IDENTIFICACIÓN DE PRODUCTOS EXITOSOS POR CLIENTE

Catherine Johanna Rojas Mendoza[1,†], Luis Maximiliano López Ramírez[2,†], Adrián Pineda Sánchez[3,†], Rogelio Lizárraga Escobar[4,†] y Rodolfo Jesús Cruz Rebollar[5, †]

[1]*ITESM, Monterrey – a01798149@tec.mx*
[2]*ITESM, Monterrey – a00833321@tec.mx*
[3]*ITESM, Monterrey – a00834710@tec.mx*
[4]*ITESM, Monterrey – a01742161@tec.mx*
[5]*ITESM, Monterrey – a01368326@tec.mx*

RESUMEN

Este reporte analiza un proyecto desarrollado para Arca Continental, centrado en la predicción de la probabilidad de que un cliente adopte un producto nuevo con un enfoque en el cumplimiento ético y normativo en el manejo de grandes volúmenes de datos e inteligencia artificial (IA). El enfoque integral permite alinear los objetivos de Arca Continental con las mejores prácticas éticas y legales en el uso de la tecnología.

Palabras-clave: *Arca Continental; clustering de clientes; predicción de éxito; inteligencia artificial; ética; normativas; protección de datos; regulación; transparencia; privacidad.*

1. INTRODUCCIÓN

La problemática que enfrenta Arca Continental es optimizar sus esfuerzos comerciales al identificar, entre sus clientes, aquellos con mayor probabilidad de éxito en la adopción de nuevos productos. Actualmente, la introducción de productos nuevos es un proceso de alto riesgo, pues no todos los clientes muestran el mismo interés o capacidad para adoptarlos exitosamente. Para abordar este reto, el proyecto aplicará modelos de inteligencia artificial que analicen patrones de compra, características de los clientes y otros datos relevantes. Esto permitirá a Arca Continental enfocar sus estrategias de marketing y ventas hacia aquellos clientes con mayores probabilidades de aceptación, optimizando recursos y aumentando la eficiencia en el lanzamiento de productos nuevos.

2. OBJETIVOS

Objetivo General

Identificar a los clientes con mayor probabilidad de éxito en la adopción de productos de lanzamiento, maximizando el beneficio tanto para Arca Continental como para sus clientes.

Objetivos Específicos

1. Seleccionar las herramientas y tecnologías necesarias para la implementación del proyecto.
2. Realizar la limpieza, transformación y preparación de los datos proporcionados por Arca Continental para su adecuado uso en los modelos.
3. Identificar las características más relevantes que permitan predecir la tasa de éxito de los productos de lanzamiento.
4. Entrenar diversos modelos de inteligencia artificial para optimizar las predicciones.
5. Evaluar los modelos empleando múltiples métricas de desempeño para seleccionar el modelo más efectivo.

3. ANTECEDENTES Y PROYECTOS RELACIONADOS

3.1. Análisis del impacto de la IA: oportunidades y desafíos éticos

La inteligencia artificial (IA) está revolucionando sectores como finanzas, salud, transporte y educación, mejorando eficiencia y personalización. No obstante, enfrenta desafíos éticos y normativos, como la privacidad de datos, sesgos y dilemas éticos en vehículos autónomos. Organismos como la UNESCO y la OMS subrayan la necesidad de regulaciones robustas para garantizar un desarrollo responsable.[1, 2, 3]

En redes sociales, la IA amplifica la desinformación, mientras que en justicia y seguridad plantea riesgos de sesgos y privacidad. En respuesta, se promueven marcos normativos que aseguren transparencia y equidad.[4, 5]

IA

La IA ofrece beneficios significativos, pero requiere cooperación internacional y marcos regulatorios sólidos para un uso ético y equitativo.

3.2. Casos Destacados de Adaptación Ética y Normativa en el Uso de Big Data e IA

Caso Cambridge Analytica y la Respuesta de Facebook (2018)

El acceso indebido a datos de 87 millones de usuarios por Cambridge Analytica evidenció fallas en la protección de datos, lo que llevó a la implementación del *GDPR*, que exige transparencia y consentimiento explícito en el manejo de datos personales.[6]

Implementación de la Ley de Gobernanza de Datos en la Unión Europea (2022)

La *Ley de Gobernanza de Datos* establece un marco para la reutilización segura de datos del sector público y fomenta el altruismo de datos, promoviendo la innovación y garantizando altos estándares de seguridad.[7]

Caso Vodafone y la Sanción por Incumplimiento del GDPR (2021)

Vodafone fue multada con 8,15 millones de euros por gestionar datos personales sin consentimiento adecuado. Este caso reforzó la necesidad de adherirse estrictamente al *GDPR*, impulsando revisiones en las políticas internas de datos.[8]

Desarrollo de Plataformas Éticas para IA (2023)

Empresas como Telefónica y BBVA implementaron plataformas para garantizar un uso ético y transparente de la IA, cumpliendo con normativas europeas que exigen responsabilidad, equidad y mitigación de sesgos.[9]

En conjunto, estos casos destacan cómo gobiernos y empresas están adoptando marcos éticos y normativos para un uso responsable de la IA y Big Data, protegiendo los derechos fundamentales en la era digital.

