Proyecto Final Modelos y Simulación II

Camilo Benavides Ramirez, Nicolás Carmona Cardona, Brayan Santiago Ramirez Silva

***Resumen*—Este trabajo evalúa distintos modelos de clasificación para predecir la intención de compra utilizando el dataset *Online Shoppers Purchasing Intention*. Se implementó un pipeline con preprocesamiento, SMOTE y reducción de dimensión. El MLP y el árbol de decisión optimizados obtuvieron los mejores resultados. PCA afectó negativamente el desempeño, mientras que UMAP mantuvo resultados estables. Además, los hallazgos son coherentes con el estado del arte para métodos tradicionales.**

***Palabras Claves*—Machine Learning, Intención de Compra, SMOTE, Reducción de Dimensión, PCA, UMAP.**

I. Introducción

El crecimiento del comercio electrónico ha intensificado la necesidad de comprender el comportamiento del consumidor en línea para optimizar la experiencia de usuario y mejorar las tasas de conversión. Un desafío crítico en este dominio es la predicción de la intención de compra, fundamental para reducir el abandono de carritos. El Aprendizaje Automático (Machine Learning) ofrece herramientas robustas para abordar este problema, permitiendo la identificación de patrones complejos a partir de los datos de navegación del usuario (*clickstream*). Este proyecto se enfoca en aplicar y evaluar diversas técnicas de clasificación supervisada sobre el conjunto de datos "Online Shoppers Purchasing Intention" con el objetivo de desarrollar un modelo predictivo preciso y eficiente.

II. Descripción del problema

El comercio electrónico ha crecido exponencialmente en los últimos años, convirtiéndose en un canal esencial para las empresas, pero enfrentando el desafío del alto abandono de compra por parte de los usuarios. Comprender y anticipar la intención de compra es clave para mejorar la experiencia del cliente y aumentar las tasas de conversión. En este contexto, una solución basada en Machine Learning (ML) resulta altamente útil, ya que permite analizar grandes volúmenes de datos de navegación, identificar patrones complejos en el comportamiento del usuario y predecir en tiempo real la probabilidad de que una sesión culmine en una compra, optimizando así las estrategias comerciales y la toma de decisiones.

El análisis exploratorio del dataset utilizado mostró que no existen valores faltantes en ninguna de las 18 columnas (17 características + 1 objetivo), lo que elimina la necesidad de aplicar estrategias de imputación de datos y simplifica el preprocesamiento. La base de datos está compuesta por variables numéricas continuas y discretas, categóricas y binarias, las cuales representan diferentes aspectos del comportamiento de los usuarios dentro de un entorno de comercio electrónico. A continuación, se presenta un resumen de las variables que componen el conjunto de datos.

TABLE I
Resumen de las Variables del Dataset

Tipo de Variable	Nombre	Descripción
Numéricas Continuas	ProductRelated_Duration, BounceRates, ExitRates	Métricas de tiempo, tasas o valores promedio (ej. tiempo en páginas, tasas de rebote/salida, valor de página).
Numéricas Discretas	Administrative, Administrative_Duration, Informational, Informational_Duration, ProductRelated, PageValues, SpecialDay, OperatingSystems, Browser, Region, TrafficType	Conteo de interacciones o códigos identificadores (ej. número de páginas visitadas, tipo de OS, navegador, región, tráfico).
Categóricas	Month, VisitorType	Variables con categorías fijas sin orden (ej. mes de la visita, tipo de visitante).
Binarias	Weekend, Revenue	Variables con dos posibles estados (ej. fin de semana sí/no, compra realizada sí/no).

Para el procesamiento de las variables categóricas y numéricas, se definió la siguiente estrategia de codificación:

Variables Categóricas Ordinales (Month): Se aplicará *Label Encoding* o *One-Hot Encoding* dependiendo del modelo, dado que los meses presentan cierta ordenación temporal.

Variables Categóricas Nominales (VisitorType): Se empleará *One-Hot Encoding* para crear variables dummy, evitando imponer un orden artificial entre las categorías.

Variables Numéricas Categóricas (OperatingSystems, Browser, Region, TrafficType): Aunque se expresan como números, representan categorías. Se aplicará *One-Hot Encoding* o *Target Encoding* según el rendimiento del modelo, con el fin de capturar relaciones no lineales.

Variables Binarias (Weekend): Se convertirán de booleanas a formato numérico (0/1) para compatibilidad con los algoritmos de aprendizaje automático.

Variables Numéricas Continuas y Discretas: Se aplicará normalización o estandarización dependiendo de los requerimientos de los modelos seleccionados, especialmente para aquellos sensibles a las escalas de las variables, como SVM, KNN y redes neuronales.

El anáslisis del dataset muestra variables predictoras correctamente estructuradas y sin valores faltantes, lo que proporciona una base confiable para el modelado. Sin embargo, se identificó un desbalance significativo en la variable objetivo (Revenue), ya que solo el 15% de las sesiones culminan en una compra, esto exige la aplicación de técnicas de balanceo de clases, como **SMOTE**, y el uso de métricas de evaluación adecuadas como un **F1 score** o **Balanced Error Rate**, en lugar de la exactitud (accuracy), la cual puede resultar engañosa en escenarios desbalanceados.

Asimismo, las variables numéricas presentan distribuciones sesgadas y valores atípicos, especialmente en las métricas de duración, mientras que la matriz de correlación sugiere posibles problemas de multicolinealidad entre variables relacionadas. Con estos hallazgos se tiene la necesidad de aplicar transformaciones y normalizaciones adecuadas, manejar los valores atípicos y realizar una selección cuidadosa de características, priorizando aquellas con mayor capacidad explicativa para optimizar la precisión. En consecuencia, la aproximación de ML propuesta se basará en un paradigma de aprendizaje supervisado, dado que la variable objetivo (*Revenue*) es binaria y representa la conversión o no del usuario. Específicamente, se trata de un problema de clasificación binaria. Este enfoque permitirá entrenar modelos capaces de predecir la probabilidad de compra a partir de los patrones de navegación observados.

III. Estado del arte

La predicción de la intención de compra en línea ha sido abordada desde diversas perspectivas dentro del aprendizaje supervisado. A continuación, se revisan cinco estudios relevantes que emplean diferentes técnicas y metodologías.

