PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

Trabajo de Titulación como requisito previo para la obtención del título de
Magíster en Tecnologías de Información mención Gestión y Administración de TI

DESARROLLO DE UN MODELO PREDICTIVO PARA DETERMINAR LA FIDELIDAD DE LOS CLIENTES DE UNA EMPRESA DE RETAIL FARMACÉUTICO

Autor: Angel Andres Espinoza Vega
Director: Henry N. Roa, PhD

Quito, 07 agosto de 2023

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

DECLARACIÓN Y AUTORIZACIÓN

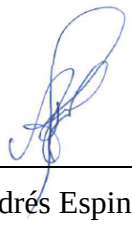
Yo, Espinoza Vega Angel Andrés, con cédula de ciudadanía No. 0704395508, autor del trabajo de graduación titulado: "**DESARROLLO DE UN MODELO PREDICTIVO PARA DETERMINAR LA FIDELIDAD DE LOS CLIENTES DE UNA EMPRESA DE RETAIL FARMACÉUTICO**", previo a la obtención del título de **Magíster en Tecnologías de Información mención Gestión y Administración de TI**, de la Facultad de Ingeniería:

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE, el referido trabajo de graduación, respetando las políticas de propiedad intelectual de Universidad.

Quito, 07 de agosto de 2023

Atentamente;



Angel Andrés Espinoza Vega

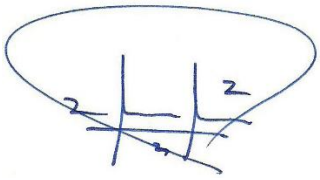
Autor del Trabajo de Titulación

Av. 12 de octubre 1076 y Ramón Roca
Apartado postal 17-01-2184
Telf.: (593) 2 2991647
Quito – Ecuador www.puce.edu.ec

APROBACIÓN DEL TUTOR

En mi carácter de Director (a) – Tutor (a) del Trabajo de Posgrado Titulado: "DESARROLLO DE UN MODELO PREDICTIVO PARA DETERMINAR LA FIDELIDAD DE LOS CLIENTES DE UNA EMPRESA DE RETAIL FARMACÉUTICO", presentado por el maestrante ANGEL ANDRÉS ESPINOZA VEGA, titular de la Cédula de Identidad Nº 0704395508 para optar al Grado de Magíster en Magíster en Tecnologías de Información mención Gestión y Administración de TI, considero que dicho Trabajo de Investigación reúne los requisitos y méritos suficientes para ser sometido a la evaluación por parte de los Lectores – Evaluadores que se designen para tal fin por parte de las autoridades de la Facultad de Ciencias de la Educación.

En la ciudad de Quito, a los 07 días de agosto de 2023.



Henry Nelson Roa Marin C.I. 1103699524

hnroa@puce.edu.ec

NRO TELEFONO: 02 2991700 ext 1212

NOTA:

Se comunica que en el servicio de análisis Turnitin, el referido trabajo de titulación alcanzó el siguiente resultado: 9 % índice de similitud con otras fuentes.

Av. 12 de octubre 1076 y Ramón Roca
Apartado postal 17-01-2184
Telf.: (593) 2 2991647
Quito – Ecuador www.puce.edu.ec

TURNITIN: INCLUIR HOJA DEL INFORME CON EL PORCENTAJE

Informe de Originalidad Turnitin

Proyecto de titulación rev 2 por Angel Espinoza

Desde Proyectos grado (Proyectos titulación)

Índice de similitud	Similitud según fuente	
9%	Internet Sources:	9%
	Publicaciones:	0%
	Trabajos del estudiante:	N/A

Procesado el 06-ago.-2023 14:27 -05
Identificador: 2142130704
Número de palabras: 8609

fuentes:

1 1% match (Internet desde 18-oct.-2022)
http://repositorio.ug.edu.ec/bitstream/redug/60129/1/SARANGO%20JUMBO%20JANDRY%20JAVIER.pdf

2 1% match (Internet desde 08-oct.-2022)
http://fundacionlasirc.org/images/Revista/REVISTALASIRCVolumen3.No.1.pdf

3 1% match (Internet desde 24-sept.-2022)
http://risti.xyz/issues/ristie28.pdf

4 1% match (Internet desde 06-ene.-2023)
https://www.ibm.com/docs/es/idr/11.3.3?topic=console-replicating-data-definition-language-ddl-changes

5 1% match (Internet desde 22-ago.-2022)
https://www.tibco.com/es/reference-center/what-is-logistic-regression

6 1% match (Internet desde 05-dic.-2020)
http://libroweb.alfaomega.com.mx/book/1007/free/data/contenidos_cap6.pdf

7 1% match (Internet desde 11-dic.-2020)
https://sites.google.com/site/luisamayateacher/limpieza-de-datos---python

8 < 1% match (Internet desde 20-dic.-2022)
http://repositorio.ug.edu.ec/bitstream/redug/59903/1/B-CISC-PTG%232062-A%c3%b1o%202022%20Guzm%c3%a1n%20Vergara%20Ronald%20Iv%c3%a1n%20-%20Pe%c3%b1afiel%20L%c3%b3pez%20Gregorio%20Francisco.pdf

9 < 1% match (Internet desde 28-oct.-2022)
https://www.coursehero.com/file/83597013/Foro-1-Qu%C3%A9-es-NoSQLdocx/

10 < 1% match (Internet desde 11-oct.-2022)
https://www.coursehero.com/file/135207421/Ejercicio-de-Flujo-de-Efectivodocx/

Av. 12 de octubre 1076 y Ramón Roca
Apartado postal 17-01-2184
Telf.: (593) 2 2991647
Quito – Ecuador www.puce.edu.ec

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, **Angel Andrés Espinoza Vega**, con cédula de ciudadanía No. 0704395508, declaro que los resultados obtenidos en la investigación "**DESARROLLO DE UN MODELO PREDICTIVO PARA DETERMINAR LA FIDELIDAD DE LOS CLIENTES DE UNA EMPRESA DE RETAIL FARMACÉUTICO**", como requisito previo para lo obtención del Grado Académico de **Magíster en Tecnologías de Información mención Gestión y Administración de TI** son absolutamente originales, auténticos y personales.

En tal virtud, declaro que el contenido, las conclusiones y los efectos legales y académicos, que se desprenden del trabajo de investigación, y luego de la redacción de este documento, son y serán de mi sola y exclusiva responsabilidad legal y académica.

En la ciudad de Quito, a los 07 días del mes de agosto de 2023.

Atentamente;

Angel Andrés Espinoza Vega

Autor del Trabajo de Titulación

ÍNDICE DE CONTENIDOS

ÍNDICE DE ILUSTRACIONES

ÍNDICE DE TABLAS

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
MAESTRIA EN TECNOLIGÍAS DE LA INFORMACIÓN MENCIÓN GESTIÓN Y ADMINISTRACIÓN DE TI

DESARROLLO DE UN MODELO PREDICTIVO PARA DETERMINAR LA FIDELIDAD DE LOS CLIENTES DE UNA EMPRESA DE RETAIL FARMACÉUTICO

Autor: Angel Andres Espinoza Vega

Director -Tutor: Henry N. Roa, PhD

Fecha: 07 de agosto de 2023

RESUMEN

Los modelos de predicción son importantes porque permiten a las empresas y organizaciones tomar decisiones informadas basadas en datos y mejorar su eficiencia y efectividad, por ello el presente trabajo se centra en desarrollar un modelo predictivo que permita determinar la alta probabilidad de abandono de los clientes en una empresa de retail farmacéutico, brindando apoyo al área de mercadeo. Se aplicó la metodología CRISP-DM y se utilizó Python como herramienta de ciencia de datos. Como algoritmo para la predicción, se evaluaron tres modelos de clasificación: Red neuronal, Árboles de decisión y Regresión logística, con el fin de determinar el modelo más adecuado el cual resultó la Red Neuronal con un Accuracy de 0.99.

Palabras clave: Red Neuronal, Python, Retail Farmacéutico, RFM, Churn.

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
MAESTRIA EN TECNOLIGÍAS DE LA INFORMACIÓN MENCIÓN GESTIÓN Y ADMINISTRACIÓN DE TI

DEVELOPMENT OF A PREDICTIVE MODEL TO DETERMINE CUSTOMER LOYALTY IN A PHARMACEUTICAL RETAIL COMPANY

Autor: Angel Andres Espinoza Vega

Director -Tutor: Henry N. Roa, PhD

Fecha: 07 de agosto de 2023

ABSTRACT

Prediction models are important because they allow companies and organizations to make informed decisions based on data and improve their efficiency and effectiveness. Therefore, this study focuses on developing a predictive model that determines the high probability of customer churn in a pharmaceutical retail company, providing support to the marketing department. The CRISP-DM methodology was applied, and Python was used as the data science tool. Three classification models were evaluated for prediction: Neural Network, Decision Trees, and Logistic Regression, in order to determine the most suitable model, which turned out to be the Neural Network with an Accuracy of 0.99.

Keywords: Neural Network, Python, Pharmaceutical Retail, RFM, Churn.

CAPÍTULO I: INTRODUCCIÓN

1.1. Formulación del problema

En el negocio del retail se encuentran las empresas que se dedican a la comercialización masiva de productos o servicios a gran cantidad de consumidores, debido a la cantidad y variedad de productos que comercializan y al tener un gran número de transacciones generadas diariamente, se convierte en uno de los sectores más difíciles de conseguir fidelización de clientes. Al ser productos commodities, son productos de alta demanda y de fácil acceso que se pueden encontrar en diferentes tiendas o negocios.

Específicamente las empresas del retail farmacéutico no son la excepción, para eso existen métodos y estrategias de mantener a un cliente fidelizado. Sin embargo, es inevitable tener un nivel de abandono de recompra elevado.

Con estos factores como antecedentes, nos lleva a plantear la siguiente problemática: ¿Como determinar los clientes que tienen una alta probabilidad de abandono o alta posibilidad de no recompra?

1.2. Objetivos

Objetivo General:

Desarrollar un modelo predictivo que permita determinar los clientes que tienen una alta probabilidad de abandono en las recompras, con el fin de que se puedan mejorar las estrategias de fidelización de los clientes.

Objetivos Específicos:

- Establecer las mejores técnicas de analítica de datos para el desarrollo del modelo predictivo.
- Determinar las variables necesarias para entrenar el modelo predictivo.
- Predecir la posibilidad alta de abandono de recompra de un cliente.

1.3. Justificación

Existe una correlación positiva entre Marketing Digital, Marketing de Contenido, Redes sociales, Motores de Búsqueda y Correo Electrónico con la fidelización de clientes de una empresa de retail mayorista, por lo cual recomienda implementar y mejorar nuevas estrategias para Marketing Digital y estas logren fidelizar a los clientes (Flores Alejos, 2019).

Los esfuerzos por tener fidelidad de los clientes se centran en facilitar su proceso de compra a través de plataformas digitales que permitan interactuar con el cliente y sus distintos productos (Vicente Carbajal, 2019). De acuerdo con los resultados obtenidos por este mismo autor se obtiene una correlación positiva regular, por lo cual recomienda que se debe mejorar el conocimiento de lo que el cliente desea.

Existen diferentes criterios para mejorar la fidelidad de los clientes. Sin embargo, en este trabajo de titulación propondremos el desarrollo de un modelo predictivo para pronosticar, por medios de probabilidades, si un cliente abandonará su fidelidad a la marca.

La analítica predictiva se basa en técnicas estadísticas y otras técnicas de analítica de datos para determinar un evento probable que suceda en el futuro, por lo que es ideal para resolver el problema planteado en este trabajo.

CAPÍTULO II: MARCO TEÓRICO

2.1. Bases de datos

Las bases de datos son indispensables para el uso de sistemas de información ya que permiten: almacenar, modificar, procesar e interconectar la información relacionada y necesaria para la correcta marcha de las aplicaciones y programas informáticos (Arcidiacono, 2021).

Una base de datos es un conjunto de datos que se almacena y se accede de manera sistemática. Existen dos tipos de sistemas de bases de datos: los sistemas de bases de datos SQL y los sistemas de bases de datos NoSQL.

2.1.1. Bases de datos relacionales

El modelo relacional (SQL) es el enfoque más usado para la administración de datos. A pesar de esta premisa, este modelo tiene varias limitaciones que pueden llevar a complicar su gestión.

Las bases de datos relaciones son un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. En una base de datos relacional, cada fila de la tabla es un registro con un ID único llamado clave. Las columnas de la tabla contienen atributos de los datos, y cada registro generalmente tiene un valor para cada atributo, lo que facilita el establecimiento de las relaciones entre los puntos de datos (Oracle, 2019).

El problema principal de las bases de datos relacionales es que no son efectivas al gestionar grandes volúmenes de datos (Jose & Abraham, 2020).

2.1.2. Bases de datos no relacionales

Las arquitecturas más usadas frecuentemente, están estructuradas por sistemas de gestión de bases de datos relacionales, las bases de datos no relacionales (NoSQL) fueron creadas y desarrolladas para solucionar las desventajas de las bases de datos relacionales,

como el diseño sin esquema, la escala horizontal y consistencia (Diogo et al., 2019).

Estas bases de datos no requieren esquemas de tablas fijas y no soportan operaciones Join (Joyanes Luis, 2019). Están optimizadas para operaciones de lectura/escrituras escalables en lugar de pura consistencia.

2.2. Cubos de datos

Los cubos de datos, también conocidos como cubos OLAP, ofrecen la rapidez y la flexibilidad necesarias para hacer consultas complejas desde diferentes perspectivas. Estas herramientas facilitan una presentación multidimensional de los datos almacenados en un Data Mart, organizándolos y resumiéndolos para mejorar la eficacia de las consultas analíticas.

OLAP es una tecnología que permite el análisis multidimensional y multinivel de un gran volumen de datos, proporcionando visualizaciones de datos agregados con diferentes perspectivas (Queiroz-Sousa & Salgado, 2019).

2.3. Postgresql

"PostgreSql es un sistema de gestión de base de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de base de datos de código abierto más potente del mercado" (León Jenner, 2020, p. 26,27).

La investigadora describe a Postgresql de la siguiente forma:

> PostgreSQL está muy bien catalogado por su estabilidad, potencia, robustez y la facilidad de administración e implementación. Adicionalmente, utiliza un sistema cliente servidor con el uso de hilos para un procesamiento correcto de las consultas hacia la base de datos (Pilicita Garrido et al., 2020, p. 10).

2.4. DML en Postgresql

De acuerdo con la documentación de IBM se detallan las sentencias DDL:

> Las sentencias DML se utilizan para controlar la información contenida en la base de

datos. Las listas siguientes ofrecen ejemplos de estos tipos de sentencias DML:

- Adición de registros a una tabla (mandato INSERT)
- Modificación de la información de una tabla (mandato UPDATE)
- Eliminación de registros de una tabla (mandato DELETE) (IBM, 2021).

El lenguaje de consulta estructurada de las bases de datos relacionadas es SQL (Structural Query Language), permite acceder a la gestión de dichas bases de datos y realizar tareas típicas como: recoger, eliminar, agregar o modificar información. (*Cambios en la réplica del lenguaje de definición de datos (DDL) - Documentación de IBM*, s/f).

2.5. Proceso de análisis de datos (CRISP-DM)

CRISP-DM (Cross-Industry Standard Process for Data Mining) es un modelo de proceso para la minería de datos y análisis de datos. Es un marco de trabajo estructurado que se utiliza para guiar a los profesionales de datos en el desarrollo de proyectos de minería de datos y análisis de datos (Huber et al., 2019).

El modelo CRISP-DM se compone de seis fases fundamentales:

1. Comprensión del negocio: En esta fase se identifica el problema empresarial y se establecen los objetivos del proyecto.
2. Comprensión de los datos: Durante esta fase se recopilan los datos y se realiza una exploración inicial de ellos para comprender su calidad y estructura.
3. Preparación de los datos: En esta fase se limpian y preparan los datos para su uso en el análisis.
4. Modelado: Durante esta fase se construyen modelos estadísticos y algoritmos para analizar los datos.
5. Evaluación: Esta fase implica la evaluación de los modelos y la selección del mejor modelo para su uso.