3.3. Arca Continental

La inteligencia artificial ha revolucionado el sector retail y de consumo, permitiendo identificar a los clientes potenciales de nuevos productos y predecir comportamientos de compra mediante el análisis de datos. Esto facilita la segmentación de audiencias y la detección de tendencias emergentes, como cambios en preferencias o nuevos segmentos de mercado. [10]

Aplicaciones de la IA

1. **Personalización de la Experiencia del Cliente:** Ofrece recomendaciones personalizadas basadas en preferencias del cliente, mejorando la satisfacción y fidelización. [11]
2. **Gestión del Inventario:** Algoritmos de aprendizaje automático optimizan la gestión del inventario y la cadena de suministro, reduciendo costos. [12]
3. **Optimización de Precios:** Ajuste de precios en tiempo real basado en la demanda y competencia para maximizar ingresos y competitividad. [12]

Análisis Ético y Normativo

El uso de IA plantea desafíos éticos y normativos:

1. **Privacidad de los Datos:** Es crucial proteger la privacidad del consumidor y cumplir regulaciones como el GDPR. [13]
2. **Transparencia y Equidad:** Los algoritmos deben ser transparentes y no discriminar, evitando perjudicar la reputación empresarial. [13]
3. **Regulación de la IA:** La Ley de IA de la UE busca equilibrar innovación y derechos fundamentales regulando los sistemas de IA según su nivel de riesgo. [14]

Implementación de IA en el sector retail

Ofrece ventajas significativas en la identificación de clientes potenciales y la optimización de operaciones. Sin embargo, es fundamental abordar los desafíos éticos y normativos asociados para garantizar un uso responsable y beneficioso tanto para las empresas como para los consumidores.

4. HERRAMIENTAS Y RECURSOS A USAR

4.1. Software y Frameworks

En el reto, se utilizó principalmente Visual Studio Code como entorno de desarrollo integrado, junto

con el lenguaje de programación Python, para llevar a cabo todo el proceso de desarrollo. Este proceso incluyó desde el preprocesamiento de datos hasta la selección del modelo con mejor desempeño tras evaluar diversas opciones.

4.1.1. Frameworks y bibliotecas de aprendizaje automático

- **Scikit-learn** fue empleado para construir modelos como:
  - **Regresión logística**: `from sklearn.linear_model import LogisticRegression`
  - **Árbol de decisión**: `from sklearn.tree import DecisionTreeClassifier`
  - **Bosque aleatorio**: `from sklearn.ensemble import RandomForestClassifier`
  - **KNN**: `from sklearn.neighbors import KNeighborsClassifier`
  - **Naive Bayes**: `from sklearn.naive_bayes import GaussianNB`
  - **RFE (Recursive Feature Elimination)**: `from sklearn.feature_selection import RFE`
- **TensorFlow** se utilizó para implementar y entrenar redes neuronales en las siguientes arquitecturas:
  - **LSTM**: `from tensorflow.keras.layers import LSTM`
  - **Modelo denso simple (Simple Dense Model)**: `from tensorflow.keras.models import Sequential, Dense`
  - **Modelo convolucional + denso (Conv + Dense Model)**: `from tensorflow.keras.layers import Conv1D, Dense`

  También se utilizó para el **callback ReduceLROnPlateau**, importado como: `from tensorflow.keras.callbacks import ReduceLROnPlateau`, para reducir la tasa de aprendizaje cuando no se detecta mejora en el desempeño del modelo durante el entrenamiento.
- **Optuna** se utilizó para la optimización de hiperparámetros en estas arquitecturas, ajustando parámetros como el número de unidades, tasa de aprendizaje, número de capas, tasa de dropout, entre otros.

4.1.2. Otras bibliotecas fundamentales:

- **NumPy:** Facilitó operaciones matemáticas y manipulación de arreglos.
- **Pandas:** Proporcionó herramientas para:
  - Gestión eficiente de datos.
  - Escalado de valores.
  - Imputación de datos faltantes.
  - Codificación categórica mediante **One-Hot Encoding**.
- **Bibliotecas de visualización:**
  - **Matplotlib**, **Seaborn** y **Plotly Express** se utilizaron para generar gráficos personalizados que permitieron explorar y comunicar insights clave.

4.2. Hardware y servicios

El proyecto se desarrolló localmente en los ordenadores portátiles del equipo, utilizando CPUs para entrenar los modelos de redes neuronales. Estas CPUs permitieron ejecutar los cálculos necesarios para el entrenamiento de manera eficiente, optimizando los recursos disponibles. Aunque las CPUs tienen una menor capacidad para cálculos masivos en paralelo en comparación con las GPUs, fueron suficientes para lograr un entrenamiento adecuado de las redes neuronales dentro de los plazos establecidos.

4.3. Fuentes de información

Las fuentes de información utilizadas para el desarrollo del reto fueron las bases de datos de clientes, ventas y productos, las cuales fueron compartidas por el socio formador, Arca Continental, y a continuación se describe cada una de ellas:

4.3.1. Base de datos de clientes

- Tiene un total de 2,041 registros y 207 columnas.

- Los datos se compartieron en formato csv por la empresa.

- Tiene 202 columnas de tipo numérico flotante, 2 de tipo numérico entero y 3 de tipo objeto o categoría.

- Los datos hacen referencia a las características de los clientes que posee la empresa de Arca Continental hasta el momento tales como su número de id, porcentaje de gastos estimados en diferentes servicios, estatus socioeconómico, entre otros aspectos.

4.3.2. Base de datos de ventas

- Posee un total de 2,347,110 registros y 4 columnas.

- Los datos fueron compartidos por la empresa en formato csv.

- la tabla tiene 3 columnas de tipo numérico entero y 1 de tipo numérico flotante.

- Los datos describen las características de las ventas realizadas por Arca Continental, como lo son: id de la venta, uni_box vendido, id del material vendido, entre otras características.

4.3.3. Base de datos de productos

- Tiene un total de 793 registros y 22 columnas.

- Los datos de compartieron en formato csv.

- La tabla de datos tiene 18 columnas de tipo objeto o categóricas y 4 de tipo numérico entero.

- Los datos describen las características de los productos que maneja la empresa de Arca Continental para su venta, tales como: refrescos, agua, jugos, entre otros.

5. METODOLOGÍA

5.1. Extracción y limpieza

Importación y carga de datos:

Se importaron los datos de tres fuentes:

- **Clientes**: `customers_sampled.csv`
- **Productos**: `20230223_productos.csv`
- **Ventas**: `ventas.csv`

Los archivos fueron leídos en DataFrames de Pandas (`df_clientes`, `df_productos`, `df_ventas`) para su manipulación.

Exploración inicial de los datos

Se aplicó `info()` sobre los DataFrames para verificar la estructura, tipo de datos y la existencia de valores nulos.

