Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ingeniería de Sistemas e Informática

Escuela Profesional de Ingeniería de Sistemas

Predicción en línea del abandono de carrito de compras de un cliente en el sitio web de un e-commerce con técnicas de machine learning

TESIS

Para optar el Título Profesional de Ingeniero de Sistemas

AUTOR

Benjamin CALDERON MENDOZA

ASESOR

Dra. Rosa Sumactika DELGADILLO AVILA

Lima, Perú

2023

Referencia bibliográfica

Calderon, B. (2023). *Predicción en línea del abandono de carrito de compras de un cliente en el sitio web de un e-commerce con técnicas de machine learning*. [Tesis de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Escuela Profesional de Ingeniería de Sistemas]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios autor/ asesor

Datos de autor	
Nombres y apellidos	Benjamin Calderon Mendoza
Tipo de documento de identidad	DNI
Número de documento de identidad	70081157
URL de ORCID	https://orcid.org/0009-0001-5865-4942
Datos de asesor	
Nombres y apellidos	Rosa Sumactika Delgadillo Avila
Tipo de documento de identidad	DNI
Número de documento de identidad	06445553
URL de ORCID	https://orcid.org/0000-0003-4899-3008
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Santiago Domingo Moquillaza Henríquez
Tipo de documento	DNI
Número de documento de identidad	08280889
Miembro del jurado 1	
Nombres y apellidos	Luis Alberto Alarcón Loayza
Tipo de documento	DNI
Número de documento de identidad	00456684
Datos de investigación	

Línea de investigación	C.0.5.8. Inteligencia artificial
Grupo de investigación	No aplica
Agencia de financiamiento	Sin financiamiento
Ubicación geográfica de la investigación	País: Perú Departamento: Lima Provincia: Lima Distrito: Lima Latitud: -12.05592 Longitud: -77.08457
Año o rango de años en que se realizó la investigación	2022 - 2023
URL de disciplinas OCDE	Ingeniería de sistemas y comunicaciones https://purl.org/pe-repo/ocde/ford#2.02.04

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERIA DE SISTEMAS E INFORMATICA
Escuela Profesional de Ingeniería de Sistemas

Acta Virtual de Sustentación de Tesis

Siendo las19:06 horas del día 12 de octubre del año 2023, se reunieron virtualmente los docentes designados como miembros de Jurado de Tesis, presidido por el Mg. Santiago D. Moquillaza Henríquez, Lic. Luis A. Alarcón Loayza (Miembro) y la Dra Rosa S. Delgadillo Ávila (Miembro Asesor), usando la plataforma Meet (https://meet.google.com/uyx-wdyg-mha), para la sustentación Virtual de la tesis Intitulada:**" PREDICCIÓN EN LÍNEA DEL ABANDONO DE CARRITO DE COMPRAS DE UN CLIENTE EN EL SITIO WEB DE UN E-COMMERCE CON TÉCNICAS DE MACHINE LEARNING",** del Bachiller: **Benjamin Calderon Mendoza;** para obtener el Título Profesional de Ingeniero de Sistemas.

Acto seguido de la exposición de la Tesis, el Presidente invitó al Bachiller a responder las preguntas formuladas por los Miembros del Jurado.

El Bachiller, en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez las preguntas formuladas por los señores miembros del Jurado.

Finalmente habiéndose efectuado la calificación correspondiente por los miembros del Jurado, el bachiller obtuvo la nota de **Dieciocho.**

A continuación, el Presidente del Jurado Mg. Santiago D. Moquillaza Henríquez, declara al Bachiller **Ingeniero de Sistemas.**

Siendo 20:10 horas, se levantó la sesión.

Mg. Santiago D. Moquillaza Henríquez
Presidente

Lic. Luis A. Alarcón Loayza
Miembro

Dra. Rosa S. Delgadillo Avila
Miembro Asesor

Yo **ROSA SUMACTIKA DELGADILLO AVILA** en mi condición de **asesor** acreditado con la Resolución Directoral N° 000005-2022-EPISI-FISI/UNMSM de la **tesis**/monografía/informe de investigación/trabajo académico, cuyo título es **PREDICCIÓN EN LÍNEA DEL ABANDONO DE CARRITO DE COMPRAS DE UN CLIENTE EN EL SITIO WEB DE UN E-COMMERCE CON TÉCNICAS DE MACHINE LEARNING**, presentado por el **bachiller**/magíster/egresado/licenciado/estudiante **BENJAMIN CALDERON MENDOZA** para optar el grado**/título**/especialidad de **INGENIERO DE SISTEMAS**, CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y deSimilitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según larevisión, análisis y evaluación mediante el software de similitud textual, el documentoevaluado cuenta con el porcentaje de **6.%** de similitud, nivel **PERMITIDO** para continuar con los trámites correspondientes y para su **publicación en el repositorio institucional.**

Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del grado/ título/ especialidad correspondiente.

Firma del Asesor _____

DNI: 06445553

Nombres y apellidos del asesor:
Rosa Sumactika Delgadillo Avila de Mauricio

DEDICATORIA

A mi padre, madre y hermano por su apoyo constante, a mi tío y tía que me brindaron su conocimiento y sabiduría; y a mis profesores que me tuvieron paciencia y me enseñaron lo necesario para poder concluir esta tesis.

Benjamin Calderon

AGRADECIMIENTOS

A mi familia por su constante apoyo en todo el proceso de desarrollo de esta tesis.

A mis profesores de la Universidad Nacional Mayor de San Marcos, por su paciencia al enseñar y por todos los conocimientos que me brindaron que fueron la base que me sirvió para el desarrollo de esta tesis. En especial a mi asesora de tesis la profesora Rosa Delgadillo que estuvo siempre dispuesta a apoyarme corrigiendo y guiándome en todo el proceso.

RESUMEN

El abandono del carrito de compras es un problema muy común que genera pérdidas potenciales a los *e-commerce*, quienes hoy en día son la nueva forma de comercio. En esta investigación se propone el uso de 3 técnicas de Machine Learning: Extreme Gradient Boosting Machine, AdaBoost y Bagging; para la predicción del abandono de carrito de compras de forma online en la página web de un *e-commerce* dedicado a la venta de libros, prediciendo la intención del cliente antes que termine su sesión. Para ello el sistema predictor implementa las técnicas mencionadas y predice el abandono o no del carrito a través de un proceso de votación de los predictores, siendo el resultado final del sistema lo que la mayoría de predictores decida. El sistema es implementado a manera de un servicio web para que la página web pueda consultar la predicción en tiempo real y brinde oportunidad a la empresa de realizar acciones de marketing que convenza al cliente de no abandonar y efectuar la compra de sus productos. Finalmente basándose en la métrica de *recall* se compara los resultados obtenidos de cada uno de los modelos predictivos contra los resultados obtenidos por el sistema de predicción online, obteniendo el sistema un valor de 0.9443 mejor que los resultados obtenidos por los modelos.

Palabras Claves: Machine Learning, predicción, servicio web, abandono del carrito de compras.

ABSTRACT

Shopping cart abandonment is a very common problem that generates potential losses to e-commerce, which today is the new form of commerce. This research proposes the use of 3 Machine Learning techniques: Extreme Gradient Boosting Machine, AdaBoost and Bagging; for the prediction of online shopping cart abandonment in the website of an e-commerce dedicated to the sale of books, predicting the customer's intention before the end of the session. For this, the predictor system implements the mentioned techniques and predicts the cart abandonment or not through a voting process of the predictors, being the final result of the system what the majority of predictors decide. The system is implemented as a web service so that the web page can consult the prediction in real time and provides the company with the opportunity to carry out marketing actions that convince the customer not to abandon and purchase its products. Finally, based on the recall metric, the results obtained from each of the predictive models are compared against the results obtained by the online prediction system, obtaining a value of 0.9443 better than the results obtained by the models.

Key words: Machine Learning, prediction, web service, shopping cart abandonment.

Índice

Lista de Tablas

Lista de Figuras

Capítulo 1: Introducción

En este capítulo se muestra a grandes rasgos lo que se busca obtener con esta investigación, además del alcance y la justificación del mismo. También se presenta el problema, sus antecedentes y la solución planteada.

1.1.Antecedentes

En el transcurso de los años, la mejora y aceptación de las tecnologías de la información ha ido en aumento. Jagatjyoti et al. (2018) señalan que la cantidad de personas que compran productos a través de páginas web aumentó, así el *e-commerce* o comercio electrónico se ha convertido en la nueva forma de vender productos a todo el mundo. Esta manera de venta presenta entre sus características una fuerte dependencia con la página web que muestra, debido a esto Dirk & Dirk (2012) mencionan que el éxito del *e-commerce* depende en gran medida de la calidad de la página web; esto obligó a que este recurso tenga que estar mejor diseñado. Por ello, los autores realizaron investigaciones para saber si una página es óptima. Llegando a encontrar factores (palabras) de éxito que son usados por *e-commerce* en función de sus patrones textuales semánticos.

Con el paso del tiempo los investigadores empiezan a abordar el problema desde otra perspectiva. En esta ocasión orientado a la aceptación del producto por parte de los compradores. Jagatjyoti et al (2018) lograron obtener este conocimiento analizando los comentarios que los usuarios podían dar de un producto, para obtener conjuntos de palabras positivas y negativas para dichos productos los cuales a través de un análisis nos darían como respuesta la aceptación del producto.

Las tecnologías continuaron mejorando y su aceptación creció, con ello la demanda de productos a los *e-commerce* se incrementó y por consiguiente también la oferta apareciendo como respuesta a este aumento de productos los sistemas de recomendación, Anahita & Mainak

(2018) mencionan que el éxito del *e-commerce* se ha ligado fuertemente a los sistemas de recomendación.

Ahora con la posibilidad de obtener datos acerca de la navegación del usuario en una página, nuevos métodos son empleados en beneficio de los *e-commerce*. Uno de ellos es determinar el comportamiento de los clientes para poder influir en su decisión de compra, esto fue investigado por Ahmet & Mehmet (2019) en la predicción de la intención de compra.

1.2.Problema

Los clientes son una parte importante en el negocio ya que son los que compran los productos generando ganancias a la empresa, por lo tanto, conocer su comportamiento y sus intenciones al estar en la página web es una gran ventaja que se puede explotar para que, al finalizar la sesión, el usuario haya realizado una compra.

En la página web de un *e-commerce* los clientes tienen la opción de añadir los productos que desean a un carrito para posteriormente poder realizar la compra de todos ellos. Sin embargo, muchos de los clientes después de haber agregado productos al carrito al final no realizan la compra y solo se retiran del sitio web. De poder determinar quiénes serán los clientes que no realizarán su compra, la empresa podría tomar acciones para enfrentar este problema.

¿Qué técnica o técnicas utilizar para conocer con anticipación que clientes no realizarán la compra de los productos añadidos a su carrito? ¿Cómo aplicar estas técnicas para el *e-commerce*? Son las preguntas que se intentan resolver en esta investigación.

1.3.Objetivo

Construir un sistema predictor que emplee modelos predictivos para predecir el abandono del carrito de compras, esto es, predecir "n" minutos después de que el cliente entró a la página web del *e-commerce*, si el cliente realizará o no la compra de sus productos.

1.4.Justificación

El problema de abandono del carrito de compras es más común de lo que aparenta, en Jae-Do (2019) menciona que el 77 % de los compradores en línea abandonan sus carritos antes de completar la compra. Como tal, una compra no realizada se puede traducir en perdidas para la empresa, en Rubin et al. (2020) mencionan que las perdidas estimadas llegan a un total de 18 mil millones de dólares anualmente. Como se puede observar el abandono del carrito de compras no es un problema que se deba dejar pasar.

Por este motivo determinar la intención de compra del cliente es útil para el *e-commerce*, ya que esto brinda información importante para poder determinar que clientes no van a efectuar su compra y poder tomar acciones sobre ello. Se menciona en Osnat et al. (2019) que este mecanismo puede guiar a un sistema de recomendación para mejorar la experiencia de compra en beneficio tanto del comprador como del vendedor.

1.5.Alcance

Esta investigación busca predecir la intención de compra de un cliente en la página web de un *e-commerce* después de que el cliente añada productos al carrito y antes de que su sesión finalice para poder determinar si el cliente abandonará la compra de los productos añadidos en su carrito o no. Para ello se tendrá como caso de estudio un *e-commerce* peruano que se ha dedicado a la venta de libros, tanto por tiendas físicas como por su página web, por muchos años y tiene mucha experiencia en el rubro.

En los siguientes capítulos se muestra algunas definiciones previas de los conceptos a utilizar, el estado del arte con los artículos revisados y las técnicas seleccionadas, el aporte de esta investigación y el desarrollo del sistema predictor, la validación de la propuesta y finalmente las conclusiones a las que se llegaron con la investigación.

Capítulo 2: Marco teórico

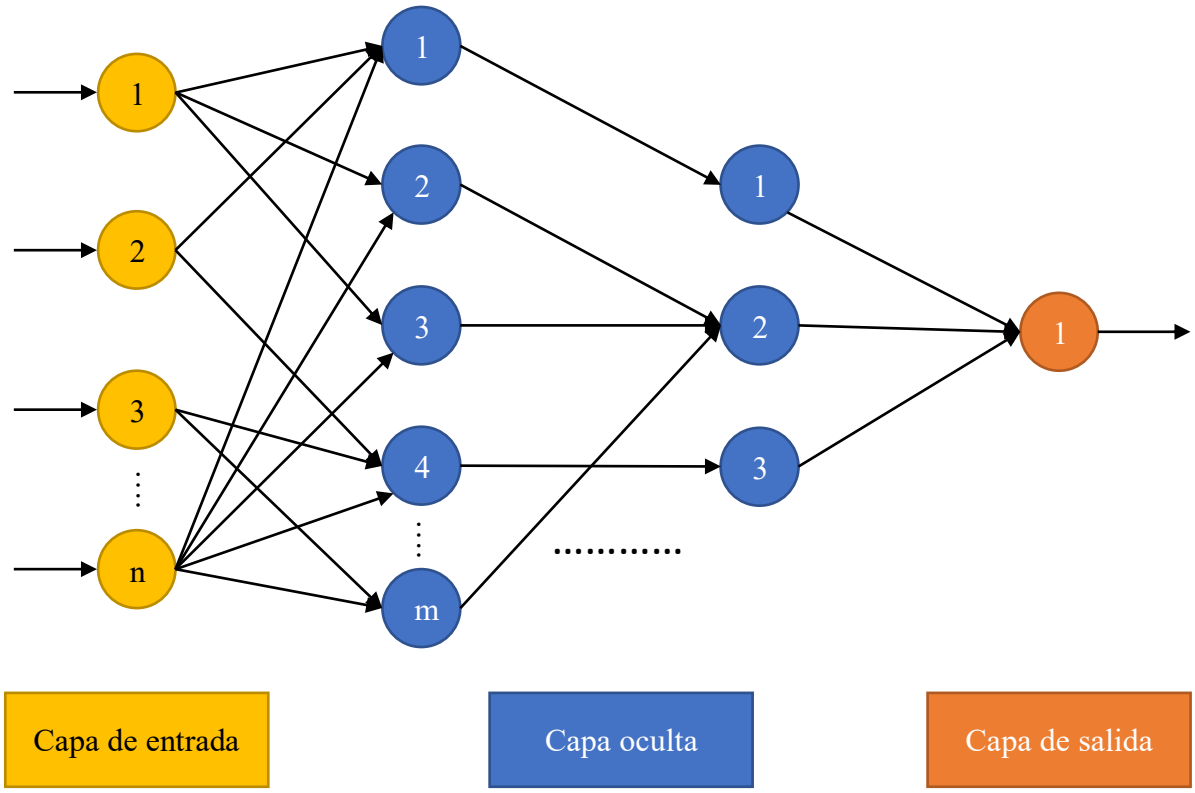
En este capítulo se define brevemente los conceptos, métricas, algoritmos y otros que se utilizarán en capítulos posteriores.

2.1.Redes Neuronales (RN)

"Una red neuronal es un método de la inteligencia artificial que enseña a las computadoras a procesar datos de una manera que está inspirada en la forma en que lo hace el cerebro humano" (Amazon Web Service, s.f.). Las redes neuronales son una técnica de *machine learning* que utiliza una estructura de capas donde se encuentran neuronas interconectadas, de forma que la red puede aprender de sus errores y lograr resolver problemas complejos.

Figura 1

Estructura estándar de una red neuronal



En la Figura 1 se muestra la estructura general de una red neuronal artificial que recibe los datos del exterior por las neuronas de la capa de entrada, estos datos serán procesados a través

de las neuronas que se encuentran en las distintas capas que conforman la capa oculta de la red para finalmente llegar el dato transformado a las neuronas de la capa de salida quienes lanzarán el nuevo dato al exterior.

Existen distintos tipos de redes neuronales de las cuales a continuación se menciona solo las que aparecen en esta investigación.

2.1.1.Red Neuronal Multicapa

Una red neuronal multicapa como su nombre menciona es una red neuronal que contiene más de una capa de neuronas ocultas artificiales para el procesamiento de información. Es el tipo de red neuronal más común y utilizada.

2.1.2.Red Neuronal Recurrente

"Las redes neuronales recurrentes (RNN) son una clase de RN de aprendizaje profundo basada en los trabajos de David Rumelhart en 1986. Las RNN son conocidas por su capacidad para procesar y obtener información de datos secuenciales" Arana (2021). Una RNN presenta conexiones arbitrarias entre distintas neuronas de forma que permite a la red que tenga memoria es decir que recuerden las salidas de las anteriores neuronas como entrada de las nuevas. Estas características las hacen ideales para el análisis de textos, sonidos o videos.

2.1.3.Red Neuronal de Memoria Corta y Larga (LSTM)

Las redes neuronales de memoria de corto y largo plazo (LSTM) son un tipo particular de RNN que solucionan el problema de las RNN asociado a la memoria de corto plazo: el desvanecimiento o decaimiento del gradiente, y su explosión (Arana, 2021). Las LSTM presentan una unidad de memoria que les permite discriminar si un dato es relevante y debe ser almacenado en la memoria o es superfluo y no debe ser utilizado en futuros cálculos.

2.1.4.Red Neuronal GMDH

Este tipo de redes implementan el Método de Agrupamiento para el Manejo de Datos (GMDH) que es un algoritmo que permite describir de forma sucesiva un sistema complejo de relaciones a partir de simples operaciones matemáticas (Hernández & Herrera, 2012).

Esta red neuronal es una técnica auto organizada de minería de datos que sirve para poder decidir el número de variables, estructura y parámetros del modelo (Sujoy, Manoj, & Felix, 2019). Para calcular el número de neuronas de cada capa se inicia con un cantidad máxima de neuronas, definido por combinatoria, en la capa y a través de un proceso de selección se determina cual es la cantidad de neuronas idónea para dicha capa (Torra Porras, 2004).

A este tipo de redes también se les llama Redes Neuronales Polinómicas.

2.1.5.Red Neuronal Profunda (DNN)

También llamada red neuronal de aprendizaje profundo debido a que se basa en dicha técnica, esta red neuronal se caracteriza por tener una mayor cantidad de capas ocultas con la intención de simular las redes neuronales que se encuentran en la estructura del cerebro humano. “El aprendizaje profundo es un método de la inteligencia artificial (IA) que enseña a las computadoras a procesar datos de una manera que se inspira en el cerebro humano.” (Amazon Web Service, s.f.). Las DNN son utilizadas mayormente en problemas de reconocimiento de patrones complejos.

2.1.6.Backpropagation

Backpropagation o también llamado “propagación hacia atrás” es un algoritmo de aprendizaje supervisado que utiliza el descenso de gradiente para ajustar los pesos de una red neuronal artificial basado en el error obtenido en la iteración anterior, es decir empieza calculando la taza de error de las neuronas de la última capa (N) para luego utilizar dicho valor en el cálculo de la taza de error de la neurona interconectada en la capa anterior (N-1). Dada una red neuronal

artificial y una función de error, el método calcula el gradiente de la función de error con respecto a los pesos de la red neuronal.