6. Despliegue: En esta fase se lleva a cabo la implementación del modelo seleccionado en el entorno de producción.

2.6. Herramientas de Ciencia de Datos

La ciencia de datos puede ser aplicada en casi toda industria u organización que exista, siempre y cuando cuente con datos de donde extraer información útil, para ello existen herramientas como Python (Pandas) y R.

2.6.1 Lenguaje Python

Python es un lenguaje de programación interpretado de alto nivel, que es ampliamente reconocido por ser fácil de aprender, pero aún capaz de aprovechar el poder de los lenguajes de programación a nivel de sistemas cuando sea necesario (Raschka et al., 2020).

Python tiene una curva de aprendizaje más suave en comparación con R, lo que lo hace más accesible para aquellos que no tienen experiencia en programación. Python tiene una sintaxis más clara y fácil de leer, lo que facilita su comprensión. Además, Python tiene una comunidad de desarrolladores muy activa y una gran cantidad de recursos y tutoriales en línea disponibles, lo que hace que sea más fácil aprender y resolver problemas.

Pandas es una biblioteca de Python que se utiliza para la manipulación y análisis de datos. Los autores destacan que Pandas es una de las bibliotecas más populares para el análisis de datos en Python, y que se utiliza para la carga, limpieza, transformación y exploración de datos (Lemenkova, 2019). Pandas es una de las bibliotecas más populares para la manipulación y el análisis de datos. Con Pandas, puedes cargar datos desde diferentes fuentes, como archivos CSV, bases de datos SQL y Excel, y luego manipularlos y analizarlos en un entorno de programación amigable y flexible. Además, Pandas ofrece herramientas para visualizar datos y realizar análisis estadísticos avanzados, como modelos de regresión logística, modelos predictivos y árboles de decisión. Todo esto hace que Python sea el lenguaje preferido de los científicos de datos y estadísticos, ya que ofrece una amplia

variedad de herramientas y bibliotecas para trabajar con datos de manera efectiva y eficiente.

NumPy es una biblioteca de Python que se utiliza para realizar cálculos matemáticos complejos y eficientes en grandes conjuntos de datos. Es ampliamente utilizado en la ciencia de datos y la investigación científica debido a su eficiencia y capacidad para trabajar con matrices y arreglos multidimensionales (Oliphant et al., 2020).

Matplotlib, que es una biblioteca de visualización de datos de Python muy popular y ampliamente utilizada en la comunidad científica. El libro cubre el uso de Matplotlib para la creación de gráficos básicos, como gráficos de línea y gráficos de barras, así como gráficos más avanzados, como gráficos de contorno y gráficos 3D (Yim et al., 2019).

Seaborn proporciona una gran cantidad de herramientas para la creación de gráficos estadísticos, incluyendo gráficos de dispersión, diagramas de violín, mapas de calor, gráficos de regresión, entre otros (Waskom, 2017).

Scikit-learn es una biblioteca de aprendizaje automático de código abierto en Python que proporciona una amplia gama de herramientas y algoritmos para la modelización y análisis de datos (Tran et al., 2022).

Scikit-learn incluye herramientas para la clasificación, la regresión, el agrupamiento, la selección de características, la normalización de datos y la evaluación del rendimiento del modelo.

2.6.2 Lenguaje R

El lenguaje R se puede definir como:

> R es un lenguaje de programación especialmente orientado al análisis estadístico y a la representación gráfica de los resultados obtenidos. R se distribuye como software libre bajo la licencia GNU y es multiplataforma (hay versiones para plataformas Windows, Mac y Linux, y de hecho algunas distribuciones de Linux lo tienen incorporado), lo que también ha facilitado su adopción y la existencia de Una

comunidad muy activa su entorno, con constantes desarrollos de nuevas funcionalidades y versiones mejoradas de las existentes (Rojas, 2020, p. 592).

R tiene una curva de aprendizaje más empinada, ya que tiene una sintaxis más complicada y requiere un conocimiento más profundo de la estadística. Sin embargo, R es muy popular en la comunidad de análisis de datos y estadísticos debido a su amplia gama de paquetes y herramientas estadísticas avanzadas (Fernández Lizana, 2020).

2.7. Jupyter Notebook

Jupyter Notebooks es un entorno interactivo de programación que permite a los usuarios combinar código, texto y visualizaciones en una única interfaz (Beg et al., 2021).

Los Jupyter Notebooks se dividen en celdas, donde cada celda puede contener código de programación, texto enriquecido, imágenes, videos y otros elementos multimedia. Los usuarios pueden escribir código en las celdas de código, ejecutar el código y ver los resultados de inmediato en la misma interfaz.

Además, los Jupyter Notebooks son altamente personalizables y extensibles, lo que significa que los usuarios pueden crear sus propias extensiones para la funcionalidad de la plataforma. Los Jupyter Notebooks también son altamente colaborativos, lo que significa que varios usuarios pueden trabajar en el mismo documento al mismo tiempo y ver los cambios en tiempo real.

2.8. Modelos de clasificación

Los modelos de clasificación son técnicas de aprendizaje automático que se utilizan para categorizar o clasificar datos en diferentes grupos o categorías. Algunos ejemplos de modelos de clasificación son:

- Árboles de decisión: Son modelos que utilizan un árbol para representar las decisiones y las posibles consecuencias de una serie de acciones (Andrés et al., 2020). Cada nodo del árbol representa una decisión y cada rama del árbol representa una posible

consecuencia.

- Regresión logística: Es un modelo estadístico que se utiliza para predecir la probabilidad de que un evento ocurra en función de una serie de variables (Robles-Velasco et al., 2020).
- Redes neuronales artificiales: Son modelos que imitan el comportamiento del cerebro humano y se utilizan para clasificar datos en diferentes categorías (Shirley & Rueda, 2022).
- Máquinas de vectores de soporte: Son modelos que se utilizan para clasificar datos en dos o más categorías y se basan en la separación de los datos en un espacio multidimensional (De et al., 2022).
- Naive Bayes: Es un modelo estadístico que se utiliza para clasificar datos en diferentes categorías en función de la probabilidad de que pertenezcan a cada una de ellas (Wongkar & Angdresey, 2019).

CAPÍTULO III: METODOLOGÍA

3.1. Tipo de Investigación

Para determinar la cantidad de abandono de una empresa de retail, el enfoque de investigación más adecuado sería la investigación cuantitativa, ya que se trata de una medición numérica que se puede cuantificar y analizar estadísticamente. En una investigación cuantitativa, se puede diseñar una encuesta o cuestionario para recolectar datos sobre el abandono de la empresa, como la frecuencia, las razones, las tasas y los patrones de abandono. Luego, se pueden analizar los datos utilizando técnicas estadísticas para identificar las tendencias y patrones de abandono y para establecer relaciones causales entre las variables. Por lo tanto, la investigación cuantitativa sería un enfoque adecuado para investigar el abandono de una empresa de retail.

Según (Bryman Alan, 2016) la investigación cuantitativa se enfoca en la medición y análisis de datos numéricos para entender los fenómenos sociales. La investigación cuantitativa a menudo es utilizada para probar hipótesis o teorías y para establecer relaciones causales entre variables. Sin embargo, Bryman también señala que la investigación cuantitativa puede ser limitada en su capacidad para capturar la complejidad de los fenómenos sociales y para comprender el significado que las personas les otorgan a sus experiencias.

3.2. Diseño de Investigación

3.2.1. Diseño Correlacional Longitudinal

El diseño correlacional longitudinal es una metodología de investigación que permite analizar cómo se relacionan variables específicas a lo largo del tiempo. Es útil para identificar tendencias y patrones en la evolución de estas relaciones en un contexto temporal prolongado.

Este diseño te permitiría explorar las relaciones entre variables cuantitativas

relacionadas con el abandono en una empresa de retail a lo largo del tiempo.

Por lo que la aplicación de la metodología CRISP-DM en el diseño "Correlacional Longitudinal" para analizar el abandono de clientes en una empresa de retail ofrece un enfoque estructurado para abordar este desafío. Siguiendo las etapas de la metodología, se espera obtener información valiosa sobre los factores que influyen en el abandono de clientes y, en última instancia, mejorar la retención de clientes y el éxito de la empresa de retail.

3.3. Unidades de Estudio

3.3.1. Población

La empresa de retail farmacéutico objeto de estudio, cuenta con una población de 932918 clientes aproximadamente en su base de datos, comprendida entre hombres y mujeres de diferentes edades y grupos etarios.

3.3.2. Muestra

La muestra seleccionada para este estudio comprende datos desde el 01/01/2022 hasta el 31/07/2023, con un total de 4770651 líneas de registros de transacciones y un total de 120352 clientes de ambos géneros comprendidos entre 18 y 80 años, el cual conforman 6 rangos etarios: Generacion_Silenciosa, Baby_Boomers, Generacion_X, Millennials, Generacion_Z, Generacion_Alpha.

Según (Vafeiadis et al., 2015) al estudiar el abandono de clientes en el sector minorista, es crucial basarse en una muestra de datos que represente fielmente la composición y características de la base total de clientes. Una muestra representativa capturará adecuadamente la diversidad geográfica, demográfica y conductual relevante de los clientes. Esto permite un análisis más preciso de los factores que influyen en la retención y deserción de clientes. Además, una buena muestra representativa resulta en hallazgos de investigación más generalizables y aplicables. En resumen, para entender a profundidad las causas del abandono de clientes minoristas, los investigadores deben invertir en crear una muestra de

datos robusta y representativa, en lugar de basarse en muestras pequeñas o sesgadas. Esto asegurará un estudio confiable y outcomes de investigación de mayor utilidad práctica.

3.4. Técnicas e instrumentos de recolección de datos

3.4.1. Red Neuronal:

La red neuronal es un modelo de aprendizaje profundo que se basa en la estructura del cerebro humano para realizar tareas de clasificación y predicción. Consiste en una red de nodos interconectados, llamados neuronas, que procesan la información y generan una salida.

Para el caso de estudio, se configuró una red neuronal con capas ocultas y una capa de salida. Se utilizaron funciones de activación adecuadas y se ajustaron los parámetros de entrenamiento para obtener un modelo óptimo. La red neuronal utilizada en el modelo se construyó utilizando la biblioteca Keras, que proporciona una interfaz de alto nivel para construir y entrenar redes neuronales. La arquitectura de la red neuronal se definió de la siguiente manera:

Tabla 1

Arquitectura de red neuronal

Capa	**Tipo**	**Número de neuronas**	**Función de Activación**
Entrada	Dense	64	ReLU
Oculta	Dense	32	ReLU
Salida	Dense	1	Sigmoid

Nota. Detalle de arquitectura de la red neuronal definida.

Parámetros de compilación:

- Función de pérdida (loss function): Binary Crossentropy
- Optimizador: Adam
- Métricas de evaluación: Accuracy

3.4.2. Árbol de decisión

Un árbol de decisión es un modelo que utiliza una estructura de árbol para tomar decisiones basadas en condiciones y características de los datos de entrada. Cada nodo del árbol representa una característica o atributo, y las ramas representan las posibles decisiones o resultados. En el caso de estudio, se construyó un árbol de decisión utilizando los atributos relevantes del conjunto de datos preparado. Se aplicaron técnicas de poda y ajuste de hiper parámetros para mejorar la precisión y evitar el sobreajuste. El modelo de árbol de decisión utilizado se implementó utilizando el clasificador DecisionTreeClassifier de la biblioteca scikit-learn.

3.4.3. Regresión Logística

La regresión logística es un modelo de clasificación que utiliza la función logística para estimar la probabilidad de pertenencia a una clase o categoría. En el caso de estudio, se aplicó la regresión logística para predecir la probabilidad de abandono de los clientes. Se ajustaron los coeficientes del modelo utilizando los datos de entrenamiento y se aplicaron técnicas de regularización para evitar el sobreajuste y mejorar la generalización.

3.5. Técnicas de Análisis de Datos

3.5.1. CRISP-DM

En esta sección se detalla el proceso metodológico del proyecto. Se ha elegido la metodología CRISP-DM (Cross Industry Standard Process for Data Mining) la cual es ampliamente reconocida y utilizada en proyectos de minería de datos debido a su enfoque estructurado y sistemático. Esta metodología proporciona un marco de trabajo completo que abarca desde la comprensión del negocio hasta el despliegue de los modelos de predicción.

La elección de utilizar la metodología CRISP-DM se basa en varias razones. En primer lugar, CRISP-DM promueve un enfoque iterativo y cíclico, lo que permite ajustar y mejorar el proceso a medida que se obtienen nuevos conocimientos y resultados. Esto es

particularmente relevante en un proyecto de minería de datos donde la exploración y el análisis de los datos pueden revelar patrones y relaciones complejas que requieren iteraciones adicionales.

Además, CRISP-DM enfatiza la importancia de la comprensión del negocio y la colaboración con los expertos del dominio. Dado que el objetivo de este proyecto es identificar clientes con alta probabilidad de abandono en el retail farmacéutico, es esencial comprender los objetivos y las necesidades del negocio farmacéutico, así como trabajar en estrecha colaboración con el departamento de mercadeo. La metodología CRISP-DM proporciona una estructura para esta interacción efectiva y garantiza que los resultados obtenidos sean relevantes y útiles para el negocio.

Otra ventaja de CRISP-DM es su enfoque holístico, que cubre todas las etapas clave de un proyecto de minería de datos, desde la comprensión del negocio y la comprensión de los datos hasta la evaluación y el despliegue de los modelos. Al seguir esta metodología, se asegura una cobertura completa de las actividades necesarias para el éxito del proyecto y se garantiza una secuencia lógica y ordenada de las fases.

3.5.2. R VS PYTHON

A continuación, se proporciona información detallada sobre el árbol de decisión:

Tabla 2

Cuadro comparativo R y Python

Característica	R	Python
Tipo de lenguaje	Lenguaje específico para análisis estadístico y visualización de datos, orientado a tareas analíticas (Arif & Basit, 2017).	Lenguaje de programación multipropósito, utilizado para desarrollo de software y análisis de datos (VanderPlas, 2016).
Curva de aprendizaje	Curva de aprendizaje empinada, enfocado en usuarios con conocimiento estadístico (Peng, 2020).	Curva de aprendizaje más suave debido a su sintaxis simple (Sengupta, 2021).

Usuarios típicos	Científicos de datos, estadísticos, analistas cuantitativos (Arif & Basit, 2017).	Desarrolladores web, ingenieros de software, científicos de datos (VanderPlas, 2016).
Orientación	Orientado a objetos pero principalmente procedimental (Peng, 2020).	Totalmente orientado a objetos (Sengupta, 2021).
Manejo de datos	Potente manejo de estructuras de datos para análisis estadístico (Peng, 2020).	Flexible con diversos formatos de datos, requiere más código para análisis (VanderPlas, 2016).
Visualización	Visualizaciones estadísticas integradas de forma nativa (Arif & Basit, 2017).	Requiere bibliotecas adicionales como Matplotlib (Sengupta, 2021).
Código	Sintaxis más compleja y verborrágica (Peng, 2020).	Sintaxis limpia y sencilla (VanderPlas, 2016).
Comunidad	Comunidad especializada en estadística (Sengupta, 2021).	Comunidad muy grande y diversa (VanderPlas, 2016).
Uso en la industria	Nicho de análisis de datos y estadística (Arif & Basit, 2017).	Ampliamente utilizado en web, software y IA (Sengupta, 2021).

Nota. R estadística; Python propósito general.

