Caracterización de Contribuyentes deudores del Impuesto Vehicular en el Departamento de Antioquia

Damian Patiño Montoya

Elizabeth Silva Rojas

Jhonatan Herrera Rios

Trabajo de Grado presentado como requisito para optar al título de:

Especialista en Big Data e Inteligencia de Negocios

Asesor: Juan Sebastián Parra Sánchez

Universidad Católica Luis Amigó

Facultad de Ingenierías y Arquitectura

Especialización en Big Data e Inteligencia de Negocios

Medellín, Colombia

2023

Resumen

Este trabajo de investigación se centra en el análisis y comprensión del perfil de los contribuyentes deudores del impuesto vehicular en el departamento de Antioquia utilizando técnicas de Aprendizaje Automático (Machine Learning). El objetivo principal es identificar patrones y tendencias de estos contribuyentes, considerando variables socioeconómicas, geográficas y tributarias. La evasión fiscal y la falta de cumplimiento son problemas comunes en muchos países, incluyendo Colombia, y tienen consecuencias económicas y sociales significativas. Por lo tanto, comprender los determinantes del incumplimiento tributario es fundamental para el desarrollo económico, social y financiero del país.

El presente proyecto se desarrolla mediante un proceso de investigación que involucra la extracción y transformación de los datos relacionados con los contribuyentes del impuesto vehicular. Se utilizó una técnica no supervisada de clusterización para agrupar a los contribuyentes en categorías con características similares, lo cual ayuda a comprender mejor su comportamiento e identificar los factores que influyen en el incumplimiento del pago del impuesto.

El trabajo se lleva a cabo utilizando herramientas de procesamiento de datos y equipos de cómputo adecuados, garantizando la confidencialidad y seguridad de la información proporcionada por la Secretaría de Hacienda de la Gobernación de Antioquia. La metodología utilizada es CRISP-DM, un proceso estándar para proyectos de minería de datos.

En la preparación y procesamiento de los datos, se ejecutaron diversas tareas en diferentes etapas. En la primera fase, llamada Entendimiento y comprensión del negocio, se obtuvieron las bases de datos necesarias y se realizó un inventario de fuentes de información. En la segunda fase, titulada Estudio y comprensión de los datos, se creó un repositorio de código fuente para el análisis exploratorio de los datos. En la tercera fase, denominada Preparación de los datos, se llevó a cabo la limpieza de los datos, la integración de los datos relevantes y la unión de los conjuntos de datos después de su preprocesamiento, finalmente se llegó a la etapa de

modelado, donde utilizando el algoritmo de K-Means, se pudieron obtener 5 grupos de deudores. A través del modelo realizado, se evalúan los resultados obtenidos con las características de los diversos clústeres.

Tabla de Contenido

Lista de Figuras

Lista de tablas

1. Introducción

La administración fiscal y la tributación son procesos de alta complejidad en los que intervienen varios actores como lo son los contribuyentes, autoridades fiscales y profesionales en impuestos. Es por esto, que el actuar de las personas responsables de dichos tributos muchas veces puede estar influenciado por las estructuras sociales, las normas y los roles que rigen actualmente la sociedad (Pickhardt. y Prinz ,2014).

Frecuentemente vemos como la evasión fiscal es un fenómeno común en nuestro país y generalizado en todos los niveles, que conlleva importantes consecuencias tanto económicas como sociales, lo que termina en la reducción de los ingresos públicos. Además, se genera un fenómeno inequidad horizontal porque las personas que tienen condiciones tributarias similares terminan con diferentes cargas fiscales; lo que, a su vez, deteriora el efecto de redistribución de los impuestos (Barone et al, 2011). Es por esto que la comprensión de los principales determinantes del incumplimiento fiscal se vuelve relevante para el desarrollo económico, social y financiero del país y las regiones.

Actualmente y debido a la gran importancia que ha adquirido en los últimos años el impuesto vehicular en el funcionamiento tributario de las regiones del país, este se ha establecido como una de las fuente de ingresos de mayor relevancia para financiar el desarrollo de infraestructura y servicios públicos en los diferentes departamentos del país; por lo que se hace necesario el desarrollo de herramientas y legislaciones para disminuir el riesgo de evasión y morosidad, así como garantizar una correcta y eficiente recaudación del rubro. La recaudación de impuestos el en departamento de Antioquia está regida por el Estatuto de Rentas, el cual tiene como finalidad la adopción de las rentas del departamento, y que contiene además las normas que regulan su administración, recaudo, determinación, discusión y cobro, así como su régimen sancionatorio y contravencional. (Asamblea Departamental de Antioquia, 2022)

El objetivo principal de esta investigación se centra en utilizar técnicas de Aprendizaje automático para analizar y comprender el perfil de los contribuyentes responsables del impuesto vehicular en Antioquia. Se busca identificar patrones y tendencias en el comportamiento fiscal de estos, considerando variables socioeconómicas, geográficas y fiscales. Esto permitirá obtener una visión integral y detallada de los responsables del impuesto vehicular.

Se plantea obtener resultados que proporcionen a las autoridades fiscales del Departamento de Antioquia información valiosa para mejorar la gestión del impuesto vehicular. Estos resultados pueden incluir la segmentación de los contribuyentes en grupos homogéneos con características similares, la implementación de estrategias de recaudación más efectivas y la generación de políticas tributarias más equitativas y justas.

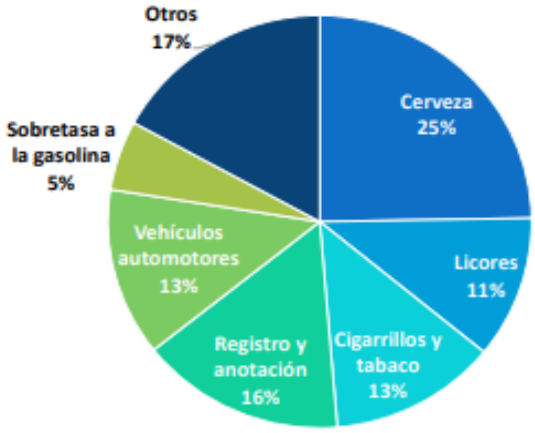
2. Planteamiento del Problema

El impuesto vehicular representa uno de los elementos tributarios con mayor importancia en la cartera de la Secretaría de Hacienda de la Gobernación de Antioquia, por lo que su recaudo eficiente garantiza el desarrollo de programas y proyectos de carácter social para la población del departamento.

Según datos publicados por el Departamento Nacional de Planeación (DNP) donde se presenta el desempeño fiscal de los diferentes departamentos durante el año 2021, el impuesto vehicular representó el 13% de los ingresos tributarios departamentales en el orden nacional, tal y como se registra en la Figura 1:

Figura 1

Principales Ingresos Tributarios Departamentales 2021.



Nota: Gráfico de la distribución porcentual del ingreso tributario. Tomado de DNP (2021).

Dado esto, es importante recalcar el alto nivel de participación que tiene este impuesto en el presupuesto fiscal antioqueño. Actualmente el 80% de los tributos relacionados con el impuesto vehicular, son destinados para la gestión de inversión de la gobernación de Antioquia y el 20% restante, para la inversión en el municipio de domicilio que registra el propietario del vehículo. En

consecuencia, el no pago de este impuesto confluye en la generación de una serie de actividades institucionales y tributarias, que representan una disposición de recursos adicionales para garantizar la correcta recaudación y también pone en riesgo la ejecución presupuestal establecida para el plan de desarrollo de la gobernación.

Por otro lado, debido al alto volumen de contribuyentes del impuesto vehicular pertenecientes al departamento de Antioquia, y a la alta edad de la cartera que se genera, se hace necesario realizar una categorización de los contribuyentes que se encuentran registrados como responsables del pago de la obligación, con el objetivo de determinar las condiciones óptimas que permitan mejorar la recaudación y recuperación de cartera.

Partiendo de lo anterior surge la siguiente pregunta problema:

¿Cómo caracterizar los contribuyentes que presentan mora en el pago del impuesto vehicular en el departamento de Antioquia?

3. Justificación

El proceso de caracterización de los contribuyentes es de vital importancia en la gestión fiscal y tributaria, esto permitirá el desarrollo sistemático de estrategias fundamentadas en la información, que ayudarán a identificar las posibles causas por las cuales un contribuyente no cumple con su obligación tributaria.

Según datos de la secretaría de hacienda, entre los periodos 2018 y 2022, el saldo de deuda correspondiente a impuesto vehicular en el departamento asciende a: Doscientos dieciocho mil, ciento treinta y ocho millones, noventa y nueve mil ciento cuatro pesos ($218.138.099.104); como se muestra a continuación en la Figura 2:

Figura 2

Partidas abiertas -SAP Gobernación de Antioquia

Clave período	Total Impte.Local
2018	$ 57.316.435.097
2019	$ 42.331.018.485
2020	$ 51.887.011.646
2021	$ 32.167.928.325
2022	$ 34.435.705.551
Total	**$ 218.138.099.104**

Nota: Elaboración propia

Por lo anterior, se hace de vital importancia realizar la categorización de los contribuyentes deudores que se encuentran registrados como responsables del pago de la obligación del impuesto vehicular, detallando los diferentes niveles que surgen del análisis de los datos. En este contexto, la implementación de técnicas de Machine Learning (ML) permitirá una identificación más precisa de patrones y características de los contribuyentes, así como la identificación de posibles factores que influyen en el comportamiento de pago, lo que posibilitará a la Secretaría

de Hacienda de la Gobernación de Antioquia diseñar estrategias de recaudo y recuperación de la cartera más eficientes.

4. Marco de Referencias

A continuación, se muestra el marco de referencias del presente trabajo, donde se especifican los principales conceptos a tratar en el desarrollo de los objetivos planteados.

Big Data: Macrodatos e inteligencia de datos son alternativas en español a la voz inglesa Big Data, que se emplea en el sector de las tecnologías de la información y de la comunicación (TIC) para aludir a un conjunto de datos que, por su volumen, variedad y velocidad de producción, no pueden ser analizados utilizando procesos o herramientas tradicionales (Carriles, 2018).

Machine Learning: Se utiliza para enseñar a las máquinas cómo manejar los datos más eficientemente. Este aprendizaje automático se basa en diferentes algoritmos para resolver problemas de datos. A los científicos de datos les gusta señalar que no existe un único tipo de algoritmo que sea el mejor para resolver un problema. El tipo de algoritmo empleado depende del tipo de problema que desee resolver, el número de variables, el tipo de modelo que mejor se adapte (Mahesh y Batta, 2018).

Segmentación del Mercado: La segmentación de mercado es la agrupación de mercados en grupos de consumidores homogéneos, donde cada grupo (parte) puede ser elegido como mercado objetivo (target) para la comercialización de un producto (Cahyana et al, 2020).

K-Means: Es un algoritmo de clustering no supervisado muy utilizado debido a su sencilla interpretación y utilización. Tiene como objetivo agrupar objetos en un número fijo de grupos o clústeres basándose en sus características, logrando asociar observaciones similares para descubrir patrones que a simple vista se desconocen. (Prada Conde, 2022).

K-Nearest-Neighbors: El algoritmo de k-vecinos más cercanos, también conocido como KNN o k-NN, es un clasificador de aprendizaje supervisado no paramétrico, que utiliza la

proximidad para hacer clasificaciones o predicciones sobre la agrupación de un punto de datos individual. Si bien se puede usar para problemas de regresión o clasificación, generalmente se usa como un algoritmo de clasificación, partiendo de la suposición de que se pueden encontrar puntos similares cerca uno del otro. (IBM, 2022)

Clustering: La agrupación puede considerarse el problema de aprendizaje no supervisado más importante; entonces, como cualquier otro problema de este tipo, se trata de encontrar una estructura en una colección de datos no etiquetados. Por lo tanto, un clúster es una colección de objetos que son "similares" entre ellos y son "diferentes" a los objetos pertenecientes a otros clústeres. (Omran et al, 2007)

Clustering Jerárquico: El algoritmo de agrupamiento aglomerativo jerárquico o agrupamiento de abajo hacia arriba (enfoque ascendente), busca construir una jerarquía de conglomerados de tipo árbol. Parte esencialmente de un grupo individual, es decir, cada punto de datos se considera grupo de un solo elemento o hoja. Luego cada grupo calcula su distancia entre sí, uniéndose aquellos clústeres que tengan la menor diferencia, formando un nodo (Prada Conde, 2022)

DBSCAN:” Density-based Spatial Clustering of Applications with Noise” es una de las técnicas más comunes de algoritmos de clustering, que se basa en la densidad del objeto para identificar clústeres de cualquier forma en un conjunto de datos con ruido y valores atípicos. En este caso, el término densidad se refiere al número de observaciones que se encuentran en una misma zona. Por lo tanto, se basa en que, si un punto en particular pertenece a un clúster, debe estar cerca de otros puntos en dicho clúster. El modelo depende de dos hiperparámetros muy importantes, que representan el número de puntos para considerar una zona como densa y la distancia entre dichos puntos (Prada Conde, 2022).

Hecho Generador: Toda cuenta u orden de pago a favor de personas naturales, jurídicas o cualquier forma de asociación jurídica que realice el Departamento de Antioquia, las entidades descentralizadas y entidades del orden nacional que funcionen en el Departamento de Antioquia, provenientes de contratos, pedidos y facturas (Asamblea Departamental de Antioquia, 2022)

Obligación Tributaria: La obligación tributaria es el vínculo jurídico en virtud del cual una persona natural, jurídica, sociedad de hecho o sucesión ilíquida está obligada a pagar a la Autoridad Tributaria Departamental una suma determinada de dinero, por haber realizado el hecho generador previsto en la Ley o en una ordenanza; además, es necesario cumplir con los deberes formales que se derivan de la obligación sustancial (Asamblea Departamental de Antioquia, 2022).

Vehículos Gravados: Están gravados con el impuesto los vehículos automotores nuevos, usados y los que se internen temporalmente en el territorio nacional (Asamblea Departamental de Antioquia, 2022).

Contribuyente: El contribuyente es una persona natural o jurídica que debe cumplir con las obligaciones tributarias impuestas por la normativa tributaria (Asamblea Departamental de Antioquia, 2022).

Impuesto Vehicular: El Impuesto sobre vehículos automotores es un impuesto de carácter directo, que recae sobre la propiedad o posesión de los vehículos gravados, que se encuentren matriculados. El Departamento de Antioquia podrá liquidar anualmente el impuesto sobre vehículos automotores (Asamblea Departamental de Antioquia, 2022).

Cartera: Son aquellos valores que corresponde a derechos exigibles por parte del Ministerio. Para efectos de los contratos y los convenios suscritos y liquidados por el Ministerio, la cartera de la entidad contablemente se constituye con el acta de liquidación bilateral suscrita por las partes o la Resolución por la cual se liquida unilateralmente el contrato o convenio debidamente ejecutoriada, siempre y cuando en estos se establezca un valor pendiente por reintegrar al Ministerio (Asamblea Departamental de Antioquia, 2022).

Fisco: Se refiere al Estado, que, en su carácter de persona jurídica, se inviste de potestad tributaria como organismo recaudador, para lograr por medio del cobro de impuestos, tasas y contribuciones, solventar necesidades de interés general y particular de los contribuyentes. La legislación tributaria o fiscal regula esa potestad del Estado dentro del marco legal (Nájera, 2022).

5. Antecedentes

Con base a los objetivos establecidos para el presente trabajo y fruto de la revisión de literatura realizada, se presentan algunos artículos de investigación que abordan el tema de la caracterización de clientes, usando modelos de ML tales como la cauterización.

Verdenhofs y Tambovceva (2019) analizaron algunos escenarios de segmentación para determinar los tipos de clientes. Estos se abordaron desde el Big Data por medio de la minería de datos con el fin de clasificar a los clientes con condiciones de pago adecuadas y sus respectivos descuentos, clientes con información básica suministrada como dirección, sexo, teléfono, fechas de compra y cuenta bancaria, esto con el fin de determinar el uso del producto. El ejercicio determinó que la empresa podía clasificar los clientes que permanecerán, y ofrecerles descuentos, y de esta manera definir qué clientes se podrían inactivar para no realizar ningún beneficio a estos.

Al saber quiénes son los mejores clientes o usuarios típicos Weinstein (2001) plantea que se puede segmentar por uso, segmentar 80/20, segmentar por nivel de satisfacción e incluso prescindir de los clientes con menor peso de compras; esto garantizará que el enfoque operativo siempre esté alineado con los clientes de mayor relevancia dentro de la operación. Así también, Zhang et al, (2021) realizaron un ejercicio de caracterización por la afinidad de los clientes respecto a sus compras. Se implementó un proceso de clusterización para una gran minorista en EE. UU, y allí se definieron por medio de estadística descriptiva 35 clúster o grupos de clientes, así también se aplicó el algoritmo de Louvain obteniendo una categorización integral de los clientes según sus compras históricas. Este procedimiento permitió a la organización generar un enfoque para diseñar y desarrollar estrategias de captación y fidelización de clientes que permitieran crear un CRM.