Transformación de la fecha en el DataFrame de ventas

La columna `calmonth` en `df_ventas` se transformó al formato de fecha (`YYYY-MM`) utilizando `pd.to_datetime`. Este paso permite el análisis de datos en base a meses.

A partir de la columna `Fecha`, se creó una columna adicional `Mes` para extraer el valor numérico del mes y facilitar la agregación mensual.

Agrupación de datos por cliente y material

Se agrupó el DataFrame `df_ventas` por las columnas `CustomerId` y `material`, aplicando las siguientes agregaciones:

- **Suma** de la columna `uni_box`, que representa la cantidad total de unidades vendidas en cajas por cliente y material.
- **Lista** de fechas de venta para cada cliente y material, proporcionando un registro de las fechas en que se realizaron las ventas.

Agregación mensual de ventas por cliente

Se realizó una segunda agrupación de `df_ventas`, esta vez por `CustomerId` y `Mes`, con el objetivo de sumar las ventas mensuales de cada cliente en la columna `uni_box`.

El resultado fue convertido en un formato de tabla con columnas para cada mes (`mes_1`, `mes_2`, ..., `mes_12`), donde los valores representan el total de unidades vendidas por cliente en cada mes. Las columnas fueron renombradas para reflejar el mes correspondiente, mejorando la claridad en la interpretación de los datos.

Integración de los resultados

Se combinó el DataFrame `df_ventas_grouped` (con la agrupación por cliente y material) con el DataFrame `df_ventas_mes` (con las ventas mensuales de cada cliente), utilizando una combinación *left join* en la columna `CustomerId`.

El resultado (`df_result`) presenta tanto el desglose de ventas por cliente y material como las ventas mensuales de cada cliente, consolidando la información en un solo DataFrame para facilitar el análisis posterior.

5.2. Transformaciones

Escalado de datos mensuales de ventas

Para normalizar los datos y facilitar el análisis de tendencias de ventas, se aplicó un escalado Min-Max en las columnas de ventas mensuales (`mes_1` a `mes_12`).

Se seleccionaron las columnas mensuales, y se aplicó `MinMaxScaler()` a cada fila individualmente para asegurar que los valores de cada cliente estén en un rango entre 0 y 1.

Los valores escalados fueron reasignados a las columnas originales de ventas mensuales en el DataFrame `df_result`.

Creación de una columna para identificar "Producto Exitoso"

Se implementaron dos funciones para identificar productos exitosos basados en la frecuencia y continuidad de compras en ciertos meses:

La función `primeros_cinco_meses_consecutivos` se aplicó al DataFrame `df_ventas_grouped` para generar una nueva columna llamada `Producto Exitoso`, que marca con "1.ª los productos que cumplen con el criterio de éxito, y "0.en caso contrario.

Finalmente, se calculó el porcentaje de productos clasificados como exitosos o no exitosos en esta columna, proporcionando una visión general del desempeño de los productos en términos de continuidad de compra.

Integración final de los datos

Se realizó la unión de los DataFrames `df_ventas_grouped` y `df_clientes` utilizando `CustomerId` como clave para combinar la información de ventas con los datos de los clientes.

A continuación, se integró `df_productos` con el DataFrame resultante (`df_ventas_clientes`), utilizando `material` de ventas y `Material` de productos para realizar un *left join*.

El DataFrame final (`df_unido`) contiene información combinada de clientes, productos y ventas, proporcionando una estructura integral que permite el análisis detallado del comportamiento de compra y la evaluación de la efectividad de los productos.

Creación de columna `instituciones`

Se creó la columna `instituciones` sumando los valores de las columnas `preescolares`, `primarias`, `secundarias`, `preparatorias` y `universidades`. Esta nueva columna permite un análisis más compacto de la presencia de instituciones educativas en un área. Las columnas individuales utilizadas para esta suma fueron eliminadas para reducir la redundancia.

Conversión binaria de ciertas columnas mediante One-Hot Encoding

Para facilitar el análisis, se transformaron las columnas `parques`, `supermercados`, `hospitales`, `instituciones` y `gimnasios` en variables binarias. Cualquier valor mayor o igual a 1 se convirtió en 1 (indicando la presencia de ese tipo de lugar), y valores menores que 1 se transformaron en 0 (indicando su ausencia).

Creación de categorías de grupos de edad

Se generaron nuevas columnas para agrupar la población en diferentes etapas de la vida:

- `infancia_0_11`: suma de columnas que representan la población de 0 a 11 años.
- `adolescencia_12_17`: suma de columnas para la población de 12 a 17 años.
- `joven_Adulto_18_29`: suma de columnas para la población de 18 a 29 años.
- `adulto_30_49`: suma de columnas para la población de 30 a 49 años.
- `adulto_mayor_50_mas`: suma de columnas para la población de 50 años en adelante.

Estas nuevas categorías simplifican el análisis al agrupar edades en rangos significativos. Las columnas originales se eliminaron para evitar duplicación de información.

Escalado de modelos de hogar familiar y no familiar

Para estandarizar los datos de modelos de hogar familiar y no familiar, se aplicó un escalado mínimo-máximo, transformando ambas columnas a un rango entre 0 y 1.

Los valores nulos resultantes de este escalado fueron reemplazados con el promedio de cada columna para mantener la integridad de los datos.

Las columnas originales (`modelos_hog_fam_conteo_hogares_300m` y `modelos_hog_no_fam_conteo_hogares_300m`) fueron eliminadas tras la transformación.

Codificación de la columna `sub_canal_comercial` con One-Hot Encoding

Para analizar los distintos subcanales comerciales (como abarrotes, carnicerías, supermercados, etc.), se aplicó One-Hot Encoding a la columna `sub_canal_comercial`, generando una columna binaria para cada tipo de subcanal comercial.

Algunas columnas resultantes tenían nombres complejos debido a caracteres especiales, por lo que se renombraron utilizando un diccionario para mejorar la legibilidad. Por ejemplo, `'_Abarrotes / Almacenes / Bodegas / Vebeveres'` se renombró a `Abarrotes_Almacenes_Bodegas_Viveres`.