A. Sakar et al. (2018) [1]

Este trabajo propone un sistema dual para predecir en tiempo real tanto la intención de compra como el abandono del sitio. Para la intención de compra, se enmarcan en una **clasificación binaria supervisada**, comparando modelos como *Random Forest (RF)*, *Support Vector Machines (SVM)* y *Multilayer Perceptron (MLP)*. Para el abandono, utilizan una red recurrente *Long Short-Term Memory (LSTM)*. La validación se realizó mediante una división *hold-out* (70/30) repetida 100 veces, evaluando con métricas como *Accuracy* y *F1-Score*. El mejor resultado fue obtenido por un **MLP** con sobremuestreo y selección de características, alcanzando un ***Accuracy*** **del 87.94%** y un **F1-Score de 0.87**.

B. Requena et al. (2020) [2]

Este estudio compara dos estrategias para la predicción: una basada en características diseñadas manualmente (*hand-crafted*) y otra en aprendizaje profundo. En el primer enfoque, se utilizaron clasificadores como *XGBoost* y redes neuronales (*NN*) sobre características como k-grams y motivos de grafos de visibilidad horizontal (HVGm). Los HGVm corresponden a patrones locales que se obtienen convirtiendo la serie temporal en un grafo mediante el criterio de visibilidad horizontal. El segundo empleó arquitecturas como *LSTM* (S2L). Utilizando una validación *hold-out* (80/20), el modelo **S2L** destacó en datos balanceados con un **F1-Score del 91.03%**. Sin embargo, en un escenario desbalanceado, una **NN** con características manuales fue superior, logrando un **AUC de 94.75%**.

C. Tanvir et al. (2023) [3]

Este trabajo se centra en optimizar la predicción mediante *Gradient Boosting*. Se compararon modelos como *SVM*, *RF*, *MLP*, *Decision Tree (DT)* y *XGBoost*, manejando el desbalance de clases con *SMOTE*. La validación se realizó con una división *hold-out* (70/30). El rendimiento se midió con un conjunto robusto de métricas, incluyendo el **Coeficiente de Correlación de Matthews (MCC)**.

El modelo que combina **selección de características, SMOTE y XGBoost** obtuvo los mejores resultados, alcanzando una ***accuracy*** **del 90.65%** y un **auROC de 0.937**.

D. Gomes et al. (2022) [4]

Este estudio introduce un enfoque novedoso que representa el comportamiento del cliente mediante *embeddings*, eliminando la ingeniería de características manual. Primero, se generan representaciones vectoriales de los eventos de clic ("touch-points") utilizando un modelo inspirado en *skip-gram*. Luego, estas secuencias se usan para entrenar varios clasificadores, incluyendo **LSTM**. Con una validación *hold-out* (85/15), la combinación de ***embeddings*** **y un predictor LSTM** demostró ser la más efectiva, superando a los enfoques de línea base. En el dataset *openCDP*, por ejemplo, alcanzó un **F1-Score de 0.872** y un **AUC de 0.931**.

E. Satu & Islam (2023) [5]

Este estudio propone un modelo de aprendizaje automático que explora exhaustivamente el efecto de la ingeniería de características. Se aplicaron múltiples transformaciones de datos (*Min-Max, Z-Score, Square root*), balanceo con *SMOTE*, y varios métodos de selección de características. Se evaluaron 14 clasificadores diferentes utilizando **validación cruzada de 10 pliegues**. El rendimiento se midió con *Accuracy, F-Score* y *AUROC*. Se encontró que **Random Forest (RF)** es el clasificador más estable y con mejor rendimiento, logrando una ***accuracy*** **máxima de 92.39%** y un **F-Score de 0.924** sobre un subconjunto de datos transformados.

IV. Entrenamiento y evaluación de los modelos

A. Configuración Experimental

1) ***Metodología de Validación****:* Los datos se dividieron de forma estratificada, asignando el 80% de los datos (9,864 muestras) para entrenamiento y el 20% restante (2,466 muestras) para prueba. La estratificación garantizó que ambos

conjuntos conservaran la proporción original de clases (84.5% no compra y 15.5% compra). Esta estrategia resulta adecuada frente al desbalanceo del conjunto de datos, ya que permite mantener la representación de la clase minoritaria en ambas particiones.

Antes de la validación se aplicó un proceso de preprocesamiento. Las variables categóricas nominales (Month, VisitorType, OS, Browser y Region) fueron transformadas mediante *One-Hot Encoding*, mientras que la variable TrafficType se codificó mediante *Target Encoding*. Para las variables numéricas se utilizó *RobustScaler*, seleccionado debido a la presencia de valores atípicos identificados en el análisis exploratorio. Todas estas transformaciones fueron ajustados únicamente con el conjunto de entrenamiento, evitando así cualquier forma de *data leakage*.

Debido al desbalance de clases, se aplicó la técnica SMOTE exclusivamente sobre el conjunto de entrenamiento y después de la división inicial. Se evaluaron tres niveles de sobremuestreo: 5%, 10% y 15% de incremento sobre el tamaño total del conjunto de entrenamiento. El conjunto de prueba se mantuvo sin modificaciones para preservar una evaluación objetiva del desempeño.

Cada modelo fue entrenado en cuatro configuraciones: una versión *baseline* sin sobremuestreo y tres variantes correspondientes a los niveles de SMOTE evaluados. Aproximadamente, estas configuraciones generaron 2,019 muestras minoritarias (SMOTE 5%), 2,512 muestras (SMOTE 10%) y 3,005 muestras (SMOTE 15%). En total, se compararon 32 configuraciones experimentales (8 modelos $\times$ 4 niveles de SMOTE), permitiendo identificar la combinación más adecuada entre algoritmo y estrategia de balanceo, además, se uso F1-Score como métrica de optimización debido a su robustez ante desbalanceo.

*2) **Configuración de Hiperparámetros**:* La optimización de hiperparámetros se llevó a cabo mediante *GridSearchCV* para los ocho modelos evaluados. Para cada algoritmo se utilizó la configuración que había mostrado mejor desempeño en la fase previa. La búsqueda se realizó mediante validación cruzada estratificada de 5 *folds*, con el fin de mantener la proporción de clases en cada partición y obtener métricas más estables.

