confirma la existencia de este comportamiento: un reducido conjunto de productos concentra un volumen muy elevado, mientras que el resto se distribuye en la larga cola de baja rotación.

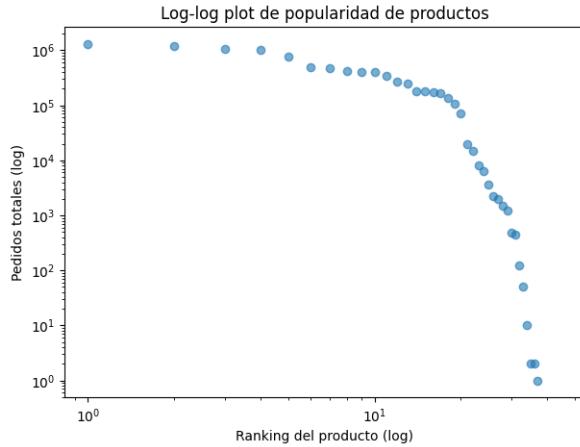


FIGURA 3.5. *Log log plot* de la popularidad de productos.

Este patrón no solo refuerza la evidencia presentada en las curvas de concentración, sino que además resalta un sesgo estructural que enfrenta cualquier sistema de recomendación en entornos de consumo masivo. Al entrenarse sobre datos históricos, los modelos tienden de manera natural a privilegiar los productos más populares, lo que reproduce el sesgo de popularidad y reduce la diversidad de las sugerencias. Este fenómeno señala la tensión entre explotación de productos estrella y exploración de la cola larga [22]. En este contexto, el desafío consiste en diseñar mecanismos que permitan balancear ambos extremos, de modo que se garantice relevancia sin sacrificar diversidad ni cobertura.

El análisis de co-ocurrencia entre los productos más relevantes, presentado en la figura 3.6, revela patrones de complementariedad en la demanda. Determinadas marcas y presentaciones tienden a aparecer de manera conjunta en los carritos de compra, lo que sugiere asociaciones naturales que pueden ser aprovechadas por un motor de recomendación. Estos resultados refuerzan la importancia de capturar no solo la popularidad individual de cada producto, sino también las relaciones de afinidad que emergen a nivel de portafolio.

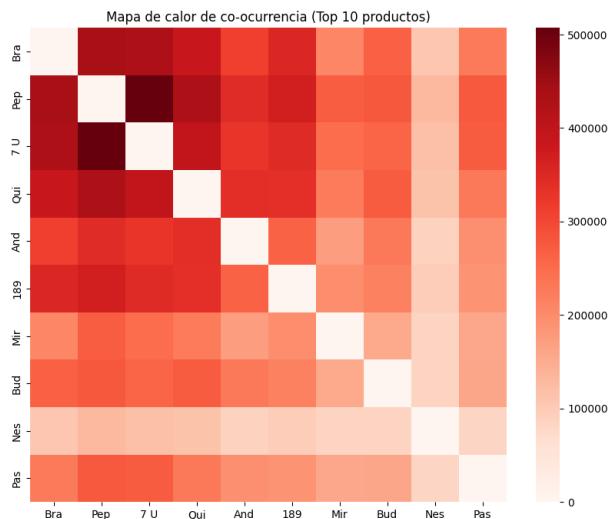


FIGURA 3.6. Mapa de calor de co-ocurrencia entre los 10 productos más relevantes.

3.2.3. Correlaciones entre variables transaccionales y digitales

El análisis de correlaciones busca identificar hasta qué punto las señales digitales anticipan comportamientos de compra y, en consecuencia, evaluar su potencial como insumos predictivos. Con el fin de evaluar la relación entre interacciones digitales y transacciones, se construyó una matriz de correlación entre las principales variables del conjunto de datos, observada en la figura 3.7.

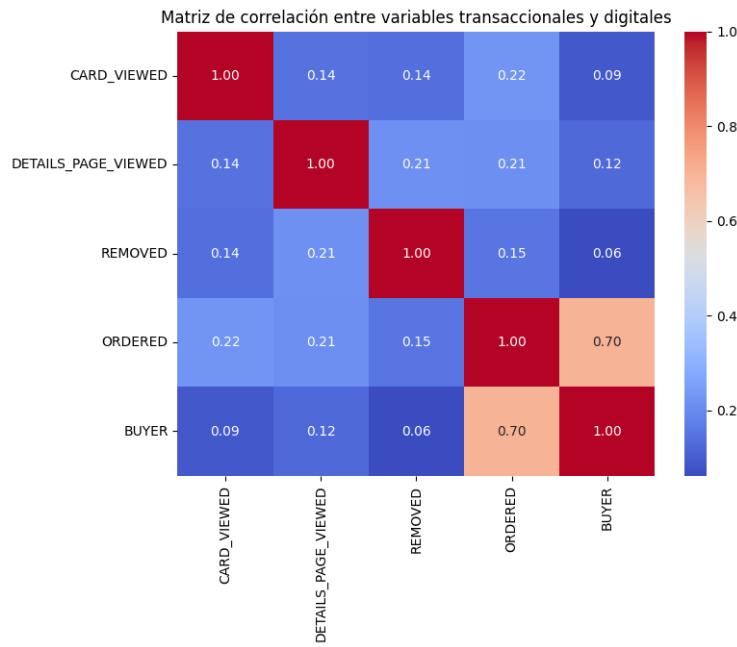


FIGURA 3.7. Matriz de correlación entre variables transaccionales y digitales.

Los resultados muestran una correlación elevada entre ORDERED y BUYER ($r = 0,70$), coherente con el hecho de que ambas variables reflejan distintos aspectos de la misma dimensión de compra. En contraste, las correlaciones de las señales digitales con las variables de compra resultan positivas pero de menor magnitud: CARD_VIEWED y DETAILS_PAGE_VIEWED muestran coeficientes bajos, lo que indica que la exposición y exploración de productos acompaña el proceso de compra, aunque no lo determina. La variable REMOVED presenta la relación más débil, lo que sugiere que los eventos de descarte contienen información ruidosa y limitada respecto de la propensión a comprar.

La correlación contemporánea entre interacciones digitales y compras confirma que las transacciones pasadas siguen siendo el principal indicador de comportamiento, mientras que las señales digitales aportan evidencia complementaria que, si bien débil de manera aislada, resulta relevante al integrarse en un modelo híbrido.

Con el fin de explorar la capacidad predictiva de estas variables, se calculó la correlación de cada una con la compra del mismo cliente–producto en el mes siguiente. Los resultados, en la figura 3.8, muestran que las transacciones pasadas (BUYER, ORDERED) son los predictores más fuertes, aunque las señales digitales también aportan información incremental. En particular, la variable CARD_VIEWED

presenta un coeficiente relevante, lo que respalda la hipótesis de que la exposición reiterada a un producto incrementa la probabilidad de recompra.

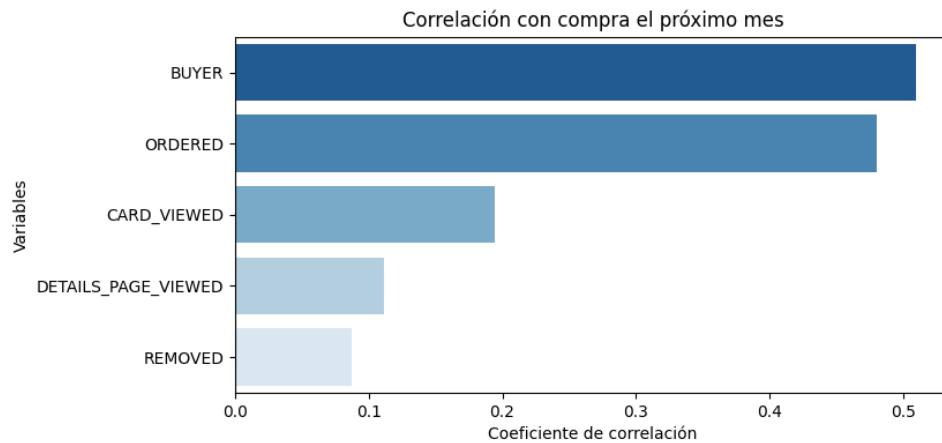


FIGURA 3.8. Correlación de variables con la compra en el mes siguiente.

Finalmente, se evaluó la tasa de recompra según la combinación de señales observadas en meses previos, en la tabla 3.1. Los clientes que registran tanto interacción como transacción presentan la mayor tasa de recompra (58,8 %), seguidos por aquellos con solo órdenes (44,1 %). En contraste, quienes solo exhiben interacciones digitales alcanzan un nivel considerablemente menor (17,6 %), incluso por debajo del grupo sin ningún registro previo (28,3 %). Este resultado sugiere que las interacciones aisladas no constituyen un predictor confiable de recompra, sino que tienden a reflejar un interés superficial que rara vez se traduce en pedidos. En cambio, la combinación de transacciones previas con señales digitales se confirma como el escenario de mayor poder explicativo, ya que aprovecha la solidez de la evidencia transaccional y, al mismo tiempo, permite mejorar la capacidad de anticipar comportamientos futuros en casos donde no existen registros abundantes de compra.

TABLA 3.1. Tasa de recompra según la combinación de señales previas.

Grupo	Tasa de recompra
Interacción y orden	58,79 %
Solo orden	44,05 %
Ninguno	28,30 %
Solo interacción	17,64 %

3.2.4. Observaciones preliminares del análisis exploratorio

El análisis exploratorio permitió identificar una serie de patrones que resultan fundamentales para orientar el diseño del motor de recomendación. En primer lugar, se confirmó que tanto el portafolio de productos como la base de clientes presentan fuertes niveles de concentración: un reducido conjunto explica la mayor parte del volumen, mientras que la mayoría se distribuye en una extensa cola

larga de baja rotación. Este comportamiento introduce un sesgo hacia popularidad que los modelos deben manejar para no sacrificar diversidad [21].

En segundo lugar, se observó que la diversidad en los portafolios de compra varía según el tipo de cliente, con autoservicios que incorporan un surtido más amplio en comparación con kioscos y tiendas tradicionales. Además, los análisis de co-ocurrencia revelaron asociaciones frecuentes entre ciertos productos, lo que sugiere la existencia de complementariedades que pueden ser aprovechadas en la generación de recomendaciones.

En tercer lugar, el estudio de correlaciones entre variables digitales y transaccionales mostró que, si bien las transacciones pasadas constituyen el predictor más sólido de comportamiento, las señales digitales aportan información incremental y se vuelven especialmente relevantes en escenarios de arranque en frío. La evaluación de tasas de recompra confirmó que la combinación de interacciones y compras pasadas es la fuente más robusta de predicción, mientras que las interacciones aisladas presentan un valor explicativo limitado.

En conjunto, estos hallazgos proporcionan una primera validación de la hipótesis central: la integración de señales transaccionales y digitales, complementadas con atributos contextuales, resulta clave para capturar la heterogeneidad del consumo y diseñar un motor de recomendación capaz de balancear precisión, diversidad y cobertura.

3.3. Preparación e ingeniería de los datos

La preparación e ingeniería de los datos constituyó uno de los pilares centrales del desarrollo del motor de afinidad, al definir cómo las distintas fuentes de información fueron transformadas en insumos consistentes, comparables y numéricamente útiles para el entrenamiento de los modelos de recomendación. El proceso integró fuentes transaccionales, digitales y contextuales, y abordó tanto la construcción de la matriz cliente–producto como la generación de atributos descriptivos de clientes y productos.

3.3.1. Fuentes y estructura general

El punto de partida son tres conjuntos principales de información: registros transaccionales, eventos digitales generados en la aplicación BEES y atributos contextuales de clientes y productos. Esta combinación permite construir una representación integral de la relación cliente–producto, en la que se entrelazan tanto preferencias explícitas como señales implícitas de interés.

Los registros transaccionales se encuentran a nivel de operación individual, con granularidad diaria y asociados a identificadores de cliente, producto, cantidad y monto. Como fue previsto en el análisis exploratorio, estos datos suelen exhibir un fuerte patrón de concentración cercano a la regla de Pareto [20], donde una pequeña fracción de marcas o ítems concentra la mayor parte del volumen. Este fenómeno, común en la industria, anticipa la necesidad de mitigar los sesgos hacia productos de alta rotación durante el modelado.

Los eventos digitales, por su parte, contienen información de búsquedas, visualizaciones, adiciones y remociones en el carrito, así como clics en promociones.

Estas interacciones permiten capturar señales tempranas de interés que no siempre se traducen en compras efectivas, pero amplían la cobertura del sistema. No obstante, su naturaleza exploratoria introduce ruido, por lo que se aplican filtros para eliminar registros residuales o no representativos, y se priorizan solo aquellos que expresan comportamientos consistentes de interés [5, 23].

Finalmente, los atributos contextuales ofrecen información complementaria sobre las características estructurales de clientes y productos. En los primeros, se incluyen variables de canal, localización y tamaño del punto de venta; en los segundos, descriptores como marca, segmento, envase o unidad de negocio. Estos factores resultan esenciales para capturar la heterogeneidad comercial que no siempre se refleja en los registros transaccionales.