Transformaciones en las columnas de `df_productos`

One-Hot Encoding en columna `Container`

Se aplicó One-Hot Encoding a la columna `Container`, generando columnas binarias para cada tipo de contenedor (por ejemplo, `bolsa`, `lata`, `vidrio`, etc.), facilitando el análisis de datos por tipo de envase.

Se mantuvo una copia de la columna original (`Container_Original`) y se renombraron las columnas de contenedores para mejorar la claridad de los nombres.

One-Hot Encoding en columna `ProductType`

Se aplicó One-Hot Encoding a la columna `ProductType`, permitiendo la creación de columnas que representan cada tipo de producto (como `AGUA_FUNCIONAL`, `BEBIDAS_ENERGETICAS`, `TE`, entre otros).

Se conservó la columna original (`ProductType_Original`), y se renombraron las nuevas columnas para mejorar la legibilidad, de acuerdo con un mapeo definido.

One-Hot Encoding en columna `ProductCategory`

Se transformó la columna `ProductCategory` utilizando One-Hot Encoding para generar columnas binarias que representan cada categoría de producto. La columna original `ProductCategory_Original` se conservó para futuras referencias.

One-Hot Encoding en columna `BrandGrouper`

Para facilitar el análisis de marcas, se aplicó One-Hot Encoding a la columna `BrandGrouper`, generando una columna binaria para cada grupo de marca.

La columna original `BrandGrouper_Original` fue conservada.

One-Hot Encoding en columna `Flavor`

La columna `Flavor` se transformó mediante One-Hot Encoding, permitiendo un análisis detallado de los sabores disponibles. Las columnas resultantes representan los distintos sabores en el DataFrame.

La columna original (`Flavor_Original`) se mantuvo para posibles referencias.

One-Hot Encoding en columna `Size`

Se aplicó One-Hot Encoding a la columna `Size`, creando columnas binarias para cada tamaño de

Categoría de Columnas Eliminadas	Ejemplos de Columnas
Datos de población y demografía	PADRON_HOMBRES_300m, LISTA_18_HOMBRES_300m
Movilidad diaria y por hora	mov_lunes, mov_8_00_9_59, autos_hora_12
Gastos promedio y específicos	gasto_promedio_300m, pc_gasto_salud_300m
Ingresos y gastos por tipo	ingreso_promedio_300m, ingreso_rentas_300m
Datos de vivienda y entorno	viviendas_300m, prob_VPH_TV_300m
Flujos de personas	flo_sem_tot_300m, flo_finde_e_300m
Datos de accesibilidad y características del entorno	accesibilidad, arboles_300m
Modelos de hogar	modelos_una_persona_conteo_hogares_300m
Descripción del producto y clasificación	Material_desc, GlobalCategory, GlobalFlavor
Información de marca y presentación	Brand, Presentation, Pack
Segmentación y grupo de mercado	SegAg, SegDet, BrandPresRet

Cuadro 1. Categorías y ejemplos de columnas eliminadas

producto. Esta transformación facilita el análisis de ventas en función del tamaño del producto.

Se conservó la columna original (`Size_Original`) para fines de comparación y validación.

One-Hot Encoding en columna `Returnability`

La columna `Returnability` fue transformada mediante One-Hot Encoding, creando columnas que indican si un producto es retornable o no. Esto es útil para identificar patrones de consumo y preferencias de retorno.

La columna original (`Returnability_Original`) fue preservada.

Creación de la columna de número de productos distintos por cliente

Se generó la columna `Num_productos_distintos`, que contabiliza la cantidad de productos únicos que cada cliente ha probado, utilizando la columna `material`.

Esta columna permite analizar la diversidad de productos consumidos por cada cliente, proporcionando una medida indirecta de la fidelización o preferencia hacia nuevos productos.

Eliminación de registros con valores nulos

Para asegurar la consistencia de los datos en el análisis, se eliminaron todos los registros que contenían valores nulos, obteniendo el DataFrame `df_final` sin datos faltantes.

Creación de columnas dummy para el mes inicial de compra

La columna `Fecha` fue utilizada para extraer el mes de la primera compra de cada cliente (almacenada en la columna `mes_inicial`).

Se creó una función que aplica One-Hot Encoding en `mes_inicial`, generando columnas dummy para el mes inicial de compra de cada cliente. Esta transformación ayuda a analizar el comportamiento de compra inicial según el mes.

Para evitar errores en los datos de referencia temporal, se excluyeron las entradas cuya primera compra fue en septiembre de 2019, retornando una serie vacía para estos casos.

Filtrado de registros después de agosto de 2022

Los registros con fecha inicial posterior a agosto de 2022 fueron eliminados para los datos de entrenamiento. Esto asegura que solo se consideren los datos dentro de un periodo de análisis específico.

5.3. Generación de modelos

Eliminación de registros posteriores a agosto 2022

Se eliminaron los registros después del 2022 debido a que el criterio para que un producto sea exitoso o no tiene que comprarlo 5 meses seguidos por lo que en los últimos 6 meses no podemos saber si les fue exitoso o no con certeza ya que se cortan los datos en diciembre 2022.

Variables predictoras y a predecir

A continuación se muestran las variables predictoras que se usaron en el entrenamiento de los modelos junto con su descripción.

La variable a predecir es la columna de "Producto Exitoso"que es una columna binaria con valores de 0 y 1 donde el 1 son los casos donde el producto le fue exitoso al cliente y 0 para caso contrario.

Modelos de Machine Learning

Los modelos básicos de Machine Learning que se utilizaron fueron:

- Regresión Logística
- Árboles de Decisión
- Bosques Aleatorios (Random Forest)
- K-Nearest Neighbors (KNN)
- Naive Bayes
- CatBoost

5.3.1. Entrenamiento y de Machine Learning

Se separaron los datos de entrenamiento y prueba de dos formas:

La primera fue de tal forma que los productos en prueba no estuvieran en entrenamiento y viceversa para simular la entrada de nuevos productos por parte de Arca Continental de los cuales no tenemos datos por parte de los clientes.