La categorización de clientes también puede ser utilizada para medir riesgos y clasificación crediticia de los clientes, como lo define Silva et al, (2022). En este caso la

metodología abordada de abstracción de datos se basó en la segmentación y la recolección de transacciones de clientes por sectores demográficos. Se utilizaron 3 modelos estadísticos y la formación de clúster para describir los compartimentos del cliente. Este ejercicio dio como resultado los clústeres de clientes superestrella, clientes típicos y clientes inactivos; y con esta información la empresa pudo hacer una gestión de sus clientes de acuerdo con el tipo de cartera y generar ventajas competitivas frente a la competencia para retener e incentivar a los clientes y tener una mejor cuota de mercado.

Por otra parte, para lograr segmentar los clientes de una compañía de seguros para generar estrategias de captación y retención del cliente, Qadadeh y Abdallah (2018) plantean que el operador correcto para esta clasificación es el método K-Means. Por medio de esta clusterización se definieron 6 centroides que contenían las características financieras y personales de los clientes, así como sus cualidades determinantes para obtener un seguro de vida. Se concluyó que la entidad debía generar condiciones de marketing para cada clúster, donde según sus condiciones particulares se establecía una promoción especial para cada grupo de clientes. Así también, Ho y Lyu (2014) llevaron a cabo un estudio en Australia sobre los patrones de gasto de los consumidores al recibir un reembolso de impuestos y cómo se pueden segmentar en diferentes grupos. La metodología utilizada incluyó una encuesta en línea para recopilar datos de los consumidores. A través del análisis de clúster K-means, los autores identificaron cuatro grupos de consumidores: ahorradores, gastadores racionales, gastadores compulsivos y pagadores de deudas. Según los resultados concluyeron que la segmentación basada en el análisis de clúster K-means es una herramienta útil para comprender los patrones de gasto de los consumidores y desarrollar estrategias de marketing más efectivas.

Para agrupar diferentes segmentos de clientes Babu et al, (2018) obtuvieron datos de varias organizaciones comerciales, como tiendas minoristas, inteligentes y otras tiendas. Aplicando el modelo de clustering DBSCAN a los registros de un año, pudieron reconocer grupos en diferentes tipos de espesor en datos de altas dimensionalidades. Esto les permitió identificar

los elementos de mayor rentabilidad, así como los clientes de alto valor y bajo riesgo. Finalmente se logró observar que mantener a los antiguos clientes genera más ganancias en comparación con la captación de nuevos.

De igual forma Dr. Arivazhagan y Dr. Vijaiprabhu (2022) describieron la técnica de segmentación conocida como clúster jerárquico, para encontrar grupos homogéneos de clientes. El estudio se aplicó a dos conjuntos de datos de los sectores de Banca y Telecomunicaciones. Se analizaron datos de información básica, como la ubicación del cliente, la edad, el estado civil, el trabajo y la educación, y también se consideraron sus regularidades de comportamiento comercial; con el fin de desarrollar estrategias de mercadeo y mejorar la velocidad de respuesta de las empresas ante sus necesidades. Los resultados mostraron que el modelo presentó dificultades respecto a su procesamiento y densidad, pero aun así pudo clasificar en tres grupos a los clientes de los tipos de empresa mencionados, en relación con sus cargos mensuales, antigüedad y edad.

Aunque estos modelos de aprendizaje automático representan una herramienta de gran importancia para las empresas en la actualidad, la dimensionalidad de algunos datos que pueden contener la información de los clientes, se convierte en un reto para la efectividad de estos algoritmos. Es por esto que, con el fin de que los algoritmos básicos cuenten con la implementación de varios métodos de segmentación, surge la implementación de algoritmos híbridos, así es como Cahyana et al, (2020) definen su procedimiento para la metodología de clústeres híbridos haciendo uso de la clusterización no jerárquica y la clusterización jerárquica, permitiendo extraer resultados del análisis en función de las características del cliente.

Para Zhou et al, (2022) el objetivo principal del agrupamiento híbrido es buscar resultados mejores y más sólidos, utilizando la combinación de información y resultados obtenidos de varios agrupamientos. El agrupamiento híbrido puede proporcionar mejores soluciones teniendo en cuenta sus puntos fuertes, se obtiene combinando los resultados de diferentes agrupamientos de

robustez, escalabilidad, estabilidad y flexibilidad que el proporciona el agrupamiento básico métodos. La precisión, la corrección y la estabilidad son características importantes métodos.

Estos casos de estudio demuestran que el uso de los modelos no supervisados de ML ha permitido a una gran cantidad de entidades económicas identificar características de sus clientes para poder así, desarrollar en base a estas condiciones, elementos de valor no solo para sus clientes, sino herramientas de gestión del riesgo financiero, optimización y estabilización de carteras, fidelización y mercadeo estratégico.

6. Objetivos

Objetivo General

Caracterizar a los contribuyentes que presentan mora en el pago del impuesto vehicular en el departamento de Antioquia entre los años 2018 y 2022, haciendo uso de técnicas no supervisadas de Machine Learning como el agrupamiento o clustering.

Objetivos Específicos

- Realizar el proceso de extracción y transformación de los datos correspondientes a los responsables del impuesto vehicular.

- Construir un modelo de clusterización a partir de los resultados obtenidos al valorar las diferentes técnicas de aprendizaje realizadas al data set destinado para este proceso.

- Identificar patrones en los datos de los contribuyentes del impuesto vehicular para entender mejor su comportamiento y determinar qué factores pueden afectar su cumplimiento tributario.

- Evaluar los hallazgos encontrados sobre la categorización de los contribuyentes del impuesto vehicular.

7. Viabilidad

Disponibilidad: Para el procesamiento de datos requerido para el presente proyecto, se utilizaron principalmente dos equipos de cómputo: HP RIZEN 5500- 16 GB, con sistema operativo Windows 11 y LEGION Core i7 novena generación-16 GB con sistema operativo kali Linux. Como herramientas de procesamiento de datos se implementó un repositorio en GITLAB para brindar acceso a todos los integrantes del grupo en cualquier momento y en tiempo real, además se usaron herramientas como Power BI y Google Colab. Se implementó el entorno de desarrollo con el IDE PyCharm para el uso del lenguaje Python durante el progreso del proyecto.

Alcance: El desarrollo del presente proyecto se basa en la implementación de diferentes modelos de clúster para la identificación de patrones y características de los contribuyentes que presentan mora en sus pagos de impuesto vehicular. Posteriormente se realizará una socialización con los equipos de la gobernación de Antioquia para las tomas de decisiones pertinentes al respecto.

Implicaciones: La realización del presente trabajo de grado requirió de algunos elementos legales, ya que al tratarse de información sensible y de uso exclusivo del área de tesorería de la gobernación de Antioquia, se tenían que delimitar los usos y alcances del proyecto. Para la obtención de los datos se realizó un derecho de petición por medio del cual se accedió al suministro de la información condicionado a puntos como la anonimización de datos y Habeas Data. Así también se requirió la aclaración de la información recibida por parte de las áreas respectivas por medio de reuniones, donde se suministraron diccionarios de datos y elementos importantes a tener en cuenta para el análisis de la información y su manejo. Se empleará la metodología CRISP-DM.

Garantías: Se garantiza absoluta reserva de la información de la información suministrada por el área de tesorería de la gobernación de Antioquia.

Consecuencia: La segmentación y categorización de los contribuyentes del impuesto vehicular permitirá identificar las principales características de los responsables de dicho rubro, así como los posibles factores que inciden en el incumplimiento del pago de la obligación. Esto se realiza con el fin de diseñar estrategias efectivas para la recuperación de la cartera por parte de la Secretaría de hacienda de la gobernación de Antioquia.

8. Metodología

Para el presente trabajo la metodología a implementar es el modelo de proceso estándar para minería de datos CRISP – DM, la cual se describe en términos de un modelo jerárquico, que consta de conjuntos de tareas descritas en cuatro niveles de abstracción (de general a específico): fases, tarea genérica, tarea especializada e instancia de proceso (Pete Chapman et al, 2000).

En el nivel superior, el proceso de minería de datos se organiza en varias fases; cada fase consta de varias tareas genéricas de segundo nivel. Este segundo nivel se llama genérico, porque pretende ser lo suficientemente general para cubrir todas las situaciones posibles de minería de datos.

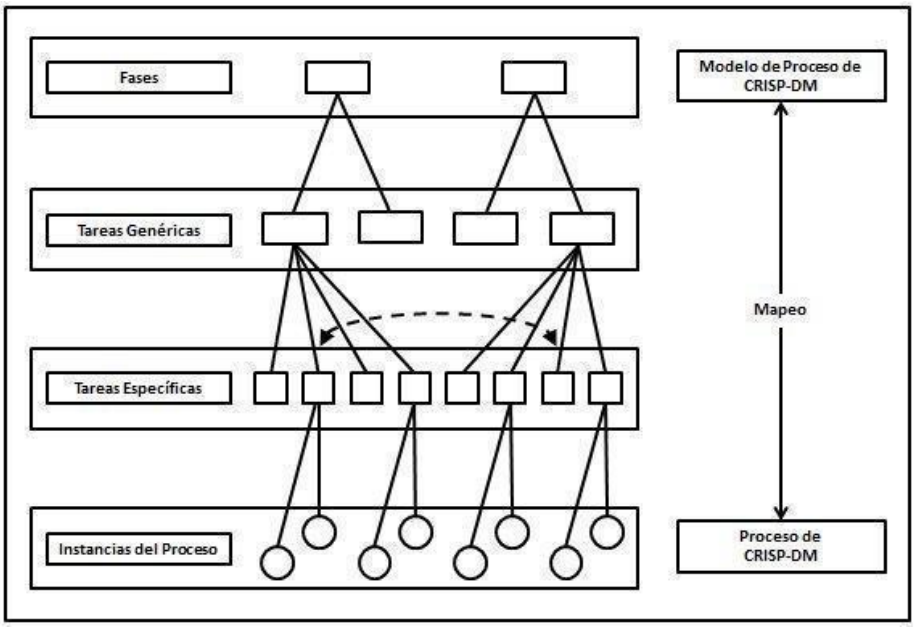
Las tareas genéricas pretenden ser lo más completas y estables posible. Completo significa que cubre tanto el proceso completo de minería de datos como todas las posibles aplicaciones de minería de datos. Estable significa que el modelo debería ser válido para desarrollos aún imprevistos, como nuevas técnicas de modelado. El tercer nivel, el nivel de tareas especializadas, es el lugar para describir cómo se deben llevar a cabo las acciones de las tareas genéricas en ciertas situaciones específicas.

Finalmente, el cuarto nivel, la instancia de proceso, es un registro de las acciones, decisiones y resultados de un compromiso real de minería de datos. Una instancia de proceso se organiza de acuerdo con las tareas definidas en los niveles superiores, pero representa lo que realmente sucedió en un compromiso particular, en lugar de lo que sucede en general.

En la Figura 3 se puede observar el orden de los mencionados niveles:

Figura 3

Desglose de cuatro niveles de la metodología CRISP-DM



Nota: Cuadro de los niveles de abstracción de la metodología CRISP-DM Tomado de Peralta (2014).

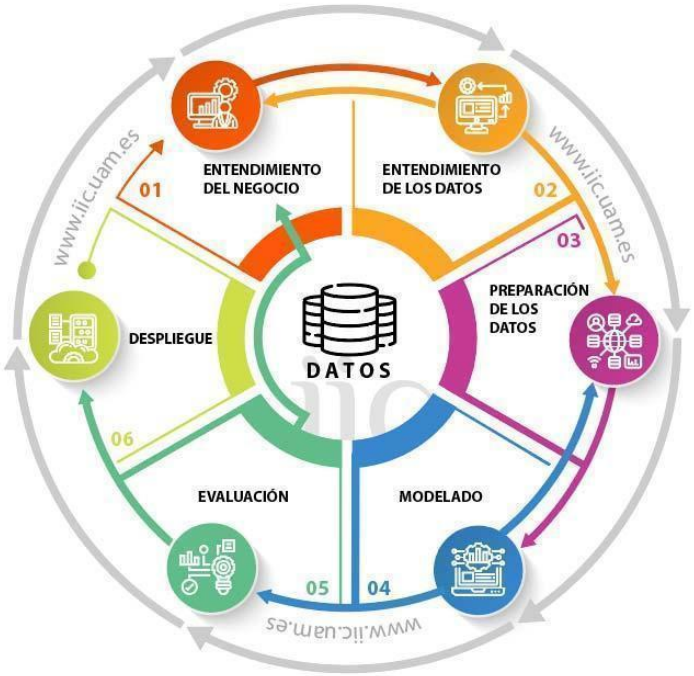
El ciclo de vida de un proyecto de minería de datos consta de seis fases. La secuencia de las fases no es rígida. Siempre es necesario avanzar y retroceder entre las diferentes fases. Depende del resultado de cada fase qué fase o qué tarea particular de una fase debe realizarse a continuación. Las flechas indican las dependencias más importantes y frecuentes entre fases.

El círculo exterior simboliza la naturaleza cíclica de la minería de datos en sí. La minería de datos no termina una vez que se implementa una solución. Las lecciones aprendidas durante el proceso y de la solución implementada pueden desencadenar nuevas preguntas comerciales, a menudo más enfocadas. Los procesos posteriores de minería de datos se beneficiarán de las experiencias de los anteriores.

La Figura 4 muestra las fases de un proceso de minería de datos.

Figura 4

Diagrama del ciclo CRISP-DM



Nota: Gráfico de diagrama sobre el ciclo de la metodología CRISP-DM Tomado de (Haya, 2022).

A continuación, se muestran objetivos propuestos en el trabajo, se utilizará la metodología con sus 6 fases; esta metodología integra todas las actividades necesarias para el desarrollo de este estudio, desde la fase inicial de comprensión del negocio hasta el despliegue del modelo predictivo propuesto como solución a través del uso de técnicas de ML. La descripción de las fases de la aplicación del modelo CRISP – DM en la ejecución del presente proyecto en base a los objetivos planteados para el mismo.

Tabla 1

Metodología CRISP-DM enfocado en objetivos específicos

OBJETIVO ESPECÍFICO	ACTIVIDAD	ENTREGABLE
1- Realizar el proceso de extracción y transformación de los datos correspondientes a los responsables del impuesto vehicular.	**FASE 1: ENTENDIMIENTO Y COMPRENSIÓN DEL NEGOCIO** -Obtención de las bases de datos -Identificar las características de los datos por medio de un inventario de fuentes.	Tabla de inventario de fuentes
	FASE 2: ESTUDIO Y COMPRENSIÓN DE LOS DATOS -Desarrollo de entornos de trabajo y procesamiento de datos	-Repositorio en GitLab. Entorno para procesamiento de los datos.
	FASE 3: PREPARACIÓN DE LOS DATOS -Realizar la limpieza de los datos -Realizar la integración de los datos relevantes definidos en los sets de datos obtenidos -Realizar la unión de los sets de datos después de su preprocesamiento, eliminación de datos nulos, atípicos e imputación de datos.	-Librerías propias para limpieza de datos -Archivos parquet con las uniones de variables relevantes de cada set de datos. -Tablas unión -union_all

2- Construir un modelo de clusterización a partir de los resultados obtenidos al valorar las diferentes técnicas de aprendizaje realizadas al data set destinado para este proceso.	**FASE 4: MODELADO** -Seleccionar la técnica de modelado a aplicar, acorde con los datos y objetivos. -Construir el modelo a partir de la aplicación de la técnica seleccionada.	-Modelo no supervisado de ML con sus respectivas métricas -Informe de construcción de los modelos
3- Identificar patrones en los datos de los contribuyentes del impuesto vehicular para entender mejor su comportamiento y determinar qué factores pueden afectar su cumplimiento tributario.	**FASE 5: EVALUACIÓN** -Determinar las características de los patrones obtenidos al clasificar los contribuyentes. -Realizar revisión de los resultados obtenidos con el modelo construido respecto con los propósitos definidos.	- Análisis y comprensión de los clústeres generados, creando así un análisis descriptivo de estos.
4-Evaluar los hallazgos encontrados sobre la categorización de los contribuyentes del impuesto vehicular. hacienda.	**FASE 5: EVALUACIÓN** -Describir los clústeres obtenidos	

Fuente: Elaboración propia

Fase I. Entendimiento o comprensión del negocio: A partir del uso de bases de datos relacionadas con los contribuyentes del impuesto vehicular del departamento de Antioquia, el área de tesorería de la Gobernación de Antioquia requiere de una categorización de los contribuyentes morosos que se encuentran registrados como responsables del pago de la obligación, con el objetivo de determinar sus principales características, así como definir y adaptar las estrategias que permitan mejorar la recaudación de cartera, y así como garantizar la eficiente recuperación de la misma.