2.2.Métricas de Evaluación

Las métricas de evaluación de un modelo de aprendizaje automático sirven para valorar el rendimiento del modelo además de estimar que tan preparado está este para la entrada de nuevos datos no presentes en la muestra usada para el aprendizaje del mismo.

Entre las métrica existentes tenemos *Precision*, *Recall*, F1 y *Accuracy* las cuales definimos a continuación.

Es importante mencionar que se utiliza los nombres de las métricas en ingles debido a que la mayoría de librerías que implementan estas métricas mantienen dichos nombre además que la traducción de estos puede llevar a confusión.

2.2.1.Matriz de Confusión

La matriz de confusión es una herramienta que permite la visualización en forma matricial del desempeño de un algoritmo de aprendizaje comparando las clases verdaderas contra las clases predichas.

Figura 2

Ejemplo de una matriz de confusión

		Valores Predichos	
		0	1
Valores Reales	0	TN	FP
	1	FN	TP

Nota. TN (*True Negative*) hace referencia a los verdaderos negativos, TP (*True Positive*) hace referencia a los verdaderos positivos, FN (*False Negative*) hace referencia a los falsos negativos y FP (*False Positive*) hace referencia a los falsos positivos.

Como resultado de esta comparación se obtienen verdaderos positivos que son predicciones donde el modelo predice 1 (Positivo) correctamente, falsos positivos que son predicciones donde el modelo predice 1 (Positivo) incorrectamente, falsos negativos que son predicciones donde el modelo predice 0 (Negativo) incorrectamente y verdaderos negativos que son predicciones donde el modelo predice 0 (Negativo) correctamente.

2.2.2.Accuracy

Esta métrica mide la cantidad de veces que el modelo ha logrado identificar correctamente los eventos en general. Para calcular esta métrica se tiene la siguiente fórmula (TP + TN) / (TP + TN + FP + FN).

Figura 3

Representación del accuracy en la matriz de confusión

		Valores Predichos	
		0	1
Valores Reales	0	TN	FP
	1	FN	TP

2.2.3.Precision

Esta métrica mide la frecuencia con la que el modelo predice correctamente la aparición del evento en análisis, de esta forma la métrica permite medir la calidad del modelo. Para calcular esta métrica se tiene la siguiente fórmula TP/(TP + FP).

Figura 4

Representación de precision en la matriz de confusión

		Valores Predichos	
		0	1
Valores Reales	0	TN	FP
	1	FN	TP

2.2.4.Recall

Esta métrica busca medir la cantidad de veces que el modelo identifica al evento en análisis. Para calcular esta métrica se tiene la siguiente fórmula TP/(TP + FN).

Figura 5

Representación de recall en la matriz de confusión

		Valores Predichos	
		0	1
Valores Reales	0	TN	FP
	1	FN	TP

2.2.5.F1 score

Esta métrica se utiliza con el objetivo de combinar las métricas *precision* y *recall* para poder comparar la precisión y exhaustividad en conjunto del modelo, siendo 1 el valor ideal y 0 el peor resultado. Para calcular esta métrica se tiene la siguiente fórmula

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall}$$

Esta fórmula es la media armónica entre *precision* y *recall*, esto con el objetivo de que el valor de la métrica no se incline a uno debido a grandes valores atípicos.

2.3.Medidas

2.3.1.Rango Recíproco Medio (MRR)

El rango recíproco medio es una medida estadística que se utiliza para evaluar cualquier modelo que tiene como salida una lista de posibles respuestas en relación a un conjunto de consultas realizadas.

Suponiendo que el usuario revisara de adelante hacia atrás en un conjunto de documentos seleccionados hasta que encuentre un documento relevante, y ese documento está en la clasificación "n", entonces la precisión del conjunto que ve es 1/n, que también es el rango recíproco. Para poder realizar dicho cálculo con Q consultas se tiene que sacar el promedio de los rangos recíprocos obtenidos, esto se traduce a la siguiente fórmula.

$$MRR = \frac{1}{Q}\sum_{i=1}^{Q}\frac{1}{rank_i}$$

Donde $rank_i$ hace referencia al rango recíproco obtenido en la consulta i.

La medida MRR es adecuada para la búsqueda de elementos conocidos, donde el usuario intenta encontrar un documento que ha visto antes o que sabe que existe. Esto se denomina búsqueda de navegación en el caso de la búsqueda web.

2.3.2.Deciles

Los deciles son una medida de posición en estadística descriptiva que representan los nueve valores que dividen una serie de datos ordenados en diez partes iguales, de manera que cada decil representa 1/10 de la muestra o población.

2.4.Algoritmos de Machine Learning

2.4.1.Regresión Logística (LR)

La regresión logística es un modelo estadístico que se utiliza en *machine learning* para ayudar a lograr predicciones precisas determinando la probabilidad de que ocurra el evento en análisis. "La regresión logística resulta útil para los casos en los que se desea predecir la presencia o ausencia de una característica o resultado según los valores de un conjunto de predictores." (IBM, s.f.).

2.4.2.Regresión Logística Binaria

La regresión logística binaria es una LR con la peculiaridad de que la variable dependiente es una variable binaria, dicotómica o también llamada *dummy* que solo puede tener dos valores. Esta técnica es muy útil cuando se quiere determinar la probabilidad de que un evento ocurra dado los valores de otras variables independientes o cuando se quiere determinar la influencia de las variables independientes sobre el evento resultante.

2.4.3.K vecinos más próximos (KNN)

El algoritmo de KNN es utilizado para resolver problemas de clasificación y regresión, basándose en la premisa de que cosas similares existen en distancias cercanas. Con esto en mente el algoritmo calcula la distancia entre un nuevo dato con los datos ya existentes del entrenamiento, selecciona los K elementos más cercanos y en el caso de clasificación se concluye con el tipo más frecuente de esos K elementos, o en el caso de regresión por el promedio.

La mayor ventaja de este algoritmo es la facilidad de entender e implementar, además que no necesita construir un modelo como otros métodos. Sin embargo con forme vaya aumentando la data a utilizar en el entrenamiento se necesitará mayor poder computacional si no se quiere que el algoritmo se vuelva lento, esto debido a que guarda la posición de todos los puntos utilizados en el entrenamiento.

2.4.4.Modelo C4.5

El modelo C4.5 es un árbol de decisión usado normalmente para problemas de clasificación que implementa la depuración de ramas para disminuir el sobre entrenamiento del modelo.

Para poder obtener el primer nodo y construir el árbol de decisión el modelo utiliza la entropía (H), nivel de desorden o incertidumbre en un conjunto de valores, la información ganada (IG)

para determinar la homogeneidad en un grupo, de esta forma entre mayor sea la información ganada más homogéneo es el grupo; y por último la ganancia promedio (GR) como función objeto para decidir si se debe dividir la data en otros nodos, también sirve para mitigar el sobre entrenamiento.

$$H(s) = \sum_{x \in X} -P(x) log_2 P(x)$$

$$IG(s,T) = H(s) - \sum_{t \in T} P(t)H(t)$$

$$GR = \frac{IG(s,T)}{H(s)}$$

2.4.5.Classification and regression tree (CART)

Como su nombre dice CART es un modelo de árbol de decisión basado en regresión el cual pude ser usado tanto para problemas de clasificación como de regresión, este modelo no requiere mucho poder computacional y permite la construcción de modelos de una manera rápida. En el caso de CART para la división de nodos se utiliza el índice Gini (GI).

$$GI = 1 - \sum_{j=1}^{c} P_j^2$$

Donde P_j hace referencia a la probabilidad de cada evento.

2.4.6.AdaBoost

AdaBoost es un algoritmo de clasificación que consiste en crear varios modelos de árboles de decisión, entrenar los pesos del primer modelo y a continuación entrenar un segundo modelo que aumenta el peso de aquellos casos que son difíciles de clasificar y disminuye el peso de los que son fáciles de clasificar con el propósito de mejorar la clasificación del modelo para los casos que no pudo el modelo anterior. Este proceso se repite la cantidad de veces que se

especifique teniendo como resultado un modelo que toma en cuenta todos los árboles anteriores y que puede realizar predicciones más certeras.

2.4.7.Gradient Boosting Machine (GBM)

GBM es un algoritmo de clasificación que realiza un proceso parecido al que realiza AdaBoost con la diferencia de que GBM utiliza la gradiente en la función de pérdida en vez de usar puntos de datos de alto peso como lo realiza AdaBoost para mejorar la clasificación del modelo.

2.4.8.Algoritmo k-means

El algoritmo *k-means* es un algoritmo de clasificación no supervisado cuyo objetivo es agrupar objetos en k grupos teniendo como base la característica de los objetos. El proceso de agrupamiento se realiza minimizando la suma de las distancias entre cada objeto y el centroide de su grupo (también llamado *cluster*), para ello primero se escoge el número de grupos, k, y se procede a establecer los k centroides de cada grupo. Segundo se pasa a asignar cada objeto a su centroide más cercano y finalmente se actualiza la posición del centroide de cada grupo, los dos últimos pasos se repiten hasta que los centroides no se mueven en la actualización.

2.5.Otros conceptos

2.5.1.Ingeniería de características

La ingeniería de características es el proceso de transformación de los datos en información útil que represente mejor el problema para las técnicas de *machine learning*, con el propósito de optimizar el desempeño de los modelos de *machine learning*.

2.5.2.Desvanecimiento de gradiente

El desvanecimiento de gradiente es un problema que ocurre al momento de entrenar una red neuronal donde la derivada parcial de la función de error se acerca a valores muy pequeños o

cero debido a la gran cantidad de capas que tiene en cuenta el algoritmo en la actualización de los pesos de la red. Esto impide que el cambio de valor de los pesos sea eficaz y en el peor de los casos puede impedir el entrenamiento de la red neuronal.

2.5.3.Scrum

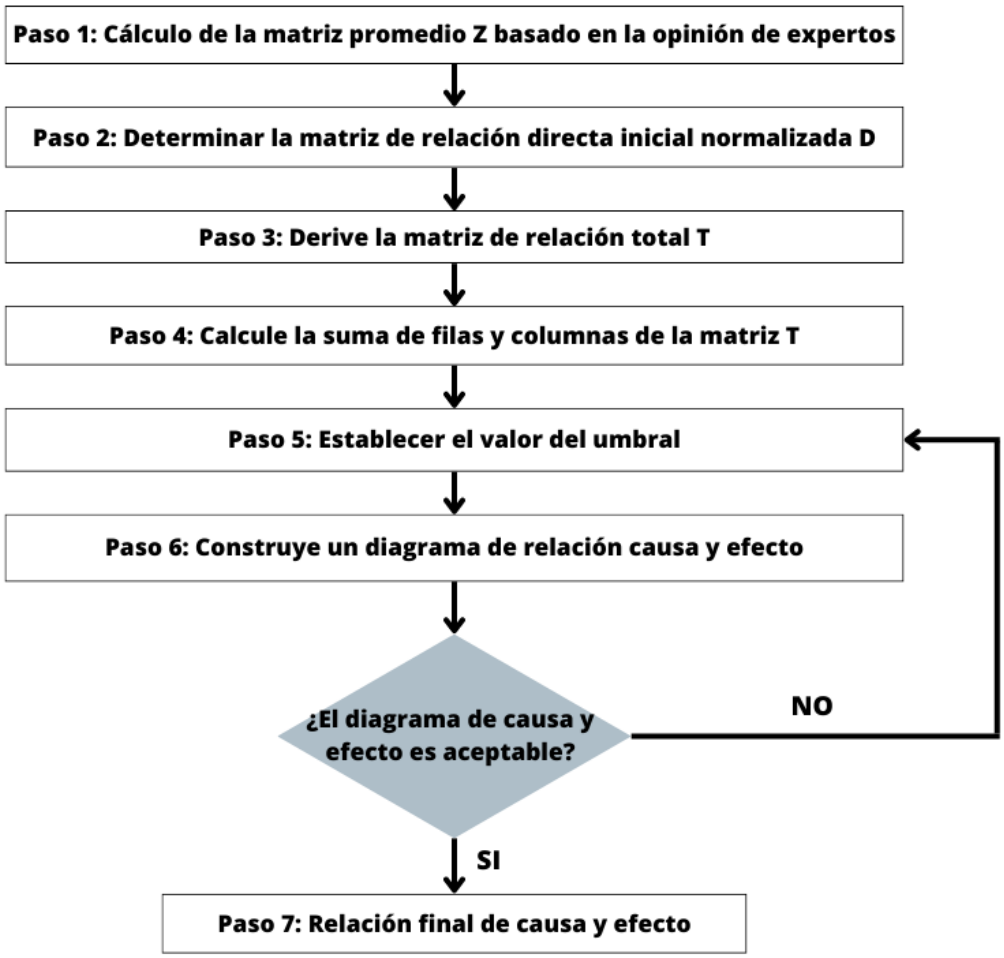
Scrum es un marco de trabajo liviano que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos (Schwaber & Sutherland, 2020). En *scrum* se trabaja con iteraciones donde al final de cada iteración se tiene una entrega parcial del producto final en la cual se aborda un avance de la solución total priorizando el beneficio y la funcionalidad de la entrega.

2.5.4.Decision Making Trial and Evaluation Laboratory (DEMATEL)

DEMATEL es un tipo de enfoque de modelado estructural que es útil para analizar las relaciones de causa y efecto entre los constituyentes de un sistema. DEMATEL se puede aplicar para confirmar la existencia de una relación/interdependencia entre componentes o reflejar el nivel relativo de relaciones dentro de ellos (Thakkar, 2021). Con este modelo se puede lograr construir una matriz que muestre la interdependencia de un grupo a través de causa y efecto, donde los valores de la matriz van desde 0 a 4 siendo 0 = sin influencia, 1 = poco influencia, 2 = influencia media, 3 = influencia alta y 4 = influencia muy alta.

Figura 6

Procedimiento DEMATEL



Nota. Recuperado y traducido de Thakkar (2021)

2.5.5. Validación cruzada

La validación cruzada es un método utilizado para evaluar el rendimiento de un modelo de machine learning con el objetivo de encontrar de forma rápida un mejor modelo.

Para aplicar la validación cruzada existen dos técnicas muy conocidas: Train-Test Split y K-Folds. La primera técnica consiste en separar de forma aleatoria un conjunto de datos para el entrenamiento y con los datos restantes se realiza una nueva separación para utilizar un subconjunto de datos como validación en el proceso de entrenamiento del modelo, los datos restantes quedan para el proceso de testeo del modelo; la segunda técnica consiste en dividir todo el conjunto de datos en K grupos donde se utiliza K-1 grupo de datos para el entrenamiento

del modelo y el grupo faltante para la validación, este proceso se repite con cada grupo y al final se queda con el modelo que obtenga la mejor métrica de evaluación deseada.

Normalmente se utiliza el Train Test Split cuando tenemos una cantidad de datos ilimitados y presentan una distribución igual entre las partes de datos, mientras que K-Folds se utiliza cuando tenemos una cantidad limitada de datos y se necesita garantizar la observación de toda la serie de datos en el entrenamiento del modelo.

Capítulo 3: Estado del arte

En este capítulo se describe brevemente los artículos revisados de la literatura sobre métodos, técnicas y algoritmos relacionados al tema de investigación. Al finalizar se realiza un breve análisis para determinar cuáles se implementan en la solución.

3.1.Metodología implementada

Para la búsqueda de la bibliografía que ayude a solventar el problema propuesto en esta investigación se tomó como guía la metodología de Kitchenham & Charters (2007) que determinan 3 fases: la fase de planeamiento donde se elaboran las preguntas de investigación, la fase de desarrollo donde se selecciona los documentos a revisar según criterios de selección y exclusión; y la fase de resultados donde se muestra el análisis realizado de los documentos revisados.

Teniendo en cuenta estas fases y adaptándolas a la investigación se plantearon las siguientes preguntas de investigación:

Q1: ¿Qué metodologías existen para la predicción del abandono del carrito de compras?

Q2: ¿Qué características se utilizan para la predicción de la intención de compra de un cliente?

Q3: ¿Qué técnicas se emplean en la obtención de los datos a utilizar en el entrenamiento de los modelos predictivos?

Una vez definidas las preguntas de investigación se pasó a realizar la búsqueda de documentos en las revistas y bancos de Journals como Decision Analytics Journal, Decision Support Systems, ScienceDirect, International Information and Engineering Technology Association (IIETA), Emerald, Electronic Commerce Research and Applications entre otras usando como metabuscadores Scopus y Springer. Las palabras claves utilizadas para la búsqueda con los

metabuscadores fueron: shopping cart abandonment, purchase intention prediction, e-commerce, machine learning y neuronal network.

Finalmente para poder filtrar los documentos obtenidos se definieron los criterios de selección y exclusión de la bibliografía.

Para los criterios de selección se tiene en cuenta lo siguiente:

- La fecha de publicación del artículo debe estar comprendido entre el 2017 al 2023.
- El artículo debe permitir responder las preguntas de investigación.
- El artículo debe estar orientado en el área de Ciencias de la computación.

Para los criterios de exclusión se tiene en cuenta lo siguiente:

- Artículos orientados a empresas que no venden sus productos por internet.
- Artículos que no tengan una versión en el idioma inglés.
- Artículos que usan técnicas que no son de inteligencia artificial.

3.2.Revisión de la literatura

De la revisión efectuada en los artículos seleccionados, se ha podido definir que técnicas se utilizarán tanto para la obtención y manejo de la información como para la construcción del sistema predictor, estas técnicas se irán explicando conforme se responda las preguntas de investigación en los siguientes párrafos.

3.2.1.Q1: ¿Qué metodologías existen para la predicción del abandono del carrito de compras?

En los artículos revisados se encontró distintas técnicas de inteligencia artificial para la predicción de la intención de compra y abandono del carrito de un cliente. En la Tabla 1 se muestra un resumen de las investigaciones seleccionadas.

Tabla 1

Resumen de los artículos revisados

Autores	Título	Año	Descripción
Bing & Yuliang	*Prediction of User's Purchase Intention Based on Machine Learning*	2017	Uso de árboles de decisión para la predicción de la intención de compra.
Grażyna & Sławomir	*Application of Neural Network to Predict Purchases in Online Store*	2017	Uso de una red neuronal multicapas con propagación para atrás para predecir la intención de compra.
Bichen & Bingwei	*A Scalable Purchase Intention Prediction System Using Extreme Gradient Boosting Machines with Browsing Content Entropy*	2018	Uso de Extreme Gradient Boosting (XGBoost) para predecir la intención de compra
Sharma & Prasanna	*A novel purchase target prediction system using extreme gradient boosting machines*	2019	Uso de XGBoost para la predicción de la intención de compra basado en los clics del cliente y la entropía de contenido.
Yunghui, Hui-Kuo & Wen-Chih	*Predicting Online User Purchase Behavior Based on Browsing History*	2019	Uso de una red neuronal recurrente para predecir la intención de compra y obtener una lista de posibles productos a adquirir.
Ahmet & Mehmet	*Prediction of Purchase Intention on the E-Commerce Clickstream Data*	2019	Uso de un modelo clasificador binario para la predicción de la intención de compra de un usuario antes que termine su sesión.
Osnat, Veronika, & Tsvi	*Will this session end with a purchase? Inferring current purchase intent of anonymous visitors*	2019	Empleo de los productos populares para predecir la intención de compra con un Extreme Gradient Boosting Machine.
Sujoy, Manoj, & Felix	*Predicting the consumer's purchase intention of durable goods: An attribute-level analysis*	2019	Empleo de los comentarios para la predicción y comparación entre un modelo lineal y no lineal.
Qian, Chun, & Shaoqing	*Prediction of Purchase Intention among E-*	2020	Determinar características claves que influyen en la intención de compra del usuario y uso de

	Commerce Platform Users Based on Big Data Analysis		XGBoost para predecir la intención de compra.
Chaudhuri, Gupta, Vamsi & Bose	*On the platform but will they buy? Predicting customers' purchase behavior using deep learning*	2021	Determinar la intención de compra con una técnica de Deep Learning, teniendo en cuenta los datos demográficos del cliente y sus datos de sesión.
Bhattacharjee, Ramesh, Jayaram & Mathad	*An integrated machine learning and DEMATEL approach for feature preference and purchase intention modelling*	2023	Uso de modelos de árboles de decisión y DEMATEL para determinar la intención de compra de celulares basado en las características de estos.