La segunda forma de separación de los datos se realizó por fechas, donde se estableció el conjunto de entrenamiento desde enero 2020 hasta diciembre 2021, el conjunto de validación desde enero 2022 hasta junio 2022, y el conjunto de prueba desde julio 2022 hasta diciembre 2022 lo cual establecido por el socio formador Arca Continental.

Para evaluar los modelos se consideraron distintos modelos, hiperparámetros, umbrales de decisión principalmente las métricas de accuracy, recall y f1 score ya que buscamos que los productos que sean

positivos efectivamente el modelo los clasifique como positivos sin importar tanto los falsos positivos.

5.4. Metodología de las Redes Neuronales utilizadas

Para la implementación de las redes neuronales se utilizó el preprocesamiento y análisis previamente realizado.

5.4.1. Selección de Características

Para mejorar la capacidad de predicción y reducir la complejidad del modelo, se aplicó la técnica de *Recursive Feature Elimination* (RFE). Esta técnica permite seleccionar las variables más relevantes del conjunto de datos, eliminando aquellas que no contribuyen de manera significativa al rendimiento de la red. Esta selección asegura que las entradas a la red neuronal contengan solo las características más informativas.

5.4.2. Optimización de Hiperparámetros

Se utilizó Optuna, una biblioteca de optimización de hiperparámetros automatizada, para encontrar la mejor configuración posible de los modelos. Los parámetros optimizados incluyen:

- Número de neuronas por capa.
- Número de capas ocultas.
- Tasa de *dropout*.
- Tipo de optimizador (por ejemplo, Adam, SGD).
- Tasa de aprendizaje.

Esta optimización garantizó que los modelos alcanzaran un rendimiento óptimo en términos de precisión y generalización.

5.4.3. Arquitectura de los Modelos

Se implementaron dos modelos de redes neuronales distintas:

1. Modelo Simple Denso El modelo simple denso es una red neuronal alimentada hacia adelante (feedforward) con la siguiente configuración:

1. **Capa de entrada**: Recibe las características de entrada seleccionadas por RFE.
2. **Capas ocultas**: Consisten en una o más capas densas (totalmente conectadas) con funciones de activación ReLU. El número de neuronas y el número de capas fueron determinados mediante Optuna.
3. **Capa de salida**: Una capa densa con una función de activación sigmoide para la clasificación binaria (producto exitoso o no).

Variables	Descripción
'mes_1', 'mes_2', 'mes_3', 'mes_4', 'mes_5', 'mes_6', 'mes_7', 'mes_8', 'mes_9', 'mes_10', 'mes_11', 'mes_12'	Ventas totales de unibox por mes de cada cliente
'mes_inicial_1', 'mes_inicial_2', 'mes_inicial_3', 'mes_inicial_4', 'mes_inicial_5', 'mes_inicial_6', 'mes_inicial_7', 'mes_inicial_8', 'mes_inicial_9', 'mes_inicial_10', 'mes_inicial_11', 'mes_inicial_12'	Mes en que el cliente compró por primera vez un producto nuevo
'MLSize'	Mililitros del producto
'bolsa', 'lata', 'lata sleek', 'plasticos', 'tetra pack', 'vidrio'	Contenedor del producto
'AGUA_FUNCIONAL', 'AGUA_MINERAL', 'AGUA_PURIFICADA', 'AGUA_SABORIZADA', 'BEBIDA_ALCOHOLICA'	Tipo de producto
'Num_productos_distintos'	Número de productos distintos probados por el cliente

2. Modelo Denso + Convolucional (Denso+Conv) Este modelo combina capas densas y convolucionales para capturar tanto patrones locales como interacciones más globales entre las características:

1. **Capa de entrada**: Similar al modelo denso, recibiendo las características seleccionadas.
2. **Capas convolucionales**: Capas 1D que procesan las secuencias de datos de entrada para detectar patrones espaciales o temporales.
3. **Capas de agrupamiento (Pooling)**: Utilizadas para reducir la dimensionalidad de las salidas convolucionales y mantener los patrones más relevantes.
4. **Capas densas**: Combinadas después de las capas convolucionales para capturar interacciones complejas entre las características.
5. **Capa de salida**: Una capa densa con una función de activación sigmoide para la predicción binaria.

5.4.4. Entrenamiento y Evaluación

Ambos modelos fueron entrenados usando el optimizador Adam y la función de pérdida binaria de entropía cruzada. Se aplicó *early stopping* para evitar el sobreajuste y asegurar la mejor generalización. Las métricas de evaluación incluyeron la precisión, el *recall* y el puntaje F1, siendo más relevante el recall Estas arquitecturas, optimizadas y ajustadas, permitieron obtener modelos con capacidades robustas para predecir la probabilidad de éxito de un producto basado en datos históricos.

6. RESULTADOS

6.1. Modelos de Machine Learning

Los mejores modelos fueron los de Random Forest y CatBoost con resultados similares, en especial los modelos de Random Forest fueron más rápidos que los de CatBoost por lo que los modelos definitivos fueron los siguientes:

Modelo 1: Random Forest 4

- Número de estimadores (n_estimators): 150
- Características máximas (max_features): `sqrt`
- Máxima profundidad (max_depth): 250
- Mínimas muestras para división (min_samples_split): 4
- Peso de clases (class_weight): `balanced`

Modelo 2: Random Forest 5

- Número de estimadores (n_estimators): 300
- Características máximas (max_features): `log2`
- Máxima profundidad (max_depth): 300
- Mínimas muestras en hojas (min_samples_leaf): 2
- Peso de clases (class_weight): `balanced`

Se fueron cambiando los umbrales de decisión para clasificar los productos exitosos y no exitosos para mejorar la métrica de Recall y F1 score sobre todo ya que es más importante que los prodcutos exitosos los clasifique bien pero sin sacrificar el balanceo que ofrece el F1 score.

