Automatización de Información en empresas de bienes y servicios

John Steven Montes Urrea

Especialización en Big Data e Inteligencia de Negocios
Facultad de Ingenierías y Arquitectura
Universidad Católica Luis Amigó
Medellín, Colombia
2024

Dedicatoria

Dedico este posgrado a mis padres, cuya guía y apoyo incondicional han sido esenciales en cada paso de mi vida. Gracias por enseñarme el valor de la perseverancia, la honestidad y el esfuerzo, principios que me han permitido alcanzar este nuevo logro. Su confianza en mí ha sido mi mayor motivación, y hoy, con gratitud y orgullo, comparto con ustedes este logro que también es suyo.

TABLA DE CONTENIDO

LISTA DE TABLAS

LISTA DE FIGURAS

1. Introducción

En la actualidad, la automatización de la información en empresas de bienes y servicios ha transformado significativamente la forma en que operan las organizaciones. La incorporación de herramientas modernas para la gestión interna ha permitido optimizar el manejo de datos, mejorando la eficiencia y aumentando la competitividad en mercados dinámicos. Estas soluciones tecnológicas brindan a las empresas la capacidad de adaptarse rápidamente a las demandas del entorno y ofrecer productos y servicios con mayor agilidad, satisfaciendo así las necesidades de los clientes en un contexto de constante cambio.

En particular, procesos manuales complejos, como la distribución de cuotas, resultan ser lentos y propensos a errores cuando involucran grandes volúmenes de datos y múltiples variables. La ineficiencia en estos procedimientos puede generar demoras, asignaciones incorrectas y afectar la precisión de los resultados, lo que disminuye la productividad y puede generar insatisfacción de los empleados al no poner una cuota justa para el punto de venta.

Para abordar estos desafíos, la implementación de modelos de machine learning surge como una solución eficaz. Un modelo entrenado con datos históricos y patrones relevantes permite automatizar la distribución de cuotas de manera rápida y precisa, minimizando los tiempos de ejecución y reduciendo significativamente el margen de error. Además, estos sistemas pueden adaptarse a cambios en las reglas de negocio y condiciones externas, mejorando continuamente sus resultados a medida que se actualizan los datos. AutoML (Aprendizaje Automático Automatizado) es un campo emergente que tiene como objetivo automatizar el proceso de construir modelos de aprendizaje automático. (Schmitt, 2023)

La principal ventaja de un enfoque basado en machine learning es su capacidad para procesar grandes volúmenes de información en cuestión de segundos, logrando una distribución más justa y equitativa. Al eliminar los riesgos asociados a los errores humanos,

se garantiza mayor precisión, consistencia en los resultados y una mejora general en la eficiencia operativa, lo que contribuye a la satisfacción del cliente y al éxito sostenible del proceso.

2. Motivación

La motivación de este trabajo surge de la creciente necesidad de automatizar la gestión de la información en las organizaciones, especialmente en un contexto donde la eficiencia y la precisión son esenciales para mantener la competitividad. Los procesos manuales de administración de datos no solo demandan tiempo, sino que también introducen riesgos de errores que afectan la toma de decisiones y el desempeño general de las empresas. En este escenario, la automatización se presenta como una solución estratégica para transformar la forma en que las organizaciones manejan su información, permitiéndoles procesar grandes volúmenes de datos de manera rápida y precisa.

Explorar la implementación de modelos de machine learning orientados a la automatización ofrece una oportunidad para maximizar la eficiencia operativa, reducir el margen de error y liberar recursos humanos para tareas de mayor valor. Esta automatización inteligente no solo facilita la adaptación a cambios en el entorno de negocio, sino que también mejora la transparencia y consistencia en los resultados, sentando las bases para un crecimiento sostenible. La motivación central de este proyecto es demostrar cómo la integración de tecnología avanzada puede revolucionar la gestión de información y convertirse en un pilar fundamental para la innovación en un entorno empresarial en constante evolución.

Además, en un contexto empresarial donde los datos crecen exponencialmente, la implementación de herramientas avanzadas para su análisis no solo es una ventaja, sino una necesidad estratégica. Los modelos de machine learning permiten detectar patrones, anticipar

tendencias y tomar decisiones fundamentadas en evidencia, lo que se traduce en una ventaja competitiva significativa.

3. Planteamiento del problema

En un entorno empresarial cada vez más dinámico y competitivo, la gestión eficiente de la información se ha convertido en un factor clave para el éxito organizacional. Sin embargo, muchas empresas aún dependen de procesos manuales para la recolección, procesamiento y análisis de la información, lo que genera limitaciones. La ejecución manual de tareas, como la asignación de recursos, incrementa significativamente el riesgo de errores, demoras e ineficiencias operativas. Esto no solo afecta la precisión en la toma de decisiones, sino que también limita la capacidad de las organizaciones para adaptarse rápidamente a las fluctuaciones del mercado y a las nuevas necesidades de las áreas.

La falta de automatización en la gestión de información conlleva una serie de desafíos críticos. Entre ellos, la carga operativa excesiva, la vulnerabilidad ante errores humanos, y la ineficiencia en el manejo de grandes volúmenes de datos, que impide la optimización de los procesos. Además, esta situación puede derivar en inconsistencias en la distribución de recursos, asignaciones incorrectas y la pérdida de oportunidades competitivas.

Ante estos desafíos, la necesidad de automatizar la gestión de información mediante tecnologías como el machine learning se vuelve cada vez más importante. Implementar modelos que puedan procesar, analizar y distribuir datos de forma automática ofrece una solución eficiente para superar las limitaciones de las tareas manuales.

4. Justificación

La gestión manual de grandes volúmenes de datos resulta insostenible debido al tiempo que consume y al alto riesgo de errores, lo que limita la capacidad de las

organizaciones para adaptarse rápidamente a las demandas del mercado. La implementación de tecnologías avanzadas, como el machine learning, ofrece una solución eficaz para mejorar la eficiencia operativa, aumentar la precisión y minimizar el margen de error en procesos críticos dentro de las organizaciones. Además, al liberar a los empleados de tareas rutinarias, las empresas pueden redirigir su talento hacia actividades estratégicas de mayor valor agregado. Este enfoque no solo optimiza el uso de recursos, sino que también mejora la toma de decisiones basada en datos, aumentando la competitividad y la satisfacción del cliente. Las herramientas de analítica de datos podrían definirse como soluciones informáticas que permiten procesar grandes volúmenes de datos desde diferentes capas, para convertirlos en información útil en la toma de decisiones (Coronado Medina, 2019)

En consecuencia, invertir en la automatización de la información se convierte en una necesidad imperante para que las organizaciones no solo sobrevivan, sino que prosperen en un entorno de constante cambio y exigencia.

5. Marco de Referencias

La IA puede proporcionar experiencias de aprendizaje personalizadas, análisis de datos en tiempo real e información basada en datos sobre las tendencias de la industria y los desafíos de sostenibilidad. Esto puede conducir a programas educativos más relevantes y eficaces que doten a los estudiantes de las habilidades y conocimientos necesarios para abordar los complejos problemas de la sostenibilidad en el contexto de la Industria 4.0. (Abulibdeh et al., 2024)

Los algoritmos RL se utilizan como agentes en simulaciones para aprender procesos logísticos. Estos algoritmos les permiten optimizar los procesos logísticos y los sistemas de automatización de una empresa. Al implementar este enfoque, la complejidad de las

operaciones en el mundo real se puede gestionar de manera efectiva para mejorar la eficiencia de los procesos logísticos (Lim & Jeong, 2023)

Los datos de series de tiempo se utilizan comúnmente en tareas de clasificación y aprendizaje automático, y existe una amplia gama de técnicas disponibles para procesar este tipo de datos antes del aprendizaje. Estas técnicas se eligen en función de las características de los datos de la serie temporal y de cómo estas características se pueden extraer o proyectar mejor a partir de los datos sin procesar (Latham & Giannetti, 2023)

En las empresas vemos una variedad de soluciones de TI que afectan el enfoque del análisis aplicado de datos utilizado por el consejo de administración de la empresa. Hoy en día, los responsables de la toma de decisiones no sólo quieren mirar informes estáticos, también están interesados en herramientas fáciles de usar para evaluar objetivos e indicadores clave de desempeño (KPI) para identificar cualquier posibilidad de avance y amenazas de fracaso (Dudycz et al., 2022)

Big data se ha vuelto cada vez más importante en los últimos años porque permite a las organizaciones obtener información a partir de estos datos. Al analizar grandes volúmenes de datos, las organizaciones pueden descubrir patrones, correlaciones y tendencias ocultos que pueden ayudarlas a tomar mejores decisiones y mejorar sus operaciones (Donta et al., 2023)

Gracias a una plataforma de big data se recolectaron datos de gran volumen y alta frecuencia de muchos equipos del área de producción. Con la plataforma, se ha aumentado la trazabilidad del proceso en el campo y los datos recopilados han servido de base para muchos estudios analíticos. Al hacer que los datos sean manejables y rastreables, se han logrado ganancias de productividad en muchas áreas diferentes, como la detección de errores, el análisis de la causa raíz y el diseño de procesos (Cakir et al., 2022)

Las plataformas de computación en la nube también brindan flexibilidad y capacidad de almacenamiento de datos grandes y asequibles para las demandas de los clientes, además de ofrecer potencia informática altamente escalable. Por lo tanto, son capaces de adaptarse a cargas de trabajo de configuración industrial. (Kabugo et al., 2020)

Existe una gran cantidad de plataformas y software de análisis y visualización de datos disponibles tanto en las instalaciones como en la nube. Cada uno de ellos tiene sus ventajas y desventajas. Es incorrecto afirmar que una herramienta es superior a otra sin considerar las necesidades específicas de un determinado caso de uso. Cada vez más, las soluciones de análisis y visualización de datos nativas de la nube aprovechan la informática sin servidor y pueden escalar automáticamente sin ninguna infraestructura que administrar. (Kahveci et al., 2022)

En todo este big data hay oportunidades sin precedentes para el descubrimiento de patrones que pueden contener pistas importantes para resolver problemas difíciles y al mismo tiempo ofrecer una comprensión complementaria del significado físico de los parámetros a otras características físicas de un sistema o proceso. Junto con la capacidad de comprender datos de alta dimensión, la IA proporciona la capacidad de transformar grandes cantidades de datos de fabricación complejos, que se han vuelto comunes en las fábricas actuales, en información útil y reveladora (Arinez et al., 2020)

En el contexto anterior y con el ánimo de otorgar solución al sistema de información optimizado se formulan los siguientes objetivos.