Fase II. Estudio y comprensión de los datos: Se obtuvieron 30 data sets en acuerdo con el área de rentas departamentales, los cuales serán empleados en el desarrollo del proyecto. Estos cuentan con datos de los contribuyentes, de los vehículos generadores del impuesto, información sobre seguridad social, información catastral y de georreferenciación, así como registros de los deudores y sus condiciones ante el ente regulador. Para la comprensión de la información recolectada se recibieron 8 diccionarios de datos que posibilitan el entendimiento de la data.

Fase III. Preparación de los datos: En esta fase se desarrollarán algunas actividades en pro de lograr una buena calidad de datos; para ello, primero se realiza la creación de un repositorio para contener el código fuente para el desarrollo del proyecto. Posterior a esto se realiza la preparación y limpieza de los datos seleccionados. Finalmente se integran las variables definidas como relevantes de cada set de datos, generando una tabla matriz con el total de atributos debidamente estandarizados a utilizar en el modelamiento.

Fase IV. Modelado: En esta etapa se implementará una técnica no supervisada de segmentación de ML, en busca de la construcción de un modelo que permita alcanzar los objetivos del proyecto, entregando las características principales de los contribuyentes del

impuesto vehicular y los pesos de estos atributos. Esta fase da respuesta al segundo objetivo de la investigación.

Fase V. Evaluación (obtención de resultados): Esta fase tiene relación con el tercer y cuarto objetivo de este trabajo, que se fundamenta en realizar la revisión de los clústeres obtenidos con el modelo construido, definir sus métricas y características principales para establecer categorías a cada grupo creado.

9. Resultados

FASE 1: ENTENDIMIENTO Y COMPRENSIÓN DEL NEGOCIO

➔ Obtención de las bases de datos

➔ Identificar las características de los datos por medio de un inventario de fuentes.

Comprende la recolección inicial de datos, con el objetivo de establecer un primer contacto con el problema, familiarizándose con ellos, identificar su calidad y establecer las relaciones más evidentes que permitan definir las primeras hipótesis. Se desarrollarán una serie de tareas en esta parte del proceso que son: la recolección de los datos iniciales, descripción y exploración de los datos y verificación de la calidad de estos.

En este punto se recibieron 30 fuentes de datos en múltiples formatos, entre los más usados se encuentra el TXT y el CSV, estas se describen en la tabla 1:

Tabla 2

Inventario de fuentes de Información

Nombre Archivo	**Categoría**	**Formato**	**Separador**	**Descripción**
PA_2018 20092022.txt	Partidas Abiertas	TXT	Punto y coma (;)	Contiene la información de partidas abiertas para el año 2018, partidas abiertas representa la cartera.
PA_2019 03042023.txt		TXT	Punto y coma (;)	Contiene la información de partidas abiertas para el año 2019, partidas abiertas representa la cartera.
PA_2020 03042023.txt		TXT	Punto y coma (;)	Contiene la información de partidas abiertas para el año 2020, partidas abiertas representa la cartera.
PA_2021 03042023.txt		TXT	Punto y coma (;)	Contiene la información de partidas abiertas para el año 2021, partidas abiertas representa la cartera.
PA_2022 03042023.txt		TXT	Punto y coma (;)	Contiene la información de partidas abiertas para el año 2022, partidas abiertas representa la cartera.
PA_2023 03042023.txt		TXT	Punto y coma (;)	Contiene la información de partidas abiertas para el año 2023, partidas abiertas representa la cartera.
BUT000_1.txt	Interlocutores Comerciales	TXT	Pipeline (|)	Contiene los datos de los interlocutores tales como el nombre, apellidos.
BUT0ID_1.txt		TXT	Pipeline (|)	Contiene los datos de identificación de los interlocutores comerciales
Catastro_Antioquia_2023.txt	Diccionario	TXT	Pipeline (|)	Contiene el inventario de los bienes inmuebles que posee persona natural o jurídica en Antioquia

Catastro_Medellin_202303.txt		TXT	Pipeline (|)	Contiene el inventario de los bienes inmuebles que posee persona natural o jurídica en Medellín.
Clase_documento.csv		CSV	Punto y coma (;)	Número de documento Contable PSCD
Clave_estadistica.csv		CSV	Punto y coma (;)	Indica si el documento que se contabilizo es real o estadístico es decir si afecta o no desde un inicio a la contabilidad
Denom_niv_reclama.csv		CSV	Punto y coma (;)	Es el momento en el que se encuentra el proceso de cobro de la deuda actualmente.
Denom_proced_reclama.csv		CSV	Punto y coma (;)	Clasificación de la causal del incumplimiento
Motivo_bloqueo_reclama.csv		CSV	Punto y coma (;)	Causal por la que se bloqueó o detuvo el proceso de reclamación.
Operacion_parcial.csv		CSV	Punto y coma (;)	Es el subconcepto o detalle por ejemplo impuesto del 80% (Gobernación) y el impuesto del 20% (Municipio)
Operacion_principal.csv		CSV	Punto y coma (;)	Clasificación del concepto de operación. Para el presente ejercicio será 4002 - renta de vehículos
Régimen contributivo.txt	Régimen	TXT	Coma (,)	Contiene la información de los aportes realizados por las cotizaciones de empleados y empleadores para sal
Régimen Subsidiado.txt		TXT	Coma (,)	Contiene la información del sistema subsidiado de salud de la población vulnerable
Oc_1_2022.txt	Objeto contrato	TXT	Tabulador(\t)	Contiene la información asociada a los contratos (placas) y las diferentes tipologías entre ellos.
Oc_2_2022.txt		TXT	Tabulador(\t)	Contiene la información asociada a los contratos (placas) y las diferentes tipologías entre ellos.
Oc_3_2022.txt		TXT	Tabulador(\t)	Contiene la información asociada a los contratos (placas) y las diferentes tipologías entre ellos.
Oc_4_2022.txt		TXT	Tabulador(\t)	Contiene la información asociada a los contratos (placas) y las diferentes tipologías entre ellos.
Oc_5_2022.txt		TXT	Tabulador(\t)	Contiene la información asociada a los contratos (placas) y las diferentes tipologías entre ellos.
Oc_6_2022.txt		TXT	Tabulador(\t)	Contiene la información asociada a los contratos (placas) y las diferentes tipologías entre ellos.
Expli_parti_abiert.xlsx	Meta data	XLSX	No requiere	Contiene el diccionario de datos general de partidas abiertas, allí se describen todos los tipos de variables según sus códigos.
Divipola.parquet		PARQUET	Tabular	División Política Administrativa (Divipola)
Regiones Antioquia.xlsx		XLSX	No requiere	Contiene la división política del departamento de Antioquia en regiones, según sus municipios.
Regiones Colombia.xlsx		XLSX	No requiere	Contiene la división regional de Colombia, según sus departamentos.

Fuente: Elaboración propia

Para el almacenamiento de la data se define la figura de dirty_lake que representa la laguna de datos origen o datos sucios, para ello se crea una carpeta en Drive compartida para todos los miembros en la que se efectúa la carga del inventario de fuentes de información y así poder ser descargada al entorno local de desarrollo.

La Figura 5 muestra el estado del data set (Partidas abiertas) al momento de su recepción.

Figura 5

Visualización Partidas abiertas

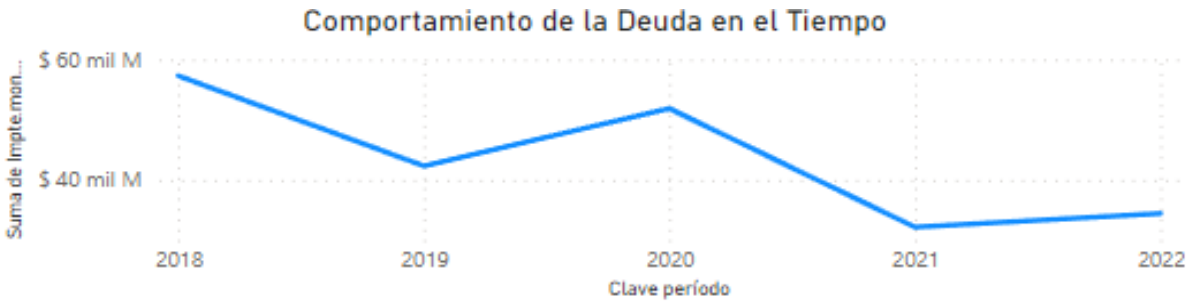

```
N° documento;Pos.repetición;Posición;Pos.parcial;Moneda;Car.determ.cta.;Mot.bloq.recl;Proced.reclam;Nivel reclam;Tp.doc;Clave estadíst,
031522804068;000;0001;000;COP;02;;;00;;G;4002;0150;171.00 ;;000AAM;020001283810;;1384900044;20190402;;2018;SI;000000307919601;1001546
006019363931;000;0001;000;COP;02;;;00;;G;4002;0260;24.00 ;;000AAM;020001283810;;1311030005;20190402;;2018;IN;000000307919601;10015468
006019363930;000;0001;000;COP;02;;;00;;G;4002;0250;96.00 ;;000AAM;020001283810;;1311030005;20190402;;2018;IN;000000307919601;10015468
003510893215;000;0002;000;COP;02;;13;08;;G;4002;1160;136.00 ;001;000AAM;020001283810;;1305335004;20190402;;2018;FR;000000307919601;10
003510893215;000;0001;000;COP;02;;13;08;;G;4002;1150;544.00 ;001;000AAM;020001283810;;1305335003;20190402;;2018;FR;000000307919601;10
031006844848;000;0002;000;COP;02;;;00;;G;4002;0360;38.00 ;;000AAM;020001283810;;1384900043;20220831;;2018;CS;000000307919601;10015468
031004349301;000;0002;000;COP;02;;;00;;G;4002;0360;342.00 ;;000AAM;020001283810;;1384900043;20190430;;2018;CS;000000307919601;1001546
031004349301;000;0001;000;COP;02;;;00;;G;4002;0350;1368.00 ;;000AAM;020001283810;;1311040001;20190430;;2018;CS;000000307919601;100154
031006844848;000;0001;000;COP;02;;;00;;G;4002;0350;152.00 ;;000AAM;020001283810;;1311020013;20220831;;2018;CS;000000307919601;1001546
031524117009;000;0001;000;COP;02;;;00;;G;4002;0150;177.50 ;;000ACP;020002572773;;1384900044;20200601;;2018;SI;000000308919268;100301
```


Fuente: Elaboración propia

Al obtener la información útil para nuestro ejercicio de cada conjunto de datos, ya es posible realizar un primer acercamiento a los contenidos propios de cada uno de ellos. Para el caso de partidas abiertas, el primero obtenido, ya podemos clasificar en función de las operaciones parciales propias del impuesto e identificar la cantidad de contratos (placas) que figuran allí, así como los importes de deuda por cada año y los interlocutores comerciales (contribuyentes) que presentan mayor cantidad de deuda y contratos asociados, esto lo podemos apreciar en las Figuras 6 y 7:

Figura 6

Gráfica de comportamiento de deuda (PA)



Nota: Tomado de PowerBi, Elaboración propia

Figura 7

Resumen de deuda por todos los periodos

Nota: Tomado de PowerBi, Elaboración propia

FASE 2: ESTUDIO Y COMPRENSIÓN DE LOS DATOS

Debido a la alta cantidad de fuentes vinculadas al proceso y a la variabilidad entre sus tipos de archivos se hizo necesario darle un orden sistemático a cada script desarrollado para el consumo de estas, para ello crea un repositorio de código fuente en GitLab con el objetivo de mantenerlos sincronizados y de esta manera brindar un entorno centralizado para gestionarlos y organizarlos de manera eficiente, permitiendo realizar un seguimiento de los cambios a lo largo del tiempo, colaborar con otros miembros del equipo y mantener un historial completo de revisiones y versiones de los scripts.

El entorno de desarrollo es creado bajo la siguiente estructura (ver Figura 8):

***Carpeta Raíz*:** Contiene todas las demás carpetas del proyecto.

documentation: Documentación requerida para compresión de la información.

task: Comprende todas las tareas realizadas sobre sobre la información, en su interior se cuenta con una estructura de subcarpetas que hacen alusión al nombre de la fuente que estandarizan.

useful: Compilado de librerías de elaboración propia para el procesamiento, graficado de datos y mapeo de rutas.

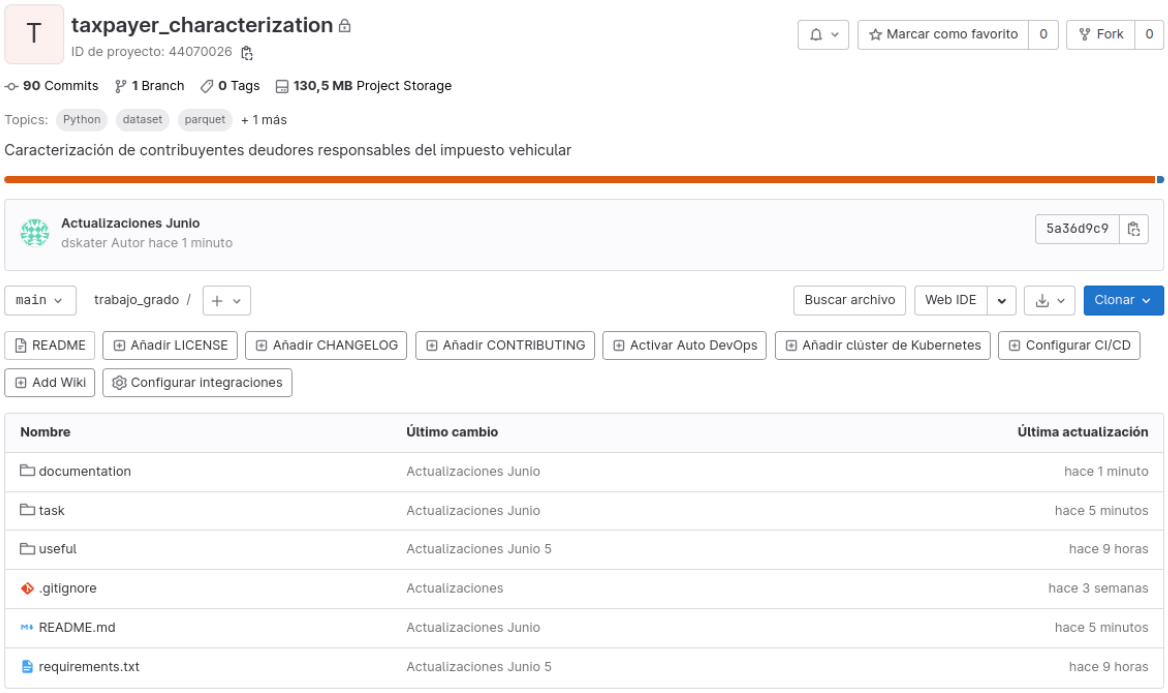
. ***gitignore:*** Archivo que contiene el mapeo de los archivos que no deben ser cargados al repositorio.

Readme.md: Documentación con respecto al clonado del repositorio.

requirements.txt: Librerías necesarias para el procesamiento de datos.

Figura 8

Entorno GitLab



Fuente: Elaboración propia

Para facilitar el consumo de los Data sets, se desarrolla un archivo llamado settings.ini el cual se encuentra mapeado en el ".gitignore" y se encarga del mapeo de las rutas en cada entorno local de desarrollo, luego este es consumido por una función llamada "environment.py" que se encuentra en el módulo "useful", que permite instanciar dichas rutas y crear un manejo estandarizado de la siguiente manera:

- DIRTY: Representa la ruta del Data Lake de las fuentes origen.
- CLEAN: Representa la ruta del Data Lake de las fuentes transformadas.
- META: Representa la ruta de la Meta data, tales como diccionarios, Divipola.

- CACHE: Representa la ruta para el almacenamiento de archivos temporales.

