Facultad de Ciencias Económicas y Empresariales

ICADE

IMPLEMENTACIÓN DE TÉCNICAS DE INTELIGENCIA ARTIFICIAL Y BIG DATA PARA LA OPTIMIZACIÓN DEL CUSTOMER JOURNEY EN L'ORÉAL: EL USO DE MACHINE LEARNING PARA LA RECOMENDACIÓN DE PRODUCTOS

Autor: Ksenia San Luis Kravtseva

Director: Raúl González Fabre

MADRID | 2024-2025

RESUMEN

La digitalización del sector de la belleza ha impulsado la necesidad de estrategias avanzadas de personalización para optimizar la experiencia del cliente y mejorar la conversión digital. En este contexto, el presente trabajo analiza la aplicación de técnicas de *machine learning* y *Big Data* en el *customer journey* de L'Oréal, con el objetivo de desarrollar modelos predictivos que optimicen la conversión de usuarios y personalicen la recomendación de productos.

Para ello, se diseñaron dos modelos basados en *machine learning*: un modelo de conversión, utilizando *Random Forest* y *XGBoost*, capaz de estimar la probabilidad de compra en función del comportamiento del usuario en la web; y un sistema de recomendación de productos mediante redes neuronales, orientado a personalizar las sugerencias de compra. Debido a restricciones de confidencialidad, se generó una base de datos sintética basada en patrones de comportamiento observados en *Google Analytics* de la compañía.

Los resultados muestran que los modelos de conversión identifican correctamente patrones de compra, aunque presentan limitaciones en la detección de usuarios con mayor propensión a la conversión. Por su parte, el modelo de recomendación alcanzó una precisión del 64.90%, demostrando su capacidad para generar recomendaciones, aunque con tendencia a favorecer ciertos productos. Estos hallazgos subrayan la importancia de mejorar la representatividad de los datos, optimizar los hiperparámetros y explorar técnicas avanzadas de modelado para su aplicación en un entorno real.

Este estudio concluye que la implementación de *machine learning* en la gestión del *customer journey* representa una oportunidad estratégica para L'Oréal, permitiendo mejorar la personalización, incrementar la conversión y optimizar las decisiones de marketing digital. No obstante, para garantizar su eficacia en un entorno empresarial, es fundamental mejorar la calidad de los datos, evaluar el desempeño de los modelos con información real y garantizar su alineación con las estrategias comerciales de la compañía.

Palabras Clave:

Customer journey , *Machine learning*, *Big Data*, *Personalización*, *Sistemas de recomendación*, *Conversión digital*, *Redes neuronales.*

ABSTRACT

The digital transformation of the beauty sector has driven the need for advanced personalization strategies to optimize customer experience and enhance digital conversion. In this context, this study analyzes the application of *machine learning* and *Big Data* techniques in L'Oréal's *customer journey*, with the objective of developing predictive models that optimize user conversion and personalize product recommendations.

To achieve this, two *machine learning*-based models were designed: a conversion prediction model using *Random Forest* and *XGBoost*, capable of estimating the probability of purchase based on user behavior on the website, and a product recommendation system utilizing neural networks, aimed at personalizing purchase suggestions. Due to confidentiality restrictions, a synthetic database was generated based on behavioral patterns observed in the company's *Google Analytics* data.

The results show that the conversion models accurately identify purchasing patterns, although they present limitations in detecting users with a higher propensity to convert. Meanwhile, the recommendation model achieved an accuracy of 64.90%, demonstrating its ability to generate recommendations, albeit with a tendency to favor certain products. These findings highlight the importance of improving data representativeness, optimizing hyperparameters, and exploring advanced modeling techniques for real-world application.

This study concludes that the implementation of *machine learning* in *customer journey* management represents a strategic opportunity for L'Oréal, enabling improved personalization, increased conversion rates, and optimized digital marketing decisions. However, to ensure its effectiveness in a business environment, it is crucial to enhance data quality, evaluate model performance with real-world data, and ensure alignment with the company's marketing strategies.

Keywords:

Customer journey , *Machine learning*, *Bi*g Data*,* Personalization*, Recommendation systems, Digital conversion, Neural networks*.

ÍNDICE

ÍNDICE DE FIGURAS

CAPÍTULO 1: INTRODUCCIÓN

La transformación digital ha cambiado de manera radical la forma en que las empresas interactúan con sus clientes. En un mundo cada vez más interconectado, las plataformas digitales no solo son un canal de venta, sino también el principal medio para construir experiencias personalizadas que fomenten la lealtad y satisfacción del cliente. Dentro de este contexto, la implementación de técnicas de Inteligencia Artificial (IA) y Big Data se ha convertido en una herramienta clave para optimizar el *customer journey*, permitiendo a las empresas comprender mejor las necesidades de sus consumidores y anticiparse a sus preferencias mediante estrategias de personalización basadas en datos.

L'Oréal, como líder mundial en la industria de la belleza, no es ajeno a estas tendencias. La compañía ha adoptado un enfoque innovador al integrar tecnologías avanzadas en sus procesos comerciales y estrategias de marketing. No obstante, actualmente L'Oréal no cuenta con un *customer journey* completamente estructurado en sus plataformas digitales, lo que representa una oportunidad para explorar su desarrollo. En este sentido, me interesa explorar la creación e implementación de un modelo que optimice el recorrido del cliente. Este proyecto no solo tiene una gran relevancia para la compañía, sino que también se alinea con mis intereses personales y académicos, convirtiéndose en una base ideal para mi Trabajo de Fin de Grado

El *customer journey* representa el recorrido completo que realiza un cliente desde su primer contacto con una marca hasta la adquisición del producto o servicio. Este camino está influenciado por una serie de interacciones que pueden ser optimizadas mediante el uso de IA y Big Data. La capacidad de predecir las necesidades de los clientes y ofrecerles soluciones personalizadas no solo mejora su experiencia, sino que también incrementa la probabilidad de conversión y la fidelización.

En un entorno empresarial altamente competitivo, la capacidad de adaptarse a las necesidades y expectativas de los consumidores es crucial. Este trabajo busca no solo proporcionar soluciones prácticas para L'Oréal, sino también contribuir al debate académico sobre el papel de la IA y el Big Data en la transformación digital de las grandes corporaciones. La combinación de tecnologías avanzadas y un enfoque centrado en el cliente se perfila como el camino hacia el éxito en la era digital.

1.1. JUSTIFICACIÓN DEL ESTUDIO

La implementación de técnicas de Inteligencia Artificial y Big Data en la gestión del *customer journey* es un tema de creciente relevancia en el ámbito empresarial. En la actualidad, las empresas buscan constantemente formas de diferenciarse en mercados saturados y altamente competitivos. Este estudio se justifica por la necesidad de explorar cómo estas tecnologías pueden transformar la manera en que las marcas, como L'Oréal, interactúan con sus clientes, ofreciendo experiencias personalizadas que mejoren su lealtad y satisfacción.

En el caso de L'Oréal, el desarrollo del *customer journey* mediante la personalización de recomendaciones de productos a través de *machine learning* tiene un impacto directo en la eficiencia de las estrategias de marketing y en la optimización del uso de recursos. Investigar y desarrollar este proyecto representa una oportunidad única para aplicar estas tecnologías, aportando un enfoque práctico y enriquecedor a mi investigación académica.

1.2. OBJETIVOS DE LA INVESTIGACIÓN

El presente proyecto se enfocará en dos áreas clave dentro del uso de *machine learning* y Big Data en la optimización del *customer journey* de L'Oréal: por un lado, la predicción de conversiones, que permitirá estimar la probabilidad de conversión en función del comportamiento del usuario, y por otro, un sistema de recomendación de productos personalizado. De esta manera, el modelo se dividirá en dos partes complementarias, que contribuirán a mejorar la experiencia del cliente y optimizar las estrategias de marketing digital de la empresa.

El objetivo principal es responder la siguiente pregunta de investigación: ¿Cómo puede L'Oréal utilizar técnicas de *machine learning* y Big Data para optimizar el *customer journey* en sus plataformas digitales, mejorando así la experiencia del cliente y aumentando la personalización de las recomendaciones de productos?

Para responder esta pregunta de investigación, nos proponemos abordar los siguientes objetivos secundarios:

Objetivos teóricos:

- Analizar el papel de la IA y el Big Data en la optimización del *customer journey*.
- Explorar cómo el *machine learning* ha sido aplicado en la personalización de la experiencia del cliente en la literatura académica y en el ámbito empresarial.

Objetivos aplicados al caso de L'Oréal:

- Examinar el estado actual del uso de IA y Big Data en el *customer journey* de L'Oréal.

- Diseñar y desarrollar un modelo predictivo basado en *machine learning* que se divida en dos componentes: (1) un modelo de predicción de conversiones, que estime la probabilidad de compra en función del comportamiento del usuario, y (2) un sistema de recomendación de productos, que personalice la oferta de acuerdo con las interacciones previas del cliente.
- Evaluar el impacto de la personalización en la experiencia del usuario y las decisiones de compra dentro del ecosistema digital de L'Oréal.
- Proponer un marco de mejora continua en la aplicación de IA en estrategias de marketing digital de la empresa.

1.3. METODOLOGÍA DE TRABAJO

El enfoque metodológico de este trabajo combina una revisión bibliográfica con un análisis empírico. Para ello, la metodología se dividirá en tres fases:

- **Revisión teórica:** Se llevará a cabo un análisis de la literatura académica sobre el uso de IA y Big Data en el *customer journey*. Esta revisión permitirá establecer el marco conceptual necesario para comprender cómo estas tecnologías han sido aplicadas en diferentes industrias y qué impacto han tenido en la personalización de la experiencia del cliente.
- **Análisis del caso de L'Oréal:** Se estudiará el estado actual del *customer journey* de L'Oréal a partir de la documentación interna y datos disponibles sobre sus estrategias digitales. Esto permitirá identificar oportunidades de mejora y definir las variables clave que se incorporarán en el modelo predictivo.
- **Desarrollo de modelos:** Se diseñará un modelo piloto basado en técnicas de *machine learning* compuesto por dos partes: (1) un modelo de predicción de conversiones para estimar la probabilidad de compra según el comportamiento del usuario, y (2) un sistema de recomendación de productos que optimice la personalización en las plataformas digitales de L'Oréal. Al tratarse de un modelo piloto, su validación se realizará exclusivamente con datos simulados y anonimizados, sin aplicarse en entornos comerciales reales. Se evaluará su efectividad en términos de optimización del *customer journey* y su viabilidad para futuras implementaciones en entornos comerciales. Finalmente, se realizará un análisis de los resultados obtenidos y se propondrán estrategias de mejora continua para su implementación en el contexto de L'Oréal.

1.4. ESTRUCTURA DEL TRABAJO

Este trabajo se divide en los siguientes capítulos:

1. **Introducción**: Se presenta el contexto del estudio, la justificación, los objetivos y la metodología empleada.
2. **Marco teórico**: Se exploran los conceptos fundamentales relacionados con la IA, Big Data y su aplicación en el *customer journey*, además de los modelos utilizados.
3. **Análisis del *customer journey* en L'Oréal**: Se examina el estado actual de la digitalización del recorrido del cliente.
4. **Desarrollo del modelo de *machine learning***: Se explica la construcción del modelo predictivo y el sistema de recomendación.
5. **Estudio de caso**: Evaluación de los modelos piloto.
6. **Conclusiones y recomendaciones**.

CAPÍTULO 2: MARCO TEÓRICO

2.1. INTELIGENCIA ARTIFICIAL Y BIG DATA

2.1.1. Definición y características

Los sistemas con Inteligencia Artificial (IA) imitan funciones cognitivas humanas, como el aprendizaje, la resolución de problemas y la toma de decisiones, a través de diseños específicamente creados para este propósito (Adam, 2018). Dentro de la IA, existen diferentes subcampos, como el *Machine Learning* (ML) y el *Deep Learning* (DL), un subconjunto del ML, el cual permiten mejorar la capacidad de las máquinas para detectar patrones y tomar decisiones basadas en datos (Adam, 2018).

Por otro lado, el Big Data hace referencia a la recopilación, almacenamiento y análisis de grandes volúmenes de información, caracterizados por las "tres V": Volumen (cantidad masiva de datos), Velocidad (procesamiento en tiempo real) y Variedad (diferentes formatos y estructuras de datos) (Metsai et al., 2021).

Estas tecnologías son interdependientes, ya que el Big Data proporciona los insumos necesarios para que la IA analice, identifique patrones y genere predicciones. Su aplicación en la industria ha transformado la manera en que las empresas procesan información, permitiéndoles obtener *insights* más detallados sobre sus clientes y operaciones (Adam, 2018).

2.1.2. Beneficios en el entorno empresarial

La integración de IA y Big Data ha permitido a las empresas mejorar su eficiencia operativa, personalizar sus ofertas y tomar decisiones estratégicas basadas en datos. Según García et al. (2024), la implementación de técnicas de ML en el análisis de datos de mercado ha permitido mejorar la segmentación y predicción de tendencias.

- **Optimización de Procesos**: El uso de IA en automatización de procesos ha demostrado reducir costos operativos y aumentar la productividad de las organizaciones (Adam, 2018; García et al., 2024).
- **Predicción de Comportamiento del Cliente**: Modelos de ML analizan el historial de compras y comportamientos pasados para predecir tendencias futuras (Ahsain & Kbir, 2022; Panarese et al., 2023).
- **Personalización de Contenidos y Recomendaciones**: Algoritmos de IA optimizan la segmentación de clientes y crean experiencias más personalizadas, aumentando la tasa de conversión y fidelización (Ahsain & Kbir, 2022).

2.1.3. Casos de éxito en grandes corporaciones

Empresas como Amazon y Netflix han demostrado la eficacia de combinar IA y Big Data para personalizar experiencias de usuario. El uso de sistemas de recomendación basados en IA ha permitido a estas compañías aumentar significativamente la retención de clientes y mejorar sus estrategias de fidelización (Panarese et al., 2023).

Otro ejemplo es la industria de la moda y la cosmética, con ejemplos como Estée Lauder y Zara, que han implementado tecnologías de IA y Big Data para personalizar la experiencia del cliente y diseñar experiencias de compra más personalizadas, basadas en el análisis del comportamiento del consumidor (Lorenzo & Romo, 2020; Dominguez, 2025; AECoc, 2021).

2.2. DEFINICIÓN Y ANÁLISIS DEL CUSTOMER JOURNEY

2.2.1. Concepto y relevancia

El *customer journey* se define como el conjunto de interacciones que un cliente tiene con una marca a lo largo de su relación. Estas etapas incluyen el descubrimiento, la consideración, la compra y la fidelización (Metsai et al., 2021; Adam, 2018). Según Metsai et al. (2021), mapear el *customer journey* es fundamental para optimizar la experiencia del usuario y mejorar la conversión.

2.2.2. Herramientas clave para la optimización del customer journey

Para optimizar el *customer journey* y mejorar la experiencia del cliente, las empresas recurren a diversas herramientas tecnológicas que les permiten analizar datos, gestionar interacciones y ofrecer recomendaciones personalizadas. Entre las herramientas más utilizadas se encuentran:

- **Customer Relationship Management (CRM):** Software que permite a las empresas gestionar la relación con los clientes, centralizando información clave sobre interacciones, ventas y preferencias (Wu et al., 2019).
- **Modelos de Analítica Predictiva:** Técnicas estadísticas y algoritmos de ML que permiten predecir el comportamiento del cliente con base en patrones históricos (Sergeevna, 2021).
- **Sistemas de Recomendación:** Algoritmos que analizan datos de clientes para ofrecer productos o servicios personalizados en tiempo real (Panarese et al., 2023).