Para el desarrollo de este proyecto se ha optado por utilizar el lenguaje de programación Python, en lugar de R u otros lenguajes, por varias razones: Python tiene una curva de aprendizaje más suave que hace que sea más accesible para un rango más amplio de programadores. Además, Python es un lenguaje multipropósito, lo que permitirá adaptarlo mejor a las diferentes etapas y componentes que requiere nuestro proyecto. Cuenta con una gran comunidad activa que facilita encontrar apoyo y bibliotecas para una amplia variedad de aplicaciones en ciencia de datos, aprendizaje automático e inteligencia artificial. Python también tiene una sintaxis clara y legible que hará que el código del proyecto sea más mantenible. Dadas estas ventajas, el uso de Python resulta adecuado para las necesidades de este proyecto.

CAPÍTULO IV: PRESENTACIÓN Y ANÁLISIS DE DATOS

4.1. Fase 1: Comprensión del Negocio

En la Fase 1 de la metodología CRISP-DM, se lleva a cabo la comprensión del negocio para establecer los objetivos, evaluar la situación actual, fijar los objetivos a nivel de minería de datos y obtener un plan de proyecto. Las actividades realizadas en esta fase son las siguientes:

4.1.1. Objetivos de negocio:

Los objetivos de negocio establecidos para este proyecto de minería de datos son los siguientes:

- Identificar los clientes con alta probabilidad de abandono (Churn) en la empresa farmacéutica.
- Optimizar las acciones de retención y fidelización de clientes.
- Desarrollar un algoritmo de predicción que permita determinar clientes con perfil de probabilidad alta de abandono.

4.1.2. Situación actual:

Se llevó a cabo una evaluación de la situación actual de la empresa farmacéutica en términos de retención de clientes y gestión del Churn. Se analizaron los datos históricos de transacciones de clientes y se identificaron áreas de mejora. Se encontró que la empresa enfrenta desafíos en la retención de clientes y busca implementar estrategias más efectivas para identificar a aquellos con alta probabilidad de abandono.

4.1.3. Objetivos a nivel de minería de datos:

- Identificar patrones de comportamiento de los clientes propensos a Churn en función de su historial de transacciones y características demográficas.
- Desarrollar modelos de predicción de Churn basados en técnicas de red neuronal,

árbol de decisión y regresión logística.

- Evaluar y comparar el rendimiento de los modelos de predicción para determinar el más efectivo en la detección de clientes propensos a Churn.

4.1.4. Plan de proyecto:

Se elaboró un plan de proyecto detallado que incluye los recursos necesarios, el cronograma y las actividades específicas a realizar en las siguientes fases. El plan establece las etapas de recolección y preparación de datos, modelado, evaluación y despliegue del modelo seleccionado.

Al finalizar la Fase 1, se obtuvo una comprensión clara del negocio, con objetivos establecidos para el proyecto de minería de datos, una evaluación de la situación actual de la empresa farmacéutica y un plan de proyecto detallado. Estos elementos proporcionan una base sólida para avanzar a la siguiente fase de la metodología CRISP-DM, la Comprensión de los datos.

4.2. Fase 2: Comprensión de los datos

En esta fase, se llevan a cabo los procesos de captura de datos, se proporciona una descripción del juego de datos, se realizan tareas de exploración de datos y se gestiona la calidad de los datos identificando problemas y proporcionando soluciones. A continuación, se describe detalladamente cada actividad de esta fase, basándose en la información proporcionada sobre la base de datos del caso de estudio:

4.2.1. Captura de datos:

En este proyecto, se obtuvo información de los datos de transacciones de los clientes del retail farmacéutico. Estos datos abarcan el período desde marzo de 2022 hasta febrero de 2023. Los registros de cada compra incluyen información como el número de identificación del cliente, la fecha de compra, el valor subtotal de la compra, si el cliente está afiliado al retail farmacéutico (variable "is_spoonity"), el género del cliente y la edad del cliente.

4.2.2. Descripción de la base de datos:

El dataset consta de un total de 2,869,628 registros de compras de clientes. Las variables disponibles en el dataset son las siguientes:

- **Número de identificación***:* Identificador único del cliente (cédula en su mayoría).
- **Fecha de compra:** Fecha en la que se realizó la compra.
- **Subtotal***:* Valor de la compra realizada por el cliente.
- **is_spoonity:** Indica si el cliente está afiliado al retail farmacéutico (1 si está afiliado, 0 si no).
- **Género:** Género del cliente (Hombre, Mujer).
- **Edad:** Edad del cliente (datos de clientes a partir de los 18 años)

Figura 1

Variables Dataset

	fecha_compra	numero_identificacion	is_spoonity	subtotal	genero	edad
0	2022-06-01	705439859	True	13.693500	FEMENINO	26
1	2022-07-25	1105059396001	False	-5.134554	MASCULINO	32
2	2022-06-19	703412403	False	0.555993	MASCULINO	44
3	2022-06-10	1102563366	False	6.120000	MASCULINO	57
4	2022-06-09	1900584978	False	15.312500	FEMENINO	30
...	...	...	...	...	...	...
2869623	2023-02-28	704317338	False	0.955134	FEMENINO	74
2869624	2023-02-28	701903643	False	2.687800	MASCULINO	21
2869625	2023-02-28	952941441	False	2.731800	MASCULINO	29
2869626	2023-02-28	703750133	True	0.998036	FEMENINO	45
2869627	2023-02-28	705918480	False	12.232900	MASCULINO	27

2869628 rows × 6 columns

Nota. Autoría propia, la información representa los primeros y últimos cuatro registros.

4.2.3. Exploración de datos:

En esta actividad, se llevan a cabo diferentes tareas para explorar y comprender los datos en profundidad. Primero se agrupan los datos para obtener la información de forma centralizada, es decir por cada cliente. Las nuevas variables que se obtienen son:

- **fecha_compra_primera:** fecha de la primera compra de cada cliente.
- **fecha_compra_ultima:** fecha de la última compra de cada cliente.

- **fecha_compra_penultima:** fecha penúltima compra de cada cliente.
- **cantidad_compras_count:** cantidad de compras que hizo el cliente.
- **subtotal:** suma de los valores totales de compra.
- **fecha_max:** fecha máxima del dataset.
- **dias_transcurridos:** cantidad de dias transcurridos entre la fecha de la última compra del cliente con relación a la fecha máxima del dataset.
- dias_transcurrido = fecha_max - fecha_compra_ultima
- **generacion:** se usa la variable edad para asignar a una respectiva categoría:

  - “Generacion_Silenciosa" (edad>=75)

  - "Baby_Boomers" (57 <= edad< 75)

  - "Generacion_X" (41 <= edad < 57)

  - "Millennials" (25 <= edad < 41)

  - "Generacion_Z (9 <= edad < 25)

Figura 2

Agrupación de datos

```
from datetime import timedelta
from datetime import datetime as dt
df=df_subset

# Convertir la columna de fecha_compra a formato de fecha
df['fecha_compra'] = pd.to_datetime(df['fecha_compra'], format='%Y-%m-%d')

# Agrupar los datos por número de identificación y obtener estadísticas de interés
df_clientes = df.groupby('numero_identificacion').agg({
    'fecha_compra': ['min', 'max', lambda x: x.nlargest(2).nsmallest(1).iloc[0],'nunique' ],
    # fecha de la primera y última compra
    'subtotal': 'sum', # total de compras
    'is_spoonity': 'max', # si el cliente está registrado en el programa de beneficios
    'genero': 'first',
    'edad':'first',
    'rango_salarios':'first'

}).reset_index()
df_clientes.columns = ['numero_identificacion','fecha_compra_primera', 'fecha_compra_ultima',
                       'fecha_compra_penultima',"cantidad_compras_count","subtotal",
                       "is_spoonity","genero",'edad','rango_salarios']
```

Nota. Autoría propia.

Figura 3

Nuevas variables

```python
# Calcular la fecha maxima
fecha_max = df_clientes['fecha_compra_ultima'].max()+ timedelta(days=1)
df_clientes['fecha_max'] =fecha_max
# Calcular la cantidad de días desde la última compra hasta la fecha actual
df_clientes['dias_transcurridos'] = (fecha_max - df_clientes['fecha_compra_ultima']).dt.days
```

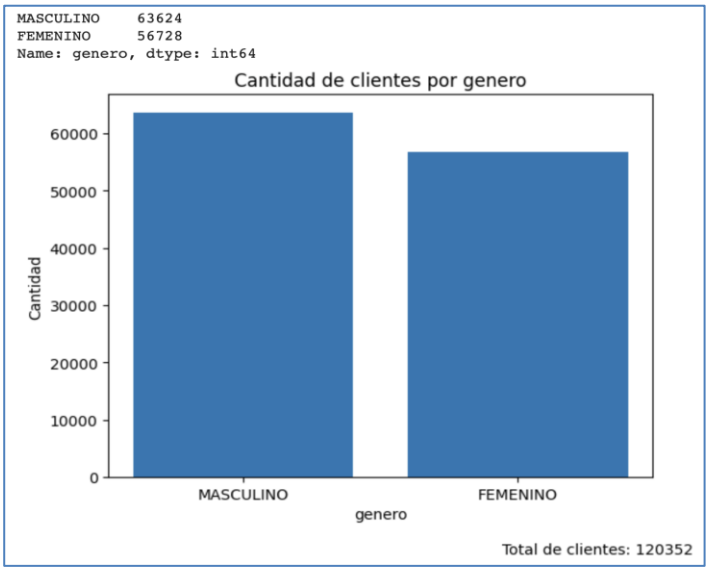
```python
##Generaciones
# Definir la función para asignar la generación
def asignar_generacion(edad):
    if edad >= 75:
        return "Generacion_Silenciosa"
    elif 57 <= edad < 75:
        return "Baby_Boomers"
    elif 41 <= edad < 57:
        return "Generacion_X"
    elif 25 <= edad < 41:
        return "Millennials"
    elif 9 <= edad < 25:
        return "Generacion_Z"
    else:
        return "Generacion_Alpha"

# Crear una columna 'generacion' basada en la columna de edad
df_clientes['generacion'] = df_clientes['edad'].apply(asignar_generacion)
```

Nota. Autoría propia. Variables fecha_max, dias_transcurridos y generación.

Análisis de la distribución de las variables: Se examina la distribución de las variables como la edad, género y el subtotal de las compras para comprender su comportamiento.

Figura 4

Cantidad de clientes por género



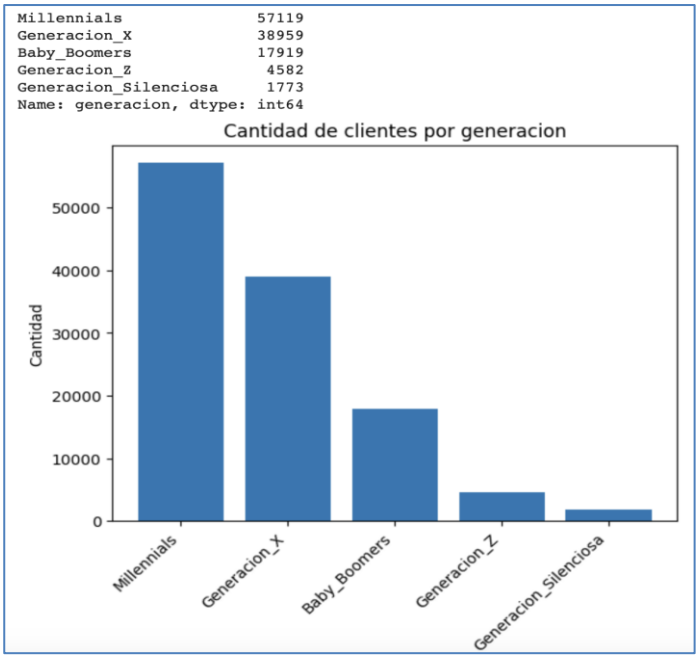
Nota. Autoría propia, la variable genero presenta una mayoría de clientes masculinos.

Por otro lado, la distribución de las generaciones es la siguiente:

Figura 5

Cantidad de clientes por generación

```
Millennials              57119
Generacion_X             38959
Baby_Boomers             17919
Generacion_Z              4582
Generacion_Silenciosa     1773
Name: generacion, dtype: int64
```



Nota. Autoría propia, la variable generación presenta una mayoría de clientes Millennials.

Figura 6

Código de gráficas de barras cantidad de clientes por edad y generación

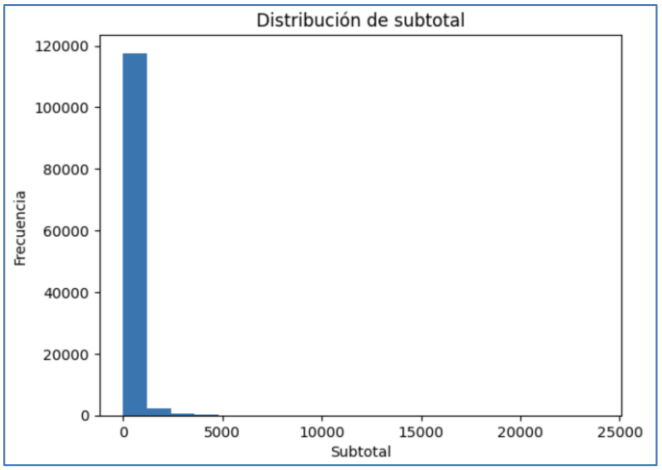
```python
import matplotlib.pyplot as plt

def plot_column_counts_1(df, column_name):
    column_counts = df[column_name].value_counts()
    total_count = column_counts.sum()
    print(column_counts)
    plt.bar(column_counts.index, column_counts.values)
    plt.xlabel(column_name)
    plt.ylabel("Cantidad")
    plt.title(f"Cantidad de clientes por {column_name}")
    plt.text(0.9, -0.2, f"Total de clientes: {total_count}", transform=plt.gca().transAxes, ha='center')
    plt.show()

def plot_column_counts_2(df, column_name):
    column_counts = df[column_name].value_counts()
    total_count = column_counts.sum()
    print(column_counts)
    plt.bar(column_counts.index, column_counts.values)
    plt.xlabel(column_name)
    plt.ylabel("Cantidad")
    plt.title(f"Cantidad de clientes por {column_name}")
    plt.xticks(rotation=45, ha='right')
    plt.show()

plot_column_counts_1(df_clientes, "genero")
plot_column_counts_2(df_clientes, "generacion")
```

Nota. Autoría propia.

Finalmente, la distribución de los valores de compra se muestra en el siguiente histograma:

Figura 7

Distribución Subtotal



Nota. Distribución de los valores de compra.