Figura 9

Settings.ini

```
[LOCAL]
DEBUG=TRUE
RENTASANT_VENV=C:\Users\jhona\Desktop\trabajo_grado\dirty_lake
RENTASANT_CACHE_FOLDER=C:\Users\jhona\Desktop\trabajo_grado\dirty_lake\cache
RENTASANT_CLEAN_FOLDER=C:\Users\jhona\Desktop\trabajo_grado\clean_lake
META=C:\Users\jhona\Desktop\trabajo_grado\useful\metadata
```

Fuente: Elaboración propia

FASE 3: PREPARACIÓN DE LOS DATOS

➔ Realizar la limpieza de los datos

➔ Realizar la integración de los datos relevantes definidos en los sets de datos obtenidos

➔ Realizar la unión de los conjuntos de datos después de su preprocesamiento, eliminación de datos nulos, atípicos e imputación de datos.

Librerías instaladas en el sistema: Se lleva a cabo instalación de la librería virtualenv, esta proporciona un espacio aislado donde se pueden instalar y gestionar las dependencias específicas de un proyecto de Python, lo que evita conflictos con las librerías instaladas en el sistema global y permite mantener un control preciso sobre las versiones de las dependencias utilizadas en el proyecto

Librerías instaladas en el repositorio: Se hace instalación mediante el archivo *requirements.txt* el conjunto de librerías necesarias con su respectivo versionamiento, para esto se crea un entorno virtual dentro del repositorio local denominado venv usando la librería virtualenv y usando Python en 3.9 (Ver Figura 10)

Figura 10

Requirements.txt

```
pandas==1.5.3
psutil==5.9.5
numpy==1.23.5
openpyxl==3.1.2
fastparquet==2023.2.0
pyarrow==11.0.0
jupyter
scikit-learn==1.2.2
matplotlib==3.7.1
seaborn==0.12.2
tqdm==4.65.0
ydata-profiling==4.2.0
```

Fuente: Elaboración propia

Posterior a la instalación de los módulos necesarios para la carga, procesamiento y graficado de los datos se desarrolla un set de librerías propias en el apartado de “useful” del repositorio de código fuente, con el objetivo de realizar una estandarización óptima de la data evitando la repetición constante de líneas de código, permitiendo dar un más orden y mejor comprensión del proceso de ETL. (Ver figura 11)

A continuación, se hace referencia a las más importantes:

decoder.py: Contiene funciones que permiten la decodificación o estandarización de distintos tipos de datos algunas de las principales son:

- ➔ *decode_scalar*: Decodifica el valor ingresado al estándar float.

- ➔ *decode_year*: Decodifica el valor ingresado al estándar de año.
- ➔ *decode_null*: Decodifica el valor ingresado al valor nulo (vacíos).
- ➔ *clean_label*: Su función es limpiar las columnas de los Dataframe eliminando caracteres adicionales y reemplazando espacio por un guion bajo.

graph.py: Contiene funciones que permiten la realización de diferentes gráficas, mediante el uso de librerías como Seaborn y Matplotlib entre las más relevantes están:

- ➔ *numeric_graphs_v2*: Permite generar gráficos sobre las variables de tipo numérico
- ➔ correlation: Función que permite generar la correlación de spearman y graficarla
- ➔ *silhouette_analysis*: Función que permite identificar el número óptimo de clúster donde se maximiza la media del coeficiente de la silueta de todas las observaciones.
- ➔ *elbow*: Función que permite identificar el número óptimo de clústeres mediante la evolución de la varianza entre clúster.
- ➔ *visualization3d*: Función que permite la visualización del clúster en 3 dimensiones, mediante el uso de Análisis de componentes principales.
- ➔ *visualization2d*: Función que permite la visualización del clúster en 2 dimensiones, mediante el uso de Análisis de componentes principales.

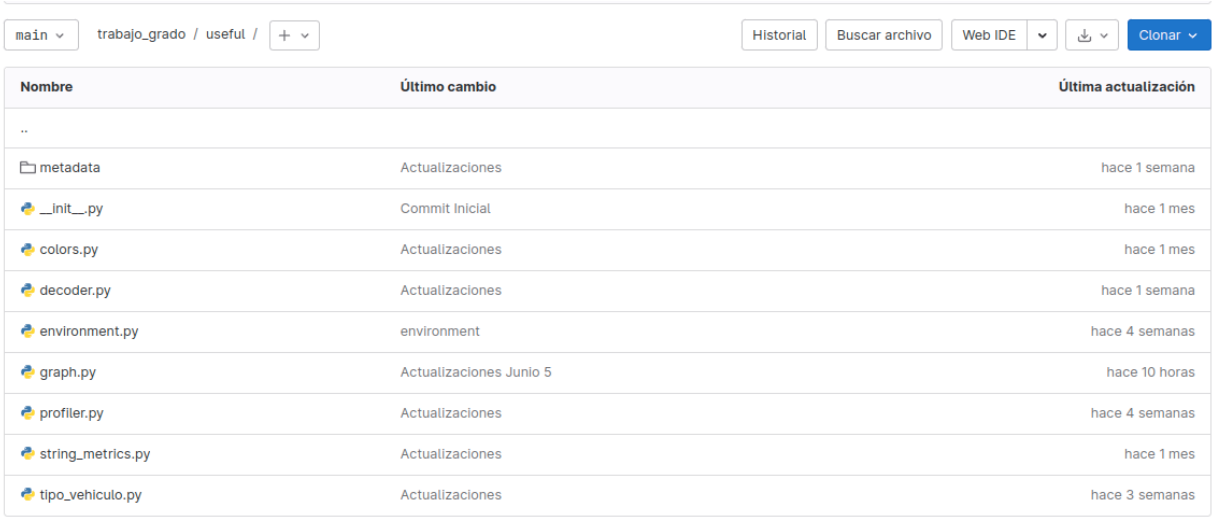
profile_dataframe.py: Esta función toma como entrada un DataFrame y genera una muestra aleatoria de n (si n no está definida toma 10 por defecto) registros para cada variable en el DataFrame. Si una variable está toda en NaN, se indica en la salida.

string_metrics.py: Función definida para el tratamiento de variables tipo string, entre algunas de sus funciones se encuentran:

- ➔ *convert_to_float*: eliminar caracteres de puntuación excepto la coma y el punto y convertir a float.
- ➔ unicode: Su función es normalizar una palabra, es decir que aquellas letras que tengan un carácter adicional, se les eliminará dicho carácter, por ejemplo, las tildes, diéresis o virgulilla.
- ➔ *without_spaces*: eliminar todos los espacios y saltos de línea.
- ➔ *without_multiple_spaces*: Su función es convertir múltiples espacios entre palabras en uno solo.
- ➔ *title_name*: Convierte en title un texto sin tener en cuenta los artículos o conectores

Figura 11

Librerías creadas



Fuente: Tomado de GitLab, Elaboración propia

Debido a la diferencia entre formatos y su composición cada proceso se hace de manera diferente en cuanto a la carga de los datos, sin embargo, el procesamiento de los datos en la mayoría de los escenarios es el mismo, acá es donde juega un papel importante las librerías desarrolladas.

A continuación, se especifica las generalidades y el tratamiento de datos que se efectúa sobre cada fuente suministrada:

Todas las fuentes:

- Se usa librería os y environment para mapeo de rutas.
- Las columnas son renombradas usando clean_label.
- Todos los sets con cargados con sus variables en tipo string para evitar que pandas infiera el tipo de dato, este se le asigna una vez se analiza y entiende la data.
- Los archivos parquet son generados con los paramentos engine=" pyarrow' y compression = "GZIP".

Para cada fuente se generan dos parquet, uno con el proceso de apilado y variables estandarizadas y el otro con la selección de variables escogidas para el proceso de modelación.

Partidas Abiertas: Usando las librerías os y environment se define la ruta donde se encuentras los archivos del set específico, mediante un ciclo for se itera cada archivo haciendo una carga en memoria de este y almacenándolos en una lista para concatenarlos usando la función concat del paquete pandas para crear un nuevo set de datos con la información de todos los archivos. (ver imagen 12)

Figura 12

Carga partidas abiertas

Construcción del Path

```
PATH = os.path.join(DIRTY, 'partidas_abiertas')
```

Apilado de archivos

```
files = os.listdir(PATH)
dfs = []
for file in sorted(files):
    if file.startswith('PA_'):
        print(f'{BColors.OKGREEN}Leyendo {file}...{BColors.ENDC}')
        df = pd.read_csv(os.path.join(PATH, file), sep=';',  dtype='str')
        df['record_source'] = file
        dfs.append(df)
```

```
Leyendo PA_2018 20092022.txt...
Leyendo PA_2019 03042023.txt...
Leyendo PA_2020 03042023.txt...
Leyendo PA_2021 03042023.txt...
Leyendo PA_2022 03042023.txt...
Leyendo PA_2023 03042023.txt...
```

volver

```
df = pd.concat(dfs)
```

Fuente: Tomado de Pycharm, Elaboración propia

Una vez terminado el proceso de apilamiento se cuenta con un Dataframe constituido por 38 columnas y 5.876.852 registros como lo muestra la Figura 13.

Figura 13

Carga partidas abiertas

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5876852 entries, 0 to 5876851
Data columns (total 38 columns):
 #   Column           Dtype
---  ------           -----
 0   nº_documento     object
 1   pos_repeticion   object
 2   posicion         object
 3   pos_parcial      object
 4   moneda           object
 5   cardetermcta     object
 6   mot_bloq_recl    object
 7   proced_reclam    object
 8   nivel_reclam     object
 9   tp_doc           object
 10  clave_estadist   object
 11  op_principal     object
 12  op_parcial       object
 13  impte_mon_local  object
 14  indicador_recl   object
 15  contrato         object
 16  cuenta_contrato  object
 17  doc_suplente     object
 18  cuenta_de_mayor  object
 19  fecha_contab     object
 20  causa_bloq_pago  object
 21  clave_periodo    object
 22  clase_documento  object
 23  referencia       object
 24  interlcomerc     object
 25  fe_ejec          object
 26  status_compens   object
 27  clv_reconcil     object
 28  transf_total     object
 29  cerrado          object
 30  cerrados_por     object
 31  cerrado_el       object
 32  tipo_novedad     object
 33  fecha_novedad    object
 34  ap_siapa         object
 35  proced_reclam2   object
 36  nivel_reclam2    object
 37  record_source    object
dtypes: object(38)
memory usage: 1.7+ GB
```

Fuente: Tomado de GitLab, Elaboración propia

Se analizan nulos por cada columna para identificar el tratamiento que se le debe dar de acuerdo a su tipo de dato, se normalizan los datos usando los diccionarios suministrados y se lleva cada variable a su tipo de dato específico; se realiza un selección de las variables que se consideraron necesarias quedando con un total de 13, sin embargo se fueron eliminando a medida que se ahondaba en el análisis; se detecta que en la data existe información que no corresponde a la Gobernación de Antioquia y se efectúa un filtrado de los datos usando la variable op_parcial de esta manera se obtiene la información de los deudores del de impuesto vehicular del Departamento separada por conceptos específicos; posteriormente se efectúa otro filtro para retirar el año 2023 pues este no se encuentra en estado vencido por lo que se usa la variable clave_periodo para tal propósito, al Dataframe resultante se le realiza un agrupamiento por interlocutores, finalmente obteniendo de estos la cantidad de periodos adeudados, y la deuda por los conceptos de Impuesto, interés y sanción, el resultado es guardado en un archivo en formato parquet en la laguna de datos clean_lake(ver Figura 14)

Figura 14

Partidas abiertas

	interlcomerc	cant_periodos	impuesto	interes	sancion
0	1000000012	2	185600.0	64800.0	304000.0
1	1000000018	5	373600.0	182400.0	766400.0
2	1000000040	3	127200.0	7200.0	145600.0
3	1000000051	2	133600.0	0.0	152000.0
4	1000000070	2	292000.0	44800.0	297600.0
...	...	...	...	...	...
323152	9000011711	1	74400.0	0.0	0.0
323153	9000011712	3	1251200.0	523200.0	944800.0
323154	9000011730	2	199200.0	54400.0	304000.0
323155	9000012246	1	9556800.0	0.0	0.0
323156	9000012774	1	74400.0	0.0	0.0

323157 rows × 5 columns

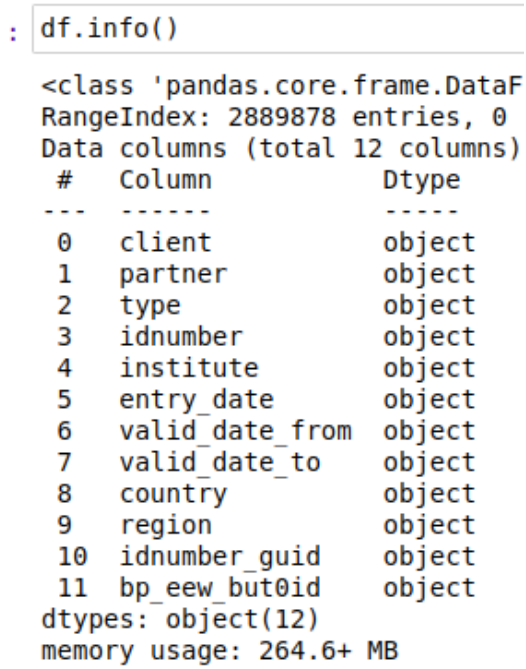
Fuente: Tomado de GitLab, Elaboración propia

Interlocutores Comerciales

BUT01D: En el tratamiento de esta fuente se detecta que la información cuenta con una anomalía que hace que pandas efectúe un desplazamiento de los datos, para solucionar esto se utilizar el parámetro on_bad_lines='skip' que permite que la librería ignore las líneas con problemas de desplazamiento, también se define un log para almacenar la posición de los registros con problemas, una vez cargado se cuenta con Dataframe de 12 columnas y 2.889.878 registros como se observa en la Figura 15:

Figura 15

BUT01ID



```
: df.info()

<class 'pandas.core.frame.DataF
RangeIndex: 2889878 entries, 0
Data columns (total 12 columns)
 #   Column           Dtype
---  ------           -----
 0   client           object
 1   partner          object
 2   type             object
 3   idnumber         object
 4   institute        object
 5   entry_date       object
 6   valid_date_from  object
 7   valid_date_to    object
 8   country          object
 9   region           object
 10  idnumber_guid    object
 11  bp_eew_but0id    object
dtypes: object(12)
memory usage: 264.6+ MB
```

Fuente: Tomado de GitLab, Elaboración propia

En el análisis de registros nulos nos encontramos que gran cantidad de la información no es capturada por la entidad y por esta razón todos sus registros tienen esta característica, por tal motivo, se procede con el retiro de estos; para la estandarización se usan funciones propias como without_spaces; se crea la variable tipo_documento usando el decodificador de tipos de

documentos, se renombra la variable idnumber a numero_documento. Como resultado final queda una tabla con 3 columnas y la misma cantidad de registros inicial la cual es guarda en formato parquet en Clean Lake (ver Figura 16); el objetivo de esta tabla es la integración de nuevas fuentes mediante el uso de una llave creada a través del tipo y número de documento

Figura 16

BUT01ID PARQUET

```
inter
```

	partner	numero_documento	tipo_documento
0	1000001167		Cédula de ciudadanía
1	1000001168		Cédula de ciudadanía
2	1000001169		Cédula de ciudadanía
3	1000001170		Cédula de ciudadanía
4	1000001171		Cédula de ciudadanía
...	...	...	...
2889873	1003842519		Cédula de ciudadanía
2889874	1003842532		Cédula de ciudadanía
2889875	1003842537		Cédula de ciudadanía
2889876	1003842574		Número de identificación tributaria
2889877	1003388761		Cédula de ciudadanía

2889878 rows × 3 columns

```
inter.to_parquet(os.path.join(CLEAN, 'but0id_V2.parquet'),
                 index=False, engine='pyarrow',
                 compression='GZIP')
```

Fuente: Tomado de GitLab*,* Elaboración propia

BUT000: Al igual que la fuente anterior presenta problemas de desplazamiento que fueron solucionadas de la misma manera, este set cuenta con 91 columnas y 2.978.070 de registros sin embargo tiene una gran cantidad de variables en estado nulo (ver Figura 17), por ello efectúa limpieza de datos retirando dichas columnas; para este set de datos no fueron suministrados diccionarios de información y no fue posible la desnormalización, por este motivo se determina que se solo es viable usar el sexo, este viene representado en dos columnas con una marcación de una x por ocurrencia, se efectúa un procedimiento de melt y estas dos columnas se convierten

en una sola con las categorías masculino y femenino, finalmente se guarda un archivo en formato parquet en Data Clean.

Figura 17

Nulos

Nulos por columna