2.2.3. Etapas del customer journey

- **Conciencia:** En esta fase, el consumidor se familiariza con una marca o un producto a través de diversos canales de comunicación, incluyendo publicidad digital, redes sociales, *influencers* y recomendaciones de terceros. La percepción inicial de la marca juega un papel crucial, y las empresas utilizan estrategias de contenido optimizadas para captar la atención del usuario (Metsai et al., 2021).
- **Consideración:** Durante esta etapa, el consumidor evalúa diversas opciones antes de tomar una decisión de compra. Los motores de búsqueda, las reseñas de productos, los foros de opinión y las experiencias previas de otros consumidores tienen un impacto significativo en este proceso. Las marcas pueden influir en la decisión del consumidor mediante estrategias de remarketing y contenido informativo detallado (Metsai et al., 2021).
- **Compra:** Esta etapa representa la conversión del cliente, donde se concreta la transacción comercial. La facilidad en el proceso de pago, la usabilidad del sitio web o la plataforma de compra y las promociones en tiempo real pueden influir en la decisión final. Según Metsai et al. (2021), la facilidad y seguridad en el proceso de compra son factores determinantes para la conversión del cliente, ya que los consumidores tienden a completar transacciones cuando experimentan una navegación intuitiva y opciones de pago confiables.
- **Fidelización:** Una vez realizada la compra, el reto principal es retener al cliente y convertirlo en un usuario recurrente. Las estrategias de fidelización pueden incluir la personalización de ofertas, programas de recompensas, seguimiento postventa y un servicio de atención al cliente eficiente. Metsai et al. (2021) destacan que la inteligencia artificial desempeña un papel clave en esta fase, ya que permite automatizar la personalización de contenidos y optimizar la retención mediante recomendaciones de productos basadas en análisis predictivo.

2.3. MACHINE LEARNING EN LA PERSONALIZACIÓN DE LA EXPERIENCIA DEL CLIENTE

Dado que la optimización del *customer journey* depende en gran medida del uso de tecnologías avanzadas, es importante analizar cómo el ML contribuye a la personalización y mejora de la experiencia del cliente.

2.3.1. Introducción al Machine Learning

El ML utiliza algoritmos que analizan datos para identificar patrones y hacer predicciones. Esta tecnología se basa en modelos matemáticos que mejoran con el tiempo al procesar más datos, haciéndola ideal para personalizar la experiencia del cliente (Adam, 2018).

- **Aprendizaje Supervisado**: Se emplea para predicciones basadas en datos etiquetados, donde el modelo aprende a partir de ejemplos previos para identificar patrones y hacer predicciones sobre nuevos datos. Este enfoque es ampliamente utilizado en la segmentación de clientes, detección de fraudes y optimización de campañas de marketing. En particular, en la clasificación de clientes, los algoritmos supervisados pueden predecir la probabilidad de conversión con base en comportamientos pasados y atributos específicos de los consumidores (García et al., 2024; Adam, 2018).
- **Aprendizaje No Supervisado**: Este enfoque permite analizar datos no etiquetados y detectar estructuras subyacentes sin intervención humana. Entre sus aplicaciones más comunes se encuentra la segmentación de clientes mediante técnicas de agrupación (*clustering*), que facilita la identificación de grupos con comportamientos similares dentro de un conjunto de datos (Ahsain & Kbir, 2022).
- **Aprendizaje por Refuerzo (*Reinforcement Learning*):** A diferencia de los métodos anteriores, el aprendizaje por refuerzo se basa en un proceso de prueba y error mediante el cual un agente interactúa con su entorno y aprende a tomar decisiones en función de las recompensas o penalizaciones que recibe por cada acción. El objetivo es maximizar una señal de recompensa acumulada a lo largo del tiempo. Este tipo de aprendizaje es especialmente útil en escenarios dinámicos donde las decisiones deben adaptarse a cambios en tiempo real. En el ámbito empresarial, se ha comenzado a aplicar para optimizar estrategias de precios, personalización de experiencias en tiempo real y gestión de campañas publicitarias digitales, especialmente en entornos donde las acciones tienen consecuencias secuenciales y no inmediatas (Monroy, 2023).

2.3.2. Aplicaciones del Machine Learning en la personalización

El uso del ML en la personalización ha revolucionado la manera en que las empresas interactúan con sus clientes. Según Ahsain y Kbir (2022), el ML permite identificar patrones en grandes volúmenes de datos y optimizar la toma de decisiones en la interacción con los usuarios. Gracias a los avances en el análisis de datos y la automatización, las marcas pueden ofrecer experiencias altamente adaptadas a las preferencias y comportamientos individuales de los usuarios (Panarese et al., 2023). A continuación, se presentan algunas de las aplicaciones más relevantes del ML en la optimización del *customer journey*:

- **Sistemas de recomendación:** Los sistemas híbridos de recomendación combinan técnicas de filtrado colaborativo y enfoques basados en contenido para ofrecer productos personalizados según datos históricos y patrones de comportamiento del usuario (Panarese et al., 2023).
- **Análisis predictivo de comportamiento:** Modelos de regresión y redes neuronales que permiten anticipar la intención de compra del cliente (Ahsain & Kbir, 2022).
- **Optimización de campañas de marketing:** Algoritmos que analizan datos de interacción y respuesta a anuncios para diseñar estrategias más efectivas (García et al., 2024).

2.3.3. Modelos de Machine Mearning para el customer journey

Para mejorar la personalización del *customer journey*, es esencial comprender las principales técnicas de ML aplicadas en este ámbito, todas ellas pertenecientes al aprendizaje supervisado. La selección de Random Forest, XGBoost y Redes Neuronales Artificiales responde a su capacidad comprobada para manejar grandes volúmenes de datos, modelar relaciones complejas y ofrecer modelos con alto rendimiento predictivo.

Random Forest y XGBoost son altamente eficaces en la predicción de conversiones, ya que pueden identificar patrones en conjuntos de datos tabulares y generar modelos con gran capacidad de generalización. Mientras que Random Forest mejora la estabilidad de las predicciones mediante la combinación de múltiples árboles de decisión, XGBoost optimiza el rendimiento al emplear *boosting* y técnicas avanzadas de regularización.

Por otro lado, las redes neuronales destacan en el procesamiento de relaciones no lineales y la extracción de representaciones profundas de los datos, lo que las hace ideales para tareas como la recomendación de productos y la personalización de experiencias de usuario. Su capacidad para capturar estructuras complejas en grandes volúmenes de información las

convierte en herramientas clave dentro de estrategias de marketing basadas en inteligencia artificial. A continuación, se explicarán en detalle dichos modelos para comprender su funcionamiento y aplicaciones.

Random Forest

Random Forest es un algoritmo de aprendizaje supervisado basado en árboles de decisión, diseñado para mejorar la precisión y estabilidad de las predicciones mediante la combinación de múltiples modelos. Se basa en la técnica de *bootstrap aggregating* o *bagging*, donde múltiples árboles de decisión son entrenados con subconjuntos aleatorios de los datos y sus predicciones se combinan para obtener un resultado más preciso y robusto (IBM, 2025a; Shafi, 2024; Rohan, 2021).

El funcionamiento de Random Forest se fundamenta en la generación de múltiples árboles de decisión que trabajan de manera independiente. Cada uno de estos árboles se construye a partir de una muestra aleatoria del conjunto de datos original, y en cada nodo del árbol se selecciona un subconjunto aleatorio de características para determinar la mejor división (RandomForestBlog, 2013; Shafi, 2024). Esta aleatorización reduce la correlación entre los árboles individuales y mejora la capacidad del modelo para generalizar a nuevos datos (IBM, 2025a).

Una de las ventajas más destacadas de Random Forest es su capacidad para reducir la varianza de los modelos individuales. Mientras que un solo árbol de decisión puede ser propenso al sobreajuste (*overfitting*), la combinación de múltiples árboles permite mitigar este problema y producir predicciones más estables (Shafi, 2024). Además, este modelo maneja eficientemente los datos desequilibrados, ya que puede ajustar el peso de cada clase y mejorar el rendimiento en problemas de clasificación con distribuciones desiguales de etiquetas (IBM, 2025a).

Desde un punto de vista interpretativo, Random Forest es una técnica ampliamente utilizada debido a su capacidad para calcular la importancia de las variables. Al analizar cuántas veces una característica aparece en los nodos de decisión y cómo influye en la reducción de la impureza del modelo, se pueden extraer *insights* valiosos sobre los factores que más afectan la predicción de un determinado fenómeno (IBM, 2025a). Esta propiedad lo hace ideal para aplicaciones en análisis de datos de mercado, donde la identificación de patrones clave es esencial para la toma de decisiones estratégicas (IBM, 2025a).

Otra fortaleza de Random Forest es su resistencia a los valores atípicos y los datos ruidosos. Dado que cada árbol contribuye con una parte de la predicción global, los efectos de

valores extremos o errores en los datos individuales se diluyen, lo que mejora la robustez del modelo (IBM, 2025a). Además, Random Forest es aplicable tanto a problemas de clasificación como de regresión, lo que lo convierte en una herramienta versátil en distintos ámbitos del aprendizaje automático (Shafi, 2024).

En términos de implementación, Random Forest se ha convertido en un estándar en la ciencia de datos debido a su facilidad de uso y escalabilidad. Puede manejar conjuntos de datos de alta dimensionalidad sin requerir una selección previa de características, lo que lo hace ideal para entornos con grandes volúmenes de información. No obstante, una desventaja de este algoritmo es su costo computacional, ya que la construcción y evaluación de múltiples árboles de decisión requiere más recursos en comparación con modelos individuales. Para abordar este desafío, se pueden optimizar los hiperparámetros del modelo, como la cantidad de árboles y la profundidad máxima de cada uno, con el fin de equilibrar precisión y eficiencia computacional (IBM, 2025a).

XGBoost

XGBoost (*eXtreme Gradient Boosting*) es un algoritmo de aprendizaje automático basado en *boosting*, que mejora su rendimiento al entrenar modelos de manera secuencial, corrigiendo los errores de iteraciones previas (IBM, 2025b). Mientras que Random Forest emplea múltiples árboles de decisión entrenados de manera independiente y luego promedia sus predicciones, XGBoost refina el proceso mediante un enfoque secuencial que ajusta progresivamente el modelo para minimizar errores residuales (IBM, 2025b).

XGBoost se basa en la técnica de *boosting*, un método de ensamblaje en el que múltiples modelos débiles son entrenados de manera secuencial para corregir los errores cometidos en iteraciones previas (IBM, 2025b). En cada paso, el modelo ajusta su peso para enfocarse en los errores más difíciles de predecir, mejorando de forma iterativa la precisión general del sistema (IBM, 2025b). Esto se logra mediante la minimización de una función de pérdida y la actualización eficiente de parámetros mediante optimización por gradiente, lo que hace que el modelo sea altamente preciso y eficiente en términos computacionales (IBM, 2025b).

Una de las características más destacadas de XGBoost es su capacidad de regularización, lo que permite reducir el riesgo de sobreajuste (*overfitting*). Implementa términos de penalización L1 y L2 en su función de costo, lo que evita que el modelo se vuelva demasiado complejo y se adapte excesivamente a los datos de entrenamiento (IBM, 2025b). Además, XGBoost incorpora estrategias avanzadas de pre-podado y submuestreo de características para mejorar la generalización del modelo (IBM, 2025b).

Otra ventaja clave de XGBoost es su escalabilidad. Está diseñado para manejar grandes volúmenes de datos y puede distribuirse en múltiples núcleos de procesamiento, lo que lo hace ideal para aplicaciones de Big Data y análisis en tiempo real (IBM, 2025b). Gracias a su velocidad de ejecución y optimización de memoria, XGBoost se ha convertido en una de las herramientas más utilizadas en competiciones de ciencia de datos. Según Chen y Guestrin (2016), en 2015, 17 de las 29 soluciones ganadoras en Kaggle utilizaron XGBoost, consolidando su popularidad en el ámbito del ML.

XGBoost ha demostrado ser una herramienta altamente eficiente para manejar grandes volúmenes de datos y optimizar el rendimiento en diversas tareas de aprendizaje automático. Gracias a su capacidad para reducir la varianza y mejorar la precisión a través del ensamblaje de múltiples modelos, es ampliamente utilizado en competiciones de ciencia de datos y en aplicaciones que requieren un alto grado de exactitud en la predicción (Chen & Guestrin, 2016).

Redes Neuronales

Las Redes Neuronales Artificiales (ANN) son modelos computacionales inspirados en la estructura y funcionamiento del cerebro humano, diseñados para aprender patrones complejos en grandes volúmenes de datos. Su arquitectura se basa en una red de neuronas interconectadas organizadas en tres capas fundamentales: una capa de entrada, que recibe los datos iniciales; una o varias capas ocultas, donde se procesan las interacciones entre variables a través de funciones de activación no lineales; y una capa de salida, que genera la predicción final en función de la tarea de clasificación o regresión (Bermúdez, Chávez & Ferro, 2013).

El entrenamiento de una red neuronal se lleva a cabo ajustando los pesos sinápticos entre neuronas mediante un proceso iterativo de optimización. Los pesos sinápticos son valores numéricos que determinan la intensidad de la conexión entre neuronas, modulando la influencia de una sobre otra. Su ajuste es clave para que la red neuronal aprenda a representar patrones en los datos de manera eficiente. Este proceso emplea algoritmos como el descenso de gradiente estocástico (SGD), que actualiza los pesos en función del error cometido en cada iteración, y la retropropagación del error, que permite ajustar los pesos en todas las capas de la red para minimizar la función de pérdida (IBM, 2025c). Las RNA han demostrado ser herramientas versátiles en el aprendizaje automático, ya que pueden identificar patrones ocultos en grandes volúmenes de datos y adaptarse a múltiples aplicaciones, como la predicción y la clasificación en entornos complejos. Además, su capacidad para procesar grandes volúmenes de información de manera autónoma las hace fundamentales en sectores como la medicina, la industria y la ciberseguridad (Pastor Rodríguez, 2023).

Existen diversos tipos de redes neuronales, dependiendo del problema a resolver. Entre las más utilizadas se encuentran las Redes Neuronales Profundas (DNN), las Redes Neuronales Convolucionales (CNN), que son particularmente eficientes en el análisis de imágenes, y las Redes Neuronales Recurrentes (RNN), utilizadas en el procesamiento de datos secuenciales como texto y series temporales. Las Redes de Memoria a Largo Plazo (LSTM) son un tipo de red neuronal recurrente diseñada para capturar dependencias de largo plazo en datos secuenciales, lo que las hace especialmente útiles en tareas como el procesamiento de lenguaje natural y la predicción de series temporales (MathWorks, 2025). Por consecuencia, se considera que esta capacidad también podría ser útil para analizar un *customer journey*, ya que permite modelar patrones de comportamiento secuenciales de los usuarios y anticipar sus interacciones futuras.