Figura 8

Código de gráficas de barras distribución subtotal

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

plt.hist(df_clientes['subtotal'], bins=20)
plt.xlabel('Subtotal')
plt.ylabel('Frecuencia')
plt.title('Distribución de subtotal')
plt.show()
```

Nota. Autoría propia.

4.3. Fase 3: Preparación de los datos

En la Fase 3 de la metodología CRISP-DM, se lleva a cabo la preparación de los datos para su posterior uso en el modelado y evaluación. Esta fase implica la selección, limpieza, transformación y consolidación de los datos. A continuación, se detallan las actividades realizadas en esta fase:

4.3.1. Selección de datos relevantes:

En esta actividad, se seleccionaron los atributos relevantes para el análisis de clasificación de clientes propensos a Churn. Se consideraron variables como el género, la generación, el subtotal de las compras y la afiliación a la farmacia (is_spoonity). Otros atributos que aportan información redundante fueron excluidos, por ejemplo, la variable edad y generación ya que para efectos de análisis la variable generación agrupa de mejor manera a

la variable edad. Esto fue realizado usando función drop, la cual elimina la columna mencionada. Por ejemplo:

Figura 9

Código función drop

```
# Eliminar la columna "edad"
df_clientes = df_clientes.drop(columns=['edad'])
```

Nota. Autoría propia.

Para generar representación de las variables se usa el siguiente código:

Figura 10

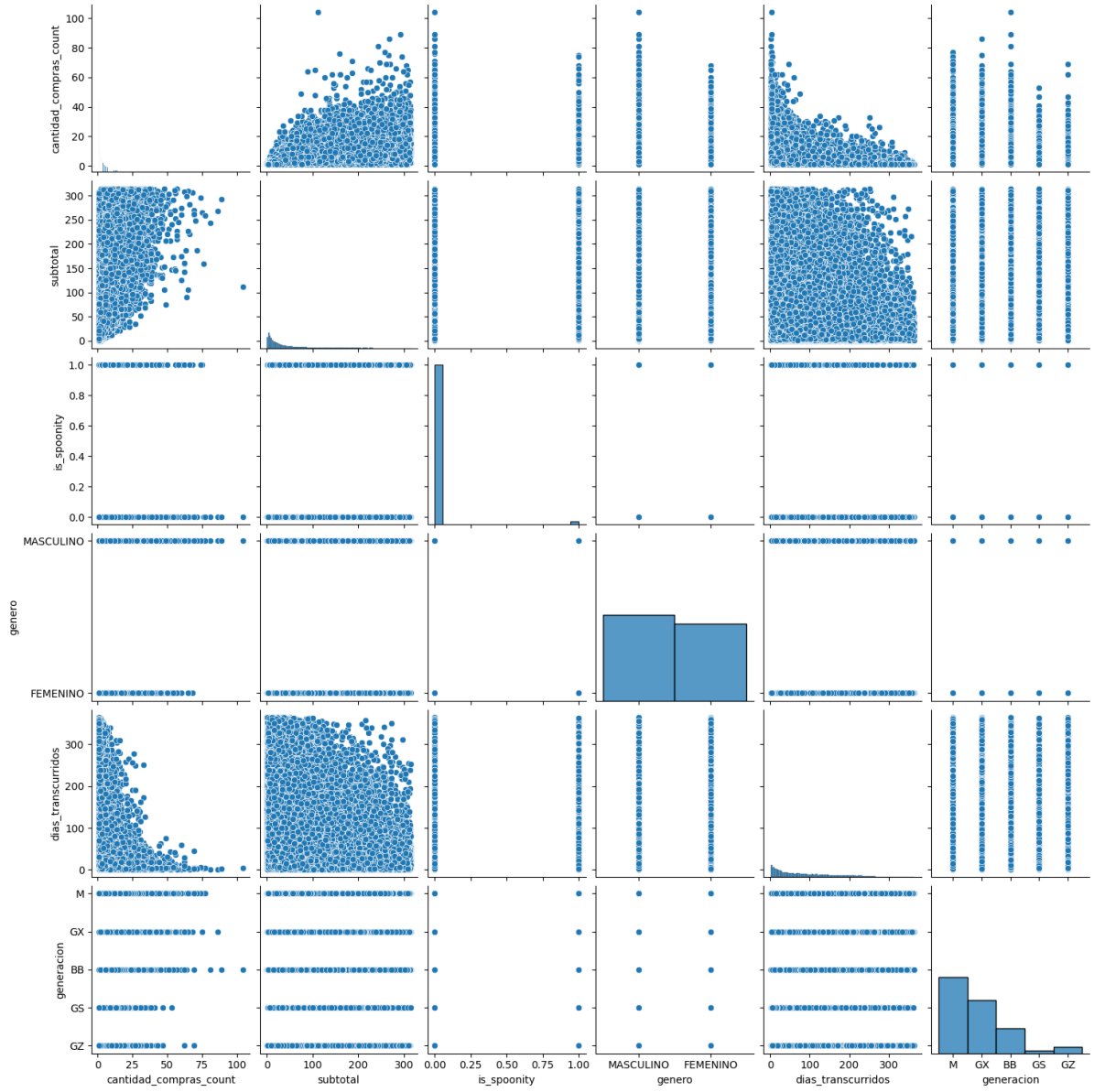
Código de presentación de variables

```
import seaborn as sns

sns.pairplot(df_clientes, vars = ['cantidad_compras_count', 'subtotal',
       'is_spoonity', 'genero',
       'dias_transcurridos', 'generacion'])
plt.savefig('output.png')
```

Nota. Autoría propia.

Figura 11

Gráfico de variables



Nota. Autoría propia. Gráfica de presentación entre algunas variables.

Esta gráfica es una matriz de gráficas la cual muestra, en cada celda, un scatter plot para cada par de variables seleccionadas. Cada punto en el scatter plot representa un cliente y su posición en el gráfico está determinada por los valores de las dos variables correspondientes. Por ejemplo, si hay una correlación positiva entre 'cantidad_compras_count' y 'subtotal', los puntos tenderán a estar agrupados en una línea ascendente en el scatter plot correspondiente. De manera similar, si se detecta una relación entre 'genero' y 'is_spoonity', los puntos se agruparán en grupos diferentes en función del género y el uso de "Spoonity".

Rango Salarios:

Esta variable es calculada usando cuartiles definiendo el nivel de salarios en base al nivel de compras, donde las etiquetas son: “Bajo”, “Medio Bajo”, “Medio Alto” y “Alto”. El código es el siguiente:

Figura 12

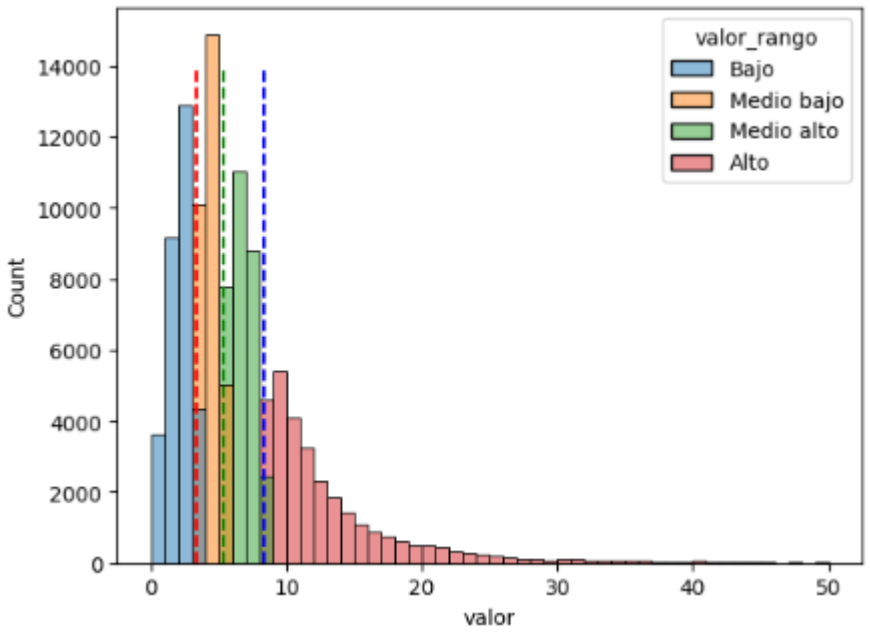
Código rango de salarios

```
# Crear una nueva columna con los cuartiles de la variable "valor"
df_final['nuevo_rango_salarios'] = pd.qcut(df_final['subtotal'], q=4, labels=['Bajo', 'Medio bajo', 'Medio alto', 'Alto'])
```

Nota. Autoría propia.

Figura 13

Gráfico rango de salarios



Nota. Autoría propia.

4.3.2. Limpieza de datos:

Durante esta actividad, se abordaron problemas de calidad de los datos, como valores faltantes, inconsistencias y errores. Se corrigieron valores incorrectos o inconsistentes. Además, se llevaron a cabo acciones para eliminar duplicados y registros irrelevantes, garantizando así la integridad de los datos.

Identificación de valores atípicos:

Se buscarán valores inusuales o extremos que puedan afectar el análisis posterior.

En el proceso de exploración de datos, se identificó la presencia de outliers en la variable "subtotal" (Figura 16), que representa el valor de las compras realizadas por los clientes. Para determinar la presencia de outliers, se utilizó la prueba de Shapiro-Wilk para evaluar la normalidad de la distribución de los datos. Esta prueba estadístico comprueba si una muestra de datos sigue una distribución normal.

Al aplicar la prueba de Shapiro-Wilk a los datos de la variable "subtotal", se obtuvo un valor de significancia (p-value) menor a 0.05, lo que indica que los datos no siguen una distribución normal. Esto sugiere la presencia de valores atípicos en la variable.

El código es el siguiente:

Figura 14

Código Shapiro-Wilk

```
# valores atípicos basado en el rango intercuartílico (IQR).
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import shapiro
# Definir la columna de interés
columna = 'subtotal'

# Realizar la prueba de Shapiro-Wilk
stat, p = shapiro(df_clientes[columna])

# Imprimir los resultados
print('Estadístico de prueba:', stat)
print('Valor p:', p)


# Definir el nivel de significancia
significancia = 0.05

# Comprobar si los datos siguen una distribución normal
if p > significancia:
    print("Los datos de la columna", columna, "siguen una distribución normal.")
else:
    print("Los datos de la columna", columna, "no siguen una distribución normal.")
```

Nota. Autoría propia.

Para abordar este problema, se decidió utilizar el método del rango intercuartílico (IQR, por sus siglas en inglés) para la limpieza de datos y la identificación de outliers. El rango intercuartílico se define como la diferencia entre el tercer cuartil (Q3) y el primer cuartil (Q1) de un conjunto de datos. Se calcula utilizando la siguiente fórmula:

$$IQR = Q3 - Q1$$

Para identificar los outliers, se aplicó la regla del rango intercuartílico, que considera cualquier valor por debajo de Q1 - 1.5 * IQR o por encima de Q3 + 1.5 * IQR como un outlier potencial. Estos valores se consideran atípicos debido a su distancia significativa respecto a la distribución de los datos.

En este caso se tiene que:

- Rango superior: 314.84…
- Rango inferior: -172.04…

Una vez identificados los outliers, se procedió a su eliminación del conjunto de datos. Esto permitirá obtener una distribución más representativa y reducirá el impacto de valores atípicos en los análisis posteriores. La eliminación de outliers asegurará la integridad y la calidad de los datos utilizados en el proceso de modelado y evaluación del proyecto de minería de datos. Todo este proceso se hace usando las siguientes líneas de código:

Figura 15

Código cálculo rango intercuartil

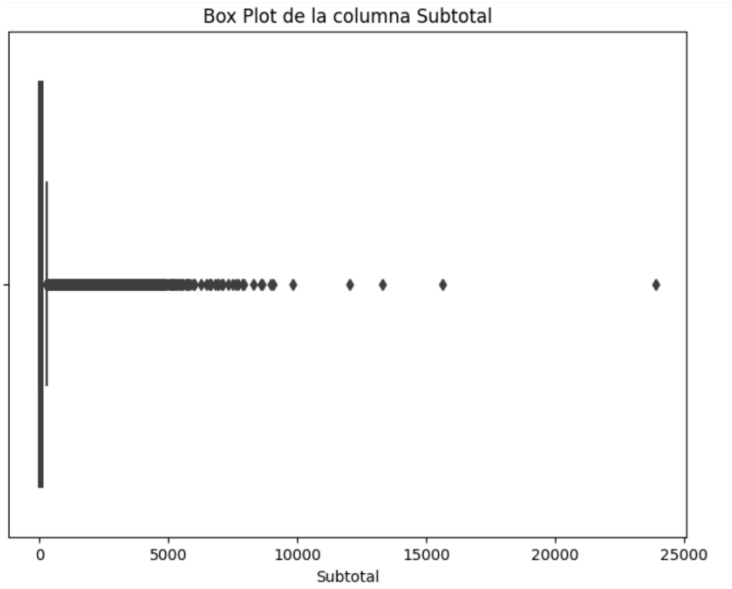
```
# Calcular el rango intercuartil (IQR)
Q1 = df_clientes[columna].quantile(0.25)
Q3 = df_clientes[columna].quantile(0.75)
IQR = Q3 - Q1

limite_inferior = Q1- 1.5 * IQR
limite_superior = Q3+ 1.5 * IQR

# Identificar los valores que son outliers
outliers = df_clientes[(df_clientes[columna] < limite_inferior) | (df_clientes[columna] > limite_superior)]

# Filtrar el DataFrame excluyendo los outliers
df_sin_outliers = df_clientes[(df_clientes[columna] >= limite_inferior) & (df_clientes[columna] <= limite_superior)]
```

Nota. Autoría propia.

La gráfica box plot que muestra los datos (outliers) es la siguiente:

Figura 16

Gráfico Blox Plot

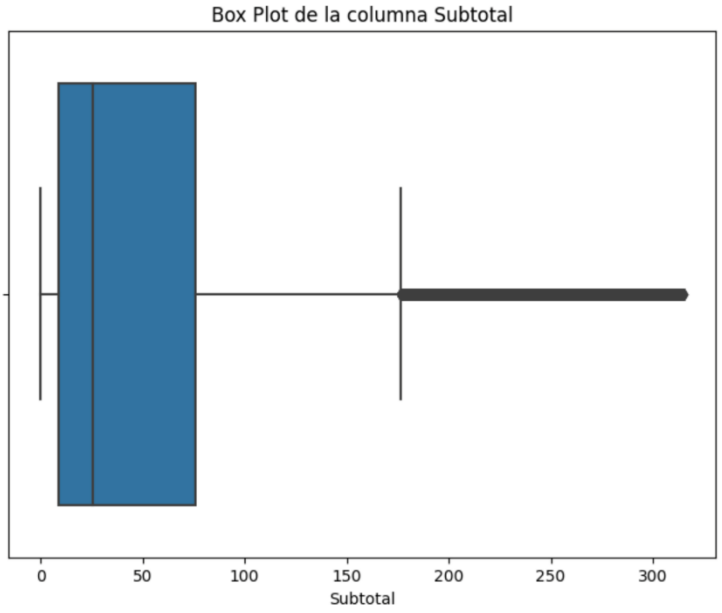


Nota. Autoría propia. Datos con Outliers.

La gráfica de box plot luego de quitar los outliers es la siguiente:

Figura 17

Gráfico Blox Plot



Nota. Autoría propia. Datos sin Outliers.

Esta muestra la distribución de la variable subtotal. Ambas gráficas son creadas de la siguiente forma:

Figura 18

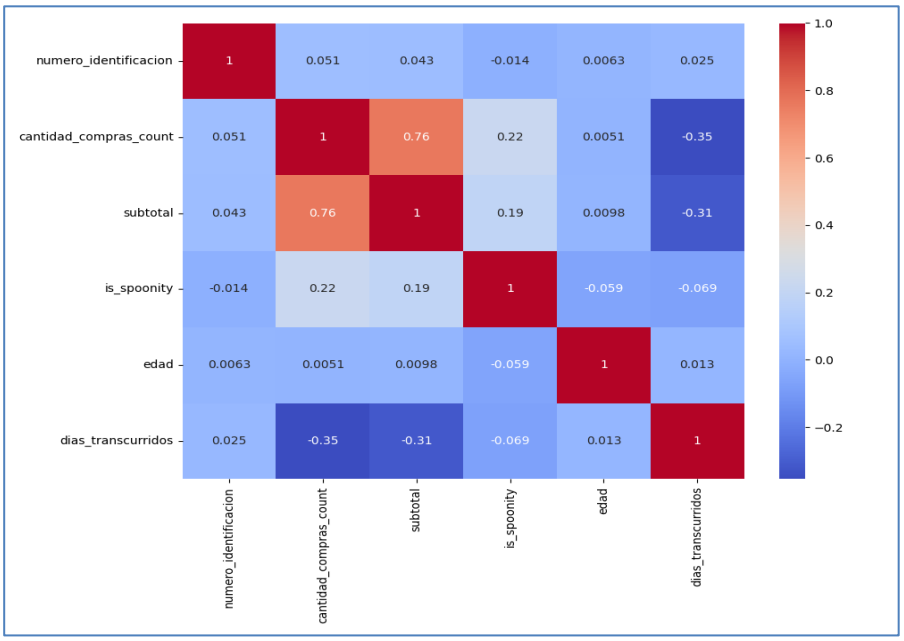
Código Blox Plot

```
# Crear el box plot
plt.figure(figsize=(8, 6))
sns.boxplot(x=df_sin_outliers['subtotal'])
plt.xlabel('Subtotal')
plt.title('Box Plot de la columna Subtotal')

# Mostrar el box plot
plt.show()
```

Nota. Autoría propia.

Figura 19

Análisis de correlación



Nota. Autoría propia. Se explora la correlación entre variables para identificar posibles relaciones significativas. Donde las variables no están altamente correlacionadas, excepto por la cantidad de compras con el subtotal, la cual se podría regularizar en la fase de modelados automáticamente con los modelos propuestos.

Figura 20

Código de creación de gráficos de correlación

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Calcula la matriz de correlación
correlation_matrix = df_clientes.corr()

# Crea una figura y un eje
fig, ax = plt.subplots(figsize=(10, 8))

# Genera el mapa de calor de la matriz de correlación
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", ax=ax)

# Ajusta el espaciado de la figura
plt.tight_layout()
plt.savefig('output3.png')
# Muestra la gráfica
plt.show()
```

Nota. Autoría propia.