```
display(df.isnull().sum().sort_values(ascending=False).head(70))
```

```
kbankl           2978070
kbanks           2978070
par_rel          2978070
bp_eew_dummy     2978070
is_org_centre    2978070
                  ...
mc_name2           92749
bpkind             12400
mc_name1            1201
valid_from            15
mem_house             15
Length: 70, dtype: int64
```

Fuente: Tomado de GitLab, Elaboración propia

Finalmente se efectúa una unión de ambos archivos parquet usando la variable partner y el resultado se guarda en formato parquet en la laguna de datos Clean.

Régimen Contributivo - Régimen Subsidiado

Tratamiento para ambos conjuntos: Cuando se intenta cargar la información, se detecta que esta no posee encabezados, para solucionar este inconveniente se usa el parámetro names para darle unos nombres a cada columna del set de datos, los nombres usados fueron los siguientes: tipo_documento, numero_documento, primer_nombre, segundo_nombre, primer_apellido, segundo_apellido, cod_municipio, fecha_nacimiento, sexo

Figura 18

Régimen contributivo

Regimen Contributivo

```
In [3]: FOLDER = 'regimenes'
        PATH = os.path.join(DIRTY, FOLDER, 'Regimen contributivo.txt')
        df = pd.read_csv(PATH, sep=",", encoding='latin-1', dtype='str', header=None,
                         names=['tipo_documento','numero_documento', 'primer_nombre',
                                'segundo_nombre', 'primer_apelllido',
                                'segundo_apellido', 'cod_municipio', 'fecha_nacimiento', 'sexo',
                                'no_identificada'])
        df
```

Out[3]:

	tipo_documento	numero_documento	primer_nombre	segundo_nombre	primer_apellido	segundo_apellido	cod_municipio	fecha_nacimiento	sexo	no_id
0	CC		IVAN	ANDRES	MESTRA	RAMOS	001	12/9/1996 00:00:00	M	
1	CC		JAVIER	ENRIQUE	ARROYO	FLOREZ	001	11/3/1998 00:00:00	M	
2	CC		GIULIANA	SUSANA	OSORIO	HOYOS	001	9/1/1999 00:00:00	F	
3	CC		IVAN	DAVID	MENDOZA	CERVANTES	001	5/1/1999 00:00:00	M	
4	CC		LUISA	FERNANDA	QUINTERO	VELEZ	001	6/5/2003 00:00:00	F	
...	...		...	...	...	...	...	...	...	
1874180	TI		PAULINA	NaN	CEBALLOS	GOMEZ	615	3/4/2009 00:00:00	F	
1874181	TI		SARA	NaN	CASTAÑO	ZAPATA	615	5/10/2008 00:00:00	F	
1874182	TI		LAURA	MARIANA	BETANCUR	CASTRILLON	615	9/10/2008 00:00:00	F	
1874183	TI		JUAN	CAMILO	RIOS	RIOS	615	28/10/2008 00:00:00	M	

Fuente: Tomado de GitLab*,* Elaboración propia

Se efectúa cálculo de la edad a través de la fecha de nacimiento, se estandariza la variable sexo y se agrega 05 al código de municipio; se crear la variable tipo_regimen de cada régimen se genera un archivo parquet; se efectúa la unión de ambos parquet de datos y se retiran registro duplicados, mediante el código del municipio se efectúa un merge con la metada de las regiones de Antioquia y se genera la columna provincia, finalmente se genera un parquet en la laguna de datos Clean.

Catastro Medellín - Catastro Antioquia

Tratamiento para ambos conjuntos: Se cargan fuentes de datos de manera independiente, catastro Antioquia cuenta con 16 columnas y 1.178.248 registros; por su parte catastro Medellín cuenta con 21 columnas y 1.917.638. Para ambos sets se retiran registros duplicados se seleccionan las siguientes variables:

“numero_documento”, “matricula”, “avaluo_total”, “derecho”. Se estandarizan datos usando funciones propias, luego se crea la variable valor para representar el valor real del predio que le corresponde al contribuyente, esta se calcula haciendo uso del derecho y el avalúo total, el “derecho” representa el porcentaje que le corresponde del predio y el “avaluo_total” hace referencia a la suma del valor del inmueble y el valor del terreno; finalmente, se efectúa un agrupamiento por número de documento en el que se cuenta el número de predios existentes para este y se suma la variable “valor”. Cada archivo es guardado por separado en un parquet y posteriormente son unidos en una única fuente que también es almacenada en la laguna de datos Clean en el formato antes mencionado. (ver Figura 19)

Figura 19

Unión CAT.MED- CAT.ANT

```
df.info()
```

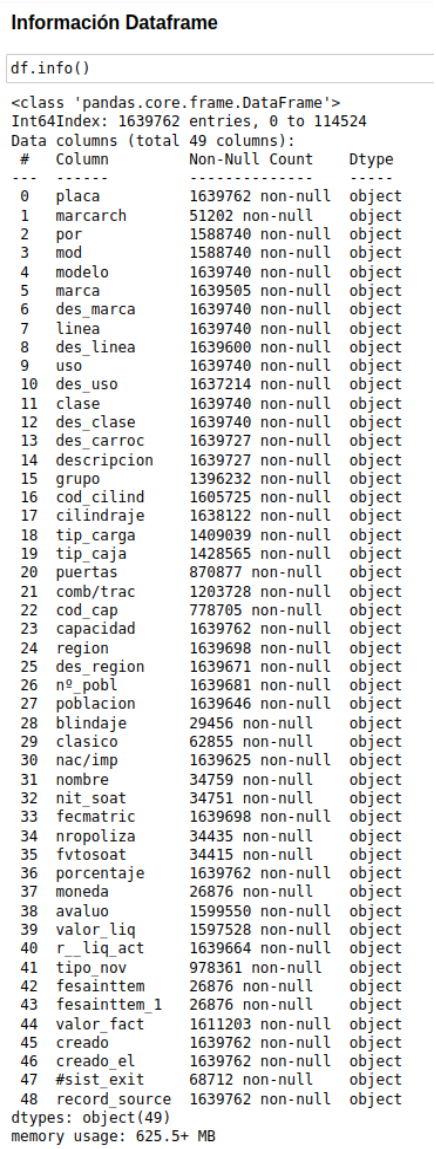
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1445626 entries, 0 to 722812
Data columns (total 3 columns):
 #   Column            Non-Null Count    Dtype
---  ------            --------------    -----
 0   numero_documento  1445626 non-null  object
 1   valor             1445626 non-null  int64
 2   cantidad_predio   1445626 non-null  int64
dtypes: int64(2), object(1)
memory usage: 44.1+ MB
```

Fuente: Tomado de GitLab, Elaboración propia

Objeto contrato: Debido a que esta fuente de información cuenta con 6 archivos, se lleva a cabo un proceso de apilamiento de estos siguiendo la lógica de las Partidas abiertas. Una vez finalizado el proceso, se eliminan los registros nulos y se obtiene un Dataframe compuesto por 49 columnas y 1.639.762 registros. (ver Figura 20)

Figura 20

Información Dataframe



Información Dataframe

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1639762 entries, 0 to 114524
Data columns (total 49 columns):
 #   Column          Non-Null Count    Dtype
---  ------          --------------    -----
 0   placa           1639762 non-null  object
 1   marcarch        51202 non-null    object
 2   por             1588740 non-null  object
 3   mod             1588740 non-null  object
 4   modelo          1639740 non-null  object
 5   marca           1639505 non-null  object
 6   des_marca       1639740 non-null  object
 7   linea           1639740 non-null  object
 8   des_linea       1639600 non-null  object
 9   uso             1639740 non-null  object
 10  des_uso         1637214 non-null  object
 11  clase           1639740 non-null  object
 12  des_clase       1639740 non-null  object
 13  des_carroc      1639727 non-null  object
 14  descripcion     1639727 non-null  object
 15  grupo           1396232 non-null  object
 16  cod_cilind      1605725 non-null  object
 17  cilindraje      1638122 non-null  object
 18  tip_carga       1409039 non-null  object
 19  tip_caja        1428565 non-null  object
 20  puertas         870877 non-null   object
 21  comb/trac       1203728 non-null  object
 22  cod_cap         778705 non-null   object
 23  capacidad       1639762 non-null  object
 24  region          1639698 non-null  object
 25  des_region      1639671 non-null  object
 26  nº_pobl         1639681 non-null  object
 27  poblacion       1639646 non-null  object
 28  blindaje        29456 non-null    object
 29  clasico         62855 non-null    object
 30  nac/imp         1639625 non-null  object
 31  nombre          34759 non-null    object
 32  nit_soat        34751 non-null    object
 33  fecmatric       1639698 non-null  object
 34  nropoliza       34435 non-null    object
 35  fvtosoat        34415 non-null    object
 36  porcentaje      1639762 non-null  object
 37  moneda          26876 non-null    object
 38  avaluo          1599550 non-null  object
 39  valor_liq       1597528 non-null  object
 40  r__liq_act      1639664 non-null  object
 41  tipo_nov        978361 non-null   object
 42  fesainttem      26876 non-null    object
 43  fesainttem_1    26876 non-null    object
 44  valor_fact      1611203 non-null  object
 45  creado          1639762 non-null  object
 46  creado_el       1639762 non-null  object
 47  #sist_exit      68712 non-null    object
 48  record_source   1639762 non-null  object
dtypes: object(49)
memory usage: 625.5+ MB
```

Fuente: Tomado de GitLab, Elaboración propia

Dado que no se dispone de un diccionario de datos, se realiza un análisis de los mismos utilizando la función profile_dataframe, solicitando 13 registros por columna. Durante dicho análisis, se detecta que los encabezados de las columnas se repiten cada cierta cantidad de registros. Para solucionar este inconveniente, se utiliza el nombre de una de las variables y se

aplica la función isin del paquete pandas para eliminar dichos registros. Además, se estandarizan las variables utilizando las librerías desarrolladas. (ver Figura 21)

Figura 21

Eliminación encabezados

```
ant = len(df)
df = df[~df.fvtosoat.isin(['FVtoSOAT'])]
desp = len(df)
print(f'Se elimnaron {ant-desp} resgistros con problema de encabezado')
```

```
Se elimnaron 26876 resgistros con problema de encabezado
```

Fuente: Tomado de GitLab, Elaboración propia

Se hace selección de las variables “placa”, “modelo”, “des_uso”, “des_clase”, “reg_col”, “blindaje”, “clasico”, “nac/imp”, “avaluo” como las variables que se usarán de este set de datos, se efectúa un renombre a las columnas se definen los tipos de datos de cada columna, se crear una variable categórica cat_modelo con la variable modelo usando las siguientes asignaciones: Antes de 1950, 1950-1964, 1965-1979, 1980-1994, 1995-2009, 2010-2024 y sin_modelo, este proceso se lleva a cabo utilizando la función cut de la librería pandas, finalmente el Dataframe resultante es guardado en formato parquet en la laguna de datos clean_lake.

Uniones: En fase del proceso ya se tiene completo conocimiento de las fuentes de las variables y de sus identificadores únicos, siendo posible la unión de algunos sets, ese proceso se lleva a cabo usando la función merge perteneciente a pandas, a continuación, se detalla su orden:

- interlocutor-catastro, la unión se hace mediante el número de documento, al set resultante se le elimina el número de documento, dejando como clave la variable partner, finalmente es guardado en formato parquet en la laguna “clean_lake”.

- interlocutor-regimen, la unión se lleva a cabo usando el número de documento del contribuyente, posterior a la unión se elimina esta variable dejando disponible la variable “partner”, al finalizar el proceso este es guardado en formato parquet en la laguna de datos limpios.
- partidas_abiertas-objeto_contrato_interlocutor, la unión se lleva a cabo mediante la variable “contrato”, posteriormente se efectúa un filtro por tipo de documento “Número único de Identificación Tributaria (NIT), debido a que no se cuenta con información para las empresas, se deja disponible la variable “partner” y se elimina “tipo_documento” y “numero_documento”; posteriormente el conjunto resultante es guardado en la laguna de datos en formato parquet.

Union_all: Finalmente todas las fuentes procesadas son unidas a través la variable “partner”, generando un Dataframe compuesto por 183 904 registros y 49 columnas, sobre este se lleva a cabo el siguiente proceso:

Imputación de datos:

La imputación de datos es un proceso crítico en el análisis de datos cuando se encuentran valores faltantes en variables importantes. En este caso, se imputaron los datos faltantes en las variables "tipo_regimen", "sexo", "provincia" y "edad".

Para las variables "tipo_regimen", "sexo" y "provincia", se utilizó la técnica de imputación por moda, es decir, se reemplazaron los valores faltantes con el valor más frecuente en cada una de estas variables. Esto permite mantener la consistencia de los datos y preservar la distribución original de las categorías.

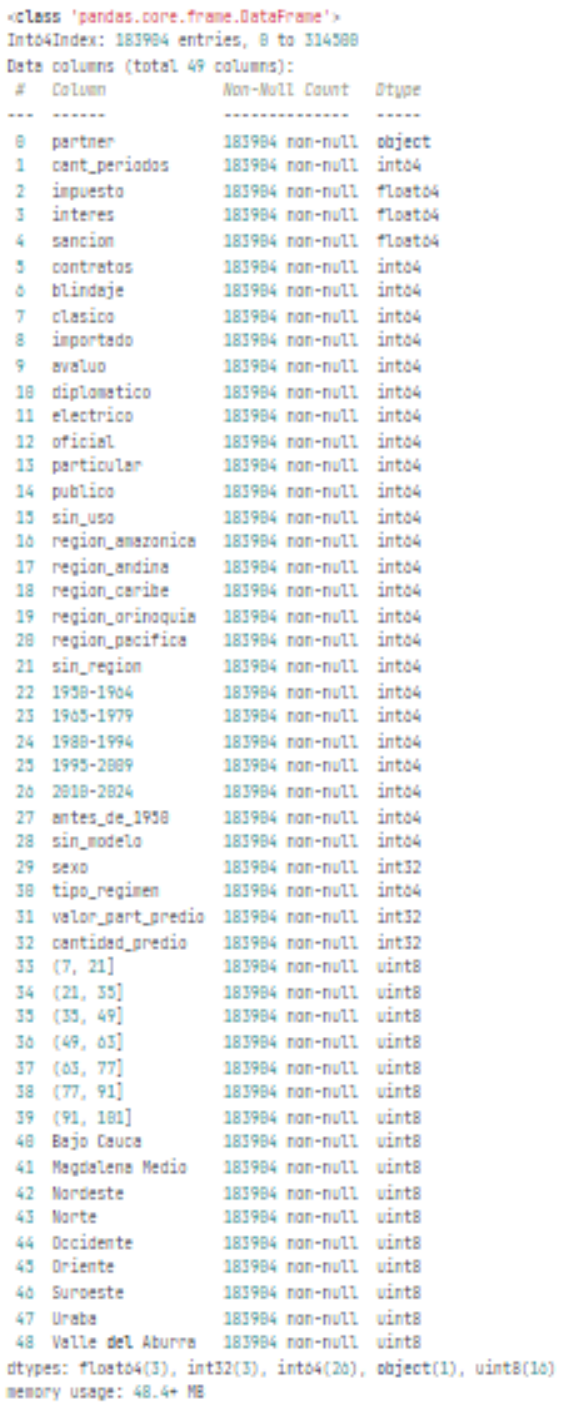
En cuanto a la variable "edad", se utilizó la técnica de imputación por promedio. Los valores faltantes fueron reemplazados por el promedio de los registros existentes en esta variable. Esto proporciona una estimación razonable de los valores faltantes y ayuda a mantener la coherencia de los datos.

Obtención de variables dummy: Después de ejecutar la limpieza de datos descrita anteriormente, se convirtieron las variables edad y modelo a variables “dummy”, con el fin de representar las categorías en valores sobre un rango continuo (variables cuantitativas). Como resultado, cada una de las categorías de las variables cualitativas del siguiente dataframe, se convierten en una variable independiente aumentando la dimensionalidad de este.

La figura 22 representa el resultado luego de la aplicación del procedimiento antes descrito.

Figura 22

Variables union_all