Se puede ver que **existen 2 grandes grupos de técnicas** utilizadas en los artículos revisados, las cuales son redes neuronales y técnicas de machine learning. Teniendo en cuenta esto, primero se va a explicar los artículos que utilizaron técnicas de redes neuronales y posteriormente los que utilizaron técnicas de machine learning.

Las redes neuronales son métodos muy famosos en lo que refiere a problemas de clasificación y predicción, debido a las fortalezas que han demostrado en la solución de dichos tipos de problema, por lo que no podían faltar en la predicción de la intención de compra. A continuación se muestra los artículos revisados con estas ténicas.

En Grażyna & Sławomir (2017) los autores proponen una red neuronal multicapa con el algoritmo de propagación para atrás para resolver el problema de predicción. Para el análisis de la información los autores eliminaron sesiones realizadas por otros entes que no son compradores, por ejemplo bots o administradores debido a que su comportamiento era distinto al de un usuario común.
Primero los autores buscan características de las sesiones de donde toman en cuenta información a nivel de HTTP, como es el caso de cantidad de *requests*, y otras características conectadas directamente con el usuario. Finalmente, deciden utilizar ocho características que consideran cruciales para determinar la intención de compra y

una novena que indica si el usuario compró o no. En la Tabla 2 se muestran las características seleccionadas.

Tabla 2

Variables que describen la sesión de un usuario

No.	Variable	Descripción de la variable	Valores
1	PagesNo	Número de páginas Web.	Entero
2	RequestsNo	Número de peticiones HTTP.	Entero
3	Duration	Duración de la sesión.	Entero
4	TimePerPage	Tiempo medio por página.	Flotante
5	TransferSize	Volumen total de datos enviados por el servidor.	Entero
6	LoginNo	Número de operaciones relacionadas con la operación exitosa de inicio de sesión del usuario.	Entero
7	AddNo	Número de operaciones relacionadas con añadir un artículo al carrito de compras.	Entero
8	Source	Fuente de la visita, es decir, la forma en que el usuario llegó al sitio en línea de la librería.	{$search_{org}$, $search_{paid}$, $search_{other}$, internal, direct, other}
9	Purchase	Indicador de cuando una transacción de compra fue finalizada exitosamente o no.	{0, 1}

Nota. Recuperado y traducido de Grażyna & Sławomir (2017)

Para el diseño de la red neuronal multicapa los autores utilizan 14 neuronas de entrada de las cuales 1 neurona es para la *bias*, 7 neuronas son para las características seleccionadas del 1 al 7 y debido a que la característica 8 es un arreglo de 6 palabras para determinar la fuente de la visita, los autores se ven en la obligación de usar 6 neuronas en vez de 1 para esta característica sumando un total de 14. Después, para las neuronas de salida los autores utilizan 11 neuronas donde 5 representan a las sesiones donde se realiza una compra y 6 donde no. De esta manera, se considera que la red

neuronal predice una compra si las neuronas de salida pertenecientes a sesiones de compra tienen valores por encima de 0.95 y las neuronas relacionadas a las sesiones de no compra tienen valores por debajo de 0.95, caso contrario se predice una no compra. Para la parte de entrenamiento de la red neuronal los autores usan el algoritmo de propagación para atrás para el aprendizaje y la hiperbólica tangente como función de activación. También dividen el *dataset* en dos partes una que se usara en el entrenamiento y otro que se usara para la verificación del aprendizaje de la red tanto de los casos de sesiones de compra como de no compra. Finalmente, los autores muestran los resultados de la predicción a través de una matriz de confusión. Estos datos serán mostrados en la Tabla 3.

Tabla 3

Matriz de confusión para la red neuronal

	Predicción de no compra	Predicción de compra
Sesiones de no compra	32,457	18
Sesiones de compra	106	762

Nota. Recuperado y traducido de Grażyna & Sławomir (2017)

De donde determina *Accuracy* = 99.6 %, *Precision* = 97.7 % y *Recall* = 87.8 %. De estas tres medidas el autor hace mención que la más importante es el *Recall* por lo que su aporte logra un resultado prometedor. Sin embargo, también menciona que la precisión es importante y que el porcentaje que logran es muy alto en este aspecto.

En Yunghui et al (2019) los autores proponen una red neuronal recurrente y a diferencia de la anterior investigación no solo buscan determinar la intención de compra sino también indicar cuál es el producto que el cliente está interesado en obtener. El problema de determinar cuál es el producto que el cliente está queriendo comprar es común y ha sido resuelto por los sistemas de recomendación, sin embargo, los autores

sostienen que debido a las características de los productos que algunos *e-commerce* ofrecen, productos con precios muy altos en comparación con productos cotidianos y que en un plazo de tiempo corto es raro que un cliente repita su compra, es un problema que necesita de una solución diferente. De esto se resalta la importancia que tienen las características del producto al momento de definir la solución.

Para la obtención de la información histórica de navegación de un usuario los autores usaron Mixpanel, la cual es una empresa que brinda servicio de seguimiento de las actividades de un usuario al ingresar a la página.

Para la extracción de la información primero los autores dividieron las acciones de los clientes, las cuales eran muy generales, en acciones más específicas. Después, analizaron las características de las actividades de los clientes como por ejemplo que día de la semana se realizaban mayor cantidad de compras. Determinando patrones en el comportamiento de los clientes, uno de ellos era que la mayoría de usuarios compraba los fines de semana.

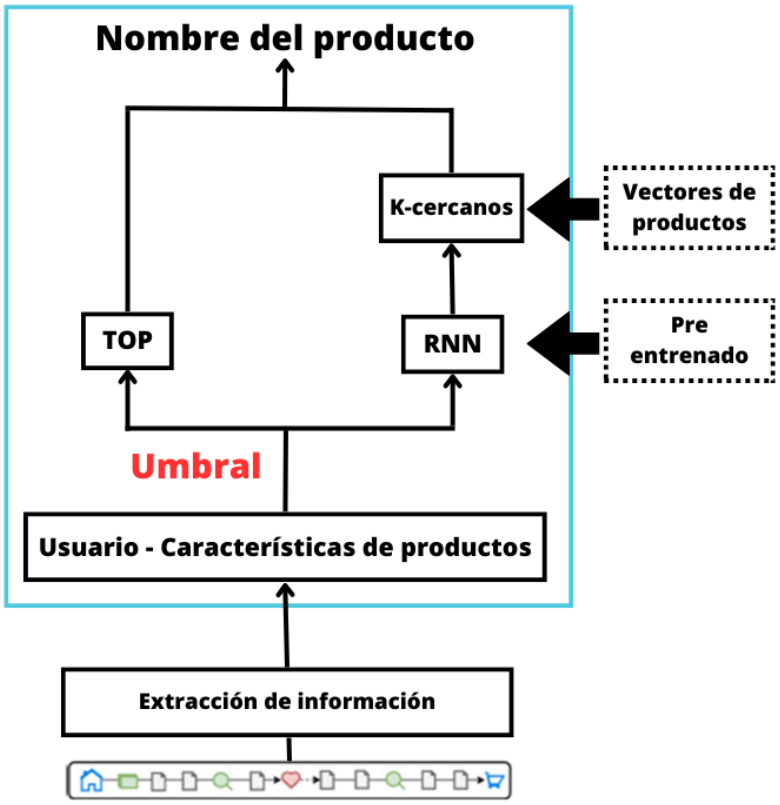
En esta investigación los autores deciden utilizar LSTM (Long Short-Term Memory) para su modelo de red neuronal debido a la cantidad de información que recibe como entrada (la entrada de la red neuronal es la data histórica de las acciones del cliente) para así no preocuparse por el problema de desvanecimiento de gradiente que podría surgir.

El modelo para la predicción de los productos que un cliente puede comprar se dividió en 2 fases y se determinó un umbral para decidir qué fase utilizar. A la primera lo llamaron “TOP” que hace mención al modelo diseñado especialmente para clientes que son muy indecisos y tienden a comparar mucho los productos antes de realizar una compra; y en la segunda fase se diseñó un modelo para los usuarios que tiene carácter

de comprador impulsivo donde se utilizó en conjunto la red neuronal y un algoritmo de *k-nearest* (k vecinos más próximos).

Figura 7

Dos fases del marco de trabajo propuesto



Nota. Recuperado de Yunghui et al. (2019)

La red neuronal es puesta a prueba comparándose con otros métodos como son el POP y el LSTM a través de las métricas "Recall" y "MRR". Además, probaron con distintos intervalos de tiempo es decir el modelo predijo con data que se obtuvo N horas antes de la compra. Los resultados obtenidos son mostrados en la Tabla 4.

Tabla 4

Resultado del testeo con datos que se registran N horas antes de la compra

N		Recall1	Recall2	Recall3	Recall4	Recall5	MRR1	MRR2	MRR3	MRR4	MRR5
6	POP	0.3763	0.4409	0.5484	0.5806	0.6129	0.3763	0.4086	0.4444	0.4525	0.4590
	LSTM	0.2151	0.3871	0.4839	0.5914	0.6667	0.2151	0.3011	0.3333	0.3602	0.3753
	NUESTRO	**0.4086**	**0.5269**	**0.5806**	**0.6452**	**0.7097**	**0.4086**	**0.4677**	**0.4857**	**0.5018**	**0.5147**

12	POP	0.2778	0.3556	0.4444	0.5000	0.5222	0.2778	0.3167	0.3463	0.3602	0.3646
	LSTM	0.1889	0.3444	0.4556	0.4889	0.5889	0.1889	0.2667	0.3037	0.3120	0.3320
	NUESTRO	**0.3544**	**0.4937**	**0.5696**	**0.6076**	**0.7215**	**0.3111**	**0.3722**	**0.3944**	**0.4028**	**0.4228**
24	POP	**0.3537**	**0.4756**	0.5122	0.5366	0.5732	**0.3537**	**0.4146**	0.4268	0.4329	0.4402
	LSTM	0.2683	0.4024	0.5000	0.5976	0.6341	0.2683	0.3354	0.3679	0.3923	0.3996
	NUESTRO	**0.3537**	**0.4756**	**0.5488**	**0.6341**	**0.6707**	**0.3537**	**0.4146**	**0.4390**	**0.4604**	**0.4677**

Nota. Recuperado de Yunghui et al. (2019)

Con estos resultados los autores demuestran que su método obtiene mejores resultados que métodos comunes como es el caso del POP y LSTM. Además, que logran primero predecir la intención de compra de un usuario y después dependiendo de ello los ítems que podrían ser comprados.

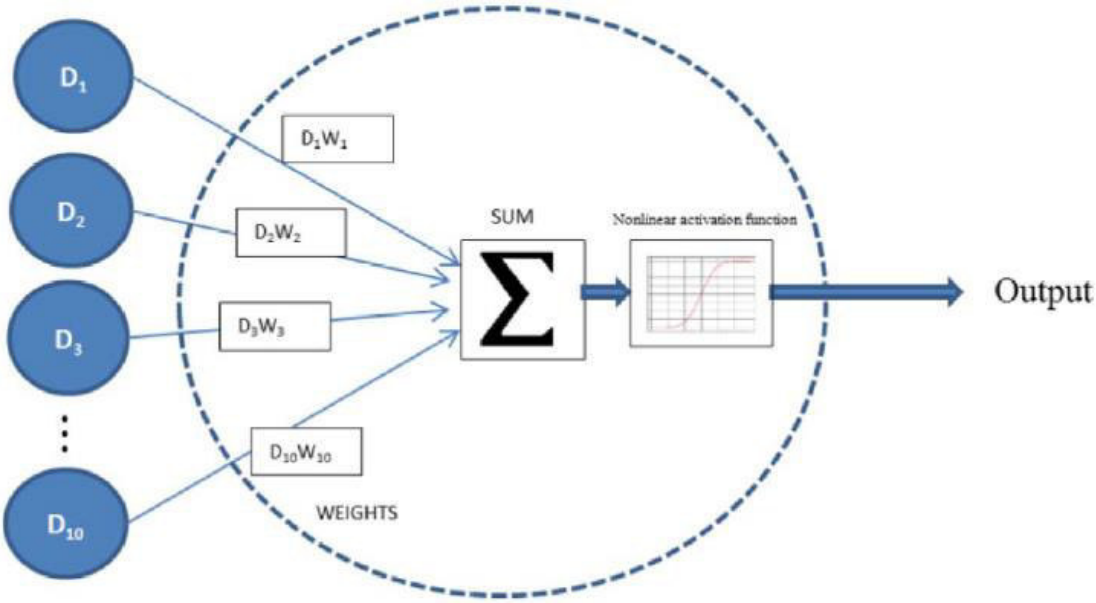
En Sujoy et al. (2019) los autores realizan una comparación de modelos lineal y no lineal para cada atributo que encuentran importante para la predicción de la intención de compra de un cliente. Para el modelo lineal utilizan regresión múltiple y para el no lineal utilizan una red neuronal GMDH. Esto marca una diferencia con las anteriores investigaciones donde decidían la cantidad de entradas según criterios propios. Luego de definir los datos, los autores dividen la metodología en 4 partes. La primera parte para el análizis de sentimiento donde inicialmente determinan si un texto es positivo, negativo o neutral para luego poder determinar la polaridad de un atributo calculando la diferencia entre los positivos con los negativos. La segunda parte para una minería de redes sociales esto debido a que los autores mencionan que las redes sociales influyen en la manera de pensar de los consumidores con respecto a la calidad de las marcas y su lealtad hacia ellas. Para esta parte los autores utilizan dos atributos, el lujo y el respeto al medioambiente, para determinar la puntuación de percepción social (SPS) que determinan como un atributo fuertemente relacionado con las emociones del consumidor. La tercera parte es utilizada para normalizar los datos obtenidos en 10 deciles iguales en una escala de 0-1. Finalmente, la cuarta parte es donde se desarrollan los métodos de predicción. El

análisis de regresión lineal múltiple fue construido con Minitab 17, un programa estadístico, donde los deciles actuaban como variables predictoras continuas. Por otra parte, para la red neuronal los autores optan por usar GMDH debido al problema de predecir los patrones de búsqueda asociados a una compra. En la red neuronal los deciles son las entradas para la red.

La estructura propuesta por los autores se puede ver en la Figura 8.

Figura 8

Estructura de la red neuronal



Nota. Recuperado de Sujoy et al (2019).

Una vez completado los dos modelos los autores empezaron con las pruebas para determinar el mejor modelo para los atributos, concluyendo en 3 puntos. El primero que esta información es útil para los *e-commerce*. Segundo que el modelo de predicción puede ayudar a los usuarios a tomar una decisión de compra más rápida y tercero que el *e-commerce* puede dar una plataforma de recomendación de productos a un menor costo.

Como se ha podido observar no existe un tipo de red neuronal preferida para la solución del problema de predicción, cada autor utilizó una red neuronal diferente. Por otra parte, también se puede observar los distintos algoritmos que se han utilizado para la mejora de la predicción ayudando a las redes en el proceso de aprendizaje y definición de las mismas.

En Chaudhuri et al. (2021) los autores proponen una red neuronal profunda para la predicción de la intención de compra de un usuario en la página web de un *e-commerce* utilizando tanto los datos demográficos del usuario como los datos de sesión que genera en su navegación. En su investigación los autores proponen que no solo son necesarios los datos de navegación sino que tambíen datos como la edad del usuario influyen en la intención de compra. Las características que utilizaron para la predicción se encuentran descritas en la Tabla 5.

Tabla 5

Estadísticos descriptivos de las variables utilizadas en esta investigación.

Variables	**Mínimo**	**Máximo**	**Promedio**	**Desviación estándar**
Atributos de interacción con la plataforma				
Hora de la sesión	1.23	23	14.48	4.37
Día de la semana de la sesión	1	7	-	-
Duración de la sesión (minutos)	0	8736.92	1415.71	1711.91
Número de inicios de sesión del cliente	1	6	2.37	0.81
Número de productos en los que se hizo clic	1	114.97	21.52	24.73
Precio más bajo entre los productos que se hizo clic	0	429.21	32.33	71.41
Precio más alto entre los productos que se hizo clic	0	882.78	96.13	149.85
Suma de precios de todos los productos en los que se hizo clic	0	8776.98	704.37	1122.17
Número de productos en el carrito	0	15.67	3.65	3.25
Precio más bajo del producto en el carrito	0	516.32	40.43	86.31
Precio más alto del producto en el carrito	0	643.37	68.17	108.73
Suma de precios de todos los productos en el carrito	0	1144.08	130.10	168.21

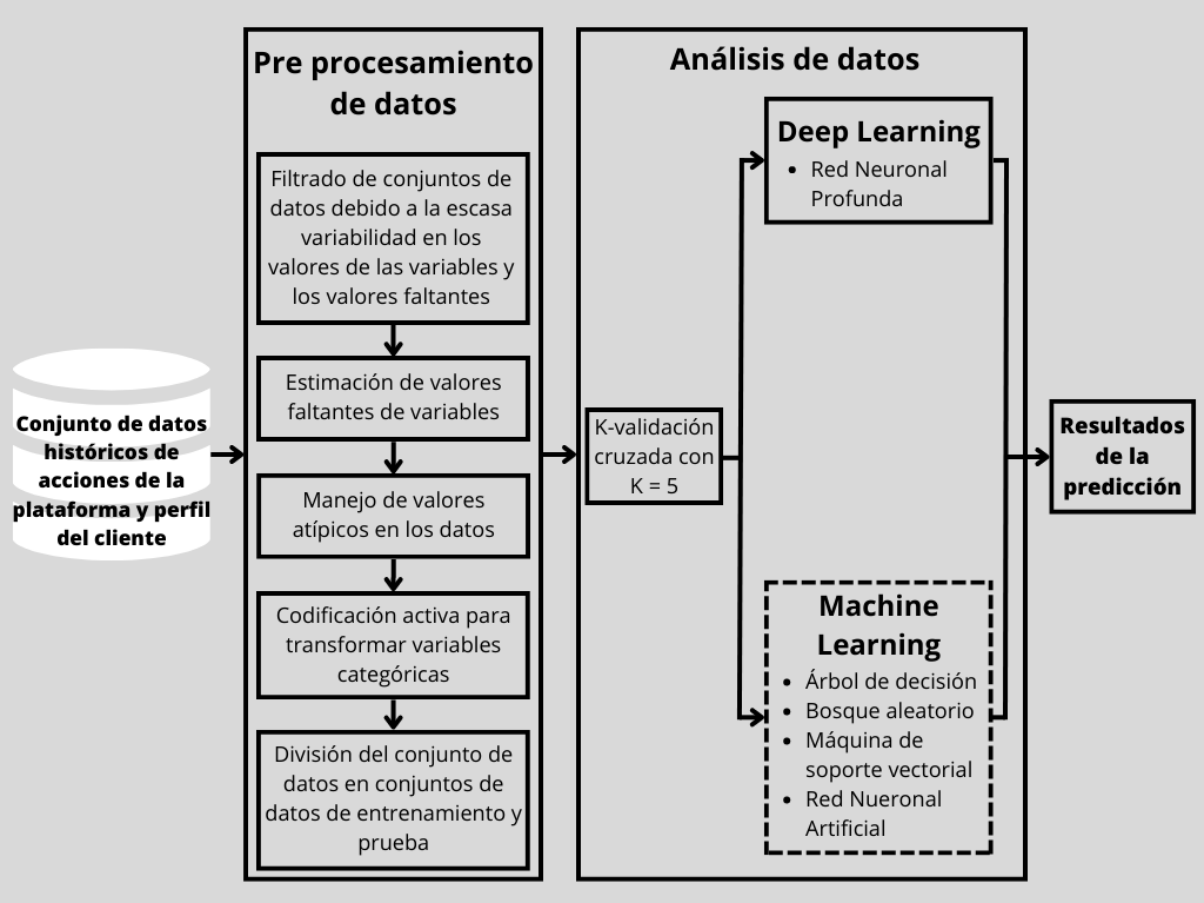
Atributos de los clientes				
Puntuación de la cuenta del cliente asignada por el minorista en línea	89.53	638	486.38	126.18
Duración de la cuenta del cliente (días)	0	602	135.56	108.32
Número de pagos realizados por el cliente	0	93.12	11.36	14.44
Edad del cliente	17	81.17	45.07	11.98
Género del cliente	1	2	-	-
Días transcurridos desde la última compra	3	738	79.88	97.61

Nota. Recuperado de Chaudhuri et al. (2021)

Para la construcción de su modelo primero los autores obtienen de la página web de un *e-commerce* de Alemania un *dataset* de 429013 sesiones las cuales pasaron por una etapa de pre procesamiento con el objetivo mejorar la precisión predictiva y la eficiencia computacional, en dicha etapa se eliminaron variables que tuvieron escasa variabilidad y alta cantidad de valores faltantes, se completaron valores faltantes de las sesiones y se aplicaron otros filtros para finalmente de los datos resultantes se divida entre el *dataset* de entrenamiento y el de test. En la Figura 9 se muestra el proceso que siguieron los autores para la construcción de su modelo.