Overfitting y Underfitting

Cuando se trabaja con modelos de aprendizaje automático, ya sea redes neuronales, Random Forest o XGBoost, un desafío común en la implementación es el sobreajuste (*overfitting*), que ocurre cuando un modelo se ajusta demasiado a los datos de entrenamiento y pierde capacidad de generalización. Para mitigar este problema, se aplican estrategias como la regularización en XGBoost, la poda de árboles en Random Forest y técnicas como Dropout y Early Stopping en Redes Neuronales. En el otro extremo, el subajuste (*underfitting*) se presenta cuando un modelo es demasiado simple para capturar patrones relevantes en los datos. Para solucionarlo, se pueden emplear modelos más complejos, mejorar la selección de características o ajustar hiperparámetros que optimicen el aprendizaje del modelo (Nodd3r, 2022).

CAPÍTULO 3: ANÁLISIS DEL CUSTOMER JOURNEY EN L'ORÉAL

3.1. DESCRIPCIÓN DE LA EMPRESA

L'Oréal es una de las compañías líderes en la industria de la belleza y cosmética a nivel mundial. Fundada en 1909 en París, Francia, la empresa ha expandido su presencia global y actualmente opera en más de 130 países (L'Oréal Paris, 2024c). Desde su origen, ha diversificado su portafolio de productos y adquirido diversas marcas para consolidar su posición en el mercado de la belleza.

3.1.1. Sectores de Actividad

L'Oréal opera en múltiples segmentos del mercado de la belleza (L'Oréal Paris, 2024a):

- **Cuidado del Cabello:** Champús, acondicionadores y tratamientos capilares.
- **Maquillaje:** Bases, labiales, sombras y otros productos para el rostro.
- **Cuidado de la Piel:** Cremas hidratantes, sueros y mascarillas.
- **Fragancias:** Perfumes y colonias bajo distintas marcas.

3.1.2. Operaciones y Distribución

La compañía cuenta con una red global de producción y distribución, con 38 fábricas y 20 centros de investigación dedicados a la innovación en cosmética (Cinco Días, 2024). En España, la fábrica de L'Oréal en Burgos destaca como un referente en sostenibilidad y avances tecnológicos (J.LG., 2024).

3.1.3. Presencia en el Mercado

L'Oréal lidera el mercado global de cosméticos con una cartera de 38 marcas. En 2023, la empresa adquirió la marca de lujo Aesop, fortaleciendo su presencia en el segmento premium (Cinco Días, 2024). Además, la compañía cotiza en Euronext de París desde el 8 de octubre de 1963 y forma parte del índice CAC 40, que agrupa a las 40 principales empresas cotizadas en la Bolsa de París (L'Oréal Finance, 2025; Live Euronext, 2025).

3.1.4. Compromiso con la Sostenibilidad

La sostenibilidad es un pilar estratégico para L'Oréal. La empresa se ha comprometido a alcanzar la neutralidad de carbono en todas sus instalaciones para 2025 y a reducir sus emisiones de carbono por producto en un 25% para 2030 (Cinco Días, 2024). Asimismo, ha desarrollado programas de inclusión y responsabilidad social, como la formación en belleza para colectivos vulnerables (Pérez Galdón, 2024).

3.1.5. Innovación y Tecnología

L'Oréal ha incrementado su inversión en tecnología digital y datos, con el objetivo de definir la belleza del futuro. En Madrid, la compañía inauguró el Campus de Excelencia en D2C e-Commerce, que centraliza el conocimiento tecnológico y las funciones de comercio electrónico directo al consumidor (D2C) en Europa (L'Oréal, 2024). Este campus gestiona más de 100 sitios web D2C de 14 marcas de L'Oréal en Europa, incluyendo Lancôme, Kiehl's, Armani, Yves Saint Laurent y L'Oréal Paris (L'Oréal, 2024).

3.2. DESCRIPCIÓN DEL CUSTOMER JOURNEY EN PLATAFORMAS DIGITALES

3.2.1. Contexto actual de las plataformas digitales de L'Oréal

L'Oréal ha implementado un ecosistema digital diverso que incluye sitios web corporativos, aplicaciones móviles, redes sociales y colaboraciones con plataformas de comercio electrónico. Sin embargo, aún enfrenta retos para mapear de manera integral el recorrido del cliente, particularmente en la integración de datos entre diferentes puntos de contacto. El *customer journey* digital de L'Oréal es actualmente una combinación de interacciones no lineales que dependen de factores como la ubicación geográfica y las preferencias culturales del cliente (Lorenzo & Romo, 2020).

3.2.2. Etapas del Customer Journey en L'Oréal

L'Oréal ha estructurado su *customer journey* digital siguiendo el modelo de cuatro etapas descrito en el marco teórico (Metsai et al., 2021):

- **Conciencia:** Los clientes descubren la marca a través de redes sociales, anuncios digitales y recomendaciones de terceros.
- **Consideración:** Evaluación de opciones en plataformas digitales, con herramientas de prueba virtual como Modiface.

- **Compra:** Transacción en plataformas propias de L'Oréal o *marketplaces* asociados, asegurando una experiencia fluida.
- **Fidelización:** Seguimiento postventa mediante emails personalizados, encuestas de satisfacción y programas de recompensas.

3.3. RECOLECCIÓN DE DATOS Y ANÁLISIS DEL COMPORTAMIENTO DEL CLIENTE

3.3.1. Fuentes de datos

L'Oréal recopila y analiza grandes volúmenes de datos a partir de múltiples fuentes para optimizar la experiencia del cliente y mejorar la personalización de sus servicios. Las principales fuentes de datos utilizadas por la empresa incluyen:

- **Plataformas digitales propias:** Los sitios web de L'Oréal y sus aplicaciones móviles permiten recopilar datos sobre la navegación de los usuarios, preferencias de productos y patrones de interacción en el comercio electrónico. Esta información es utilizada para segmentar audiencias y mejorar las estrategias de marketing digital (Metsai et al., 2021).
- **Redes sociales y *engagement* digital:** L'Oréal implementa herramientas avanzadas para analizar interacciones en redes sociales, evaluar la percepción de la marca y medir la efectividad de sus campañas digitales. Este análisis de *engagement* permite ajustar estrategias en tiempo real para optimizar la comunicación con los consumidores (Lorenzo & Romo, 2020).
- **Historial de compras y comportamiento de clientes:** A través de sus programas de fidelización y plataformas de comercio electrónico, L'Oréal recopila información sobre las compras de los clientes, la frecuencia de compra y las categorías de productos adquiridos. Estos datos se utilizan para diseñar estrategias de recomendación de productos basadas en machine learning (Panarese et al., 2023).
- **Análisis de campañas publicitarias:** Los datos derivados de campañas publicitarias digitales se procesan para evaluar su impacto en la conversión de clientes y mejorar la asignación de presupuestos. El uso de algoritmos predictivos permite mejorar la segmentación de audiencias y maximizar el retorno de inversión en marketing digital (Sergeevna, 2021).
- **Tecnología de reconocimiento facial y análisis de imágenes:** L'Oréal emplea tecnologías avanzadas como Modiface para analizar imágenes faciales y recomendar productos de belleza personalizados. Estas herramientas de realidad aumentada han

mejorado significativamente la experiencia del cliente al permitir la prueba virtual de productos antes de su compra (Wu et al., 2019).

3.4. RETOS ACTUALES EN LA OPTIMIZACIÓN DEL CUSTOMER JOURNEY

3.4.1. Integración de datos

Uno de los principales desafíos para L'Oréal es consolidar la información proveniente de diversas fuentes digitales. La fragmentación de datos puede dificultar la personalización de experiencias y la segmentación precisa de audiencias (Metsai et al., 2021). Para abordar este problema, la compañía ha adoptado plataformas de datos de clientes (CDP) que permiten una visión unificada del consumidor (Panarese et al., 2023).

3.4.2. Privacidad y cumplimiento normativo

L'Oréal debe garantizar el cumplimiento de regulaciones como el GDPR en Europa. Para ello, ha implementado medidas de anonimización y encriptación de datos, asegurando que los clientes tengan control sobre su información personal. La empresa ha desarrollado plataformas donde los usuarios pueden gestionar su privacidad y ejercer sus derechos conforme a la legislación vigente (Wu et al., 2019; Sergeevna, 2021).

3.4.3. Escalabilidad y sostenibilidad

Con el crecimiento de su ecosistema digital, L'Oréal enfrenta el reto de mantener estrategias sostenibles. La empresa ha optimizado sus plataformas tecnológicas para reducir el consumo energético y mejorar la eficiencia en logística, minimizando su impacto ambiental (García et al., 2024). Además, su compromiso con la belleza sostenible ha llevado a iniciativas como el uso de envases reciclables y reducción de emisiones de carbono en sus procesos productivos (J.LG., 2024).

En conclusión, la optimización del *customer journey* en L'Oréal requiere una combinación de innovación tecnológica, cumplimiento normativo y estrategias sostenibles. La implementación de inteligencia artificial y Big Data seguirá siendo clave para mejorar la experiencia del cliente y reforzar su liderazgo en la industria de la belleza digital (L'Oréal Paris, 2024b).

CAPÍTULO 4: TÉCNICAS DE MACHINE LEARNING APLICADAS AL CUSTOMER JOURNEY

4.1. CONSTRUCCIÓN DE LA BASE DE DATOS SINTÉTICA

Debido a restricciones de confidencialidad, L'Oréal no proporcionó acceso directo a una base de datos real de clientes. Como alternativa, se realizó un análisis detallado de los datos de Google Analytics de la compañía y se diseñó una base de datos sintética de 60.000 filas que replica los patrones de comportamiento del usuario observados.

4.1.1. Proceso de Creación de la Base de Datos Sintética

Análisis de Datos Reales

El primer paso consistió en realizar un análisis exploratorio de los datos disponibles en Google Analytics de L'Oréal para comprender el comportamiento de los usuarios y sus interacciones con la plataforma. Este análisis permitió identificar patrones clave en la navegación de los clientes, así como sus principales puntos de abandono en el embudo de conversión. Se examinaron métricas clave como:

- **Tasas de conversión de usuarios**: Se analizaron las tasas de conversión de diferentes segmentos de clientes para determinar qué factores influyen en la decisión de compra.
- **Duración promedio de las sesiones**: Se midió el tiempo que los usuarios permanecen en la plataforma antes de realizar una conversión o abandonar la navegación.
- **Número de productos visualizados por usuario**: Se estudió la cantidad de productos explorados en cada sesión y su relación con la probabilidad de compra.
- **Frecuencia de visitas antes de la compra**: Se identificó cuántas visitas son necesarias en promedio antes de que un usuario tome una decisión de compra.
- **Páginas de salida y puntos de abandono del embudo de conversión**: Se analizaron los momentos críticos en los que los usuarios abandonan la plataforma sin completar una compra, con el objetivo de mejorar la retención en estas etapas.
- **Fuentes de tráfico y patrones de navegación**: Se evaluaron los diferentes canales de adquisición (tráfico directo, redes sociales, campañas pagadas, búsqueda orgánica, etc.) y su impacto en la conversión y retención de clientes.

Este análisis sirvió de base para la creación de la base de datos sintética, procurando que los datos generados se asemejen lo máximo posible al comportamiento real de los usuarios en la plataforma digital de L'Oréal. Aunque se tomaron en cuenta múltiples factores para

reflejar fielmente las tendencias observadas, es importante reconocer que esta base de datos sintética no puede replicar con absoluta precisión la complejidad de los datos reales.

Definición de Variables Clave

A partir del análisis de Google Analytics, se establecieron las siguientes variables clave para la base de datos sintética:

- **Identificador único del usuario** (*user_id*).
- **Identificador de sesión** (*session_id*).
- **Fuente de tráfico** (*source*) (ej. Directo, Búsqueda Orgánica, Búsqueda Paga, Redes Sociales, etc.).
- **Dispositivo utilizado** (*device*) (ej. Móvil, Escritorio, Tableta).
- **Navegador del usuario** (browser).
- **Nuevo usuario** (*new_user*) (Sí/No).
- **Segmento de audiencia** (*audience_segment*) (ej. Visitante Nuevo, Cliente Leal, Cliente Recurrente).
- **Categoría del evento** (*event_category*) (ej. Interacción, Compra, Promoción, Gestión de Cuenta, etc.).
- **Tipo de evento** (*event_type*) (ej. vista de producto, añadir al carrito, compra, login, búsqueda interna).
- **Tiempo de sesión** (*session_engagement_time*).
- **Información del producto** (*product_id, item_name, item_category, item_price*).
- **Interacción con el producto** (*item_interaction_type*) (ej. Ver, Clic, Añadir al carrito, Compartir, Valorar/Reseñar).
- **Estado del carrito de compra** (*cart_status*) (ej. Vacío, Con Productos).
- **Etapa del checkout alcanzada** (*checkout_step*).
- **Ingresos generados** (*revenue*).
- **Conversión final** (*conversion_flag*) (Sí/No).
- **Abandono de carrito** (*abandoned_cart_flag*) (Sí/No).
- **Productos abandonados** (*abandoned_products*).
- **Tiempo antes del abandono** (*time_spent_before_abandonment*).

Para mayor detalle, en los anexos se incluye una tabla descriptiva llamada “Análisis de Variables” que define en mayor profundidad las variables utilizadas y su relevancia dentro del modelo.

Generación de Datos Sintéticos

La generación de la base de datos sintética se realizó utilizando Python, empleando librerías especializadas como:

- **random**: Para asignar valores aleatorios ponderados a variables categóricas.
- **pandas**: Para estructurar los datos en un DataFrame y realizar manipulaciones de datos.
- **numpy**: Para generar distribuciones de valores continuos en variables como *session_engagement_time* y *time_spent_before_abandonment*.

Cada variable se generó utilizando métodos específicos:

- **Selección aleatoria ponderada**: Para *source*, *device*, *browser*, *event_type*, y *audience_segment*, asegurando una distribución realista basada en tendencias de tráfico y comportamiento del usuario.
- **Reglas condicionales**: Se aplicaron en la asignación de valores como *checkout_step* y *conversion_flag*, garantizando coherencia en la simulación. Por ejemplo, solo los eventos de tipo Shopping pueden activar un *checkout_step*, y únicamente aquellos con *checkout_step = Purchase* pueden marcar *conversion_flag = YES*.
- **Distribución uniforme**: Para variables como *session_engagement_time* y *time_spent_before_abandonment*, estableciendo un rango entre 5 y 600 segundos para reflejar comportamientos típicos de navegación.
- **Dependencia entre variables**: Se aseguró que variables como *abandoned_cart_flag* y *checkout_abandonment_step* fueran coherentes con los estados previos del usuario en el proceso de compra.

Adicionalmente, la información sobre los productos provenía de un archivo externo en formato Excel, que contenía datos como identificadores de producto, nombres, categorías y precios. Estos valores fueron integrados en la base de datos sintética, asegurando que la asignación de productos a los eventos de compra reflejara las condiciones establecidas en el diseño del modelo. Véase código empleado en Anexo 2: *Código de Generación de Dataset Sintético*.

Validación de la Base de Datos Sintética

Para garantizar que la base de datos sintética refleje patrones de comportamiento realistas y pueda utilizarse con fiabilidad en el desarrollo de modelos predictivos, se implementaron diversas técnicas de validación que aseguran la coherencia, representatividad y calidad de los datos generados. Estas pruebas permiten evaluar si la base de datos reproduce adecuadamente las características observadas en un entorno de *e-commerce* real.