6. Objetivos

6.1. Objetivo General

Diseñar un modelo para la optimización de la gestión de la información en la empresa Medivelius aplicando técnicas de analítica de datos.

6.2. Objetivos Específicos

1. Configurar un entorno óptimo para la gestión de los datos.
2. Aplicar técnicas para el procesamiento de datos de manera eficiente.
3. Crear una herramienta que permita la optimización y la visualización de la información almacenada bajo el mecanismo de la predicción.

7. Viabilidad

Para abordar los desafíos en la gestión de la información, el proyecto requiere el uso de herramientas de analítica de datos para consolidar y estructurar los datos de manera segura y eficiente. Además, es esencial contar con un analista de datos que diseñe procedimientos para la optimización del manejo de la información, acompañado de un computador Dell Core i7-1255U con 16 GB de Ram DDR4, con SSD de 1TB y gráficos Intel Iris Xe.

El proyecto propuesto requiere abordar varias consideraciones normativas y legales, entre ellas el cumplimiento de la regulación de protección de datos, la seguridad de la información, el uso de herramientas de analítica y visualización que cumplan con las normativas y políticas de la empresa. También es importante asegurar que todo software utilizado debe tener las licencias necesarias y que estén claramente definidas para proteger a la empresa y evitar futuras violaciones de seguridad. Como consecuencia, el uso de herramientas de analítica de datos en la empresa permitirá consolidar y estructurar la información de manera eficiente, reduciendo el tiempo y los recursos de la compañía.

Finalmente, este proyecto está definido para tener un alcance en el uso de la herramienta para la construcción de procedimientos para la optimización de la información en la empresa, finalizando con la visualización de los datos de manera clara y accesible.

El proyecto traerá como consecuencia una mayor eficiencia operativa al consolidar y estructurar la información, lo que reducirá el tiempo y los recursos de la empresa. Además, con la intervención de un analista de datos y el uso de un equipo informático adecuado, se optimizarán los procesos de manejo de información. El cumplimiento de las normativas de protección de datos y la seguridad de la información será clave para evitar riesgos legales y garantizar el uso de herramientas con las licencias necesarias. Finalmente, el proyecto permitirá una visualización clara y accesible de los datos, mejorando la toma de decisiones estratégicas y protegiendo a la empresa de posibles violaciones de seguridad.

8. Metodología

En este capítulo, se presenta de manera detallada el procedimiento para alcanzar cada uno de los objetivos específicos relacionados con la optimización de la gestión de la información en empresas de bienes y servicios. Se describen las diversas actividades, métodos y técnicas necesarias para implementar los pasos de la metodología CRISP-DM, alineándose con la problemática de los procesos manuales que limitan la eficiencia y precisión en la distribución de cuotas. Este enfoque metodológico busca abordar directamente los desafíos identificados, facilitando la adopción de soluciones automatizadas que mejoren la capacidad de respuesta de las organizaciones ante un entorno empresarial en constante cambio.

Objetivo específico	Actividad	Entregable	Fase CRISP-DM

1. Configurar un entorno óptimo para el almacenamiento de los datos.	Diseñar el entorno necesario para almacenar de manera segura y eficiente los datos recolectados.	Esquema de la arquitectura del almacenamiento de datos y manual de implementación.	Comprensión del negocio
2. Diseñar protocolos para el procesamiento de datos de manera eficiente.	Revisar y transformar los datos recopilados de las diferentes fuentes para garantizar su calidad. Desarrollar un protocolo para la limpieza y preprocesamiento de datos, asegurando que sean aptos para el modelo de machine learning.	Documento con la caracterización de la base de datos y el proceso de limpieza, incluyendo análisis exploratorio y tratamiento de datos faltantes.	Preparación de los Datos
3. Crear una herramienta que permita la optimización y la visualización de la información almacenada bajo el mecanismo de la predicción.	Desarrollar un modelo de machine learning para la asignación de cuotas basado en los datos preprocesados. Evaluar el modelo de machine learning utilizando datos reales y proponer estrategias de visualización para las asignaciones de cuotas.	Modelo de machine learning en Google Colab, con un informe de evaluación que describe cómo se distribuyen las cuotas y su justificación. Informe de evaluación del modelo con análisis detallado de los resultados y propuestas de estrategias para optimizar la asignación de cuotas.	Modelado

Tabla 1: Metodología CRISP-DM asociado a los objetivos.

La metodología CRISP-DM comienza con la comprensión del negocio para definir objetivos, seguido de la comprensión de los datos. En la preparación de los datos, se limpian y organizan conjuntos de datos para el análisis. En la fase de modelado, se aplican algoritmos de machine learning y se evalúan los modelos generados de acuerdo a la necesidad de la empresa. Finalmente, la fase de despliegue asegura la implementación y monitoreo en un entorno real.

8.1. Comprensión del negocio

La primera fase de este proyecto de optimización de la información consiste en entender a fondo el problema a resolver. Aquí se definen los requisitos y objetivos siempre desde la necesidad de la empresa, donde posteriormente se traducirán en conceptos técnicos y un plan de trabajo. Se evalúa la situación actual de la compañía, y se elabora un plan que detalla los pasos y procedimientos a seguir.

8.2. Comprensión de los datos

En esta fase, se recolectan y exploran los datos para establecer un primer contacto con el problema. Se recogen datos iniciales y se adaptan a las necesidades del proyecto de optimización de la asignación de cuotas. Luego se describen formalmente los datos, se aplican técnicas de estadística descriptiva y se verifica su consistencia para detectar posibles errores o valores atípicos.

8.3. Preparación de los datos

La preparación de datos implica seleccionar, limpiar y organizar conjuntos de datos suministrados para poder realizar el modelo, asegurando que estén en las mejores condiciones posibles para ser utilizados en los análisis. Esto incluye la corrección de datos erróneos, la eliminación de valores duplicados o nulos, y la transformación de las variables para hacerlas compatibles con los algoritmos de machine learning.

8.4. Modelado

Durante la fase de modelado, se crean modelos de conocimiento a partir de los datos procesados. Se seleccionan algoritmos apropiados, se elabora un plan de pruebas para los parámetros de machine learning y se definen métricas de evaluación. Los modelos se construyen y se evalúan para verificar que cumplen con los criterios y realmente va a funcionar para solucionar el problema inicial.

8.5. Evaluación

En esta fase, se revisan los modelos creados para comprobar si cumplen con los objetivos del negocio. Se analizan los resultados obtenidos en las etapas anteriores, evaluando lo que ha funcionado bien y lo que no, con el fin de detectar posibles errores que podrían afectar el resultado final.

8.6. Despliegue

En la fase de despliegue, se pone en funcionamiento el modelo, se supervisa su rendimiento y se hacen los ajustes necesarios. Se establecen estrategias para detectar y corregir posibles problemas, garantizando el funcionamiento a futuro. Finalmente, se repasa todo el proceso del proyecto, resaltando lo que se hizo bien y lo que se puede mejorar, con el objetivo de aprender de la experiencia.

9. Resultados

A continuación se presenta el análisis de los resultados obtenidos en función de los tres objetivos específicos planteados en este proyecto. Estos resultados reflejan el progreso alcanzado en cada fase, comenzando con la configuración de un entorno óptimo para el almacenamiento de datos, seguido del diseño de protocolos para el procesamiento eficiente de la información, y culminando con la creación de una herramienta que optimiza y visualiza los datos mediante técnicas de predicción. Este análisis permite evaluar la efectividad de las estrategias implementadas y cómo han contribuido a mejorar la gestión de la información en la empresa.

9.1. Objetivo específico 1: Configurar un entorno óptimo para el almacenamiento de los datos.

9.1.1. Comprensión del negocio

Medivelius es una empresa especializada en la comercialización de productos dermatológicos para el cuidado de la piel. Fundada en el año 2005, esta compañía se dedica a importar productos de alta calidad, que luego distribuye en el mercado colombiano, destacándose por su compromiso con la salud y el bienestar dermatológico.

Entre los laboratorios más reconocidos y prestigiosos, Medivelius comercializa Uriage, Cantabria Labs, Apivita, Genove y Sensilis, todas enfocadas en ofrecer soluciones innovadoras y científicamente respaldadas para el cuidado de la piel. Estas marcas a su vez están categorizadas por unidades de negocio Apur, Cange y Norsen.

Un equipo esencial para el éxito de Medivelius son sus aproximadamente 140 dermoconsultoras, profesionales capacitadas que asesoran a los clientes en los puntos de venta. Estas dermoconsultoras trabajan en reconocidas tiendas de cuidado de la piel en Colombia, tales como Medipiel, Bella Piel, Line Estética, Farmatodo y Dermatológica, brindando recomendaciones personalizadas y contribuyendo a mejorar la experiencia de compra de los consumidores.

Gracias a su enfoque en la calidad y en la atención al cliente, Medivelius se ha posicionado como un referente en el sector dermatológico en Colombia, ganando la confianza de sus clientes y consolidándose como un líder en el cuidado de la piel.