Figura 9

El diagrama de proceso del estudio.



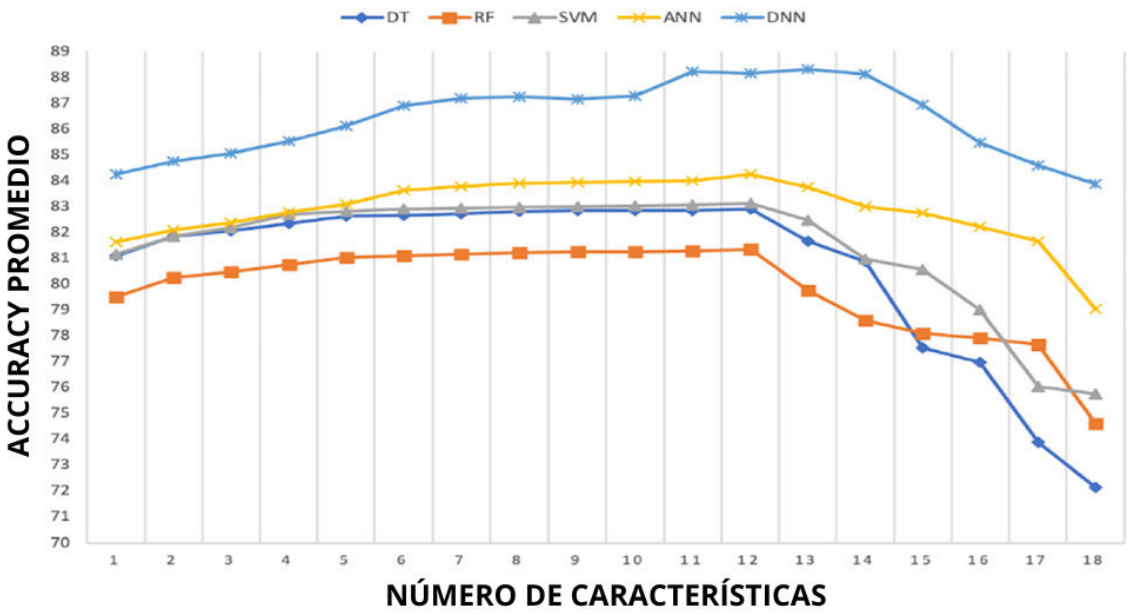
Nota. Recuperado de Chaudhuri et al. (2021)

Segundo para la construcción de su modelo de DNN los autores utilizaron el método de descenso de gradiente estocástico para entrenar su modelo realizando pruebas con distintos números de capas y distintos números de neuronas por capas, obteniendo al final el mejor resultado con una red de 5 capas y 128 neuronas por capa.

Finalmente los autores compararon su modelo de DNN con otros modelos de *machine learning* construidos, obteniendo un *accuracy* de 0.89 mayor frente a los demás modelos; concluyendo la utilidad de usar una DNN en la predicción de la intención de compra, además que concluyen que los atributos de interacción con la plataforma tienen un mayor impacto en la predicción de la intención de compra que los atributos de los clientes siendo la hora de la sesión y el día de la semana de la sesión las características que más influyeron en la predicción.

Figura 10

El cambio en los accuracies promedio de las técnicas de ML y DL con selección de características.



Nota. Recuperado de Chaudhuri et al. (2021)

Por otra parte, **las técnicas de machine learning** han ganado popularidad en los últimos años viendo que en investigaciones más recientes se opta por utilizar uno de estos modelos en vez de un modelo de red neuronal, se explican a continuación los encontrados en la literatura.

En Bichen & Bingwei (2018) los autores proponen un Extreme *Gradient Boosting Machine* para predecir la intención de compra de un usuario teniendo en cuenta las características de entropía de contenido. La metodología usada por los autores busca obtener un modelo conjunto más fuerte que los clasificadores atómicos usados en pasos anteriores con el objetivo de disminuir la función de costo. Los autores mencionan sobre la entropía que una baja entropía de contenido representa una sesión de navegación con menos variedad de contenido, en otras palabras de que el usuario está concentrándose en un tipo de contenido. Esto es muy útil para poder determinar las

intenciones que tiene el usuario para comprar o no los productos que seleccionó. Primero los autores analizan el *dataset* obtenido de un *e-commerce* europeo, obteniendo gráficas que muestran el comportamiento de los datos. Esto no afecta al momento de dividir la data para el entrenamiento y la validación, sino que es utilizado para poder determinar las características de los usuarios que influyen en la intención de compra. Adicional a esto los autores emplean la ingeniería de características para poder añadir a las características ya encontradas, otras relacionadas a la entropía del contenido. Una vez que tienen completa su tabla de características los autores pasan a desarrollar su modelo de *Extreme Gradient Boosting* para clasificar las sesiones.

El *dataset* de aprendizaje es usado para que la máquina pueda formar el modelo conjunto y esté listo para poder realizar las predicciones. Para el caso de la validación del modelo los autores comparan su método con otros métodos tradicionales los cuales son LR, GBM y AdaBoost. Los resultados de las pruebas se muestran en la Tabla 6.

Tabla 6

Resultados experimentales y comparaciones

Method	Accuracy	Precision	Recall	F_1score
XGBoost	91.19%	29.15%	41.81%	34.35%
LR	94.26%	29.57%	2.97%	5.40%
GBM	58.54%	10.36%	85.27%	18.48%
AdaBoost	94.50%	53.32%	1.74%	3.37%

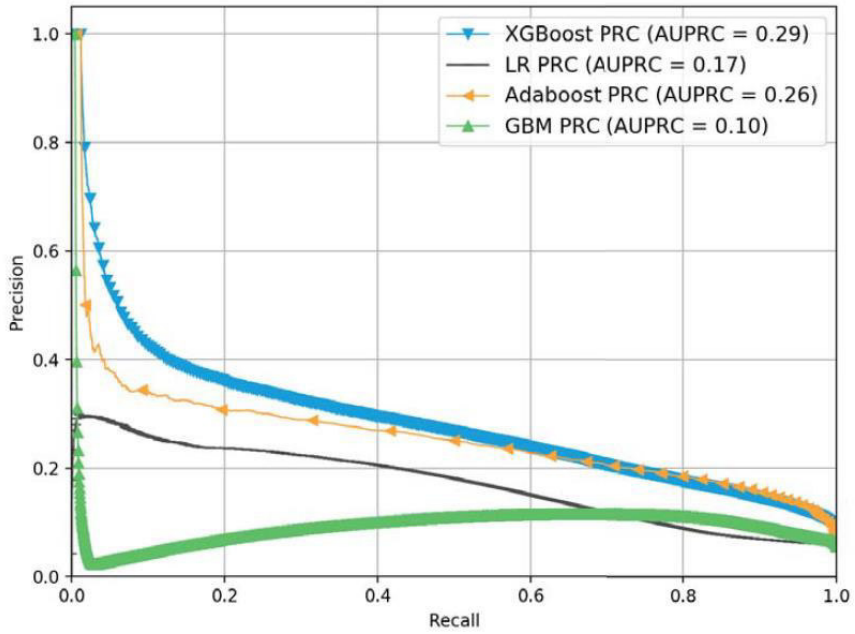
Nota. Recuperado de Bichen & Bingwei (2018)

Los autores usan cuatro indicadores para poder medir los distintos métodos. Ellos mencionan “Debido a la relación de desequilibrio entre las muestras compradoras y no compradoras, la precisión puede no ser un buen indicador del rendimiento del modelo.” Y mencionan que en los indicadores de Recall y F1 es donde su método demuestra su

fortaleza en comparación a los demás. A pesar de que GBM demuestra tener un mejor Recall los autores mencionan que la baja en la métrica Precision del modelo causaría costos innecesarios en los falsos positivos predichos. Por estos motivos los autores mencionan que su método es superior tomando en cuenta la curva Precision-Recall que se muestra en la Figura 11.

Figura 11

Curva Precision-Recall



Nota. Recuperado de Bichen & Bingwei (2018)

En Sharma & Prasanna (2019) proponen un Extreme Gradient Boosting Machine basado en la entropía de contenido. Esta investigación se concentra en hacer las predicciones de la intención de compra teniendo como *input* los lugares donde hizo clic el cliente y la entropía de contenido. Entre sus aportes se encuentra que refuerza la idea de la utilidad de la entropía de contenido para las predicciones de intención de compra, además hace una comparación entre su modelo desarrollado en XGBoost frente a GBM y AdaBoost, resultando el modelo del autor con mejores predicciones que los ya

mencionados. Tanto en esta investigación como en la anterior la métrica utilizada para medir los modelos fue el F1.

En Osnat et al (2019) los autores proponen un modelo de predicción para la intención de compra de un cliente anónimo teniendo en cuenta data obtenida al momento de la sesión y la tendencia del producto desde hace "n" días. Con clientes anónimos se refieren a clientes que no se han registrado en la página del *e-commerce* por lo tanto no existe un historial de acciones pasadas que haya realizado dicho cliente. Esta característica diferencia a este trabajo de otros al no usar data histórica del cliente.

Por otra parte, para la tendencia del producto los autores hacen un análisis de cuantos clientes dieron clic a un producto y cuantos de ellos lo compraron desde hace "n" días para tener un porcentaje de tendencia. Esta medida será utilizada al momento de la predicción debido a que los autores sostienen que los usuarios tienden a comprar los productos que se clasifican como populares según el porcentaje de tendencia análisado anteriormente.

La información utilizada para las pruebas fue obtenido de 2 *datasets*, el primer *dataset* es del *e-commerce* YooChoose RecSys y el otro es de Zalando. Además, las sesiones fueron separadas para solo tener las de usuarios anónimos, cabe recalcar que la información de un *dataset* fue trabajada de manera independiente del otro *dataset*. Lo primero que hicieron los autores fue obtener la tendencia de los productos. Para esto las sesiones se contabilizaron en tablas día x producto; fueron 2 las tablas que se manejó por *dataset*, una tabla para las sesiones donde los clientes realizaban la compra de sus productos y otra para las sesiones donde los clientes no realizaban una compra. Los datos que mostraban estas tablas eran la cantidad de veces que un producto se compraba o no en un día.

Con esta información se logró calcular la tendencia de un producto teniendo en cuenta "n" días. Luego con el análisis de estos datos se determinó las características que se tomaron en cuenta para la predicción de la intención de compra del usuario. Finalmente, debido a la característica del *dataset* donde la cantidad de sesiones de compra es mucho menor a la cantidad de sesiones de no compra, un problema podría ocurrir al momento del aprendizaje de la máquina predictora debido a la superioridad de casos de no compra. Para resolver este problema los autores usaron una técnica llamada SMOTE que sirve para que, estos casos de compra que se encuentran en menor cantidad puedan ser identificados y aprendidos correctamente por la máquina. Con respecto a los modelos de predicción, los autores hacen una comparación entre Logistic Regression, Bagging, NBTree, XGBoost y redes neuronales. Los datos de la comparación de los modelos en los *e-commerce* YooChoose y Zalando se muestran en la Tabla 7 y Tabla 8 respectivamente.

Tabla 7

YooChoose: Calidad de predicción en diferentes ventanas de tiempo.

Días	Características	Logistic	Bagging	NBTree	XGBoost
2 días	Con Tendencia	0.739	0.904	**0.916**	0.7853
	Sin Tendencia	0.733	0.886	**0.888**	0.765
3 días	Con Tendencia	0.72	0.886	**0.899**	0.798
	Sin Tendencia	0.71	0.854	**0.855**	0.76
4 días	Con Tendencia	0.7	0.882	**0.883**	0.8236
	Sin Tendencia	0.686	0.816	**0.817**	0.758
5 días	Con Tendencia	0.68	**0.889**	0.867	0.8559
	Sin Tendencia	0.664	**0.786**	**0.786**	0.75
6 días	Con Tendencia	0.677	0.899	0.873	**0.9**
	Sin Tendencia	0.65	**0.796**	0.795	0.7788

Nota. Recuperado de Osnat et al (2019).

Tabla 8

Zalando: Calidad de predicción en diferentes ventanas de tiempo.

Días	Características	Logistic	Bagging	NBTree	XGBoost
2 días	Con Tendencia	0.702	0.859	**0.905**	0.7554
	Sin Tendencia	0.701	**0.745**	0.731	0.7281
3 días	Con Tendencia	0.702	**0.892**	**0.892**	0.7543
	Sin Tendencia	0.701	**0.761**	0.735	0.7222
4 días	Con Tendencia	0.705	**0.859**	0.806	0.7861
	Sin Tendencia	0.7	**0.807**	0.762	0.7246
5 días	Con Tendencia	0.725	**0.876**	0.848	0.8681
	Sin Tendencia	0.718	**0.87**	0.825	0.7409
6 días	Con Tendencia	0.761	0.893	0.883	**0.9438**
	Sin Tendencia	0.755	**0.896**	0.875	0.786

Nota. Recuperado de Osnat et al (2019).

Donde demuestran que la herramienta de XGBoost con una ventana de 6 días obtiene resultados mejores que las otras metodologías. Para los resultados de la red neuronal los autores solo mencionan que obtienen valores de 0.8 y 0.84 para la predicción en YooChoose y Zalando respectivamente, siendo superior XGBoost a la red neuronal.

De este artículo se puede rescatar la comparación que realizan los autores entre XGBoost y otros métodos, inclusive una red neuronal.

En Qian et al (2020) los autores proponen un modelo de predicción para la intención de compra de un usuario en la plataforma de un *e-commerce*. En este caso se busca predecir después de la visita de un usuario a la página web si volverá y comprará un producto en un intervalo de 5 días o a lo máximo 15 días.

Para la extracción de información del comportamiento del usuario, los autores utilizan técnicas de *big data* y extraen información de 2 tipos de usuarios los que navegan en la PC y los que lo hacen a través del celular. La información obtenida se almacenó en tablas. Una para las características de los usuarios y otra para almacenar sus acciones.

Una vez almacenada esta información los autores utilizan la ingeniería de características para determinar características en el comportamiento de los usuarios. A

esto añaden un análisis de las características obtenidas retirando las que son típicas. Al final presentan una tabla con los 10 comportamientos más importantes del usuario y otra con los 10 comportamientos menos importantes.

Para el caso de la predicción los autores desarrollan dos modelos con la intención de compararlos. El primero es un modelo basado en LR el cual usa la información recabada. Este método fue entrenado y validado con la *data* obtenida tanto de usuarios que usan la PC como los que usan su celular demostrando que podían hacer las predicciones. El segundo era un modelo de predicción basado en XGBoost que es un clasificador de árbol de decisión, que se aproxima al valor verdadero con un grupo de K árboles de regresión. De la misma manera que el anterior este modelo es entrenado y validado pero solo con la data de usuarios que navegan con el celular.

Los resultados obtenidos para el primer y segundo modelo se pueden visualizar en las Tabla 9 y Tabla 10 respectivamente.

Tabla 9

Resultados del modelo base LR para usuarios de celulares

Ciclo de entrenamiento	Ciclo de testeo	Proporción de muestras positivas a muestras negativas	Accuracy	Recall	F1
9.10-9.14	9.15-9.19	1:78	0.2776	0.2331	0.2535
9.10-9.14	9.16-9.20	1:72	0.2481	0.2783	0.26005
9.10-9.14	9.17-9.21	1:56	0.2572	0.2551	0.2547
9.10-9.14	9.18-9.22	1:53	0.2432	0.2562	0.2498
9.10-9.14	9.19-9.23	1:73	0.2647	0.2296	0.2452
9.10-9.14	9.20-9.24	1:56	0.2683	0.2422	0.2561
9.10-9.14	9.21-9.25	1:76	0.2625	0.2402	0.2510

Nota. Recuperado de Bichen & Bingwei (2018)

Tabla 10

Resultados del modelo basado en XGBoost para usuarios de celulares

Ciclo de entrenamiento	Ciclo de testeo	Proporción de muestras positivas a muestras negativas	Accuracy	Recall	F1
9.10-9.14	9.15-9.19	1:77	0.3307	0.2296	0.2701
9.10-9.14	9.16-9.20	1:74	0.3036	0.2561	0.2769
9.10-9.14	9.17-9.21	1:57	0.3128	0.2427	0.2726

9.10-9.14	9.18-9.22	1:52	0.3148	0.2368	0.2677
9.10-9.14	9.19-9.23	1:74	0.3258	0.2349	0.2728
9.10-9.14	9.20-9.24	1:55	0.3325	0.2331	0.2759
9.10-9.14	9.21-9.25	1:75	0.2981	0.2438	0.2670

Nota. Recuperado de Bichen & Bingwei (2018)

Tomando en cuenta la métrica F1, los autores corroboran que el XGBoost obtiene mejores resultados de predicción que el LR, tomando ese método para su modelo.

En Bing & Yuliang (2017) los autores realizan una comparación entre el método de Naive Bayes y el método de árbol de decisión para la predicción de la intención de compra. Los autores deciden hacer esta comparación debido a que el método de clasificación Naive Bayes (NB) es un método muy común en machine learning conocido por su gran eficiencia y exactitud. Sin embargo mencionan que dicho método necesita que los atributos sigan una distribución de Gauss y esta condición no siempre se cumple.

Por este motivo los autores proponen un árbol de decisión C4.5 como solución al problema de distribución de la información.

Para poder comparar los métodos mencionados anteriormente los autores hacen uso del software WEKA, donde construyen ambos algoritmos con los parametros por defecto del software para poder asegurar la objetividad de la comparación.

Los autores realizan las pruebas donde los métodos deben clasificar a los clientes en con intención o sin intención de compra.