1. Evaluación de la Distribución de Eventos y Categorías

Se realizó un análisis de distribución de los eventos de interacción y compra para verificar que reflejen tendencias realistas en plataformas digitales. Para ello:

- Se comparó la frecuencia de los distintos *event_type* dentro de cada *event_category*.
- Se aseguró que eventos como *add_to_cart* y *purchase* representaran tasas esperadas dentro del *funnel* de conversión.
- Se analizaron patrones de interacción en función de los segmentos de audiencia (*new_user, returning_customer, loyal_customer*) para verificar su coherencia con comportamientos reales de clientes recurrentes y nuevos visitantes.

2. Comparación con Tendencias del Sector

Para validar la representatividad de los datos respecto a la industria de la belleza y el *retail* digital, se contrastaron valores clave con referencias sectoriales:

- **Tasa de conversión:** Se verificó que la proporción de *conversion_flag* = "YES" estuviera alineada con valores promedios de conversión en *e-commerce* de belleza y moda, observándose una tasa de conversión entre el 20% y el 30% tras los análisis realizados.
- **Duración media de sesión:** Se comparó el tiempo de sesión medio con *benchmarks* de plataformas similares.
- **Interacción con promociones:** Se analizó si los valores de *promotion_interaction* seguían tendencias esperadas según estrategias comunes en *retail* digital. Se observó, por ejemplo, que los usuarios con mayor intención de compra y tiempos de sesión más largos tenían una mayor probabilidad de buscar ofertas y descuentos antes de realizar una conversión, lo que refleja un comportamiento típico de consumidores en plataformas de *e-commerce*.

3. Pruebas de Correlación y Relaciones Esperadas

Se realizaron análisis de correlación entre variables clave para asegurar que los datos reflejen comportamientos reales:

- **Relación entre *session_engagement_time* y *conversion_flag*:** Se comprobó que tiempos de sesión más largos estuvieran correlacionados con una mayor probabilidad de conversión.

- **Impacto de la fuente de tráfico en la conversión:** Se evaluó si ciertas fuentes de tráfico (*Paid Search, Organic Search, Social Media*) presentaban tasas de conversión diferentes.
- **Abandono de carrito y fase del *checkout*:** Se verificó que *abandoned_cart_flag* solo estuviera presente en sesiones donde *checkout_step* no alcanzó la fase de Purchase.
- **Comparación con datos de Google Analytics:** Se analizaron los comportamientos de navegación y conversión en comparación con patrones observados en Google Analytics de la compañía para asegurar que la distribución de los datos sea representativa de usuarios reales y alineada con la dinámica específica del negocio.

4. Revisión de Coherencia de Valores y Anomalías

Para garantizar que los datos fueran consistentes y no presentaran errores lógicos, se implementaron validaciones adicionales:

- **Detección de valores atípicos:** Se analizaron distribuciones para identificar valores extremos en *revenue, session_engagement_time* y *item_price*.
- **Verificación de relaciones condicionales:** Se comprobó que variables condicionales, como *checkout_step* y *conversion_flag*, mantuvieran una relación lógica.
- **Pruebas de consistencia en la segmentación de usuarios:** Se aseguró que los valores de *audience_segment* y *new_user* fueran coherentes en todas las filas.

5. Introducción de Ruido Controlado en el *Dataset*

Para mejorar la capacidad de generalización del modelo y evitar que simplemente "memorice" los patrones sintéticos, se introdujo ruido de manera controlada en el *dataset*. Esta modificación se realizó alterando ciertos valores de forma aleatoria, pero siempre respetando la lógica interna de los datos. Es decir, no se realizaron cambios arbitrarios que pudieran corromper la coherencia del *dataset*, como asignar nombres de productos en la variable *audience_segment*. En su lugar, se llevaron a cabo ajustes dentro de los límites naturales de variabilidad esperados en un entorno real.

El propósito de esta estrategia es evitar que el modelo de aprendizaje automático acepte ciegamente los datos sintéticos sin realmente aprender de ellos. Si el *dataset* fuera demasiado estructurado y "perfecto", el modelo no desarrollaría una capacidad real de predicción, sino que simplemente replicaría los resultados sin entender patrones subyacentes. Al introducir esta variabilidad de manera intencionada, se logra un entrenamiento más robusto y alineado con situaciones reales en entornos de *e-commerce*.

En consecuencia, la correcta estructuración y validación de esta base de datos sintética garantiza su utilidad como insumo clave para el desarrollo de modelos de predicción de conversiones y recomendación de productos en los siguientes capítulos. Gracias a estas pruebas, se confirma que los datos son representativos de un entorno real de *e-commerce* y pueden emplearse para entrenar algoritmos de ML con una base confiable y estadísticamente sólida.

4.2. PREPARACIÓN DEL ENTORNO, PREPROCESAMIENTO Y DIVISIÓN DE DATOS

4.2.1. Configuración del Entorno de Trabajo

El desarrollo del modelo de ML requirió configurar un entorno adecuado para el procesamiento de datos y la implementación de algoritmos de aprendizaje automático. Se empleó Python como lenguaje principal, dada su amplia compatibilidad con librerías especializadas en ML y manipulación de datos.

Las librerías utilizadas fueron:

- **Pandas:** Para la carga, manipulación y limpieza de los datos.
- **NumPy:** Para operaciones matemáticas y manejo de arreglos multidimensionales.
- **Matplotlib:** Para la visualización de datos mediante gráficos.
- **Seaborn:** Para la generación de gráficos estadísticos avanzados.
- **Random:** Para la generación de valores aleatorios y el control de la aleatorización en los experimentos.
- **TensorFlow y Keras:** Para la construcción y entrenamiento de redes neuronales en el sistema de recomendación de productos.
- **XGBoost:** Para el desarrollo del modelo de predicción de conversiones, aprovechando su capacidad de optimización y rendimiento en grandes volúmenes de datos.
- **Bayesian Optimization:** Para la optimización de hiperparámetros del modelo, mejorando su desempeño sin recurrir a una búsqueda exhaustiva.

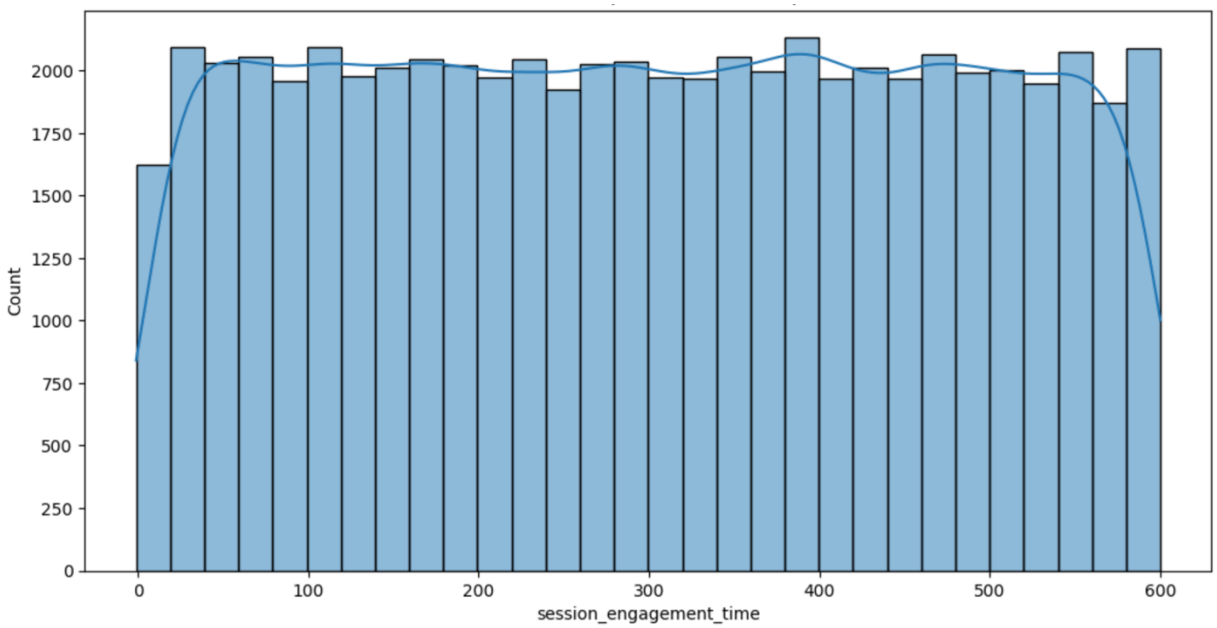
El entorno se configuró en Google Colab (ver Anexo 3: *Código de Preparación del Entorno*), aprovechando su compatibilidad con GPUs y facilitando el procesamiento de grandes volúmenes de datos. Para garantizar la reproducibilidad de los resultados, se fijó la semilla aleatoria en 42 mediante las funciones *np.random.seed(42)*, *random.seed(42)* y *tf.random.set_seed(42)*. Esto aseguró la consistencia en la partición de datos y en el entrenamiento del modelo, evitando variaciones indeseadas en los resultados.

4.2.2. Carga y Exploración de Datos

Los datos sintéticos generados en la etapa anterior se cargaron en un *DataFrame* de Pandas para una exploración inicial. Se realizaron las siguientes acciones (ver Anexo 4: *Código de Exploración y Preprocesamiento de Datos*):

- **Carga del *dataset*:** Se importaron los datos desde archivos Excel generados previamente.
- **Revisión de la estructura de los datos:** Se verificó la integridad de la información, identificando valores nulos y evaluando la distribución de las variables. Se comprobó que identificadores como *user_id* y *session_id* estaban completos, lo que garantiza un seguimiento adecuado de los usuarios. Las variables de comportamiento y navegación se encontraban completas, ofreciendo una base sólida para el análisis del *customer journey*. No obstante, las variables relacionadas con productos y compras presentaban valores nulos, lo que indica que no todas las sesiones incluían interacciones con productos. Asimismo, las variables de conversión mostraron un número limitado de registros, sugiriendo la necesidad de balancear las clases en el modelo predictivo. Con base en estos hallazgos, se definieron estrategias de preprocesamiento específicas para asegurar la calidad y representatividad de los datos.
- **Resumen estadístico:** Se utilizó el método .describe() para analizar la media, desviación estándar, y los valores mínimos y máximos de las variables numéricas.
- **Visualización de distribuciones:** Se generó un histograma (ver figura 1) para evaluar patrones de comportamiento en variables clave. En *datasets* de comportamiento de usuario, el tiempo de sesión puede presentar valores extremadamente altos debido a factores como usuarios que dejan la pestaña abierta sin interacción, errores en la captura de datos o comportamientos atípicos difíciles de modelar. Detectar y, de ser necesario, tratar estos valores es crucial para evitar distorsiones en las estadísticas y en los modelos predictivos. En este caso, la distribución de *session_engagement_time* es uniforme y no presenta valores extremos evidentes, ya que los datos son sintéticos y el tiempo de sesión se ha limitado artificialmente a 600 segundos, evitando la presencia de *outliers* naturales. Por ello, eliminar datos por percentiles no resultó necesario y podría haber afectado la coherencia del *dataset*.

Figura 1 Distribución del tiempo de interacción por sesión



Estos pasos permitieron identificar posibles problemas en los datos y definir estrategias de preprocesamiento adecuadas. Además, se realizó un análisis de la distribución de las variables numéricas para detectar posibles *outliers*; sin embargo, dada la naturaleza sintética de los datos y las restricciones predefinidas, no se identificaron valores atípicos significativos que requirieran eliminación o transformación.

4.2.3. Manejo de Valores Nulos y Datos Atípicos

El tratamiento de los valores nulos fue esencial para evitar sesgos en el modelo. Se implementaron diversas estrategias según el tipo de variable (ver Anexo 4: *Código de Exploración y Preprocesamiento de Datos*):

- Se completaron los valores faltantes en *conversion_flag* con "NO", indicando que la sesión no culminó en una conversión.
- En las variables *revenue* e *item_price*, se asignó el valor 0 en aquellas sesiones en las que no se realizó ninguna compra, garantizando la coherencia en los datos financieros.
- Para *time_spent_before_abandonment*, se imputó un valor de 0 en los casos donde no se registró abandono del carrito, asegurando que esta métrica reflejara únicamente las sesiones con abandono.

Es importante resaltar que no se imputaron todos los valores nulos, ya que algunas variables representaban la ausencia natural de ciertos eventos y no errores en la captura de datos. Por ejemplo, la variable *item_name* puede estar vacía si el usuario está realizando una acción que no involucra un producto, como el registro en la página web. Se mantuvo la

estructura original del dataset para preservar la fidelidad del *customer journey* y evitar sesgos en el análisis.

4.2.4. Codificación de Variables Categóricas

Las variables categóricas se transformaron a formato numérico mediante Label Encoding para garantizar su compatibilidad con los modelos de ML.

Cada variable fue convertida a valores numéricos utilizando LabelEncoder() de la librería *sklearn.preprocessing*, lo que aseguró una representación adecuada sin pérdida de información. Esta codificación facilitó la interpretación de los datos por parte de los algoritmos y optimizó la eficiencia de los modelos predictivos. Véase en el apartado correspondiente de Anexo 4: *Código de Exploración y Preprocesamiento de Datos*.

4.2.5. Normalización y Escalado de Variables

El escalado de las variables numéricas fue fundamental para mejorar la estabilidad y la convergencia de los modelos de aprendizaje automático. Se aplicó el método de MinMax Scaling a las variables *session_engagement_time***,** *time_spent_before_abandonment* y *revenue*, dado que sus valores presentaban rangos bien definidos. Se optó por MinMaxScaler en lugar de StandardScaler porque este último asume que los datos siguen una distribución normal, lo que no se cumple en este conjunto. Al escalar los valores en un rango de 0 a 1, MinMaxScaler permitió conservar la estructura original de los datos sin alterar su distribución, lo cual es crucial para modelos que dependen de relaciones proporcionales entre variables. Esto facilitó la estandarización de las variables y mejoró la precisión de los modelos predictivos. Ver Anexo 4: *Código de Exploración y Preprocesamiento de Datos*.

4.2.6. División del Conjunto de Datos

Finalmente, se dividieron los datos en conjuntos de entrenamiento y prueba para evaluar el rendimiento del modelo. Se adoptó la estrategia 80% para entrenamiento y 20% para prueba, asegurando que la distribución de las clases en la variable objetivo se mantuviera representativa. Este equilibrio permitió un aprendizaje efectivo sin comprometer la capacidad de generalización del modelo.

4.3. DESARROLLO DEL MODELO PREDICTIVO DE CONVERSIONES

4.3.1. Definición del Problema y Selección de la Variable Objetivo

El objetivo de este modelo es predecir la probabilidad de conversión de los usuarios en función de su comportamiento dentro del *customer journey*. Para ello, se ha seleccionado la variable *conversion_flag* como la variable objetivo, la cual indica si una sesión de usuario resultó en una conversión (compra) o no.

Dado que la variable original estaba en formato categórico con valores "YES" y "NO", se realizó una transformación a valores numéricos binarios, donde 1 representa una conversión exitosa y 0 indica que no hubo conversión. Esta transformación permite que los algoritmos de Machine Learning procesen la información de manera eficiente y optimicen la predicción de conversiones.