4.3.3. Definición de umbrales (Churn etiquetado)

En esta etapa se decide aplicar 5 técnicas para definir umbrales e identificar clientes con alta probabilidad de abandono. El objetivo es que con estas 5 técnicas al final se tome una decisión de si el cliente es propenso a abandonar o no. Las técnicas son:

- Umbral Manual
- Umbral Mediana
- Umbral Distribución
- Umbral Análisis RFM
- Umbral Análisis de comportamiento

Umbral Manual:

Este umbral de días transcurridos se hace en base a la experiencia del negocio. Para este trabajo fue elegido un valor umbral igual a 150.

Figura 21

Código para introducir umbral manual en el data frame

```python
## UMBRAL MANUAL
umbral_manual=150

# Definimos una función para determinar el valor del target
def determinar_abandono_manual(row):
    if row['dias_transcurridos'] > umbral_manual:
        return 1
    else:
        return 0
# Agregamos el target al dataframe
df_clientes["target_manual "+str(umbral_manual)] = df_clientes.apply(determinar_abandono_manual, axis=1)
```

Nota. Autoría propia.

Umbral Mediana:

Se calcula la mediana de los días transcurridos, en esta base la mediana tiene un valor igual a 67 días.

Figura 22

Código para introducir umbral mediana en el data frame

```
# Definir el umbral de abandono como la mediana de los días transcurridos
umbral_abandono_mediana = df_clientes['dias_transcurridos'].median()
# Definir la función para determinar el target
def determinar_abandono_mediana(row):
  if (row['dias_transcurridos'] - umbral_abandono_mediana) > 0:
        return 1
  else:
        return 0
    #return (row['dias_transcurridos'] - umbral_abandono) > 0

# Agregar la columna de target al dataframe de clientes
df_clientes['target_mediana '+str(umbral_abandono_mediana)] = df_clientes.apply(determinar_abandono_mediana, axis=1)
```

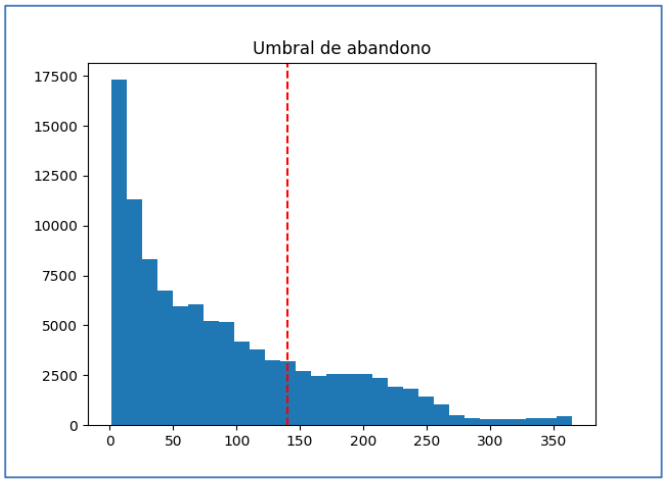
Nota. Autoría propia.

Umbral Distribución:

Se examina la distribución de los días transcurridos entre la última compra y la fecha actual para determinar el umbral de abandono. Por ejemplo, si la mayoría de los clientes realizan compras dentro de los 30 días, se define un umbral de 60 días para considerar a un cliente como abandonado. Para esto se usa el percentil 75, que en este caso es igual a 140 días.

Figura 23

Umbral de Abandono



Nota. Estos umbrales siguen principios básicos como la regla del codo para identificar comportamientos y son motivados por la experiencia del negocio. Con esos tres primeros umbrales se obtiene la siguiente distribución de clientes identificados como propensos al abandono y aquellos que no.

Figura 24

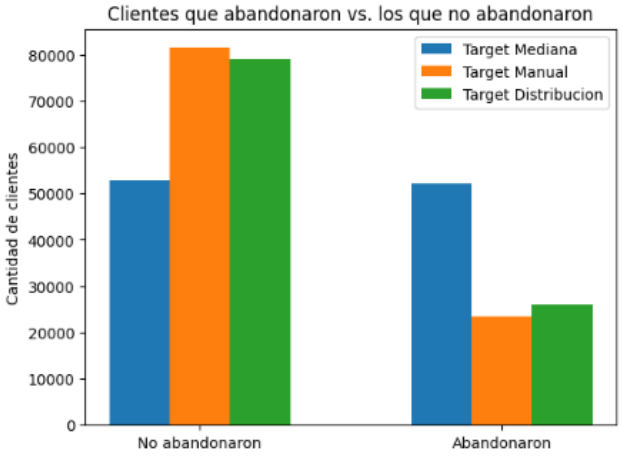
Código para introducir umbral distribución en el data frame y gráfico de distribución de umbral de abandono.

```
# Graficar la distribución de los días transcurridos
plt.hist(df_clientes['dias_transcurridos'], bins=30)
plt.title('Distribución de los días transcurridos')
plt.xlabel('Días transcurridos')
plt.ylabel('Frecuencia')
plt.show()

# Calcular el umbral de abandono
umbral_distribucion_dias_transcurridos = np.percentile(df_clientes['dias_transcurridos'], 75)


# Definimos una función para determinar el valor del target
def determinar_abandono_distribucion(row):
    if row['dias_transcurridos'] > umbral_distribucion_dias_transcurridos:
        return 1
    else:
        return 0
# Agregamos el target al dataframe
df_clientes["target_distribucion "+
  str(umbral_distribucion_dias_transcurridos)] = df_clientes.apply(determinar_abandono_distribucion, axis=1)

# Graficar el umbral de abandono
plt.hist(df_clientes['dias_transcurridos'], bins=30)
plt.axvline(umbral_distribucion_dias_transcurridos, color='r', linestyle='--')
plt.title('Umbral de abandono')
plt.xlabel('Días transcurridos')
plt.ylabel('Frecuencia')
plt.show()
```

Nota. Autoría propia.

Figura 25

Gráfico Clientes abandonaron vs No abandonaros

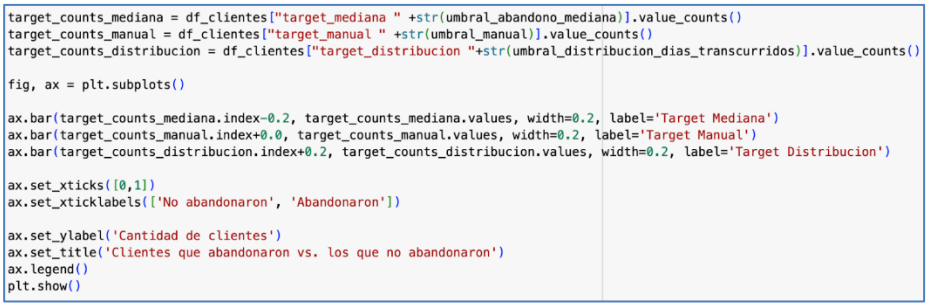


Nota. Estadística de cantidad de clientes que abandonan y no abandonas por análisis de

frecuencia de compra.

Figura 26

Código de creación de distribución de clientes identificados como propensos al abandono



```
target_counts_mediana = df_clientes["target_mediana " +str(umbral_abandono_mediana)].value_counts()
target_counts_manual = df_clientes["target_manual " +str(umbral_manual)].value_counts()
target_counts_distribucion = df_clientes["target_distribucion "+str(umbral_distribucion_dias_transcurridos)].value_counts()

fig, ax = plt.subplots()

ax.bar(target_counts_mediana.index-0.2, target_counts_mediana.values, width=0.2, label='Target Mediana')
ax.bar(target_counts_manual.index+0.0, target_counts_manual.values, width=0.2, label='Target Manual')
ax.bar(target_counts_distribucion.index+0.2, target_counts_distribucion.values, width=0.2, label='Target Distribucion')

ax.set_xticks([0,1])
ax.set_xticklabels(['No abandonaron', 'Abandonaron'])

ax.set_ylabel('Cantidad de clientes')
ax.set_title('Clientes que abandonaron vs. los que no abandonaron')
ax.legend()
plt.show()
```

Nota. Autoría propia.

Umbral Análisis RFM:

Se lleva a cabo un análisis RFM (Recencia, Frecuencia, Monetario) para obtener una comprensión más profunda del comportamiento de los clientes. El análisis RFM es una técnica ampliamente utilizada en marketing que clasifica a los clientes en función de tres variables clave: la recencia de sus compras, la frecuencia con la que realizan compras y el valor monetario de sus compras.

En el contexto del caso de estudio de clasificación de clientes propensos a Churn en retail farmacéutico, se aplicó el análisis RFM para segmentar a los clientes según su comportamiento de compra. La variable de recencia representa el tiempo transcurrido desde la última compra realizada por un cliente, mientras que la variable de frecuencia indica la cantidad de compras realizadas en un período de tiempo determinado. La variable monetaria refleja el valor total de las compras realizadas por cada cliente. El código usado es el siguiente:

Figura 27

Código umbral análisis RFM

```
df['recency']=df['dias_transcurridos']
df['frequency'] = df['cantidad_compras_count']
df['monetary_value']=df['subtotal']

# Asignar valores numéricos a cada indicador
df['R'] = pd.qcut(df['recency'], q=4, labels=range(4, 0, -1))
df['F'] = pd.qcut(df['frequency'], q=2, labels=range(1,3))
df['M'] = pd.qcut(df['monetary_value'], q=4, labels=range(1,5))

# Concatenar los valores R, F, y M para crear el valor RFM completo
df['RFM'] = df['R'].astype(int) + df['F'].astype(int) + df['M'].astype(int)

columnas_deseadas=['recency','frequency','monetary_value','R','F','M','RFM']

# Calcular el percentil 80 de la distribución de RFM de los clientes existentes
percentil_80 = df['RFM'].quantile(q=0.8)

# Definimos una función para determinar el valor del target
def determinar_abandono_rfm(row):
    if row['RFM'] < percentil_80:
        return 1
    else:
        return 0
# Agregamos el target al dataframe
df["target_rfm "+str(percentil_80)] = df.apply(determinar_abandono_rfm, axis=1)
```

Nota. Autoría propia.

Mediante el análisis RFM, se obtuvieron segmentos de clientes basados en su comportamiento de compra. Estos segmentos pueden proporcionar información valiosa para el posterior modelado y clasificación de los clientes propensos a Churn.

Figura 28

Análisis RFM

	recency	frequency	monetary_value	R	F	M	RFM
79536	1	4	59.940141	4	2	3	9
84807	1	10	101.577678	4	2	4	10
50078	1	5	21.809100	4	2	2	8
17760	1	17	217.976311	4	2	4	10
75755	1	16	96.599846	4	2	4	10
...	...	...	...	...	...	...	...
119561	365	1	22.161764	1	1	2	4
59154	365	1	20.591342	1	1	2	4
107125	365	1	11.095000	1	1	2	4
117607	365	1	13.203750	1	1	2	4
20604	365	1	8.440100	1	1	1	3

105003 rows × 7 columns

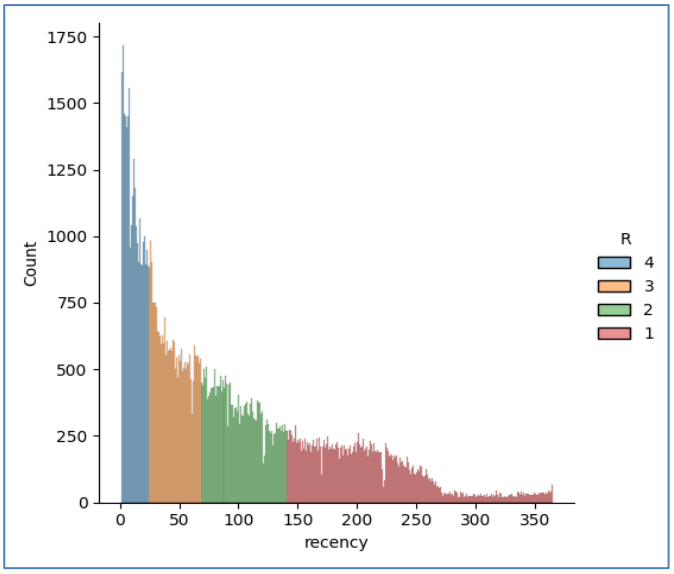
Nota. Aquellos clientes que presenten una baja recencia, baja frecuencia y bajo valor monetario son considerados como clientes con mayor probabilidad de abandono. Por otro lado, los clientes con alta recencia, alta frecuencia y alto valor monetario son identificados

como clientes leales y de alto valor para el retail farmacéutico.

Los valores de R, F y M son dados de acuerdo con la distribución de los datos. Los cuales se ven de la siguiente forma:

Figura 29

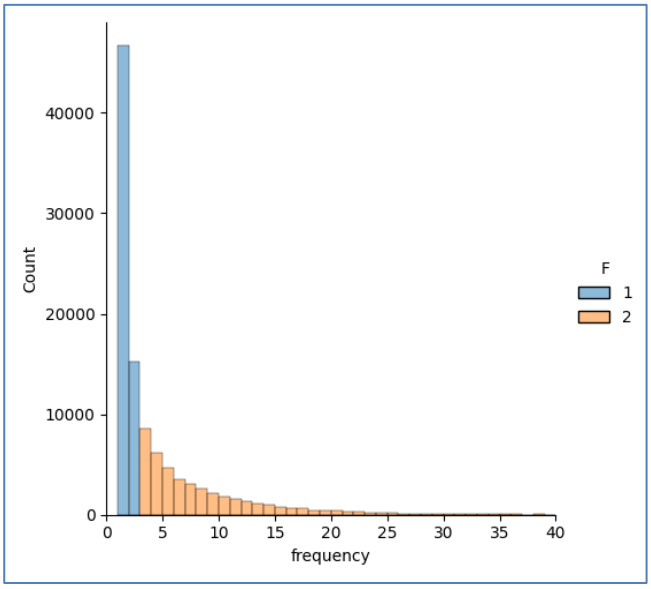
Análisis R



Nota. Autoría propia. Distribución de datos de análisis R.