A continuación se muestran los reportes de clasificación para los mejores modelos con ambas separaciones de datos así como sus matrices de confusión:

Evaluación del mejor modelo separando por productos

Reporte de Clasificación:

Clase	Precisión	Recall	F1-Score	Soporte
0	0.96	0.91	0.94	35082
1	0.66	0.82	0.73	7318
Exactitud	0.90 (42400 muestras)			
Promedio macro	0.81	0.87	0.83	42400
Promedio ponderado	0.91	0.90	0.90	42400

Cuadro 2. Reporte de clasificación para el modelo RF4 con umbral de 0.30

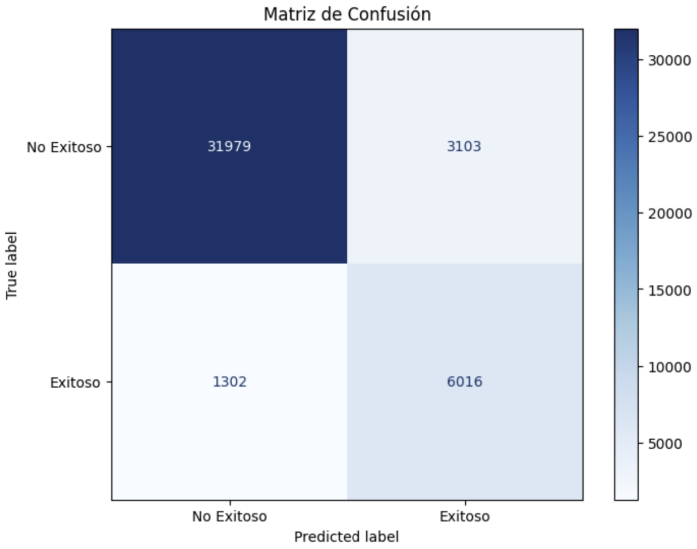

Figura 1. Matriz de confusión del modelo Random Forest 4 con umbral de 0.30

Evaluación del mejor modelo separando por fechas

Reporte de Clasificación:

Clase	Precisión	Recall	F1-Score	Soporte
0	0.98	0.92	0.95	25349
1	0.44	0.75	0.55	2213
Exactitud	0.90 (27562 muestras)			
Promedio macro	0.71	0.83	0.75	27562
Promedio ponderado	0.93	0.90	0.91	27562

Cuadro 3. Reporte de clasificación para el modelo RF4 con umbral de 0.10

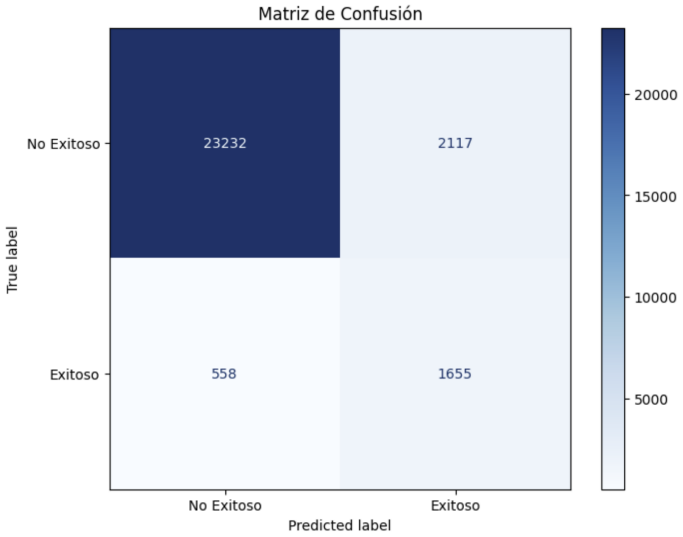

Figura 2. Matriz de confusión del modelo Random Forest 4 con umbral de 0.10

El primer modelo, que separa datos por productos distintos, es más equilibrado al identificar tanto productos exitosos (clase 1) como no exitosos (clase 0). Aunque su precisión para productos exitosos es moderada (66 %), logra un buen recall (82 %), lo que significa que captura la mayoría de los productos exitosos. Esto lo hace adecuado para aplicaciones donde identificar productos exitosos sea prioritario.

Por otro lado, el segundo modelo, que separa datos por fechas históricas, es excelente para detectar productos no exitosos (precisión del 98 % y recall del 92 %), pero tiene dificultades para predecir correctamente productos exitosos, con una precisión baja (44 %). Esto podría deberse a cambios en las tendencias del mercado no reflejados en los datos históricos.

6.2. Modelos de Deep Learning

Para los modelos de Deep Learning se encontraron los siguientes mejores parámetros para cada red:

- **Parámetros del modelo denso simple:** {n_layers: 4, n_units: 160, dropout_rate: 0.11171220212777039, learning_rate: 0.0015001879760607399}
  - **n_layers:** Número de capas densas en la red. Un valor de 4 indica una red con 4 capas ocultas.
  - **n_units:** Número de unidades en cada capa. En este caso, 160 unidades por capa proporcionan la capacidad de modelado.
  - **dropout_rate:** Tasa de abandono aplicada para prevenir el sobreajuste. Un valor de 0.111 indica que se omite el 11.1 % de las neuronas durante el entrenamiento.
  - **learning_rate:** Tasa de aprendizaje que controla el tamaño de los pasos de actualización de los pesos. Un valor de 0.0015 proporciona un ajuste más lento y cuidadoso.
- **Parámetros del modelo convolucional + denso:** {n_conv_filters: 48, conv_kernel_size: 5, n_dense_units: 64, dropout_rate: 0.25577715716603544, learning_rate: 0.009670678620590806}
  - **n_conv_filters:** Número de filtros utilizados en la capa convolucional. Un valor de 48 define la cantidad de detectores de características.
  - **conv_kernel_size:** Tamaño del kernel de la convolución. Un tamaño de 5 significa que se usan filtros de 5x5.
  - **n_dense_units:** Número de unidades en la capa densa posterior. Aquí, 64 unidades se utilizan para la parte densa de la red.
  - **dropout_rate:** Tasa de abandono de 0.255, lo que implica que se omite el 25.5 % de las neuronas.
  - **learning_rate:** Tasa de aprendizaje de 0.0096, que permite una actualización más agresiva de los pesos.
- **Parámetros del modelo LSTM:** {n_lstm_units: 32, n_dense_units: 128, dropout_rate: 0.2914766948736778, learning_rate: 0.004294165409294777}

Clase	Precisión	Recall	F1-Score	Soporte
0	0.94	1.00	0.97	29176
1	0.84	0.24	0.37	2427
Exactitud		0.94 (31603 muestras)		
Promedio macro	0.89	0.62	0.67	31603
Promedio ponderado	0.93	0.94	0.92	31603

Cuadro 4. Reporte de clasificación para el modelo convolucional + denso.