3.3.2. Construcción de la matriz cliente–producto

La matriz cliente–producto constituye el núcleo del sistema de recomendación, al representar de forma estructurada las interacciones históricas y digitales entre los puntos de venta y el portafolio. Su construcción requirió integrar los eventos válidos depurados, agregados a nivel mensual por cliente y producto, con la contabilización de la frecuencia de ocurrencia de cuatro tipos principales: ORDERED, CARD_VIEWED, DETAILS_PAGE_VIEWED y REMOVED. Los dos primeros incluyeron subtipos que reflejan el origen de la recomendación, como búsquedas populares o programas de fidelización, preservados por su relevancia estadística.

El diseño temporal siguió la lógica operativa del sistema: las predicciones correspondientes al mes N se generan durante el mes $N - 1$ a partir de la información disponible previamente. Por ello, se implementó un esquema de ventanas móviles de seis meses ($N - 7$ a $N - 2$) con horizontes de observación de uno, tres y seis meses. Este enfoque permite capturar simultáneamente señales recientes y patrones de comportamiento estables, lo que mantiene la coherencia temporal del modelo [24, 25].

El conjunto de variables presentaba una alta heterogeneidad en magnitudes, derivada de la coexistencia de clientes con volúmenes dispares y productos de distinta rotación. Para mitigar los efectos de escala, se aplicó *winsorizing (clipping)* sobre los percentiles extremos [26] y se normalizaron las variables dentro del grupo cliente–categoría, de modo que cada valor representara la importancia relativa de un producto dentro del portafolio del cliente.

A partir de estas variables se estimaron pesos específicos por tipo de evento y horizonte temporal, combinados para obtener un *score* compuesto de preferencia implícita. Los pesos se ajustaron mediante optimización bayesiana con Optuna [27], con el objetivo de maximizar la métrica *Precision@10* sobre validación temporal. Se observó un patrón consistente de mayor relevancia para señales recientes sobre las que incorporan historia más lejana y para eventos transaccionales respecto de los digitales. El resultado fue una matriz de afinidad que sintetiza 60,2 millones de pares cliente–producto, con una densidad del 13,5 %, una correlación Buyer–Preference de 0,39 y un AUC de 0,8766, lo que evidencia una alta capacidad discriminante (tabla 3.2).

TABLA 3.2. Resumen de métricas descriptivas de la matriz cliente–producto.

Indicador	Valor
Total de pares cliente–producto	60 220 260
Densidad de la matriz (% de celdas no nulas)	13,50 %
Clientes en arranque en frío	2 981 (1,50 %)
Productos en arranque en frío	1 (0,54 %)
Media del <i>score</i> de preferencia	0,0046
Desvío estándar del <i>score</i>	0,0277
Correlación Buyer–Preference	0,3935
Área bajo la curva (AUC)	0,8766

3.3.3. Diseño de atributos de cliente y producto

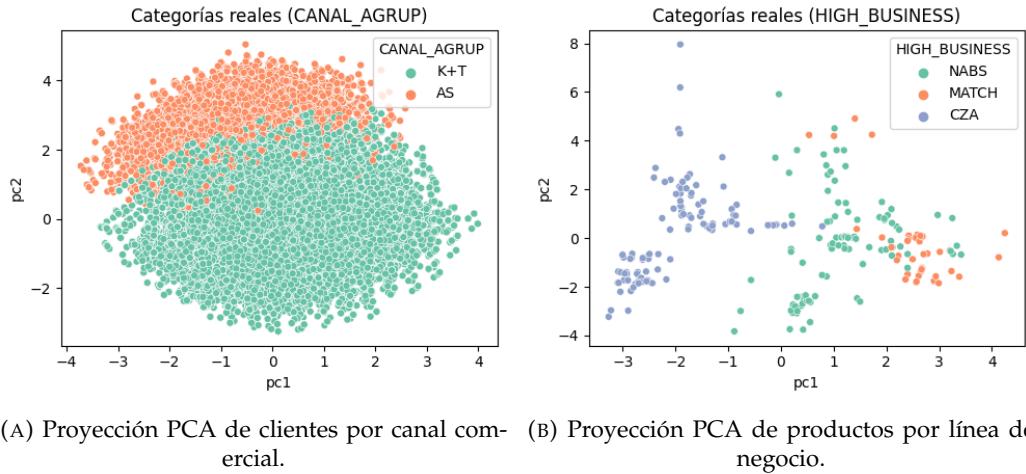
Además del *score* de interacción, se incorporaron atributos estructurales de clientes y productos para enriquecer las señales de afinidad. Para los puntos de venta se construyeron seis bloques principales: frecuencia de compra, estabilidad temporal, volumen y crecimiento, diversidad del mix, comportamiento de compra y atributos contextuales. En los productos, se modelaron variables de volumen, penetración, desempeño, diversidad geográfica y de canal, junto con descriptores estructurales como segmento y tipo de envase.

Todas las variables numéricas fueron tratadas con *winsorizing* y normalización *z-score* por grupo de negocio o canal, seguidas de discretización en tres cuantiles (bajo, medio, alto). Las variables categóricas se estandarizaron y completaron con etiquetas neutras. De este modo, se obtuvieron representaciones comparables, robustas y fácilmente integrables en modelos que emplean *feature embeddings*.

Evaluación de representatividad

Para evaluar la calidad informativa de los atributos, se analizó la varianza, la correlación con la ocurrencia de compra y la similitud geométrica entre entidades. En los clientes (95 atributos válidos), las mayores varianzas correspondieron a variables de volumen y frecuencia, mientras que las correlaciones más altas con la compra se asociaron con continuidad y madurez comercial, con coeficientes próximos a 0,32 y una similitud coseno promedio de 0,9985. En los productos (61 atributos), las correlaciones más fuertes (0,84) se observaron en penetración de clientes y cobertura comercial, con similitud coseno promedio de 0,64, lo que indica una mayor dispersión relativa del portafolio.

Complementariamente, un análisis de componentes principales (PCA) permitió visualizar la estructura de estas representaciones. En los clientes, la proyección reveló una diferenciación coherente entre los canales Autoservicio (AS) y Kiosco + Tradicional (K+T), mientras que en los productos emergieron grupos claramente definidos por unidad de negocio. Esto demuestra que las *features* capturan patrones estructurales reales y diferencian adecuadamente la heterogeneidad comercial de la base (figuras 3.9a y 3.9b).



La preparación e ingeniería de los datos permitió consolidar una base analítica sólida, donde la matriz cliente–producto captura las señales implícitas de afinidad y los atributos de cliente y producto aportan contexto y generalización, lo que habilita el uso combinado de enfoques colaborativos e híbridos en las etapas siguientes.

3.4. Desarrollo de modelos

El desarrollo de los modelos constituye la fase central del sistema de recomendación, donde la matriz cliente–producto construida en etapas previas se transforma en un mecanismo capaz de estimar la afinidad entre ambos. El objetivo es asignar a cada combinación posible un puntaje continuo que refleje la probabilidad relativa de interés, lo que permite ordenar los productos según la relevancia esperada para cada cliente.

Para abordar este desafío se exploraron distintos enfoques de modelado, que combinan estrategias colaborativas, basadas en contenido y de aprendizaje profundo. En primer lugar, se implementó un modelo de filtrado colaborativo mediante el algoritmo *Alternating Least Squares* [15, 7], que aprende representaciones latentes de clientes y productos a partir de los patrones de interacción observados. En segundo lugar, se desarrolló un modelo híbrido con *LightFM* [16], capaz de integrar señales colaborativas con atributos explícitos de clientes y productos, lo que mitiga el problema del arranque en frío.

Complementariamente, se incorporó una variante basada en redes neuronales del tipo *Two Towers* [28], que aprende representaciones de clientes y productos a partir de sus atributos categóricos y numéricos mediante arquitecturas de *embeddings*. Sus representaciones pueden combinarse con las generadas por ALS en un esquema híbrido de ensamble, lo que fortalece la robustez y la capacidad de generalización del sistema. La segunda corresponde al enfoque de *Neural Collaborative Filtering* [29], que reemplaza la combinación lineal de *embeddings* por un modelo neuronal capaz de capturar relaciones no lineales de afinidad.

En conjunto, estos modelos representan una progresión desde métodos clásicos hacia aproximaciones más flexibles y expresivas. Las siguientes subsecciones describen los fundamentos, la formulación y las particularidades de cada uno, lo que sienta las bases para la comparación de desempeño y costos que se desarrolla en el capítulo siguiente.

3.4.1. Filtrado colaborativo con ALS

El primer enfoque desarrollado fue un modelo de filtrado colaborativo basado en factorización matricial mediante el algoritmo *Alternating Least Squares* [9]. Este método permite descomponer la matriz cliente–producto en dos espacios latentes de menor dimensión, uno para los clientes y otro para los productos, de modo que la afinidad entre ambos se estime como el producto interno de sus vectores representativos. El objetivo es capturar patrones de coocurrencia en las interacciones históricas y proyectarlos hacia combinaciones no observadas, lo que permite generar recomendaciones personalizadas a partir del comportamiento colectivo.

Configuración del modelo

En este trabajo se utilizó la variante de ALS para *feedback* implícito, apropiada para contextos donde las señales de preferencia provienen de interacciones observadas en lugar de calificaciones explícitas [7]. Bajo este esquema, la ausencia de interacción no se interpreta como una valoración negativa, sino como falta de evidencia. La matriz de entrada corresponde al *score* de preferencia compuesto descrito en la sección anterior, el que integra eventos transaccionales y digitales ponderados según su relevancia y temporalidad. Cada celda representa un grado de afinidad implícita entre el cliente y el producto, estimado a partir de seis meses de comportamiento histórico.

El modelo se configuró con un número de factores latentes ajustable (*rank*), un parámetro de regularización λ para controlar el sobreajuste y un parámetro de confianza α que pondera la influencia de las observaciones positivas frente a las ausentes [7]. El entrenamiento se realizó de manera iterativa, con la alternancia en la actualización de las matrices de clientes y productos hasta alcanzar la convergencia.

Optimización de hiperparámetros

Para la selección de hiperparámetros se implementó un proceso de optimización bayesiana con Optuna [27], compuesto por veinte iteraciones orientadas a maximizar la métrica *Precision@5* sobre un conjunto de validación temporal. La búsqueda abarcó los principales parámetros del modelo: dimensión latente (*rank* $\in [10, 50]$), número máximo de iteraciones (*maxIter* $\in [5, 20]$), nivel de regularización (*regParam* $\in [10^{-4}, 10^{-1}]$) y parámetro de confianza (*alpha* $\in [10^{-2}, 10]$).

En cada iteración, el modelo fue entrenado sobre las interacciones comprendidas entre los meses $N - 7$ y $N - 2$, y evaluado sobre el mes $N - 1$, lo que reprodujo el flujo real de generación de recomendaciones.

Resultados y conclusiones

Los resultados obtenidos mostraron un comportamiento consistente y robusto. El modelo alcanzó una *Precision@10* de 31,5 % y una *Recall@10* de 33,7 %, lo que indica una buena capacidad para priorizar los productos efectivamente comprados en el siguiente período. Se observó además que el ALS tiende a capturar de manera eficiente las relaciones entre clientes con historial suficiente, pero su desempeño disminuye en escenarios de arranque en frío o cuando la matriz presenta elevada dispersión. Por ello, este modelo se adoptó como línea base sobre la que se construyeron enfoques híbridos más expresivos en las siguientes etapas.

El modelo ALS demostró ser una herramienta eficaz para extraer representaciones latentes de afinidad a partir del comportamiento histórico [9]. Su estructura matemática simple, estabilidad en entornos de gran escala y adecuación al feedback implícito lo convierten en un componente esencial del *pipeline* de recomendación desarrollado.

3.4.2. Modelo híbrido con LightFM

El segundo enfoque explorado fue un modelo híbrido basado en la biblioteca *LightFM* [16], que combina técnicas de filtrado colaborativo y modelos basados en contenido dentro de un mismo marco de aprendizaje. Este modelo extiende la factorización matricial tradicional al incorporar vectores de características (*feature embeddings*) asociados tanto a los usuarios como a los ítems, lo que permite capturar información contextual incluso para aquellos pares que no poseen interacciones históricas, lo que mitiga el problema de arranque en frío identificado en la sección anterior.

A diferencia del ALS, que aprende representaciones exclusivamente a partir de la matriz de interacciones, LightFM incorpora atributos estructurales de clientes y productos como señales adicionales [16]. En este trabajo, las representaciones de cliente incluyeron variables de frecuencia, estabilidad, volumen, mezcla, comportamiento y contexto, mientras que las de producto comprendieron características de volumen, penetración, composición de canal, segmento, diversidad y desempeño. Cada conjunto de variables fue normalizado, discretizado y codificado antes de su integración al modelo, conforme al *pipeline* de ingeniería descrito previamente.

El modelo *LightFM* combina señales colaborativas y de contenido dentro de un espacio latente común, y representa tanto a los clientes como a los productos mediante vectores que integran información transaccional y contextual. Su entrenamiento parte de una matriz binaria de interacciones, donde cada par cliente–producto indica la existencia o no de contacto en el período de análisis. Estas interacciones positivas constituyen la evidencia de afinidad sobre la que el modelo aprende a distinguir entre ítems relevantes y no relevantes para cada usuario.