Adicionalmente, se generaron nuevas variables para mejorar la capacidad predictiva del modelo, incluyendo:

- ***total_sessions*:** Número total de sesiones por usuario.
- ***avg_session_time*:** Tiempo promedio de cada sesión.
- ***conversion_rate*:** Proporción de sesiones en las que un usuario ha convertido.

Estas características fueron añadidas al *dataset* original para mejorar la calidad de los insumos del modelo. Véase Anexo 5: *Código de Definición de Variable Objetivo para Modelo de Coversión para mayor detalle*.

4.3.2. Selección de Modelos de Machine Learning

Se ha optado por utilizar Random Forest y XGBoost para la predicción de conversiones debido a su capacidad de manejar grandes volúmenes de datos y capturar relaciones complejas entre variables.

Random Forest: Su capacidad de manejar datos no balanceados y su interpretabilidad lo hacen una opción adecuada para este problema (García et al., 2024).

XGBoost: Como se mencionó en apartados anteriores, es altamente eficiente en conjuntos de datos con alta dimensionalidad y permite mejorar la predicción de conversiones mediante la reducción del sesgo y la varianza (Mathotaarachchi et al., 2024).

La elección de estos modelos se basa en su rendimiento probado en problemas de predicción de comportamiento de usuario, su capacidad para manejar datos tabulares y la interpretabilidad de sus resultados (Ahsain & Kbir, 2022).

4.3.3. Definición y Ajuste de Hiperparámetros

Los hiperparámetros han sido ajustados cuidadosamente para evitar el sobreajuste, considerando que la base de datos es sintética. Se ha reducido la complejidad de los modelos y se ha incorporado regularización para mejorar la generalización.

Los hiperparámetros son valores configurables antes del entrenamiento del modelo que determinan su comportamiento y rendimiento. Ajustar estos valores de manera óptima permite mejorar la precisión y generalización del modelo.

Para Random Forest, se seleccionaron los siguientes hiperparámetros ajustados:

- ***n_estimators***: 4, reduciendo la cantidad de árboles para evitar sobreajuste.
- ***max_depth***: 2, limitando la profundidad para evitar la sobreespecialización.
- ***min_samples_split***: 6, asegurando mayor estabilidad en las divisiones.
- ***min_samples_leaf***: 4, estableciendo un número mínimo de muestras por hoja.
- ***max_features***: "sqrt", para aumentar la aleatorización en la selección de variables.
- ***bootstrap***: True, permitiendo una mayor aleatorización en el muestreo.
- ***random_state***: 42, garantizando reproducibilidad en los resultados.

En el caso de XGBoost, los hiperparámetros optimizados fueron:

- ***n_estimators***: 60, reduciendo la cantidad de árboles para mejorar la generalización.
- ***learning_rate***: 0.015, tasa de aprendizaje más controlada.
- ***max_depth***: 2, limitando la profundidad para evitar sobreajuste.
- ***subsample***: 0.3, reduciendo el número de datos utilizados en cada iteración para mayor diversidad.
- ***colsample_bytree***: 0.15, disminuyendo la dependencia entre características.
- ***min_child_weight***: 18, aumentando la regularización para reducir divisiones innecesarias.
- ***lambda***: 6.0, mayor regularización L2.
- ***alpha***: 5.0, mayor regularización L1.
- ***gamma***: 12.0, penalización más fuerte para evitar divisiones innecesarias.
- ***random_state***: 42, asegurando reproducibilidad.

La optimización de estos hiperparámetros se realizó mediante un ajuste manual basado en prueba y error, evaluando diferentes combinaciones de parámetros para mejorar la generalización del modelo y evitar el sobreajuste. Se priorizó la simplicidad y la regularización

en la configuración de los hiperparámetros, buscando un equilibrio entre rendimiento y eficiencia computacional. Para validar los ajustes, se empleó validación cruzada, lo que permitió seleccionar los valores óptimos que minimizan el sobreajuste y mejoran la capacidad predictiva en datos no vistos. Véase ANEXO 6: *Código de Definición de Hiperparámetros para Modelo de Conversión*.

4.3.4. Entrenamiento y Evaluación de los Modelos

El entrenamiento de los modelos se realizó utilizando validación cruzada para asegurar que los resultados fueran robustos y no dependieran de un único conjunto de datos de prueba. Se aplicaron diferentes técnicas para evaluar y mejorar el rendimiento de cada modelo (ver Anexo 7: *Código de Entrenamiento del Modelo de Conversión*).

Para Random Forest, se implementó validación cruzada con 10 pliegues (*k-fold cross-validation*), obteniendo un desempeño promedio del 90.09% de precisión. Este resultado indica un alto nivel de exactitud en la clasificación, aunque también resalta la importancia de evaluar si el modelo está generalizando correctamente o si existe riesgo de sobreajuste debido a la naturaleza sintética del *dataset*.

En el caso de XGBoost, se empleó una estrategia de optimización de gradiente con validación cruzada de 5 pliegues, alcanzando una precisión del 83.71%. Aunque esta precisión es sólida, se debe considerar que el *dataset* sintético puede facilitar la identificación de patrones más predecibles que en datos reales.

4.4. DESARROLLO DEL SISTEMA DE RECOMENDACIÓN DE PRODUCTO

4.4.1. Definición del Problema y Selección de la Variable Objetivo

El objetivo de este sistema es desarrollar un modelo de recomendación de productos basado en una red neuronal. Se busca predecir el producto más adecuado para recomendar a un usuario en función de su comportamiento dentro del *customer journey*. Para ello, se ha seleccionado la variable *recommended_product* como la variable objetivo, derivada de diferentes interacciones del usuario con productos y categorías en la plataforma. Véase Anexo 8: *Código de Definición de Variable Objetivo para Modelo de Recomendación*.

La estrategia para definir esta variable sigue un enfoque jerárquico basado en el comportamiento del usuario:

- Si el usuario realizó una compra, se recomienda el producto más caro adquirido, siempre y cuando pertenezca a una categoría de consumo recurrente, como maquillaje, cuidado

facial o productos capilares. Esta decisión se fundamenta en que, en el sector cosmético, una parte significativa del catálogo está compuesta por artículos de reposición periódica, cuyo consumo puede estimarse en función de su frecuencia de uso habitual. No obstante, para evitar que el sistema recomiende productos de baja frecuencia de recompra —por ejemplo, fragancias de alta gama o tratamientos intensivos de larga duración—, se introdujo una lógica de exclusión temporal que impide recomendar artículos recientemente adquiridos.

- Si no compró, pero agregó productos al carrito, se recomienda el último añadido. Esta elección refleja la intención de compra más reciente y, por tanto, es un buen indicador de preferencia inmediata.
- Si no hay productos en el carrito, pero hubo interacciones, se recomienda el producto con más interacciones. La interacción reiterada se interpreta como una señal de interés, incluso en ausencia de una intención de compra explícita.
- Si no hay interacción con productos, se recomienda la categoría más visitada. Se asume que la navegación repetida en una categoría indica afinidad, aunque no se haya manifestado aún mediante acciones directas.
- Si no hay suficiente información, se sugiere un producto genérico basado en tendencias de compra del usuario (recordemos que un *user_id* puede aparecer más de una vez en el *dataset*).

Esta metodología permite que el sistema de recomendación adapte sus sugerencias según el nivel de interacción del usuario, optimizando la personalización del *customer journey*. A la vez, introduce mecanismos que previenen recomendaciones poco útiles, como sugerir la recompra inmediata de un producto de consumo ocasional o con un ciclo de reposición largo.

4.4.2. Selección del Modelo de Machine Learning

Se ha elegido una Red Neuronal Artificial (RNA) para la tarea de recomendación de productos, debido a su capacidad para capturar patrones complejos y generalizar mejor en escenarios con múltiples clases (Bermúdez, Chávez & Ferro 2013). A diferencia de enfoques basados en árboles de decisión, las redes neuronales permiten manejar relaciones no lineales y aprender representaciones más profundas de los datos de usuario.

La arquitectura de la red neuronal incluye:

- **Capas densas (*Fully Connected Layers*):** Capturan la estructura de las interacciones entre usuarios y productos.

- **Función de activación ReLU:** Mejora la propagación de la información y acelera la convergencia del modelo.
- ***Dropout*:** Se incorpora para reducir el sobreajuste.
- **Capa de salida con activación *softmax*:** Permite modelar la recomendación como un problema de clasificación multiclase.

4.4.3. Definición y Ajuste de Hiperparámetros

Para optimizar el rendimiento de la red neuronal, se ajustaron los siguientes hiperparámetros mediante *Bayesian Optimization*, una técnica de optimización basada en modelos probabilísticos que busca encontrar el mejor conjunto de hiperparámetros evaluando de manera eficiente el espacio de búsqueda en lugar de probar todas las combinaciones posibles (Shahriari et al., 2016). Se identificaron los mejores valores de *batch size* (~58) y *learning rate* (0.0095), lo que permitió estabilizar la precisión del modelo en torno al 64.39% después de pocas épocas. Sin embargo, la función de pérdida, aunque se redujo inicialmente, se mantuvo elevada. Véase Anexo 9: *Código de Optimización y Entrenamiento de Red Neuronal*.

4.4.4. Arquitectura de la Red Neuronal

La red neuronal utilizada para la recomendación de productos cuenta con una capa de entrada cuya dimensión es igual a la cantidad de características extraídas del usuario. A continuación, dispone de dos capas ocultas con 128 y 64 neuronas**,** respectivamente, cada una con activación ReLU y un mecanismo de *Dropout* del 40% para mitigar el sobreajuste. Finalmente, la capa de salida tiene un número de neuronas equivalente a la cantidad de productos distintos y utiliza una activación *Softmax*. Para el entrenamiento, se empleó la función de pérdida *categorical crossentropy*, la cual mide la diferencia entre las predicciones del modelo y las etiquetas reales, penalizando aquellas incorrectas. El algoritmo de optimización seleccionado fue *Adam*, debido a su capacidad para ajustar adaptativamente la tasa de aprendizaje y mejorar la estabilidad en redes neuronales profundas.

4.4.5. Entrenamiento y Evaluación del Modelo

El conjunto de datos se dividió en 80% para entrenamiento y 20% para prueba, con el objetivo de garantizar una evaluación representativa de la capacidad de generalización del modelo. Tras el entrenamiento, la red neuronal alcanzó una precisión del 64.90% en el conjunto

de prueba, mientras que la validación cruzada reflejó una precisión estable de 64.39%, lo que indica que el modelo ha logrado cierta estabilidad en su desempeño, aunque todavía presenta margen de mejora.

A pesar de que la precisión de entrenamiento y validación son similares, la función de pérdida en validación sigue siendo elevada, lo que sugiere que el modelo no está generalizando de manera óptima. Para mejorar su desempeño, se podrían realizar ajustes en la regularización mediante la modificación del *Dropout* o la incorporación de técnicas como *Batch Normalization*. Asimismo, explorar arquitecturas más complejas con capas más profundas o modificaciones en la activación de la última capa podría contribuir a una mejor predicción. Dado que los datos utilizados son sintéticos, incrementar la variabilidad en la simulación podría enriquecer los patrones aprendidos y mejorar la capacidad de generalización del modelo.

A pesar de estos desafíos, la red neuronal ha demostrado ser capaz de captar patrones relevantes en la recomendación de productos, lo que sugiere que con optimizaciones adicionales podría mejorar su rendimiento y aplicabilidad en escenarios reales. Véase Anexo 9: *Código de Optimización y Entrenamiento de Red Neuronal*.

CAPÍTULO 5: EVALUACIÓN Y COMPARACIÓN DE LOS MODELOS

5.1. MODELOS DE CONVERSIÓN (RANDOM FOREST Y XGBOOST)

Ambos modelos han generado predicciones mayoritariamente en 0 (no conversión), lo que indica que están aprendiendo un patrón en los datos donde la conversión es un evento poco frecuente. Esto refuerza el hallazgo previo en la exploración del *dataset*, donde se observó que la mayoría de las sesiones no generan ingresos ni llegan al *checkout*, lo que sugiere un *funnel* de conversión largo con muchas salidas. Sin embargo, la diferencia en la predicción de la primera clase entre Random Forest y XGBoost sugiere que podrían estar manejando de manera distinta la distribución de los datos.

El modelo de Random Forest presenta una precisión perfecta (1.0000), pero su *recall* es bajo (0.4338), lo que indica que no está detectando todos los casos positivos de conversión. Su AUC-ROC de 0.7169 sugiere un desempeño moderado en la clasificación. En contraste, XGBoost muestra un desempeño deficiente en la detección de conversiones, con un *recall* extremadamente bajo (0.0362) y un AUC-ROC de 0.5181, lo que sugiere que el modelo no está diferenciando bien entre las clases y su rendimiento es similar a una clasificación aleatoria.

A continuación, vemos como las matrices de confusión muestran diferencias significativas en su desempeño:

Figura 15. Matriz de confusión - Random Forest

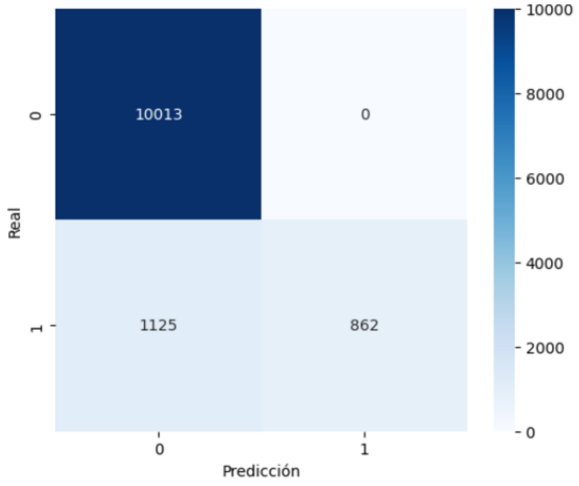
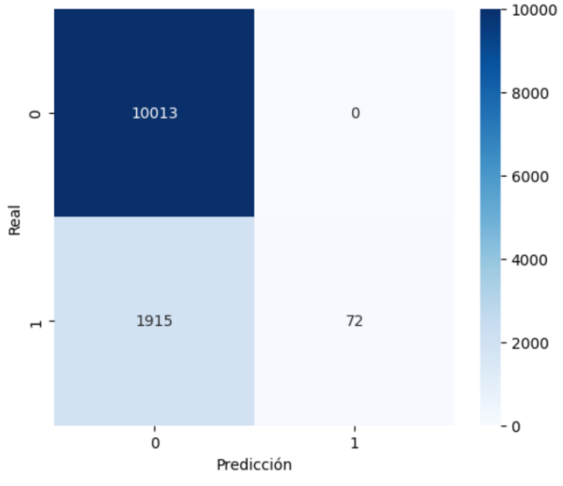


Figura 12. Matriz de confusión - XGBoost



Random Forest presenta una clasificación de la clase 0 con una precisión perfecta (10,013 verdaderos negativos y 0 falsos positivos). Sin embargo, tiene 1,125 falsos negativos en la clase 1, lo que indica que no está detectando correctamente una gran cantidad de casos positivos. La sensibilidad (*recall*) en la clase 1 es baja, lo que sugiere que el modelo favorece la clasificación de la clase mayoritaria. XGBoost, en contraste, tiene un rendimiento inferior

en la clase 1 con 1,915 falsos negativos y solo 72 verdaderos positivos. Esto significa que el modelo apenas logra identificar correctamente los casos positivos, lo que indica un fuerte sesgo hacia la clase mayoritaria (0). Ambos modelos muestran problemas en la detección de la clase minoritaria, con XGBoost presentando el peor desempeño en términos de *recall* para la clase 1.