Tabla 11

Comparación del tiempo de modelado

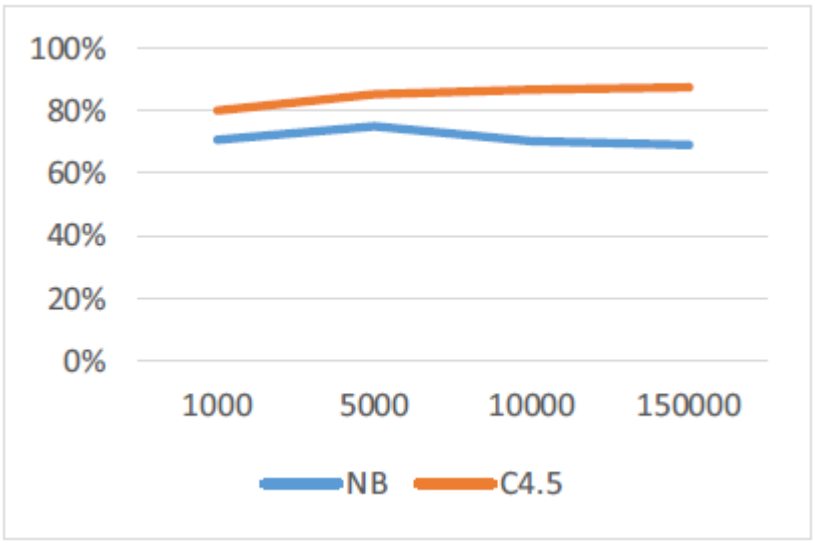
Method	NB	C4.5
S1-2	0.42s	0.34s
S1-3	0.41s	0.38s
S2-1	0.42s	0.36s
S2-3	0.37s	0.35s

S3-1	0.43s	0.35s
S3-2	0.41s	0.37s
Average	0.41s	0.36s

Nota. Recuperado de Bing & Yuliang (2017)

Figura 12

Gráfica de la comparación de la precisión



Nota. Recuperado de Bing & Yuliang (2017)

En la Tabla 11 se puede observar el tiempo que demoro entrenar cada método para cada prueba y al final de la tabla un valor promedio. Con estos resultados los autores demuestran que C4.5 tiene mayor eficiencia que NB.

Por otra parte, en la Figura 12 el eje de las X representa la cantidad de datos utilizados en las pruebas mientras que el eje Y representa la exactitud del modelo de predicción. De esto se puede observar que el C4.5 obtiene mayor exactitud. Además, se mantiene en su exactitud a pesar del aumento de datos, cosa que no sucede con NB donde pasado los 5000 datos empieza a disminuir su exactitud de predicción.

Con estos datos los autores demuestran la superioridad del árbol de decisión C4.5 frente al método de Naive Bayes en la predicción de la intención de compra.

En Ahmet & Mehmet (2019) los autores proponen un modelo de aprendizaje de clasificador binario para predecir la intención de compra del usuario antes de que deje la página.

Los datos obtenidos fueron a través de los flujos de clic, de estos datos los autores determinaron que el modelo solo iba a trabajar con datos de sesiones normales. Para poder separar los tipos de sesiones los autores usaron la extracción de propiedades, esto ayudó para la estimación del comportamiento de compra. Además de las acciones del usuario también los registros de la sesión fueron añadidos como por ejemplo el tiempo de duración de la sesión.

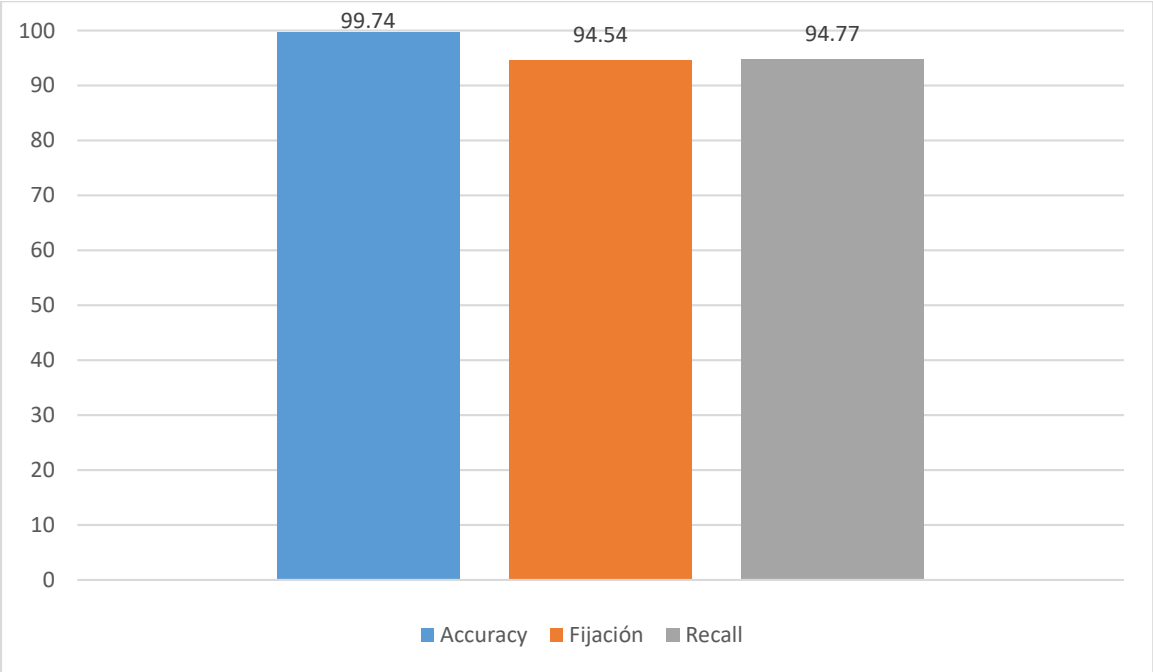
Después de todo este procesamiento cada sesión debe tener asociado un vector de propiedades que resume las características de las sesiones utilizando solo las más cruciales.

Las sesiones son distribuidas, basado en sus propiedades, en clústeres existentes con algoritmos *k means*. Aquellas sesiones que no puedan ser incluidos en un clúster son considerados contrarios, los cuales no serán usados por el modelo. Para la predicción de la intención de compra los autores crearon un modelo de aprendizaje de clasificador binario, que etiquetaba a una sesión con un 1 si se realizaba la compra o con un 0 caso contrario.

El modelo fue entrenado con los datos de sesiones de compra. Después de la fase de entrenamiento el modelo estaba listo para cuando aparezca una nueva sesión pueda estimar a que clase pertenece y poder etiquetarlos. Los autores crean un prototipo y prueban su metodología, mostrando sus resultados en la Figura 13.

Figura 13

Gráfica de los resultados de las pruebas.



Nota. Recuperado de Ahmet & Mehmet (2019)

Utilizando las métricas de accuracy, precisión y recall se pudo observar el rendimiento del prototipo. Afirmando que se pueden obtener resultados satisfactorios en la intención de compra.

En Bhattacharjee et al (2023) los autores proponen la aplicación de modelos de árboles de desición para predecir la intención de compra de celulares por parte de un cliente, además para determinar características influyentes en la compra de los clientes los autores proponen el uso de la técnica DEMATEL para determinar las características de compra relacionadas a los distintos segmentos de clientes. Cabe mencionar que a diferencia de otras investigaciones en esta los autores se centran en las características de los productos y no en las acciones del usuario.

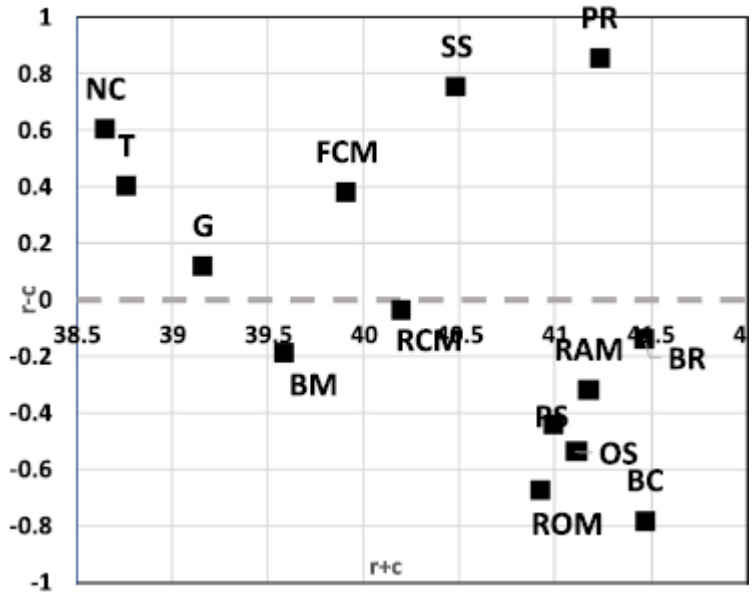
Primero para poder recolectar los datos necesarios, los autores hacen uso de la herramienta Google forms que facilita la creación de formularios y muestra estadísticas

de los resultados obtenidos en cada respuesta. El formulario que construyeron estubo dividido en dos partes la primera para obtener los datos biograficos del cliente tales como edad, sexo, etc y la segunda parte para obtener los datos acerca de las características de los celulares que prefieren. Con este formulario los autores obtuvieron 1107 datos que serán utilizados en el entrenamiento y testeo de los modelos predictivos implementados.

Segundo con los datos obtenidos los autores implementan el modelo DEMATEL para determinar la relación entre las características de los celulares, Tamaño de pantalla (SS), *Random access memory* (RAM), *Read-only memory* (ROM), Velocidad del procesador (PS), Grosor (T), Capacidad de almacenamiento de la batería (BC), Precio (PR), Agarre (G), Megapíxeles de la cámara frontal (FCM), Megapíxeles de la cámara trasera (RCM), Número de cámaras (NC), Material del cuerpo (BM) y Sistema operativo (OS), que utilizan como input para la predicción de la intención de compra de un celular. En Figura 14 podemos ver el diagrama resultante de utilizar DEMATEL sobre estas características.

Figura 14

Diagrama causa y efecto.



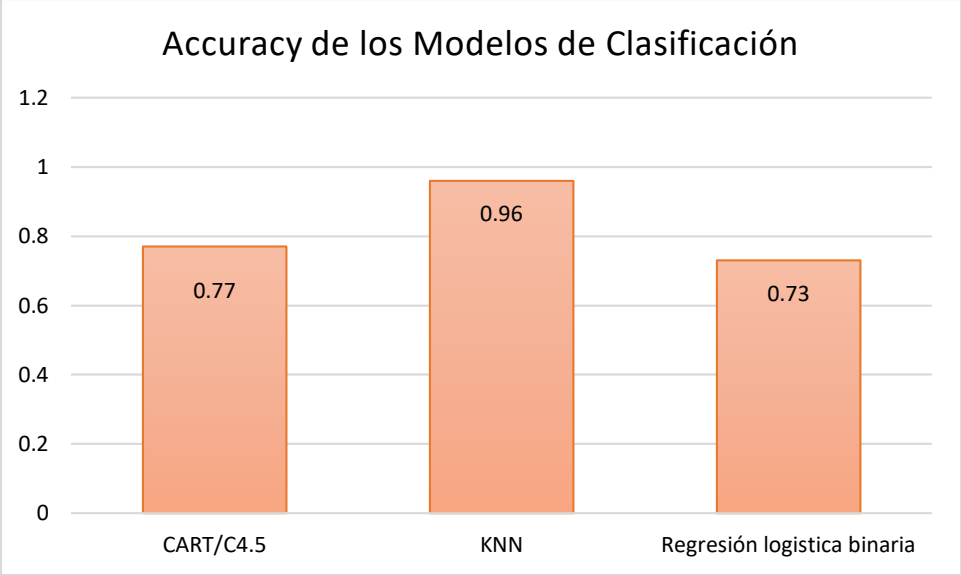
Nota. Recuperado y traducido de Thakkar (2021)

Donde $r + c$ replesenta el grado de importancia de cada criterio y $r - c$ representa si es un valor positivo que el criterio tiende a ser una causa en el grupo, de otra manera si es negativo el criterio tiende a ser un efecto en el grupo.

Finalmente los autores construyen los modelos predictivos CART, C4.5, K vecinos más próximos (KNN) y regresión logística binaria para comparar las predicciones de la inteción de compra tomando como input la matriz obtenida de implementar DEMATEL. En la Figura 15 podemos ver los resultados obtenidos en los distintos modelos según la métrica accuracy siendo KNN el que logró mejor puntuación.

Figura 15

Clasificación de la comparación de accuracy para la preferencia de compra de teléfonos inteligentes.



Nota. Recuperado y traducido de Thakkar (2021)

Con esta investigación los autores logran encontrar caráterísticas que son influyentes al momento de comprar un celular, además que logran implementar un modelo predictivo, en este caso KNN, que obtiene buenas predicciones de la intención de compra con un accuracy del 0.96.

De todos las técnicas encontradas en la bibliografía se decide que en la construcción del sistema predictor se empleará 3 modelos predictivos Extreme Gradient Boosting Machine, AdaBoost y Bagging para la predicción del abandono del carrito de compras. Estos modelos realizarán su predicción de forma independiente al resto y finalmente el sistema predictor determinará el abandono o no según lo que prediga la mayoría.

Se decidió utilizar principalmente la técnica de *Extreme Gradient Boosting Machine* ya que se ha visto que en los últimos años es muy usada, debido a las buenas predicciones de la intención de compra que logra. Esto es mencionado por distintos autores en las conclusiones de sus trabajos: con los algoritmos basados en XGBoost, el sistema de predicción de la intención de compra demuestra su fortaleza para identificar correctamente las actividades de navegación con potencial de compra (Bichen &

Bingwei, 2018); se descubrió que el modelo XGBoost es más efectivo a través de un experimento con usuarios de ECP que usan teléfonos celulares (Qian et al., 2020); y en Osnat et al (2019) se puede observar una comparación entre modelos donde el modelo del autor, XGBoost con el algoritmo SMOTE para el aprendizaje, logra obtener mejores predicciones superando inclusive a una red neuronal. Por otro lado AdaBosst y bagging han sido fuertes competidores con la técnica mencionada anteriormente obteniendo buenas predicciones en las comparaciones realizadas por los autores ya mencionados.

3.2.2.Q2: ¿Qué características se utilizan para la predicción de la intención de compra de un cliente?

En los artículos revisados de la Tabla 1, no se encontró un conjunto de características uniforme que se utilicen en todas las predicciones de la intención de compra sin embargo se encontraron 3 características que se repiten en la mayoría de artículos, estas características son la cantidad de clics realizados por el usuario, el tiempo que permaneció en la página del e-commerce y la fecha en que el usuario entró a la página. Las otras características utilizadas en la bibliografía cambiaban dependiendo el enfoque que cada autor le daba a su investigación y no mencionaban la utilización de alguna herramienta para poder obtener estas características, por ese motivo en esta investigación aparte de las 3 características que se repiten se realizará un análisis de datos para poder encontrar otras características que influyen en la intención de compra.

3.2.3.Q3: ¿Qué técnicas se emplean en la obtención de los datos a utilizar en el entrenamiento de los modelos predictivos?

Para la obtención de información en la bibliografía revisada de la Tabla 1 no se especifica algún método o herramienta a utilizar para poder captar los datos de los usuarios, por ese motivo la obtención de datos dependerá de las facilidades y

herramientas que nos proporcione el e-commerce sobre su página web. Además, se aplicará la ingeniería de características para optimizar los datos de entrada de los modelos predictivos, esto debido a que se encontró en Bichen & Bingwei (2018) y en Qian et al (2020) que utilizaban este proceso para mejorar el entrenamiento de sus modelos predictivos y obtener mejores predicciones.

Capítulo 4: Sistema predictor en línea de abandono del carrito de compras

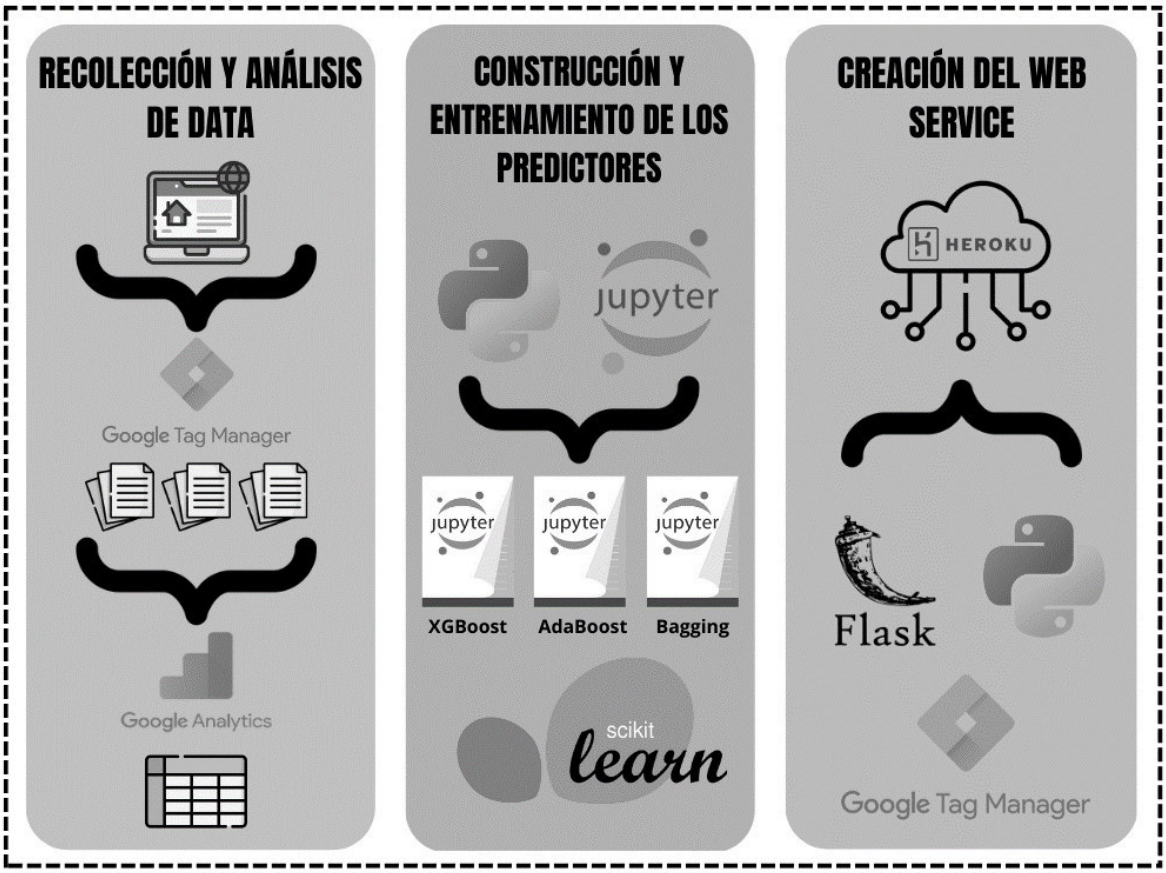
En el siguiente capítulo se muestra las herramientas, algoritmos de *machine learning* y metodologías a utilizar en la construcción del sistema predictor para la predicción en línea del abandono de carrito de compras.

4.1.Sistema predictor propuesto

Para la construcción del sistema predictor se definen 3 fases que abarcan desde la obtención de la información hasta el despliegue del servicio de predicción de abandono del carrito de compras para que la página del *e-commerce* pueda realizar consultas en tiempo real. En la Figura 16 se muestra las fases definidas y el proceso a seguir en cada una de ellas.

Figura 16

Proceso de construcción del sistema predictor



La primera fase es la de "Recolección y Análisis de Data", el objetivo de esta fase es identificar las características que influyen en el abandono del carrito de compras y obtener el conjunto de datos que será utilizado en la construcción y entrenamiento de los modelos predictivos.