A diferencia del enfoque de ALS, que busca ajustar magnitudes continuas de preferencia implícita, *LightFM* se entrena mediante la optimización de una función de pérdida orientada al ordenamiento relativo de los ítems en el ranking. Este tipo de funciones, ampliamente utilizadas en escenarios de *feedback* implícito, penalizan los errores en las primeras posiciones y favorecen que los productos con mayor probabilidad de interacción ocupen los primeros lugares en las recomendaciones.

A cada interacción se le asignó además un peso relativo o *sample weight*, que determina su influencia durante el entrenamiento. Estos pesos se derivaron del puntaje de preferencia implícito calculado en la etapa de ingeniería de atributos y se transformaron mediante una función logarítmica que reduce la dispersión entre observaciones extremas. Posteriormente, los valores fueron normalizados alrededor de su media global, de manera que las interacciones más intensas, como compras frecuentes o múltiples eventos asociados al mismo producto, tuvieran mayor impacto que aquellas esporádicas.

Este esquema permite que el modelo capture no sólo la ocurrencia de una interacción, sino también su intensidad relativa, ya que integra señales de distinta fuerza en el proceso de aprendizaje. A diferencia de ALS, que optimiza una función cuadrática ponderada con el objetivo de reconstruir las magnitudes observadas de preferencia implícita, *LightFM* utiliza un criterio de aprendizaje basado en el ordenamiento, donde los pesos asignados a cada interacción afectan directamente la probabilidad de que un ítem sea priorizado en el ranking final [30, 16]. Esto permite capturar diferencias más finas en la intensidad de las señales, ya que integra tanto la frecuencia como la relevancia relativa de cada evento dentro del proceso de entrenamiento.

El entrenamiento se llevó a cabo sobre el conjunto de interacciones ponderadas, con el empleo de las matrices de características de usuario y producto generadas en la etapa anterior. El modelo se optimizó durante veinte épocas, mediante el uso de procesamiento paralelo en cuatro hilos, hasta alcanzar estabilidad en la función objetivo. Esta configuración resultó adecuada para equilibrar precisión y eficiencia computacional, y sirvió como base para los distintos ensayos de complejidad incremental desarrollados a continuación.

Evaluación experimental del modelo LightFM

Con el objetivo de analizar el aporte incremental de las variables contextuales, se realizaron tres configuraciones experimentales del modelo *LightFM* con distintos niveles de complejidad en las representaciones de usuario y producto. Se describen los detalles metodológicos de cada ensayo, así como la selección de variables, la arquitectura del modelo y los criterios de evaluación. El detalle completo de la experimentación se presenta en el Anexo B.

Los resultados mostraron una mejora progresiva en la capacidad predictiva al incorporar variables discretizadas y explicativas, seguida de una estabilización del desempeño tras la depuración final de atributos. Las métricas globales se mantuvieron estables y competitivas, con valores de *Precision@10* y *Recall@10* cercanos al 26 % y 28 %, respectivamente (tabla 3.3).

TABLA 3.3. Resumen de métricas de desempeño para las tres configuraciones experimentales del modelo *LightFM*.

Configuración	Precision@10	Recall@10
Test 1 – Contexto categórico reducido	25,3 %	27,1 %
Test 2 – Atributos discretizados y explicativos	26,1 %	27,8 %
Test 3 – Depuración y selección de <i>embeddings</i>	25,8 %	27,5 %

Estos resultados confirman que el modelo híbrido logra capturar relaciones no lineales y de alta dimensionalidad en entornos de señales implícitas dispersas, lo que valida la coherencia de las representaciones latentes y establece una base sólida para la versión final optimizada que se presenta en la siguiente sección.

Optimización bayesiana de hiperparámetros

Con el objetivo de maximizar la precisión del ranking y validar la robustez del modelo frente a diferentes configuraciones, se llevó a cabo un proceso de optimización bayesiana mediante la biblioteca Optuna. La búsqueda se orientó a maximizar la métrica *Precision@5*, con prioridad en la capacidad del sistema para ubicar los productos más relevantes en las primeras posiciones de recomendación.

El espacio de búsqueda incluyó combinaciones de número de componentes latentes (`no_components` ∈ {32, 64, 96, 128, 192}), funciones de pérdida WARP y BPR, tasas de aprendizaje en el rango $[5 \times 10^{-4}, 5 \times 10^{-3}]$, y parámetros de regularización `item_alpha` y `user_alpha` en el intervalo $[10^{-6}, 10^{-4}]$. También se evaluaron valores de `max_sampled` entre 5 y 15, con el fin de lograr un balance adecuado entre exploración y costo computacional.

Resultados y conclusiones

El modelo alcanzó una *Precision@10* de 29,2 % y un *Recall@10* de 31,2 %, superó las versiones anteriores tanto en precisión como en cobertura, y se consolidó como la mejor alternativa dentro del conjunto de modelos evaluados.

Estos resultados confirman que la combinación de la función de pérdida BPR con un número intermedio de componentes latentes ofrece un equilibrio óptimo entre capacidad representacional y generalización, lo que maximiza la recuperación de productos relevantes sin sobreajustar a las interacciones más frecuentes. El modelo resultante constituye la versión final del motor híbrido de afinidad, el cual integra señales transaccionales y contextuales dentro de un espacio latente de alta coherencia semántica.

3.4.3. Modelo basado en contenido con arquitectura Two-Tower

El modelo *Two-Tower* representa una extensión moderna del enfoque híbrido introducido con *LightFM*, y se caracteriza por la incorporación de una red neuronal de dos torres entrenada para aprender representaciones continuas (*embeddings*) de clientes y productos a partir de sus características estructurales [31].

A diferencia de *LightFM*, que combina señales colaborativas y de contenido en una factorización lineal, la arquitectura *Two-Tower* permite capturar interacciones no lineales de alta dimensionalidad mediante capas densas y funciones de activación, lo que otorga mayor poder expresivo y capacidad de generalización en escenarios complejos. Mientras *LightFM* aprende relaciones principalmente lineales entre las *features* y las preferencias, el modelo *Two-Tower* introduce una parametrización no lineal capaz de capturar interacciones complejas entre variables numéricas y categóricas.

Cada torre de la red, una para clientes y otra para productos, procesa sus respectivos vectores de características contextuales y numéricas, y genera un espacio latente común donde las entidades con patrones de comportamiento o atributos

similares quedan próximas entre sí [31, 32]. El modelo se entrena con el objetivo de maximizar la similitud coseno entre pares positivos (cliente–producto con interacción) y de minimizarla frente a un conjunto de ejemplos negativos muestreados por contexto (*hard negatives*) [28]. Esta estrategia de entrenamiento orientada al ranking, basada en la combinación de pérdidas *MarginRankingLoss* y *Binary Cross-Entropy*, permite equilibrar la discriminación entre ítems relevantes y la estabilidad del aprendizaje.

Diseño y configuración del modelo

El conjunto de características de entrada se definió a partir de los atributos estructurales más representativos de clientes y productos utilizados en los modelos anteriores.

El modelo se configuró con un tamaño de *embedding* de 128 dimensiones y dos capas ocultas de 128 y 64 neuronas en cada torre, conectadas mediante funciones de activación ReLU. La figura 3.10 muestra la arquitectura general del modelo, donde ambas torres, una para los clientes y otra para los productos, aprenden representaciones independientes que luego se combinan mediante una medida de similitud en el espacio latente para estimar la afinidad entre ambas entidades.

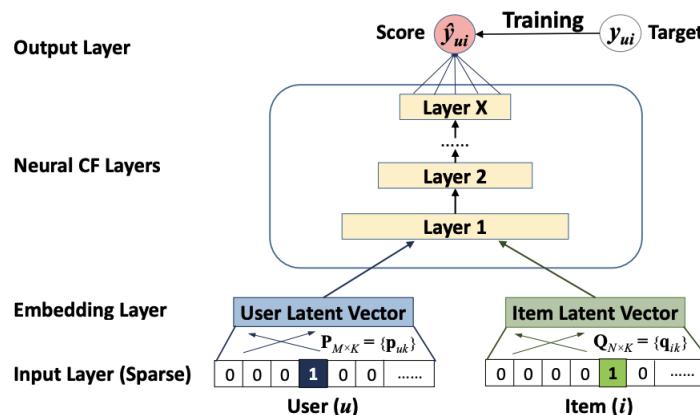


FIGURA 3.10. Arquitectura del modelo *Two Towers*¹.

El entrenamiento se llevó a cabo durante cinco épocas con un tamaño de lote de 1024 y una tasa de aprendizaje de 0,001, con el uso del optimizador Adam. Esta configuración equilibró adecuadamente la capacidad representacional y la estabilidad del aprendizaje, lo que mantuvo un costo computacional razonable para los volúmenes de datos involucrados.

Un aspecto central del proceso de entrenamiento fue la generación de ejemplos negativos (*negative sampling*) para complementar los pares positivos de cliente y producto observados [28]. Por cada interacción positiva se generaron ocho combinaciones negativas, seleccionadas de forma controlada dentro del mismo contexto de negocio o segmento del producto. A diferencia de un muestreo aleatorio, que puede introducir casos triviales o poco informativos, este enfoque de *hard negative sampling* busca construir ejemplos desafiantes, es decir, productos similares pero no adquiridos por el cliente. De esta manera, el modelo aprende a

¹Imagen tomada de <https://arxiv.org/pdf/1708.05031.pdf> [29]

distinguir entre afinidades reales y coincidencias superficiales, lo que refuerza su capacidad discriminante y la calidad de las representaciones en el espacio latente.

Ensamble híbrido con ALS

Con el objetivo de integrar señales de naturaleza distinta, colaborativas y de contenido, se implementó un ensamblaje entre el modelo ALS y el *Two-Tower*. Ambos generan puntuaciones continuas de afinidad sobre el espacio cliente–producto, las que fueron combinadas mediante una función de promedio ponderado, representado en la ecuación 3.1, donde λ controla el peso relativo de la componente colaborativa (ALS) y de la componente de contenido (*Two-Tower*).

$$\text{Score}_{hybrid} = \lambda \cdot \text{Score}_{ALS} + (1 - \lambda) \cdot \text{Score}_{TT} \quad (3.1)$$

El valor óptimo de λ se ajustó empíricamente en base a la métrica *Precision@10*, observándose que ponderaciones de $\lambda = 0,8$ lograron el mejor equilibrio entre precisión y cobertura.

Este enfoque híbrido aprovecha simultáneamente la densidad informativa de las interacciones históricas y la riqueza contextual de las representaciones aprendidas por el modelo neuronal, lo que consolida un sistema de recomendación más robusto frente a la escasez de datos y la variabilidad estructural del portafolio.

Resultados y conclusiones

El modelo *Two-Tower* mostró un desempeño superior respecto de los enfoques previos, lo que consolidó su efectividad para capturar relaciones no lineales entre clientes y productos a partir de atributos contextuales y numéricos. En la evaluación sobre el conjunto de validación, alcanzó una *Precision@10* de 32,3 % y un *Recall@10* de 34,5 %, y superó tanto al modelo ALS como a las variantes híbridas de *LightFM* en ambas métricas.

Estos resultados reflejan la capacidad del modelo para generar representaciones continuas de mayor expresividad y generalización, integrando de manera efectiva las señales de contexto estructural, la intensidad de las interacciones y las relaciones latentes aprendidas en los espacios de *embedding*.

3.4.4. Neural Collaborative Filtering

El modelo *Neural Collaborative Filtering* (*NCF*) constituye una evolución directa de los enfoques basados en factorización latente, al reemplazar la combinación lineal de *embedding* por una arquitectura neuronal capaz de modelar interacciones no lineales entre usuarios y productos [29]. Este modelo integra dos componentes complementarios: una capa *Generalized Matrix Factorization* (*GMF*) que conserva la naturaleza lineal y explicativa del filtrado colaborativo clásico, y una red *Multi-Layer Perceptron* (*MLP*) que aprende patrones de interacción de mayor complejidad.

La fusión de ambas salidas en la capa final, denominada *NeuMF*, permite capturar simultáneamente señales de proximidad latente y relaciones no lineales de alto orden, lo que extiende la capacidad de generalización del sistema de recomendación.

En comparación con el modelo *Two-Tower*, que entrena torres independientes a partir de características contextuales, el NCF opera directamente sobre los identificadores embebidos de usuarios y productos, y modela de forma explícita las interacciones entre ambos espacios latentes. Esta característica introduce un valor adicional en escenarios de datos implícitos, ya que permite aprender una función de similitud más expresiva sin que dependa de features contextuales externas, y aprovecha las señales puramente colaborativas del histórico transaccional.