Figura 30

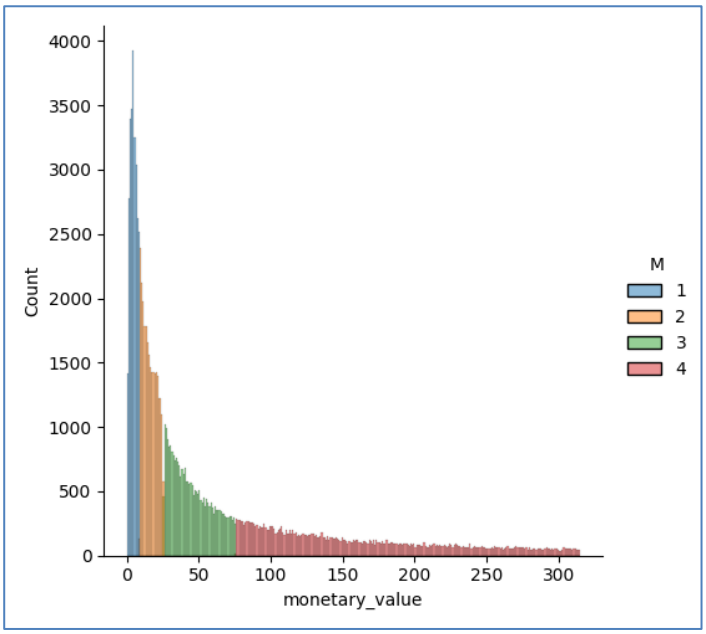
Análisis F



Nota. Autoría propia. Distribución de datos de análisis F.

Figura 31

Análisis M

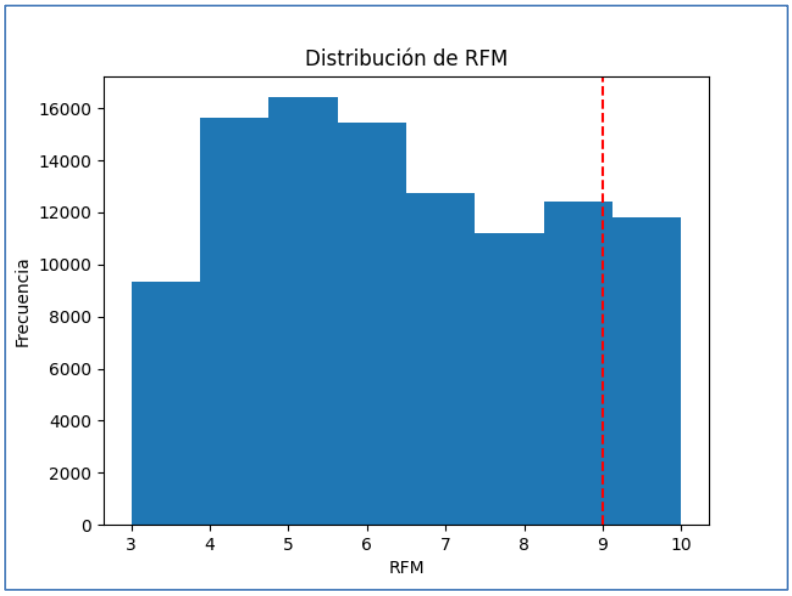


Nota. Autoría propia. Distribución de datos de análisis M.

Finalmente, usando el percentil 80, se elige el valor umbral de 9.

Figura 32

Distribución RFM



Nota. Autoría propia. Aquellos que su RFM=R+F+M sea igual o mayor que 9 son etiquetados como clientes con baja probabilidad de abandono.

Figura 33

Código gráfico distribución RFM

```
import pandas as pd
import matplotlib.pyplot as plt
# crear el histograma
plt.hist(df['RFM'], bins=8)

# agregar etiquetas y títulos
plt.axvline(percentil_80, color='r', linestyle='--')
plt.xlabel("RFM")
plt.ylabel("Frecuencia")
plt.title("Distribución de RFM")
# mostrar el gráfico
plt.show()
```

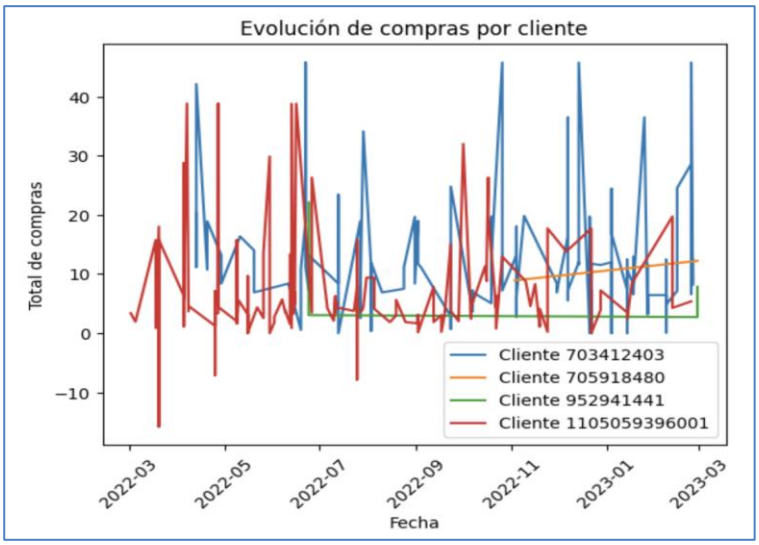
Nota. Autoría propia.

Umbral Análisis de comportamiento:

El enfoque propuesto utiliza el análisis de regresión lineal para identificar aquellos clientes que presentan una tendencia a disminuir su nivel de compras a lo largo del tiempo. La idea es analizar la relación entre la fecha de compra y el subtotal de cada transacción para cada cliente en particular.

Figura 34

Evolución de compras por cliente



Nota. Autoría propia. Se muestra una selección de clientes y se filtran los datos para obtener únicamente sus compras.

Con todos los datos de los clientes, se realiza un análisis de regresión lineal para cada cliente, donde la fecha de compra se considera como la variable independiente y el subtotal como la variable dependiente.

A partir de los coeficientes de la regresión, específicamente el coeficiente asociado a la fecha se determina si existe una tendencia negativa en las compras del cliente. Si el coeficiente es negativo, indica que a medida que avanza el tiempo, las compras tienden a disminuir. Por otro lado, si el coeficiente es positivo, sugiere que no hay una tendencia clara de disminución en las compras del cliente.

El resultado final, una columna adicional llamada "target_tendencia_disminucion", que indica con un valor de 1 si cada cliente presenta una tendencia a disminuir sus compras o 0 si no.

Donde de los clientes, un total de 79040 no tienen tendencia negativa y 41312 si la tienen. Este enfoque ayuda a identificar de manera objetiva aquellos clientes que están disminuyendo su nivel de compras a lo largo del tiempo.

El código para realizar el análisis es el siguiente:

Figura 35

Código análisis comportamiento

```
import statsmodels.api as sm
# Crear un nuevo dataframe para almacenar los resultados
df_resultados = pd.DataFrame(columns=['numero_identificacion', 'tendencia_disminucion'])

# Realizar análisis de regresión para cada cliente
for cliente, datos_cliente in test_df.groupby('numero_identificacion'):
    # Obtener los valores de fecha y subtotal
    X = sm.add_constant(datos_cliente['fecha_compra'].astype(np.int64) // 10**9)
    y = datos_cliente['subtotal']

    # Ajustar el modelo de regresión lineal
    model = sm.OLS(y, X)
    results = model.fit()
    # Obtener el coeficiente de la fecha y determinar si hay una tendencia negativa
    coef_fecha = results.params['fecha_compra']
    tendencia_disminucion = True if coef_fecha < 0 else False
    # Agregar los resultados al dataframe de resultados
    df_resultados = df_resultados.append({'numero_identificacion': cliente,
                                          'tendencia_disminucion': tendencia_disminucion},
                                         ignore_index=True)
```

Nota. Autoría propia.

Las gráficas son generadas con el siguiente código:

Figura 36

Código de gráfico análisis de comportamiento

```
import pandas as pd
import matplotlib.pyplot as plt
# Seleccionar clientes específicos
clientes_seleccionados = [ 705918480,952941441,1105059396001,703412403]

# Filtrar los datos por clientes seleccionados
df_seleccionados = test_df[test_df['numero_identificacion'].isin(clientes_seleccionados)]

# Crear el gráfico de series de tiempo
fig, ax = plt.subplots()
for cliente, datos_cliente in df_seleccionados.groupby('numero_identificacion'):
    ax.plot(datos_cliente['fecha_compra'], datos_cliente['subtotal'], label=f'Cliente {cliente}')

# Agregar título y etiquetas de los ejes
ax.set_title('Evolución de compras por cliente')
ax.set_xlabel('Fecha')
ax.set_ylabel('Total de compras')

# Mostrar la leyenda
ax.legend()

# Rotar las etiquetas del eje x para mayor claridad
plt.xticks(rotation=45)

# Mostrar el gráfico
plt.show()
```

Nota. Autoría propia.

Target Final:

En esta sección se decide que aquellos clientes que, por los umbrales, haya sido marcado como posible cliente que abandona, será finalmente clasificado como uno.

Figura 37

Agrupación de umbrales

	numero_identificacion	target_manual 150	target_mediana 67.0	target_distribucion 140.0	target_rfm 9.0	target_tendencia_disminucion
0	1102900576	0	0	0	0	0
1	1103820542	0	0	0	0	1
2	705225159	0	0	0	1	0
3	701264194	0	0	0	0	1
4	1102107578	0	0	0	0	1
...	...	...	...	...	...	...
104998	704607589001	1	1	1	1	0
104999	706662848	1	1	1	1	0
105000	1758063752	1	1	1	1	0
105001	1900772094	1	1	1	1	0
105002	701806820	1	1	1	1	0

105003 rows × 6 columns

Nota. Definición de Target final al cumplir 3 (1) de 5 umbrales.

Donde esta nueva variable será identificada como “target_final”. El código usado es:

Figura 38

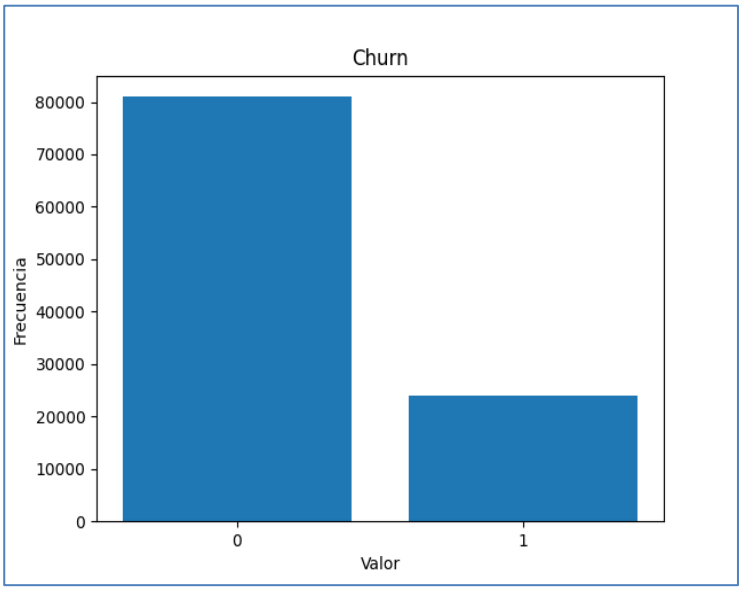
Código agrupación de target final

```python
def decision_final(row, umbral):
    suma = row.drop('numero_identificacion').sum()
    if suma > umbral:
        return 1
    else:
        return 0



umbral = 3 ¡
df_target['target_final'] = df_target.apply(decision_final, axis=1, args=(umbral,))
print(df_target)
```

Nota. Autoría propia.

Figura 39

Resultado Churn



Nota. Autoría propia. La siguiente gráfica muestra la diferencia entre clientes que finalmente fueron etiquetados como probables a abandonar (1) o no (0).

Figura 40

Código gráfico target final

```python
# contar el número de ocurrencias de cada valor
valores, conteos = np.unique(df_target['target_final'], return_counts=True)

# crear el histograma
plt.bar(valores, conteos)

# agregar etiquetas y títulos
plt.title('Churn')
plt.xlabel('Valor')
plt.ylabel('Frecuencia')

# Establecer las etiquetas del eje x con los valores únicos de la variable
plt.xticks(df_target['target_final'].unique())
# mostrar el histograma
plt.show()
```

Nota. Autoría propia.

4.3.4. Transformación de datos:

En esta etapa, se realizaron transformaciones en los datos para mejorar su calidad y adecuarlos a los requisitos del modelo. Esto incluyó la normalización de variables numéricas, como la edad y el subtotal de las compras, para que estuvieran en la misma escala. También se aplicaron técnicas de codificación, como la codificación one-hot, para variables categóricas como el género y la afiliación al retail farmacéutico.

Para comenzar, se identificaron las columnas categóricas en el conjunto de datos. Estas columnas representan variables como el género y la afiliación a la farmacia. A continuación, se aplicó la técnica de One Hot Encoding para convertir las variables categóricas en una representación numérica adecuada para su uso en modelos de machine learning. Una vez completada la codificación, se eliminaron las columnas categóricas originales de la base de datos, ya que ahora están representadas de manera adecuada por las columnas codificadas. El resultado final fue un dataframe llamado `df_encoded` que contiene tanto las variables numéricas originales como las columnas codificadas. Por ejemplo:

Figura 41

Resumen DataFrame `df_encoded`

genero_FEMENINO	genero_MASCULINO	generacion_Baby_Boomers	generacion_Generacion_Silenciosa	generacion_Generacion_X
1.0	0.0	0.0	0.0	1.0
0.0	1.0	0.0	0.0	1.0
1.0	0.0	0.0	0.0	0.0
0.0	1.0	0.0	0.0	1.0
1.0	0.0	1.0	0.0	0.0
...	...	...	...	...
1.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0
0.0	1.0	0.0	0.0	1.0
1.0	0.0	0.0	0.0	1.0
1.0	0.0	1.0	0.0	0.0

Nota. Contiene tanto las variables numéricas originales como las columnas codificadas.

Para continuar con el proceso de modelado, los datos se dividieron en conjuntos de

entrenamiento (80%) y prueba (20%). Esto permite evaluar la capacidad predictiva del modelo en datos no vistos previamente.