El criterio de optimización seleccionado fue el F1-Score, dado que combina precisión y *recall* y es más adecuado en escenarios con desbalanceo de clases. Las mallas de hiperparámetros incluyeron rangos amplios para parámetros de regularización, diversas profundidades y tamaños de ensamble, topologías de redes neuronales de distinta complejidad y múltiples combinaciones de kernels para SVM.

En total se evaluaron 2,410 combinaciones únicas de hiperparámetros, distribuidas entre los ocho modelos, permitiendo identificar configuraciones robustas y comparables para el análisis final. Estos valores pueden observarse en detalle en la Tabla II.

TABLE II
HIPERPARÁMETROS EVALUADOS MEDIANTE GRIDSEARCHCV

Modelo	Tipo	Hiperparámetros	Malla de Valores	Comb.
Logistic Regression	Paramétrico	C, penalty, solver	[0.01, 0.1, 1, 10, 100]; [l2]; [lbfgs, liblinear]	10
K-Nearest Neighbors	No paramétrico	n_neighbors, weights, metric, p	[3, 5, 7, 9, 11, 15]; [uniform, distance]; [euclidean, manhattan, minkowski]; [1, 2]	72
Decision Tree	Árbol individual	max_depth, min_samples_split, min_samples_leaf, criterion	[3, 5, 10, 15, 20, None]; [2, 5, 10]; [1, 2, 4]; [gini, entropy]	108
Random Forest	Ensamble de árboles	n_estimators, max_depth, min_samples_split, min_samples_leaf, max_features	[50, 100, 200]; [10, 20, 30, None]; [2, 5, 10]; [1, 2, 4]; [sqrt, log2]	144
XGBoost	Gradient Boosting	n_estimators, learning_rate, max_depth, min_child_weight, subsample, colsample_bytree	[50, 100, 200]; [0.01, 0.05, 0.1, 0.2]; [3, 5, 7, 9]; [1, 3, 5]; [0.8, 0.9, 1.0]; [0.8, 0.9, 1.0]	1728
CatBoost	Gradient Boosting	iterations, learning_rate, depth, l2_leaf_reg	[50, 100, 200]; [0.01, 0.05, 0.1, 0.2]; [4, 6, 8, 10]; [1, 3, 5, 7]	256
SVM	SVM clásico	C, gamma, kernel	[1, 10]; [scale, 0.01]; [rbf]	4
MLP	Red Neuronal	hidden_layer_sizes, activation, alpha, learning_rate	[(50,), (100,), (100,50), (100,100), (150,100,50)]; [relu, tanh]; [0.0001, 0.001, 0.01]; [constant, adaptive]	60

Debido a la alta complejidad computacional de SVM se realizó una búsqueda enfocada únicamente en los hiperparámetros más influyentes. Se evaluó el *kernel* RBF, adecuado para capturar relaciones no lineales, junto con valores intermedios de *C* y *gamma* basados en recomendaciones de la literatura para problemas de clasificación binaria.

*3) **Métricas de Evaluación**:* Para evaluar el sistema de predicción de intención de compra se utilizaron métricas complementarias que permiten analizar distintos aspectos del desempeño en un escenario de clasificación binaria con desbalance de clases. Las métricas seleccionadas fueron las siguientes:

a) F1-Score (Métrica principal): Se empleó como métrica principal ya que combina la precisión y el *recall*, ofreciendo un equilibrio adecuado entre ambas. Es muy pertinente en contextos desbalanceados, donde la *accuracy* puede ser muy engañosa. En nuestro caso, un buen F1-Score refleja la capacidad del modelo para identificar compradores potenciales sin incurrir en un número excesivo de falsos positivos.

b) Recall (Sensibilidad): Indica la proporción de compradores reales que el modelo logra detectar. Su relevancia esta en que un falso negativo representa una oportunidad comercial perdida. Un valor alto de *recall* garantiza que la mayoría de los usuarios con intención de compra sean correctamente identificados.

c) Precision: Mide la proporción de predicciones positivas que son correctas, es una métrica importante para la eficiencia de campañas de marketing, ya que un exceso de falsos positivos implica costos innecesarios. Una precisión elevada optimiza el uso de recursos en acciones comerciales.

d) ROC-AUC: Evalúa la capacidad del modelo para discriminar entre ambas clases a lo largo de diferentes umbrales de decisión. Un valor cercano a 1.0 indica una excelente capacidad discriminativa. Es independiente del umbral y buena ante el desbalanceo, lo que la convierte en

TABLE III
RESULTADOS DE OPTIMIZACIÓN DE HIPERPARÁMETROS CON GRIDSEARCHCV

Modelo	Config. SMOTE	F1 CV (μ)	F1 CV (σ)	IC 95%	F1 Test	Prec. Test	Recall Test	ROC-AUC	Tiempo (min)
MLP	SMOTE 15%	0.7795	0.0229	[0.7347, 0.8243]	0.6690	0.6135	0.7356	0.9144	1.83
Decision Tree	SMOTE 15%	0.7829	0.0362	[0.7119, 0.8539]	0.6650	0.6342	0.6990	0.9149	0.17
XGBoost	Baseline	0.6730	0.0229	[0.6281, 0.7178]	0.6524	0.7192	0.5969	0.9296	4.69
Random Forest	SMOTE 10%	0.7768	0.0807	[0.6186, 0.9350]	0.6464	0.7240	0.5838	0.9192	3.14
CatBoost	SMOTE 15%	0.8161	0.0839	[0.6517, 0.9806]	0.6313	0.6766	0.5916	0.9204	14.76
KNN	SMOTE 15%	0.8300	0.0259	[0.7792, 0.8808]	0.6215	0.5775	0.6728	0.8549	1.11
SVM	SMOTE 15%	0.7823	0.0141	[0.7548, 0.8099]	0.6110	0.6094	0.6126	0.8832	1.39
Logistic Regression	SMOTE 15%	0.6411	0.0205	[0.6009, 0.6814]	0.5958	0.6958	0.5209	0.8881	0.27

una métrica adecuada para comparar modelos de forma objetiva.

e) Accuracy: Aunque su capacidad para informar es limitada en contextos desbalanceados, se incluye como referencia general del rendimiento global del modelo. Sin embargo, no se utilizó como métrica primaria al estar influenciada por la clase mayoritaria.

f) Matriz de Confusión: Nos da un desglose detallado de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. Permite analizar el tipo de errores más frecuentes y comprender las implicaciones prácticas del modelo en el negocio.