```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 183984 entries, 0 to 314588
Data columns (total 49 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   partner            183984 non-null  object
 1   cant_periodos      183984 non-null  int64
 2   impuesto           183984 non-null  float64
 3   interes            183984 non-null  float64
 4   sancion            183984 non-null  float64
 5   contratos          183984 non-null  int64
 6   blindaje           183984 non-null  int64
 7   clasico            183984 non-null  int64
 8   importado          183984 non-null  int64
 9   avaluo             183984 non-null  int64
 10  diplomatico        183984 non-null  int64
 11  electrico          183984 non-null  int64
 12  oficial            183984 non-null  int64
 13  particular         183984 non-null  int64
 14  publico            183984 non-null  int64
 15  sin_uso            183984 non-null  int64
 16  region_amazonica   183984 non-null  int64
 17  region_andina      183984 non-null  int64
 18  region_caribe      183984 non-null  int64
 19  region_orinoquia   183984 non-null  int64
 20  region_pacifica    183984 non-null  int64
 21  sin_region         183984 non-null  int64
 22  1958-1964          183984 non-null  int64
 23  1965-1979          183984 non-null  int64
 24  1980-1994          183984 non-null  int64
 25  1995-2009          183984 non-null  int64
 26  2010-2024          183984 non-null  int64
 27  antes_de_1958      183984 non-null  int64
 28  sin_modelo         183984 non-null  int64
 29  sexo               183984 non-null  int32
 30  tipo_regimen       183984 non-null  int64
 31  valor_part_predio  183984 non-null  int32
 32  cantidad_predio    183984 non-null  int32
 33  (7, 21]            183984 non-null  uint8
 34  (21, 35]           183984 non-null  uint8
 35  (35, 49]           183984 non-null  uint8
 36  (49, 63]           183984 non-null  uint8
 37  (63, 77]           183984 non-null  uint8
 38  (77, 91]           183984 non-null  uint8
 39  (91, 181]          183984 non-null  uint8
 40  Bajo Cauca         183984 non-null  uint8
 41  Magdalena Medio    183984 non-null  uint8
 42  Nordeste           183984 non-null  uint8
 43  Norte              183984 non-null  uint8
 44  Occidente          183984 non-null  uint8
 45  Oriente            183984 non-null  uint8
 46  Suroeste           183984 non-null  uint8
 47  Uraba              183984 non-null  uint8
 48  Valle del Aburra   183984 non-null  uint8
dtypes: float64(3), int32(3), int64(26), object(1), uint8(16)
memory usage: 48.4+ MB
```

Fuente: Tomado de GitLab, Elaboración propia

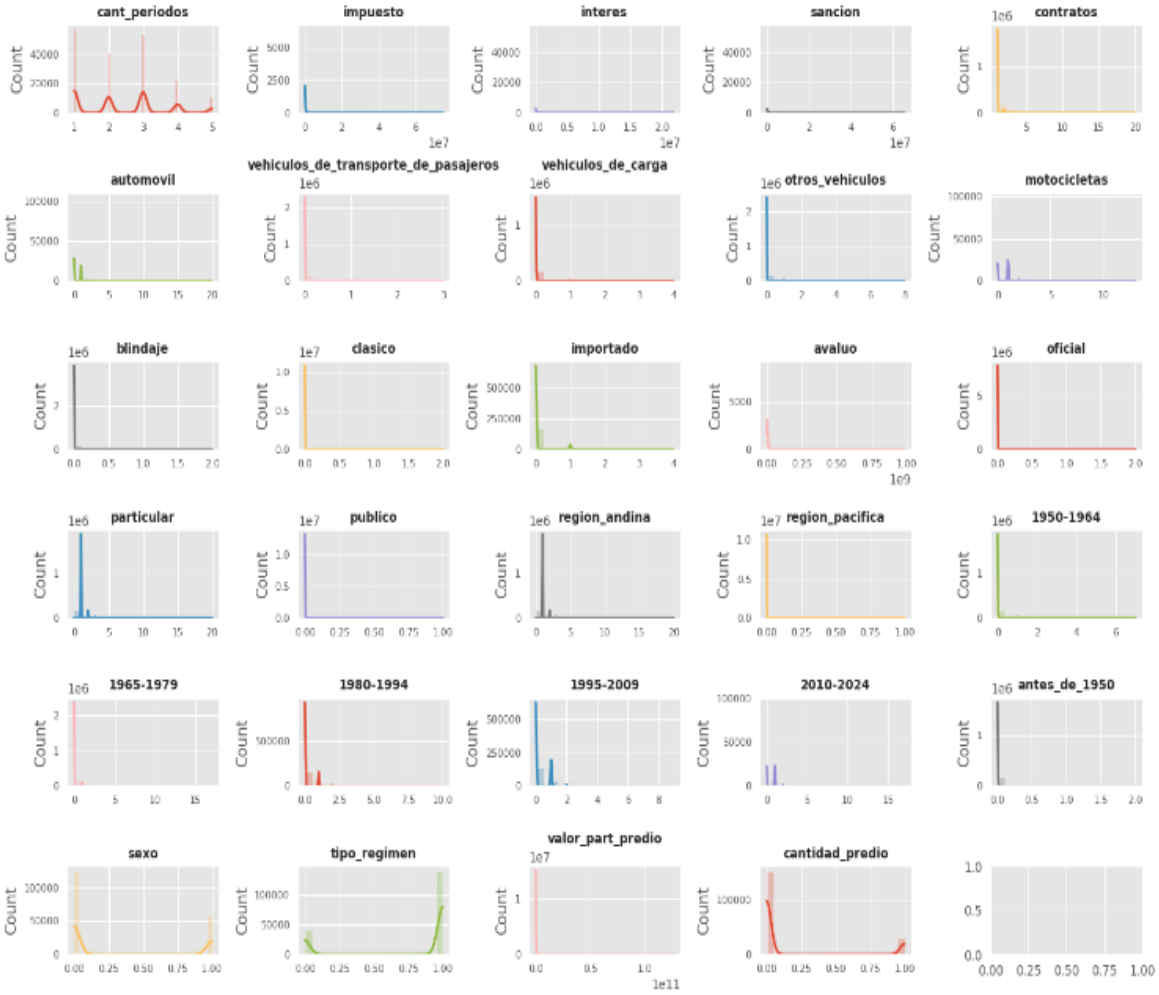
A continuación, se realiza un paso de limpieza de datos, donde se eliminan las variables que contienen únicamente valores cero. Esto se hace con el objetivo de evitar introducir ruido innecesario en el proceso de modelado.

Posteriormente, se procede a generar gráficos para visualizar la dinámica de las variables restantes. Estos gráficos proporcionan información visual sobre la distribución y comportamiento de cada variable numérica en el conjunto de datos. (Ver Figura 23)

Finalmente, se guarda el resultado de este proceso en un archivo en formato parquet llamado "caracterizacion. parquet", el cual se almacena en la laguna de datos "clean_lake". Este archivo contiene la caracterización de las variables numéricas del conjunto de datos, lo cual resulta útil para futuros análisis y modelado.

Figura 23

Distribución de variables numéricas



Fuente: Tomado de GitLab, Elaboración propia

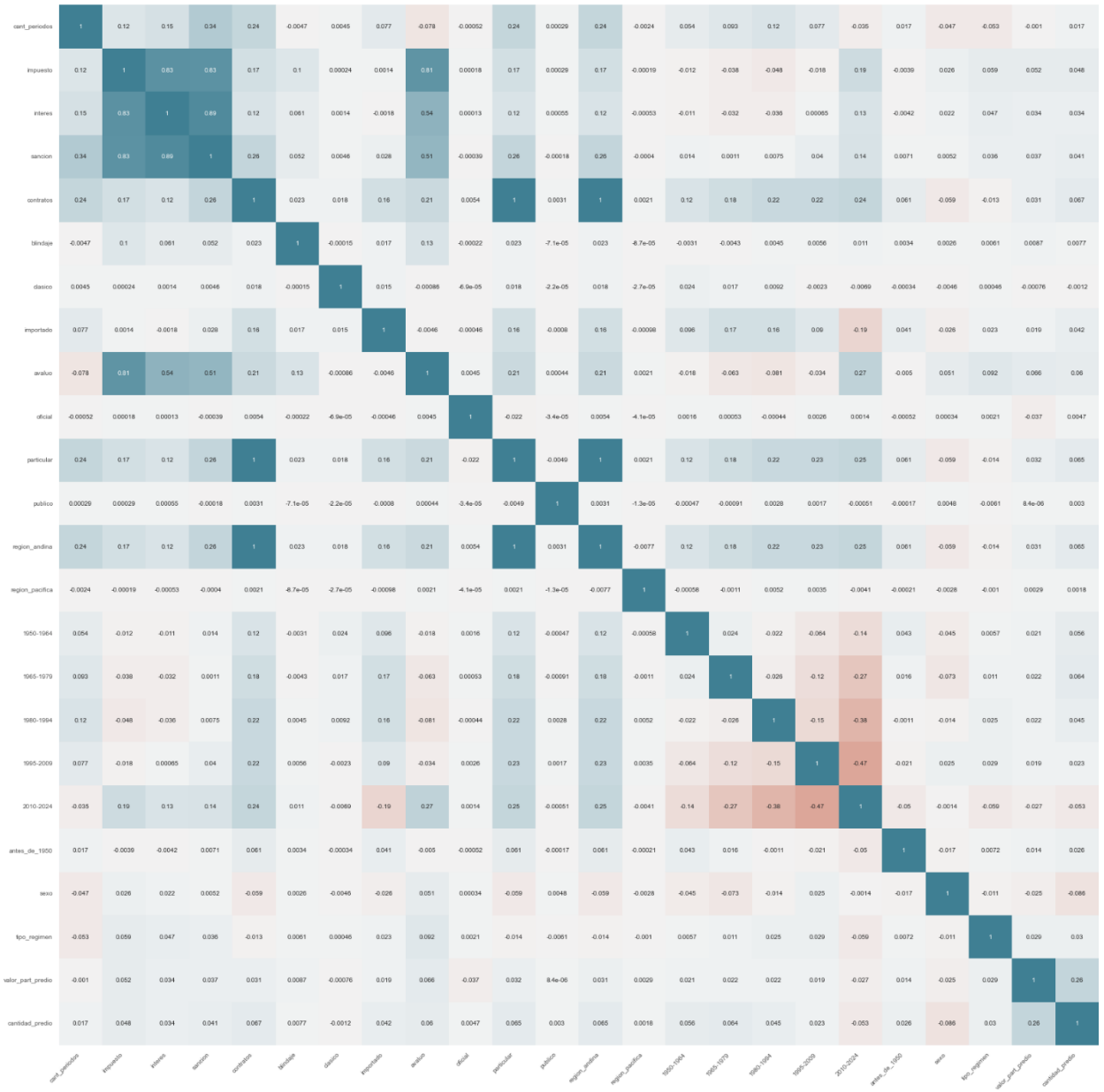
En el análisis de correlación, se utilizó el coeficiente de correlación de Pearson para evaluar la relación lineal entre variables en un conjunto de datos. Este coeficiente proporciona información sobre la fuerza y dirección de la relación, basándose en la covarianza entre las variables. Los valores de correlación de Pearson se normalizan y varían entre -1 y 1.

Después de examinar la matriz de correlación, se tomaron decisiones con respecto a las variables que mostraron una alta correlación. Específicamente, se encontró una alta correlación entre la variable "contratos" y la variable “particular”, así como entre la variable "contratos" y la variable "region_andina". Para evitar problemas de multicolinealidad, se decidió eliminar la variable "contratos" del análisis, además, se observó una alta correlación entre la variable "avalúo" y las variables "impuesto", "sanción" e "interés". Con el fin de simplificar el análisis y reducir la redundancia, se decidió eliminar la variable "avalúo".

Para obtener más detalles y visualizar estos hallazgos, puedes consultar la Figura 24, donde se presentan las correlaciones identificadas en la matriz.

Figura 24

Matriz de correlación



Fuente: Tomado de GitLab, Elaboración propia

FASE 4: MODELADO

- ➔ Seleccionar la técnica de modelado a aplicar, acorde con los datos y objetivos.

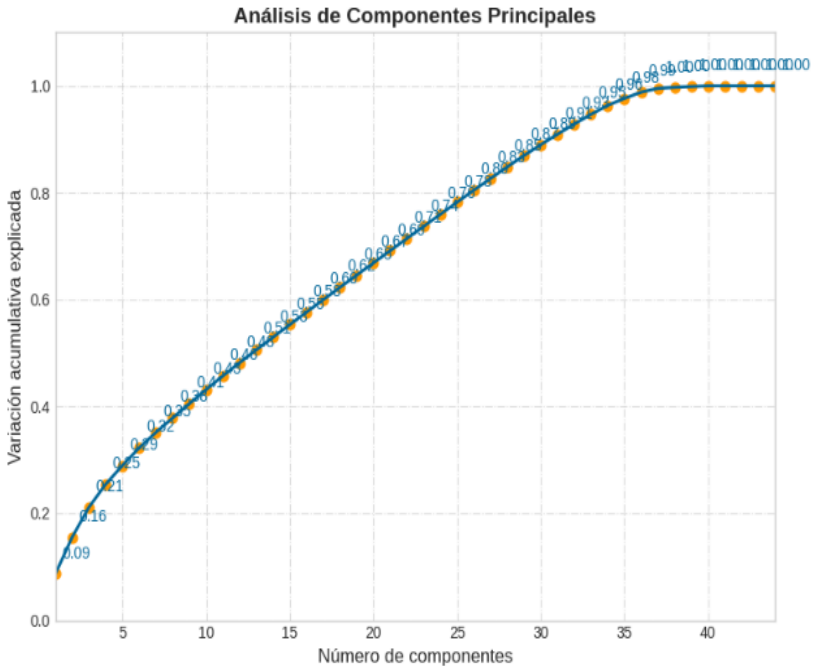
- ➔ Construir el modelo a partir de la aplicación de la técnica seleccionada.

En esta etapa se detecta que se cuenta con un conjunto de datos con una dimensión bastante considerable, para mitigar este efecto se intenta llevar a cabo un análisis de componentes principales (PCA). Este es un método que tiene como objetivo identificar las variables o características de los datos que aportan más información, pudiendo descartar aquellas menos relevantes, reduciendo así la dimensionalidad del conjunto de datos de trabajo. Es decir, se trata de una técnica de extracción de características donde se combinan las entradas de una manera específica, buscando entender qué relación existe entre ellas.

Al aplicar dicho método a nuestro ejercicio, se pudo determinar que el 80 % de los datos son explicados con 25 variables, lo que significa que se podía reducir el conjunto inicial de 49 variables a solo 25. Es decir, estas 25 variables deberían ser capaces de capturar la mayor parte de la variabilidad presente en los datos originales. Al comprender existe un compromiso importante entre la cantidad de información que se retiene y el número de variables que se utilizan, se concluyó que se podrían perder registros de información que se encuentra en las variables restantes.

Figura 25

Variación Acumulativa Explicada (PCA)



Fuente: Tomado de GitLab, Elaboración propia

Por lo tanto, y ante el riesgo de perder coherencia en los datos para su posterior modelamiento y análisis, se decide no realizar dicha reducción de dimensiones, esto permitirá conservar la información detallada y evitar la pérdida de registros importantes contenidos en las variables restantes. Aunque la alta dimensionalidad puede plantear desafíos en términos de análisis y visualización, se consideró que mantener todas las variables era fundamental para capturar la variabilidad completa presente en los datos.

Para el modelado se seleccionó la técnica no supervisada K-Means (K-Medias). El algoritmo de k-medias toma el parámetro de entrada, k, y divide un conjunto de n objetos en k grupos, de modo que la similitud intracluster resultante es alta pero la similitud entre clusters es baja. La similitud del grupo se mide en relación con el valor medio de los objetos en un grupo, que puede verse como el centroide o centro de gravedad del cúmulo. Este modelo selecciona aleatoriamente k de los objetos, cada uno de los cuales representa inicialmente un centro o media de grupo. Para cada uno de los objetos restantes, se asigna un objeto al conglomerado al que es más similar, en función de la distancia entre el objeto y la media del conglomerado. Luego calcula la nueva media para cada grupo. Este proceso itera hasta que la función de criterio converge. (Han y Kamber, 2006)

Figura 26

Ecuación K-Medias

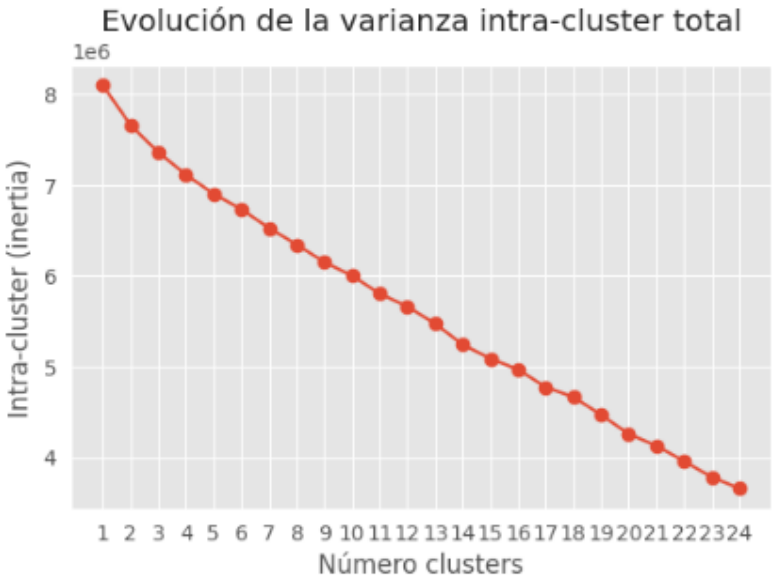
$$\underset{C}{\operatorname{argmin}} \sum_{j=1}^{k} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2$$

Fuente: Tomada de Han y Kamber, (2006)

Identificar número de clústeres: Para identificar la cantidad de clústeres o grupos a implementar en el modelo, se aplicó el método Elbow, El método del codo utiliza los valores de la ***inercia*** (la distancia media de las observaciones a su centroide). Es decir, se fija en las distancias intra-cluster. Cuanto más grande es el número de clusters k, la varianza intra-cluster tiende a disminuir. Cuanto menor sea la distancia intra-cluster mejor, ya que significa que los clusters son más compactos. El método del codo busca el valor k que satisfaga que un incremento de k no mejore sustancialmente la distancia media intra-cluster.

Figura 27

Método del codo



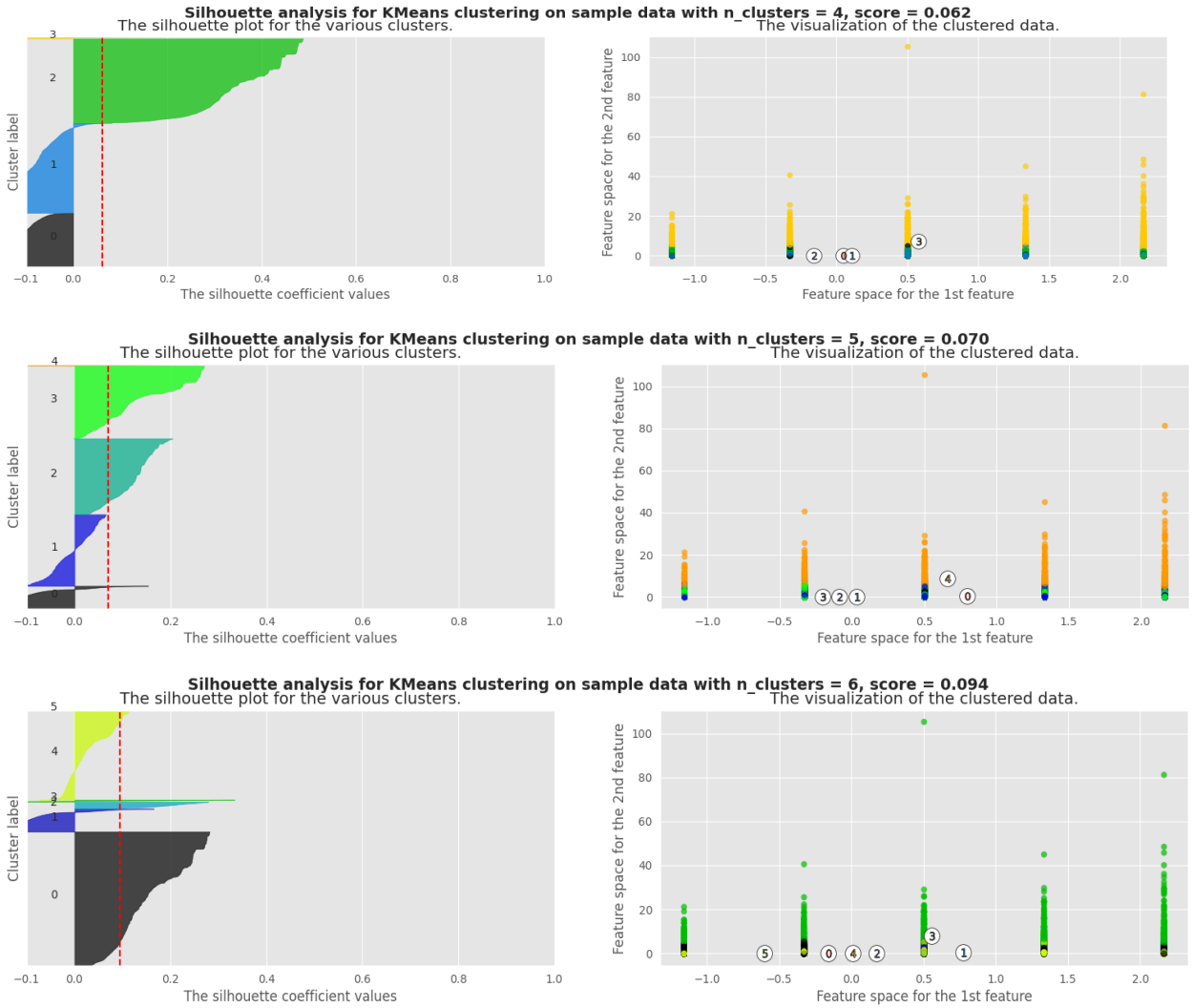
Fuente: Tomado de GitLab, Elaboración propia

Así también se evaluó el método de silueta que ha sido definido por von Luxburg (2020) como una evaluación de clustering que cuantifica la calidad de la asignación de los objetos a los clústeres. El enfoque de la silueta condensada utiliza una curva de silueta condensada para identificar el número óptimo de clústeres. La silueta condensada se calcula mediante una técnica de filtrado optimizado que elimina los clústeres redundantes. El objetivo principal del método propuesto es encontrar una representación compacta y significativa de los clústeres en el conjunto de datos, evitando la inclusión de clústeres redundantes o insignificantes.

Los coeficientes de silueta cercanos a +1 indican que la muestra está lejos de los clusters vecinos. Un valor de 0 indica que la muestra está muy cerca del límite de decisión entre dos conglomerados vecinos y los valores negativos indican que esas muestras podrían haber sido asignadas al conglomerado incorrecto.

Figura 28

Método del Silhouette



Fuente: Tomado de GitLab, Elaboración propia

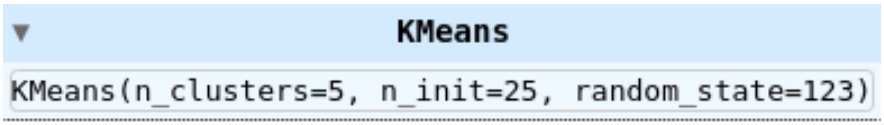
Al analizar tanto el método del codo como el método de silueta en la gráfica mencionada, se ha determinado que el número óptimo de clústeres es de 5. En la gráfica, se observa un cambio leve en la evolución de la inercia, donde la curva adquiere una forma similar a la de un brazo, y su punto de inflexión sugiere que 5 clústeres serían la cantidad adecuada. Sin embargo, debido a la falta de claridad en la interpretación visual, se decidió utilizar el método de silueta como una herramienta complementaria para obtener un valor más preciso.

Después de aplicar dicho método, se ha confirmado que efectivamente el número óptimo de clústeres es de 5. Esto significa que los datos pueden ser agrupados de manera más significativa y representativa mediante la formación de 5 clústeres distintos, para la implementación del modelo se utilizan los siguientes hiperparámetros:

- **n_clusters:** determina el número K de clusters que se van a generar.
- **n_init:** determina el número de veces que se va a repetir el proceso, cada vez con una asignación aleatoria inicial distinta. Es recomendable que este último valor sea alto, entre 10-25, para no obtener resultados subóptimos debido a una iniciación poco afortunada del proceso.
- **random_state:** semilla para garantizar la reproducibilidad de los resultados, para este modelo se utiliza 123 como semilla

Figura 29

Hiperparámetros



Fuente: Tomado de GitLab, Elaboración propia

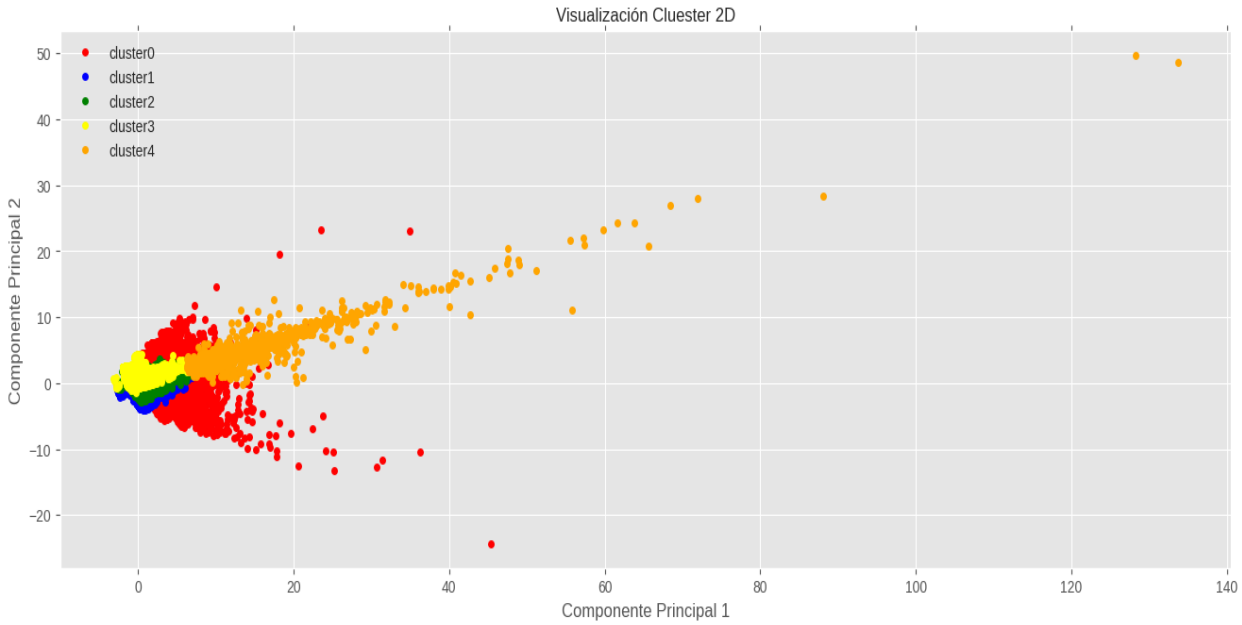
Respecto a la dimensionalidad del presente proyecto, se realizó una indagación buscando una forma óptima de representar visualmente los clústeres reduciendo a 2 o 3 dimensiones, esto con el fin de adelantar una exploración inicial de los datos y obtener una idea general de su estructura. La representación gráfica se obtiene mediante la función visualization2d, que utiliza el análisis de (PCA) para reducir las dimensiones de los datos a dos componentes principales. Cada

punto en la gráfica representa un dato del conjunto de datos, y se utiliza un código de colores para distinguir los diferentes clústeres.

Se puede apreciar que los puntos que pertenecen al “cluster 0” y al “cluster4” están más dispersos, lo que indica una mayor variabilidad en los datos de esos clusters. En contraste, los puntos de los otros clústeres se encuentran más cercanos entre sí, lo que indica una mayor cohesión y una menor dispersión de los datos dentro de cada clúster. (Ver Figura 30)

Figura 30

Visualización 2D de los clústeres



Fuente: Tomado de GitLab*,* Elaboración propia

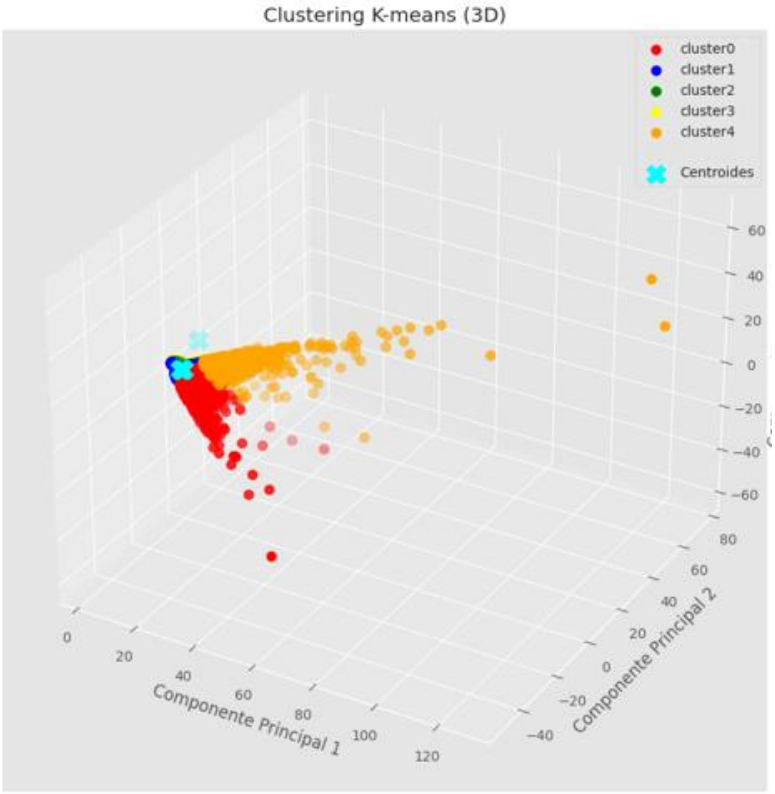
En la Figura 31 se observa la gráfica generada por la función visualization3d, en esta se puede apreciar la presencia de diferentes clusters que no se solapan entre sí. Además, los centroides de cada cluster se encuentran ubicados muy cerca de los puntos que representan a los clusters correspondientes. Este resultado es esperable, ya que los centroides son calculados como la media de los datos dentro de cada clúster.

La gráfica se representa en un espacio tridimensional, donde los ejes representan las tres componentes principales obtenidas mediante el análisis de PCA (Análisis de Componentes Principales). Cada cluster se visualiza mediante puntos de diferentes colores, y los centroides se representan como puntos marcados con una "X" y en un color distinto.

Además, se ha ajustado el rango de los ejes para tener una escala uniforme y facilitar la visualización de los clusters y sus centroides.

Figura 31

Visualización 3D de los clústeres



Fuente: Tomado de GitLab, Elaboración propia

FASE 5: EVALUACIÓN MODELO

- ➔ Determinar las características de los patrones obtenidos al clasificar los contribuyentes.

- ➔ Realizar revisión de los resultados obtenidos con el modelo construido respecto a los propósitos definidos.

Al realizar la caracterización en base a los 5 clusters seleccionados, se encontraron las siguientes condiciones:

Cluster 0: Deudores Habitualmente Morosos

Este cluster representa el 15.96% de la deuda total, compuesta en un 34.89 % por concepto de impuesto, el 57.72 % corresponde a sanción y el 7.40% a interés de la deuda. La cantidad de contribuyentes registrados en este cluster es de 16.919, que representan un 9.20 % del universo total de contribuyentes incluidos en el modelo. Estos se encuentran distribuidos en un 76.85 % por hombres y el 23.15 % por mujeres. El 75.04 % de los deudores se encuentran afiliados al régimen contributivo y el 24.96 % al régimen subsidiado de salud. Se identifica que, del total de obligados del impuesto definidos en este cluster, el 23.6 % son propietarios de por lo menos un inmueble.

El 37.44 % de los deudores se encuentran en un rango de edad entre los 35 y 49 años, quienes provienen en su mayoría del valle de Aburrá y el bajo cauca antioqueño. Del total de deudores de este cluster, el 38.90 % acumulan 3 periodos de impuestos en mora, el 27.43 % tienen 4 periodos, el 15.51 % acumulan 5 periodos y el 18.17 % poseen entre 1 y 2 periodos.

Se identificaron para este cluster un total de 37.362 vehículos que representan el 18.21% del total del universo de vehículos incluidos en el modelo. De estos, el 50.88 % son motocicletas, el 44.55 % son automóviles, el 2.36 % son otros vehículos; así también el 1.76 % corresponde a

vehículos de carga y el 0.44 % son vehículos de transporte de pasajeros. Del total de vehículos vinculados al cluster el 14.07 % son vehículos importados; además el 42.29 % son vehículos de modelos entre el 2010 y el 2024, el 25.68 % son de modelos entre 1995 y 2009, el 18.31 % son de modelos entre 1980 y 1994 y el 10.09 % son modelos entre 1965 y 1979.El 3.63 % de los vehículos son de modelos anteriores a 1965.

Se logró identificar que el 0.12 % de los vehículos del cluster son blindados y el 0.03 % son clásicos, el 99.85 % no poseen ninguna de estas características.

Cluster 1: Deudores con Deuda Sostenida

Este cluster representa el 22.92% de la deuda total, compuesta en un 34.82 % por concepto de impuesto, el 57.21 % corresponde a sanción y el 7.97% a interés de la deuda. La cantidad de contribuyentes registrados en este cluster es de 53.777, que representan un 9.20 % del universo total de contribuyentes incluidos en el modelo. Estos se encuentran distribuidos en un 64.69 % por hombres y el 35.31 % por mujeres. El 79.22 % de los deudores se encuentran afiliados al régimen contributivo y el 20.78 % al régimen subsidiado de salud. Se identifica que, del total de obligados del impuesto definidos en este cluster, el 15.44 % son propietarios de por lo menos un inmueble.

El 52.47 % de los deudores se encuentran en un rango de edad entre los 49 y 63 años, quienes provienen en su mayoría del valle de Aburrá y el oriente antioqueño. Del total de deudores de este cluster, el 30.65 % acumulan 3 periodos de impuestos en mora, el 26.97 % tienen 1 periodo, el 25.52 % acumulan 2 periodos y el 16.85 % poseen entre 4 y 5 periodos.

Se identificaron para este cluster un total de 53.644 vehículos que representan el 26.15% del total del universo de vehículos incluidos en el modelo. De estos, el 73.61 % son automóviles, el 22.63 % son motocicletas, el 2.07 % son vehículos de carga; así también el 1.15 % corresponde a otros vehículos y el 0.53 % son vehículos de transporte de pasajeros. Del total de vehículos vinculados al cluster el 10.76 % son vehículos importados; además el 10.16 % son vehículos de

modelos entre el 2010 y el 2024, el 37.12 % son de modelos entre 1995 y 2009, el 31.08 % son de modelos entre 1980 y 1994 y el 16.96 % son modelos entre 1965 y 1979.El 4.98 % de los vehículos son de modelos anteriores a 1965.

Se logró identificar que el 0.04 % de los vehículos del cluster son blindados y el 0.01 % son clásicos, el 99.95 % no poseen ninguna de estas características.

Cluster 2: Deudores Recientes

Este clúster representa el 15.96% de la deuda total, compuesta en un 37.67 % por concepto de impuesto, el 54.13 % corresponde a sanción y el 8.20% a interés de la deuda. La cantidad de contribuyentes registrados en este cluster es de 57.221, que representan un 29.24 % del universo total de contribuyentes incluidos en el modelo. Estos se encuentran distribuidos en un 69.11 % por hombres y el 30.89 % por mujeres. El 78.46 % de los deudores se encuentran afiliados al régimen contributivo y el 21.54 % al régimen subsidiado de salud. Se identifica que, del total de obligados del impuesto definidos en este cluster, el 23.50 % son propietarios de por lo menos un inmueble.

El 99.9 % de los deudores se encuentran en un rango de edad entre los 35 y 49 años, quienes provienen en su mayoría del valle de Aburrá y el bajo cauca antioqueño. Del total de deudores de este cluster, el 34.21 % acumulan 1 periodo de impuestos en mora, el 28.08 % tienen 3 periodos, el 22.52 % acumulan 2 periodos y el 15.19 % poseen entre 1 y 2 periodos.

Se identificaron para este cluster un total de 57.195 vehículos que representan el 27.88% del total del universo de vehículos incluidos en el modelo. De estos, el 60.05 % son motocicletas, el 37.53 % son automóviles, el 1.58 % son otros vehículos; así también el 0.72 % corresponde a vehículos de carga y el 0.11 % son vehículos de transporte de pasajeros. Del total de vehículos vinculados al cluster el 3.37 % son vehículos importados; además el 58.32 % son vehículos de modelos entre el 2010 y el 2024, el 27.03 % son de modelos entre 1995 y 2009, el 10.01 % son

de modelos entre 1980 y 1994 y el 3.50 % son modelos entre 1965 y 1979.El 1.13 % de los vehículos son de modelos anteriores a 1965.

Se logró identificar que el 0.07 % de los vehículos del cluster son blindados y el 0.01 % son clásicos, el 99.93 % no poseen ninguna de estas características.

Cluster 3: Deudores Recientes con Bajo Monto

Este cluster representa el 20.26 % de la deuda total, compuesta en un 35.65 % por concepto de impuesto, el 57.15 % corresponde a sanción y el 7.20% a interés de la deuda. La cantidad de contribuyentes registrados en este cluster es de 54.736, que representan un 31.11 % del universo total de contribuyentes incluidos en el modelo. Estos se encuentran distribuidos en un 68.02 % por hombres y el 31.96 % por mujeres. El 73.98 % de los deudores se encuentran afiliados al régimen contributivo y el 26.02 % al régimen subsidiado de salud. Se identifica que, del total de obligados del impuesto definidos en este cluster, el 7.47 % son propietarios de por lo menos un inmueble.

El 73.8 % de los deudores se encuentran en un rango de edad entre los 21 y 35 años, quienes provienen en su mayoría del valle de Aburrá y el Urabá antioqueño. Del total de deudores de este cluster, el 40.29 % acumulan 1 periodo de impuestos en mora, el 25.72 % tienen 3 periodos, el 20.98 % acumulan 2 periodos y el 13.1 % poseen entre 4 y 5 periodos.

Se identificaron para este cluster un total de 55.032 vehículos que representan el 26.83% del total del universo de vehículos incluidos en el modelo. De estos, el 88.25 % son motocicletas, el 10.29 % son automóviles, el 1.39 % son otros vehículos; así también el 0.07 % son vehículos de carga. Del total de vehículos vinculados al cluster el 0.65 % son vehículos importados; además el 94.48 % son vehículos de modelos entre el 2010 y el 2024, el 5.32 % son de modelos entre 1995 y 2009, el 0.16 % son de modelos entre 1980 y 1994 y el 0.03 % son modelos entre 1965 y 1979.

Se logró identificar que el 0.01 % de los vehículos del cluster son blindados, el 99.99 % no poseen ninguna característica.

Cluster 4: Deudores con Alto Monto