Diseño y configuración

El modelo se implementó bajo la arquitectura *NeuMF*, que combina dos componentes complementarios en paralelo: un bloque GMF, responsable de modelar las interacciones multiplicativas lineales mediante *embeddings* de 64 dimensiones, y un bloque MLP que incorpora una parametrización no lineal del mismo tamaño de *embedding*, seguido por dos capas densas de 128 y 64 neuronas con activación ReLU y regularización Dropout (0.1). Ambas representaciones se integran en una capa de fusión que concatena las salidas de los dos caminos y las proyecta en una capa final lineal con activación sigmoide, encargada de estimar la probabilidad de interacción positiva entre cada par usuario–producto. La figura 3.11 ilustra la estructura híbrida del modelo, y muestra cómo las ramas GMF y MLP convergen en una representación conjunta que combina el poder predictivo de la factorización lineal con la expresividad de las redes neuronales profundas.

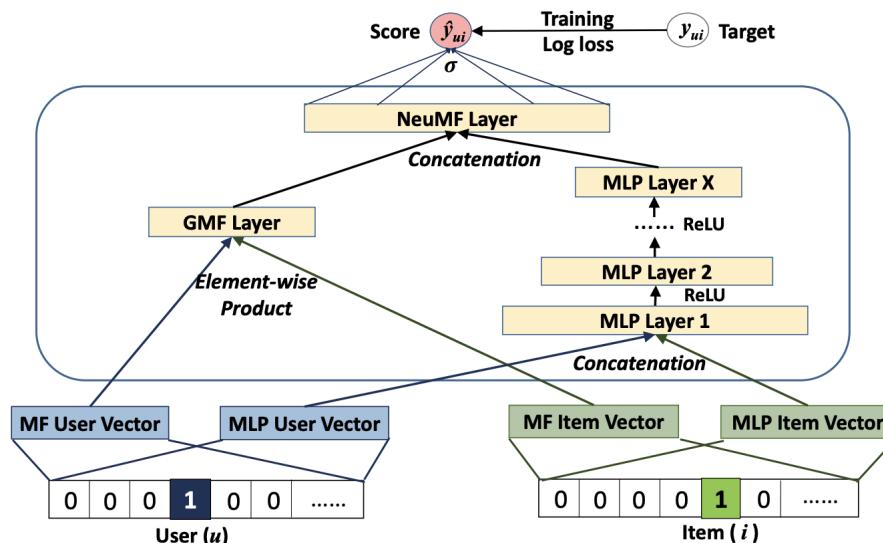


FIGURA 3.11. Arquitectura del modelo *Neural Collaborative Filtering* híbrido².

El entrenamiento se realizó durante cinco épocas con un tamaño de lote de 4096 y una tasa de aprendizaje de 0,001. Por cada interacción positiva observada, se generaron cuatro negativos contextuales seleccionados dentro del mismo segmento de negocio y familia de producto. Este procedimiento, conocido como *hard negative sampling*, busca aumentar la dificultad de aprendizaje del modelo al exponerlo a ejemplos negativos más informativos y semánticamente próximos, lo que fortalece la capacidad de discriminación del clasificador.

²Imagen tomada de <https://arxiv.org/pdf/1708.05031.pdf> [29]

Resultados y conclusiones

El modelo NCF alcanzó una *Precision@10* de 32,4 % y un *Recall@10* de 34,3 % sobre el conjunto de validación. Estos resultados evidencian una mejora significativa respecto de los modelos puramente basados en factorización o redes separadas, lo que demuestra la efectividad del enfoque híbrido GMF + MLP para capturar tanto relaciones lineales como no lineales entre las entidades.

Además de su mayor capacidad predictiva, el modelo presentó una convergencia estable y un comportamiento robusto frente a distintas configuraciones de negativos y tamaños de *embedding*, lo que lo convierte en una alternativa eficiente para escenarios de datos altamente implícitos.

En conjunto, el *Neural Collaborative Filtering* representa el paso más avanzado dentro del *pipeline* de afinidad, lo que consolida una arquitectura neuronal completamente diferenciable y optimizable de extremo a extremo, que integra de manera orgánica los principios de factorización matricial y aprendizaje profundo.

3.5. Implementación

La implementación del sistema de recomendación requirió articular los distintos componentes desarrollados dentro de un flujo de trabajo unificado, reproducible y escalable. Para ello se diseñó un *pipeline* modular que integra los procesos de ingestión, transformación, modelado y evaluación, con soporte para el versionado y monitoreo de artefactos en producción.

3.5.1. Diseño del *pipeline* de procesamiento

El flujo completo se estructuró en cuatro etapas principales: ingestión, preparación, modelado y predicción. En la fase de ingestión se integraron las fuentes de datos transaccionales, digitales y contextuales en un entorno distribuido, lo que garantizó la consistencia de los identificadores y la alineación temporal entre registros.

Durante la preparación, se aplicaron las transformaciones de limpieza, agregación, codificación y normalización para construir la matriz cliente–producto que sirve de insumo a los modelos.

En la etapa de modelado, se ejecutaron los distintos enfoques desarrollados, y se almacenaron sus métricas, parámetros y versiones.

Finalmente, en la fase de predicción se generaron los puntajes de afinidad y las listas *Top-K* para cada cliente, que constituyen la salida principal del sistema.

3.5.2. Integración con la infraestructura tecnológica

La ejecución del *pipeline* se realizó en la plataforma Databricks [12], que permitió procesar grandes volúmenes de datos de forma distribuida mediante el uso de PySpark. Este entorno facilitó la orquestación de tareas, la parallelización de los cálculos y la trazabilidad de los resultados.

Para la gestión del ciclo de vida de los modelos se empleó MLflow [14], herramienta que permitió registrar los experimentos, almacenar los parámetros y métricas, y versionar los artefactos generados durante el entrenamiento. Cada ejecución de modelo quedó asociada a un identificador único, lo que posibilita

reproducir resultados, comparar configuraciones y recuperar versiones históricas de los modelos entrenados.

Esta integración entre Databricks y MLflow conformó una infraestructura robusta y escalable, adecuada tanto para la experimentación iterativa como para la implementación de *pipelines* automatizados.

3.5.3. Estrategias de versionado y monitoreo

Con el fin de garantizar la trazabilidad del sistema, se adoptaron prácticas de control de versiones y monitoreo continuo.

El código fuente y los scripts asociados al *pipeline* se gestionaron mediante GitHub [19], lo que permitió organizar el desarrollo de manera colaborativa y mantener un historial de cambios documentado.

Por otro lado, los modelos registrados en MLflow se acompañaron de sus métricas de validación y fecha de generación, lo que posibilitó un seguimiento temporal de su desempeño.

Además, se establecieron controles de consistencia sobre los datos de entrada y validaciones automáticas del formato de salida, lo que aseguró la estabilidad operativa del sistema en cada ejecución.

En conjunto, esta arquitectura permitió implementar un flujo de trabajo integrado, auditible y escalable, lo que garantizó la reproducibilidad de los resultados y sentó las bases para la futura incorporación de componentes en producción.

Capítulo 4

Ensayos y resultados

Este capítulo presenta los ensayos experimentales realizados para evaluar el desempeño de los modelos de recomendación desarrollados. Se analizan los resultados obtenidos a partir de diferentes enfoques, desde baselines simples hasta arquitecturas híbridas y neuronales, con el propósito de comparar su capacidad para modelar afinidades entre clientes y productos en el entorno B2B de consumo masivo.

4.1. Metodología de evaluación

El proceso de evaluación se diseñó con el objetivo de medir de forma consistente la capacidad predictiva, la estabilidad temporal y la aplicabilidad práctica de los modelos de recomendación. Para ello se adoptó un enfoque experimental reproducible, basado en ventanas móviles y métricas estandarizadas de ranking, que permite comparar distintos algoritmos bajo condiciones equivalentes de información.

Cada modelo se entrenó mediante el uso de la información histórica comprendida entre los meses $N - 7$ y $N - 2$, y se evalúa de forma prospectiva sobre el mes $N - 1$, lo que replica las condiciones reales de operación del sistema de recomendación en producción. Esta estrategia evita el uso de divisiones aleatorias del conjunto de datos y preserva la coherencia temporal entre entrenamiento y validación, aspecto crítico en dominios donde las preferencias y el portafolio evolucionan mes a mes.

Las métricas principales de evaluación fueron *Precision@K* y *Recall@K*, que miden la proporción de productos relevantes correctamente recomendados dentro del conjunto de los K primeros resultados. En particular, se empleó $K = 10$, criterio que permite concentrar el análisis en las primeras posiciones del ranking, donde se observan las recomendaciones más relevantes para el usuario final. Además, se registraron métricas complementarias de cobertura, diversidad y área bajo la curva ROC, utilizadas para analizar la robustez y el equilibrio entre exploración y explotación del sistema.

El conjunto de validación se compone de todas las combinaciones cliente–producto que registraron interacciones positivas durante el mes objetivo, mientras que las recomendaciones se generan para el universo completo de clientes activos en la ventana de entrenamiento. De esta forma, cada modelo es evaluado sobre un escenario de predicción realista, en el que se busca maximizar la recuperación de productos efectivamente comprados al tiempo que se mantiene un nivel adecuado de variedad y personalización.

La comparación entre modelos se realizó en dos niveles: mediante métricas agregadas globales, que permiten evaluar la performance general del sistema, y mediante análisis segmentados por canal, subregión y unidad de negocio, que permiten identificar diferencias estructurales en el comportamiento de los algoritmos según las características del mercado. Este esquema integral de evaluación garantiza una lectura equilibrada entre desempeño predictivo, interpretabilidad y aplicabilidad en un entorno productivo de gran escala.

Además del desempeño en términos de precisión, recuperación y calidad del ranking, la evaluación incorpora métricas diseñadas para analizar la diversidad, la cobertura y el sesgo hacia productos populares, elementos que aportan una visión más completa del comportamiento del sistema más allá de los aciertos directos. Estas medidas conforman el conjunto de métricas de performance, cuyo objetivo es cuantificar la calidad del ranking y la utilidad práctica de las recomendaciones generadas. Dichas métricas se detallan en la tabla 4.1.

TABLA 4.1. Métricas utilizadas para evaluar la calidad del ranking generado por los modelos.

Métrica	Descripción
Precision@K	Proporción de productos relevantes dentro del top- K recomendado.
Recall@K	Porcentaje de productos comprados que aparecen en el top- K .
MAP@K	Promedio de la precisión acumulada, que pondera la posición de cada acierto en el ranking.
NDCG@K	Mide la calidad de la jerarquía del ranking, al asignar mayor peso a los aciertos en posiciones superiores.
Diversity@K	Disimilitud promedio entre los productos sugeridos dentro del top- K .
Popularity Bias@K	Grado de concentración del ranking en productos de alta frecuencia histórica.

Complementariamente, se incluyen criterios de eficiencia computacional, fundamentales para estimar la viabilidad real de cada modelo en un entorno distribuido y de gran escala como BEES. Este segundo grupo evalúa tiempos de entrenamiento, tiempos de inferencia, uso de memoria y capacidad de escalado, lo que permite determinar qué enfoques son factibles de sostener en producción dadas las restricciones operativas. Las métricas asociadas a esta dimensión se presentan en la tabla 4.2.

TABLA 4.2. Indicadores utilizados para evaluar la eficiencia operativa y escalabilidad de los modelos.

Métrica	Descripción
Tiempo de entrenamiento	Duración total requerida para ajustar el modelo sobre el conjunto histórico, que incluye preprocesamiento y construcción de <i>embeddings</i> .
Tiempo de inferencia	Tiempo necesario para generar recomendaciones completas para toda la base de clientes, lo que determina la factibilidad de ejecuciones diarias o semanales.
Uso de memoria	Cantidad de memoria RAM utilizada durante las fases de entrenamiento e inferencia, con atención al tamaño del modelo y a los datos procesados.
Escalabilidad	Capacidad del modelo para mantener tiempos razonables al aumentar el volumen de datos o la cantidad de nodos en el clúster distribuido.
Costo computacional	Estimación del impacto en recursos del clúster asociado a la ejecución del modelo.

4.2. Modelos *baseline*

Con el fin de contextualizar el desempeño del sistema de recomendación propuesto, se implementaron tres enfoques *baseline* utilizados como puntos de referencia. Estos modelos establecen límites inferiores e intermedios de desempeño esperado bajo distintos niveles de información y complejidad de modelado.

4.2.1. Modelo nulo

El modelo nulo estima la probabilidad global de compra en todo el universo cliente–producto, sin incorporar ningún tipo de personalización ni información contextual. En términos operativos, calcula la razón entre el número de pares cliente–producto que registran una compra en el mes siguiente y el total de combinaciones posibles. Este *baseline* constituye una cota inferior estadística que refleja la esparsidad intrínseca de la matriz de interacciones.

4.2.2. Modelo aleatorio

El modelo aleatorio establece una referencia no informativa al ordenar aleatoriamente los productos disponibles para cada cliente. Las métricas de desempeño, como *Precision@K* y *Recall@K*, se calculan a partir de estos rankings generados de manera aleatoria. Este enfoque permite cuantificar los resultados esperados en ausencia total de estructura o aprendizaje y sirve como punto de comparación para determinar el rendimiento mínimo aceptable de cualquier modelo de recomendación.