Con este objetivo en mente se plantea obtener la información de las sesiones de los usuarios de la página y a través de un análisis de los datos determinar las características más resaltantes que ayuden a representar el comportamiento del usuario. Para ello se utilizará el Google Tag Manager y Google Analytics que nos proporciona el *e-comerce* para capturar y analizar la data de las sesiones de los usuarios con la intención de determinar las características que influyen en el abandono del carrito de compras. Al término del análisis las características seleccionadas fueron número de clics, fecha, el tiempo que permaneció, cantidad de productos seleccionados, logeado, el monto total a pagar y el descuento

La segunda fase es la de "Construcción y Entrenamiento de los Predictores", el objetivo de esta fase es tener entrenados los 3 modelos predictores XGboost, Adaboost y bagging, los cuales serán implementados con el lenguaje de programación *Python,* que serán usados en el servicio de predicción de abandono del carrito de compras. Primero para poder tener un mejor entrenamiento de los predictores se empezó esta fase con la optimización de los datos de entrada a través de un proceso de ingeniería de características, en este proceso se completará datos vacíos y se eliminará valores extremos que influyen negativamente en el entrenamiento de los predictores. Después con los datos optimizados se pasará al entrenamiento de los predictores para que al final se puedan serializar, guardar las implementaciones en archivos, de forma que se puedan reutilizar en el servicio de predicción sin necesidad de tener que volver a entrenarlos.

La tercera y última fase es la de "Creación del Web Service", el objetivo de esta fase es la construcción y despliegue el servicio web de predicción de abandono de carrito de compras en el ambiente de Heroku para que pueda ser consultado en todo momento por la página web del *e-commerce*. Para esto el servicio contará con la implementación de los 3 modelos predictivos de la fase anterior para predecir, a través de un proceso de votación, si el cliente abandonará o no su carrito de compras. Una vez se tenga el programa codificado se subirá a la nube para que se le pueda hacer consultas de predicción en tiempo real, además para establecer la comunicación entre la página del *e-commerce* y el *web service* se utilizará la herramienta de Google Tag Manger.

Todo este proceso de construcción se desarrolló bajo el marco de trabajo *Scrum* donde se implementó cada fase del proceso en distintos *sprints*, de manera que en continua comunicación con el *e-commerce* se fue desarrollando todo el sistema. Los *sprints* tuvieron una duración de 2 semanas y al final de cada *sprint* se mostraba los avances realizados tanto en análisis, documentación o funcionalidad del sistema dependiendo de la fase en la que se encuentre.

Capítulo 5: Implementación del sistema predictivo

En el siguiente capítulo se muestra a detalle el proceso realizado para la construcción y entrenamiento del sistema predictor anteriormente definido.

5.1.Metodología implementada

Para implementar la metodología *scrum* primero se definió un *product backlog* inicial con las tareas necesarias para lograr la funcionalidad de predicción del abandono del carrito de compras en tiempo real, sin embargo este fue cambiando con el paso de los *sprints* para poder acomodarse a todas las necesidades. En la Tabla 12 se muestra el *product backlog* final que se tuvo a lo largo de la construcción del sistema.

Tabla 12

Product Backlog

Código	Título	Descripción
T01	Informe personalizado de abandono de carrito de compras.	Creación de un informe personalizado de las características más influyentes en el abandono del carrito de compras de sus clientes.
T02	Almacenamiento de características influyentes	Se debe almacenar los datos de las sesiones de los clientes a través de etiquetas para poder ser mostrados en el informe de abandono de carrito.
T03	Limpieza y optimización de información	Se debe eliminar la fila de datos donde se encuentren valores nulos, extremos o duplicados para optimizar el conjunto de datos de entrada.
T04	Implementación del modelo XGBoost	Se debe generar el archivo “ipynb” con el código para el entrenamiento del modelo predictivo XGBoost que tenga como entrada las características influyentes del abandono de carrito y como salida la predicción del abandono.
T05	Testear y serializar el modelo predictivo XGBoost	Probar y validar el correcto funcionamiento del predictor XGBoost para serializarlo con sus configuraciones establecidas en un archivo “pkl”.
T06	Implementación del modelo AdaBoost	Se debe generar el archivo “ipynb” con el código para el entrenamiento del modelo predictivo AdaBoost que tenga como entrada las características influyentes del abandono de carrito y como salida la predicción del abandono.

T07	Testear y serializar el modelo predictivo AdaBoost	Probar y validar el correcto funcionamiento del predictor AdaBoost para serializarlo con sus configuraciones establecidas en un archivo “pkl”.
T08	Implementación del modelo bagging	Se debe generar el archivo “ipynb” con el código para el entrenamiento del modelo predictivo bagging que tenga como entrada las características influyentes del abandono de carrito y como salida la predicción del abandono.
T09	Testear y serializar el modelo predictivo bagging	Probar y validar el correcto funcionamiento del predictor bagging para serializarlo con sus configuraciones establecidas en un archivo “pkl”.
T10	Optimizar la búsqueda de parámetros ideales en el modelo XGBoost	Mejorar la búsqueda de parámetros ideales del modelo predictivo XGBoost automatizando la combinación de valores de los parámetros del modelo para obtener de manera óptima la mejor configuración de parámetros, dentro de un grupo de valores dados.
T11	Optimizar la búsqueda de parámetros ideales en el modelo AdaBoost.	Mejorar la búsqueda de parámetros ideales del modelo predictivo AdaBoost automatizando la combinación de valores de los parámetros del modelo para obtener de manera óptima la mejor configuración de parámetros, dentro de un grupo de valores dados.
T12	Optimizar la búsqueda de parámetros ideales en el modelo bagging	Mejorar la búsqueda de parámetros ideales del modelo predictivo bagging automatizando la combinación de valores de los parámetros del modelo para obtener de manera óptima la mejor configuración de parámetros, dentro de un grupo de valores dados.
T13	Creación del servicio web de predicción	Se debe crear un nuevo servicio en Python que a través de una consulta a la ruta “/prediction” con los datos de sesión del cliente como entrada, obtenga como respuesta la predicción del abandono del carrito de compras utilizando el modelo XGBoost serializado.
T14	Desplegar servicio web en la nube	Se debe desplegar el servicio web de predicción creado a la nube para que pueda ser consultado en tiempo real.
T15	Agregar los modelos AdaBoost y bagging; y actualizar la lógica de predicción	Se debe actualizar la lógica del servicio predicción para utilizar los modelos serializados AdaBoost y bagging en la predicción del abandono del carrito de compras.

T16	Testeo del sistema predictivo	Testear con data real el sistema predictivo compuesto por los 3 modelos predictivos serializados.
T17	Capturar los datos de la sesión del cliente	Generar etiquetas y variables en Google Tag Manager que calculen y almacenen los datos de sesión necesarios para la predicción de abandono.
T18	Comunicar el sistema predictivo con la página de la tienda	Crear una etiqueta predicción en Google Tag Manager que se encargue de enviar los datos de sesión al sistema predictor y salvar la respuesta de abandono.
T19	Crear activador de predicción de abandono de carrito	Crear un activador en Google Tag Manager que lance la etiqueta predicción cuando el cliente entre al carrito de compras.
T20	Re factorizar lógica de activador de predicción	Re factorizar la lógica del activador para que lance la etiqueta predicción a los 4, 10, 20, 30 y 40min de permanencia del cliente en la página web.

Una vez definido las tareas iniciales se pasó a planificar los *sprints* los cuales tuvieron una duración de 2 semanas. A cada *sprint* se le fue asignado un conjunto de tareas del *product backlog* para su desarrollo y en caso no se terminara la tarea hasta el final del *sprint* esta se movería para el siguiente *sprint*.

En total se planificó 8 *sprints* para el desarrollo del sistema predictor, la disposición de las tareas asignadas a cada *sprint* se muestra en la Tabla 13.

Tabla 13

Planificación de los sprints

Número de Sprint	Objetivo del Sprint	Tareas del Sprint
Sprint 1	Determinar las características influyentes en la predicción del abandono de carrito de compras y obtener un conjunto de datos de prueba.	T1 T2 T3
Sprint 2	Implementación del modelo predictivo XGBoost en Python. Entrenamiento, testeo y serialización del predictor.	T4 T5
Sprint 3	Construcción y despliegue del servicio web de predicción.	T13 T14
Sprint 4		T6

	Implementación del modelo predictivo AdaBoost en Python. Entrenamiento, testeo y serialización del predictor.	T7
Sprint 5	Implementación del modelo predictivo bagging en Python. Entrenamiento, testeo y serialización del predictor.	T8 T9
Sprint 6	Mejorar la predicción del abandono de carrito de compras agregando los modelos predictivos AdaBoost y bagging al sistema, además optimizar los entrenamientos de los modelos.	T15 T10 T11 T12
Sprint 7	Actualizar la lógica de la página web para comunicarse con el sistema predictor y guardar la predicción del abandono de carrito de compras.	T17 T18 T19
Sprint 8	Testeo con data real del sistema predictor desarrollado.	T16 T20

Como se muestra en la planificación las tareas están direccionadas a ir avanzando a través de tres fases en el proceso de construcción: Recolección y Análisis de Datos, Construcción y entrenamiento de los predictores, y Creación del web service las cuales se explicarán a mayor detalle en los siguientes puntos. Una vez pasado por estas fases se vuelve a iniciar otro ciclo con la intención de mejorar la predicción ya realizada por el sistema predictor.

5.2.Desarrollo de la propuesta

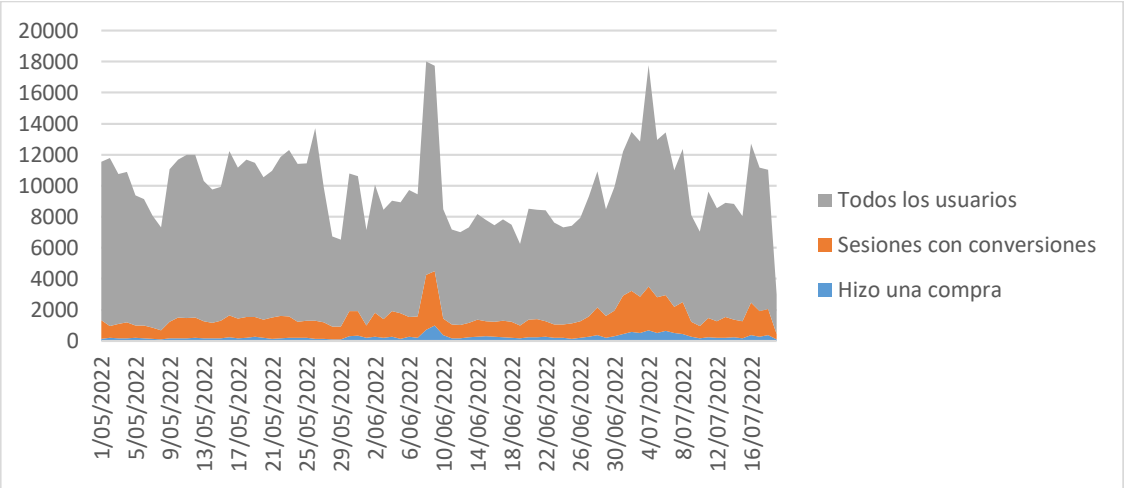
A continuación se muestran las fases mencionadas anteriormente indicando a más detalle el proceso seguido en cada fase y los softwares utilizados.

5.2.1.Recolección y análisis de los datos

En esta etapa se hizo el tratamiento de la data para poder obtener las características que influyen en la intención de compra de un cliente. Para ello se obtuvo información de las sesiones de los clientes de la página web de un *e-commerce* que se dedica a la venta de libros. En la Figura 17 se muestra la cantidad de usuarios que entran a la página y cuántos de ellos realizan una compra.

Figura 17

Gráfico de área de usuarios en la página entre mayo y julio



A diferencia de la obtención directa de datos que ocurre en los artículos revisados, en este caso se necesitará hacer un paso intermedio debido a que la empresa no dará acceso a esta información sino que dará acceso al Google Analytics y Google Tag Manager relacionado a su página web. Por dicho motivo se adaptará este paso de la siguiente forma, primero se utilizará el tag manager para a través de etiquetas y activadores poder capturar información sobre la sesión del cliente. Finalmente, la información obtenida se volcará sobre el analytics para poder visualizar el comportamiento de los usuarios y determinar qué características influyen en su decisión de abandono del carrito de compras.

Con la información obtenida de los usuarios que visitaron la página entre los meses de mayo y julio se destacaron las siguientes características: número de clics, fecha, el tiempo que permaneció, cantidad de productos seleccionados, logeado, el monto total a pagar y el descuento. En la Tabla 14 se muestra una descripción de las características seleccionadas y en la Tabla 15 se muestra como se almaceno la data de las sesiones.

Tabla 14

Características influyentes en la predicción de abandono del carrito de compras

Nº	Característica	Descripción
1	Número de Clics	Cantidad de clics realizados antes de la predicción.
2	Fecha	Fecha, en formato día mes año, en el que el cliente entro a la página.
3	Tiempo que permaneció	Tiempo expresado en segundos que se mantuvo el cliente dentro de la página web.
4	Cantidad de productos seleccionados	Cantidad de productos añadidos al carrito de compras antes de realizar la predicción.
5	Logeado	Bandera para determinar si el cliente había iniciado sesión en la página web.
6	Monto	El monto total de los productos añadidos al carrito de compras.
7	Descuento	Porcentaje de productos añadidos al carrito de compras que se encuentran con descuento.
8	Compra	Cantidad de productos comprados por el cliente.

Tabla 15

Datos de las sesiones de los clientes

Id Sesión	Numero Clics	Fecha	Tiempo que permaneció	Cantidad de productos seleccionados	Logeado	Monto	Descuento	compra
GA123458	40	20/05/2020	40	5	0	575.59	1	0
GA784612	80	30/05/2020	60	1	0	220.0	0	1
GA159263	15	15/06/2020	10	2	1	120.0	0.50	2
GA154864	90	15/06/2020	60	5	0	135.50	0.2	0

Todas las características pasaron a ser almacenadas bajo su mismo nombre en una tabla a excepción de la fecha que se dividió en día y mes, donde el día está en un intervalo

de 0-6 empezando desde el domingo, y el mes del 1-12. Además, los valores de la característica de compra se transformaron en solo 0 y 1 donde el valor es 0 si el cliente compró al menos un producto o 1 si el cliente abandono su carrito y no compro ningún producto.

Las características “Número de clics”, “Fecha” y “Tiempo que permaneció” utilizadas fueron escogidas debido a su constante mención en los artículos revisados; y las características “Cantidad de productos seleccionados”, “Logeado”, “Monto”, “Descuento” fue añadida debido al consejo de profesionales en marketing y ventas de la empresa analizada, además que se encontró que los clientes que están logeados o tienen productos con descuentos en su carrito de compras son más propensos a concretar la compra de sus productos a diferencia de otros.

Luego de haber almacenado los datos de las sesiones en la tabla se realizó un análisis de la data para determinar si se encontraba balanceada o no, descubriendo que se presentaba un desbalance debido a la gran cantidad de sesiones donde se abandonaba el carrito de compras a comparación de las que no viéndose necesario aplicar la técnica SMOTE encontrada en la bibliografía para el balanceo de data.

5.2.2.Construcción y entrenamiento de los predictores

En esta etapa se explicará las distintas librerías que se usaron para la construcción y entrenamiento de los modelos predictivos.

Cabe mencionar primero que al *dataset* obtenido se tuvo que aplicar una ingeniería de características eliminando registros incompletos o con valores extremos, campos con valores demasiado grandes o pequeños que se repiten muy pocas veces, que afectan negativamente en el entrenamiento de los predictores. El *dataset* resultante de esta

operación fue de 55440 sesiones donde el 75% será para el aprendizaje y el otro 25% para el testeo de los modelos.

Segundo que en esta etapa se utilizó el lenguaje de programación Python para el entrenamiento de los modelos predictivos.

Finalmente es importante mencionar que en el entrenamiento de los predictores se utilizó GridSearchCV de la librería sklearn para optimizar la búsqueda de los parámetros ideales para cada modelo, ya que permite definir un conjunto de valores para cada parámetro del modelo y entrenar tantos modelos como combinaciones entre valores existan dando como resultado el mejor modelo según la métrica (accuracy, recall, precisión o F1) que se defina; además, nos permite realizar validación cruzada.

De los 3 modelos a implementar **primero** para la construcción del extreme gradient boosting machine se utilizó la librería XGBoost, una librería de código abierto que proporciona una serie de parámetros y métodos para poder implementar esta técnica. Para empezar con el entrenamiento se especificó a xgboost los siguientes parámetros: la data de entrenamiento como input, las rondas de parada temprana, esto significa que si en "n" iteraciones el valor de la función objeto no mejora entonces se da por terminado el aprendizaje del modelo; y el tipo de clasificación que realizó que en este caso es una clasificación binaria. Por otra parte también se definió los parámetros cuyos valores se pretende encontrar los cuales son:

- Límite de árboles a crear en el *random forest*, para que una vez pasada esta cantidad termine la iteración.
- La función objeto, en este caso se puso "LogLoss" el cual castiga al modelo cuando realiza una predicción errónea, por lo tanto el modelo busca minimizar esta función; y "error" que calcula la tasa de error de clasificación binaria.

- Taza de aprendizaje del modelo.
- Profundidad de los árboles.

Estos últimos parámetros en el entrenamiento fueron cambiando según los rangos de valores ingresados hasta que termine el entrenamiento y se obtuvo la combinación de parámetros con la mejor predicción.

Figura 18

Código para el entrenamiento del modelo XGBoost

XGBoost

```
from sklearn.model_selection import GridSearchCV
xgb = xgboost.XGBClassifier()
parametros={
    'objective':['binary:logistic'],
    'learning_rate':[0.3,0.5,0.6,0.8,1,1.2],
    'n_estimators':[100,150,200,300,400,500,600],
    'max_depth':[4,6,8,10,12,14,16],
    'eval_metric':['error','logloss']
}
fit_parametros = {
    'early_stopping_rounds':10,
    'eval_set':[(X_test1,Y_test1)]
}
clf = GridSearchCV(xgb,parametros,scoring='recall')
clf.fit(X_resampled,Y_resampled,**fit_parametros)
```

```
[0]     validation_0-error:0.50640
Will train until validation_0-error hasn't improved in 10 rounds.
[1]     validation_0-error:0.45276
[2]     validation_0-error:0.50838
[3]     validation_0-error:0.50076
[4]     validation_0-error:0.49604
[5]     validation_0-error:0.49741
[6]     validation_0-error:0.46144
[7]     validation_0-error:0.45078
[8]     validation_0-error:0.45245
[9]     validation_0-error:0.44743
[10]    validation_0-error:0.45261
[11]    validation_0-error:0.44956
[12]    validation_0-error:0.44392
[13]    validation_0-error:0.44240
[14]    validation_0-error:0.43371
[15]    validation_0-error:0.43737
[16]    validation_0-error:0.43737
[17]    validation_0-error:0.43554
```

Segundo, para la construcción de adaboost se utilizó AdaBoostClassifier y DecisionTreeClassifier de la librería sklearn. Para empezar con el entrenamiento se necesitó especificar el estimador base, que es el modelo inicial que utilizó adabost para ir mejorando la predicción, en este caso fue un árbol de decisión. Después se especificó los parámetros cuyos valores se pretende encontrar:

- La taza de aprendizaje

- El número de estimadores, que es el límite de estimadores creados para incrementar la predicción. Una vez superado este límite termina el entrenamiento del modelo.
- Profundidad del árbol de decisión.

Igual que el anterior modelo se definió un rango de valores para estos últimos parámetros y se empezó el entrenamiento.