Uno de los procesos más demandantes y desgastantes en Medivelius es la asignación de cuotas a cada uno de los puntos de venta, una tarea que implica una considerable inversión de tiempo y esfuerzo. La empresa actualmente tiene cobertura en más de 600 puntos de venta

distribuidos entre los clientes de la empresa, de los cuales necesita recibir una cuota de ventas específica. Este proceso de asignación es realizado de forma manual, una metodología que, si bien ha sido funcional, genera una carga operativa significativa.

Debido a que las cuotas se deben de asignar a cada unidad de negocio y por tienda, significa que, para completar el proceso de asignación de cuotas, se deben gestionar aproximadamente 1,800 datos de forma manual, considerando las variaciones y especificaciones para cada punto de venta y unidad de negocio. Este volumen de datos requiere una dedicación exhaustiva del equipo responsable, que debe revisar y ajustar individualmente cada cuota, asegurándose de que los objetivos reflejan las particularidades de cada punto de venta.

9.1.2. Comprensión de los datos

La empresa ha estado realizando la asignación de cuotas de manera manual durante aproximadamente un año, lo que ha permitido acumular un valioso registro histórico de datos. Este historial representa una base sólida para desarrollar un modelo de aprendizaje automático que pueda analizar y simular el comportamiento de asignación de cuotas. Al entrenar un modelo con estos datos históricos, se puede automatizar el proceso, permitiendo que el sistema prediga y asigne cuotas de manera eficiente y precisa, imitando las decisiones que anteriormente se tomaban de forma manual.

A continuación se explica cada una de las variables de el dataset anteriormente explicado:

Mes (str):

Variable de tipo texto que indica el mes en el que se registró la cuota. En este ejercicio, contamos con un historial de 6 meses: abril, mayo, junio, julio, agosto y septiembre.

Cliente (str):

Esta variable de tipo texto se refiere al cliente al que se venden los productos de Medivelius. Entre nuestros clientes más destacados, y aquellos que cuentan con el mayor apoyo de nuestro personal dermo consultor, se encuentran Medipiel, Línea Estética, Bella Piel y Farmatodo.

Eean PDV (str):

Esta variable de tipo texto representa el código interno único de cada punto de venta. Este código es crucial para los análisis, ya que permite diferenciar puntos de venta que pueden tener el mismo nombre pero pertenecen a clientes distintos. Así, el código asegura una identificación única en nuestras bases de datos.

Punto de venta (str):

Variable de tipo texto que corresponde a los nombres de las tiendas para cada uno de los clientes, en este momento se tienen un total de 598 tiendas a las cuales se les debe de realizar cuota, sin embargo no todas van a tener una dermoconsultora asignada, puesto que existen variables como los cupos, la rotación y la asignación dependiendo de las decisiones estratégicas de la compañía.

Distrito (str):

El distrito es una variable de tipo texto que permite identificar a nivel nacional donde se encuentra el punto de venta asignado, los distritos que actualmente se manejan en Medivelius son los siguientes: Distrito 1: Bogotá Distrito 2: Antioquia Distrito 3: Pacífico Distrito 4: Costa Distrito 5: Bucaramanga y Cúcuta

UN (str):

La unidad de negocio es una variable de tipo texto que indica la categoría en la cual se encuentra el producto, esta categoría depende del laboratorio y existen actualmente tres de ellas, a continuación se detalla cada una con sus respectivas marcas. Apur: Uriage y Apivita Cange: Heliocare, Endocare, Biretix y Genové Sensilis: Sensilis

Crec obj (float):

Variable de tipo número decimal, hace referencia al crecimiento objetivo que nos pidieron realizar desde gerencia y que posteriormente será repartido según la venta en todos las tiendas, el crecimiento objetivo es un número porcentual que va desde un 0% hasta un 100%, aunque normalmente varía de 15 a 40% dependiendo de las circunstancias. Este crecimiento nos lo asignan mes a mes y varía dependiendo de cada unidad de negocio.

Promedio (int):

Corresponde a una variable de tipo número entero y se calcula de acuerdo al promedio de los últimos 3 meses de la venta por cada tienda. Esta variable es importante porque permite dar visual de qué tan bien viene el punto de venta y que tanto se le puede colocar en la cuota para ser lo más justo posible.

Peso (str):

El peso es una variable de tipo número decimal y es un cálculo que nace del promedio de ventas de cada punto de venta por cada unidad de negocio dividido la suma total de ese mismo dato por cada unidad de negocio.

Mtd-1 (int):

Las ventas del año pasado en el mismo mes (MTD-1) es una variable de tipo número entero, se utiliza para saber cuánto vendió la tienda en el año anterior y poder aplicarle un crecimiento acorde a esos datos, es una de las variables más importantes dado que es la base para poder asignar una cuota que apunte a los crecimientos de la compañía.

Tgt fc (int):

El target factorial (TGT FC) es la multiplicación de cada punto de venta entre las columnas del MTD-1 por CREC OBT. Este cálculo se realiza para tener una visual de cómo sería la cuota si tenemos en cuenta una relación directa entre la venta del año anterior y el crecimiento objetivo propuesto por gerencia. Es muy útil para tener una columna de comparación cuando se evalúan las cuotas.

Tgt peso (int):

El target asignado por el peso (TGT FC) es una fórmula matemática producto de la multiplicación del peso anteriormente calculado por el promedio de ventas de los últimos tres meses, se realiza para tener una sugerencia dada por la fórmula matemática de cuál debería ser la cuota si se tiene en cuenta sólo el valor de venta del año en curso de acuerdo a la participación de la tienda sobre el total.

Cuota (int):

La variable objetivo, conocida como cuota, se calcula manualmente considerando el promedio de ventas de los últimos tres meses, junto con el target factorial y el target de crecimiento. Con base en estos parámetros, se estima un valor que varía según las ventas de cada variable. Si el valor resultante es muy alto, se ajusta la cuota para evitar un crecimiento excesivo y mantener un nivel de ventas manejable. Por otro lado, si el valor es bajo, se busca aumentar la cuota, ya que el crecimiento sería más alcanzable para la dermo consejera.

El objetivo del modelo de aprendizaje automático es automatizar este proceso, eliminando la necesidad de cálculos manuales y permitiendo una asignación de cuotas más precisa y eficiente. Con el modelo, se espera optimizar la predicción de cuotas, adaptándolas dinámicamente a las tendencias de ventas y al comportamiento del mercado, lo que permitirá establecer objetivos más realistas y alcanzables para las dermo consejeras.

Este archivo contiene un total de 8945 filas y 13 columnas.

9.2. Objetivo específico 2: Diseñar protocolos para el procesamiento de datos de manera eficiente.

9.2.1. Preparación de los datos

Iniciamos el proceso de preprocesamiento de los datos revisando la estructura general del dataset, evaluando el número de filas y tipos de datos presentes, además de analizar la existencia de duplicados y valores nulos.

Dado que este análisis parte de una base previamente empleada para la asignación manual de cuotas, verificamos la integridad de los datos. Confirmamos que no existen filas duplicadas ni valores nulos, lo cual facilita el análisis al no requerir la eliminación o imputación de valores. Sin embargo, se detectaron 6 variables categóricas que se deben modificar para continuar con el ejercicio:

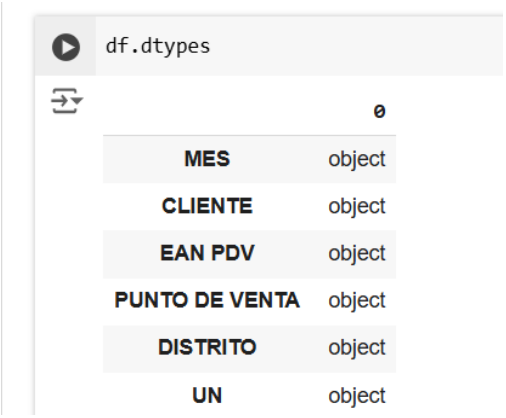

Figura 1: Tipos de datos

Para determinar el preprocesamiento adecuado, analizaremos la cantidad de categorías únicas en cada una de estas variables. Esta exploración nos permitirá comprender la

diversidad de valores y definir si es viable aplicar técnicas como codificación one-hot o de etiquetas, o si es preferible agrupar ciertas categorías para reducir la dimensionalidad.

```
[ ]  num_categorias_Mes = df['MES'].nunique()
     num_categorias_Cliente = df['CLIENTE'].nunique()
     num_categorias_Ean_PDV = df['EAN PDV'].nunique()
     num_categorias_Punto_de_Venta = df['PUNTO DE VENTA'].nunique()
     num_categorias_Distrito = df['DISTRITO'].nunique()
     num_categorias_UN = df['UN'].nunique()

     print(num_categorias_Mes)
     print(num_categorias_Cliente)
     print(num_categorias_Ean_PDV)
     print(num_categorias_Punto_de_Venta)
     print(num_categorias_Distrito)
     print(num_categorias_UN)
```

```
6
15
744
714
8
3
```

Figura 2: Valores únicos por fila

Como tenemos muchas categorías de diferentes variables, One-Hot Encoding puede generar un gran número de columnas, lo que puede ser un problema en términos de espacio y tiempo de procesamiento

Adicionalmente, aplicar el Label Encoding tampoco es una opción, puesto que esto podría generar un problema de sobreajuste y sabemos de antemano que estos clientes no tienen relación alguna con la variable objetivo, por lo cual procedemos a eliminar todas estas columnas.

Realizamos un mapa de calor para saber la correlación de todas las columnas con nuestra variable de salida.