5.2. RED NEURONAL PARA RECOMENDACIÓN DE PRODUCTOS

El modelo de Red Neuronal ha mostrado un desempeño aceptable en términos de precisión global, alcanzando un 64.90%. Ha logrado identificar correctamente 7,788 de los 12,000 ejemplos evaluados, lo que indica una capacidad significativa para reconocer ciertos patrones en los datos. No obstante, se ha observado una tendencia del modelo a asignar predominantemente la clase 3583, lo que sugiere que aún no está logrando una distribución completamente equilibrada en sus predicciones. A pesar de esta limitación, el modelo es capaz de capturar relaciones en los datos y ofrecer recomendaciones con cierto nivel de precisión en determinados segmentos.

Este comportamiento puede estar influenciado por distintos factores, entre ellos, un desbalance en la distribución de clases dentro del conjunto de entrenamiento o la necesidad de ajustes adicionales en la arquitectura y configuración del modelo. Si bien algunas clases presentan una representación más baja en las predicciones, el modelo logra identificar correctamente categorías con mayor frecuencia en los datos, lo que indica que está aprendiendo ciertos patrones de comportamiento de los usuarios.

En términos de clasificación general, la red neuronal ha demostrado una capacidad de reconocimiento estable en comparación con los modelos de Random Forest y XGBoost, aunque con una menor diversidad en sus predicciones. La distribución de las clases predichas aún muestra cierto grado de dispersión, con algunas categorías bien identificadas y otras con menor representación. Esta situación refleja tanto la estructura de los datos como la manera en que el modelo procesa la información para generar recomendaciones.

Véase Anexo 10: *Código de Evaluación de Modelos* para el capítulo entero.

CAPÍTULO 6: CONCLUSIONES Y RECOMENDACIONES

6.1. RESPUESTA A LA PREGUNTA DE INVESTIGACIÓN

El presente trabajo tuvo como objetivo responder a la siguiente pregunta de investigación: ¿Cómo puede L'Oréal utilizar técnicas de ML y Big Data para optimizar el *customer journey* en sus plataformas digitales, mejorando así la experiencia del cliente y aumentando la personalización de las recomendaciones de productos?

Para abordar esta cuestión, se desarrollaron dos modelos basados en técnicas de ML: (1) un modelo predictivo de conversión que permite estimar la probabilidad de que un usuario realice una compra en función de su comportamiento dentro del *customer journey* lo que a su vez permite a la empresa anticipar en qué fase del recorrido se encuentran los usuarios y activar acciones personalizadas (como promociones, recordatorios o recomendaciones específicas) para guiarlos hacia la conversión de forma más eficaz y (2) un sistema de recomendación de productos basado en redes neuronales, diseñado para personalizar la oferta de productos de acuerdo con las interacciones previas del usuario.

Los resultados obtenidos evidencian que el uso de ML puede aportar mejoras significativas en la optimización del *customer journey* de L'Oréal. La modelización del comportamiento de los usuarios mediante Random Forest y XGBoost permitió identificar patrones de conversión, mientras que la red neuronal empleada en el sistema de recomendación demostró ser capaz de generar sugerencias personalizadas. Sin embargo, los resultados también señalaron desafíos importantes, como el sesgo hacia ciertas categorías de productos en la recomendación y la dificultad de los modelos de conversión para detectar correctamente la clase minoritaria (usuarios que convierten). Estos aspectos destacan la necesidad de seguir optimizando los modelos y de explorar nuevas estrategias para mejorar su capacidad de generalización en entornos reales.

En términos generales, los hallazgos de este estudio confirman que la implementación de técnicas de ML y Big Data en la gestión del *customer journey* representa una oportunidad clave para L'Oréal. No obstante, la eficacia de estos modelos dependerá de su ajuste continuo, la calidad de los datos utilizados y la integración con estrategias de negocio orientadas a la personalización y fidelización de clientes.

6.2. RECOMENDACIONES PARA FUTURAS IMPLEMENTACIONES

A partir de los resultados obtenidos y del análisis de los modelos implementados, se proponen las siguientes recomendaciones para futuras aplicaciones de ML en la optimización del *customer journey* en L'Oréal:

1. **Mejora del Balance de Clases en el Modelo de Conversión:**
   - Implementar técnicas avanzadas de reequilibrio de datos, como *Synthetic Minority Over-sampling Technique* (SMOTE), una técnica que genera nuevas muestras sintéticas de la clase minoritaria basándose en la interpolación de los vecinos más cercanos en el espacio de características (Maklin, 2019) o estrategias de ponderación de clases, para mejorar la capacidad del modelo de predecir conversiones en usuarios con menor representación en el *dataset*.
   - Explorar enfoques basados en *ensembles* adaptativos, un enfoque de aprendizaje automático que combina múltiples modelos para mejorar la precisión y robustez de las predicciones, como Balanced Random Forest o XGBoost con *sampling* ponderado, para mitigar el sesgo hacia la clase mayoritaria.
2. **Optimización de la Red Neuronal para Recomendación de Productos:**
   - Ajustar la arquitectura de la red neuronal, aumentando el número de capas o incorporando mecanismos como *Batch Normalization* y Regularización L2, para mejorar la capacidad de generalización del modelo.
   - Evaluar modelos alternativos, como redes neuronales recurrentes (RNNs), que podrían captar mejor la secuencia de interacciones del usuario y proporcionar recomendaciones más precisas.
   - Refinar la estrategia de segmentación de usuarios para que la recomendación no solo considere eventos recientes, sino también patrones históricos de compra y comportamiento.
3. **Enriquecimiento de la Base de Datos Sintética:**
   - Incorporar nuevas fuentes de datos, como análisis de redes sociales, datos de satisfacción del cliente o información contextual sobre campañas de marketing, para mejorar la representatividad del *dataset*.
   - Aumentar la diversidad de la base de datos sintética mediante técnicas de generación de datos sintéticos avanzadas, como modelos generativos adversariales (GANs), una arquitectura de *deep learning* que permite generar nuevos datos sintéticos mediante la competencia entre dos redes neuronales: un

generador y un discriminador, que permitirían una simulación más realista del comportamiento de los usuarios.

- Aplicar metodologías de evaluación más rigurosas para validar la calidad de los datos generados y su representatividad en relación con los datos reales de L'Oréal.

4. **Validación en Datos Reales y Pruebas en Entornos de Producción:**
    - Realizar pruebas controladas en un entorno real de L'Oréal, integrando los modelos en una plataforma de prueba antes de su despliegue comercial definitivo.
    - Implementar un sistema de aprendizaje continuo para que los modelos se actualicen dinámicamente con nuevos datos de usuario, mejorando su precisión y capacidad adaptativa a lo largo del tiempo.
    - Monitorear el desempeño de los modelos en producción mediante métricas clave, como tasa de conversión real, precisión en la recomendación de productos y *feedback* de los clientes, para realizar ajustes iterativos según sea necesario.

6.3. LIMITACIONES DEL ESTUDIO Y POSIBLES MEJORAS

Si bien este estudio ha demostrado la viabilidad del uso de ML para mejorar el *customer journey* en L'Oréal, es importante reconocer algunas limitaciones que podrían abordarse en futuras investigaciones:

1. **Uso de Datos Sintéticos:**
    - Aunque la base de datos sintética fue diseñada para reflejar patrones reales de interacción de los usuarios en plataformas digitales, la falta de datos reales puede limitar la capacidad de los modelos para generalizar correctamente en entornos de producción.
    - La evaluación en datos sintéticos introduce un margen de error en las predicciones, por lo que futuras investigaciones deberían enfocarse en la validación de los modelos con datos de clientes reales, cumpliendo con normativas de privacidad y anonimización.
2. **Evaluación de Modelos en un Contexto Real:**
    - El presente estudio se centró en la optimización algorítmica, pero no se llevaron a cabo experimentos en un entorno real de L'Oréal. Sería más apropiado implementar estas técnicas en futuras aplicaciones para medir el impacto directo de los modelos en la experiencia de los clientes y en los resultados de negocio.

- La integración de los modelos en sistemas de recomendación en tiempo real representa un desafío adicional que requeriría infraestructura tecnológica avanzada y estrategias de despliegue en producción.

3. **Impacto de la Explicabilidad y Transparencia de los Modelos:**
   - A pesar del buen desempeño de los modelos utilizados, su interpretabilidad sigue siendo un reto. Modelos como XGBoost y redes neuronales pueden ser difíciles de explicar en términos de lógica de decisión, lo que podría afectar su adopción en entornos empresariales.
   - Explorar enfoques de Machine Learning explicable (XAI), un conjunto de técnicas diseñadas para aumentar la transparencia y comprensión de los modelos de inteligencia artificial permitiría ofrecer mayor transparencia en los resultados y facilitar la confianza en la toma de decisiones basada en IA.
4. **Consideraciones Éticas y Regulatorias:**
   - La implementación de sistemas de ML en la personalización del *customer journe*y debe considerar aspectos éticos, como la privacidad del usuario y la equidad en la recomendación de productos.
   - La recopilación y el uso de datos de clientes deben cumplir estrictamente con normativas como el Reglamento General de Protección de Datos (GDPR), asegurando el consentimiento informado y la protección de la información personal de los usuarios.

6.4. CONCLUSIÓN

El presente estudio ha demostrado que la aplicación de técnicas de ML y Big Data tiene un alto potencial para optimizar el *customer journey* en plataformas digitales de L'Oréal. Si bien los modelos desarrollados han mostrado resultados prometedores en términos de predicción de conversiones y personalización de recomendaciones, también se han identificado desafíos importantes que requieren mejoras adicionales para garantizar su aplicabilidad en entornos reales.

La clave del éxito en la implementación de estos modelos radica en la mejora continua de los datos de entrenamiento, la optimización de los algoritmos y la integración de soluciones IA dentro de estrategias de marketing y personalización centradas en el usuario. En este sentido, este estudio podría representar un paso en la exploración del uso de ML en L'Oréal, abriendo la puerta a futuras investigaciones y aplicaciones en la industria de la belleza digital.

BIBLIOGRAFÍA

Adam, M. B. (2018). Improving complex sale cycles and performance by using machine learning and predictive analytics to understand the customer journey. Doctoral dissertation, Massachusetts Institute of Technology.

AECoc. (25 de agosto de 2021). Zara apuesta por la inteligencia artificial, el big data, la analítica avanzada y las inversiones. https://www.aecoc.es/innovation-hub-noticias/zara-apuesta-por-la-inteligencia-artificial-el-big-data-la-analitica-avanzada-y-las-inversiones/

Ahsain, S., & Kbir, M. A. (2022). Predicting the client's purchasing intention using Machine Learning models. In E3S Web of Conferences (Vol. 351, p. 01070). EDP Sciences.

Bermúdez, G. M. T., Chávez, J. S., & Escobar, R. F. (2013). Modelación de sistemas de recomendación aplicando redes neuronales artificiales. Visión electrónica, 7(2), 45-56.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. https://arxiv.org/abs/1603.02754

Cinco Días. (4 de noviembre de 2024). L'Oréal: Belleza, innovación y compromiso sostenible desde París al mundo. https://cincodias.elpais.com/companias/2024-11-04/loreal-belleza-innovacion-y-compromiso-sostenible-desde-paris-al-mundo.html

Dominguez, L. (7 de marzo de 2025). How Estée Lauder's AI agent ConsumerIQ is consolidating data and streamlining R&D. Consumer Goods. https://consumergoods.com/how-estee-lauders-ai-agent-consumeriq-consolidating-data-and-streamlining-rd

García, A. J. C., Basurto, N. A. C., Cevillano, A. N. Z., & Ponce, V. L. M. (2024). TÉCNICAS DE MACHINE LEARNING APLICADAS A LA INTERPRETACIÓN DE DATOS DE MERCADO. Ciencia y Desarrollo, 27(2), 217-226.

IBM. (2025a). Random Forest: An Introduction. https://www.ibm.com/mx-es/think/topics/random-forest

IBM. (2025b). XGBoost: qué es y cómo funciona. https://www.ibm.com/es-es/think/topics/xgboost

IBM. (2025c). ¿Qué es el descenso del gradiente?. https://www.ibm.com/mx-es/think/topics/gradient-descent

J.LG. (4 de octubre de 2024). Cuando la tecnología es una aliada para hacer más bello el futuro. El País. https://elpais.com/sociedad/2024-10-04/cuando-la-tecnologia-es-una-aliada-para-hacer-mas-bello-el-futuro.html

Live Euronext. (2025). L'Oréal Stock Information. https://live.euronext.com/en/product/equities/FR0000120321-XPAR

Lorenzo, E. Y., & Romo, A. R. (2020). La evolución de los insights desde la escucha social a la lectura por imagen: El caso L'Oreal. aDResearch ESIC International Journal of Communication Research, 23(23), 8-29.

L'Oréal. (2024). Campus de Excelencia en D2C e-Commerce de L'Oréal. https://www.loreal.com/es-es/espana/blog/tecnologia/campus-excelencia-d2c-ecommerce-loreal/

L'Oréal Finance. (2025). Share price & stock information. https://www.loreal-finance.com/eng/share-price

L'Oréal Paris. (2024a). Acerca de nosotros. https://www.loreal-paris.es/acerca-de-nosotros

L'Oréal Paris. (2024b). Mejorando todos nuestros productos. https://www.loreal-paris.es/mejorando-todos-nuestros-productos

L'Oréal Paris. (2024c). Nuestra historia. https://www.loreal-paris.es/nuestra-historia

Maklin, C. (14 de mayo de 2022). Synthetic Minority Over-sampling Technique (SMOTE). Medium. https://medium.com/@corymaklin/synthetic-minority-over-sampling-technique-smote-7d419696b88c

Mathotaarachchi, K. V., Hasan, R., & Mahmood, S. (2024). Advanced Machine Learning Techniques for Predictive Modeling of Property Prices. Information, 15(6), 295. https://doi.org/10.3390/info15060295

MathWorks. (s.f.). Introducción a la memoria a corto-largo plazo (LSTM). https://la.mathworks.com/discovery/lstm.html

Metsai, A. I., Tabakis, I. M., Karamitsios, K., Kotrotsios, K., Chatzimisios, P., Stalidis, G., & Goulianas, K. (2021). Customer journey: applications of AI and machine learning in E-commerce. In Interactive Mobile Communication, Technologies and Learning (pp. 123-132). Cham: Springer International Publishing.

Monroy, S. (23 de marzo de 2023). ¿Qué es el Reinforcement Learning? APD. https://www.apd.es/que-es-reinforcement-learning/

Nodd3r. (6 de septiembre de 2022). Underfitting vs Overfitting: diferencias clave y soluciones. https://nodd3r.com/blog/underfitting-vs-overfitting

Panarese, A., Settanni, G., & Galiano, A. (2023). Modeling an e-Commerce Hybrid Recommender System Based on Machine Learning Algorithms. In ICAART (3) (pp. 706-712).