4.2.3. Modelo basado en reglas

El tercer *baseline*, denominado modelo basado en reglas, representa un enfoque heurístico que utiliza reglas simples de agregación y frecuencia para generar recomendaciones. El método asigna puntuaciones de recomendación a partir del análisis de compras históricas sobre ventanas temporales predefinidas y segmentos de distribución específicos.

La lógica de este enfoque se estructura a partir de un conjunto de reglas secuenciales. En primer lugar, se realiza un filtrado del universo activo, con la selección únicamente de aquellos clientes y productos que registran actividad de compra entre los últimos 60 y 180 días, con el objetivo de asegurar una población representativa y actualizada. Posteriormente, se calculan las frecuencias de compra de cada producto por canal y región de distribución, lo que permite capturar patrones locales de popularidad. A continuación, se incorporan señales de preferencia en forma de indicadores binarios que identifican si un punto de venta ha interactuado previamente con productos pertenecientes a la misma unidad de negocio, familia de marca o tipo de empaque. Finalmente, todos estos componentes se combinan en un puntaje compuesto que equilibra la fuerza relativa de compra con la similitud del producto. El término base refleja la frecuencia de compra normalizada, mientras que los factores adicionales introducen pequeños ajustes que ponderan la afinidad con marcas o formatos previamente adquiridos.

Este modelo basado en reglas explota patrones observables de coocurrencia sin recurrir a factorizaciones latentes ni representaciones mediante *embeddings*. Por lo tanto, funciona como un *baseline* intermedio e interpretable entre el modelo aleatorio y los enfoques híbridos basados en aprendizaje, y proporciona una referencia significativa para evaluar el valor incremental aportado por las arquitecturas neuronales.

4.2.4. Resultados de los modelos *baseline*

La evaluación inicial se centra en tres modelos de referencia que permiten establecer límites inferiores de desempeño y cuantificar la ganancia relativa obtenida por los enfoques avanzados. Estos modelos base sirven para contextualizar el aporte incremental de la personalización y la representación latente frente a estrategias puramente estadísticas o heurísticas.

Los resultados obtenidos se resumen en la tabla 4.3.

TABLA 4.3. Resumen de métricas de desempeño promedio para los modelos de referencia durante los tres períodos de evaluación.

Modelo	Precision@10	Recall@10
Modelo nulo	11,4 %	100,0 %
Modelo aleatorio	10,9 %	12,5 %
Modelo basado en reglas	16,3 %	18,6 %

4.2.5. Comparación global de desempeño frente a los *baseline*

La figura 4.1 y 4.2 presentan la comparación directa entre los modelos avanzados y los tres *baseline* evaluados.

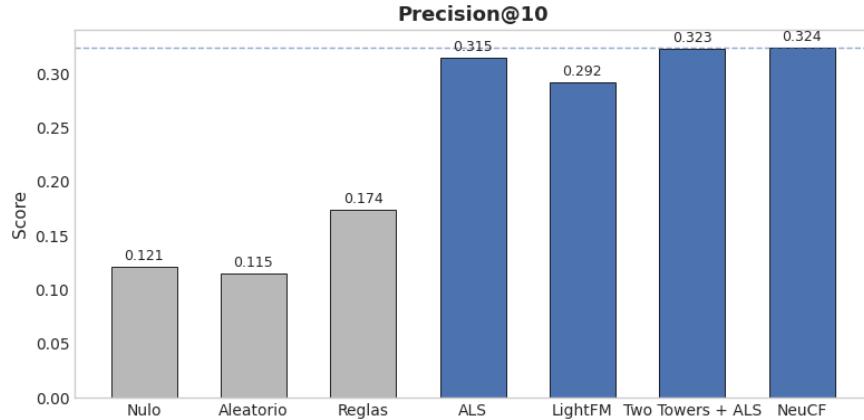


FIGURA 4.1. Comparación global de *Precision@10* entre los modelos avanzados y los modelos de referencia.

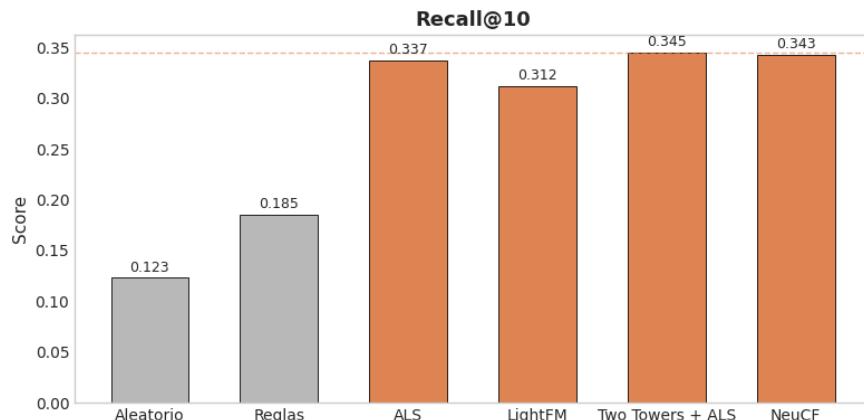


FIGURA 4.2. Comparación global de *Recall@10* entre los modelos avanzados y los modelos de referencia.

En términos de *Precision@10*, los modelos basados en aprendizaje superan ampliamente a los enfoques no informados. Mientras que los baseline alcanzan valores entre el 11 % y 17 %, los modelos ALS, LightFM y las arquitecturas neuronales logran precisiones superiores al 29 %, con picos de 32,3 % en el modelo *Two Towers + ALS* y 32,4 % en el modelo *NeuCF*.

El comportamiento es consistente en *Recall@10*. Los baseline mantienen valores moderados (12–18 %), mientras que los modelos avanzados superan ampliamente este nivel. En particular, ALS alcanza un 33,7 %, seguido por *Two Towers + ALS* con 34,5 % y *NeuCF* con 34,3 %. Estos resultados confirman que las arquitecturas híbridas y neuronales permiten recuperar una mayor proporción de compras efectivas dentro del ranking recomendado.

En conjunto, todos los enfoques evaluados superan de manera consistente a los modelos *benchmark* en todas las métricas consideradas. Esto confirma que incluso los modelos más simples basados en factores latentes ya capturan información estructural que los baseline no pueden modelar, mientras que las arquitecturas híbridas y neuronales logran mejorar el desempeño al explotar patrones no lineales y relaciones implícitas en los datos.

4.3. Análisis de resultados

Esta sección presenta una síntesis integrada del comportamiento de los modelos evaluados, y articula tanto métricas de calidad del ranking como indicadores de eficiencia operativa. En primer lugar, se analizan los resultados de performance, y se compara la capacidad de cada enfoque para recuperar productos relevantes, ordenar correctamente el ranking y generar recomendaciones diversas y no sesgadas. A continuación, se examinan los aspectos de eficiencia computacional, con atención a los tiempos de entrenamiento e inferencia, el uso de memoria y la escalabilidad en el entorno distribuido de Databricks. Finalmente, se ofrece una conclusión conjunta que resume los hallazgos y destaca el aporte relativo de cada modelo dentro del ecosistema de recomendaciones evaluado.

4.3.1. Resultados de performance

La tabla 4.4 presenta una comparación integral del desempeño de los modelos avanzados evaluados, que considera las métricas de ranking, diversidad y sesgo hacia productos populares. Los resultados permiten observar diferencias estructurales entre los enfoques colaborativos, híbridos y neuronales, así como su mejora sustancial respecto de las líneas base discutidas previamente.

TABLA 4.4. Resumen comparativo demétricas de performance promedio para los modelos evaluados.

Modelo	Precision@10	Recall@10	MAP@10	NDCG@10	Diversity@10	Popularity Bias@10
ALS	31,5 %	33,7 %	52,1 %	68,0 %	90,9 %	14,5 %
LightFM	29,2 %	31,2 %	54,7 %	69,7 %	82,1 %	26,9 %
Two Towers + ALS	32,3 %	34,5 %	50,1 %	66,5 %	89,9 %	15,5 %
NeuCF	32,4 %	34,3 %	40,4 %	57,3 %	88,9 %	16,0 %

En términos de precisión y recuperación, los cuatro modelos confirman la relevancia de incorporar representaciones latentes y arquitecturas basadas en *embeddings*. El modelo ALS se destaca por su solidez y alcanza un 31,5 % de *Precision@10* y un 33,7 % de *Recall@10*. Los modelos neuronales presentan mejoras marginales: *Two Towers* y *NeuCF* obtienen los valores más altos, con una *Precision@10* cercana al 32,3–32,4 % y una *Recall@10* de hasta 34,5 %.

Por otro lado, las métricas de ranking fino muestran comportamientos diferenciados. Si bien *LightFM* no alcanza los niveles de precisión global de los modelos profundos, presenta el mejor desempeño en *MAP@10* (54,7 %) y *NDCG@10* (69,7 %), lo que indica una mejor jerarquización de los primeros lugares del ranking. Este resultado sugiere que su arquitectura híbrida beneficia especialmente la calidad posicional de las recomendaciones.

La diversidad del top-10 se mantiene elevada en todos los enfoques, con valores entre 82 % y 91 %, lo que evidencia que los modelos no colapsan hacia un conjunto reducido de productos, incluso en un entorno altamente concentrado como el mercado B2B. En cuanto al *popularity bias*, los resultados muestran que los modelos avanzados logran contener el sesgo hacia productos de alta rotación: ALS, *Two Towers* y *NeuCF* mantienen niveles bajos (14–16 %), mientras que *LightFM* exhibe mayor inclinación hacia ítems populares (26,9 %).

Si bien cada arquitectura presenta fortalezas particulares, la convergencia en métricas muestra que los modelos avanzados son capaces de capturar de manera efectiva las relaciones complejas entre clientes y productos, y ofrecen una mejora sustancial sobre heurísticas tradicionales.

4.3.2. Resultados de eficiencia computacional

La comparación entre modelos no se limita al desempeño en las métricas de ranking, sino que también considera su viabilidad operativa dentro del entorno distribuido de Databricks. Todos los experimentos se ejecutaron sobre un clúster Standard_E16a_v4 (Spark 3.5, 16 núcleos y 128 GB de memoria por nodo), con entre uno y cuatro *workers* asignados según la carga de trabajo requerida por cada arquitectura.

La tabla 4.5 sintetiza estos resultados en términos de tiempo de entrenamiento, tiempo de inferencia, uso de memoria y costo computacional relativo.

TABLA 4.5. Resumen comparativo de eficiencia computacional para los modelos evaluados.

Modelo	Tiempo de entrenamiento	Tiempo de inferencia	Uso de memoria	Costo relativo
ALS	~40–60 min	~10–15 min	Bajo	1,0×
<i>LightFM</i>	~1,5–2 h	~15–20 min	Medio	1,3×
<i>Two Towers + ALS</i>	~2,5–3 h	~20–30 min	Alto	1,7×
<i>NeuCF</i>	~3–4 h	~20–30 min	Alto	2,0×

A partir de los tiempos observados durante los entrenamientos, el modelo *Neural Collaborative Filtering* se posiciona como el enfoque más costoso desde el punto de vista computacional. El ajuste de cinco épocas sobre aproximadamente 39 millones de interacciones demanda entre 3 y 4 horas de procesamiento, que incluye la fase previa de construcción del conjunto de entrenamiento y la serialización del modelo.

El enfoque *Two Towers* combinado con ALS presenta un costo intermedio: el entrenamiento completo, que comprende la factorización implícita con ALS y la posterior optimización de las dos torres neuronales basadas en atributos, requiere entre 2,5 y 3 horas en total.

En contraste, el modelo ALS puro muestra una eficiencia computacional significativamente mayor y completa su entrenamiento en aproximadamente 40–60 minutos gracias a su ejecución nativa y distribuida sobre Spark.

La variante híbrida *LightFM* se ubica en un punto medio, con tiempos de entrenamiento del orden de 1,5–2 horas, consistentes con la necesidad de mover datos fuera del motor distribuido y optimizar parámetros en un entorno no nativamente paralelo.

En la etapa de inferencia, todos los enfoques resultan operativamente viables para su ejecución mensual a escala completa. ALS y *LightFM* generan el ranking completo para toda la base en aproximadamente 10–20 minutos, mientras que *Two Towers* y *NeuCF*, que dependen de componentes en PyTorch y funciones UDF

para el cálculo de similitudes, se sitúan entre 20 y 30 minutos, manteniéndose dentro de una ventana razonable para despliegues periódicos en producción.

En conjunto, los resultados muestran que las arquitecturas basadas en Spark (ALS) son significativamente más eficientes en entornos distribuidos, mientras que los modelos neuronales ofrecen mayor capacidad expresiva a costa de mayores recursos y tiempos de cómputo.

4.3.3. Evaluación final del desempeño y eficiencia

Los resultados indican que las arquitecturas neuronales obtienen las mejores métricas de *Precision@10* y *Recall@10*, lo que confirma su capacidad para modelar relaciones no lineales y capturar interacciones complejas entre clientes y productos. Sin embargo, la magnitud de estas mejoras es relativamente acotada: los incrementos respecto de ALS oscilan típicamente entre 0.5 y 1.0 puntos porcentuales, una ganancia real pero moderada dentro del contexto operativo del negocio.