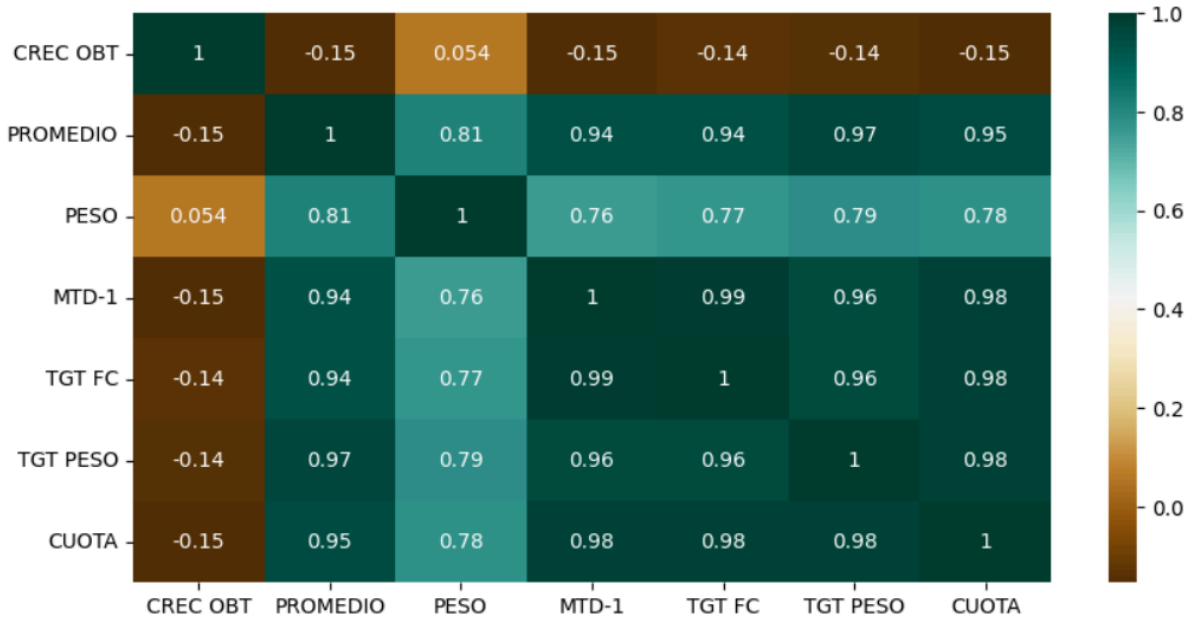

Figura 3: Mapa de calor

Como podemos observar, algunas variables presentan una alta correlación entre sí. Por ello, realizaremos un gráfico de dispersión para analizar las relaciones entre TGT PESO, TGT FC y MTD-1, ya que estas son las variables con mayor correlación respecto a la variable objetivo.

Cuota vs TGT Peso:

```
[ ]  fig, ax = plt.subplots(figsize=(10,6))
     ax.scatter(df['TGT PESO'], df['CUOTA'])
     ax.set_xlabel('TGT PESO')
     ax.set_ylabel('CUOTA')
     plt.show()
```

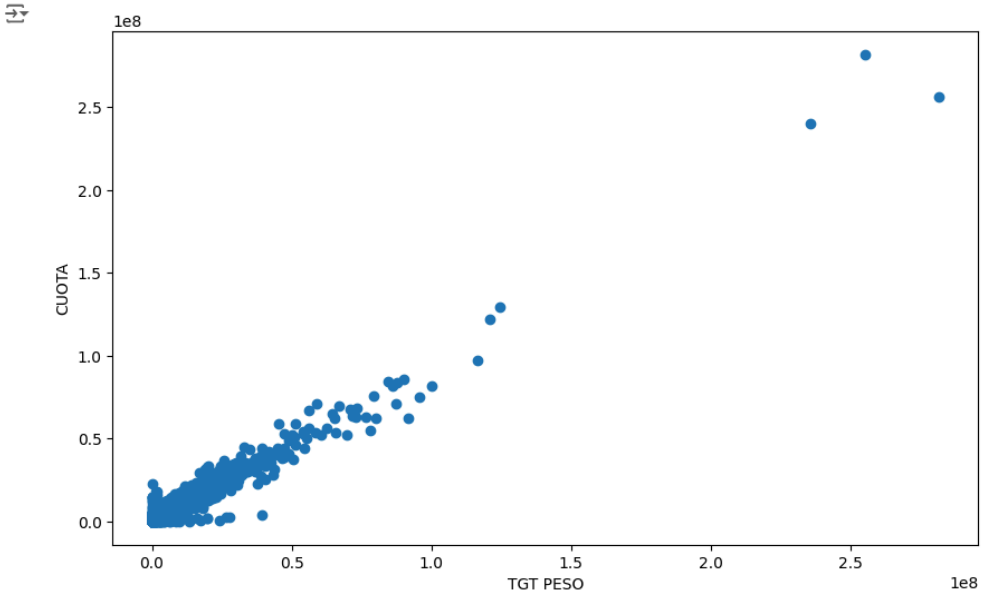

Figura 4: Correlación Cuota vs TGT Peso

A simple vista, parece que existe una relación positiva fuerte entre ambas variables: a medida que TGT PESO aumenta, CUOTA también tiende a incrementarse. Esto sugiere una posible correlación directa entre estas dos variables, lo cual podría ser útil para predecir CUOTA utilizando TGT PESO en un modelo. Adicionalmente se identifican datos lejanos que podrían ser indicios de valores atípicos.

Cuota vs TGT FC:

```
fig, ax = plt.subplots(figsize=(10,6))
ax.scatter(df['TGT FC'], df['CUOTA'])
ax.set_xlabel('TGT FC')
ax.set_ylabel('CUOTA')
plt.show()
```

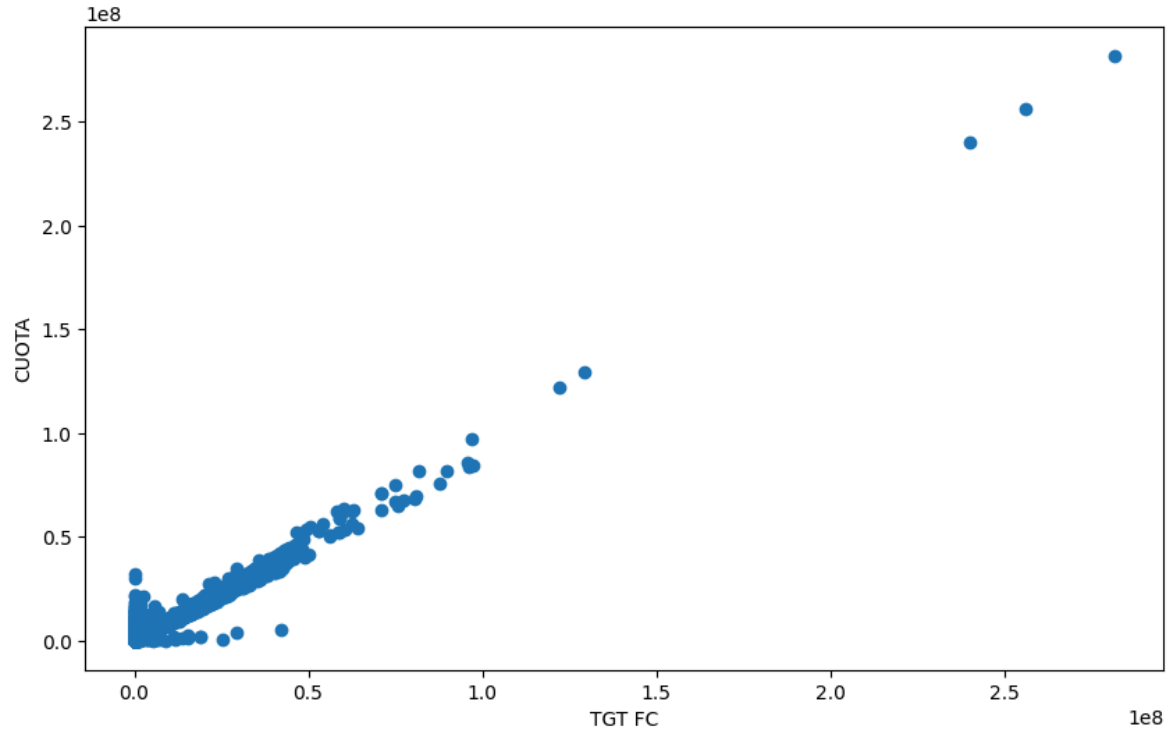

Figura 5: Correlación Cuota vs TGT FC

En este gráfico de dispersión, observamos la relación entre las variables TGT FC y CUOTA. Al igual que en gráfico anterior, se evidencia una fuerte relación positiva: a medida que TGT FC aumenta, CUOTA también lo hace de manera consistente. Esta relación sugiere que TGT FC podría ser una variable importante para predecir CUOTA.

Cuota VS MTD-1:

```
fig, ax = plt.subplots(figsize=(10,6))
ax.scatter(df['MTD-1'], df['CUOTA'])
ax.set_xlabel('MTD-1')
ax.set_ylabel('CUOTA')
plt.show()
```

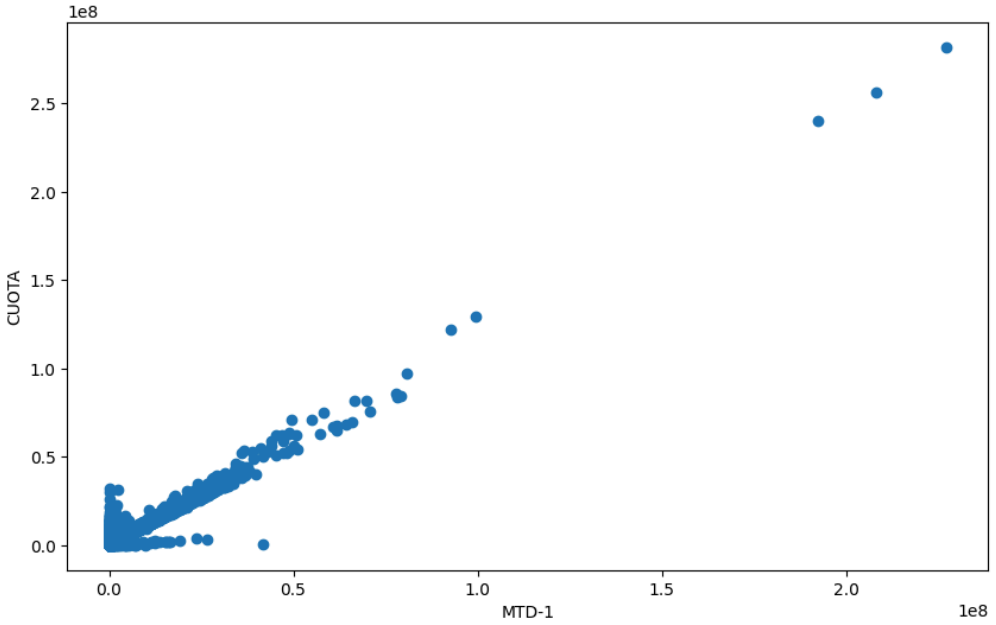

Figura 6: Correlación Cuota vs MTD-1

El gráfico de dispersión entre CUOTA y MTD-1 presenta un patrón similar a los gráficos previos, es probable que también muestre una relación positiva entre ambas variables. Esto significa que a medida que el valor de MTD-1 aumenta, CUOTA también tiende a incrementarse, indicando que MTD-1 podría ser un buen predictor para CUOTA.