Pastor Rodríguez, J. (2023). Machine Learning y Redes Neuronales Artificiales (RNA). Microbacterium. https://microbacterium.es/machine-learning-y-redes-neuronales-artificiales-rna

Pérez Galdón, B. (29 de noviembre de 2024). La formación en belleza, un camino para la inclusión en el mercado laboral. El País. https://elpais.com/economia/especial-rsc/2024-11-29/la-formacion-en-belleza-un-camino-para-la-inclusion-en-el-mercado-laboral.html

RandomForestBlog. (8 de mayo de 2013). Random Forests: Definición y funcionamiento. https://randomforest2013.blogspot.com/2013/05/randomforest-definicion-random-forests.html

Rohan, P. (2021). Implementation of Bootstrap in Random Forest. Kaggle. https://www.kaggle.com/code/paulrohan2020/implementation-of-bootstrap-in-random-forest

Shafi, A. (2024). Random forests classifier in Python. DataCamp. https://www.datacamp.com/es/tutorial/random-forests-classifier-python

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & De Freitas, N. (2016). Taking the human out of the loop: A review of Bayesian optimization. Proceedings of the IEEE, 104(1), 148-175. https://doi.org/10.1109/JPROC.2015.2494218

Wu, Q., Hsu, W. L., Xu, T., Liu, Z., Ma, G., Jacobson, G., & Zhao, S. (2019). Speaking with actions-learning customer journey behavior. En *2019 IEEE 13th International Conference on Semantic Computing (ICSC)* (pp. 279-286). IEEE.

ANEXOS

ANEXO 1: ANÁLISIS DE VARIABLES

1. Datos de Identificación del Usuario y la Sesión		
Variable	**Descripción**	**Relevancia**
user_id	Identificador único del usuario.	Permite analizar el comportamiento de cada usuario a lo largo del tiempo y construir perfiles de compra.
session_id	Identificador único de la sesión.	Fundamental para diferenciar interacciones dentro de un mismo usuario y analizar el flujo de navegación.
Justificación: Esencial para realizar un análisis basado en el historial de cada usuario y sus sesiones, permitiendo identificar patrones de comportamiento.		
2. Datos sobre la Fuente y el Dispositivo del Usuario		
source	Fuente de tráfico (ej., *Organic*, *Social*, *Paid Social*, etc.).	Permite analizar qué canales son más efectivos para atraer y convertir usuarios.
device	Tipo de dispositivo utilizado (Mobile, Tablet, Desktop).	Ayuda a segmentar a los usuarios según su experiencia en distintos dispositivos.
browser	Navegador utilizado.	Puede afectar la experiencia del usuario y ser relevante para la optimización técnica de la web.
Justificación: Variables clave para entender cómo llegan los usuarios y si existen diferencias en conversión según dispositivo o fuente.		
3. Datos sobre la Audiencia y el Estado del Usuario		
new_user	Indica si el usuario es nuevo o recurrente.	Importante para diferenciar estrategias de adquisición vs. retención.
audience_segment	Segmento al que pertenece el usuario (*New Visitor*, *Returning Customer*, etc.).	Permite analizar el comportamiento de distintos tipos de usuarios y personalizar estrategias de marketing.
Justificación: Son variables fundamentales para personalizar la experiencia del usuario y analizar la fidelización.		
4. Datos de Eventos y Comportamiento en la Web		
event_category	Categoría del evento (ej. *Shopping, Engagement, Technical Events*).	Ayuda a clasificar el comportamiento del usuario en la web.
event_type	Tipo específico de evento (ej. *view_search_results, internal_search, exception*).	Permite entender qué acciones realiza el usuario en la sesión.
session_engagement_time	Tiempo total de interacción en la sesión.	Un proxy útil para medir el interés del usuario en la web.

Justificación: Estas variables permiten mapear el *customer journey* y analizar qué interacciones están relacionadas con la conversión o el abandono.		
5. Datos del Proceso de Compra		
cart_status	Indica si hay productos en el carrito.	Clave para analizar si el usuario está en una etapa avanzada del *funnel* de conversión.
checkout_step	Fase en la que se encuentra el usuario en el proceso de *checkout*.	Permite analizar cuántos usuarios avanzan en el embudo de compra.
revenue	Ingreso generado en la sesión.	Indicador directo de conversión y del valor de cada usuario.
conversion_flag	Indica si el usuario realizó una conversión (compra).	Esencial para modelar predicciones de conversión.
Justificación: Estas variables son críticas para analizar qué factores afectan la conversión y la tasa de abandono.		
6. Datos sobre Abandono y *Retargeting*		
abandoned_cart_flag	Indica si el usuario abandonó el carrito.	Clave para estrategias de recuperación de carritos abandonados.
abandoned_products	Productos abandonados en el carrito.	Permite realizar recomendaciones personalizadas basadas en intentos de compra.
checkout_abandonment_step	Paso del *checkout* donde el usuario abandonó.	Identifica los puntos críticos donde se pierde más tráfico.
time_spent_before_abandonment	Tiempo que el usuario pasó antes de abandonar.	Ayuda a detectar si el usuario estuvo indeciso o si abandonó rápidamente.
Justificación: Son variables esenciales para identificar patrones de abandono y diseñar estrategias de *retargeting* y *remarketing*.		
7. Datos sobre Interacciones con Promociones y Búsqueda		
promotion_interaction	Indica si el usuario interactuó con una promoción.	Permite analizar la efectividad de las promociones y su impacto en la conversión.
organic_search_query	Términos de búsqueda orgánica utilizados por el usuario.	Útil para entender la intención del usuario y mejorar la estrategia de contenido.
Justificación: Variables clave para evaluar la efectividad de estrategias promocionales y de búsqueda interna.		
8. Identificación del Producto		
product_id	Identificador único del producto.	Esencial para rastrear qué productos interesan a cada usuario y analizar patrones de compra.
item_name	Nombre del producto.	Facilita la interpretación de los datos y la segmentación basada en atributos de producto.

Justificación: Estas variables permiten rastrear qué productos fueron visualizados, añadidos al carrito o comprados. Son fundamentales para un sistema de recomendación, ya que ayudan a entender qué productos son populares y cómo se relacionan entre sí.		
9. Características del Producto		
item_category	Categoría a la que pertenece el producto.	Permite identificar patrones de interés en distintas categorías de productos y mejorar recomendaciones.
item_price	Precio del producto.	Es un factor clave en el análisis del comportamiento del usuario y en la conversión.
Justificación: La categoría de producto (*item_category*) permite analizar qué tipos de productos tienen mejor rendimiento en diferentes segmentos de clientes. El precio (*item_price*) ayuda a segmentar usuarios según sensibilidad al precio y realizar estrategias de personalización de promociones y descuentos.		
10. Interacción del Usuario con el Producto		
item_interaction_type	Tipo de interacción con el producto (ej. *"view", "add_to_cart", "purchase"*).	Permite rastrear el comportamiento del usuario en distintas fases del *funnel* de compra.
Justificación: Esta variable es clave para definir el *customer journey* de un usuario, desde la exploración hasta la compra. Se puede utilizar para detectar usuarios con alta intención de compra (ej., aquellos que añaden productos al carrito repetidamente pero no compran). También permite analizar qué productos tienen altas tasas de conversión y cuáles generan interés sin compra.		

ANEXO 2: CÓDIGO DE GENERACIÓN DE DATASET SINTÉTICO

```
import pandas as pd
import numpy as np
import random

# Subir "Info producto.xlsx" a Google Colab antes de ejecutar este código
file_path = "Info producto.xlsx"
product_data = pd.read_excel(file_path)

# Definir la cantidad de filas
num_rows = 60000

# Generar user_id y session_id
user_ids = [f"user_{random.randint(16001, 32000)}" for _ in
range(num_rows)]
session_ids = [f"session_{i}" for i in range(num_rows)]

# Fuente de tráfico
sources = ["Direct", "Organic Search", "Paid Search", "Referral", "Social",
"Email",
           "Paid Social", "Organic Social", "Paid Shopping",
"Notifications"]
source_values = random.choices(sources, k=num_rows)

# Dispositivos y navegadores
devices = ["Mobile", "Desktop", "Tablet"]
device_values = random.choices(devices, k=num_rows)

browsers = ["Safari", "Chrome", "Samsung Internet", "Edge", "Firefox",
"Opera"]
browser_values = random.choices(browsers, k=num_rows)

# new_user y audience_segment
new_users = random.choices(["YES", "NO"], k=num_rows)
audience_segments = ["New Visitor" if new_users[i] == "YES" else
random.choice(["Loyal Customer", "Returning Customer"])
                     for i in range(num_rows)]

# Asignar event_category
event_categories = ["Shopping"] * int(num_rows * 0.35) + ["Product
Interaction"] * int(num_rows * 0.35)
event_categories += random.choices(["Engagement", "Account Management",
"Promotions & Rewards", "Technical Events"],
                                   k=num_rows - len(event_categories))
random.shuffle(event_categories)

# Definir event_type
category_to_events = {
```

```
    "Shopping": ["view_cart", "add_to_cart", "remove_from_cart",
"begin_checkout", "purchase"],
    "Product Interaction": ["view_item", "add_to_wishlist",
"compare_products", "write_review"],
    "Engagement": ["page_view", "user_engagement", "menu_click"],
    "Account Management": ["login", "logout", "account_registration"],
    "Promotions & Rewards": ["select_promotion", "view_promotion"],
    "Technical Events": ["privacy_consent", "exception"]
}
event_types = [random.choice(category_to_events[cat]) for cat in
event_categories]

# Tiempo de participación en la sesión
session_engagement_time = [random.randint(5, 600) for _ in range(num_rows)]

# Producto solo si es Shopping o Product Interaction
product_info_needed = [cat in ["Shopping", "Product Interaction"] for cat
in event_categories]
product_ids = [random.choice(product_data["product_id"].tolist()) if
product_info_needed[i] else "" for i in range(num_rows)]
item_names = [product_data.loc[product_data["product_id"] ==
product_ids[i], "item_name"].values[0] if product_ids[i] else "" for i in
range(num_rows)]
item_categories = [product_data.loc[product_data["product_id"] ==
product_ids[i], "item_category"].values[0] if product_ids[i] else "" for i
in range(num_rows)]
item_prices = [product_data.loc[product_data["product_id"] ==
product_ids[i], "item_price"].values[0] if product_ids[i] else "" for i in
range(num_rows)]

# Asignar item_interaction_type
item_interaction_types = ["View", "Click", "Add to Cart", "Zoom", "Share",
"Rate/Review"]
item_interaction_values = [random.choice(item_interaction_types) if
product_info_needed[i] else "" for i in range(num_rows)]

# Asignar "Has Items" al 60% de los eventos de "Shopping" o "Product
Interaction"
shopping_product_indices = [i for i, cat in enumerate(event_categories) if
cat in ["Shopping", "Product Interaction"]]
num_has_items = int(len(shopping_product_indices) * 0.6)
selected_has_items = random.sample(shopping_product_indices, num_has_items)
cart_status_values = ["Has Items" if i in selected_has_items else "Empty"
if product_info_needed[i] else "" for i in range(num_rows)]

# Asignar checkout_step solo si cart_status es "Has Items"
checkout_steps = ["Begin checkout", "Add shipping", "Add payment",
"Purchase"]
checkout_step_values = [random.choice(checkout_steps) if
cart_status_values[i] == "Has Items" else "" for i in range(num_rows)]
```

```
# Asignar revenue solo si checkout_step es "Purchase"
revenues = [item_prices[i] if checkout_step_values[i] == "Purchase" else ""
for i in range(num_rows)]

# Asignar conversion_flag basado en checkout_step
conversion_flags = ["YES" if checkout_step_values[i] == "Purchase" else
"NO" if checkout_step_values[i] else "" for i in range(num_rows)]

# Asignar promotion_interaction
promotion_interaction = ["YES" if cat in ["Shopping", "Promotions &
Rewards", "Product Interaction"] else "NO" for cat in event_categories]

# Asignar organic_search_query
organic_search_query = [item_names[i] if "Search" in source_values[i] else
"" for i in range(num_rows)]

# Asignar abandoned_cart_flag, abandoned_products,
checkout_abandonment_step y time_spent_before_abandonment
abandoned_cart_flags = ["YES" if conversion_flags[i] == "NO" else "NO" for
i in range(num_rows)]
abandoned_products = [product_ids[i] if abandoned_cart_flags[i] == "YES"
else "" for i in range(num_rows)]
checkout_abandonment_steps = [checkout_step_values[i] if
abandoned_cart_flags[i] == "YES" else "" for i in range(num_rows)]
time_spent_before_abandonment = [random.randint(5, 600) if
abandoned_cart_flags[i] == "YES" else "" for i in range(num_rows)]

# Crear DataFrame
df = pd.DataFrame({
    "user_id": user_ids,
    "session_id": session_ids,
    "source": source_values,
    "device": device_values,
    "browser": browser_values,
    "new_user": new_users,
    "audience_segment": audience_segments,
    "event_category": event_categories,
    "event_type": event_types,
    "session_engagement_time": session_engagement_time,
    "product_id": product_ids,
    "item_name": item_names,
    "item_category": item_categories,
    "item_price": item_prices,
    "item_interaction_type": item_interaction_values,
    "cart_status": cart_status_values,
    "checkout_step": checkout_step_values,
    "revenue": revenues,
    "conversion_flag": conversion_flags,
    "promotion_interaction": promotion_interaction,
```

```
    "organic_search_query": organic_search_query,
    "abandoned_cart_flag": abandoned_cart_flags,
    "abandoned_products": abandoned_products,
    "checkout_abandonment_step": checkout_abandonment_steps,
    "time_spent_before_abandonment": time_spent_before_abandonment
})
# Guardar el DataFrame en un archivo Excel
df.to_excel("dataset_sintetico.xlsx", index=False)

print("Archivo generado :)")
```

ANEXO 3: CÓDIGO DE PREPARACIÓN DEL ENTORNO

```
# Importar librerías necesarias
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import random
import tensorflow as tf

# Librerías de Machine Learning
from sklearn.model_selection import train_test_split, cross_val_score,
GridSearchCV
from sklearn.preprocessing import StandardScaler, MinMaxScaler,
LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score, f1_score, roc_auc_score,
    confusion_matrix, classification_report
)
from xgboost import XGBClassifier

# Librerías de Deep Learning
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Input
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.optimizers import Adam

# Librería para optimización bayesiana
!pip install bayesian-optimization
from bayes_opt import BayesianOptimization

# Configuración del entorno
random_seed = 42  # Fijamos la semilla para reproducibilidad
np.random.seed(random_seed)
random.seed(random_seed)
tf.random.set_seed(random_seed)
print("Entorno preparado correctamente :)")
```