Este avance marginal se contrapone a un costo computacional claramente superior. Los modelos neuronales requieren varias horas de entrenamiento, un volumen importante de memoria y la ejecución de componentes en PyTorch dentro de un entorno distribuido, lo que introduce mayor complejidad en la orquestación, el mantenimiento y la reproducibilidad. En escenarios donde las recomendaciones deben generarse de forma periódica y estable, estos costos adicionales no se traducen en ganancias lo suficientemente significativas como para justificar su adopción como modelo principal.

En contraste, ALS exhibe el mejor balance global entre desempeño y eficiencia. Aunque sus métricas no alcanzan el máximo absoluto, se mantienen muy competitivas y lo suficientemente cercanas a las variantes neuronales como para que, en la práctica, las diferencias tengan impacto limitado en la calidad final de las recomendaciones. Su entrenamiento nativo en Spark, su bajo uso de memoria y su velocidad lo posicionan como una solución altamente costo–efectiva y confiable para pipelines recurrentes de producción.

En conjunto, estos resultados sugieren que, si bien las arquitecturas más complejas pueden ofrecer pequeñas mejoras incrementales, ALS logra el punto óptimo entre calidad predictiva y costo operativo. En este contexto, la simplicidad, estabilidad y escalabilidad del ALS superan la ventaja marginal de performance de los modelos neuronales, consolidándolo como la opción más adecuada para un entorno productivo de gran escala.

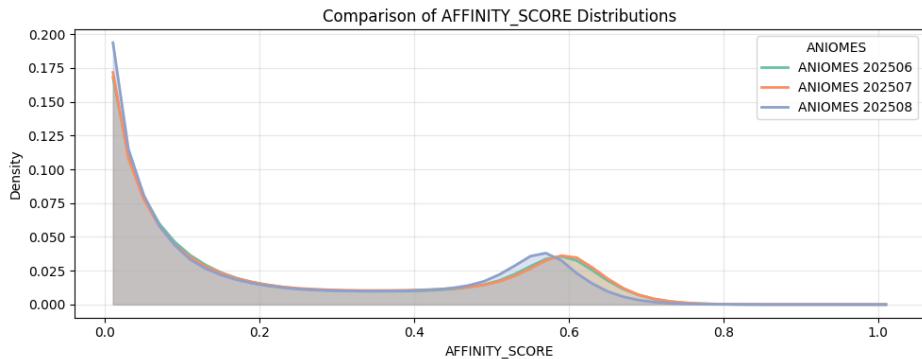
4.4. Análisis de robustez

El análisis de robustez evalúa si el desempeño del modelo se mantiene estable bajo diferentes condiciones temporales, segmentos comerciales y niveles de historial disponible. La estabilidad temporal se observa en la tabla 4.6, donde las métricas de *Precision@10* y *Recall@10* presentan variaciones mínimas entre los meses 202506 y 202508. Estas fluctuaciones acotadas indican que el modelo no es sensible a cambios estacionales ni a variaciones propias del ciclo comercial mensual.

TABLA 4.6. *Precision@10* y *Recall@10* por mes de evaluación.

Año-Mes	Precision@10	Recall@10
202506	29.9 %	34.1 %
202507	30.1 %	33.5 %
202508	31.5 %	33.7 %

Esta estabilidad también se refleja en la distribución del *affinity score*, representada en la figura 4.3. Las curvas correspondientes a los tres meses prácticamente se superponen, lo que confirma la ausencia de *data drift* tanto en la construcción de señales como en el pipeline de entrenamiento.

FIGURA 4.3. Distribuciones comparadas del *AFFINITY_SCORE* para los meses 202506, 202507 y 202508.

Al segmentar los resultados por unidad de negocio, se observa un comportamiento igualmente estable. La tabla 4.7 muestra que CZA y NABS mantienen niveles consistentes de precisión y recall entre meses, mientras que MATCH presenta un recall elevado y una precisión inferior. Lo importante es que estas diferencias permanecen prácticamente constantes en el tiempo, lo que demuestra que el modelo no amplifica variaciones estructurales del negocio.

TABLA 4.7. *Precision@10* y *Recall@10* por unidad de negocio.

Año-Mes	Unidad de negocio	Precision@10	Recall@10
202506	CZA	24.4 %	52.9 %
202506	MATCH	17.8 %	83.9 %
202506	NABS	25.5 %	48.7 %
202507	CZA	25.8 %	52.3 %
202507	MATCH	17.3 %	82.9 %
202507	NABS	25.8 %	47.5 %
202508	CZA	26.9 %	52.2 %
202508	MATCH	16.9 %	82.5 %
202508	NABS	26.4 %	47.4 %

Finalmente, el análisis por tipo de cliente (tabla 4.8) revela patrones coherentes con la disponibilidad de historial: los clientes nuevos presentan métricas más bajas debido a la escasez de señales, los inestables alcanzan valores intermedios y

los estables obtienen los mejores resultados, con precisiones superiores al 76 % y *recall* cercanos al 87 %. Lo relevante es que estas relaciones se mantienen prácticamente inalteradas entre los meses analizados, lo que refuerza la consistencia del modelo.

TABLA 4.8. *Precision@10* y *Recall@10* según tipo de cliente.

Año-Mes	Clasificación	<i>Precision@10</i>	<i>Recall@10</i>
202506	nuevos	10.4 %	39.2 %
202506	estables	76.5 %	86.5 %
202506	inestables	35.1 %	63.9 %
202507	nuevos	11.6 %	37.8 %
202507	estables	76.2 %	87.4 %
202507	inestables	36.8 %	63.9 %
202508	nuevos	12.1 %	37.5 %
202508	estables	76.2 %	87.6 %
202508	inestables	38.3 %	64.6 %

En conjunto, estos resultados confirman que el modelo es robusto en todas las dimensiones evaluadas: mantiene un desempeño estable en el tiempo, responde de forma consistente entre segmentos comerciales y opera de manera predecible según la disponibilidad de historial del cliente. Esta estabilidad es un requisito clave para su uso en producción y garantiza que el sistema no introduce variabilidad artificial ni deriva comportamientos inesperados en ciclos operativos mensuales.

Capítulo 5

Conclusiones

Este capítulo presenta las conclusiones generales del trabajo y sintetiza los principales aportes derivados del desarrollo del motor de afinidad. A partir de la integración de múltiples fuentes de datos, técnicas avanzadas de preprocesamiento y la comparación sistemática de distintos enfoques de recomendación, se establecen aquí las lecciones centrales obtenidas, el valor añadido por la solución propuesta y su relevancia dentro del ecosistema comercial de BEES. Asimismo, se discuten las oportunidades de mejora y las líneas de trabajo futuras que permitirían ampliar el alcance del sistema, fortalecer su desempeño y consolidar su impacto operativo.

5.1. Conclusiones generales

El trabajo realizado permitió diseñar, construir y validar un motor de afinidad integral para personalización de portafolio en un entorno B2B de consumo masivo, caracterizado por alta heterogeneidad, fuerte concentración en productos y clientes, y un nivel de rotación y estacionalidad significativamente mayor que en los benchmarks clásicos de recomendación. A lo largo del proyecto se desarrolló una solución completa que abarca desde la consolidación y depuración de múltiples fuentes de datos hasta la implementación y comparación de modelos avanzados, que incorpora además prácticas modernas de ingeniería y MLOps necesarias para operar a gran escala.

Un primer aporte central del trabajo fue la integración de un conjunto diverso de fuentes informativas: transacciones históricas, señales digitales, atributos estructurales de clientes y productos y ventanas temporales de comportamiento. Estas fuentes fueron sometidas a un proceso riguroso de curación, normalización, discretización y estandarización que permitió construir insumos robustos, comparables y libres de sesgos extremos de escala. El diseño de la matriz cliente–producto, junto con el score compuesto derivado de seis meses de señales ponderadas, constituye un artefacto original del proyecto y un insumo que no existía previamente en la operación.

El segundo aporte clave se encuentra en la profundidad del preprocesamiento y en la construcción sistemática de atributos contextuales. El trabajo no se limitó a generar una matriz de interacciones, sino que modeló bloques completos de features de negocio para clientes y productos, evaluó su calidad informativa mediante análisis de varianza, correlación, similitud geométrica y PCA, y demostró que estas representaciones capturan patrones reales del comportamiento comercial. Este análisis permitió no solo mejorar el desempeño de los modelos híbridos,

sino también aportar un entendimiento más profundo de la estructura comercial del ecosistema.

En términos de modelado, el proyecto avanzó de manera progresiva desde métodos clásicos hasta arquitecturas modernas. Se implementaron *baselines*, un modelo ALS optimizado, modelos híbridos con *LightFM*, una arquitectura *Two-Tower* con embeddings aprendidos y un modelo neuronal de tipo NCF, que combina enfoques colaborativos, de contenido y de aprendizaje profundo. La comparación sistemática de estos modelos y su evaluación temporal permitió identificar claramente el aporte incremental de cada técnica y construir una solución que balancea precisión, diversidad y costo computacional en un contexto real de negocio.

A diferencia de los *benchmarks* tradicionales, basados en datasets densos, abundancia de señales explícitas y comportamiento de consumo individual, este trabajo enfrentó un contexto B2B con interacción dispersa, señales implícitas ruidosas, productos con vida útil acotada y clientes con ciclos de compra irregulares. El motor de afinidad desarrollado se diferencia de los estándares B2C al incorporar criterios de negocio, representaciones híbridas, tratamiento explícito del arranque en frío y un pipeline de validación temporal, elementos imprescindibles para que las recomendaciones sean útiles en un entorno distribuido y operativo como el de consumo masivo.

Otro aporte distintivo del proyecto fue el desarrollo de una infraestructura completa de MLOps, que incluye el pipeline de preparación, entrenamientos reproducibles en Spark, optimización automática de hiperparámetros con Optuna, registración estructurada de artefactos en MLflow, versionado de modelos y diseño conceptual de despliegue. Esta dimensión técnica asegura que el sistema no sea únicamente un prototipo académico, sino una solución realista, escalable y alineada con los requerimientos del entorno de producción.

Finalmente, el análisis de robustez temporal demostró que el sistema mantiene estabilidad y generalización frente a variaciones mensuales del comportamiento, diferencias estructurales entre unidades de negocio y heterogeneidad entre clasificaciones de clientes. Este comportamiento consistente valida la solidez del motor de afinidad y confirma que la integración de múltiples señales y atributos permite capturar patrones de demanda relevantes incluso en entornos altamente volátiles.

El trabajo aporta una solución integral que combina rigurosidad técnica, viabilidad operativa y comprensión profunda del negocio, y constituye un avance relevante tanto para la empresa como para la literatura aplicada de sistemas de recomendación en entornos B2B de consumo masivo.

5.2. Próximos pasos

Si bien el motor de afinidad desarrollado constituye una solución robusta, escalable y ya apta para operación productiva, existen múltiples líneas de evolución que pueden ampliar su alcance, mejorar su precisión y potenciar su impacto en negocio. Estas oportunidades abarcan la incorporación de nuevas señales, el desarrollo de representaciones más ricas, el avance hacia modelos neuronales especializados, la optimización de infraestructura y la validación experimental a gran escala.

En primer lugar, un camino natural consiste en enriquecer el conjunto de señales que incorpora fuentes adicionales aún no explotadas. Entre ellas se destacan elasticidades de precio por cliente, indicadores de disponibilidad en tiempo real, atributos derivados del calendario promocional y métricas detalladas del *funnel* digital, como secuencias de navegación. La literatura reciente demuestra que la combinación de señales transaccionales, contextuales y secuenciales mejora de forma sustantiva la capacidad de predicción en entornos de comercio electrónico y retail [33, 34, 23]. Integrar estas fuentes en el score compuesto permitiría capturar capas de comportamiento más finas y anticipar variaciones sensibles al contexto comercial.

En segundo lugar, resulta especialmente prometedor avanzar hacia métodos de segmentación dinámica y clustering avanzado que capten similitudes estructurales entre clientes y productos. El uso de técnicas como HDBSCAN [35], clustering jerárquico o modelos basados en densidad permitiría identificar subpoblaciones relevantes que no emergen de forma explícita del análisis transaccional tradicional [36, 37]. Estas estructuras pueden utilizarse para construir representaciones grupales o *cluster embeddings*, que combina atributos individuales con patrones colectivos, lo que facilita una generalización más sólida en escenarios de datos escasos o productos nuevos.

Un tercer eje de desarrollo se orienta a explorar arquitecturas de modelado más expresivas. Los modelos secuenciales, tales como GRU4Rec [38], SASRec [39] o variantes Transformer especializadas en recomendación, han demostrado una capacidad superior para capturar dependencias temporales y dinámicas de corto plazo. Su incorporación permitiría ir más allá del esquema estático de seis meses utilizado en este trabajo, y ofrece recomendaciones sensibles a la evolución reciente del comportamiento del cliente. Asimismo, las técnicas multimodales basadas en embeddings visuales y textuales [40, 41] abren la puerta a mejorar la capacidad del sistema para generalizar ante productos con poco historial o recientemente introducidos al portafolio.