Para mejorar la precisión de nuestro modelo y reducir posibles sesgos, procederemos a realizar un análisis y tratamiento de valores atípicos en las variables TGT PESO, TGT FC y MTD-1, las cuales han mostrado una alta correlación con la variable objetivo CUOTA. En los gráficos de dispersión previos, observamos la presencia de algunos puntos alejados del patrón general, lo que indica la existencia de estos datos.

El tratamiento de estos valores atípicos es esencial, ya que pueden influir de manera desproporcionada en el modelo, afectando la interpretación y la predicción. A través de este

proceso, identificamos y evaluamos la naturaleza de estos valores extremos para determinar si deben ajustarse, transformarse o excluirse, garantizando así una base de datos más robusta y confiable para el modelado.

```
#Calculo de valores atípicos


#Calculo de Q1 t Q3
Q1 = np.percentile(df['CUOTA'], 25, interpolation = 'midpoint')
Q3 = np.percentile(df['CUOTA'], 75, interpolation = 'midpoint')

#Cálculo del rango intercuartil
IQR = Q3 - Q1

#Cálculo de valor mínimo y máximo para los valores atípicos
VAInf = Q1 - 1.5*IQR
VASup = Q3 + 1.5*IQR

print(f'Valor atípico leve inferior:{VAInf}')
print(f'Valor atípico leve superior:{VASup}')
```

Figura 7: Cálculo de valores atípicos

Este código calcula los valores atípicos en la columna CUOTA de un DataFrame utilizando el método del rango intercuartil (IQR). Primero, obtiene los percentiles 25 (Q1) y 75 (Q3) de la columna CUOTA, que representan el primer y tercer cuartil, respectivamente. Luego, calcula el rango intercuartil (IQR) como la diferencia entre Q3 y Q1. Con el IQR, determina los límites inferior (VAInf) y superior (VASup) para los valores atípicos leves, utilizando la fórmula Q1−1.5×IQRQ1 − 1.5 Wtimes IQRQ1−1.5×IQR y Q3+1.5×IQRQ3 + 1.5 Wtimes IQRQ3+1.5×IQR, respectivamente. Finalmente, imprime estos límites, indicando los valores por

debajo o por encima de los cuales los datos pueden ser considerados atípicos leves.

Posteriormente se eliminan los valores anteriormente calculados:

```
# Se eliminan los valores atípicos
df = df.drop(df[df['CUOTA']>VASup].index)

#Reiniciar el indice
df.reset_index(drop=True, inplace=True)
```

Figura 8: Eliminación de valores atípicos

Procedemos a realizar nuevamente el cálculo de las columnas para saber la cantidad de datos que se eliminaron dado el paso anterior:

```
[ ] df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8192 entries, 0 to 8191
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   CREC OBT  8192 non-null   float64
 1   PROMEDIO  8192 non-null   float64
 2   PESO      8192 non-null   float64
 3   MTD-1     8192 non-null   float64
 4   TGT FC    8192 non-null   float64
 5   TGT PESO  8192 non-null   float64
 6   CUOTA     8192 non-null   float64
dtypes: float64(7)
```

Figura 9: Tipos de dato por columnas

De los 8,945 datos iniciales, nos quedamos con 8,192, lo que indica que se eliminaron 753 registros considerados como valores atípicos que podían influir negativamente en el rendimiento de nuestro modelo. Finalmente, generamos nuevamente los gráficos de

correlación para visualizar cómo ha cambiado la relación entre las variables luego de la eliminación de estos valores atípicos.

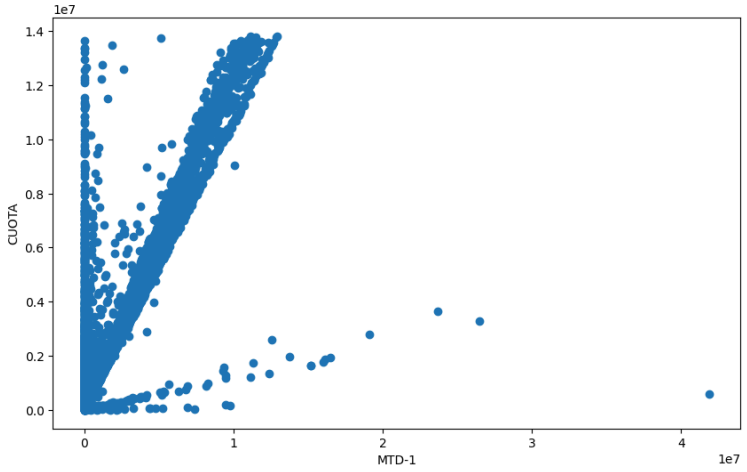

Figura 10: Correlación Cuota vs MTD-1 sin datos atípicos

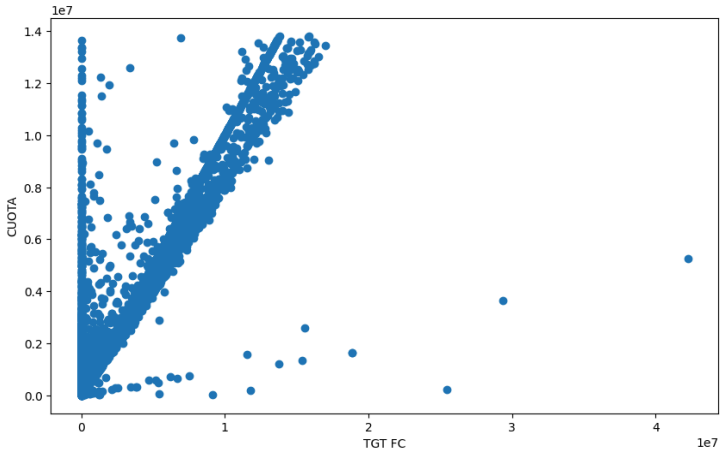

Figura 11: Correlación Cuota vs TGT FC sin datos atípicos

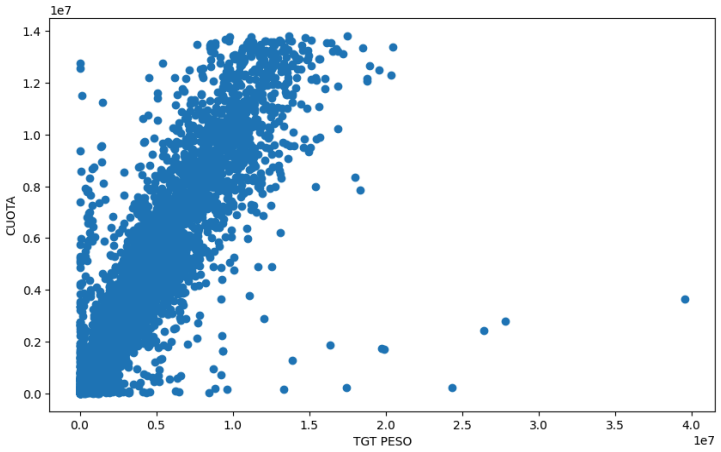

Figura 12: Correlación Cuota vs TGT Peso sin datos atípicos

Una vez que completamos la limpieza de datos en nuestra base, procedimos a generar un gráfico de densidad para visualizar la distribución de las variables. Estos gráficos nos permitirán identificar la dispersión en los datos, facilitando un análisis más profundo de la información.

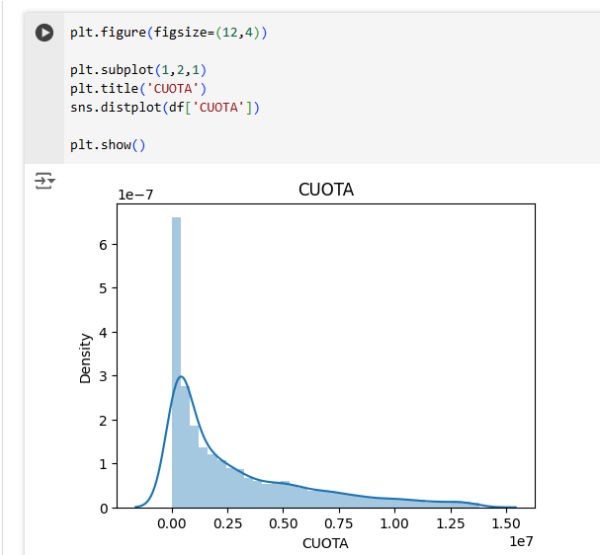

Figura 13:Distribución de las cuotas

Podemos observar que la distribución de las cuotas se encuentra mayormente concentrada en valores más pequeños, con una cola hacia la derecha. Esto indica que la mayoría de los puntos de venta tienden a tener cuotas bajas, mientras que un número reducido presenta cuotas significativamente más altas.