Además, se realizó la estandarización de las variables numéricas utilizando la técnica de escalamiento mediante la clase ‘StandardScaler’ del módulo ‘sklearn.preprocessing’. Esto asegura que todas las variables numéricas tengan una escala comparable, evitando así que una variable domine sobre las demás debido a sus magnitudes. Los conjuntos de entrenamiento (`X_train`) y prueba (`X_test`) fueron transformados utilizando el mismo escalador para mantener la consistencia en la escala de los datos. El código es el siguiente:

Figura 42

Código transformación de datos

```
# identificar columnas categóricas
categorical_cols = df.select_dtypes(include=['object']).columns

# aplicar One Hot Encoding a las columnas categóricas
onehot_encoder = OneHotEncoder(sparse=False)
encoded_cols = pd.DataFrame(onehot_encoder.fit_transform(df[categorical_cols]))
encoded_cols.columns = onehot_encoder.get_feature_names_out(categorical_cols)

# eliminar columnas categóricas originales del dataframe
df = df.drop(columns=categorical_cols)

# unir el dataframe original con las columnas codificadas
df_encoded = pd.concat([df, encoded_cols], axis=1)

# Dividir los datos en entrenamiento y prueba
X = df_encoded.drop(['numero_identificacion', 'target_final'], axis=1)
y = df_encoded['target_final']


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


display(df_encoded)
# Escalar las variables
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Nota. Autoría propia.

Al finalizar esta etapa de transformación de datos, se obtuvo un conjunto de datos preparado y listo para ser utilizado en el modelado y evaluación del proyecto de minería de datos. La transformación de variables categóricas mediante One Hot Encoding y la estandarización de las variables numéricas garantizan que los datos estén en un formato adecuado para aplicar diferentes algoritmos de clasificación y obtener resultados precisos y

confiables.

4.4. Fase 4: Modelado de los datos

En la Fase 4 de la metodología CRISP-DM, se procede a seleccionar y entrenar los modelos de machine learning utilizando los datos preparados en las etapas anteriores. En el caso de estudio, se utilizaron tres modelos: red neuronal, árbol de decisión y regresión logística.

A continuación, se proporciona una descripción de cada uno de estos modelos:

4.4.1. Red Neuronal:

La red neuronal es un modelo de aprendizaje profundo que se basa en la estructura del cerebro humano para realizar tareas de clasificación y predicción. Consiste en una red de nodos interconectados, llamados neuronas, que procesan la información y generan una salida.

Para el caso de estudio, se configuró una red neuronal con capas ocultas y una capa de salida. Se utilizaron funciones de activación adecuadas y se ajustaron los parámetros de entrenamiento para obtener un modelo óptimo. La red neuronal utilizada en el modelo se construyó utilizando la biblioteca Keras, que proporciona una interfaz de alto nivel para construir y entrenar redes neuronales. La arquitectura de la red neuronal se definió de la siguiente manera:

Tabla 3

Arquitectura de red neuronal

Capa	**Tipo**	**Número de neuronas**	**Función de Activación**
Entrada	Dense	64	ReLU
Oculta	Dense	32	ReLU
Salida	Dense	1	Sigmoid

Nota. Detalle de arquitectura de la red neuronal definida.

Parámetros de compilación:

- Función de pérdida (loss function): Binary Crossentropy
- Optimizador: Adam
- Métricas de evaluación: Accuracy

4.4.2. Árbol de decisión

Un árbol de decisión es un modelo que utiliza una estructura de árbol para tomar decisiones basadas en condiciones y características de los datos de entrada. Cada nodo del árbol representa una característica o atributo, y las ramas representan las posibles decisiones o resultados. En el caso de estudio, se construyó un árbol de decisión utilizando los atributos relevantes del conjunto de datos preparado. Se aplicaron técnicas de poda y ajuste de hiper parámetros para mejorar la precisión y evitar el sobreajuste. El modelo de árbol de decisión utilizado se implementó utilizando el clasificador DecisionTreeClassifier de la biblioteca scikit-learn.

4.4.3. Regresión Logística

La regresión logística es un modelo de clasificación que utiliza la función logística para estimar la probabilidad de pertenencia a una clase o categoría. En el caso de estudio, se aplicó la regresión logística para predecir la probabilidad de abandono de los clientes. Se ajustaron los coeficientes del modelo utilizando los datos de entrenamiento y se aplicaron técnicas de regularización para evitar el sobreajuste y mejorar la generalización.

4.5. Fase 5: Evaluación del modelo

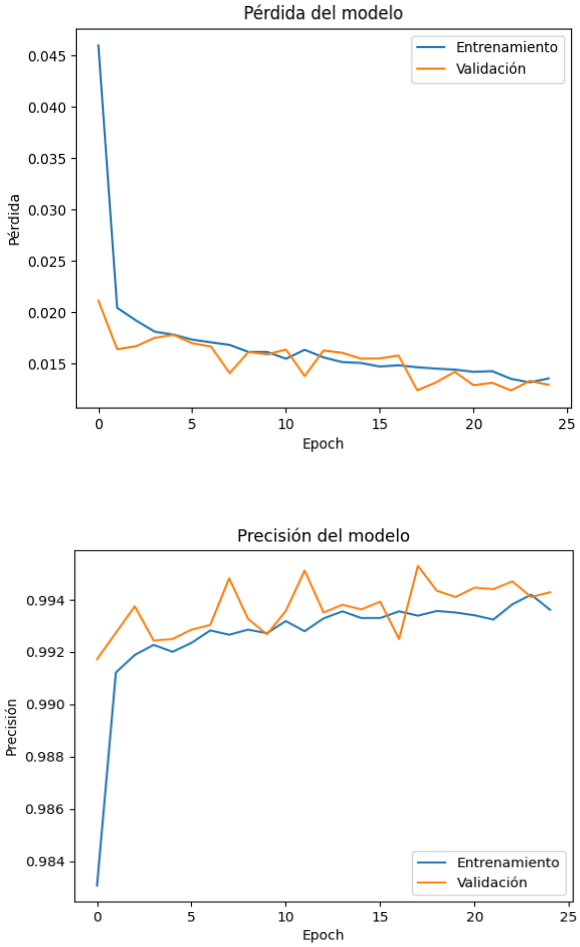
A continuación, se proporciona una descripción y resultados de cada uno de estos modelos:

4.5.1. Red Neuronal

Luego de una fase de entrenamiento de 25 épocas, los resultados de las ejecuciones son las siguientes:

Figura 43

Fase Entrenamiento

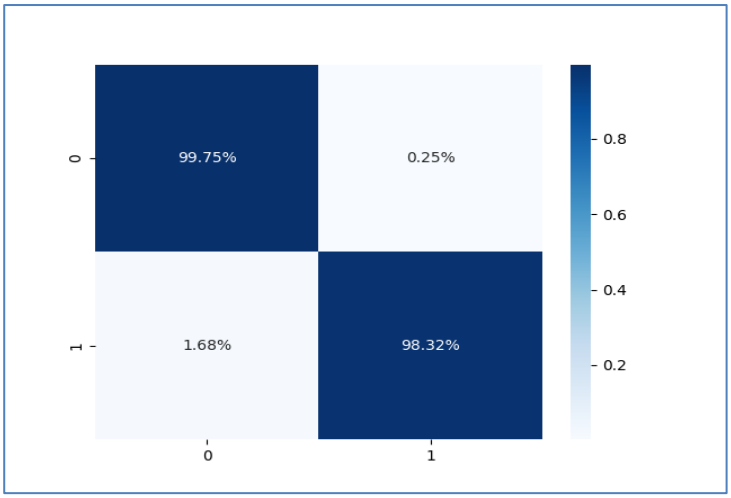


Nota. Autoría propia. Fase de entrenamiento de 25 épocas.

La matriz de confusión muestra altos niveles de precisión al clasificar ambas clases.

Figura 44

Matriz de confusión



Nota. Autoría propia. Se evidencia altos niveles de precisión al clasificar ambas clases.

Figura 45

Código red neuronal

```
# Definir el modelo
model = Sequential()
model.add(Dense(64, input_dim=X_train.shape[1], activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

# Compilar el modelo
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# Entrenar el modelo
history = model.fit(X_train, y_train, epochs=25, batch_size=32, validation_split=0.2)
```

```
# Hacer predicciones con el modelo
y_pred = model.predict(X_test)
y_pred = np.round(y_pred)

# Visualizar la matriz de confusión
from sklearn.metrics import confusion_matrix
sns.heatmap(confusion_matrix(y_test, y_pred, normalize='true'), annot=True, cmap='Blues', fmt='.2%')

print(y_test.value_counts())
```

Nota. Autoría propia.

4.5.2. Árbol de decisión

A continuación, se proporciona información detallada sobre el árbol de decisión:

Parámetros:

random_state: 42. El parámetro random_state se utiliza para inicializar el generador de números aleatorios y garantizar la reproducibilidad de los resultados. Al establecerlo en 42, se garantiza que el modelo se construya de la misma manera cada vez que se ejecute.

Entrenamiento del modelo: El modelo de árbol de decisión se entrena utilizando el conjunto de datos de entrenamiento (x_train y y_train) mediante el método fit. Durante el entrenamiento, el modelo aprende a tomar decisiones basadas en las características proporcionadas y las etiquetas de clase correspondientes.

Figura 46

Código entrenamiento del Árbol de decisión

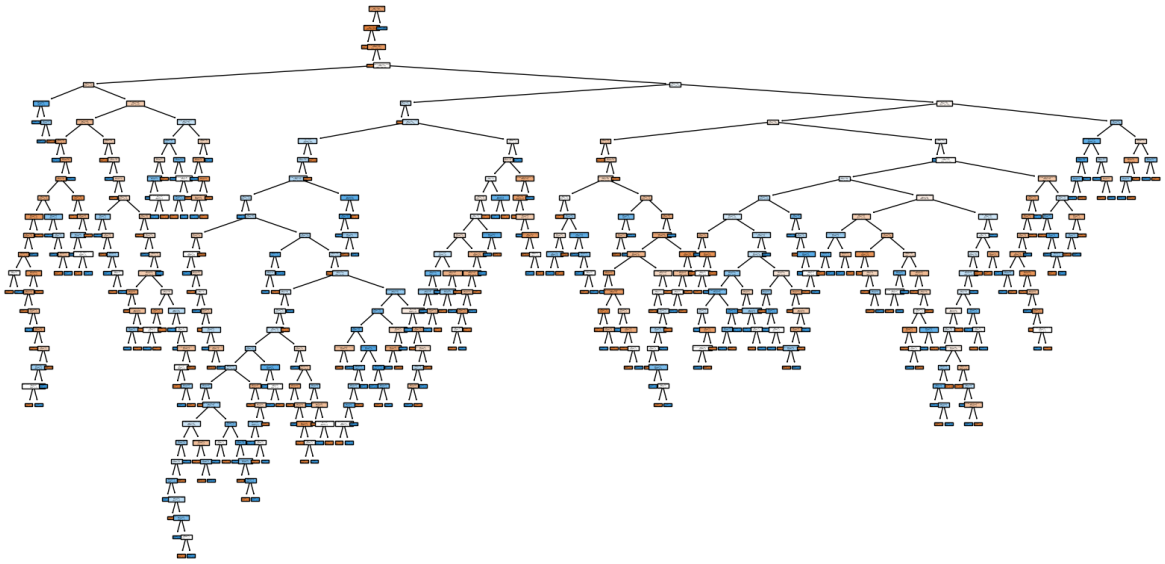
```
# Definir y entrenar el modelo
dt = DecisionTreeClassifier(random_state=42)
dt.fit(X_train, y_train)

# Predecir los valores de la muestra de prueba
y_pred = dt.predict(X_test)
```

Nota. Autoría propia.

El árbol de decisión es una estructura jerárquica compuesta por nodos y ramas, donde cada nodo representa una decisión basada en una característica y cada rama representa una posible salida de esa decisión. El modelo de árbol de decisión divide el conjunto de datos en función de las características y sus valores, con el objetivo de maximizar la homogeneidad de las muestras en cada nodo. Una muestra de cómo queda el árbol final es:

Figura 47

Árbol de decisión



Nota. Autoría propia. División del conjunto de datos en función de las características y sus valores.

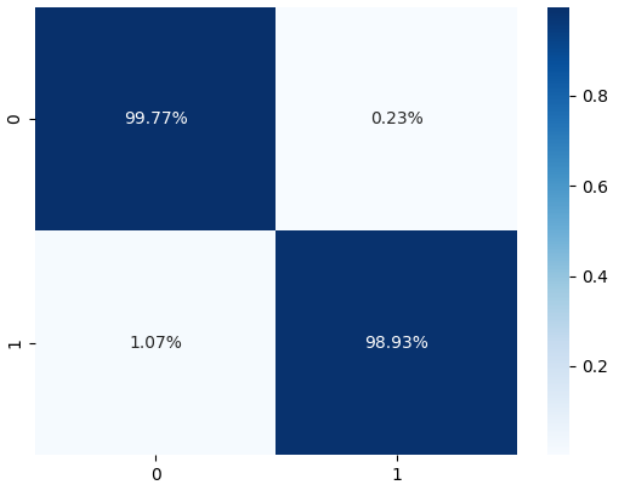
Al entrenar el modelo de árbol de decisión, se utilizan diversos algoritmos y estrategias para construir y optimizar la estructura del árbol, como el algoritmo CART (Classification and Regression Trees). El árbol resultante puede ser visualizado para comprender mejor las decisiones tomadas por el modelo.

Cabe destacar que el árbol de decisión puede sufrir de sobreajuste si no se controla adecuadamente. Por lo tanto, es importante considerar técnicas de poda y ajustar los hiper parámetros para optimizar el rendimiento y evitar la sobre complejidad del modelo.

La matriz de confusión es la siguiente:

Figura 48

Matriz de confusión



Nota. Autoría propia. Se evidencia altos niveles de precisión al clasificar ambas clases.

4.5.3. Regresión logística

La regresión logística es un modelo de clasificación que utiliza la función logística para estimar la probabilidad de pertenencia a una clase o categoría. En el caso de estudio, se aplicó la regresión logística para predecir la probabilidad de abandono de los clientes. Se

ajustaron los coeficientes del modelo utilizando los datos de entrenamiento y se aplicaron técnicas de regularización para evitar el sobreajuste y mejorar la generalización. El código es el siguiente:

Figura 49

Código Regresión logística

```
# Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

# Crear una instancia del modelo de regresión logística
model = LogisticRegression()

# Entrenar el modelo con los datos de entrenamiento
model.fit(X_train, y_train)

# Hacer predicciones en los datos de prueba
y_pred = model.predict(X_test)
```

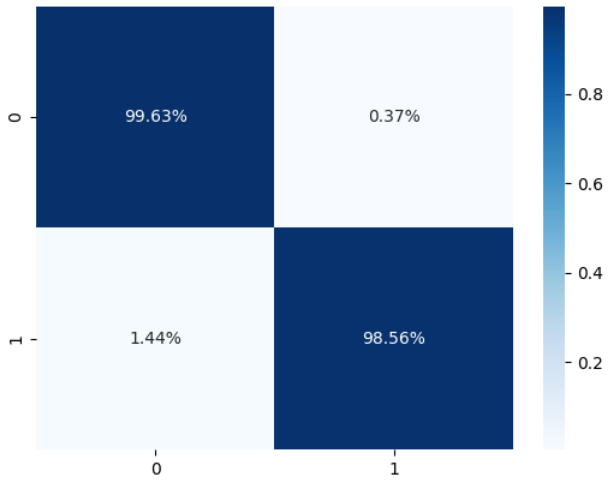
Nota. Autoría propia.

Para determinar si un cliente se considera churn o no, se utiliza un umbral de decisión. Si la probabilidad predicha es mayor que el umbral, se clasifica como churn; de lo contrario, se clasifica como no churn. El umbral puede ajustarse según las necesidades del negocio y los objetivos de clasificación.

La matriz de confusión resultante es la siguiente:

Figura 50

Matriz de confusión



Nota. Autoría propia. Se evidencia altos niveles de precisión al clasificar ambas clases.

El modelo elegido es la Red Neuronal ya que presenta mejor accuracy y desenvolvimiento con los datos.