ANEXO 4: CÓDIGO DE EXPLORACIÓN Y PREPROCESAMIENTO DE DATOS

```
# Cargar datos sintéticos
data_path = "Datos_sinteticos_finales.xlsx"
df = pd.read_excel(data_path)

# --------------------
# EXPLORACIÓN DE DATOS
# --------------------

# Información general del dataset
print("\033[1mInformación general del dataset:\033[0m")
print("--------------------------------\n")
print(df.info())

# Resumen estadístico
print("\033[1mResumen estadístico:\033[0m")
print("--------------------\n")
print(df.describe())

# Valores nulos
print("\033[1mValores nulos por columna:\033[0m")
print("-------------------------\n")
print(df.isnull().sum())

# Visualización de distribuciones
plt.figure(figsize=(12, 6))
sns.histplot(df["session_engagement_time"], bins=30, kde=True)
plt.title("Distribución del tiempo de interacción por sesión")
plt.show()

# -----------------
# LIMPIEZA DE DATOS
# -----------------

# Manejo valores nulos
df.fillna({"conversion_flag": "NO", "revenue": 0,
"time_spent_before_abandonment": 0, "item_price": 0}, inplace=True)

# Conversión de variables categóricas
label_encoders = {}
categorical_cols = ["source", "device", "browser", "new_user",
"audience_segment",
                    "event_category", "event_type", "cart_status",
"checkout_step",
                    "promotion_interaction", "organic_search_query",
"abandoned_cart_flag",
                    "abandoned_products", "checkout_abandonment_step",
"product_id", "item_name", "item_category",
                    "item_interaction_type"]
```

```
for col in categorical_cols:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col].astype(str))
    label_encoders[col] = le

# ---------------------------------
# NORMALIZACIÓN Y ESCALADO DE DATOS
# ---------------------------------

# Convertir las columnas a numéricas, manejando valores no válidos
df["session_engagement_time"] =
pd.to_numeric(df["session_engagement_time"], errors='coerce')
df["time_spent_before_abandonment"] =
pd.to_numeric(df["time_spent_before_abandonment"], errors='coerce')
df["revenue"] = pd.to_numeric(df["revenue"], errors='coerce')

# Reemplazar NaN generados por conversiones fallidas con 0
df[["session_engagement_time", "time_spent_before_abandonment", "revenue"]]
= df[["session_engagement_time", "time_spent_before_abandonment",
"revenue"]].fillna(0)

# Escalado con MinMaxScaler
scaler = MinMaxScaler()
scaled_cols = ["session_engagement_time", "time_spent_before_abandonment",
"revenue"]
df[scaled_cols] = scaler.fit_transform(df[scaled_cols])
```

ANEXO 5: CÓDIGO DE DEFINICIÓN DE VARIABLE OBJETIVO PARA MODELO DE CONVERSIÓN

```
# Conversión de la variable objetivo a valores numéricos binarios
df["conversion_flag"] =
df["conversion_flag"].fillna("NO").str.upper().str.strip()  # Manejo de
valores nulos y normalización de texto
y_conversion = df["conversion_flag"].map({"NO": 0, "YES": 1}).astype(int)

#Conversión de user_id en features relevantes
print("Generando features a partir de user_id...")
user_features = df.groupby('user_id').agg(
    total_sessions=('session_id', 'count'),
    avg_session_time=('session_engagement_time', 'mean'),
    conversion_rate=('conversion_flag', lambda x: (x == "YES").mean())
).reset_index()

# Guardar el mapeo de user_id a sus nuevas features (por si acaso lo
necesitamos más adelante para identificar sesiones de user individuales)
user_feature_dict =
user_features.set_index("user_id").to_dict(orient="index")

# Unir las nuevas features al dataset
```

```
df = df.merge(user_features, on='user_id', how='left')

# Eliminar user_id y session_id ya que ahora tenemos sus features
df.drop(columns=['user_id', 'session_id'], inplace=True, errors='ignore')

print("Features generadas y user_id/session_id eliminados.")

# Aplicar Label Encoding a nuevas features
categorical_cols = ['total_sessions', 'avg_session_time',
'conversion_rate']
label_encoders = {}

for col in categorical_cols:
    if col in df.columns:
        le = LabelEncoder()
        df[col] = le.fit_transform(df[col].astype(str))
        label_encoders[col] = le
```

Después de modificar el *dataset*, es necesario volver a dividir los datos en entrenamiento y prueba, así nos aseguramos de que los modelos trabajen con la versión correcta.

```
# División del dataset en entrenamiento y prueba

X = df.drop(columns=["conversion_flag", "recommended_product"])

y_conversion = df["conversion_flag"].map({"NO": 0, "YES": 1}).astype(int)

# Aplicar Label Encoding a la variable objetivo de recomendación asegurando
todas las clases posibles
y_rec_encoder = LabelEncoder()
y_rec_encoder.fit(df["recommended_product"].astype(str).unique())
y_recommendation =
y_rec_encoder.transform(df["recommended_product"].astype(str))

# División para predicción de conversión
X_train_conv, X_test_conv, y_train_conv, y_test_conv = train_test_split(
    X, y_conversion, test_size=0.2, random_state=42, stratify=y_conversion)

# División para recomendación de productos
X_train_rec, X_test_rec, y_train_rec, y_test_rec = train_test_split(
    X, y_recommendation, test_size=0.2, random_state=42)

# Reasignar las etiquetas después de la división
y_train_rec = y_rec_encoder.transform(y_train_rec.astype(str))
y_test_rec = y_rec_encoder.transform(y_test_rec.astype(str))

# Obtener número de clases asegurando que coincide con la capa de salida
num_classes = len(y_rec_encoder.classes_)
```

ANEXO 6: CÓDIGO DE DEFINICIÓN DE HIPERPARÁMETROS PARA MODELO DE CONVERSIÓN

Random Forest

```
rf_params = {
    "n_estimators": 4,  # Reducido para evitar sobreajuste
    "max_depth": 2,  # Menor profundidad
    "min_samples_split": 6,
    "min_samples_leaf": 4,
    "max_features": "sqrt",  # Mayor aleatorización
    "bootstrap": True,  # Mayor aleatorización
    "random_state": 42
}
rf_model = RandomForestClassifier(**rf_params)
```

XGBoost

```
xgb_params = {
    "n_estimators": 60,  # Reducimos la cantidad de árboles
    "learning_rate": 0.015,  # Aprendizaje más controlado
    "max_depth": 2,  # Seguimos limitando la profundidad
    "subsample": 0.3,  # Menos datos en cada iteración para mayor
diversidad
    "colsample_bytree": 0.15,  # Aún menor dependencia de features
    "min_child_weight": 18,  # Aumentado para reducir divisiones
innecesarias
    "lambda": 6.0,  # Mayor regularización L2
    "alpha": 5.0,  # Mayor regularización L1
    "gamma": 12.0,  # Penalización aún más fuerte
    "random_state": 42
}
xgb_model = XGBClassifier(**xgb_params)
```

ANEXO 7: CÓDIGO DE ENTRENAMIENTO DEL MODELO DE CONVERSIÓN

Random Forest

```
print("Entrenando Random Forest con validación cruzada...")
cv_scores_rf = cross_val_score(rf_model, X_train_conv, y_train_conv, cv=10)
rf_model.fit(X_train_conv, y_train_conv)
print(f"Random Forest entrenado correctamente. Validación cruzada (accuracy
media): {np.mean(cv_scores_rf):.4f}")
```

XGBoost

```
print("Entrenando XGBoost con optimización de gradiente...")
xgb_model.fit(X_train_conv, y_train_conv, eval_set=[(X_test_conv,
y_test_conv)], verbose=False)
print("XGBoost entrenado correctamente.")
```

```
# Validación cruzada para XGBoost
cv_scores_xgb = cross_val_score(xgb_model, X_train_conv, y_train_conv,
cv=5)
print(f"XGBoost validado con validación cruzada. Accuracy media:
{np.mean(cv_scores_xgb):.4f}")
```

ANEXO 8: CÓDIGO DE DEFINICIÓN DE VARIABLE OBJETIVO PARA MODELO DE RECOMENDACIÓN

```
# Generación de la variable "recommended_product"
def get_recommended_product(group):
    if group["revenue"].sum() > 0:  # Si el usuario compró, recomendamos el
producto más caro
        return group.loc[group["item_price"].idxmax(), "product_id"]
    elif "added" in group["cart_status"].values:  # Si agregó al carrito,
recomendamos el más reciente
        return group.loc[group["cart_status"] == "added",
"product_id"].iloc[-1]
    elif group["item_interaction_type"].notna().sum() > 0:  # Si interactuó
con productos, el más frecuente
        return group["product_id"].mode()[0]
    elif group["item_category"].notna().sum() > 0:  # Si no hay
interacciones, la categoría más visitada
        return group["item_category"].mode()[0]
    else:
        return "Unknown_Product"  # Si no hay datos suficientes, un
producto genérico

# Aplicamos la función a cada usuario
df["recommended_product"] =
df.groupby("user_id").apply(get_recommended_product).reset_index(level=0,
drop=True)

# Variable objetivo para recomendación de productos (multiclase)
le = LabelEncoder()
df["recommended_product"] =
df["recommended_product"].fillna("Unknown_Product").astype(str).str.upper()
.str.strip()  # Manejo de valores nulos y normalización de texto
df["recommended_product"] = le.fit_transform(df["recommended_product"])
y_recommendation = df["recommended_product"]
```

ANEXO 9: CÓDIGO DE OPTIMIZACIÓN Y ENTRENAMIENTO DE RED NEURONAL

```
# ------------------------------------------
# OPTIMIZACIÓN BAYESIANA PARA LA RED NEURONAL
# ------------------------------------------

def nn_optimization(learning_rate, batch_size):
    batch_size = int(batch_size)
    model = Sequential([
        Input(shape=(X_train_rec.shape[1],)),
        Dense(128, activation='relu'),
        Dropout(0.4),
        Dense(64, activation='relu'),
        Dropout(0.4),
        Dense(num_classes, activation='softmax')
    ])

model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=learning_rat
e), loss='sparse_categorical_crossentropy', metrics=['accuracy'])

    early_stopping = EarlyStopping(monitor='val_loss', patience=5,
restore_best_weights=True)
    history = model.fit(
        X_train_rec, y_train_rec,
        epochs=10, batch_size=batch_size, validation_split=0.2,
callbacks=[early_stopping], verbose=0
    )
    return max(history.history['val_accuracy'])

param_bounds = {'learning_rate': (0.0001, 0.01), 'batch_size': (16, 128)}
optimizer = BayesianOptimization(f=nn_optimization, pbounds=param_bounds,
random_state=42)
optimizer.maximize(init_points=2, n_iter=5)
print(f"Mejores hiperparámetros de la Red Neuronal: {optimizer.max}")

# ------------------------------------------------------------
# ENTRENAMIENTO DE RED NEURONAL CON HIPERPARÁMETROS OPTIMIZADOS
# ------------------------------------------------------------
best_params = optimizer.max['params']
nn_model = Sequential([
    Input(shape=(X_train_rec.shape[1],)),
    Dense(128, activation='relu'),
    Dropout(0.4),
    Dense(64, activation='relu'),
    Dropout(0.4),
    Dense(num_classes, activation='softmax')
])
```

```
nn_model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=best_para
ms['learning_rate']), loss='sparse_categorical_crossentropy',
metrics=['accuracy'])

early_stopping = EarlyStopping(monitor='val_loss', patience=5,
restore_best_weights=True)
nn_model.fit(X_train_rec, y_train_rec, epochs=50,
batch_size=int(best_params['batch_size']), validation_split=0.2,
callbacks=[early_stopping], verbose=1)
print("Red Neuronal entrenada correctamente.")

# Evaluación de la Red Neuronal
y_pred_nn = nn_model.predict(X_test_rec)
y_pred_nn = np.argmax(y_pred_nn, axis=1)
accuracy_nn = accuracy_score(y_test_rec, y_pred_nn)
print(f"Precisión en el conjunto de prueba para la Red Neuronal:
{accuracy_nn:.4f}")
```

ANEXO 10: CÓDIGO DE EVALUACIÓN DE MODELOS

```
# ------------------------------
# PREDICCIONES DATOS DE PRUEBA
# ------------------------------

# Predicciones para conversión (modelos binarios)
y_pred_rf = rf_model.predict(X_test_conv)
y_pred_xgb = xgb_model.predict(X_test_conv)

# Predicciones para recomendación de productos (modelo multiclase)
y_pred_nn = nn_model.predict(X_test_rec)
y_pred_nn = np.argmax(y_pred_nn, axis=1)  # Tomar la clase con mayor
probabilidad

print(" Predicciones obtenidas para cada modelo.")
print(f"Ejemplo de predicciones - Random Forest: {y_pred_rf[:10]}")
print(f"Ejemplo de predicciones - XGBoost: {y_pred_xgb[:10]}")
print(f"Ejemplo de predicciones - Red Neuronal: {y_pred_nn[:10]}")

# ------------------------------------------------------------
# CALCULAR MÉTRICAS DE EVALUACIÓN PARA PREDICCIÓN DE CONVERSIÓN
# ------------------------------------------------------------
def evaluate_binary_model(name, y_true, y_pred):
    precision = precision_score(y_true, y_pred)
    recall = recall_score(y_true, y_pred)
    f1 = f1_score(y_true, y_pred)
    auc_roc = roc_auc_score(y_true, y_pred)
    print(f"\n {name} Metrics (Conversión):")
    print(f"Precision: {precision:.4f}")
    print(f"Recall: {recall:.4f}")
    print(f"F1-Score: {f1:.4f}")
```

```
    print(f"AUC-ROC: {auc_roc:.4f}")
    print("\nClassification Report:")
    print(classification_report(y_true, y_pred))

# Evaluación de modelos binarios
evaluate_binary_model("Random Forest", y_test_conv, y_pred_rf)
evaluate_binary_model("XGBoost", y_test_conv, y_pred_xgb)

# ---------------------------------------------------------------
# CALCULAR MÉTRICAS DE EVALUACIÓN PARA RECOMENDACIÓN DE PRODUCTOS
# ---------------------------------------------------------------
def evaluate_multiclass_model(name, y_true, y_pred):
    accuracy = accuracy_score(y_true, y_pred)
    print(f"\n {name} Metrics (Recomendación de Productos):")
    print(f"Accuracy: {accuracy:.4f}")
    print("\nClassification Report:")
    print(classification_report(y_true, y_pred))

# Evaluación del modelo de recomendación de productos
evaluate_multiclass_model("Red Neuronal", y_test_rec, y_pred_nn)

# ---------------------------
# VISUALIZACIÓN DE RESULTADOS
# ---------------------------

# Visualización de Matrices de Confusión (Random Forest y XGBoost)
def plot_confusion_matrix(y_true, y_pred, model_name):
    cm = confusion_matrix(y_true, y_pred)
    plt.figure(figsize=(6,5))
    sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
    plt.xlabel("Predicción")
    plt.ylabel("Real")
    plt.title(f"Matriz de Confusión - {model_name}")
    plt.show()
    print(f"\n Matriz de Confusión - {model_name}:")
    print(cm)

# Mostrar matrices de confusión para Random Forest y XGBoost
plot_confusion_matrix(y_test_conv, y_pred_rf, "Random Forest")
plot_confusion_matrix(y_test_conv, y_pred_xgb, "XGBoost")

# Tabla resumen de matriz de confusión para Red Neuronal
def confusion_matrix_summary(name, y_true, y_pred):
    cm = confusion_matrix(y_true, y_pred)
    cm_summary = pd.DataFrame(cm).sum(axis=1)
    print(f"\n Matriz de Confusión - {name} (Resumida):")
    print(f"Total predicciones correctas: {cm.trace()}")
    print(f"Total ejemplos evaluados: {cm.sum()}")
    print(f"Distribución por clase:\n{cm_summary}")
confusion_matrix_summary("Red Neuronal", y_test_rec, y_pred_nn)
```