9.3. Objetivo específico 3: Crear una herramienta que permita la optimización y la visualización de la información almacenada bajo el mecanismo de la predicción.

9.3.1. Modelado

Primero, dividimos los datos en dos grupos con una relación de 90-10. Esto significa que el 90% de los datos se utilizará para el análisis principal, mientras que el 10% restante se reservará para pruebas y validaciones. Posteriormente, guardamos cada grupo en archivos separados en formato CSV, lo que nos permitirá acceder fácilmente a ellos más adelante para su uso en el modelo.

```python
# Porcentaje de filas para la primera parte (90%)
percentage_first_part = 0.90

# Número de filas para la primera parte
n_rows_part1 = int(len(df) * percentage_first_part)

# Obtener índices aleatorios para la primer parte
indices_part1 = df.sample(n=n_rows_part1, random_state=123).index

# Obtener índices para la segunda parte (resto de las filas)
indices_part2 = df.index.difference(indices_part1)

# Dividir el DataFrame en dos partes
f = df.loc[indices_part1]
p = df.loc[indices_part2]
```

```python
# Exportamos el dataframe a un archivo CSV
f.to_csv('/content/BD_CUOTAS_prep.csv', index=False)
p.to_csv('/content/BD_CUOTAS_prue.csv', index=False)
```

Figura 14: Distribución en datos de preparación y prueba

9.3.1.1. Random Forest

Para comenzar con el objetivo de modelación de la información, se realizó inicialmente un modelo de Random Forest debido a su capacidad para combinar múltiples árboles de decisión y generar predicciones robustas y precisas.

Como primer paso, se desarrolló un código para determinar el número óptimo de árboles que maximiza el desempeño del modelo. Para ello, se probó con diferentes configuraciones, comenzando desde un árbol y aumentando progresivamente en intervalos de 10, hasta alcanzar un máximo de 200 árboles. Este proceso permitió evaluar el impacto del número de árboles en métricas clave de desempeño, como precisión, sensibilidad y tiempo de ejecución, garantizando un balance adecuado entre exactitud y eficiencia computacional.

```python
## IMPORTAMOS LAS LIBRERÍAS ##
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split

## CARGAMOS LOS DATOS Y LOS DIVIDIMOS EN CONJUNTOS DE ENTRENAMIENTO Y PRUEBA ##
data = load_wine()
X = data.data
y = data.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

## EVALUAMOS EL MODELO CON DIFERENTES CANTIDADES DE ÁRBOLES (n_estimators) ##
n_estimators_list = range(1, 201, 10)  # Probamos de 1 a 201, de 10 en 10
r2_scores = []  # Guardamos los resultados de R^2

for n in n_estimators_list:
    rf_model = RandomForestRegressor(n_estimators=n, random_state=42)
    rf_model.fit(X_train, y_train)
    y_test_pred = rf_model.predict(X_test)
    r2 = r2_score(y_test, y_test_pred)
    r2_scores.append(r2)

## GRAFICAMOS LOS RESULTADOS ##
plt.figure(figsize=(10, 6))
plt.plot(n_estimators_list, r2_scores, marker='o')
plt.title('R^2 del modelo Random Forest vs. n_estimators')
plt.xlabel('Número de estimadores (n_estimators)')
plt.ylabel('R^2 Score')
plt.xticks(n_estimators_list)
plt.grid()
plt.show()
```

Figura 15: Estimación del hiperparámetro para el random forest

Obtenemos los siguientes resultados:

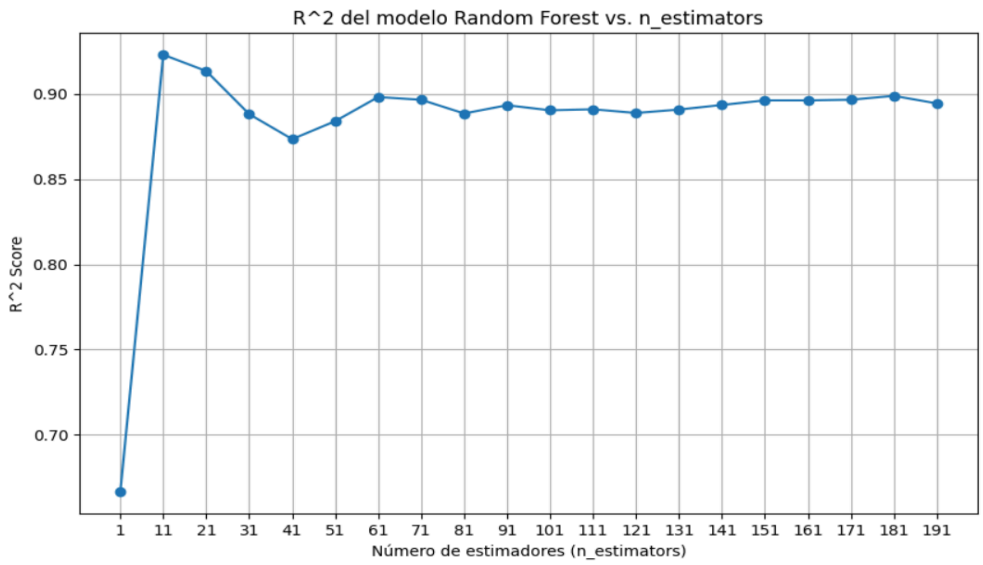

Figura 16: Número óptimo de árboles a utilizar

La gráfica resultante ilustra el desempeño del modelo medido a través del R2 Score evaluando cada configuración correspondiente a diferentes cantidades de árboles en el Random Forest. Este análisis permite identificar que el resultado óptimo se alcanza con el hiperparámetro correspondiente al número 11, que representa la mejor configuración dentro de los primeros 200 árboles que podríamos escoger para este modelo.

Se procede a realizar el modelado de datos con el hiperparámetro anteriormente calculado:

```python
## REALIZAMOS EL MODELO DE RANDMON FOREST CON 11 ÁRBOLES DE DECISIÓN**

model = RandomForestRegressor(n_estimators=11, random_state=42)
model.fit(X_train, y_train)

# Generamos predicciones
y_pred = model.predict(X_test)

# Calculamos la precisión del modelo en el conjunto de entrenamiento
precision_train = model.score(X_train, y_train)
print(f'Precisión en el conjunto de entrenamiento: {precision_train}')

# Calculamos el Error Cuadrático Medio (MSE) y la Raíz del Error Cuadrático Medio (RMSE)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

# Calculamos el Coeficiente de Determinación R^2 en el conjunto de prueba
r2 = r2_score(y_test, y_pred)

# Mostramos los resultados
print(f'Mean Squared Error (MSE): {mse}')
print(f'Root Mean Squared Error (RMSE): {rmse}')
print(f'R^2 Score: {r2}')
```

Figura 17: Modelo Random Forest

Se presentan los resultados:

```
Precisión en el conjunto de entrenamiento: 0.9927296946584235
Mean Squared Error (MSE): 309899644059.2821
Root Mean Squared Error (RMSE): 556686.3066928107
R^2 Score: 0.9705748772162102
```

Figura 18: Resultado Random Forest

Se observa que el modelo alcanza una precisión casi perfecta, superior al 99%. Este resultado puede interpretarse de dos maneras: o el modelo tiene una excelente capacidad para predecir la cuota, o podría haber aprendido excesivamente los patrones específicos del conjunto de datos de entrenamiento, cayendo en sobreajuste (overfitting). Para determinar cuál de estas situaciones es la correcta, procedemos a realizar los resultados reales de la predicción de las cuotas:

CREC OBT	PROMEDIO	PESO	MTD-1	TGT FC	TGT PESO	Predicción_CUOTA	CREC
0,25	$ 42.152.580	0,016949	$ 34.172.981	$ 42.716.226	$ 38.902.193	$ 3.638.796	-89%
0,25	$ 19.623.930	0,00789	$ 15.154.147	$ 18.942.684	$ 18.110.728	$ 2.846.448	-81%
0,25	$ 18.948.109	0,007619	$ 18.705.893	$ 23.382.366	$ 17.487.020	$ 2.744.905	-85%
0,25	$ 24.212.138	0,009735	$ 16.079.077	$ 20.098.846	$ 22.345.139	$ 2.846.579	-82%
0,25	$ 28.855.221	0,011602	$ 20.257.019	$ 25.321.274	$ 26.630.193	$ 4.041.258	-80%
0,25	$ 45.568.839	0,018322	$ 40.391.898	$ 50.489.872	$ 42.055.024	$ 3.638.796	-91%
0,25	$ 10.753.078	0,004324	$ 15.983.193	$ 19.978.991	$ 9.923.907	$ 2.687.747	-83%
0,25	$ 37.497.742	0,015077	$ 25.847.313	$ 32.309.142	$ 34.606.289	$ 3.734.434	-86%
0,25	$ 30.208.955	0,012146	$ 22.407.943	$ 28.009.928	$ 27.879.541	$ 4.041.258	-82%
0,25	$ 17.039.170	0,006851	$ 21.494.627	$ 26.868.284	$ 15.725.279	$ 3.340.257	-84%
0,25	$ 50.974.706	0,020496	$ 32.006.432	$ 40.008.040	$ 47.044.044	$ 3.638.796	-89%
0,25	$ 25.429.643	0,010225	$ 17.319.749	$ 21.649.687	$ 23.468.762	$ 3.373.251	-81%
0,25	$ 36.866.696	0,014823	$ 22.035.043	$ 27.543.803	$ 34.023.904	$ 3.734.434	-83%

Figura 19: Evaluación Random Forest

Como resultado del análisis de los datos, se concluye que el modelo presenta un caso evidente de sobreajuste. Esto se refleja en predicciones que subestiman significativamente las cuotas asignadas, alcanzando valores hasta 10 veces menores que los esperados. Este comportamiento indica que el modelo se ha ajustado demasiado a los patrones del conjunto de entrenamiento, perdiendo la capacidad de generalizar en nuevos datos.