Estrategia de Evaluación: Las métricas se calcularon tanto en entrenamiento como en prueba. Durante el entrenamiento se utilizó validación cruzada estratificada de 5 *folds*, mientras que el conjunto de prueba proporcionó las métricas finales. También se estimaron intervalos de confianza a partir de la variabilidad entre *folds*.

La selección del modelo óptimo priorizó el F1-Score en el conjunto de prueba, complementado con ROC-AUC para evaluar capacidad discriminativa y con *recall* para asegurar la detección de compradores potenciales.

B. Resultados del Entrenamiento de Modelos

Esta sección presenta los resultados de la experimentación realizada sobre los ocho algoritmos evaluados. Para la optimización de hiperparámetros se empleó GridSearchCV con validación cruzada estratificada de 5 *folds*, utilizando F1-Score como métrica principal debido al desbalanceo de clases presente en el dataset.

*1) **Desempeño Global de los Modelos Optimizados**:* La Tabla III presenta un resumen comparativo del desempeño de los ocho modelos después de la optimización de hiperparámetros. Se reportan las métricas obtenidas tanto en validación cruzada como en el conjunto de prueba, junto con intervalos de confianza al 95% calculados a partir de la desviación estándar de los 5 *folds* de validación cruzada.

Los resultados muestran que MLP obtuvo el mejor desempeño con un F1-Score de 0.6690 en el conjunto de prueba, seguido por Decision Tree con 0.6650 y XGBoost con 0.6523.

La Figura 1 ilustra la comparación entre el F1-Score obtenido en validación cruzada y en el conjunto de prueba, incluyendo barras de error que representan los intervalos de confianza al 95%. Esta visualización permite evaluar la capacidad de generalización de cada modelo.

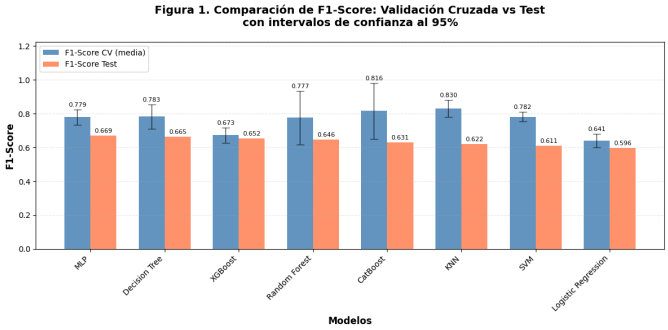

Fig. 1. Comparación entre el F1-Score de validación cruzada y de prueba, con intervalos de confianza al 95%.

*2) **Análisis del Trade-off Precision-Recall**:* En problemas de clasificación desbalanceados como la predicción de intención de compra, es fundamental analizar el balance entre precision y recall. La Figura 2 presenta este análisis para todos los modelos evaluados.

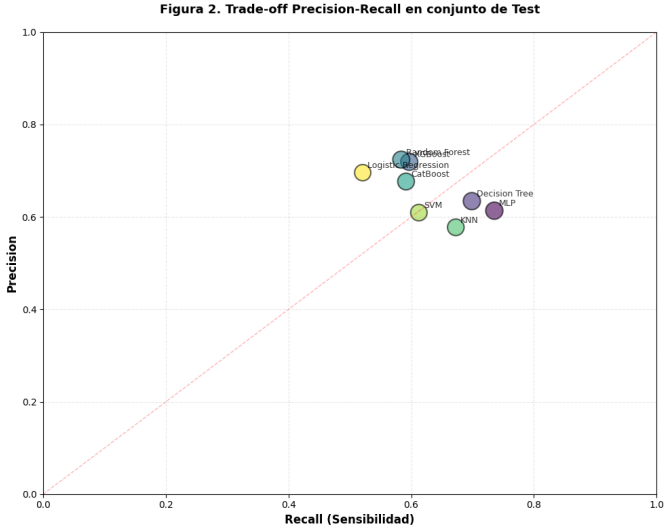

Fig. 2. Trade-off entre Precision y Recall en el conjunto de prueba. Los modelos más a la esquina superior derecha representan el balance óptimo.

Los resultados muestran que XGBoost y Random Forest

logran alta precision (¿0.72) pero con recall moderado (¡0.60), priorizando la reducción de falsos positivos. Por otro lado, MLP y Decision Tree presentan mejor balance con precision $\approx$ 0.61-0.63 y recall $\approx$ 0.70-0.74, maximizando la detección de compradores reales. Esta información es muy importante según las prioridades del negocio: detectar todos los posibles compradores (alto recall) versus minimizar recursos en usuarios sin intención de compra (alta precision).

*3) **Analisis del mejor modelo**:* Para evaluar el comportamiento del modelo MLP en términos de errores de clasificación, a continuación se presenta su matriz de confusión en la Figura 3.

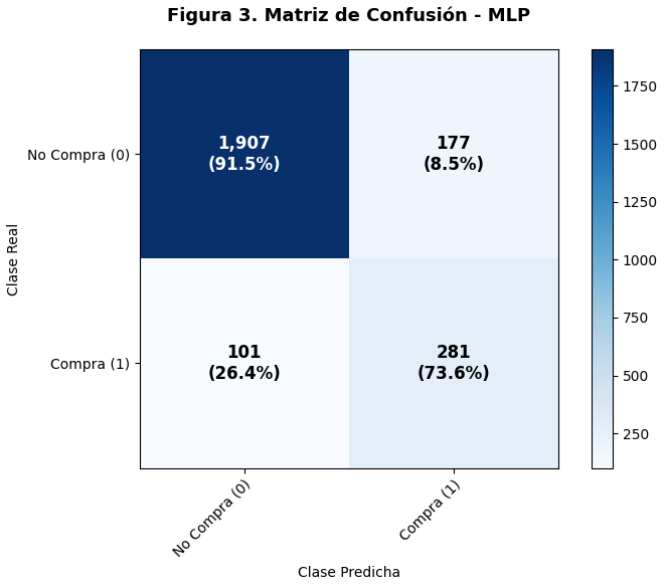

Fig. 3. Matriz de confusión del modelo MLP optimizado en el conjunto de prueba. Los valores indican número de observaciones y porcentaje por fila.