```
Precisión de la Regresión Logística: 0.97
Precisión del Árbol de Decisión: 0.98
Precisión de la Red Neuronal: 0.99
```

4.6. Fase 6: Despliegue

En la fase de despliegue de la metodología CRISP-DM, se implementará el modelo de red neuronal entrenado para clasificar una nueva base de datos que abarca los meses de marzo, abril y mayo del 2023. El objetivo es identificar los clientes con alta probabilidad de abandono (churn) y proponer alternativas estratégicas para retenerlos.

Figura 51

Despliegue Nuevo Dataset

	numero_identificacion	cantidad_compras_count	subtotal	is_spoonity	genero	generacion	dias_transcurridos	nuevo_rango_salarios
0	706250461	2	25.839500	False	FEMENINO	Generacion_Z	1	Medio alto
1	1712369030	3	33.887262	True	MASCULINO	Generacion_X	1	Medio alto
2	1102135868	3	26.412178	False	FEMENINO	Baby_Boomers	1	Medio alto
3	705018802	5	47.497140	False	MASCULINO	Baby_Boomers	1	Alto
4	707053773	1	0.649800	False	MASCULINO	Millennials	1	Bajo
...	...	...	...	...	...	...	...	...
79050	200957132	1	1.672800	False	FEMENINO	Baby_Boomers	92	Bajo
79051	920666120	1	2.909755	False	MASCULINO	Millennials	92	Bajo
79052	1728228956	1	20.975000	False	MASCULINO	Millennials	92	Medio alto
79053	702514290	1	34.880000	False	FEMENINO	Generacion_X	92	Medio alto
79054	1723215396	1	5.796500	False	FEMENINO	Millennials	92	Bajo

79055 rows × 8 columns

Nota. Autoría propia. El nuevo dataset incluye información de los meses marzo, abril y mayo del 2023.

Para llevar a cabo esta tarea, se aplicará el modelo de red neuronal desarrollado con anterioridad al conjunto de datos de prueba, que incluye información detallada de cada venta realizada durante esos tres meses. El modelo utilizará las características de cada cliente, como su historial de compras, género, edad y afiliación a la farmacia, para realizar la clasificación. Se aplica el mismo procesamiento para que la red neuronal trabaje con los mismos tipos de

datos. La clasificación se realiza usando el siguiente código muy similar al entrenamiento pero que no contiene las etiquetas de target:

Figura 52

Código de clasificación despliegue

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.preprocessing import LabelEncoder, OneHotEncoder

# Supongamos que tenemos un nuevo cliente con los siguientes datos:
data = data.drop(['target_final'], axis=1)
new_data = data

categorical_cols = new_data.select_dtypes(include=['object']).columns
# aplicar One Hot Encoding a las columnas categóricas
onehot_encoder = OneHotEncoder(sparse=False)
encoded_cols = pd.DataFrame(onehot_encoder.fit_transform(new_data[categorical_cols]))
encoded_cols.columns = onehot_encoder.get_feature_names_out(categorical_cols)

# eliminar columnas categóricas originales del dataframe
new_data = new_data.drop(columns=categorical_cols)

# unir el dataframe original con las columnas codificadas
new_df_encoded = pd.concat([new_data, encoded_cols], axis=1)

# Dividir los datos en entrenamiento y prueba
X = new_df_encoded.drop(['numero_identificacion'], axis=1)
```

```
from sklearn.metrics import confusion_matrix
import numpy as np
from sklearn.metrics import accuracy_score

# Escalar las variables
scaler = StandardScaler()
X = scaler.fit_transform(X)

# Hacer predicciones con el modelo
y_pred = model.predict(X)
y_pred = np.round(y_pred)
```

Nota. Autoría propia.

Una vez que el modelo clasifica a los clientes como propensos a abandonar o no, se podrán tomar decisiones estratégicas para retener a aquellos con alta probabilidad de churn. Estas decisiones podrían incluir el diseño de programas de lealtad personalizados, la oferta de descuentos o promociones especiales, o la implementación de campañas de marketing dirigidas. La distribución de los resultados se genera con el siguiente código:

Figura 53

Código de distribución de resultados

```
# Suponiendo que tu arreglo se llama 'my_array'
my_array_flattened = np.reshape(y_pred, (-1,))
# Imprimir la forma del nuevo arreglo
print(my_array_flattened.shape)
my_array_flattened_int = my_array_flattened.astype(int)

# Calcular el conteo de elementos en cada clase
class_counts_pred = np.bincount(my_array_flattened_int)

# Crear las etiquetas para las clases
class_labels = ['Clase 0', 'Clase 1']

# Crear el gráfico de barras
plt.bar(class_labels, class_counts_pred)

# Establecer etiquetas y título del gráfico
plt.xlabel('Clases')
plt.ylabel('Cantidad')
plt.title('Distribución de Clases de la Prediccion')
# Mostrar el gráfico
plt.show()
```

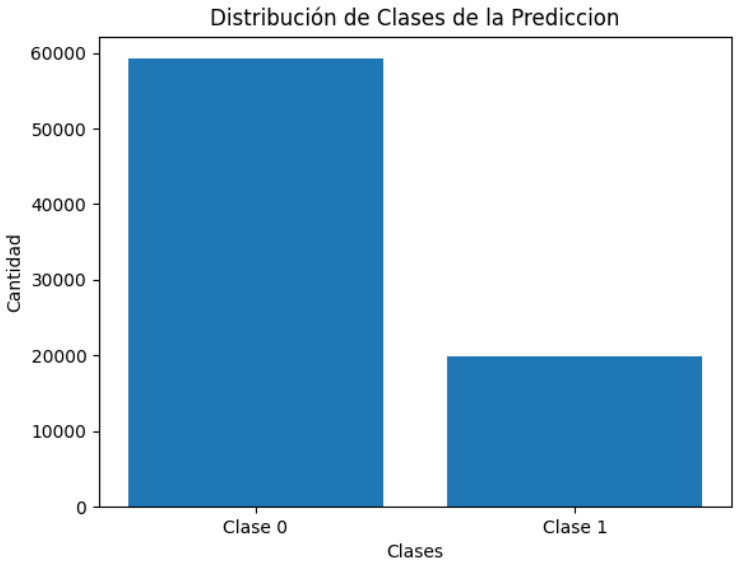
Nota. Autoría propia.

Además, es importante considerar la interpretación de los resultados del modelo y comprender qué características o variables influyen más en la predicción del churn. Esto proporcionará información valiosa para el desarrollo de estrategias más efectivas y focalizadas en la retención de clientes.

Es crucial destacar que el despliegue del modelo requiere una infraestructura adecuada para su implementación en tiempo real o en un entorno de producción. Esto puede incluir la configuración de servidores, la integración del modelo en una aplicación o plataforma existente, y la implementación de medidas de seguridad y privacidad para proteger los datos de los clientes.

Figura 54

Distribución de clases de Predicción



Nota. La clase 1 corresponde a los clientes con pronóstico de abandono.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

En el presente estudio, se aplicó la metodología CRISP-DM para desarrollar un sistema de clasificación de clientes propensos a abandonar (churn) en una empresa farmacéutica. A través de las diferentes fases del proceso, se logró comprender el negocio y establecer los objetivos de minería de datos, lo que permitió definir claramente el problema a abordar. Se realizó un exhaustivo análisis de los datos, identificando y gestionando la calidad de estos, y se aplicaron técnicas de transformación para prepararlos adecuadamente.

Posteriormente, se seleccionaron tres modelos de clasificación: una red neuronal, un árbol de decisión y una regresión logística. Estos modelos fueron entrenados utilizando el conjunto de datos de compras de los clientes recopilado durante un año. Se evaluaron y compararon sus resultados utilizando métricas como la precisión.

Se encontró que la red neuronal, con su arquitectura de dos capas ocultas (64 y 32 neuronas) y función de activación ReLU, fue el modelo que obtuvo el mejor rendimiento en términos de precisión y capacidad de predicción. Esto indica que la red neuronal es una herramienta efectiva para identificar clientes propensos a churn en la farmacia.

En el proceso de despliegue, se utilizó la red neuronal entrenada para clasificar una nueva base de datos que abarcaba los meses de marzo, abril y mayo del 2023. Este despliegue permitió identificar a los clientes que según el modelo tenían una alta probabilidad de abandonar y proponer alternativas estratégicas para retenerlos. Esto podría incluir el diseño de programas de lealtad personalizados, descuentos especiales u otras estrategias de retención específicas para cada cliente.

En conclusión, la aplicación de la metodología CRISP-DM y el desarrollo de un sistema de clasificación de churn han brindado a la empresa farmacéutica herramientas efectivas para identificar y retener a los clientes con mayor riesgo de abandonar. La

implementación de un modelo de red neuronal y la integración de estrategias de retención personalizadas pueden tener un impacto significativo en la retención de clientes y en el logro de los objetivos de negocio. Sin embargo, se recomienda mantener un enfoque de mejora continua y adaptación a medida que evoluciona el comportamiento de los clientes y las necesidades del mercado.

RECOMENDACIONES

Continuar monitoreando y actualizando el modelo: Dado que el comportamiento de los clientes puede cambiar con el tiempo, es importante realizar un seguimiento constante del desempeño del modelo y actualizarlo periódicamente con datos más recientes. Esto garantizará que las predicciones sean precisas y relevantes.

Implementar un sistema de retroalimentación y seguimiento de resultados: Es recomendable establecer un sistema que permita recopilar información sobre las acciones tomadas para retener a los clientes identificados como propensos a churn y evaluar su efectividad. Esto proporcionará retroalimentación valiosa para mejorar las estrategias de retención en el futuro.

Considerar la integración con sistemas de gestión de clientes: Para maximizar el impacto de las estrategias de retención, es recomendable integrar el sistema de clasificación de churn con otros sistemas de gestión de clientes de la empresa. Esto permitirá un enfoque más holístico y una mejor coordinación de las actividades relacionadas con la retención.

Realizar análisis adicionales de segmentación de clientes: Además de identificar a los clientes propensos a churn, se recomienda realizar análisis de segmentación para comprender mejor los diferentes perfiles de clientes y adaptar las estrategias de retención a cada segmento específico. Esto permitirá una personalización más efectiva y una mayor satisfacción del cliente.

REFERENCIAS

Andrés, C., Preciado, C., & Herrera González, D. A. (2020). *Árbol de decisión para el consumidor en la ciudad de Bogotá en el Hard Discount*. https://repository.universidadean.edu.co/handle/10882/10146

Arcidiacono, J. (2021, noviembre 15). *Base de datos: qué es, para qué sirven y cómo elegir la mejor para su empresa*.

Arif, M., & Basit, A. (2017). Comparative study of R and Python programming languages. *International Journal of Computer Science and Information Technology Research*, *5*(1), 75–79.

Beg, M., Taka, J., Kluyver, T., Konovalov, A., Ragan-Kelley, M., Thiéry, N. M., & Fangohr, H. (2021). *Using Jupyter for reproducible scientific workflows*. https://doi.org/10.1109/MCSE.2021.3052101

Bryman Alan. (2016). *Social Research Methods* (Oxford University Press, Ed.; 5th Edition). https://books.google.com.ec/books?hl=es&lr=&id=N2zQCgAAQBAJ&oi=fnd&pg=PP1&dq=Bryman,+A.+(2016).+Social+research+methods.+Oxford:+Oxford+University+Press.&ots=dpRuBWP2ve&sig=FXBBmTGamTpIQLoRLqVC-gSuPV8#v=onepage&q=Bryman%2C%20A.%20(2016).%20Social%20research%20methods.%20Oxford%3A%20Oxford%20University%20Press.&f=false

Cambios en la réplica del lenguaje de definición de datos (DDL) - Documentación de IBM. (s/f). Recuperado el 19 de julio de 2023, de https://www.ibm.com/docs/es/idr/11.3.3?topic=console-replicating-data-definition-language-ddl-changes

De, C., Marín, C., David, A., Vargas, H., José, C., Ángel, I. E., & Sánchez, R. (2022). *Desarrollo de aplicativo web basado en máquinas de vectores de soporte(svm) de aprendizaje supervisado para la predicción en la recomendación de cultivos mediante datos ambientales para fincas agroecológicas del cantón La Maná, provincia de Cotopaxi*. http://repositorio.utc.edu.ec/handle/27000/8457

Diogo, M., Cabral, B., & Bernardino, J. (2019). Consistency Models of NoSQL Databases. *Future Internet 2019, Vol. 11, Page 43*, *11*(2), 43. https://doi.org/10.3390/FI11020043

Fernández Lizana, M. I. (2020). Ventajas de R como herramienta para el Análisis y Visualización de datos en Ciencias Sociales. *Revista Científica de la UCSA*, *7*(2), 97–111. https://doi.org/10.18004/UCSA/2409-8752/2020.007.02.097

Huber, S., Wiemer, H., Schneider, D., & Ihlenfeldt, S. (2019). DMME: Data mining methodology for engineering applications – a holistic extension to the CRISP-DM model. *Procedia CIRP*, *79*, 403–408. https://doi.org/10.1016/J.PROCIR.2019.02.106

IBM. (2021, marzo 4). *Cambios en la réplica del lenguaje de definición de datos (DDL)*. InfoSphere Data Replication.

Joyanes Luis. (2019). Inteligencia De Negocios Y Analítica De Datos por Luis Joyanes - 9789587785418 - Alpha Editorial. En *Alfaomega* (Vol. 1). https://www.alpha-editorial.com/Papel/9789587785418/Inteligencia+De+Negocios+Y+Anal%C3%ADtica+De+Datos

Lemenkova, P. (2019). PROCESSING OCEANOGRAPHIC DATA BY PYTHON LIBRARIES NUMPY, SCIPY AND PANDAS. *Aquatic Research*, *2*(2), 73–91. https://doi.org/10.3153/AR19009

León Jenner. (2020). *Análisis comparativo de sistemas gestores de bases de datos postgresql y mysql en procesos crud* [Universidad Señor de Sipán]. https://repositorio.uss.edu.pe/handle/20.500.12802/7012

Oliphant, T. E., Harris, C. R., Smith, N. J., & Reddy, T. (2020). *NumPy quickstart — NumPy v2.0.dev0 Manual*. https://numpy.org/devdocs/user/quickstart.html

Oracle. (2019). *Qué es una base de datos relacional | Oracle Argentina*. https://www.oracle.com/ar/database/what-is-a-relational-database/

Peng, R. D. (2020). *R programming for data science.* (Vol. 1). Leanpub.

Pilicita Garrido, A., Borja López, Y., & Gutiérrez Constante, G. (2020). Rendimiento de MariaDB y PostgreSQL. *Revista Científica y Tecnológica UPSE*, *7*(2), 9–16. https://repositorio.upse.edu.ec/handle/46000/7315

Queiroz-Sousa, P., & Salgado, A. (2019). A Review on OLAP Technologies Applied to Information Networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *14*(1). https://doi.org/10.1145/3370912

Raschka, S., Patterson, J., & Nolet, C. (2020). Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. En *Information (Switzerland)* (Vol. 11, Número 4). MDPI AG. https://doi.org/10.3390/info11040193

Robles-Velasco, A., Cortés, P., Muñuzuri, J., & Barbadilla-Martín, E. (2020). Aplicación de la regresión logística para la predicción de roturas de tuberías en redes de abastecimiento de agua. *Dirección y Organización*, *0*(70), 78–85. https://doi.org/10.37610/DYO.V0I70.570

Rojas, E. M. (2020). *Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo*.

Sengupta, S. (2021). *R vs Python: What's the Difference and Which is Better?* IBM Cloud Education.

Shirley, K., & Rueda, C. (2022). APLICACIÓN DE REDES NEURONALES ARTIFICIALES PARA EL PRONÓSTICO DE PRECIOS DE CAFÉ APPLICATION OF ARTIFICIAL NEURONAL NETWORKS FOR COFFEE PRICE FORECAST. *Revista Colombiana de Tecnologías de Avanzada*, *1*.

Tran, M. K., Panchal, S., Chauhan, V., Brahmbhatt, N., Mevawalla, A., Fraser, R., & Fowler, M. (2022). Python-based scikit-learn machine learning models for thermal and electrical performance prediction of high-capacity lithium-ion battery. *International Journal of Energy Research*, *46*(2), 786–794. https://doi.org/10.1002/ER.7202

Vafeiadis, T., Diamantopoulos, A., & Chatzithomas, L. (2015). A descriptive model of customer churn prediction in telecommunications: Advancing knowledge discovery in databases. *Journal of Business Research*, *68*(2), 293–304.

VanderPlas, J. (2016). *Python data science handbook: Essential tools for working with data.* (Vol. 1). O'Reilly Media, Inc.

Waskom, M. (2017). Seaborn: A Python Library for Attractive and Informative Statistical Graphics. *Journal of Open Source Software*.

Wongkar, M., & Angdresey, A. (2019). Sentiment Analysis Using Naive Bayes Algorithm Of The Data Crawler: Twitter. *Proceedings of 2019 4th International Conference on Informatics and Computing, ICIC 2019*. https://doi.org/10.1109/ICIC47613.2019.8985884

Yim, A., Chung, C., & Yu, A. (2019, abril). *Matplotlib for Python Developers: Effective techniques for data visualization with Python*. Second edition. https://books.google.es/books?hl=es&lr=&id=G99YDwAAQBAJ&oi=fnd&pg=PP1&dq=Matplotlib:+Data+Visualization+Made+Easy&ots=tyd8xD4M2b&sig=kXHMQ2sxH5vsnn1R3pS4wdyeZhE#v=onepage&q=Matplotlib%3A%20Data%20Visualization%20Made%20Easy&f=false