Figura 19

Código para el entrenamiento del modelo AdaBoost

AdaBoost

```
adaboost = AdaBoostClassifier()
tree1 = DecisionTreeClassifier(max_depth=6)
tree2 = DecisionTreeClassifier(max_depth=8)
parametros={
    'base_estimator':[tree1,tree2],
    'learning_rate':[0.05,0.3,0.5],
    'n_estimators':[200,250,300],
}
clf = GridSearchCV(adaboost,parametros,scoring='recall')
clf.fit(X_resampled,Y_resampled)
```

```
GridSearchCV(estimator=AdaBoostClassifier(),
             param_grid={'base_estimator': [DecisionTreeClassifier(max_depth=6),
                                            DecisionTreeClassifier(max_depth=8)],
                         'learning_rate': [0.05, 0.3, 0.5],
                         'n_estimators': [200, 250, 300]},
             scoring='recall')
```

Tercero, para la construcción de bagging se utilizó BaggingClassifier y DecisionTreeClassifier de la librería de sklearn. Al igual que el anterior modelo se definió el estimador base que utilizó el bagging como un árbol de decisión y posteriormente se definió los parámetros cuyos valores se pretende encontrar los cuales son:

- El número de estimadores
- La profundidad del árbol de decisión.

Igualmente se definió un rango de valores para cada parámetro y se empezó el entrenamiento.

Figura 20

Código para el entrenamiento del modelo Bagging

Bagging

```
bagging = BaggingClassifier()
tree1 = DecisionTreeClassifier(max_depth=4)
tree2 = DecisionTreeClassifier(max_depth=6)
tree3 = DecisionTreeClassifier(max_depth=8)
tree4 = DecisionTreeClassifier(max_depth=10)
tree5 = DecisionTreeClassifier(max_depth=12)
tree6 = DecisionTreeClassifier(max_depth=14)
tree7 = DecisionTreeClassifier(max_depth=16)
parametros = {
    'base_estimator':[tree1,tree2,tree3,tree4,tree5,tree6,tree7],
    'n_estimators':[5,10,15,20,30,100,150,200,300,400]
}
clf = GridSearchCV(bagging,parametros,scoring='recall')
clf.fit(X_resampled,Y_resampled)
```

```
GridSearchCV(estimator=BaggingClassifier(),
             param_grid={'base_estimator': [DecisionTreeClassifier(max_depth=4),
                                            DecisionTreeClassifier(max_depth=6),
                                            DecisionTreeClassifier(max_depth=8),
                                            DecisionTreeClassifier(max_depth=10),
                                            DecisionTreeClassifier(max_depth=12),
                                            DecisionTreeClassifier(max_depth=14),
                                            DecisionTreeClassifier(max_depth=16)],
                         'n_estimators': [5, 10, 15, 20, 30, 100, 150, 200, 300,
                                          400]},
             scoring='recall')
```

Finalmente, los valores obtenidos en las métricas de recall y F1 para los modelos entrenados son mostrados en la Tabla 16.

Tabla 16

Valores de las métricas obtenidas en el entrenamiento de los modelos

Modelos	Recall	F1
XGBoost	0.754	0.77
AdaBoost	0.916	0.85
Bagging	0.796	0.88

5.2.3.Creación del Web Service

Para que los predictores puedan hacer las predicciones en línea es necesario que puedan ser consultados por la página web del *e-commerce* en tiempo real. Para ello se desarrolló un *web service* que ofrezca el servicio de predicción de abandono de carrito de compras a la página web. Dicho servicio web recibe los datos de sesión del cliente, utiliza los

modelos entrenados para realizar la predicción y devuelve a la página la predicción de abandono del carrito de compras. Es necesario mencionar que el servicio web desarrollado presentará una interfaz de usuario sencilla debido a que no va a ser consultado por los usuarios finales que son los clientes sino por la página web del *e-commerce*. Esta interfaz fue desarrollada por motivos de documentación y pruebas del *endpoint* de predicción y se implementó a través de la herramienta *swagger-ui*, herramienta que autogenera una interfaz con la documentación de las capacidades del servicio, para que el *e-commerce* tenga la información necesaria de que datos debe enviar, que datos va a recibir, a que url apuntar entre otros datos para poder realizar consultas al servicio web de predicción de abandono de forma exitosa.

Para el desarrollo del servicio web se utilizó flask, un *framework* escrito en Python que permite crear aplicaciones web rápidamente, esto facilitó la implementación del servicio web de predicción de abandono. Una vez terminada la etapa de construcción y entrenamiento de los predictores se serializó los modelos entrenados con sus parámetros finales para después ser invocados en el servicio web.

El servicio web fue implementado de forma que envíe los datos de sesión obtenidos por la página web a cada predictor serializado para que realice su predicción y al final para determinar la predicción del sistema se toma en cuenta lo que la mayoría de predictores decide. Este cálculo se obtiene a través de la suma de los resultados obtenidos por cada predictor, siendo los posibles resultados 0 si no hay abandono del carrito de compras y 1 si hay abandono del carrito, si la mayoría predice que hay abandono entonces la suma será mayor o igual a 2 caso contrario significará que la mayoría predice que no hay abandono. Teniendo esto en cuenta el servicio web devuelve el mensaje de “Abandona” o “No abandona” a la página web.

El servicio web hecho en flask junto con los predictores serializados fue subido a la plataforma Heroku dejándolo preparado para recibir peticiones en tiempo real.

En la Figura 21, Figura 22 y Figura 23 se ve la interfaz gráfica implementada con la documentación necesaria del servicio web indicada anteriormente para que un servicio externo pueda consumir el servicio expuesto de predicción de abandono de carrito de compras.

Figura 21

Documentación de los endpoints del servicio web

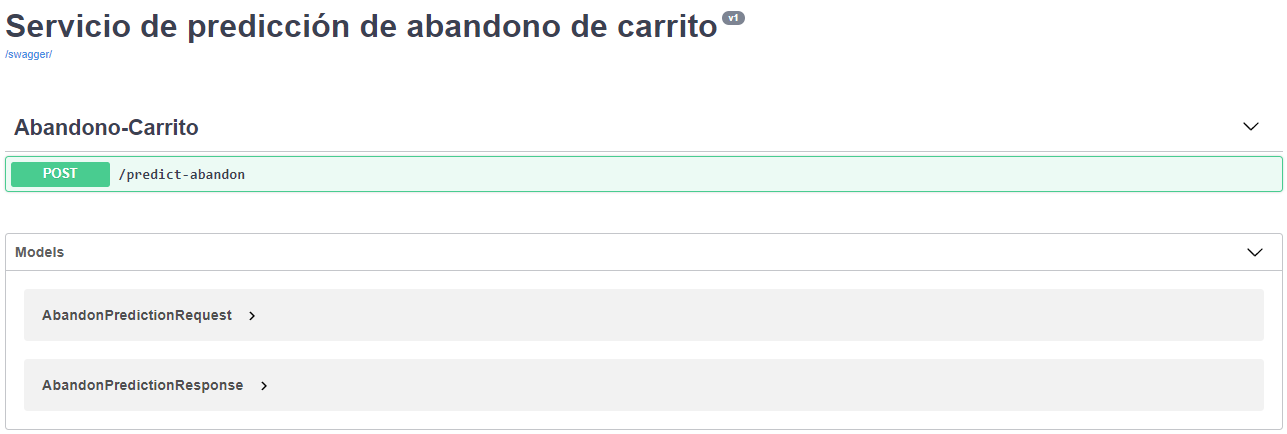


Figura 22

Documentación de los datos de entrada y salida del servicio web desarrollado

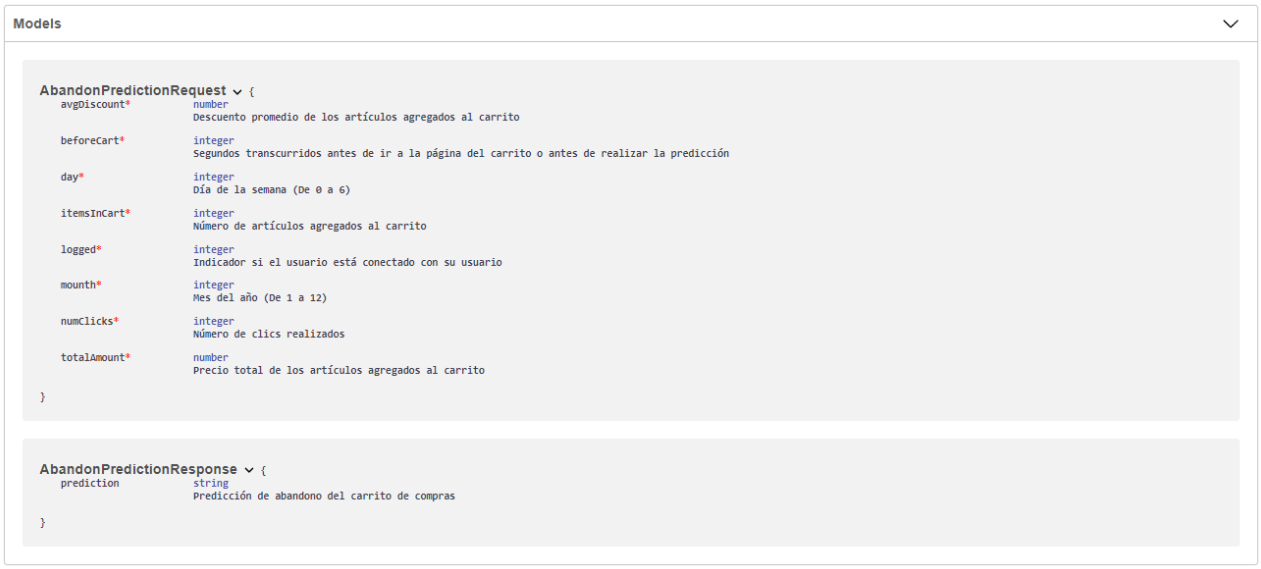
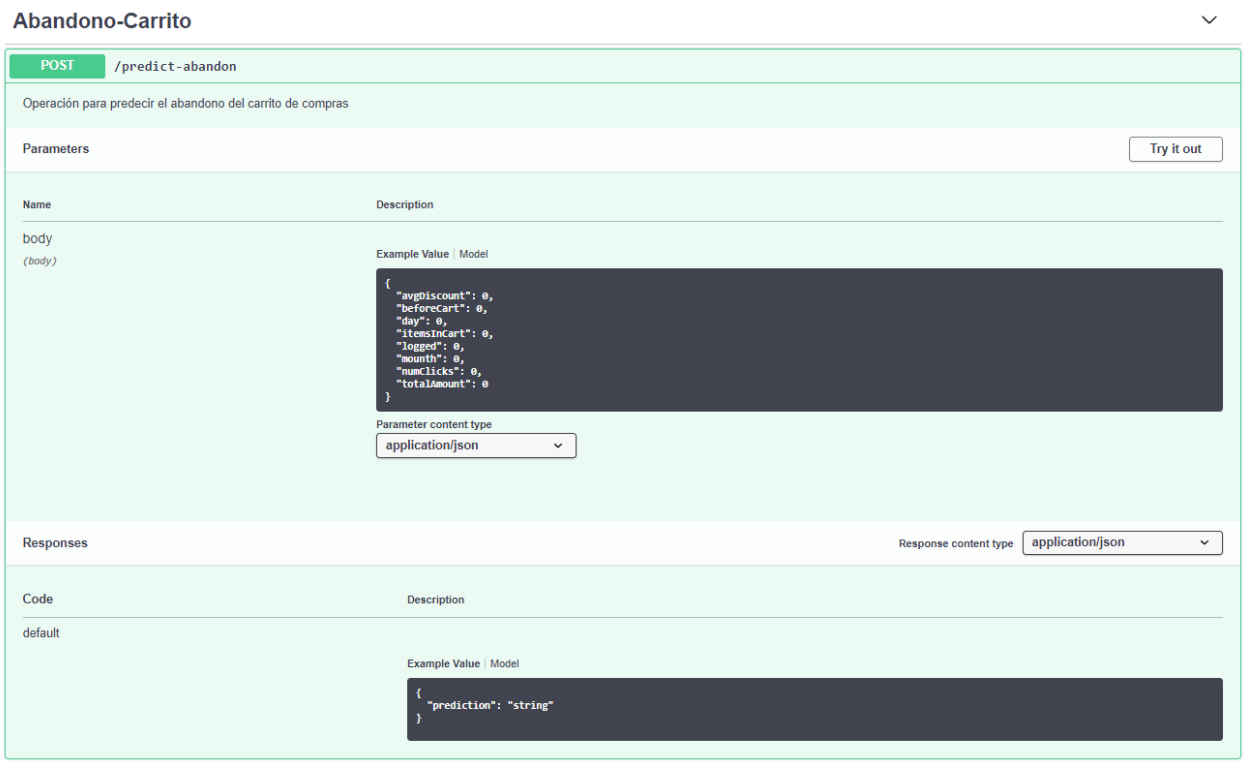


Figura 23

Documentación del endpoint de predicción de abandono de carrito



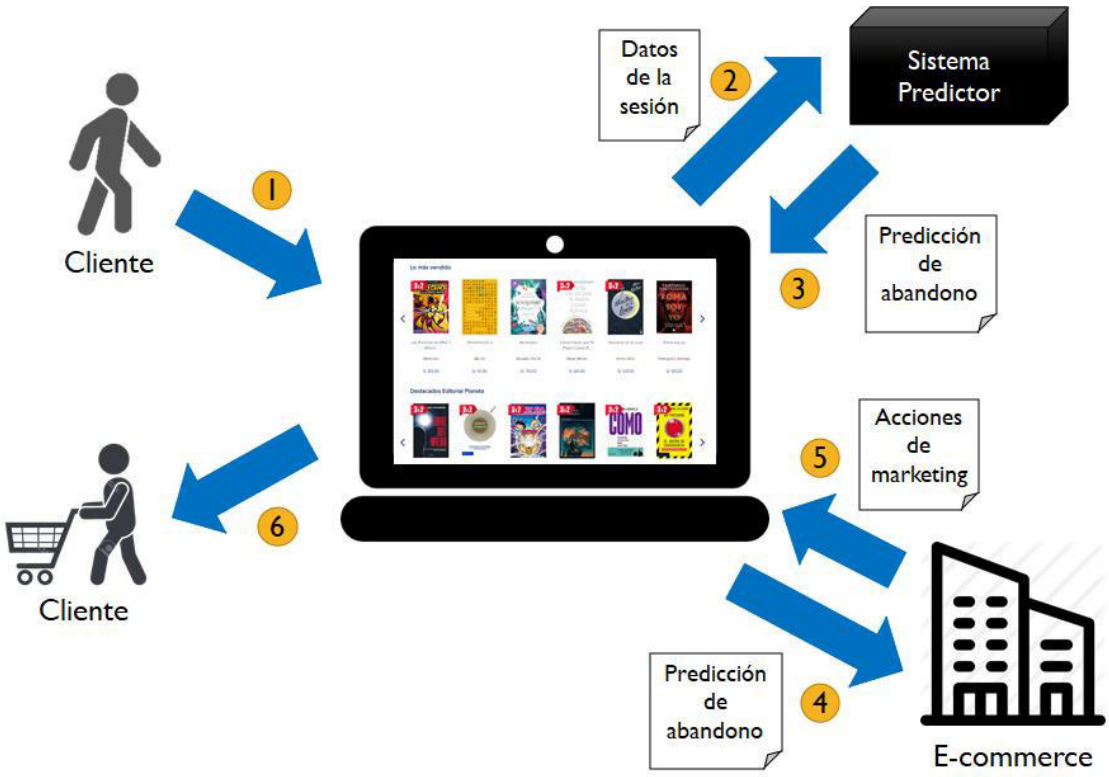
Por último para que la página web pueda hacer uso del servicio web de predicción del abandono de carrito se utilizó Google Tag Manager, una herramienta que facilita la implementación de código en las páginas web a través de la creación de etiquetas,

activadores y variables. Con esta herramienta se crearon las etiquetas y variables necesarias para que se almacenen los datos de sesión solicitados por el sistema predictor y se comunique con el servicio web para recibir la respuesta del mismo. Además se configuró las llamadas al servicio de predicción para determinar en qué momento predecir el abandono, para ello se realizó un previo análisis de los datos de sesiones donde se encontró que más del 80% de usuarios se mantienen en la página web hasta un máximo de 10 minutos. Teniendo esto en cuenta se configuró las llamadas al servicio web en 5 posibles momentos: cuando haya pasado 4 minutos el usuario en la sesión, 6 minutos, 8 minutos, 10 minutos y cuando el usuario entre a su carrito de compras. Siendo el intervalo de 4 minutos donde se encontró la mayor cantidad de predicciones.

En la Figura 24 se muestra el flujo de venta propuesto para la empresa con el sistema predictor implementado.

Figura 24

Flujo propuesto con el sistema predictor



Cabe mencionar que esta investigación se enfocó en los puntos 2, 3 y 4; por otra parte los puntos 5 y 6 mostrados en el gráfico no fueron abordados, queda por parte de la empresa el determinar qué acciones de marketing implementar en su página web que busque dar beneficios al cliente. De manera que se utilice la predicción de abandono del carrito de compras como activador de la acción de marketing, logrando así que los usuarios compren los productos que añadieron a su carrito.

Capítulo 6: Validación

En este capítulo se abordará el proceso de validación de la propuesta donde se especificará el método utilizado y los resultados obtenidos. Cabe mencionar que la validación del sistema predictor se realizó en el caso de estudio de un *e-commerce* dedicado a la venta de libros que tiene una página web para vender sus productos, dicho *e-commerce* brindó acceso a datos sensibles como son los datos de sus clientes y al código de su página web para poder testear el sistema predictor y los modelos predictivos con datos en tiempo real, sin embargo por pedido de la empresa no se menciona su nombre. Debido a la confidencialidad de los datos no fue posible trabajar con otra empresa del mismo rubro para la validación.

Los pasos realizados para la validación fueron los siguientes: primero se obtuvo los valores de la aplicación de las métricas de evaluación a cada modelo predictivo implementado independientemente, segundo se obtuvo los valores de las mismas métricas para el sistema predictor y finalmente se realizó la validación comparando los valores obtenidos del sistema predictor contra los valores obtenidos por cada modelo implementado. Para esto se utilizó la métrica *recall* debido a que le da mayor importancia a la detección de clientes que abandonaran el carrito de compras; además esta métrica fue muy utilizada por los autores de la bibliografía revisada para la validación y comparación de sus respectivos aportes. Con esta métrica se muestra que porcentaje de los clientes que abandonan su carrito de compras se logra predecir.

Para la validación del sistema predictor se obtuvo, a través de Google Analytics, un reporte con los datos de sesión de 13195 clientes que entraron a la página web del *e-commerce* entre los meses de noviembre y diciembre del año 2022, además de las predicciones realizadas a dichos clientes por el sistema desarrollado.

En el reporte obtenido se tuvo una columna con la predicción en tiempo real realizado por el sistema predictor y otra columna con la cantidad de productos comprados en la sesión además de los otros datos utilizados en la predicción.