El modelo clasificó correctamente 1,907 usuarios sin intención de compra (especificidad=91.5%) y 281 compradores reales (sensibilidad=73.6%). Los 101 falsos negativos representan compradores no detectados, lo cual implica oportunidades de venta perdidas. Por otro lado, los 177 falsos positivos corresponden a recursos desperdiciados en usuarios sin intención real de compra. En nuestro contexto, este balance favorece la detección de compradores potenciales sobre la minimización de falsos positivos.

*4) **Impacto de las Técnicas Aplicadas**:* La comparación entre las tres fases experimentales muestra que la aplicación de SMOTE generó una mejora promedio del 7.40% en F1-Score, mientras que la optimización de hiperparámetros añadió un incremento adicional del 2.11%, alcanzando una mejora total media del 9.66%. El modelo con mayor ganancia global fue Decision Tree (+23.20%), mientras que CatBoost presentó una ligera disminución en el rendimiento (-2.78%). En general, los resultados evidencian que tanto el sobremuestreo como la optimización contribuyen positivamente en la mayoría de los modelos, aunque su impacto no es uniforme.

En conclusión, los resultados muestran que las dos configuraciones con mejor desempeño fueron MLP y Decision Tree, ambos combinados con SMOTE al 15% y posteriormente optimizados mediante GridSearchCV. Aunque MLP obtuvo el mayor F1-Score en el conjunto de prueba, la diferencia respecto a Decision Tree fue mínima, por lo que ambos modelos pueden considerarse alternativas competitivas.

V. Reducción de Dimensión

A. Análisis individual de variables

Para identificar las características con mayor capacidad discriminativa y las candidatas a eliminación, se realizó un análisis utilizando varias medidas estadísticas. Este análisis permite comprender el poder predictivo de cada variable antes de aplicar técnicas de reducción de dimensión.

*1) **Metodología**:* Se emplearon tres métodos complementarios para evaluar la capacidad discriminativa de las 43 características del conjunto de datos:

- **Mutual Information:** Mide la dependencia no lineal entre cada característica y la variable objetivo. Valores más altos indican mayor capacidad predictiva y permiten capturar relaciones complejas que otros métodos podrían pasar por alto.
- **ANOVA F-Score:** Evalúa la capacidad discriminativa mediante el estadístico F del análisis de varianza, determinando si las medias de cada característica difieren significativamente entre clases. Se complementa con el *p-valor* para medir la significancia estadística.
- **Point-Biserial Correlation:** Para variables numéricas continuas, calcula la correlación entre cada predictor y la variable objetivo binaria, proporcionando una medida directa de asociación lineal con la clase.

Estos métodos se escogieron porque permiten capturar diferentes tipos de relaciones (lineales y no lineales) y son apropiados para problemas de clasificación binaria con características mixtas (numéricas y categóricas).

*2) **Resultados del Análisis Discriminativo**:* La Tabla IV presenta las 10 características más discriminativas según los tres métodos empleados.

TABLE IV
Top 10 características más discriminativas según MI, F-Score y correlación Point-Biserial

Feature	MI Score	F-Score	PB Correlation	P-Value (F)
PageValues	0.170702	3345.118895	0.503271	0.000000e+00
ExitRates	0.039179	467.190709	0.212674	2.862918e-101
ProductRelated_Duration	0.039157	214.719519	0.145974	4.104388e-48
BounceRates	0.025837	240.240058	0.154210	1.488840e-53
ProductRelated	0.025233	244.683751	0.155596	1.687386e-54
TrafficType_Encoded	0.022888	328.362762	0.179507	3.219742e-72
Administrative	0.019464	204.971821	0.142691	4.961557e-46
Administrative_Duration	0.018574	98.084428	0.099236	5.133507e-23
Month_Nov	0.012061	212.267929	NaN	1.370068e-47
Region_5	0.010435	0.481787	NaN	4.876304e-01

Se observa que:

- **PageValues** es la característica más importante, con un MI Score de **0.1707** y un F-Score de **3345.12**, lo que evidencia una alta capacidad para distinguir entre compradores y no compradores.
- **ExitRates** y **BounceRates** presentan valores significativos (MI $> \mathbf{0.02}$), confirmando que las tasas de abandono son indicadores clave del comportamiento de compra.
- **ProductRelated_Duration** muestra una correlación Point-Biserial de **0.1460**, indicando que el tiempo dedicado a páginas de productos constituye un predictor lineal relevante.
- Las variables temporales como **Month_Nov** exhiben capacidad discriminativa moderada (F-Score $> \mathbf{200}$), lo que sugiere la presencia de patrones estacionales en el comportamiento de compra.

En la Tabla V se muestran 5 de las características con menor poder discriminativo, estas incluyen principalmente:

- Variables categóricas de baja frecuencia (algunos browsers, regiones y sistemas operativos específicos).
- Variables binarias de meses con menor actividad comercial.

TABLE V
5 CARACTERÍSTICAS CON MENOR CAPACIDAD DISCRIMINATIVA

Feature	MI Score	F-Score	P-Value (F)
VisitorType_Other	0.0	0.250716	0.616583
Weekend	0.0	11.943411	0.000551
Region_6	0.0	0.913587	0.339188
Region_4	0.0	0.723481	0.395025
Browser_5	0.0	2.475705	0.115650

La Figura 4 muestra las 20 características más importantes según Mutual Information (izquierda) y ANOVA F-Score (derecha). Ambos métodos convergen en identificar las mismas variables principales, validando la robustez del análisis. Se observa una clara separación entre las características con alto poder discriminativo (*PageValues*, *ExitRates*, *BounceRates*) y el resto.