En términos de arquitectura, es relevante investigar modelos híbridos que combinen la escalabilidad y estabilidad de ALS con la expresividad de los modelos neuronales. La integración entre *Two-Tower* y ALS, por ejemplo, puede derivar en esquemas de entrenamiento conjunto donde los embeddings aprendidos alimenten directamente la factorización matricial, lo que reduce inconsistencias entre espacios latentes y aumenta la precisión final. Adicionalmente, técnicas de *knowledge distillation* [42, 43] permitirían transferir el comportamiento de modelos complejos hacia modelos más livianos y rápidos, que disminuye el costo computacional sin sacrificar calidad.

Otra línea de mejora se vincula con la optimización de la infraestructura computacional. El uso de GPUs o *clusters* heterogéneos podría acelerar significativamente los modelos neuronales; mientras que la incorporación de particionamiento inteligente, difusión selectiva de embeddings (*broadcast*) y UDFs optimizadas contribuiría a reducir tiempos y costos. La adopción de componentes como Delta Live Tables, Feature Store o *pipelines* CI/CD basados en MLflow Model Registry reforzaría la gobernanza y automatización del sistema, alineándose con las mejores prácticas de ingeniería de ML [44, 45].

Finalmente, un paso esencial es avanzar hacia una validación directa en negocio mediante experimentos A/B, pilotos geográficos o pruebas controladas en

segmentos específicos. La literatura de experimentación controlada [46, 47, 48] demuestra que este enfoque es fundamental para medir impacto real en métricas operativas como crecimiento incremental de volumen, expansión del surtido, aumento del ticket promedio o reducción de compras erráticas. Complementariamente, la incorporación de explicabilidad en las recomendaciones fortalecería la adopción por parte de los equipos comerciales y mejoraría la interacción entre usuarios y el sistema.

En conjunto, estos desarrollos delinean una hoja de ruta clara para evolucionar el motor de afinidad hacia una plataforma más precisa, inteligente, multimodal y estrechamente alineada con la dinámica comercial, lo que amplía su valor estratégico dentro del ecosistema de recomendación B2B.

Apéndice A

Optimización de pesos por evento y ventana temporal

Este anexo documenta los resultados del proceso de optimización bayesiana realizado con Optuna, cuyo objetivo fue estimar los pesos relativos de cada tipo de evento y de cada ventana temporal para la construcción del *score* de preferencia cliente–producto. El procedimiento se orientó a maximizar la métrica *Precision@10* sobre un conjunto de validación temporal, lo que garantizó un balance adecuado entre la relevancia de las señales recientes y la estabilidad de los patrones históricos.

A.1. Estrategia de optimización

La optimización se ejecuta mediante búsqueda bayesiana a lo largo de 100 iteraciones. En cada *trial* se ajustan simultáneamente los pesos temporales y los pesos por tipo de evento (α_e), aplicándolos en la generación del *score* compuesto de afinidad. La métrica de desempeño se calcula mediante el uso de ventanas móviles de seis meses, con el entrenamiento basado en el histórico comprendido entre los meses $N - 7$ y $N - 2$, y la validación de la capacidad predictiva sobre el mes $N - 1$.

Los resultados evidencian un patrón consistente: las señales más recientes (1M) aportan mayor información predictiva que las históricas (6M), y los eventos transaccionales tienden a tener un peso superior al de los digitales, especialmente aquellos asociados a recomendaciones personalizadas o flujos de recompra recurrente.

A.2. Pesos óptimos por ventana temporal

La tabla A.1 presenta los valores óptimos obtenidos para cada horizonte temporal. Se observa una clara preferencia hacia las señales más recientes, lo que indica que los comportamientos de compra recientes aportan mayor valor predictivo que los históricos, en línea con la dinámica de rotación del portafolio en el entorno B2B.

TABLA A.1. Pesos óptimos obtenidos para cada horizonte temporal.

Ventana temporal	Peso (β_w)
1 mes (reciente)	0,3925
3 meses (intermedia)	0,4735
6 meses (larga)	0,1340

A.3. Pesos óptimos por tipo de evento

En la tabla A.2 se muestran los pesos óptimos estimados para cada tipo de evento. Las señales vinculadas a compras efectivas y órdenes generadas (BUYER, ordered_*) son las más relevantes, seguidas por aquellas relacionadas con interacciones promocionales o de exposición de producto. Este patrón refuerza la importancia de las señales transaccionales en la predicción de recompra y en la calibración del *score* de afinidad.

TABLA A.2. Pesos óptimos estimados para cada tipo de evento (*event_weights*).

Evento	Peso (α_e)
BUYER	0,2208
ordered_QUICK_ORDER	0,1170
ordered	0,0700
card_viewed_QUICK_ORDER	0,1318
card_viewed_FORGOTTEN_ITEMS	0,0630
details_page_viewed	0,0066
card_viewed_CROSS_SELL_UP_SELL	0,0916
ordered_CROSS_SELL_UP_SELL	0,1882
ordered_FORGOTTEN_ITEMS	0,2174
ordered_RECENT_SEARCHES	0,0734
ordered_CLUB_B	0,1008
ordered_POPULAR_SEARCHES	0,1896
removed	0,1228
card_viewed_RECENT_SEARCHES	0,1786
card_viewed_POPULAR_SEARCHES	0,2028
card_viewed	0,0132
card_viewed_CLUB_B	0,0128

A.4. Análisis e interpretación

Los pesos reflejan una jerarquía coherente con el proceso de compra en la plataforma BEES: las órdenes efectivas (BUYER, ordered_*) constituyen las señales más predictivas de recompra, seguidas por las interacciones promocionales y de exposición de producto (card_viewed_*). Las categorías vinculadas a mecanismos de recomendación específicos, como *Cross-Sell/Up-Sell* y *Forgotten Items*, presentan una fuerte correlación con la conversión, lo que respalda su priorización en el cálculo del *score* final.

En contraste, los eventos de exploración (`details_page_viewed`) o de fricción (`removed`) tienen menor peso, lo que confirma que su valor informativo es limitado cuando se los considera de forma aislada.

El conjunto de ponderaciones resultante se utiliza en la construcción del *score* de preferencia compuesto, definido en la ecuación A.1 donde α_e representa el peso por tipo de evento y β_w el peso temporal.

$$\text{Score}_{ij} = \sum_{e \in E} \sum_{w \in W} \alpha_e \cdot \beta_w \cdot x_{ij}^{(e,w)} \quad (\text{A.1})$$

Este valor resume la intensidad y actualidad de las interacciones del cliente i con el producto j , y constituye el insumo principal de la matriz cliente–producto empleada en los modelos de recomendación.

Apéndice B

Ensayos experimentales del modelo LightFM

Este anexo detalla las tres configuraciones experimentales del modelo *LightFM* desarrolladas con el fin de analizar el aporte incremental de las variables contextuales y la calidad de las representaciones latentes generadas. Cada *test* incorpora progresivamente mayor complejidad en las features de usuario y producto, y conserva la misma arquitectura base y función de pérdida *WARP*.

B.1. *Test 1 – Modelo base con contexto categórico reducido*

El primer experimento con el modelo *LightFM* se diseñó como una prueba inicial para evaluar el aporte del contexto estructural más básico sobre la capacidad predictiva del sistema. En esta configuración, se incorporaron únicamente atributos categóricos que describen propiedades intrínsecas de los clientes y productos, sin incluir aún las variables derivadas del comportamiento transaccional histórico.

La representación vectorial de cada cliente se construyó a partir de tres campos estructurales: la subregión, el canal de venta y un indicador binario de venta de alcohol. Estas variables permiten capturar diferencias sistemáticas entre tipos de puntos de venta en función de su localización y mix comercial, dos factores que influyen fuertemente en los patrones de compra observados.

Para los productos, se incluyeron seis descriptores fundamentales: la familia de marca, el tipo de empaque, el segmento de marca, la unidad de negocio y el indicador de contenido alcohólico. Estos campos resumen la posición del producto dentro del portafolio y su rol dentro del surtido, aspectos clave para modelar afinidades en un contexto B2B.

Todas las variables se codificaron como entidades categóricas y se proyectaron en el espacio latente del modelo mediante *feature embeddings*. El entrenamiento se realizó con el uso de la función de pérdida *WARP* (*Weighted Approximate-Rank Pairwise*), orientada a optimizar directamente la posición relativa de los productos relevantes dentro del ranking de recomendaciones. Este criterio, ampliamente adoptado en contextos de *feedback* implícito [30], penaliza de forma más intensa los errores en las primeras posiciones del ranking, lo que favorece la recuperación de ítems con mayor probabilidad de interacción.

Aun con esta configuración reducida, centrada exclusivamente en variables categóricas estáticas, el modelo alcanzó una *Precision@10* de 25,3 % y *Recall@10* de 27,1 %.

Esto evidencia que incluso en ausencia de señales históricas, las relaciones estructurales entre clientes y productos contienen información predictiva relevante, capaz de guiar recomendaciones con un nivel competitivo de precisión. Las recomendaciones generadas mostraron coherencia semántica, ya que agruparon puntos de venta del mismo canal con surtidos de características similares en empaque, segmento o tipo de negocio.

En conjunto, este primer ensayo permitió validar la arquitectura híbrida de LightFM en su forma más simple, y estableció una línea base a partir de la cual se incorporaron progresivamente variables discretizadas y filtros de relevancia en las siguientes configuraciones.

B.2. **Test 2 – Modelo con selección explicativa de atributos discretizados**

El segundo experimento extendió la configuración inicial mediante la incorporación de un conjunto ampliado de atributos categóricos, seleccionados a partir de un proceso sistemático de análisis estadístico y reducción de redundancia. El objetivo fue evaluar si la inclusión de variables discretizadas derivadas del comportamiento histórico y de la estructura de mercado mejoraba la capacidad predictiva del modelo.

Para la selección preliminar de variables se aplicaron pruebas de independencia Chi-cuadrado sobre las variables categóricas y coeficientes de correlación de *Spearman* sobre las numéricas, dando prioridad a aquellas con mayor fuerza de asociación con la variable objetivo de compra. Posteriormente, se eliminó la multicolinealidad mediante un umbral de correlación absoluta de $|r| > 0,85$, y se preservaron únicamente las variables más representativas y no redundantes. Este procedimiento permitió conformar un subconjunto interpretativo de atributos explicativos que combinan señales de estabilidad, volumen y diversidad, sin introducir ruido ni sobreajuste.

En el caso de los clientes, se agregaron descriptores de estabilidad temporal y madurez, como la consistencia de ventas y la cantidad de meses consecutivos con actividad; medidas de especialización, como los indicadores binarios de exclusividad por unidad de negocio y variables de estructura del portafolio, entre las que se incluyen la diversidad de marcas, el número de categorías y el ticket promedio. También se incorporaron proporciones asociadas a segmentos específicos, como la participación de cervezas premium y de bebidas sin alcohol dentro del mix de cada cliente.

En el caso de los productos, se sumaron atributos relacionados con su desempeño y alcance, tales como la penetración de clientes, el volumen total vendido, la diversidad de clientes y la proporción de ventas dentro de la unidad de negocio cervecera. Todas las variables numéricas fueron previamente discretizadas en tres cuantiles (*low*, *medium* y *high*), lo que aseguró comparabilidad y robustez frente a escalas heterogéneas.

El modelo se entrenó bajo la misma configuración general, con el uso de la función de pérdida WARP. La métrica *Precision@10* fue de 26,1 % y el *Recall@10* de 27,8 %, lo que mejoró significativamente el desempeño del modelo base.

El análisis cualitativo de las recomendaciones reveló una mayor capacidad de segmentación: los clientes con comportamiento estable y surtido diverso recibieron sugerencias más alineadas con su perfil, mientras que los puntos de venta especializados tendieron a recibir productos consistentes con su unidad de negocio principal. Este resultado confirma el valor de las señales discretizadas y explicativas en la construcción de representaciones híbridas, lo que fortalece la capacidad del modelo para capturar patrones de preferencia más finos sin comprometer la generalización.

B.3. Test 3 – Análisis y depuración de representaciones latentes

El tercer experimento tuvo como objetivo evaluar la calidad de las representaciones aprendidas por el modelo LightFM y depurar el conjunto de atributos en función de su contribución efectiva al espacio de *embeddings*. A diferencia de los ensayos anteriores, en esta etapa se analizaron directamente las propiedades geométricas de los vectores generados durante el entrenamiento, con el empleo de métricas de norma, cohesión y separación para cuantificar la estructura latente subyacente.

Se diseñó un *pipeline* específico de evaluación de *embeddings*, capaz de extraer los vectores de usuario y producto almacenados en los artefactos del modelo, calcular estadísticas de dispersión y estimar su coherencia semántica. Las normas promedio de los *embeddings* de cliente y producto fueron de 0,031 y 0,118 respectivamente, con baja dispersión intra-grupo, lo que indica una adecuada regularización y una representación estable. La relación entre similitudes intra e inter-categoría fue de 1,17, lo que evidencia que los productos tienden a agruparse coherentemente dentro de sus unidades de negocio, sin perder capacidad de generalización hacia otros segmentos.