Dado este resultado, el modelo de Random Forest queda descartado como una solución viable para el problema de asignación de cuotas.

Probar con modelos más complejos podría llevar al mismo problema de sobreajuste, ya que tienden a capturar incluso las particularidades irrelevantes del conjunto de entrenamiento. Sin embargo, un modelo de regresión lineal podría ser una alternativa adecuada, ya que su simplicidad reduce el riesgo de sobreajuste y permite interpretar de manera clara la relación entre las variables independientes y la cuota a predecir.

9.3.1.2. Regresión lineal

Ahora procedemos a desarrollar un modelo de regresión lineal, el cual se basa en establecer una relación matemática entre varias variables independientes que son llamadas las predictoras y una variable dependiente que para este ejercicio es la cuota. Este modelo asume

que dicha relación puede expresarse como una línea recta en el espacio de las variables mediante la siguiente ecuación matemática:

$$Y=\beta 0+\beta 1X1+\beta 2X2+\ldots+\beta nXn+\epsilon$$

Como primer paso, definimos la función que nos va a permitir realizar el modelamiento:

```
[ ]  #Funcion Lineal: dado una pendiente 'm', un valor x, y un coeficiente 'b', retorna el valor de 'y'

     def f(m, x, b):
         return (m*x)+b
```

Figura 20: Ecuación lineal

Se llevó a cabo el proceso de modelado utilizando un modelo denominado “model”, el cual fue entrenado con los conjuntos de datos **X_train** y **y_train**, definidos previamente. El modelo se ajustó utilizando las características especificadas como variables independientes y la variable objetivo.

Una vez entrenado, el modelo fue almacenado en un archivo específico dentro del entorno de Google Colab, asegurando su disponibilidad para futuros análisis o implementaciones. Esto permite conservar el estado del modelo entrenado sin necesidad de repetir el proceso de entrenamiento.

```python
# Creamos un modelo de regresión lineal y lo ajustamos a los datos.
model = LinearRegression(fit_intercept=True)  # Inicializa el modelo.
model.fit(X_train, y_train)  # Ajusta el modelo a los datos.

# Guardar el modelo entrenado en un archivo
model_path = '/content/modelo_regresion_lineal.joblib'
joblib.dump(model, model_path)

# Generamos valores para predecir y predecimos los valores de y
y_pred = model.predict(X_test)

# Calculamos la precisión del modelo en el conjunto de entrenamiento
precision = model.score(X_train, y_train)
print(f'Precisión en el conjunto de entrenamiento: {precision}')

# Calculamos el Error Cuadrático Medio (MSE) y la Raíz del Error Cuadrático Medio (RMSE)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

# Calculamos el Coeficiente de Determinación R^2 en el conjunto de prueba
r2 = r2_score(y_test, y_pred)
```

Figura 21: Modelo Regresión Lineal

9.3.2. Evaluación

A continuación, se evaluó el desempeño del modelo mediante las siguientes métricas clave:

- **Precisión del modelo:** Se calculó utilizando los datos de prueba, permitiendo medir qué tan bien predice el modelo en datos no vistos.
- **Error Cuadrático Medio (MSE):** Este valor proporciona una medida del promedio de los errores al cuadrado entre los valores predichos y los reales.
- **Raíz del Error Cuadrático Medio (RMSE):** Representa una interpretación más intuitiva del error, dado que está en las mismas unidades que la variable objetivo.
- **Coeficiente de Determinación (R2 Score):** Permite evaluar qué proporción de la variabilidad de la variable dependiente es explicada por el modelo.

```python
# Mostramos los resultados
print(f'Mean Squared Error (MSE): {mse}')
print(f'Root Mean Squared Error (RMSE): {rmse}')
print(f'R^2 Score: {r2}')
```

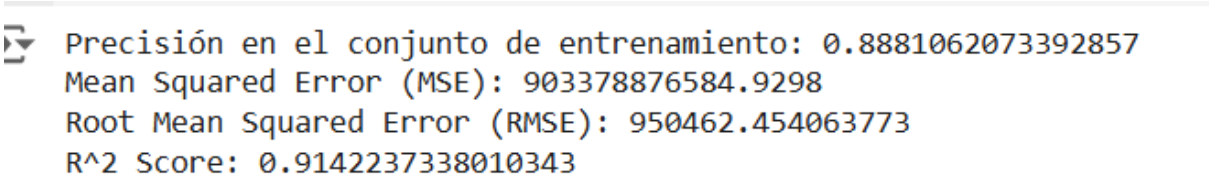

Figura 22: Resultados Regresión Lineal

Los resultados indican un desempeño razonablemente sólido para el modelo de regresión lineal. A continuación, se interpretan las métricas en detalle:

1. **Precisión en el conjunto de entrenamiento: 88.81%**
    - Este valor muestra que el modelo ajusta bien los datos de entrenamiento, aunque no alcanza una precisión perfecta, lo cual es deseable sabiendo que queremos evitar el tema de sobreajuste.
2. **Mean Squared Error (MSE): 903,378,876,584.93**
    - Aunque su valor absoluto es elevado, estoes normal sabiendo que la variable objetivo tiene una escala muy grande.
3. **Root Mean Squared Error (RMSE): 950,462.45**
    - Indica que, en promedio, el modelo tiene un error cercano a **950,462** en sus predicciones de cuota a asignar
4. **R^2 Score: 0.9142 (91.42%)**
    - Este coeficiente indica que el modelo explica aproximadamente el **91.42%** de la variabilidad en los datos. Es un indicador sólido de que el modelo captura la mayor parte de los patrones relevantes en los datos.

A continuación se realiza una gráfica de dispersión, comparando los valores predichos del modelo vs los valores reales y se realiza la línea ideal realizada por el modelo:

```python
import matplotlib.pyplot as plt

# Gráfico de dispersión de valores reales vs predichos
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, color='blue', label='Predicciones', alpha=0.6)
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red', linewidth=2, label='Línea ideal')

# Etiquetas y título del gráfico
plt.xlabel('Valores Reales')
plt.ylabel('Predicciones')
plt.title('Valores Reales vs Predicciones')
plt.legend()
plt.show()
```

Figura 23: Código del gráfico de dispersión

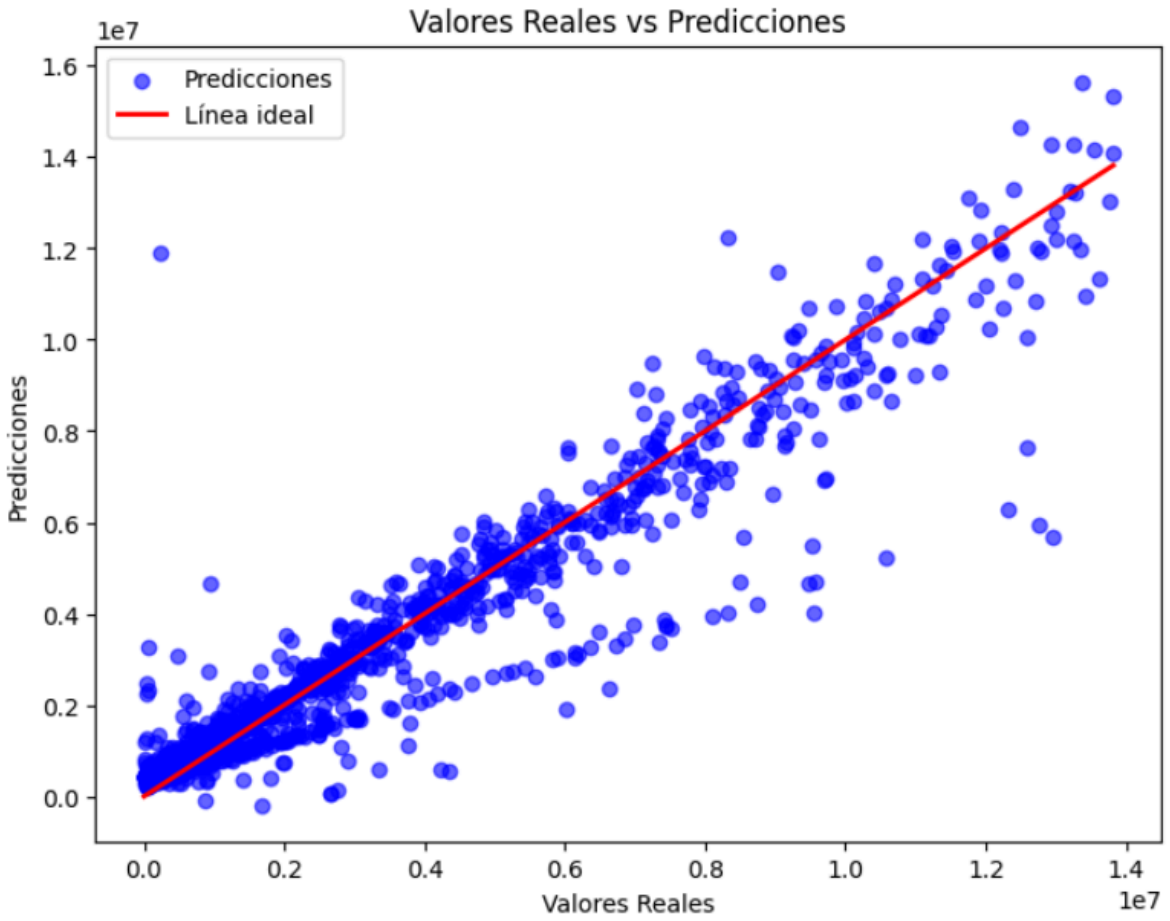

Figura 24: Gráfico de dispersión, valores reales vs predicciones

La dispersión de los puntos muestra una clara relación positiva entre los valores reales y las predicciones. Es decir, cuando los valores reales aumentan, las predicciones también

tienden a aumentar, lo que sugiere que el modelo está capturando correctamente la relación entre las variables.