Este cluster representa el 14.89 % de la deuda total, compuesta en un 45.82 % por concepto de impuesto, el 38.34 % corresponde a sanción y el 15.84% a interés de la deuda. La cantidad de contribuyentes registrados en este cluster es de 12.51, que representan un 29.76% del universo total de contribuyentes incluidos en el modelo. Estos se encuentran distribuidos en un 67.63 % por hombres y el 32.37 % por mujeres. El 92.41 % de los deudores se encuentran afiliados al régimen contributivo y el 7.59 % al régimen subsidiado de salud. Se identifica que, del total de obligados del impuesto definidos en este cluster, el 32.37 % son propietarios de por lo menos un inmueble.

El 50.44 % de los deudores se encuentran en un rango de edad entre los 35 y 49 años, quienes provienen en su mayoría del valle de Aburrá y el bajo cauca antioqueño. Del total de deudores de este cluster, el 23.50 % acumulan 5 periodos de impuestos en mora, el 22.30 % tienen 3 periodos, el 20.62 % acumulan 4 periodos y el 33.58 % poseen entre 1 y 2 periodos.

Se identificaron para este cluster un total de 1.913 vehículos que representan el 0.93% del total del universo de vehículos incluidos en el modelo. De estos, el 86.93 % son automóviles, el 10.09 % son motocicletas, el 1.93 % son otros vehículos; así también el 0.94 % corresponde a vehículos de carga y el 0.10 % son vehículos de transporte de pasajeros. Del total de vehículos vinculados al cluster el 4.48 % son vehículos importados; además el 87.66 % son vehículos de modelos entre el 2010 y el 2024, el 9.25 % son de modelos entre 1995 y 2009, el 2.20 % son de modelos entre 1980 y 1994 y el 0.68 % son modelos entre 1965 y 1979.El 0.21 % de los vehículos son de modelos anteriores a 1965.

Se logró identificar que el 2.77 % de los vehículos del cluster son blindados, el 97.23 % no poseen ninguna característica.

10. Conclusiones

El presente trabajo de investigación se ha centrado en el análisis y comprensión del perfil de los contribuyentes deudores del impuesto vehicular en el departamento de Antioquia, utilizando técnicas de Aprendizaje Automático. Este representó un reto desde el aspecto técnico y procedimental, ya que debido a la cantidad de set de datos obtenidos y la gran densidad de registros contenidos en estos, se debió focalizar el esfuerzo principalmente en la preparación y transformación de los datos. Para esto se estableció inicialmente una metodología que permitiera enfocar el desarrollo del proyecto, siguiendo sus fases establecidas estructuralmente y generando solución a los objetivos planteados.

La etapa de comprensión del negocio también significó un gran desafío debido a que solo se obtuvieron algunos diccionarios sobre los sets de datos a utilizar durante el proyecto. Esto significó el tener que definir funciones como profile_dataframe para poder tener un acercamiento a los registros de cada variable.

En el proceso de preparación de los datos se realizaron múltiples actividades para definir los registros óptimos para el despliegue del modelo. Entre estas se encuentra el desarrollo de un repositorio de código fuente en GitLab con el objetivo de mantener los datos sincronizados y de esta manera brindar un entorno centralizado para gestionarlos y organizarlos de manera eficiente. Esto permitió la consolidación de toda la data, la limpieza de datos y detección de anomalías, eliminando registros debido a sus valores nulos por medio de librerías establecidas para esta función, asimismo se desarrolló una transformación de diferentes variables en aras de aumentar el nivel de calidad de los datos; así también fue posible realizar una estructuración de uniones de set de datos homónimos para visualizar los atributos relevantes para nuestro desarrollo.

Debido a la alta dimensionalidad de los registros, se requirió definir un análisis de componentes principales (PCA), por medio del cual determinar el número de elementos óptimo para explicar la información en un 80%, pero al analizar que aún seguía presentándose un alto número de variables y se podía perder información importante, dado esto se definió solo visualizar el modelo en 2D y 3D reduciendo sus dimensiones. Para realizar el modelamiento, se definió el uso del algoritmo de aprendizaje automático no supervisado K-Means, donde después de establecer el número de clusters adecuado, se hallaron 5 grupos o categorías que explican ciertas características útiles para que, el área de cartera de impuesto vehicular pueda encaminar sus procesos de cobranza.

11. Recomendaciones

Se recomienda la implementación de más fuentes de datos al proceso con el fin de poder obtener información de contribuyentes que se encuentren por fuera del departamento de Antioquia.

Evaluación y selección de características: Dado que los conjuntos de datos de alta dimensionalidad suelen contener características irrelevantes o redundantes, se recomienda investigar técnicas de selección de características que permitan identificar las características más relevantes para mejorar el rendimiento de K-means.

Mejoras en el rendimiento del algoritmo: El algoritmo K-means puede volverse ineficiente cuando se aplica a conjuntos de datos de alta dimensionalidad. En este sentido, se propone explorar técnicas para mejorar la eficiencia y la escalabilidad del algoritmo, como la reducción de dimensionalidad previa o la adaptación del algoritmo para lidiar con características redundantes o irrelevantes.

Implementar un modelo de clusterización más robusto que k-means, que pueda enfrentarse de una manera óptima a los problemas de alta dimensionalidad, se propone la implementación del algoritmo DBSCAN.

Visualización de resultados: La visualización de resultados en conjuntos de datos de alta dimensionalidad puede ser un desafío. Se recomienda investigar técnicas de visualización que permitan representar de manera efectiva los grupos generados por K-means en espacios de alta dimensión. Esto puede implicar técnicas de reducción de dimensionalidad, como UMAP o t-SNE, seguidas de visualizaciones en 2D o 3D.

Efectuar un modelo de clusterización sobre la data segmentada por el valle de Aburra debido a que la mayor cantidad de registros resultantes están georreferenciados en esta zona.

Referencias

Asamblea Departamental de Antioquia. (26 de agosto de 2022). Ordenanza. *Estatuto de rentas del departamento de Antioquia*, 44. Medellín, Antioquia, Colombia: Gazeta Departamental.

B. Sekhar Babu, P. Lakshmi Prasanna, & P. Vidyullatha. (2018). Customer Data Clustering using Density based algorithm. *International Journal of Engineering & Technology*, 35-38. doi:10.14419/ijet. v7i2.32.13520

Barone, Guglielmo, & Mocetti, Sauro. (2011). Tax morale and public spending inefficiency. *International Tax and Public Finance*, 724-749. Obtenido de https://doi.org/10.1007/s10797-011-9174-z

Cahyana, B. E., Nimran, U., Utami, H. N., & Iqbal, M. (2020). Hybrid cluster analysis of customer segmentation of sea transportation users. *Journal of Economics, Finance and Administrative Science*, 321–337. Obtenido de https://doi.org/10.1108/JEFAS-07-2019-0126

DNP. (2021). *Principales Ingresos Tributarios Departamentales [Gráfico].* Obtenido de DNP: https://colaboracion.dnp.gov.co/CDT/Desarrollo%20Territorial/Desempeno_Fiscal/Resultados_Nuevo_IDF_2021.pdf

Dr.B. Arivazhagan, & Dr.G.Vijaiprabhu. (2022). An Enhanced Hierarchical Model for Customer Segmentation in Customer Relationship Management with Demographic, Recency, Frequency and Monetary Values. *International Journal of Mechanical Engineering*, 1878-1886.

Federico Carlos Peralta. (2014). Proceso de Conceptualización del Entendimiento del Negocio para Proyectos de Explotación de Información. *Revista Latinoamericana de Ingeniería de Software*, 273-306.

Haya, P. (2022). *www.iic.uam.es*. Obtenido de La metodología CRISP-DM en ciencia de datos: https://www.iic.uam.es/innovacion/metodologia-crisp-dm-ciencia-de-datos/

Hossain, M., Akter, S., & Yanamandram, V. (2020). Customer Analytics Capabilities in the Big Data Spectrum. *Technological Innovations for Sustainability and Business Growth*, 1-17. doi: doi:http://dx.doi.org/10.4018/978-1-5225-9940-1.ch001

IBM. (2022). Obtenido de What is the k-nearest neighbors algorithm?.

IBM. (s.f.). *IBM*. Obtenido de https://www.ibm.com/topics/knn

Jiawei Han, & Micheline Kamber. (2006). Data Mining: Concepts and Techniques. *University of Illinois at Urbana-Champaign*.

Kelly, S. (2003). Mining data to discover customer segments. *Interactive Marketing*, 235–242. Obtenido de https://doi.org/10.1057/palgrave.im.4340185

Luis Prada Conde. (2022). *Aplicación de técnicas de clustering como paso previo a la detección de anomalías en redes definidas por software.* A Coruña.

Michael Pickhardt a, & Aloys Prinz b. (2014). Behavioral dynamics of tax evasion – A survey. *Journal of Economic Psychology*, 1-19.

Nájera, J. (2022). *Fisco*. Obtenido de ACTIUN: https://www.actiun.com/glosario/fisco

Omran, M. G. H, Engelbrecht, A. P., & Salman, A. (2007). An overview of clustering methods. *In Intelligent Data Analysis*, 583-605. Obtenido de https://doi.org/10.3233/ida-2007-11602

Qadadeh, W., & Abdallah, S. (2018). Customers Segmentation in the Insurance Company (TIC) Dataset. *Procedia Computer Science*, 277-290. Obtenido de https://doi.org/10.1016/j.procs.2018.10.529

R.M. Moreno-Carriles. (2018). Big data, ¿pero ¿qué es? *Angiología*, 191-194.

Silva, D. M. B., Pereira, G. H. A., & Magalhães, T. M. (2022). A class of categorization methods for credit scoring models. *European Journal of Operational Research*, 323-331. Obtenido de https://doi.org/10.1016/j.ejor.2021.04.029.

Verdenhofs, A., & Tambovceva, T. (2019). Evolution of Customer Segmentation in the Era of Big Data. *Marketing and Management of Innovations, Marketing and Management of Innovations*, 238-243. Obtenido de doi:http://doi.org/10.21272/mmi.2019.1-20

von Luxburg, U. (2020). Good (K-means) clusterings are unique (up to small perturbations). *Journal of Machine Learning Research*, 2903-2904.

Weinstein, A. (2001). Customer-Specific Strategies Customer Retention: A Usage Segmentation and Customer Value Approach. *ournal of Targeting, Measurement and Analysis for Marketing*.

Zhang, L., Priestley, J., Demaio, J., Ni, S., & Tian, X. (2021). Measuring Customer Similarity and Identifying Cross-Selling Products by Community Detection. *Big Data*, 132-143. Obtenido de https://doi.org/10.1089/big.2020.0044

Zhou, B., Lu, B., & Saeidlou, S. (2022). A Hybrid Clustering Method Based on the Several Diverse Basic Clustering and Meta-Clustering Aggregation Technique. *Cybernetics and Systems*, 1–27. Obtenido de https://doi.org/10.1080/01969722.2022.2110682