Figura 25

Informe parcial con los resultados de la predicción obtenidos

Analytics | Web pública All Web Site Data

Ir al informe

Abandono de carrito de compras

Todos los usuarios
100,00 % totalAmount

1 nov 2022 - 31 dic 2022

Pestaña de informe

ga	Día de la semana	Mes del año	isLogged	prediction	totalAmount	numClicks	itemInCart	beforeCart	avgDiscount	Cantidad de productos comprados
1. GA1.3.202549373.1650892153	4	11	1	No abandonara	72.493,66 PEN (1,35 %)	10.274 (0,08 %)	589 (0,54 %)	2.915.019 (0,00 %)	0,00 PEN (0,00 %)	0 (0,00 %)
2. GA1.3.107333268.1668532154	2	11	0	No abandonara	46.232,02 PEN (0,86 %)	516 (0,00 %)	475 (0,43 %)	370.460 (0,00 %)	31,85 PEN (0,09 %)	0 (0,00 %)
3. GA1.3.148359946.1668390081	2	11	1	No abandonara	42.222,96 PEN (0,78 %)	585 (0,00 %)	519 (0,47 %)	195.236 (0,00 %)	51,88 PEN (0,15 %)	0 (0,00 %)
4. GA1.3.437976403.1669260530	1	11	1	No abandonara	38.010,25 PEN (0,71 %)	9.907 (0,08 %)	2.706 (2,47 %)	1.638.939 (0,00 %)	241,16 PEN (0,68 %)	0 (0,00 %)
5. GA1.1.279865697.1596807317	3	11	1	No abandonara	30.093,18 PEN (0,56 %)	3.030 (0,02 %)	1.500 (1,37 %)	2.293.032 (0,00 %)	150,02 PEN (0,42 %)	0 (0,00 %)
6. GA1.1.626616578.1666668610	4	11	1	No abandonara	25.067,50 PEN (0,47 %)	1.286 (0,01 %)	322 (0,29 %)	41.461 (0,00 %)	32,25 PEN (0,09 %)	0 (0,00 %)
7. GA1.3.198795993.1668558879	2	11	0	No abandonara	24.501,13 PEN (0,46 %)	1.990 (0,02 %)	414 (0,38 %)	270.847 (0,00 %)	65,90 PEN (0,19 %)	0 (0,00 %)
8. GA1.3.202549373.1650892153	2	11	1	No abandonara	23.765,41 PEN (0,44 %)	3.952 (0,03 %)	286 (0,26 %)	332.937 (0,00 %)	47,67 PEN (0,13 %)	0 (0,00 %)
9. GA1.3.437976403.1669260530	2	11	1	No abandonara	23.618,25 PEN (0,44 %)	1.663 (0,01 %)	1.206 (1,10 %)	318.922 (0,00 %)	109,08 PEN (0,31 %)	0 (0,00 %)
10. GA1.3.1268009216.1668225919	1	11	1	Abandonara el carrito	20.045,70 PEN (0,37 %)	5.753 (0,04 %)	953 (0,87 %)	850.928 (0,00 %)	95,34 PEN (0,27 %)	0 (0,00 %)

Primero los datos recuperados se utilizaron como datos de entrada para la predicción del abandono en cada modelo predictivo implementado para poder medir la efectividad de las predicciones por separado. Con esto se encontró que el modelo de XGBoost realiza mejores predicciones que AdaBoost y Bagging obteniendo un recall de 0.935. En la Figura 27, Figura 29 y Figura 31 se puede ver los resultados obtenidos de la predicción del abandono y en la Tabla 17, Tabla 18 y Tabla 19 las métricas de evaluación obtenidas de los predictores mencionados.

Figura 26

Código utilizado para generar la matriz de confusión de XGBoost

```
dataFrame = pd.read_excel('dataAbandono.xlsx')
dataFrame
```

```
dataFrame['Cantidad de productos comprados'] = np.where(dataFrame['Cantidad de productos comprados'] > 0 ,0,1)
dataFrame.drop(['ga', 'Prediccion'], axis=1, inplace=True)
dataFrame
```

```
X = dataFrame.loc[:,dataFrame.columns!='Cantidad de productos comprados']
Y = dataFrame.loc[:,'Cantidad de productos comprados']
```

```
plot_confusion_matrix(xgb,X,Y,values_format='d',display_labels=["no abandona","abandona"])
```

Nota. Se muestra el código principal utilizado para la generación de la matriz de confusión.

Figura 27

Matriz de confusión del predictor XGBoost entrenado

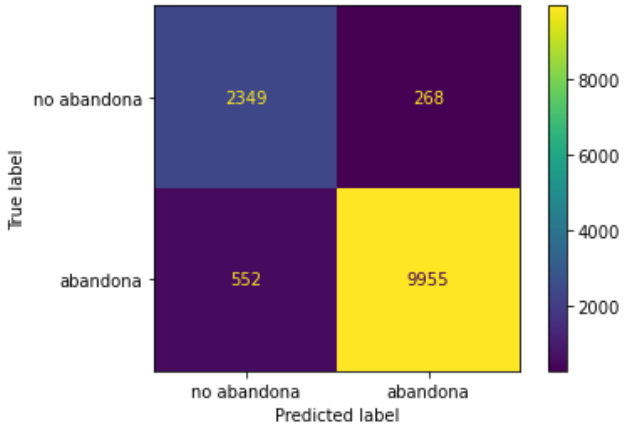


Tabla 17

Métricas de evaluación del predictor XGBoost

Métricas	Valor
Recall	0.9358
F1	0.9525
Precision	0.9698
Accuracy	0.9253

Figura 28

Código utilizado para generar la matriz de confusión de AdaBoost

```
dataFrame = pd.read_excel('dataAbandono.xlsx')
dataFrame
```

```
dataFrame['Cantidad de productos comprados'] = np.where(dataFrame['Cantidad de productos comprados'] > 0 ,0,1)
dataFrame.drop(['ga', 'Prediccion'], axis=1, inplace=True)
dataFrame
```

```
X = dataFrame.loc[:,dataFrame.columns!='Cantidad de productos comprados']
Y = dataFrame.loc[:,'Cantidad de productos comprados']
```

```
plot_confusion_matrix(adaboost,X,Y,values_format='d',display_labels=["no abandona","abandona"])
```

Nota. Se muestra el código principal utilizado para la generación de la matriz de confusión.

Figura 29

Matriz de confusión del predictor AdaBoost entrenado

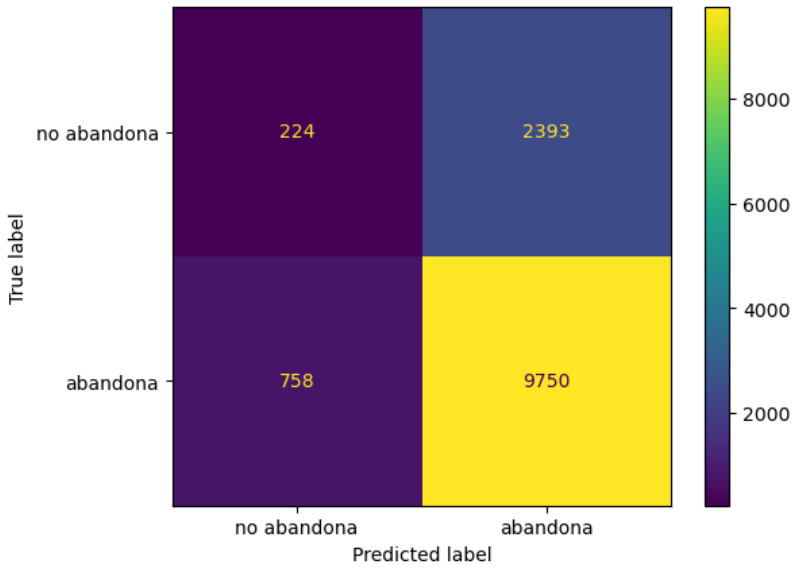


Tabla 18

Métricas de evaluación del predictor AdaBoost

Métricas	Valor
Recall	0.9278
F1	0.8608
Precision	0.8029
Accuracy	0.7599

Figura 30

Código utilizado para generar la matriz de confusión de Bagging

```
dataFrame = pd.read_excel('dataAbandono.xlsx')
dataFrame
```

```
dataFrame['Cantidad de productos comprados'] = np.where(dataFrame['Cantidad de productos comprados'] > 0 ,0,1)
dataFrame.drop(['ga', 'Prediccion'], axis=1, inplace=True)
dataFrame
```

```
X = dataFrame.loc[:,dataFrame.columns!='Cantidad de productos comprados']
Y = dataFrame.loc[:,'Cantidad de productos comprados']
```

```
plot_confusion_matrix(bagging,X,Y,values_format='d',display_labels=["no abandona","abandona"])
```

Nota. Se muestra el código principal utilizado para la generación de la matriz de confusión.

Figura 31

Matriz de confusión del predictor Bagging entrenado

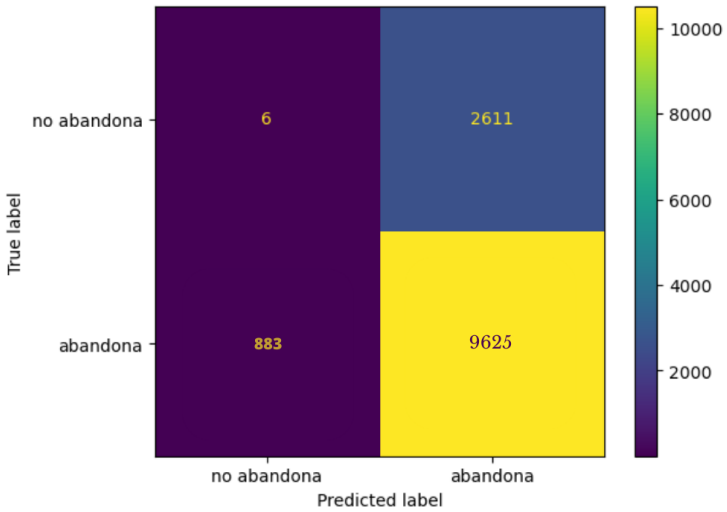


Tabla 19

Métricas de evaluación del predictor Bagging

Métricas	Valor
Recall	0.9160
F1	0.8464
Presicion	0.7866
Accuracy	0.7338

Segundo se usaron las columnas de predicción del abandono, obtenido a través de la predicción del sistema (votación de los resultados de predicción obtenidos de los modelos implementados), y la cantidad de productos comprados durante la sesión para construir la matriz de confusión del sistema predictor. En la Figura 32 se muestra la matriz de confusión

resultante de la predicción en tiempo real del abandono de carrito de compras y en la Tabla 20 las métricas correspondientes.

Figura 32

Matriz de confusión del sistema predictor

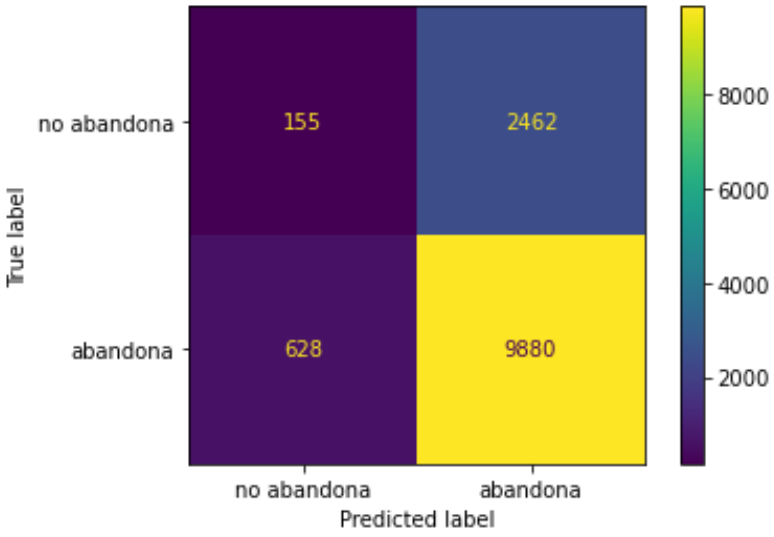


Tabla 20

Métricas de evaluación del sistema predictor

Métricas	Valor
Recall	0.9443
F1	0.8664
Presicion	0.8004
Accuracy	0.7669

De las métricas de evaluación calculadas del sistema predictor entrenado se obtiene un *recall* de 0.9443 mayor al obtenido por los modelos predictivos de manera independiente y mayor al encontrado en la bibliografía revisada de Osnat et al (2019), donde obtiene un *recall* de 0.9 y 0.9438 en los *e-commerce* donde aplica su modelo, demostrando la utilidad de la herramienta para realizar predicciones en tiempo real utilizando solo los datos obtenidos de la sesión del usuario.

Capítulo 7: Conclusiones y trabajos futuros

En este capítulo se exponen las conclusiones de la presente investigación, así como los posibles trabajos futuros.

7.1.Conclusiones

- Se implementó extreme gradient boosting machine, adaboost y bagging; tres técnicas de aprendizaje supervisado de machine learning para la predicción del abandono del carrito de compras. Estas técnicas fueron entrenadas con el 75% de un *dataset* de 55440 sesiones utilizando validación cruzada y un conjunto de parámetros para mejorar el proceso de entrenamiento. Posteriormente se utilizó el otro 25% para testear los modelos logrando un *recall* no menor a 0.94 en cada uno.
- Se construyó un sistema que utiliza los 3 modelos predictivos implementados y a través de votación determina la predicción del sistema de acuerdo a lo que la mayoría de los modelos predijo. Para esto se serializó los modelos predictivos ya entrenados para que el sistema pueda invocarlos y realizar la predicción sin tener que pasar por el entrenamiento otra vez. El sistema predictor recibe como entrada los datos de sesión del cliente y devuelve como respuesta a la página el mensaje de "Abandona el carrito" o "No abandona".
- Se logró que el sistema prediga el abandono del carrito de compras de un cliente antes de que el cliente abandone la página web. Para ello se construyó el servicio web que provee como único servicio la predicción del abandono del carrito de compras, con esto el sistema puede realizar predicciones en tiempo real para la página web. Además, se configuró las llamadas al servicio para que se pueda consultar múltiples veces la predicción de abandono de un cliente en un intervalo de tiempo no mayor a 10 minutos. El sistema fue probado con sesiones de clientes entre el mes de Noviembre y Diciembre

del año 2022 y validado a través de la métrica *recall* siendo 0.9443 el valor obtenido por el sistema mayor que los valores encontrados en la bibliografía revisada.

7.2.Trabajos futuros

Los modelos de predicción implementados pueden ser mejorados aumentando las características utilizadas para la predicción. Como se mostró en el estado del arte hay otras características como la entropía de contenido que no fueron utilizadas en esta investigación pero que pueden ser implementadas con el fin de mejorar la predicción de los modelos. De la misma forma se puede mejorar el entrenamiento de los predictores al aumentar el rango de valores de los parámetros o agregar otros parámetros en la grilla para buscar la mejor combinación.

Por otra parte como se mostró en la Figura 24 hay pasos que pueden ser cubiertos en futuras investigaciones, como es el caso de las acciones a tomar por la empresa para poder captar a los clientes que abandonarán. Al tener la predicción del abandono se pueden diseñar sistemas que partiendo de esta información realicen promociones o recomendaciones de productos para captar al cliente, es decir sistemas que realicen acciones de marketing de forma automática para conseguir que los clientes que abandonaran compren sus productos.

Bibliografía

Ahmet, G., & Mehmet, S. A. (2019). Prediction of Purchase Intention on the E-Commerce Clickstream Data. *Signal Processing and Communications Applications Conference.* Sivas.

Amazon Web Service. (s.f.). *¿Qué es el aprendizaje profundo?* Retrieved Abril 13, 2023, from https://aws.amazon.com/es/what-is/deep-learning/#:~:text=El%20aprendizaje%20profundo%20es%20un,inspira%20en%20el%20cerebro%20humano.

Amazon Web Service. (s.f.). *¿Qué es una red neuronal?* Retrieved Setiembre 21, 2022, from https://aws.amazon.com/es/what-is/neural-network/#:~:text=Una%20red%20neuronal%20es%20un,lo%20hace%20el%20cerebro%20humano.

Anahita, D., & Mainak, C. (2018). Social trust model for rating prediction in recommender systems: Effects of similarity, centrality, and social ties. *Elsevier*.

Arana, C. (2021). *Redes neuronales recurrentes: Análisis de los modelos especializados en datos secuenciales.* Buenos Aires: Valeria Dowding.

Bhattacharjee, D., Ramesh, K., Jayaram, E. S., & Mathad, M. S. (2023). An integrated machine learning and DEMATEL approach for feature preference and purchase intention modelling. *Decision Analytics Journal*.

Bichen, Z., & Bingwei, L. (2018). A Scalable Purchase Intention Prediction System Using Extreme Gradient Boosting Machines with Browsing Content Entropy. *IEEE International Conference on Consumer Electronics*, (pp. 1-4). Las Vegas.

Bing, L., & Yuliang, S. (2017). Prediction of User's Purchase Intention Based on Machine Learning. *International Conference on Soft Computing and Machine Intelligence*, (pp. 99-103). Dubai.

Chaudhuri, N., Gupta, G., Vamsi, V., & Bose, I. (2021). On the platform but will they buy? Predicting customers' purchase behavior using deep learning. *Decision Support Systems*.

Dayal, K. B., Madhabananda, D., & Subhra, S. (2019). Predicting Users' Preferences for Movie Recommender System Using Restricted Boltzmann Machine. *Springer Nature Singapore*.

Dirk, T., & Dirk, V. d. (2012). Predicting e-commerce company success by mining the text of its publicly-accessible website. *Elsevier*.

Georg, L. G., Angelo, C., C., H. B., Duncan, A. L., & Benjamin, P. C. (2019). A Recurrent Neural Network Survival Model: Predicting Web User Return Time. *Imperial College London*.

Grażyna, S., & Sławomir, S. (2017). Application of neural network to predict purchases in online store. *Conferencia de Advances in Intelligent Systems and Computing*, (pp. 221-231). Opole.

Hernández, F., & Herrera, F. (2012). Identificación Inteligente de un Proceso Fermentativo Usando el Algoritmo GMDH Modificado. *Iberoamericana de Automática e Informática industrial 9* , 3-13.

IBM. (s.f.). *Regresión Logística*. Retrieved setiembre 21, 2022, from https://www.ibm.com/docs/es/spss-statistics/saas?topic=regression-logistic

Jae-Do, S. (2019). A Study on Online Shopping Cart Abandonment: A Product Category Perspective. *Internet Commerce*.

Jagatjyoti, G. T., Ashish, G., Sachit, S., Ujjen, M. B., & K., B. (2018). Predictive Analysis of E-Commerce Products. *Springer Nature*.

Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering version 2.3. *Elsevier*.

Meiling, L., & Leihui, C. (2018). Improving the Accuracy of Rating Prediction with User Attention Degree. *East China Normal University*.

Mena, F. G. (2019, 01 17). *Gestion*. Retrieved from Gestion El Comercio: https://gestion.pe/tu-dinero/10-emprendimientos-peruanos-llegan-exito-parte-fracaso-255933-noticia/

Osnat, M., Veronika, B., & Tsvi, K. (2019). Will this session end with a purchase? Inferring current purchase intent of anonymous visitors. *Electronic Commerce Research and Applications*.

Qian, G., Chun, Y., & Shaoqing, T. (2020). Prediction of Purchase Intention among E-Commerce Platform Users Based on Big Data Analysis. *International Information and Engineering Technology Association*.

Rubin, D., Martins, C., Ilyuk, V., & Hildebrand, D. (2020). Online shopping cart abandonment: a consumer mindset perspective. *Consumer Marketing*.

Schwaber, K., & Sutherland, J. (2020, Noviembre). *La Guía de Scrum.* Retrieved from Scrum Guides: https://scrumguides.org/download.html

Sharma, S. N., & Prasanna, S. (2019). A novel purchase target prediction system using extreme gradient boosting machines. *International Journal of Innovative Technology and Exploring Engineering*.

Sujoy, B., Manoj, K. T., & Felix, T. C. (2019). Predicting the consumer's purchase intention of durable goods: An attribute-level analysis. *Journal of Business Research*.

Thakkar, J. J. (2021). *Studies in Systems, Decision and Control* (Vol. 336). Vadodara: Springer, Singapore.

Torra Porras, S. (2004). Siniestralidad en seguros de consumo anual de las entidades de previsión social, La. Perspectiva probabilística y econométrica. Propuesta de un modelo econométrico neuronal para Cataluña. *[Tesis de Doctorado, Universitat de Barcelona].* Retrieved from http://diposit.ub.edu/dspace/handle/2445/35334

Vishal, S., & Kyumin, L. (2018). Predicting Highly Rated Crowdfunded Products. *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, (pp. 357-362). Barcelona.

Yunghui, C., Hui-Kuo, Y., & Wen-Chih, P. (2019). Predicting Online User Purchase Behavior Based on Browsing History. *Proceedings of the 30th ACM Conference on Hypertext and Social Media*, (pp. 143-152). Hof.