- **n_lstm_units:** Número de unidades en la capa LSTM. Un valor de 32 indica la capacidad de la memoria de la red recurrente.
- **n_dense_units:** Número de unidades en la capa densa que sigue a la LSTM, en este caso, 128.
- **dropout_rate:** Tasa de abandono de 0.291, lo que implica que se omite el 29.1 % de las neuronas durante el entrenamiento.
- **learning_rate:** Tasa de aprendizaje de 0.0042, ajustando los pesos a un ritmo moderado.

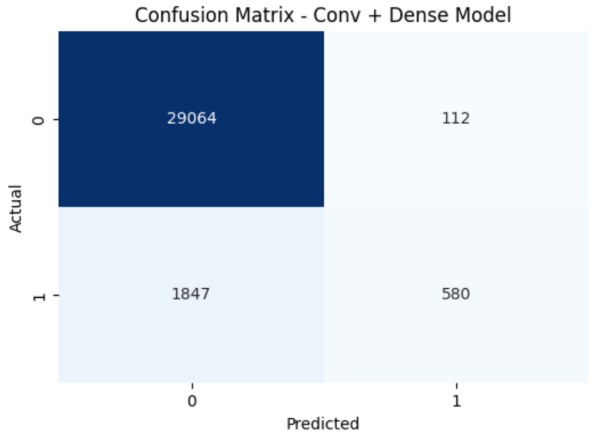

Figura 3. Matriz de confusión del modelo convolucional + denso

Sin embargo, el modelo LSTM y el denso simple no lograron aprender de manera adecuada, por lo que el único modelo con resultados fue el convolucional + denso:

6.3. Métricas del Modelo Conv + Denso

- **F1 Score:** 0.67. Esta métrica representa la media armónica entre la precisión y la exhaustividad, indicando un rendimiento global bajo en la clasificación.
- **Precisión:** 0.89. Muestra qué tan a menudo el modelo predijo correctamente la clase positiva de entre todas sus predicciones positivas.
- **Recall (Exhaustividad):** 0.62. Indica qué porcentaje de los verdaderos positivos fueron detectados por el modelo, sugiriendo que muchas muestras positivas no fueron capturadas.
- **Exactitud (Accuracy):** 0.94. Muestra el porcentaje de predicciones correctas sobre todas las muestras.

Reporte de Clasificación:

La matriz de confusión presentada muestra el rendimiento del modelo, donde las celdas de la diagonal principal (29064 y 580) representan las predicciones correctas de los productos no exitosos y exitosos (0 y 1, respectivamente). Las celdas fuera de la diagonal (112 y 1847) indican las predicciones erróneas, donde 112 muestras de los productos no exitosos fueron incorrectamente clasificados como exitosos y 1847 muestras de los productos exitosos fueron clasificadas incorrectamente como no exitosos. Como podemos observar, la red neuronal no logra clasificar correctamente los productos exitosos, por lo que el modelo no es bueno para clasificar productos exitosos. En el cuadro 4 se muestra el reporte de clasificación del modelo convolucional + denso, donde se muestran resultados no tan favorecedores para la clase 1.

6.4. Clientes que adoptaran un producto

La función `calcular_probabilidad_indicador` predice la probabilidad de que un cliente adopte un producto o indicador específico con base en ciertas características. El enfoque combina preprocesamiento de datos, escalado y modelado predictivo. Los pasos principales son:

1. **Definición de opciones de entrada:** Se definen opciones válidas para *empaque*, *bebida* y *mes*, las cuales se convierten en variables binarias mediante técnicas de codificación como *One-Hot Encoding*.
2. **Construcción del conjunto de datos:** Se crean columnas para las variables seleccionadas y se combinan con datos específicos de los clientes (proporcionados en el DataFrame `clientes_unicos`). Los datos son replicados y concatenados para asociar cada cliente con el registro del producto o indicador.
3. **Escalado de características:** Las variables del conjunto de entrenamiento (`X_train`) se escalan utilizando `MinMaxScaler`, normalizando los datos y asegurando que estén en el rango adecuado para el modelo.
4. **Predicción con Random Forest:** Se ajusta un modelo de *Random Forest* (`rf_model`) utilizando los datos de entrenamiento escalados y las etiquetas correspondientes (`y_train`). A continuación, se calcula la probabilidad de adopción para cada cliente en el conjunto replicado y escalado.
5. **Selección y ordenamiento:** Los clientes se ordenan por su probabilidad de adoptar el producto, devolviendo los 100 clientes con mayor probabilidad o la cantidad de clientes que se desee visualizar.

Esta función se pensó principalmente para aplicarla en escenarios donde se desea predecir la adopción de productos por parte de clientes con base en características de mercado, hábitos de consumo y variables asociadas al producto. Con ello, Arca Continental puede:

1. **Evaluación de nuevos productos:** Estimar la probabilidad de adopción de productos lanzados recientemente.
2. **Segmentación de clientes:** Identificar a los clientes con mayor probabilidad de adoptar un producto para orientar estrategias de marketing.
3. **Optimización de estrategias:** Mejorar la toma de decisiones comerciales mediante predicciones basadas en datos.