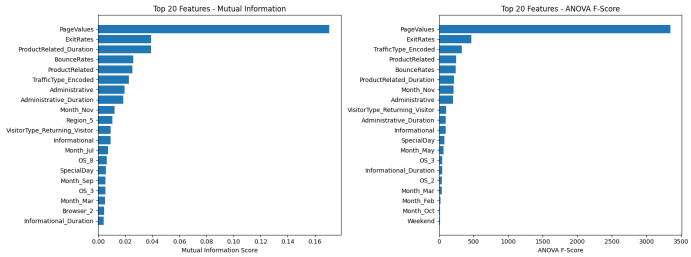

Fig. 4. Top 20 características más importantes según Mutual Information (izquierda) y ANOVA F-Score (derecha).

*3) **Identificación de Características Candidatas a Eliminación**:* Para identificar características candidatas a eliminación, se aplicó un criterio conservador múltiple:

- MI Score < 0.001: información mutua prácticamente nula.
- F-Score < 1.0: bajo poder discriminativo paramétrico.
- P-Value > 0.05: ausencia de significancia estadística.

Aplicando estos criterios simultáneamente, se identificaron 6 características candidatas a eliminación, presentadas en la tablas VI:

TABLE VI
CARACTERÍSTICAS CON MENOR CAPACIDAD DISCRIMINATIVA

Feature	MI Score	F-Score	P-Value
Browser_Other	0.0000	0.0680	0.7943
Region_4	0.0000	0.7235	0.3950
Region_6	0.0000	0.9136	0.3392
Region_8	0.0000	0.4017	0.5262
Region_9	0.0004	0.3259	0.5681
VisitorType_Other	0.0000	0.2507	0.6166

Estas características presentan valores bajos en todos los índices evaluados, sugiriendo que aportan información irrelevante al modelo.

*4) **Interpretación**:* Del análisis individual de las variables se observa que las características asociadas al comportamiento del usuario tienen los predictores más influyentes del comportamiento de compra. En particular, las métricas de interacción, como las duraciones de navegación, las tasas de salida y rebote, y el valor de las páginas visitadas, estas dominan el aporte discriminativo dentro del conjunto de características. Adicionalmente, se identifican efectos estacionales, variables temporales como *Month_Nov* muestran capacidad discriminativa, aunque su contribución es considerablemente menor en comparación con las variables de comportamiento.

El análisis también muestra que las 43 características resultantes del preprocesamiento tienen niveles adecuados de poder predictivo, con pocas variables candidatas a eliminación directa. Sin embargo, la dimensionalidad alta del conjunto de datos justifica la aplicación de técnicas de reducción o extracción de características, tanto para mejorar la eficiencia como para facilitar la interpretabilidad del modelo.

B. Extracción de Características Lineal mediante PCA

En esta sección se aplica el método de *Principal Component Analysis* (PCA) con el objetivo de reducir la dimensionalidad del conjunto de datos y evaluar su efecto sobre los dos mejores modelos obtenidos en la Sección 4: una red neuronal *Multi-Layer Perceptron* (MLP) y un árbol de decisión optimizado. Para analizar la estructura lineal del espacio de características, se calculó la varianza explicada por cada componente principal y su acumulación progresiva. La Figura 6 presenta tanto la varianza individual como la varianza acumulada, permitiendo observar cómo se distribuye la información en el espacio transformado.

*1) **Criterio para la Selección del Número de Componentes**:* Para seleccionar el número adecuado de componentes principales, se tomó el criterio estándar de tener un porcentaje mínimo de varianza explicada acumulada. El equipo definió

conservar al menos el 95% de la varianza total, este umbral es utilizado en aplicaciones de reducción dimensional supervisada, ya que mantiene la mayor parte de la estructura estadística original al tiempo que reduce el riesgo de sobreajuste y simplifica la representación. El PCA fue aplicado inicialmente sin restricciones sobre el número de componentes, con el propósito de analizar cómo se distribuye la varianza a lo largo de ellas y determinar cuántas son suficientes para alcanzar el umbral escogido. Como puede observarse en la Figura 6, la curva de varianza acumulada tiene un crecimiento rapido en la primera componente, alcanzando por sí sola el 98.20% de la varianza total. Esto indica que una única componente es suficiente para cumplir el umbral del 95% y también el del 90%.

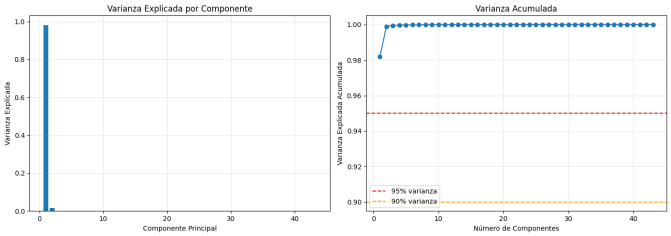

Fig. 5. Varianza explicada por componente principal y varianza acumulada del método PCA aplicado al conjunto de entrenamiento.

*2) **Resultados de la Varianza Explicada y Selección de Componentes**:* Los resultados obtenidos muestran un comportamiento particular. El conjunto de datos original contiene 43 variables, pero la primera componente principal explica por sí sola el 98.20% de la varianza total. Esto implica que la estructura estadística del conjunto está altamente concentrada en una sola dirección del espacio, lo que refleja una correlación fuerte entre las características originales. En consecuencia, tanto el umbral del 90% como el del 95% se alcanzan utilizando únicamente una componente principal. La reducción dimensional lograda se cuantifica como:

$$\text{Reducción} = 1 - \frac{1}{43} = 97.67\%.$$

Este resultado indica que se puede representar prácticamente toda la variabilidad de los datos usando únicamente un 2.33% del espacio original.

*3) **Análisis de las Cargas de la Componente Principal**:* Al examinar las cargas asociadas a la primera componente principal, se encontró que la variable `Informational_Duration` domina completamente la estructura lineal del PCA, con un valor de carga de 1.0000, tal como se muestra en la Tabla VII. El resto de las variables presenta pesos extremadamente pequeños (menores al 1%), lo cual indica que prácticamente no contribuyen a la construcción de la componente. Este comportamiento confirma lo observado en la Figura 6, aunque la primera componente explica casi toda la varianza, dada esta información, se ve como prácticamente se determina por una sola variable, limitando el valor informativo real del PCA para este problema y produciendo una reducción dimensional que elimina relaciones relevantes entre características.