Finalmente, el modelo entrenado se guarda en un archivo específico para su posterior uso en la etapa de despliegue. Este proceso asegura que el modelo pueda ser reutilizado sin necesidad de repetir el entrenamiento, lo que optimiza tiempo y recursos.

9.3.3. Despliegue

En la etapa de despliegue, donde se ejecuta el modelo previamente entrenado y guardado, el primer paso consiste en importar el modelo de regresión lineal que ha sido previamente entrenado.

```
[ ]  import joblib

     # Cargar el modelo desde el archivo
     model = joblib.load('/content/modelo_regresion_lineal.joblib')
```

Figura 25: Carga del modelo

Posteriormente, se ejecuta el modelo para generar las predicciones de cuotas, las cuales se guardan en un archivo llamado BD_CUOTAS_Resultados. Este archivo actúa como contenedor de los valores de las cuotas predichas por el modelo, permitiendo que estos resultados sean fácilmente accesibles para su posterior análisis o uso en otros procesos

```python
import pandas as pd
import joblib

# Cargamos el modelo
model_path = '/content/modelo_regresion_lineal.joblib'
model = joblib.load(model_path)

# Cargamos los nuevos datos a excel
nuevos_datos = pd.read_excel('/content/BD_CUOTAS_prue.xlsx', sheet_name='BD_CUOTAS_prue')

# Realizamos las predicciones
predicciones = model.predict(nuevos_datos)

# Convertimos las predicciones a un dataframe
predicciones_df = pd.DataFrame(predicciones, columns=['Predicción_CUOTA'])

resultados = nuevos_datos.copy()  # Copiamos los datos originales
resultados['Predicción_CUOTA'] = predicciones_df  # Agregamos las nuevas columnas con las predicciones

# Exportamos los resultados a excel
resultados.to_excel('/content/BD_CUOTAS_resultados.xlsx', index=False)

# Mostramos los primeros resultados
print(resultados.head())
```

Figura 26: Exportación de los resultados del modelo

Revisamos los resultados del modelo en el archivo anteriormente guardado:

CREC OBT	PROMEDIO	PESO	MTD-1	TGT FC	TGT PESO	Predicción_CUOTA
15%	$ 11.379.151	16%	$ 4.296.248	$ 4.940.685	$ 6.094.295	$ 9.914.431
15%	$ 10.114.801	14%	$ 3.818.887	$ 4.391.720	$ 5.417.151	$ 8.881.223
15%	$ 9.482.626	13%	$ 3.580.206	$ 4.117.237	$ 5.078.579	$ 8.364.619
15%	$ 8.218.276	11%	$ 3.102.846	$ 3.568.272	$ 4.401.435	$ 7.331.412
15%	$ 7.586.101	10%	$ 2.864.165	$ 3.293.790	$ 4.062.863	$ 6.814.808
15%	$ 6.953.926	9%	$ 2.625.485	$ 3.019.307	$ 3.724.291	$ 6.298.204
15%	$ 5.689.575	8%	$ 2.148.124	$ 2.470.342	$ 3.047.147	$ 5.264.997
15%	$ 2.535.561	3%	$ 2.416.527	$ 2.779.006	$ 1.357.962	$ 3.554.449
25%	$ 7.629.976	14%	$ 3.582.480	$ 4.478.100	$ 5.393.846	$ 8.813.970
25%	$ 6.782.201	12%	$ 3.184.427	$ 3.980.533	$ 4.794.530	$ 7.884.910
25%	$ 6.358.314	11%	$ 2.985.400	$ 3.731.750	$ 4.494.871	$ 7.420.380
25%	$ 4.285.179	8%	$ 2.877.016	$ 3.596.271	$ 3.029.314	$ 5.713.291
25%	$ 3.886.558	7%	$ 2.609.387	$ 3.261.734	$ 2.747.517	$ 5.223.909
30%	$ 4.516.088	15%	$ 1.893.278	$ 2.461.262	$ 3.381.006	$ 7.411.100
30%	$ 4.014.300	14%	$ 1.682.914	$ 2.187.788	$ 3.005.339	$ 6.628.851
30%	$ 3.763.407	13%	$ 1.577.732	$ 2.051.051	$ 2.817.505	$ 6.237.727
30%	$ 3.261.619	11%	$ 1.367.368	$ 1.777.578	$ 2.441.838	$ 5.455.479
30%	$ 3.010.725	10%	$ 1.262.185	$ 1.640.841	$ 2.254.004	$ 5.064.354

Figura 27: Evaluación del modelo

Es posible observar que el modelo realizó una predicción adecuada para las cuotas, presentando un ligero margen de error. Sin embargo, este margen es aceptable durante este experimento, ya que el modelo logra capturar la tendencia general de los datos y está dentro de un rango razonable. Aunque se pueden observar algunas variaciones entre las predicciones y los valores reales, el modelo proporciona resultados suficientemente confiables para la toma de decisiones, especialmente considerando que la precisión es adecuada para los propósitos del área en la asignación de las cuotas.

10. Conclusiones

1. Para que la empresa pueda implementar el modelo de predicción de manera adecuada y sostenible, es crucial cumplir con dos tareas fundamentales:

- **Mantener una base de datos dinámica para entrenamiento**:

  Es necesario contar con una base de datos que sirva como referencia para el entrenamiento del modelo. Esta base debe ser constantemente actualizada con nuevos datos, ya que un modelo entrenado en información limitada puede volverse obsoleto rápidamente.

- **Disponer de una base de datos con la estructura del modelo para predicciones**:

  Se debe diseñar una segunda base de datos, ya estructurada con las variables requeridas por el modelo para realizar predicciones. Esto asegura que el modelo pueda procesar la información de manera automática y eficiente.

2. El proyecto incluye un proceso de preprocesamiento de datos que se ejecuta automáticamente cada vez que se corre el archivo. Este preprocesamiento abarca las siguientes tareas clave:

- **Eliminación de columnas categóricas**:

Se identifican y eliminan aquellas columnas categóricas que no aportan directamente al análisis o que no son compatibles con el modelo de predicción utilizado. Esto asegura que los datos sean exclusivamente numéricos o que sean transformados adecuadamente si se requiere su inclusión en el modelo.

- **Cálculo y eliminación de valores atípicos**:

  Se identifican los valores atípicos dentro de las variables numéricas. Una vez detectados, estos valores se eliminan para evitar que distorsionen los resultados del modelo.

3. Se optó por un modelo de regresión lineal debido a su simplicidad y capacidad para evitar problemas de sobreajuste, que suelen presentarse al utilizar modelos más complejos, lo que podría generar errores en las predicciones y limitar la capacidad de generalización del modelo a nuevos datos. Este enfoque permite un balance entre precisión y robustez, asegurando un mejor desempeño en contextos reales.

11. Referencias

Abulibdeh, A., Zaidan, E., & Abulibdeh, R. (2024). Navigating the confluence of artificial intelligence and education for sustainable development in the era of industry 4.0: Challenges, opportunities, and ethical dimensions. *Journal of Cleaner Production*, *437*. https://doi.org/10.1016/j.jclepro.2023.140527

Arinez, J. F., Chang, Q., Gao, R. X., Xu, C., & Zhang, J. (2020). Artificial Intelligence in Advanced Manufacturing: Current Status and Future Outlook. *Journal of Manufacturing Science and Engineering*, *142*(11). https://doi.org/10.1115/1.4047855

Cakir, A., Akın, Ö., Deniz, H. F., & Yılmaz, A. (2022). Enabling real time big data solutions for manufacturing at scale. *Journal of Big Data*, *9*(1). https://doi.org/10.1186/s40537-022-00672-6

Coronado Medina, L. A. (2019). *Analítica de datos un estudio de caso de su uso para identificar riesgos estratégicos en grandes compañías de Medellín*.

Donta, P. K., Sedlak, B., & Casamayor Pujol, V. (2023). Governance and sustainability of distributed continuum systems: a big data approach. *Journal of Big Data*, *10*(1). https://doi.org/10.1186/s40537-023-00737-0

Dudycz, H., Stefaniak, P., & Pyda, P. (2022). Problems and Challenges Related to Advanced Data Analysis in Multi-Site Enterprises. *Vietnam Journal of Computer Science*, *9*(1), 1–17. https://doi.org/10.1142/S2196888822500063

Kabugo, J. C., Jounela, S. L., Schiemann, R., & Binder, C. (2020). Industry 4.0 based process data analytics platform: A waste-to-energy plant case study. *International Journal of Electrical Power and Energy Systems*, *115*. https://doi.org/10.1016/j.ijepes.2019.105508

Kahveci, S., Alkan, B., Ahmad, M. H., Ahmad, B., & Harrison, R. (2022). An end-to-end big data analytics platform for IoT-enabled smart factories: A case study of battery module assembly system for electric vehicles. *Journal of Manufacturing Systems*, *63*, 214–223. https://doi.org/10.1016/j.jmsy.2022.03.010

Latham, S., & Giannetti, C. (2023). A Tool to Combine Expert Knowledge and Machine Learning for Defect Detection and Root Cause Analysis in a Hot Strip Mill. *SN Computer Science*, *4*(5). https://doi.org/10.1007/s42979-023-02104-5

Lim, J. Bin, & Jeong, J. (2023). Factory Simulation of Optimization Techniques Based on Deep Reinforcement Learning for Storage Devices. *Applied Sciences*, *13*(17). https://doi.org/10.3390/app13179690

Schmitt, M. (2023). Automated machine learning: AI-driven decision making in business analytics. *Intelligent Systems with Applications*, *18*. https://doi.org/10.1016/j.iswa.2023.200188