Sobre esta base se aplicó un análisis de importancia de atributos, en el cual se reconstruyeron los *embeddings* asociados a cada *feature* explícita del modelo. La norma del vector correspondiente a cada atributo fue utilizada como indicador de relevancia latente: las características con mayor norma poseen mayor influencia en la formación del espacio de representación. Este análisis permitió identificar qué variables aportaban mayor poder discriminante y cuáles eran redundantes o marginales.

En el caso de los clientes, los atributos con mayor norma promedio correspondieron a las variables asociadas al tipo de canal comercial, la venta de productos con alcohol, la proporción de bebidas sin alcohol y la participación de cervezas premium dentro del portafolio. Estos resultados reflejan la relevancia de las señales estructurales y de composición del surtido en la caracterización de la afinidad entre puntos de venta y productos.

Entre los productos, las dimensiones más relevantes fueron la diversidad de clientes, el volumen total vendido, la cantidad de puntos de venta alcanzados y la penetración promedio, lo que evidencia una fuerte relación entre el alcance comercial y la estabilidad de la demanda en la estructura latente aprendida por el modelo.

A partir de este análisis se implementó un proceso de selección automática, en el cual se conservaron únicamente aquellas variables cuya norma superaba el 20 %

del valor máximo dentro de su grupo y hasta dos categorías por campo. El resultado fue un conjunto final de 28 atributos de cliente y 4 de producto, lo que conformó una base más parsimoniosa y explicable. Entre los factores retenidos se destacan las dimensiones de estabilidad temporal, diversidad de surtido y volumen de compra, que capturan aspectos complementarios del comportamiento de los puntos de venta y resultan fundamentales para modelar su propensión a interactuar con distintos productos.

La evaluación del modelo sobre el conjunto de validación temporal evidenció un desempeño estable en comparación con las configuraciones previas, con una *Precision@10* de 25,8 % y un *Recall@10* de 27,5 %. Si bien las métricas se mantuvieron en niveles similares, la depuración de atributos permitió alcanzar una representación más compacta y explicable sin comprometer la capacidad predictiva del sistema.

Este refinamiento redujo la complejidad del modelo y mejoró su interpretabilidad, al identificar de forma explícita las señales con mayor contribución a la estructura latente de afinidad. En conjunto, los resultados validan que las representaciones generadas por LightFM capturan de manera coherente los patrones de relación entre clientes y productos, lo que preserva la semántica de las variables originales y establece una base sólida para futuras extensiones y modelos de mayor complejidad.

B.4. Resultados comparativos de los ensayos experimentales

Los tres experimentos permitieron validar y refinar progresivamente la arquitectura híbrida basada en LightFM. El primer modelo, centrado en atributos estructurales, demostró que la información contextual básica es suficiente para capturar afinidades significativas entre clientes y productos. El segundo ensayo confirmó el valor de incorporar variables discretizadas de comportamiento y desempeño, lo que mejoró la capacidad de segmentación y la precisión de las recomendaciones. Finalmente, el tercer experimento consolidó la robustez del enfoque al identificar, mediante el análisis geométrico de los *embeddings*, un subconjunto reducido de variables con alta relevancia latente, y logró un equilibrio óptimo entre explicabilidad y rendimiento.

En términos globales, las métricas de desempeño mostraron un comportamiento estable y competitivo entre los distintos ensayos, con valores de *Precision@10* y *Recall@10* cercanos al 26 % y 28 %, respectivamente, tal como se resume en la tabla B.1. Estos resultados confirman la capacidad del modelo para capturar relaciones no lineales y de alta dimensionalidad en entornos con señales implícitas dispersas, y ofrecen un punto de partida sólido para su ajuste fino y extensión futura.

TABLA B.1. Resumen de métricas de desempeño para las tres configuraciones experimentales del modelo LightFM.

Configuración	Precision@10	Recall@10
<i>Test 1</i> – Contexto categórico reducido	25,3 %	27,1 %
<i>Test 2</i> – Atributos discretizados y explicativos	26,1 %	27,8 %
<i>Test 3</i> – Depuración y selección de <i>embeddings</i>	25,8 %	27,5 %

Bibliografía

- [1] James Bennett y Stan Lanning. «The Netflix Prize». En: (2007). Available at: https://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf.
- [2] Jonathan L. Herlocker et al. «An Algorithmic Framework for Performing Collaborative Filtering». En: (2000), págs. 230-237. DOI: [10.1145 / 312624.312682](https://doi.org/10.1145/312624.312682).
- [3] Xinrui Zhang y Hengshan Wang. «Study on Recommender Systems for Business-To-Business Electronic Commerce». En: *Communications of the IIMA* 5.4 (2005), Article 8. DOI: [10.58729/1941-6687.1282](https://doi.org/10.58729/1941-6687.1282). URL: <https://scholarworks.lib.csusb.edu/ciima/vol5/iss4/8>.
- [4] Confidential. [GENERAL RANK] Purchase Preference (EN). Inf. téc. Internal technical report, not publicly available. Confidential organization, 2023.
- [5] Francesco Ricci, Lior Rokach y Bracha Shapira. *Recommender Systems Handbook*. 2nd. Springer, 2015. DOI: [10.1007/978-1-4899-7637-6](https://doi.org/10.1007/978-1-4899-7637-6).
- [6] Gediminas Adomavicius y Alexander Tuzhilin. «Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions». En: *IEEE Transactions on Knowledge and Data Engineering* 17.6 (2005), págs. 734-749. DOI: [10.1109/TKDE.2005.99](https://doi.org/10.1109/TKDE.2005.99).
- [7] Yifan Hu, Yehuda Koren y Chris Volinsky. «Collaborative Filtering for Implicit Feedback Datasets». En: *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM)*. 2008, págs. 263-272. DOI: [10.1109 / ICDM.2008.22](https://doi.org/10.1109/ICDM.2008.22).
- [8] Badrul Sarwar et al. «Item-based Collaborative Filtering Recommendation Algorithms». En: *Proceedings of the 10th International Conference on World Wide Web (WWW)*. 2001, págs. 285-295. DOI: [10.1145/371920.372071](https://doi.org/10.1145/371920.372071).
- [9] Yehuda Koren, Robert Bell y Chris Volinsky. «Matrix Factorization Techniques for Recommender Systems». En: *Computer* 42.8 (2009), págs. 30-37. DOI: [10.1109/MC.2009.263](https://doi.org/10.1109/MC.2009.263).
- [10] Michael J. Pazzani y Daniel Billsus. «Content-based Recommendation Systems». En: *The Adaptive Web*. Vol. 4321. Lecture Notes in Computer Science. Springer, 2007, págs. 325-341. DOI: [10.1007/978-3-540-72079-9_10](https://doi.org/10.1007/978-3-540-72079-9_10).
- [11] Paul Covington, Jay Adams y Emre Sargin. «Deep Neural Networks for YouTube Recommendations». En: *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys)*. 2016, págs. 191-198. DOI: [10.1145 / 2959100.2959190](https://doi.org/10.1145/2959100.2959190).
- [12] Databricks Inc. *Databricks: Unified Data Analytics Platform*. <https://docs.databricks.com>. Official product documentation, accessed: 2025-09-28. 2024.
- [13] Matei Zaharia et al. «Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing». En: *Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. 2012, págs. 2-2.
- [14] Matei Zaharia et al. «Accelerating the Machine Learning Lifecycle with MLflow». En: *IEEE Data Engineering Bulletin* 41.4 (2018), págs. 39-45.

- [15] Yifan Hu, Yehuda Koren y Chris Volinsky. «Collaborative Filtering for Implicit Feedback Datasets». En: *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM)* (2008), págs. 263-272. DOI: [10.1109/ICDM.2008.22](https://doi.org/10.1109/ICDM.2008.22).
- [16] Maciej Kula. «Metadata Embeddings for User and Item Cold-start Recommendations». En: *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys)*. 2015, págs. 279-282. DOI: [10.1145/2792838.2799663](https://doi.org/10.1145/2792838.2799663).
- [17] John D. Hunter. «Matplotlib: A 2D Graphics Environment». En: *Computing in Science & Engineering* 9.3 (2007), págs. 90-95. DOI: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55).
- [18] Michael L. Waskom. «Seaborn: statistical data visualization». En: *Journal of Open Source Software* 6.60 (2021), pág. 3021. DOI: [10.21105/joss.03021](https://doi.org/10.21105/joss.03021).
- [19] GitHub Inc. *GitHub: Software Development Platform*. <https://github.com>. Accessed: 2025-09-28. 2024.
- [20] Richard Koch. *The 80/20 Principle: The Secret to Achieving More with Less*. Doubleday, 1998.
- [21] Chris Anderson. *The Long Tail: Why the Future of Business Is Selling Less of More*. Hyperion, 2006.
- [22] Oscar Celma. *Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Springer Theses. Springer, 2010. DOI: [10.1007/978-3-642-13287-2](https://doi.org/10.1007/978-3-642-13287-2).
- [23] Shuai Zhang et al. «Deep learning based recommender system: A survey and new perspectives». En: *ACM Computing Surveys* 52.1 (2019), págs. 1-38.
- [24] Yehuda Koren. «Collaborative Filtering with Temporal Dynamics». En: *Communications of the ACM* 53.4 (2010), págs. 89-97. DOI: [10.1145/1721654.1721677](https://doi.org/10.1145/1721654.1721677).
- [25] Charu C. Aggarwal. *Recommender Systems: The Textbook*. Springer, 2016. DOI: [10.1007/978-3-319-29659-3](https://doi.org/10.1007/978-3-319-29659-3).
- [26] Charu C. Aggarwal. *Data Mining: The Textbook*. Springer, 2015. DOI: [10.1007/978-3-319-14142-8](https://doi.org/10.1007/978-3-319-14142-8).
- [27] Takuya Akiba et al. «Optuna: A next-generation hyperparameter optimization framework». En: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2019), págs. 2623-2631.
- [28] Xiangnan Yi, Lichan Hong, Ed H. Chi et al. «Sampling-Bias-Corrected Neural Modeling for Large Corpus Item Recommendations». En: (2019), págs. 269-277. DOI: [10.1145/3298689.3346996](https://doi.org/10.1145/3298689.3346996).
- [29] Xiangnan He et al. «Neural Collaborative Filtering». En: (2017), págs. 173-182. DOI: [10.1145/3038912.3052569](https://doi.org/10.1145/3038912.3052569).
- [30] Steffen Rendle et al. «BPR: Bayesian Personalized Ranking from Implicit Feedback». En: (2009), págs. 452-461. URL: <https://arxiv.org/abs/1205.2618>.
- [31] Po-S. Huang et al. «Learning deep structured semantic models for web search using clickthrough data». En: (2013), págs. 2333-2338. DOI: [10.1145/2505515.2505665](https://doi.org/10.1145/2505515.2505665).
- [32] TensorFlow Recommenders Team. *Two-Tower Retrieval Models*. https://www.tensorflow.org/recommenders/examples/two_tower_retrieval. Accessed: 2025-10-31. 2021.
- [33] Ahmed Elgendi et al. «Demand forecasting and price elasticity modeling in retail». En: *Applied Sciences* (2021).
- [34] Ravi Gupta et al. «Predicting sales using machine learning». En: *International Journal of Computer Applications* (2019).

- [35] Leland McInnes, John Healy y Steve Astels. «HDBSCAN: Hierarchical density based clustering». En: *Journal of Open Source Software* (2017).
- [36] Wei Zhang et al. «Customer segmentation via clustering in retail». En: *Procedia Computer Science* (2017).
- [37] Badrul Sarwar et al. «Recommender systems for large-scale customer groups». En: *WWW*. 2002.
- [38] Balázs Hidasi et al. «Session-based recommendations with recurrent neural networks». En: *ICLR* (2016).
- [39] Wang-Cheng Kang y Julian McAuley. «Self-attentive sequential recommendation». En: *ICDM* (2018).
- [40] Tomas Mikolov et al. «Distributed representations of words and phrases and their compositionality». En: *NeurIPS* (2013).
- [41] Mihajlo Grbovic y Haibin Cheng. «Real-time Personalization using Embeddings for Search Ranking at Airbnb». En: (2018), págs. 311-320. DOI: [10.1145/3219819.3219885](https://doi.org/10.1145/3219819.3219885).
- [42] Weinan Tang y Rui Wang. «Ranking distillation: Learning compact ranking models with knowledge distillation». En: *KDD*. 2018.
- [43] Geoffrey Hinton, Oriol Vinyals y Jeff Dean. «Distilling the knowledge in a neural network». En: *arXiv:1503.02531* (2015).
- [44] David Sculley et al. «Hidden technical debt in machine learning systems». En: *NeurIPS* (2015).
- [45] Matei Zaharia et al. «Spark: Cluster computing with working sets». En: *USENIX HotCloud*. 2010.
- [46] Eytan Bakshy, Dean Eckles y Michael Bernstein. «Designing and deploying online field experiments». En: *WWW*. 2014.
- [47] Ron Kohavi et al. «Practical guide to controlled experiments on the web». En: *Data Mining and Knowledge Discovery* (2009).
- [48] Alex Deng y Ya Xu. «Improving the sensitivity of online controlled experiments». En: *KDD* (2013).