TABLE VII
CARGAS DE LAS PRIMERAS 5 VARIABLES EN LA PRIMERA COMPONENTE PRINCIPAL (PC1)

Variable	Carga en PC1
Informational_Duration	1.0000
Informational	0.0057
PageValues	0.0041
ProductRelated_Duration	0.0037
Administrative_Duration	0.0034

*4) **Evaluación de los Modelos con PCA**:* Se reentrenaron los modelos *MLP* y *Decision Tree* utilizando únicamente la componente principal seleccionada. A pesar de conservar el 98.20% de la varianza, el rendimiento de ambos modelos se vio afectado de forma significativa. El MLP colapsó completamente, prediciendo únicamente la clase negativa y obteniendo valores nulos de *precision*, *recall* y *F1-score*. El árbol de decisión mantuvo cierta capacidad discriminativa, pero su desempeño disminuyó considerablemente, especialmente en *recall*. Esto confirma que, aunque la varianza retenida es alta, la información útil para la clasificación no se preserva adecuadamente tras la transformación.

*5) **Resultados Comparativos**:* La Tabla VIII resume el desempeño de los modelos antes y después de aplicar PCA. Se observa una desmejora marcada, particularmente en el MLP, cuyo modelo deja de ser funcional bajo la representación reducida.

TABLE VIII
COMPARACIÓN DE DESEMPEÑO ENTRE MODELOS OPTIMIZADOS Y MODELOS CON PCA.

Modelo	Accuracy	Precision	Recall	F1-Score	ROC-AUC
MLP (Optimizado)	0.8873	0.6135	0.7356	0.6690	0.9144
MLP (PCA)	0.8451	0.0000	0.0000	0.0000	0.7652
Decision Tree (Optimizado)	0.8909	0.6342	0.6990	0.6650	0.9149
Decision Tree (PCA)	0.8658	0.6474	0.2932	0.4036	0.8376

Los resultados permiten concluir que, aunque PCA logra una reducción dimensional del 97.67%, esta transformación no contribuye al desempeño predictivo en nuestro caso. Tal como se ve en la Figura 6, la varianza está concentrada casi exclusivamente en una única variable, lo que limita el valor discriminativo de la representación reducida. Por esto, lo recomendable es no incorporar PCA, aunque este análisis es fundamental para comprender la estructura interna del conjunto de datos.

C. Extracción de Características No Lineal

En esta sección se aplica el método Uniform Manifold Approximation and Projection (UMAP) debido a que el resultado con PCA fué inadecuado para este problema al tener la concentración de la varianza en una única variable. UMAP a diferencia de PCA, es capaz de preservar tanto la estructura local como la global de los datos mediante la construcción de

un grafo de vecindarios. Por lo tanto este tipo de técnica es efectiva en problemas con relaciones características complejas y que no son necesariamente lineales.

*1) **Configuración Óptima**:* El análisis reveló que 5 componetntes logran el mejor equilibrio y este alcanza una reducción de 88.37% de dimensionalidad y mantiene métricas competitivas. Con esto se es claro que UMAP preserva las relaciones discrimativas entre características frente a PCA.

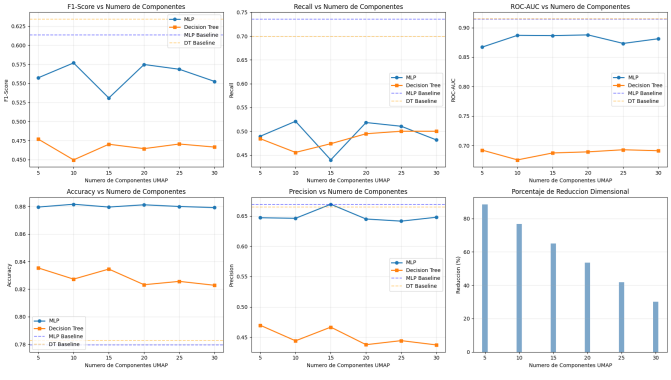

Fig. 6. Comparacion de componentes UMAP.

*2) **Medotología de evaluación**:* En este caso se evaluaron diferentes números de componentes (5, 10, 15, 20, 25, 30) manteniendo los hiperparámetros fijos. Se entrenaron los modelos (MLP y Decision Tree) utilizando los datos con SMOTE al 15% y se midió su desempeño. El criterio usado para la selección consistió en la maximización de la métrica F1-Score promedio entre ambos modelos, con esto aseguramos la reducción de dimensionalidad de al menos un 50%.

*3) **Tabla de resultados**:* La Tabla IX presenta la comparación entre las tres aproximaciones de reducción dimensional evaluadas. Se observa que UMAP supera significativamente a PCA, manteniendo métricas competitivas cercanas al baseline.

TABLE IX
COMPARACIÓN DE DESEMPEÑO ENTRE BASELINE, PCA Y UMAP.

Modelo	Acc.	Prec.	Rec.	F1	AUC
MLP (Baseline)	0.7795	0.6690	0.7356	0.6135	0.9144
MLP (PCA)	0.8451	0.0000	0.0000	0.0000	0.7286
MLP (UMAP)	0.8812	0.6450	0.5183	0.5747	0.8877
DT (Baseline)	0.7829	0.6650	0.6990	0.6342	0.9149
DT (PCA)	0.8256	0.4268	0.3665	0.3944	0.6369
DT (UMAP)	0.8232	0.4375	0.4948	0.4644	0.6891

*4) **Discusión y Conclusiones**:* Por último, el análisis individual de cada variable permitió identificar algunas características de comportamiento por parte del usuario (`PageValues_ExitRates_BounceRates_ProductRelated-Duration`) son las más discrimantivas a la hora de predecir la intención de compra. Algunas variables que son muestran una capacidad de predicción sustancialmente bajas como lo son (`Browser_Region_BounceRates_OperatingSystems`). Algunas variables que son temporales muestran una capacidad de predicción baja pero aún así es un poco más alta que las mencionadas anteriormente, una variable que se encuentra aquí es (`Month`).

Es importante resaltar que PCA logró una reduccion dimensional significativa logrando mantener la varianza en 95% pero con un impacto sumamente negativo en el rendimiento, en el caso de MLP obtuvimos un colapso muy considerable (F1=0.0000) y el Decision Tree tuvo una caida considerable (F1=0.4036 vs 0.6342 baseline).