6.4.1. Ejemplo de aplicación

El DataFrame incluye dos columnas principales: **CustomerId y Probabilidad_Indicador.** Los clientes están ordenados de mayor a menor probabilidad de adopción.

Resultados específicos de una implememtación:

El cliente con el ID 510655461 tiene la probabilidad más alta (0.670740) de adoptar el producto para las características dadas: MLSize=355, empaque='lata', bebida='colas light' y mes='septiembre'.

Al comparar a los clientes más probables de adoptar este producto con los clientes que compraron en los últimos seis meses de información, detectamos que para este específico caso, coincidieron 21 clientes. Esto le puede proveer a Arca Continental información para lograr la optimización de campañas dirigidas al segmento de clientes más receptivo.

Referencias

[1] Autor(es). «Título del artículo en arXiv». En: *arXiv preprint arXiv:2401.12223* (2024). URL: https://arxiv.org/abs/2401.12223.

[2] Juan Sossa. *Inteligencia Artificial*. 2002. URL: https://ru.iibi.unam.mx/jspui/bitstream/IIBI_UNAM/89/1/01_inteligencia_artificial_juan_sossa.pdf.

[3] Autor(es). «Título del artículo en RIITE». En: *Revista Interuniversitaria de Investigación en Tecnología Educativa* N/A (2023). URL: https://revistas.um.es/riite/article/view/584501.

[4] Gabriela Jumbo Quichimbo. *Aplicaciones de Inteligencia Artificial en la Educación*. 2020. URL: https://reunir.unir.net/bitstream/handle/123456789/8166/JUMBO%20QUICHIMBO%2C%20GABRIELA.pdf.

[5] PwC. *Sectores en los que la Inteligencia Artificial tendrá mayor impacto*. 2024. URL: https://www.pwc.com/co/es/pwc-insights/sectores-tendra-impacto-ia.html.

[6] Grupo Atico34. *10 casos reales de vulneración del derecho a la privacidad*. 2023. URL: https://protecciondatos-lopd.com/empresas/casos-privacidad-digital/.

[7] European Council. *El Consejo aprueba la Ley de Gobernanza de Datos*. 2022. URL: https://www.consilium.europa.eu/es/press/press-releases/2022/05/16/le-conseil-approuve-l-acte-sur-la-gouvernance-des-donnees/.

[8] Business Insider. *Vodafone, CaixaBank, BBVA, Mercadona o Telefónica: Protección de Datos*. 2021. URL: https://www.businessinsider.es/9-empresas-espanolas-multadas-proteccion-datos-2021-981745.

[9] El País. *Las grandes compañías buscan cómo asegurar un desarrollo ético y legal de la inteligencia artificial*. 2023. URL: https://elpais.com/tecnologia/2023-12-20/las-grandes-companias-desarrollan-plataformas-para-asegurar-un-desarrollo-etico-y-legal-de-la-inteligencia-artificial.html.

[10] Growfik. *IA en la predicción de tendencias de mercado: anticipando cambios y oportunidades*. (n.d.) URL: https://www.growfik.com/blog/ia-en-la-prediccion-de-tendencias-de-mercado-anticipando-cambios-y-oportunidades.

[11] Customer Target. *La IA en el sector retail: la experiencia del cliente a un nuevo nivel*. (n.d.) URL: https://www.customertarget.com/la-ia-en-el-sector-retail-la-experiencia-del-cliente-a-un-nuevo-nivel/.

[12] Inmediatum. *Cómo se utiliza en el 2021 la IA en retail*. 2021. URL: https://inmediatum.com/blog/estrategia/como-se-utiliza-en-el-2021-la-ia-en-retail/.

[13] IBM. *Impacto ético de la IA*. (n.d.) URL: https://www.ibm.com/es-es/impact/ai-ethics.

[14] DataCamp. *AI regulation: what the EU's new proposal means for the future of artificial intelligence*. (n.d.) URL: https://www.datacamp.com/es/blog/ai-regulation.

[15] Contributors to blog.maestriasydiplomados.tec.mx. *El impacto de la Inteligencia Artificial en la Actualidad*. Tecnológico de Monterrey, Blog Maestrías y Diplomados. 2023. URL: https://blog.maestriasydiplomados.tec.mx/el-impacto-de-la-inteligencia-artificial-en-la-actualidad.

[16] Itop. *Scikit-learn*. 2024. URL: https://www.itop.es/soluciones-tecnologicas/business-analytics-business-intelligence/scikit-learn.html.

[17] Isha Bansal. *Python catboost module: A Brief Introduction to CatBoost Classifier*. 2021. URL: https://www.askpython.com/python-modules/catboost-module.

[18] Contributors to emprendedores.es. *Sectores impactados por la Inteligencia Artificial*. Emprendedores Revista Digital. 2023. URL: https://emprendedores.es/inteligencia-artificial/sectores-impacto-inteligencia-artificial/.

[19] International Monetary Fund. *AI Will Transform the Global Economy: Let's Make Sure It Benefits Humanity*. IMF Blog. 2024. URL: https://www.imf.org/es/Blogs/Articles/2024/01/14/ai-will-transform-the-global-economy-lets-make-sure-it-benefits-humanity.

[20] Contributors to osha.europa.eu. *Impacto de la Inteligencia Artificial en la Seguridad y Salud Ocupacional*. OSHA European Agency. 2024. URL: https://osha.europa.eu/es/publications/impact-artificial-intelligence-occupational-safety-and-health.

[21] Contributors to zendesk.com.mx. *Cómo afecta la Inteligencia Artificial en la Economía*. Zendesk Blog. 2023. URL: https://www.zendesk.com.mx/blog/como-afecta-la-inteligencia-artificial-en-la-economia/.

[22] Emprendedores.es. *Sectores en los que la Inteligencia Artificial tendrá mayor impacto*. 2024. URL: https://emprendedores.es/inteligencia-artificial/sectores-impacto-inteligencia-artificial/.

[23] ISDI. *El Impacto de la IA en Diferentes Sectores*. 2024. URL: https://www.isdi.education/es/blog/impacto-de-la-ia.