UMAP demostró ser totalmente superior a PCA al preservar mejor la estructura discriminativa de los datos.Con la configuración óptima (determinada por maximizar F1-Score promedio con reducción $\geq 50\%$), ambos modelos mantuvieron rendimiento cercano al baseline, confirmando que las relaciones no lineales capturadas por UMAP son más apropiadas para este problema de clasificación.

Con esto podemos decir que los modelos baseline con SMOTE 15% (MLP F1=0.6135, Decision Tree F1=0.6342) ofrecen el mejor rendimiento absoluto y que UMAP es la técnica de reducción dimensional recomendada cuando se requiere balance entre eficiencia computacional y rendimiento predictivo, logrando obtener reducciones altas (más del 50%) teniendo una caída mínima. PCA, aunque logra alta reducción dimensional, no es apropiado para este problema debido a la concentración de varianza en una única variable.

VI. DISCUSIÓN Y CONCLUSIONES

La evaluación del sistema demuestra que el preprocesamiento, el balanceo mediante SMOTE y la optimización con GridSearchCV permitieron obtener modelos aceptables para predecir la intención de compra teniendo en cuenta que se tenía un conjunto de datos muy desbalanceado. Los dos modelos con mejor desempeño fueron el MLP y el árbol de decisión optimizado, ambos combinados con SMOTE al 15%. El MLP alcanzó un F1-Score de 0.6690 y un AUC de 0.9144, mientras que el árbol de decisión obtuvo un F1-Score de 0.6650 y un AUC de 0.9149. Estos valores muestran una buena capacidad discriminativa, especialmente en recall, lo cual resulta importante para reducir la pérdida de compradores reales no identificados. Por otro lado, modelos como Random Forest y XGBoost mostraron una precisión ligeramente superior pero sacrificaron recall, lo cual incrementa la cantidad de falsos negativos.

La comparación con el estado del arte muestra una coherencia general con los estudios revisados. Aunque trabajos como los de Sakar et al. [1] y Requena et al. [2] reportan métricas superiores (F1 $\approx$ 0.87–0.91), estos emplean arquitecturas más avanzadas como LSTM, MLP profundos o aprendizaje de embeddings, así como procesos de selección de características más completos. Dado que el proyecto se limita a modelos más tradicionales, los resultados obtenidos están dentro de un rango esperado. De igual forma, los valores de AUC de nuestros modelos basados en árboles ($\approx$ 0.92) son

comparables con los reportados por Tanvir et al. [3], donde utilizan XGBoost y SMOTE bajo condiciones similares.

Para analizar esta comparación, en la Tabla X se presentan los resultados más relevantes frente a los estudios de la Sección III.

TABLE X
COMPARACIÓN GENERAL CON EL ESTADO DEL ARTE

Estudio	**Modelo Destacado**	**Métrica**	**Comparación**
Sakar et al. (2018)	MLP + SMOTE + FS	F1 = 0.87	Superior; usan modelos más profundos
Requena et al. (2020)	S2L-LSTM	F1 $\approx$ 0.91	Superior; arquitectura secuencial
Tanvir et al. (2023)	XGBoost + SMOTE	AUC = 0.937	Similar; nuestro AUC = 0.9296
Gomes et al. (2022)	Embeddings + LSTM	F1 = 0.872, AUC = 0.931	Similar AUC; superior en F1; usan embeddings y modelo secuencial
Satu & Islam (2023)	Random Forest	F1 = 0.924	Superior; pipeline de mayor complejidad

En cuanto a la reducción de dimensión, aunque PCA tuvo el 98.20% de la varianza, redujo el desempeño del MLP (F1 = 0.0000) y bajó el del árbol de decisión, debido a que la primera componente estuvo dominada casi exclusivamente por una sola variable. Esto nos hace ver que alta varianza explicada no implica alta capacidad predictiva. Por el contrario, UMAP logró preservar relaciones no lineales y mantuvo niveles de desempeño cercanos al baseline con reducciones superiores al 50%, lo que lo convierte en una buena alternativa si la prioridad es la eficiencia computacional.

En conclusión, los modelos optimizados con SMOTE al 15% representan una solución efectiva y coherente con la literatura para métodos tradicionales. Aunque arquitecturas secuenciales o basadas en embeddings podrían mejorar el desempeño. UMAP es la técnica recomendada para reducción de dimensión, mientras que PCA no resulta adecuado para este problema debido a la estructura interna del dataset.

REFERENCES

[1] C. O. Sakar, S. O. Polat, M. Katircioglu, and Y. Kastro, "Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks," *Neural Computing and Applications*, vol. 31, no. 10, pp. 6893–6908, 2018. [Online]. Available: https://2024.sci-hub.cat/6856/e5cfb30c80dd18fbc513644a1aa089ea/10.1007@s00521-018-3523-0.pdf

[2] B. Requena, G. Cassani, J. Tagliabue, C. Greco, and L. Lacasa, "Shopper intent prediction from clickstream e-commerce data with minimal browsing information," *Scientific Reports*, vol. 10, no. 1, p. 16983, 2020. Available: https://www.nature.com/articles/s41598-020-73622-y

[3] A.-A. Tanvir, I. A. Khandokar, A. M. Muzahidul Islam, S. Islam, and S. Shatabda, "A gradient boosting classifier for purchase intention prediction of online shoppers," *Heliyon*, vol. 9, no. 4, p. e15163, 2023. [Online]. Available: https://www.cell.com/heliyon/fulltext/S2405-8440(23)02370-8

[4] M. A. Gomes, R. Meyes, P. Meisen, and T. Meisen, "Will This Online Shopping Session Succeed? Predicting Customer's Purchase Intention Using Embeddings," in *Proceedings of the 31st ACM Int'l Conference on Information and Knowledge Management (CIKM '22)*, 2022, pp. 2873–2882. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/3511808.3557127

[5] M. S. Satu and S. F. Islam, "Modeling online customer purchase intention behavior applying different feature engineering and classification techniques," *Discover Artificial Intelligence*, vol. 3, no. 1, p. 36, 2023. [Online]. Available: https://link.springer.com/article/10.1007/s44163-023-00086-0